Interspeech 2024
1-5 September 2024, Kos, Greece

Self-Supervised Learning with Multi-Head Multi-Mode Knowledge Distillation for Speaker Verification

Zezhong Jin, Youzhi Tu, and Man-Wai Mak

Dept. of Electrical and Electronic Engineering, The Hong Kong Polytechnic University
Hong Kong SAR

zezhong.jin@connect.polyu.hk, 918tyz@gmail.com, enmwmak@polyu.edu.hk

Abstract

Training speaker verification (SV) systems without labeled data is challenging. To tackle the challenge, we propose Multi-Head, Multi-Mode (MeMo) self-supervised learning based on knowledge distillation. Unlike DINO, the teacher in MeMo uses two distinct architectures to learn collaboratively, and so does the student. MeMo employs two distillation modes: self- and cross-distillations, with the teacher and student having the same and different architectures, respectively. To reduce the output discrepancy caused by different architectures, we divide the projection head into self- and cross-heads so that each head is responsible for distillation in its respective mode. We also discover that contrastive learning at the embedding level is supportive only in early training stages. To address this issue, we propose dynamically stopping the contrastive learning while continuing knowledge distillation. MeMo achieves an impressive EER of 3.10% on Voxceleb1 using a small ECAPA-TDNN backbone.

Index Terms: speaker verification, self-supervised learning, knowledge distillation, DINO, cross-distillation

1. Introduction

With the advancement of deep neural networks, there has been an increasing number of neural network-based methods in speaker verification (SV), e.g., ResNet speaker embedding [1], ECAPA-TDNN [2], and CAM++ [3]. Although these methods have achieved impressive performance in SV, the reliance on labeled data poses a challenge to system development. Because manually labeling massive amounts of data is expensive and time-consuming. In recent years, self-supervised learning, a technique that does not rely on labeled data for training, has gained wide attention [4–12].

Self-supervised learning can be divided into two categories: contrastive learning [4–9] and non-contrastive learning [10–12]. The former aims to reduce the distance between (positive) samples from the same speaker and maximize the distance between (negative) samples from different speakers. However, it faces the challenge of false negative samples [13], i.e., having multiple audio samples from the same speaker in a mini-batch. This can lead to the model pushing the embeddings of these incorrect negatives away from the anchor during optimization. To address the issue, many researchers have shifted their focus to non-contrastive frameworks. For example, researchers in computer vision have introduced a novel approach called Bootstrap Your Own Latent (BYOL) [14], which focuses solely on positive pairs. By doing so, they effectively mitigate the problem of false negatives. Subsequently, another study advanced this research by introducing a label-free self-distillation method known as DINO [15]. DINO streamlines the model architecture of BYOL and incorporates a more efficient training strategy, enhancing the overall effectiveness of self-supervised learning. This paper focuses on using the DINO framework for text-independent SV.

DINO has been applied to SV. For instance, a clustering approach was utilized in [10] to obtain more reasonable global and local views for DINO. The effectiveness of curriculum learning in the DINO framework was demonstrated in [16]. Additionally, the authors in [12] introduced two regularization terms to address the issue of model collapse in DINO.

The methods above only utilized one model architecture (student and teacher have the same architecture). In [17], the authors demonstrated that performing knowledge distillation between different models can boost the performance of both models. This finding motivates us to use multiple teachers with different architectures to collectively educate a student network for SV. We propose an approach that involves self-distillation within the same architecture and cross-distillation across different architectures. However, due to the differences in model architecture, there is a significant possibility of a mismatch in the output distributions between the teachers and the students. This can result in different teachers teaching different information to a student, which can pose challenges for the student during the optimization process. To address this problem, we propose a **M**ulti-H**e**ad, **M**ulti-m**o**de (MeMo) distillation framework. Each network consists of one encoder and multiple heads, with each head specialized to a specific teacher's knowledge distillation. This specialization of projection heads leads to more accurate teacher signals, reducing the chance of confusing the students.

