Raw waveform speaker verification for supervised and self-supervised learning

Jee-Weon Jung\textsuperscript{1,*}, You Jin Kim\textsuperscript{1,*}, Hee-Soo Heo\textsuperscript{1}, Bong-Jin Lee\textsuperscript{1}, Youngki Kwon\textsuperscript{1}, Joon Son Chung\textsuperscript{2}

\textsuperscript{1}Naver Corporation, South Korea, \textsuperscript{2}KAIST, South Korea

\{jeeweon.jung, youjin.kim117\}@navercorp.com, joonson@kaist.ac.kr

Abstract

Speaker verification models that directly operate upon raw waveforms are receiving growing attention. However, their performances are less competitive than the state-of-the-art hand-crafted feature-based counterparts, demonstrating equal error rates under 1\% on the benchmark VoxCeleb1 evaluation protocol. In addition, they have yet not been explored with self-supervised learning frameworks. This paper proposes a new raw waveform speaker verification model that incorporates techniques proven effective for speaker verification, including the Res2Net backbone module and the aggregation method considering both context and channels. Under the best performing configuration, the model shows an equal error rate of 0.89\%, competitive with state-of-the-art models. We also explore the proposed model with a self-supervised learning framework and show the state-of-the-art performance in this line of research. Finally, we show that leveraging the model trained with self-supervision successfully serves as a pre-trained model under the semi-supervised scenario where it is assumed that only a limited amount of data has a ground truth label and a bigger data has no label.

Index Terms: raw waveform, speaker verification, self-supervised learning

1. Introduction

In audio signal processing tasks, models typically input hand-crafted features (e.g., mel-filterbanks or mel-frequency cepstral coefficients) extracted from raw waveforms [1–7]. Models that directly operate upon raw waveforms are becoming more familiar with recent advances in deep learning and audio signal processing techniques. In the existing works, raw waveforms are processed mainly by either vanilla convolutional layer or parameterised filterbank layers [8–11].

The trend can also be observed in the speaker verification literature. Starting with the first raw waveform speaker verification model that employs a VGGNet-styled vanilla convolutional layer [12], several studies now adopt direct modelling of raw waveforms [13–17]. However, raw waveform speaker verification models suffer from less competitive performance in supervised learning. Even the latest architecture demonstrates EER of 1.29 \% [17], whereas the widely adopted ECAPA-TDNN architecture and its variants have consistently reported EERs under 1\% [2, 18, 19].

Meanwhile, self-supervised learning has arisen as an alternative to the currently dominant supervised learning, which involves human-annotated ground truth labels. Various frameworks in self-supervised learning train the model without ground truth labels (i.e., unlabelled), leveraging pretext tasks which can be solved by learning the inherent data properties [20]. Most recent prevailing frameworks in self-supervised learning take the form of either contrastive- [8, 21, 22], reconstruction- [23, 24], or non-negative (positive pairs only)-based learning [25–27].

Self-supervised learning is also an ongoing active field of research in speaker verification [28–32]. However, none of these works has yet adopted a model architecture that directly operates upon raw waveforms, leaving the potential open.

In this paper, we contribute to various aspects of the raw waveform speaker verification literature by exploring the above-mentioned issues that the literature requires. Concretely, we make the below contributions:

- we propose a new raw waveform speaker verification architecture, namely RawNet3, that demonstrates EER under 1\% in the VoxCeleb1 evaluation protocol;
- we explore raw waveform speaker verification model, for the first time, with self-supervised learning and demonstrate state-of-the-art performance;
- we show the efficacy of self-supervised learning pre-training under semi-supervised learning scenario.

2. Proposed architecture

Figure 1 illustrates the proposed model architecture, named RawNet3 for brevity. The architecture is in a hybrid form of RawNet2 [33] and ECAPA-TDNN [2]. We first apply pre-emphasis to the raw waveform following [34] and feed it through an instance normalisation layer [35]. Then the output
is processed into a time-frequency representation using parameterised analytic filterbanks [11] where complex-valued filterbanks are learned. This layer is an extension of sinc-convolution layer [10], which has been adopted in RawNet2 [33], where real-valued parameterised filterbanks are learned. At this step, the extent of sequence compression is configured via the stride size, where a smaller stride slows processing but produces better performance. In contrast, a bigger stride delivers faster processing but less competitive performance. The default kernel length and stride configuration are 251 and 48, respectively, generating approximately 15 ms window and 3 ms shift per frame.

