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

Théo Lepage, Réda Dehak

EPITA Research Laboratory (LRE), France

Corresponding author: theo.lepage@epita.fr

Source code and resources: https://github.com/theolepage/sslsv

Introduction
Speaker Verification (SV)

Speaker Verification (SV) corresponds to the task of determining whether an unknown voice matches a claimed speaker identity.

- **Objective:** Learn representations that capture a speaker's identity, enabling comparison to produce a score used to accept or reject the target-speaker hypothesis.
- **Applications:** Forensic, Authentication, Information structuring, Human-machine interactions

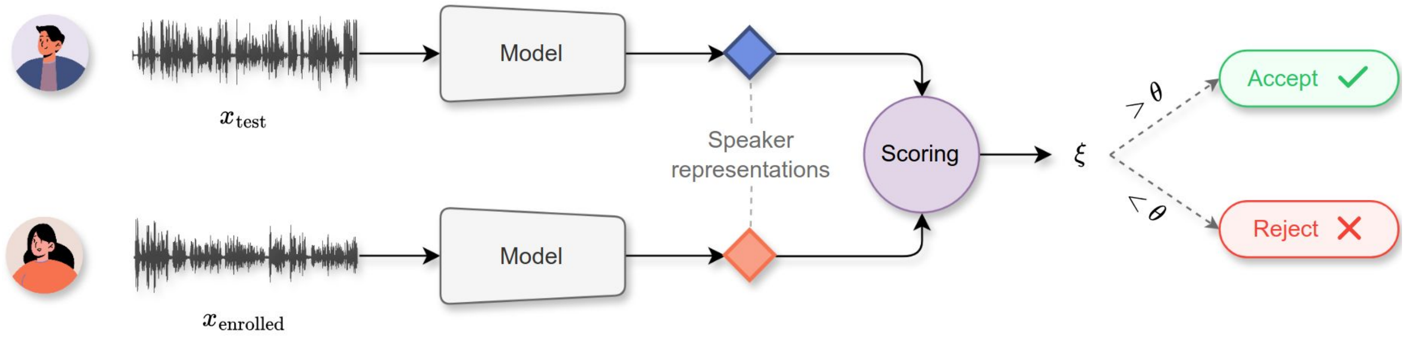


Figure 1. Overview of the general SV framework.

Introduction
Speaker Verification (SV)

State-of-the-art methods pre-train Deep Neural Networks (DNN) on a speaker classification task to learn these speaker representations [1, 2].

Optimal speaker representations should:

- maximize inter-speaker distances ;
- minimize intra-speaker variance ;
- discard extrinsic variabilities
  (e.g., channel, noise/environment, age, health ...).

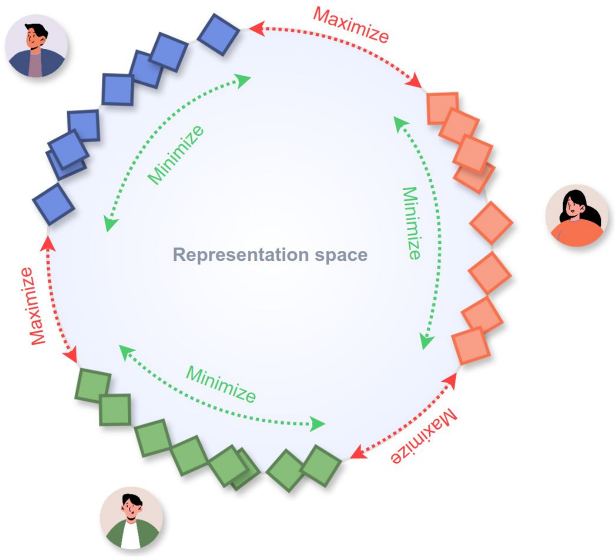


Figure 2. Illustration of an optimal speaker representation space.

[1] D. Snyder et al., “X-Vectors: Robust DNN Embeddings for Speaker Recognition”, ICASSP, 2018.
[2] J. S. Chung et al., “Delving into VoxCeleb: Environment Invariant Speaker Recognition”, Odyssey, 2020.

