



SSPS: Self-Supervised Positive Sampling for Robust Self-Supervised Speaker Verification

Theo Lepage, Reda Dehak

EPITA Research Laboratory (LRE), France

theo.lepage@epita.fr, reda.dehak@epita.fr

Abstract

Self-Supervised Learning (SSL) has led to considerable progress in Speaker Verification (SV). The standard framework uses same-utterance positive sampling and data-augmentation to generate anchor-positive pairs of the same speaker. This is a major limitation, as this strategy primarily encodes channel information from the recording condition, shared by the anchor and positive. We propose a new positive sampling technique to address this bottleneck: Self-Supervised Positive Sampling (SSPS). For a given anchor, SSPS aims to find an appropriate positive, i.e., of the same speaker identity but a different recording condition, in the latent space using clustering assignments and a memory queue of positive embeddings. SSPS improves SV performance for both SimCLR and DINO, reaching 2.57% and 2.53% EER, outperforming SOTA SSL methods on VoxCeleb1-O. In particular, SimCLR-SSPS achieves a 58% EER reduction by lowering intra-speaker variance, providing comparable performance to DINO-SSPS.

Index Terms: Self-Supervised Learning, Speaker Verification, Speaker Representations

1. Introduction

The main application of speaker recognition is Speaker Verification (SV), which determines whether a given speech sample matches a claimed identity. SV systems aim to define a representation space that minimizes inter-speaker similarities and maximizes intra-speaker similarities while ensuring robustness against extrinsic variabilities (e.g., environmental noise, channel, and mismatching recording devices). Deep learning has significantly advanced the field, surpassing traditional approaches such as i-vectors [1] with models such as x-vectors [2], ResNet [3], and ECAPA-TDNN [4] architectures. These methods learn to associate speech samples with their speaker identities in a supervised fashion on large-scale labeled datasets [5]. The effectiveness of deep learning models improves with larger training datasets. However, this reliance on extensive labeled data poses a significant challenge, as high-quality annotated speech samples are scarce and expensive.

Self-Supervised Learning (SSL) has emerged as a promising approach to overcome this limitation by deriving informative representations directly from the input data. Taking advantage of the vast availability of unlabeled speech, SSL enhances model scalability and reduces the reliance on annotated datasets. Several SSL frameworks have been developed around the joint-embedding architecture, where an anchor and a positive are derived from different views of the same input

data, preserving the underlying high-level information. Methods based on contrastive learning, such as SimCLR [6] and MoCo [7], seek to maximize the similarity within positive pairs while minimizing the similarity within negative pairs, which are sampled from the batch or a larger memory queue. Self-distillation, such as DINO [8], leverages knowledge distillation and the student-teacher paradigm, where a student model is trained to match the teacher model's output. For SV, methods predominantly rely on contrastive learning [9, 10, 11], and self-distillation [12, 13, 14, 15]. These approaches rely on the assumption that the anchor and the positive are from the same speaker identity since both frames are derived from the same utterance. Thus, data-augmentation is fundamental to avoid encoding channel information, shared between the two segments.

However, data-augmentation alone cannot mitigate the impact of SSL same-utterance positive sampling, as it introduces channel characteristics coming from the recording condition into speaker representations, increasing intra-speaker variance. To address this, several methods have been proposed: AP+AAT [9] employs an adversarial loss to discourage the model from encoding channel information; i-mix [16] applies data-driven augmentation by interpolating training utterances to emphasize key distinguishing features; DPP [17] finds diverse positives by relying on speech and face data; CA-DINO [18] performs clustering to select positives from the anchor class. Additionally, alternative SSL positive sampling strategies have been explored in computer vision, notably NNCLR [19] and GPS-SSL [20], which identify positives in the latent space using nearest neighbor search.

This paper presents a novel positive sampling strategy for SSL, termed **Self-Supervised Positive Sampling (SSPS)**. Rather than selecting a positive sample from the same utterance as the anchor, SSPS identifies a pseudo-positive from a distinct utterance by leveraging the knowledge progressively acquired through SSL. After several epochs of conventional SSL training, it is assumed that samples from the same speaker, recorded under different conditions, will have representations close to the anchor representation. This approach enables learning more robust speaker representations by matching various recording conditions with the same speaker identity. SSPS effectively enhances SV performance across major SSL frameworks by reducing intra-speaker variance. For a detailed analysis and additional results, see the extended version [21].

