Golden Gemini is All You Need: Finding the Sweet Spots for Speaker Verification

Tianchi Liu, *Student Member, IEEE*, Kong Aik Lee, *Senior Member, IEEE*, Qiongqiong Wang, *Member, IEEE*, and Haizhou Li, *Fellow, IEEE*

***Abstract*—The residual neural networks (ResNet) demonstrate the impressive performance in automatic speaker verification (ASV). They treat the time and frequency dimensions equally, following the default stride configuration designed for image recognition, where the horizontal and vertical axes exhibit similarities. This approach ignores the fact that time and frequency are asymmetric in speech representation. We address this issue and postulate *Golden-Gemini Hypothesis,* which posits the prioritization of temporal resolution over frequency resolution for ASV. The hypothesis is verified by conducting a systematic study on the impact of temporal and frequency resolutions on the performance, using a trellis diagram to represent the stride space. We further identify two optimal points, namely *Golden Gemini*, which serves as a guiding principle for designing 2D ResNet-based ASV models. By following the principle, a state-of-the-art ResNet baseline model gains a significant performance improvement on VoxCeleb, SITW, and CNCeleb datasets with 7.70%/11.76% average EER/minDCF reductions, respectively, across different network depths (ResNet18, 34, 50, and 101), while reducing the number of parameters by 16.5% and FLOPs by 4.1%. We refer to it as *Gemini* ResNet. Further investigation reveals the efficacy of the proposed *Golden Gemini* operating points across various training conditions and architectures. Furthermore, we present a new benchmark, namely the *Gemini* DF-ResNet, using a cutting-edge model.**

***Index Terms*—Speaker verification, speaker recognition, 2D CNN, ResNet, stride configuration, temporal resolution.**

Manuscript received 17 October 2023; revised 14 February 2024; accepted 17 March 2024. Date of publication 12 April 2024; date of current version 19 April 2024. This work was supported in part by the Agency for Science, Technology and Research (A*STAR), Singapore, through its Council Research Fund under Grant CR-2021-005, in part by the National Natural Science Foundation of China under Grant 62271432, in part by the Shenzhen Science and Technology Research Fund Fundamental Research Key Project under Grant JCYJ20220818103001002, and in part by the Internal Project Fund from Shenzhen Research Institute of Big Data under Grant T00120220002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiao-Lei Zhang. *(Corresponding Author: Kong Aik Lee.)*

Tianchi Liu is with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632, and also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: liu_tianchi@i2r.a-star.edu.sg).

Kong Aik Lee is with the Department of Electrical and Electronic Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: kong-aik.lee@polyu.edu.hk).

Qiongqiong Wang is with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: wang_qiongqiong@i2r.a-star.edu.sg).

Haizhou Li is with the Shenzhen Research Institute of Big Data, School of Data Science, Chinese University of Hong Kong, Shenzhen 518172, China, and also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: haizhouli@cuhk.edu.cn).

Codes and pre-trained models are available at https://github.com/Tianchi-Liu9/Golden-Gemini-for-Speaker-Verification.

Digital Object Identifier 10.1109/TASLP.2024.3385277

I. Introduction

AUTOMATIC speaker verification (ASV) aims to verify the claimed identity of a speaker according to his/her voice [1]. Currently, deep learning-based speaker embedding has emerged as the predominant method [2]. In this approach, fixed-dimensional representations are extracted from enrollment and test speech utterances [3]. These representations, rich in voice characteristics, are referred to as speaker embeddings [2]. The neural networks responsible for extracting these embeddings are known as the embedding extractors. The recognition procedure is often done by measuring the similarity between embeddings, using methods such as cosine similarity or probabilistic linear discriminant analysis (PLDA) [4], [5], [6], [7], [8], [9].

Typical speaker-embedding neural networks consist of three components [10]. First, an encoder is used to extract frame-level features from an input utterance. It is followed by a temporal aggregation layer that combines the frame-level features from the encoder into a fixed-length condensed representation of the entire input sequence. Commonly used temporal aggregation techniques include average pooling [11], statistical pooling [12], attentive pooling [13], [14], and posterior inference [15]. The output stage of the neural network constitutes a decoder that classifies utterance-level representations into speaker classes [16], [17], [18]. It utilizes a stack of fully-connected layers, including a bottleneck layer specifically designed for extracting speaker embeddings. Among these, the encoder is often the heaviest part of the model. The efficacy and efficiency of its design are instrumental to the performance of the model.

Many prior studies have investigated and designed numerous powerful networks as encoders. These backbone networks can be broadly categorized into four main types:

- 2D convolutional neural network (CNN) [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31],
- Time-delay neural network (TDNN) [31], [32], [33], [34], [35], [36],
- Transformer [37], [38], and
- Combinations of the aforementioned three [39], [40], [41], [42], [43], [44], [45], [46], [47].

Among these architectures, 2D CNN is the most widely used for ASV. It is worth mentioning that in the VoxCeleb Speaker Recognition Challenge (VoxSRC) 2021 [48] and 2022 [49], the best-performing models are based on 2D CNNs, with ResNet [50] being the preferred choice [51], [52], [53], [54]. ResNet is not only popular in ASV but also widely employed

in other speech-related tasks, such as speaker extraction [55], [56], [57], [58], [59], target-speaker voice activity detection [60], [61], [62], speaker diarization [63], [64], and speech anti-spoofing [65], [66], [67], [68]. Therefore, investigating the ResNet architecture for speech-related tasks holds significant importance.

The ResNet architecture was initially designed for image recognition [50] where the horizontal and vertical dimensions of images have similar implications [69], [70] and are often uniform in size, typically $N \times N$ pixels with commonly used values such as 224 and 384. Consequently, it is intuitive to treat these two dimensions equally with the default equal-stride configuration in ResNet [50]. However, when dealing with speech representations, the time and frequency axes of speech spectrograms possess distinct implications [71] and often vary in size (e.g., 80×301 [40]). Therefore, the techniques that work for image recognition may not be suitable for ASV, thus necessitating appropriate modifications. Despite these notable differences in feature properties between images and speech signals, existing ASV systems based on the ResNet models [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [51], [52], [53], [54] continue to treat the frequency and temporal resolutions equally by adopting the default stride configuration as the original ResNet. Doubts arise regarding the adequacy of this equal-stride configuration for ASV.

The preservation of temporal resolution in various existing ASV methods [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [47], [72], [73], [74] has led to the hypothesis that ASV may be more sensitive to temporal resolution than frequency resolution. TDNN-based models [31], [32], [33], [34], [35], [36], [39], [40], [41], [42], [47] preserve the temporal resolution across the stacked layers. Similarly, recurrent networks, such as the long short-term memory (LSTM), preserve the number of frames [73], [74]. Recent studies [37], [38], [47] adopt the Transformer architecture as the encoder, ensuring the preservation of the temporal resolution across stacked Transformer blocks. Should temporal resolution prove to be of greater significance, the equal-stride configuration may not be optimal since it diminishes the temporal resolution. The current understanding of the impact of temporal and frequency resolutions on the performance of ResNet-based ASV models remains limited, leaving a research gap to be filled. Consequently, this motivates us to explore the relative importance of temporal and frequency resolution in the feature representation process of ASV. Building upon this investigation, we identify the optimal stride configurations that account for the inherent characteristics of speech signals to better align with the requirements of ASV, leading to improved performance. We also conduct a meticulous analysis of the trade-offs between performance and model complexity to ensure both efficacy and efficiency. The major contributions of this work are summarized as follows:

- We postulate *Golden-Gemini Hypothesis*, which posits that the preservation of temporal resolution is to be prioritized over frequency resolution for the optimal extraction of speaker characteristics.
- We systematically analyze the joint effects of temporal and frequency resolutions through a carefully designed trellis diagram. Two optimal spots on the trellis diagram are identified and named *Golden Gemini*.
- Based on the insights gained from the trellis diagram analysis, we summarize a set of guiding principles for designing ResNet-based models for ASV.
- The compatibility and efficacy of the proposed *Golden Gemini* models are evaluated under various aspects, including model sizes, structures (backbones, attention, pooling layers and micro design), training strategies, and in/cross-domain test sets.
- We introduce the *Gemini* DF-ResNet, as the new state-of-the-art (SOTA) benchmark for ASV.

TABLE I
COMPARISON BETWEEN THE ORIGINAL RESNET34 [50], MODIFIED RESNET [19] AND THE PROPOSED *GEMINI* RESNET34

Stage	Layer	original ResNet34 Stride	original ResNet34 Output	modified ResNet34 Stride	modified ResNet34 Output	*Gemini* ResNet34 Stride	*Gemini* ResNet34 Output
conv1	7×7, C	(2,2)	F/2×T/2	-	-	-	-
	3×3, C	-	-	(1,1)	F×T	(1,1)	F×T
conv2	Max Pooling	(2,2)	F/4×T/4	-	-	-	-
	$\begin{bmatrix} 3\times3, C \\ 3\times3, C \end{bmatrix}\times3$	(1,1)	F/4×T/4	(1,1)	F×T	(2,1)	F/2×T
conv3	$\begin{bmatrix} 3\times3, C\times2 \\ 3\times3, C\times2 \end{bmatrix}\times4$	(2,2)	F/8×T/8	(2,2)	F/2×T/2	(2,2)	F/4×T/2
conv4	$\begin{bmatrix} 3\times3, C\times4 \\ 3\times3, C\times4 \end{bmatrix}\times6$	(2,2)	F/16×T/16	(2,2)	F/4×T/4	(2,1)	F/8×T/2
conv5	$\begin{bmatrix} 3\times3, C\times8 \\ 3\times3, C\times8 \end{bmatrix}\times3$	(2,2)	F/32×T/32	(2,2)	F/8×T/8	(2,1)	F/16×T/2

A 2D CNN layer is represented in the format of [kernel size × kernel size, number of channels (C)]. In the original ResNet34, C is set to 64, while the modified ResNet and gemini ResNet use a value of 32. The symbol '-' indicates the layer is not employed in the model. When applicable, a (2,2) stride is performed in the first CNN layer of the stage.

II. BACKGROUND

A. ResNet Architecture

ResNet is first proposed for image recognition [50]. A standard ResNet comprises five stages. The first stage is a 7×7 convolutional (*conv*) layer, followed by four stages. Each stage contains multiple residual blocks, as shown in Table I. Residual blocks are defined as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{W}_s\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors of a residual block. The function $\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ represents the residual mapping to be learned. The operation $\mathcal{F} + \mathbf{x}$ is performed by a shortcut connection and element-wise addition. $\mathbf{W}_s$ denotes a linear projection used in the shortcut to match the dimensions of $\mathbf{x}$ to $\mathcal{F}$. The design of $\mathcal{F}$ is flexible and commonly categorized into two types: basic block and bottleneck block. Basic blocks utilize two (3×3) convolutional layers, whereas the bottleneck blocks are composed of (1×1), (3×3), and (1×1) convolutions. The weights of these layers are denoted as $\{\mathbf{W}_i\}$, and the bias is omitted for simplicity [50].

