Left and right hand source separation on synthesized audio of piano MIDI compositions

Haris Kodžaga
h.kodzaga@student.vu.nl
Vrije Universiteit Amsterdam

June 9, 2021

Abstract

Audio source separation [1] is the audio processing task where a recording is separated in two or more subcomponents. Such subcomponents can be a speaker and background noise [2], a singer and the instrumentals or even the set of individual musical instruments [3]. This research investigates the effectiveness of a supervised audio source separation method against unsupervised source separation methods on a trivial audio source separation task. The research also investigates the question whether the time and labor sacrifice in the creation of a new training set for a supervised method would be compensated by the performance gains said method would achieve relative to unsupervised audio source separation methods.

The task the proposed audio source separation methods are being evaluated against is a new use case that is also being introduced in this research, namely left and right hand (i.e. playing hands) separation on synthesized MIDI compositions of Boogie Woogie piano music. For this use case a dataset had to be created as well, called SepiWoogie. It is a dataset consisting of a total of 95 manually separated songs (55 train, 10 validation and 30 test). The songs are segmented by left hand and right hand performance on the piano and audio source separation methods are expected to replicate this separation in the proposed use case. The use case will in this paper also be referred to as "SepiWoogie hand separation".

The methods used consist of three different unsupervised audio source separation methods and one Deep Learning supervised audio source separation method. The unsupervised methods are 2DFT [7], REPET [8] and REPET-SIM [8],while the supervised method is Open-Unmix [6] by SigSep. Evaluation on performance was mainly performed through empirical analysis using the signal-invariant performance metrics proposed by Le Roux et. al [19]. This evaluation is further supported through an additional listening test however. Ultimately, results have shown evidence both in the quantitative as well as the qualitative analysis that the supervised method well-outperformed the unsupervised methods. This thereby also concludes that even for trivial audio source separation tasks, the effort required for preparing a dataset for a supervised method appears to remain the best option at the time of this writing.

Contents

1 Introduction

Audio source separation (ASS) [1] is the task of separating a target audio signal into two or more of its constituent subcomponents. Such subcomponents can for instance be the set of separated human speech and background noise or the set of separated musical instruments. Therefore as it is to be expected, the target domains in which ASS is applied differ. Popular examples at the moment are the WSJ0-2mix speaker/noise separation task [2] and the MUSDB18 4-track music source separation task [3]. The latter being more recent and arguably more complex, since in the MUSDB18 4-track music source separation task the challenge is to separate a target song into its 4 components. The resulting output is supposed to be 4 separated audio signals consisting of the vocals, bass, drums and "other" separately [3].

As previously mentioned, WSJ0-2mix [2] and MUSDB18 [3] are the most popular datasets and tasks in audio source separation and this is in large part due to a lack of datasets available for audio source separation. There is active work on datasets in the field of course. WHAM! [4] is for instance an extension built on the WSJ0-2mix [2] dataset, while MUSDB18 [3] itself has been published as recently as 2017/18. Regardless however, audio source separation seems to always have had remained focused largely on separating either music or human speech with background noise into subcomponents. The focus on these two specific use cases did admittedly allow for well-developed real world utility. Practical examples of what is currently possible with audio source separation methods are for instance voice isolation [2], processing songs for karaoke [8] (by removing the vocals), speech enhancement [2] or post-production mixing [3].

The fact remains however that development in the field of audio source separation has remained rather vertical. Research remains largely consolidated around the aforementioned use cases, with research in audio source separation prioritizing improving performance on existing use cases instead of trying to explore new avenues. Exploring new avenues is of course easier said than done, with one major hurdle of course being the availability of appropriate datasets. One of the biggest challenges that arises when investigating a new task/use case might very well be the availability of good labeled data. The lack of labeled data not only limits the possibilities of training supervised methods, more importantly however, it is in the way of performing an objective and a statistically sound evaluation of performance. Of course listening tests are an option, however quantitative tests are a norm in audio source separation research for good reason. Performance metrics such as the signal-to-distortion ratio (SDR) [5] allow for a standardized, quantitative means of evaluating audio quality and source separation performance.

This paper aims to address the points that just have been made by investigating a new use case within audio source separation, namely playing hand source separation on the piano. The use case is essentially a bass/treble source separation task and the focus is specifically on synthesized MIDI compositions in the Boogie Woogie piano playing style. The reason to first of all investigate the use case of playing hand source separation on the piano was the relative triviality of the task itself actually. The assumption was that this would be a use case in which unsupervised methods might be a relatively viable option against a supervised audio source separation. Prior to this research there were no labelled datasets which were suitable for the proposed use case of audio source separation applied to the playing hands on the piano played in the Boogie Woogie piano style. Such a labelled dataset would thus have to be created. This is not only to train supervised methods, but also to evaluate the performance of **any** audio source separation method on said use case. Now preparing a labelled dataset is a laborious task and therefore the question might arise whether this

would even be necessary for an audio source separation use case as straightforward such as this one. Furthermore more specifically regarding the domain of the use case itself; one of the characteristics that makes the piano interesting is its wide range (also known as the chromatic range). In fact, it is an acoustic instrument with one of the widest ranges of all acoustic instruments. This means that the pitch range of most acoustic instruments can be contained within the range of the piano. The wide chromatic range, combined with its ergonomics allow the piano to have a versatility to be played with two hands where either hand can play independently from one another. The left hand can for instance take on the task of setting a rhythm with bass patterns, while the right hand plays a melody at a higher register. The pair of bass guitar (left hand) and acoustic guitar (right hand) makes for a good analogy to describe what is possible on the piano.

This delegation of musical tasks across the two hands is also rather common in genres such as Baroque and especially Boogie Woogie music in fact. The left hand is of equal, if not of greater importance than the right hand in the Boogie Woogie piano style [9]. The left hand plays a consistent rhythm at bass (also called a basso ostinato), while the right hand is free to improvise. The genre lends itself very well as a use case for playing hand audio source separation on the piano. This great distinction between the left-hand and right-hand performance namely translates well to a clear separation space in the audio, making any attempt at applying audio source separation methods to this piano style highly feasible.

Putting all of the aforementioned together, the research itself had the following contributions:

- Creating a new dataset[1] consisting of synthesized MIDI piano compositions in Boogie Woogie style had to be created. The end product is a dataset of synthesized songs, called SepiWoogie, in which each song in the set consists of two audio signals in which each file is an isolated mapping from the composition to either the left or right hand. I.e. each song is manually separated by left and right hand performance respectively. This dataset is both to enable evaluation on the proposed use case as well as to train the supervised method for the proposed use case, namely SepiWoogie hand separation.

- Training a supervised, deep learning model on the training set of the newly-developed SepiWoogie dataset. The deep learning method used is Open-Unmix by Stöter et. al. [6]. This supervised method is compared against a set of unsupervised methods. These are 2DFT [7], REPET [8] and REPET-SIM [8]. The aim of this comparison was to investigate whether effort that was required to develop the SepiWoogie dataset is justified by the performance difference a supervised method such as Open-Unmix [6] has over unsupervised methods.

- Performance was evaluated by means of quantitative, as well as qualitative analysis. Quantitative evaluation was performed by evaluating performance of all methods against the test set of the newly-developed SepiWoogie dataset dataset. Qualitative analysis is performed by means of a listening test. It should be clarified however that the results from quantitative evaluation are leading and the results from qualitative evaluation either only support existing findings or attempt to address any inconsistencies observed in quantitative results.

[1] `https://doi.org/10.34740/kaggle/dsv/2099606`

The above contributions had the aim to not only investigate the newly-proposed use case itself, whether the labor for developing a use case-specific dataset for a supervised method is justified. For this a research question has been formalized, namely:

> **Will the labor required to build a labeled dataset for relatively a straightforward audio source separation task be compensated by the performance gain of a supervised method compared to unsupervised methods?**

Out of the main research question above, a sub-question can be formulated, namely: **how would an unsupervised audio source separation method compare against a supervised audio source separation method in the use case of playing hand separation on synthesized audio signals of Boogie Woogie music?** Answering this question first will assist in answering the main research question. It might just be that a use case such as the one proposed in this paper would be sufficiently trivial for an unsupervised method to perform comparatively well to a supervised method. This could thereby also support a claim that preparing a labelled dataset for audio source separation use cases such as this one would not be worth developing when comparing the returns of separation performance against the costs of developing a dataset.

Besides investigating the aforementioned research questions, this work attempts to contribute to the field of audio source separation by being a brief introduction to the field as a whole, however again, also as a challenge and/or an inspiration to the field itself to think outside of the box in terms of use cases for audio source separation. Research in AI/ML generally tends to have a great focus on performance alone by attempting to set a new benchmark such to establish a new state-of-the-art in the field. As previously mentioned, audio source separation is not much different. This is something which becomes more evident when consulting the leaderboards on paperswithcode.com. Both the WSJ0-2mix[2] and the MUSDB18[3] leaderboards show rapid competition between works which started as late as 2018. While again, this constant drive for improvement is of course great for the field, considerable amounts of value could also be gained by thinking more creatively in means of how audio source separation could be applied to other use cases outside of the established ones such as speaker/noise separation and 4-track music source separation.

As far as the paper itself is concerned, this section will be closed by a brief subsection below in which the Boogie Woogie piano playing style will be further explained. After this a related work section follows, where a more detailed explanation on the field of audio source separation is given, along with more detailed descriptions on evaluation metrics, supervised methods, unsupervised methods and related tasks in their own respective subsections. The methodology section then explains the exact approach in how the experiment was conducted, including the evaluation method and the methods that have been used. Due to the significance of the dataset in this research, it has a section of its own, after which the experimental setup goes into further detail regarding the experiments. Results of the experiments are then reported in the results section. Discussion and the conclusion follow and the paper ends with suggestions for future work in their own section. Next to this, there is also an appendix. The appendix contains commands that have been used, along with links to code, samples and the dataset that has been used for the experiments, as well more detailed observations and results from the experiments held.

[2] `https://paperswithcode.com/sota/speech-separation-on-wsj0-2mix`
[3] `https://paperswithcode.com/sota/music-source-separation-on-musdb18`

1.1 The use case: Boogie Woogie

Boogie Woogie is a genre that first of all appears to be somewhat limited in terms of literature. The most substantial body of work on Boogie-Woogie is Peter J. Silvester's [9] "The Story of Boogie-Woogie: A Left Hand Like God". Other works that assisted in describing the background and theory behind Boogie Woogie are by Newberger [10] and Logan [11]. Boogie Woogie can be defined as a piano playing style that is particularly known for its repetitive, rhythmic bass lines. Rhythmic patterns of this kind are also known as an Ostinato [11] and playing the Ostinato in the lower register (as is characteristic of Boogie Woogie) is also known as a Basso Ostinato. Basso Ostinato patterns are played with the left hand on the piano, leaving the right hand to play accompanying melodies in a higher register [9, 10, 11]. Silvester perhaps summarizes the characteristics of Boogie Woogie most favorably as a use case for this research in his following excerpt [9]:

> "Where the guitar player switched between playing in the bass and treble ranges of his instrument, the pianist was able to produce a continuous series of varied and contrasting tones at the same time in both registers, each hand working independently of the other—a defining attribute of boogie-woogie playing. The left hand will endlessly repeat an ostinato bass pattern, usually of eight beats to the bar in the three blues chord positions (C, F, and G in the key of C), while the right hand supported or played across the bass rhythm, and in so doing, produced complex cross rhythms."

Other characteristics that stand out are the following:

- Boogie Woogie music is a genre that is typically also described as a piano playing style [9]. It is a playing style that has its origins in the piano and it is hence intended to be played on the piano.
- The (basso) ostinato is considered to be crucial to the genre [9, 10, 11].
- The basso ostinato is played with the left hand on the piano. Therefore by extension, the left hand is of great importance in Boogie Woogie music as well. Sylvester's book even has "A Left Hand Like God" [9] in the title.
- Given the great importance of not only the left playing hand but also the distinction between the two hands, it will thus also be important to be able to distinguish either hands from one another.

The distinction between left and right can also be seen in figure 1. The score of Albert Ammon's "Woo Woo" has pairs of rows (also known as staves) for the treble clef and bass clef respectively. The distance between the two staves as seen on the score generally appears to be at least one octave (e.g. a distance between C4 and C5 or similar). Furthermore judging from the score, the left hand does not play higher than A3. This note corresponds with a frequency of 220Hz.

These aforementioned observations are reflected in the spectrogram in figure 1. There is a cluster of relatively higher density between 0Hz and 220Hz and second cluster above it, mainly concentrated around 410Hz and 520Hz in the spectrogram of figure 1. This translates to a range between B4 and C5. Not only do the notes written in the treble staves in figure 1 commonly fall within this aforementioned range, the distance between A3 and B4 is more than one octave. Thereby also confirming the first observation made about from score in figure 1.

