

DRAFT: A Novel Framework to Reduce Domain Shifting in Self-supervised Learning and Its Application to Children’s ASR

Ruchao Fan¹, Abeer Alwan¹

¹Dept. of Electrical and Computer Engineering, University of California, Los Angeles, USA
fanruchao@g.ucla.edu, alwan@ee.ucla.edu

Abstract

Self-supervised learning (SSL) in the pretraining stage using un-annotated speech data has been successful in low-resource automatic speech recognition (ASR) tasks. However, models trained through SSL are biased to the pretraining data which is usually different from the data used in finetuning tasks, causing a domain shifting problem, and thus resulting in limited knowledge transfer. We propose a novel framework, domain responsible adaptation and finetuning (DRAFT), to reduce domain shifting in pretrained speech models through an additional adaptation stage. In DRAFT, residual adapters (RAs) are inserted in the pretrained model to learn domain-related information with the same SSL loss as the pretraining stage. Only RA parameters are updated during the adaptation stage. DRAFT is agnostic to the type of SSL method used and is evaluated with three widely used approaches: APC, Wav2vec2.0, and HuBERT. On two child ASR tasks (OGI and MyST databases), using SSL models trained with un-annotated adult speech data (LibriSpeech), relative WER improvements of up to 19.7% are observed when compared to the pretrained models without adaptation. Additional experiments examined the potential of cross knowledge transfer between the two datasets and the results are promising, showing a broader usage of the proposed DRAFT framework.

Index Terms: self-supervised learning, domain adaptation, children’s ASR, end-to-end speech recognition

1. Introduction

Recently, self-supervised learning (SSL) for speech has been investigated [1–5] because of its great potential in improving low-resource ASR tasks. In SSL, pseudo-labels are generated for un-annotated data for model pretraining, and then the learned knowledge is transferred to a downstream supervised task through finetuning. For example, autoregressive predictive coding (APC) uses a shifted input sequence as supervision such that the model predicts future frames from previous frames [6–8]. Different from APC, Wav2vec-based methods include sampled negative frames in a contrastive loss to increase discrimination between frames in a way that the learned embedding is closer to the positive frame and more distant to the negative frames [9–12]. A more recent SSL framework, HuBERT [13, 14], creates a pseudo-label for each speech frame using clustering techniques like K-means. Models learned with SSL objectives can be used in two manners: 1) feature extraction as a replacement of hand-crafted speech features [15–17]; or 2) model initialization for finetuning downstream tasks [18–20]. SSL has been shown to be effective in ASR of low-resource languages [21, 22], noisy speech [23], accented speech [24], and child ASR [25, 26]. In [25], we extend APC to learn bidirectional contexts for pretraining from adult speech data.

This paper was supported in part by the NSF and the UCLA-Amazon Science Hub.

However, a main weakness of SSL is that training from one domain causes domain shifting when finetuning on data from a different domain [27, 28]. To address this issue, previous work presented robust pretrained models by adding target domain data during pretraining [29, 30]. However, including target domain data might not be feasible at the pretraining stage. In addition, retraining a large-scale SSL model with both the source and target domain data may not be computationally efficient. In [31], un-annotated target domain data are used for semi-supervised learning during the finetuning stage to alleviate the performance degradation caused by domain shifting. However, no previous work, to our knowledge, has investigated methods for performing adaptation of self-supervised models with finetuning data.

In this paper, we propose a novel framework, domain responsible adaptation and finetuning (DRAFT), to reduce domain shifting in SSL-pretrained speech models. In DRAFT, residual adapters (RAs) are placed between blocks in the transformer and are responsible for learning domain specific information during an additional adaptation stage. The additional adaptation stage trains the model with finetuning data and with the same SSL loss that was used in the pretraining stage. To prevent catastrophic forgetting of the learned knowledge from source domain data, only RA parameters are updated during the adaptation stage. Hence, DRAFT has a lower computational cost than retraining the pretrained models with both the source and target domain data. Note that DRAFT is universal to different SSL methods. When performing DRAFT on SSL-pretrained speech models (trained with adult speech data) for child ASR tasks, we obtain significant improvements over baselines without adaptation for both causal (pretrained with APC) and non-causal transformers (pretrained with Wav2vec2.0 or HuBERT). We presented partial results of DRAFT in a paper that is in review [32]. In this paper, more and different experiments are presented. For example, when the learned RAs from one task are used for finetuning another task (cross transfer), improvements are still observed, showing a broader usage of the proposed framework. We also include results of adapter finetuning experiments for more detailed comparisons.

