applied sciences

MDPI

Article

Self Attention Networks in Speaker Recognition

Pooyan Safari *, Miquel India and Javier Hernando

TALP Research Center, Universitat Politecnica de Catalunya, 08034 Barcelona, Spain; miquel.angel.india@upc.edu (M.I.); javier.hernando@upc.edu (J.H.)
* Correspondence: pooyan.safari@upc.edu

Abstract: Recently, there has been a significant surge of interest in Self-Attention Networks (SANs) based on the Transformer architecture. This can be attributed to their notable ability for parallelization and their impressive performance across various Natural Language Processing applications. On the other hand, the utilization of large-scale, multi-purpose language models trained through self-supervision is progressively more prevalent, for tasks like speech recognition. In this context, the pre-trained model, which has been trained on extensive speech data, can be fine-tuned for particular downstream tasks like speaker verification. These massive models typically rely on SANs as their foundational architecture. Therefore, studying the potential capabilities and training challenges of such models is of utmost importance for the future generation of speaker verification systems. In this direction, we propose a speaker embedding extractor based on SANs to obtain a discriminative speaker representation given non-fixed length speech utterances. With the advancements suggested in this work, we could achieve up to 41% relative performance improvement in terms of EER compared to the naive SAN which was proposed in our previous work. Moreover, we empirically show the training instability in such architectures in terms of rank collapse and further investigate the potential solutions to alleviate this shortcoming.

Keywords: speaker recognition; self-attention networks; transformer; speaker embeddings

Citation: Safari, P.; India, M.; Hernando, J. Self Attention Networks in Speaker Recognition. *Appl. Sci.* **2023**, *13*, 6410. https://doi.org/10.3390/app13116410

Academic Editor: Douglas O'Shaughnessy

Received: 24 April 2023
Revised: 17 May 2023
Accepted: 22 May 2023
Published: 24 May 2023

1. Introduction

The use of Deep Learning (DL) to extract a single low dimensional vector representation of the speaker characteristics from speech signal has become the de facto technique in many Speaker Recognition (SR) systems. These vector representations, often called speaker embeddings, can be constructed in an unsupervised (e.g., [1,2]), supervised (e.g., [3–6]), and self-supervised (e.g., [7–9]) regime. In the supervised approach, using speaker labels from the development set, a neural network is typically trained for classification purposes. Then, during the inference, the output layer is dropped, the network is fed by a sequence of feature vectors from an unknown speaker, and the combined representation from one or more hidden layers is considered as the speaker embedding [4,5].

On the other hand, the success of sequence-to-sequence (seq2seq) models in tasks like Neural Machine Translation (NMT) or Automatic Speech Recognition (ASR) is mostly due to attention mechanisms [10–12]. These techniques are used in seq2seq models to assist the decoder in determining which portion of the sequence has to be translated or recognized. There are different methods to apply the attention mechanism, one of which is self-attention. In self-attention, the attention mechanism is applied to every position of the input itself. This kind of attention has shown promising results in a variety of tasks, such as NMT [11], semantic role labeling [13], language representations [14], and recently in vision [15] for image recognition. The use of self-attention in networks like Transformer [11] has become very common because of its strong parallelization capacity for computation and adaptability in modeling dependencies regardless of distance. Multi-head attention is a recently developed attention mechanism that was first put forth in [11] for

a Transformer architecture and showed great promise in a number of downstream tasks, including [11,15–17].

During the last few years, discriminative models based on attention for speaker embedding extraction became popular in the SR community. Typically, this attention mechanism is employed in the pooling layer as a substitute for statistical pooling. Its purpose is to extract a single, condensed vector representation of the speaker characteristics from a speech signal [6]. Attention is utilized in [18–20] to consolidate hidden states of a Recurrent Neural Network (RNN) and create speaker embeddings for text-dependent speaker verification. A unified approach that combines attention for both speaker and language recognition is presented in [21]. In [22], a multi-attention mechanism is introduced, inspired by [23], which employs multiple attention models to capture diverse information from encoder features. Additional studies such as [24,25] have investigated different adaptations of multi-head attention for the pooling layer. Even though there are various attempts in employing the self-attention mechanism for speaker embedding extraction, only a few works tried to show the efficiency of models that are solely relied on Self-Attention Networks (SANs) to extract speaker embeddings [26,27].

Despite the attempts in employing the SANs, there are difficulties in applying these architectures to different downstream tasks. There are several studies to enhance the efficiency of Multi-Head Attentions (MHAs), such as [28,29]. It is usually believed that the efficiency of transformer-based architectures is rooted in their ability to jointly attend multiple positions mainly due to the MHA mechanism. However, there are multiple studies that try to answer this question, is MHA really necessary for SANs? Michel et al. in [30] observed that in models trained with multiple attention heads, a considerable number of heads can be removed during the inference without substantially affecting the model's performance. Interestingly, it was shown that in practice some layers can even be reduced to a single head. In another attempt, Voita et al. in [31] showed that for an NMT task, translation relies on only a limited number of heads that are crucial. They also conclude that significant heads possess specific and interpretable functionalities in the model. These functionalities are associated with attention toward adjacent words and tokens in specific syntactic dependency relations. It is also demonstrated in [32], that attending multiple positions at once is not exclusively a characteristic of MHA. They have shown that multi-layer single-head attention also attends multiple positions and yields better results. However, they claimed that the primary advantage of multi-head attention lies in its training stability, as it requires fewer layers compared to single-head attention while attending the same number of positions. For instance, a 24-layer 16-head Transformer (BERT-large) and a 384-layer single-head Transformer have equivalent total attention head numbers and about the same model size, but the multi-head approach is considerably less deep. Since due to the recent advances in DL, the obstacle of training difficulty is no longer a limiting factor, it is possible to achieve consistent performance improvement without tuning hyper-parameters by going for a deeper single-head Transformer. In other words, single head attention with more layers is more efficient than MHA with less number of layers.

