SELF-SUPERVISED LEARNING METHOD USING MULTIPLE SAMPLING STRATEGIES FOR GENERAL-PURPOSE AUDIO REPRESENTATION

Ibuki Kuroyanagi[1,2]*, Tatsuya Komatsu*[1]*,*

[1] LINE Corporation, Tokyo, Japan, [2] Nagoya University, Nagoya, Japan

ABSTRACT

We propose a self-supervised learning method using multiple sampling strategies to obtain general-purpose audio representation. Multiple sampling strategies are used in the proposed method to construct contrastive losses from different perspectives and learn representations based on them. In this study, in addition to the widely used clip-level sampling strategy, we introduce two new strategies, a frame-level strategy and a task-specific strategy. The proposed multiple strategies improve the performance of frame-level classification and other tasks like pitch detection, which are not the focus of the conventional single clip-level sampling strategy. We pre-trained the method on a subset of Audioset and applied it to a downstream task with frozen weights. The proposed method improved clip classification, sound event detection, and pitch detection performance by 25 %, 20 %, and 3.6 %.

***Index Terms*—** contrastive learning, metric learning, pitch shift, sampling strategy

1. INTRODUCTION

Various sound-based applications have been studied, such as speech recognition, speaker identification, and command recognition as speech-related tasks, sound event detection (SED), pitch detection (PD), and instrument estimation as non-speech-related tasks. These tasks have achieved high performance with the development of neural networks [1]–[5]. However, these methods require much effort because they require collecting training data and annotating teacher labels for each task. For audio tagging tasks, fine-tuning with pre-trained models like PANNs [6], which are obtained by supervised training with a large dataset such as Audioset [7], have shown their effectiveness for other audio tagging datasets with small dataset size. However, their applicability is limited to tagging tasks because the models are also trained via supervised training for audio tagging. Task-specific large datasets are required for obtaining pre-training models for other tasks. There is a need to establish training methods for general pre-trained models, which can be trained with unlabeled datasets and applied to various tasks.

Self-supervised learning is gaining attention to acquire useful representations for various tasks using unlabeled data. There are GPT [8] and BERT [9] in natural language processing, CPC [10], SIMCLR [11], MOCO [12] in the field of computer vision, and CURL [13] in the field of reinforcement learning. These methods have achieved high performance in their respective fields. In the field of audio, CPC-based methods design unsupervised loss functions based on regression tasks of representations [14], [15], and wav2vec2.0 solves a contrast task defined based on the quantization of the learned representation [16]. However, many methods have achieved high performance in speech-related tasks, and few studies have focused on non-speech-related tasks. TRILL [17] and [18] are examples of research to obtain general-purpose audio representations for both speech and non-speech-related tasks. These studies sampled anchors, positives, and negatives from unlabeled data. They used a metric learning method with triplet loss to minimize / maximize the distance between the anchor and positive / negative pairs. More recently, COLA [19] achieved higher performance, which is based on self-supervised learning with contrastive loss. COLA samples positive pair as segments from the same audio clip and negative pair as segments from different clips, and maximize / minimize the similarity between positive / negative pairs based on multiclass cross-entropy. COLA has been evaluated on multiple tasks, including speech, music, acoustic scenes, and animal sounds, and outperformed the performance of TRILL.

The key of training with contrastive loss is the sampling strategy, i.e., the definition of anchor, positive, and negative samples. COLA assumes that data from the same audio clip is similar in time, so a pair of segments sampled from the same audio clip is a positive sample. Different audio clips are assumed to have different characteristics from an anchor and are negative samples. The sampling strategy is reasonable to the wide range of audio tasks like audio-tagging and speaker identification because they are clip-level classification. The segments in clips should be the same classes in these tasks. However, in some tasks, the samples do not necessarily belong to the same class even when they are highly similar in time. For example, SED requires frame-level classification of events; different segments in the same audio clip can not be the same class. Also, it is necessary to identify the time changes from the same source in PD. COLA is also unsuitable for the task because it maximizes the similarity between the

