

# Wav2vec-S: Semi-Supervised Pre-Training for Low-Resource ASR

Han Zhu<sup>1,2</sup>, Li Wang<sup>1</sup>, Jindong Wang<sup>3</sup>, Gaofeng Cheng<sup>1</sup>, Pengyuan Zhang<sup>1,2</sup>, Yonghong Yan<sup>1,2</sup>

<sup>1</sup>Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics CAS, China

<sup>2</sup>University of Chinese Academy of Sciences, China <sup>3</sup>Microsoft Research Asia, China

zhuhan@hcccl.ioa.ac.cn

## Abstract

Self-supervised pre-training could effectively improve the performance of low-resource automatic speech recognition (ASR). However, existing self-supervised pre-training are *task-agnostic*, i.e., could be applied to various downstream tasks. Although it enlarges the scope of its application, the capacity of the pre-trained model is not fully utilized for the ASR task, and the learned representations may not be optimal for ASR. In this work, in order to build a better pre-trained model for low-resource ASR, we propose a pre-training approach called *wav2vec-S*, where we use *task-specific* semi-supervised pre-training to refine the self-supervised pre-trained model for the ASR task thus more effectively utilize the capacity of the pre-trained model to generate task-specific representations for ASR. Experiments show that compared to wav2vec 2.0, wav2vec-S only requires a marginal increment of pre-training time but could significantly improve ASR performance on in-domain, cross-domain and cross-lingual datasets. Average relative WER reductions are 24.5% and 6.6% for 1h and 10h fine-tuning, respectively. Furthermore, we show that semi-supervised pre-training could close the representation gap between the self-supervised pre-trained model and the corresponding fine-tuned model through canonical correlation analysis.

**Index Terms:** pre-training, self-supervised learning, semi-supervised learning, speech recognition, wav2vec 2.0

## 1. Introduction

The performance of automatic speech recognition (ASR) heavily relies on the amount of labeled data, which is costly and not available in many low-resource scenarios. To alleviate this issue, the self-supervised learning [1, 2, 3, 4, 5, 6, 7] can be used to build the self-supervised pre-trained model with massive unlabeled data. However, since the self-supervised pre-training is *task-agnostic*, the capacity of the pre-trained model is not fully utilized for ASR. And the representations of the self-supervised pre-trained model may not be optimal for ASR [8]. As an alternative, conventional transfer learning approaches [9, 10, 11, 12, 13, 14] typically build a supervised pre-trained model with labeled data in the high-resource domain. Since the labeled data provide task-specific information, the supervised pre-trained model is *task-specific* to ASR. However, since a considerable amount of unlabeled data are unused, the supervised pre-trained model could be unsatisfactory in performance.

In order to build a better pre-trained model for ASR, we propose a simple pre-training pipeline: *wav2vec-S*, which uses both labeled and unlabeled data to learn the ASR task-specific representations. Specifically, on the basis of the *task-agnostic self-supervised* pre-training, we further conduct *task-specific*

*semi-supervised* pre-training to learn task-specific representation. The reason we use semi-supervised pre-training instead of supervised pre-training is that the amount of labeled data is limited. Since the unlabeled data in semi-supervised pre-training is utilized through pseudo-labeling [15, 16, 17, 18], the semi-supervised pre-training is also learning task-specific representations. The two steps in wav2vec-S, i.e., self-supervised and semi-supervised pre-training, are loosely coupled. Thus the same strategy for semi-supervised pre-training can be used on the basis of different self-supervised pre-training approaches.

