

# CONTRASTIVE-MIXUP LEARNING FOR IMPROVED SPEAKER VERIFICATION

Xin Zhang<sup>†\*</sup>

Texas A&M University  
Dept. of Computer Science & Engineering  
College Station, TX, USA  
xin.tamu@tamu.edu

Minho Jin<sup>†</sup>, Roger Cheng, Ruirui Li,  
Eunjung Han, Andreas Stolcke

Amazon Alexa AI  
Sunnyvale, CA, USA  
{minhoj, rogcheng, ruirul}@amazon.com  
{cehan, stolcke}@amazon.com

## ABSTRACT

This paper proposes a novel formulation of prototypical loss with mixup for speaker verification. Mixup is a simple yet efficient data augmentation technique that fabricates a weighted combination of random data point and label pairs for deep neural network training. Mixup has attracted increasing attention due to its ability to improve robustness and generalization of deep neural networks. Although mixup has shown success in diverse domains, most applications have centered around closed-set classification tasks. In this work, we propose contrastive-mixup, a novel augmentation strategy that learns distinguishing representations based on a distance metric. During training, mixup operations generate convex interpolations of both inputs and virtual labels. Moreover, we have reformulated the prototypical loss function such that mixup is enabled on metric learning objectives. To demonstrate its generalization given limited training data, we conduct experiments by varying the number of available utterances from each speaker in the VoxCeleb database. Experimental results show that applying contrastive-mixup outperforms the existing baseline, reducing error rate by 16% relatively, especially when the number of training utterances per speaker is limited.

*Index Terms*— mixup, metric learning, speaker verification, prototypical loss.

## 1. INTRODUCTION

Extensive research has been devoted to improving speaker recognition systems. The objective of speaker verification is to answer the question “Was the input spoken by an enrolled speaker?”. The performance of a speaker verification system relies on access to sufficient and clean training data for supervised training [1]. However, one of the challenges in training speaker verification models is the lack of large amounts of well-labelled training data. In other domains, researchers have developed various data augmentation techniques to overcome this bottleneck, enhancing the generalization of deep networks given limited data. For example, data augmentation has been used in computer vision [2], natural language processing [3], and semi-supervised learning [4].

Some well-known data augmentation approaches are applicable to speech recognition. Examples include adding noise, time stretching, pitch shift, and SpecAugment [5, 6]. The common idea behind these approaches is to deform either the raw audio or the spectrogram with various operations, such as time and frequency masking.

By applying these augmentations, embedding extractors learn to be more robust to variations and thus enable better generalization [7].

In this paper, we propose contrastive mixup for training of a neural network’s embedding extractor. Mixup is a regularization technique that trains the network with linear interpolations of input samples and corresponding interpolated labels. While the mixup technique has proven effective for closed-set classification tasks [8], it is not clear how well it works for open-set applications such as speaker verification. Recent work has shed light on why mixup leads to improved robustness and generalization of the trained model from a theoretical perspective [9]. However, to the best of our knowledge, little work has been done to apply mixup to speaker verification systems. To fill this gap, we have developed contrastive mixup, a variant of the original mixup technique that is compatible with the training of speaker verification models.

The key innovation of this paper is to demonstrate how to implement mixup for contrastive learning with metric learning objectives. We focus on speaker verification, which represents an open-set classification task. The speaker embeddings are extracted using a ResNet-based backbone model. To make a verification decision, the distance (cosine distance in this paper) is computed between the extracted embedding and the profile. The model is trained with batches consisting of a predefined number of utterances from a predefined number of speakers. During training, the prototype is calculated based on the original utterances from each individual speaker. Importantly, the query utterance is obtained by conducting linear interpolations of utterances from different speakers. Moreover, we choose angular prototypical loss to establish different baselines in speaker verification systems due to the results reported in previous work [10, 11].

The prototypical networks are originally formulated for problems of few-shot learning, where each class can be represented and discriminated based on the mean of corresponding examples. One of the advantages of prototypical networks is that trained models can learn rare cases after being exposed to a small amount of prior information [12]. However, it still remains unclear how to integrate the mixup algorithm with prototypical loss because the original prototypical loss function solely relies on a distance metric between samples, not involving the use of labels. Consequently, we have to reformulate the prototypical loss function by taking advantage of label information such that the mixup operation can be incorporated in the metric learning objective.

