



# Music Source Separation in the Waveform Domain

Alexandre Défossez, Nicolas Usunier, Léon Bottou, Francis Bach

## ► To cite this version:

Alexandre Défossez, Nicolas Usunier, Léon Bottou, Francis Bach. Music Source Separation in the Waveform Domain. 2021. hal-02379796v2

**HAL Id: hal-02379796**

**<https://hal.archives-ouvertes.fr/hal-02379796v2>**

Preprint submitted on 28 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Music Source Separation in the Waveform Domain

---

**Alexandre Défossez**  
Facebook AI Research  
INRIA

École Normale Supérieure  
PSL Research University  
defossez@fb.com

**Nicolas Usunier**  
Facebook AI Research  
usunier@fb.com

**Léon Bottou**  
Facebook AI Research  
leonb@fb.com

**Francis Bach**  
INRIA  
École Normale Supérieure  
PSL Research University  
francis.bach@ens.fr

## Abstract

Source separation for music is the task of isolating contributions, or *stems*, from different instruments recorded individually and arranged together to form a song. Such components include voice, bass, drums and any other accompaniments. Contrarily to many audio synthesis tasks where the best performances are achieved by models that directly generate the waveform, the state-of-the-art in source separation for music is to compute masks on the magnitude spectrum. In this paper, we compare two waveform domain architectures. We first adapt Conv-Tasnet, initially developed for speech source separation, to the task of music source separation. While Conv-Tasnet beats many existing spectrogram-domain methods, it suffers from significant artifacts, as shown by human evaluations. We propose instead Demucs, a novel waveform-to-waveform model, with a U-Net structure and bidirectional LSTM. Experiments on the MusDB dataset show that, with proper data augmentation, Demucs beats all existing state-of-the-art architectures, including Conv-Tasnet, with 6.3 SDR on average, (and up to 6.8 with 150 extra training songs, even surpassing the IRM oracle for the bass source). Using recent development in model quantization, Demucs can be compressed down to 120MB without any loss of accuracy. We also provide human evaluations, showing that Demucs benefit from a large advantage in terms of the naturalness of the audio. However, it suffers from some bleeding, especially between the vocals and other source.

## 1 Introduction

Cherry first noticed the “cocktail party effect” [Cherry, 1953]: how the human brain is able to separate a single conversation out of a surrounding noise from a room full of people chatting. Bregman later tried to understand how the brain was able to analyse a complex auditory signal and segment it into higher level streams. His framework for auditory scene analysis [Bregman, 1990] spawned its computational counterpart, trying to reproduce or model accomplishments of the brains with algorithmic means [Wang and Brown, 2006], in particular regarding source separation capabilities.

When producing music, recordings of individual instruments called *stems* are arranged together and mastered into the final song. The goal of source separation is to recover those individual stems from the mixed signal. Unlike the cocktail party problem, there is not a single source of interest to

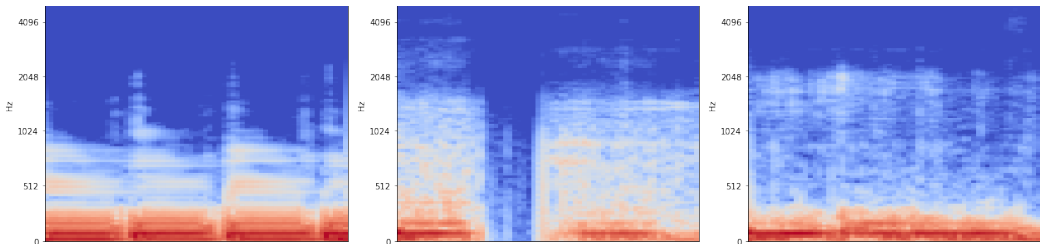


Figure 1: Mel-spectrogram for a 0.8 seconds extract of the bass source from the track “Stich Up” of the MusDB test. From left to right: ground truth, Conv-Tasnet estimate and Demucs estimate. We observe that Conv-Tasnet missed one note entirely.

differentiate from an unrelated background noise, but instead a wide variety of tones and timbres playing in a coordinated way. In the SiSec Mus evaluation campaign for music separation [Stöter et al., 2018], those individual stems were grouped into 4 broad categories: (1) drums, (2) bass, (3) other, (4) vocals. Given a music track which is a mixture of these four sources, also called the mix, the goal is to generate four waveforms that correspond to each of the original sources. We consider here the case of supervised source separation, where the training data contain music tracks (i.e., mixtures), together with the ground truth waveform for each of the sources.

State-of-the-art approaches in music source separation still operate on the spectrograms generated by the short-time Fourier transform (STFT). They produce a mask on the magnitude spectrums for each frame and each source, and the output audio is generated by running an inverse STFT on the masked spectrograms reusing the input mixture phase [Takahashi and Mitsufuji, 2017, Takahashi et al., 2018, Takahashi and Mitsufuji, 2020]. Several architectures trained end-to-end to directly synthesize the waveforms have been proposed [Lluís et al., 2018, Jansson et al., 2017], but their performances are far below the state-of-the-art: in the last SiSec Mus evaluation campaign [Stöter et al., 2018], the best model that directly predicts waveforms achieves an average signal-to-noise ratio (SDR) over all four sources of 3.2, against 5.3 for the best approach that predicts spectrogram masks (also see Table 1 in Section 6). An upper bound on the performance of all methods relying on masking spectrograms is given by the SDR obtained when using a mask computed using the ground truth sources spectrograms, for instance the Ideal Ratio Mask (IRM) or the Ideal Binary Mask (IBM) oracles. For speech source separation, Luo and Mesgarani [2019] proposed Conv-Tasnet, a model that reuses the masking approach of spectrogram methods but learns the masks jointly with a convolutional front-end, operating directly in the waveform domain for both the inputs and outputs. Conv-Tasnet surpasses both the IRM and IBM oracles.

Our first contribution is to adapt the Conv-Tasnet architecture, originally designed for monophonic speech separation and audio sampled at 8 kHz, to the task of stereophonic music source separation for audio sampled at 44.1 kHz. While Conv-Tasnet separates with a high accuracy the different sources, we observed artifacts when listening to the generated audio: a constant broadband noise, hollow instruments attacks or even missing parts. They are especially noticeable on the drums and bass sources and we give one example on Figure 1, and measure it with human evaluation in Section 6.

To overcome the limitations of Conv-Tasnet, we introduce Demucs, a novel architecture for music source separation. Similarly to Conv-Tasnet, Demucs is a deep learning model that directly operates on the raw input waveform and generates a waveform for each source. Demucs is inspired by models for music synthesis rather than masking approaches. It is a U-net architecture with a convolutional encoder and a decoder based on wide transposed convolutions with large strides inspired by recent work on music synthesis [Défossez et al., 2018]. The other critical features of the approach are a bidirectional LSTM between the encoder and the decoder, increasing the number of channels exponentially with depth, gated linear units as activation function [Dauphin et al., 2017] which also allow for masking, and a new initialization scheme.

