# Denoising Student Features with Diffusion Models for Knowledge Distillation in Speaker Verification

Zezhong Jin, Youzhi Tu, Zhe Li, Zilong Huang, Chong-Xin Gan, and Man-Wai Mak
*Dept. of Electrical and Electronic Engineering,*
*The Hong Kong Polytechnic University, Hong Kong SAR, China*

***Abstract*—In recent years, there has been a surge in the use of a pre-trained speech model as a feature extractor for speaker verification (SV). To reduce model complexity, researchers transfer knowledge from a pre-trained model to a lightweight student model, enabling the latter to reach a performance level not attainable by conventional methods. However, due to the differences in model capacity, the student features contain more noise. This results in discrepancies between the teacher and student features at the intermediate layers, negatively impacting feature-level knowledge distillation (KD). To address this issue, we employ a diffusion model to denoise the student features for KD (DenoKD). This approach enables more effective feature-level distillation. Our method, trained with a small ECAPA-TDNN, achieved a 13% improvement over the baseline on the VoxCeleb1-O test set. Further more, the DenoKD mechanism is found to be effective for SV on short test utterances.**

***Index Terms*—Speaker verification, knowledge distillation, diffusion models, short-utterance, pre-trained speech models**

## I. Introduction

Speaker verification (SV) plays a crucial role in various domains, such as biometric authentication, e-banking, and access control. By leveraging sophisticated models [1]–[4], large datasets [5], [6], and carefully designed loss functions [7]–[9], SV systems have achieved excellent performance. In recent years, researchers have leveraged features extracted from large-scale pre-trained automatic speech recognition (ASR) models [10]–[12] and fed the features into speaker embedding networks, achieving state-of-the-art results [13], [14]. However, these models are computationally expensive and have a large number of trainable parameters, making them difficult to deploy on edge devices [15], [16].

Knowledge distillation (KD) [17] is a learning process that allows a lightweight student model to mimic a more powerful teacher model so that the student model can achieve performance close to the teacher model. In SV, knowledge can be distilled at the feature level [18], [19] and the label level [15]. The former enables the student network to mimic the behaviors of the teacher network by reducing the distance between the intermediate layer's outputs of the two networks. The latter minimizes the Kullback–Leibler (KL) divergence between the output probabilities of the teacher and student networks.

In [18], the author demonstrated that minimizing the mean square error (MSE) and cosine distance between the embeddings of the teacher and student models from the same utterance is an effective distillation method. The author in [19] distilled the complex capabilities of an ASR Conformer [20] into an SV model and demonstrated that distillation between the student and teacher models on frame-level features is also effective. Truong et al. [15] applied decoupled KD [21] at the label level and enhanced the non-target speakers' information to achieve effective distillation in SV. The authors in [16] extended Troung et al.'s work by sorting the output probabilities and dividing the sorted list into head and tail groups.

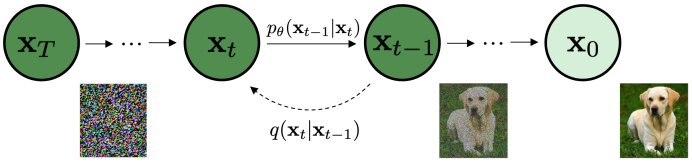



Fig. 1: Forward and denoise processes of a diffusion model.

The methods above overlook an important issue in knowledge distillation: how to transfer knowledge from the teacher to the student by matching the intermediate features. Due to the difference in the teacher's and student's capacity, there is a significant discrepancy between the features extracted at the frame-level layers of the teacher and the student [19]. By the same token, a large discrepancy also exists between the features extracted from their embedding layers. These discrepancies make the feature-level knowledge distillation less effective (as demonstrated in Section IV).