Experiments show that wav2vec-S consistently improves the self-supervised model thus could act as the alternative to the vanilla self-supervised model for the downstream ASR task. Moreover, we performed detailed ablation studies for the semi-supervised pre-training step in wav2vec-S and the main conclusions are as follows:

- Semi-supervised pre-training can improve the performance and generalization of the self-supervised pre-trained model, i.e., improvements on in-domain, cross-domain and cross-lingual datasets.
- Character-level supervision is better than phone-level for monolingual semi-supervised pre-training even on a cross-lingual downstream dataset, which could alleviate the efforts to generate the phoneme transcriptions.
- Monolingual semi-supervised pre-training has a trade-off between performance of the source language and other languages. With more training updates, the model would become more language-specific, and the cross-lingual generalization ability is thus degraded.
- The semi-supervised pre-training step costs much less time than self-supervised pre-training. Thus wav2vec-S only has a marginal increment of pre-training time than vanilla self-supervised pre-training.
- Semi-supervised pre-training effectively improves different self-supervised pre-trained models, e.g., wav2vec 2.0 [1], data2vec [5].
- We analyze the representation similarity before and after fine-tuning for pre-trained models with canonical correlation analysis (CCA), and show semi-supervised pre-training closes the representation gap between the pre-trained and fine-tuned models.

## 2. Related works

The idea to adapt the task-agnostic self-supervised pre-trained model to an ASR task-specific pre-trained model is also explored in other works. [19] forces the model to concentrate on ASR-related information by adding the self-supervised losses on intermediate layers. Other works [20, 21, 22, 23] utilize labeled data to inject ASR task information into the pre-trained model. Our work belongs to this category. Among them, Unispeech [20] uses multi-task learning to conduct semi-supervised

This work is partially supported by the National Key Research and Development Program of China (No. 2020AAA0108002).

pre-training, where contrastive loss is used on the unlabeled data and CTC loss is used on the labeled data. JUST [21] jointly optimizes two self-supervised losses and a supervised RNN-T loss. XLST [22] uses supervised training as the initialization and then conducts self-training on the unlabeled data. In our work, CTC loss is used on both labeled and unlabeled data, where the ground-truth labels are used for labeled data and pseudo labels are used for unlabeled data. Since previous work mostly conducts task-specific pre-training from scratch, substantial training time is required for each task. In this work, we treat semi-supervised pre-training as the task-specific refinement of the self-supervised pre-training. Thus, it can benefit from the initialization of the self-supervised pre-trained model for faster convergence. Concurrent works [24, 25] also explored the combination of self-supervised pre-training and semi-supervised learning, where [24] focused on the domain adaptation and [25] focused on the large-scale applications.

### 3. Proposed approach

#### 3.1. Problem Formulation

We denote the pre-training and fine-tuning dataset as the source domain  $\mathcal{S}$  and target domain  $\mathcal{T}$ . Suppose the source domain consists of an unlabeled dataset  $\mathbb{U}^{\mathcal{S}} = \{\mathbf{x}_1^{\mathcal{S}}, \dots, \mathbf{x}_N^{\mathcal{S}}\}$  and a labeled dataset  $\mathbb{L}^{\mathcal{S}} = \{(\mathbf{x}_1^{\mathcal{S}}, \mathbf{y}_1^{\mathcal{S}}), \dots, (\mathbf{x}_M^{\mathcal{S}}, \mathbf{y}_M^{\mathcal{S}})\}$ , where  $M \leq N$ . These two datasets are used for pre-training, where the self-supervised pre-training uses  $\mathbb{U}^{\mathcal{S}}$  and semi-supervised pre-training uses both  $\mathbb{U}^{\mathcal{S}}$  and  $\mathbb{L}^{\mathcal{S}}$ . Fine-tuning is performed on an additional target domain labeled dataset  $\mathbb{L}^{\mathcal{T}} = \{(\mathbf{x}_1^{\mathcal{T}}, \mathbf{y}_1^{\mathcal{T}}), \dots, (\mathbf{x}_O^{\mathcal{T}}, \mathbf{y}_O^{\mathcal{T}})\}$

#### 3.2. Model Structure

We adopt the model structure in wav2vec 2.0 [1], which is shown in the first step of Fig. 1. A convolutional feature encoder is used to map the input raw audio  $\mathcal{X}$  to higher-level latent speech representations  $\mathcal{Z}$ , which is then fed to the transformer context network to build context representations  $\mathcal{C}$ . During pre-training or fine-tuning, the mask module masked a proportion of the feature encoder outputs  $\mathcal{Z}$  to  $\mathcal{Z}'$  before feeding them into the context network. Note that the masked dimension is only the time dimension during self-supervised pre-training, while it consists of both time and channel dimensions during semi-supervised pre-training and fine-tuning like SpecAugment [26]. The masked time steps are denoted as gray color in Fig. 1.