Introduction

Self-Supervised Learning (SSL)

Supervised learning is considered a bottleneck to the development of more intelligent systems:

1. Labeling datasets is expensive, tedious and slow.
2. Manual labeling is not scalable to the amount of data available today.
3. Human annotators can introduce biases.

Self-Supervised Learning (SSL) relies on supervisory signals generated from the data itself without human supervision. The model is pre-trained on a pretext task to learn relevant representations for a downstream task.

Introduction

Self-Supervised Learning (SSL)

The general **SSL training framework**:

1. Generates an anchor and a positive from an unlabeled audio waveform with data-augmentation;
2. Creates representations (evaluation/downstream task) and embeddings (training/pretext task);
3. Employs a loss that maximizes the similarity between the embeddings of the anchor and the positive.

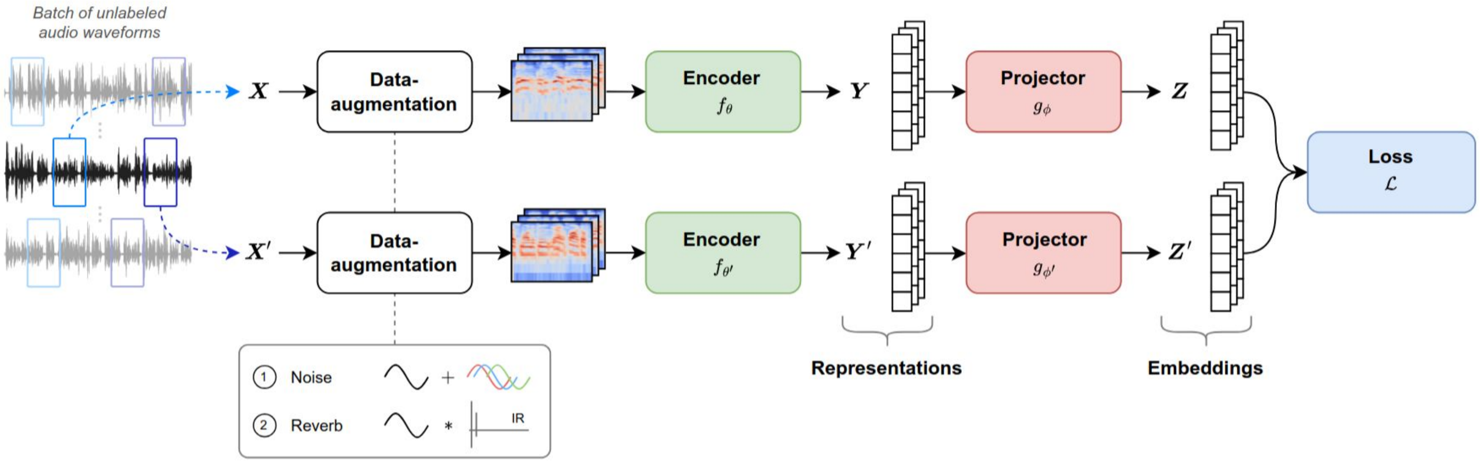


Figure 3. Standard SSL training framework for SV.

Introduction

Self-Supervised Learning (SSL)

SimCLR [1] is based on contrastive learning:

➔ Maximize the similarity of anchor-positive pairs while minimizing the similarity of anchor-negative pairs.

➔ Negatives are sampled from the current training batch.

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

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

DINO [2] is based on self-distillation:

➔ A student is trained to predict the output distribution of a teacher.

➔ The teacher's weights are updated via an EMA of the student's weights and centering + sharpening are applied to avoid collapse.

➔ Additional views: 4 short and 2 long segments.

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

where $H(\boldsymbol{a}, \boldsymbol{b}) = -\,\text{softmax}(\boldsymbol{a}) \log\left(\text{softmax}(\boldsymbol{b})\right)$, τ_{t} is the temperature for the teacher, τ_{s} is the temperature for the student, and $\boldsymbol{c}$ is a running mean on the teacher outputs.

[1] T. Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations", ICML, 2020.
[2] M. Caron et al., "Emerging Properties in Self-Supervised Vision Transformers", ICCV, 2021.

Method: Self-Supervised Positive Sampling

Motivation

SSL frameworks rely heavily on channel information (*e.g., VoxCeleb videos collected "in the wild"*) even with data-augmentation techniques because **anchor-positive pairs** are sampled from **the same utterance**.

