Deep Speaker Vector Normalization with Maximum Gaussianality Training

Yunqi Cai, Lantian Li, Dong Wang and Andrew Abel

***Abstract*—Deep speaker embedding represents the state-of-the-art technique for speaker recognition. A key problem with this approach is that the resulting deep speaker vectors tend to be irregularly distributed. In previous research, we proposed a deep normalization approach based on a new discriminative normalization flow (DNF) model, by which the distributions of individual speakers are arguably transformed to homogeneous Gaussians. This normalization was demonstrated to be effective, but despite this remarkable success, we empirically found that the latent codes produced by the DNF model are generally neither homogeneous nor Gaussian, although the model has assumed so. In this paper, we argue that this problem is largely attributed to the maximum-likelihood (ML) training criterion of the DNF model, which aims to maximize the likelihood of the observations but not necessarily improve the Gaussianality of the latent codes. We therefore propose a new Maximum Gaussianality (MG) training approach that directly maximizes the Gaussianality of the latent codes. Our experiments on two data sets, SITW and CNCeleb, demonstrate that our new MG training approach can deliver much better performance than the previous ML training, and exhibits improved domain generalizability, particularly with regard to cosine scoring.**

***Keywords*—Speaker Recognition; Speaker Embedding; Normalization Flow**

I. Introduction

Speaker recognition, more precisely, speaker verification, is the task of verifying the identity of a person by their voice [1], [2], [3]. Early speaker recognition methods are based on statistical models such as the Gaussian mixture model-universal background model (GMM-UBM) [4] and the i-vector model [5]. Recently, most powerful systems are based on deep learning methods [6], [7], [8], [9], [10], [11], which exploit the power of deep neural nets (DNN) in high-level feature learning, and often deliver more robust performance compared to conventional statistical methods.

Current deep learning methods can be categorized into two approaches: the *end-to-end* approach [9], [10], [11] and the *speaker embedding* approach [6], [7], [8]. While the end-to-end approach learns a neural net that decides whether two utterances are from the same speaker, the speaker embedding approach learns a neural net that generates vector-based representation (usually called *speaker vectors*) for each utterance, and then employ a scoring model to make the decision. Both approaches have their own advantages, but due to the simplicity of model training and state-of-the-art performance [8], [12] of the speaker embedding approach, in this paper, we focus on this method. Recently, numerous studies have been conducted to improve the deep embedding approach [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], and amongst the proposed models, the x-vector model [8] has become one of the most popular with researchers.

A. Deep normalization for speaker vectors

Compared to the end-to-end approach, a key advantage of the speaker embedding approach is that it employs a statistical model to infer the scores for test trials. By this statistical inference, the uncertainty during the decision process (e.g., uncertainty caused by limited enrollment data) is addressed in a method based on solid principles. Among all the scoring models, the probabilistic linear discriminant analysis (PLDA) [28] model has been shown to be very effective [8], [29]. This model assumes that the data are linear Gaussian, i.e., the between-class distribution and the within-class distributions of individual classes are all Gaussians, and the within-class distributions of different speakers are homogeneous (with different means but the same covariance matrix). It can be shown that if the speaker vectors are truly linear Gaussian, then optimal decisions can be made based on the PLDA score, where the optimum represents minimum Bayes risk (MBR) [30].

However, speaker vectors produced by the embedding model are not necessarily linear Gaussian, as the DNN-based embedding model does not consider the distribution of the produced speaker vectors. This means that speaker vectors can be distributed in any form, and in general are not linear Gaussian, as shown in Fig. 1. This irregular distribution will seriously degrade performance of the PLDA scoring.

Recently, we presented a deep normalization approach and achieved very promising results [31], based on a novel discriminative normalization flow (DNF) model, which transforms speaker vectors to latent codes, where within-class distributions of individual speakers are supposed to be homogeneous Gaussians. Our experiments showed that based on PLDA scoring, DNF-based normalization reduced the equal error rate (EER) by 31% on an in-domain test and 9.3% on an out-of-domain test. Fig. 1 shows the improved Gaussianality of the speaker vectors after the DNF-based normalization.

This work was supported by the National Natural Science Foundation of China (NSFC) under the project No.61633013 and No.61371136. Thanks to Dr. Zhiyuan Tang's valuable discussion. (Corresponding author: Dong Wang.)

Y. Cai is with the Center for Speech and Language Technologies (CSLT) and the department of Computer Science at Tsinghua University, Beijing 100084, China (e-mail: caiyq@cslt.org).

L. Li and D. Wang are with the Center for Speech and Language Technologies (CSLT), BNRist at Tsinghua University, Beijing 100084, China (e-mail: wangdong99@mails.tsinghua.edu.cn, lilt@cslt.org).

A. Abel is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: andrew.abel@xjtlu.edu.cn).

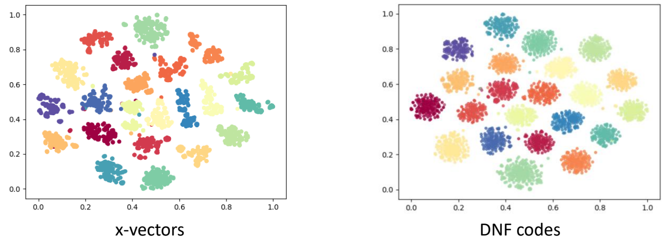


Fig. 1: Original x-vectors (left) and latent codes produced by DNF-based deep normalization (right). After the normalization, the within-class distributions of individual speakers are more Gaussian.

If the normalization is perfect, PLDA scoring will produce MBR optimal decisions, leading to the best verification performance [30]. This encourages us to focus on the normalization model, rather than more powerful scoring models (e.g., discriminative PLDA [32] or heavy-tailed PLDA [33]), or more complicated score calibration methods [34], [35].

B. *Motivation and Contribution*

Normalization based on our previous DNF model, although promising, is not perfect. Empirically, we found that the latent codes produced by DNFs are generally not homogeneous, although the model has assumed so (Fig. 10). This problem can be attributed to the maximum likelihood (ML) criterion used for DNF training, where the primary goal is to match the transformed Gaussian prior to the training samples, rather than the Gaussianality of the latent codes, the goal of our normalization model. In other words, ML training does not match the goal of speaker vector normalization.

To solve this issue, in this paper, we propose a Maximum Gaussianality (MG) training approach for the DNF model. Our new training approach firstly defines two Gaussian metrics to measure the Gaussianality of the latent codes, and then trains the DNF model to maximize these metrics. A length metric and an angle metric are defined in this work, to reflect the following two properties of a high-dimensional Gaussian: (1) the length of most samples concentrates on a small annulus; (2) any two samples tend to be orthogonal. Since the two properties are necessary conditions of a high-dimensional Gaussian, maximizing the length metric and the angle metric will directly optimize the Gaussianlity of the latent codes. For the DNF model, our proposed MG training can be applied to optimize both the between-class distribution and the within-class distributions of individual classes, as shown in Fig. 2.

The remainder of this paper is organized as follows: Section II briefly reviews our DNF model, and Section III presents our proposed MG training approach. Experimental results and analysis are presented in Section IV, and the paper is concluded in Section V. The code will be available at http://project.cslt.org.

