IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 May 2023, accepted 7 June 2023, date of publication 14 June 2023, date of current version 21 June 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3286034*

RESEARCH ARTICLE

# Analysis-Based Optimization of Temporal Dynamic Convolutional Neural Network for Text-Independent Speaker Verification

**SEONG-HU KIM, HYEONUK NAM, AND YONG-HWA PARK, (Member, IEEE)**
Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea

Corresponding author: Yong-Hwa Park (yhpark@kaist.ac.kr)

This work was supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and the Ministry of Trade, Industry and Energy of Korean Government under Grant UM22409RD4.

**ABSTRACT** Temporal dynamic convolution neural networks (TDY-CNNs) extract speaker embeddings considering the time-varying characteristics of speech and improve text-independent speaker verification performance. In this paper, we optimize TDY-CNNs based on the detailed analysis of the network architecture. The temporal dynamic convolution generates attention weight of basis kernels from features defined by concatenating average channel and frequency data, resulting in a reduction in network parameters by 26%. In addition, the temporal dynamic convolutions replace vanilla convolutions in earlier layers, while the optimized temporal dynamic convolutions of latter layers use a steady kernel regardless of time bin data. As a result, Opt-TDY-ResNet-34(×0.50) shows the best speaker verification performance with EER of 1.07% among speaker verification models without data augmentation including ResNet-based baseline networks and other state-of-the-art networks. Moreover, we validate that Opt-TDY-CNNs adapt to time-bin data through various methods. By comparing the inter and intra phoneme distance of attention weights, it was confirmed that the temporal dynamic convolution uses different kernels depending on the phoneme groups directly related to the time-bin data. In addition, by applying gradient-weighted class activation mapping (Grad-CAM) on speaker verification to obtain speaker activation map (SAM), we showed that temporal dynamic convolution extracts speaker information from frequency characteristics of time bins such as phonemes' formant frequencies while vanilla convolution extracts vague outline of Mel-spectrogram.

**INDEX TERMS** Speaker verification, text-independent, temporal dynamic convolution, temporal data-dependent kernel.

## I. INTRODUCTION

Speaker verification aims to verify whether a test utterance is spoken by the enrolled speaker whose utterances were pre-recorded. This research topic has been developed for various applications such as biometric authentication, forensics, security, and speaker diarization, etc. Recently, speaker representing vectors also known as speaker embeddings have been extracted using deep neural networks (DNN), and DNN-based speaker embedding framework has become a dominant approach to perform speaker verification. One example of speaker embedding is x-vector [1], which is extracted using the time-delay neural network (TDNN) [2], [3]. From then, various methods such as F-TDNN [4], E-TDNN [5], and ECAPA-TDNN [6] have improved the TDNN based speaker verification. In addition, neural networks based on VGG [7] and ResNet [8], which are originally proposed for image recognition in computer vision, were also applied to speaker verification [9], [10], [11] using 2D audio data formant such as spectrogram, Mel-spectrogram, and MFCCs. Besides the works on improving neural network architecture, there have been other approaches on DNN training methods such as metric learning and adversarial training are applied to extract

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Crippa.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

speaker embeddings with small intra-class distances and large inter-class distance in various environments [12].

Among previous studies on speaker verification, only few studies delved into a comprehensive understanding of speaker embedding networks while most studies merely focused on improving the verification performance. However, such approach to deepen the understanding of deep-learning based speaker verification is essential in order to effectively improve speaker verification performance and actually apply-and-assess them in real life uses. One such study observed the characteristics of frame-level speaker embeddings by moving the pooling layer to the back end of the neural network [13]. The results showed that frame-level embeddings of vowels and nasals have sufficient information to discriminate speakers, while fricatives and stops are not useful to discriminate speakers as they show high similarity scores even when they came from different speakers. Other studies have also shown that vowels and nasals have major influences on speaker verification while other phonemes are less useful [14], [15]. By the way, other phonemes such as stops, fricatives and affricates are generated from speakers' vocal tracts exhibiting different acoustic characteristics. Therefore, they also are acoustically affected by speakers' unique vocal tracts thus should exhibit speakers' biometric information. From the previous studies [13], [14], [15], it can be inferred that most previous speaker verification models are biasedly trained to recognize only speaker information from vowels and nasals. It is a natural consequence considering that extracting speaker information from the temporal portion of phonemes having the most distinct speaker information is easier than extracting speaker information from the entire time range. However, such training methods could result in speaker verification model overfitted to few specific phonemes, which is not robust enough to be used in real circumstances. As other phonemes do have sufficient speaker information also, speaker verification models' performance and robustness would be improved if we could train them to extract speaker information from more phonemes whether they are voiced or not.

In the text-independent speaker verification task, similar speaker information should be extracted from various utterances composed of different phoneme configurations given the utterances are form the same speaker. However, phonemes are generated through different pathways and mechanisms from speakers' vocal tract, so acoustic characteristics differ from each other. Thus, *phonetic variability* arising from random text presents a major challenge in achieving accuracy in text-independent speaker recognition [16], and speaker embedding extraction algorithms to consider phoneme diversity have been proposed. Before the development of DNN-based models, segmentation of speech signals into broad phonetic classes was used as a pre-processing step, and speaker models were structured phonetically [17], [18], [19], [20]. Gaussian mixture models were used to model various speaker features that depend on phonetic variability [21], [22]. DNN-based models have dramatically improved text-independent speaker verification performance by proposing a segment-level training approach rather than frame-level containing phonetic variability [1]. Considering that previous DNN-based speaker verification methods apply convolution which extract speaker information using fixed trained kernel [13], [14], [15], single static kernel on each convolution layer would likely to result in extracting information from specific phoneme groups only: vowels and nasals. It is almost impossible to obtain single convolution kernel capable of extracting speaker information from different phonemes having varied acoustic characteristics. To overcome such limitation, we proposed an adaptive convolutional neural network (ACNN) in which the kernels adaptively change along time segments of time-frequency domain input [23]. ACNN was applied to baseline models and improve speaker recognition performance compared to the conventional CNN-based models. This study proved that a temporal-data dependent network such as ACNN is suitable to consider different text in text-independent speaker recognition tasks. Based on this study, we proposed a temporal dynamic convolutional neural network (TDY-CNN), a generalized network of ACNN [24]. TDY-CNN uses kernels extracting speaker information adapts to time bins rather than time segments. TDY-CNN improved text-independent speaker verification performance by effectively capturing time-varying speech information arising from phonemes varying over time. However, there are two drawbacks of TDY-CNN: large network parameters and insufficient analysis of kernels. First drawback is that TDY-CNNs have approximately 9 times larger network parameters compared to conventional CNN-based models. This can lead to overfitting and instability of training, preventing the networks from achieving the best performance. Second drawback is that the previous study showed the relationship between kernels and phoneme groups through several example utterances. This analysis does not generalize the hypothesis that TDY-CNNs extract speaker information considering the time-varying information such as phonemes in text-independent situation. In addition, it is unknown whether the kernels depending on the time bins data are suitable for extracting speaker information. Thus, it is necessary to improve the network structure and verify the hypothesis through new analysis techniques.

In this paper, we optimized the structure of temporal dynamic convolutional neural networks (TDY-CNNs) and named the resulting architecture Opt-TDY-CNNs. Based on the analysis and case studies on the utilization of channel and frequency data in time bins, we determined that the attention weights of kernels were generated by average of channel and frequency data. In addition, through the analysis of kernel variation with different layers, optimized temporal dynamic convolutions are only applied to the earlier layers where phoneme's acoustic information is dominant in the network. Moreover, we analyzed how Opt-TDY-CNNs operate on different phonemes without providing phoneme information during training using intra/inter-phoneme distance of attention weights and gradient-weighted class activation mapping (Grad-CAM) [25].

The remainder of the paper is organized as follows. Section II explains related works on speaker recognition models considering phoneme information and input-adaptive networks. Section III introduces the structure of optimized temporal dynamic convolutional neural network (Opt-TDY-CNN) and its differences from conventional TDY-CNN. Section IV describes experimental setup and details, and section V shows the experiment results and discussion. Lastly, Section VI presents conclusions.

## II. RELATED WORKS

### A. EXTRACTION OF SPEAKER EMBEDDING CONSIDERING PHONETIC INFORMATION

Utterances are composed of rapidly varying phonemes, those having different acoustic characteristics due to different generation mechanisms. Therefore, it is important to extract speaker information from varying acoustic characteristics of phonemes in text-independent speaker verification. Consequently, there have been studies proposed to extract speaker embeddings considering phonetic information have been proposed. These studies can be broadly categorized into two categories: multi-task learning and domain-adversarial learning.

Multi-task learning [26], [27] aims to improve regularization and performance by leveraging domain-specific information contained in the data of inter-related tasks. In speaker verification task, phonetic information has been considered as domain-specific information. Speaker embedding extraction layers of speaker verification model are shared with related tasks, and models are trained simultaneously by minimizing criteria for speaker classification and related tasks. By incorporating tasks those discriminates text phrases or phonemes into speaker verification as multi-task learning, text-independent as well as text-dependent speaker verification performance have been improved [28], [29], [30], [31].

Similarly, domain-adversarial learning [32] is domain adaptation approach for predicting results by training neural networks to extract features excluding domain information of data, so that model cannot discriminate which domain (source or test) of input data is from. Gradient reversal layer (GRL) is added in front of the domain discriminating network, and model is trained to prevent domain discrimination. This idea is applied to text-independent speaker verification by replacing the domain-specific network with phoneme recognition network, so that extracted speaker embeddings would not include phoneme information [33], [34]. Such approaches successfully enhanced speaker verification performance.