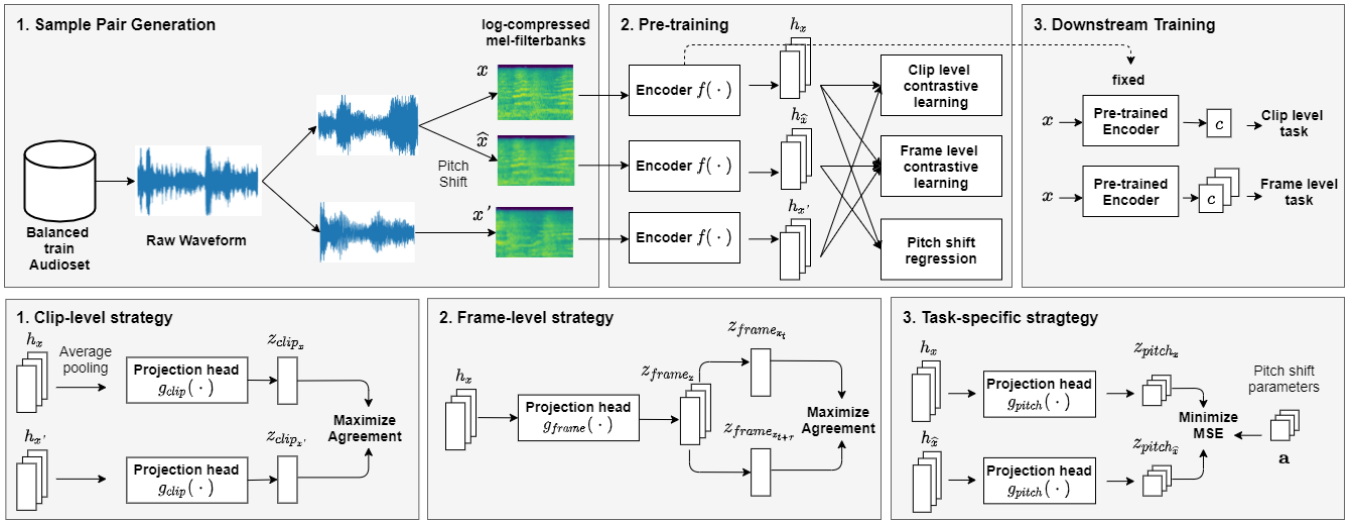

Fig. 1. Overview of proposed method.

pair of segments in the same clip, even when the pitch of the sound is different. So that we need to design the strategy carefully for each task, and it is difficult to learn general-purpose representations with a single sampling strategy.

We propose a self-supervised learning method using multiple sampling strategies for general-purpose audio representation. The proposed method performs self-supervised learning with multiple contrastive losses based on multiple strategies. By using multiple complementary strategies, a single model can be used for various tasks that the single conventional strategy cannot cover. In this study, we introduce two new strategies, frame-level strategy and task-specific strategy, in addition to the clip-level sampling strategy of COLA, to improve the performance of frame-level classification and identify the time variation from the same source.

2. METHOD

In this study, we perform self-supervised learning using three sampling strategies shown in Fig. 1. The proposed method is a multi-task learning method with three contrastive losses computed based on three different strategies. The first strategy is a clip-level strategy that focuses on differences between audio clips. The second strategy is a frame-level strategy that focuses on time changes in audio segments. The third strategy is a task-specific strategy that focuses on spectrum changes in a sound source. In the following, we discuss each strategy.

2.1. Clip-level strategy

The first strategy is based on the clip-level sampling strategy used in COLA. The strategy defines a positive sample as audio segments from the same clip as an anchor segment, and negative segments are sampled from different audio clips. COLA calculates the similarity between speech segments in two steps. First, encoder f maps the acoustic feature data $x_{1:T} \in \mathbb{R}^{N \times T}$ into embedding vectors $h = f(x_{1:T}) \in \mathbb{R}^{d \times T'}$, where N, T and T' are the number of frequency bins, time frames and the frame length of h, respectively. Then, after global average pooling along with temporal frames, the shallow neural network g_{clip} maps h to the clip representations $z_{clip} = g_{clip}(h) \in \mathbb{R}^d$, where bilinear comparisons are performed. The similarity of the representations of two segments (x, x') can be expressed:

$$s_{clip}(x, x') = g_{clip}(f(x_{1:T}))^T W g_{clip}(f(x'_{1:T})), \quad (1)$$

where W is the bilinear parameter, x' is the feature value of the segment that is different from x. As an objective function, we rely on multiclass cross-entropy applied to the similarities:

$$\mathcal{L}_{clip} = -\log \frac{\exp(s_{clip}(x, x^+))}{\sum_{x^- \in \mathcal{X}^-(x) \cup \{x^+\}} \exp(s_{clip}(x, x^-))}, \quad (2)$$

where x^+ is the positive associated to anchor x, $\mathcal{X}^-(x)$ refers to the set of negative distractors. The embedding vectors h obtained from the loss are effective for clip-level classification. However, the embedding vectors are unsuitable for frame-level classification of events and identifying the time variation of the sound emitted from the same source.

