



UNet-DenseNet for Robust Far-Field Speaker Verification

Zhenke Gao, Man-Wai Mak, and Weiwei Lin

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

zhenke.gao@connect.polyu.hk, enmwamak@polyu.edu.hk, weiwei.lin@connect.polyu.hk

Abstract

Far-field speaker verification (SV) has always been critical but challenging. Data augmentation is commonly used to overcome the problems arising from far-field microphones, such as high background noise levels and reverberation effects. On top of data augmentation, this paper tackles these problems by introducing a UNet-based speech enhancement (SE) module as a front-end processor for the speaker embedding module. To prevent the SE module from distorting speaker information, we propose two improvements to the speech enhancement–speaker embedding pipeline. (1) A UNet-DenseNet joint training scheme in which the UNet is optimized by both the MSE and speaker classification losses. (2) A semi-joint training scheme that stops the UNet training but continues the DenseNet training when overfitting of the UNet is detected. Extensive experiments on noise-contaminated Voxceleb1 and the VOICES Challenge 2019 demonstrate the effectiveness of the two training schemes.

Index Terms: Far-field speaker verification, speech enhancement, UNet, DenseNet, speaker embedding

1. Introduction

Speaker verification (SV) is to verify whether an utterance is spoken by the expected person. Over the past decades, the focus of SV has shifted from the statistical methods such as GMM-UBM and i-vector to deep speaker embedding [1, 2, 3, 4, 5, 6]. Although many advanced systems have achieved excellent performance in clean environments, far-field speaker verification is still challenging. Many efforts have been put into far-field SV. Data augmentation is one of the most employed methods. The training set comprises the original audios and augmented audios, which make the SV systems more robust to noise and reverberation. However, in realistic situations, the noise types in the training set and in the deployment environment are different.

Recently, focuses have been shifted to using a cascade of a speech enhancement network and a speaker embedding network to address the noise and reverberation issues. For example, in [7], a DNN was trained to predict the time-frequency binary masks for speech enhancement; reverberation-dependent classifiers then classified the enhanced speech features. Shon *et al.* [8] proposed the VoiceID loss enabling the SE network to generate a ratio mask that retains the necessary components of the spectrogram for the subsequent SV task. However, this method did not generalize very well on unseen noise. The authors in [9] cascaded a speech enhancement module and an attention-based speaker recognition module into one joint training model. However, the SE and SV modules were optimized by a single loss

at the speaker embedding network’s output. In [10], an LSTM-ResNet structure with a novel joint training strategy was proposed to balance the convergence rates of the LSTM and the ResNet. An MSE loss was also injected into the SE part to optimize the LSTM. By doing this, the gradient directions of the two losses will compete with each other, alleviating the information loss caused by the MSE [11]. However, the authors used spectrograms as the input features, which require hundreds of FFT points, leading to feature redundancy. Moreover, LSTM-based front-ends are difficult to train, and parallel computation is difficult to implement during inferencing [12].

When simultaneously optimizing the SE and the speaker embedding networks using two loss functions, the two networks usually exhibit different convergence behaviors. In particular, the SE network will converge much earlier, leading to overfitting if the joint training continues. To address this problem, we propose a semi-joint training strategy in which the weights of the SE network will be frozen once it reaches convergence and the optimization of the speaker embedding network continues.

This paper presents a pipeline for robust far-field speaker verification shown in Fig. 1. We combine UNet [13, 14] and DenseNet [4, 5, 6] into a joint training framework. Besides, we used the channel-wise feature concatenation [10] to further reduce the loss in speaker information due to the SE task. We used filterbank features instead of spectrograms to reduce the feature redundancy. Inspired by early stopping, we propose a semi-joint training method to alleviate the overfitting of the noise in the training data set. Furthermore, we also trained a UNet and DenseNet independently to investigate the information loss caused by the speech enhancement task.

