

# Layer-aware TDNN: Speaker Recognition Using Multi-Layer Features from Pre-Trained Models

Jin Sob Kim, Hyun Joon Park, Wooseok Shin, Juan Yun, Sung Won Han

*Department of Industrial and Management Engineering*

*Korea University*

*Seoul, Republic of Korea*

{jinsob, winddori2002, wsshin95, yunjuan, swhan}@korea.ac.kr

**Abstract**—Recent advances in self-supervised learning (SSL) on Transformers have significantly improved speaker verification (SV) by providing domain-general speech representations. However, existing approaches have underutilized the multi-layered nature of SSL encoders. To address this limitation, we propose the layer-aware time-delay neural network (L-TDNN), which directly performs layer/frame-wise processing on the layer-wise hidden state outputs from pre-trained models, extracting fixed-size speaker vectors. L-TDNN comprises a layer-aware convolutional network, a frame-adaptive layer aggregation, and attentive static pooling, explicitly modeling of the recognition and processing of previously overlooked layer dimension. We evaluated L-TDNN across multiple speech SSL Transformers and diverse speech-speaker corpora against other approaches for leveraging pre-trained encoders. L-TDNN consistently demonstrated robust verification performance, achieving the lowest error rates throughout the experiments. Concurrently, it stood out in terms of model compactness and exhibited inference efficiency comparable to the existing systems. These results highlight the advantages derived from the proposed layer-aware processing approach. Future work includes exploring joint training with SSL frontends and the incorporation of score calibration to further enhance verification state-of-the-art performance.

**Index Terms**—Speaker recognition, speaker verification, speech pre-trained model, multi-layer features, layer-aware processing.

## I. INTRODUCTION

Speaker verification (SV) authenticates an identity by extracting speaker-specific features from speech. The field has advanced significantly due to deep neural network (DNN) improvements and enlarged data resources. One of the earliest, the  $x$ -vector [1] established a foundational DNN-based architecture upon handcrafted acoustic features (e.g., MFCCs). It comprises a stack of time-delay neural network (TDNN) [2] layers, temporal pooling, and utterance-level processing. The pipeline has been refined with deeper TDNNs [3]–[6], attention-based pooling strategies [6]–[8], and margin-based training loss functions [9], [10] for more discriminative speaker embeddings.

Meanwhile, the rise of self-supervised learning (SSL) transformers [11]–[13] has spurred recent research into using pre-trained representations for SV. Two main approaches have

This research was supported by the BK21 FOUR funded by the Ministry of Education of Korea and National Research Foundation of Korea. This research was also results of a study on the “Leaders in INdustry-university Cooperation 3.0” Project, supported by the Ministry of Education and National Research Foundation of Korea.

emerged: fine-tuning a pre-trained model into an end-to-end SV system [14], [15], or using SSL models as feature extractors for a downstream model [13], [16]–[18]. Notably, the SUPERB benchmark [16] introduced a weighted sum of hidden states from all layers to produce downstream features. [13] and [17] has achieved state-of-the-art SV performance by combining the SUPERB with a powerful ECAPA-TDNN [6].

However, current methods for exploiting pre-trained models in SV have limitations. First, some still rely on the final layer’s output [14], [15], [18], while several recent analyses [13], [17] have shown that speaker cues are concentrated in lower layers. Second, most studies adopt the trivial layer aggregation from SUPERB [16]. This static summation cannot capture frame-level variability of inter-layer importance, and its scalar weights are fixed after training, which limits generalization.

To overcome these limitations, we propose a dedicated backend architecture that fully exploits multi-layer SSL representations. Our contributions are summarized as three-fold: (1) we introduce a layer-aware TDNN (L-TDNN) that operates directly on stacked hidden states to enrich speaker characteristics; (2) we devise a frame-adaptive attention pooling strategy for dynamic layer aggregation; and (3) we validated comprehensive experiments that the proposed strategy consistently and efficiently outperforms existing methods for leveraging speech pre-trained models for SV.

## II. RELATED WORK

### A. Leveraging Pre-trained Models for Speaker Verification

Starting from Wav2vec [19], SSL Transformers [11]–[13], [20] have become widely accepted in contemporary speech-processing research. These models aim to extract task-agnostic representations by directly processing raw waveforms and generally comprise convolution layers followed by a Transformer encoder. Efforts to leverage such pre-trained models for SV have followed two main routes.

