



# A Parameter-Efficient Learning Approach to Arabic Dialect Identification with Pre-Trained General-Purpose Speech Model

Srijith Radhakrishnan<sup>1,2,4</sup>, Chao-Han Huck Yang<sup>1,3</sup>, Sumeer Ahmad Khan<sup>1,4</sup>  
Narsis A. Kiani<sup>1</sup>, David Gomez-Cabrero<sup>1</sup>, Jesper N. Tegner<sup>1,4</sup>

<sup>1</sup>King Abdullah University of Science and Technology, Saudi Arabia

<sup>2</sup>Manipal Institute of Technology, India; <sup>3</sup>Georgia Institute of Technology, USA; <sup>4</sup>SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence, Thuwal 23952, Saudi Arabia

{srijith.radhakrishnan, sumeer.khan, jesper.tegner}@kaust.edu.sa, huckiyang@gatech.edu

## Abstract

In this work, we explore Parameter-Efficient-Learning (PEL) techniques to repurpose a General-Purpose-Speech (GSM) model for Arabic dialect identification (ADI). Specifically, we investigate different setups to incorporate trainable features into a multi-layer encoder-decoder GSM formulation under frozen pre-trained settings. Our architecture includes residual adapter and model reprogramming (input-prompting). We design a token-level label mapping to condition the GSM for Arabic Dialect Identification (ADI). We achieve new state-of-the-art accuracy on the ADI-17 dataset by vanilla fine-tuning. We further reduce the training budgets with the PEL method, which performs within 1.86% accuracy to fine-tuning using only 2.5% of (extra) network trainable parameters. Our study demonstrates how to identify Arabic dialects using a small dataset and limited computation with open source code at <https://github.com/Srijith-rkr/KAUST-Whisper-Adapter>

**Index Terms:** Parameter-Efficient Learning, Dialect Identification, Arabic Dialect

## 1. Introduction

Dialect identification [1, 2] (DI) amounts to identifying similar dialects belonging to the same language family. It is a specific case of language identification [3] (LID) task. However, DI is more challenging than LID owing to the fact that dialects share similar acoustic and linguistic characteristics compared to different languages. Very minute differences in pronunciation [4] and accent are used as cues to identify dialects. Moreover, DI does not share the advantage of publicly available speech recognition models pre-trained on large speech data corpora for network initialization. Despite these challenges, DI remains relatively unexplored compared to LID.

In this study, we leverage upon a recent open-access and general-purpose speech recognition architecture, Whisper [5], pre-trained on a large speech corpus from OpenAI, to address DI in resource-constrained and data-limited conditions. We use Parameter-Efficient Learning [6, 7] (PEL) to adapt a large pre-trained model by training small additive modules embedded into the frozen pre-trained model. By doing so, we require less training time and computing resources to fine-tune the model for DI. Figure 1 is a schematic of the proposed parameter-efficient learning framework.

We choose to perform DI in Arabic owing to its substantial regional variations and the widespread use of Arabic as an official language in over 22 countries [8]. Notably, significant differences exist between the standard written form, referred to as Modern Standard Arabic, and the local colloquial dialects spoken in each region. Interestingly, not all dialects are

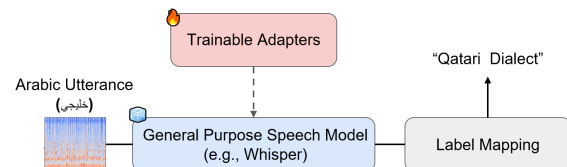


Figure 1: Overview of proposed parameter-efficient learning framework for Arabic dialect identification building upon parameter-efficient learning [7] and label mapping [10].

mutually intelligible.

In this study, we present several contributions: (1) Firstly, we introduce the novel use of Parameter-Efficient-Learning (PEL) for this task, marking the first application of this approach to Arabic dialect identification [8]. (2) We investigate *different designs* to incorporate trainable features into a multi-layer encoder-decoder frozen model. (3) We achieve new state-of-the-art accuracy on the **official** testing and development sets of ADI-17 [9] dataset using only 30.95% of the training data. (4) Lastly, we demonstrate that our PEL method achieves equivalent performance to full fine-tuning using only **2.5%** of (extra) network parameters.

