An Efficient Short-Time Discrete Cosine Transform and Attentive MultiResUNet Framework for Music Source Separation

THOMAS SGOUROS, (Member, IEEE), ANGELOS BOUSIS, (Member, IEEE), AND NIKOLAOS MITIANOUDIS, (Senior Member, IEEE)

Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

Corresponding author: Nikolaos Mitianoudis (nmitiano@ee.duth.gr)

ABSTRACT The music source separation problem, where the task at hand is to estimate the audio components that are present in a mixture, has been at the centre of research activity for a long time. In more recent frameworks, the problem is tackled by creating deep learning models, which attempt to extract information from each component by using Short-Time Fourier Transform (STFT) spectrograms as input. Most approaches assume that one source is present at each time-frequency point, which allows to allocate this point from the mixture to the desired source. Since this assumption is strong and is reported not to hold in practice, there is a problem that arises from the use of the magnitude of the STFT as input to these networks, which is the absence of the Fourier phase information during the separated source reconstruction. The recovery of the Fourier phase information is neither easily tractable, nor computationally efficient to estimate. In this paper, we propose a novel Attentive MultiResUNet architecture, that uses real-valued Short-Time Discrete Cosine Transform data as inputs. This step avoids the phase recovery problem, by estimating the appropriate values within the network itself, rather than employing complex estimation or post-processing algorithms. The proposed novel network features a U-Net type structure with residual skip connections and an attention mechanism that correlates the skip connection and the decoder output at the previous level. The proposed network is used for the first time in source separation and is more computationally efficient than state-of-the-art separation networks and features favourable performance compared to the state-of-the-art with a fraction of the computational cost.

INDEX TERMS Music source separation, short-time DCT, deep learning, phase recovery, MultiResUNet, attention module.

I. INTRODUCTION Music production is achieved, when recordings of individual sources (vocals and instruments) are arranged together and combined into an audio mixture. Music source separation (MSS) is the process of estimating these isolated sources, also called stems, from the audio mixture. A general mixing model can be expressed as follows:

\[ x(n) = As(n) \]  

where \( A \) represents a \( K \times L \) mixing matrix and \( n \) is the sample index of \( N \) available data values. The general under-determined source separation problem (where \( K \ll L \)) is impossible to solve, since there is an infinite number of solutions. Several previous research attempts have offered methods to trace those solutions that are relevant to source separation. For more information on the identifiability and uniqueness of solutions to the underdetermined problem, one can refer to [1] and [2]. In the case of MSS, the problem can be expressed as follows:

\[ x(n) = As(n) \]
features a two-channel (stereo $K = 2$) mixture that contains $L \gg 2$ audio sources. Over the last years, the music source separation community, through the MUSDB18 [3] and DSD100 [4] datasets and corresponding competitions, has focused on a more feasible song stem decomposition, i.e. decompose modern pop/rock music stereo songs into four basic stems: vocals, bass, drums, other (a stem containing all the remaining instruments). This is a more viable task, since it is possible to extract these four stems from most modern mixtures, even in the case of monophonic mixtures.

In order to tackle this problem and estimate the sources $s$, many approaches have been proposed, which can be divided into two different categories. The first group includes the signal processing based methods, where by exploiting possible cues, such as the sparse statistical profile of the source signals and their orientation in the space, observed by the mixture signals, they attempt to estimate the mixing matrix $A$ and consequently extract the source signals. Methods belonging to the first category span from Non-negative Matrix Factorization (NMF) [5], Independent Subspace Analysis (ISA) [6], Independent Component Analysis [7], Directional Clustering [8], [9], Bayesian modeling [10] to Robust Principal Component Analysis (PCA) [11]. For more traditional approaches, one can refer to [12] and [1]. The second group contains the deep learning frameworks, where deep neural networks have been employed as an alternative to the previously dominant audio source separation methods. Some of the most recent and state-of-the-art networks for this task include the CWS-PResUNet [13], the KUIELab-MDX-Net [14], the D3NET [15], the DEMUCS [16] and the hybrid DEMUCS [17], while there are multiple more different implementations. In this paper, our discussion will focus only on frameworks that belong to the second category, since the presented approach is based on deep learning.

Modern deep learning networks follow different steps for tackling the MSS problem. Among the first applications of deep convolutive networks in source separation, Simpson et al. attempted to separate either all sources [18] or vocals to create karaoke audio [19]. In [20], Huang et al. introduced the use of deep recurrent networks to solve the separation problem. In [21], Uhlich et al. introduced a combination of deep dense layers along with bidirectional LSTM layers to achieve separation. Some models, such as the DEMUCS [16], the Conv-Tas-Net [22] and the Meta-Tas-Net [23] attempt to separate the signals in the time domain by using the mixtures waveform as input to the network. Other approaches, such as the X-UMX [24], the LASAFT [25], the D3NET [15], the CWS-PResUNet [13] and the KUIELab-MDX-Net [14], are transforming the data into sparse representations, in order to enhance the signals’ features and separate the sources more efficiently. Most approaches tend to rely on traditional transformations for this task, including the Short-Time Fourier Transform (STFT), instead of learned overcomplete dictionary approaches [26] that offer more sparsity. The advantage is that there are fast algorithms for calculating the traditional transformations and their properties are well documented. Additionally, there are models, which address the problem in both the time and the time-frequency domain, as performed by the Hybrid DEMUCS [17]. Since the mixing is considered linear and the transformation performed to the observed signals are linear, it is mathematically equivalent to perform separation either in the time-domain or in these transform domains.

In [13], [14], [15], [24], [25], [27], and [17], the signal is transformed into a spectrogram using the Short Time Fourier Transform (STFT). The spectrogram contains complex values from which we extract the magnitude and the phase of the signal. The magnitude is then used as input for training the networks and separating the sources. Most MSS approaches assume a time-frequency mutual exclusion between sources in the STFT domain. This implies that in the STFT domain, only one source is dominant at each time-frequency point. This property is theoretically supported in the work by Liutkus et al. [28], assuming smoothness, local stationarity or periodicity for the time-frequency representation of the sources. In this case, mutually exclusive masks for each source are estimated and are applied to the mixture to create the STFT representation of each separated source. The problem appears, when this assumption ceases to hold formally, which, unfortunately, seems to be the case in real-world song mixtures, where there is time-frequency overlap between the requested stems of a song. In addition, this problem becomes a phase recovery problem, since many approaches relax the mutual exclusiveness of the magnitude of the spectrogram, allowing it to take arbitrary values after separation, but instead continue to use the phase information of the mixture. This problem has been documented in the works of Magron et al. [29], Magron, Drossos et al. [30], Stoller et al. [31], where solutions are discussed especially for the phase recovery problem. In [17], the use of a temporal and a spectral branch in combination provide important information to the network for performing efficient separation. A similar approach is applied in [14], since the signals’ time waveform enhances the network performance. On the other hand, in [13] the phase information is extracted after processing the spectrogram’s magnitude in the employed network, while the LASAFT network in [25] decomposes the complex data to their real and imaginary values and transforms their stereo spectrograms into a 4-channel input, so that no information is lost and thus the phase information is estimated by the algorithm. Unfortunately, such kind of processing leads to increased complexity networks without significant performance improvement.