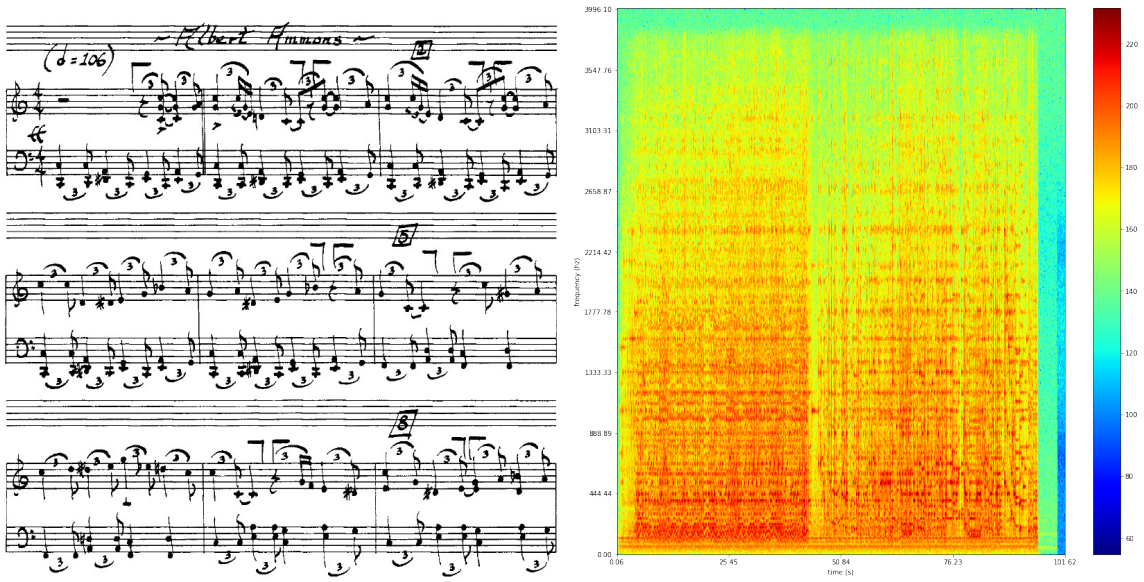

Figure 1: "Woo Woo" by Albert Ammons. A segment of the written score of the song [10] on the left and a spectrogram of the recording of the same song at the right.

2 Related Work

As established in the introduction, the aim of audio source separation [1] is to separate an input audio signal into two or more of its constituent subcomponents. Audio signals are most commonly separated through means of time-frequency masking over the time-frequency representation of a target audio signal [1]. The time-frequency domain [1], to put it simply, is an alternative representation of audio to the widely-known waveform domain. In the waveform domain, signal magnitude over time is clearly shown, while signal frequency is less obvious. Transformations to the time-frequency domain solves the latter, transforming an audio signal to a representation in which frequencies rather than magnitudes of an audio signal are shown over time on the y-axis.

$$\textbf{STFT}\,\{\mathrm{x(t)}\} \equiv \mathrm{X}(\pi,\omega) = \int_{\infty}^{-\infty} \mathrm{x(t)w(t-\tau)e^{-j\omega t}dt} \tag{1}$$

Applying a transformation such as the Short-Time Fourier Transform (STFT) [1] over the waveform representation of an audio signal will change this representation to a spectrogram (time-frequency domain). The formula for the STFT can be seen in 1 above. This is in content still the same audio signal prior to transformation, however in a spectrogram the signal is represented as frequencies over time instead, i.e. the time-frequency domain. Figures 2 and 3 show a pair of audio signals that have been transformed from a waveform to a spectrogram. Especially the example in figure 3 consisting of the mixed sinewave signals highlights the difference between an audio signal before and after being transformed to a time-frequency representation. Figures 2 and 3 also show how the time-frequency representation is useful for performing audio source separation, as the mix of audio sources within a given audio signal become more visually distinguishable.

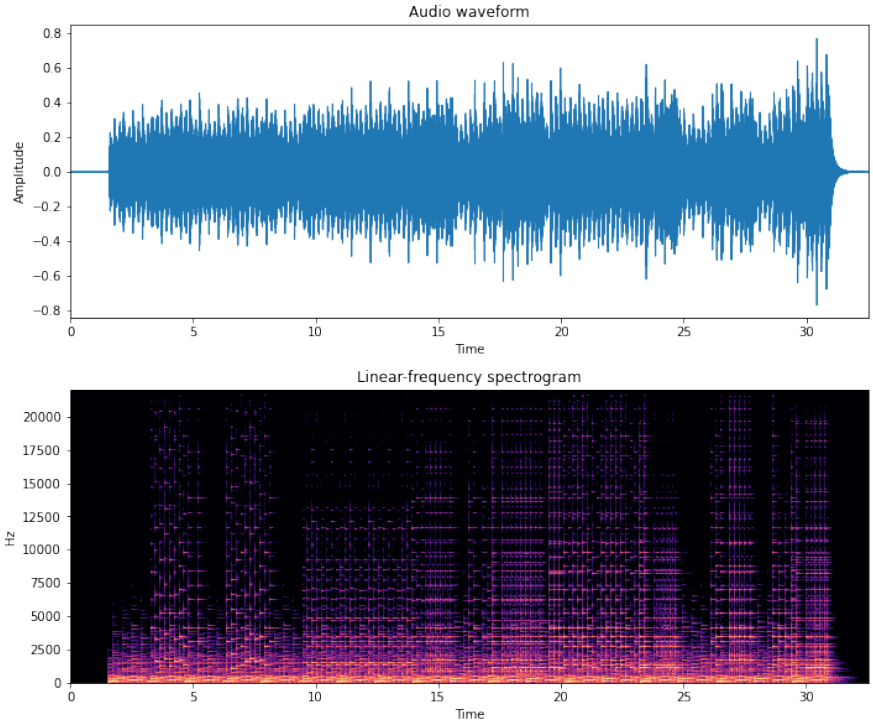

Figure 2: A sample of a Boogie Woogie piano song visualised both as a waveform (upper image) and a spectrogram (lower image).

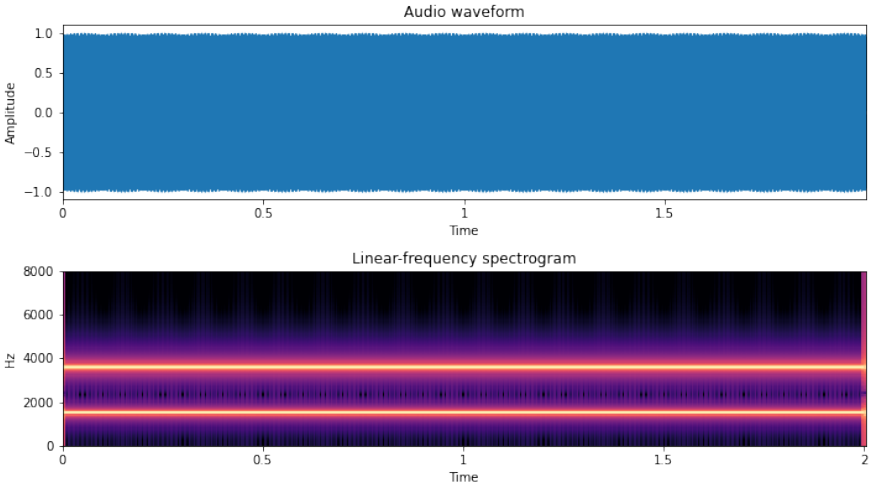

Figure 3: A sample of a pair of sine waves visualised both as a waveform (upper image) and a spectrogram (lower image).

This more obvious distinction thus translates well to audio source separation tasks, over which time-frequency masks can be applied. Masking, to put it simply, is a filtering method in which subcomponents are isolated from a target input. In the case of audio source separation, a mask approximates the subset of the input signal it wishes to separate and keep, while ignoring the rest. In tasks such as speaker/noise separation the number of masks is equal to two, since the speaker and the background noise are their own respective subsets which have to be separated from each other. In tasks such as the MUSDB-18 4-track music source separation [3] tasks the number of masks required is equal to four, as there are four subsets of instruments/sources which have to be separated from each other.

While spectral masking in the time-frequency domain has been a widely used method over time and across different tasks [2, 3] (both WSJ0-2mix speaker/noise separation and MUSDB18 4-track music source separation), there are some drawbacks [14] when working in the time frequency domain. Chief among these drawbacks is the fact that phase and magnitude data is lost when transforming to the time-frequency domain. With the phase and magnitude data missing, any audio data transformed to the time-frequency domain cannot render to a playable audio signal. This is because frequency data contains information regarding the frequency only, while phase and magnitude data contain information on the volume of an audio signal. Therefore without phase and magnitude data, there is no volume in a an audio signal.

Transformation from a time-frequency representation back to a waveform representation is thus required. Such a transformation is also called a reconstruction, or an inverse STFT [12]. A reconstruction/inverse STFT requires the original waveform signal for phase and magnitude data. To reconstruct, in this case, a separated subcomponent of an input audio recording in the time-frequency representation, the phase and magnitude information over time are taken and applied over the frequencies over time to reconstruct a playable audio signal [12]. When the original mixed input signal is mapped over a separated subcomponent in the time-frequency representation, any phase and magnitude data mapped on frequencies that have been filtered out (i.e. empty/Null data) will in turn not be observable and thus audible. This is because not all frequency data is available to construct the complete mixed signal. In other words, the output of this reconstruction is an audio signal of a separated subcomponent. It reconstructs phase and magnitude data on the remaining frequency data.

One way to avoid these aforementioned obstacles [12] surrounding the time-frequency representation is by avoiding the time-frequency representation entirely. While a time-frequency representation is a different representation of an audio signal, a waveform is essentially the audio signal itself. Thus end-to-end waveform audio source separation [13, 14] is experiencing ever-increasing traction within the field and it appears that many of the state-of-the-art performing methods [15, 16, 17] currently rely on end-to-end waveform audio source separation. More exact details on methods such as these will be covered in the sections further below.

2.1 Evaluation metrics used for Audio Source Separation

The general task of audio source separation rather stands out for its rather unique set of evaluation metrics which are used to measure separation performance. Therefore in order to be able to interpret performance results on audio source separation tasks, one needs to understands the evaluation metrics which are unique to the field of audio source separation first. Broadly speaking,

there is a target signal s which is a pre-separated subcomponent of a mixed audio signal. The estimated output subcomponent audio which has been separated by an audio source separation method is denoted as $\hat{s}$. The aim for any audio source separation method is to have an output $\hat{s}$ that approximates s and therefore minimizes the error of s - $\hat{s}$.

Vincent et. al [5] popularized a set of four metrics back in 2006 which to this day still remain to be used as the main set of evaluation metrics for audio source separation tasks, such the MUSDB18 4-track music source separation task. The four metrics which were proposed by Vincent et. al [5] are SDR (signal-to-distortion ratio), SIR (signal-to-interference ratio), SNR (signal-to-noise ratio) and SAR (signal-to-artifact ratio). Scores for all four metrics are expressed as the signal strength expressed in dB. This can essentially be interpreted as the signal strength over the amount of error (can be general distortion, interference, noise or artifacts) that has been introduced to the target signal. The higher the score is in dB, the lower the amount of errors are present in a target signal, i.e. the closer a separated signal is also to the ground truth.

Also contrary to metrics such accuracy where an absolute ceiling of 100% performance can be achieved, the ceiling with metrics like SDR for instance is relative. In other words, a median SDR score of 100dB would for example not necessarily be indicative of "perfect performance." The relative ceiling of performance for a target signal can be set by using a upper baseline method such as the ideal binary mask [18] or the subcomponents of the target signal itself. Let us say the SDR for instance of the subcomponent of the signal would be 40dB, this would in turn be the ceiling of performance.

Out of the four metrics, SDR is the most popular. SDR was in fact proposed to be used as a global metric for separation performance [5], as it essentially incorporated all other three proposed metrics from that paper into a single metric. SIR, the signal-to-interference ratio expresses the amount of non-target audio remaining in the target signal. In the case of speaker/noise separation, SIR would express how much background noise appears to remain in the separated speaker signal. SNR is the signal-to-noise ratio and it is more or less a measure that expresses the amount of audio quality loss that might have occurred due to separating an audio signal. Lastly, SAR is the signal-to-artifact ratio and it is a description of the amount of unwanted audio artifacts that might have been created in the process of separating an audio signal. Examples of artifacting might for instance be hissing or pops in the audio that were not present before.

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

$$\text{SIR} := 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \tag{3}$$

$$\text{SNR} := 10\log_{10}\frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{noise}}\|^2} \tag{4}$$

$$\text{SAR} := 10\log_{10}\frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2} \tag{5}$$

Criticism has been leveled towards the metrics proposed by Vincent et. al [5] however. Le Roux et. al [19] proposed to make the metrics "scale-invariant" to address their own claims that that the metrics from Vincent et. al [5] are not sufficiently robust. As mentioned in at the beginning

of this section, the evaluation metrics for audio source separation can be generalized as the error (i.e. distance) of a separated signal $\hat{s}$ and a manually, pre-separated target signal s. As with the metrics by Vincent et. al, the estimate $\hat{s}$ and the ground truth s essentially both exist as vectors in a two-dimensional space and the distance between the two is the error. Le Roux et. al [19] claim that such error evaluation can be tampered by boosting or decreasing the signal of $\hat{s}$ relative to s. This reduces the distance and in turn artificially boosts the score. Le Roux et. al [19] essentially addresses these shortcomings through an evaluation method that is invariant on scaling differences (hence the name "scale-invariant") and in turn evaluating error on cosine similarity only between the vectors s and $\hat{s}$.

