



Parameter-Efficient Fine-tuning with Instance-Aware Prompt and Parallel Adapters for Speaker Verification

Shengyu Peng¹, Wu Guo¹, Jie Zhang¹, Yu Guan¹, Lipeng Dai¹, Zuoliang Li¹

¹NERC-SLIP, University of Science and Technology of China, Hefei, China

shengyupeng@mail.ustc.edu.cn

Abstract

In this paper, we propose a parameter-efficient fine-tuning method to tailor a pre-trained model for speaker verification. The proposed method simultaneously considers adapter tuning and prompt tuning in one framework. Instead of conventional static prompts, we first insert a prompt generator between two neighboring transformer layers of the pre-trained model, which can incorporate utterance-specific clues to dynamically generate instance-aware prompts. Meanwhile, we append parallel adapter branches to the multi-head attention and feed-forward modules in the transformer layers in order to capture speaker-related information. Experimental results on the VoxCeleb datasets demonstrate the superiority of our method in case of updating fewer than 10% of the parameters.

Index Terms: Speaker verification, pre-trained model, adapter, instance-aware prompt

1. Introduction

With the excellent generalization capacity, self-supervised pre-trained models have shown impressive performance on various downstream speech tasks, including automatic speech recognition (ASR), speaker verification (SV) and emotion recognition [1–4]. The key advantage of these models is that they can leverage universal knowledge from large-scale speech datasets, enhancing the robustness of downstream tasks [5]. In SV, the use of pre-trained models (e.g., Wav2Vec2 [6], HuBERT [7], and WavLM [8]) as feature extractor to derive representations from raw audio, instead of traditional handcrafted features [9, 10], followed by backend classifiers (e.g., ECAPA-TDNN [11], ResNet [12] and their variants [13]), has become the mainstream.

Generally speaking, the pre-trained models must be tailored to downstream tasks by full fine-tuning using task-dependent data. However, full fine-tuning pre-trained models becomes increasingly challenging as their scale expands from hundreds of millions to billions of parameters. For instance, WavLM [8] consists of 316 million parameters, and Whisper [14] contains up to 1.55 billion parameters, making their fine-tuning computationally prohibitive [15]. Furthermore, the limitation of task-dependent data can cause over-fitting and catastrophic forgetting in the fine-tuning process [16].

For this, it is straightforward to use linear probing, where the pre-trained model is kept fixed and only the classification head is fine-tuned for downstream tasks. However, this method always results in significant performance degradation compared to full fine-tuning [17]. Parameter-efficient fine-tuning is a feasible method by utilizing lightweight trainable parameters while keeping most pre-trained parameters frozen [18]. In the fields of natural language processing (NLP) and computer vision (CV),

methods like adapter tuning [19], prefix tuning [20], prompt tuning [21] and LoRA [22] have achieved similar or even better results compared to full fine-tuning over a wide range of downstream tasks. The static prompt tuning was employed in [23,24], where a set of learnable parameters is appended to the input to guide the pre-trained model in generating tailored outputs. However, this approach uses the same prompts obtained from the training set during the inference stage, resulting in poor generalization in unseen speakers and complex acoustic scenarios. A prompt pool [25] was applied to SV to address this issue, but its ability to cover the speaker variability during the inference stage still remains limited since the prompt pool comes from the training set and the lack of consideration for interrelations across multiple levels. Adapter tuning is also widely used in SV. Since various acoustic and linguistic clues tend to be encoded in different layers of pre-trained models [8, 17], inner-layer and inter-layer adapters were designed in [26] to capture multi-level information of different speakers. Following the same principle, Intra-block and Cross-block adapters are designed in [16] to respectively capture local and global information for better speech modeling.

In this paper, we propose a novel parameter-efficient fine-tuning framework for SV, which can combine the strengths of the adapter tuning and prompt tuning. The advantage of this method is twofold. First, it can suppress the session variability of the input audio by instance-aware prompts instead of static ones. Second, it can fully exploit the multi-level information of the pre-trained model through carefully designed adapter modules. Specifically, to replace the conventional static prompts, we insert a prompt generator between two neighboring transformer layers of the pre-trained model, which can dynamically generate instance-aware prompts using the information from the output of the neighboring transformer layer. For the adapter tuning, we separately append parallel adapter branches to the multi-head self-attention (MHSA) and feed-forward network (FFN) modules within the transformer layers. The adapter modules can further guide the pre-trained model to downstream SV task by combining the output representations and instance-aware prompts. Experimental results on the VoxCeleb datasets demonstrate that our proposed method surpasses other parameter-efficient fine-tuning approaches and achieves performance comparable to full fine-tuning in case of updating fewer than 10% parameters. Meanwhile, our method achieves greater benefits under low-resource training conditions.