Multi-task and domain-adversarial learning improve text-independent speaker verification performance without changing speaker embedding extraction network. These results indicate that the phoneme information has both necessary and unnecessary information for speaker verification. However, both learning methods cannot selectively utilize portions of phoneme information due to the models' structure and phoneme-related training criteria. On the other hand, TDY-CNN [24] can consider phoneme information naturally through temporal adaptive structure. In addition, TDY-CNN is trained using speaker verification criteria only without requiring additional phoneme information.

### B. DYNAMIC NEURAL NETWORK

Conventional deep neural networks have fixed structure and parameters regardless of input given to the network after training. So there are inevitable limitations in representation power, efficiency, and interpretability [35], [36]. Unlike static neural networks, dynamic neural networks adapt their network structures or parameters to the input contents. For example, early exiting [37], [38], [39] and skipping layers [40], [41], [42], [43], [44] alter network structure by changing networks' depth depending on the input. Also, there are networks those change width by selecting neurons [45], [46], [47] or branches [48], [49], [50], [51]. Unlike such networks with dynamic structure, networks with dynamic parameters have fixed network structures and change network parameters on the inputs instead. There are two main approaches to obtain networks with dynamic parameters: parameter generation and parameter adjustment. A straightforward way to adapt parameter on inputs is to generate parameters directly from the input [52], and such approaches on CNNs and RNNs have been proposed [53], [54], [55]. Directly generating network parameters increases computational cost exponentially as it requires many parameters to generate kernels with many channels. On the other hand, parameter adjustment methods adjust trained parameters adaptively, requiring relative less computation to adjust parameters. CNN-based dynamic networks with parameter adjustment perform soft attention on basis convolutional kernels for adaptive ensemble of parameters with minimal increase in computation [56], [57], [58], [59]. Although various dynamic neural networks have been proposed, these studies have been mainly conducted in computer vision [57], [60], [61] and natural language processing [62], [63].

Recently, various dynamic neural networks for speaker verification have been proposed [23], [64], [65]. Adaptive X-vector model [64] adjusts convolution parameters depending on the utterance by linear combination of trained convolution kernels and biases. Adaptive convolutional neural network (ACNN) [23] generates convolution kernels depending on short time segments using the time and frequency domain information of each time segment. In addition, global-local information-based dynamic convolution neural network (GLIDCNN) [65] generates convolution kernels depending on time-frequency regions using local and global features of utterance. These studies attempt to apply dynamic neural networks to text-independent speaker verification at the segment-level. On the other hand, acoustic characteristics

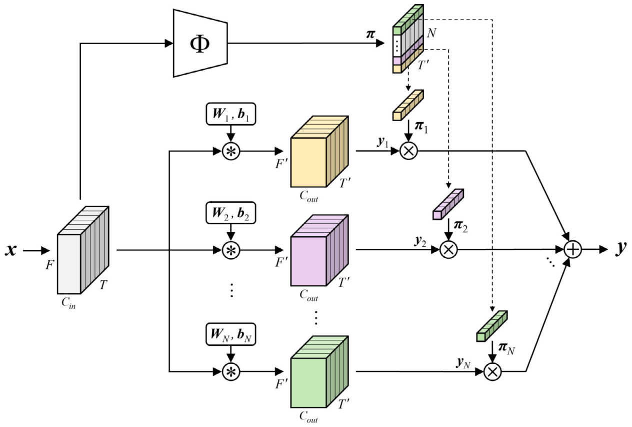


**FIGURE 1.** Structure of the temporal dynamic convolution for speaker verification. The temporal dynamic convolution result is derived as a weighted sum of $y_n$ which is the convolution result of $k$-th basis kernel.

of speech change over time, and a time bin is the minimum unit in time-frequency domain data. To fully consider random phoneme information changing over time, we proposed dynamic neural networks to text-independent speaker verification task at *time-bin-level* to consider speaker information in time-varying data [24].

## III. OPTIMIZED TEMPORAL DYNAMIC CONVOLUTIONAL NEURAL NETWORKS

### A. TEMPORAL DYNAMIC CONVOLUTION

The temporal dynamic convolution uses kernels adapted to time bins data. The kernel is obtained by weighted summation of basis kernels using attention weights, similar to the kernel adjustment mechanisms of conditionally parameterized convolution (CondConv) [58] and dynamic convolution [59]. However, unlike CondConv and dynamic convolution that use one kernel for one input, the temporal dynamic convolution uses different kernels for every single time bin of one input. However, kernel adjustment and convolution for every single time bin requires a lot of memory and computation. For computational efficiency of temporal dynamic convolution, we aggregated convolution results of basis kernels using attention weights as shown in Fig. 1. Given a time-frequency domain feature $\boldsymbol{x}\in\mathbb{R}^{C_{in}\times F\times T}$ as module input, the convolution result $\boldsymbol{y}_n\in\mathbb{R}^{C_{out}\times F'\times T'}$ of the $n$-th basis kernel $\boldsymbol{W}_n\in\mathbb{R}^{C_{out}\times C_{in}\times K\times K}$ and bias $\boldsymbol{b}_n\in\mathbb{R}^{C_{out}}$ is computed as

$$\begin{aligned}\boldsymbol{y}_n &= \boldsymbol{W}_n * \boldsymbol{x} + \boldsymbol{b}_n\\ s.t.\ n &= 1, 2, 3, \cdots, N,\end{aligned} \tag{1}$$

where $*$ denotes convolution. $C_{in}$ and $C_{out}$ are the numbers of input and output channels. $F$ and $F'$ are the number of frequency bins of input and output, respectively. $T$ and $T'$ are the number of time bins of input and output, respectively. $K$ is the (square) kernel shape, and $N$ is the number of basis kernels and biases. A total of $N$ convolution results $\boldsymbol{y}_n$ are aggregated using the attention matrix $\boldsymbol{\pi}_n \in \mathbb{R}^{N\times T'}$ as follows:

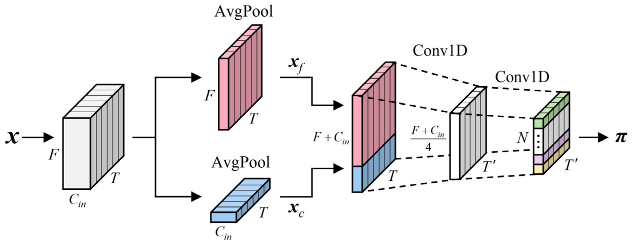


**FIGURE 2.** Structure of the attention weight generator Φ in the temporal dynamic convolution. The attention weight is derived from a feature generated by concatenating average of frequency and channel.

$$\boldsymbol{y} = \sigma\left(\sum_{n=1}^{N} \boldsymbol{y}_n \otimes \boldsymbol{\pi}_n\right), \tag{2}$$

where $\otimes$ denotes element-wise multiplication, $\sigma$ denotes the non-linear activation function ReLU, and $\boldsymbol{y}\in\mathbb{R}^{C_{out}\times F'\times T'}$ is output of temporal dynamic convolution. The attention matrix $\boldsymbol{\pi}_n$ represents $N$-dimensional attention weights those differ by time bins of time features $T'$. As different attentions are utilized on different time bins to derive the output, the temporal dynamic convolution is the same as the convolution using kernels adapted to time bins.

### B. OPTIMIZED ATTENTION WEIGHT GENERATOR

In the overall process of the temporal dynamic convolution, attention weight generator Φ produce $N$-dimensional attention weight from channel $C_{in}$ and frequency $F$ data of input for each time bin to implement kernels adapted to time bin. Both frequency and channel data contain speaker information. The conventional temporal dynamic convolution [24] computes the attention weight from a feature $\boldsymbol{x}_{fc} \in \mathbb{R}^{FC_{in}\times T}$ generated by flattening along the channel and frequency dimensions. This has the advantage of being able to consider all information within each time bin. However, since the attention weights are extracted from $FC_{in}$-dimensional data, computational complexity increases exponentially as the channel and frequency dimensions increase.

To efficiently utilize the channel and frequency data for each time bin, we proposed a novel approach to represent the channel-frequency 2-dimensional data as a 1-dimensional channel vector and a 1-dimensional frequency vector. The attention weight generator computes attention weights from a feature generated by concatenating $\boldsymbol{x}_f \in \mathbb{R}^{F\times T}$ and $\boldsymbol{x}_c \in \mathbb{R}^{C_{in}\times T}$ as shown in Fig. 2. $\boldsymbol{x}_f$ produced by average pooling along channel axis, and $\boldsymbol{x}_c$ produced by average pooling along frequency axis. Two 1D convolution layers are applied to the concatenate feature $\boldsymbol{x}_{f+c}\in\mathbb{R}^{(F+C_{in})\times T}$, and the attention weights are normalized by softmax function as follows:

$$\boldsymbol{\pi}_n = \Phi\left(\boldsymbol{x}\right) = \sigma_{soft}(conv(\sigma(conv(\boldsymbol{x}_{f+c})))), \tag{3}$$

where $\sigma_{soft}$ denotes the softmax function and *conv* represents a 1D convolution operation. Softmax constraint compresses the space of aggregated kernels, so it can easily train the attention with high accuracy using less kernels per layer [59].

