

SOA: REDUCING DOMAIN MISMATCH IN SSL PIPELINE BY SPEECH ONLY ADAPTATION FOR LOW RESOURCE ASR

Natarajan Balaji Shankar¹, Ruchao Fan¹, Abeer Alwan¹

¹University of California, Los Angeles, Department of Electrical and Computer Engineering

ABSTRACT

Recently, speech foundation models have gained popularity due to their superiority in finetuning downstream ASR tasks. However, models finetuned on certain domains, such as LibriSpeech (adult read speech), behave poorly on other domains (child or noisy speech). One solution could be collecting as much labeled and diverse data as possible for joint finetuning on various domains. However, collecting target domain speech-text paired data and retraining the model is often costly and computationally expensive. In this paper, we introduce a simple yet effective method, speech only adaptation (SOA), based on speech foundation models (Wav2vec 2.0), which requires only speech input data from the target domain. Specifically, the Wav2vec 2.0 feature encoder is continually pre-trained with the Wav2vec 2.0 loss on both the source and target domain data for domain adaptation, while the contextual encoder is frozen. Compared to a source domain finetuned model with the feature encoder being frozen during training, we find that replacing the frozen feature encoder with the adapted one provides significant WER improvements to the target domain while preserving the performance of the source domain. The effectiveness of SOA is examined on various low resource or domain mismatched ASR settings, including adult-child and clean-noisy speech.

Index Terms— Automatic Speech Recognition, Self Supervised Learning, Children’s Speech, Unsupervised Domain Adaptation

1. INTRODUCTION

Self-supervised learning (SSL) has gained popularity in speech processing in recent years [1–6]. SSL is capable of leveraging vast amounts of unannotated data to learn domain knowledge, through a process known as pretraining. This model, containing domain knowledge, can then be used as the starting point for downstream tasks, known as finetuning. SSL models can be utilized in two ways: 1) as a substitute for handcrafted speech features by performing feature extraction [7, 8], or 2) as a starting point for model initialization for downstream tasks by performing finetuning [9–11].

One significant drawback of SSL is that training on one domain can lead to domain shifting when finetuning on data from a different domain [12]. Previous studies have attempted to address this issue by incorporating target domain data during pretraining to develop robust pretrained models [13–16]. In [17], it is proposed to continually pretrain the base model on in-domain data before commencing finetuning. Previous work such as [18] attempted to use unannotated target domain data for semi-supervised learning for improving ASR performance on unseen languages. In [13], the effect of adding in-domain data during the pretraining process is analyzed,

and the authors note the significant increase in performance and generalizability. The work in [19] utilizes a combined loss incorporating labels from the source domain, while simultaneously continually pretraining on the source and target domain.

Several studies have also attempted to tackle the domain shift problem through the addition of parameters to the base model. In [20], an adapter based approach is proposed to address the domain shift between adult speech data used for pretraining and child speech data used for finetuning. The study in [21] explores the usage of convolutional adapters to the feature encoder for feature adaptation, while [22] explores the usage of continual pretraining with the presence of adapters for improved ASR performance of accented speech. In [23], the authors explore the effect of adapter tuning in an encoder-decoder framework for a variety of speech classification and sequence generation tasks.

While these methods are effective, they either require the presence of extra parameters [20, 21], or incur a performance penalty on the original source domain due to finetuning [13, 18]. To mitigate this, a proposed solution is to perform joint finetuning utilizing both target domain and source domain data to maintain performance on both sets of data. However, due to the vast size of the training data, this might not always be computationally feasible. The joint finetuning process also necessitates the availability of transcribed speech-text data from the source domain, as well as the new target low resource domain, both of which might not be readily available.

In this work, we propose Speech Only Adaptation (SOA), a simple yet effective strategy for increasing performance on the target domain using unlabeled speech data without performance degradation on the source domain. Our adaptation strategy consists of performing continual pretraining on the Wav2vec 2.0 feature encoder using a mix of unlabeled source and target domain data, while keeping the layers of the contextual encoder frozen. We then replace the frozen feature encoder from a source finetuned model with a feature encoder from the continually pretrained model.

