

Real-Time Noise Cancellation and Noise Suppression For Enhanced Speech Clarity using Speech Separation Models

Balacharan V¹ and Sankar Kiran M¹
Narmatha B² and Rathnamala S²

¹Department of Artificial Intelligence and Data Science
Sethu Institute of Technology, Pulloor, Kariapatti

17v.cheran@gmail.com, sankarkiran5758@gmail.com

Abstract

This paper introduces a dual-signal transformation LSTM network (DTLN) for real-time speech enhancement as part of the Deep Noise Suppression Challenge (DNS-Challenge). This approach combines a short-time Fourier transform (STFT) and a learned analysis and synthesis basis in a stacked-network approach with less than one million parameters. The model was trained on 500 h of noisy speech provided by the challenge organizers. The network is capable of real-time processing (one frame in, one frame out) and reaches competitive results. Combining these two types of signal transformations enables the DTLN to robustly extract information from magnitude spectra and incorporate phase information from the learned feature basis. The method shows state-of-the-art performance and outperforms the DNS-Challenge baseline by 0.24 points absolute in terms of the mean opinion score (MOS).

Index Terms: noise suppression, deep-learning, real-time, speech enhancement, deep learning, audio

1. Introduction

The task of noise suppression is an important discipline in field of speech enhancement; it is for instance of special importance in work-from-home scenarios where a robust and effective noise reduction can improve the communication quality and thereby reduce the cognitive effort of video conferencing. With the up-rising of deep neural networks, several novel approaches for audio processing methods based on deep models were proposed [1, 2, 3, 4]. However, these have often been developed for offline processing which does not require real-time capabilities or the consideration of causality in the processing chain. Such models process complete sequences and exploit past and future information of the signals to suppress undesired signal parts. Classic signal processing algorithms [5, 6] often work on sample or frame level to provide a low input-output delay. When designing frame-based algorithms with neural networks, recurrent neural networks (RNN) are a common choice. RNNs have produced convincing results in the field of speech enhancement [7, 8] and speech separation [9, 10, 11]. Long short term memory networks (LSTM) [12] represent the state-of-the-art in separation [13]. The best-performing networks are often build in a non-causal way by using bidirectional LSTMs where the time sequence is processed causally as well in the reversed direction. Bidirectional RNNs always require a full sequence as input and are therefore principally not suited for real-time frame processing.

The baseline system of the the deep-noise-suppression challenge (DNS-Challenge) [14] is called NSNet [15] and is also based on RNN layers and provides real-time capability by calculating one output frame per input frame. Based on the log power spectrum of the short-time Fourier transform (STFT) of the noisy time signal, this model predicts a gain or mask which

is applied to a noisy STFT. The predicted speech signal is reconstructed by using the estimated magnitude and the phase of the noisy mixture. This approach results in a competitive baseline system, but it does not incorporate any phase information, which could be useful for enhanced speech quality. Different approaches are tackling phase estimation such as estimating the masks for the real and imaginary part of the STFT instead of the magnitude [16] or calculating an iterative phase reconstruction [17]. Research studies such as [11, 18, 19] have shown promising results for speaker separation tasks with a learned analysis and synthesis basis that is not decoupling magnitude and phase information. The representation is calculated by multiplying time-domain frames with learned basis functions. This approach was also applied in [20] for separating speech and noise.

The motivation of the current study is to merge both analysis and synthesis approaches in one model by using a stacked dual signal transformation LSTM network (DTLN). Stacked or cascaded networks were already used in the Deep Clustering speaker separation approach [9] where an additional enhancement network was added after the separation network. In related research, cascaded models were used for denoising and dereverberation [19]. The proposed model presented here cascades two separation cores, the first features an STFT signal transformation while the second used a learned signal representation similar to [18]. This order was chosen to create a robust magnitude estimation with the first core and enable the second core to further enhance the signal with phase information. This combination is explored for the first time in the context of noise reduction and could provide beneficial effects due to the complementarity of classic and learned features transformations while maintaining a relatively small computational footprint. The stacked network in this paper is considerably smaller as most previously proposed LSTM networks and ensures real-time capability in terms of computational complexity.