Furthermore, we conducted experiments to compare different types of implementations of the mixup algorithms, in both time-domain speech samples and Mel spectrogram features. The results demonstrate that contrastive mixup is effective in improving speaker

\*Work done during an internship at Amazon.

<sup>†</sup>Equal contributions

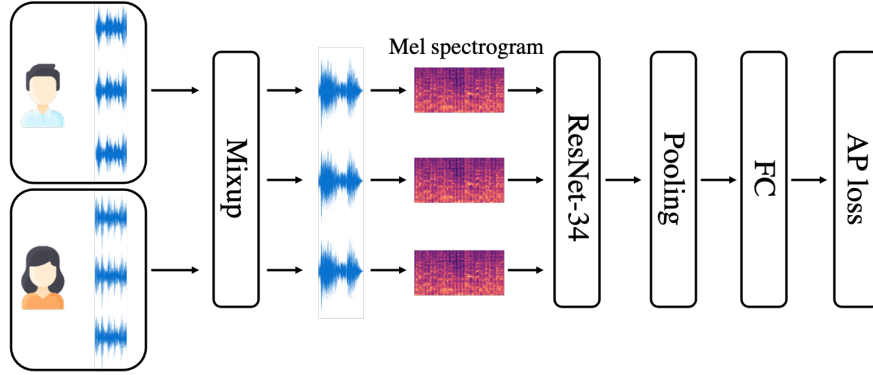


Fig. 1. Schematic diagram of contrastive mixup

verification performance, especially when the number of utterances for each speaker is limited. Differently from existing augmentation approaches, such as adding noise [13] and room response simulation [14], we believe contrastive mixup offers a novel and effective augmentation strategy that improves generalization while introducing negligible computational overhead.

The remainder of this paper is structured as follows: Section 2 describes prior work related to mixup. Section 3 presents the proposed contrastive mixup, Section 4 shows results and observations, and Section 5 summarizes our findings.

## 2. RELATED WORK

Hongyi *et al.* [8] proposed the original version of the mixup algorithm by generating linear interpolations of data-label pairs. They empirically demonstrated that the linear behavior reduces the amount of undesirable oscillations when predicting outside the training examples. Recent works have shed light on the theoretical understanding of mixup regularization on deep neural networks. For example, [15] shows that neural networks trained with mixup are significantly better calibrated and less prone to over-confident predictions on random noise data. [16] points out that random perturbation of the new interpretation of mixup leads to label smoothing and reduction of the Lipschitz constant of the estimator. In addition, several mixup variants have been proposed across a variety of tasks and models. For instance, feature-level interpolation and other types of transformations have been studied [17, 18, 19]. [20] proposes the CutMix augmentation strategy, improving the performance of image classifier.

From the perspective of training functions, the mixup method has been primarily utilized in conjunction with classification loss where virtual data points and virtual labels are fabricated by randomly interpolating original data points and ground-truth labels. Unlike classification tasks that aim to avoid misclassification, the goal of metric learning is to minimize the intra-class distances while expanding the inter-class distances. Metric learning objectives have been successfully employed for applications including speaker verification. For example, triplet loss [21] and generalized end-to-end (GE2E) loss [22] have shown remarkable performance for speaker verification when compared to conventional classification loss [10]. Therefore, it becomes an intriguing question how mixup augmentation might be combined with metric learning objectives to improve speaker verification performance.

## 3. CONTRASTIVE MIXUP MODEL

Fig. 1 illustrates the overall architecture of contrastive mixup. Given a batch of utterances, utterances from different speakers are randomly mixed to fabricate mixed inputs. Mixed utterances are fed into the backbone network, and finally a novel contrastive mixup loss function is applied.

### 3.1. Angular Prototypical Loss

The original prototypical network is designed for few-shot learning tasks [12] where good generalization can be achieved by training on a small number of samples in each class. Recent work demonstrates that metric learning objectives, such as generalized end-to-end and angular prototypical networks, outperform conventional classification objectives for speaker recognition. In particular, Chung *et al.* [10] presented a comprehensive study for metric learning in speaker recognition and conclude that angular prototypical loss outperforms state-of-the-art methods. In this paper, we choose angular prototypical (AP) loss to establish our baselines.

