Semi-Supervised Contrastive Learning with Generalized Contrastive Loss and Its Application to Speaker Recognition

Nakamasa Inoue and Keita Goto
Tokyo Institute of Technology, Tokyo, Japan
E-mail: inoue@c.titech.ac.jp

***Abstract*—This paper introduces a semi-supervised contrastive learning framework and its application to text-independent speaker verification. The proposed framework employs generalized contrastive loss (GCL). GCL unifies losses from two different learning frameworks, supervised metric learning and unsupervised contrastive learning, and thus it naturally determines the loss for semi-supervised learning. In experiments, we applied the proposed framework to text-independent speaker verification on the VoxCeleb dataset. We demonstrate that GCL enables the learning of speaker embeddings in three manners, supervised learning, semi-supervised learning, and unsupervised learning, without any changes in the definition of the loss function.**

I. Introduction

With the development of various optimization techniques, deep learning has become a powerful tool for numerous applications, including speech and image recognition. To build high-performance models, supervised learning is the most popular methodology, in which labeled samples are used for optimizing model parameters. It is known that deep neural networks (e.g., ResNet [1]) having more than a million parameters outperform hand-crafted feature extraction methods. As such, optimizing parameters with a well-designed objective function is one of the most important research topics in deep learning.

In recent years, supervised metric learning methods for deep neural networks have attracted attention. Examples of these include triplet loss [2] and prototypical episode loss [3], which predispose a network to minimize within-class distance and maximize between-class distance. They are also effective for text-independent speaker verification, as shown in [4], because cosine similarity between utterances from the same speaker is directly maximized in the training phase.

Nevertheless, unsupervised learning methods have grown greatly, thanks to large-scale collections of unlabeled samples. Some studies have recently proven that self-supervised learning achieves performance very close to that of supervised learning. For example, the simple framework for contrastive learning of representations (SimCLR) [5] provides superior image representation by introducing contrastive NT-Xent loss using data augmentation on unlabeled images. For speaker verification, these methods motivate us to explore unsupervised and semi-supervised ways to learn speaker embeddings by effectively using unlabeled utterances.

In general, supervised learning and unsupervised learning depend on different methodologies. However, supervised metric learning and unsupervised contrastive learning share a common idea to maximize or minimize the similarity between samples. This implies the possibility of unifying these two learning frameworks.

In this paper, we propose a semi-supervised contrastive learning framework based on generalized contrastive loss (GCL). GCL provides a unified formulation of two different losses from supervised metric learning and unsupervised contrastive learning. Thus, it naturally works as a loss function for semi-supervised learning. In experiments, we applied the proposed framework to text-independent speaker verification on the VoxCeleb dataset. We demonstrated that GCL enables the network to learn speaker embeddings in three manners, supervised learning, semi-supervised learning, and unsupervised learning, without any changes in the definition of the loss function.

II. Related Work

A. Supervised Metric Learning

Supervised metric learning is a framework to learn a metric space from a given set of labeled training samples. For recognition problems, such as audio and image recognition, the goal is typically to learn the semantic distance between samples.

A recent trend in supervised metric learning is to design a loss function at the top of a deep neural network. Examples include contrastive loss for Siamese networks [6], triplet loss for triplet networks [2], and episode loss for prototypical networks [3]. To measure the distance between samples, Euclidean distance is often used with these losses.

For face identification from images, measuring similarity by cosine similarity often improves the performance. ArcFace [7], CosFace [8], and SphereFace [9] are its popular implementations. Their effectiveness is also shown in speaker verification from audio samples with some extended loss definitions, such as ring loss [10], [11]. One of the best choices for speaker verification is angle-prototypical loss [4], which introduces cosine similarity to episode loss, as shown in [4] with thorough experiments.

978-988-14768-8-3/20/$31.00 ©2020 APSIPA

B. Unsupervised Contrastive Learning

Unsupervised learning is a framework to train a model from a given set of unlabeled training samples. Classic methods for unsupervised learning include clustering methods such as K-means clustering [12]. Most of them are statistical approaches with some objectives based on means and variances.