The depth of ResNet is determined by the number of layers M, which is dictated by the types and number m of residual

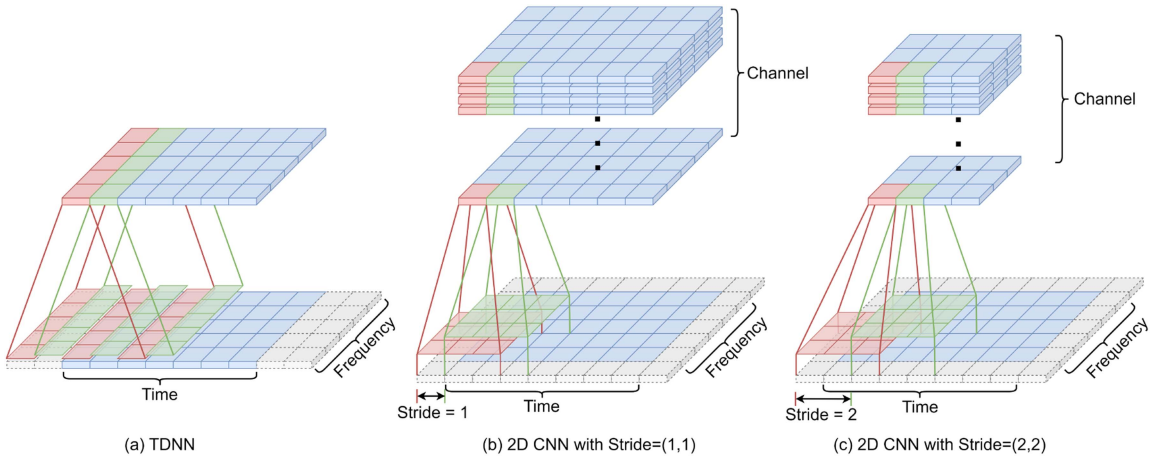


Fig. 1. Illustration of convolution operations in (a) TDNN, (b) 2D CNN with stride = (1, 1), and (c) stride = (2, 2). The blue and grey cuboids represent time-frequency bins of feature maps and paddings, respectively.

blocks. It is formulated as:

$$M = \begin{cases} 2 \times m + k, & \text{if basic block} \\ 3 \times m + k, & \text{if bottleneck block} \end{cases}, \tag{2}$$

where k accounts for the convolution layer in the first *conv1* stage and the bottleneck layer in the decoder, typically assigned a value of 2. ResNets with $M = 18/34/50/101/152$ layers are commonly adopted [50]. The depth can be further extended, such as 233 [24] and 1202 [50] layers. In addition to expanding the depth, previous studies explore various variations of ResNet architecture from different perspectives to improve the performance, including ResNeXt [75], ConvNeXt [70], Res2Net [76], squeeze-and-excitation network (SENet) [77], depth-first ResNet (DF-ResNet) [24], [25], separate downsampling ResNet (SD-ResNet) [70], modified ResNet [19], [21], thin-ResNet [20] and fast ResNet [18].

We observe that the five-stage structure remains intact, despite the adjustments to network depth or modifications to the model architecture [18], [19], [20], [21], [24], [25], [70], [75], [76], [77]. Therefore, in this work, we validate our hypothesis by adopting the five-stage design, while acknowledging that the hypothesis itself is applicable to architectures with arbitrary stages. The generality of our proposed method allows its application to all 2D CNN models following the five-stage design, including Res2Net [76], SENet [77], DF-ResNet [24], [25], SD-ResNet [70], and modified ResNet [19], [21], as validated through experiments.

B. Extensions of ResNet

The ResNet initially designed for an image recognition task [50], exhibits inferior performance when directly applied to speaker verification [20]. Our initial findings also suggest the same, highlighting the inherent differences between image and speech, and the necessity of customizing ResNet for speech-related tasks.

Preserve Resolutions: By simply removing the stride operations (2, 2) in the first and second stages, a modified ResNet gains a remarkable improvement [19], [21]. A comparison of the original ResNet [50] and the modified structure [19] is shown in Table I. We believe that removing the stride operations in the first two stages preserves the time and frequency resolutions, allowing for the extraction of low-level features. This assumption emphasizes the significance of resolutions as an important aspect of ASV. Nevertheless, it remains uncertain whether the time resolution, frequency resolution, or both are significant to the overall performance, which warrants further investigation.

Prioritize depth over width: Previous studies adopt a computationally efficient operation by reducing the width of ResNet [18], [19], [28]. Recent work further investigates the trade-off between the depth and width of networks, highlighting that depth plays a more important role in ASV [24]. In this paper, we examine ResNet-based networks from a different perspective, focusing on investigating how time and frequency resolutions affect performance, as well as considering the model size and FLOPs. Our findings complement the depth-first rule [24] presented in Section V-E.

C. Stride and Resolution

In this subsection, we provide an overview of how the stride configuration influences the temporal and frequency resolutions in the 1D TDNN and 2D CNN models. This forms the basis for our subsequent exploration and investigation in the following sections.

As illustrated in Fig. 1(a), a TDNN network implemented with dilated 1D CNN layers [32] treats the input as 1D features, while considering the frequency dimension as channels. TDNN-based models are not included in this study due to the absence of the frequency dimension, and existing TDNN models generally maintain time resolution [31], [32], [33], [34], [35], [36], [39], [40], [41], [42], [47]. Unlike TDNNs, a 2D CNN considers the input feature as a 3-dimensional tensor $C \times F \times T$, where C, F, and T represent the channel, frequency, and time dimensions, respectively [24]. By employing multiple 2D CNN layers, the number of channels increases, while the frequency and temporal resolutions decrease by downsampling operations to reduce computational complexity [50]. The output

dimension of the downsampling operation is mainly controlled by the stride. Fig. 1(b) and (c) illustrate that by adjusting the stride in each dimension, the temporal and frequency resolutions can be controlled independently. For instance, setting the stride to 2 on the time dimension and 1 for the frequency dimension roughly halves the time resolution while keeping the frequency resolution the same.

In addition to stride (S), the output resolution (R_{out}) is also affected by the input resolution (R_{in}), padding (P), dilation (D), and the kernel size (K), as follows:

$$R_{\text{out}} = \frac{R_{\text{in}} + 2 \times P - D \times (K-1) - 1}{S} + 1 \simeq \frac{R_{\text{in}}}{S}. \quad (3)$$

In summary, the temporal and frequency resolutions are primarily controlled by the stride configuration employed on each dimension. In this paper, we investigate the impact of time and frequency resolutions on ASV performance by comparing different stride configurations. We aim to identify the optimal stride configurations for ASV.

III. Golden-Gemini is All You Need

A. Golden-Gemini Hypothesis

Considering the distinct physical implications of the two dimensions in speech representations, we raise doubts regarding the appropriateness of employing the default equal-stride configuration, originally designed for image recognition. Furthermore, given that existing studies show the benefit of preserving the temporal resolution during the feature extraction stage [31], [32], [33], [34], [35], [36], [37], [39], [40], [41], [42], [73], [74], we postulate the following hypothesis:

Golden-Gemini Hypothesis: In the context of a ResNet architecture, characterized by a sequence of multiple stages (typically 5), there exist operational states that yield optimal performance. These states can be determined by following a temporal-frequency stride configuration that prioritizes the preservation of temporal resolution over frequency resolution. We refer to these specific operational states as the *Golden-Gemini* configurations.

The *Golden-Gemini Hypothesis* posits that the preservation of temporal resolution is to be prioritized over frequency resolution for the optimal extraction of speaker characteristics.

The uniqueness of a person's voice results from the combination of physiological characteristics inherent in the vocal tract and the learned speaking habits of different individuals [78]. The vocal tract shape is an important physical distinguishing factor [78], wherein the laryngeal features encompass pitch and glottal pulse shape, while the supra-laryngeal features are associated with the formant frequencies, bandwidths, and intensities [79]. These features appear across various time scales, underscoring the significance of maintaining adequate temporal resolution for the convolution filters. By progressively covering larger local regions as the network deepens, these filters extract meaningful representations from neighboring frames. On the other hand, the learned speaking habits, including speaking rate and prosodic effects [78], vary along the time dimension. By preserving temporal resolution, models can effectively capture these time-dependent patterns. Conversely, downsampling in the time domain leads to a loss of neighboring frame information and diminishes its ability to capture fine-grained details necessary for robust speaker discrimination.

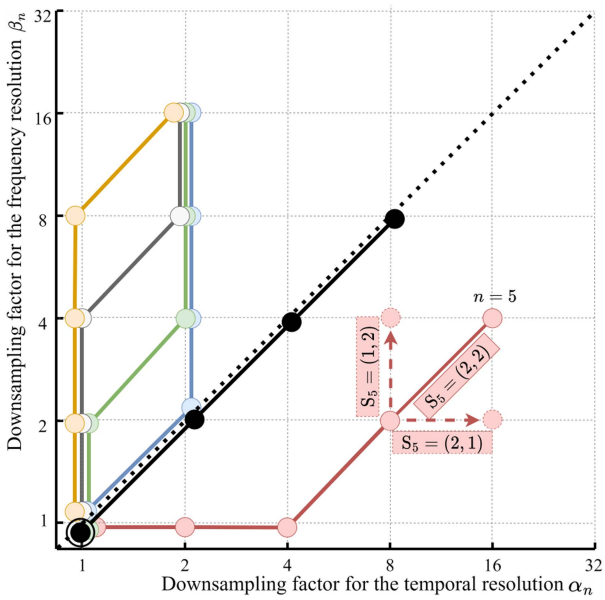


Fig. 2. Exemplar trellis diagram. Each node on the trellis diagram represents the time and frequency downsampling factors, α_n and β_n, at the output of each stage in a ResNet. Each path represents a stride configuration consisting of five sequential stages, for $n = 1, 2, \ldots, 5$. The node with a circular outer ring ⊙ indicates that it remains at the same position by using a stride of (1, 1). Dashed arrows represent two alternative options controlled by different stride operations.

