AN EFFICIENT TRANSFORMER-BASED MODEL FOR VOICE ACTIVITY DETECTION

Yifei Zhao *Benoit Champagne*

Department of Electrical & Computer Engineering, McGill University, Montreal, Canada

2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP) | 978-1-6654-8547-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLSP55214.2022.9943501

ABSTRACT

Voice Activity Detection (VAD) aims to distinguish, at a given time, between desired speech and non-speech. 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). To deal with this issue, we propose a novel transformer-based architecture for VAD with reduced computational complexity by implementing efficient depth-wise convolutions on feature patches. The proposed model, named Tr-VAD, demonstrates better performance compared to baseline methods from the literature in a variety of scenarios considered with the smallest possible number of parameters. The results also indicate that the use of a combination of Audio Fingerprinting (AFP) features with Tr-VAD can guarantee better performance.

***Index Terms*—** Voice activity detection, transformer-based architecture, audio fingerprinting

1. INTRODUCTION

Voice Activity Detection (VAD) refers to a family of methods that that can determine the presence or absence of human speech in a signal at a given time. It often serves as an important preprocessor for many speech-based applications, including speaker identification, speech recognition, keyword spotting and hearing aids [1, 2]. The primary difficulty in developing robust VAD systems lies in distinguishing speech from a variety of stationary and non-stationary noises, especially in low SNR environments. Early VAD studies focused on power calculations in the time domain [3, 4]. Subsequently, further methods were developed that rely on the use of classical or handcrafted features of speech signals, such as Zero Crossing Rate (ZCR) [5], spectral or cepstral features [6, 7], higher order statistics [8] and pitch detection [9]. The Likelihood Ratio Test (LRT), which assumes prior knowledge of the speech signal and noise distributions, is wildly used in VAD [10].

Unlike conventional methods which seek to exploit underlying properties and modeling of acoustic features, data-driven machine learning methods, such as linear discriminant analysis [11], Support Vector Machines (SVM) [12], sparse coding [13], have shown good classification results on the VAD task. These methods provide additional flexibility in incorporating prior knowledge, such as manually labeled data and fusing multiple acoustic features. Recognizing the effectiveness of data-driven methods, several VAD approaches based on deep learning model, including Deep Neural Networks (DNN) [14, 15], Deep Belief Network (DBF) [16], Convolutional Neural Network (CNN) [17] and Recurrent Neural Network (RNN) [18, 19] which demonstrate better performance over conventional methods. More recently, there has been a growing interest in the use of transformer-based architectures for natural language processing (NLP) [20], computer vision (CV) [21, 22] and automatic speech recognition (ASR) [23]. These architectures have demonstrated state-of-the-art performance in many tasks. Typically, while RNNs have difficulty in learning long-term dependencies, transformer-based approaches overcome the issue with the use of self-attention mechanism.

Inspired by these recent methods, we herein propose a novel transformer-based DNN architecture for VAD, called Tr-VAD, which performs efficient convolutions on feature patches. Compared to the original transformer approach in [20] which uses self-attention mechanism to capture global dependencies, the proposed model splits the acoustic features into non-overlapping patches and applies depth-wise convolutions to further introduce locality to the transformer architecture. To the best of our knowledge, this is the first attempt to apply transformer-based architecture to the VAD task. The performance of the proposed Tr-VAD is evaluated by means of F1-score and Detection Cost Function (DCF) metrics, and compared to other state-of-the-art approaches. Our experiments show that the proposed method achieves superior performance in almost all scenarios considered, with the fewest possible number of parameters.

The rest of the paper is organized as follows. Section 2 briefly describes the acoustic features used in our study. Section 3 develops the architecture of the proposed Tr-VAD model. Section 4 presents the experimental setup and compares the performance of different methods. This is followed by a conclusion in Section 5.

