MONAURAL SINGING VOICE SEPARATION WITH SKIP-FILTERING CONNECTIONS AND RECURRENT INFERENCE OF TIME-FREQUENCY MASK

Stylianos Ioannis Mimilakis[*], *Konstantinos Drossos*[†], *João F. Santos*[§‡],
Gerald Schuller[*], *Tuomas Virtanen*[†], *Yoshua Bengio*[§¶]

[*]Fraunhofer IDMT, Ilmenau, Germany
[†]Tampere University of Technology, Tampere, Finland
[§]Université de Montréal, Montreal, Canada
[‡]INRS-EMT, Montreal, Canada
[¶]CIFAR Fellow

ABSTRACT

Singing voice separation based on deep learning relies on the usage of time-frequency masking. In many cases the masking process is not a learnable function or is not encapsulated into the deep learning optimization. Consequently, most of the existing methods rely on a post processing step using the generalized Wiener filtering. This work proposes a method that learns and optimizes (during training) a source-dependent mask and does not need the aforementioned post processing step. We introduce a recurrent inference algorithm, a sparse transformation step to improve the mask generation process, and a learned denoising filter. Obtained results show an increase of 0.49 dB for the signal to distortion ratio and 0.30 dB for the signal to interference ratio, compared to previous state-of-the-art approaches for monaural singing voice separation.

***Index Terms*—** Singing voice separation, recurrent encoder decoder, recurrent inference, skip-filtering connections

1. INTRODUCTION

The problem of music source separation has received a lot attention in the fields of audio signal processing and deep learning [1]. The most adopted solution is the estimation of a time-varying and source-dependent filter, which is applied to the mixture [2]. Performing the filtering operation is done by treating audio signals as wide-sense stationary. This involves transforming the mixture signal using the short-time Fourier transform (STFT). Then, the source-dependent filtering operation is applied to the complex-valued coefficients of the mixture signal. More formally, let $\mathbf{x}$ be the time-domain mixture signal vector of J sources. $\mathbf{Y} \in \mathbb{C}^{M \times N}$ is the complex-valued STFT representation of $\mathbf{x}$, comprising of M overlapping time frames and N frequency sub-bands. The estimation of the j-th target source ($\hat{\mathbf{Y}}^j \in \mathbb{C}^{M \times N}$) is achieved through:

$$\hat{\mathbf{Y}}^j = \mathbf{Y} \odot \mathbf{M}^j, \tag{1}$$

where $\odot$ is element-wise product and $\mathbf{M}^j \in \mathbb{R}_{\geq 0}^{M \times N}$ is the j-th source-dependent filter, henceforth denoted as mask. In [2] was shown that a preferred way for estimating the j-th source is to derive the mask through the generalized Wiener filtering using α-power magnitude spectrograms as:

$$\mathbf{M}^j = \frac{|\hat{\mathbf{Y}}^j|^{\circ \alpha}}{\sum\limits_j |\hat{\mathbf{Y}}^j|^{\circ \alpha}}, \tag{2}$$

where, $|\cdot|$ and $\circ$ denote the entry-wise absolute and exponentiation operators respectively, and α is an exponent chosen based on the assumed distributions that the sources follow. Finding α (and thus an optimal $\mathbf{M}^j$ for the source estimation process [2]) is an open optimization problem [2, 3].

Deep learning methods for music source separation are trained using synthetically created mixtures $\mathbf{Y}$ (adding signals $\mathbf{Y}^j$ together, i.e., knowing the target decomposition). They can be divided into two categories. In the first category, the methods try to predict the mask directly from the mixture magnitude spectrum [4] (i.e. $f_1 : |\mathbf{Y}| \rightarrow \mathbf{M}^j$). This requires that an optimal $\mathbf{M}^j$ is given (e.g. all the non-linear mixing parameters of the target source are known) during training as a target. However, such information for the $\mathbf{M}^j$ is unknown, and an approximation of $\mathbf{M}^j$ is computed from the training data using Eq. (2) and empirically chosen α values, under the hypothesis that the source magnitude spectra are additive, which is not true for realistic audio signals [2, 3]. This implies that such models are optimized to predict non-optimal masks. The methods in the second category try to estimate all sources from the mixture(i.e. $f_2 : |\mathbf{Y}| \rightarrow |\hat{\mathbf{Y}}^j|^{\circ \alpha} \; \forall j \in J$) and then use these estimates to compute a mask [5, 6, 7, 8, 9]. This approach is widely adopted, since it is straightforward by employing denoising autoencoders [10], with noise corresponding to the addition of other sources. However, the