Recently, self-supervised learning has proven to be effective for pre-training deep neural networks. For example, Jigsaw [13] and Rotation [14] define a pretext task on unlabeled data and pre-train networks for image recognition by solving it. Deep InfoMax [15] and its multiscale extension AMDIM [16] focus on mutual information between representations extracted from multiple views of a context. SimCLR [5] introduces contrastive learning using data augmentation. The effectiveness of contrastive learning is also shown in MoCo V2 [17], [18]. These methods achieve performance comparable with that of supervised learning in tasks of image representation learning.

Cross-modal approaches are also effective if more than one source is available. For speaker verification, Nagrani et al. [19] proposed a cross-modal self-supervised learning method, which uses face images as supervision of audio signals to identify speakers.

C. Semi-Supervised Learning

Semi-supervised learning is a framework to train a model from a set consisting of both labeled and unlabeled samples. To effectively incorporate information from unlabeled samples into the parameter optimization step, a regularization term is often introduced into the objective function. For example, consistency regularization [20] is used to penalize sensitivity to augmented unlabeled samples.

For speaker verification, Stafylakis et al. [21] proposed self-supervised speaker embeddings. A pre-trained automatic speech recognition system is utilized to make a supervision signal of phoneme information on unlabeled utterances.

III. Preliminary

A. Supervised Metric Learning

Let $\mathcal{D}$ be a training dataset for supervised learning, which consists of sample pairs $\boldsymbol{x}$ and their discrete class label y. The goal of supervised metric learning is to learn a metric function $d(\boldsymbol{x}, \boldsymbol{x}')$, which assigns a small distance between samples belonging to the same class, and relatively large distance between samples from different classes. Assuming that the training phase has iterations for parameter updates, a mini-batch B is sampled from $\mathcal{D}$ at each iteration. For convenience, two-step sampling is often used [4]. First, a set of N different classes are randomly sampled from the set of training classes. We denote the sampled classes by $y_1, y_2, \cdots, y_N$. Second, K independent samples are randomly sampled from each of N classes. We denote the samples from the class y_i as $\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \cdots, \boldsymbol{x}_i^K$. As a result, a mini-batch $B = \{(\boldsymbol{x}_i^k, y_i) : i = 1, 2, \cdots, N, k = 1, 2, \cdots, K\}$ consists of NK samples.

As an example of supervised metric learning, we show the training process of a prototypical network [3]. The main idea of a prototypical network is to make prototype representations of each class and to minimize the distance between a query sample and its corresponding prototype. Its loss for parameter updates is computed as follows:

1) Sample a mini-batch B from $\mathcal{D}$ and split it into a query set $Q = \{(\boldsymbol{x}_i^1, y_i) \in B : k = 1\}$ and a support set $S = \{(\boldsymbol{x}_i^k, y_i) \in B : k > 1\}$.
2) Extract query representations $\boldsymbol{z}_i^1$ from Q by
$$\boldsymbol{z}_i^1 = f_\theta(\boldsymbol{x}_i^1), \tag{1}$$
where f_θ is a neural network for embedding (i.e., a network without the final loss layer) and θ is a set of parameters.
3) Construct prototype representations $\boldsymbol{z}_i^2$ from S by
$$\boldsymbol{z}_i^2 = \frac{1}{K-1} \sum_{k=2}^{K} f_\theta(\boldsymbol{x}_i^k). \tag{2}$$
4) From a representation batch $Z = \{\boldsymbol{z}_i^k : i = 1, 2, \cdots, N, k = 1, 2\}$, compute the episode loss defined by
$$L = -\frac{1}{N} \sum_i \log \frac{s(\boldsymbol{z}_i^1, \boldsymbol{z}_i^2)}{\sum_j s(\boldsymbol{z}_i^1, \boldsymbol{z}_j^2)}, \tag{3}$$
where s is the exponential function of negative distance between representations $s(\boldsymbol{z}, \boldsymbol{z}') := \exp(-d(\boldsymbol{z}, \boldsymbol{z}'))$, and d is the squared Euclidean distance.

B. Unsupervised Contrastive Learning

Let $\mathcal{U}$ be a training dataset for unsupervised learning, which consists of unlabeled samples $\boldsymbol{u}$. The goal of unsupervised learning is to train networks without any manually attached labels.