2. FEATURE EXTRACTION

In this section, we briefly discuss the preprocessing steps needed for the extraction of the acoustic features used in this work. The input noisy speech signal $x[n]$ is modeled as:

$$x[n] = s[n] + w[n] \tag{1}$$

where $s[n]$ denotes the clean speech signal, $w[n]$ denotes an additive background noise, and $n \in \mathbb{Z}$ is the discrete-time index. Processing is implemented in the frequency domain by applying the Short-Time-Fourier-Transform (STFT) to $x[n]$:

$$X(t,f) = \sum_{n=0}^{N-1} x[n + tL_{\text{hop}}]h[n]e^{-j2\pi fn/N} \tag{2}$$

where t is the frame index, L_{hop} is the frame advance, $f \in \{0, 1, 2, ..., N/2\}$ 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 warped according to the Mel scale using a bank of spectral shaping filters, in

Support for this work was provided by a grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada, with industrial sponsor Microchip Technology (Ottawa, Canada).

978-1-6654-8547-0/22//$31.00 ©2022 IEEE

order to adapt the frequency resolution to the properties of the human ear. The logarithm function is then applied to the output of each filter, yielding

$$\mathrm{FB}(t,b) = 20\log_{10}\{\sum_{f=l_b}^{h_b} u_b(f)\,|X(t,f)|^2\} \tag{3}$$

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

The Discrete Cosine Transform (DCT) – Type III [24] is applied to the log-Mel filter bank features to obtain the Mel-Frequency Cepstral Coefficients (MFCC):

$$\mathrm{MFCC}(t,b) = \frac{1}{20}\sqrt{\frac{2}{B}}\sum_{b=0}^{B-1}\mathrm{FB}(t,b)\cos\left(\frac{p\pi}{B}(b-0.5)\right) \tag{4}$$

We define the MFCC feature vector of the current data frame as: $\mathbf{MFCC}_t = [\mathrm{MFCC}(t,0),...,\mathrm{MFCC}(t,B-1)]$.

The Spectral Subband Centroid (SSC) [25] is often used to measure the central frequency of a subband spectrum. To calculate the SSC for the b^{th} shaping filter, a weighted average is applied as follows:

$$\mathrm{SSC}(t,b) = \frac{\sum_{f=l_b}^{h_b} f\,u_b'(f)\,|X(t,f)|^2}{\sum_{f=l_b}^{h_b} u_b'(f)\,|X(t,f)|^2} \tag{5}$$

where $u_b'(f)$ is the subband filter. For simplicity, the same set of filters $u_b(f)$ is used in this work for the calculation of the MFCC and SCC features. For efficient training, we use the Normalized SSC (NSSC), taking values in $[-1,1]$ and computed as: $\mathrm{NSSC}(t,b) = (\mathrm{SSC}(t,b) - (h_b - l_b))/(h_b - l_b)$. Similarly, the NSSC feature vector at the t^{th} frame is defined as: $\mathbf{NSSC}_t = [\mathrm{NSSC}(t,0),...,\mathrm{NSSC}(t,B-1)]$.

The Audio Fingerprinting Combination (AFPC), a combination of MFCC and NSCC, has demonstrated superior performance for speech enhancement when used as input to a generative adversarial network (GAN) [26]. In our work, we shall make use of a similar concatenation of features:

$$\begin{aligned}\mathbf{AFPC}_t = [&\mathbf{MFCC}_t, \Delta\mathbf{MFCC}_t, \Delta^2\mathbf{MFCC}_t,\\ &\mathbf{NSSC}_t, \Delta\mathbf{NSSC}_t, \Delta^2\mathbf{NSSC}_t]\end{aligned} \tag{6}$$

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

3. PROPOSED TRANSFORMER-BASED VAD ARCHITECTURE

In this section, we propose a novel transformer-based VAD method, called Tr-VAD, which 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 Tr-VAD model consists of a feature embedding layer, several transformer encoder blocks, and a classifier as illustrated in Fig. 1. These components are described in more details in the following sections.

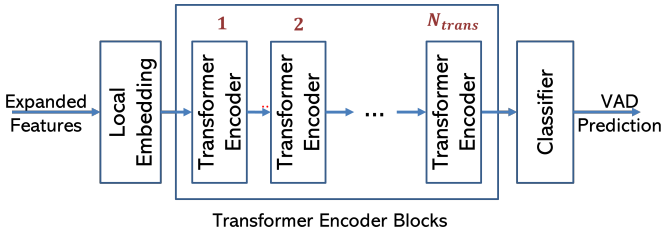


Fig. 1. Architecture of the proposed model.

