2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) | 979-8-3503-0261-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSPW59220.2023.10192957

IMPROVING DINO-BASED SELF-SUPERVISED SPEAKER VERIFICATION WITH PROGRESSIVE CLUSTER-AWARE TRAINING

Bing Han, Wen Huang, 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

ABSTRACT

Self-supervised contrastive learning has recently emerged as one of the promising approaches in speaker verification task, due to its independence from labeled data. Among them, the DINO-based self-supervised framework, trained without exploiting negative pairs, is very popular and achieves excellent performance in the speaker verification task. However, limited by the duration of utterance, there exist many overlaps which may mislead the model to pay attention to irrelevant information. To tackle this problem, we propose a cluster-aware (CA) training strategy to make the model crop positive segments from several utterances in the same cluster rather than from a single utterance. Besides, in the clustering stage, we also investigate strategies of fixed number clustering as well as progressive clustering. With these strategies, our CA-DINO achieves the state-of-the-art result on Vox-O test set. Finally, we explore the effect of fine-tuning CA-DINO with a small amount of labeled data. Our proposed model with only 10% labeled data outperforms the fully supervised system trained on all data.

***Index Terms*—** speaker verification, self-supervised, dino, cluster-aware, progressive clustering

1. INTRODUCTION

Speaker verification (SV) is a task that verifies a person's identity based on the features of their voice. Deep learning-based methods have thrived in recent years and achieved excellent performance in speaker verification tasks. To achieve better performance and robustness, researchers have designed various model architecture [1–3], training objection [4, 5], pooling methods [6, 7] for speaker verification task. However, these deep learning-based methods are usually based on the fully-supervised training manner, which requires massive well-labeled data. Nevertheless, collecting well-labeled data at scale is difficult and expensive, while unlabeled data is relatively easy to collect in large quantities.

In this case, to fully utilize these unlabeled data and reduce the dependence on labeled data, many researchers turn their attention to self-supervised learning which obtains supervisory signals from the data itself and designs a pretext task to help the model learn the representation. Firstly, with the help of the text-to-speech (TTS) task, a generative method has been proposed in [8] to separate speaker representation based on phone information. Although no speaker annotations are used here, the performance is not ideal. Then, inspired by the success of the frame-level pre-trained models in ASR such as Wav2Vec series [9], Hubert [10] and so on, some researchers [11, 12] explored fine-tuning them on speaker verification task directly but it will bring huge parameters comparing with traditional models. In the following, by observing the data structure, a hypothesis is proposed to assume that speech segments truncated from the same utterances belong to the same speaker while those from different utterances belong to different speakers [13]. Based on this hypothesis, many efforts adopt contrastive learning to obtain discriminative speaker representations by maximizing and minimizing the positive and negative pairs respectively [13–16]. Then, to tackle the problem of negative pairs caused by inaccurate assumption, a non-contrastive framework DINO (distillation with no labels) [17] is introduced to speaker verification [18–22] and brought a huge performance improvement. For traditional DINO, the two distributions of positive pairs are minimized by cross-entropy where the positive pair is formed with several segments sampled from one utterance. Due to the short duration of each utterance, there are a lot of overlaps among these segments which might mislead the model to focus on irrelevant information (content, channel, and so on) and neglect the speaker information in the audio.

To tackle this problem, we propose several new strategies for self-supervised learning in speaker verification tasks. First, we adopt the traditional DINO framework as the initial model in the first pre-training stage. Next, we propose a cluster-aware (CA) training strategy for DINO which samples positive segments from the same cluster generated by the clustering algorithm. This strategy can minimize channel and context effects and increase the diversity of data. Besides, inspired by the clustering work [23], we also explore the progressive cluster aware strategy in the clustering stage, which works in adapting to network convergence and preventing the contamination of pseudo-labels. With these strategies, our progressive CA-DINO achieves the **state-of-the-art** performance on Voxceleb [24] evaluation set. In addition, we also conduct fine-tune experiments to examine our proposed CA-DINO with only a small amount of labeled data. Compared with another self-supervised model such as SimCLR [14] and the fully supervised model, it outperforms all of them with only 10% labeled data.