2. Methods

2.1. Signal transformations

In speaker separation, a time-frequency masking approach is often chosen to separate the speakers' signals. Noise suppression is a related source separation problem, but is different in that it only returns the speech signal and discards the noise. In the time frequency domain, the separation problem can be formulated as follows: The microphone signal y is described by

$$y[n] = x_s + x_n \quad (1)$$

where x_s and x_n are the speech and noise components of time signals, respectively.

In a noise suppression task, the desired signal is the speech signal. When the signal y is transformed with an STFT in a

complex time-frequency representation (TF), the TF representation of the estimated speech signal \hat{X}_s can be predicted as follows:

$$\hat{X}_s(t, f) = M(t, f) \cdot |Y(t, f)| \cdot e^{j\varphi_Y}, \quad (2)$$

where $|Y|$ is the magnitude of the STFT of y . M is a mask (with masking values ranging from 0 to 1) that is applied to Y , and $e^{j\varphi_Y}$ is the phase of the noisy signal. \hat{X}_s can now be transformed back with an inverse STFT to \hat{x}_s . In this formulation, the phase of the noisy signal is used to predict the clean speech signal.

The second signal transformation of the DTLN was first proposed by Luo and colleagues [11]. The formulation of the approach is described in the following: The mixture is split into overlapping frames y_k of length L with frame index k . The frames are multiplied by U , which has $N \times L$ learned basis functions

$$w_k = y_k U \quad (3)$$

to create the feature representation w_k with dimension $N \times 1$ of frame y_k . To recover the speech representation d_k from w_k , a mask m_k can be estimated given by

$$\hat{d}_k = m_k \cdot w_k \quad (4)$$

where \hat{d}_k is the feature representation at index k of the estimated speech signal. \hat{d}_k can be transformed back to the time domain by

$$\hat{x}_k = \hat{d}_k V, \quad (5)$$

where V contains N learned basis functions of length L . \hat{x}_k is the estimated frame at index k . The estimated time signal \hat{x}_s is reconstructed by using an overlap-add procedure.

2.2. Network architecture

The stacked dual-signal transformation LSTM network architecture introduced in this paper has two separation cores containing two LSTM layers followed by a fully-connected (FC) layer and a sigmoid activation to create a mask output. The first separation core uses an STFT analysis and synthesis base. The mask predicted by the FC layer and the sigmoid activation is multiplied by the magnitude of the mixture and transformed back to the time domain using the phase of the input mixture, but without reconstructing the waveform. The frames coming from the first network are processed by an 1D-Conv layer to create the feature representation. The feature representation is processed by a normalization layer before it is fed to the second separation core. The predicted mask of the second core is multiplied with the unnormalized version of the feature representation. The result is used as input to a 1D-Conv layer for transforming the estimated representation back to the time domain. In a last step, the signal is reconstructed with an overlap and add procedure. The architecture is visualized in Figure 1.

To account for the real-time character of the model, instant layer normalization (iLN) is used. Instant layer normalization is similar to standard layer normalization [21] and was introduced as channel-wise layer normalization in [22]. All frames are normalized individually without accumulating statistics over time and are scaled with the same learnable parameters. In the current work, this normalization scheme is referred to as instant layer normalization to differentiate from cumulative layer normalisation [18].

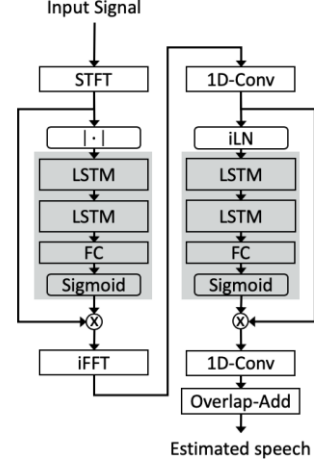


Figure 1: Illustration of the proposed network architecture. The processing chain on the left shows the first separation core using the STFT signal transformation while the building blocks on the right represent the second core with learned feature transformations based on 1D-Conv layers.

