



# Temporal Convolutional Network with Smoothed and Weighted Losses for Distant Voice Activity and Overlapped Speech Detection

Shaojie Li<sup>1</sup>, Qintuya Si<sup>2</sup>, De Hu<sup>1,\*</sup>

<sup>1</sup>College of Computer Science, Inner Mongolia University, China

<sup>2</sup>College of Electronic and Information Engineering, Inner Mongolia University, China

lishaojie@mail.imu.edu.cn, siqty@imu.edu.cn, cshood@imu.edu.cn

## Abstract

Voice Activity Detection (VAD) and Overlapped Speech Detection (OSD) are key steps in various audio/speech processing tasks. Recent advances in VAD or OSD are moving toward using Temporal Convolutional Networks (TCNs) with frame-independent cross-entropy loss, which may be unable to cope with transient errors or boundary errors (caused by weak recordings at speech boundaries). In this paper, we formulate two novel losses, namely smoothed loss and weighted loss, in which the former copes with transient errors while the latter deals with boundary errors. In addition, we adopt Mel Frequency Cepstral Coefficients (MFCCs) and Instantaneous Correlation Coefficients (ICCs) as the acoustic and spatial features to drive the model. To improve computing efficiency, we also propose a spatial feature extraction module by selecting those frequencies with information-rich ICCs, which delivers good lightweight nature. Numerical experiments validate the efficacy of the proposed method.

**Index Terms:** voice activity detection, overlapped speech detection, temporal convolutional network, loss function.

## 1. Introduction

Voice Activity Detection (VAD) [1, 2] and Overlapped Speech Detection (OSD) [3, 4] are crucial steps in various audio/speech processing tasks, including speaker diarization [5, 6, 7], automatic speech recognition [8, 9] and speaker recognition [10, 11], to name a few. VAD aims to distinguish speech and non-speech segments in audio streams while OSD aims to identify segments containing at least two simultaneously active speakers.

Early studies on VAD were based on simple detection rules by leveraging acoustic features like energy feature [12], zero-crossing rate [13] and linear predictive coding [14]. Acoustic features were also used to train statistical models [15, 16, 17]. With the development of deep learning, Convolutional Neural Networks (CNN) [18, 19, 20] was used in VAD, which improved the performance significantly. Similarly, most recent OSD approaches were also based on deep neural networks. Geiger et al. [21] applied the LSTM neural networks to OSD, and some other works were developed based on CNNs [22, 23]. Due to the strong correlation between VAD and OSD, recent advances are moving toward implementing them together via Temporal Convolutional Networks (TCNs) [24, 25].

Most existing VAD and OSD researches were developed in a close-talk scenario, where the speech signals are captured by microphones near the speakers. However, in some cases (e.g.

\* Corresponding author. The source code is available at <https://doi.org/10.5281/zenodo.15516737>.

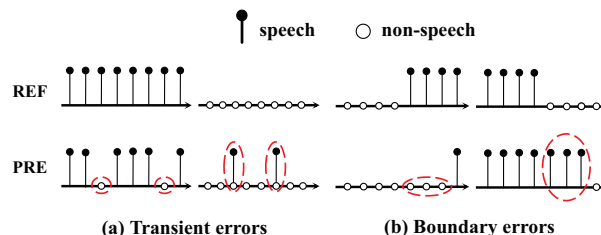


Figure 1: Two typical difficult-to-handle errors in VAD or OSD tasks, where REF and PRE represent the sequence of binary labels and the sequence of predictions, respectively.

the meeting context), the speech signals are often recorded with a distant microphone (or microphone array). In [24], Cornell *et al.* first proposed a TCN-based architecture optimized with cross-entropy loss for distant VAD+OSD, which uses acoustic features (e.g. Mel Frequency Cepstral Coefficients (MFCCs)) as model input. This work was extended in [25] by extracting interaural phase differences (IPDs) and cosine-sine inter-channel phase differences (CSIPDs) spatial features from multi-microphone signals. Mariotte *et al.* also [26] proposed a set of spatial features based on direction-of-arrival estimates in the circular harmonic domain (CH-DOA) and showed that these spatial features were effective for VAD and OSD tasks. The above methods usually focused on the design of spatial features and adopted the same TCN-based architecture optimized with the cross-entropy loss function. *However, as the cross-entropy loss is constructed in a frame-independent manner, which may hinder the model to handle frame-related errors.* By implementing these models, we confirmed that the cross-entropy loss was prone to transient errors caused by data disturbances in Figure 1 (a) or boundary errors caused by weak recordings at speech boundaries in Figure 1 (b).