2. METHODS

2.1. DINO-based Self-Supervised Speaker Verification

In this section, we will give a description of DINO and the whole framework is shown in Fig.1. DINO follows an architecture similar to knowledge distillation, consisting of not only a *student* encoder but also a momentum *teacher* encoder. Both encoders are trained in

†Corresponding Author.

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

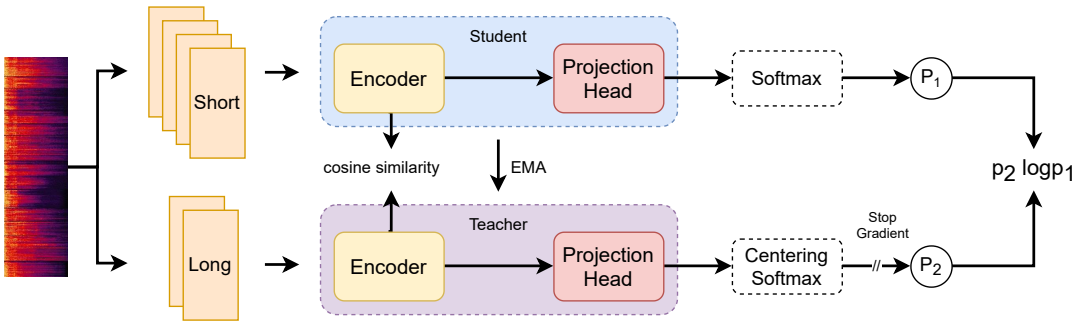

Fig. 1. Framework of distillation with no label (DINO) for self-supervised speaker representation learning

parallel. The outputs of the *teacher* encoder are used as ground truth to optimize the *student* encoder.

Similar to [17], different views of each utterance are constructed with the multi-crop strategy. More precisely, from a given utterance, we randomly sample 4 short $\{x_1^s, x_2^s, x_3^s, x_4^s\}$ and 2 long segments $\{x_1^l, x_2^l\}$, these segments should overlap as little as possible. As in the previous work [13–16], the assumption that segments cropped from the same utterance belong to the same speaker is still followed. Then, we apply different kinds of data augmentation on these crops by adding noise or reverberation for robust performance. After augmentation, all segments are passed through the *student* while only the long segments are passed through the *teacher*.

The *teacher* and *student* own the same architecture but with different parameters due to the different update methods. The *student* is updated by gradient descent while the *teacher* is updated by the exponential moving average (EMA) of the *student*'s parameters. EMA's update rule is $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$. where λ is adjusted by a cosine scheduler from 0.996 to 1 during training. Speaker embeddings are extracted by Encoder and then fed into the Projection Head, which contains a 3-layers perceptron with hidden dimension 2048 followed by ℓ_2 normalization and a weight normalized fully connected layer with K dimensions.

We encourage the short-to-long correspondences by minimizing the cross-entropy loss $H(\cdot)$ between two distributions as the following Equation.1:

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

where output distributions of momentum *teacher* network f_{θ_t} and *student* network f_{θ_s} are denoted by P_t and P_s respectively. Moreover, P can be computed by using a softmax function to normalize the output:

$$P_s(x) = Softmax(\frac{f_{\theta_s}(x)}{\epsilon_s}) \quad (2)$$

where $\epsilon_s > 0$ is a temperature parameter that can control the sharpness of the output distribution. Similarly, there is a formula holds for P_t with temperature $\epsilon_t > 0$, too. Moreover, a mean computed over batches is used for centering *teacher* model's output distribution. During the training, both sharpening and centering are applied to avoid trivial solution [17].

Additionally, we add a cosine-based consistency loss to ensure that the speaker embedding is encoded into cosine space which is more suitable for later scoring and clustering. It maximizes the cosine similarity among the embeddings extracted from the same speaker. Finally, the total loss is summarized with coefficient α:

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

where e represents the extracted speaker embedding from encoder.

2.2. Progressive-Cluster-Aware Training strategy

For traditional DINO, the positive pairs are formed by the segments sampled from the same utterance. As mentioned above, the cross-entropy between the two distributions of positive pairs is minimized during the optimization of DINO to encourage short-to-long correspondence. Although we try to make these segments from the same utterance overlap as little as possible when sampling, in practice these segments usually overlap to a large extent due to the limited duration of the utterance. Under the influence of these overlapping parts, the model might pay more attention to the content, channel and other irrelevant information of the overlapping parts, while ignoring the speaker information in the audio. Although we will also add different types of data augmentation to them later, the data still lacks diversity, which may lead the model to optimize in the wrong direction.

