# Self-supervised Reflective Learning through Self-distillation and Online Clustering for Speaker Representation Learning

Danwei Cai, Zexin Cai, Ze Li, and Ming Li, Senior Member, IEEE

Abstract—Speaker representation learning is crucial for voice recognition systems, with recent advances in self-supervised approaches reducing dependency on labeled data. Current twostage iterative frameworks, while effective, suffer from significant computational overhead due to repeated rounds of clustering and training. They also struggle with noisy pseudo labels that can impair model learning. This paper introduces self-supervised reflective learning (SSRL), an improved framework that addresses these limitations by enabling continuous refinement of pseudo labels during training. Through a teacher-student architecture and online clustering mechanism, SSRL eliminates the need for iterative training rounds. To handle label noise, we incorporate noisy label modeling and pseudo label queues that maintain temporal consistency. Experiments on VoxCeleb show SSRL's superiority over current two-stage iterative approaches, surpassing the performance of a 5-round method in just a single training round. Ablation studies validate the contributions of key components like noisy label modeling and pseudo label queues. Moreover, consistent improvements in pseudo labeling and the convergence of cluster counts demonstrate SSRL's effectiveness in deciphering unlabeled data. This work marks an important advancement in efficient and accurate self-supervised speaker representation learning through the novel reflective learning paradigm.

Index Terms—Self-supervised learning, self-labeling, knowledge distillation, noisy label modeling, speaker recognition

## I. INTRODUCTION

PEAKER representation learning is a core component of voice recognition systems that aims to extract discriminative speaker characteristics from speech signals. Recent research has demonstrated significant advances in self-supervised learning approaches for speaker representation learning [1]–[7]. These methods enable the utilization of unlabeled data, reducing dependency on manual annotation and facilitating deployment in practical applications.

Our previous work introduced a two-stage iterative framework for unsupervised speaker representation learning [8], [9]. The first stage performs self-supervised speaker representation learning, while the second stage combines clustering with discriminative training. In this framework, clustering algorithms analyze the learned representations to generate pseudo labels

Danwei Cai and Zexin Cai are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, 27705, USA, e-mail: {danwei.cai, zexin.cai}@duke.edu

Ze Li and Ming Li are with Suzhou Municipal Key Laboratory of Multimodal Intelligent Systems, Digital Innovation Research Center, Duke Kunshan University, Kunshan, China and School of Computer Science, Wuhan University, Wuhan China, e-mail: {ze.li, ming.li369}@dukekunshan.edu.cn Corresponding author: Ming Li.

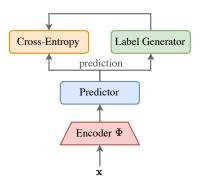


Fig. 1: A naive solution to bypass the iterative process of clustering and discriminative training in two-stage framework.

for unlabeled data. These pseudo labels, despite containing some noise, are then used to train the network through discriminative learning. The process iterates, with each training round refining the representations and improving the quality of pseudo labels, leveraging DNNs' inherent robustness to label noise.

However, this two-stage iterative approach, while effective, introduces significant computational overhead. The primary limitation stems from the iterative process of generating pseudo labels through clustering followed by discriminative training. Furthermore, the initial pseudo labels derived from clustering contain substantial noise, which impairs the model's ability to learn discriminative speaker features.

To streamline learning and enhance efficiency, we sought to bypass the iterative nature of the two-stage framework by updating pseudo labels at every training step rather than after each full training round. As shown in Figure 1, A naive solution is to directly generate labels from model predictions – e.g., assigning the class with the highest posterior probability. However, this approach can lead to a degenerate solution, where the model suffers from confirmation bias, collapsing to a single prediction across all samples [10].

The key to avoiding this degenerate solution is to decouple training from label assignment, ensuring the model does not directly train on its own predictions. In the two-stage iterative framework, this decoupling is achieved by clustering pseudo labels after model convergence. To generate pseudo label at every training step while preventing degenerate solution, we propose self-supervised reflective learning (SSRL), which achieves this decoupling via self-supervised knowledge dis-

tillation [10]–[13]. The framework employs a teacher model to generate pseudo labels for training a student model. The teacher model, updated as an exponential moving average (EMA) of the student, functions as an ensemble of past model states, thereby stabilizing pseudo label generation and preventing overfitting. This mechanism enables continuous knowledge integration from previous training steps while maintaining stable learning dynamics. Essentially, the student model learns from its own reflection – a feedback-driven process where insights from previous training steps guide and improve future learning.

Another limitation of the two-stage iterative framework is the high noise in pseudo labels, which degrades learning. We address this with two key strategies. First, we maintain a queue of historical pseudo labels to filter out outlier predictions and ensures consistent and reliable pseudo labels. Second, we integrate a label noise modeling strategy using a two-component Gaussian mixture model (GMM) to capture the loss distribution of training samples, as described by [14]. This approach leverages the tendency of DNNs to prioritize correct targets, providing a clean label probability for each sample, which is then used to adjust the final loss.

This paper presents several contributions to the field of selfsupervised speaker representation learning:

- The introduction of a new learning paradigm, 'self-supervised reflective learning', that bridges self-supervised representation learning with self-supervised knowledge distillation and online clustering, eliminating iterative bottlenecks of the two-stage unsupervised framework.
- A detailed examination of how noisy label modeling, when combined with self-supervised knowledge distillation, can handle label noise, thereby improving the robustness of the learning process.
- Our approach surpasses existing iterative methodologies, marking a significant step forward in efficient and accurate unsupervised speaker representation learning.

#### II. RELATED WORKS

## A. Self-supervised knowledge distillation

In self-supervised learning, knowledge distillation involves processing two distinct views through separate encoders and mapping one to the other using a predictor. A potential pitfall is the convergence of outputs to a uniform constant.

To addressed this issue, 'bootstrap your own latent' (BYOL) introducing a momentum-based teacher network to generate targets for the student network [12]. Both networks process distinct views of the same instance through data augmentation. The student network aligns its outputs with the teacher network using a predictor, while the teacher network is updated through an EMA of the student network's weights. He *et al.* later introduced the 'simple siamese' (SimSam) approach, which simplified BYOL by removing the momentum mechanism [10], showing that while EMA wasn't essential, it could enhance performance.

Similar to BYOL, 'self-distillation with no labels' (DINO) focuses on regression from student to momentum encoder

representations [13]. However, DINO differs by using crossentropy instead of Mean square error (MSE) or cosine similarity for alignment, and by centralizing the teacher's output using a running mean with temperature-scaled softmax. With its large output dimension (65536 in the original paper), DINO effectively functions as an online clustering mechanism.

Our proposed SSRL approch deviates from the DINO approach in multiple ways. Firstly, SSRL builds on a two-stage iterative unsupervised framework, starting with more reliable pseudo labels rather than random initialization. Unlike DINO's continuous class probability distributions, SSRL delivers discrete class predictions thus enables discriminative training of the model. Additionally, SSRL adopts noisy student training where only the student network undergoes augmentation and noise, allowing the teacher to generate higher quality pseudo labels. Lastly, SSRL integrates a pseudo label queue and noisy label modeling handle label inaccuracies.

Self-supervised knowledge distillation has also been applied to speech representation learning [15]–[17]. For instance, DinoSR [16] adapts DINO for speech representation learning with sequence modeling, combining masked language modeling (MLM), self-distillation, and online clustering. While DinoSR focuses on frame-level pretraining using an explicit codebook for pseudo-label generation, our work targets utterance-level modeling of speaker characteristics.

## B. Self-supervised pseudo labeling

In self-supervised learning, many approaches use clustering-derived pseudo labels for discriminative training. A primary approach, known as deep clustering (DC), combines conventional clustering with classification loss for network training [18]. However, DC faces several challenges: Firstly, the conventional off-the-shelf clustering requires feature extraction across the full dataset for every epoch. Secondly, the clustering alters cluster indexes across epochs, necessitating a reset of the parametric classifier, leading to unstable network training. Thirdly, The combination of discriminative and clustering losses can lead to degenerate solutions where all samples map to the same pseudo label [19]. DC avoids the issue by optimizing only one loss and keeping the other loss fixed between training epochs.

Prototypical contrastive learning (PCL) addresses the cluster index permutation by replacing the classification layer with cluster centroids [20]. It generates class probabilities by contrasting samples with centroids and incorporates instance discrimination. However, PCL still requires per-epoch feature extraction and faces challenges with the divergence between evolving representations and fixed centroids.

Online deep clustering (ODC) improves upon DC using sample and centroid memories for pseudo label generation [21]. It updates sample memory through moving averages and assigns labels based on nearest centroids. To prevent degenerate solutions, ODC implements 'merge-and-split' operations and loss re-weighting for small clusters.

Recently, Asano *et al.* introduced a self-labeling algorithm (SeLa) addressing the degenerate solutions in combined clustering and representation learning [19]. Contrary

to DC's direct clustering application, SeLa determines label assignments  $q(y|\mathbf{x}_i)$  from the network-derived class posterior probabilities  $p(y|\mathbf{x}_i)$ . Here, y denotes labels and  $\mathbf{x}_i$  denotes data samples. SeLa solves the cross-entropy optimization problem  $\min_q \sum_i \sum_y q(y|\mathbf{x}_i) \log p(y|\mathbf{x}_i)$  with equal-sized partition constraints using the Sinkhorn-Knopp algorithm. While this unifies network training and clustering objectives, SeLa's offline cluster assignment limits its scalability.

SwAV (Swapping Assignments between Views) advanced SeLa by introducing online clustering through optimal transport within mini-batches [22]. Instead of processing the full dataset, SwAV enforces consistency between cluster assignments from multiple augmented views of the same image. The swapped prediction task – predicting one view's code from another's representation – enables online training while maintaining equipartition constraints via the Sinkhorn-Knopp algorithm. Chang *et al.* apply SwAV to self-supervised speech representation learning by processing original speech and speaker-perturbed versions through shared encoders to create two views [23].

Unlike the aforementioned methods which often rely on pseudo labels generated from randomly initialized feature representations, our approach harnesses the strength of a pretrained self-supervised model. This foundational difference enables our clustering module to produce semantic pseudo labels, sidestepping the pitfalls of arbitrary cluster assignments.