2.2 Supervised audio source separation

As with supervised learning research in general, the availability of labeled datasets for certain drives research for said tasks. This can also be seen in the field of audio source separation, where the most popular tasks at the moment are speaker/noise separation and the MUSDB18 4-track music source separation task. Both tasks have labeled datasets on which supervised audio source separation methods can be trained on to tackle the aforementioned tasks. The most popular dataset at the moment for speaker/noise separation is WSJ0-2mix [2] and SigSep[4] has developed the MUSDB18 dataset [3] for the MUSDB18 4-track music source separation task. Both tasks have seen extensive research throughout the years, as evident by the leaderboards on WSJ0-2mix[5] and MUSDB18[6] respectively. Also while the introduction at section 1 did mention that audio source separation methods (such as the Conv-Tasnet by Luo et. al [13]) generally can be trained and function on both tasks, there are also instances of more bespoke solutions to one of the tasks at hand. DEMUCS by Défossez et. al [24] for instance, is a variational autoencoder method developed to tackle the MUSDB18 4-track music source separation task. Next to the 100 song-counting training set available by MUSDB18, DEMUCS is able to train on additional unlabeled (musical) data. This enabled DEMUCS to be the best performing method on the MUSDB18 4-track music source separation task at the time of its writing [24].

Breakthroughs in Deep Learning research has of course impacted the field of Machine Learning as a whole and this includes audio source separation as well. The use of Convolutional Neural Network (CNN) [13, 14, 15, 20, 21] layers seem to be favored in the field for their balance between performance and efficiency. Recurrent Neural Networks (RNNs) [22] initially also seem to be a great fit for audio source separation, due to the temporal nature of the task. However, such architectures tend to be computationally expensive and inefficient [23] and therefore do not see much action as audio source separation methods. Other great alternative are (Bi-directional) Long Short-Term Memory (LSTM) and Variational Autoencoder (VAE) networks. Examples of such architectures in action are Open-Unmix by Stöter et. al [6] and DEMUCS by Défossez et. al [24] respectively.

One notable phenomenon that also seems to occur in the field of audio processing as a whole is the adaptation [25] of solutions which were initially from entirely different fields of research [26]. One such example within the field of audio source separation specifically is the work by Jansson et. al [27], in which the U-Net system by Ronneberger et. al [28] has been adapted to perform

[4] `https://sigsep.github.io/`

[5] `https://paperswithcode.com/sota/speech-separation-on-wsj0-2mix`

[6] `https://paperswithcode.com/sota/music-source-separation-on-musdb18`

audio source separation. U-Net [28] was initially developed for image segmentation purposes and it tackled this task through an architecture of CNNs that perform multi-layer convolutions across several levels of increasingly-downsampled feature maps in parallel. U-Net seems similar to variational autoencoders [24] or encoder/decoder [13, 6] architectures, however rather than learning how to reconstruct/generate a decoded output from an encoded input, it is directly performing repeated convolutions throughout its downsampling and upsampling process.

Improvements to the U-Net architecture [27] have been introduced by Stoller et. al [14] with their proposal of Wave-U-Net. One of the main improvements was its ability to perform audio source separation directly in the waveform domain, as opposed to requiring transformation from waveform to time-frequency representations (e.g. using STFTs) such to be able to perform audio source separation. This allowed it to reap the benefits which were already discussed in the last paragraph of the related work, section 2. This made it along with Wavenet [29], TasNet [30] and SEGAN [31] one of the first methods to perform "end-to-end" audio source separation, i.e. audio source separation directly within the waveform domain itself. New iterations based on these aforementioned methods can be seen in examples, such as DEMUCS [24] and Conv-TasNet [13].

Lastly, reports on performance on especially the MUSDB18 4-track music source separation task [3] has shown that performance of audio source separation methods remain to be very much tied to the amount of training data that is available to any given method. The aforementioned DEMUCS architecture by Défossez et. al [24] has been trained on additional sets of 2000 unlabeled samples of data and 100 labeled samples of data to boost its performance. The training set from the MUSDB18 dataset counts 100 songs, 14 of which are to be used for validation.

When considering the additional training data available to DEMUCS, the performance it managed to achieve becomes less impressive. The same applies to Deezer's Spleeter by Hennequin et. al [21], it is a U-Net [27] architecture which was trained on a private dataset which is owned by Deezer itself. The size of the training set is unknown, however there is evidence to suggest that it is at least greater than the MUSDB18 dataset. This can firstly be explained by the fact that Spleeter achieved state-of-the-art performance [21] on the MUSDB18 task [3] at the time of its writing using the relatively outdated U-Net architecture. The U-Net architecture was already surpassed by different methods at the time [13, 6, 29] and GitHub users have also proven[7] that Spleeter sees a considerable drop in performance when trained on only the MUSDB18 training set. One of the highest scores on the MUSDB18 4-track separation task at the time of this writing has been achieved by Lancaster et. al [32] using the TasNet architecture, an architecture [30] which was introduced to tackle the WSJ0-2mix speaker/noise separation task [2] back in 2017. No changes appear to be made to the existing TasNet architecture, just the addition of more more training data which is an estimated 30 times larger [32] than the MUSDB18 training set itself.

Related work such as the aforementioned [21, 24, 32] are one of the motivations to consider investigate the opportunities for unsupervised methods. The addition of more training data remains to seemingly outperform architectural improvements in Deep Learning, however the creation/acquisition of more labeled training data (especially 300 hours of additional data [32]) remains to be a significant investment in terms of time and labor.

[7] `https://github.com/deezer/spleeter/issues/81#issuecomment-633524469`

2.3 Unsupervised audio source separation

Unsupervised music source separation approaches attempt to perform music source separation without the system being trained on a ground truth (as is the case with supervised methods). Unsupervised methods try to circumvent the lack of ground truth data through either a strictly knowledge-driven approach or unsupervised learning. Unsupervised learning implies the learning to generalize patterns observed in data without the presence of a ground truth. Modern examples of unsupervised learning applied to audio source separation is the exploitation of spatial differences in multi-channel recordings as done by Seetharam et. al [33] and Bando et. al [34]. By making recordings using a set of microphones, the recorded sources should sound slightly different through each recorded channel. These differences ultimately allow an audio source separation system to identify and in turn separate one source from the rest.

Exploiting spatial information in multi-channel recordings is not always possible however, such use cases could for instance be single-channel audio recordings. Well-known alternatives for unsupervised learning in audio source separation applications are for instance NNMF (Non-Negative Matrix Factorization) and ICA (Independent Component Analysis) [35]. ICA for instance, is a linear transformation method which attempts to find independence. It assumes that there are subcomponents in a signal which are independent from each other. The end result is a transformed feature space that can consist of k-number of subcomponents. The content of the sum of these subcomponents should be as similar as possible to the original signal, while maintaining an statistical independence (i.e. difference) between the subcomponents which is as high as possible.

Where the former examples are data-driven examples of unsupervised audio source separation, there are also knowledge-driven alternatives. With knowledge-driven applications some assumptions are made which are based on domain knowledge. In the case of singer and accompaniment separation for instance, the assumption is that the accompaniment generally has some form of repetition while the vocals tend to have more liberties. One such example exploiting this assumption is REPET (REpeating Pattern Extraction Technique) [8]. REPET is an unsupervised audio source separation method originally developed for singer/song separation, i.e. separating vocals from the instrumentals in songs. As claimed by Rafii et. al [8], REPET's advantage is in its claimed simplicity. All it in fact does is performing audio source separation through detecting repeating audio patterns. Adaptations to REPET in favor of greater generalization lead to REPET-SIM, as proposed by Rafii et. al [8] again. REPET-SIM's functionality is still grounded in the assumption that low-density, repetitive signals is noise which can be separated from a foreground signal. REPET-SIM is however claimed to achieve greater generalizability by not relying on periodicity to separate a foreground signal from a background [8]. This has namely opened the door REPET-SIM to be applicable for other audio source separation use cases beyond singer/song separation, such as speaker/noise separation.

More modern approaches to unsupervised audio source separation can be seen in work for Seetharam et. al [36] for instance. In this work, an unsupervised model was trained in a two-step approach. An ensemble of primitive clustering methods (such as the aforementioned [8, 35]) generated soft-masks as output after processing an input signal. Now, the soft-mask itself is now adequate to already perform audio source separation on the input signal. However going one step further, a set of mixed input signals and corresponding output soft-masks were used to bootstrap a deep learning model. The model effectively learns to generalize on non-ground truth examples by using the output soft-masks from the primitive clustering ensemble as input for itself.

2.4 Piano hand detection

Taking a short sidestep away from audio source separation and into (MIDI) piano hand detection. Piano hand detection is, as the name itself might already reveal the task of detecting which hand is playing what on the piano. All hand detection-related tasks discovered so far [37, 38] deal with data in MIDI representation. Works by Hadjakos et. al [37] and Nakamura et. al [38] are examples which are closest aligned to the task imagined for this project.

However as stated before, these systems detect what notes are played by which hands at a MIDI level, while the proposed task aims to tackle a similar challenge, however in the domain raw audio signals. In other words, in order for the systems by either Hadjakos et. al [37] or Nakamura et. al [38] to function for the proposed task, they will have to be adjusted to suit the proposed task. Work by Nakamura et. al [38] is even going as far as to estimate fingering. More on Nakamura et. al [38], one of the noteworthy things is also that they managed to achieve better performance on a Higher-Order Hidden Markov Model across all of their evaluation metrics when compared to their deep learning LSTM implementation.

The system by Hadjakos et. al [37] does show the greater potential for adaptation to the proposed piano signal hand separation task however. Signal transcription to MIDI [39, 40] can be implemented where a piano recording will be transformed to MIDI such to be used as input for the piano hand detection system. The input data, transcribed and hence transformed to MIDI can be processed by the piano hand detection system [37]. The output of this system could in theory then be two separated MIDI files (one file for each playing hand) which can be rendered to a raw audio signal. These rendered audio signals can then be used as masks to filter out the target signal in similar fashion as what is common in audio source separation.

While the above is certainly a possible and interesting way in addressing this particular use case of playing hand source separation, it is first of all something which is outside of the scope of the this paper's research at the moment. Furthermore, the hypothetical end product would be dependent on the proper functioning of two complex subtasks, namely MIDI piano hand detection and audio signal to MIDI transcription. Besides the latter, the set of separated outputs runs significant risk of being of inadequate quality. The aforementioned suggestion namely assumes that an inference mask generated from a separated synthesized MIDI composition will have a similar audio signal to the target signal it had transcribed. Audio source separation on the other hand, is a well-researched task with work which has proven its effectiveness on challenges (MUSDB18 4-track music source separation [3] and WSJ0-2mix speaker/noise source separation [2]) which are arguably exceedingly more complex than the current use case at hand.

Therefore when everything considered, adapting MIDI piano hand detection to perform audio source separation does not seem as a feasible, let alone good solution and it will therefore neither be further investigated nor considered in this paper. The greater complexity of implementation relative to common audio source separation methods [6, 8, 13, 24] makes this approach of adapting MIDI piano hand detection unattractive, especially when considering that matching performance to that with traditional audio source separation methods would also be highly unlikely. This would therefore make all the required effort to realize the above not worthwhile.

3 Methodology

This section will go into more detail regarding the methods that were used and it will also cover this research's approach to evaluating audio source separation performance. Next to the supervised method and the unsupervised methods, lower and upper baseline methods were established as well. These will be explained further in the experimental setup at section 5 however. Pragmatism, obstacles encountered during research and a limited scope due to a time limitations has driven a large part of the design choices behind the methods used. This did lead to a preference to seek off-the-shelf tools and libraries for both evaluation and the audio source separation methods themselves. The set of methods ultimately used consists of one supervised method, namely Open-Unmix [6] by SigSep[8] and three unsupervised methods, namely 2DFT [7], REPET [8] and REPET-SIM [8].

Moving on to evaluation, this consisted of two parts. First, a quantitative evaluation on separation quality was performed as per standard in the field of audio source separation. The second part builds on the aforementioned quantitative evaluation with a qualitative evaluation in the form of a set of listening tests. The quantitative analysis consisted of the set of observations over the samples of the test set, median scores for each evaluated method and the differences in performance between methods were ultimately tested for significance. Significance testing was performed using either the Wilcoxon test if at least one of the distributions were not normally distributed or the T-test when both distributions were normally distributed. The metrics used to for the quantitative evaluation is the set of the scale-invariant metrics which were proposed in Le Roux et. al [19]. These metrics are SI-SDR, SI-SIR and SI-SAR. For all three metrics higher values equate to better performance on said metrics.

To briefly recap the metrics used [5, 19]; SI-SDR is the scale-invariant signal-to-distortion ratio. It is generally seen as a global measure of separation quality. SI-SIR is the scale-invariant signal-to-interference ratio and it is a metric that describes the lack of presence of non-target data in the separated signal. SI-SAR is the scale-invariant signal-to-artifacts ratio and it shows the amount of artifacts introduced in the target signal due to audio source separation. It should also be said that SI-SAR is arguably the least important metric of the three. The SI-SAR score of a method really only becomes insightful once its SI-SIR score is already high. The point is; maintaining a lower amount of artifacting is more impressive when a method is also effective at separating a target source compared to when its source separation performance (i.e. SI-SIR score) is already low.

Next to quantitative results as discussed above, the listening tests attempt to offer a different perspective on the results through a qualitative evaluation of the performance of the different audio source separation methods. What this entails to is that the qualitative evaluation aims to serve either a supporting or an investigating purpose. Results from quantitative analysis are leading and the qualitative analysis serves to support evidence from quantitative analysis. Only when potential inconsistencies and/or abnormalities are observed from the quantitative results will qualitative analysis attempt to serve an investigative purpose.

