DIFFERENTIABLE CONSISTENCY CONSTRAINTS FOR IMPROVED DEEP SPEECH ENHANCEMENT

Scott Wisdom, John R. Hershey, Kevin Wilson, Jeremy Thorpe, Michael Chinen, Brian Patton, Rif A. Saurous

Google Research

ABSTRACT

In recent years, deep networks have led to dramatic improvements in speech enhancement by framing it as a data-driven pattern recognition problem. In many modern enhancement systems, large amounts of data are used to train a deep network to estimate masks for complex-valued short-time Fourier transforms (STFTs) to suppress noise and preserve speech. However, current masking approaches often neglect two important constraints: STFT consistency and mixture consistency. Without STFT consistency, the system's output is not necessarily the STFT of a time-domain signal, and without mixture consistency, the sum of the estimated sources does not necessarily equal the input mixture. Furthermore, the only previous approaches that apply mixture consistency use real-valued masks; mixture consistency has been ignored for complex-valued masks.

In this paper, we show that STFT consistency and mixture consistency can be jointly imposed by adding simple differentiable projection layers to the enhancement network. These layers are compatible with real or complex-valued masks. Using both of these constraints with complex-valued masks provides a 0.7 dB increase in scale-invariant signal-to-distortion ratio (SI-SDR) on a large dataset of speech corrupted by a wide variety of nonstationary noise across a range of input SNRs.

Index Terms— Speech enhancement, STFT consistency, mixture consistency, deep learning

1. INTRODUCTION

In recent years, deep neural networks (DNNs) have led to dramatic improvements in speech enhancement. Typically deep networks are trained to estimate masks for complex-valued short-time Fourier transforms (STFTs) to suppress noise and preserve speech. That is, given N time-domain samples of an input mixture of J sources:

$$\mathbf{y} = \sum_{j=1}^{J} \mathbf{x}_j, \tag{1}$$

a DNN is trained to produce a mask $\mathbf{M}_j$ for each source given the mixture. For each of the J sources, the mask is applied to the mixture STFT $\mathbf{Y} = \mathcal{S}\{\mathbf{y}\}$ of shape $F \times T$, and the separated audio signal is recovered by applying the inverse STFT operator:

$$\hat{\mathbf{x}}_j = \mathcal{S}^{-1}\big\{\mathbf{M}_j \odot \mathbf{Y}\big\}, \tag{2}$$

where $\mathcal{S}$ and $\mathcal{S}^{-1}$ are forward and inverse STFT operators.

Such masking-based DNN approaches have been very successful [1, 2, 3, 4]. However, existing approaches have two deficiencies. First, the loss function used to train the enhancement network is typically measured on the masked noisy STFT. The problem with this is that applying an arbitrary mask does not produce a consistent STFT, in the sense that the masked STFT could not be computed from any real-valued time-domain signal. Therefore, if the masked STFT is inverted then recomputed, the resulting magnitudes and phases will be different. Second, some approaches that use a real-valued mask and all approaches that use a complex-valued mask do not leverage the basic assumption given by the model (1): that the separated sources should add up to the original mixture signal.

In this paper, we show that STFT and mixture consistency can be enforced by adding simple end-to-end layers in the enhancement network. These two techniques can be combined with any masking-based speech enhancement model to improve performance. Also, these constraints are compatible with complex-valued masks, which for the first time allows systems that use complex-valued masks to take advantage of the basic assumption of mixture consistency.

2. RELATION TO PRIOR WORK

STFT consistency has been exploited by a number of works [5, 6, 7, 8, 9], but usually in a model-based context that requires iterative algorithms. In contrast to our work, none of these approaches have combined STFT consistency with DNNs. Since we use a single differentiable consistency constraint layer within a DNN, we do not require iterations to enforce consistency. A few recent works [10, 11, 12, 13] implicitly take STFT consistency into account by using time-domain loss functions or by performing enhancement directly in the time domain.

Gunawan and Sen [14] proposed adding mixture consistency constraints to improve Griffin-Lim, resulting in the multiple input spectrogram inversion (MISI) algorithm. Wang et al. [10] trained a real-valued masking-based system for speech separation through unfolded MISI iterations. Strumel and Daudet [15] proposed iterative partitioned phase reconstruction (PPR), which also relied on mixture consistency. In contrast to these works, we do not require multiple iterations for phase estimation, and furthermore we experiment with estimating phase using a complex-valued mask.