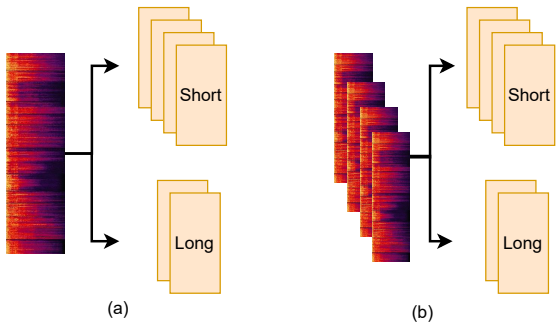

Fig. 2. Difference between traditional DINO and cluster-aware training DINO. (a) Traditional DINO: long and short segments are sampled from the same utterance to compose the positive pairs. (b) Cluster-aware training DINO: through a simple clustering algorithm, we consider that the same speaker in the same cluster shares the same identity and segments are cropped from the corresponding cluster.

In order to reduce the overlaps of segments and increase the diversity of data, we propose a clustering-aware (CA) training strategy for DINO while maintaining the original assumptions as much as possible, which is named CA-DINO in the following. The model

training is divided into two stages. In the first stage of training, we optimize the model according to the traditional DINO training manner. Then the training process will enter the next stage when the model is able to extract discriminative speaker representation. In the following, clustering algorithm such as k-means is adopted here to cluster the extracted speaker embeddings. With the assumption that the utterances in the same cluster belong to the same speaker, these clustered utterances can be used to generate crops with fewer overlapped parts and more diversity. As shown in Fig. 2, unlike the traditional DINO strategy, the current positive pairs are sampled from several different utterances in the same cluster rather than from a single utterance. These positive pairs come from the same speaker but have different content and channel information, which significantly enhances the diversity of data and reduces overlapping parts so that the model can pay more attention to speaker information rather than irrelevant information. These positive pairs will be used as new inputs for subsequent model training. Considering the resource consumption of extracting speaker embeddings, the clustering process will be performed after several training epochs.

In addition, in the clustering step, we also introduce the Progressive Clustering (PC) method. In the early stage of representation learning, setting a small number of clusters during clustering may lead to class-inconsistent samples within some clusters, resulting in the contamination of pseudo labels, further hindering the growth of the model's representation ability. As the network converges, we can gradually lessen the number of clusters to reduce intra-class distance and make the feature space more compact and class-consistent. Specifically, two strategies for reducing the number of clusters are employed: linear decline and logarithmic decline, which are called PC-Linear and PC-Log respectively in the following. To represent the strategy of PC-Log, we note that the initial number of clusters is N_i, the final fixed number of clusters is N_f, the number of clusters in the t-th epoch is N_t, which is formulated as:

$$\log(N_t) = \max((1 - \frac{t}{T})\log(N_i), N_f) \tag{4}$$

where T denotes the total training epochs. As shown in Eq. 4, N_t declines fast in early epochs and slower in later epochs. When $t = 0$, N_t equals to N_i; when $t > t_0$, we fix N_t as N_f.

3. EXPERIMENT SETUP

3.1. Cluster-Aware DINO

3.1.1. DINO

For DINO, following [19,20], we adopt ECAPA-TDNN [2] with 512 channels as the audio encoder to learn discriminative speaker representation, which is a time-delay neural network (TDNN) [1] based backbone with emphasized channel attention, propagation, and aggregation.

The development set of Voxceleb 2 [24] is adopted for training the networks without using any speaker labels, following [18–20]. The training set comprises $1,091,251$ utterances among $5,994$ speakers collected from YouTube. For each utterance, two long (3 seconds) and four short (2 seconds) segments are randomly cropped and regarded as positive pairs. The extracted segments are augmented with MUSAN [25] and RIR[1]. After that, they are encoded into 192-dimensional speaker embeddings by the encoder. Similar to the configuration in [17], the K in the DINO projection head is set as $65,536$. Temperatures for the teacher ϵ_t and the student ϵ_s are 0.04 and 0.1 respectively. In addition, we set cosine loss weight α as 1.0 to balance the two losses. The whole training process will last 150 epochs. Model parameters are updated using stochastic gradient descent (SGD) algorithm with weight decay $5e$-5. The learning rate is linearly ramped up from 0 to 0.2 in the first 20 epochs, and then it decays to $1e$-5 with cosine scheduler. Moreover, the momentum also follows the cosine schedule from 0.996 to 1.0.

