2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP) | 978-1-7281-6994-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISCSLP49672.2021.9362090

Speaker Embedding Augmentation with Noise Distribution Matching

Xun Gong[1], Zhengyang Chen[1], Yexin Yang[1], Shuai Wang[1], Lan Wang[2] and Yanmin Qian[1]

[1] MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
[2] CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems,
Shenzhen Institutes of Advanced Technology
{gongxun,zhengyang.chen,yangyexin,feixiang121976,yanminqian}@sjtu.edu.cn,
lan.wang@siat.ac.cn

Abstract

Data augmentation (DA) is an effective strategy to help building robust systems with good generalization ability. In the embedding based speaker verification, data augmentation could be applied to either the front-end embedding extractor or the back-end PLDA. Unlike the conventional back-end augmentation method which adds noises to the raw audios and then extracts augmented embeddings, in this work, we proposed a noise distribution matching (NDM) based algorithm in the speaker embedding space. The basic idea is to use distributions such as Gaussian to model the difference between the clean and original augmented noisy speaker embeddings. Experiments are carried out on SRE16 dataset, where consistent performance improvement could be obtained by the novel NDM. Furthermore, we found that the proposed NDM could be robustly estimated using only a small amount of training data, which saves time and disk cost compared to the conventional augmentation method.

Index Terms: speaker embedding, data augmentation, distribution matching, speaker verification

1. Introduction

Speaker verification (SV) aims to verify a user's claimed identity, given his or her speech segment. Recently, the deep neural network (DNN) based speaker embedding learning has boosted the performance of speaker verification task and became the dominant approach [1, 2]. Researchers have investigated different architectures [3, 4, 5], different loss functions [6, 7, 8, 9, 10, 11, 12, 13], and different model compensation methods [14] to further improve the system's performance. Currently, the x-vector style deep speaker embeddings are the dominating approach in the SV tasks, which shows great superiority to conventional methods such as i-vector [15] and d-vector [1]. The key idea behind x-vector [4] or r-vector [5] is the segment-level optimization in the training stage, which is consistent with the evaluation stage. A typical speaker embedding based SV system consists of two parts, the embedding extractor from which we extract speaker embeddings and the scoring back-end which makes the final decision. In most cases, we use the probabilistic linear discriminant analysis (PLDA) as the scoring back-end.

Despite the great advancement of SV research thanks to the deep speaker embedding learning, building a usable SV system for real-world applications still faces challenges. The first challenge is the lack of data. To effectively train deep models in a supervised manner, a vast amount of labeled data is needed, which is not always available. The second challenge is that the environment could be very complicated, where different kinds of noises could easily corrupt the speech. Data augmentation (DA) is a prevalent method to deal with both challenges. The basic idea of DA is to increase the quantity and diversity of the training data so that we could train a more robust system with better generalization ability. It would also be very effective to augment the data with the noises in the target application scenarios if it's available.

In terms of speaker embedding learning, data augmentation could be applied to either the front-end embedding extractor [4] or the back-end PLDA [4, 16, 17]. For the front-end augmentation, we usually manually add noises or reverberation to the clean audios to generate the augmented version, which would be further used to train the speaker embedding extractors. For the back-end augmentation which prepares the data for PLDA training [4], the conventional way is extract noisy embeddings from the augmented audios. In the literature [16, 17], deep generative models such as generative adversarial (GAN) [18] and variational auto-encoder (VAE) [19] are used to describe the distribution of noisy speaker embeddings, which directly augment the speaker embeddings for the back-end PLDA.

In this work, instead of using complex deep generative models to learn the distribution of the noisy embeddings, we proposed a simple but effective back-end augmentation method, which is named as noise distribution matching (NDM). In the proposed NDM augmentation strategy, we assume the difference (the pure noise part) between the clean and noisy embeddings could be modeled by simple distributions such as Gaussian. After estimating the parameters of the noise distribution, we sample noise from the distribution and directly add it to the clean embedding to generate a noisy embedding. Experiments carried out on the SRE16 [20] dataset show that, despite its simpleness, our proposed NDM based back-end augmentation method could achieve impressive improvement compared to the baseline with no PLDA augmentation and outperforms the conventional manual augmentation method. Furthermore, we show that the NDM could be robustly estimated using a small amount of training data, which saves both time and disk.

2. Embedding based speaker verification

In this section, we will briefly go through the embedding based speaker verification. Two different front-end embeddings, and the back-end PLDA model will be introduced.