2. UNet-DenseNet

Suppose the speech enhancement (SE) and speaker-embedding modules were optimized separately for their respective tasks. Then, it is not desirable to directly use the SE’s output as the acoustic features for the speaker-embedding network. The reason is that there is no mechanism for the SE module to keep the speaker information, thus reducing the speaker-discriminative power of the enhanced features. Another issue is that the two modules were trained from speech collected from different acoustic environments, causing domain mismatch during inferencing. To address these issues, we propose a UNet-DenseNet model in which the SE module (UNet) and the speaker-embedding module (DenseNet) are jointly trained. A semi-joint training scheme is also developed to prevent the SE module from overfitting the speech enhancement task and thus losing speaker information.

This work was in part supported by the National Natural Science Foundation of China (NSFC), Grant No. 61971371.

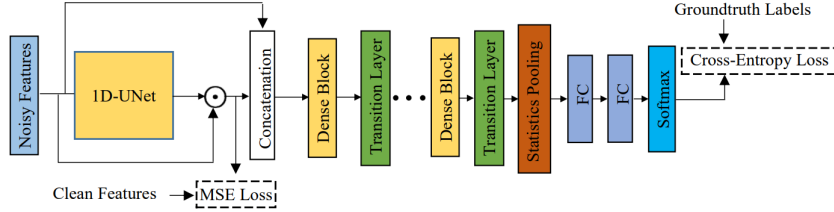


Figure 1: The UNet-DenseNet for robust speaker verification

Table 1: Comparison of different models. “DenseNet” denotes using the DenseNet without speech enhancement for speaker embedding. “VoiceID” means training the UNet by minimizing the cross-entropy loss at the speaker embedding network’s output. “UNet-DenseNet CFC” denotes jointly minimizing the MSE loss and the cross-entropy loss with channel-wise feature concatenation. The SV training set comprises Voxceleb1-dev and its augmented utterances. “D” denotes the development set and “D*” is its augmented set. “Original” shows the performance on the original Voxceleb1 test set.

SV training set		Voxceleb1(D)				Voxceleb1(D*)					
Noise type	SNR	DenseNet		VoiceID		DenseNet		VoiceID		UNet-DenseNet CFC	
		EER	DCF	EER	DCF	EER	DCF	EER	DCF	EER	DCF
Music	20	5.57	0.547	5.67	0.554	5.38	0.531	5.01	0.483	4.61	0.505
	15	6.52	0.607	6.53	0.673	5.78	0.573	5.50	0.513	4.94	0.541
	10	8.54	0.702	8.17	0.717	6.69	0.615	6.41	0.605	5.61	0.592
	5	12.05	0.853	11.18	0.835	8.57	0.749	8.06	0.739	7.18	0.661
	0	18.03	0.941	16.09	0.909	11.56	0.866	11.26	0.846	9.62	0.752
Noise	20	6.19	0.601	6.27	0.591	5.47	0.561	5.14	0.528	4.93	0.531
	15	7.58	0.678	7.45	0.624	5.96	0.599	5.72	0.543	5.16	0.548
	10	9.58	0.723	8.95	0.711	6.88	0.657	6.48	0.628	5.89	0.623
	5	12.47	0.805	10.89	0.769	7.91	0.675	7.62	0.657	7.36	0.652
	0	16.98	0.868	14.79	0.841	9.88	0.784	9.83	0.745	8.74	0.745
Babble	20	5.49	0.549	5.88	0.574	5.39	0.531	5.13	0.528	4.75	0.484
	15	6.75	0.632	7.07	0.672	6.65	0.587	6.15	0.574	5.43	0.558
	10	9.74	0.727	9.91	0.736	9.23	0.693	8.73	0.689	7.64	0.674
	5	15.56	0.911	15.19	0.903	14.07	0.859	13.57	0.864	12.54	0.839
	0	24.53	0.984	23.59	0.975	22.88	0.976	21.99	0.975	20.95	0.985
Original		4.70	0.506	5.12	0.499	4.80	0.485	4.56	0.451	4.28	0.472
Average		10.64	0.727	10.17	0.724	8.57	0.671	8.19	0.648	7.48	0.637