Several studies have used contrastive loss to increase the distance between different classes in the teacher and student model [18]. For example, [19] uses the contrastive framework as the first stage to find a better initial parameter for DINO. Inspired by these works, we incorporate contrastive learning at the embedding level between the teachers and students. However, we observed that contrastive loss provides significant benefits to the system only in the early training stages, and it could hurt performance at the later stages. We noticed that the degradation is due to the false negatives in the absence of speaker labels. To address this issue, we propose a strategy called Contrastive Training with Early Stopping (CTES). We utilize contrastive loss as an auxiliary training objective to assist the system during the initial training stage and remove this objective when the successive contrastive loss ceases to drop. The goal is to enable the model to learn a good speaker representation during the initial training phase. In this way, models can find a better solution during subsequent optimization and avoid the issue of getting stuck in trivial solutions in the early stages.

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 15210122

10.21437/Interspeech.2024-360

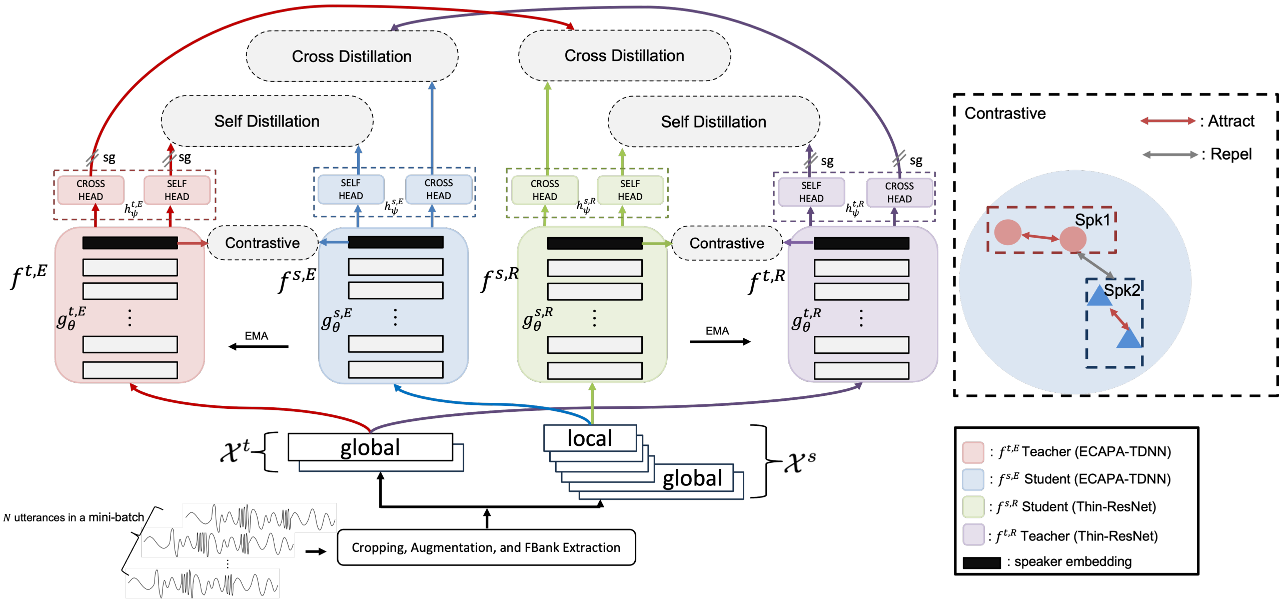

Figure 1: *The MeMo Framework. "EMA" and "sg" stand for exponential moving average and stop gradient, respectively. The contrastive loss of each positive pair depends on all utterances in a training batch (see Eq 4). For clarity, we show the FBank feature sets ($\mathcal{X}^s$ and $\mathcal{X}^t$) of one utterance only.*

Our contributions can be summarized as follows: 1) We proposed the MeMo framework, which incorporates cross-distillation for SV. To the best of our knowledge, we are the first to apply cross-distillation in SV. 2) We proposed a multi-head projection layer to prevent architecture-dependent bias arising from cross-distillation between different architectures. 3) We proposed a CTES strategy to leverage the contrastive during the early training stage.

The rest of the paper is organized as follows. Section 2 introduces the DINO and details the MeMo framework. Section 3 presents the experimental settings, and Section 4 shows the results and analyses. We draw a conclusion in Section 5.