To mitigate these issues, we formulate two losses, named as smoothed loss and weighted loss, to jointly optimize the TCN model in this work. The smoothed loss penalizes transient errors by monitoring the probability changes between consecutive frames, outputting smooth TCN predictions. The weighted loss allows the TCN to focus more on speech-noise boundaries to reduce the boundary errors. Besides, we propose a lightweight spatial feature based on the Instantaneous Correlation Coefficient with Frequency Selection (ICCFs), by selecting those informative frequencies that contain the spatial cue of sound sources. In order to further reduce the computational load, inspired by [25], we adopt only a few pairs of microphones instead of all possible pairs. The effectiveness of the proposed loss functions and spatial features are demonstrated on AMI meeting corpus [27].

## 2. Problem Formulation

VAD and OSD can be formulated together as a three-category classification task: non-speech ( $n_{spk} = 0$ ), single speech ( $n_{spk} = 1$ ), and overlapped speech ( $n_{spk} \geq 2$ ) with  $n_{spk}$  being the number of active speakers. Let  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T] \in \mathbb{R}^{E \times T}$  be a sequence of characteristic vectors, where  $t$  is the time frame index,  $T$  and  $E$  are the numbers of frames and features, respectively. In addition, the associated sequence of one-hot labels is defined as  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_t, \dots, \mathbf{y}_T] \in \mathbb{R}^{3 \times T}$ . Then, the VAD+OSD can be implemented by finding a mapping  $f(\mathbf{X})$  to output the prediction sequence  $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_t, \dots, \hat{\mathbf{y}}_T] \in \mathbb{R}^{3 \times T}$ , where  $\hat{\mathbf{y}}_t = [p(n_{spk} = 0|\mathbf{x}_t), p(n_{spk} = 1|\mathbf{x}_t), p(n_{spk} \geq 2|\mathbf{x}_t)]^T$  denotes the probabilities of each class in the  $t$ -th frame. Based on above, the final outputs of VAD and OSD tasks at frame  $t$  can be formulated by

$$\text{VAD} = \begin{cases} 1, & p(n_{spk} \geq 1|\mathbf{x}_t) \geq \epsilon \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

$$\text{OSD} = \begin{cases} 1, & p(n_{spk} \geq 2|\mathbf{x}_t) \geq \epsilon \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $\epsilon$  is the threshold, which is often set to 0.5 empirically.

## 3. Proposed Method

### 3.1. Architecture

We adopt the network architecture depicted in Figure 2. The feature matrix  $\mathbf{X}$  is obtained by concatenating the spatial feature  $\mathbf{S}$  and the acoustic feature  $\mathbf{A}$ , which is fed to the TCN [24] to obtain  $\hat{\mathbf{Y}}$ . This architecture is similar to recent work [26], but the difference lies in spatial feature extraction and loss function.

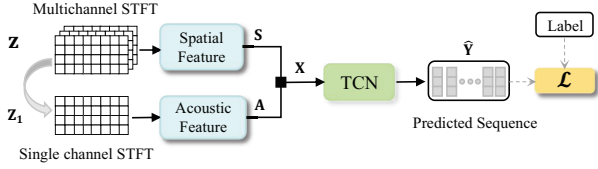


Figure 2: Overall architecture of VAD+OSD. ■ represents the concatenation operation.