2.1. DenseNet for Deep Speaker Embedding

A speaker embedding network consists of several frame-level layers, a statistics pooling layer, and several utterance-level layers. A commonly used frame-level layer is the time delayed neural network (TDNN) [1]; it maps a variable-length utterance to intermediate frame-level features. A recent trend in speaker embedding is to enhance the capability of frame-level processing by replacing the TDNN with ResNet [2, 3] and DenseNet [4, 5]. We employed a DenseNet with a structure shown in Table 2 for the frame-level processing in our model.

2.2. UNet-based Speech Enhancement

Speech enhancement [16] aims to recover the target clean signal from the corrupted signal. Specifically, the clean speech $s(n)$ is assumed to be corrupted by background noise $b(n)$ through an additive model in the time domain: $y(n) = s(n) + b(n)$, where $y(n)$ is the observed noisy speech. To use a 2D-UNet for speech enhancement, the short time Fourier transform (STFT) is applied to noisy speech so that noisy spectrograms are used as the input. Fig. 2 shows the UNet architecture.

Denote the STFT of the clean and noisy speech as $S_{t,f}$ and $Y_{t,f}$, respectively, where t indexes the frames and f indexes the frequency bins. Denote $M_{t,f}$ as the ideal ratio mask (IRM)

in the time-frequency domain for recovering the clean spectrogram from the noisy spectrogram, i.e., $S_{t,f} = Y_{t,f} \odot M_{t,f}$. Our goal is to train a UNet to predict $M_{t,f}$ from the noisy spectrogram $Y_{t,f}$. Because the prediction can never be perfect, we can only partially recover the clean spectrogram, i.e., $\hat{S}_{t,f} = Y_{t,f} \odot \hat{M}_{t,f}$, where $\hat{M}_{t,f}$ is the predicted ratio mask. The UNet is trained to minimize the mean squared error (MSE) between the clean and the recovered spectrograms:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{TF} \sum_{t=1}^T \sum_{f=1}^F (S_{t,f} - \hat{S}_{t,f})^2. \quad (1)$$

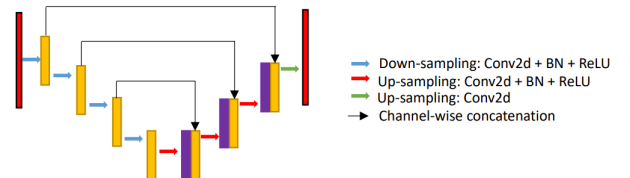


Figure 2: The architecture of UNet for speech enhancement

Because the dimensionality of STFT features is too high for the speaker embedding network, a set of filter banks are applied

Table 2: The configuration of the DenseNet used in our model. “conv” denotes the sequence BN-ReLU-conv1d, where BN is batch normalization [15]. “conv k” represents one-dimensional convolution with kernel size k.

Layer	output size	Operation
Convolution	80×400	conv 3
Dense Block 1	320×400	$\begin{bmatrix} \text{conv } 1 \\ \text{conv } 3 \end{bmatrix} \times 6$
Transition Layer 1	160×200	conv 2 stride 2
Dense Block 2	560×200	$\begin{bmatrix} \text{conv } 1 \\ \text{conv } 3 \end{bmatrix} \times 10$
Transition Layer 2	280×100	conv 2 stride 2
Dense Block 3	840×100	$\begin{bmatrix} \text{conv } 1 \\ \text{conv } 3 \end{bmatrix} \times 14$
Transition Layer 3	420×50	conv 2 stride 2
Dense Block 4	820×50	$\begin{bmatrix} \text{conv } 1 \\ \text{conv } 3 \end{bmatrix} \times 10$
Stats-pooling Layer	820	-
FC1	512	-
FC2	256	-
Softmax Layer	# of classes	-

to the STFT features to reduce the dimension to 40. The resulting features are commonly called filterbank features. To make the UNet amendable to speaker embedding extraction, we use filterbank features for the input and output of the UNet. Because the dimension is now reduced to 40, we applied 1D-UNet (with 1D convolution) instead of 2D-UNet in our work. Therefore, f in Eq. 1 indexes the filters in the filterbank instead of the frequency bins.

