Augmentation-Agnostic Regularization for Unsupervised Contrastive Learning with Its Application to Speaker Verification

Nakamasa Inoue, Tsubasa Maruyama, Keita Goto
Tokyo Institute of Technology, Japan
E-mail: inoue@c.titech.ac.jp, maruyama.t.at@m.titech.ac.jp, goto.k.al@m.titech.ac.jp

Abstract—This paper presents a regularization method for unsupervised contrastive learning and its application to speaker verification. The proposed method, called Augmentation-Agnostic Regularization, enhances the training of speaker embeddings in an adversarial manner. Our main idea is to use an augmentation seed classifier, which learns to classify the randomization seeds used in data augmentation methods, and to train an embedding network with a regularization term to fool the classifier. This method prevents the characteristics of the augmentation procedure from remaining in the embeddings, facilitating the extraction of speaker characteristics. In experiments, we demonstrate the effectiveness of the proposed regularization in two challenging data-deficient conditions, namely a small-sample training condition and a short-utterance testing condition, and show performance improvements over the conventional augmented adversarial training method. The unsupervised model trained with our method achieved comparable performance with the supervised x-vector baseline model.
Index Terms: Contrastive Learning, Unsupervised Learning, Text-Independent Speaker Verification, Data Augmentation.

I. Introduction

Unsupervised representation learning, which aims to train an embedding network without using ground-truth labels, has attracted increasing attention from researchers due to its wide range of application. For speaker verification and identification, recent studies have shown that contrastive learning approaches are effective for learning speaker embeddings without supervision. The basic idea of contrastive learning is to minimize the distance between the embeddings of two augmented utterances obtained from a single utterance and maximize the distance between the embeddings of different utterances. This is implemented in loss functions, such as augmented adversarial training (AAT) loss [1], generalized contrastive loss (GCL) [2], and momentum contrast (MoCo) loss [4], [3], for training embedding networks.

A limitation of contrastive learning is the requirement of a large amount of training data. The cost of data collection for unsupervised learning is generally smaller than that for supervised learning because manual annotations are not needed. However, training data collected from the Internet may have unintended biases such as those regarding gender, race, social status, and socio-economic status. This is a problem when deploying and training networks because biases in large-scale data are difficult to analyze and remove. This motivated us to analyze unsupervised contrastive learning in two challenging conditions, namely a small-sample training condition and a short-utterance testing condition. This research will facilitate the development of data-efficient learning systems with small-scale protected or private data.

A straightforward approach for fully leveraging small and short-utterance data is to enhance the data augmentation process. However, if we simply increase the number of augmented samples, information about the augmentation process may remain in the embeddings after training. For example, if additive noise is used for data augmentation, noise characteristics may remain in the learned embeddings even if we want the embeddings to be agnostic about noise.

The proposed method, called Augmentation-Agnostic Regularization, overcomes this problem by using an augmentation seed classifier and by training a network with a regularization loss to fool the classifier. This is a form of adversarial training and can be viewed as an extension of Augmentation Adversarial Training [1]. In experiments, we demonstrate the effectiveness of the proposed regularization method on the VoxCeleb1 dataset [5] and the SdSVC 2021 dataset [6], [7]. This study makes the following contributions.

1) We propose a regularization method called Augmentation-Agnostic Regularization for contrastive learning.
2) We conduct experiments in two challenging data-deficient conditions, namely a small-sample training condition and a short-utterance testing condition, and demonstrate that the proposed method improves speaker verification performance.

II. Related Work

A. Speaker Embeddings

This paper focuses on the learning of speaker embeddings for text-independent speaker verification. Statistical modeling methods are commonly used to capture speaker characteristics. For example, i-vectors [8] use a mixture of Gaussians to estimate the distributions of audio features such as mel-frequency cepstral coefficients

and are often used with probabilistic linear discriminant analysis. x-vectors [9] use a time-delay neural network to extract embeddings. Many architectures have been proposed, including Thin-ResNet [10], Dense-TDNN [11], and ECAPA-TDNN [12]. For a large-scale training dataset such as VoxCeleb2 [5], these network-based embeddings outperform i-vectors. However, with large-scale training, it is not always easy to avoid unintended biases such those regarding gender and race. This motivated us to analyze the trade-off between verification performance and data efficiency and to explore data-efficient learning.

