Interspeech 2024
1-5 September 2024, Kos, Greece

Parameter-efficient Fine-tuning of Speaker-Aware Dynamic Prompts for Speaker Verification

Zhe Li[1], *Man-wai Mak*[1], *Hung-yi Lee*[2], *Helen Meng*[3]

[1] Dept. of Electrical and Electronic Engineering, The Hong Kong Polytechnic University
[2] Graduate Institute of Communication Engineering, National Taiwan University
[3] Dept. of Systems Engineering & Engineering Management, The Chinese University of Hong Kong

lizhe.li@connect.polyu.hk, man.wai.mak@polyu.edu.hk, hungyilee@ntu.edu.tw, hmmeng@se.cuhk.edu.hk

Abstract

Prompt tuning can effectively reduce tunable parameters in pre-trained Transformers. However, it is weak at capturing speaker traits because the prompts can easily overfit the adaptation utterances, resulting in poor generalization to unseen speakers. This paper introduces a prompt pool comprising learnable prompts to tackle this issue. Unlike the traditional method that learns a fixed set of prompts for each training utterance, our method uses a dynamic selection strategy to select the best matching prompts in a pool for tuning, resulting in each prompt being tuned by its closely matched speaker. The objective is to make the prompts in the pool form speaker clusters, enhancing speaker prediction in the downstream classifier while maintaining the plasticity of the pre-trained Transformers. Our experiments on language mismatch in speaker verification demonstrate that the dynamic prompt pool provides a memory- and computation-efficient solution to fine-tune pre-trained Transformers.

Index Terms: Speaker verification; parameter-efficient tuning; prompt tuning; pre-trained Transformer; prompt pool

1. Introduction

Applying pre-trained models (PTMs) to speaker verification (SV) is a promising direction. This approach's main advantage is leveraging knowledge from large-scale speech datasets, enhancing the robustness of downstream SV tasks. However, full fine-tuning of PTMs is challenging as their size grows from hundreds of millions to billions of parameters. For instance, Whisper [1] contains 1.55 billion parameters.

Recently, researchers have proposed parameter-efficient transfer learning (PETL) methods to tune PTMs using lightweight trainable parameters while keeping most pre-trained parameters frozen [2, 3, 4, 5, 6]. Fig. 1 shows the trade-off between speaker verification performance and the number of tunable parameters in these methods compared to the prompt tuning approach. Evidently, for the same number of tunable parameters, prompt tuning has distinct advantages. Prompt tuning involves concatenating trainable prompt tokens with Transformer block's inputs to facilitate few-shot learning in speech recognition [7, 8], text-to-speech [9], and other speech processing tasks [5, 10, 11]. In particular, soft prompts can be appended to the Transformer encoders' input to incorporate additional soft constraints and biases, thereby effectively adapting a Transformer model to a new domain without extensive re-training or fine-tuning.

Recent studies have shown that directly updating trainable tokens may lead to unstable optimization and performance degradation [3, 12]. To tackle these challenges, a prompt encoder, such as a multilayer perceptron (MLP), is employed to reparameterize the token embeddings [3, 13]. In speaker verification, static prompts often lead to poor generalization to unseen speakers and reduced improvements even with additional prompts or tunable parameters.

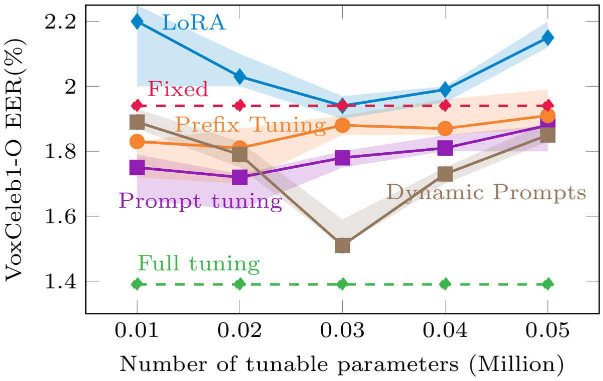

Figure 1: *Various parameter-efficient transfer learning methods reveal a trade-off between an Equal Error Rate (EER) and the number of tunable parameters in a single adaptation architecture. The pre-trained model used in this study is WavLM Large.*

