Investigation on instance mixup regularization strategies for self-supervised speaker representation learning

Woo Hyun Kang, Jahangir Alam, Abderrahim Fathan

Computer Research Institute of Montreal (CRIM)

woohyun.kang,jahangir.alam,abderrahim.fathan@crim.ca

Abstract

Over the recent years, various self-supervised contrastive embedding learning methods for deep speaker verification were proposed. The performance of the self-supervised contrastive learning framework highly depends on the data augmentation technique, but due to the sensitive nature of speaker information within the speech signal, most speaker embedding training relies on simple augmentations such as additive noise or simulated reverberation. Thus, while the conventional self-supervised speaker embedding systems can yield minimum within-utterance variability, their capability to generalize to out-of-set utterance is limited. In order to alleviate this problem, we investigate the utilization of the instance mix (i-mix) regularization for training a self-supervised speaker embedding system. Moreover, we propose a new mixup strategy that applies i-mix on the latent space, instead of the raw acoustic feature domain. The proposed method was evaluated on the VoxCeleb1 dataset and showed noticeable performance improvement over the standard self-supervised embedding method.

Introduction

Speaker verification is the task of verifying the claimed speaker identity based on the given speech samples and has become a key technology for personal authentication in many commercial, forensics and law enforcement applications (Hansen and Hasan 2015). Commonly, utterance-level fixed-dimensional vectors (i.e. embedding vectors) are extracted from the enrollment and test speech samples and then fed into a scoring algorithm (e.g., cosine distance, probabilistic linear discriminant analysis) to measure their similarity or likelihood of being spoken by the same speaker. Classically, the i-vector framework has been one of the most dominant approaches for speech embedding (Dehak et al. 2011), (Kenny 2012). The widespread popularity of the i-vector framework in the speaker verification community can be attributed to its ability to summarize the distributive pattern of the speech with a relatively small amount of training data in an unsupervised manner.

In recent years, various methods have been proposed utilizing deep learning architectures for extracting embedding vectors and have shown better performance than the i-vector framework when a large amount of training data with enough diversity is available (Snyder et al. 2018). In (Snyder et al. 2018, 2017), a speaker recognition model consisting of a time-delay neural network (TDNN)-based frame-level network and a segment-level network was trained and the hidden layer activation of the segment-level network, denoted as x-vector, was extracted as the embedding vector. In (Desplanques, Thienpondt, and Demuynck 2020), an ECAPA-TDNN architecture was proposed, which has shown state-of-the-art performance by introducing residual and squeeze-and-excitation (SE) components to the widely used TDNN-based embedding system. Although the deep embedding methods have outperformed the i-vector framework in various speaker verification benchmarks, since most of these models are trained in a fully supervised fashion, they require a large amount of speaker labeled dataset for optimization.

To overcome this limitation, a number of self-supervised embedding learning methods for deep speaker verification were proposed over the past couple of years (Huh et al. 2020; Mun et al. 2020; Ding, He, and Wan 2020; Zhang, Zou, and Wang 2021). Many of these researches employ the contrastive learning scheme for optimization, where the embeddings from the same utterance (positive pairs) are trained to be close to each other while pushing away embeddings from different utterances (negative pairs). In order to effectively capture the utterance-dependent variability into the embedding, different types of augmentations are usually applied to the positive pair utterances. However, since speaker-dependent information can be easily distorted under severe augmentation, most speaker embedding training relies on simple augmentations such as noise/reverberation mixing (Huh et al. 2020) or frequency-/time-masking (Ding, He, and Wan 2020). Due to this constraint, while the augmentation can help minimize the within-utterance variability, its capability to generalize to out-of-set utterances is limited.

One way to mitigate this is to employ mixup regularization technique, which creates new data by linearly interpolating two training samples (Zhang et al. 2017). Despite its simple formulation, mixup have shown promising performance in various tasks including image classification (Zhang et al. 2017), supervised speaker recognition (Zhu, Ko, and Mak 2019), and anti-spoofing (Tomilov et al. 2021). However, since the mixup strategy requires interpolation on labels, it cannot be directly applied to self-supervised learning scenarios. Therefore, the instance mix (i-mix) regular-

ization was proposed, which expands the mixup formulation to apply interpolation on the pseudo-labels (Lee et al. 2021). The i-mix framework has shown potential in not only image classification (Lee et al. 2021), but also in some audio tasks (e.g., sound classification) (Niizumi et al. 2021).