The research leading to these results has received funding from the European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet. Part of the computations leading to these results were performed on a TITAN-X GPU donated to K. Drossos from NVIDIA. The authors would like to thank Paul Magron for the precious feedback.

masks are dependent on the initial α-power magnitude estimates of the sources ($|\hat{\mathbf{Y}}^j|^{\circ\alpha}$), and the mask computation is not a learned function. Instead, the mask computation uses a deterministic function which takes as inputs the outcomes ($|\hat{\mathbf{Y}}^j|^{\circ\alpha}\forall j \in J$) of deep neural networks, e.g. as in [8].

An exception to the above are the works presented in [11] and [12], where these methods jointly learned and optimized the masking processes described by Eq. (1) and (2). In [11], highway networks [13] were shown to be able to approximate a masking process for monaural solo source separation and in [12], a more robust alternative to [11] is presented. The approach in [12] uses a recurrent encoder-decoder with skip-filtering connections, which allow a source-dependent mask generation process, applicable to monaural singing voice separation. However, the generated masks are not robust against interferences from other music sources, thus requiring a post-processing step using the generalized Wiener filtering [12].

In this work we present a method for source separation that learns to generate a source-dependent mask which does not require the generalized Wiener filtering as a post-processing step. To do so, we introduce a novel recurrent inference algorithm inspired by [14] and a sparsifying transform [15] for generating the mask $\mathbf{M}^j$. The recurrent inference allows the proposed method to have a stochastic depth of RNNs during the mask generation process, computing hidden, latent representations which are presumably better for generating the mask. The sparsifying transform is used to approximate the mask using the output of the recurrent inference. In this method the mask prediction is not based on the above mentioned assumptions about the additivity of the magnitude spectrogram of the sources, is part of an optimization process, and is not based on a deterministic function. Additionally, the method incorporates RNNs instead of feed-forward or convolutional layers for the mask prediction. This allows the method to exploit the memory of the RNNs (compared to CNNs) and their efficiency for modeling longer time dependencies of the input data. The rest of the paper is organized as follows: Section 2 presents the proposed method, followed by Section 3 which provides information about the followed experimental procedure. Section 4 presents the obtained results from the experimental procedure and Section 5 concludes this work.

2. PROPOSED METHOD

Our proposed method takes as an input the time domain samples of the mixture, and outputs time domain samples of the targeted source. The model consists of four parts. The first part implements the analysis and pre-processing of the input. The second part generates and applies a mask, thus creating the first estimate of the magnitude spectrogram of the targeted source. The third part enhances this estimate by learning and applying a denoising filter, and the fourth part constructs the time domain samples of the target source. We call the second part the "Masker" and the third the "Denoiser". We differentiate between the Masker and the Denoiser because the Masker is optimized to predict a time-frequency mask, whereas the Denoiser enhances the result obtained by time-frequency masking. We implement the Masker using a single layer bi-directional RNN encoder (RNN$_{\text{enc}}$), a single layer RNN decoder (RNN$_{\text{dec}}$), a feed-forward layer (FFN), and skip-filtering connections between the magnitude spectrogram of the mixture and the output of the FFN. We implement the Denoiser using one FFN encoder (FFN$_{\text{enc}}$), one FFN decoder (FFN$_{\text{dec}}$), and skip-filtering connections between the input to the Denoiser and the output of the FFN$_{\text{dec}}$. We jointly train the Masker and the Denoiser using two criteria based on the generalized Kullback-Leibler divergence (D_{KL}), as it is shown in [3, 16] to be a robust criterion for matching magnitude spectrograms. All RNNs are gated recurrent units (GRU). The proposed method is illustrated in Figure 1.