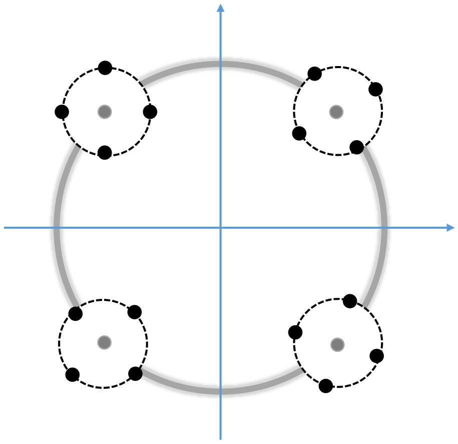


Fig. 2: Our Maximum Gaussianality (MG) training applied to the DNF model. The length metric is maximized to encourage the samples to be concentrated on a spherical surface, and the angle metric is maximized to encourage the samples to be distributed evenly on the surface. MG training can be employed to maximize the Gaussianality of the between-class distribution (denoted by the thick gray circle), as well as the within-class distribution of *each* individual speaker (denoted by the thin black circle).

II. Discriminative Normalization Flow

Here, we briefly review the DNF model proposed in previous work which will also be used in this paper. For more details, please refer to the original paper [31].

A. *Normalization flow*

We first introduce to the normalization flow (NF) model. More details can be found in the review paper by Papamakarios [36]. Suppose that a latent variable $\mathbf{z}$ and an observation variable $\mathbf{x}$ are linked by an invertible transform $\mathbf{x} = f(\mathbf{z})$. Their probability densities have the following relationship [37]:

$$\ln p(\mathbf{x}) = \ln p(\mathbf{z}) + \ln \left| \det \frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|, \tag{1}$$

where the first term on the right hand side is the prior probability of the latent codes, and the second is the entropy term representing the volume change associated with the transform f. It has been shown that if f is flexible enough, a simple distribution on $\mathbf{z}$, which we assume is a standard Gaussian, can be transformed to a complex distribution on $\mathbf{x}$ [36].

Usually, f is implemented as a composition of a sequence of simple invertible transforms and each transform can be a neural net with a special structure to ensure a tractable computation for the entropy term. This sequence of invertible transforms is called a normalization flow [38]. NF is invertible, and the inverse function f^{-1} will normalize a complex distribution on $\mathbf{x}$ to a simple distribution on $\mathbf{z}$.

The NF model is often trained with the maximum likelihood (ML) criterion. Note that Eq.1 formulates a density $p(\mathbf{x})$ on the observation $\mathbf{x}$, and so the ML training can be conducted by maximizing the following objective function:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_i \ln p(\mathbf{x}_i) = \sum_i \ln p(\mathbf{z}_i) + \sum_i \sum_{t=1}^{T+1} \ln \left| \det \frac{\partial f_{t-1}^{-1}(\mathbf{z}_{it})}{\partial \mathbf{z}_{it}} \right|,$$

where $\boldsymbol{\theta}$ represents the parameters of the NF model. Once the model has been well trained, the NF model can be employed to transform a complex distribution on $\mathbf{x}$ to a Gaussian distribution on $\mathbf{z}$.

B. Discriminative normalization flow

The NF model does not consider any class labels. Therefore, there is no class-related structure exhibited in the latent space, and so it cannot be used to normalize class-based distributions. To solve this, we previously introduced the discriminative normalization flow (DNF) model [31].

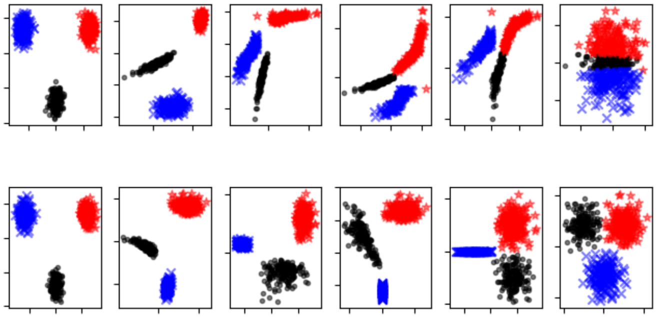


Fig. 3: Normalization process of vanilla NF (top) and DNF (bottom) with 2-dimensional data sampled from 3 Gaussian components. The DNF model treats each Gaussian as a single class. The plots from left to right show the output of each flow step, with the first and lasts plot representing the observation $\mathbf{x}$ and latent code $\mathbf{z}$ respectively. Vanilla NF pulls all classes together in the latent space and cannot regularize individual classes, while DNF (bottom) separates data from different classes and normalizes individual classes to be Gaussian.

Compared to vanilla NF, DNF allows different Gaussian priors for different classes, and these priors share the same covariance but possess different means, formulated as follows:

$$p_y(\mathbf{z}) = N(\mathbf{z}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}), \tag{2}$$

where y is the class label. By setting class-specific means, different classes will be separated from each other in the latent space, and each class is normalized to be a Gaussian. The different behaviors of NF and DNF are shown in Fig. 3.

Training DNF follows the ML criterion, similarly to vanilla NF. The key difference is that the probability of an observation $\mathbf{x}$ should be evaluated with the prior corresponding to its class label, defined by:

$$p(\mathbf{x}) = p_{y_{\mathbf{x}}}(\mathbf{z}) \left| \det \frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|,$$

where $y_{\mathbf{x}}$ is the class label of $\mathbf{x}$, and $\mathbf{z} = f^{-1}(\mathbf{x})$. Pooling all the training data, we obtain the objective function for DNF training:

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_i \log(p_{y_{\mathbf{x_i}}}(\mathbf{z_i})) + \log \left| \det \frac{\partial f^{-1}(\mathbf{x}_i)}{\partial \mathbf{x}_i} \right|, \tag{3}$$

where $\boldsymbol{\Theta} = \{\{\boldsymbol{\mu}_y\}_y, \boldsymbol{\Sigma}, \boldsymbol{\theta}\}$ involves all the parameters of the model. In practice, we set $\boldsymbol{\Sigma} = \mathbf{I}$ and let the flow handle the volume change.

After training, the DNF model will establish a latent space $\mathcal{Z}$, where the distribution $p_y(\mathbf{z})$ of every class y is a Gaussian with covariance $\mathbf{I}$. With this model, an observation $\mathbf{x}$ can be transformed to its latent code $\mathbf{z}$ by the inverse transform $f^{-1}(\mathbf{x})$ without knowing its class labels, and the latent codes from the same class (maybe unknown) tend to form a Gaussian.

It should be highlighted that this DNF model only concerns the likelihood with respect to the within-class distributions. For clarity, we call this DNF objective the within-class maximum likelihood, formulated as:

$$\mathcal{R}_{WC}^{ML} = \sum_i \log(p_{y_{\mathbf{x_i}}}(\mathbf{z_i})) + \log \left| \det \frac{\partial f^{-1}(\mathbf{x}_i)}{\partial \mathbf{x}_i} \right|. \tag{4}$$

III. Maximum Gaussianality DNF

The DNF model shows great potential [31], however, experimental evidence demonstrated that the DNF-based normalization is not perfect, as will be presented in detail. This can be largely attributed to the maximum likelihood criterion used for training. This paper presents a novel Maximum Gaussianality (MG) training approach to solve this problem.

A. Non-Gaussianality with DNF

The DNF model assumes that the within-class distributions of individual classes are homogeneous Gaussian (see. Eq. (2)), however, empirical results presented here show that the latent codes of most speakers are not Gaussian, and the distributions of different speakers are not homogeneous (ref. to Fig. 10 and Table II). One possible explanation is that if the samples of a speaker are limited, its distribution cannot be well optimized, but this does not explain why the DNF model cannot obtain the distributions that it assumes.

