UNSUPERVISED INTERPRETABLE REPRESENTATION LEARNING FOR SINGING VOICE SEPARATION

Stylianos I. Mimilakis
Semantic Music Techn. Group
Fraunhofer-IDMT
Ilmenau, Germany
`mis@idmt.fraunhofer.de`

Konstantinos Drossos
Audio Research Group
Tampere University
Tampere, Finland
`konstantinos.drossos@tuni.fi`

Gerald Schuller
Applied Media Systems Group
Technical University of Ilmenau
Ilmenau, Germany
`gerald.schuller@tu-ilmenau.de`

ABSTRACT

In this work, we present a method for learning interpretable music signal representations directly from waveform signals. Our method can be trained using unsupervised objectives and relies on the denoising auto-encoder model that uses a simple sinusoidal model as decoding functions to reconstruct the singing voice. To demonstrate the benefits of our method, we employ the obtained representations to the task of *informed* singing voice separation via binary masking, and measure the obtained separation quality by means of scale-invariant signal to distortion ratio. Our findings suggest that our method is capable of learning meaningful representations for singing voice separation, while preserving conveniences of the the short-time Fourier transform like non-negativity, smoothness, and reconstruction subject to time-frequency masking, that are desired in audio and music source separation.

Keywords Representation learning, unsupervised learning, denoising auto-encoders, singing voice separation

1 Introduction

A particular task in music signal processing that has attracted a lot of research interest is the estimation of the singing voice source from within an observed mixture signal [1]. To that aim, deep supervised learning is shown to yield remarkable results. Approaches that rely on deep supervised learning can be discriminated in two categories, the ones that operate in the short-time Fourier transform (STFT) domain [2, 3], and we denote as spectral-based approaches, and the ones that operate directly on the waveform signals [4, 5], that we denote as waveform-based approaches. Spectral and waveform based approaches have in common that they implicitly compute source-dependent masks that are applied to the mixture signal, prior to the reconstruction of the target signals [2, 3, 4, 5][1].

Although the implicit masking is shown to be a simple and robust method to learn source dependent patterns for source separation [6], one could expect that waveform based approaches would outperform the spectral ones. That is because waveform based approaches are optimized using time-domain signals that also contain the phase information, that unarguably carries important signal information [7, 8] and has been neglected by many spectral based approaches [2, 3, 9, 10]. However, experimental evidence shows that spectral based approaches outperform the waveform ones [4, 5, 2]. Since the state-of-the-art (SOTA) methods for both waveform and spectral approaches rely on deep neural networks, and in both spectral and waveform approaches a considerable engineering effort has been directed to the employed neural architecture, it is evident that the difference in the performance between the two different approaches can be attributed to the utilized signal representation. For the waveform-based ones is the time domain samples, but for the spectral-based is the non-negative signal representation offered by the magnitude of the STFT. Thus, we believe that representation learning [11] for music signals is an intriguing direction for music source separation research.

In this work, we focus on representation learning for singing voice separation in an attempt to bridge the gap between spectral and waveform based approaches. To this aim, we propose a simple method for unsupervised representation learning from waveform signals, alleviating the need of having paired training data (i.e. matched multi-track audio

---

[1]Regarding the masking strategy, we are referring to the adaptation of Conv-TasNet for music signals also presented in [4].

data). However, the method still requires isolated source's audio signals. More specifically, our method is based on the denoising auto-encoder (DAE) model [12], but for the decoding functions our model inherits a simple and real-valued sinusoidal model. The sinusoidal model consists of amplitude-modulated cosine functions that overlap in time, and whose parameters are jointly optimized with the rest of the DAE. The motivation behind using a sinusoidal model as a decoding function is to guide (via back-propagation) the encoding layers to learn and convey information regarding the energy of specific cosine functions that compose the audio signal, leading to interpretable and STFT-like representations.