In light of this, we explore the adaptation of i-mix augmentation scheme to the self-supervised embedding learning for speaker verification. Unlike the conventional augmentations for self-supervised speaker verification, which simply augment the waveform or spectrogram of the speech by introducing adversaries via masking or additive noise, the i-mix scheme aims to create a synthetic training sample with a new target identity by interpolating different samples along with their utterance identity. Therefore the i-mix strategy can efficiently enhance the generalization of the self-supervised speaker representation learning process, which will enable the network to produce robust embeddings which can perform well on verifying out-of-set speakers.

Moreover, we propose a new mixup strategy that applies i-mix on the latent space instead of the raw data domain. Unlike the previous attempts in applying mixup on the hidden representations trained jointly with the downstream network (Verma et al. 2019; Chen et al. 2020), we aim to extract the latent representations with no context of the speaker. This way, applying i-mix on the latent representation may introduce not only new speakers, but also new non-speaker variabilities (e.g., channel, environment). In order to apply i-mix on the latent space of the speech, we incorporate a variational autoencoder (VAE) encoder (Kingma and Welling 2014) to extract the latent variable of the given acoustic features. The mixed latent variable is fed into the VAE decoder to generate a synthetic sample, which will have different pattern from the samples generated via the standard i-mix strategy.

The contributions of this paper are as follows:

- We incorporate the i-mix framework to the self-supervised angular prototypical objective function for speaker embedding learning.
- We propose a latent space i-mix strategy (l-mix), which performs i-mix on the latent space of the speech.
- We compare the speaker verification performances of systems trained with different i-mix and l-mix hyperparameters.

Related work and Baseline model

Baseline self-supervised representation learning

Most recent self-supervised embedding learning methods use contrastive loss to produce embedding vectors with maximum utterance discriminability (Huh et al. 2020; Mun et al. 2020; Ding, He, and Wan 2020; Zhang, Zou, and Wang 2021). As shown in Figure 1, in the self-supervised contrastive embedding learning framework, two samples are generated per utterance by applying different augmentations. The augmented samples are then passed through an embedding network to generate embedding vectors. The network is optimized via contrastive learning (e.g., prototypical loss), which minimizes the distance between the embeddings from the same utterance, while maximizing the distance between different utterance embeddings.

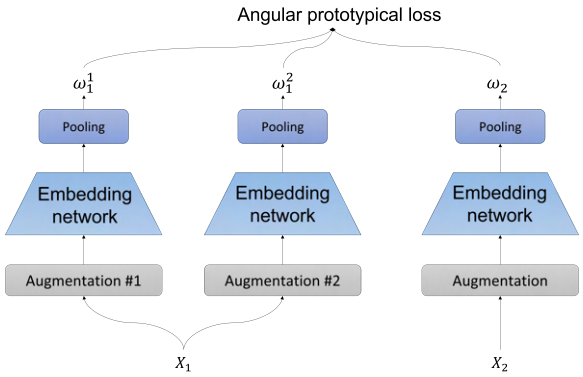


Figure 1: The general framework for the self-supervised contrastive speaker embedding learning.

Embedding network In our research, we have experimented with the ECAPA-TDNN encoder (Desplanques, Thienpondt, and Demuynck 2020), an architecture that achieved state-of-the-art performance in text-independent speaker recognition. The ECAPA-TDNN uses squeeze-and-excitation along with channel- and context-dependent statistics pooling and multi-layer aggregation. The embedding network takes the acoustic feature as input and outputs the frame-level representations. The network outputs are aggregated via self-attention pooling, which computes the weighted average of the frame-level representations to obtain an utterance-level fixed dimensional embedding vector.

Angular prototypical objective In order to train the embedding network with no speaker labels, we have used an utterance-discriminative contrastive loss, more specifically the angular prototypical loss function (Zhang, Zou, and Wang 2021), (Huh et al. 2020). Given a batch of prototype embedding vectors ω_i^1 and query embeddings ω_i^2, where ω_i^k indicates the embedding extracted from the i^{th} utterance X_i applied with augmentation $\#k$, the angular prototypical function is defined as follows:

$$L_{AP} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{exp(cos(\omega_i^1,\omega_i^2))}{\sum_{j=1}^{N}exp(cos(\omega_i^1,\omega_j^2))}, \quad (1)$$

where cos represents the cosine similarity operation. Equation 1 can be interpreted as the cross-entropy loss which aims to maximize the similarity between the embeddings extracted from the same utterance, while minimizing the similarity between different utterance embeddings.