2. Methods

2.1. DINO-based Self-Supervised Learning

DINO [15] facilitates knowledge transfer from a teacher network to a student network by leveraging their output distributions. The teacher and student networks consist of an encoder g_θ and a projection network h_ψ, where g and h are functions parameterized by θ and ψ, respectively. Unlike traditional knowledge distillation, the teacher and student networks in DINO have the same structure but different parameters. During training, the student network is optimized by minimizing the cross-entropy between the outputs of the two networks. The parameters of the teacher network are updated from the student network using the exponential moving average (EMA) algorithm [20].

We employ a multi-crop strategy [21] to sample four short (*l*ocal) segments $\{\mathbf{x}_{l1}, \mathbf{x}_{l2}, \mathbf{x}_{l3}, \mathbf{x}_{l4}\}$ and two long (*g*lobal) segments $\{\mathbf{x}_{g1}, \mathbf{x}_{g2}\}$ from each training utterance, where g and l denote the global and local views, respectively. We apply different types of noise and reverberation to the segments, followed by extracting their filter-bank (FBank) features to obtain two groups of spectral matrices: $\mathbb{X}_g = \{\boldsymbol{X}_{g1}, \boldsymbol{X}_{g2}\}$ and $\mathbb{X}_l = \{\boldsymbol{X}_{l1}, \boldsymbol{X}_{l2}, \boldsymbol{X}_{l3}, \boldsymbol{X}_{l2}\}$. In DINO, we define the teacher data as $\mathcal{X}^t \equiv \mathbb{X}_g$ and present it to the teacher network, and we define the student data as $\mathcal{X}^s \equiv \{\mathbb{X}_g, \mathbb{X}_l\}$ and present it to the student network. This causes the teacher and student networks to output the probability vectors $\mathcal{P}^t = \{\boldsymbol{y}^t_{\mathbf{X}_{g1}}, \boldsymbol{y}^t_{\mathbf{X}_{g2}}\}$ and $\mathcal{P}^s = \{\boldsymbol{y}^s_{\mathbf{X}_{g1}}, \boldsymbol{y}^s_{\mathbf{X}_{g2}}, \boldsymbol{y}^s_{\mathbf{X}_{l1}}, \boldsymbol{y}^s_{\mathbf{X}_{l2}}, \boldsymbol{y}^s_{\mathbf{X}_{l3}}, \boldsymbol{y}^s_{\mathbf{X}_{l4}}\}$ respectively at the softmax layer, where $\boldsymbol{y}$ is a vector with a dimension equal to the hypothesized number of speakers in the training set. The logit output of the teacher network is subject to centering before applying the softmax function to avoid model collapse. The DINO loss is defined as:

$$L_{\text{DINO}} = \sum_{\mathbf{X} \in \mathcal{X}^t} \sum_{\substack{\mathbf{X}' \in \mathcal{X}^s \\ \mathbf{X}' \neq \mathbf{X}}} \text{CrossEntropy}(\boldsymbol{y}^t_{\mathbf{X}}, \boldsymbol{y}^s_{\mathbf{X}'}), \tag{1}$$

where $\text{CrossEntropy}(\boldsymbol{a}, \boldsymbol{b}) = -\sum_{k=1}^{K} a_k \log b_k$. After training, we utilize the embeddings produced by the teacher encoder as the speaker embeddings.

2.2. Multi-head Knowledge Distillation

Traditional DINO has two networks: student network $f^s \equiv g^s_\theta \circ h^s_\psi$ and teacher network $f^t \equiv g^t_\theta \circ h^t_\psi$. In MeMo, on the other hand, there are two student networks ($f^{s,E}$ and $f^{s,R}$) and two teacher networks ($f^{t,E}$ and $f^{t,R}$), where the superscript E and R denote the ECAPA-TDNN and Thin-ResNet architectures, respectively. Figure 1 shows the knowledge distillation among these four networks and how they use the global and local speech segments.