B. Unsupervised Contrastive Learning

Unsupervised learning, which aims to train a model without using ground-truth labels on training samples, has been proven to be effective for learning representations. In particular, approaches that use contrastive learning mechanisms are promising. For example, MoCo [3] and SimCLR [13] use a contrastive loss function in which many data augmentation methods are applied to improve robustness against perturbations such as noise. They achieve image recognition performance comparable to that of supervised learning. Contrastive learning methods have been proposed for speaker verification. Generalized contrastive loss [2] for speaker verification works without supervision or with semi-supervision. AAT [1] uses a contrastive loss that separates speaker information from channel information. MoCo and SimCLR have been applied to speaker verification [4]. With these methods, MUSAN noise [14] and room impulse response (RIR) data [15], [1] are often used for data augmentation. For short-utterance speaker verification, most methods rely on supervised learning [6]. Examples include adversarial training using generative adversarial networks [20], [19], teacher-student learning [18], meta learning on imbalanced-length utterances [21], and extended probabilistic linear discriminant analysis models [16], [17]. An evaluation of unsupervised learning in a short-utterance testing condition would be interesting as a next step.

III. Proposed Method

This section presents the proposed method, called Augmentation-Agnostic Regularization. An overview of the learning framework is given followed by the details of the method. Our main idea is to use an augmentation seed classifier and to train an embedding network with a regularization term to fool the classifier. This is a form of adversarial training and can be viewed as an extension of Augmentation Adversarial Training [1].

A. Overview

Let $\mathcal{X}$ be a set of unlabeled samples (utterances) for training. The goal is to train an embedding network E, which maps samples $x \in \mathcal{X}$ into a real-valued vector space as $z = E(x) \in \mathbb{R}^d$. Figure 1 shows an overview of the framework, which uses the sum of two losses for training:

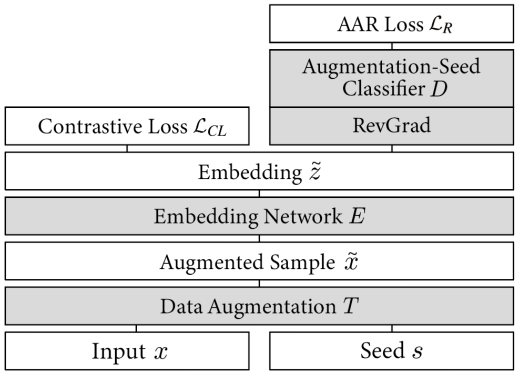

Fig. 1. Framework overview.

$$\mathcal{L} = \mathcal{L}_{\mathrm{CL}} + \lambda \mathcal{L}_{\mathrm{R}}, \tag{1}$$

where $\mathcal{L}_{\mathrm{CL}}$ is the contrastive loss, $\mathcal{L}_{\mathrm{R}}$ is the augmentation-agnostic regularization (AAR) loss, and λ is a weighting hyperparameter. We use the unsupervised extension [1], [2] of angular prototypical loss [24], [25] for $\mathcal{L}_{\mathrm{CL}}$. The details of the regularization are described below.

The AAR loss is defined with an augmentation seed classifier, which is placed at the top of the embedding network. Here, T denotes the augmentation function, which generates an augmented sample $\tilde{x}$ by

$$\tilde{x} = T(x, s), \tag{2}$$

where x is an input and s is the seed used for randomization. For simplicity, we assume that the number of seeds is finite. The number of seeds is denoted by S, i.e., $s \in \{1, 2, \cdots, S\}$. The augmentation seed classifier D is a discriminator that predicts the seed s from the augmented sample $\tilde{x}$ as follows:

$$\hat{s} = D(\tilde{x}), \tag{3}$$

where $\hat{s}$ is the predicted seed. In the training phase, the augmentation seed classifier D learns to classify seeds and the embedding network E learns to fool D. This adversarial training strategy is implemented with a gradient reversal layer (RevGrad) [26], as shown in Figure 1.

B. Metric Learning

For training the augmentation seed classifier, the standard cross-entropy loss is not a good choice in practice because S is typically very large. The proposed framework thus uses metric learning.

