Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Clustering-based hard negative sampling for supervised contrastive speaker verification

Piotr Masztalski[1,2], *Michał Romaniuk*[1], *Jakub Żak*[1], *Mateusz Matuszewski*[1], *Konrad Kowalczyk*[2]

[1]Samsung R&D Institute Poland,
[2]AGH University of Krakow, Poland

{p.masztalski,m.romaniuk2,j.zak,m.matuszews2}@samsung.com, konrad.kowalczyk@agh.edu.pl

Abstract

In speaker verification, contrastive learning is gaining popularity as an alternative to the traditionally used classification-based approaches. Contrastive methods can benefit from an effective use of hard negative pairs, which are different-class samples particularly challenging for a verification model due to their similarity. In this paper, we propose CHNS - a clustering-based hard negative sampling method, dedicated for supervised contrastive speaker representation learning. Our approach clusters embeddings of similar speakers, and adjusts batch composition to obtain an optimal ratio of hard and easy negatives during contrastive loss calculation. Experimental evaluation shows that CHNS outperforms a baseline supervised contrastive approach with and without loss-based hard negative sampling, as well as a state-of-the-art classification-based approach to speaker verification by as much as 18 % relative EER and minDCF on the VoxCeleb dataset using two lightweight model architectures.

Index Terms: speaker verification, supervised contrastive learning, hard negative sampling, clustering

1. Introduction

Speaker verification (SV) is a task that aims to verify the identity of a speaker based on voice characteristics. In recent years, researchers have been improving the performance of SV systems using various deep learning approaches with great success [1, 2, 3, 4, 5]. The standard method involves learning a fixed-size speaker embedding, which is used as a reference during pairwise comparisons in the evaluation process. The majority of current state-of-the-art systems are trained in a classification-based fashion, where during inference, the classification layer is discarded, and embeddings from the encoder are used for scoring. These methods are often trained with the Additive Angular Margin Softmax loss function (AAMSoftmax) introduced initially for face verification [6]. While the classification-based approach is the most prevalent in the literature, there also exist multiple works that show the benefit of using contrastive learning or metric learning for SV, some of which report similar or better performance than the classification-based counterpart [7, 8, 9]. The contrastive approach to speaker verification has been explored primarily in self-supervised learning (SSL) [10, 11, 12, 13, 14]. Most SSL approaches are based on the SimCLR [15] framework adapted from image representation learning, which maximizes agreement between differently augmented views of the same sample (positive pairs), while at the same time maximizes the distance between different samples (negative pairs). The training objective is achieved using the normalized temperature-scaled cross-entropy (NTXent) loss, which contrasts the similarity of a positive pair against the similarities between the anchor and negative examples in a batch. Due to the lack of class labels in SSL-based SV, positive pairs are typically constructed by augmenting the same audio utterance, which can result in the model learning channel characteristics (e.g. recording device, acoustic environment) instead of voice-dependent features [10]. Furthermore, a necessary assumption for SSL is that one always obtains segments of two different speakers when randomly sampling two audio utterances, which is not always true and can result in injecting false negatives into the loss calculation.

Supervised contrastive learning [16] is a natural extension of SimCLR to labeled datasets, and has also been used in the context of speaker verification, however, to a much lesser extent [7, 8, 17]. In this approach, class labels are used to obtain diverse positive pairs that originate from different utterances of the same speaker, which forces the model to focus strictly on speaker-dependent audio features. Negative pairs are also selected based on speaker labels, mitigating the chance for any false negatives. Supervised contrastive learning utilizes the SupCon [16] loss function, which is analogous to the NTXent loss with a simple modification to take class labels into account.

The contrastive learning approach seems to be intuitive for SV since it directly replicates the inference process during training, with pairwise embedding comparisons using cosine similarity. We also note an additional benefit of not requiring a classification layer, which can have a great impact on the parameter count of the model during training, when dealing with datasets containing a very large number of speakers.

