Unsupervised Music Source Separation Using Differentiable Parametric Source Models

Kilian Schulze-Forster, Gaël Richard, Liam Kelley, Clement Doire, Roland Badeau

Unsupervised Music Source Separation Using Differentiable Parametric Source Models

Kilian Schulze-Forster, Gaël Richard, *Fellow, IEEE*, Liam Kelley, Clement S. J. Doire, and Roland Badeau, *Senior Member, IEEE*

***Abstract*—Supervised deep learning approaches to underdetermined audio source separation achieve state-of-the-art performance but require a dataset of mixtures along with their corresponding isolated source signals. Such datasets can be extremely costly to obtain for musical mixtures. This raises a need for unsupervised methods. We propose a novel unsupervised model-based deep learning approach to musical source separation. Each source is modelled with a differentiable parametric source-filter model. A neural network is trained to reconstruct the observed mixture as a sum of the sources by estimating the source models' parameters given their fundamental frequencies. At test time, soft masks are obtained from the synthesized source signals. The experimental evaluation on a vocal ensemble separation task shows that the proposed method outperforms learning-free methods based on nonnegative matrix factorization and a supervised deep learning baseline. Integrating domain knowledge in the form of source models into a data-driven method leads to high data efficiency: the proposed approach achieves good separation quality even when trained on less than three minutes of audio. This work makes powerful deep learning based separation usable in scenarios where training data with ground truth is expensive or nonexistent.**

***Index Terms*—Unsupervised learning, audio source separation, signal processing, model-based, deep learning.**

I. Introduction

AUDIO source separation is the task of estimating the individual signals of several sound sources when only their mixture can be observed. When the sources are musical instruments (including singing voice), we refer to the task as Musical Source Separation (MSS) [1]. It has many applications, for example in up-mixing or re-mixing of recordings whose individual source signals are not accessible. It is also used to create play-along tracks for students of musical instruments. Furthermore, MSS is an important pre-processing step for several music information retrieval tasks such as automatic lyrics transcription [2]. Music mixtures are especially challenging because the source signals are usually highly correlated in time and frequency as opposed to speech or speech-noise mixtures [3]. Beyond, certain instruments may be present multiple times as distinct sources in music mixtures, e.g. several singers in a choir. Hereafter, we refer to this issue as *homogeneous sources*. State-of-the-art performance in MSS is achieved by Deep Neural Networks (DNNs) which are trained in a supervised fashion [4], [5], [6]. However, they have two shortcomings which we address in this paper.

Firstly, they are not able to separate homogeneous sources. For example, the methods in [4], [5], [6] are able to separate *all* singing voices from an instrumental accompaniment but provide only the mixture of these voices instead of further separating them into the different singer signals. Hence, they can neither be used to obtain only the *lead* vocals nor to separate vocal ensembles or violin quartets, for example.

Secondly, they require training data with available ground truth, i.e. mixtures for which target source signals are available in isolation. However, such isolated signals are difficult, sometimes impossible, to obtain for music mixtures. If the instruments were recorded separately, the ground truth signals exist but are usually not distributed. This is usually the case for pop music. For most other genres such as jazz, classical music, or folk, it is common practice that the musicians perform together in the same room and only the mixture of the instrument signals is recorded. Hence, no isolated signal recordings exist. Special recording sessions may be arranged in order to record signals in isolation [7], however, this is not only extremely costly but also leads to unnatural conditions for the musicians.

Therefore, there is a need for separation methods that do not require ground truth signals for training. Such methods may be learning-free or unsupervised.

Learning-free methods estimate all parameters directly from the test mixture [3]. Hence, they do not require any training data. Nonnegative Matrix Factorization (NMF) [8] and its numerous extensions have successfully been used for learning-free MSS [3]. Using side information such as musical scores [9], [10] or fundamental frequency (F0) [11], NMF-based methods can separate homogeneous sources.

Unsupervised methods have a training stage and require only mixtures (no isolated sources) for learning. At test time, their parameters are fixed. They have the potential to provide

Manuscript received 26 July 2021; revised 21 January 2022, 2 November 2022, and 27 January 2023; accepted 17 February 2023. Date of publication 3 March 2023; date of current version 24 March 2023. This work was supported by the European Union Horizon 2020 Research and Innovation Programme - Marie Skłodowsa-Curie under Grants 765068 and ERC, HI-Audio, 101052978. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jun Du. *(Corresponding author: Gaël Richard.)*

Kilian Schulze-Forster, Gaël Richard, Liam Kelley, and Roland Badeau are with the LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: gael.richard@telecom-paris.fr).

Clement S. J. Doire is with Sonos Inc., 75002 Paris, France.

Digital Object Identifier 10.1109/TASLP.2023.3252272

superior performance similar to supervised methods while being less demanding regarding data. Recently proposed unsupervised deep learning methods for audio source separation are based on assumptions such that the sources are uncorrelated [12], [13] or not homogeneous [14], [15]. Therefore, they are not applicable to music mixtures where sources are correlated and possibly homogeneous.

In this work, we propose and evaluate a novel approach to unsupervised source separation which does not make such assumptions. It is hence also applicable but not limited to music mixtures. The approach is inspired by the recent line of research which integrates signal processing models in DNNs to incorporate domain knowledge [16], [17]. Each source is modeled with a differentiable parametric source model. During training, the task of the DNN is to re-synthesize the observed mixture as a sum of the sources by estimating the source parameters. Separation is achieved because the F0s for all sources are estimated from the mixture and assigned to the sources beforehand. This can be done using existing methods such as [18], [19].

Besides being unsupervised and able to separate homogeneous sources, the approach has further advantages: high data efficiency as well as parametric, hence interpretable and modifiable, source estimates. Briefly, the contributions of this work are:

- a novel unsupervised deep learning approach for audio source separation,
- the integration of parametric source models in deep learning based audio source separation,
- a new differentiable procedure to estimate *stable* time-varying all-pole filters with a DNN using line spectral frequency parameterization,
- an extensive experimental evaluation of the proposed method on a musical source separation task and comparison to learning-free and supervised baselines,
- the open source code[1] for the proposed method and experiments.

The rest of the paper is structured as follows: In Section II we review related work on audio source separation and model-based deep learning. The proposed method is explained in Section III and its experimental evaluation is outlined in Section IV. We present and discuss results in Section V and conclude in Section VI.

II. Related Work

In this section we review work on homogeneous musical source separation, learning-free and unsupervised source separation, and, finally, on the integration of signal processing models in deep neural networks.

Homogeneous audio sources are not easily distinguishable in the time-frequency domain and pose a permutation problem [20], [21]. While permutation-invariant training is used for supervised speech separation [21], [22], methods for musical homogeneous source separation exploit side-information such as F0 estimates [11], [23] or a musical score [9], [10], [24] to guide the separation. Two deep learning approaches for supervised choir separation were proposed recently. In this context, a choir is composed of four homogeneous sources: a soprano, alto, tenor, and a bass singer. Petermann et al. [23] modified the conditioned U-Net [25] so that the target source can be selected and separated using its F0 information. Results show that this leads to improved objective separation quality compared to using non-informed source-specific models. However, ground truth source signals are needed for training and they are rare for choir recordings. This motivated Gover and Depalle [24] to synthesize choir singing from MIDI files and to use this synthetic data for training of a score-informed DNN. When tested on real choir recordings, the model is outperformed by the learning-free, score-informed NMF proposed in [9]. This shows that the performance of supervised DNNs depends strongly on the quality and quantity of the training data.

Therefore, learning-free methods are a powerful alternative in limited data settings. Several separation methods based on NMF are learning-free and can exploit side-information to separate homogeneous sources. NMF approximates a spectrogram with a matrix product of two low-rank matrices containing spectral templates and their activations, respectively [3]. Ewert and Müller [9] proposed to initialize both templates and activations using musical score information. This leads to improvements compared to random initialization. Using the score allows even to separate notes played by the left and the right hand in piano recordings. Similarly, Hennequin et al. [10] used a musical score to initialize the activations whereas the templates consist of parametric frequency atoms. Durrieu et al. [11] formulated an advanced signal model using multiple NMF decompositions. The predominant source is modeled with a source-filter model and all other sources are captured by an unconstrained NMF. First, the F0 of the predominant target source is estimated using the signal model. Then, the F0 is used to guide the separation. Nakamura and Kameoka [26] proposed a powerful signal model combining NMF and harmonic-temporal clustering and integrated a source-filter model. It allows for blind, learning-free separation of harmonic sounds. A drawback of NMF-based methods is the low degree of flexibility because only a fixed number of spectral templates is used to describe a signal. This limits their performance, especially when inherent assumptions are violated. Recently, efforts have been made to make more flexible deep learning based source separation also usable in cases where no mixture-target pairs are available for training. Most works focus on creating learning targets artificially from mixtures or side-information in order to train DNNs in a supervised way in the absence of real targets. Seetharaman et al. [15] obtain targets for singing voice/accompaniment separation by clustering time-frequency bins of mixtures using several simple perceptual cues. Hung et al. [27] obtain harmonic target masks from well-aligned musical scores and further support the training process using score transcription models. Also deep clustering models [20] have been trained for speaker separation without ground truth signals [28], [29]. The targets are obtained by clustering the mixture based on spatial information. The methods above yield good results but require substantial amounts

[1][Online]. Available: https://github.com/schufo/umss

of (unlabeled) training data and cannot separate homogeneous correlated sources.

As an alternative, it has been proposed to train deep generative models on isolated source signals to use them subsequently for source separation [14] or speech enhancement [30]. However, this strategy is challenging for MSS because it requires a large amount of isolated source signals and uncorrelated sources.