The problem is that most methods associate the prompts with training speakers explicitly, leading to the static prompts overfitting these speakers. Because each utterance has its own prompts, they tend to be associated with the utterance rather than the speaker of the utterance, resulting in poor generalization to unseen speakers. Consequently, increasing the parameters in the prompt encoder does not guarantee the capture of more speaker information, resulting in minimal improvements due to prompt underutilization.

We propose constructing speaker-trait-aware prompts to enhance the generalization to unseen speakers and effectively utilize the prompt embeddings. The speaker-trait-aware prompts have three advantages. First, recent research has shown that allowing the prompts to learn the context from multiple instances can improve generalization to unseen answers in visual question answering [14] and unseen classes in image recognition [15] and reduce catastrophic forgetting in continual learning [16, 17]. Thus, allowing each prompt to learn from multiple instances can enhance prompt generalization. Second, the speaker-trait-aware prompts can capture the complex relationships between speakers, resulting in well-utilized prompt embeddings. Third, by putting the well-utilized prompts into a prompt pool, we can improve performance with fewer parameters, thereby enhancing the parameter efficiency of prompt tuning.

To create a prompt pool, we employ learning a set of

10.21437/Interspeech.2024-295

dynamic prompts that guide a pre-trained Transformer to extract frame-level features that can generalize to unseen speakers. Specifically, prompts in the pool are organized in dynamic key-prompt pairs, where the dynamic keys are the means of the Transformer encoders' inputs and the dynamic prompts are updated by minimizing the cross-entropy speaker loss. A dynamic selection strategy is developed to find the appropriate prompts for each training utterance. The prompt pool ensures that the shared prompts can encode transferable knowledge across speakers and that the individual prompts can capture speaker-specific knowledge. The selected prompts are prepended to the Transformer encoders' inputs, thus implicitly providing speaker-trait instructions to pre-trained models.

In summary, this work makes the following contributions:

- We leverage a speaker prompt pool to adapt PTMs. This new mechanism tackles prompt tuning challenges by introducing a prompt pool memory space, which serves as parameterized instructions for pre-trained models to learn speaker identity.
- Our query mechanism dynamically selects prompts relevant to speaker traits, thereby effectively distinguishing speaker identity. This selection strategy minimizes the interference from knowledge unrelated to speaker identity mixed into speaker representations during optimization.

2. Methodology

Fig. 2 illustrates the proposed model. This section explains the dynamic prompt selection and updating processes and how the prompts can be used for adapting a pre-trained Transformer.

2.1. Dynamic Prompt Pool

Because the speakers during inferencing are usually different from those for training the speaker embedding network, letting the utterance-dependent prompts be optimized for their respective speakers is not flexible. The limitation is that these prompts are fixed after training and will be used as input to the respective Transformer layers during inferencing. However, these utterance-dependent prompts will limit the model's ability to generalize from seen to unseen speakers.

To overcome this limitation, we employ a dynamic prompt pool with each prompt updated by multiple similar speakers. A dynamic selection strategy that finds the closest match between the prompts and the Transformer encoding layers' inputs determines the association between similar speakers and the prompts. This strategy encourages knowledge sharing and avoids catastrophic forgetting.

We denote $\boldsymbol{X}_i \in \mathbb{R}^{D\times T}$ as the output feature maps of the i-th layer of the PTM before concatenating with the prompts. D is the number of output channels and T is the frame count. We denote $\boldsymbol{X}_0 \in \mathbb{R}^{D\times T}$ as the CNN encoder's output. The prompt pool is defined as:

$$\mathcal{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_M\}, \tag{1}$$

where M is the number of prompts in the pool and $\boldsymbol{P}_j \in \mathbb{R}^{D\times T'}$ represents a single prompt of length T' with embedding size D.