Unlike DINO, MeMo's knowledge distillation is done within the same architecture (self) and across different architectures (cross). As shown in Figure 1, we feed $\mathcal{X}^t$ into the teachers $f^{t,E}$ and $f^{t,R}$ and $\mathcal{X}^s$ into the students $f^{s,E}$ and $f^{s,R}$. The self-distillation loss is the cross-entropy between the outputs of the student and teacher networks with the same architecture. For the cross-distillation loss, the cross-entropy is calculated between the outputs of the student and teacher networks with different architectures. Taking $f^{s,E}$ as an example, these two losses are defined as:

$$L_{\text{self}} = \sum_{\boldsymbol{X} \in \mathcal{X}^t} \sum_{\substack{\boldsymbol{X}' \in \mathcal{X}^s \\ \boldsymbol{X}' \neq \boldsymbol{X}}} \text{CrossEntropy}(\boldsymbol{y}^{t,E}_{\boldsymbol{X}}, \boldsymbol{y}^{s,E}_{\boldsymbol{X}'}) \quad (2)$$

$$L_{\text{cross}} = \sum_{\boldsymbol{X} \in \mathcal{X}^t} \sum_{\substack{\boldsymbol{X}' \in \mathcal{X}^s \\ \boldsymbol{X}' \neq \boldsymbol{X}}} \text{CrossEntropy}(\boldsymbol{y}^{t,R}_{\boldsymbol{X}}, \boldsymbol{y}^{s,E}_{\boldsymbol{X}'}), \quad (3)$$

where $\boldsymbol{y}$'s with different superscripts denote the respective probability vectors of the teacher and student networks. The losses corresponding to $f^{s,R}$ have similar forms.

While cross-distillation allows knowledge transfer between teacher and student and between different architectures, there is a catch. Specifically, problems arise if there is a big disagreement between the teacher and student, which becomes more likely when they are of different architectures. For example, the teacher $f^{t,E}$ may confidently indicate that an audio segment belongs to Speaker "A", whereas the teacher $f^{t,R}$ strongly believes it is from Speaker "B". The probability of such a mismatch increases when the architectural difference increases. This discrepancy causes the two teachers to teach different information to the two students, confusing the students during the optimization process. To address this problem, we divided the MLP's output layer into self- and cross-heads (the reason for the name **M**ulti-h**e**ad). The encoder's output is fed into both heads. The self-head is used for self-distillation, while the cross-head is used for cross-distillation (as shown in Figure 1). Splitting the MLP's output layer into two projection heads allows each head to specialize in one type of distillation, which is less demanding than requiring a single head to handle both types of distillation or knowledge transfer. This specialization of projection heads leads to more accurate teacher signals, reducing the chance of confusing the students.

2.3. Knowledge Distillation via Contrastive Loss

Besides self- and cross-distillation, knowledge distillation can also occur at the embedding layer through the contrastive loss (the reason for the name **M**ulti-m**o**de). The contrastive learning aims to bring the embeddings of the same speaker closer and to push the embeddings of different speakers apart. Let us take $f^{t,E}$ and $f^{s,E}$ as an example. Given an audio sample $\mathbf{x}$, we consider its global view segments, $\boldsymbol{X}_{g1}$ and $\boldsymbol{X}_{g2}$. After feeding them into $g^{t,E}_{\theta}$ and $g^{s,E}_{\theta}$, we obtain four sets of embeddings: $\mathbf{e}^{t,E}_{gi} = g^{t,E}_{\theta}(\boldsymbol{X}_{gi})$ and $\mathbf{e}^{s,E}_{gi} = g^{s,E}_{\theta}(\boldsymbol{X}_{gi})$, where $i = 1, 2$ indexes the two global views. We perform contrastive learning on the embeddings $\mathbf{e}^{t,E}_{g1}$ and $\mathbf{e}^{s,E}_{g2}$ to enable the encoders to produce robust embeddings because $\boldsymbol{X}_{g1}$ and $\boldsymbol{X}_{g2}$ were obtained from different augmentations of $\mathbf{x}$.

