Additive Margin in Contrastive Self-Supervised Frameworks to Learn Discriminative Speaker Representations

Theo Lepage, Reda Dehak

EPITA Research Laboratory (LRE), France

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

Abstract

Self-Supervised Learning (SSL) frameworks became the standard for learning robust class representations by benefiting from large unlabeled datasets. For Speaker Verification (SV), most SSL systems rely on contrastive-based loss functions. We explore different ways to improve the performance of these techniques by revisiting the NT-Xent contrastive loss. Our main contribution is the definition of the NT-Xent-AM loss and the study of the importance of Additive Margin (AM) in SimCLR and MoCo SSL methods to further separate positive from negative pairs. Despite class collisions, we show that AM enhances the compactness of same-speaker embeddings and reduces the number of false negatives and false positives on SV. Additionally, we demonstrate the effectiveness of the symmetric contrastive loss, which provides more supervision for the SSL task. Implementing these two modifications to SimCLR improves performance and results in 7.85% EER on VoxCeleb1-O, outperforming other equivalent methods.

1. Introduction

Most Speaker Recognition (SR) methods aim at learning an embedding space making each class distribution compact with a large dispersion between classes. The objective is to learn representations with small intra-speaker distances and large inter-speaker distances while being robust to extrinsic information. Different speech feature extraction methods and machine learning frameworks were proposed for this task. The i-vectors [1] is an unsupervised method that learns an embedding capturing the main variabilities of the speech signal and relies on a back-end (LDA and/or PLDA) to focus on the speaker or language information [2]. Several solutions based on deep neural networks were recently proposed to learn this embedding space [3]. The d-vector [4], and the x-vectors [5, 6, 7] approaches embed speakers audio features into a vector space using deep neural networks.

The drawback of these supervised methods is that they require large labeled datasets during training. Despite the impressive progress made with supervised learning, this paradigm is now considered a bottleneck for building more intelligent systems. Manually annotating data is complex, expensive, and tedious, especially when dealing with signals such as images, text, and speech. Moreover, the risk is creating biased models that do not perform well in real-life scenarios, notably in difficult acoustic conditions.

Recently, methods based on Self-Supervised Learning (SSL) have been developed to train deep neural networks in an unsupervised way by benefiting from massive amounts of unlabeled data. Motivated by the performance obtained by these approaches in computer vision and natural language processing (NLP), several techniques have been applied to learn speaker representations [8, 9, 10, 11, 12, 13]. These frameworks rely on the assumption that segments extracted from the same utterance belong to the same speaker while those from different utterances belong to distinct speakers. This assumption does not always hold (*class collision* issue), but the impact on the training convergence is negligible. The main challenge resulting from this training scenario is that segments extracted from the same utterance share several similar characteristics, such as channel, language, speaker, and sentiment information [14]. Thus, augmenting speech segments is fundamental to only encode speaker-related information.

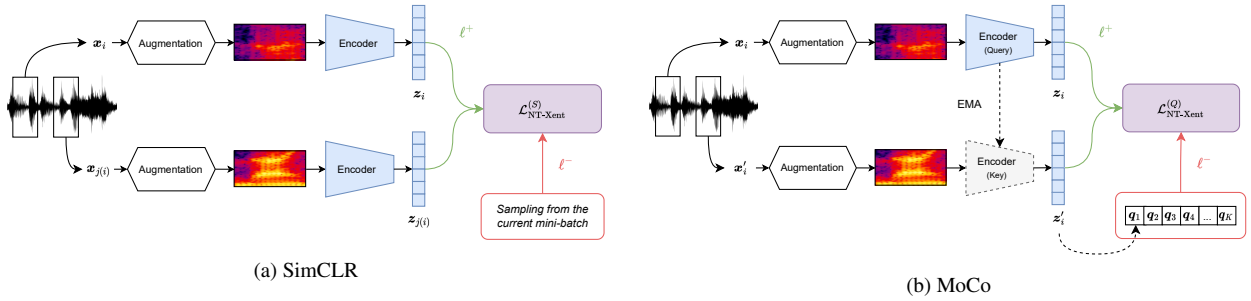
Most SSL methods proposed for Speaker Verification (SV) are contrastive. SimCLR [11] and MoCo [10] have been successfully applied to this field of research by focusing on how to define the model architecture and sample negative pairs for the training. These approaches rely on objective functions based on the Normalized Temperature-scaled Cross Entropy (*NT-Xent*) loss. According to the self-supervised assumption, the contrastive loss aims at reducing the distance between positive pairs while increasing the distance between negative pairs in a latent space.

