



Mutual Information Enhanced Training for Speaker Embedding

Yuzhi Tu, Man-Wai Mak

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University,
Hong Kong SAR, China

918tyz@gmail.com, enmwmak@polyu.edu.hk

Abstract

Mutual information (MI) is useful in unsupervised and self-supervised learning. Maximizing the MI between the low-level features and the learned embeddings can preserve meaningful information in the embeddings, which can contribute to performance gains. This strategy is called deep InfoMax (DIM) in representation learning. In this paper, we follow the DIM framework so that the speaker embeddings can capture more information from the frame-level features. However, a straightforward implementation of DIM may pose a dimensionality imbalance problem because the dimensionality of the frame-level features is much larger than that of the speaker embeddings. This problem can lead to unreliable MI estimation and can even cause detrimental effects on speaker verification. To overcome this problem, we propose to squeeze the frame-level features before MI estimation through some global pooling methods. We call the proposed method squeeze-DIM. Although the squeeze operation inevitably introduces some information loss, we empirically show that the squeeze-DIM can achieve performance gains on both Voxceleb1 and VOICES-19 tasks. This suggests that the squeeze operation facilitates the MI estimation and maximization in a balanced dimensional space, which helps learn more informative speaker embeddings.

Index Terms: speaker verification, speaker embedding, mutual information, variational lower bound

1. Introduction

Speaker embedding plays a vital role in modern speaker recognition systems. Ideal speaker embeddings should not only be speaker discriminative but also be robust against adverse conditions, e.g., noise, reverberation, domain mismatch, etc. Since the emergence of the x-vectors [1], most speaker embeddings have turned to use deep neural networks (DNNs) to capture the speaker information in the speech segments. Commonly, these networks share a similar structure: a frame-level network, a pooling layer, and a segment-level network. To preserve as much speaker information as possible, several strategies have been exploited in different levels of the embedding networks.

In the frame-level network, shortcut connections [2, 3] have been widely used to aggregate information from the previous convolutional layers [4, 5, 6]. Simultaneously, various attention mechanisms have also been adopted in the convolutional neural networks to attend to the speaker-related feature maps. For example, squeeze-excitation (SE) blocks [7] were used to capture the interdependencies across the channel dimension in [6], which can be seen as a channel-attention operation. Besides the channel attention, temporal-frequency attention was also applied in [8]. As for the pooling layer, various utterance-level aggregation methods have been proposed to emphasize the speaker-discriminative frames, e.g., attentive pooling [9, 5, 10], information preservation pooling [11], short-time spectral pool-

ing [12], etc. However, mechanisms that directly retain information at the segment level are rare. Instead, a common strategy is to transform the speaker embeddings with information maximization using a light-weight network after extracting the embeddings. An example is the Info-maximized variational domain adversarial neural networks proposed in [13].

Studies have shown that features at the lower layers of a DNN are generally more class-agnostic, while those at the upper layers are more class-specific [14, 7]. Accordingly, the prediction uncertainty decreases when signals flow from the lower layers to the upper layers, suggesting that training is a process of information loss. Therefore, some speaker information will inevitably diminish in the upper layers when training a speaker embedding network. As such, an intuitive way to enhance speaker information in the embeddings is to explicitly maximize the mutual information (MI) between the frame-level features and the segment-level embeddings, so that the embeddings can learn extra speaker information from the more general low-level features.

In fact, exploiting MI to learn meaningful speaker embeddings is not new. In [13], InfoVDANN was introduced to maximize the MI between the transformed embeddings and the input embeddings so that the transformed embeddings are more speaker discriminative. However, this method is operated at the segment level, which forbids it from leveraging useful information in the frame-level layers. Attributed to the deep InfoMax (DIM) [15] framework for representation learning, estimating the MI between the frame-level features and the segment-level embeddings has become feasible via MI neural estimators (MINEs) [16]. In [11], the authors followed the idea of MINE and proposed an information preservation pooling method to feed more information from the last frame-level layer to the statistics aggregation layer.

