Domain Adaptation approaches for end2end ASR models

Kwok Chin Yuen

School of Computer Science & Engineering

A thesis submitted to the Nanyang Technological University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

Oct. 2023

.....

Date

.....

Kwok Chin Yuen

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

Oct. 2023

.....

Date

.....

Prof. Chng Eng Siong XXX

Authorship Attribution Statement

Please select one of the following; *delete as appropriate:

*(B) This thesis contains material from 1 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Please amend the typical statements below to suit your circumstances if (B) is selected.

Chapter 4.2 is published as Kwok Chin Yuen, Li Haoyang and Eng-Siong Chng. ASR Model Adaptation for Rare Words Using Synthetic Data Generated by Multiple Text-To-Speech Systems. 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference.

The contributions of the co-authors are as follows:

- A/Prof Eng-Siong Chng provided the initial project direction and edited the manuscript drafts.
- Haoyang and I prepared the manuscript drafts. The manuscript was revised by A/Prof Eng-Siong Chng.
- I designed the study and performed the laboratory work on ASR adaptation at the School of Computer Science and Engineering.
- Haoyang performed the laboratory work on TTS-synthesized data at the School of Computer Science and Engineering. He also analyzed the data.

Oct. 2023

.....

Date

.....

Kwok Chin Yuen

Acknowledgements

I wish to express my greatest gratitude to my advisor Chng Eng Siong, for his invaluable guidance throughout the process of writing my thesis. His profound insights and constructive feedback have been the bedrock upon which my academic growth and confidence were built during this journey.

Abstract

End-to-end Automatic Speech Recognition (ASR) technology utilizes deep learning models to convert spoken human language into written text. ASR has diverse applications, ranging from supporting Voice Assistance and Voice Commands to the intricate task of preserving meeting records and accurately transcribing complex medical documents. To enhance the quality of ASR services in these domains, extensive endeavors have been dedicated to improving transcription accuracy and minimizing latency.

Nonetheless, a significant challenge within the realm of ASR lies in its limited ability to effectively adapt to unfamiliar audio domains. To illustrate, an ASR model fine-tuned for transcribing American English accents may demonstrate exceptional performance for this particular accent. However, its transcription accuracy may substantially deteriorate when faced with different accents, such as British English. This phenomenon, referred to as the out-of-domain problem, has considerably constrained the practical utility of ASR models.

In order to address this issue, ASR adaptation techniques are employed to enhance the performance of an ASR model when dealing with unfamiliar domains. Nevertheless, ASR adaptation presents notable challenges, primarily because it is often conducted within constrained contexts. These constraints typically encompass: 1) the limitation of low-resource data, involving adaptation on a small size dataset which may lead to overfitting and catastrophic forgetting, and 2) the constraint of having access only to text data, rather than paired audio-text data for adaptation purposes. This thesis will center its attention on two specific approaches for mitigating these challenges: layerwise adaptation and synthesized text-to-speech (TTS) data adaptation.

In the context of layer-wise adaptation, its objective is to address the challenges of overfitting and catastrophic forgetting encountered within the constraints of low resouce data. This approach tackles these issues by training a model on a subset

of its layers while keeping the layers susceptible to overfitting and catastrophic forgetting fixed. Previous research determined the optimal subset of layers for training through heuristic methods and trial-and-error, often resorting to grid-search techniques to search for various layer combinations. However, this search process becomes prohibitively expensive and impractical when computational resources are constrained.

In the context of TTS data synthesized adaptation, its primary objective is to facilitate model adaptation when dealing with scenarios where only text data is accessible. Conventionally, models are trained using paired audio-text data within a supervised learning framework. To enable model adaptation using solely text data, TTS data synthesized adaptation harnesses a Text-to-Speech system to generate synthetic audio from the existing real text data, thus creating artificial audio-text pairs for model training. However, prior research in TTS data adaptation has predominantly relied on a single Text-to-Speech system for generating synthetic audio, which has restricted the diversity of the synthesized data.

To address the challenge of conducting a costly search for layers to be frozen during layerwise adaptation, this thesis introduces an automated approach for selecting the layers to be frozen. Specifically, when dealing with a residual network, a performance metric will be calculated using each layer's skip connection, and the decision to freeze layers will be based on the validation performance of each individual layer. Experimental results demonstrate that when this method is applied to adapt a LibriSpeech-pretrained Conformer model to a 10 hours rare word dataset which contains numerous road names, it can efficiently identify the optimal set of layers to freeze. Furthermore, it outperforms heuristic-based selection strategies, resulting in a relative reduction of 6.2% in word error rate.

In response to the challenge of limited diversity in synthesized data from a single Text-to-Speech (TTS) system in TTS data synthesized adaptation, this thesis explores harnessing multiple TTS systems to generate synthetic audio, thereby creating a variety of artificial audio-text pairs for model training. This strategy aims to enhance the diversity of synthesized audio and mitigate the risk of the ASR model becoming overly biased towards a specific TTS output distribution. Experimental results demonstrate that when this method is applied to adapt a Librispeech pretrained Conformer model to the same 10 hours rare word dataset,

it leads to a significant improvement in transcription accuracy, resulting in a relative reduction of 9.8% in word error rate compared to using data synthesized from a single TTS system.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\mathcal{H}$	the Euclidean space
$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
$\|\cdot\|_G$	the induced norm of a vector in G-space
$\|\cdot\|_E$	the induced norm of a vector or matrix in probabilistic space
$\odot$	the Hadamard (component-wise) product
$\otimes$	the Kronecker product
$\langle\cdot,\cdot\rangle$	the inner product of two vectors
$\circ$	the composition of functions

T	the audio length
x	the audio signal in a sequence of vectors
h	the high-level representation of audio in a sequence of vectors
S	the transcript sequence length
Q	the sequence of query vectors
K	the sequence of key vectors
V	the sequence of value vectors

Acronyms

E2E	end-to-end
ASR	Automatic Speech Recognition
GAN	Generative Adversarial Network
MTR	Multi-style Training

TTS	Text-to-Speech
DNN	Deep Neural Network
FFN	Feed Forward Network
AM	Acoustic Model
LM	Language Model
LAS	Listen, Attend and Spell
CLAS	Contextual Listen, Attend and Spell
AED	Attention-Encoder-Decoder
RNN	Recurrent Neural Network
HMM	Hidden Markov Models
GMM	Gaussian Mixture Models
CTC	Connectionist temporal classification
RNN-T	Recurrent Neural Network Transducer
RNA	Recurrent Neural Aligner
HAT	Hybrid Auto-regressive Transducer
BLSTM	Bidirectional Long Short-Term Memory
AR	Attention Autoregressive
FHL	Factorized Hidden Layer
SVD	Singular Valud Decomposition
LRPD	Low-rank Plus Diagonal
TTS	Text-to-Encoder
MLLR	Maximum Likelihood Linear Regression
fDLR	feature Discriminative Linear Regression
oDLR	output-feature Discriminative Linear Regressions
LDA	Linear Discriminant Analysis
HLDA	Heteroscedastic Discriminant Analysis
VTLN	Vocal Tract Length Normalization
ASA	Adversarial Speaker Adaptation
SD	Speaker Dependent
SI	Speaker Independent
KLD	Kullback-Leibler Divergence
SPK	Speaker
WER	Word Error Rate

CER	Character Error Rate
IPA	International Phonetic Alphabet

Chapter 1

Introduction

1.1 Background

End-to-end Automatic Speech Recognition (ASR) is a technology that focuses on transcribing human speech mostly using deep learning models. These models are designed to take in speech audio as input and produce the corresponding text transcription as output. ASR holds immense significance due to its diverse range of applications, with the potential to exert a substantial impact on various facets of our daily lives.

For instance, ASR technology plays a pivotal role in enhancing communication by enabling voice commands for smart devices [7], facilitating speech-to-text transcription services [8], and providing real-time translation capabilities [9].

To further enhance the quality and performance of ASR services, extensive research efforts have been directed towards improving transcription accuracy and reducing latency, thereby ensuring more effective and responsive ASR systems.

1.1.1 Deep learning and Automatic Speech Recognition

In recent decades, there has been a growing interest in harnessing deep learning technologies to enhance the transcription accuracy of Automatic Speech Recognition (ASR) systems [5, 10, 11]. This approach is characterized by several key features. Firstly, it uses neural networks with many layers [12], which enables

these networks to autonomously extract relevant audio features directly from raw audio inputs. Additionally, the training process may involve iteratively adjusting the model to minimize its transcription errors on the training data via a technique called back-propagation [13]. Thirdly, deep learning in ASR may rely on the utilization of large datasets for better model performance [14]. As a result of advancements in deep learning, ASR models employing deep learning frameworks have achieved state-of-the-art performance across numerous ASR tasks [15, 16].

1.1.2 The out-of-domain problem in ASR

Despite the advancement in transcription accuracy, a major problem of ASR is its lack of generalization to transcript accurately on unseen audio domains. For example, an ASR model optimized to transcribe American accent English can only transcribe this specific kind of accent well, and its transcription accuracy may degrade significantly on other accents [17]. This is known as the out-of-domain problem [18] and has greatly limited the usefulness of an ASR model.

To solve the problem, ASR adaptation [19–22] is used to improve an ASR model's performance on target domains. This is done by further training the model with data within the target domain. After adaptation, the model has learned target domain specific knowledge, hence its performance in the domain can be improved.

1.2 Motivation

However, ASR adaptation are usually subject to different resource constraints: 1) as training data is usually of small size, training a model on such a small dataset is prone to overfitting [23] and catastrophic forgetting [24]. 2) In case there are only text data instead of paired audio-text data, training an ASR model is not straight-forward [25]. This is referred as the text-only data training constraint.

Overfitting: In a more specific context, overfitting is characterized by an ASR model becoming over-tuned to the training data, to the extent that the model begins to capture not only the underlying patterns but also extraneous noise and random variations present in the data. This may cause the model performance to degrade on unseen data. Overfitting frequently emerges as a significant challenge in

ASR adaptation and poses a substantial risk to the model's robustness to provide speech transcriptions.

Catastrophic forgetting: On the other hand, catastrophic forgetting refers to a situation where an ASR model firstly trained on dataset A loses its ability to accurately transcribe speech on dataset A after it is further trained or adapted on dataset B. In other words, when adapting an ASR model to a new domain or task, the model may forget or degrade its performance on the previously learned patterns or knowledge from the original domain.

Text-only data training constraint: Lastly, ASR model adaptation is usually performed with paired audio-text data. When only text data is present, a different adaptation approach is needed to carry out the training.

Numerous efforts are made to solve the above three problems for ASR adaptation [20, 21, 26, 27]. This thesis will focus on 2 categories of ASR adpatations, which are model adaptation [22] and text-only adaptation [28] in particular.

1.2.1 Layerwise adaptation

Model adaptation is a method to improve a model's performance in a specific domain, by updating the model weights to bias the model to the domain.

One strategy to address the overfitting and catastrophic forgetting issues involve implementing partial weight freezing [29–32] during the model adaptation process. This technique entails training and updating only a portion of the model's weights, while keeping another portion unchanged. The primary objective is to preserve the knowledge acquired by the original model prior to adaptation, thereby mitigating the risks associated with overfitting and catastrophic forgetting. By freezing specific model weights, the model is forced to generate intermediate feature representations that align with the fixed model weights. This serves to stabilize the training process, making the adaptation more controlled and less susceptible to extreme weight updates that could result in overfitting. Additionally, maintaining a subset of model weights in an unaltered state facilitates the retention of valuable knowledge encapsulated within these fixed weights, thereby effectively mitigating catastrophic forgetting.

To choose the model weights to freeze, previous works adopts a layer-wise freezing approach and relies on heuristics or a trial-and-error method to select the layer weights to freeze [33, 34]. Examples of some heuristics may include adapting the decoder only in the attention-encoder-decoder (AED) model architecture [2] and all the encoder layers should be frozen, if the goal is to adapt to a new text domain. Similarly, adaptation may only be applied to the encoder only and all decoder layers may be frozen, if the goal is to adapt to a new acoustic domain. This relies on the assumption that an encoder only performs acoustic modelling, and an decoder only performs language modelling.

However, the assumption is unrealistic as the encoder may also have language modelling capabilities. This is because [5] has shown that the encoder can be trained to predict words or sub-word units [35] with good accuracy for AED models trained with CTC auxiliary objectives [36]. If the encoder only learns the pronunciation information but not word semantic, the exact spelling of the predicted words or sub-words cannot be predicted with such high accuracy. This suggests that the encoder also has some language modelling capabilities.

Therefore, a more fine-grained selection approach is later adopted, where instead of freezing the entire encoder or decoder, the approach focuses on freezing the first few [37] of last few layers [38] of an encoder or decoder. It has been shown that the first few layers of an encoder has stronger acoustic modelling capabilities [39], while the last few layers of an encoder has stronger language modelling capabilities [40]. Therefore, the first few layers may be adapted for acoustic domain adaptation, and the last few layers may be adapted for text domain adaptation.

Nonetheless, the previously mentioned heuristics may prove less effective when confronted with situations where the adaptation domain is uncertain. In such cases, it may not be readily apparent which specific acoustic or text domain the pretrained model is pretrained on, nor does it provide clarity regarding the acoustic or text domains encompassed within the training data. Furthermore, the complexity increases when dealing with scenarios that entail multiple subdomains within a given domain. Consequently, determining the most suitable layers to freeze becomes a non-trivial task in such circumstances.

To find the optimal layers, trial-and-error grid-search approaches are used to sweep through all the possible layer combinations and adapt a separate model for each

combination. The optimal layer selection will be determined by the best performing model among all the adapted models in the search space. This is highly expensive and infeasible when computation resources are limited.

1.2.2 Text-to-speech synthesized data adaptation

An alternative approach to conducting ASR adaptation is through text-only adaptation, particularly suited for situations where only text, but not audio-text pairs, are available data sources. One approach for text-only adaptation is Text-to-Speech (TTS) data adaptation [41–45], where a TTS system is employed to generate synthetic audio from the existing text-only data. This synthetic audio is then combined with the original text to create artificial audio-text pairs to form new training data.

As improving the speaker diversity of the synthetic training data can improve an ASR model's robustness [46, 47], previous works use a single TTS model conditioned on multiple speakers to produce different speaker voices. However, one drawback of such approach is that only speaker voices from the TTS model's train set can be synthesized. This greatly limited the flexibility of producing diverse speaker voices.

To synthesize speaker voices not within the training data, [48] uses a single TTS system to morph different speaker voices by tuning the system's parameters to control the speaker characteristics of the synthesized audio, However, their methods are restricted by the trade-off [49] between the data's diversity and naturalness. To morph a speaker voice that is different from the training data, the quality of the synthesized audio will be compromised.

Lastly, to the best of our knowledge, all previous works for TTS data adaptation only leverages a single TTS system to synthesize fake audios. This limits the diversity of the synthesized data as the data only follows a single TTS output distribution.

1.3 Contribution

Layerwise adaptation: To address the issue of expensive grid search for layers to freeze, this thesis proposes an automatic strategy to choose the layers to freeze. Specifically, given a residual network, a layer-wise performance metric will be computed from each layer's skip connection, and the layers will be frozen based on each layer's validation performance. Experiments are conducted by applying our method to adapt a LibriSpeech-pretrained Conformer model [10] to a 10 hours rare word dataset which consists of many road names.

Our contributions are:

1. To the best of our knowledge, we are the first to propose an automatic method that can effectively find the optimal set of layers to freeze
2. Our method outperforms heuristic based selection strategy by relative 6.2% word error rate

Text-to-speech synthesized data adaptation: To address the issue of lack of diversity in single TTS synthesized data, this thesis proposes to leverage multiple TTS systems to synthesize fake audios to form fake audio-text pairs for model training, to improve synthesized audio diversity and prevent the ASR model to overfit to a specific TTS output distribution. Experiments are conducted by applying our method to adapt a LibriSpeech-pretrained Conformer model to a 10 hours rare word dataset which consists of many road names.

Our contributions are:

1. We explore using multiple TTS systems for TTS data synthesis
2. Our method can improve transcription accruacy by relative 9.8% lower word error rate compared to using single TTS system synthesized data.

1.4 Report Outline

This thesis is organised as follows: In Chapter 2, a thorough overview of end-to-end ASR and end-to-end ASR adpatation is provided. Chapter 3 provides baseline

results of the current state-of-the-art ASR adaptation methods, and a discussion of the limiting factors that affect the performance of such methods. In Chapter 4, TTS synthesized data with multiple TTS systems is proposed to improve to improved the diversity of the data. A layerwise adaptation method is proposed to automatically find good layers to freeze, and the experimental results demonstrate the strong performance of the method compared to the best-performing baseline techniques. Finally, the thesis concludes in Chapter 5 with a summary of the contributions and possible future research directions

Chapter 2

Literature Review

2.1 Overview of chapter

Thanks to advancements in deep learning, end-to-end Automatic Speech Recognition (ASR) models utilizing deep learning frameworks have made significant strides, achieving state-of-the-art results in various ASR tasks [5, 15]. However, a critical challenge persists in deep learning ASR models: their limited ability to generalize and accurately transcribe audio from unseen domains [50]. To address this issue, researchers employ end-to-end ASR adaptation techniques to enhance model performance when faced with new domains [19–22].

In this chapter, Section 2.1 serves as an introduction to end-to-end ASR. It commences by defining the concept of "end-to-end" ASR and subsequently explores the advantages these models hold over traditional statistical or hybrid approaches. Additionally, it outlines the overarching research objectives within the realm of end-to-end ASR and highlights key research directions aimed at improving transcription accuracy.

Following this, Section 2.2 offers an introduction to end-to-end ASR adaptation. It begins by delving into the various types of domains that require adaptation for ASR. Subsequently, it presents a taxonomy of ASR adaptation methods and provides an overview of state-of-the-art approaches in text-only adaptation and regularized adaptation. The section concludes with an analysis of the limitations associated with these adaptation techniques

2.2 End-to-end ASR overview

An end-to-end Automatic Speech Recognition (ASR) system represents a holistic approach to speech recognition, effectively transcribing spoken language into written text without necessitating intermediary processes, such as phoneme recognition [51] or pronunciation modeling [52]. These systems streamline the conventional ASR pipeline by harnessing the power of deep learning models, notably recurrent neural networks (RNNs) [1] and attention mechanisms [1, 2].

2.2.1 From statistical to end-to-end ASR models

In the early stages of Automatic Speech Recognition (ASR), models predominantly relied on statistical architectures [53, 54]. These models entailed a combination of Hidden Markov Models (HMMs) [55] to represent phonetic sequence distributions and Gaussian Mixture Models (GMMs) [56] to capture acoustic distributions. HMMs conceptualized speech signals as sequences of states, each corresponding to a specific acoustic feature distribution, while GMMs, operating within the framework of HMMs, aimed to model the probability distributions of acoustic features. Each state within the HMM was associated with a GMM describing the likelihood of observed features given that state.

However, this explicit modeling of acoustic signals with Gaussian distributions and phonetic sequences with HMMs may imposed limitations on the model's ability to capture intricate audio relationships [57]. Consequently, end-to-end ASR models have gradually supplanted statistical or hybrid models, emerging as the state-of-the-art solution for tasks with ample training data [58]. End-to-end Automatic Speech Recognition (ASR) systems offer a host of advantages over traditional statistical or hybrid ASR systems [59].

Firstly, end-to-end ASR systems [1] streamline the ASR pipeline by training a model with full capabilities of acoustic, pronunciation and language modelling. This obviates the need to train 3 different models to preform the 3 tasks separately and simplifies the training of ASR systems. Secondly, the end-to-end training process enables the joint optimization of all ASR system components [52]. This entails training the acoustic, pronunciation and language modeling components together,

promoting seamless integration and potentially enhances the model's adaptability to model more intricate audio relationships.