Following the idea of diffusion knowledge distillation [22], we consider the features extracted from the student a noisy version of the teacher and apply a diffusion model to denoise the student's features [23]. As shown in Fig. 1, the diffusion model contains two processes: a diffusion process where noise is gradually added to the clean image and a reverse process where the model learns to reconstruct the clean image. We use the teacher model's outputs as clean data to train the diffusion model. Then, we feed the student features into the reverse process for denoising, obtaining refined student features that align well with the original teacher features for feature-level KD. We refer to the method as DenoKD.

We use a lightweight diffusion model and reduce the number of denoising steps to minimize computational cost. We apply DenoKD to both the frame-level features and the speaker embeddings. Due to the lower dimensionality of the speaker

This work was supported by the RGC of Hong Kong SAR, Grant Nos. PolyU 15210122 and 15228223.

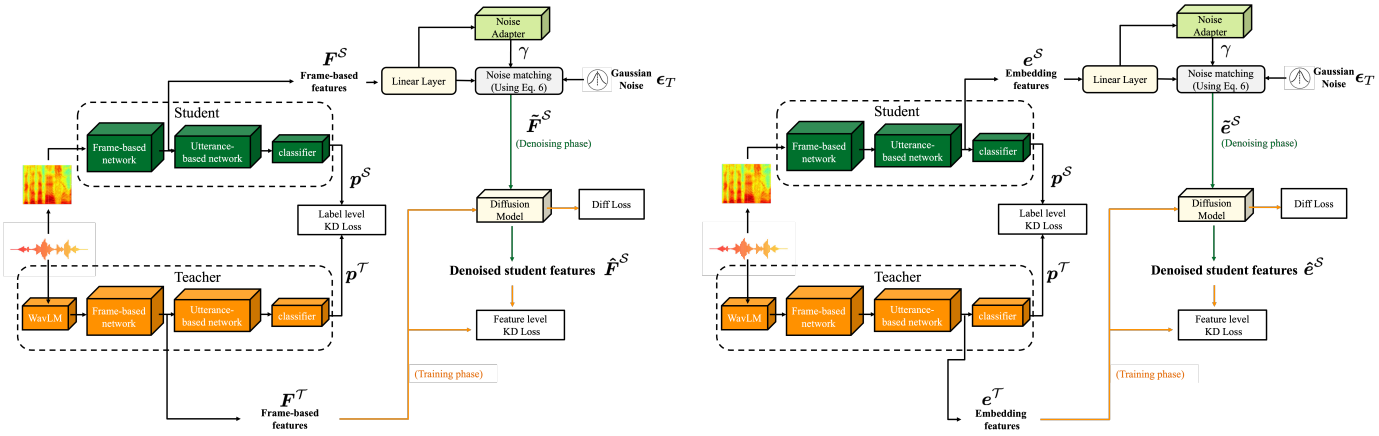



Fig. 2: Process of DenoKD at (a) the frame level and (b) the embedding level.

embeddings, the embedding-level DenoKD incurs only a small additional computational cost. Our contributions are summarized as follows:

1) We proposed a DenoKD framework that employs a diffusion model to denoise the student features for SV. To the best of our knowledge, this is the first attempt to use a diffusion model to bridge the gap between teacher and student models in SV.
2) Our experiments demonstrate that DenoKD is resource-friendly and effective for both frame-level features and speaker embeddings, and it also enhances performance on short-duration utterances.

## II. Methodology

### A. Diffusion Model

A diffusion model [23] works by gradually adding noise to the data in each step of a diffusion process and then learns to reverse this process to recover the original data, effectively transforming random noise into meaningful samples. The diffusion process is typically modeled as a Markov chain, where the transition at step $t$ ($t \in \{0, 1, \ldots, T\}$) is described by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{1}$$

where $\mathbf{x}_0$ is a given clean data and $T$ is the number of steps in the diffusion process. $\mathcal{N}(\cdot)$ denotes a normal distribution, and $\alpha_t = 1 - \beta_t$, with $\beta_t$ being the variance at the $t$-th step. The transition from $\mathbf{x}_0$ to $\mathbf{x}_t$ can be expressed as

