Interspeech 2024
1-5 September 2024, Kos, Greece

Efficient Fine-tuning of Audio Spectrogram Transformers via Soft Mixture of Adapters

Umberto Cappellazzo[1], *Daniele Falavigna*[2], *Alessio Brutti*[2]

[1]University of Trento, Trento, Italy
[2]Fondazione Bruno Kessler, Trento, Italy

umberto.cappellazzo@unitn.it, {falavi,brutti}@fbk.eu

Abstract

Mixture of Experts (MoE) architectures have recently started burgeoning due to their ability to scale model's capacity while maintaining the computational cost affordable, leading to state-of-the-art results in numerous fields. While MoE has been mostly investigated for the pre-training stage, its use in parameter-efficient transfer learning (PETL) settings is underexplored. To narrow this gap, this paper attempts to demystify the use of MoE for PETL of Audio Spectrogram Transformers to audio and speech downstream tasks. Specifically, we propose Soft Mixture of Adapters (Soft-MoA). It exploits adapters as the experts and, leveraging the recent Soft MoE method, it relies on a soft assignment between the input tokens and experts to keep the computational time limited. Extensive experiments across 4 benchmarks demonstrate that Soft-MoA outperforms the single adapter method and performs on par with the dense MoA counterpart. We finally present ablation studies on key elements of Soft-MoA. Our code is available at `https://github.com/umbertocappellazzo/PETL_AST`.

Index Terms: Audio Spectrogram Transformer, Efficient Fine-tuning, Adapters, Mixture of Experts, Soft Mixture of Adapters

1. Introduction

Large pre-trained audio and speech models have exhibited outstanding performance when fine-tuned with task-specific data [1, 2]. A common practice entails the adaptation of the whole model to each downstream task (i.e., full fine-tuning) [3, 4]. However, this paradigm has two major limitations: 1) adapting the entire pre-trained model is expensive and usually demands a significant volume of training data; 2) storing a copy of the model for each downstream task is unfeasible and impractical.

Given these shortcomings, current research mainly revolves around learning a small fraction of task-specific parameters, while keeping the pre-trained model frozen. This approach is known as *parameter-efficient transfer learning* (PETL) and includes several nuances. For example, prompt-tuning methods [5, 6] introduce trainable task-specific tokens into one or multiple layers. LoRA [7, 8] uses trainable low-rank matrices to approximate the weight matrices. Adapter-based methods [9, 10, 11] add lightweight modules (adapters) with bottleneck architecture comprising two fully-connected layers. The adapter can be inserted after both the multi-head self-attention and fully-connected feed-forward network blocks (*Houlsby*) [12], or only after the feed-forward (*Pfeiffer*) [10]. Adapters can also be scaled and shifted to modulate the pre-trained features [13], or their down/up projections can be shared across different layers and low-dimensional re-scaling coefficients are learned [14]. In the speech field, PETL methods have been recently investigated and compared in [15, 16].

Very recently, Mixture of Experts (MoE) models have shown remarkable results in natural language processing, pushing large language models to the limit, facilitating the effective scaling of Transformers and State Space Models while concurrently reducing computational costs [17, 18, 19, 20]. The MoE paradigm relies on the idea that sub-modular components, the *experts*, can specialize in different inputs and scale the model's capacity. While most works have focused on the use of MoE during the pre-training stage, only few works have leveraged MoE for efficient fine-tuning [18, 21, 22]. In the latter case, each expert is usually represented by a single adapter, and the model is referred to as Mixture of Adapters (MoA). However, these works usually target language-based tasks, whereas pure audio/speech classification tasks have not been taken into account before. Therefore, in this paper, we investigate the use of MoA for the Audio Spectrogram Transformer (AST), a powerful foundation model achieving state-of-the-art results on various audio/speech tasks [2], and we ask the following question:

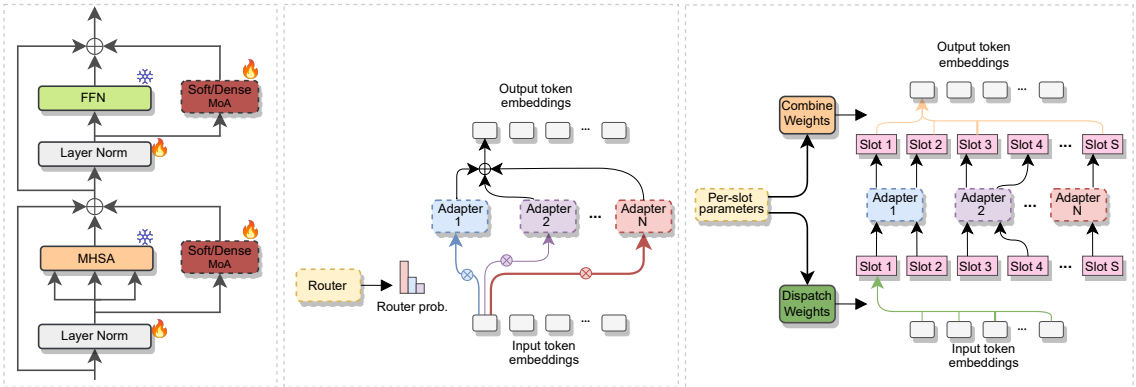
> **(Q)** *Can we leverage MoAs for the efficient fine-tuning of AST to audio/speech downstream tasks?*

To answer the above research question **(Q)**, we study the MoA's adoption for PETL of AST on four popular audio and speech benchmarks. Specifically, we propose to adapt a recent *sparse* version of MoE called *Soft-MoE* [23] to our PETL setting, whereby each expert only handles a small number of slots that are the result of a weighted combination of all input tokens. We call it **Soft-MoA**, and we compare it with the standard single adapter approach and with the dense version of MoA that requires each adapter to process all the input tokens (we refer to it as Dense-MoA). *By doing this, we are able to scale the number of adapters while keeping the computational cost limited as well as updating only a small fraction of parameters, thus leveraging the strengths of both the MoE and PETL paradigms*. We empirically show that both Soft and Dense MoA outperform the single adapter approach, both for the Pfeiffer and Houlsby configuration, leading to accuracy improvement of up to 2.5%; also, Soft-MoA attains performance parity with Dense-MoA while drastically trimming down the training cost. Finally, we further demonstrate the effectiveness of Soft-MoA by carrying out extensive ablation experiments revealing that ❶) both Soft and Dense-MoA gains over the single adapter strategy are more evident when fewer parameters are available, ❷) Soft-MoA is robust to "*expert imbalance*", thus ensuring that all experts are involved in the learning process, and ❸) Soft-MoA attains the best performance accuracy when few slots ($1/2$) and several experts are used rather than the opposite case as multiple slots tend to learn redundant information.

2. Methodology

In this section, we first give a brief recap of the AST model and the standard single adapter approach. In section 2.3, we present

10.21437/Interspeech.2024-38



(a) AST layer + Soft/Dense MoA. **(b)** Dense MoA. **(c)** Soft MoA.

Figure 1: ***(a)*** *For each AST layer, the Soft/Dense MoA blocks are inserted parallel to MHSA (Pfeiffer) or parallel to both MHSA and FFN sub-layers (Houlsby).* ***(b)*** *Illustration of Dense-MoA, whereby each expert contribution, scaled by the router's distribution (thickness of the arrows), is summed to produce the final output.* ***(c)*** *In Soft-MoA, each expert only processes a subset of slots (here 2), and each slot accepts as input a weighted combination of all input tokens (thickness of the arrows). Note that the trainable parameters are represented by dashed blocks. Best viewed in color.*

the details of the Dense-MoA and Soft-MoA approaches.

2.1. AST model recap

The Audio Spectrogram Transformer (AST) is an attention-based model that achieves state-of-the-art results on various audio and speech tasks [2, 24]. The AST model receives as input audio spectrograms that are patchified and then a linear projection is applied to each patch. This results in a sequence of L tokens of size $d = 768$, which we refer to as $\mathbf{X} \in \mathbb{R}^{L \times d}$. AST comprises 12 attention layers, each of which is composed of two sub-layers: a multi-head self-attention (MHSA) and a fully-connected feed-forward (FFN) module.