The basic idea of metric learning is to minimize the anchor-positive distance $d(x_a, x_p)$ and maximize the anchor-negative distance $d(x_a, x_n)$, where x_a is a randomly sampled anchor, x_p is a positive sample that has the same ground-truth label as that of the anchor, and

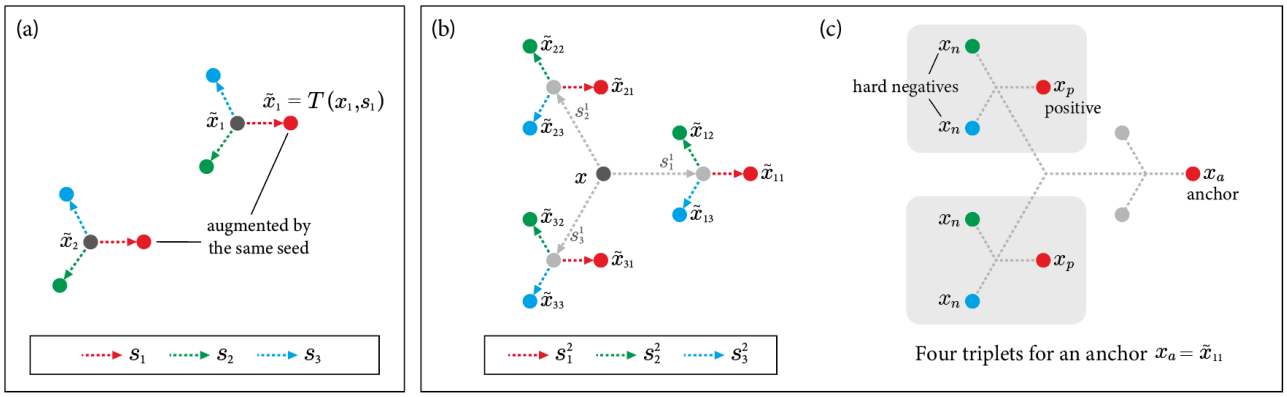

Fig. 2. Data augmentation process and triplets for computing augmentation-agnostic regularization loss. (a) Six augmented samples obtained for two examples, x_1 and x_2, with three seeds, namely s_1, s_2, and s_3. (b) Two-step data augmentation for a single sample x. Nine augmented samples $\tilde{x}_{ij}$ are obtained with three-by-three seeds. (c) Four triples for x_{11} with positive samples and hard negative samples.

x_n is a negative sample that has a different ground-truth label from that of the anchor. These distances are often computed over triplets (x_a, x_p, x_n). A naive method for obtaining triples for training the augmentation seed classifier is to repeat the following three steps.

1) Draw an anchor x_a from X, where $X = \{\tilde{x} : x \in \mathcal{X}\}$ is a set of augmented training samples generated with randomly selected seeds.
2) Draw a positive sample x_p from
$$X_p = \{\tilde{x} \in X : \ell(\tilde{x}) = \ell(x_a)\} \setminus \{x_a\}, \quad (4)$$
where $\ell(\tilde{x}) = s$ is the seed used in data augmentation.
3) Draw a negative sample x_n from
$$X_n = \{\tilde{x} \in X : \ell(\tilde{x}) \neq \ell(x_a)\}. \quad (5)$$

Note that $\ell(\cdot)$ indicates ground-truth labels, and seeds are used as labels. Figure 2a shows an example where two samples, x_1 and x_2, are augmented with three seeds, namely s_1, s_2, and s_3. In this case, D learns to classify these three seeds. However, this problem is difficult to solve if the two original samples x_1 and x_2 are far from each other.

C. Local Triplets with Hard Negatives

To overcome the above problem, we locally optimize the augmentation seed classifier by generating triplets around each training sample. Specifically, we apply two-step augmentation to each sample x as follows to make triplets around it:

$$\tilde{x}_{ij} = T_2(T_1(x, s_i^1), s_j^2), \quad (6)$$

where T_1 and T_2 are augmentation functions and s_i^1 and s_j^2 are seeds for $i = 1, 2, \cdots, N_1$ and $j = 1, 2, \cdots, N_2$, respectively. This means that we obtain a set of $N_1 \times N_2$ augmented samples around x:

$$X = \{\tilde{x}_{ij} : 1 \leq i \leq N_1, 1 \leq j \leq N_2\}. \quad (7)$$

Figure 2b shows an example where $N_1 = 3$ and $N_2 = 3$.

From the obtained set of augmented samples X, triples are made using the following two steps.

1) Make a set of anchor-positive pairs by
$$S_P = \bigcup_{x_a \in X} \{(x_a, x_p) : x_p \in X_p\}, \quad (8)$$
where X_p is a set of positive samples given by
$$X_p = \{\tilde{x} \in X : \ell_2(\tilde{x}) = \ell_2(x_a)\} \setminus \{x_a\}, \quad (9)$$
and $\ell_2(\tilde{x}) = s^2$ is the second seed used in the two-step augmentation.
2) Make a set of triplets by
$$S_T = \bigcup_{(x_a, x_p) \in S_P} \{(x_a, x_p, x_n) : x_n \in X_n\}, \quad (10)$$
where X_n is a set of negative samples given by
$$X_n = \{\tilde{x} \in X : \ell_2(\tilde{x}) \neq \ell_2(x_a), \ell_1(\tilde{x}) = \ell_1(x_p)\}, \quad (11)$$
and $\ell_1(x) = s^1$ is the first seed used in the two-step augmentation.