Although DIM makes it viable to relate the speaker embeddings with the frame-level features through MI maximization, it may pose a dimensionality imbalance problem. Because the dimensionality of the (flattened) frame-level features is generally much larger than that of the speaker embeddings, the learned MI estimators may be biased towards the frame-level features. In this case, MI cannot be accurately approximated and directly applying MI maximization can fail to learn useful information. Therefore, it would be amenable to perform dimensionality reduction before MI estimation and maximization. Inspired by the squeeze operation in the SE networks [7], we propose to perform global pooling on the frame-level features for each channel before maximizing the MI between the frame-level features and the speaker embeddings. The global pooling essentially reduces the dimensionality of the frame-level features, which avoids imbalanced dimensionality in the MI estimation. We call the resulting method *squeeze-DIM*, which uses the MI estimation between the squeezed frame-level features and the speaker embeddings as a proxy to that between the original pairs. Although it

seems counterintuitive to squeeze the frame-level features before MI maximization because this operation inevitably introduces information loss, we empirically show that the squeeze operation facilitates the MI estimation/maximization, thus contributing to performance improvement. Different from DIM which aims for unsupervised learning, we use squeeze-DIM as a regularizer with the main task being speaker classification.

2. Variational mutual information estimation

In this section, the deep InfoMax (DIM) framework [15] and some variational lower bounds [17] to the MI are introduced.

2.1. Deep InfoMax

The MI between two random variables X and Y is defined as the Kullback-Leibler (KL) divergence between their joint distribution and the product of their marginals:

$$I(X; Y) = D_{\text{KL}}(P(X, Y) \| P(X)P(Y)).$$

However, MI is difficult to estimate, especially when X and Y locate in a continuous, high-dimensional space. To scale with the dimension, various variational bounds are introduced in combination with neural networks when estimating MI. For the scenario where the objective is to learn encoded representations $Y = E_\phi(X)$ from the input X , we maximize the MI between the input and output of the encoder [15, 18]:

$$(\hat{\phi}, \hat{\theta}) = \underset{\phi, \theta}{\operatorname{argmax}} I_\theta(X; E_\phi(X)), \quad (1)$$

where ϕ and θ parameterize the encoder E_ϕ and the MI estimator I_θ , respectively.

2.2. Variational lower bounds on mutual information

There are several popular variational lower bounds on MI [16, 17, 19]. The basic idea behind these bounds is that if we can train a discriminator (MI estimator) that is able to accurately differentiate the samples drawn from the joint distribution and those from the product of the marginals, we obtain a good estimate of the true MI.

One variational estimator is called InfoNCE [17, 20]:

$$I(X; Y) \geq \mathbb{E} \left[\frac{1}{B} \sum_{i=1}^B \log \frac{e^{f(\mathbf{x}_i, \mathbf{y}_i)}}{\frac{1}{B} \sum_{j=1}^B e^{f(\mathbf{x}_j, \mathbf{y}_i)}} \right] \triangleq I_{\text{InfoNCE}}(X; Y), \quad (2)$$

where the expectation is over B independent samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^B$ drawn from the joint distribution $P(\mathbf{x}, \mathbf{y})$ and B is the mini-batch size. $f(\cdot, \cdot)$ denotes the *critic*, which takes a pair of samples and outputs a scalar score. Common critics can be a bilinear function $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{W} \mathbf{y}$ where \mathbf{W} is a weight matrix to be learned, a separable function $f(\mathbf{x}, \mathbf{y}) = g_{\theta_1}(\mathbf{x})^\top g_{\theta_2}(\mathbf{y})$ where $g_{\theta_1}(\cdot)$ and $g_{\theta_2}(\cdot)$ are functions characterized by networks with parameters θ_1 and θ_2 , respectively, and a concatenated function $f(\mathbf{x}, \mathbf{y}) = h_\theta([\mathbf{x}, \mathbf{y}])$ where $h_\theta(\cdot)$ denotes a network parameterized by θ [17].