However, stacking more layers in SAN-based architectures introduces another common problem in neural networks, namely rank collapse. It was demonstrated in [33] that in the absence of residual connections, the rank of the sequence representation in Transformer collapses in a doubly exponential manner with increasing depth. In the presence of a rank collapse, the model fails to differentiate between the representations of different tokens. The rank collapse and the conditions to alleviate such a problem are heavily studied in [34] for an NMT task. It was shown that stacking self-attention layers during initialization can lead to the rank collapse of token representation. The intensity of this phenomenon makes it even more difficult for the model to learn meaningful attention weights. Additionally, a technique called DeepNorm that modifies both initialization and the residual connection was put forth in [35].

The main contributions of this work are two-fold. We first introduce a speaker embedding extractor for speaker verification, which is designed based on inspiration from SANs, operating in an end-to-end manner. We present architectural advances to improve the performance compared to our previous SAN-based approach. It is demonstrated that self-attention mechanisms effectively capture time-invariant features from speech, leading to a discriminative representation of the speaker. The proposed end-to-end architecture consists of a pre feed-forward feature extractor, an encoder, a pooling layer, and a Deep Neural Network (DNN) classifier. The encoder and pooling layer exclusively utilize self-attention and feed-forward networks to generate discriminative speaker embeddings. The difference between the newly proposed architecture and our previous work in [26] can be summarized into (1) the addition of the pre feed-forward feature extractor, (2) the placement of the layer normalization, and (3) the replacement of ReLU activations by Gaussian Error Linear Unit (GELU) [36] in the position-wise feed-forward network of the encoder. Unlike our prior work where the focus was on mobile devices and the lightness of the model [26], here we focus on the performance of the model considering that the model is hosted on a cloud or more powerful devices compared to mobiles. In order to show the efficacy of the proposed approach, the speaker verification system is evaluated on *Voxceleb1-O*, *VoxCeleb1-E*, and *VoxCeleb1-H* protocols [37].

Second, we empirically show the training instability in the framework of rank collapse and further study DeepNorm strategy as a potential mechanism to alleviate this shortcoming. Rank collapse is introduced as a source of the problem that results in performance degradation in SAN-based architectures. This is of utmost importance, especially while trying to take advantage of large language models pre-trained on huge amounts of speech data for other speech-related tasks and then fine-tuned for speaker verification.

The rest of the paper is organized as follows. Section 2 introduces the Self-Attention Networks and their formulation in this paper. The system architecture is presented in Section 3, where you can find all the building blocks of the proposed network. There is an introduction to rank collapse and its analysis in SAN architectures in Section 4. Section 5 presents the experimental setup such as the dataset and training hyper-parameters for all the experiments conducted in this paper. The results and discussion are explained in Section 6. Finally, the conclusions are provided in Section 7.

2. Self Attention Networks

Self-Attention Networks are built by stacking one or more blocks of SAN. Each block of SAN usually comprises two sub-layers. A Multi-Head Self-Attention mechanism makes up the first sub-layer, while a position-wise feed-forward network makes up the second [11]. Around each of the two sub-layers, a residual connection is used, and it is either followed (Post-LN) or preceded (Pre-LN) by a layer normalization [38]. The attention is applied to every position of the input sequence to produce representations. This is accomplished by employing a series of linear projections. Consider an input sequence $X = [x_1, x_2, \ldots, x_L]^T$ of length L, with $x_l \in \mathbb{R}^{d_m}$, and a set of parameters $\{W_Q, W_K\} \in \mathbb{R}^{d_m \times d_k}$, $W_V \in \mathbb{R}^{d_m \times d_v}$, the model transforms the input into namely queries $Q \in \mathbb{R}^{L \times d_k}$, keys $K \in \mathbb{R}^{L \times d_k}$, and values $V \in \mathbb{R}^{L \times d_v}$:

$$Q = XW_Q,\ K = XW_K,\ V = XW_V \tag{1}$$

The calculation for the output of each time instance, represented as o_l in the attention layer, is as follows:

$$o_l = (Attn(q_l, K)V)W_O \tag{2}$$

where $Attn(\cdot)$ is a function that for a given query q_l and the keys K returns the attention value. Additive attention [10] and dot-product attention [39] are the two most often employed attention functions. The attention mechanism used here is a scaled-down variation of the dot-product attention mechanism, which was first put forth in [11]. In comparison to additive attention, it is in practice significantly faster and more space-

efficient. The additive attention performs better than the dot-product attention when the dimension d_k increases [39]. In order to bridge the performance difference between the two while maintaining its benefits, the scaling factor is useful. The formulation of the attention mechanism's ultimate output representation O is as follows:

$$O = \left(Softmax(\frac{QK^T}{\sqrt{d_k}})V \right) W_O \tag{3}$$

where $W_O \in \mathbb{R}^{d_v \times d_m}$ is a linear layer that projects the final output into $O \in \mathbb{R}^{L \times d_m}$. The output of the attention mechanism is passed on to the position-wise feed-forward network. It is applied to each position separately and identically. This includes two linear transformations with a GELU [36] activation in between:

$$FFN(u) = gelu(uW_1 + b_1)W_2 + b_2 \tag{4}$$

where u and $FFN(u)$ are the input and output of the feed-forward network, respectively. Although these linear transformations vary from layer to layer, they are the same across different positions. $W_1 \in \mathbb{R}^{d_m \times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff} \times d_m}$ can be considered as two convolutions with kernel size of 1, while d_{ff} is the feed-forward dimension. The GELU non-linearity is defined as the expected transformation of the stochastic regularizer on an input. This statement, roughly speaking, says that we scale x according to how much greater it is than other inputs. Since the error function is frequently used to calculate the cumulative distribution function $\Phi(x)$ of a Gaussian, we define the GELU as:

$$gelu(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2}[1 + erf(\frac{x}{\sqrt{2}})] \tag{5}$$

We can approximate the GELUs with:

$$gelu(x) \simeq 0.5x(1 + tanh(\sqrt{2/\pi}(x + 0.044715x^3))) \tag{6}$$

The placement of the layer normalization is important for both performance and training stability in Transformer based architectures. The original Transformer used Post-LN while people in [40] found Pre-LN more useful for their task. Both placements were studied in [41] and was shown that the Pre-LN residual connections improve the stability of Transformers. However, as shown in [42] Pre-LN performs worse than Post-LN because its gradients in the bottom layers frequently tend to be larger than those at the top layers. In order to address the aforementioned problem, the optimization of deep Transformers has been improved through better initialization [43–45], or better architecture [42,46–48]. In this work we considered the Pre-LN to more stabilize the training process.