## 2. Related work

### 2.1. Existing works on Dialect prediction

Several works exist in applying Natural Language Processing to the Arabic text due to sufficient open-source Arabic textual data from multiple sources, such as Newspaper articles [11], and Twitter [12]. Unfortunately, minimal open-source Arabic speech data is available, as reported in [13]. Most existing Arabic dialect identification methods use machine learning and deep learning methods. [14] used phonetic and lexical features obtained from a speech recognition system combined with a multi-class support vector machine [15] to identify Arabic dialects. [16] used a Siamese neural network along with i-vector post-processing to learn similarities and dissimilarities among Arabic dialects.

### 2.2. Parameter-Efficient Learning

Large pre-trained models have been very successful at various tasks in natural language processing. Parameter-efficient learning is a research direction that aims to reduce the computational cost by adapting large pre-trained models to downstream tasks by updating only a subset of (extra) parameters. In this section, we introduce several state-of-the-art parameter-efficient learn-

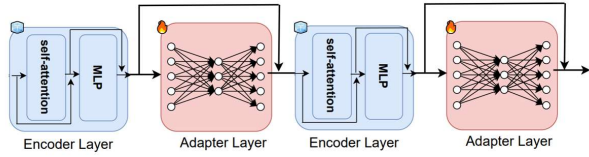


Figure 2: Illustration of the transformer architecture embedded with adapter layers.

ing methods.

**Residual Adapters** : Residual Adapters are small trainable blocks inserted between the layers of a transformer architecture [17, 18, 19, 20]. They down-project the latent dimension from the previous layer and apply a nonlinear activation function, followed by an up-projection. A residual connection surrounds the adapter layer. This setup encourages parameter sharing between the frozen components and localizes all the weight updates to the adapter modules.

**Neural Reprogramming**: Neural reprogramming can be used to repurpose a frozen pre-trained model to out-of-domain prediction tasks by adding trainable parameters to the input of the pre-trained model [10, 21]. The frozen model is followed by a label-mapping strategy to map the source labels to the out-of-domain target labels. The trainable input noise aligns the latent distribution of the target domain with being more similar to that of the source domain, using the pre-existing decision boundaries of the frozen model. Neural reprogramming works well when the input size of the target data is comparably smaller than the source data, as demonstrated in [10, 22, 6].

**BitFit**: BitFit refers to Bias-Term Fine-Tuning, which only updates the bias-terms and the task-specific classification layer of the frozen pre-trained model [23].

**Others**: Other state-of-the-art parameter-efficient learning techniques include LoRa [24], in which trainable low-rank matrices are embedded inside the transformer attention layers to approximate parameter updates and speech prompting methods [25, 26] as one approach shared similar motivation to input reprogramming, such trainable input and label mapping.

### 3. Method

We evaluate two fine-tuning strategies. First vanilla fine-tuning, in which we fine-tune different components of the network parameters. Next, parameter-efficient fine-tuning, in which we train (extra) modules added to the network architecture and investigate their performance.

#### 3.1. Vanilla Fine-tuning

To incorporate the new dialect classes into the network architecture, we append the new classes to the pre-existent multilingual tokenizer and modify the token embedding matrix  $W_e \in \mathbb{R}^{t \times n}$  to  $W'_e \in \mathbb{R}^{(t+d) \times n}$ . Here  $t$  and  $n$  are the dimensions of the tokenizer and network, respectively, and  $d$  denotes the number of dialects. The weights of  $W_e$  are then copied with random padding to  $W'_e$  for initialization. A cross-entropy loss is used to fine-tune different components of the network.

#### 3.2. Input Reprogramming

We begin our experiments in parameter-efficient fine-tuning with Input reprogramming. If  $x$  is the input to the pre-trained

frozen model  $W_\Theta(\cdot)$  parameterized by theta, the frozen model predicts the dialect  $\hat{y}$  as  $W_\Theta(x) \rightarrow \hat{y}$ . Input reprogramming aims to add trainable noise  $x_t$  to the input  $x$ . In our application, the input to the frozen pre-trained model is the Log-Mel Spectrogram computed from 30 seconds of Arabic speech. We add trainable parameters of the same dimensions as  $x$  to minimize the prediction error between  $\hat{y}$  and the true label while not updating  $\Theta$ , the parameters of the model as  $W_\Theta(x + x_t) \rightarrow \hat{y}$