SSPS is presented in Section 2, following a brief description of the SimCLR and DINO SSL frameworks. The experimental setup is described in Section 3. The impact of SSL positive sampling is first highlighted, followed by a study of SSPS hyperparameters, a comparison of SSPS performance on SV with SOTA methods, and a visualization of speaker representations in Section 4. Finally, the article concludes in Section 5.

Code: <https://github.com/theolepage/sslsv>

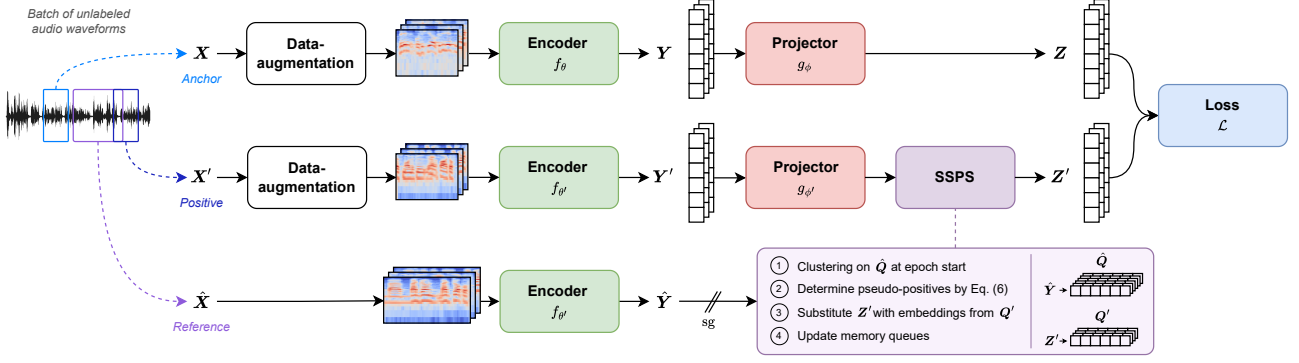


Figure 1: SSL training framework for SV with Self-Supervised Positive Sampling (SSPS).

2. Methods

2.1. SSL frameworks

The self-supervised training framework uses a joint-embedding architecture to create a pair of embeddings from an unlabeled audio sample. Consider a training set of size N , with indices denoted by $\mathcal{I} \equiv \{1, \dots, N\}$. At each iteration, B utterances are selected with $\mathcal{B} \subseteq \mathcal{I}$ the batch indices. From a given utterance $u_i \in \{u_i\}_{i \in \mathcal{B}}$, two segments, \mathbf{x}_i (anchor) and \mathbf{x}'_i (positive), are randomly extracted. Then, random data-augmentation is applied and their mel-scaled spectrograms are used as input features. The architecture revolves around two branches: (1) an encoder f_θ followed by a projector g_ϕ ; (2) an encoder $f_{\theta'}$ followed by a projector $g_{\phi'}$. Encoders f_θ and $f_{\theta'}$ map \mathbf{x}_i and \mathbf{x}'_i to representations \mathbf{y}_i and \mathbf{y}'_i with dimension D_{repr} . Projectors g_ϕ and $g_{\phi'}$ transform \mathbf{y}_i and \mathbf{y}'_i to embeddings \mathbf{z}_i and \mathbf{z}'_i with dimension D_{emb} . Representations are employed for speaker verification, while embeddings are used to compute the loss \mathcal{L} . Batches are denoted as $\mathbf{X} = \{\mathbf{x}_i\}_{i \in \mathcal{B}}$, $\mathbf{Y} = \{\mathbf{y}_i\}_{i \in \mathcal{B}}$, and $\mathbf{Z} = \{\mathbf{z}_i\}_{i \in \mathcal{B}}$, with their corresponding counterparts of the other branch denoted as \mathbf{X}' , \mathbf{Y}' , and \mathbf{Z}' , respectively. By default, SSL frameworks rely on the *symmetrical* joint-embedding architecture where the weights are identical across branches (e.g., SimCLR). When employing the *asymmetrical* version, one branch is designated as the *student* and the other as the *teacher* (e.g., DINO). In this case, the gradient is not computed for the teacher since its weights are updated using an Exponential Moving Average (EMA) of student weights with $m \in [0, 1)$ the momentum coefficient.

