electronics

MDPI

Article

Res2Former: Integrating Res2Net and Transformer for a Highly Efficient Speaker Verification System

Defu Chen [1,2], Yunlong Zhou [1,2,*], Xianbao Wang [2], Sheng Xiang [1], Xiaohu Liu [1] and Yijian Sang [1]

[1] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China; defuchen@zjut.edu.cn (D.C.); xiangsheng@zjut.edu.cn (S.X.); 201806060508@zjut.edu.cn (X.L.); 221123030254@zjut.edu.cn (Y.S.)

[2] Binjiang Institute of Artificial Intelligence, Zhejiang University of Technology, Hangzhou 310056, China; wxb@zjut.edu.cn

* Correspondence: 221123030303@zjut.edu.cn

Abstract: Speaker verification (SV) is an exceptionally effective method of biometric authentication. However, its performance is heavily influenced by the effectiveness of the extracted speaker features and their suitability for use in resource-limited environments. Transformer models and convolutional neural networks (CNNs), leveraging self-attention mechanisms, have demonstrated state-of-the-art performance in most Natural Language Processing (NLP) and Image Recognition tasks. However, previous studies indicate that standalone Transformer and CNN architectures present distinct challenges in speaker verification. Specifically, while Transformer models deliver good results, they fail to meet the requirements of low-resource scenarios and computational efficiency. On the other hand, CNNs perform well in resource-constrained environments but suffer from significantly reduced recognition accuracy. Several existing approaches, such as Conformer, combine Transformers and CNNs but still face challenges related to high resource consumption and low computational efficiency. To address these issues, we propose a novel solution that enhances the Transformer model by introducing multi-scale convolutional attention and a Global Response Normalization (GRN)-based feed-forward network, resulting in a lightweight backbone architecture called the lightweight simple transformer (LST). We further improve LST by incorporating the Res2Net structure from CNN, yielding the Res2Former model—a low-parameter, high—precision SV model. In Res2Former, we design and implement a time-frequency adaptive feature fusion(TAFF) mechanism that enables fine-grained feature propagation by fusing features at different depths at the frame level. Additionally, holistic fusion is employed for global feature propagation across the model. To enhance performance, multiple convergence methods are introduced, improving the overall efficacy of the SV system. Experimental results on the VoxCeleb1-O, VoxCeleb1-E, VoxCeleb1-H, and Cn-Celeb(E) datasets demonstrate that Res2Former achieves excellent performance, with the Large configuration attaining Equal Error Rate (EER)/Minimum Detection Cost Function (minDCF) scores of 0.81%/0.08, 0.98%/0.11, 1.81%/0.17, and 8.39%/0.46, respectively. Notably, the Base configuration of Res2Former, with only 1.73M parameters, also delivers competitive results.

Keywords: embedded systems; speaker verification; transformer; res2net; machine learning

Academic Editor: Ping-Feng Pai

Received: 11 May 2025
Revised: 5 June 2025
Accepted: 16 June 2025
Published: 19 June 2025

Citation: Chen, D.; Zhou, Y.; Wang, X.; Xiang, S.; Liu, X.; Sang, Y. Res2Former: Integrating Res2Net and Transformer for a Highly Efficient Speaker Verification System. *Electronics* **2025**, *14*, 2489. https://doi.org/10.3390/electronics14122489

1. Introduction

Speaker verification (SV) technology automatically determines whether an audio sample belongs to a specific individual by analyzing unique vocal characteristics, aiming to

recognize or confirm the speaker's identity [1,2]. Owing to its ability to accurately identify individuals, this technology has wide-ranging applications, including voice authentication for personal smart devices and enhancing the security of banking transactions and remote payments [3,4]. Furthermore, it is utilized in forensic investigations to identify suspects, thereby contributing to both the convenience and security of everyday life [5,6]. Particularly, with the proliferation of mobile devices and Internet of Things (IoT) applications, there's a growing demand for SV systems that are not only accurate but also computationally efficient and have a small memory footprint, enabling on-device authentication and personalization.

In the early development of SV, deep neural networks (DNNs) played a pivotal role. The initial d-vector approach [7] laid the foundation for the field by mapping speech segments into a fixed-length vector space to represent speaker characteristics. Building upon this, the X-vector method [8] introduced a time-delay neural network (TDNN) architecture and a global statistical pooling layer to generate fixed-length speaker embedding vectors, further enhancing the representation of speaker features. These advancements have had a profound impact on SV tasks.

With the continuous evolution of convolutional neural networks (CNNs), models such as ResNet [9], Res2Net [10], and ResNeXt [11] were successively introduced and widely adopted as backbone networks in SV systems. Building upon these advancements, architectures like ECAPA-TDNN [12] have gained attention for their ability to extract multi-scale features and enhance channel attention mechanisms, becoming mainstream in the field. Additionally, NeXt-TDNN [13], which leverages multi-scale hierarchical feature extraction and multi-layer feature aggregation, has further improved performance, underscoring the importance of deep feature fusion in SV. Nevertheless, despite these advancements, CNNs face challenges in integrating global context and modeling long-range dependencies. These limitations have driven researchers to explore new architectures.

In recent years, the Transformer architecture has demonstrated outstanding performance in the field of Natural Language Processing (NLP) due to its self-attention mechanism, which effectively handles long-range dependencies [14–16]. Building on this success, researchers have gradually applied Transformers to SV tasks. SAEP [17] was the first to introduce Transformers into SV, employing a two-layer stacked Transformer encoder and DNN to extract speaker embeddings. However, despite efforts in parameter optimization, the model's performance remained suboptimal because the shallow Transformer encoder failed to capture the rich feature representations necessary for optimal performance.

Building upon the initial application of Transformers in SV, subsequent models such as Wav2Vec2.0 [18], HuBERT [19], and WavLM [20] have incorporated multi-head decomposed attention pooling to aggregate frame-level representations into speaker embeddings, further advancing performance in SV tasks. These models use self-supervised learning techniques to extract generalized feature representations from large amounts of unlabeled speech data, thereby enhancing their generalization capabilities. To simultaneously address both global dependency modeling and local feature extraction, researchers have proposed Transformer models with multi-view self-attention mechanisms, leading to notable performance improvements. However, the increased complexity of these models often results in a large number of parameters, which raises computational and storage costs and limits their practical deployment.

To address the limitations of previous models, DT-SV [21] optimized the Transformer structure by introducing a learnable time-domain feature extractor (TDFE), which reduced the number of model parameters while attempting to maintain performance. However, the actual performance gains were limited. Building on this, the MFA-Conformer [22] model recently adopted the Conformer [23] structure as the backbone for speaker embedding extraction, significantly enhancing performance. The Conformer combines CNN

with the Transformer, retaining the Transformer's global dependency modeling ability while improving local information capture. This approach indicates that incorporating convolutional networks into Transformers to strengthen local information modeling is an effective strategy.

In summary, while Transformer architectures have advanced SV tasks, they still face challenges in effectively balancing global dependency modeling and local feature extraction. Future research may focus on designing more efficient hybrid models to further improve SV performance while reducing model complexity and computational costs.

With each new generation, Transformer-based models in SV build upon previous work, continuously improving and innovating. As a result of this progression, substantial improvements have been achieved in both the accuracy and efficiency of SV tasks. However, existing studies [24,25] indicate that the effectiveness of Transformers in speech verification often relies on complex pre-training processes and large model parameters, which may pose limitations in practical applications. Thus, a clear research gap exists for an SV architecture that synergistically combines the strengths of CNNs and Transformers to achieve state-of-the-art accuracy while concurrently addressing the critical needs for low model parameters, reduced computational complexity, and faster inference speeds, particularly for deployment in resource-limited environments. Existing hybrid models like Conformer still present challenges in terms of resource consumption.

