Voice Activity Detection using Attention-Based Complex Ideal Ratio Mask and Transformer-Based Deep Neural Networks

Yifei Zhao

Department of Electrical & Computer Engineering

McGill University, Montreal, Quebec

July 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science.

Abstract

Voice Activity Detection (VAD) is often treated as a classification problem where the goal is to discriminate, at a given time, between desired speech and background noise. Although many state-of-the-art approaches for increasing the performance of VAD have been proposed, they are still not robust enough to be applied under adverse noise conditions with low signal-to-noise ratio (SNR). In this work, we first introduce a novel attention model-based phase-aware deep neural network architecture for VAD which takes advantage of complex Ideal Ratio Mask (cIRM). The proposed method, named AM-cIRM, includes a cIRM extractor and a VAD module. The cIRM extractor learns auxiliary features by estimating the magnitude and phase of clean speech, providing information that is complementary to commonly used acoustic features. Combining and exploiting that information from cIRM and other acoustic features, the VAD module determines which frequency and temporal components are more important for detection by applying attention mechanisms. We subsequently present an efficient transformer-based network, which includes a feature embedding module for effective feature extraction, several depth-wise transformer blocks, and a classifier. In contrast to the former method, the transformer-based method, which we called Tr-VAD, implements efficient operations on feature patches with the smallest possible number of parameters. Experimental results show that both proposed methods achieve improved VAD performance compared to baseline methods from the literature in low to moderate SNR environments. However, Tr-VAD is more efficient than AM-cIRM as it requires fewer network parameters to achieve a similar performance. The results also indicate that the use of additional audio fingerprinting features with Tr-VAD can guarantee better performance.

Sommaire

La détection d'activité vocale (VAD) est souvent traitée comme un problème de classification où le but est de discriminer, à un moment donné, entre la parole souhaitée et le bruit de fond. Bien que de nombreuses approches de pointe pour augmenter les performances de la VAD aient été proposées, elles ne sont toujours pas assez robustes pour être appliquées dans des conditions de bruit défavorables avec un faible rapport signal sur bruit (SNR). Dans ce travail, nous introduisons d'abord une nouvelle architecture de réseau de neurones profond à détection de phase basée sur un modèle d'attention pour VAD qui tire parti du masque de rapport idéal (cIRM) complexe. La méthode proposée, nommée AM-cIRM, comprend un extracteur cIRM et un module VAD. L'extracteur cIRM apprend les caractéristiques auxiliaires en estimant l'amplitude et la phase de la parole non-bruitée, fournissant des informations complémentaires aux caractéristiques acoustiques couramment utilisées. En combinant et en exploitant ces informations du cIRM et d'autres caractéristiques acoustiques, le module VAD détermine quelles fréquences et quelles composantes temporelles sont les plus importantes pour la détection en appliquant des mécanismes d'attention. Nous présentons ensuite un réseau efficace basé sur des transformateurs, qui comprend un module d'intégration de caractéristiques pour une extraction efficace des caractéristiques, plusieurs blocs de transformateurs en profondeur et un classificateur. Contrairement à la première méthode, la méthode basée sur le transformateur, que nous avons appelée Tr-VAD, implémente des opérations efficaces sur des portions de descripteurs avec le plus petit nombre possible de paramètres. Les résultats expérimentaux montrent que les deux méthodes proposées permettent d'obtenir des performances VAD améliorées par rapport aux méthodes de base de la littérature dans des environnements à faible SNR. Cependant, Tr-VAD est plus efficace que AM-cIRM car il nécessite moins de paramètres réseau pour obtenir des performances similaires. Les résultats indiquent également que l'utilisation d'empreintes audio comme descripteurs additionnels avec Tr-VAD peut garantir de meilleures performances.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my research supervisor Prof. Benoit Champagne for giving me the opportunity to do research and providing invaluable guidance through this research. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. I also appreciate the patient guidance and suggestions from Dr. Yazid Attabi (post-doctoral fellow), who provided constructive and useful comments throughout my research work.

I am extremely grateful to my beloved parents for their love, prayers, caring and sacrifices for educating and preparing me for my future.

This journey in Montreal would not have been the same without the encouragement from my friend Yuntian Zhang who always cheers me up when I am down. Last but not least, I also appreciate the support from other friends, you make my journey colorful.

Contents

List of Figures

List of Tables

List of Acronyms

AFPC	Audio Fingerprinting Combination
ASR	Automatic Speech Recognition
bDNN	boosted Deep Neural Network
CE	Cross Entropy
cIRM	complex Ideal Ratio Mask
DBNs	Deep Belief Networks
DCF	Detection Cost Function
DCT	Discrete Cosine Transform
DNNs	Deep Neural Networks
DW	Depth-Wise Convolution
FCN	Fully Connected Network
FFN	Feedforward Network
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model

IBM	Ideal Binary Mask
IRM	Ideal Ratio Mask
ISTFT	Inverse Short-Time-Fourier-Transform
LRT	Likelihood Ratio Test
MFCC	Mel-Frequency Cepstral Coefficients
MHSA	Multi-Headed Self Attention
MLP	Multilayer Perceptron
MSNE	Minimum Statistics Noise Estimation
NSSC	Normalized Spectral Subband Centroids
SE	Speech Enhancement
SNR	Signal to Noise Ratio
SSC	Spectral Subband Centroids
SVM	Support Vector Machines
VAD	Voice activity detection
VoIP	Voice over Internet Protocol
wSDR	weighted-Source-to-Distortion Ratio loss
ZCR	Zero Crossing Rate

Chapter 1

Introduction

This chapter provides a general overview of the thesis. We first introduce the problem of Voice Activity Detection (VAD) and present a literature survey on representative methods aiming at solving the problem. Then the research objectives and main contributions of the thesis are summarized. Finally, the organization of the upcoming chapters is outlined.

1.1 Overview of Voice Activity Detection

Voice activity detection, also known as speech endpoint detection, refers to a family of methods that classify frames of audio signals into speech and non-speech. Given the recent growth of Voice over Internet Protocol (VoIP) applications and the burst of connected devices that are increasingly getting voice calling functionality, saving on computation and on network bandwidth are becoming more and more important. As VAD determines the presence or absence of human voice, it can facilitate speech processing and avoid unnecessary coding/transmission of non-speech segments, such as silence and background noises. Thus, VAD often serves as an important preprocessor for many speech-related applications including speaker recognition, automatic speech recognition, keyword spotting, and hearing aids [1], [2].

The primary difficulty in developing VAD systems lies in distinguishing between the desired speech signal from a wide variety of stationary and non-stationary noise backgrounds. Ideally, a robust VAD system should be independent of language and should still provide good performance in low SNR environments. In practice, speech signals often contaminated by high level of noise (i.e. low SNR) which brings great challenges for the accurate operation of a VAD system.

1.2 Literature Review

With the continuous development of speech technologies, the research on VAD has become a topic of continuing interest. Conventional VAD methods were based on power calculations in the time domain [1], [3] under the assumption that the speech power is greater than the noise power. Other methods were subsequently developed that rely on the use of classical or handcrafted features of speech signals, such as zero crossing rate (ZCR) [4], spectral or cepstral features [4], [5], pitch detection [3], [6], and higher order statistics [7]. For example, pitch information plays an important role in rVAD [3] which uses pitch as an anchor to locate potential speech regions. However, these methods only reflect partial characteristics of human voice, and may be ineffective in some complex scenarios when used alone. In the past, several VAD methods have also been developed based on the likelihood ratio test (LRT) [8], assuming *a prior* knowledge of the speech signal and noise distributions. Typical model distributions used in this context include Gaussian [9], [10], Laplacian [11], Gamma distributions [12] and so.

However, these methods use limited and temporally localized data sets to estimate model parameters, and require considerable understanding of the audio environment for proper choice of models and estimation of associated parameters. Although these conventional methods perform reasonably well in some scenarios, they have difficulty handling a wide variety of speech signals from different speakers and under diverse range of real-world noise environments. Especially, real-world data distributions may be too complicated to be modeled accurately by a predefined mathematical model.

Recently, machine learning methods have demonstrated good classification results for the purpose of VAD. They are flexible in incorporating prior knowledge, such as manually labeled data, and also good at extracting multiple acoustic features. Existing supervised models for classification include linear discriminant analysis [13], support vector machines (SVM) [14], sparse coding [15], Gaussian Mixture Model (GMM) [16], Hidden Markov Model (HMM) [17], and especially deep neural networks (DNNs). Recognizing DNNs' unprecedented effectiveness, many researchers have proposed DNN-based methods and have shown superior performance over conventional ones. In contrast to classic methods, these approaches learn to implicitly model data without assuming an explicit model of noisy speech signals. One strength of DNN methods is their flexible capture of speech variability by using non-linear transformation functions.

To detect voice activity, Deep Belief Networks (DBNs) [18] were applied and outperformed the conventional SVM-based VAD. Recurrent Neural Networks (RNNs) [19] were also successfully applied to VAD. However, RNNs suffer from state saturation problems when the utterance is long [20]. A combined end-to-end VAD system is introduced by [21] which utilizes WaveNet-based network [22] for acoustic feature extraction and a deep residual network for video feature extraction. Reference [23] proposes a three-tiered model for boosting contextual information by incorporating boosted Deep Neural Networks (bDNN) with Multi-Resolution Stacking (MRS) and MultiResolution CochleaGram (MRCG) features. Inspired by the results of [24], [25] showing that auxiliary features, such as phoneme information, can improve speech enhancement performance, reference [26] shows the improvement in the performance of DNN-based VAD by using auxiliary features output from two types of auxiliary speech models.

More recently, there has been a growing interest in the use of attention mechanism for ASR applications [27], [28]. Inspired by their effectiveness, the attention models have also been applied to the VAD task. The Adaptive Context Attention Model (ACAM), proposed by [29], adopts an attention mechanism to exploit temporal information. However, ACAM's reinforcement loss function often tends to make the model training unstable and sensitive to hyperparameters. Reference [30] further improves the VAD performance by applying attention mechanisms to both contextual and spectral information.

1.3 Thesis Contributions

To further improve the robustness in noisy environments, we firstly propose a novel attention model-based phase-aware deep neural network architecture for VAD, called AM-cIRM, which takes advantages of complex Ideal Ratio Masks (cIRM). consists of three modules: a cIRM extractor, a feature transformation module, and an attention model-based VAD module. Combining phase and magnitude information from the cIRM and noisy spectrogram as input, the detector exploits useful information and makes decisions by considering which part of the features is more important using attention modules. Secondly, we further proposed a transformer-based DNN architecture for VAD, refer to as Tr-VAD. This second method implements efficient convolution operations on split patches of input features. It contains an embedding layer for feature extraction, several depth-wise transformer blocks, and a classifier. To the best of our knowledge, this is the first attempt to adopt transformer-based architecture for the VAD task. The performance of the two proposed methods is evaluated by means of the standard F1 score and Detection Cost Function (DCF), using a data set comprised of a wide variety of speech signals contaminated by different types and levels of noise. The experimental results confirm that the proposed methods can achieve notable performance improvements over state-of-the-art VAD approaches.

1.4 Organization

The remaining part of the thesis is organized as follows. Chapter 2 describes background theory including: the acoustic features used in our work, a recent but non-DNN-based VAD method, and a state-of-the-art DNN-based VAD method. Chapter 3 presents the architecture and components of the proposed AM-cIRM while Chapter 4 discusses the architecture of the Tr-VAD model. Chapter 5 describes the experimental setup and compares the performance of different models. Finally, a summary of our work along with concluding remarks are presented in Chapter 6.

Chapter 2

Background Theory

In this chapter, we present the definitions of the acoustic feature vectors used in our work as well as a brief overview of two VAD methods, i.e., the unsupervised segment-based method for robust Voice Activity Detection (rVAD) [3] and the Spectro-Temporal Attention-based VAD model (STAM) [30]. Both rVAD and STAM will later serve as benchmarks.

2.1 Acoustic Features for VAD

The input noisy speech signal $x[n]$ is modeled as:

$$x[n] = s[n] + w[n] \tag{2.1}$$

where $s[n]$ denotes the clean speech signal, $w[n]$ denotes the noise signal, and $n \in \mathbb{Z}$ is the discrete-time index.

To extract acoustic features, the entire audio samples are first pre-emphasized via linear filtering to boost the highband formants:

$$\bar{x}[n] = x[n] - \alpha x[n-1] \tag{2.2}$$

where α is the pre-emphasis coefficient, with $0.95 \leq \alpha \leq 1$.

Then Short-Time-Fourier-Transform (STFT) is applied to the pre-emphasized signal $\bar{x}[n]$ as follows:

$$X(t,f) = \sum_{n=0}^{N-1} \bar{x}\left[n + tL_{\text{hop}}\right] h[n] e^{-j2\pi f n/N} \tag{2.3}$$

where t is the frame index, L_{hop} is the frame advance, $f \in \{0, 1, 2, \dots, N/2\}$ is the frequency bin index, N is the window size and $h[n]$ is a window function.

The power of the transformed output $|X(t,f)|^2$ is then warped according to the Mel scale in order to adapt the frequency resolution to the properties of the human ear. The corresponding frequency transformation can be expressed as: $\phi_f = 2595 \log_{10}\left(1 + \frac{l_f}{700}\right)$, where ϕ_f is the Mel frequency and l_f is the linear frequency in Hz. Warping of the power spectrum can be realized by applying a series of overlapping triangular spectral shaping filters to the power spectrum. These filters are defined by their center frequencies and width, the former being obtained by applying the transformation ϕ_f to a set of uniformly spaced frequencies. Finally, the logarithm function is applied to the output of each filter:

$$\text{FBank}(t,b) = 20\log_{10}\left\{\sum_{f=l_b}^{h_b} u_b(f)|X(t,f)|^2\right\} \tag{2.4}$$

where $b \in \{0,1, \dots, B-1\}$ is the filter index, B is the number of triangular filters in the Mel filter bank, $u_b(f)$ is the spectral shaping filter of the b^{th} subband, l_b and h_b are the lower and upper Mel frequency limits of $u_b(f)$ respectively. The vector of log-Mel filter bank features computed at the t^{th} frame is denoted as:

$$\mathbf{FBank}_{\text{t}} = [\text{FBank}(t,0), \dots, \text{FBank}(t,b), \dots, \text{FBank}(t,B-1)] \tag{2.5}$$

The Discrete Cosine Transform (DCT) - Type III [31] is applied to the log-Mel filter bank features to obtain Mel-Frequency Cepstral Coefficients (MFCC):

$$\mathrm{MFCC}(t,b') = \sqrt{\frac{2}{B}}\sum_{b=0}^{B-1}\mathrm{FBank}(t,b)\cos\left(\frac{p\pi}{B}(b-0.5)\right) \tag{2.6}$$

where $b' \in \{0,1,\dots,B'-1\}$ and B' is the number of coefficients. We define the MFCC feature vector of the current data frame as:

$$\mathbf{MFCC}_{\mathrm{t}} = [\mathrm{MFCC}(t,0),\dots,\mathrm{MFCC}(t,B'-1)] \tag{2.7}$$

The Spectral Subband Centroids (SSC), introduced in [32], are often used to measure the center of mass of a subband spectrum in terms of frequency. The SSC have demonstrated robustness against the equalization, data compression and additive noise, as these modifications do not significantly alter the peak frequencies at moderate to high SNR [33]. It was also reported that the SSC-based features resulted in higher audio recognition accuracy than MFCC-based ones. To calculate SSC, a weighted average technique using a bank of spectral weighting filters is applied as follows:

$$\mathrm{SSC}(t,b) = \frac{\sum_{f=l_b}^{h_b} f u_b'(f)|X(t,f)|^2}{\sum_{f=l_b}^{h_b} u_b'(f)|X(t,f)|^2} \tag{2.8}$$

where $u_b'(f)$ is the corresponding subband filter. For simplicity, in this thesis, we use the same set of filters $u_b(f)$ for the calculation of the MFCC and SCC features.