In this paper, we propose to use an alternative transform, in order to avoid the phase recovery issue. The use of Short-Time Discrete Cosine Transform (STDCT) is proposed, instead of the commonly used STFT. The main motivation is that the STDCT is equally sparse and linear, but most importantly it is a real-valued transform. Thus, the transform values can be directly presented to the network as input, and the network can infer the real values of the transformed separated sources. Then, the return to the time-domain can be achieved.
without any further post-processing. In addition, we introduce a novel MultiResUNet architecture with attention modules for stereo audio mixtures. To increase performance, we train $L$ separate Attentive MultiResUNets, one for each desired component. The proposed architecture is based on a similar architecture, presented in [32], i.e. a fully convolutional layered network, which is used for Biomedical Image Segmentation. This architecture is based on the original U-Net [33], but employs residual blocks that connect similar levels of the encoder and the decoder, making the network more robust and capable of analysing objects at different scales. This network was adapted to fit the source separation task. In addition, an attention module is incorporated at the end of the residual skip connection path that connects the same level encoder and decoder layers. To the best of our knowledge, this is the first application of a multi-resolution U-Net with residual skip connections and attention on audio signal processing and source separation. Its major advantage is the considerably decreased computational cost, compared to other state-of-the-art source separation networks, while featuring performance that ranks behind only far more complicated networks. The outline of the proposed network is depicted in Fig. 1.

The paper is organised as follows. At first, the details behind the data pre-processing are presented with the introduction of the STDCT and its importance over STFT-based spectrograms. Next, the architecture of the proposed network is described in detail, with emphasis on the proposed “multires block”, the “res path” and the attention modules and their customization for music source separation. The post-training steps that were introduced in order to complete the source separation tasks are also presented in detail. Finally, the performance of the proposed framework in MSS is then investigated and compared with state-of-the-art approaches on the MUSDB18 dataset [3], with promising results. A detailed ablation study is also presented to validate the chosen hyper-parameters and modules of the architecture.

II. PROPOSED METHOD

The task here is to separate a stereo mixture signal $x_s(n) = [x_1(n), x_2(n)]^T$ and estimate the $L$ participant source signals $s_l(n) = [s_{l1}(n), s_{l2}(n), \ldots, s_{lL}(n)]^T$. Following most current literature, we also use the MUSDB18 dataset [3] in our experiments, which contains a collection of modern stereo songs, that are separated into $L = 4$ components, i.e. vocals, bass, drums and other. The other category includes anything that does not belong into the first three categories.

A. PREPROCESSING

The first step was to prepare the dataset for training the network. The MUSDB18 tracks were downsampled to 16 kHz and segmented into $m$-second patches. These stereo patches were transformed using either the STFT or the STDCT. To avoid possible overfitting, no augmentation was used, apart from the random choice of $m$-second patches for every
batch and training step. In [34], [35], and [36] various augmentation techniques were proposed, however, we preferred to keep the training framework as simple as possible.

### B. DATA TRANSFORMATION

In recent years, as mentioned in Section I, most researchers have been using the STFT as a transform. By using the STFT, the mixing model in (1) is transformed to \( X(t, f) = A(f)S(t, f) \), where \( t \) is the time frame index and \( f \) represents the normalised frequency, \( X(t, f) = [X_1(t, f), \ldots , X_K(t, f)]^T \) and \( S(t, f) = [S_1(t, f), \ldots , S_L(t, f)]^T \). Assuming that the mixing is stationary and instantaneous (as in (1)) and since STFT is a linear transformation, it follows that \( A(f) = A \), where \( A \) is the mixing matrix of (1). The main idea behind the STFT is to divide a signal to shorter overlapping segments of equal length, apply the Fourier transform to each of these and place them in a 2D fashion to create a 2D image, i.e. time-frequency spectrogram. The resulting spectrogram contains complex values, which are computationally expensive for neural networks to process. Therefore, the magnitude of this 2D image is commonly used for processing and separation. After separation, the separated output must be transformed back to the time domain using the inverse STFT. However, in order to complete this task, this process requires the separated sources’ phase, a piece of information that is unknown, since only the magnitude was estimated. As mentioned in the previous section, there are many methods to estimate the phase of each individual source. All these methods can invert the signal back to the time domain, but this process is complex, computationally expensive and the offered improvement is not always noteworthy.

For this reason, we propose an alternative to the STFT, which is the **Short-Time Discrete Cosine Transform (STDCT)**. The STDCT follows the same mechanism as the STFT, but uses the Discrete Cosine Transform, instead of the Fourier Transform. More specifically, the audio signal is segmented into short overlapping segments of equal duration. Each of these frames is windowed and the 1D-DCT is applied on each frame. The DCT type-II is used, thus assuming an 2N\(_1\) segmenting sources using real values and produce real-valued DCT “spectrograms” as source estimates.

In addition, the real-valued DCT “spectrogram” source estimate can be inverted directly to the time domain, since no phase recovery post-processing is necessary. Each column is transformed to the time domain using the 1D-iDCT. For the inverse transformation, DCT type-III is employed, which is defined as follows [37]:

\[
\begin{align*}
    s(n) &= \sqrt{\frac{2}{N_1}} \sum_{k=0}^{N_1-1} \beta(k) S(k) \cos \left( \frac{\pi k(2n+1)}{2N_1} \right), \\
    &\forall n \in [0, N_1 - 1]
\end{align*}
\]

where \( \beta(k) \) is given by (3). Once the segments are inverted to the time domain, reconstruction of the complete audio waveform is performed using the overlap-and-add (OLA) method, in a similar manner to STFT reconstruction. Perfect reconstruction using the STDCT was investigated in the past [38] and is guaranteed via careful window selection, similarly again to the STFT. In our system, the Hamming window was selected after an exhaustive ablation study over a number of possible windows, which is not included in the paper, due to limited interest. Overall, this yields a more elegant and computationally efficient solution, compared to the aforementioned STFT-based approaches.