B. Finding the Sweet Spots on the Trellis Diagram

To validate the *Golden-Gemini Hypothesis* and to determine the optimal stride configuration for ASV, we carefully design a search strategy in this subsection. As introduced in Section II-C, the temporal and frequency resolutions are primarily controlled by the stride configuration employed on each dimension during the convolution operations. In order to visually represent the various stride configurations, we utilize a trellis diagram as a graphical tool to aid our study, as illustrated in Fig. 2. This diagram effectively captures the essence of each stride configuration by illustrating a series of sequential stride operations originating from the start point.

Consider the ResNet structure comprising five stages as detailed in Section II-A and Table I. For each stride configuration is represented by five sequential steps on the trellis diagram in Fig. 2, with each step denoting a stride operation performed in a ResNet stage. For the n-th stage, the stride operation is denoted as $\mathbf{S}_n = (s_{\text{t},n}, s_{\text{f},n})$, indicating a reduction in time and frequency resolutions by a pair of stride factor of $s_{\text{t},n}$ and $s_{\text{f},n}$, respectively. When $s_{\text{t},n}$ or $s_{\text{f},n}$ equals to 1, the corresponding resolution remains unchanged. It's important to highlight that strides of 1 or 2 are the most commonly used and are also the default choices in ResNet [50]. Therefore, in this work, we exclusively focus on these two stride operations.

In Fig. 2, α_n and β_n are the downsampling factors at the output of the n-th stage in a ResNet for the temporal and frequency

resolutions, respectively. They are given by the products of the stride factors, and are formulated as follows.

$$\alpha_n = \prod_{i=1}^{n} s_{\mathrm{t},i}, \text{ and } \beta_n = \prod_{i=1}^{n} s_{\mathrm{f},i}. \tag{4}$$

The output temporal resolution $R_{\mathrm{out,t},n}$ and frequency resolution $R_{\mathrm{out,f},n}$ of the n-th stage in a ResNet are derived as:

$$R_{\mathrm{out,t},n} \simeq \frac{R_{\mathrm{init,t}}}{\alpha_n}, \text{ and } R_{\mathrm{out,f},n} \simeq \frac{R_{\mathrm{init,f}}}{\beta_n}, \tag{5}$$

where R_{init} is the initial input resolution for the first stage in a ResNet.

For the red path in Fig. 2, a stride of (2, 2) reduces both time and frequency resolutions by half at stage $n = 5$ simultaneously. The two dashed arrows indicate the alternative stride configurations for reducing the resolution by half in either the frequency dimension alone with a stride of (1, 2), or the time dimension alone with a stride of (2, 1). Additionally, it is allowed to stay at the same node on the trellis diagram, thereby preserving both time and frequency resolutions with a stride of (1, 1). This option is denoted as ⊙ in Fig. 2, represented by the black node at coordinates (1, 1). The early stages often employ this option to retain sufficient information in low-level features. This design aligns with that of the modified ResNet [19], [21], where the first two stages use a stride of = (1, 1).

In the trellis diagram depicted in Fig. 2, the stride configurations that prioritize the preservation of either frequency or time resolution are delineated. The endpoints located on the black dotted line indicate stride configurations that treat time and frequency resolutions equally. This black dotted line also divides the diagram into two partitions. Within the upper-left partition, the stride configurations give precedence to the preservation of temporal resolution over frequency resolution. Conversely, the lower-right partition represents configurations that accentuate frequency resolution, while compromising temporal resolution. In our experiments, we search all possible stride configurations within this trellis diagram to identify the optimal stride configuration. The experimental results and analysis of this search are presented in Section V-B.

In addition, there are multiple paths on the diagram that lead to a single endpoint, each representing a specific stride configuration. Fig. 2 shows an exemplar trellis diagram illustrating four paths, each represented by a different color, converging to the endpoint on (2, 16). The difference among stride configurations that lead to the same endpoint lies in the specific stages within the total of five stages where the downsampling operation with a stride of (2, 2) is applied. Performing the downsampling operation in an early stage reduces the resolution of the output feature map, resulting in a smaller feature size that needs to be convolved by 2D convolutions. Consequently, this reduction in resolution contributes to a decrease in FLOPs. Therefore, the stride configurations towards the same endpoint require for different FLOPs while maintaining the same number of parameters. To assess the impact of early or late downsampling, we explored different paths towards the same point with various FLOPs. The experimental results and analysis of the findings are comprehensively presented in Section V-C.

TABLE II
DEVELOPMENT AND TEST SETS STATISTICS

Test set	# of speakers	# of utterances	# of pairs
VoxCeleb1-O	40	4,708	37,611
VoxCeleb1-H	1,190	137,924	550,894
VoxCeleb1-E	1,251	145,160	579,818
SITW	-	-	721,788
CNCeleb	-	-	3,484,292

IV. EXPERIMENTAL SETUPS

A. Dataset

The experiments are conducted on four large-scale datasets, including the VoxCeleb1 [80], VoxCeleb2 [81], *Speaker in the Wild* (SITW) [82] and CNCeleb [83] datasets.

Training set: During training, only the development partition of the VoxCeleb2 dataset is used, which consists of 5,994 speakers and 1,092,009 utterances. This protocol for training on the VoxCeleb2 dataset is widely adopted [24], [25], [32], [39], [40], [41], [42], [44], [47]. Additionally, a randomly selected 2% portion of this development partition is reserved as the validation set. This small validation set is used to identify the best model for testing on the development and testing sets.

Development set: The VoxCeleb1-Original (Vox1-O) test set is utilized as the development set in this work to conduct a performance comparison of all stride configurations. The outcomes of the tests on this development set are analyzed, leading to the formulation of observations.

Test set: In order to verify the observations across various scenarios, we comprehensively encompass testing scenarios that include in-domain, out-domain, large-scale, and challenging cases. Specifically, VoxCeleb1-Hard (Vox1-H) and VoxCeleb1-Extended (Vox1-E) are used as in-domain large hard cases and a large test set, respectively. The SITW core-core test set serves the purpose of cross-domain testing, while the CNCeleb test set is employed to assess challenging cases within a cross-domain scenario. The statistics of these four test sets are shown in Table II. It's important to highlight that there is no overlap between any of the test sets and the training set or development set.

B. Training Strategy

The experiments are conducted using the Pytorch framework.[1] We adopted two training strategies as detailed below.

Training strategy 1: The SpeechBrain Toolkit[2] [84] is used. For fair comparisons, all systems are trained under the same training strategy following that in [32], [40]. Specifically, the loss function is the additive angular margin softmax (AAM-softmax) [16] with a margin of 0.2 and a scale of 30. The Adam optimizer [85] with cyclical learning rate [86] following a triangular policy [86] is used for training all models. A weight decay of 2×10^{-5} is used for all the weights in the model. The maximum and minimum learning rates of the cyclical scheduler

[1][Online]. Available: https://pytorch.org/
[2][Online]. Available: https://speechbrain.github.io/

are 2×10^{-3} and 2×10^{-8}, and the batch size is 64 each with 5 types of augmented data. For Res2Net and ResNet101, learning rates and batch size are reduced to half due to the large memory occupation.

All training samples are cut into 3-second segments. We employ five augmentation techniques to increase the diversity of the training data. The first two follow the idea of random frame dropout in the time domain [87] and speed perturbation [88]. The remaining three are a set of reverberate data, noisy data, and a mixture of both, achieved by combining with the Room Impulse Response (RIR) dataset [89]. The s-norm [90] is applied to normalize the scores.

Training strategy 2: Wespeaker Toolkit[3] [21] is used. This training strategy follows that in [24] for the purpose of re-implementing DF-ResNet [24] and is only applied to re-implemented DF-ResNet and *Gemini* DF-ResNet reported in Section V-E. Specifically, the loss function is an AAM-softmax [16] with a margin of 0.2 and a scale of 32. The total number of training epochs is set to 165. The AdamW [91] optimizer with 0.05 weight decay is used. The base learning rate (l_{base}) decreases from 1.25×10^{-4} to 1×10^{-6} with the exponential scheduler as the learning rate regulator. The learning rate (l) for training is adjusted according to the batch size (b) and formulated as $l = l_{\text{base}} \times b/64$. All the samples are cut into 200-frame segments with the augmentations of reverberation, noise, and speed perturbation [88] during training. The as-norm [92] is applied to normalize the scores.

C. Evaluation Protocol

We report the performances in terms of the equal error rate (EER) and the minimum detection cost function (minDCF) with $P_{\text{target}} = 0.01$ and $C_{\text{FA}} = C_{\text{Miss}} = 1$. The scores are produced by calculating the cosine distance between embeddings.

V. Results and Analysis

It is worth noting that the FLOPs calculation is correlated with the duration of the sample. We select the most commonly used options of 2 seconds [21], [24], [25], [32], [39], [42] and 3 seconds [30], [37], [40], [41], [46], [84], [93] to calculate FLOPs. The results are labeled as '2 s/3s'.

A. Original ResNet vs. Modified ResNet (Baseline)

We first compare the modified ResNet [19] and original ResNet [50]. The results are presented in Table III. It is obvious that the modified ResNet outperforms the original ResNet. The improved performance of the modified ResNet is attributed to the adequate preservation of frequency-time resolution by changing the stride configurations from (2, 2) to (1, 1) in the first two layers. However, these changes also lead to an increase in FLOPs.

[3][Online]. Available: https://github.com/wenet-e2e/wespeaker

TABLE III
Performance in EER(%) and minDCF of Original ResNet [50] and Modified ResNet [19], [21]

Model	Params (Million)	FLOPs (Giga)	Vox1-O EER / minDCF	Vox1-H EER / minDCF	Vox1-E EER / minDCF	SITW EER / minDCF	CNCeleb EER / minDCF
original ResNet18	11.3	**0.90**	2.903 / 0.315	5.106 / 0.433	2.934 / 0.322	3.609 / 0.396	16.373 / 1.000
modified ResNet18	**3.45**	3.25	**1.760** / **0.177**	**2.785** / **0.244**	**1.600** / **0.170**	**2.132** / **0.210**	**12.301** / **0.657**
original ResNet34	21.41	**1.82**	2.744 / 0.291	4.598 / 0.400	2.614 / 0.284	3.308 / 0.379	15.072 / 1.000
modified ResNet34	**6.63**	6.88	**1.101** / **0.128**	**2.221** / **0.208**	**1.252** / **0.139**	**1.584** / **0.161**	**12.113** / **0.623**

The FLOPs are calculated based on a 3-second sample.