Currently, efficiently applying CNN and Transformers to SV tasks remains challenging due to several key issues:

- Traditional CNNs are inefficient in extracting deep speech features because they struggle to capture long-range dependencies and global features, which limits the discriminative power of the feature representations they produce.
- Transformers, although excellent at modeling global dependencies, have a large number of parameters and are complex to train. Their high computational complexity makes them difficult to deploy in lightweight or resource-constrained environments.
- The complexity of both CNN and Transformers results in slower inference speeds. Significant computational overhead during inference limits their applicability in real-time scenarios that require rapid responses.

Thus, a clear research gap exists for an SV architecture that synergistically combines the strengths of CNNs and Transformers to achieve state-of-the-art accuracy while concurrently addressing the critical needs for low model parameters, reduced computational complexity, and faster inference speeds, particularly for deployment in resource-limited environments. Existing hybrid models like Conformer still present challenges in terms of resource consumption.

The contributions of this work are as follows:

- We propose a lightweight simple transformer (LST) architecture that retains the advantages of Transformers while significantly reducing both model parameters and computational complexity by simplifying the self-attention mechanism and optimizing the network design.
- We introduce Res2Former, an effective SV model that builds on LST by incorporating the Res2Net architecture. By leveraging the multi-scale feature extraction capability of Res2Net, Res2Former enhances its ability to capture fine-grained characteristics of speech signals, thereby improving the discriminative performance of speaker embeddings.
- We design feature processing strategies and time-frequency adaptive feature fusion(TAFF) mechanisms at different network depths. By introducing targeted feature processing methods at various layers and integrating attention mechanisms from both the time and frequency domains, we enhance the richness and discriminative power of the feature representations.

- By employing a combination of pre-training and large-margin fine-tuning strategies, we optimize pre-trained models and further improve model performance.

2. Method

2.1. Lightweight Simple Transformer (LST)

We have designed an LST structure that replaces the standard multi-head self-attention (MHSA) with a simplified multi-scale convolutional attention (MSCA) module and incorporates a feedforward network enhanced by Global Response Normalization (GRN). The goal is to reduce the model's computational complexity and training challenges while retaining the performance of the Transformer architecture and extracting richer input feature representations. As shown in Figure 1, the LST structure processes input features by first passing them through a Layer Normalization layer, followed by the MSCA module with a residual connection. Next, the output undergoes another Layer Normalization before being fed into the GRN-based feedforward network, which also includes a residual connection. The corresponding equations are as follows:

$$X' = \text{MSCA}(\text{LayerNorm}(X)) \tag{1}$$

$$X_{\text{MHSA}} = X + X' \tag{2}$$

$$X'' = \text{FFN}(\text{LayerNorm}(X')) \tag{3}$$

$$X_{\text{OUT}} = X_{\text{MHSA}} + X'' \tag{4}$$

where X' denotes the output of simplified multi-scale convolutional attention, X_{MHSA} is the output of multi-head attention with residual connection, X'' is the output of the feed-forward network, and X_{OUT} is the output of the LST.

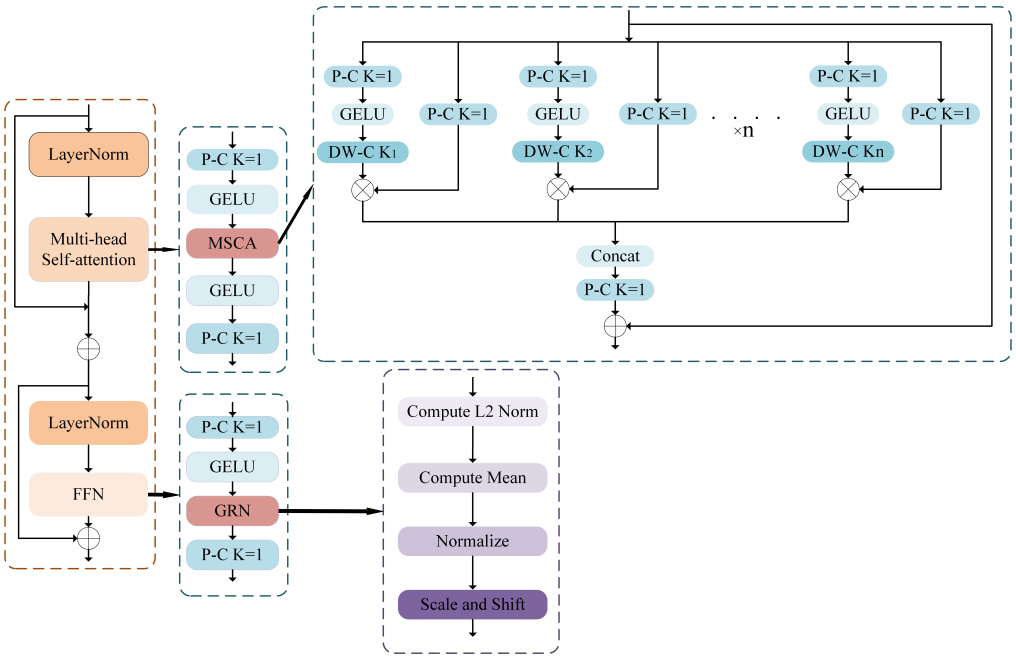

Figure 1. Lightweight Simple Transformer (LST) overall architecture based on multi-scale convolutional attention and Global Response Normalization (GRN) based feedforward Network.where P-C K = 1 denotes point-by-point convolution with convolution kernel 1, DW-C K = 1 denotes depthwise convolution with convolution kernel 1, and DW-C Ki denotes depthwise convolution with convolution kernel Ki.

2.1.1. Simplified Multi-Scale Convolutional Attention

Figure 1 illustrates our simplified multi-scale convolutional attention mechanism. Unlike the traditional Transformer that computes the similarity score matrix A using the standard self-attention formula, our proposed approach simplifies the self-attention mechanism by modulating the value V through convolutional features. Specifically, given an input feature tensor $X \in R^{B \times F \times T}$, where B is the batch size, F is the number of feature channels, and T is the time steps, we apply depthwise convolutions with different kernel sizes to mimic the multi-head attention mechanism. These multi-scale convolutions capture multi-level and multi-scale information from the input features, similar to how different attention heads in multi-head attention capture distinct information. By incorporating depthwise convolutions, we can significantly reduce computational complexity and parameter count without sacrificing the ability to model spatial relationships at both local and global scales. After obtaining the convolutional features, we fuse them element-wise with the original input X using the Hadamard product (element-wise multiplication). This step modulates the original features, emphasizing important feature dimensions. Additionally, we incorporate residual connections by adding the fused features back to the input features, ensuring stable gradient flow and efficient training. The mathematical expression for the above process is as follows:

$$A_i = W_i X + b_i \tag{5}$$

$$V_i = DW\text{-}Conv_{k \times k}(GELU(W_i X + b_i)) \tag{6}$$

$$X_{MSCA} = P\text{-}Conv_{1 \times 1}\left(X + Concat\left(\sum_{i=1}^{n} A_i \odot V_i\right)\right) \tag{7}$$

where W_i and b_i are the weights and biases, A_i is the simplified attention weight, V_i is the value matrix, $DW\text{-}Conv_{k \times k}$ represents the depthwise convolution with kernel size K, $P - Conv_{1 \times 1}$ represents the pointwise convolution with kernel size 1, and X_{MSCA} is the output of the simplified multi-scale convolution attention. The simplified multi-scale convolution attention mechanism associates each spatial position (F, T) with all pixels within a $k \times k$ square region centered around it. This approach not only retains sensitivity to local feature regions but also reduces computational complexity. Furthermore, channel-wise information interaction is achieved through linear layers, which enhances the model's expressive capacity and promotes synergy across different feature dimensions.