2.1.1. SimCLR

SimCLR [6] is based on contrastive learning as it aims at maximizing the similarity within anchor-positive pairs while maximizing the distance between anchor-negative pairs. Positives are derived from the same utterances as their anchor (same speaker throughout an utterance), and negatives are sampled from the current batch by assuming that negatives will belong to a different speaker identity. $\mathcal{L}_{\text{SimCLR}}$ is defined as:

$$\mathcal{L}_{\text{SimCLR}} = -\frac{1}{B} \sum_{i \in \mathcal{B}} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}'_i) / \tau)}{\sum_{j \in \mathcal{B}} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}'_j) / \tau)}, \quad (1)$$

where $\text{sim}(\mathbf{a}, \mathbf{b})$ represents the cosine similarity between \mathbf{a} and \mathbf{b} , and τ is a temperature hyperparameter.

2.1.2. DINO

DINO [8] adopts a self-distillation training framework in which a *student* is trained to predict the outputs of a *teacher*. The teacher's weights are updated using an EMA of the student's weights. A larger set of augmented utterances of different lengths is considered, resulting in four short (*local*) and two long (*global*) segments. All inputs are processed by the student, but only global views are processed by the teacher. The student and teacher projectors output an embedding normalized with a temperature-softmax. *Centering* and *sharpening* are applied to the teacher embeddings to prevent one dimension from prevailing while discouraging collapse to the uniform distribution. $\mathcal{L}_{\text{DINO}}$ is defined as:

$$\mathcal{L}_{\text{DINO}} = \frac{1}{B} \sum_{i \in \mathcal{B}} \sum_{t=1}^2 \sum_{\substack{s=1 \\ s \neq t}}^{2+4} H\left(\frac{\mathbf{z}'_{i,t} - \mathbf{c}}{\tau_t}, \frac{\mathbf{z}_{i,s}}{\tau_s}\right), \quad (2)$$

where $H(\mathbf{a}, \mathbf{b}) = -\text{softmax}(\mathbf{a}) \log(\text{softmax}(\mathbf{b}))$, $\mathbf{z}'_{i,t}$ and $\mathbf{z}_{i,s}$ are the t -th teacher and s -th student embeddings of sample i , τ_t is the temperature for the teacher, τ_s is the temperature for the student, and \mathbf{c} is a running mean on the teacher outputs.

2.2. Self-Supervised Positive Sampling (SSPS)

The performance of SSL frameworks primarily depends on how anchor-positive pairs are defined, as it helps model the distribution of each class [22]. SSL commonly relies on data-augmentation to generate a *positive* for a given *anchor*. However, standard data-augmentation techniques may not be sufficient to represent the diversity of acoustic conditions among samples from the same speaker. Thus, SSL models trained for SV are prone to encoding channel-related information because anchor-positive pairs are derived from the same utterances. Self-Supervised Positive Sampling (SSPS) is proposed to sample positives from *different recording conditions* of the same speaker. It is assumed that SSL same-utterance positive sampling group utterances of the same recordings, with similar channel characteristics, before modeling speaker identities. This implies that the latent space is organized by groups of utterances of the same recording and then by subgroups of utterances of the same speaker. Given a training utterance u_i ($i \in \mathcal{B}$), from which the anchor is sampled, let $\text{pos}(i)$ denote the index of an utterance $u_{\text{pos}(i)}$ used to sample the positive. While standard SSL approaches create the positive from the same utterance (i.e., $\text{pos}(i) = i$), SSPS determines a *pseudo-positive* from a different utterance (i.e., $\text{pos}(i) \neq i$) in the latent space based on

clustering assignments, as detailed in the following. The SSPS training framework is depicted in Figure 1.

2.2.1. Framework

To capture the unaltered audio patterns, SSPS introduces a *reference* segment \hat{x}_i sampled from u_i using a longer audio segment and no data-augmentation. Additionally, two memory queues are implemented in the framework: \hat{Q} with size $(|\hat{Q}|, D_{\text{repr}})$ for storing reference representations $\{\hat{y}_i\}_{i \in \mathcal{I}}$, and Q' with size $(|Q'|, D_{\text{emb}})$ for storing positive embeddings $\{z'_i\}_{i \in \mathcal{I}}$. After a pre-defined number of standard SSL training epochs, SSPS is enabled and pseudo-positive embeddings are sampled from Q' such that z'_i is replaced by $q'_{\text{pos}(i)}$ in the previously defined SSL objective functions.

2.2.2. Pseudo-positives sampling

At the beginning of each SSPS epoch, k-means clustering is performed on reference representations in \hat{Q} to group utterances into K clusters, allowing the assignments to be progressively refined as SSL representations improve. c_i denotes the cluster index of the i -th utterance, and m_k represents the centroid for the k -th cluster. The proposed method considers the following techniques to determine the cluster \hat{c}_i from which to sample a pseudo-positive.