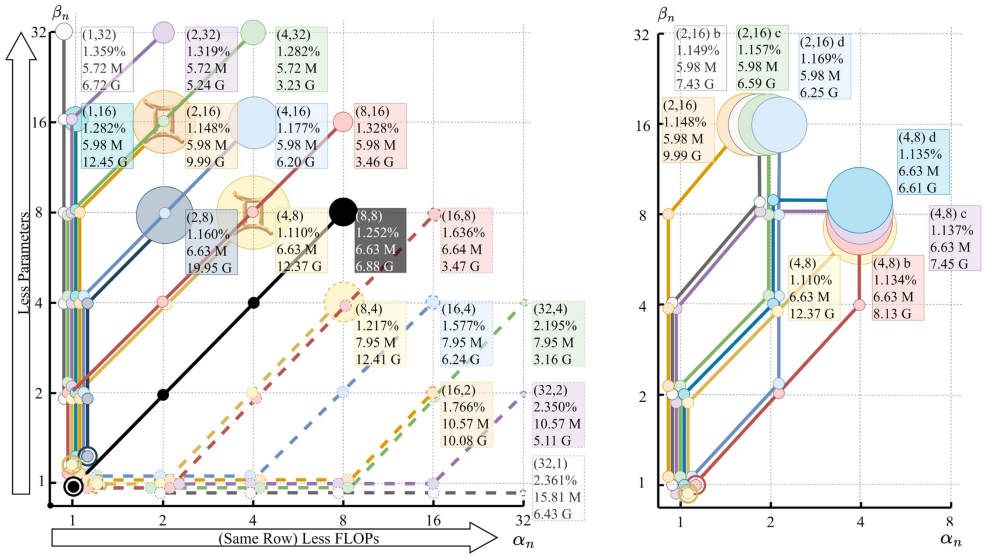
In addition, as this modified ResNet [19] achieves SOTA performance using the equal-stride configuration, we adopt it as the baseline model in this work.

B. Finding the Sweet Spots on the Trellis Diagram

We perform a strategic search on the trellis diagram for optimal stride configuration, as shown in Fig. 3(a). All stride configurations are evaluated on the development set, and the results are reported in the left sub-table of Table IV. These results yield the following observations:

Observation 1: Models that prioritize preservation of temporal resolution over frequency resolution (indexed starting with 'T') tend to outperform the default equal-stride configuration (indexed as MOD). Conversely, configurations that emphasize frequency resolution (indexed starting with 'F') generally result in poorer performance. Fig. 3(a) provides clear evidence that models utilizing stride configurations located in the upper-left partition of the trellis diagram prioritize the preservation of temporal resolution, resulting in a considerable advantage as indicated by the presence of a large bubble. In contrast, models positioned in the lower-right partition demonstrate an opposite trend. These observations strongly support the *Golden-Gemini Hypothesis*, which posits that temporal resolution plays a more important role than frequency resolution in capturing the speaker characteristics of speech signals.

Observation 2: The performance of models with endpoints located on the boundary will significantly deteriorate. As shown in Table IV, models indexed as T05, T15, T25, F52, F51, and F50 exhibit notable performance degradation compared to neighboring models on the trellis diagram. Unlike a TDNN that utilizes large channel numbers (e.g., 512, 1024, or 2048) [32], [39], [40], ResNet employs a smaller channel number (such as 32 or 64) at early stages for low-dimensional information representations [19], [21], [24], [25]. This aligns with the design principle discussed in Section II-B that emphasizes the importance of depth over width. Consequently, when the temporal or frequency resolution is rapidly compressed, and constrained by a limited number of filters, it leads to the loss of information in that specific dimension. This results in a notable degradation of performance. Therefore, when designing a narrow ResNet with



(a) Strategic search for optimal stride configurations (b) Different paths towards the *Golden Gemini*

Fig. 3. Trellis diagrams of (a) the strategic search for optimal stride configurations and (b) different paths towards *Golden Gemini*. ♊ in (a) indicates proposed *Golden-Gemini* stride configurations. In the rectangle box, from top to bottom are: the downsampling factors (α_5, β_5), performance in EER (%) on VoxCeleb-E test set, number of parameters, and FLOPs. The size of the endpoint bubble indicates the performance, and the larger the bubble, the better the performance. The node with a circular outer ring forming as ⊙ indicates that it remains at the same position by using a stride of (1, 1). The solid line represents a stride configuration that prioritizes temporal resolution over frequency resolution, while the dashed line configuration reflects the opposite.

TABLE IV
PERFORMANCE IN EER(%) AND MINDCF OF THE ORIGINAL RESNET34 (ORI) [50] AND THE MODIFIED RESNET34 (MOD) [19], [21] WITH DIFFERENT STRIDE CONFIGURATIONS DEMONSTRATED IN FIG. 3(A)

Index of Stride Config.	Downsampling Factors (α_5, β_5)	Stride Config. [Time] [Frequency]	Params (Million)	FLOPs (Giga) 2s/3s	Vox1-O EER minDCF
ORI	(32,32)	[2,2,2,2,2] [2,2,2,2,2]	21.41	1.25/1.82	2.744 0.291
MOD	(8,8)	[1,1,2,2,2] [1,1,2,2,2]	6.63	4.63/6.88	1.101 0.128
T05	(1,32)	[1,1,1,1,1] [2,2,2,2,2]	5.72	4.49/6.72	1.250 0.131
F50	(32,1)	[2,2,2,2,2] [1,1,1,1,1]	15.81	4.44/6.43	2.526 0.241
T15	(2,32)	[1,1,1,1,2] [2,2,2,2,2]	5.72	3.50/5.24	1.303 0.161
F51	(32,2)	[2,2,2,2,2] [1,1,1,1,2]	10.57	3.52/5.11	2.505 0.243
T25	(4,32)	[1,1,1,2,2] [2,2,2,2,2]	5.72	2.16/3.23	1.218 0.133
F52	(32,4)	[2,2,2,2,2] [1,1,1,2,2]	7.95	2.17/3.16	2.228 0.219
T14	(2,16)	[1,1,1,1,2] [1,2,2,2,2]	5.98	6.68/9.99	**1.058** **0.092**
F41	(16,2)	[1,2,2,2,2] [1,1,1,1,2]	10.57	6.87/10.08	1.882 0.150
T24	(4,16)	[1,1,1,2,2] [1,2,2,2,2]	5.98	4.15/6.20	1.111 0.104
F42	(16,4)	[1,2,2,2,2] [1,1,1,2,2]	7.95	4.24/6.24	1.563 0.135
T34	(8,16)	[1,1,2,2,2] [1,2,2,2,2]	5.98	2.32/3.46	1.260 0.128
F43	(16,8)	[1,2,2,2,2] [1,1,2,2,2]	6.64	2.35/3.47	1.691 0.180
T23	(4,8)	[1,1,1,2,2] [1,1,2,2,2]	6.63	8.27/12.37	1.101 0.095
F32	(8,4)	[1,1,2,2,2] [1,1,1,2,2]	7.95	8.35/12.41	1.223 0.105
T04	(1,16)	[1,1,1,1,1] [1,2,2,2,2]	5.98	8.32/12.45	1.276 0.117
T13	(2,8)	[1,1,1,1,2] [1,1,2,2,2]	6.63	13.33/19.95	1.127 0.107

Index of Stride Config.	Vox1-H EER minDCF	Vox1-E EER minDCF	SITW EER minDCF	CNCeleb EER minDCF	↑↓ EER minDCF
ORI	4.598 0.400	2.614 0.284	3.308 0.379	15.072 1.000	+87.27% +98.30%
MOD	2.221 0.208	1.252 0.139	1.584 0.161	12.113 0.623	Benchmark Benchmark
T05	2.297 0.211	1.359 0.141	1.914 0.177	12.149 0.587	+8.27% +1.79%
F50	3.932 0.329	2.361 0.249	3.262 0.316	12.898 0.755	+69.51% +63.83%
T15	2.251 0.210	1.319 0.137	1.832 0.172	12.115 0.626	+5.60% +1.54%
F51	3.914 0.332	2.350 0.245	3.216 0.314	12.678 0.797	+67.91% +64.58%
T25	2.266 0.212	1.282 0.138	1.640 0.174	13.027 0.680	+3.88% +4.61%
F52	3.720 0.330	2.195 0.235	2.925 0.280	12.937 0.804	+58.58% +57.83%
T14	1.998 0.185	1.148 **0.120**	1.505 0.148	11.670 0.549	-6.75% -11.14%
F41	2.982 0.256	1.766 0.185	2.433 0.240	12.222 0.667	+32.46% +28.11%
T24	2.084 0.193	1.177 0.125	1.558 0.149	12.008 0.607	-3.66% -6.80%
F42	2.765 0.245	1.577 0.178	2.105 0.227	11.878 0.768	+20.35% +27.52%
T34	2.392 0.222	1.328 0.147	1.750 0.169	12.487 0.705	+6.83% +7.75%
F43	2.916 0.266	1.636 0.178	2.132 0.227	12.504 0.864	+24.95% +33.93%
T23	**1.965** **0.182**	**1.110** 0.121	**1.394** **0.140**	**11.141** 0.572	-10.72% -11.62%
F32	2.124 0.199	1.217 0.131	1.476 0.161	12.034 0.621	-3.65% -2.61%
T04	2.182 0.197	1.282 0.133	1.777 0.170	11.484 0.600	+1.92% -1.94%
T13	2.009 0.183	1.160 0.121	1.563 0.144	11.400 **0.548**	-6.02% -11.87%

Experiments are conducted on the development set (Vox1-O) in the left sub-table and on the test sets (VoX1-H, VoX1-E, SITW, CNCeleb) in the right sub-table. The stride configuration shows the stride factors for time and frequency dimensions in the five stages of ResNet architecture. The symbol ↑↓ indicates the average relative changes across all four test sets over the benchmark model.