The SSC are usually normalized to the range $[-1,1]$ for efficient training. Especially, the Normalized SSC (NSSC) features are obtained as:

$$\mathrm{NSSC}(t,b) = \frac{\mathrm{SSC}(t,b) - (h_b - l_b)}{h_b - l_b} \tag{2.9}$$

Similarly, we define the NSSC feature vector of signal $x[n]$ at the t^{th} frame as:

$$\mathbf{NSSC}_{\mathrm{t}} = [\mathrm{NSSC}(t,0), \dots, \mathrm{NSSC}(t,B-1)] \tag{2.10}$$

In [34], a combination of MFCC and NSCC used as inputs to a generative adversarial network has demonstrated superior performance for speech enhancement; in out work, we shall make use of a similar concatenation of features. The resulting Audio Fingerprinting Combination (AFPC) at the t^{th} frame is defined as:

$$\mathbf{AFPC}_{\mathrm{t}} = [\mathbf{MFCC}_{\mathrm{t}}, \Delta\mathbf{MFCC}_{t}, \Delta^2\mathbf{MFCC}_{\mathrm{t}}, \mathbf{NSSC}_{\mathrm{t}}, \Delta\mathbf{NSSC}_{\mathrm{t}}, \Delta^2\mathbf{NSSC}_{\mathrm{t}}] \tag{2.11}$$

where Δ and Δ^2 are the delta and double-delta operations, respectively.

Conventional VAD systems usually consist of feature extraction module, decision making module and decision smoothing (or hangover scheme). These algorithms use hangover scheme as a post processing step to refine the decision boundaries. *J. Sohn et al*. [9] applied a hangover scheme to prevent the clipping of weak speech tails, this scheme is based on a HMM whereby the speech decision of a current frame only depends on the current frame and the previous frame. A hangover scheme which simply delayed the transition from a speech declaration to a non-speech declaration was also implemented by *D. Ying et al.* [10] to account for the low energy regions of the tail end of utterances.

2.2 Classical Method: Robust VAD

The main steps of the rVAD method [3] are summarized in Fig. 2.1. The method includes two passes of denoising followed by a VAD stage. In the first pass, the smoothed $a\ posteriori$ SNR weighted energy difference of two consecutive frames is calculated. This is done through the following steps:

1) Calculate the *a posteriori* SNR weighted energy difference of two consecutive frames as:

$$d(t) = \sqrt{|E_x(t) - E_x(t-1)| \max(\mathrm{SNR}_{\mathrm{post}}(t), 0)} \tag{2.12}$$

where $E_x(t)$ is the energy of the t^{th} frame of noisy speech $x[n]$, and $\mathrm{SNR}_{\mathrm{post}}(t)$ is *a posteriori* SNR that is calculated as the logarithmic ratio of $E_x(t)$ to the estimated energy of the t^{th} frame of noisy speech $w[n]$:

$$\mathrm{SNR}_{\mathrm{post}}(t) = 10 \log_{10} \frac{E_x(t)}{\bar{E}_w(t)} \tag{2.13}$$

In Eq. (2.13), the energy of the noisy signal is calculated as $E_x(t) = \sum_n x[n]^2$, where the sum extends over all samples in the t^{th} frame. The noise energy $\bar{E}_w(t)$ is estimated as follows. First, the speech signal $x[n]$ is split into non-overlapping super segments of $M^s = 200$ frames: $\boldsymbol{x}(p) = \boldsymbol{s}(p) + \boldsymbol{w}(p), p = 1, \dots, P$, where P is the number of super-segments in an utterance. For each super-segment $\boldsymbol{x}(p)$, the noise energy $E_w^s(p)$ is calculated as the energy of the frame ranked at 10% of lowest energy within the super-segment. The noise energy is then smoothed as follows:

$$\bar{E}_w^s = 0.9\bar{E}_w^s(p-1) + 0.1E_w^s(p) \tag{2.14}$$

Smoothed noise energy at the t^{th} frame, $\bar{E}_w(t)$, is taken as the energy value $\bar{E}_w^s(p)$ of the p^{th} super segment to which the t^{th} frame belongs to.

2) Calculate the central-smoothed *a posteriori* SNR weighted energy difference:

$$\bar{d}(t) = \frac{1}{2K+1}\sum_{i=-K}^{K} d(t+i) \tag{2.15}$$

where K is the smoothing parameter, i.e., number of frames considered on each side of the current frame t, and i is the frame index relative to the t.

3) Classify a frame as a high-energy frame if $\bar{d}(t)$ is greater than a threshold $\theta_r(t)$. For each super segment p, $\theta_r^s(p)$ is computed as follows:

$$\theta_r^s(p) = \alpha^s \max\{E_x(M^s(p-1)+1), \ldots, E_x(t) \ldots, E_x(M^s p)\} \tag{2.16}$$

where α^s is the scale factor set to 0.25. $\theta_r(t)$ is taken as the threshold value $\theta_r^s(p)$ of the p^{th} super segment to which the t^{th} frame belongs.

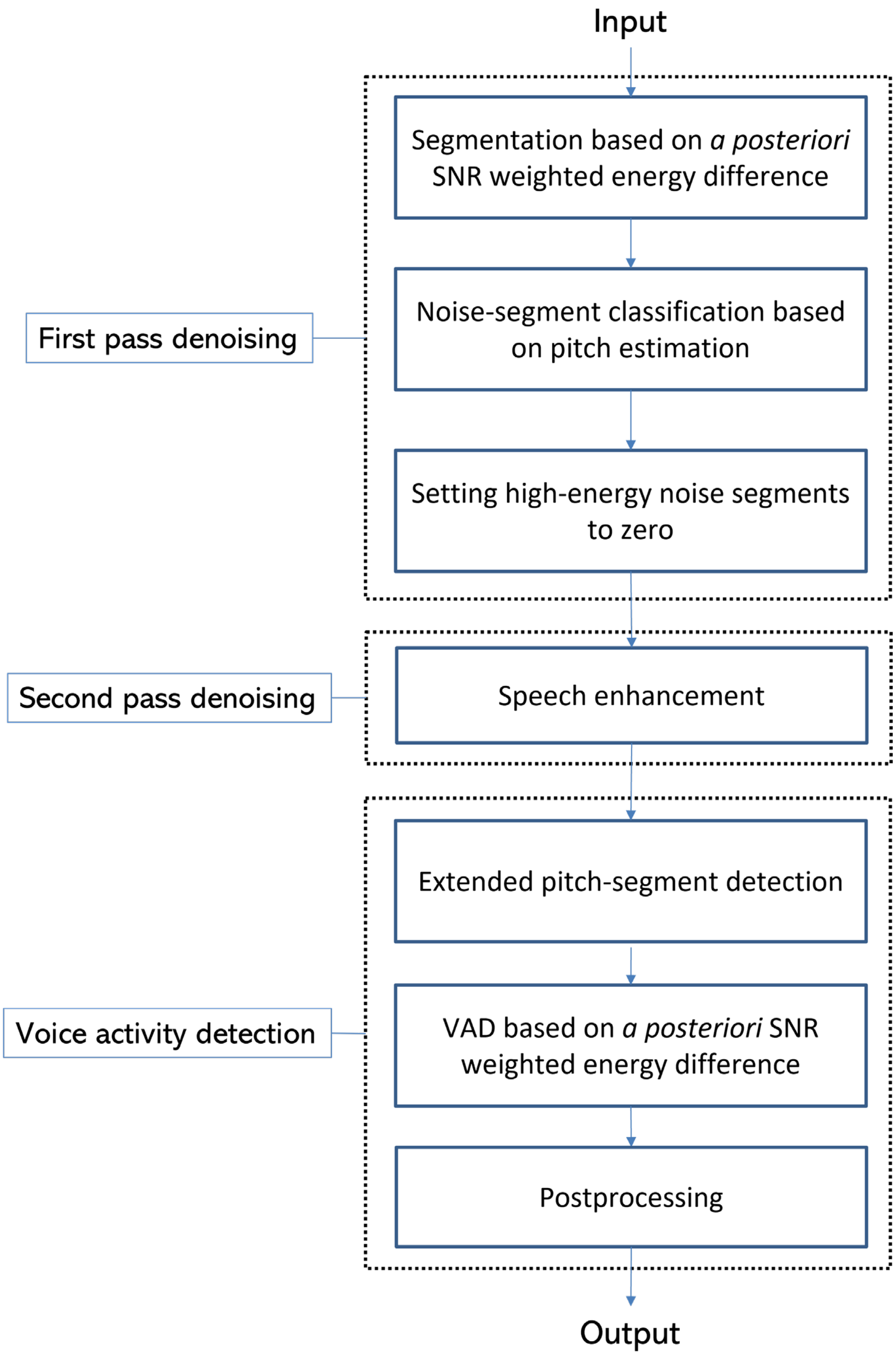
Input
Segmentation based on *a posteriori* SNR weighted energy difference
Noise-segment classification based on pitch estimation
Setting high-energy noise segments to zero
First pass denoising
Speech enhancement
Second pass denoising
Extended pitch-segment detection
VAD based on *a posteriori* SNR weighted energy difference
Postprocessing
Voice activity detection
Output

Fig. 2.1 Processing flow of rVAD method.

4) Consecutive high-energy frames are grouped together to form high-energy segments.
5) Within a high-energy segment, if no more than two pitch frames are found, the segment is considered as noise segment and frames in the segment are set to zero.

In the second pass, the speech signal is denoised by applying a modified version of the Minimum Statistics Noise Estimation (MSNE) method, which is a type of spectral subtraction [35]. The unbiased noise power estimation in the conventional MSNE can be expressed as:

$$\hat{\lambda}_w(t,f) = B_{min}(t,f)\,\min\{P(t,f), P(t-1,f), \dots, P(t-l,f)\} \tag{2.17}$$

where f is the frequency bin index, $B_{min}(t,f)$ is the bias compensation factor, $P(t,f) = |X(t,f)|^2$ is the periodogram of input signal $x[n]$, and l is the number of previous frames used in the search of a minimum. In the modified MSNE, the noise power estimation $\hat{\lambda}_w(t,f)$ is not updated during the detected high-energy noise segments (which are set to zero). Besides, if more than half of the energy is located within the first few frequency bins, the values of these frequency bins are set to zero to remove low-frequency noise.

In the VAD stage, the rVAD method assumes that all speech segments should contain certain number of speech frames with pitch. Therefore, pitch frames are grouped together to form pitch segments, which are then further extended from both ends by 60 segments to include voiced, unvoiced sounds and likely non-speech parts, for reasons explained in [3]. Then the smoothed *a posteriori* SNR weighted energy difference $\bar{d}'(t)$ is calculated as in Eq. (2.12)-(2.15). The presence or absence of speech is decided by comparing the value of $\bar{d}'(t)$ with a threshold θ_r' which is calculated as follows:

$$\theta_r' = \beta_r \frac{1}{L_r} \sum_{j=1}^{L_r} \bar{d}'(j) \tag{2.18}$$

where L_r is the total number of pitch frames in the extended pitch segment and β_r is a scale factor set to 0.4. Finally, post-processing is applied to further refine the frame classification under the assumptions that speech frames should not be too far away from their closest pitch frame, and that within a speech segment, there should be a certain number of speech frames without pitch.

2.3 Spectro-Temporal Attention Model

In this section, the feature expansion strategy and the model architecture of the Spectro-Temporal Attention Model (STAM) [30] are presented.

2.3.1 Feature Expansion

Suppose we are given T pairs of acoustic feature vectors and classification labels, i.e., $\{\boldsymbol{X}_t, y_t^{\text{truth}}\}_{t=0}^{T-1}$, where $\boldsymbol{X}_t \in \mathbb{R}^D$ and $y_t^{\text{truth}} \in \{0,1\}$ are the acoustic feature vector with dimension D and VAD label for frame t respectively. The STAM exploits contextual information by using $L = \lfloor 2\big((R-1)/u\big) + 3 \rfloor$ neighboring frames indexed by $\mathcal{T} = \{-R, -R+\mu, -R+2\mu, \dots, -1,0,1, \dots, R-2\mu, R-\mu, R\}$, where integer R defines the radius of the context and integer μ, with $1 \le u \le R$, is a skip parameter. These frames are used to form the expanded data set $\{\boldsymbol{V}_t, \boldsymbol{y}_t^{\text{truth}}\}_{t=0}^{T-1}$:

$$\boldsymbol{V}_t = \{\boldsymbol{X}_{t+l}: l \in \mathcal{T}\} \in \mathbb{R}^{L\times D}, \quad \boldsymbol{y}_t^{\text{truth}} = \{y_{t+l}^{\text{truth}} : l \in \mathcal{T}\} \in \mathbb{R}^{L} \tag{2.19}$$

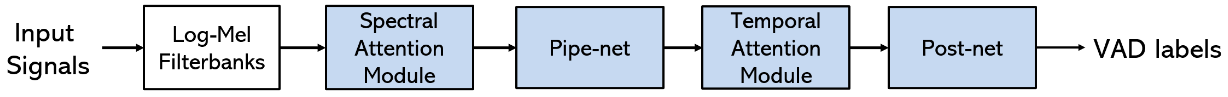

Fig. 2.2 Model architecture of STAM.

2.3.2 Model architecture of STAM

The STAM includes a spectral attention module, a pipe-net, a temporal attention module, and a post-net as shown in Fig. 2.2. The purposes of these modules are explained as follows:

- **Spectral Attention**: As illustrated in Fig. 2.3, this module consists of a cascade of N_{spec} blocks with each block composed of a pair of $k_{spec} \times k_{spec}$ convolutional filters and a 1-D max pooling layer which is applied along the frequency axis. Each of the convolutional layers in the first block contains N_{c} convolutional filters and the number of filters N_{c} is doubled after each block, which is repeated N_{spec} times. For example, the output sizes of the first two blocks are $\frac{D}{2} \times L \times N_{\mathrm{c}}$ and $\frac{D}{4} \times L \times 2N_{\mathrm{c}}$ respectively. Each block produces several mask matrices with each element of which outputs a number between zero and one. These mask metrics are directly multiplied pointwise with another spectral feature map which indicates how much of each spectral component should be attended by VAD.