#### 3.3. Wav2vec-S

We illustrate the wav2vec-S procedure in Fig. 1. The pre-training consists of two steps. Firstly, self-supervised pre-training is performed on the unlabeled source dataset  $\mathbb{U}^{\mathcal{S}}$  using the self-supervised loss. Then, semi-supervised pre-training is applied on both labeled source dataset  $\mathbb{L}^{\mathcal{S}}$  and unlabeled source dataset  $\mathbb{U}^{\mathcal{S}}$ . The total loss for semi-supervised learning is:

$$\mathcal{L}_{\text{semi}} = \mathcal{L}_{\text{label}} + \lambda \mathcal{L}_{\text{unlabel}}, \quad (1)$$

where  $\mathcal{L}_{\text{label}}$  and  $\mathcal{L}_{\text{unlabel}}$  denotes the loss for labeled and unlabeled data, respectively.  $\lambda$  is the hyperparameter to be tuned, which is fixed to 1 in this work.

Specifically, for ASR, during semi-supervised pre-training, both labeled and unlabeled losses are CTC [27]. For labeled

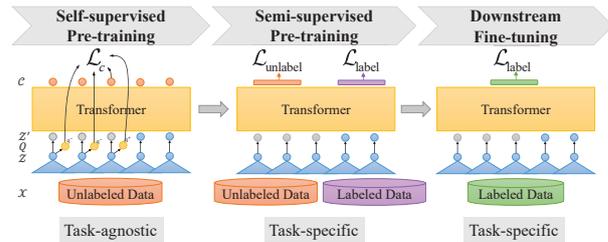


Figure 1: Illustration of the wav2vec-S procedure.

data, it is straightforward to compute the CTC loss as:

$$\mathcal{L}_{\text{label}} = -\mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p(\mathbf{x}, \mathbf{y})} \log p_{\theta}(\mathbf{y} | a(\mathbf{x})), (\mathbf{x}, \mathbf{y}) \in \mathbb{L}^{\mathcal{S}} \quad (2)$$

where  $(\mathbf{x}, \mathbf{y})$  is the sample-label pair,  $p(\mathbf{x}, \mathbf{y})$  is the distribution of samples from  $\mathbb{L}^{\mathcal{S}}$ ,  $\theta$  is the model parameter, and  $a(\cdot)$  is the augmentation function.

However, for the unlabeled data, since the ground-truth labels are not available, pseudo labels are used instead:

$$\mathcal{L}_{\text{unlabel}} = -\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \log p_{\theta}(\hat{\mathbf{y}} | a(\mathbf{x})), \mathbf{x} \in \mathbb{U}^{\mathcal{S}} \quad (3)$$

where  $\hat{\mathbf{y}}$  denotes the pseudo label which is generated through:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \log p_{\theta}(\mathbf{y} | \mathbf{x}), \quad (4)$$

where  $\operatorname{argmax}$  denotes the greedy decoding, which first takes the maximum probability tokens in each frame and then removes repeated and blank tokens. Note that the pseudo labels are generated using the up-to-date model  $\theta$  on-the-fly as in [15].

After pre-training, the task-specific labeled loss  $\mathcal{L}_{\text{label}}$  is also used to fine-tune the pre-trained model on the target domain labeled dataset  $\mathbb{L}^{\mathcal{T}}$ .