2.1.2. A Feedforward Network Based on Global Response Normalization

The Feed-Forward Network (FFN) is a core component of the Transformer encoder, primarily responsible for performing nonlinear processing and spatial transformations on features generated by the attention mechanism. This enables the model to capture complex nonlinear relationships between features. Traditionally, the FFN module is constructed using a Fully Connected Network (FCN), which can suffer from issues such as gradient vanishing and information loss. To address these problems, the GRN structure is introduced. By applying global response normalization to the main pathway channels on a per-sample basis, GRN enables flexible control of information flow, enhancing both the network's expressive power and training stability. Building upon GRN, we have constructed a fully connected network that mitigates the limitations of traditional FFN. For the input $X \in R^{B \times F \times T}$, the detailed formula for GRN is as follows: For each sample b and time step t, we first compute its global response (L2 norm), which is the L2 norm of the activation values across all feature channels at each time step:

$$R_{b,t} = \sqrt{\sum_{f=1}^{F} X_{b,f,t}^2} \tag{8}$$

where $R_{b,t}$ is the global response of the bth sample at time step t. To prevent numerical instability, a small regularization constant ϵ is added during the computation of the response to avoid division by zero. Typically, ϵ is a very small value, such as 10^{-6}. As a result, the normalized response is given by:

$$\hat{R}_{b,t} = \frac{R_{b,t}}{\sqrt{R_{b,t}^2 + \epsilon}} \tag{9}$$

The normalized response is then used to normalize the feature channels at each time step:

$$X_{GRN} = \frac{X_{b,f,t}}{\hat{R}_{b,t}} \tag{10}$$

where X_{GRN} is the output of the GRN normalization.To enhance the representational capacity of the normalized features, learnable scaling parameter γ and shifting parameter β are introduced. As a result, the formula can be expanded as follows:

$$X_{GRN} = \gamma_f \frac{X_{b,f,t}}{\hat{R}_{b,t}} + \beta_f \tag{11}$$

where γ and β are learnable parameters for each feature channel.

2.2. Res2Former Fusing LST Based on Res2Net Structure

2.2.1. Overall Overview of Res2Former

To enhance the modeling capability of both fine-grained frame-level features and global speech information, we introduce a Time-Frequency Attention Feature Fusion (TAFF) mechanism into the MSCA module of the LST framework. Additionally, we design a frame-level adaptive fusion module based on the Res2Net architecture, termed the Res2Former Block. Building upon this, we further propose a four-stage global time-frequency adaptive fusion strategy. Based on these enhancements, we construct a speaker verification framework, Res2Former, which achieves a favorable balance between recognition accuracy and computational efficiency. Figure 2 illustrates the overall structure of Res2Former.

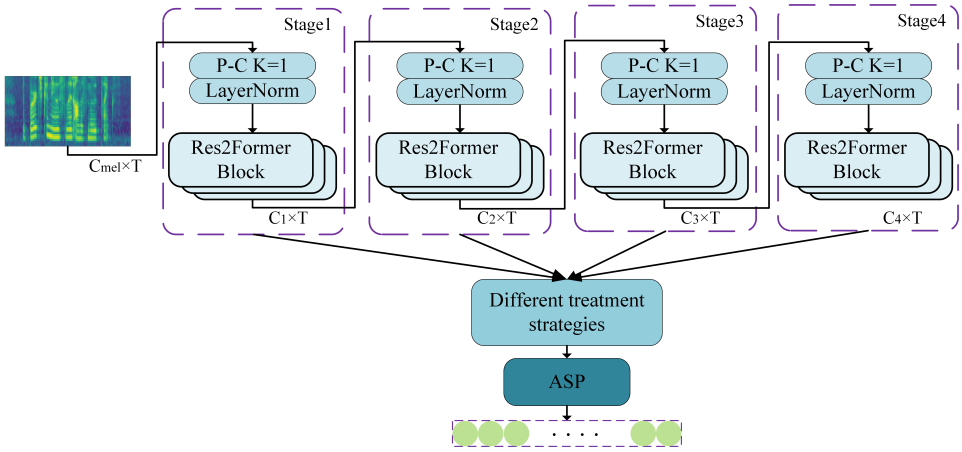

Figure 2. The overall architecture of Res2Former.

The Res2Former consists of four stages, each comprising the following two modules.

1. The Convolutional Normalization Layer (P-C K = 1 & LayerNorm) processes the input features through a convolutional layer with a kernel size of 1, followed by layer normalization. The formula is as follows:

$$X' = \text{LayerNorm}(\text{P-Conv}_{1\times1}(X)) \tag{12}$$

2. The core module of each stage is the Res2Former Block, which combines the multi-scale convolutional capability of Res2Net with the feature processing efficiency of the Transformer. The formula can be expressed as follows:

$$\begin{aligned} Y = {} & X' + \text{MSCA}\big(\text{LayerNorm}(X')\big) \\ & + \text{FFN}\big(\text{LayerNorm}\big(X' + \text{MSCA}\big(\text{LayerNorm}(X')\big)\big)\big) \end{aligned} \tag{13}$$

where X' is the output of the convolutional normalization layer and Y is the output of each Res2Former Block.

The number of output channels at each stage is a variable, denoted as C_1, C_2, C_3, and C_4. Different stages represent the model's extraction of features from low to high levels. After each stage, the extracted features are aggregated using various processing strategies. This module combines and filters the features from different stages to form the final feature representation:

$$Y_{\text{out}} = f(Y_1, Y_2, Y_3, Y_4) \tag{14}$$

where f denotes the feature processing strategy at different depths, and the explanation of f will be provided in subsequent sections. Y_{out} is the final output.Finally, all features processed through different strategies are fed into the Attentive Statistics Pooling (ASP) module, resulting in the speaker embedding vector:

$$Y_{\text{embedding}} = \text{ASP}(Y_{\text{out}}) \tag{15}$$

2.2.2. The Time-Frequency Adaptive Feature Fusion Mechanism

Attention-based methods have shown significant potential in modeling spatial relationships within the field of computer vision. By enabling models to focus on key features during recognition, suppress irrelevant ones, and capture global relationships among features, attention mechanisms enhance the feature representation capacity of CNN. Extending this concept to the time-frequency domain, attention methods can similarly model relationships across time and frequency. Leveraging these relationships, we construct a time-frequency weight matrix, where the weight values are positively correlated with the frequencies at each specific location. The specific structure is shown in Figure 3.

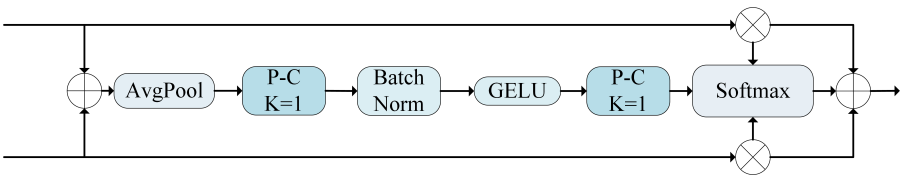

Figure 3. Time-frequency adaptive feature fusion(TAFF) mechanism.

To further improve the learning process within convolutional layers, we adopt the softmax function to generate the weight matrix. The use of softmax promotes competition among features, encouraging the convolutional layers to learn diverse representations. Using the generated weight matrix, we adjust the activation magnitudes of the original feature maps. Features at the same time-frequency location are scaled by the same weight,

whereas features at different time-frequency locations are scaled by different weights. The process is as follows:

$$s_{b,f} = \frac{1}{T}\sum_{t=1}^{T}(X+Y)_{b,f,t} \tag{16}$$

$$g_{b,f'} = \mathrm{GELU}\left(\mathrm{BatchNorm}\left(\sum_{f=1}^{F} W_{f',f}^{(1)} \cdot s_{b,f} + b_{f'}^{(1)}\right)\right) \tag{17}$$

$$v_{b,f'} = \mathrm{BatchNorm}\left(\sum_{f'=1}^{F'} W_{f',f}^{(2)} \cdot g_{b,f'} + b_{f'}^{(2)}\right) \tag{18}$$

$$\mathrm{att} = \frac{\exp(v_{b,f})}{\sum_{\kappa=1}^{F}\exp(v_{b,\kappa})} \tag{19}$$

$$O = (X \odot \mathrm{att}) + (Y \odot \mathrm{att}) \tag{20}$$

where W_i and b_i are the weights and biases, A_i is the simplified attention weight, and V_i is the value matrix, for the input features $X, Y \in R^{B\times F\times T}$ the output $s_{b,f}$ is obtained after global average pooling in the time and frequency dimensions, respectively, and for $s_{b,f}$ the output $s_{b,f'}$ is obtained by performing a convolutional and normalized to obtain $v_{b,f'}$ containing time and frequency information, convolve $v_{b,f'}$ again and apply Softmax function to obtain the time-frequency attention weight matrix att, and dot-multiply att with the inputs X, Y to obtain the output O.