The primary advantage of SOA is that it does not require the presence of paired speech-text data from the target domain, and is thus free of any finetuning. Therefore, it can be readily adapted to various low resource scenarios. With models finetuned on the source domain, SOA enables easy adaptation to a target domain with minimal unlabeled target domain data. Furthermore, SOA maintains performance on the source domain while improving performance on the target domain. To validate these claims, we evaluate this technique on two different low resource domain tasks: child (including a zero-shot case) and noisy speech.

The remainder of this paper is organized as follows. Section 2 introduces the framework for the proposed method. Experimental setups are described in Section 3. Results are shown and discussed in Section 4, and we conclude the paper in Section 5.

This work was supported in part by the NSF.

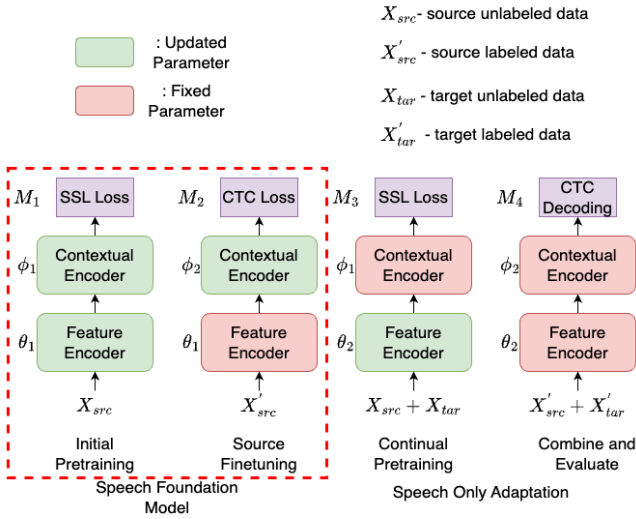


Fig. 1. An overview of steps involved in SOA. The steps involved in obtaining the speech foundation model (initial pretraining, and source finetuning) need to be performed just once, and the models obtained can be used for SOA on different target domains

2. METHODOLOGY

2.1. Background

Let x_{src} be unlabeled data available from the source domain, x'_{src}, y_{src} be paired speech-text (labeled) data from the source domain and x_{tar} be unlabeled data available from the specified target domain.

We model the requirements for the pretraining procedure as possessing a large quantity of speech audio data, x , typically on the order of hundreds of hours of data. For the purpose of finetuning, we require paired speech-text data, $\{x', y\}$ on the order of tens of hours of data. The pretraining process is modeled as learning a function $f: x \rightarrow z$ for learning latent space representations from the raw audio waveforms, and finetuning as learning $f': x \rightarrow y$, by iterating on f to obtain the mapping from the audio to the vocabulary for the given dataset. By pretraining and finetuning on the source domain in this manner, we obtain the functions g and g' respectively.

The issue with a domain shift is that the different distribution of the target domain data from the source domain implies models trained on the source domain do not generalize enough to learn the distribution of the target domain, i.e., g' does not act as a good approximation of h' for mapping between x'_{tar} to y_{tar} . The goal of domain adaptation then is to 'adapt' g' to closer approximate h' .

The simplest way to perform this would be to continually train the model on the available x_{tar} , thus adapting g to h . Even after continual training, the model obtained is a mapping between x_{tar} and z_{tar} , and thus without training on a CTC loss cannot be used for ASR. Thus, this requires a way to combine the latter layers of a model finetuned on a CTC loss, which is provided by SOA, as through the combination of encoder model layers, we approximate the iterating process to obtain h' .

2.2. SSL Pipeline - Pretraining and Finetuning

In this work, we utilize the base Wav2vec 2.0 model [24] as the speech foundation model for its relative computational simplicity. The model consists of a convolutional feature encoder $\theta: x \rightarrow z$

to map raw audio x to latent representations z_1, \dots, z_N . These representations are input to a transformer based contextual encoder $\phi: z \rightarrow c$ to output context representations c_1, \dots, c_N . During pre-training, latent representations are discretized to q_1, \dots, q_N using a vector quantization module. The model is trained to identify the true quantized latent q_t using c_t for each masked time step within a set of distractors sampled from other masked time steps using a contrastive loss [24]. During finetuning, we shift to using a Connectionist Temporal Classification (CTC) loss [25] objective, and task the model with predicting the characters of the paired text data using the representations c_t .