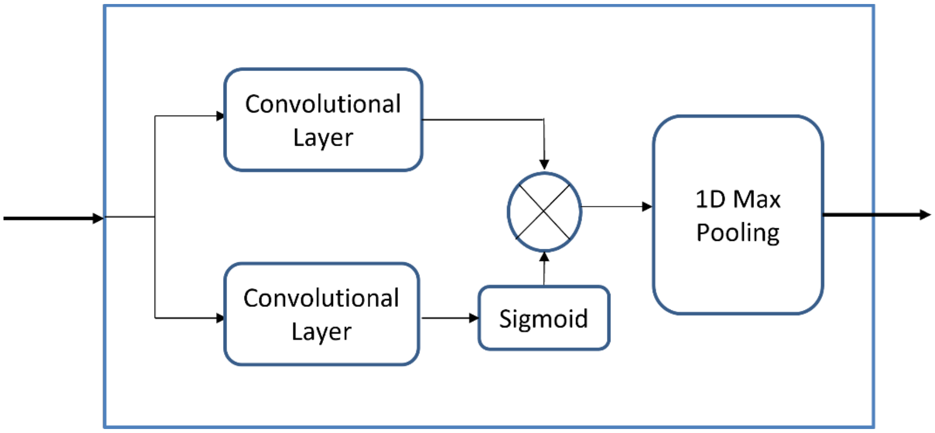

Fig. 2.3 Spectral attention block.

- **Pipe-Net**: The pipe-net contains two Fully Connected Networks (FCN)s with each FCN consists of a linear layer followed by a dropout layer and an activation layer [30]. This module acts as an information bridge between the spectral attention module and the temporal attention module. The output of the pipe-net is represented by matrix $\boldsymbol{G} \in \mathbb{R}^{N_{pipe} \times L}$, where N_{pipe} is the hidden dimension. Another FCN with a single unit and sigmoid activation are applied to $\boldsymbol{G}$ to calculate the loss of pipe net. Note that this loss is for training only.
- **Temporal Attention**: The STAM adopts multi-headed self-attention allowing the model to simultaneously focus attention to information at different positions. The query $\boldsymbol{q}$, key $\boldsymbol{K}$ and value $\boldsymbol{V}$ are calculated using the pipe-net output $\boldsymbol{G}$ as follows:

$$\boldsymbol{q} = \sigma\left(\boldsymbol{W}_q \boldsymbol{g}\right) \in \mathbb{R}^{N_d} \tag{2.20}$$

$$\boldsymbol{K} = \sigma(\boldsymbol{W}_K \boldsymbol{G}) \in \mathbb{R}^{N_d \times L} \tag{2.21}$$

$$\boldsymbol{V} = \sigma(\boldsymbol{W}_V \boldsymbol{G}) \in \mathbb{R}^{N_d \times L} \tag{2.22}$$

where N_d denotes the attention dimension, σ is an activation function, and $\boldsymbol{g} \in \mathbb{R}^{N_{pipe}}$ is obtained by averaging $\boldsymbol{G}$ along the frame dimension (i.e., over the column index L). $\boldsymbol{W}_q$, $\boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{N_d \times N_{pipe}}$ are the affine transformation matrices.

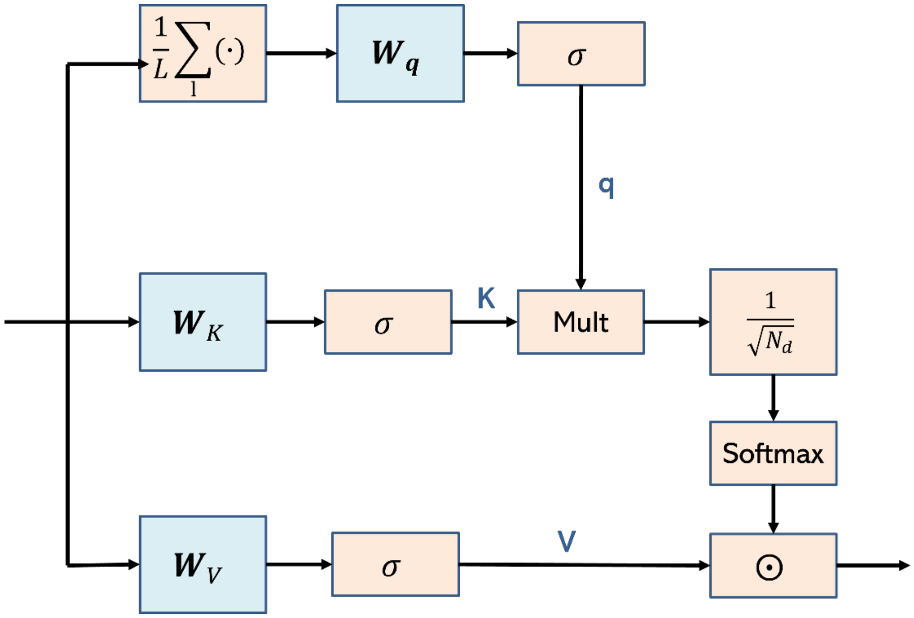

Fig. 2.4 Single-headed attention function in STAM.

The attention function calculates the so-called attention vector for multiple frames by means of a softmax operation as follows:

$$\text{Attention}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{q}^T\boldsymbol{K}}{\sqrt{N_d}}\right) \cdot \boldsymbol{V} \tag{2.23}$$

where $\cdot$ is the element-wise product. These operations are presented in block-diagram form in Fig. 2.4.

To allow the model to attend to multiple frames instead of a single frame, the multi-headed attention operation is used:

$$\text{MultiHead}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) \tag{2.24}$$

$$\text{head}_h = \text{Attention}(\boldsymbol{q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) \tag{2.25}$$

where H is the number of parallel attention layers, or heads, $\boldsymbol{q}_h$, $\boldsymbol{K}_h$ and $\boldsymbol{V}_h$ are the h^{th} components of $\boldsymbol{q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$, respectively, and $h \in \{1, \dots, H\}$ is the head index.

Similar to the pipe net, another FCN with a single unit and sigmoid activation are applied to the output of the multi-head attention module to calculate the loss of pipe net. This loss is for training only too.

- **Post-Net**: The post-net includes two FCNs followed by a sigmoid activation to predict the probability of the presence of speech. The predicted t^{th} frame label, $\hat{y}_t$, can be computed by aggregating all the soft predictions $\boldsymbol{y}_t = \{y_{t+l}: l \in \mathcal{T}\} \in \mathbb{R}^L$, where y_{t+l} is the soft prediction for the $(t+l)^{th}$ neighboring frame, relative to the current frame t across l:

$$\hat{y}_t = \frac{1}{L}\sum\nolimits_{l\in\mathcal{T}} y_{t+l} \tag{2.26}$$

The final decision label $\bar{y}_t$ is obtained by comparing the $\hat{y}_t$ with a threshold θ_{VAD}:

$$\bar{y}_t = \begin{cases} 1, & \text{if } \hat{y}_t \geq \theta_{\text{VAD}} \\ 0, & \text{otherwise} \end{cases} \tag{2.27}$$

2.4 Summary

In this Chapter, we have presented background material, including the definitions of acoustic feature vectors used in the work, as well as the underlying principles of the unsupervised rVAD method and the supervised attention-based STAM model for VAD. This material will be used in the following chapters to develop and evaluate the performance of the newly proposed VAD schemes, including the cIRM-based AM-cIRM and transformer-based Tr-VAD models.

Chapter 3

Complex IRM-Aware Training for VAD using Attention Model

In this chapter, we propose a novel attention model-based phase-aware VAD method, called AM-cIRM, which takes advantages of complex-valued Ideal Ratio Masks (cIRM) and uses attention mechanisms to focus on more important information of acoustic features.

3.1 Overview of the Proposed System

The proposed AM-cIRM model, whose block diagram is shown in Fig. 3.1, consists of three modules: a cIRM extractor, a feature transformation module, and an attention-based VAD module. The cIRM extractor processes input features obtained by applying STFT to the noisy input signal, and outputs the cIRM. The feature transformation module acts like a preprocessor for the VAD module, producing transformed features that can be better processed by the following VAD module. Specifically, the magnitude and phase information of the cIRM is aggregated[1] by applying a linear transformation to them. The transformed cIRM are concatenated with noisy log-Mel spectrogram features and then expanded by incorporating contextual information in order to form

[1] Feature aggregation refers to a family of methods that combine features to form new ones, usually with reduced dimensionality.

final feature tensor. Taking the tensor as input, the VAD module outputs speech/non-speech predictions by applying attention mechanisms to the spectral and temporal information contained in the final feature tensor. These modules are explained in greater details below, along with a discussion of the loss function used for training the model.

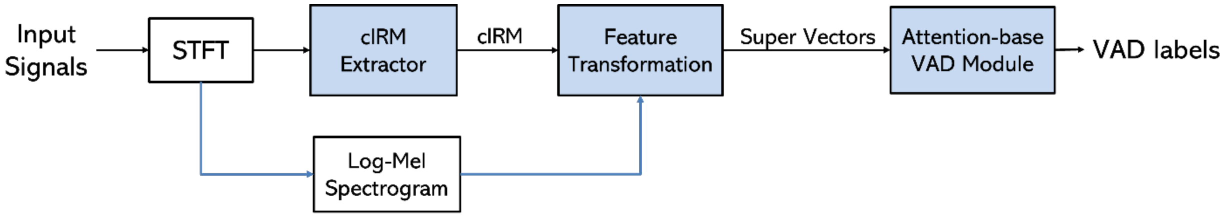

Fig. 3.1 Model architecture of the AM-cIRM.

3.2 cIRM Feature Extractor

Similar to VAD, Speech Enhancement (SE) has been widely used as a prepossessing step in speech applications where one of the goals is to remove background noise from a noisy speech signal. Besides the classical SE methods based on statistical modeling, e.g. [36], many recent studies have focused on DNN-based SE methods. Among the later, the Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM)-based [37] methods have shown excellent SE performance. However, such DNN methods overlook phase information as they reconstruct the estimated clean speech by employing the phase of the noisy input speech instead of estimating the magnitude and phase of the clean speech simultaneously. To alleviate this problem, the complex-valued Ideal Ratio Masks (cIRM), estimated by using the U-Net [38]-based complex-valued network DCUnet [39], has been proposed and shown better SE performance compared to the earlier IBM and IRM-based methods. Inspired by the effectiveness of the cIRM-based DNN methods in extracting important speech information from noisy speech signals, the DCUnet is chosen as the cIRM extractor for the proposed AM-cIRM model in this thesis.

Before getting into details of the cIRM extractor, we first introduce some notations used throughout the Chapter. We denote by $\boldsymbol{x}_t = \left[x\left[tL_{\text{hop}}\right], \dots, x\left[tL_{\text{hop}} + N - 1\right]\right]$ the vector of noisy speech samples corresponding to the t^{th} frame, $t \in \{0,1,\dots,T-1\}$, L_{hop} is the frame advance and T is the total number of frames. The input features of the cIRM extractor, represented by $\boldsymbol{X}_{\text{STFT}} \in \mathbb{C}^{F\times T}$, where superscript F denotes the number of frequency bins, are obtained by applying the STFT mentioned in Section 2.1 to the noisy signal frames $\boldsymbol{x}_t$. The ground truth cIRM, denoted as $\boldsymbol{M}_{\text{STFT}} \in \mathbb{C}^{F\times T}$, is obtained by element-wise division of the clean spectra by the noisy spectra, i.e., $\boldsymbol{M}_{\text{STFT}} = \boldsymbol{S}_{\text{STFT}}/\boldsymbol{X}_{\text{STFT}}$, where $\boldsymbol{S}_{\text{STFT}} \in \mathbb{C}^{F\times T}$ are the STFT coefficients of clean speech frames $\boldsymbol{s}_t = \left[s\left[tL_{\text{hop}}\right], \dots, s\left[tL_{\text{hop}} + N - 1\right]\right]$, and $/$ denotes the element-wise division. The estimated cIRM are represented as $\widehat{\boldsymbol{M}}_{\text{STFT}} \in \mathbb{C}^{F\times T}$. The enhanced speech frames, denoted as $\hat{\boldsymbol{s}}_t = \left[\hat{s}\left[tL_{\text{hop}}\right], \dots, \hat{s}\left[tL_{\text{hop}} + N - 1\right]\right]$, are estimated by applying the Inverse Short-Time-Fourier-Transform (ISTFT) to the STFT coefficients of the enhanced speech $\widehat{\boldsymbol{S}}_{\text{STFT}} \in \mathbb{C}^{F\times T}$ which are obtained by multiplying the estimated cIRM $\widehat{\boldsymbol{M}}_{STFT}$ with the noisy input $\boldsymbol{X}_{\text{STFT}}$:

$$\widehat{\boldsymbol{S}}_{r,\text{STFT}} = \widehat{\boldsymbol{M}}_{r,\text{STFT}} \cdot \boldsymbol{X}_{r,\text{STFT}} - \widehat{\boldsymbol{M}}_{i,\text{STFT}} \cdot \boldsymbol{X}_{i,\text{STFT}} \tag{3.1}$$

$$\widehat{\boldsymbol{S}}_{i,\text{STFT}} = \widehat{\boldsymbol{M}}_{r,\text{STFT}} \cdot \boldsymbol{X}_{i,\text{STFT}} + \widehat{\boldsymbol{M}}_{i,\text{STFT}} \cdot \boldsymbol{X}_{r,\text{STFT}} \tag{3.2}$$

where $\cdot$ is the element-wise product, subscripts r and i denote real and imaginary components, e.g., $\widehat{\boldsymbol{S}}_{\text{STFT}} = \widehat{\boldsymbol{S}}_{r,\text{STFT}} + j\widehat{\boldsymbol{S}}_{i,\text{STFT}}$.

The block diagram of the DCUnet-based [39] cIRM feature extractor is shown in Fig. 3.2, where each encoder block of the complex-valued network consists of a complex convolutional layer, a complex batch normalization layer, and a Leaky Rectified Linear Unit (Leaky ReLU). In the decoding phase, skip connections are implemented by concatenating the outputs from the last decoder and corresponding encoder. The decoder is similar to the encoder except that the complex convolutional layer is replaced by a complex transposed convolutional layer.

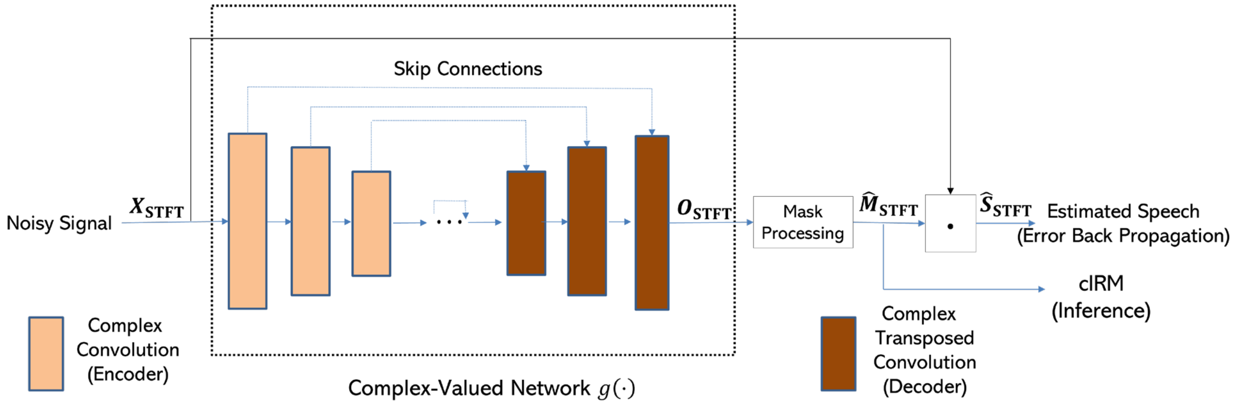
Skip Connections
Noisy Signal
$\boldsymbol{X}_{\text{STFT}}$
$\boldsymbol{O}_{\text{STFT}}$
Mask Processing
$\widehat{\boldsymbol{M}}_{\text{STFT}}$
$\widehat{\boldsymbol{S}}_{\text{STFT}}$
Estimated Speech
(Error Back Propagation)
cIRM
(Inference)
Complex Convolution (Encoder)
Complex-Valued Network $g(\cdot)$
Complex Transposed Convolution (Decoder)

Fig. 3.2 Illustration of the DCUnet-based cIRM extractor.