2.2. Adapters

Adapters are light subnetworks that are inserted into every layer of the AST model. To keep the parameters limited, adapters exploit a bottleneck architecture. The input sequence of hidden dimension d is first down-projected into a low-dimensional space with size r (the bottleneck dimension), and then up-projected back to the original dimension d. A non-linear activation function is also applied in-between the two fully-connected layers.

While the bottleneck adapter is the most common design, recent works have also explored convolution-based adapters mainly for vision tasks (e.g., Convpass) [25, 26]. In addition to this, adapters usually follow a Pfeiffer [10] or Houlsby [12] configuration: the former places the adapter parallel or sequentially to the MHSA or FFN sub-layer, whereas the latter includes the adapter on both sub-layers.

2.3. Dense and Soft MoA

Dense-MoA. It encompasses a set of N *"expert" adapters* $E_1, \dots E_N$ and a *router network* R that learns the optimal distribution over the adapters for a given input sequence. In its simplest form [27, 18], the router is a dense fully-connected layer with weights $\mathbf{W} \in \mathbb{R}^{d \times N}$ followed by a *softmax* function that takes as input the sequence $\mathbf{X}$ and merges the output of each adapter using the gating scores $g_1, \dots g_N$ to yield the output sequence $\mathbf{Y}$:

$$g_i = R(\mathbf{X})_i = \text{softmax}(\mathbf{XW}), \tag{1}$$

$$\mathbf{Y} = \sum_{i=1}^{N} g_i \cdot E_i(\mathbf{X}). \tag{2}$$

If all the N adapters take part in the computation of the output of a given input (scaled by the router's distribution), then we refer to this as *Dense*-MoA (alternatively we can think of this as *ensemble* MoA). Whereas this approach would cater to exact computation of gradients and end-to-end-learning, it would also incur a substantial increase in computational costs since each input token is computed by every expert rather than a single expert. To circumvent the above issue, we propose to adapt a recent method called *Soft Mixture of Experts* [23] to our PETL setting where each expert is an adapter, and we call it *Soft-MoA*. Note that in our setting only the adapters are actually learned whilst the backbone model is frozen.

Soft-MoA. Rather than feeding all input tokens to each expert, Soft-MoA passes a different weighted soft combinations of all input tokens to each expert. Unlike other sparse techniques like Top-k [28] whereby only the k experts that are assigned the highest router's probability are activated, Soft-MoA provides fully-differentiable operations, better training stability, and immunity to "token dropping" and "expert imbalance" issues [23]. In practice, each adapter processes p slots, and each slot has a corresponding d-dimensional vector of parameters. These parameters are denoted by $\mathbf{\Phi} \in \mathbb{R}^{d \times (N \cdot p)}$. The input slots, $\tilde{\mathbf{X}}$, are computed as the convex combination of all the L input tokens:

$$\tilde{\mathbf{X}} = \mathbf{D}^\top \mathbf{X}, \quad \mathbf{D}_{i,j} = \frac{\exp((\mathbf{X\Phi})_{i,j}}{\sum_{h=1}^{L} \exp((\mathbf{X\Phi})_{h,j})}. \tag{3}$$

$\mathbf{D}$ is called the *dispatch weights* and corresponds to applying a softmax along the columns of $\mathbf{X\Phi}$. At this point, each adapter processes the corresponding slots: $\tilde{\mathbf{Y}}_i = E_{\lfloor i/p \rfloor}(\tilde{\mathbf{X}}_\mathbf{i})$. Finally, the output tokens $\mathbf{Y}$ are the result of a convex combination of all $(N \cdot p)$ slots:

$$\mathbf{Y} = \mathbf{C}\tilde{\mathbf{Y}}, \quad \mathbf{C}_{i,j} = \frac{\exp((\mathbf{X\Phi})_{i,j}}{\sum_{h=1}^{N \cdot p} \exp((\mathbf{X\Phi})_{i,h})}. \tag{4}$$

The matrix $\mathbf{C}$ is referred to as the *combine weights*, and is equivalent to applying a softmax over the rows of $\mathbf{X\Phi}$.