It has been shown in [18, 19, 20, 17] that contrastive learning can benefit from emphasizing hard negative samples. A hard negative is defined as a data point that belongs to a different class than an anchor sample, while at the same time producing a high similarity score. The standard approach to hard negative sampling for contrastive learning relies on incorporating a hardening function into the loss, so that negatives with higher similarity to the anchor sample have a larger influence on model updates than easy negatives. In supervised contrastive learning, this loss-based hard negative sampling method has been introduced in [19] as H-SCL.

In SV, hard negative sample pairs are utterances from different speakers with similar voice traits. A real-life example of that are speech samples from members of the same family [21]. This scenario is commonly found in a smart home setting, where a voice assistant should have a personalized approach based on the speaker identity, which makes it crucial for SV systems to be able to differentiate between similar speakers. The smart home setting also requires SV models to run efficiently on edge devices like smart vacuum cleaners, fridges or air conditioning units. Motivated by this fact, in our experiments, we consider lightweight model architectures suitable for real-time deployment on edge devices with limited compute capabilites.

10.21437/Interspeech.2025-442

Recent works focusing on contrastive and non-contrastive self-supervised speaker verification have shown that a clustering approach can be used to discover pseudo-labels in the training dataset [12, 13, 22, 23, 24]. After a pre-training phase, the model is used to compute embeddings for training utterances, and cluster these embeddings into groups of similar samples. These groups are used to form positive pairs originating from different audio utterances, which results in improved SV performance compared to only augmenting the same segment.

While there is no need to use pseudo-labels in a supervised setting, in this paper we adapt the clustering method to find hard negative pairs and use them in the training process. Building on that idea, we introduce CHNS (**c**lustering-based **h**ard **n**egative **s**ampling), a hard negative sampling approach for supervised contrastive speaker representation learning. Contrary to H-SCL, we do not alter the loss function, and instead focus solely on the batch composition. We scan our training dataset for speakers with similar voice characteristics and cluster them into distinct groups. Our algorithm leverages these clusters to tune the ratio of hard negative samples inside each batch, and as a result improve performance of SV models.

2. Proposed Method

Our proposed solution uses supervised contrastive learning to train a model that produces utterance-level speaker representations. We use a contrastive loss function during the training process, which calculates the relationship between speaker representations on a within-batch basis. This means that batch composition has a significant impact on the training process. In this section, we describe our batch sampling approach in detail.

2.1. Clustering-based hard negative sampling

In contrast to a standard way of selecting hard negative samples on an utterance level, in CHNS, we discover hard negatives on the speaker (class) level. For each speaker in the training dataset, we sample 10 utterances, calculate an embedding for each utterance using a pre-trained SV model, and then calculate a centroid from these 10 embeddings. We use a baseline supervised contrastive model trained with random sampling to obtain the initial embeddings. This process is analogous to speaker enrollment in a real-life SV system and provides an average voice representation for each speaker. We will refer to these speaker embedding centroids as voiceprints.

Having obtained a voiceprint for each speaker in the training dataset, we run the K-Means clustering algorithm on them to create disjoint sets of similar speakers. We select K-Means specifically, because it uses squared Euclidean distance as a similarity measure. There is a linear relationship between squared Euclidean distance and cosine similarity when operating on normalized vectors [25], and during inference, the calculated embeddings are compared using cosine similarity.

We assume that if there is a similarity between speaker voiceprints in each cluster, all of the pairs created from utterances that belong to different within-cluster speakers can be considered hard negatives. We argue that running the clustering algorithm on the voiceprint level is superior to clustering individual utterance embeddings, as it takes much less time to calculate and memory to store the results.

For the training process, we implement a custom batch sampler, which is designed to use the pre-calculated clusters for batch composition. First, we define a parameter called *hard ratio*, which specifies what proportion of the batch is composed of