We present experiments on the MusDB [Rafi et al., 2017] benchmark. While Conv-Tasnet outperforms several existing spectrogram-domain methods, it does suffer from large audio artifacts as measured by human evaluations. On the other hand, with proper augmentation, the Demucs architecture surpasses all existing spectrogram or waveform domain architectures in terms of SDR [Vincent et al., 2006], with 6.3 points of SDR without extra training data (against 6.0 for the best existing method D3Net),

and up to 6.8 with extra training data. In fact, for the bass source, Demucs is the first model to surpass the IRM oracle, with 7.6 SDR (against 7.1 for the IRM). We confirm the usefulness of pitch/tempo shift augmentation, as first noted by [Cohen-Hadria et al. \[2019\]](#), with a gain of 0.4 points of SDR, in particular for a model with a large number of parameters like Demucs, while it can be detrimental to Conv-TasNet.

We discuss in more detail the related work in the next Section. We then describe the original Conv-Tasnet model of [Luo and Mesgarani \[2018\]](#) and its adaptation to music source separation. Our Demucs architecture is detailed in Section 4. We present the experimental protocol in Section 5, and the experimental results compared to the state-of-the-art in Section 6. Finally, we describe the results of the human evaluation and the ablation study.

## 2 Related Work

A first category of methods for supervised music source separation work on time-frequency representations. They predict a power spectrogram for each source and reuse the phase from the input mixture to synthesise individual waveforms. Traditional methods have mostly focused on blind (unsupervised) source separation. Non-negative matrix factorization techniques [[Smaragdis et al., 2014](#)] model the power spectrum as a weighted sum of a learnt spectral dictionary, whose elements are grouped into individual sources. Independent component analysis [[Hyvärinen et al., 2004](#)] relies on independence assumptions and multiple microphones to separate the sources. Learning a soft/binary mask over power spectrograms has been done using either HMM-based prediction [[Roweis, 2001](#)] or segmentation techniques [[Bach and Jordan, 2005](#)].

With the development of deep learning, fully supervised methods have gained momentum. Initial work was performed on speech source separation [[Grais et al., 2014](#)], followed by works on music using simple fully connected networks over few spectrogram frames [[Uhlich et al., 2015](#)], LSTMs [[Uhlich et al., 2017](#)], or multi scale convolutional/recurrent networks [[Liu and Yang, 2018](#), [Takahashi and Mitsufuji, 2017](#)]. [Nugraha et al. \[2016\]](#) showed that Wiener filtering is an efficient post-processing step for spectrogram-based models and it is now used by all top performing models in this category. Those methods have performed the best during the last SiSec 2018 evaluation [[Stöter et al., 2018](#)] for source separation on the MusDB [[Rafii et al., 2017](#)] dataset. After the evaluation, a reproducible baseline called Open Unmix has been released by [Stöter et al. \[2019\]](#) and matches the top submissions trained only on MusDB. MMDenseLSTM, a model proposed by [Takahashi et al. \[2018\]](#) and trained on 807 unreleased songs currently holds the absolute record of SDR in the SiSec campaign. More recently, [Takahashi and Mitsufuji \[2020\]](#) improved the spectrogram-domain state-of-the-art with D3Net, which uses dilated convolutions with dense connection. [Hennequin et al. \[2020\]](#) propose another U-Net architecture for spectrogram masking, called Spleeter, that is trained on the Bean dataset [Prétet et al. \[2019\]](#), composed of short extracts from nearly 25,000 songs. While Spleeter provides strong performance and has been widely adopted by the digital music industry, it is now outperformed by more advanced spectrogram domain architectures like D3Net, even though it is trained on much more data.

More recently, models operating in the waveform domain have been developed, so far with worse performance than those operating in the spectrogram domain. A convolutional network with a U-Net structure called Wave-U-Net was used first on spectrograms [[Jansson et al., 2017](#)] and then adapted to the waveform domain [[Stoller et al., 2018](#)]. Wave-U-Net was submitted to the SiSec 2018 evaluation campaign with a performance inferior to that of most spectrogram domain models by a large margin. A Wavenet-inspired, although using a regression loss and not auto-regressive, was first used for speech denoising [[Rethage et al., 2018](#)] and then adapted to source separation [[Luís et al., 2018](#)]. Our model significantly outperforms Wave-U-Net. Given that the Wavenet inspired model performed worse than Wave-U-Net, we did not consider it for comparison.

In the field of monophonic speech source separation, spectrogram masking methods have enjoyed good performance [[Kolbæk et al., 2017](#), [Isik et al., 2016](#)]. [Luo and Mesgarani \[2018\]](#) developed Tasnet, a waveform domain methods using masking over a learnable front-end obtained from a LSTM that reached the same accuracy. Improvements were obtained by [Wang et al. \[2018\]](#) for spectrogram methods using the unfolding of a few iterations of a phase reconstruction algorithm in the training loss. In the mean time, [Luo and Mesgarani \[2019\]](#) refined their approach, replacing the LSTM with a superposition of dilated convolutions, which improved the SDR and definitely surpassed spectrogram

based approaches, including oracles that use the ground truth sources such as the ideal ratio mask (IRM) or the ideal binary mask (IBM). We show that, while surpassing many existing spectrogram domain methods, Conv-Tasnet does not benefit much from pitch/tempo shift augmentation, and we could only reach a SDR of 5.7 with it. Besides, its audio samples suffer from significant audio artifacts as measured by human evaluations. However, it has less contamination from other sources than Demucs. [Luo et al., 2020] introduced the Dual-Path RNN architecture for speech separation. Nachmani et al. [2020] improved on it, with strong results on the MusDB dataset (5.8 SDR). It is unclear whether it could benefit from pitch/tempo shift augmentation.

Samuel et al. [2020] also adapted the Conv-Tasnet architecture to the task of music source separation, using meta-learning (i.e., the weights for each source are generated from a meta neural network), multi-resolution separation and auxiliary losses. Their model matches the performance of our Conv-Tasnet baseline, but would likely suffer from the same limitation. Finally, Lancaster and Souviraà-Labastie [2020] trained an original Tasnet model, with 30 times more training data than in the Musdb dataset. While this model achieves strong results, it fails to be competitive when considering with models like Conv-Tasnet or Demucs, especially considering the large amount of training data used.

### 3 Adapting Conv-Tasnet for music source separation

We describe in this section the Conv-Tasnet architecture of Luo and Mesgarani [2018] and give the details of how we adapted the architecture to fit the setting of the MusDB dataset.