2.2.3. Frame-Level Time-Frequency Adaptive Feature Fusion

Figure 4a illustrates our frame-level TAFF method. To capture finer-grained features and enhance local information interaction, we optimized the MSCA within the time-frequency domain using a top-down hierarchical feature processing approach. By adaptively integrating features across different scales, we achieved frame-level adaptive feature fusion.

$$Y_i = \begin{cases} P\text{-}DW\text{-}Conv_{K_i\times K_i}(X_i), & i = 1 \\ P\text{-}DW\text{-}Conv_{K_i\times K_i}(TAFF(X_i, Y_{i-1})), & i > 1 \end{cases} \tag{21}$$

where Y_i is the output of each block, and $P\text{-}DW\text{-}Conv_{K_i\times K_i}$ denotes that it is first subjected to a pointwise convolution with a kernel size of 1 and then to a depthwise convolution with a kernel size of K_i.

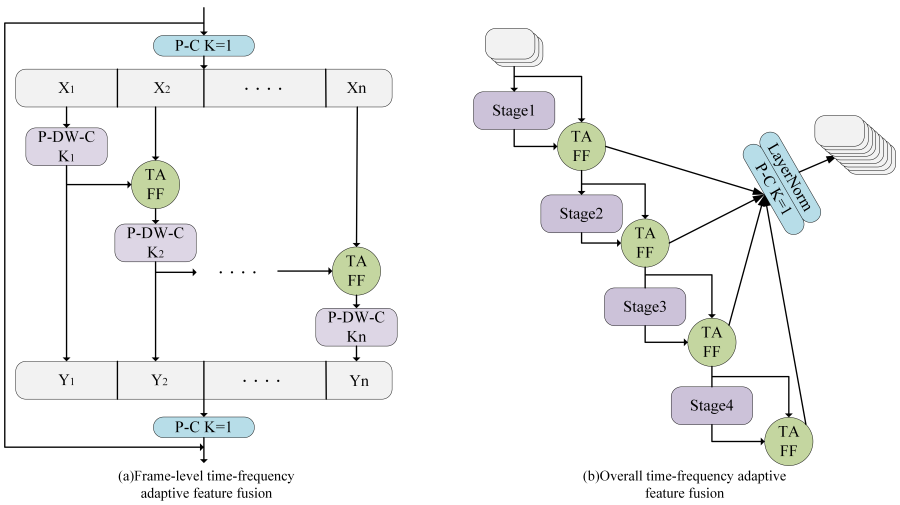

Figure 4. Frame-level and Overall time-frequency adaptive feature fusion mechanism.

2.2.4. Overall Time-Frequency Adaptive Feature Fusion

Figure 4b illustrates how the overall TAFF operates differently across network layers. In the lower layers of the neural network, neurons possess more constrained receptive fields, capturing fine-grained details. By contrast, neurons in higher layers capture broader context due to their larger receptive fields. This fusion mechanism seeks to improve the interaction of information by adjusting features across different temporal scales within the top-down process.

$$X_{\text{stage}_{i+1}} = \text{Stage}_i\left(X_{\text{stage}_i}\right) \tag{22}$$

$$Z = P\text{-}Conv_{1\times1}\left(LayerNorm\left(\sum_{i=1}^{3} TAFF\left(X_{\text{stage}_{i+1}}, X_{\text{stage}_i}\right)\right)\right) \tag{23}$$

where X_{stage} is the output of each $Stage$, $X_{\text{stage}_1} = X$, and Z is the overall output.

2.3. Feature Processing Strategies and ASP at Different Depths

Feature processing and fusion at different depths constitute essential components of feature engineering in deep learning, particularly when handling multimodal data or multi-scale features. This is equally applicable to SV systems. Whether using a CNN-based or Transformer-based architecture, feature processing strategies and fusion techniques at different depths significantly enhance system performance.

In order to enhance the feature discovery for different depths, we designed four different depths of feature processing methods for each STAGE whether TAFF and Concat, as shown in Figure 5. TypeI outputs the result following the application of overall TAFF at each stage. TypeII builds on TypeI by incorporating a weighted average prior to producing the output. TypeIII bypasses the TAFF at each stage and directly outputs the result. TypeIV omits feature processing entirely and outputs directly.

$$O_1 = \text{LayerNorm}(P\text{-}Conv_{1\times1}(\text{Concat}(TAFF_1, TAFF_2, TAFF_3, TAFF_4))) \tag{24}$$

$$O_2 = \text{WeightAvg}(\text{Concat}(TAFF_1, TAFF_2, TAFF_3, TAFF_4)) \tag{25}$$

$$O_3 = \text{LayerNorm}(P\text{-}Conv_{1\times1}(\text{Concat}(Stage_1, Stage_2, Stage_3, Stage_4))) \tag{26}$$

$$O_4 = Stage_4 \tag{27}$$

where O_1 is the output of TypeI, O_2 is the output of TypeII, O_3 is the output of TypeIII, and O_4 is the output of TypeIV.

To more effectively embed speech features into fixed-dimensional speaker representations, we employ the ASP method [26]. Traditional speaker embedding approaches typically use basic statistical pooling techniques, such as computing the mean and variance, to aggregate frame-level features. Nonetheless, these approaches face challenges in effectively capturing global context and the differing importance of input features. ASP introduces an attention mechanism that assigns different weights to each data point, allowing the model to more effectively capture long-term dependencies and emphasize important features. As a result, this method has been widely adopted in SV systems.

Specifically, ASP involves two key steps. First, an attention mechanism computes normalized scores a_t for each frame, assigning different weights based on the frame's significance. This is achieved using the following equation:

$$a_t = \text{Softmax}(V^T f(WH_t + b)) \tag{28}$$

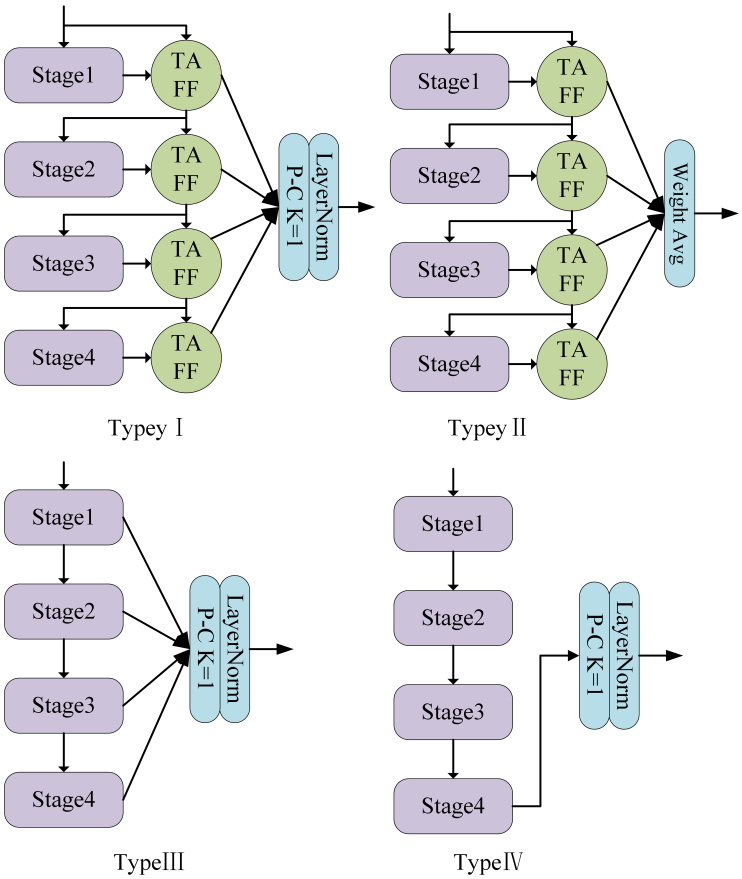

Figure 5. Four different feature processing methods.