The first approach builds an end-to-end verification system by fine-tuning the SSL encoder. Both [14] and [15] explored adapting Wav2vec 2.0 [11] for SV. [14] formed an utterance-level representation by simply averaging the Transformer outputs and trained the network jointly on language and speaker classification. [15] compared several pooling strategies for aggregating speaker information from the output sequence and

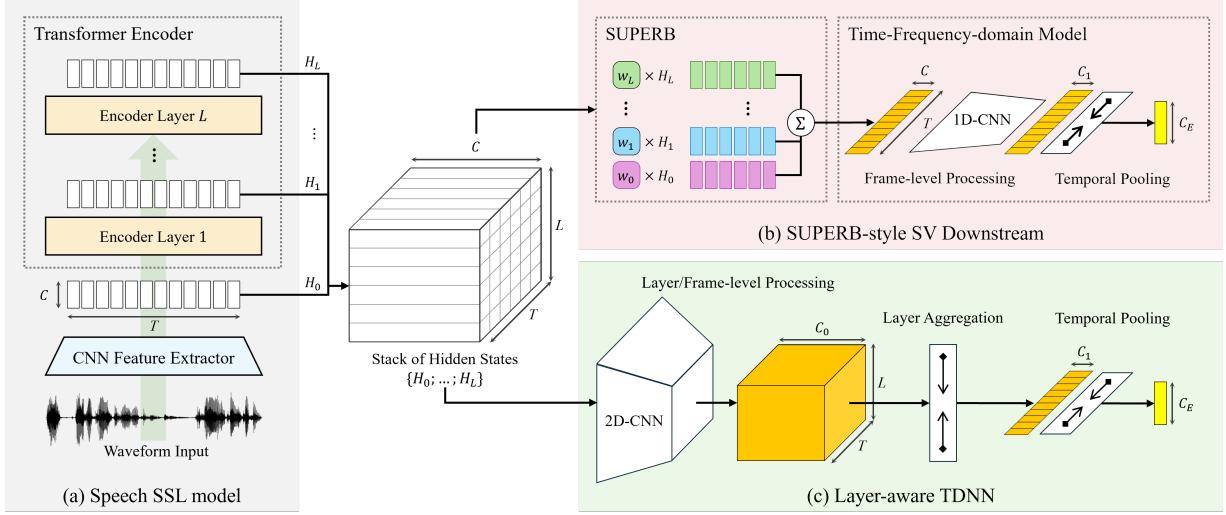


Fig. 1: Comparison of the speaker verification pipeline leveraging the multi-layer features from the speech SSL model.

proposed the insertion of a constant class (*CLS*) token, which is inspired by BERT [21].

The other approach treats the SSL encoder as a frontend feature extractor so that an SV downstream model processes its output. For example, [18] proposed a backend model, comprising two TDNN layers, statistic pooling, and a maxout linear layer, on top of Wav2vec 2.0 [11]. Although the proposal of SUPERB [16] was not exclusive to SV, its idea of combining hidden states from each layer of the pre-trained model inspired many subsequent studies. While SUPERB adopted the *x*-vector [1] to process a weighted summation of layer-wise representations, [13] and [17] considered a more powerful off-the-shelf downstream architecture, ECAPA-TDNN [6], to achieve strong verification performances.

#### B. Limitations of Prior Works and Preliminaries

As surveyed above, recent SV studies have shifted to the SSL paradigm, guaranteeing faster convergence and strong downstream performance. A diverse array of strategies has been discussed to exploit the pre-trained encoders, yet these approaches still face notable limitations.

While end-to-end approaches represent the speaker solely with the final layer output, the layer-wise probing across diverse SSL models [13], [17], [22], [23] demonstrated that speaker cues are prominent at lower layers. Adapting the pre-trained encoder in an end-to-end manner, therefore, starts at a disadvantage for speaker discrimination. SUPERB-based systems mitigate this issue by incorporating multi-layer hidden states. However, its static and global-constant aggregation ignores frame-level variability and limits the capability of layer-wise representation.