#### 3.3. Latent Space Efficient Learning

We insert small trainable modules called adapters between the encoder layers of our model as shown in Fig 2. These adapter layers contain a linear down projection of the latent input dimensions  $n$  to a bottleneck dimension  $b$  using  $W_{dp} \in \mathbb{R}^{n \times b}$ . We apply  $g(\cdot)$ , the GELU [27] activation function on the bottleneck dimension followed by an up projection with  $W_{up} \in \mathbb{R}^{b \times n}$ . Residual connections are applied around the adapter layer as represented in Equation 1. We compare the performance of multiple bottleneck dimensions  $b$ , set to  $\frac{n}{2}$ ,  $\frac{n}{4}$ , and  $\frac{n}{8}$  against multiple sample sizes in our experiments.

$$x \leftarrow x + g(x.W_{dp}) W_{up} \quad (1)$$

#### 3.4. Token Mapping in Whisper

We utilize many-to-one hard-label mappings to map the source label of the model to our target dialect classes, motivated by [10]. Specifically, we randomly assign unique language tokens of the model to each dialect, sum over their logits, and apply the softmax function to calculate respective dialect probabilities. Performance significantly depreciated when we attempted to implement a trainable label mapping setup.

## 4. Experimental Setup

In this section, we describe our experimental setup and present results for Arabic dialect identification on the ADI-17 dataset.

#### 4.1. General Purpose Speech Model

We use Whisper<sub>Base</sub> [5] as the underlying pretrained model composed of an encoder-decoder architecture for our experiments. A multi-task general purpose speech model trained on 680,000 hours of multilingual data from 99 languages ensuring rich representational knowledge. The input to the model is the log magnitude Mel spectrogram representation computed from 30 seconds of an audio clip. We pad the audio clip with zeros if the duration is less than 30 seconds. More details can be found at [5]. For training we use Adam optimizer for all tasks. We explore learning rates of  $1e-2$ ,  $1e-3$ ,  $1e-4$ , and utilize the optimal learning rate with a linear learning rate scheduler to train for 50 epochs until convergence. We trained our models using 4 V100 GPUs with an adequate batch size of 64 and 0.1 weight decay. Our implementation and pre-trained weights are open source at <https://github.com/Srijith-rkr/KAUST-Whisper-Adapter>.

#### 4.2. Dataset

We evaluate our findings on the ADI-17 dataset [9] released as part of the MGB5 challenge. The dataset contains audio data for 17 Arabic dialects collected from YouTube. The training set contains 3033 hours of audio data. The dev and test set contain 58 hours of audio data. The training set of ADI17 is unbalanced in terms of the number of utterances and hours of data. For ex-

Table 1: An overview of the model performance of fine-tuning and parameter-efficient learning on utterances over 20 seconds as in standard ADI-17 setup. The development and test accuracy were evaluated in the official setup and outperformed three previous state-of-the-art results [9, 28, 29] using its official dev and test settings. Noted that directly using frozen pre-trained Whisper (second row) with label mapping performs 2.52% to 4.44% in the dialect identification task, which motivates its demand of efficient model designs.

Method	Trainable Para. ( $\downarrow$ )	Trainable Ratio (%)	Dev. Acc. ( $\uparrow$ )	Test Acc. ( $\uparrow$ )	Utility Score( $\uparrow$ )
Frozen Whisper <sub>Base</sub> [5]	0	0%	2.52%	4.44%	-
Full fine-tuning	71.8M	100%	95.55%	93.34%	11.88
Encoder fine-tuning	18.9M	26.32%	<b>96.39%</b>	<b>95.01%</b>	13.06
Decoder fine-tuning	52M	72.42%	95.07%	93.75%	12.15
BitFit	75.8K	0.10%	59.01%	57.68%	11.82
Encoder BitFit	32.3K	0.04%	44.59%	41.83%	9.28
Decoder BitFit	43.5K	0.06%	36.78%	39.42%	8.50
Input Reprogramming	240K	0.33%	27.04%	27.91%	5.19
Adapters-64	642K	0.89%	92.79%	89.47%	<b>15.41</b>
Adapters-128	1M	1.39%	93.99%	91.50%	15.25
Adapters-256	1.8M	2.50%	<b>95.55%</b>	<b>93.15%</b>	14.89
ADI-17 [9]	13.1M	-	93.70%	90.4%	13.17
CNN+Transformer [29]	64.3M	-	94.04%	93.06	11.92
Supervised clustering [28]	>50M	-	95.09%	94.35%	~12.25

