Transactions of the International Society for Music Information Retrieval

Fabbro, G., Uhlich, S., Lai, C., Choi, W., Ramirez, M., Liao, W., Gadelha, I., Ramos, G., Hsu, E., Rodrigues, H., Stöter, F., Défossez, A., Luo, Y., Yu, J., Chakraborty, D., Mohanty, S., Solovyev, R., Stempkovskiy, A., Habruseva, T., Goswami, N., Harada, T., Kim, M., Lee, J., Dong, Y., Zhang, X., Liu, J., and Mitsufuji, Y. (2023). The Sound Demixing Challenge 2023 – Music Demixing Track, *Transactions of the International Society for Music Information Retrieval*, V(N), pp. xx–xx, DOI: https://doi.org/xx.xxxx/xxxx.xx

~~OVERVIEW ARTICLE~~

The Sound Demixing Challenge 2023 – Music Demixing Track

Giorgio Fabbro[*], Stefan Uhlich[*], Chieh-Hsin Lai[†], Woosung Choi[†], Marco Martínez-Ramírez[†], Weihsiang Liao[†], Igor Gadelha[‡], Geraldo Ramos[§], Eddie Hsu[‡], Hugo Rodrigues[§], Fabian-Robert Stöter[¶], Alexandre Défossez[||], Yi Luo[**], Jianwei Yu[**], Dipam Chakraborty[††], Sharada Mohanty[‡‡], Roman Solovyev[§§], Alexander Stempkovskiy[§§], Tatiana Habruseva[¶¶], Nabarun Goswami[***], Tatsuya Harada[***†††], Minseok Kim[‡‡‡], Jun Hyung Lee[‡‡‡], Yuanliang Dong[§§§], Xinran Zhang[§§§], Jiafeng Liu[§§§], and Yuki Mitsufuji[†]

Abstract

This paper summarizes the music demixing (MDX) track of the Sound Demixing Challenge (SDX'23). We provide a summary of the challenge setup and introduce the task of robust music source separation (MSS), i.e., training MSS models in the presence of errors in the training data. We propose a formalization of the errors that can occur in the design of a training dataset for MSS systems and introduce two new datasets that simulate such errors: *SDXDB23_LabelNoise* and *SDXDB23_Bleeding*[1]. We describe the methods that achieved the highest scores in the competition. Moreover, we present a direct comparison with the previous edition of the challenge (the Music Demixing Challenge 2021): the best performing system under the standard MSS formulation achieved an improvement of over 1.6dB in signal-to-distortion ratio over the winner of the previous competition, when evaluated on MDXDB21. Besides relying on the signal-to-distortion ratio as objective metric, we also performed a listening test with renowned producers/musicians to study the perceptual quality of the systems and report here the results. Finally, we provide our insights into the organization of the competition and our prospects for future editions.

Keywords: Music Source Separation, Deep Learning, Neural Networks, Robust Training, Sound, Signal Processing

1. Introduction

Audio source separation has a long history in research, partially motivated by the many applications it enables. Thanks to source separation, music producers and artists are able to make use of material that was created decades ago or under limited recording conditions. Movie production studios have now the possibility to revive old classics[2], for which only single-channel master tracks exist, and bring them back to the theatre taking advantage of newer innovations, such as spatial audio. In the future, people who are experiencing hearing difficulties, a condition which makes it challenging for them to communicate in loud and noisy surroundings, will be able to successfully engage in conversations.

[*]Sony Europe B.V., Stuttgart, Germany
[†]Sony Group Corporation, Tokyo, Japan
[‡]Moises.ai, João Pessoa, Brazil
[§]Moises.ai, Salt Lake City, USA
[¶]AudioShake, San Francisco, USA
[||]Meta AI, Paris, France
[**]Tencent AI Lab, Shenzhen, China
[††]AIcrowd, Bengaluru, India
[‡‡]AIcrowd, Lausanne, Switzerland
[§§]Institute for Design Problems in Microelectronics of Russian Academy of Sciences, Moscow, Russian Federation
[¶¶]Independent researcher, Cork, Ireland
[***]The University of Tokyo, Japan
[†††]RIKEN, Japan
[‡‡‡]Korea University, Seoul, South Korea
[§§§]Central Conservatory of Music, Beijing, China

In the context of research, the more recent success of audio source separation should be attributed also to the presence of benchmarks that allowed different methods to be effectively compared: one example is the MUSDB18 dataset (Rafii et al., 2017), introduced during the Signal Separation Evaluation Campaign (SiSEC) in 2018 (Stöter et al., 2018). The dataset includes 150 different songs, along with corresponding separate recordings for each musical instrument. These tracks were grouped into four classes for consistency across songs: *vocals*, *bass*, *drums*, and a final class named *other* for any other instrument present. This allowed participants to carry out training and evaluation of their models on a standardized and consistent pool of data.

In 2021 we continued the tradition of SiSEC competitions with the Music Demixing Challenge (MDX'21) (Mitsufuji et al., 2022). We decided to keep MUSDB18 as the reference dataset for training, but introduced a new benchmark for testing based on data that could not be accessed by the participants. The submissions to the MDX'21 challenge were evaluated on a new hidden dataset called MDXDB21 (Mitsufuji et al., 2022), which contained 30 songs produced by Sony Music Entertainment Japan for the purpose of the challenge.

Two years later, we organized a new edition of the challenge, named Sound Demixing Challenge 2023 (SDX'23): while MDX'21 focused exclusively on music source separation, SDX'23 presented two independent tracks, one for music source separation (MSS) and one for cinematic sound separation (Uhlich et al., 2023). In the music track, we again used MDXDB21 as test benchmark. This allowed a comparison of the submissions across the two editions of the challenge.

While we wanted to offer the prospective participants a familiar research playground (e.g., keeping the usual four instrument classes introduced by MUSDB18), we complemented that with a novel aspect of the source separation problem: *robust music source separation*. In robust MSS the trained system needs to be able to handle errors and inconsistencies in the training data. We decided to focus one part of SDX'23 on this topic after we analyzed the outcome of the previous edition of the challenge.

At the end of MDX'21 it was evident that the volume of data accessible for training can have a significant benefit on the quality of the final source separation system. At the same time, since state-of-the-art methods for source separation are still dominated by supervised learning approaches, the availability of appropriate data is limited. This formulation requires individual tracks, each containing the signal of a single instrument, which are difficult to obtain, especially in large quantities. Simulation of the recordings using MIDI and virtual instruments (e.g., the Slakh dataset (Manilow et al., 2019)) can be a compromise: the size of the dataset grows very easily, but the results lack realism and have some critical limitations (e.g., they often do not contain vocals).

In the context of the challenge, rather than tackling the problem of the scarcity of the data, we assumed that such data would be available and looked further down the road: is having more songs enough to obtain better models? How can we best leverage and curate a large corpus of tracks to improve the quality of the separation model?

A set of internal experiments revealed that training source separation models on a large volume of high-quality data does not guarantee better network convergence. The convergence behavior of our models was dramatically affected by sparse errors in the data. These were mostly related to the ground truth labels used when training (i.e., the identity of the instruments present in each audio recording). For example, tracks labeled as *vocals* actually contained the signal of a guitar. Intuitively, the impact of such sparse errors should decrease proportionally to the amount of data used to train the model. Against our expectations, increasing the amount of data was causing the models to stop converging. Only through an expensive activity of data cleaning we were able to make the model converge again.

The process of cleaning a dataset is very expensive and does not scale easily with its size: manually checking the annotations for a large amount of tracks is not feasible. While automatic methods can provide an initial solution (e.g., audio classification and tagging models (Garcia et al., 2021; Mahanta et al., 2021; Kong et al., 2020)), some errors are intrinsically difficult to fix: if a single track erroneously contains the signal of two different instruments, simply changing its label will not solve the problem. Source separation methods can also be used to exclude noisy samples in a large dataset, to ensure that the new model is trained only on clean data (Rouard et al., 2023).

A more interesting solution would be to make the training of the network invariant to these errors. Ideally, if the learning process is robust to such inconsistencies, adding new data upon availability becomes easier, as we can avoid a cleaning activity that is expensive, likely incomplete and potentially ineffective. The research community has proposed some robust training methods to address label noise in classification problems (Han et al., 2018; Li et al., 2020; Cheng et al., 2020; Wang et al., 2022) or handle out-of-distribution samples (Lai et al., 2019, 2023; Hendrycks et al., 2021; Mukherjee et al., 2021). However, to the best of our knowledge no existing approach explicitly tackles audio source separation[3]. With the SDX'23 challenge we aimed to bring the attention of the research community to the topic of robust MSS and provided the participants with an environment that presented the issues outlined above, even when working on small datasets.

In this paper we summarize the music track of SDX'23: we show the challenge setup in Sec. 2, we introduce the topic of robust music source separation in Sec. 3, we outline the parts of the challenge related to the standard formulation of MSS in Sec. 4, we present the challenge results, together with the descriptions of the winning approaches in Sec. 5 and 6 and we elaborate on the technical challenges in the organization of the competition in Sec. 7.

2. MDX Challenge Setup

In the following, we summarize the structure of the competition. Similarly to the previous edition, the challenge was hosted on AIcrowd[4].

2.1 Task Definition

Participants in the music track (MDX) of the Sound Demixing Challenge 2023 were asked to submit systems to separate a stereo song $\mathbf{x}(n) \in \mathbb{R}^2$ into one stereo track for *vocals* ($\mathbf{s}_\mathrm{V}(n) \in \mathbb{R}^2$), one for *bass* ($\mathbf{s}_\mathrm{B}(n) \in \mathbb{R}^2$), one for *drums* ($\mathbf{s}_\mathrm{D}(n) \in \mathbb{R}^2$) and one for *other* ($\mathbf{s}_\mathrm{O}(n) \in \mathbb{R}^2$), where the song can be obtained as:

$$\mathbf{x}(n) = \mathbf{s}_\mathrm{V}(n) + \mathbf{s}_\mathrm{B}(n) + \mathbf{s}_\mathrm{D}(n) + \mathbf{s}_\mathrm{O}(n), \tag{1}$$

and n denotes the time index. All signals are sampled at 44100Hz.