## C. Self-training in semi-supervised learning and unsupervised domain adaptation

In semi-supervised learning scenarios, where there exists a limited labeled dataset complemented by a larger pool of unlabeled data, the objective is to harness the intrinsic structures or patterns prevailing within the unlabeled data to enhance the learning algorithm [24]. A prevalent approach to realizing this objective is self-training [25]-[30]. This method operates iteratively: initially, a model is trained using the available labeled data, serving as a basis to generate predictions on the unlabeled dataset. Predictions made with high confidence, termed pseudo labels, are integrated into the training set, forming an augmented dataset on which the model undergoes further training. This iterative cycle continues until convergence is achieved or a set number of iterations are completed. The core intent of self-training is to exploit the inherent but latent structures within the unlabeled data, thereby augmenting the model's capacity to generalize effectively.

In the context of unsupervised domain adaptation (UDA), the labeled data are derived from a specific source domain, whereas the unlabeled data are from a distinct, but related, target domain. The pivotal challenge lies in adeptly fine-tuning the model, which has been preliminarily trained on the source domain, ensuring its optimized performance when applied to the target domain by effectively utilizing the unlabeled target data. The self-training method, due to its intrinsic reliance on unlabeled data, finds substantial applicability in UDA, seamlessly aligning with its fundamental principles [31]–[34].

Numerous variants of the self-training technique have been innovated for semi-supervised learning and unsupervised domain adaptation. For instance, one variant employs the teacher-student architecture to impose a consistency regularization [35], [36]. Here, metrics such as the MSE or Kullback-Leibler divergence are commonly used to apply prior constraint assumptions on the unlabeled data. Central to consistency regularization is that the model's output remains robust to specific perturbations. Moreover, there exist models like deep co-training [37] and Tri-Net [38], based on the disagreement-based paradigm. These models foster the simultaneous training of multiple models, leveraging the disagreements among them as a critical aspect of the learning process.

In contrast to self-training, our proposed SSRL approach operates within a purely unsupervised setting, devoid of any reliance on labeled data. Unlike self-training, where the set of labels is predetermined, there is no prior knowledge of class counts or explicit label information in the unsupervised setting. The proposed SSRL method uses the teacher-student framework, and the teacher provides pseudo labels based on an online clustering mechanism. This dynamic mechanism fosters the creation of evolving clusters, which are adaptable and capable of undergoing refinements throughout the learning process. Thus, the SSRL method, with its capacity for dynamic clustering, promises enhanced performance and robustness in unsupervised learning scenarios.

## D. Speaker representation learning

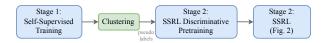
Before the emergence of deep learning, speaker representation learning primarily relied on statistical modeling techniques. Among these, the Gaussian Mixture Model - Universal Background Model (GMM-UBM) was widely used to model speaker features probabilistically [39]–[41]. This approach was later enhanced by i-vector representations, which leveraged factor analysis to map speaker characteristics into a low-dimensional space, improving speaker recognition and verification performance [42].

With the advent of deep neural networks, speaker modeling transitioned to data-driven feature extraction, enabling more robust and discriminative speaker embeddings. One of the earliest breakthroughs was the x-vector framework, which introduced Time-Delay Neural Networks (TDNNs) along with a statistics pooling layer to aggregate features at the utterance level [43], [44]. This significantly improved robustness to variable-length speech segments. Following this, ResNet-based architectures were applied to speaker modeling, incorporating convolutional layers to effectively capture local speaker features [45], [46]. Further advancements led to ECAPA-TDNN, which introduced channel-wise attention mechanisms and multi-scale feature learning, enhancing speaker discriminability while maintaining a compact model size [47]. More recently, Transformer-based architectures, such as Conformer, have been explored, integrating self-attention mechanisms with convolutional layers to better capture both global and local speaker characteristics [48]–[50].

A major paradigm shift in speaker modeling has been the adoption of large-scale self-supervised pre-trained models. Models such as wav2vec 2.0 [51], HuBERT [52], and WavLM [53] have been utilized as feature extractors and fine-tuned for



(a) Two-stage framework with iterative training.



(b) Improved two-stage framework with SSRL.

Fig. 2: A comparison of the proposed SSRL method with our previously proposed two-stage method with iterative training.

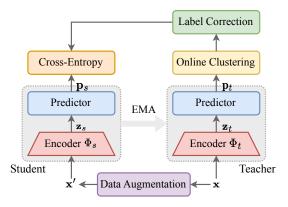


Fig. 3: The proposed self-supervised reflective learning (SSRL) method.

speaker-related tasks [1], [54], [55]. These approaches leverage self-supervised learning as a pretraining method, allowing for scalable and data-efficient speaker modeling. By learning from vast amounts of unlabeled speech data, these models significantly enhance generalizability and reduce reliance on manually labeled datasets.

## E. Two-stage iterative framework for unsupervised speaker representation learning

Multi-stage unsupervised learning frameworks have been widely adopted in various domains, including hyperspectral image processing [56]–[58], where structured priors and iterative refinements enhance representation quality. In speaker representation learning, a two-stage iterative framework has been proposed to leverage large-scale unlabeled data efficiently [8], [9], [59]. In the first stage, self-supervised methods are used to extract initial speaker embeddings and pseudo labels. The second stage involves an iterative discriminative training process to refine these embeddings.

To enhance stage one, more advanced self-supervised representation learning such as DINO are proposed [60]. Zhao et al. [61] further refined DINO with the Prototype Division method, effectively mitigating speaker confusion and enhancing overall performance. Additionally, Tao et al. [62] proposed a multimodal contrastive learning technique using diverse positive pairs by cross-referencing speech and face data, improving the robustness of speaker encoders.

A significant challenge in this framework is the presence of noisy pseudo labels. To address this, several methods have been developed. Tao et al. [63] introduced a technique that extracts reliable labels based on the neural network's fitting ability during training. Han et al. [64] and Zhou et al. [5] proposed using a Gaussian Mixture Model (GMM) to dynamically model loss distribution, distinguishing between reliable and unreliable labels, and correcting the unreliable ones using model predictions. Chen et al. [65] developed a method that coordinates information between audio and visual modalities through an "update by disagreement" strategy, improving pseudo label quality by leveraging inter-modal disagreements.

While these methods improve pseudo label quality, they typically require iterative training stages. Fang et al. [66] employed a label ensemble approach to smoothly correct noisy speaker labels by the exponential moving average of model predictions at each training epoch. Similarly, the approach presented in this paper dynamically improves pseudo labels at each epoch, eliminating the need for multiple training rounds and enhancing both efficiency and accuracy in speaker representation learning.

## III. METHODS

This section introduces the self-supervised reflective learning (SSRL) approach, which improves the two-stage iterative framework [8]. In the original two-stage framework, the first stage applies self-supervised representation learning and generates initial pseudo labels. The second stage consists of multiple training rounds, each comprising: (1) pseudo-label generation through clustering, and (2) discriminative training using these labels.

While maintaining the first stage unchanged, the proposed SSRL method replaces the multi-round process with continuous label refinement during a single training phase. As illustrated in Figure 2, SSRL requires an initialization step to ensure stable pseudo-label generation. This can be achieved either through brief discriminative training using Stage 1 pseudo labels, or by directly employing the Stage 1 self-supervised model as the encoder with a predictor initialized using pseudo cluster centroids. Following initialization, SSRL dynamically updates pseudo labels during training through its reflective learning mechanism.

Figure 3 illustrates the proposed SSRL method, with the detailed procedure outlined in Algorithm 1.

## Algorithm 1 Self-Supervised Reflective Learning (SSRL)

```
Require: Unlabeled dataset \mathcal{D} = \{\mathbf{x}_i | i=1,\cdots,N\}; Initial pseudo labels \mathcal{Y} = \{y_i | i=1,\cdots,N\}
      1, \dots, N; Teacher encoder \Phi_t with parameters \phi_t; Teacher predictor h_t with parameters
      \psi_t; Student encoder \Phi_s with parameters \phi_s; Student predictor h_s with parameters \psi_s
  1: procedure ReflectiveLearning(\mathcal{D}, \mathcal{Y})
           Train student network (\Phi_s and h_s) with dataset \{\mathcal{D}, \mathcal{Y}\} for E_1 epochs
  2:
           Initialize \Phi_t with \Phi_s and h_t with h_s
  3:
                                                            ▷ Initialize empty pseudo label queue for all samples
  4.
           Q_i \leftarrow \text{Queue}(\text{length} = L)
           p_{\text{clean}}(\ell_{t,i}) \leftarrow 1
                                                           ▶ Initialize clean label probability to 1 for all samples
  5:
  6:
           for epoch in 1 to E_2 do
  7:
                 for batch \mathcal{B} = \{\mathbf{x}_i | i = 1, \cdots, B\} in \mathcal{D} do
  8:
  9:
                      Crop a short segment for each training sample \mathcal{B}_s = \{\mathbf{x}_i'\}
                      Apply data augmentation to \mathcal{B}_s
 10:
                      \begin{aligned} \mathbf{p}_{s,i} \leftarrow \operatorname{softmax}(h_s \circ \Phi_s(\mathbf{x}_i')) \\ \mathcal{L} \leftarrow -\frac{1}{B} \sum_{i=1}^{B} p_{\operatorname{clean}}(\ell_{t,i}) \log p_s(y_i|\mathbf{x}_i') \end{aligned}
                                                                                                                    11:

    ▶ Training loss

12:
13:
14:
                      Crop a long segment for each training sample \mathcal{B}_t = \{\tilde{\mathbf{x}}_i\}
                      \mathbf{p}_{t,i} \leftarrow \operatorname{softmax}(h_t \circ \Phi_t(\tilde{\mathbf{x}}_i))

    ▶ Teacher prediction

15:
                                                                                                                ▷ Online clustering
                      y_i \leftarrow \text{clustering}(\mathbf{p}_{t,i})
16:
                      Enqueue y_i to Q_i
17:
                      y_i \leftarrow \text{mode of labels in } Q_i
18:

    ▶ Label correction

                      \ell_{t,i} \leftarrow -\log p_t(y_i|\tilde{\mathbf{x}}_i)
                                                                                              > Cross entropy loss of teacher
19:
20:
                      Update student parameters using gradients from \mathcal{L}
21:
                      \phi_t \leftarrow \lambda \phi_t + (1 - \lambda) \phi_s
                                                                                          ▶ EMA update of teacher encoder
22:
                      \psi_t \leftarrow \lambda \psi_t + (1 - \lambda) \psi_s

    ► EMA update of teacher predictor

23:
24:
                 end for
25:
                 Fit \{\log \ell_{t,i} | i = 1, \cdots, N\} with a GMM

    Noisy label modeling

26:
                 Update p_{\text{clean}}(\ell_{t,i}) using GMM
27:
           end for
28:
29: end procedure
```