Lee et al. [16] recently used a soft squared error penalty between the warped magnitudes of the true and estimated mixture STFTs to promote mixture consistency. In contrast, we use an exact projection step implemented as a neural network layer.

3. CONSISTENCY

3.1. STFT consistency

When a STFT uses overlapping frames, applying a mask to the STFT of a mixture signal, whether the mask is real or complex-valued, does not necessarily produce a consistent STFT [5]. A consistent STFT

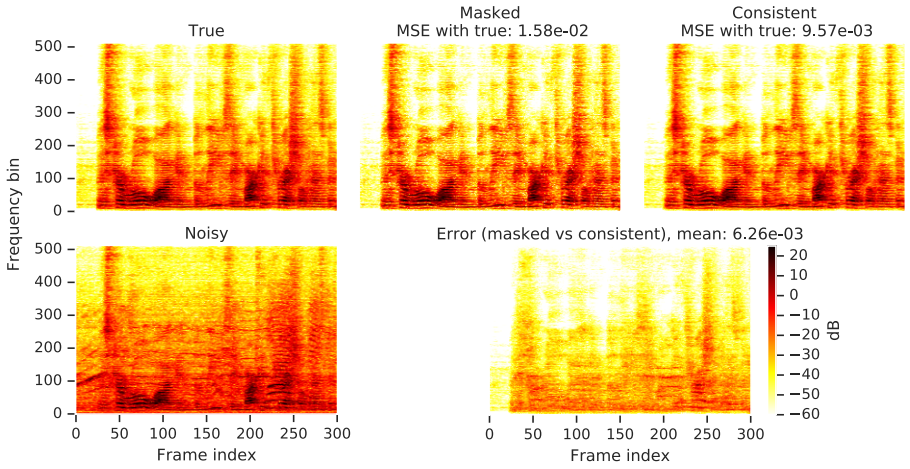

Fig. 1. Illustration of the effect of STFT consistency. A noisy STFT (spectrogram in lower left) is masked using an ideal phase-sensitive mask (4) to recover a target speech signal (spectrogram in upper left). Then, this masked STFT is made consistent using the projection (3). Notice that the masked spectrogram (upper middle panel) is substantially different from the STFT-consistent spectrogram (upper right panel), with the magnitude-squared error visualized in the lower right panel.

$\mathbf{X}$ is one such that there exists a real-valued time-domain signal $\mathbf{x}$ satisfying $\mathbf{X} = \mathcal{S}\{\mathbf{x}\}$. A STFT $\mathbf{X}$ is consistent if it satisfies

$$\mathbf{X} = \mathcal{S}\Big\{\mathcal{S}^{-1}\{\mathbf{X}\}\Big\} = \mathcal{P}_S\{\mathbf{X}\}, \tag{3}$$

where $\mathcal{P}_S$ refers to the projection performed by the sequence of inverse and forward STFT operations.

An important reason to enforce consistency on separated estimates $\mathbf{M}_j \odot \mathbf{Y}$ is that these estimates are used in loss functions and metrics used during deep network training. If the masked STFT is simply used in a loss function, e.g. mean-squared error between magnitudes of the masked and ground-truth spectrograms, the magnitudes $|\hat{\mathbf{X}}_j| = |\mathbf{M}_j \odot \mathbf{Y}|$ do not necessarily correspond to the STFT magnitudes of the reconstructed time-domain signal, $|\mathcal{S}\{\hat{\mathbf{x}}_j\}|$. Thus, the loss function will not be accurately measuring the spectral magnitude of the estimate.

3.1.1. Illustration of STFT consistency

Figure 1 illustrates the effect of STFT consistency. A clean speech example $\mathbf{s}$ (spectrogram in upper left panel) is embedded in non-stationary noise $\mathbf{v}$ at 8 dB SNR (mixture spectrogram in lower left panel). Then, we compute an oracle phase-sensitive mask [2] as

$$\frac{|S_{f,t}|}{|S_{f,t} + V_{f,t}|} \cdot \cos(\angle S_{f,t} - \angle Y_{f,t}). \tag{4}$$

This mask is applied to the noisy STFT to create a masked STFT (spectrogram shown in upper middle panel). Then, we use the projection (3) to create a consistent STFT of the enhanced speech (spectrogram in upper right panel). The lower right panel visualizes the magnitude-squared error between the masked and consistent STFTs.