The reason for this hierarchical difference in importance between quantitative and qualitative analysis lies in the inherent risks in qualitative analysis and an attempt to mitigate these. Qualitative analysis is namely performed as a listening test in which 6 randomly sampled songs from the SepiWoogie test set across all different methods are evaluated on their audio quality. This evaluation has been performed by the author only. In other words, due to the risks from a limited set

[8] `https://sigsep.github.io/`

of evaluated samples and the subjective bias of performing a listening test such as in this research, the role of the qualitative analysis is limited to clarifying existing results or uncertainties observed from the quantitative analysis.

As mentioned above, qualitative analysis was performed on 6 randomly sampled songs[9] from the SepiWoogie test set, the different methods were evaluated on three different criteria. These criteria ultimately functioned as their own frameworks in which to differently evaluate all audio source separation methods while listening to the randomly picked samples. The criteria which are being proposed are **separation performance**, **separation cleanliness** and **source preservation**. Inspiration has been taken from the popular performance evaluation metrics [5, 19] that are being used in audio source separation, although other than that the proposed criteria are not based on any prior work and were contrived during this research itself.

Separation performance is similar to SI-SIR in terms of what it is exactly evaluating. In other words; how well a method/model is able to separate source signals from each other. Separation cleanliness and sound preservation describe levels of distortion in their own ways. Separation cleanliness describes the amount of distortion and/or artifacting introduced on top of the target source signal. Sound preservation describes the amount of information loss that may have occurred due to processing. Information loss can be in the form of loss of audio signal, distortion, muffling et cetera. Lastly, there is also a rating for overall separation quality. It should be emphasized however that the overall separation quality score is not the average of the aforementioned criteria. This is rather to be seen as a final conclusion after all things considered.

3.1 Supervised Method

As stated in the introduction, Open-Unmix [6] been selected as the Music Source Separation system for this research. It has near state-of-the-art performance, good documentation and it allows for relatively easy adaptation of the system to datasets other than just MUSDB18 [8]. More exact details on the implementation and parameters used for the experiment in this research can be found under section 5.1.1 of the experimental setup.

Open-Unmix (abbreviated as UMX) [6] is a multi-model audio source separation system that performs audio source separation within the time-frequency domain, i.e. on and using spectrograms. It is a multi-model system is the sense that for each source one wishes to separate from an input signal, a separate model would have to be trained for that specific source. In use cases, such as speaker/noise separation [2], training just a single model would suffice, since the speaker signal is the target signal that is to be separated from the mixed signal (speaker + background noise). For tasks such as the MUSDB18 4-track music source separation task however [3], four models are to be trained if said task is to be tackled properly. The reason for this is the fact that the a model for each of the four sources to be separated from the mixed signal is to be trained.

Moving on, the architecture of Open-Unmix is described as consisting of "very classical elements" [6] and it is built around a three-layer BLSTM network. As figure 4 shows, bandwidth of input data is firstly limited to a 16 kHz bandwidth. In preparation for for the three-layer BLSTM network, dimensionality reduction/encoding is furthermore applied to the input data through a fully-connected network and this data is afterwards passed through a tanh activation layer as well. The three-layer

[9] `https://www.dropbox.com/sh/bmi3ziyxgwwc1nm/AACipL0jK5R8tQP9iU2GTi5la?dl=0`

BLSTM proceeds to generalize on spectrogram of the target source/subcomponent relative to the spectrogram of the mixed signal containing the target source/subcomponent.

The output of the BLSTM network is afterwards passed through fully-connected network and a ReLu activation twice. This is to decode the output to a reconstructed spectrogram filter mask (in the initial dimensionality of the spectrogram itself). Batch normalization is also applied at several stages throughout the process, a skip connection exists over the three-layer BLSTM network and also a multiplicative skip-connection across the entirety of the aforementioned process. The purpose of the skip-connections is to ultimately smooth out the gradient descent [42].

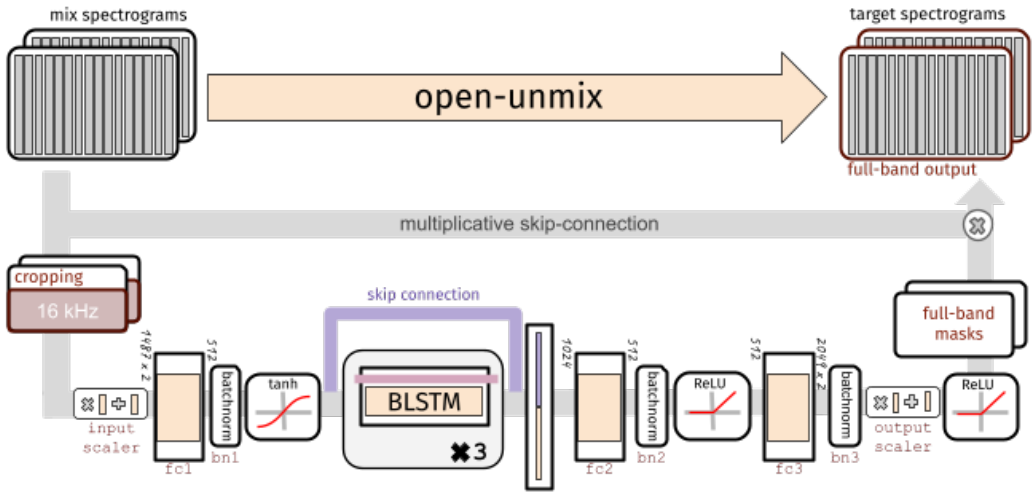

Figure 4: The architecture behind the Open-Unmix [6] audio source separation method. The image has been taken from SigSep's GitHub page (`https://github.com/sigsep/open-unmix-pytorch`) of Open-Unmix.

3.2 Unsupervised Methods

As already stated in the research proposal in the introduction, the unsupervised methods used consisted of 2DFT [7], REPET [8] and REPET-SIM [8]. This section will go into further detail on REPET and REPET-SIM. 2DFT will not receive similar attention, since this method serves more as a lower baseline of some sorts in terms of performance. Section 2.3 in the related work about unsupervised methods already touched briefly on both REPET and REPET-SIM. These methods are knowledge-driven rather than data-driven, meaning that neither of these methods train models to generalize on a dataset in any shape or form. What they do actually do however, is performing separation/calculations which are based on assumptions on the domain/task/application/problem they are being used for. Assumptions that both REPET [8] and REPET-SIM [8] seem to use is the fact low-density, repetitive signals are likely noise which should be separated as background noise from a foreground signal (hence the name REpeating Pattern Extraction Technique, REPET).

Also the general stages behind the process of both REPET and REPET-SIM can be summarized as identification (1), modelling (2) and extraction/separation (3) [8]. Figure 5 shows a visualization of this aforementioned process and it also makes clear how the two methods differ from each other. REPET aims to generate a single repeating segment S that is meant to be a generalized sequence of

the background signal. REPET-SIM is able to find and generate a set of several different repeating sequences to make a generalization on the background noise. This allows REPET-SIM to achieve its claimed [8] greater flexibility and in turn, a wider range of possible applications.

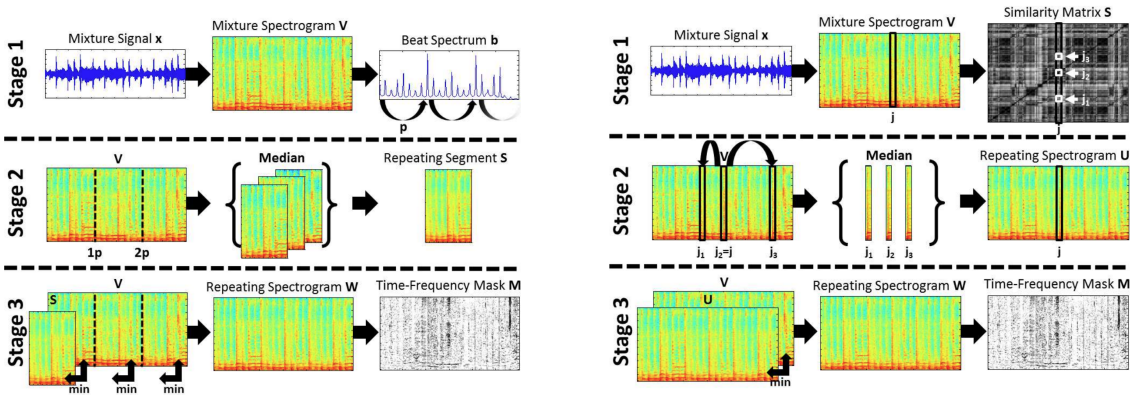

Figure 5: The three-stage process (identification (1), modelling (2) and extraction/separation (3)) behind REPET (left) and REPET-SIM (right). The image has been taken from "REPET for Background/Foreground Separation in Audio" by Rafii et. al [8]

Moving on into further detail however and starting off with REPET, the first identification stage results in periodicities that had been detected in the beat spectrum of a target signal. A beat spectrum b of a target signal x is obtained by applying an autocorrelation function on the mixture spectrogram V of the transformed target signal x. The autocorrelation is essentially the cross-correlation a target signal has with itself, i.e. how similar a signal is with itself. The beat spectrum b is ultimately computed by taking the mean of the over the rows of the matrix that is the output returned by the autocorrelation. The repeating period p in the beat spectrum b is then able to be identified and thereby completing the first step of the process.

The second stage of modelling consists of segmenting the mixture spectrogram V of the transformed target signal x into repeating periodicities $p_1, ..., p_n$. Out of this set of repeating periodicities, the median is taken and used as the repeating segment S. The method then proceeds to the final stage and derives a repeating spectrogram model W using the repeating segment model S. This repeating spectrogram model W is derived by taking the minimum element-wise distance between the repeating segment S and each of the segments $p_1, ..., p_n$ that exist in the mixture spectrogram V of the transformed target signal x. A soft time-frequency mask M can then be derived from repeating spectrogram model W by normalizing the spectrogram model W by the mixture spectrogram V, such that the values of the elements in the matrix of the soft time-frequency mask M always are between 0 and 1.

Section 2.3 of the related work about the unsupervised methods already briefly touched on the improvements of REPET-SIM [8]. REPET is namely restricted by its ability in only being able to detect repeating periodicities, while struggling with more intermittent noise [8]. REPET-SIM mainly addresses these shortcomings with the use of the similarity matrix rather than relying on autocorrelation to detect repeating elements [8]. Starting with a mixture spectrogram V of the

target signal x, the aim in the first stage is to identify the unique set of repeating frames and for each repeating frame j to find its set of j_k frames across the target signal's duration that are similar to j. First, the columns of the matrix V are normalized by their Euclidean norm, after which its similarity matrix S is computed by multiplying the matrix of V with the transposed matrix of V.

The computation of the similarity matrix S essentially allows for the measurement of the cosine similarity between two frames j_a (where $j_a = j$) and j_b. When the similarity between the two frames is sufficient, j_b is added to the subset of frames that are similar to the repeating frame j. This process is repeated until all frames in a target signal are assigned to a subset of a unique repeating frame. To model the repeating spectrogram U, the median of each subset is computed and used to create a reconstructed repeating spectrogram U. From this spectrogram a refined repeating spectrogram can be derived, which will in turn be used to create a soft time-frequency mask M for the target signal x.

4 Boogie Woogie synthesized piano dataset, SepiWoogie

As stated in the introduction, there was no fitting dataset to serve the purposes of this particular research. This resulted in the creation of the SepiWoogie dataset, a dataset for training and evaluating audio source separation methods on left and right playing hand separation. The dataset can be accessed here: `https://doi.org/10.34740/kaggle/dsv/2099606`

Development on the dataset started with collecting MIDIs of piano compositions which were in Boogie Woogie style. These MIDI compositions were found throughout the internet. Links to the sources of the MIDIs which were used can be found in the documentation of the dataset itself. As far as selection criteria are concerned, the MIDIs clearly had to be Boogie Woogie compositions and therefore had to meet the following criteria [9, 10, 11]:

- Clear "task" separation between hands. This often also translates to strong bass and treble separation.
- Bass needs to consist of repetitive, rhythmic bass lines, i.e. the Basso Ostinato.
- Clear use of Blues chord progressions (C, F, and G in the key of C), especially in bass.

After collecting and inspecting the set of Boogie Woogie MIDIs, Ableton was used to manually separate these MIDI compositions into two parts. As figure 6 shows, each part is the separate performance of the left hand and the right hand respectively. Both parts were then rendered to separate audio signals through a piano preset which was tuned to sound similar to an upright piano. The resulting rendered audio signals are in .WAV format.

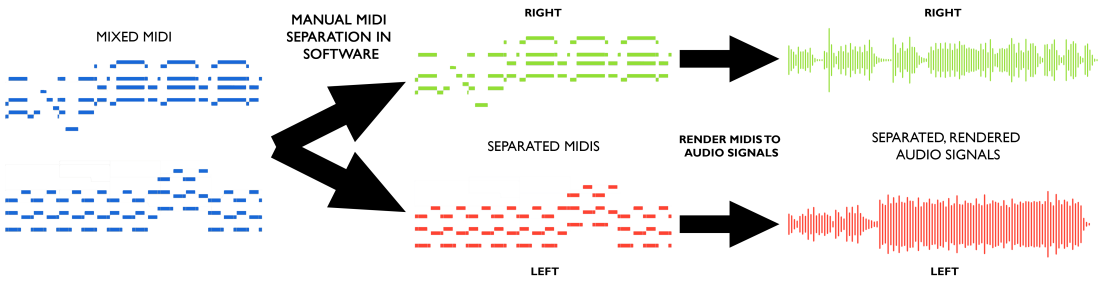

Figure 6: A simplified overview of the process to preparing the SepiWoogie dataset.