2.3. Datasets

The training dataset was created from the provided audio data of the DNS-Challenge. The speech data is part of the Librispeech corpus [23], and the noise signals originated from the Audioset corpus [24], Freesound and DEMAND corpus [25]. 500 h of data were created by using the provided scripts. The default SNR range (0 to 40 dB) was changed to -5 to 25 dB to include negative SNRs and limit the total range. To cover a more fine-grained SNR distribution, the number of SNR levels was increased from 5 to 30. All further parameters remained unchanged. The 500 h dataset was divided into training (400 h) and cross validation data (100 h), which corresponds to the common 80:20 % split. All training data was sampled at 16 kHz.

The challenge organizers also provided a test set which contains four different categories each containing 300 samples. The categories are synthetic clips without reverb, synthetic clips with reverb, real recordings collected internally at Microsoft and real recordings from Audioset. The synthetic data was taken from the Graz University’s clean speech dataset [26]. The SNRs of the synthetic data were randomly distributed from 0 to 25 dB SNR. The impulse responses of the reverberant data were measured in multiple rooms at Microsoft with reverberation times (RT60) ranging from 300 to 1300 ms. Further, a blind test set was created by the organizers which is evaluated in an ITU P-808 [27] setup. The full details of training and test sets are provided in [14].

To correctly estimate the performance with all objective measures in a noisy reverberant environment, the reverberant single speaker and noise test set of the WHAMR corpus [19] at 16 kHz sampling frequency was used. We turned to this dataset because some objective measures need a properly delayed but clean reference signal for correct calculation. Since these signals are not provided in the DNS-Challenge test set, we used the WHAMR dataset, which has clean non-reverberant speech files accounting for the delay of the impulse response. The used WHAMR test set consists of 3000 mixtures. The speech files are taken from the WSJ0-mix corpus [28] that are often used in speaker separation. The speech files are convolved with room impulse responses with the RT60 ranging from 100 to 1000 ms

that were simulated with Pyroomacoustics [29]. The noise consists of real-life recordings of situations such as coffee shops, restaurants, bars, office buildings and parks. The SNRs range from -3 to 6 dB relative to the speech.

2.4. Model configuration and training setup

The DTLN in this paper¹ has 128 units in each of its four LSTM layers. The frame size is 32 ms and the shift 8 ms. The FFT size is 512 and equal to the frame length. The 1D-Conv Layer to create the learned feature representation has 256 filters. During training, 25% of dropout is applied between the LSTM layers. The Adam optimizer is used with a learning rate of 10e-3 and a gradient norm clipping of 3. The learning rate is halved if the loss on the validation set does not improve for three consecutive epochs. Early stopping is applied if loss on the validation set does not decrease for ten epochs. The model is trained on a batch size of 32, and each sample has the length of 15 s. The average time for one training epoch on a Nvidia RTX 2080 TI is around 21 minutes.

As training objective the scale-sensitive negative SNR [20] was used. Compared to the Scale Invariant Signal to Noise Ratio (SI-SNR) [11] it should avoid possible level offsets between input mixture and predicted cleaned speech, which is desirable in real-time-processing systems. Also, since it operates in time domain, the phase information can implicitly be considered. In contrast the Mean Squared Error between the estimated and clean magnitude STFT of the speech signal as training objective is not able to use any phase information in the optimization process.

2.5. Baselines

The first baseline is the noise suppression network (NSNet) provided by the challenge organizers. NSNet was optimized with an MSE-based speech distortion weighted loss in frequency domain and was trained on a rather small corpus of 84 h of speech and noise mixtures. It consists of three recurrent layer with 256 gated recurrent units (GRU) [30] and a fully-connected layer with sigmoid activation for mask prediction. The frame size is 20 ms and the frame shift 10 ms. GRUs are similar to LSTMs but without a cell state passed over time.