Because $I_{\text{InfoNCE}}(X; Y)$ is a multi-sample lower bound, it has low variance. However, $I_{\text{InfoNCE}}(X; Y)$ is biased and is upper bounded by $\log B$, which means that this bound will be loose when the true MI $I(X; Y) > \log B$.

Another MI estimator is based on the variational f -divergence estimation specialized to KL divergence (f -GAN KL) [21]:

$$I(X; Y) \geq \mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[f(\mathbf{x}, \mathbf{y})] - \mathbb{E}_{p(\mathbf{x})p(\mathbf{y})} \left[e^{f(\mathbf{x}, \mathbf{y})-1} \right] \triangleq I_{\text{NWJ}}(X; Y). \quad (3)$$

$I_{\text{NWJ}}(X; Y)$ is unbiased but presents high variance [17].

There are also other MI estimation such as the non-linearly interpolated lower bound [17] and the smoothed mutual information lower-bound [19]. These estimators have a better bias-variance trade-off than $I_{\text{InfoNCE}}(X; Y)$ and $I_{\text{NWJ}}(X; Y)$.

3. Mutual information enhanced training

This paper aims to learn informative speaker embeddings through maximizing the MI between the frame-level features and the embeddings in combination with the conventional speaker classification task. Two instances of the MI-enhanced network are detailed.

3.1. DIM regularized speaker embedding

We adopt the DIM framework as a regularization on the embeddings so that more low-level information can be incorporated in the embeddings during training. As shown in Figure 1, there are two branches in the MI-enhanced training. The upper branch represents a standard speaker classification task, while the lower is a DIM regularizer.

Let \mathbf{x} , \mathbf{x}_{conv} , \mathbf{x}_{emb} be the input acoustic feature vectors, the immediate convolutional feature maps, and the speaker embeddings, respectively. Without loss of generality, we use a separable function as the *critic* in the MI estimator, although other *critics* can also be applied. To preserve extra information in \mathbf{x}_{emb} , we maximize the MI between \mathbf{x}_{conv} and \mathbf{x}_{emb} as in (1):

$$(\hat{\phi}, \hat{\theta}_1, \hat{\theta}_2) = \underset{\phi, \theta_1, \theta_2}{\operatorname{argmax}} I_{\theta_1, \theta_2}(X_{\text{conv}}; X_{\text{emb}}), \quad (4)$$

where ϕ parameterizes the encoding network (within the red dashed box in Figure 1) between \mathbf{x}_{conv} and \mathbf{x}_{emb} , i.e., $\mathbf{x}_{\text{emb}} = E_\phi(\mathbf{x}_{\text{conv}})$. θ_1 and θ_2 constitute the MI estimator with a separable *critic* as follows:

$$f(\mathbf{x}_{\text{conv}}, \mathbf{x}_{\text{emb}}) = g_{\theta_1}(\text{Flatten}(\mathbf{x}_{\text{conv}}))^\top g_{\theta_2}(\mathbf{x}_{\text{emb}}). \quad (5)$$

The MI estimator I_{θ_1, θ_2} can be I_{InfoNCE} and I_{NWJ} in (2) and (3), respectively.

Denote the classification loss in the upper branch of Figure 1 as

$$\mathcal{L}_{\text{cls}}(\omega) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log C_\omega(\mathbf{x}_{ik}), \quad (6)$$

where y_{ik} is an element of the one-hot speaker labels, and $C_\omega(\cdot)$ represents the whole speaker classifier parameterized by ω . N and K denote the number of training samples and the number of speakers, respectively. Note that the parameter of the encoder ϕ is a subset of ω . If we define the total loss of the network as

$$\mathcal{L}(\omega, \theta_1, \theta_2) = \mathcal{L}_{\text{cls}}(\omega) - \alpha I_{\theta_1, \theta_2}(X_{\text{conv}}; X_{\text{emb}}), \quad (7)$$

where α is a hyperparameter weighting the contribution of MI regularization, then MI-enhanced training can be expressed as follows:

$$(\hat{\omega}, \hat{\theta}_1, \hat{\theta}_2) = \underset{\omega, \theta_1, \theta_2}{\operatorname{argmin}} \mathcal{L}(\omega, \theta_1, \theta_2). \quad (8)$$