Note that residual adapters have been used before to learn domain specific parameters for adaptation [33–36]. In [37] residual adapters are inserted to achieve a parameter efficient adaptation for disordered speech. In [38], the same idea is used for SSL-pretrained models for efficient adaptation, which is known as adapter tuning in natural language processing. However, these methods use residual adapters for supervised tasks or during finetuning, unlike our proposed method.

The remainder of this paper is organized as follows. Section 2 describes the proposed DRAFT framework. Experimental setups are described in Section 3. Results are shown and discussed in Section 4. We conclude the paper in Section 5.

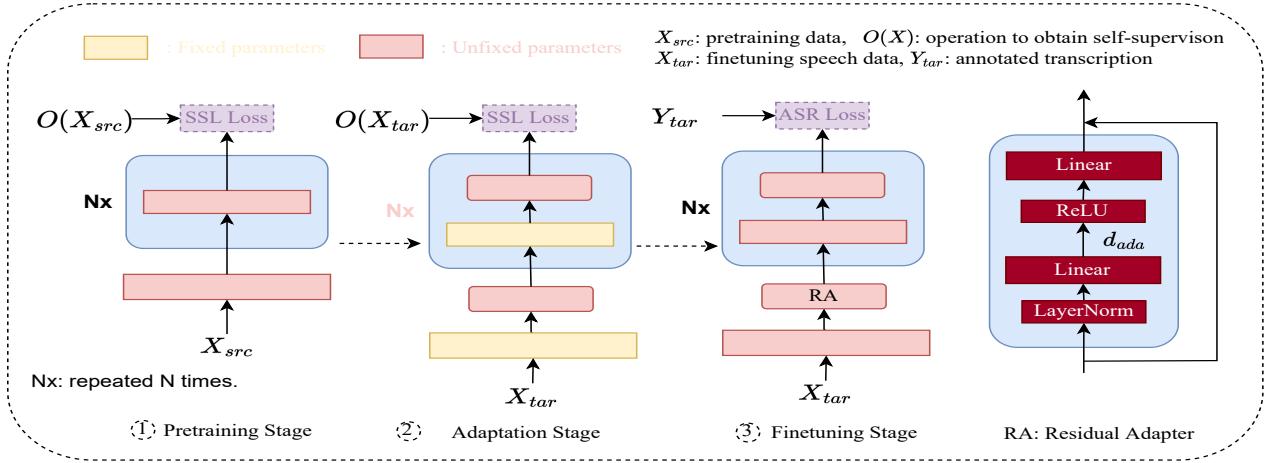


Figure 1: An overview of DRAFT. d_{ada} is the output dimension of the first linear layer in the residual adapter (RA).

2. Method

In this section, we introduce the proposed framework: domain responsible adaptation and finetuning (DRAFT). Fig.1 shows an overview of DRAFT and the structure of the residual adapters.

2.1. Residual Adapters

We start with a description of residual adapters (RAs) (shown in the right side of Fig.1) since they are the most important modules in DRAFT. Specifically, an RA consists of two feed-forward layers with a residual connection. The activation function between the two feed-forward layers makes the adapter non-linear. A layer normalization is inserted at the beginning, which is similar to a self-attention block. We define the dimension of the first feed-forward layer output as d_{ada} , which determines the number of parameters of the RA. The effect of d_{ada} on performance will be analysed in Sec.4.1. Note that RA can be placed anywhere in the model. In our case, we insert one residual adapter after the convolution block and one after each encoder block as shown in Fig.1. We assume that the output of each block needs to be transformed to be similar to that of the target domain data so that the model can easily converge.

2.2. Reducing Domain Shifting in SSL-pretrained Models

Learning from self-supervised pretrained models as a starting point is helpful because of the benefits of learning from large amounts of un-annotated data. However, performance improvement could potentially be limited due to the domain mismatch between the pretraining and finetuning data. As mentioned in the Introduction, including data from the target domain in the pretraining stage can further improve the performance of the target task, but it requires re-training of the self-supervised model, which is time-consuming and computationally expensive. It would be more practical to adapt the pretrained models with only target data. However, direct adaptation on self-supervised models may lead to a catastrophic forgetting problem. We therefore propose DRAFT, a domain responsible adaptation and finetuning framework to prevent catastrophic forgetting that happens when finetuning the entire pretrained model, and to address the domain shifting problem of the conventional self-supervised pretraining and finetuning paradigm.