Lastly, mixture invariant training has been proposed recently in [12] and refined in [13] for unsupervised learning of audio source separation without a need for artificial targets. During training, the sum of two mixtures is given as an input and the DNN has to separate all sources so that, given the respective optimal binary mixing matrices, the two mixtures can be reconstructed individually. Since it is necessary that the sources are uncorrelated [13], this approach is not an option for MSS.

The method proposed in this paper uses F0 information to separate the (possibly homogeneous) sources like the learning free-methods of [9], [11] and the supervised methods of [23], [24]. It provides better performance than learning-free methods and does not require expensive labeled data like supervised methods. Our learning strategy is fundamentally different from other unsupervised methods: it is not limited to uncorrelated sources like [13] and does not rely on artificial source targets which require the availability of aligned scores [27], sufficient spatial information in the mixture [28], [29], or non-homogeneous sources [15]. The proposed training objective is to re-synthesize the mixture with differentiable parametric source models. The only assumptions are that the number of sources is known and that their F0s can be estimated. In contrast to the unsupervised methods reviewed above, the proposed one can separate homogeneous sources, requires only a small amount of unlabelled data, and provides interpretable and modifiable source estimates.

There is a recent line of research that explores the combination of data-driven and knowledge-based methods to take advantage of both paradigms [16], [17], [31]. The integration of differentiable source models in the DNN-based source separation process is inspired by this model-based deep learning research. Specifically related to our work are recent speech synthesis methods which use differentiable parametric voice models and estimate their parameters using DNNs [32], [33]. We use similar voice models but in a different context. Engel et al. [17] implemented a code library for differentiable digital signal processing and show the advantages of model-based deep learning for tasks such as synthesis, timbre transfer and dereverberation. The DNN architectures and the differentiable signal processing implementations we use in our experiments are inspired by their work. To the best of our knowledge, the proposed method is the first one that uses model-based deep learning for MSS.

III. Method

We observe the single-channel mixture $m(t) = \sum_{j=1}^{J} s_j(t)$ of J monophonic source signals $s_j(t)$ where $t \in \{1, \ldots, T\}$ indexes discrete time samples. Our goal is to estimate all source signals s_j. We propose a novel approach to train a DNN for this task without access to any isolated source signal. The sources are modeled with differentiable parametric source models which we describe in Section III-A. The DNN estimates the source parameters given the F0 as explained in Section III-B. The objective of the unsupervised training strategy is to re-synthesize the mixture. Details are given in Section III-C and an overview of the procedure is presented in Fig. 1. At test time, the synthesized source signals can either be used directly as source estimates or soft masks can be derived from them for Wiener filtering of the mixture. Implementation details are described in Section III-D

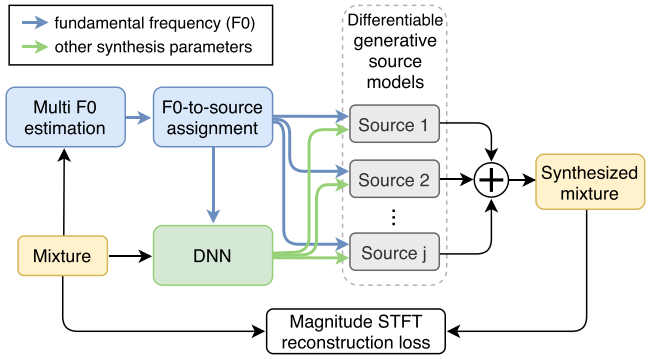


Fig. 1. Overview of the proposed unsupervised training procedure of a Deep Neural Network (DNN) for audio source separation.

A. Source Model

The proposed method is not specific to any particular source model and any parametric model may be used as long as it can be formulated in a differentiable way. This is often facilitated by automatic differentiation software such as TensorFlow [34] or PyTorch [35]. In this work, we use the source-filter model of speech production [36]. It describes a signal as an excitation signal from a sound source (e.g. the glottis) which is modified by a time-varying filter (e.g. the vocal tract) [36]. It is used to model a wide range of signals such as human voice [33], [36], [37] and musical instruments [11], [38]. An visualization of our source-filter model is presented in Fig. 2. In the following, we assume that the true source signal $s_j(t)$ is segmented into N frames of length T' samples. The n-th frame is given by

$$s_j(n,t) = s_j(t + nB), \quad t \in \{1, \ldots, T'\} \tag{1}$$

where B is the hop size between frames in samples and $n \in \{1, \ldots, N\}$. We denote the estimate of the source signal frame generated by the source model using a tilde: $\tilde{s}_j(n,t)$. The source model may be formulated in the z-domain as

$$\tilde{S}_j(n,z) = E_j(n,z)\frac{1}{A_j(n,z)}. \tag{2}$$

$E_j(n,z)$ is the z-transform of the excitation signal $e_j(n,t)$ and $\frac{1}{A_j(n,z)}$ is the transfer function of a time-varying all-pole filter of order K. We drop the source index j for brevity hereafter but we would like to emphasize that each source is modeled with its dedicated model. The filtering process in (2) is best described

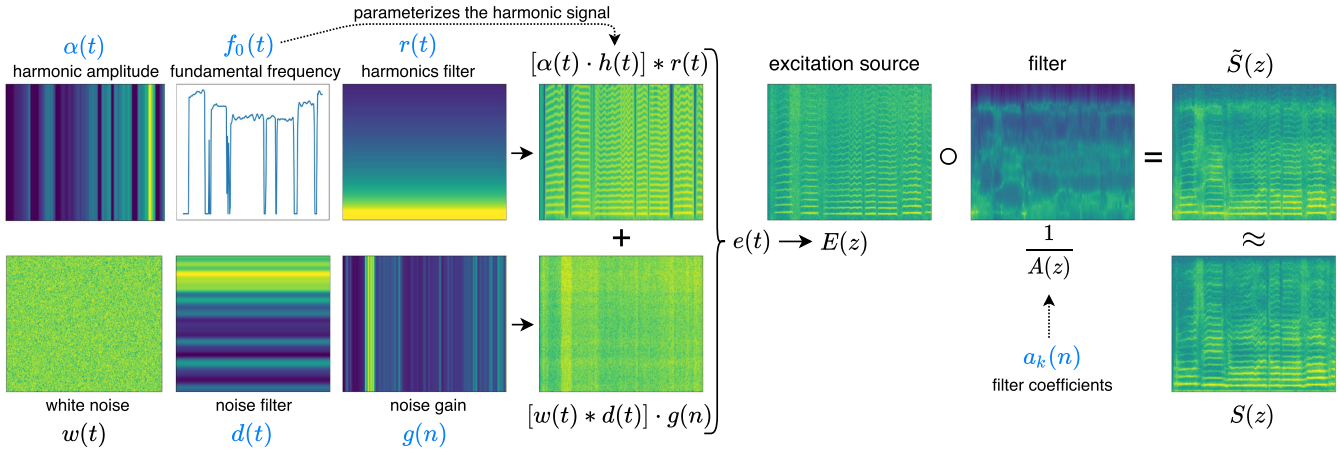


Fig. 2. Overview of the source-filter model decomposition. The model parameters are denoted in blue font. The '$\circ$' denotes element-wise multiplication. Although most components are visualized through magnitude spectrograms, processing is not necessarily done in the time-frequency domain.

by the difference equation

$$\tilde{s}(n,t) = e(n,t) - \sum_{k=1}^{K} a_k(n) \cdot \tilde{s}(n,t-k) \tag{3}$$

where $a_k(n)$ are the filter coefficients for frame n and '$\cdot$' denotes scalar multiplication. We explain how to deal with frame boundaries and other implementation details in Section III-D.

A sinusoids plus noise model is employed to generate the excitation signal $e(n,t)$. It is an expressive synthesis model for music [39] and speech signals [40], [41], [42] which synthesizes sound as a sum of sinusoids and filtered white noise. A differentiable version was recently implemented by Engel et al. [17], [43] who showed impressive results using it for model-based deep learning. Since we model a monophonic source, we constrain the sinusoid frequencies to be integer multiples of a fundamental frequency. The model thus reduces to the *harmonics plus noise* model [17], [40] which we formulate as

$$e(n,t) = [\alpha(n,t) \cdot h(n,t)] * r(t) + [w(t) * d(t)] \cdot g(n) \tag{4}$$

where $*$ denotes the convolution operator, $\alpha(n,t)$ is the time-varying amplitude of the harmonic signal $h(n,t)$, and $r(t)$ and $d(t)$ are Impulse Responses (IR) of time-invariant finite impulse response (FIR) filters. $w(t)$ is a uniform white noise signal and $g(n)$ is the constant noise gain for frame n.

The harmonic signal $h(n,t)$ is defined as

$$h(n,t) = \sum_{i=1}^{I} \sin(\phi_i(n,t)) \tag{5}$$

$$\phi_i(n,t) = 2\pi \sum_{v=1}^{t} i \cdot f_0(n,v)/f_s \tag{6}$$

where ϕ_i is the instantaneous phase of the i-th harmonic, f_0 is the fundamental instantaneous frequency, and f_s is the sampling frequency. The initial phase is assumed to be zero. Equation (6) is a numerical approximation of integration based on *sample and hold* [44, Ch. 4]. Note that the signal $h(n,t)$ is fully parameterized by the time-varying fundamental frequency f_0.