3. System Architecture

As depicted in Figure 1 the system described in this study can be broken down into four main components, namely a pre feed-forward feature extractor, an encoder, a pooling layer, and a DNN classifier. The very first layer is a linear feed-forward neural network which is applied on the frequency dimension of the input utterance. The encoder is a stack of N identical SAN blocks with a single-head self-attention mechanism as proposed in [32]. The encoder network produces a sequence of feature vectors as its output. These encoded features are subsequently transformed into a comprehensive speaker representation through the pooling mechanism. The fourth stage of the network involves a DNN classifier, which generates speaker posteriors. The output from the pooling layer serves as speaker embeddings, which can be utilized for tasks like speaker verification using methods such as cosine scoring or more advanced back-end techniques.

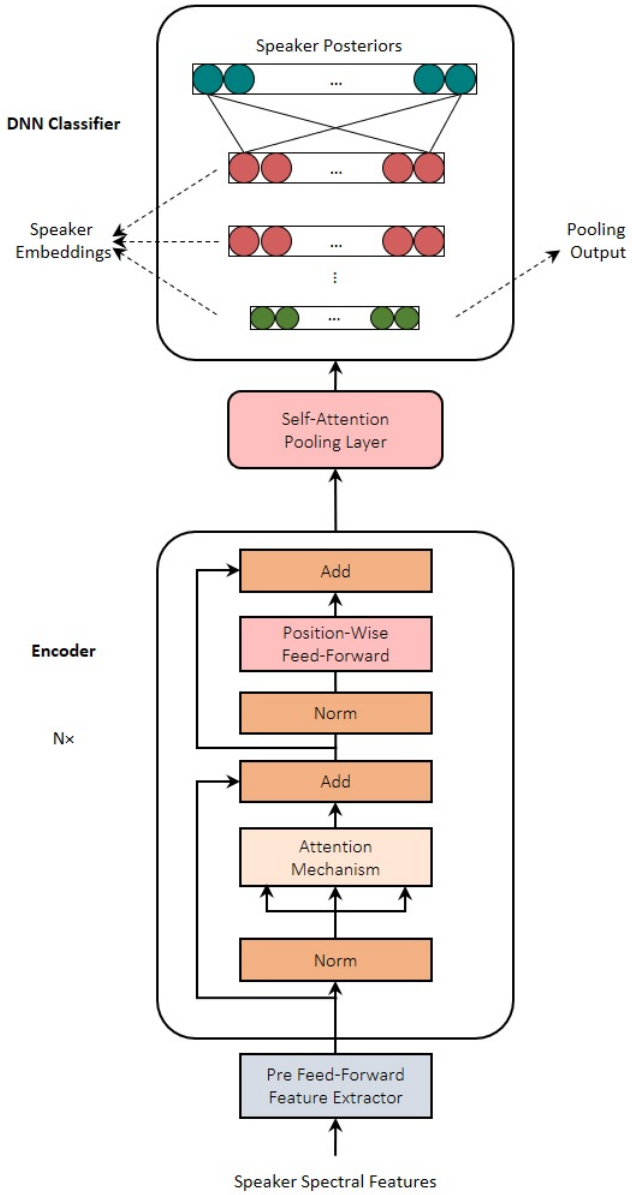

Figure 1. The architecture diagram of the proposed system. It shows that the encoder consists of N layers, where each layer is composed of a sequence of attention mechanisms followed by position-wise feed-forward networks. Residual connections and layer normalization are employed for each sub-layer.

The encoder based on SAN generates a sequence of speaker representations at the short-term frame level. To transform these frame-level features into an utterance-level vector, a self-attention pooling layer is utilized, following the approach outlined in [26]. This pooling layer employs an additive attention mechanism that calculates the compatibility function using a feed-forward network featuring a single hidden layer. Given the sequence of encoded features $H = [h_1, h_2, \ldots, h_L]^T \in \mathbb{R}^{L \times d_m}$, we compute the segment-level representation C as:

$$C = Softmax(W_c H^T)H \tag{7}$$

where $W_c \in \mathbb{R}^{d_m}$ is a trainable parameter. This self-attention mechanism can be interpreted as a dot product attention, where the keys and values represent the same representation, while the query is a trainable parameter. Consequently, C is calculated as a weighted average of the encoder sequence of features. These weights reflect the alignment learned by the self-attention mechanism. It is important to note that unlike [22,49], here the attention is not

used to extract the statistics necessary for the pooling. Rather, the entire statistical pooling process is substituted with a straightforward additive attention mechanism, resulting in reduced computational overhead.

The pooling layer produces a fixed-length vector, which is then passed into a DNN classifier to generate speaker posteriors. However, unlike [26] we only consider a softmax layer as our classifier to calculate a multi-class cross-entropy objective during training. Once the network is trained, there can be one or multiple points where we can extract the speaker embeddings. For example in [26], embeddings are derived from one of the DNN layers after the application of a non-linearity function. Instead of utilizing the previous layer prior to the softmax operation as the speaker embedding, the authors opted for the second previous layer, as done in [50]. This choice is motivated by the findings of [4], which suggest that the second previous layer exhibits improved generalization and contains speaker-discriminative information that is less specific to the training data. However, in our recent experiments, we directly extract the embeddings after applying the pooling. There are other studies such as [51], where people considered more than one layer for the embeddings and they combine multiple speaker embeddings to shape a single embedding for a speaker.