As an example of unsupervised learning, we show the training process of SimCLR [5]. SimCLR maximizes the similarity between representations of two augmented samples $t_1(\boldsymbol{u})$ and $t_2(\boldsymbol{u})$, where t_1 and t_2 are two randomly selected augmentation functions. Its loss for parameter updates is computed as follows:

1) Sample a mini-batch $B = \{\boldsymbol{u}_i : i = 1, 2, \cdots, N\}$ from $\mathcal{U}$.
2) Extract the first representation $\boldsymbol{z}_i^1$ by
$$\boldsymbol{z}_i^1 = f_\theta(t_1(\boldsymbol{u}_i)). \tag{4}$$
Note that t_1 is randomly selected from a set of augmentation functions for each i.
3) Extract the second representation $\boldsymbol{z}_i^2$ by
$$\boldsymbol{z}_i^2 = f_\theta(t_2(\boldsymbol{u}_i)). \tag{5}$$
4) From a representation batch $Z = \{\boldsymbol{z}_i^k : i = 1, 2, \cdots, N, k = 1, 2\}$, compute the NT-Xent loss [5] defined by
$$L_s = \frac{1}{2}(\ell_{12} + \ell_{21}), \tag{6}$$

where

$$\ell_{12} = -\frac{1}{N}\sum_i \log \frac{s(\boldsymbol{z}_i^1, \boldsymbol{z}_i^2)}{\sum_j s(\boldsymbol{z}_i^1, \boldsymbol{z}_j^2) + \sum_{j\neq i} s(\boldsymbol{z}_i^1, \boldsymbol{z}_j^1)}, \tag{7}$$

$$\ell_{21} = -\frac{1}{N}\sum_i \log \frac{s(\boldsymbol{z}_i^2, \boldsymbol{z}_i^1)}{\sum_j s(\boldsymbol{z}_i^2, \boldsymbol{z}_j^1) + \sum_{j\neq i} s(\boldsymbol{z}_i^2, \boldsymbol{z}_j^2).} \tag{8}$$

Here s is the exponential of similarity between representations $s(\boldsymbol{z}, \boldsymbol{z}') = \exp(\cos(g_{\theta'}(\boldsymbol{z}), g_{\theta'}(\boldsymbol{z}'))/\tau)$, $g_{\theta'}$ is a fully connected layer with a parameter θ', and τ is a hyper-parameter.

We note that by omitting the second summation in the denominator of Eq. (7) or (8) we obtain Eq. (3). This opens a way to bridge the two losses for supervised metric learning and unsupervised contrastive learning.

IV. Proposed Method

This section presents 1) Generalized contrastive loss (GCL) and 2) GCL for semi-supervised learning. GCL unifies losses from two different learning frameworks, supervised metric learning and unsupervised contrastive learning, and thus it naturally works as a loss function for semi-supervised learning.

A. Generalized Contrastive Loss

Let $Z = \{\boldsymbol{z}_i^k : i = 1, 2, \cdots, N, k = 1, 2\}$ be a representation batch obtained from a mini-batch for either supervised metric learning or unsupervised contrastive learning (see Step 4 in Sec. III-A and Sec. III-B). We define the GCL as

$$L_\alpha = \frac{1}{2N}\sum_{i,k} \log \frac{\sum_{j,l}\langle\alpha_{ij}^{kl}\rangle s(\boldsymbol{z}_i^k, \boldsymbol{z}_j^l)}{\sum_{j,l}|\alpha_{ij}^{kl}| s(\boldsymbol{z}_i^k, \boldsymbol{z}_j^l) + \epsilon}, \tag{9}$$

where α_{ij}^{kl} is a fourth-order affinity tensor, $\langle\cdot\rangle$ denotes the application of Macaulay brackets to the ramp function as $\langle a\rangle = \max(0, a)$, and $\epsilon \simeq 0$ is a constant to avoid a division by zero. Note that a positive value for α_{ij}^{kl} predisposes two representations $\boldsymbol{z}_i^k$ and $\boldsymbol{z}_j^l$ to be close to each other, a negative value does the opposite. The episode loss can be viewed as a special case of GCL when Z is made from a mini-batch of labeled samples via prototypes, as shown in Sec. III-A, and the affinity tensor is defined by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k < l, i = j) \\ -1 & (k < l, i \neq j) \\ 0 & (\text{otherwise}) \end{cases}. \tag{10}$$

Note that i is the category index and k is the sample index in this case.