The filter $r(t)$ imposes a fixed spectral shape on $h(n,t)$. Without $r(t)$, all sinusoids have the same amplitude. However, for certain sound sources a specific time-invariant spectral shape can be assumed, e.g. the spectral roll-off of the glottal signal [36]. Alternatively, a specific amplitude parameter may be used for each sinusoid in $h(n,t)$ [39], [40]. However, we choose to make the gain dependent on the frequency and not on the harmonic number. Similarly, $d(t)$ determines the spectral shape of the noise component. Both filters are time-invariant so that they only account for the global spectral shape. Short term variations, e.g. due to articulations of words, are modeled by the all-pole filter $\frac{1}{A(n,z)}$.

The source model parameters are $\{a_k(n), \alpha(t), f_0(t), r(t), g(n), d(t)\}$. In the next section, it is explained how they are obtained. α and f_0 need to vary slowly enough over time for the model to be mathematically identifiable. This is indirectly enforced by the way these parameters are estimated which leads to smooth trajectories.

B. Parameter Estimation

We assume that the fundamental frequencies for each of the J sources can be obtained from the mixture signal with a multiple F0 estimation system. Given that many such systems exist [18], [45], [46] and that it is still an active research area, we are confident that this is a reasonable assumption. When all F0s are obtained, each F0 value needs to be assigned to one specific source. Various solutions for the F0-to-source assignment problem have been proposed [19], [47], [48]. Most of them are based on principles such as temporal pitch continuity, low voice crossing probability, and minimal temporal gaps within a voice [48]. In our experiments we use a heuristic based on these principles, cf. Section IV-B. F0 estimates are usually provided at a frame rate which is smaller than the sample rate [18], [45], [46]. Therefore, following [17], the source specific F0 time series are upsampled to the sample rate using bilinear interpolation. This leads to smooth trajectories.

In the following, we describe how the remaining synthesis parameters are estimated with a DNN for each source given its

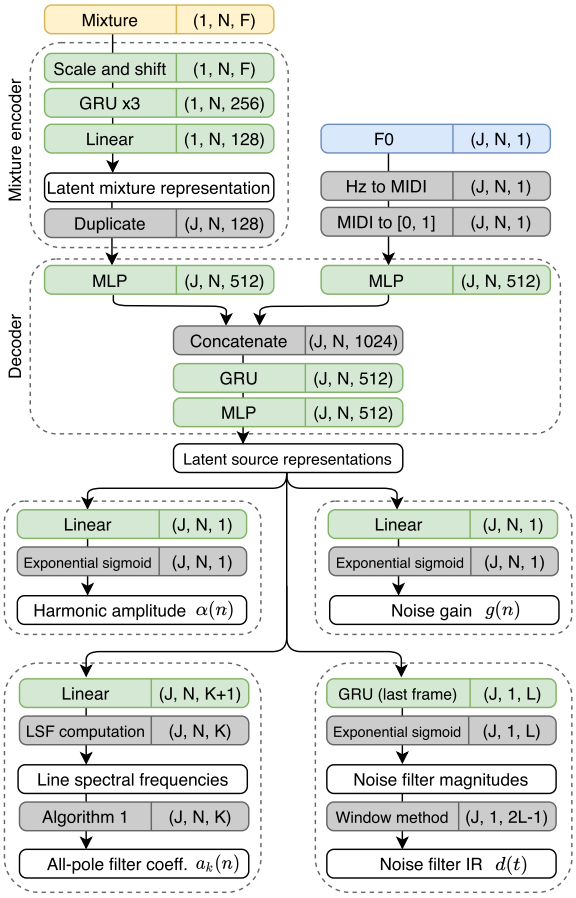


Fig. 3. Overview of the processing steps for the parameter estimation. Transformations with learnable parameters are shown in green, predefined processing steps in gray, (intermediate) outputs in white boxes. The output shape of a transformation is shown in the right part of the box.

F0. The task the DNN has to solve is similar to the one of NMF in the context of learning-free F0-informed source separation in [9], [11]. Note that the differentiable source models do not put any constraints on the neural network type or architecture which is used to estimate the parameters. Here we use a simple DNN as in [17] and focus on the advantages of including parametric source models in deep learning based separation.

The mixture signal is represented by the logarithmic magnitude of its spectrogram obtained by a Short Time Fourier Transform (STFT) of $m(t)$. The spectrogram has F frequency bins and N time frames. Each spectrogram is normalized by subtraction of its mean and division by its standard deviation. Then, each frequency bin is scaled and shifted by dedicated learned scalars. The DNN architecture is similar to the one used in [17]. An overview of the DNN and further processing steps for the parameter estimation is presented in Fig. 3. We use linear layers and unidirectional Recurrent Neural Networks (RNN) with Gated Recurrent Units (GRUs) [49]. The Multi-Layer Perceptron (MLP) consists of three repetitions of linear layer, layer normalization [50], Leaky ReLU activation [51].

The mixture encoder learns a latent representation of the mixture and then creates as many duplicates as there are sources. Each latent mixture copy is then combined with the F0 information of one source by the decoder. The F0 is provided at the frame rate of the mixture STFT. The F0 values are converted from Hertz to MIDI note numbers which are then normalized to the interval $[0, 1]$. The decoder computes a separate latent representation for each source. The source model parameters are obtained from this source representation by one last transformation with learned parameters (linear layer or GRUs) followed by some predefined processing steps. The frame-wise harmonic amplitude $\alpha(n)$ and the noise gain $g(n)$ are computed with a linear layer with an exponential sigmoid activation function [17] defined as

$$y = y_{\text{max}} \cdot \text{sigmoid}(x)^{\log(10)} + 10^{-7} \tag{7}$$

where x and y are the input and output value, respectively, and y_{max} is a scalar determining the upper bound of y. Following [17], the harmonic amplitude is then upsampled to the sample rate using overlapping Hann windows which yields a smooth $\alpha(t)$. The noise gain is only required at frame rate.

The filter with impulse response $d(t)$ is time-invariant. Therefore, the network output from which $d(t)$ is computed should summarize information about the whole source signal. We obtain such an output by processing the latent source representation with a unidirectional RNN with GRUs and then using only the output at the last time frame for further processing. This last output frame is processed with the exponential sigmoid presented in (7) which results in a tensor of shape $(J, 1, L)$. The tensor contains L samples of the magnitudes of the single-sided frequency responses of the noise filters for J sources. The samples define a zero-phase FIR filter according to the frequency sampling method [52]. Using the window method [53], we obtain the impulse response $d(t)$ as it is also done in [17].

The impulse response $r(t)$ of the time-invariant harmonics filter can be obtained in the same way as $d(t)$ from a DNN output. One may also wish to make the filters time-varying by using a linear layer for the last transformation or using all GRU outputs. However, for the scope of this work, we fix $r(t)$ manually. More details about $r(t)$ are given in Section IV-B where we describe the experimental setup.

For the estimation of the parameters we addressed so far, practical ways have already been proposed by Engel et al. [17]. More care needs to be taken when obtaining Infinite Impulse Response (IIR) filters such as $\frac{1}{A(z)}$ from DNN outputs because it must be avoided that the filter becomes unstable. The filter $\frac{1}{A(z)}$ of order K is fully defined by the filter coefficients a_k with $k \in \{1, \ldots, K\}$ (see also the difference equation in (3)). However, no condition which guarantees stability can be formulated for the filter coefficients directly.

Different parameterizations of all-pole filters exist which allow for the formulation of stability criteria. One option would be to estimate K reflection coefficients [54] with the DNN. Stability is guaranteed if the coefficients are within the interval $]-1, 1[$. They can be converted to the filter coefficients with a simplified version of the Levinson-Durbin algorithm [55], [56], see also [54]. This approach was used in [33] to define the all-pole vocal tract filter with a DNN for speech synthesis. The drawback of this method is that conclusions about the filter's

frequency response can neither be drawn from the reflection coefficients nor the filter coefficients.

Therefore, we choose to parameterize the all-pole filter with Line Spectral Frequencies (LSFs) [57]. LSFs are related to the positions of the filter poles and thus to the frequency response [54]. Hence, they provide an interpretable parametrization. They also allow the formulation of constraints to control the filter response and can be interpolated [58]. LSFs were introduced in [57] as an alternative representation of linear prediction coefficients. Below, we briefly explain their definition and how we use them. For a comprehensive overview of LSFs, we refer the reader to [58], [59], [60].

The polynomial $A(z) = 1 - \sum_{k=1}^{K} a_k z^{-k}$ can be decomposed into the symmetric and antisymmetric polynomials $P(z)$ and $Q(z)$ of order $K+1$ as

$$A(z) = \frac{P(z) + Q(z)}{2}. \tag{8}$$

It can be shown that if the roots of $P(z)$ and $Q(z)$ alternate on the unit circle, the corresponding filter $\frac{1}{A(z)}$ is stable and minimum-phase [59]. The unit circle in the z-plane is described by $z = e^{-j\omega}$ where ω is the phase angle in radiants. Hence, ω describes the location of the roots. If K is even, $P(z)$ has a root at $z = -1$ and $Q(z)$ has a root at $z = +1$. The remaining roots occur in complex conjugate pairs. Therefore, it is sufficient to consider only the roots on the upper semicircle. The angles ω_k defining the locations of these complex roots are called LSFs. Two to three LSFs tend to be close together when a filter pole is close to the unit circle in their proximity which corresponds to a peak in the frequency response, hence their frequency domain interpretation. If K is even, $P(z)$ and $Q(z)$ have $K/2$ complex roots on the upper unit semicircle each, for which the following relation holds:

$$0 < \omega_k < \omega_{k+1} < \pi. \tag{9}$$

When k is odd, ω_k defines a root of $P(z)$; when k is even, it defines a root of $Q(z)$ for $k \in \{1, \ldots, K\}$. In other words, a stable minimum-phase filter $\frac{1}{A(z)}$ of order K is defined by K LSFs fulfilling the relation in (9).