## 4. Experiments

#### 4.1. Corpus

The pre-training (source) dataset is LibriSpeech [28], where the 100h clean subset is used as the labeled dataset  $\mathbb{L}^{\mathcal{S}}$  and the other 860h is the unlabeled dataset  $\mathbb{U}^{\mathcal{S}}$ . As for fine-tuning (target) labeled datasets  $\mathbb{L}^{\mathcal{T}}$ , Wall Street Journal (WSJ) and the US accented part of AESRC [29] is used as the in-domain dataset since they are all read datasets. To verify the generalization ability, conversational dataset SwitchBoard (SWBD) [30] and lecture dataset TED-LIUM3 (TED) [31] are used as cross-domain datasets. Moreover, Mandarin Chinese dataset AISHELL-1 [32] and French dataset from Common Voice (CV French) [33] are used as cross-lingual datasets. We concentrate on the low-resource scenario, thus randomly selecting 1h or 10h subset of the above datasets for fine-tuning.

#### 4.2. Implementation Details

All experiments are conducted with fairseq [34]. For self-supervised pre-training, we use the open-source wav2vec 2.0 or data2vec base model pre-trained on Librispeech 960h. For semi-supervised pre-training, we use gradient accumulation to achieve an effective batch size of 25.6m samples. The maximum learning rate is  $3 \times 10^{-5}$ , and the tri-state learning rate schedule from [1] is used. And the convolutional feature encoder is fixed during training. For fine-tuning, batch size and learning rate are the same with semi-supervised pre-training.

Table 1: 1h and 10h fine-tuning with different pre-training approaches.

Method	Pre-training Data		WER (%)														AVG
			In-domain				Cross-domain				Cross-lingual						
	Librispeech		WSJ		AESRC		SWBD		TED		AISHELL-1		CV French				
	Labeled	Unlabeled	dev93	eval92	dev	test	RT03	H-SB	H-CH	dev	test	dev	test	dev	test		
<b>1h fine-tune</b>																	
Supervised Pre-train	960h	×	7.1	4.0	16.8	17.5	29.1	20.0	32.0	13.5	14.4	59.2	60.2	71.3	72.9	32.2	
Wav2vec 2.0	×	960h	8.4	6.4	16.0	16.8	28.1	19.9	28.9	17.1	15.1	67.3	66.8	61.0	63.4	31.9	
Wav2vec-S	100h	860h	<b>5.4</b>	<b>3.8</b>	<b>11.3</b>	<b>10.9</b>	<b>22.6</b>	<b>14.2</b>	<b>22.7</b>	<b>10.0</b>	<b>9.9</b>	<b>48.9</b>	<b>48.7</b>	<b>51.2</b>	<b>53.9</b>	<b>24.1</b>	
<b>10h fine-tune</b>																	
Supervised Pre-train	960h	×	6.2	3.6	13.5	13.6	25.8	15.6	29.7	12.2	12.8	27.0	27.8	46.8	49.9	21.9	
Wav2vec 2.0	×	960h	5.1	3.5	9.7	10.7	19.6	11.8	19.6	10.8	10.2	14.8	14.6	32.3	35.3	15.2	
Wav2vec-S	100h	860h	<b>4.4</b>	<b>2.9</b>	<b>8.7</b>	<b>9.1</b>	<b>18.7</b>	<b>10.8</b>	<b>18.8</b>	<b>9.0</b>	<b>8.8</b>	<b>13.6</b>	<b>14.0</b>	<b>31.2</b>	<b>34.5</b>	<b>14.2</b>	

Apart from the convolutional feature encoder, the transformer context network is also fixed for the first 10k updates. The total training updates for 10h and 1h fine-tuning are 20k and 13k, respectively. Beam-search decoding with a dataset-specific 4-gram language model is used for evaluation.

### 4.3. Main Results

As shown in Table 1, we perform 1h and 10h fine-tuning on different pre-trained models. All pre-trained models are trained on the Librispeech but with different amounts of labeled/unlabeled data. The wav2vec 2.0 model trained on unlabeled data and the supervised pre-trained model trained on labeled data from scratch are used for comparison.