[1] https://www.openslr.org/28

3.1.2. Cluster-Aware Training

We train the model generally as described in DINO in the first 90 epochs for the cluster-aware training strategy. After that, clustering algorithm is applied on the whole training set every 5 epochs. Considering the time complexity and the amount of training data, we only utilize k-means here which requires few extra computation.

Our models are evaluated on 3 trials as defined in [24]: the Original, Extended, and Hard Voxceleb test sets. **Vox-O** is the original test set of Voxceleb 1 contains $37,720$ trials from 40 speakers. **Vox-E** is an trial list which (using the entire dataset) contains $581,480$ trials from 1251 speakers. **Vox-H** is a hard evaluation list consisting of $552,536$ pairs sampled from 1190 speakers in Voxceleb 1, all of which are from the same nationality and gender.

3.2. Fine-tuning pre-trained self-supervised model

Fine-tuning experiment is conducted on in-domain Voxceleb 1 [26] and out-of-domain CN-celeb 1 [27] to prove the robustness of our model. The dev set of Voxceleb 1 consists of $148,642$ utterances from $1,211$ speakers. And CN-celeb 1 contains $53,288$ (we concatenate the short utterances from the same genre and same speaker to make them longer because there exists many short utterances less than 2s) from 800 speakers.

In the fine-tuning phase, we use 2s training segments. Additive angular margin (AAM) [4] loss is used here to optimize the model. And we set the margin and scale of AAM to 0.2 and 32.0 respectively. The fine-tune process will last 100 epochs with learning rate decrease from initial 0.01 to final $1e$-5 exponentially.

4. RESULTS

4.1. Comparison with other self-supervised models

Table 1 reports the speaker verification performance of our proposed methods and other previous self-supervised models. All the methods are trained on Voxceleb 2 without any speaker label and evaluated on the Vox-O test set following the setup of previous works. From the table, we can find out that negative-pairs-free DINO outperforms all previous traditional methods [28–30] and contrastive-based methods [13–16], which prove 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 observe that DINO without exponential moving average (EMA) obtains an abysmal result, which reveals that EMA is indispensable to prevent the model from collapsing. After applying the progressive cluster-aware (CA) strategy when training DINO, the performance has been further improved. It outperforms the best system [20] by relative **23.74%** on Vox-O test set, which is a great performance leap.

We also provide the experiment to explore the effects of the different numbers of clusters and different decline strategies on performance. And the results are reported in Table 2. Compared with the baseline system (1080k), it can be found that the cluster-aware strategy can improve the Normalized Mutual Information (NMI) effectively, and show its robustness to the number of clusters because it

Table 1. Performance comparison of the proposed CA-DINO with other self-supervised speaker verification methods. SSL means Self-Supervised Learning. EER (%) and minDCF (p=0.05) are evaluated on Vox-O test set.

SSL Methods	EER (%)	$\text{minDCF}_{0.05}$
Disent [28]	22.090	-
CDDL [29]	17.520	-
GCL [30]	15.260	-
i-vector [13]	15.280	0.630
AP + AAT [13]	8.650	0.450
SimCLR + Uniform [14]	8.280	*0.610
MoCo + WavAug [15]	8.230	*0.590
Unif+CEL [16]	8.010	-
DINO [18]	6.160	*0.524
DINO [19]	4.830	*0.463
DINO + Curriculum Learning [20]	4.470	0.306
DINO	31.233	0.989
+ EMA	4.221	0.299
+ + Cluster Aware (CA)	3.536	0.247
+ + + Progressive Cluster (PC)	**3.409**	**0.232**

* The minDCF are given with p=0.01.

Table 2. Performance comparison of cluster-aware training with different cluster numbers. EER (%) is evaluated on Vox-O, Vox-E and Vox-H test sets. 1080k here means that one utterance is one class, which is equivalent to training without the cluster-aware strategy.