We obtain such LSFs as follows. The latent source representations are transformed by a linear layer which yields a tensor of shape $(J,\ N,\ K+1)$. It is processed by an exponential sigmoid activation with $y_{max} = 2$. The resulting tensor can be viewed as $J \cdot N$ vectors $\mathbf{v} \in \mathbb{R}^{K+1}$. The vectors are normalized so that their entries v_k sum up to π:

$$\bar{\mathbf{v}} = \frac{\mathbf{v}}{\sum_{k=1}^{K+1} v_k} \cdot \pi. \tag{10}$$

The K LSFs respecting (9) are then obtained by the cumulative sum

$$\omega_k = \sum_{i=1}^{k} \bar{v}_i \quad \text{for} \quad k = 1, \ldots, K. \tag{11}$$

Algorithm 1: Compute filter coefficients a_k from ω_k [60], [61]

Input: $(\omega_k)_{k=1:K}$
  Define: $x_k = \cos(\omega_k)$
  Initialize: $p'_{-1} = q'_{-1} = 0; \quad p'_0 = q'_0 = 1$
  Initialize: $p'_1 = -2x_1; \quad q'_1 = -2x_2$
  for $k = 2$ **to** $K/2$ **do**
    $p'_k = -2p'_{k-1}x_{2k-1} + 2p'_{k-2}$
    $q'_k = -2q'_{k-1}x_{2k} + 2q'_{k-2}$
    for $i = (k-1)$ **to** 1 **do**
      $p'_i = p'_i - 2p'_{i-1}x_{2k-1} + p'_{i-2}$
      $q'_i = q'_i - 2q'_{i-1}x_{2k} + q'_{i-2}$
    end for
  end for
  for $k = 1$ **to** $K/2$ **do**
    $p_k = p'_k + p'_{k-1}$
    $q_k = q'_k - q'_{k-1}$
  end for
  for $k = 1$ **to** $K/2$ **do**
    $a_k = (p_k + q_k)/2$
    $a_{(K/2+k)} = (p_{(K/2-k+1)} - q_{(K/2-k+1)})/2$
  end for
Output: $(a_k)_{k=1:K}$

Finally, the LSFs can be converted to filter coefficients using Algorithm 1 [61],[2] [58], [60].

To sum up the parameter estimation, F0s are estimated from the mixture and assigned to the sources using existing methods. $a_k(n)$, $\alpha(t)$, $g(n)$, and $d(t)$ are obtained with a DNN and $r(t)$ is fixed manually in this work but may also be estimated by a DNN.

C. Unsupervised Training

The proposed training procedure requires only mixture signals, no isolated source signals are needed. During training, the task of the DNN is to reconstruct the observed mixture by estimating the corresponding parameters of the source models. A schematic overview of the training process is presented in Fig. 1. The generated mixture estimate $\tilde{m}(t)$ is the sum of the source signals generated by the source models:

$$\tilde{m}(t) = \sum_{j=1}^{J} \tilde{s}_j(t). \tag{12}$$

In theory, the source models make it possible to synthesize a mixture estimate $\tilde{m}(t)$ which is perceptually identical to the true mixture $m(t)$. Since absolute phase offsets are irrelevant for human perception, the true and estimated mixtures do not need to have the same phase. Therefore, the reconstruction loss $\mathcal{L}_{rec}$ is formulated as a multi-scale spectral loss [17]

$$\mathcal{L}_c = \|\mathbf{M}_c - \tilde{\mathbf{M}}_c\|_1 + \|\log(\mathbf{M}_c) - \log(\tilde{\mathbf{M}}_c)\|_1 \tag{13}$$

[2]The formulation of Algorithm 1 which we present in this paper has been presented in [61]. Some equations in the main body of [61] contain errors but the Matlab code in the Appendix is correct. A less general formulation is found in [58, Ch. 8]. The conversion was formally introduced in [60].

$$\mathcal{L}_{rec} = \sum_c \mathcal{L}_c \tag{14}$$

where $\mathbf{M}_c$ and $\tilde{\mathbf{M}}_c$ denote the magnitude spectrograms of the input mixture and its estimate, respectively, and $c = [2048, 1024, 512, 256, 128, 64]$ indicates the FFT size used to compute the STFT. The frames overlap by 75%.

The separation of the sources is essentially ensured by the assignment of the F0s to the sources similar to score/F0-informed separation with NMF [9], [11]. The DNN has to estimate the remaining parameters for each source in order to minimize the loss. At test time, the DNN parameters are fixed and a soft mask for source j is obtained by the element-wise division $\tilde{\mathbf{S}}_j / \sum_{j=1}^{J} \tilde{\mathbf{S}}_j$ where $\tilde{\mathbf{S}}_j$ is the magnitude spectrogram of the generated source signal $\tilde{s}_j$. The *final* time-domain source estimates, $\hat{s}_j$ are obtained by Wiener filtering using the soft masks.

D. Implementation Details

We implemented the proposed method using the PyTorch framework [35]. For the differentiable source models, we make use of the DDSP library [17]. We re-implemented it in PyTorch and added extensions such as Algorithm 1 and an all-pole filter. The code is available online.[3]

Using an all-pole filter in the proposed framework entails two challenges. Firstly, the autoregressive filtering process is slow because it does not allow for precise parallel processing of frames. Secondly, the filter is time-varying, i.e. its coefficients are different at every frame. Therefore, extra care must be taken to ensure a smooth transition between frames to avoid artefacts. The DNN operates at a frame rate which is determined by the FFT size T' and hop size B used to compute the STFT of the mixture. Hence, the DNN estimates a set of K filter coefficients for each frame. We apply the all-pole filter to all frames in parallel using the difference equation in (3) in order to make filtering faster. The initial states $\tilde{s}(n, t)$ with $t \leq 0$ are set to zero for each frame. The output frames are then multiplied with a Hann window and the final output signal is obtained by the overlap-add method. It is therefore important that the hop size B is chosen so that the Hann window respects the constant overlap-add condition. We use $B = T' / 2$ in our experiments. Windowing and 50% overlap make the transition between frames smooth. The errors that are introduced by setting the initial states to zero instead of taking samples of the previous frame into account (which is not possible in parallel processing) are negligible: Firstly, the errors are larger at the start of each frame where their importance is mitigated by the window. Secondly, since the filter coefficients are different at each frame, the importance of samples from the previous frame is reduced.

We found it to be critical to implement Algorithm 1 with double precision (64-bit floating point) because it is more sensitive to rounding errors with increasing filter order, which can lead to unstable filters.

The excitation signal $e(t)$ is computed as follows. The harmonic component $\alpha(t) \cdot h(t)$ and the noise $w(t)$ are generated in the time domain for the entire signal length T. The time-invariant FIR filters $r(t)$ and $d(t)$ and the noise gain $g(n)$ are applied frame-wise in the frequency domain followed by overlap-add.

IV. Experiments

We evaluate the proposed approach on an *a cappella* vocal ensemble separation task. The goal is to estimate the individual signals of J singers from their mixture. This task is a good choice for evaluation because sources in vocal ensembles are homogeneous and correlated. Moreover, singing voice is a challenging musical source. It has a strongly time-varying spectral envelop and also produces sounds without any harmonic content such as unvoiced consonants. Also, only small amounts of data for supervised training are available for vocal ensemble separation. This makes unsupervised learning an important alternative.

A. Data

As training and validation data, we use the Bach Chorals (BC) dataset[4] and the Barbershop Quartet (BQ) dataset[5]. The BC set contains 26 chorals sung by a vocal quartet with the voices Soprano, Alto, Tenor, Bass (SATB). The BQ set contains 22 songs performed by a vocal quartet comprising the voices tenor, lead, baritone and bass. All voices are available in isolation for both sets. This allows us to compare the proposed unsupervised approach to supervised baselines.

We combine the BC and BQ sets to generate what we call the *full* training and validation sets. The *full* validation set comprises songs 8 and 9 of the BC set and songs 8 and 9 of the BQ set and has a total length of 9 minutes and 10 s. The remaining songs build the *full* training set with a total length of 91 minutes and 20 s. We also build a *small* training set consisting of BC song 1 with a length of 2 minutes and 40 s and a *small* validation set consisting of BC song 2 with a length of 2 minutes and 20 s. When mixtures with less than four singers are created from the individual voice recordings, all possible combinations of the four voices with the desired number of singers are used with the constraint of using only one singer per voice.

As test data, we use the Choral Singing Dataset [7]. It comprises three songs performed by an SATB choir with four singers per voice. All 16 singer signals are available in isolation which allows to evaluate the separation with objective metrics. We add the signals of individual singers (max. one per voice) to produce the test mixtures. For mixtures of $J = 4$ singers, the test set has a length of 6 minutes and 48 s. For mixtures of $J = 2$ singers, the test set has a length of 40 minutes and 48 s due to more possible voices combinations.

We resample the training, validation, and test data to a sample rate of 16 kHz. The training examples are excerpts of 4 s length which are randomly drawn from the training set. The validation and test set are split into fixed excerpts of 4 s length. There is no overlap regarding singers, songs, or recording setup between the test and training data. While the training data contain a

[3][Online]. Available: https://github.com/schufo/umss

[4][Online]. Available: https://www.pgmusic.com/bachchorales.htm

[5][Online]. Available: https://www.pgmusic.com/barbershopquartet.htm

considerable amount of reverberation, the test recordings are much less reverberant.