Secondly, the between-class distribution is generally non-Gaussian, which is unsurprising, as DNF does not set any constraints on the class means. However, when we placed a Gaussian prior on the class means, the Gaussianality of the between-class distribution was not improved significantly. Specifically, we placed the following prior on the class mean $\boldsymbol{\mu}$:

$$\mathcal{R}_{BC}^{ML} = \sum_y \left\{ p(\boldsymbol{\mu}_y) + \ln \left| \det \frac{\partial f(\boldsymbol{\mu}_y)}{\partial \boldsymbol{\mu}_y} \right|^{-1} \right\}, \tag{5}$$

where:

$$p(\boldsymbol{\mu}) = N(\boldsymbol{\mu}; \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}}).$$

Note that we added the entropy term for $\boldsymbol{\mu}_y$, as the goal is to maximize the likelihood of the class centers in the observation space (represented by $f(\boldsymbol{\mu}_y)$), rather than those in the latent space (represented by $\boldsymbol{\mu}_y$). We will show in Section IV that this regularization does offer expected performance gains.

In both the aforementioned cases, the fundamental problem is that the latent codes are not regulated to follow the distribution assumed by the model. This is a serious problem for deep normalization, as the assumed distribution is the central goal of the normalization model.

B. Maximum Gaussianality criterion

As we have discussed, latent codes deviating from the assumed distribution is a fundamental problem of the DNF model, and we argue that this is mostly caused by the ML criterion used for the model training. For simplicity, the analysis will be conducted based on the NF model; the conclusions naturally generalize to the DNF model.

In theory, the transform implemented by the NF can be unbounded complex and can transfer the Gaussian prior to any complex distribution. Unfortunately, ML training can exploit this flexibility to improve the likelihood without constraints, leading to unbounded likelihood and unbounded complex transform. This unexpected behavior may cause multiple problems that result in irregular latent codes.

One issue is related to spurious modes. Consider an NF model that transforms an observation $\mathbf{x}$ to latent code $\mathbf{z}$. Intuitively, ML training can be regarded as a process of manipulating the Gaussian prior, with the goal of increasing the likelihood of the training data to be as large as possible, when measured by the manipulated prior. This manipulation can be in any form, and may lead to spurious modes as illustrated in Fig. 4. Since spurious modes take some volume in the latent space, the latent codes of the true observations will not be distributed like a full Gaussian. Nalisnick et al. [39] provided empirical evidence for this. They found that some out-of-domain data may be assigned even higher probability than the in-domain data by the NF model. Since the model they used was a constant-volume transform, the out-of-domain data must occupy some high-density volume of the latent space, and so the latent codes of the in-domain data cannot be a full Gaussian, despite being assumed by the NF model.

The second problem is related to under-representation and overfitting. When the training data is limited, it does not represent the full distribution of the data, meaning the trained NF model does not represent the true distribution, resulting in non-Gaussian codes for the test data, as illustrated in Fig. 5.

In summary, ML training does not guarantee a Gaussian distribution for the latent codes. Although the analysis is based on the NF model, the same problem exists for the DNF model. In this paper, we present a new Maximum Gaussianality (MG) training approach to solve this. Compared to ML training, MG training maximizes the Gaussianality of the latent codes directly, therefore avoiding the problems caused by ML training.

C. Gaussian metrics

MG training requires Gaussianality to be defined. Although high-order statistics such as Kurtosis, Skewness and Negentropy [40] have been widely used to measure Gaussianality, these metrics require a relatively large amount of samples to achieve a reasonable estimation. In speaker recognition, some speakers may have a low number of samples, meaning statistics-based Gaussian metrics are not applicable. We therefore define two Gaussian metrics based on the necessary conditions of a high-dimensional multi-variant Gaussian. A key advantage of these new Gaussian metrics is that they are properties of single samples or pairs of samples, and therefore can be applied to measure the Gaussianality of a small group of samples.

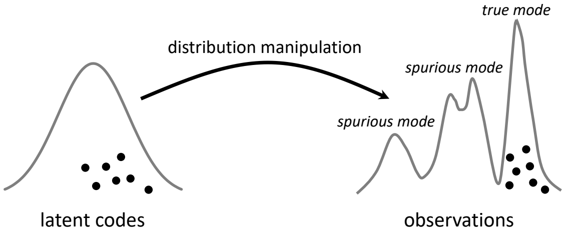


Fig. 4: An NF model trained with ML manipulates the Gaussian prior to maximize the likelihood of the training data. The training samples may be located in one of the modes of the transformed distribution. The other modes are spurious modes, produced by the distribution manipulation.

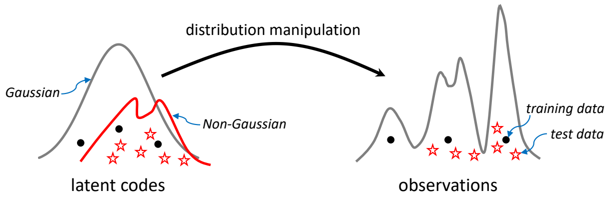


Fig. 5: Overfitting problem caused by limited training data. The black dots are training samples and red start are test samples. The distribution of the latent codes of the test data are not the one assumed by the model.

According to the Gaussian annulus theorem [41], a high-dimensional Gaussian distribution $N(0, \epsilon \mathbf{I})$ possesses two important properties: (1) The majority of the probability mass is concentrated in a thin annulus of width $O(1)$ at radius $\sqrt{\epsilon d}$, where d represents the dimension; (2) Any two samples tend to be orthogonal. Note that these two properties are *necessary* conditions for any dataset sampled from a high-dimensional Gaussian. Therefore, encouraging the samples to gain these properties will directly improve the Gaussianality of the data. This leads to two Gaussian metrics, length and angle, as shown in Fig. 6.

- Length metric: Probability theory shows that for a set of independent random variables $\{x_i\}$ following a standard normal distribution, the square root of the sum of the

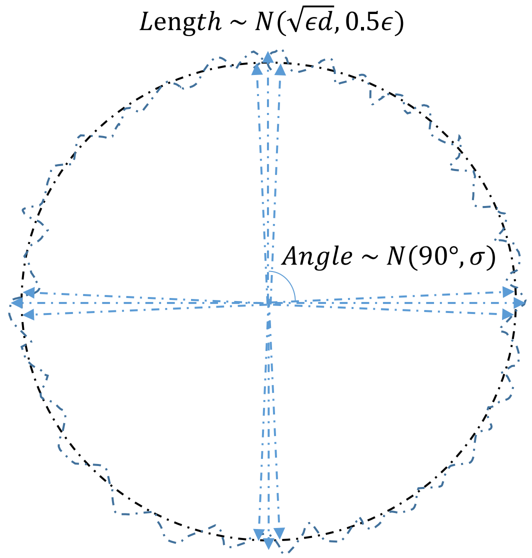

$Length \sim N(\sqrt{\epsilon d}, 0.5\epsilon)$
$Angle \sim N(90°, \sigma)$

Fig. 6: Length and angle metrics of a high-dimensional Gaussian $N(0, \epsilon)$. The length of the samples can be approximated by a Gaussian $N(\sqrt{\epsilon d}, 0.5\epsilon)$, and the angle of two samples can be approximated by a Gaussian $N(90°, \sigma)$.

squares $c = \sqrt{\sum_{i=1}^{k} x_i^2}$ follow a χ distribution:

$$p(c) = \frac{1}{2^{(k/2)-1}\Gamma(k/2)} c^{k-1} e^{-c^2/2},$$

where:

$$\mu = \sqrt{2}\frac{\Gamma((k+1)/2)}{\Gamma(k/2)},$$

and:

$$\sigma^2 = k - \mu^2.$$

Thus the length $\ell(\mathbf{x})$ of any sample $\mathbf{x}$ from a Gaussian $N(0, \epsilon\mathbf{I})$ follows a χ distribution. In a high dimensional space, elementary simulation experiments show that the χ distribution can be approximated by a Gaussian distribution as follows:

$$p(\ell(\mathbf{x})) = N(\sqrt{\epsilon d}, 0.5\epsilon).$$

This leads to a length metric:

$$\mathcal{R}_\ell = -\sum_i ||\ell(\mathbf{x}_i) - \sqrt{\epsilon d}||^2$$

where i indexes all the samples of the distribution. A larger length metric indicates that the samples are more Gaussian.