2.2.2 Development of end-to-end ASR model architecture

To enhance the quality of end-to-end ASR models, two critical factors warrant consideration: transcription accuracy and latency reduction. Over the past decade, extensive efforts have been dedicated to enhancing transcription accuracy for end-to-end ASR models [52, 59]. Existing research endeavors focused on improving accuracy can be categorized into three primary areas: model architecture, learning paradigms, and data engineering. This thesis will focus on discussing the first two areas.

The evolution of end-to-end ASR model architecture is characterized by the distinct modeling approaches on the handling of the audio-to-text alignments. Early endeavors in this field involved modeling alignments explicitly through the incorporation of latent variables that were subsequently marginalized out during both training and inference. Examples of such approaches encompassed connectionist temporal classification (CTC) [60], the recurrent neural network transducer (RNN-T) [61], the recurrent neural aligner (RNA) [62], and the hybrid auto-regressive transducer (HAT) [63].

More recent ASR model architectures relies on implicit alignment, exemplified by attention-based encoder-decoder (AED) architectures [2], which initially gained prominence in the context of machine translation [64]. In contrast to explicit alignment models like CTC, attention-based encoder-decoder models employ an attention mechanism to acquire the capacity to establish a correspondence between the entire acoustic sequence and individual labels [1].

2.2.3 The advantage of implicit alignment over explicit alignment

A critical distinguishing factor between explicit and implicit alignment modeling approaches lies in the granularity of the prediction token. For scenarios involving frame-level predictions, where word or subword [65] text tokens must be assigned to

frame-level logit outputs [66], explicit alignment is imperative to ascertain precisely at which frame location a word or subword is enunciated. Conversely, for word or subword-level prediction tasks, the alignment of text token labels to the logit outputs becomes unnecessary, as both the labels and the logit outputs operate at the same word or subword level. This affords implicit alignment several advantages over explicit alignment.

Firstly, we argue that implicit alignment may have greater flexibility, as it obviates the need to map specific words or subwords to explicit locations within the audio signal. This proves advantageous in cases where the audio contains slang words or unclear pronunciations. For instance, consider the word "we will" in International Phonetic Alphabet (IPA) format, which may be pronounced as "$/v//vill/$" or "$/wl/$". In the former case, it is apparent that one can map the word "we" to the frames corresponding to the pronunciation "$/v/$" and "will" to the frames corresponding to "$/vill/$." However, the mapping of "we will" to the frames associated with "$/wl/$" is ambiguous.

Secondly, models utilizing explicit alignment may exhibit higher memory consumption, particularly in relation to vocabulary size, when compared to models employing implicit alignment. Consider a multilingual AED model like Whisper [14], featuring a vocabulary of 51,865 tokens. If a 30-second audio input corresponds to 1,500 frames in Whisper's encoder output, explicit alignment at the frame level results in logits with a size of $51,865 \times 1,500$. In contrast, if the logits are implicitly aligned at the word or subword level, assuming a transcript with a sequence length of 50, the logits would only occupy a size of $51,865 \times 50$. This size reduction is orders of magnitude smaller than the former, highlighting the efficiency gains associated with implicit alignment.

2.2.4 Development of AED model architectures

Numerous efforts are made to improve the AED ASR model architecture to improve transpiration accuracy. In the following Sections 2.2.4.1 to 2.2.4.4, we will discuss four end-to-end AED ASR models that modeled alignments implicitly.

2.2.4.1 Listen, Attend and Spell (LAS)

The Listen, Attend and Spell model [1] is the first end-to-end ASR model that models alignments implicitly by taking in audio input to output sequences of characters including a blank symbol.

Comprising two sub-modules, the LAS model encompasses the listener and the speller. The listener serves as an acoustic encoder, with its primary operation aptly named "Listen." Through this process, the listener transforms the initial audio signal denoted as x into a high-level representation, denoted as $h = (h_1, ..., h_U)$, where $U \leq T$. In tandem, the speller functions as an attention-based character decoder, with its principal function referred to as "AttendAndSpell". Within the AttendAndSpell operation, the model consumes the high-level representation h and and produces a probability distribution $P(y|x)$ over character sequences y. [1] formulates it as:

$$h = Listen(x) \tag{2.1}$$

$$P(y|x) = AttendAndSpell(h, y) \tag{2.2}$$

Figure 2.1 from [1] visualizes LAS with these two components.

2.2.4.2 Transformer

The LAS model offers a notable advantage in its capacity to allow text-level features denoted as s (as illustrated in Figure 2.1) direct access to any position within frame-level features h. This enhancement significantly bolsters the model's capability to capture long-distance cross-modal relationships effectively.

Nonetheless, the LAS model presents a drawback in its inability to model long-distance intra-modal relationships. In other words, text-level features may not directly attend to other text-level features, and frame-level features may similarly lack direct interactions with one another. This limitation stems from the architectural underpinning of the listener and the speller, both of which employ the Bidirectional Long Short-Term Memory (BLSTM) architecture [67]. BLSTM

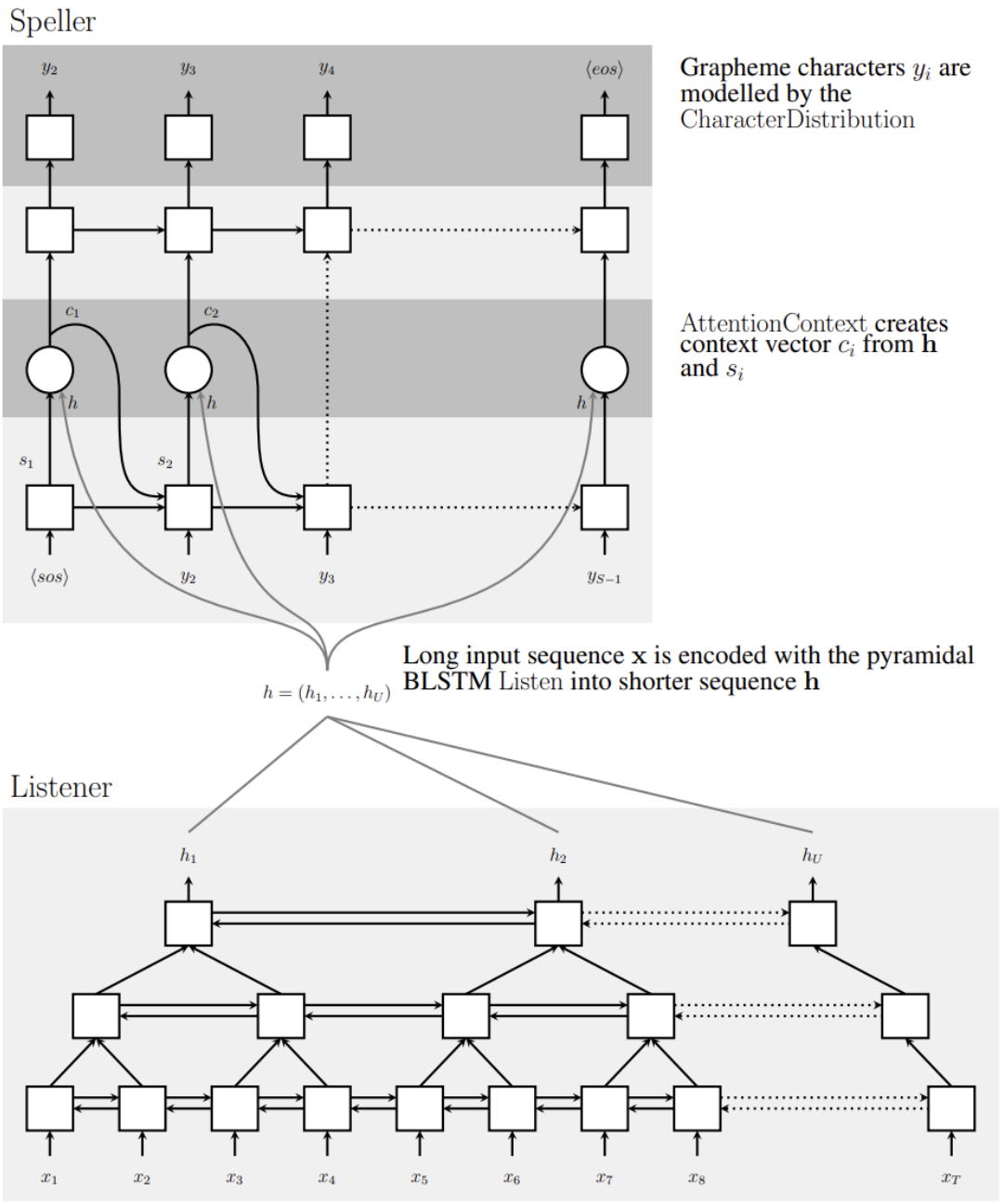

FIGURE 2.1: A diagram of the Listen, Attend and Spell (LAS) model shown in [1]. The listener is a pyramidal BLSTM encoding our input sequence x into high level features h, the speller is an attention-based decoder generating the y characters from h.

processes features sequentially and employs a single vector to summarize all encountered features. Consequently, the next input feature to the BLSTM can only indirectly attend to other features by accessing the summarization vector.

To address this constraint, [2] introduces the Transformer architecture, a paradigm that foregoes recurrence and relies entirely on an attention mechanism to establish global dependencies between features. Figure 2.2 from [2] provides an illustrative example of the attention mechanism.

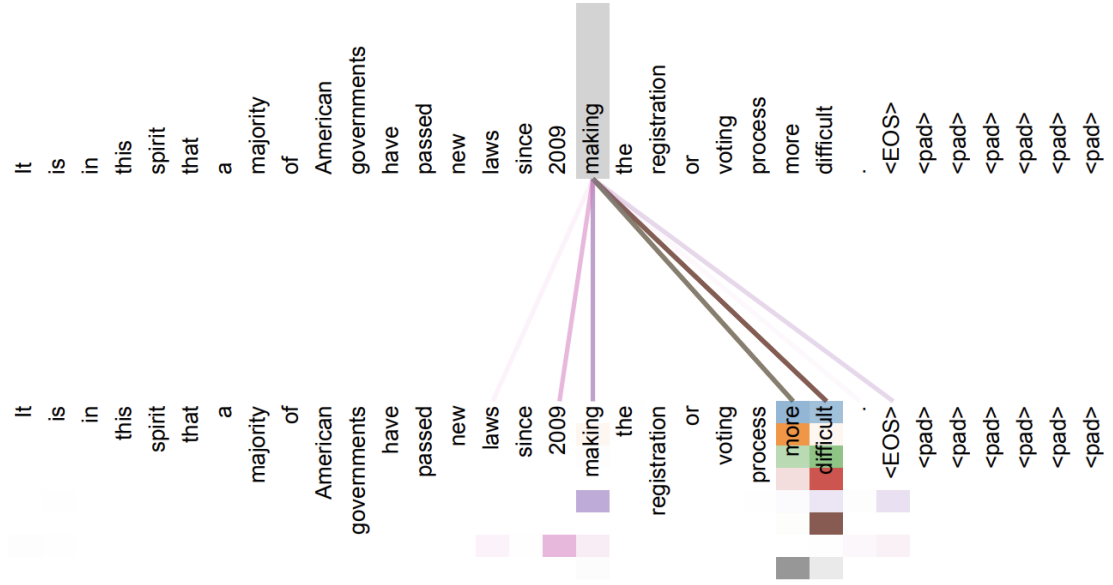

FIGURE 2.2: An example of the attention mechanism in Transformer shown in [2]. A line connecting the bottom word to the upper word means that the bottom word feature is attending directly to the upper word features. From this example, many of the word features on the bottom row attend to a distant dependency of the word 'making' on the upper row, completing the phrase 'making...more difficult'. This suggests that long-distance dependencies are modelled by the attention mechanism.

The Transformer architecture shares similarities with LAS in that both employ an encoder-decoder structure. However, instead of utilizing BLSTMs in the encoder and decoder, the Transformer replaces them with transformer blocks that rely on a global attention mechanism. More specifically, the transformer block comprises a stack of attention and feedforward layers.

Resembling a key-value database, [2] describes the function of the attention layer as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are vectors. The output is computed as a weighted sum of the

values, with the weight assigned to each value being determined by a compatibility function between the query and the corresponding key.

In this process, each feature from a sequence s undergoes a series of transformations through three separate linear layers to yield query, key, and value vectors, each of dimension d_k. Subsequently, the sequences of query, key, and value vectors are organized into matrices denoted as Q, K, and V respectively, facilitating the subsequent attention mechanism:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.3}$$

The introduction of the Attention mechanism has bestowed upon models the flexibility to direct their focus towards features at any position, thereby enhancing their capacity to model intricate relationships between distant features.

An illustrative instance of a Transformer-based model is Whisper [14]. Whisper is trained on an extensive and diverse audio dataset, and it showcases multitasking capabilities, capable of tasks such as multilingual speech recognition, speech translation, and language identification. This model has shown strong generalisation capability, consistently delivering state-of-the-art results across various audio tasks spanning diverse domains.

2.2.4.3 Conformer

While attention layers offer robust global modeling capabilities, they exhibit a limitation in efficiently capturing local features, i.e. i.e. features that are temporally close [10]. Conversely, convolution layers have demonstrated success in ASR by progressively capturing local context through a series of layers, each with its own local receptive field [68]. To harness the strengths of both approaches, Conformer [10] combines convolution with attention layers within the encoder.

As depicted in Figure 2.3, a Conformer block is constructed by stacking four modules together. These modules include a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module at the end.

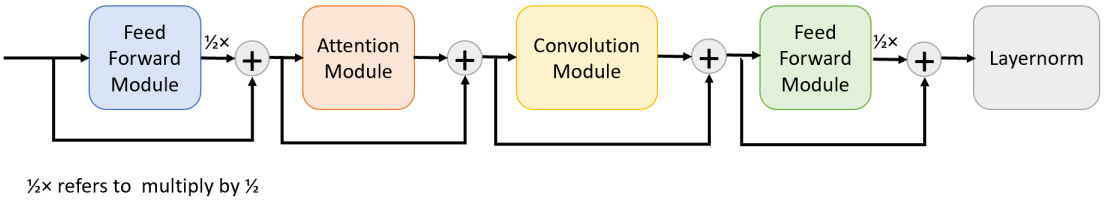

FIGURE 2.3: A Conformer block is composed of two macaron-like [3] feed-forward layers with half-step residual connections, which sandwiches a multi-headed self-attention and convolution module. Layer normalisation [4] is applied at the end of the block.

2.2.4.4 E-branchformer

The Conformer architecture integrates convolution and attention layers in a sequential manner. However, this static single-branch architecture poses challenges in terms of interpretability and modifiability. To facilitate the design of more adaptable architectures, [11] introduces the Branchformer framework. Branchformer aims to enhance our understanding of the local and global relationships employed in different encoder layers by introducing multiple parallel branches with learnable merge weights. Specifically, it employs dedicated branches for gated convolution [69] and self-attention while integrating local and global context from each branch. The fusion of context outputs is achieved through a linear combination.

Consequently, through the merge weights of the parallel branches, researchers gain insights into the significance of each layer and whether certain components play a more prominent role in the layers.

Nonetheless, subsequent research by [5] demonstrates that the point-wise and linear combination of outputs in Branchformer may not be optimal. This leads to the introduction of E-Branchformer, an enhanced version that incorporates an effective merging method and stacking additional point-wise modules.

The architecture of the merging method is illustrated in Figure 2.4. As depicted, E-Branchformer harmonizes the local and global context outputs through the utilization of depth-wise convolution with varying kernel sizes. Additionally, a squeeze-and-excitation block is incorporated immediately after the convolution module. [5] shows that this new architecture outperformed Conformer in terms of transcription accuracy.

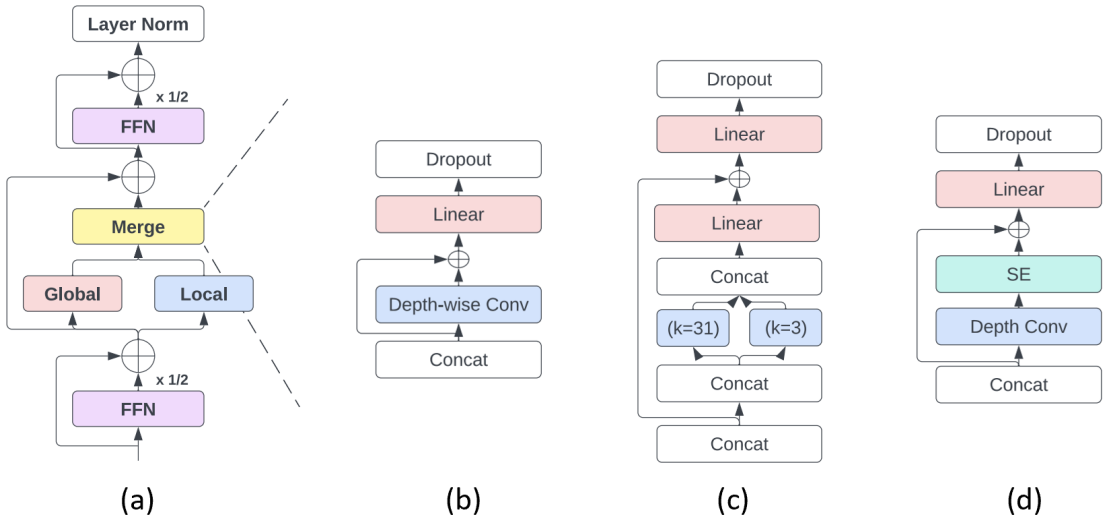

FIGURE 2.4: A figure from [5]. (a) the proposed E-Branchformer block and different methods for the merge module: (b) is applying a depth-wise convolution, and (c) uses multiple convolutions with different kernel size, e.g., 31 and 3. (d) employs a squeeze-and-excitation block

2.2.5 Self-supervised Learning ASR

Up to this point, our discussion has revolved around supervised learning [70], where models are trained using labeled text-audio pairs. Another significant learning paradigm to consider is self-supervised learning [71], which involves training models solely on audio data when corresponding text labels are unavailable. Self-supervised learning holds immense potential due to its ability to leverage a vast amount of unlabeled data for training [72]. While this thesis primarily focuses on supervised learning, it is essential to acknowledge self-supervised learning due to its remarkable success in the field of ASR. Here, we delve into two notable examples of self-supervised learning approaches for ASR.

Wav2vec2.0 [6] presents a framework for self-supervised learning to derive representations from raw audio data. This approach involves encoding speech audio through a multi-layer convolutional neural network and subsequently masking segments within the resulting latent speech representations [73]. These latent representations are then input into a Transformer network to generate contextualized representations. The model undergoes training via a contrastive task, where the goal is to distinguish the genuine latent representation from distractors [74]. The architecture of the model is illustrated in Figure 2.5.

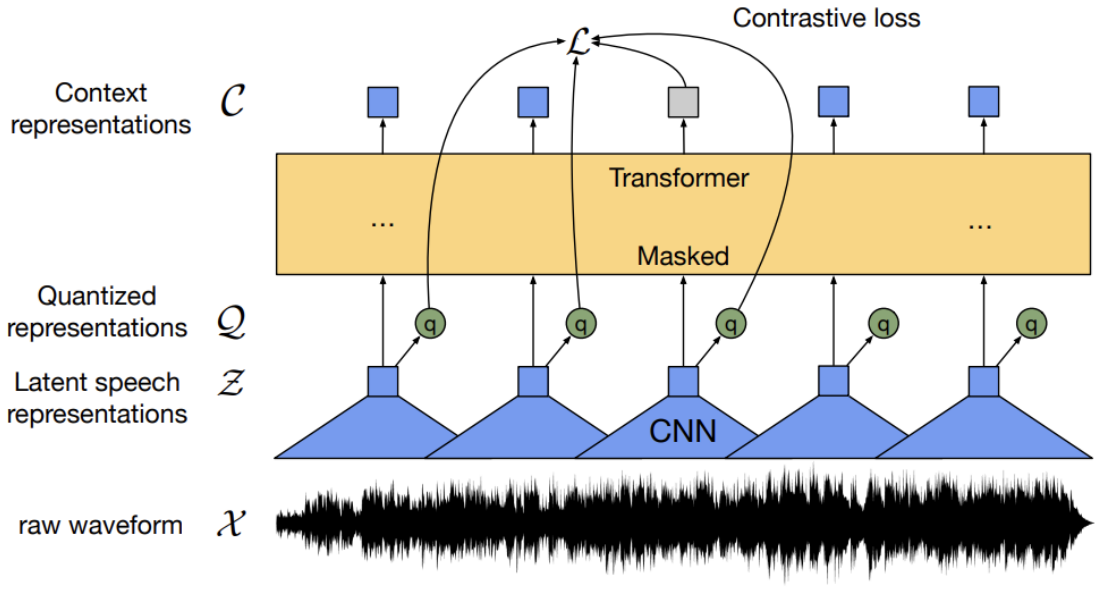

FIGURE 2.5: Visualization of the wav2vec2.0 framework from [6], wherein the model jointly learns contextualized speech representations and a collection of discretized speech units. Partially masked representations Z are fed to the Transformer encoder to recover the representations at the masked positions. A contrastive learning framework is used to pull the recovered representation in C to be close to the quantized version of the representation at the masked position in Z and push it away from other quantized representations not at that position.