**Overall framework** Each source  $s$  is represented by a waveform  $x_s \in \mathbb{R}^{C,T}$  where  $C$  is the number of channels (1 for mono, 2 for stereo) and  $T$  the number of samples of the waveform. The mixture (i.e., music track) is the sum of all sources  $x := \sum_{s=1}^S x_s$ . We aim at training a model  $g$  parameterized by  $\theta$ , such that  $g(x) = (g_s(x; \theta))_{s=1}^S$ , where  $g_s(x; \theta)$  is the predicted waveform for source  $s$  given  $x$ , that minimizes

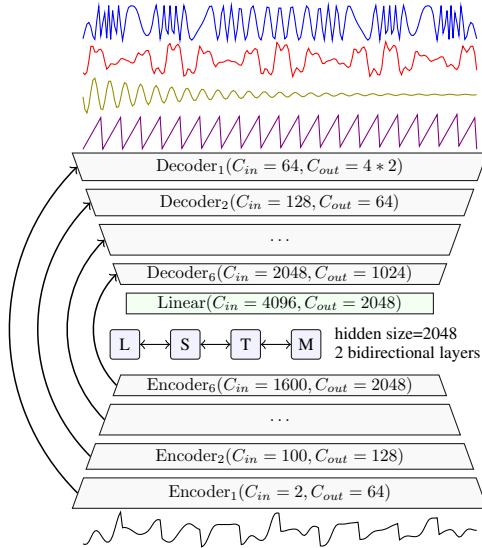
$$\min_{\theta} \sum_{x \in \mathcal{D}} \sum_{s=1}^S L(g_s(x; \theta), x_s) \quad (1)$$

for some dataset  $\mathcal{D}$  and reconstruction error  $L$ . The original Conv-Tasnet was trained using a loss called scale-invariant source-to-noise ratio (SI-SNR), similar to the SDR loss described in Section 5. We instead use a simple L1 loss between the estimated and ground truth sources. We discuss in more details regression losses in the context of our Demucs architecture in Section 4.2.

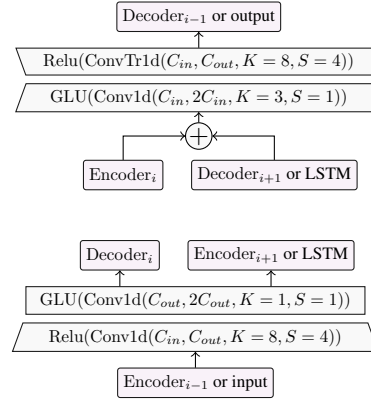
**The original Conv-Tasnet architecture** Conv-Tasnet [Luo and Mesgarani, 2018] is composed of a learnt front-end that transforms back and forth between the input monophonic mixture waveform sampled at 8 kHz and a 128 channels over-complete representation sampled at 1 kHz using a convolution as the encoder and a transposed convolution as the decoder, both with a kernel size of 16 and stride of 8. The high dimensional representation is masked through a separation network composed of stacked residual blocks. Each block is composed of a 1x1 convolution, a PReLU [He et al., 2015a] non linearity, a layer wise normalization over all channels jointly [Ba et al., 2016], a depth-wise separable convolution [Chollet, 2017, Howard et al., 2017] with a kernel size of 3, a stride of 1 and a dilation of  $2^{n \bmod N}$ , with  $n$  the 0-based index of the block and  $N$  an hyper-parameter, and another PReLU and normalization. The output of each block participates to the final mask estimation through a skip connection, preceded by a 1x1 convolution. The original Conv-Tasnet counted  $3 \times N$  blocks with  $N = 8$ . The mask is obtained summing the output of all blocks and then applying ReLU. The output of the encoder is multiplied by the mask and before going through the decoder.

**Conv-Tasnet for music source separation** We adapted their architecture to the task of stereophonic music source separation: the original Conv-Tasnet has a receptive field of 1.5 seconds for audio sampled at 8 kHz, we take  $N = 10$  and increased the kernel size (resp. stride) of the encoder/decoder from 16 (resp. 8) to 20 (resp. 10), leading to the same receptive field at 44.1 kHz. We observed better results using  $4 \times N$  blocks instead of  $3 \times N$  and 256 channels for the encoder/decoder instead of 128. With those changes, Conv-Tasnet obtained state-of-the-art performance on the MusDB dataset, surpassing all known spectrogram based methods by a large margin as shown in Section 6.

**Separating entire songs** The original Conv-Tasnet model was designed for short sentences of a few seconds at most. When evaluating it on an entire track, we obtained the best performance by first



(a) Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections.



(b) Detailed view of the layers  $\text{Decoder}_i$  on the top and  $\text{Encoder}_i$  on the bottom. Arrows represent connections to other parts of the model. For convolutions,  $C_{in}$  (resp  $C_{out}$ ) is the number of input channels (resp output),  $K$  the kernel size and  $S$  the stride.

Figure 2: Demucs complete architecture on the left, with detailed representation of the encoder and decoder layers on the right.

splitting the input track into chunks of 8 seconds each. We believe this is because of the global layer normalization. During training, only small audio extracts are given, so that a quiet part or a loud part would be scaled back to an average volume. However, when using entire songs as input, it will most likely contain both quiet and loud parts. The normalization will not map both to the same volume, leading to a difference between training and evaluation. We did not observe any side effects when going from one chunk to the next, so we did not look into fancier overlap-add methods.

## 4 The Demucs Architecture

The architecture we propose, which we call Demucs, is described in the next few subsections, and the reconstruction loss is discussed in Section 4.2. Demucs takes a stereo mixture as input and outputs a stereo estimate for each source ( $C = 2$ ). It is an encoder/decoder architecture composed of a convolutional encoder, a bidirectional LSTM, and a convolutional decoder, with the encoder and decoder linked with skip U-Net connections. Similarly to other work in generation in both image [Karras et al., 2018, 2017] and sound [Défossez et al., 2018], we do not use batch normalization [Ioffe and Szegedy, 2015] as our early experiments showed that it was detrimental to the model performance. The overall architecture is depicted in Figure 2a.

### 4.1 Convolutional auto-encoder

**Encoder** The encoder is composed of  $L := 6$  stacked convolutional blocks numbered from 1 to  $L$ . Each block  $i$  is composed of a convolution with kernel size  $K := 8$ , stride  $S := 4$ ,  $C_{i-1}$  input channels,  $C_i$  output channels and ReLU activation, followed by a convolution with kernel size 1,  $2C_i$  output channels and gated linear units (GLU) as activation function [Dauphin et al., 2017]. Since GLUs halve the number of channels, the final output of block  $i$  has  $C_i$  output channels. A block is described in Figure 2b. Convolutions with kernel width 1 increase the depth and expressivity of the model at low computational cost. As we show in our ablation study 6.2, the usage of GLU activations after these convolutions significantly boost performance. The number of channels in the input mixture is  $C_0 = C = 2$ , while we use  $C_1 := 64$  as the number of output channels for the first encoder block. The number of channels is then doubled at each subsequent block, i.e.,  $C_i := 2C_{i-1}$  for  $i = 2..L$ , so the final number of channels is  $C_L = 2048$ . We then use a bidirectional LSTM with 2 layers and a hidden size  $C_L$ . The LSTM outputs  $2C_L$  channels per time position. We use a linear layer to take that number down to  $C_L$ .



**Decoder** The decoder is mostly the inverse of the encoder. It is composed of  $L$  blocks numbered in reverse order from  $L$  to 1. The  $i$ -th blocks starts with a convolution with stride 1 and kernel width 3 to provide context about adjacent time steps, input/output channels  $C_i$  and a ReLU activation. Finally, we use a transposed convolution with kernel width 8 and stride 4,  $C_{i-1}$  outputs and ReLU activation. The  $S$  sources are synthesized at the final layer only, after all decoder blocks. The final layer is linear with  $S \cdot C_0$  output channels, one for each source (4 stereo channels in our case), without any additional activation function. Each of these channels directly generate the corresponding waveform.