2.2. Instance-wise Prompt Searching

As illustrated in Fig. 2, we employ a dynamic key-to-prompt searching strategy to select suitable prompts for various inputs. The layerwise Transformer outputs determine which prompts to choose via key-to-prompt matching. To achieve this, we introduce a key function $q\colon \mathbb{R}^{T\times D} \to \mathbb{R}^D$, encoding input $\boldsymbol{X}_i$ to match the key's dimension, with $\boldsymbol{k}_i = q(\boldsymbol{X}_i) \in \mathbb{R}^D$. Also we define a prompt function $p\colon \mathbb{R}^{T'\times D} \to \mathbb{R}^D$ to map the prompt $\boldsymbol{P}_j$ to a vector of D dimensions, i.e., $\boldsymbol{p}_j = p(\boldsymbol{P}_j) \in \mathbb{R}^D$. Both $p(\cdot)$ and $q(\cdot)$ are implemented by computing the mean along the time axis, meaning that both functions do not have any learnable parameters.

For each key $\boldsymbol{k}_i$, we select a subset of prompts from $\mathcal{P}$ according to the similarity of their encoded vectors $\boldsymbol{p}_j$'s to the key. We define $\{s_t\}_{t=1}^N$ as a set of N indices from $[1, M]$. Given $\{s_t\}_{t=1}^N$, we define $\mathcal{P}_s = \{\boldsymbol{P}_{s_1}, \boldsymbol{P}_{s_2}, \dots, \boldsymbol{P}_{s_N}\}$ as the set of top-N prompts chosen from $\mathcal{P}$. For an input $\boldsymbol{X}_i$, we use $\boldsymbol{k}_i = q(\boldsymbol{X}_i)$ as a key to select the top-N prompts by solving the following objective:

$$\begin{aligned}\{s_t^i\}_{t=1}^N &= \underset{\{s_r\}_{r=1}^N \subset [1,M]}{\operatorname{argmax}} \sum_{u=1}^N \mathit{Sim}(q(\boldsymbol{X}_i), p(\boldsymbol{P}_{s_u})) \\ \mathcal{P}_{s^i} &= \{\boldsymbol{P}_{s_1^i}, \boldsymbol{P}_{s_2^i}, \dots, \boldsymbol{P}_{s_N^i}\}\end{aligned} \tag{2}$$

where $\mathit{Sim} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a similarity function such as cosine.

2.3. Speaker Prompt Tuning

Speaker prompt tuning introduces learnable parameters into the Transformer's input space while freezing the PTM's parameters during downstream training or PTM adaptation.

We introduce a set of prompt embeddings for the i-th layer of a PTM, $\mathcal{P}_{s^i} = \{\boldsymbol{P}_{s_t^i} \in \mathbb{R}^{D\times T'}; 1 \le t \le N\}$, where N is the number of selected prompts. As illustrated in Fig 2, prompts are inserted into each Transformer layer's input space as learnable D-dimensional vectors. With prompting, the Transformer encoder's output at Layer i is:

$$\boldsymbol{Z}_i = \text{Encoder}\left(\left[\boldsymbol{P}_{s_1^i}; \boldsymbol{P}_{s_2^i}; \dots, \boldsymbol{P}_{s_N^i}; \boldsymbol{X}_{i-1}\right]\right), i = 1, 2, \dots, L \tag{3}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{D\times(NT'+T)}$. Then, the first NT' frames of $\boldsymbol{Z}_i$ are dropped, and the remaining T frames are assigned to $\boldsymbol{X}_i$. This process is repeated for Layer $i+1$, with a new prompt subset $\mathcal{P}_{s^{i+1}}$ prepended to $\boldsymbol{X}_i$. In Eq. 3, L is the number of encoder layers, the colors ● and ● indicate learnable and frozen parameters, respectively. and the symbol "; " denotes concatenation along the time dimension.