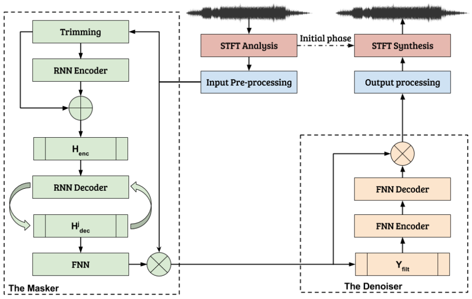

Fig. 1. Illustration of our proposed method.

2.1. Input preprocessing

Let $\mathbf{x}$ be the vector containing the time-domain samples of a monaural mixture from J sources, sampled at 44.1kHz. We compute the STFT of $\mathbf{x}$ from time frames of $N = 2049$ samples, segmented with Hamming window and a hop size of 384 samples. Each time frame is zero-padded to $N' = 4096$. Subsequent to the STFT we retain only the positive frequencies, corresponding to the first $N = 2049$ frequency sub-bands. This yields the complex-valued time-frequency representation of $\mathbf{x}$, $\mathbf{Y} \in \mathbb{C}^{M\times N}$, and the corresponding magnitude $|\mathbf{Y}| \in \mathbb{R}_{\geq 0}^{M\times N}$. We split $|\mathbf{Y}|$ in $B = \lceil M/T \rceil$ subsequences, with T being the length of the subsequence, and $\lceil\cdot\rceil$ is the ceiling function. Each subsequence b overlaps with the preceding one by an empirical factor of $L \times 2$, in order to use some context information for the encoding stage. We use each subsequence b in $|\mathbf{Y}|$, denoted as $|\mathbf{Y}_{\text{in}}|$ as an input to the skip-filtering connections (presented later). Furthermore we produce a low-bandwidth version of $|\mathbf{Y}|$, which is used for encoding, by preserving only the first $F = 744$ frequency sub-bands at each frame yielding $|\mathbf{Y}_{\text{tr}}| \in \mathbb{R}_{\geq 0}^{T\times F}$ $|\mathbf{Y}_{\text{tr}}|$. This operation retains information up to 8 kHz, in order to reduce the number of trainable parameters but preserving the most relevant information of the singing voice source.

2.2. The Masker

RNN encoder We use $|\mathbf{Y}_{\text{tr}}|$ as an input to the RNN$_{\text{enc}}$. The forward GRU of the RNN$_{\text{enc}}$ takes $|\mathbf{Y}_{\text{tr}}|$ as an input. The backward one takes as an input the $|\overleftarrow{\mathbf{Y}}_{\text{tr}}| = [|\mathbf{y}_{\text{tr}_T}|, \ldots, |\mathbf{y}_{\text{tr}_t}|, \ldots,$

$|\mathbf{y}_{\text{tr}_1}|]$, where $|\mathbf{y}_{\text{tr}_t}| \in \mathbb{R}^{F}_{\geq 0}$ is a vector in $|\mathbf{Y}_{\text{tr}}|$ at time frame t, and ← indicates the direction of the sequence. The outputs from the Bi-GRU $\mathbf{h}_t$ and $\overleftarrow{\mathbf{h}}_t$ are updated at each time frame t using residual connections [17] and then concatenated as

$$\mathbf{h}_{\text{enc}_t} = \big[(\mathbf{h}_t + |\mathbf{y}_{\text{tr}_t}|)^{\text{T}}, (\overleftarrow{\mathbf{h}}_t + \overleftarrow{|\mathbf{y}}_{\text{tr}_t}|)^{\text{T}}\big]^{\text{T}}. \qquad (3)$$

The output of the RNN_{enc} for all $t \in T$ is denoted as $\mathbf{H}_{\text{enc}} \in \mathbb{R}^{T \times (2 \times F)}$ and is followed by the context information removal defined as:

$$\tilde{\mathbf{H}}_{\text{enc}} = [\mathbf{h}_{\text{enc}_{1+L}}, \mathbf{h}_{\text{enc}_{2+L}}, \ldots, \mathbf{h}_{\text{enc}_{T-L}}], \qquad (4)$$

yielding $\tilde{\mathbf{H}}_{\text{enc}} \in \mathbb{R}^{T' \times (2 \times F)}$ for $T' = T - (2 \times L)$. Residual connections are used to ease the RNN training [17].