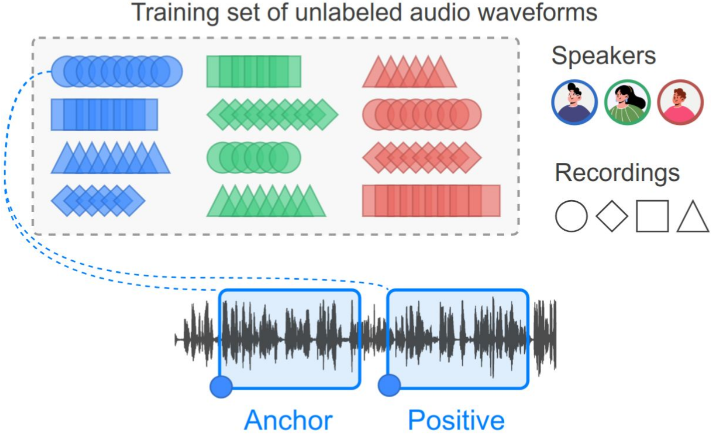


Figure 4. Overview of the default SSL same-utterance positive sampling.

Method	Pos. sampling	EER (%)	$\text{minDCF}_{0.01}$
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 1. SV performance with SSL and Supervised positive sampling using SimCLR and DINO frameworks (ECAPA-TDNN).

Method: Self-Supervised Positive Sampling

Related work

Several methods have been proposed in the literature to address the limitation of the default SSL same-utterance positive sampling.

- ❏ AAT [1] — Adversarial loss penalizing the model from learning data-aug information
- ❏ i-mixup [2] — Mixing utterances to create diverse synthetic training samples
- ❏ DPP [3] — Find diverse positives by relying on speech and face data
- ❏ CA-DINO [4] — Cluster speaker representations to determine appropriate positives

[1] J. Huh et al., “Augmentation Adversarial Training for Self-Supervised Speaker Representation Learning”, NeurIPS Workshop, 2020.
[2] W. H. Kang et al., “Robust Self-Supervised Speaker Representation Learning Via Instance Mix Regularization”, ICASSP, 2022.
[3] R. Tao et al., “Self-Supervised Training of Speaker Encoder with Multi-Modal Diverse Positive Pairs”, IEEE TASLP, 2023.
[4] B. Han et al., “Self-Supervised Learning With Cluster-Aware-DINO for High-Performance Robust Speaker Verification”, IEEE TASLP, 2024.

Method: Self-Supervised Positive Sampling
Overview

Objective: Finding anchor-positive pairs <u>from different recordings</u> of the same speaker.

Assumption: SSL same-utterance positive sampling group samples of the same recordings (sharing similar channel information) before modeling speaker identities.

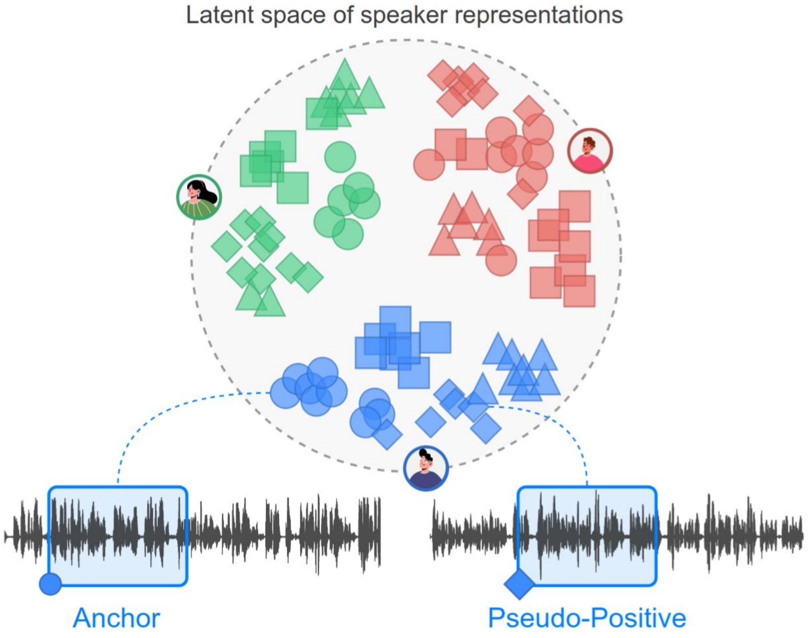


Figure 5. Overview of the positive sampling proposed in SSPS.

