INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Adapter Incremental Continual Learning of Efficient Audio Spectrogram Transformers

Nithish Muthuchamy Selvaraj[*1], *Xiaobao Guo*[*†12], *Adams Kong*[2], *Bingquan Shen*[3], *Alex Kot*[1]

[1]Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]DSO National Laboratories, Singapore

{ms.nithish, adamskong, eackot}@ntu.edu.sg, xiaobao001@e.ntu.edu.sg, sbingqua@dso.org.sg

Abstract

Efficient tuning of neural networks for continual learning with minimal computational resources remains a challenge. In this paper, we propose continual learning of audio classifiers with parameter and compute efficient Audio Spectrogram Transformers (AST). To reduce the trainable parameters without performance degradation we propose AST with Convolutional Adapter, which has less than 5% of trainable parameters of full fine-tuning. To reduce the computational complexity of self-attention, we introduce a novel Frequency-Time factorized Attention (FTA) method that achieves competitive performance with only a factor of the computations. Finally, we formulate our method called Adapter Incremental Continual Learning (AI-CL), as a combination of the parameter-efficient Convolutional Adapter and the compute-efficient FTA. Experiments on ESC-50, SpeechCommandsV2, and Audio-Visual Event benchmarks show that our proposed method efficiently learns new tasks and prevents catastrophic forgetting. *Code is available at* https://github.com/NMS05/Adapter-Incremental-Continual-Learning-AST.

Index Terms: Continual Learning, Audio Spectrogram Transformer, Adapter, Self-Attention

1. Introduction

Continual learning [1] of new knowledge and skill acquisition are the desirable traits for intelligent machines. However, in Deep Learning, neural networks may forget previous knowledge [2] due to the optimization of network weights for new tasks, leading to catastrophic forgetting. Many works have been proposed to address this issue by constraining the weights of neural nets [3, 4] or using data (pseudo-data) of previous tasks [5]. A simple way to mitigate this issue is to assign task-specific sub-networks, where only a sub-network is optimized for new tasks while other parameters are task-independent and can be shared across tasks. This approach is particularly effective for Task Incremental Continual Learning (TI-CL), which requires a task-ID to route the data to the corresponding sub-network. As the model is incrementally trained on new tasks, its size grows sub-linearly.

This paper explores TI-CL of audio classifiers with Audio Spectrogram Transformers (AST) [6], which achieved state-of-the-art results on several audio benchmarks [7, 8, 9]. However, there are two main issues with AST that must be addressed for sequential training: parameter inefficiency and computational inefficiency.

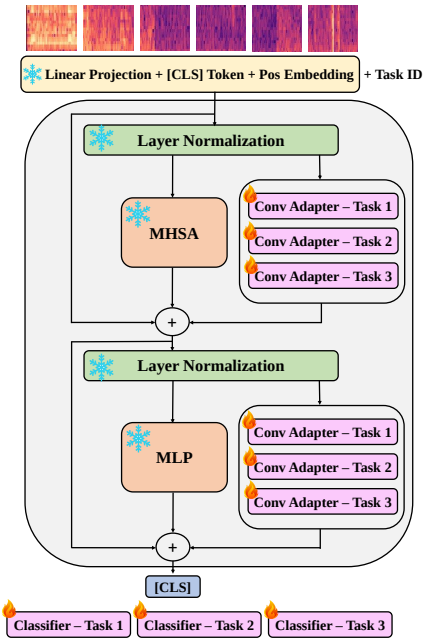


Figure 1: *Adapter Incremental Continual Learning of Audio Spectrogram Transformers.*

Parameter Inefficiency. In TI-CL, the use of pre-trained transformer-based models like AST can lead to parameter inefficiency due to a large number of trainable parameters in full-finetuning for sequential tasks. This can cause overfitting, especially when the sequential tasks have limited data.

Computational Inefficiency. The transformer's self-attention mechanism [10] has quadratic computational complexity. Hence, a large number of tokens extracted from larger spectrograms (from long duration audio) rapidly increases the number of computations. However, audio spectrograms cannot be resized since their characteristics are determined by the audio duration and the number of frequency bins. Resizing audio spectrograms can lead to a loss of critical information and adversely affect their quality. Hence, transformer-based AST shows significant computational inefficiency when processing long-duration audio.