Recurrent inference and mask prediction Inspired by recent optimization methods employing stochastic depth [14], we propose a recurrent inference algorithm that processes the latent variables of the RNN_{dec} which affect the mask generation. We use this algorithm in order to employ a stochastic depth for the network parts responsible for predicting the mask, increasing the performance of our method. The recurrent inference is an iterative process and consists in reevaluating the latent variables $\mathbf{H}^j_{\text{dec}}$, produced by the RNN_{dec}, until a convergence criterion is reached, thus avoiding the need to specify a fixed number of applications of the RNN_{dec}. The stopping criterion is a threshold on the mean-squared-error ($\mathcal{L}_{\text{MSE}}$) between the consecutive estimates of $\mathbf{H}^j_{\text{dec}}$, with a $\mathcal{L}_{\text{MSE}}$ threshold τ_{term}. A maximum number of iterations ($iter$) is used to avoid having infinite iterations for convergence between the above mentioned consecutive estimates. $\mathbf{H}^j_{\text{dec}}$ is used only for the singing voice, i.e. $j = 1$. Let $\mathcal{G}^j_{\text{dec}}$ be the *source-dependent* and trainable function of the RNN_{dec}. The recurrent inference is performed using Algorithm 1. $\mathbf{H}^j_{\text{dec}}$ is then given to the FFN layer with shared weights through time frames, in order to approximate the j-th source-dependent mask as:

Algorithm 1. Recurrent Inference

```
$\mathbf{S}^j_0 \leftarrow \mathcal{G}^j_{\text{dec}}(\tilde{\mathbf{H}}_{\text{enc}})$
for $i \in \{1, \ldots, iter\}$ do
    $\mathbf{H}^j_{\text{dec}} \leftarrow \mathcal{G}^j_{\text{dec}}(\mathbf{S}^j_{i-1})$
    if $\mathcal{L}_{\text{MSE}}(\mathbf{S}^j_{i-1}, \mathbf{H}^j_{\text{dec}}) < \tau_{\text{term}}$ then
        Terminate the process
    $\mathbf{S}^j_i \leftarrow \mathbf{H}^j_{\text{dec}}$
return $\mathbf{H}^j_{\text{dec}}$
```

$$\tilde{\mathbf{M}}^j = \text{ReLU}(\mathbf{H}^j_{\text{dec}} \mathbf{W}_{\text{mask}} + \mathbf{b}_{\text{mask}}), \qquad (5)$$

where ReLU is the element-wise rectified linear unit function producing a sparse [15] approximation of the target source mask $\tilde{\mathbf{M}}^j \in \mathbb{R}^{T' \times N}_{\geq 0}$. The sparsification is performed in order to improve the interference reduction of [12]. The ReLU function can produce high positive values inducing distortions to the audio signal. However, the reconstruction loss (see Eq. (9)) will alleviate that. $\mathbf{W}_{\text{mask}} \in \mathbb{R}^{(2 \times F) \times N}$ is the weight matrix of the FFN comprising a dimensionality expansion up to N, in order to recover the original dimensionality of the data. $\mathbf{b}_{\text{mask}} \in \mathbb{R}^N$ is the corresponding bias term.