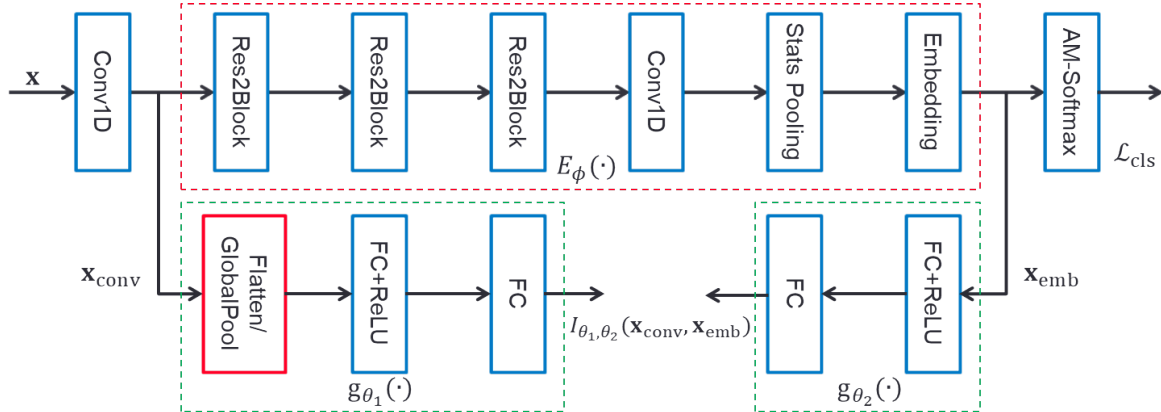


Figure 1: Schematic of MI-enhanced training. The whole network comprises two sub-networks: an upper speaker classifier C_ω and a lower MI estimator I_{est} . The MI estimator is instantiated by a separable critic as in (5) and (9) for DIM regularization and squeeze-DIM regularization, respectively. For DIM regularization, a flatten layer is used as the first layer of g_{θ_1} , while a global pooling layer is applied for squeeze-DIM regularization. FC denotes the fully-connected layer.

3.2. Squeeze-DIM regularized speaker embedding

One problem of the DIM regularized speaker embedding is that $I_{\theta_1, \theta_2}(X_{\text{conv}}; X_{\text{emb}})$ can be unreliable when the dimensionality of the flattened \mathbf{x}_{conv} is much larger than that of \mathbf{x}_{emb} , because the critic $f(\mathbf{x}_{\text{conv}}, \mathbf{x}_{\text{emb}}) = g_{\theta_1}(\text{Flatten}(\mathbf{x}_{\text{conv}}))^\top g_{\theta_2}(\mathbf{x}_{\text{emb}})$ would be biased towards learning the information in \mathbf{x}_{conv} only, other than the *mutual* information between X_{conv} and X_{emb} .

To address the problem of dimensionality imbalance, we propose to squeeze \mathbf{x}_{conv} using some global pooling methods for each channel before MI estimation. In this case, the *critic* becomes

$$f(\mathbf{x}_{\text{conv}}, \mathbf{x}_{\text{emb}}) = g_{\theta_1}(\text{GlobalPool}(\mathbf{x}_{\text{conv}}))^\top g_{\theta_2}(\mathbf{x}_{\text{emb}}). \quad (9)$$

Common global pooling operations can be global average pooling, statistics pooling [1], attentive pooling [9, 10], etc. The optimization is the same as (8).

Note that \mathbf{x}_{conv} does not necessarily have to be at the output of the first convolutional layer as shown in Figure 1. Instead, it can be the output of any frame-level layers, and it can even be the input acoustic features.