During training, suppose each batch contains  $M$  utterances from  $N$  different speakers. The embedding of each speaker is  $x_{j,i}$ , where  $1 \leq j \leq N$  and  $1 \leq i \leq M$ . For the prototypical network, each batch consists of a support set  $S$  and a query set  $Q$ . The  $M$ th utterance is set to be the query utterance. The centroid of speaker  $j$  is computed:

$$c_j = \frac{1}{M-1} \sum_{m=1}^{M-1} \mathbf{x}_{j,m} \quad (1)$$

Then, the distance between  $\mathbf{x}_{j,M}$  and  $\mathbf{c}_k$  are computed using trainable scale and bias coefficients,  $w$  and  $b$ , as follows:

$$\mathbf{S}_{j,k} = w \cos(\mathbf{x}_{j,M}, \mathbf{c}_k) + b \quad (2)$$

Finally, the angular prototypical loss is computed using exponentiation as follows:

$$L_p = -\frac{1}{N} \sum_{j=1}^N \log \frac{e^{\mathbf{S}_{j,j}}}{\sum_{k=1}^N e^{\mathbf{S}_{j,k}}} \quad (3)$$

As shown in (3), and as reported in [23], angular prototypical loss outperforms the original prototypical networks and GE2E for speaker recognition tasks. As a result, we use AP loss as the baseline loss function for model training in our experiments.

### 3.2. Mixup

Mixup training is based on the principle of vicinal risk minimization where the classifier is trained in the vicinity of each training sample [8]. Despite its simplicity, mixup has succeeded in a wide range of applications including computer vision [8], natural language processing [24], and semi-supervised learning [25]. In mixup, the input data and its associated label is modified as follows:

$$\begin{aligned}\bar{x}_i &= \lambda x_i + (1 - \lambda)x_{R_i} \\ \bar{y}_i &= \lambda y_i + (1 - \lambda)y_{R_i}\end{aligned}\quad (4)$$

where  $x_i (0 \leq i \leq B - 1)$  is the  $i$ th data in the batch of size  $B$ , and  $y_i$  is the one-hot encoding of the label.  $R_i$  is the  $i$ th randomly shuffled index. For each training batch, the interpolation parameter  $\lambda$  is sampled from a symmetric  $Beta(\alpha, \alpha)$  distribution with  $\alpha \in (0, \infty)$ . Given fabricated input-label pairs,  $(\bar{x}, \bar{y})$ , the loss function for the classification task is computed as follows:

$$L_{mixup}(\bar{x}, y, y_R, \lambda) = \lambda CE(\bar{x}, y) + (1 - \lambda)CE(\bar{x}, y_R), \quad (5)$$

where CE denotes the cross-entropy loss. To apply mixup to AP loss, our first approach is to use cross entropy losses with respect to the ground truth and the shuffled labels as follows:

$$L_{CE, mixup} = -\frac{1}{N} \sum_{i=1}^N \left[ \lambda \log \frac{e^{\bar{s}_{i,i}}}{\sum_{k=1}^N e^{\bar{s}_{i,k}}} + (1 - \lambda) \log \frac{e^{\bar{s}_{i,R_i}}}{\sum_{k=1}^N e^{\bar{s}_{i,k}}} \right] \quad (6)$$

by interpolating the losses with the ground truth and the shuffled labels. In the rest of this paper, (6) is referred to CE mixup. When applying mixup for raw speech signals, it is critical to normalize the volume when interpolating utterances from different speakers as it adjusts sound from diverse sources to the same volume level.

### 3.3. Contrastive-mixup Loss Function

While mixup approach has been widely used for classification tasks, it remains a question how to implement mixup with metric learning objectives. Herein, we reformulate the AP loss by introducing a binary label function  $d_{j,k}$ :

$$L_{c, mixup} = -\frac{1}{N} \sum_{j=1}^N \log \frac{\sum_{k=1}^N d_{j,k} e^{\bar{s}_{j,k}}}{\sum_{k=1}^N e^{\bar{s}_{j,k}}} \quad (7)$$

where  $0 \leq d_{j,k} \leq 1$ . When  $d_{j,k} = 1$  for  $j = k$  and 0 otherwise, this becomes identical to (3). In this paper, we set  $d_{j,k}$  as follows:

$$d_{j,k} = \begin{cases} \lambda, & \text{if } k = j \\ 1 - \lambda, & \text{if } k = R_j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

In this scenario, the binary label  $d_{j,k}$  can act as the virtual label for speaker/utterance pairs. When applying mixup on reformulated AP loss, the convex interpolation can be performed on both original utterances and binary labels. We investigate the performance of contrastive mixup relative to the vanilla AP loss function.