The rest of this paper is organized as follows. Section 2 describes the overall structure of the proposed method in detail. Experimental setup and results are presented in Section 3 and Section 4, respectively. Finally, Section 5 concludes this work.

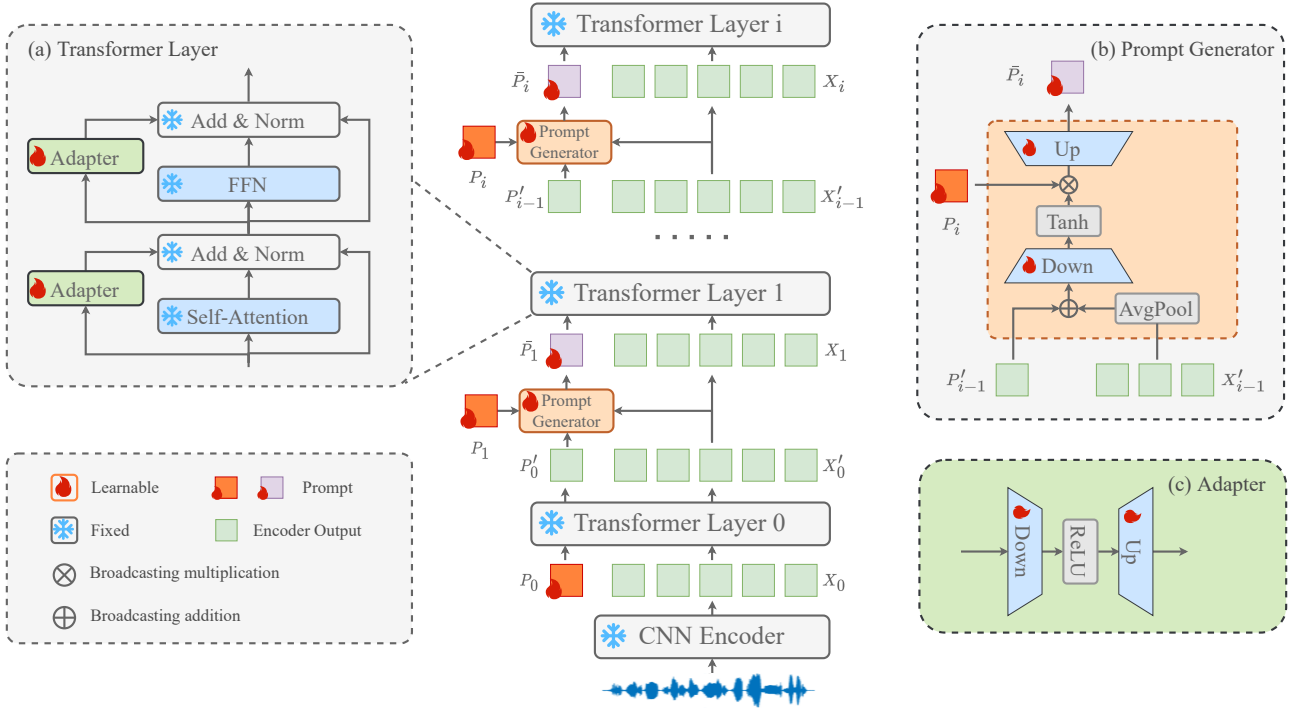


Figure 1: Proposed instance-aware prompt tuning and adapters framework, where the input of each Transformer layer is composed of the speech representations concatenated with the instance-aware prompts, and the instance-aware prompts are dynamically generated by the prompt generator: (a) insertion position of the parallel encoder adapter; (b) prompt generator and (c) parallel encoder adapters.