However, as depicted in Figure 2.5, Wav2vec2.0 learns to recover representations from Z, which are features encoded by shallow Convolution layers. These features may not contain the rich semantic information found in the intermediate representations of the deeper layers within the Transformer encoder. Consequently, these shallow Convolution layers may not be the most suitable features for reconstruction, largely due to the limited capacity they offer. [75] has conducted ablation studies to support the claim.

To enable learning from representations deeper within the Transformer encoder, as opposed to relying on the weaker representations encoded by the shallow Convolution layers, Hubert [75] adopts a form of self-training that involves employing K-means clustering [76] to generate pseudo labels from the intermediate representations within the Transformer encoder. Subsequently, the same model is trained to predict these pseudo labels. The newly trained model is then utilized to provide representations for creating pseudo labels in the subsequent training iterations.

To enhance the quality of the pseudo labels, Hubert employs two key strategies.

Firstly, an ensemble of K-means models with various codebook sizes is employed to create targets of different granularity for pseudo-labeling. Secondly, cluster assignments are iteratively refined, using the latest updated model's intermediate representations to generate pseudo labels for training a new model in the subsequent iteration. [75] shows that the method has better performance than wav2vec2.0 in terms of transcription accuracy.

2.3 End-to-end ASR Adaptation Overview

2.3.1 The Out-of-domain problem

There are many speech audio domains in the real-world scenarios which can generally be categorized by the following key characteristics: speaker, accent/dialect, language, background noise, audio codec, speech context and recording environment. Due to the complicated real-world environment, ASR models trained for scenario A often contains a set of domains that may be different then some other scenario B in three ways. 1) Scenario A and B contains a completely disjoint set of domains, i.e. they are completely unrelated. 2) A partial set of domains in Scenario A and B intersects, i.e. they are kind of related but not completely. 3) The domains in Scenario A is a superset of domains in Scenario B or vice versa.

When there arises a need to transcribe speech in a different context or scenario, an appealing solution is to repurpose an existing ASR model originally trained for another scenario. However, when an ASR model attempts to transcribe speech in an unfamiliar domain, it may not exhibit the same level of performance as it does in its native domain. This discrepancy is attributed to variations in domain characteristics, leading to what is commonly termed the "out-of-domain" or "domain mismatch" problem [77].

To enhance the performance of a pretrained ASR model when applied to a new domain, it becomes imperative to adjust the model to align more closely with the characteristics of the target domain. This strategy is commonly referred to as ASR adaptation, and it entails customizing the model to better suit the specific domain, ultimately improving its transcription capabilities in the new scenario.

2.3.2 ASR adaptation Taxonomy

Different ASR adaptation approaches are developed to handle different adaptation scenarios. A taxonomy of the adaptation approaches are shown in Figure 2.6. Noted that there are many more adaptation techniques, but we try to give the ones that we have come across for ASR adaptation.

As shown in Figure 2.6, existing research efforts devoted to ASR adaptation can be grouped into the following research topics: regularized adaptation, knowledge injection, efficient adaptation and common adaptation approaches for statistical/hybrid models.

For regularized adaptation, the goal is to mitigate overfitting and catastrophic forgetting when adapting to small amount of trainig data. One attempt [78–80] performs data balancing by uniformly sampling data samples from multiple domain sources, to ensure the adapted model is not biased to domains with relatively more data samples. Alternatively, [81] applies a separate scalar learning rate for each data domain to prioritize learning for domain with scarce data.

Another line of research regularize adaptation by learning an auxiliary loss. They include L2 regularization [82], Kullback Leibler divergence [83] and adversarial multitask learning [84]. A third line of research focus on adapting certain layers or subset of parameters [29, 30, 32].

For knowledge injection, the goal is to inject domain specific knowledge to the ASR model to improve performance in the domain. To adapt on different language domains, a one hot encoding [78, 85] or a learnt embedding [86] is inserted to the input sequence of RNN encoder, the RNN decoder or both. To adapt on different speakers, i-vectors [87–89] or x-vectors [90] are concatenated to every frame of acoustic representations. To adapt on different noise domains, a scalar noise estimation is appended to the input vector [91]. [92] modifies the prompt input to a decoder to inject task-specific knowledge to the model.

For efficient adaptation, the goal is to perform adaptation under various resource constraints. One line of research uses an extra model component like Adapter [93–95] or shifting and scaling parameters [96] to modify layer activations. LHUC [97, 98] and BLHUC [99] linearly re-combines hidden units by learning a scaling factor for each hidden unit. Another line of research uses low rank matrix to

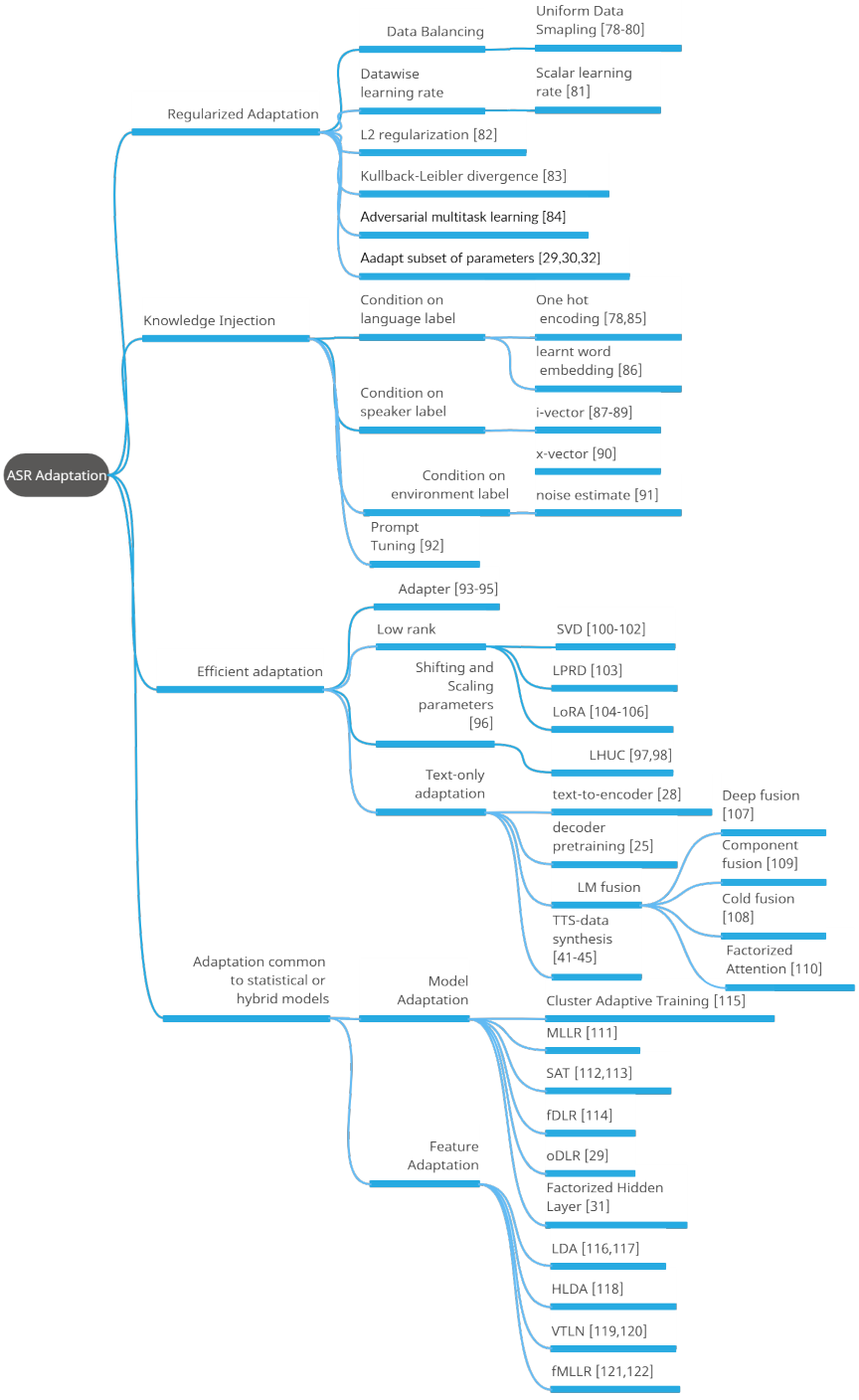

FIGURE 2.6: A taxonomy of ASR adpatation approaches

perform adaptations. They include singular value decomposition (SVD) pruning followed by linear layer in lower dimensional space [100–102], low-rank plus diagonal (LRPD) decomposition [103] and LoRA [104–106] which freezes the pretrained model weights and injects trainable rank decomposition matrices into each model layer.

A third line of research focus on text-only adaptation which aims to perform adaptation given that text is the only data source available. To solve the problem of the absence of speech representation during training, [28] builds a text-to-encoder (TTE) model which learns to predict the hid-den states of the E2E-ASR encoder rather than the speech. [25] directly pretrain a decoder using text only to incorporate text knowledge into the ASR model. [41–45] uses a TTS system to generate synthesized speech to form fake audio-text pairs as the training data. Further more, structural LM fusion methods, such as deep fusion [107], cold fusion [108], component fusion [109] and Factorized Attention [110] directly adapt an external language model (LM) on the text-only data and learn the combination of E2E ASR system and the external LM.

For adaptation approaches that are designed for statistical/hybrid models, one line of research performs model adaptation which include maximum likelihood linear regression (MLLR) [111], where the mean or variance of Gaussian mixtures is transformed by a linear transformation; speaker adaptive training (SAT) [112, 113], which uses two distinct sets of parameters are introduced to separately model speech and non-speech variabilities; feature discriminative linear regression (fDLR) [114]; and output-feature discriminative linear regression (oDLR) [29] which adapts on the hidden layer closest to the output side. Cluster adaptive training [115] uses an interpolation vector to combine multiple DNN bases into a single adapted DNN. Factorized Hidden Layer (FHL) [31] parameterizes the affine transformation of a hidden layer as a linear interpolation of a set of bases,

Another line of research performs feature adaption which include dimension reduction techniques like Linear discriminant analysis (LDA) [116, 117] and heteroscedastic discriminant analysis (HLDA) [118] which further applies a diagonalizing linear transformation, and feature normalization techinques like VTLN [119, 120] which warps the frequency axis to normalize the effect of varying vocal-tract resonances and fMLLR [121, 122] which a linear transformation is used to adapt both mean and variance simultaneously.

As this thesis will focus on text-only adaptation and regularized adaptation, their related works will be discussed in more details in the following sections.

2.3.3 Text-only adaptation

A well-trained end-to-end ASR model need an extensive dataset of human-transcribed audio, a resource that can be challenging to amass. In contrast, unpaired data, particularly text data, is readily obtainable but has not been fully harnessed in the training of end-to-end ASR models.

The subsequent section will explore diverse approaches aimed at harnessing text corpora only to enhance the overall performance of end-to-end ASR systems.

2.3.3.1 Text-to-encoder model

In the context of attention-encoder-decoder (AED)-based end-to-end ASR systems, the ASR model comprises an audio encoder and a decoder. As illustrated in the left part of Figure 2.7, the audio encoder processes audio input to generate real acoustic representations. In the central part of the figure, the decoder takes a partial text sequence and these real acoustic representations to transcribe the next word, such as "apple", in the text sequence. However, when audio data is absent during training, the output of the audio encoder becomes unavailable. To address this absence, an alternative solution, as depicted on the right part of the figure, involves constructing a text-to-encoder (TTE) model. This TTE model receives a complete text sequence as input and is trained to predict synthetic or "fake" acoustic representations.

The following are some works that build on the TTE model. [28] developed a text-to-encoder (TTE) model that focuses on predicting the hidden states of the E2E-ASR encoder. Notably, this TTE model does not rely on speaker or prosody information, which typically do not exist in the text data. In a related study, [123] introduced cycle-consistency loss during the training of an ASR-TTE system. This addition to the training process helps mitigate the discrepancies between the original encoder hidden states and the reconstructed versions, ultimately enhancing the performance and robustness of the system.

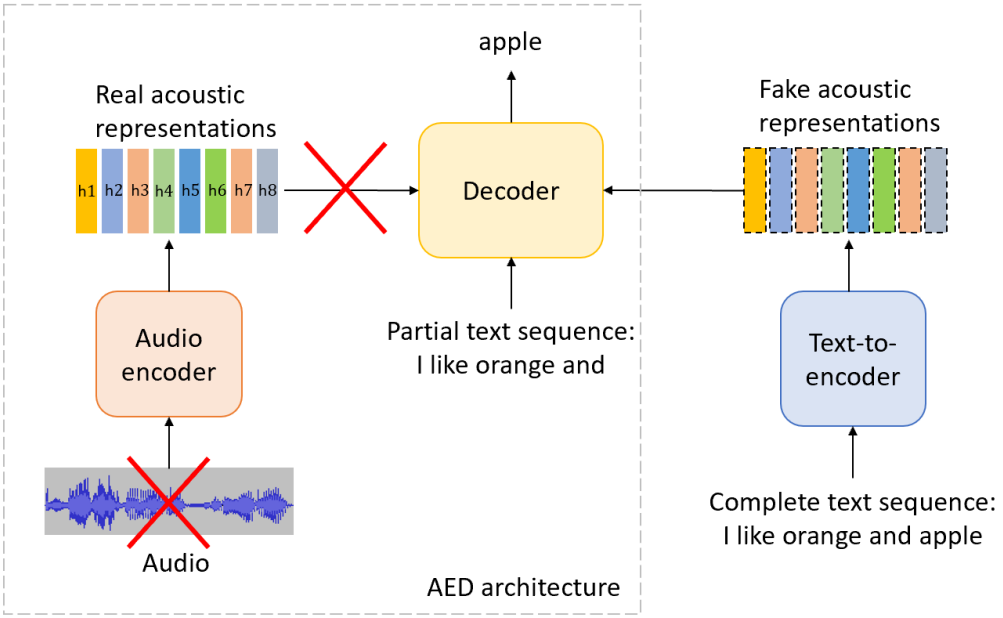

FIGURE 2.7: When audio (bottom-left) is not present, a text-to-encoder (right-middle) will output fake acoustic representations (top-right) as a replacement to real acoustic representations (top-left), which is the output of the audio encoder (left-middle), to be fed into the decoder (middle).

2.3.3.2 Decoder pretraining

However, one challenge associated with the text-to-encoder approach lies in its intricate training procedure, which comprises three distinct stages. First, it involves the collection of text data paired with corresponding real acoustic representations obtained from the pretrained audio encoder to constitute the training dataset for the text-to-encoder. Subsequently, the text-to-encoder model undergoes training, often accompanied by hyperparameter tuning to optimize its performance. Lastly, in scenarios where only unpaired text data is available for training, the text-to-encoder replaces the audio encoder to generate synthetic acoustic representations, which are then fed into the decoder during training.

To obviate the necessity for training an explicit text-to-encoder model, [25] proposes an alternative approach. In this method, empty or artificial states are introduced to substitute the real acoustic representations, eliminating the need to provide the decoder with synthetic acoustic representations. Specifically, the AED ASR model leverages unpaired text data through language model pretraining. During this pretraining phase, the decoder is supplied with partial text sequences and empty states in lieu of actual acoustic representations, allowing the model to effectively

learn from the unpaired text data. However, this method also requires an additional model component like the text-to-encoder approach, which complicates the adaptation process.

2.3.3.3 TTS data adaptation

While decoder pretraining has proven to be effective in enabling AED ASR models to learn from unpaired text data without necessitating an additional text-to-encoder model component, the training process remains somewhat complex, requiring a two-stage approach. This entails initially pretraining the decoder and subsequently training the complete AED model, adding an extra layer of intricacy to the overall training procedure.

Recent advancements in text-to-speech (TTS) systems have elevated speech synthesis to a level where the generated utterances closely approximate human speech [41].

As an alternative means of leveraging text-only data for ASR model training, employing a TTS system to produce synthetic audio is gaining traction. This synthetic audio can then be paired with corresponding text, creating additional source of training data. The method does not require extra training stages or modification to an ASR model

The impact of using different TTS models to generate synthetic data for various ASR architectures is explored in [124–127]. The authors of [128] propose a combination of generative adversarial networks (GAN) and multi-style training (MTR) to enhance the acoustic diversity within the synthesized data. Additionally, several studies have sought to augment training data with synthesized speech, particularly in low-resource ASR scenarios [129]. [42] focuses on leveraging a TTS engine to provide synthetic audio, thereby improving the recognition of out-of-vocabulary words that are entirely absent from the original training data. Furthermore, [43] investigates the utilization of contemporary speech synthesis techniques to customize ASR systems for specific target domains, relying solely on relevant text corpora for this purpose.

2.3.4 Regularized adaptation

ASR adaptation typically involves working with a limited amount of training data, which tends to be relatively small in size. However, it has been observed that training a model on such a small dataset increases the risk of overfitting [130] and catastrophic forgetting [131]. To address these challenges, one effective strategy is to implement regularized adaptation.

2.3.4.1 Partial weights freezing

One approach to regularized adaptation involves the practice of not updating, effectively "freezing," specific layers or components within an ASR model during the adaptation process. These frozen weights remain constant and do not undergo any changes throughout the training process. While weight freezing also falls within the realm of efficient adaptation techniques, this thesis will primarily delve into its role in regularizing the adaptation process.

The act of freezing weights serves to preserve the knowledge embedded in the original model prior to adaptation, thereby mitigating the risks associated with overfitting and catastrophic forgetting [29, 132]. By locking certain portions of the model's weights, the model is compelled to generate intermediate feature representations that align with the frozen weight configurations. This contributes to the stabilization of the training process, rendering adaptation more controlled and less susceptible to extreme weight updates that could result in overfitting. Additionally, as a segment of the model's weights remains unaltered, it helps preserve the knowledge encoded within these frozen parameters, thus acting as a safeguard against catastrophic forgetting.

2.3.4.2 Regularizing parameters with auxiliary losses

While weight freezing stands as a straightforward and effective method for regularization during adaptation, there are cases where this approach is inferior than directly adapting the whole model [133]. This occurs because frozen weights are unable to learn from new data during the adaptation process. To provide an alternative that balances adaptation and regularization, auxiliary losses come into play,

constraining weight updates while still affording a limited degree of adaptability to new tasks.

One commonly employed regularization technique is L2 regularization. [82] utilizes L2 regularization for weight decay, discouraging the updated weights to deviate from the original weights. This practice has been shown to enhance the generalization capability of ASR models. L2 regularization is achieved by introducing a squared penalty term, effectively augmenting the training loss function. This additional term encourages the updated weights to align with the unadapted network weights. Given that at step i, weight update Δw_i follows the equation:

$$\Delta w_i = -\eta \nabla_w E(w_i) + \alpha \Delta w_{i-1} \tag{2.4}$$

where Δw represents the gradient operator with respect to the weight vector w, $E(w)$ the training loss function, η the learning rate, and α is a parameter for mementum update [134]. After adding L2 regularation, the new equation for weight update is then:

$$\Delta w_i = -\eta \nabla_w E(w_i) + \alpha \Delta w_{i-1} - \beta(w_{i-1} - w_0) \tag{2.5}$$

where β is the weight decay factor on the L2 penalty term which decays the weights towards the original model weights.