B. Experimental Setup

We perform two sets of experiments: one using mixtures of $J = 2$ singers for training and testing, and the other using mixtures of $J = 4$ singers.

The F0s are obtained from the mixture signals using the multiple F0 estimation model of Cuesta et al. [18]. We use the pre-trained "Model 3" which is available online.[6] For the F0-to-source assignment on the given data, we found that a simple heuristic is sufficient. It is based on the same principles as more advanced solutions such as temporal pitch continuity, low voice crossing probability, and minimal temporal gaps within a voice [19], [48]. The F0 estimator provides x F0 values at each time frame. First, we process all frames where $x = J$. The F0 values are sorted according to magnitude and assigned to the voices assuming they do not cross. Subsequently, the remaining frames are processed. When $x < J$ we assume that some voices are silent. We assign each F0 value to the source which has the closest F0 value in a previous or subsequent frame (pitch continuity principle). The zero value is assigned to silent sources. In the rare case that $x > J$, we sort the values according to magnitude and select J F0s using the pitch continuity principle and assign them to the sources.

The mixture spectrograms are computed using an FFT size of $T' = 512$ and a hop size of $B = 256$ samples. Hence, they have $F = 257$ frequency bins and $N = 250$ time frames. We fix the impulse response $r(t)$ so that the frequency response of the FIR filter falls off with a rate of 6 dB/octave, with a reference frequency of 200 Hz below which the response is flat. We chose this rate because it accounts for the combined spectral characteristics of the glottal source and lip radiation [36]. Estimating $r(t)$ with the DNN instead did not lead to improvements. We set the order of the all-pole filter to $K = 20$. The spectrograms of the synthesized source signals $\tilde{\mathbf{S}}_j$ to compute the soft masks are computed with an FFT size of 2048 and a hop size of 256 samples.

Training is done with the ADAM optimizer [62], a batch size of 16 and a learning rate of 0.0001. Training is stopped after 200 consecutive epochs without improvement of the validation loss.

We train the model with the proposed unsupervised approach on the *full* and on the *small* training set. We call the experiments UnSupervised-Full (US-F) and UnSupervised-Small (US-S), respectively. As a reference, we also train the same model in a supervised way on the same data. In this case, the loss is computed for each source estimate individually using its target. The total loss is the sum of the "source losses". We call these experiments SuperVised-Full (SV-F) and SuperVised-Small (SV-S).

Since the proposed method is dependent on available F0 estimations, we also evaluate its robustness to F0 estimation errors. The corresponding experiments and results are explained in Section V-B.

[6]https://github.com/helenacuesta/multif0-estimation-polyvocals

C. Baselines

We compare the proposed unsupervised approach to two learning-free methods and one supervised approach. The baselines also exploit F0 information and compute soft masks for Wiener filtering. The first learning-free baseline was proposed by in [9]. It approximates the mixture magnitude spectrogram with a simple NMF decomposition:

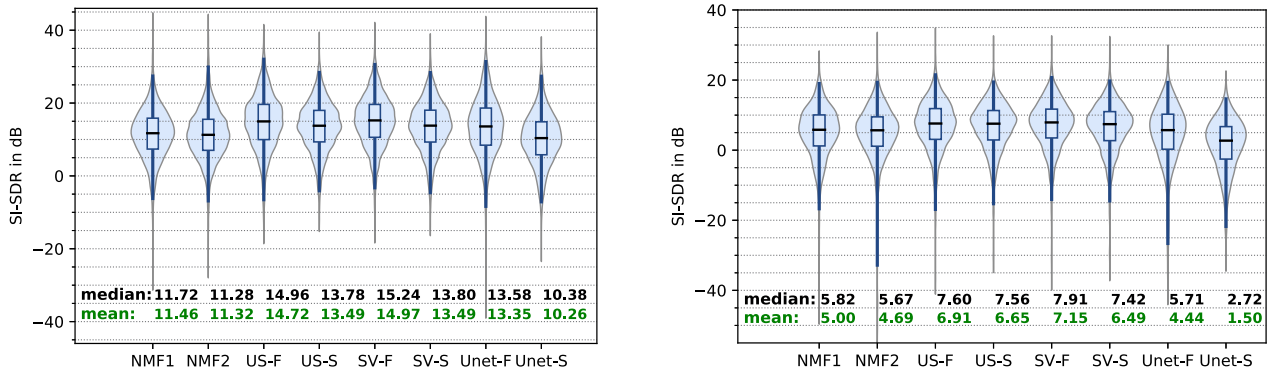
$$\mathbf{M} \approx \hat{\mathbf{M}} = \mathbf{W}\mathbf{H} \tag{15}$$

where $\mathbf{W} \in \mathbb{R}^{F\times R}$ is a matrix of R spectral templates and $\mathbf{H} \in \mathbb{R}^{R\times N}$ contains their activations over N time frames. In [9], $\mathbf{W}$ and $\mathbf{H}$ are initialized using information from an aligned musical score. One spectral template per semitone is used. In our experiments, we have F0 information available, which is more precise than a semitone scale. Therefore, we use a scale with a precision of $\frac{1}{10}$ of a semitone. The F0 values are converted from Hertz to MIDI numbers which are rounded to one decimal place for this purpose. The F0s are used for initialization and for the separation to determine which activations belong to which source. After testing different combinations, we obtained the best results with an FFT size of 2048 and a hop size of 256 samples to compute the spectrograms. We call this method NMF1.

The second learning-free baseline is the method proposed by Durrieu et al. [11]. The target source is modeled with a source-filter model and the residual sources are modeled with a conventional NMF. The method approximates the power spectrogram of the mixture $\mathbf{M}_{\text{pow}}$ as

$$\mathbf{M}_{\text{pow}} \approx \hat{\mathbf{M}}_{\text{pow}} = \underbrace{(\mathbf{W}^{\Gamma}\mathbf{H}^{\Gamma}\mathbf{H}^{\Phi})}_{\text{filter}} \circ \underbrace{(\mathbf{W}^{F0}\mathbf{H}^{F0})}_{\text{source}} + \underbrace{(\mathbf{W}^{O}\mathbf{H}^{O})}_{\text{residual}} \tag{16}$$

where $\circ$ denotes element-wise multiplication. $\mathbf{W}^{\Gamma} \in \mathbb{R}^{F\times P}$ contains P spectral atoms consisting of shifted Hann windows with 75% overlap so that the whole frequency range is covered across $\mathbf{W}^{\Gamma}$. The matrix $\mathbf{H}^{\Gamma} \in \mathbb{R}^{P\times K}$ contains their activations to combine them to smooth filters and $\mathbf{H}^{\Phi} \in \mathbb{R}^{K\times N}$ contains activations to combine the smooth filters. $\mathbf{W}^{F0} \in \mathbb{R}^{F\times U}$ contains a fixed set of U spectral templates defined by the glottal source model KLGLOTT88 [63]. There is one spectral template for each F0 in steps of $\frac{1}{20}$ semitone between a minimum and a maximum frequency. $\mathbf{H}^{F0} \in \mathbb{R}^{U\times N}$ contains the activations of the spectral templates. In [11], $\mathbf{H}^{F0}$ is initialized using F0 information of the predominant source estimated using the signal model in (16). We initialize $\mathbf{H}^{F0}$ using the F0 information we obtained from the multi-pitch estimation [18]. In [11], the spectral templates of the residual sources $\mathbf{W}^{O} \in \mathbb{R}^{F\times R}$ and their activations $\mathbf{H}^{O} \in \mathbb{R}^{R\times N}$ are initialized randomly. We initialize them using the F0 information for the corresponding sources as done in NMF1. This leads to improvements. We call this baseline NMF2. The parameters to be estimated are $\{\mathbf{H}^{\Gamma}, \mathbf{H}^{\Phi}, \mathbf{H}^{F0}, \mathbf{W}^{O}\mathbf{H}^{O}\}$. For NMF2, we obtained the best results using an FFT size of 1024 and a hop size of 128 samples. To the best of our knowledge, these two baselines are among the best learning-free, informed methods for musical and homogeneous source separation.



(a) $J = 2$ sources (b) $J = 4$ sources

Fig. 4. Violin plots and box plots of the SI-SDR values in dB for all evaluation frames. The boxes extend from the first to the third quartile, the medians are marked with a black horizontal line. The box plot whiskers (dark blue) extend from the first to the 99th percentile. The violin plots extend over the whole data range. In (b), NMF2 has five outliers between -60 and -80 dB which are not shown.

Furthermore, we train the F0-informed supervised deep learning approach for vocal ensemble separation proposed by Petermann et al. [23] on our data. They use a classical U-Net architecture with a control mechanism [25]. The F0 information is used to select the target source and to guide the separation. For this baseline, mixture and target source spectrograms are computed using an FFT size of 1024 and a hop size of 256 samples. Wiener filtering is applied at test time using all J source estimates to compute soft masks. It is trained with the ADAM optimizer [62], a batch size of 16 and a learning rate of 0.001. We train this baseline on the full and the small training set and call the experiments Unet-F and Unet-S, respectively. Note that all baselines make use of the F0 information for the separation.

V. Results and Discussion

A. Experimental Results

The separation quality was evaluated using the objective metric Scale-Invariant Source-to-Distortion ratio (SI-SDR) [64]. It is computed on evaluation frames of one second length without overlap as usually done for musical source separation evaluation [65]. The results for the cases of $J = 2$ and $J = 4$ sources are shown in Fig. 4(a) and (b), respectively. The data points for the box plots and violin plots are the SI-SDR values in dB for all evaluation frames in the test set. Target source frames, in which the sum of squares of the samples is below a threshold of 10, are considered to be silent and thus excluded from the evaluation. For methods in which random numbers are involved, the evaluation was run with five different seeds to initialize the pseudorandom number generator. These methods are NMF2 (random initialization of $\mathbf{H}^{\Gamma}$ and $\mathbf{H}^{\Phi}$) and the proposed approach (random white noise) used in experiments US-F, US-S, SV-F, and SV-S.