Speech augmentation For training the embedding network via angular prototypical objective, we have used waveform-level data augmentations including additive noise and room impulse response (RIR) simulation (Snyder et al. 2018). In addition to the waveform-level augmentations, we have also applied augmentation over the extracted Mel-frequency cepstral coefficient (MFCC) feature, denoted here

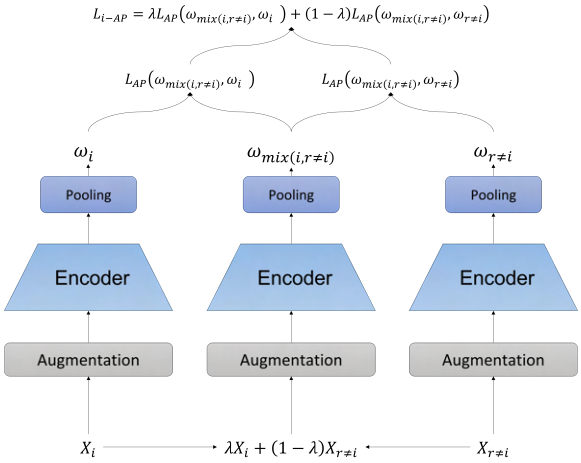


Figure 2: The general framework for the i-mix angular prototypical learning.

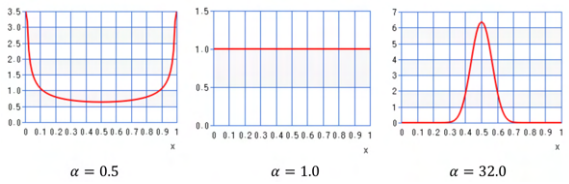


Figure 3: Beta distributions with different α value.

as cepsaugment, which is similar to specaugment scheme often used for automatic speech recognition (ASR) (Park et al. 2019). Analogous to the specaugment, in cepsaugment, a randomly selected time-cepstral bin is selected and masked before being fed into the embedding network.

Instance mix (i-mix) regularization strategy

The i-mix is a data-driven augmentation strategy for improving the generalization of the learned representation (Lee et al. 2021). For arbitrary objective function $L_{pair}(x, y)$, where x is the input sample and y is the corresponding pseudo-label, given two data instances (x_i, y_i) and (x_j, y_j), the i-mix loss is defined as follows:

$$\begin{aligned} &L_{pair}^{i-mix}((x_i, y_i), (x_j, y_j)) \\ &= L_{pair}(\lambda x_i + (1-\lambda)x_j, \lambda y_i + (1-\lambda)y_j), \end{aligned} \tag{2}$$

where $\lambda \sim Beta(\alpha, \alpha)$ is a mixing coefficient. For cross-entropy-based L_{pair}, such as prototypical loss, equation 2 can be rewritten as,

$$\begin{aligned} &L_{pair}^{i-mix}((x_i, y_i), (x_j, y_j)) \\ &= \lambda L_{pair}(x_i, y_i) + (1-\lambda) L_{pair}(x_j, y_j). \end{aligned} \tag{3}$$

i-mix angular prototypical objective

In this paper, we integrate the angular prototypical loss and the i-mix strategy for robust self-supervised embedding learning. More specifically, as depicted in Figure 2, we propose to perform i-mix on the prototype embedding vectors:

$$\begin{aligned} L_{i-AP} = &-\lambda \frac{1}{N} \sum_{i=1}^{N} \log \frac{exp(cos(\omega_{mix(i,r\neq i)}, \omega_i))}{\sum_{j=1}^{N} exp(cos(\omega_{mix(i,r\neq i)}, \omega_j))} \\ &- (1-\lambda) \frac{1}{N} \sum_{i=1}^{N} \log \frac{exp(cos(\omega_{mix(i,r\neq i)}, \omega_{r\neq i}))}{\sum_{j=1}^{N} exp(\omega_{mix(i,r\neq i)}, \omega_j))}, \end{aligned} \tag{4}$$

where $\omega_{r\neq i}$ is an embedding randomly sampled from the batch $[\omega_1, \omega_2, ..., \omega_N]$ excluding ω_i, and $\omega_{mix(i,r\neq i)}$ is an embedding extracted from mixed utterance $\lambda X_i + (1-\lambda) X_{r\neq i}$.