Additionally, our DTLN method is compared to four models with the same training setup as the proposed model: These models are explored to quantify the effect of using one feature representation only in two different topologies (stacked versus densely-connected LSTMs): The first and the second model consist of four LSTM layers followed by a fully-connected layer with a sigmoid activation to predict a mask. The first one (B1) uses an STFT analysis and synthesis basis, while B2 used a learned basis of size 256. The third (B3) and the fourth model (B4) are stacked models similar to the proposed method. Both separation kernels of B3 are using an STFT base. B4 has a learned feature base of size 256 for both separation kernels. The size of the LSTM layers is chosen with the aim of obtaining a similar size as the DTLN method in terms of the number of parameters. The configurations are again shown in Table 1.

2.6. Objective and subjective evaluation

For comparison of the DTLN approach and the baselines, we use three objective measures, i.e., the Perceptual Evaluation of

Table 1: Number of parameters and RNN units in each layer for the proposed DTLN approach as well as for the baseline systems.

Method	# Params	# Units
NSNet (3 Layer, STFT)	1.27 M	256
B1 (4 Layer, STFT)	988 K	166
B2 (4 Layer, learned)	984 K	139
B3 (2x2 Layer, STFT)	988 K	156
B4 (2x2 Layer, learned)	987 K	95
DTLN (2x2 layer, STFT+learned)	987 K	128

Speech Quality (PESQ) [31], the Scale-Invariant Signal to Distortion Ratio (SI-SDR) [32] and the Short Time Objective Intelligibility measure (STOI) [33].

The subjective evaluation was performed with the a ITU-T P.808 setup on the Amazon mechanical Turk (AMT) implemented and organized by Microsoft. In total, there were two evaluation runs, one on the known test data set of the DNS-Challenge and one on a blind test set provided later on. Each file was rated by five or ten judges in the first and second run, respectively.

3. Results

The results of the objective evaluation are shown in Table 2 and the subjective evaluation in Table 3. The results are described in the following:

Objective results for the non-reverberant DNS-Challenge test set: In the non-reverberant condition, all models produce improvements over the noisy condition. NSNet is outperformed by the DTLN and all additional baselines. All models trained on 500 h of data are producing similar results. The best results in terms of PESQ, SI-SDR and STOI were reached by the DTLN network. The high values obtained with B3 and the DTLN show the strength of stacked models. Even though B4 is also a stacked model, it performed considerably worse, which is discussed in Section 4.

Objective results for the reverberant DNS-Challenge test set: In this condition, results are not as clear as in the non-reverberant condition. In terms of PESQ, only B4 shows a slight improvement over the noisy condition. For the SI-SDR all models show an improvement, while STOI predicts the highest quality for the original noisy condition. One issue with the intrusive or double-ended measures is that they require a reference signal which is in this case the reverberant clean speech. With this reference signal a potential dereverberation effect by any speech enhancement model would result in a decrease of the objective measures, which presumably is an important factor for these results.

Objective results on the WHAMR test set: All methods are showing an improvement over the noisy condition, with the best scores obtained by the DTLN approach. Similar performance levels are again reached by B3. The baseline shows just a slight improvement for all objective measures. It should be mentioned that the mixtures used in this corpus have a smaller SNR range around 0 and is therefore a more challenging condition for the models.

Subjective results on the DNS-Challenge test sets: The subjective results for the known non-reverberant test set are in line with the objective results. For the reverberant test set, the sub-

¹A Keras implementation of the DTLN can be found at <https://github.com/breizhn/DTLN>

Table 2: Results in terms of PESQ [MOS], SI-SDR [dB] and STOI [%] of the non reverberant test set, the reverberant test set of the DNS challenge and the reverberant single mix test set of the WHAMR corpus.

