Self-Supervised Speaker Verification Using Dynamic Loss-Gate and Label Correction

Bing Han, Zhengyang Chen, Yanmin Qian†

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

{hanbing97, zhengyang.chen, yanminqian}@sjtu.edu.cn

Abstract

For self-supervised speaker verification, the quality of pseudo labels decides the upper bound of the system due to the massive unreliable labels. In this work, we propose dynamic loss-gate and label correction (DLG-LC) to alleviate the performance degradation caused by unreliable estimated labels. In DLG, we adopt Gaussian Mixture Model (GMM) to dynamically model the loss distribution and use the estimated GMM to distinguish the reliable and unreliable labels automatically. Besides, to better utilize the unreliable data instead of dropping them directly, we correct the unreliable label with model predictions. Moreover, we apply the negative-pairs-free DINO framework in our experiments for further improvement. Compared to the best-known speaker verification system with self-supervised learning, our proposed DLG-LC converges faster and achieves 11.45%, 18.35% and 15.16% relative improvement on Vox-O, Vox-E and Vox-H trials of Voxceleb1 evaluation dataset.

Index Terms: self-supervised, speaker verification, dino, dynamic loss-gate, label correction

1. Introduction

Recently, the deep learning based methods have been broadly applied for speaker verification (SV) task and obtained excellent performance [1–4]. However, all these methods require large amounts of training data with speaker labels, while the collection of data with annotated speaker labels is often very difficult and expensive.

In order to make full use of a large quantity of unlabeled data, many efforts [5–11] have been made to obtain good speaker representations in a self-supervised learning manner. Following the iterative framework proposed in [12], the current state-of-the-art self-supervised speaker verification systems usually include two stages. In stage I, a speaker encoder is trained by contrastive learning based loss [9]. In stage II, estimating pseudo labels from the pre-trained model and then training a new model based on the estimated pseudo labels are iteratively performed to continuously improve the performance.

This stage-wise training framework has achieved excellent performance, while there still exist some unreasonable assumptions, which limits the further improvements of system performance. In stage I, contrastive learning based methods assume that speech segments from the different utterances belong to different speakers. When training the network, they will be regarded as negative pairs and be pushed away from each other in the embedding space. There is no doubt that this inaccurate assumption will bring some mistakes, because different utterances may come from the same speaker. In stage II, a clustering

algorithm is adopted to generate the pseudo labels for each utterance based on the previous learned representation and the estimated pseudo labels will be used in the subsequent supervised training. However, [13] shows that there are massive amount of unreliable pseudo labels. Such low-quality labels will confuse and degrade the model during the training, which indicates that finding a way to select high-quality labels is the key to improving the performance [14]. Cai *etal.* [12] applied an aggressive method to purify the pseudo labels based on clustering confidence, but the improvement is minor. Besides, Tao *etal.* [15] observed that the data with lower loss is more reliable than those with unreliable labels. Based on this hypothesis, they set a loss threshold in each iteration to distinguish the reliable labels from unreliable labels and only use the samples with reliable labels to update the network. Although this approach brings further improvement, there is still a lot of room for improvement. The manually set threshold in each iteration is not flexible and the data with unreliable labels is not fully utilized.

To tackle these problems and improve the existing methods, in this work, we introduce DINO (distillation with no labels) [16] in stage I, which is only based on maximizing the similarity between feature distributions of differently augmented segments from the same utterance so it's free from the negative pair issue. Our experiments show that DINO outperforms all previous contrastive learning based methods in Voxceleb dataset [17, 18]. In stage II, we model the loss distribution of all the data using Gaussian Mixture Model (GMM), more specifically, GMM with two components, in which each Gaussian component can represent the loss distribution of the data with reliable labels or the data with unreliable labels. With the estimated GMM, we can automatically obtain the loss threshold to distinguish the two types of labels, which is more flexible than the manually tuned threshold in [15]. Besides, in many semi-supervised learning works [19–21], a pre-trained model is often used to label an unlabeled dataset. Similarly, instead of directly discarding the data with unreliable labels in [15], we leverage the model prediction as an alternate label and use it to correct the unreliable label. We named the above two strategies that we proposed in stage II as *dynamic loss-gate and label correction* (DLG-LC). With DLG-LC, we build a more robust self-supervised speaker verification system and obtain an equal error rate of 1.468% on Voxceleb 1 test set which is the **state-of-the-art** nowadays.