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i$. Therefore, we can express $\mathbf{x}_t$ as a linear combination of $\mathbf{x}_0$ and noise variable $\boldsymbol{\epsilon}_t \in \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t. \tag{3}$$

In the reverse process, we start from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and gradually denoise it to reconstruct the clean data $\mathbf{x}_0$. The reverse process can be expressed as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \Phi_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}). \tag{4}$$

The reverse process requires a noise estimation model $\Phi_\theta(\mathbf{x}_t, t)$, which is designed to predict the noise present in $\mathbf{x}_t$ relative to $\mathbf{x}_0$. The term $\sigma_t^2$ represents the transition variance in DDIM [24], which helps speed up the denoising process. Instead of progressing step by step from $\mathbf{x}_t$ to $\mathbf{x}_0$, DDIM allows us to take larger steps from $\mathbf{x}_T$ to $\mathbf{x}_0$, i.e., $\mathbf{x}_T \rightarrow \mathbf{x}_{T-\Delta} \rightarrow \cdots \rightarrow \mathbf{x}_0$, where the sampling interval is $\Delta$.

The diffusion model can be trained by minimizing the MSE between the added noise and the predicted noise.

$$\mathcal{L}_{\text{diff}} = \|\boldsymbol{\epsilon}_t - \Phi_\theta(\mathbf{x}_t, t)\|_2^2. \tag{5}$$

### B. Denoising Student Features

In knowledge distillation, due to the difference in the capacity of the teacher and student, the student may not be able to acquire the full knowledge from the teacher, causing noise in the student's features. To address this issue, we use a diffusion model trained on the teacher features to denoise the student model's output features and perform feature-level KD between the original teacher features and the refined student features.

As shown in Fig. 2, we use a diffusion model to denoise the student features at the frame and embedding levels, respectively. For the frame-level DenoKD, we denoise the student features $\tilde{\boldsymbol{F}}^{\mathcal{S}}$ using a diffusion model trained with teacher features $\boldsymbol{F}^{\mathcal{T}}$, where $\mathcal{T}$ and $\mathcal{S}$ represent the teacher and student model, respectively. We start the diffusion process from $\boldsymbol{F}^{\mathcal{T}}$ through $q(\boldsymbol{F}_t^{\mathcal{T}}|\boldsymbol{F}^{\mathcal{T}})$ using Eq. 2, and we perform the reverse process in the form $p_\theta(\boldsymbol{F}_{t-1}^{\mathcal{T}}|\boldsymbol{F}_t^{\mathcal{T}})$ using Eq. 4. Then, we train the diffusion model using Eq. 5.

In the denoising phase, we feed the student features $\boldsymbol{F}^{\mathcal{S}}$ to a linear layer to match its feature dimension with the input of the diffusion model, i.e., the dimension of $\boldsymbol{F}^{\mathcal{T}}$. We also use a noise adapter to ensure that the noise level of the student features matches that of the noisy features at the initialization

step $T$. This is achieved by learning a weight parameter $\gamma$ such that

$$\tilde{\boldsymbol{F}}_T^{\mathcal{S}} = \gamma \boldsymbol{F}^{\mathcal{S}} + (1-\gamma)\boldsymbol{\epsilon}_T. \tag{6}$$

The student features are denoised using the process $p_\theta(\tilde{\boldsymbol{F}}_{t-1}^{\mathcal{S}}|\tilde{\boldsymbol{F}}_t^{\mathcal{S}})$ in Eq. 4. After denoising, we obtain the refined student features $\hat{\boldsymbol{F}}^{\mathcal{S}}$.

The process of embedding-level DenoKD is similar to the frame-level DenoKD. The difference is that the denoised student features used for feature-level knowledge distillation are speaker embeddings $\hat{\boldsymbol{e}}^{\mathcal{S}}$.