2.1. Front-end embeddings

2.2. x-vector

In the x-vector framework, a time-delay neural network (TDNN) is trained to discriminate different speakers in the training set. Acoustic features first go through several frame-level layers, after which a statistics pooling layer is adopted to aggregate the frame-level deep features into a segment-level representation. One or more embedding layers can be incorpo-

rated in the segment-level layers to extract speaker embeddings, and more details could be referred to in [4].

2.3. r-vector

Besides the TDNN architecture, ResNet also showed impressive results for speaker embedding learning. Unlike the 1D convolution used in TDNN, ResNet adopts 2D convolution as the main computation paradigm. Following the terminology in [5], we denote the embeddings extracted from the ResNet as r-vector. More details about this model could be found in [5].

2.4. Back-end PLDA

Probabilistic Linear Discriminant Analysis (PLDA) is a popular scoring back-end for embedding based speaker verification [4, 21, 15]. The PLDA model assumes embeddings are generated following a probabilistic model, where the log-likelihood ratio of target and non-target hypothesis could be computed for a given recording as the scores. In the SRE16 evaluation condition, to utilize the provided unlabeled in-domain data, the simple unsupervised PLDA adaptation method implemented in Kaldi is used. The basic idea is to take the unlabeled embeddings from the target domain and use their mean and variance to adapt the PLDA matrices, and implementation details can be referred to in "ivector-adapt-plda.cc" in Kaldi.

3. Embedding augmentation with noise distribution matching

As mentioned in Sec. 1, the augmentation for the speaker embedding learning could be either applied to the front-end extractor or the back-end PLDA. The conventional front-end data augmentation method is to manually add noise or reverberation to the raw audios. For instance, in the Kaldi recipe [22] for speaker verification, noises from the MUSAN dataset are added to the original audios to generate the corrupted version, and both data are pooled together for the speaker embedding extractor training. For the back-end augmentation, the simplest way is to extract noisy embeddings from the augmented audios. In previous works[16, 17], the authors investigated to use deep generative models to learn the distribution of noisy embeddings and directly generate sample new speaker embeddings from the learned distribution. In this work, instead of directly model the distribution of the noisy embeddings, we assume a "noisy" speaker embedding $\mathbf{e}_{noisy}$ (extracted from augmented audio) can be decomposed into a clean speaker embedding $\mathbf{e}_{clean}$ (extracted from the original audio) and a residual noise term $\mathbf{e}_{noise}$ in equation 1, in which the $\oplus$ represents the combination operation.

$$\mathbf{e}_{noisy} = \mathbf{e}_{clean} \oplus \mathbf{e}_{noise} \tag{1}$$

Based on this assumption, we introduce a novel noise distribution matching (NDM) based back-end augmentation method, which aims to directly learn a distribution to model the difference between the paired $\mathbf{e}_{clean}$ and $\mathbf{e}_{noisy}$, i.e. the noise term $\mathbf{e}_{noise}$. The NDM based back-end augmentation pipeline is shown in Fig. 1. We first extract the parallel "clean" and "noisy" embeddings from the original and augmented audios, respectively. Then the difference between the paired embeddings is calculated by a simple element-wise subtract operation, which would be used to estimate the distribution of the residual noise term[1]. Here, we tried several common distributions, which would be discussed in Section 3.1. After estimating the distribution of $\mathbf{e}_{noise}$, we could directly generate "noisy" embedding by simply sampling a noise embedding from the estimated distribution and adding it to the $\mathbf{e}_{clean}$.

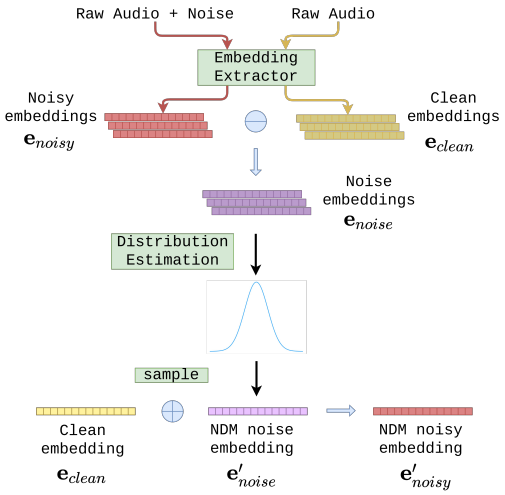

Figure 1: *NDM based embedding augmentation*

3.1. Noise Distributions

To implement NDM in an elegant way, we hypothesize that

1. Different dimensions of the noise term $\mathbf{e}_{noise}$ are i.i.d (independent and identically distributed), thus we could focus on every single dimension.
2. Each dimension of $\mathbf{e}_{noise}$ could be modeled by simple distributions such as univariate Gaussian.