Comparing the supervised pre-trained model and wav2vec 2.0 model, we find that for 1h fine-tuning, the supervised pre-trained model outperforms wav2vec 2.0 on 3 out of 6 datasets (WSJ, TED and AISHELL-1). However, when fine-tuning data increases to 10h, wav2vec 2.0 consistently outperforms the 960h supervised pre-trained model on all datasets. It illustrates the effectiveness and generalization of self-supervised pre-training. The proposed wav2vec-S model consistently outperforms the supervised and wav2vec 2.0 model on all datasets, demonstrating the effectiveness of the simple pipeline of wav2vec-S. Note that in Table 1, only 100h labeled data is used in wav2vec-S, although more labeled data could provide better results (shown in subsection 4.4). Moreover, pre-training and fine-tuning both used character-level supervision and training updates is 20k. We will further discuss the impact of supervision level and training updates in subsection 4.5 and subsection 4.6. The following experiments are conducted with 10h fine-tuning on three representative datasets (WSJ, SWBD, AISHELL-1).

### 4.4. Semi-supervised Pre-training data

Table 2: Wav2vec-S performance with different semi-supervised pre-training data.

Pre-training Data		WER (%)								
		WSJ		SWBD			AISHELL-1		AVG	
Labeled	Unlabeled	dev93	eval92	RT03	H-SB	H-CH	dev	test		
100h	0h	4.6	2.7	19.1	11.2	18.8	14.1	14.2	12.1	
960h	0h	4.3	2.6	19.0	10.8	18.6	13.5	13.8	11.8	
100h	860h	4.4	2.9	18.7	10.8	18.8	13.6	14.0	11.9	

We compare using different amounts of labeled and unlabeled data during semi-supervised pre-training. As shown in

Table 2, the performance is the best when using all 960h labeled data and is the worst when using only 100h labeled data. Semi-supervised pre-training with 100h labeled and 860h unlabeled data effectively bridges the performance gap and achieves comparable performance with the 960h labeled one.

### 4.5. Supervision Level

We discuss the optimal supervision level for semi-supervised pre-training. Specifically, We consider phone-level supervision and character-level supervision. The phoneme transcripts are generated using phonemizer<sup>1</sup>.

Table 3: Wav2vec-S performance with different supervision level for semi-supervised pre-training and fine-tuning.

Pre-train	Fine-tune	WER (%)								
		WSJ		SWBD			AISHELL-1		AVG	
		dev93	eval92	RT03	H-SB	H-CH	dev	test		
Phone	Phone	5.9	4.7	20.2	13.1	20.2	15.8	15.3	13.6	
Char	Phone	5.6	4.8	19.9	13.2	20.2	15.9	15.3	13.6	
Phone	Char	4.8	3.3	19.3	11.3	19.1	14.7	15.5	12.6	
Char	Char	4.4	2.9	18.7	10.8	18.8	13.6	14.0	11.9	

As shown in Table 3, when phone-level fine-tuning is used, phone-level and character-level pre-training perform similarly. On the other hand, when character-level fine-tuning is used, character-level pre-training clearly outperforms phone-level. It shows that the higher-level supervision (character) during pre-training can generalize well to the lower level (phone) but not vice versa. This conclusion also stands in the cross-lingual dataset (AISHELL-1), although the character supervision during pre-training and fine-tuning are in different languages. Therefore, we could conclude that character-level supervision is better for semi-supervised pre-training.

### 4.6. Training Updates

We illustrate the relation between the number of training updates and downstream performance in Table 4.

When training updates increase, the WERs on source language datasets decrease, including in-domain (validation, WSJ) and the cross-domain (SWBD) datasets. On the contrary, the cross-lingual (AISHELL-1) WER increases. This indicates the trade-off between performances of the source language and other languages: with more training updates, the wav2vec-S model becomes more language-specific and the cross-lingual generalization ability is thus degraded.