Training the embedding network with the i-mix angular prototypical objective L_{i-AP} can be thought of as optimizing the network on a out-of-set utterance $X_{mix} = \lambda X_i + (1-\lambda) X_{r\neq i}$, which retains utterance-dependent attributes from both X_i and $X_{r\neq i}$. Hence the resulting embedding vector can generalize well on samples that are not included in the training dataset.

Analogous to the standard i-mix described in equation 3, we also use λ randomly sampled from $Beta(\alpha, \alpha)$. As depicted in Figure 3, the shape of the $Beta(\alpha, \alpha)$ distribution varies heavily depending on the α, and resulting lambda decides the expected behavior of the utterance interpolation $\lambda X_i + (1-\lambda) X_{r\neq i}$. For example, for $\alpha < 1.0$, the beta distribution is U-shaped, thus the sampled λ is likely to have value close to 1.0 or 0.

On the other hand, using $\alpha > 1.0$ creates a bell-shaped beta distribution, which is similar to a Gaussian distribution with mean 0.5. The λ sampled from this distribution is likely to have value near 0.5, hence in the interpolation process, the two utterances will be added with similar power-level. Such overlapping speech samples are known to be challenging for the speaker recognition system, even for speakers observed during training (Tran and Tsai 2020). Therefore, using $\alpha > 1.0$ may hinder the learning capacity of the embedding networks, thus resulting in an embedding vector with insufficient speaker-dependent information.

Latent space mix (l-mix) angular prototypical objective

Although applying mixup augmentation to the raw data have proven its strength in generalization in many tasks (e.g., speech recognition, image classification), there is still room for improvement when it comes to speaker recognition. For speaker recognition, the main purpose of the mixup strategy is to let the speaker embedding system generalize to unseen speakers. However, due to the nature of speech, the samples created by mixup are likely to be very unrealistic in terms of speech production. For example, simply weighted summing two speech samples will create a speech sample with two speakers talking over each other, instead of creating a speech sample with speaker similar to both speakers. As this will introduce adversaries to the training sample, it will help make the system robust. But in terms of learning the manifold of the speech distribution, the effect of standard mixup strategy could be limited.

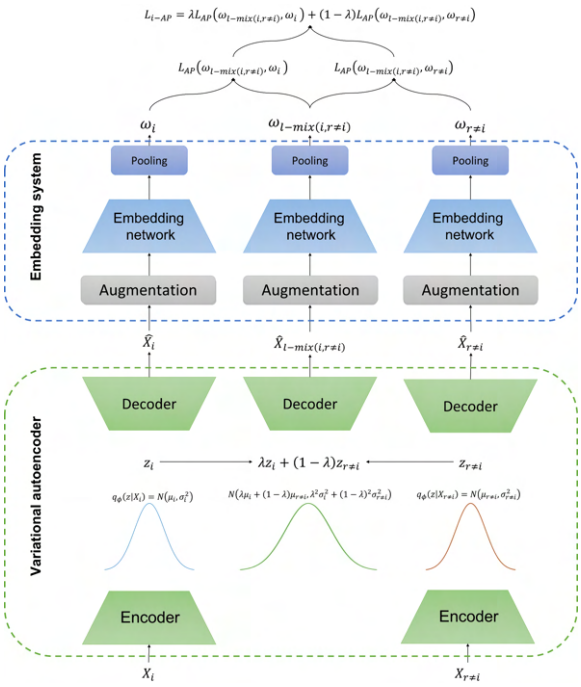


Figure 4: The general framework for the proposed latent space i-mix (l-mix) angular prototypical learning.

In order to overcome this limitation, we propose an i-mix strategy applied to the latent space of speech (l-mix). Since the latent variable of speech will include essential, disentangled information of various speech attributes, we assume that mixing up on latent variables will create a more realistic speech sample. Moreover, unlike the standard i-mix framework, which creates new samples within the line between the raw acoustic features, as the proposed l-mix interpolates on the latent space, we expect the synthetic samples created by the l-mix strategy to be much diverse.