- Angle metric: Let $\omega(\mathbf{x}_1, \mathbf{x}_2)$ denote the angle between two samples $\mathbf{x}_1$ and $\mathbf{x}_2$, with both sampled from $N(0, \epsilon\mathbf{I})$. The exact form for $\omega(\mathbf{x}_1, \mathbf{x}_2)$ is very complex [42]. However, we do know that it is close to 90°, and therefore can be approximated by a Gaussian $N(90°, \sigma)$:

$$p(\omega(\mathbf{x}_1, \mathbf{x}_2)) = N(90°, \sigma),$$

where σ can be estimated by a simulation experiment. For simplicity, we compute the cosine distance $\phi(\mathbf{x}_1, \mathbf{x}_2)$ and assume it follows a Gaussian distribution $N(0, \xi)$, which leads to the following angle metric:

$$\mathcal{R}_\phi = -\sum_i \sum_j \frac{||\phi(\mathbf{x}_i, \mathbf{x}_j)||^2}{2\xi},$$

where ξ can be found by a simulation experiment. A larger angle metric indicates that the samples are more Gaussian.

Our proposed MG training maximizes the length and angle metrics. According to the Radial Gaussianization (RG) theory [43], MG training will result in a Gaussian distribution. RG theory states that if a distribution is *spherical and symmetric*, and if the length distribution is normalized to a χ distribution, then the transformed samples will follow a Gaussian distribution. In MG training, maximizing the angle metric encourages the samples to distribute spherically and symmetrically, and maximizing the length metric regulates the length distribution to a χ distribution. Therefore, MG training is equivalent to radial Gaussianization, and the RG theory ensures that MG training pursues a transform that produces Gaussian codes.

D. Simulation for MG training

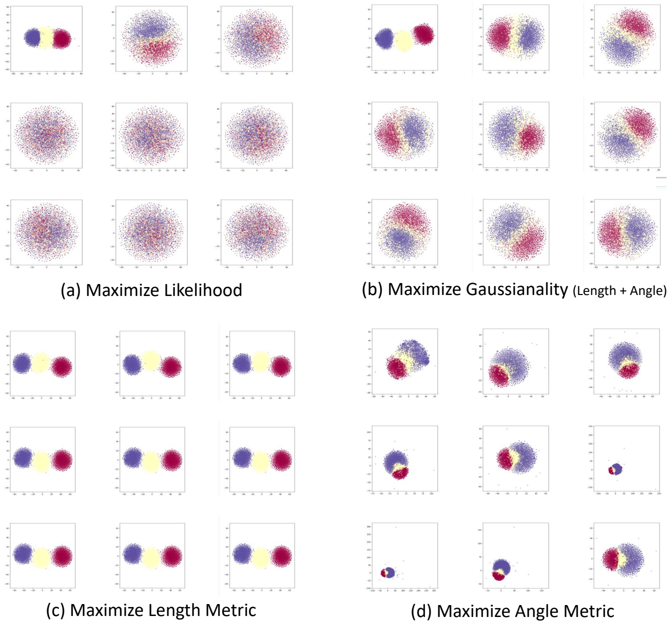
Here, we verify the validity of our new MG training approach, using a simulation experiment. Two NF models are trained using data sampled from a 3-component Gaussian mixture, by employing the ML and MG criteria respectively. The models share the same architecture, with the only difference being the training objective. The MG objective is formulated as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{R}_\ell + \mathcal{R}_\phi = -\sum_i ||\ell(\mathbf{z}_i) - \sqrt{d}||^2 - \sum_i \sum_j \frac{||\phi(\mathbf{z}_i, \mathbf{z}_j)||^2}{2\xi},$$

where we have set $\epsilon = 1$. The NFs trained with the ML and MG criteria are denoted by ML-NF and MG-NF, respectively.

Fig. 7 shows the latent codes generated by ML-NF and MG-NF. By maximizing both the length and angle metrics (Fig. 7 top right), MG-NF can produce Gaussian latent codes. Compared to the latent codes produced by ML-NF (Fig. 7 top left), MG-NF is less distorted, as the samples of different clusters are less mixed up. Fig. 7 also shows that maximizing either the length or angle metrics individually is not sufficient to attain a reasonable NF model.

Fig. 8 shows the likelihood $p(\mathbf{x})$ and Gaussian metrics (length metric $\mathcal{R}_\ell$ and angle metric $\mathcal{R}_\phi$) during ML and MG training. Firstly observe that both ML (solid black lines) and MG training (red dashed lines) improve the Gaussianality of the latent codes. This indicates that the ML criterion, although not providing a guarantee of maximum Gaussianality, generally produces Gaussian codes. Similarly MG training, although not aiming for maximum likelihood, also improves



(a) Maximize Likelihood

(b) Maximize Gaussianality (Length + Angle)

(c) Maximize Length Metric

(d) Maximize Angle Metric

Fig. 7: Latent codes generated by NFs trained with ML (top left), MG (top right), maximum length metric (bottom left) and maximum angle metric (bottom right). Figures plotted using t-SNE [44], best viewed in color.

the likelihood during the training. Compared to ML training, MG training obtains a clearly lower likelihood. More careful analysis shows that the high likelihood with ML training is mainly due to a large entropy term. This suggests that after a large number of iterations, the transform function trained by ML may have been twisted to improve the likelihood of the training samples.

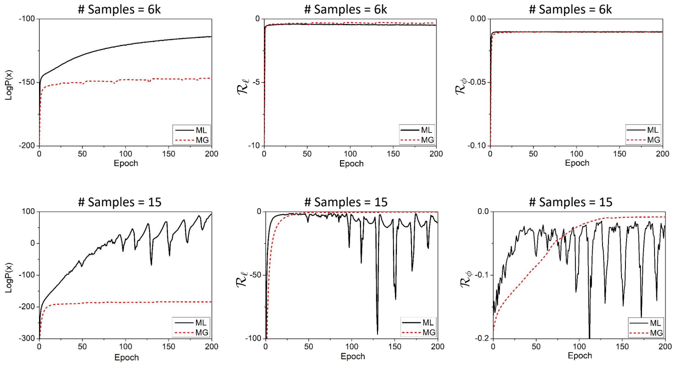


Fig. 8: Likelihood $p(\mathbf{x})$, length metric $\mathcal{R}_\ell$ and angle metric $\mathcal{R}_\phi$ during ML training (solid black) and MG training (dash red). The top row shows the case with sufficient training data, and the bottom row shows the case with only 15 training samples.