**TABLE 1.** Architectures of Proposed Opt-TDY-ResNet-34(×0.25).

| Layer | Structure | Output Shape |
|---|---|---|
| Input | - | $1 \times 64 \times T$ |
| Conv1 | Conv (7 , 16), stride $2 \times 1$ | $16 \times 32 \times T$ |
| Conv2 | $\begin{bmatrix}\text{TDY-Conv}(3,16)\\ \text{TDY-Conv}(3,\ 16)\end{bmatrix} \times 3,\ \text{stride}\ 1$ | $16 \times 32 \times T$ |
| Conv3 | $\begin{bmatrix}\text{TDY-Conv}(3,32)\\ \text{TDY-Conv}(3,32)\end{bmatrix} \times 4,\ \text{stride}\ 2$ | $32 \times 16 \times T\,/\,2$ |
| Conv4 | $\begin{bmatrix}\text{Conv}(3,64)\\ \text{Conv}(3,64)\end{bmatrix} \times 6,\ \text{stride}\ 2$ | $64 \times 8 \times T\,/\,4$ |
| Conv5 | $\begin{bmatrix}\text{Conv}(3,128)\\ \text{Conv}(3,128)\end{bmatrix} \times 3,\ \text{stride}\ 1$ | $128 \times 8 \times T\,/\,4$ |
| Flatten | flatten | $1024 \times T\,/\,4$ |
| Pooling | ASP | 2048 |
| Linear | FC(512) | 512 |

Conv and TDY-Conv denote vanilla convolution and temporal dynamic convolution, respectively. Numbers inside parentheses of Conv and TDY-Conv refer to (size of square kernel, the number of channels). FC denotes the fully connected layer with the number of output nodes. ASP denotes attentive statistical pooling. Nonlinear function ReLU and batch normalization are applied after every convolution.

This approach generates the kernel with each time bin from $F + C_{in}$ data, in contrast to the conventional TDY convolution that creates the kernel with each time bin from $F \times C_{in}$ data. The concatenate feature has both compressed channel and frequency information. So that, not only input feature dimension of attention weight generator is greatly reduced, but also information is preserved to enhance the performance.

The conventional attention weight generator and the optimized attention weight generator have the same hidden layer size and the number of basis kernels, except for the input feature dimension. When the hidden layer size is $C_h$ and the number of basis kernels is $N$, the conventional generator has $FC_{in}C_h + C_hN$ of the number of parameters and $2FC_{in}C_h + 2C_hN$ of floating-point operations (FLOPs). The optimized generator has $(F+C_{in})C_h + C_hN$ of the number of parameters and $2FC_{in} + 2(F + C_{in})C_h + 2C_hN$ of FLOPs containing average pooling operations. In general, $F + C_{in}$ is smaller than $FC_{in}$, so the optimized generator has fewer model parameters and less FLOPs than the conventional generator. Thus, we optimize temporal dynamic convolution by reducing the number of parameters and operations.

### C. OPTIMIZED TEMPORAL DYNAMIC CONVOLUTIONAL NEURAL NETWORK FOR TEXT-INDEPENDENT SPEAKER VERIFICATION

ResNet [8] based on 2D convolution has recently been applied to the field of speaker recognition as well as image classification, showing good recognition performance [9], [10], [11], [66], [67], [68]. In this paper, we selected ResNet-18 and ResNet-34 as baseline networks for text-independent speaker verification. The channels of ResNet-18 and ResNet-34 are modified by half (ResNet-18(×0.50) and ResNet-34(×0.50)) and a quarter (ResNet-18(×0.25) and ResNet-34(×0.25)) to reduce the computation load and prevent overfitting. To extract speaker information by adapting to each time bin, TDY-CNNs [24] were implemented by utilizing the temporal dynamic convolution instead of the vanilla convolution in ResNet-18 and ResNet-34. The main characteristic of conventional TDY-CNNs is the replacement of all vanilla convolutions with temporal dynamic convolutions. However, if the attention weight remains consistent for each time bin, the temporal dynamic convolution is equivalent to the vanilla convolution. This case leads to an increase in model parameters without any significant performance improvement. In light of this concern, we  investigated the attention weight of temporal dynamic convolutions in each layer and observed that consistent attention weights mostly emerged in the later layers of the networks. Further details regarding this investigation will be discussed in Section V. Thus, we  utilized the temporal dynamic convolution with optimized attention weight generator in earlier layers and the vanilla convolution in later layers. We  decided to call this optimized temporal dynamic convolutional neural network (Opt-TDY-CNN) by prefixing Opt-TDY- to the baseline network name. The structure of Opt-TDY-ResNet-34(×0.25) is shown in Table 1, and temporal dynamic convolution is applied in Conv2 and Conv3 layers. The networks consist of a first convolution layer that extracts global features [59] and four residual layers with convolution. Opt-TDY-ResNet-18(×0.25), Opt-TDY-ResNet-18(×0.50) and Opt-TDY-ResNet-34(×0.50) also have the same structure, but only the number of layers and channels are different. Nonlinear function ReLU and batch normalization are applied after every convolution. The extracted frame-level speaker features are aggregated by attentive statistical pooling (ASP), and the utterance-level speaker embedding is a 512-dimensional vector of linear layer result. We tried to confirm whether Opt-TDY-CNN is suitable for text-independent speaker verification and analyze how the network operates on different time bins.

## IV. EXPERIMENTAL SETUP

### A. DATASET

We trained the text-independent speaker verification models using VoxCeleb2 [10] development set with total 1,092,009 utterances for 5,994 speakers. The models are tested on VoxCeleb1 dataset [9] that has no overlap with VoxCeleb2 development set. Original Voxceleb1 test set contains 37,720 pairs generated from 40 speakers. Voxceleb1 may not provide sufficient regularization performance due to the small number of speakers, so we test the models on VoxCeleb1-E test set consisting of 581,480 pairs generated from 1,251 speakers. In addition, VoxCeleb1-H test set consisting of 552,536 pairs of the same nationality and gender is also utilized to verify the model performance.

### B. INPUT REPRESENTATIONS

For input of speaker verification models, 64-dimensional log Mel-spectrograms are extracted using a hamming window of width 25ms with step 10ms and number of fast Fourier transform 512. We randomly crop the Mel-spectrogram

**TABLE 2. Text-Independent Speaker Verification Performance of Opt-TDY-ResNet-34(×0.25) on Different Number of Basis Kernels.**

| Opt-TDY-ResNet-34(×0.25) | #Parm | EER (%) | MinDCF |
|---|---|---|---|
| $N = 2$ | 2.77M | 1.41 | 0.114 |
| $N = 4$ | 2.96M | 1.40 | 0.106 |
| $N = 6$ | 3.14M | 1.37 | 0.104 |
| $N = 8$ | 3.33M | **1.32** | **0.103** |
| $N = 10$ | 3.52M | 1.35 | 0.113 |

**TABLE 3. Text-Independent Speaker Verification Performance of Opt-TDY-ResNet-34(×0.25) on Different Pooling Layers.**

| Temporal Pooling Layer | #Parm | EER (%) | MinDCF |
|---|---|---|---|
| Average pooling | 2.55M | 1.63 | 0.129 |
| Statistical pooling | 3.07M | 1.41 | 0.105 |
| Self-attentive pooling | 2.81M | 1.36 | 0.108 |
| Attentive statistical pooling | 3.33M | **1.32** | **0.103** |

segment for 2 seconds from each utterance and use the Mel-spectrograms with a size of 64 × 200 for training with no data augmentation. Mean and variance normalization is performed on every frequency bin of the Mel-spectrogram [69]. In Opt-TDY-ResNets, the TDY convolution in Conv2 layer determines the kernel for each 200 time bins, and the TDY convolution in Conv3 layer determines the kernel for each 100 time bins.

### C. LOSS FUNCTION AND IMPLEMENTATION DETAILS

Our implementation is based on the PyTorch with 4 NVIDIA TITAN RTX. The Adam optimizer with weight decay $5 \times 10^{-5}$ is used. An initial learning rate is $10^{-3}$ decreasing by a factor of 0.75 every 10 epochs. Batch normalization is used with a fixed mini-batch size of 256 on each GPU, and no data augmentation is performed during training. For TDY-CNN, near-uniform attention in early training epochs can address this optimization problem [59], so we reduced the temperature in softmax constraint from 30 to 1 linearly in the first 10 epochs.

The networks are trained using a loss function combining the Angular Prototypical (AP) loss with the vanilla softmax loss. It demonstrates a better verification performance than using each or other combinations of the loss functions [66]. In addition, we do not apply data augmentation and score normalization [70], [71], which are additional performance improvement techniques, to isolate the impact of temporal dynamic convolution on performance improvement, specifically regarding time-varying characteristic of speech.