2.2 Leaderboards

The previous edition of the MDX challenge (Mitsufuji et al., 2022) focused on the standard formulation of music source separation. This year, we devoted two of three leaderboards to the issue of training source separation models with data containing errors and inconsistencies. We elaborate on this concept in Section 3 and name this task *robust music source separation*.

On top of that, we provided a third leaderboard that was devoted to the standard music source separation task, without any constraint on the training data. By allowing training under any condition, we are interested in tracking the progress of the source separation community.

In summary, the submissions were categorized under the following three leaderboards:

- *Leaderboard A* was designated for models trained on data suffering from *label noise*,
- *Leaderboard B* was designated for models trained on data suffering from *bleeding*,
- *Leaderboard C* was designated for models trained on any data.

For the definitions of *label noise* and *bleeding*, we refer the reader to Sec. 3.2.

2.3 Ranking Metric

The evaluation of the systems followed the same strategy as in MDX'21: we used the global *signal-to-distortion ratio* (SDR) as metric, which is defined for one song as

$$\mathrm{SDR} = \frac{1}{4}\Big(\mathrm{SDR}_\mathrm{V} + \mathrm{SDR}_\mathrm{B} + \mathrm{SDR}_\mathrm{D} + \mathrm{SDR}_\mathrm{O}\Big), \tag{2}$$

with

$$\mathrm{SDR}_c = 10\log_{10}\frac{\sum_n \|\mathbf{s}_c(n)\|^2}{\sum_n \|\mathbf{s}_c(n) - \hat{\mathbf{s}}_c(n)\|^2}, \tag{3}$$

where $\mathbf{s}_c(n) \in \mathbb{R}^2$ and $\hat{\mathbf{s}}_c(n) \in \mathbb{R}^2$ denote the stereo target and estimate for source $c \in \{\mathrm{V}, \mathrm{B}, \mathrm{D}, \mathrm{O}\}$. Finally, the global SDR of (2) is averaged over all songs in the hidden test dataset to obtain the final score.

In addition to the objective metric above, this edition of the challenge also featured a subjective evaluation based on a listening test carried out on the estimates of the systems that achieved the highest SDR scores in Leaderboard C. We refer the reader to Sec. 4.2 and Sec. 6 for details on the subjective evaluation.

2.4 Timeline

The challenge featured two rounds. Phase I started on January 23$^{\mathrm{rd}}$ 2023, while Phase II started on March 6$^{\mathrm{th}}$ 2023. Due to the submission system experiencing difficulties in handling the surge in the number of submissions towards the end of the challenge, the end date of Phase II was extended by one week. Originally scheduled to conclude on May 1st, 2023, the challenge was extended to May 8, 2023, to ensure a fair competition for all teams. A *warm-up round* was also organized, which began on December 8$^{\mathrm{th}}$ 2022 and lasted until the beginning of Phase I. During this round, participants could get acquainted with the submission system and prepare their submissions for the challenge.

The challenge evaluation was carried out on the same hidden test set used in the previous edition: each submission was used to separate 27 songs from MDXDB21 and its objective performance was computed. Three songs were held out from the dataset to carry out the evaluation: two of them were used to give feedback to the participants about their performance; the third one was excluded since the bass track is silent. During Phase I, the scores available to the participants were computed on one third of MDXDB21 (nine songs). During Phase II, nine more songs were added to the scores visualized on the leaderboards. Once the competition ended and the participants could not submit anymore to be eligible for prizes, the final scores computed on the whole test set were displayed in the leaderboards. This strategy is the same as the one used in MDX'21.

We choose to carry out the evaluation in this way in order to prevent participants from implicitly adapting their submissions to the test set throughout the course of the challenge. Nevertheless, this encouraged participants to maximize the number of submissions during Phase II, so that they would increase the likelihood of *indirectly* adapting their system to the test set. For this reason, at the end of Phase II we asked each participant in every leaderboard to *manually* select three

candidate submissions that would move on to the final evaluation: only these submissions were then evaluated on the whole test set. The best out of three was then displayed as final entry for each team.

3. Robust Music Separation

From the results of the last edition of the challenge (Mitsufuji et al., 2022) it became clear to the source separation community that the performance of a model very often correlates with the amount of data used to train it. One example was the participant *defossez*: his model trained only on MUSDB18 achieved an average SDR score of 7.32dB[5], while training on additional data improved its performance by almost 1dB[6].

It would be safe to assume that increasing the amount of data used for training increases the performance of the model, but this is not necessarily the case. While performing experiments using internal data, we experienced that models whose validation loss used to converge when trained on a small high-quality dataset were not converging anymore when trained on a high-quality dataset one order of magnitude larger (see Figure 1). After careful inspection, we realized how sparse errors in such a large pool of data were responsible for this phenomenon.

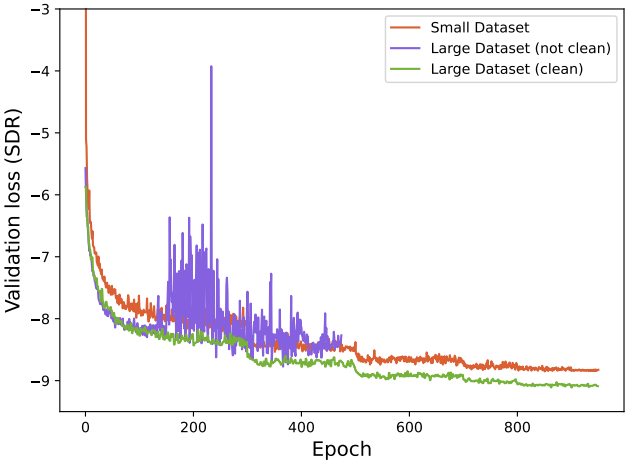


Figure 1: Comparison of validation loss when training the same model on a small dataset (red), a large dataset with errors (purple) and the same large dataset once the errors have been corrected (green). All experiments were evaluated on the same validation set.

3.1 Why Robust Music Separation?

Intuitively, the more data we have, the lower the impact of any incorrect recording should be. However, increasing the amount of data also increases the number of incorrect recordings and the likelihood that globally the dataset presents inconsistencies. Examples of such errors can be recordings where the recorded instrument does not match the instrument label, or individual recordings that contain the signal of more than one instrument. As Figure 1 shows, only through an expensive activity of *data cleaning* we were able to make those models converge again, confirming that those sparse errors were the cause of the issue.

Given how time-consuming and resource-expensive a data cleaning activity is, we decided to focus the research community on this issue during the Sound Demixing Challenge 2023. Our intention was to constrain the participants in the data at their disposal, so as to make them devise strategies that would allow training models *robust* to inconsistencies in the training data.

3.2 Formalizing Errors in the Training Data

We aimed at providing a common framework to characterize errors and inconsistencies in the training data from the perspective of audio source separation. We assumed that the training of a source separation model requires individual stems and does not rely exclusively on mixtures. Under these assumption, the integrity of the individual recordings becomes paramount. We identified two categories of errors that can occur:

- the recorded instrument and the label identifying it do not agree,
- the recording contains signals belonging to more than one instrument.

We named the first category **label noise**, since a (possibly) random change in the label of the recording is responsible for it; we named the second category **bleeding**, as it occurs when the signal of a second instrument *bleeds* into the recording of the first one. The leaderboards A and B in the challenge were dedicated to the two error categories respectively.

Label Noise Being a creative process, music production does not follow conventional workflows. Different music producers adhere to different conventions for their procedures, which has direct consequences in how their final deliverable is organized. Often producers are asked to deliver not only the final mix of their song, but also individual stems: these tracks usually group multiple recordings that belong to a single identifiable instrument (e.g., all microphones in the drum set). The choice of which stems to deliver and how to name them is typically influenced by the preferences and background of the producer and by the style of the song. In the context of music production there is no immediate benefit in following a strict convention when naming the stems.

When we repurpose these recordings for training source separation models, we need to collect stems coming from different music producers. For one stem, the only information we require is the identity of the recorded instrument. Typically, the only label we possess that indicates the instrument identity in a recording is its file name. Some producers follow more orga-

nized workflows and might keep some extra metadata, but this does not hold for all of them. The stem file name is chosen manually: a process that is prone to errors and not conforming to any naming convention. The result is a very large collection of valuable audio recordings and a chaotic set of instrument names.

Even if we were able to efficiently collapse all the variations of one instrument name (e.g., *el_guitar*, *electric_guitar*, *el_gtr*, etc...), we would still have some which are intrinsically ambiguous. Some examples are: *lead* (which could refer to vocals, guitar, or any other instrument playing a leading part), *choir* (which could be a human choir or a synthesizer imitating a choir), *bells* (which could refer to church bells, chime bells or a synthesizer sound), attributes used as names (such as *clean*, *dark*, *bright*), *sfx* (which could potentially refer to any sound), and others.

Bleeding When producing music in a studio, the priority is given to the performance. Each performer must be in the condition of delivering the perfect rendition of the artist's vision. One example is for the whole band to play together, instead of recording one instrument at a time. This improves the musical interpretation of the song, as each musician can directly react to small changes in the performance of the others: this leads to increased interactions that translate into a more truthful and lively recording. Recording studios are designed to maximize acoustic isolation between different rooms and booths, but there are limits. For example, low frequencies are notoriously difficult to isolate, due to their long wavelength: a bass amplifier produces sound that can overcome the isolation barriers in the studio and reach the microphones devoted to other instruments. If those instruments are recorded at the same time, some signal from the bass amplifier will *bleed* into their tracks. This is not an issue in the context of the song production, as those signals will be summed anyway to produce the final mix. But if these recordings are now repurposed as training material for a source separation system, this bleeding becomes problematic.