Method: Self-Supervised Positive Sampling
Framework

- **Epoch initialization:** SSPS performs clustering at the beginning of each epoch on reference representations derived from longer and non-augmented audio segments.
- **Training iteration:** The positive is substituted by a pseudo-positive, which is retrieved from a memory queue of previous positive embeddings, based on the anchor's clustering assignment.

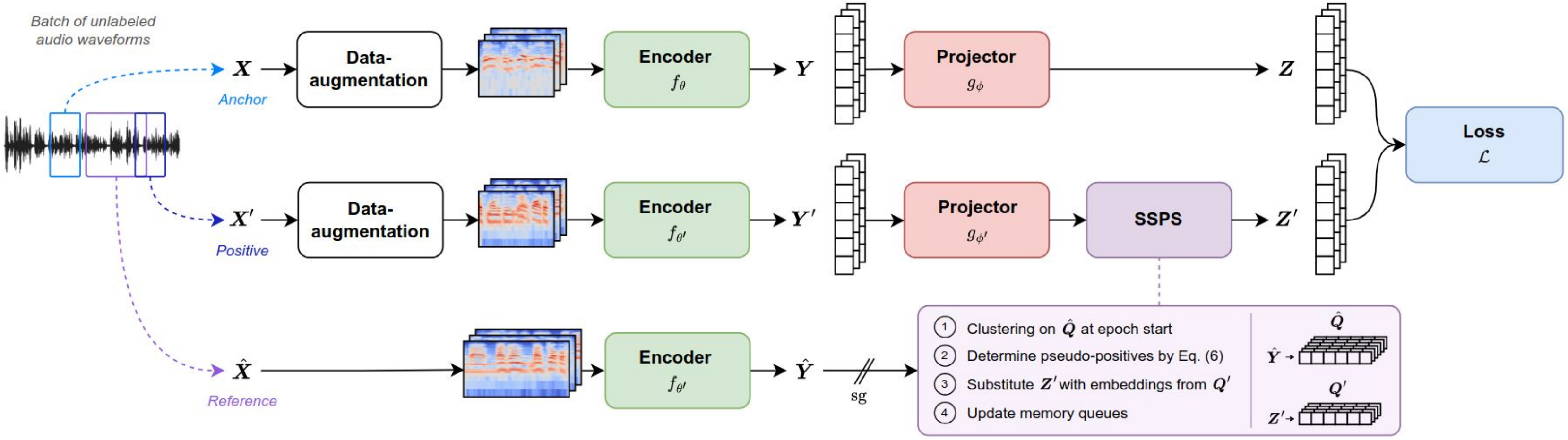


Figure 6. SSL training framework for SV with Self-Supervised Positive Sampling (SSPS).

Method: Self-Supervised Positive Sampling

Pseudo-positive sampling

Same-cluster sampling:

Utterances from the same cluster can be considered as pseudo-positives if K tends to the number of speaker identities in the train set.

$$\hat{c}_i = c_i$$

Neighboring-clusters sampling:

Utterances from neighboring clusters can also be considered as pseudo-positives if K tends to the number of recording sessions in the train set.

$$\hat{c}_i = \mathrm{sample}\left(\mathcal{C}_{c_i}\right)$$

$$\mathcal{C}_k \triangleq \underset{j \neq k}{\mathrm{top}\, M} \left(\left\{\mathrm{sim}\left(\boldsymbol{m}_k, \boldsymbol{m}_j\right), \forall j \in [1, K]\right\}\right)$$

Notations:

K is the number of clusters
M is the size of the sampling window
c_i is the cluster index of the i-th utterance
$\boldsymbol{m}_k$ is the centroid for the k-th cluster

Experimental setup

Datasets

VoxCeleb [1] is a large-scale audio dataset consisting of speech extracted from interview videos on YouTube.