### D. EVALUATION METRICS

We sample ten 4-second segments at the same intervals from each test segment, and compute the 10 × 10 (total 100) cosine similarities between every pair of segments. The mean of the 100 similarities is used as the final pairwise score [10], [66]. Based on the average similarity score, we calculate Equal Error Rate (EER) and the minimum value of the cost function $C_{det}$ as evaluation metrics of verification. EER is the rate at which both acceptance and rejection errors are equal. The $C_{det}$ is a weighted sum of false-reject and false-accept error probabilities as follows:

**TABLE 4. Text-Independent Speaker Verification Performance of Opt-TDY-ResNet-34(×0.25) using Different Features for Generating Attention Weight.**

| Network | #Parm | FLOPs | EER (%) | MinDCF |
|---|---|---|---|---|
| Opt-TDY-ResNet-34(×0.25)<br>using average frequency features | 12.1M | 22.73G | 1.43 | 0.109 |
| Opt-TDY-ResNet-34(×0.25)<br>using average channel features | 12.1M | 22.73G | 1.39 | 0.107 |
| Opt-TDY-ResNet-34(×0.25)<br>using concatenated features | 12.1M | 22.74G | **1.32** | **0.100** |
| TDY-ResNet-34(×0.25)<br>using flattened features [24] | 16.7M | 23.21G | 1.34 | 0.103 |

$$C_{det} = C_{miss} \times P_{tar} \times P_{miss} + C_{fa} \times (1 - P_{tar}) \times P_{fa} \quad (4)$$

where missed detection cost $C_{miss}$ is 1, spurious detection cost $C_{fa}$ is 1, and priori target probability $P_{tar}$ is 0.05 [28], [35].

## V. RESULT AND ANALYSIS

In this section, ablation studies are performed on Opt-TDY-ResNet-34(×0.25), and we evaluate the Opt-TDY-CNNs for text-independent speaker verification. Also, we analyze the kernel variation and operation of temporal dynamic convolution with respect to time bins data with phoneme information.

### A. TEXT-INDEPENDENT SPEAKER VERIFICATION USING OPTIMIZED TEMPORAL DYNAMIC CONVOLUTIONAL NEURAL NETWORK

#### 1) THE NUMBER OF BASIS CONVOLUTION KERNELS

The temporal dynamic convolution produces kernels as weighted summation of basis kernels. The number of basis kernels $N$ is directly related to the model complexity affecting the speaker verification performance. We compared speaker verification performance of Opt-TDY-ResNet-34(×0.25) with various $N$ and the result is shown in Table 2. Text-independent speaker verification performance is getting improved until $N$ reaches 8 to have the best performance with EER of 1.32%. On the contrary, the performance is degraded when $N$ is 10 because of the difficulty of optimization and the overfitting as the representation power of the model increases, similar to previous result [59]. Thus, the temporal dynamic convolution using 8 basis kernels is an optimal for text-independent speaker verification, so we decided that $N$ is 8 as default setup.

#### 2) TEMPORAL POOLYING LAYER

The temporal pooling layer that converts frame-level features into an utterance-level feature is one of the factors influencing the speaker verification performance, so we compared the performance of Opt-TDY-CNN on different type of pooling layers. Table 3 shows the text-independent speaker verification performance of Opt-TDY-ResNet-34(×0.25)

**TABLE 5.** Text-Independent Speaker Verification Performance of Opt-TDY-ResNet-34(×0.25) depending on Application of Optimized Temporal Dynamic Convolution and Vanilla Convolution to Each Layer.

| Network | Layer | | | | #Parm | EER (%) | MinDCF |
|---|---|---|---|---|---|---|---|
| | Conv2 | Conv3 | Conv4 | Conv5 | | | |
| ResNet-34(×0.25) | V | V | V | V | 2.65M | 1.52 | 0.118 |
| Opt-TDY-ResNet-34(×0.25) | T | V | V | V | 2.83M | 1.41 | 0.109 |
| | T | T | V | V | 3.33M | **1.32** | **0.103** |
| | T | T | T | V | 6.35M | 1.36 | 0.109 |
| | T | T | T | T | 12.1M | **1.32** | **0.100** |

'V' indicates that vanilla convolution is applied to that, and 'T' indicates temporal dynamic convolution is applied to that layer.

using following temporal pooling methods: average pooling [72], [73], [74], statistical pooling [1], [75], self-attentive pooling [76], [77], [78], and attentive statistical pooling [79]. Self-attentive pooling and attentive statistical pooling methods utilize self-attention mechanism, whereas average pooling and statistical pooling methods apply the same weight to all frame-level features. The self-attention mechanism of temporal pooling layer provides additional improvement on the speaker verification performance, and attentive statistical pooling achieves the best performance with EER of 1.32%. Since temporal dynamic convolution extracts frame-level speaker features from all frames and self-attention mechanism excludes unnecessary frame-level speaker features in non-speech frames, this result demonstrates that the two methods work complementarily to improve speaker verification performance. Thus, we selected the attentive statistical pooling (ASP) method as the temporal pooling layer of Opt-TDY-CNN and used it for the text-independent speaker verification.

### 3) ATTENTION WEIGHT GENERATION USING DIFFERENT FEATURES

The attention weight of basis kernels is generated from each time bin data of input using two fully-connected layers. The data of each time bin consists of channel and frequency dimensions, and speaker verification performance varies depending on which dimension of data is used. So, we compare the performance of Opt-TDY-ResNet-34(×0.25) generating attention using average frequency features, average channel features, concatenated features, and flatten feature. The average frequency and channel features are derived by applying average pooling to channel and frequency dimension of each time bin, respectively. The concatenated feature is derived by concatenating average frequency and channel features. The flattened feature is a 1-dimensional vector consisting of all channel-frequency data, and conventional TDY-CNN [24] generated attention using the flatten feature. In this ablation study, temporal dynamic convolution replaces all vanilla convolution except the first convolution. Table 4 shows the speaker verification results of Opt-TDY-ResNet-34(×0.25) using different features for generating attention weight. Comparing the effect of channel and frequency dimension, Opt-TDY-ResNet-34(×0.25) using average channel features outperformed Opt-TDY-ResNet-34(×0.25) using average frequency features. This result indicates that channel dimension data has more speaker representation power than frequency dimension data. Since not only channel dimension but also frequency dimension is able to represent time bin data, Opt-TDY-ResNet-34(×0.25) using concatenated features showed the best performance as EER of 1.32%. In addition, Opt-TDY-ResNet-34(×0.25) using concatenated features showed better performance by utilizing fewer network parameters and FLOPs compared to TDY-ResNet-34(×0.25) that utilizes all data of each time bin. This may be due to either the presence of irrelevant information in all data of each time bin or underfitting caused by a large number of model parameters. Thus, generating attention weight using concatenated features in temporal dynamic convolution is appropriate for the text-independent speaker verification.

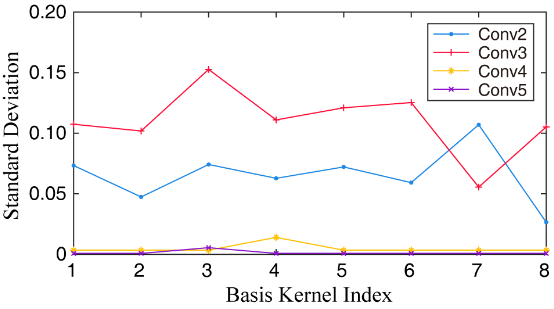


**FIGURE 3.** Standard deviation of attention weights for 8 basis kernels in Conv2, Conv3, Conv4, and Conv5 layers of Opt-TDY-ResNet-34(×0.25).

### 4) TEMPORAL DYNAMIC CONVOLUTION AT DIFFERENT LAYERS

Opt-TDY-CNNs refine only speaker information from spectrogram that includes speaker, phoneme, intonation information, etc. The temporal dynamic convolution extracts speaker information by adapting to the temporal information of layer input, so it should be applied from the earlier layer of network that utilizes the data with relatively large change over time. We compared text-independent speaker verification performance when temporal dynamic convolution was sequentially applied from Conv2 to Conv5 of Opt-TDY-ResNet-34(×0.25) as shown in Table 5. ResNet-34(×0.25) using only vanilla convolution achieved EER of 1.52%, and all Opt-TDY-ResNet-34(×0.25) outperforms ResNet-34(×0.25). The performance improved as the number of layers using temporal dynamic convolution increased. Opt-TDY-ResNet-34(×0.25) using temporal dynamic convolution at all layers, which is the same structure of TDY-CNN, achieves the best performance with EER of 1.32%. The best performance is also achieved with temporal dynamic convolution applied to Conv2 and Conv3 layers, and even the network size decreased significantly from 12.1M to 3.33M in these two best EER cases. To analyze the cause of this result, we compared the standard deviation of attention weights for 8 basis kernels in different layer groups across 4,874 utterances of Voxceleb1 test dataset [9], as shown in Fig. 3.