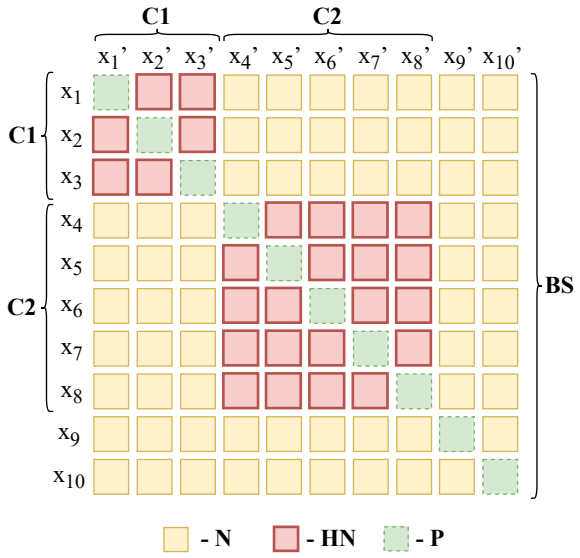

Figure 1: *Proposed within-batch similarity matrix with hard ratio set to 0.8. C1, C2 - cluster sizes, BS - batch size, N - negative, HN - hard negative, P - positive, (x_n, x_n') - same speaker utterance pair.*

speakers belonging to specific clusters. We start with an empty batch, and sample the first speaker cluster randomly. If a sampled cluster contains less than *hard ratio* $*$ *batch size* speakers, another random cluster is sampled and its speakers are added to the batch. This process is repeated until we satisfy the *hard ratio* condition. The remainder of the batch is sampled randomly from the rest of the training speakers. We sample two utterances per speaker to create a positive pair for each identity, and the remaining utterances in each batch form negative pairs. Using our algorithm, during the loss calculation we obtain a similarity matrix as depicted in Figure 1. The intra-cluster utterance pairs from different speakers form hard negatives, and the rest of the negative pairs are of random hardness.

The number of clusters we choose for the K-Means algorithm will determine the average cluster size and is dependent on the number of speakers in the dataset. The larger the number of clusters, the smaller number of speakers belong to each cluster and the average distance of within-cluster speakers gets closer. This means that the number of clusters together with the value of *hard ratio* plays a vital role in the final batch composition, and tuning these parameters to the training dataset is instrumental in obtaining optimal results.

2.2. Loss function

The objective function in our contrastive experiments is given by:

$$\mathcal{L} = \sum_{i=1}^{K} \frac{-1}{|P_i|} \sum_{x^+ \in P_i} \log \frac{e^{s(x_i,x^+)/\tau}}{e^{s(x_i,x^+)/\tau} + \sum\limits_{x^- \in N_i} H e^{s(x_i,x^-)/\tau}} \tag{1}$$

where x_i is the anchor sample, P_i and N_i are sets of positive and negative samples x^+ and x^- corresponding to the anchor sample, $s(x, y)$ is the cosine similarity function, H is the hardening function, τ is the temperature parameter and K is the batch size. Following H-SCL [19], we employ the exponential hardening function given by $H = e^{\beta s(x,y)}$, where β is a hyperparameter. For $\beta = 0$ the loss function is equivalent to the

Table 1: *Comparison between the number of clusters (groups of similar speakers) and different hard ratio values on VoxCeleb1-H.*

Hard ratio	**10 clusters**		**20 clusters**		**50 clusters**		**100 clusters**	
	EER[%]	**minDCF**	**EER[%]**	**minDCF**	**EER[%]**	**minDCF**	**EER[%]**	**minDCF**
0.2	3.04	0.1839	2.90	0.1807	3.02	0.1831	3.15	0.1903
0.5	3.01	0.1866	2.92	0.1844	2.86	0.1753	2.91	0.1797
0.8	2.92	0.1822	2.86	0.1777	2.79	0.1721	2.88	0.1745
1	2.85	0.1754	2.89	0.1801	**2.70**	**0.1635**	2.83	0.1743

SupCon loss from [16] without any weights to the negatives in the batch. When using one positive pair per speaker and $\beta = 0$, $\mathcal{L}$ is also equivalent to the N-pair loss [26].

The SupCon loss is the baseline comparison for both CHNS and H-SCL approaches to hard negative sampling. We also test the compatibility of our CHNS method with the additional weighing of negatives that is present in the H-SCL loss. The experimental results and findings are presented in Sec. 4.

3. Experimental setup and datasets

We conduct a comprehensive experimental evaluation to find the best parameters for the proposed CHNS algorithm, and compare our solution to existing training methods: a supervised contrastive approach with random batch sampling (SupCon), a supervised contrastive approach with loss-based hard negative sampling (H-SCL), and an industry standard classification-based approach with the AAMSoftmax loss function. For a fair comparison, in all experiments, we use the exact same model, data and training parameters. We do not use additional data augmentation or score normalization.