Additive Margin from CosFace (*AM-Softmax*) [15] and Additive Angular Margin from ArcFace (*AAM-Softmax*) [16] have been successfully applied to improve angular softmax-based objective functions which are at the core of supervised SR systems [17, 18, 19]. Inspired by the performance obtained with these techniques on speaker verification, we introduce *additive margin* into the *NT-Xent* loss to improve the discriminative capacity of the embeddings by increasing speaker separability. We demonstrate the importance of this concept in a self-supervised setup as we obtained better performance when training SimCLR and MoCo SSL frameworks with *additive margin*. We also verify the impact of class collisions as *additive margin* could be applied between two samples from the same class. Moreover, we show that using a *symmetric* contrastive loss using all possible positive and negative pairs is essential to improve the downstream performance of SimCLR.

This work extends the study presented in [20] and provides results with models trained on VoxCeleb2 rather than VoxCeleb1. With this larger training set, we have seen that the projector is not useful. Moreover, we implement *additive margin* in the MoCo framework to extend this concept to other SSL methods. Finally, we verify that class collisions are not an issue when implementing *additive margin* in a self-supervised training scenario.

Our self-supervised training framework is described in Sec-

The code associated with this article is publicly available at https://github.com/theolepage/sslsv.



(a) SimCLR

(b) MoCo

Figure 1: Diagram of our contrastive self-supervised training framework to learn speaker representations.

tion 2. In particular, we revisit the standard contrastive loss and introduce *additive margin* by defining *NT-Xent-AM*. In Section 3, we present our experimental setup. We report our results and assess the effect of *additive margin* in Section 4. Finally, we conclude in Section 5.

2. Method

Our self-supervised training framework relies on a simple siamese architecture to produce a pair of embeddings for a given unlabeled utterance.

For each training step, we randomly sample N utterances from the dataset. Let $i \in I \equiv \{1 \dots N\}$ be the index of a random sample from the mini-batch. We extract two non-overlapping frames, denoted as $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$, from each utterance. Then, we apply random augmentations to both copies and use their mel-scaled spectrogram as input features. Using distinct frames and applying data augmentation is fundamental to avoid collapse and to produce robust representations that mainly contain speaker identity. An encoder transforms $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ to their respective representations $\boldsymbol{z}_i$ and $\boldsymbol{z}'_i$. During training, mini-batches are created by stacking $\boldsymbol{z}_i$ samples into $\boldsymbol{Z}$ and $\boldsymbol{z}'_i$ samples into $\boldsymbol{Z}'$. Representations are used to perform speaker verification and to compute the loss.

2.1. Contrastive-based self-supervised learning

Contrastive learning aims at maximizing the similarity within positive pairs while maximizing the distance between negative pairs. In self-supervised learning, supervision is provided by assuming that two randomly sampled utterances belong to different speakers. Positive pairs are constructed with embeddings derived from the same utterances, while negative pairs are sampled from the current mini-batch or a memory queue.