2.4. Optimizing the Prompts

The frame-level speaker embeddings $\boldsymbol{Z}_i$'s at all encoding layers are linearly combined to produce a frame-level speaker feature matrix $\boldsymbol{H}^*$. The matrix is then passed to the speaker encoder g_ϕ to give an utterance-level speaker embedding vector. For each training utterance, the speaker encoder's parameters (ϕ), the selected prompts $\{\mathcal{P}_{s^i}\}_{i=1}^L$, and the combination weights $\{w_i\}_{i=1}^L$ are updated by backpropagation through minimizing the AAM-Softmax loss [18]. In Eq. 2, each prompt will be updated by the utterances of some similar speakers in a mini-batch.

3. Experiments and Results

3.1. Implementation Details

Pre-trained Model and Speaker Encoder. We chose HuBERT Large [19] and WavLM Large [20] as the PTMs and ECAPA-TDNN [21] as the speaker encoder.

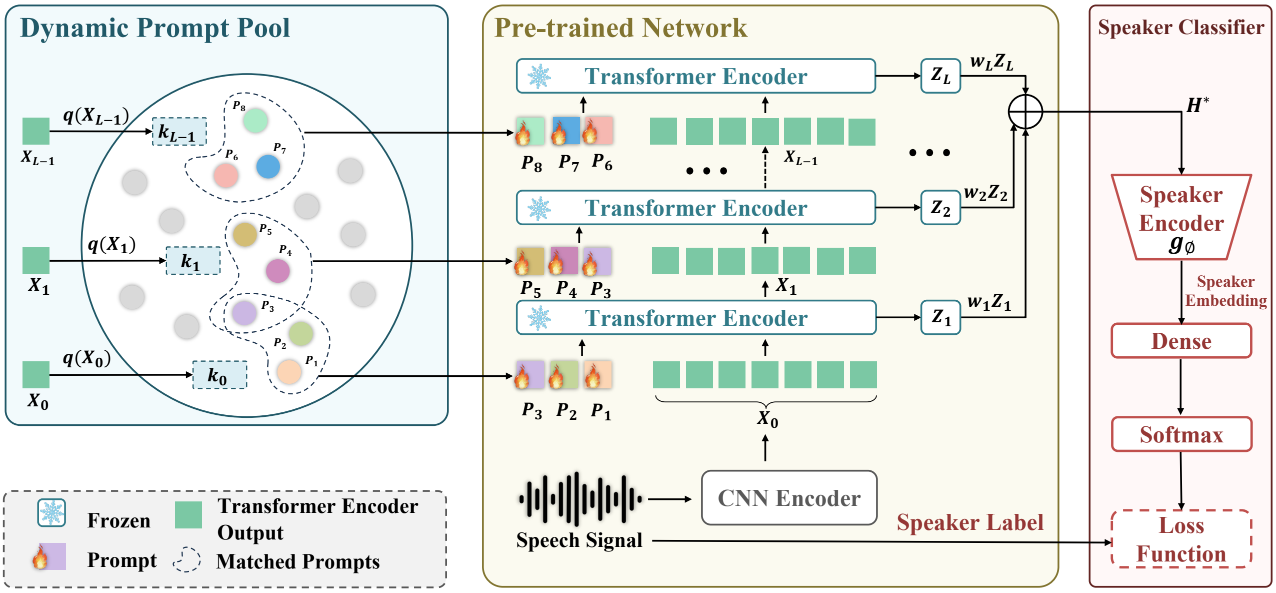

Figure 2: *Illustration of the dynamic prompt selection and updating processes. First, we select a subset of prompts from a key-prompt paired pool using a query mechanism. Then, the selected prompts are prepended to the input vectors of each Transformer encoder. Finally, the extended vectors are fed into the encoders, and the selected prompts in the prompt pool are optimized by minimizing the AAM-Softmax loss. The objective is to select and update the prompts to guide the PTM's predictions.*

Datasets. We used VoxCeleb1-dev [22], CN-Celeb1 [23], and CU-MARVEL [24] to fine-tune the PTMs and train the ECAPA-TDNN. CU-MARVEL, a Cantonese dementia data comprised of 280 speakers, was repurposed for speaker verification experiments. To create a challenging scenario, we trained the models on VoxCeleb1-dev and tested them on the VOiCES Challenge 2019 evaluation set (Voices19c) [25] due to the drastic difference in their acoustic conditions.