Given a complex-valued convolutional filter $\boldsymbol{W} = \boldsymbol{A} + j\boldsymbol{B}$ with real-valued matrix components $\boldsymbol{A}$ and $\boldsymbol{B}$, the 2-D complex convolution of a complex-valued input matrix $\boldsymbol{H} = \boldsymbol{C} + j\boldsymbol{D}$ with $\boldsymbol{W}$ is realized as:

$$\boldsymbol{W} * \boldsymbol{H} = (\boldsymbol{A} * \boldsymbol{C} - \boldsymbol{B} * \boldsymbol{D}) + j(\boldsymbol{B} * \boldsymbol{C} + \boldsymbol{A} * \boldsymbol{D}) \qquad (3.3)$$

where $*$ denotes 2-D convolution for real-valued matrices. Hence, complex convolution can be implemented by means of 4 real-valued 2-D convolutions with shared real-value filters $\boldsymbol{A}$ and $\boldsymbol{B}$ [39].

The magnitude and phase components of the estimated cIRM are finally obtained by processing the output $\boldsymbol{O}_{\text{STFT}} = g(\boldsymbol{X}_{\text{STFT}}) \in \mathbb{C}^{F \times T}$ of the complex-valued network $g(\cdot)$. The cIRM magnitudes are first obtained by applying the element-wise hyperbolic tangent non-linearity function to the magnitude of the entries of matrix $\boldsymbol{O}_{\text{STFT}}$ in order to bound them to the interval $[0, 1)$:

$$\widehat{\mathbf{M}}_{\text{STFT}}^{\text{mag}} = \tanh(|\boldsymbol{O}_{\text{STFT}}|) \tag{3.4}$$

where $|\mathbf{O}_{\text{STFT}}|$ stands for the element-wise magnitude of $\mathbf{O}_{\text{STFT}}$. The cIRM phase are obtained from the phase of the corresponding entries in $\boldsymbol{O}_{\text{STFT}}$:

$$\widehat{\mathbf{M}}_{\text{STFT}}^{\text{phase}} = \boldsymbol{O}_{\text{STFT}}/|\boldsymbol{O}_{\text{STFT}}| \tag{3.5}$$

The estimated cIRM $\widehat{M}_{\text{STFT}}$ are then obtained by applying element-wise product to the new magnitude component and the original phase component:

$$\widehat{\boldsymbol{M}}_{\text{STFT}} = \widehat{\mathbf{M}}_{\text{STFT}}^{\text{mag}} \cdot \widehat{\mathbf{M}}_{\text{STFT}}^{\text{phase}} \tag{3.6}$$

3.3 Feature Transformation

As shown in Fig. 3.3, the feature transformation module consists of four steps. Firstly, the estimated mask $\widehat{\boldsymbol{M}}_{\text{STFT}}$ is transposed to $\widetilde{\boldsymbol{M}}_{\text{STFT}} \in \mathbb{C}^{T \times F}$ for convenience in subsequent developments. Secondly, the transposed matrix is then aggregated by applying a linear transformation. The linear transformation contains a 1-D convolutional layer aiming at compressing the magnitude and phase information of the cIRM. The output is the aggregated mask $\overline{\boldsymbol{M}} \in \mathbb{R}^{T \times D}$, where $D < F$ is the resulting feature dimension for each frame.

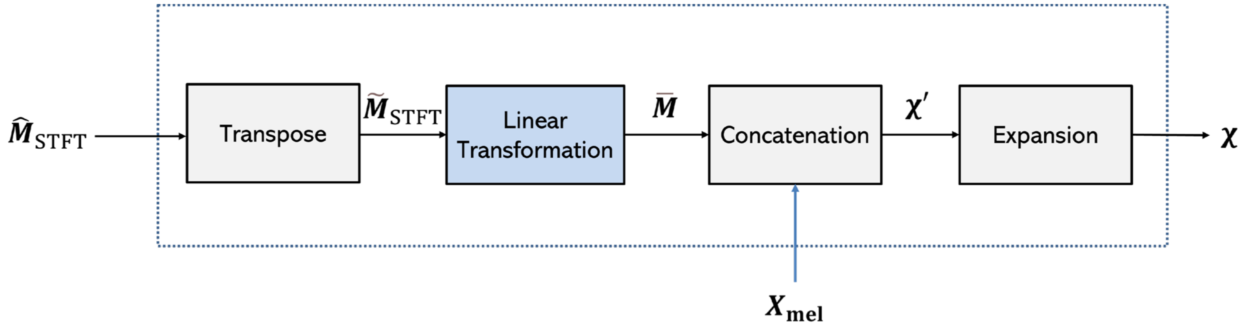
$\widehat{\boldsymbol{M}}_{\mathrm{STFT}}$
Transpose
$\widetilde{\boldsymbol{M}}_{\mathrm{STFT}}$
Linear Transformation
$\bar{\boldsymbol{M}}$
Concatenation
$\boldsymbol{\chi}'$
Expansion
$\boldsymbol{\chi}$
$\boldsymbol{X}_{\mathbf{mel}}$

Fig. 3.3 Block diagram of the feature transformation module.

Thirdly, to provide auxiliary information that is complementary to the spectrogram, the mask $\bar{\boldsymbol{M}}$ is concatenated with log-Mel spectrogram matrix $\boldsymbol{X}_{\mathrm{mel}} \in \mathbb{R}^{T\times D}$ to form a new matrix $\boldsymbol{\chi}' \in \mathbb{R}^{T\times 2D}$. Each row of matrix $\boldsymbol{X}_{\mathrm{mel}}$ is obtained by computing the log-Mel coefficients of the noisy-frame $\boldsymbol{x_t}$ using Mel filterbank consisting of D filters, as explained in Sections 2.

Finally, taking L neighboring frames into account, the matrix $\boldsymbol{\chi}'$ is expanded to form the final feature tensor $\boldsymbol{\chi} \in \mathbb{R}^{(T-2R)\times L\times 2D}$ of VAD module, by using L neighboring frames indexed by $\mathcal{T}$, where R, L, and $\mathcal{T}$ are the same parameters as defined in the construction of neighboring frames discussed in Section 2.3.1. The expanded data set can also be represented as $\{\boldsymbol{\chi}_t, \boldsymbol{y}_t^{\mathrm{truth}}\}_{t=R}^{T-R-1}$, where we define:

$$\boldsymbol{\chi}_t = \{\boldsymbol{\chi}'_{t+l} : l \in \mathcal{T}\} \in \mathbb{R}^{L\times 2D}, \qquad \boldsymbol{y}_t^{\mathrm{truth}} = \{y_{t+l}^{\mathrm{truth}} : l \in \mathcal{T}\} \in \mathbb{R}^{L} \tag{3.7}$$

where vector $\boldsymbol{\chi}'_{t+l} \in \mathbb{R}^{2D}$ contains the $(t+l)^{th}$ row of matrix $\boldsymbol{\chi}'$, and scalar y_{t+l}^{truth} is the ground truth VAD label for the $(t+l)^{th}$ frame.

3.4 Voice Activity Detector

Considering the effectiveness of the STAM model [30], it is chosen here as the VAD module of the AM-cIRM. Referring to the STAM model architecture in Fig. 2.2, since, since the feature dimension of the acoustic feature matrix $\boldsymbol{\chi}_t \in \mathbb{R}^{L\times 2D}$ is doubled in this work (due to the concatenation with the log-Mel spectrogram $\boldsymbol{X}_{\mathrm{mel}}$ and transformed mask $\bar{\boldsymbol{M}}$), the number of input channels of the first pair of convolution filters in the spectral attention module is also doubled, while the number of output channels N_c remains the same. The remaining three processing blocks of the STAM-based VAD module, i.e., the pipe-net, the temporal attention module, and the post-net, use the same parameter settings as the original STAM.

The post-net output can be represented by vector $\boldsymbol{y}_t = \{y_{t+l}: l \in \mathcal{T}\} \in \mathbb{R}^L$, where y_{t+l} is the soft label prediction for the $(t+l)^{th}$ neighboring frame. These soft predictions are finally aggregated into a single number:

$$\hat{y}_t = \frac{1}{L}\sum\nolimits_{l\in\mathcal{T}} y_{t+l} \tag{3.8}$$

The final VAD prediction for the t^{th} frame is obtained via the following test:

$$\bar{y}_t = \begin{cases} 1, & \text{if } \hat{y}_t \geq \theta_{\mathrm{cIRM}} \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

where $\theta_{\mathrm{cIRM}} \in (0,1)$ is the detection threshold, and $\bar{y}_t$ denotes the hard prediction at the t^{th} frame.

3.5 Loss Functions

The complete temporal sequence of estimated speech samples, represented by vector $\hat{\boldsymbol{s}} = [\hat{s}[0], \hat{s}[1], \dots, \hat{s}[N-1+L_{\mathrm{hop}}(T-1)]]$ is obtained by applying a modified overlap-add reconstruction equation from [40] to the estimated speech frames $\hat{\boldsymbol{s}}_t$ obtained by application of the ISTFT to the corresponding column of $\widehat{\boldsymbol{S}}_{\text{STFT}}$:

$$\hat{s}[n] = \frac{\sum_t \hat{\boldsymbol{s}}_t[n]h\big[n - tL_{\text{hop}}\big]}{\sum_t h^2\big[n - tL_{\text{hop}}\big]} \tag{3.10}$$

where $h[n]$ is the window function with $\sum_t h^2\big[n - tL_{\text{hop}}\big]$ is constant for all n . This reconstruction procedure needs not be applied to the noisy speech and clean speech since they are available as part of the training data. We let $\boldsymbol{s} = \big[s[0], s[1], \dots, s[N-1+L_{\text{hop}}(T-1)]\big]$ and $\boldsymbol{x} = \big[x[0], x[1], \dots, x[N-1+L_{\text{hop}}(T-1)]\big]$ denote the corresponding vectors. Finally, we introduce the time domain vectors $\boldsymbol{w} = \boldsymbol{x} - \boldsymbol{s}$ and $\widehat{\boldsymbol{w}} = \boldsymbol{x} - \hat{\boldsymbol{s}}$ containing the corresponding noise and estimated noise samples.

To prevent the vanishing gradients and accelerate convergence, we employ two different loss functions, one being calculated from the cIRM extractor and the other one from the VAD module. For the cIRM extractor, the weighted Source-to-Distortion Ratio loss (wSDR) proposed in [38] is calculated as follows:

$$\mathcal{L}_{wSDR}(\boldsymbol{x}, \boldsymbol{s}, \hat{\boldsymbol{s}}) = \alpha\, \mathcal{L}_{SDR}(\boldsymbol{s}, \hat{\boldsymbol{s}}) + (1-\alpha)\, \mathcal{L}_{SDR}(\mathbf{w}, \widehat{\mathbf{w}}) \tag{3.11}$$

$$\mathcal{L}_{SDR}(\mathbf{s}, \hat{\mathbf{s}}) = -\frac{\langle \mathbf{s}, \hat{\mathbf{s}} \rangle}{\parallel \mathbf{s} \parallel \parallel \hat{\mathbf{s}} \parallel}, \qquad \mathcal{L}_{SDR}(\mathbf{w}, \widehat{\mathbf{w}}) = -\frac{\langle \mathbf{w}, \widehat{\mathbf{w}} \rangle}{\parallel \mathbf{w} \parallel \parallel \widehat{\mathbf{w}} \parallel} \tag{3.12}$$

where $\alpha = \frac{\|\boldsymbol{s}\|^2}{\|\boldsymbol{s}\|^2 + \|\boldsymbol{w}\|^2}$ provides a measure of the energy ratio between clean speech $\boldsymbol{s}$ and the noisy speech $\boldsymbol{s} + \boldsymbol{w}$, while $\|\quad\|$ is the norm operator, and $\langle\ ,\ \rangle$ is the inner product operator. The time-domain signals are used for computing loss function and assisting in error back-propagation only. In spite of the fact that the wSDR loss function is based on time-domain calculations, it can be back-propagated through the network. Specifically, the STFT and ISTFT operations can be implemented as consisting of fixed filters initialized with discrete Fourier transform matrix [39].

For the VAD module, the Cross Entropy (CE) loss is calculated after sequential processing by pipe-net, the attention module, and the post-net, as proposed in [30]:

$$\mathcal{L}_{\psi} = -\sum_{t=R}^{T-R-1} \sum\nolimits_{l\in\mathcal{T}} \left(y_{t+l}^{\text{truth}} \log\, y_{t+l} + \left(1 - y_{t+l}^{\text{truth}}\right)\log\left(1 - y_{t+l}\right)\right) \tag{3.13}$$

where y_{t+l}^{truth} and y_{t+l} are the $(t+l)^{th}$ component of the ground true label vector $\boldsymbol{y}_t^{\text{truth}} \in \mathbb{R}^L$ and soft prediction vector $\boldsymbol{y}_t \in \mathbb{R}^L$, respectively. Then the total loss for the proposed model is defined as:

$$\mathcal{L}_{total} = \lambda_1\, \mathcal{L}_{wSDR} + \lambda_2 \mathcal{L}_{pipe} + \lambda_3\, \mathcal{L}_{att} + \lambda_4 \mathcal{L}_{post} \tag{3.14}$$

where the subscripts $pipe$, att, and $post$ respectively stand for the pipe-net, temporal attention module and post-net. The parameters λ_1, λ_2, λ_3 and λ_4 are the weights given to the loss functions of the different modules. These loss functions are used in such a combination to train the network, i.e., by adjusting the network parameters to minimize the total loss. Additional implementation details of the network training, including batch size, optimizer, etc., are provided in Section 5.1.2.

3.6 Summary

In this Chapter, we have presented a novel attention model-based VAD method, called AM-cIRM, which takes advantages of cIRM and uses attention mechanisms of STAM to focus on more important information within the acoustic feature set. The AM-cIRM will be evaluated and compared with other methods in Chapter 5.

Chapter 4

Efficient Transformer with Feature Patches for VAD

In this chapter, we propose a novel transformer-based [41] VAD method that splits the acoustic features into patches and applies depth-wise convolutions, thereby allowing the model to predict the presence or absence of speech more efficiently. The proposed transformer-based VAD method, called Tr-VAD, consists of a feature embedding[2] layer, N_{trans} transformer encoder blocks, and a classifier as illustrated in Fig. 4.1. These components are described in more details in the following sections.