Our method is inspired by the concept of differentiable digital signal processing [13] where the parameters of common digital signal processing functions are optimized by means of back-propagation, and in our case we back-propagate through the parameters of a simple signal model. Furthermore, our method is similar to the Sinc-Network presented in [14], that uses sinc functions in the encoding layers of convolutional kernels for interpretable deep learning, and its extension to complex-valued representations for speaker separation [15]. However, our method differs from [15] as the representation of the proposed method is real-valued, alleviating the cumbersome signal processing operations on complex numbers. It also differs from approaches that initialize the front-end parts of the networks with cosine functions [16] that are then updated by means of back-propagation, by inheriting the cosine functions as a part of the model to be optimized. The rest of the document is organized as follows: Section 2 presents the proposed method, Section 3 describes the followed experimental procedure, Section 4 demonstrates and discusses the obtained results, and Section 5 concludes this work.

2 Proposed Method

Our proposed method employs two functions, the encoder E, and the decoder D. The input to our method is a music signal $\mathbf{x}$, and the output is the learned representation of $\mathbf{x}$, denoted as $\mathbf{A}$. The encoder E learns the representation $\mathbf{A}$ with the help of the decoder D. D is responsible for reconstructing the signal, given the representation computed by E. The reconstructed signal can then be used to optimize E and D using a reconstruction objective. To enforce interpretability for the representation $\mathbf{A}$, we use a differentiable sinusoidal synthesis model for the decoder D. An illustration of the proposed method is given in Figure 1.

2.1 The Encoder

The encoder E gets as an input a given signal $\mathbf{x}$ and outputs its representation $\mathbf{A}$. In order for E to learn to yield the representation $\mathbf{A}$, an initial stage of training is performed, using two signals created by the singing voice signal ($\mathbf{x}_v$). The first signal is termed as $\tilde{\mathbf{x}}_m$ and is the result of a corruption process for $\mathbf{x}$ with an additive generic multi-modal distribution-based noise (e.g. a randomly sampled signal that contains accompaniment music, like a mixture of drums, guitars, synthesizers, and bass). The second signal is termed as $\tilde{\mathbf{x}}_v$ and is the result of a corruption process for $\mathbf{x}$ using an additive Gaussian noise.

Both signals are of length N samples, i.e., $\tilde{\mathbf{x}}_m, \tilde{\mathbf{x}}_v \in \mathbb{R}^N$, and are independently used as an input to E, resulting into two representations $\mathbf{A}_m, \mathbf{A}_v \in \mathbb{R}^{C \times T}$, respectively. C is the number of templates and T is the temporal length of each template (similarly as the time-frames in STFT-related representations). To compute each representation $\mathbf{A}$, E comprises of two one-dimensional strided convolutions, with appropriate zero-padding. The first operation involves a convolution of each signal with a set of C number of kernels of temporal length L and a stride S. The stride S is a hyper-parameter and affects the expected number of time-frames T by $T = \lceil N/S \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. The resulting latent signal is given to the second convolution, which is a *dilated* one-dimensional convolution [17] with C number of kernels, a smaller temporal length $L' << L$, and a stride equal to 1. The output of the second convolution is updated by means of residual connections using the output from the first convolution, followed by the rectified linear unit (ReLU) activation function [18]. The ReLU function promotes a non-negative representation and sparse representation by preserving positive values and setting the rest to zero [19], and is desired in many spectral-based approaches [2, 3, 9, 10] in music source separation and in general modelling of audio signals [20].