<sup>1</sup><https://github.com/bootphon/phonemizer>

Table 4: *Wav2vec-S* performance with different training updates during semi-supervised pre-training. Valid denotes the validation WER on dev-other subset.

Updates	WER (%)								
	Valid	WSJ		SWBD			AISHELL-1		Avg
		dev93	eval92	RT03	H-SB	H-CH	dev	test	
10k	8.3	4.7	2.8	19.3	11.0	19.2	13.5	13.9	12.1
20k	7.7	4.4	2.9	18.7	10.8	18.8	13.6	14.0	11.9
40k	7.3	4.2	2.4	18.7	10.8	18.5	13.9	14.2	11.8

#### 4.7. Training Time

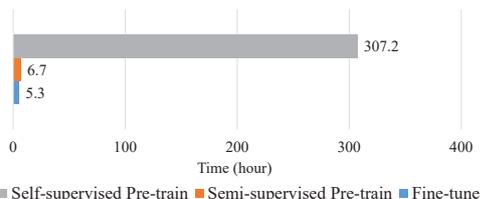


Figure 2: Comparison of training time for self-supervised, semi-supervised pre-training and fine-tuning.

We conduct experiments using 8 V100 GPUs to show the training time for the two steps in wav2vec-S and fine-tuning in Fig. 2. The semi-supervised pre-training requires much less training time than the wav2vec 2.0 self-supervised training. The reason is that self-supervised pre-training can speed up the convergence of the followed semi-supervised pre-training. Moreover, since the self-supervised pre-training is task-agnostic, it can be reused by all downstream tasks. Therefore, for a new task, only semi-supervised pre-training is required to be conducted before fine-tuning.

#### 4.8. On Different Self-supervised Pre-trained Models

Table 5: *Wav2vec-S* performance with wav2vec 2.0 or data2vec as the self-supervised pre-training approach.

Method	WER (%)								
	WSJ		SWBD			AISHELL-1		AVG	
	dev93	eval92	RT03	H-SB	H-CH	dev	test		
<b>1h fine-tune</b>									
wav2vec 2.0	8.4	6.4	28.1	19.9	28.9	67.3	66.8	32.3	
wav2vec 2.0 + semi	5.4	3.8	22.6	14.2	<b>22.7</b>	48.9	48.7	23.8	
data2vec	6.8	5.1	25.6	15.8	26.0	51.2	50.9	25.9	
data2vec + semi	<b>5.3</b>	<b>3.4</b>	<b>22.6</b>	<b>13.3</b>	22.8	<b>45.9</b>	<b>45.5</b>	<b>22.7</b>	
<b>10h fine-tune</b>									
wav2vec 2.0	5.1	3.5	19.6	11.8	19.6	14.8	14.6	12.7	
wav2vec 2.0 + semi	4.4	2.9	18.7	10.8	18.8	<b>13.6</b>	<b>14.0</b>	11.9	
data2vec	4.6	3.0	19.2	10.7	19.2	14.0	14.2	12.1	
data2vec + semi	<b>4.3</b>	<b>2.8</b>	<b>18.7</b>	<b>10.4</b>	<b>18.8</b>	13.8	14.2	<b>11.9</b>	

We use another self-supervised pre-trained model data2vec to test the generalization of wav2vec-S on different self-supervised pre-trained models. As shown in Table 5, data2vec outperforms wav2vec 2.0 on all datasets, illustrating the effectiveness of data2vec. And the wav2vec-S approach still consistently improves the performance of data2vec with the additional semi-supervised pre-training step. Therefore, wav2vec-S could act as a universal refinement approach to enhance a given self-supervised pre-trained model.

#### 4.9. Analysis with Representation Similarity

We analyze the representation similarity before and after fine-tuning for different pre-trained models to show if a particular layer of the pre-trained model is suitable for the ASR task. Specifically, the fine-tuning is performed on the 10h subset. We follow the practice in [8] and compute the CCA similarity between representations from each layer of a pre-trained model and the same layer of the corresponding fine-tuned model, where the lower CCA similarity means the representation changes more significantly during fine-tuning.