We conducted two-sided t-tests [66] to assess whether the means of the SI-SDR score distributions are significantly different for each pair of experiments in our study. We used a Levene test [67] to assess whether a pair of SI-SDR score distributions has the same variance or not. If true, the comparison was made with a Student's t-test. If false, Welch's t-test [68] was used. The resulting p-values [66] are shown in Fig. 5(a) and (b) for $J = 2$ and $J = 4$, respectively. Most p-values are extremely small being in the order of 10^{-4} or smaller. This indicates that the corresponding means are significantly different. It can be seen that a few p-values are considerably larger. In this case it is more likely that the true means are not different.

In general, the SI-SDR is higher for the separation of mixtures of two sources compared to the four sources case. However, the relative performance of the methods is the same for both cases with the exception that Unet-F outperforms NMF1 and NMF2 when $J = 2$ but not when $J = 4$. This is related to the small amount of training data for a supervised deep learning model. Listening examples are available online[7].

The proposed unsupervised method (US-F, US-S) performs better than the baselines. Its performance is very close to the one which is reached by the same model trained in a supervised way: SV-F is only slightly better than US-F, while SV-S and US-S have the same performance (p-values of 0.9507 and 0.164 for $J = 2$ and $J = 4$). This means that the proposed method achieves almost the same performance whether isolated target sources are available for training or not. This can be explained by the fact that the F0 information is used very efficiently by the proposed method. The F0 fully parameterizes the harmonic source component $h(t)$ and, hence, defines the corresponding source to a large extent. The DNN has to determine the remaining parameters which, given the F0, can be inferred from the mixture. Hence, isolated source targets do not carry major additional information.

Another interesting observation is that the performance of the proposed method does not drop drastically when the amount of training data is decreased by 97% (US-F vs. US-S and SV-F vs. SV-S). For $J = 2$, a decrease in SI-SDR can be seen but it is smaller than for the supervised baseline (Unet-F vs. Unet-S). For $J = 4$, the performance difference of the proposed approach is very small when comparing training on the full and the

[7]https://schufo.github.io/umss/

	NMF1	NMF2	US-F	US-S	SV-F	SV-S	Unet-F	Unet-S
NMF1	-	0.2154	$1.2 \cdot 10^{-164}$	$7.8 \cdot 10^{-72}$	$5.3 \cdot 10^{-193}$	$2.3 \cdot 10^{-69}$	$1.5 \cdot 10^{-31}$	$9.0 \cdot 10^{-16}$
NMF2	0.2154	-	$< 10^{-300}$	$3.2 \cdot 10^{-224}$	$< 10^{-300}$	$1.8 \cdot 10^{-220}$	$6.0 \cdot 10^{-52}$	$5.9 \cdot 10^{-19}$
US-F	$1.2 \cdot 10^{-164}$	$< 10^{-300}$	-	$2.3 \cdot 10^{-70}$	$3.9 \cdot 10^{-4}$	$3.9 \cdot 10^{-70}$	$2.2 \cdot 10^{-24}$	$2.8 \cdot 10^{-275}$
US-S	$7.8 \cdot 10^{-72}$	$3.2 \cdot 10^{-224}$	$2.3 \cdot 10^{-70}$	-	$9.7 \cdot 10^{-109}$	0.9507	0.2858	$3.5 \cdot 10^{-157}$
SV-F	$5.3 \cdot 10^{-193}$	$< 10^{-300}$	$3.9 \cdot 10^{-4}$	$9.7 \cdot 10^{-109}$	-	$3.7 \cdot 10^{-108}$	$6.1 \cdot 10^{-34}$	$< 10^{-300}$
SV-S	$2.3 \cdot 10^{-69}$	$1.8 \cdot 10^{-220}$	$3.9 \cdot 10^{-70}$	0.9507	$3.7 \cdot 10^{-108}$	-	0.3006	$2.3 \cdot 10^{-156}$
Unet-F	$1.5 \cdot 10^{-31}$	$6.0 \cdot 10^{-52}$	$2.2 \cdot 10^{-24}$	0.2858	$6.1 \cdot 10^{-34}$	0.3006	-	$5.7 \cdot 10^{-78}$
Unet-S	$9.0 \cdot 10^{-16}$	$5.9 \cdot 10^{-19}$	$2.8 \cdot 10^{-275}$	$3.5 \cdot 10^{-157}$	$< 10^{-300}$	$2.3 \cdot 10^{-156}$	$5.7 \cdot 10^{-78}$	-

(a) $J = 2$ sources

	NMF1	NMF2	US-F	US-S	SV-F	SV-S	Unet-F	Unet-S
NMF1	-	0.1739	$1.6 \cdot 10^{-18}$	$1.2 \cdot 10^{-14}$	$3.0 \cdot 10^{-23}$	$2.9 \cdot 10^{-12}$	0.0684	$1.5 \cdot 10^{-35}$
NMF2	0.1739	-	$2.1 \cdot 10^{-53}$	$2.0 \cdot 10^{-46}$	$5.3 \cdot 10^{-70}$	$1.2 \cdot 10^{-39}$	0.3357	$1.8 \cdot 10^{-43}$
US-F	$1.6 \cdot 10^{-18}$	$2.1 \cdot 10^{-53}$	-	0.0335	0.0554	$5.2 \cdot 10^{-4}$	$4.0 \cdot 10^{-22}$	$1.3 \cdot 10^{-121}$
US-S	$1.2 \cdot 10^{-14}$	$2.0 \cdot 10^{-46}$	0.0335	-	$2.9 \cdot 10^{-5}$	0.164	$2.0 \cdot 10^{-18}$	$1.5 \cdot 10^{-113}$
SV-F	$3.0 \cdot 10^{-23}$	$5.3 \cdot 10^{-70}$	0.0554	$2.9 \cdot 10^{-5}$	-	$2.5 \cdot 10^{-8}$	$2.2 \cdot 10^{-26}$	$3.0 \cdot 10^{-132}$
SV-S	$2.9 \cdot 10^{-12}$	$1.2 \cdot 10^{-39}$	$5.2 \cdot 10^{-4}$	0.164	$2.5 \cdot 10^{-8}$	-	$4.0 \cdot 10^{-16}$	$1.4 \cdot 10^{-107}$
Unet-F	0.0684	0.3357	$4.0 \cdot 10^{-22}$	$2.0 \cdot 10^{-18}$	$2.2 \cdot 10^{-26}$	$4.0 \cdot 10^{-16}$	-	$2.1 \cdot 10^{-21}$
Unet-S	$1.5 \cdot 10^{-35}$	$1.8 \cdot 10^{-43}$	$1.3 \cdot 10^{-121}$	$1.5 \cdot 10^{-113}$	$3.0 \cdot 10^{-132}$	$1.4 \cdot 10^{-107}$	$2.1 \cdot 10^{-21}$	-

(b) $J = 4$ sources

Fig. 5. The p-values of pair-wise t-tests between the distributions of SI-SDR values for all experiments.

small training set. For the unsupervised version the difference is probably not significant since the p-value of 0.0335 for the comparison of US-F and US-S is larger than most other p-values. In contrast, the SI-SDR of the Unet baseline drops strongly for $J = 4$ as well. This shows that it is beneficial to integrate domain knowledge in the form of explicit source models in the separation model.

We believe that the main difference in performance between US-S and Unet-S is indeed related to the usage of the F0 information. In Unet, the F0 information is globally used. In our case, the F0 information is directly exploited to produce harmonic signals using the explicit source production models. The neural network only infers the remaining information (vocal tract filter, noise content, etc). This makes the task easier for the neural network which can explain the substantially higher performance.

To sum up, the proposed unsupervised model-based deep learning approach to source separation performs better than learning-free and supervised purely data-driven baselines. It is also extremely efficient in learning from data. The method is useful in many scenarios where homogeneous sources need to be separated and/or only a very small amount of data (possibly without ground truth) is available for training. Besides choir separation as in our experiments, such scenarios may be the separation of lead from background vocals or of traditional music with less common instrumentation. Since only mixtures are needed for training, the proposed model may also be trained directly on the mixtures at hand which are to be separated. Given sufficient computational resources, parameter optimization may also be done directly on each test mixture individually, which would make the method learning-free.

B. Robustness to F0 Estimation Errors

We propose herein two experiments to analyse the impact of multi-F0 estimation errors on the performances of our models, for the more complex case of 4 sources.

1) Building a Reference multi-F0 Annotation: For this analysis study, we built a reference multi-F0 annotation for each time frame (16 ms) of the test set. Since we have access to the individual solo music tracks for each mixture signal, we opted for an automatic annotation of the solo tracks using a state-of-the-art pitch estimator (CREPE) [69]. Our proposed models are named *oracle models* when they are run using this reference multi-F0 annotation.