- **Same-cluster sampling.** Utterances from the anchor cluster can be considered as pseudo-positives if K tends to the number of speaker identities in the train set, similarly to CA-DINO [18], such that:

$$\hat{c}_i = c_i. \quad (3)$$

- **Neighboring-clusters sampling.** According to the assumption that channel-related information is modeled before speaker-related information, utterances from neighboring clusters can also be considered as pseudo-positives when selecting a larger value for K , such that:

$$\hat{c}_i = \text{sample}(\mathcal{C}_{c_i}), \quad (4)$$

where $\text{sample}(S)$ is a uniform random selection from S , and \mathcal{C}_k consists of the M nearest clusters to the k -th cluster:

$$\mathcal{C}_k \triangleq \text{top } M \left(\{\text{sim}(\mathbf{m}_k, \mathbf{m}_j), \forall j \in [1, K]\} \right), \quad (5)$$

where $\text{top } M(S)$ returns the indices of the largest M values from S in descending order.

SSPS selects a pseudo-positive for the i -th sample according to \hat{c}_i , such that:

$$\text{pos}(i) = \text{sample}(\mathcal{S}_{\hat{c}_i}), \quad (6)$$

where $\mathcal{S}_c \triangleq \{i \in \mathcal{I} \text{ s.t. } c_i = c\}$ corresponds to the training sample indices from a given cluster. Note that it falls back to default SSL positive sampling if $q'_{\text{pos}(i)}$ is not present in Q' .

3. Experimental setup

3.1. Datasets and feature extraction

Models are trained on VoxCeleb2 [5] *dev* set, consisting of 1,092,009 utterances distributed among 145,569 recordings from 5,994 speakers. Speaker labels are discarded for the SSL training. The evaluation is conducted on VoxCeleb1 [23] *original* test set. Input features are 40-dimensional log-mel spectrograms extracted with the torchaudio library, using a Hamming window length of 25 ms and a frame-shift of 10 ms. Data-augmentation, including reverberation and background noises,

is applied with the Simulated RIR Database [24] and the MUSAN corpus [25].

3.2. SSL frameworks

The encoder f is either based on Fast ResNet-34 [3] or ECAPA-TDNN (C=1024) [4] for preliminary and final results, respectively. Output representations have a dimension of $D_{\text{repr}} = 512$. The supervised baseline corresponds to a model trained using the AAM-Softmax loss with a scale of 30 and a margin of 0.2. The implementation is based on PyTorch, and the trainings are conducted on $2 \times$ and $4 \times$ NVIDIA Tesla A100 80 GB.

3.2.1. SimCLR

The duration of audio segments is 2 s. The projector g is discarded as it degrades the performance [26]. The loss temperature is $\tau = 0.03$. The model is trained during 100 epochs with Adam using a batch size set to 256, and a learning rate of 0.001 which is reduced by 5% every 5 epochs.

3.2.2. DINO

Local and global segments are four 2 s and two 4 s audio chunks, respectively. The projector g consists of an MLP composed of three linear layers and a final weight-normalized linear layer. The hidden dimensions are set to 2048, 2048, and 256. The last layer maps the l_2 -normalized embeddings to $D_{\text{emb}} = 65,536$ units. The student temperature is $\tau_s = 0.1$ and the teacher temperature is $\tau_t = 0.04$. For the EMA update, m goes from 0.996 to 1.0 with a cosine scheduler. The model is trained during 80 epochs with SGD, a weight decay of $5e^{-5}$, a batch size of 128, and a learning rate linearly increased to 0.2 during a 10-epochs warm-up before applying a cosine scheduler.

3.3. SSPS

Different positive samplings are compared: ‘SSL’, ‘SSPS’, and ‘Supervised’ (uses train set labels to define anchor-positive pairs). Models are obtained by resuming the SSL training with the corresponding positive sampling for 20 epochs. The duration of the reference frame is 4 s. Memory queue lengths are set to $|\hat{Q}| = N$ and $|Q'| = K$. K-means is performed using a PyTorch GPU implementation for 10 iterations.

3.4. Evaluation protocol

The scoring of each test trial is the cosine similarity of l_2 -normalized representation pairs, derived from the full-length utterances. The performance is reported in terms of EER and minDCF with $P_{\text{target}} = 0.01$, $C_{\text{miss}} = 1$, and $C_{\text{fa}} = 1$.

