An efficient clustering algorithm for self-supervised speaker recognition

Abderrahim Fathan, Xiaolin Zhu, Jahangir Alam.
Computer Research Institute of Montreal, Montreal (Quebec) H3N 1M3, Canada
abderrahim.fathan@crim.ca, alice.zhuxl@gmail.com, jahangir.alam@crim.ca

Abstract

Clustering-based pseudo-labels (PLs) are widely used to optimize speaker embedding (SE) networks and train self-supervised speaker verification (SV) systems. However, PL-based self-supervised training depends on high-quality PLs and clustering performance relies heavily on time- and resource-consuming data augmentation regularization. In this paper, we propose an efficient and general-purpose multi-objective clustering algorithm that outperforms all other baselines used to cluster SEs. Our approach avoids explicit data augmentation for fast training and low memory and compute resource usage. It is based on three principles: (1) Self-Augmented Training to enforce representation invariance and maximize the information-theoretic dependency between samples and their predicted PLs (2) Virtual Mixup Training to impose local-Lipschitzness and enforce the cluster assumption (3) Supervised contrastive learning to learn more discriminative features and pull samples of same class together and push apart samples of different clusters, while improving robustness to natural corruptions. We provide a thorough comparative analysis of the performance of our clustering method vs. baselines using a variety of clustering metrics and show that we outperform all other clustering benchmarks, perform an ablation study to analyze the contribution of each component including two other augmentation-based objectives, and show that our multi-objective approach provides beneficial complementary information. Moreover, using the generated PLs to train our SE system allows us to achieve state-of-the-art SV performance.

1 Introduction

Speaker verification (SV) is the task of confirming, based on a speaker's known utterances, that the identity of a speaker is who they purport to be. In recent years, it has become a key technology for personnel authentication in numerous applications [27]. Typically, utterance-level fixed-dimensional embedding vectors are extracted from the enrollment and test speech samples and then fed into a scoring algorithm (e.g., cosine distance) to measure their likelihood of being from the same speaker. Classically, the i-vector framework has been one of the most dominant approaches for speech embedding [13, 33] thanks to its ability to summarize the distributive patterns of speech in an unsupervised manner and with relatively small training datasets. It generates fixed-sized compact vectors that represent the speaker's identity in a speech utterance regardless of its length. Besides, in the past years, various deep learning-based architectures and techniques have been proposed to extract embeddings [3, 32, 21]. They have shown great performance when large training datasets are available, particularly with a sufficient number of speakers [51]. One widely employed architecture for this purpose is ECAPA-TDNN [15], which has achieved state-of-the-art (SOTA) performance in text-independent speaker recognition. The latter uses squeeze-and-excitation (SE), employs channel- and context-dependent statistics pooling & multi-layer aggregation and applies self-attention pooling to obtain an utterance-level embedding vector.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Indeed, most of the deep embedding models are trained in a fully supervised manner and require large speaker-labeled datasets for training. However, well-annotated datasets can be expensive and time-consuming to prepare, which has led the research community to explore more affordable self-supervised learning (SSL) techniques using larger unlabeled datasets. One common way to solve this issue for SV systems is to use a one-stage "clustering-classification" scheme [31, 32, 21] by employing clustering algorithms (e.g., K-means, agglomerative hierarchical clustering, spectral clustering) or other SSL-based objectives (e.g., SimCLR, MoCo [59]) to generate Pseudo-Labels (PLs) and train the speaker embedding network using these labels in a discriminative fashion. More recently, better-performing ways have started to appear which are now widely adopted in the SV domain. These frameworks are based on two-stage progressive/iterative "clustering-classification" learning [46, 52]. The first stage consists of SSL training (e.g., contrastive InfoNCE loss [52]) to train an encoder model to generate speaker embeddings, followed by a second stage of clustering those embeddings to produce pseudo-labels in order to jointly train the encoder with a classifier in a supervised manner. The two stages are repeated sequentially until no gains are obtained. Despite the impressive performance of these PL-based Self-Supervised SV schemes, clustering performance remains a bottleneck in all above approaches [52, 25] as downstream performance relies greatly on accurate PLs since these are in general noisy and inaccurate due to the discrepancy between the clustering objective(s) and the final SV task. Besides, even with iterative clustering-classification paradigms, the erroneous information from the wrong PLs keeps propagating iteratively, which degrades the final performance [52, 38]. Thus, the need for better-performing clustering algorithms to generate less noisy and more accurate PLs. Rather than using SOTA deep clustering models which rely on heavy domain-specific data augmentations, these approaches usually employ classical clustering algorithms such as K-means or Spectral clustering as these are easier to use, faster, and less resource-consuming in terms of memory and GPU/CPU resources to train. More discussion about related research and the motivation of our work is available in Appendix A.

In this paper, we propose an efficient and general-purpose multi-objective clustering algorithm that outperforms all other baselines used to cluster speaker embeddings. Our approach avoids using explicit data augmentation for fast training and low memory and compute resource usage. It is based on the combination of three principles: (1) Self-Augmented Training to enforce representation invariance and maximize the information-theoretic dependency between samples and their predicted pseudo-labels through the Information Maximizing Self-Augmented Training (IMSAT) clustering framework [29](2) Virtual Mixup Training (VMT) [40] to impose local-Lipschitzness which enforces the cluster assumption (3) Supervised contrastive learning [34] by leveraging on-the-fly generated pseudo-labels, to pull samples of same class together and push samples of different clusters apart. Instead of mixing up inputs or using contrastive loss for the sole goal of enforcing smoother model responses and compactness of the embeddings, our method leverages successfully these predictions as additional supervisory signals to better guide the cluster assignment for more robust, stable, and better-performing data clustering.
The contributions of this paper are as follows:

- We propose a novel general-purpose multi-objective clustering algorithm for large-scale datasets or/and a high number of clusters.
- We explore various recent SOTA SSL objectives for clustering where we show that multi-objective clustering often provides beneficial complementary information.
- Our proposed method outperformed a large set of clustering baselines. Besides, using the generated PLs to train our SV systems, we were able to achieve high SV performance. Furthermore, employing augmentation-based SSL objectives, allowed us to achieve both SOTA speaker embedding clustering and SV performance.

2 Our proposed clustering approach

Given a deep neural network-based clustering model f to train with a predefined number of clusters C, our clustering approach constrains the output predictions of the model to remain unchanged under local perturbations and implicit Virtual Mixup Training (VMT) [40] data augmentations L_{Mixup} in an end-to-end fashion in order to improve robustness against perturbations and impose local-Lipschitzness on the learned weights to favore the cluster assumption [23] (if samples are in the same cluster, they come from the same class) which is a critical condition for successful clustering. Besides, employing Information Maximizing Self-Augmented Training (IMSAT) L_{IMSAT} [29] maximises mutual information (MI) in an end-to-end fashion between data and their clustering assignments by encouraging the prediction of the neural network to remain invariant under data

perturbation/augmentation, while maximizing the information-theoretic dependency between data and their predicted discrete representations. Additionally, leveraging the Supervised Contrastive loss L_{SupCon} [34] in an unsupervised way (using online generated pseudo-labels as labels and l2-normalized logits as feature embeddings) allows us to leverage online clustering assignments so that we use nearest-neighbors as positives rather than augmentations and clusters of points belonging to the same class are pulled together in embedding (logit) space, while simultaneously pushing apart clusters of samples from different classes. In our case, L_{SupCon} helps us to learn more discriminative features and has the advantage of improving robustness to natural corruptions and to out-of-distribution data [34]. It intrinsically performs hard positive/negative mining, and does also allow for multiple positives per anchor leveraging pseudo-label information, which can mitigate the risk of false positives during clustering. This is especially suitable and even more critical in our approach which avoids any type of external or domain specific transformations.
Our approach minimizes the following L_{total} objective:

$$L_{total} = L_{IMSAT} + L_{Mixup} + L_{SupCon} \tag{1}$$

$$\text{where} \quad L_{IMSAT} = R_{SAT}(\theta, T_{VAT}) + \lambda(H(Y|X) - \mu H(Y)), \tag{2}$$

$$\text{and} \quad L_{Mixup} = \frac{1}{N}\sum_{i=1}^{N} KL(\alpha_i p_i + (1-\alpha_i)p_{r_i} || f(\alpha_i x_i + (1-\alpha_i)x_{r_i})). \tag{3}$$