## A. Self-supervised knowledge distillation

At the heart of our approach is the self-supervised knowledge distillation technique. Given an unlabeled dataset, the teacher network generates cluster assignments which guide the training of the student network. The teacher encoder, represented as  $\Phi_t(\cdot)$ , transforms the data sample  $\mathbf{x}$  into a D-dimensional feature representation  $\mathbf{z}_t \in \mathbb{R}^D$ :

$$\mathbf{z}_t = \Phi_t(\mathbf{x}) \tag{1}$$

Subsequently, a linear predictor,  $h_t(\cdot)$ , is employed to compute the probability distribution over K clusters via a softmax operator. Let  $p_t(k|\mathbf{x})$  denotes the posterior probability that the sample  $\mathbf{x}$  belongs to the  $k^{\text{th}}$  cluster, the vector  $\mathbf{p}_t$  aggregates these probabilities for all K clusters:

$$\mathbf{p}_t = \operatorname{softmax}(h_t(\mathbf{z}_t)) = \operatorname{softmax}(h_t \circ \Phi_t(\mathbf{x}))$$
 (2)

where  $p_t(k|\mathbf{x})$  is the  $k^{\text{th}}$  element of  $\mathbf{p}_t$ . An online clustering mechanism then extracts cluster assignments  $y \in \{1, 2, \dots, K\}$  from  $\mathbf{p}_t$  for the training sample  $\mathbf{x}$ .

Following a parallel structure, the student encoder  $\Phi_s(\cdot)$ , coupled with the student predictor  $h_s(\cdot)$  – analogous in archi-

tecture to the teacher - produce the feature  $\mathbf{z}_s$  and the class prediction  $\mathbf{p}_s$  from another view of the same input  $\mathbf{x}'$ . The student model's training utilizes the cross-entropy loss, under the supervision of the pseudo label y derived from the teacher:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \log p_s(y_i | \mathbf{x}_i')$$
 (3)

where N represents the number of data samples in a training batch.

1) Enhancing the student's model capacity: Drawing inspiration from the noisy student method in semi-supervised learning [30], our approach amplifies the student's modeling capacity by imposing noise into the training samples during the student's training. Specifically, a short segment is extracted from the training utterance, followed by data augmentation techniques introducing background noise or convolutional reverberation to this segment. Consequently, the student model processes these augmented snippets. The teacher model, on the other hand, processes a longer clip of the same utterance in its unaltered form, facilitating the generation of stable pseudo labels. Morever, other deep neural network training

strategies can further improve the student's model capacity. For instance, employing dropout can mitigate the risk of overfitting to the imprecise pseudo labels [67]. Another approach involves the use of angular margin-based cross entropy [68] as a loss function, fostering the student model to capture a more discerning feature space.

2) Teacher model update mechanism: Traditional knowledge distillation typically employs a teacher network, trained with labeled data and possessing superior model capacity. However, under self-supervised settings, acquiring such a pretrained teacher model is not feasible. We hypothesize that the student model's capacity undergoes enhancement after each training cycle, courtesy of noisy student training. As such, an advanced teacher model can be obtained by ensembling student models from previous training steps. In specific terms, we employ an EMA technique on the student's parameters to refine the teacher model [12], [13], [69] . Denoting the parameters of student encoder  $\Phi_s(\cdot)$  as  $\phi_s$  and the parameters of student predictor  $h_s(\cdot)$  as  $\psi_s$ , the teacher's parameters  $\phi_t$  and  $\psi_t$  undergo an update as:

$$\phi_t \leftarrow \lambda \phi_t + (1 - \lambda)\phi_s$$
  
$$\psi_t \leftarrow \lambda \psi_t + (1 - \lambda)\psi_s$$
 (4)

where  $\lambda \in [0,1)$  serves as a momentum coefficient. Through the EMA update mechanism, the teacher consistently outperforms the student during the training process, thereby facilitating the student's learning by providing pseudo labels of higher quality.

#### B. Online clustering

In the cluster assignment task, the objective is to maximize the alignment of the cluster assignments  $q(k|\mathbf{x}_i)$  with the predicted class probabilities  $p_t(k|\mathbf{x}_i)$  provided by a teacher model, ensuring that each data point is assigned to the cluster where it best fits according to these predictions.

$$\max_{q} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|\mathbf{x}_{i}) p_{t}(k|\mathbf{x}_{i})$$
subject to  $\forall k : q(k|\mathbf{x}_{i}) \in \{0,1\}$  and  $\sum_{k=1}^{K} q(k|\mathbf{x}_{i}) = 1$ 

To address this, we explore two online clustering methodologies:

1) Direct maximum probability assignment: The most intuitive method generates cluster assignment based on the highest predicted probability class from the teacher:

$$q(k|\mathbf{x}_i) = \delta\left(k - \arg\max_{j} p_t(j|\mathbf{x}_i)\right)$$
 (6)

Here,  $\delta(k-\arg\max_j p_t(j|x_i))$  is the Kronecker delta function, defined as 1 when  $k=\arg\max_j p_t(j|x_i)$  and 0 otherwise. Essentially, each data sample is allocated to the cluster corresponding to the class that the teacher model is most confident in.

2) Cluster assignment through optimal transport: Drawing inspiration from SeLa [19], we introduce an added constraint to the objective in Equation 5, ensuring that the N training samples are distributed evenly across the K clusters:

$$\max_{q} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|\mathbf{x}_{i}) p_{t}(k|\mathbf{x}_{i})$$
subject to  $\forall k : q(k|\mathbf{x}_{i}) \in \{0,1\}, \sum_{i=1}^{N} q(k|\mathbf{x}_{i}) = \frac{N}{K}$ 

$$(7)$$

Such constraints ensure a distinct label for every data point and a uniform distribution of the N samples over the K classes, preventing identical pseudo labeling for all training samples.

Building on the perspective of SeLa [19], the optimization problem depicted in Equation 7 can be mapped to an optimal transport problem [70]. To understand this, let's define P as the  $K \times N$  matrix where  $P_{ki} = \frac{1}{N} p_t(k|\mathbf{x}_i)$ , and Q as the  $K \times N$  matrix of assigned joint probabilities between a and b with  $Q_{ki} = \frac{1}{N} q(k|\mathbf{x}_i)$ . Following the notation in [70], Q can be conceptualized as an element of the transportation polytope:

$$U(\mathbf{r}, \mathbf{c}) := \left\{ Q \in \mathbb{R}_{+}^{K \times N} | Q \mathbb{1} = \mathbf{r}, Q^{T} \mathbb{1} = \mathbf{c} \right\}$$
(8

where 1 is the vector of all ones of appropriate dimension. Based on the given constraints, we get:

$$\mathbf{r} = \frac{1}{K} \cdot \mathbb{1}; \quad \mathbf{c} = \frac{1}{N} \cdot \mathbb{1}$$
 (9)

Given matrices P and Q, the objective function in Equation 7 can be recast as:

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|\mathbf{x}_i) p_t(k|\mathbf{x}_i) = \langle Q, P \rangle$$
 (10)

where  $\langle \cdot \rangle$  is the Frobenius dot-product between two matrices. Consequently, Equation 7 can be translated into an optimal transport problem between  $\mathbf{r}$  and  $\mathbf{c}$  with a cost of -P:

$$\min_{Q \in U(\mathbf{r}, \mathbf{c})} \langle Q, -P \rangle \tag{11}$$

To expedite the optimal transport solver, an entropic constraint was integrated into the classical optimal transport problem as introduced by Cuturi [70]. This regularization of the problem is defined by:

$$U_{\alpha}(\mathbf{r}, \mathbf{c}) := \{ Q \in U(\mathbf{r}, \mathbf{c}) \mid \mathrm{KL}(Q \| \mathbf{r} \mathbf{c}^T) \le \alpha \}$$
 (12)

where KL represents the Kullback-Leibler divergence. Given the concavity of entropy, we have  $U_{\alpha}(\mathbf{r}, \mathbf{c}) \subset U(\mathbf{r}, \mathbf{c})$ . Consequently, the optimal transport problem (as shown in Equation 11) is reframed as:

$$\min_{Q \in U_{\alpha}(\mathbf{r}, \mathbf{c})} \langle Q, -P \rangle \tag{13}$$

Introducing a Lagrange multiplier for the entropy constraint, we arrive at the dual optimization problem:

$$\min_{Q \in U(\mathbf{r}, \mathbf{c})} \langle Q, -P \rangle + \frac{1}{\lambda} KL(Q \| \mathbf{r} \mathbf{c}^T)$$
 (14)

From the Lagrangian of Equation 14, we can express the minimizer of Equation 14 as:

$$Q = \operatorname{diag}(\mathbf{u})e^{\lambda P}\operatorname{diag}(\mathbf{v}) \tag{15}$$

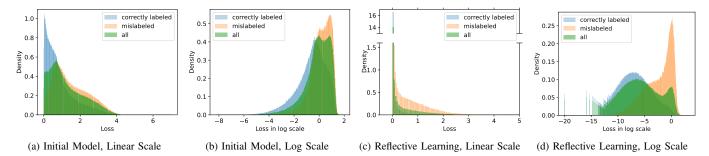


Fig. 4: Histogram of the cross entropy loss between the model's prediction and the pseudo label. (a) and (b) are produced by the model trained with the initial pseudo label without applying the proposed SSRL approach. (c) and (d) are produced by the teacher model after 30 epochs of SSRL.

In the equation, exponentiation is carried out element-wise. Additionally,  $\mathbf{u}$  and  $\mathbf{v}$  are two non-negative vectors that serve as scaling coefficients, ensuring the resulting matrix Q adheres to the probability matrix standards.