2.2.1. Simple Adaptation and Finetuning (SAFT)

Before introducing DRAFT, we would like to discuss the most direct way of doing adaptation by re-training the pretrained models, which we refer to as simple adaptation and finetuning (SAFT). In SAFT, an adaptation stage is inserted between the pretraining and finetuning stage, and the model is adapted with an SSL loss and with the finetuning data. All model parameters are updated at the adaptation stage with a smaller learning rate than the pretraining stage to prevent overfitting. The model after the adaptation stage is used as initialization for the finetuning stage with an ASR loss. SAFT updates the parameters of the entire model, and thus might have catastrophic forgetting of learned knowledge from the pretraining stage [39].

2.2.2. Domain Responsible Adapters for Finetuning (DRAFT)

DRAFT is a three-stage (pretraining, adaptation, and finetuning) training paradigm as shown in Fig.1. Different from SAFT, residual adapters are inserted in the model at the adaptation stage to learn knowledge from the finetuning data. Let θ_{ada} be the parameters in residual adapters, θ_f the parameters in the backbone model (without residual adapters), θ_g the parameters in the last embedding mapping layer for the self-supervised task, and θ'_g the parameters in the last linear layer for the ASR task. Suppose source domain data are S_{src} and target domain data are S_{tgt} , DRAFT can be described as:

Stage 1: Initialize a model $\{\theta_f^0, \theta_g^0\}$, update the parameters using data S_{src} and self-supervised loss L_{ssl} , and obtain a pretrained model $\{\theta_f^1, \theta_g^1\}$.

Stage 2: From model $\{\theta_f^1, \theta_g^1\}$, insert residual adapters after each block initialized with θ_{ada}^0 , freeze $\{\theta_f^1, \theta_g^1\}$ and update θ_{ada}^0 using data S_{tgt} and the same self-supervised loss L_{ssl} , and obtain an adapted model $\{\theta_f^1, \theta_{ada}^1, \theta_g^1\}$.

Stage 3: From model $\{\theta_f^1, \theta_{ada}^1, \theta_g^1\}$, replace θ_g^1 with a new generator that can map the embedding space to token space as θ_g^0 , update the entire model with data S_{tgt} and an ASR loss such as the connectionist temporal classification (CTC) loss function, and obtain the final ASR model $\{\theta_f^2, \theta_{ada}^1, \theta_g^0\}$.

Note that the superscript in each θ is the number of times the parameters are updated. For example, θ_f^2 means that backbone model has been updated twice, once in stage one and the other in stage three. Since the backbone model is frozen during stage 2, catastrophic forgetting is effectively prevented.

A similar concurrent work to the proposed approach is [40] where residual adapters are also used to prevent catastrophic forgetting, but for continually learning representations from various languages. Different from [40], finetuning data are used in an additional stage with the purpose of adapting SSL-pretrained models in our method. Furthermore, we update all the parameters during the finetuning stage instead of fixing the backbone model parameters because fixing the parameter does not perform well in our experiments.

3. Experimental Settings

Because of the availability of large databases of adult speech, we explore how SSL methods trained with adult speech can help the development of child ASR systems.

3.1. Databases

Librispeech adult speech corpus [41] is used at the pretraining stage. It contains 960 hours of read speech. During finetuning, we target child ASR tasks on two datasets: OGI Kids’ Speech Corpus (OGI) [42] and My Science Tutor (MyST) [43, 44]. For OGI, the scripted part is used, which contains child speech from approximately 100 speakers per grade (from kindergarten to grade 10). The utterances are randomly split into train (70%), development (15%) and test (15%) sets without speaker overlap [25]. As a result, nearly 50 hours of child data are used to train the child ASR system. MyST contains 499 hours of speech data with 244,069 utterances of conversational speech between children and a virtual tutor from 1,372 students between third and fifth grades. However, only 42% of the corpus (240 hours) is annotated for ASR. We use the annotated part of the corpus to verify the effectiveness of our proposed methods. The corpus also contains a development set and test set for evaluation.