# Cluster	NMI	Vox-O	Vox-E	Vox-H
1080k	0.753	4.221	4.508	7.614
30k	0.891	3.680	3.816	6.784
20k	0.902	3.536	3.779	6.681
10k	0.912	3.547	3.726	6.604
5k	0.898	3.546	3.776	6.713
PC-Linear (from 30k to 5k)	-	**3.404**	3.699	6.575
PC-Log (from 30k to 5k)	-	3.409	**3.642**	**6.541**

can bring significant and stable improvement for all fixed clustering numbers. Meanwhile, CA-DINO with progressive clustering (PC) outperforms other systems with a fixed number of clusters, showing that progressive clustering can improve performance to a certain extent.

4.2. Fine-tuning CA-DINO with a few Labeled Data

In order to better illustrate the superior performance of CA-DINO, experiments of self-supervised learning with the pretrain-finetune framework are conducted. We fine-tune the self-supervised model with different amounts of labeled data in the downstream speaker verification task. We split the dev set of Voxceleb 1 in different proportions 10%/20%/50%/100% from 'the number of speakers' or 'the number of utterances for each speaker' respectively.

As shown in Table 3, frame-level pre-trained model Wav2Vec only obtains unpromising results which is reasonable because it is designed for speech recognition, not speaker task. Then, we can observe that self-supervised models SimCLR and our proposed CA-DINO show significant improvements compared to the model training from scratch. This suggests that pre-trained models with better initialization are important under low-resource conditions. Moreover, the proposed CA-DINO performs significantly better than SimCLR, and CA-DINO still performs well with only a small amount of labeled data. The reduction of labeled data does not cause tremendous performance degradation. More promisingly, with only 10% labeled utterances, finetuning with pretrained CA-DINO even achieves a better result than the fully supervised system trained on all labeled data, i.e., 2.510% vs. 2.755%.

Table 3. EER(%) comparison of finetuning the pre-trained self-supervised model with different amounts of labeled data from Voxceleb 1. Results are evaluated on Vox-O which is the test set of Voxceleb 1.

Initial	Random	SimCLR	Wav2Vec [11]	CA-DINO
None	32.78	8.547	15.62	**3.409**
10% utts	6.893	4.388	-	**2.510**
10% spks	8.595	6.481	-	**4.620**
20% utts	5.276	3.797	-	**2.408**
20% spks	6.529	5.412	-	**3.329**
50% utts	3.691	3.266	-	**2.111**
50% spks	3.643	3.649	-	**2.626**
100%	2.755	2.936	3.61	**1.930**

In addition, from Table 3, we can also find that when the sampling ratio is the same, the training performance of using a small number of utterances for each speaker is better than using a small number of speakers with all utterances. This discovery also gives us a new idea for collecting data in the case of limited resources. It seems better to collect data from different speakers as much as possible, rather than collecting utterances from each speaker as much as possible, which is very meaningful to economize lots of manual annotation.

Table 4. EER(%) and minDCF(p=0.01) comparison of finetuning the self-supervised model with out-of-domain dataset CN-celeb 1. Results are evaluated on the evaluation set of CN-celeb 1.

Initial	EER(%)	$\text{minDCF}_{0.01}$
Random	12.076	0.6162
SimCLR [14]	10.120	0.5681
DINO + Curriculum Learning [20]	10.980	-
CA-DINO	**10.031**	**0.5387**

Finally, we also provide results of fine-tuning CA-DINO on out-of-domain CN-celeb 1 in Table 4. According to the results, it's observed that our proposed CA-DINO still has better performance even fine-tuning on different domains which demonstrates the robustness and generalization of CA-DINO.

5. CONCLUSION

In this work, we propose the cluster-aware (CA) strategy to reduce the overlap problem when training traditional DINO. With this strategy, the model can utilize positive pairs sampled from several different utterances in the same cluster rather than from a single utterance which can increase the diversity of data and obtain the **state-of-the-art** performance. Besides, in the clustering stage, we also investigate strategies of fixed number clustering or progressive clustering. Finally, we explore the effect of fine-tuning different self-supervised speaker verification models with a small amount of labeled data. With only 10% labeled data, our proposed CA-DINO exceeds the fully supervised system trained on all labeled data.