The Sinkhorn-Knopp algorithm is employed to determine the optimal Q. This algorithm iteratively adjusts the rows and columns of the matrix utilizing diagonal matrices until a convergence point is reached:

$$\forall k : \mathbf{u}_k \leftarrow \frac{\mathbf{r}_k}{[e^{\lambda P} \mathbf{v}]_k}; \quad \forall i : \mathbf{v}_i \leftarrow \frac{\mathbf{c}_i}{[\mathbf{u}^T e^{\lambda P}]_i}$$
 (16)

To generate pseudo labels using the Sinkhorn-Knopp algorithm, we employ a batched approach. Specifically, we accumulate the matrix P over M batches with batch size B, ensuring total number of training samples  $N=M\times B$  is larger than the number of cluster K. Every M batches, we update the cluster assignments utilizing the Sinkhorn-Knopp algorithm. This method provides a computationally efficient way to handle large datasets, ensuring consistent and optimized pseudo-label assignments in line with the teacher's predictions.

Either with direct maximum probability assignment or optimal transport-based cluster assignment, the assigned clusters are determined based on the teacher model's predictions, without constraints ensuring that every output class will receive data samples. As training progresses, clusters with extremely low prediction confidence shrink and eventually disappear. This phenomenon is observed in our experiments, as described later in Section V-B and Figure 6, where the number of active clusters gradually decreases during training until a stable count is reached.

SSRL naturally maintains stable and meaningful cluster assignments throughout training. Starting from initial pseudo labels, the online clustering continuously refines assignments as the teacher model's discrimination ability improves. When a sample's current assignment becomes suboptimal, the teacher model reassigns it based on learned representations. Online clustering works organically with EMA updates - EMA ensures smooth model evolution while preserving previous knowledge, enabling stable and consistent refinements to pseudo labels.

## C. Pseudo label correction

To further refine the pseudo label generation process, we introduce a label correction mechanism employing a pseudo label queue. This queue retains a history of pseudo labels previously generated by the teacher model for each training sample. With a predetermined fixed length L, the queue ensures consideration only of the most recent L predictions. To filter out sporadic or outlier predictions and cultivate a robust pseudo label, we employ a statistical mode evaluation of the labels within the queue. This ensures that the most frequently occurring label in the recent history is selected as the final pseudo label, thereby enhancing the reliability of the label assignment and mitigating the effects of transient erroneous predictions.

## D. Noisy label modeling

To mitigate the challenges posed by noisy pseudo labels, our framework incorporates a strategy to model label noise following the approach in [14]. Prior research [14] has shown that deep neural networks (DNNs) tend to learn correctly labeled samples first before gradually fitting to mislabeled ones. As a result, mislabeled samples typically exhibit higher loss values compared to correctly labeled ones, allowing us to leverage this property for noise modeling.

Figure 4 shows an illustration of such behavior. Since the pseudo label is estimated in the self-supervised setting, we do not have the ground truth references for the correct and incorrect labels. To estimate this noisy label information, we employ the Hungarian algorithm, mapping the pseudo labels to the ground truth labels. Figure 4 exhibit a bimodal distribution with two distinct peaks of the logarithmically scaled losses. By modeling this loss distribution, we can effectively segregate accurately labeled data from the mislabeled, which then aids in computing cleaner label probabilities for the training set.

To achieve this, we use a two-component GMM to model the logarithmically scaled losses generated by the teacher model. Mathematically, the mixture model can be expressed as:

$$p(\ell_t) = \pi \cdot \mathcal{N}(\log(\ell_t); \mu_1, \sigma_1^2) + (1 - \pi) \cdot \mathcal{N}(\log(\ell_t); \mu_2, \sigma_2^2)$$
(17)

For sample  $\mathbf{x}_i$ ,  $\ell_{t,i}$  represents the cross entropy loss between the teacher's prediction and its pseudo label. The term  $p(\ell_t)$ 

represents the probability distribution of  $\log(\ell_t)$ . The coefficient  $\pi$  is the mixture weight, and  $\mathcal{N}(\log(\ell_t); \mu, \sigma^2)$  is the Gaussian distribution parameterized by mean  $\mu$  and variance  $\sigma^2$ .

The GMM aids in distinguishing between the loss distributions of clean labels and those of noisy labels. After establishing this loss distribution model, a clean label probability is assigned to each training sample as:

$$p_{\text{clean}}(\ell_t) = \frac{\pi \cdot \mathcal{N}(\log(\ell_t); \mu_1, \sigma_1^2)}{p(\ell_t)}$$
(18)

Given that samples with clean labels yield lower losses, the Gaussian component  $\mathcal{N}(\log(\ell_t); \mu_1, \sigma_1^2)$  associated with these samples has a smaller mean, i.e.,  $\mu_1 < \mu_2$ . Utilizing the clean label probability,  $p_{\text{clean}}(\ell_t)$ , the final loss is adjusted, directing the model to give greater emphasis to samples deemed to have accurate labels:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} p_{\text{clean}}(\ell_{t,i}) \log p_s(y_i|\mathbf{x}_i')$$
 (19)

Combining all the methods discussed above, Algorithm 1 presents the complete procedure of the proposed SSRL method. Here, we apply the direct maximum probability assignment as the online clustering method. It can easily be extended to the optimal transport method.

#### IV. EXPERIMENTAL SETUPS

#### A. Data

The experiments are conducted on the VoxCeleb dataset [71], [72]. For model training, we use the development set of VoxCeleb 2, which contains 1,092,009 audio files from 5,994 speakers. While speaker identity labels are available, they are only used for experimental analysis and not for model training.

For evaluation, we report speaker verification results using three trial lists from the VoxCeleb 1 dataset as defined in [72]:

- VoxCeleb 1-O: The original trial list with 37,720 trials from 40 speakers.
- VoxCeleb 1-E: An extended trial list with 581,480 trials from 1,251 speakers.
- VoxCeleb 1-H: A hard trial list with 552,536 trials from 1,190 speakers, where all test pairs share the same language and gender.

## B. Data Augmentation

Data augmentation is effective for deep speaker representation learning in both supervised learning [73] and contrastive self-supervised learning [74]–[76]. We utilized two primary strategies:

- Additive noise augmentation: The MUSAN dataset [77] was used as our noise source, adding ambient noise, musical sounds, and babble noise to our audio files. Babble noise was generated by merging three to eight separate speech files from the MUSAN dataset, with signal-to-noise ratios (SNR) ranging from 0 to 20 dB.
- Convolutional reverberation noise augmentation: We used 40,000 simulated room impulse responses (RIR) from small to medium-sized rooms, as described in [78].

To maintain variability during training, we applied on-the-fly data augmentation. In SSRL training, the student network was trained with two-thirds of the data augmented utterances, while the teacher network used unaltered speech data.

## C. Implementation details

We evaluate the proposed methods on two different network architectures for speaker representation learning: ResNet [79] and ECAPA-TDNN [47]. The baseline method used for comparison is the two-stage iterative framework. For each network architecture, a supervised model is trained to serve as a reference point (upper bound) for model performance, using the same training hyperparameters as those in the second stage of the two-stage iterative framework.

1) ResNet – two-stage iterative framework: We first use the two-stage iterative framework trained on ResNet [79] as the baseline, following our previous research on the two-stage iterative framework [8], [9].

In the first stage, we apply contrastive self-supervised learning (CSL) [76] to learn speaker representations. In the second stage (iterative training), initial pseudo labels for the training dataset are generated using K-means clustering on speaker embeddings from CSL. The number of clusters is set to 6,000, the same as in [9], where it was determined using the elbow method. For network architecture, hyperparameters, and other training details, readers can refer to [9].

2) ResNet – improved two-stage framework with SSRL: For the improved two-stage framework with SSRL trained on ResNet, the first stage remains the same as in the two-stage iterative framework. To initiate second-stage training, the number of clusters for K-means is set to 8,000, which is higher than the 6,000 clusters used in the two-stage iterative framework. This adjustment is made for two reasons: (1) The elbow method identifies a reasonable cluster count between 5,000 and 8,000 [9]. (2) As discussed in Section III-B, the cluster count naturally decreases due to the online clustering process. Setting a higher initial cluster count ensures sufficient granularity, allowing the model to refine pseudo labels without collapsing clusters too early.

In the second stage, to initialize SSRL training, the ResNet-based speaker embedding network is trained for 55 epochs with initial pseudo labels. A cosine annealing scheduler adjusts the learning rate from 1e-3 to 1e-5, including a 5-epoch warm-up phase. The batch size is set to 512, and the Adam optimizer is applied.

During the SSRL training phase, audio waveforms are cropped to 2 seconds for the student model and 6 seconds for the teacher model. The student network is trained for 100 epochs using the Adam optimizer, with the learning rate scheduled via cosine annealing from 5e-4 to 1e-5. The loss function used is cross entropy. The pseudo label queue length is set to 5 unless stated otherwise. The EMA momentum parameter, denoted as  $\lambda$  in Equation 4, linearly increases from 0.999 to 0.9999 during SSRL training.

3) ECAPA-TDNN – two-stage iterative framework: To compare with other studies, we also adopt the ECAPA-TDNN-based speaker embedding network [47] as an alternative backbone.

TABLE I: Comparison of two self-supervised pretrained models. EER is evaluated on VoxCeleb 1-O; labeling metrics are based on k-means clustering with 8,000 clusters.

	l Network Architecture	#Parameters	EER↓	NMI ↑	Accuracy ↑	Purity†
CSL	ResNet	1.37M	8.86%	0.7744	36.87%	55.32%
DINO	ECAPA-TDNN	63.65M <sup>1</sup>	2.94%	0.9319	65.26%	88.52%

For the first stage self-supervised training, ECAPA-TDNN speaker embedding network [47] is pretrained with DINO [13]. Following the structure in [2], the ECAPA-TDNN network has channels sequenced as 1024, 1024, 1024, 1024, and 3072 across the initial TDNN layer and four TDNN blocks. After the ECAPA-TDNN encoder, we use attentive statistical pooling followed by a 512-dimensional fully connected layer for speaker embeddings. The DINO projection head includes four fully connected layers with hidden dimensions of 2048, 2048, 8192, and 256, ending with a 65536-dimensional weight-normalized fully connected layer. We employ multicrop data augmentation, giving the EMA teacher two 4-second data-augmented views and the student four 2-second data-augmented views for each training sample.