**U-network structure** Similarly to Wave-U-Net [Jansson et al., 2017], there are skip connections between the encoder and decoder blocks with the same index, as originally proposed in U-networks [Ronneberger et al., 2015]. While the main motivation comes from empirical performances, an advantage of the skip connections is to give a direct access to the original signal, and in particular allows to directly transfers the phase of the input signal to the output, as discussed in Section 4.2. Unlike Wave-U-Net, we use transposed convolutions rather than linear interpolation followed by a convolution with a stride of 1. For the same increase in the receptive field, transposed convolutions require 4 times less operations and memory. This limits the overall number of channels that can be used before running out of memory. As we observed that a large number of channels was key to obtaining good results, we favored the use of transposed convolutions, as explained in Section 6.

**Motivation: synthesis vs masking** The approach we follow uses the U-Net architecture [Ronneberger et al., 2015, Stoller et al., 2018, Jansson et al., 2017], and builds on transposed convolutions with large number of channels and large strides (4) inspired by the approach to the synthesis of music notes of Défossez et al. [2018]. The U-Net skip connections and the gating performed by GLUs imply that this architecture is expressive enough to represent masks on a learnt representation of the input signal, in a similar fashion to Conv-Tasnet. The Demucs approach is then more expressive than Conv-Tasnet, and its main advantages are the multi-scale representations of the input and the non-linear transformations to and from the waveform domain.

**Resampling** We observed improved performance when performing upsampling of the input by a factor of 2, and downsampling of the output to recover the proper sample rate. We perform this operation as part of the end-to-end training loss, with a sinc resampling filter Smith and Gossett [1984]. Note that the same positive impact of resampling has been noted for speech denoising [Défossez et al., 2020].

## 4.2 Loss function

For the reconstruction loss  $L(g_s(x; \theta), x_s)$  in (1), we either use the average mean square error or average absolute error between waveforms: for a waveform  $x_s$  containing  $T$  samples and corresponding to source  $s$ , a predicted waveform  $\hat{x}_s$  and denoting with a subscript  $t$  the  $t$ -th sample of a waveform, we use one of  $L_1$  or  $L_2$ :

$$L_1(\hat{x}_s, x_s) = \frac{1}{T} \sum_{t=1}^T |\hat{x}_{s,t} - x_{s,t}| \quad L_2(\hat{x}_s, x_s) = \frac{1}{T} \sum_{t=1}^T (\hat{x}_{s,t} - x_{s,t})^2. \quad (2)$$

In generative models for audio, direct reconstruction losses on waveforms can pose difficulties because they are sensitive to the initial phases of the signals: two signals whose only difference is a shift in the initial phase are perceptually the same, but can have arbitrarily high  $L_1$  or  $L_2$  losses. It can be a problem in pure generation tasks because the initial phase of the signal is unknown, and losses on power/magnitude spectrograms are alternative that do not suffer from this lack of specification of the output. Approaches that follow this line either generate spectrograms [e.g., Wang et al., 2017], or use a loss that compares power spectrograms of target/generated waveforms [Défossez et al., 2018].

The problem of invariance to a shift of phase is not as severe in source separation as it is in unconditional generation, because the model has access to the original phase of the signal. The phase can easily be recovered from the skip connections in U-net-style architectures for separation, and is directly used as input of the inverse STFT for methods that generate masks on power spectrograms. As such, losses such as  $L_1/L_2$  are totally valid for source separation. Early experiments with an additional term including the loss of Défossez et al. [2018] did not suggest that it boosts performance, so we did not pursue this direction any further. Most our experiments use  $L_1$  loss, and the ablation study presented in Section 6.2 suggests that there is no significant difference between  $L_1$  and  $L_2$ .

### 4.3 Weight rescaling at initialization

The initialization of deep neural networks is known to have a critical impact on the overall performances [Glorot and Bengio, 2010, He et al., 2015b], up to the point that Zhang et al. [2019] showed that with a different initialization called fixup, very deep residual networks and transformers can be trained without batch normalization. While Fixup is not designed for U-Net-style skip connections, we observed that the following different initialization scheme had great positive impact on performances compared to the standard initialization of He et al. [2015b] used in U-Networks.

Considering the so-called Kaiming initialization [He et al., 2015b] as a baseline, let us look at a single convolution layer for which we denote  $w$  the weights after the first initialization. We take  $\alpha := \text{std}(w)/a$ , where  $a$  is a reference scale, and replace  $w$  by  $w' = w/\sqrt{\alpha}$ . Since the original weights have element-wise order of magnitude  $(KC_{\text{in}})^{-1/2}$  where  $K$  is the kernel width and  $C_{\text{in}}$  the number of output channels, it means that our initialization scheme produces weights of order of magnitude  $(KC_{\text{in}})^{-1/4}$ , together with a non-trivial scale. Based a search over the values [0.01, 0.05, 0.1], we select  $a = 0.1$  for all the regular and transposed convolutions, see Section 6 for more details. We experimentally observed that on a randomly initialized model applied to an audio extract, it kept the standard deviation of the features along the layers of the same order of magnitude. Without initial rescaling, the output the last layer has a magnitude 20 times smaller than the first.

### 4.4 The shift trick

A perfect source separation model is time equivariant, i.e. shifting the input mixture by  $X$  samples will shift the output  $Y$  by the exact same amount. Thanks to its dilated convolutions with a stride of 1, the mask predictor of Conv-Tasnet is naturally time equivariant and even if the encoder/decoder is not strictly equivariant, Conv-Tasnet still verifies this property experimentally [Luo and Mesgarani, 2019]. Spectrogram based method will also verify approximately this property. Shifting the input by a small amount will only reflect in the phase of the spectrogram. As the mask is computed only from the magnitude, and the input mixture phase is reused, the output will naturally be shifted by the same amount. On the other hand, we noticed that our architecture did not naturally satisfy this property. We propose a simple workaround, the *shift trick*<sup>1</sup>, where we sample  $S$  random shifts of an input mixture  $x$  and average the predictions of our model for each, after having applied the opposite shift. This technique does not require changing the training procedure or network architecture. Using  $S = 10$ , we obtained a 0.3 SDR gain, see Section 6.2 for more details. It does make evaluation of the model  $S$  times slower, however, on a V100 GPU, separating 1 minute of audio at 44.1 kHz with Demucs takes only 0.8 second. With this technique, separation of 1 minute takes 8 seconds which is still more than 7 times faster than real time.

## 5 Experimental setup

### 5.1 Evaluation framework

**MusDB and additional data** We use the MusDB dataset [Rafi et al., 2017], which is composed of 150 songs with full supervision in stereo and sampled at 44100Hz. For each song, we have the exact waveform of the drums, bass, other and vocals parts, i.e. each of the sources. The actual song, the mixture, is the sum of those four parts. The first 84 songs form the *train set*, the next 16 songs form the *valid set* (the exact split is defined in the musdb python package) while the remaining 50 are kept for the *test set*. We collected raw stems for 150 tracks, i.e., individual instrument recordings used in music production software to make a song. We manually assigned each instrument to one of the sources in MusDB. We call this extra supervised data the *stem set*. We also report the performances of Tasnet and Demucs trained using these 150 songs in addition to the 84 from MusDB, to analyze the effect of adding more training data.