We provide an overview of Soft and Dense MoA in Figure 1. Finally, for our experiments, following [15] that show that inserting the adapter in parallel achieves better performance than

Table 1: *Performance evaluations of Dense and Soft-MoA on 4 benchmarks for the Pfeiffer configuration. We report the top-1 accuracy for each dataset, the average over the four datasets (**Avg**), and the average train step time in milliseconds (**Time**).*

Method	# params	ESC-50	US8K	GSC	FSC	Avg	Time (ms)
Full FT	85.5M	87.48	84.31	97.31	93.29	90.07	645
Linear	9-40K	75.85	77.93	41.78	27.52	55.77	226
BitFit	102K	86.05	82.17	85.51	63.85	79.40	513
DPT	230K	86.52	83.67	89.18	68.60	81.99	561
Pref-T	221K	82.93	81.39	83.46	55.75	75.88	529
LoRA	221K	86.45	83.83	93.61	76.00	84.97	525
Bottleneck Adapter							
Single	470K	88.65	83.36	93.53	78.19	85.93	513
D-MoA 14	535K	89.55	84.30	93.89	82.43	**87.54**	**1689**
S-MoA 14	535K	89.08	84.88	93.91	82.48	**87.59**	626
Convpass Adapter							
Single	491K	87.93	83.38	93.47	77.62	85.60	515
D-MoA 14	535K	89.30	84.32	93.70	83.52	**87.71**	**1727**
S-MoA 14	535K	88.43	84.29	93.36	80.36	**86.61**	638

sequentially, we place the MoA block *parallel* to the MHSA layer only (i.e., Pfeiffer) or *parallel* to both the MHSA and FFN layers (i.e., Houlsby). The number of slots p is an hyper-parameter, and we elaborate on its optimal value on Section 3.3.

3. Experiments and Discussion

3.1. Implementation Details

For our experiments, we mainly follow the implementation details of [15] to provide a fair comparison.

Datasets. We evaluate the PETL methods on three audio/speech downstream classification tasks. (1) **Audio classification**: we use the ESC-50 and UrbanSound8K (US8K) datasets. ESC-50 [29] consists of $2,000$ 5-second-long environmental audio recordings of 50 classes. US8K [30] includes $8,732$ labeled sound excerpts of urban sounds from 10 classes. (2) **Keyword spotting**: Speech Commands V2 [31] has $105,829$ 1-second recordings of 35 speech commands. (3) **Intent classification**: Fluent Speech Commands (FSC) [32] includes $30,043$ English utterances spanning 31 classes.

PETL baselines. We include two traditional fine-tuning strategies: **full fine-tuning** (Full-FT), which finetunes the full pre-trained AST model; and **linear probing**, which only finetunes the classification head. Following [15] we include some common PETL baselines: **BitFit** [33], **deep prompt-tuning** (DPT) [5], **prefix-tuning** (Pref-T) [6] and **LoRA** [7]. For the analysis of MoA, we take into account both *Bottleneck* [12] and *Convpass* [34] adapters. We report **Dense** and **Soft-MoA** (D/S-MoA) with 14 or 7 adapters for the *Pfeiffer* and *Houlsby* configuration, respectively, and we compare them with the standard implementation using a single adapter per layer (**Single**).

Training Details. For all experiments we use the AST model pre-trained on ImageNet-21K [35] and AudioSet [36] provided by the Huggingface Transformers library [37]. The model has around 85.5 million parameters, and the hidden size is 768. Please refer to [15] for the training details of the baselines (LoRA, DPT etc.). For MoA experiments, we use AdamW optimizer with cosine annealing scheduler and weight decay set to 0.1. For the ESC-50 and US8K datasets, we run 5-fold and 10-fold cross validation as suggested in the original papers. Except US8K that does not provide a validation set by default, for the others we set the hyper-parameters using the validation set.