Note that the masked STFT (upper middle panel) differs substantially from the consistent STFT (upper right panel). This indicates the importance of STFT consistency: if a neural network training loss is measured on the masked magnitude instead of the consistent magnitude, then the loss is not looking at the actual spectrogram of the enhanced time-domain signal. In particular, the magnitude-squared error between consistent and true STFTs (9.57e-3) is less than the magnitude-squared error between masked and true STFTs (1.58e-2). This discrepancy emphasizes the importance of having the network compute training losses on consistent STFTs, since otherwise the network is focusing on irrelevant parts of the error.

3.1.2. Backpropagating through STFTs

A natural way to enforce STFT consistency is to simply project estimated signals using the constraint (3). Since the forward and inverse STFT operations are linear transforms implemented in TensorFlow[1], this projection can simply be treated as an extra layer in the network.

3.2. Mixture consistency

An obvious constraint on separated signals comes from the original mixing model (1), as it is natural to assume that estimates of the separated signals should add up to the original mixture. This is equivalent to the complex STFTs of the estimated sources adding up to the complex mixture STFT:

$$\sum_j \underline{X}_{j,f,t} = Y_{f,t} \quad \forall t, f, \tag{5}$$

where $\underline{X}_{j,f,t}$ is the mixture-consistent TF bin for source j.

In the past, this mixture consistency constraint has been enforced by using masks constrained to sum to one across the sources, which is equivalent to using the projection (7) alone. However, when mixture consistency is combined with STFT consistency, constraining masks to sum to one across sources is no longer sufficient for enforcing mixture consistency, since the phases for different sources will change within the same TF bin after applying the STFT consistency projection (3).

In this section, we describe a simple differentiable mixture projection layer that is compatible with STFT consistency and can enforce mixture consistency for any type of masking-based enhancement or separation method, including real-valued and complex-valued masks and explicit phase prediction.

3.2.1. Backpropagating through mixture-consistent projection

To ensure mixture consistency without putting explicit constraints on the masks, we project the masked estimates to the nearest points on the subspace of mixture-consistent estimates. To do this, we solve

[1] https://www.tensorflow.org/api_docs/python/tf/contrib/signal/inverse_stft

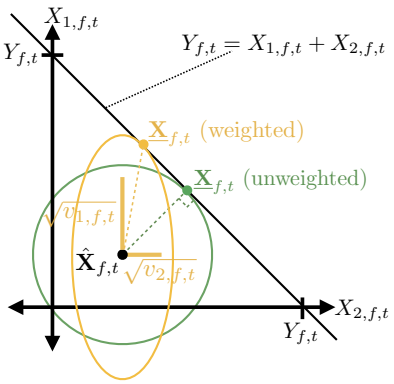

Fig. 2. Geometric illustration of unweighted (green) and weighted (yellow) mixture consistency for two real-valued sources in a single time-frequency bin.

the following optimization problem for each TF bin:

$$\begin{aligned} \underset{\underline{\mathbf{X}}_{f,t} \in \mathbb{C}^J}{\text{minimize}} \quad & \frac{1}{2} \sum_j |\underline{X}_{j,f,t} - \hat{X}_{j,f,t}|^2 \\ \text{subject to} \quad & \sum_j \underline{X}_{j,f,t} = Y_{f,t}. \end{aligned} \tag{6}$$

Using the method of Lagrange multipliers and defining the estimated mixture $\hat{Y}_{f,t} = \sum_j \hat{X}_{j,f,t}$ yields the following update, which is a simple projection that can be added as a layer in the network and backpropagated through:

$$\underline{X}_{j,f,t} = \hat{X}_{j,f,t} + \frac{1}{J}(Y_{f,t} - \hat{Y}_{f,t}). \tag{7}$$

3.2.2. Weighted mixture-consistent projection with uncertainty

If we have *a priori* knowledge of the uncertainty of the source estimates, a weighted version of the problem (6) can be used to project the sources. Assume that in each frequency bin, each estimated source can be modeled as a zero-mean complex-valued circular Gaussian with variance $v_{j,f,t}$.