The NT-Xent loss can also be viewed as a special case of GCL when Z is made from a mini-batch of unlabeled samples using augmentation, as shown in Sec. III-B, and the affinity tensor is defined by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k \neq l, i = j) \\ 0 & (k = l, i = j) \\ -1 & (\text{otherwise}) \end{cases}. \tag{11}$$

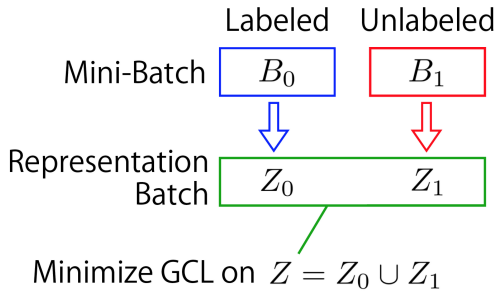

Fig. 1. Semi-supervised learning using generalized contrastive loss (GCL). From a given mini-batch (B_0, B_1), which includes both labeled and unlabeled samples, a representation batch $Z = Z_0 \cup Z_1$ is constructed. Z_0 is constructed in the same way as in supervised metric learning, for example, with anchors and prototypes. Z_1 is constructed in the same way as in unsupervised contrastive learning, for example, with data augmentation functions.

Note that i is the sample index and k is the augmentation type index in this case.

Other types of losses, including generalized end-to-end loss [22] and angle-prototypical loss [4], can also be obtained by changing the definitions of Z, α, and s. Note that the complete definition of GCL includes more instances of metric learning methods, as discussed in the Appendix.

B. GCL for Semi-Supervised Learning

In semi-supervised learning, a training dataset includes both labeled and unlabeled samples. Thus, a mini-batch is given by a pair $B = (B_0, B_1)$ of a set of labeled samples B_0 and a set of unlabeled samples B_1. To apply GCL to B, its representation batch is constructed by $Z = Z_0 \cup Z_1$, where

$$Z_0 = \{\boldsymbol{z}_{i|0}^k : i = 1, 2, \cdots, N, k = 1, 2\} \tag{12}$$

is a representation batch of B_0 given from a supervised metric learning method and

$$Z_1 = \{\boldsymbol{z}_{i|1}^k : i = 1, 2, \cdots, N', k = 1, 2\} \tag{13}$$

is a representation batch of B_1 given from an unsupervised contrastive learning method, as shown in Figure 1.

The GCL for semi-supervised learning is then defined on Z by

$$L_\alpha = \sum_{i,k,u} \log \frac{\sum_{j,l,v}\langle\alpha_{ij|uv}^{kl}\rangle s(\boldsymbol{z}_{i|u}^k, \boldsymbol{z}_{j|v}^l)}{\sum_{j,l,v}|\alpha_{ij|uv}^{kl}| s(\boldsymbol{z}_{i|u}^k, \boldsymbol{z}_{j|v}^l),} \tag{14}$$

where $u, v \in \{0, 1\}$ denote labeled or unlabeled samples. Note that affinity tensor $\alpha_{ij|uv}^{kl}$ becomes a sixth-order tensor to predispose similarity between $\boldsymbol{z}_{i|u}^k$ and $\boldsymbol{z}_{j|v}^l$ to be close or far.

Here, we provide an example definition of $\alpha_{ij|uv}^{kl}$ for semi-supervised learning. Compared with NT-Xent loss, we relax the affinity between unlabeled samples because some labeled

TABLE I
RESULTS OF SEMI-SUPERVISED, UNSUPERVISED, AND SUPERVISED LEARNING. EQUAL ERROR RATE (EER) ON THE VOXCELEB 1 TEST IS REPORTED.