We start by defining the similarity between two embeddings $\boldsymbol{u}$ and $\boldsymbol{v}$ as $\ell(\boldsymbol{u}, \boldsymbol{v}) = e^{\cos(\theta_{\boldsymbol{u},\boldsymbol{v}})/\tau}$ where τ is a temperature scaling hyper-parameter and $\theta_{\boldsymbol{u},\boldsymbol{v}}$ is the angle between the two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. $\cos(\theta_{\boldsymbol{u},\boldsymbol{v}})$ corresponds to the cosine similarity and is obtained by computing the dot product between the two l_2 normalized embeddings $\boldsymbol{u}$ and $\boldsymbol{v}$. Then, the Normalized Temperature-scaled Cross Entropy loss (*NT-Xent*) is defined as

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{N}\sum_{i\in I}\log\frac{\ell\left(\boldsymbol{z}_i, \boldsymbol{z}'_i\right)}{\sum\limits_{a\in I}\ell\left(\boldsymbol{z}_i, \boldsymbol{z}'_a\right)}. \quad (1)$$

We refer to $\boldsymbol{z}_i$ as the *anchor*, $\boldsymbol{z}'_i$ as the *positive*, and $\boldsymbol{z}'_a$ as a *negative* when $a \neq i$. Thus, N positive pairs are created, and compared to $N-1$ negatives.

Finally, we propose to compute the similarity of positive and negative pairs differently to introduce *additive margin* later on, such that

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{N}\sum_{i\in I}\log\frac{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}'_i\right)}{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}'_i\right) + \sum\limits_{a\in A(i)}\ell^-\left(\boldsymbol{z}_i, \boldsymbol{z}'_a\right)}, \quad (2)$$

where $\ell^+(\mathbf{u}, \mathbf{v}) = \ell^-(\mathbf{u}, \mathbf{v}) = e^{\cos(\theta_{\boldsymbol{u},\boldsymbol{v}})/\tau}$ and $A(i) \equiv I \setminus \{i\}$.

2.1.1. SimCLR – Sampling negatives from the mini-batch

Based on the above-mentioned technique, we adopt the SimCLR training framework, depicted in Figure 1 (a). The previous formulation of the contrastive loss, which has been adopted in [21, 11], does not consider all possible positive and negative pairs. Instead, following the original implementation [22], we propose to use the *symmetric* formulation of the *NT-Xent* loss to increase the number of contrastive samples.

We now consider $\boldsymbol{z}_i$ to be the i-th element of a set of all embeddings created by concatenating $\boldsymbol{Z}$ and $\boldsymbol{Z}'$. Let $i \in \hat{I} \equiv \{1 \dots 2N\}$ be the index of a random augmented frame, $j(i)$ be the index of the other augmented frame from the same utterance being in the same mini-batch and $\hat{A}(i) \equiv \hat{I} \setminus \{i, j(i)\}$. The *Symmetric NT-Xent* loss is defined as

$$\mathcal{L}^{(S)}_{\text{NT-Xent}} = -\frac{1}{2N}\sum_{i\in\hat{I}}\log\frac{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}_{j(i)}\right)}{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}_{j(i)}\right) + \sum\limits_{a\in\hat{A}(i)}\ell^-\left(\boldsymbol{z}_i, \boldsymbol{z}_a\right)} \quad (3)$$

Note that this results in $2N$ positive pairs which will be compared against a set of $2(N-1)$ negatives (all utterances in the mini-batch except the positive and the anchor).

2.1.2. MoCo – Sampling negatives from a memory queue

We also explore the MoCo training framework, depicted in Figure 1 (b). Instead of being constrained by the batch size N, this method samples negatives from a larger memory queue to increase the number of negative samples. The major difference with SimCLR is that the gradient is not propagated through the second branch of the siamese architecture as the *key (momentum)* encoder is frozen and updated with an Exponential Moving Average (EMA) of the *query* encoder parameters. The queue contains K elements and is updated dynamically at each step of the training with embeddings from the *key* encoder. Thus, all N positive pairs are compared to K negatives.

We define the *Queue-based NT-Xent* loss such that

$$\mathcal{L}^{(Q)}_{\text{NT-Xent}} = -\frac{1}{N}\sum_{i\in I}\log\frac{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}'_i\right)}{\ell^+\left(\boldsymbol{z}_i, \boldsymbol{z}'_i\right) + \sum\limits_{b\in B}\ell^-\left(\boldsymbol{z}_i, \boldsymbol{q}_b\right)}, \quad (4)$$

where $B \equiv \{1 \dots K\}$ and $\boldsymbol{q}_i$ is the i-th element of the queue.