Given variances $v_{j,f,t}$, the weighted problem is

$$\begin{aligned} \underset{\underline{\mathbf{X}}_{f,t} \in \mathbb{C}^J}{\text{minimize}} \quad & \frac{1}{2} \sum_j \frac{1}{v_{j,f,t}} \left|\underline{X}_{j,f,t} - \hat{X}_{j,f,t}\right|^2 \\ \text{subject to} \quad & \sum_j \underline{X}_{j,f,t} = Y_{f,t}. \end{aligned} \tag{8}$$

Using the method of Lagrange multipliers yields

$$\underline{X}_{j,f,t} = \hat{X}_{j,f,t} + \frac{v_{j,f,t}}{\sum_{j'} v_{j',f,t}}(Y_{f,t} - \hat{Y}_{f,t}). \tag{9}$$

Notice that if all sources have equal uncertainty, i.e. that $v_{j,f,t} = v_{j',f,t}$ for all $j \neq j'$, then the weighted projection (9) is equivalent to the unweighted projection (7).

Figure 2 illustrates unweighted and weighted mixture consistency for two sources. For visualization, we depict the sources' TF bins $X_{1,f,t}$ and $X_{2,f,t}$ as real-valued instead of complex-valued. Note that the projected mixture-consistent estimate $\underline{\mathbf{X}}_{f,t}$ is given by the intersection of the mixture constraint line and the ellipse representing contours of a diagonal-covariance Gaussian pdf. When the variances $v_{j,f,t}$ are the same for all sources, the ellipse becomes a circle, and the mixture-consistent projection is orthogonal to the constraint surface.

3.3. Order of consistency operations

The STFT and mixture consistency operations (3) and (7) are linear projections. Though linear projections do not in general commute, the order in which these projections are applied does not matter. As a proof, we start from the case where STFT consistency (3) is applied after mixture consistency (7). Notice that since $\mathcal{P}_S$ is linear, and since a linear combination of consistent STFTs is still a consistent STFT, we can write this as being equivalent to applying mixture consistency after STFT consistency:

$$\begin{aligned} \underline{\mathbf{X}}_j &= \mathcal{P}_S\{\hat{\mathbf{X}}_j + \frac{1}{J}(\mathbf{Y} - \sum_{j'} \hat{\mathbf{X}}_{j'})\} \\ \Leftrightarrow \underline{\mathbf{X}}_j &= \mathcal{P}_S\{\hat{\mathbf{X}}_j\} + \frac{1}{J}\mathbf{Y} - \frac{1}{J}\sum_{j'} \mathcal{P}_S\{\hat{\mathbf{X}}_{j'}\}. \end{aligned} \tag{10}$$

However, weighted mixture projections that apply a different weight per TF bin are not orthogonal to STFT consistency projection, since the relation (10) relies on $1/J$ not depending on time or frequency. Despite this lack of commutativity, we do still observe benefits of combining STFT consistency with weighted mixture consistency. Jointly imposing STFT and weighted mixture consistency is more complicated, which we defer to future work.

4. EXPERIMENTS

4.1. Dataset

We use a dataset constructed from publicly-available data. Speech is taken from a subset of LibriSpeech [17], where train, validation, and test sets use nonoverlapping sets of speakers, and audio has been filtered using WADA-SNR [18] to ensure low levels of background noise. Noise data is sourced from `freesound.org`, which contains a large variety of music, musical instruments, field recordings, and sound effects. Files that are very long, have a large number of zero samples, or exhibit clipping are filtered out. The sampling rate is 16kHz.

Speech and noise are mixed with normally-distributed SNRs, with a mean of 5 dB and standard deviation of 10 dB. We also apply a random gain to the mixture signals, with a mean of -10 dB and a standard deviation of 5 dB. These random gain parameters were chosen such that the clipping is minimal in the audio. After mixing, the duration of the training set is 134.2 hours, and the validation and test sets are 6.2 hours each.

4.2. Model architecture and training

Our model, shown in Figure 3, uses a modified version of a similar architecture that recently achieved state-of-the-art performance on the CHiME2 dataset [4]. The model consists of a convolutional front-end operating on spectral input features, a single unidirectional LSTM with residual connections and width of 400, and two fully-connected layers with 600 hidden units each. The input features are power-compressed STFTs, computed as $X^{0.3} := |X|^{0.3} e^{j \angle X}$. STFTs are computed using 50 ms Hann windows with a 10 ms hop and FFT length of 1024.