2.2. Frame-level strategy

The second strategy computes contrastive loss for frame-level embedding vectors based on the assumption that neighboring audio segments are similar and audio segments that are apart in time are dissimilar. The similarity for frame-level embedding vectors has two differences from the first clip-level strat-

egy. The first difference is that in 2.1, cross-entropy is calculated as negative examples for all clips except for the same clip. In 2.2, cross-entropy is extended to calculate the embedding vectors of $T'' = \{\tau \mid 2m + 1 \cap 0 \leqq m \leqq \frac{T'}{2} \cap m \in \mathbb{Z}\}$ frames around the target frame as a positive example. Second, while 2.1 calculates the cross-entropy for a mini-batch, 2.2 calculates the cross-entropy for the similarity of the embedding vectors of each sample and then calculates and averages the value for each sample of the mini-batch.

A shallow neural network g_{frame} maps the embedding vectors h to the frame representations $z_{frame} = g_{frame}(h) \in \mathbb{R}^{d \times T'}$. The bilinear similarity of the frame representations of two segments $(x_t, x_{t+\tau})$ is calculated as:

$$s_{frame}(x_t, x_{t+\tau}) = g_{frame}(f(x_t))^T W g_{frame}(f(x_{t+\tau})). \tag{3}$$

The objective function adapts the extended multiclass cross-entropy to the similarity of the frame representations:

$$\mathcal{L}_{frame}(x) = -\frac{1}{N'M} \sum_{n=1}^{N'} \log \frac{\sum_{\tau}^{T''} \exp(s_{frame}(x_t, x_{t+\tau}))}{\sum_{i=1}^{T'} \exp(s_{frame}(x_t, x_i))}, \tag{4}$$

where M is the number of elements in T'', and N' is the number of elements in the mini-batch. In training, we use:

$$\mathcal{L}_{frame} = \frac{1}{3}\left\{\mathcal{L}_{frame}(x) + \mathcal{L}_{frame}(x') + \mathcal{L}_{frame}(\hat{x})\right\}, \tag{5}$$

where $\hat{x}$ are the frame representations at the same time as x transformed by pitch shift for x. The embedding vectors h are effective for frame-level classification.

2.3. Task-specific strategy

2.1 and 2.2 assume that, despite differences in temporal resolution, temporal neighbors are sounds from the same source, i.e., similar embedding vectors. However, for tasks such as PD, where changes in the sound source spectrum are important, it is not suitable to make different segments positive because even small changes in the spectrum in the temporal neighborhood can be an important discriminant for the task. Our strategy is to artificially generate a positive sample for each segment by designing a new, unique data augmentation for each task. In this study, we consider the problem of pitch-shift estimation. We create a pitch-shifted anchor $\hat{x}$ by pitch-shifting the anchor x with a shift width a and detect how much the pitch-shifted anchor has been shifted for the anchor using the least-squares method. We use a shallow neural network g_{pitch} that maps the embedding vectors h to a pitch shift detector, mapping it to the frame representations $z_{pitch} = g_{pitch}(h) \in \mathbb{R}^{T'}$. Since the pitch-shifted width is a relative measure, the loss function is calculated as follows:

$$\mathcal{L}_{pitch} = \|g_{pitch}(f(\hat{x}_{1:T})) - g_{pitch}(f(x_{1:T})) - \mathbf{a}\|^2, \tag{6}$$

where $\mathbf{a}$ is a vector for length T'' that contains pitch shift parameter a. In this case, we considered data augmentation for pitch changes, but the data augmentation method and loss function should be changed for each task when applied to other problems. The embedding vectors h obtained from the loss are effective for PD.

2.4. Final loss function

Finally, we combine the three loss functions for self-supervised learning:

$$\mathcal{L} = \mathcal{L}_{clip} + \alpha\mathcal{L}_{frame} + \beta\mathcal{L}_{pitch}, \tag{7}$$

where α and β are the hyperparameters.

2.5. Transfer learning for downstream tasks

The application to the downstream tasks is made in two steps: 1) the encoder f of the pre-trained model is extracted and used as the feature extractor, 2) the feature extractor is frozen, and only the classifier is trained.