3.3 Creating Datasets with Corruptions

In the context of the challenge, we wanted participants to tackle the issues above while ensuring that the overall competition remained fair when systems are evaluated against a common test set. Ideally, we would force the participants to train their systems using the same datasets, which have been corrupted using label noise and bleeding. On top of that, we would need to prevent access to an error-free version of those datasets.

For both categories of errors we have the option of simulating them on existing recordings. Although simulation might result in a loss of realism, it preserves the conditions required during the setup of the challenge and does not change the underlying task that needs to be solved. For this reason, we decided to simulate both label noise and bleeding starting from error-free recordings.

Although clean recordings are easy to find in the community, we needed to avoid participants having access to the source material, as it would have given them an unrealistic advantage. For this reason, we made use of MoisesDB (Pereira et al., 2023), which was not released until after the end of the challenge. It contains 240 individual tracks sourced from 45 diverse artists, spanning twelve distinctive musical genres. Each track is broken down into its constituent audio components and is categorized within a hierarchical, two-tier taxonomy of stems.

We selected 203 songs from MoisesDB to be the source data for our corrupted datasets. Then, we generated two versions of such data, one containing label noise (*SDXDB23_LabelNoise*), another containing bleeding (*SDXDB23_Bleeding*), and shared them with the participants. The new datasets follow the same structure and design of MUSDB18 and MDXDB21, where each song is composed of four stems: *vocals*, *bass*, *drums* and *other*. The original song can be recovered as the summation of the four stems[7]. We make both datasets available for download[8].

When simulating the errors, our objective was to cause a degradation in the training loss of Open-Unmix (Stöter et al., 2019) (upon convergence) of approximately 1dB SDR, when trained individually on label noise and bleeding, compared to training the same model on error-free data. Moreover, we chose not to split the datasets into a training and validation part.

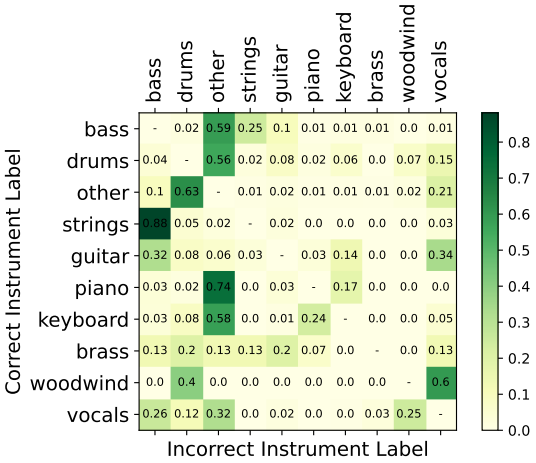


Figure 2: Statistics collected during our internal data cleaning activity. The values in the rows are normalized so that they sum to 1. For example, in all the errors we found in our internal data, the chances that a guitar was labeled as bass are 32 %.

Label Noise The simulation of label noise is based on randomly changing the label of a stem. This is applied on a subset of the stems in the clean dataset: this allowed us to effectively simulate label noise as occurs in real life. We chose to apply label noise to 20 % of the stems in the source data, independently of their label. If a stem was subject to label noise, the new label was sampled from a distribution that reflected the label noise we found in our internal datasets. To do this, we collected statistics over the frequency of corrections we performed during our data cleaning activity: if a stem initially labeled as c_{src} was corrected to c_{dst}, we increased the likelihood that during the simulation of label noise we would change c_{dst} into c_{src}. These statistics are reported in Figure 2.

Please note that our statistics refer to a taxonomy of ten musical instruments, while the final dataset contains stems for four: this means that some corruptions we perform will not have an effect in final version of the dataset. For example, changing *drums* into *bass* represents an error in the final dataset, changing *guitar* to *piano* does not (as they both belong to the class *other*). In the end, 34 % of the stems in *SDXDB23_LabelNoise* were affected by label noise.

Bleeding Simulating bleeding was a more elaborate process. The amount of bleeding in a recording is usually low (in terms of signal-to-noise ratio). In order to achieve the target impact of 1dB SDR on the model convergence, we decided to apply corruptions to *every single stem* in *SDXDB23_Bleeding*. More specifically, *every stem bleeds into every other stem* in the same song.

The bleeding component in a stem was obtained as a copy of another stem, where we applied gain reduction and filtering. The scaled and filtered signal was then summed to the stem that would contain the corruption. We randomly scaled each bleeding signal in the range $[-7,-12]$dB and applied either a low-pass filter or a band-pass filter (we randomly selected for every stem). The order of the filter was randomly chosen in the range $[3,10)$. When we applied a low-pass filter, the cut-off frequency was randomly chosen in the range $[900,9000)$Hz. When we applied a band-pass filter, we randomly chose the low and high cut-off frequencies in the range $[200,600]$Hz and $[8,10]$kHz, respectively. All random choices were drawn from uniform distributions. The ranges were designed empirically to compromise between bleeding realism and desired impact on the model convergence.

3.4 Robust Baseline Model

We present here a simple baseline model for the task of robust MSS, which can be used for both label noise and bleeding. This method is invariant to the choice of network architecture: in our experiments we used Open-Unmix (Stöter et al., 2019).

We first trained the model on the full noisy dataset D_1, without any data cleaning, and obtained a system that achieved suboptimal performance: we name this $UMX^{(1)}$. We then ran inference of $UMX^{(1)}$ *on the same data it was trained upon*. In other words, we used $UMX^{(1)}$ to remove some of the errors present in D_1: although the model had only suboptimal performance, we could expect it to be able to remove part of the errors in the data. This step created a new improved version of D_1, which we name D_2. We then trained a new model $UMX^{(2)}$, from scratch, on D_2: we expected this model to achieve better performance than $UMX^{(1)}$. We could then repeat this iterative refinement of the training data N times and train the final model $UMX^{(N)}$. The maximum number of iterations N was found empirically: to realize our baseline, we used $N = 2$. Please note how this method can be interpreted as a distillation approach, where the current model $UMX^{(i)}$ acts as a teacher during the training of the next model $UMX^{(i+1)}$ (the student) (Hinton et al., 2015).

The creation of the improved dataset at every iteration can be performed in two ways: we name them *redistributed* and *filtered*. Every song in the dataset D_i at iteration i is composed of stems $\mathbf{s}_{\bar{c}}^{(i)}(n), \forall \bar{c} \in C$, where $\bar{c}$ is the stem label in the dataset. These stems are used as input to the current model $UMX^{(i)}$ to create D_{i+1}. $UMX^{(i)}$ outputs estimates $\mathbf{y}_{\bar{c},c}^{(i)}(n), \forall c \in C$ for an input stem $\mathbf{s}_{\bar{c}}^{i}(n)$. Please note that $\bar{c}$ indicates the label of the input stem in D_i, while c indicates the separated instrument in the current estimate of $UMX^{(i)}$. In other words, $\mathbf{y}_{\bar{c},c}^{(i)}(n)$ is the estimate of instrument c of the model $UMX^{(i)}$ given as input a stem of instrument $\bar{c}$.

In the *redistributed* method we define a new stem as the following:

$$\mathbf{s}_c^{(i+1)}(n) = \sum_{\bar{c} \in C} \mathbf{y}_{\bar{c},c}^{(i)}(n), \tag{4}$$

that is, we sum together all the estimates for one instrument c given as input all the stems in the current song. The contributions of the current instrument c present in the wrong input stems (i.e., when $\bar{c} \neq c$) are preserved in the output.

In the *filtered* method we define a new stem as the following:

$$\mathbf{s}_c^{(i+1)}(n) = \mathbf{y}_{c,c}^{(i)}(n), \tag{5}$$

that is, we only consider as input the stem of the current instrument c. Any contribution of the current instrument c present in a stem other than its own is discarded.

4. Standard Music Separation

Part of the success of the last edition of the challenge was due to the fact that not only researchers in academia but also industry players participated and submitted their systems. This was made possible by the presence of a leaderboard where no constraint was

given: any model of any complexity, trained on any data, could participate[9].

For this edition, we offered the same in Leaderboard C. Our intention was to measure the improvement in the performance of the submitted systems with respect to the same benchmark in 2021.

4.1 Baseline Models

We provided several baselines to the participants. The role of a baseline model was not only to appear in the leaderboard as a reference for existing methods; each participants could choose also to use it as starting point for their submissions.

For robust MSS, we trained models separately on *SDXDB23_LabelNoise* and *SDXDB23_Bleeding*: we provided two baselines based on Open-Unmix (Stöter et al., 2019), two based on Hybrid Demucs (Défossez, 2021) and two based on MDX-Net (Kim et al., 2021).

For standard MSS, we provided Open-Unmix (Stöter et al., 2019) (the large model, *UMX-L*), Band-Split RNN (Luo and Yu, 2023) and *X-UMX-M* (Sawata et al., 2021). The former was already provided for the previous edition of the challenge, while the other two were new. As described in the original paper (Luo and Yu, 2023), to create the BSRNN baseline we trained 4 separate BSRNN models for the four instruments in the MUSDB18 dataset (Rafii et al., 2017). We used the same band-split schemes for the four tracks but used slightly smaller model sizes. More specifically, we used 10 BSRNN blocks instead of 12, and set the feature dimension to 80 instead of 128. We used the publicly available training pipeline on MUSDB18 dataset for model training[10], where we used 8 GPUs and set the per-GPU batch size to 2. The full pipeline as well as the model weights are available online[11]. The *X-UMX* model corresponds to the one trained on 20k songs, as described in (Sawata et al., 2023).