4. Rank Collapse in Self-Attention Networks

It is widely recognized that the expressive power of neural networks is enhanced by increasing the depth of the network [52]. However, increasing the depth in gradient-based methods it is usually led to a slowdown of the learning process which is commonly a consequence of unstable gradient norms in deep networks [53]. The initialization of the layer weights is one of the key players in training neural networks. The common practice in random initialization of layer weights involves selecting i.i.d entries from a zero-mean Gaussian or uniform distribution. It is sometimes accompanied by a scaling factor to maintain a constant variance of the layer activation across different layers [54,55]. Saxe et al. in [56] observed that such initialization for linear neural networks causes the majority of singular values in the last layer's activation to collapse toward zero with increasing depth. In other words, this initialization approach cannot avoid spectral instabilities in deep networks [57]. Additional methods such as residual connections and normalization layers are required to enhance the stabilization of the training process of deep architectures with such standard i.i.d initialization schemes as for example appeared in [55,57–59].

It was demonstrated in [33] that when there are no residual connections in the Transformer model, the rank of the sequence representation decreases rapidly by increasing depth. It was also shown that layer normalization and fully connected layers can to some extent mitigate the rate of degeneracy. This shows the importance of the initialization step in Transformer-based architectures. When there is a rank collapse, the model fails to differentiate between representations of distinct tokens, as they become perfectly aligned in the feature space during initialization.

Noci et al. in [34] showed for an NMT task that stacking self-attention layers can bring about the rank collapse in tokens' representation at the start of the training process by causing the known vanishing (small) gradients phenomenon in the keys and queries' parameter matrices. This issue significantly hampers the model's ability to acquire meaningful attention weights, and its impact is amplified with increasing depth. As the rank deficiency and consequent vanishing gradient problem of the queries and keys manifest across multiple layers, the problem is further exacerbated. The conditions under which the rank collapse can be avoided were also investigated in [34] at initialization. Their findings revealed that scaling the residual branches in a manner dependent on the depth, not only does it stabilize the activation norms during initialization, but it also approximately preserves the cosine of the angle between tokens. Consequently, this stabilization also helps maintain the rank of the propagating sequence. They also highlighted a different dependence on the input dimension d_m and the length of the input sequence, which suggests that the gradient

norm of a subset of parameters has the potential to differ significantly in magnitude. This observation aligns with previous empirical evidence from related studies [47].

In another attempt, Wang et al. proposed a method in [35] called DeepNorm which modifies both initialization and residual connection. It combines the residual connections in each attention and feed-forward layer for training a Transformer. In order to avoid large parameter updates, it scaled up each residual connection's computation by a constant α. Mathematically, residual connections usually output $x + f(x)$, where $f(x)$ is the function computed by the previous layer. However, in DeepNorm the output of the residual connection is computed using $\alpha x + f(x)$. It also scaled down the initial parameter values to avoid large updates in the early stages of training. Using this approach they were able to train Transformer models with 1000 layers.

Before going further we first give the formal definition of the rank collapse in matrices. Assuming our weight matrix $W \in \mathbb{R}^{N \times M}$, the Singular Value Decomposition (SVD) can be formulated as:

$$W = U\Sigma V^T \tag{8}$$

with $\nu_i = \Sigma_{ii}$ for $i = 1, \cdots, M$ as singular values. The hard rank of this matrix is simply defined as the non-zero singular values, $\nu_i > 0$. If all the singular values are non-zero, we say W is full rank. If one or more $\nu_i \sim 0$, then we say W is singular. It loses expressiveness, and the model is undergone rank collapse.

5. Experimental Setup

The VoxCeleb datasets [37,60,61] are used in all of our experiments. They consist of speech segments extracted from a wide range of YouTube videos, encompassing diverse scenarios such as clean studio interviews, red carpet interviews, outdoor environments with background noise, and situations involving multiple speakers. The videos comprise a mixture of professionally recorded content as well as footage captured using handheld or unrefined recording devices, without any editing. The VoxCeleb datasets were obtained through an automated pipeline that relied on computer vision techniques. The noise, reverberation, compression, and other artifacts in the corpus have not been removed from the original recorded speech. Since speech signals are extracted from interviews, the segments are usually short and the speech is conversational. The segments come in different duration and may include background speech from other identities. There may be several speech signals per speaker available in the corpus.

For all the experiments the systems are evaluated on the fixed-condition VoxSRC 2019 training restrictions [62]. The fixed training condition mandates training exclusively on the VoxCeleb2 dev dataset [37], which includes 1,092,009 utterances from a total of 5994 speakers. During the test time for the VoxCeleb1 protocol, it is evaluated over the VoxCeleb1 test pairs. The VoxCeleb1-E test protocol consists of a collection of 581,480 random pairs extracted from the complete VoxCeleb1 dataset, which includes both the development and test subsets. These pairs encompass 1251 different speakers. The VoxCeleb1-H test protocol involves a set of 552,536 pairs sampled exclusively from the VoxCeleb1 dataset. All pairs within this set share the same nationality and gender. The list comprises a total of 18 combinations of nationality and gender, with each combination containing a minimum of 5 individuals.

A small subset of about 10% of the data is reserved as a validation set for the best model selection. For all the trained systems in our study, SpeechBrain [63] is employed for the whole training/test pipeline. It is used to extract the 128-dimensional (unless otherwise stated) Mel-Frequency Cepstral Coefficients (MFCCs) with the standard 25 ms window size and 10 ms shift to represent the speech signal. To form the input for the networks, the deltas and double deltas of the MFCCs are combined, resulting in a 384-dimensional representation. Before training, all features undergo sentence-level Cepstral mean normalization. The models are then trained using speech feature chunks consisting of 300 frames. During testing, embeddings are extracted from the entire speech signal.