We have therefore demonstrated that MG is a valid criterion, and will next demonstrate that ML training is ill-conditioned and suffers from having an unbounded objective function when the training data are limited, while MG training does not have this problem. The experimental setting is the same, but only 15 samples are used for training. The likelihood, length metric and angle metric are shown in Fig. 8 (bottom row). In this scenario, ML training is highly unstable and quickly diverges with unbounded likelihood. In contrast, the MG training is quite stable and gradually improves the three metrics. Note that the values of the two Gaussian metrics change abruptly with ML training, confirming our argument that ML training does not guarantee the Gaussianality of the latent codes, even for the training data.

E. Maximum Gaussianality training for DNF

For the DNF model, we need to normalize the between-class distribution and the within-class distributions of individual speakers. Both tasks can be performed by employing our proposed MG training.

Firstly we normalize the between-class distribution by MG training. Letting $\boldsymbol{\mu}_y$ denote the class mean of the y-th class, the MG objective for the between-class distribution can be formulated as follows:

$$\begin{aligned}\mathcal{R}^{MG}_{BC} &= -\alpha \max(0, \sum_y ||\ell(\boldsymbol{\mu}_y) - \sqrt{\epsilon d}||^2 - \delta) \\ &\quad -\beta \max(0, \sum_y \sum_{y'} ||\phi(\boldsymbol{\mu}_y, \boldsymbol{\mu}_{y'})||^2 - \delta'). \end{aligned} \tag{6}$$

where α and β are hyper-parameters to balance the contribution of the length and angle metrics, respectively. For simplicity, the covariance of the between-class distribution $\boldsymbol{\Sigma}_{\boldsymbol{\mu}}$ has been set to $\mathbf{I}$, which leads to a simple form ($\epsilon = 1$) for the length metric. The parameter ξ has been absorbed by δ' and β.

We have chosen a hinge loss for the two metrics, with the tolerances set to be δ and δ' respectively. This is because the two Gaussian metrics are not the true necessary conditions of the underlying Gaussian (note that they are derived from Gaussian approximations for the true length and angle distributions), and so setting a tolerance will prevent overfitting.

Similarly, the MG criterion can be applied to train the within-class distribution of each individual speaker. The objective function is:

$$\begin{aligned}\mathcal{R}^{MG}_{WC} &= -\alpha \max(0, \sum_y \sum_{y_{\mathbf{z}}=y} ||\ell(\mathbf{z} - \boldsymbol{\mu}_y) - \sqrt{d}||^2 - \delta) \\ &\quad -\beta \max(0, \sum_y \sum_{y_{\mathbf{z}}=y_{\mathbf{z}'}=y} ||\phi(\mathbf{z} - \boldsymbol{\mu}_y, \mathbf{z}' - \boldsymbol{\mu}_y)||^2 - \delta') \end{aligned} \tag{7}$$

where the covariances of the within-class distributions are all set to $\mathbf{I}$. Since the Gaussian metrics have the same form for all the speakers, the distributions of different speakers trained with this criterion tend to be homogeneous.

IV. Experiments

A. Datasets

Three datasets were used in our experiments, VoxCeleb [45], [13], SITW [46] and CNCeleb [47]. VoxCeleb was used for training all models (x-vector systems, PLDA, DNF), while the others were used for performance evaluation.

VoxCeleb: A large-scale speaker database collected by University of Oxford. All the speech signals were collected from open-source media and therefore involve rich variation in channel, style, and ambient noise. This dataset, after removing the utterances shared by SITW, contains 2000+ hours of speech signals from 7000+ speakers. Data augmentation was applied to improve robustness, by using the MUSAN corpus [48] to generate noisy utterances, and the room impulse responses (RIRS) corpus [49] to generate reverberant utterances. The augmented data was used for x-vector model training. For PLDA/DNF training, we only use the 4000 speakers with more than 10 utterances.

SITW: A standard evaluation dataset excerpted from VoxCeleb1, which consists of 299 speakers. In our experiments, the *Eval. Core*, which contains $3,658$ target trials and $718,130$ imposter trials, were used for evaluation. Note that the acoustic condition of SITW is similar to that of the training set VoxCeleb, so this test can be regarded as an **in-domain test**.

CNCeleb: A large-scale free speaker recognition dataset collected by Tsinghua University, containing more than $130k$ utterances from $1,000$ Chinese celebrities. It covers 11 diverse genres, which makes speaker recognition on this dataset much more challenging than on SITW [**?**]. By pair-wised composition, $107,984,700$ trials were constructed, including $119,983$ target trials and $107,864,717$ imposter trials. Note that the acoustic condition of CNCeleb is quite different from that of VoxCeleb, and so the test on CNCeleb was used as an **out-of-domain test**.

B. Model settings

Our speaker recognition system involves three components: an x-vector frontend that produces speaker vectors, a normalization model that regularizes the distribution of the speaker vectors, and a scoring model that produces pair-wise scores for genuine/imposter decision.

1) X-vector frontend: The x-vector frontend was built with the Kaldi toolkit [50], following the SITW recipe. The main architecture contains three components. The first is the feature-learning component, which involves 5 time-delay (TD) layers to learn frame-level speaker features. The slicing parameters for the five TD layers are: $\{t\text{-}2, t\text{-}1, t, t\text{+}1, t\text{+}2\}$, $\{t\text{-}2, t, t\text{+}2\}$, $\{t\text{-}3, t, t\text{+}3\}$, $\{t\}$, $\{t\}$. The second is the statistic pooling component, which computes the mean and standard deviation of the frame-level features from a speech segment. Finally, the third is the speaker-classification component, which discriminates between speakers. This has 2 full-connection (FC) layers and the size of its output is $7,185$, corresponding to the number of speakers in the training set. Once trained, the 512-dimensional activations of the penultimate FC layer are read out as an x-vector. Length normalization is employed before the x-vectors are fed to the normalization or scoring component.

2) Normalization model: All normalization models are based on the DNF architecture, though different criterion may be employed to normalize the between-class and within-class distributions. All models are based on the same masked autoregressive flow (MAF) architecture [51], which consists of 10 MAF blocks, with each block being an inverse autoregressive transformation. The Adam optimizer was used to train the model. More details of the model can be found in the original DNF paper [31].

The nomenclature for the DNF models is **DNF-[N/L/G]-[L/G]**, where the first and second options are the training criterion for the between-class and the within-class distributions respectively; and **N**, **L** and **G** denote None, ML and MG respectively. For example, **DNF-L-LG** means the between-class distribution is normalized by ML and the within-class distributions are normalized by combining ML and MG.

The ML objectives are $\mathcal{R}^{ML}_{BC}$ and $\mathcal{R}^{ML}_{WC}$ for the between-class and within-class distributions, as defined in Eq. (5) and Eq. (4) respectively; the MG objectives are $\mathcal{R}^{MG}_{BC}$ and $\mathcal{R}^{MG}_{WC}$ for the between-class and within-class distributions, as defined in Eq. (6) and Eq. (7) respectively. Note that for all the **DNF-[N/L/G]-G** models, the entropy term is involved in the objective as it improves numerical stability of the training process.

We will test 6 DNF variants: **DNF-N-L**, **DNF-L-L**, **DNF-G-G**, **DNF-N-LG**, **DNF-G-L**, **DNF-G-LG**. Table I summarizes the details for these models. Note that **DNF-N-L** is the conventional DNF model presented in [31]. In all these DNF variants, it was found the performance is not sensitive to the setting of the hyperparameters. We therefore empirically set them as follows: $\alpha = 10$, $\delta = 0.03$, $\delta' = 0.002$. β is set to 10 for within-class normalization, but for between-class normalization it is set to 500 in order to enforce an even distribution of the class means.