N is the size of data (or mini-batches), $r_i \in \{1, .., N\}$ is a random index, and $\alpha_i \in [0, 1]$ is the mixup interpolation coefficient. $KL(.||.)$ refers to the Kullback-Leibler divergence. $p_i = f(x_i) \in \mathbb{R}^{1xC}$, $p_{r_i} = f(x_{r_i})$ correspond to the predictions of data samples x_i and x_{r_i}. Figure 1 in Appendix D shows a schematic diagram of our framework. Our aim is to harness these objectives as additional supervisory signals to regularize the clustering model to produce consistent assignments. Moreover, we follow the general training framework depicted in Figure 2. For lack of space, all remaining mathematical details are included in Appendix E.

Table 1: An ablation study of our proposed clustering system including various SSL-based loss objectives that do not employ data augmentation (only original data samples). C denotes the predefined number of clusters. Results are reported in terms of Clustering metrics and the corresponding EER (%) downstream SV evaluation performance when using the generated pseudo-labels to train our studied SV system. Details about the clustering metrics can be found in Appendix B.

Model	**Clustering Metrics**										**Speaker Verification**
	ACC	**AMI**	**NMI**	**No. of clusters**	**Completeness**	**Homogeneity**	**FMI**	**Purity**	**Silhouette**	**CHS**	**DBS**	**EER (%)**
L_{Mixup} (C: 10k)	0.013	0.016	0.432	10000	0.413	0.452	0.001	0.026	-0.019	1.001	17.633	9.767
L_{SupCon} (C: 10k)	0.015	0.02	0.419	10000	0.404	0.434	0.001	0.027	-0.036	1.001	19.5	20.074
L_{VICReg} [4] (C: 5k)	0.018	0.082	0.27	4496	0.309	0.239	0.004	0.021	-0.134	1.001	18.031	11.612
L_{IMSAT} (C: 5k)	0.578	0.731	0.822	5000	0.83	0.815	0.552	0.604	-0.033	**1.002**	26.56	4.507
L_{IMSAT} (C: 5994)	0.600	0.743	0.833	5993	0.834	0.831	0.583	0.636	-0.074	0.999	23.915	4.295
L_{IMSAT} (C: 10k)	0.621	0.754	0.844	9844	0.836	0.853	0.616	0.678	-0.122	0.999	16.897	4.438
$L_{Mixup} + L_{SupCon}$ (C: 10k)	0.015	0.034	0.354	9639	0.36	0.348	0.002	0.023	-0.133	0.999	**15.563**	12.54
$L_{IMSAT} + L_{Mixup} + L_{VICReg'variance} + L_{VICReg'covariance}$ (C: 5k)	0.013	0.018	0.369	5000	0.367	0.371	0.001	0.017	**-0.015**	1.0	25.571	19.952
$L_{IMSAT} + L_{Mixup} + L_{SupCon} + L_{VICReg'variance} + L_{VICReg'covariance}$ (C: 5k)	0.014	0.02	0.36	5000	0.361	0.359	0.001	0.017	-0.022	0.999	26.667	21.84
$L_{IMSAT} + L_{Mixup}$ (C: 10k)	0.628	0.764	0.852	9791	0.841	0.862	0.615	0.692	-0.149	1.0	17.297	4.321
$L_{IMSAT} + L_{SupCon}$ (C: 5k)	0.497	0.688	0.784	4996	0.81	0.76	0.347	0.516	-0.065	0.999	24.809	4.623
$L_{IMSAT} + L_{SupCon}$ (C: 5994)	0.514	0.697	0.793	5974	0.814	0.774	0.347	0.538	-0.117	0.999	22.164	4.475
$L_{IMSAT} + L_{SupCon}$ (C: 10k)	0.548	0.717	0.813	9585	0.823	0.803	0.361	0.589	-0.138	1.001	15.941	4.348
$L_{IMSAT} + L_{Mixup} + L_{SupCon}$ (C: 5k)	0.602	0.751	0.836	4999	0.842	0.831	0.579	0.632	-0.071	0.999	26.905	**4.231**
$L_{IMSAT} + L_{Mixup} + L_{SupCon}$ (C: 5994)	0.619	0.761	0.846	5989	0.845	0.846	0.6	0.66	-0.125	1.002	24.301	4.321
$L_{IMSAT} + L_{Mixup} + L_{SupCon}$ (C: 10k)	**0.639**	**0.776**	**0.86**	9685	**0.847**	**0.873**	**0.642**	**0.71**	-0.136	0.998	17.599	4.252
$L_{IMSAT} + L_{Mixup} + L_{SupCon} + L_{aug}$ (C: 10k)	0.714	0.834	0.894	7810	0.887	0.901	0.728	0.773	**-0.129**	0.999	19.768	3.377
$L_{IMSAT} + L_{Mixup} + L_{SupCon} + L_{aug} + L_{InfoNCE}$ (C: 10k)	**0.725**	**0.842**	**0.9**	8500	**0.89**	**0.91**	**0.746**	**0.792**	-0.134	**1.0**	18.407	**3.362**

3 Results and Discussion

As input to all of our clustering algorithms, we employ 400-dim i-vectors. The compact i-vectors, which are unsupervised speaker representations, allow us here to perform clustering in a more efficient way and to avoid high dimensionality of the MFCC acoustic features.

In order to evaluate the performance of our proposed clustering approach and the generated PLs for self-supervised speaker verification, we conducted a set of experiments based on the VoxCeleb2 dataset [9]. To train the embedding networks, we used the development subset of the VoxCeleb2 dataset, consisting of 1,092,009 utterances collected from 5,994 speakers. The evaluation was performed according to the original VoxCeleb1 trials list [43], which consists of 37,720 trials of 4,874 utterances spoken by 40 speakers.

For our ECAPA-TDNN-based SV system, the acoustic features used in the experiments were 40-dimensional Mel-frequency cepstral coefficients (MFCCs) extracted at every 10 ms, using a 25 ms

Hamming window via Kaldi toolkit [47]. Moreover, to follow other SV works in training the ECAPA-TDNN-based systems, we have used waveform-level data augmentations including additive noise and room impulse response (RIR) simulation [51]. In addition to the waveform-level augmentations, we have also applied augmentation over the extracted MFCCs feature, analogous to the specaugment scheme [45].