For the models trained with data augmentation, a total of 5 extra samples are generated for each utterance. The first two augmentations are provided using speed perturbation by slightly speeding up or slowing down an audio signal. This is accomplished via resampling the audio signal at a rate that is similar to the original rate, to achieve a slightly slower or slightly faster signal [64]. The remaining three augmentations are generated by adding a mix of noise and reverberation using the RIR dataset provided in [65].

During the training process, all models are trained using the cyclical learning rate technique, which varies between 1×10^{-8} and 1×10^{-3}. The learning rate follows a triangular policy as described in [66]. The Adam optimizer [67] is utilized. Each cycle consists of 130,000 iterations. Additive Angular Margin softmax (AAM-softmax) [68,69] is employed with a margin of 0.2 and softmax prescaling of 30. To mitigate overfitting, a weight decay of 2×10^{-6} is applied to all model weights. The mini-batch size for training is 64. All models are trained for 150 epochs and the best model is chosen according to the error rate of the validation data. To compute singular values we take advantage of the linear algebra routine of the NumPy package [70].

For all experiments including SAN architectures proposed in this work, we use a stack of $N = 2$ identical layers with $d_k = d_v = 768$ (double the feature input size), and position-wise feedforward dimension of $d_{ff} = 3072$ (eight times the feature input size). In the encoder network, a dropout rate of 0.1 is applied, while the remaining parts of the network utilize a fixed dropout rate of 0.2. The speaker embedding dimension is considered to be equal to the key-value pair which is 768. There have been no conducted experiments to investigate the optimal size for the embedding. The pre feed-forward feature extractor is a single layer of size 768.

6. Results and Discussion

Although some state-of-the-art methods for speaker verification are based on convolution operation [5,6], attention-based architectures are clearly state-of-the-art in speech recognition, Natural Language Processing (NLP), and many other deep-learning tasks [11,13–17]. In particular, this approach is very convenient in our domain because of multi-purpose large language models, which are trained for a mother task such as speech recognition and can be fine-tuned for a specific downstream task such as speaker verification. For this reason, we focus our experiments and discussion on SANs to improve the performance of such models in speaker recognition.

The results for the text-independent speaker verification task for three different protocols are reported in Table 1. In all these models comprise two blocks of SAN, $N = 2$, which are stacked on top of each other. The scoring of different systems is based on the cosine distance. The performance of these systems has been assessed using the Equal Error Rate (EER) metric. The newly proposed SAN-based architecture, A-SAN (Advanced SAN), is compared with the one originally proposed in [26]. There is more than 41% and 38% relative improvement in VoxCeleb1-O, and VoxCeleb1-E protocols, respectively, without any Data Augmentation (DA). There is an additional 25% and 29% performance improvement in both VoxCeleb1-O and VoxCeleb1-E protocols, respectively, by employing DA techniques. For the VoxCeleb1-H which is the hardest protocol for the verification, we achieve 27% relative improvement by adding DA to the training pipeline.

Table 1. Comparison of the evaluation results reported on VoxCeleb datasets with and without Data Augmentation (DA). A-SAN is referred to the SAN model proposed in this work. The results reported for SAN model from [26] are without any DA. Cosine distance is employed for scoring.

Architecture	VoxCeleb1-O EER (%)	VoxCeleb1-E EER (%)	VoxCeleb1-H EER (%)
SAN [26]	5.44	5.62	-
A-SAN w/o DA	3.20	3.43	5.54
A-SAN w/ DA	2.41	2.42	4.02

Figure 2, compares the singular values of the classifier layer in descending order for the three different systems from Table 1 for the VoxCeleb1-E protocol. For a matter of clarity, only the first 200 singular values are depicted in the figure. The singular values for each system are normalized with respect to their corresponding maximum value in order to be in the same dynamic range (between 0 and 1). As is shown all systems suffer from rank collapse. This rank collapse in the case of SAN in [26] is very severe and it becomes smoother with the modifications proposed in this work. In other words, the proposed changes directly affect the rank collapse, which also impacts the EER performance of the embedding extractor system.

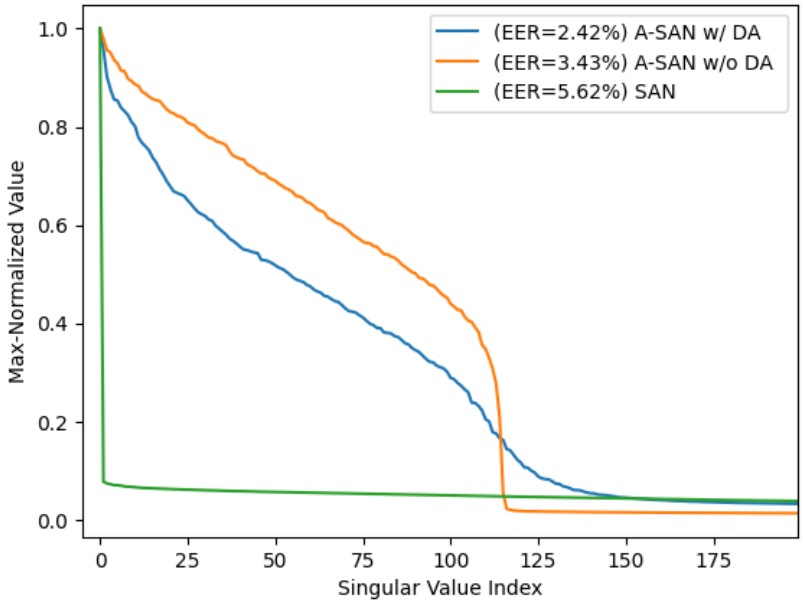

Figure 2. Comparison of the singular values of the classifier layer for different models with and without Data Augmentation (DA). The evaluation is performed over VoxCeleb1-E protocol.