TABLE I: Training criteria of DNF variants.

Models	Between-Class Criterion	Within-Class Criterion
DNF-N-L	N/A	Maximum Likelihood
DNF-L-L	Maximum Likelihood	Maximum Likelihood
DNF-G-G	Maximum Gaussianality	Maximum Gaussianality
DNF-G-L	Maximum Gaussianality	Maximum Likelihood
DNF-G-LG	Maximum Gaussianality	Maximum Likelihood & Gaussianality

3) Scoring model: Two scoring models were used in this study: the simple **Cosine scoring** which is based on the cosine distance, and more complicated **PLDA scoring** which is based on PLDA [28].

C. Quantitative analysis

This section will qualitatively investigate the behavior of the MG training when applied to normalize the between-class and within-class distributions.

First, we study the effect of the ML/MG training on the between-class distribution by computing the variation on each dimension for the x-vectors and the latent codes produced by three DNF models: **DNF-N-L**, **DNF-L-L**, **DNF-G-L**. The curves of the sorted values are shown in Fig. 9. These curves reflect the distribution of the between-class variation over the dimensions.

It can be seen that the conventional DNF (**DNF-N-L**) does not change the variation distribution in a significant way. With our proposed MG training (**DNF-G-L**), the variation is more evenly distributed among all the dimensions, and the overall variation is increased. This is an expected result, as our MG training encourages the distribution of the class means on a spherical surface. The improved overall variation implies improved between-class discrimination, hence better performance in general; and the evenly distributed variation is mostly desirable for cosine scoring. ML training (**DNF-L-L**) changes the between-class distribution in the same way as MG training, which is also expected, as the Gaussian prior placed by the ML training assumes an even distribution for the between-class variation. However, the change with ML training is not as significant as with our MG training. This demonstrates that the MG criterion is stronger than the ML criterion with regard to regulating/normalizaing the between-class distribution.

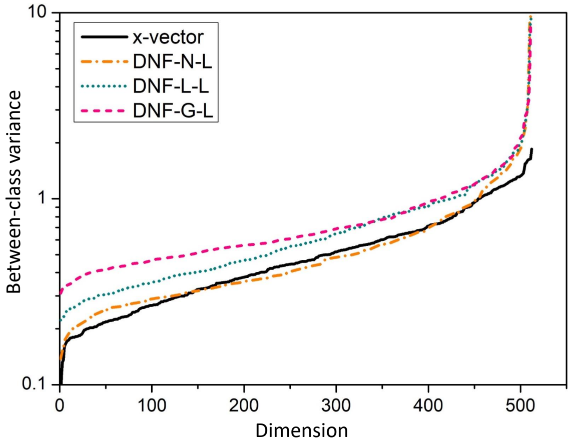


Fig. 9: Between-class variation distributed amongst dimensions, computed on the original x-vectors and the latent codes produced by three DNF models.

Our second experiment examines the effect of ML/MG training on the within-class distributions. The first analysis is for the homogeneity. To do this, we compute the dimension-averaged variation of the within-class distribution for each speaker, and then plot the distribution of variation values in Fig. 10. This distribution reflects the homogeneity of the speakers, an important property for both cosine and PLDA scoring. The distributions of the original x-vectors and the latent codes produced by the three DNF models (**DNF-N-L**, **DNF-G-L** and **DNF-G-G**) are plotted on Fig. 10. Here, it can be seen that for the original x-vectors, the variations of different speakers are broadly distributed, indicating that the distributions of different speakers are significantly heterogeneous. With the conventional DNF (**DNF-N-L**), the homogeneity is not improved, supporting our argument that ML training cannot guarantee that the latent codes are as distributed as the model assumes. MG training on the between-class distribution (**DNF-G-L**) does not improve this situation, but our new MG training approach used on the within-class class distributions (**DNF-G-G**) improves the homogeneity substantially.

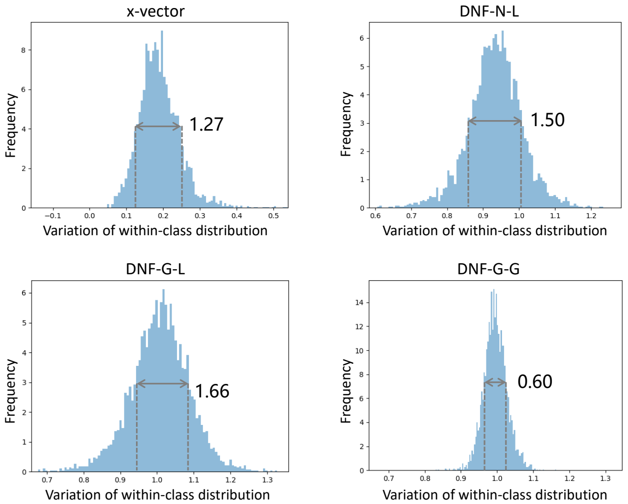


Fig. 10: The distribution of the dimension-averaged variation of the within-class distributions, computed on the original x-vectors and the latent codes produced by three DNF models. The x-axis represents the variation of the within-class distribution.

The second analysis investigates the Gaussanality, for which we compute the mean and variance of the length metric $\mathcal{R}_l$ and the angle metric $\mathcal{R}_\phi$ as measures. The results shown in Table II demonstrate that the Gaussianality of the within-class distributions is greatly improved by all the DNF models, however the improvement provided by our MG training (**DNF-G-G**) is more significant than provided by ML training (**DNF-G-L**).

TABLE II: Gaussianality of the within-class distributions, measured by the length metric and angle metric.

Models	Length Metric ($\mathcal{R}_l$)		Angle Metric ($\mathcal{R}_\phi$)	
	Mean	Var	Mean	Var
x-vector	-177.79	3457.53	-3.20e-2	1.35e-6
DNF-N-L	-6.50	77.17	-2.09e-3	2.67e-10
DNF-G-L	-6.45	97.00	-2.09e-3	2.78e-10
DNF-G-G	-1.62	4.33	-2.00e-3	2.56e-10

D. Speaker recognition results

This section focuses on speaker recognition performance, using the databases discussed previously, the in-domain dataset (SITW) and the out-of-domain dataset (CNCeleb). The results in terms of the equal error rate (EER) are reported in Table III.

1) In-domain results: Firstly, looking at the in-domain results. For the cosine scoring, it can be seen that conventional DNF (**DNF-N-L**) shows significant improved performance over the x-vector baseline (8.53% vs 17.20%), verifying results presented in previous research [31]. This demonstrates the effectiveness of the initial ML-based within-class normalization. However, applying ML-based between-class normalization (**DNF-L-L**) does not offer any performance benefit, showing reduced performance (8.53% vs 10.47%). This is a little surprising as Fig. 9 has shown that the between-class discrimination is improved with this normalization. More detailed analysis shows that this is because the ML-based between-class normalization significantly reduces the length metric, thus making the between-class distribution less Gaussian (compared to **DNF-N-L**). This indicates that ML-based between-class normalization does not work.