3.2. DRAFT settings

Based on pilot experiments, we applied speed perturbation [45] and SpecAug [46] to both child datasets for better performance. During evaluation, greedy search decoding is used during evaluation for all the experiments. We examine the proposed DRAFT framework on three widely-used SSL approaches: APC, Wav2vec2.0, and HuBERT.

3.2.1. APC

We use multiple shifted sequences as supervisions to construct a multi-task training objective for APC to learn richer information from pretraining. 80-dimensional log-filter-bank features are extracted as model input without any concatenation or frame skipping. The model consists of a two-layer convolution block with a sub-sampling of four along the time axis, 12 transformer encoder blocks with a causal mask in self-attention computation, and a linear layer for each shifted sequence. The output has 320 dimensions because of the sub-sampling in the convolution block. Adam optimizer is used with a noam-based scheduler, where the noam factor is 5 and the warmup step is 15k. The model is updated for 130k steps with a batch size of 256.

At the adaptation stage, residual adapters (RAs) are inserted into the pretrained model and only the RA parameters are updated. For the OGI data, RAs are updated in 55k steps with a noam factor of 8, and warmup steps of 10k. For the MyST data, RAs are updated in 74k steps with a noam factor 4 and a warmup step of 15k. The batch size is set to 64 for both datasets. We also perform experiments using SAFT with the above con-

Table 1: *WER results of different values of d_{ada} in residual adapters using APC. SAFT is the sample adaptation and finetuning that updates the entire model at the adaptation stage. DRAFT is the proposed domain responsible adaption and finetuning that updates only residual adapters at the adaptation stage. The total number of updated parameters are also shown in absolute and relative values (compared to the SAFT). All DRAFT performance improvements are statistically significant.*

d_{ada}	OGI		MyST		Updated Params	
	dev	test	dev	test	total	relative
Baseline	0	5.9	7.0	36.7	36.3	-
Finetune	0	5.0	6.1	32.2	31.6	-
SAFT	0	5.0	5.9	33.4	32.9	39.2M
	64	4.9	5.7	31.9	31.0	0.9M
	128	4.7	5.6	31.6	30.9	1.7M
	256	4.6	5.3	31.1	30.4	3.4M
DRAFT	512	4.4	5.2	30.9	30.2	6.8M
	1024	4.4	4.9	30.1	29.4	13.7M
	2048	4.4	4.9	30.0	29.3	27.3M
						70%

figurations but parameters of the entire model are updated.

At the finetuning stage, CTC loss is used. The model is updated in 240k steps with a batch size of 32, a noam factor of 2, and a warmup step of 10k steps for the OGI data. For the MyST data, the model is updated in 340k steps with a batch size of 64, a noam factor of 2, and a warmup step of 15k steps.

3.2.2. Wav2vec2.0 and HuBERT

For Wav2vec2.0 and HuBERT, we directly use the open-sourced pretrained models in the Fairseq toolkit [47]. We choose the base model that has about 95M parameters to evaluate the effectiveness of the proposed DRAFT framework. Note that the number of parameters in the APC pretraining models are about 39M. In addition, the pretrained Wav2vec2.0 and HuBERT models are non-causal transformers.

At the adaptation stage, the residual adapters of Wav2vec2.0/HuBERT are updated in 200k/100k steps with learning rate ramping up from 0 to the peak learning rate in 32k/8k steps, and then decays linearly back to 0, where the peak learning rate is 5e-4. The batch size is set to 16.

At the finetuning stage, the model is updated with a batch size of 64 in 40k steps with a multi-step scheduler where the warmup steps are set to 4k. The peak learning rate of 3e-5/7e-5 holds for the next 16k steps, then exponentially decays to the ratio λ of the initial learning rate, where λ is set to 0.05.

4. Results and Discussion

4.1. Effect of d_{ada} in DRAFT

We conducted experiments with different values of d_{ada} in the RAs to examine the impact of the number of adapter parameters, because this number influences both WERs and adaptation efficiency. The experiments are conducted on the OGI and MyST datasets using the APC method. Specifically, d_{ada} values are selected from 64 to 2048 and the results are shown in Table 1. For reference, we also include the results for the baseline, finetuning from APC and SAFT. Both WER results and the number of parameters that need to be updated during training are also shown in the table. We observe that the WER drops when we increase the number of parameters in the RAs. However, the cost is increased training time at the adaptation stage because more parameters need to be updated. For exam-