The DINO pretraining uses a stochastic gradient descent (SGD) optimizer over 100 epochs, with a cosine annealing scheduler modulating the learning rate from 0.2 to 1e-5, including a 10-epoch warm-up phase. The temperature hyperparameters for cross-entropy are set to 0.04 for the teacher and 0.1 for the student. For more detailed training procedures, refer to [13] and [2].

The comparison of different first stage models used in this work can be found in Table I. Unlike random initialization, stage 1 provides a structured representation for clustering, enabling the first clustering round to generate more reliable pseudo labels. This improves the quality of subsequent second stage training, ensuring the model refines meaningful speaker representations rather than noise.

In the second stage of iterative training, pseudo labels are generated by applying K-means clustering to the speaker embeddings from the previous training round, targeting 8,000 clusters. Each training round employs the Adam optimizer with a batch size of 480, and the learning rate is managed by a cosine annealing scheduler, transitioning from 1e-4 to 1e-5 over 40 epochs.

To ensure stable training, we initialize the encoder's parameters with DINO pre-trained parameters for the first training round. In subsequent training rounds, the encoder retains the parameters from the previous training round. The predictor, i.e., the final linear layer for speaker classification, is reinitialized using K-means cluster centers.

4) ECAPA-TDNN – improved two-stage framework with SSRL: For the improved two-stage framework with SSRL

<sup>1</sup>The ECAPA-TDNN encoder has a total of 22.73 million parameters. The DINO projection head contains 40.92 million parameters. The projection head is only used during DINO training; speaker embeddings are extracted from the output of the ECAPA-TDNN encoder.

trained on ECAPA-TDNN, the first stage remains the same as in the two-stage iterative framework.

In the second stage, we directly initialize both the student and teacher networks using DINO-pretrained parameters and apply SSRL. The predictor, a single linear layer, has its weights initialized with the 8,000 K-means cluster centers, while the biases are set to zero. The training batch size for the ECAPA-TDNN model is 480, and other training configurations for SSRL remain the same as those for the ResNet-based pipeline. For the training objective, in addition to cross-entropy loss, we train another ECAPA-TDNN with SSRL using the additive angular margin (AAM) loss [68] to further enhance the model's capacity. The AAM loss margin is set to 0.2, and the scaling factor is 32.

#### D. Evaluation metric

- 1) Speaker verification evaluation: We assess the effectiveness of speaker verification systems by measuring the equal error rate (EER) and the minimum detection cost (minDCF) [80]. For the detection cost function, we configure the parameters as  $C_{\rm Miss} = 1$ ,  $C_{\rm FA} = 1$ , and  $P_{\rm Target} = 0.05$ .
- 2) Clustering evaluation: To evaluate clustering quality, we use three metrics as outlined in [81] and [9]:
  - Normalized mutual information (NMI): This metric measures the agreement between our clustering and the true data grouping, providing a score between 0 and 1, where 0 indicates no match and 1 indicates a perfect match.
  - Clustering accuracy: We evaluate accuracy by comparing pseudo labels to ground truth labels, using the Hungarian algorithm [82] to establish label correspondence.
  - Mean maximal purity per cluster: This metric assesses the semantic purity of each pseudo cluster in comparison to the ground truth labels:

purity = 
$$\frac{1}{K} \sum_{k \in K} \max \left( p\left(y | \hat{y} = k\right) \right)$$
 (20)

where K is the number of pseudo clusters,  $\hat{y}$  represents a pseudo cluster and  $p\left(y|\hat{y}=k\right)$  is the distribution of ground-truth labels within pseudo cluster k.

## V. EXPERIMENTAL RESULTS

This section evaluates the improved two-stage framework with SSRL in terms of speaker verification performance and pseudo-labeling robustness. We also investigates the contributions of different individual components in the proposed SSRL method.

#### A. Speaker verification performance

1) Comparing SSRL with iterative training in two-stage framework: The primary objective of our experiments is to compare the proposed SSRL method with the iterative training in the two-stage framework. Table II shows that the SSRL-trained ResNet model achieves an EER of 2.39% on the VoxCeleb 1-O trial in just one training round, surpassing the fifth-round model in the two-stage iterative framework (2.74%).

TABLE II: ResNet results: speaker verification performance (minDCF and EER[%]) on VoxCeleb 1 test trials.

Model		VoxCele	b 1-0	VoxCele	b 1-E	VoxCele	b 1-H
		minDCF	EER	minDCF	EER	minDCF	EER
Supervised	l	0.097	1.51	0.102	1.59	0.178	3.00
Two- Stage Iterative Framework [9]	CSL (Stage 1) Round 1 Round 2 Round 3 Round 4 Round 5	0.508 0.257 0.214 0.190 0.184 0.173	8.86 3.64 2.99 2.93 2.85 2.74	0.570 0.299 0.234 0.214 0.202 0.201	10.15 4.11 3.41 3.23 3.16 3.08	0.710 0.459 0.362 0.334 0.314 0.311	16.20 7.68 6.25 5.85 5.54 5.48
SSRL (one round)		0.163	2.39	0.183	2.63	0.285	4.74

TABLE III: ECAPA-TDNN results: Speaker verification performance (minDCF and EER[%]) on VoxCeleb 1 test trials.

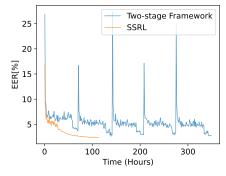
Model		VoxCele	b 1-C	VoxCele	b 1-E	VoxCele	b 1-H
1,10001				minDCF	EER	minDCF	EER
Supervised		0.143	1.88	0.136	1.98	0.237	3.96
Supervised + AAM		0.075	0.99	0.081	1.22	0.144	2.35
Two-	DINO (Stage 1)	0.202	2.94	0.218	3.05	0.364	5.88
Stage	Round 1	0.181	2.49	0.183	2.73	0.288	5.01
Iterative	Round 2	0.174	2.34	0.180	2.66	0.282	4.90
Framework	Round 3	0.177	2.28	0.184	2.70	0.288	4.95
SSRL (one round)		0.131	1.77	0.127	1.85	0.217	3.59
SSRL (one round) + AAM		<b>0.101</b>	<b>1.25</b>	<b>0.098</b>	<b>1.47</b>	<b>0.174</b>	<b>2.86</b>

Similarly, for the ECAPA-TDNN model in Table III, the SSRL method demonstrates superior performance with an EER of 1.77% in one training round, compared to 2.28% EER from the third-round model in the two-stage iterative framework. The integration of AAM loss in SSRL suggests even more potential, with EER dropping to 1.25%.

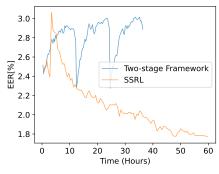
The supervised results in Tables II and III serve as upper bounds for model performance. Compared to the supervised model, both self-supervised methods do not surpass supervised performance. However, SSRL significantly reduces the performance gap. For the ResNet-based pipeline, SSRL achieves an EER of 2.39% on VoxCeleb 1-O, compared to 2.74% for the best model in two-stage iterative framework. For ECAPATDNN, SSRL achieves 1.77% EER, improving upon the two-stage iterative framework's best result of 2.28%, bringing it closer to the supervised model's 1.88% EER.

The superiority of the SRRL second stage over the iterative second stage can be ascribed to its robust pseudo-labeling mechanism. Unlike the two-stage iterative framework which employs static pseudo labels for a whole training round, SSRL benefits from dynamically updated labels via self-supervised knowledge distillation and online clustering. This continuous refinement ensures the student model always benefits from the latest supervision signals, eliminating the 'stale' label problem observed in the two-stage iterative framework. Furthermore, SSRL's incorporation of a pseudo label queue and noisy label modeling techniques further improve the reliability and robustness of the pseudo labels, enhancing overall model performance.

2) Efficiency of the SSRL approach: Unlike the iterative second stage which requires multiple training rounds, SSRL



(a) ResNet-based pipeline



(b) ECAPA-TDNN-based pipeline

Fig. 5: Comparison of training time vs. EER for iterative training and SSRL training in the two-stage framweork

introduces a more streamlined approach. This eliminates the need for iterative training, leading to improved efficiency. This is illustrated in Figure 5, which compares the EER over training time between the iterative second stage and SSRL second stage in the two-stage framework.<sup>2</sup>

For the ResNet-based pipeline, it is apparent from the visualization that SSRL achieves quicker convergence and maintains a more stable EER than iterative training. The iterative approach exhibits fluctuations due to its clustering process, where pseudo labels are re-generated between rounds, requiring random initialization of the final linear layer. This causes temporary EER spikes before stabilization. In contrast, SSRL continuously refines pseudo labels within a single training round, enabling smoother training dynamics and improved efficiency. These advantages demonstrate SSRL's potential for applications where training time and computational resources are critical considerations.

The ECAPA-TDNN-based pipeline fails to converge and experiences overfitting during each training round in the iterative second stage. The verification performance (EER) shows minimal improvement before rapidly deteriorating. This occurs because we initialize the network using parameters from the previous round and k-means centers for the final linear layer, causing rapid data fitting in early epochs. Due to

<sup>2</sup>All models are trained on two NVIDIA GeForce RTX 3090 GPUs. The estimated training time focuses solely on ideal conditions, accounting only for the forward and backward propagation time (model training time of a single batch). It excludes time allocations for data loading, preprocessing pipeline, model validations, and procedures like k-means clustering and GMM modeling. These processes, being brief in nature, are considered negligible.

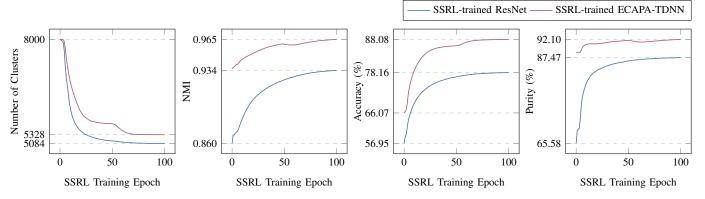


Fig. 6: Evolution of pseudo labeling across training epochs during the SSRL training phase.