Second, these weights are used to compute the mean $\tilde{\mu}$ and the standard deviation $\tilde{\sigma}$ from the frame-level features H_t, forming the final speaker embedding. The calculations are as follows:

$$\tilde{\mu} = \sum_{t=1}^{T} a_t H_t \tag{29}$$

$$\tilde{\sigma} = \sqrt{\sum_{t=1}^{T} a_t H_t^2 - \tilde{\mu}^2} \tag{30}$$

The output of the pooling layer is formed by concatenating the mean μ and the standard deviation σ along the channel dimension. Ultimately, a fully connected layer is used to derive a compact, low-dimensional representation of the speaker embedding.

$$\text{Speaker Embedding} = \text{Linear}(\text{Concat}(\tilde{\mu}, \tilde{\sigma})) \tag{31}$$

2.4. Loss Function and Large Margin Fine-Tuning

In fine-grained classification and verification tasks, the Additive Angular Margin Softmax (AAM-Softmax) loss function [27] is widely adopted and highly effective. It extends the standard Softmax loss by incorporating L2-normalized embeddings and introducing an angular margin regularization applied throughout the training process. This angular margin regularization motivates the network to enhance the angular separation between distinct classes, while also fostering intra-class compactness as the target class log-likelihood is estimated. Consequently, AAM-Softmax boosts the separability between

classes and strengthens compactness within each class, which is particularly beneficial for fine-grained tasks such as SV. The AAM-Softmax loss is defined by the following equation:

$$L_{\text{AAM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_{j,i})}}\right) \tag{32}$$

where $\theta_{y_i,i}$ denotes the angle between the feature vector x_i of the ith sample and the weight vector W_{y_i} of the correct class y_i, s is a scaling factor, and m is the additional angular margin.

The scaling factor s amplifies the output values of the Softmax function, thereby affecting the gradient magnitude during training. The angular margin m enhances the model's discriminative ability by increasing the angular distance between different classes. Generally, a larger value of m leads to greater inter-class separation and tighter intra-class compactness, thus improving overall model performance. However, setting a high angular margin can make model convergence difficult, while a small margin may lead to underfitting. To address this, we propose a dynamic fine-tuning strategy with an adaptive high-margin approach. Initially, we set the angular margin m to zero and gradually increase it to a small, predefined value to allow the network to converge. Subsequently, we fine-tune the network by progressively increasing the margin from this low value to a higher one. In addition, to ensure stable training, we enhanced data preprocessing and optimized the learning rate across different training stages.

3. Experimental Setup

3.1. Dataset

VoxCeleb1 [28], VoxCeleb2 [29], CN-Celeb1 and CN-Celeb2 [30,31] are widely used speech datasets for speaker recognition tasks. These datasets encompass speech data from various real-world scenarios, making them effective for handling complex and diverse conditions.

For the VoxCeleb experiments, we trained our model on the development set of VoxCeleb2, which includes 5994 speakers. The model was evaluated on the VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H evaluation sets, containing 40, 1251, and 1190 speakers, respectively.

Similarly, for the CN-Celeb experiments, we used the combined development sets of CN-Celeb1 and CN-Celeb2 for training, totaling 2785 speakers, and evaluated on CN-Celeb(E), which consists of 200 speakers.

To enhance system robustness, we applied three data augmentation techniques: online data augmentation [32], MUSAN [33], and RIR [34]. Additionally, during data preprocessing, we concatenated shorter utterances to ensure that each utterance used for training was at least 6 s long.

3.2. Evaluation Metrics

To assess performance, all experiments were evaluated using cosine similarity scoring. Common metrics in SV tasks were adopted, including Equal Error Rate (EER), Minimum Detection Cost Function (minDCF), and Real-Time Factor (RTF), with $P_{target} = 0.01$ and $C_{FA} = C_{Miss} = 1$. Real-time inference was conducted on an Intel Core i9-12900KF processor.

3.3. Model Configurations

In our experiments, we trained several models, including LST, Transformer [14], ECAPA-TDNN [12], Res2Net [10], MFA-Conformer [22], NeXt-TDNN [13], and Res2Former. Among these, LST and Res2Former are the models we developed, while the others serve as baseline models. Their detailed configurations are as follows:

LST: Based on the LST structure, this model integrates an MFA stacking structure with an ASP module. We experimented with varying numbers of blocks, each with a feature dimension of 256, yielding an output embedding dimension of 192.

Transformer: This model utilizes a Transformer backbone combined with the MFA stacking structure and ASP module. We tested different numbers of blocks, each with a 256-dimensional feature space, producing a final embedding of 192 dimensions.

ECAPA-TDNN: This model employs a 3-layer SE-Res2Block backbone, each layer with a feature dimension of 1024. An ASP module is used, resulting in an output embedding of 192 dimensions.

Res2Net: Divided into four stages, each stage uses a Res2NetBlock backbone with TSTP, producing a 192-dimensional embedding output.

MFA-Conformer: Built with six ConformerBlocks, this model integrates a 1/2 sampling rate and four attention heads with ASP. Each block has a feature dimension of 256, and the output embedding is 192-dimensional.

Next-TDNN: This model employs nine TS-ConvNeXt Block with a convolutional kernel size of (7, 65), using ASP. Each block has a feature dimension of 256, producing a final embedding of 192 dimensions.

Res2Former: Structured into four stages, each stage consists of B optimized LST structures. The feature dimension for each stage is denoted as C(C_1, C_2, C_3, C_4), and kernel sizes are represented as K($K_1 = 5$, $K_2 = 9$, $K_3 = 11$, $K_4 = 11$). The model uses an ASP module, with a 192-dimensional embedding as output.

3.4. Training Options

We used the PyTorch (version 1.11.0) framework to build and train the LST, Res2Former, and baseline networks. The input features consisted of 80-dimensional Fbank acoustic features, extracted from each 25-ms frame with a 10-ms window shift, while the sampling rate was maintained at 16,000 Hz. All experiments employed AAM-Softmax loss and the Adam optimizer, with scaling factors set to 30 and weight decay set to 1×10^{-6}. The training procedure was split into two stages: pre-training and fine-tuning with a large margin.

Pre-training: Three-second fixed-length segments were randomly selected, with a batch size of 256, and the model was trained for 40 epochs. The learning rate was adjusted using a Warmup Cosine Scheduler, and we applied a dynamic fine-tuning strategy to modify the additional angular margin m of AAM-Softmax. In the first 3 epochs, a warm-up phase was implemented, with the learning rate gradually increasing from a minimum of 5×10^{-4} to a maximum of 1×10^{-3}, after which it gradually decreased. Starting from epoch 5 to epoch 15, the additional angular margin m progressively increased from 0 to 0.3.

Large-margin fine-tuning: We connected shorter utterances to ensure at least 6-s fixed-length segments. The batch size was configured to 128, and the model underwent training for 10 epochs. Similar to the pre-training phase, the learning rate was adjusted using the Warmup Cosine Scheduler, and the dynamic fine-tuning strategy was applied to further modify the additional angular margin m. In the first epoch, a warm-up phase was introduced, where the learning rate gradually increased from 5×10^{-5} to 1×10^{-5}, then gradually decreased. From epoch 2 to epoch 5, the additional angular margin m increased from 0.3 to 0.5.