Table 2: *WER results of SAFT and DRAFT for APC, Wav2vec2.0 and HuBERT on the OGI and MyST datasets. We do not provide baseline results for Wav2vec2.0 and HuBERT because we failed to obtain a reasonable WER without pretraining. Finetune is a system using pretraining but without adaptation, and Adapter Finetune is a system where only residual adapters are updated during finetuning. SAFT and DRAFT are systems with finetuning and an additional adaptation stage, and SAFT does not use residual adapters but DRAFT does. Bold numbers: the best results achieved and improvements are statistically significant. NC: no convergence.*

	APC				Wav2vec2.0				HuBERT			
	OGI		MyST		OGI		MyST		OGI		MyST	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
Baseline (w/o SSL)	5.9	7.0	36.7	36.3	-	-	-	-	-	-	-	-
Finetune	5.0	6.1	32.2	31.6	2.27	2.70	17.84	17.16	2.07	2.48	17.40	16.71
Adapter Finetune [38]	8.6	10.1	47.4	47.3	NC	NC	NC	NC	NC	NC	NC	NC
Self-Transfer												
SAFT	5.0	5.9	33.4	32.9	2.22	2.67	17.85	17.28	2.02	2.43	17.52	16.89
DRAFT	4.4	4.9	30.1	29.4	2.11	2.51	17.21	16.70	1.85	2.05	16.79	16.53
Cross-Transfer												
SAFT	5.2	6.2	37.8	37.3	2.33	2.85	21.24	20.28	2.11	2.30	17.67	17.20
DRAFT	4.7	5.5	31.4	30.8	2.13	2.63	17.95	17.36	2.03	2.28	17.13	16.65

ple, even if 2% of the parameters are updated, the WER can decrease from 5.9% to 5.7% on the OGI test set. Hence, the choice of d_{ada} in DRAFT can be adjusted according to scenarios. For example, one can use a small value of d_{ada} to achieve a fast adaptation of the self-supervised model when the computational resources are limited. A large value of d_{ada} can be used to achieve a better performance for the finetuning task. All subsequent DRAFT experiments will use 1024 for d_{ada} since it is a good trade-off between performance and efficiency.

4.2. Main Results for DRAFT

We evaluate the DRAFT framework for three widely used SSL methods: APC, Wav2vec2.0 and HuBERT on both the OGI and MyST datasets. Results are shown in Table 2. As a comparison, we also include the results of adapter finetuning [38], which uses residual adapters directly during the finetuning stage. Though it performs well in [38], adapter finetuning does not work well in our case. Specifically, the performance of adapter finetuning is much worse than that of finetuning the whole system. For Wav2vec2.0 and HuBERT, adapter finetuning does not even converge (100% WER). We also compare DRAFT with SAFT (described in Sec.2.2.1). The table shows that SAFT yields a small improvement and sometimes a negative effect on the WERs compared to the finetuning baselines (without adaptation). The reason may be that updating the entire model causes a catastrophic forgetting of the knowledge learned from adult speech, even though we use a smaller learning rate to prevent overfitting for SAFT. However, when the proposed DRAFT framework is used, the WERs of the three SSL methods decrease for both the OGI and MyST datasets compared to the finetuning baselines (without adaptation). Specifically, we achieve relative WER improvements of 19.7%/7.0%, 7.4%/2.7%, and 16.0%/1.1% on the OGI/MyST test sets for APC, Wav2vec2.0, and HuBERT, respectively. The relative WER improvements with the OGI data are larger than those with the MyST data, which might be because SSL methods are more beneficial for a lower-resource task. Overall, HuBERT achieves the best WER for both datasets. Wav2vec2.0 and HuBERT are better than APC because they use non-causal transformers while APC uses a causal transformer. In addition, the number of APC parameters (39M) is smaller than that of Wav2vec2.0 and HuBERT (95M).

4.3. Cross Knowledge Transfer of Residual Adapters

We also examine the cross knowledge transfer ability of the residual adapters (RAs) learned from one dataset to another as they learn domain-related information. Specifically, the RAs trained with OGI data during the adaptation stage are used for training MyST data at the finetuning stage and vice versa. The results are shown in Table 2. Results show that when using the residual adapters learned from MyST data, DRAFT can improve the WER performance on the OGI dataset consistently with the three SSL methods. This shows that DRAFT can be used for efficiently learning knowledge from one domain (child spontaneous speech), and then achieve better finetuning for another related domain (child read speech). However, the use of RAs learned from OGI data does not help the performance for MyST data. It might be because OGI has only 50-hour speech data, which is not large enough for adapting to the MyST data (240 hours). However, DRAFT always performs better than SAFT in cross knowledge transfer settings, which again shows DRAFT’s ability at preventing catastrophic forgetting. We are also interested in determining whether the learned residual adapters from different domains can be fused together to further improve the performance. However, we do not observe improvements from preliminary results.

