INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Adapter-tuning with Effective Token-dependent Representation Shift for Automatic Speech Recognition

Dianwen Ng[1,2], *Chong Zhang*[1], *Ruixi Zhang, Yukun Ma*[1], *Trung Hieu Nguyen*[1], *Chongjia Ni*[1], *Shengkui Zhao*[1], *Qian Chen*[1], *Wen Wang*[1], *Eng Siong Chng*[2], *Bin Ma*[1]

[1]Speech Lab of DAMO Academy, Alibaba Group
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
{dianwen.ng, b.ma}@alibaba-inc.com

Abstract

The use of self-supervised pre-trained speech models has greatly improved speech tasks in low-resource settings. However, fine-tuning the entire model can be computationally expensive and not scalable for multiple tasks (e.g., personalized ASR). While recent approaches have tried to solve this issue by training adapters, they fail to match the performance of full fine-tuning models, possibly due to the challenge of task domain transferability. Our proposed method enhances the performance of vanilla adapter tuning for ASR by using a simple yet effective token-dependent bias. This approach adds a token-specific representation shift (bias) to the intermediate representations of a pre-trained model, which better maps the latent features of the frozen network to the task domain. Our approach yields better recognition results with the adapter tuning strategy and achieves the performance of a full fine-tuning model on clean LibriSpeech while maintaining its lightweight nature.

Index Terms: Automatic Speech Recognition, Self-Supervised Pre-training, Adapter-tuning, Transfer Learning

1. Introduction

In recent years, there has been a surge of interest in using self-supervised pre-training speech models to overcome the challenges posed by low-resource datasets. These models [1, 2, 3] have proven to be highly effective in capturing and expressing the inherent structure of natural speech in the context of machine learning. In practice, the model learns to construct the representations of speech data from large amounts of unlabeled utterances that benefit diverse downstream speech tasks, such as speech recognition [4, 5, 6, 7, 8] and speaker identification [9, 10]. Thus, they have emerged as the method of choice for researchers and practitioners working in low-resource settings to avoid problems like poor generalization and improve speech task performance.

However, the objective function [2, 11] for pre-training the self-supervised model is typically task-agnostic, which results in feature representations that are more general and less task-targeted. This suggests that the model needs to undergo supervised fine-tuning to achieve the best performance for any downstream task. Despite success in overcoming the challenge of collecting costly annotated datasets, fine-tuning the whole model can be computationally expensive and does not scale well to multiple tasks. In the instance of personalized Automatic Speech Recognition (ASR), a complete set of fine-tuned parameters must be learned and stored in the server for every unique user. Each personalized network often carries a massive size of 100M parameters with reference to the currently available state-of-the-art (SOTA) model choices. Then, to tackle these issues, a more parameter-efficient training strategy is proposed to employ the adapter module [12] in the transformer architecture.

Adapters [13] are lightweight trainable neural blocks that can be inserted within the layers of a pre-trained network while keeping the rest of its original network parameters frozen. These would usually amount to a small fraction of the entire network size, where the optimization process relies on them to tune the frozen network towards the downstream task. Adapters have been employed in several Natural Language Processing (NLP) applications [14, 15, 16] to study the parameter-efficient aspect and demonstrated to achieve comparable performance to the fully fine-tuned model. Furthermore, [17] has investigated the effectiveness of adapters and found that they work well for low-resource fine-tuning and cross-domain tasks since they generate representations that deviate less from those of the initial Pre-trained Language Model (PLM). Likewise, recent works in ASR [18, 19, 20] have also successfully trained the recognition system with the lightweight adapters. However, adopting adapter tuning in ASR still fails to match the performance of a full fine-tuning model with degradation in its recognition error rate [19]. This could be due to the low deviation from the pre-training features [17] produced by the model, which affects the domain transferability needed for mapping speech input to text.