Another targeted (and useful) property of the representation is that of smoothness [16, 20], especially useful when real-valued cosine functions are involved in an auto-encoding or separation models [16]. That is because audio signal modelling based on cosine functions requires the phase information for reconstruction. Phase information is usually encoded as the sign (positive or negative value) of the real-valued representation that varies along the time-frames of the representation. Since the negative values are nullified by the application of the ReLU function, neighbouring time-frames, that convey similar information for music signals are expected to be non-smooth. To compensate for that, the second convolution operation of E is using dilated convolutions that aggregate temporal information from neighboring time-frames [17, 21]. The residual connections, between the two convolution operations and prior to the application of the ReLU function, are used because they were experimentally found to help in minimizing the cost of Eq. (5).
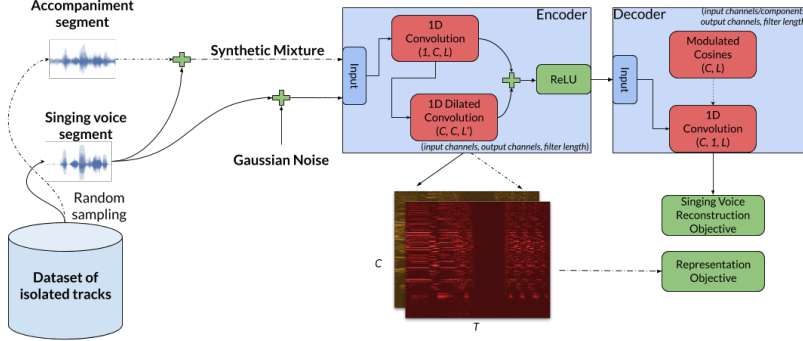
Figure 1: Overview of the proposed method.

In order to enforce the smoothness of $\mathbf{A}$, we employ a representation objective that the encoder has to minimize. Specifically, we use the representation of $\tilde{\mathbf{x}}_m$, $\mathbf{A}_m$, to compute the total variation denoising [22] $\mathcal{L}_{\text{TV}}$ as

$$\mathcal{L}_{\text{TV}}(\mathbf{A}_m) = \frac{1}{CT}\Big(\sum_{c=1}^{C-1} ||\mathbf{A}_{m[c;...]} - \mathbf{A}_{m[c-1;...]}|| + \sum_{t=1}^{T-1} ||\mathbf{A}_{m[...;t]} - \mathbf{A}_{m[...;t-1]}|| \Big) , \tag{1}$$

where $\mathbf{A}_{m[c;...]}$ and $\mathbf{A}_{m[...;t]}$ are the c-th row and t-th column vectors (repsectively) of the matrix $\mathbf{A}_m$, and $|| \cdot ||$ is the ℓ_1 vector norm. Eq. (1) penalizes E by the norm of the first order difference across both time-frames T and templates C. The former promotes slow time varying representations as the magnitude of the STFT representation, and the latter promotes a frequency dependent grouping of the activity.

The representation $\mathbf{A}_v$ is given to the decoder D for computing the reconstruction loss $\mathcal{L}_{\text{neg-SNR}}$, given in Section 2.2, that is used for the total loss for E computed as

$$\mathcal{L} = \mathcal{L}_{\text{neg-SNR}} + \lambda \mathcal{L}_{\text{TV}} \tag{2}$$

with λ being a weight for the impact of $\mathcal{L}_{\text{TV}}$ in the learning signal for E (i.e., a scalar that controls the smoothness).

2.2 The Decoder

The decoder D accepts the representation $\mathbf{A}_v$, models the clean singing voice signal $\mathbf{x}$ as a sum of C signal components that overlap in $\mathbb{R}^N$, and yields $\hat{\mathbf{x}}$ which is the approximation of $\mathbf{x}$. The aforementioned C components are computed by a strided convolution between the representation template $\mathbf{A}_{v[c;...]}$ and a kernel $\mathbf{w}_c \in \mathbb{R}^L$ of temporal length L as[2]

$$\mathbf{x} \approx \hat{\mathbf{x}} := \sum_{c=0}^{C-1} \mathbf{A}_{v[c;...]} * \mathbf{w}_c. \tag{3}$$

We use only $\mathbf{A}_v$ as input to the decoder D that computes $\hat{\mathbf{x}}$, because we are interested in learning a good representation for the singing voice, i.e., structured with low reconstruction error. To achieve a good representation in an unsupervised fashion by means of the DAE model [12], we assume that the distribution of the corruption process is constant for all segments in the data-set [23]. This cannot be assumed for music signal mixtures, as even the distribution of the accompaniment instruments can vary dramatically from one segment to another. Consequently, by making such an assumption it could lead to a degenerate singing voice representation.