Skip filtering connections and first estimate of the targeted source We obtain an estimate of the magnitude spectrum of the target source $|\hat{\mathbf{Y}}^j_{\text{filt}}| \in \mathbb{R}^{T' \times N}_{\geq 0}$ through the skip-filtering connections as:

$$|\hat{\mathbf{Y}}^j_{\text{filt}}| = |\tilde{\mathbf{Y}}_{\text{in}}| \odot \tilde{\mathbf{M}}^j, \text{ where} \qquad (6)$$

$$|\mathbf{Y}_{\text{in}}| = [|\mathbf{y}_{\text{in}_L}|, \cdots, |\mathbf{y}_{\text{in}_{T-L}}|]. \qquad (7)$$

2.3. The Denoiser

The output of the Masker is likely to contain interferences from other sources [12]. The Denoiser aims to learn a denoising filter for enhancing the magnitude spectrogram estimated by this masking procedure. This denoising filter is implemented by an encoder-decoder architecture with the FFN_{enc} and FFN_{dec} of Fig. 1. FFN_{enc} and FFN_{dec} have shared weights through time frames. The final enhanced magnitude spectrogram estimate of the target source $|\hat{\mathbf{Y}}^j|$ is computed using

$$|\hat{\mathbf{Y}}^j| = \text{ReLU}(\text{ReLU}(|\hat{\mathbf{Y}}^j_{\text{filt}}|\mathbf{W}_{\text{enc}} + \mathbf{b}_{\text{enc}})\mathbf{W}_{\text{dec}} + \mathbf{b}_{\text{dec}}) \odot |\hat{\mathbf{Y}}^j_{\text{filt}}|, \qquad (8)$$

where $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{N \times \lfloor N/2 \rfloor}$ and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{\lfloor N/2 \rfloor \times N}$ are the weight matrices of the FFN_{enc} and FFN_{dec}, with the corresponding biases $\mathbf{b}_{\text{enc}} \in \mathbb{R}^{\lfloor N/2 \rfloor}$, $\mathbf{b}_{\text{dec}} \in \mathbb{R}^N$, respectively. $\lfloor \cdot \rfloor$ denotes the floor function.

2.4. Training details and post-processing

We train our method to minimize the objective consisting of a reconstruction and a regularization part as:

$$\begin{aligned}\mathcal{L} = D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j|) + \lambda_{\text{rec}} D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j_{\text{filt}}|) \\ + \lambda_{mask}|\text{diag}\{\mathbf{W}_{\text{mask}}\}|_1 + \lambda_{\text{dec}}||\mathbf{W}_{\text{dec}}||^2_2,\end{aligned} \qquad (9)$$

where $|\mathbf{Y}^j|$ is the magnitude spectrogram of the true source, $\text{diag}\{\cdot\}$ denotes the elements on the main diagonal of a matrix, $|\cdot|_1$, and $||\cdot||^2_2$ are the ℓ_1 vector norm and the squared matrix L_2 norm respectively, and λ_{mask}, and λ_{dec} are scalars. For λ_{rec} the following condition applies:

$$\lambda_{\text{rec}} = \begin{cases} 1, & \text{if } D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j_{\text{filt}}|) \geq \tau_{\text{rec}} \\ & \text{and } D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j|) \geq \tau_{\text{min}}\ , \\ 0, & \text{otherwise} \end{cases} \qquad (10)$$

where τ_{rec} and τ_{min} are hyper-parameters penalizing the mask generation process, allowing a collaborative minimization of the overall objective. The usage of $\lambda_{\text{rec}} D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j_{\text{filt}}|)$ will ensure that $\mathbf{M}^j$ can be used to initially estimate the target source, which is then improved by employing the Denoiser and $D_{KL}(|\mathbf{Y}^j| \,||\, |\hat{\mathbf{Y}}^j|)$. The penalization of the elements in the main diagonal of $\mathbf{W}_{\text{mask}}$ will ensure that the generated mask is not something trivial (e.g. a voice activity detector), while the reconstruction losses using the D_{KL} will ensure that a source-dependent mask is generated, that minimizes the aforementioned distance. The squared matrix L_2 norm is employed to improve the generalization of the model.

By processing each subsequence using the proposed method, the estimates are concatenated together to form $|\hat{\mathbf{Y}}^j| \in \mathbb{R}_{\geq 0}^{M \times N}$. For the singing voice we retrieve the complex-valued STFT $\hat{Y}^{j=1}$ by means of 10 iterations of the Griffin-Lim algorithm (least squares error estimation from modified STFT magnitude) [18] initialized with the mixture's phase and using $|\hat{\mathbf{Y}}^j|$. The time-domain samples $\hat{\mathbf{x}}^{j=1}$ are obtained using inverse STFT.