4. Results

4.1. Effect of SSL positive sampling on SV performance

SimCLR, DINO, and the supervised baseline achieve 6.30%, 3.07%, and 1.34% EER on VoxCeleb1-O, respectively, as shown in Table 1. The performance of SSL methods is significantly improved when using the train set labels to generate anchor-positive pairs from different recordings of the same speaker with distinct channel characteristics. This supervised positive sampling reduces the EER by $\sim 73\%$ for SimCLR and $\sim 23\%$ for DINO, highlighting the negative impact of SSL same-utterance positive sampling.

Table 1: SV performance with SSL and Supervised positive sampling using SimCLR and DINO frameworks (ECAPA-TDNN).

Method	Pos. sampling	EER (%)	minDCF _{0.01}
SimCLR	SSL	6.30	0.5286
	Supervised	1.72	0.2395
DINO	SSL	3.07	0.3616
	Supervised	2.36	0.2712
Supervised		1.34	0.1521

Table 2: Effect of SSPS hyper-parameters (K , M) on SV performance using SimCLR (Fast ResNet-34).

Pos. sampling	K	M	EER (%)	minDCF _{0.01}
SSL			9.41	0.6378
SSPS	6,000	0	6.63	0.5493
		10,000	0	6.82
	25,000	0	7.30	0.5805
		1	5.80	0.5250
Supervised	150,000	2	5.73	0.5258
		0	8.29	0.6170
	1	1	7.54	0.5923
		2	7.13	0.5711
Supervised			3.93	0.3900

4.2. Selection of SSPS hyperparameters

To select SSPS hyperparameters, the preliminary results using the Fast ResNet-34 encoder are reported with different values of K and M in Table 2. The systems are compared against the SSL same-utterance positive sampling baseline, achieving 9.41% EER on VoxCeleb1-O. As expected, sampling from the anchor cluster ($M = 0$) and using a number of classes close to the number of speaker identities in the train set ($K = 6,000$) reduces the EER to 6.63%. Sampling from a neighboring cluster ($M = 1$) and using $K = 25,000$ further reduces the EER to 5.80% and represents the best system for minimizing the minDCF. This value of K , smaller than the total number of recordings within the train set (i.e., $\sim 150,000$), suggests that some recordings are already grouped in the latent space. These results show that sampling positives from a cluster close to their anchor cluster generates appropriate and diverse anchor-positive pairs, which effectively improve SV performance.

4.3. Final evaluation and comparison to other methods

Table 3 presents the final performance of SimCLR and DINO, using ECAPA-TDNN, with and without SSPS (bottom), compared to other state-of-the-art SSL approaches for SV (top). SSPS ($K = 25,000$ and $M = 1$) improves the EER and minDCF of both frameworks on the VoxCeleb1-O benchmark. The best performance is obtained by DINO with SSPS, reaching 2.53% EER and 0.2843% minDCF. Additionally, SimCLR provides performance on par with DINO by achieving 2.57% EER and 0.3033 minDCF, a remarkable improvement over its baseline (58% relative EER reduction). This finding is very prospective as SimCLR relies on a simpler training framework and reaches the best SSL performance using the Supervised positive sampling strategy in Table 1, which implies that there is potential for further improvement of SSL contrastive-based

Table 3: Evaluation of SSL methods on SV (VoxCeleb1-O). The results for the top rows are drawn from the literature.

Method	EER (%)	minDCF _{0.01}
AP + AAT [9]	8.65	
Contrastive + VICReg [27]	8.47	0.6400
SimCLR + MSE loss [10]	8.28	0.6100
MoCo + ProtoNCE [11]	8.23	0.5900
CEL [28]	8.01	
SSReg [29]	6.99	
DINO + Cosine loss [30]	6.16	0.5240
DINO [12]	4.83	0.4630
DINO + Curriculum [13]	4.47	
CA-DINO [18]	3.59	0.3529
RDINO [15]	3.29	
MeMo [31]	3.10	
RDINO + W-GVKT [32]	2.89	0.3330
SimCLR	6.30	0.5286
SimCLR-SSPS	2.57	0.3033
DINO	3.07	0.3616
DINO-SSPS	2.53	0.2843

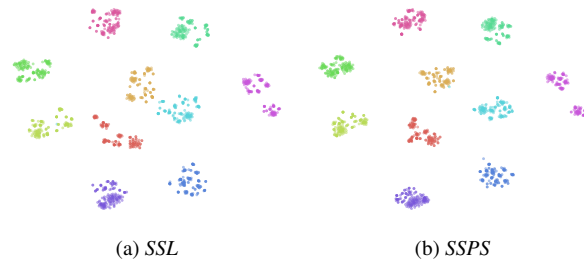


Figure 2: t -SNE of 10 speakers from VoxCeleb1 with SSL and SSPS positive sampling using SimCLR (ECAPA-TDNN).

methods. Therefore, SSPS improves the SV performance of major SSL frameworks, reducing the performance gap with the fully supervised baseline of 1.34% EER. Finally, the proposed SimCLR-SSPS and DINO-SSPS outperform other state-of-the-art SSL methods for SV by providing an explicit solution to their main limitation.