To simplify the notations for $\mathbf{e}$, we drop the superscript E, s, and t and the subscript g in the sequel. Therefore, given N utterances in a mini-batch, we have embeddings $\mathbf{e}_{i,j}$, where $i = 1, 2$ and $j = 1, \ldots, N$. Because the embeddings $\mathbf{e}_{1,j}$ and $\mathbf{e}_{2,j}$ are drawn from the same utterance, they share the same speaker identity and therefore form a positive pair. For each positive pair with utterance index j, the other $2(N-1)$ embeddings in the mini-batch are considered negatives. Therefore, for a positive pair of speaker embeddings $\{\mathbf{e}_{i,j}, \mathbf{e}_{|3-i|,j}\}$, the contrastive loss is defined as:

$$l_{i,j} = -\log \frac{\exp(\cos(\mathbf{e}_{i,j}, \mathbf{e}_{|3-i|,j}))}{\sum_{k=1}^{N} \sum_{l=1}^{2} \mathbb{1}_{[l \neq i, k \neq j]} \exp(\cos(\mathbf{e}_{i,j}, \mathbf{e}_{l,k}))}. \quad (4)$$

The contrastive loss corresponding to $f^{t,E}$ and $f^{s,E}$ for each mini-batch is then given by:

$$L_{\text{scl}} = \frac{1}{2N} \sum_{j=1}^{N} \sum_{i=1}^{2} l_{i,j}. \quad (5)$$

Note that for each pair of embeddings in Eq 4, one comes from the teacher and another from the student, constituting the knowledge distillation. The contrastive loss corresponding to the ResNets ($f^{t,R}$ and $f^{s,R}$) has the same form as Eqs. 4 and 5.

We found that contrastive loss is helpful in the early training stage only. Due to the issue of false negatives, the contrastive loss limits the system's performance during the later training stage. Therefore, we propose a strategy called Contrastive Training with Early Stopping (CTES). Specifically, we calculate the difference between the current contrastive loss $L_{\text{scl},t}$ and the previous loss $L_{\text{scl},t-1}$, where t is the iteration index. When $L_{\text{scl},t-1} - L_{\text{scl},t} < \tau$ (a threshold parameter), we consider the auxiliary effect of contrastive loss on the system to be minimal and remove the contrastive loss from the training objective.

3. Experimental Setup

3.1. Datasets

The training set utilized in our study is Voxceleb2 [1], which consists of 1,092,009 utterances from 5,994 speakers. Notably, we did not use the speaker labels during training. The system's effectiveness was assessed using Vox1-O, Vox1-E, and Vox1-H [22]. We adhered to the Kaldi recipe for data augmentation, integrating noise from the MUSAN [23] dataset and reverberation from the RIR [24] dataset.

3.2. System configuration

We used the ECAPA-TDNN [2] with 512 channels (a smaller version of ECAPA-TDNN) as $f^{t,E}$ and $f^{s,E}$ and the Thin-ResNet [27] (a smaller version of ResNet) as $f^{t,R}$ and $f^{s,R}$. The embedding dimensions for the ECAPA-TDNN and Thin-ResNet were set to 256. The cross-head and self-head in Figure 1 have the same architecture. They consist of three fully-connected layers, where the hidden layer has 2048 nodes. This is followed by an L2-normalization layer and a weight normalization layer [28]. This architecture aims to map the speaker embeddings to an output layer with K dimensions. We set K to 65536, following the default setting in DINO.

We applied the multi-crop strategy to each utterance during MeMo training. In this strategy, we considered 4-second segments as the global views, while 2-second segments were treated as the local views. The ECAPA-TDNNs and Thin-ResNets receive 80-dimensional filter bank (FBank) features as input. Each FBank vector was computed from 25ms of speech, with a 10-ms frameshift. Mean and variance normalization was applied to the FBank features. The temperature parameter of the student network's softmax function was set to 0.1. The temperature parameter of the teacher's softmax function was linearly increased from 0.04 to 0.07 during the initial 30 epochs and fixed afterward. The weights for combing different types of loss were set to 1, and the threshold τ in Section 2.3 was set to 0.01. The ECAPA-TDNNs and Thin-ResNets were optimized by an SGD optimizer using a cosine scheduler. The initial learning rate was set to 0.2, and the final learning rate was set to 0.00005.