3.1. Models

In the majority of performed experiments, we use the ECAPA-TDNN architecture [4] with a reduced number of convolution channels (256 channels which yields 2 M model parameters). We decrease the number of parameters to satisfy our aim of obtaining a model suitable for deployment on edge devices. For the classification-based training, we add one linear classification layer on top, which will be discarded during inference. For the contrastive methods, we decide not to use a non-linear projection head, which is contrary to SimCLR [15], and instead calculate the contrastive loss directly on the outputs of the encoder. Omitting the projection head seems intuitive in SV, since we do not fine-tune the model on a different downstream task as done in SimCLR. We have also found this approach to provide the best results. In order to test the transferability of our training method to other architectures, we additionally perform experiments using a 1.4 M parameter Thin ResNet-34 implementation from [9].

3.2. Data

We use the VoxCeleb2 [27] development dataset to train the models. The dataset consists of over 1 M multilingual utterances from 5994 speakers. We extract a 100-speaker subset of the training dataset for validation. The training segments are cropped to 3 seconds at a 16 kHz sampling rate and converted to mel spectrograms. For testing purposes, we use VoxCeleb1-H - a challenging split from the VoxCeleb1 dataset, consisting only of speaker pairs of the same gender and nationality. In addition, we use the CNCeleb(E) [28] evaluation dataset to test the cross-dataset performance of the models trained on VoxCeleb. We do not finetune the models on the CNCeleb training dataset.

As part of one experiment, we train the models on an internal dataset consisting of 1 s utterances of the "*Bixby*" phrase from over 120 k diverse speakers. We evaluate the performance of these models on an in-house evaluation dataset to which we will refer as Bixby Eval.

For evaluation we use a standard set of SV metrics, namely Equal Error Rate (EER) and minimum Detection Cost Fuction (minDCF) with $c_{miss} = c_{fa} = 1$ and $p_{target} = 0.05$.

3.3. Training setup

For the classification-based training, we use the AAMSoftmax loss with margin and scale parameters set to 0.2 and 30 respectively. In case of contrastive experiments we use loss from (1) with the τ parameter set as trainable starting with a value of 0.1. When employing H-SCL we set the β parameter to 0.1. We arrive at this value experimentally, as it balances training stability and model performance. For contrastive training without CHNS, we require that each batch contains only one positive pair per speaker, as we find this approach produces the best results. For all experiments, we use the Adam optimizer, and a cosine decay learning rate schedule with warm-up with maximum learning rate set to 0.01. Every training is run on a single NVIDIA A100 GPU for 200 epochs with a batch size of 1300. We share the code for reproducing our experiments online[1].

4. Experiments and results

In the first experiment we establish the best parameters for CHNS. The two variables in the algorithm are the number of clusters (which also correlates with the average cluster size) and *hard ratio*. We test 4 cluster numbers of 10, 20, 50 and 100, and 4 *hard ratio* values: 0.2, 0.5, 0.8 and 1. We compare results on the VoxCeleb1-H evaluation dataset to choose the best parameter combination. The results of these comparisons are presented in Table 1, which shows that the best performance is obtained when setting the number of clusters to 50 and *hard ratio* to 1. In case of 50 clusters, the average cluster size is around 117 speakers, since there are 5894 speakers in our training dataset. This results in around 5.5 clusters in each batch when using a batch size of 1300 with *hard ratio* = 1 (650 unique speakers per batch as there are 2 samples per each speaker). With a lower number of clusters, the number of speakers per cluster rises, and a bigger overall number of hard negatives are inserted into each batch. However, this comes at a cost of lower diversity within the batch, and the hard negatives are easier, on average, than in case of smaller clusters. The 50-cluster variant with *hard ratio* of 1 setting seems to strike the best balance between batch "hardness" and speaker variability in case of the VoxCeleb2 dataset.