Similar to the VarMixup proposed for image classification (Mangla et al. 2021), we use a VAE for extracting the latent variable from the given MFCC. Before training the embedding system, given training MFCC x, the VAE is trained according to the following objective:

$$L_{VAE} = D_{KL}(q_\phi(z|x)||p_\theta(z)) - E_{q_\phi(z|x)}[log_\theta(x|z)], \quad (5)$$

where z is the latent variable, ϕ is the encoder parameter and θ is the decoder parameter. The VAE is composed of two networks: encoder and decoder networks. The encoder network takes the MFCC sample as input and generates the mean and log-variance of the posterior latent distribution $q_\phi(z|x)$, assuming that the latent variables have Gaussian distributions. The decoder network takes a latent sample and reconstructs the MFCC. In our experiments, we set the latent prior $p_\theta(z)$ to be a standard normal distribution.

Once the VAE has been trained, we use the VAE to perform mixup on the latent space. Since the latent variables of the VAE are assumed to have a Gaussian distribution, the mixed-up latent variable will have a Gaussian distribution as well. For example, given two latent variables $z_1 \sim N(\mu_1, \sigma_1^2)$ and $z_2 \sim N(\mu_2, \sigma_2^2)$, the mixup between them will result in:

$$\begin{aligned} z_{mix} &= \lambda z_1 + (1-\lambda) z_2 \\ &\sim N(\lambda\mu_1 + (1-\lambda\mu)_2, \lambda^2\sigma_1^2 + (1-\lambda)^2\sigma_2^2), \end{aligned} \quad (6)$$

where $\lambda \sim Beta(\alpha, \alpha)$. The mean of the mixed up latent variable z_{mix} is fed into the decoder network to generate an MFCC sample x_{l-mix}.

The decoder generated MFCC samples are then fed into the embedding network to generate an embedding vector ω_{l-mix}, and we can train the embedding network using the same formulation with Equation 4:

$$\begin{aligned} L_{l-AP} = &-\lambda \frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega_{l-mix(i,r\neq i)}, \omega_i))}{\sum_{j=1}^{N} exp(cos(\omega_{l-mix(i,r\neq i)}, \omega_j))} \\ &- (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega_{l-mix(i,r\neq i)}, \omega_{r\neq i}))}{\sum_{j=1}^{N} exp(\omega_{l-mix(i,r\neq i)}, \omega_j))}. \end{aligned} \quad (7)$$

The general framework of the proposed latent space i-mix learning is depicted in Fig. 4.

Table 1: Architecture for the variational autoencoder (VAE) used for extracting the latent variable from the MFCCs.

Layer #	Encoder	Decoder
1	3×3 2D-Conv, 32 ReLU, stride 3	64×32 FC
2	3×3 2D-Conv, 64 ReLU, stride 3	3×3 2D-TransposedConv, 32 ReLU, stride 3
3	3×3 2D-Conv, 32 ReLU, stride 3	3×3 2D-TransposedConv, 64 ReLU, stride 3
4	3×3 2D-Conv, 32 ReLU, stride 3	3×3 2D-TransposedConv, 32 ReLU, stride 3
5	32×64 FC for each μ and $log\sigma^2$	3×3 2D-TransposedConv, 1 ReLU, stride 3

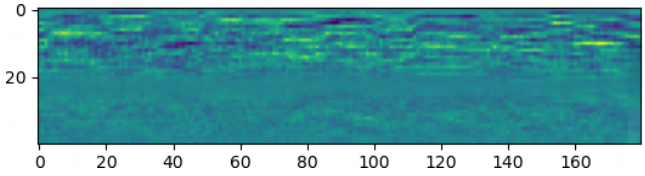
Experiments

Experimental setup

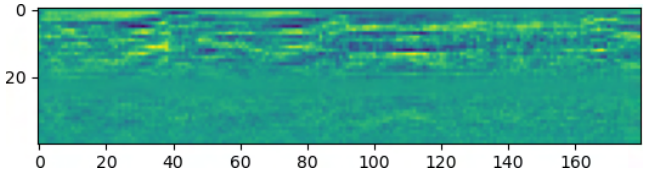
In order to evaluate the performance of the proposed technique for self-supervised speaker verification, a set of experiments were conducted based on the VoxCeleb2 dataset (Chung, Nagrani, and Zisserman 2018). For training the embedding networks, we used the *development* subset of the VoxCeleb2 dataset, consisting of 1,092,009 utterances collected from 5,994 speakers. The evaluation was performed according to the original VoxCeleb1 trial list (Nagrani, Chung, and Zisserman 2017), which consists of 4,874 utterances spoken by 40 speakers.