In this paper, we aim to enhance the performance of an ASR model trained with adapters as we work towards closing the gap between full model fine-tuning and parameter-efficient adapter tuning. To improve the task domain transferability of vanilla adapter tuning, we propose adding token-specific task-representation shift (i.e., bias) to the intermediate representations of a pre-trained model. The added bias better maps the latent representations of the frozen network to our downstream task with the relevant task-domain shift [21, 22], resulting in improved task-specific domain transferability for speech-to-text recognition. Besides, similar approaches, such as those discussed in [23, 24], have shown that fine-tuning only the bias in the transfer learning setting is effective for NLP and computer vision tasks. Likewise, our work further supports this by empirically demonstrating the effectiveness of the bias parameters, which yield a better ASR system than the vanilla adapter model. Our experimental results show that our proposed adapter tuning with token-dependent representation shift outperforms the full fine-tuning model on dev-clean and test-clean of the LibriSpeech dataset while only using trainable parameters amounting to only 15% of the original model. The proposed token-dependent bias only introduces around 0.095% trainable weights, maintaining the lightweight nature of adapter tuning. We also conducted detailed ablation studies, which prove the importance of the proposed bias terms in achieving contextual shifts in the representations and improving the transferability of adapter tuning. To the best of our knowledge, this is the first work to investigate the efficacy of introducing trainable bias into

10.21437/Interspeech.2023-1221

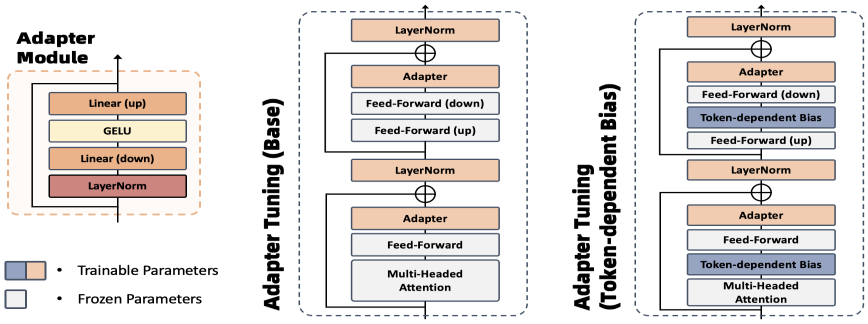

Figure 1: *An illustration of the model architectures for both vanilla and the proposed token-dependent bias adapter tuning. The orange and blue colors indicate the trainable parameters, while the gray blocks represent the frozen parameters during training.* ***Left****: the structure of the adapter module, which includes layer normalization, feed-forward down-projection, a non-linearity, and an up-projection, with a residual connection.* ***Center****: the vanilla adapter tuning [19] architecture, where adapter modules are inserted after the self-attention layer and the output of the feed-forward layer.* ***Right****: our proposed* ***Token-dependent Bias Adapter-tuning (TBA)*** *architecture, which includes token-dependent bias added after the multi-head attention and in the midst of the feed-forward layer.*

adapter tuning for ASR.

2. Methodology

2.1. Adapter-tuning

The structure of an adapter module usually consists of two layers which are a feed-forward down-projection layer and an up-projection layer [13]. The feed-forward layer takes the output of the pre-trained speech model as input and performs a non-linear transformation (i.e., using a GELU activation [25]) to obtain the intermediate representations. The projection layer then maps them to the task-specific output dimension. In addition, we followed [26] to add a layer normalization before the down-projection for smoother gradient and implement a residual connection between the input and the output of the module. A graphical illustration is provided in Figure 1. Generally, adapters can be added in different positions between any sub-layers of the transformer block. However, [19] reported that deploying the adapters after the self-attention block and dense layers results in a better performance in ASR. Hence, we use the same architectural design as the baseline for our work. Moreover, we report that the dense projection layers need not be initialized as a near identity function [13] as we show in our experiments that we have successfully trained our adapter modules with BERT-base [27] weights initialization (i.e., normal distribution).