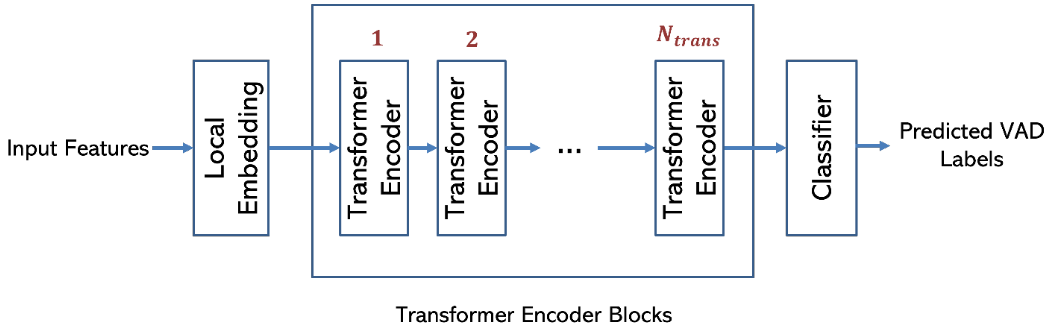

Fig. 4.1 Architecture of Tr-VAD.

[2] In many speech-based applications (e.g., speaker/language/emotion recognition tasks) and in the context of machine learning, the term "embedding" refers to fixed dimensional vector representation of an utterance, and is also called utterance-level representation. In order to avoid confusion, "Local Embedding" is used to refer to the first module, here "local" means frame level.

4.1 Feature Embedding

Similar to Section 2.3, the acoustic feature vectors and label pairs can be presented as $\{(\boldsymbol{X}_t, y_t^{\text{truth}})\}_{t=0}^{T-1}$, where $\boldsymbol{X}_t \in \mathbb{R}^D$ is the acoustic feature vector with feature dimension D and $y_t^{\text{truth}} \in \{0,1\}$ is the VAD label for the t^{th} frame. To exploit contextual information, the feature vector is then expanded by using $L' = 2k + 1$ neighboring frames indexed by $\mathcal{T}' = \{-k\mu', -(k-1)\mu', \dots, -\mu', 0, \mu', \dots, (k-1)\mu', k\mu'\}$, where the positive integers k and μ' control which neighboring frames will be used by the model. The extended data set is denoted as $\{(\boldsymbol{X}'_t, \mathbf{y}_t^{\text{truth}})\}_{t=k\mu'}^{T-k\mu'-1}$:

$$\boldsymbol{X}'_t = \{\boldsymbol{X}_{t+l'}^T : l' \in \mathcal{T}_t'\} \in \mathbb{R}^{L' \times D}, \qquad \boldsymbol{y}_t^{\text{truth}} = \{y_{t+l'}^{\text{truth}} : l' \in \mathcal{T}_t'\} \in \mathbb{R}^{L'} \tag{4.1}$$

The local embedding layer aims to project features from the original space into a new space which allows the model to extract more useful information and conduct effective learning. This layer consists of an FCN layer and a 1-D convolutional layer. Compared to the absolute positional embedding strategy (such as sinusoid positional embedding and learnable 1-D position embedding), the convolutional embedding is able to extract relative positional information and learns useful short-range spectral-temporal patterns [42].

4.2 Depth-Wise Transformer Block with Feature Patches

Recently, the vision transformer [43] and its variations [44]-[46] have achieved state-of-the-art performance on image classification, object detection, semantic segmentation, etc. Inspired by their effectiveness, we propose to apply transformer encoder-based network to the VAD task.

As illustrated in Fig. 4.1, the proposed Tr-VAD architecture includes a sequence of N_{trans} depth-wise transformer encoder blocks, where the internal structure of a transformer block is given in Fig. 4.2. The depth-wise transformer block includes two normalization layers, a Multi-Headed Self Attention (MHSA) module, and a Feed-Forward Network (FFN) configured as shown.

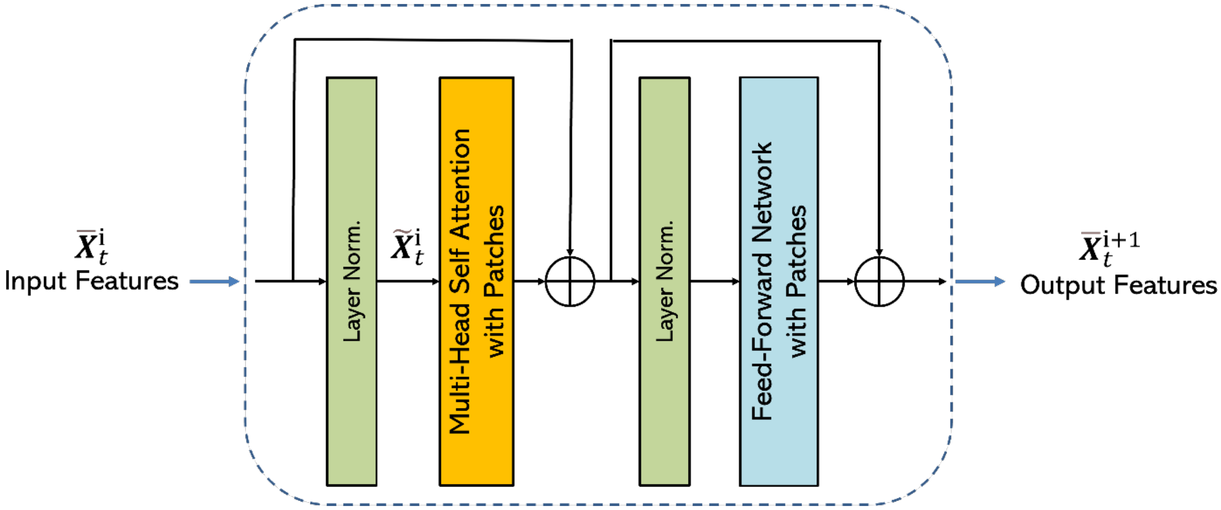
$\overline{\boldsymbol{X}}_t^{\mathrm{i}}$
Input Features
Layer Norm.
$\widetilde{\boldsymbol{X}}_t^{\mathrm{i}}$
Multi-Head Self Attention with Patches
Layer Norm.
Feed-Forward Network with Patches
$\overline{\boldsymbol{X}}_t^{\mathrm{i}+1}$
Output Features

Fig. 4.2 Block diagram of the i^{th} Depth-Wise Transformer Block

4.2.1 Multi-Head Self Attention with Feature Patches

Let the input features matrix of the i^{th} transformer block be denoted as $\overline{\boldsymbol{X}}_t^{\mathrm{i}} \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$, where $\tilde{L}$ and $\tilde{D}$ denote the temporal and feature dimensions, respectively. The feature matrix $\overline{\boldsymbol{X}}_t^{\mathrm{i}}$ is obtained from the previous module which is either the feature embedding module or the previous transformer block. As illustrated in Fig. 4.2, the layer normalization [47] is applied to the feature matrix $\overline{\boldsymbol{X}}_t$. Then the normalized feature matrix $\widetilde{\boldsymbol{X}}_t^{\mathrm{i}} \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$ is passed to the MHSA module whose structure is described below.

The internal structure of the MHSA module is illustrated in Fig. 4.3. The input $\widetilde{\boldsymbol{X}}_t^{\mathrm{i}}$ is first split into patches: the temporal dimension $\tilde{L}$ and the feature dimension $\tilde{D}$ are split into $P_1 \times P_2$ non-overlapping pieces:

$$\widetilde{\boldsymbol{X}}_{t,S}^{\mathrm{i}} = \mathrm{Split}\left(\widetilde{\boldsymbol{X}}_t^{\mathrm{i}}\right) \in \mathbb{R}^{D_{\mathrm{split}} \times P_1 \times P_2} \tag{4.2}$$

where integer P_1 and P_2 specify the split factors, $\mathrm{Split}(\cdot)$ stands for the split operation, and $D_{\mathrm{split}} = \frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2}$ is the total number of pieces.

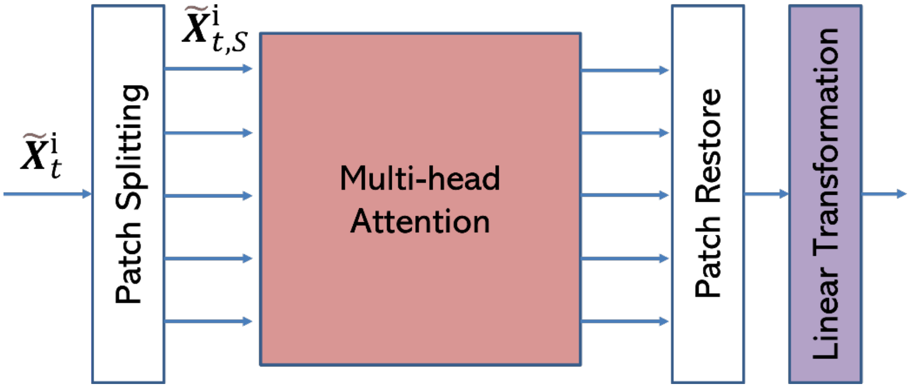
$\widetilde{\boldsymbol{X}}_t^{\mathrm{i}}$
Patch Splitting
$\widetilde{\boldsymbol{X}}_{t,S}^{\mathrm{i}}$
Multi-head Attention
Patch Restore
Linear Transformation

Fig. 4.3 Illustration of the MHSA module with feature patches.

In contrast to the *vanilla transformer* [41] which employs the MHSA along the feature dimension only, we employ it along both temporal and feature dimensions simultaneously. The new MHSA scheme allows the model to attend to multiple frames and features at different positions. More importantly, it allows the model to reduce the computation cost in the calculation of the attention matrix. Indeed, the Swin Transformer [45] uses shifted windows to introduce communication among different patches and to increase the receptive field. But in our case, the input acoustic features already include information from neighboring frames which includes contextual redundancy. Therefore, shifted widows are not necessary for our work.

Compared to the *vanilla transformer* architecture in [41] which uses FCN to map three feature matrices (or so called: query, key, and value) come from previous layers, depth-wise separable convolution blocks [48] are used in the proposed model instead. These blocks consist of a depth-wise convolutional layer, a batch normalization layer, and a scaling (or equivalently, a 1×1 point-wise convolutional) layer. The point-wise convolution is performed over multiple input channels while the depth-wise convolution applies a single convolutional filter to each input channel. Such depth-wise convolution can provide a more precise mechanism for local information aggregation, which is missing in the FFN of the *vanilla transformer* [46].

As illustrated in Fig. 4.4, we use a stride of 2 to map the feature matrix $\widetilde{\boldsymbol{X}}_{t,S}^{i}$. let $\mathrm{DW}(\cdot)$ be the Depth-Wise convolutional mapping operation, the mapped feature matrix $\mathrm{DW}\left(\widetilde{\boldsymbol{X}}_{t,S}^{i}\right) \in \mathbb{R}^{D_{\mathrm{split}} \times \frac{P_1}{2} \times \frac{P_2}{2}}$ is reshaped to $\widetilde{\boldsymbol{X}}_{t,DW}^{i} \in \mathbb{R}^{\frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2} \times \frac{P_1 P_2}{4}}$ as follows:

$$\mathbf{Q}_p = \mathrm{Reshape}\left(\mathrm{DW}\left(\widetilde{\boldsymbol{X}}_{t,S}^{i}\right)\right) \in \mathbb{R}^{\frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2} \times \frac{P_1 P_2}{4}} \tag{4.3}$$

$$\mathbf{K}_p = \mathrm{Reshape}\left(\mathrm{DW}\left(\widetilde{\boldsymbol{X}}_{t,S}^{i}\right)\right) \in \mathbb{R}^{\frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2} \times \frac{P_1 P_2}{4}} \tag{4.4}$$

$$\mathbf{V}_p = \mathrm{Reshape}\left(\mathrm{DW}\left(\widetilde{\boldsymbol{X}}_{t,S}^{i}\right)\right) \in \mathbb{R}^{\frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2} \times \frac{P_1 P_2}{4}} \tag{4.5}$$

where $\mathrm{Reshape}(\cdot)$ is the reshape operation, the $\mathbf{Q}_p$, $\mathbf{K}_p$, and $\mathbf{V}_p$ are obtained by applying depth-wise convolutions with different weights to the split feature matrix $\widetilde{\boldsymbol{X}}_{t,DW}^{i}$. Then the scaled dot-product attention operation is applied as follows:

$$\widetilde{\boldsymbol{X}}_{t,att}^{i} = \mathrm{Softmax}\left(\frac{\mathbf{Q}_p^T \mathbf{K}_p}{\sqrt{N_p}} + \boldsymbol{B}_p\right) \cdot \mathbf{V}_p \tag{4.6}$$

where $\cdot$ is the element-wise product, $\boldsymbol{B}_p \in \mathbb{R}^{\frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2} \times \frac{\tilde{P}}{P_2}}$ is a bias term that can be learned during the training, and $N_p = \frac{P_1 P_2}{4}$ is the scaling factor.

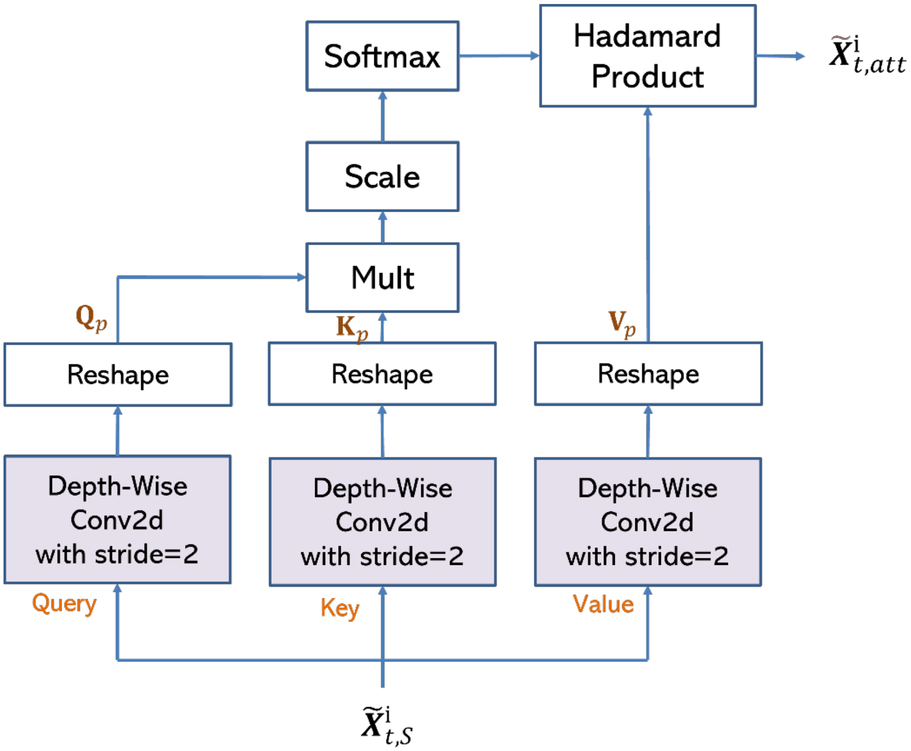
Softmax
Hadamard Product
$\widetilde{\boldsymbol{X}}^{\mathrm{i}}_{t,att}$
Scale
Mult
$\mathbf{Q}_p$
$\mathbf{K}_p$
$\mathbf{V}_p$
Reshape
Reshape
Reshape
Depth-Wise Conv2d with stride=2
Depth-Wise Conv2d with stride=2
Depth-Wise Conv2d with stride=2
Query
Key
Value
$\widetilde{\boldsymbol{X}}^{\mathrm{i}}_{t,S}$

Fig. 4.4 Illustration of the attention function in Tr-VAD.