Three backbone blocks with residual connections then position where their outputs are concatenated identical to the ECAPA-TDNN architecture. We also input the summation of the first and second block outputs to the third block. This is similar to but slightly different from the ECAPA-TDNN that inputs all previous block outputs as the following block input. We apply max pooling of sizes 5 and 3 to the first two backbone blocks, different to ECAPA-TDNN, following the RawNet2 architecture. Additional reduction of sequence length is mandatory in raw waveform speaker verification models. This is because the sequence length is extremely longer than the handcrafted feature-based model counterparts. The adoption of max pooling is based on our previous empirical results that it mitigates overfitting in raw waveform speaker verification models [36]. Each backbone block, illustrated in Figure 2 and referred to as AFMS-Res2MP-block, is based on Res2Net [37]. It is similar to the backbone block of the ECAPA-TDNN with two modifications: (i) we adopt AFMS used in RawNet2 backbone block in place of squeeze-excitation [38] based on previous empirical results, and (ii) we optionally apply max pooling before applying AFMS.

After the three backbone AFMS-Res2MP-blocks, one convolution layer with batch normalisation [39] positions, identical to the ECAPA-TDNN. For supervised learning with classification loss (Section 3.1), the classification head exists as the output layer. For self-supervised learning with the DINO framework (Section 3.2), additional DINO head layers are added.

**Comparison with RawNet2 architecture.** Several different design choices have been adopted in RawNet3 compared to the previous RawNet2 architecture. First, parameterised analytic filterbank layer [11] is utilised instead of sinc-convolution layer [10]. Second, log and mean normalisation is applied to analytic filterbank output. Third, the number of backbone blocks and their connections have been adapted following the ECAPA-TDNN alike topology. Last, the channel and context-dependent statistic pooling replaces a uni-directional gated recurrent unit layer [40].

### 3. Adopted frameworks

#### 3.1. Supervised learning: classification with AAM-softmax

The two most prevailing supervised learning frameworks in speaker verification are classification and metric learning. We adopt the classification-based training framework. In the classification framework, a model is trained as a closed set speaker identification model with a classification head that has a dimensionality equal to the number of speakers in the trainset (i.e., the d-vector framework [41]). A categorical cross-entropy objective function is adopted, which calculates the loss by comparing the softmax non-linearity applied classification head and the one-hot ground truth label. After the training is complete, the classification head is removed, and the model is used as a speaker embedding extractor.

Specifically, we adopt the AAM-softmax objective function [42], also known as the ArcFace. The AAM-softmax can enforce larger gaps between the nearest speakers. Let \( x_i \), \( y_i \), and \( W \) be the speaker embedding, its corresponding label, and the weight matrix of the classification head where \( i \) is the index of an utterance within a mini-batch of size \( N \), \( 1 < i < N \). The AAM-softmax loss is formulated as:

\[
L_A = \frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{e^{s \cos(\theta_{v,i})}}{\sum_{j=1, j \neq y_i}^{N} e^{s \cos(\theta_{v,i})}} \right) + \sum_{j=1}^{N} e^{s \cos(\theta_{v,i})}
\]

where \( s \) is the hyper-parameter for a scale factor and \( \cos(\theta_{v,i}) \) is the dot product between \( i \) normalised \( x_i \) and \( W_j \). Readers are guided to [42] for full details.

#### 3.2. Self-supervised learning: DINO

The DINO framework [26] is one of the most competitive frameworks in the recent self-supervised learning literature. It compares multiple different views generated from a single utterance like the BYOL [25] framework and employs a self-distillation framework similar to the data2vec [27] framework.

DINO involves a teacher and a student network with an identical architecture but different parameters. Multi-crop training is utilised where a set of \( V \) views, two different global \( \{a_i\} \) and \( \{a_j\} \) and several local views \( \{a'\} \), of an utterance are exploited. The teacher digests only global views, whereas the student digests all views. Only the parameters of the student network are updated using the loss function. The parameters of the teacher network are updated using an exponential moving average of the student networks.