Settings. We truncated each training utterance's waveform to 2 seconds and used mini-batches of 128 utterances for fine-tuning and training. We employed AAM-Softmax [18], setting the margin to 0.2 and the scaling factor to 30. The learning rate was reduced by 3% after each epoch. For HuBERT Large and WavLM Large, the settings were $L = 24$ and $D = 1024$. We set T' to 5, N to 3 and M to 15.

3.2. Results and Analysis

Table 1 shows that using a pre-trained model for frame-level feature extraction can improve SV performance, particularly when fine-tuning the PTM is applied. Our prompt pool performs well, utilizing fewer parameters than other parameter-efficient methods. The performance improvement is attributed to our prompt pool, which dynamically learns speaker-aware prompts with significantly fewer tunable parameters.

We observed that full fine-tuning performs badly on the CU-MARVEL dataset, even worse than the performance of the fixed model (without fine-tuning). We speculate that this underperformance may arise from a language mismatch between the pre-trained model and the dataset and the limited number of speakers in CU-MARVEL. This could negatively affect the pre-trained model's parameters during full tuning. In contrast, the larger speaker count of CN-Celeb facilitates the training of a more effective speaker encoder. Thus, this issue is less pronounced in the CN-Celeb dataset. While LoRA is effective for natural language processing, its efficacy in speaker verification is inferior, as shown in Table 1. The performance gap may be due to the focus on capturing the phonetic properties of utterances during the pre-training phase [26]. In contrast, speaker verification demands discrimination between speakers, which is not achievable by merely modifying the attention weights.

3.3. Ablation Study

Table 2 (row 1) shows that removing the prompt pool but using a set of static prompts for each Transformer encoder layer leads to a significant drop in performance. This performance drop indicates severe catastrophic forgetting and knowledge interference among speakers when using static prompts. Conversely, our prompt pool can effectively encode speaker-specific knowledge.

Table 2 (row 2) demonstrates that randomly selecting the prompts from the pool adversely affects performance. This result underscores the critical role played by the key-prompt search to ensure that each prompt is adapted by a group of relevant speakers whose speeches, after transformation by the Transformer encoders, are close to the prompt.

3.4. Effect of Hyperparameters on Dynamic Prompts

Our prompt tuning has three key hyperparameters: prompt pool size M, single prompt T', and the prompt selection size N. Intuitively, M determines the capacity of learnable prompts, T' represents the capacity of a single prompt to encode knowledge, and NT' represents the capacity of the layerwise prompts in adapting the corresponding Transformer layer.

We fixed T' to 5 and M to 15 and then continuously increased N to identify the optimal prompt length. Results in Fig. 3 (upper panel) show that a too-small T' negatively impacts performance, whereas an oversized prompt can lead to knowledge overfitting. We hypothesize that optimal capacity

Table 1: *Results on the test sets of VoxCeleb1, CN-Celeb1, and CU-MARVEL. Using HuBERT Large or WavLM Large as PTM and ECAPA-TDNN as the speaker encoder. In the column "#Parames," the first and second values are the number of adaptation parameters in a single tuning architecture for fine-tuning the PTM and the number of parameters in the ECAPA-TDNN, respectively.*