In conclusion, the only difference between the proposed embedding and the DIM regularized embedding in Section 3.1 is that the former applies a channel-wise global pooling operation on the convolutional feature maps instead of flattening them. The squeeze operation facilitates the MI estimation although it introduces some information loss, which may explain the empirical performance improvement in Section 5.

4. Experimental setup

We evaluated the performance of squeeze-DIM on the VoxCeleb1 test set (clean) [22] and the VOiCES 2019 development and evaluation sets [23].

4.1. Training of speaker embedding extractor

Both VoxCeleb1 development and VoxCeleb2 development data were used for training, which amounts to 2,105,949 utterances from 7,185 speakers. We followed the Kaldi’s VoxCeleb recipe to prepare the training data, i.e., using 40-dimensional

filter bank features, performing energy-based voice activity detection, implementing augmentation (by adding reverberation, noise, music, and babble to the original speech files), applying cepstral mean normalization with a window of 3 seconds, and filtering out utterances with a duration less than 4 seconds.¹ Totally, we had approximately twice the number of clean utterances for training the embedding network.

The embedding extractor in the upper branch of Figure 1 is used as the baseline. The number of output filters of the convolutional layers (or blocks) is 512 except that it is 1,536 for the last convolutional layer. The kernel sizes of the convolutions are 5, 3, 3, 3, and 1, respectively, and the dilation rates are 1, 2, 3, 4, and 1, respectively. The scale and the number of convolutional filters of all three Res2blocks [24] are 8 and 64, respectively. We used an embedding size of 192. For the MI-enhanced training, we followed the structure in Figure 1 and set the number of nodes in all fully-connected layers in the MI estimator (the lower part of Figure 1) to 64. I_{InfoNCE} in (2) was used as the MI estimator because we found that it is more stable to optimize than I_{NWJ} (see (3)) and other MI estimators [17, 19] in our experimental setups. The hyperparameter α for weighting the MI estimation was set to 0.1. We used a global average pooling layer for the squeeze operation in the squeeze-DIM regularized speaker embedding. To further verify the effectiveness of MI regularization, we also included an embedding that integrates the SE block [7] with a reduction factor of 16 as a comparison.

The additive margin softmax loss [25] was used for training. The additive margin and the scaling factor were set to 0.25 and 30, respectively. The mini-batch size was set to 128 and there are around 2,337 mini-batches in one epoch. Each mini-batch was created by randomly selecting speech segments of 2–4 seconds from the training data. However, for DIM regularized embeddings, the duration of the speech segments was set to 2 seconds.² We used an Adam [26] optimizer. The learning

¹<https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb/v2>.

²Because the convolutional feature maps \mathbf{x}_{conv} will be flattened for the DIM regularized embedding, we need to fix the dimension of \mathbf{x}_{conv} along the frame axis so that the network structure is static during training; otherwise the network cannot be trained. In fact, we also tried the durations of 3s and 4s, but the configuration of 2s slightly outperforms the other setups. This is the reason why we used the duration of 2s for the training segments under this scenario.

Table 1: Performance on VoxCeleb1, VOiCES19-dev, and VOiCES19-eval. The upper part (Rows 1–4) is the main result of MI-enhanced training, while the lower part (Rows 5–10) shows the ablation study by varying the source of \mathbf{x}_{conv} in Figure 1. The layer in the parenthesis denotes where \mathbf{x}_{conv} comes from, e.g., ‘1st conv’ means that \mathbf{x}_{conv} is the output of the first convolutional layer, etc.