3. EXPERIMENTS

The important point of the experiments is to determine if the pre-trained embedding vectors are adaptable across audio domains and recording conditions, not only for the clip classification task but also for SED and PD.

3.1. Datasets and Tasks

We pre-trained neural network models by the proposed method using a balanced subset of Audioset. The dataset contains 18,939 samples of 10-second audio excerpts from YouTube videos. Since our method is self-supervised, we do not use any labels. In this study, we used three types of tasks for evaluation. First, we used Google speech commands (SC) as a scene-based multi-class classification [20]. The top one error was used for evaluation. Second, we used DCASE 2016 for SED in synthesized sounds [21], [22]. The evaluation followed the original DCASE and used the onset F-measure. Third, we used NSynth as a scene-based multi-class PD. The task classifies a standard MIDI piano sound into one of 88 pitches in a multi-class fashion [23]. As with CREPE, we evaluated it on pitch accuracy and chroma accuracy [24].

3.2. Experimental conditions

Given an audio input sequence, a log-compressed mel spectrogram was extracted with a window size of 25 ms, a hop size of 10 ms, and $N = 64$ mel spaced frequency bins in the range of 60 - 7800 Hz corresponding to $T = 96$ frames, 960 ms. These features are passed to the encoder f based

Table 1. Test scores of a linear classifier trained on top of proposed embeddings or baseline pre-trained embeddings.

	SC	DCASE	NSynth	
	acc	F1	pitch	chroma
PANNs	0.083	0.754	0.012	0.096
COLA	0.459	0.232	0.434	0.470
$\mathcal{L}_{clip} + \mathcal{L}_{frame}$	0.535	**0.344**	0.424	0.462
$\mathcal{L}_{clip} + \mathcal{L}_{pitch}$	**0.585**	0.244	0.428	0.463
$\mathcal{L}_{clip} + \mathcal{L}_{frame} + \mathcal{L}_{pitch}$	0.572	0.278	**0.452**	**0.487**

Table 2. Test scores with different similarity functions.

	SC	DCASE	NSynth	
	acc	F1	pitch	chroma
Cosine Similarity	0.401	0.094	0.212	0.238
Bilinear Similarity	**0.572**	**0.278**	**0.452**	**0.487**

on EfficientNet-B0 [25], a lightweight and scalable convolutional neural network. Apply average pooling in the frequency direction of the last layer of the encoder to obtain embedding vectors h of size 512×3. This layer contains a 512-unit fully connected layer, followed by the Layer Normalization [26] and tanh activation functions. We trained the encoder by iterating 250 epochs for the subset of Audioset, using a batch size of 1024, a learning rate of 10^{-4}, Adam [27]. The hyperparameter m of T'' was set to 0 due to $T'' = 3$. α and β in Eq. 7 was set to 1.0. The shift width of the pitch shift was transformed every epoch to a random value in the range of 0.8 to 1.2. In the experiment, we investigated the effect on each loss function and the effect of the similarity function.

For the downstream tasks, we trained the classifier by iterating 250 epochs for each dataset using a batch size of 1024, a learning rate of 10^{-4}, Adam, and a dropout of 0.1, with a fully connected layer using the pre-computed embedding vectors as input. We evaluated whether the embedding vectors trained by pre-training were universally adaptable regardless of the audio domain, recording conditions, or tasks.

3.3. Results

Table 1 showed the scores in the three downstream tasks, using two baselines: the pre-trained model with Cnn14-DecisionLevelAtt in PANNs, and the model with COLA self-supervised trained on the subset of Audioset. We also compared the combination of loss functions. First, we used the same procedure as in [10], [12], [28], [29] to evaluate the embedding vectors of the pre-trained proposed method with a linear classifier on the frozen embedding vectors. Compared to the other self-supervised learning methods, PANNs performed poorly on SC and NSynth, despite being trained on the full set of Audioset. The result indicated that some tasks were not suitable for different learning methods. In addition, when comparing COLA with the proposed method, performance was improved in all tasks by adapting loss functions that combined three sampling strategies. The results showed that multiple perspectives allowed us to capture the same source's time variation and feature changes. The ablation study of each loss function showed that combining L_{frame} and L_{pitch} improved the performance for NSynth. In addition, the performance improvement is the largest when using L_{frame} for SC and L_{pitch} for DCASE, while the performance improvement is the smallest when using L_{pitch} for SC and L_{frame} for DCASE. There is a trade-off between L_{frame} and L_{pitch}. Since L_{pitch} was calculated at the frame-level, it can be considered a variant of frame-level loss. There may be a trade-off between frame-level loss, which made the distance of the same source closer, and pitch loss, which results in different embedding when the pitch was different even for the same source. These results showed that the combination of the three-loss functions was found to be important.