Let  $\mathbf{Z}_m \in \mathbb{C}^{F \times T}$  be the STFT spectrogram of the  $m$ -th microphone, where  $F$  represents the number of frequency bins. In [26], the acoustic feature  $\mathbf{A} \in \mathbb{R}^{E_a \times T}$  is obtained by extracting the  $E_a \times T$  dimensional MFCCs feature from  $\mathbf{Z}_1$ , and the spatial feature  $\mathbf{S} \in \mathbb{R}^{E_s \times T}$  is determined by extracting the  $E_s \times T$  dimensional CH-DOA feature from  $\mathbf{Z} = \text{concat}(\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_M) \in \mathbb{C}^{M \times F \times T}$  with  $\text{concat}(\cdot)$  denoting the concatenation operator. Besides, the TCN is optimized by minimizing cross-entropy loss  $\mathcal{L}_{ce}$  [24, 25, 26]

$$\mathcal{L}_{ce} = -\frac{1}{T} \sum_{t=1}^T \sum_{i=1}^3 y_{t,i} \log(\hat{y}_{t,i}), \quad (3)$$

where  $y_{t,i}$  and  $\hat{y}_{t,i}$  are the  $i$ -th elements of  $\mathbf{y}_t$  and  $\hat{\mathbf{y}}_t$ , respectively.

As (3) is constructed in a frame-independent manner, it may restrict the model to handle transient noise and weak speech boundaries as illustrated in Figure 1. Moreover, the spatial features are not lightweight. To address these issues, we formulate two novel loss functions and design a lightweight spatial feature extraction module in the next section.

### 3.2. Loss Function

#### 3.2.1. Smoothed Loss

The transition of speech states (non-, single, and overlapped speech) is often gradual and does not occur frequently within a short time interval. In practice, however, the model may generate incoherent predictions due to interference factors (e.g. data instability or noise), leading to transient errors in Figure 1 (a).

To alleviate the adverse effect of transient errors, we improve the continuity of predictions by introducing a smoothed loss function  $\mathcal{L}_s$ . Specifically, it penalizes fluctuations in predicted state probabilities among consecutive frames [28], with larger fluctuations leading to higher penalties, which can be formulated as

$$\mathcal{L}_s = \frac{1}{TC} \sum_t \sum_i \delta_{t,i}^2, \quad (4)$$

$$\delta_{t,i} = \begin{cases} |\log \hat{y}_{t,i} - \log \hat{y}_{t-1,i}|, & |\log \hat{y}_{t,i} - \log \hat{y}_{t-1,i}| \leq \tau \\ \tau, & \text{otherwise,} \end{cases} \quad (5)$$

where  $C = 3$  is the number of classes and  $\tau$  is the threshold to truncate the smoothed loss, preventing excessive penalization when consecutive frames are confidently predicted to belong to different states.

#### 3.2.2. Weighted Loss

In general, speech signals are weak at the beginning or end of a speaker's utterance, which makes speech indistinguishable from noise. As a consequence, boundary errors are encountered in model predictions, as shown in Figure 1 (b). To mitigate this, we design a cross-entropy loss with frame-level weights, where the introduced weights increase near the speech and non-speech boundaries, thus encouraging the model to enhance accuracy at these boundaries.

Let  $\tilde{\mathbf{y}} = [\tilde{y}_1, \dots, \tilde{y}_t, \dots, \tilde{y}_T] \in \mathbb{R}^T$  denote a sequence of binary class labels corresponding to (1), where  $\tilde{y}_t \in \{0, 1\}$  with  $\tilde{y}_t = 1$  indicating that frame  $t$  contains speech and vice versa. Note that  $\tilde{y}_t$  can be obtained by summing the last two class labels in  $\hat{\mathbf{y}}_t$ . Given  $\tilde{y}_t$ , we define the left label sequence  $\tilde{\mathbf{y}}_t^l$  and the right label sequence  $\tilde{\mathbf{y}}_t^r$  as follows

$$\tilde{\mathbf{y}}_t^l = [\tilde{y}_{t-\mu}, \tilde{y}_{t-\mu+1}, \dots, \tilde{y}_{t-1}], \quad (6)$$

$$\tilde{\mathbf{y}}_t^r = [\tilde{y}_{t+1}, \tilde{y}_{t+2}, \dots, \tilde{y}_{t+\mu}], \quad (7)$$

where  $\mu$  is the context length. Then, the frame-level weight  $w_t$  is introduced for time  $t$ , which can be computed as

$$w_t = \alpha \cdot \log(s_t + 1) + 1, \quad (8)$$

$$s_t = \sum_{n=1}^{\mu} (\tilde{y}_{t,n}^l \oplus \tilde{y}_{t,n}^r), \quad (9)$$