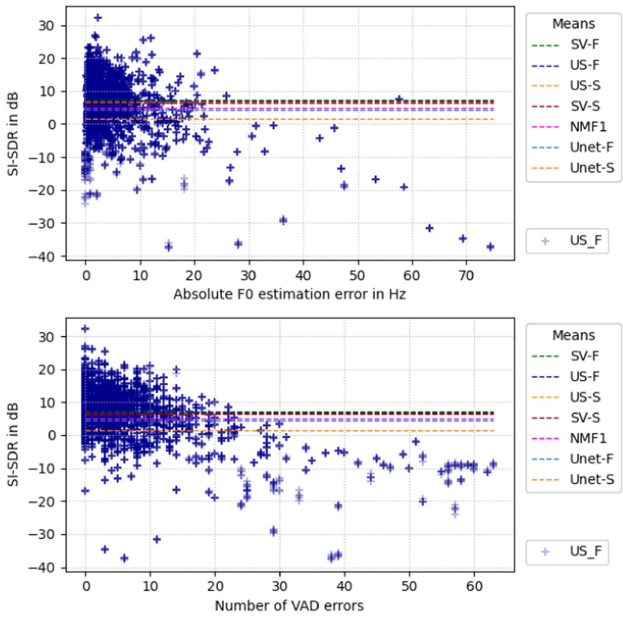


Fig. 6. Separation performance for the US-F model for each evaluation frame of 1 s of the test set (4 sources) as a function of F0 precision (top) and VAD errors (bottom).

2) Experiment 1: Analysis of F0 Estimation Errors: This experiment aims at analysing further the evaluation results obtained in Section V-A for the model US-F. Using the reference multi-F0 annotation described above as ground truth, we can identify the errors made by our multi-F0 estimator [18] in our experiments for each time frame of 16 ms. A *Voice Activity Detection (VAD) error* is observed where one of the sources is declared active in the reference multi-F0 annotation and not by our multi-F0 estimator or vice-versa. When there are no VAD errors, it is possible to evaluate the *F0-precision* which is defined as the absolute deviation in Hertz between the value given by the reference multi-F0 annotation and our multi-F0 estimator.

Each evaluation frame of 1 s of the test set can then be labelled with the number of effective VAD errors and the mean F0-precision for the correctly detected voices. The computed SI-SDR obtained by our model US-F for each of these frames are displayed on Fig. 6. It can be observed that the performances

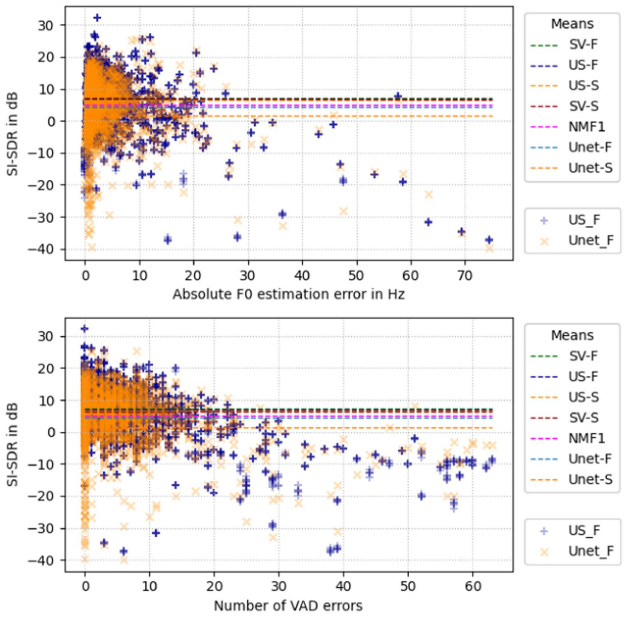


Fig. 7. Comparison of the separation performances for the US-F and Unet-F models for each evaluation frame of 1 s of the test set (4 sources) as a function of F0 precision (top) and VAD errors (bottom).

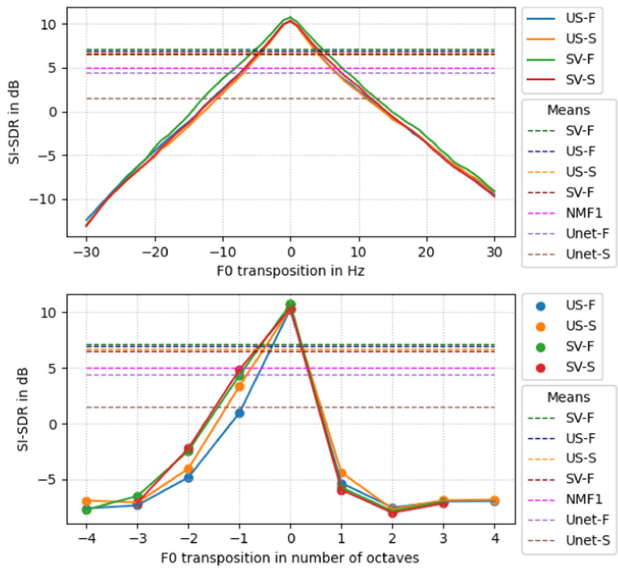


Fig. 8. Effect of F0 values transposition. (top) transposition in Hz (bottom) transposition in octaves. The dotted lines show the mean SI-SDR using the multi-F0 estimation method without manual degradation (cf. Fig. 4).

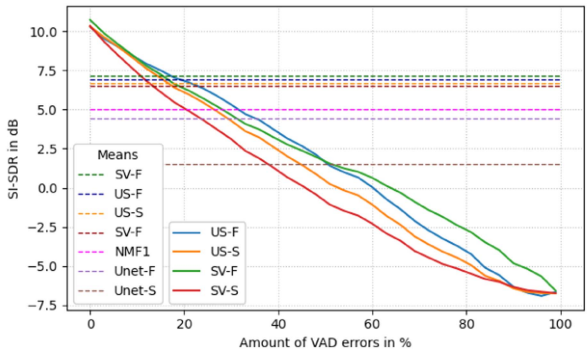


Fig. 9. Effect of voice muting.

are, as expected, impacted by F0 estimation errors. The degradation of performances remains limited for frames with moderate amount of errors and the algorithm is more robust to VAD errors. The performances distribution for the supervised methods are similar overall but, in the case of the Unet methods, there are more frames with low SI-SDR when there are no VAD/F0 errors (see Fig. 7). The unsupervised approaches seem to benefit more than the supervised approaches from a correct estimation of VAD and F0, as already discussed in Section V-A, but they are slightly more fragile in the case of severe estimation errors.

Note that the level of SI-SDR $= -7.23$ dB corresponds to the result obtained on the test set by a "dummy" separator where all estimated sources are attenuated replicas of the mixture. It can then be noticed that our model is inoperative or detrimental (SI-SDR ≤ -7.23 dB) only on a very limited number of evaluation frames, which mostly correspond to frames with particularly high numbers of VAD errors or high F0 estimation deviations.

3) Experiment 2: Robustness to Noisy multi-F0 Estimations: We evaluate the impact of noisy F0 estimations on the performance of our models by manually degrading the reference F0 annotations. Two different alterations are considered, namely:

- *Transposition:* All reference F0 frequencies are shifted upwards or downwards by a predefined number of Hertz or by one or several octaves
- *Voice-muting:* All voices for a given time frame of 16 ms have a given probability to be muted (the corresponding multi-F0 values are set to zero) forcing as a consequence a predefined VAD error rate.

The results obtained for the effect of transposition are given in Fig. 8. First, it can be seen that the oracle models (our proposed models using the reference F0 values without degradation) obtain better results than the same models using F0 values given by our multi-F0 estimator (more than +3 dB on average). Second, the degradation of the performance is smoothly varying with the precision in Hz of the F0 values demonstrating the degree of tolerance of the algorithm with respect to multi-F0 estimation errors. Third, for a perturbation of less than 5 Hz, the model is still outperforming all baseline models. And finally, the models do not increase the mean SI-SDR ($= -7.23$ dB) of the "dummy" separator only when the shift exceeds roughly 23 Hz (which corresponds to at least a semi-tone for all notes below A4 (440 Hz) which can be considered a severe error). For octave errors, the degradation is very rapid when the F0 values are transposed upwards but remains moderate when transposing downwards. This may be explained by the fact that when the fundamental frequency is twice as low as the true value, every even harmonic of the source falls exactly at the position of the true target source.

Fig. 9 gives the results when one or several voices are locally muted. The models remain efficient when the percentage of VAD errors is below 20%. The roughly linear shape of the degradation curve underlines the robustness of the model to such detection

errors. The models trained on the full training set (US-F and SV-F) are more robust, but the supervised model seems more fragile than the unsupervised model when trained with the small dataset.

C. Limitations and Perspectives

The experimental evaluation showed many advantages of the proposed approach compared to various alternatives. Nevertheless, there are some limitations. First, our approaches assume that the number of sources is known. Although we have shown that our methods are somewhat robust to moderate VAD errors, they are currently limited in cases where the number of sources is unknown. However, the most striking limitation is more precisely that the method requires F0 estimates which are assigned to the sources. As for all F0-informed separation methods, the sources should exhibit mainly harmonic content and be monophonic so that the separation can be guided by the F0 information. It requires that good F0 estimates can be obtained for all sources from the mixture. As shown in the experiments, this is possible with existing methods. Progress in research on multiple F0 estimation may lead to further improvements. An extension of our method to polyphonic sources as well as estimating the F0 jointly with the other source parameters may be an interesting direction for future work. In its current form, the model would not perform well on more diverse mixtures of music sources as are for example contained in the popular MUSDB dataset [70] which includes drums, inharmonic and polyphonic sources. Moreover, audio effects such as reverberation or distortion, which may have been applied to the sources, should be explicitly modeled in the source models and must hence be known beforehand. Lastly, the space complexity grows linearly with the number of sources to be modeled.