Table 2: Dataset sample statistics

#samples per class	#hours per class	fraction of ADI-17
0.5k	4.1	2.33%
1k	8.2	4.60%
2k	15.5	8.69%
5k	32.8	18.40%
10k	53.1	30.95%

ample, the Iraq (IRQ) dialect has 815.8 hours of training data, whereas the Jordan (JOR) dialect contains only 25.9 hours of training data. The audio clips have varying lengths, between 1 second to 26 minutes. To deal with class imbalance during training, we over-sample random windows of 30 seconds of audio data from clips from the minority classes containing more than 20 seconds in duration. The dataset is further divided into three subcategories in terms of duration, Short duration (< 5s), medium duration (5-20s), and long duration (> 20s). We perform our experiments using the long duration (> 20s) subsection of the dataset since we only use a fraction of the dataset to fine-tune different transfer learning methods to evaluate the effect of data on performance. The sample sizes and their duration are found in Table 2.

### 4.3. Baseline

The baseline reported by ADI-17 [9] and the performance of other methods on ADI-17 with utterances over 20 seconds have been reported in Table 1. ADI-17 [9] utilized a convolutional neural network-based (CNN) architecture with a softmax output. The [29] used a fusion of transformer-based architecture and CNN with downsampling. [28] used a supervised clustering-based algorithm with triplet loss. The results in the table show that all three methods perform well in terms of accuracy. However, there are notable differences in the number of parameters used by each method, with ADI-17 [9] utilizing the fewest parameters at 13.1M and Supervised clustering [28] requiring over 50M parameters.

### 4.4. Performance Studies

Whisper<sub>Base</sub> is an encoder-decoder-based transformer [30] architecture with a total trainable parameter of 70M. We

fine-tuned the encoder and decoder separately to examine the impact of the architecture on DI. Similarly, we evaluate the same setup for bias-only fine-tuning. The performance of fine-tuning with 10,000 samples and the number of fine-tuned parameters is reported in Table 1. Fine-tuning only the encoder achieves new state-of-the-art accuracy of 95.01% on ADI-17 while using only 30.95% of the training set. Fine-tuning the decoder alone yields comparable results to fine-tuning the entire model, but fine-tuning only the encoder achieves better results. However, fine-tuning the encoder bias terms performs significantly better than fine-tuning decoder bias terms. Input reprogramming does not perform well compared to bias-only fine-tuning, although they share a similar number of parameters.

Whisper<sub>Base</sub> has 512 latent dimensions from its transformer encoder. Thus, we train our residual adapter layers with bottleneck dimensions of 64, 128, and 256 to examine the impact of the bottleneck dimensions on performance. We also tested (1) adapters with bridge connections, (2) adapters with dense connections [31], (3) adapters without residual connections, (4) adapters with self-attention backbone, (4a) with Gelu activation, (4b) with residual connections. We observed that the setup in Section 3.3 had the best performance. The adapter approach with 256 latent dimensions has the same performance as fully fine-tuning the model, despite it updating only 2.5% of the network parameters. This is advantageous since a separate copy of all model parameters does not have to be created for each downstream task of Whisper. Instead, only the adapter weights need to be stored for each downstream task.

The performance comparison of fine-tuning methods on the development set against the number of samples per class is displayed in Figure 3. Here we observe that although encoder fine-tuning, decoder fine-tuning, and full fine-tuning achieve similar performance against 10,000 samples per class, encoder fine-tuning performs significantly better with a lower number of samples. We conjecture that this is because only the encoder of Whisper<sub>Base</sub> processes the audio input. The decoder receives the output of the encoder. A similar trend is observed with bias-only fine-tuning. Furthermore, the adapter methods with 64, 128, and 256 latent dimensions display comparable performance, with minor improvements in performance as the bottle-neck dimensions increase. Notably, the adapter methods

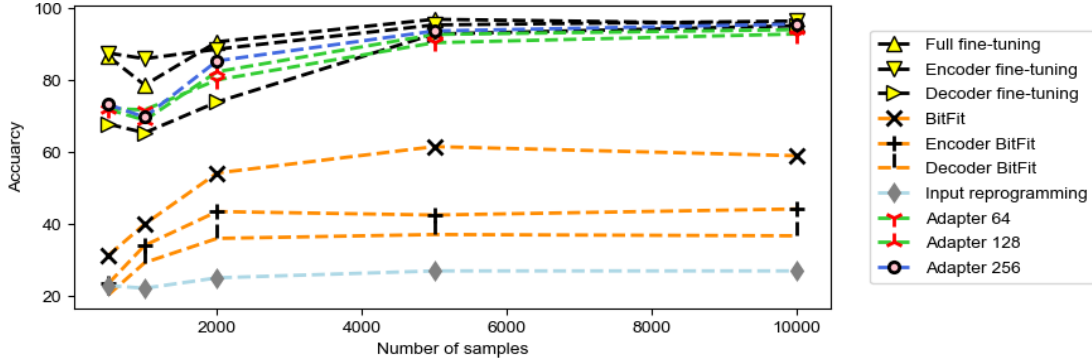


Figure 3: Performance comparison of fine-tuning methods using the development set against the number of samples per class.