In Table 1, we performed a large-scale ablation study to analyze the contribution of all components of our system and the influence of the predefined number of clusters. We also study the VICReg method [4] which comprises a term $L_{VICReg'variance}$ that maintains the variance of each embedding dimension above a threshold and a term $L_{VICReg'covariance}$ that decorrelates each pair of variables. Results show that there is complementary information between all loss terms in our proposed objective and that each help to boost the performance of the overall clustering framework. We also observe that choosing a much higher number of clusters than ground truth leads to improved clustering performance across all studied systems. Additionally, compared to a large variety of 15 clustering benchmarks in [21, Tab. 1], we can observe that our proposed method outperforms all other baselines in terms of clustering metrics achieving 63.9% unsupervised clustering accuracy compared to 58.7% for AHC which was the best performing method (8.9% relative improvement), while having a compute time comparable to classical clustering models (3-4 days). Using our proposed system's generated PLs to train our speaker embedding system, also allowed us to achieve a very competitive downstream SV EER performance outperforming all other benchmarks, except the AHC PLs which lead to a slightly better performance. Moreover, using audio data augmentation by incorporating L_{aug} to push two augmented versions of the same sample closer and $L_{InfoNCE}$ for contrastive learning helps to further boost both clustering and downstream SV performance better than all our studied baselines, and shows that there is complementarity between our studied objectives. Details of these objectives are included in Appendix F.
Furthermore, to improve generalization and mitigate the effect of noisy PLs during training of our speaker embedding system, in Table 3 in Appendix C we extend our investigation to other recent margin-based softmax losses. Unlike the widely used AAMSoftmax loss in speaker verification, to our knowledge, our results indicate for the first time that variants such as OCSoftmax using one-class learning instead of multi-class classification and not assuming the same distribution for all speakers (which is more realistic in our case) or the recent AdaFace loss which emphasizes misclassified samples according to the quality of speaker embeddings (via feature norms), perform consistently better across all pseudo-labels and the ground truth labels. We could also observe, in the case of IMSAT and our proposed system, that even if clustering performance is better when the predefined number of clusters is high (10000), the speaker verification performance tends to be better when this number is close to the ground truth 5994 (e.g., 5000).

Table 2: Some recent SOTA SSSV approaches in EER (%) compared to our simple SV system trained with our PLs. All models are based on ECAPA-TDNN. Results are reported on the original VoxCeleb1 test set (Voxceleb1_O).

SSL Objective	**EER (%)**
MoBY [59]	8.2
InfoNCE [52]	7.36
MoCo [8]	7.3
ProtoNCE [59]	7.21
PCL [59]	7.11
CA-DINO [26]	3.585
i-mix [20]	3.478
l-mix [20]	3.377
Iterative clustering [52]	3.09
Our approach (without data augmentation)	**3.924**
Our approach (with data augmentation)	**3.001**

Finally, Table 2 shows a comparison of our approach (w/ and w/o augmentation) for Self-Supervised SV (SSSV) training using our system-based PLs compared to recent SOTA SSSV approaches employing diverse SSL objectives with the same ECAPA-TDNN model encoder. The results show clearly that our approach provides very competitive performance while being simple and fast. Besides, when employing augmentations, our approach outperforms all the baselines, which suggests that regularization through data augmentation is still crucial and that further gains can be made by simply improving the clustering modules of current self-supervised speaker recognition systems.

4 Conclusion

In this paper, we propose a general-purpose and multi-objective clustering method. Our approach avoids using explicit data augmentation for fast and efficient training. It is based on three principles: (1) Self-Augmented Training to enforce representation invariance and maximize the information-theoretic dependency between samples and their predicted pseudo-labels (2) Virtual Mixup Training to impose local-Lipschitzness which enforces the cluster assumption (3) Supervised contrastive learning by leveraging on-the-fly generated pseudo-labels to pull samples of same class together and push samples of different clusters apart. Moreover, we explored various recent SOTA SSL objectives for clustering, including two data augmentation-based objectives, where we showed that our multi-objective approach provides beneficial complementary information. Our approach outperformed all other baselines used to cluster speaker embeddings and provided very competitive speaker verification performance outperforming all the benchmarks.

5 Acknowledgment

The authors wish to acknowledge the funding from the Government of Canada's New Frontiers in Research Fund (NFRF) through grant NFRFR-2021-00338 and Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN-2019-05381. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NFRF & NSERC.

References

[1] D. Arpit et al. A closer look at memorization in deep networks. In *ICML*, 2017.

[2] D. Bahri et al. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

[3] Z. Bai and X.-L. Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 2021.

[4] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[5] J. Bridle et al. Unsupervised classifiers, mutual information and'phantom targets. *NeurIPS*, 4, 1991.

[6] T. Caliński et al. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 1974.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] J. Cho et al. The jhu submission to voxsrc-21: Track 3. *arXiv preprint arXiv:2109.13425*, 2021.

[9] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[10] P. Dahal. Learning embedding space for clustering from deep representations. In *IEEE Big Data*, 2018.

[11] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE transactions on PAMI*, 1979.

[12] W. H. Day et al. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1984.

[13] N. Dehak et al. Front-end factor analysis for speaker verification. *IEEE/ACM Trans. Audio Speech Lang*, 2011.

[14] J. Deng et al. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on PAMI*, 2021. doi: 10.1109/TPAMI.2021.3087709.

[15] B. Desplanques et al. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech*. ISCA, 2020.

[16] N. Dilokthanakul et al. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[17] A. Dosovitskiy et al. Discriminative unsupervised feature learning with convolutional neural networks. *NeurIPS*, 2014.

[18] G. F. Elsayed et al. Large margin deep networks for classification, 2018.

[19] P. A. Estévez et al. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 2009.

[20] A. Fathan and J. Alam. On the influence of the quality of pseudo-labels on the self-supervised speaker verification task: a thorough analysis. In *IWBF*. IEEE, 2023.

[21] A. Fathan, J. Alam, and W. Kang. On the impact of the quality of pseudo-labels on the self-supervised speaker verification task. In *NeurIPS ENLSP Workshop*, 2022.

[22] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

[23] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *NeurIPS*, 17, 2004.

[24] S. Guha et al. Cure: An efficient clustering algorithm for large databases. *SIGMOD Rec.*, jun 1998. ISSN 0163-5808. URL `https://doi.org/10.1145/276305.276312`.

[25] B. Han, Z. Chen, and Y. Qian. Self-supervised speaker verification using dynamic loss-gate and label correction. *arXiv preprint arXiv:2208.01928*, 2022.

[26] B. Han et al. Self-supervised learning with cluster-aware-dino for high-performance robust speaker verification. *arXiv preprint arXiv:2304.05754*, 2023.

[27] J. H. Hansen and T. Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015. doi: 10.1109/MSP.2015.2462851.

[28] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, pages 100–108, 1979.

[29] W. Hu et al. Learning discrete representations via information maximizing self-augmented training. PMLR, 2017.

[30] Z. Jiang et al. Variational deep embedding: A generative approach to clustering. *CoRR, abs/1611.05148*, 1, 2016.

[31] W. H. Kang, J. Alam, and A. Fathan. An analytic study on clustering-based pseudo-labels for self-supervised deep speaker verification. In *SPECOM*, 2022.

[32] W. H. Kang, J. Alam, and A. Fathan. l-mix: a latent-level instance mixup regularization for robust self-supervised speaker representation learning. *JSTSP*, 2022.

[33] P. Kenny. A Small Footprint I-vector Extractor. In *Odyssey*, pages 1–6, 2012.

[34] P. Khosla, P. Teterwak, et al. Supervised contrastive learning. *NeurIPS*, 2020.

[35] M. Kim, A. K. Jain, and X. Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18750–18759, 2022.

[36] A. Krause et al. Discriminative clustering by regularized information maximization. *NeurIPS*, 23, 2010.