	DNS test set no reverb			DNS test set with reverb			WHAMR test set		
Method	PESQ	SI-SDR	STOI	PESQ	SI-SDR	STOI	PESQ	SI-SDR	STOI
Noisy	2.45	9.07	91.52	2.75	9.03	86.62	1.83	-2.73	73.00
NSNet	2.70	12.47	90.56	2.47	9.18	82.15	1.91	0.34	73.02
B1	3.00	16.05	94.53	2.75	11.33	85.41	2.20	1.95	79.93
B2	3.00	15.87	94.22	2.74	10.92	85.05	2.18	1.88	79.34
B3	3.03	16.27	94.74	2.70	10.84	84.80	2.23	1.94	80.23
B4	2.96	15.51	93.86	2.76	10.77	84.90	2.20	1.80	78.90
DTLN	3.04	16.34	94.76	2.70	10.53	84.68	2.23	2.12	80.40

jective evaluation shows a clear benefit for DTLN relative to the noisy condition and the baseline. This effect is not reflected by the objective measures with exception of the SI-SDR, which shows some improvement over the baseline and the noisy condition. The decrease in quality of the NSNet predicted by PESQ and STOI in the reverberant condition is also observed in the subjective data. Consistent results are obtained with the real recordings, both for the conditions *known* and *blind*.

Table 3: Subjective ratings in terms of the MOS for the *known* and *blind* test set of the DNS-Challenge. The overall 95% confidence intervals for the *known* and *blind* test sets are 0.04 and 0.02, respectively.

Method	No reverb		With reverb		Real recordings	
	known	blind	known	blind	known	blind
Noisy	3.02	3.32	2.44	2.78	3.01	2.97
NSNet	3.14	3.49	2.16	2.64	2.99	3.00
DTLN	3.41	3.58	2.56	2.95	3.14	3.21

Results on execution time: In the context of the DNS-Challenge, the execution time of one 32 ms frame on a quad-core I5 6600K CPU was measured. The measurement was performed by either processing a complete sequence or by using frame-wise processing. Execution times of 0.23 ms and 2.08 ms were measured for the sequence and frame-wise processing, respectively. The large difference between sequence and frame processing can be explained by the overhead caused by calling models for prediction in Keras. Converting the model to Tensorflow’s SavedModel format reduces the execution time for frame by frame processing to 0.65 ms, which is a large improvement. However, the sequence processing time is nearly three times lower and demonstrates the potential performance on a CPU.

4. Discussion

In the following, we first discuss differences between baseline systems, which also has implications for the components of the DTLN system. The results on the non-reverberant, the reverberant and the WHAMR test are showing better results for the systems B1 and B3 (using STFT features) than for B2 and B4 (that used learned feature representations). One potential reason for the better performance with STFTs is the fixed number of parameters across networks, and - since the STFT is fixed and rule-based - it is possible that B1 and B3 exploit the higher

number of parameters available for LSTM layers in comparison to learned-feature approaches.

Secondly, we assume that STFT features provide a higher robustness for noisy input since phase information - which is not useful in high-noise conditions - is discarded. Vice versa, networks using learned features have to determine masks implicitly for both magnitude and phase information. Another possible reason for the difference could be the compression which is performed by the learned feature representation in this work. The learned feature representation maps 512 audio samples to a feature representation of size 256. Feature representations with greater size would have cost even more parameters and it was empirically found that the reduction of the feature representation doesn’t have a great impact on the speech quality of the proposed model.

The results also show that stacking networks using STFT and learned feature transformation slightly improves overall baseline systems by using fewer LSTM units than the pure STFT systems. LSTM units are computationally more complex as fully-connected or 1D-Conv layers, i.e., a reduction of units is especially desirable for this network type. However, the relatively small difference in terms of objective measures between DTLN and the related systems (B1-B4) also suggests that a part of the performance is generated by the large amount of training data and the training setup.