2.2. Improving speaker separability with additive margin

The contrastive loss is at the core of most self-supervised learning frameworks. However, it aims to penalize classification errors instead of producing discriminative representations relevant to the context of speaker verification.

Inspired by state-of-the-art techniques for face recognition, *additive margin* were successfully applied for training end-to-end speaker verification models in a supervised way [17, 18, 19], which justifies our motivation to adapt this concept for self-supervised learning. Following CosFace (AM-Softmax) [15], we introduce *additive margin* in cosine space to increase the similarity of same-speaker embeddings and improve the discriminative capacity of our contrastive-based objective functions.

We define the Normalized Temperature-scaled Cross Entropy with Additive Margin loss (*NT-Xent-AM*), $\mathcal{L}_{\text{NT-Xent-AM}}$, by setting

$$\ell^{+}(\mathbf{u},\mathbf{v}) = e^{\left(\cos\left(\theta_{\mathbf{u},\mathbf{v}}\right)-m\right)/\tau}, \tag{5}$$

where $m \geq 0$ is a fixed scalar introduced to control the magnitude of the cosine margin, while $\ell^{-}(\mathbf{u},\mathbf{v})$ remains unchanged.

Considering an *anchor* $\boldsymbol{z}_a$, a *positive* $\boldsymbol{z}_p$, and a *negative* $\boldsymbol{z}_n$, the effect of this technique is to learn a constraint of the form $\cos\left(\theta_{\boldsymbol{z}_a,\boldsymbol{z}_p}\right) - m > \cos\left(\theta_{\boldsymbol{z}_a,\boldsymbol{z}_n}\right)$ instead of $\cos\left(\theta_{\boldsymbol{z}_a,\boldsymbol{z}_p}\right) > \cos\left(\theta_{\boldsymbol{z}_a,\boldsymbol{z}_n}\right)$, which set the positive similarity at least greater than the maximal negative similarity plus the margin constant.

3. Experimental setup

3.1. Datasets and feature extraction

Our method is trained on the VoxCeleb2 dev set which is composed of $1,092,009$ utterances from $5,994$ speakers. The evaluation is performed on VoxCeleb1 *original* test set containing $4,874$ utterances from 40 speakers. According to our self-supervised training protocol, speaker labels are discarded.

The duration of extracted audio segments is 2 seconds during training and 3.5 seconds when performing speaker verification. Input features are obtained by computing 40-dimensional log-mel spectrogram features with a Hamming window of 25 ms length and 10 ms frame-shift. As training data consists mostly of continuous speech segments, Voice Activity Detection (VAD) is not applied. The input features are normalized using instance normalization.

3.2. Data-augmentation

Self-supervised learning frameworks commonly rely on extensive data-augmentation techniques to learn representations robust against extrinsic variabilities (noise from the environment, mismatching recording device...). Because positives are sampled from the same utterance, providing different augmented versions of the same utterance is fundamental to avoid encoding channel characteristics, allowing speaker identity to be the only distinguishing factor between two representations.

Thus, at each training step, we randomly apply two types of transformations to the input signal. First, we add background noises, overlapping music tracks, or speech segments using the MUSAN corpus. To simulate various real-world scenarios, we randomly sample the Signal-to-Noise Ratio (SNR) between $[13; 20]$ dB for speech, $[5; 15]$ dB for music, and $[0; 15]$ dB for noises. Then, to further enhance the robustness of our model, we apply reverberation to the augmented utterances using the simulated Room Impulse Response database.

3.3. Models architecture and training

The encoder neural network follows the Fast ResNet-34 [7] model architecture with 512 output units. Utterance-level representations are generated using Self-Attentive Pooling (SAP).

