ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447489

LEVERAGING IN-THE-WILD DATA FOR EFFECTIVE SELF-SUPERVISED PRETRAINING IN SPEAKER RECOGNITION

Shuai Wang[1,*,†], *Qibing Bai*[1,2,*], *Qi Liu*[3], *Jianwei Yu*[3], *Zhengyang Chen*[4],
Bing Han[4], *Yanmin Qian*[4], *Haizhou Li*[2,1]

[1]Shenzhen Research Institute of Big Data, Shenzhen, China
[2]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
[3]Tencent, Shenzhen, China
[4]Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Current speaker recognition systems primarily rely on supervised approaches, constrained by the scale of labeled datasets. To boost the system performance, researchers leverage large pretrained models such as WavLM to transfer learned high-level features to the downstream speaker recognition task. However, this approach introduces extra parameters as the pretrained model remains in the inference stage. Another group of researchers directly apply self-supervised methods such as DINO to speaker embedding learning, yet they have not explored its potential on large-scale in-the-wild datasets. In this paper, we present the effectiveness of DINO training on the large-scale WenetSpeech dataset and its transferability in enhancing the supervised system performance on the CNCeleb dataset. Additionally, we introduce a confidence-based data filtering algorithm to remove unreliable data from the pretraining dataset, leading to better performance with less training data. The associated pretrained models, confidence files, pretraining and finetuning scripts will be made available in the Wespeaker toolkit.

***Index Terms*—** self-supervised learning, DINO, in-the-wild, speaker recognition

1. INTRODUCTION

Deep speaker embedding learning plays a central role in applications related to speaker identity modeling, especially in the field of speaker recognition. Current state-of-the-art systems predominantly adhere to the supervised training paradigm, where speaker labels are employed as the optimization target during training. Due to the need for annotated data, large-scale datasets like VoxCeleb [1, 2] and CNCeleb[3, 4] with speaker labels have attracted significant attention among researchers. To further enhance the performance of related systems, some researchers have turned to universal speech models pretrained on large-scale in-the-wild data, such as WavLM [5], Wav2Vec [6] and Hubert [7]. They extract higher-level features from these models [8, 9] or utilize them as initialization models for finetuning [10]. Additionally, some researchers found that using models trained in Automatic Speech Recognition (ASR) as initialization can also improve the performance of speaker recognition systems [11, 12].

In contrast, researchers have been investigating the integration of various self-supervised training methods for direct training of speaker representation models. Notable methods in this domain include SimCLR [13], MoCo [14], and DINO [15]. Among these approaches, DINO has exhibited particularly impressive performance [16, 17, 18]. However, despite their ability to leverage unlabeled data, current research appears to be primarily focused on pretraining speaker representation models on well-labeled datasets such as VoxCeleb. There is limited exploration of these methods on unlabeled in-the-wild data, which can be attributed primarily to two factors. Firstly, our validation indicates that self-supervised models trained on large-scale data do not exhibit strong performance when directly applied to speaker recognition datasets. Secondly, these self-supervised learning methods inherently impose certain data requirements that might not be met by in-the-wild datasets. For example, it is crucial for a training segment to contain only one speaker, which necessitates meticulous data cleaning, particularly when dealing with in-the-wild data.

In this paper, we propose a strategy that leverages self-supervised training on large-scale in-the-wild data to initialize supervised speaker models. Our approach offers several key advantages compared with the methods mentioned above: 1) Unlike methods employing large models such as WavLM, our approach introduces no additional parameters or computational overhead during the inference stage, making it more efficient. 2) In contrast to strategies that rely on speech recognition models for initialization, our approach does not require any labels, including training transcripts. Our contributions can be summarized as follows:

- We introduce a novel learning method that leverages large-scale in-the-wild unlabeled data to significantly enhance the performance of speaker recognition systems. Our approach achieves an overall 12.4% reduction in Equal Error Rate (EER) on the CNCeleb dataset, exhibiting the immense potential of in-the-wild unlabeled data.
- Building upon the consistency assumption of DINO, which assumes the presence of only one speaker in each training segment, we introduce a data filtering technique utilizing confidence scores generated by a speaker diarization system. Through this simple data-cleaning process, we establish that superior pretraining results can be achieved with fewer yet high-quality data, while incorporating more unreliable data does not necessarily improve the performance.