Table 1: Details of the ECAPA-TDNN, where Conv1d is shown in (kernel size, channel dimension, and scale in Res2net Block) and N_{spk} denotes the number of speakers in classification. The input size is $T \times 512$. This model contains 7.5M parameters.

Block	Configurations	Output size
0	Conv1d(5, 512)	$T \times 512$
1, 2, 3	Res2Block $\left[\begin{array}{l} \text{Conv1d}(1, 512) \\ \text{Conv1d}(3, 512, s=8) \\ \text{Conv1d}(1, 512) \\ \text{SElayer} \end{array} \right]$	$T \times 512$
-	Concat & Conv1d	$T \times 1536$
-	Pooling & Linear	256
-	AAM-Softmax	N_{spk}

2. Methodology

The SV system follows the same structure as described in [17], which contains a front-end feature extractor based on the pre-trained model and a backend ECAPA-TDNN model (see Table 1 for details) as the classifier. This work only focuses on the pre-trained model and the overall structure of the proposed method is depicted in Fig. 1, which primarily consists of a frozen pre-trained model with inserted learnable parameters, such as preset learnable prompt vectors and parallel encoder adapters in the Transformer layers.

2.1. Instance-aware prompts

For prompt tuning, some learnable prompt vectors (denoted as P_0, P_1, P_i in Fig. 1) are always used. In static prompt

tuning works [24], these learnable prompt vectors are always fixed during the inference stage once trained. We dynamically adjust learnable prompt vectors based on input audio to capture the variability across different speakers. The pre-trained model comprises several stacked CNN encoder layers followed by 12/24 Transformer layers. For the output $X_0 \in \mathbb{R}^{d \times T}$ of last CNN encoder layer, we prepend a collection of continuous prompt embeddings denoted as $P_0 = \{p_0^t \in \mathbb{R}^d; 1 \leq t \leq T'\}$, where d is the number of channels, T and T' the number of frames and prompt vectors, respectively. We concatenate P_0 and X_0 along the time dimension to form $[P_0; X_0] \in \mathbb{R}^{d \times (T'+T)}$ as the input of the first Transformer layer (denoted as layer 0 in Fig. 1), given by

$$Z_0 = [P'_0; X'_0] = f_0([P_0; X_0]), \quad (1)$$

where f_0 represents the first Transformer layer. We appoint the first T' frames of Z_0 as the prompt output P'_0 and the remaining frames as the representation output X'_0 .

For other Transformer layers $f_i, i = 1, \dots, N-1$, instance-aware prompts are generated using the outputs P'_{i-1} and X'_{i-1} of the previous Transformer layer. As shown in Fig. 1(b), we first apply adaptive average pooling to X'_{i-1} for aligning P'_{i-1} in time dimension. Then, a down-projection linear layer $W_{down} \in \mathbb{R}^{d \times d'}$ is employed to generate the weighted feature map M as

$$M = W_{down} \left(P'_{i-1} + \text{AvgPool}(X'_{i-1}) \right), \quad (2)$$

After applying broadcasting multiplication \otimes to combine M and the learnable prompt vectors $P_i = \{p_i^t \in \mathbb{R}^d; 1 \leq t \leq$

Table 2: Results on VoxCeleb1-O, VoxCeleb1-E and VoxCeleb1-H datasets, where HuBERT Base and WavLM Base+ have 94.6M and 94.7M parameters, respectively, the backend classifier contains 7.5M parameters, and LM-FT denotes large-margin fine-tuning.

Front-end Model	Systems	Parameters	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
			EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF
HuBERT Base	Fixed	0M + 7.5M	1.616	0.177	1.797	0.201	3.34	0.302
	Full-tuning	94.6M + 7.5M	1.164	0.117	1.243	0.141	2.342	0.219
	Proposed	9.3M+7.5M	1.047	0.105	1.309	0.137	2.177	0.21
	Proposed (LM-FT)	9.3M+7.5M	0.978	0.109	1.203	0.128	2.223	0.214
WavLM Base+	Fixed	0M + 7.5M	1.191	0.151	1.27	0.137	2.511	0.245
	Full-tuning	94.7M + 7.5M	0.814	0.08	0.887	0.099	1.825	0.184
	Proposed	9.3M+7.5M	0.713	0.061	0.906	0.097	1.626	0.161
	Proposed (LM-FT)	9.3M+7.5M	0.654	0.08	0.849	0.094	1.643	0.165

Table 3: Statistics of the training data.