These findings lead to motivation for the next step for SV in leveraging pre-trained models, a backend that can actively process the entire stack of layer-wise hidden states. Conventional downstream networks, designed for the time-frequency domain, are not equipped to handle this additional layer

dimension. Therefore, in this study, we introduce a dedicated backend for SSL encoders, which refines speaker features across layers and frames, and also dynamic aggregation of each dimension.

### III. SPEAKER EXTRACTION USING SSL TRANSFORMERS

Fig. 1 provides an overview of the processing pipelines discussed in this section. (a) draws the common architecture of contemporary speech SSL models such as Wav2vec 2.0 [11], HuBERT [12], and WavLM [13]. A convolutional (CNN) feature extractor first produces a latent representation  $H_0 \in \mathbb{R}^{C \times T}$  directly from the raw waveform, then Transformer layers  $\{1, \dots, L\}$  process the feature. The hidden states from each layer are stacked, forming the initial input tensor  $\{H_0; \dots; H_L\} \in \mathbb{R}^{C \times T \times L}$  for the SV downstream.

Afterwards, parts (b) and (c) demonstrate how the SUPERB-based downstream and the proposed L-TDNN deal with the given tensor, respectively. As (b) illustrates, earlier studies opted for the conventional time-frequency-based backends to process the SSL model features; therefore, they adopted the static weighted summation strategy to integrate the layer dimension in advance. On the other hand, (c) depicts the proposed L-TDNN transforming the tensor into a speaker embedding, where we design the network to process the input tensor directly. L-TDNN comprises three stages: a convolutional processing network at the layer and frame level, a layer aggregation layer, and a temporal pooling layer.

#### A. Layer and Frame-level Processing Network

We extend the architecture from one of the powerful downstream models, ECAPA-TDNN [6], so that the network operates on a two-dimensional feature map. SE-Res2Block, proposed from ECAPA-TDNN, benefits from combining the Squeeze-and-Excitation (SE) [24] and the Res2Net [25] module that processes multi-scale features through the hierarchical residual connections. Moreover, the dense connection [26] of

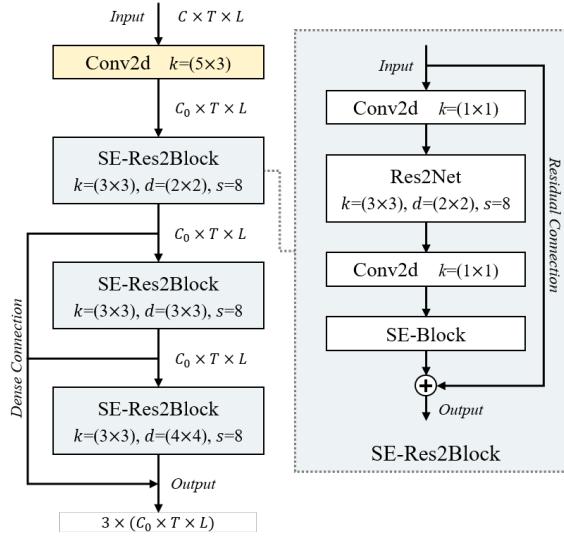


Fig. 2: Details of the layer/frame-level processing network.

each SE-Res2Block enables the shallow layers to contribute to a stronger foundation of speaker embedding. Fig. 2 depicts the details of the convolutional network and its feature maps.  $C$ ,  $T$ , and  $L$  denote the hidden size, the number of frames, and the number of SSL model hidden layers composing the input tensor  $\{H_0, \dots, H_L\}$ .  $k$ ,  $d$ , and  $s$  are arguments for the convolutional operations, which are the kernel size, dilation, and scale at the Res2Net module, respectively. During the expansion, we adjust the hidden size of the convolutional topology,  $C_0 = 256$ , to be smaller than the original's, keeping the number of parameters in L-TDNN comparable to SUPERB-style approaches [13], [17].

### B. Frame-adaptive Layer Aggregation

The following steps involve aggregating and pooling the feature map into a single vector representation of the speaker embedding. To better exploit the rich information across multiple layers, we explore a more advanced strategy for aggregating layer-wise speaker cues than static weighted summation. Inspired by one designed for multi-sequence aggregative processing [27], we devise a layer pooling layer based on SE [24] combined with multi-head projection [28].