2.2. Token-dependent Bias Adapter-tuning (TBA)

Token-dependent bias adapter-tuning introduces a learnable representation bias to the intermediate representations, i.e., a trained vector embedding weighted conditionally to each token in the input sequence. The central goal of this term is to add token-specific shifts to the output representations of the frozen pre-trained sub-layer so that they adapt better to the downstream task. As in the previous works, [22, 24] have shown in their experiments the significance and the role of using the bias terms to achieve competitive performance to the full model fine-tuning for NLP tasks. Such bias helps to shift the latent representations to the task domain, making it more task-specific. More importantly, it requires very few additional parameters and computational complexity to implement in practice. Likewise, we propose to apply this to our adapter tuning strategy with some modifications towards the ASR task.

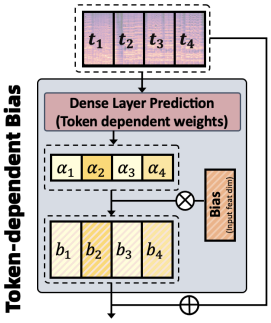

Figure 2: *A graphical representation of the proposed token-dependent bias layer. It consists of a linear dense prediction layer to compute the weights of the bias to be added for each token. Here, bias b is a trainable vector, α is the weight derived from each token t.*

Concretely, we present the token-dependent bias layer architecture in Figure 2. The token-dependent bias layer consists of a trainable bias vector b and a linear layer f_α. The linear layer f_α takes in the latent features from the previous sub-layer to generate token-dependent weights α_i, for each frame token $i \in [1, \ldots, N]$ of a N-length sequence. This weighs the added bias term, b, according to the importance of the representation shift of each frame token to enhance the contextual information of speech representation and helps ASR adapt towards contextual domains. Then, the weighted bias is added to the current latent features and we define the process as

$$x = x + b \otimes f_\alpha(x) \tag{1}$$

where

$$f_\alpha(x) = X.W_\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$$

$x \in \mathbb{R}^{t \times c}$ is the latent features, $W_\alpha \in \mathbb{R}^{c \times 1}$ is the parameters of the linear layer and α is the resultant weighing coefficient. We note that vector bias b has the same channel dimension as x.

In our proposed method, we apply the token-dependent bias twice in each transformer block over the vanilla adapter tuning structure. Specifically, we inserted one after the multi-head attention (dim = 768) and another one after the first feed-forward layer (dim = 3072). We illustrate the proposed TBA architecture in Figure 1.

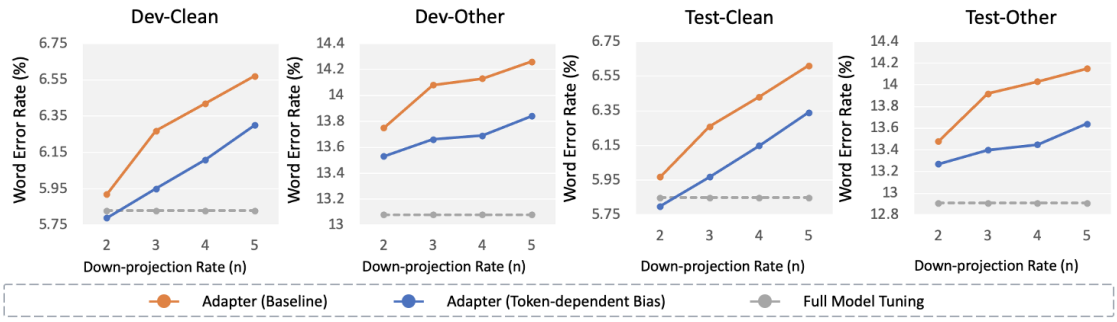

Figure 3: *A comparison of the performance with the vanilla and the proposed token-dependent bias adapter tuning against the full model fine-tuning trained on 100h LibriSpeech. The line plots show the word error rate* (↓) *of each architecture over different down-projection rates* ($n = 2, 3, 4, 5$) *on the LibriSpeech development and testing dataset, without LM.* ***TBA outperforms the baseline adapter on all test sets while outperforming full model fine-tuning on dev-clean and test-clean when the down-projection rate*** $n = 2$.