**Table 1.** Statistics on downsampled dataset with limited utterances per speaker

Number of speakers	Utterances per speakers	Total utterances	Percentage of VoxCeleb2
5994	2	11,988	1.0%
	3	17,982	1.6%
	5	29,970	2.6%
	10	59,940	5.1%

## 4. EXPERIMENTS

### 4.1. Data Description

Our models are trained using the development set of VoxCeleb2 dataset [23]. VoxCeleb2 dataset contains more than 1 million utterances collected from nearly 6,000 speakers under controlled conditions. In order to obtain a fair comparison with previous results, the model evaluation is performed based on the test set of VoxCeleb1 [26].

### 4.2. Training Details

In our experiments, we employed PyTorch code [23] as a baseline. During training, 2s-long segments are randomly extracted from each utterance. The input feature, log Mel-spectrogram, was extracted every 10 ms using a 25-ms window size. As shown in [23], Fast ResNet34, which has a quarter-size channel compared to the original ResNet-34 [27], was used as a backbone model. The encoded output is aggregated using self-attentive pooling (SAP) [28] to generate utterance-level representations.

The models in this paper were trained using four NVIDIA V100 Tensor core GPUs with distributed configuration. We use the Adam optimizer with a learning rate of 0.001, decreasing by 5% for every 10 epochs. Unless specified elsewhere, all models are trained for 500 epochs using a distributed configuration with four GPUs.

As indicated in [23], large batch size leads to improved performance for metric learning methods due to the ability to sample hard negative samples within the batch. Therefore, we intentionally choose the largest batch size without exceeding the memory limits of the GPUs. For Quarter ResNet-34, the batch size is  $400 \times 2$ , namely the number of speakers  $\times$  the number of utterances per speaker. For fair and reliable comparisons, all experiments are repeated three times, with mean and standard deviation calculated.

In our effort to investigate the effect of contrastive mixup, given that utterances per speaker are limited, we build a small training dataset by randomly sampling a certain number of utterances from each speaker. As shown in Table 1, the reduced training datasets consists of 2, 3, 5, and 10 utterances per speaker, which is significantly smaller compared to the original VoxCeleb2. Then, we compare different training strategies using those down-sampled datasets.

The evaluation metric used in this work is equal error rate (EER) in %, where false acceptance rate (FAR) and false rejection rate (FRR) are closest. To demonstrate the effect of contrastive mixup across different training settings, we conduct experiments and compare the resulting EERs for the following cases:

1. original AP loss without augmentation and mixup;
2. original AP loss plus augmentation without mixup;
3. contrastive mixup without augmentation;
4. contrastive mixup plus augmentation.

**Table 2.** Results with original AP loss ( $L_p$ ) compared to two versions of mixup loss ( $L_{CE,mixup}$  (6) and  $L_{c,mixup}$  (7)).

Type	Loss function	EER (%)	Relative improvement
Baseline	Original AP [23]	$2.21 \pm 0.03$	-
Mix-up	CE (6)	$2.19 \pm 0.01$	+0.90%
	Contrastive (7)	$2.11 \pm 0.02$	+4.52%

Here augmentation refers to noise addition and room impulse response (RIR) application based on the widely-used MUSAN corpus [13], which should be distinguished from the proposed contrastive mixup approach.

### 4.3. Baseline and Evaluation Details

We use the VoxCeleb1 [26] test set to evaluate model performance. Unlike the training stage, we sample ten 4-second temporal crops at regular intervals from each test segment. The similarities between all segment pairs across utterances are computed. The mean value of the similarities is calculated as the final score. A similar protocol can be found in [10].

We focus on two critical aspects. First, how does contrastive mixup compare to the vanilla AP loss, and second, how does contrastive mixup behave with limited training data? To this end, we train the models using different sizes of datasets. Note that all the other settings and hyperparameters are kept constant.