In addition to penalizing weights through L2 regularization, another approach involves enforcing the adapted model to yield output distributions akin to those of the unadapted model. [83] accomplishes this by applying Kullback–Leibler divergence (KLD) regularization. This technique forces the senone distribution estimated by the adapted model to closely resemble that estimated by the unadapted model. [83] has shown that this form of regularization is equivalent to modifying the target distribution within the conventional backpropagation algorithm.

An alternative method for regularizing the model's output distribution involves adversarial learning. For instance, [84] employs an adversarial speaker adaptation (ASA) scheme, where adversarial learning is employed to regulate the distribution of deep hidden features in a speaker-dependent (SD) deep neural network (DNN)

acoustic model, bringing it closer to that of a fixed speaker-independent (SI) DNN acoustic model during adaptation.

Within ASA, an auxiliary discriminator network is introduced to classify whether input deep features originate from an SD or SI acoustic model. Using a fixed SI acoustic model as a reference, the discriminator network is trained concurrently with the SD acoustic model, optimizing both the primary task of minimizing the senone classification loss and the secondary task of minimizing the SD/SI discrimination loss on the adaptation data. Through this adversarial multitask learning framework, senone-discriminative deep features are acquired in the SD model, mirroring the distribution of the SI model. This results in a regularized and adapted deep feature, which is expected to yield improved ASR performance on test speech data from the target speaker.

2.3.4.3 Comparison of partial weights freezing and auxiliary loss regularization

Despite the efficacy of utilizing auxiliary losses as a means to regularize adaptation, this approach does come with certain limitations. Firstly, we argue that auxiliary loss does not have the added benefit of reducing training cost as partial weights freezing does, as all the weights are required to be updated. Partial weight freezing only focuses on the weights that needed to be updated.

Secondly, the introduction of auxiliary losses may still cause small modification to all the model weights and potentially lead to performance degradation in situations where the model exhibits extreme sensitivity to even minor weight changes, a phenomenon typically associated with overtraining. On the other hand, partial weight freezing can guarantee that the weights that are prone to overfitting are never modified.

As a result of these considerations, weight freezing has emerged as an alternative approach due to its simplicity and the versatility it offers, making it applicable to a wide range of adaptation tasks.

2.4 Summary of chapter

In summary, this chapter explores the the development of model architectures in end-to-end Automatic Speech Recognition. End-to-end ASR is preferred over conventional statistical methods due to its high flexibility in modelling more complicated relationships in audios. Four end-to-end ASR AED model architectures are discussed to improve the audio modelling capability.

Next, this chapter gives an overview of the previous works on ASR adaptation and focus on approaches mainly related to text-only adaption and regularized adaption. For text-only adaptation, text-to-encoder, decoder pretraining and TTS data synthesis approaches are discussed, and TTS data synthesis excels as a simple method for text-only adaptation without involving complicated training procedures or the addition of any model components.

For Regularized adaptation, weight freezing and different auxiliary losses for regularization are discussed. Weight freezing has emerged as an alternative approach to adaptation due to its simplicity and its broad applications.

Chapter 3

Baseline End-to-end Transformer ASR adaptation

3.1 Overview of chapter

For ASR adaptation, the goal is to adapt a pretrained ASR model to improve its performance on a target domain. The following sections will describe the baseline methods we use to reach state-of-the-art performance for different ASR adaptation tasks.

Section 3.2 of this chapter aims to provide an overview of the available ASR datasets for the training and testing of ASR models. This thesis will further focus on three corpus to be used in the experiments of the baseline methods. The evaluation metrics for ASR models will also be discussed.

Section 3.3 of this chapter will discuss baseline adaptation methods based on the Conformer model architecture, where the architecture has been discussed in Section 3.4. The section will provide details on the network configurations of the models used, and the dataset and training configuration to prepare a pretrained ASR model. Then the dataset and training configuration to adapt the pretrained model will be discussed. Finally, the section will show and analyze the experiment results.

Section 3.4 of this chapter will discuss baseline adaptation methods based on the Whisper ASR model, which uses the Transformer model architecture as discussed in Section 2.2.4.2. The section will provide details on the network configurations of

the models used, and the dataset and training configuration to prepare Whisper. Then the dataset and training configuration to adapt Whisper will be discussed. Finally, the section will show and analyze the experiment results.

3.2 Corpus

There are many ASR datasets available on the internet for training and testing ASR models that covers different domains and use cases [14, 58]. The dataset sizes can vary from 10 hours to 1000 hours of audios. Some languages that have the most abundant ASR dataset resources may include English, Chinese, German and Spanish [14]. This thesis will focus on English and Chinese ASR tasks. For the use cases, the thesis will focus on adapting ASR models to 1) transcribe rare words which are often associated with named entities like names and location and 2) transcribe on low resource data domains. As a result, this thesis will use 4 datasets to conduct experiments for ASR adaptation. They are Librispeech, AISHELL-1, Datatang-ZH and IMDA2 which will be discussed below.

LibriSpeech [17] is an English dataset. It is a read speech dataset based on LibriVox's audio books [135]. The speech is sampled at 16 kHz. The corpus is freely available under the permissive CC BY 4.0 license [136]. The training portion of the corpus is split into three subsets named train-clean-100, train-clean-360 and train-other-500, with approximate size 100, 360 and 500 hours respectively. Additional data in the same domain as train-clean-100 is pooled into a 5 hours development and test set separately. They are named dev-clean and test-clean. Similarly, the same is applied for the data in the same domain as train-other-500 to obtain dev-other and test-other. We only use the train-clean-100 subset and evaluate the model on the test-clean and test-other subset.

AISHELL-1 [137] is a Chinese dataset. It contains 400 speakers and over 170 hours of Mandarin speech data, which cover topics in finance, science and technology, sports, entertainments and news. The corpus is freely available under Apache 2.0 license [138]. The corpus includes training set, development set and test sets. Training set contains 120,098 utterances from 340 speakers; development set contains 14,326 utterance from the 40 speakers; Test set contains 7,176 utterances

from 20 speakers. For each speaker, around 360 utterances which sums to about 26 minutes of speech are released.

Datatang-ZH is a spontaneous Chinese dataset. It is an in-house telephony mandarin spontaneous dataset provided by Datatang [139]. Only its test-set is used for our experiments. It contains 2171 speakers and 7 hours of Mandarin speech data. The test-set is considered more challenging than AISHELL-1 as there are telephony background noise and the speech type is spontaneous.

IMDA2 is a English road names dataset. It contains road names as rare words, we subset 10 hours of real audio-text training data pairs containing road names and addresses from National Corpus SG [140] as our real train set. They are recordings of people asking for directions and consists of 13K unique utterances. We also partition 1.63h of parallel data as the test set. Table 3.1 shows the details of the rare word (road names and addresses) coverage for the train and test set. Table 3.2 shows some example sentences in the train and test set.

TABLE 3.1: Percentage of rare words (road names and addresses) in the train and test set's text transcript. W_T is the total number of words, W_R is the total number of rare words, W_L is the total number of overlapped rare words in train and test set

Dataset	W_T	W_R	W_R/W_T (%)	W_L
Real train set	115119	53549	46.52	5466
Test set	11015	5825	52.88	-

TABLE 3.2: Example sentences in the IMDA2 rare words dataset

Example sentences
can i take the m r t to tiong poh avenue
weld road singapore centre for chinese language and telok blangah hill
farrer park road jalan selanting and sofitel singapore resort and spa
jalan mariam jalan shaer and holland road west
berrima road auckland road west and seraya road
universal studios singapore south east asia hotel and bedford road
faith bible dunman road and institute of high performance computing
i am waiting for shasta to come back from kee sun avenue
gek poh shopping centre fernhill road and peiying primary school
regent street aljunied avenue two and sky suites

3.3 Evaluation metric for ASR

The word error rate (WER) metric [141] is used for ASR evaluations. WER quantifies the difference between the recognized words produced by an ASR system and the ground truth words from the target transcription. It provides a numerical measure of how well or poorly the ASR system performs in converting spoken language into text. A lower WER in speech-to-text means better accuracy in recognizing speech. WER is formally defined as:

$$WER = \frac{S + D + I}{N} \tag{3.1}$$

where N is the total number of words in the ground truth transcript. S, D and I are the minimal number of substitution, deletion and insertion operations that are required to convert the predicted transcription to the ground-truth transcript. For example given a predicted transcript "I eat aple". It requires 1 substitution operation to replace "aple" to "apple" to convert the predicted transcript to the ground truth transcript "I eat apple".

The minimal number of operations to convert a predicted transcript to the ground truth transcript is referred as the edit distance, and it is calculated using the Levenshtein distance algorithm [142]. Another metric character error rate is also similar to WER, except that it calculates edit distance on the character level instead of the word level.

3.4 Conformer Adaptation

This chapter will discuss baseline adaptation methods based on the Conformer model architecture, where the architecture has been discussed in Section .

3.4.1 Network configurations for Confomer models

The network configurations of the Conformer model is shown in Figure 3.1. The model parameter size is $116M$. It follows an attention-encoder-decoder architecture [2], which consists of a Conformer encoder and an attention autoregressive (AR)

decoder. Log Mel filterbank features [143] are extracted with a 25ms and 32ms window size and a 10ms and 16ms stride, model feature hidden dimension is 512, encoder Feed Forward Network (FFN) intermediate dimension is 2048 respectively. All our models use 80 dim. log Mel features, 4 attention heads, 6 decoder layers and 12 encoder layers. Swish activation [144] is used in FFN and the convolution modules [145] in the encoder has a convolution kernel of size 31. Decoder FFN intermediate dimension is 2048 and dropout rate is 0.1. Below the encoder is a convolutional subsampling module that receives log Mel feature inputs, and consists of a ReLU [146], two 2D convolution layers and a linear layer. A 2×2 stride and a 3×3 kernel is used in each convolution layer, effectively subsampling the input log Mel features by a factor of 4, and the convolution layer's channel size is the same as the model feature hidden dimension.

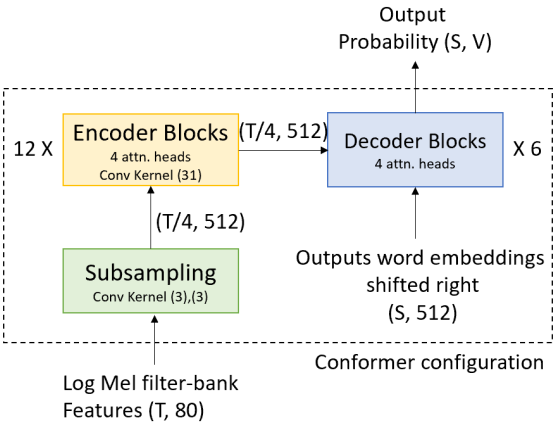

FIGURE 3.1: A brief overview of our Conformer setup. T is the input audio length. V is the vocabulary size. S is the text token length.

3.4.2 Dataset and pretraining configuration

The Conformer model is pretrained on the train-clean-100 subset of LibriSpeech dataset. During pretraining, SpecAug [147] is applied and set to $mF = 2$, $F = 27$, $mT = 5$ and $p = 0.05$. Our dataset is augmented by speed perturbation with scaling factors of 0.9, 1.0, and 1.1. A joint CTC-attention loss function is employed with a CTC loss weight of 0.3. A label smoothing weight [148] of 0.1 is used in the attention loss, and the Adam optimizer [149] is used with a learning rate of 0.0025. For the learning rate scheduling scheme, we follow ESPnet [150] implementation which has a warm-up and a decay stage. The warm-up stage has 25000 steps and

the model is trained for 50 epochs. The final model is selected by averaging 10 checkpoints with the highest validation accuracy.

3.4.3 Dataset and adaptation training configuration

The LibriSpeech-pretrained Conformer model is adapted on the IMDA2 dataset. During adaptation, we freeze the first 11 encoder layers and adapt only the last encoder layer, the CTC decoder and the attention decoder in the Conformer model. We also reduce the learning rate to 0.000025 and the warmup steps to 1K, and train for 400 epochs.

3.4.4 Results and discussions

Table 3.3 shows the results of the baseline adaptation method for the Conformer model. Before adapting the pretrained Conformer model, the model gives WER of 57.6% on IMDA2 test set, which shows that more than half of the predicted transcripts are wrong. After adapting the model to 10 hours of IMDA2 train set, the WER is reduced significantly from 57.6% to 10.3%, giving a relative 86% improvement.

The results show that the pretrained model performance on rare words dataset can be significantly improved by adapting on as few as 10 hours of training data.

TABLE 3.3: Effects of adapting a LibriSpeech-pretrained Conformer model on the 10 hours train set from the road name dataset IMDA2. WER, substituion (sub.), insertion (ins.) and deletion (del.) errors are reported. "None" indicates that no dataset is used for adapted.

Dataset	Total # of sentences	WER (%)	Error (%)		
			sub.	ins.	del.
None	0	57.6	44.0	5.5	8.2
10h train set	13000	10.3	8.2	0.8	1.3

3.5 Whisper Adaptation

This chapter will discuss baseline adaptation methods based on the Whisper ASR model [14].

3.5.1 Network configurations for Whisper models

The network configurations of Whisper is shown in Figure 3.2. We use the "whisper-small" model variant with a parameter size of $244M$. The encoder processes the audio features input with a small stem consisting of two convolution layers with a filter width of 3 and the GELU activation function [151] where the second convolution layer has a stride of two. Sinusoidal position embeddings are then added to the output of the stem after which the encoder Transformer blocks are applied. The transformer uses pre-activation residual blocks [152], and a final layer normalization is applied to the encoder output. The decoder uses learned position embeddings and tied input-output token representations [153]. The encoder and decoder have the same width 768 and 12 transformer blocks.

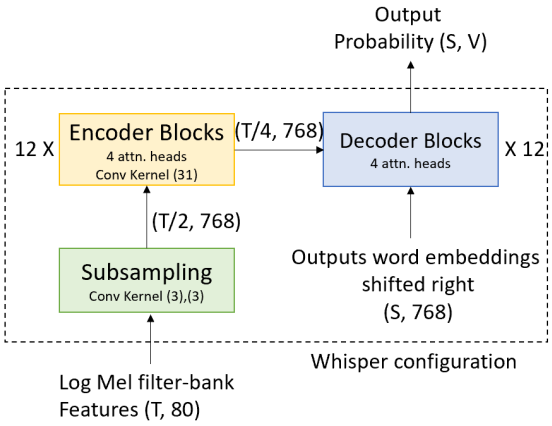

FIGURE 3.2: A brief overview of our Whisper setup. T is the input audio length. V is the vocabulary size. S is the text token length.

3.5.2 Dataset and pretraining configuration

Whisper is trained on 680K hours of audios sourced from the internet. It is trained with data parallelism across accelerators using FP16 with dynamic loss scaling and activation checkpointing [154, 155]. Models were trained with AdamW [156]

and gradient norm clipping [157] with a linear learning rate decay to zero after a warmup over the first 2048 updates. A batch size of 256 segments was used, and the models are trained for 220 updates which is between two and three passes over the datset.

3.5.3 Dataset and adaptation training configuration

The Whisper ASR model is adapted on the Librispeech and AISHELL-1 dataset separately. During adaptation, SpecAug [147] is applied and set to $mF = 2$, $F = 27$, $mT = 5$ and $p = 0.05$. Our dataset is augmented by speed perturbation with scaling factors of 0.9, 1.0, and 1.1. The model is optimized on only the attention loss function but not the CTC loss. A label smoothing weight [148] of 0.1 is used in the attention loss, and the AdamW optimizer [156] is used with a learning rate of $1e^{-5}$. For the learning rate scheduling scheme, we follow ESPnet [150] implementation which has a warm-up and a decay stage. The warm-up stage has 1500 steps and the model is trained for 50 epochs. The final model is selected by averaging 10 checkpoints with the highest validation accuracy.

3.5.4 Results and Discussions

Table 3.4 shows the results of the baseline adaptation method for the Whisper ASR model. Before adapting Whisper, the model gives a CER of 13.5% on the AISHELL-1 testset and a WER of 3.3% on LibriSpeech test-clean. After adapting the model to AISHELL-1 train set, the CER on the AISHELL-1 testset is reduced significantly from 13.5% to 4.8%, giving a relative 64.4% improvement. However, the WER on LibriSpeech test-clean has significantly increased from 3.3% to 88.1%. It shows that there is catastrophic forgetting of the English transcription ability for Whisper after adapting it to Chinese-only dataset.

On the other hand, before adaptating Whisper, the model gives a WER of 3.3% on LibriSpeech test-clean and a WER of 7.4% on LibriSpeech test-other. Unexpectely, after adapting the model on Librispeech train-clean-100, there is a slight degradation of WER from 3.3% to 3.4% on test-clean, and from 7.4% to 7.6% on test-other. It shows that further adapting Whisper to 100 hours of LibriSpeech train set does not help to improve its performance on the same domain test set.

We hypothesize that as Whisper is already very strong on transcribing audio within the LibriSpeech domain, its learning capability for LibriSpeech has almost satuated. Therefore, simply adapting it further on a small amount of data will not have much effect.

TABLE 3.4: Effects of adapting the Whisper ASR model on LibriSpeech test-clean-100 train subset or AISHELL-1 train set. "None" indicates that no dataset is used for adapted.

Dataset	Model Size (M)	AISHELL-1 (CER%)	LibriSpeech (WER%) test-clean	test-other
None	244M	13.5	3.3	7.4
LibriSpeech	244M	10.3	3.4	7.6
AISHELL-1	244M	4.8	88.1	-

3.6 Summary of chapter

In this chapter, we describe the baseline methods we use to reach state-of-the-art performance for Whisper and Conformer models ASR adaptation tasks. For Conformer adaptation on IMDA2 dataset, the WER is reduced significantly from 57.6% to 10.3% on the IMDA2 test-set, giving a relative 86% improvement. For Whisper adaptation on AISHELL-1 dataset, the CER is reduced significantly from 13.5% to 4.8% on the AISHELL-1 test-set, giving a relative 64.4% improvement. However, For Whisper adaptation on Librispeech data, the WER on the corresponding test-set has is slight increase 3.3% to 3.4% on the Librispeech test-clean test-set. It suggested that adaptation is more effective when the model is initially no performing well on a target domain, e.g. WER larger than 10%. However, it is less effective if the model is already performing well, e.g. WER less than 5%.

Chapter 4

ASR Adaptation using layer-wise freezing and TTS synthesized data

4.1 Overview of chapter

In this chapter, two novel approaches will be discussed to improve the state-of-the-art performance for TTS synthesized data adaptation and layer-wise adaptation.

TTS synthesized data adaptation: Section 4.2 of this chapter will discuss using multiple TTS systems to synthesize training data for rare word ASR adaptation. The section will first discuss the issue of TTS data diversity and motivate the use of multiple TTS systems. Then, it will give a literature review on the related works for TTS synthesized data adapatation. Next, it will proceed to discuss the proposed method and show and analyze the experiment results.

Layer-wise adaptation: Section 4.3 of this chapter will discuss the use of per-layer loss to automatically select the layers to adapt in Layer-wise adaptation. The section will discuss the issue of the expensive cost in finding the optimal layers to adapt and motivate the use of an automatic search strategy. Then, it will give a literature review on the related works for Layer-wise adaptation. Next, it will proceed to discuss the proposed method and show and analyze the experiment results.

4.2 ASR Model Adaptation for Rare Words Using Synthetic Data Generated by Multiple Text-To-Speech Systems

Abstract: Automatic speech recognition (ASR) for rare words is difficult as there are little relevant text-audio data pairs to train an ASR model. To obtain more text-audio pairs, text-only data are fed to Text-To-Speech (TTS) systems to generate synthetic audio. Previous works use a single TTS system conditioned on multiple speakers to produce different speaker voices to improve the output data's speaker diversity, and they show that training an ASR model on the more diverse data can avoid overfitting and improve the model's robustness.