6. REFERENCES

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

[2] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[3] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[4] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *Proc. APSIPA ASC*. IEEE, 2019, pp. 1652–1656.

[5] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *Proc. IEEE ICASSP*. IEEE, 2016, pp. 5115–5119.

[6] Shuai Wang, Yexin Yang, Yanmin Qian, and Kai Yu, "Revisiting the statistics pooling layer in deep speaker embedding learning," in *Proc. ISCSLP*. IEEE, 2021, pp. 1–5.

[7] Miao Zhao, Yufeng Ma, Yiwei Ding, Yu Zheng, Min Liu, and Minqiang Xu, "Multi-query multi-head attention pooling and inter-topk penalty for speaker verification," in *Proc. IEEE ICASSP*. IEEE, 2022, pp. 6737–6741.

[8] Themos Stafylakis, Johan Rohdin, Oldrich Plchot, Petr Mizera, and Lukas Burget, "Self-supervised speaker embeddings," *arXiv preprint arXiv:1904.03486*, 2019.

[9] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NIPS*, vol. 33, pp. 12449–12460, 2020.

[10] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. ASLP.*, vol. 29, pp. 3451–3460, 2021.

[11] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[12] Zhengyang Chen, Sanyuan Chen, Yu Wu, Yao Qian, Chengyi Wang, Shujie Liu, Yanmin Qian, and Michael Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *Proc. IEEE ICASSP*. IEEE, 2022, pp. 6147–6151.

[13] Jaesung Huh, Hee Soo Heo, Jingu Kang, Shinji Watanabe, and Joon Son Chung, "Augmentation adversarial training for unsupervised speaker recognition," *arXiv preprint arXiv:2007.12085*, 2020.

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

[15] Wei Xia, Chunlei Zhang, Chao Weng, Meng Yu, and Dong Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive learning," in *Proc. IEEE ICASSP*. IEEE, 2021, pp. 6723–6727.

[16] Sung Hwan Mun, Woo Hyun Kang, Min Hyun Han, and Nam Soo Kim, "Unsupervised representation learning for speaker recognition via contrastive equilibrium learning," *arXiv preprint arXiv:2010.11433*, 2020.

[17] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. ICCV*, 2021, pp. 9650–9660.

[18] Bing Han, Zhengyang Chen, and Yanmin Qian, "Self-supervised speaker verification using dynamic loss-gate and label correction," in *Proc. ISCA Interspeech*, 2022, pp. 4780–4784.

[19] Jaejin Cho, Jesús Villalba, Laureano Moro-Velazquez, and Najim Dehak, "Non-contrastive self-supervised learning for utterance-level information extraction from speech," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[20] Hee-Soo Heo, Jee-weon Jung, Jingu Kang, Youngki Kwon, You Jin Kim, and Bong-Jin Lee abd Joon Son Chung, "Self-supervised curriculum learning for speaker verification," *arXiv preprint arXiv:2203.14525*, 2022.

[21] Chunlei Zhang and Dong Yu, "C3-dino: Joint contrastive and non-contrastive self-supervised learning for speaker verification," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[22] Zhengyang Chen, Yao Qian, Bing Han, Yanmin Qian, and Michael Zeng, "A comprehensive study on self-supervised distillation for speaker representation learning," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 599–604.

[23] Yifei Zhang, Chang Liu, Yu Zhou, Wei Wang, Weiping Wang, and Qixiang Ye, "Progressive cluster purification for unsupervised feature learning," in *Proc. ICPR*. IEEE, 2021, pp. 8476–8483.

[24] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. ISCA Interspeech*, 2018, pp. 1086–1090.

[25] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[26] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. ISCA Interspeech*, 2017, pp. 2616–2620.

[27] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 7604–7608.

[28] Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman, "Disentangled speech embeddings using cross-modal self-supervision," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 6829–6833.

[29] Soo-Whan Chung, Hong Goo Kang, and Joon Son Chung, "Seeing voices and hearing voices: learning discriminative embeddings using cross-modal self-supervision," *arXiv preprint arXiv:2004.14326*, 2020.

[30] Nakamasa Inoue and Keita 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.