4. Experimental Results

4.1. Experimental Results on VoxCeleb1

(1) LST: Table 1 compares the performance of our proposed LST network with the ECAPA-TDNN and Transformer baselines across the Voxceleb1-O, E, and H test sets, evaluated using parameter count, EER, and minDCF. The results demonstrate that LST

consistently outperforms both baselines under all three test conditions. Crucially, LST achieves superior accuracy while using significantly fewer parameters. For instance, the 6-block LST configuration (10.7 M) uses less than half the parameters of ECAPA-TDNN (20.8 M) yet achieves substantially lower error rates, exhibiting average reductions of approximately 16% in EER and 21% in minDCF. Compared to the Transformer baseline, LST also shows significant improvements; the 9-block LST model (14.6 M), striking an optimal balance between model size and performance, achieves average reductions of approximately 28% in EER and 19% in minDCF relative to the Transformer baseline.

Complementing the numerical results, Figure 6 presents the Detection Error Tradeoff (DET) curves for models with varying block sizes (B = 6, 9, 12). Across all Voxceleb test conditions, the DET curves for LST consistently reside closer to the lower-left corner than those of ECAPA-TDNN and Transformer. This positioning reflects LST's superior ability to balance false-positive and false-negative rates, resulting in higher accuracy and stability.

Table 1. Evaluation results of different baselines under three experimental conditions of Voxceleb1-O, Voxceleb1-E, and Voxceleb1-H(EER%,minDCF (p = 0.01)).

Backbone	Block	Params(M)	Voxceleb-O		Voxceleb-E		Voxceleb-H	
			EER%	minDCF	EER%	minDCF	EER%	minDCF
ECAPA-TDNN	3	20.8	1.34	0.16	1.60	0.18	3.17	0.30
Transformer	6	11.8	1.64	0.15	1.86	0.20	3.15	0.29
	9	16.5	1.62	0.14	1.78	0.18	3.11	0.28
	12	21.1	1.62	0.17	1.81	0.19	3.13	0.29
LST	6	10.7	1.16	0.12	1.38	0.15	2.52	0.24
	9	14.6	**1.02**	**0.11**	1.32	0.15	2.44	**0.23**
	12	18.5	1.12	0.12	**1.30**	**0.14**	**2.40**	0.23

Bold indicates the best performance in each column.

These findings demonstrate that LST is a highly efficient and effective backbone for speaker verification systems. It achieves superior accuracy compared to established baselines such as ECAPA-TDNN and Transformer while maintaining a significantly lower parameter count. This unique balance of high performance and resource efficiency makes LST ideal for practical applications with limited computational resources.

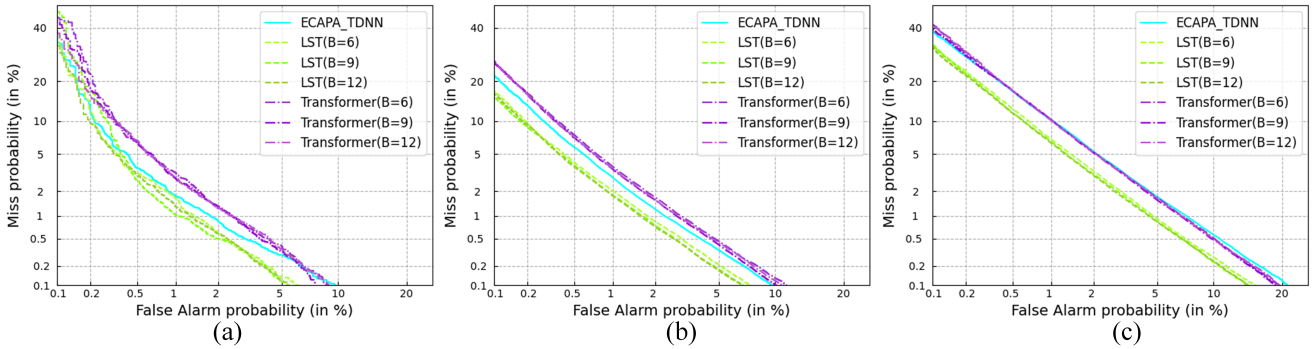

Figure 6. Comparative Detection Error Tradeoff (DET) curves of different baselines on (**a**) VoxCeleb1-O, (**b**) VoxCeleb1-E, and (**c**) VoxCeleb1-H test condition.

(2) Res2Former: To evaluate the effectiveness of our proposed Res2Former architecture, we conducted experiments on the standard VoxCeleb1 dataset under three common testing protocols: VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. The results are summarized in Table 2 and Figure 7.

Table 2 compares the EER and minDCF of Res2Former with several state-of-the-art baselines, including MFA-Conformer, ECAPA-TDNN, NeXt-TDNN, and Res2Net. Our

Res2Former models are evaluated in both Base and Large configurations, with varying architectural capacities.

Table 2. Evaluation results of different models in three test conditions, Voxceleb1-O, Voxceleb1-E, and Voxceleb1-H(EER%,minDCF (p = 0.01)).

	Model	Params(M)	Voxceleb-O		Voxceleb-E		Voxceleb-H	
			EER%	minDCF	EER%	minDCF	EER%	minDCF
	MFA-Conformer	28.35	1.03	0.12	1.36	0.15	2.59	0.24
	ECAPA-TDNN	20.8	1.34	0.16	1.60	0.18	3.17	0.30
	NeXt-TDNN	7.14	0.92	0.10	1.03	0.11	1.91	0.18
	Res2Net	4.03	1.86	0.19	1.69	0.18	2.92	0.26
Base	Res2Former (B = 6, C = 80)	**1.73**	1.23	0.12	1.36	0.14	2.38	0.23
Base	Res2Former (B = 3, C = 128)	2.39	1.03	0.09	1.22	0.13	2.11	0.21
Base	Res2Former (B = 2, C = 192)	3.81	0.99	0.09	1.13	0.12	1.99	0.19
Large	Res2Former (B = 2, C = 256)	6.62	**0.81**	**0.08**	**0.98**	**0.11**	**1.81**	**0.17**
Large	Res2Former (B = 2, C = 288)	8.31	0.91	0.09	1.07	0.12	2.00	0.20
Large	Res2Former (B = 1, C = 384)	9.06	1.01	0.08	1.11	0.12	1.97	0.19

Bold indicates the best performance in each column.

As shown in Table 2, the Res2Former consistently achieves superior performance across all three test conditions. For the Large setting, Res2Former (B = 2, C = 256) attains the lowest EER of 0.81% and the lowest minDCF of 0.08 on VoxCeleb1-O, outperforming all baseline models. Similarly, it achieves the best results on VoxCeleb1-E (EER = 0.98%, minDCF = 0.11) and VoxCeleb1-H (EER = 1.81%, minDCF = 0.17). Notably, the Res2Former maintains competitive performance even with a relatively small number of parameters (1.73M) compared to other models.

The DET curves in Figure 7 further illustrate the superior discriminative capability of Res2Former. Across all operating points, the miss probability of Res2Former is consistently lower than that of the baseline systems, confirming its robustness under varying thresholds. The performance gain is especially pronounced under challenging test conditions (VoxCeleb1-H), where Res2Former demonstrates clear advantages.

The experimental results validate the Res2Former architecture as a powerful solution for speaker verification tasks. Its lightweight design, combined with high performance, positions it as a compelling choice for real-world scenarios with resource constraints.