**Source separation metrics** Measurements of the performance of source separation models was developed by Vincent et al. for blind source separation [Vincent et al., 2006] and reused for supervised source separation in the SiSec Mus evaluation campaign [Stöter et al., 2018]. Similarly to previous work [Stoller et al., 2018, Takahashi and Mitsufuji, 2017, Takahashi et al., 2018], we focus on the

---

<sup>1</sup>Fabian-Robert Stöter is to be credited for the name.



SDR (Signal to Distortion Ratio) which measures the log ratio between the volume of the estimated source projection onto the ground truth, and the volume of what is left out of this projection, typically contamination by other sources or artifacts. Other metrics can be defined (SIR and SAR) and we present them in the supplementary material. We used the python package `museval` which provide a reference implementation for the SiSec Mus 2018 evaluation campaign. As done in the SiSec Mus competition, we report the median over all tracks of the median of the metric over each track computed using the `museval` package.

## 5.2 Baselines

We compare to several state-of-the-art baselines, both in the spectral and temporal domain. Open Unmix [Stöter et al., 2019], a 3-layer BiLSTM model with encoding and decoding fully connected layers on spectrogram frames. It was released by the organizers of the SiSec 2018 [Stöter et al., 2018] to act as a strong reproducible baseline and matches the performances of the best candidates trained only on MusDB. We also selected MMDenseLSTM [Takahashi et al., 2018], a multi-band dense net with LSTMs at different scales of the encoder and decoder. This model was submitted as TAK2 and trained with 804 extra labeled songs<sup>2</sup>.

We also compare to the more recent options like the D3Net [Takahashi and Mitsufuji, 2020] model, which used dilated convolutions with dense connections, as well as the Spleeter model [Hennequin et al., 2020], a U-Net model trained on a very diverse dataset. MMDenseLSTM, D3Net, Spleeter, and Open Unmix use Wiener filtering [Nugraha et al., 2016] as a last post processing step. We provide the metrics for the Ideal Ratio Mask oracle (IRM), which computes the best possible mask using the ground truth sources and is the topline of spectrogram based method [Stöter et al., 2018].

The only waveform based method submitted to the SiSec 2018 evaluation campaign is Wave-U-Net [Stoller et al., 2018] with the identifier STL2. We also compare to the more recent variant of DPRNN by Nachmani et al. [2020], Meta-Tasnet [Samuel et al., 2020], and Tasnet trained on extra data [Lancaster and Souviraà-Labastie, 2020].

## 5.3 Training procedure

**Epoch definition and augmentations** One epoch over the dataset is defined as a pass over all 11-second extracts with a stride of 1 second. We use a random time shift between 0 and 1 second and keep 10 seconds of audio from there as a training example. We perform the following data augmentation [Uhlich et al., 2017], also used by Open Unmix and MMDenseLSTM: shuffling sources within one batch to generate a new mix, randomly swapping channels, random scaling by a uniform factor between 0.25 and 1.25. We additionally multiply each source by  $\pm 1$  [Nachmani and Wolf, 2019]. Following [Cohen-Hadria et al., 2019], we also experiment with pitch/tempo shift. 20% of the time, we randomly change the pitch by -2, -1, 0, +1, or +2 semitones, and the tempo by a factor taken uniformly between 0.88 and 1.12, using the Soundstretch utility<sup>3</sup>. When trained with extra data, we do not use pitch/tempo augmentation for Conv-Tasnet as it deteriorates its performance.

All Demucs models were trained over 360 epochs. Conv-Tasnet was trained for 360 epochs when trained only on MusDB and 240 when trained with extra data and using only 2-seconds audio extracts, due to its high memory usage.

**Training setup and hyper-parameters** The Demucs models are trained with 8 V100 GPUs with 16GB of RAM, while the Conv-Tasnet models were trained with 16 V100 with 32GB of RAM. We use a batch size of 64, the Adam [Kingma and Ba, 2015] optimizer with a learning rate of  $3e-4$ . For the Demucs model, we experimented with an initial number of channels in [32, 48, 64], with the best performance achieved for 64 channels. Based on early experiments, we set the initial weight rescaling reference level described in Section 4.3, to 0.1. We computed confidence intervals using 3 random seeds in Table 1. For the ablation study on Table 4, we provide metrics for a single run.

**Quantization** For quantization, we use the DiffQ method [Défossez et al., 2021]. DiffQ uses additive noise on the weights as a proxy for the true quantization error. The scale of the noise depends on learnable parameters that controls the number of bits used per group of 8 weight entries. Because

<sup>2</sup>Source: <https://sisec18.unmix.app/#/methods/TAK2>

<sup>3</sup><https://www.surina.net/soundtouch/soundstretch.html>

Table 1: Comparison of Conv-Tasnet and Demucs to state-of-the-art models that operate on the waveform (Wave-U-Net, Meta-Tasnet, DPRNN-like, Tasnet trained with extra data) and on spectrograms (Open-Unmix without extra data, D3Net, MMDenseLSTM with extra data) as well as the IRM oracle on the MusDB test set. The *Extra?* indicates the number of extra training songs used. We report the median over all tracks of the median SDR over each track, as done in the SiSec Mus evaluation campaign [Stöter et al., 2018]. The A11 column reports the average over all sources. Demucs metrics are averaged over 3 runs, the confidence interval is the standard deviation over  $\sqrt{3}$ . In bold are the values that are statistically state-of-the-art either with or without extra training data.

| Architecture              | Wav? | Extra? | Test SDR in dB   |                  |                  |             |             |
|---------------------------|------|--------|------------------|------------------|------------------|-------------|-------------|
|                           |      |        | All              | Drums            | Bass             | Other       | Vocals      |
| IRM oracle                | ✗    | N/A    | 8.22             | 8.45             | 7.12             | 7.85        | 9.43        |
| Wave-U-Net                | ✓    | ✗      | 3.23             | 4.22             | 3.21             | 2.25        | 3.25        |
| Open-Unmix                | ✗    | ✗      | 5.33             | 5.73             | 5.23             | 4.02        | 6.32        |
| Meta-Tasnet               | ✓    | ✗      | 5.52             | 5.91             | 5.58             | 4.19        | 6.40        |
| Conv-Tasnet <sup>†</sup>  | ✓    | ✗      | 5.73 ±.10        | 6.02 ±.08        | 6.20 ±.15        | 4.27 ±.03   | 6.43 ±.16   |
| DPRNN                     | ✓    | ✗      | 5.82             | 6.15             | 5.88             | 4.32        | 6.92        |
| D3Net                     | ✗    | ✗      | 6.01             | <b>7.01</b>      | 5.25             | <b>4.53</b> | <b>7.24</b> |
| Demucs <sup>†</sup>       | ✓    | ✗      | 6.28 ±.03        | 6.86 ±.05        | <b>7.01</b> ±.19 | 4.42 ±.06   | 6.84 ±.10   |
| Spleeter                  | ✗    | ~ 25k* | 5.91             | 6.71             | 5.51             | 4.55        | 6.86        |
| TasNet                    | ✓    | ~ 2.5k | 6.01             | 7.01             | 5.25             | 4.53        | 7.24        |
| MMDenseLSTM               | ✗    | 804    | 6.04             | 6.81             | 5.40             | 4.80        | 7.16        |
| Conv-Tasnet <sup>††</sup> | ✓    | 150    | 6.32 ±.04        | 7.11 ±.13        | 7.00 ±.05        | 4.44±.03    | 6.74 ±.06   |
| D3Net                     | ✗    | 1.5k   | 6.68             | 7.36             | 6.20             | <b>5.37</b> | <b>7.80</b> |
| Demucs <sup>†</sup>       | ✓    | 150    | <b>6.79</b> ±.02 | <b>7.58</b> ±.02 | <b>7.60</b> ±.13 | 4.69 ±.04   | 7.29 ±.06   |

\*: each track is only 30 seconds, †: from current work, ††: trained without pitch/tempo augmentation, as it deteriorates performance.