3.1. Feature Embedding

Let $\{\mathbf{X}_t, y_t^{\text{truth}}\}_{t=0}^{T-1}$ represents the acoustic data available for training the model, where $\mathbf{X}_t \in \mathbb{R}^D$ is the acoustic feature vector at the t^{th} frame, $y_t^{\text{truth}} \in \{0,1\}$ is the corresponding true VAD label, and T is the total number of frames. The acoustic data in each frame is first expanded by using $L = 2k+1$ neighboring frames with relative index $l \in \mathcal{T} = \{-ku,-(k-1)u,...,-u,0,u,...,(k-1)u,ku\}$, where integer u is the step size, and k determines the number of neighboring frames. The expanded data is represented by:

$$\mathbf{X}_t' = \{\mathbf{X}_{t+l} : l \in \mathcal{T}\} \in \mathbb{R}^{L\times D}, \mathbf{y}_t^{\text{truth}} = \{y_{t+l}^{\text{truth}} : l \in \mathcal{T}\} \in \mathbb{R}^L \tag{7}$$

The expanded feature vectors are used as input to the embedding module, which consists of a Fully Connected Network (FCN) followed by a 1-D convolutional layer. Compared to absolute positional embedding strategies (such as sinusoid positional embedding and learnable 1-D positional embedding), convolutional embedding can extract relative positional information and learn useful short-range spectral-temporal patterns [23], which are quite important for speech related models where the local audio signals are highly correlated. We denote the output of the embedding layer as $\bar{\mathbf{X}}_t^1 \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$, where $\tilde{L}$ and $\tilde{D}$ denote the temporal and feature dimensions, respectively.

3.2. Depth-Wise Transformer Blocks with Feature Patches

Each one of the N_{trans} depth-wise transformer blocks in Fig. 1 includes two normalization layers, a Multi-Headed Self Attention (MHSA) module, and a Feed-Forward Network (FFN), configured as shown in Fig. 2. Let the input features of the i^{th} transformer block be denoted as $\bar{\mathbf{X}}_t^i \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$, where $i \in \{1,2,...,N_{\text{trans}}\}$. As

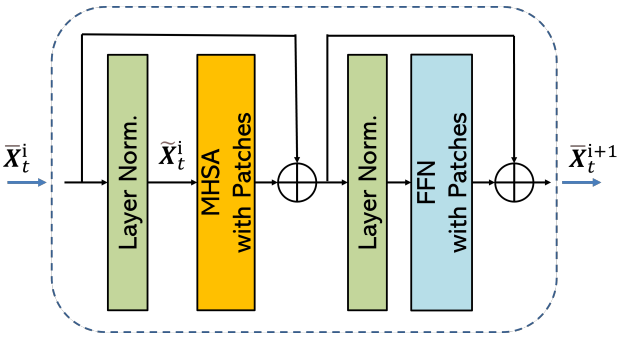
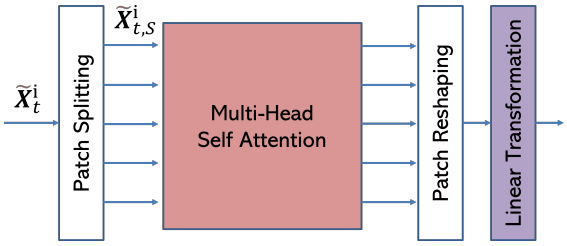


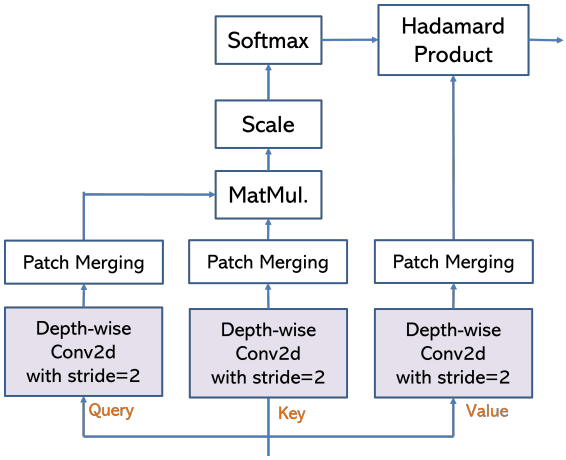
Fig. 2. Block diagram of the i^{th} transformer block.

illustrated in Fig. 2, the layer normalization is applied to the feature

matrix $\bar{\mathbf{X}}_t^i$, resulting in the normalized feature matrix $\widetilde{\mathbf{X}}_t^i \in \mathbb{R}^{\tilde{L}\times\tilde{D}}$ which is passed to the MHSA module whose structure is described below.