PTM	Fine-tuning Method	#Params	VoxCeleb1-O		CN-Celeb1		CU-MARVEL	
			EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
-	-	14.7M	2.96	0.30	12.49	0.67	7.20	0.77
HuBERT Large	Fixed	0.0M+14.7M	2.76	0.30	12.05	0.61	10.40	0.93
	Full fine-tuning	316M+14.7M	1.98	0.22	10.51	0.60	11.65	0.98
	Adapter	0.5M+14.7M	2.13	0.24	10.89	0.62	8.10	0.95
	LoRA	0.5M+14.7M	2.38	0.23	10.48	0.60	9.11	0.92
	Static prompt	0.6M+14.7M	2.26	0.23	10.69	0.59	8.31	0.88
	Dynamic prompts (Ours)	0.3M+14.7M	2.17	0.21	10.61	0.58	8.20	0.86
WavLM Large	Fixed	0.0M+14.7M	1.94	0.22	11.17	0.59	6.66	0.88
	Full fine-tuning	316M+14.7M	1.39	0.16	10.47	0.56	9.09	0.94
	Adapter	0.5M+14.7M	1.68	0.19	10.83	0.63	5.58	0.81
	LoRA	0.5M+14.7M	1.88	0.21	10.89	0.63	6.83	0.88
	Static prompt	0.6M+14.7M	1.65	0.18	10.57	0.58	6.42	0.88
	Dynamic prompts (Ours)	0.3M+14.7M	1.51	0.17	10.38	0.59	6.62	0.83

Table 2: *Ablation studies on VoxCeleb1. The train and test data are VoxCeleb1-dev and VoxCeleb1-eval, respectively.*

Ablated component	EER(%)	minDCF
w/o prompt pool	1.65	0.18
w/o key-value pairs	1.71	0.18
Dynamic prompts (Ours)	1.51	0.17

Table 3: *The performance of dynamic prompts and conventional fine-tuning methods on Voices19c. The train data is VoxCeleb1-dev.*

Fine-tuning Method	v19-eval		v19-eval-wpe	
	EER(%)	minDCF	EER(%)	minDCF
Adapter	20.02	0.97	18.62	0.97
Static prompts	20.22	0.96	17.75	0.87
Dynamic prompts (Ours)	19.06	0.93	15.99	0.86

for prompts is essential for encoding specific aspects of shared knowledge.

We also set T' to 5 and N to 3 and progressively increased M. Results in Fig. 3 (lower panel) suggest that enlarging the prompt pool size enhances performance, demonstrating the necessity of a sufficiently large pool to encode diverse speaker-specific knowledge. However, excessively increasing the prompt pool size does not significantly enhance performance.

3.5. Generalization Analysis

We trained the model on VoxCeleb1 and tested them on the evaluation set of Voices19c (v19-eval), acknowledging the acoustic differences between VoxCeleb and VOiCES. Speech files in v19-eval-wpe were subject to weighted prediction error (WPE) processing. Table 3 shows that adapters and static prompts yield similar results, whereas dynamic prompts exhibit improvement. This result suggests that dynamic prompts are better generalized to unseen speakers in different acoustic environments.

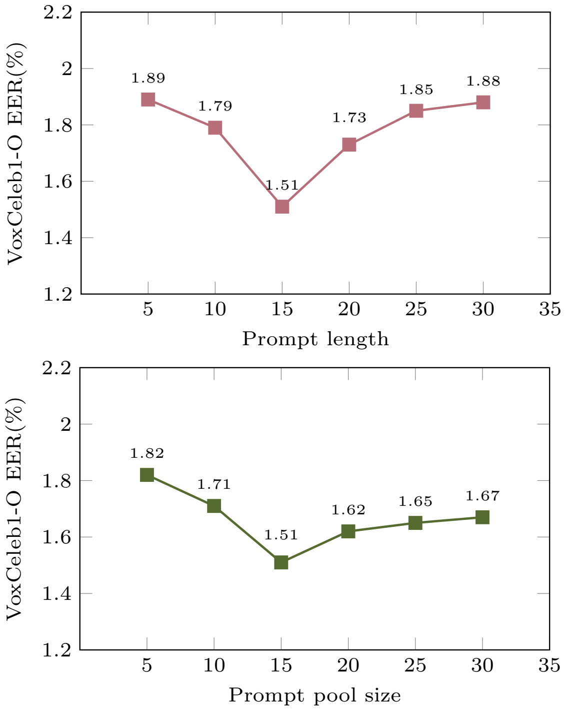

Figure 3: *Results on Voxceleb1-O. The training dataset is VoxCeleb1-dev, and the PTM is WavLM Large. The total length of the prompt is NT'.*