3. Experiments

We use HuBERT [2] as our self-supervised pre-training model backbone. The model is pre-trained with the full 960 hours LibriSpeech [28] dataset. During pre-training, the model learns from masked predictive coding [11] that predicts the HuBERT codes from the masked features. These codes are generated based on the K-Means ($K = 500$) algorithm from the intermediate latent representations of the sixth layer of HuBERT's transformer at the second iteration. The pre-trained model can be downloaded from the GitHub link[1].

3.1. Dataset

The fine-tuning process in our experiments trains on the official 100 hours LibriSpeech set. In addition, we also train on a subset of ten hours to analyze the performance for a smaller training set. The results are reported on the official *dev-clean*, *dev-other*, *test-clean*, and *test-other* of LibriSpeech.

3.2. Experimental Setup

We compare our method with the vanilla adapter tuning (baseline) in this work. During fine-tuning, we use the pre-trained checkpoint from FairSeq and follow the standard base configuration setup[1] for 100 hours and ten hours of training data. Here, only the layers as shown in Figure 1 are trained, and the rest of the parameters are kept frozen for adapter tuning. Formally, the optimization process can be described as

$$\min_{\theta'} \mathcal{L}_{\text{CTC}}(y, \hat{y}; \theta, \theta') \tag{2}$$

where y, $\hat{y}$ refer to the target text and prediction, respectively. θ' represents the trainable parameters and θ denotes the parameters from the original pre-trained model. The learning rate is grid-searched in the range [2e-5, 1e-4], with 1e-4 being the best on dev-clean set for all adapter tuning methods. Moreover, since changing the down-projection rate affects the quality of the information passed, we investigate the performance of different down-projection rates (n) for the feed-forward layer in the adapter. This corresponds to the channel dimension size of 384, 256, 192, 153, for $n = 2$, 3, 4, 5. Models are trained to convergence, and we take the best checkpoint based on the validation word error rate (WER) to conduct the final evaluations. Note that our work do not use any RNNs decoder [29] during model fine-tuning and the results are based only on the encoder with CTC optimization.

[1] https://github.com/facebookresearch/fairseq/tree-main/examples/hubert

3.3. Experimental Results

We report the result of all our experiments without using a language model (LM). In Figure 3, we obtain a line plot of the performance between the baseline and our proposed token-dependent bias adapter with increased down-projection rates. The y-axis measures the WER of the official testing set (*dev* and *test*) in LibriSpeech. The gray dotted line denotes the performance of the full model tuning. We observe a trend that clearly indicates model degradation with increasing down-projection rates. This is likely caused by the information loss, with the increased reduction in the dimensionality of the down-projection layer. Nevertheless, we show that adding the bias to the vanilla adapter tuning helps improve the general performance of the speech recognition systems with lower WERs. Besides, our model with a down-projection rate of 2 and 14.37M trainable parameters has achieved better performance than the full model tuning for the clean dataset. However, it appears that both the vanilla and our adapter tuning methods did not perform well on dev-other and test-other sets, which may have been affected by the noise disturbance in the *other* LibriSpeech dataset.

To investigate the impact of a smaller dataset, we present experimental results comparing the performance of an ASR model trained on 100 hours and 10 hours of LibriSpeech, as shown in Table 1. We present the results of our experiments using a down-sampling rate of 3, which achieves comparable recognition performance to the fully fine-tuned model while using fewer parameters. Specifically, the performance difference is less than 3% while using only 67% of the parameters compared to the down-sampling rate of 2. We observe that the model's degradation from the smaller training dataset of ten hours appears to be more significant than that from the 100h dataset, as seen when compared to the full fine-tuned model. The deterioration could be as high as 8.25% in the *test-other* set for the vanilla adapter tuning. Nevertheless, our proposed model with token-dependent bias has been demonstrated to mitigate the deterioration and achieve lower WERs, reducing the performance gap between adapter tuning and full model tuning with the added bias terms. Lastly, we found that applying a similar token-dependent biasing-only technique as AdapterBias [22], which was successful in NLP, did not yield good results in ASR, highlighting the distinctions between NLP and ASR.