Table 2: Mean Opinion Scores (MOS) evaluating the quality and absence of artifacts of the separated audio. 38 people rated 20 samples each, randomly sample from one of the 3 models or the ground truth. There is one sample per track in the MusDB test set and each is 8 seconds long. Ratings of 5 means that the quality is perfect (no artifacts).

| Architecture | Quality Mean Opinion Score |           |           |           |           |
|--------------|----------------------------|-----------|-----------|-----------|-----------|
|              | All                        | Drums     | Bass      | Other     | Vocals    |
| Ground truth | 4.46 ±.07                  | 4.56 ±.13 | 4.25 ±.15 | 4.45 ±.13 | 4.64 ±.13 |
| Open-Unmix   | 3.03 ±.09                  | 3.10 ±.17 | 2.93 ±.20 | 3.09 ±.16 | 3.00 ±.17 |
| Demucs       | 3.22 ±.09                  | 3.77 ±.15 | 3.26 ±.18 | 3.32 ±.15 | 2.55 ±.20 |
| Conv-Tasnet  | 2.85 ±.08                  | 3.39 ±.14 | 2.29 ±.15 | 3.18 ±.14 | 2.45 ±.16 |

this transformation is completely differentiable, one can simply use a model size penalty (expressed in MegaBytes), with a weight of 0.0003, added to the main reconstruction loss.

## 6 Experimental results

In this section, we first provide experimental results on the MusDB dataset for Conv-Tasnet and Demucs compared with state-of-the-art baselines. We then dive into the ablation study of Demucs.

### 6.1 Comparison with baselines

We provide a comparison with the state-of-the-art baselines on Table 1. The models on the top half were trained without any extra data while the lower half used unreleased training songs. As no previous

Table 3: Mean Opinion Scores (MOS) evaluating contamination by other sources. 38 people rated 20 samples each, randomly sampled from one of the 3 models or the ground truth. There is one sample per track in the MusDB test set and each is 8 seconds long. Ratings of 5 means no contamination by other sources.

| Architecture | Contamination Mean Opinion Score |                |                |                |                |
|--------------|----------------------------------|----------------|----------------|----------------|----------------|
|              | All                              | Drums          | Bass           | Other          | Vocals         |
| Ground truth | 4.59 $\pm$ .07                   | 4.44 $\pm$ .18 | 4.69 $\pm$ .09 | 4.46 $\pm$ .13 | 4.81 $\pm$ .11 |
| Open-Unmix   | 3.27 $\pm$ .11                   | 3.02 $\pm$ .19 | 4.00 $\pm$ .20 | 3.11 $\pm$ .21 | 2.91 $\pm$ .20 |
| Demucs       | 3.30 $\pm$ .10                   | 3.08 $\pm$ .21 | 3.93 $\pm$ .18 | 3.15 $\pm$ .19 | 3.02 $\pm$ .20 |
| Conv-Tasnet  | 3.42 $\pm$ .09                   | 3.37 $\pm$ .17 | 3.73 $\pm$ .18 | 3.46 $\pm$ .17 | 3.10 $\pm$ .17 |

work included confidence intervals, we considered the single metric provided by for the baselines as the exact estimate of their mean performance.

**Quality of the separation** We first observe that Demucs and Conv-Tasnet outperform previous waveform domain methods such as Wave-U-Net. Demucs surpasses the best spectrogram domain method, D3Net [Takahashi and Mitsufuji, 2020], when averaging over all sources, by 0.3 dB of SDR. When looking into more details, we see a clear advantage for spectrogram domain methods for the `other` and `vocals` sources. Without extra training data, D3Net also beats Demucs for the `drums`. However, we will see in the next section that human evaluations show that spectrogram domain methods tend to deteriorate the attack of an instrument, so that it is still possible that this high SDR hides significant degradation of the attack of percussive sounds. As no audio samples were released for D3Net, this is impossible to verify.

When trained with extra training data, we still see a small advantage overall from Demucs over D3Net (0.1 point of SDR). Again we notice that the `bass` and `drums` sources are better handled in the temporal domain, while `other` and `vocals` are better handled in the spectrogram domain. Note that for the first time for music source separation, a temporal domain model beats the IRM oracle for the `bass` source, by 0.5 points of SDR.

An interesting experiment would be retrain D3Net with pitch/tempo augmentation and see if it benefits from it. Interestingly, the Conv-Tasnet model achieved the same SDR whether this augmentation was used or not, and the SDR even deteriorated when trained with extra tracks. For Demucs, which comparatively has more tunable weights (see Table 5), the augmentation always improves performance, most likely by preventing some overfitting. We conjecture that pitch/tempo shift lead to deteriorated audio, but reduce overfitting. For Demucs, this is a clear win, while for smaller models, it might not be as beneficial.

**Human evaluations** We noticed strong artifacts on the audio separated by Conv-Tasnet, especially for the `drums` and `bass` sources: static noise between 1 and 2 kHz, hollow instrument attacks or missing notes as illustrated on Figure 1. In order to confirm this observation, we organized a mean opinion score survey. We separated 8 seconds extracts from all of the 50 test set tracks for Conv-Tasnet, Demucs and Open-Unmix. We asked 38 participants to rate 20 samples each, randomly taken from one of the 3 models or the ground truth. For each one, they were required to provide 2 ratings on a scale of 1 to 5. The first one evaluated the quality and absence of artifacts (1: many artifacts and distortion, content is hardly recognizable, 5: perfect quality, no artifacts) and the second one evaluated contamination by other sources (1: contamination if frequent and loud, 5: no contamination). We show the results on Tables 2 and 3.

Note that for this experiment, with used an early version of Demucs that was trained with 100 channels, and without pitch/tempo augmentation. We had trouble differentiating between the latest version of Demucs and this early version in an informal blind test performed by the authors, so that we believe the results presented here stay mostly valid for the best performing Demucs model presented on Table 1.