[37] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1):7–21, 2005.

[38] Y. Li et al. Contrastive clustering. In *AAAI*, 2021.

[39] W. Liu, Y. Wen, et al. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, 2016.

[40] X. Mao et al. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*, 2019.

[41] L. Meng et al. Mixspeech: Data augmentation for low-resource automatic speech recognition. In *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[42] T. Miyato et al. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *PAMI*, 2018.

[43] A. Nagrani et al. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[44] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[45] D. S. Park et al. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.

[46] J. Peng et al. Progressive Contrastive Learning for Self-Supervised Text-Independent Speaker Verification. In *Proc. of Odyssey Workshop*, 2022.

[47] D. Povey et al. The kaldi speech recognition toolkit. In *In IEEE workshop*, 2011.

[48] M. Ronen et al. Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of IEEE/CVF*, 2022.

[49] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of EMNLP-CoNLL*, pages 410–420, 2007.

[50] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[51] D. Snyder et al. X-vectors: Robust dnn embeddings for speaker recognition. In *IEEE-CASSP*, 2018. doi: 10.1109/ICASSP.2018.8461375.

[52] R. Tao et al. Self-supervised speaker recognition with loss-gated learning. In *ICASSP*. IEEE, 2022.

[53] A. Tomilov et al. STC Antispoofing Systems for the ASVspoof2021 Challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 61–67, 2021. doi: 10.21437/ASVSPOOF.2021-10.

[54] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[55] V. N. Vapnik et al. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[56] V. Verma et al. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

[57] F. Wang et al. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[58] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[59] W. Xia et al. Self-supervised text-independent speaker verification using prototypical momentum contrastive learning. In *ICASSP*. IEEE, 2021.

[60] J. Xie et al. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487. PMLR, 2016.

[61] N. Xuan et al. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. 2010.

[62] H. Zhang et al. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[63] T. Zhang et al. BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1997. ISSN 13845810. doi: 10.1023/a:1009783824328.

[64] Y. Zhang et al. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 2021.

A Background and Related Work

Diverse methods for clustering have been proposed. For instance, classical models include K-means [28], Gaussian mixture model (GMM), BIRCH [63], CURE [24], Agglomerative Hierarchical Clustering (AHC) [12], etc. However, these methods can only fit linear boundaries between data representations. Recently, the powerful representative ability of deep networks has been leveraged to model the non-linearity of complex data and to scale to large datasets. For instance, Deep Embedded Clustering (DEC) [60] proposed to use deep models to simultaneously learn feature representations and cluster assignments, while DeepCWRN [10] approach employs an autoencoder to simultaneously learn feature representations and embeddings suitable for clustering by encouraging the separation of natural clusters in the embedding space. Besides, other deep models have been proposed based on generative models [16, 30] or dynamic architectures [48].

While data augmentation remains a crucial component to regularize deep neural networks for clustering and unsupervised representation learning in order to model the invariance of learned representations [17], augmentation has the downside of increasing the training set which can lead to severalfold more training time, especially for large-scale datasets and neural networks. Besides, using blind augmentations can have a negative effect on the task of speaker verification/recognition as transformations like pitch perturbation or spectral augmentation can alter the identity of a speaker, leading sometimes to the creation of misleading data samples. Moreover, for real-world tabular data applications [2] such as genomics and clinical data, generating additional augmented views is not an obvious task and can be prohibited.

B Clustering Evaluation Metrics

Following the commonly used evaluation metrics for clustering, we evaluate our clustering models by thoroughly assessing the quality of their generated pseudo-labels from different perspectives.

We employ a list of 7 supervised metrics that are based on both the PLs and true labels (Unsupervised Clustering Accuracy, Normalized Mutual Information [19], Adjusted MI [61], Completeness score [49], Homogeneity score [49], Purity score, and Fowlkes-Mallows index [22]). Among the criteria that these metrics assess, we can list the following: clustering accuracy and mutual information as measures of the consistency between the true labels and the generated PLs, homogeneity, completeness, and purity of clusters, and precision and recall. Additionally, we compute 3 unsupervised metrics (Silhouette score [50], Calinski-Harabasz score [6], and Davies-Bouldin score [11]) that are solely based on the generated PLs and the data samples, and which allow us to measure how compact or scattered are the clusters (e.g., intra-class dispersion, between-cluster distances, nearest-cluster distance).

More details and discussion are available at [21], which found a very high correlation between these metrics and SV performance. Additionally, using the 3 unsupervised clustering metrics allows us to assess objectively our clustering performance and avoid arbitrary techniques such as t-SNE visualizations [54]. To compute these metrics, we use available implementations from the scikit-learn toolkit. Details of the clustering metrics are as follows:

- **Unsupervised Clustering Accuracy (ACC)**: measures the consistency between the true labels and the generated PLs. $ACC = \max_{m} \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = m(c_i)\}}{N}$ where y_i is the true label, c_i is the generated PL assignment, and m is a mapping function which ranges over all possible one-to-one mappings between true labels and assignments. The optimal mapping can be efficiently computed using the Hungarian algorithm [37].
- **Normalized Mutual Information (NMI)** [19]: $NMI(Y, C) = \frac{I(Y,C)}{\frac{1}{2}[H(Y)+H(C)]}$ where Y and C denote the ground-truth labels and the clustering assignments, respectively. H is the entropy function and I denotes the MI metric. NMI is the harmonic mean between below homogeneity and completeness scores.
- **Adjusted MI (AMI)** [61]: Since the NMI measure is not adjusted for chance, including the adjusted MI score might be preferred for comparison in some of our cases.
- **Completeness score** [49]: A clustering assignment satisfies completeness if all the data points that are members of a given class are elements of the same cluster. The scores are between 0 and 1, where 1 stands for perfectly complete assignment.

- **Homogeneity score** [49]: A clustering assignment satisfies homogeneity if all of its clusters contain only data points that are members of a single class. The score is between 0 and 1, where 1 stands for perfectly homogeneous assignment.
- **Purity score**: Each cluster is assigned to the class that is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned samples and dividing by number of samples N. Cluster purity measures how pure clusters are. If a cluster is composed of members of the same class, then it is completely pure.
- **Fowlkes-Mallows index (FMI)** [22]: Measures the similarity of two clusterings by computing the geometric mean between the precision and recall. A higher score indicates a good similarity between two clusters.
- **Silhouette score** [50]: The Silhouette score is calculated using (a) the mean intra-cluster distance and (b) the mean nearest-cluster distance for each sample. The Silhouette Coefficient for a sample is $\frac{(b-a)}{\max(a,b)}$.
- **Calinski-Harabasz score (CHS)** [6]: Taking into account the data samples and the PLs (regardless of the original labels), CHS is defined as the ratio of the sum of between-cluster dispersion and of within-cluster dispersion. It is commonly used to compare assignments of different methods and numbers of clusters. The higher the value, the better is the assignment.
- **Davies-Bouldin score (DBS)** [11]: The average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters that are farther apart and less dispersed will result in a better score. Lower values indicate better clustering.