Similar to Sinc-Net [14] and it's complex-valued extension for speech enhancement [15], we do not allow each $\mathbf{w}_c$ to be updated directly using back-propagation. Instead, we re-parameterize each $\mathbf{w}_c$ using sinusoidal functions and back-propagate through their corresponding parameters. More specifically, we compute each $\mathbf{w}_c$ using

$$\mathbf{w}_c = \cos(2\pi f_c^2 \odot \mathbf{t} + \phi_c) \odot \mathbf{b}_c , \tag{4}$$

where cos and $\odot$ are the element-wise cosine function and product, respectively, and $\mathbf{t} \in \mathbb{Z}^L$ is a vector denoting the integer time indices $\{0, \ldots, L-1\}$ of the kernels. These parameters of the cosine function are considered constants and are shared between the kernels. The sampling-rate-normalized carrier frequency f_c, the phase ϕ_c (in radians), and the modulating signal $\mathbf{b}_c$ are learnable and different for each kernel. The non-linear squaring operation applied to f_c is

---

[2]An appropriate zero-padding technique is assumed to be applied in order to deal with the differences between T and L.

motivated by the increased frequency resolution in lower frequencies that music signals commonly have [1], and is an experimental finding that is studied in Section 4. Using Eq. (4) for all C, our method constructs $\mathbf{W} \in \mathbb{R}^{C \times L}$ by stacking the corresponding outcome. After the stacking, a sorting operation is applied to $\mathbf{W}$, which sorts the kernels $\mathbf{w}_c$ in ascending order based on the normalized and squared carrier frequency f_c, promoting an intuitive representation. Then the decoding operation for $\mathbf{A}_v$ takes place using Eq. (3).

There are three reasons for using modulated cosine functions for decoding $\mathbf{A}_v$: a) cosine functions promote interpretability [14], i.e. the representation $\mathbf{A}$ is expected to convey amplitude related information for driving a well established synthesis model based on sinusoidal functions [24], b) the auto-encoding operation shares many similarities with the STFT yet without having to deal directly with the phase information, for which supervised based separation works remarkably well [2, 3], and c) amplitude modulations allow an extra degree of freedom in reconstructing signals that cannot be described by pure sinusoidal functions [24]. The latter statement is supported by the convolution theorem which states that the element-wise product of two vectors can be expressed in the Fourier domain as their corresponding convolution. Since in our re-parameterization scheme (i.e. Eq. (4)) one of the signals is a cosine function, then $\mathbf{b}_c$ is expected to convey information regarding fricatives and/or formants of the singing voice signal $\mathbf{x}$. Regarding on whether the proposed decoder is efficient in reconstructing the singing voice compared to either cosine functions or commonly employed convolutional layers, the reader is kindly referred to Section 4.

The optimization objective for D is the negative signal-to-noise ratio (neg-SNR) [25], defined as:

$$\mathcal{L}_{\text{neg-SNR}}(\mathbf{x}, \hat{\mathbf{x}}) = -10 \log_{10}\Big(\frac{||\mathbf{x}||_2^2}{||\mathbf{x} - \hat{\mathbf{x}}||_2^2} \Big) \text{, where} \tag{5}$$

$|| \cdot ||_2^2$ is the squared ℓ_2 vector norm, and the negative sign is used to cast the logarithmic SNR as a minimization problem.

3 Experimental Procedure

For training and evaluating the proposed method we use the MUSDB18 data-set [26] that consists of 150 two-channel multi-tracks, sampled at 44100Hz and split into training (100 multi-tracks) and testing (50 multi-tracks) subsets. During training we sample a set of four multi-tracks from which we use the vocals and the accompaniment sources. Each sampled multi-track is down-mixed to a single channel and is partitioned into overlapping segments of $N = 44100$ samples with an overlap of 22050 samples. We then randomly shuffle the segments for each source and corrupt the singing voice signal as described in Section 2. The standard deviation of the additive Gaussian noise corruption is set to $1e-4$. A batch of 8 segments per signal is used for optimizing the parameters of the proposed method, minimizing Eqs. (1)–(5) using adam algorithm [27] with a learning rate of $1e-4$. For choosing the convolution hyper-parameters we conducted an informal experiment employing 20 tracks from the training subset, followed by informal listening tests. This resulted into the following hyper-parameters: $C = 800$, $S = 384$, $L = 2048$, $L' = 5$, $D = 10$, and $\lambda = 0.5$. During the informal experiments, we noticed that the method converges fast so we set the total number of iterations throughout the whole data to 10. The choice for $N = 44100$ samples was based on the available computational resources.