**TABLE 6.** Text-Independent Speaker Verification Performances of Networks on Voxceleb1, Voxceleb1-E, and Voxceleb1-H Test Sets without Data Augmentation.

| Networks | #Parm | VoxCeleb1 cl. | | VoxCeleb1-E cl. | | VoxCeleb1-H cl. | |
|---|---|---|---|---|---|---|---|
| | | EER (%) | MinDCF | EER (%) | MinDCF | EER (%) | MinDCF |
| ResNet-18(×0.25) | 2.01M | 1.88 | 0.146 | 1.97 | 0.139 | 3.68 | 0.234 |
| ResNet-18(×0.29) | 2.40M | 1.79 | 0.131 | 1.85 | 0.131 | 3.47 | 0.223 |
| Opt-TDY-ResNet-18(×0.25) | 2.40M | **1.71** | **0.116** | 1.72 | 0.119 | 3.20 | 0.206 |
| ResNet-18(×0.50) | 5.42M | 1.38 | 0.111 | 1.49 | 0.107 | 2.92 | 0.190 |
| ResNet-18(×0.59) | 6.92M | 1.32 | 0.105 | 1.38 | 0.098 | 2.80 | 0.184 |
| Opt-TDY-ResNet-18(×0.50) | 6.92M | **1.24** | **0.093** | 1.36 | 0.096 | 2.68 | 0.173 |
| ResNet-34(×0.25) | 2.65M | 1.52 | 0.118 | 1.65 | 0.116 | 3.15 | 0.204 |
| ResNet-34(×0.29) | 3.34M | 1.41 | 0.110 | 1.52 | 0.105 | 2.99 | 0.191 |
| Opt-TDY-ResNet-34(×0.25) | 3.33M | **1.32** | **0.103** | 1.47 | 0.104 | 2.84 | 0.183 |
| ResNet-34(×0.50) | 7.95M | 1.27 | 0.103 | 1.33 | 0.095 | 2.67 | 0.172 |
| ResNet-34(×0.59) | 10.6M | 1.18 | 0.095 | 1.30 | 0.093 | 2.64 | 0.171 |
| Opt-TDY-ResNet-34(×0.50) | 10.6M | **1.07** | **0.092** | 1.29 | 0.091 | 2.52 | 0.161 |
| ResNet-50 [10] | 67.0M | 3.95 | 0.429 | 4.42 | 0.523 | 7.33 | 0.673 |
| Thin ResNet-34 [11] | 12.4M | 2.87 | 0.310 | 2.95 | - | 4.93 | - |
| RawNet2 [80] | 13.3M | 2.48 | - | 2.57 | - | 4.89 | - |
| ResNetSE-34-Q/SAP [67] | 1.40M | 1.47 | 0.119 | 1.74 | 0.130 | 3.44 | 0.229 |
| ResNetSE-34-H/ASP [67] | 8.00M | 1.21 | 0.098 | 1.42 | 0.099 | 2.77 | 0.175 |
| ECAPA-TDNN [6] | 15.4M | 1.28 | 0.099 | 1.41 | 0.102 | 2.93 | 0.190 |

The average standard deviations of attention weights for Conv2 and Conv3 layers are 0.065 and 0.110, respectively. In contrast, for Conv4 and Conv5 layers, the standard deviations of attention weights are 0.006 and 0.001, indicating that the attention weights generated from time bins data are almost identical. Since the kernel is determined by weighted summation of basis kernel, the kernels of temporal dynamic convolution in Conv4 and Conv5 almost constant regardless of the time bins data. Thus, this result indicates that temporal dynamic convolution is almost equivalent to vanilla convolution in Conv4 and Conv5 layers. The reason is that only the speaker information but not the time-varying information remains as the dominant features in the outputs of the relatively late layers. In addition, applying temporal dynamic convolution to all layers like TDY-CNNs increases the chance of instability of parameter optimization, which can lead to performance disorder. Thus, we applied temporal dynamic convolution only for Conv2 and Conv3 layers in an optimal sense.

### 5) TEXT-INDEPENDENT SPEAKER VERIFICATION RESULTS

The structure of Opt-TDY-CNN was determined through the various ablation studies in previous sections. Moreover, we compared text-independent speaker verification performance between baseline networks using vanilla convolution and Opt-TDY-CNNs. Table 6 shows the speaker verification performance of ResNets and Opt-TDY-ResNets without data augmentation. ResNets, the baseline networks, had the same structure with ASP as Opt-TDY-ResNets in Table 1, but utilized only vanilla convolution. Among the baseline networks, ResNet-34(×0.50) shows good speaker verification performance with EER of 1.27% on Voxceleb1 test set. Opt-TDY-ResNet-34(×0.50) outperforms ResNet-34(×0.50) with the best speaker verification performance of EER 1.07% on Voxceleb1 test set. Moreover, all Opt-TDY-ResNets outperform ResNets in Voxceleb1, Voxceleb1-E, and Voxceleb1-H test sets.

Although temporal dynamic convolution uses the same channel and kernel size as vanilla convolution to extract speaker information, the speaker verification performance is improved by utilizing kernels adapted to the information of every time bins. However, temporal dynamic convolution requires additional computational effort for generating the kernel that adapts to time bins using eight basis kernels. Thus, it is necessary to compare speaker verification performance between the baseline networks and Opt-TDY-CNNs under the similar number of parameters. We increased channel size of the baseline networks by the ratio in parentheses next to the network name so that the number of parameters become close to that of Opt-TDY-CNNs. As shown in Table 6, the channel increment of ResNets improved speaker verification performance compared to the conventional ResNets, but they do not outperform the Opt-TDY-ResNets. This indicates that the kernel adapted to time bin is more effective than simple channel increasing of static kernel to improve speaker verification performance. That is, Opt-TDY-CNN considering changes in acoustic characteristics over time is effective for text-independent speaker verification.

In addition, we also compared text-independent speaker verification performance between Opt-TDY-ResNets and the state-of-the-art networks: ResNet-50 [10], Thin-ResNet-34 [11], RawNet2 [80], ResNet34 Q/SAP [67], ResNet34 H/ASP [67], and ECAPA-TDNN(C=1024) [6]. These state-of-the-art networks were trained with Voxceleb2 without data augmentation and tested with Voxceleb1, just like our experiment conditions. We used the pre-trained network results of the state-of-the-art networks, and the result of ECAPA-TDNN was carried out by our implementation. The text-independent speaker verification results of the state-of-the-art networks are shown in Table 6. Opt-TDY-ResNet-34(×0.50) with EER of 1.07% has the best speaker verification performance compared to the state-of-the-art networks. From the speaker verification results, we conclude that TDY-CNN are more effective for text-independent speaker

**TABLE 7.** Phoneme Groups of 58 Phonemes in TIMIT Dataset.

| Phoneme Group | Phoneme Symbol |
|---|---|
| Vowels | iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, ux, er, ax, ix, axr, ax-h |
| Semivowels and glides | l, r, w, y, hh, hv, el |
| Nasals | m, n, ng, em, en, eng, nx |
| Fricatives and Affricates | s, sh, z, zh, f, th, v, dh, jh, ch |
| Stops and Closures | b, d, g, p, t, k, dx, q, bcl, dcl, gcl, pcl, tck, kcl |

verification than the state-of-the-art networks using vanilla convolution.

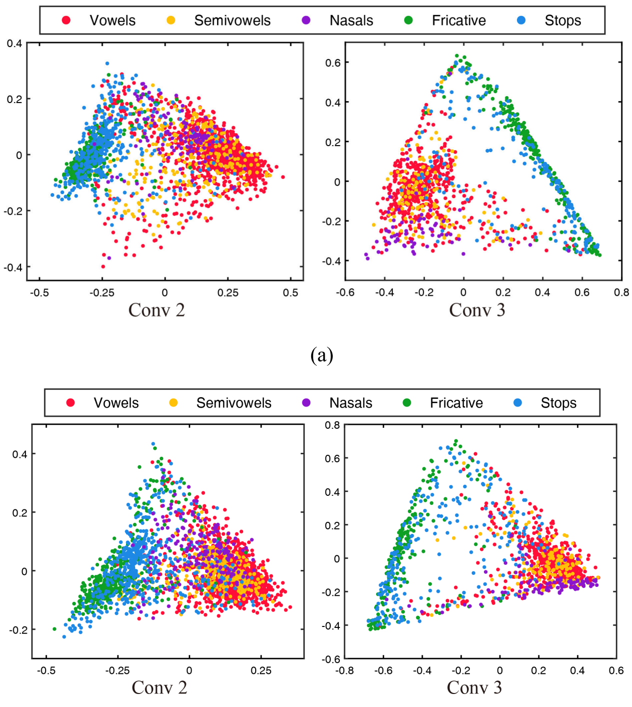
### B. ANALYSIS OF KERNELS ON TEMPORAL DYNAMIC CONVOLUTION WITH RESPECT TO PHONEMES

Opt-TDY-CNN improves the text-independent speaker verification performance compared to the baseline networks by extracting speaker information using kernels adapted to time bins. In this section, we compare the variation of the kernels on different phonemes and analyze the time-frequency characteristics of utterances from which the kernels extract speaker information.

#### 1) VARIATION OF KERNELS WITH RESPECT TO PHONEMES

As the first step in a detailed analysis of Opt-TDY-CNN, we analyzed the kernel variation of temporal dynamic convolution depending on the information of time bins to verify how the temporal dynamic convolution adapts to utterances for the text-independent speaker verification. The smallest unit that can split utterances is *phonemes*, so the kernels of temporal dynamic convolution are compared depending on the phoneme information corresponding to time bins. In addition, the kernels of temporal dynamic convolution are obtained by the weighted summation of basis kernels, so attention weights of basis kernels indicate the input adaptation tendency of the kernel. Thus, we focus on the correlation between the attention weights and corresponding phonemes on different layer depths.

In this experiment, we used TIMIT dataset [81] which provides 6,300 utterances with 58 phoneme labels from 630 speakers. A total of 58 phonemes are categorized into five groups (vowels; semivowels and glides; nasals; fricatives and affricates; stops and closures), as shown in Table 7. We denote these five phoneme groups as vowels, semivowels, nasals, fricatives, and stops, respectively. Attention weights are extracted on the five phoneme groups from the pre-trained Opt-TDY-ResNet-34(×0.50), which showed the best performance of 1.07% EER. In addition, attention weights are also compared at two layers, Conv2 and Conv3, in network description of Table 1. Conv layers of ResNet are composed of multiple convolutions, so the attention weights of the intermediate temporal dynamic convolution in the layers are compared.