### C. Knowledge Distillation with Denoised Student Features

As shown in Fig. 2, knowledge is transferred by minimizing the MSE between the refined student features and the original teacher features.

$$\mathcal{L}_{\text{Feature-KD}} = \|\hat{\boldsymbol{F}}^{\mathcal{S}} - \boldsymbol{F}^{\mathcal{T}}\|_2^2. \tag{7}$$

To enhance performance, we applied label-level KD by minimizing the Kullback-Leibler (KL) divergence between the student's and teacher's output probability distributions:

$$\mathcal{L}_{\text{Label-KD}} = \mathbf{KL}(\boldsymbol{p}^{\mathcal{T}}||\boldsymbol{p}^{\mathcal{S}}) = \sum_{i=1}^{C} p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}}, \tag{8}$$

where $\mathbf{p}^{\mathcal{T}}$ and $\mathbf{p}^{\mathcal{S}}$ denote the output probabilities of the teacher and student networks, respectively, and $C$ is the number of speakers in the training set. The total loss is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Feature-KD}} + \mathcal{L}_{\text{Label-KD}} + \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{cls}}, \tag{9}$$

where $\mathcal{L}_{\text{cls}}$ is the classification loss calculated with the ground truth labels, such as AAMSoftmax loss [25].

## III. Experimental Setup

### A. Datasets

Our models were trained on the VoxCeleb2 [5] development set, which includes 1,092,009 utterances from 5,994 speakers. We followed the data augmentation strategy outlined in Kaldi's recipes [26], i.e., adding noise, music, and babble to the training data using MUSAN [27] and creating reverberated speech data based on RIR [28]. For evaluation, the VoxCeleb1 [29] test set (Vox1-O), which comprises 40 speakers, was used as the evaluation set. To test performance on short-duration utterances, we randomly cropped the test audio into 2s, 3s, and 4s for testing.

### B. Network Training

For the student model, we used ECAPA-TDNN [4] with 512 channels, and the speaker embedding dimension was set to 192. For the teacher model, we used the same configuration as in [15], using WavLM large [12] as the frame-level feature extractor.[1] The dimension of the teacher model's speaker embedding is 256. For frame-level DenoKD, to reduce the computational cost, we used a lightweight diffusion model consisting of two bottleneck blocks of ResNet [3] and a convolutional layer. The noise adapter has one bottleneck block of ResNet, an average pooling layer, and a linear layer with one output node. All convolutional layers implement 1D convolution. For embedding-level DenoKD, the diffusion model contains three stacked-layers, each comprising a convolutional, a batch-normalization, and an activation layer. The embedding-level noise adapter contains two stacked-layers and a linear layer.

For the student model, we extracted 80-dimensional filterbank (Fbank) features from 16 kHz audio signals using a 25ms window with a 10ms frameshift. The batch size was set to 256. We used the AdamW optimizer with a cosine annealing learning-rate scheduler, and a linear learning-rate warm-up scheduler was applied during the first 5 epochs. We used AAM-Softmax loss as the classification loss in all experiments. The AAMSoftmax loss has a scale of 32 and a margin scheduler. The margin was initially set to 0 for the first 20 epochs. It was exponentially increased to 0.2 in the next 20 epochs and was kept constant thereafter. We utilized DDIM [24] as the noise scheduler in the diffusion process. The final step count $T$ for adding noise in the diffusion process was set to 1000. In the denoising process, the initial timestep was set to 500. We used a cosine backend in all experiments.

### C. Evaluation Metrices

The performance metrics include equal error rate (EER) and minimum detection cost function (minDCF) with $P_{\text{target}} = 0.01$. All experiments and tests were conducted based on the 3D-Speaker toolkit [30].[2]

## IV. Results and Discussions

### A. Main Results

Table I shows the performance of various KD methods on Vox1-O using the full duration of the test utterances and the cropped utterances with various durations. Comparing Row 2 with Row 3, we observe that combining $\mathcal{L}_{\text{cls}}$ with $\mathcal{L}_{\text{Label-KD}}$ outperforms that using $\mathcal{L}_{\text{cls}}$ only. However, there is no significant performance improvement when further incorporating a frame-level KD, which can be observed by comparing Row 3 with Row 5. A similar observation is shown (Row 3 and Row 4) when an embedding-level KD is included. These observations suggest that using noisy student features directly for feature-level KD is not effective.