Table 1: *Experimental results on the official LibriSpeech development and testing set without LM. The Word Error Rates (WERs) for the vanilla adapter (baseline) and the proposed token-dependent bias adapter are based on the down-projection rate of $n = 3$. The comparison of the proposed method with the popular adapter method and the full fine-tuning model is presented here.*

Method	Param Size (M)	WER (%) of LibriSpeech (↓)			
		Dev-Clean	Dev-Other	Test-Clean	Test-Other
Fine-tuning: 100-hours of LibriSpeech					
HuBERT (Full Fine-tuning)	94.5	5.83	13.08	5.85	12.91
AdapterBias [22]	0.09	26.05	33.65	26.27	34.19
Adapter (Baseline)	9.56	6.27	14.08	6.26	13.92
Token-dependent Bias Adapter (Ours)	9.65	**5.95**	**13.66**	**5.97**	**13.40**
Fine-tuning: 10-hours of LibriSpeech					
HuBERT (Full Fine-tuning)	94.5	10.40	18.65	10.64	18.92
AdapterBias [22]	0.09	36.33	42.99	36.70	43.35
Adapter (Baseline)	9.56	11.12	19.96	11.46	20.48
Token-dependent Bias Adapter (Ours)	9.65	**10.79**	**19.59**	**10.89**	**19.82**

3.4. Ablation Study

In this section, we ask two questions to investigate the effectiveness of the proposed bias terms. We report the subsequent experiments of adapters with a down-projection rate of 3 and training on 100 hours of LibriSpeech.

Q1. How would the token-dependent bias-only model perform?

To assess the effectiveness of the bias terms in learning the ASR task, we train our network solely with the bias and layer normalization in every transformer block, while removing the adapter modules. We also compare the results with those obtained by freezing the entire HuBERT and training only on the prediction head, as shown in Table 2. We observed that using only the bias did not yield good performance. However, it is important to note that this setup introduced only 156K parameters compared to 9.65M. Despite the degradation in performance, the bias-only model still achieved three times better results than a frozen HuBERT, suggesting that the bias was able to learn a useful contextual representation shift for better recognition. Nevertheless, we believe that the representation shift alone is not enough to master the ASR task from the frozen network since the task involves mapping from waveform to text, which requires a more complex non-linearity projection for effective learning.

Table 2: *Comparison of the performance (WER%) between including and removing the adapter module with LibriSpeech.*

Method	Dev (↓)		Test (↓)	
	Clean	Other	Clean	Other
Token-dependent Bias Adapter (Proposed)	5.95	13.66	5.97	13.44
Token-dependent Bias (without Adapter Module)	21.30	28.77	21.14	28.90
Frozen HuBERT	62.55	71.14	63.07	70.87

Q2. How would the performance be affected by removing the bias in the proposed trained network?

To evaluate the impact of the bias terms in the proposed network, we ablate the token-dependent bias layers in the model and conduct validation testing on the LibriSpeech test set, as shown in Table 3 that presents the breakdowns of the insertion, deletion, and substitution error of the ASR prediction. We refer to the bias after the multi-headed attention as *Attn bias* and the bias between the feed-forward layers as *FFN bias*. Examining the table, we can observe that removing the bias terms did not significantly affect insertion and deletion errors. However, it had a considerable impact on substitution errors. This might indicate that the bias terms did play a crucial role in providing the contextual shift [22] such that the representations are more content-specific, improving the model's ability to recognize the spoken words correctly. Also, the FFN bias is more impactful compared to the Attn bias. The reason for this could be the larger dimensionality of the bias, which allows for more fine-grained representations. Specifically, the FFN bias has a dimension of 3072, while the Attn bias is only 768.