C Study of various maximum margin-based softmax loss objectives

In order to improve performance on previously unseen data and to generalize to out-of-domain speech samples, in this section we study various maximum margin-based softmax variants based on different objectives. Indeed, softmax suffers from several drawbacks such as that (1) its computation of inter-class margin is intractable [18] and (2) the learned projections are not guaranteed equi-spaced. Indeed, the projection vectors for majority classes occupy more angular space compared to minority classes [39]. To solve these problems, several alternatives to softmax have been proposed [14, 57, 64, 35, 58]. For instance, AM Softmax loss applies an additive margin constraint in the angular space to the softmax loss for maximizing inter-class variance and minimizing intra-class variance. To provide a clear geometric interpretation of data samples and enhance the discriminative power of deep models, AAMSoftmax (angular additive margin softmax) objective (aka ArcFace) introduces an additive angular margin to the target angle (between the given features and the target center). Due to the exact correspondence between the angle and arc in the normalized hypersphere, AAMSoftmax can directly optimize the geodesic distance margin, thus its other name ArcFace. Additionally, CosFace (large margin cosine loss) reformulates the softmax loss as a cosine loss by L2 normalizing both features and weight vectors to remove radial variations, based on which a cosine margin term is introduced to further maximize the decision margin in the angular space. On the other hand, OCSoftmax uses one-class learning instead of multi-class classification and does not assume the same distribution for all classes/speakers. More recently, AdaFace loss has been proposed which emphasizes misclassified samples according to the quality of speaker embeddings (via feature norms).

Table 3 summarizes our results using different predefined numbers of clusters and different clustering-based pseudo-labels.

Our experimental results show clearly that our adopted softmax variants are very effective in improving the generalization of our speaker verification systems. In particular, unlike the widely used AAMSoftmax loss in speaker verification, to our knowledge, our results indicate for the first time that variants such as OCSoftmax (does not assume the same distribution for all speakers which is more realistic in our case) or the recent AdaFace loss, perform consistently better across all pseudo-labels and the ground truth labels. Indeed, AAMSoftmax is susceptible to massive label noise [14]. This is because if a training sample is a noisy sample, it does not belong to the corresponding positive class. In AAMSoftmax, this noisy sample generates a large wrong loss value, which impairs the model training. This partially explains the underperformance of AAMSoftmax compared to other variants when using pseudo-labels for training.

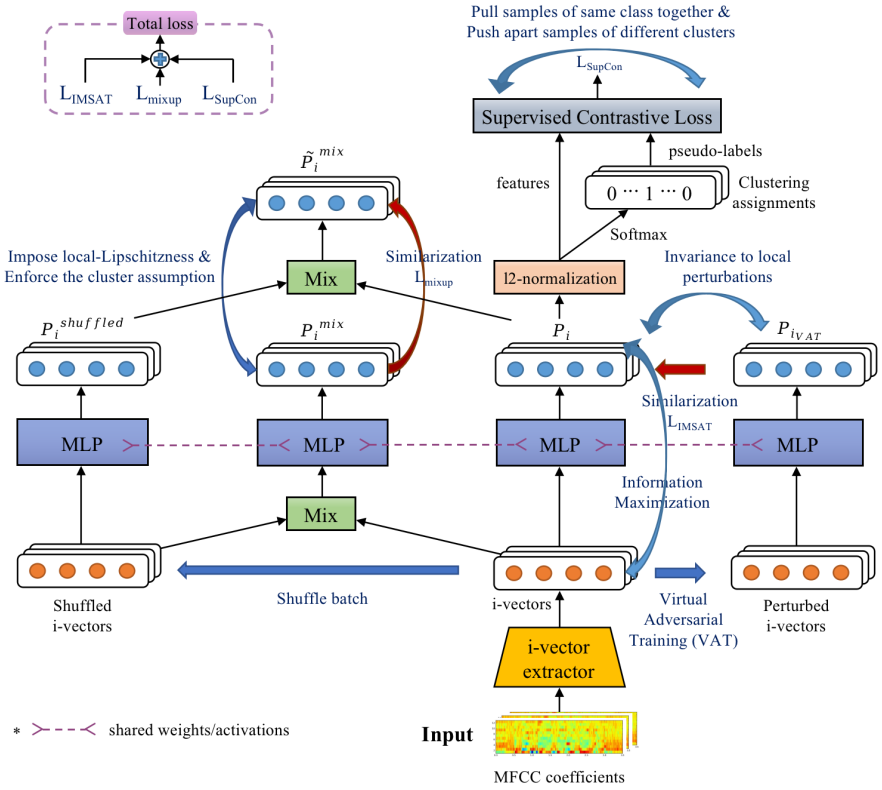

Figure 1: The pipeline of our proposed clustering method depicting the data flow and the different losses employed for clustering.

D System description

D.1 Our proposed clustering method

Figure 1 depicts the framework of our proposed clustering method based on the combination of the proposed Self-Supervised Learning-based objectives that do not employ explicit or domain-specific data augmentation.

D.2 Our clustering-based self-supervised speaker embedding framework

Figure 2 shows a schematic diagram of our general clustering-based self-supervised speaker verification process that we follow throughout the paper. During our work, we try to explore various

Table 3: A study of various margin-based softmax losses for better generalization of our speaker verification system, using different pseudo-labels.

Pseudo-labels	**No. of clusters**	**AAMSoftmax** [14]	**AMSoftmax** [57]	**OCSoftmax** [64]	**AdaFace** [35]	**CosFace** [58]	**Softmax**
True labels	5,994	1.437	1.522	1.416	**1.326**	1.463	3.489
GMM	5,000	5.429	4.851	**4.682**	5.095	4.862	8.425
AHC	5,000	3.621	3.664	3.584	**3.526**	3.6	6.479
IMSAT	5,000	4.507	4.141	3.881	**3.807**	4.083	7.206
	5,994	4.146	3.961	3.892	4.008	**3.696**	7.333
	10,000	4.438	4.024	**4.003**	4.046	4.072	7.370
$L_{IMSAT} + L_{SupCon}$	5,000	4.623	4.401	**4.396**	4.576	4.48	7.179
	5,994	4.475	4.502	4.491	4.443	**4.427**	7.179
	10,000	4.348	4.221	4.189	4.343	**4.173**	7.174
$L_{IMSAT} + L_{Mixup} + L_{SupCon}$	5,000	4.231	4.046	**3.924**	4.056	4.332	7.391
	5,994	4.321	4.146	**4.024**	4.125	4.199	7.28
	10,000	4.252	**4.03**	4.146	4.21	4.093	7.259
$L_{IMSAT} + L_{Mixup} + L_{SupCon} + L_{aug}$	10,000	3.377	**3.287**	3.293	3.399	3.298	5.695
$L_{IMSAT} + L_{Mixup} + L_{SupCon} + L_{aug} + L_{InfoNCE}$	10,000	3.362	**3.001**	3.006	3.043	3.181	5.801

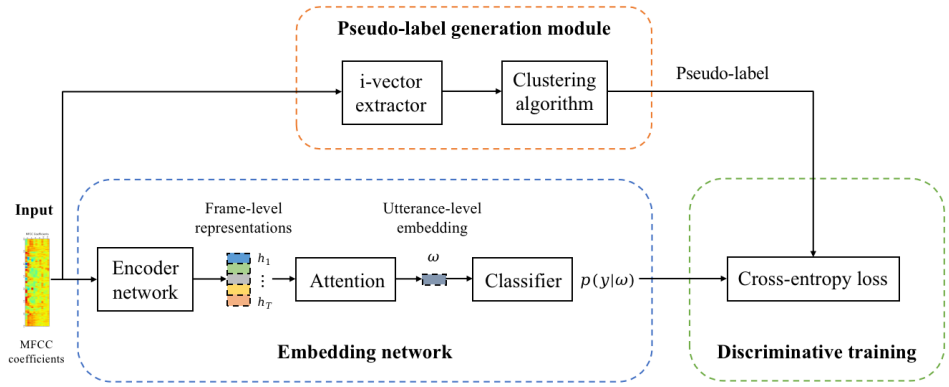

Figure 2: General process for training our clustering generated pseudo-label-based self-supervised speaker embedding networks.

clustering algorithms and conduct different analyses on the impact of pretraining, in particular clustering, on speaker verification performance. We employ ECAPA-TDNN as our speaker embedding network and angular additive margin softmax (AAMSoftmax) objective loss to train our systems using pseudo-labels generated by the various clustering algorithms.