To investigate the role of the similarity function, we compared the pre-training of the model using cosine and bilinear similarity. Cosine similarity was calculated as:

$$s(x, x') = \frac{g(f(x))^T \cdot g(f(x'))}{\|g(f(x))\| \|g(f(x'))\|}, \tag{8}$$

where g was each objective function's mapping function, x and x' were chosen appropriately for each objective function. The calculated cosine similarity was normalized by dividing by 0.2. Eq. 8 was used instead of Eq. 1 and Eq. 3. Table 2 showed the results. The best results were obtained when bilinear similarity was used, indicating its effectiveness.

For future studies, we will conduct experiments with the full set of Audioset. Experiments with the full set will provide a better comparison. The extension of the proposed method to other modalities is also one of the important future tasks. We believe that using multiple sampling strategies is also key to the tasks in the other modalities, such as audio-video. A model trained without considering time information cannot be applied to tasks with frame-level classification, such as lip-reading or emotion recognition. For example, in [30], they proposed two different loss functions for speaker verification and speech recognition. However, they trained two separate models for each task. By designing multiple sampling strategies for other modalities, we can build more general pre-trained models.

4. CONCLUSION

This paper proposes a self-supervised learning method using multiple sampling strategies for general-purpose audio representation by designing loss functions using audio pairs obtained using multiple sampling strategies. Our method improves the performance of all tasks compared to existing methods in the experiment of the subset of Audioset. The results show that it is effective to design and combine the loss functions according to the tasks.

5. REFERENCES

[1] Y. Zhang, J. Qin, D. S. Park, *et al.*, *Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition*, 2020.

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

[3] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted Residual Learning for Efficient Keyword Spotting," in *Proc. Interspeech*, 2021, pp. 4538–4542.

[4] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda, "Acoustic Event Detection Method Using Semi-Supervised Non-Negative Matrix Factorization with Mixtures of Local Dictionaries," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 2016, pp. 45–49.

[5] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A Convolutional Representation for Pitch Estimation," in *Proc. ICASSP*, 2018, pp. 161–165.

[6] Q. Kong, Y. Cao, T. Iqbal, *et al.*, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[7] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[8] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 1877–1901.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[10] A. van den Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, 2019.

[11] R. Qian, T. Meng, B. Gong, *et al.*, "Spatiotemporal contrastive video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6964–6974.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020, arXiv:2004.04136.

[14] E. Kharitonov, M. Rivière, G. Synnaeve, *et al.*, "Data augmenting contrastive learning of speech representations in the time domain," in *Proc. SLT*, 2021, pp. 215–222.

[15] M. Rivière, A. Joulin, P.-E. Mazar'e, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7414–7418, 2020.

[16] A. Baevski, S. Schneider, and M. Auli, "Vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2020.

[17] J. Shor, A. Jansen, R. Maor, *et al.*, "Towards Learning a Universal Non-Semantic Representation of Speech," in *Proc. Interspeech*, 2020, pp. 140–144.

[18] A. Jansen, M. Plakal, R. Pandya, *et al.*, "Unsupervised learning of semantic audio representations," in *Proc. ICASSP*, 2018.

[19] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *Proc. ICASSP*, 2021, pp. 3875–3879.

[20] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *ArXiv e-prints*, 2018.

[21] A. Mesaros, T. Heittola, E. Benetos, *et al.*, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

[22] G. Lafay, E. Benetos, and M. Lagrange, "Sound event detection in synthetic audio: Analysis of the DCASE 2016 task results," in *Proc. WASPAA*, 2017, pp. 11–15.

[23] J. Engel, C. Resnick, A. Roberts, *et al.*, *Neural audio synthesis of musical notes with wavenet autoencoders*, 2017.

[24] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. ICASSP*, 2018, pp. 161–165.

[25] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 6105–6114.

[26] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[28] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry, and D. Roblek, *Self-supervised audio representation learning for mobile devices*, 2019.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 1597–1607.

[30] S.-W. Chung, H.-G. Kang, and J. S. Chung, "Seeing voices and hearing voices: Learning discriminative embeddings using cross-modal self-supervision," *Interspeech 2020*, 2020.