Therefore, we propose a TI-CL method based on AST and address the issues of parameter and computational efficiency. We leverage Parameter Efficient Transfer (PET) methods to improve the parameter efficiency of AST. Our study evaluates the efficacy of various PET methods for AST on ESC-50 [7] and SpeechCommandsV2 [8] benchmarks and proposes Convolutional Adapters to address parameter inefficiency. Note that the performance of PET methods for AST audio classifiers has

[*]These authors contributed equally to this work
[†]Corresponding Author

10.21437/Interspeech.2023-1189

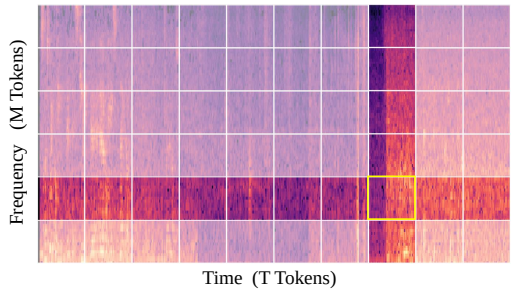


Figure 2: *Frequency-Time factorized Attention for a (yellow) token along the frequency and time axis.*

not been studied before. The convolutional adapters perform as well as fully fine-tuned models in high-resource settings and even outperform them in low-resource settings with <5% of the trainable parameters.

Next, we propose Frequency-Time factorized Attention (FTA) to address computational inefficiency in self-attention for long-duration audio spectrograms. Unlike traditional self-attention, FTA enables an arbitrary token to attend only to the frequency and temporal tokens that share the same position index in either axis, thereby leveraging the orthogonal nature of frequency and time in spectrograms (see Fig. 2). This factorization greatly reduces complexity and improves computational efficiency. To achieve both parameter and computational efficiency, we combine Convolutional Adapter and FTA for TI-CL of audio classification.

The main contributions of this paper can be summarized as follows,

- We provide an empirical study on the performance of various PET methods for AST.
- We propose TI-CL of audio classifiers with parameter-efficient AST, using Convolutional Adapters.
- We introduce a novel Frequency-Time factorized Attention (FTA) for compute-efficient AST.
- Through comprehensive experiments we demonstrate the advantages of the proposed approach for TI-CL of audio classifiers.

2. Related work

2.1. Continual Learning for Audio

To prevent catastrophic forgetting in continual learning, various methods have been proposed. For example, GIM [11] incrementally adds new modules to capture drifts in input distribution, DFWF [12] uses a knowledge distillation loss to preserve memory from the original model, and static memory networks [13] introduce static memory to reduce memory usage and model complexity. Few-shot CL [14] enables fast and interactive model updates in a few-shot learning framework to expand the audio classifier to recognize novel classes, while CTR [15] addresses both catastrophic forgetting and knowledge transfer issues with a pair of continual learning plugin modules.

2.2. Parameter Efficient Transfer

Many recent works have focused on efficient transfer learning and fine-tuning techniques for downstream tasks, such as Adapter for NLP [16] and similar methods like LoRA [17], AdaptFormer [18], and ConvPass [19]. These methods achieve efficient fine-tuning by inserting small trainable bottleneck modules at different locations inside a transformer encoder while freezing other parameters during training. Commonly used methods involve a down projection followed by an up projection. Other methods tune specific parameters in the network, such as BitFit [20], which adapts the model for different tasks by tuning the bias terms of the transformer layers, LayerNorm Tune [21], which tunes the affine transformation parameters in the encoder normalization layers, and Prompt Tuning [22], which optimizes a set of learnable latent tokens that are prepended to the input sequence at every encoder layer for transfer learning.