The acoustic features used in the experiments were 40-dimensional MFCCs extracted at every 10 ms, using a 25 ms Hamming window via Kaldi toolkit (Povey et al. 2011). The embedding networks are trained with segments consisting of 180 frames, using the ADAM optimization technique (Kingma and Ba 2015).

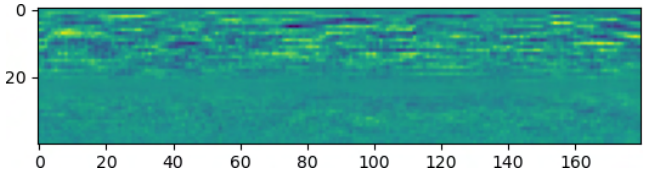
For the l-AP training, we have trained a convolutional VAE with 5 layered encoder and decoder networks, where each layer is configured as described in Table 1. The detailed information on the implementation of this VAE is described in (Ha and Schmidhuber 2018). The VAE was trained for 100 epochs on the VoxCeleb2 dataset with learning rate 0.001.



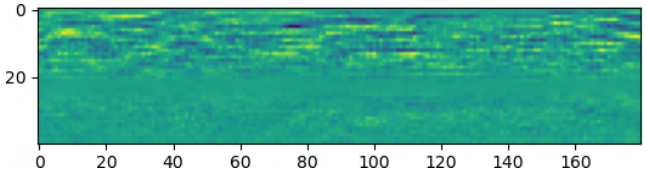
(a) MFCC of utterance X_1.



(b) MFCC of utterance X_2.



(c) Synthetic MFCC created via i-mix.



(d) Synthetic MFCC created via l-mix.

Figure 5: MFCCs of training utterances X_1 and X_2, and synthetic samples created via applying i-mix and l-mix strategy on them with $\lambda = 0.503457$.

All the experimented networks were implemented via PyTorch, based on the voxceleb-unsupervised open-source project (Huh et al. 2020)[1]. The networks were trained with initial learning rate 0.001 decayed with ratio 0.95 for 150 epochs, and the models from the best performing checkpoint were selected. The batch size for training was set to be 200. Cosine similarity was used for computing the verification scores in the experiments.

Experimental results

Analysis on synthetic samples In this section, we analyze the difference between synthetic MFCC samples generated via i-mix and l-mix. Figure 5 depicts the MFCCs of two training utterances and the synthetic MFCCs created using i-mix and l-mix strategy. As shown in the figure, even when using the same mixup coefficient ($\lambda = 0.503457$), the l-mix strategy is able to create a different sample from the i-mix augmentation. This is more apparent in Figure 6, which depicts the euclidean distance of the synthetic samples from the line between their respective original sample pair on the utterance-level MFCC supervector space (which concatenates the frame-level MFCC features. From this figure, it could be seen that the i-mix synthetic MFCCs are very close to the line between the original MFCCs, which is a natural behaviour as the i-mix samples are created via linear interpolation.

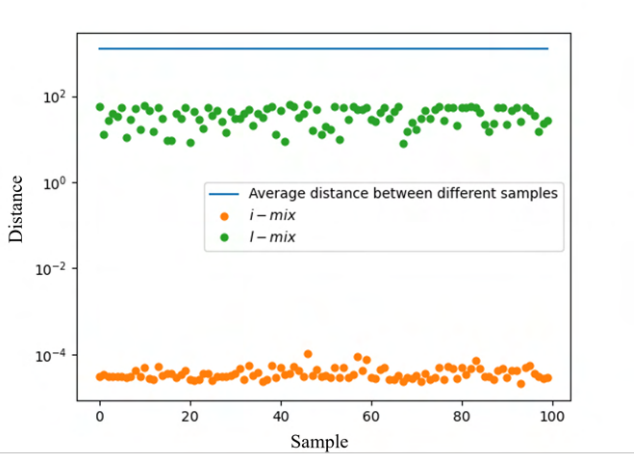


Figure 6: The euclidean distances between the MFCCs generated using i-mix or l-mix and the line defined by the training MFCCs X_1 and X_2. For each sample, the same λ, X_1, and X_2 were used for generating both i-mix and l-mix. Lower distance indicates that the synthetic MFCC is closer to the line between the original MFCCs.