In the experiments above, the final source estimates were obtained by Wiener filtering of the mixture. To this end, soft masks were obtained from the source signals $\tilde{s}_j$ generated by the source models. We also evaluated the quality of the generated signals $\tilde{s}_j$ as source estimates. The metric used for this evaluation was the spectral source-to-noise ratio [71]. It can be seen as a SI-SDR which is computed on magnitude spectrograms. We used this spectral metric because the phase of the generated signals is known not to be the same as the one of the ground truth signals. This makes a time-domain evaluation not applicable.

In terms of this metric, the quality of such source estimates was inferior to the baselines and to $\hat{s}_j$ obtained using soft masks. This is because the synthesis of the signals $\tilde{s}_j$ is less constrained than masking of the mixture. The output of masking is limited by the frequency content of the mixture, since masking can only keep or remove such content. In contrast, frequency content which is not present in any source can be contained in $\tilde{s}_j$. In fact, the DNN tends to overestimate the noise content of the sources. While this is clearly audible in $\tilde{s}_j$, no noise is added in $\hat{s}_j$.

Nevertheless, we believe that source estimates generated by parametric models are a worthwhile goal for future research. They provide a complete parameterization of the mixture signal which can be exploited for tasks such as timbre or style transfer, transposition, and melody editing of single sources. We included the generated source signals $\tilde{s}_j$ and their sum $\tilde{m}$ in the audio examples[8]. Moreover, we provide two examples of melody editing for which the mixture parametrization was exploited.

VI. Conclusion

In this work, we presented a method for (musical) audio source separation which overcomes two limitations of state-of-the-art supervised deep learning methods: They do not separate homogeneous sources and require large datasets of mixtures with the corresponding sources in isolation for training. We proposed a novel unsupervised model-based deep learning approach. It integrates domain knowledge in the form of differentiable parametric source models in a data-driven method and exploits F0 information. Experiments show that it outperforms learning-free and supervised baselines. Furthermore, the method performs well even when trained on less than three minutes of audio data. It allows to apply powerful deep learning based separation in domains where training data is expensive or nonexistent.

Acknowledgment

The authors would like to thank Emmanouil Benetos for providing the Bach Chorals and Barbershop Quartet dataset.

References

[1] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, "Musical source separation: An introduction," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 31–40, Jan. 2019.
[2] E. Demirel, S. Ahlbäck, and S. Dixon, "Automatic lyrics transcription using dilated convolutional neural networks with self-attention," in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2020, pp. 1–8.
[3] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. Hoboken, NJ, USA: Wiley, 2018.
[4] N. Takahashi and Y. Mitsufuji, "D3Net: Densely connected multidilated densenet for music source separation," 2020, *arXiv:2010.01733*.
[5] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," 2019, *arXiv:1909.01174*.
[6] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, no. 41, 2019, Art. no. 1667.
[7] H. Cuesta, E. Gómez Gutiérrez, A. Martorell Domínguez, and F. Loáiciga, "Analysis of intonation in unison choir singing," in *Proc. Int. Conf. Music Percep. Cogn.*, 2018.
[8] D. Lee and S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[9] S. Ewert and M. Müller, "Using score-informed constraints for NMF-based source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 129–132.
[10] R. Hennequin, B. David, and R. Badeau, "Score informed audio source separation using a parametric model of non-negative spectrogram," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 45–48.
[11] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1180–1191, Oct. 2011.
[12] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey, "Unsupervised sound separation using mixture invariant training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 3846–3857.
[13] S. Wisdom, A. Jansen, R. J. Weiss, H. Erdogan, and J. R. Hershey, "Sparse, efficient, and semantic mixture invariant training: Taming in-the-wild unsupervised sound separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acous.*, 2021, pp. 51–55.

[8] https://schufo.github.io/umss/

[14] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias, "Unsupervised audio source separation using generative priors," in *Proc. Interspeech* 2020, pp. 2657–2661.

[15] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo, "Bootstrapping unsupervised deep music separation from primitive auditory grouping principles," in *Proc. Workshop Self-Supervision Audio Speech 37th Int. Conf. Mach. Learn.*, 2020.

[16] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," 2020, *arXiv:2012.08405*.

[17] J. Engel et al., "DDSP: Differentiable digital signal processing," in *Proc. Int. Conf. Learn. Representations*, 2019.

[18] H. Cuesta, B. McFee, and E. Gómez, "Multiple f0 estimation in vocal ensembles using convolutional neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 302–309.

[19] R. Schramm et al., "Multi-pitch detection and voice assignment for a cappella recordings of multiple singers," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 552–559.

[20] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 31–35.

[21] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 241–245.

[22] 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 Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.

[23] D. Petermann, P. Chandna, H. Cuesta, J. Bonada, and E. Gomez, "Deep learning based source separation applied to choir ensembles," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 733–739.

[24] M. Gover and P. Depalle, "Score-informed source separation of choral music," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 231–239.

[25] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2019, pp. 159–165.

[26] T. Nakamura and H. Kameoka, "Harmonic-temporal factor decomposition for unsupervised monaural separation of harmonic sounds," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 68–82, 2021.

[27] Y.-N. Hung, G. Wichern, and J. Le Roux, "Transcription is all you need: Learning to separate musical mixtures with score as supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 46–50.

[28] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 695–699.

[29] E. Tzinis, S. Venkataramani, and P. Smaragdis, "Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 81–85.

[30] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 101–105.

[31] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[32] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5916–5920.

[33] A. R. MV and P. K. Ghosh, "SFNet: A computationally efficient source filter model based neural speech synthesis," *IEEE Signal Process. Lett.*, vol. 27, pp. 1170–1174, 2020.

[34] M. Abadi et al., "Tensorflow: A system for large-scale machine learning," in *Proc. Symp. Oper. Syst. Des. Implementation*, 2016, pp. 265–283.

[35] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[36] G. Fant, *Acoustic Theory of Speech Production*. Berlin, Germany: Walter de Gruyter, 1970.

[37] G. Degottex, P. Lanchantin, A. Roebel, and X. Rodet, "Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis," *Speech Commun.*, vol. 55, no. 2, pp. 278–294, 2013.

[38] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 327–332.

[39] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Comput. Music J.*, vol. 14, no. 4, pp. 12–24, 1990.

[40] J. Laroche, Y. Stylianou, and E. Moulines, "HNM: A simple, efficient harmonic+noise model for speech," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 1993, pp. 169–172.

[41] G. Richard and C. d'Alessandro, "Analysis/synthesis and modification of the speech aperiodic component," *Speech Commun.*, vol. 19, no. 3, pp. 221–244, 1996.

[42] D. Erro, I. Sainz, E. Navas, and I. Hernaez, "Harmonics plus noise model based vocoder for statistical parametric speech synthesis," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 2, pp. 184–194, Apr. 2014.

[43] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," *in Proc. Workshop Self-Supervision Audio Speech Int. Conf. Mach. Learn.*, 2020.

[44] P. Horowitz and W. Hill, *The Art of Electronics*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[45] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2006, pp. 216–221.

[46] W. Zhang, Z. Chen, and F. Yin, "Multi-pitch estimation of polyphonic music based on pseudo two-dimensional spectrum," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2095–2108, 2020.

[47] E. Chew and X. Wu, "Separating voices in polyphonic music: A contig mapping approach," in *Proc. Int. Symp. Comput. Music Model. Retrieval*, 2004, pp. 1–20.

[48] A. McLeod and M. Steedman, "HMM-based voice separation of MIDI performance," *J. New Music Res.*, vol. 45, no. 1, pp. 17–26, 2016.

[49] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics Struct. Stat. Transl.*, pp. 103–111, 2014.

[50] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[51] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 30, no. 1, p. 3.

[52] J. O. Smith, *Spectral Audio Signal Processing: Frequency Sampling Method*. Stanford, CA, USA: W3K Publishing, 2011. [Online]. Available: https://ccrma.stanford.edu/jos/sasp/Frequency_Sampling_Method_FIR.html

[53] J. O. Smith, *Spectral Audio Signal Processing: Generalized Window Method*. Stanford, CA, USA: W3K Publishing, 2011. [Online]. Available: https://ccrma.stanford.edu/jos/sasp/Generalized_Window_Method.html

[54] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*. Hoboken, NJ, USA: Prentice Hall Press, 2010.

[55] N. Levinson, "The wiener (root mean square) error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, no. 1–4, pp. 261–278, 1946.

[56] J. Durbin, "The fitting of time-series models," *Revue de l'Institut International de Statistique*, vol. 28, pp. 233–244, 1960.

[57] F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals," *J. Acoust. Soc. Amer.*, vol. 57, no. S1, pp. S35–S35, 1975.

[58] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. Hoboken, NJ, USA: Wiley, 2004.

[59] F. Soong and B. Juang, "Line spectrum pair (LSP) and speech data compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 1984, vol. 9, pp. 37–40.

[60] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 6, pp. 1419–1426, Dec. 1986.

[61] I. V. McLoughlin, "Line spectral pairs," *Signal Process.*, vol. 88, no. 3, pp. 448–467, 2008.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[63] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *J. Acoust. Soc. Amer.*, vol. 87, no. 2, pp. 820–857, 1990.

[64] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 626–630.

[65] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, 2018, pp. 293–305.

[66] D. S. Moore and S. Kirkland, *The Basic Practice of Statistics*, vol. 2. New York, NY, USA: WH Freeman, 2007.

[67] H. Levene, "Robust tests for equality of variances," *Contributions Probability Statist. Essays Honor Harold Hotelling*, pp. 279–292, 1961.
[68] B. L. Welch, "The generalization of 'STUDENT'S' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.
[69] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 161–165.
[70] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372
[71] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.