2.3. Multi-objective Learning

To achieve noise robust speaker verification, a speech enhancement network can be employed to recover the clean signal. UNet is an excellent speech enhancement network for this purpose. However, the speaker information in the enhanced features could be distorted because the network is optimized to reduce the MSE loss, leading to performance degradation in the subsequent speaker embedding network. We propose a multi-objective function to optimize the UNet and DenseNet to overcome this problem. Specifically, the speaker classification loss will be backpropagated to the UNet to prevent it from distorting speaker information when minimizing the MSE loss.

2.4. Channel-wise Feature Concatenation

To maintain a good representation quality, we apply channel-wise feature concatenation (CFC) [10] to combine the corrupted input and the enhanced features. The method aims at maintaining both the original representation and the enhanced representation. At high SNR, the original representation contains uncontaminated speaker information, facilitating the speaker embedding network to extract speaker information. At low SNR, the noise will mask the speaker information in the original representation. In such case, the enhanced representation becomes the primary source of speaker information.

2.5. Semi-joint Optimization

It is challenging to ensure that different learning tasks will converge at a comparable rate during training. Most often, a task

may have already converged while the others are still in the course of convergence. In SE-SV, if the SE task converges much earlier than the SV task, the SE module will over-fit the seen noise. To overcome this problem, we propose a semi-joined optimization method that leverages the idea of early stopping in DNN training. Specifically, instead of jointly train the UNet and DenseNet from the beginning to the end, we stop training the UNet once it converges and continue training the DenseNet while freezing the weights of the UNet.

3. Experimental Setup

3.1. Data preparation

Experiments were conducted on the Voxceleb1 dataset [17] and the VOICES Challenge 2019 evaluation set [18] (Voices19c-eval). The training set comprises 1251 speakers and 148,642 utterances from the Voxceleb1 development set (Voxceleb1-dev). We evaluated the model’s performance on the Voxceleb1 test set (Voxceleb1-test) and the Voices19c-eval. In Voxceleb1-dev, we used the Musan dataset [19] to augment the training utterances and to contaminate the test utterances in Voxceleb1-test.

There are three types of noise in Musan: noise, speech, and music. We divided noise and music into two disjoint sets. One set was used for augmenting the training utterances with the SNR uniformly distributed between 0dB and 20dB, while the other set and the “us-gov” speech were used to contaminate the test set at an SNR of 0dB, 5dB, 10dB, 15dB, and 20dB, respectively. This procedure assures that the noise types in the test sessions are unseen.

3.2. Network Architecture and Network Training

For each audio file in the training set, a 4-second segment was randomly cut from the waveform. Then, the segment was converted to 40-dimensional filterbank feature vectors using a 25-ms window with a frameshift of 10ms. No mean-normalization and voice activity detection were performed. Stochastic gradient descent (SGD) with a momentum of 0.95 was employed to optimize the models. We used the CosineAnnealingWarmRestarts scheduler in PyTorch [20] to update the learning rate. The initial learning rate was set to 0.01. And 128 shuffled samples were grouped into a batch. An 1D-UNet was employed in the de-noising part. During the testing stage, the whole utterance in each audio file was used. The equal error rate (EER) and minimum detection cost function (minDCF) were used as the performance metrics.

UNet: The UNet has 12 layers, with 6 layers in the encoder and 6 layers in the decoder. The numbers of output channels in the encoder are 64, 128, 128, 256, 256, and 512, respectively. For the decoder, they are 256, 256, 128, 128, 64, and 40. The kernel size of the 1D-UNet was set to 3 for the first 3 layers of the encoder and was set to 5 for the last 3 layers of the encoder. The decoder is symmetric to the encoder.