where  $\alpha$  is the scaling factor,  $\oplus$  is the XOR operation,  $\tilde{y}_{t,n}^l$  and  $\tilde{y}_{t,n}^r$  represent the  $n$ -th elements of  $\tilde{\mathbf{y}}_t^l$  and  $\tilde{\mathbf{y}}_t^r$ , respectively, and  $s_t$  represents the degree of deviation between  $\tilde{y}_{t,n}^l$  and  $\tilde{y}_{t,n}^r$  at time  $t$ . In (8),  $w_t$  takes a larger value if frame  $t$  lies at the speech-noise boundaries. This is because the closer the  $t$ -th frame is to the speech-noise boundary, the larger the returned value of  $\tilde{y}_{t,n}^l \oplus \tilde{y}_{t,n}^r$ . As a special case,  $\tilde{y}_{t,n}^l \oplus \tilde{y}_{t,n}^r$  returns the largest value if frame  $t$  is exactly the speech-noise boundary. Finally, the weighted loss  $\mathcal{L}_w$  is defined by

$$\mathcal{L}_w = -\frac{1}{T} \sum_{t=1}^T \sum_{i=1}^C w_t \cdot y_{t,i} \log(\hat{y}_{t,i}), \quad (10)$$

through which the boundary errors can be suppressed efficiently.

### 3.2.3. Total Loss

Based on above, we construct a total loss function  $\mathcal{L}_{sw}$ , which combines the smoothed loss (4) and the weighted loss (10), i.e.

$$\mathcal{L}_{sw} = \mathcal{L}_w + \lambda \mathcal{L}_s, \quad (11)$$

where  $\lambda \geq 0$  is a model hyper-parameter for determining the weight of  $\mathcal{L}_s$ .

### 3.3. Spatial Feature

The instantaneous correlation coefficient (ICC) is an important spatial feature that includes the position information of sound sources. To be specific,  $\text{ICC}_{i,j}(f, t)$  of the channels  $i$  and  $j$  is given by

$$\text{ICC}_{i,j}(f, t) = Z_i(f, t) \cdot Z_j^*(f, t), \quad (12)$$

where  $f$  is frequency index, the superscript  $*$  is the complex-conjugate operator, and  $Z_i(f, t)$  is the  $(f, t)$ -th element of  $\mathbf{Z}_i$ . The  $\text{ICC}_{i,j}(t)$  feature vector at time  $t$  is defined as

$$\text{ICC}_{i,j}(t) = [\text{ICC}_{i,j}(1, t), \dots, \text{ICC}_{i,j}(F, t)]. \quad (13)$$

Obviously, the computational complexity of ICC feature depends on the value of  $F$ , and a larger  $F$  can result in a large amount of computations, especially for a large-scale microphone array. Therefore, we select only  $k$  ( $k \ll F$ ) based on

$$\{\tilde{f}_1, \dots, \tilde{f}_k\} = \arg \max_f^{(k)} (|Z_1(f, t)|), \quad (14)$$

where  $|\cdot|$  denotes the modulus operation, thus (14) selects the frequencies with  $k$  largest amplitudes in  $Z_1(f, t)$ , and  $\tilde{f}_k$  represents the index of selected frequencies. As the amplitudes of all microphone signals are same in a far-field scenario, we utilize only the first channel in (14). Besides, the frequency with lower amplitudes may not include speech information, which in turn brings adverse impact on VAD or OSD tasks (as we shall see in 4.3). For this reason, we establish (14) based on frequency amplitudes.

Instantaneous correlation coefficient with frequency selection  $\text{ICCFS}_{i,j}(t)$  for channels  $i$  and  $j$  can be constructed as

$$\text{ICCFS}_{i,j}(t) = [\text{ICC}_{i,j}(\tilde{f}_1, t), \dots, \text{ICC}_{i,j}(\tilde{f}_k, t)]. \quad (15)$$

To further reduce the computational complexity, inspired by [25], we adopt only a few pairs of microphones by selecting appropriate  $i$  and  $j$ . Finally, the complex-valued vector  $\text{ICCFS}_{i,j}(t)$  is converted to real-valued ones by concatenating real and imaginary parts to form the spatial feature.