A set of guidelines to preparing the dataset were also established. Compositions had to fit within the chromatic range of the piano, such to ensure that the synthesized compositions used at least sounded like something which could be reproduced on a real piano. This ultimately meant fitting the composition within the bounds of keys A0 (27.5Hz) and C8 (4186Hz). Furthermore, the composition required a tempo which is true to the style of Boogie Woogie [9]. This meant that synthesized songs had to have a BPM of around 175 beats per minute (BPM). The conclusion of desiring a BPM of 175 also arrived from a set of observations that have been made after listening to different Boogie Woogie songs. While C4 is technically the middle key and therefore also the border between bass and treble, this separation did not always occur around this point in the compositions that were encountered. Shifting these compositions up or down in octaves was not always desirable either, since this either threw parts of a composition outside of the chromatic range of a piano or simply made the composition itself sound off. Therefore, not all compositions in the set of the Boogie Woogie synthesized piano dataset had its separation point at around C4. Rather, compositions were separated at points where a clear contrast/separation between left hand performance and right hand performance was observed.

The resulting dataset consists of 95 songs, each of which consists of *(1)* a left-hand performance file, *(2)* a right-hand performance file and *(3)* a file which is the mix of both aforementioned files. All audio files are in .WAV format and there is a total of 285 .WAV audio files that make up the dataset. Of the 95 songs, 55 are part of the training set, the validation set counts 10 songs and the test set has a total of 30 songs. The split of the dataset over the training, validation and test set is therefore 57.9%, 31.6%, 10.5% respectively. Songs were distributed over the training set, validation set and test set through means of random sampling.

Regarding the labor involved in preparing the dataset, research and quality inspection of MIDIs took an estimated 5 hours. Each song took an average of roughly 30 minutes to prepare. Considering that there is a total of 95 songs, the time required to manually separate and synthesize all songs equaled to an estimated 48 hours. When adding the time spent on research and quality inspection, the total adds up to 52 hours. Thus on average, preparation of the training, validation and test set took around 30, 16 and 6 hours respectively.

5 Experimental Setup

This section will go through the details of the experiment. The methodology at section 3 already touched on details regarding the approach to the evaluation method along with the methods which were used in this research. Section 4 meanwhile described details on the dataset which has been developed and used in this research. This section being the experimental setup will go into further detail on information such as the resources which have been used to realize this experiment and the exact parameters which have been used for the different methods in this experiment.

The calculations used for the evaluation metrics (as described in section 3)[10] and the implementation of the unsupervised methods (2DFT [7], REPET [8] and REPET-SIM [8]) have been realized using Nussl [41]. Nussl [41] is an audio source separation Python library[11] developed by the The Northwestern University and it contains a full suite of functionality for holding a complete audio source separation experiment. This includes features such as dataset building, audio source separation methods (both unsupervised and supervised) and support for performance evaluation. Implementation of the supervised method was not done using Nussl however. The supervised method which was used, is Open-Unmix [6] by SigSep[12]. Evaluation was performed largely through Nussl [41].

5.1 Methods used

Next to the supervised method and the unsupervised methods used, lower and upper bounds were established as well using lower and upper baseline methods respectively. A lower baseline was set using a high/lowpass filter, while the upper baseline was established using an ideal binary mask [18]. The high/lowpass filter is essentially a constant separation border at a set frequency. Considering how the C4 key is generally regarded as the "middle key", i.e. the key separating the bass (left hand) and the treble (right hand), the separation border was set at its frequency of 261.6256 Hz. The ideal binary mask [18] is a mask inference method in which the mask is simply the original source signal itself. The IBM is intended to be a upper baseline of what is achievable using mask inference methods for audio source separation. The next subsections will explain under which exact parameters the supervised method and unsupervised methods were used.

5.1.1 Supervised Method

When training Open-Unmix [6] on a custom dataset, the documentation[13] states that there are 4 different arguments to choose from. The "trackfolder_fix" argument was considered the best-suited option for the datasets used in this research. This setting assumes that the target training set has a fixed set of sources/files for each folder (i.e. song) it encounters. Every song in the SepiWoogie dataset consists of a left-hand track and right-hand track. When passing the "trackfolder_fix" argument, UMX (Open-Unmix) also requires arguments in which is specified what the target track is and what tracks are supposed to act as the interfering signals. During training, tracks are loaded in as a mix as input and the output consists of the two separated signals, namely the target track and the remaining interfering tracks. In this 2-track left-/right-hand separation task the left-hand

[10] The complete implementation of the quantitative evaluation can be found here: `https://github.com/bosnyan/SepiWoogieEval`

[11] `https://nussl.github.io/docs/index.html`

[12] `https://sigsep.github.io/`

[13] `https://github.com/sigsep/open-unmix-pytorch/blob/master/docs/training.md`

track was designated as the interfering signal and the right-hand was the target.

Evaluating UMX output for this task using Nussl did require some new code to be written. This was to enable audio source separation on a non-MUSDB18 task, namely the piano playing hands task. The code which had been written can be found here: `https://github.com/bosnyan/open-unmix-pytorch`. All of the exact commands used for this research can also be found under section 9.1 in the appendix. All parameters used for this task are the following:

Training parameters:

- batch size: 1
- epochs: 1000
- patience: 140
- learning rate: 0.001
- learning rate decay patience: 80
- learning rate decay gamma: 0.3
- weight decay: 0.00001
- random seed: 42

Model Parameters:

- Sequence duration: 6.0
- Unidirectional LSTM: False
- STFT FFT size/Window size: 2048
- STFT hop size: 512
- Hidden size: 128
- Max bandwidth (in Hz): 16000
- Number of channels: 1
- Number of workers: 0

5.1.2 Unsupervised Methods

As stated in the methodology in section 3.2, the unsupervised methods which were used were 2DFT [7], REPET [8] and REPET-SIM [8]. 2DFT [7] is the most basic unsupervised method of the three and in terms of performance it should in fact not be much greater than just the lower baseline method itself. REPET and REPET-SIM [8] were both more extensively covered in section 2.3 in the related work and in section 3.2 in the methodology. The parameters used for the respective methods are the following:

2DFT:

- Neighborhood size: 1,15
- High pass cutoff: 180.0
- Quadrants to keep: 0,1,2,3
- Filter approach: Local STD
- Use background 2DFT: True
- Mask type: Soft
- Mask threshold: 0.9

REPET:

- Minimum period: None
- Maximum period: None
- Period: None
- High pass cutoff: 200.0
- Mask type: Soft
- Mask threshold: 0.9

REPET-SIM:

- Similarity threshold: 0.3
- Minimum distance between frames: 1
- Maximum repeating frames: 100
- High pass cutoff: 200.0
- Mask type: Soft
- Mask threshold: 0.9

6 Results

The performance of the six different methods has been evaluated using the SI-SDR, SI-SIR and SI-SAR metrics. The distributions of the results on the three different metrics can be seen in figure 7, 8 and 9 respectively. The results on all three metrics have been tested for statistical significance as well. All resulting significance levels of the significance tests on the SI-SDR results, SI-SIR results and SI-SAR results can be seen in table 1, 2 and 3 respectively. Significance levels range from NS/not significant ($p \geq 0.05$), *-level significance ($p < 0.05$), **-level significance ($p < 0.01$), ***-level significance ($p < 0.001$) to ****-level significance ($p < 0.0001$). Tables containing the exact p-values of the significance tests can be found in the appendix at section 9.2. Abbreviations for the methods evaluated have also been used throughout the tables, plots and the text. These abbreviations are IBM (Ideal Binary Mask), HP (high/lowpass filter), 2DFT (2D Fourier Transform), RT (REPET), RT-S (REPET-SIM) and UMX (Open-Unmix).

The first thing what stands out is the SI-SDR performance of UMX relative to the upper baseline method, the ideal binary mask as seen in figure 7. The statistical significance test resulted in a ****-level of significance (i.e. a p-value < 0.0001) as seen in table 1, indicating that the difference between the two distributions is highly statistical significant. Now while this does mean that the margin of SI-SDR performance of the IBM above UMX is statistically significant (i.e. the observed difference is not due to chance), the narrow margin between the results do suggest that the performance of UMX is rather matched to that of the IBM, i.e. upper baseline performance.

This assumption is further supported by results seen in the SI-SIR and SI-SAR metrics in figure 8 and 9 respectively. UMX has similar SI-SIR performance to the upper baseline and in figure 8 UMX seems to outperform the IBM in several test samples. It means that on the SepiWoogie test set, it is able to separate sources with likely the similar performance an upper baseline method could. The previous claim cannot be fully confirmed however, since the results of the significance tests between the distributions of the SI-SIR results in table 2 show that the difference in performance between UMX and the IBM is not statistically significant. Then again, these results can also be interpreted as such where the conclusion is that the SI-SIR performance between UMX and the IBM is actually statistically indistinguishable.

Lastly, figure 9 shows that the SI-SAR performance of UMX (median score of 9.11) is well ahead of the unsupervised methods and the lower baseline, and close to the SI-SAR performance of the upper baseline method that is the IBM (median score of 11.29). This difference itself does not come as a surprise however. Consistent low artifacting (and therefore a high median SI-SAR score) is something that is to be expected from a an upper baseline method. What is in fact surprising however is that UMX actually managed to get this close to an upper baseline method in terms of SI-SAR whilst also having such a gap in SI-SAR performance between itself and the unsupervised methods. Furthermore, the observed differences in SI-SAR performance over the unsupervised methods and the lower baseline all have ****-level significance, i.e. differences which are highly statistically significant. This means that on the SepiWoogie test set, the resulting separated output by the UMX

method is likely to have a lack of distortion and artifacts which is significantly less than the unsupervised methods and the lower baseline method, and in fact not too far off from the upper baseline.

Moving on to the unsupervised methods; none of the three methods come close to the performance of the supervised UMX model trained on the SepiWoogie training set. Of the three methods, REPET-SIM seems to achieve the best performance overall however. Its median SI-SDR and SI-SAR scores are greater than that of the lower baseline and the other two unsupervised methods (2DFT and REPET). In fact, REPET-SIM is the only method which has SI-SDR results which are statistically significant as seen in table 1. In terms of SI-SIR, the median scores as seen in figure 8 of 20.26 and 20.72 for 2DFT and REPET respectively would suggest that REPET-SIM, having a median SI-SIR score of 18.0, has been outperformed by the former on this particular metric.

Significance testing has shown however that the differences between REPET-SIM and the aforementioned unsupervised are not statistically significant. It can therefore not be confirmed whether the observed greater performance by 2DFT and REPET over REPET-SIM in terms of SI-SIR are by mere chance or not. The remaining two unsupervised methods furthermore lag behind with performance and are in fact rather similar to or in fact worse than the lower baseline method, the high/lowpass filter. Both REPET and 2DFT have a narrow edge over the lower baseline method in terms of SI-SDR. The significance tests have shown however that there is no statistical significance in the observed differences between the lower baseline method and the REPET method, as well the lower baseline method and the 2DTF.

Lastly, it appears that the lower baseline method has the highest SI-SIR score as seen in figure 8, thereby even seeming to outperform the upper baseline in terms of SI-SIR. These observed SI-SIR results in figure 8 therefore seem counter-intuitive, since upper and lower baseline were used as a means of establishing an upper and lower bound respectively for the experiment. However, the lower bound now seems to perform better than the upper bound. The next subsection therefore attempts, among others, to explain the observed SI-SIR performance by the high/lowpass filter lower baseline method. Listening tests on a subset of the test set output from all six MSS methods should report performance at a level of detail which the SI-SDR, SI-SIR and SI-SAR are not able to cover.

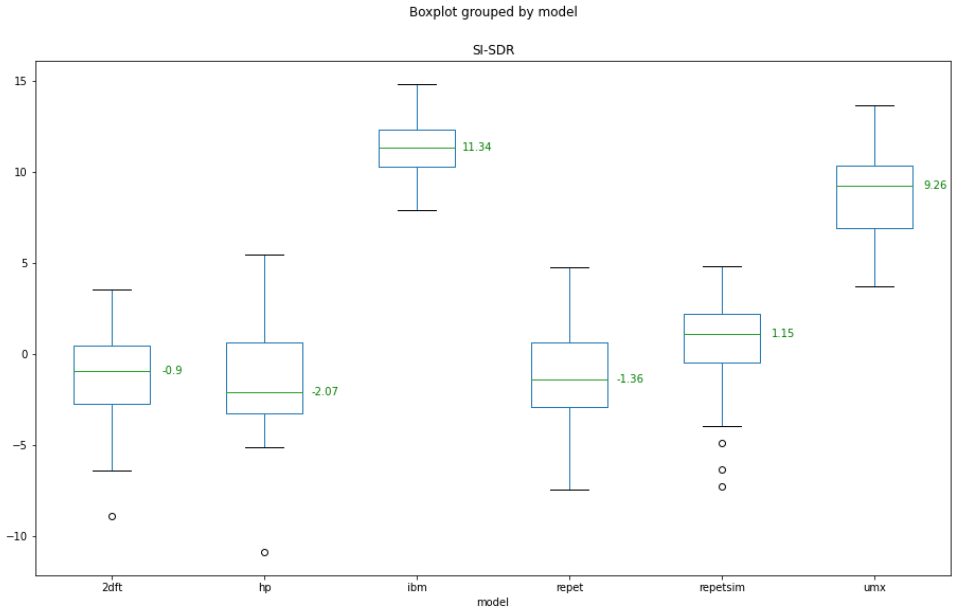

Figure 7: The distributions of SI-SDR results by the different audio source separation methods on the SepiWoogie test set. The median score is indicated in green. **Lower baseline** method: hp, **upper baseline method:** ibm, **supervised method:** umx and **unsupervised methods:** 2dft, repet and repetsim.