3. Methodology

3.1. Continual Learning (CL) and AST audio classifier

The objective of continual learning is to sequentially train a parameterized model $f_{\boldsymbol{\theta}}$ over a set of n tasks $D \in \{D_1, D_2, ..., D_n\}$. Each task is defined by $D_i = (X_i, Y_i), i \in [1, n]$, where X is a set of input samples and Y is a set of corresponding labels. The parameterized function $f_{\boldsymbol{\theta}} : x \rightarrow y$ maps the input $x \in X$ to the corresponding label $y \in Y$ and the goal of CL is to train $f_{\boldsymbol{\theta}}$ such that it can correctly predict the label y for an unseen arbitrary input x sampled across D.

If D is an audio classification task, then $f_{\boldsymbol{\theta}}$ is a pre-trained AST model with total weights $\boldsymbol{\theta}$, $x \in X$ is a spectrogram image and $y \in Y$ is the corresponding audio class label. $f_{\boldsymbol{\theta}}$ extracts tokens $\boldsymbol{Z} = \{z_1, z_2, ..., z_{MT+1}\}$ from x, where $z \in \mathbb{R}^d$, M and T denote the number of tokens in frequency and time axis, d is the embedding dimension and 1 denotes the class token. These tokens are processed by a series of 12 transformer encoders with Multi-Head Self-Attention (MHSA), Multi Layer Perceptron (MLP) and Layer Normalization (LN) sublayers, and can be formulated as,

$$\begin{aligned} \boldsymbol{Z'_l} &= MHSA(LN_1(\boldsymbol{Z_{l-1}})) + \boldsymbol{Z_{l-1}}, \\ \boldsymbol{Z_l} &= MLP(LN_2(\boldsymbol{Z'_l})) + \boldsymbol{Z'_l}, \end{aligned} \tag{1}$$

where l denotes the layer number and $\boldsymbol{Z_l}$ is the extracted tokens from layer l.

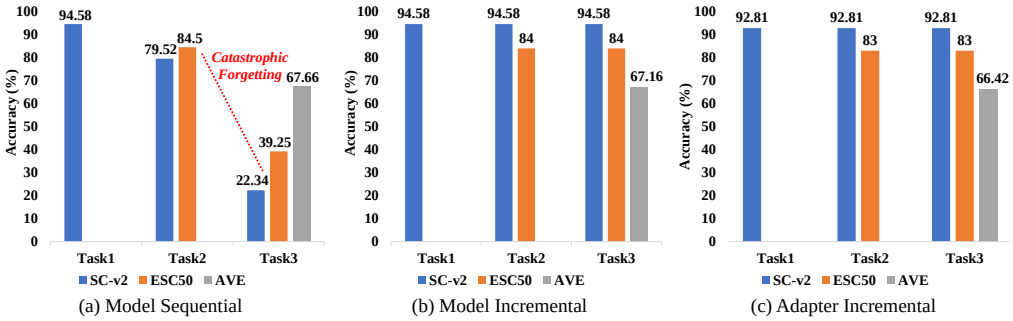
3.2. Adapter Incremental Continual Learning of AST

Task Incremental Continual Learning is one of the three scenarios for CL [23], where it assumes that the tasks D_i are disjoint and the task ID i is known both during training and inference. Full-finetuning $f_{\boldsymbol{\theta}}$ on the sequential tasks by optimizing $\boldsymbol{\theta}$ may not be efficient and may lead to the overfitting issue. A parameter incremental approach to solve TI-CL involves training a parameterized network with multiple task-specific sub-modules denoted as $f_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}}$, where θ is the shared task-independent parameter, $\delta\boldsymbol{\theta} \in \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta_n}\}$ are the task-specific parameters and $\boldsymbol{\theta}$ is much larger than $\delta\boldsymbol{\theta}$.