When replacing the existing ML-based normalization with our new MG-based normalization for the between-class distribution, **DNF-G-L** obtains significant performance improvement compared to **DNF-L-L** (6.89% vs 10.47%) and **DNF-N-L** (6.89% vs 8.53%). This is an expected result, as our MG-based between-class normalization *evenly* distributes the class means across a *spherical* surface (as shown in Fig. 9), which makes cosine scoring more applicable, compared to conventional DNF where the class means are not spherically distributed. Additional performance gains are identified when substituting the ML-based normalization for the MG-based normalization for the within-class distribution, as shown by the comparative results of **DNF-G-G** and **DNF-G-L** (6.30% vs 6.89%). Combining the ML-based and MG-based normalization for the within-class distribution does not exhibit further advantages, as shown by the comparative results of **DNF-G-LG** and **DNF-G-G** (6.42% vs 6.30%).

In summary, the results above demonstrate that our proposed MG-based normalization is effective when applied to both the between-class and within-class distributions, and even more effective when compared to the ML-based normalization.

The PLDA scoring results show the same trends as cosine scoring when comparing the results with different DNF variants. One key difference is that with PLDA scoring, the performance gain offered by the MG-based between-class normalization (3.66% vs 3.45%) is not as significant as with cosine scoring (8.53% vs 6.89%). This is understandable, as PLDA scoring does not require a spherical distribution for the class means, and so the improvement contributed by our MG-based between-class normalization is limited.

2) Out-of-domain results: With regard to out-of-domain results, it can be seen that with cosine scoring, the relative performance with different DNF variants is highly consistent with the results obtained on SITW (the in-domain test). This indicates that the performance gains obtained with the DNF models, especially with the MG-based normalization (either between-class or within-class distributions), are reliable. This in turn indicates that to a large extent, the DNF-based normalization models are generalizable with respect to data mismatch.

The PLDA results are a little more complex, as the data mismatch affects not only the normalization models (i.e., DNFs), but also the scoring model (i.e., PLDA). Table III, shows that there is no significant difference between the performance of different DNF variants. Arguably the only consistent observation is that normalization on the between-class distribution improves the model performance, no matter whether it is based on ML training or MG training.

Comparing the cosine scoring and the PLDA scoring results, it can be clearly seen that with DNF-based normalization, particularly with MG-based training on both the between-class and within-class distributions (**DNF-G-G**), the performance gap between the best performance with the two scoring methods becomes rather marginal (12.13% vs 11.39%), which is not the case in the in-domain test (6.30% vs 3.37%). This can be attributed to the fact that cosine scoring does not rely on a separate statistical model and thus suffers less from the data mismatch problem, making it more reliable in the out-of-domain test.

E. Results with different training dataset sizes

Here, we examine the performance of different DNF variants, using the in-domain test (SITW) and different volumes of data. The number of speakers used for training varies between 1000 to 4000, with the results shown in Fig. 11. A clear benefit can be seen from using MG training in all cases. In particular, the cosine scoring is the most noteworthy (top left in Fig. 11). This indicates that with our new MG-based normalization, good performance can be obtained with only 1000 speakers. More speakers do not offer much help with cosine scoring. This indicates that DNF models do not require too many training data, if the between-class and within-class distributions are well constrained. For PLDA scoring (Fig. 11, top right), a larger number of speakers is important to obtain a good performance. Since DNFs do not require many data, the data required here is mainly used for training a robust PLDA model.

We also investigated varying the number of utterances per speaker. Three configurations are tested, with 2, 5, and 30 utterances per speaker respectively. The results are shown in Fig. 11. The cosine scoring performance (bottom left in Fig. 11) shows that performance plateaus after 5 utterances per speaker with conventional DNF. However, with our proposed MG training, the performance is significantly improved in all test conditions. One interesting observation is that MG training on the within-class distributions (**DNF-G-G** and **DNF-G-LG**) improves the performance with even 2 utterances per speaker. This clearly demonstrates the advantage of our MG training in scenarios with very limited data. However, this advantage is almost entirely lost when the scoring is based on PLDA (Fig. 11, bottom right). In this case, having sufficient data for each speaker is important, as the PLDA model requires the data to obtain a reasonable estimation for the within-class variance.

TABLE III: EER(%) results on SITW and CNCeleb with DNF variants.

Models	Between-Class	Within-Class	SITW		CNCeleb	
	Criterion	Criterion	Cosine	PLDA	Cosine	PLDA
x-vector	N/A	N/A	17.20	5.30	16.32	13.03
DNF-N-L	N/A	ML	8.53	3.66	14.22	11.82
DNF-L-L	ML	ML	10.47	3.72	15.83	**11.39**
DNF-G-G	MG	MG	**6.30**	**3.37**	**12.13**	11.72
DNF-G-L	MG	ML	6.89	3.45	13.99	11.46
DNF-G-LG	MG	ML+MG	6.42	3.36	12.96	11.51

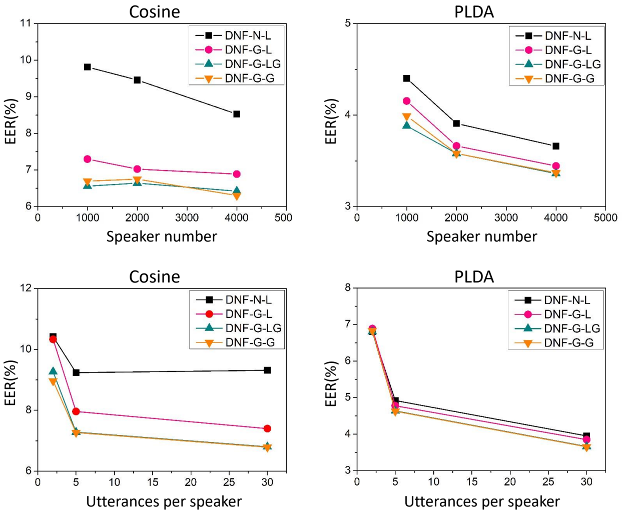


Fig. 11: Performance with different DNF variants and different training datasets, with varying number of speakers (top) and number of utterances per speaker (bottom).

F. Discussion

The results presented in this paper have clearly demonstrated that our new MG training approach is effective when applied to normalize both the between-class and within-class distributions. For the between-class normalization, it evenly distributes the class means to a spherical surface, leading to more regulated between-class distributions and improved between-class discrimination. For the within-class normalization, it maximises the length metric based on the same length distribution for different speakers, leading to improved homogeneity among different speakers. By these normalizations, the latent codes are enforced to be distributed as a linear Gaussian, a desirable property for both cosine scoring and PLDA scoring. The existing ML training method cannot guarantee this property, due to the problems discussed previously in Section III-B.

The remarkable performance with cosine scoring deserves more discussion. As mentioned previously, PLDA scoring is optimal if the speaker vectors are linear Gaussian and the between-class and within-class variations can be well estimated [30]. Cosine scoring is a good approximation for PLDA scoring under the linear Gaussian assumption, but it requires the between-class distribution to be spherical. Our MG-based normalization (for both between-class and within-class distributions) offers more benefit with cosine scoring, since it enforces the requirement for the between-class distribution to be spherical. This has been clearly observed from the in-domain results in Table III.

Moreover, PLDA scoring relies on an explicit statistical model while cosine scoring does not. This leads to an additional advantage for cosine scoring as it suffers less from data mismatch, as demonstrated by the out-of-domain results in Table III. Putting them together, our proposed MG-based normalization makes the cosine scoring a strong competitor to PLDA scoring, and even a more preferable choice in complex conditions.