In the second step, Eq. (11) introduces the restriction $\ell_1(x) = \ell_1(x_p)$ to select hard negative samples. Figure 2c shows an example with $\tilde{x}_{11}$ as an anchor. In this case, we obtain two anchor-positive pairs (colored red) in the first step. Each pair has two hard negatives (colored green and blue).

Finally, we obtain $|S_T| = N_1 N_2 (N_1 - 1)(N_2 - 1)$ triplets from a single sample x. This rich number of locally generated triplets enhances contrastive learning even if the number of training samples is small.

D. Definition of Loss

Given a mini-batch B, the AAR loss is defined over generated triplets as follows:

$$\mathcal{L}_{\mathrm{R}} = \frac{1}{|B||S_T|} \sum_{x \in B} \sum_{\tau \in S_T} L(x_a, x_p, x_n), \tag{12}$$

where $\tau = (x_a, x_p, x_n)$ is a triplet and $L(x_a, x_p, x_n)$ is the metric learning loss. Notably, if the loss is symmetric with respect to the anchor-positive pairs, i.e., if $L(x_a, x_p, x_n) = L(x_p, x_a, x_n)$, half of the triplets are redundant, and thus we only need $|S_T|/2$ triplets to compute the loss. In the following, we show two definitions for L.

Triplet loss [22]. This loss is a standard metric leaning loss defined over triplets with a margin hyperparameter m. It is given by

$$L(x_a, x_p, x_n) = \max(0, d(h_a, h_p) - d(h_a, h_n) + m), \tag{13}$$

where h_a, h_p, and $h_n \in \mathbb{R}^{d'}$ are the output of the augmentation seed classifier corresponding to x_a, x_p, and x_n, respectively, and d is the Euclidian distance.

AAT loss. This loss is used in AAT [1]. It is given by

$$L(x_a, x_p, x_n) = \sum_{(h,y) \in H} \log \frac{\exp(h[y])}{\sum_{y'=1}^{2} \exp(h[y'])}, \tag{14}$$

where $h \in \mathbb{R}^2$ is the output of the augmentation seed classifier. Here, the augmented seed classifier D is modified to accept concatenated embeddings $v_p = E(x_a) + E(x_p)$ and $v_n = E(x_a) + E(x_n)$ as proposed in [1], where $+$ indicates concatenation. H attaches labels 1 and 2 to v_p and v_n, respectively, i.e., $H = \{(D(v_p), 1), (D(v_n), 2)\}$.

IV. Experiments

A. Settings

We conducted experiments of speaker verification in two challenging data-deficient conditions, namely a small-sample training condition and a short-utterance testing condition. For training, we used the VoxCeleb1 Dev set, which consists of 148,642 utterances by 1,211 speakers. To evaluate the performance of contrastive learning with a small amount of training data, we varied the amount of training data from 1% to 100% by random sampling. The evaluation sets were VoxCeleb1 Test and SdSVC 2021 Dev, which consist of 37,611 and 7,071 verification pairs, respectively. We also report results of our SdSVC 2021 submission. The evaluation measures are the equal error rate (EER) and the minimum detection cost function (MinDCF).

B. Implementation Details

All models were implemented in PyTorch [23]. Specifically, we followed the settings in [1] with the official implementation [1] The details are as follows. The backbone network is ResNetSE34L. The contrastive loss $\mathcal{L}_{\mathrm{CL}}$ in Eq. (1) is the unsupervised version of the angular prototypical loss [24], [25]. The data augmentation process consists of random cropping (segment length = 240) and the noise or RIR augmentation [1], where MUSAN noise [14] or RIR is applied with the default hyperparameters. For the proposed AAR loss, random cropping was used as the first augmentation T_1 and the noise or RIR augmentation was used as the second augmentation T_2. The number of random seeds in Eq. (7) was set to $N_1 = 3$ and $N_2 = 3$. For each utterance, $N_1 + N_2$ seeds were randomly chosen by the numpy.randint function, and then these seeds were used to generate $N_1 \times N_2$ augmented samples, as in Eq. (6). The augmentation seed classifier was a network that consisted of two blocks of a fully-connected layer (hidden size = 512), ReLU activation, and batch normalization. The output dimension was 64 for triplet loss (with margin $m = 1.0$) and 2 for AAT loss for its binary classification. For training, the ADAM optimizer was used for 150 epochs. The learning rate was 0.001; it decayed by 0.95 every 5 epochs. λ was set to 5.0. The batch size was set to $B = 150$. A single NVIDIA P100 GPU was used for training.

