ACOUSTIC SOUND RECORDING AND MUSIC SOUND SEPARATION

Loïs Guerci, Louis Gosselin, Samy Benzaïm, Antonin Longeot, Jean-Henri Nothias

IRCAM, Sorbonne Université & Telecom Paris

ABSTRACT

Over the past several years, significant progress has been made in Music Source Separation (MSS). From the vibration of musical instruments to the audio signal present in a finalized recording, numerous factors come into play. These include the acoustic radiation of the instrument, the movements associated with musical performance, the room's response to a moving acoustic source, the microphone array configuration, and the transformations applied during post-production. Due to the complexity and variability of these factors, establishing a new state-of-the-art approach remains challenging. From this perspective, the objective of source separation is to design algorithms capable of virtually reversing this entire chain of transformations and mixing processes in order to estimate the signal produced by each individual musical instrument. This can be achieved by incorporating prior knowledge about both the structure of the source signals and the nature of the transformations they have undergone. In this paper, we present a novel hybrid source separation pipeline that leverages CLAP-based Language-Queried Audio Source Separation (CLASS-net) [1] as a prior for initializing the Fast Multichannel Non-Negative Matrix Factorization (FastMNMF) algorithm. Despite the complexity of the task, we demonstrate that the model can achieve respectable results compared to the state-of-the-art methods. The separated audio samples, pictures of the record session and source code are available at : `https://loguerci.github.io/CLASS_website/`

Index Terms— multi channel source separation, music information retrieval, data augmentation, sound recording,

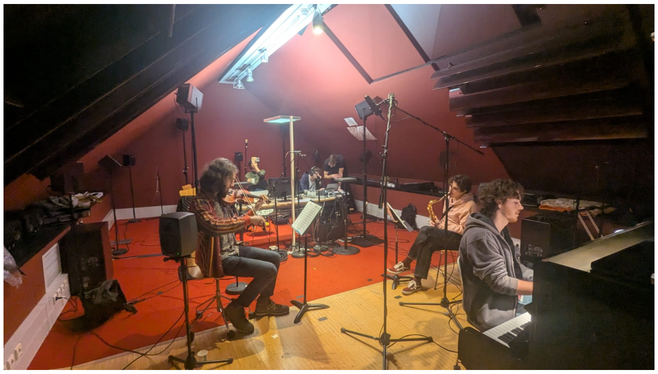

Fig. 1. Trio Pop music session

1. INTRODUCTION

In Blind Source Separation (BSS), the objective is to recover unobserved source signals from an observed mixture. One of the main challenges, particularly in multichannel recording scenarios, is to accurately isolate and localize each individual source. A well-known example of this problem arises in the recording of a musical orchestra, where multiple instruments, each corresponding to a distinct source, are captured simultaneously. In such settings, the difficulty of the separation task increases significantly depending on the acoustic characteristics of the room, as well as the number and spatial configuration of the microphones. In recent years, substantial progress has been achieved through deep learning–based approaches. Among the most effective methods, detailed in Section 2, Rouard et al. introduced the Hybrid Transformer Demucs model [2], which achieves remarkable performance. This architecture processes both the raw waveform and its time–frequency representation, obtained via the Short-Time Fourier Transform (STFT), in parallel. It is designed to separate music mixtures into up to four stems (drums, bass, vocals, and other), leveraging both temporal and spectral information for improved separation quality. A major limitation of Demucs is that it operates only on mono or stereo waveforms, meaning it can handle at most two channels, and therefore does not explicitly exploit the spatial information of individual sources. In other words, while it outperforms many state-of-the-art approaches on instantaneous mixtures, it remains limited when dealing with acoustic mixtures, where spatial cues play a crucial role. In such scenarios, more established methods such as Multi-channel Non-negative Matrix Factorization (MNMF) [3] remain highly competitive. However, deep learning techniques can still be leveraged to mitigate some of the limitations of MNMF. In particular, neural networks can be used to refine or guide the separation process, a strategy commonly referred to as speech enhancement.