In the second experiment, we take the previously selected best model trained with CHNS (denoted as SupCon + CHNS) and compare it to the classification-based model (AAMSoft-

[1] https://github.com/masztalskipiotr/chns

Table 2: *Comparison between AAMSoftmax, SupCon, H-SCL and CHNS on VoxCeleb1-H, CNCeleb(E) and Bixby Eval.*

Training strategy	VoxCeleb1-H		CNCeleb(E)		Bixby Eval	
	EER[%]	minDCF	EER[%]	minDCF	EER[%]	minDCF
AAMSoftmax	3.19	0.1912	21.88	0.6784	1.99	0.1226
SupCon	3.17	0.1919	21.72	0.7076	1.82	0.1152
H-SCL [19]	3.02	0.1817	20.07	0.6903	1.68	0.1123
SupCon + CHNS (ours)	2.70	0.1635	16.61	0.5969	1.59	0.1066
H-SCL + CHNS (ours)	**2.60**	**0.1569**	**16.26**	**0.5874**	**1.33**	**0.0947**

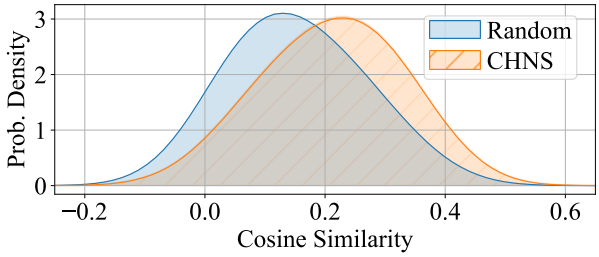

Figure 2: *Average distribution of negative pair similarities in the training batch depending on the batch sampling method.*

max), a supervised contrastive model with random sampling (SupCon), and a supervised contrastive model with loss-based hard negative sampling (H-SCL) [19]. To push the limits of hard negative sampling, as a last solution we also combine the H-SCL method with CHNS (denoted as H-SCL + CHNS). Table 2 presents the results of this experiment, which clearly indicate the usefulness of CHNS compared to the state-of-the-art on the three datasets. The comparison between default contrastive methods such as SupCon and H-SCL, and their corresponding CHNS versions (the proposed approach), shows that by changing only the batch composition method one can achieve significant performance improvements even though the training framework stays exactly the same. Additionally, when directly comparing the two hard negative sampling approaches, we observe that SupCon + CHNS significantly outperforms H-SCL, where the only difference between the two is the way of utilizing hard negative pairs. The advantage of CHNS is that it does actual *sampling* of hard negatives instead of only assigning higher importance to some negative samples that are already in the batch, and therefore the average distribution of negatives within the batch skews more towards larger values compared to random sampling used in all competing methods (as shown in Figure 2). It is also worth noting that both contrastive methods which employ any form of hard negative sampling, usually outperform the classification-based AAMSoftmax model.

Looking at the results on CNCeleb(E) we can see a similar trend, as on VoxCeleb1-H, which indicates that the benefit of our method is transferable to other datasets. The larger absolute values of EER and minDCF are consistent with literature and are resulting from the lack of fine-tuning and the overall challenging nature of the CNCeleb dataset.

When training and evaluating on our internal dataset, the HSCL + CHNS combination yields a relative improvement of 33 % in EER and 23 % in minDCF over the classification-based approach. We also note a meaningful advantage of using contrastive learning when training on datasets with a very large number of classes which is the lack of a classification layer. When training on our internal dataset using the AAMSoftmax loss, the classification layer alone contains over 23 M parameters which is more than 10 times the size of the applied ECAPA-TDNN encoder, and leads to much slower training times.