3. EXPERIMENTAL PROCEDURE

We use the development subset of Demixing Secret Dataset (DSD100)[1] and the non-bleeding/non-instrumental stems of MedleydB [19] for the training and validation of the proposed method. The evaluation subset of DSD100 is used for testing the objective performance of our method. For each multi-track contained in the audio corpus, a monaural version of each of the four sources is generated by averaging the two available channels. For training, the true source $|\mathbf{Y}^j|$ is the outcome of the ideal ratio masking process [20], element-wise multiplied by a factor of 2. This is performed to avoid the inconsistencies in time delays and mixing gains between the mixture signal and the singing voice (apparent in MedleydB dataset). The length of the sequences is set to $T = 60$, modeling approximately 0.5 seconds, and $L = 10$. The thresholds for the minimization of Eq.(9) are $\tau_{\text{rec}} = 1.5$ and $\tau_{\text{min}} = 0.25$ and the corresponding scalars are $\lambda_{\text{mask}} = 1e^{-2}$, and $\lambda_{\text{dec}} = 1e^{-4}$. The hidden to hidden matrices of all RNNs were initialized using orthogonal initialization [21] and all other matrices using Glorot normal [22]. All parameters are jointly optimized using the Adam algorithm [23], with a learning rate of $1e^{-4}$, over batches of 16, an L_2 based gradient norm clipping equal to 0.5 and a total number of 100 epochs. All of the reported parameters were chosen experimentally with two random audio files drawn from the development subset of DSD100. The implementation is based on PyTorch[2].

We compared our method with other state-of-the-art approaches dealing with monaural singing voice separation, following the standard metrics, namely signal to noise ratio (SIR) and signal to distortion ratio (SDR) expressed in dB, and the rules proposed in the music source separation evaluation campaign [1] (e.g. using the proposed toolbox for SIR and SDR calculation). The compared methods are: i) GRA: Deep FFNs [4] for predicting both binary and soft masks [20] which are then combined to provide source estimates, ii) CHA: A convolutional encoder-decoder for magnitude source estimation, without a trainable mask approximation [6] iii) MIM-HW: Deep highway networks for music source separation [11] approximating the filtering process of Eq.(1), retrained using the development subset of DSD100, and iv) MIM-DWF, MIM-DWF$^+$: The two GRU encoder-decoder models combined with generalized Wiener filtering [12], trained on the development subset of DSD100 (MIM-GRUDWF) and the additional stems of MedleydB (MIM-DWF$^+$). The methods denoted as MIM-HW, MIM-DWF, and MIM-DWF$^+$ were re-implemented for the purposes of this work. For the rest of the methods we used their reported evaluation results obtained from [1]. Our proposed methods are denoted as GRU-NRI, which does not include the recurrent inference algorithm, and two methods using different hyper-parameters for the recurrent inference algorithm: GRU-RIS$^{\text{s}}$, parametrized using a maximum number of iterations $iter = 3$, and $\tau_{\text{term}} = 1e^{-2}$, and GRU-RIS$^{\text{l}}$ parametrized using a maximum number of iterations $iter = 10$ and $\tau_{\text{term}} = 1e^{-3}$, which where selected according to their performance in minimizing Eq. (9).

Table 1. Median SDR and SIR values in dB for the investigated approaches. Proposed approaches are underlined. Higher is better.