D.3 Clustering-based pseudo-label generation

For clustering, we have extracted i-vector [13, 33] using the Kaldi toolkit [47], which is a statistical unsupervised fixed-dimensional representation from each training utterance and performed clustering on top of them. After training the clustering algorithms, we selected the aligned cluster for each utterance and used the cluster-id as pseudo-label. With the clustering-based pseudo-labels, we can train the speaker embedding network via softmax-based objectives, analogous to supervised learning.

For all of our clustering benchmarks, we have set the number of clusters to be 5000 which [32] found to lead to the best results (except self-organizing maps (SOM) where number of clusters was set to be the size of the map 71*71=5041).

D.4 Input features and datasets

As input to all of our clustering algorithms, we employ 400-dim i-vectors. The compact i-vectors, which are unsupervised speaker representations, allow us here to perform clustering in a more efficient way and to avoid high dimensionality of the MFCC acoustic features.

In order to evaluate the performance of our proposed clustering approach and the generated PLs for self-supervised speaker verification, we conducted a set of experiments based on the VoxCeleb2 dataset [9]. To train the embedding networks, we used the development subset of the VoxCeleb2 dataset, consisting of 1,092,009 utterances collected from 5,994 speakers. The evaluation was performed according to the original VoxCeleb1 trials list [43], which consists of 37,720 trials of 4,874 utterances spoken by 40 speakers.

For our ECAPA-TDNN-based SV system, the acoustic features used in the experiments were 40-dimensional Mel-frequency cepstral coefficients (MFCCs) extracted at every 10 ms, using a 25 ms Hamming window via Kaldi toolkit [47]. Moreover, to follow other SV works in training the ECAPA-TDNN-based systems, we have used waveform-level data augmentations including additive noise and room impulse response (RIR) simulation [51]. In addition to the waveform-level augmentations, we have also applied augmentation over the extracted MFCCs feature, analogous to the specaugment scheme [45].

D.5 Clustering models and training details

To improve generalization, we also use additive angular margin softmax (AAMSoftmax) objective [14] to train our self-supervised speaker embedding network (with scale factor $s = 30$ and angular

margin $m = 0.1$). Cosine similarity was used as a backend for verification scoring between enrollment and test embeddings.

Following IMSAT setup, we use the same MLP-based d-S-S-C architecture, where $d = 400$ and C are input and output dimensionality, respectively. $S = 20800$ neurons is the width of the network. We use RELU for all the hidden activations, apply batch normalization to hidden layers, and use softmax in the output layer. Regarding optimization, we use the Momentum algorithm with an initial learning rate of 0.01, a momentum of 0.9, and an exponential rate decay of 0.996. $\lambda = 0.5$, $\mu = 3.5$. We use a batch size of 10240 i-vectors, and inputs are normalized independently along the samples axis to unit l2-norm to avoid losing speaker information. We use alpha=1 as the coefficient of the Beta distribution used for mixup interpolation. We ran experiments for 150 epochs using 64 CPU cores for each clustering algorithm. Besides, all speaker verification experiments have been run for 7 days using a single RTX2080Ti GPU, with a batch size of 200 MFCC samples. All code and methods in our experiments are based on Tensorflow.

E Mathematical details of our equations

In this section, we provide additional mathematical details of our proposed method above. Indeed, inspired from the Regularized Information Maximization method [36], and based on Self-Augmented Training (SAT) regularization, $R_{SAT}(\theta; T) = \frac{1}{N}\sum_{n=1}^{N} R_{SAT}(\theta; x_n, T(x_n))$ is a loss term that allows the representations of the augmented samples to be further pushed close to those of the original samples while also regularizing the complexity of the network against local perturbations using Virtual Adversarial Training (VAT) [42]. $R_{SAT}(\theta; x, T(x)) = -\sum_{c=1}^{C}\sum_{y_c=0}^{1} p_{\hat{\theta}}(y_c|x) log p_\theta(y_c|T(x))$. Where $p_{\hat{\theta}}(y_c|x)$ is the prediction of sample x, and $\hat{\theta}$ the current parameters of the network. $T_{VAT}(x) = x + r$ is the augmentation function using local perturbations to enforce invariance where $r = \arg\max_{r'}\{R_{SAT}(\hat{\theta}; x, x + r'); \|r'\|_2 \leq \epsilon\}$ is an adversarial direction. $H(.)$ and $H(.|.)$ are the marginal and conditional entropy, respectively, and their difference represents the MI between input X and label Y that we maximize. $H(Y) = h(p_\theta(y)) = h(\frac{1}{N}\sum_{i=1}^{N} p_\theta(y|x))$, and $H(Y|X) = \frac{1}{N}\sum_{i=1}^{N} h(p_\theta(y|x_i))$, where $p_\theta(y|x)$ is our probabilistic classifier modeled by parameters θ of a deep network, and $h(p(y)) = -\sum_{y'} p(y')\log p(y')$ is the entropy function. Hyperparameters $\lambda, \mu \in \mathbb{R}$ control the trade-offs between the complexity regularization of the model (through R_{SAT}) and the MI maximization, and between the two entropy terms, respectively. Basically, increasing the entropy H(Y) amounts to encouraging the cluster sizes to be uniform and prevent collapsing into degenerate solutions, while minimizing the conditional entropy $H(Y|X)$ enables less ambiguous cluster assignments and forces the classifier to be confident on the training samples [5]. During our experiments, we find the L_{IMSAT} loss to be critical for good clustering performance. For more details, please refer to [42, 29].

Moreover, as the L_{SupCon} loss requires labels, the novelty of our usage is to use online predictions of our model as input labels, which allows us to use it in a completely unsupervised fashion without the need for ground-truth labels. As the performance of our clustering gradually improves, the online PLs are progressively more reliable, thus helping to generate better and more compact clusters.

Finally, inspired from VMT [40] regularization method which encourages the model to behave linearly in-between training points, this allows us to enforce representation smoothness during clustering and enforce consistent predictions between the surrounding and training points. Indeed, mixup [62] which is a strategy to augment data by interpolating different data samples alongside their labels, often leads to better generalization to out-of-set samples. Mixup was also found by [21] to lead to better generalization of self-supervised speaker verification systems when the clusters are not compact or not well distanced as it can dilute label noise and induce better class separation. Following the work of [40], instead of directly mixing probabilities in the L_{Mixup} loss, we perform mixup over logits, followed by softmax for better training and to prevent early information loss during the mix of probabilities. During experiments, we found this to considerably improve results and convergence compared to mixup on probabilities. We follow the general training framework depicted in Figure 2. Code of our clustering framework is available at `https://github.com/fathana/CAMSAT_clustering`.