In the DINO framework, sharpening the output with individual temperatures is applied to both teacher and student networks to avoid representation collapse during training. Centering is additionally applied to the teacher output for the same purpose, which subtracts center from the teacher output. The centre is updated using an exponential moving average. Centering uniformly flattens the probability distribution while sharpening narrow-downs the width of the distribution. It is argued that a good balance between centering and sharpening hyperparameters leads to successful training without requiring architecture differentiation. The DINO loss is then defined as:

\[
L_D = \sum_{a \in \{a_i, a'_j\}} \sum_{a' \in V \setminus a \neq a} H(P_t(a), P_s(a'))
\]

where \( P_t \) and \( P_s \) correspond to teacher and student network outputs and \( H(\cdot) \) is the cross-entropy. Readers are guided to [26] for full details.
We present three sets of experiments using the proposed RawNet3 model under various frameworks: (i) supervised learning using ground truth label-based classification (Table 1); (ii) self-supervised learning using the DINO framework (Table 2); and additionally, (iii) semi-supervised (Table 3). In the semi-supervised scenario, we first pre-train the model using the DINO self-supervised learning framework. We then fine-tune the model using supervised learning with ground truth label-based classification. We also provide comparison with recent models in the literature in Tables 4 and 5.

4.1. Dataset

We use the VoxCeleb 1 & 2 datasets [3, 43] throughout this paper, in which speakers of both datasets involve various nationalities and are collected in the wild scenario, involving various background noises. The two datasets each include 1,251 and 6,112 speakers’ utterances of more than 340 and 2,440 hours of speech. VoxCeleb1 is divided into two subsets: the development set, which involves 1,211 speakers and the evaluation set, which involves 11 speakers. VoxCeleb2 is also divided into two subsets: the development set, which involves 5,994 speakers and the evaluation set, which involves 40 speakers. VoxCeleb2 is also divided into two subsets: the development set, which involves 5,994 speakers and the evaluation set, which involves 40 speakers.

4.2. Data configurations and metrics

For supervised learning experiments, we employ the development sets of the VoxCeleb1&2 datasets. For self-supervised learning experiments, the development set of VoxCeleb2 is used. For fine-tuning the self-supervised pre-trained model, i.e., the semi-supervised scenario, we pre-train the model using the VoxCeleb2 development set and fine-tune the model using the VoxCeleb1 development set. The VoxCeleb1-O benchmark evaluation protocol that involves the VoxCeleb1 test set is used to measure the performance for all experiments.

The widely adopted EER (%) is the primary metric. We also report performances using the minimum detection cost function (minDCF) metric. For the minDCF, hyper-parameters of $P_{target}=0.05$ and $C_{falsealarm} = C_{miss}=1$ are adopted. Lower values depict superior performances for both metrics.

4.3. Implementation details

Common. All experiments have been conducted using the PyTorch Python library with four Nvidia V100 GPUs. The model architecture, training recipe and pre-trained model weights will be made freely accessible.

Supervised learning. The learning rate is scheduled between $10^{-3}$ and $6 \cdot 10^{-5}$ and restarts every eight epochs. The model is trained for 40 epochs. AAM-softmax has a margin $m$ of 0.3 and scale $s$ of 30. The model is trained using randomly cropped 3 seconds utterances. The size of a mini-batch is 128. We apply batch augmentation, similar to that of [46] where augment methods include waveform masking, speed, noise and reverberation addition.

Self-supervised learning. The learning rate is scheduled between $10^{-3}$ and $10^{-5}$ and restarts every 16 epochs. The model is trained for 80 epochs. The size of a mini-batch is 400 unless mentioned otherwise. We adopt two global views of 4 seconds for the DINO framework and five local views of 2 seconds. Augmentation involves noise and reverberations with additional data curriculum augmentation [1]. Temperatures for the teacher and the student RawNet3 models are 0.04 and 0.1, respectively. Momentum values for the teacher model and centre update are 0.987 and 0.9, respectively.

