CRIM

CENTRE DE
RECHERCHE INFORMATIQUE
DE MONTRÉAL

INVESTIGATION ON INSTANCE MIXUP REGULARIZATION STRATEGIES FOR SELF-SUPERVISED SPEAKER REPRESENTATION LEARNING

WOO HYUN KANG,
JAHANGIR ALAM,
ABDERRAHIM FATHAN

SELF-SUPERVISED SPEAKER REPRESENTATION LEARNING

- Why do we need self-supervised speaker representation learning?
  - Nowadays, vast speech data can be obtained for training a speaker verification system
    - E.g., YouTube, Soundcloud, TikTok, etc.
  - However, **most of these speech samples do not have any speaker labels**
    - Also, collecting speaker labeled speech samples can be very expensive in terms of resources

CONVENTIONAL SPEAKER EMBEDDING SYSTEMS

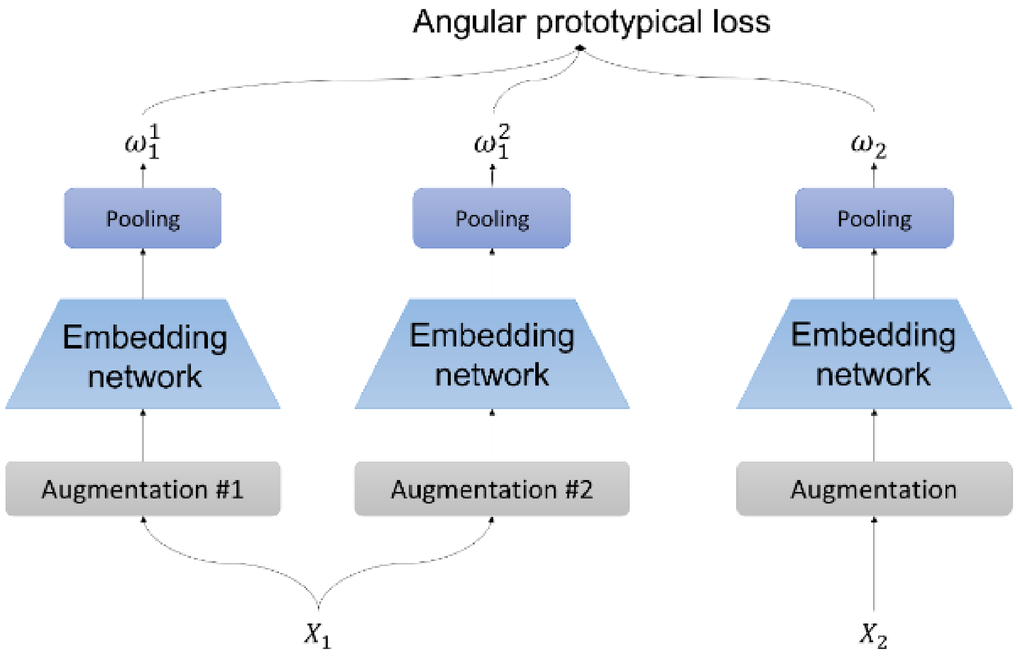
- Although the conventional deep embedding schemes showed impressive performance, **they require speaker labels to be trained**:
  - End-to-end systems: require speaker labels to define positive and negative pairs for contrastive objective functions

→Therefore we should utilize pseudo-labels to apply these frameworks to self-supervised scenarios