Method	**SDR**	**SIR**	**Method**	**SDR**	**SIR**
GRA[4]	-1.75	1.28	MIM-DWF$^+$[12]	3.71	8.01
MIM-HW[11]	1.49	7.73	GRU-NRI	3.62	7.06
CHA[6]	1.59	5.20	GRU-RIS$^{\text{s}}$	3.41	**8.32**
MIM-DWF[12]	3.66	8.02	GRU-RIS$^{\text{l}}$	**4.20**	7.94

4. RESULTS & DISCUSSION

Table 1 summarizes the results of the objective evaluation for the aforementioned methods by showing the median values obtained from the SDR and SIR metrics. The proposed method based on recurrent inference and sparsifying transform is able to provide state-of-the-art results for monaural singing voice separation, without the necessity of post-processing steps such as generalized Wiener filtering, and/or additionally trained deep neural networks. Compared to methods that approximate the masking processes (GRA, MIM-HW, MIM-DWF, and MIM-DWF$^+$) there are significant improvements in overall median performance of both the SDR and SIR metrics, especially when the masks are not a learned function, such as in the case of CHA. Using the proposed method, a gain of 0.49 dB for the SDR is observed between MIM-DWF$^+$ and GRU-RIS$^{\text{l}}$ and 0.30 dB for the SIR between the MIM-DWF and GRU-RIS$^{\text{s}}$. Finally, by allowing a larger number of iterations during the recursive inference the mask generation performance and using skip-filtering connections we see an increase in SDR which outperforms the previous methods MIM-DWF and MIM-DWF$^+$, but at the cost of a loss in SIR. A demo for the proposed method is available at `https://js-mim.github.io/mss_pytorch/`.

5. CONCLUSION

In this work we presented an approach for singing voice separation that does not require post-processing using generalized Wiener filtering. We introduced to the skip-filtering connections [12] a sparsifying transform yielding comparable results to approaches that rely on generalized Wiener filtering. Furthermore, the introduced recurrent inference algorithm was shown to provide state-of-the-art results in monaural singing voice separation. Experimental results show that these extensions outperform previous deep learning based approaches for singing voice separation.

[1] `http://www.sisec17.audiolabs-erlangen.de`

[2] `http://pytorch.org/`

6. REFERENCES

[1] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation: 13th International Conference, LVA/ICA 2017*, 2017, pp. 323–332.

[2] A. Liutkus and R. Badeau, "Generalized Wiener filtering with fractional power spectrograms," in *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, April 2015, pp. 266–270.

[3] D. FitzGerald and R. Jaiswal, "On the use of masking filters in sound source separation," in *15th International Conference on Digital Audio Effects (DAFx-12)*, September 2012.

[4] E.-M. Grais, G. Roma, A.J.R. Simpson, and M.-D. Plumbley, "Single-channel audio source separation using deep neural network ensembles," in *Audio Engineering Society Convention 140*, May 2016.

[5] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015, pp. 2135–2139.

[6] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *Latent Variable Analysis and Signal Separation: 13th International Conference, LVA/ICA 2017*, 2017, pp. 258–266.

[7] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2017)*, Oct 2017.

[8] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, Dec 2015.

[9] A.-A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 1748–1752.

[10] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 899–907. Curran Associates, Inc., 2013.

[11] S.-I. Mimilakis, E. Cano, J. Abeßer, and G. Schuller, "New sonorities for jazz recordings: Separation and mixing using deep neural networks," in *Audio Engineering Society 2nd Workshop on Intelligent Music Production*, 2016.

[12] S.-I. Mimilakis, K. Drossos, G. Schuller, and T. Virtanen, "A recurrent encoder-decoder approach with skip-filtering connections for monaural singing voice separation," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017.

[13] R.-K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *CoRR*, vol. abs/1505.00387, 2015.

[14] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," *CoRR*, vol. abs/1603.09382, 2016.

[15] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *Journal of Machine Learning Research*, vol. 18, no. 83, pp. 1–52, 2017.

[16] A. Liutkus, D. Fitzgerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2015)*, Oct 2015, pp. 1–5.

[17] Y. Wu et al, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[18] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, Apr 1984.

[19] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th International Society for Music Information Retrieval (ISMIR)*, Oct 2014, pp. 66–70.

[20] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *40th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, April 2015, pp. 708–712.

[21] A.-M. Saxe, J.-L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *CoRR*, vol. abs/1312.6120, 2013.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, 2010, pp. 249–256.

[23] D.-P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.