In [14], the authors suggest that their network’s training is more efficient, provided a long frame window is chosen and only the lower frequencies are retained as input to the network. This results to a better separation scheme, which is reasonable, since the most significant information of most source components resides in the lower frequencies. Therefore, this process aids at zooming only in the essential information of the mixture and thus forcing the network to recognise more efficiently the features that belong to each component. The only exception to this behaviour are the drums, where all frequencies contain information of equal importance. Consequently, we have included this strategy in our framework, by choosing different frame windows, in the same concept as in [14], for every component and keeping only the significant lower frequencies, apart from the drums.

### C. PROPOSED NETWORK ARCHITECTURE

The network we present in this study is a two-part architecture that follows the general U-Net configuration, containing an encoder and a decoder [33]. Each level of the encoder concatenates its output to the other input of the decoder at the same level, i.e. skip connections. In a similar manner to [32], we incorporate two components, known as **MultiRes blocks** and **Res paths** in the general U-Net architecture (see Fig. 1).
1) MultiRes BLOCKS

The MultiRes blocks consist of three different groups of $3 \times 3$ Convolutional blocks with a gradually increasing number of filters $F$. In every group, we have an increasing number of Convolutional blocks from 1 to 3 with each block containing a $3 \times 3$ Convolutional layer, followed by a Rectified Linear Unit (ReLU). The $F$ metric, as proposed in [32], is used to create a connection between our model and the original U-Net [33] and is estimated as follows:

$$F = \gamma \times P$$

where $P$ is the number of filters at the corresponding layer of the U-Net and $\gamma$ is a scaling value. As suggested in [32], we assigned $F/6$, $F/3$ and $F/2$ filters to the Convolutional blocks of each respective group. Finally, there is a residual connection with a $1 \times 1$ Convolutional layer, which is able to contribute some additional spatial information, thus the network is able to learn the data features more efficiently. By gradually increasing the number of filters there is a compromise between heavy memory operations and the quality of feature extraction, therefore we are able to use larger size data inputs and acquire better audio quality. The architecture of the MultiRes blocks is shown in Figure 2, along with the size of each filter.

![Proposed MultiRes block component.](image)

2) RES BLOCKS

The Res path, on the other hand, is a shortcut between the encoder and the decoder, similar to U-Net’s skip connections [33]. In [32], Ibtehaz and Rahman argued that there is a way to improve the U-Net’s skip connections by incorporating some Convolutional layers along these connections and fill a possible gap between the encoder’s and the decoder’s extracted features. Thus, the Res path is formed as a chain of Convolutional layers, which have residual connections, as shown in Figure 3. The size of the filters are also shown in Figure 3. Using this path, the feature maps from the encoder are transferred to the decoder. There, they can be concatenated with the decoder’s features, since they have the same size. The Res path assists the network in extracting improved features, since the information is more accurate, leading to better results. This has been documented for image processing related tasks, however, in this study, we confirm that this is the case for audio-related spectrogram images.

3) ATTENTION BLOCKS

A major upgrade we propose here is the introduction of a self-attention mechanism [39], [40] that is incorporated at the end of the residual convolutional layers that connect each level of the encoder with the corresponding level of the decoder. This module has the ability to preserve the key features of the target source, while suppressing the features of the other components. It receives as input the output of the residual path of the corresponding level of the encoder $x_r$ and the decoder output of the previous level $x_d$ (see Fig. 1).

The attention module then (see Fig. 4) adds the two inputs after they have passed a $1 \times 1$ convolutional layer $W_r$, $b_r$ and $W_d$, $b_d$ respectively. The output passes through a ReLU, then processed with a $1 \times 1$ convolutional layer $W_{a1}$, $b_{a1}$ and then passes through a Softmax activation function, to discern the features belonging to the target source from the irrelevant ones, before entering another $1 \times 1$ Convolutional layer $W_{a2}$, $b_{a2}$. The product of this process form the attention coefficients, which are multiplied by the output of the residual path and then processed by another convolutional layer $W_{a3}$, $b_{a3}$ and a ReLU activation function, in order to form the final response that proceeds to the next layer. The proposed attention module was inspired by [40], but is now modified with additional convolutional layers and without the Sigmoid non-linearities, in order to accommodate the non-negative data. The attention coefficients $a_i^t$ for each pixel $i$ are given by

$$x_i^t = a_{a1}^t(\text{ReLU}(W_{a1}^T x_i^t + b_{a1} + W_d^T x_d^t + b_d)) + b_{a1}$$  

$$a_i^t = W_{a2}^T \text{Softmax}(x_i^t) + b_{a2}$$  

The output is given by the hadamard product between the residual path and the attention image.

$$y_i^t = \text{ReLU}(W_{a3}^T (a_i^t x_i^t) + b_{a3})$$

4) ARCHITECTURE SUMMARY

As presented in detail in Figure 1, the encoder is structured by chaining a repeated pattern of MultiRes blocks and $2 \times 2$ Max Pooling layers, which decrease the data size by half each time and keep the number of channels intact. The Res paths are placed prior to the pooling operation of the encoder in order to transfer vital information to the decoder, without being summarised by the pooling operation. On the other hand, the decoder uses a symmetrical cascade of MultiRes Blocks followed by $2 \times 2$ Transposed Convolutional layers to upsample the feature maps by 2 and reduce the channels by half. Each MultiRes Block of the decoder receives features from the previous level, as well as features from the Attention module. The only exception is the last MultiRes Block prior to the output, which receives the output of the previous level.
and the first Res skip connection. The output follows a final 2D Convolutional layer and yields the separated output real STDCT “spectrogram” image. In total, four levels of MultiRes blocks and Res paths with Attention are used in our implementation.

Another novel element is that all convolutional layers in the network are activated by the ELU activation function [41]. In addition, batch normalisation is used in these layers. Another major difference with [32] is the fact that we replaced the Sigmoid activation function of the output layer with the Linear function, since we need the output values to be real, i.e. contain negative values as well.

Finally, the loss function that was used for the training process was the Mean Square Error (MSE), instead of the commonly used cross-entropy, since the objective is to perform regression, i.e. infer real numbers, and not perform classification. It is important to stress that the network delivers as output the mono STDCT representation of one desired stem. In other words, a different network is trained independently for each of the desired stems. The network can be trained to infer stereo outputs of the source with similar performance, however, this was not included in this study.