Row	Embedding	VoxCeleb1		VOiCES19-dev		VOiCES19-eval	
		EER	minDCF	EER	minDCF	EER	minDCF
1	Baseline	1.85	0.187	2.16	0.270	5.88	0.468
2	Baseline + DIM	1.94	0.189	1.85	0.236	5.73	0.449
3	Baseline + squeeze-DIM	1.62	0.167	1.82	0.224	5.36	0.408
4	Baseline + SE [7]	1.78	0.182	1.69	0.238	5.72	0.446
5	Baseline + squeeze-DIM (input)	1.78	0.184	1.96	0.236	5.78	0.431
6	Baseline + squeeze-DIM (1st conv)	1.62	0.167	1.82	0.224	5.36	0.408
7	Baseline + squeeze-DIM (2nd conv)	1.65	0.183	1.73	0.217	5.51	0.419
8	Baseline + squeeze-DIM (3rd conv)	1.80	0.185	1.97	0.232	5.64	0.420
9	Baseline + squeeze-DIM (4th conv)	1.87	0.192	1.99	0.227	5.48	0.422
10	Baseline + squeeze-DIM (5th conv)	1.73	0.192	2.19	0.228	5.57	0.430

rate was initialized at 1.0×10^{-3} and it was decayed by half at Epoch 25. At Epoch 50, we increased the learning rate to 1.0×10^{-3} and decreased it by half again at Epoch 75. Totally, the networks were trained for 100 epochs.

4.2. PLDA training

We used Gaussian PLDA backends [27] for both evaluation tasks. For VoxCeleb1, the PLDA model was trained on the \mathbf{x} -vectors extracted from the clean utterances in the training set for the embedding network. For VOiCES 2019, we trained the backend on the concatenated speech with the same video session and used utterances augmented with reverberation and noise. Before PLDA training, the \mathbf{x} -vectors were projected onto a 192-dimensional space by LDA for VoxCeleb1 and 150-dimensional space by LDA for VOiCES 2019, followed by whitening and length normalization. The LDA projection matrix was trained on the same dataset as for training the PLDA models. For VOiCES 2019, we also applied adaptive score normalization [28]. The cohort was selected from the longest two utterances of each speaker in the PLDA training data.

5. Results and discussions

The main result of MI-enhanced training is shown in the upper part (Rows 1–4) of Table 1. We can see that DIM regularized embedding only achieves marginal improvement over the baseline on VOiCES 2019, whereas the squeeze-DIM regularized embedding remarkably outperforms the DIM regularized version on all tasks. Although DIM regularization should theoretically incorporate more information in the embeddings than the baseline, the practical implementation of the MI estimator can be severely biased towards the information of the convolutional feature maps due to their higher dimensionality than the embeddings. As a result, the MI estimation is unreliable and this limits the regularization effect on the speaker embeddings. In contrast, although the squeeze operation can introduce information loss, it facilitates the MI estimation, which helps feed useful information from the low-level features into the embedding. This verifies the motivation of the proposed squeeze-DIM regularization. By comparing Row 3 and Row 4, we observe that squeeze-DIM is more effective than applying SE, which further verifies the effectiveness of squeeze-DIM.

The lower part (Rows 5–10) of Table 1 shows the performance by varying the source of \mathbf{x}_{conv} in Figure 1. In general,

we can achieve the best performance if \mathbf{x}_{conv} is the output from the first frame-level layer (Row 6). When \mathbf{x}_{conv} moves from the first layer to upper layers, the performance degrades gradually. This may be because the information becomes more speaker-specific and attenuates gradually while propagating to the upper layers. We also see that all the squeeze-DIM regularized embeddings can achieve better performance than the baseline on VOiCES 2019. However, when \mathbf{x}_{conv} comes from the fourth and fifth convolutional layers (Row 9 and Row 10), their performance is slightly worse than the baseline on VoxCeleb1. This suggests that MI maximization is more effective on noisy data.

Another interesting observation is that maximizing the mutual information between the inputs and the embeddings is less effective than maximizing the mutual information between the immediate convolutional features and the embeddings, which can be verified by comparing Row 5 and Rows 6–7. This means that although the input filter-bank features contains more information than the middle layers, it is difficult to extract speaker information directly from the input layer.