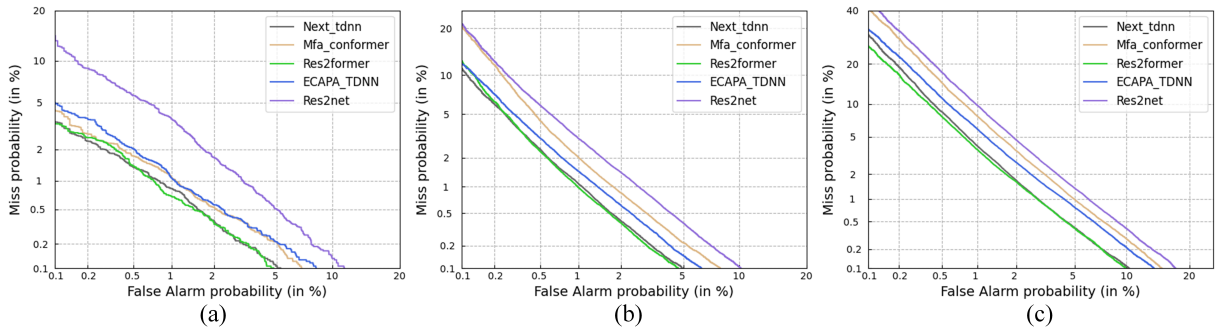

Figure 7. Comparative DET curves of different SV models on (**a**) VoxCeleb1-O, (**b**) VoxCeleb1-E, and (**c**) VoxCeleb1-H test condition.

4.2. Experimental Results on Cn-Celeb(E)

To assess the suitability of Res2Former for robust SV, particularly in relation to its strong object separation capabilities, we conducted experiments on the challenging Cn-Celeb dataset. Additionally, we evaluated the Res2Former architecture on the Cn-Celeb(E) dataset, which features diverse and demanding conditions for speaker verification. The results, presented in

Table 3 and Figure 8, illustrate the performance of Res2Former in comparison to state-of-the-art baselines, including MFA-Conformer, ECAPA-TDNN, NeXt-TDNN, and Res2Net.

As shown in Table 3, Res2Former strikes an impressive balance between computational efficiency and recognition accuracy. Both the Base and Large configurations exhibit significantly lower RTF than MFA-Conformer (33.06×10^{-3}) and ECAPA-TDNN (42.14×10^{-3}), which suffer from quadratic self-attention complexity and high-dimensional feature processing, respectively. This efficiency is attributed to Res2Former's multi-scale hierarchical design and optimized attention mechanisms. Notably, Res2Former (B = 2, C = 256) achieves a minDCF of 0.46 and an EER of 8.39%, improving on ECAPA-TDNN by 23% in minDCF and 28% in EER, while operating 8 times faster.

Table 3. Evaluation results of different models under Cn-Celeb(E) test conditions (RTF ($\times 10^{-3}$), EER%, minDCF ($p = 0.01$)).

	Model	RTF($\times 10^{-3}$)	CN-Celeb(E)	
			EER%	minDCF
	MFA-Conformer	33.06	12.13	0.62
	ECAPA-TDNN	42.14	11.66	0.60
	NeXt-TDNN	5.30	10.48	0.54
	Res2Net	6.85	11.12	0.58
Base	Res2Former (B = 6, C = 80)	6.45	9.89	0.57
Base	Res2Former (B = 3, C = 128)	4.25	9.19	0.56
Base	Res2Former (B = 2, C = 192)	**4.14**	8.43	0.47
Large	Res2Former (B = 2, C = 256)	5.16	**8.39**	**0.46**
Large	Res2Former (B = 2, C = 288)	6.14	8.99	0.48
Large	Res2Former (B = 1, C = 384)	7.11	9.16	0.49

Bold indicates the best performance in each column.

In the Large configuration, increasing the channel dimensions (C) from 192 to 384 slightly improves robustness, as evidenced by the decrease in minDCF. However, the EER remains relatively stable, indicating the scalability of Res2Former for computationally efficient configurations.

The DET curves in Figure 8 further corroborate the superior performance of Res2Former. Across all operating points, Res2Former demonstrates a significantly lower miss probability compared to baseline systems, indicating its robustness under varying thresholds. The largest performance gap is observed at high false alarm probabilities, where Res2Former effectively minimizes mistakes.

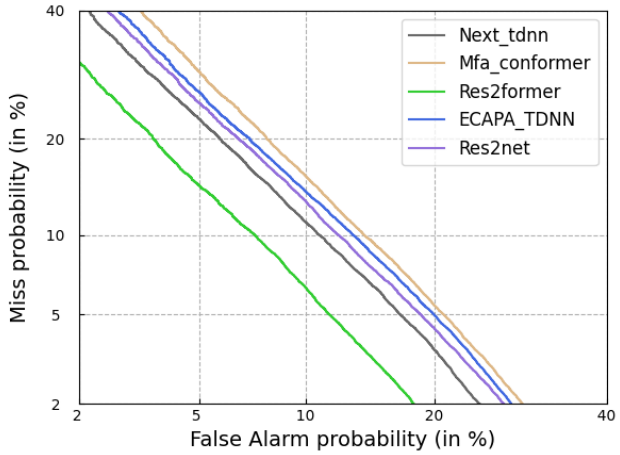

Figure 8. Comparative DET curves of different SV models on Cn-Celeb(E) test condition.

On Cn-Celeb(E), the Res2Former achieves state-of-the-art performance while retaining computational efficiency, reaffirming its suitability for applications with stringent resource limitations.

4.3. Memory Consumption and Ablation

This section analyzes the memory consumption and ablation results for the proposed SV model.

(1) Memory Efficiency: As shown in Figure 9, traditional models such as NeXt-TDNN and ECAPA-TDNN consume a large amount of memory, with NeXt-TDNN being the most costly at approximately 580 MB. While MFA-Conformer reduces memory usage to about 430 MB, it still requires a significant amount of resources. On the other hand, Res2Net, though compact in size, fails to achieve the desired inference efficiency due to its complex bottleneck blocks. Despite reducing the overall memory cost to about 210 MB, this complexity results in inefficiencies in real-time processing. This suggests that while smaller models may reduce static memory costs, the architectural complexity leads to additional memory overhead during dynamic inference.

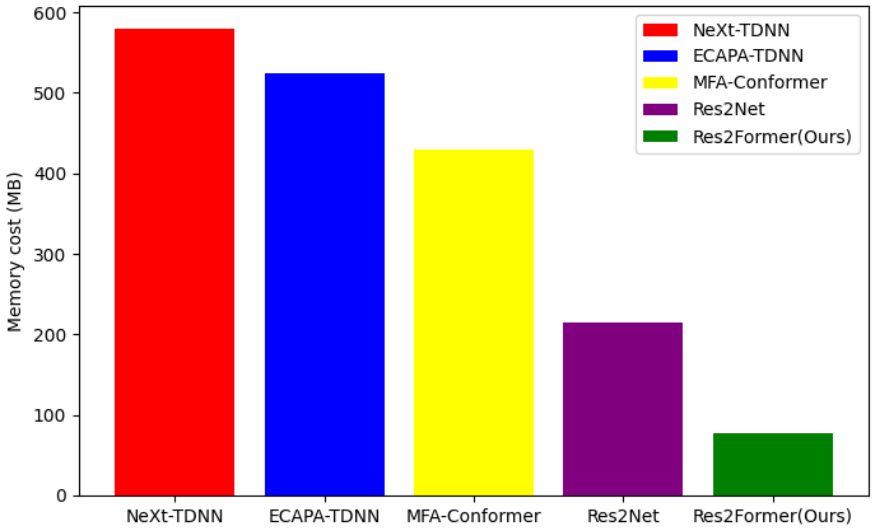

Figure 9. Memory consumption during inference for different models.

In contrast, our proposed Res2Former optimizes both memory efficiency and inference speed. With a minimum memory cost of approximately 80 MB, Res2Former has the lowest memory usage among the compared models. More importantly, it enhances inference efficiency by about 2.6 times compared to Res2Net. This balance of low memory cost and high inference speed makes Res2Former an ideal choice for applications that require efficient and scalable deployment.