## 4. Experiments

### 4.1. Dataset

Experiments were conducted on the AMI meeting corpus [27], which contains 100 hours of realistic meeting recordings. As audio signals were captured using different devices, we adopted the AMI *Array1* data that was captured by an 8-microphone circular array placed in the center of the table. Training, Development (Dev), and Evaluation (Eval) partitions followed the protocol proposed in [29]. During the training phase, ground truth was used via Forced-Alignment (FA) [24]. The results on the Dev and Eval sets were evaluated using the official annotation. Speech signals were sampled at 16kHz.

### 4.2. Experimental setup

Acoustic features were extracted from the signal captured by the first channel of *Array1*. We used MFCCs as the acoustic feature with  $E_a = 80$ , extracted from frames with length 25 ms and with 60% overlap. For spatial features, we chose only 4 pairs of opposing microphones from *Array1* instead of using all the 28 possible pairs. Additionally, to further save computational resources, we selected only  $k = 5$  frequencies, resulting in a frame-level feature dimension of  $E_s = 5 \times 2 \times 4 = 40$ .

TCN consists of 1D convolutional layers with exponentially increasing dilation rates to capture long-range temporal dependencies efficiently [24]. We utilized the TCN structure that consists of  $R = 3$  residual convolutional blocks replicated  $P = 3$  times. A 1-dimensional convolutional layer followed by a *softmax* activation function was added after the TCN to generate the final classification probabilities.

The TCN was trained using 5 seconds of audio segments randomly sampled from the training set. We used ADAM [30] with a mini-batch size of 32. The weight decay was 0.0001 and the momentum was 0.9. The base learning rate was 0.001 and we divided it by 10 every 10 epochs. All experiments were executed on an NVIDIA L40S GPU. In the following, we used average precision (AP) and F1 score to evaluate VAD and OSD, and reported the results on Dev and Eval sets.

### 4.3. Comparison of different spatial features

Table 1 presents the comparison results of different spatial features. For the sake of fairness, these spatial features are combined with the same acoustic feature, i.e., MFCCs, when using the architecture in Figure 2. The TCN is optimized by minimizing the original cross-entropy loss (3) in [25, 26]. We can find that the proposed spatial feature ICCFS achieves the best F1 score on VAD. For OSD, the F1 score of ICCFS is slightly lower than IPD on the Dev set, but ICCFS is more lightweight. This advantage is due to the fact that effective frequencies are selected by our strategy in (14). It can exclude the frequency with lower amplitudes, which may hinder the model to output accurate VAD and OSD predictions. In subsection 4.5.3, we further discuss the impact of selecting different numbers of frequencies on VAD and OSD performance.

Table 1: *F1 score (%) performance of different spatial features. Param.(M) is the number of model's parameters. Bold values indicate the best-performing results. † denotes re-implementation under our experimental setup.*

Spatial Feature	Param.(M)	VAD		OSD	
		Dev	Eval	Dev	Eval
IPD <sup>†</sup>	0.217	94.37	94.85	<b>71.83</b>	66.64
CSIPD <sup>†</sup>	0.270	94.60	94.94	70.39	66.49
CH-DOA <sup>†</sup>	0.190	94.63	95.04	70.72	66.36
ICCFS (Ours)	<b>0.167</b>	<b>94.73</b>	<b>95.09</b>	71.64	<b>67.06</b>

### 4.4. Comparison of TCN-based methods with different loss functions

To study the effectiveness of the proposed  $\mathcal{L}_{sw}$  in (11), we replaced the original cross-entropy loss  $\mathcal{L}_{ce}$  in recent distant VAD and OSD methods [25, 26] with the proposed  $\mathcal{L}_{sw}$ . Detailed results were presented in Table 2. It is evident that the AP performance of both VAD and OSD tasks significantly improves across all methods by replacing  $\mathcal{L}_{ce}$  with  $\mathcal{L}_{sw}$ . TCN-IPD [25] shows the most significant improvements, with a 5.53% in-

crease on the Dev set and a 5.64% increase on the Eval set for the OSD task. Moreover, on the VAD task, our method with  $\mathcal{L}_{sw}$  achieves comparable performance to other methods with  $\mathcal{L}_{sw}$ . On the OSD task, our method achieves the best performance on the Eval set (71.39%) but performs slightly worse than TCN-CSIPD on the Dev set. However, as discussed in 4.3, our model has only 0.167 M parameters, which is significantly fewer than the parameters of 0.27 M in TCN-CSIPD.