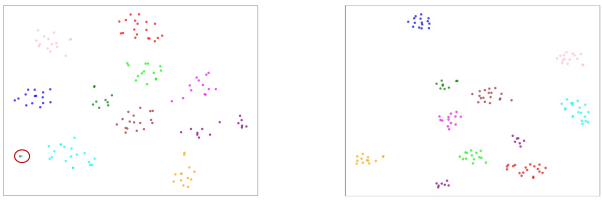
Comparing Row 7 and Row 3, we observe that the embedding-level DenoKD enhances performance by 10% on the full duration of test utterances. The best result was achieved when frame-level denoising was applied, with an impressive EER of 0.78% on VoxCeleb1-O, representing a 13% improvement compared to the 5-th row. This demonstrates that DenoKD can enhance the effectiveness of feature-level knowledge distillation. We observe that DenoKD consistently outperforms all other KD combinations in cases where short

[1] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification

[2] https://github.com/modelscope/3D-Speaker

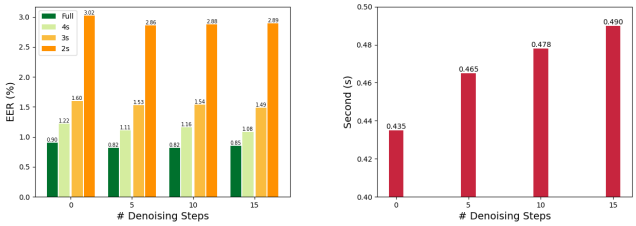
TABLE I: Performance of different KD methods on the VoxCeleb1-O using test utterances of different durations.

| System | Row | Distillation Method | Full EER(%) | Full minDCF | 4s EER(%) | 4s minDCF | 3s EER(%) | 3s minDCF | 2s EER(%) | 2s minDCF |
|---|---|---|---|---|---|---|---|---|---|---|
| *Teacher model* WavLM-TDNN [12] | 1 | – | 0.43 | – | – | – | – | – | – | – |
| *Student model* ECAPA-TDNN | 2 | – | 1.02 | 0.106 | 1.41 | 0.162 | 2.26 | 0.257 | 4.20 | 0.396 |
| | 3 | KD (label) | 0.90 | 0.098 | 1.22 | 0.154 | 1.60 | 0.194 | 3.02 | 0.326 |
| | 4 | KD (label + embedding) | 0.90 | 0.104 | 1.23 | 0.130 | 1.74 | 0.193 | 2.99 | 0.337 |
| | 5 | KD (label + frame) | 0.88 | 0.103 | 1.19 | 0.160 | 1.64 | **0.188** | 3.00 | **0.302** |
| | 6 | DKD | 0.84 | 0.109 | 1.18 | 0.160 | 1.72 | 0.217 | 3.13 | 0.350 |
| | 7 | DenoKD (label + embedding) | 0.82 | **0.097** | 1.11 | **0.125** | 1.53 | 0.201 | 2.86 | 0.319 |
| | 8 | DenoKD (label + frame) | **0.78** | 0.104 | **1.07** | 0.133 | **1.53** | 0.192 | **2.77** | 0.338 |



(a) (b)

Fig. 3: t-SNE feature learned by (a) KD and (b) DenoKD.




(a) (b)

Fig. 4: (a) The EER performance and (b) training time per batch using different numbers of denoising steps in the reverse diffusion process.

test utterances are considered. This indicates that DenoKD enhances the student model's robustness to short audio segments. Additionally, we reproduced the recent state-of-the-art method DKD [21]. By comparing these two methods, we found that DenoKD outperforms DKD in both full-duration and short-duration scenarios, demonstrating its effectiveness.