SELF-SUPERVISED ANGULAR PROTOTYPICAL LOSS

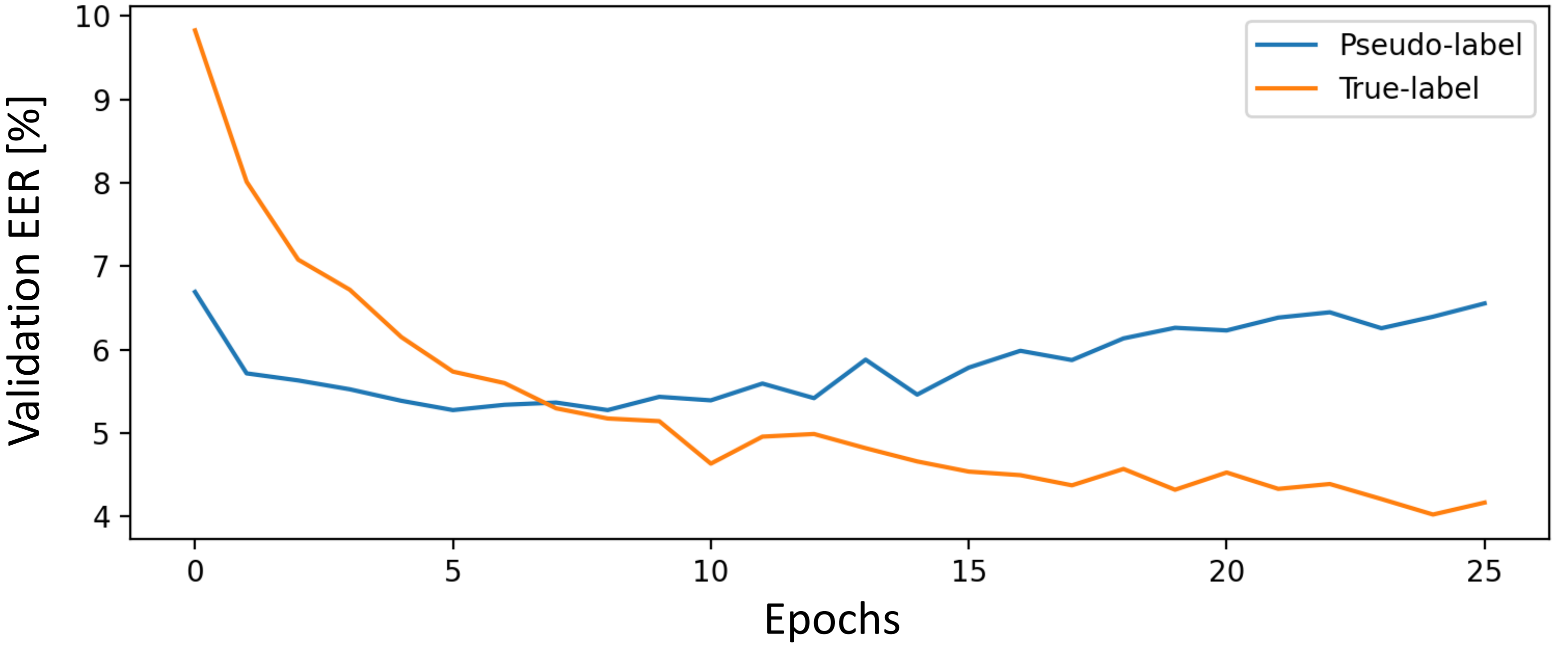
- For **contrastive objectives**, we need to define positive pairs and negative pairs
  - In a self-supervised scenario, we can **consider the utterance identity as pseudo-labels**
  - For each training utterance, we apply **two different types of augmentations**, resulting in two samples
    - Samples created from the same utterance are considered as positive pair
    - Samples created from different utterances are considered as negative pair
  - This way, we can apply contrastive loss functions, such as angular prototypical objective

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



PROBLEM WITH TRAINING WITH PSEUDO-LABELS

- Using pseudo-labels can allow us to train speaker embedding systems with unlabeled dataset, but the **performance is limited as these are not actual speaker labels**
  - Therefore, overfitting on the pseudo-labels can cause critical performance degradation
    - As the system is optimized more to the pseudo-labels, it Is likely for the **system to learn non-speaker attributes**



INSTANCE MIXUP (I-MIX)

- I-mix is a data-driven augmentation strategy for **improving the generalization of the self-supervised representation**
  - For arbitrary objective function $L_{pair}(x, y)$, where x is the **input sample** and y is the corresponding **pseudo-label**, giving two data instances (x_i, y_i) and (x_j, y_j),

$$L_{pair}^{i-mix}((x_i, y_i), (x_j, y_j)) \\ = L_{pair}(\lambda x_i + (1-\lambda)x_j, \lambda y_i + (1-\lambda)y_j),$$

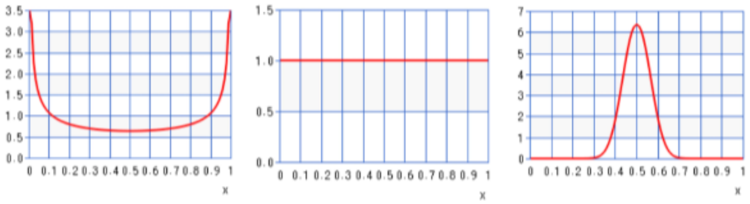
  - For cross-entopy-based loss (e.g., prototypical loss), this equation can be rewritten as

$$L_{pair}^{i-mix}((x_i, y_i), (x_j, y_j)) \\ = \lambda L_{pair}(x_i, y_i) + (1-\lambda)L_{pair}(x_j, y_j).$$