Indeed, depth-wise convolution is efficient in terms of both the number of parameters and computational complexity. Since we use a stride of 2 to map the query, key, and value, the temporal and feature dimensions are both reduced by a factor of 2, and the computational cost for the scaled dot-product attention operation in Eq. (4.6) is thus reduced by 4^3 times. Such strategy comes with negligible performance degradation as the input features contain redundant information.

Going back to Fig. 4.3, the attention output $\widetilde{\boldsymbol{X}}^{\mathrm{i}}_{t,att} \in \mathbb{R}^{\frac{\tilde{L}}{P_1}\times\frac{\tilde{D}}{P_2}\times\frac{P_1P_2}{4}}$ is then reshaped to $\overline{\boldsymbol{X}}_{t,att} \in \mathbb{R}^{\frac{L}{2}\times\frac{D}{2}}$. Finally, linear transformations including a 1×1 1-D convolutional layer and an FCN are applied so that the output of the MHSA module shares the same shape as the input $\widetilde{\boldsymbol{X}}^{\mathrm{i}}_{t} \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$.

4.2.2 Feed-Forward Network with Feature Patches

In this work, as illustrated in Fig. 4.5, we propose to use the convolution-based FFN [49] instead of the FCN-based counterparts. Similar to the MHSA module, the proposed FFN also splits the input features into $P_1 \times P_2$ patches. The FFN includes two 1×1 point-wise convolutions which are applied to expand and squeeze the hidden dimension by γ_{FFN} times. One 3×3 depth-wise separable convolution block is used to introduce local dependencies. Feature patches are restored so that the output of FFN shares the same shape as the input of the i^{th} depth-wise transformer block $\overline{\boldsymbol{X}}_t^{\mathrm{i}} \in \mathbb{R}^{\tilde{L} \times \widetilde{D}}$.

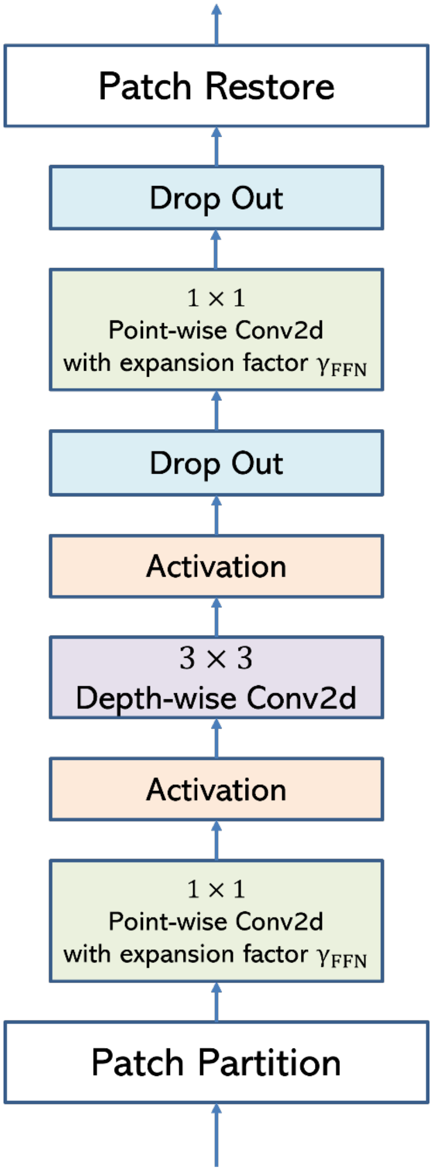

Fig. 4.5 Architecture of the Feed-Forward Network with patches.

4.3 Classifier

Recall from Fig. 4.1 that the output of the transformer encoder blocks is finally fed to the classier. As illustrated in Fig. 4.6, the classifier splits the features into patches, then a 5×5 depth-wise convolution block with stride 2 is applied to the feature patches. Let the output feature matrix of the convolution block be denoted as $\boldsymbol{X}_{t,c} \in \mathbb{R}^{D_{\text{split}} \times \frac{P_1}{2} \times \frac{P_2}{2}}$ which is then reshaped into $\widetilde{\boldsymbol{X}}_{t,c} \in \mathbb{R}^{\frac{P_1}{2} \times \frac{P_2 D_{\text{split}}}{2}}$. The first FCN in the classifier expands the last dimension by γ_c times while the second one compresses this dimension to 1. The output of the second FCN passes through a sigmoid activation to predict the probability of the presence of speech $\boldsymbol{y}_t \in \mathbb{R}^{\frac{P_1}{2}}$. In our work, $P_1 = 2L'$, thus the probability can also be denoted as $\boldsymbol{y}_t = \{y_{t+l'}: l' \in \mathcal{T}_t'\} \in \mathbb{R}^{L'}$. The soft prediction corresponding to the t^{th} frame $\hat{y}_t$ can be computed by aggregating all the soft predictions $\boldsymbol{y}_t$ relative to the current frame t across l', and the hard prediction can be obtained by thresholding the soft prediction with θ_{T} as discussed in Section 2.3 and Section 3.4.

For training, the cross entropy loss is calculated after the classifier:

$$\mathcal{L}_{\mathrm{Tr}} = -\sum_{t=k\mu'}^{T-k\mu'-1} \sum\nolimits_{l' \in \mathcal{T}_t'} \left(y_{t+l'}^{\text{truth}} \log y_{t+l'} + \left(1 - y_{t+l'}^{\text{truth}}\right) \log(1 - y_{t+l'})\right) \tag{4.7}$$

where $y_{t+l'}^{\text{truth}}$ and $y_{t+l'}$ are the $(t + l')^{th}$ component of the ground true label vector $\boldsymbol{y}_t^{\text{truth}}$ and soft prediction vector $\boldsymbol{y}_t$, respectively. More details about the training process, including batch size, optimizer, learning rate, etc., can be found in Section 5.1.2.

4.4 Summary

In this Chapter, we have presented a novel transformer-based VAD method, Tr-VAD, which applies depth-wise convolutions on feature patches, thereby allowing the model to predict the presence of speech/non-speech more efficiently. The Tr-VAD will be evaluated and compared with other methods in Chapter 5.

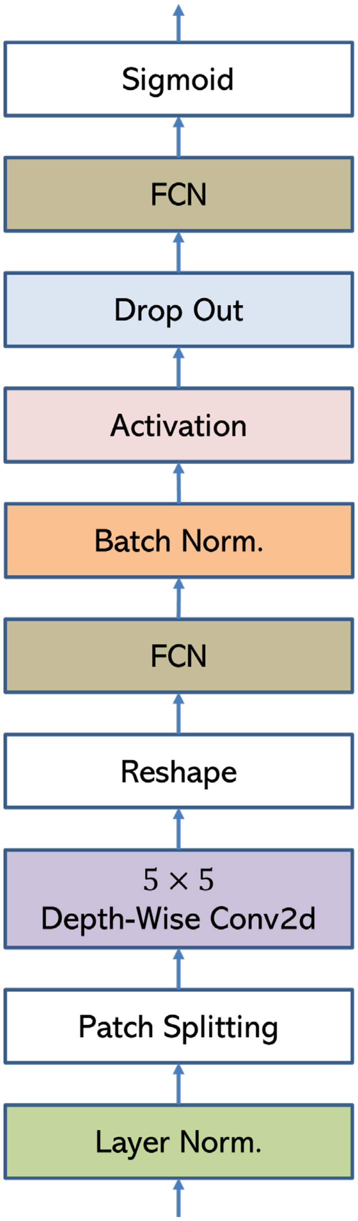

Fig. 4.6 Architecture of the Classifier.

Chapter 5

Experiments and Results

In this chapter, we firstly describe the methodology used to evaluate the performance of the proposed methods. This includes the discussion of the dataset, parameter setting, baseline methods used for comparison and evaluation metrics. We subsequently compare the performance of the proposed AM-cIRM and Tr-VAD and baseline methods by presenting the experimental results.

5.1 Methodology

In this section, we describe the dataset used to carry out our experiments, the parameter setting of the proposed methods, the baseline methods used for comparison, and evaluation metrics used for comparative evaluation.

5.1.1 Dataset

The clean utterances used to construct the training dataset were selected from the TIMIT corpus [50] which contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus provides speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. It also includes time-aligned orthographic, phonetic, and word transcriptions as well as a 16-bit, 16 kHz speech waveform file for each utterance.

Since the TIMIT utterances have considerably shorter silences than speech, the speech/non-speech class imbalance problem may occur. To mitigate the problem, we added 1-second-long silent segments before and after each utterance.

NOISEX-92 dataset [51] which includes 13 different types of noises sampled at 16 kHz are used to corrupt the clean TIMIT training dataset. In our experiments only 8 of them are used: babble, F16, destroyer operation room, M109, Volvo, white, and two types of factory noises. SNR values are set at: -10, -5, 0, 5, 10 dB. This augmentation resulted in 189,420 training segments and added up about 267 hours of audio stream in total. The proposed models and baseline models were trained with 95% of the training data and the remaining 5% was left as the validation set.

In the testing phase, the TIMIT test dataset and the subset ‘test_clean’ from LibriSpeech corpus [52] were used to evaluate the performance of different methods. The LibriSpeech dataset is derived from audiobooks and contains 1000 hours of speech sampled at 16 KHz. But unlike the TIMIT dataset which has ground truth labels for the VAD task, the LibriSpeech does not have ground truth labels, thus rVAD [3] was applied to generate pseudo ground truth labels.

Similar to the training phase, all 8 types of unseen noises from the AURORA noise dataset [53] including babble, airport, car, exhibition, restaurant, street, subway and train noises, were used to augment and corrupt the clean testing utterances. The SNRs are set to -5, 0, 5, and 10 dB. For the TIMIT test dataset, each utterance in the dataset was padded with 0.5-second, 1-second, and 1.5-second-long ‘silence’ (denoted as TIMIT-0.5, TIMIT-1, and TIMIT-1.5) before and after each utterance.

5.1.2 Parameter Setting

For the two proposed methods, detailed parameter settings and training strategies are discussed as follows:

A. Parameter Setting for AM-cIRM

The input complex-valued spectrograms are obtained by applying STFT with $N = 1024$. The cIRM Feature Extractor follows the same parameter setting as DCUnet-10 in [39] as indicated in the Table 5.1:

Table 5.1 Parameter Setting for cIRM Extractor

	E1(1)	E2	E3	E4	E5	D1(2)	D2	D3	D4	D5
# Output Channel	45	90	90	90	90	90	90	90	45	1
Filter Size	(7×5)	(7×5)	(5×3)	(5×3)	(5×3)	(5×3)	(5×3)	(5×3)	(7×5)	(7×5)
Stride Size	(2,2)	(2,)	(2,2)	(2,2)	(2,1)	(2,1)	(2,2)	(2,2)	(2,2)	(2,2)

(1), (2): E and D stand for the encoder and decoder, respectively, E1 refers to the first encoder.

The Linear Transformation uses a 1-D convolutional layer with kernel size 2, stride 1, and output channels $D = 80$. When the network only uses the magnitude information instead of using both phase and magnitude information, the 1-D convolution is replaced with a fully connected layer with hidden units 80. Log-Mel filter banks with the same feature dimension D is concatenated with the transformed cIRM. R, μ, and L are set to 19, 9 and 7 respectively to form the expanded feature vector.

The voice activity detector uses a similar parameter setting as STAM in [30], except that the number of input channels of the first convolution pair in spectral attention is doubled, while the number of output channels N_{c} remains the same. In the spectral attention module, N_{spec}, N_{c} and k_{spec} are set to 4, 16 and 3 respectively. The output units of the two FCNs in the pipe net are set to $N_{pipe} = 256$. N_d from the temporal attention module is set to 128 and the number of heads in the multi-headed attention operation is 8. The hidden units of first FCN in the post net are 256 while the unit for the second FCN is 1.

During the training, the mini batch with batch size of 550 is applied, hence $T = 550$. The model is optimized with Adam optimizer. The learning rate starts from 10^{-3} for the first 50,000 iterations, and exponentially decays every 25,000 iterations with decay rate 0.8. The final learning rate is 10^{-5}. Parameters λ_1, λ_2, λ_3, λ_4 and θ_{cIRM} are set to 0.5, 1, 0.2, 1 and 0.5, respectively, these parameters are near optimal and were obtained by trial and error. The dropout rate is set to 0.5.

B. Parameter Setting for Tr-VAD

Each utterance from the training and test dataset is framed by applying a 32 ms Hann window with 16 ms window shifts. For the feature construction, the complex-valued spectrograms are obtained by applying STFT with $N = 512$, the Tr-VAD uses AFPC features discussed in Section 2.1. Specifically, they contain 16 coefficients from MFCC, $\Delta\mathrm{MFCC}$, $\Delta^2\mathrm{MFCC}$, NSSC, and $\Delta\mathrm{NSSC}$, respectively, resulting in 80-D AFPC features. For training, mini batch with a batch size of 512 is applied. The Tr-VAD is optimized with AdamW optimizer [54] using a cosine decay learning rate scheduler and 5000 iterations of linear warm-up. An initial learning rate 10^{-3}, a weight decay 0.05 and a final learning rate 5×10^{-6} after 4×10^5 iterations are used. Activation function Gaussian error linear units (GELU) [55] is chosen. For feature construction, k, μ', and L' are set to 4, 2, and 9, respectively. Model parameters D, $\tilde{L}$, $\widetilde{D}$, P_1, P_2, D_{split}, θ_{T}, and N_{trans} are set to 80, 54, 162, 18, 18, 27, 0.5, and 6, respectively. In the proposed transformer block, the Tr-VAD uses $P_1 = P_2 = 9$

heads for the MHSA calculation. The dropout rate is set to 0.1. Other model parameter settings can be found in Table 5.2.

Table 5.2 Parameter Setting for Tr-VAD

Layer Name	# Channels In (Units In)	# Channels Out (Units Out)	Kernel Size	Stride Size
FCN in Embedding	80	324		
1-D Conv.[(1)] in Embedding	9	54	5	2
DW[(2)] in MHSA	27	27	(3, 3)	(2, 2)
1-D Conv. in MHSA	27	54	1	1
FCN in MHSA	81	162		
1st 2-D Conv. in FFN	27	108	(1, 1)	(1, 1)
DW in FFN	108	27	(3, 3)	(1, 1)
2nd 2-D Conv. in FFN	108	27	(1, 1)	(1, 1)
DW in Classifier	27	27	(5, 5)	(2, 2)
1st FCN in Classifier	243	486		
2nd FCN in Classifier	486	1		

[(1)] Conv. stands for convolutional layer; [(2)] DW stands for the depth-wise convolution block.