The training loss, L, for all networks is as follows:

$$\begin{aligned} L = \sum_{j=1}^{2} z_j \sum_{f,t} \Big[& \left(|X_{j,f,t}|^{0.3} - |\hat{X}_{j,f,t}|^{0.3}\right)^2 \\ & + 0.2 \cdot \left|X^{0.3}_{j,f,t} - \hat{X}^{0.3}_{j,f,t}\right|^2 \Big], \end{aligned} \tag{11}$$

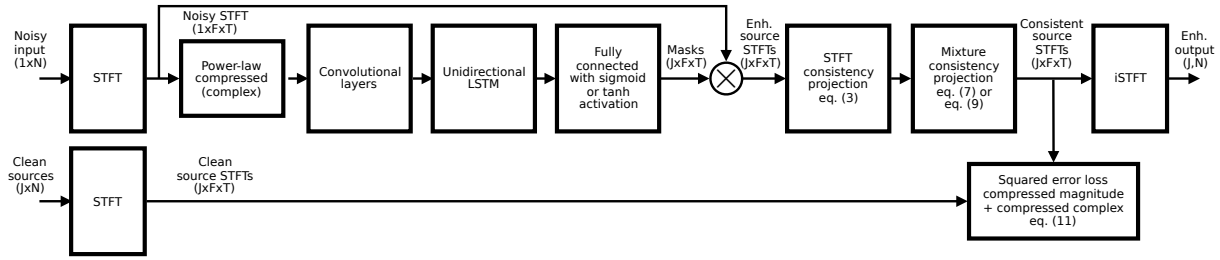

Fig. 3. System architecture when both STFT consistency and mixture consistency are used.

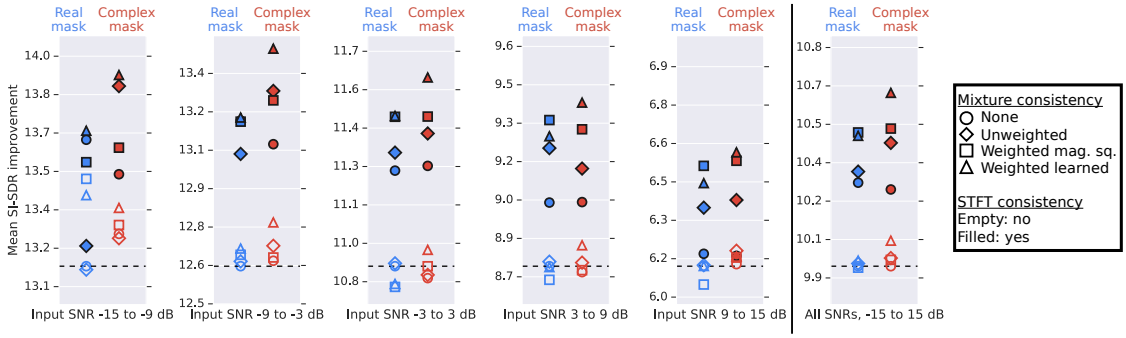

Fig. 4. Mean improvement in SI-SDR for different input SNRs for the the test set. Notice that using both types of consistency almost always improves performance, for both real-valued and complex-valued masks. The dashed lines indicate the performance of the baseline model which uses a real-valued mask with noisy phase and neither consistency constraint (blue empty circle).

where $z_1 = 0.8$ is the speech loss weight and $z_2 = 0.2$ is the noise loss weight. We train and validate on fixed-length, three-second clips. The Adam optimizer [19] is used with a batch size of 8, learning rate of 3×10^{-5}, and default parameters otherwise.

Our proposed consistency constraints are compatible with both real and complex-valued masks. For real-valued masking, the network predicts a single scalar value through a sigmoid nonlinearity for each TF bin, and the noisy phase is used for reconstruction. To perform complex-valued masking, we use an approach similar to that of Williamson et al. [3]. For each TF bin, the network predicts the real and imaginary parts of a complex-valued mask through a hyperbolic tangent (tanh) nonlinearity. This mask is multiplied with the complex noisy STFT, then reconstructed.

To implement weighted mixture consistency (9), we consider two types of weighting schemes:

1. Weights $v_{j,f,t}$ are the squared estimated source magnitudes, $|\hat{X}_{j,f,t}|^2$. This has the advantage of not adding any correction signal to TF bins that have a low magnitude, which helps when there is only one signal active within a TF bin.
2. Weights are learned by the network. The network outputs a scalar for each time, frequency, and source. A sigmoid nonlinearity is applied, producing weights $w_{1,f,t}$ for the speech source, and $w_{2,f,t} = 1 - w_{1,f,t}$ for the noise source. The terms $v_{j,f,t}/(\sum_{j'} v_{j',f,t})$ in (9) are replaced by $w_{j,f,t}$.