→Essentially, this **creates synthetic training samples with new pseudo-identities**

INSTANCE MIXUP (I-MIX)

- Here, the mixup coefficient $\lambda \sim Beta(\alpha, \alpha)$
  - This distribution yields λ **with value between 0 and 1**
  - Depending on the α, the distribution shape varies (symmetric)
    - $\alpha < 1.0$: U-shaped distribution, where the sampled λ is likely to have value close to either 1.0 or 0.0
    - $\alpha = 1.0$: a uniform distribution across 0 to 1
    - $\alpha > 1.0$: a bell-shaped distribution, where the sampled λ is likely to have value close to 0.5

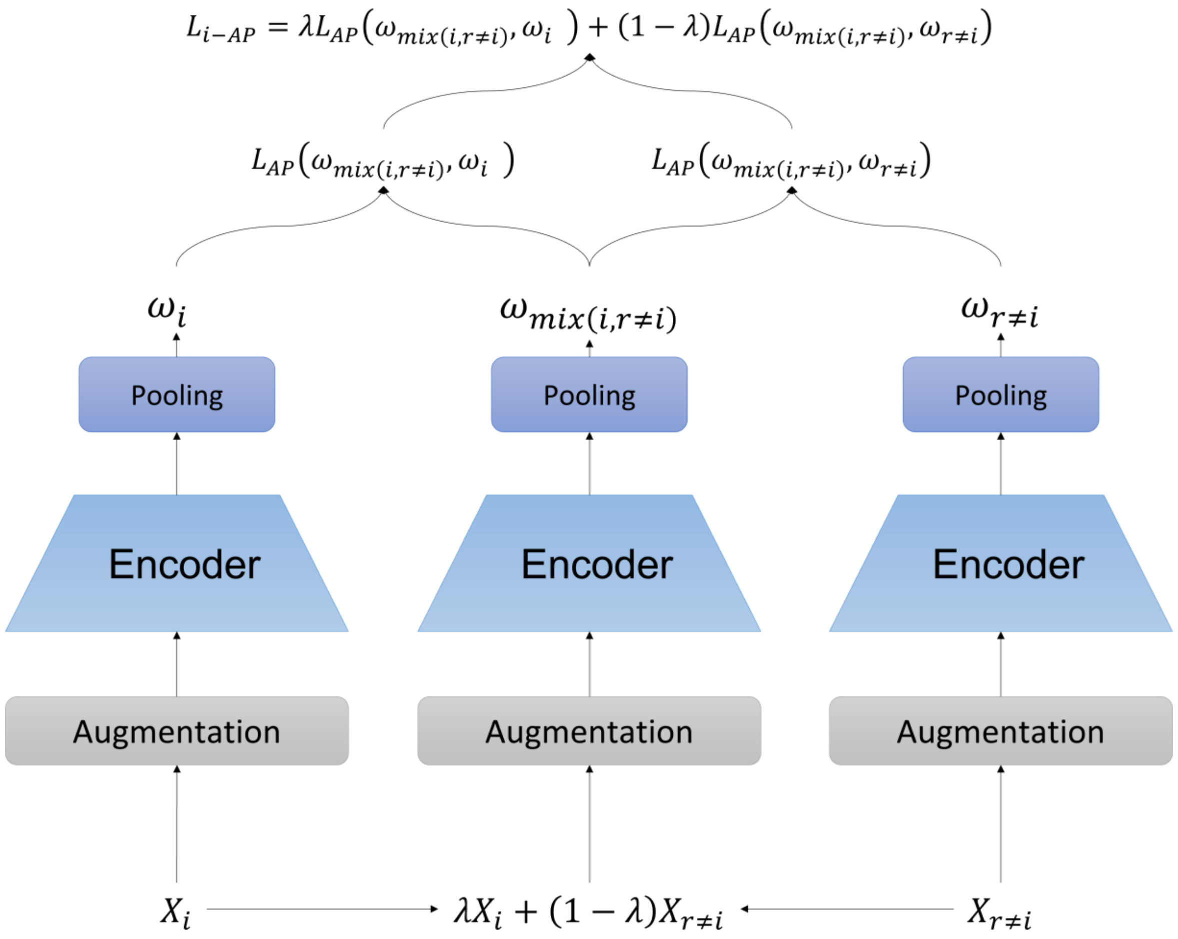


$\alpha = 0.5$ $\qquad$ $\alpha = 1.0$ $\qquad$ $\alpha = 32.0$

I-MIX ANGULAR PROTOTYPICAL OBJECTIVE (I-AP)