2. CASCADE SPEAKER EMBEDDING LEARNING

In this section, we describe our cascade speaker embedding learning pipeline, adept at harnessing large-scale, real-world data to im-

*: Equal Contribution, †: Corresponding author

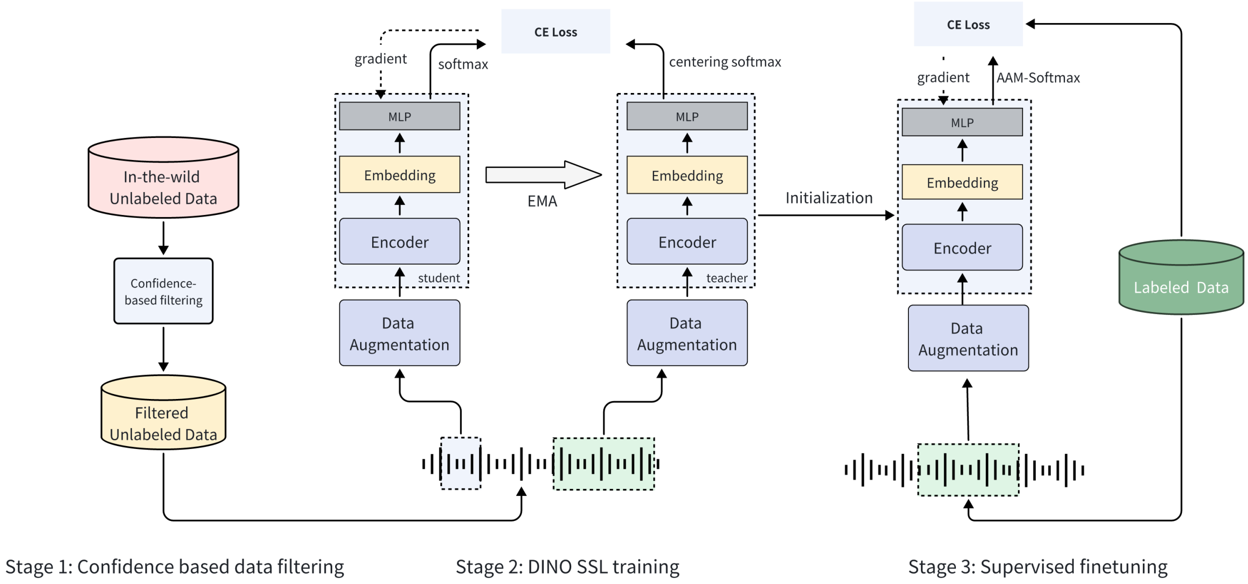

Fig. 1: The cascade speaker embedding learning pipeline

prove the efficacy of supervised training. The complete pipeline is illustrated in Figure 1 and encompasses three stages: 1) Confidence-based data filtering to acquire high-quality unlabeled data for pre-training. 2) DINO-based pretraining using the filtered unlabeled data. 3) Supervised finetuning using the pretrained DINO model for initialization.

2.1. Self-supervised Learning on in-the-wild data

The quality of “in-the-wild” datasets is usually diverse. Without manual verification, it becomes challenging to ensure that each segment used in the final training data contains only one speaker. Furthermore, it is crucial to note that increasing the amount of such data does not necessarily guarantee improved results in speaker recognition tasks. The WenetSpeech dataset [19] serves as an example highlighting this challenge. Therefore, in this context, we propose a data processing pipeline based on speaker diarization. This pipeline enables effective data filtering, retaining segments of relatively high quality for subsequent pretraining objectives.