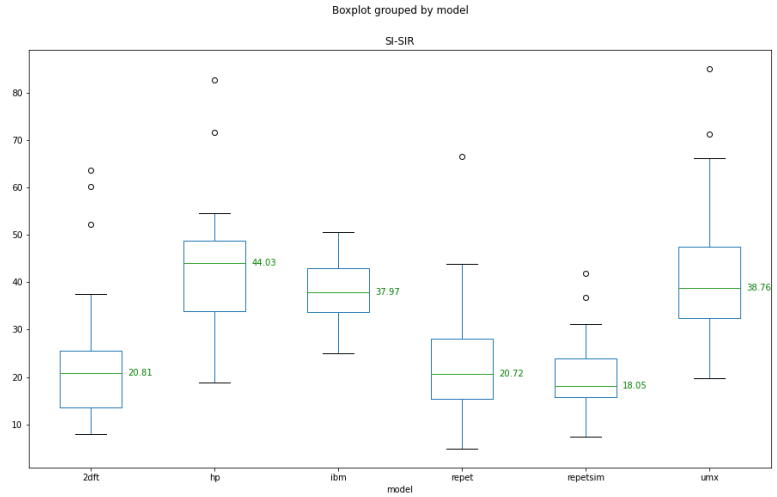

Figure 8: The distributions of SI-SIR results by the different audio source separation methods on the SepiWoogie test set. The median score is indicated in green. **Lower baseline** method: hp, **upper baseline method:** ibm, **supervised method:** umx and **unsupervised methods:** 2dft, repet and repetsim.

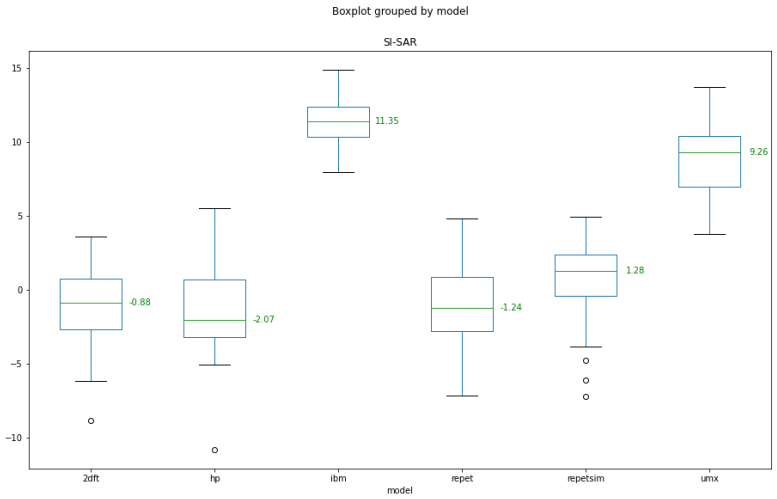

Figure 9: The distributions of SI-SAR results by the different audio source separation methods on the SepiWoogie test set. The median score is indicated in green. **Lower baseline** method: hp, **upper baseline method:** ibm, **supervised method:** umx and **unsupervised methods:** 2dft, repet and repetsim.

Table 1: Obtained significance levels from paired significance tests between the different methods on the distributions of achieved SI-SDR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		****	****	****	****	****
HP	****		NS	***	NS	****
RT	****	NS		****	NS	****
RT-S	****	***	****		****	****
2DFT	****	NS	NS	****		****
UMX	****	****	****	****	****	

Table 2: Obtained p-values from paired significance tests between the different methods on the distributions of achieved SI-SDR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		NS	****	****	****	NS
HP	NS		****	****	***	NS
RT	****	****		NS	NS	***
RT-S	****	****	NS		NS	****
2DFT	****	***	NS	NS		***
UMX	NS	NS	***	****	***	

Table 3: Obtained significance levels from paired significance tests between the different methods on the distributions of achieved SI-SDR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		****	****	****	****	****
HP	****		NS	***	NS	****
RT	****	NS		****	NS	****
RT-S	****	***	****		****	****
2DFT	****	NS	NS	****		****
UMX	****	****	****	****	****	

6.1 Listening Test

Next to the quantitative evaluation which has been covered in the previous section 6, this section reports on the results obtained from qualitative evaluation, the listening test. Exact details on how the listening test was performed can be found again in the latter half of section 3. Evaluation on all six different audio source separation methods has been performed on six randomly sampled songs from the SepiWoogie test set. The listening tests have been done by the author themselves. The results for the listening tests are scores across four different criteria, namely separation performance, separation cleanliness, source preservation and overall separation quality. As mentioned in section 3 before, all four criteria have taken inspiration from quantitative metrics [5, 19] which are popular in the field of audio source separation.

Just to quickly recap, separation performance is a judgement on how well a method is able to separate source signals from each other. Both separation cleanliness and source preservation describe output audio quality. Separation cleanliness is reports on any additional distortion and/or artifacting that has been caused by audio source separation. Source preservation on the other hand reports on the information loss incurred on a source signal due to audio source separation. Lastly, overall separation quality is **not** the mean of the scores of the former three criteria. It should be interpreted as a final conclusion on audio source separation quality after all things considered.

The mean results and the observed standard deviations of the listening test can be seen in table 4. Keep in mind however that the standard deviation scores should be interpreted with a grain of salt, due to to the low number of samples in the listening test. See section 9.3 for the more in-depth results of the listening test of each song individually. Also to repeat section 3 again, the role of the qualitative analysis (i.e. the listening test) is limited to clarifying existing results or uncertainties observed from the quantitative analysis. The reasons behind this approach has been elaborated in section 3 itself. Furthermore in this subsection, abbreviations for the methods evaluated have also been used. Just to clarify again; these abbreviations are IBM (Ideal Binary Mask), HP (high-/lowpass filter), 2DFT (2D Fourier Transform), RT (REPET), RT-S (REPET-SIM) and UMX (Open-Unmix). The randomly sampled subset of songs used for the listening test can be accessed here: `https://www.dropbox.com/sh/bmi3ziyxgwwc1nm/AACipL0jK5R8tQP9iU2GTi5la?dl=0`

The previous subsection ended with the observation that the lower baseline method, the high-/lowpass filter appeared to have the greatest SI-SIR performance among all methods evaluated. This should suggest that the lower baseline method has the best performance when strictly speaking of separation quality, even beating the upper baseline. However the listening tests have proven that the SI-SIR results for the lower baseline method as seen in figure 8 are not representative of the high/lowpass filter's real-world separation performance. The overall separation quality and the separation performance specifically of the lower baseline method as seen in table 4 is instead rated as the worst performing method among all six methods which were evaluated.

The issue with the high/lowpass filter is its highly crude approach to performing audio source separation. By assuming 261.6256 Hz (i.e. the C4 middle key) as a constant and hard separation border, its separation performance was generally far from sufficient to be usable. To be more specific, separation often resulted in output in which the left-hand output signal only contained low fundamental frequency and the right-hand output signal containing the remaining signal. Sound preservation was therefore generally also scored poorly, due to the great amount of information loss present in the left-hand output signal. Mainly these two criteria weighed in to the rating of the overall separation quality. Separation cleanliness was great however, since the separation process

itself did not introduce any additional distortion in the form of artifacting or anything else at all. It separation cleanliness was therefore not seen as meaningful in the context of the lower baseline method, as it performed below par on the other two metrics.

Table 4: Overview of the ratings of the methods on the listening test. The values on the left are the average scores per category, per method. The values on the right are the standard deviations over the samples per category, per method.

	Separation performance	**Separation cleanliness**	**Source preservation**	**Overall separation quality**
IBM	10 / 0	10 / 0	7.17 / 0.41	8.33 / 0.52
HP	1.67 / 1.03	10 / 0	2 / 0.89	1.67 / 1.03
RT	5 / 1.67	5.83 / 1.83	5 / 0.89	5.2 / 1.17
RT-S	5.5 / 1.22	5.5 / 1.76	5.67 / 1.37	5.67 / 0.82
2DFT	1.67 / 0.82	3.33 / 0.82	2.83 / 0.41	2.33 / 0.52
UMX	9.33 / 0.82	9 / 1.26	8.33 / 1.51	8.5 / 0.84

Moving on, the overall separation quality scores as seen in table 4 does seem to align rather well with the SI-SDR scores as seen in figure 7. UMX and the upper baseline method are closely matched, and so are REPET and REPET-SIM. Of the non-baseline methods, the 2DFT method had the worst performance. This follows expectations as well however, since 2DFT could be regarded as a baseline method by its own. The comparison between UMX and the upper baseline is especially interesting however. The IBM has a higher separation performance, although this is to be expected, since the output signals are based on the original source signals themselves. UMX is not far behind, just suffering from slight audio leakage in the output signals. UMX does impress in terms of source preservation relative to the IBM. Output signals from the IBM often tend to sound more muffled, have a muddier bass than the original source signals and suffer from some information loss after processing the target signal. These issues are less prevalent with UMX and it therefore scores higher on source preservation than the IBM. The greater amount of distortion the IBM output tends to suffer from relative to UMX is more jarring than the slight amount of leakage present in UMX. This is ultimately reflected in the respective overall separation quality scores of both the IBM and UMX. UMX is judged to overall the better audio source separation method on the SepiWoogie test set by a narrow margin, although table 4 does show that performance does appear to deviate a bit more in comparison to IBM.

Now lastly, the SI-SDR results in figure 7 already showed the unsupervised methods REPET and REPET-SIM to trail rather significantly behind UMX. This discrepancy becomes clearer during listening testing. The output from UMX can sometimes be hard to distinguish from the original source signals, whereas the output quality of both REPET and REPET-SIM is greatly lower. As stated earlier in this subsection, differences between REPET and REPET-SIM themselves as seen in table 4 are rather minimal. REPET-SIM stands out in terms of consistency when inspecting its standard deviations in table 4 and its scores in section 9.3 of the appendix, on separation performance and overall separation quality. REPET had some stand outs with separation performance relative to REPET-SIM, however it mostly compensated its separation quality through its greater lack of distortion, artifacting and information loss. REPET tends to preserve audio quality a bit better and hence making the listening experience more enjoyable. REPET-SIM on the other hand sees better separation performance in general at the expense of audio quality.

These observations do seem to align with how both methods go about performing audio source separation. REPET [8] requires periodicity and by finding it in the beat spectrum, it constructs a

repeating spectrum from a single repeating segment. This ultimately creates an inflexible, but clean filter mask. REPET-SIM [8] on the other hand has greater flexibility by using the similarity matrix to find several (different) repeating frames in an audio signal. This approach impacts separation performance for the better, however it also appears to impact the audio quality of the output.

7 Discussion

One attribute this paper attempted to emphasize several times was that the dataset which was created for this use case consisted of synthesized audio signals. With "synthesized" therefore saying that audio signals have been rendered from MIDI compositions. The emphasis is on *synthesized* Boogie Woogie piano music and not *just* Boogie Woogie piano music or Boogie Woogie piano *recordings*, since the dataset itself is actually largely not representative of real Boogie Woogie music. The reason being is the fact that Boogie Woogie music saw its greatest popularity around the period between 1930 and 1950 [9]. Recordings from that era were of significantly lower quality and contained higher amounts of noise when compared to modern-day standards. The synthesized dataset (SepiWoogie) on the other hand, is virtually flawless in terms of audio quality, since the audio signals were direct renderings from MIDI compositions.

The decision was therefore also made to avoid making any claims yet on calling this use case for instance "left and right hand audio source separation on Boogie Woogie piano recordings" in general. Doing well on the task when defined as the aforementioned would namely also imply that any methods performing well on this particular task would also perform well on real Boogie Woogie piano recordings from roughly 70 years ago. As long as performance on real Boogie Woogie piano recordings is not directly verified by means of evaluation, such a claim can therefore not be made.

This realization regarding the rather stark difference between the SepiWoogie dataset and real Boogie Woogie music set in at a late stage of the research however and thus pivoting or making other adjustments was not feasible within the time and scope of the project anymore. It should be said however, that while the specific task in this research and the sub-question itself ultimately required a different framing, the main research question was able to remain unchanged. The core of the research namely was whether on rather trivial audio source separation tasks, the effort required for developing a training set over which a supervised method could generalize over would be compensated by the increase in performance which said supervised method could gain over an unsupervised one.

In contrast, the evaluation method itself was the highlight of this research. The quantitative analysis went beyond of what is convention in the field of audio source separation. Signal-to-distortion ratio (SDR) [5] being the best global metric for performance in audio source separation is a double-edged sword. While it is useful to have a metric that gives a general impression of performance similar to what the F1-score represents for instance, (SI-)SDR [5, 19] tends to be presented as the only metric in the results of other work at times. This type of somewhat shallow evaluation is further characterized by for instance a lack of statistical testing and evaluation, and a lack of a qualitative listening test. Furthermore, it is common in audio source separation research to only present single median values, while results in figures 7, 8 and 9 in section 6 presented the entire range of scores. Such results can inform the reader on the consistency of performance (or the lack thereof) and what the respective highs and lows are of different methods. Statistical testing further added to the depth of the results, as it allowed for an understanding on whether certain observed

performance differences in the results were even statistically significant.