5. Conclusions

In this paper, we introduce self-supervised learning (SSL) methods for children’s ASR when unannotated adult speech data are used in the pretraining stage. Specifically, a domain responsible adaptation and finetuning (DRAFT) framework was proposed to alleviate the domain shifting problem between pre-training (adult speech) and finetuning (child speech) data. The DRAFT framework performed well on APC (using causal transformers), Wav2vec2.0 and HuBERT methods (using non-causal transformers). When compared to the conventional finetuning baselines without adaptation, we achieved relative WER improvements of up to 19.7% on the two child ASR tasks. The cross knowledge transfer experiments show a broader usage of the proposed DRAFT framework. To our knowledge, this and our previous papers [25, 32] are the first to develop SSL techniques for child ASR. A future direction of research could be to investigate the fusion of residual adapters learned from various domains for greater WER improvements on this and other low-resource ASR tasks.

6. References

[1] S. Chen, C. Wang *et al.*, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *arXiv preprint arXiv:2110.13900*, 2021.

[2] Y. Zhang, D. S. Park *et al.*, “Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition,” *arXiv preprint arXiv:2109.13226*, 2021.

[3] Y.-A. Chung, Y. Zhang *et al.*, “W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” *ASRU*, 2021.

[4] J. Ao, R. Wang *et al.*, “Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing,” in *Proceedings of Annual Meeting of the ACL*, 2022.

[5] A. T. Liu, S.-W. Li, and H.-y. Lee, “Tera: Self-supervised learning of transformer encoder representation for speech,” *IEEE/ACM TASLP*, vol. 29, pp. 2351–2366, 2021.

[6] Y.-A. Chung *et al.*, “An unsupervised autoregressive model for speech representation learning,” in *Proc. Interspeech 2019*.

[7] Y.-A. Chung and J. Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *IEEE ICASSP*, 2020, pp. 3497–3501.

[8] V. Ravi *et al.*, “Exploring the Use of an Unsupervised Autoregressive Model as a Shared Encoder for Text-Dependent Speaker Verification,” in *Proc. Interspeech 2020*, 2020, pp. 766–770.

[9] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *NIPS 2018*, 2018.

[10] S. Schneider *et al.*, “wav2vec: Unsupervised pre-training for speech recognition.” in *INTERSPEECH*, 2019.

[11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *NeurIPS*, vol. 33, 2020.

[12] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *International Conference on Learning Representations*, 2019.

[13] W.-N. Hsu *et al.*, “HuBERT: How much can a bad teacher benefit asr pre-training?” in *IEEE ICASSP*, 2021, pp. 6533–6537.

[14] ———, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM TASLP*, vol. 29, pp. 3451–3460, 2021.

[15] S.-w. Yang, P.-H. Chi *et al.*, “SUPERB: Speech processing universal performance benchmark,” *Proc. Interspeech 2021*, 2021.

[16] S. Evain, H. Nguyen *et al.*, “LeBenchmark: A reproducible framework for assessing self-supervised representation learning from speech,” *Proc. Interspeech 2021*, 2021.

[17] X. Chang, T. Maekaku *et al.*, “An exploration of self-supervised pretrained representations for end-to-end speech recognition,” *ASRU*, 2021.

[18] D. Jiang, W. Li *et al.*, “A further study of unsupervised pre-training for transformer based speech recognition,” in *IEEE ICASSP*, 2021, pp. 6538–6542.

[19] W. Wang, Q. Tang, and K. Livescu, “Unsupervised pre-training of bidirectional speech encoders via masked reconstruction,” in *IEEE ICASSP*, 2020, pp. 6889–6893.

[20] A. Misra, D. Hwang *et al.*, “A comparison of supervised and unsupervised pre-training of end-to-end models,” in *Proc. Interspeech 2021*, 2021.

[21] M. Riviere, A. Joulin, P.-E. Mazaré, and E. Dupoux, “Unsupervised pretraining transfers well across languages,” in *IEEE ICASSP*, 2020, pp. 7414–7418.