4.4. Visualization of speaker representations

To illustrate the improvement in class compactness, Figure 2 presents the t -SNE of 10 speaker representations from the test set with SSL and SSPS positive sampling techniques. SSPS allows for reducing intra-class variance by matching different recording conditions to the same speaker identity during the training. Note that same-speaker representations already far apart in the latent space are not considered to belong to the same speaker class when employing SSPS, as detailed in [21].

5. Conclusions

This work proposes a new method for sampling positives in SSL frameworks to address the limitations of the same-utterance positive sampling. SSPS samples positives that belong to the same or a neighboring cluster as their corresponding anchor in latent space, such that anchor-positive pairs originate from the same speaker identities but different recordings. This approach achieves SOTA performance with SimCLR and DINO on VoxCeleb1-O by reducing intra-speaker variance.

6. Acknowledgements

This work was performed using HPC resources from GENCI-IDRIS (Grant 2023-AD011014623) and has been partially funded by the French National Research Agency (project AP-ATE - ANR-22-CE39-0016-05).

7. References

- [1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2011.
- [2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [3] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.
- [4] B. Desplanques, J. Thienpondt, and K. Demuyck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020.
- [5] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018.
- [6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.
- [7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [8] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [9] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. S. Chung, "Augmentation adversarial training for unsupervised speaker recognition," in *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2020.
- [10] H. Zhang, Y. Zou, and H. Wang, "Contrastive self-supervised learning for text-independent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [11] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [12] J. Cho, J. Villalba, L. Moro-Velazquez, and N. Dehak, "Non-contrastive self-supervised learning for utterance-level information extraction from speech," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 2022.
- [13] H.-S. Heo, J.-w. Jung, J. Kang, Y. Kwon, Y. J. Kim, B.-J. Lee, and J. S. Chung, "Curriculum learning for self-supervised speaker verification," *Interspeech*, 2023.
- [14] 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 (JSTSP)*, 2022.
- [15] Y. Chen, S. Zheng, H. Wang, L. Cheng, and Q. Chen, "Pushing the limits of self-supervised speaker verification using regularized distillation framework," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [16] W. H. Kang, J. Alam, and A. Fathan, "L-mix: A latent-level instance mixup regularization for robust self-supervised speaker representation learning," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 2022.
- [17] 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 (TASLP)*, 2023.
- [18] 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 Processing (TASLP)*, 2024.
- [19] D. Dwivedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [20] A. Feizi, R. Balestrieri, A. Romero-Soriano, and R. Rabbany, "Gps-ssl: Guided positive sampling to inject prior into self-supervised learning," *arXiv preprint library*, 2024.
- [21] T. Lepage and R. Dehak, "Self-supervised frameworks for speaker verification via bootstrapped positive sampling," *Submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2025.
- [22] R. Balestrieri, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," *arXiv preprint library*, 2023.
- [23] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech*, 2017.
- [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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [25] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint library*, 2015.
- [26] T. Lepage and R. Dehak, "Additive margin in contrastive self-supervised frameworks to learn discriminative speaker representations," in *The Speaker and Language Recognition Workshop (Odyssey)*, 2024.
- [27] T. Lepage and R. Dehak, "Label-efficient self-supervised speaker verification with information maximization and contrastive learning," in *Interspeech*, 2022.
- [28] S. H. Mun, W. H. Kang, M. H. Han, and N. S. Kim, "Unsupervised representation learning for speaker recognition via contrastive equilibrium learning," *arXiv preprint library*, 2020.
- [29] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [30] B. Han, Z. Chen, and Y. Qian, "Self-supervised speaker verification using dynamic loss-gate and label correction," in *Interspeech*, 2022.
- [31] Z. Jin, Y. Tu, and M.-W. Mak, "W-gvkt: Within-global-view knowledge transfer for speaker verification," in *Interspeech*, 2024.
- [32] Z. Jin, Y. Tu, and M.-W. Mak, "Self-supervised learning with multi-head multi-mode knowledge distillation for speaker verification," in *Interspeech*, 2024.