- To enhance the generalization of the self-supervised speaker embedding system, we **applied the i-mix strategy to the angular prototypical objective**
  - We apply **interpolation on the input acoustic features and utterance identity pseudo-labels**

$$
\begin{aligned}
L_{i-AP} = & -\lambda \frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega^1_{mix(i,r\neq i)}, \omega^2_i))}{\sum_{j=1}^{N} exp(cos(\omega^1_{mix(i,r\neq i)}, \omega^2_j))} \\
& - (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega^1_{mix(i,r\neq i)}, \omega^2_{r\neq i}))}{\sum_{j=1}^{N} exp(\omega^1_{mix(i,r\neq i)}, \omega^2_j))},
\end{aligned}
$$



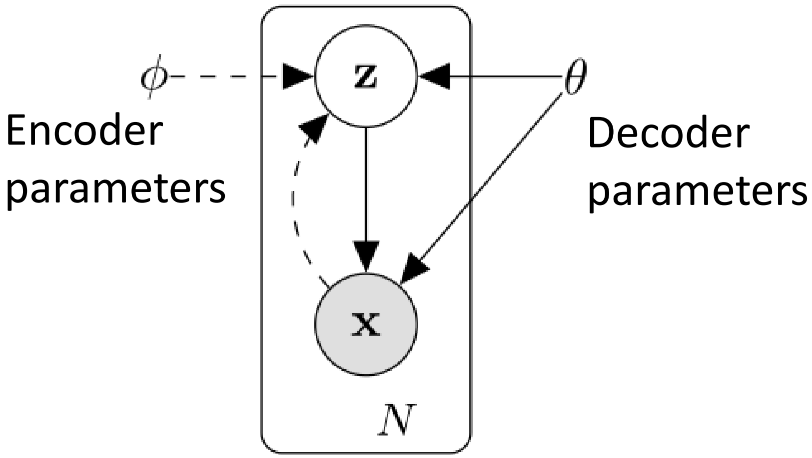
POSSIBLE LIMITATION OF THE I-AP

- Although applying mixup augmentation to the raw data have proven its strength in many tasks (e.g., speech recognition, image classification), there is room for improvement
  - **Due to the linear interpolation, i-mix strategy can only generate new samples between the original samples on the feature space**
  - This restricts the diversity of the synthetic training samples, thus limiting the generalization of the system

LATENT SPACE INSTANCE MIXUP (L-MIX)

- In order to overcome this limitation, we propose **an i-mix strategy applied to the latent space of speech (l-mix)**
  - The latent variable of speech will include essential, disentangled information of various speech attributes
  - We use a **variational autoencoder for extracting the latent variable from the given acoustic features (i.e., MFCC)**
    - Prior to training the embedding system, we train a VAE for reconstructing the acoustic features

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



LATENT SPACE INSTANCE MIXUP (L-MIX)

- Once the VAE is trained, we can use this for extracting the latent variable and reconstructing the acoustic feature
  - The VAE encoder generates the Gaussian posterior latent distribution $z \sim N(\mu, \sigma^2)$
  - **The latent distributions are linearly interpolated**, which yields a new Gaussian distribution (weighted sum of independent normal distributions)