In Tables 2–4, the baseline EER refers to training using original AP loss without augmentation. The mixup EER refers to training using the proposed contrastive-mixup loss without augmentation. The augmentation EER refers to training using original AP loss with augmentation [13]. The mixup + augmentation EER refers to training using contrastive-mixup loss as well as augmentation. In addition, when employing contrastive-mixup loss, we fine-tune the mixing coefficient  $\lambda$  to achieve the optimal performance.

### 4.4. Results and Discussions

In our preliminary analysis, we found that the performance of contrastive mixup is closely dependent on the choice of hyperparameter  $\lambda$ . Therefore, we have explored a range of  $\lambda \sim \text{Beta}(\alpha, \alpha)$ , where  $\alpha$  equals 0.1, 0.2, 0.4, or 0.6, and compared the resulting EERs. Experimental results indicate that when training with the entire VoxCeleb2 dataset, setting  $\alpha$  to be a smaller value (0.1) gives the best performance, whereas setting  $\alpha$  to a larger value (e.g., 0.4) gives the smallest EER when training data is limited. A previous study [15] shows that relatively small values of  $\alpha \in [0.1, 0.4]$  produce the best performance for classification, whereas a large value of  $\alpha$  results in significant under-fitting. Our results agree well with previous observations in different domains [8, 29]. Also, the mixup is performed at the raw waveform level instead of the Mel-spectrum level based on our preliminary results, where the waveform level and the Mel-spectrum level give EERs of  $2.11 \pm 0.02$  and  $2.44 \pm 0.06$ , respectively. Unless specified, the EER and its standard deviation in this paper were computed by repeating the experiments three times.

Table 2 presents the EERs of baseline and two different versions (loss functions) of the mixup algorithm. Given the entire VoxCeleb2 training data set, we observed that mixup implementations improved with a modest gain of 4.52%. Based on this result, in the follow-up experiments, we used contrastive mixup (7).

**Table 3.** Comparison of original AP loss ( $L_p$ ) and contrastive-mixup loss ( $L_{c,mixup}$ ) on small datasets.

Utterances per speaker	Baseline EER (%)	Mixup EER (%)	Relative improvement
2	$14.80 \pm 0.25$	$12.38 \pm 0.24$	+16.3%
3	$12.32 \pm 0.33$	$10.53 \pm 0.12$	+14.5%
5	$9.43 \pm 0.14$	$8.26 \pm 0.07$	+12.4%
10	$6.63 \pm 0.08$	$6.05 \pm 0.14$	+8.7%

**Table 4.** Comparison of mixup on top of online augmentation

Utterances per speaker	Augmentation EER (%)	Mixup + Aug. EER (%)	Relative improvement
2	$11.21 \pm 0.28$	$10.75 \pm 0.15$	+4.1%
3	$9.15 \pm 0.33$	$8.98 \pm 0.21$	+1.8%
5	$7.65 \pm 0.12$	$7.55 \pm 0.03$	+1.3%
10	$5.86 \pm 0.12$	$5.84 \pm 0.07$	+0.3%

Table 3 compares the EERs with limited training data, where the number of utterances per speaker is limited to 2, 3, 5, or 10, to support our hypothesis that mixup improves generalization. The gain with mixup was modest when the full VoxCeleb training data was used, as shown in Table 2. As the number of utterances per speaker is reduced, we observed that contrastive mixup shows larger performance improvements relative to the baseline. When training on only two utterances per speaker, contrastive mixup can reduce the EER by up to 16.3%. This supports our hypothesis that mixup works as a generalization enhancement, especially with limited training data.

In Table 4, contrastive mixup is applied on top of online augmentation. In these experiments, we fine-tune the mixing coefficient by choosing  $\text{Beta}(0.6, 0.6)$  for 2 utterances,  $\text{Beta}(0.2, 0.2)$  for 3 utterances,  $\text{Beta}(0.4, 0.4)$  for 5 utterances, and  $\text{Beta}(0.1, 0.1)$  for 10 utterances. While augmentation improves over the baseline, adding contrastive mixup on top of augmentation gives additional improvements, as shown in Table 4.

As indicated by the evaluation results, while performance improvement is modest when training on the full VoxCeleb dataset, contrastive mixup consistently outperforms the baseline when the number of utterances per speaker is limited. The relative improvement between contrastive mixup and the baseline increases as the number of training utterance is reduced. Thus, our experimental results support the hypothesis that mixup boosts the generalization of neural network models.