We utilized equal error rate (EER) and the minimum detection cost function (MinDCF) as the performance metrics. The

Table 1: *Comparison of the Proposed MeMo with the baseline on Voxceleb1 test sets. "SeMo" means single-head multi-mode distillation, where the MLP's output was not split into multiple heads. SCL and CTES stand for self-supervised contrastive loss and contrastive training with early stopping, respectively (see Section 2.3).*

			Speaker Embedding	Vox1-O		Vox1-E		Vox1-H	
Row	Knowledge Distillation Method	SCL	Network	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
1	DINO (Baseline)	N	Thin-ResNet	6.33	0.394	6.82	0.411	11.69	0.599
2	SeMo	N		6.14	0.368	6.66	0.392	11.02	0.559
3	MeMo	N		5.86	0.376	6.47	**0.388**	10.71	0.543
4	MeMo (w/o CTES)	Y		6.32	0.393	6.91	0.435	12.02	0.599
5	MeMo (w/ CTES)	Y		**5.78**	**0.359**	**6.40**	0.399	**10.71**	**0.553**
6	DINO (Baseline)	N	ECAPA-TDNN	4.03	0.301	4.40	0.336	8.65	0.592
7	SeMo	N		3.53	0.252	4.02	0.300	7.52	**0.478**
8	MeMo	N		3.19	0.246	3.68	0.351	7.68	0.488
9	MeMo (w/o CTES)	Y		5.35	0.355	6.19	0.405	11.47	0.588
10	MeMo (w/ CTES)	Y		**3.10**	**0.229**	**3.53**	**0.297**	**7.04**	0.569

Table 2: *Comparison of our method (MeMo) with other start-of-art methods on the Vox1-O test set. The minDCF with a * was calculated using Ptarget = 0.01 instead of Ptarget = 0.05.*

System	EER (%)	minDCF
AP+AAT [5]	8.65	0.454
MoCo+WavAug [8]	8.23	0.590*
Contrastive first stage [9]	7.36	N/R
Contrastive second stage [9]	3.52	N/R
SSReg [25]	6.99	0.434
DINO [26]	4.83	N/R
DINO+CL [16]	4.47	0.3057
CA DINO [10]	3.585	0.353
RDINO [12]	3.29	0.247
MeMo w/ SCL+CTES (Ours)	**3.10**	**0.229**

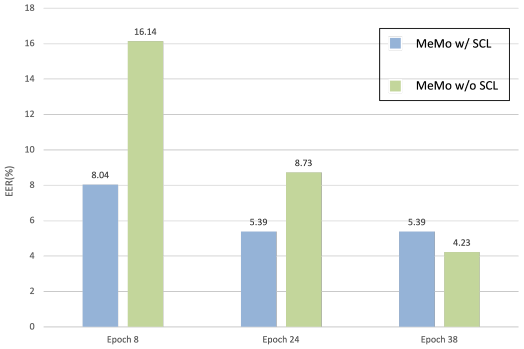

Figure 2: *The EERs achieved by MeMo with and without SCL at three different epochs, with speaker embeddings extracted from the ECAPA-TDNN.*

MinDCF was determined using the parameters P_{target} = 0.05 and C_{fa} = C_{miss} = 1. We used the cosine similarities of embedding pairs as verification scores.

4. Results and Discussions

Table 1 presents our main results. We conducted experiments on three test scenarios in Voxceleb1 and tested on the speaker embeddings extracted from either the ECAPA-TDNN or the Thin-ResNet to demonstrate the effectiveness of MeMo. The term SeMo stands for single-head multi-mode, meaning that only one projection head was used for self- and cross-distillation. Row 4 indicates that MeMo also uses contrastive loss for training, but does not use the CTES strategy.