On the other hand, the l-mix synthetic samples are generated with more diversity, which are not necessarily created near the line between the original samples. Attributed to this, we can assume that the l-mix will enable the network to generalize better as it will generate samples with more variability. While both i-mix and l-mix strategies are able to create new samples unobserved in the training set, they create distinctive samples from each other due to the different levels they apply interpolation on. From this observation, we can assume that training the system with both i-mix and l-mix synthesized samples can improve the generalization over the system trained with a single type of mixup strategy.

Speaker verification performance comparison between systems trained with different augmentations In this section, we compare the ECAPA-TDNN-based self-supervised systems with different mixup regularizations and conventional systems trained with contrastive loss functions. As depicted in Table 2, it could be noticed that even when using the same angular prototypical objective, the performance differs depending on the type of augmentation applied to the input audio. For example, using cepsaugment with waveform-level augmentation was able to outperform the ECAPA-TDNN system trained with only waveform-level augmentation with a relative improvement of 0.31% in terms of EER. This reassures that the selection of data augmentation method is important for obtaining optimal self-supervised embedding vectors.

On the other hand, the i-mix angular prototypical objective was able to improve the performance in all augmentation settings (i.e., waveaug, waveaug+cepsaug). In most

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

Table 2: EER (%) comparison between the embedding networks trained with different augmentations and objectives.

Augmentation	Objective	EER [%]
	Human Benchmark (Huh et al. 2020)	15.7700
None	i-vector (Huh et al. 2020)	15.2800
	AP (FastResNet34) (Huh et al. 2020)	25.3700
waveaug	GCL (ResNet18) (Inoue and Goto 2020)	15.2600
	AP (FastResNet34) (Huh et al. 2020)	11.6000
waveaug	AP	11.6384
	i-AP ($\alpha = 0.5$)	11.9618
	i-AP ($\alpha = 1.0$)	11.2407
	i-AP ($\alpha = 32.0$)	11.8240
	l-AP ($\alpha = 0.5$)	11.8876
	l-AP ($\alpha = 1.0$)	**10.7741**
	l-AP ($\alpha = 32.0$)	11.7179
waveaug +cepsaug	AP	11.6013
	i-AP ($\alpha = 0.5$)	10.6257
	i-AP ($\alpha = 1.0$)	10.9279
	i-AP ($\alpha = 32.0$)	12.1633
	l-AP ($\alpha = 0.5$)	**10.4931**
	l-AP ($\alpha = 1.0$)	10.5408
	l-AP ($\alpha = 32.0$)	11.8399

cases, i-AP with $\alpha = 0.5, 1.0$ outperformed AP, while $\alpha = 32.0$ hindered the performance. Especially in ECAPA-TDNN with waveaugment and cepsaugment, i-AP ($\alpha = 0.5$) outperformed AP with a relative improvement of 8.41% in terms of EER. This shows that with the right choice of α, the self-supervised embedding network can be improved significantly via incorporating i-mix regularization to the objective.

While instance mixup on data-level (i-AP) showed promising results, the proposed latent-level instance mixup (l-AP) was able to further enhance the performance in all augmentation settings. Analogous to the i-AP results, in most cases l-AP was found to be beneficial to the system with $\alpha = 0.5, 1.0$. The best performance was observed from ECAPA-TDNN with waveaugment and caepsaugment trained via l-AP ($\alpha = 0.5$), which outperformed AP with a relative improvement of 9.55% in terms of EER. Such improvement may be attributed to the l-APs larger capability in generalization, which we have observed from Figure 6.

Conclusion

In this paper, we investigated the utilization of the i-mix regularization for self-supervised speaker embedding learning in order to increase the generalization of the embedding vectors on out-of-domain utterances. Furthermore, we proposed l-mix, a mixup strategy that applies i-mix on the latent space, instead of the raw MFCC feature domain.

In order to evaluate the i-mix and l-mix strategies, we have conducted several experiments on the VoxCeleb dataset. Our results showed that both i-mix and l-mix can significantly improve the generalization of the self-supervised embeddings with the right choice of hyperparameters, hence outperforming the systems trained with the standard angular prototypical objective. The best performance was observed when using l-mix along with wave-level augmentation and cepsaugment, which outperformed the system trained with standard angular prototypical objective with a relative improvement of 9.55% in terms of EER.