As an alternative way to improve the diversity, we study the speaker embedding distribution of audios synthesized by different TTS systems and found that the audios synthesized by different TTS systems have different speaker distributions even when they are conditioned on the same speaker. Inspired by this, this paper proposes to condition multiple TTS systems repeatedly on a single speaker to synthesize more diverse speaker data, so ASR models can be trained more robustly.

When we apply our method to a rare word dataset partitioned from National Speech Corpus SG, which contains mostly road names and addresses in its text transcripts, experiments show that a pretrained ASR model adapted to our multi-TTS-same-SPK data gives relatively 9.8% lower word error rate (WER) compared to the ASR models adapted to same-TTS-multi-SPK data of the same data size, and our overall adaptation improves the model's WER from 57.6% to 16.5% without using any real audio as training data.

4.2.1 Introduction

Rare words pose an ongoing challenge in the development of high-quality automatic speech recognition (ASR) systems [158]. These rare words, often associated with named entities like names and locations, play a vital role in preserving the intended meaning within the decoded transcript. Due to their limited occurrence in the audio-text pairs forming the training set of an ASR system, correctly predicting these "tail" words becomes inherently difficult.

One way to increase the number of text-audio training data pairs containing these rare words is to generate synthetic data using a Text-To-Speech (TTS) system. This method requires only text data as inputs to a TTS system, and does not require any change in the parameter or architecture of ASR models. Although there are many other methods like language model integration [159] and speech augmentation methods [147] [160] that can achieve a similar goal, [161] shows that the performance improvements are mostly independent.

As improving the speaker diversity of the synthetic training data can improve an ASR model's robustness [46] [47], previous works use a single TTS system conditioned on multiple speakers to produce different speaker voices. Alternatively, we found that using multiple TTS systems conditioned on a single speaker can also produce slightly different speaker voices. This is in line with several studies [162] [163], which shows that there can be noticeable differences in the real and TTS synthesized speaker voices. An analysis of the speaker embedding distribution of audios synthesized by conditioning different TTS systems on the same speaker is shown in Section 4.2.8.

Inspired by this, we propose to synthesize data by repeatedly conditioning multiple TTS systems on the same speaker to synthesize more diverse speaker voices. Compared to methods that use a single TTS system to improve the speaker diversity by either conditioning the TTS system on multiple speakers or by tuning its parameters to control the speaker characteristics [48] of the synthesized audio, our method is less restricted by the trade-off [49] between the data's diversity and naturalness, i.e. improving the diversity of the synthesized audios costs less degradation in their quality. Also, our method can be applied to single-speaker TTS as the diversity can be compensated by the use of multiple TTS.

Our contributions are fourfold:

1. We explore the use of multiple TTS systems to synthesize training data to perform ASR adaptation.

2. We show that to generate diverse synthetic data, using multiple TTS systems conditioned on the same speaker is more effective than using a single TTS system conditioned on multiple speakers.

3. We show that freezing part of an ASR model when adapting to synthetic data gives better result than adapting the whole ASR model.

4. We provide baseline results on a high quality, publicly available rare word dataset, which consists of 13K utterances containing road names and addresses from National Corpus SG.

4.2.2 Related Works

Several directions have been explored for improving rare word recognition. The Contextual Listen, Attend, and Spell (CLAS) model [164] represents an end-to-end ASR system that builds upon the foundational Listen, Attend and Spell (LAS) model initially proposed by Chan et al. [1]. A notable distinguishing characteristic of the CLAS model is its incorporation of contextual information in an entirely end-to-end fashion, setting it apart from the LAS model. In [165], phonetic fuzzing of proper nouns are used to find hard negative examples for training CLAS. These methods requires real text-audio pairs during training.

Several techniques requires text-only data to bias ASR model outputs to rare words. They include shallow fusion [166–168], cold fusion [169], and deep fusion [170] of an external language model. Other techniques focus on modifying the training data by either data augmentation [147] [160] or TTS data generation [171] [172] [173] [174]. Our work aims to synthesize training data of diverse speaker voices for rare word ASR.

4.2.2.1 Diverse TTS Synthetic Data Generation

Numerous efforts are made to improve a TTS systems ability to synthesize data with diverse speaker voices. [46] uses a Tacotron 2 model through conditioning the network on multiple speaker embeddings. [161] investigates the use of different types of speaker embeddings for conditioning the TTS. [47] increases the speaker diversity through conditioning a Tacotron 2 model on randomly chosen virtual speakers sampled from a latent space.

All the above mentioned works, however, focuses on using just a single TTS system to improve the speaker diversity of the synthetic data. In our work, we show that

an ASR model's robustness can be further improved by increasing the speaker diversity of the synthetic training data using multiple TTS systems. To the best of our knowledge, we are the first to leverage multiple TTS systems to improve the speaker diversity of the synthetic data for ASR adaptation.

4.2.3 Methodology

We now introduce our method to improve the speaker diversity of TTS synthetic speech. Specifically given a set of N transcripts $\{u_1, \ldots, u_N\}$, we first feed each transcript u_n to a TTS system M_1 and condition M_1 on a randomly sampled speaker s_i, where $i \in \{1, \ldots, K\}$ and K is the number of speakers. We then synthesize another audio for u_n by conditioning a different TTS system M_2 on the same speaker s_i. We call our approach multi-TTS-same-SPK as we use multiple TTS conditioned on the same speaker to generate audio for each transcript. In contrary, we call the approach used by previous works to synthesize audio as same-TTS-multi-SPK, as they use the same TTS conditioned on multiple different speakers for each transcript.

4.2.3.1 Multiple TTS Systems

We use two state-of-the-art TTS systems of different architecture to generate more diverse data. The first TTS system uses TransformerTTS [175] as the acoustic model and HiFi-GAN [176] as the vocoder. The second TTS system is VITS [177].

4.2.3.2 TransformerTTS and HiFi-GAN

TransformerTTS is an autoregressive TTS that utilizes transformers with multi-head attentions in an encoder-attention-decoder based architecture. It is the acoustic model in the TTS system which generates mel-spectrogram frames sequentially from textual input. The generated mel-spectrogram is then passed into HiFi-GAN, a GAN based vocoder to generate high-fidelity audio waveform by upsampling the input through a series of transpose convolutions with a Multi-Receptive Field Fusion module following each transpose convolution to process patterns of different lengths in parallel.

4.2.3.3 VITS

VITS is a fully end-to-end TTS that utilizes variational inference augmented with normalizing flows and an adversarial training phase to capture high-fidelity speech from textual input. VITS composes of a prior encoder for both text encoding and normalizing flow, a posterior encoder for spectrogram encoding, a duration predictor, a HiFi-GAN generator for decoding and a multi-period discriminator from HiFi-GAN.

4.2.3.4 Speaker Conditioning

To enable multi-speakers capability for the TTS systems, speaker embedding vectors that capture speaker style information are added into TransformerTTS and VITS so that the TTS systems can use the information to generate audios that sound similar to the target speaker.

4.2.4 Experimental Setup

4.2.4.1 TTS model configuration

All the TTS models we use are trained on VCTK [178], a benchmark dataset for TTS containing about 44 hours of speech data from 109 native English speakers with various accents. The ESPnet2-TTS toolkit [179] is used for all TransformerTTS and HiFi-GAN experiments, while the Coqui-ai TTS toolkit [180] is used for all VITS experiments. All models uses 80-dimensional mel-spectrograms. Also, all input text are converted to phonemes before feeding into the models. As the TransformerTTS and VITS TTS systems originally generates waveforms at a sampling rate of 22050 Hz and 24000 Hz respectively, and the ASR model is trained to receive only 16000 Hz audio as input, down sampling of the TTS synthesized audio is required before they are fed into the ASR model. The process may cause the loss of high frequency acoustic information in the audio. However, we hypothesize that such information loss has little effect for ASR adaptation as a sampling rate of 16000 Hz is sufficient to cover most of the human voice frequency range [181] and keep most of the needed information for ASR. Future works are needed to study the effect of down sampling on ASR adaptation performance.

The TransformerTTS model and HiFi-GAN model we use is pretrained and released by ESPnet2-TTS [179]. The FFT, window, and hop size for both models were set to 2048, 1200, and 300, respectively. The TransformerTTS model uses 6 layers of transformer blocks in both its encoder and decoder with 8 heads for multi-head attention. The decoder prenet has 2 linear layers with 256 hidden units, and the postnet has 5 CNN layers with channel and filter size 256 and 5 respectively. The model was trained using the Adam optimizer [149] with the addition of a guided attention loss to help learning diagonal attentions [182]. The HiFi-GAN generator uses hidden dimension size of 512, kernel sizes of [10, 10, 8, 6] for the transposed convolutions, kernel sizes of [3, 7, 11] and dilation rates of [[1, 1], [3, 1], [5, 1]] $\times$ 3] for the MRF modules. The HiFi-GAN model was trained using the Adam optimizer with a learning rate of 0.0002.

The VITS model we use is pretrained and released by Coqui-ai TTS. The model's configurations follow the original work [177], with the FFT, window, and hop size set to 2048, 1024, and 256, respectively. The model was trained with the AdamW optimizer [156] with $\beta 1 = 0.8$, $\beta 2 = 0.99$, weight decay $\gamma = 0.01$ and learning rate of 0.001. A windowed generator training is used with a window size of 32 to improve efficiency and lowering memory consumption during training.

4.2.4.2 TTS data synthesis

The TTS models are used to synthesize additional training data for ASR adaptation based on the rare word road name dataset described in Section 3.2.

Given the 13K text-only transcripts we obtained from the real train set, we first create a mapping named SPKMAP1 to map the text transcription of every utterance in the train set to a randomly sampled speaker in the VCTK corpus. We then obtain another mapping SPKMAP2 similarly, such that the mapped speaker for every text in SPKMAP2 is different from the mapped speaker in SPKMAP1. After creating SPKMAP1 and SPKMAP2, we generate 4 sets of synthetic audio data sets by varying the 2 speaker mappings SPKMAP1 and SPKMAP2 and varying the 2 TTS systems VITS and TransformerTTS systems used and name them VITS-SPKSET1, VITS-SPKSET2, TRANS-SPKSET1 and TRANS-SPKSET2.

Figure 4.1 shows the step-by-step procedure to obtain VITS-SPKSET1. In step 1, we first use SPKMAP1 to obtain the VCTK speaker id that is mapped to a text

transcription in the train set. In step 2, we extract the speaker embedding vector that correspond to the VCTK speaker id obtained in step 1. In step 3, 4 and 5, we feed the text transcription (used in step 1) and the speaker embedding vector (extracted in step 2) into our VITS model to generate an synthetic audio file. We repeat step 1-5 for all text transcriptions in the train set, and the synthetic audio files obtained formed VITS-SPKSET1. To obtain VITS-SPKSET2, we simply replace SPKMAP1 with SPKMAP2, and repeat the pipeline just described. To obtain TRANS-SPKSET1 and TRANS-SPKSET2, we repeat the same steps using our TransformerTTS+HiFi-GAN TTS system in replacement of our VITS TTS system. Table 4.1 summarizes the basic information of all our datasets.

TABLE 4.1: Dataset overview

Dataset	Dataset Type	Total # of words	Total # of sentences	Total # of speakers	Total audio duration (h)
Real train set	real	115119	13000	186	17.4
VITS-SPKSET1	fake	115119	13000	108	11.6
VITS-SPKSET2	fake	115119	13000	108	11.6
TRANS-SPKSET1	fake	115119	13000	108	10.1
TRANS-SPKSET2	fake	115119	13000	108	10.2
Test set	real	11015	1176	76	1.63

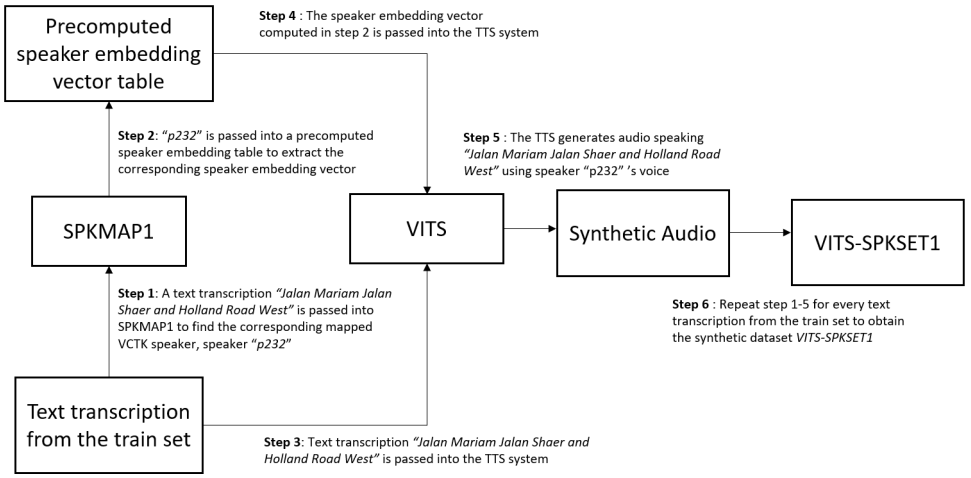

FIGURE 4.1: Audio generation pipeline to obtain synthetic audio dataset *VITS-SPKSET1*. A similar pipline is also used to obtain synthetic audio dataset *VITS-SPKSET2*, *TRANS-SPKSET1* and *TRANS-SPKSET2*.

4.2.4.3 ASR adaptation

The ASR models ar pretrained as described in Section 3.4.2. The adaptation configuration is similar to Section 3.4.3 except that we use different training data. Firstly, we adapt the pretrained model to VITS-SPKSET1 (same-TTS-same-SPK) and TRANS-SPKSET1 (same-TTS-same-SPK) separately, to obtain 2 baseline ASR models. We then adapt the pretrained model to VITS-SPKSET1 + VITS-SPKSET2 (same-TTS-multi-SPK), and TRANS-SPKSET1 + TRANS-SPKSET2 (same-TTS-multi-SPK) respectively to form our second set of baselines. For our method, we adapt the pretrained model to VITS-SPKSET1 + TRANS-SPKSET1 (multi-TTS-same-SPK). Lastly we adapt the pretrained model to the real train set (ground-truth dataset) to find the performance upperbound.

4.2.5 Results and Discussion

Table 4.2 shows that our model adapted to multi-TTS-same-SPK synthesized data outperforms all 4 baseline models adapted to same-TTS-same-SPK or same-TTS-multi-SPK synthesized data. Our model improves relative WER by 9.8% over the best baseline TRANS-SPKSET1+TRANS-SPKSET2, and improves relative WER by 71.4% over an unadapted model, showing the effectiveness of our adaptation method. However, there are still gaps between models adapted to real and synthesized data. Also, if we compare models adapted to single TTS data only, TransformerTTS performs better than VITS in all cases.

4.2.6 Discussion on why the multi-TTS-same-SPK approach outperform the same-TTS-multi-SPK approach in the experiments

As TTS synthesized data may contain artifacts that does not exist in the real data, we hypothesis that one factor contributing to the speaker diversity is the unique artifact pattern that exist in each TTS system. We further hypothesize that by adapting the ASR model to multiple TTS output distributions, it is less likely for the ASR model to over-fit to a specific TTS output distribution and therefore retaining the ASR model's capacity on processing real data.

TABLE 4.2: Effects of adapting a pretrained model on different combination of synthesized data. WER, substituion (sub.), insertion (ins.) and deletion (del.) errors are reported.

Data Combination	Data Type	Total # of sentences	WER (%)	Error (%) sub.	ins.	del.
None	n/a	0	57.6	44.0	5.5	8.2
Real train set	Real	13000	10.3	8.2	0.8	1.3
Our baselines						
VITS-SPKSET1	Fake	13000	21.1	16.2	1.1	3.8
TRANS-SPKSET1	Fake	13000	22.0	16.6	1.3	4.1
VITS-SPKSET1 +VITS-SPKSET2	Fake	26000	19.2	14.9	0.9	3.3
TRANS-SPKSET1 +TRANS-SPKSET2	Fake	26000	18.3	13.4	1.2	3.7
Our works						
VITS-SPKSET1 +TRANS-SPKSET1	Fake	26000	**16.5**	**12.6**	1.0	**3.0**

TABLE 4.3: Effects of adapting only specific parts of the pretrained ASR model on VITS-SPKSET1. WER, substituion (sub.), insertion (ins.) and deletion (del.) errors are reported on Aishell-1 dataset.

Encoder parts adapted	Decoders adapted	WER (%)	Error (%) sub.	ins.	del.
Unadapted					
-	-	57.6	44.0	5.5	8.2
Adapt encoder					
layer 1	-	62.5	47.3	3.7	11.6
layer 12	-	39.4	31.1	2.1	6.2
Adapt decoder					
-	CTC	62.5	46.8	4.1	11.6
-	attention-AR	43.1	32.3	2.8	8.0
-	CTC + attention-AR	27.5	20.8	1.9	4.9
Adapt encoder and decoder					
all	all	23.0	18.1	0.9	4.1
layer 12	attention-AR	21.7	16.8	1.2	3.8
layer 12	CTC + attention-AR	**21.1**	**16.2**	**1.1**	**3.8**

4.2.7 Freeze model weights

Previous studies show that the top [33] or bottom [183] layers can be adapted to improve an ASR model performance in new domain data. Table 4.3 shows the

result of adapting only the specific parts of the pretrained ASR model on VITS-SPKSET1. Compared to the unadapted results, adapting the last encoder layer improves relative WER by 31.6% and gives better result than adapting the first encoder layer. On the other hand, adapting both the CTC and attention-AR decoders improves relative WER by 52.3% and is the best adaptation result if the encoder is not adapted. Based on the results, we adapt the last encoder layer, the CTC and attention-AR decoders together and acheives the best WER of 21.1%, which improves relative WER 63.4% over the unadapted results, and improves relative WER by 8.3% over the results of adapting the whole ASR model.

4.2.8 Analysis on whether increasing the number of TTS systems can improve speaker diversity

Previous works have shown that conditioning a multi-speaker TTS system on a target speaker may produce synthesized speaker voice that sounds slightly different from the target speaker's ground-truth audio [162] [163]. We therefore hypothesize that a gap in speaker similarity may also exist between synthetic audios generated by different TTS systems conditioned on the same target speaker.

To investigate this hypothesis, we extract X-vector speaker embeddings from the ground-truth and synthesized audios and visualize the results using UMAP [184] in Figure 4.2. The structure of Figure 4.2 is as follows: It contains 4 sub-figures where each sub-figure shows the speaker embeddings of a target speaker. Within each sub-figure, there are 3 groups of speaker embeddings which corresponds to the ground-truth audio, the VITS TTS synthesized audio and the TransformerTTS synthesized audio respectively. Within each of the 3 groups, the speaker embeddings are extracted from audios speaking different transcripts.

From the results, we see that distinct clusters are formed from the ground-truth and synthetic audios. This shows that the speaker embeddings extracted from synthesized audios containing different transcripts are relatively close to each other if the audios are produced by the same TTS system conditioned on the same speaker, but the speaker embeddings are relatively far from each other if the audios are produced by different TTS systems conditioned on the same speaker.