Let us now focus on the experiments without any DA to only show the efficiency of the architectural and training modifications. The performance of SAN architectures in terms of different numbers of SAN layers is illustrated in Figure 3. As it is observed, the performance of the original A-SAN architecture is decreased by increasing the depth. In order to stabilize the performance, a DeepNorm [35] training mechanism is applied. As it is shown, the performance does not change substantially by stacking more layers on top of each other. In the following, we discuss this stabilizing effect in the framework of rank collapse.

Figure 4 compares the rank collapse of the pooling layer for the A-SAN systems with different numbers of layers, with and without the DeepNorm method. Note that, the pooling layer is the uppermost layer of the embedding extractor during the inference phase. From the figure, it appears that the rank collapse follows a similar trend for all numbers of layers when DeepNorm is used, while the rank collapse depends on the number of layers without DeepNorm. In general, these figures suggest that the use of DeepNorm can stabilize the training process and reduce rank collapse in A-SAN systems, regardless of the number of layers. This may lead to improved performance of the model while adding more layers.

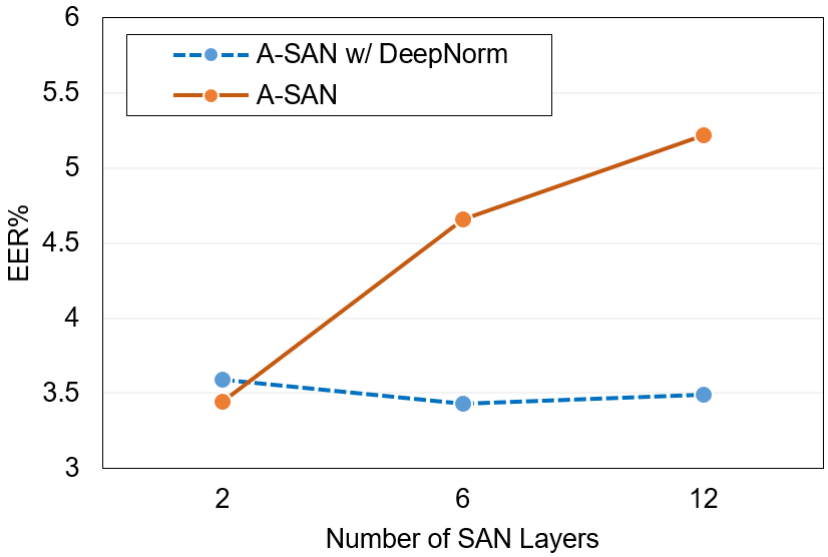

Figure 3. Performance evaluation for A-SAN architectures with different numbers of SAN layers with and without applying DeepNorm method. All experiments are conducted without any data augmentation over VoxCeleb1-E protocol.

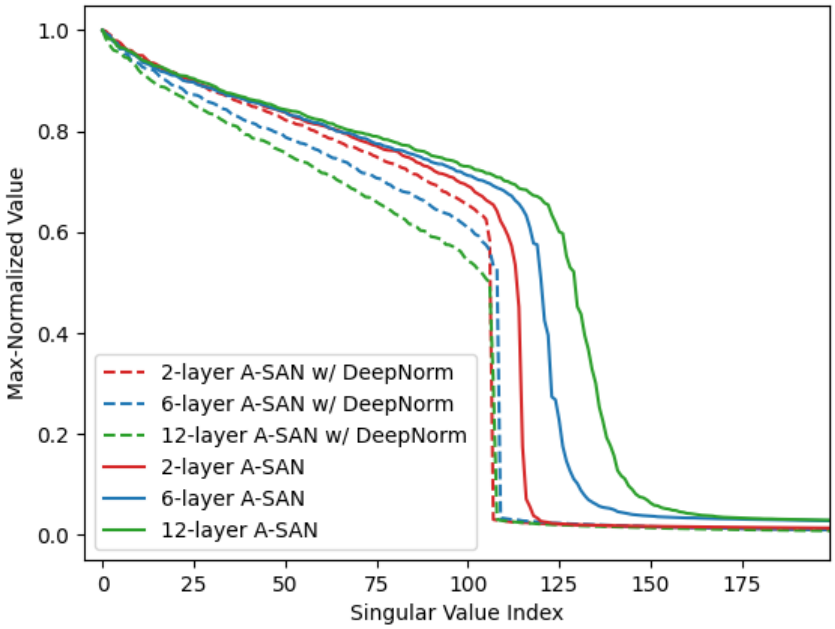

Figure 4. Comparison of the singular values of the pooling layer for different numbers of SAN layers. All experiments are conducted without any data augmentation over VoxCeleb1-E protocol.

Figure 5 illustrates the rank collapse for the self-attention output matrix W_O in the first layer. It is observed that for all the systems rank collapse is more severe than uppermost layers. However, this deterioration of the rank is much more dramatic for systems without DeepNorm strategy. On the other hand for these systems, the rank degrades faster by increasing the number of layers. This can justify the performance degradation in these systems (Figure 3) by stacking more layers of SANs. This can support the idea that rank collapses much faster in the lower layers than in the upper ones.