Let the speech foundation model (Wav2vec 2.0) be represented by $M_1: \{\theta_1, \phi_1\}$, where θ_1 represents the parameters present in the feature encoder and ϕ_1 represents the parameters of the contextual encoder. Finetuning of M_1 using paired data $\{x'_{src}, y_{src}\}$, while keeping the weights in the feature encoder frozen, results in model $M_2: \{\theta_1, \phi_2\}$, where M_2 is now a model optimized for performance on the source domain. This model M_2 can be reused for multiple target domains.

2.3. Proposed Framework

SOA is a two-stage (continual pretraining, and combination) training paradigm as shown in Figure 1. SOA can be described as:

Stage 1: Continual pretraining of M_1 using the Wav2vec 2.0 loss with data $x_{src} \cup x_{tar}$ while keeping the parameters of ϕ_1 frozen. This results in model $M_3: \{\theta_2, \phi_1\}$.

Stage 2: Combining the contextual encoder ϕ_2 from M_2 with the feature encoder θ_2 from M_3 , resulting in $M_4: \{\theta_2, \phi_2\}$.

Note that by freezing the parameters of θ_1 during finetuning on the source data, and ϕ_1 during continual pretraining, catastrophic forgetting is effectively prevented.

The advantage in the usage of SOA is twofold: 1) As only the contrastive loss during pretraining is employed for the target domain adaptation, we do not require any paired speech-text data $\{x'_{tar}, y_{tar}\}$, which is difficult to procure for low resource tasks. 2) Models M_1 and M_2 pretrained and finetuned on the source domain can be reused for subsequent adaptation to several target domains, thus reducing the computational cost.

3. EXPERIMENTAL SETUP

3.1. Datasets

Our experiments utilized a diverse array of datasets, each categorized as follows. The foundational base models were initially pretrained using the LibriSpeech corpus [26], consisting of 960 hours of read adult speech. We refer to the model optimized for clean, adult speech as M_2 , achieved through finetuning on the 100-hour clean section of this dataset. In pursuit of enhancing performance for various low resource tasks, we utilized the following datasets for the continuous pretraining of M_3 .

3.1.1. Children's Speech

For child speech experimentation, we employed the MyST Children's Speech Corpus [27], encompassing a total of 499 hours of speech data comprising 244,069 conversational utterances exchanged between children and a virtual tutor. This dataset involved interactions from 1,372 students spanning the third to fifth grades. However, only 42% of the corpus, equivalent to 240 hours, has ASR annotations. The corpus also contains dedicated development and

test sets designed for evaluation purposes. We investigate training on varying subsets of this data to assess the efficacy of our framework.

In addition, we conducted assessments of our method’s performance using the CMU Kids Corpus [28] to illustrate transferability of performance across different children’s speech corpora. The corpus contains 5180 utterances of read speech from 76 speakers, amounting to a total of 9 hours of child speech. We utilize the entirety of the corpus to perform zero-shot inference to demonstrate the efficacy of SOA.

3.1.2. Noise Robustness

To evaluate the performance of our framework on noisy speech, we generate noisy speech files through the addition of noise obtained from the MUSAN dataset [29] to files from the clean section of the LibriSpeech dataset. To simulate noisy speech data, we randomly select noise samples from FreeSound [30] subset of the MUSAN corpus and mix them with the LibriSpeech clean samples, at a randomly selected SNR from [0, 15] dB. We evaluate the performance of this method through the creation of noisy dev and test sets by a similar process of addition of noise to the dev-clean and test-clean sections of the LibriSpeech set as well.

3.2. Model Settings

For the base foundation model M_1 , we use the open-sourced base Wav2vec 2.0 model (95M parameters) from the fairseq toolkit [31]. During different stages of the framework, we obtain different models with varying numbers of updated parameters. Note that M_2 (with updated encoder layers) has close to 92M updated parameters, while M_3 (with only updated convolution layer weights) has close to 5M updated weights, thus leading to an efficient finetuning process for domain adaptation, as M_2 does not need to be retrained for every new domain task.