Fig. 3 illustrates the proposed layer-aggregation strategy, where the entire sequence of operations is shared across frames, and we achieve the frame-dependent usage of the layer dimension. Given an output  $X \in \mathbb{R}^{3C_0 \times T \times L}$  from the preceding convolution network, the pooling process comprises three steps. First, we project the channel dimension with a learnable matrix  $W_{in} \in \mathbb{R}^{C_k \times 3C_0}$ :

$$x = W_{in} \cdot X \quad (1)$$

where  $x \in \mathbb{R}^{C_k \times T \times L}$  represents one of the  $H$  head projections. In this study, we set  $H = 8$  and  $C_k = \frac{3C_0}{H}$ .

Then, we compute layer-wise weights through the SE operation for each head. We start with taking the maximum and mean over the latent dimension,  $x_{\max}, x_{\text{mean}} \in \mathbb{R}^{L \times T}$ . These

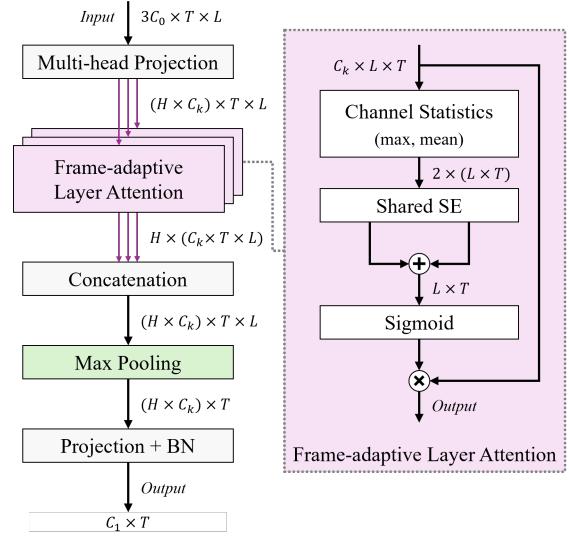


Fig. 3: Layer aggregation based on the MCA [27] strategy.

statistics pass through the learnable parameters  $W_{sq} \in \mathbb{R}^{r \times L}$  and  $W_{sq} \in \mathbb{R}^{L \times r}$  as:

$$\begin{aligned} \alpha_z &= W_{ex} \cdot \text{ReLU}(W_{sq} \cdot z) \\ \alpha &= \sigma(\sum_z \alpha_z), \quad z \in \{x_{\max}, x_{\text{mean}}\} \end{aligned} \quad (2)$$

where  $r = \frac{1}{2}L$ ,  $\sigma(\cdot)$  denotes the sigmoid activation that maps latent values within 0–1, and  $\alpha \in \mathbb{R}^{L \times T}$  implies the layer importance at each frame.

At last, we pool the most salient cues over layers by pooling the maximum after applying the weights.

$$x = \max_L(\alpha \odot x). \quad (3)$$

As above, we define the process within a single head, followed by head-wise concatenation and projection through the parameter  $W_{out} \in \mathbb{R}^{C_1 \times (H \times C_k)}$  where  $C_1 = 512$ .

### C. Attentive Statistic Pooling

The self-attention mechanism has proven to be successful in aggregating speaker embeddings from a sequence of frame-level features [6]–[8]. We adopt the strategy of ECAPA-TDNN [6], which uses the concept of the weighted mean and standard deviation for channel-dependent statistics.

Given the input  $X \in \mathbb{R}^{C_1 \times T}$ , the attention mechanism estimates  $\alpha \in \mathbb{R}^{C_1 \times T}$ , which implies the importance of each frame for its channel-wise statistics. Therefore, we compute the weighted statistics by:

$$\begin{aligned} \hat{\mu} &= \sum_t^T \alpha_{c,t} \cdot X_{c,t} \\ \hat{\sigma} &= \sqrt{\sum_t^T \alpha_{c,t} \cdot (X_{c,t})^2 - (\hat{\mu}_c)^2}. \end{aligned} \quad (4)$$

We concatenate the attention-weighted statistics  $\tilde{\mu}, \tilde{\sigma} \in \mathbb{R}^C$ , and the linear transformation  $W \in \mathbb{R}^{2C_1 \times C_E}$  and batch normalization follow, where  $C_1 = 512$  and  $C_E = 192$ .