(a) Block Diagram of the proposed MHSA module.



(b) Internal Structure of MHSA in Tr-VAD.

Fig. 3. Multi-headed Self Attention Module

Multi-Head Self Attention with Feature Patches: The internal structure of the MHSA module is illustrated in Fig. 3(a). The input $\widetilde{\mathbf{X}}_t^i$ is first split into patches by decomposing the temporal dimension $\tilde{L}$ and feature dimension $\tilde{D}$ are split into non-overlapping $P_1 \times P_2$ pieces, as represented by:

$$\widetilde{\mathbf{X}}_{t,S}^i = \text{Split}(\widetilde{\mathbf{X}}_t^i) \in \mathbb{R}^{D_{\text{split}}\times P_1\times P_2} \tag{8}$$

where integer P_1 and P_2 specify the split factors, $\text{Split}(\cdot)$ stands for the split operation, and $D_{\text{split}} = \frac{\tilde{L}}{P_1} \times \frac{\tilde{D}}{P_2}$. In contrast to earlier works where MHSA is applied along a single dimension, here we propose to extend this concept to both temporal and feature dimensions. This new scheme allows the model to attend to multiple frames and feature information at different positions. In contrast, the Swin Transformer [22] uses shifted windows to allow communication among different patches and to increase the receptive field. In our case however, the acoustic features already include information from neighboring frames which carries contextual redundancy; hence the use of shifted widow is not necessary.

Depth-Wise (DW) separable convolution blocks [27] are used in this work to obtain the attention matrices. The DW convolution emphasizes local information which is missing in the FFN-based transformer network. Therefore, the DW convolution-based transformer is cable of modelling both local (short-range) and global (long-range) dependencies with reduced parameters and computational cost. Each block consists of a DW convolutional layer, a batch normalization layer, and a 1×1 Point-Wise (PW) convolutional layer. As indicated in Fig. 3(b), we use a stride of 2 when applying convolution to the feature matrix $\widetilde{\mathbf{X}}_{t,S}^i$. Letting $\text{DW}(\cdot)$ denote the DW convolution operation, the mapped feature matrix $\text{DW}(\widetilde{\mathbf{X}}_{\text{t,S}}^{\text{i}}) \in \mathbb{R}^{D_{\text{split}}\times\frac{P_1}{2}\times\frac{P_2}{2}}$ is then reshaped into $\tilde{\mathbf{X}}_{t,\text{DW}}^i$:

$$\tilde{\mathbf{X}}_{t,\text{DW}}^i = \text{Reshape}\left(\text{DW}\left(\widetilde{\mathbf{X}}_{t,S}^i\right)\right) \in \mathbb{R}^{\frac{\tilde{L}}{P_1}\times\frac{\tilde{D}}{P_2}\times\frac{P_1P_2}{4}} \tag{9}$$

where $\text{Reshape}(\cdot)$ is the reshape operation. Let $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the query, key and value, respectively, as obtained from matrix $\tilde{\mathbf{X}}_{t,\text{S}}^i$ by using different convolution weights in Eq. (9). The scaled dot-product attention operation is applied as follows:

$$\tilde{\mathbf{X}}_{t,att}^i = \text{Softmax}\left(\frac{\mathbf{Q}^T\mathbf{K}}{\sqrt{N_d}} + B_p\right)\cdot\mathbf{V} \in \mathbb{R}^{\frac{\tilde{L}}{P_1}\times\frac{\tilde{D}}{P_2}\times\frac{P_1P_2}{4}} \tag{10}$$

where $\cdot$ is the element-wise product, and $B_p \in \mathbb{R}^{\frac{\tilde{L}}{P_1}\times\frac{\tilde{D}}{P_2}\times\frac{\tilde{D}}{P_2}}$ is a learnable bias term. Since we use a stride of 2 for the queue, key, and value mapping, the size of the temporal and feature dimensions are reduced by a factor of 2, and the computational cost for self-attention operation is thus reduced by a factor of 4^3. Such strategy comes with negligible performance degradation as the input features contain redundant information.