TABLE V
PERFORMANCE IN EER(%) AND MINDCF OF THE MODIFIED RESNET34 [19], [50] AND *GOLDEN GEMINI* MODELS WITH DIFFERENT PATHS DEMONSTRATED IN FIG. 3(B)

Index of Stride Config.	Downsampling Factors (α_5, β_5)	Stride Config. [Time] [Frequency]	Params (Million)	FLOPs (Giga) 2s/3s	Vox1-O EER minDCF
MOD	(8,8)	[1,1,2,2,2] [1,1,2,2,2]	6.63	4.63/6.88	1.101 0.128
T14	(2,16)	[1,1,1,1,2] [1,2,2,2,2]	5.98	6.68/9.99	1.058 **0.092**
T14b		[1,1,1,2,1] [1,2,2,2,2]	5.98	4.97/7.43	1.056 0.093
T14c		[1,1,2,1,1] [1,2,2,2,2]	5.98	4.41/6.59	**1.053** **0.092**
T14d		[1,2,1,1,1] [1,2,2,2,2]	5.98	4.18/6.25	1.154 0.115
T23	(4,8)	[1,1,1,2,2] [1,1,2,2,2]	6.63	8.27/12.37	1.101 0.095
T23b		[1,1,2,2,1] [1,1,2,2,2]	6.63	5.45/8.13	1.095 0.099
T23c		[1,1,1,2,2] [1,2,2,2,1]	6.63	4.99/7.45	1.122 0.112
T23d		[1,1,2,1,2] [1,2,2,2,1]	6.63	4.43/6.61	1.095 0.104

Index of Stride Config.	Vox1-H EER minDCF	Vox1-E EER minDCF	SITW EER minDCF	CNCeleb EER minDCF	↑↓ EER minDCF
MOD	2.221 0.208	1.252 0.139	1.584 0.161	12.113 0.623	Benchmark Benchmark
T14	1.998 0.185	1.148 0.120	1.505 0.148	11.670 0.549	-6.75% -11.14%
T14b	2.023 0.190	1.149 0.124	1.531 0.151	11.715 0.559	-5.94% -9.07%
T14c	2.010 0.186	1.157 0.124	1.504 0.146	11.828 **0.540**	-6.13% -10.99%
T14d	2.040 0.189	1.169 0.128	1.531 0.155	11.867 0.569	-5.04% -7.35%
T23	**1.965** **0.182**	**1.110** 0.121	**1.394** **0.140**	**11.141** 0.572	**-10.72%** **-11.62%**
T23b	1.992 0.184	1.134 0.125	1.449 **0.140**	11.315 0.588	-8.70% -10.08%
T23c	2.010 0.182	1.137 **0.118**	1.476 0.146	12.014 0.589	-6.59% -10.48%
T23d	2.017 0.188	1.135 0.120	1.581 0.144	11.805 0.584	-5.32% -9.94%

Experiments are conducted on the development set (VoX1-O) in the left sub-table and on the test sets (VoX1-H, VoX1-E, SITW, CNCeleb) in the right sub-table. The stride configuration shows the stride factors for time and frequency dimensions in the five stages of ResNet architecture. The symbol ↑↓ indicates the average relative changes across all four test sets compared to the benchmark model.

a smaller width, it is advisable to avoid the stride configurations located on the boundary.

Observation 3: Leveraging an optimal stride configuration effectively utilizes the computational resources of model size and FLOPs. The trellis diagram in Fig 3(a) clearly shows that models in the lower right region with large FLOPs and model size perform poorly. Even the largest model (indexed as F50), which employs an unfavorable stride configuration, performs the worst. On the contrary, models that preserve temporal resolution achieve good performance with less complexity than the baseline.

Observation 4: Models indexed as T14 and T23 show the best performance among all the models. This supports the *Golden-Gemini Hypothesis*that there exist operational states that yield optimal performance for ASV. We refer to these two endpoints on the trellis diagram representing a pair of optimal operational states as the *Golden Gemini*.

Points closer to the start points are not explored for two reasons. Firstly, T13 shows inferior performance compared to *Golden Gemini*. Secondly, points closer to the start points would significantly increase computational complexity, resulting in decreased efficiency compared to utilizing a deeper model with the proposed *Golden-Gemini* stride configuration.

In the right sub-table of Table IV, the testing results for all stride configurations are presented. It is evident that across all four testing sets, covering in-domain, out-domain, large-scale, and hard-case scenarios, the testing results exhibit a consistent trend similar to that observed in the development set. This consistency strongly supports the above observations.

C. Evaluation on Different Paths Towards Golden Gemini

There are multiple paths leading to the *Golden Gemini*, as shown in Fig. 3(b), each representing a stride configuration. As discussed in Section III-B, we further investigate these paths to assess the impact of early or late downsampling. The experimental results of the development set are presented in the left sub-table of Table V. Following are the two observations:

Observation 5: All paths towards the Golden Gemini points outperform the baseline (indexed as MOD) that uses an equal-stride configuration. This observation supports *Golden-Gemini Hypothesis*that the pair of operational states engage in competition and yield optimal performance.

Observation 6: Different path options offer the flexibility to trade off between FLOPs and performance, with increased FLOPs generally resulting in improved results. This flexibility in model design allows for better adaptation to specific application scenarios. In addition, previous work demonstrates superiority by comparing FLOPs as a metric [24], [25], [40], [41], [50], [70], [76], [94], [95]. This practice is based on the common understanding that bigger FLOPs often correlate with better performance. However, rather than simply increasing FLOPs, experimental results show that an optimal stride configuration utilizes FLOPs more efficiently.

The testing results reported in the right sub-table of Table V demonstrate a consistent trend similar to that observed on the development set. This further verifies the two observations mentioned above. In addition, all the stride configurations depicted in both trellis diagrams in Fig. 3 are visualized in Fig. 4, comparing their performance, model size, and FLOPs. Among these configurations, the *Golden-Gemini* T14c achieves average EER/minDCF reductions of 5.78%/14.37% over the modified ResNet baseline (indexed as MOD) across all four test sets while reducing the model size by 9.8% and the computational complexity by 4.2%. Considering the efficacy and efficiency, we designate the T14c stride configuration as the principal stride configuration in this work. Networks that adopt the proposed *Golden-Gemini* stride configurations are referred to as the *Gemini* networks, such as the *Gemini ResNet*. The structure comparison of the proposed *Gemini* ResNet with T14c stride

TABLE VI
PERFORMANCE IN EER(%) AND MINDCF OF DIFFERENT SIZES OF RESNET MODELS WITH EQUAL-STRIDE CONFIGURATION OR THE PROPOSED *GOLDEN-GEMINI* STRIDE CONFIGURATION (T14C) ON VOXCELEB1, SITW AND CNCELEB TEST SETS

Model	Params (Million)	FLOPs (G) 2s/3s	Vox1-O EER/minDCF	Vox1-H EER/minDCF	Vox1-E EER/minDCF	SITW EER/minDCF	CNCeleb EER/minDCF	Avg. Reduction EER/minDCF
ResNet18	4.11	2.22/3.30	1.760/0.177	2.785/0.244	1.600/0.170	2.132/0.210	12.301/0.657	Benchmark
***Gemini* ResNet18**	**3.45 (-16.1%)**	**2.17/3.25**	**1.319/0.139**	**2.474/0.226**	**1.462/0.153**	**2.050/0.190**	**12.211/0.592**	-9.89%/-11.57%
ResNet50	11.13	5.22/7.76	1.329/0.141	2.213/0.205	1.249/0.134	1.613/0.158	11.856/0.692	Benchmark
***Gemini* ResNet50**	**8.51 (-23.5%)**	**4.92/7.35**	**1.196/0.121**	**2.016/0.189**	**1.147/0.119**	**1.449/0.145**	**11.608/0.603**	-7.87%/-11.22%
ResNet101	15.89	10.07/15.00	1.101/0.100	2.051/0.194	1.121/0.121	1.367/0.140	11.884/0.633	Benchmark
***Gemini* ResNet101**	**13.27 (-16.5%)**	**9.72/14.54**	**0.962/0.099**	**1.836/0.167**	**1.035/0.108**	**1.320/0.125**	**11.625/0.553**	-7.26%/-9.88%

Avg. Reduction indicates averaged relative reduction across all five sets over the benchmark.

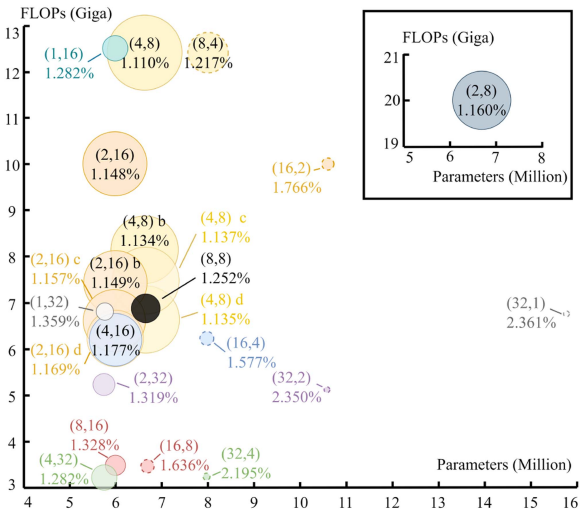


Fig. 4. Performance versus FLOPs and the number of parameters for different stride configurations in Fig. 3. The color is consistent with Fig. 3(a). The size of the bubble indicates the performance in EER (%) on the VoxCeleb-E test set, and the larger the bubble, the better the performance.

configuration, original ResNet [50], and modified ResNet [19], [21] is presented in Table I.

D. Evaluation on Compatibility

The changes in time and frequency resolutions occur once per stage in both ResNet and its variant networks, such as DF-ResNet [24], Res2Net [76], and SD-ResNet [70]. Given that the *Golden Gemini* is concluded from investigating the significance of time and frequency resolution for speaker verification, it is expected to apply to all ResNet series networks that still adhere to the original ResNet's five-stage structural design. This subsection aims to confirm the consistently superior performance of the proposed *Golden-Gemini* stride configuration across various conditions and its compatibility with different techniques. We exemplify with the *Golden Gemini* T14c stride configuration, conducting experiments to compare the models using the *Golden Gemini* T14c and the default equal-stride configuration under the following conditions:

Different model sizes: The ResNet models have different depths, resulting in different model sizes and computational resource requirements. For any proposed new method, adapting to ResNet models of various sizes is important as it allows for trade-offs between performance and complexity, enabling better adaptation to different application scenarios. We further extend the application of the proposed *Golden-Gemini* stride configuration from ResNet34 to a smaller model (ResNet18) and the larger models (ResNet50 and ResNet101). The experimental results are presented in Table VI and Fig. 5. The results demonstrate that the proposed *Golden-Gemini* stride configuration consistently improves the performance by an average of 7.70%/11.76% EER/minDCF reduction across the entire range of model sizes, while reducing parameters and FLOPs by 16.5% and 4.1%, respectively.

Data augmentations: The training of neural networks benefits from data augmentations [87]. All previous experiments are trained with augmented data as described in Section IV-B. We conduct training without data augmentation to assess the compatibility of *Golden Gemini*, and the results are shown in Table VII. It is observed that the *Golden-Gemini* stride configuration achieves an average relative reduction of 8.30%/.88% in EER/minDCF across five sets, and reduces complexity.

Squeeze-and-excitation (SE) attention module [77]: SE is one of the most widely used attention modules. We validate the compatibility of the proposed *Golden-Gemini* stride configuration with SE (reduction ratio $r = 4$), and the results are shown in Table VII. We can observe that the proposed *Golden Gemini* outperforms the equal-stride configuration on most of the test sets, with an average EER/minDCF reduction of 8.00%/10.75%. However, the SE block does not improve the performance, which may require further investigation.