TABLE I: Evaluation with direct fine-tuning approaches in diverse training environments

Verification System	VCTK			LibriSpeech			VoxCeleb1			VoxCeleb2		
	EER*	EER	minDCF									
<b>Input: MFCC</b>												
<i>x</i> -vector	16.15	16.22	0.898	8.08	7.99	0.468	12.07	12.08	0.742	7.94	7.97	0.490
ECAPA-TDNN	5.44	5.35	0.585	2.45	1.91	0.117	5.52	5.28	0.486	2.44	2.34	0.154
* BASE models, pre-trained on 960 hrs (LibriSpeech)												
<b>Wav2vec 2.0</b>												
Temporal mean	6.64	6.64	<b>0.758</b>	3.03	2.54	0.174	7.88	7.83	0.621	4.63	4.71	0.322
[CLS] insertion	9.77	9.46	0.980	3.40	2.79	0.175	3.73	3.69	0.384	<b>2.25</b>	2.27	0.187
<b>L-TDNN</b> (proposed)	<b>3.51</b>	<b>3.58</b>	0.773	<b>1.83</b>	<b>1.16</b>	<b>0.095</b>	<b>2.58</b>	<b>2.38</b>	<b>0.243</b>	<b>2.25</b>	<b>2.19</b>	<b>0.144</b>
<b>HuBERT</b>												
Temporal mean	5.56	5.44	<b>0.591</b>	26.12	26.17	0.717	32.81	33.73	0.888	32.72	31.17	0.713
[CLS] insertion	24.02	24.20	0.997	12.21	12.03	0.772	22.84	22.65	0.991	15.59	15.48	0.933
<b>L-TDNN</b>	<b>3.78</b>	<b>3.84</b>	0.599	<b>1.38</b>	<b>0.95</b>	<b>0.062</b>	<b>2.42</b>	<b>2.23</b>	<b>0.211</b>	<b>2.00</b>	<b>1.95</b>	<b>0.135</b>
<b>WavLM</b>												
Temporal mean	4.63	4.59	<b>0.451</b>	27.29	27.20	0.773	33.01	33.93	0.890	30.11	29.53	0.798
[CLS] insertion	25.80	25.66	0.995	12.07	11.89	0.794	20.99	20.85	0.989	9.89	9.81	0.728
<b>L-TDNN</b>	<b>3.43</b>	<b>3.55</b>	0.555	<b>1.61</b>	<b>1.21</b>	<b>0.081</b>	<b>2.15</b>	<b>1.96</b>	<b>0.218</b>	<b>1.94</b>	<b>1.88</b>	<b>0.121</b>

## IV. EXPERIMENT

### A. Datasets and Implementation Details

We evaluated our systems on multiple datasets to cover diverse training scenarios. VCTK CSTR Corpus (VCTK) [29] provides clean recordings from 108 English speakers. LibriSpeech [30] contains about 1,000 hours of speech from 2,484 speakers. Finally, the VoxCeleb 1 & 2 datasets [31], [32] offer speech data within a variety of acoustic environments and noises by 1,369 and 6,152 speakers, respectively. All audio was resampled to 16kHz.

Our model was trained with AAM-softmax [10] with scale 30 and margin 0.2. Adam optimizer [33] was adopted with a one-cycle learning rate (*lr*) scheduling [34] with the maximum *lr* = 0.003 using an initial 10% warmup phase. Minibatches comprise 128 samples, and each is truncated to 3s. We implemented data augmentation to drop a random span from the SSL model output, inspired by SpecAugment [35] and Patchout [36]. On evaluation, we used cosine similarity.

Throughout our experiments, SSL models were kept frozen while they served as feature extractors, and no score calibration was applied. This setup aims to isolate and validate the effectiveness of the proposed layer-aware processing on leveraging pre-trained features. We further discuss the joint training of the SSL frontend and the speaker backend and also post-processing techniques in Section V.

### B. Evaluation Metrics

We use two general SV evaluation metrics: equal error rate (EER) and minimum detection cost function (minDCF), with minDCF parameters set to  $C_{\text{Miss}} = 1$  and  $C_{\text{FA}} = 1$ , and  $P_{\text{target}} = 0.01$ . Both work by finding optimal decision threshold based on the similarity scores of a given evaluation set.