Train data: VoxCeleb2 (1,092,009 utterances from 5,994 speakers)

Data-augmentation: Noise + Reverberation

Test data: VoxCeleb1 (148,642 utterances from 1,211 speakers)

Trials: Original (O), Extended (E) and Hard (H)

Evaluation

For each enrollment-test pair, the score is the cosine similarity between the representations.

Metrics:

- EER (Equal Error Rate) ↓
- minDCF (mininum Detection Cost Function) ↓

Models & Training

Input: 2/4 seconds → 40-d log-mel spectrogram

Encoder: Fast ResNet-34 / ECAPA-TDNN (1024)

Hyperparams: *Refer to the article*

GPUs: 2x/4x NVIDIA A100 80GB

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

SSPS

Activation: 20 epochs at the end of SSL training

Clustering: k-means (10 iterations) with custom PyTorch GPU implementation

Queues: $|\hat{\boldsymbol{Q}}| = N \quad |\boldsymbol{Q}'| = K$

Reference frame: 4 seconds (no data-aug.)

[1] J. S. Chung et al., "VoxCeleb2: Deep Speaker Recognition", Interspeech, 2018.

Results

Hyper-parameters search

Pos. sampling	K	M	EER (%)	$\mathbf{minDCF_{0.01}}$
SSL			9.41	0.6378
SSPS	6,000	0	6.63	0.5493
	10,000	0	6.82	0.5629
	25,000	0	7.30	0.5805
		1	5.80	**0.5250**
		2	**5.73**	0.5258
	150,000	0	8.29	0.6170
		1	7.54	0.5923
		2	7.13	0.5711
Supervised			3.93	0.3900

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

- Sampling from the same speaker class as the anchor reduces the EER to 6.63%.
- Sampling from a neighboring recording class further reduces the EER to 5.80%.
- This value of K (< 150,000) suggests that some recordings are already grouped in the latent space.
- This demonstrates the effectiveness of the neighboring-clusters sampling strategy to generate appropriate and diverse anchor-positive pairs.

Results

Performance on SV

- SSPS improves the performance of both SimCLR and DINO on VoxCeleb1-O, reducing the gap with the supervised baseline of 1.34% EER.
- SimCLR achieves a remarkable improvement over its baseline (58% EER reduction).
- SimCLR matches DINO's performance with a simpler framework and achieves the best SSL performance using Supervised positive sampling (Table 1), highlighting the potential for further improvements of SSL contrastive methods.
- SimCLR-SSPS and DINO-SSPS outperform other state-of-the-art SSL methods for SV by providing an explicit solution to their main limitation.

Method	EER (%)	$\text{minDCF}_{0.01}$
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**