4.2 Listening Test

For non-robust MSS only, in addition to the objective evaluation based on SDR, we organized a subjective evaluation based on a listening test where professionals in the music industry rated the separations of the best systems. The main goal of the listening test was to assess the performance of various source separation models via comparative AB sampling. We included in the listening test one model for each of the three teams that achieved the highest SDR scores on leaderboard C (*SAMI-ByteDance*, *ZFTurbo* and *kimberley_jensen*). During the listening test, each assessor was in charge of judging segments of either the output of a source separation model or its residual (i.e., we subtract the output of the model from the input mixture). The latter reveals how good the model can suppress a source, highlighting potential residues of the signal of interest that are not correctly suppressed. The listening test was carried out by a panel of seven assessors who bring experience from various domains of music. This panel comprised Grammy and Latin award-winning singers, songwriters, composers, music producers, sound engineers, and an educator. Each assessor's profile and their contributions are reported in Table 2. The assessors were adequately trained to understand and identify common issues associated with source separation, such as distortion and artifacts. Each assessor was expected to complete a minimum of 72 comparisons, equivalent to three comparisons for each pair of models, across the four instrument classes plus their residuals.

To generate the evaluation data for the test, we selected ten songs from those in MoisesDB not used to create *SDXDB23_LabelNoise* and *SDXDB23_Bleeding*. The songs we used are listed in Table 1. We applied the three candidate models to each song and obtained the separated signals. We did not have the assessors evaluate the complete separated signals. Rather, we identified in each song four segments of three musical bars each, where the audio energy was sufficiently high, and we used only the separated signals of those segments. We then created the residual signals of the separations. Finally, we paired separations of the same segment obtained with different models and randomly assigned them to the assessors.

5. Challenge Results and Winning Approaches

In this section we provide the results of the MDX track of the challenge. Each section describing a winning system is written by the corresponding team. Tables 11, 12 and 13 show the final scores on the challenge leaderboards.

5.1 Iterative Refinement Baseline

We report in Table 3 the performance of our iterative refinement baseline. First of all, we highlight the impact that the errors in the data have on the performance of the model: training on *SDXDB23_LabelNoise* degrades the average separation quality by 1.4dB, while training on *SDXDB23_Bleeding* degrades it by 0.8dB.

We then use these models to improve the training data, first by using our *redistributed* approach. If we train a new model on this improved dataset, average performance on *SDXDB23_LabelNoise* increases by 0.4dB in SDR, while we experience virtually no change for *SDXDB23_Bleeding*.

If we use our *filtered* approach to improve the training data, we see an average improvement of 0.9dB for *SDXDB23_LabelNoise* and of 0.5dB for *SDXDB23_Bleeding*.

Please note that we also report scores for when we apply our iterative approach to the clean data only (*MoisesDB*): we experience a loss of 0.2dB only when using the *redistributed* strategy, likely due to distortions

N.	Artist	Title	Genre
0	Andy Bennett	Grace	Singer/Songwriter
1	Virtual Tongues	Pontiac	Rock
2	Horizon	Too Late	Electronic
3	Ben James	Yesterday	Pop
4	Iain Kerr & Friends	Dreaming ”Bout Being With You	Jazz
5	ProRata	Broken	Rock
6	Mansa Musa	Sightseers	Rock
7	FNDEF	Time To Show	Rap
8	Horizon	Get It Together	Electronic
9	Greenbacks	Thinking It”s a Sign	Rock

Table 1: Songs used in the listening test.

Assessors' Name	**Experience**
Autumn Rowe	Singer-Songwriter, 1x Grammy Award Winner, 2x Grammy nominee
Alexandre Kassin	Music Producer, Latin Grammy Award Winner, 2x Latin Grammy nominee
Daniel Morris	Educator/Performer, Senior Professor at Berklee College of Music since 1988
Jordan Rudess	Composer/Performer, 1x Grammy Award Winner performing for Dream Theater
Kaléu Runze	Producer/Sound Engineer, 1x Latin Grammy Award Winner, 2x Latin Grammy Award nominee
Yuri Queiroga	Producer/Artist, 1x Latin Grammy Winner, 2x Latin Grammy nominee
Felipe Vassão	Music Producer/Songwriter, Latin Grammy Award Winner

Table 2: Board of assessors of the listening test.

and artifacts introduced by the model during the first iteration.

5.2 Team *ZFTurbo* (Roman Solovyev, Alexander Stempkovskiy, Tatiana Habruseva)

5.2.1 Approach

Our approach is based on the ensemble of the models suited best for a particular stem. We filter vocals first, before separating the other stems, and employ a weighted ensemble of different models and checkpoints. To find optimal weights for the ensemble we used our own developed benchmarks for sound demixing tasks and leaderboards (Solovyev et al., 2023)[12].

To separate the vocals, we used three pre-trained models: UVR-MDX1[13], UVR-MDX2[14] from the Ultimate Vocal Remover project[15], and HT Demucs (fine-tuned)[16]. The vocals were separated independently by all of these models, then the results were combined with weights. UVR-MDX2 is an instrumental prediction model, so to get the vocals we need to subtract results from the original track.

One augmentation technique we used is an inversion. We invert the mixture $\mathbf{x}(n)$ by multiplying the waveform vector by -1, run the inference on it, and then revert the results back. Combining inference obtained on the mixture and its inversion provides a test-time augmentation for the waveform.

In the following we denote Demucs as $D(\mathbf{x},\theta)_{\text{model}}$, where $\mathbf{x}$ is the input we feed to the model, *model* is one of *demucs_ft*, *demucs*, *demucs_6s* and *demucs_mmi*, and θ is the overlap parameter. All models were applied with the shift parameter equal to 1. The ensemble for vocal separation is:

$$\begin{aligned}
\mathbf{y}_{\text{V}}^{(1)} &= \text{UVR-MDX1}\big(\mathbf{x}, \text{overlap}{=}0.6\big),\\
\mathbf{y}_{\text{V}}^{(2)} &= \mathbf{x} - (-1)\cdot\text{UVR-MDX2}\big(-\mathbf{x}, \text{overlap}{=}0.6\big),\\
\mathbf{y}_{\text{V}}^{(3)} &= \frac{D\big(\mathbf{x}, 0.6\big)_{\text{demucs_ft}} - D\big(-\mathbf{x}, 0.6\big)_{\text{demucs_ft}}}{2},
\end{aligned}$$

where $\mathbf{x}(n)$ is the original mixture.

The three estimates were used to produce the final vocals as:

$$\mathbf{y}_{\text{V}} = \alpha_1\mathbf{y}_{\text{V}}^{(1)} + \alpha_2\mathbf{y}_{\text{V}}^{(2)} + \alpha_3\mathbf{y}_{\text{V}}^{(3)},$$

where the optimal weights were found to be $\alpha_1 = 10, \alpha_2 = 4$ and $\alpha_3 = 2$.

Next, we applied four versions of Demucs models to the instrumental track only $\mathbf{y}_{\text{A}}(n) = \mathbf{x}(n) - \mathbf{y}_{\text{V}}(n)$ (A stands for *accompaniment*) and its inversion.

$$\begin{aligned}
\mathbf{y}_{\text{B,D,O}}^{(1)} &= \frac{D\big(\mathbf{y}_{\text{A}}, 0.5\big)_{\text{demucs_ft}} - D\big(-\mathbf{y}_{\text{A}}, 0.5\big)_{\text{demucs_ft}}}{2},\\
\mathbf{y}_{\text{B,D,O}}^{(2)} &= \frac{D\big(\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs}} - D\big(-\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs}}}{2},\\
\mathbf{y}_{\text{B,D,O}}^{(3)} &= \frac{D\big(\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs_6s}} - D\big(-\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs_6s}}}{2},\\
\mathbf{y}_{\text{B,D,O}}^{(4)} &= \frac{D\big(\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs_mmi}} - D\big(-\mathbf{y}_{\text{A}}, 0.6\big)_{\text{demucs_mmi}}}{2}.
\end{aligned}$$

Based on our Multisong dataset (Solovyev et al., 2023), we combined the models' results with the fol-

	Global SDR (dB)				
	Mean	Bass	Drums	Other	Vocals
MoisesDB (203 songs)					
Original dataset	4.43	4.65	5.06	**3.02**	**5.00**
Improved dataset (redistributed)	4.27	4.68	4.93	2.72	4.75
Improved dataset (filtered)	**4.46**	**5.07**	**5.16**	2.77	4.86
SDXDB23_LabelNoise					
Original dataset	3.01	3.76	2.83	1.62	3.82
Improved dataset (redistributed)	3.44	4.00	3.81	1.86	4.08
Improved dataset (filtered)	**3.90**	**4.57**	**4.57**	**2.22**	**4.25**
SDXDB23_Bleeding					
Original dataset	3.60	3.90	3.84	2.50	4.17
Improved dataset (redistributed)	3.59	3.73	4.07	2.40	4.17
Improved dataset (filtered)	**4.09**	**4.65**	**4.76**	**2.52**	**4.44**

Table 3: Results of our iterative refinement baseline. We use a source separation algorithm trained on corrupted data to improve the dataset: training the same model on the improved data increases the separation quality.

lowing weights:

$$\bar{\mathbf{y}}_B = 19\cdot\mathbf{y}_B^{(1)} + 4\cdot\mathbf{y}_B^{(2)} + 5\cdot\mathbf{y}_B^{(4)} + 8\cdot\mathbf{y}_B^{(4)},$$
$$\bar{\mathbf{y}}_D = 18\cdot\mathbf{y}_D^{(1)} + 2\cdot\mathbf{y}_D^{(2)} + 4\cdot\mathbf{y}_D^{(4)} + 9\cdot\mathbf{y}_D^{(4)},$$
$$\bar{\mathbf{y}}_O = 14\cdot\mathbf{y}_O^{(1)} + 2\cdot\mathbf{y}_O^{(2)} + 5\cdot\mathbf{y}_O^{(4)} + 10\cdot\mathbf{y}_O^{(4)}.$$

Note, the model *demucs_mmi* has a slightly different architecture (Demucs3); it can be included with a larger weight for diversification.