(2) Ablation Study: Table 4 presents an investigation of the impact of the GRN and various feature processing strategies. The results from the FFN without GRN indicate that while the GRN does affect model efficiency, its absence does not lead to a significant decline in performance. Neither the weighted averaging method nor the use of concat alone achieves performance comparable to the baseline setup. In particular, the "weighted averaging" method results in the highest EER% in some cases, suggesting that simple averaging may undermine the model's ability to differentiate between speakers effectively. The absence of the TAFF module also leads to a certain degree of decline in effectiveness, with noticeable increases in EER and minDCF values. Moreover, any combined fusion of features at different depths proves to be better than relying solely on a single output.

Table 4. Ablation study of Res2Former in three evaluation conditions.

	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
	EER%	**minDCF**	**EER%**	**minDCF**	**EER%**	**minDCF**
Res2Former (B = 2, C = 256)	**0.81**	**0.08**	**0.98**	**0.11**	**1.81**	**0.17**
Without GRN	0.85	0.09	1.04	0.11	1.85	0.17
Without Concat	1.22	0.11	1.25	0.13	2.21	0.21
Without TAFF	1.10	0.09	1.20	0.12	2.07	0.19
Weight Avg	1.26	0.11	1.21	0.13	2.13	0.19
Only Concat	0.90	0.09	1.05	0.11	1.86	0.17

Bold indicates the best performance in each column.

5. Conclusions

In this paper, we propose enhancements to the Transformer architecture by introducing multi-scale convolutional attention and a GRN-based feed-forward network, culminating in a lightweight backbone architecture termed LST. LST enables flexible control of information flow and fine-grained modeling of speaker embeddings through multi-branch convolution and GRN. Building upon LST, we integrate the Res2Net architecture to develop Res2Former—a low-parameter, high-precision SV model. The Res2Former, grounded in LST, achieves an optimal balance between performance and inference efficiency, fulfilling high-performance requirements even in resource-constrained environments.

While LST and Res2Former successfully reduce overall model parameters, there remains potential for further optimization, particularly for deployment in extremely resource-constrained devices or real-time systems. The performance benefits of multi-scale convolutional attention and GRN-based feed-forward networks are likely contingent on meticulous hyperparameter tuning and architectural decisions, which may introduce challenges in model design and maintenance. Future work could focus on exploring advanced techniques for model compression and quantization to further minimize model size and inference time without compromising accuracy, thereby facilitating broader deployment.

Author Contributions: Conceptualization, D.C., Y.Z., X.W., S.X., X.L. and Y.S.; methodology, D.C.; software, D.C. and Y.Z.; validation, D.C., Y.Z. and X.W., formal analysis, D.C. and Y.Z.; investigation, D.C.; resources, D.C. and Y.Z.; writing—original draft preparation, D.C. and Y.Z.; writing—review and editing, D.C. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Hangzhou Science and Technology Commission of China (2024SZD1A19).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kinnunen, T.; Li, H. An overview of text-independent speaker recognition: From features to supervectors. *Speech Commun.* **2010**, *52*, 12–40. [CrossRef]
2. Naika, R. An overview of automatic speaker verification system. In *Proceedings of the Intelligent Computing and Information and Communication: Proceedings of 2nd International Conference, ICICC 2017*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 603–610.
3. Chen, G.; Chenb, S.; Fan, L.; Du, X.; Zhao, Z.; Song, F.; Liu, Y. Who is real bob? adversarial attacks on speaker recognition systems. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 694–711.
4. Hayashi, V.T.; Ruggiero, W.V. Hands-free authentication for virtual assistants with trusted IoT device and machine learning. *Sensors* **2022**, *22*, 1325. [CrossRef] [PubMed]
5. Sigona, F.; Grimaldi, M. Validation of an ECAPA-TDNN system for Forensic Automatic Speaker Recognition under case work conditions. *Speech Commun.* **2024**, *158*, 103045. [CrossRef]

6. Waghmare, K.; Gawali, B. Speaker Recognition for forensic application: A Review. *J. Posit. Sch. Psychol.* **2022**, *6*, 984–992.
7. Variani, E.; Lei, X.; McDermott, E.; Moreno, I.L.; Gonzalez-Dominguez, J. Deep neural networks for small footprint text-dependent speaker verification. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 4052–4056.
8. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
11. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
12. Desplanques, B.; Thienpondt, J.; Demuynck, K. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv* **2020**, arXiv:2005.07143.
13. Heo, H.J.; Shin, U.H.; Lee, R.; Cheon, Y.; Park, H.M. NeXt-TDNN: Modernizing multi-scale temporal convolution backbone for speaker verification. In Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 11186–11190.
14. Vaswani, A. Attention is all you need. In Proceedings of the NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
15. Khan, M.; Ahmad, J.; Gueaieb, W.; De Masi, G.; Karray, F.; El Saddik, A. Joint Multi-Scale Multimodal Transformer for Emotion Using Consumer Devices. *IEEE Trans. Consum. Electron.* **2025**, *71*, 1092–1101. [CrossRef]
16. Chen, H.; Zendehdel, N.; Leu, M.C.; Moniruzzaman, M.; Yin, Z.; Hajmohammadi, S. Repetitive Action Counting Through Joint Angle Analysis and Video Transformer Techniques. In Proceedings of the International Symposium on Flexible Automation. American Society of Mechanical Engineers, Seattle, DC, USA, 21–24 July 2024; Volume 87882, p. V001T08A003.
17. Safari, P.; India, M.; Hernando, J. Self-attention encoding and pooling for speaker recognition. *arXiv* **2020**, arXiv:2008.01077.
18. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
19. Hsu, W.N.; Bolte, B.; Tsai, Y.H.H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **2021**, *29*, 3451–3460. [CrossRef]
20. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 1505–1518. [CrossRef]
21. Zhang, N.; Wang, J.; Hong, Z.; Zhao, C.; Qu, X.; Xiao, J. Dt-sv: A transformer-based time-domain approach for speaker verification. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–7.
22. Zhang, Y.; Lv, Z.; Wu, H.; Zhang, S.; Hu, P.; Wu, Z.; Lee, H.y.; Meng, H. Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification. *arXiv* **2022**, arXiv:2203.15249.
23. Gulati, A.; Qin, J.; Chiu, C.C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv* **2020**, arXiv:2005.08100.
24. Peng, J.; Stafylakis, T.; Gu, R.; Plchot, O.; Mošner, L.; Burget, L.; Černockỳ, J. Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
25. Peng, J.; Plchot, O.; Stafylakis, T.; Mošner, L.; Burget, L.; Černockỳ, J. An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification. In Proceedings of the 2022 IEEE Spoken Language Technology Workshop (SLT), Rhodes Island, Greece, 4–10 June 2023; pp. 555–562.
26. Okabe, K.; Koshinaka, T.; Shinoda, K. Attentive statistics pooling for deep speaker embedding. *arXiv* **2018**, arXiv:1803.10963.
27. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
28. Nagrani, A.; Chung, J.S.; Zisserman, A. Voxceleb: A large-scale speaker identification dataset. *arXiv* **2017**, arXiv:1706.08612.
29. Chung, J.S.; Nagrani, A.; Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv* **2018**, arXiv:1806.05622.
30. Fan, Y.; Kang, J.; Li, L.; Li, K.; Chen, H.; Cheng, S.; Zhang, P.; Zhou, Z.; Cai, Y.; Wang, D. Cn-celeb: A challenging chinese speaker recognition dataset. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 4–9 May 2020; pp. 7604–7608.
31. Li, L.; Liu, R.; Kang, J.; Fan, Y.; Cui, H.; Cai, Y.; Vipperla, R.; Zheng, T.F.; Wang, D. Cn-celeb: Multi-genre speaker recognition. *Speech Commun.* **2022**, *137*, 77–91. [CrossRef]

32. Cai, W.; Chen, J.; Zhang, J.; Li, M. On-the-fly data loader and utterance-level aggregation for speaker and language recognition. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **2020**, *28*, 1038–1051. [CrossRef]
33. Snyder, D.; Chen, G.; Povey, D. Musan: A music, speech, and noise corpus. *arXiv* **2015**, arXiv:1510.08484.
34. Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M.L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5220–5224.