4. Conclusions

This paper introduces a dynamic prompt-tuning method for speaker verification. Specifically, our dynamic prompts approach uses speaker representations as conditions to generate speaker-aware prompts, avoiding implicit correlations with previously seen speakers. Furthermore, we employ a prompt pool to minimize the number of tunable parameters without sacrificing the effectiveness of prompt embeddings. Our experiments in various settings demonstrate that our method surpasses current parameter-efficient baselines in speaker verification.

5. Acknowledgment

This work was supported by the RGC of Hong Kong SAR, Theme-based Research Scheme (T45-407/19-N).

6. References

[1] 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*, 2023, pp. 28 492–28 518.

[2] 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*, 2019, pp. 2790–2799.

[3] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Aug. 2021, pp. 4582–4597.

[4] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2021.

[5] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "An exploration of prompt tuning on generative spoken language model for speech processing tasks," in *Proc. Interspeech*, 2022, pp. 5005–5009.

[6] K.-W. Chang, Y.-K. Wang, H. Shen, I.-t. Kang, W.-C. Tseng, S.-W. Li, and H.-y. Lee, "SpeechPrompt v2: Prompt tuning for speech classification tasks," *arXiv preprint arXiv:2303.00733*, 2023.

[7] C.-H. H. Yang, B. Li, Y. Zhang, N. Chen, R. Prabhavalkar, T. N. Sainath, and T. Strohman, "From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] P. Peng, B. Yan, S. Watanabe, and D. Harwath, "Prompting the hidden talent of web-scale speech models for zero-shot task generalization," in *Proc. InterSpeech*, 2023, pp. 396–400.

[9] Z. Guo, Y. Leng, Y. Wu, S. Zhao, and X. Tan, "PromptTTS: Controllable text-to-speech with text descriptions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[10] H. Gao, J. Ni, K. Qian, Y. Zhang, S. Chang, and M. Hasegawa-Johnson, "WAVPrompt: towards few-shot spoken language understanding with frozen language models," in *Proc. Interspeech*, vol. 2022, 2022, pp. 2738–2742.

[11] Z. Li, M.-W. Mak, and H. M.-L. Meng, "Dual parameter-efficient fine-tuning for speaker representation via speaker prompt tuning and adapters," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[12] H. Yang, J. Lin, A. Yang, P. Wang, C. Zhou, and H. Yang, "Prompt tuning for generative multimodal pretrained models," *CoRR*, vol. abs/2208.02532, 2022.

[13] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 61–68.

[14] B. Yuan, S. You, and B.-K. Bao, "Self-PT: Adaptive self-prompt tuning for low-resource visual question answering," in *Proc. of the 31st ACM International Conference on Multimedia*, 2023, pp. 5089–5098.

[15] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 816–16 825.

[16] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proc. European Conference on Computer Vision*, 2022, pp. 631–648.

[17] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[19] 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.

[20] 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.

[21] 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.

[22] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[23] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-Celeb: a challenging chinese speaker recognition dataset," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7604–7608.

[24] H. Meng, B. Mak, M.-W. Mak, H. Fung, X. Gong, T. Kwok, X. Liu, V. Mok, P. Wong, J. Woo *et al.*, "Integrated and enhanced pipeline system to support spoken language analytics for screening neurocognitive disorders," in *Proc. InterSpeech*, 2023, pp. 1713–1717.

[25] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828*, 2019.

[26] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černockỳ, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.