Three different distributions, Uniform ($\mathcal{U} = \mu(a < x < b)$), Laplace (exponential, $\mathcal{L} = \frac{1}{2b}\exp(-\frac{|x-\mu|}{b})$) and Gaussian (normal, $\mathcal{N} = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$) are investigated in the proposed NDM framework. We will first validate the performance achieved by each distribution and select the best one for further experiments.

3.2. Estimation

The estimation of the parameters of the distributions follows the standard Maximum likelihood Estimation (MLE), which would not be repeated. Here are some more details of the training data for this estimation. As mentioned above and shown in Fig. 1, the noise distribution of $\mathbf{e}_{noise}$ depends on the original noise added to the raw audios. In our experiments, we follow the standard manual augmentation method in the Kaldi recipe [23], where four different noise types are considered[2]. We could either pool all four types of noises together and estimate one total distribution, or we can also treat every noise type independently and estimate a distribution for each of them. Considering the simpleness of the chosen distribution, the latter strategy is more reasonable and indeed outperforms the former in our experiments (Sec.5.1). Thus we are able to model each noise type independently.

When preparing the parallel embedding pairs for the NDM estimation, although we could add noises to the original audios and extract the corresponding noisy embeddings for all of them, the process of both adding noise and extracting noisy embedding could be disk- and time-consuming. It would be interesting to see whether we could still make a robust estimation when we

[1]Here, we use subtraction as the decomposition for simplicity, more complicated methods will be investigated in future work

[2]Noise, Music, Babble from the MUSAN dataset and Reverberation

only use a small subset of the whole dataset, and related experiments and analysis could be found in Sec.5.3. We found that only with less than 10% of the training data, the good performance could still be maintained with the newly proposed noise distribution matching approach, which is another big advantage of the proposed NDM.

4. Experimental setup

4.1. Dataset

For training, following the settings in [16], SWBD portion and SRE portion are used. The SWBD portion consists of Switchboard phases 2,3 and Switchboard Cellular 1,2, while the SRE portion contains the NIST SRE 2004-2010. For evaluation, the standard SRE16 evaluation set is used, which consists of Tagalog and Cantonese conversational telephone speech. The unlabelled development set (SRE16 major) is provided for unsupervised PLDA adaptation.

Speaker embedding extractors are trained on both SWBD and SRE portions, while the PLDA training and NDM-based embedding augmentation are carried out only on the SRE portion. When training the extractors, speakers with a small amount of speech are filtered out, resulting in a training list of 3419 speakers. In our experiments, the data preparation follows the Kaldi recipe [22, 23] with two different settings: 1) 40-dimensional Fbanks are used instead of MFCC. 2) Besides the augmented extractor, we also included the clean version which is only trained on the original audios.

4.2. System Configuration

4.2.1. Embedding extractor

- **x-vector:** We use the standard x-vector framework, which consists of 5 frame-level time-delay layers, a statistic pooling layer and 2 segment-level embedding layers. The speaker embeddings are extracted from the first embedding layer, and the dimension is set to 512.
- **r-vector:** The same structure described in [5] is adopted in this work, which is a 34-layer ResNet, and the dimension of r-vector is set to 256. More details could be referred to in [5].

Both models are optimized using SGD with a momentum set to 1e-4. The learning rate is set to 0.1 initially and gradually reduced to 1e-6.

4.2.2. Scoring strategy

Standard Kaldi SRE16 scoring strategy [22] is adopted. LDA is first applied to reduce speaker embeddings to 150 and 128 dimensions for x-vector, r-vector respectively. PLDA is adopted as the scoring back-end. We trained the PLDA using 50644 clean embeddings. And when we augmented the training set for PLDA, another 50644 "noisy" embeddings are added. Besides, the unsupervised PLDA adaptation mentioned in Sec.2.4 is applied to compensate for the domain mismatch.

5. Results and analysis

5.1. Distribution Selection for Noise Distribution Matching

In this section, we will explore the impact of different distribution functions for estimating the $\mathbf{e}_{noise}$. Three kinds of distribution introduced in section 3.1 will be investigated. As a validation experiment, we chose the basic setup, i.e., clean extractors trained on the original audios. The augmented audios are not used for the extractor training, but only for noisy embedding extraction. Following the NDM approach described in Sec.3, we estimate the noise distribution and sample the noise term from it for embedding augmentation. The results achieved by different distribution types can be found in Table. 1.