Method	Training Scenario	Additional Data/Model	EER (%)
SSL embedding [21]	Semi-supervised	Speech recognition	6.31
Ours	Semi-supervised	-	**6.01**
Cross-modal [19]	Unsupervised	Video (face images)	20.09
Ours	Unsupervised	-	**15.26**
AM-Softmax	Supervised	-	**1.81**
Ours	Supervised	-	2.56

samples are available for training.

$$\alpha^{kl}_{ij|00} = \begin{cases} 1 & (k \neq l, i = j) \\ 0 & (k = l, i = j) \\ -1 & (\text{otherwise}) \end{cases} \tag{15}$$

$$\alpha^{kl}_{ij|11} = \begin{cases} 1 & (k \neq l, i = j) \\ 0 & (k = l, i = j) \\ -1 & (\text{otherwise}) \end{cases} \tag{16}$$

$$\alpha^{kl}_{ij|01} = -1 \tag{17}$$

$$\alpha^{kl}_{ij|10} = -1. \tag{18}$$

This definition is effective for semi-supervised learning for speaker verification, where labeled utterances are from a predefined set of speakers and unlabeled utterances are from another (different) set of unknown speakers. For the similarity measure, we use $s(\boldsymbol{z}, \boldsymbol{z}') = \exp(\gamma \cos(\boldsymbol{z}, \boldsymbol{z}') + \beta)$. This definition is used in [4].

V. EXPERIMENTS

A. Experimental Settings

We used the VoxCeleb dataset [23], [24] for evaluating our proposed framework. The training set (voxceleb_2_dev) consists of 1,092,009 utterances of 5,994 speakers. The test set (voxceleb_1_test) consists of 37,611 enrollment-test utterance pairs. The equal error rate (EER) was used as an evaluation measure.

For semi-supervised learning experiments, we randomly selected P speakers from the set of 5,994. We used their labeled samples and the remaining unlabeled samples for training. This is the same evaluation setting proposed in [21]. For unsupervised learning experiments, we did not use speaker labels. This evaluation setting is more difficult than the cross-modal self-supervised setting in [25] because we did not use videos (face images) for training. For supervised learning experiments, we used all labeled samples for training. This is the official evaluation setting on the VoxCeleb dataset.

We used the ResNet18 convolutional network with an input of 40-dimensional filter bank features. For data augmentation to construct a representation batch from unlabeled samples, we used four Kaldi data augmentation schemes with the MUSAN (noise, music, and babble) and the RIR (room impulse response) datasets. For semi-supervised learning, 10 % of samples in each mini-batch were unlabeled and the others were labeled.

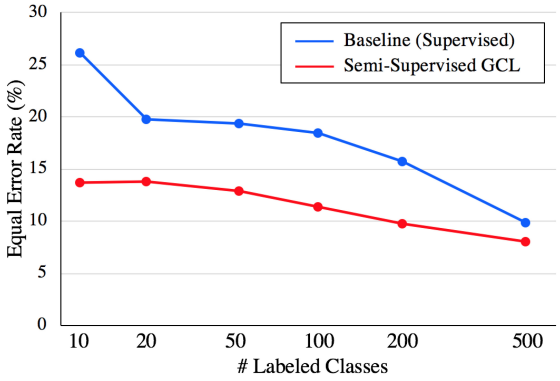

Fig. 2. Results for semi-supervised experiments. The equal error rate on the VoxCeleb 1 test set is reported. The baseline uses only labeled samples. Semi-supervised GCL uses both labeled and unlabeled samples.

B. Results

Table I summarizes EERs for semi-supervised, unsupervised, and supervised learning settings. The results demonstrate that GCL enables the learning of speaker embeddings in the three different settings without any changes in the definition of the loss function.

For semi-supervised learning experiments, we compared the results with those of [21] by using the same number of labeled speakers ($P = 899$). The results show that our framework achieves comparable performance. Note that the method in [21] uses an automatic speech recognition model pre-trained on another dataset, but we did not use such pre-trained models. Comparison with a supervised learning method is shown in Figure 2. We see that adding unlabeled utterances improved the performance, in particular when the number of available labeled utterances was small.

For unsupervised learning experiments, our method outperformed the cross-modal self-supervised method in [19]. Note that our method did not use any visual information, such as face images, for supervision. Audio-visual unsupervised learning with our framework is promising as a next step.