Table 3: *CHNS initialization step vs. results on VoxCeleb1-H.*

Curriculum	EER[%]	minDCF
From start	**2.70**	**0.1635**
After 5 % of training	2.88	0.1746
After 50 % of training	2.94	0.1833

Table 4: *Comparison between training strategies with Thin ResNet-34 on VoxCeleb1-H.*

Training strategy	EER[%]	minDCF
AAMSoftmax	3.17	0.1882
SupCon	3.42	0.2107
H-SCL [19]	3.25	0.2006
SupCon + CHNS (ours)	2.94	0.1845
H-SCL + CHNS (ours)	**2.88**	**0.1784**

For the next experiment, we implement a curriculum learning strategy that starts with a basic SupCon approach and gradually incorporates CHNS at different stages of training of the ECAPA-TDNN model. Specifically, we compare our default approach of using CHNS throughout the entire training process to introducing it after 10 epochs (5% of the total training time) and after 100 epochs (50 % of the total training time). For the curriculum learning approaches, we calculate the clusters for CHNS using the models trained for 10 or 100 epochs, thus avoiding the need for a pre-trained model to calculate voiceprint clusters before training. The results provided in Table 3 indicate that starting CHNS training from the beginning yields the best performance. However, they also suggest that a model after just 5 % of the total training time is sufficient to produce the embeddings required to calculate the voiceprints in our method.

In the final experiment, we verify the model agnostic properties of CHNS, by comparing different training strategies on another lightweight model, i.e. Thin ResNet-34 [9]. We train the Thin ResNet-34 encoder on the VoxCeleb2 training dataset and test on VoxCeleb1-H. The results presented in Table 4 confirm a clear advantage of using the proposed CHNS approach with another popular architecture for speaker verification.

5. Conclusion

In this paper, we proposed CHNS - a clustering-based hard negative sampling approach for supervised contrastive speaker verification. Through a series of experiments using lightweight models suitable for use on edge devices, we demonstrated that our method outperforms standard supervised contrastive learning with and without loss-based hard negative sampling, as well as a state-of-the-art classification based approach to speaker verification by as much as 18 % relative EER and minDCF, while being dataset and model agnostic. We open-sourced our training code.

6. References

[1] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[2] Y.-Q. Yu and W.-J. Li, "Densely Connected Time Delay Neural Network for Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 921–925.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[4] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, Oct. 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2650

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, M. Zeng, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1505–1518, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:239885872

[6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[7] Z. Li, M.-W. Mak, and H. M.-L. Meng, "Discriminative speaker representation via contrastive learning with class-aware attention in angular space," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] Y. Tang, J. Wang, X. Qu, and J. Xiao, "Contrastive learning for improving end-to-end speaker verification," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–7.

[9] 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 *Proc. Interspeech 2020*, 2020, pp. 2977–2981.

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

[11] T. Lepage and R. Dehak, "Label-Efficient Self-Supervised Speaker Verification With Information Maximization and Contrastive Learning," in *Proc. Interspeech 2022*, 2022, pp. 4018–4022.

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

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

[14] T. Lepage and R. Dehak, "Experimenting with Additive Margins for Contrastive Self-Supervised Speaker Verification," in *Proc. INTERSPEECH 2023*, 2023, pp. 4708–4712.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[16] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[17] C. Go, Y. H. Lee, T. Kim, N. I. Park, and C. Chun, "Contrastive speaker representation learning with hard negative sampling for speaker recognition," *Sensors*, vol. 24, no. 19, 2024.

[18] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=CR1XOQ0UTh-

[19] R. Jiang, T. Nguyen, P. Ishwar, and S. Aeron, "Supervised contrastive learning with hard negative samples," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[20] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[21] P. Gomez and E. San Segundo, "Voice biometrical match of twin and non-twin siblings," in *Proceedings of the 8th International Workshop Models and analysis of vocal emissions for biomedical applications*, 06 2013.

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

[23] B. Han, W. Huang, Z. Chen, and Y. Qian, "Improving dino-based self-supervised speaker verification with progressive cluster-aware training," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2023, pp. 1–5.

[24] H. Mao, F. Hong, and M.-w. Mak, "Cluster-guided unsupervised domain adaptation for deep speaker embedding," *IEEE Signal Processing Letters*, vol. 30, pp. 643–647, 2023.

[25] E. Schubert, "A triangle inequality for cosine similarity," in *Similarity Search and Applications*, N. Reyes, R. Connor, N. Kriege, D. Kazempour, I. Bartolini, E. Schubert, and J.-J. Chen, Eds. Cham: Springer International Publishing, 2021, pp. 32–44.

[26] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[28] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *IEEE ICASSP 2020*. IEEE, 2020, pp. 7604–7608.