A different backbone network – Res2Net [76]: The proposed *Golden-Gemini* stride configuration is not limited to ResNet models and can be applied to other 2D CNN-based models as well. Res2Net [76] is a well-known 2D CNN-based architecture recognized for its ability to extract multi-scale features. In the design of multi-scale frequency-channel attention TDNN (MFA-TDNN) [40] and multi-scale feature aggregation convolution-augmented transformer (MFA-Conformer) [46], multi-scale features have been proven to benefit ASV. Previous studies have explored the application of Res2Net in ASV [23], [27], [63]. We compare the Res2Net34 model using the default equal-stride configuration with that using the proposed *Golden-Gemini* stride configuration. The scale (s) of Res2Net is set to 4. The results in Table VII show that the *Golden-Gemini* stride configuration improves performance while reducing complexity compared to the equal-stride configuration. Additionally, Res2Net34 shows better performance in ASV compared to ResNet34.

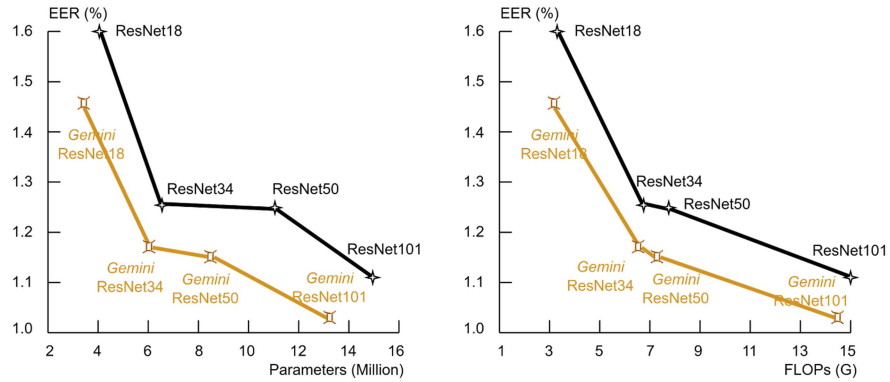


Fig. 5. Performance and complexity comparison of proposed *Gemini* ResNet and modified ResNet [19], [21] with different model sizes on Vox1-E test set.

TABLE VII
PERFORMANCE IN EER(%) AND MINDCF OF THE NETWORKS WITH EQUAL-STRIDE CONFIGURATION AND PROPOSED *GOLDEN-GEMINI* T14C STRIDE CONFIGURATION UNDER DIFFERENT CONDITIONS ON VOXCELEB1, SITW, AND CNCELEB TEST SETS

Model	Aug.	Params (Million)	FLOPs (G) 2s/3s	Vox1-O EER/minDCF	Vox1-H EER/minDCF	Vox1-E EER/minDCF	SITW EER/minDCF	CNCeleb EER/minDCF	Avg. Reduction EER/minDCF
ResNet34 [19]	×	6.63	4.63/6.88	1.489/0.155	2.500/0.224	1.423/0.158	2.378/0.208	12.737/0.639	Benchmark
***Gemini* ResNet34**	×	**5.98**	**4.41/6.59**	**1.375/0.132**	**2.261/0.209**	**1.294/0.139**	**2.102/0.189**	**12.271/0.628**	-8.30%/-8.88%
ResNet34 + SE [77]	✓	6.79	4.63/6.89	1.287/0.141	2.512/0.241	1.370/0.157	1.640/0.187	**12.408/0.688**	Benchmark
***Gemini* ResNet34 + SE [77]**	✓	**6.14**	**4.41/6.60**	**1.053/0.104**	**2.264/0.219**	**1.223/0.143**	**1.531/0.165**	13.078/0.703	-8.00%/-10.75%
Res2Net34 [76]	✓	6.57	4.74/7.05	1.071/0.103	2.073/0.195	1.184/0.129	1.524/0.157	11.571/**0.568**	Benchmark
***Gemini* Res2Net34**	✓	**5.92**	**4.46/6.68**	**1.048/0.100**	**1.914/0.176**	**1.092/0.115**	**1.422/0.135**	**11.135**/0.577	-5.60%/-7.18%
ResNet34 + xi [15]	✓	7.30	4.66/6.93	1.159/0.111	2.192/0.215	1.288/0.136	1.602/0.161	12.627/0.652	Benchmark
***Gemini* ResNet34 + xi [15]**	✓	**6.31**	**4.47/6.69**	**1.101/0.105**	**2.100/0.199**	**1.169/0.124**	**1.480/0.149**	**11.902/0.644**	-6.37%/-6.07%
SD-ResNet38 [70]	✓	7.37	5.20/7.74	1.202/0.130	2.133/0.203	1.187/0.131	1.586/0.161	11.580/0.618	Benchmark
***Gemini* SD-ResNet38 [70]**	✓	**6.72**	**4.97/7.43**	**1.085/0.099**	**1.974/0.185**	**1.130/0.117**	**1.523/0.147**	**11.507/0.553**	-5.32%/-12.60%

Aug. Indicates whether the system is trained with data augmentations. Avg. reduction means the average relative reduction across all five sets over the benchmark.

TABLE VIII
STRUCTURE COMPARISON BETWEEN DF-RESNET [24] WITH DEFAULT EQUAL-STRIDE CONFIGURATION AND THE PROPOSED *GOLDEN-GEMINI* STRIDE CONFIGURATION

Stage	Layer	DF-ResNet182[4] Stride	DF-ResNet182[4] Output	*Gemini* DF-ResNet183[4] Stride	*Gemini* DF-ResNet183[4] Output
conv1	3×3, 32	(1,1)	32×F×T	(1,1)	32×F×T
conv2	3×3, 32 (SD) $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128(dw) \\ 1\times1, 32 \end{bmatrix}\times3$	- (1,1)	- 32×F×T	(2,1) (1,1)	32×F/2×T 32×F/2×T
conv3	3×3, 64 (SD) $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256(dw) \\ 1\times1, 64 \end{bmatrix}\times8$	(2,2) (1,1)	64×F/2×T/2 64×F/2×T/2	(2,2) (1,1)	64×F/4×T/2 64×F/4×T/2
conv4	3×3, 128 (SD) $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512(dw) \\ 1\times1, 128 \end{bmatrix}\times45$	(2,2) (1,1)	128×F/2×T/2 128×F/4×T/4	(2,1) (1,1)	128×F/4×T/2 128×F/8×T/2
conv5	3×3, 256 (SD) $\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024(dw) \\ 1\times1, 256 \end{bmatrix}\times3$	(2,2) (1,1)	256×F/2×T/2 256×F/8×T/8	(2,1) (1,1)	256×F/4×T/2 256×F/16×T/2
Temporal Statistics Pooling Layer		N/A	256×F/8×2	N/A	256×F/16×2
Fully Connected Layer		(5120, 256)		(2560, 256)	
# Parameters		9.84×10^6		$\mathbf{9.20}\times10^6$	
FLOPs (2s / 3s)		8.64×10^9 / 12.87×10^9		$\mathbf{8.25}\times10^9$ / $\mathbf{12.34}\times10^9$	

SD and DW indicate separate downsampling and depth-wise convolution, respectively [24].

A different temporal aggregation layer – xi posterior inference (xi) [15]: As introduced in Section I, an embedding extractor network consists of three components – an encoder, a temporal aggregation layer, and a decoder. The previous experiments focus on the encoder component, and for a fair comparison, a default temporal statistics pooling [12] is applied across all experiments. We further validate the compatibility of the

TABLE IX
PERFORMANCE IN EER(%) AND MINDCF OF THE PROPOSED *GOLDEN GEMINI* DF-RESNET AND SOTA SYSTEMS ON VOXCELEB1 TEST SETS

System	Para.	Vox1-O EER	Vox1-O minDCF	Vox1-E EER	Vox1-E minDCF	Vox1-H EER	Vox1-H minDCF
Res2Net-14w8s [23]	5.6	1.60	0.178	1.60	0.184	2.83	0.280
ResNet18 [25]	4.11	1.48	0.174	1.52	0.175	2.72	0.244
ECAPA-TDNN (C=512) [32]	6.2	1.01	0.127	1.24	0.142	2.32	0.218
MFA-TDNN (Lite) [40]	5.93	0.968	0.091	1.138	0.121	2.174	0.199
DF-ResNet56[4] [24], [25]	4.49	0.96	0.103	1.09	0.122	1.99	0.184
DF-ResNet59[4] (re-implemented)	4.69	0.973	0.097	1.060	0.120	1.866	0.175
***Gemini* DF-ResNet60[4]**	**4.05**	**0.941**	**0.089**	**1.051**	**0.116**	**1.799**	**0.166**
E-TDNN [32]	6.8	1.49	0.160	1.61	0.171	2.69	0.242
Res2Net-26w8s [23]	9.3	1.45	0.147	1.47	0.169	2.72	0.272
ResNet34 [25]	6.63	0.96	0.089	1.01	0.121	1.86	0.177
H/ASP AP+softmax [28]	8.0	0.88	-	1.07	-	2.21	-
MFA-TDNN (Standard) [40]	7.32	0.856	0.092	1.083	0.118	2.049	0.190
RecXi with $\mathcal{L}_{\text{ssp}}$ [72]	7.06	0.984	0.091	1.075	0.114	1.857	0.179
PCF-ECAPA (C=512) [44]	8.9	0.718	0.086	0.792	0.114	1.802	0.175
CAM++ [41]	7.18	0.73	0.091	0.89	0.100	1.76	0.173
DF-ResNet110[4] [24], [25]	6.98	0.75	0.070	0.88	0.100	1.64	0.156
***Gemini* DF-ResNet114[4]**	**6.53**	**0.686**	**0.067**	**0.863**	**0.097**	**1.490**	**0.144**
E-TDNN (large) [32]	20.4	1.26	0.140	1.37	0.149	2.35	0.215
ResNet18 [32]	13.8	1.47	0.177	1.60	0.179	2.88	0.267
ResNet34 [32]	23.9	1.19	0.159	1.33	0.156	2.46	0.229
ECAPA-TDNN (C=1024) [32]	14.7	0.87	0.107	1.12	0.132	2.12	0.210
MFA-Conformer (1/2) [46]	20.5	0.64	0.081	1.29	0.137	1.63	0.153
P-vectors (SFA) [47]	15.1	0.856	0.120	1.117	0.120	2.112	0.208
ResNet52-C2D-32 [97]	10.34	0.771	0.107	0.939	0.111	1.816	0.180
SKA-TDNN [42]	34.9	0.78	-	0.90	-	1.74	-
SimAM-ResNet34 (GSP) [29]	21.54	0.718	0.071	0.993	0.103	1.647	0.159
DS-TDNN-L [98]	20.5	0.64	0.082	0.93	0.112	1.55	0.149
PCF-ECAPA (C=1024) [44]	22.2	0.718	0.089	0.891	0.102	1.707	0.175
NEMO [43]	15.88	0.74	0.110	0.90	0.105	1.90	0.189
Branch-ECAPA-TDNN(b) [99]	24.11	0.72	0.084	0.92	0.098	1.69	0.166
ECAPA++ (Big) [100]	23.9	0.65	0.080	0.84	0.098	1.54	0.154
DF-ResNet179[4] [24], [25]	9.84	0.62	**0.061**	**0.80**	**0.090**	1.51	0.148
***Gemini* DF-ResNet183[4]**	**9.20**	**0.596**	0.064	0.806	**0.090**	**1.440**	**0.137**

Models with our proposed golden-gemini stride configuration are highlighted in grey[5].

proposed *Golden Gemini* with another temporal aggregation method – xi posterior inference, which is designed to estimate uncertainty [15]. The experimental results shown in Table VII demonstrate the consistently superior performance of *Golden Gemini* over the equal-stride configuration while reducing the model size by 13.6% and FLOPs by 4.1%.