Referring to the Fig. 3(b), the attention output $\widetilde{\mathbf{X}}_{t,att}^i$ is reshaped and then passed to a linear transformation module which includes an 1-D convolutional layer and an FCN.

Feed-Forward Network with Feature Patches: Similar to the MHSA module, convolution-based FFN are used in the proposed method. The FFN firstly splits the input features into $P_1 \times P_2$ patches, then sequentially applies a 1×1 PW convolution, a 3×3 DW convolution block and another 1×1 PW convolution to the feature patches. Finally, the shape of the feature patches are restored.

3.3. Classifier

Recall from Fig.1 that the output of the transformer encoder blocks is finally fed to the classifier. As illustrated in Fig.4, the classifier processes feature patches by applying a DW convolution block, and the output feature matrix $\mathbf{X}_{t,c} \in \mathbb{R}^{D_{\text{split}}\times\frac{P_1}{2}\times\frac{P_2}{2}}$ is then reshaped to $\widetilde{\mathbf{X}}_{t,\text{c}} \in \mathbb{R}^{\frac{P_1}{2}\times\frac{P_2 D_{\text{split}}}{2}}$. The following two FCNs probe the hidden information and compress the last feature dimension to 1. Finally, the output of the FFNs is passed through a sigmoid activation to predict the probability of the presence of speech, represented by compressed vector $\mathbf{y}_t \in \mathbb{R}^{\frac{P_1}{2}} = \mathbb{R}^L$, since in our work, $P_1 = 2L$. The soft prediction corresponding to the t^{th} frame, $\hat{y}_t$, can be computed by aggregating all the soft predictions $\mathbf{y}_t$ relative to frame t across $l \in \mathcal{T}$. The final decision label $\bar{y}_t$ is obtained by comparing the soft prediction $\hat{y}_t$ with a threshold θ_{T}:

$$\hat{y}_t = \frac{1}{L}\sum_{l\in\mathcal{T}} y_{t+l}, \quad \bar{y}_t = \begin{cases} 1, & \text{if } \hat{y}_t \geq \theta_{\text{T}} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where y_{t+l} is the $(t+l)^{th}$ component of $\mathbf{y}_t$.

For training the proposed transformer based VAD network, the cross entropy loss is calculated based on the classifier output:

$$\mathcal{L} = -\sum_{t=ku}^{T-ku-1}\sum_{l'\in\mathcal{T}}\left(y_{t+l}^{\text{truth}}\log y_{t+l} + \left(1-y_{t+l}^{\text{truth}}\right)\log\left(1-y_{t+l}\right)\right) \tag{12}$$

where y_{t+l}^{truth} is the l^{th} component of ground truth label $\mathbf{y}_t^{\text{truth}}$.

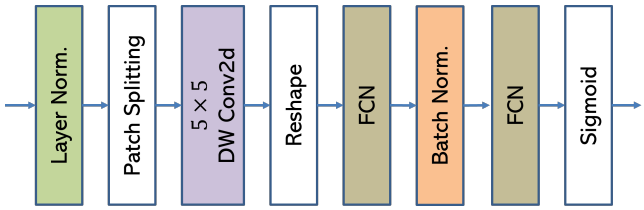


Fig. 4. Architecture of the Classifier.

4. RESULTS

In this section, we first describe our experimental setup and then present the comparative results of different methods.

4.1. Experimental Setup

The TIMIT training dataset [28] is used to train and validate the proposed and baseline models, with 95% of speech utterances used for training, and 5% for validation. The TIMIT dataset has ground truth labels. To balance the testing conditions, a 1-second silence section is added before and after each utterance. Eight types of noises (babble, F16, destroyer, M109, Volvo, white, and two factory noises) from the NOISEX-92 dataset [29] are used to corrupt the training set, with SNR levels of -10, -5, 0, 5, 10 dB. The TIMIT test dataset is used in the test phase, where as above, a 1-second silence is added before and after each utterance. All 8 types of unseen noises from the AURORA noise dataset [30] are used to corrupt the clean speech, with SNRs of -5, 0, 5 and 10 dB.