For evaluation we use the rest 50 tracks, that are down-mixed to a single channel and segmented into non-overlapping signals. The shuffling and mixing are not considered in the evaluation stage. However, for evaluation we discard silent segments. We test the usefulness of the representation by performing masking-based informed singing voice separation, following the work presented in [28]. We do so because masking is an important operation that has been extensively used in both supervised-based singing voice separation [6] and assessing neural network-based latent representations [28]. To that aim, we employ the trained decoder (according to the previously described procedure), and reconstruct the time-domain signals of the un-corrupted singing voice representation and the binary masked mixture representation, respectively. The binary mask is computed using the encoded singing voice, accompaniment, and their corresponding mixture signals, that are available in the test sub-set. It is our intention to examine if this way of separation could be even possible in learned representations, as binary masking is a crude way to separate a signal in a given representation, due to the introduced *music distortion* caused by the abrupt changes of mask across the time-frames of the representation [1]. The reconstructed time-domain signals are used for computing the scale-invariant signal-to-distortion ratio (SI-SDR) [29] defined as

$$\text{SI-SDR}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10}\Big(\frac{||\alpha \mathbf{x}||_2^2}{||\alpha \mathbf{x} - \hat{\mathbf{x}}||_2^2} \Big), \text{ for } \alpha = \frac{\tilde{\mathbf{x}}^T \mathbf{x}}{||\mathbf{x}||_2^2} , \tag{6}$$

and is used computed for each segment. In the following section, we report the median value of SI-SDR across segments and three experimental runs.

Table 1: Results reflecting the decoding performance, by means of SI-SDR. Bold-faced numbers denote the best performance.

E/D Setup	Non-linearity	C	SI-SDR	N_P
conv/cos	N/A f_c^2	952	20.83 22.34	6.483M
conv/conv	N/A tanh(decoder)	800	31.25 30.50	6.476M
conv/mod-cos	N/A f_c^2	800	28.72 **32.62**	6.478M
sinc/mod-cos	f_c^2	952	26.82	6.487M

Using the above described procedure, we conduct two experiments. In the first experiment, we examine whether the modulated cosine functions (mod-cos) are a good synthesis model by measuring the reconstruction performance, after being optimized for the denoising task. We optimize various models that use the proposed training scheme presented in Section 2 *without* the random mixing corruption processes, and by employing the early stopping mechanism to terminate the training procedure if the model has stopped decreasing the loss expressed in Eq. (5) during the updates of the previous epoch. We consider various decoding strategies such as non-modulated cosine functions (cos), and common one-dimensional convolutional networks (conv) with and without the tanh non-linearity at the output. We also examine whether Sinc-Net [14] (sinc) can be used as the first encoding stage as proposed in [14]. For this experiment we have adapted C so that approximately the same number of parameters is used in each model.

For the second experiment we re-train the best combination of the above, using various values for the number of components $C \in \{400, 800, 1600\}$ and perform the reconstruction of the binary masked mixture signal. To examine the regularization effect of the total-variation (Eq. (1)) computed using the random mixing corruption process, we report each model's performance by using Eq. (1) for both $\mathbf{A}_v$ and $\mathbf{A}_m$, respectively. For comparison, we employ the STFT and perform the above described operations of analysis, masking, and synthesis. The STFT uses a hop-size of 384 samples, a window size of 2048 samples is performed and the hamming windowing function. The main difference between the first and the second set of experiments is that for the second set of experiments the modulated cosine functions are sorted after each gradient update, as explained in Section 2.2, whereas in the first they are not.