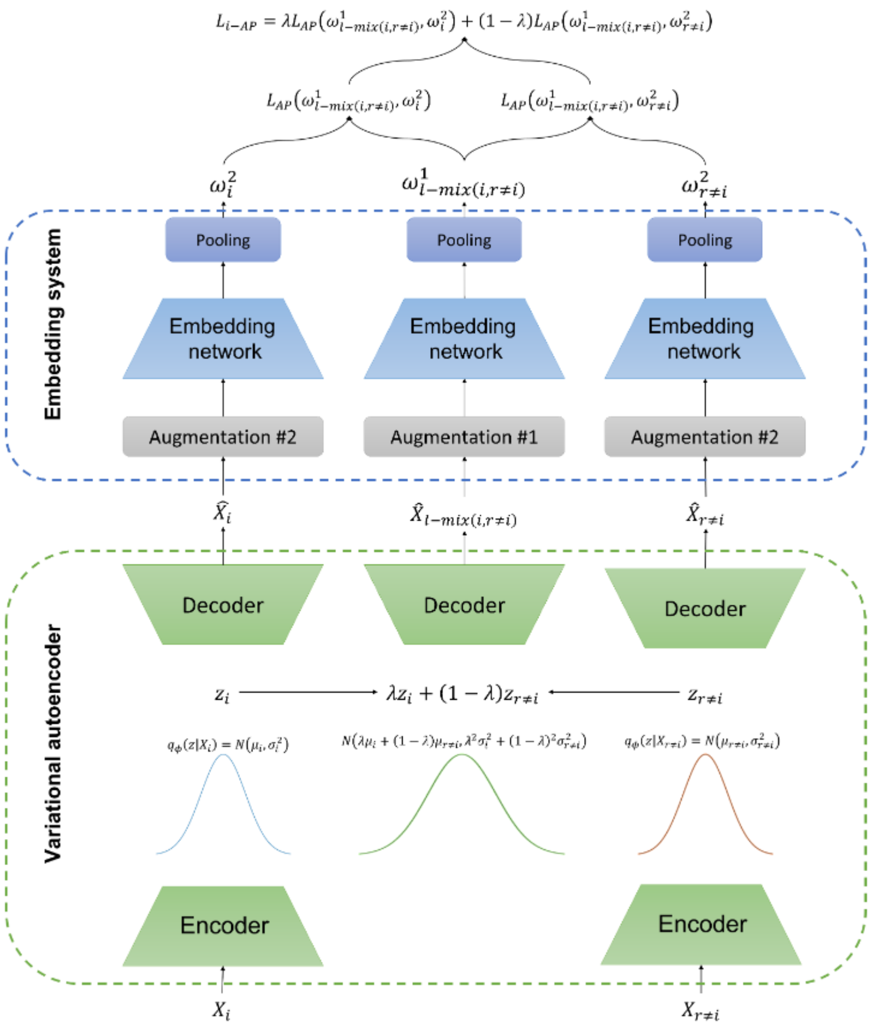
$$z_{mix} = \lambda z_1 + (1 - \lambda) z_2$$
$$\sim N(\lambda\mu_1 + (1 - \lambda\mu)_2, \lambda^2\sigma_1^2 + (1 - \lambda)^2\sigma_2^2),$$

  - **The mixed up latent variable is fed into the decoder network to generate a synthetic acoustic feature $\boldsymbol{x}_{l-mix}$**

L-MIX ANGULAR PROTOTYPICAL OBJECTIVE (L-AP)

- Analogous to i-AP, we can apply the l-mix strategy to the angular prototypical objective

$$L_{l-AP} = -\lambda \frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega^1_{l-mix(i,r\neq i)}, \omega^2_i))}{\sum_{j=1}^{N} exp(cos(\omega^1_{l-mix(i,r\neq i)}, \omega^2_j))}$$
$$- (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(cos(\omega^1_{l-mix(i,r\neq i)}, \omega^2_{r\neq i}))}{\sum_{j=1}^{N} exp(\omega^1_{l-mix(i,r\neq i)}, \omega^2_j))}.$$



EXPERIMENT

- **VoxCeleb dataset**
  - Training set
    - VoxCeleb2 development set
      - 5994 speakers included (no labels were used for our experiments)
  - Evaluation set
    - VoxCeleb1 trial

- **Acoustic features**
  - 40 dim. MFCC (mel filterbank cepstral coefficients) features
  - Augmentations:
    - Wave-level augmentation: MUSAN noise or RIR simulation
    - Cepstrum-level augmentation: random cepstrum/frame masking (similar to SpecAugment)