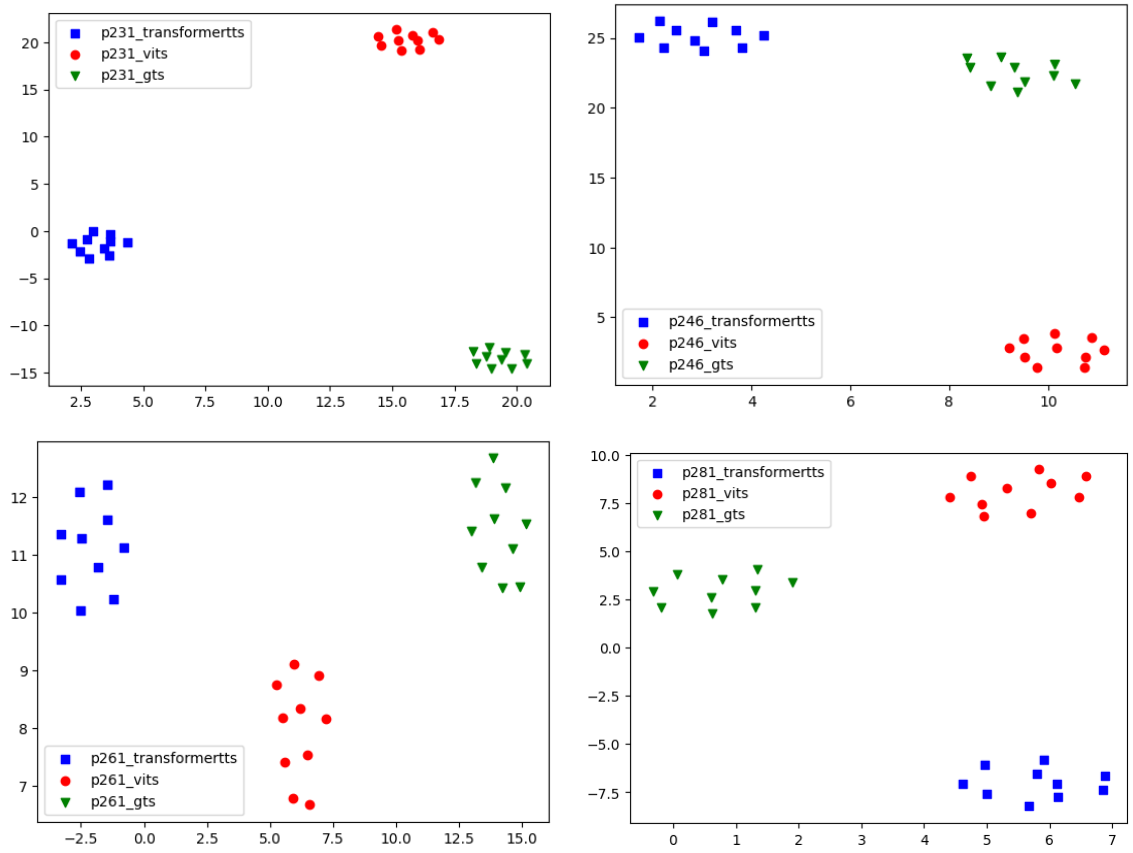

FIGURE 4.2: UMAP plots of speaker embeddings extracted from ground-truth audios (gts), VITS TTS synthesized audios (vits) and TransformerTTS synthesized audios (transformertts). Best viewed in color.

4.3 Layerwise Adaptation by using per-layer loss for automatic layer selection

Abstract: While most contemporary neural networks are trained using academic datasets , their efficacy can decline when applied to real-world scenarios, primarily due to the domain mismatch between the training and testing data. To solve the problem, neural network adaptation is used to enhance model performance within the specific domain of the test data. Nonetheless, adaptation introduces significant challenges, notably overfitting, which jeopardizes the model's ability to generalize to unseen data, and catastrophic forgetting, which leads to the loss of previously acquired knowledge during pre-training, ultimately resulting in diminished performance of the adapted models.

One approach to mitigate the issues is layerwise adaptation. It involves the selective freezing of specific layer weights that are susceptible to overfitting and catastrophic forgetting, thereby mitigating these issues. Simultaneously, other layer weights are allowed to be updated to enhance the model's performance within the target domain.

However, determining the ideal layers to freeze presents a significant challenge. Previous approaches have depended on either heuristics or a trial-and-error process to manually select the layers for freezing. Consequently, this process still demands expert knowledge and substantial computational resources to identify the most effective layers to freeze.

To address this challenge, we introduced an automated search approach for identifying the optimal set of layers to be frozen. This method only applies to residual networks. Specifically, it will estimate the performance of each residual network layer based on their skip connections. If the performance degrades during adaptation, the approach will proceed to perform a layerwise early stopping to freeze the respective layers.

Experiment results show that our method can effectively regularize adaptation through layerwise weight freezing without the requirement of any expert knowledge and expensive computation powers to find the optimal layers. Our method can reduce overfitting and improves relative CER by 5.7% on a Chinese Datatang-ZH test-set when we adapt Whisper on a Chinese AISHELL-1 train set. Also, our model can reduce catastrophic forgetting and improves relative WER by 16.4% on the English Librispeech test-set when we adapt Whisper on the Chinese AISHELL-1 train set.

4.3.1 Introduction

Nowadays, most of the neural networks are trained with academic datasets [77, 185]. However, their performance may experience a decline when subjected to real-world scenarios, primarily due to the domain mismatch between the training and testing data [18]. Neural network adaptation [19–22] emerges as an essential technique for enhancing a model's performance within the context of the testing data domain.

Nevertheless, adaptation introduces prominent challenges, notably overfitting [23] and catastrophic forgetting [24].

Precisely, overfitting [186] denotes a situation in which a model becomes over-tuned to the training data during the adaptation process. In this scenario, the model starts to capture not only the fundamental underlying patterns but also incorporates extraneous noise and random variations inherently present in the data.

On the other hand, catastrophic forgetting [187] refers to a situation where an ASR model firstly trained on dataset A loses its ability to accurately transcribe speech on dataset A after it is further trained or adapted on dataset B.

In either case, they may cause the adapted model performance to degrade on unseen data.

To mitigate the issues, different regualized adaptation approaches are developed. For example, one line of research incorporate an auxiliary loss into the training loss to guide the optimization process. They include L2 regularization [82], Kullback Leibler divergence [83] and adversarial multitask learning [84]. Another line of research performs data balancing [78–80] by uniformly sampling data samples from multiple domain sources, to ensure the adapted model is not biased to domains with relatively more data samples. Alternatively, [81] applies a separate scalar learning rate for each data domain to prioritize learning for domain with scarce data.

A third line of research is partial weight freezing. Some examples include [29–32] that adapt certain layers or subset of parameters. One of the approach is laye-wise adaptation. By not updating part of the layer weights that are most prone to overfitting and catastrophic forgetting, the issues can be mitigated while still allowing another part of the layers weights to be updated to improve the model's performance on the target domain.

However, identifying the ideal set of layers for freezing presents a challenging task. Prior research primarily depends on heuristic methods [29, 82] or trial-and-error [31] approaches to manually select the layers for freezing. Consequently, this process demands expert knowledge and substantial computational resources to determine the optimal layers.

To address this challenge, we introduced an automated search strategy aimed at identifying the most suitable layers to freeze. It's important to note that this

strategy is specifically designed for use with residual networks [188]. Our approach capitalizes on the observation that gradients computed from the training loss function are directly propagated back through the skip connections within each layer of the residual network, i.e. the gradients at the loss function do not change as it is propagated to the layers. Consequently, we can compute an estimate of each layer's performance directly based on these skip connections. We have devised three distinct strategies that utilize the layers' performance as a conditioning factor for selecting which layers to freeze.

First, we conduct two sets of training, which we shall refer to as the trial and actual training phases. In both scenarios, the training and validation datasets are drawn from the target domain. During the trial training phase, we train all layers of a model, while simultaneously estimating each layer's performance based on the validation dataset. Subsequently, in the actual training phase, we consider freezing a layer if it exhibits performance fluctuations during the trial training phase. This process aims to identify layers that are susceptible to overfitting.

In the second strategy, we examines the potential elimination of the trial training phase by directly early stopping a layer during the actual training phase if its performance deteriorates within that very phase.

In the third strategy, rather than assessing a layer's performance using the validation dataset from the target domain, we substitute the dataset with the data on which the model was initially pretrained. In this approach, we early stop a layer if its performance declines on the pretrained data during the actual training. This is to identify layers susceptible to catastrophic forgetting.

Our contributions are fourfold:

1. We show a method to estimate layer-wise performance for residual networks by using each layer's skip connection and analyze the results.

2. We show that if a layer's performance is found to fluctuate in the trial training, freezing the layer in the actual training can reduce overfitting.

3. We show that if a layer's performance is found to degrade in the actual training, early stopping the layer in the actual training can reduce overfitting.

4. We show that if a layer's performance on the pretrained data domain is found to degrade, early stopping the layer in the actual training can reduce catastrophic forgetting.

4.3.2 Related Works

Our work is under the category of partial weight freezing, which is the practice of not updating, effectively "freezing" specific weights or components within an ASR model during the adaptation process. These frozen weights remain constant and do not undergo any changes throughout the training process.

Numerous efforts are made on improving adaptation efficiency and robustness by partial weight freezing. [30] adapts the hidden activation functions of a hybrid ASR model [189]. [32] adapts only the bias terms of a Transformer-based Masked Language-model [190] and shows competitive results compared with fine-tuning the entire model in low-resource tasks.

Besides hidden activation function or bias terms adaptations, weights can also be freezed layerwise. [29, 82] adapts the last layer of an ASR model. [31] adapts each layer in a model separately and reported that adapting the first layer gives the best results. [82] adapts the first and the last layer of a hybrid ASR model.

Among all the weight freezing methods, we argue that layerwise adaptation is more interpretable. This is because it is more intuitive to adapt/freeze a layer where each layer is found to contribute more to a different subtask. For example, one may prefer adapting the first layer of an ASR model for acoustic domain adaptation [191], as the layer is found to contribute more to the acoustic modelling task [39].

However, previous works relies on heuristics [29, 82] or trial-and-error [31] method to select the layer weights to freeze. This requires expert knowledge and intensive computation to search for the optimal set of layers. As such, we propose an automatic strategy to search for the optimal layers to reduce the above needs.

As far as we know, our method is the first to perform automatic search of optimal layers in layerwise adaptation using layer-wise performance. Previous works [192] progressively freezes partial layers with the highest correlation ratios for each adaptation task to improve training computation efficiency and memory efficiency by

comparing the gradient projection norm between the current task and prior tasks. Other works have investigated the different convergence speed exhibited in different layers[193], or calculates a layerwise performance metric by introducing multiple classifiers on top of multiple layers to improve a model's modelling capacity [194]. [195] performs gradient descent with respect to a meta-learned distance metric, which warps the activation space to be more sensitive to a target task identity. However, all the previous methods does not study the contribution of each layer to ASR adaptation by directly calculating a layerwise performance metric.

On the other hand, pruning based methods are used to zero-out or remove partial model weights during adaptation. [196] uses pruning to zero out partial weights during domain adaptation. Another work focus on automatically pruning a subset of model weights to reduce model size. [197] applies uniform quantization and unstructured pruning methods to both the weights and activations of deep neural networks during training. [198] proposes a tractable heuristic for pruning in which they select weights for simultaneous removal and combine it with a single-pass systematic update of unpruned weights.

The three above methods automatically select the weights to be pruned or zero out if they are insignificant to model performance, which is determined before training starts. In contrast, our method automatically select the layerwise weights to be frozen base on the training dynamics, i.e. if the weights start to overfit during training. In addition, our method is more intuitive in solving overfitting or catastrophic forgetting problem in that it records and reacts to layer performance degradation, which is a direct sign of the two problems.

4.3.3 Methodology

Our method is an automatic search strategy to find the optimal set of layers to freeze during layerwise adaptation. Specifically, the strategy only works for residual networks. Our strategy relies on the fact that the gradients computed from the final loss function is directly back-propagated to the skip connections in each of the residual network layers. As a result, an estimate of each layer's performance can be directly computed from their skip connections. We develop three strategies that condition on the layers performance for layer freezing selection.

In strategy S_A, we conduct two sets of training which are referred as the trail and actual training. In both settings, the training and validation dataset are in the target domain. In the trail training, a model is trained with all the layers and each layer's performance is estimated on the validation dataset. Then, we explore to freeze a layer in the actual training if the layer's performance is found to fluctuate in the trial training. This is to identify the layers that are prone to overfitting.

In strategy S_B, we explore the possibility of removing the need of a trial training by freezing a layer in the actual training if the layer's performance is found to degrade in the actual training. This is to identify the layers that are prone to overfitting.

In strategy S_C, instead of estimating a layer's performance on the validation dataset in the target domain, the validation dataset is replaced by the data that the model is pretrained on. Then, we freeze a layer in the actual training if the layer's performance is found to degrade on the pretrained data. This is to identify the layers that are prone to catastrophic forgetting.

4.3.3.1 Layer performance in residual network

The following discuss how we estimate layerwise performance of a Residual network. Recall the key component of a Residual network is formulated as:

$$x_l = H(x_{l-1}) + x_{l-1} \tag{4.1}$$

where for each layer $l-1$ in the network, its output x_{l-1} is fed to a residual function $H(.)$, and the result is added back to x_{l-1} to form the output x_l of the next layer l.

In the above equation, a skip connection exists from layer $l-1$ to layer l. To study the relationship between layers that are separated by one or more layer, one may exploit the recursion pattern exists in Equation 4.1:

$$x_l = H(x_{l-1}) + H(x_{l-2}) + \ldots + H(x_{l-j}) + x_j \tag{4.2}$$

where $0 < j < l$. The equation shows that there exists a skip connection that links any layer before layer l to layer l.

Given that a loss function is computed on the last layers output of a residual network during model training. As it has been shown in Equation 4.2 that each of the

residual network layer outputs are merged to form the final output through addition, the gradients computed from the loss function is directly back-propagated to the skip connections in each layer, i.e. the gradients at the loss function does not change as it is backpropagated to the layers. As each layer's output is directly optimized for the loss function, an estimate of each layer's performance can therefore be computed from their skip connections.

4.3.3.2 Early stopping

Early Stopping [199] is a regularization technique for deep neural network. It stops training of the whole network when its parameter updates no longer improve model performance on a validation set. However, we argue that early stopping the whole model may not be optimal if only part of the model contributes to the performance degradation. Therefore, we propose to estimate layerwise performance in Residual network to perform layerwise early stopping.

Specifically, our method stops updating the weights of a target layer when its performance degrades. We follow the conventional early stopping to use a patience value of τ, and early stop a layer if its performance continue to degrade for τ intermediate checkpoints.

Unlike the conventional early stopping, our method still continues to update other parts of the model even when early stopping is triggered for a target layer. As such, training of the other parts of the model needs to be resumed at a checkpoint where the target layer performance has not yet been degraded.

Therefore, after early stopping a target layer, we propose to resume training of the other parts of a model at the checkpoint where the target layer has the best layer performance among all intermediate checkpoints. This is applied to strategy S_A and S_B where a layer's performance is evaluated on the validation dataset in the target domain.

However, for strategy S_C, the validation dataset is within the domain of the pretrained dataset. The validation performance on the pretrained domain is expected to degrade if the model is adapted to a new target domain. Therefore, another parameter $0 \leq \epsilon < 1$ is defined for degradataion tolerance, where a layer will be early stopped if its layer performance has dropped more than ϵ relatively.

4.3.3.3 Encoder layer performance in AED models

Section 4.3.3.1 has shown that layer performance can be estimated for a vanilla Residual network with sequentially stacked residual blocks. However, another prominent model architecture Transformer [2] which involves the use of residual blocks follows an attention-encoder-decoder architecture (AED), where the output of the encoder is directly connected to each layer of the decoder. As a result, the encoder layers' performance cannot be estimated. This is because the gradients computed from the loss function is not directly back-propagated to the encoder layers in the sense the the gradients at the loss function has changed during the back-propagation as it reaches the layers.

To solve this, we propose to define a new loss function L_{enc} for the encoder M_{enc} in terms of the original loss function L_{orig} and the decoder M_{dec}. Following the equation formulation in Section 2.2.4.1, the training loss is calculated as:

$$h = M_{enc}(x) \tag{4.3}$$

$$P(y|x) = M_{dec}(h, y) \tag{4.4}$$

$$L_{orig}(Q(y|x), P(y|x)) = CE(Q(y|x), P(y|x)) \tag{4.5}$$

where $CE(.)$ is the cross entropy loss [200] and $Q(y|x)$ is the ground-truth probability distribution.

The new loss function is defined as:

$$L_{enc}(Q(y|x), h, y) = CE(Q(y|x), M_{dec}(h, y)) \tag{4.6}$$

By treating the decoder term as part of the loss function, the gradients computed from the new loss function can directly back-propagate to the encoder layers. A layer performance estimate for the encoder can therefore be calculated.

4.3.3.4 Grouping of layers

As training needs to be resumed from previous checkpoints every time a layer is early stopped, the training will take more time to complete. To speed up training,

one can reduce computation cost by grouping and freezing multiple layers at once. Specifically, we define ρ is the number of layers to group. Layers l to $l-(\rho-1)$ will be early stopped if the layer performance of layer l degrades.

4.3.4 Experimental Setup

For the experiments to be conducted with strategy S_A, the ASR models are pretrained as described in Section 3.4.2. The adaptation configuration is similar to Section 3.4.3. The models are adapted on VITS-SPKSET1, which is a TTS synthesized audio and real text pairs dataset as described in Table 4.1.

For the experiments to be conducted with strategy S_A and S_B, the ASR models are pretrained as described in Section 3.5.2. The adaptation configuration is similar to Section 3.5.3. For the adaptation methods, we reduce the interval to save an intermediate model checkpoint from 1 epoch to 0.05 epoch. We set τ to 10, such that model training will stop when its validation performance has not been improved for 10 checkpoints. Additionally, we freeze the positional embeddings in the decoder.

4.3.5 Results and Discussion

The following subsections show the results of our strategies. The strategies are 1) S_A: Freeze a layer in the actual training if the layer's performance is found to fluctuate in the trial training. 2) S_B: Freeze a layer in the actual training if the layer's performance is found to degrade in the actual training. S_C: Freeze a layer in the actual training if the layer's performance on the pretrained data domain is found to degrade.

4.3.5.1 Freeze layer if fluctuate in trail training (S_A)

Table 4.4 shows the result of freezing different layers of the model. The unadapted ASR model gives WER of 57.6% on the IMDA2 rare word test set. To improve its performance, TTS synthesized data is used for adaptation. However, adaptation of TTS synthesized data is prone to overfitting. To mitigate the issue, layerwise adaptation is applied for regularization.

To manually search for the optimal set of layers to adapt, the effects of adapting the first and the last layer of the encoder are investigated. The results are shown in "Baseline: Adapt encoder" of table 4.4. Compared with the unadapted model, adapting only the first encoder layer shows worse results while adapting the last encoder layer shows a relative 32% WER improvement. It suggested that adapting the last encoder layer is preferred when adapting the encoder.

Next, the effects of adapting the CTC decoder, the attention-AR decoder and adapting both the decoders are investigated. The results are shown in "Baseline: Adapt decoder" of table 4.4. The results show that adapting both the decoders gives the lowest WER of 27.5%.

Given the previous results, 3 more experiments are conducted to manually search for the best layers to adapt in both the encoder and decoder. The results are shown in "Baseline: Adapt encoder and decoder". Compared with adapting the whole model, layerwise adaption with manual search can reduce WER from 23.0% to 21.1% and achieves a relative WER improvement of 8.3%.

Finally, we compare our strategy S_A with manual search for layerwise adaptation. A trial training is first performed by adapting the whole model to obtain each encoder layer's character error rate (CER) performance as shown in Figure 4.3. According to the figure, a rough inspection shows that the CER of layer 3-10 has some fluctuations across epochs. Therefore, we follow strategy S_A to only adapt layers 1-2,11-12 in the actual training and obtain the best WER of 19.8 for the adapted model, which is relative 6.7% better than the manual search strategy.