Finally, we obtain the values of the final stem tracks as follows:

$$\mathbf{y}_B = \frac{\mathbf{y}_A - \bar{\mathbf{y}}_O - \bar{\mathbf{y}}_D + 2\cdot\bar{\mathbf{y}}_B}{3},$$
$$\mathbf{y}_D = \frac{\mathbf{y}_A - \bar{\mathbf{y}}_O - \bar{\mathbf{y}}_B + 2\cdot\bar{\mathbf{y}}_D}{3},$$
$$\mathbf{y}_O = 2\cdot\frac{\mathbf{y}_A - \bar{\mathbf{y}}_B - \bar{\mathbf{y}}_D + \bar{\mathbf{y}}_O}{3}.$$

5.2.2 Results

The results for this ensemble are in Table 4.

	Global SDR (dB)				
Dataset	Mean	Bass	Drums	Other	Vocals
MultiSong MVSep	10.11	12.68	11.68	6.67	9.62
MDX23 public test	9.41	9.87	9.52	7.43	10.81
MDX23 private test	9.25	9.94	9.53	7.05	10.51

Table 4: (Team *ZFTurbo*) SDR scores for the final ensemble on the MultiSong MVSep datasets (Solovyev et al., 2023) and MDX23 test sets (leaderboard C).

The solution ranked 2nd place in leaderboard C of the challenge. The source code of our solution is publicly available on GitHub[17].

5.3 Team *subatomicseer* (Nabarun Goswami, Tatsuya Harada)

5.3.1 Approach

We leverage the multi-resolution analysis (MRA) capabilities of discrete wavelet transform (DWT) and propose two new models for source separation. We also propose a noise-robust training scheme to handle corrupted data.

The first architecture, named Wavelet HTDemucs (WHTDemucs), builds upon the HTDemucs model (Rouard et al., 2023) and extends it by introducing a third DWT branch. This branch includes independent encoders, decoders, and cross-transformers, with the frequency branch acting as a residual bridge between the DWT branch and the temporal branch.

The second model, called DWT Transformer UNet (DTUNet), utilizes a simpler architecture. It comprises independent encoders and decoders for MRA signals, with a single cross-transformer block to combine with the temporal branch. A source-independent post filter is applied to the added branch outputs.

Additionally, we propose the following noise-robust training procedure:

- we employ the L1 loss to exploit the dominant instrument and mitigate the permutation problem at the beginning;
- the Mixture Consistency loss (Wisdom et al., 2019) is used consistently throughout the training process;
- the unsupervised MixIT Loss (Wisdom et al., 2020) is applied. We create mixtures of mixtures by randomly partitioning and summing the four sources into two groups;
- later in training, the Mean Teacher loss (Tar-

vainen and Valpola, 2017) is incorporated, utilizing the Exponential Moving Average (EMA) of the model weights.

In the first version (V1), mean teacher targets were computed by processing each noisy input stem with the EMA model and summing the stems. In version 2 (V2), we simplified the process by using a single pass of the EMA model on the input mixture to reduce computational costs.

For Leaderboard A, we evaluated the label noise dataset using the DTUNet trained with V2 loss. We set a minimum threshold of 9dB SDR and manually checked a subset of the filtered stems, which were then used to train a set of lightweight Band-split RNN models (BSRNN) (Luo and Yu, 2023). Finally, we blended the outputs from WHTDemucs (V1), DTUNet (V2), and BSRNN (clean). For Leaderboard B, we blended the outputs from WHTDemucs (V1) and DTUNet (V2). For Leaderboard C, we trained DTUNet and BSRNN on 347 songs, including the MUSDB test set (Rafii et al., 2017). The outputs of these two models were blended with HTDemucs (trained on 800 songs and released by the original authors).

5.3.2 Results and Discussion

We report our results in Tables 5 to 8. In the tables, underlined scores represent the overall best score, while **bold** scores refer to the best among our proposed models, models trained by us on the same dataset and official baselines.

	Global SDR (dB)				
Model	Mean	Bass	Drums	Other	Vocals
DTUNet (347)	**8.788**	**8.749**	10.652	**6.764**	8.988
BSRNN (347)	8.652	8.063	**10.796**	6.380	**9.369**
HTDemucs (800)	9.190	9.683	10.759	7.165	9.151

Table 5: (Team *subatomicseer*) SDR scores on local validation set for Leaderboard C.

	Global SDR (dB)				
Model	Mean	Bass	Drums	Other	Vocals
ByteDance(LB1)	9.97	11.15	10.27	7.08	11.36
Ours (blend, LB8)	**8.537**	**9.328**	**9.328**	**6.182**	**9.311**
UMXL-Baseline	6.520	6.619	6.838	4.891	7.732
BSRNN-Baseline	6.142	5.628	6.534	4.425	7.983

Table 6: (Team *subatomicseer*) Comparison of Leaderboard C scores.

From Table 5, we see that the proposed DTUNet outperforms BSRNN when trained on the same data, however, for vocals, the BSRNN model has a significant advantage. Table 6 shows the performance of our final submission in comparison to the top leaderboard score and the baselines.

	Global SDR (dB)				
Model	Mean	Bass	Drums	Other	Vocals
WHTDemucs(V1)	5.933	6.411	5.731	4.416	7.173
DTUNet(V2)	5.930	5.837	6.706	4.095	7.083
BSRNN(clean)	-	-	-	-	-
Blend	**6.601**	**6.696**	**7.026**	**4.611**	**8.072**

Table 7: (Team *subatomicseer*) Comparison of Leaderboard A scores. BSRNN(clean) was not evaluated individually.

	Global SDR (dB)				
Model	Mean	Bass	Drums	Other	Vocals
WHTDemucs(V1)	5.860	5.901	5.613	**4.676**	7.252
DTUNet(V2)	5.618	5.373	6.177	3.917	7.004
Blend	**6.314**	**6.331**	**6.864**	4.591	**7.469**

Table 8: (Team *subatomicseer*) Comparison of Leaderboard B scores.

Tables 7 and 8 show the performance of the models for Leaderboards A and B. Since we used two different model architectures for V1 and V2 losses, they cannot be directly compared. However, blending definitely improves separation performance. We leave further analysis and ablation studies regarding the proposed models and noise-robust training scheme to future works.

5.4 Team *kuielab* (Minseok Kim, Jun Hyung Lee)

5.4.1 Approach

Our approach can be summarized into two parts: an improved version of TFC-TDF-UNet (Choi et al., 2020; Kim et al., 2021) (which we call v3) and a loss masking (truncation) approach for noise-robustness, where training batch elements with high loss are discarded before model weight update. TFC-TDF-UNet v3 was used for all final submissions: for Leaderboard A/B, we used v3 models trained with loss masking and for Leaderboard C, we used an ensemble of TFC-TDF-UNet v3 and Demucs models (Défossez, 2021; Rouard et al., 2023). More details can be found in (Kim and Lee, 2023).

5.4.2 Results and Discussion

Other than the models used for challenge submissions, we trained an additional v3 model for a quantitative comparison with v2, which is shown in Table 9.

We also provide in Table 10 an ablation study on loss masking. *modelA* and *modelB* refer to v3 models trained on *SDXDB23_LabelNoise* and *SDXDB23_Bleeding*, respectively. We report the challenge evaluation results.

TFC-TDF-UNet v3 improves v2 on SDR performance as well as being faster at inference time. Also, training with loss masking was effective for all instrument classes, especially for label noise. As future work, we would like to focus on a more realistic setting where

	Global SDR (dB)				
Model	Bass	Drums	Other	Vocals	Speed
v2	6.85	6.87	5.44	8.96	12.8x
v3	**7.36**	**8.81**	**6.19**	**9.22**	**15.0x**

Table 9: (Team *kuielab*) Comparison of TFC-TDF-UNets v2 and v3 on the MUSDB18-HQ benchmark. "Speed" denotes the relative GPU inference speed with respect to real-time on the challenge evaluation server.

	Global SDR (dB)			
Model	Bass	Drums	Other	Vocals
modelA	**6.43**	**6.38**	**4.64**	**7.58**
modelA w/o loss masking	5.31	5.31	3.45	6.12
modelB	**6.58**	**6.20**	**4.69**	**7.41**
modelB w/o loss masking	6.11	5.86	4.36	6.87

Table 10: (Team *kuielab*) Ablation study on loss masking. Note that modelA/modelB are "single" models and not the final submission ensembles.

clean sources are also available as well as noisy ones. We can investigate how much a model trained with clean data can aid in robust training, or ways to improve existing models by fine-tuning them with abundant noisy data.

5.5 Team *CCOM* (Yuanliang Dong, Xinran Zhang, Jiafeng Liu)

The training of our final model on leaderboard A has two consecutive parts. Part 1 was trained from scratch using *loss truncation*, and Part 2 was built upon the model from Part 1.

5.5.1 Part 1: Robust Training with Loss Truncation

The idea of *loss truncation* was introduced in (Kang and Hashimoto, 2020). Suppose an oracle model that perfectly separates each stem with SDR=∞, then true label samples will have loss = 0, and false label samples will have loss > 0. Then the oracle model is a perfect classifier for noisy label vs clean label stems using (quantile of) loss as the classification criterion. To filter out noisy labels, sort the loss value in descending order in a batch of samples, calculate some quantile of the losses as a threshold, then drop samples above the threshold, and the samples with noisy labels can be completely dropped. In practice, we are not allowed to use any oracle or external model. So we heuristically apply the loss truncation function directly from the very beginning of training.

We use the training architecture of Demucs, in which we simply re-write the loss function into the truncated version. Our results show that although we do not have such oracle models to perfectly distinguish noise labels, by directly training from scratch on noisy data with loss truncation it still achieves competitive results (average SDR = 6.277dB after part 1).