We then present, as part of the ATIAM master's program, our approach to separating a musical trio composed of a tenor saxophonist, a violinist, and a pianist, recorded simultaneously in an IRCAM studio, as if each instrument had been captured independently, as shown in Fig. 2.

Building on this recording setup, we propose a hybrid method that combines a Language-Queried Audio Source Separation network (operating on a mono channel) with the FastMNMF algorithm. In this framework, the neural network is used for prior initialization, while FastMNMF performs the subsequent spatial reconstruction of the sources. The LASS-net model [1] is a separation network that extract a source from mixture based on a textual prompt, describing the given source. In out approach, the text encoder that embed the prompt has been replaced by a Constractive Language Audio Pretraining model (CLAP), which will be detailed in Section 3. We also perform a data augmentation based on the Slakh dataset [4] and our sound recording to fine-tune our model.

2. RELATED WORK

2.1. Acoustic sound recording

The way sound is recorded is crucial for the post-production workflow. While the literature mainly focuses on how the recording sounds to the listener, the physical properties of the microphone setup, like coherence and directivity, determine the quality of the signal. This quality is what makes subsequent processing, like source separation, possible or not.

2.1.1. Stereo Techniques and Signal Correlation

Recording techniques can be divided into four main categories: Coincident, Near-Coincident, Spaced, and Multi-microphone techniques [5]. Coincident techniques, such as XY or MS (Mid-Side), use intensity differences rather than time differences. As noted in [5], these setups produce a very stable stereo image where the left and right channels are highly correlated. On the other hand, spaced techniques use time delays to capture a more diffuse sound field. This makes the sound feel more ”spacious,” but it also means the signals are less correlated. [5] also points out that the choice of microphone directivity (for example, cardioid vs. omnidirectional) changes the balance between the direct sound and the reverb. This choice depends on the context, such as the size of the room or whether the audio will be played back on headphones or loudspeakers.

2.1.2. Multichannel Arrays: Localization vs. Envelopment

For recording orchestral music in immersive formats, [6] compares six different microphone arrays, including the DECCA Tree, 2L-Cube, ESMA-3D, Triple-MS, and Square Antiprism. Using perception tests based on simulated recordings, the study highlights a clear trade-off:

- **Precision of Localization:** The study found that the *Square Antiprism* (which estimates second-order Ambisonics) and the *Triple-MS* system gave the best localization. The authors explain that this is because these coincident (or near-coincident) arrays preserve strong correlation between the channels and have stable Interaural Time Differences (ITD) [6].
- **Envelopment:** However, these coincident arrays did not score as well on ”Envelopment” (the feeling of being surrounded by sound). Spaced arrays like the *ESMA-3D* performed better here because they capture more non-frontal late energy. This suggests that if you want precise localization, you often lose some of the immersive room effect [6].

2.1.3. Implications for Acquisition Strategies

The results in [6] suggest that one single microphone array is not enough to get everything right. The authors conclude that coincident systems are great for localization—which fits well with object-based audio—but they do not fully capture the ”sense of space” of a real performance. Therefore, the standard solution is a hybrid approach: using a main array to define the general space, combined with spot microphones to add precision and definition to specific instruments.

Even though these articles do not explicitly discuss BSS, their findings are important for this project. The ”Precision of Localization” mentioned in [6] is a key factor: arrays that keep directional cues clear (high localization) should theoretically be easier to process with separation algorithms, whereas arrays that focus on envelopment might make separation harder due to the lack of correlation.