2.1.1. *Confidence-based data filtering*

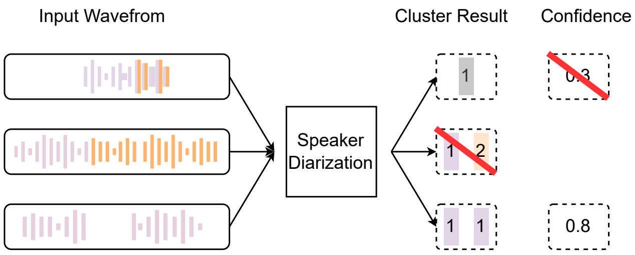

Fig. 2: Filtering segments based on diarization results. Colors represent different speakers. Multi-speaker or low-confidence segments will be dropped.

The pipeline of the proposed data filtering method is illustrated in Figure 2. Each audio segment in the dataset is processed using a standard clustering-based diarization pipeline, resulting in two possible outcomes: 1) only one speaker is detected, or 2) multiple speakers are present.

For segments falling into the second category, where multiple speakers are present, we simply discard them. However, for segments in the first category, potentially featuring a single speaker, we employ a standard sliding window approach to compute a series of short-chunk embeddings. These embeddings are subsequently compared with corresponding cluster centers to measure their similarity, which is regarded as a form of confidence score. If the average score is relatively low, it suggests the possibility of either noisy data or unsuccessful separation of multiple speakers. In either scenario, we consider the segment as low-quality and suitable for filtering out.

2.1.2. *DINO Training*

As shown in Figure 1, the DINO algorithm is implemented using a self-distillation paradigm. The system consists of two sub-networks: the student network and the teacher network, both sharing an identical neural network architecture. During training, following the methodology described in [17], we sample M short segments (local views) and N long segments (global views) from the same utterance. All global and local views are then inputted into the student network, while only global views are inputted into the teacher network. The outputs from the teacher network serve as pseudo labels to guide the optimization of the student network. The optimization loss for the student network can be formulated as follows:

$$\mathcal{L}_{DINO} = \frac{1}{N(N+M-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N+M} H(F_t(x^i), F_s(x^j)) \tag{1}$$

where x^i denotes a global view, x^j denotes a local view, F_t denotes the teacher network, F_s denotes the student network, and $H(\cdot,\cdot)$ represents the cross-entropy loss.

In contrast to the student network, which is optimized using the loss $\mathcal{L}_{DINO}$, the teacher network undergoes updates through the exponential moving average (EMA) applied to the student network. Regarding the architecture of both networks, an initial speaker encoder maps the input to the speaker embedding. This low-dimensional embedding is then processed by several MLP layers to obtain a high-dimensional vector. Finally, a softmax function is applied to this vector, transforming it into a probability distribution. Notably, the teacher network incorporates an additional centering operation before the softmax function.

2.2. Supervised finetuning

After stage 2, where we pretrain the model using DINO on unlabeled data, we use this pretrained model as the initial point for our supervised learning stage. It is worth noting that unlike previous studies [10, 20], where additional models or parameters were introduced for adaptation, our approach maintains the exact same architecture while discarding specific prediction layers utilized in stage 2.

3. EXPERIMENTS

3.1. Dataset

WenetSpeech: WenetSpeech is an open-source ASR corpus that contains over 10,000 hours of Mandarin data from various sources, including YouTube and Podcasts. Since it is collected from real-world data, WenetSpeech serves as a large-scale in-the-wild dataset and is used for the DINO-based self-supervised pretraining.

CNCeleb: For the supervised finetuning dataset, we merge the development sets of CNCeleb1 (274 hours) and CNCeleb2 (1090 hours) to create the final training set (**CNCeleb-Train**). The standard CNCeleb test set is used for evaluation (**CNCeleb-Eval**). Following the Wespeaker recipe[1], we concatenate short utterances from the same speaker to ensure that the samples in the training set are longer than 6 seconds. During scoring, we average the embeddings of multiple enrollment utterances to get a single enrollment embedding for each speaker.