D. POST-PROCESSING

Since the task at hand was to extract the $L$ participant sources, we trained different networks, one for each component. Therefore, we created 4 different networks, with the architecture shown in Figure 1, in order to predict the required sources, i.e. vocals, bass, drums, other. The output of each network is in the transform domain, thus it needs to be inverted back to the time domain to be audible. To enhance the separation quality of the separated source, we employed the following three steps:

1) SIGNAL ENERGY THRESHOLDING

Brunner et. al [42] claimed that listening to separated audio sources, provides a direct indication of when an audio source should be suppressed. Essentially, the residues of separation from other sources can be suppressed by applying a binary mask to each estimated audio source, with the appropriate signal energy threshold. Since the proposed approach is not using masks on the separated outputs or the mixture signal, there is bound to be some residual energy in non-relevant time-frequency points, which will contaminate the separated sources. To alleviate this, we applied a signal energy thresholding technique with a different approach to [42]. More specifically, we checked the energy $E_i(j)$ of each of the output’s data frames estimated as:

$$E_i(j) = \sum_{j=0}^{N-1} \hat{S}_i(j)^2$$  \hspace{1cm} (9)

If the energy of the frame is less than the applied threshold $t_i$, we set the whole frame equal to zero:

$$E_i(j) < t_i \Rightarrow \hat{S}_i(j) = 0$$  \hspace{1cm} (10)

In a similar manner to [42], we set the threshold value at $t_i = 10^{-3}$ for vocals, bass and drums and $t_i = 10^{-4}$ for the other stem. We proceed by sliding one frame at a time until the end of the data. This method helped us to not only successfully suppress noisy parts of the estimated signals,
containing residues of other stems, but also avoid rough transitions between low energy audio signal segments and high energy audio signal segments and as a result produce smooth and clean (as possible) audio signal outputs.

2) SOURCES RECONSTRUCTION TO TIME-DOMAIN
The output of the network is a 2D representation of each desired stem, that contains real values (both positive and negative). The reconstruction of each separated source can be generally described as follows:

$$\hat{s}_i(t) = iSTDCT(\hat{S}_i(f,t))$$

More specifically, each column of the 2D representation is inverted to the time-domain using (4). The resulting frames are used to reconstruct the time-domain version of the stem using overlap-and-add (OLA). It is worth noting that the goal of the network is to produce monophonic versions of the original stems. Nonetheless, it can be amended to create stereo versions, if required.

3) LOW-PASS FILTERING FOR HIGH FREQUENCY ARTIFACT COMPONENTS SUPPRESSION
The reconstructed signals, especially the bass and the other stems, need further manipulation in order to suppress present high-frequency artifact residues. To further remove high-pass noise from the separated signals, we applied 5th order Butterworth low-pass filtering on all components, as a post-processing step. Low-pass filtering is not applied on drums, since percussive organs feature important energy levels over all audible frequencies. In this work, a cutoff frequency of $f_c = 5$kHz, for the vocals and other, and a cutoff frequency of $f_c = 1$kHz, for the bass, were chosen. Therefore, no significant degradation was observed, compared to the other sources.

III. EXPERIMENTS
A. DATASET
The proposed framework was trained and tested on the MUSDB18 dataset [3], which contains 150 full-length stereo audio tracks of different genres, sampled at 44100kHz. Each track consists of 4 different stems (i.e. vocals, bass, drums and other) with every stem included separately in the dataset. Out of the 150 tracks, 100 are used for training and 50 for testing. Furthermore, the training set has a pre-defined split that separates it into 86 tracks for training and 14 tracks for validating the network. In order to train the proposed network all signals were downsampled at 16 kHz and no additional training data were involved in the process.

B. EVALUATION BENCHMARK
The performance of the proposed framework was evaluated by estimating the Signal-to-Distortion Ratio (SDR), developed by Vincent et al. [43]. For this process, the 50 tracks of the testing set were employed after being divided into 1-second windows. The separation process was assessed by estimating the median SDR of all training segments, as suggested in [44].

C. TRAINING PROCESS
The dataset preparation was a crucial part for the training procedure. Each stereo mixture song, along with the selected target source, was divided into $m = 4$ seconds pieces. As suggested in [14], depending on the target source, the data were transformed by the STDCT with different frame lengths. Therefore, for each source, i.e. vocals, bass, drums and other, the corresponding frame length of $W = [3072, 4096, 2048, 4096]$ was selected with a hop size of $H = 12.5\%$. Despite the different frame lengths, which lead to different DCT lengths, only the first 2048 DCT bins are retained and presented as input to the network. The remaining high frequency bins are dropped, resulting into a form of low-pass filtering. Thus, the input data size was always $2048 \times 128 \times 2$. Afterwards, the data were shuffled and put together with $batch_{size} = 6$. The size of the network’s output is $2048 \times 128$. The remaining bins are completed with zeros in order, to reach the selected DCT length for each source.

Concerning the network configurations, the encoder consists of $P = [32, 64, 128, 256, 512]$ filters while the decoder has the inverse sequence, i.e. $P' = [512, 256, 128, 64, 32]$ filters. The filters in the Attention module follow the structure of the intermediate layers of the decoder with $P_A = [256, 128, 64]$. To estimate the number of filters $F$, shown in (5) for the MultiRes blocks, we selected the scaler coefficient $\gamma$ to be 1.75. This value maximizes the network’s capacity and prevents overfitting and underfitting. For the separation task, a different model was trained for each target source, thus $L = 4$ identical models were implemented and they were all trained with the Adam optimizer [45] and a learning rate of 0.0005.

Finally, the training epoch is defined by selecting all data batches in sequence, presenting them to the network and comparing the output with the desired target source, by using the Mean Square Error (MSE) as loss function. After each epoch the data are shuffled again to avoid overfitting.

Before estimating the performance of the separation, a 5th order Butterworth low-pass filter was applied to all the sources except for the drums, where no significant degradation was observed compared to the others, as mentioned in Section II-D. This step was applied in the time domain for further high-pass denoising, after inverting the output with the iSTDCT.

All the models were trained on a computer running Ubuntu 20.04 with a NVIDIA GeForce 3090 GPU of 24GB, an Intel i9-11900F and 64GB RAM memory. The networks were implemented in Tensorflow v2.5. The networks were trained between 20 – 25 epochs each, depending on their validation status, i.e. each training automatically stopped after 5 consecutive epochs without improvement in performance.