2. Methods

2.1. DINO based Self-Supervised Learning

Inspired by [16], we introduce DINO which doesn't need negative pairs, and the whole framework is shown in Fig. 1. Firstly,

---

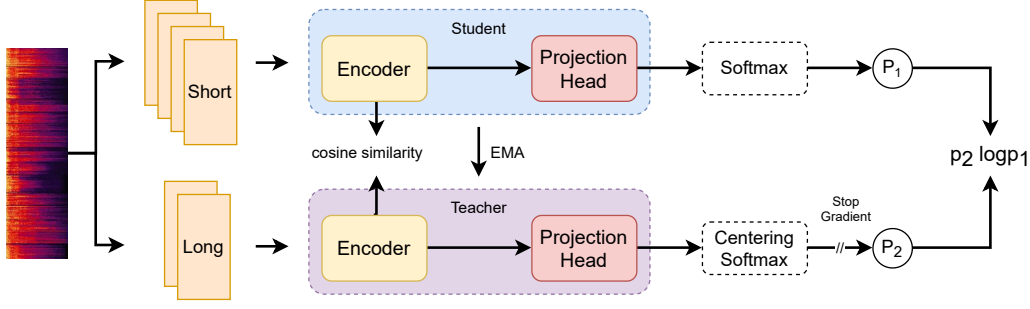†Yanmin Qian is the corresponding author

Figure 1: *Framework of Distillation with no label (DINO) for self-supervised speaker representation learning*

we sample 4 short segments $\{x_1^s \ldots x_4^s\}$ and 2 long segments $\{x_1^l, x_2^l\}$ from a utterance with a multi-crop strategy [22], where the long segments can be used to extract more stable speaker representation. We assume that the segments cropped from the same utterance share the same speaker identity and then augment them in different ways by adding noise or room impulse response for robust performance. In the following, all short segments are passed through the student while only the long segments are passed through the teacher, therefore encouraging *short-to-long* correspondences by minimizing the cross-entropy $H(\cdot)$ between two distributions:

$$L_{ce} = \sum_{x \in \{x_1^l, x_2^l\}} \sum_{x' \in \{x_1^s \ldots x_4^s\}} H(P_t(x) \mid P_s(x')) \quad (1)$$

where P_t and P_s represent the output probability distributions of teacher network g_{θ_t} and student network g_{θ_s} respectively, and we can compute P by normalizing the output with a softmax function:

$$P_s = Softmax(g_{\theta_s}(x)/\epsilon_s) \quad (2)$$

with $\epsilon_s > 0$ a temperature parameter that controls the sharpness of the output distribution, and a similar formula holds for P_t with temperature ϵ_t. In addition, the output of the teacher network is centered with a mean computed over the batch. Sharpening and centering are both adopted for avoiding trivial solution [16].

Both teacher and student have the same architecture but different parameters. The student is updated by gradient descent while the teacher is updated by an exponential moving average (EMA) of the student's parameters. The update rule is $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$, with λ following a cosine schedule from 0.996 to 1 during training. We extract speaker embeddings by Encoder and feed them into Projection Head which consists of a 3-layer MLP with hidden dimension 2048 followed by ℓ_2 normalization and a weight normalized fully connected layer with K dimensions, whose architecture is similar to [22].

In order to make DINO more suitable for speaker verification, we also add a consistency loss to maximize the cosine similarity between the embeddings from the same utterance. This loss is design to ensure that the speaker is embedded in cosine space which is more suitable for scoring and clustering in the following. Then, the final loss is jointed by coefficient α:

$$L_{dino} = L_{ce} + \alpha \sum_{e \in \{e_1^l, e_2^l\}} \sum_{e' \in \{e_1^s \ldots e_4^s\}} (1 - \frac{e \cdot e'}{\|e\| \|e'\|}) \quad (3)$$

where e denotes the speaker embedding extracted by encoder.

2.2. Dynamic Loss-Gate

Following the iteration framework proposed in [12], we use the speaker encoder pre-trained by DINO to extract the speaker embeddings for each utterance. Then, we apply k-means clustering algorithm to assign the same pseudo labels to the utterances which belong to the same cluster. Based on these pseudo labels, a new speaker encoder can be trained to generate more robust speaker embeddings. This process will be repeated several iterations until the model converges. In this process, data with unreliable labels will undoubtedly degrade the performance of the model. In [15], they observed a distinct separation between the loss of the data with reliable and unreliable labels in the final converged stage and proposed a loss-gated learning (LGL) method to effectively select the reliable data using a fixed threshold. With LGL, it has achieved amazing results, but the threshold setting of this method is very dependent on human experience, and the unreliable data are not fully utilized.