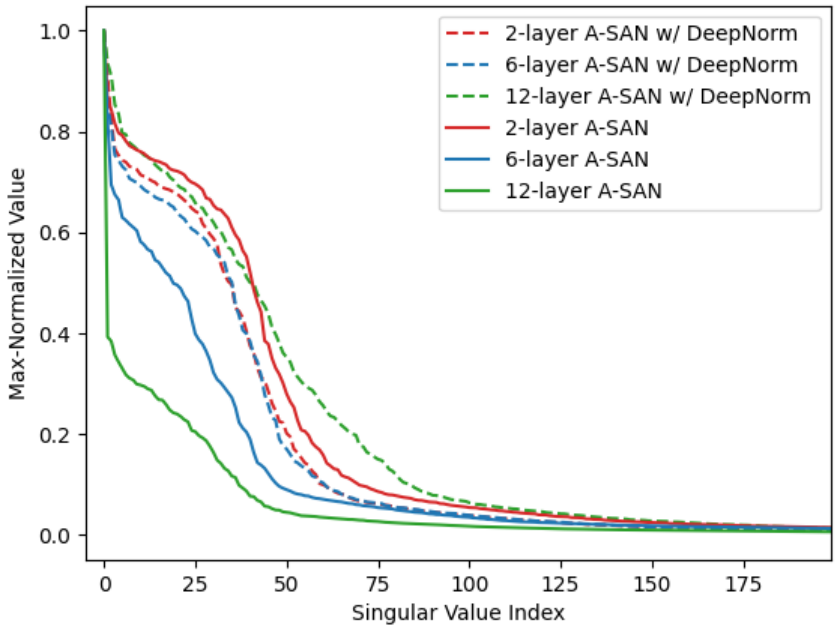

Figure 5. Comparison of the singular values of the first self-attention output layer W_O (first layer) for the different number of SAN layers. All experiments are conducted without any data augmentation over VoxCeleb1-E protocol.

7. Conclusions

In this paper, we introduced a SAN-inspired end-to-end speaker embedding extractor for text-independent speaker verification. A pre feed-forward feature extractor, an encoder, a pooling layer, and a DNN classifier were all included in the end-to-end suggested design. All these building blocks rely on self-attention and feed-forward networks. We evaluated its performance by applying it to three different VoxCeleb protocols, namely VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. Our proposed modifications resulted in 41% and 38% relative improvement for VoxCeleb1-O and VoxCeleb1-E, respectively, compared to naive SAN architecture. Moreover, it was shown that employing data augmentation during training can lead to additional improvements in performance. Having used data augmentation, we achieved 25%, 29%, and 27% relative improvement for VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H protocols, respectively. In addition, we studied the training instability in terms of the rank collapse phenomenon. All SAN models trained for this task suffered from rank collapse in different layers and it was empirically shown that there is a correlation between rank collapse and the performance of the verification system in terms of EER. Finally, we have shown that it is possible to improve the training stability while adding more layers of SAN by employing DeepNorm method. Although DeepNorm strategy stabilized the training, it could not improve the performance of the network by adding more layers of SAN. Potential future research directions can be to further study more advanced attention techniques such as Double MHA [25] to improve the performance of the network while stacking more SAN layers in the encoder.

Author Contributions: P.S., M.I., and J.H. conceptualized the study, developed the model, designed the experiments, and wrote/reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish Project ADAVOICE PID2019-107579RB-I00 (MICINN).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

DL	Deep Learning
SR	Speaker Recognition
NMT	Neural Machine Translation
ASR	Automatic Speech Recognition
RNN	Recurrent Neural Network
SAN	Self-Attention Network
MHA	Multi-Head Attention
DNN	Deep Neural Network
GELU	Gaussian Error Linear Unit
SVD	Singular Value Decomposition
MFCC	Mel-Frequency Cepstral Coefficient
EER	Equal Error Rate
AAM-softmax	Additive Angular Margin Softmax
LN	Layer Normalization

References

1. Safari, P.; Ghahabi, O.; Hernando, J. From features to speaker vectors by means of restricted Boltzmann machine adaptation. In Proceedings of the ODYSSEY, Bilbao, Spain, 21–24 June 2016; pp. 366–371.
2. Ghahabi, O.; Hernando, J. Restricted Boltzmann machines for vector representation of speech in speaker recognition. *Comput. Speech Lang.* **2018**, *47*, 16–29. [CrossRef]
3. 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.
4. Snyder, D.; Garcia-Romero, D.; Povey, D.; Khudanpur, S. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 999–1003.
5. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In Proceedings of the ICASSP, Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
6. Desplanques, B.; Thienpondt, J.; Demuynck, K. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 3830–3834.
7. 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]
8. Chen, S.; Wu, Y.; Wang, C.; Chen, Z.; Chen, Z.; Liu, S.; Wu, J.; Qian, Y.; Wei, F.; Li, J.; et al. Unispeech-sat: Universal speech representation learning with speaker aware pre-training. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6152–6156.
9. 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]
10. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
12. Chan, W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964.
13. Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; McCallum, A. Linguistically-informed self-attention for semantic role labeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 5027–5038.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
15. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.