F Details of our augmentation-based SSL objectives

The following list provides the description details of the augmentation-based SSL objectives used in our experiments:

- **Data augmentation loss** L_{aug}: L_{aug} forces the predicted representations of augmented samples to be close to those of the original data points by minimizing the KL-divergence between both predictions, as follows:

$$L_{aug} = \frac{1}{N} \sum_{i=1}^{N} KL(p_i^{aug_{r_i}} || p_i) \tag{4}$$

  with $J = \{aug_1, ..., aug_{|J|}\}$ is the ensemble of available data augmentations and $r_i \in \{1, .., |J|\}$ refers to a random augmentation from J. $KL(.||.)$ refers to the Kullback-Leibler divergence, and N is the size of data (or mini-batches). $p_i = f(x_i) \in \mathbb{R}^{1xC}$, and $p_i^{aug_j} = f(x_i^{aug_j})$ correspond to the predictions of data sample x_i and its augmented version $x_i^{aug_j}$, respectively.

- **Contrastive Self-Supervised Learning (InfoNCE)**: InfoNCE [44], where NCE stands for Noise-Contrastive Estimation, is a type of contrastive loss function used for self-supervised learning in SimCLR [7], also known as the NT-Xent loss (Normalized Temperature-scaled Cross-Entropy). The goal is to maximize the similarity between the representations of two augmented versions of the same input, i.e., Z_i and Z_j while minimizing it to all other examples in the batch.

  In short, the InfoNCE loss compares the similarity of Z_i and Z_j to the similarity of Z_i to any other representation in the batch by performing a softmax over the similarity values. The InfoNCE loss $l_{i,j}$ for pair (i,j) can be written as follows:

  $l_{i,j} = -log \frac{\exp sim(Z_i, Z_j)/\tau}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp sim(Z_i, Z_k)/\tau}$.

  $\mathbb{1}_{k \neq i} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$, and $\tau = 1$ denotes the temperature parameter. The final $L_{InfoNCE}$ loss is computed across all positive pairs, both (i, j) and (j, i), in a mini-batch (a sample and its augmented version). The default similarity metric that is used is cosine similarity, defined as: $sim(Z_i, Z_j) = \frac{{Z_i}^T . Z_j}{\|Z_i\| \|Z_j\|}$. We also studied KL-divergence as a distance metric in the appendix but the results were worse than cosine distance.

- **Supervised Contrastive Loss (SupCon)**:

  L_{SupCon} from [34] extends the self-supervised batch contrastive approach of the NT-Xent loss (Normalized Temperature-scaled Cross Entropy) [7] to the fully-supervised setting, allowing us to effectively leverage label information. For that, clusters of points belonging to the same class are pulled together in normalized embedding space, while simultaneously pushing apart clusters of samples from different classes. The SupCon extension allows for multiple positives per anchor instead of a single sample in addition to many negatives, and draws from samples of the same class as the anchor, rather than being data augmentations of the anchor, as done in previous works. It showed benefits for robustness to natural corruptions and is more stable to hyperparameter settings such as optimizers and data augmentations.

  Since the SupCon loss requires labels, the novelty of our usage is to use online generated labels as input labels to the SupCon loss function, which allows us to use it in a completely unsupervised fashion without the need for ground-truth labels. Additionally, our framework does not rely on any additional modules such as a projection network or a separate encoder. We use the implementation from `https://github.com/wangz10/contrastive_loss/blob/master/losses.py` with temperature=1 and base_temperature=1.

- **Variance-Invariance-Covariance Regularization (VICReg)**: VICReg [4] aims to maximize the agreement between representations of augmented views of the same instance while preventing the collapse problem. It uses two regularization terms (1) a term $L_{VICReg'variance}$ that maintains the variance of each embedding dimension above a threshold, (2) a term $L_{VICReg'covariance}$ that decorrelates each pair of variables. VICReg loss is

composed of a variance, invariance and covariance loss terms that are added to each other as follows: $L_{VICReg} = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')]$

The variance regularization term is: $v(Z) = \frac{1}{C}\sum_{j=1}^{C} \max(0, \gamma - S(Z^j, \epsilon))$ where $S(x, \epsilon) = \sqrt{Var(x) + \epsilon}$ and $\gamma = 1$ is a constant target value for the standard deviation. Z^j denotes the vector composed of all values at dimension j in all logit vectors in batch matrix Z and $\epsilon = 0.0001$ is a small scaler for stability. The covariance regularization term is computed as follows: $c(Z) = \frac{1}{C}\sum_{i \neq j}[C(Z)]_{i,j}^2$. With $C(Z) = \frac{1}{N-1}\sum_{i=1}^{N}(Z_i - \overline{Z})(Z_i - \overline{Z})^T$ and $\overline{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$. Z_i denotes the logits of sample i. The invariance criterion is simply: $s(Z, Z') = \frac{1}{N}\sum_i \|Z_i - Z_i^{'}\|_2^2$

In our implementation, we use $\lambda = 1$, $\mu = 1$, and $\nu = 0.5$. Additionally, Since the covariance matrix requires large memory usage, we set the batch size to 640 samples and use $C = 5000$ clusters.

G Discussion of our multi-objective clustering approach

Our approach in this paper tries to extend/improve the IMSAT method by incorporating and studying in a multi-objective fashion additional SSL objectives to the current available framework. Our study tries to investigate useful self-supervised objective losses for the purpose of speaker clustering and recognition. Our aim is to harness these objectives as additional supervisory signals during clustering to regularize the clustering model to produce consistent feature representations. Besides, this can increase the model's expressiveness via various inductive biases, maximize the amount of information learned per sample, and help it learn weights that can better disambiguate hard/complex data examples, be able to self-correct its early mislabelling, and reduce the likelihood of learning spurious features since in that case the weights are constrained to simultaneously satisfy all the training objectives (i.e. assume the simplest hypothesis). Our work also analyzes the complementary information between these objectives using our large MLP-based architecture, without interference from other architectural biases.

The IMSAT framework, which is the backbone of our clustering approach helps to avoid degenerate solutions, that other clustering methods are susceptible to, by been rigorously grounded in information theory. Indeed, due to the entropy maximisation component within MI, the loss objective is not minimised if all inputs are assigned to the same class. At the same time, it is optimal for the model to predict for each input a single class with certainty (i.e. one-hot) due to the additional conditional entropy component that we minimize. Hence, we avoid clusters disappearing during training or a single cluster starts dominating the predictions. During our experiments, we find the L_{IMSAT} loss to be critical for good clustering performance.

Inspired from VMT [40] regularization method which encourages the model to behave linearly in-between training points, this allows us to enforce representation smoothness during clustering and enforce consistent predictions between the surrounding and training points [56]. Indeed, mixup [62] which is an efficient strategy to augment data by interpolating different data samples alongside their labels, often leads to better generalization to out-of-set samples. It has proven its strength in various tasks (e.g., image classification [62], anti-spoofing [53] and speech recognition [41]). [62] has shown that mixup not only reduces the memorization to adversarial samples, but also performs better than Empirical Risk Minimization [55]. Mixup has also been experimentally found by [21] to be effective against label noise memorization [1], and to lead to better generalization of self-supervised speaker verification systems when the clusters are not compact or not well distanced. We find this property to be especially important during clustering of speaker embeddings to mitigate the strong label noise at the first training epochs and avoid early convergence to suboptimal cluster assignments. As Mixup can dilute the label noise in online generated pseudo-labels and create synthetic samples around the borders that lead to smoothing the data manifold and better class separation, we believe this can help slow down the memorization of noisy pseudo-labels and learn long enough from the simple patterns available to lead to better clusters and induce robustness, better generalization capability, and better online clustering stability for large-scale datasets.