4.3.5.2 Freeze layer if degrade in adapted domain (S_B)

Table 4.5 shows the results of adapting the Whisper-small ASR model on the AISHELL-1 train-set. The unadapted model gives CER of 13.5% on AISHELL-1 test-set and 46.2% on Datatang-ZH test-set. After adapting the whole model on the train-set without freezing weights, there is a relative CER improvement of 64.4% and 17.1% on AISHELL-1 and Datatang-ZH test sets respectively. However, we hypothesize that as the AISHELL-1 train set is relatively small, the model may be overfitted to the train set during adaptation and affect the model's performance on Datatang-ZH. Therefore, we apply strategy S_B to mitigate the issue. Specifically, we will early stop a layer if its performance does not improve for $\tau = 10$ checkpoints

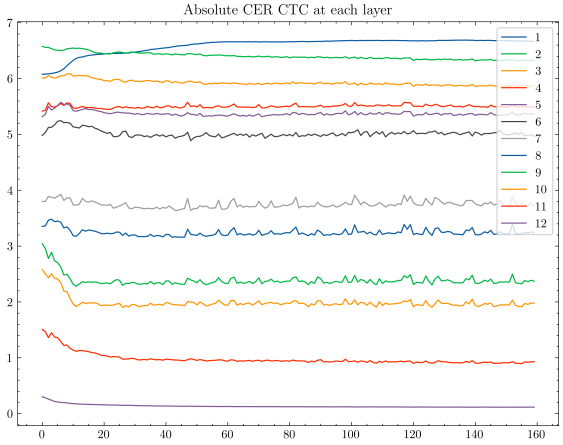

FIGURE 4.3: Layerwise CER computed by feeding each encoder layers skip connection into CTC decoder. Each color line represents the CER across epochs of one of the 12 encoder layers. Best view in color.

TABLE 4.4: Effects of adapting only specific parts of the pretrained ASR model on VITS-SPKSET1. WER, substituion (sub.), insertion (ins.) and deletion (del.) errors are reported on Aishell-1 dataset.

Encoder parts adapted	Decoders adapted	WER (%)	Error (%) sub.	ins.	del.
Unadapted					
-	-	57.6	44.0	5.5	8.2
Baseline: Adapt encoder					
layer 1	-	62.5	47.3	3.7	11.6
layer 12	-	39.4	31.1	2.1	6.2
Baseline: Adapt decoder					
-	CTC	62.5	46.8	4.1	11.6
-	attention-AR	43.1	32.3	2.8	8.0
-	all	27.5	20.8	1.9	4.9
Baseline: Adapt encoder and decoder					
all	all	23.0	18.1	0.9	4.1
layer 12	attention-AR	21.7	16.8	1.2	3.8
layer 12	all	21.1	16.2	1.1	3.8
Our strategy S_A					
layer 1-2,11-12	all	**19.8**	**15.4**	1.1	**3.3**

on the AISHELL-1 validataion set. Results show that our strategy achieves a relative CER improvement of 5.7% on the Datatang-ZH test set over the baseline strategy while maintaining the same CER on AISHELL-1 test set.

TABLE 4.5: Effects of adapting the Whisper ASR model on AISHELL-1 train set. "None" indicates that no adaptation is performed. Full adapt refers to adapting the whole model without freezing weights.

Adaptation Method	Model Size (M)	AISHELL-1 (CER%)	Datatang-ZH (CER%)
None	244M	13.5	46.2
Full adapt	244M	4.8	38.3
S_B adapt	244M	4.8	**36.1**

4.3.5.3 Freeze layer if degrade in pretrained domain (S_C)

Table 4.6 shows the results of adapting the Whisper-small ASR model on the AISHELL-1 train-set. The unadapted model gives CER of 13.5% on AISHELL-1 test-set and WER of 3.3% on Librispeech test-set. After adapting the whole model on the train-set for just 1.75 epochs, there is a relative CER improvement of 58.5% on AISHELL-1 test-set. However, catastrophic forgetting also occurs and the WER on Librispeech test-set degrades by 66.7%. To mitigate the issue, we apply our strategy S_C. Specifically, we choose only 5 samples from Librispeech test-set as our new validation set, and we will early stop a layer during adaptation if its performance degrades by more than $\epsilon = 0.07$ relatively.

To compare the results of our strategy with the the full adaptation baseline strategy, we choose the model adapted for 1.875 epochs using our strategy and compare it with the baseline model adapted for 1.75 epochs, as they have similar CER on AISHELL-1 test-set. Results shows that while the model using our strategy and the baseline model has a similar CER, our model achieves a relative 16.4% lower WER on the Librispeech test-set.

4.4 Summary of chapter

In this chapter, two novel approaches are discussed to improve the state-of-the-art performance for TTS synthesized data adaptation and layer-wise adaptation.

For TTS synthesized data adaptation, we propose to leverage multiple TTS systems to improve the diveresity of the TTS synthesized data. Our model using multiple TTS data improves relative WER by 9.8% over the baseline single TTS

TABLE 4.6: Effects of adapting the Whisper-small ASR model on AISHELL-1 train set. "None" indicates that no adaptation is performed. Full adapt refers to adapting the whole model without freezing weights.

Adaptation Method	Epochs	Model Size (M)	AISHELL-1 (CER%)	Librispeech (WER%)	MER%
None	0	244M	13.5	3.3	8.4
Baselines					
Full Adapt	1.75	244M	5.6	5.5	5.6
Full Adapt	5	244M	5.6	5.5	5.6
Adapter	1.75	244M	8.7	5.4	7.1
Adapter	5	244M	8.0	5.4	6.7
Our strategy S_C					
S_C Adapt	1.75	244M	5.9	4.5	5.2
S_C Adapt	1.875	244M	5.7	4.6	5.2
S_C Adapt	5	244M	5.2	4.8	**5.0**

data, and improves relative WER by 71.4% over an unadapted model, showing the effectiveness of our adaptation method.

For layer-wise adaptation, we introduced an automated search approach for identifying the optimal set of layers to be frozen for residual networks. Our method can reduce overfitting and improves relative CER by 5.7% on a Chinese Datatang-ZH test-set when we adapt Whisper on a Chinese AISHELL-1 train set. Also, our model can reduce catastrophic forgetting and improves relative WER by 16.4% on the English Librispeech test-set when we adapt Whisper on the Chinese AISHELL-1 train set.

Chapter 5

Conclusions and Future Work

This thesis has focused on the topic of end-to-end ASR adaptation, specifically focusing on improving ASR performance within the constraints of limited data resources and exclusive access to text data for adaptation, highlighting their challenges such as the risk of overfitting, catastrophic forgetting, and the training limitation. Layerwise adaptation and synthesized text-to-speech (TTS) data adaptation were explored to address the issues.

However, previous works on layerwise adaptation suffers from the expensive search process to identify the optimal subset of layers to freeze. For synthesized text-to-speech (TTS) data adaptation, it suffers from the lack of diversity in its synthesized data as only a single TTS system is used in the synthesis process.

To address the issues, this thesis has developed two novel approaches. In particular, to address the challenge of conducting a costly search for layers to be frozen during layerwise adaptation, this thesis introduces an automated approach for selecting the layers to be frozen.

In response to the challenge of limited diversity in synthesized data from a single Text-to-Speech (TTS) system in TTS data synthesized adaptation, this thesis explores harnessing multiple TTS systems to generate synthetic audio, thereby creating a variety of artificial audio-text pairs for model training.

To conclude the work, Section 6.1 first summarises the contributions proposed in this thesis. Finally, Section 6.2 discusses some of the future directions that can be explored, and the challenges that are still to be faced.

5.1 Contributions

This thesis has developed two novel approaches for TTS synthesized data adaptation and layer-wise adaptation. They are summarized in the following subsections.

5.1.1 TTS synthesized data adaptation

For TTS synthesized data adaptation, it is an approach to adapt ASR models when only text data are present. This is challenging as text-audio data pairs are required to train an ASR model under the supervised learning framework. To solve the problem, a TTS system is employed to generate synthetic audio from the existing text-only data. This synthetic audio is then combined with the original text to create artificial audio-text pairs for training.

Previous works have shown that training an ASR model on more diverse synthesized data can avoid overfitting and improve the model's robustness. However, previous works only use a single TTS model conditioned on multiple speakers to produce different speaker voices while the use of multiple TTS systems to improve data diversity is rarely explored.

To address this, we explore the use of multiple TTS systems to synthesize training data to perform ASR adaptation. Our findings are as follows: 1) We find that to generate diverse synthetic data, using multiple TTS systems conditioned on the same speaker is more effective than using a single TTS system conditioned on multiple speakers. 2) Also, we find that freezing part of an ASR model when adapting to synthetic data gives better result than adapting the whole ASR model.

When we apply our method to a rare word dataset partitioned from National Speech Corpus SG, which contains mostly road names and addresses in its text transcripts, experiments show that a pretrained ASR model adapted to our multi-TTS-same-SPK data gives relatively 9.8% lower word error rate (WER) compared to the ASR models adapted to same-TTS-multi-SPK data of the same data size, and our overall adaptation improves the model's WER from 57.6% to 16.5% without using any real audio as training data.

5.1.2 Layer-wise adaptation

For layer-wise adaptation, it is an approach to adapt only certain layers of an ASR model. It involves the selective freezing of specific layer weights that are susceptible to the overfitting and catastrophic forgetting problems, thereby mitigating them. Simultaneously, other layer weights are allowed to be updated to enhance the model's performance within the target domain.

Previous works have shown that to determine the subset of layers to freeze, they either depend on heuristics or a trial-and-error process to find the subset which results in better model performance. Consequently, this process still demands expert knowledge and substantial computational resources to identify the most effective layers to freeze.

To address this challenge, we introduced an automated search approach for identifying the optimal set of layers to be frozen. This method only applies to residual networks. Our findings are as follows: 1) We find that the layerwise performance estimated from the skip connections in residual networks are effective predictors to identify overfitting layers or layers that undergo catastrophic forgetting. 2) Once the layers start to overfit or undergo catastrophic forgetting, early stopping them can effectively mitigate the issues.

Experiment results show that our method can effectively regularize adaptation through layerwise weight freezing without the requirement of any expert knowledge and expensive computation powers to find the optimal layers. Our method can reduce overfitting and improves relative CER by 5.7% on a Chinese Datatang-ZH test-set when we adapt Whisper on a Chinese AISHELL-1 train set. Also, our model can reduce catastrophic forgetting and improves relative WER by 16.4% on the English Librispeech test-set when we adapt Whisper on the Chinese AISHELL-1 train set.

5.2 Future Directions

Based on the results and conclusions of this thesis, the following recommendations are made for future study in order to expand the knowledge on this subject: 1) For TTS data synthesized data adaptation, our work only levearge two TTS systems

to improve the synthesized data diversity. In the furture, synthesizing data with more TTS systems can be explored. Also, an analysis on the contribution of each individual TTS system can be done. 2) For layerwise adaptation, our work only shows that by estimating a layerwise performance from residual networks, we can identify and early stop the layers that overfit and undergo catastrophic forgetting. In the future, an exploration on the wider applications of the estimated layerwise performance can be done. Also, the possibility of estimating a layerwise performan on non-residual networks can be explored by adding extra classification heads to each model layer.

Bibliography

[1] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015. xix, 10, 11, 13, 14, 44

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. xix, 4, 10, 11, 15, 34, 60

[3] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019. xix, 17

[4] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. xix, 17

[5] Kwangyoun Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, Kyu J Han, and Shinji Watanabe. E-branchformer: Branchformer with enhanced merging for speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 84–91. IEEE, 2023. xix, 1, 4, 9, 17, 18

[6] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020. xix, 18, 19

[7] Shih-An Li, Yu-Ying Liu, Yun-Chien Chen, Hsuan-Ming Feng, Pi-Kang Shen, and Yu-Che Wu. Voice interaction recognition design in real-life scenario mobile robot applications. *Applied Sciences*, 13(5):3359, 2023. 1

[8] Dominik Macháček, Raj Dabre, and Ondřej Bojar. Turning whisper into real-time transcription system. *arXiv preprint arXiv:2307.14743*, 2023. 1

[9] Anil Kumar Gupta, Rachna Somkunwar, Anjali Kumari, Ankita Kumari, Komal Godhke, and Jayshree Repale. Web based multilingual real time speech transcription transliteration and translation system. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–5, 2021. doi: 10.1109/ISCON52037.2021.9702481. 1

[10] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020. 1, 6, 16

[11] Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning*, pages 17627–17643. PMLR, 2022. 1, 17

[12] Xianchao Wu. Deep sparse conformer for speech recognition. *arXiv preprint arXiv:2209.00260*, 2022. 1

[13] John Waldo. A comparative study of back propagation and its alternatives on multilayer perceptrons. *arXiv preprint arXiv:2206.06098*, 2022. 2

[14] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 2, 12, 16, 32, 37

[15] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*, 2020. 2, 9

[16] Qiu-Shi Zhu, Jie Zhang, Zi-Qiang Zhang, and Li-Rong Dai. Joint training of speech enhancement and self-supervised model for noise-robust asr. *arXiv preprint arXiv:2205.13293*, 2022. 2

[17] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015. 2, 32

[18] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006. 2, 53

[19] Katrin Tomanek, Vicky Zayats, Dirk Padfield, Kara Vaillancourt, and Fadi Biadsy. Residual adapters for parameter-efficient asr adaptation to atypical and accented speech. *arXiv preprint arXiv:2109.06952*, 2021. 2, 9, 53

[20] Vladimir Bataev, Roman Korostik, Evgeny Shabalin, Vitaly Lavrukhin, and Boris Ginsburg. Text-only domain adaptation for end-to-end asr using integrated text-to-mel-spectrogram generator. *arXiv preprint arXiv:2302.14036*, 2023. 3

[21] Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu. Prompting large language models for zero-shot domain adaptation in speech recognition. *arXiv preprint arXiv:2306.16007*, 2023. 3

[22] Abhayjeet Singh, Arjun Singh Mehta, Jai Nanavati, Jesuraja Bandekar, Karnalius Basumatary, Sandhya Badiger, Sathvik Udupa, Saurabh Kumar, Prasanta Kumar Ghosh, Priyanka Pai, et al. Model adaptation for asr in low-resource indian languages. *arXiv preprint arXiv:2307.07948*, 2023. 2, 3, 9, 53

[23] Daniel Bashir, George D Montañez, Sonia Sehra, Pedro Sandoval Segura, and Julius Lauw. An information-theoretic perspective on overfitting and underfitting. In *AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33*, pages 347–358. Springer, 2020. 2, 54

[24] Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*, 2019. 2, 54

[25] Changfeng Gao, Gaofeng Cheng, Runyan Yang, Han Zhu, Pengyuan Zhang, and Yonghong Yan. Pre-training transformer decoder for end-to-end asr model with unpaired text data. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6543–6547, 2021. URL `https://api.semanticscholar.org/CorpusID:235780064`. 2, 23, 25

[26] Vrunda N Sukhadia and S Umesh. Domain adaptation of low-resource target-domain models using well-trained asr conformer models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 295–301. IEEE, 2023. 3

[27] Tsendsuren Munkhdalai, Zelin Wu, Golan Pundak, Khe Chai Sim, Jiayang Li, Pat Rondon, and Tara N Sainath. Nam+: Towards scalable end-to-end contextual biasing for adaptive asr. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 190–196. IEEE, 2023. 3

[28] Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramón Fernández Astudillo, and K. Takeda. Back-translation-style data augmentation for end-to-end asr. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 426–433, 2018. URL `https://api.semanticscholar.org/CorpusID:51879045`. 3, 23, 24

[29] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong. Adaptation of context-dependent deep neural networks for automatic speech recognition. *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 366–369, 2012. URL `https://api.semanticscholar.org/CorpusID:11308291`. 3, 21, 23, 27, 54, 56

[30] Sabato Marco Siniscalchi, Jinyu Li, and Chin-Hui Lee. Hermitian polynomial for speaker adaptation of connectionist speech recognition systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 21:2152–2161, 2013. URL `https://api.semanticscholar.org/CorpusID:372006`. 21, 56

[31] Lahiru Samarakoon and Khe Chai Sim. Factorized hidden layer adaptation for deep neural network based acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24:2241–2250, 2016. URL `https://api.semanticscholar.org/CorpusID:18187402`. 23, 54, 56

[32] Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199, 2021. URL `https://api.semanticscholar.org/CorpusID:231672601`. 3, 21, 54, 56

[33] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong. Adaptation of context-dependent deep neural networks for automatic speech recognition. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 366–369. IEEE, 2012. 4, 50

[34] Yanmin Qian, Xun Gong, and Houjun Huang. Layer-wise fast adaptation for end-to-end multi-accent speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2842–2853, 2022. 4

[35] Sachin Singh, Ashutosh Gupta, Aman Maghan, Dhananjaya Gowda, Shatrughan Singh, and Chanwoo Kim. Comparative study of different tokenization strategies for streaming end-to-end asr. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 388–394. IEEE, 2021. 4

[36] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE, 2017. 4

[37] Prashanth Gurunath Shivakumar and Panayiotis Georgiou. Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Computer speech & language*, 63:101077, 2020. 4

[38] Xinwei Li, Yue Pan, Matthew Gibson, and Puming Zhan. Dnn online adaptation for automatic speech recognition. In *Konferenz Elektronische Sprachsignalverarbeitung*, pages 46–53. TUDpress, Dresden, 2018. 4

[39] Zied Elloumi, Laurent Besacier, Olivier Galibert, and Benjamin Lecouteux. Analyzing learned representations of a deep asr performance prediction model. *arXiv preprint arXiv:1808.08573*, 2018. 4, 56

[40] Kyuhong Shim, Jungwook Choi, and Wonyong Sung. Understanding the role of self attention for efficient speech recognition. In *International Conference on Learning Representations*, 2021. 4

[41] Gary Wang, Andrew Rosenberg, Zhehuai Chen, Yu Zhang, Bhuvana Ramabhadran, Yonghui Wu, and Pedro J. Moreno. Improving speech recognition using consistent predictions on synthesized speech. *ICASSP 2020 - 2020*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7029–7033, 2020. URL `https://api.semanticscholar.org/CorpusID:216343650`. 5, 23, 26

[42] Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett. Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5674–5678, 2020. URL `https://api.semanticscholar.org/CorpusID:227126398`. 26

[43] Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, and Tatsuya Kawahara. Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 477–484, 2018. URL `https://api.semanticscholar.org/CorpusID:61806902`. 26

[44] Shaofei Xue, Jian Tang, and Yazhu Liu. Improving speech recognition with augmented synthesized data and conditional model training. *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 443–447, 2022. URL `https://api.semanticscholar.org/CorpusID:256669482`.