1The developed python code is publicly available at https://github.com/tsgouros09/
D. PRELIMINARY COMPARISON BETWEEN STDCT AND STFT

This section presents a preliminary study to inspect the properties of the STFT and the STDCT domains in the source separation problem. It is important to mention that in the STDCT domain, the term phase information denotes the sign of each data point. For this reason, two experiments were conducted, in order to compare the properties of the two transforms in the source separation problem and more specifically, the resilience of the phase information, in the two transforms, to interference from other sources. The same parameters are used for the two transforms, i.e. frame size \( W = 2048 \), hop size \( H = 12.5\% \) and a Hamming window. In the first experiment, the aim was to evaluate the resilience of the phase, in the two transforms, to interference from other sources. More specifically, each ground-truth stem was moved to the transform-domain along with the corresponding magnitude of the source. The magnitude of the source was kept intact, but the original phase was replaced by the mixture phase. The steps for this procedure are outlined below:

1) Transform the mixture and the corresponding ground-truth target stem with the STFT or the STDCT.
2) Combine the phase information of the mixture and the magnitude information of the ground-truth target stem.
3) Invert the produced signal back to the time domain using the corresponding inverse transform.

The results of this process are shown in Table 1. The interference in the phase seems to affect more the STFT than the STDCT. The average SDR performance of 7.92dB for STFT-based source separation frameworks is much lower compared to the performance of STDCT-based frameworks with an average SDR of 11.81dB. This might be due to the fact that the phase information in the STDCT is a sign, instead of a real number in the case of the STFT.

The second experiment is to replicate a scenario closer to the source separation problem. Here, Ideal Ratio Masks (IRM) [46], extracted from the ground-truth stems, are employed as oracle masks that are applied on the song mixture to separate each stem. The two transforms are used in comparison for the oracle mask estimation and separation. This scenario uses the assumption of time-frequency exclusion between stems in their time-frequency representation (STFT or STDCT). The IRM mask was selected, since it is transparent to the transform choice. The Multichannel Wiener Filter (MWF) [47], [48] oracle mask, which is commonly used by the yearly stem unmixing contest [49], was not selected in this experiment, since the STDCT cannot translate convolution directly into multiplication [50] and thus the derived Wiener solution in [47] and [48] cannot be applied directly. The results using IRM oracle masks for the two transforms are shown in Table 2. The performance cap imposed by the mutual exclusion assumption appears as well with the STDCT, but this is extended by approximately 1.7 dB. Consequently, the STDCT seems a more efficient framework for the source separation problem.

| TABLE 1. Average median SDR (dB) performance to investigate the resilience of the STFT and STDCT transforms to phase interference. |
|-----------------|---------|---------|---------|-------|-------|
|                  | Vocals  | Bass    | Drums   | Other | Average |
| STFT             | 10.18   | 6.05    | 8.85    | 6.61  | 7.92   |
| STDCT            | 13.62   | 10.28   | 12.06   | 11.26 | 11.81  |

The two experiments have demonstrated the superiority of the STDCT in two cases important for source separation. This was the initial guide to start investigating the use of the STDCT transform, instead of the STFT, in deep-learning based source separation frameworks. The performance cap that still exists, even in the STDCT case, if we use the time-frequency mutual exclusion for the stems (Table 2) and in the case that phase, containing interference, is used during reconstruction (Table 1) has motivated us to adapt the network to use negative values as inputs and not use mutually exclusive masks for separation. Consequently, the network was forced to infer the sign of the STDCT, yielding a direct solution to this problem. In addition, the network is forced to un-tangle possible overlapping of the stems in several time-frequency points, thus relaxing the initial assumption. Possible energy residual in some time-frequency points is removed by thresholding after separation to filter out musical noise. The proposed solution offers a straightforward architecture that requires no sophisticated post-processing steps for phase estimation of the stems.

E. NETWORK PERFORMANCE

As mentioned in Section III-B, the SDR metric was employed to measure the proposed Attentive MultiResUNet's performance. The results are shown in Table 3, where the framework was compared to the performance of the networks presented in [13], [14], [15], [16], [17], [22], [23], [25], and [24], when tested on the MUSDB18 dataset. Audio samples from the separated output stems by the proposed MultiResUNet, along with their ground truth can be found here.2

The second experiment is to replicate a scenario closer to the source separation problem. Here, Ideal Ratio Masks (IRM) [46], extracted from the ground-truth stems, are employed as oracle masks that are applied on the song mixture to separate each stem. The two transforms are used in comparison for the oracle mask estimation and separation. This scenario uses the assumption of time-frequency exclusion between stems in their time-frequency representation (STFT or STDCT). The IRM mask was selected, since it is transparent to the transform choice. The Multichannel Wiener Filter (MWF) [47], [48] oracle mask, which is commonly used by the yearly stem unmixing contest [49], was not selected in this experiment, since the STDCT cannot translate convolution directly into multiplication [50] and thus the derived Wiener solution in [47] and [48] cannot be applied directly. The results using IRM oracle masks for the two transforms are shown in Table 2. The performance cap imposed by the mutual exclusion assumption appears as well with the STDCT, but this is extended by approximately 1.7 dB. Consequently, the STDCT seems a more efficient framework for the source separation problem.

| TABLE 2. Average median SDR (dB) performance comparison between the STFT and STDCT transforms in a source separation scenario using IRM oracle masks. |
|-----------------|---------|---------|---------|-------|
|                 | Vocals  | Bass    | Drums   | Other |
| STFT            | 11.18   | 7.35    | 9.8     | 8.05  |
| STDCT           | 12.78   | 9.48    | 11.5    | 9.78  |

The two experiments have demonstrated the superiority of the STDCT in two cases important for source separation. This was the initial guide to start investigating the use of the STDCT transform, instead of the STFT, in deep-learning based source separation frameworks. The performance cap that still exists, even in the STDCT case, if we use the time-frequency mutual exclusion for the stems (Table 2) and in the case that phase, containing interference, is used during reconstruction (Table 1) has motivated us to adapt the network to use negative values as inputs and not use mutually exclusive masks for separation. Consequently, the network was forced to infer the sign of the STDCT, yielding a direct solution to this problem. In addition, the network is forced to un-tangle possible overlapping of the stems in several time-frequency points, thus relaxing the initial assumption. Possible energy residual in some time-frequency points is removed by thresholding after separation to filter out musical noise. The proposed solution offers a straightforward architecture that requires no sophisticated post-processing steps for phase estimation of the stems.