For a more practical evaluation, we introduce an additional measure, EER\*, where the test set is unseen during threshold setting. As below, EER\* evaluates the test set ( $\mathcal{Z}$ ) using a

fixed threshold ( $\tau$ ) derived from the EER of the validation set ( $\mathcal{U}$ ). It takes the mean of the resulting test set FAR and FRR.

$$\text{EER}^* = \frac{\text{FAR}_{\mathcal{Z}}(\tau) + \text{FRR}_{\mathcal{Z}}(\tau)}{2} \quad (5)$$

s.t.  $\text{EER}_{\mathcal{U}} = \text{FAR}_{\mathcal{U}}(\tau) = \text{FRR}_{\mathcal{U}}(\tau)$

### C. Baselines and Pre-trained SSL Models

We established two baseline groups that use pre-trained models. The first was to fine-tune an SSL model and pool a speaker embedding from the final hidden layer using non-parameterized ways. Either temporal mean pooling [14] or a CLS token insertion [15] was adopted. The second group followed the SUPERB pipeline, which uses a weighted sum of multi-layer features to produce an input for the downstream SV model. We paired this strategy with *x*-vector [1] and ECAPA-TDNN [6]. Experiments were conducted using three representative speech SSL models: Wav2vec 2.0 [11], HuBERT [12], and WavLM [13].

### D. Experimental Results

#### 1) Comparison with End-to-end Fine-tuning Approaches:

Table I compares L-TDNN against two end-to-end fine-tuning approaches as well as traditional MFCC-based verification systems. Across all three SSL encoders and every evaluation corpus, L-TDNN consistently demonstrates superior performance, achieving the lowest EER and EER\* in all conditions.

In contrast, the fine-tuning baselines show poor generalization, exhibiting instability with SSL Transformers other than Wav2vec 2.0, likely due to differing pre-training objectives. Notably, the temporal mean pooling [14] often performed worse than the MFCC-based ECAPA-TDNN, while the CLS insertion [15] has only proved highly limited model and dataset combinations. These results reveal the stability of leveraging multi-layer features from pre-trained models, and vice versa, the high sensitivity of naive fine-tuning while relying on the final layer output. L-TDNN's stable and superior

TABLE II: Comparison with SUPERB-based SV systems

Verification System	VoxCeleb1			VoxCeleb2		
	EER*	EER	minDCF	EER*	EER	minDCF
* BASE models, pre-trained on 960 hrs (LibriSpeech)						
<b>Wav2vec 2.0</b>						
<i>x</i> -vector	4.64	4.65	0.468	4.08	3.95	0.351
ECAPA-TDNN	3.24	2.82	0.346	2.42	2.33	0.157
<b>L-TDNN (proposed)</b>	<b>2.58</b>	<b>2.38</b>	<b>0.243</b>	<b>2.25</b>	<b>2.19</b>	<b>0.144</b>
<b>HuBERT</b>						
<i>x</i> -vector	4.00	3.97	0.423	3.66	3.43	0.305
ECAPA-TDNN	3.18	2.53	0.309	2.24	2.06	0.149
<b>L-TDNN</b>	<b>2.42</b>	<b>2.23</b>	<b>0.211</b>	<b>2.00</b>	<b>1.95</b>	<b>0.135</b>
<b>WavLM</b>						
<i>x</i> -vector	4.01	3.97	0.428	3.73	3.61	0.325
ECAPA-TDNN	2.55	2.15	0.257	2.07	2.05	0.137
<b>L-TDNN</b>	<b>2.15</b>	<b>1.96</b>	<b>0.218</b>	<b>1.94</b>	<b>1.88</b>	<b>0.121</b>
* BASE model, pre-trained on 94K hrs <sup>†</sup>						
<b>WavLM+</b>						
ECAPA-TDNN	2.21	1.84	0.217	1.79	1.72	0.094
<b>L-TDNN</b>	<b>1.93</b>	<b>1.62</b>	<b>0.198</b>	<b>1.70</b>	<b>1.63</b>	<b>0.092</b>
* LARGE model, pre-trained on 94K hrs <sup>†</sup>						
<b>WavLM</b>						
ECAPA-TDNN	2.19	1.78	0.213	1.79	1.72	0.110
<b>L-TDNN</b>	<b>1.85</b>	<b>1.51</b>	<b>0.167</b>	<b>1.54</b>	<b>1.61</b>	<b>0.104</b>

<sup>†</sup>Libri-Light, GigaSpeech, and VoxPopuli

performance supports its approach, benefiting from diverse types of speech SSL models.