(a)

(b)

**FIGURE 4.** Low-dimensional PCA projection of attention weights in speaker (a) MPDF0 and (b) FSLS0 for Conv2 and Conv3 layers on five phoneme groups using Opt-TDY-ResNet-34(×0.50). Semivowels label contains glides, fricative label contains affricative, and stops label contains closures. The attention weights of temporal dynamic convolution are phoneme-dependent in both speakers.

In order to confirm the correlation between the attention weights and the five phoneme groups, we visualized the attention weights distribution with respect to phonemes. Among the 630 speakers in TIMIT dataset, speaker MPDF0 with the largest variance of attention weights and FSLS0 with the smallest variance of attention weights are selected. The attention weights of MPDF0 and FSLS0 are visualized presenting five phoneme groups at Conv2 and Conv3 layers using a principal component analysis (PCA) applied to the attention weights of 8 basis kernels as shown in Fig. 4. The attention weights distribution of semivowels is similar to the distribution of vowels, and the attention weights distribution of stops is similar to the distribution of fricatives. The distributions of these two groups are close within their own group while the groups are distinctively distributed. The attention weights of nasals are located close to the vowels, but they are grouped separately. Based on these results, it can be inferred that the attention weights vary depending on the phoneme groups corresponding to each time bin.

However, with only a few samples, we cannot verify the hypothesis that the temporal dynamic convolution utilizes different kernels depending on the time-varying information such as phonemes in time bins. Thus, we tried to compare the Euclidian distance between the attention weight distributions extracted from 6,300 utterances in TIMIT dataset by considering phoneme information of time bins. An average attention weight is assigned as the centroid of attention weights in each phoneme group. The inter-phoneme distance, which is the distance between the attention weights of phoneme groups, is defined as the Euclidian distance between the centroid

**TABLE 8.** Average intra-phoneme distance and average inter-phoneme distance of attention weights at (a) Conv2 and (b) Conv3 layers on five phoneme groups in TIMIT dataset.

(a) Conv2 Layer

| **Phoneme** | Vowels | Semivowels | Nasals | Fricatives | Stops |
|---|---|---|---|---|---|
| Vowels | 0.170 | 0.100 | 0.175 | 0.446 | 0.362 |
| Semivowels | - | 0.192 | 0.092 | 0.391 | 0.296 |
| Nasals | - | - | 0.194 | 0.349 | 0.239 |
| Fricatives | - | - | - | 0.168 | 0.146 |
| Stops | - | - | - | - | 0.226 |

(b) Conv3 Layer

| **Phoneme** | Vowels | Semivowels | Nasals | Fricatives | Stops |
|---|---|---|---|---|---|
| Vowels | 0.166 | 0.054 | 0.171 | 0.575 | 0.491 |
| Semivowels | - | 0.205 | 0.143 | 0.540 | 0.453 |
| Nasals | - | - | 0.204 | 0.580 | 0.482 |
| Fricatives | - | - | - | 0.298 | 0.111 |
| Stops | - | - | - | - | 0.353 |

attention weights. The intra-phoneme distance, which is the distance within attention weights of the same phoneme group, is defined as average Euclidian distance of the attention weights from the centroid attention weight. The inter and intra-phoneme distance calculation are performed on the utterances within the same speaker in order to exclude the effect of speaker variation on the adaptive kernel. We average the inter and intra-phoneme distances for 630 speakers, and the results are shown in Table 8.

Diagonal values indicate the average intra-phoneme distance, and upper triangular values indicate the average inter-phoneme distances. At Conv3 layer, the average inter-phoneme distance between vowels and semivowels is the shortest distance of 0.054. Since this distance is about 1/4 of the average intra-phoneme distances of 0.166 and 0.205, the attention weights distributions of vowels and semivowels are close together. This result indicates that the attention weights of vowels and semivowels are similar. Likewise, fricatives and stops groups also show a short inter-phoneme distance of 0.111, which is about 1/3 of the average intra-phoneme distances of 0.166 and 0.205, so fricatives and stops have similar attention weights. However, the average inter-phoneme distance between vowels and fricatives is 0.575, which is about 3.5 times larger than the average intra-phoneme distance of vowels. This means that the attention weights of vowels and fricatives are not similar. Nasals also have different attention weights compared to the fricatives because the average inter-phoneme distance between nasals and fricatives is the largest distance of 0.580. The average inter-phoneme distance between vowels and nasals is 0.170, so the attention weights of nasals are similar to the attention weights of vowels rather than fricatives, but not as semivowels. These tendencies of inter and intra-phoneme distances of phonemes are also shown at Conv2 layer. Thus, these results imply that the attention weights of temporal dynamic convolution are changed with respect to the three phoneme groups: vowels and semivowels; fricatives and stops; and nasals.

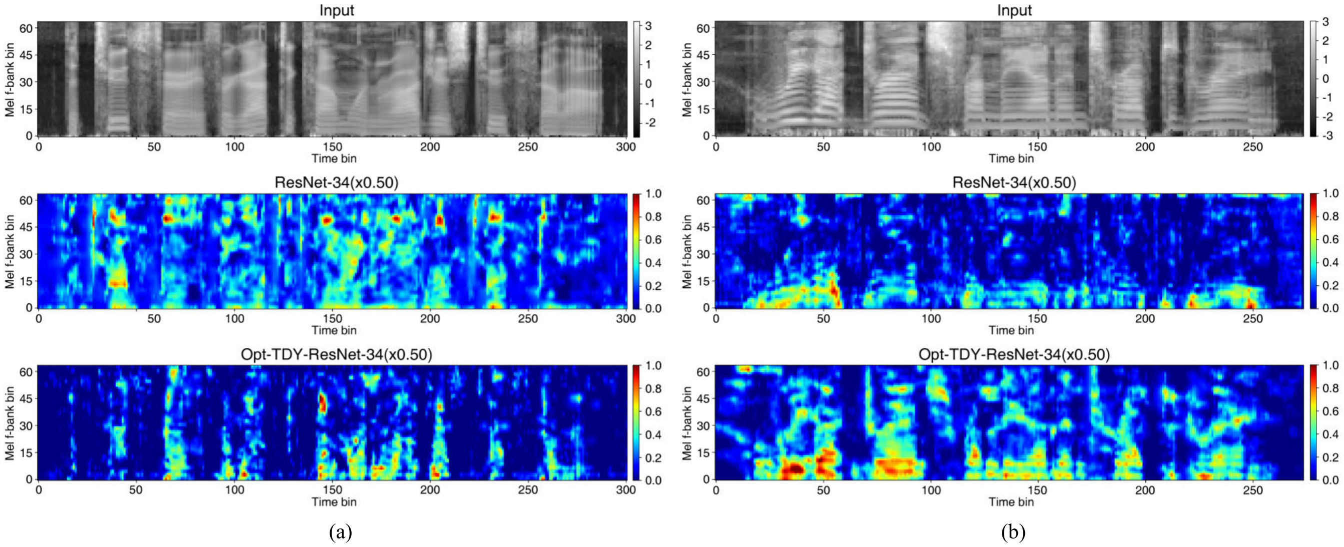
**TABLE 9.** Average Intra-Phoneme Distance and Average Inter-Phoneme Distance of Attention Weights at Conv2 and Conv3 Layers on Three Phoneme Groups in TIMIT Dataset.

| Layer | Conv2 | | | Conv3 | | |
|---|---|---|---|---|---|---|
| Phoneme | V+SV | N | F+S | V+SV | N | F+S |
| V+SV | 0.178 | 0.157 | 0.387 | 0.174 | 0.164 | 0.525 |
| N | - | 0.194 | 0.290 | - | 0.204 | 0.531 |
| F+S | - | - | 0.215 | - | - | 0.333 |

'V' indicates vowels, 'SV' indicates semivowels, 'N' indicates nasals, 'F' indicates fricatives, and 'S' indicates stops.

Moreover, we compared the inter/intra-phoneme distances of attention weights between these three phoneme groups, and the results are presented in Table 9. Regarding the fricatives and stops group, the inter-phoneme distances between this group and the other two phoneme groups were comparable to the sum of the intra-phoneme distances within each group. Specifically, distance between the centroids of the two distributions is similar to the sum of the standard deviations of each group. This result indicates that the attention distributions do not overlap significantly, implying that the attention weights of the fricatives and stops group differ significantly from the other two groups. On the other hand, the inter-phoneme distance between the nasals group and the vowels and semivowels group appeared similar to the intra-phoneme distance within each group. More precisely, the distance between the centroids of the two distributions was comparable to the standard deviations of each group. This result suggests the existence of some similarities in the attention weights between the nasals group and the vowels and semivowels group. However, it is important to note that these similarities do not indicate a completely consistent attention pattern between the two groups. These trends observed in phoneme distances align with the results presented in Table 8. Another noteworthy observation is that the intra-phoneme distance within the vowels and semivowels group is similar to the intra-phoneme distances within the vowels group and the semivowels group. This implies that the attention weights of the vowels group and the semivowels group are almost identical, a similarity also observed in the fricatives and stops group. Thus, we confirm that the attention weights can be categorized into the three distinct phoneme groups.