6. Conclusions

In this paper, we aim to preserve more speaker information in the embeddings through MI-enhanced training. We propose a squeeze-DIM regularized embedding to address the problem of dimensionality imbalance between the frame-level features and the embeddings during MI maximization. The evaluation results on both VoxCeleb1 and VOiCES 2019 show that the proposed method outperforms the baseline, DIM regularized embedding, and the SE-integrated embedding, verifying the effectiveness of the proposed method.

7. Acknowledgements

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 152137/17E and National Natural Science Foundation of China (NSFC), Grant No. 61971371.

8. References

- [1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2018, pp. 5329–5333.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning

- for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.
- [4] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, “Utterance-level aggregation for speaker recognition in the wild,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 5791–5795.
- [5] W. Lin, M. W. Mak, and L. Yi, “Learning mixture representation for deep speaker embedding using attention,” in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2020, pp. 210–214.
- [6] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnns based speaker verification,” in *Proc. Annual Conference of the International Speech Communication Association*, 2020, pp. 3830–3834.
- [7] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.
- [8] S. Yadav and A. Rai, “Frequency and temporal convolutional attention for text-independent speaker recognition,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6794–6798.
- [9] K. Okabe, T. Koshinaka, and K. Shinoda, “Attentive statistics pooling for deep speaker embedding,” in *Proc. Annual Conference of the International Speech Communication Association*, 2018, pp. 2252–2256.
- [10] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” in *Proc. Annual Conference of the International Speech Communication Association*, 2018, pp. 3573–3577.
- [11] M. Han, W. Kang, S. Mun, and N. Kim, “Information preservation pooling for speaker embedding,” in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2020, pp. 60–66.
- [12] Y. Tu and M. W. Mak, “Short-time spectral aggregation for speaker embedding,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2021, pp. 6708–6712.
- [13] Y. Tu, M. W. Mak, and J. T. Chien, “Variational domain adversarial learning with mutual information maximization for speaker verification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2013–2024, 2020.
- [14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.
- [15] R. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *International Conference on Learning Representations*, 2019.
- [16] M. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, and Y. Bengio, “Mutual information neural estimation,” in *Proc. International Conference on Machine Learning*, 2018, pp. 531–540.
- [17] B. Poole, S. Ozair, A. van den Oord, A. Alemi, and G. Tucker, “On variational bounds of mutual information,” in *Proc. International Conference on Machine Learning*, 2019, pp. 5171–5180.
- [18] M. Tschannen, J. Djolonga, P. Rubenstein, S. Gelly, and M. Lucic, “On mutual information maximization for representation learning,” in *International Conference on Learning Representations*, 2020.
- [19] J. Song and S. Ermon, “Understanding the limitations of variational mutual information estimators,” in *International Conference on Learning Representations*, 2020.
- [20] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” in *arXiv preprint arXiv:1807.03748*, 2018.
- [21] S. Nowozin, B. Cseke, and R. Tomioka, “f-GAN: Training generative neural samplers using variational divergence minimization,” in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.
- [22] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech & Language*, vol. 60, 2020.
- [23] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, M. Lawson, and A. Barrios, “The VOICES from a distance challenge 2019 evaluation plan,” in *arXiv preprint arXiv:1902.10828*, 2019.
- [24] S. Gao, M. Cheng, K. Zhao, X. Zhang, M. Yang, and P. Torr, “Res2Net: A new multi-scale backbone architecture,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.
- [25] F. Wang, J. Cheng, W. Liu, and H. Liu, “Short term spectral analysis, synthesis, and modification by discrete Fourier transform,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 235–238, 2018.
- [26] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations*, 2015.
- [27] S. Ioffe, “Probabilistic linear discriminant analysis,” in *Proc. European Conference on Computer Vision*, 2006, pp. 531–542.
- [28] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. Sánchez, and J. Černocký, “Analysis of score normalization in multilingual speaker recognition,” in *Proc. Annual Conference of the International Speech Communication Association*, 2017, pp. 1567–1571.