In order to obtain an appropriate threshold, we implement the LGL and conduct an experiment to analyze the loss distribution on Voxceleb 2 [18]. The histogram is shown in Figure 2. From the loss distribution, it is obvious to find that the distribution has two sharp peaks. Similar experiments in [23] have shown that these two peaks can represent the data with reliable and unreliable labels respectively. If we can find a way to model this distribution, then the threshold can be calculated directly and dynamically varies with the loss distribution, which can also avoid laborious manual tuning.
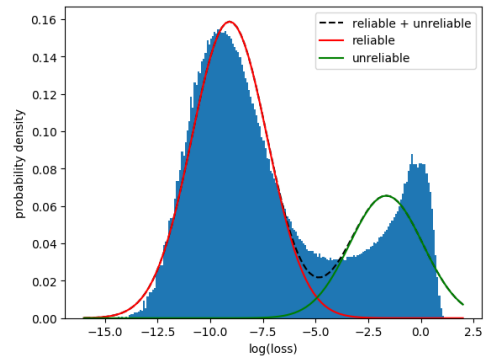

Figure 2: *Loss distribution of LGL on Voxceleb 2. Loss value is scaled by log function. And the lines are estimated by GMM.*

Gaussian distribution is an important type of continuous probability distribution for a real-valued random variable. And

the general form of its probability density function is in Eq.4.

$$\mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2) \qquad (4)$$

where μ is the location parameter and σ is the scale parameter. It is bell-shaped, with low on both sides and high in the middle, whose shape is very similar to "peaks" of loss in Figure 2. In this case, Gaussian Mixture Model (GMM) with 2 components can be employed to fit the distribution of reliable and unreliable samples respectively:

$$p(x) = \lambda_1 \mathcal{N}\left(\mu_1, \sigma_2^2\right) + \lambda_2 \mathcal{N}\left(\mu_2, \sigma_2^2\right) \qquad (5)$$

where λ is the weight for each Gaussian component. We plot the fitted curves in Figure 2, it's obvious that the two "peaks" can be approached by two weighted Gaussian distributions. Then we can easily obtain a threshold τ to distinguish the reliable and unreliable data by finding a loss value that the probabilities belonging to both components are equal:

$$\tau : p_1(\tau) = p_2(\tau) \qquad (6)$$

where $p_1(x) = \lambda_1 \mathcal{N}\left(\mu_1, \sigma_1^2\right)$ and $p_2(x) = \lambda_2 \mathcal{N}\left(\mu_2, \sigma_2^2\right)$. For each epoch, we will re-estimate the GMM based on the recorded loss values in that epoch, so τ can be adjusted dynamically according to the current training condition.

Similar to the LGL, we introduce this threshold τ into the speaker classification loss function ArcMargin Softmax (AAM) [24] to filter the data with small loss and only use the retained data to update the parameters of the network.

$$L_{DLG} = \sum_{i=1}^{N} \mathbb{1}_{l_i < \tau} \log \frac{e^{s(\cos(\theta_{y_i}, i+m))}}{Z} \qquad (7)$$

where $Z = e^{s(\cos(\theta_{y_i}, i+m))} + \sum_{j=1, j\neq i}^{c} e^{s(\cos(\theta_{y_i}, i))}$, $\theta_{j,i}$ is the angle between the column vector W_j and embedding x_i. s is the scaling factor and m is hyperparameter to control the margin. AAM can enforce larger gaps between the nearest speakers and is widely used in speaker recognition tasks.

2.3. Label Correction

To effectively utilize the unreliable data with large losses instead of directly throwing them away, we also propose label correction (LC) to correct pseudo labels dynamically. As the model is continuously optimized, the output of the model for unreliable samples can reflect their true labels to some extent [25]. To leverage this ability, we assume that the model's output prediction is more reliable than pseudo labels which are generated by clustering. Therefore, we incorporate the predicted posterior probability, as the target labels, into the objective function to prevent the model from fitting into incorrect labels. However, not all prediction labels are suitable for training. Inspired by [20, 26], we assume that a prediction label owns high confidence if the model assigns a high probability to one of the possible classes. Therefore, we introduce another threshold τ_2 to retain prediction labels whose largest class probability is above τ_2. And the label correction loss is defined as the following:

$$L_{LC} = \sum_{i=1}^{N} \mathbb{1}_{l_i > \tau, \max(\hat{p_i}) > \tau_2} H(\hat{p_i} \mid p_i) \qquad (8)$$

where p_i and $\hat{p_i}$ represent the output probability of augmented segments and their corresponding clean version respectively.