Datasets	Speakers	Utterances
VoxCeleb2	5994	1,092,009
Low-resource	300	52,656

T' }, an up-projection linear layer W_{up} is used to restore the channel dimension as

$$\bar{P}_i = W_{up} (P_i \otimes \tanh(M)), \quad (3)$$

where $\tanh(\cdot)$ stands for the tanh activation function. Similarly, we concatenate the instance-aware prompts \bar{P}_i with the speech representations X_i and feed them into the next Transformer layer, given by

$$\begin{aligned} X_i &= X'_{i-1}, \\ Z_i &= [P'_i; X'_i] = f_i([\bar{P}_i; X_i]), \end{aligned} \quad (4)$$

where Z_i stands for the output.

2.2. Parallel encoder adapters

Besides the prompt tuning, we also integrate the adapter tuning in this work. Different from sequential adapter design [19], we insert two parallel encoder adapters in each Transformer layer for the performance, as shown in Fig. 1 (a). To reduce model parameters and accelerate computation, the parallel encoder adapter is primarily composed of a down-projection linear layer (Down) and an up-projection linear layer (Up), with a ReLU activation function (see Fig. 1 (c)) to enable non-linear transformation. The embedding size for adapters is 128.

For the input feature map x of MHSA/FFN, the corresponding output y can be formulated as

$$y = \text{LN} (f(x) + x' + x), \quad (6)$$

where x' is the output feature map of the adapter, $f(\cdot)$ and $\text{LN}(\cdot)$ represent the MHSA/FFN operation and layer normalization, respectively.

3. Experimental Setup

Datasets: We use the VoxCeleb2 dataset [27] for training and evaluate the systems on the VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H trials [28]. Besides the conventional setup, we also simulate a low-resource training scenario to evaluate the

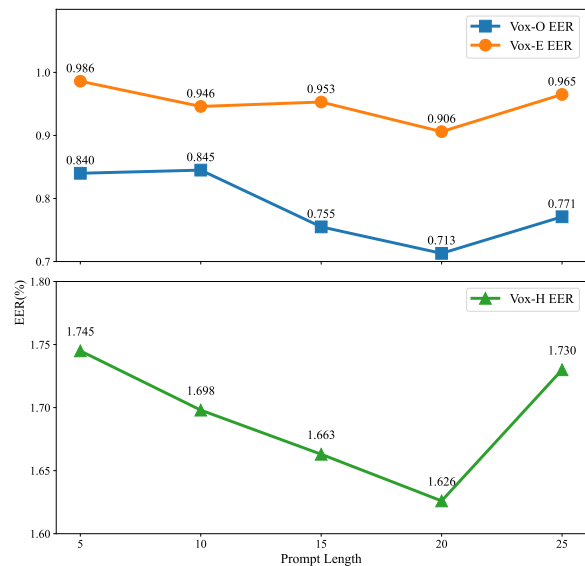


Figure 2: Impact of varying lengths of instance-aware prompts (trained on VoxCeleb2 and evaluated on the VoxCeleb1).

model’s performance. We randomly selected 300 speakers from the VoxCeleb2 dataset as the training set, with the total number of audio files comprising less than 5% of the full dataset. The statistics of the training data are presented in Table 3. Online data augmentation [29] is applied at a probability of 60%, including MUSAN [30], room impulse response (RIR) [31] and speed perturbation [32].

Implementation: We choose HuBERT and WavLM as the front-end feature extractors and ECAPA-TDNN as the back-end classifier. We apply model averaging to the temporal models from the last three epochs. The final speaker embeddings are extracted with a dimension of 256. Meanwhile, we utilize the Adam optimizer with a weight decay of $1e-4$ and a learning rate of $1e-3$. The margin and scaling factors of AAM-Softmax loss [33] are set to 0.2 and 32, respectively. We also use the cosine similarity and s-norm [34] to normalize the scores. After using the 3-second waveforms for 20 epochs, we apply large-margin fine-tuning (LM-FT) [35] to further boost the performance. Specifically, we select utterances over 5 seconds and set the margin to 0.5 for another 2 epochs. In experiments, we

Table 4: Ablation study on (instance-aware) prompts and (parallel encoder) adapters with different pre-trained models.