3.2. Experimental settings

Backbone: ECAPA-TDNN [21] is selected as the backbone model for both the DINO pretraining and supervised finetuning. There are 1024 channels in the frame-level convolutional layers, and the final output dimension of DINO is set to 65536.

Training Details: The DINO model is trained using the Wespeaker toolkit [22][2]. For each training sample, we extract two global views and four local views, and then apply random augmentations to all the sampled segments. Both the student and the teacher receive the global views, each spanning 0.3 seconds in duration. The local views, each lasting 0.2 seconds, are only fed to the student. After obtaining the pretrained model, we finetune it for additional 50 epochs on the CNCeleb dataset, initialized from either the teacher model or the student model.

Evaulation Metrics: The cosine back-end serves as the scoring method and the performance of all systems is assessed based on two metrics: Equal Error Rate (EER) and minimum Detection Cost Function (minDCF) with $P_{target} = 0.01$.

Data Filtering Configuration: First, we obtain segments based on the Voice Activity Detection (VAD) information provided in WenetSpeech. Once we have this information, we follow the speaker diarization recipe[3] implemented in the Wespeaker toolkit, while the pretrained *ResNet293* model [4] is used as the speaker embedding extractor. For each segment, we perform standard clustering-based diarization. This involves using a sliding window to extract embeddings, followed by applying spectral clustering. After obtaining the clustering results, we eliminate all segments that contain multiple speakers. For the remaining segments, we calculate the cosine similarity between each embedding and its corresponding cluster center, which serves as the confidence score. We retain only those segments with an average score greater than 0.4. It's worth noting that before these steps, we apply denoising to all the speech data using a denoising tool based on Band-split Recurrent Neural Network (BSRNN) [23], considering that WenetSpeech is quite noisy. After filtering, we discard approximately half of the WenetSpeech dataset, resulting in the differences shown in Table 1.

Table 1: Statistics of WenetSpeech before and after filtering

Dataset	Number of Segments	Total Duration
WenetSpeech	17,848,005	12483.35h
Filtered WenetSpeech	9,661,524	6816.43h

3.3. Results and Analysis

3.3.1. Baseline system

To establish a strong baseline, we incorporated large-margin finetuning [24] and adaptive symmetric normalization (AS-norm) [25] into our pipeline. This resulted in a noticeable performance improvement, as indicated in Table 2. The same processes will be applied to all the systems in the subsequent context.

Table 2: Performance of the baseline system trained on CNCeleb-Train and evaluated on CNCeleb-Eval

System	EER (%)	MinDCF
Baseline	7.879	0.420
+ AS-norm	7.412	0.379
++ Large-margin finetuning	**7.395**	**0.375**

3.3.2. Effectiveness of the proposed pipeline

The performance of the pretrained DINO models is first evaluated considering different training data, as demonstrated in Table 4. In the literature, it has not been explicitly mentioned whether to employ the teacher or student model for evaluation and finetuning [27, 16]. Considering the siamese architecture of DINO, both teacher and student networks are assessed.

From Table 4, we observed that the teacher model exhibits better stability in performance compared to the student model, which aligns with the findings in the paper [28]. This observation might be

[1] `https://github.com/wenet-e2e/wespeaker/tree/master/examples/cnceleb/v2`

[2] The WenetSpeech training recipe and pretrained models will be released in Wespeaker toolkit.

[3] `https://github.com/wenet-e2e/wespeaker/tree/master/examples/voxconverse/v1`

[4] `https://github.com/wenet-e2e/wespeaker/blob/master/docs/pretrained.md`

Table 3: Comparison of performance on CNCeleb-Eval with other pretrain-finetune methods. CNCeleb-Train contains both CNCeleb1 and CNCeleb2 as defined in Section 3.1.