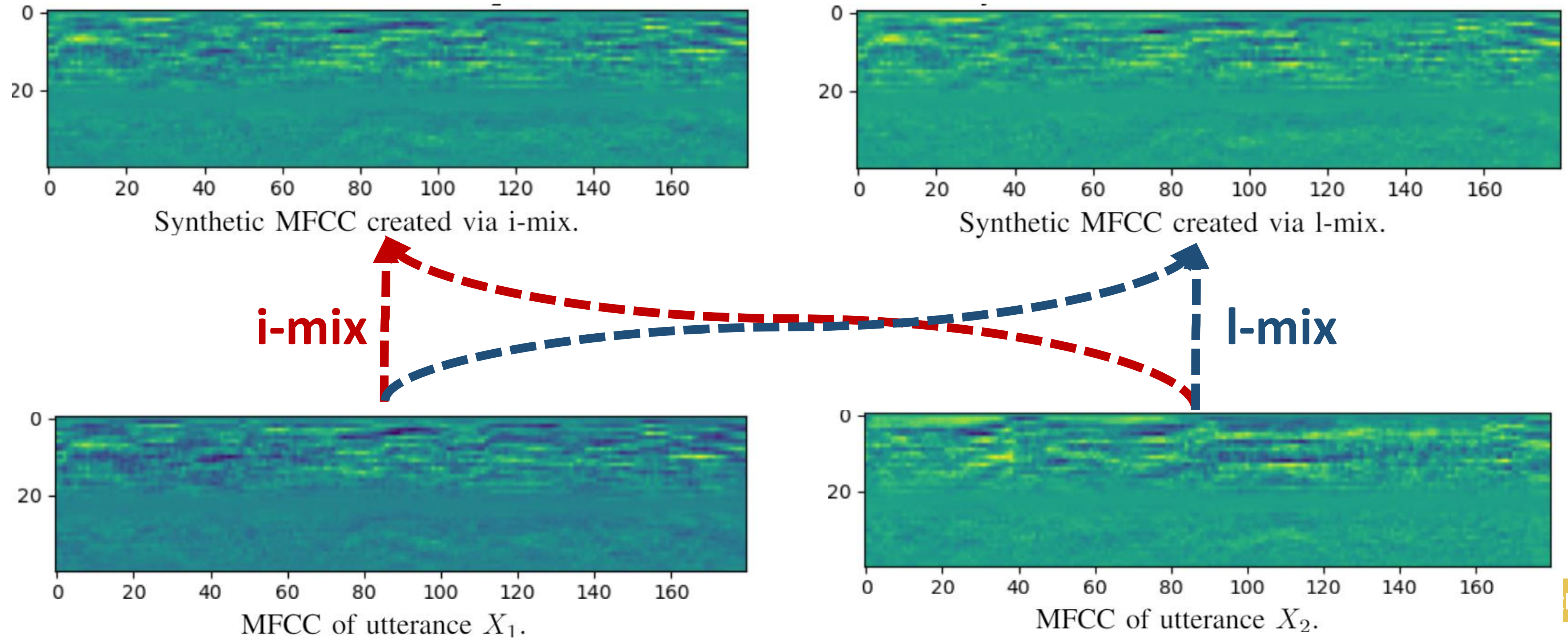
EXPERIMENT

- **Embedding system**
  - **ECAPA-TDNN architecture: state-of-the-art system for supervised text-independent speaker recognition**
  - Attentive channel- and context-dependent statistics pooling
  - Multi-layer aggregation
  - Embedding dimension: 512