Unlike most self-supervised frameworks we do not rely on a projector module, similar to other works on SSL for SV [11, 21]. Our previous work [13] relied on a projector as it resulted in better performance when training on VoxCeleb1. As positive pairs are extracted from the same utterances, contrastive SSL methods are prone to encoding too much channel information while the projector allows extracting latent representations containing more speaker information. Thus, we hypothesize that this module was required on a smaller training set (VoxCeleb1) to reduce the overfitting on channel characteristics, avoiding a misalignment between the SSL objective and the downstream SV task.

We set the temperature to $\tau = \frac{1}{30}$ to match the scaling factor $s = 30$ used commonly when training AM-Softmax supervised losses for speaker verification. When training the framework based on MoCo, we use a memory queue of $10,000$ negatives and update the frozen encoder with an exponential moving average (EMA) using a coefficient of 0.999. We optimize the model with Adam optimizer and a learning rate of 0.001, which is reduced by 5% every 5 epochs, with no weight decay. We use a batch size of 200 and train SimCLR for 150 epochs and MoCo for 100 epochs. Our implementation is based on the PyTorch framework and we conduct our experiments on $2 \times$ NVIDIA Tesla V100 16 GB.

3.4. Evaluation protocol

To evaluate our model's performance on speaker verification, we extract embeddings from 10 evenly spaced frames for each test utterance. Then, to determine the scoring of each trial, we compute the average of the cosine similarity between all 100 combinations of l_2-normalized embeddings pairs. Following VoxCeleb and NIST Speaker Recognition evaluation protocols, we report the performance of our model in terms of Equal Error Rate (EER) and minimum Detection Cost Function (minDCF) with $P_{target} = 0.01$, $C_{\text{miss}} = 1$ and $C_{\text{fa}} = 1$.

4. Results and discussions

4.1. Effect of our improvements to the self-supervised contrastive loss

We conduct a study to assess the role of the different components of our self-supervised training framework based on SimCLR and report the results in Table 1. Our baseline, similar to the work of [11], trained with the $\mathcal{L}_{\text{NT-Xent}}$ loss achieves 8.98% EER and 0.6714 minDCF.

Firstly, we show that relying on more positive and negative pairs with the symmetric contrastive loss results in 8.41% EER and 0.6235 minDCF. This validates our intuition that providing more supervision and more negative samples is beneficial to improve the self-supervised system's downstream performance.

Secondly, we evaluate the role of *additive margin* by showing that the choice of the margin value significantly impacts speaker verification results. The best setting is obtained with $m = 0.1$ achieving 7.85% EER and 0.6168 minDCF, representing 12.6% relative improvement of the EER over the baseline. Intuitively, a small margin does not affect the results, but a very large margin increases the training task complexity. It is noteworthy that $m = 0.1$ corresponds to the value often used

Table 1: The effect of the symmetric contrastive loss and additive margin on SimCLR self-supervised training for speaker verification.

Loss	Sym.	Margin	EER (%)	minDCF$_{0.01}$
$\mathcal{L}_{\text{NT-Xent}}$	×	0	8.98	0.6714
$\mathcal{L}^{(S)}_{\text{NT-Xent}}$	✓	0	8.41	0.6235
$\mathcal{L}^{(S)}_{\text{NT-Xent-AM}}$	✓	0.05	8.35	0.6098
		0.1	7.85	0.6168
		0.2	8.13	0.6211

Table 2: The effect of additive margin on MoCo self-supervised training for speaker verification.

Loss	Margin	EER (%)	minDCF$_{0.01}$
$\mathcal{L}^{(Q)}_{\text{NT-Xent}}$	0	9.59	0.6974
$\mathcal{L}^{(Q)}_{\text{NT-Xent-AM}}$	0.1	9.36	0.6403

for AM-Softmax supervised training and that, according to our previous experiments on VoxCeleb1 [20], learning the margin value jointly with the model degrades the performance.

Finally, we observed the same positive effect when training MoCo with *additive margin*. As shown in Table 2, $\mathcal{L}^{(Q)}_{\text{NT-Xent-AM}}$ with $m = 0.1$ improves the EER from 9.59% to 9.36% when compared to $\mathcal{L}^{(Q)}_{\text{NT-Xent}}$. The performances on MoCo are not as competitive compared to SimCLR because we did not optimize the different hyper-parameters related to this algorithm.