DenseNet: We used an 80-layer DenseNet for speaker embedding. The architecture is shown in Table 2.

VoiceID: The SE-SV network is composed of a UNet and a pre-trained DenseNet. The parameters of the DenseNet were frozen while optimizing the UNet using the cross-entropy loss at the DenseNet’s output.

UNet-DenseNet CFC: It is a cascade of an 1D-UNet and an 80-layer DenseNet, where channel-wise feature concatenation (see Fig. 1) was adopted.

Table 3: Comparison of separate training and joint training.

SV training set		Voxceleb1(D^*)			
Noise type	SNR	Separate		Joint CFC	
		EER	DCF	EER	DCF
Music	20	4.53	0.461	4.61	0.505
	15	5.08	0.491	4.94	0.541
	10	5.79	0.562	5.61	0.592
	5	7.38	0.669	7.18	0.661
	0	10.83	0.823	9.62	0.752
	-5	16.77	0.940	14.69	0.892
Noise	20	4.65	0.482	4.93	0.531
	15	5.24	0.503	5.16	0.548
	10	5.75	0.576	5.89	0.623
	5	7.27	0.676	7.36	0.652
	0	9.26	0.783	8.74	0.745
	-5	13.07	0.882	11.13	0.832
Babble	20	4.68	0.471	4.75	0.484
	15	5.71	0.546	5.43	0.558
	10	7.91	0.642	7.64	0.674
	5	12.48	0.822	12.54	0.839
	0	20.84	0.967	20.95	0.985
	-5	30.78	0.999	30.54	0.999
Original		4.20	0.447	4.28	0.472
Average		9.59	0.671	9.26	0.678

4. Results

4.1. Comparing Training Schemes

Table 1 shows the performance of our jointly trained model (UNet-DenseNet CFC) and other models. It shows that the UNet-DenseNet performs the best among all models, which achieves over 5% reduction in EER compared with the DenseNet and with the UNet cascaded with a pre-trained DenseNet (VoiceID).

Table 3 shows the performance of separate training and joint training. ‘‘Separate’’ means training a UNet and DenseNet separately and then joined them together. The column ‘‘Joint’’ means that we jointly trained the UNet and DenseNet. We observed that separate training performs worse at low SNR. Therefore, we added a row to show the performance at -5 dB. Table 3 shows that although separate training can achieve comparable performance (or even better) at high SNR, the performance degrades sharply at low SNR such as 0dB and -5 dB. The result shows that joint training achieves a relative improvement of 2.8% in average EER and a 7% relative improvement at SNR of -5 dB.

Table 4 and Table 5 show that semi-joint training leads to obvious improvement compared with joint training. Also, semi-joint training performs better on unseen babble noise and realistic noise, with a relative reduction of 7.2% in EER and 7.5% in minDCF.

We applied weighted-prediction error [21] to de-reverberate the speech in Voices19c-eval, followed by presenting the de-reverberated speech to the UNet-DenseNet. Table 5 shows the results. Despite the lack of training data and small model size (which cause higher than usual EER in Voice19c-eval), semi-joint training method achieves a relative improvement of 6.8% in EER and 2% in minDCF compared with joint training, which suggests that the UNet-DenseNet architecture together with semi-joint training can generalize to unseen realistic noise.

Table 4: The performance of semi-joint training and joint training on the Voxceleb1 test set.

SV training set		Voxceleb1(D^*)			
Noise type	SNR	Joint CFC		Semi-joint CFC	
		EER	DCF	EER	DCF
Music	20	4.61	0.505	4.32	0.468
	15	4.94	0.541	4.54	0.463
	10	5.61	0.592	5.30	0.566
	5	7.18	0.661	6.52	0.631
	0	9.62	0.752	8.89	0.762
	Noise	20	4.93	0.531	4.49
15		5.16	0.548	4.79	0.509
10		5.89	0.623	5.47	0.567
5		7.36	0.652	6.37	0.602
0		8.74	0.745	7.98	0.683
Babble		20	4.75	0.484	4.46
	15	5.43	0.558	5.35	0.485
	10	7.64	0.674	7.24	0.610
	5	12.54	0.839	11.89	0.798
	0	20.95	0.985	19.46	0.968
	Original	4.28	0.472	4.07	0.425
Average		7.48	0.637	6.94	0.589

Table 5: The performance of semi-joint training and joint training on the Voices19c test set.