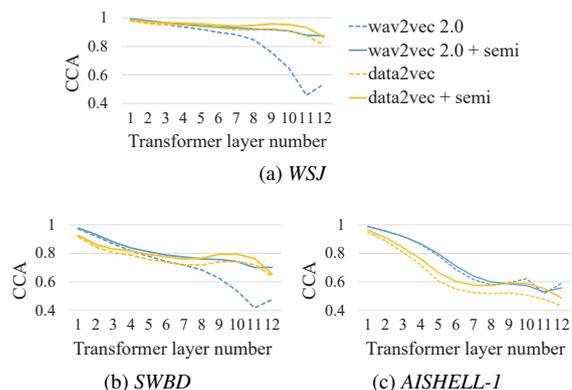


Figure 3: CCA similarity between each layer of a pre-trained model and the same layer of corresponding fine-tuned model.

As shown in Fig. 3, the last few layers of wav2vec 2.0 change significantly during fine-tuning, which illustrates the representations in its last few layers are less effective for ASR. This phenomenon is alleviated in data2vec on both in-domain and cross-domain datasets (WSJ and SWBD), which could be the reason why data2vec outperforms wav2vec 2.0. With the semi-supervised pre-training, both wav2vec 2.0 and data2vec have more similar representation to the fine-tuned model, illustrating semi-supervised pre-training effectively closes the representation gap between the task-agnostic self-supervised pre-trained model and task-specific fine-tuned model.

There are some different phenomena on the cross-lingual dataset AISHELL-1. Firstly, although data2vec performs better than wav2vec 2.0, it has lower CCA similarities in all layers. It means the CCA similarity could not directly reflect the ASR performance on the cross-lingual dataset, which is a significantly out-of-distribution dataset for pre-trained models. Secondly, the semi-supervised pre-training can not consistently improve the similarity in the last few layers of wav2vec 2.0. The reason might be that the labels in semi-supervised pre-training and fine-tuning are in different languages, which calls for the exploration of multi-lingual semi-supervised pre-training [35].

## 5. Conclusions

In this work, we propose wav2vec-S to build a better pre-trained model for low-resource ASR, which improves self-supervised pre-trained models via the task-specific refinement of semi-supervised pre-training. Experiments show that wav2vec-S consistently improves ASR performance on in-domain, cross-domain and cross-lingual datasets over self-supervised pre-trained models like wav2vec 2.0 and data2vec with a marginal increment of pre-training time.