Thus, *additive margin* is essential in self-supervised contrastive frameworks to improve results in speaker verification. We hypothesize that these improvements could benefit other downstream tasks related to verification.

4.2. Impact of class collisions and class imbalance on the self-supervised training

Class collisions occur when at least one negative, sampled from the mini-batch or the memory queue, belongs to the same speaker as the positive sample. The resulting false negative samples could represent an issue when using *additive margin* with contrastive self-supervised training.

We conducted two experiments to test the effect of class collisions and class imbalance on the training. For the first experiment, we trained our SimCLR framework but used the train set labels to prevent class collisions in each mini-batch. For the second experiment, we limited the number of samples for over-represented speakers in the training set.

As shown in Table 3, removing class collisions does not result in better downstream performance. Our assumption is that the probability of class collisions is too small as VoxCeleb2 contains many speakers compared to the batch size. Moreover, preventing class imbalance by limiting the number of samples for some speakers does not improve final speaker verification results. Thus, *additive margin* can be used reliably in a self-supervised scenario.

Table 3: The impact of class collisions and class imbalance, which stems from the self-supervised training, on speaker verification results. The baseline is our SimCLR framework trained with $\mathcal{L}^{(S)}_{\text{NT-Xent-AM}}$ ($m = 0.1$).

Class collisions	Class imbalance	EER (%)	minDCF$_{0.01}$
✓	✓	7.85	0.6168
×	✓	7.95	0.6241
×	×	8.41	0.6390

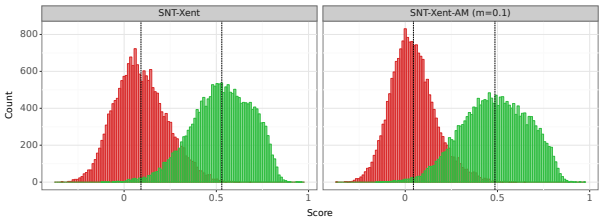


Figure 2: Positive (green) and negative (red) trials score distribution on VoxCeleb1-O and obtained after training our SimCLR framework with $\mathcal{L}^{(S)}_{\text{NT-Xent}}$ and $\mathcal{L}^{(S)}_{\text{NT-Xent-AM}}$ ($m = 0.1$) losses. The mean of each distribution is represented by a dashed line.

4.3. Visualization of trials score distribution with and without additive margin

In Figure 2, we plot the distribution of scores computed on VoxCeleb1 original test set to assess the effect of *additive margin* on SimCLR with $m = 0.1$. Visually, we can notice two improvements when introducing *additive margin*: (1) the means of the two distributions are further apart; (2) the spread of negative scores is smaller.

This observation is consistent with the improvement of the EER and the minDCF. It shows that *additive margin* positively affects the discriminative capacity of representations learned by self-supervised systems designed for verification tasks.

4.4. Comparison to other self-supervised contrastive methods for speaker verification

We report the final results on speaker verification in Table 4 to compare to other contrastive self-supervised methods. Methods reported in the top part are trained with objective functions equivalent to $\mathcal{L}_{\text{NT-Xent}}$. Thus, they do not rely on the symmetric formulation of the contrastive loss and do not compute the similarity of positive and negative pairs differently to introduce *additive margin*.

We outperform other contrastive approaches as the best result is obtained with our framework based on SimCLR, which achieves 7.85% EER. Therefore, we showed that standard contrastive methods can be further improved by introducing simple changes tailored for the self-supervised training condition (providing more supervision with additional positives and negatives) and for the downstream task (increasing speaker separability for enhanced speaker verification).

Table 4: Final results of different self-supervised contrastive methods on speaker verification. Note that $\mathcal{L}_{\text{AP}}$ corresponds to $\mathcal{L}_{\text{NT-Xent}}$ without a temperature but a learnable weight and bias.