The resulting algorithm is highly scalable, fast, more robust than IMSAT to corruptions and shifts in the data during online clustering, is simple to implement, and adds limited computational overhead to IMSAT. We believe our proposed clustering method can be considerably beneficial to further

optimize current self-supervised SV frameworks by replacing the simple clustering methods being employed (e.g. k-means, spectral clustering). It can also be used in speaker diarization frameworks to improve the clustering aspects of speaker diarization methods where clustering is one of the important modules. Finally, our proposed clustering approach is a general-purpose method and can be applied to other problems and domains other than speech.

H Comparison of our approach to other clustering benchmarks

In Table 4, we provide the results for a large variety of clustering benchmarks compared to our proposed method without explicit data augmentation. According to the results, our approach outperforms all other baselines in terms of clustering metrics achieving 63.9% unsupervised accuracy, while having a compute time comparable to classical clustering models (3-4 days). Using our proposed system's generated PLs to train our speaker embedding system, also allowed us to achieve a very competitive downstream SV EER performance outperforming all other benchmarks, except the AHC PLs which lead to a slightly better performance.

Table 4: A comparison study of our proposed clustering method compared to a large set of benchmarks (classical and deep-learning based models). Results are reported in terms of Clustering performance (clustering metrics) and the corresponding EER (%) downstream SV evaluation performance when using the generated pseudo-labels to train our studied speaker verification system. l2Norm refers to normalizing i-vector inputs independently along the samples axis to unit l2-norm instead of mean and standard deviation scaling (StandardScaler) of i-vectors along the features axis.

Model	**Clustering Metrics**											**Speaker Verification**
	ACC	**AMI**	**NMI**	**No. of clusters**	**Completeness**	**Homogeneity**	**FMI**	**Purity**	**Silhouette**	**CHS**	**DBS**	**EER (%)**
Supervised (True Labels)	**1.0**	**1.0**	**1.0**	5994	**1.0**	**1.0**	**1.0**	**1.0**	**−0.006**	31.708	4.692	**1.437**
GMM (Full cov.)	0.45	0.631	0.747	5000	0.767	0.728	0.312	0.566	-0.015	39.266	4.673	5.143
GMM (Full cov., l2Norm)	0.504	0.678	0.789	5000	0.792	0.785	0.415	0.633	-0.015	41.568	5.114	5.429
Bayesian GMM (γ=1e-5, μ=1)	0.45	0.629	0.746	5000	0.766	0.727	0.312	0.566	-0.015	39.257	4.673	5.143
Bayesian GMM (l2Norm, γ=1e-5, μ=1)	0.504	0.678	0.789	5000	0.792	0.785	0.415	0.633	-0.015	**41.57**	5.115	5.159
Divisive HC	0.097	0.204	0.477	5000	0.479	0.474	0.035	0.132	-0.06	18.044	9.068	13.531
KMeans	0.302	0.468	0.591	5000	0.645	0.546	0.194	0.311	-0.114	24.936	**2.714**	6.978
CURE	0.151	0.218	0.393	5000	0.466	0.34	0.011	0.216	-0.052	17.77	5.372	6.994
BIRCH	0.299	0.374	0.54	5000	0.725	0.43	0.013	0.353	-0.027	24.348	4.901	5.642
DEC	0.029	0.122	0.365	4911	0.386	0.345	0.007	0.036	-0.084	8.734	7.266	11.957
SOM	0.025	0.088	0.402	5041	0.404	0.4	0.01	0.037	-0.041	10.148	18.402	15.806
DeepCWRN	0.003	0.006	0.15	1008	0.179	0.129	0.001	0.003	-0.217	3.841	41.521	38.171
IMSAT	0.393	0.491	0.649	4987	0.668	0.63	0.297	0.426	-0.044	22.887	6.668	5.912
AHC	0.587	0.74	0.825	5000	0.841	0.81	0.311	0.684	**−0.01**	39.561	4.991	3.685
AHC (l2Norm)	0.602	0.756	0.838	5000	0.849	0.827	0.375	0.693	-0.034	39.638	5.147	3.621
Our approach	0.639	0.776	0.86	9685	0.847	0.873	0.642	0.71	-0.136	0.998	17.599	4.252
Our approach (with data augmentation & InfoNCE)	**0.725**	**0.842**	**0.9**	8500	**0.89**	**0.91**	**0.746**	**0.792**	-0.134	1.0	18.407	**3.362**

I The evolution of clustering metrics over time

In figure 3, we show the evolution of clustering metrics and the number of clusters discovered during the training process. Results show that regularization through data augmentation helps considerably to improve performance and that using augmentations through objectives L_{aug} or $L_{InfoNCE}$ takes considerably more epochs to achieve the best clustering performance. As incorporating these objectives also consumes more computing resources, this results in the whole training process taking around 10 times longer for training compared to our proposed augmentation-free clustering approach which requires around 3 days to converge to its best performance.

J Self-supervised angular additive margin softmax (AAMSoftmax) objective

The angular additive margin softmax (AAMSoftmax) objective is one of the most popular methods for training a speaker embedding network [14]. The AAMSoftmax objective is formulated as follows:

$$L_{AAMSoftmax} = -\frac{1}{N}\sum_{i=1}^{N} log(\frac{e^{s(cos(\theta_{y_i,i}+m))}}{K_1}), \tag{5}$$

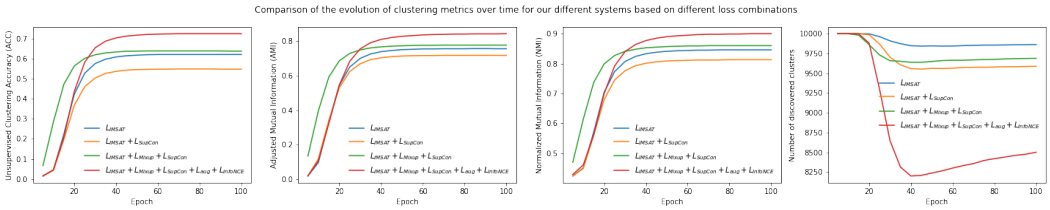

Figure 3: The evolution of clustering metrics over epochs and the number of clusters discovered during training of our clustering systems based on various loss combinations.

where $K_1 = e^{s(cos(\theta_{y_i,i}+m))} + \sum_{j=1,j\neq i}^{c} e^{scos\theta_{j,i}}$, N is the batch size, c is the number of classes, y_i corresponds to label index, $\theta_{j,i}$ represents the angle between the column vector of weight matrix W_j and the i-th embedding ω_i, where both W_j and ω_i are normalized. The scale factor s is used to make sure the gradient is not too small during the training and m is a hyperparameter that encourages the similarity of correct classes to be greater than that of incorrect classes by a margin m.

The training of AAMSoftmax for self-supervised speaker embedding learning is made possible by the use of our generated pseudo-labels as the above objective requires speaker labels for training.

K ECAPA-TDNN Architecture details

Table 5: Standard ECAPA-TDNN architecture. T indicates the duration of features in number of frames and d the feature vector dimensionality and N_c is the number pf classes. Batch normalization is further employed after each layer except temporal pooling.

Layer	***Input Dimension***	***Output dimension***
Conv1d+ReLU+BN	d × T	512 × T
SE-Res2Block	512 × T	512 × T
SE-Res2Block	512 × T	512 × T
SE-Res2Block	512 × T	512 × T
Conv1d+ReLU	512 × T	1536 × T
Attentive Statistics Pooling + BN	1536 × T	3072×1
FC + BN	3072×1	192×1
AAM-Softmax	192×1	N_c×1