Table 2: *Results of D/S-MoA for the Houlsby configuration. The number of parameters coincides with Pfeiffer as we still use 14 adapters split equally between MHSA and FFN layers.*

Method	ESC-50	US8K	GSC	FSC	Avg
Bottleneck Adapter					
Single	88.00	82.80	91.75	78.71	85.32
D-MoA 7	87.33	83.78	94.11	82.64	**86.97**
S-MoA 7	87.13	83.77	93.67	81.41	**86.50**
Convpass Adapter					
Single	87.15	82.75	92.55	77.79	85.06
D-MoA 7	87.31	83.77	93.20	82.26	**86.63**
S-MoA 7	88.13	83.87	92.69	81.69	**86.60**

3.2. Main Results and Discussion

Table 1 presents the performance comparisons between the single adapter approach and Soft/Dense MoA, as well as some other common PETL methods. The single adapter approach has bottleneck dimension equal to 24, whereas Soft/Dense-MoA include 14 adapters, each with bottleneck dimension 1, and one slot is used for each adapter. From table 1 we observe that both MoAs outperform the single adapter, leading to up to 2.5 % performance improvement on average for the Bottleneck case, while for Convpass we notice that Soft-MoA is slightly worse than Dense-MoA, but still better than the single adapter. In general, the biggest gain is obtained with the FSC dataset (up to 5.5 and 7.6 %). Indeed, FSC is the more challenging dataset as it includes longer speech audio data, thus we argue that multiple adapters can specialize in learning different information, and consequently leading to better performance. We also notice that the GSC dataset does not benefit much from the use of MoA architecture. We surmise that a single adapter already achieves very competitive performance and so the use of multiple smaller adapters is not helpful. Another pivotal aspect is the extra computational cost brought by MoAs, estimated as the average train step time in milliseconds. Whereas Dense-MoA incurs a considerable increase in time (more than 3x with respect to the single adapter), S-MoA, instead, requires only a limited extra time, while guaranteeing on-par performance.

Finally, we test Soft-MoA's efficacy for the Houlsby configuration, where the MoA block is also inserted parallel to the

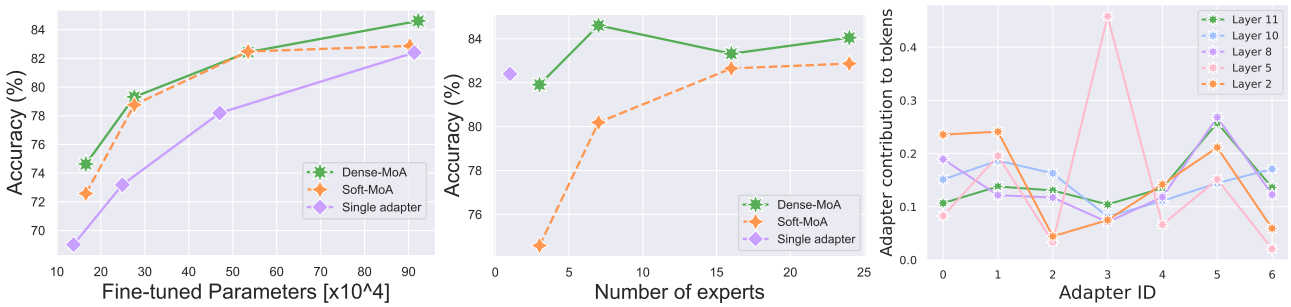


Figure 2: ***(Left)***. *The accuracy trend as more parameters are used.* ***(Middle)***. *The effect of the number of adapters given a fixed parameters budget.* ***(Right)***. *Adapters contribution to the output tokens for various layers. Results reported for FSC.*

FFN sub-layer. Table 2 confirms the superiority of both MoAs over the single adapter.

3.3. Ablation Studies

We now conduct some ablation studies to evaluate the effectiveness of Soft-MoA under different settings. We focus on the Pfeiffer Bottleneck configuration, and on the FSC dataset.