2) **Comparison with SUPERB-based Approaches:** Table II compares L-TDNN against SUPERB-based baselines paired with *x*-vector [1] and ECAPA-TDNN [6] backends, using various scales of SSL frontends. L-TDNN surpasses the baselines across all evaluation metrics, regardless of the scale of a pre-trained encoder. On average, L-TDNN achieves an EER\* improvement of 45% over the *x*-vector baseline and 14% over the stronger ECAPA-TDNN baseline.

Furthermore, L-TDNN demonstrates strong scalability as the pre-training data and model capacity increase. It maintains a significant performance margin over the ECAPA-TDNN with relative EER\* improvements of 9% (WavLM base+) and 15% (WavLM large); 9% and 11% on EER, respectively. It is encouraging that L-TDNN yields significant gains over the architecturally similar ECAPA-TDNN. This highlights the effectiveness of modeling inter-layer information from hidden states of pre-trained models for the downstream task.

3) **Analyses on Model Efficiency:** Fig. 4 analyzes the efficiency of L-TDNN against SUPERB-based systems by comparing (a) parameter counts and (b) inference latency. We measured the latency on “Vox-O” trials using a single NVIDIA RTX A6000 GPU, reporting the median. L-TDNN is the most parameter-efficient model, being approximately two-thirds the size of ECAPA-TDNN while delivering the better performance shown in Table II. Regarding inference speed, L-TDNN also holds a slight advantage over ECAPA-TDNN. Although the simpler *x*-vector backend is faster, it suffers a significant trade-off in verification performance. These results demonstrate that L-TDNN offers a competitive advantage of accuracy, compactness and efficiency in leveraging features from the pre-trained Transformers.

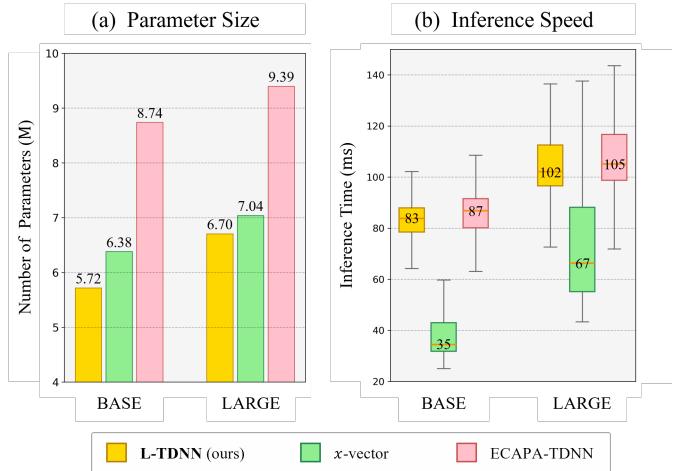


Fig. 4: Comparison of model efficiency across speaker embedding backends and SSL-frontend scales.

## V. CONCLUSION

### A. Limitations and Future Works

In this study, we froze the SSL model parameters to isolate our backend’s contribution. A key direction for future work is to explore joint training of the SSL frontend and the speaker backend, unleashing more powerful, task-specific representations. Additionally, we will investigate the integration of common post-processing techniques, such as score calibration [13], [37], to further push the model’s verification performance toward the state-of-the-art in SV benchmarks.

### B. Summary

This study introduced L-TDNN, a novel backend architecture designed to effectively leverage the multi-layered nature of pre-trained speech encoders for speaker verification. L-TDNN directly addresses the layer dimension, given the stack of hidden states produced from each layer of an SSL Transformer, which is previously overlooked. It is achieved through a dedicated structure comprising a layer-aware convolutional network, a frame-adaptive layer aggregation, and attentive temporal pooling, allowing it to robustly model inter-layer speaker characteristics.