Each utterance from the training and test dataset, with sampling rate 16kHz, is framed by applying a 32 ms Hanning window with 16 ms window shift. Accordingly, the size of the STFT used for spectral analysis in (2) is set to $N = 512$. The Tr-VAD method employs the AFPC features as discussed in Section 2.1. Specifically, 16 coefficients are computed for each one of MFCC, ΔMFCC, Δ^2 MFCC, NSSC, and Δ NSSC, resulting in a total of 80 AFPC features. Parameters k, u, and L needed to construct the expanded data set, are chosen as 4, 4, and 9, respectively. For training, a mini-batch approach with a batch size of 512 is applied, along with the AdamW optimizer [31] using a cosine decay learning rate scheduler and 5000 iterations of linear warm-up. An initial learning rate of 10^{-3}, a weight decay of 0.05 and a final learning rate of 5×10^{-6} after 4×10^5 iterations are used. The Gaussian Error Linear Unit (GELU) [32] is chosen as the activation function. Model parameters D, $\tilde{L}$, $\tilde{D}$, 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, while the dropout rate is 0.1. Other model parameter settings can be found in Table 1.

Tr-VAD is compared with the following baseline methods:

• **rVAD [9]:** Unsupervised VAD method exploiting pitch information by calculating the *a posteriori* SNR weighted energy difference.

• **Adaptive Contextual Attention Model (ACAM) [19]:** Original attention-based VAD model which only applies temporal attention.

• **Spectro-Temporal Attention Model (STAM) [15]:** Extended attention-based VAD model which exploits both spectral and temporal information.

• **DCU-10 [33]:** DNN-based speech enhancement model including 10 complex layers, which we extend to predict VAD labels. That is, the estimated complex ideal ratio mask is averaged along the frequency axis and the magnitude of the resulting average is compared with a threshold.

Table 1. Parameter setting for the Proposed Tr-VAD method

Layer Name	Units In	Units Out	Kernel Size	Stride Size
FCN in Embedding	80	324		
1-D Conv. in Embedding	9	54	5	2
DW 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	108	(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		

Table 2. Comparison of F1-Score and DCF versus SNR

SNR	Metric	rVAD	DCU-10	ACAM	STAM	Tr-VAD
-5 dB	Fl	79.5	86.4	85.9	97.7	**98.6**
	DCF	8.3	7.8	6.2	1.5	**0.8**
0 dB	FI	86.0	89.8	90.7	98.0	**98.8**
	DCF	5.8	5.7	3.7	1.3	**0.7**
5 dB	F1	92.4	92.3	95.4	98.3	**99.0**
	DCF	3.9	4.0	2.6	1.2	**0.6**
10 dB	Fl	94.0	94.2	96.0	98.4	**99.1**
	DCF	3.4	2.8	2.3	1.1	**0.6**

A default parameter setting is employed for rVAD, while other methods are trained using the same approaches as proposed in the above references.

For comparison, the F1-score and the Detection Cost Function (DCF) [8] are selected as evaluation metrics. The F1-score is a common evaluation index for binary classification problems, defined as:

$$\text{F1} = 2\,\text{TP}\,/(2\,\text{TP} + \text{FP} + \text{FN}) \tag{13}$$

where TP, FP, FN represent the number of true positive, false positive, and false negative cases, respectively. The DCF reflects the wrong performance of the model and is defined as:

$$\text{DCF} = (1-\beta)\,\text{P}_{\text{FN}} + \beta\,\text{P}_{\text{FP}} \tag{14}$$

where P_{FP} is the rate of FP, P_{FN} is the rate of FN, and β is a weight set to 0.25 in order to penalize missed speech frames more heavily. Higher/lower values of the F1-score and DCF metrics indicate better performance.

4.2. Results and Discussion

Table 2 shows the averaged results of F1-score and DCF on TIMIT dataset for different SNR levels. It is clear that DNN-based methods generally achieve better results than rVAD, and the attention-based methods, ACAM and STAM, further improve the performance. The proposed Tr-VAD outperforms all baseline methods across all SNR levels.

Table 3 shows the size of different models and the average run time for processing a 10-second utterance. Experiments were conducted on a platform equipped with Intel Core i7-10700F CPU and NVIDIA GeForce RTX 2070 SUPER GPU. The results validate the efficiency of the proposed Tr-VAD with 32% less parameters and 38% faster execution time.