Method	**Loss**	**EER (%)**	**minDCF$_{0.01}$**
AP [21]	$\mathcal{L}_{\text{AP}}$	9.56	–
SimCLR [11]	$\mathcal{L}_{\text{AP}}$	8.28	0.6100
MoCo [10]	$\mathcal{L}_{\text{NT-Xent}}$	8.23	0.5900
SimCLR	$\mathcal{L}^{(S)}_{\text{NT-Xent-AM}}$	7.85	0.6168
MoCo	$\mathcal{L}^{(Q)}_{\text{NT-Xent-AM}}$	9.36	0.6403

5. Conclusions

This article introduces *additive margin* to self-supervised contrastive frameworks by defining *NT-Xent-AM* to learn more robust speaker representations. We demonstrated that this improvement results in a lower EER on the VoxCeleb test set and a better discrepancy between scores of positive and negative trials, even in a self-supervised training scenario. We showed that *additive margin* combined with the symmetric formulation of the contrastive loss applied to SimCLR outperforms the other equivalent contrastive approaches. We expect NT-Xent-AM to be also effective in training SSL contrastive models for other tasks and modalities in image or speech processing (language recognition, ...).

6. Acknowledgements

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

7. References

[1] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE TASLP*, 2011.

[2] Najim Dehak, Pedro A. Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Interspeech*, 2011.

[3] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A. Torres-Carrasquillo, and Najim Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations," *Computer Speech & Language*, 2020.

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*, 2014.

[5] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *ICASSP*, 2018.

[6] Joon Son Chung, Jaesung Huh, and Seongkyu Mun, "Delving into VoxCeleb: Environment invariant speaker recognition," in *Odyssey*, 2020.

[7] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In Defence of Metric Learning for Speaker Recognition," in *Interspeech*, 2020.

[8] Themos Stafylakis, Johan Rohdin, Oldřich Plchot, Petr Mizera, and Lukáš Burget, "Self-Supervised Speaker Embeddings," in *Interspeech*, 2019.

[9] Jaejin Cho, Piotr Żelasko, Jesús Villalba, Shinji Watanabe, and Najim Dehak, "Learning Speaker Embedding from Text-to-Speech," in *Interspeech*, 2020.

[10] Wei Xia, Chunlei Zhang, Chao Weng, Meng Yu, and Dong Yu, "Self-Supervised Text-Independent Speaker Verification Using Prototypical Momentum Contrastive Learning," in *ICASSP*, 2021.

[11] Haoran Zhang, Yuexian Zou, and Helin Wang, "Contrastive Self-Supervised Learning for Text-Independent Speaker Verification," in *ICASSP*, 2021.

[12] Jaejin Cho, Jesus Villalba, Laureano Moro-Velazquez, and Najim Dehak, "Non-Contrastive Self-supervised Learning for Utterance-Level Information Extraction from Speech," *IEEE JSTSP*, 2022.

[13] Théo Lepage and Réda Dehak, "Label-Efficient Self-Supervised Speaker Verification With Information Maximization and Contrastive Learning," in *Interspeech*, 2022.

[14] Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe, "Self-supervised speech representation learning: A review," *IEEE JSTSP*, 2022.

[15] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018.

[16] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019.

[17] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP*, 2019.

[18] Yi Liu, Liang He, and Jia Liu, "Large Margin Softmax Loss for Speaker Verification," in *Interspeech*, 2019.

[19] Ya-Qi Yu, Lei Fan, and Wu-Jun Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP*, 2019.

[20] Théo Lepage and Réda Dehak, "Experimenting with Additive Margins for Contrastive Self-Supervised Speaker Verification," in *Interspeech*, 2023.

[21] Jaesung Huh, Hee Soo Heo, Jingu Kang, Shinji Watanabe, and Joon Son Chung, "Augmentation adversarial training for unsupervised speaker recognition," in *Workshop on Self-Supervised Learning for Speech and Audio Processing, NeurIPS*, 2020.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *ICML*, 2020.