We confirmed that the presence of artifacts in the output of Conv-Tasnet degrades the user experience, with a rating of  $2.85 \pm .08$  against  $3.22 \pm .09$  for Demucs. On the other hand, Conv-Tasnet samples

Table 4: Ablation study for the novel elements in our architecture described in Section 4. We use only the train set from MusDB and report best L1 loss over the valid set throughout training as well the SDR on the test set for the epoch that achieved this loss.

| Difference                               | Valid set | Test set |
|--|-----------|----------|
|  | L1 loss   | SDR      |
| no BiLSTM                                | 0.171     | 5.40     |
| no pitch/tempo aug.                      | 0.156     | 5.80     |
| no initial weight rescaling              | 0.156     | 5.88     |
| no 1x1 convolutions in encoder           | 0.154     | 5.89     |
| ReLU instead of GLU                      | 0.157     | 5.92     |
| no BiLSTM, depth=7                       | 0.160     | 5.94     |
| MSE loss                                 | N/A       | 5.99     |
| no resampling                            | 0.153     | 6.03     |
| no shift trick                           | N/A       | 6.05     |
| kernel size of 1 in decoder convolutions | 0.153     | 6.11     |
| Reference                                | 0.150     | 6.28     |

Table 5: Impact of the initial number of channels on the model size, and on the performance. We also report the results obtained from quantizing the model with DiffQ. The bottom part consist in the base Demucs and Conv-Tasnet models presented on Table 1.

| Model                                | Model size (MB) | Valid set    | Test set    |
|--------------------------------------|-----------------|--------------|-------------|
|                                      |                 | L1 loss      | SDR         |
| Demucs, 32 channels                  | 254             | 0.154        | 6.08        |
| Demucs, 48 channels                  | 570             | 0.151        | 6.20        |
| Demucs, 64 channels, DiffQ quantized | 120             | <b>0.150</b> | <b>6.28</b> |
| Demucs, 64 channels                  | 1014            | <b>0.150</b> | <b>6.28</b> |
| Conv-Tasnet                          | 42              | 0.152        | 5.73        |

had less contamination by other sources than Open-Unmix or Demucs, although by a small margin, with a rating of  $3.42 \pm .09$  against  $3.30 \pm .10$  for Demucs and  $3.27 \pm .11$  for Open-Unmix.

**Training speed** We measured the time taken to process a single batch of size 16 with 10 seconds of audio at 44.1kHz (the original Wave-U-Net being only trained on 22 kHz audio, we double the time for fairness), on a single GPU, ignoring data loading and using `torch.cuda.synchronize` to wait on all kernels to be completed. MMDenseLSTM does not provide a reference implementation. Wave-U-Net takes 1.2 seconds per batch, Open Unmix 0.2 seconds per batch and Demucs 1.4 seconds per batch. Conv-Tasnet cannot be trained with such a large sample size, however a single iteration over 2 seconds of audio with a batch size of 4 takes 0.7 seconds.

## 6.2 Ablation study for Demucs

We provide an ablation study of the main design decisions for Demucs in Table 4. Given the cost of training a single model, we did not compute confidence intervals for each variation.

We notice strong contributions for different design decisions we made. The L1 loss seems to be more robust than the MSE. The LSTM part plays a crucial role, and replacing it by an extra layer of convolution is not sufficient to recover the best possible SDR. The initial weight rescaling described in Section 4.3 provides significant gain, with an extra 0.4 dB for the SDR.

We introduced extra convolutions in the encoder and decoder, as described in Sections 4.1. The two proved useful, improving the expressivity of the model, especially when combined with GLU activation. Using a kernel size of 3 instead of 1 in the decoder further improves performance. We

conjecture that the context from adjacent time steps helps the output of the transposed convolutions to be consistent through time and reduces potential artifacts arising from using a stride of 4.

The pitch/tempo augmentation gives one of the largest gain, with almost 0.5 dB extra for the SDR, which highlights the importance of strong data augmentation when training data is limited.

We observe that using the shift trick as described in Section 4 gives a gain of almost 0.3 SDR. We did not report the validation loss as we only use the stabilization when computing the SDR over the test set. We applied the randomized stabilization to Open-Unmix and Conv-Tasnet with no gain, since, as explained in Section 4.4, both are naturally equivariant with respect to initial time shifts.

### 6.3 Quantization of Demucs

One of the main drawbacks of the Demucs model when compared to other architecture is its large model size, more than 1014MB, against 42MB for Conv-TasNet. On Table 5, we propose two ways to reduce this issue, either by reducing the initial number of channels (32 or 48 channels), which will improve both the model size, as well as reduce the computational complexity of the model, or using the DiffQ quantization technique [Défossez et al., 2021].

While it is possible to use 48 channels without suffering from a large loss in SDR, it seems getting to 32 channels will lead to a decrease of 0.2 dB in performance. On the other hand, quantization will reduce the model size down to 120MB without any loss of SDR. This is still more than the 42MB of Conv-Tasnet, but close to 10x improvement over the uncompressed baseline.

Note that the 120MB is achieved if each integer is exactly coded on the right number of bits. As this is currently not implemented in Python, we rely on `gzip` to approximately reach this ideal size. This leads to a small overhead, so that the download size for our quantized model is 150MB.

## Conclusion

We experimented with two architectures for music source separation in the waveform domain: Demucs and Conv-Tasnet. We show that with proper data augmentation, Demucs surpasses all state-of-the-art architecture in the waveform or spectrogram domain by at least 0.3 dB of SDR. However, there is no clear winner between waveform and spectrogram domain models, as the former seems to dominate for the bass and drums sources, while the latter obtain the best performance on the `vocals` and other sources, as measured both by objective metrics and human evaluations. We conjecture that spectrogram domain models have an advantage when the content is mostly harmonic and fast changing, while for sources without harmonicity (drums) or with strong and emphasized attack regimes (bass), waveform domain will better preserve the structure of the music source.

In terms of training and architecture, we confirm the importance of using pitch/tempo shift augmentations (although the Conv-Tasnet architecture does not seem to benefit from it), as well as using LSTM for long range dependencies, and powerful encoding and decoding layers with 1x1 convolutions and GLU activations.

When trained with extra data, Demucs surpasses for the first time the IRM oracle for the bass source. On the other hand, Demucs still suffers from larger leakage than other architectures, specifically for the `vocals` and other sources, which we will try to reduce in future work.

## References

- E. Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustic Society of America*, 1953.
- A. S. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, MA, 1990.
- DeLiang Wang and Guy J. Brown, editors. *Computational Auditory Scene Analysis*. IEEE Press, Piscataway, NJ, 2006.
- Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. In *14th International Conference on Latent Variable Analysis and Signal Separation*, 2018.
- Naoya Takahashi and Yuki Mitsufuji. Multi-scale multi-band densenets for audio source separation. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017.
- Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. Technical Report 1805.02410, arXiv, 2018.
- Naoya Takahashi and Yuki Mitsufuji. D3net: Densely connected multidilated densenet for music source separation. *arXiv preprint arXiv:2010.01733*, 2020.
- Francesc Lluís, Jordi Pons, and Xavier Serra. End-to-end music source separation: is it possible in the waveform domain? Technical Report 1810.12187, arXiv, 2018.
- Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *ISMIR 2018*, 2017.
- Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.
- Alexandre Défossez, Neil Zeghidour, Usunier Nicolas, Leon Bottou, and Francis Bach. Sing: Symbol-to-instrument neural generator. In *Advances in Neural Information Processing Systems 32*, 2018.
- Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the International Conference on Machine Learning*, 2017.
- Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The musdb18 corpus for music separation, 2017.
- Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, 2006.
- Alice Cohen-Hadria, Axel Roebel, and Geoffroy Peeters. Improving singing voice separation using deep u-net and wave-u-net with data augmentation. In *27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019.
- Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3), 2014.
- Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons, 2004.
- Sam T. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems*, 2001.
- Francis Bach and Michael I. Jordan. Blind one-microphone speech separation: A spectral learning approach. In *Advances in neural information processing systems*, 2005.
- Emad M. Grais, Mehmet Umut Sen, and Hakan Erdogan. Deep neural networks for single channel source separation. In *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.
- Stefan Uhlich, Franck Giron, and Yuki Mitsufuji. Deep neural network based instrument extraction from music. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.



- Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- Jen-Yu Liu and Yi-Hsuan Yang. Denoising auto-encoder with recurrent skip connections and residual regression for music source separation. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018.
- Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. Multichannel music separation with deep neural networks. In *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016.
- F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji. Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software*, 2019.
- Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020. doi: 10.21105/joss.02154. URL <https://doi.org/10.21105/joss.02154>. Deezer Research.
- Laure Prétet, Romain Hennequin, Jimena Royo-Letelier, and Andrea Vaglio. Singing voice separation: A study on training data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (icassp)*, pages 506–510. IEEE, 2019.
- Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. Technical Report 1806.03185, arXiv, 2018.
- Dario Rethage, Jordi Pons, and Xavier Serra. A wavenet for speech denoising. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen, Morten Kolbaek, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 2017.
- Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-channel multi-speaker separation using deep clustering. Technical Report 1607.02173, arXiv, 2016.
- Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R Hershey. End-to-end speech separation with unfolded iterative phase reconstruction. Technical Report 1804.10204, arXiv, 2018.
- Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–50. IEEE, 2020.
- Eliya Nachmani, Yossi Adi, and Lior Wolf. Voice separation with an unknown number of multiple speakers. In *International Conference on Machine Learning*, pages 7164–7175. PMLR, 2020.
- David Samuel, Aditya Ganeshan, and Jason Naradowsky. Meta-learning extractors for music source separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 816–820. IEEE, 2020.
- Emery Lancaster and Nathan Souviraà-Labastie. A frugal approach to music source separation. 2020.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 2015a.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. Technical Report 1607.06450, arXiv, 2016.
- François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. Technical Report 1704.04861, arXiv, 2017.
- Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. Technical Report 1812.04948, arXiv, 2018.

- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. Technical Report 1710.10196, arXiv, 2017.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Technical Report 1502.03167, arXiv, 2015.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.
- Julius Smith and Phil Gossett. A flexible sampling-rate conversion method. In *ICASSP*, volume 9, pages 112–115. IEEE, 1984.
- Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Interspeech*, 2020.
- Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. Technical Report 1703.10135, arXiv, 2017.
- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 2015b.
- Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.
- Eliya Nachmani and Lior Wolf. Unsupervised singing voice conversion. Technical Report 1904.06590, arXiv, 2019.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable model compression via pseudo quantization noise. *arXiv preprint arXiv:2104.09987*, 2021.

# Appendix

## A Results for all metrics

Reusing the notations from Vincent et al. [2006], let us take a source  $j \in 1, 2, 3, 4$  and introduce  $P_{s_j}$  (resp  $P_s$ ) the orthogonal projection on  $s_j$  (resp on  $\text{Span}(s_1, \dots, s_4)$ ). We then take with  $\hat{s}_j$  the estimate of source  $s_j$

$$s_{\text{target}} := P_{s_j}(\hat{s}_j) \quad e_{\text{interf}} := P_s(\hat{s}_j) - P_{s_j}(\hat{s}_j) \quad e_{\text{artif}} := \hat{s}_j - P_s(\hat{s}_j)$$

We can now define various signal to noise ratio, expressed in decibels (dB): the source to distortion ratio

$$\text{SDR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{artif}}\|^2},$$

the source to interference ratio

$$\text{SIR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}$$

and the sources to artifacts ratio

$$\text{SAR} := 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{artif}}\|^2}.$$

As explained in the main paper, extra invariants are added when using the museval package. We refer the reader to Vincent et al. [2006] for more details. In the following, we only provide the metrics for some of the baselines. We refer the reader to the original papers for the results for the other baselines.

| Architecture | Wav?     | Extra?   | Test SIR in dB |            |            |           |            |
|--------------|----------|----------|----------------|------------|------------|-----------|------------|
|              |          |          | All            | Drums      | Bass       | Other     | Vocals     |
| IRM oracle   | <b>X</b> | N/A      | 15.53          | 15.61      | 12.88      | 12.84     | 20.78      |
| Open-Unmix   | <b>X</b> | <b>X</b> | 10.49          | 11.12      | 10.93      | 6.59      | 13.33      |
| Wave-U-Net   | <b>✓</b> | <b>X</b> | 6.26           | 8.83       | 5.78       | 2.37      | 8.06       |
| Demucs       | <b>✓</b> | <b>X</b> | 11.93 ±.04     | 13.46 ±.12 | 13.42 ±.16 | 6.37 ±.08 | 14.48 ±.07 |
| Conv-Tasnet  | <b>✓</b> | <b>X</b> | 11.75 ±.20     | 12.94 ±.89 | 12.62 ±.19 | 7.15 ±.23 | 14.27 ±.59 |
| Demucs       | <b>✓</b> | 150      | 12.86 ±.05     | 14.72 ±.34 | 14.36 ±.08 | 7.82 ±.06 | 14.53 ±.22 |
| Conv-Tasnet  | <b>✓</b> | 150      | 12.24 ±.09     | 13.66 ±.14 | 13.18 ±.13 | 8.40 ±.08 | 13.70 ±.22 |
| MMDenseLSTM  | <b>X</b> | 804      | 12.24          | 11.94      | 11.59      | 8.94      | 16.48      |

| Architecture | Wav?     | Extra?   | Test SAR in dB |           |           |           |           |
|--------------|----------|----------|----------------|-----------|-----------|-----------|-----------|
|              |          |          | All            | Drums     | Bass      | Other     | Vocals    |
| IRM oracle   | <b>X</b> | N/A      | 8.31           | 8.40      | 7.40      | 7.93      | 9.51      |
| Open-Unmix   | <b>X</b> | <b>X</b> | 5.90           | 6.02      | 6.34      | 4.74      | 6.52      |
| Wave-U-Net   | <b>✓</b> | <b>X</b> | 4.49           | 5.29      | 4.64      | 3.99      | 4.05      |
| Demucs       | <b>✓</b> | <b>X</b> | 6.08 ±.01      | 6.18 ±.03 | 6.41 ±.05 | 5.18 ±.06 | 6.54 ±.04 |
| Conv-Tasnet  | <b>✓</b> | <b>X</b> | 6.13 ±.04      | 6.19 ±.05 | 6.60 ±.07 | 4.88 ±.02 | 6.87 ±.05 |
| Demucs       | <b>✓</b> | 150      | 6.50 ±.02      | 7.04 ±.07 | 6.68 ±.04 | 5.26 ±.03 | 7.00 ±.05 |
| Conv-Tasnet  | <b>✓</b> | 150      | 6.57 ±.02      | 7.35 ±.05 | 6.96 ±.08 | 4.76 ±.05 | 7.20 ±.05 |
| MMDenseLSTM  | <b>X</b> | 804      | 6.50           | 6.96      | 6.00      | 5.55      | 7.48      |