Table 3: *Comparison of the performance based on the proposed TBA network with removing the token-dependent bias over the specified layer.*

Method	Error Rate on Testing Sets (%) (↓)			
	Insert	Delete	Substitute	WER
LibriSpeech (Test-Clean)				
TBA (Proposed)	0.38	0.45	5.14	5.97
(-) Attn Bias	0.35	0.54	5.22	6.11
(-) FFN Bias	0.45	0.42	6.20	7.07
(-) All Bias	0.43	0.48	6.31	7.22
LibriSpeech (Test-Other)				
TBA (Proposed)	0.84	1.21	11.35	13.40
(-) Attn Bias	0.73	1.39	11.48	13.60
(-) FFN Bias	0.93	1.07	12.89	14.89
(-) All Bias	0.84	1.24	13.04	15.12

4. Conclusion

Our paper introduces a new approach to ASR called token-dependent bias adapter tuning, which can be easily utilized on a self-supervised pre-training backbone. The proposed TBA outperforms the vanilla adapter baseline and even outperforms the full model fine-tuning at the down-projection rate of 2 on LibriSpeech clean sets, while maintaining its lightweight nature. We also conducted ablation studies, which demonstrate the importance of the bias terms in improving content representations by providing the contextual shift. In the future, the proposed TBA can be extended to multi-task scenarios.

5. Acknowledgements

This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore

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, pp. 12 449–12 460, 2020.

[2] W. N. Hsu, B. Bolte, Y. H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "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.

[3] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *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.

[4] D. Ng, R. Zhang, J. Q. Yip, Z. Yang, j. Ni, C. Zhang, Y. Ma, C. Ni, E. S. Chng, and B. Ma, "dehubert: Disentangling noise in a self-supervised model for robust speech recognition," *2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2023.

[5] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He, "Large-scale asr domain adaptation using self-and semi-supervised learning," in *2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2022, pp. 6627–6631.

[6] Y. Zhang, J. Qin, D. S. Park, W. Han, C. C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *NeurIPS SAS 2020 Workshop*, 2020.

[7] J. Ni, Y. Ma, W. Wang, Q. Chen, D. Ng, L. Han, T. H. Nguyen, C. Zhang, B. Ma, and E. Cambria, "Adaptive knowledge distillation between text and speech pre-trained models," *2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2023.

[8] Z. Yang, D. Ng, C. Zhang, X. Fu, R. Jiang, W. Xi, Y. Ma, C. Ni, E. S. Chng, B. Ma, and J. Zhao, "Dual acoustic linguistic self-supervised representation learning for cross-domain speech recognition," in *Proc. Interspeech 2023*, 2023.

[9] N. Vaessen and D. A. Van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," in *2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2022, pp. 7967–7971.

[10] Z. C. Chen, C. L. Fu, C. Y. Liu, S. W. D. Li, and H. y. Lee, "Exploring efficient-tuning methods in self-supervised speech models," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2022, pp. 1120–1127.

[11] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[12] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych, "Adapterhub: A framework for adapting transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 46–54.

[13] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[14] N. S. Moosavi, Q. Delfosse, K. Kersting, and I. Gurevych, "Adaptable adapters," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 3742–3753.

[15] Y. Wang, S. Mukherjee, X. Liu, J. Gao, A. H. Awadallah, and J. Gao, "Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[16] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 487–503.

[17] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J. Low, L. Bing, and L. Si, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 2208–2222.

[18] R. Fan and A. Alwan, "Draft: A novel framework to reduce domain shifting in self-supervised learning and its application to children's asr," in *Proc. Interspeech 2022*, 2022, pp. 4900–4904.

[19] B. Thomas, S. Kessler, and S. Karout, "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7102–7106.

[20] S. Kessler, B. Thomas, and S. Karout, "An adapter based pre-training for efficient and scalable self-supervised speech representation learning," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3179–3183.

[21] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4884–4896.

[22] C. L. Fu, Z. C. Chen, Y. R. Lee, and H. Y. Lee, "Adapterbias: Parameter-efficient token-dependent representation shift for adapters in nlp tasks," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 2608–2621.

[23] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 11 285–11 297.

[24] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.

[25] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[26] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier, "Lightweight adapter tuning for multilingual speech translation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 817–824.

[27] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[29] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," in *Proc. Interspeech 2021*, 2021.