## 6. References

- [1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [2] A. T. Liu, S.-W. Li, and H.-y. Lee, “Tera: Self-supervised learning of transformer encoder representation for speech,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.
- [3] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi *et al.*, “SUPERB: Speech Processing Universal PERFORMANCE Benchmark,” in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.
- [4] K. Deng, S. Cao, and L. Ma, “Improving Accent Identification and Accented Speech Recognition Under a Framework of Self-Supervised Learning,” in *Proc. Interspeech 2021*, 2021, pp. 1504–1508.
- [5] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” *arXiv preprint arXiv:2202.03555*, 2022.
- [6] C. Gao, G. Cheng, Y. Guo, Q. Zhao, and P. Zhang, “Data augmentation based consistency contrastive pre-training for automatic speech recognition,” *arXiv preprint arXiv:2112.12522*, 2021.
- [7] A. Misra, D. Hwang, Z. Huo, S. Garg, N. Siddhartha, A. Narayanan, and K. C. Sim, “A comparison of supervised and unsupervised pre-training of end-to-end models,” in *Proc. Interspeech*, vol. 2021, 2021, pp. 731–735.
- [8] A. Pasad, J.-C. Chou, and K. Livescu, “Layer-wise analysis of a self-supervised speech representation model,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.
- [9] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, “Adaptation algorithms for neural network-based speech recognition: An overview,” *IEEE Open Journal of Signal Processing*, vol. 2, pp. 33–66, 2020.
- [10] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, “Large-scale domain adaptation via teacher-student learning,” *Proc. Interspeech 2017*, pp. 2386–2390, 2017.
- [11] H. Zhu, J. Zhao, Y. Ren, L. Wang, and P. Zhang, “Multi-accent adaptation based on gate mechanism,” *Proc. Interspeech 2019*, 2019.
- [12] Z. Meng, H. Hu, J. Li, C. Liu, Y. Huang, Y. Gong, and C.-H. Lee, “L-vector: Neural label embedding for domain adaptation,” in *ICASSP*. IEEE, 2020, pp. 7389–7393.
- [13] H. Zhu, J. Zhao, Y. Ren, L. Wang, and P. Zhang, “Domain adaptation using class similarity for robust speech recognition,” *Proc. Interspeech 2020*, pp. 4367–4371, 2020.
- [14] W. Hou, H. Zhu, Y. Wang, J. Wang, T. Qin, R. Xu, and T. Shinozaki, “Exploiting adapters for cross-lingual low-resource speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 317–329, 2021.
- [15] Y. Chen, W. Wang, and C. Wang, “Semi-supervised asr by end-to-end self-training,” *Proc. Interspeech 2020*, pp. 2787–2791, 2020.
- [16] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, “slimIPL: Language-Model-Free Iterative Pseudo-Labeling,” in *Proc. Interspeech 2021*, 2021, pp. 741–745.
- [17] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, “Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition,” in *Proc. Interspeech 2021*, 2021, pp. 726–730.
- [18] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, “Improved noisy student training for automatic speech recognition,” *Proc. Interspeech 2020*, pp. 2817–2821, 2020.
- [19] C. Wang, Y. Wu, S. Chen, S. Liu, J. Li, Y. Qian, and Z. Yang, “Improving self-supervised learning for speech recognition with intermediate layer supervision,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7092–7096.
- [20] C. Wang, Y. Wu, Y. Qian, K. Kumatani, S. Liu, F. Wei, M. Zeng, and X. Huang, “Unispeech: Unified speech representation learning with labeled and unlabeled data,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10937–10947.
- [21] J. Bai, B. Li, Y. Zhang, A. Bapna, N. Siddhartha, K. C. Sim, and T. N. Sainath, “Joint unsupervised and supervised training for multilingual asr,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6402–6406.
- [22] Z.-Q. Zhang, Y. Song, M.-H. Wu, X. Fang, and L.-R. Dai, “Xlst: Cross-lingual self-training to learn multilingual representation for low resource speech recognition,” *arXiv preprint arXiv:2103.08207*, 2021.
- [23] Y.-C. Chen, S.-w. Yang, C.-K. Lee, S. See, and H.-y. Lee, “Speech representation learning through self-supervised pretraining and multi-task finetuning,” *arXiv preprint arXiv:2110.09930*, 2021.
- [24] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohmaier, F. Beaufays, and Y. He, “Large-scale asr domain adaptation using self-and semi-supervised learning,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6627–6631.
- [25] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang *et al.*, “Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition,” *arXiv preprint arXiv:2109.13226*, 2021.
- [26] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” *Proc. Interspeech 2019*, pp. 2613–2617, 2019.
- [27] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*. IEEE, 2015, pp. 5206–5210.
- [29] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, and L. Xie, “The accented english speech recognition challenge 2020: open datasets, tracks, baselines, results and methods,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6918–6922.
- [30] J. Godfrey and E. Holliman, “Switchboard-1 release 2 ldc97s62,” *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.
- [31] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, “Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation,” in *International conference on speech and computer*. Springer, 2018, pp. 198–208.
- [32] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *O-COCOSD*. IEEE, 2017, pp. 1–5.
- [33] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.
- [34] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [35] L. Lugosch, T. Likhomanenko, G. Synnaeve, and R. Collobert, “Pseudo-labeling for massively multilingual speech recognition,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7687–7691.