V. Conclusions

This paper presented a new maximum Gaussianality (MG) training approach for DNF-based speaker vector normalization models. We firstly analyzed the widely used maximum likelihood (ML) training and found that an ML-based DNF cannot guarantee that the latent codes produced are linear Gaussian, a property that is essential for speaker vector normalization. Our MG training approach solves this problem by maximizing the Gaussianality of the latent codes directly. Our experiments on the SITW corpus show that significant performance improvement was obtained when applying MG training to normalize either the between-class distribution or the within-class distributions, and that additional performance improvement was obtained when both were MG normalized. Results on the CNCeleb out-of-domain dataset demonstrated that the improvement can be effectively applied to unseen datasets, especially when cosine scoring is used, demonstrating that MG-based DNFs are not only effective but also generalizable.

MG is a general criterion and can be employed to train various models that aim to establish a Gaussian latent space. Crucially, with our successfully demonstrated MG training method, it is not necessary to compute the entropy term and so any invertible structure can be used. This eliminates the primary hindrance for flow-based models and may significantly extend their application. Future work will study the behavior of MG training, with different model structures and different training schemes. We will also investigate applying the MG training to other models and other applications, for instance speech recognition and anomaly detection.

References

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *IEEE international conference on Acoustics, speech, and signal processing (ICASSP)*, vol. 4. IEEE, 2002, pp. IV–4072.

[3] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[4] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[7] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1542–1546.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[9] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[10] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[11] F. R. rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[12] S. O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2018 NIST Speaker Recognition Evaluation," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 1483–1487.

[13] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086–1090.

[14] J. weon Jung, H.-S. Heo, J. ho Kim, H. jin Shim, and H.-J. Yu, "RawNet: Advanced End-to-End Deep Neural Network Using Raw Waveforms for Text-Independent Speaker Verification," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 1268–1272.

[15] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2252–2256.

[16] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[17] J. S. C. W. Xie, A. Nagrani and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5791–579.

[18] N. Chen, J. Villalba, and N. Dehak, "Tied mixture of factor analyzers layer to combine frame level representations in neural speaker embeddings," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2948–2952.

[19] L. Li, D. Wang, C. Xing, and T. F. Zheng, "Max-margin metric learning for speaker recognition," in *10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2016, pp. 1–4.

[20] W. Ding and L. He, "MTGAN: Speaker verification through multitasking triplet generative adversarial networks," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3633–3637.

[21] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. Brudno1, "Centroid-based deep metric learning for speaker recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3652–3656.

[22] Z. Bai, X.-L. Zhang, and J. Chen, "Partial auc optimization based deep speaker embeddings with class-center learning for text-independent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6814–6818.

[23] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 361–365.

[24] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep Speaker Embedding Extraction with Channel-Wise Feature Responses and Additive Supervision Softmax Loss Function," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2883–2887.

[25] R. Li, N. L. D. Tuo, M. Yu, D. Su, and D. Yu, "Boundary discriminative large margin cosine loss for text-independent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6321–6325.

[26] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. Cernocky, "On the Usage of Phonetic Information for Text-Independent Speaker Embedding Extraction," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 1148–1152.

[27] T. Stafylakis, J. Rohdin, O. Plchot, P. Mizera, and L. Burget, "Self-Supervised Speaker Embeddings," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2863–2867.

[28] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 531–542.

[29] L. Li, Z. Tang, Y. Shi, and D. Wang, "Gaussian-constrained training for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6036–6040.

[30] D. Wang, "Remarks on optimal scores for speaker recognition," CSLT@Tsinghua University, 2020. [Online]. Available: http://wangd.cslt.org/public/pdf/nl.pdf

[31] Y. Cai, L. Li, D. Wang, and A. Abel, "Deep normalization for speaker vectors," *arXiv preprint arXiv:2004.04095*, 2020.

[32] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and

N. Brümmer, “Discriminatively trained probabilistic linear discriminant analysis for speaker verification,” in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 4832–4835.

[33] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in *Odyssey*, vol. 14, 2010.

[34] D. A. Van Leeuwen and N. Br, “The distribution of calibrated likelihood-ratios in speaker recognition,” in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2013, pp. 1619–1623.

[35] S. Cumani and P. Laface, “Tied normal variance - mean mixtures for linear score calibration,” in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2019.

[36] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *arXiv preprint arXiv:1912.02762*, 2019.

[37] W. Rudin, *Real and complex analysis*. Tata McGraw-hill education, 2006.

[38] E. G. Tabak and C. V. Turner, “A family of nonparametric density estimation algorithms,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, 2013.

[39] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, “Do deep generative models know what they don’t know?” *arXiv preprint arXiv:1810.09136*, 2018.

[40] P. Comon, “Independent component analysis, a new concept?” *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[41] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of data science*. Cambridge University Press, 2020.

[42] *Products of Random Variables*. Springer, 2002, pp. 49–60.

[43] S. Lyu and E. P. Simoncelli, “Nonlinear extraction of independent components of natural images using radial gaussianization,” *Neural computation*, vol. 21, no. 6, pp. 1485–1519, 2009.

[44] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[45] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” *arXiv preprint arXiv:1706.08612*, 2017.

[46] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, “The speakers in the wild (SITW) speaker recognition database.” in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2016, pp. 818–822.

[47] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, “Cn-celeb: a challenging chinese speaker recognition dataset,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[48] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.

[49] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[50] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[51] G. Papamakarios, T. Pavlakou, and I. Murray, “Masked autoregressive flow for density estimation,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2338–2347.

Yunqi Cai studied as a phd candidate at Institute of Physics ,Beijing, from 2014 to 2018. During 2016-2017 he went to Oak ridge national lab, the United States, as a visiting scholar under the funding of the Joint PhD Training Program scholarship offered by UCAS 2016. He received his Ph.D. degree from Institute of Physics, Chinese Academy of Sciences, Beijing in 2018. He is now a Postdoctoral Fellow with the Center for Speech and Language Technologies (CSLT) and the Department of Computer Science at Tsinghua University, Beijing, China. His research interests are focused on the machine learning algorithms of speech and language technology.

Lantian Li received the B.Sc. degree from China University of Mining and Technology, Beijing in 2013. He received the Ph.D. degree from the Department of Computer Science, Tsinghua University in 2018. Since 2018, he has been with the Center for Speech and Language Technology (CSLT), Tsinghua University as a postdoctoral fellow. His research interest is speaker recognition with machine learning methods.

Dong Wang (M’09, SM’16) received the B.Sc. and M.Sc. degrees in computer science from Tsinghua University in 1999 and 2002. He received the Ph.D. degree (supported by a Marie Curie fellowship) from CSTR, University of Edinburgh, in 2010. He was employed with Oracle China during 2002–2004 and IBM China during 2004–2006. He joined CSTR, University of Edinburgh, in 2006 as a Research Fellow. From 2010 to 2011, he was with EURECOM as a Postdoctoral Fellow, and from 2011 to 2012, was a Senior Research Scientist with Nuance. He is now an Associate Professor with Tsinghua University, Beijing, China.

Andrew Abel is a lecturer in Computing Science at Xi’an Jiaotong-Liverpool University (XJTLU) in Suzhou, China, and received his Ph.D. from the University of Stirling in Scotland in 2013, after conducting research into developing signal-image processing algorithms for enabling multi-modal speech processing technologies. Before XJTLU, he worked as a researcher at Stirling, working on the testing of novel MEMS/CMOS microphone technology, at Anhui University in Hefei focusing on image processing, and more recently at Stirling as a research-CI on the development of next generation hearing aid technology aimed at developing hearing aids that can “see”. His research interests are focused on the use of multiple modalities (particularly vision) to estimate and process speech and communication.