2https://tsgouros09.github.io/Attentive-MultiResUNet/
TABLE 3. Average median SDR (dB) performance in dB of our framework compared to the state-of-the-art networks and their model parameters in millions.

| Method            | Vocals | Bass  | Drums | Other | Average | Params  |
|-------------------|--------|-------|-------|-------|---------|---------|
| Hybrid Demucs [17]| 8.13   | 8.76  | 8.24  | 5.59  | 7.68    | 83.6M   |
| KUJELab-MDX-Net [14]| 9.00  | 7.86  | 7.33  | 5.95  | 7.54    | 141.2M  |
| Attentive MultiResUNet | 8.37  | 5.88  | 7.63  | 5.14  | 6.81    | 8.6M    |
| CWS-PresUNet [13] | 8.92   | 5.93  | 6.38  | 5.84  | 6.77    | 208M    |
| DEMUCS [16]       | 6.84   | 7.01  | 6.86  | 4.42  | 6.28    | 133.8M  |
| DjNet [15]        | 7.24   | 5.25  | 7.01  | 4.33  | 6.01    | 3M      |
| LASAFT [25]       | 7.33   | 5.63  | 5.68  | 4.87  | 5.88    | 8.5M    |
| X-UMX [24]        | 6.61   | 5.43  | 6.47  | 4.64  | 5.79    | 9.5M    |
| Conv-Tas-Net [22] | 6.81   | 5.66  | 6.08  | 4.37  | 5.73    | 5M      |
| Meta-Tas-Net [23] | 6.40   | 5.58  | 5.91  | 4.19  | 5.52    | 45.4M   |

TABLE 4. SDR (dB) results using the STFT and the STDCT in the proposed network.

| Method            | STFT  | STDCT |
|-------------------|-------|-------|
| Hybrid Demucs     | 6.78  |       |
| KUJELab-MDX-Net   |       | 8.27  |

FIGURE 5. SDR performance vs model parameters.

size, with more than 80 million model parameters. This size is justified by the complexity of their separation schemes, however, their need for computational power and their time consuming training, may discourage their casual use. On the other hand, the proposed Attentive MultiResUNet is a much smaller network (8.6 M parameters), it is easier to train, since it needs less resources, and yields noteworthy results.

Therefore, the proposed framework, even though it may not have the best performance, compared to all the state-of-the-art networks, its small size and robustness, compared to other more complex networks, shows its merits.

F. ABLATION STUDY

The configurations and parameters of the proposed framework were chosen after constant experimentation. In these experiments we observed the networks behaviour with different configurations in the following aspects: a) STFT vs STDCT, b) Loss Function choice, c) Final Loss Scheme, d) Activation Function choice, e) Optimiser choice, f) Attention module, g) Scaler Coefficient $\gamma$, h) Low-pass Filtering vs No Low-pass Filtering, i) Dropout vs No-Dropout.

For simplicity, the experiments were performed only for separating the vocals, with $W = 2048$ and $H = 12.5\%$, unless mentioned otherwise, and the results are shown in Tables 4-12.

1) STFT VS STDCT USING THE PROPOSED NETWORK

In this experiment, the aim is to compare the performance of the STFT and the STDCT in the proposed network. The stereo inputs were transformed using the STFT, using the aforementioned window length $W$ and hop size $H$. The complex stereo STFT representation was re-arranged as a 4-channel input (real-Left, imaginary-Left, real-Right, imaginary-Right) and presented as input to the proposed network for training. For the STDCT, the same window length and hop size was used, yielding a 2-channel input. Note that the STFT representation features $W/2$ elements along the frequency dimension (removing negative frequency content), whereas the STDCT has $W$ elements along the frequency dimension. This shows that the two representations have equal size tensors in terms of number of elements, but different arrangement ($1024 \times 128 \times 4$ for the STFT and $2048 \times 128 \times 2$ for the STDCT).

In this experiment, the STFT is modified to present also real numbers as input to the network, in a similar manner to the STDCT. The results in Table 4 indicate the boost in performance offered by the STDCT. This might be due to the fact that the real and imaginary parts of the same signal are considered independent inputs and the receptive fields of the first layers cannot encode their correlations and combine the information there in a more productive manner.

2) LOSS FUNCTION CHOICE

An essential part of a successful training is the loss function used for calibrating the network after each batch. The most common loss functions for the task were the Mean Square Error (MSE) and the Mean Absolute Error (MAE). Therefore, we conducted a series of experiments in order to check which yields the best results.

As shown in Table 5, the MAE fails to separate the vocals from the mixture with a median SDR of $-15.36$ dB, while the MSE succeeded with SDR of 8.27 dB. To verify the same behaviour of the MAE, we repeated the same experiment with the other sources. Performance is improved, but is inferior to MSE. Nonetheless, due to the very low performance at vocals, the MAE was excluded from our implementation.
3) FINAL LOSS SCHEME

After selecting the appropriate loss function, we investigated on the formulation of the final loss scheme. The proposed network is trained in the transform domain, but it was necessary to conclude whether the transform or the time domain would be selected for calibrating the network, before conducting the rest of the experiments. Thus, we conducted 3 different experiments:

1) Training using the MSE in the transform domain (Loss 1)
2) Training using the MSE in the time domain (Loss 2)
3) Training with a combination of MSE loss in the transform domain and MSE loss in the time domain (Loss 3)

As presented in Table 6, the network underperforms using Loss 2. As expected, it was more difficult for the network to separate the mixture in the time domain. On the other hand, the network in the third case, where we combine the two loss schemes, the separation quality is 7.76 dB in terms of SDR, however, it is the first case, where the loss function is applied only in the transform domain, that yields the best results. Therefore, the first choice was to use the transform domain loss, i.e. Loss 1.

4) ACTIVATION FUNCTION

Another important configuration for training a network is the activation function used in its layers. Usually, MSS deep networks employ the ReLU activation function, however there are more advanced functions, which might improve network’s performance. Therefore, we experimented with four different activation functions, the ReLU, the ELU, the GELU [51] and the SWISH [52] and the results are shown in Table 7. Out of the three, ELU scored the best results in the evaluation, with SDR of 8.27 dB, followed by GELU, with SDR of 8.10 dB, SWISH, with SDR of 7.91 dB and ReLU, with SDR of 7.83. The ability of ELU, SWISH and GELU to allow negative values seems to boost the overall performance of the network, but ELU shows a solid improvement compared to ReLU, therefore, it was included in the final framework.