[45] Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. Data augmentation for asr using tts via a discrete representation. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 68–75, 2021. URL `https://api.semanticscholar.org/CorpusID:246532199`. 5, 23

[46] Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6161–6165. IEEE, 2019. 5, 43, 44

[47] Chenpeng Du and Kai Yu. Speaker augmentation for low resource speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7719–7723. IEEE, 2020. 5, 43, 44

[48] Jan Melechovsky, Ambuj Mehrish, Dorien Herremans, and Berrak Sisman. Learning accent representation with multi-level vae towards controllable speech synthesis. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 928–935. IEEE, 2023. 5, 43

[49] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, Andrew Rosenberg, Bhuvana Ramabhadran, and Yonghui Wu. Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6699–6703. IEEE, 2020. 5, 43

[50] Murali Karthick Baskar, Andrew Rosenberg, Bhuvana Ramabhadran, and Yu Zhang. Reducing domain mismatch in self-supervised speech pretraining. 2022. 9

[51] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989. 10

[52] Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. End-to-end speech recognition: A survey. *arXiv preprint arXiv:2303.03329*, 2023. 10, 11

[53] Karel Veselý, Arnab Ghoshal, Luká Burget, and Daniel Povey. Sequence-discriminative training of deep neural networks. In *Interspeech*, 2013. URL `https://api.semanticscholar.org/CorpusID:2827512`. 10

[54] S Karpagavalli and E Chandra. Phoneme and word based model for tamil speech recognition using gmm-hmm. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–5. IEEE, 2015. 10

[55] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996. 10

[56] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009. 10

[57] Mousumi Malakar and Ravindra B Keskar. Progress of machine learning based automatic phoneme recognition and its prospect. *Speech Communication*, 135:37–53, 2021. 10

[58] Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*, 2023. 10, 32

[59] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018, 2019. 10, 11

[60] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. 11

[61] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. 11

[62] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays. Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping. In *Interspeech*, volume 8, pages 1298–1302, 2017. 11

[63] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley. Hybrid autoregressive transducer (hat). In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6139–6143. IEEE, 2020. 11

[64] Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023. 11

[65] Weiran Wang, Guangsen Wang, Aadyot Bhatnagar, Yingbo Zhou, Caiming Xiong, and Richard Socher. An investigation of phone-based subword units for end-to-end speech recognition. *arXiv preprint arXiv:2004.04290*, 2020. 11

[66] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4280–4284. IEEE, 2015. 12

[67] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005. 13

[68] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv preprint arXiv:2005.03191*, 2020. 16

[69] Jin Sakuma, Tatsuya Komatsu, and Robin Scheibler. Mlp-based architecture with variable length input for automatic speech recognition. 2021. 17

[70] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008. 18

[71] Xuankai Chang, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, Shu-wen Yang, Yu Tsao, Hung-yi Lee, et al. An exploration of self-supervised pretrained representations for end-to-end speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 228–235. IEEE, 2021. 18

[72] Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabaleiro, Kun Qian, Xin Jing, Alexander Kathan, Bin Hu, and Bjoern W Schuller. Audio self-supervised learning: A survey. *Patterns*, 3(12), 2022. 18

[73] Dongwei Jiang, Xiaoning Lei, Wubo Li, Ne Luo, Yuxuan Hu, Wei Zou, and Xiangang Li. Improving transformer-based speech recognition using unsupervised pre-training. *arXiv preprint arXiv:1910.09932*, 2019. 18

[74] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019. 18

[75] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. 19, 20

[76] John A. Hartigan and M. Anthony. Wong. A k-means clustering algorithm. 1979. URL `https://api.semanticscholar.org/CorpusID:53880671`. 19

[77] Juan Zuluaga-Gomez, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan. How does pre-trained wav2vec 2.0 perform on domain-shifted asr? an extensive benchmark on air traffic control communications. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 205–212. IEEE, 2023. 20, 53

[78] Anjuli Kannan, Arindrima Datta, Tara N Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. Large-scale multilingual speech recognition with a streaming end-to-end model. *arXiv preprint arXiv:1909.05330*, 2019. 21, 54

[79] Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very deep multilingual convolutional neural networks for lvcsr. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4955–4959. IEEE, 2016.

[80] Ana Isabel Garcia-Moral, Rubén Solera-Urena, Carmen Pelaez-Moreno, and Fernando Diaz-de Maria. Data balancing for efficient training of hybrid ann/hmm automatic speech recognition systems. *IEEE Transactions on audio, speech, and language processing*, 19(3):468–481, 2010. 21, 54

[81] Tanel Alumäe, Stavros Tsakalidis, and Richard M. Schwartz. Improved multilingual training of stacked neural network acoustic models for low resource languages. In *Interspeech*, 2016. URL `https://api.semanticscholar.org/CorpusID:1573641`. 21, 54

[82] Hank Liao. Speaker adaptation of context dependent deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7947–7951. IEEE, 2013. 21, 28, 54, 56

[83] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7893–7897. IEEE, 2013. 21, 28, 54

[84] Zhong Meng, Jinyu Li, and Yifan Gong. Adversarial speaker adaptation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5721–5725. IEEE, 2019. 21, 28, 54

[85] Bo Li, Tara N. Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Z. Chen, Yan-Qing Wu, and Kanishka Rao. Multi-dialect speech recognition with a single sequence-to-sequence model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4749–4753, 2017. URL `https://api.semanticscholar.org/CorpusID:338426`. 21

[86] Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro J. Moreno, Eugene Weinstein, and Kanishka Rao. Multilingual speech recognition with a single end-to-end model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908, 2017. URL `https://api.semanticscholar.org/CorpusID:3267237`. 21

[87] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, 2013. URL `https://api.semanticscholar.org/CorpusID:10257575`. 21

[88] Andrew W. Senior and Ignacio Lopez-Moreno. Improving dnn speaker independence with i-vector inputs. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 225–229, 2014. URL `https://api.semanticscholar.org/CorpusID:12995181`.

[89] Vishwa Gupta, Patrick Kenny, Pierre Ouellet, and Themos Stafylakis. I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6334–6338, 2014. URL `https://api.semanticscholar.org/CorpusID:10509359`. 21

[90] Zhiyun Fan, Jie Li, Shiyu Zhou, and Bo Xu. Speaker-aware speech-transformer. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 222–229, 2019. URL `https://api.semanticscholar.org/CorpusID:209862771`. 21

[91] Michael L. Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402, 2013. URL `https://api.semanticscholar.org/CorpusID:10310847`. 21

[92] Puyuan Peng, Brian Yan, Shinji Watanabe, and David F. Harwath. Prompting the hidden talent of web-scale speech models for zero-shot task generalization. *ArXiv*, abs/2305.11095, 2023. URL `https://api.semanticscholar.org/CorpusID:258762742`. 21

[93] Bethan Thomas, Samuel Kessler, and Salah Karout. Efficient adapter transfer of self-supervised speech models for automatic speech recognition. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7102–7106, 2022. URL `https://api.semanticscholar.org/CorpusID:246634891`. 21

[94] Genta Indra Winata, Guangsen Wang, Caiming Xiong, and Steven C. H. Hoi. Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition. In *Interspeech*, 2020. URL `https://api.semanticscholar.org/CorpusID:227253672`.

[95] Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P. Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann. Contextual adapters for personalized speech recognition in neural transducers. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8537–8541, 2022. URL `https://api.semanticscholar.org/CorpusID:249152195`. 21

[96] Taesup Kim, Inchul Song, and Yoshua Bengio. Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition. In *Interspeech*, 2017. URL `https://api.semanticscholar.org/CorpusID:2767731`. 21

[97] Pawel Swietojanski and Steve Renals. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 171–176, 2014. URL `https://api.semanticscholar.org/CorpusID:2052273`. 21

[98] Pawel Swietojanski and Steve Renals. Differentiable pooling for unsupervised speaker adaptation. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4305–4309, 2015. URL `https://api.semanticscholar.org/CorpusID:10677123`. 21

[99] Xurong Xie, Xunying Liu, Tan Lee, Shoukang Hu, and Lan Wang. Blhuc: Bayesian learning of hidden unit contributions for deep neural network speaker adaptation. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5711–5715, 2019. URL `https://api.semanticscholar.org/CorpusID:145975590`. 21

[100] Shaofei Xue, Hui Jiang, Lirong Dai, and Qingfeng Liu. Speaker adaptation of hybrid nn/hmm model for speech recognition based on singular value decomposition. *Journal of Signal Processing Systems*, 82:175–185, 2014. URL `https://api.semanticscholar.org/CorpusID:10779174`. 23

[101] Jian Xue, Jinyu Li, Dong Yu, Michael L. Seltzer, and Yifan Gong. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6359–6363, 2014. URL `https://api.semanticscholar.org/CorpusID:12971356`.

[102] Kshitiz Kumar, Chaojun Liu, Kaisheng Yao, and Yifan Gong. Intermediate-layer dnn adaptation for offline and session-based iterative speaker adaptation. In *Interspeech*, 2015. URL `https://api.semanticscholar.org/CorpusID:27713304`. 23

[103] Yong Zhao, Jinyu Li, and Yifan Gong. Low-rank plus diagonal adaptation for deep neural networks. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5005–5009, 2016. URL `https://api.semanticscholar.org/CorpusID:10506309`. 23

[104] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL `https://api.semanticscholar.org/CorpusID:235458009`. 23

[105] Wei Ming Liu, Ying Qin, Zhiyuan Peng, and Tan Lee. Sparsely shared lora on whisper for child speech recognition. *ArXiv*, abs/2309.11756, 2023. URL `https://api.semanticscholar.org/CorpusID:262084086`.

[106] Qingru Zhang, Minshuo Chen, Alexander W. Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. *ArXiv*, abs/2303.10512, 2023. URL `https://api.semanticscholar.org/CorpusID:257631760`. 23

[107] Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *ArXiv*, abs/1503.03535, 2015. URL `https://api.semanticscholar.org/CorpusID:15352384`. 23

[108] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. Cold fusion: Training seq2seq models together with language models. In *Interspeech*, 2017. URL `https://api.semanticscholar.org/CorpusID:31004450`. 23

[109] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie. Component fusion: Learning replaceable language model component for end-to-end speech recognition system. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5361–5635, 2019. URL `https://api.semanticscholar.org/CorpusID:145876149`. 23

[110] Xun Gong, Wei Wang, Hang Shao, Xie Chen, and Yanmin Qian. Factorized aed: Factorized attention-based encoder-decoder for text-only domain adaptive asr. *ICASSP 2023 - 2023 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP), 2023. URL https://api.semanticscholar.org/CorpusID:258529852. 23

[111] Chris Leggetter and Philip C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Comput. Speech Lang.*, 9:171–185, 1995. URL https://api.semanticscholar.org/CorpusID:14708613. 23

[112] Yajie Miao, Hao Zhang, and Florian Metze. Towards speaker adaptive training of deep neural network acoustic models. In *Interspeech*, 2014. URL https://api.semanticscholar.org/CorpusID:14639970. 23

[113] Tasos Anastasakos, John W. McDonough, Richard M. Schwartz, and John Makhoul. A compact model for speaker-adaptive training. *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, 2:1137–1140 vol.2, 1996. URL https://api.semanticscholar.org/CorpusID:15921794. 23

[114] Frank Seide, Gang Li, Xie Chen, and Dong Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 24–29, 2011. URL https://api.semanticscholar.org/CorpusID:9933050. 23

[115] Tian Tan, Yanmin Qian, and Kai Yu. Cluster adaptive training for deep neural network based acoustic model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(3):459–468, 2015. 23

[116] Reinhold Häb-Umbach and Hermann Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1:13–16 vol.1, 1992. URL https://api.semanticscholar.org/CorpusID:12645539. 23

[117] Ernst Günter Schukat-Talamazzini, Joachim Hornegger, and Heinrich Niemann. Optimal linear feature transformations for semi-continuous hidden markov models. *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1:369–372 vol.1, 1995. URL https://api.semanticscholar.org/CorpusID:12952529. 23

[118] George Saon, Mukund Padmanabhan, Ramesh A. Gopinath, and Scott Saobing Chen. Maximum likelihood discriminant feature spaces. *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, 2:II1129–II1132 vol.2, 2000. URL https://api.semanticscholar.org/CorpusID:8625628. 23

[119] Romain Serizel and Diego Giuliani. Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition. *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 135–140, 2014. URL https://api.semanticscholar.org/CorpusID:43745751. 23

[120] Romain Serizel and Diego Giuliani. Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 135–140. IEEE, 2014. 23

[121] Vassilios V. Digalakis, Dimitry Rtischev, and Leonardo Neumeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Trans. Speech Audio Process.*, 3:357–366, 1995. URL `https://api.semanticscholar.org/CorpusID:8462692`. 23

[122] Mark John Francis Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Comput. Speech Lang.*, 12:75–98, 1998. URL `https://api.semanticscholar.org/CorpusID:9241826`. 23

[123] Takaaki Hori, Ramón Fernández Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux. Cycle-consistency training for end-to-end speech recognition. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6271–6275, 2018. URL `https://api.semanticscholar.org/CorpusID:53219757`. 24

[124] Jason Li, Ravi Teja Gadde, Boris Ginsburg, and Vitaly Lavrukhin. Training neural speech recognition systems with synthetic speech augmentation. *ArXiv*, abs/1811.00707, 2018. URL `https://api.semanticscholar.org/CorpusID:53295159`. 26

[125] Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro J. Moreno, Yonghui Wu, and Zelin Wu. Speech recognition with augmented synthesized speech. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 996–1002, 2019. URL `https://api.semanticscholar.org/CorpusID:202889273`.

[126] Daria Soboleva, Ondrej Skopek, M'arius vSajgal'ik, Victor Cuarbune, Felix Weissenberger, Julia Proskurnia, Bogdan Prisacari, Daniel Valcarce, Justin Lu, Rohit Prabhavalkar, and Balint Miklos. Replacing human audio with synthetic audio for on-device unspoken punctuation prediction. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7653–7657, 2020. URL `https://api.semanticscholar.org/CorpusID:224803173`.

[127] Nick Rossenbach, Mohammad Zeineldeen, Benedikt Hilmes, Ralf Schlüter, and Hermann Ney. Comparing the benefit of synthetic training data for various automatic speech recognition architectures. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 788–795, 2021. URL `https://api.semanticscholar.org/CorpusID:233210536`. 26

[128] Zhehuai Chen, Andrew Rosenberg, Yu Zhang, Gary Wang, Bhuvana Ramabhadran, and Pedro J. Moreno. Improving speech recognition using gan-based speech synthesis and contrastive unspoken text selection. In *Interspeech*, 2020. URL `https://api.semanticscholar.org/CorpusID:226206599`. 26

[129] Aleksandr Laptev, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey V. Rybin. You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 439–444, 2020. URL `https://api.semanticscholar.org/CorpusID:218630223`. 26

[130] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. Variational information bottleneck for effective low-resource fine-tuning. *arXiv preprint arXiv:2106.05469*, 2021. 27

[131] Tiezheng Yu, Zihan Liu, and Pascale Fung. Adaptsum: Towards low-resource domain adaptation for abstractive summarization. *arXiv preprint arXiv:2103.11332*, 2021. 27

[132] Steven Vander Eeckt et al. Using adapters to overcome catastrophic forgetting in end-to-end automatic speech recognition. *arXiv preprint arXiv:2203.16082*, 2022. 27

[133] Shamil Ayupov and Nadezhda Chirkova. Parameter-efficient finetuning of transformers for source code. *arXiv preprint arXiv:2212.05901*, 2022. 27

[134] DE Rumerhart. Learning internal representation by error propagation. *Parallel distributed processing*, 1:318–362, 1986. 28

[135] Jodi Kearns. Librivox: Free public domain audiobooks. *Reference Reviews*, 28(1):7–8, 2014. 32

[136] Minjeong Kim. The creative commons and copyright protection in the digital era: Uses of creative commons licenses. *Journal of computer-mediated communication*, 13(1):187–209, 2007. 32

[137] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*, pages 1–5. IEEE, 2017. 32

[138] Andrew Sinclair. License profile: Apache license, version 2.0. *IFOSS L. Rev.*, 2:107, 2010. 32

[139] Ltd Beijing DataTang Technology Co. Mandarin conversational speech data from datatang. 33

[140] Jia Xin Koh, Aqilah Mislan, Kevin Khoo, Brian Ang, Wilson Ang, Charmaine Ng, and YY Tan. Building the singapore english national speech corpus. *Malay*, 20(25.0):19–3, 2019. 33

[141] Dietrich Klakow and Jochen Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1-2):19–28, 2002. 34

[142] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966. 34

[143] Amit Meghanani, CS Anoop, and AG Ramakrishnan. An exploration of log-mel spectrogram and mfcc features for alzheimer's dementia recognition from spontaneous speech. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–677. IEEE, 2021. 35

[144] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 35

[145] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021. 35

[146] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 35

[147] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 35, 38, 43, 44

[148] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 35, 38

[149] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 35, 47

[150] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018. 35, 38

[151] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 37

[152] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 37

[153] Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016. 37

[154] Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, 26 (1):19–45, 2000. 37

[155] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 37

[156] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 37, 38, 47

[157] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013. 38

[158] Cal Peyser, Sepand Mavandadi, Tara N Sainath, James Apfel, Ruoming Pang, and Shankar Kumar. Improving tail performance of a deliberation e2e asr model using a large text corpus. *arXiv preprint arXiv:2008.10491*, 2020. 42

[159] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. A comparison of techniques for language model integration in encoder-decoder speech recognition. In *2018 IEEE spoken language technology workshop (SLT)*, pages 369–375. IEEE, 2018. 43

[160] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *Interspeech*, pages 1420–1424, 2014. 43, 44

[161] Nick Rossenbach, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7069–7073. IEEE, 2020. 43, 44

[162] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018. 43, 51

[163] Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE, 2020. 43, 51

[164] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. Deep context: end-to-end contextual speech recognition. In *2018 IEEE spoken language technology workshop (SLT)*, pages 418–425. IEEE, 2018. 44

[165] Uri Alon, Golan Pundak, and Tara N Sainath. Contextual speech recognition with difficult negative training examples. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444. IEEE, 2019. 44

[166] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie. Attention-based end-to-end speech recognition on voice search. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4764–4768. IEEE, 2018. 44

[167] Takaaki Hori, Shinji Watanabe, and John R Hershey. Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 287–293. IEEE, 2017.

[168] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE, 2016. 44

[169] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*, 2017. 44

[170] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*, 2015. 44

[171] Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett. Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5674–5678. IEEE, 2021. 44

[172] Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, and Tatsuya Kawahara. Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 477–484. IEEE, 2018. 44

[173] Shaofei Xue, Jian Tang, and Yazhu Liu. Improving speech recognition with augmented synthesized data and conditional model training. In *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 443–447. IEEE, 2022. 44

[174] Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. Data augmentation for asr using tts via a discrete representation. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 68–75. IEEE, 2021. 44

[175] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713, 2019. 45

[176] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020. 45

[177] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR, 2021. 45, 47

[178] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017. 46

[179] Tomoki Hayashi, Ryuichi Yamamoto, Takenori Yoshimura, Peter Wu, Jiatong Shi, Takaaki Saeki, Yooncheol Ju, Yusuke Yasuda, Shinnosuke Takamichi, and Shinji Watanabe. Espnet2-tts: Extending the edge of tts research. *arXiv preprint arXiv:2110.07840*, 2021. 46, 47

[180] Gölge Eren and The Coqui TTS Team. Coqui TTS, January 2021. URL `https://github.com/coqui-ai/TTS`. 46

[181] Yasha Iravantchi, Mayank Goel, and Chris Harrison. Digital ventriloquism: giving voice to everyday objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2020. 46

[182] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4784–4788. IEEE, 2018. 47

[183] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022. 50

[184] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 51

[185] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022. 53

[186] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019. 54

[187] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017. 54

[188] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 55

[189] Aku Rouhe, Anja Virkkunen, Juho Leinonen, Mikko Kurimo, et al. Low resource comparison of attention-based and hybrid asr exploiting wav2vec 2.0. *Interspeech 2022*, pages 3543–3547, 2022. 56

[190] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 56

[191] Xun Gong, Yizhou Lu, Zhikai Zhou, and Yanmin Qian. Layer-wise fast adaptation for end-to-end multi-accent speech recognition. *arXiv preprint arXiv:2204.09883*, 2022. 56

[192] Li Yang, Sen Lin, Fan Zhang, Junshan Zhang, and Deliang Fan. Efficient self-supervised continual learning with progressive task-correlated layer freezing. *arXiv preprint arXiv:2303.07477*, 2023. 56

[193] Yixiong Chen, Alan Yuille, and Zongwei Zhou. Which layer is learning faster? a systematic exploration of layer-wise convergence rate for deep neural networks. In *The Eleventh International Conference on Learning Representations*, 2022. 57

[194] Xiaojie Jin, Yunpeng Chen, Jian Dong, Jiashi Feng, and Shuicheng Yan. Collaborative layer-wise discriminative learning in deep neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 733–749. Springer, 2016. 57

[195] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pages 2927–2936. PMLR, 2018. 57

[196] Vasista Sai Lodagala, Sreyan Ghosh, and S Umesh. Pada: Pruning assisted domain adaptation for self-supervised speech representations. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 136–143. IEEE, 2023. 57

[197] Xinyu Zhang, Ian Colbert, Ken Kreutz-Delgado, and Srinjoy Das. Training deep neural networks with joint quantization and pruning of weights and activations. *arXiv preprint arXiv:2110.08271*, 2021. 57

[198] Xin Yu, Thiago Serra, Srikumar Ramalingam, and Shandian Zhe. The combinatorial brain surgeon: pruning weights that cancel one another in neural networks. In *International Conference on Machine Learning*, pages 25668–25683. PMLR, 2022. 57

[199] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002. 59

[200] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. *arXiv preprint arXiv:2304.07288*, 2023. 60