TABLE IV: Comparison of the proposed SSRL method with two-stage iterative framework variants. EERs [%] from Vox-Celeb 1-O test trial; all models use ECAPA-TDNN. 'Filter' denotes mislabeled sample filtering, 'LC' for label correction.

Method	Loss	Filter	LC	Other	#Rounds	Stage 1 EER	EER
Thienpondt et al. [59]	AAM	-	-	-	7	7.3	2.1
Mun et al. [83]	AAM	-	-	score norm	5	3.65	1.66
Tao et al. [63]	AAM	$\checkmark$	-	-	5	7.36	1.66
Han et al. [64]	AAM	$\checkmark$	$\checkmark$	-	5	6.16	1.47
Tao et al. [62]	AAM	-	-	audio-visual	$\geq 2$	2.89	1.44
Chen et al. [65]	AAM	-	-	audio-visual	1 7	7.16	1.27
Chen et al. [84]	AAM	-	$\checkmark$	WavLM	5	-	1.25
SSRL (proposed)	CE	-	<b>√</b>	-	1	2.94	1.77
SSRL (proposed)	AAM	-	$\checkmark$	-	1	2.94	1.25
SSRL (proposed)	AAM	-	$\checkmark$	WavLM	1	2.94	1.04

this overfitting tendency, we terminated training after the third round. These results suggest that with a strong initialization (DINO pretrained model), the two-stage iterative framework cannot substantially enhance performance. In contrast, the proposed SSRL method dynamically adjusts clustering, leading to further performance improvements even when starting with a relatively well-pretrained model.

3) Comparative analysis with other two-stage iterative framework variants: In Table IV, the performance of the proposed SSRL method is compared with various two-stage iterative framework variants, all leveraging the ECAPA-TDNN model. A remarkable observation is the efficiency and efficacy of SSRL when trained with the AAM loss: it surpasses all other methods, achieving superior performance within a single training round.

Two comparisons deserve special mention. First, when contrasted with the work of Chen *et al.* [65] – which incorporates an additional visual modality during training – our SSRL method delivers performance on par, even though it relies exclusively on audio information. Secondly, another variant from Chen *et al.* [84] makes use of a subset of WavLM [53], a large self-supervised speech model trained on extensive data, for feature extraction. Our SSRL approach, devoid of any large-scale pre-trained model, emerges with a similar performance.

To further evaluate our approach, we integrated WavLM as a

TABLE V: Pseudo labeling performance on training data.

Model	Method	#Clusters	NMI	Accuracy	Purity
ResNet	Iterative round 5 SSRL	$6000 \\ 8000 \rightarrow 5085$			
ECAPA	Iterative round 3 SSRL	8000 8000→5328			

feature extractor alongside the proposed SSRL method. Specifically, we extracted features from every layers of WavLM-Large encoder and combined them using a learnable weighted sum to create composite features for input to ECAPA-TDNN. Our training strategy involved initially freezing the WavLM parameters during early epochs, followed by gradual finetuning. This integration proved highly effective: the system achieved 1.04% EER on Vox1-O, representing a significant 16.8% improvement over the SSRL model trained with Melfilterbank features (1.25% EER).

#### B. Pseudo labeling performance

Table V details the pseudo labeling performance of the ResNet and ECAPA models on the training data. The SSRL method consistently shows superior metrics across both model architectures. For instance, with the ResNet model, the SSRL technique achieved an accuracy of 78.12% compared to the 68.93% from the two-stage iterative framework in its fifth training round. This superior performance can be attributed to the online clustering achieved through self-supervised knowledge distillation, coupled with additional strategies to enhance the quality of pseudo labels.

In Figure 6, the evolution of pseudo labeling throughout the training epochs using the SSRL method is depicted. As observed, during the SSRL training process, there's a consistent reduction in the number of clusters across epochs until a stable count is reached. For the ResNet-based SSRL, this stable number is 5084, whereas for the ECAPA-TDNN-based SSRL, it's 5328. For reference, the training data contains a total of 5994 speakers. These observations indicate that the SSRL method is adept at filtering out pseudo clusters that have lower confidence, thereby progressively optimizing the labeling performance. Moreover, metrics such as NMI,

TABLE VI: Performance comparison of the SSRL approach with different component configurations.  $p_{\text{clean}}$  represents the proposed noisy label modeling method. K represents the converged number of clusters. The actual cluster counts is 5994.

Online	EMA	Label	$p_{ m clean}$	Verifi	cation EER	[%] ↓		Pseudo I	abeling	
Clustering		Queue	r cican	Vox1-O	Vox1-E	Vox1-H	NMI ↑	Acc ↑	Purity ↑	K
argmax	<b>√</b>	<b>√</b>	<b>√</b>	2.39	2.63	4.74	0.9333	78.12%	87.42%	5085
argmax	X	$\checkmark$	$\checkmark$	2.48	2.97	5.31	0.9261	77.23%	87.76%	4943
argmax	$\checkmark$	Х	$\checkmark$	2.51	2.68	4.82	0.9297	77.31%	86.96%	4801
argmax	$\checkmark$	$\checkmark$	X	2.76	2.94	5.29	0.9300	75.55%	86.55%	6152
argmax	Х	X	Х	5.19	6.52	11.73	0.8567	66.88%	90.64%	4306
Sinkhorn	$\checkmark$	$\checkmark$	$\checkmark$	2.41	2.57	4.61	0.9402	79.26%	84.45%	5974

accuracy, and maximal purity per cluster show that with each passing epoch, the SSRL-trained models fine-tune their performance, reflecting continuous improvement.

#### C. Ablation study

The SSRL approach employs components designed to enhance the model's performance on the unlabeled dataset. To inspect the contributions of these components, we conduct ablation studies on the ResNet-based SSRL, shown in Table VI.

1) Speaker verification performance analysis: When the EMA update for the teacher model is integrated into SSRL, we observe an improvement in speaker verification performance, with EERs of 2.39%, 2.63%, and 4.74% across the VoxCeleb test trials. In contrast, the model without the EMA update shows higher EERs of 2.48%, 2.97%, and 5.31%, respectively. This empirical evidence underscores the crucial role of EMA in SSRL for enhancing speaker verification performance.

Furthermore, the pseudo label queue further improves the SSRL model. Its integration not only amplifies speaker verification capabilities but also buffers against potential pitfalls associated with pseudo labeling. From Table VI, we can see that the experiment without using the label queue results in worse EERs and pseudo-labeling performance compared to the one that includes it. Specifically, the final cluster count (4,801) is significantly smaller, indicating that many classes were removed during training. This suggests that training with noisy labels leads to bias toward certain classes, and without correction, the model reinforces this bias, causing pseudo labels to collapse into fewer clusters. The label queue serves as a buffer against these potential pitfalls by stabilizing pseudo labels and preventing the excessive merging of speaker identities.

Notably, introduction of noisy label modeling with  $p_{\text{clean}}$  provides an additional layer of refinement to the SSRL approach. By guiding predictions towards cleaner samples, this mechanism mitigates the challenges associated with noisy label updates. A degradation in speaker verification performance is observed in the absence of noisy label modeling, with EER increase to 2.76%, 2.94%, and 5.29% across the test trials.

Additionally, we evaluated a simplified version of SSRL. This variant, devoid of the EMA updates, pseudo label queue, and noisy label modeling, preserves only the noisy student training strategy. As observed in Table VI, this simplified version undergoes a significant performance drop, with an EER increase of 2.8 percentage points  $(2.39\% \rightarrow 5.19\%)$ 

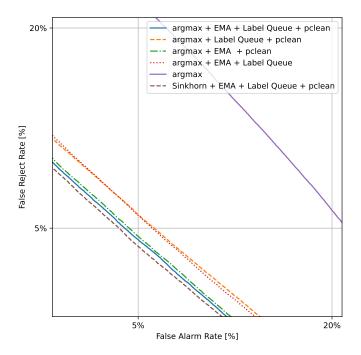


Fig. 7: DET curves on VoxCeleb 1-H test trial: comparing SSRL methods with different component configurations.

compared to the full SSRL approach on the VoxCeleb 1-O test trial. In fact, this simplified SSRL had difficulty converging. These observations underscore the collective significance of the various components in achieving optimal performance with SSRL. The detection error tradeoff (DET) plots on VoxCeleb 1-H test trial is shown in Figure 7 to compare the SSRL approach with different component configurations.

2) Pseudo labeling analysis: The dynamism inherent in the online clustering mechanism deserves mention. Through SSRL's online clustering, certain clusters are filtered out or merged as training progresses, thereby stabilizing the number of clusters towards the conclusion. Referring to Table VI, the full version of SSRL, equipped with the EMA update, pseudo label queue, and noisy label modeling, has a converged cluster count of 5085. However, models without either the EMA update or the pseudo label queue end with smaller cluster counts, registering at 4943 and 4801, respectively. This observation indicates that the EMA update and pseudo label queue jointly act as regularizers for pseudo cluster prediction, fostering stability throughout the training epochs, thus preventing the cluster counts shrink too quickly. Specifically, the pseudo label

TABLE VII: Performance comparison of the SSRL approach using varying numbers of clusters for the initial pseudo labels.  $K_{\text{init}}$  denotes the initial number of clusters;  $K_{\text{converged}}$  indicates the number of clusters upon convergence. The arrow illustrates the transition from the model trained with the fixed initial clustering for 50 epochs to the converged SSRL.

K <sub>init</sub>	Verifi	Verification EER[%] ↓			Pseudo Labeling					
mit	Vox1-O	Vox1-E	Vox1-H	NMI ↑	Acc ↑	Purity ↑	$K_{\text{converged}}$			
1,000 <sup>3</sup>	< 5.98 6.73		ς 11.87 13.93	Ç 0.6189 0.6658	ς 19.84% 21.84%	ς 21.58% 45.58%	714			
8,000	< 4.05 2.39	\$\bigs\{ \frac{4.61}{2.63}}\$	< 8.58 <b>4.74</b>	Ç 0.7744 0.9333	\$\begin{cases} 36.87\\ <b>78.12\%</b> \end{cases}\$	ς 55.32% 87.42%	5,085			
20,000					ς 27.68% 73.35%		9,956			

TABLE VIII: Performance comparison of the SSRL approach with different pseudo label queue length  $L.\ K$  represents the converged number of clusters.