System	Pretraining Configurations			Finetuning Configurations		EER(%)	MinDCF
	Data	Model	Role	Data	Model		
[16]	VoxCeleb2	ECAPA-TDNN	Init	CNCeleb1	ECAPA-TDNN	10.65	-
[26]	VoxCeleb2	ECAPA-TDNN	Init	CNCeleb1	ECAPA-TDNN	8.710	0.422
[27]	VoxCeleb2	ECAPA-TDNN	Init	CNCeleb1	ECAPA-TDNN	10.03	0.539
[20]	CNCeleb1	HuBERT (94.6M)	Frontend	CNCeleb1	HuBERT + ECAPA-TDNN	10.86	-
[20]	CNCeleb-Train	HuBERT (94.6M)	Frontend	CNCeleb-Train	HuBERT + ECAPA-TDNN	8.890	-
[20]	CNCeleb-Train	Conformer (172.2M)	Frontend	CNCeleb-Train	Conformer + MHFA	7.730	0.406
[10] *	Mix 94k hr	WavLM (94.7M)	Frontend	VoxCeleb2 + CNCeleb-Train	WavLM+MAM+MHFA	6.890	0.378
[11]**	WenetSpeech	Conformer (18.8M)	Init	CNCeleb-Train	Conformer	7.420	0.443
Ours	WenetSpeech	ECAPA-TDNN	Init	CNCeleb1	ECAPA-TDNN	7.373	0.383
Ours	+ filtering	ECAPA-TDNN	Init	CNCeleb1	ECAPA-TDNN	7.339	0.377
Ours	WenetSpeech	ECAPA-TDNN	Init	CNCeleb-Train	ECAPA-TDNN	6.738	0.338
Ours	+ filtering	ECAPA-TDNN	Init	CNCeleb-Train	ECAPA-TDNN	**6.474**	**0.331**

* The publicly available WavLM Base+ checkpoint is used, which has been pretrained on a significantly larger dataset (94k hours).
** The pertaining process requires transcriptions because it operates within the ASR framework.

Table 4: Performance comparison of self-supervised DINO models without and with finetuning

Pretraining	Finetuning	DINO Teacher		DINO Student	
Data	Data	EER (%)	MinDCF	EER (%)	MinDCF
CNCeleb-Train	-	13.74	0.563	14.11	0.576
WenetSpeech	-	15.40	0.605	15.22	0.625
+ Filtering	-	15.03	0.560	15.87	0.585
CNCeleb-Train	CNCeleb-Train	7.339	0.366	7.378	0.364
WenetSpeech	CNCeleb-Train	6.738	0.338	6.815	0.341
+ filtering	CNCeleb-Train	**6.474**	**0.331**	6.528	0.331

attributed to the teacher model updating its parameters through the exponential moving average (EMA) of the student model.

Performance comparison between the systems that incorporate models pretrained on WenetSpeech, with and without filtering, exhibits the effectiveness of our proposed data filtering strategy. Notably, better results are achieved with nearly half the training data, as evidenced by improvements in both EER and DCF. This enhancement remains consistent regardless of whether we employ the teacher or student model for initialization. In comparison with the baseline, our training pipeline decreases the EER from 7.395% to 6.474%, achieving a relative reduction of 12.45%.

3.3.3. The effect of self-pretraining

In this section, we evaluate the finetuning performance on CNCeleb when leveraging different self-supervised pretraining models and present the results in Table 4. First, similar to the self-pretraining approach discussed in [20], we conducted both pretraining and finetuning on the CNCeleb dataset. However, we observed only marginal performance improvement compared to the baseline results presented in Table 2. Additionally, it is noteworthy that the model pretrained on WenetSpeech without data filtering surpasses the self-pretrained model, even though the self-pretrained model is initially the top-performing one prior to finetuning.