In our future study, we will be investigating the potential of l-mix strategy in self-supervised speaker verification more in-depth, by applying l-mix into different types of self-supervised objective functions else than angular prototypical loss. Moreover, as our proposed l-mix is essentially an i-mix performed on latent space, it still relies on linear interpolation when mixing samples. Although linear interpolation is a viable method in image domain, such interpolation may not be optimal for speech spectral features as the amount of speaker-dependent information greatly differs depending on the frequency or quefrency region. In order to tackle this issue, we will further expand the l-mix technique, by formulating a more speech-adequate and sophisticated method for mixing samples on the latent level.

References

Chen, J.; Wang, Z.; Tian, R.; Yang, Z.; and Yang, D. 2020. Local Additivity Based Data Augmentation for Semi-supervised NER. arXiv:2010.01677.

Chung, J. S.; Nagrani, A.; and Zisserman, A. 2018. VoxCeleb2: Deep Speaker Recognition. In *INTERSPEECH*.

Dehak, N.; Kenny, P. J.; Dehak, R.; Dumouchel, P.; and Ouellet, P. 2011. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4): 788–798.

Desplanques, B.; Thienpondt, J.; and Demuynck, K. 2020. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In Meng, H.; Xu, B.; and Zheng, T. F., eds., *Interspeech 2020*, 3830–3834. ISCA.

Ding, K.; He, X.; and Wan, G. 2020. Learning Speaker Embedding with Momentum Contrast. arXiv:2001.01986.

Ha, D.; and Schmidhuber, J. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31*, 2451–2463. Curran Associates, Inc. https://worldmodels.github.io.

Hansen, J. H.; and Hasan, T. 2015. Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6): 74–99.

Huh, J.; Heo, H. S.; Kang, J.; Watanabe, S.; and Chung, J. S. 2020. Augmentation adversarial training for unsupervised speaker recognition. In *Workshop on Self-Supervised Learning for Speech and Audio Processing, NeurIPS*.

Inoue, N.; and Goto, K. 2020. Semi-Supervised Contrastive Learning with Generalized Contrastive Loss and Its Application to Speaker Recognition. *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1641–1646.

Kenny, P. 2012. A small footprint i-vector extractor. In *Odyssey*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Lee, K.; Zhu, Y.; Sohn, K.; Li, C.-L.; Shin, J.; and Lee, H. 2021. i-Mix: A Domain-Agnostic Strategy for Contrastive Representation Learning. In *ICLR*.

Mangla, P.; Singh, V.; Havaldar, S.; and Balasubramanian, V. 2021. On the benefits of defining vicinal distributions in latent space. *Pattern Recognition Letters*.

Mun, S. H.; Kang, W. H.; Han, M. H.; and Kim, N. S. 2020. Unsupervised Representation Learning for Speaker Recognition via Contrastive Equilibrium Learning. arXiv:2010.11433.

Nagrani, A.; Chung, J. S.; and Zisserman, A. 2017. VoxCeleb: a large-scale speaker identification dataset. In *INTERSPEECH*.

Niizumi, D.; Takeuchi, D.; Ohishi, Y.; Harada, N.; and Kashino, K. 2021. BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation. arXiv:2103.06695.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, 2613–2617.

Povey, D.; Ghoshal, A.; Boulianne, G.; Goel, N.; Hannemann, M.; Qian, Y.; Schwarz, P.; and Stemmer, G. 2011. The kaldi speech recognition toolkit. In *In IEEE 2011 workshop*.

Snyder, D.; Garcia-Romero, D.; Povey, D.; and Khudanpur, S. 2017. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In *INTERSPEECH*.

Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; and Khudanpur, S. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5329–5333.

Tomilov, A.; Svishchev, A.; Volkova, M.; Chirkovskiy, A.; Kondratev, A.; and Lavrentyeva, G. 2021. STC Antispoofing Systems for the ASVspoof2021 Challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 61–67.

Tran, V.-T.; and Tsai, W.-H. 2020. Speaker Identification in Multi-Talker Overlapping Speech Using Neural Networks. *IEEE Access*, 8: 134868–134879.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6438–6447. Long Beach, California, USA: PMLR.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, H.; Zou, Y.; and Wang, H. 2021. Contrastive Self-Supervised Learning for Text-Independent Speaker Verification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6713–6717.

Zhu, Y.; Ko, T.; and Mak, B. 2019. Mixup Learning Strategies for Text-Independent Speaker Verification. In *Proc. Interspeech 2019*, 4345–4349.