5.5.2 Part 2: Label Noise Filter

Since the dataset used in Leaderboard A has noisy stems, we designed a label noise filter to automatically determine whether a stem is clean or not. After Part 1, we could assume that the music source separation model has the ability to perform well on this discriminative task.

An important challenge is overfitting: the model may *memorize* the wrong stems when a noisy sample was used for training in Part 1, although loss truncation could alleviate this phenomenon to some extent. We used some tricks to avoid overfitting during inference. First, we change the input from mixture to a single stem. We use the pretrained model to separate each stem in the noisy dataset and then observe the amplitude of the four output audio. If the stem is clean, the amplitude of the correct audio would be far larger than the other audio. Second, we did out-of-range data augmentation compared with training phase. During training, ± 2 semitones pitch and 12 % tempo change was made on the dataset, while the audio was sped up to 2x and transposed up by 6 semitones during inference. We use this filter to recreate a new *clean* dataset, on which we trained a Demucs model from scratch (the discrimination-recreation-training was conducted twice recursively).

6. Listening Tests Results

In this Section, we report the results of the listening test. The assessors conducted a total of 583 comparisons. Each comparison involved the separated outputs produced by two of the three models under evaluation and the original mixture to serve as reference.

First, we quantified the interactions of each assessor with the audio samples. From the data gathered on the Moises.ai testing platform, we found that the assessors played each comparison for an average duration of 27.95 ± 17.88 seconds. The assessors performed an average of 3.13 ± 1.93 switches between the segments of each comparison.

We report the global results of the listening test in Figure 3. The table in Figure 3a shows the number of times a model in each row won an evaluation against a model in each column. We also report a second table (Figure 3b) showing the same results normalized by the number of evaluations for each pair of models.

In order to detect potential biases as a result of our choice of assessors, we group them into two categories: *Producer* and *Musician-Educator* (we consider musicians and educators in the same category, since our assessors panel only includes one educator, who is also a performer). We report the results of the listening test independently for the two categories in Figure 4. Figure 4a show the results for *Producer*, while Figure 4b show the results for *Musician-Educator*. We

Rank	Participant	Global SDR (dB)					Submissions to Ldb A	
		Mean	Bass	Drums	Other	Vocals	1st phase	2nd phase
Submissions								
1.	CCOM	7.455	8.117	7.993	5.342	8.369	7	50
2.	subatomicseer	6.601	6.696	7.026	4.611	8.072	65	33
3.	kuielab	6.513	6.707	6.712	4.816	7.816	99	25
4.	aim-less	6.444	6.746	7.190	4.557	7.284	10	22
5.	yang_tong	6.334	6.292	7.455	3.937	7.651	-	2
Baselines								
	UMX	3.013	3.766	2.837	1.624	3.825		
	Demucs	4.836	5.553	5.679	2.886	5.225		
	MDX-Net	3.486	4.256	2.841	2.423	4.424		

Table 11: Final leaderboard A (models trained only on *SDXDB23_LabelNoise*; top-5)

Rank	Participant	Global SDR (dB)					Submissions to Ldb B	
		Mean	Bass	Drums	Other	Vocals	1st phase	2nd phase
Submissions								
1.	kuielab	6.581	6.975	6.646	4.962	7.741	99	13
2.	ZFTurbo	6.384	6.937	6.861	4.622	7.115	32	4
3.	subatomicseer	6.314	6.331	6.864	4.591	7.469	65	11
4.	CCOM	6.203	6.337	6.323	4.284	7.867	7	17
5.	alina_porechina	5.873	6.009	6.098	4.087	7.299	99	118
Baselines								
	UMX	3.607	3.901	3.847	2.504	4.174		
	Demucs	5.334	5.899	5.557	3.691	6.188		
	MDX-Net	3.5611	4.001	2.300	2.648	5.293		

Table 12: Final leaderboard B (models trained only on *SDXDB23_Bleeding*; top-5)

Rank	Participant	Global SDR (dB)					Submissions to Ldb C	
		Mean	Bass	Drums	Other	Vocals	1st phase	2nd phase
Submissions								
1.	SAMI-ByteDance	9.965	11.153	10.269	7.075	11.363	13	5
2.	ZFTurbo	9.259	9.941	9.533	7.050	10.511	32	24
3.	kimberley_jensen	9.181	10.056	9.465	6.804	10.398	86	134
4.	kuielab	8.971	9.716	9.433	6.721	10.012	99	54
5.	alina_porechina	8.625	9.921	9.287	6.225	9.069	99	172
Baselines								
	UMX-L	6.520	6.619	6.838	4.891	7.732		
	BSRNN	6.142	5.628	6.534	4.425	7.983		
	X-UMX-M	6.296	5.851	6.869	4.418	8.044		

Table 13: Final leaderboard C (models trained on any data; top-5)

observe that assessors in the category *Producer* showed a higher preference for *kimberley_jensen*, while those in *Musician-Educator* preferred *SAMI-ByteDance*.

The audio segments used in the test were obtained by either extracting one of the four instruments, or by removing it. Figures 5, 6, 7 and 8 show the results on extracting and removing bass, drums, other and vocals respectively.

6.1 TrueSkill Ratings

In order to generate a valid global ranking of the three models, we employed the TrueSkill ranking system (Herbrich et al., 2007) to summarize the results of our test. TrueSkill generates a ranking based on a series of matches (i.e., the comparisons in our listening test) between pairs of players (i.e., the models under evaluation). The final ranking is shown in Table 14. We see that the differences between the three models are very small. This might be due to the models having similar performance, but also to the relatively contained size of the listening test.

Finally, TrueSkill enables us to compute the probability of a hypothetical draw between any two models. For the match *SAMI-ByteDance* vs *ZFTurbo* the draw probability is 0.981, for the match *ZFTurbo* vs *kimberley_jensen* it is 0.98 and for the match *SAMI-ByteDance* vs *kimberley_jensen* it is 0.975.

Position	Model	μ	σ
1	*kimberley_jensen*	24.793	0.779
2	*ZFTurbo*	24.362	0.779
3	*SAMI-ByteDance*	24.011	0.779

Table 14: Final ranking obtained with TrueSkill. We used the default parameters for each player ($\mu = 25$ and $\sigma = 8.33$).

6.2 Human Preference vs Objective Metric

Figure 9 shows how often the human judgments agree with the objective scores based on SDR. Each point refers to a single evaluation of the listening test. We encode in the horizontal axis the SDR difference between the two competing models in that evaluation. In the vertical axis we encode whether the model with higher SDR was also selected by the assessor. As we have multiple human judgements for the same combination of model pair, song and instrument extraction or separation, we average those results.

The figure does not show a clear correlation between objective scores and the judgements made by the listeners. The quality of all the three models participating in the listening test is high and this makes it difficult to confidently choose the best one. Nevertheless, we find that the model by *kimberley_jensen* achieved first place in the final ranking, despite being third in the leaderboard obtained using SDR.

7. Organizing the Challenge and Future Editions

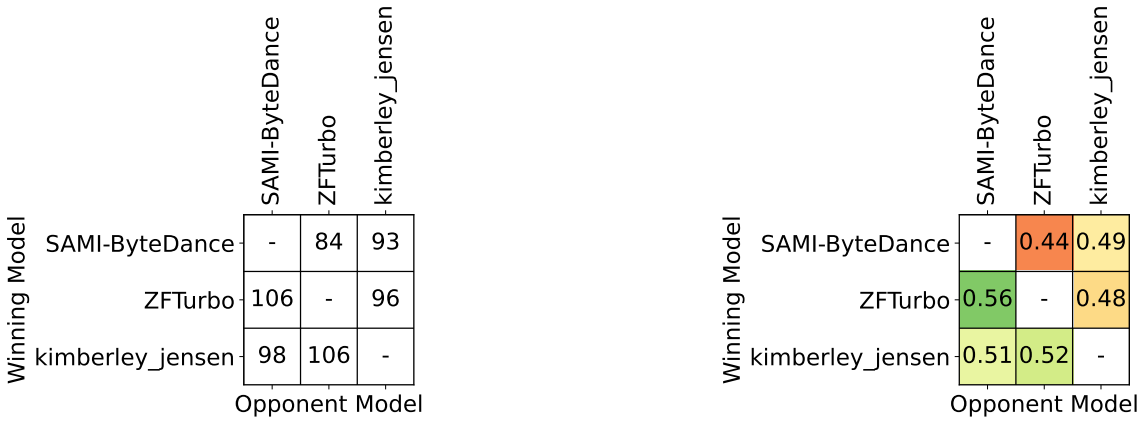
At the end of MDX'21, we stated (Mitsufuji et al., 2022) that source separation can still bring benefits to many application and research areas, and this motivated the need for future editions of the competition. For this reason, we organized the Sound Demixing Challenge (SDX'23). Our aim was to provide once more the familiar benchmark that participants knew already, but also expand its formulation in various directions. At the time of MDX'21, we designed the competition so that researchers both new and experienced on this topic would be able to evaluate their models against a new test set. We tried to keep the competition fair, by hiding the test data from the participants, and we maximized the visibility of the event by hosting it on AIcrowd[18].

In 2023, the principles we used in organizing the challenge have not changed. Rather, we built upon them and expanded the scope of the whole competition, to include more application areas. Including cinematic sound separation (specifically, dedicating a whole challenge track to it (Uhlich et al., 2023)) and introducing robust music source separation are two directions that we wanted to highlight as promising items for research. Among many reasons, this expansion has been successful also thanks to an enlarged pool of organizers and entities involved: Sony, Moises and Mitsubishi Electric Research Labs joined forces and shared efforts and resources in order to realize a bigger competition.