Table 1: Results on supervised learning using the AAM-softmax [42] objective function. Trained on VoxCeleb1&2 development sets. The two numbers in Hz denote frame resolutions after the first parameterised filterbank and the last max pooling layer.

| Configurations | EER(%) | minDCF |
|----------------|--------|--------|
| RawNet2 [33]   | 2.48   | N/R    |
| RawNet3 (stride=48) | 1.05 | 0.0763 |
| → param fbank log | 1.27  | 0.0852 |
| → param fbank norm | 1.22  | 0.0838 |
| → param fbank log&norm | 1.23  | 0.0927 |
| → ch&context stat pool | 1.45   | 0.0975 |
| → stride=10, 1600Hz→106Hz | 0.89 | 0.0639 |
| → stride=16, 1000Hz→66Hz | 0.90 | 0.0593 |
| → stride=24, 666Hz→44Hz | 0.96 | 0.0773 |
| → stride=64, 250Hz→16Hz | 1.11 | 0.0851 |
| → stride=96, 166Hz→11Hz | 1.31 | 0.0937 |

Table 2: Results on self-supervised learning using the DINO [26] framework. Trained on VoxCeleb&2 development set.

| Configurations | EER(%) | minDCF |
|----------------|--------|--------|
| RawNet3        | 5.74   | 0.3507 |
| → param fbank log | 10.46  | 0.5775 |
| → param fbank mean norm | 8.87  | 0.4969 |
| → param fbank log&mean norm | 9.98  | 0.5386 |
| → DINO temp warm-up | 5.89 | 0.4004 |
| → DINO last layer norm | 5.40 | 0.3396 |
| → DINO T momentum 0.99 | 6.17 | 0.3987 |
| → half batch size (400→1200) | 6.87 | 0.4513 |

Table 3: Results on fine-tuning the pre-trained model. Trained on VoxCeleb1 development set.

| Configurations | EER(%) | minDCF |
|----------------|--------|--------|
| RawNet3 (w/ pre-train) | 2.18  | 0.1519 |
| RawNet3 (w/o pre-train) | 2.98  | 0.2268 |

1 will be available in https://github.com/Jungjee/RawNet and https://github.com/clovaai/voxceleb_trainer prior to publication.
Table 4: Comparison with recent literature of supervised and self-supervised speaker verification.

| Framework | In Feat | EER(%) | minDCF |
|-----------|---------|--------|--------|
| Desplanques et al. [2] | MFCC | 0.87 | 0.1066 |
| Ravanelli et al. [18] | Fbank | 0.69 | N/R |
| Kuzmin et al. [19] | Fbank | **0.66** | **0.0640** |
| Zhu et al. [13] | Waveform | 2.60 | 0.2390 |
| Li et al. [15] | Waveform | 2.51 | N/R |
| Lin et al. [16] | Waveform | 1.95 | 0.2030 |
| Kim et al. [17] | Waveform | 1.29 | 0.1420 |
| **Ours** – stride=10 | Waveform | **0.89** | **0.0659** |
| **Ours** – stride=16 | Waveform | **0.90** | **0.0593** |

5. Results

5.1. Supervised learning

Table 1 delivers the result of RawNet3 architecture under supervised learning with ground truth label-based classification. Compared to the previous RawNet2 architecture, RawNet3 demonstrates superior performance with EER reduced from 2.48% to 1.05%, showing over 57% improvement relative.

Rows 3 to 6 report ablation experiments on the RawNet3 architecture by excluding modified components. Exclusion of either logarithm or mean subtraction normalisation degrades the performance. Channel and context statistical pooling had more impact as a single component, degrading the EER to 1.45% when excluded.

The last five rows demonstrate a trade-off between computation and performance via different stride sizes in the parameterised filterbank layer. Smaller stride size generates longer sequence frames that bring further performance improvement at the cost of more computation. On the other hand, a larger stride size generates shorter sequence frames, resulting in less computation at the cost of performance degradation. With a stride size of 10, the EER reduced to 0.89%.