Lastly, the supplementary results that were obtained by the qualitative evaluation, i.e. the listening test, arguably tied everything together. There were obvious risks in regards to the bias during evaluation (as the listening tests were performed only by the author itself), or the lack of statistical significance of the results on their own. By limiting the role the listening tests to solely function as an investigative or confirming tool to the quantitative analysis, the aforementioned risks were limited. The listening test added context to the results obtained form the quantitative analysis, also answered any new and/or remaining questions after quantitative analysis and perhaps aimed to answer one of the most important questions in any audio source separation task, namely *"How good does it sound like?"* While this question might be something which is already answered by the quantitative analysis itself, this form of analysis really only manages to touch on the question *"How good will it likely sound?"* instead.

In other words, quantitative analysis in audio source separation research limits itself to estimations on performance. The field has defined metrics [5, 19] which are a good estimation on several aspects of audio quality, however it is not a true observation. Evidence to this claim can also be seen in figure 8 in section 6, where the results suggested that the lower baseline method had the greatest separation performance of all evaluated methods by having the highest median SI-SIR score. The results on the listening test painted an entirely different picture however. The lower baseline method in fact turned out to be the worst method in terms of separation performance.

8 Conclusion

To raise the research question and the sub-question from the introduction again:

- Will the labor required to build a labeled dataset for relatively a straightforward audio source separation task be compensated by the performance gain of a supervised method compared to unsupervised methods?
- How would an unsupervised audio source separation method compare against a supervised audio source separation method in this use case of playing hand separation on synthesized audio signals of Boogie Woogie piano music?

To answer the sub-question first, the results from the quantitative evaluation and the listening test have shown that training a supervised Deep Learning model to perform playing hand separation on synthesized audio signals of Boogie Woogie music performs significantly better than unsupervised audio source separation methods such as REPET [8] and REPET-SIM [8]. The supervised method, Open-Unmix [6], performs significantly better by a great margin over the unsupervised methods on every single metric. The performance of the supervised method even appears to be statistically indistinguishable from the upper baseline method at least in terms of SI-SIR, while having SI-SDR and SI-SAR scores which are not too far off from the upper baseline.

The results of the listening test also further highlighted the relatively matched performance between the upper baseline method and Open-Unmix. Released back in 2019, Open-Unmix [6] has by now been outperformed[14] by newer methods on its original task [3, 6], such as D3Net [20]. This also implies performance to already be at, or close to a theoretical limit, while Open-Unmix itself

[14] `https://paperswithcode.com/sota/music-source-separation-on-musdb18`

[6] is not even the state-of-the-art audio source separation method at the moment.

The conclusions above in turn also answer the main research question, since the great margin in performance between the supervised Deep Learning method and the unsupervised methods arguably even greatly outweighed the effort of roughly 30 hours of labor which was required for preparing the labeled training set (the remaining 22 of the 52 hours were went towards making the test and validation set). This conclusion does not necessarily imply however that even for relatively trivial audio source separation tasks such as this one, creating a labeled dataset is worth the effort. All it does imply is that this approach does seem as the best option at the time of this writing. As long as research in unsupervised audio source separation methods does not reach a level where it can perform comparatively well against task-trained supervised audio source separation methods on relatively trivial audio source separation tasks (such as the use case of SepiWoogie hand separation), the aforementioned conclusion will remain to hold.

8.1 Future Work

After further thought, the realization also occurred that the contrast in potential performance between the Deep Learning method and the unsupervised methods. This contrast exists in a large part due to the size of the training set which was made available to the Deep Learning method. Rather than attempting to compare the proverbial David against Goliath as done in this research, it might have been more insightful to evaluate performance of the Deep Learning method several times while gradually decreasing the size of the training set after each iteration of evaluation.

This in turn also allows for finding a "sweet spot" in which the trade-off between labor (hours spent preparing the training set) and resulting performance strike a strong balance. Such approach to research could arguably answer the main research question more effectively as well. The current conclusion to the research question claims that even 30 hours of labor on preparing a training set is justified, since the resulting performance is close to hitting the upper limit of performance on this task itself. What if similar performance can be achieved with only half of the training set (and thus half of the required labor)? How far would Open-Unmix's performance go when it has been trained on a smaller training set [43] consisting of 15 songs (this equates to roughly 8 hours of labor) only for instance? Addressing subquestions such as these would arguably in turn also allow the main research question to be answered in more detail.

Also one of the first points of critique mentioned in the discussion in the section above was that the SepiWoogie dataset of synthesized MIDI Boogie Woogie piano music (which is pre-separated by left and right hand) is in all likelihood not representative of the real Boogie Woogie music at large. One way to verify this assumption is by performing a listening test similar to what has been described in section 3. The Open-Unmix [6] method which has been trained on the SepiWoogie dataset would be evaluated by its observed performance to separate period-correct Boogie Woogie recordings. The results in the listening test in section 6.1 largely followed the results presented in the quantitative results in section 6. Furthermore, the listening test results would also suggest the performance of Open-Unmix [6] to be the greatest among the the evaluated methods in this research. What this also means is that any degradation in separation performance by the Open-Unmix [6] method on real Boogie Woogie recordings would be evidence for poor generalizability by the supervised method **because of** the poor real-world representation of the dataset.

One way to of course address the dataset's poor generalizability on real Boogie Woogie recordings would be by bringing the dataset itself closer to the reality of most Boogie Woogie music it was supposed to represent. This would basically mean to introduce artificial noise, artifacts and generally degrade the audio quality of the dataset on purpose. This would be a first step towards aligning the dataset closer to period-correct Boogie Woogie recordings and this could be verified by repeating the listening test which was proposed above. Only difference of course being that the Open-Unmix method would be trained on the degraded dataset instead.

Continuing on the topic of generalizability, this research arguably attempted to answer a rather broad research question through a rather narrow use case. Boogie Woogie can be seen as a sub genre that developed out of Blues and the piano style of Jazz and Blues is again just one of the many piano styles in existence. Finally the piano itself, it is also merely one of the one of the many possible instruments through which one can play tonal music. Of course, the highly varied nature of music itself would ultimately mean that unsupervised methods would likely be at a great advantage relative to supervised methods when challenged to generalize on separating bass/treble pairs of musical instruments in the broadest sense possible. However, it would be valuable to increasingly broaden the scope of the bass/treble audio source separation task from the initial starting point which has been set in this research. This would namely allow for a border between complexity and triviality to be determined.

How well will the Open-Unmix method, trained on the SepiWoogie dataset, perform relative to the unsupervised methods (2DFT, REPET and REPET-SIM) when tasked to separate music from genres related to Boogie Woogie, such as Blues, Jazz or Ragtime? How well will these same aforementioned methods perform in separating recordings from less similar piano styles, such as Baroque? Moreover, how will performance of namely the supervised method translate to a bass/treble separation task on recordings of non-piano instrument pairs (e.g. a bass guitar and an electric guitar)? Finally, how could and would any of these possible future results impact the current conclusion as seen in section 8 on the research question "will the labor required to build a labeled dataset for relatively a straightforward audio source separation task be compensated by the performance gain of a supervised method compared to unsupervised methods?". The task and its underlying use case investigated in this research arguably sit on a rather extreme end of triviality and investigating the above would also be an interesting search in finding this border between trivial and complex.

Conversely however, this research has proven that even for trivial audio source separation tasks, supervised methods currently appear to have a strong advantage over the unsupervised methods used. Looking further into better alternatives, the unsupervised ensemble-to-deep learning method by Seetharaman et. al [36] might already be an improvement over REPET [8] or REPET-SIM [8] for instance. However, it might also be interesting to gain inspiration from other fields of research in a similar way the research by Jansson et. al [27] was realized as mentioned in section 2.2 of the related work. Their adaptation of U-Net [28] enabled them perform audio source separation.

Other examples of task adaption in the field of audio processing is the work by Lu et. al [25] where timbre style transfer was investigated. Timbre style transfer is an audio processing tasks in which the aim is simply put, to transform the sound of one instrument to another while preserving the content of the original input signal. Lu et. al [25] proposed a timbre style transfer method which is essentially an adaptation on MUNIT [26], an Image-to-Image style transfer system. While at first glance the visual and auditory domain appear to be rather different, both Lu et. al [25] and Jansson et. al [27] recognized how spectrograms in the time-frequency domain could bridge

the gap between the aforementioned two domains. It is thereby for audio source separation the suggestion to also look again into the visual domain for unsupervised clustering solutions as well. This is especially interesting when considering the relatively more straightforward and less intensive process of preparing an unlabeled training set for an unsupervised method to generalize on.

References

[1] E. Cano, D. Fitzgerald, A. Liutkus, M. Plumbley, and F.-R. Stöter, "Musical source separation: An introduction," *IEEE Signal Processing Magazine*, vol. 36, 01 2018.

[2] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Other," 2007. [Online]. Available: https://doi.org/10.7910/DVN/ZVU9HF

[3] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[4] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "Wham!: Extending speech separation to noisy environments," in *Proc. Interspeech*, Sep. 2019.

[5] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[6] F.-R. St
"oter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix - a reference implementation for music source separation," *Journal of Open Source Software*, 2019. [Online]. Available: https://doi.org/10.21105/joss.01667

[7] P. Seetharaman, F. Pishdadian, and B. Pardo, "Music/voice separation using the 2d fourier transform," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 36–40.

[8] Z. Rafii, A. Liutkus, and B. Pardo, *REPET for Background/Foreground Separation in Audio*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 395–411. [Online]. Available: https://doi.org/10.1007/978-3-642-55016-4_14

[9] P. J. Silvester, *The Story of Boogie-Woogie: A Left Hand Like God*, 2009, no. 2.

[10] E. H. Newberger, "Archetypes and antecedents of piano blues and boogie woogie style," *Journal of Jazz Studies*, no. 4, pp. 84–109, 1976.

[11] W. Logan, "The ostinato idea in black improvised music: A preliminary investigation," *The Black Perspective in Music*, no. 12, pp. 193–215, 1984.

[12] S. Ouelha, S. Touati, and B. Boashash, "An efficient inverse short-time fourier transform algorithm for improved signal reconstruction by time-frequency synthesis: Optimality and computational issues," *Digital Signal Processing*, vol. 65, pp. 81–93, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105120041730057X

[13] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, p. 1256–1266, Aug 2019. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2019.2915167

[14] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *CoRR*, vol. abs/1806.03185, 2018. [Online]. Available: http://arxiv.org/abs/1806.03185

[15] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," 2020.

[16] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," 2020.

[17] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, "Sandglasset: A light multi-granularity self-attentive network for time-domain speech separation," 2021.

[18] D. Wang, *On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis*. Boston, MA: Springer US, 2005, pp. 181–197. [Online]. Available: https://doi.org/10.1007/0-387-22794-6_12

[19] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr - half-baked or well done?" 2018.

[20] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," 2021.

[21] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and state-of-the art music source separation tool with pre-trained models," 2019.

[22] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," 2020.

[23] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," 2021.

[24] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," 2019.

[25] C.-Y. Lu, M.-X. Xue, C.-C. Chang, C.-R. Lee, and L. Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," 2018.

[26] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," 2018.

[27] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," October 2017. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/19289/

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[29] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5069–5073.

[30] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," 2018.

[31] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," 2017.

[32] E. Pierson Lancaster and N. Souviraà-Labastie, "A frugal approach to music source separation," Nov. 2020, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02986241

[33] P. Seetharaman, G. Wichern, J. L. Roux, and B. Pardo, "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," 2018.

[34] Y. Bando, Y. Sasaki, and K. Yoshii, "Deep bayesian unsupervised source separation based on a complex gaussian mixture model," 2019.

[35] T. Virtanen, *Unsupervised Learning Methods for Source Separation in Monaural Music Signals*. Boston, MA: Springer US, 2006, pp. 267–296. [Online]. Available: https://doi.org/10.1007/0-387-32845-9_9

[36] P. Seetharaman, G. Wichern, J. L. Roux, and B. Pardo, "Bootstrapping deep music separation from primitive auditory grouping principles," 2019.

[37] A. Hadjakos, S. Waloschek, and A. Leemhuis, "Detecting hands from piano midi data," in *Mensch und Computer 2019 - Workshopband*. Bonn: Gesellschaft für Informatik e.V., 2019.

[38] E. Nakamura, Y. Saito, and K. Yoshii, "Statistical learning and estimation of piano fingering," *CoRR*, vol. abs/1904.10237, 2019. [Online]. Available: http://arxiv.org/abs/1904.10237

[39] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, "Unsupervised transcription of piano music," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 1538–1546. [Online]. Available: http://papers.nips.cc/paper/5432-unsupervised-transcription-of-piano-music.pdf

[40] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 121–124.

[41] E. Manilow, P. Seetharaman, and B. Pardo, "The northwestern university source separation library." in *ISMIR*, 2018, pp. 297–305.

[42] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," 2018.