For the finetuning of M_2 , we update using a noam scheduler [32] with warm up steps of 8k, and a peak learning rate of $3e-5$. The peak learning rate holds for the next 32k steps, then exponentially decays to the ratio λ of the initial learning rate, where λ is set to 0.05.

We perform continual pretraining of M_3 using the Adam optimizer. During the first 8% of all training steps, the learning rate increases to $3e-5$ and then decays polynomially. We experiment with different target domains x_{tar} , as well as varying the size of source x_{src} and target x_{tar} domain corpus used for SOA.

Training for all models is conducted on 2 Nvidia A4000 GPUs. The number of Floating Point Operations needed for training is computed by multiplying the training time of the model, the number of GPUs used during training, and an estimate of the single-precision floating-point capacity of the GPU (19.17 TFLOPS).

4. RESULTS AND DISCUSSION

4.1. Child Speech

4.1.1. Baseline

We first offer a comparison of our method (SOA) with the performance of other similarly sized models in Table 1. For a fair comparison, we list the respective Word Error Rates of HuBERT [6]base models finetuned on LibriSpeech. We use a 4-gram LM trained on the LibriSpeech corpus for decoding. The performance on these corpora is similar to previously published results [20, 33]. Our SOA model listed involved unsupervised domain adaptation of model M_3

till convergence (100k updates) using the 240 hour MyST train corpus with the 100 hour LibriSpeech train corpus.

We also offer a comparison of our method (SOA) with other domain adaptation methods in Table 2, as well as an estimate of the time required for the adaptation. The jointly finetuned model can be viewed as an upper bound on the performance on the specified task. Finetuning on just MyST leads to reduced performance on the source domain compared to both the LibriSpeech finetuned model and SOA. We also continually pretrain a model on the target domain data before finetuning and utilize the M2DS2 framework [19]. Note that SOA outperforms other unsupervised methods while requiring less computational effort, highlighting its effectiveness.

We note that across the Wav2vec 2.0 and HuBERT models that finetuning on either the LibriSpeech or MyST corpus leads to a loss of generalizability and subsequent performance on the other corpus, and the SOA model is able to maintain performance on the LibriSpeech (source) domain, while reducing the WER on the MyST (target) domain. SOA also does not require the presence of any labeled data from the MyST dataset, while the finetuned model utilizes 240 hours of labeled data for its improved WER. We note that the SOA model is able to achieve a 6.44% reduction in relative Word Error Rate, a statistically significant ($p < 0.05$) result compared to the baseline performance.

Model	Decoding	LibriSpeech				MyST	
		dev-clean	dev-other	test-clean	test-other	development	test
HuBERT	w/o LM	5.31	13.15	5.39	12.77	32.82	35.15
	LM	3.19	8.45	3.44	8.32	26.26	28.62
Wav2vec 2.0	w/o LM	5.33	13.85	5.41	13.15	32.72	35.07
	LM	3.12	8.76	3.39	8.65	26.45	28.76
SOA	w/o LM	5.34	14.19	5.48	13.47	30.59	32.81
	LM	3.21	8.94	3.43	8.84	24.91	27.13

Table 1. WER results of HuBERT, Wav2vec 2.0 and SOA on the LibriSpeech and MyST datasets. The HuBERT and Wav2vec 2.0 models were finetuned on the LibriSpeech dataset. A 4-gram LibriSpeech model was used for LM decoding

Model	LibriSpeech		MyST		Training Cost (FLOPs)
	test-clean	test-other	development	test	
LibriSpeech finetuned	5.41	13.15	32.72	35.07	-
Supervised Methods					
MyST finetuned	22.38	35.22	13.45	15.08	-
Jointly finetuned	6.08	14.03	14.08	15.57	-
Unsupervised Methods					
Continual Pretraining	5.45	13.41	32.05	34.2	$6.8 \cdot 10^{18}$
M2DS2 [19]	5.70	13.19	32.43	34.46	$2.6 \cdot 10^{18}$
SOA (Ours)	5.48	13.47	30.59	32.81	$1.5 \cdot 10^{18}$