3.3.4. Comparison with other pretraining methods

To provide a more intuitive demonstration of the effectiveness of our method compared to other pretraining approaches, we summarize a list of results reported in the literature for CNCeleb in Table 3. This list includes details about the pretraining data and the adaptation/finetuning methods employed. The role the pretrained model plays can be classified into two categories: initialization point and feature extraction front-end. For the former category, the finetuning and inference stages do not introduce any new parameters and maintain the backbone exactly the same. However, for the latter one, adaptation layers or a common speaker embedding back-end must be added to the pretrained model. Considering that some of the comparison systems are finetuned only using CNCeleb1, we also present our systems under the same setup to ensure a fair comparison.

From the results, we can observe that our top-performing system surpasses other systems that adhere to the pretrain-finetune approach, even though they may incorporate more pretraining data, additional parameters, or require other label types like ASR transcription. Moreover, it is worth noting that our system, which is only finetuned on the small CNCeleb1 dataset, outperforms the supervised baseline trained on the larger CNCeleb-Train dataset (7.339% versus 7.395% in terms of EER).

4. CONCLUSION

In this paper, we present a pipeline that is able to take advantage of extensive data in the wild to boost the performance of a speaker recognition system. In contrast to prior research that tunes DINO using limited-scale labeled datasets or introduces pretrained models with additional computational cost, we demonstrate that the DINO pretraining on large data can benefit speaker recognition without introducing any extra parameters. Through an efficient data filtering method based on automatic confidence assessment, the proposed system achieves superior results while utilizing only half of the pretraining data.

5. ACKNOWLEDGEMENTS

This research is supported by Internal Project of Shenzhen Research Institute of Big Data under grant No.J00220230014 and No.T00120220002.

6. REFERENCES

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

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

[3] 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 *ICASSP 2020*. IEEE, 2020, pp. 7604–7608.

[4] Lantian Li, Ruiqi Liu, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[5] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[6] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[7] 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 Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[8] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *arXiv preprint arXiv:2111.02735*, 2021.

[9] 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 *ICASSP 2022*. IEEE, 2022, pp. 6147–6151.

[10] Junyi Peng, Themos Stafylakis, Rongzhi Gu, Oldřich Plchot, Ladislav Mošner, Lukáš Burget, and Jan Černockỳ, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[11] Dexin Liao, Tao Jiang, Feng Wang, Lin Li, and Qingyang Hong, "Towards a unified conformer structure: from asr to asv task," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[12] Danwei Cai, Weiqing Wang, Ming Li, Rui Xia, and Chuanzeng Huang, "Pretraining conformer with asr for speaker verification," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. CVPR*, 2020, pp. 9729–9738.

[15] 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.

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

[17] 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.

[18] 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*, vol. 16, no. 6, pp. 1273–1283, 2022.

[19] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *ICASSP 2022*. IEEE, 2022.

[20] Junyi Peng, Oldřich Plchot, Themos Stafylakis, Ladislav Mosner, Lukáš Burget, and Jan "Honza" Černocký, "Improving speaker verification with self-pretrained transformer models," in *Proc. Interspeech 2023*, 2023, pp. 5361–5365.

[21] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[22] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *ICASSP 2023*. IEEE, 2023, pp. 1–5.

[23] Yi Luo and Jianwei Yu, "Music source separation with band-split rnn," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[24] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *ICASSP 2021*. IEEE, 2021, pp. 5814–5818.

[25] Pavel Matějka, Ondřej Novotnỳ, Oldřich Plchot, Lukáš Burget, Mireia Diez Sánchez, and Jan Černockỳ, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Interspeech 2017*, 2017, pp. 1567–1571.

[26] Jingu Kang, Jaesung Huh, Hee Soo Heo, and Joon Son Chung, "Augmentation adversarial training for self-supervised speaker representation learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1253–1262, 2022.

[27] Bing Han, Wen Huang, Zhengyang Chen, and Yanmin Qian, "Improving dino-based self-supervised speaker verification with progressive cluster-aware training," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE, 2023, pp. 1–5.

[28] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017, vol. 30.