5) OPTIMISER

Generally, when training a network for MSS, the usual optimiser for the task is Adam, however, there are cases, like in [14], where it is suggested to use the RMSProp optimiser [53]. Furthermore, the past few years, new optimisers, such as the AdaBelief [54], have shown promising results in deep learning, thus, it was crucial to test these available options for the proposed network. As shown by the results in Table 8, AdaBelief does not perform well in the proposed framework, with an SDR of 5.96 dB, and the RMSProp underperforms as well, with SDR of 7.63 dB, thus we excluded them from the final implementation.

6) ATTENTION MODULE

One of the main features of the proposed Attentive-MultiResUNet is the inclusion of the Attention module in its architecture. It was necessary for the authors to test its performance compared to a network without this module. Therefore, we conducted a series of experiments to investigate the network’s behaviour. The results in Table 9 show that the proposed attention module gives a significant boost to the primal network, with SDR of 8.27 dB, compared to a network without attention, which performs a SDR of 7.71 dB. Consequently, we incorporated the module to the final architecture.

7) SCALER COEFFICIENT $\gamma$

The scaler coefficient $\gamma$, introduced in [32], plays an essential role in the proposed framework, since it specifies the number of filters in the MultiRes Blocks, therefore, it was essential to select a value that would boost the network’s performance. The $\gamma$ values that were selected in the experiments had to keep the right balance between the number of the MultiRes blocks filters and the networks efficiency. As shown in Table 10, with $\gamma = 1.50$ the MultiRes blocks’ filters are insufficient to process the data, while with $\gamma = 2.00$ the number of the filters lead to a degrading of the network’s performance. Instead, a value of $\gamma = 1.75$ yields a more balanced Attentive MultiResUNet, giving its best performance.

8) LOW-PASS FILTERING VS NO LOW-PASS FILTERING

As mentioned in Section II-B, [14] proposes the use of reduced spectrograms for training, keeping only the lower frequencies of all components, except for the drums, where the authors discard nothing. This process zooms in the information of the transformed data and enhances the separation scheme. It is observed that most of the sources keep essential
information in the lower frequencies, hence, we observed our network’s behaviour, when keeping all the frequencies and when removing some of the higher frequencies. The experiments that were conducted in the second case used a frame length $W = 3072$ and the results are shown in Table 11. It appears that, by keeping only the lower frequencies, the separation is improved with SDR of 8.57 dB instead of 8.27 dB that was before removing the bins, therefore, this mechanism leads to a more efficient separation.

### Table 10. SDR (dB) results for different values of the scalar coefficient $\gamma$.

| $\gamma$ | 1.50 | 1.75 | 2.00 |
|----------|------|------|------|
| SDR      | 7.89 | 8.27 | 7.94 |

### Table 11. SDR (dB) results keeping all the frequency range vs keeping the lower frequencies.

|         | No LP Filtering | With LP Filtering |
|---------|----------------|-------------------|
| SDR     | 8.27           | 8.57              |

9) **DROPOUT VS NO-DROPOUT**

In the final experiment, we investigate possible benefits from the existence of dropout layers [55], which are documented to prevent deep networks from overfitting and increase its performance. As shown in Table 12, the performance in both cases is a bit similar, however, the computational cost of the dropout layers led to their exclusion from the final implementation, since each epoch’s duration was almost double in time, compared to its duration for network without dropout layers.

### Table 12. SDR (dB) results with or without using dropout layers in the network.

|         | With DropOut | No Dropout |
|---------|--------------|------------|
| SDR     | 8.16         | 8.27       |

IV. **CONCLUSION AND FUTURE WORK**

In this paper, we present a robust and efficient Attentive MultiResUNet for Music Source separation. This network is less complex than most state-of-the-art networks (using less than 10% of the runner up’s parameters) and achieves the third best performance in terms of SDR in our study. One of its most important offerings is the use of the STDCT as a transform, which helps the network avoid the phase recovery problem and separate the mixtures without further processing. The addition of an attention module seemed to boost performance. In the future, we will be looking into more complex networks to convey information from the encoder to the decoder, as well as introducing temporal attention blocks, in conjunction with spatial attention that is used in the present architecture.

**REFERENCES**

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. New York, NY, USA: Academic, 2010, p. 856.

[2] J. Eriksson and V. Koivunen, “Identifiability and separability of linear ICA models revisited,” in *Proc. Int. Workshop Independ. Compon. Anal. Blind Signal Separat.* (ICA), Nara, Japan, 2003, pp. 23–27.

[3] Z. Rafii, A. Liutkus, F.-R. Stoter, S. I. Mimalakis, and R. Bittner, “The MUSDDB18 corpus for music separation,” *Tech. Rep.*, 2017.

[4] A. Liutkus, F.-R. Stoter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, “The 2016 signal separation evaluation campaign,” in *Proc. 12th Int. Conf. Latent Variable Anal. Signal Separat.* (LVA/ICA), P. Tichavský, M. Babaie-Zadeh, O. J. J. Michel, and N. Thirion-Moreau, Eds., Liberec, Czech Republic: Cham, Switzerland: Springer, Aug. 2017, pp. 323–332.

[5] N. Q. K. Duong, A. Ozerv, L. Chevallier, and J. Sirot, “An interactive audio source separation framework based on non-negative matrix factorization,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 1567–1571.

[6] M. A. Casey and A. Westner, “Separation of mixed audio sources by independent subspace analysis,” in *Proc. ICMC*, 2000.

[7] A. Hyvärinen, “Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood,” *Neurocomputing*, vol. 22, nos. 1–3, pp. 49–67, Nov. 1998.

[8] N. Mitianoudis and T. Stathaki, “Batch and online underdetermined source separation using Laplacian mixture models,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 6, pp. 1818–1832, Aug. 2007.

[9] N. Mitianoudis, “A generalized directional Laplacian distribution: Estimation, mixture models and audio source separation,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 9, pp. 2397–2408, Nov. 2012.

[10] A. Ozerv, P. Philippe, F. Bimbot, and R. Grironval, “Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 5, pp. 1564–1578, Jul. 2007.

[11] Y. Ikeniya, K. Itayama, and K. Yoshii, “Singing voice separation and vocal F0 estimation based on mutual combination of robust principal component analysis and subharmonic summation,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 11, pp. 2084–2095, Nov. 2016.

[12] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, NY, USA: Wiley, 2001.