$H(\cdot)$ denotes the cross-entropy between two probability distributions. We also apply a sharpening operation on $\hat{p_i}$ to encourage a peaky distribution with ϵ_c which is defined in Eq. 2.

Finally, these two losses are combined to optimize the speaker model.

$$L = L_{DLG} + L_{LC} \qquad (9)$$

3. Experiment

3.1. Dataset and Configuration

In our experiments, we adopt the development set of Voxceleb2 [18] for training the networks and no speaker label is used during this process. It comprises 1,092,009 utterances among 5,994 speakers. For the evaluation, we report the experimental results on 3 trial sets as defined in [17]: the Original, Extended, and Hard Voxceleb test sets. The main metrics adopted in this paper are equal error rate (EER) and normalized minimum detection cost function (minDCF).

For both two stages in our experiments, we use 80-dimensional filter bank with 25ms windows and 10ms shift as the acoustic features. In addition, online data augmentation [27] is applied with noisy MUSAN [28] and RIRs [29] datasets.

3.1.1. Stage I : DINO

For DINO, considering the training time and memory limitation, we adopt a thin version ResNet [30] to learn speaker representation from the long (3 seconds) and short (2 seconds) segments, and then they are encoded into 256-dimensional speaker embedding by networks. Similar to the configuration in [16], the K in the DINO projection head is set 65536. Temperatures for the teacher ϵ_t and the student ϵ_s are 0.04 and 0.1 respectively. In addition, we set α to 0.001 to balance two losses during the training process.

3.1.2. Stage II : Iterative Pseudo Label Learning

In this stage, to have a better comparison with the results in [15], we use ECAPA-TDNN [31] as our encoder. For clustering, we adopt the k-means algorithm to assign the pseudo label to the training set which is supported by faiss library [32]. To verify the robustness of our method, we choose 7500 instead of 6000 [12] as the cluster number. When training the network on pseudo labels with DLG, we set the margin as 0.2, scale as 32 in the AAM. For LC, we set sharpen parameters ϵ_c as 0.1 and threshold τ_2 as 0.5. The learning rate decays from 0.1 to $5e$-5 exponentially with SGD optimizer. Our training process repeats 5 iterations. In the last iteration, we extend the channel size of the encoder (ECAPA-TDNN) to 1024 following [15].

3.2. Comparison of DINO with Previous Work

To throughly evaluate the DINO, we conduct experiments and corresponding results are listed in Table 1. All the methods are trained on Voxceleb 2 and evaluated on Vox-O test set. From the table, we can find that negative-pairs-free DINO outperforms all previous contrastive-based methods, which show that negative pairs are indeed a bottleneck for performance improvement. In addition, we also provide the ablation study of DINO at the bottom of Table 1. We can see that DINO without EMA obtains a very bad result which shows that EMA is the key to preventing the model from collapse. When we add multi-crop and cosine strategies during training respectively, both of them can improve the DINO performance in different degrees and demonstrate their necessity.

Table 1: *Comparison of DINO with other self-supervised speaker verification work. EER (%) and minDCF are evaluated on Vox-O test set.*

Method	EER (%)	minDCF
Disent [5]	22.09	-
CDDL [6]	17.52	-
GCL [7]	15.26	-
i-vector [8]	15.28	0.63 (p=0.05)
AP + AAT [8]	8.65	0.45 (p=0.05)
SimCLR + uniform [9]	8.28	0.610
MoCo + WavAug [10]	8.23	0.590
Unif+CEL [11]	8.01	-
DINO	31.23	0.990
+ EMA	7.02	0.579
+ + Multi-Crop	6.35	0.566
+ + + Cosine loss	**6.16**	**0.524**

3.3. Dynamic Loss-gate with Label Correction

Table 2: *EER (%) comparison on Vox-O, E, H of the proposed DLG-LC in Iteration 1. In this experiment, pseudo labels are estimated from our pre-trained DINO system. DINO means we simply used all the data with the estimated pseudo labels as the supervisory signal without any loss-gate during system training.*