Table 3. Number of Parameters and Average Running Time

Methods	rVAD	DCU-10	ACAM	STAM	Tr-VAD
Parameters	NA	2808K	957K	559K	**376K**
Run Time (ms)	86	251	1263	132	**82**

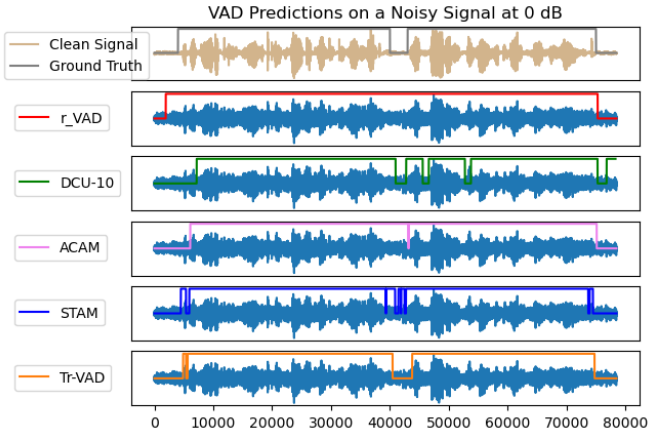


Fig. 5. Comparison of the hard VAD decisions made by different methods.

Fig. 5 compares the hard VAD decisions predicted by different methods on a representative utterance. The clean signal sample is chosen from the 'test_clean' dataset of LibriSpeech [34]. The transcript of the 4.9-second 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 and the true VAD label, while the remaining sub-figures show the hard VAD decisions obtained by applying different methods to a noisy version of the utterance. The latter is obtained by adding the 'airport' noise from the AURORA noise corpus to the clean signal, with the SNR set to 0 dB. It can be seen from Fig. 5 that the proposed Tr-VAD accurately predicts the start and end points of the utterance, and can detect as well the non-speech part (less than 0.2 second) near the middle.

To validate the effectiveness of each part of the proposed method, we further conduct ablation studies on it. Specifically, five different variations of Tr-VAD are used as follows:

- Tr-VAD$_0$: Baseline Tr-VAD using AFPC feature, DW-based FFN and DW-based MHSA.
- Tr-VAD$_1$: Similar to Tr-VAD$_0$ but using **Log-Mel filter bank** features instead of AFPC features.
- Tr-VAD$_2$: Similar to Tr-VAD$_0$ but using **MFCC** features instead of AFPC features.
- Tr-VAD$_3$: Similar to Tr-VAD$_0$ but using AFPC features but the **FFN** is FCN based.
- Tr-VAD$_4$: Similar to Tr-VAD$_0$ but using the AFPC features but the **MHSA** is FCN based.

As shown in Table 4, the baseline Tr-VAD using the AFPC features exhibits a 0.4% increase in F1-score compared to the modified Tr-VAD$_1$ using instead the Log-Mel filter bank features and Tr-VAD$_2$ using MFCC features. With the use of FCN-based FFN, Tr-VAD$_3$ achieves similar F1 score and DCF as the baseline Tr-VAD$_0$, but requires about 4 times more parameters. Tr-VAD$_4$, which uses the original FCN-based MHSA in [20], only achieves a minor improvement of 0.1% on DCF but requires more than twice as many parameters

Table 4. Ablation Study on the Proposed Method

Evaluation Metrics	Tr-VAD$_0$	Tr-VAD$_1$	Tr-VAD$_2$	Tr-VAD$_3$	Tr-VAD$_4$
# Parameters	**376K**	**376K**	**376K**	1527K	901K
F1 Score	**98.9**	98.5	98.5	98.8	**98.9**
DCF	0.7	0.8	0.9	0.7	**0.6**

Table 5. Influence of Neighboring Frames on the Proposed Method

Evaluation Metrics	$u = 4$	$u = 3$	$u = 2$
F1 Score	**98.9**	98.8	98.6
DCF	**0.7**	**0.7**	0.8

The influence of neighboring frames on the performance of the proposed Tr-VAD method is also studied. In the above experiments, $L = 2k + 1 = 9$ frames are used, where each frame is separated by $u = 4$, however u may be too large for real-time applications, as the resulting window covers $2ku = 16$ frames from the past and future signal streams, which may result in high latency in some scenarios. As shown in Table 5, by reducing the step size u and keeping the total number of frames L and k the same, it becomes easier to implement Tr-VAD in real-time applications, at the cost of slight loss in performance. It is noteworthy that with $k = 4$, $u = 2$, resulting in 8 neighboring frames, the Tr-VAD still outperforms the STAM model which requires 19 neighboring frames in Table 2.