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

Results

Visualization of speaker representations

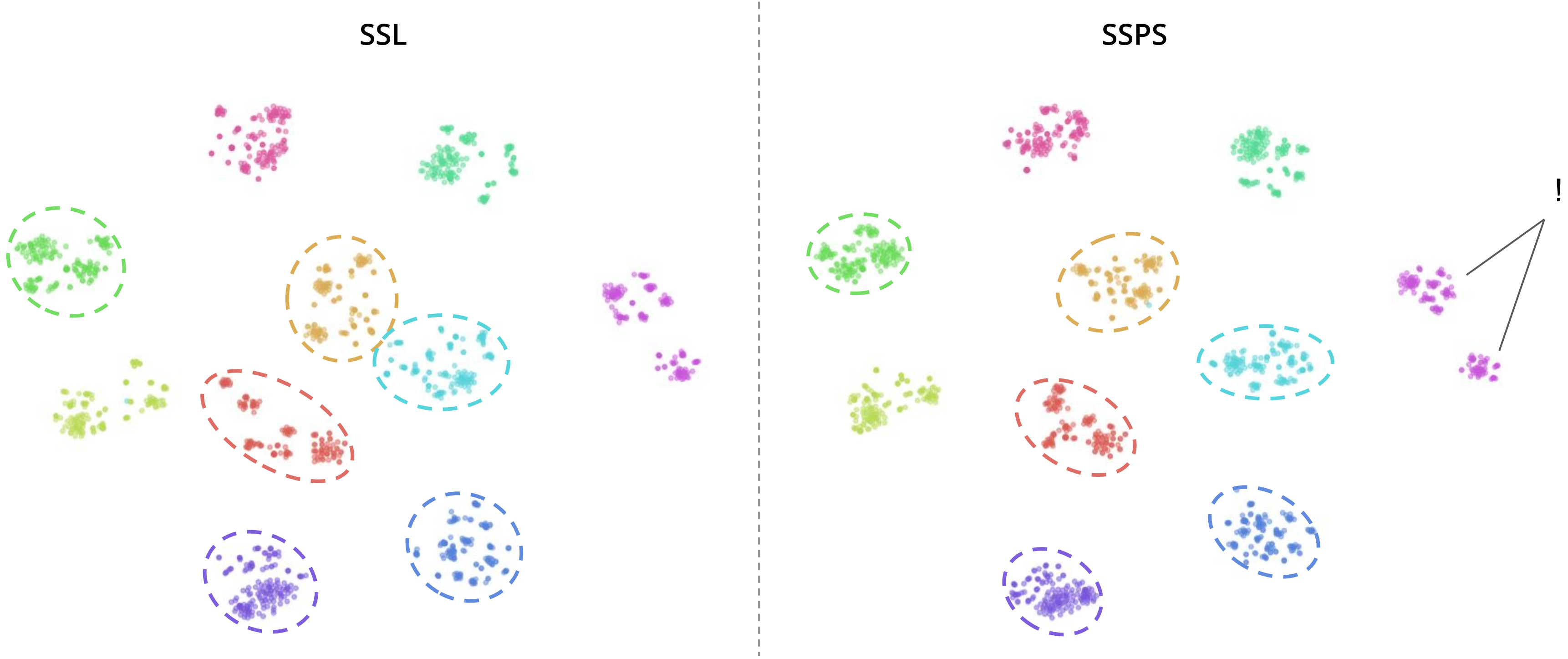


Figure 7. t-SNE of representations from 10 speakers of VoxCeleb1 with SSL and SSPS.

Additional results

… from the extended journal version [1]

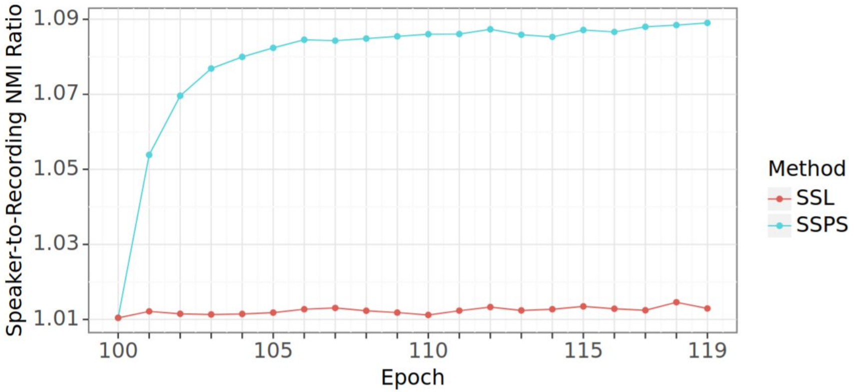


Figure 8. NMI ratio of speaker and recording information with SSL and SSPS across training epochs.

↓

SSPS learns representations that are more robust to extrinsic variabilities, arising from the different recording conditions.

SSPS is robust to the absence of data-augmentation.

This property is very important as data-augmentation is fundamental for SSL but presents shortcomings.

↑

Positive sampling	Data-aug.	VoxCeleb1-O	
		EER (%)	$\mathrm{minDCF}_{0.01}$
SSL	✓	**6.30**	**0.5286**
	✗	15.00	0.7575
SSPS	✓	**2.57**	0.3033
	✗	2.77	**0.2840**