[1] https://github.com/joonson/voxceleb_unsupervised

TABLE I
Unsupervised learning performance. EER (%) and MinDCF are shown for the VoxCeleb1 Test set and the SdSVC 2021 Dev set. The VoxCeleb1 Dev set was used for training. Ours (AAT+Triplet) is the average late fusion of Ours (AAT) and Ours (Triplet).

Method	VoxCeleb1 Test		SdSVC 2021 Dev	
	EER	MinDCF	EER	MinDCF
Baseline	10.61	0.503	12.81	0.585
AAT [1]	10.15	0.481	11.87	0.585
Ours (Triplet)	8.65	0.407	10.76	0.536
Ours (AAT)	8.90	0.428	11.31	0.539
Ours (Triplet+AAT)	8.63	0.418	10.42	0.518

TABLE II
Effect of number of local triplets on performance. EER is shown for the VoxCeleb1 Test set. N_1 and N_2 are the numbers of seeds for the first and second augmentation functions, respectively.

# of triplets per utterance	$N_1 \times N_2$	EER (%)
4	2×2	9.55
6	2×3	9.16
6	3×2	8.95
8	2×4	9.05
8	4×2	8.93
9	3×3	8.86

C. Results

Table 1 shows the unsupervised learning performance for four methods, namely the baseline without using regularization loss, the conventional AAT loss [1], and the proposed AAR loss with triplet loss and AAT loss (see Sec. 3.4). Our method outperforms the conventional method on both VoxCeleb1 Test and SdSVC 2021 Dev in terms of EER and MinDCF. This confirms the effectiveness of AAR for contrastive learning. There was no significant difference between the use of triplet loss and

TABLE III
Small-sample training performance. Unlabeled utterances randomly sampled from the VoxCeleb1 Dev set were used for training (1% of data includes 1.5k utterances). The i-vector result is from [1].

	Amount of Training Data					
	1%	2%	5%	10%	20%	50%
AAT	20.74	15.45	13.91	12.85	11.68	10.89
Ours	16.57	14.35	12.89	11.95	10.44	9.84
i-vector	15.28					

TABLE IV
SdSVC 2021 submission results. EER is shown for the SdSVC 2021 Test set. We use the VoxCeleb1 Dev set for training, and the non-speech subset of MUSAN noises and RIR for augmentation, following the evaluation plan of SdSVC 2021. The x-vector baseline is trained on VoxCeleb1 and VoxCeleb2, and the full set of MUSAN and RIR is used for augmentation.

Method	Voxceleb1	SdSVC 2021	
	Test	Dev	Test
Ours (Triplet)	9.16	11.17	10.96
Ours (AAT)	9.09	11.17	11.06
Ours (Triplet + AAT)	8.86	10.95	10.49
x-vector (supervised)	-	8.11	10.65

AAT loss. This means that the proposed triplet generation algorithm is mainly responsible for the performance improvement.

Table 2 shows the effect of the number of local triplets on performance. EER decreases with increasing number of local triplets $N_1 \times N_2$. This supports our assumption that the local optimization of the metric space with augmentation helps contrastive learning. Table 3 shows the results obtained in the small-sample training condition, where a subset of the VoxCeleb1 Dev set was used for training. The results confirm that AAR improves performance regardless of the number of training samples. Even with 2% of the utterances from the VoxCeleb1 Dev set, the proposed method outperforms the i-vector baseline [8], [1]. The proposed method will thus facilitate the development of learning systems that use small-scale protected data for training.

Table 4 summarizes the results of our SdSVC 2021 submission. We used the VoxCeleb1 Dev set for training and the non-speech subset of MUSAN noises (omitted music and noise) and RIR for augmentation, because augmentation using speech data is prohibited. If we compare results in Table 1 and Table 4, this change caused a small performance drop. Finally, on the leaderboard, our method achieved comparable performance with the official supervised x-vector baseline, which is trained on the VoxCeleb1 and VoxCeleb2 Dev sets. As future work, analyzing and improving the datasets for data augmentation would be interesting as a next step.