4.3. Results

Results are shown in Figure 4 in terms of scale-invariant SDR (SI-SDR) improvement. SI-SDR measures signal fidelity with respect to a reference signal while allowing for a gain mismatch [20, 21]:

$$\text{SI-SDR} = 10 \log_{10} \frac{\|\alpha \mathbf{x}\|^2}{\|\alpha \mathbf{x} - \hat{\mathbf{x}}\|^2}, \qquad (12)$$

where $\alpha = \text{argmin}_a \|a\mathbf{x} - \hat{\mathbf{x}}\|^2 = \mathbf{x}^T \hat{\mathbf{x}} / \|\mathbf{x}\|^2$.

These results are grouped into five bins based on input SNR, from -15 dB to 15 dB with bin width of 6 dB. Models that use both STFT and mixture consistency constraints almost always outperform models that do not use these constraints. The best models tend to use weighted mixture consistency with learned weights. Phase prediction provides a slight improvement in performance, especially when using a complex mask for phase prediction at lower input SNRs. The system with the best overall mean SI-SDR improvement of 10.6 dB uses complex masking, STFT consistency, and weighted mixture consistency with learned weights. This improves 0.7 dB over the baseline system, which has mean SI-SDR improvement of 9.9 dB.

5. CONCLUSION

In this paper, we have shown that the simple addition of differentiable neural network layers can be used to enforce STFT and mixture consistency on source estimates of an audio source separation network. Adding these consistency constraint layers improves speech enhancement performance on a dataset using a wide variety of nonstationary noise and SNR levels. These constraints are also compatible any type of STFT-based enhancement systems, including those that use complex-valued masks. In future work, we will combine these constraints with other types of phase estimation and generative models, as well as use them for general audio source separation, rather than just speech enhancement. Another interesting direction of future work is defining a weighted version of STFT consistency and joint weighted STFT and mixture consistency, both of which have the potential to further improve performance.

6. REFERENCES

[1] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. ICASSP*, May 2013.

[2] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, Apr. 2015.

[3] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for joint enhancement of magnitude and phase," in *Proc. ICASSP*, Mar. 2016.

[4] K. Wilson, M. Chinen, J. Thorpe, B. Patton, J. Hershey, R. A. Saurous, J. Skoglund, and R. F. Lyon, "Exploring tradeoffs in models for low-latency speech enhancement," in *Proc. IWAENC*, Sep. 2018.

[5] J. Le Roux, N. Ono, and S. Sagayama, "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction," in *Proc. SAPA*, Sep. 2008.

[6] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. 13th International Conference on Digital Audio Effects (DAFx-10)*, Sep. 2010.

[7] J. Le Roux, E. Vincent, Y. Mizuno, H. Kameoka, N. Ono, and S. Sagayama, "Consistent Wiener filtering: Generalized time-frequency masking respecting spectrogram consistency," in *Proc. LVA/ICA*, Sep. 2010.

[8] P. Magron, J. Le Roux, and T. Virtanen, "Consistent anisotropic wiener filtering for audio source separation," in *Proc. WASPAA*, Oct. 2017.

[9] J. Le Roux and E. Vincent, "Consistent wiener filtering for audio source separation," *IEEE Signal Processing Letters*, vol. 20, no. 3, 2013.

[10] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," *Proc. Interspeech*, Sep. 2018.

[11] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *arXiv:1810.01395*, 2018.

[12] A. Pandey and D. Wang, "A new framework for supervised speech enhancement in the time domain," in *Proc. Interspeech*, Sep. 2018.

[13] Y. Luo and N. Mesgarani, "Tasnet: Surpassing ideal time-frequency masking for speech separation," *arXiv:1809.07454*, 2018.

[14] D. Gunawan and D. Sen, "Iterative phase estimation for the synthesis of separated sources from single-channel mixtures," *IEEE Signal Processing Letters*, vol. 17, no. 5, 2010.

[15] N. Sturmel and L. Daudet, "Iterative phase reconstruction of wiener filtered signals," in *Proc. ICASSP*, Mar. 2012.

[16] J. Lee, J. Skoglund, T. Shabestary, and H.-G. Kang, "Phase-sensitive joint learning algorithms for deep learning-based speech enhancement," *IEEE Signal Processing Letters*, vol. 25, no. 8, 2018.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, Apr. 2015.

[18] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Proc. Interspeech*, Sep. 2008.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *Proc. Interspeech*, Sep. 2016.

[21] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" *Proc. ICASSP*, May 2019.