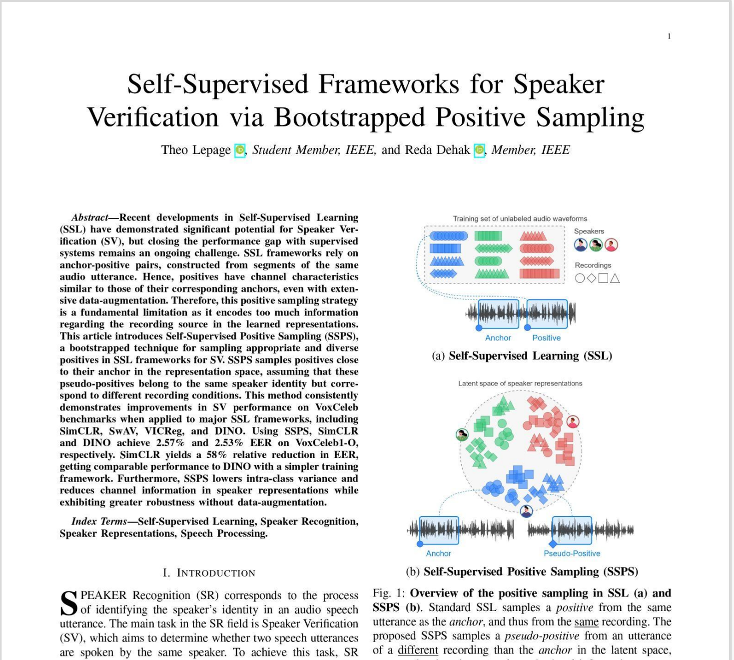
Table 4. Effect of data-augmentation on SV performance with SSL and SSPS.

[1] T. Lepage et al., "Self-Supervised Frameworks for Speaker Verification via Bootstrapped Positive Sampling", IEEE TASLP, 2025.

Conclusions

- SSPS overcomes the main limitation of SSL frameworks (i.e., same-utterance positive sampling) by reducing intra-speaker variance.
- SimCLR-SSPS and DINO-SSPS achieve 2.57% and 2.53% EER on VoxCeleb1-O, advancing the field towards supervised performance.
- SimCLR-SSPS results in a 58% EER reduction which motivates the need to re-consider SSL contrastive-based frameworks for SV.

Read more about this work in the extended journal article: https://ieeexplore.ieee.org/document/11075552



1

Self-Supervised Frameworks for Speaker Verification via Bootstrapped Positive Sampling

Theo Lepage, *Student Member, IEEE,* and Reda Dehak, *Member, IEEE*

***Abstract*—Recent developments in Self-Supervised Learning (SSL) have demonstrated significant potential for Speaker Verification (SV), but closing the performance gap with supervised systems remains an ongoing challenge. SSL frameworks rely on anchor-positive pairs, constructed from segments of the same audio utterance. Hence, positives have channel characteristics similar to those of their corresponding anchors, even with extensive data-augmentation. Therefore, this positive sampling strategy is a fundamental limitation as it encodes too much information regarding the recording source in the learned representations. This article introduces Self-Supervised Positive Sampling (SSPS), a bootstrapped technique for sampling appropriate and diverse positives in SSL frameworks for SV. SSPS samples positives close to their anchor in the representation space, assuming that these pseudo-positives belong to the same speaker identity but correspond to different recording conditions. This method consistently demonstrates improvements in SV performance on VoxCeleb benchmarks when applied to major SSL frameworks, including SimCLR, SwAV, VICReg, and DINO. Using SSPS, SimCLR and DINO achieve 2.57% and 2.53% EER on VoxCeleb1-O, respectively. SimCLR yields a 58% relative reduction in EER, getting comparable performance to DINO with a simpler training framework. Furthermore, SSPS lowers intra-class variance and reduces channel information in speaker representations while exhibiting greater robustness without data-augmentation.**

***Index Terms*—Self-Supervised Learning, Speaker Recognition, Speaker Representations, Speech Processing.**

(a) **Self-Supervised Learning (SSL)**

(b) **Self-Supervised Positive Sampling (SSPS)**

Fig. 1: **Overview of the positive sampling in SSL (a) and SSPS (b)**. Standard SSL samples a *positive* from the same utterance as the *anchor*, and thus from the same recording. The proposed SSPS samples a *pseudo-positive* from an utterance of a different recording than the *anchor* in the latent space,

I. Introduction

SPEAKER Recognition (SR) corresponds to the process of identifying the speaker's identity in an audio speech utterance. The main task in the SR field is Speaker Verification (SV), which aims to determine whether two speech utterances are spoken by the same speaker. To achieve this task, SR

Corresponding author: theo.lepage@epita.fr

Source code and resources: https://github.com/theolepage/sslsv