[43] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning - the good, the bad and the ugly," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

9 Appendix

9.1 Code and links

Running training and testing of UMX on the SepiWoogie dataset

```
python3 train.py --dataset trackfolder_fix --root ./audio
--dataset trackfolder_fix --target-file left.wav --interferer-files right.wav
--batch-size 1 --nfft 2048 --nhop 512 --hidden-size 128 --nb-channels 1

python3 alt-test.py input ./audio/test/test/mixture.wav
--model ./open-unmix --targets ['left', 'right']
```

Links to code, samples and the dataset:

- Link to custom UMX test code: `https://github.com/bosnyan/open-unmix-pytorch`
- Link to evaluation and statistical analysis code: `https://github.com/bosnyan/SepiWoogieEval`
- Link to samples used in the listening test: `https://www.dropbox.com/sh/bmi3ziyxgwwc1nm/AACipL0jK5R8tQP9iU2GTi5la?dl=0`
- Link to the SepiWoogie dataset: `https://doi.org/10.34740/kaggle/dsv/2099606`

9.2 P-Values

Table 5: Obtained p-values from paired significance tests between the different methods on the distributions of achieved SI-SDR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		4.61e-24	3.06e-24	1.73e-06	9.69e-26	8.02e-05
HP	4.61e-24		7.38e-01	1.89e-04	9.64e-01	5.82e-20
RT	3.06e-24	7.38e-01		3.52e-06	7.62e-01	8.26e-20
RT-S	1.73e-06	1.89e-04	3.52e-06		3.52e-06	1.73e-06
2DFT	9.69e-26	9.64e-01	7.62e-01	3.52e-06		6.40e-21
UMX	8.02e-05	5.82e-20	8.26e-20	1.73e-06	6.40e-21	

Table 6: Obtained p-values from paired significance tests between the different methods on the distributions of achieved SI-SIR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		6.74e-02	2.84e-05	2.13e-06	4.07e-05	5.86e-01
HP	6.74e-02		8.92e-05	7.69e-06	1.15e-04	3.93e-01
RT	2.84e-05	8.92e-05		1.78e-01	2.99e-01	1.89e-04
RT-S	2.13e-06	7.69e-06	1.78e-01		7.66e-01	1.97e-05
2DFT	4.07e-05	1.15e-04	2.99e-01	7.66e-01		1.61e-04
UMX	5.86e-01	3.93e-01	1.89e-04	1.97e-05	1.61e-04	

Table 7: Obtained p-values from paired significance tests between the different methods on the distributions of achieved SI-SAR results.

	IBM	**HP**	**RT**	**RT-S**	**2DFT**	**UMX**
IBM		4.72e-24	1.73e-06	1.73e-06	1.73e-06	3.52e-06
HP	4.72e-24		5.17e-01	1.74e-04	9.26e-01	1.73e-06
RT	1.73e-06	5.17e-01		2.88e-06	2.29e-01	1.73e-06
RT-S	1.73e-06	1.74e-04	2.88e-06		3.88e-06	1.73e-06
2DFT	1.73e-06	9.26e-01	2.29e-01	3.88e-06		1.73e-06
UMX	3.52e-06	1.73e-06	1.73e-06	1.73e-06	1.73e-06	

9.3 Listening Test results

Table 8: Observed performance scores on the different criteria across the different methods on "1. RobRio - R3OCKHOU".

1	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	Sounds clearly muddy/muffled to the original.	7	Muffled audio did really affect the perceived audio quality.	8
HP	It basically separated the low-end of the bass source and classified the remaining signal (of both sources) as right-hand signal.	1	No additional artifacts from separation. It is clean.	10	Right hand signal of course has perfects preservation of the original signal. However, this is not an entirely fair observation, since the left source signal is largely missing.	2	Output is not usable at all. Better off doing manual pitch filtering instead.	1
RT	Left source is not very well separated, however the right source sounds better.	5	There is of course some leakage from the non-target signals in the target signal, however these do not sound distorted themselves nor are there any audible artifacts.	8	Output signal sounds very muffled. Pretty poor preservation of the target signals.	4	Somewhat usable. It gets outshone by REPET-SIM however.	5
RT-S	Similar story to REPET, albeit slightly better performance overall.	6	There is of course some leakage from the non-target signals in the target signal, however these do not sound distorted themselves nor are there any audible artifacts.	8	Pretty OK preservation. Audio is leaking in both directions however (target signals overlapping each other).	6	Pretty respectable performance. Slightly better than REPET overall.	6
2DFT	There was SOME separation. The bass signal sounded more or less like the original mix input however.	2	The audio sounds distorted and the excessive amount of leakage does not help either.	5	Both sources mix into each other, leading to poor preservation overall.	3	Generally just bad. Unusable in any use case.	2
UMX	Very slight leakage of the other source. It is barely noticable in fact.	9	No artifacts or distortion at all.	10	It sounds pretty much as close as it currently seems to be possible to the original.	10	The output sounds excellent. Nothing to note aside that it suffers from the occasional minor leaks	9

Table 9: Observed performance scores on the different criteria across the different methods on "3. TerryButters - shuffle boogie with trad changes (2)".

2	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	Output sounds somewhat off, distorted and muffled.	7	Great separation performance. Information loss ruins the perceived quality however.	8
HP	Left track barely has any audio, only the lower end/part of the supposed output signal.	1	No additional artifacts from separation. It is clean.	10	Barely anything remained in the left signal output	1	Output is not usable at all. Better off doing manual pitch filtering instead.	1
RT	Separation performance is generally just insufficient, however it really the best performing unsupervised method on this particular sample.	4	Distortion present in both output tracks. Slightly less bad compared to REPET-SIM's performance.	4	Similar to REPET-SIM; audio sounds hollow, muffled and distorted. It is not as bad as REPET-SIM however	5	Somewhat usable performance when no alternatives are available. It beats out REPET-SIM by a slight margin.	5
RT-S	Poor separation performance in general. Sounds like a slightly better version of 2DFT,	3	Significant distortion present in both output tracks.	3	Audio sounds hollow, muffled and distorted.	4	Similar to REPET in terms of general performance, just slightly worse in general.	4
2DFT	Poor separation. The left output tracks sounds more or less like the original mix, while the right output track sounds like a low-quality version of the original mix signal.	2	Both the intended source track audio as well as the leaked-in signal on both tracks suffer from great distortion.	3	The left track output signal seems to have reasonably good sound preservation. This becomes less impressive when realizing the performance on the right track signal output is horrible instead.	3	Slightly better than the baseline, but still far from usable.	3
UMX	Good separation. Some slight leakage however, especially in the left output track.	8	Rather considerable amount of distortion present, especially on the left track of the output.	7	Source signals sound somewhat hollow and distorted.	6	Pretty O.K. performance, but the flaws which are present are definitely noticeable.	7

Table 10: Observed performance scores on the different criteria across the different methods on "4. BitMidi - Cadillac-Avenue-Boogie".

3	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	Sounds slightly different from the original source audio, however this is hardly noticeable.	8	Near perfect performance.	9
HP	Left track barely has any audio, only the lower end/part of the supposed output signal.	1	No additional artifacts from separation. It is clean.	10	The right-hand output signal of course has perfect preservation, however there is barely anything left in the left-hand output signal.	2	Output is not usable at all. Better off doing manual pitch filtering instead.	1
RT	Decent separation, especially on te right-hand output signal. However, performance should still be seen as insufficient, since the left-hand performance leaves a lot to be desired.	4	While there was leakage, the leakage itself wasn't too bothering however, since there was minimal distortion.	6	Slight muffling of the bass in the left-hand output signal. Other than that not much else which was very noticeable.	6	Right-hand output signal is OK, the left-hand signal is not usable however.	4
RT-S	Very similar to REPET in most occassions. It does outperform REPET in terms of general separation quality in some instances. This in the end helps to elevate performance to sufficient levels.	6	The leaking audio signal sounded rather distorted.	4	While there is audio leakage, there does not seem to be any audible signal loss in either output sources.	8	A generally OK option when no supervised learning model is available.	6
2DFT	Right-hand output signal has OK-ish separation. Left sounds very mixed however, i.e. bad.	3	Great amount of distortion in general. Both in leaked audio as well as source audio.	3	Both sources have poor preservation of the original sound.	2	The output is not usable in any imaginable use case.	3
UMX	Near-perfect separation, just some slight leakage.	9	The audio which did leak sounded slightly distorted.	8	Original sources sound well-preserved without any clear signs of audio distortion.	10	Near perfect performance.	9

Table 11: Observed performance scores on the different criteria across the different methods on "5. MIDI-Pianos - (1)-Boogie".

4	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	It sounds muffled and there are signs of significant loss.	7	Great separation, rather disappointing amounts of distortion.	8
HP	Hardly constitutes as audio source separation. It shows that mere frequency filtering does not work.	1	No additional artifacts from separation. It is clean.	10	Right-hand signal of course has perfect preservation, however the same cannot be said regarding the left-hand signal.	1	Not usable.	1
RT	It is an attempt at separation, however it never sounded good enough to be considered passable.	3	Significant amounts of distortion coming from the leaking audio.	4	The left-hand output signal sounds rather difficult to listen to due to the great amount of distortion present there. The right hand signal does sound considerably better however.	4	Right-hand output signal is OK, the left-hand signal is not usable however.	4
RT-S	Surprisingly good performance, although it does tend to drop off a bit at times.	6	Some artifacting coming from leaking audio due to distortion.	6	Pretty good signal preservation in general, however the right-hand output signal does sound significantly worse.	6	A generally OK option when no supervised learning model is available.	6
2DFT	Separation barely took place. The right-hand output signal sounds like a low-quality version of the left-hand output signal.	1	Both the intended source track audio as well as the leaked-in signal on both tracks suffer from great distortion.	3	The left track output signal seems to have reasonably good sound preservation. This becomes less impressive when realizing the performance on the right track signal output is horrible instead.	3	Just about better than the high/lowpass filter. The output is otherwise not usable in any imaginable use case.	2
UMX	Nearly perfect. Slight hints of audio leakage. This is hardly noticeable however.	10	Ever-so-slight artifacting from the remaining signal.	9	Sounds slightly off, nothing too bad however. Audio is muffled and there is some distortion.	8	Great performance, general impressions are even better than the benchmark.	9

Table 12: Observed performance scores on the different criteria across the different methods on "7. AlLevy - 3blind".

5	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	It sounds muffled and there are signs of significant loss.	7	Great separation, rather disappointing amounts of distortion.	8
HP	Separation performance in insufficient, albeit surprising considering its crudeness.	3	No additional artifacts from separation. It is clean.	10	Subpar preservation performance. Almost impossible to recognize the left-hand output signal as a piano.	3	Not usable.	3
RT	Separation on the left-hand output signal sounds better than REPET-SIM. Other than that it is hard to distinguish the two on this sample.	7	No artifacts at all really. The leaky audio sounded clean, especially on the left-hand output signal. The right side might be a bit less convicing.	8	Not all information was kept after separation. Especially the left-hand output signal sounds rather boomy, along with some distortion as well.	6	Surprisingly good quality.	7
RT-S	Decent separation, it could have been better on the left-hand output signal however.	6	Some clear signs of distortions, especially throughout the left-hand output signal.	6	Similar performance to REPET. Left-hand output signal sounds even boomier however.	5	Passable. It has some more noticeable flaws in comparison to REPET however.	6
2DFT	Separation barely took place. The right-hand output signal sounds like a low-quality version of the left-hand output signal.	1	Both the intended source track audio as well as the leaked-in signal on both tracks suffer from great distortion.	3	The left track output signal seems to have reasonably good sound preservation. This becomes less impressive when realizing the performance on the right track signal output is horrible instead.	3	Just about better than the high/lowpass filter. The output is otherwise not usable in any imaginable use case.	2
UMX	Nearly perfect. Slight hints of audio leakage. This is hardly noticeable however.	10	Ever-so-slight artifacting from the remaining signal.	10	There is some distortion and it can get especially noticeable in the right-hand output signal.	8	Great performance, general impressions are even better than the benchmark.	9

Table 13: Observed performance scores on the different criteria across the different methods on "10. RTpress - piggy_sk".

6	Separation performance	Rating	Separation cleanliness	Rating	Sound preservation	Rating	Overall separation quality	Rating
IBM	Total separation.	10	No additional artifacts from separation. It is clean.	10	It sounds muffled and there are signs of significant loss.	7	Great separation, rather disappointing amounts of distortion.	9
HP	Separation performance in insufficient, albeit surprising considering its crudeness.	3	No additional artifacts from separation. It is clean.	10	Subpar preservation performance. Almost impossible to recognize the left-hand output signal as a piano.	3	Not usable.	3
RT	Generally good separation performance, especially the right-hand output signal stands out.	7	Presence of leaky audio, especially in the right-hand output signal sounds distorted.	5	Sounds very muffled on the left-hand output signal.	5	General passable allround performance.	6
RT-S	Separation performance on the right-hand output signal is pretty good, the left side disappoints however.	6	Generally passable. The leaky high notes in the left-hand output signal can be annoying though.	6	Higher notes appear to be rather distorted.	5	Overall quality has greater peaks and lows relative to REPET.	6
2DFT	Separation barely took place. The right-hand output signal sounds like a low-quality version of the left-hand output signal.	1	Both the intended source track audio as well as the leaked-in signal on both tracks suffer from great distortion.	3	The left track output signal seems to have reasonably good sound preservation. This becomes less impressive when realizing the performance on the right track signal output is horrible instead.	3	Just about better than the high/lowpass filter. The output is otherwise not usable in any imaginable use case.	2
UMX	Nearly perfect. Slight hints of audio leakage. This is hardly noticeable however.	10	No additional artifacts from separation. It is clean.	10	There is some distortion and it can get especially noticeable in the right-hand output signal.	8	Great performance, general impressions are even better than the benchmark.	9