Method	Vox-O	Vox-E	Vox-H
DINO	4.255	4.900	8.005
+ LGL [15]	3.590	4.373	6.935
+ DLG	3.202	3.525	5.805
++ LC	**2.723**	**3.179**	**5.442**

Based on pseudo labels generated by pre-trained DINO in stage I, we conducted some experiments to illustrate the effectiveness of our proposed method. Firstly, we give an ablation study of DLG-LC in Iteration 1. From Table 2, it can be observed that the LGL can bring significant improvement compared with the baseline which is trained without any data selection. It means that loss-gate can effectively filter the reliable labels which are of benefit to the model. However, the choice of threshold also has a non-negligible impact on model performance [15]. Based on the estimated GMM, our proposed DLG can dynamically adjust the threshold according to the current training situation and obtains better performance than LGL which adopt a fixed threshold during the whole training process. In addition, we add the LC strategy to make full use of data with unreliable labels and the results are further improved.

Then, we summarize the performance in each iteration with DLG-LC on Vox-O set and the results are presented in Table 3. We first compare the iterative results of generating pseudo labels using DINO and SimCLR respectively. And they are both trained on all training data without any loss-gate strategies. Obviously, DINO owns consistent advantages over SimCLR in each iteration which demonstrates DINO can produce more discriminative speaker embedding for generating more reliable pseudo labels. After applying the loss-gate policy, results have been significantly improved. Our proposed DLG-LC with dynamic threshold shows great advantages compared with LGL. It not only has superior performance but also has a faster convergence speed. Only with 3 iterations, DLG-LC has achieved

Table 3: *EER (%) comparison on Vox-O test set for different iterations of the proposed DLG-LC with other strategies. SimCLR and DINO mean that we simply used all the estimated pseudo labels without any loss-gate during training process.*

Method	SimCLR [9]	DINO	LGL [15]	DLG-LC
Loss-Gate	×	×	✓	✓
Iter-1	6.281	4.255	3.520	**2.723**
Iter-2	5.914	3.946	2.410	**1.888**
Iter-3	5.547	3.824	2.070	**1.670**
Iter-4	4.872	3.691	1.950	**1.495**
Iter-5	4.484	3.510	1.660	**1.468**

comparable results with the final iteration of LGL, which is benefit from the dynamic threshold and label correction.

Finally, a performance comparison with other self-supervised speaker verification systems and our proposed DLG-LC is given in Table 4. Most of them are from the latest Voxceleb speaker recognition challenge (VoxSRC) [33, 34] which represent the most advanced performance nowadays. From the table, it's obviously observed that DLG-LC exceeds all the existing methods and outperforms the best one by relative 11.45%, 18.35% and 15.16% on Vox-O, Vox-E and Vox-H sets respectively with only 5 iterations. Moreover, the clustering algorithm we adopted is k-means which is simpler than agglomerative hierarchical clustering (AHC) used in other systems. Besides, we employ 7,500 instead of 6,000 as the number of clusters because 5,994 is the real number of speakers in the training set. Nevertheless, our system still achieves the **state-of-the-art** performance, which also shows the robustness of our system.

Table 4: *EER (%) comparison for different self-supervised speaker verification methods on Vox-O, E, H*

Method	Clustering	Iter	Vox-O	Vox-E	Vox-H
ID [35]	AHC(7500)	7	2.10	-	-
JHU [36]	AHC(7500)	5	1.89	-	-
DKU [37]	K-M(6000)	4	1.81	-	-
SNU [38]	AHC(7500)	5	1.66	-	-
LGL [15]	K-M(6000)	5	1.66	2.18	3.76
DLG-LC	K-M(7500)	5	**1.47**	**1.78**	**3.19**

4. Conclusions

In this work, we propose dynamic loss-gate and label correction (DLG-LC) to alleviate the performance degradation of self-supervised speaker verification caused by inaccurate assumptions and labels. By modeling the loss histogram with Gaussian distribution, we can obtain a dynamic loss-gate to select reliable data when training on pseudo labels. Instead of dropping unreliable data, we incorporate a predicted posterior probability as the target distribution to prevent fitting into incorrect samples. The experiments on Voxceleb show that our proposed DLG-LC is more robust and achieves the **state-of-the-art** performance.