We propose an adapter incremental method for TI-CL called Adapter Incremental Continual Learning (AI-CL), where a Convolutional Adapter (CA) is incrementally added and trained for each task while keeping the shared $\boldsymbol{\theta}$ frozen. We denote the weights of task-specific CA as $\delta\boldsymbol{\theta_i}$ for every new task D_i. CA has a bottleneck structure, which consists of a down-projection followed by an up-projection with an additional 2D convolution layer in between. The inputs tokens are reshaped to $M \times T$ before the convolution operation, with the exception of the class token, and then reverted back to its original shape before up-projection. CA processes arbitrary length input tokens $\boldsymbol{z} \in \mathbb{R}^d$ as,

$$CA(\boldsymbol{z}) = \boldsymbol{W_{up}}(GELU(Conv2D(\boldsymbol{W_{down}}(\boldsymbol{z})))), \tag{2}$$

where $\boldsymbol{W_{down}} \in \mathbb{R}^{d \times d'}$, $\boldsymbol{W_{up}} \in \mathbb{R}^{d' \times d}$ and $d' << d$. CA



(a) Model Sequential (b) Model Incremental (c) Adapter Incremental

Figure 3: *Performance of the AST model in TI-CL setup for three training modes.*

runs parallel to both MHSA and MLP layers, which can be represented as,

$$\boldsymbol{Z_l'} = MHSA(LN_1(\boldsymbol{Z_{l-1}})) + \boldsymbol{Z_{l-1}} + CA_1(LN_1(\boldsymbol{Z_{l-1}})),$$
$$\boldsymbol{Z_l} = MLP(LN_2(\boldsymbol{Z_l'})) + \boldsymbol{Z_l'} + CA_2(LN_2(\boldsymbol{Z_l'})).$$

The proposed AI-CL method using CA is parameter efficient since only the CA weights $\delta\boldsymbol{\theta_i}$ are trainable and saving these weights also occupies less storage. The backbone weights $\boldsymbol{\theta}$ are frozen and shared across tasks, both during the training and inference stage. During inference, when a test audio spectrogram x is passed along with the task ID i, the AST model routes the tokens $\boldsymbol{Z}$ to the corresponding CA with the parameter $\delta\boldsymbol{\theta_i}$ and the corresponding classifier. The AST model with multiple task-specific CAs is illustrated in Fig 1.

3.3. Frequency-Time factorized Attention (FTA)

While the AI-CL approach is parameter-efficient, the use of self-attention in AST results in a quadratic increase in computations (*i.e.*, the number of floating point operations or FLOPS) for larger spectrograms. To address this issue, prior alternatives to self-attention either limit self-attention to a local window [24] or factorize self-attention along two orthogonal axis [25], but they were developed for images and videos.

Inspired by the factorization approach [25, 26], we propose Frequency-Time factorized Attention (FTA) in the AI-CL method as shown in Figure 2. It factorizes self-attention across the frequency and time axis of a spectrogram, by masking out the undesired tokens. This approach makes AST more computationally efficient, with attention along the frequency (vertical) axis learning the distribution of various frequency components at a given time interval, and attention along the time (horizontal) axis learning how a frequency component evolves over time. The only exception is the $[CLS]$ token, which attends to all the tokens (including itself) since it must summarize the semantic information in a spectrogram. For a token $\boldsymbol{Z} \in \mathbb{R}^{(MT+1)\times d}$, the computation complexity $\mathcal{O}$ of Global Self-Attention (GSA) and FTA can be calculated as follows,

$$\begin{aligned} \mathcal{O}_{GSA} &= (MT+1)^2 * d, \\ \mathcal{O}_{FTA} &= (MT(M+T+1)+1) * d, \end{aligned} \tag{3}$$

where $(M+T) << MT$. Thus, when M and T grow, FTA has much fewer computations than GSA. Empirically, we show that the proposed Frequency Time factorized Attention (FTA) achieves competitive performance to global self-attention with only a fraction of the computations.

4. Results

4.1. Experimental Setup

Datasets. The datasets used for PET evaluation and TI-CL experiments are:

- ESC-50 [7], which contains 2,000 5-second audio recordings organized into 50 classes for environmental sound classification. The standard 5-fold cross-validation is used unless otherwise specified.
- Speech Commands V2 (SCv2) [8], which includes 105k 1-second recordings of 35 speech classes for speech recognition. The standard training and test set split is used with 84,843 and 11,005 samples respectively.
- AVE [27], an event localization dataset of 4,143 samples covering 28 events with a duration of 10 seconds (long duration). Only the audio modality is used, and the original train-test split for audio classification is followed.