[22] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Applying Wav2vec2.0 to speech recognition in various low-resource languages,” *arXiv preprint arXiv:2012.12121*, 2020.

[23] Y. Wang, J. Li *et al.*, “Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition,” *IEEE ICASSP*, 2022.

[24] J. Li, V. Manohar *et al.*, “Accent-robust automatic speech recognition using supervised and unsupervised wav2vec embeddings,” *arXiv preprint arXiv:2110.03520*, 2021.

[25] R. Fan *et al.*, “Bi-apc: Bidirectional autoregressive predictive coding for unsupervised pre-training and its application to children’s asr,” in *IEEE ICASSP*, 2021, pp. 7023–7027.

[26] J. Wang, Y. Zhu *et al.*, “Low resource german asr with untranscribed data spoken by non-native children–interspeech 2021 shared task spapl system,” *Proc. Interspeech 2021*, 2021.

[27] Y. Meng, Y.-H. Chou, A. T. Liu, and H.-y. Lee, “Don’t speak too fast: The impact of data bias on self-supervised speech models,” *IEEE ICASSP*, pp. 3258–3262, 2022.

[28] R. Sanabria, W.-N. Hsu, A. Baevski, and M. Auli, “Measuring the impact of individual domain factors in self-supervised pre-training,” *arXiv preprint arXiv:2203.00648*, 2022.

[29] W.-N. Hsu, A. Sriram *et al.*, “Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training,” in *Proc. Interspeech 2021*, 2021.

[30] D. Hwang, A. Misra *et al.*, “Large-scale ASR domain adaptation using self-and semi-supervised learning,” *IEEE ICASSP*, 2022.

[31] S. Khurana, A. Laurent, and J. Glass, “Magic dust for cross-lingual adaptation of monolingual wav2vec-2.0,” *IEEE ICASSP 2022*, pp. 6647–6651, 2022.

[32] R. Fan *et al.*, “Towards Better Domain Adaptation for Self-supervised Models: A Case Study of Child ASR,” <https://drive.google.com/file/d/1Klj5Vh5ekpFUaFpCoHjJRMQHb3i3BvCK/view?usp=sharing>, in *IEEE Journal of Selected Topics in Signal Processing*, 2022, in review.

[33] W. Hou, H. Zhu *et al.*, “Exploiting adapters for cross-lingual low-resource speech recognition,” *IEEE/ACM TASLP*, 2021.

[34] A. Kannan, A. Datta *et al.*, “Large-scale multilingual speech recognition with a streaming end-to-end model,” *Proc. Interspeech 2019*, 2019.

[35] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, “Learning multiple visual domains with residual adapters,” in *NIPS 2017*. Neural Information Processing Systems Foundation, Inc, 2017, pp. 506–516.

[36] K. Tomanek, V. Zayats *et al.*, “Residual adapters for parameter-efficient asr adaptation to atypical and accented speech,” in *Proceedings of EMNLP*, 2021, pp. 6751–6760.

[37] N. Houlsby, A. Giurgiu *et al.*, “Parameter-efficient transfer learning for nlp,” in *ICML*. PMLR, 2019, pp. 2790–2799.

[38] B. Thomas, S. Kessler, and S. Karout, “Efficient adapter transfer of self-supervised speech models for automatic speech recognition,” *IEEE ICASSP*, 2022.

[39] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, 1999.

[40] S. Kessler, B. Thomas, and S. Karout, “Continual-wav2vec2: an application of continual learning for self-supervised automatic speech recognition,” *IEEE ICASSP*, 2022.

[41] V. Panayotov, G. Chen *et al.*, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015, pp. 5206–5210.

[42] K. Shobaki, J.-P. Hosom, and R. A. Cole, “The ogi kids’ speech corpus and recognizers,” in *ICSLP*, 2000.

[43] W. Ward, R. 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.

[44] W. Ward, R. Cole, and S. Pradhan, “My science tutor and the myst corpus,” 2019.

[45] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in *Proc. Interspeech 2015*, 2015.

[46] D. S. Park, W. Chan *et al.*, “Specgaugment: A simple data augmentation method for automatic speech recognition,” *Interspeech 2019*, pp. 2613–2617, 2019.

[47] 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 *NAACL-HLT (Demonstrations)*, 2019.