5.2. Self-supervised learning

Table 2 delivers the result of the proposed RawNet3 architecture under the DINO self-supervised learning framework. The result in the first row shows an EER of 5.74% without utilising any ground truth label. Rows from 2 to 4 show the results of excluding either logarithm, mean normalisation, or both after the first parameterised filterbank layer. Compared to supervised learning, logarithm and mean normalisation affected the performance significantly. We analyse that this phenomenon is related to the property of the DINO framework, which does not involve negative pairs. Because comparison is only made among the same utterances with different augmentations, robustness towards channel variation can be weaker, which the logarithm and mean normalisation offer.

Rows 5 and 6 show ablations by testing the officially recommended hyper-parameter changes. Both hyper-parameter tended to act differently than anticipated. Applying warm-up did not affect the performance noticeably, and normalising the last layer (no normalisation is expected to perform better) further improved the EER to 5.40%. The last two rows present additional ablation experiments by changing the teacher network’s momentum value and halving the batch size.

Table 5: Comparison with self-supervised learning models.

| Framework | In Feat | EER(%) | minDCF |
|-----------|---------|--------|--------|
| Huh et al. [28] | AP+AAT | 8.65 | 0.4540 |
| Xia et al. [29] | MOCO+Wav-Aug(ProtoneCE) | 8.23 | 0.5900 |
| Mun et al. [30] | CEL | 8.01 | N/R |
| Tao et al. [31] | Contrastive | 7.36 | N/R |
| Sang et al. [32] | SsReg | 6.99 | 0.4340 |
| **Ours** | DINO | **5.40** | **0.3396** |

5.3. Semi-supervised learning

Supervised learning. Table 4 compares the RawNet3 trained with ground truth labels with recent handcraft feature- and raw waveform-based models [2, 13, 15–19]. Compared to the recent state-of-the-art models, our RawNet3 shows a competitive performance of EER 0.89%, whereas EER of the best model [19] is 0.66%. In terms of the minDCF, RawNet3 performed the best with a value of 0.0593. Our RawNet3 outperforms the previous best model among raw waveform speaker verification models by 31% relative (1.29% vs 0.89%).

Self-supervised learning. Table 5 compares the proposed RawNet3 trained using the DINO self-supervised framework with other recent works in the literature. All five mentioned existing works [28–32] can be seen as contrastive learning variants. The DINO framework, which does not adopt negative pairs, demonstrated the best performance, showing potential in this line of frameworks that operate upon positive pairs only.

Relation with iterative clustering. Iterative clustering has been proven effective for self-supervised speaker verification models [31, 47]. Our model can be viewed as the initial model of iterative clustering. We note that our model can also serve as the initial model and can be further improved using iterative clustering methods.

6. Conclusions

This paper proposes RawNet3, a new raw waveform speaker verification model and evaluates it under various scenarios. In supervised learning, the model demonstrates an EER of 0.89%, competitive with state-of-the-art handcrafted feature-based counterparts and shows the lowest minDCF of 0.0593. Using the DINO self-supervision framework, the model also demonstrates state-of-the-art performance with an EER of 5.40%. Utilising the DINO-trained model as pre-training, fine-tuning with a smaller dataset is also effective, showing 25% improvement relative to the model trained with random initialisation.

https://github.com/facebookresearch/dino
7. References

[1] D. Snyder, D. Garcia-Romero, G. Sell et al., “X-vectors: Robust dnn embeddings for speaker recognition,” in Proc. ICASSP, 2018.

[2] B. Desplanques, J. Thienpondt and K. Demuynck, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” Proc. Interspeech, 2020.

[3] J.-S. Chung, A. Nagrani and A. Zisserman, “Voxceleb2: Deep speaker recognition,” Proc. Interspeech, pp. 1086–1090, 2018.

[4] Y. Jung, S. M. Kye, Y. Choi et al., “Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances,” in Proc. Interspeech, 2020.

[5] W. Cai, J. Chen, J. Zhang and M. Li, “On-the-fly data loader and utterance-level aggregation for speaker and language recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1038–1051, 2020.

[6] C. H. You, K. A. Lee and H. Li, “An svm kernel with gmm-supervisor based on the bhattacharyya distance for speaker recognition,” IEEE Signal processing letters, vol. 16, no. 1, pp. 49–52, 2008.