Interestingly, the tendency of attention weight distributions depending on the phoneme groups are related to the phoneme generation mechanism and its acoustic characteristics. The pronounced sounds of vowels and semivowels are dominantly affected by vocal cords and tract. Phonemes of vowels and semivowels have similar acoustic characteristics, so the attention weights of vowels and semivowels are similar. In addition, the pronounced sounds of nasals are dominantly affected by nasal cavity. The vowels, semivowels, and nasal also have similar acoustic characteristics based on resonance, but they are slightly different due to different production path. For this reason, the attention weights of nasals are distributed relatively far from the attention weights of vowels and semivowels. On the other hand, the pronounced sounds of fricatives are dominantly affected by the turbulent



(a) (b)

**FIGURE 5.** Speaker activation maps of Conv2 and Conv3 layers in ResNet-34(×0.50) and Opt-TDY-ResNet-34(×0.50) for (a) SX282 by MPDF0 and (b) SX66 by FSLS0.

sound caused at the constriction in mouth cavity, and the stops involve abrupt and impulsive sounds. Phonemes of fricatives and stops have noise-like acoustic characteristics so that they have similar attention weights. However, the noise-like acoustic characteristics share no similarity with characteristics of voiced sound such as vowels, semivowels, and nasals. So, the attention weight of fricatives and stops are well distinguished from the attention weights of vowels, semivowels, and nasals. Therefore, each phoneme group is distinguished by its pronunciation mechanism which determines acoustic characteristics, so that the kernel of temporal dynamic convolution changes accordance with the acoustic characteristics of time bins.

### 2) SPEAKER ACTIVATION MAP OF OPT-TDY-CNN USING GRAD-CAM

The kernel of temporal dynamic convolution adapts to acoustic characteristics of time bins, and it is necessary to verify whether the kernels are meaningful filters for speaker embedding extraction. In this section, we explain Opt-TDY-CNN using gradient-weighted class activation mapping (Grad-CAM) [25], which generates a class activation map (CAM) using class-specific gradients flowing into the final convolutional layer. However, there are two problems in directly applying Grad-CAM to speaker verification models. Firstly, the speaker verification model only aims to extract speaker embeddings without recognizing speakers, so it is difficult to clearly define gradients of the target speaker to be recognized. To address this problem, we change the speaker verification task to speaker identification task by attaching a classification layer to the end of pre-trained speaker verification model. Grad-CAM is applied to the speaker identification model consisting of the classification layer and the pre-trained speaker verification model. The classification layer was trained using 630 speakers of TIMIT dataset. Of the ten utterances of each speaker in TIMIT dataset, two utterances are used as test set, and the remaining eight utterances are used as training set [82], [83]. Secondly, all time-frequency data of an utterance comes from a specific speaker according to the definition of speaker verification task, so it is meaningless to display CAM for a specific speaker. In addition, the purpose of this analysis is to explain how Opt-TDY-CNN extracts speaker information depending on the time bins data within the utterance. Thus, we use gradients flowing into the first convolution instead of the gradients from last layer, which we named as speaker activation map (SAM).

The SAMs of ResNet-34(×0.50) and Opt-TDY-ResNet-34(×0.50) were visualized using the gradients of Conv2 and Conv3 layers for SX282 by MPDF0 and SX66 by FSLS0 as shown in Fig. 5. Opt-TDY-ResNet-34(×0.50) activates the low-frequency section for voiced sounds and the high-frequency section for unvoiced sounds. Especially, formant frequencies and harmonics of fundamental frequency are emphasized. This activation maps of Opt-TDY-ResNet-34(×0.50) match the frequency pattern of phonemes. These results indicate that the temporal dynamic convolution extracts speaker information from the frequency patterns related to phonemes. Similarly, ResNet-34(×0.50) activates wide frequency region of utterances and vague outline of each phoneme, but does not emphasize the detailed frequency patterns of phonemes like Opt-TDY-ResNet-34(×0.50). That is, the temporal dynamic convolution considers more detailed frequency patterns in a precise way than the vanilla convolution. Therefore, we verified that Opt-TDY-CNN extracts speaker information from the significant part of a given utterance following the detailed frequency pattern of phonemes compared to the baseline network.

## VI. CONCLUSION

In this paper, we optimized the temporal dynamic convolutional neural network which extracts speaker embedding by adapting to time-varying data of input utterance based on the various analysis. The temporal dynamic convolution utilized kernels obtained by the weighted summation of basis kernels with softmax constraint on attention weights. The contribution of this work is that the attention weight of basis kernels is produced from average channel and frequency features rather than all data in each time bin. Compared to the conventional TDY-ResNet-34(×0.25), the model parameters of Opt-TDY-ResNet-34(×0.25) were reduced by 26%, and the EER was also improved from 1.34% to 1.32%. However, temporal dynamic convolutions in later layers use steady attention weights of basis kernels regardless of time bins, which means that temporal dynamic convolution is equal to vanilla convolution in later layers. Thus, we applied the temporal dynamic convolutions to only the earlier layers of networks replacing vanilla convolution in the structure of Opt-TDY-CNNs. As a result, Opt-TDY-ResNet-34(×0.50) showed the best performance with 1.07% of EER. Although the vanilla convolution has more channels than temporal dynamic convolution, Opt-TDY-CNNs outperformed the baseline networks by 9.32%. Furthermore, kernels of temporal dynamic convolution adapt itself to the time-varying information of each time bin such as phonemes using only speaker information during training. The kernel of temporal dynamic convolution extracts speaker information from frequency characteristics of each time bin. Therefore, the optimized temporal dynamic convolutional neural network can give precise and efficient text-independent speaker verification by adapting to time-varying acoustic characteristics of utterances with random texts.

Opt-TDY-CNNs were designed to address the time-varying characteristic of speech resulting from phonetic variability in random text. However, in a real application, external factors such as changes in the acoustic environment and the presence of noise can introduce contamination to the speaker information. Since the kernel for each time is influenced by both noise and speaker-related information in the channel and frequency data, performance degradation can occur in the presence of noisy speech. As part of our future work, we plan to evaluate the performance of Opt-TDY-CNNs in the presence of noisy speech by employing data augmentation for extracting speaker information while considering the time-varying characteristic of speech, excluding the influence of noise. Furthermore, the time-varying characteristic of speech encompasses not only textual variations but also language differences. To investigate the effectiveness of Opt-TDY-CNNs in handling language-related variations, we intend to utilize NIST SRE datasets [84], [85], which introduce language mismatch between the training and test data. This analysis will provide insights into the ability of Opt-TDY-CNNs to handle diverse time-varying characteristic caused by language variations. Moreover, we will apply Opt-TDY-CNNs to various state-of-the-art networks for text-independent speaker verification to validate whether the temporal dynamic convolution can be applied to various networks as well as ResNets. Also, we plan to analyze principle components of the basis kernels rather than attention weights and improve temporal dynamic convolution for better text-independent speaker verification.

## REFERENCES

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5329–5333.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[3] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[4] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proc. Interspeech*, Sep. 2018, pp. 3743–3747.

[5] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using X-vectors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5796–5800.

[6] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, Oct. 2020, pp. 3830–3834, doi: 10.21437/Interspeech.2020-2650.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[9] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, Aug. 2017, pp. 2616–2620.

[10] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, Sep. 2018, pp. 1086–1090.

[11] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "VoxCeleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, Mar. 2020, Art. no. 101027.

[12] J. Huh, H. Soo Heo, J. Kang, S. Watanabe, and J. Son Chung, "Augmentation adversarial training for self-supervised speaker recognition," 2020, *arXiv:2007.12085*.

[13] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 1007–1013.

[14] J. P. Eatock and J. S. Mason, "A quantitative assessment of the relative speaker discriminating properties of phonemes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 1994, pp. 1–133.

[15] C.-S. Jung, M. Young Kim, and H.-G. Kang, "Selecting feature frames for automatic speaker recognition using mutual information," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1332–1340, Aug. 2010.

[16] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, vol. 52, no. 1, pp. 12–40, Jan. 2010.

[17] S. K. Gupta and M. Savic, "Text-independent speaker verification based on broad phonetic segmentation of speech," *Digit. Signal Process.*, vol. 2, no. 2, pp. 69–79, Apr. 1992.

[18] R. Faltlhauser and G. Ruske, "Improving speaker recognition using phonetically structured Gaussian mixture models," in *Proc. 7th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Sep. 2001, pp. 1–4.

[19] S. S. Kajarekar and H. Hermansky, "Speaker verification based on broad phonetic categories," in *Proc. Speaker Odyssey Speaker Recognit. Workshop*, 2001, pp. 1–5.

[20] M. Hebert and L. P. Heck, "Phonetic class-based speaker verification," in *Proc. 8th Eur. Conf. Speech Commun. Technol. (Eurospeech )*, Sep. 2003, pp. 1–4.

[21] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, ''Speaker verification using adapted Gaussian mixture models,'' *Digit. Signal Process.*, vol. 10, nos. 1–3, pp. 19–41, Jan. 2000.

[22] E. G. Hansen, R. E. Slyh, and T. R. Anderson, ''Speaker recognition using phoneme-specific GMMs,'' in *Proc. ODYSSEY Speaker Language Recognit. Workshop*, 2004, pp. 179–184.

[23] S.-H. Kim and Y.-H. Park, ''Adaptive convolutional neural network for text-independent speaker recognition,'' in *Proc. Interspeech*, Aug. 2021, pp. 66–70.