16. Chiu, C.C.; Sainath, T.N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R.J.; Rao, K.; Gonina, E.; et al. State-of-the-art speech recognition with sequence-to-sequence models. In Proceedings of the ICASSP, Calgary, AB, Canada, 15–20 April 2018; pp. 4774–4778.
17. Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; Kaiser, Ł. Universal transformers. *arXiv* **2018**, arXiv:1807.03819.
18. Bhattacharya, G.; Alam, M.J.; Kenny, P. Deep Speaker Embeddings for Short-Duration Speaker Verification. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 1517–1521.
19. Zhang, S.X.; Chen, Z.; Zhao, Y.; Li, J.; Gong, Y. End-to-end attention based text-dependent speaker verification. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, USA, 13–16 December 2016; pp. 171–178.
20. Chowdhury, F.; Wang, Q.; Moreno, I.L.; Wan, L. Attention-based models for text-dependent speaker verification. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5359–5363.
21. Cai, W.; Chen, J.; Li, M. Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System. In Proceedings of the Odyssey, Les Sables d'Olonne, France, 26–29 June 2018; pp. 74–81.
22. Zhu, Y.; Ko, T.; Snyder, D.; Mak, B.; Povey, D. Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification. In Proceedings of the Interspeech 2018, Hyderabad, India, 2–6 September 2018.
23. Lin, Z.; Feng, M.; Santos, C.N.d.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
24. India, M.; Safari, P.; Hernando, J. Self Multi-Head Attention for Speaker Recognition. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019.
25. India, M.; Safari, P.; Hernando, J. Double multi-head attention for speaker verification. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6144–6148.
26. Safari, P.; India, M.; Hernando, J. Self-attention encoding and pooling for speaker recognition. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 941–945.
27. Han, B.; Chen, Z.; Qian, Y. Local Information Modeling with Self-Attention for Speaker Verification. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6727–6731.
28. Gu, S.; Feng, Y. Improving multi-head attention with capsule networks. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Dunhuang, China, 9–14 October 2019; pp. 314–326.
29. Cordonnier, J.B.; Loukas, A.; Jaggi, M. Multi-head attention: Collaborate instead of concatenate. *arXiv* **2020**, arXiv:2006.16362.
30. Michel, P.; Levy, O.; Neubig, G. Are sixteen heads really better than one? *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
31. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5797–5808.
32. Liu, L.; Liu, J.; Han, J. Multi-head or single-head? An empirical comparison for transformer training. *arXiv* **2021**, arXiv:2106.09650.
33. Dong, Y.; Cordonnier, J.B.; Loukas, A. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 2793–2803.
34. Noci, L.; Anagnostidis, S.; Biggio, L.; Orvieto, A.; Singh, S.P.; Lucchi, A. Signal Propagation in Transformers: Theoretical Perspectives and the Role of Rank Collapse. *arXiv* **2022**, arXiv:2206.03126.
35. Wang, H.; Ma, S.; Dong, L.; Huang, S.; Zhang, D.; Wei, F. Deepnet: Scaling transformers to 1,000 layers. *arXiv* **2022**, arXiv:2203.00555.
36. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
37. Chung, J.S.; Nagrani, A.; Zisserman, A. VoxCeleb2: Deep Speaker Recognition. In Proceedings of the INTERSPEECH 2018, Hyderabad, India, 2–6 September 2018.
38. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
39. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1412–1421.
40. Chen, M.X.; Firat, O.; Bapna, A.; Johnson, M.; Macherey, W.; Foster, G.; Jones, L.; Parmar, N.; Schuster, M.; Chen, Z.; et al. The best of both worlds: Combining recent advances in neural machine translation. *arXiv* **2018**, arXiv:1804.09849.
41. Nguyen, T.Q.; Salazar, J. Transformers without tears: Improving the normalization of self-attention. In Proceedings of the 16th International Conference on Spoken Language Translation, Hong Kong, China, 2–3 November 2019.
42. Shleifer, S.; Weston, J.; Ott, M. Normformer: Improved transformer pretraining with extra normalization. *arXiv* **2021**, arXiv:2110.09456.
43. Zhang, B.; Titov, I.; Sennrich, R. Improving deep transformer with depth-scaled initialization and merged attention. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 898–909.
44. Zhang, H.; Dauphin, Y.N.; Ma, T. Fixup initialization: Residual learning without normalization. *arXiv* **2019**, arXiv:1901.09321.
45. Huang, X.S.; Perez, F.; Ba, J.; Volkovs, M. Improving transformer optimization through better initialization. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 4475–4483.

46. Wang, Q.; Li, B.; Xiao, T.; Zhu, J.; Li, C.; Wong, D.F.; Chao, L.S. Learning deep transformer models for machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1810–1822.
47. Liu, L.; Liu, X.; Gao, J.; Chen, W.; Han, J. Understanding the difficulty of training transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5747–5763.
48. Bachlechner, T.; Majumder, B.P.; Mao, H.; Cottrell, G.; McAuley, J. Rezero is all you need: Fast convergence at large depth. In Proceedings of the Uncertainty in Artificial Intelligence, PMLR, Online, 27–30 July 2021; pp. 1352–1361.
49. Okabe, K.; Koshinaka, T.; Shinoda, K. Attentive statistics pooling for deep speaker embedding. In Proceedings of the Interspeech 2018, Hyderabad, India, 2–6 September 2018; pp. 2252–2256.
50. Zeinali, H.; Burget, L.; Rohdin, J.; Stafylakis, T.; Cernocky, J.H. How to improve your speaker embeddings extractor in generic toolkits. In Proceedings of the ICASSP 2019, Brighton, UK, 12–17 May 2019; pp. 6141–6145.
51. Monteiro, J.; Alam, J.; Falk, T.H. Multi-level self-attentive TDNN: A general and efficient approach to summarize speech into discriminative utterance-level representations. *Speech Commun.* **2022**, *140*, 42–49. [CrossRef]
52. Telgarsky, M. Benefits of depth in neural networks. In Proceedings of the Conference on Learning Theory, PMLR, New York, NY, USA, 23–26 June 2016; pp. 1517–1539.
53. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness-Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
54. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
56. Saxe, A.M.; McClelland, J.L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv* **2013**, arXiv:1312.6120.
57. Daneshmand, H.; Kohler, J.; Bach, F.; Hofmann, T.; Lucchi, A. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18387–18398.
58. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
59. Salimans, T.; Kingma, D.P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
60. Nagrani, A.; Chung, J.S.; Zisserman, A. VoxCeleb: A large-scale speaker identification dataset. In Proceedings of the INTERSPEECH, Stockholm, Sweden, 20–24 August 2017.
61. Nagrani, A.; Chung, J.S.; Xie, W.; Zisserman, A. Voxceleb: Large-scale speaker verification in the wild. *Comput. Speech Lang.* **2020**, *60*, 101027. [CrossRef]
62. Chung, J.S.; Nagrani, A.; Coto, E.; Xie, W.; McLaren, M.; Reynolds, D.A.; Zisserman, A. VoxSRC 2019: The first VoxCeleb speaker recognition challenge. *arXiv* **2019**, arXiv:1912.02522.
63. Ravanelli, M.; Parcollet, T.; Plantinga, P.; Rouhe, A.; Cornell, S.; Lugosch, L.; Subakan, C.; Dawalatabad, N.; Heba, A.; Zhong, J.; et al. SpeechBrain: A General-Purpose Speech Toolkit. *arXiv* **2021**, arXiv:2106.04624.
64. Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
65. 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.
66. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.
67. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
68. 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.
69. Xiang, X.; Wang, S.; Huang, H.; Qian, Y.; Yu, K. Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; pp. 1652–1656.
70. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]