5. Acknowledgements

6. References

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. IEEE ICASSP*. IEEE, 2018, pp. 5329–5333.

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

[3] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," *Proc. ISCA Interspeech*, pp. 941–945, 2020.

[4] Z. Chen and S. Wang and Y. Qian, "Adversarial Domain Adaptation for Speaker Verification Using Partially Shared Network," in *Proc. ISCA Interspeech*, 2020, pp. 3017–3021.

[5] A. Nagrani, J. S. Chung, S. Albanie, and A. Zisserman, "Disentangled speech embeddings using cross-modal self-supervision," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 6829–6833.

[6] S.-W. Chung, H. G. Kang, and J. S. Chung, "Seeing voices and hearing voices: learning discriminative embeddings using cross-modal self-supervision," *arXiv preprint arXiv:2004.14326*, 2020.

[7] N. Inoue and K. Goto, "Semi-supervised contrastive learning with generalized contrastive loss and its application to speaker recognition," in *Proc. IEEE APSIPA ASC*. IEEE, 2020, pp. 1641–1646.

[8] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. S. Chung, "Augmentation adversarial training for unsupervised speaker recognition," *arXiv preprint arXiv:2007.12085*, 2020.

[9] H. Zhang, Y. Zou, and H. Wang, "Contrastive self-supervised learning for text-independent speaker verification," in *Proc. IEEE ICASSP*. IEEE, 2021, pp. 6713–6717.

[10] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive learning," in *Proc. IEEE ICASSP*. IEEE, 2021, pp. 6723–6727.

[11] S. H. Mun, W. H. Kang, M. H. Han, and N. S. Kim, "Unsupervised representation learning for speaker recognition via contrastive equilibrium learning," *arXiv preprint arXiv:2010.11433*, 2020.

[12] D. Cai, W. Wang, and M. Li, "An iterative framework for self-supervised deep speaker representation learning," in *Proc. IEEE ICASSP*. IEEE, 2021, pp. 6728–6732.

[13] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. ECCV*, 2018, pp. 132–149.

[14] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," *arXiv preprint arXiv:2101.06329*, 2021.

[15] R. Tao, K. A. Lee, R. K. Das, V. Hautamäki, and H. Li, "Self-supervised speaker recognition with loss-gated learning," *arXiv preprint arXiv:2110.03869*, 2021.

[16] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. ICCV*, 2021, pp. 9650–9660.

[17] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. ISCA Interspeech*, 2017, pp. 2616–2620.

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. ISCA Interspeech*, 2018, pp. 1086–1090.

[19] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[20] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Proc. NIPS*, vol. 33, pp. 596–608, 2020.

[21] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Proc. NIPS*, vol. 34, 2021.

[22] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. NIPS*, 2020.

[23] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *Proc. ICML*. PMLR, 2019, pp. 312–321.

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

[25] F. Tong, Y. Liu, S. Li, J. Wang, L. Li, and Q. Hong, "Automatic Error Correction for Speaker Embedding Learning with Noisy Labels," in *Proc. ISCA Interspeech*, 2021, pp. 4628–4632.

[26] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. ICML*, vol. 3, no. 2, 2013, p. 896.

[27] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE/ACM Trans. ASLP.*, vol. 28, pp. 1038–1051, 2020.

[28] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE ICASSP*. IEEE, 2017, pp. 5220–5224.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[32] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[33] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "Voxsrc 2020: The second voxceleb speaker recognition challenge," *arXiv preprint arXiv:2012.06867*, 2020.

[34] A. Brown, J. Huh, J. S. Chung, A. Nagrani, and A. Zisserman, "Voxsrc 2021: The third voxceleb speaker recognition challenge," *arXiv preprint arXiv:2201.04583*, 2022.

[35] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxceleb speaker recognition challenge 2020 system description," *arXiv preprint arXiv:2010.12468*, 2020.

[36] J. Cho, J. Villalba, and N. Dehak, "The jhu submission to voxsrc-21: Track 3," *arXiv preprint arXiv:2109.13425*, 2021.

[37] D. Cai and M. Li, "The dku-dukeece system for the self-supervision speaker verification task of the 2021 voxceleb speaker recognition challenge," *arXiv preprint arXiv:2109.02853*, 2021.

[38] S. H. Mun, M. H. Han, and N. S. Kim, "Snu-hil system for the voxceleb speaker recognition challenge 2021."