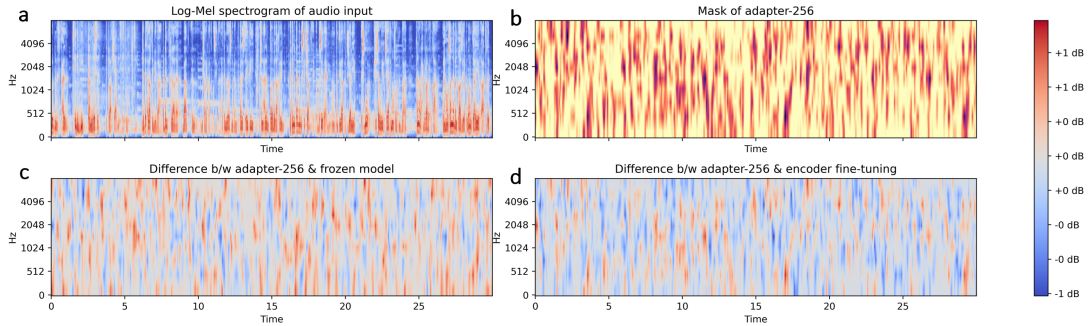


Figure 4: (a) Log-Mel spectrogram of the audio input, (b) Mask of the adapter-256 model, (c) Difference between the masks of adapter-256 and frozen model and (d) Difference between the masks of adapter-256 and fine-tuned encoder model.

outperform decoder fine-tuning with fewer samples.

#### 4.5. Utility Discussion and Efficient Module Selection

In addition to accuracy, we also report Utility scores as defined in Equation (2) to compare the efficiency against performance. The results of this analysis are presented in Table 1.

$$Utility\ score(i) = \frac{Acc.test(i)}{\log(Nof\ trainable\ parameters)} \quad (2)$$

The results in Table 1 show that adapters demonstrate the highest Utility scores among all methods, as they perform well with a smaller number of trainable parameters. Specifically, adapter-64 (ninth row) achieves the optimal performance versus efficiency tradeoff, indicating that this method may be the most efficient approach for the task.

#### 4.6. Neural Saliency Analysis on Acoustic Features

As a preliminary study investigating PEL learning experiments for Arabic dialect identification tasks, we aim to provide attribute-based interpretation based on frozen pre-trained Whisper model. Neural saliency methods [32, 33] provide interpretable intuitions behind black-box networks by analyzing the weight distribution over hidden neurons. In this work, we employ a perturbation-based saliency map [32] for feature-level interpretations. This algorithm masks parts of the input to determine regions responsible for the classifier decision to understand better the patterns the model uses to predict Arabic di-

alects. The results of our analysis are presented in Fig 4. Fig 4a represents the Log-Mel spectrogram of 30 seconds of Qatari dialectal speech. The mask of adapter-256 in Fig 4b emphasizes the parts of the input the model focuses on to make predictions. These regions are highlighted in red color. The difference between the masks of the adapter, frozen model, and fine-tuned encoder model have been plotted in Fig 4c and 4d. The bluish plot in Fig 4d compared to Fig 4c indicates that the adapter-256 and fine-tuned encoder model observe similar input regions compared to the adapter-256 and frozen Whisper.

## 5. Conclusion

This paper presents parameter-efficient approach for Arabic dialect identification, as one very first study on this low-resource application. The proposed method utilizes frozen pre-trained speech recognition models by incorporating trainable parameters at the input and latent space levels. By using only 30.95% of the ADI-17 training data and 2.5% of the model parameters, the proposed method achieves state-of-the-art accuracy on the ADI-17 benchmark, comparable to the performance of full fine-tuning. These results showcase the effectiveness of our efficient learning approach. Further research on this approach may enable dialect prediction and other tasks for low-resource and long-tail data [34, 35], which deserves more studies.

## 6. Acknowledgements

We gratefully acknowledge SDAIA-KAUST for supporting this work.