Table 2: AP (%) performance of different methods by using the proposed loss  $\mathcal{L}_{sw}$  and the cross-entropy loss  $\mathcal{L}_{ce}$ .

Method	Loss	VAD		OSD	
		Dev	Eval	Dev	Eval
TCN-IPD[25] <sup>†</sup>	$\mathcal{L}_{ce}$	94.44	97.14	69.94	65.18
	$\mathcal{L}_{sw}$	<b>95.96</b>	<b>98.24</b>	<b>75.47</b>	<b>70.82</b>
TCN-CSIPD[25] <sup>†</sup>	$\mathcal{L}_{ce}$	94.72	97.10	73.55	69.17
	$\mathcal{L}_{sw}$	<b>95.93</b>	<b>98.26</b>	<b>75.81</b>	<b>70.64</b>
TCN-CHDOA[26] <sup>†</sup>	$\mathcal{L}_{ce}$	95.44	97.10	71.73	67.85
	$\mathcal{L}_{sw}$	<b>95.98</b>	<b>98.25</b>	<b>75.80</b>	<b>70.81</b>
TCN-ICCFS (Ours)	$\mathcal{L}_{ce}$	95.75	97.62	74.39	70.20
	$\mathcal{L}_{sw}$	<b>95.85</b>	<b>98.25</b>	<b>75.52</b>	<b>71.39</b>

#### 4.5. Impact of the hyper-parameters

The proposed loss function and ICCFS features are controlled by some hyper-parameters. Here, we conducted ablation experiments on the Eval set to explore the impact of hyper-parameters.

##### 4.5.1. Impact of $\alpha$ and $\mu$

The proposed weighted loss  $\mathcal{L}_w$  in (10) is controlled by two hyper-parameters, i.e., the scaling factor  $\alpha$  and the context length  $\mu$ . Their influence on VAD and OSD is tested in this experiment. From Table 3 we can see that VAD accuracy increases with increasing  $\mu$ , and  $\mathcal{L}_w$  outperforms cross-entropy  $\mathcal{L}_{ce}$  when  $\mu=20$  or  $\mu=40$ . However, OSD performance decreases slightly as  $\mu$  increases. This may be due to the fact that  $\mathcal{L}_w$  only considers the boundary between speech and non-speech, while neglecting the boundary between single speech and overlapping speech. Then we tested the impact of  $\alpha$  after setting  $\mu = 20$  to achieve a trade-off between VAD and OSD performance. As shown in Table 3, the smaller the value of  $\alpha$ , the better the performance of both VAD and OSD tasks. The main reason is that the boundary frames are overemphasized when  $\alpha$  is too high, neglecting the importance of other frames. Based on the above analysis, we set  $\mu=20$  and  $\alpha=0.1$  in the following experiments.

Table 3: Impact of  $\mu$  and  $\alpha$  on AP performance.

Loss	VAD	OSD
$\mathcal{L}_{ce}$	97.62	<b>70.20</b>
$\mathcal{L}_w$ ( $\mu=10, \alpha=0.1$ )	96.77	<b>69.78</b>
$\mathcal{L}_w$ ( $\mu=20, \alpha=0.1$ )	97.66	69.50
$\mathcal{L}_w$ ( $\mu=40, \alpha=0.1$ )	<b>98.04</b>	68.72
$\mathcal{L}_w$ ( $\alpha=0.1, \mu=20$ )	<b>97.66</b>	<b>69.50</b>
$\mathcal{L}_w$ ( $\alpha=0.3, \mu=20$ )	97.32	69.36
$\mathcal{L}_w$ ( $\alpha=0.5, \mu=20$ )	97.04	68.98