Front-end	Methods		EER(%)		
	Adapters	Prompts	Vox1-O	Vox1-E	Vox1-H
HuBERT Base	×	×	1.616	1.797	3.34
	×	✓	1.186	1.358	2.37
	✓	×	1.484	1.495	2.213
	✓	✓	1.047	1.309	2.177
WavLM Base+	×	×	1.191	1.27	2.511
	×	✓	0.872	1.065	1.849
	✓	×	0.851	0.939	1.976
	✓	✓	0.713	0.906	1.626

set $T' = 20$ and $d' = 256$. The performance is evaluated in terms of equal error rate (EER) and minimum of the normalized detection cost function (MinDCF) with $P_{target} = 0.01$ and $C_{fa} = C_{miss} = 1$.

4. Experimental Results

4.1. Results on the Voxceleb

The main results on the Voxceleb datasets are listed in Table 2. As expected, the system with the fixed front-end models performs the worst. This implies that the pre-trained models must be fine-tuned using task-specific data for better domain adaptation. The proposed method outperforms the full-tuning system on Vox1-O and Vox1-H in case of updating fewer than 10% of the parameters. After applying the LM-FT training strategy, the performance of our system becomes more stable, despite a small decrease in some metrics. This demonstrates that our method, which integrates the universal representations from the pre-trained model with task-specific information encoded in the learnable parameters, is more advantageous for the SV task.

4.2. Ablation study

In order to evaluate the function of each component within the proposed method, we conduct ablation experiments to show the impacts of instance-aware prompts and parallel encoder adapter modules. As shown in Table 4, the separate uses of prompts and adapters can both improve the performance with different front-end pre-trained models. This is due to the fact that both introduce learnable parameters that enable better adaptation to the downstream task. Their combination achieves the best performance. This is because prompts enable speech adaptation from the input perspective, and the adapters inserted within the Transformer layers further strengthen speaker modeling.

We further investigate the impact of prompt length on the performance in Figure 2. It is clear that increasing the length of the prompts can improve the model’s ability to capture speaker-specific information. However, the performance begins to degrade when the prompt length becomes excessively long. To make an acceptable tradeoff, we set $T' = 20$ in this work.

4.3. Comparison with other adaptation methods

In order to show the superiority of the proposed method over existing adaptation methods, we compare it with the widely used fine-tuning approaches on the Voxceleb. As shown in Table 5, the proposed method significantly outperforms other adaptation methods, including the full-tuning system. In Table 5, the sec-

Table 5: Comparison with other popular adaptation methods.

Methods	EER(%)		
	Vox1-O	Vox1-E	Vox1-H
Full-tuning	0.814	0.887	1.825
Encoder-Adapter [19]	0.861	0.97	2.001
Static-Prompt [24]	0.915	1.133	1.956
LoRA [22]	0.909	0.932	1.955
MAM-Adapter [36]	0.72	0.92	2.05
Dynamic-prompts [25]	1.51	-	-
Proposed	0.713	0.906	1.626
Proposed (LM-FT)	0.654	0.849	1.643

Table 6: Performance evaluation in the low-resource condition.

Methods	EER(%)		
	Vox1-O	Vox1-E	Vox1-H
Fixed	4.259	4.744	9.722
Full-tuning	3.912	4.316	8.918
Encoder-Adapter	3.654	3.9	7.718
Static-Prompt	3.899	4.382	8.287
Proposed	3.478	3.834	7.393

ond best system is MAM-Adapter [36], which utilizes a more advanced pooling method called multi-head factorized attentive pooling [15]. This further validates that the proposed framework can generate more flexible prompts, resulting in better discriminative features for SV.

4.4. Evaluation in the low-resource scenario

We construct a low-resource training set consisting of a small amount of audio from 300 speakers (see Table 3). As shown in Table 6, the full-tuning system exhibits worse performance compared to other parameter-efficient fine-tuning methods. This might be caused by severe overfitting in large pre-trained models. Our method effectively captures the distinctions between speakers, resulting in superior performance. It can thus be concluded that it is more useful to fine-tune a small number of parameters instead of the whole pre-trained model in SV given a limited amount of training data.