## 5. CONCLUSIONS

We propose contrastive mixup, a data augmentation strategy for contrastive representation learning of speaker verification systems. The key contribution of our work is a reformulation of the mixup loss function for metric learning objectives, specifically with angular prototypical loss. We show empirically that contrastive mixup consistently improves the performance of speaker verification models, especially when there are limited number of utterances per training speaker. Moreover, we observe that contrastive mixup can be applied on top of existing augmentation techniques to achieve further performance gains.

## 6. ACKNOWLEDGMENTS

We would like to thank Oguz Elibol, Jasha Droppo and the Alexa SpeakerID team for their helpful feedback and discussions.

## 7. REFERENCES

- [1] V. Gudivada, A. Apon, and J. Ding, “Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations,” *International Journal on Advances in Software*, vol. 10.1, pp. 1–20, 2017.
- [2] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” arxiv:1712.04621, 2017.
- [3] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, “A survey of data augmentation approaches for NLP,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 20*, Aug. 2021, pp. 968–988.
- [4] K. Chaitanya, N. Karani, C. Baumgartner, O. Donati, A. S. Becker, and E. Konukoglu, “Semi-supervised and task-driven data augmentation,” in *Proc. 26th International Conference on Information Processing in Medical Imaging*, 2019.
- [5] S. Wei, S. Zou, F. Liao, and W. Lang, “A comparison on data augmentation methods based on deep learning for audio classification,” *Journal of Physics: Conference Series*, vol. 1453, p. 012085, Jan. 2020.
- [6] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. Interspeech*, Sep. 2019, pp. 2613–2617.
- [7] S. Wu, H. Zhang, G. Valiant, and C. Ré, “On the generalization effects of linear transformations in data augmentation,” in *Proc. International Conference on Machine Learning*, 2020, pp. 10410–10420.
- [8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” in *Proc. International Conference on Learning Representations*, 2018.
- [9] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, “How does Mixup help with robustness and generalization?” in *Proc. ICLR*, 2021.
- [10] J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep speaker recognition,” *Proc. Interspeech*, pp. 1086–1090, Sep. 2018.
- [11] H. S. Heo, B.-J. Lee, J. Huh, and J. S. Chung, “Clova baseline system for the VoxCeleb speaker recognition challenge 2020,” *arXiv preprint arXiv:2009.14153*, 2020.
- [12] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in *Proc. 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp. 4080–4090.
- [13] D. Snyder, G. Chen, and D. Povey, “MUSAN: A music, speech, and noise corpus,” arxiv:1510.08484, 2015.
- [14] J. M. Kates and E. J. Brandewie, “Adding air absorption to simulated room acoustic models,” *The Journal of the Acoustical Society of America*, vol. 148, no. 5, pp. EL408–EL413, 2020.
- [15] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak, “On mixup training: Improved calibration and predictive uncertainty for deep neural networks,” arxiv:1905.11001, 2020.
- [16] L. Carratino, M. Cissé, R. Jenatton, and J.-P. Vert, “On mixup regularization,” arxiv:1905.11001, 2020.
- [17] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, “Manifold mixup: Better representations by interpolating hidden states,” in *Proc. 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., Jun. 2019, pp. 6438–6447.
- [18] C. Summers and M. J. Dinneen, “Improved mixed-example data augmentation,” in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1262–1270.
- [19] H. Guo, Y. Mao, and R. Zhang, “Mixup as locally linear out-of-manifold regularization,” in *Proc. AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3714–3722.
- [20] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6023–6032.
- [21] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2015.
- [22] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *Proc. IEEE ICASSP*, 2018, pp. 4879–4883.
- [23] 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,” in *Proc. Interspeech*, 2020, pp. 2977–2981.
- [24] H. Guo, “Nonlinear mixup: Out-of-manifold data augmentation for text classification,” in *Proc. AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4044–4051.
- [25] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *arXiv preprint arXiv:1905.02249*, 2019.
- [26] A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” *Proc. Interspeech*, pp. 2616–2620, Aug. 2017.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [28] G. Bhattacharya, M. J. Alam, and P. Kenny, “Deep speaker embeddings for short-duration speaker verification,” in *Proc. Interspeech*, 2017, pp. 1517–1521.
- [29] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee, “i-Mix: A domain-agnostic strategy for contrastive representation learning,” in *Proc. ICLR*, Sep. 2021.