5. Conclusions

This paper introduced an approach for noise suppression based on a stacked dual signal transformation LSTM network for real-time speech enhancement, which was trained on a large-scale data set. We were able to show an advantage of using two types of analysis and synthesis bases in a stacked network approach. The DTLN model works robustly in noisy reverberant environments. Although we combined a basic training setup with a straight-forward architecture, we observed absolute improvements of 0.22 in terms of MOS over all subjective evaluations relative to the noisy conditions.

6. Acknowledgements

This research was supported by the DFG (Cluster of Excellence 1077/1 Hearing4all; URL: <http://hearing4all.eu>). The architecture was partially developed on a GPU donated by the Nvidia GPU Grant program. Thanks goes to challenge organizers from Microsoft for conducting the DNS-Challenge and providing the data and the scripts.

7. References

- [1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “An experimental study on speech enhancement based on deep neural networks,” *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.
- [2] K. Han, Y. Wang, and D. Wang, “Learning spectral mapping for speech dereverberation,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4628–4632.
- [3] S. Pascual, A. Bonafonte, and J. Serra, “Segan: Speech enhancement generative adversarial network,” *arXiv preprint arXiv:1703.09452*, 2017.
- [4] S. R. Park and J. Lee, “A fully convolutional neural network for speech enhancement,” *arXiv preprint arXiv:1609.07132*, 2016.
- [5] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.
- [6] L. Griffiths and C. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Transactions on antennas and propagation*, vol. 30, no. 1, pp. 27–34, 1982.
- [7] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, “Investigating rnn-based speech enhancement methods for noise-robust text-to-speech,” in *SSW*, 2016, pp. 146–152.
- [8] J.-M. Valin, “A hybrid dsp/deep learning approach to real-time full-band speech enhancement,” in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2018, pp. 1–5.
- [9] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, “Single-channel multi-speaker separation using deep clustering,” *arXiv preprint arXiv:1607.02173*, 2016.
- [10] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [11] Y. Luo and N. Mesgarani, “Tasnet: time-domain audio separation network for real-time, single-channel speech separation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.
- [12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [13] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation,” *arXiv preprint arXiv:1910.06379*, 2019.
- [14] C. K. A. Reddy, E. Beyrami, H. Dubey, V. Gopal, R. Cheng, R. Cutler, S. Matuskevych, R. Aichner, A. Aazami, S. Braun, P. Rana, S. Srinivasan, and J. Gehrke, “The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework,” 2020.
- [15] Y. Xia, S. Braun, C. K. A. Reddy, H. Dubey, R. Cutler, and I. Tashev, “Weighted speech distortion losses for neural-network-based real-time speech enhancement,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 871–875.
- [16] D. S. Williamson, Y. Wang, and D. Wang, “Complex ratio masking for monaural speech separation,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 24, no. 3, pp. 483–492, 2015.
- [17] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, “End-to-end speech separation with unfolded iterative phase reconstruction,” *arXiv preprint arXiv:1804.10204*, 2018.
- [18] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [19] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, “Whamr!: Noisy and reverberant single-channel speech separation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 696–700.
- [20] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, “Universal sound separation,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 175–179.
- [21] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [22] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation,” *arXiv preprint arXiv:1809.07454*, 2018.
- [23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [24] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.
- [25] J. Thiemann, N. Ito, and E. Vincent, “The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings,” *The Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3591–3591, 2013.
- [26] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, “A pitch tracking corpus with evaluation on multipitch tracking scenario,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [27] “Itu-t p. 808: Subjective evaluation of speech quality with a crowdsourcing approach,” 2018.
- [28] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 2016, pp. 31–35. [Online]. Available: <https://www.merl.com/publications/TR2016-003>
- [29] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.
- [30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [31] “Itu-t p. 862: Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.” 2001.
- [32] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “Sdr-half-baked or well done?” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.
- [33] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “A short-time objective intelligibility measure for time-frequency weighted noisy speech,” in *2010 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2010, pp. 4214–4217.