Model. Our system is built upon the AST model, a ViT/B-16 model with 12 transformer encoders pre-trained on the ImageNet-21k dataset (weights obtained from timm library). We process audio input by converting the waveform into a log mel spectrogram with 128 Mel bins, a 25ms Hamming window, and a hop length of 10ms, without any data augmentation. Tokens are extracted using a convolutional feature extractor with a kernel size of 16, a stride of 10, and a dimensionality of 768, with position embeddings added via bilinear interpolation. The model is trained using Adam optimizer with a learning rate of 3e-4 and cross-entropy loss, with batch sizes of 128/32/12 for the SCv2/ESC-50/AVE datasets. We train the model for 5/20/15 epochs on the respective datasets.

4.2. Evaluation of PET methods

While several PET methods have been proposed for NLP and Vision tasks, their effectiveness in audio classification remains largely unexplored. In this study, we evaluated several PET methods on the ESC-50 and SCv2 datasets, and found that AdaptFormer [18] and ConvPass [19] achieved the highest performance (see Table 2). The Linear method was simply adding a trainable linear layer for classification in Table 2. Notably, ConvPass achieved comparable performance to full fine-tuning on SCv2 (with 2.7k samples per class), and even outperformed it on ESC-50 (with only 40 samples per class) while using less than 5% of trainable parameters. The evaluation provides compelling evidence for the effectiveness of a parameter efficient strategy. Therefore, we adopted the Convolutional Adapter for further investigation in TI-CL.

4.3. Adapter Incremental Continual Learning of AST

Formulation. The TI-CL setup consists of three tasks: SCv2, ESC-50, and AVE, which are performed in a sequential order.

Table 1: *Computational efficiency of the proposed FTA. k denotes the factor of GSA computations required by FTA.*

Dataset	Duration	Spectrogram Shape	Freq (M Tokens)	Time (T Tokens)	$\mathcal{O}_{GSA}/d$	$\mathcal{O}_{FTA}/d$	k
SCv2	1s	[128,101]	12	9	11881	2377	0.2
ESC-50	5s	[128,501]	12	49	346921	36457	0.105
AVE	10s	[128,1006]	12	100	1442401	135601	0.094

Table 2: *Evaluation of PET methods for AST.*

Method	**Params (Million)**	**Accuracy (%)**	
		ESC-50	**SCv2**
Linear	0.26	71.05	81.44
LayerNorm Tune [21]	0.27	72.75	89.2
BitFit [20]	0.32	72	87.91
AdaptFormer [18]	1.43	83	92.3
Prompt Tuning [22]	2.17	78.85	91.64
LoRA [17]	2.6	79.05	92.14
Houlsby [16]	2.62	69.75	90.83
ConvPass [19]	**3.5**	**83.3**	**93.42**
Full Fine Tuning	**86.33**	**82.3**	**94.58**

Table 3: *Comparison of parameter and storage cost for three training modes in TI-CL setup.*

	Trainable Params	Total Params	Storage
Model Seq.	86.5M	86.62M	348MB
Model Inc.	86.5M	259.63M	1.02GB
Adapter Inc.	3.5M	96.6M	47MB

In each task of the TI-CL, only the corresponding dataset is available for training and the datasets from previous tasks are no longer available. Only the test data of previous tasks are used to evaluate the model performance after training on current task.

Training Modes. To demonstrate the proposed approach's advantages, we trained the AST model in three different modes, following the sequential training order. These modes are:

- Model Sequential: The same AST model is trained repeatedly on new tasks.
- Model Incremental: For every new task, a new AST model is trained independently.
- Adapter Incremental: The proposed approach described in Section 3, where new adapter modules are added to the frozen backbone with FTA for new tasks.

The first two modes rely on GSA and the ESC-50 task is evaluated with single fold.