5.1.3 Baseline Methods

For performance evaluation, the proposed methods, AM-cIRM and Tr-VAD from Chapter 3 and 4, are compared with the following baseline approaches:

- rVAD [3]: An unsupervised VAD method which exploits pitch information by calculating the $a\ posteriori$ SNR weighted energy difference.
- ACAM [29]: First attention-based VAD model which only applies temporal attention.
- STAM [30]: An attention-based VAD model which exploits both spectral and temporal information.
- DCU-10 [39]: A DNN-based SE model comprised of 10 complex layers which is still capable of predicting VAD labels. To this end, the cIRM $\widehat{\mathbf{M}}_{STFT}$ is averaged along the frequency axis and a VAD decision is made by comparing the magnitude of the averaged cIRM with a threshold θ_{DCU}:

$$\widehat{M}(t) = \frac{1}{F}\sum_{f=0}^{F-1} \widehat{\mathrm{M}}_{t,f} \tag{5.1}$$

$$\mathrm{VAD_{DCU}(t)} = \begin{cases} 1 & \text{if } |\widehat{M}(t)| > \theta_{\mathrm{DCU}} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

  where $\widehat{\mathrm{M}}_{t,f}$ is the f^{th} frequency component of the t^{th} frame of $\widehat{\mathbf{M}}_{STFT}$, $|\widehat{M}(t)|$ is the magnitude of $\widehat{M}(t)$, and θ_{DCU} is a threshold set to 0.2 by conducting grid search on validation set to optimize VAD performance.
- AM-Mag: AM-Mag stands for the cIRM-based method which only uses the magnitude information from the cIRM.

All supervised methods including DCU-10, ACAM, and STAM models were trained using the same training methods as proposed by the authors in the original papers, and default parameters settings were also applied to rVAD.

5.1.4 Evaluation Metrics

For comparison, the F1-score and Detection Cost Function (DCF) [3] are selected as the main evaluation metrics for VAD.

These metrics are explained in more detail below.

The F1-score takes both accuracy and recall metrics into account, and is commonly used as evaluation index of binary classification problems [3]:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{5.3}$$

where TP, FP, FN represent the number of true positive, false positive, and false negative cases, respectively. Higher values of the F1-score metrics suggest better performance.

The DCF reflects the wrong performance of the model, and it is defined as follows:

$$\mathrm{DCF} = (1 - \beta)\mathrm{P}_{\mathrm{FN}} + \beta\mathrm{P}_{\mathrm{FP}} \tag{5.4}$$

where the β is a weighting factor, P_{FP} is the rate of FP (also called probability of false alarm) while P_{FN} is the rate of FN (also called probability of missed detection). In practice, β is set to 0.25, which penalizes missed speech frames more heavily. Lower values of the DCF metrics suggest better performance.

5.2 Results and Discussion

In this section, the performance of two proposed methods is compared with the baseline methods under different conditions. We first investigate the performance of AM-cIRM against that of the baseline methods, then we compare the performance of the Tr-VAD with AM-cIRM.

5.2.1 Experimental Results for AM-cIRM

The comparison results of F1 score (in percent) for the AM-cIRM and baseline methods are presented in Table 5.3. The results are averaged over different SNRs and noise types. The numbers of parameters of different methods are presented in the second line of the first row. In order to compare the robustness of different methods, the noise corrupted TIMIT test datasets with 0.5-second, 1-second, and 1.5-second-long silence padding and the noise corrupted LibriSpeech test set are used. It is clear that all attention-based methods (including ACAM, STAM, and the proposed method) achieve better results than non-attention-based methods (including rVAD and DCU-10). For the TIMIT testing dataset, in comparison with ACAM, STAM greatly improves the performance by introducing convolutional blocks and multi-headed attention module. AM-Mag improves the results by 0.3% on F1 score with the use of magnitude information from cIRM. With phase information, AM-cIRM further increases the F1 score by more than 0.1%. For the Librispeech testing dataset, compared with STAM, the proposed method improves the performance by 1.7%.

Table 5.3 Comparison of F1 (in precent) Score for AM-cIRM and Baseline Methods

Model Name (#parameters)	rVAD (NA)	DCU-10 (2808K)	ACAM (957K)	STAM (559K)	AM-Mag (3572K)	AM-cIRM (3613K)
TIMIT-0.5	90.37	91.52	92.56	98.06	98.43	**98.54**
TIMIT-1	87.35	90.78	91.21	98.15	98.48	**98.64**
TIMIT-1.5	83.51	88.27	89.44	98.22	98.54	**98.72**
LibriSpeech	NA	82.52	87.51	88.39	90.07	**90.17**

Table 5.4 shows the comparison results of DCF which is averaged over different types of noise and SNR levels. The AM-Mag improves the results by 0.6% on DCF while the AM-cIRM further

improves the performance by more than 0.03%. For the Librispeech testing dataset, compared with STAM, the proposed method decreases the DCF by about 3%.

Table 5.4 Comparison of DCF (in precent) for AM-cIRM and Baseline Methods

Model Name (#parameters)	rVAD (NA)	DCU-10 (2808K)	ACAM (957K)	STAM (559K)	AM-Mag (3572K)	AM-cIRM (3613K)
TIMIT-0.5	5.72	6.14	4.60	1.78	0.94	**0.91**
TIMIT-1	5.38	5.12	3.72	1.32	0.72	**0.67**
TIMIT-1.5	5.75	5.22	3.38	1.05	0.58	**0.52**
LibriSpeech	NA	15.29	11.74	13.42	10.55	**10.32**

Table 5.5 Comparison of F1 Score (in precent) for Different SNRs on TIMIT-1 for AM-cIRM

Noise Level	rVAD	DCU-10	ACAM	STAM	AM-Mag	AM-cIRM
-5 dB	79.51	86.48	85.91	97.78	97.79	**98.01**
0 dB	86.03	89.85	90.70	98.08	98.36	**98.54**
5 dB	92.44	92.36	95.45	98.30	98.80	**98.91**
10 dB	93.98	94.19	96.03	98.46	99.02	**99.10**

Table 5.5 and Table 5.6 show the detailed results of test TIMIT-1 with different SNR levels ranging from -5 dB to 10 dB. For Table 5.5, it is interesting to note that DCUnet and ACAM achieve similar performance at low SNRs (-5 dB and 0 dB), which verifies our assumption that the cIRM contain useful information for the detection of the presence/absence of speech. AM-cIRM and STAM achieve similar F1 score at low SNRs while AM-Mag provides more accurate predictions at higher SNRs. For Table 5.6, AM-Mag also provides better performance than baseline methods. With the use of phase information, AM-cIRM further slightly improves the robustness of the method.

Table 5.6 Comparison of DCF (in precent) for Different SNRs on TIMIT-1 for AM-cIRM

Noise Level	rVAD	DCU-10	ACAM	STAM	AM-Mag	AM-cIRM
-5 dB	8.31	7.83	6.24	1.50	1.07	**1.03**
0 dB	5.83	5.69	3.70	1.36	0.75	**0.70**
5 dB	3.90	4.07	2.32	1.25	0.57	**0.51**
10 dB	3.47	2.87	2.64	1.18	0.49	**0.43**

The influence of neighboring frames on the performance of AM-cIRM is also studied. As shown in Table 5.7, the relative index of the farthest neighboring frame $R = 19$ may be too large for real-time applications, as we need the R frames from the past and future signal streams, which may result in high latency in some scenarios. By reducing the values of R and μ and keeping the total number of frames L the same, the proposed method is likely to be implemented in real-time applications with a slight performance cost.

Table 5.7 The Influence of Neighboring Frames on the performance of AM-cIRM for the TIMIT-1 case (in percent)

Evaluation Metrics	$R = 19, \mu = 9$	$R = 13, \mu = 6$	$R = 9, \mu = 4$	$R = 7, \mu = 3$
F1 Score	**98.64**	98.52	98.34	98.29
DCF	0.67	**0.61**	0.66	0.69

5.2.2 Further Comparison of AM-cIRM and Tr-VAD

Table 5.8 shows the comparison results of the two proposed methods on TIMIT test dataset. We can conclude that the two methods achieve similar performance. Specifically, the AM-cIRM model outperforms the Tr-VAD on TIMIT-0.5, while the latter one achieves higher F1 score on TIMIT-1. Considering the number of parameters used by the two methods, the Tr-VAD is more efficient in tackling VAD problems.

Table 5.8 Averaged F1 Score and DCF (in percent) for the Proposed Methods on TIMIT Dataset

Methods	# Parameters	TIMIT-0.5		TIMIT-1		TIMIT-1.5	
		F1	DCF	F1	DCF	F1	DCF
AM-cIRM	3613K	**98.54**	**0.91**	98.64	**0.67**	98.72	0.52
Tr-VAD	**376K**	98.22	1.81	**98.91**	0.69	**98.75**	**0.49**

Table 5.9 shows the comparison of F1 score and DCF (in percent) for different SNRs on the TIMIT-1 dataset. The Tr-VAD demonstrates its robustness under low SNR conditions with 0.6% and 0.2% improvement on the F1 score and DCF at -5 dB, respectively. The two proposed methods achieve similar performance at higher SNRs.

Table 5.9 Comparison of F1 Score and DCF (in precent) for Different SNRs on the TIMIT-1 for AM-cIRM and Tr-VAD

Methods	-5 dB		0 dB		5 dB		10 dB	
	F1	DCF	F1	F1	DCF	DCF	F1	DCF
AM-cIRM	98.01	1.03	98.54	**0.70**	98.91	**0.51**	99.10	**0.43**
Tr-VAD	**98.63**	**0.84**	**98.87**	0.71	**99.03**	0.63	**99.13**	0.58

Fig. 5.1 shows the comparison of the hard VAD decisions produced by baseline methods and the proposed methods. The clean signal sample is chosen from the ‘test_clean’ dataset of LibriSpeech corpus and is indexed by ‘*61-70968-000*’. The transcript of the 4.9-second-long clean signal is: “He began a confused complaint against the wizard who had vanished behind the curtain on the left”. The top sub-figure shows the waveform of the clean signal, the remaining sub-figures show the hard VAD decisions obtained by applying different methods to a noisy signal which is obtained by adding the ‘airport’ noise from the AURORA noise corpus to the clean signal, the SNR of the noisy signal is 0 dB. DCU-10 and rVAD use different thresholding strategies as

discussed in Section 5.1.3 and Section 2.2, respectively; the remaining methods uniformly use a threshold of 0.5 for hard thresholding. From the figure we can tell that, the AM-cIRM correctly predicted the starting and the ending of the speech despite that it wrongly predicted the middle non-speech part. The Tr-VAD precisely predicted the starting and the ending of the speech as well as the middle non-speech parts.

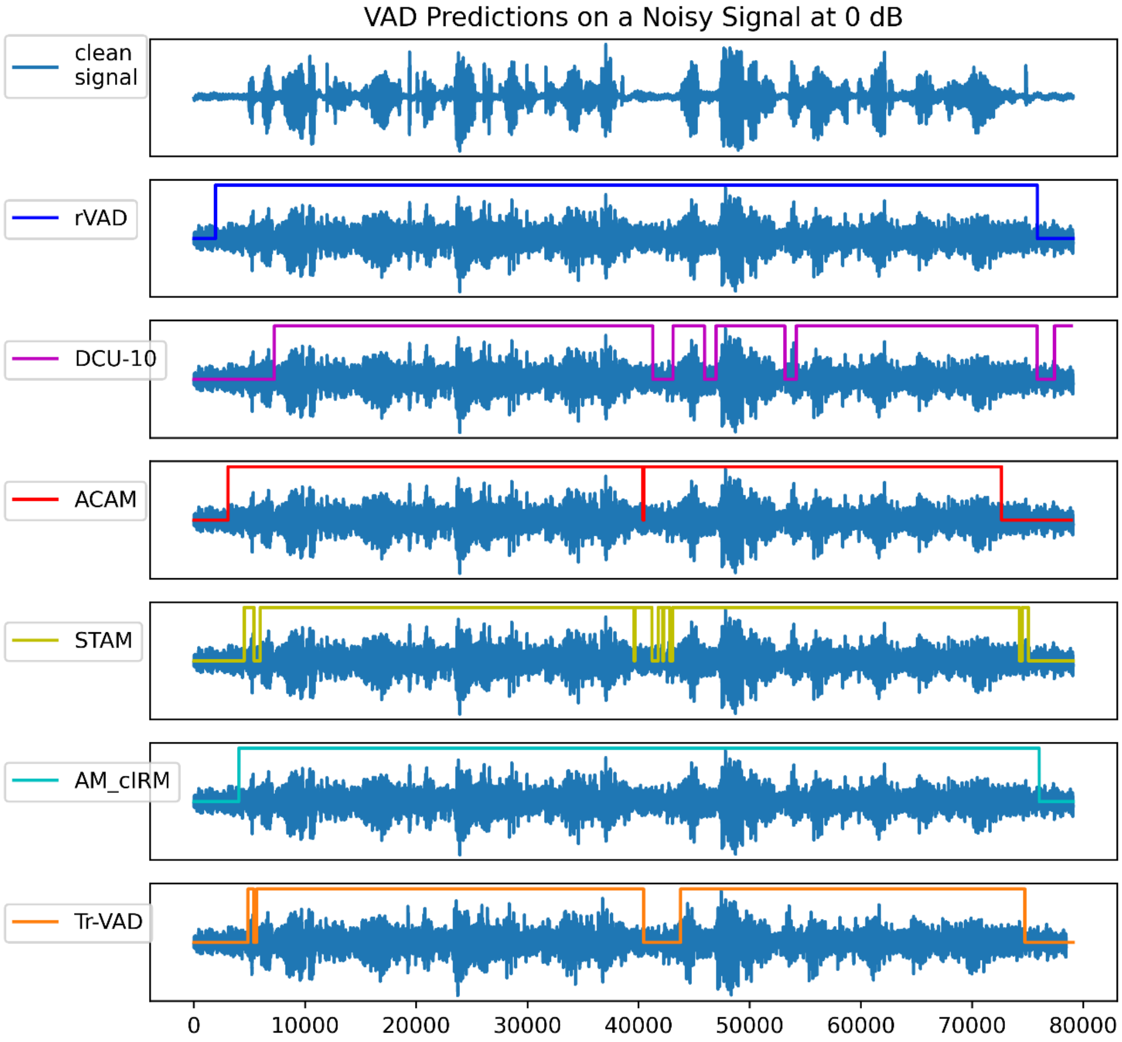

Fig. 5.1 Comparison of the hard VAD decisions produced by different methods.

Table shows the running time required by different VAD methods to process 1680 utterances (about 2.38 hours) from TIMIT-1 dataset. The experiments were conducted on a platform equipped with Intel Core i7-10700F CPU, NVIDIA GeForce RTX 2070 SUPER GPU. Please note that the rVAD only used CPU while the rest of them used both CPU and GPU. Again, the Tr-VAD demonstrates its efficiency with significantly less execution time compared to other DNN-based methods.

Table 5. 10 Running time required for inference

Running Time	rVAD	DCU-10	ACAM	STAM	AM-cIRM	Tr-VAD
Total time (in seconds)	86	251	1263	132	269	**82**
Averaged time[(2)] (in milliseconds)	100.4	293.0	1474.1	154.1	314.0	**95.7**

[(1)] ACAM needs 1263 seconds for MRCG feature extraction and seconds for decision making.

[(2)] Averaged time required for inferencing a 10-second-long utterance.

To validate the effectiveness of each part of the Tr-VAD network, we further conduct ablation studies on it. As shown in Table 5.11, the baseline Tr-VAD uses the AFPC features discussed in the Section 2.1, and has shown about 0.4% increase in F1 score compared to the one using log-Mel filter bank coefficients, while the use of MFCC contributes no better performance than the use of filter bank coefficients. The use of MultiLayer Perceptron (MLP)-based FFN results in similar F1 score and DCF to the baseline method with the cost of 3 times more parameters used. Similarly, the architecture with the original Multi-Headed Attention (MHA) operation used in the *vanilla transformer* [40] also requires more parameters to be trained but does not contribute to obvious performance improvement.