Table 2. WER results of different domain adaptation methods on the LibriSpeech and MyST datasets. The training cost in Floating Point Operations (FLOPs) is estimated as detailed in Section 3.2

4.1.2. Effect of the size of Low Resource Domain Data

To evaluate the effect of the size of the pretraining corpus used for SOA, we repeat the unsupervised domain adaptation experiments by varying the amount of target domain data x_{tar} available. For all of these experiments, we keep the amount of source domain data x_{src} fixed by utilizing the entire LibriSpeech corpus (100 hours). In Table 3 we present the non-LM decoding WERs of these models. We note that, as expected, an increase in the size of the target domain corpus available for SOA leads to a reduction in the WER of the model, with an effect noted even using just 1 hour of data for SOA.

MyST Training Data	LibriSpeech				MyST	
	dev-clean	dev-other	test-clean	test-other	development	test
Baseline	5.33	13.85	5.41	13.15	32.72	35.07
1h	5.33	14.04	5.42	13.25	32.52	34.72
10h	5.35	14.00	5.48	13.29	31.85	34.09
100h	5.36	14.16	5.49	13.40	30.92	33.14
240h	5.34	14.10	5.48	13.48	30.66	32.90

Table 3. WER results of SOA models trained using a different number of hours of the MyST training data on the LibriSpeech and MyST datasets

4.1.3. Zero-shot performance on auxiliary children’s corpus

Thus far, we have evaluated the performance of SOA with the target domain data x_{tar} obtained from the MyST corpus through testing on the same corpus. To demonstrate the effectiveness of the method in learning features inherent to children’s speech, we test the zero-shot performance of SOA performed using varying amounts of the MyST corpus on the CMU Kids dataset, as shown in Table 4. For these experiments, we do not use any amount of the training corpus from the CMU Kids corpus for either unsupervised domain adaptation or for finetuning. We note that an increase in the amount of MyST data used in the SOA process leads to a decrease in WER on the CMU Kids corpus as well, with training on the entire MyST corpus resulting in an 8.06% reduction in relative Word Error Rate.

MyST Training Data	MyST		CMU Kids
	development	test	
Baseline	32.72	35.07	35.35
1h	32.52	34.72	35.27
10h	31.85	34.09	33.93
100h	30.92	33.14	32.91
240h	30.66	32.90	32.50

Table 4. Zero-shot WER results of SOA models trained using a different number of hours of the MyST training data on the MyST and CMU Kids datasets

4.2. Noise Robustness

We also evaluate the effectiveness of SOA on adaptation for noisy speech in Table 5 through the variation in the number of training hours of the noisy (target) data used for SOA. Here, we refer to Baseline as model M_2 finetuned on only LibriSpeech (source) data without any SOA, and note that the performance for the baseline method is consistent with previously published results [34]. SOA was performed for all the listed models for a total of 50k updates. We note that by utilizing 100 hours of the target domain, we are able to achieve a 28.9% relative Word Error Rate reduction on the noisy test set. We also report the results from decoding using 4-gram LibriSpeech LM for the baseline and best performing SOA model.

Noisy Training Data		LibriSpeech				Noisy	
		dev-clean	dev-other	test-clean	test-other	development	test
Baseline	w/o LM	5.33	13.85	5.41	13.15	16.87	15.29
	LM	3.12	8.76	3.39	8.65	12.54	11.2
1h		5.33	13.98	5.44	13.25	16.03	14.46
10h		5.33	13.92	5.41	13.24	13.83	12.52
100h	w/o LM	5.33	13.91	5.44	13.28	12.11	10.86
	LM	3.15	8.79	3.38	8.75	8.35	7.64

Table 5. WER results of SOA models trained using a different number of hours of the noisy training data on the LibriSpeech and Noisy datasets

4.3. Feature Encoder Output Analysis

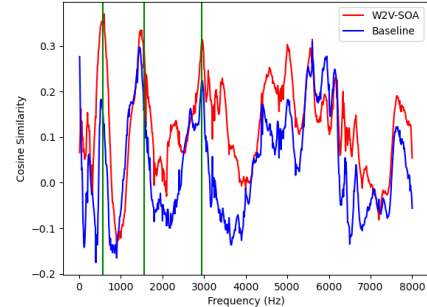


Fig. 2. Cosine Similarity between feature space representations of pure sinusoids and sample audio for baseline model (blue) and SOA model (red). The formant frequencies for the audio file lie at 568 Hz, 1559 Hz and 2944 Hz respectively (green).