Model	Semi-Joint	Joint
EER (%) /minDCF	11.68/0.883	12.54/0.901

4.2. Analysis of the Ratio Masks

To further investigate how joint training and separate training affect the ratio masks. We selected an audio file from the test set and augmented it with fixed random noise at an SNR of -5 dB. Then we present the noisy filterbank features to the UNet. The masks produced by the UNet are shown in Fig. 3, which shows that the mask produced by separate training can remove more noise (shown in the red circle region, with dark means the corresponding components will be de-noised after masking). However, it may destroy the useful frequency components beneficial for speaker embedding.

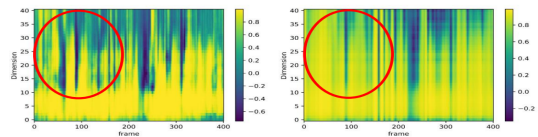


Figure 3: The masks produced by separate training (left) and joint training (right).

5. Conclusions

We proposed a UNet-DenseNet pipeline for far-field speaker verification. The results show that the proposed joint training scheme performs much better than separate training at low SNR on the Voxceleb1 test set. A semi-joint training method is proposed to optimize the joint model so that the convergence of the SE and SV modules are balanced. Results on VOICES Challenge 2019 show that the semi-joint training method leads to a model that can generalize well on unseen noise.

6. References

- [1] 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.
- [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 (CVPR)*, 2016, pp. 770–778.
- [3] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, "Deep discriminative embeddings for duration robust speaker verification." in *Proc. Interspeech*, 2018, pp. 2262–2266.
- [4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.
- [5] W. W. Lin and M. W. Mak, "Mixture representation learning for deep speaker embedding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 968–978, 2022.
- [6] Y. Jiang, Y. Song, I. V. McLoughlin, Z. Gao, and L.-R. Dai, "An effective deep embedding learning architecture for speaker verification." in *Proc. Interspeech*, 2019, pp. 4040–4044.
- [7] X. Zhao, Y. Wang, and D. L. Wang, "Robust speaker identification in noisy and reverberant conditions," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 4, pp. 836–845, 2014.
- [8] S. Shon, H. Tang, and J. Glass, "Voiceid loss: Speech enhancement for speaker verification," in *Proc. Interspeech*, 2019, pp. 2888–2892.
- [9] Y. Shi, Q. Huang, and T. Hain, "Robust Speaker Recognition Using Speech Enhancement And Attention Model," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020, pp. 451–458.
- [10] Y. Wu, L. Wang, K. A. Lee, M. Liu, and J. Dang, "Joint feature enhancement and speaker recognition with multi-objective task-oriented network," in *Proc. Interspeech*, 2021, pp. 1089–1093.
- [11] Z. Oo, Y. Kawakami, L. Wang, S. Nakagawa, X. Xiao, and M. Iwahashi, "DNN-based amplitude and phase feature enhancement for noise robust speaker identification." in *Proc. Interspeech*, 2016, pp. 2204–2208.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.
- [14] H. S. Choi, J. Kim, J. Huh, A. Kim, J. W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex U-Net," in *Proc. International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=SkeRTsAcYm>
- [15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning*, 2015, pp. 448–456.
- [16] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.
- [17] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.
- [18] M. K. Nandwana, J. van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *CoRR*, vol. abs/1902.10828, 2019. [Online]. Available: <http://arxiv.org/abs/1902.10828>
- [19] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.
- [20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [21] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.