[13] H. Liu, Q. Kong, and J. Liu, “CWS-PResUNet: Music source separation with channel-wise subband phase-aware ResUNet,” 2021, arXiv:2112.04685.

[14] M. Kim, W. Choi, D. Lee, and S. Jung, “KUIELab-MDX-Net: A two-stream neural network for music demixing,” 2021, arXiv:2111.12203.

[15] N. Takahashi and Y. Mitsufuji, “D3Net: Densely connected multiliddent DenseNet for music source separation,” 2020, arXiv:2010.01733.

[16] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” 2019, arXiv:1911.13254.

[17] A. Défossez, “Hybrid spectrogram and waveform source separation,” 2021, arXiv:2111.03600.

[18] A. J. R. Simpson, “Probabilistic binary-mask cocktail-party source separation in a convolutional deep neural network,” 2015, arXiv:1503.06962.

[19] J. R. A. Simpson, G. Roma, and D. M. Plumbley, “Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network,” in *Proc. IVA/ICA*, 2015, pp. 429–436.

[20] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaradías, “Joint optimization of masks and deep recurrent neural networks for monaural source separation,” *IEEE/ACM Trans. Audio Speech Language Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[21] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 261–265.

[22] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[23] D. Samuel, A. Ganeshan, and J. Naradowsky, “Meta-learning extractors for music source separation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 816–820.

[24] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, “All for one and one for all: Improving music separation by bridging networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 51–55.
[25] W. Choi, M. Kim, J. Chung, and S. Jung, “LaSAFT: Latent source attentive frequency transformation for conditioned source separation,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 171–175.

[26] G. Taubock, S. Rajbamshi, and P. Baláz, “Dictionary learning for sparse audio inpainting,” IEEE J. Sel. Topics Signal Process., vol. 15, no. 1, pp. 104–119, Jan. 2021.

[27] K. W. E. Lin, B. T. Balamurali, E. Koh, S. Lui, and D. Herremans, “Singing voice separation using a deep convolutional neural network trained by ideal binary mask and cross entropy,” Neural Comput. Appl., vol. 32, no. 4, pp. 1037–1050, Feb. 2020.

[28] A. Liutkus, R. Badeau, and G. Richard, “Gaussian processes for underdetermined source separation,” IEEE Trans. Signal Process., vol. 59, no. 7, pp. 3155–3167, Jul. 2011.

[29] P. Magron, R. Badeau, and B. David, “Model-based STFT phase recovery for audio source separation,” IEEE ACM Trans. Audio Speech Language Process., vol. 26, no. 6, pp. 1091–1101, Mar. 2018.

[30] P. Magron, K. Drossos, S. L. Mimilakis, and T. Virtanen, “Reducing interference with phase recovery in DNN-based monaural singing voice separation,” in Proc. Interspeech, Sep. 2018, pp. 332–336.

[31] D. Stoller, S. Ewert, and S. Dixon, “Wave-U-Net: A multi-scale neural network for end-to-end audio source separation,” in Proc. ISMIR, E. Gómez, X. Hu, E. Humphreys, and E. Benetos, Eds., 2018, pp. 334–340.

[32] N. Itshaez and M. Sohel Rahman, “MultiResU-Net: Rethinking the U-Net architecture for multimodal biomedical image segmentation,” CoRR, vol. abs/1902.04049, pp. 74–87, Jan. 2019.

[33] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation. Cham, Switzerland: Springer, 2015, pp. 234–241.

[34] S. Uhlich, M. Porcu, F. Giron, M. Ennekil, T. Kemp, N. Takahashi, and Y. Mitsuflji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2017, pp. 261–265.

[35] J. Schlüter and T. Grill, “Exploring data augmentation for improved singing voice detection with neural networks,” in Proc. ISMIR, 2015, pp. 121–126.

[36] B. McFee, E. J. Humphrey, and J. P. Bello, “A software framework for musical data augmentation,” in Proc. ISMIR, 2015, pp. 248–254.

[37] A. V. Oppenheim and R. Schafer, Discrete-Time Signal Processing, 3rd ed. USA: Pearson, 2009, p. 1144.

[38] J. Princen, A. V. Oppenheim, and R. Schafer, “Analysis/synthesis filter bank design based on Parks’ model,” IEEE Trans. Signal Process., vol. 30, no. 8, pp. 1455–1464, Aug. 1982.

[39] J. Schlüter and T. Grill, “Exploring data augmentation for improved singing voice detection with neural networks,” in Proc. ISMIR, 2015, pp. 121–126.

[40] A. Liutkus, R. Badeau, and G. Richard, “Low bitrate informed source separation of realistic mixtures,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 66–70.

[41] Y. Mitsuflji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défoncez, M. Kim, W.-S. Choi, C.-Y. Yu, and K. W. Cheuk, “Music demixing challenge 2021,” Frontiers Signal Process., vol. 1, pp. 1–14, Jan. 2022, Art. no. 808395.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Gómez, P. Uszkoreit, and X. Lin, “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS), NY, USA, 2017, pp. 6000–6010.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

THOMAS SGOUROS (Member, IEEE) received the Diploma degree in electronic and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 2014, where he is currently pursuing the Ph.D. degree, working on audio source separation using machine learning techniques. His research interests include machine learning, deep learning, and blind source separation/extraction.

ANGELOS BOUSIS was born in Ioannina, Greece, in 1997. He received the integrated M.Sc. degree in electrical and computer engineering from the Democritus University of Thrace, in 2021. During the period of specialization in his studies, he followed the Telecommunications and Space Science Sector and currently works as a Software Developer. His interests include software engineering, audio signal processing, deep learning, pattern recognition and electronic music production.

NIKOLAOS MITIANOUDIS (Senior Member, IEEE) received the Diploma degree in electronic and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1998, the M.Sc. degree in communications and signal processing from Imperial College London, U.K., in 2000, and the Ph.D. degree in audio source separation using independent component analysis from the Queen Mary University of London, U.K., in 2004. From 2003 to 2009, he was a Research Associate at Imperial College London, working on the Data Information Fusion-Defense Technology Centre Project “Applied Multi-Dimensional Fusion,” sponsored by General Dynamics U.K., and QinetiQ. From 2009 to 2010, he was an Academic Assistant at the International Hellenic University, Greece. Since 2010, he has been with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece, where he is currently serves as an Associate Professor in audio and image processing. His research interests include machine learning, deep learning, computer vision, music information retrieval, and blind source separation/extraction. Since 2018, he has been serving as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

VOLUME 10, 2022 119459