[24] S. Kim, H. Nam, and Y. Park, ''Temporal dynamic convolutional neural network for text-independent speaker verification and phonemic analysis,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 6742–6746.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[26] R. Caruana, ''Multitask learning,'' *Mach. Learn.*, vol. 28, pp. 41–75, Dec. 1997.

[27] S. Ruder, ''An overview of multi-task learning in deep neural networks,'' 2017, *arXiv:1706.05098*.

[28] N. Chen, Y. Qian, and K. Yu, ''Multi-task learning for text-dependent speaker verification,'' in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1–15.

[29] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, ''Deep feature for text-dependent speaker verification,'' *Speech Commun.*, vol. 73, pp. 1–13, Oct. 2015.

[30] S. Dey, T. Koshinaka, P. Motlicek, and S. Madikeri, ''DNN based speaker embedding using content information for text-dependent speaker verification,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5344–5348.

[31] S. Sreekanth, S. M. Rafi B, K. S. R. Murty, and S. Bhati, ''Speaker embedding extraction with virtual phonetic information,'' in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2019, pp. 1–5.

[32] Y Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, ''Domain-adversarial training of neural networks,'' *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, Jan. 2016.

[33] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, ''On the usage of phonetic information for text-independent speaker embedding extraction,'' in *Proc. Interspeech*, 2019, pp. 1148–1152.

[34] N. Tawara, A. Ogawa, T. Iwata, M. Delcroix, and T. Ogawa, ''Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6799–6803.

[35] S. Sabour, N. Frosst, and G. E. Hinton, ''Dynamic routing between capsules,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–8.

[36] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, ''Dynamic neural networks: A survey,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, Nov. 2022.

[37] S. Teerapittayanon, B. McDanel, and H. T. Kung, ''BranchyNet: Fast inference via early exiting from deep neural networks,'' in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2464–2469.

[38] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, ''Adaptive neural networks for efficient inference,'' in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 527–536.

[39] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Weinberger, ''Multi-scale dense networks for resource efficient image classification,'' in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–11.

[40] A. Graves, ''Adaptive computation time for recurrent neural networks,'' 2016, *arXiv:1603.08983*.

[41] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, and J. E. Gonzalez, ''SkipNet: Learning dynamic routing in convolutional networks,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 409–424.

[42] A. Veit and S. Belongie, ''Convolutional networks with adaptive inference graphs,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–18.

[43] L. Liu and J. Deng, ''Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1–10.

[44] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris, ''BlockDrop: Dynamic inference paths in residual networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8817–8826.

[45] Y. Bengio, N. Léonard, and A. Courville, ''Estimating or propagating gradients through stochastic neurons for conditional computation,'' 2013, *arXiv:1308.3432*.

[46] K. Cho and Y. Bengio, ''Exponentially increasing the capacity-to-computation ratio for conditional computation in deep learning,'' 2014, *arXiv:1406.7362*.

[47] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup, ''Conditional computation in neural networks for faster models,'' 2015, *arXiv:1511.06297*.

[48] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, ''Adaptive mixtures of local experts,'' *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[49] D. Eigen, M. Ranzato, and I. Sutskever, ''Learning factored representations in a deep mixture of experts,'' 2013, *arXiv:1312.4314*.

[50] N. Shazeer, K. Fatahalian, W. R. Mark, and R. T. Mullapudi, ''HydraNets: Specialized dynamic architectures for efficient inference,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8080–8089.

[51] S. Cai, Y. Shu, and W. Wang, ''Dynamic routing networks,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3587–3596.

[52] J. Schmidhuber, ''Learning to control fast-weight memories: An alternative to dynamic recurrent networks,'' *Neural Comput.*, vol. 4, no. 1, pp. 131–139, Jan. 1992.

[53] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, ''Dynamic filter networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 667–675.

[54] D. Ha, A. Dai, and Q. V. Le, ''HyperNetworks,'' 2016, *arXiv:1609.09106*.

[55] N. Ma, X. Zhang, J. Huang, and J. Sun, ''WeightNet: Revisiting the design space of weight networks,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 776–792.

[56] A. W. Harley, K. G. Derpanis, and I. Kokkinos, ''Segmentation-aware convolutional networks using local attention masks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5048–5057.

[57] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, ''Pixel-adaptive convolutional neural networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11158–11167.

[58] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, ''CondConv: Conditionally parameterized convolutions for efficient inference,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1307–1318.

[59] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, ''Dynamic convolution: Attention over convolution kernels,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11027–11036.

[60] J. Z. Esquivel, A. C. Vargas, P. L. Meyer, and O. Tickoo, ''Adaptive convolutional kernels,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jan. 2019, pp. 1–8.

[61] C. Chen and Q. Ling, ''Adaptive convolution for object detection,'' *IEEE Trans. Multimedia*, vol. 21, no. 12, pp. 3205–3217, Dec. 2019.

[62] D. Shen, M. R. Min, Y. Li, and L. Carin, ''Learning context-sensitive convolutional filters for text processing,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1839–1848.

[63] B.-J. Choi, J.-H. Park, and S. Lee, ''Adaptive convolution for text classification,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2475–2485.

[64] B. Gu, W. Guo, L. Dai, and J. Du, ''An adaptive X-vector model for text-independent speaker verification,'' in *Proc. Interspeech*, 2020, pp. 1506–1510.

[65] B. Gu and W. Guo, ''Dynamic convolution with global-local information for session-invariant speaker representation learning,'' *IEEE Signal Process. Lett.*, vol. 29, pp. 404–408, 2022.

[66] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, ''In defence of metric learning for speaker recognition,'' in *Proc. Interspeech*, Oct. 2020, pp. 2977–2981.

[67] Y. Kwon, H. Heo, B. Lee, and J. S. Chung, ''The ins and outs of speaker recognition: Lessons from VoxSRC 2020,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5809–5813.

[68] H. Soo Heo, B.-J. Lee, J. Huh, and J. Son Chung, ''Clova baseline system for the VoxCeleb speaker recognition challenge 2020,'' 2020, *arXiv:2009.14153*.

[69] D. Ulyanov, A. Vedaldi, and V. Lempitsky, ''Instance normalization: The missing ingredient for fast stylization,'' 2016, *arXiv:1607.08022*.

[70] Z. N. Karam, W. M. Campbell, and N. Dehak, ''Towards reduced false-alarms using cohorts,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 4512–4515.

[71] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications," in *Proc. Interspeech*, Aug. 2011, pp. 2365–2368.

[72] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech*, Aug. 2017, pp. 1487–1491.

[73] S. Yadav and A. Rai, "Learning discriminative features for speaker identification and verification," in *Proc. Interspeech*, 2018, pp. 2237–2241.

[74] N. Li, D. Tuo, D. Su, Z. Li, D. Yu, and A. Tencent, "Deep discriminative embeddings for duration robust speaker verification," in *Proc. Interspeech*, Sep. 2018, pp. 2262–2266.

[75] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, Aug. 2017, pp. 999–1003.

[76] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, Aug. 2017, pp. 1517–1521.

[77] G. Bhattacharya, M. J. Alam, V. Gupta, and P. Kenny, "Deeply fused speaker embeddings for text-independent speaker verification," in *Proc. Interspeech*, Sep. 2018, pp. 3588–3592.

[78] F. A. Rezaur rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5359–5363.

[79] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," 2018, *arXiv:1803.10963*.

[80] J.-W. Jung, S.-B. Kim, H.-J. Shim, J.-H. Kim, and H.-J. Yu, "Improved RawNet with feature map scaling for text-independent speaker verification using raw waveforms," in *Proc. Interspeech*, Oct. 2020, pp. 1496–1500.

[81] 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," *STIN*, vol. 93, p. 27403, Feb. 1993.

[82] N. C. Ward and D. R. Dersch, "Text-independent speaker identification and verification using the TIMIT database," in *Proc. 5th Int. Conf. Spoken Lang. Process. (ICSLP)*, Nov. 1998, pp. 1–4.

[83] J. Gudnason and M. Brookes, "Voice source cepstrum coefficients for speaker identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 4821–4824.

[84] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," in *Proc. Interspeech*, Aug. 2017, pp. 1353–1357.

[85] S. O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Proc. Interspeech*, Sep. 2019, pp. 1483–1487.

**SEONG-HU KIM** received the B.S. and M.S. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include text-independent speaker identification, text-independent speaker verification, and sound event detection.

**HYEONUK NAM** received the B.S. and M.S. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include automatic speech recognition, speech dereverberation, semi-supervised sound event detection, and sound event localization and detection.

**YONG-HWA PARK** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 1991, 1993, and 1999, respectively. In 2000, he joined the Aerospace Department, University of Colorado at Boulder, as a Research Associate. From 2003 to 2016, he was with the Visual Display Division, Samsung Electronics, and the Samsung Advanced Institute of Technology (SAIT), as a Research Master in the field of micro-optical systems with applications to imaging and display systems. Since 2016, he has been an Associate Professor of noise and vibration control plus (NOVIC+) with the Department of Mechanical Engineering, KAIST, devoted to research on vibration, acoustics, vision sensors, and recognitions for human–machine interactions. His research interests include structural vibration, event/condition recognition from sound and vibration signatures utilizing AI, blood pressure and health monitoring sensors, 3D sensors, and lidar for motion measurements. He is a Board Member of KSME, KSNVE, KSPE, and SPIE. He has been the Conference Chair of MOEMS and miniaturized systems in SPIE Photonics West, since 2013.