- **Variational autoencoder (VAE)**
  - **10 layered convolutional VAE**

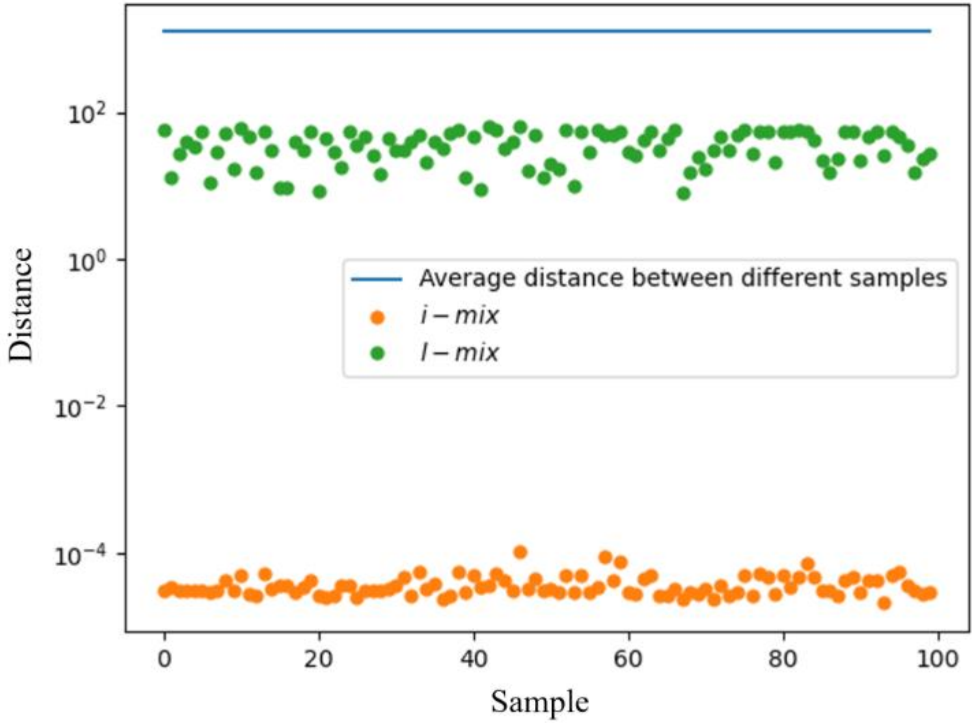
Layer #	Encoder	Decoder
1	3×3 2D-Conv, 32 ReLU, stride 3	64×32 FC
2	3×3 2D-Conv, 64 ReLU, stride 3	3×3 2D-TransposedConv, 32 ReLU, stride 3
3	3×3 2D-Conv, 32 ReLU, stride 3	3×3 2D-TransposedConv, 64 ReLU, stride 3
4	3×3 2D-Conv, 32 ReLU, stride 3	3×3 2D-TransposedConv, 32 ReLU, stride 3
5	32×64 FC for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

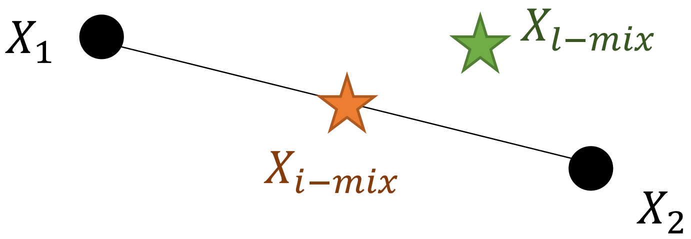
- **Analysis on synthetic samples**
  - Since i-mix and l-mix applies mixup on different space, **they can create very different samples even when using the same mixup coefficient**



EXPERIMENT

- **Analysis on synthetic samples**
  - Since the **i-mix strategy** applies linear interpolation on the feature space, the **generated samples are placed on the line between the two original samples**
  - On the other hand, the **samples created via l-mix are not necessarily placed on the line**
    - This indicates that the **l-mix can create samples with more diversity on the feature space**





EXPERIMENT

- **Speaker verification performance**
  - Here, we compare performance of the systems trained with various objective functions and augmentations on the VoxCeleb1 evaluation set

→ i-mix and l-mix can both improve the performance when the right coefficient is used

➔The **best performance was observed when using l-mix** along with wave-level augmentation and cepsaugment

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	w/o regularization
	i-AP ($\alpha = 0.5$)	11.9618	
	i-AP ($\alpha = 1.0$)	11.2407	I-MIX
	i-AP ($\alpha = 32.0$)	11.8240	
	l-AP ($\alpha = 0.5$)	11.8876	
	l-AP ($\alpha = 1.0$)	**10.7741**	L-MIX
	l-AP ($\alpha = 32.0$)	11.7179	
waveaug +cepsaug	AP	11.6013	w/o regularization
	i-AP ($\alpha = 0.5$)	10.6257	I-MIX
	i-AP ($\alpha = 1.0$)	10.9279	
	i-AP ($\alpha = 32.0$)	12.1633	
	l-AP ($\alpha = 0.5$)	**10.4931**	L-MIX
	l-AP ($\alpha = 1.0$)	10.5408	
	l-AP ($\alpha = 32.0$)	11.8399	

CONCLUSION

- We incorporate the i-mix strategy to the self-supervised speaker embedding learning framework for robust speaker verification
- We also propose a latent space i-mix strategy (l-mix), which performs i-mix on the latent space of the speech
- Our experimental results show that the self-supervised speaker embedding learning can benefit greatly from the i-mix regularization strategy
- Moreover, the proposed l-mix strategy can further improve the performance, by yielding much diverse synthetic training samples

Q & A