Our baseline is the standard DINO model with two global views and four local views. Our method (MeMo) performs better than the baseline using the embeddings from either ECAPA-TDNN or Thin-ResNet. This indicates that under the MeMo framework, two different models can learn collaboratively in a self-supervised manner. By comparing the results of Rows 1 and 2 and Rows 6 and 7, we observe that diversifying the model architecture is effective for DINO. By comparing the results of Rows 2 and 3 and Rows 7 and 8, we observe that MeMo, with distillation-type-dependent heads, is superior. Comparing the results of Rows 4 and 9 with the baseline suggests that a naive application of contrastive learning under the MeMo framework can hurt performance. However, as shown in Figure 2, with the assistance of contrastive learning, MeMo can obtain a better speaker representation in the early training stage. The contrastive loss may prevent the model from getting stuck in a trivial solution in the early stages. Therefore, we propose the CTES strategy. Rows 5 and 10 in Table 1 indicate that the CETS strategy can boost system performance.

To further demonstrate the effectiveness of MeMo, we also compared it with state-of-the-art self-supervised methods in recent years. From Table 2, our method achieved 3.10% on Voxceleb1 by using a smaller ECAPA-TDNN backbone. This result is better than those of other recent self-supervised methods, including the previously state-of-the-art RDINO [12], demonstrating the effectiveness of MeMo.

5. Conclusions

We found that increasing model architecture diversity is effective for knowledge distillation under the DINO framework. We proposed dividing the classification heads into distillation-type-dependent heads to overcome the architecture-dependent bias in the teacher and student networks. We also introduced knowledge distillation at the embedding layer through contrastive learning, discovering that stopping contrastive learning at the early learning stage is critical. Early stopping can help MeMo learn better speaker representation at the later stage of training.

6. References

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

[2] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," *in Proc. Interspeech*, pp. 3830–3834, 2020.

[3] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "CAM++: A fast and efficient network for speaker verification using context-aware masking," *in Proc. Interspeech*, pp. 5301–5305, 2023.

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

[5] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. Son Chung, "Augmentation adversarial training for self-supervised speaker recognition," in *Workshop on Self-Supervised Learning for Speech and Audio Processing, NeurIPS*, 2020.

[6] D. Cai, W. Wang, and M. Li, "An iterative framework for self-supervised deep speaker representation learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6728–6732.

[7] H. Zhang, Y. Zou, and H. Wang, "Contrastive self-supervised learning for text-independent speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6713–6717.

[8] 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 International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6723–6727.

[9] R. Tao, K. A. Lee, R. K. Das, V. Hautamäki, and H. Li, "Self-supervised speaker recognition with loss-gated learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6142–6146.

[10] B. Han, Z. Chen, and Y. Qian, "Self-supervised learning with cluster-aware-DINO for high-performance robust speaker verification," *arXiv preprint arXiv:2304.05754*, 2023.

[11] J. Cho, R. Pappagari, P. Żelasko, L. Moro-Velazquez, J. Villalba, and N. Dehak, "Non-contrastive self-supervised learning of utterance-level speech representations," *in Proc. Annual Conference of the International Speech Communication Association*, pp. 4028–4032, 2022.

[12] Y. Chen, S. Zheng, H. Wang, L. Cheng, and Q. Chen, "Pushing the limits of self-supervised speaker verification using regularized distillation framework," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[13] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," *in Proc. International Conference on Machine Learning*, pp. 5628–5637, 2019.

[14] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent: a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[15] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9650–9660.

[16] H.-S. Heo, J.-W. Jung, J. Kang, Y. Kwon, Y. J. Kim, and B.-J. L. JS Chung, "Self-supervised curriculum learning for speaker verification," *arXiv preprint arXiv:2203.14525*, 2022.

[17] K. Song, J. Xie, S. Zhang, and Z. Luo, "Multi-mode online knowledge distillation for self-supervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 848–11 857.

[18] L. Zhang, Q. Wang, K. A. Lee, L. Xie, and H. Li, "Multi-level transfer learning from near-field to far-field speaker verification," *in Proc. Interspeech*, pp. 1094–1098, 2021.

[19] C. Zhang and D. 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.

[20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[21] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[22] A. Nagrani, J. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *in Proc. Interspeech*, pp. 2616—-2620, 2017.

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

[24] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[25] M. Sang, H. Li, F. Liu, A. O. Arnold, and L. Wan, "Self-supervised speaker verification with simple siamese network and self-supervised regularization," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6127–6131.

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

[27] 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.

[28] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 458–463, 2016.