7.1 Robust Music Source Separation

Our objective of having participants develop solutions for robust training of separation models has been met only partially. The provision of two new training sets (one for label noise and one for bleeding) has been positively received. Given how low the availability of training data is in the research world, the participants saw this as an opportunity for more experimentation and this contributed to the success of leaderboards A and B.

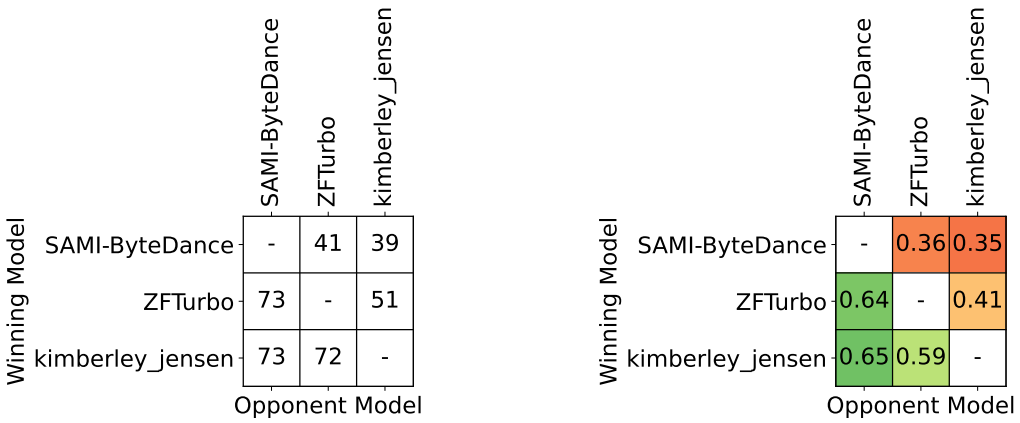
On the other hand, the necessarily small size of the datasets we provided (203 songs) allowed the participants to resort to manual cleaning strategies (at least in the case of label noise): some participants spent valuable time identifying which stems were corrupted and excluded them from the dataset. From the perspective of the challenge, this is acceptable; at the same time, though, it undermined our initial motivation for exploring such topic, as such an activity would not scale with the amount of data and, as a consequence, would not be realistic. Some participants reported that bleeding was clearly more difficult to tackle than label noise: we believe this to be related to the fact that *SDXDB23_Bleeding* contained exclusively problematic stems, while *SDXDB23_LabelNoise* only had approximately 34 % of problematic stems. For bleeding, no



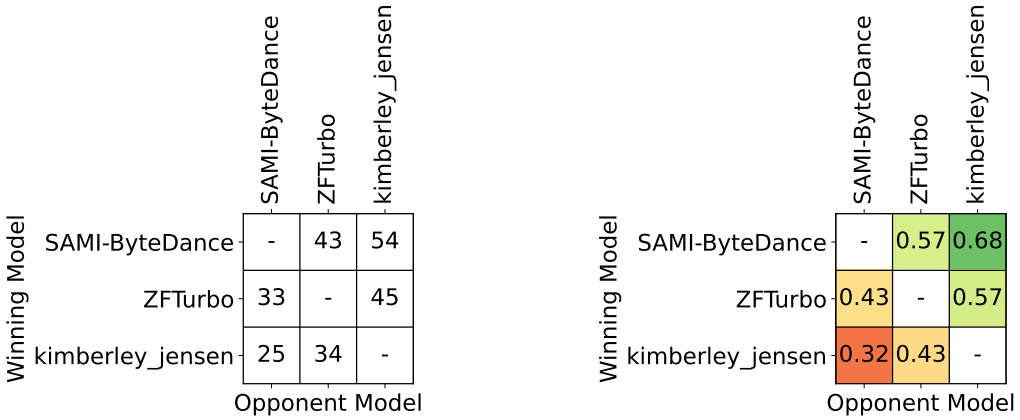
(a) Number of evaluations won by each model.

(b) Normalized number of evaluations won by each model.

Figure 3: Results of the listening test.

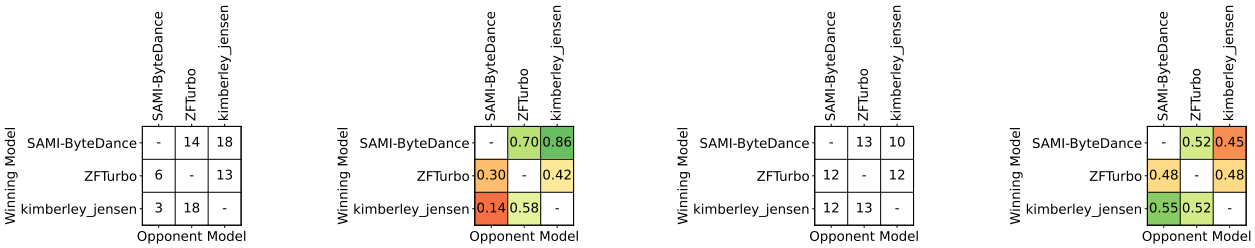


(a) Results for assessors in category *Producer*.



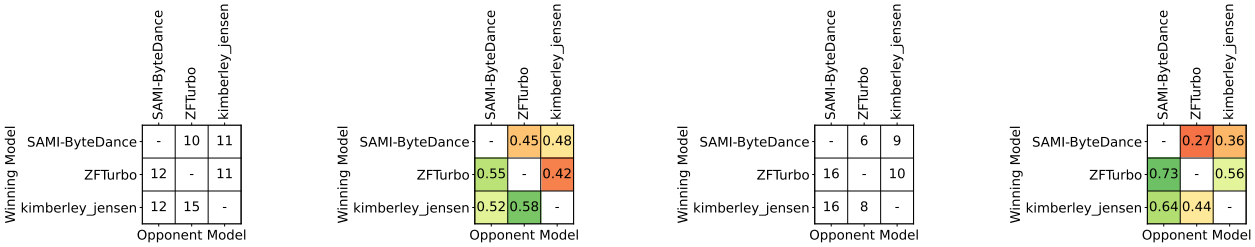
(b) Results for assessors in category *Musician-Educator*.

Figure 4: Results of the listening test by assessor category.



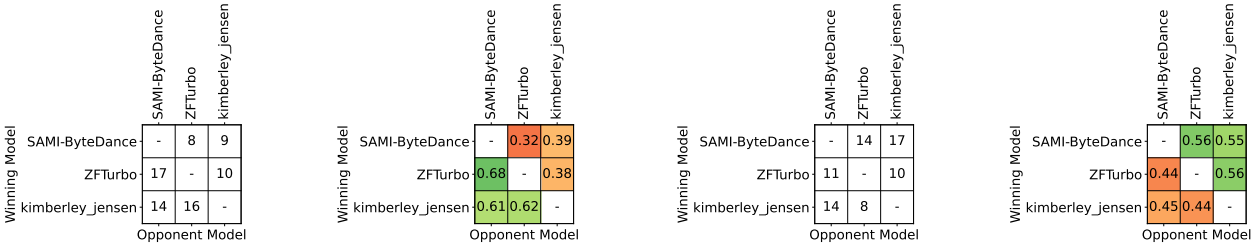
(a) The evaluations performed on bass extraction. **(b)** The evaluations performed on bass removal.

Figure 5: Results of the listening test on bass removal and extraction.



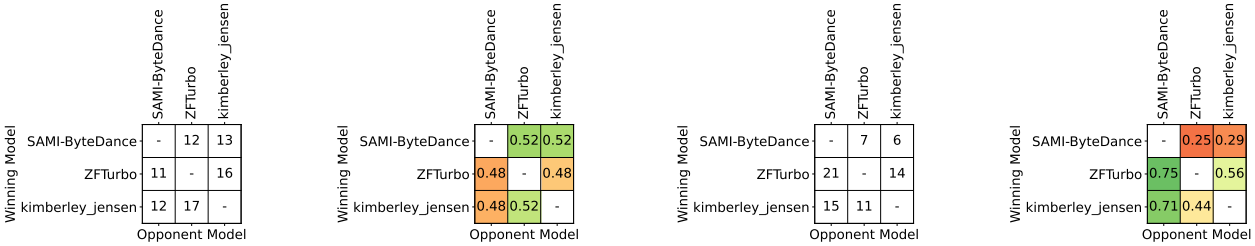
(a) The evaluations performed on drums extraction. **(b)** The evaluations performed on drums removal.

Figure 6: Results of the listening test on drums removal and extraction.



(a) The evaluations performed on other extraction. **(b)** The evaluations performed on other removal.

Figure 7: Results of the listening test on other removal and extraction.



(a) The evaluations performed on vocals extraction. **(b)** The evaluations performed on vocals removal.

Figure 8: Results of the listening test on vocals removal and extraction.

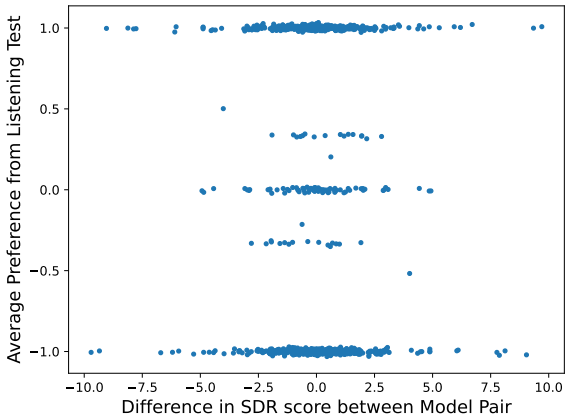


Figure 9: Correlation between the SDR scores and the results of the listening test.

manual selection of the data was possible (i.e., the settings were more realistic): as a consequence, it was perceived as more difficult to solve.