To explore how the SOA model shifts the representations of the features vectors z , we perform the following experiment. We first feed both the baseline and SOA models a 1-second signal $x_1 = \sin(2\pi f_i t)$ with f_i ranging from 10Hz to 8kHz at 10Hz intervals, as in [35], to obtain the output representation z_1 . We then feed the models with a speech signal x_2 from the MyST dataset, whose output representation is z_2 . We proceed to compute the cosine similarity between z_1 and z_2 for different values of f_i to demonstrate the ability of the feature encoder to ‘capture’ the formant frequency.

Figure 2 demonstrates a plot of the cosine similarities of the feature space encodings as a function of frequency for the vowel in the word ‘Good’ from the MyST dataset. We note that there is a spike in the values of the cosine similarities at the formant frequencies for the vowel ($F_1 : 568\text{Hz}$, $F_2 : 1559\text{Hz}$, $F_3 : 2944\text{Hz}$). We see that the SOA model shows sharper and more pronounced peaks, indicating that the SOA method leads to the feature encoder being more attuned to the formant frequencies of the target domain. While this plot is only for an individual audio sample, this trend holds up across a variety of utterances, indicating that this shift in the representation space could be the reason behind the improved performance of SOA.

5. CONCLUSION

In this paper, we introduce a novel method, Speech Only Adaptation (SOA), for utilizing unlabeled data from a target domain to perform unsupervised domain adaptation. Specifically, by continually pre-training the feature encoder while keeping the contextual encoder frozen, and replacing the frozen contextual encoder with one obtained during finetuning we demonstrate that it is possible to improve performance on a low resource target domain, while maintaining performance on the source domain. When compared to the conventional finetuning baselines without adaptation, we achieved relative WER improvements of up to 6.4% on the MyST child ASR, and 28.9% on noisy ASR, demonstrating the efficacy of this method. We also illustrate the cross-corpus transferability of performance through an 8.06% relative WER reduction on zero-shot evaluation of the CMU Kids corpus. In scenarios where one can only access unlabeled data (e.g., YouTube recordings), either directly from the target domain or from a closely-related distribution, SOA allows the reuse of finetuned source domain models, making the proposed framework promising for future low resource ASR tasks.