For supervised learning experiments, our method achieves a 2.56 % EER without using data augmentation. However, there is still room for improvement, because training the same network with Softmax and AM-Softmax losses (training with Softmax and fine-tuning with AM-Softmax) achieves a 1.81 % EER. Introducing a more effective network structure, such as ECAPA-TDNN [26] and AutoSpeech-NAS [25], to our framework would be also interesting as future work.

VI. CONCLUSION

This paper proposed a semi-supervised contrastive learning framework with GCL. We showed via experiments on the VoxCeleb dataset that the proposed GCL enables a network to learn speaker embeddings in three manners, namely, supervised learning, semi-supervised learning, and unsupervised learning. Furthermore, this was accomplished without making any changes to the definition of the loss function.

ACKNOWLEDGMENT

This work was partially supported by the Japan Science and Technology Agency, ACT-X Grant JPMJAX1905, and the Japan Society for the Promotion of Science, KAKENHI Grant 19K22865.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Proceedings of the International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*, pages 84–92, 2015.

[3] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 4077–4087, 2017.

[4] 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. *arXiv preprint arXiv:2003.11982*, 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[6] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019.

[8] 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 *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274, 2018.

[9] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 212–220, 2017.

[10] Yutong Zheng, Dipan K. Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5089–5097, 2018.

[11] Yi Liu, Liang He, and Jia Liu. Large Margin Softmax Loss for Speaker Verification. In *Proceedings of Interspeech*, 2019.

[12] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, vol. 28, no. 2, pages 129–137, 1982.

[13] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–84, 2016.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations*, 2018.

[15] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations*, 2019.

[16] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 15509–15519, 2019.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[18] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[19] Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman. Disentangled speech embeddings using cross-modal self-supervision. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6829–6833, 2020.

[20] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1163–1171, 2016.

[21] Themos Stafylakis, Johan Rohdin, Oldrich Plchot, Petr Mizera, and Lukas Burget. Self-supervised speaker embeddings. *Proceedings of Interspeech*, 2019.

[22] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883, 2018.

[23] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Proceedings of Interspeech*, 2017.

[24] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Proceedings of Interspeech*, 2018.

[25] Shaojin Ding, Tianlong Chen, Xinyu Gong, Weiwei Zha, and Zhangyang Wang. Autospeech: Neural architecture search for speaker recognition, *arXiv preprint arXiv:2005.03215*, 2020.

[26] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

APPENDIX

The complete form of the proposed GCL is defined over a representation batch $Z = \{\boldsymbol{z}_i^k : i = 1, 2, \cdots, N, k = 1, 2, \cdots, N\}$ by

$$L = -\frac{1}{KN} \sum_{i,k} \Psi \left(\sum_{j,l} s(\boldsymbol{z}_i^k, \boldsymbol{z}_j^l; \alpha_{ij}^{kl}) \right), \tag{19}$$

where α_{ij}^{kl} is an affinity tensor, $s(\boldsymbol{z}, \boldsymbol{z}'; \alpha)$ is the similarity between $\boldsymbol{z}$ and $\boldsymbol{z}'$ given an affinity value α, and Ψ is a normalization or clipping function.

Table II summarizes how to obtain popular loss functions from GCL. We hope this provides an overview of recent progress and helps other researchers develop new unsupervised, semi-supervised, and supervised learning methods.

A. Affinity Type

Four types of affinity tensor definitions are used in Table II. With all of them, a positive value for α_{ij}^{kl} predisposes two representations $\boldsymbol{z}_i^k$ and $\boldsymbol{z}_j^l$ to be close to each other, a negative value does the opposite. The density of α_{ij}^{kl} increases in the order of Types 1 to 4, as shown in Figure 3. Definitions of the types are given below. Note that $K = 2$ is assumed for simplicity.