Through extensive experiments, we demonstrated that L-TDNN consistently outperforms primary existing approaches such as end-to-end fine-tuning and SUPERB-based feature extraction. This strong performance was validated across diverse SSL models (Wav2vec 2.0, HuBERT, VoxCeleb 1 & 2). Moreover, we verified that L-TDNN provides these performance gains while also being more parameter-efficient and computationally faster than comparable backend architectures. These findings confirm L-TDNN as a robust, generalizable, and efficient solution, highlighting the benefits of dedicated layer-aware processing for speaker verification. Future work will incorporate jointly training of the SSL frontend and the speaker backend, as well as score calibration techniques, to further push the system beyond its current capabilities.

## 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 International Conference on Acoustics, Speech, and Signal Processing*, 2018, pp. 5329–5333.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[3] 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 International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 5796–5800.

[4] Y.-Q. Yu and W.-J. Li, “Densely connected time delay neural network for speaker verification,” in *Proc. Interspeech*, 2020, pp. 921–925.

[5] R. Zhang, J. Wei, W. Lu, L. Wang, M. Liu, L. Zhang, J. Jin, and J. Xu, “Aret: Aggregated residual extended time-delay neural networks for speaker verification,” in *Proc. Interspeech*, 2020, pp. 946–950.

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

[7] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” in *Proc. Interspeech*, 2018, pp. 3573–3577.

[8] K. Okabe, T. Koshinaka, and K. Shinoda, “Attentive statistics pooling for deep speaker embedding,” in *Proc. Interspeech*, 2018, pp. 2252–2256.

[9] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.

[12] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[13] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[14] Z. Fan, M. Li, S. Zhou, and B. Xu, “Exploring wav2vec 2.0 on speaker verification and language identification,” in *Proc. Interspeech*, 2021, pp. 1509–1513.

[15] N. Vaessen and D. A. van Leeuwen, “Fine-tuning wav2vec2 for speaker recognition,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2022, pp. 7967–7971.

[16] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, “SUPERB: Speech processing universal performance benchmark,” in *Proc. Interspeech*, 2021, pp. 1194–1198.

[17] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, “Large-scale self-supervised speech representation learning for automatic speaker verification,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2022, pp. 6147–6151.

[18] S. Novoselov, G. Lavrentyeva, A. Avdeeva, V. Volokhov, N. Khmelev, A. Akulov, and P. Leonteva, “On the robustness of wav2vec 2.0 based speaker recognition systems,” in *Proc. Interspeech*, 2023, pp. 3177–3181.

[19] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “Wav2vec: Unsupervised pre-training for speech recognition,” in *Proc. Interspeech*, 2019, pp. 3465–3469.

[20] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *Proc. 8th International Conference on Learning Representations*, 2020.

[21] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[22] T. Ashihara, M. Delcroix, T. Moriya, K. Matsuura, T. Asami, and Y. Iijima, “What do self-supervised speech and speaker models learn? new findings from a cross model layer-wise analysis,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024, pp. 10166–10170.

[23] S. Chen, Y. Wu, C. Wang, S. Liu, Z. Chen, P. Wang, G. Liu, J. Li, J. Wu, X. Yu, and F. Wei, “Why does self-supervised learning for speech recognition benefit speaker recognition?” in *Interspeech*, 2022, pp. 3699–3703.

[24] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[25] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, “Res2net: A new multi-scale backbone architecture,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[27] J. S. Kim, H. J. Park, W. Shin, D. Park, and S. W. Han, “WAY: Estimation of vessel destination in worldwide AIS trajectory,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 5961–5977, 2023.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[29] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, “Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92),” *University of Edinburgh. The Centre for Speech Technology Research*, pp. 271–350, 2019.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 5206–5210.

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

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

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd International Conference on Learning Representations*, 2015.

[34] L. N. Smith and N. Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in *Proc. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[35] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *Proc. Interspeech*, 2019, pp. 2613–2617.

[36] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Interspeech*, 2022, pp. 2753–2757.

[37] J. Thienpondt, B. Desplanques, and K. Demuynck, “The Idlab VoxSRC-20 submission: Large margin fine-tuning and quality-aware score calibration in DNN based speaker verification,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 5814–5818.