Increasing the Parameters Budget. We examine the methods' behaviour as we increase the number of trainable parameters. For the single adapter, we increase the parameters by making the bottleneck dimension r larger, while for MoAs we keep it to 1 and we increase the number of adapters. From Figure 2 **(Left)** we observe that Soft-MoA outperforms the single adapter, although when more and more parameters are available the two methods tend to achieve similar results, thus showing that using a single adapter is a good alternative when scaling the number of parameters is sustainable.

Few-big vs Many-small Adapters. We now investigate how the MoA methods scale with respect to the number of adapters N. Regardless of N, we fix the number of learnable parameters to around 900K to have a fair comparison. In this way, we want to figure out if having more adapters with a smaller bottleneck dimension is better than having a few but "bigger" (in terms of parameters) adapters. The Figure 2 **(Middle)** shows that Dense-MoA, due to its intrinsic dense structure, reaches the peak performance when $N = 7$, and then adding more adapters does not lead to additional improvement. On the contrary, Soft-MoA depends heavily on N, and only when this number is large enough does it attain good performance. This trend is in line with that of the original Soft MoE paper [23].

Adapters Contribution to the Output Tokens and Specific Classes. By design, the computation of the final output tokens depends on a linear combination of all the adapters' slots. We want to verify whether all adapters contribute to the output sequence. We fix one slot per adapter and consider 7 adapters, and we approximate the contribution of each adapter by averaging their coefficients in the linear combinations for all output tokens. We average over all the batches of the test set and report the adapter contribution for different layers in Figure 2 **(Right)**. We see that some adapters have a bigger impact than others, but all of them contribute to the final output tokens. Therefore, Soft-MoA does not suffer from the expert imbalance issue, namely few adapters monopolize the output contribution while the others are overshadowed, an issue that affects other routing strategies like Top-k [28, 23]. In addition to this, *we compute the contribution of each adapter to each class*. To do this, for each sample of each class, we compute the contribution of each adapter and then we average over the total number of samples per class (for this reason the sum of each row of the heatmap does not sum to 1). We observe from Figure 3 that some adapters specialize more for some classes than others (adapter 0 has a high contribution for classes 26-29, adapter 1 for classes 7, 20-22). We also see that the adapter with ID 5 has a strong contribution for several classes.

N/p	Acc
2/14	78.52
4/6	80.26
6/4	81.65
8/3	82.36
12/2	**83.24**
24/1	**82.87**

Table 3: *Optimal trade-off between the number of adapters N and slots p.*

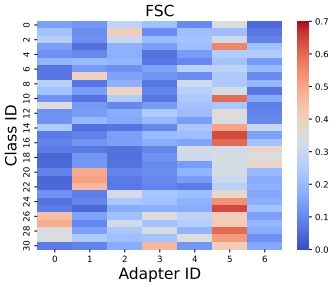


Figure 3: *Distribution of expert activation frequencies per class.*

Optimal Trade-off between Slots and Adapters. The number of slots p is an important hyper-parameter of Soft-MoA, thus we examine its optimal value. We notice that if we set the number of slots equal to the number of tokens L, Soft-MoA boils down to Dense-MoA, so it is crucial to keep p small. For our experiments, depending on the dataset, L is between 100 and 500, and setting p up to 14 is a reasonable choice. We report the results for FSC in Table 3 and we see that, with the same number of trainable parameters, having more adapters with few slots brings better results than having few adapters but many slots. We speculate that this happens because multiple slots corresponding to the same adapter might have a tendency to learn similar concepts and become redundant, whereas using more adapters ends up learning more diverse information.

4. Conclusion

In this paper, we propose Soft Mixture of Adapters to efficiently fine-tune the AST model on various audio/speech downstream tasks. Soft-MoA relies on multiple adapters that take as input a soft convex combination of all the input tokens, thus reducing the computational cost of the dense counterpart. Extensive experiments on 4 benchmarks show that Soft-MoA performs on par with Dense-MoA, and it outperforms the single adapter strategy, confirming itself as a strong method also for parameter-efficient transfer learning settings. To strengthen our analysis, we carry out ablation studies revealing that Soft and Dense MoA provide bigger gains over the single adapter when the parameters budget is limited. We also show that Soft-MoA scales better with the number of adapters and that it is sufficient to use only 1 or 2 slots to achieve the optimal performance.