Table 1: *Performance (EER [%]) comparison using different distribution functions for noise distribution matching with TDNN x-vector model.*

Distribution	Model			
	TDNN		Resnet	
	Tagalog	Cantonese	Tagalog	Cantonese
-	15.34	5.81	13.26	4.21
Laplace	15.61	6.11	13.72	5.36
Uniform	14.46	5.35	13.11	4.04
Normal	**13.48**	**4.75**	**12.05**	**3.60**

As shown in Table 1, the proposed NDM using uniform and normal distributions both can get obvious improvements, and the normal distribution achieves the best performance, which will be used for noise modeling in the following experiments.

5.2. NDM results on SRE16

Since the proposed NDM is a back-end augmentation method, we would like to examine its effectiveness with/without the conventional front-end augmentation method.

5.2.1. Results without front-end augmentation

In this section, we trained our embedding extractor using the original audio data. The augmentation is only applied to the embeddings for training back-end PLDA. The results are shown in the upper part of Table 2. For the PLDA augmentation, "Manual" means that the conventional method to extract noisy embeddings from the augmented audios [4]. "NDM" denotes our proposed method with Gaussian as the noise distribution, while "Combine" denotes pooling both the embeddings generated by Manual and NDM (half from each). In all augmentation methods, the clean embeddings are always combined with the augmented ones, and the total amount of the pooled embeddings are kept the same for a fair comparison.

Results show that both the manual and the proposed NDM based back-end augmentation methods could achieve noticeable performance improvement compared to the baseline system with no PLDA augmentation. NDM exceeds the manual augmentation method on all test cases when using EER as the evaluation metric. Notably, when we combine the embeddings generated by two augmentation methods, the performance could be further enhanced in most cases.

5.2.2. Results with front-end augmentation

Here, we trained our embedding extractors using augmented data by manually adding noises following the Kaldi recipe [22]. The same back-end augmentation methods as the ones in Sec.5.2.1 are used. Results are shown at the bottom part of Table 2, where we can find all the systems outperform their counterpart with no front-end augmentation. Therefore, the augmentations on the front-end extractor and the back-end PLDA are both important and they can complement each other. The performance improvement obtained by individual stages could be further boosted by combining the two methods. Based on the augmented extractor, our proposed NDM still outperforms the manual method, and the combined mode leads to further improvement, which is consistent with the findings in Sec.5.2.1.

Table 2: *Performance comparison of different data augmentation methods for different models on SRE16 (with PLDA adaptation). The best and second best results are marked as* ***bold*** *or* underlined

Augmentation		TDNN				Resnet			
Extractor	PLDA	Tagalog		Cantonese		Tagalog		Cantonese	
		EER(%)	minCprimary	EER(%)	minCprimary	EER(%)	minCprimary	EER(%)	minCprimary
No	No	15.34	0.8612	5.81	0.5028	13.26	0.8486	4.21	0.3950
	Manual	13.87	**0.8276**	5.08	0.4729	12.14	**0.8082**	3.81	**0.3682**
	NDM	13.48	0.8581	4.75	**0.4476**	12.05	0.8654	3.60	0.3712
	Combine	**13.02**	0.8581	**4.53**	0.4491	**11.86**	0.8741	**3.57**	0.3764
Manual	No	13.35	0.7983	4.62	0.3992	10.74	0.7729	3.38	0.3429
	Manual	12.87	**0.7729**	4.14	0.3791	10.38	**0.7819**	3.29	0.3336
	NDM	11.86	0.7866	3.65	0.3516	10.25	0.8071	2.92	0.3171
	Combine	**11.04**	0.7874	**3.44**	**0.3452**	**9.89**	0.8047	**2.80**	**0.3127**

5.3. Evaluation on Data Sizes for Distribution Estimation

As mentioned in Sec.3.2, it would be more appealing if we could use a small amount of data for the NDM estimation. We chose the best system, r-vector with front-end augmentation, for illustration. Results for NDM with different amounts of training data are shown in Fig.2.