$\frac{}{L}$	Verific	ation EEF	R[%] ↓		Pseudo L	abeling	
	Vox1-O	Vox1-E	Vox1-H	NMI ↑	Acc ↑	Purity ↑	K
14	2.51	2.68	4.82	0.9297	77.31%	86.96%	4801
5	2.39	2.63	4.74	0.9333	78.12%	87.42%	5085
10	2.45	2.65	4.79	0.9322	77.58%	87.31%	5320
20	2.56	2.73	4.93	0.9296	76.87%	86.97%	5485

queue serves as a buffer against erratic predictions, enhancing stability and minimizing outliers, while the EMA component ensures that the network remains consistent in its cluster assignment predictions.

Conversely, the noisy label modeling approach appears to have an opposing effect on the converged cluster count compared to the EMA update and pseudo label queue. Excluding noisy label modeling culminates in an increased cluster count of 6152. This suggests that noisy label modeling prioritizes predictions for more confident clusters, diminishing those with less confidence, which consequently reduces the overall cluster count.

3) Direct maximum probability versus Sinkhorn-based online clustering: To investigate the impact of different online clustering techniques, we evaluated two approaches: direct maximum probability assignment and cluster assignment through optimal transport.

In terms of speaker verification, both methods prove efficacious, achieving comparable EERs across test trials. On VoxCeleb 1-E and VoxCeleb 1-H, the Sinkhorn approach registers marginal improvements with EERs of 2.57% and 4.61% compared to 2.63% and 4.74% using direct assignment. This suggests that the two techniques are largely comparable in enhancing speaker verification capabilities.

Regarding pseudo labeling, the Sinkhorn method manifests an edge, garnering superior metrics of clustering accuracy (79.26% vs 78.12%) and NMI (0.9402 vs 0.9333). This indicates an enhanced capacity for accurate pseudo label generation using the optimal transport approach. Inspecting the converged number of clusters, Sinkhorn retains more clusters

upon convergence at 5974, contrasted with 5085 using direct assignment. This aligns with the constraint in the Sinkhorn algorithm to distribute samples evenly across clusters. Conversely, the direct assignment aggressively merges smaller, outlier clusters. In summary, both online clustering techniques prove effective and validate the online clustering mechanism's efficacy in SSRL.

- 4) Interplay of initial cluster count: Table VII shows the interplay between the initial cluster count  $K_{\rm init}$  and the SSRL approach's performance. An overly conservative choice for  $K_{\rm init}$  (e.g., 1,000) seems to restrict the model's ability to capture the data's inherent diversity, leading to suboptimal results. In contrast, an overly aggressive  $K_{\rm init}$  (e.g., 20,000) does allow for improved pseudo labeling metrics, but doesn't necessarily translate to the best verification EER. In summary, the choice of  $K_{\rm init}$  is crucial. It acts as a balance between providing enough granularity for capturing data diversity and ensuring the model remains focused on meaningful clusters.
- 5) Impact of pseudo label queue length L: Table VIII shows the impact of pseudo label queue length L on the model's performance. The pseudo label queue filters transient inconsistencies, and ensuring continuity in predicted pseudo labels across training epochs. An observation is the marginal degradation in performance as L increases beyond a certain threshold. With L=1, essentially indicating no pseudo label queue, the verification EER on VoxCeleb 1-O test trial is 2.51% and the converged number of clusters K stands at 4801. Increasing L to 5 yields a better EER of 2.39% and a higher K of 5084. Further increments in L to 10 and 20, however, show worse EERs and expanding Ks. This trend suggests an optimal range for L where the benefits of temporal stabilization maximize. An excessively long queue might integrate older, potentially less relevant pseudo labels, causing slight deteriorations in performance. This observation aligns with the inherent trade-off: while having some history aids in stabilization, overly long histories might dilute the recent advancements the model has achieved.

## D. Fine-tuning

In this section, the SSRL pre-trained ECAPA-TDNN speaker model is fine-tuned with small-scale labeled datasets. We use the VoxCeleb 1 development set (1,211 speakers) [71] for fine-tuning and create an additional subset of 600 randomly selected speakers to evaluate self-supervised pre-training on

<sup>&</sup>lt;sup>3</sup>When trained with an initial cluster count of 1,000, the model could not converge, so we stopped the training after 50 epochs of SSRL.

 $<sup>^4</sup>$ Label queue method is disabled when label queue length L=1.

TABLE IX: Fine-tune the self-supervised model with different labeled data in VoxCeleb 1 development set.

Fine-tuning Data		None		600 S <sub>J</sub>	peakers	1,211 Speakers		
Pre-trained	Model n	ninDCF	EER[%]	minDCF	EER[%]	minDCF	EER[%]	
None SSRL		- 0.101	- 1 25	0.295	3.94 1.05	0.175 0.075	2.31	

smaller datasets. Results are reported on the VoxCeleb 1-O test trials.

As shown in Table IX, fine-tuning the SSRL model with labeled data significantly improves performance: fine-tuning on only 600 speakers achieves an EER of 1.05%, compared to 3.94% without SSRL pre-training. Fine-tuning the SSRL model on all labeled speakers in VoxCeleb 1 (1,211 speakers) further reduces the EER to 0.95%, compared to 2.31% without SSRL pre-training. These results demonstrate that SSRL provides a strong self-supervised foundation, which can be further enhanced with labeled data for improved speaker verification.

#### VI. CONCLUSION

This paper introduces self-supervised reflective learning (SSRL), a novel paradigm for unsupervised speaker representation learning. SSRL streamlines existing two-stage iterative frameworks by integrating self-supervised knowledge distillation with online clustering. A teacher model continually refines pseudo labels through clustering, providing dynamic supervision to train the student model. The method also employs techniques like label correction and noisy label modeling to further improve pseudo label quality.

Our experiments demonstrate SSRL's superiority over current two-stage iterative approaches. On VoxCeleb 1 test trials, SSRL surpasses the performance of a 5-round iterative method in just a single training round. Ablation studies validate the contributions of key components like noisy label modeling, pseudo label queues, and EMA teacher updates. Moreover, the consistent improvement in pseudo labeling throughout the training phase, coupled with the convergence of cluster count, reaffirms SSRL's prowess in deciphering pertinent clusters within unlabeled data.

This work marks a pivotal advancement in efficient and accurate speaker representation learning. By combining self-supervised distillation and online clustering, SSRL eliminates previous iterative bottlenecks. The reflective learning paradigm introduces new horizons for developing scalable, unsupervised systems. Future work should assess SSRL on larger datasets and expand hyperparameter optimizations. Integrating SSRL into end-to-end pipelines is another research direction.

## VII. ACKNOWLEDGMENTS

This research is funded in part by the National Natural Science Foundation of China (62171207), Science and Technology Program of Suzhou City (SYC2022051) and Guangdong Science and Technology Plan (2023A1111120012). Many thanks for the computational resource provided by the Advanced Computing East China Sub-Center.

#### REFERENCES

- Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "Large-Scale Self-Supervised Speech Representation Learning for Automatic Speaker Verification," in *Proceeding of ICASSP*, 2022, pp. 6147–6151.
- [2] Y. Chen, S. Zheng, H. Wang, L. Cheng, and Q. Chen, "Pushing the Limits of Self-Supervised Speaker Verification Using Regularized Distillation Framework," in *Proceeding of ICASSP*, 2023, pp. 1–5.
- [3] Y. Tu, M.-W. Mak, and J.-T. Chien, "Contrastive Self-Supervised Speaker Embedding with Sequential Disentanglement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [4] Y. Liu, L.-F. Wei, C.-F. Zhang, T.-H. Zhang, S.-L. Chen, and X.-C. Yin, "Self-Supervised Contrastive Speaker Verification with Nearest Neighbor Positive Instances," *Pattern Recognition Letters*, vol. 173, pp. 17–22, 2023.
- [5] Z. Zhou, H. Yang, and T. Shinozaki, "Self-Supervised Speaker Verification with Adaptive Threshold and Hierarchical Training," in *Proceeding* of ICASSP, 2024, pp. 12141–12145.
- [6] A. Fathan and J. Alam, "An Analytic Study on Clustering Driven Self-Supervised Speaker Verification," *Pattern Recognition Letters*, vol. 179, pp. 80–86, 2024.
- [7] S. Wang, Q. Bai, Q. Liu, J. Yu, Z. Chen, B. Han, Y. Qian, and H. Li, "Leveraging in-the-wild data for effective self-supervised pretraining in speaker recognition," in *Proceeding of ICASSP*, 2024, pp. 10901– 10905
- [8] D. Cai, W. Wang, and M. Li, "An Iterative Framework for Self-Supervised Deep Speaker Representation Learning," in *Proceeding of ICASSP*, 2021, pp. 6728–6732.
- [9] D. Cai, W. Wang, and M. Li, "Incorporating Visual Information in Audio Based Self-Supervised Speaker Recognition," *IEEE/ACM Transactions* on Audio, Speech, and Language Processing, vol. 30, pp. 1422–1435, 2022.
- [10] X. Chen and K. He, "Exploring Simple Siamese Representation Learning," in *Proceedings of CVPR*, 2021, pp. 15750–15758.
- [11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in NeurIPS Deep Learning and Representation Learning Workshop, 2015.
- [12] 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," NeurIPS, vol. 33, pp. 21271–21284, 2020.
- [13] 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 ICCV*, 2021, pp. 9650–9660.
- [14] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Unsupervised Label Noise Modeling and Loss Correction," in *Proceedings of the International Conference on Machine Learning*, 2019.
- [15] G. Elbanna, N. Scheidwasser-Clow, M. Kegler, P. Beckmann, K. El Hajal, and M. Cernak, "byol-S: Learning Self-supervised Speech Representations by Bootstrapping," in *HEAR: Holistic Evaluation of Audio Representations*. PMLR, 2022, pp. 25–47.
- [16] A. H. Liu, H.-J. Chang, M. Auli, W.-N. Hsu, and J. Glass, "DinoSR: Self-distillation and Online Clustering for Self-supervised Speech Representation learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [17] Q.-S. Zhu, L. Zhou, J. Zhang, S.-J. Liu, Y.-C. Hu, and L.-R. Dai, "Robust Data2VEC: Noise-Robust Speech Representation Learning for ASR by Combining Regression and Improved Contrastive Learning," in Proceeding of ICASSP, 2023, pp. 1–5.
- [18] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features," in *Proceedings of ECCV*, 2018.
- [19] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-Labelling Via Simultaneous Clustering and Representation Learning," in ICLR, 2020.
- [20] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical Contrastive Learning of Unsupervised Representations," in *ICLR*, 2021.
- [21] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online Deep Clustering for Unsupervised Representation Learning," in *Proceedings* of CVPR, 2020, pp. 6687–6696.
- [22] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," *NeurIPS*, vol. 33, pp. 9912–9924, 2020.
- [23] H.-J. Chang, A. H. Liu, and J. Glass, "Self-supervised Fine-tuning for Improved Content Representations by Speaker-invariant Clustering," in *Proceeding of Interspeech*, 2023, pp. 2983–2987.