4   Results & Discussion

The obtained results from the two experiments are presented in Tables 1 and 2. Additional results, illustrations, and audio examples can be found online[3]. Table 1 demonstrates the median SI-SDR expressed in dB (the higher the better) yielded by the first experiment, along with additional information regarding the various setups for the encoder E and the decoder D, the number of parameters N_P (in millions M), the used number of components C, and the employed non-linearities. The results in Table 1 highlight three trends. Firstly, the application of the non-linearity to the normalized frequencies f_c results into better reconstruction performance compared to the linear case. The observed improvement is of $\sim$ 5dB on average across experimental configurations. Secondly, the modulated cosine functions serve as a good differentiable synthesis model for singing voice signals, outperforming simple cosine functions by approximately 8 dB on average, with respect to the two experimental configurations (with and without frequency scaling of the normalized frequency), and by 1.4 dB the best configuration of convolution based model (conv). Since SI-SDR is invariant to scale modifications of the assessed signal, 1.4 dB is a significant improvement of signal quality and does not imply a simple matching of the gain that the model based on modulated cosine functions might have exploited. Thirdly, Sinc-Net [14] does not bring further improvements to the proposed method.

Focusing on the separation performance of the obtained representations, Table 2 presents the median SI-SDR values of the binary masking separation scenario for three values for the hyper-parameter C and two regularization strategies including two different signal representations, the corrupted by Gaussian noise $\mathbf{A}_v$, and the synthetic mixtures using the accompaniment signals $\mathbf{A}_m$. The obtained results are compared to the STFT that has perfect reconstruction properties and masking techniques work very well in practice [1]. The results of Table 2 underline two main experimental findings. The first finding is that the binary masking can be used to separate sources using the proposed approach for representation learning. This can be seen from the $C = 1600$ model that uses the synthetic mixtures as an input to the unsupervised representation objective and achieves a SI-SDR median value of 6.68 dB. The second finding is that the proposed unsupervised representation objective, i.e., Eq. (1) with the synthetic mixtures, can be used to improve the reconstruction of the masked mixture signals without additional supervision, as previous studies suggest [28].

---

[3] https://github.com/Js-Mim/rl_singing_voice

Table 2: Results reflecting the informed separation performance, using a binary mask (BM), by means of the SI-SDR metric. Bold-faced numbers denote the best performance.

E/D Setup	C	$\mathcal{L}_{\text{TV}}(*)$	SI-SDR	BM SI-SDR	N_P
conv/mod-cos	400	$\mathbf{A}_\text{v}$	30.46	3.66	2.439M
		$\mathbf{A}_\text{m}$	30.73	5.93	
	800	$\mathbf{A}_\text{v}$	**32.28**	4.39	6.478M
		$\mathbf{A}_\text{m}$	32.11	6.28	
	1600	$\mathbf{A}_\text{v}$	31.94	4.68	19.356M
		$\mathbf{A}_\text{m}$	31.54	6.68	
STFT/iSTFT	1025	N/A	N/A	**8.80**	N/A

This claim is supported by the observed improvement of ~ 2 dB, on average across models of various components C, when the synthetic mixtures are used for the unsupervised representation objective. Nonetheless, there is much room for improvements in order to obtain the quality of the STFT/iSTFT approach that outperforms the best masked approximation of the proposed method by 2.12 dB.

5 Conclusions

In this work we presented a method for learning music signal representations in an unsupervised way. Our method is based on the denoising autoencoder model [12] and the differential digital signal processing concept [13]. The befits of our method are interpretability, non-negativity for real-valued music signal representations for driving an established synthesis model, based on cosine functions. We conducted a series of experiments where we investigated the reconstruction capabilities of the proposed method subject to auto-encoding and informed source separation using binary masks. Our results demonstrate a reconstruction above 30 dB of scale-invariant signal-to-distortion ratio, and that separation by masking is possible using the obtained representation. The latter, opens up directions for supervised approaches to masking-based separation. However, compared to the short-time Fourier transform and its inverse counterpart our results suggest that there is much room for improvements in order to achieve the benefits of that transform.