Performance vs Parameter-Efficieny. Figure 3 displays the performance of the AST model for three training modes. In the Model Sequential setting, catastrophic forgetting occurred, where the model weights optimized for a new task forgot the knowledge gained from previous tasks, leading to a significant performance drop. However, Model Incremental setting trained the models independently for each task, thereby resolving this issue. The proposed Adapter Incremental method also addressed the catastrophic forgetting issue by training independent task-specific adapter modules and showed competitive performance on all three tasks. However, the Model Sequential and Model Incremental settings were less efficient than the Adapter Incremental method in terms of total model parameters and trainable parameters, as illustrated in Table 3. Note that the total number of parameters were those required for inference upon the completion of sequential training on three tasks. The Model Incremental setting had a large number of total parameters, and both Model Sequential and Model Incremental settings required nearly 25 times more trainable parameters than Adapter Incremental. Overall, the proposed Adapter Incremental method for TI-CL combined the best of performance and parameter efficiency, delivering stable performance and minimizing the number of trainable parameters. Also, Adapter Incremental setting has substantially lower storage cost because only the adapter weights need to be saved, unlike the other two setting which stores the weights of the entire models(s).

Table 4: *Performance of FTA vs GSA on three tasks.*

Method	Accuracy (%)		
	SCv2	ESC-50	AVE (Audio)
GSA	93.57	85.25	69.1
FTA	92.81	83	66.42

4.4. Impact of FTA

We conducted a study to compare the computational efficiency and performance of our proposed FTA with Global Self-Attention (GSA) on three datasets, each with varying maximum audio durations. In Table 1, we summarize the details of our comparison. Our results showed that FTA required significantly fewer computations than GSA, especially with longer audio durations (the larger spectrograms). To further evaluate the performance of FTA and GSA, we implemented both methods using the Convolutional Adapter model and measured their audio classification accuracies on the three datasets. The results are presented in Table 4. We found that FTA performed competitively with GSA in terms of accuracy, but with only a fraction of the computational resources required by self-attention. Overall, our study demonstrates that FTA is a promising approach for audio classification tasks, as it achieves comparable accuracy to GSA while using significantly fewer computational resources.

5. Conclusions

In this work, we proposed a new method called Adapter Incremental Continual Learning (AI-CL) for audio classification in the context of Task Incremental Continual Learning (TI-CL) of AST audio classifiers. AI-CL improved parameter efficiency with the introduction of Convolutional Adapters for AST. To enhance compute efficiency for longer audio streams, we proposed a new method called Frequency-Time factorized Attention. Our experiments have shown that AI-CL is both parameter-efficient and compute-efficient. AI-CL enables continual learning with minimal resources, which can be scaled effectively for a large number of tasks.

6. Acknowledgements

This work was carried out at the Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore. The research is supported by the DSO National Laboratories, under the project agreement No. DSOCL21238.

7. References

[1] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[2] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[4] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International conference on machine learning*. PMLR, 2017, pp. 3987–3995.

[5] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[6] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

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

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

[9] 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] A. Cossu, A. Carta, and D. Bacciu, "Continual learning with gated incremental memories for sequential data processing," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[12] H. Ma, J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Wang, "Continual Learning for Fake Audio Detection," in *Proc. Interspeech 2021*, 2021, pp. 886–890.

[13] S. Karam, S.-J. Ruan, and Q. M. ul Haq, "Task incremental learning with static memory for audio classification without catastrophic interference," *IEEE Consumer Electronics Magazine*, vol. 11, no. 5, pp. 101–108, 2022.

[14] Y. Wang, N. J. Bryan, M. Cartwright, J. P. Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 321–325.

[15] Z. Ke, B. Liu, N. Ma, H. Xu, and L. Shu, "Achieving forgetting prevention and knowledge transfer in continual learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 443–22 456, 2021.

[16] 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.

[17] 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 *International Conference on Learning Representations*, 2022.

[18] S. Chen, G. Chongjian, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," in *Advances in Neural Information Processing Systems*, 2022.

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

[20] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[21] K. Kim, M. Laskin, I. Mordatch, and D. Pathak, "How to adapt your large-scale vision-and-language model," 2021.

[22] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 709–727.

[23] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[25] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[26] Y. L. Tan, A. W.-K. Kong, and J.-J. Kim, "Pure transformer with integrated experts for scene text recognition," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*. Springer, 2022, pp. 481–497.

[27] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 247–263.