5. Acknowledgments

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

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 *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[2] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *Proceedings of Interspeech*, 2021.

[3] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of pre-trained cnns for audio classification using transfer learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 72, 2021.

[4] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *arXiv preprint arXiv:2111.02735*, 2021.

[5] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–727.

[6] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *Proceedings of ACL*, 2021.

[7] 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," *Proceedings of ICLR*, 2022.

[8] Y. Zeng and K. Lee, "The expressive power of low-rank adaptation," *Proceedings of ICLR*, 2024.

[9] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[10] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," *Proceedings of EACL*, 2021.

[11] S. Jie, H. Wang, and Z.-H. Deng, "Revisiting the parameter efficiency of adapters from the perspective of precision redundancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 217–17 226.

[12] 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.

[13] D. Lian, D. Zhou, J. Feng, and X. Wang, "Scaling & shifting your features: A new baseline for efficient model tuning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 109–123, 2022.

[14] W. Dong, D. Yan, Z. Lin, and P. Wang, "Efficient adaptation of large vision transformer via adapter re-composing," *Advances in Neural Information Processing Systems*, 2023.

[15] U. Cappellazzo, D. Falavigna, A. Brutti, and M. Ravanelli, "Parameter-efficient transfer learning of audio spectrogram transformers," *arXiv preprint arXiv:2312.03694*, 2023.

[16] T.-H. Lin, H.-S. Wang, H.-Y. Weng, K.-C. Peng, Z.-C. Chen, and H.-y. Lee, "Peft for speech: Unveiling optimal placement, merging strategies, and ensemble techniques," *arXiv preprint arXiv:2401.02122*, 2024.

[17] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[18] T. Zadouri, A. Üstün, A. Ahmadian, B. Ermiş, A. Locatelli, and S. Hooker, "Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning," *Proceedings of ICLR*, 2024.

[19] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.

[20] M. Pióro, K. Ciebiera, K. Król, J. Ludziejewski, and S. Jaszczur, "Moe-mamba: Efficient selective state space models with mixture of experts," *arXiv preprint arXiv:2401.04081*, 2024.

[21] Y. Wang, S. Agarwal, S. Mukherjee, X. Liu, J. Gao, A. H. Awadallah, and J. Gao, "AdaMix: Mixture-of-adaptations for parameter-efficient model tuning," in *Proceedings of EMNLP*, 2022, pp. 5744–5760.

[22] A. Mehrish, A. R. Kashyap, L. Yingting, N. Majumder, and S. Poria, "Adaptermix: Exploring the efficacy of mixture of adapters for low-resource tts adaptation," *Interspeech*, 2023.

[23] J. Puigcerver, C. Riquelme, B. Mustafa, and N. Houlsby, "From sparse to soft mixtures of experts," *Proceedings of ICLR*, 2024.

[24] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[25] S. Jie and Z.-H. Deng, "Convolutional bypasses are better vision transformer adapters," *arXiv preprint arXiv:2207.07039*, 2022.

[26] H. Chen, R. Tao, H. Zhang, Y. Wang, W. Ye, J. Wang, G. Hu, and M. Savvides, "Conv-adapter: Exploring parameter efficient transfer learning for convnets," *arXiv preprint arXiv:2208.07463*, 2022.

[27] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.

[28] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *Proceedings of ICLR*, 2017.

[29] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[30] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[31] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[32] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *Proceedings of Interspeech*, 2019.

[33] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of ACL*, 2022, pp. 1–9.

[34] S. Jie and Z.-H. Deng, "Convolutional bypasses are better vision transformer adapters," *arXiv preprint arXiv:2207.07039*, 2022.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[36] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[37] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.