A micro design – separate downsampling (SD) [70]: Unlike ResNet [50], which performs downsampling at the first 2D CNN layer in each stage, Swin Transformer [94] introduces a separate downsampling layer between stages. This micro design is also extended to ResNet, resulting in notable improvements [70]. In this work, we explore this micro design for ASV by implementing four 3×3 2D CNN layers between the five stages and name SD-ResNet. It is worth noting that this modification adds four additional 2D CNN layers, resulting in the expansion of the ResNet**34** [19], [21] architecture to SD-ResNet**38**. The results in Table VII demonstrate that SD-ResNet outperforms the modified ResNet [21] (indexed as MOD in Table IV). Moreover, the integration of *Golden Gemini* leads to additional improvements in terms of EER/minDCF, with averaged reductions of 5.32% and 12.60%, respectively.

In summary, the experimental results validate the compatibility of the proposed *Golden-Gemini* stride configuration with various existing techniques and training conditions. *Golden Gemini* consistently improves performance while reducing complexity. Its superiority can be attributed to the importance of temporal resolution. By maintaining temporal resolution, *Golden Gemini* ensures adequate representations of both vocal tract features and learned speaker characteristics across various scales of local time regions and is expected to further benefit the temporal aggregation layer, leading to significant performance improvements.

E. New SOTA Benchmark

DF-ResNet [24], [25] is a series of powerful SOTA models introduced in Section II-B. We first re-implement a small venison, namely DF-ResNet59. The results reported in Table IX demonstrate that the re-implemented DF-ResNet model slightly outperforms the one reported in [24], [25]. Notably, we do not apply SpecAugment [87] which is used in [24], [25]. SpecAugment has been proven effective in automatic speech recognition (ASR) [87]. However, it can have adverse effects on the fundamental frequency of the audio, which is a critical characteristic for speaker discrimination [96]. Prior work [21] shows that combining SpecAugment with other augmentation methods in ASV can pose compatibility challenges. Our experiments demonstrate a similar trend.

[4]For the re-implemented DF-ResNet and proposed *Gemini* DF-ResNet models, we count the separate downsampling layers as part of the total layer count. This differs from the counting method used in the original DF-ResNet [24], [25]. As an example, the DF-ResNet**179** in [24], [25] is referred to as DF-ResNet**182** in this work. However, for the experimental results cited in Table IX, we follow the original work [24], [25].

[5]Pre-trained models and codes of the proposed *Gemini* DF-ResNet are available at https://github.com/Tianchi-Liu9/Golden-Gemini-for-Speaker-Verification and https://github.com/wenet-e2e/wespeaker.

Similar to other ResNet models, DF-ResNet adopts the default equal-stride configuration, treating temporal and frequency dimensions equally. By replacing the stride configuration with our proposed *Golden-Gemini* T14c, we see a notable 4.9% average performance boost and a 7.6% reduction in model size, as detailed in Table IX. Also, Table VIII shows a 4.5% decrease in FLOPs. It's important to note that DF-ResNet, our chosen baseline, achieves SOTA performance with a relatively small model size, emphasizing its meticulous design and efficiency. In this context, achieving further performance gains becomes challenging given the already very low EER and minDCF. Further analysis indicates that, as the model size increases from the smallest to largest, relative performance improvements decrease from 7.4% to 5.8%, and then to 1.7%. This trend aligns with the inherent difficulty of achieving significant performance improvements over a robust baseline and low EER/minDCF. Nevertheless, our proposed *Golden-Gemini* stride configuration still brings improvements, securing the best performance among all systems. This outcome supports the remarkable capabilities of the *Golden-Gemini* stride configuration and *Golden-Gemini Hypothesis*, which emphasizes the critical significance of temporal resolution in attaining superior results in ASV.

F. Golden-Gemini *Guiding Principles*

The experiments conducted above consistently demonstrate the superiority of the proposed *Golden Gemini* over the default equal-stride configuration from various perspectives. The underlying logic behind the *Golden Gemini* is the utilization of a series of guiding principles that align with the natural properties of speech signals for designing 2D CNN-based networks for ASV. Based on the aforementioned observations, we summarize the *Golden-Gemini* guiding principles as follows:

- Preserve sufficient temporal resolutions during the feature representation instead of preserving the frequency resolution.
- Avoid frequently diminishing any dimension at the early stage when using a narrow network.
- A correct stride configuration surpasses mere FLOP increments. Prioritize the adoption of the optimal stride configuration followed by the trade-off between FLOPs and performance according to computation resources.

VI. Conclusion

We investigate efficient stride configurations for speaker verification. Through a strategic search on a trellis diagram, we analyze the impact of temporal and frequency resolution on the ASV performance. Experimental results on the VoxCeleb, SITW, and CNCeleb test sets highlight the significance of the temporal resolution. This leads us to identify two points, named *Golden Gemini*, representing two series of optimal stride configurations for ASV. We also present a set of guiding principles that comprehensively describe the *Golden Gemini* for designing 2D ResNet for ASV. Further experiments demonstrate the consistent superiority and excellent compatibility of the proposed *Golden Gemini* with various structures across different conditions. Moreover, our approach is simple yet effective and

can be easily applied to any 2D ResNet architecture style, offering improved performance while reducing model complexity. Based on the *Golden-Gemini* guiding principles, we introduce a powerful benchmark for ASV, namely the *Gemini* DF-ResNet. These findings indicate the promising value of our method in real-world applications. Additionally, the significance of time and frequency resolutions may extend beyond speaker verification, holding great potential for related tasks such as speaker diarization, speaker extraction, emotion recognition, and speech anti-spoofing.

REFERENCES

[1] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Commun.*, vol. 60, pp. 56–77, 2014.

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

[3] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2016, pp. 165–170.

[4] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[5] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 4832–4835.

[6] Q. Wang, K. Okabe, K. A. Lee, and T. Koshinaka, "Generalized domain adaptation framework for parametric back-end in speaker recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 3936–3947, 2023.

[7] Q. Wang, K. A. Lee, and T. Liu, "Scoring of large-margin embeddings for speaker verification: Cosine or PLDA?," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 600–604.

[8] L. Ferrer, M. McLaren, and N. Brümmer, "A speaker verification backend with robust performance across conditions," *Comput. Speech Lang.*, vol. 71, 2022, Art. no. 101258.

[9] A. Sholokhov, N. Kuzmin, K. A. Lee, and E. S. Chng, "Probabilistic back-ends for online speaker recognition and clustering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[10] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3214–3218.

[11] S. Wang, Y. Yang, Y. Qian, and K. Yu, "Revisiting the statistics pooling layer in deep speaker embedding learning," in *Proc. IEEE 12th Int. Symp. Chin. Spoken Lang. Process.*, 2021, pp. 1–5.

[12] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 999–1003.

[13] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2252–2256.

[14] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 3573–3577.

[15] K. A. Lee, Q. Wang, and T. Koshinaka, "Xi-vector embedding for speaker recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 1385–1389, 2021.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.

[17] Z. Bai, J. Wang, X.-L. Zhang, and J. Chen, "End-to-end speaker verification via curriculum bipartite ranking weighted binary cross-entropy," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 1330–1344, 2022.

[18] J. S. Chung et al., "In defence of metric learning for speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2977–2981.

[19] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," 2019, *arXiv:1910.12592*.

[20] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5791–5795.

[21] H. Wang et al., "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[22] X. Miao, I. McLoughlin, W. Wang, and P. Zhang, "D-MONA: A dilated mixed-order non-local attention network for speaker and language recognition," *Neural Netw.*, vol. 139, pp. 201–211, 2021.

[23] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 301–307.

[24] B. Liu, Z. Chen, and Y. Qian, "Depth-first neural architecture with attentive feature fusion for efficient speaker verification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1825–1838, 2023.

[25] B. Liu, Z. Chen, S. Wang, H. Wang, B. Han, and Y. Qian, "DF-ResNet: Boosting speaker verification performance with depth-first design," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 296–300.

[26] W. Lin and M.-W. Mak, "Robust speaker verification using deep weight space ensemble," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 802–812, 2023.

[27] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, and J. Qi, "An enhanced Res2Net with local and global feature fusion for speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 2228–2232.

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

[29] X. Qin, N. Li, C. Weng, D. Su, and M. Li, "Simple attention module based speaker verification with iterative noisy label detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6722–6726.

[30] H. Shen et al., "Improving fairness in speaker verification via group-adapted fusion network," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 7077–7081.

[31] C. Zeng, X. Wang, E. Cooper, X. Miao, and J. Yamagishi, "Attention back-end for automatic speaker verification with multiple enrollment utterances," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6717–6721.

[32] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3830–3834.

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

[34] Y. Ma, K. A. Lee, V. Hautamäki, and H. Li, "PL-EESR: Perceptual loss based end-to-end robust speaker representation extraction," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2021, pp. 106–113.

[35] R. Tao, K. A. Lee, Z. Shi, and H. Li, "Speaker recognition with two-step multi-modal deep cleansing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[36] R. K. Das, R. Tao, J. Yang, W. Rao, C. Yu, and H. Li, "HLT-NUS submission for 2019 NIST multimedia speaker recognition evaluation," in *Proc. IEEE Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2020, pp. 605–609.

[37] B. Han, Z. Chen, and Y. Qian, "Local information modeling with self-attention for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6727–6731.

[38] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 941–945.

[39] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in TDNNs and frequency positional information in 2D ResNets to enhance speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2302–2306.