Acknowledgements

References

[1] Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An Overview of Lead and Accompaniment Separation in Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, Aug 2018.

[2] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - A Reference Implementation for Music Source Separation," *Journal of Open Source Software*, 2019. [Online]. Available: https://doi.org/10.21105/joss.01667

[3] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A Fast And State-of-the Art Music Source Separation Tool With Pre-trained Models," Late-Breaking/Demo ISMIR 2019, November 2019, deezer Research.

[4] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," HAL, Tech. Rep. 02379796v1, 2019.

[5] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-learning Extractors for Music Source Separation," in *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, May 2020.

[6] S. I. Mimilakis, K. Drossos, E. Cano, and G. Schuller, "Examining the Mapping Functions of Denoising Autoencoders in Singing Voice Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 266–278, 2020.

[7] P. Magron and T. Virtanen, "Online Spectrogram Inversion for Low-Latency Audio Source Separation," *IEEE Signal Processing Letters*, vol. 27, pp. 306–310, 2020.

[8] P. Magron, K. Drossos, S. I. Mimilakis, and T. Virtanen, "Reducing Interference with Phase Recovery in DNN-based Monaural Singing Voice Separation," in *Proc. Interspeech 2018*, 2018, pp. 332–336. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1845

[9] K. Drossos, S. I. Mimilakis, D. Serdyuk, G. Schuller, T. Virtanen, and Y. Bengio, "MaD TwinNet: Masker-Denoiser Architecture with Twin Networks for Monaural Sound Source Separation," in *Proceedings of the 2018 IEEE International Joint Conference on Neural Networks (IJCNN)*, July 2018.

[10] S. I. Mimilakis, K. Drossos, J. F. Santos, G. Schuller, T. Virtanen, and Y. Bengio, "Monaural Singing Voice Separation with Skip-Filtering Connections and Recurrent Inference of Time-Frequency Mask," in *Proceedings of the 43rd International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, 2018.

[11] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[13] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *International Conference on Learning Representations*, 2020.

[14] M. Ravanelli and Y. Bengio, "Interpretable Convolutional Filters with SincNet," in *International Conference on Neural Information Processing Systems: Workshop on Interpretability and Robustness for Audio, Speech and Language*, 2018.

[15] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank Design for End-to-end Speech Separation," 2019.

[16] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-To-End Source Separation With Adaptive Front-Ends," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018, pp. 684–688.

[17] Y. Fisher and K. Vladlen, "Multi-Scale Context Aggregation by Dilated Convolutions," in *International Conference on Learning Representations (ICLR)*, May 2016.

[18] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10.   Madison, WI, USA: Omnipress, 2010, p. 807–814.

[19] 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. [Online]. Available: http://jmlr.org/papers/v18/16-505.html

[20] P. Smaragdis and S. Venkataramani, "A Neural Network Alternative to Non-Negative Audio Models," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 86–90.

[21] K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, "Sound Event Detection with Depthwise Separable and Dilated Convolutions," 2020.

[22] D. Strong and T. Chan, "Edge-preserving and Scale-Dependent Properties of Total Variation Regularization," *Inverse Problems*, vol. 19, no. 6, pp. S165–S187, nov 2003.

[23] P. Vincent, "A Connection between Score Matching and Denoising Autoencoders," *Neural Comput.*, vol. 23, no. 7, p. 1661–1674, Jul. 2011. [Online]. Available: https://doi.org/10.1162/NECO_a_00142

[24] X. Serra, "A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition," Ph.D. dissertation, Stanford University, 1989. [Online]. Available: http://hdl.handle.net/10230/34072

[25] 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)*, Oct 2019, pp. 175–179.

[26] Z. Rafii, A. Liutkus, F. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 Corpus for Music Separation," Dec 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR-15)*, 2015.

[28] E. Tzinis, S. Venkataramani, Z. Wang, C. Subakan, and P. Smaragdis, "Two-Step Sound Source Separation: Training on Learned Latent Targets," in *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, May 2020.

[29] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – Half-baked or Well Done?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 626–630.