6. REFERENCES

- [1] Yu Zhang, Daniel S Park, et al., “Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1519–1532, 2022.
- [2] Sanyuan Chen, Chengyi Wang, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [3] Abdelrahman Mohamed, Hung-yi Lee, et al., “Self-supervised speech representation learning: A review,” *IEEE Journal of Selected Topics in Signal Processing*, 2022.
- [4] Yu Zhang, James Qin, et al., “Pushing the limits of semi-supervised learning for automatic speech recognition,” *arXiv preprint arXiv:2010.10504*, 2020.
- [5] Arun Babu, Changhan Wang, et al., “Xls-r: Self-supervised cross-lingual speech representation learning at scale,” *arXiv preprint arXiv:2111.09296*, 2021.
- [6] Wei-Ning Hsu, Benjamin Bolte, et al., “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [7] Shu-wen Yang, Po-Han Chi, et al., “SUPERB: speech processing universal performance benchmark,” *Proceedings Interspeech 2021*, pp. 1194–1198, 2021.
- [8] Xuankai Chang, Takashi Maekaku, et al., “An exploration of self-supervised pretrained representations for end-to-end speech recognition,” *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 228–235, 2021.
- [9] Dongwei Jiang, Wubo Li, et al., “A further study of unsupervised pretraining for transformer based speech recognition,” *2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6538–6542, 2021.
- [10] Ananya Misra, Dongseong Hwang, et al., “A comparison of supervised and unsupervised pre-training of end-to-end models,” *Proceedings Interspeech 2021*, pp. 731–735, 2021.
- [11] Zih-Ching Chen, Chin-Lun Fu, et al., “Exploring efficient-tuning methods in self-supervised speech models,” *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1120–1127, 2023.
- [12] Ramon Sanabria, Wei-Ning Hsu, et al., “Measuring the impact of individual domain factors in self-supervised pre-training,” *arXiv preprint arXiv:2203.00648*, 2022.
- [13] Wei-Ning Hsu, Anuroop Sriram, et al., “Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training,” *Proceedings Interspeech 2021*, pp. 721–725, 2021.
- [14] Dongseong Hwang, Ananya Misra, et al., “Large-scale asr domain adaptation using self-and semi-supervised learning,” *2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6627–6631, 2022.
- [15] Yiming Wang, Jinyu Li, et al., “Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition,” *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7097–7101, 2022.
- [16] Lucas Maison and Yannick Esteve, “Improving accented speech recognition with multi-domain training,” *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.
- [17] Suchin Gururangan, Ana Marasovic, et al., “Don’t stop pretraining: Adapt language models to domains and tasks,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, 2020.
- [18] Sameer Khurana, Antoine Laurent, and James Glass, “Magic dust for cross-lingual adaptation of monolingual wav2vec 2.0,” *2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6647–6651, 2022.
- [19] Georgios Paraskevopoulos, Theodoros Kouzelis, et al., “Sample-efficient unsupervised domain adaptation of speech recognition systems: A case study for modern greek,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [20] Ruchao Fan and Abeer Alwan, “DRAFT: A novel framework to reduce domain shifting in self-supervised learning and its application to children’s ASR,” *Proceedings Interspeech 2022*, pp. 4900–4904, 2022.
- [21] Zih-Ching Chen, Yu-Shun Sung, and Hung-yi Lee, “Chapter: Exploiting convolutional neural network adapters for self-supervised speech models,” *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5, 2023.
- [22] Anshu Bhatia, Sanchit Sinha, et al., “Don’t stop self-supervision: Accent adaptation of speech representations via residual adapters,” *Proceedings Interspeech 2023*, 2023.
- [23] Kai-Wei Chang, Ming-Hsin Chen, et al., “Prompting and adapter tuning for self-supervised encoder-decoder speech model,” *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8, 2023.
- [24] Alexei Baevski, Yuhao Zhou, et al., “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.
- [25] Alex Graves, Santiago Fernández, et al., “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, 2006.
- [26] Vassil Panayotov, Guoguo Chen, et al., “Librispeech: An ASR corpus based on public domain audio books,” *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.
- [27] Wayne Ward, Ronald Cole, et al., “My science tutor: A conversational multimedia virtual tutor for elementary school science,” *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 7, no. 4, pp. 1–29, 2011.
- [28] Maxine Eskenazi, Jack Mostow, and David Graff, “The cmu kids corpus,” *Linguistic Data Consortium*, vol. 11, 1997.
- [29] David Snyder, Guoguo Chen, and Daniel Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.
- [30] Frederic Font, Gerard Roma, and Xavier Serra, “Freesound technical demo,” *Proceedings of the 21st ACM international conference on Multimedia*, pp. 411–412, 2013.
- [31] Myle Ott, Sergey Edunov, et al., “fairseq: A fast, extensible toolkit for sequence modeling,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 48–53, 2019.
- [32] Ashish Vaswani, Noam Shazeer, et al., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [33] Ruchao Fan, Yunzheng Zhu, et al., “Towards better domain adaptation for self-supervised models: A case study of child asr,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1242–1252, 2022.
- [34] Yuchen Hu, Chen Chen, et al., “Wav2code: Restore clean speech representations via codebook lookup for noise-robust asr,” *arXiv preprint arXiv:2304.04974*, 2023.
- [35] Kwanghee Choi and Eun Jung Yeo, “Opening the black box of wav2vec feature encoder,” *arXiv preprint arXiv:2210.15386*, 2022.