##### 4.5.2. Impact of $\lambda$ and $\tau$

The hyper-parameter  $\lambda$  controls the weight of the smoothed loss  $\mathcal{L}_s$  in the total loss  $\mathcal{L}$ , while the hyper-parameter  $\tau$  determines

the threshold in  $\mathcal{L}_s$ . As given in Table 4, using both  $\mathcal{L}_w$  and  $\mathcal{L}_s$  can achieve better VAD and OSD performance than using only  $\mathcal{L}_w$  under all different settings of  $\lambda$  or  $\tau$ . This illustrates the effectiveness of  $\mathcal{L}_s$ . In addition, the best performance of both VAD and OSD tasks is obtained when  $\lambda = 0.25$  and  $\tau = 4$ . A larger  $\lambda$  ( $\tau$ ) or a smaller  $\lambda$  ( $\tau$ ) causes partial performance degradation. This may be due to the fact that a smaller  $\lambda$  ( $\tau$ ) may not sufficiently penalize transient errors, while a larger  $\lambda$  ( $\tau$ ) overemphasizes continuity, neglecting the accurate identification of actual speech boundaries.

Table 4: Impact of  $\lambda$  and  $\tau$  on AP performance.

Loss	VAD	OSD
$\mathcal{L}_w$	97.66	69.50
$\mathcal{L}_w + \mathcal{L}_s$ ( $\lambda=0.05, \tau=4$ )	98.14	71.15
$\mathcal{L}_w + \mathcal{L}_s$ ( $\lambda=0.25, \tau=4$ )	<b>98.25</b>	<b>71.39</b>
$\mathcal{L}_w + \mathcal{L}_s$ ( $\lambda=0.65, \tau=4$ )	98.06	71.14
$\mathcal{L}_w + \mathcal{L}_s$ ( $\tau=2, \lambda=0.25$ )	98.23	71.05
$\mathcal{L}_w + \mathcal{L}_s$ ( $\tau=4, \lambda=0.25$ )	<b>98.25</b>	<b>71.39</b>
$\mathcal{L}_w + \mathcal{L}_s$ ( $\tau=6, \lambda=0.25$ )	98.21	70.93

##### 4.5.3. Impact of $k$

To investigate the impact of  $k$  selected frequencies of the spatial feature ICCFS on the performance of VAD and OSD tasks, we conducted experiments with different values of  $k$ . As shown in Figure 3, we can observe that the quantity of parameters rises significantly as  $k$  increases, while VAD or OSD performance is nearly unchanged (or even decreased) with increasing  $k$ . Particularly, OSD performance declines noticeably when  $k = 150$ , which indicates that the frequencies with lower amplitudes may not provide positive effects on VAD and OSD tasks. This experiment further validates the necessity of frequency selection in spatial feature extraction.

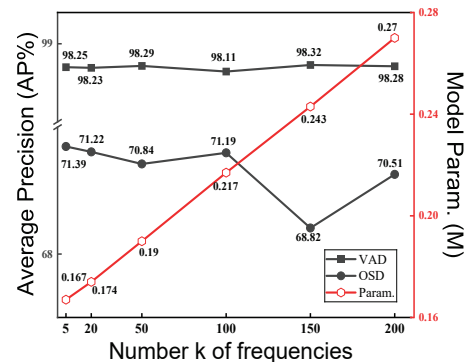


Figure 3: Effect of the number of frequencies on VAD and OSD tasks.

## 5. Conclusion

In this work, we formulated a weighted loss and a smoothed loss for optimizing TCN on VAD and OSD tasks. The former focuses on boundary error correction while the latter focuses on transient error correction. The proposed loss can further improve the performance of recent TCN-based methods. Moreover, we constructed a lightweight spatial feature module based on the Instantaneous Correlation Coefficient with Frequency Selection (ICCFS). Experiments confirmed the effectiveness of the proposed loss as well as the ICCFS features.