[40] T. Liu, R. K. Das, K. A. Lee, and H. Li, "MFA: TDNN with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 7517–7521.

[41] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "CAM++ : A fast and efficient network for speaker verification using context-aware masking," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 5301–5305.

[42] S. H. Mun, J.-w. Jung, M. H. Han, and N. S. Kim, "Frequency and multi-scale selective kernel attention for speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2023, pp. 548–554.

[43] D. Cai, W. Wang, M. Li, R. Xia, and C. Huang, "Pretraining conformer with ASR for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[44] Z. Zhao, Z. Li, W. Wang, and P. Zhang, "PCF: ECAPA-TDNN with progressive channel fusion for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[45] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Neural acoustic-phonetic approach for speaker verification with phonetic attention mask," *IEEE Signal Process. Lett.*, vol. 29, pp. 782–786, 2022.

[46] Y. Zhang et al., "MFA-Conformer: Multi-scale feature aggregation conformer for automatic speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 306–310.

[47] X. Wang, F. Wang, B. Xu, L. Xu, and J. Xiao, "P-vectors: A parallel-coupled TDNN/transformer network for speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 3182–3186.

[48] A. Brown, J. Huh, J. S. Chung, A. Nagrani, D. Garcia-Romero, and A. Zisserman, "VoxSRC 2021: The third VoxCeleb speaker recognition challenge," 2022, *arXiv:2201.04583*.

[49] J. Huh et al., "VoxSRC 2022: The fourth VoxCeleb speaker recognition challenge," 2023, *arXiv:2302.10248*.

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

[51] R. Makarov, N. Torgashov, A. Alenin, I. Yakovlev, and A. Okhotnikov, "ID R&D system description to VoxCeleb Speaker Recognition Challenge 2022," 2022.

[52] Z. Zhao, Z. Li, W. Wang, and P. Zhang, "The HCCL system for VoxCeleb speaker recognition challenge 2022," 2023, *arXiv:2305.12642*.

[53] D. Cai and M. Li, "The DKU-DukeECE system for the self-supervision speaker verification task of the 2021 VoxCeleb speaker recognition challenge," 2021, *arXiv:2109.02853*.

[54] M. Zhao, Y. Ma, M. Liu, and M. Xu, "The SpeakIn system for VoxCeleb speaker recognition challenge 2021," 2021, *arXiv:2109.01989*.

[55] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "SpEx+ : A complete time domain speaker extraction network," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 1406–1410.

[56] Z. Pan, R. Tao, C. Xu, and H. Li, "Selective listening by synchronizing speech with lips," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1650–1664, 2022.

[57] X. Wang, N. Pan, J. Benesty, and J. Chen, "On multiple-input/binaural-output antiphasic speaker signal extraction," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[58] Y. Jiang, R. Tao, Z. Pan, and H. Li, "Target active speaker detection with audio-visual cues," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 3152–3156.

[59] Z. Pan, W. Wang, M. Borsdorf, and H. Li, "ImagineNet: Target speaker extraction with intermittent visual cue through embedding inpainting," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[60] W. Wang, X. Qin, and M. Li, "Cross-channel attention-based target speaker voice activity detection: Experimental results for the m2met challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9171–9175.

[61] M. Cheng, W. Wang, Y. Zhang, X. Qin, and M. Li, "Target-speaker voice activity detection via sequence-to-sequence prediction," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[62] R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou, and H. Li, "Is someone speaking? Exploring long-term temporal features for audio-visual active speaker detection," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 3927–3935.

[63] X. Xiao et al., "Microsoft speaker diarization system for the VoxCeleb speaker recognition challenge 2020," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 5824–5828.

[64] J.-W. Jung et al., "In search of strong embedding extractors for speaker diarisation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[65] R. Yan, C. Wen, S. Zhou, T. Guo, W. Zou, and X. Li, "Audio deepfake detection system with neural stitching for add 2022," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9226–9230.

[66] X. Chen, J. Wang, X.-L. Zhang, W.-Q. Zhang, and K. Yang, "LMD: A learnable mask network to detect adversarial examples for speaker verification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2476–2490, 2023.

[67] B. Huang, S. Cui, J. Huang, and X. Kang, "Discriminative frequency information learning for end-to-end speech anti-spoofing," *IEEE Signal Process. Lett.*, vol. 30, pp. 185–189, 2023.

[68] R. K. Das, J. Yang, and H. Li, "Data augmentation with signal companding for detection of logical access attacks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6349–6353.

[69] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.

[70] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.

[71] L. Tóth, "Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 190–194.

[72] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Disentangling voice and content with self-supervision for speaker recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 50221–50236.

[73] T. Liu, R. K. Das, M. Madhavi, S. Shen, and H. Li, "Speaker-utterance dual attention for speaker and utterance verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4293–4297.

[74] T. Liu, M. Madhavi, R. K. Das, and H. Li, "A unified framework for speaker and utterance verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 4320–4324.

[75] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[76] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[77] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[78] J. Campbell, "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep. 1997.

[79] D. Rentzos, S. Vaseghi, E. Turajlic, Q. Yan, and C.-H. Ho, "Transformation of speaker characteristics for voice conversion," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2003, pp. 706–711.

[80] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 2616–2620.

[81] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 1086–1090.

[82] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 818–822.

[83] Y. Fan et al., "CN-Celeb: A challenging chinese speaker recognition dataset," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7604–7608.

[84] M. Ravanelli et al., "SpeechBrain: A general-purpose speech toolkit," 2021, *arXiv:2106.04624*.

[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[86] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 464–472.

[87] D. S. Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2613–2617.

[88] H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, "Speaker augmentation and bandwidth extension for deep speaker embedding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 406–410.

[89] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 5220–5224.

[90] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Process. Speaker Lang. Recognit. Workshop*, 2010, Art. no. 14.

[91] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019.

[92] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 1567–1571.

[93] Q. Wang, K. A. Lee, and T. Liu, "Incorporating uncertainty from speaker embedding estimation to speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[94] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[95] Q. Li, L. Yang, X. Wang, X. Qin, J. Wang, and M. Li, "Towards lightweight applications: Asymmetric enroll-verify structure for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 7067–7071.
[96] F. Tong et al., "ASV-Subtools: Open source toolkit for automatic speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6184–6188.
[97] J. Li, Y. Tian, and T. Lee, "Convolution-based channel-frequency attention for text-independent speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[98] Y. Li, J. Gan, and X. Lin, "DS-TDNN: Dual-stream time-delay neural network with global-aware filter for speaker verification," 2023, *arXiv:2303.11020*.
[99] J. Yao, C. Liang, Z. Peng, B. Zhang, and X.-L. Zhang, "Branch-ECAPA-TDNN: A parallel branch architecture to capture local and global features for speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 1943–1947.
[100] B. Liu and Y. Qian, "ECAPA++ : Fine-grained deep embedding learning for TDNN based speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc*, 2023, pp. 3132–3136.

Tianchi Liu (Student Member, IEEE) received the M.Sc. degree in 2019 from the National University of Singapore, Singapore where he is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering. He is also a Senior Research Engineer with the Institute for Infocomm Research (I^2R), Agency for Science, Technology, and Research (A*STAR), Singapore. His research interests include speaker recognition, speech anti-spoofing, audio-visual representation learning, large language model, and speech foundation model.

Kong Aik Lee (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. From 2006 to 2018, he was a Research Scientist and then a Strategic Planning Manager (concurrent appointment) with the Institute for Infocomm Research, Singapore. From 2018 to 2020, he was a Senior Principal Researcher with the Data Science Research Laboratories, NEC Corporation, Tokyo, Japan. He was an Associate Professor with the Singapore Institute of Technology, Singapore, while holding a concurrent appointment as a Principal Scientist and a Group Leader with the Agency for Science, Technology and Research (A*STAR), Singapore. He is currently an Associate Professor with the Hong Kong Polytechnic University, Hong Kong. His research interests include the automatic and para-linguistic analysis of speaker characteristics, ranging from speaker recognition, language and accent recognition, voice biometrics, spoofing, and countermeasures. From 2017 to 2021, he was an Associate Editor for IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. Since 2016, he has been an Editorial Board Member of Elsevier *Computer Speech and Language*. He is an elected Member of the IEEE Speech and Language Processing Technical Committee and was the General Chair of the Speaker Odyssey 2020 Workshop. He was the recipient of the Singapore IES Prestigious Engineering Achievement Award 2013 and the Outstanding Service Award by IEEE ICME 2020.

Qiongqiong Wang (Member, IEEE) received the B.E. degree from the Undergraduate School of Physics, Shanghai Jiao Tong University, Shanghai, China, in 2011, and the M.E. degree in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 2013. From 2013 to 2021, she was a Researcher with the Biometrics Research Laboratories, NEC Corporation, Tokyo, Japan. She is currently a Lead Research Engineer with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. Her research interests include speaker recognition, deception detection, emotion recognition, speech anti-spoofing, speech enhancement, large language model, and speech foundation model.

Haizhou Li (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronic engineering from the South China University of Technology, Guangzhou, China, in 1984, 1987, and 1990, respectively. He was with The University of Hong Kong, Hong Kong, during 1988–1990, and South China University of Technology, during 1990–1994. He was a Visiting Professor with CRIN, France, during 1994–1995, Research Manager with the AppleISS Research Centre during 1996–1998, the Research Director with Lernout & Hauspie Asia Pacific during 1999–2001, the Vice President with InfoTalk Corporation Ltd. during 2001–2003, and Principal Scientist and Department Head of human language technology with the Institute for Infocomm Research, Singapore, during 2003–2016. He is currently a Presidential Chair Professor and the Executive Dean of the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. He is also an Adjunct Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include automatic speech recognition, speaker and language recognition, and natural language processing. Dr. Li was the Editor-in-Chief of IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING during 2015–2018, an elected Member of IEEE Speech and Language Processing Technical Committee during 2013–2015, the President of the International Speech Communication Association during 2015–2017, President of Asia Pacific Signal and Information Processing Association during 2015–2016, and President of Asian Federation of Natural Language Processing during 2017–2018. Since 2012, he has been a Member of the Editorial Board of *Computer Speech and Language*. He was the General Chair of ACL 2012, INTERSPEECH 2014, ASRU 2019 and ICASSP 2022. He was named one of the two Nokia Visiting Professors in 2009 by the Nokia Foundation, and U Bremen Excellence Chair Professor in 2019. He is a Fellow of the ISCA, and a Fellow of the Academy of Engineering Singapore. He was the recipient of the National Infocomm Award 2002, and President's Technology Award 2013 in Singapore.