Type 1 makes pairs $(\boldsymbol{z}_i^1, \boldsymbol{z}_j^2)$ and its output is 1 if two samples are from the same class (i.e., $i = j$) and -1 if two samples are from different classes (i.e., $i \neq j$). An example definition of this type is given by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k < l, i = j) \\ -1 & (k > l, i = j - 1 \bmod N) \\ 0 & (\text{otherwise}) \end{cases}. \tag{20}$$

Type 2 makes triplets $(\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, \boldsymbol{z}_j^2)$ where $i \neq j$. With respect to an anchor $\boldsymbol{z}_i^1$, $\boldsymbol{z}_i^2$ is marked as positive and $\boldsymbol{z}_j^2$ is marked

TABLE II

COMPARISON OF RECENT LOSS DEFINITIONS IN GCL FORMULATION. THE AFFINITY TENSOR MAKES PAIRS, TRIPLETS, $(N+1)$-TUPLES, OR $2N$-TUPLES, AS SHOWN IN FIGURE 3. REPRESENTATION BATCH Z IS CONSTRUCTED FROM LABELED SAMPLES, UNLABELED SAMPLES, AND/OR PARAMETERS. SEE THE DEFINITION OF GCL IN SEC. IV FOR THE MEANING OF s, $\tilde{\alpha}$, AND $\Psi(v)$. m IS A MARGIN HYPER-PARAMETER, AND $M = \sum_{j,l} s(\boldsymbol{z}_i^k, \boldsymbol{z}_j^l; \alpha_{ij}^{kl}) + \epsilon$.

Loss	Affinity	Representation Batch Z	Similarity $s(\boldsymbol{z}, \boldsymbol{z}'; \alpha)$	$\tilde{\alpha}$	$\Psi(v)$
Contrastive loss [6]	Type 1	Labeled	$\alpha d(\boldsymbol{z}, \boldsymbol{z}')$	α	$-(\langle v \rangle - \chi(v<0)\langle v+m \rangle)/2$
Triplet loss [2]	Type 2	Labeled	$\alpha d(\boldsymbol{z}, \boldsymbol{z}')$	α	$-\langle v+m \rangle$
ArcFace (AAM loss) [7]	Type 3	Labeled+weights	$\lvert\alpha\rvert \exp(\cos(\angle(\boldsymbol{z}, \boldsymbol{z}') + m\langle\alpha\rangle))$	$\langle\alpha\rangle$	$2\log(v/M)$
SphereFace [9]	Type 3	Labeled+weights	$\lvert\alpha\rvert \exp((1+m\langle\alpha\rangle)\cos(\boldsymbol{z}, \boldsymbol{z}'))$	$\langle\alpha\rangle$	$2\log(v/M)$
CosFace [8]	Type 3	Labeled+weights	$\lvert\alpha\rvert \exp(\cos(\boldsymbol{z}, \boldsymbol{z}') - m\langle\alpha\rangle)$	$\langle\alpha\rangle$	$2\log(v/M)$
Prototypical episode loss [3]	Type 3	Labeled	$\lvert\alpha\rvert \exp(-d(\boldsymbol{z}, \boldsymbol{z}'))$	$\langle\alpha\rangle$	$2\log(v/M)$
Angle-prototypical loss [4]	Type 3	Labeled	$\lvert\alpha\rvert \exp(\gamma\cos(\boldsymbol{z}, \boldsymbol{z}') + \beta)$	$\langle\alpha\rangle$	$2\log(v/M)$
SimCLR (NT-Xent loss) [5]	Type 4	Unlabeled	$\lvert\alpha\rvert \exp(\cos(g(\boldsymbol{z}), g(\boldsymbol{z}'))/\tau)$	$\langle\alpha\rangle$	$\log(v/M)$
Our experimental setting	Type 4	Labeled+unlabeled	$\lvert\alpha\rvert \exp(\gamma\cos(\boldsymbol{z}, \boldsymbol{z}') + \beta)$	$\langle\alpha\rangle$	$\log(v/M)$