During the course of the challenge, many participants asked us whether we would allow the usage of pretrained models available on the Internet. We explicitly forbade the usage of such pretrained models, as that would have enabled the participants to clean the data with e.g., existing high quality source separation models. This would have again undermined our initial motivation for exploring robust training of source separation models. On top of that, although using a pretrained model is a practically viable solution for corrupted data, it would have threatened the aspects related to fairness in the competition, enabling different teams to use different tools. For this reason, we prevented participants from potentially achieving higher scores, but ensured that everyone would use the same resources to develop their models.

7.2 Standard Music Source Separation

Overall, the major trend in the solutions still remains blending multiple models (Uhlich et al., 2017). What the participants provided were in general not individual neural networks, but many, whose outputs were combined together to create the final estimates. We observed the same phenomenon also in MDX'21: a possibility is that, by providing trained baseline models to the participants, we implicitly encouraged this behavior. Although in SDX'23 we provided significantly less baselines than in the previous edition, the winning models from MDX'21 were made open source after the challenge, so they represented a viable alternative to our baselines.

We have noticed a larger degree of interest towards leaderboard C than towards the other two: the standard formulation for source separation was still the most competitive playground we offered. This allowed us to compare the evolution of the systems between the two editions of the challenge. In MDX'21, the highest score on the final leaderboard was achieved by the team *Audioshake*, with an average SDR score of 8.326dB. In SDX'23, the highest score on the final leaderboard was achieved by the team *SAMI-ByteDance*, with an average SDR score of 9.965dB. In other words, the highest public score achieved on our benchmark has increased by approximately 1.6dB over the course of two years.

Finally, running a full listening test on the top-performing submissions allowed us to get an alternative source of evaluation for the separation quality, besides the SDR score. This made possible the involvement of professionals in the music industry, who represent potential users of the technology: their feedback is therefore a very important signal to take into consideration when judging the quality of a system.

8. Conclusions

This paper summarized the music demixing (MDX) track of the Sound Demixing Challenge 2023. We provided a description of the challenge setup, we presented the topic of robust music source separation (MSS) and formalized the errors that can occur in a training dataset for source separation: label noise and bleeding. We explained how we realized two new datasets for robust MSS: *SDXDB23_LabelNoise* and *SDXDB23_Bleeding*. Then, we described the outcome of the challenge and reported the final results, together with a description of the winning approaches. We detailed how the evaluation has taken place, in particular with the introduction of a listening test specifically carried out with professional figures in the music industry. We believe that the SDX'23 challenge has given benefits to the source separation community and hope that we will continue organizing a long series of competitions in the future.

Notes

[1] The datasets are available for download at `https://developer.moises.ai/research#datasets`

[2] `https://www.youtube.com/watch?v=jcWINJxnw70`

[3] Ref. (Koo et al., 2023) was published during the preparation of this article which already makes use of our new proposed dataset *SDXDB23_LabelNoise*. Their approach is not based on training the model on the noisy data, but on automatically correcting the labels in the data before training the model.

[4] `https://www.aicrowd.com/challenges/sound-demixing-challenge-2023`

[5] `https://www.aicrowd.com/challenges/music-demixing-challenge-ismir-2021/leaderboards?challenge_leaderboard_extra_id=868&challenge_round_id=886`

[6] `https://www.aicrowd.com/challenges/music-demixing-challenge-ismir-2021/leaderboards?challenge_round_id=886`

[7] Please note that summing the four stems will not yield the exact mixture for *SDXDB23_Bleeding* as we simulate it by adding bleeding components to the original stems.

[8] https://developer.moises.ai/research#datasets

[9] We only imposed a limit on the model's inference time on a GPU.

[10] https://github.com/bytedance/music_source_separation

[11] https://gitlab.aicrowd.com/Tomasyu/sdx-2023-music-demixing-track-starter-kit

[12] https://mvsep.com/quality_checker/

[13] Checkpoint "Kim_Vocal_1.onnx" available at https://github.com/TRvlvr/model_repo/releases/download/all_public_uvr_models/Kim_Vocal_1.onnx

[14] Checkpoint "UVR–MDX–NET–Inst_HQ_2.onnx" available at https://github.com/TRvlvr/model_repo/releases/download/all_public_uvr_models/UVR-MDX-NET-Inst_HQ_2.onnx

[15] https://github.com/Anjok07/ultimatevocalremovergui

[16] Checkpoint "htdemucs_ft" available at https://github.com/facebookresearch/demucs

[17] https://github.com/ZFTurbo/MVSEP-MDX23-music-separation-model

[18] https://www.aicrowd.com/challenges/music-demixing-challenge-ismir-2021

Acknowledgment

The authors would like to thank Sony Music Entertainment Japan for the creation of MDXDB21.

The authors would like to thank the assessors who took part in the listening test.

The work of Nabarun Goswami and Tatsuya Harada was partially supported by JST Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015, and Basic Research Grant (Super AI) of the Institute for AI and Beyond of the University of Tokyo.

The work of Minseok Kim and Jun Hyung Lee was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1A2C2011452).

References

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. (2020). Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*.

Choi, W., Kim, M., Chung, J., Lee, D., and Jung, S. (2020). Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 192–198.

Défossez, A. (2021). Hybrid spectrogram and waveform source separation. In *Proc. the ISMIR 2021 Workshop on Music Source Separation*.

Garcia, H. F., Aguilar, A., Manilow, E., and Pardo, B. (2021). Leveraging hierarchical structures for few-shot musical instrument recognition. *arXiv preprint arXiv:2107.07029*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349.

Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill$^{\text{tm}}$: A bayesian skill rating system. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19 (NIPS-06)*, pages 569–576. MIT Press.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network.

Kang, D. and Hashimoto, T. B. (2020). Improved natural language generation via loss truncation. *ArXiv*, abs/2004.14589.

Kim, M., Choi, W., Chung, J., Lee, D., and Jung, S. (2021). KUIELab-MDX-Net: A two-stream neural network for music demixing. In *Proc. the ISMIR 2021 Workshop on Music Source Separation*.

Kim, M. and Lee, J. H. (2023). Sound demixing challenge 2023 music demixing track technical report: TFC-TDF-UNet v3.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. (2020). Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894.

Koo, J., Chae, Y., Jeon, C.-B., and Lee, K. (2023). Self-refining of pseudo labels for music source separation with noisy labeled data.

Lai, C.-H., Zou, D., and Lerman, G. (2019). Robust subspace recovery layer for unsupervised anomaly detection. *arXiv preprint arXiv:1904.00152*.

Lai, C.-H., Zou, D., and Lerman, G. (2023). Robust variational autoencoding with wasserstein penalty for novelty detection. In *International Conference on Artificial Intelligence and Statistics*, pages 3538–3567. PMLR.

Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Luo, Y. and Yu, J. (2023). Music source separation with

band-split rnn. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Mahanta, S. K., Khilji, A. F. U. R., and Pakray, P. (2021). Deep neural network for musical instrument recognition using mfccs. *Computación y Sistemas*, 25(2):351–360.

Manilow, E., Wichern, G., Seetharaman, P., and Le Roux, J. (2019). Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 45–49. IEEE.

Mitsufuji, Y., Fabbro, G., Uhlich, S., Stöter, F.-R., Défossez, A., Kim, M., Choi, W., Yu, C.-Y., and Cheuk, K.-W. (2022). Music demixing challenge 2021. *Frontiers in Signal Processing*, 1:18.

Mukherjee, D., Guha, A., Solomon, J. M., Sun, Y., and Yurochkin, M. (2021). Outlier-robust optimal transport. In *International Conference on Machine Learning*, pages 7850–7860. PMLR.

Pereira, I., Araújo, F., Korzeniowski, F., and Vogl, R. (2023). Moisesdb: A dataset for source separation beyond 4-stems.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The musdb18 corpus for music separation.

Rouard, S., Massa, F., and Défossez, A. (2023). Hybrid transformers for music source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Sawata, R., Takahashi, N., Uhlich, S., Takahashi, S., and Mitsufuji, Y. (2023). The whole is greater than the sum of its parts: Improving DNN-based music source separation. *arXiv preprint arXiv:2305.07855*.

Sawata, R., Uhlich, S., Takahashi, S., and Mitsufuji, Y. (2021). All for one and one for all: Improving music separation by bridging networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 51–55. IEEE.

Solovyev, R., Stempkovskiy, A., and Habruseva, T. (2023). Benchmarks and leaderboards for sound demixing tasks. *arXiv preprint arXiv:2305.07489*.

Stöter, F.-R., Liutkus, A., and Ito, N. (2018). The 2018 signal separation evaluation campaign. In *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Guildford, UK, July 2–5, 2018, Proceedings 14*, pages 293–305. Springer.

Stöter, F.-R., Uhlich, S., Liutkus, A., and Mitsufuji, Y. (2019). Open-unmix-a reference implementation for music source separation. *Journal of Open Source Software*, 4(41):1667.

Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Uhlich, S., Fabbro, G., Masato, H., Takahashi, S., Wichern, G., Le Roux, J., Chakraborty, D., Mohanty, S., Li, K., Luo, Y., Yu, J., Gu, R., Solovyev, R., Stempkovskiy, A., Habruseva, T., Sukhovei, M., and Mitsufuji, Y. (2023). The sound demixing challenge 2023 – cinematic demixing track. Under review.

Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. (2017). Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 261–265. IEEE.

Wang, H., Xiao, R., Dong, Y., Feng, L., and Zhao, J. (2022). Promix: Combating label noise via maximizing clean sample utility. *arXiv preprint arXiv:2207.10276*.

Wisdom, S., Hershey, J. R., Wilson, K., Thorpe, J., Chinen, M., Patton, B., and Saurous, R. A. (2019). Differentiable consistency constraints for improved deep speech enhancement. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 900–904.

Wisdom, S., Tzinis, E., Erdogan, H., Weiss, R., Wilson, K., and Hershey, J. (2020). Unsupervised sound separation using mixture invariant training. *Advances in Neural Information Processing Systems*, 33:3846–3857.