We used the student model trained with DenoKD and KD to extract speaker embeddings and used t-SNE to visualize them, as shown in Fig. 3. We selected 10 speakers from the VoxCeleb1-dev set, each having 30 utterances. According to Fig. 3, DenoKD generally presents higher cluster compactness and fewer outliers compared with KD. In particular, KD exhibits one incorrect cluster assignment, highlighted with a red circle; on the other hand, such an error does not occur in DenoKD.

### B. Impact of Number of Denoising Steps

For diffusion models, the number of denoising steps is a crucial hyperparameter that impacts performance and computation cost. We conducted experiments using embedding-level DenoKD with varying number of denoising steps, as shown in Fig. 4. Notably, when the number of denoising steps was 0, diffusion model was not applied. The performance is worse when the diffusion model is not used. When the diffusion model is applied, using 5 denoising steps yields the best results on the full-length test utterances. However, as the number of denoising steps increases to 15, the system shows improved performance on short-duration test utterances. Additionally, as the number of denoising steps increases, the training time also rises. However, compared to not using the diffusion model, setting a small number of denoising steps results in only a minimal increase in training time.

TABLE II: Effect of using Noise Adapter in DenoKD.

| Distillation Method | Noise Adapter | EER (%) | minDCF |
|---|---|---|---|
| DenoKD (label + embedding) | ✗ | 0.85 | 0.102 |
| | ✓ | **0.82** | **0.097** |

### C. Effect of Noise Adapter

To validate the effectiveness of the noise adapter, we conducted experiments using DenoKD with and without the noise adapter. As shown in Table II, DenoKD performs better with the noise adapter.

## V. Conclusions

In this paper, we employ a diffusion model trained on the teacher features to denoise the student features for knowledge distillation (DenoKD). By treating the student features as a noisy version of the teacher features and using a diffusion model for multi-step denoising, we obtained refined student features that better align with the teacher features, facilitating feature-level knowledge distillation. Additionally, our experiments demonstrate that DenoKD is effective for both frame-based and speaker-embedding KD, and it can improve the student model's performance on short test utterances.

## REFERENCES


[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[4] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," *in Proc. Interspeech*, pp. 3830–3834, 2020.

[5] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[6] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. ,pp. 77–91, 2022.

[7] L. Li, R. Nai, and D. Wang, "Real additive margin softmax for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7527–7531.

[8] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification." in *Proc. Interspeech*, 2018, pp. 3623–3627.

[9] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[11] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.

[12] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[13] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[14] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6147–6151.

[15] D.-T. Truong, R. Tao, J. Q. Yip, K. A. Lee, and E. S. Chng, "Emphasized non-target speaker knowledge in knowledge distillation for automatic speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024, pp. 10 336–10 340.

[16] L. Xu, J. Ren, Z. Huang, W. Zheng, and Y. Chen, "Improving knowledge distillation via head and tail categories," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. ,pp. 3465–3480, 2023.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *in NIPS Deep Learning and Representation Learning Workshop*, 2015.

[18] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. Cernockỳ, "On the usage of phonetic information for text-independent speaker embedding extraction." in *Proc. Interspeech*, 2019, pp. 1148–1152.

[19] D. Cai and M. Li, "Leveraging ASR pretrained Conformers for speaker verification through transfer learning and knowledge distillation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[20] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "MFA-Conformer: Multi-scale feature aggregation conformer for automatic speaker verification." in *Proc. Interspeech*, 2022, pp. 306–310.

[21] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 953–11 962.

[22] T. Huang, Y. Zhang, M. Zheng, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge diffusion for distillation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[24] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[25] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1652–1656.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE workshop on automatic speech recognition and understanding*, no. CONF, 2011.

[27] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[28] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Proc. 16th International Conference on Digital Signal Processing*, 2009.

[29] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[30] Y. Chen, S. Zheng, H. Wang, L. Cheng, T. Zhu, C. Song, R. Huang, Z. Ma, Q. Chen, S. Zhang *et al.*, "3D-speaker-toolkit: An open source toolkit for multi-modal speaker verification and diarization," *arXiv preprint arXiv:2403.19971*, 2024.