5. Conclusion

In this paper, we proposed a novel and effective parameter-efficient fine-tuning framework based on pre-trained models for SV, which primarily consists of instance-aware prompts and parallel encoder adapters. Experiments on the VoxCeleb and the low-resource datasets demonstrate the advantages of the proposed method. In the future, we will investigate the applicability of parameter-efficient fine-tuning methods to various downstream speech tasks.

6. Acknowledgements

This work was supported by Anhui Province Major Science and Technology Research Project (Grant No.S2023Z20004).

7. References

- [1] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Disentangling voice and content with self-supervision for speaker recognition," *Advances in Neural Information Processing Systems (NIPS)*, vol. 36, pp. 50221–50236, 2023.
- [2] H. Wu, J. Zhang, Z. Zhang, W. Zhao, B. Gu, and W. Guo, "Robust spoof speech detection based on multi-scale feature aggregation and dynamic convolution," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2024, pp. 10156–10160.
- [3] S. Peng, W. Guo, H. Wu, Z. Li, and J. Zhang, "Fine-tune pre-trained models with multi-level feature fusion for speaker verification," in *Proc. Interspeech*, 2024, pp. 2110–2114.
- [4] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2022, pp. 7517–7521.
- [5] D. Cai and M. Li, "Leveraging asr pretrained conformers for speaker verification through transfer learning and knowledge distillation," *arXiv preprint arXiv:2309.03019*, 2023.
- [6] A. Baeovski, 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 (NIPS)*, vol. 33, pp. 12449–12460, 2020.
- [7] 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 (TASLP)*, vol. 29, pp. 3451–3460, 2021.
- [8] 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 (JSTSP)*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [9] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: constant q cepstral coefficients," in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, 2016, pp. 283–290.
- [10] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [11] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [13] B. Gu, W. Guo, and J. Zhang, "Memory storable network based feature aggregation for speaker representation learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 31, pp. 643–655, 2023.
- [14] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. International Conference on Machine Learning (ICML)*, 2023, pp. 28492–28518.
- [15] J. Peng, O. Plchot, T. Stafylakis, L. Mošner, L. Burget, and J. Černocký, "An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 555–562.
- [16] H. Wu, W. Guo, S. Peng, Z. Li, and J. Zhang, "Adapter learning from pre-trained model for robust spoof speech detection," in *Proc. Interspeech*, 2024, pp. 2095–2099.
- [17] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2022, pp. 6147–6151.
- [18] N. Chen, Y. Wang, and F. Bao, "Parameter-efficient adapter based on pre-trained models for speech translation," in *Proc. Interspeech*, 2024, pp. 357–361.
- [19] 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 *Proc. International Conference on Machine Learning (ICML)*, 2019, pp. 2790–2799.
- [20] X. L. Li and P. Liang, "Prefix-tuning: optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.
- [21] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- [22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
- [23] H. Ma, Z. Peng, M. Shao, J. Li, and J. Liu, "Extending whisper with prompt tuning to target-speaker asr," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2024, pp. 12516–12520.
- [24] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 709–727.
- [25] Z. Li, M.-w. Mak, H.-y. Lee, and H. Meng, "Parameter-efficient fine-tuning of speaker-aware dynamic prompts for speaker verification," in *Proc. Interspeech*, 2024, pp. 2675–2679.
- [26] M. Sang and J. H. Hansen, "Efficient adapter tuning of pre-trained speech models for automatic speaker verification," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2024, pp. 12131–12135.
- [27] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.
- [28] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.
- [29] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 1038–1051, 2020.
- [30] D. Snyder, G. Chen, and D. Povey, "Musan: a music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.
- [31] T. Ko, V. Peddinti, D. Povey *et al.*, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2017, pp. 5220–5224.
- [32] W. Wang, D. Cai, X. Qin, and M. Li, "The dku-dukecece systems for voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.12731*, 2020.
- [33] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: additive angular margin loss for deep face recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.
- [34] Z. N. Karam *et al.*, "Towards reduced false-alarms using cohorts," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 4512–4515.
- [35] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2021, pp. 5814–5818.
- [36] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černocký, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2023, pp. 1–5.