5. CONCLUSION

In this paper, a novel transformer-based VAD model, referred to as Tr-VAD, was proposed and validated. The proposed approach reduces computational complexity by splitting the acoustic features into patches and applying depth-wise convolutions, thereby allowing the model to predict the presence or absence of speech more efficiently. The proposed model consists of a feature embedding layer, several transformer encoder blocks, and a classifier. The experimental results show that compared to state-of-the-art benchmark approaches, our proposed Tr-VAD method achieves better results in term of F1 score and DCF under different noise conditions, and this with much fewer parameters and significantly faster execution. The results also indicate that the use of a combination of AFPC features with Tr-VAD can guarantee better performance.

6. REFERENCES

[1] J. Ramirez, J. M. Girriz and J. C. Segura, "Voice activity detection. Fundamentals and speech recognition system robustness," in M. Grimm and K. Kroschel (Eds.), *Robust Speech Recognition and Understanding*, I-Tech: Vienna, 2007, pp. 1-22.

[2] X. Yang, B. Tan, J. Ding, J. Zhang, and J. Gong, "Comparative study on voice activity detection algorithm," in *Proc. Int. Conf. Electric. Control Eng.*, pp. 599–602, Wuhan, China, Jun. 2010.

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

[4] K. Srinivasan and A. Gersho, "Voice activity detection for cellular networks," in *Proc. IEEE Speech Coding Workshop*, pp. 85–86, Quebec, Canada, Oct. 1993.

[5] A. Benyassine, E. Shlomot, H.Y. Su, D. Massaloux, C. Lamblin and J.P. Petit, "ITU-T Recommendation G.729 Annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," in *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64-73, Sept. 1997.

[6] J. A. Haigh and J. S. Mason, "Robust voice activity detection using cepstral features," in *Proc. IEEE Region 10 Int. Conf. on Computers, Communications and Automation*, vol. 59, pp. 321–324, Beijing, China, Oct. 1993.

[7] J. Shen, J. Hung and L. Lee, "Robust entropy-based endpoint detection for speech recognition in noisy environments," in *Proc. Int. Conf. on Spoken Language Processing*, pp. 232–235, Sydney, Australia, Nov. 1998.

[8] 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.

[9] 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. 2019.

[10] 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.

[11] 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, USA, Mar. 2005.

[12] 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.

[13] P. Teng and Y. Jia, "Voice activity detection via noise reducing using non-negative sparse coding," *IEEE Signal Process. Lett*., vol. 20, no. 5, pp. 475–478, Mar. 2013.

[14] 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.

[15] Y. Lee, J. Min, D. K. Han and H. Ko, "Spectro-temporal attention-based voice activity detection," *IEEE Signal Processing Lett.*, vol. 27, pp. 131-135, Dec. 2020.

[16] X. Zhang and J. Wu, "Deep belief networks based voice activity detection," in *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697-710, April 2013.

[17] 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, 2019.

[18] G. Wang and W. Zhang, "An RNN and CRNN Based Approach to Robust Voice Activity Detection," in *Proc. ASPIPA ASC*, Lanzhou, China, pp. 1347-1350, Nov. 2019.

[19] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Lett.*, vol. 25, no. 8, pp. 1181-1185, Aug. 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[21] 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.

[22] 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.

[23] 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.

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

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

[26] F. Faraji, Y. Attabi, B. Champagne and W. P. Zhu, "On the use of audio fingerprinting features for speech enhancement with generative adversarial network," *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 1-6, Coimbra, Portugal, 2020.

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

[28] 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.

[29] 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.

[30] 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.

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

[32] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with Gaussian error linear units," *arXiv preprint aeXiv:1606.08415*, 2016.

[33] 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 *Int. Conf. on Learning Representations*, Sep. 2018.

[34] 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.