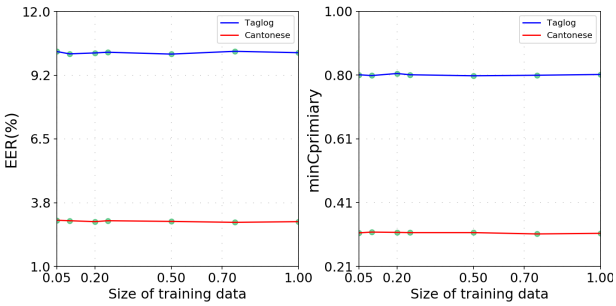

Figure 2: *Performance comparison of different training data sizes used in the NDM estimation. Green points denote the ratio of the whole training data used in our experiments, corresponding to* $0.05, 0.1, 0.2, 0.25, 0.5, 0.75, 1.0$ *on the x-axis.*

The results shown in Figure 2 reveal that *the NDM estimation is robust to the size of the training data, which means we could only prepare a small amount of training data to estimate a good noise distribution, which could save both time and disk.*

5.4. Visualization of the Embeddings Generated by NDM

In order to better understand this noisy distribution matching method, real "noisy" embedding samples and generated "noisy" embedding samples using our proposed method are visualized using t-SNE [24] and plotted in Figure 3.

The illustration shows that the generated "noisy" embeddings preserve the speaker identity to a good extent, which is essential for the supervised trained PLDA. Besides, the real "noisy" embedding samples and generated "noisy" embedding samples follow a similar distribution, which shows that the shift caused by noises added to the raw audios could be effectively captured by our newly proposed NDM method.

6. Conclusions

In this paper, we proposed a novel back-end embedding augmentation method for embedding based speaker verification, which is termed as *Noise Distribution Matching (NDM)*. Instead of adding noises to the raw audios and then extract augmented embeddings, NDM first estimates a Gaussian distribution to model the difference between the clean and original augmented noisy speaker embeddings, and then generates new noisy embeddings by adding the noises sampled from the estimated distribution to the clean speaker embeddings. Experiments on SRE16 show that the novel NDM exhibits better results than the conventional manual back-end augmentation methods. Our best system achieves EERs of 9.89% and 2.80% on the Tagalog and Cantonese evaluation sets, respectively. Furthermore, we show that even with a small set of training data, NDM could still be estimated accurately and maintain a good performance, which can save both time and disk.

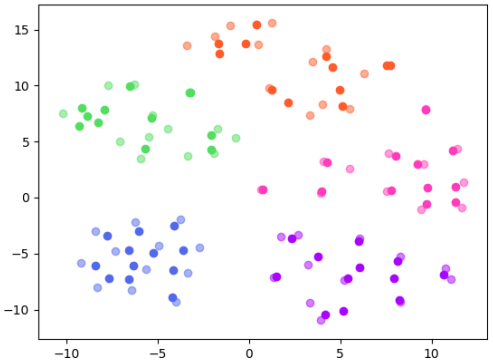

Figure 3: *T-SNE visualization of "noisy" embeddings generated by the manual mode (darker point) and NDM (lighter point) for* 5 *speakers. Points in the same color are from the same speaker.*

7. Acknowledgements

This work was supported by the China NSFC projects (No. 62071288 and No. U1736202) and the innovation project from CAS (No. 2014DP173025). Experiments have been carried out on the PI supercomputers at Shanghai Jiao Tong University.

8. References

[1] 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), 2014*. IEEE, 2014, pp. 4052–4056.

[2] S. Wang, Y. Qian, and K. Yu, "What does the speaker embedding encode?" *Proc. Interspeech 2017*, pp. 1497–1501, 2017.

[3] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-

dependent speaker verification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," 04 2018, pp. 5329–5333.

[5] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[6] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017*. IEEE, 2017, pp. 5430–5434.

[7] Z. Huang, S. Wang, and Y. Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*. IEEE, 2018.

[8] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[9] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3623–3627, 2018.

[10] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," *arXiv preprint arXiv:1906.07317*, 2019.

[11] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *arXiv preprint arXiv:1904.03479*, 2019.

[12] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

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

[14] S. Wang, Z. Huang, Y. Qian, and K. Yu, "Discriminative neural embedding learning for short-duration text-independent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1686–1696, 2019.

[15] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788 – 798, 06 2011.

[16] Y. Yang, S. Wang, M. Sun, Y. Qian, and K. Yu, "Generative adversarial networks based x-vector augmentation for robust probabilistic linear discriminant analysis in speaker verification," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 205–209.

[17] Z. Wu, S. Wang, Y. Qian, and K. Yu, "Data augmentation using variational autoencoder for embedding based speaker verification," *Proc. Interspeech 2019*, pp. 1163–1167, 2019.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[20] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation," 2017.

[21] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," 08 2017, pp. 999–1003.

[22] Kaldi. Sre16 speaker verification recipe in kaldi. [Online]. Available: https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.

[24] 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.