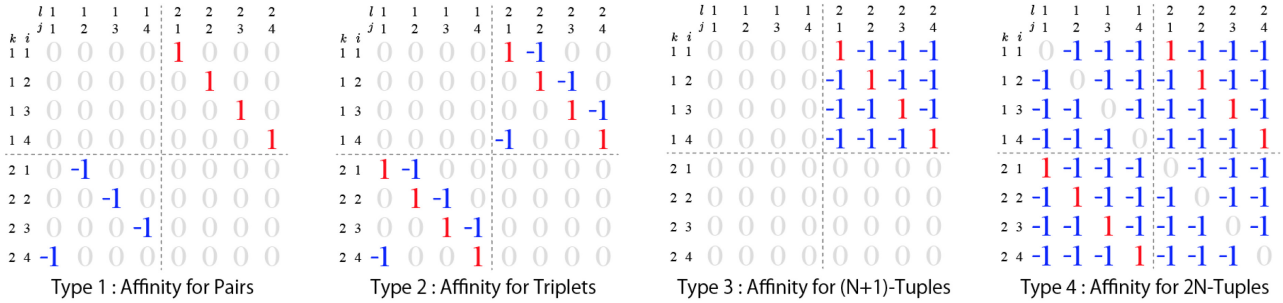

Fig. 3. Four types of the affinity tensor α_{ij}^{kl}. The values 1 and -1 denote representation pairs predisposed to be close to and far from each other, respectively. The diagonal 0 values are for anchors, and the other 0 values make no restriction on sample pairs.

as negative. An example definition of this type is given by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k \neq l, i = j) \\ -1 & (k \neq l, i = j-1 \bmod N) \\ 0 & (\text{otherwise}) \end{cases}. \tag{21}$$

Type 3 makes $(N+1)$-tuples $(\boldsymbol{z}_i^1, \boldsymbol{z}_1^2, \cdots, \boldsymbol{z}_N^2)$. With respect to an anchor $\boldsymbol{z}_i^1$, $\boldsymbol{z}_i^2$ is marked as positive and all the others are marked as negative. The definition of this type is given by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k < l, i = j) \\ -1 & (k < l, i \neq j) \\ 0 & (\text{otherwise}) \end{cases}. \tag{22}$$

Type 4 makes $2N$-tuples $(\boldsymbol{z}_i^1, \boldsymbol{z}_1^2, \cdots, \boldsymbol{z}_N^2, \boldsymbol{z}_1^1, \cdots, \boldsymbol{z}_{i-1}^1, \boldsymbol{z}_{i+1}^1, \boldsymbol{z}_N^1)$. With respect to an anchor $\boldsymbol{z}_i^1$, $\boldsymbol{z}_i^2$ is marked as positive and all the others are marked as negative. The definition of this type is given by

$$\alpha_{ij}^{kl} = \begin{cases} 1 & (k \neq l, i = j) \\ 0 & (k = l, i = j) \\ -1 & (\text{otherwise}) \end{cases}. \tag{23}$$

B. Representation batch

Table II gives three types of definition for the representation batch $Z = \{\boldsymbol{z}_i^k : i = 1, 2, \cdots, N, k = 1, 2\}$.

Labeled: With labeled samples for supervised learning, $\boldsymbol{z}_i^k$ denotes the k-th representation from class i. A representation $\boldsymbol{z}_i^k$ is defined by sample representation $\boldsymbol{z}_i^k = f_\theta(\boldsymbol{x}_i^{k'})$ or a statistical representation, such as a mean representation (prototype) computed from some samples in $B' \subset B$, specifically, $\boldsymbol{z}_i^k = \frac{1}{|B'|}\sum_{k' \in B'} f_\theta(\boldsymbol{x}_i^{k'})$. Here, $B = \{(\boldsymbol{x}_i^{k'}, y_i) : i = 1, 2, \cdots, N, k' = 1, 2, \cdots, K'\}$ is a mini-batch of labeled samples.

Labeled+weights: This type uses parameters as prototypes, where $\boldsymbol{z}_i^1 = f_\theta(\boldsymbol{x}_i^1)$ is a representation from class i and $\boldsymbol{z}_i^2 = \boldsymbol{w}_i$ is a weight parameter for class i.

Unlabeled: With unlabeled samples for unsupervised learning, $\boldsymbol{z}_i^k$ denotes the representation of the i-th sample with the k-th augmentation. With this type, prototypes can also be introduced in the same way as prototypes are constructed for labeled samples, that is, by taking the mean of representations from more than one augmentation function.