Table 5.11 Ablation Study on Tr-VAD on TIMIT-1 (in percent)

Evaluation Metrics	Baseline Tr-VAD [*]	Log-Mel Filter Bank	MFCC	MLP-based FFN	Original MHA[**]
# Parameters	**376K**	**376K**	**376K**	1527K	901K
F1 Score	**98.91**	98.53	98.52	98.88	98.87
DCF	0.69	0.76	0.91	0.71	**0.65**

[*] The baseline Tr-VAD uses AFPC features, depth-wise convolution-based FFN, and depth-wise convolution-based MHSA.

[**] The original MHA uses 9 heads in this experiment.

The influence of neighboring frames on the performance of Tr-VAD is also studied. With step size $\mu' = 4$, $k = 4$, $\mu' \times k = 16$ frames from the past and future signal streams are needed. As shown in Table 5.12, by reducing the step size u and keeping the total number of frames L' and k the same, the proposed method is likely to be implemented in real-time applications with a slight performance cost.

Table 5.12 The Influence of Neighboring Frames on Tr-VAD on TIMIT-1 (in percent)

Evaluation Metrics	$k = 4, \mu' = 4$	$k = 4, \mu' = 3$	$k = 4, \mu' = 2$
F1 Score	**98.91**	98.78	98.57
DCF	**0.69**	0.73	0.83

5.3 Summary

In this Chapter, we described the methodology used to evaluate the performance of the proposed AM-cIRM and Tr-VAD networks, and subsequently compared their performance to that of baseline methods by presenting experimental results. The results showed that both proposed methods achieve improved VAD performance compared to baseline methods from the literature in low to medium SNR environments. However, Tr-VAD is more efficient than AM-cIRM as it requires fewer network parameters to achieve a similar performance. The results also indicate that the use of AFPC features with Tr-VAD can guarantee better performance.

Chapter 6

Conclusion and Future Work

This chapter provides some concluding remarks about the research presented in this thesis. Specifically, Section 6.1 presents a brief summary of the thesis work, while Section 6.2 lists suggestions for possible future work in this area.

6.1 Thesis Overview and Contributions

In this thesis, we proposed a novel voice activity detection method that uses a complex Ideal Ratio Mask extractor for auxiliary feature extraction, and a voice activity detector to aggregate features and estimate the presence/absence of speech. Below, we provide a chapter-wise sequential overview of the main topics discussed in this work:

- In Chapter 1, a concise summary of the voice activity detection problem was presented. This was followed by a comprehensive literature survey on the conventional and deep neural network-based methods of voice activity detection, such as DNN-based VAD models using auxiliary features and attention mechanism-based methods.
- In Chapter 2, background theories including the acoustic feature extraction process, one non-DNN-based method, and one state-of-the-art attention-based model STAM were given. The two methods served as benchmarks and were compared with the proposed methods.

- In Chapter 3, the proposed attention-based complex IRM-aware DNN method for VAD, AM-cIRM, was introduced. This framework is composed of a cIRM feature extractor, a feature transformation module, and a VAD module. The AM-cIRM method takes advantages of cIRM and uses attention mechanisms of STAM to focus on more important information within the acoustic feature set.
- In Chapter 4, a novel transformer-based DNN method for VAD, Tr-VAD, was proposed. The new architecture splits the input features into patches and applies efficient depth-wise convolution operations on them, this strategy significantly reduces computation complexity.
- In Chapter 5, the proposed methods were compared with baseline methods and were evaluated based on F1 score and detect cost function. The results showed that the two proposed methods achieved state-of-the-art performance. Chosen as cIRM extractor, the DCUnet demonstrates decent VAD performance compared to classical method. Despite that STAM is able to achieve excellent performance, the adding of cIRM extractor allows the model more robust under low SNRs conditions. Tr-VAD, on the other hand, demonstrates its efficiency by providing similar performance with AM-cIRM with about 9 times fewer parameters. The experiments also show that the depth-wise convolution-based feed-forward network and multi-headed self-attention module save considerable computation cost and a significant amount of parameters. The use of a combination of audio fingerprinting features with Tr-VAD also can guarantee better performance.

6.2 Future Work

In this section, we point out some possible directions for future research work.

- The use of DUCnet-based cIRM extractor allows the network to learn auxiliary features. However, since a large number of parameters are required for the cIRM feature extractor, it is desirable to explore other DNN-based methods using a low-complexity approach. It would also be interesting to explore the influence of other acoustic features on the performance of the voice activity detection system.
- The proposed transformer-based method, Tr-VAD, is non-hierarchical, using the same architecture for all transformer encoder blocks. It would be interesting to construct a hierarchical structure by starting from small-sized patches and gradually merging neighboring patches in deeper transformer layers. Indeed, many attempts have been made and they demonstrated improvement using the hierarchical transformer-based architecture on computer vision [44] and natural language processing tasks [56].
- The experiments are limited to additive noise, however, in practice strong reverberation may negatively impact the VAD decisions. Thus, it would be interesting to study the effect of reverberation on the proposed schemes and attempt to improve the robustness of the systems.
- VAD is an integral and important part of many speech-based applications such as speech/speaker recognition, speech enhancement and so on. Improved stand-alone performance of VAD algorithms does not always guarantees to achieve improved performance of a speech-based application where the VAD algorithm is used. Therefore, it would be interesting to see how the proposed approaches behave when integrated into other speech-related applications.

References

[1]. J. Sohn, N. S. Kim and W. Sung, “A statistical model-based voice activity detection,” *IEEE Signal Process. Lett.,* vol. 6, no. 1, pp. 1-3, Jan. 1999.

[2]. J. C. Junqua, B. Reaves and B. Mak, “A study of endpoint detection algorithms in adverse conditions: Incidence on a DTW and HMM recognize,” in *EUROSPEECH*, pp. 1371–1374, Genova, Italy, 1991.

[3]. Z. H. Tan, A. Sarkar and N. Dehak, “rVAD: An Unsupervised Segment-Based Robust Voice Activity Detection Method,” *Computer Speech and Language*, vol. 59, pp. 1-21, Jan. 2020.

[4]. J. A. Haigh and J. S. Mason, “Robust voice activity detection using cepstral features,” in *Proc. of TENCON ’93. IEEE Region 10 International Conference on Computers, Communications and Automation*, vol. 3 pp. 321–324, Beijing, China, 1993.

[5]. J. Shen, J. Hung and L. Lee, “Robust entropy-based endpoint detection for speech recognition in noisy environments,” in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pp. 232–235, 1998.

[6]. R. Tucker, “Voice activity detection using a periodicity measure,” *IEE Proceedings I (Communications, Speech and Vision)*, vol. 139, no. 4, pp. 377–380, Aug. 1992.

[7]. E. Nemer, R. Goubran and S. Mahmoud, “Robust voice activity detection using higher-order statistics in the LPC residual domain,” *IEEE Trans., Speech, Audio Process*., vol. 9, no. 3, pp. 217–231, Mar. 2001.

[8]. Y. Tachioka, T. Narita, T. Hanazawa and J. Ishii, “Voice activity detection based on density ratio estimation and system combination,” in *Proceedings of APSIPA*, pp. 1–4, Kaohsiung, Taiwan, China, 2013.

[9]. J. Sohn, N. S. Kim and W. Sung, “A statistical model-based voice activity detection,” *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, Jan. 1999.

[10]. D. Ying, Y. Yan, J. Dang and F. K. Soong, “Voice Activity Detection Based on an Unsupervised Learning Framework,” in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 8, pp. 2624-2633, Nov. 2011.

[11]. J. H. Chang and N. S. Kim, “Voice activity detection based on complex Laplacian model,” *Electron. Lett*., vol. 39, no. 7, pp. 632–634, Apr. 2003.

[12]. J. Shin, J. H. Chang and N. S. Kim, “Statistical modeling of speech signals based on generalized gamma distribution,” *IEEE Signal Process. Lett*., vol. 12, no. 3, pp. 258–261, Feb. 2005.

[13]. J. Padrell, D. Macho and C. Nadeu, “Robust speech activity detection using LDA applied to FF parameters,” in *Proc. ICASSP*, vol. 1, pp. I–557, Philadelphia, U.S, Mar. 2005.

[14]. J. Wu and X. L. Zhang, “Efficient multiple kernel support vector machine based voice activity detection,” *IEEE Signal Process. Lett*., vol. 18, no. 8, pp. 466–499, Jun. 2011.

[15]. P. Teng and Y. Jia, “Voice activity detection via noise reducing using nonnegative sparse coding,” in *IEEE Signal Process. Lett*., vol. 20, no. 5, pp. 475–478, Mar. 2013.

[16]. T. Ng, B. Zhang and N. L, “Developing a speech activity detection system for the DARPA RATS program,” in *Proceedings of INTERSPEECH*, 2012.

[17]. V. H and S. H, “Hidden-Markov-model-based voice activity detector with high speech detection rate for speech enhancement,” *IET Signal Processing*, vol. 6, no. 1, pp. 54–63, Feb. 2012.

[18]. X. Zhang and J. Wu, “Deep Belief Networks Based Voice Activity Detection,” in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697-710, Apr. 2013.

[19]. T. Hughes and K. Mierle, “Recurrent neural networks for voice activity detection,” in *Proc. ICASSP*, pp. 7378–7382, Vancouver, CA, May 2013.

[20]. S. Y. Chang, B. Li, G. Simko, T. N. Sainath, A. Tripathi, A. V. D. Oord and O. Vinyals, “Temporal modeling using dilated convolution and gating for voice-activity-detection,” in *Proc. ICASSP*, pp. 5549–5553, Calgary, CA, Apr. 2018.

[21]. I. Ariav and I. Cohen, “An end-to-end multimodal voice activity detection using WaveNet encoder and residual networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 265–274, May 2019.

[22]. A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, Sep. 2016.

[23]. X. Zhang and D. Wang, “Boosting contextual information for deep neural network based voice activity detection,” in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 252–264, Dec. 2016.

[24]. F. Sohrab and H. Erdogan, "Recognize and separate approach for speech denoising using nonnegative matrix factorization," in *Proceedings of EUSIPCO*, pp. 1083-1087, Nice, France, Aug. 2015.

[25]. K. Kinoshita, M. Delcroix, A. Ogawa and T. Nakatani, "Text-informed speech enhancement with deep neural networks," in *Proceedings of INTERSPEECH*, pp. 1760–1764, 2015.

[26]. Y. Tachioka, "Dnn-Based Voice Activity Detection Using Auxiliary Speech Models in Noisy Environments," in *Proc. ICASSP*, pp. 5529-5533, Calgary, CA, Apr. 2018.

[27]. N. Moritz, T. Hori and J. L. Roux, "Triggered Attention for End-to-end Speech Recognition," in *Proc. ICASSP*, pp. 5666-5670, Brighton, UK, May 2019.

[28]. X. Wang, R. Li, S. H. Mallidi, T. Hori, S. Watanabe and H. Hermansky, "Stream Attention-based Multi-array End-to-end Speech Recognition," in *Proc. ICASSP*, pp. 7105-7109, Brighton, UK, May 2019.

[29]. J. Kim and M. Hahn, "Voice Activity Detection Using an Adaptive Context Attention Model," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1181-1185, Aug. 2018.

[30]. Y. Lee, J. Min, D. K. Han and H. Ko, "Spectro-Temporal Attention-Based Voice Activity Detection," *IEEE Signal Processing Letters*, vol. 27, pp. 131-135, Dec. 2020.

[31]. A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, Upper Saddle River, NJ, USA:Prentice-Hall, Aug. 2009.

[32]. K. K. Paliwal, "Spectral subband centroids as features for speech recognition," *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 124-131, Santa Barbara, U.S, Dec. 1998.

[33]. N. Q. Duong and H.T. Duong, “A review of audio features and statistical models exploited for voice pattern design,” *arXiv preprint arXiv:1502.06811*, Feb. 2015.

[34]. F. Faraji, Y. Attabi, B. Champagne and W. P. Zhu, “On the Use of Audio Fingerprinting Features for Speech Enhancement with Generative Adversarial Network,” *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 1-6, Coimbra, Portugal, 2020.

[35]. R. Martin, “Noise power spectral density estimation based on optimal smoothing and minimum statistics,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504-512, Jul. 2001.

[36]. M. Parchami, W. Zhu, B. Champagne and E. Plourde, “Recent Developments in Speech Enhancement in the Short-Time Fourier Transform Domain,” *IEEE Circuits and Systems Magazine*, vol. 16, no. 3, pp. 45-77, Aug. 2016.

[37]. G. W. Lee and H. K. Kim, “Multi-task learning UNet for single-channel speech enhancement and mask-based voice activity detection,” *Applied Sciences*, vol. 10, no. 9, May 2020.

[38]. O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer assisted intervention*, pp. 234–241, Springer, Cham, Oct. 2015.

[39]. H.S. Choi, J.H. Kim, J. Huh, A. Kim, J.W. Ha and K. Lee, “Phase-aware speech enhancement with deep complex u-net,” in *International Conference on Learning Representations*, Sep. 2018.

[40]. D. Griffin and Jae Lim, “Signal estimation from modified short-time Fourier transform,” in *ICASSP*, pp. 804-807, Boston, USA, Apr. 1983.

[41]. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need,” in *Neural Information Processing Systems (NIPS)*, pp. 5998-6008, 2017.

[42]. Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig and M. L. Seltzer, “Transformer-Based Acoustic Modeling for Hybrid Speech Recognition,” in *Proc. ICASSP*, pp. 6874-6878, Barcelona, Spain, May 2020.

[43]. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, Oct. 2021.

[44]. H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan and L. Zhang, “CvT: Introducing Convolutions to Vision Transformers,” *arXiv preprint arXiv:2103.15808*, Mar. 2021.

[45]. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” *arXiv preprint arXiv:2103.14030*, Mar. 2021.

[46]. Y. Li, K. Zhang, J. Cao, R. Timofte and L. V. Gool, “LocalViT: Bringing Locality to Vision Transformers,” *arXiv preprint arXiv:2104.05707*, Apr. 2021.

[47]. J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv*:*1607.06450*, Jul. 2016.

[48]. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. CVPR*, pp. 1251–1258, 2017.

[49]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. CVPR*, pp. 4510–4520, 2018.

[50]. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus and D. S. Pallett, “DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1,” *NASA STI/Recon*, USA, Tech. Rep. NISTIR 4930, vol. 93, Feb. 1993.

[51]. A. Varga and H. J. M. Steeneken, “Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems,” *Speech Commun*., vol. 12, no. 3, pp. 247–251, Jul. 1993.

[52]. V. Panayotov, G. Chen, D. Povey and S. Khudanpur, “LibriSpeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, pp. 5206–5210, South Brisbane, Australia, Apr. 2015.

[53]. H.G. Hirsch and D. Pearce, “The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *Proc. Autom. Speech Recognit.: Challenges Millenium ISCA Tut. Res. Workshop*, pp. 181–188, Paris, France, Sep. 2000.

[54]. I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, Nov. 2017.

[55]. D. Hendrycks and K. Gimpel, “Bridging nonlinearities and stochastic regularizers with gaussian error linear units,” *arXiv preprint aeXiv:1606.08415*, 2016.

[56]. X. Zhang, F. Wei and M. Zhou, “HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization,” *arXiv preprint arXiv:1905.06566*, 2019.