## 6. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 62201297 and 62361045.

## 7. References

- [1] T. Sun, T. Lei, X. Zhang, Y. Hu, C. Zhu, and J. Lu, "A lightweight hybrid multi-channel speech extraction system with directional voice activity detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 1486–1490.
- [2] Q. Yang, Q. Liu, N. Li, M. Ge, Z. Song, and H. Li, "Svad: A robust, low-power, and light-weight voice activity detection with spiking neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 221–225.
- [3] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2008, pp. 4353–4356.
- [4] K. Boakye, O. Vinyals, and G. Friedland, "Two's a crowd: Improving speaker diarization by automatically identifying and excluding overlapped speech," in *Proc. Interspeech*, 2008, pp. 32–35.
- [5] J. Han, F. Landini, J. Rohdin, M. Diez, L. Burget, Y. Cao, H. Lu, and J. Černocký, "Diacorrect: Error correction back-end for speaker diarization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 11 181–11 185.
- [6] M. Rybicka, J. Villalba, T. Thebaud, N. Dehak, and K. Kowalczyk, "End-to-end neural speaker diarization with non-autoregressive attractors," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 3960–3973, 2024.
- [7] F. Hao, X. Li, and C. Zheng, "End-to-end neural speaker diarization with an iterative adaptive attractor estimation," *Neural Netw.*, vol. 166, pp. 566–578, 2023.
- [8] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Inf. Fusion*, vol. 109, p. 102422, 2024.
- [9] A. Saif, X. Cui, H. Shen, S. Lu, B. Kingsbury, and T. Chen, "Joint unsupervised and supervised training for automatic speech recognition via bilevel optimization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal*, 2024, pp. 10 931–10 935.
- [10] K. Radha, M. Bansal, and R. B. Pachori, "Speech and speaker recognition using raw waveform modeling for adult and children's speech: A comprehensive review," *Eng. Appl. Artif. Intell.*, vol. 131, p. 107661, 2024.
- [11] S. Wang, Z. Chen, K. A. Lee, Y. Qian, and H. Li, "Overview of speaker modeling and its applications: From the lens of deep speaker representation learning," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2024.
- [12] R. Chengalvarayan, "Robust energy normalization using speech/nonspeech discriminator for german connected digit recognition," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 99, 1999, pp. 61–64.
- [13] H. Ghaemmaghami, B. Baker, R. Vogt, and S. Sridharan, "Noise robust voice activity detection using features extracted from the time-domain autocorrelation function," in *Proc. Interspeech*, 2010, pp. 3118–3121.
- [14] 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, 2001.
- [15] 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, 1999.
- [16] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselý, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, vol. 9, 2012, pp. 1–4.
- [17] R. Sarikaya and J. H. Hansen, "Robust detection of speech activity in the presence of noise," in *Proc. ICSLP*, 1998, pp. 1455–8.
- [18] S. Thomas, S. Ganapathy, G. Saon, and H. Soltan, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 2519–2523.
- [19] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks," in *Proc. Interspeech*, 2013, pp. 728–731.
- [20] M. Lavechin, M.-P. Gill, R. Bousbib, H. Bredin, and L. P. Garcia-Perera, "End-to-end domain-adversarial voice activity detection," *Proc. Interspeech*, 2020.
- [21] J. T. Geiger, F. Eyben, B. Schuller, and G. Rigoll, "Detecting overlapping speech with long short-term memory recurrent neural networks," in *Proc. Interspeech*, 2013.
- [22] J.-w. Jung, H.-S. Heo, Y. Kwon, J. S. Chung, and B.-J. Lee, "Three-class overlapped speech detection using a convolutional recurrent neural network," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3086–3090.
- [23] M. Kunešová, M. Hruš, Z. Zajíc, and V. Radová, "Detection of overlapping speech for the purposes of speaker diarization," in *Proc. Int. Conf. Speech Comput.*, 2019, pp. 247–257.
- [24] S. Cornell, M. Omologo, S. Squartini, and Vincent, "Detecting and counting overlapping speakers in distant speech scenarios," in *Proc. Interspeech*, 2020.
- [25] S. Cornell, M. Omologo, S. Squartini, and E. Vincent, "Overlapped speech detection and speaker counting using distant microphone arrays," *Comput. Speech Lang.*, vol. 72, p. 101306, 2022.
- [26] T. Mariotte, A. Larcher, S. Montrésor, and J.-H. Thomas, "Multi-microphone automatic speech segmentation in meetings based on circular harmonics features," in *Proc. Interspeech*, 2023, pp. 2783–2787.
- [27] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, and M. Kronenthal, "The AMI meeting corpus: A pre-announcement," in *Proc. Int. Workshop Mach. Learn. Multimodal Interact.*, 2005.
- [28] Y. A. Farha and J. Gall, "MS-TCN: Multi-stage temporal convolutional network for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3575–3584.
- [29] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (VBX) in speaker diarization: Theory, implementation and analysis on standard tasks," *Comput. Speech Lang.*, vol. 71, p. 101254, 2022.
- [30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.