V. Conclusion

This paper proposed a method called Augmentation-Agnostic Regularization for unsupervised contrastive learning that uses an augmentation seed classifier for adversarial training. Experiments on VoxCeleb1 and SdSVC 2021 datasets showed that the proposed method improves speaker verification. We hope that this work will facilitate the development of data-efficient learning systems with small-scale protected or private data.

VI. Acknowledgement

This work was supported in part by the Japan Science and Technology Agency, ACT-X Grant JPMJAX1905.

References

[1] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. S. Chung, Augmentation adversarial training for unsupervised speaker recognition, Proc. NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing, 2020.
[2] N. Inoue and K.Goto, Semi-supervised contrastive learning with generalized contrastive loss and its application to speaker recognition, Proc. APSIPA, 2020.
[3] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, Momentum contrast for unsupervised visual representation learning, Proc. CVPR, 2020.
[4] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, Self-supervised text-independent speaker verification using prototypical momentum contrastive learning, Proc. ICASSP, 2021.
[5] A. Nagrani, J. S. Chung, W. Xie, A. Zisserman Voxceleb: Large-scale speaker verification in the wild, Computer Science and Language, vol. 60, 2020.
[6] H. Zeinali, K. A. Lee, J. Alam, L. Burget, SdSV Challenge 2020: Large-scale evaluation of short-duration speaker verification, Proc. Interspeech, 2020.
[7] H. Zeinali, L. Burget, and J. Cernocky, A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: The DeepMine database, Proc. ASRU, 2019.
[8] 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 LanguageProcessing, 19(4):788-798, 2011.
[9] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, X-vectors: Robust DNN embeddings for speaker recognition, Proc. ICASSP, 2018.
[10] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, Utterance-level aggregation for speaker recognition in the wild, Proc. ICASSP, 2019.
[11] Y. Q. Yu, W. J. Li, Densely connected time delay neural network for speaker verification, Proc. Interspeech, 2020.
[12] B. Desplanques, J. Thienpondt, K. Demuynck, ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification, Proc. Interspeech, 2020.
[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, A simple framework for contrastive learning of visual representations Proc. ICML, 2020.
[14] D. Snyder, G. Chen, and D. Povey, Musan: A music, speech, and noise corpus, arXiv preprint arXiv:1510.08484, 2015.
[15] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, A study on data augmentation of reverberant speech for robust speech recognition, Proc. ICASSP, 2017.
[16] A. Lozano-Diez, A. Silnova, B. Pulugundla, J. Rohdin, K. Veselu, L. Burget, O. Plchot, O. Glembek, O. Novotny, P. Matejka, BUT text-dependent speaker verification system for SdSV Challenge 2020, Proc. Interspeech, 2020.
[17] Z. Chen and Y. Lin, Improving x-vector and PLDA for text-dependent Speaker Verification Proc. Interspeech, 2020.
[18] J. W. Jung, H. S. Heo, H. J. Shim, and H. J. Yu, Short utterance compensation in speaker verification via cosine-based teacher-student learning of speaker embeddings, Proc. ACRU, 2019.
[19] K. Liu, H. Zhou, Text-independent speaker verification with adversarial learning on short utterances, Proc. ICASSP, 2020.

[20] J. Zhang, N. Inoue, and K. Shinoda, I-vector transformation using conditional generative adversarial networks for short utterance speaker verification, Proc. Interspeech, 2018.
[21] S. M. Kye, Y. Jung, H. B. Lee, S. J. Hwang, H. Kim, Meta-learning for short utterance speaker recognition with imbalance length pairs, Proc. Interspeech, 2020.
[22] E. Hoffer and N. Ailon, Deep metric learning using triplet network, Proc. SIMBAD, pp. 84–92, 2015.
[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, Proc. NeurIPS, 2019.
[24] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B. J. Lee, and I. Han, In defence of metric learning for speaker recognition, Proc. Interspeech, 2020.
[25] J. Snell, K. Swersky, and R. Zemel, Prototypical networks for few-shot learning, Proc. NeurIPS, 2017.
[26] Y. Ganin and V. Lempitsky, Unsupervised domain adaptation by backpropagation, Proc. ICML, 2015.
[27] A. Povey, Ghoshal, G. Boulianne, L. Burget, O. Glembek,N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarzet, The Kaldi speech recognition toolkit, Proc. ASRU, 2011.
[28] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, Proc. CVPR, 2016.