- [24] X. Zhu and A. B. Goldberg, Introduction to Semi-Supervised Learning. Springer Nature, 2022.
- [25] X. Yang, Z. Song, I. King, and Z. Xu, "A Survey on Deep Semi-Supervised Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934–8954, 2023.
- [26] M.-R. Amini, V. Feofanov, L. Pauletto, E. Devijver, and Y. Maximov, "Self-training: A survey," arXiv:2202.12040, 2022.
- [27] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the Limits of the Teacher in Semi-supervised Learning," in *Proceedings* of CVPR, 2019, pp. 6728–6736.
- [28] 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," *NeurIPS*, vol. 33, pp. 596–608, 2020.
- [29] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised Semantic Segmentation with Cross Pseudo Supervision," in *Proceedings of CVPR*, 2021, pp. 2613–2622.
- [30] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with Noisy Student Improves ImageNet Classification," in *Proceedings of CVPR*, 2020, pp. 10687–10698.
- [31] P. P. Busto, A. Iqbal, and J. Gall, "Open Set Domain Adaptation for Image and Action Recognition," *IEEE Transactions on Pattern Analysis* and Machine Intelligence, vol. 42, no. 2, pp. 413–429, 2018.
- [32] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for Visual Domain Adaptation," in *ICLR*, 2018.
- [33] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence Regularized Self-training," in *Proceedings of CVPR*, 2019, pp. 5982–5991.
- [34] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised Domain Adaptation for Semantic Segmentation via Class-balanced Self-training," in *Proceedings of ECCV*, 2018, pp. 289–305.
- [35] A. Tarvainen and H. Valpola, "Mean Teachers are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-supervised Deep Learning Results," *NeurIPS*, vol. 30, 2017.
- [36] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," in *ICLR*, 2016.
- [37] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep Cotraining for Semi-supervised Image Recognition," in *Proceedings of ECCV*, 2018, pp. 135–152.
- [38] W. Dong-DongChen and Z. WeiGao, "Tri-net for Semi-supervised Deep Learning," in *Proceeding of IJCAI*, 2018, pp. 2014–2020.
- [39] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification using Adapted Gaussian Mixture Models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.
- [40] J.-L. Gauvain and C.-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.
- [41] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis versus Eigenchannels in Speaker Recognition," *IEEE Transac*tions on Audio, Speech, and Language Processing, vol. 15, no. 4, pp. 1435–1447, 2007.
- [42] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [43] D. Snyder, D. Garcia-Romero, and D. Povey, "Time Delay Deep Neural Network-based Universal Background Models for Speaker Recognition," in *Proceeding of IEEE Automatic Speech Recognition and Understanding Workshop*, 2015, pp. 92–97.
- [44] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in Proceeding of ICASSP, 2018, pp. 5329–5333.
- [45] W. Cai, J. Chen, and M. Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Proceeding of The Speaker and Language Recognition Workshop* (Odyssey), 2018.
- [46] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net Structures for Speaker Verification," in *Proceeding of IEEE Spoken Language Technology Workshop*, 2021, pp. 301–307.
- [47] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proceeding of Interspeech*, 2020, pp. 3830–3834.
- [48] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "MFA-Conformer: Multi-scale Feature Aggregation Conformer for Automatic Speaker Verification," in *Proceeding of Interspeech*, 2022, pp. 306–310.

- [49] D. Liao, T. Jiang, F. Wang, L. Li, and Q. Hong, "Towards A Unified Conformer Structure: from ASR to ASV Task," in *Proceeding of ICASSP*, 2023, pp. 1–5.
- [50] D. Cai and M. Li, "Leveraging ASR Pretrained Conformers for Speaker Verification Through Transfer Learning and Knowledge Distillation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3532–3545, 2024.
- [51] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," Advances in Neural iInformation Processing Systems, vol. 33, pp. 12449–12460, 2020.
- [52] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How Much Can a Bad Teacher Benefit ASR Pre-Training?" in Proceeding of ICASSP, 2021, pp. 6533–6537.
- [53] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "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.
- [54] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on Speaker Verification and Language Identification," in *Proceeding of Interspeech*, 2021, pp. 1509–1513.
- [55] N. Vaessen and D. A. van Leeuwen, "Fine-Tuning wav2vec2 for Speaker Recognition," in *Proceeding of ICASSP*, 2022, pp. 7967–7971.
- [56] J. Li, K. Zheng, J. Yao, L. Gao, and D. Hong, "Deep Unsupervised Blind Hyperspectral and Multispectral Data Fusion," *IEEE Geoscience* and Remote Sensing Letters, vol. 19, pp. 1–5, 2022.
- [57] J. Li, K. Zheng, W. Liu, Z. Li, H. Yu, and L. Ni, "Model-Guided Coarse-to-Fine Fusion Network for Unsupervised Hyperspectral Image Super-Resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.
- [58] J. Li, K. Zheng, L. Gao, L. Ni, M. Huang, and J. Chanussot, "Model-Informed Multistage Unsupervised Network for Hyperspectral Image Super-Resolution," *IEEE Transactions on Geoscience and Remote Sens*ing, vol. 62, pp. 1–17, 2024.
- [59] J. Thienpondt, B. Desplanques, and K. Demuynck, "The IDLAB Vox-Celeb Speaker Recognition Challenge 2020 System Description," in VoxSRC workshop, 2020.
- [60] B. Han, Z. Chen, and Y. Qian, "Self-Supervised Learning With Cluster-Aware-DINO for High-Performance Robust Speaker Verification," *IEEE/ACM Transactions on Audio, Speech, and Language Pro*cessing, vol. 32, pp. 529–541, 2024.
- [61] Z. Zhao, Z. Li, X. Zhang, W. Wang, and P. Zhang, "Prototype Division for Self-Supervised Speaker Verification," *IEEE Signal Processing Letters*, vol. 31, pp. 880–884, 2024.
- [62] R. Tao, K. A. Lee, R. K. Das, V. Hautamäki, and H. Li, "Self-Supervised Training of Speaker Encoder with Multi-Modal Diverse Positive Pairs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1706–1719, 2023.
- [63] R. Tao, K. A. Lee, R. K. Das, V. Hautamäki, and H. Li, "Self-Supervised Speaker Recognition with Loss-Gated Learning," in *Proceeding of ICASSP*, 2022, pp. 6142–6146.
- [64] B. Han, Z. Chen, and Y. Qian, "Self-Supervised Speaker Verification Using Dynamic Loss-Gate and Label Correction," in *Proceeding of Interspeech*, 2022, pp. 4780–4784.
- [65] H. Chen, H. Zhang, L. Wang, K. A. Lee, M. Liu, and J. Dang, "Self-Supervised Audio-Visual Speaker Representation with Co-Meta Learning," in *Proceeding of ICASSP*, 2023, pp. 1–5.
- [66] Z. Fang, L. He, L. Li, and Y. Hu, "Improving Speaker Verification with Noise-Aware Label Ensembling and Sample Selection: Learning and Correcting Noisy Speaker Labels," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2988–3001, 2024.
- [67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [68] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of CVPR*, 2019, pp. 4685–4694.
- [69] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proceedings of CVPR*, 2020, pp. 9729–9738.
- [70] M. Cuturi, "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," NeurIPS, vol. 26, 2013.
- [71] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A Large-Scale Speaker Identification Dataset," in *Proceeding of Interspeech*, 2017, pp. 2616–2620.

- [72] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," in *Proceeding of Interspeech*, 2018, pp. 1086–1090.
- [73] D. Cai, W. Cai, and M. Li, "Within-Sample Variability-Invariant Loss for Robust Speaker Recognition Under Noisy Environments," in *Proceeding* of ICASSP, 2020, pp. 6469–6473.
- [74] N. Inoue and K. Goto, "Semi-Supervised Contrastive Learning with Generalized Contrastive Loss and its Application to Speaker Recognition," in *Proceeding of APSIPA ASC*, 2020, pp. 1641–1646.
- [75] J. Kang, J. Huh, H. S. Heo, and J. S. 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.
- [76] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the ICML*, 2020, pp. 1597–1607.
- [77] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484*, 2015.
- [78] 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 *Proceeding of ICASSP*, 2017, pp. 5220–5224.
- [79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.
- [80] "NIST 2016 Speaker Recognition Evaluation Plan," 2016. [Online]. Available: https://www.nist.gov/system/files/documents/2016/ 10/07/sre16\_eval\_plan\_v1.3.pdf
- [81] Y. M. Asano, M. Patrick, C. Rupprecht, and A. Vedaldi, "Labelling Unlabelled Videos from Scratch with Multi-Modal Self-Supervision," *NeurIPS*, vol. 33, pp. 4660–4671, 2020.
- [82] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [83] S. H. Mun, M. H. Han, and N. S. Kim, "SNU-HIL System for the VoxCeleb Speaker Recognition Challenge 2021," in *VoxSRC workshop*, 2021.
- [84] Z. Chen, J. Wang, W. Hu, L. Li, and Q. Hong, "Unsupervised Speaker Verification Using Pre-Trained Model and Label Correction," in *Pro*ceeding of ICASSP, 2023, pp. 1–5.