[7] N. Dehak, P. J. Kenny, R. Dehak et al., “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2010.

[8] A. Baevski, Y. Zhou, A. Mohamed and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” Proc. NeurIPS, vol. 33, pp. 12449–12460, 2020.

[9] T. Sainath, R. J. Weiss, K. Wilson et al., “Learning the speech front-end with raw waveform cdlnns,” in Proc. Interspeech, 2015.

[10] H. Bao, L. Dong and F. Wei, “Beit: Bert pre-training of image transformers,” arXiv preprint:2106.08254, 2021.

[11] M. Caron, H. Touvron, I. Misra et al., “Emerging properties in self-supervised vision transformers,” in Proc. ICCV, 2021.

[12] A. Baevski, W.-N. Hsu, Q. Xu et al., “Data2vec: A general framework for self-supervised learning in speech, vision, and language,” arXiv preprint:2202.01355, 2022.

[13] J. Huh, H. S. Heo, J. Kang et al., “Augmentation adversarial training for unsupervised speaker recognition,” in NeurIPS workshop on Self-Supervised Learning for Speech and Audio Processing, 2020.

[14] W. Xia, C. Zhang, C. Weng et al., “Self-supervised text-independent speaker verification using prototypical momentum contrastive learning,” in Proc. ICASSP, 2021, pp. 6723–6727.

[15] S. H. Mun, W. H. Kang, M. H. Han and N. S. Kim, “Unsupervised representation learning for speaker recognition via contrastive equilibrium learning,” arXiv preprint:2010.11433, 2020.

[16] R. Tao, K. A. Lee, R. K. Das et al., “Self-supervised speaker recognition with loss-gated learning,” arXiv preprint:2110.03869, 2021.

[17] M. Sang, H. Li, F. Liu et al., “Self-supervised speaker verification with simple siamese network and self-supervised regularization,” arXiv preprint:2112.04459, 2021.

[18] J.-w. Jung, J.-S. Heo, H.-j. Shim et al., “Improved rawnet with filter-wise rescaling for text-independent speaker verification using raw waveforms,” in Proc. Interspeech, 2020, pp. 1496–1500.

[19] J.-w. Jung, H.-S. Heo, I.-H. Yang et al., “A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification results,” in Proc. ICASSP, 2018.

[20] D. Ulyanov, A. Vedaldi and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” arXiv preprint:1607.08022, 2016.

[21] J.-w. Jung, H.-S. Heo, I.-h. Yang et al., “Avoiding speaker overfitting in end-to-end dnn’s using raw waveform for text-independent speaker verification,” Proc. Interspeech, pp. 3583–3587, 2018.

[22] S.-H. Gao, M.-M. Cheng, K. Zhao et al., “Rés2Net: A new multi-scale backbone architecture,” IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 2, pp. 652–662, 2019.

[23] J. Hu, L. Shen and G. Sun, “Squeeze-and-excitation networks,” in Proc. CVPR, 2018, pp. 7132–7141.

[24] J. Deng, J. Guo, N. Xue and S. Zafeiriou, “Angular margin loss for deep face recognition,” in Proc. ICASSP, 2018, pp. 448–456.

[25] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in NIPS Workshop on Deep Learning, 2014.

[26] E. Variani, X. Lei, E. McDermott et al., “Deep neural networks for small footprint text-dependent speaker verification,” in Proc. ICASSP, 2014, pp. 4052–4056.

[27] J. Deng, J. Guo, N. Xue and S. Zaferiou, “Arcface: Additive angular margin loss for deep face recognition,” in Proc. CVPR, 2019, pp. 4690–4699.

[28] A. Nagrani, J. S. Chung and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” in Proc. Interspeech, 2017.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, 2015.

[30] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” in Proc. ICLR, 2017.

[31] E. Hoffer, T. Ben-Nun, I. Hubara et al., “Augment your batch: Improving generalization through instance repetition,” in Proc. CVPR, 2020, pp. 8129–8138.

[32] D. Cai, W. Wang and M. Li, “An iterative framework for self-supervised deep speaker representation learning,” in Proc. ICASSP, 2021.