## 7. References

- [1] M. A. Zissman, T. P. Gleason, D. M. Rekart, and B. L. Losiewicz, "Automatic dialect identification of extemporaneous conversational, latin american spanish speech," in *Proc. of ICASSP*, vol. 2. IEEE, 1996, pp. 777–780.
- [2] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.
- [3] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proc. ASRU*. IEEE, 2017, pp. 265–271.
- [4] H. Al-Haj, R. Hsiao, I. Lane, A. W. Black, and A. Waibel, "Pronunciation modeling for dialectal arabic speech recognition," in *2009 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2009, pp. 525–528.
- [5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.
- [6] C.-H. H. Yang, B. Li, Y. Zhang, N. Chen, R. Prabhavalkar, T. N. Sainath, and T. Strohmman, "From English to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition," *Proc. of ICASSP*, 2023.
- [7] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *International Conference on Learning Representations*, 2022.
- [8] O. F. Zaidan and C. Callison-Burch, "Arabic dialect identification," *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, 2014.
- [9] S. Shon, A. Ali, Y. Samih, H. Mubarak, and J. Glass, "Adi17: A fine-grained arabic dialect identification dataset," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8244–8248.
- [10] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2series: Reprogramming acoustic models for time series classification," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 808–11 819.
- [11] I. A. El-khair, "1.5 billion words arabic corpus," 2016.
- [12] W. Zaghouani and A. Charfi, "Arap-tweet: A large multi-dialect Twitter corpus for gender, age and language variety identification," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.
- [13] A. Ahmed, N. Ali, M. Alzubaidi, W. Zaghouani, A. A. Abdalrazaq, and M. Househ, "Freely available arabic corpora: A scoping review," *Computer Methods and Programs in Biomedicine Update*, vol. 2, p. 100049, 2022.
- [14] A. Ali, N. Dehak, P. Cardinal, S. Khurana, S. H. Yella, J. Glass, P. Bell, and S. Renals, "Automatic Dialect Detection in Arabic Broadcast Speech," in *Proc. Interspeech 2016*, 2016, pp. 2934–2938.
- [15] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.
- [16] S. Shon, A. Ali, and J. Glass, "Mit-qcri arabic dialect identification system for the 2017 multi-genre broadcast challenge," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 374–380.
- [17] N. Hounsby, A. Giurgiu, S. Jastrzebski *et al.*, "Parameter-efficient transfer learning for NLP," in *Proc. of ICML*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799.
- [18] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [19] K. Tomanek, V. Zayats *et al.*, "Residual adapters for parameter-efficient ASR adaptation to atypical and accented speech," in *Proc. of EMNLP*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6751–6760.
- [20] Y.-N. Hung, C.-H. H. Yang, P.-Y. Chen, and A. Lerch, "Low-resource music genre classification with cross-modal neural model reprogramming," in *Proc. of ICASSP*. IEEE, 2023.
- [21] H. Yen, P.-J. Ku, C.-H. H. Yang, H. Hu, S. M. Siniscalchi, P.-Y. Chen, and Y. Tsao, "Neural model reprogramming with similarity based mapping for low-resource spoken command classification," *arXiv preprint arXiv:2110.03894*, 2021.
- [22] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," 2018.
- [23] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022.
- [24] 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.
- [25] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H.-y. Lee, "An exploration of prompt tuning on generative spoken language model for speech processing tasks," *Proc. of Interspeech*, 2022.
- [26] 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.
- [27] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
- [28] Z. Alhakeem, Y. Kwon, and H.-G. Kang, "Disentangled representations for arabic dialect identification based on supervised clustering with triplet loss," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 526–530.
- [29] W. Lin, M. Madhavi, R. K. Das, and H. Li, "Transformer-based arabic dialect identification," in *2020 International Conference on Asian Language Processing (IALP)*. IEEE, 2020, pp. 192–196.
- [30] 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.
- [31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.
- [32] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3429–3437.
- [33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.
- [34] T. Reitmaier, E. Wallington, D. Kalarikalayil Raju *et al.*, "Opportunities and challenges of automatic speech recognition systems for low-resource language speakers," in *Proc. of CHI*, ser. CHI '22. New York, NY, USA: ACM, 2022.
- [35] B. Labrador, G. Zhao, I. L. Moreno, A. S. Scarpati, L. Fowl, and Q. Wang, "Exploring sequence-to-sequence transformer-transducer models for keyword spotting," in *Proc. of ICASSP*. IEEE, 2023, pp. 1–5.