2.2. Source separation algorithm

2.2.1. Classical methods

Classical source separation methods are characterized by algorithms adapted to explicit mixing setups and environmental models. One of the most influential model-based approaches is Non-negative Matrix Factorisation (NMF) introduced by Lee and Seung [7]. This algorithm has proven highly effective in a single channel context, but suffers from major limitations : this factorization assumes that sources have distinct spectral characteristics and can be represented as linear combinations of a limited number of spectral patterns. This can be remedied in the multichannel context, with multi-channel NMF, which exploits spatial information gained from multiple non-colocalized sources, in the form of covariance across channels. The spatial covariance for each basis encodes the mixing characteristics, while the NMF component models spectral content. Févotte et al. [3] introduced a multichannel Gaussian model in which each NMF component is associated with a spatial covariance matrix and derived update rules using Itakura-Saito divergence. Experiments showed that with two microphones and three instruments, MNMF can separate sources by leveraging joint spectral-spatial structure [8]. However, this approach assumes time-invariant mixing (sources do not move) and inherits the spectral limitations of standard NMF, while requiring substantially more computational resources for joint optimization.

For underdetermined or reverberant scenarios, spatial covariance models are necessary. Rank-1 spatial covariances model instantaneously or convolutively mixed point sources, assuming a single direct path from source to microphones with low reverberation. [3] While computationally efficient, these models fail in reverberant environments where reflections create complex spatial patterns. Full-rank spatial covariance models address this by allowing each source's contribution in each time-frequency bin to be represented as a zero-mean Gaussian with a full covariance matrix encoding complete spatial mixing characteristics. Duong, Vincent & Gribonval [9] introduced such models to capture reverberation and complex spatial spreads, deriving an expectation-maximization (EM) algorithm to estimate covariance matrices and spectral variances from mixtures. This enables learning multichannel Wiener filters optimal for Gaussian models, improving separation in reverberant audio compared to rank-1 models.

Other methods include *independent component analysis* (ICA) and *independent vector analysis* (IVA) : ICA, introduced in [10], recovers sources by finding a demixing matrix that produces statistically independent outputs, assuming sources are non-Gaussian. This approach works for determined mixtures (equal number of microphones and sources) but suffers from a permutation problem in the frequency domain, where output order can differ across frequency bins. [11] IVA, introduced in [12] addresses this by treating each source as a vector across all frequencies, enforcing consistent permutations through full-band statistical modeling. For determined cases, Independent Low-Rank Matrix Analysis (ILRMA) unifies IVA with NMF source models, combining frequency-domain ICA with low-rank spectral structure. [13] ILRMA achieved state-of-the-art results for determined music separation in the mid-2010s. However, ICA-based methods require at least as many microphones as sources and assume statistical independence, which is systematically violated in music where instruments share harmonic content and correlated rhythms.

2.2.2. *Evaluation*

Evaluating a source separation algorithm is challenging, as there is often a gap between human perception of audio quality and the objective metrics used to measure it. In the context of BSS, two types of evaluation metrics are commonly used.

The first type consists of objective metrics, originally proposed by Vincent et al. [14]. Useful, faster and cheaper to obtain, those objectives measures are the Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), and Source-to-Artifact Ratio (SAR). They allow us to compare the target source modified by an allowed distortion s_{target} to the predicted one $\hat{s_i}$ and assume that the latest is made of four components defined as

$$\hat{\mathbf{s}}_{\text{i}} = \mathbf{s}_{\text{target}} + \mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{artif}} \tag{1}$$

where e_{interf}, e_{noise}, and e_{artif} are error terms for interference, noise, and added artifacts, respectively. Those components represent the part of perceived as $\hat{s_i}$ coming from the target source s_i , from other unwanted sources s_j $(i \neq j)$, from sensor noises $(n_i)_{1 \leq i \leq m}$, and from other causes. Their decompositions are based on orthogonal projections, with $\Pi_{\{y_1,..,y_k\}}$ the orthogonal projector onto the subspace spanned by vectors $y1, .., yk$ of length T, in the shape of a T × T matrix. Three orthogonal projectors are defined by

$$\begin{aligned} \mathbf{P}_{\mathbf{s}_j} &:= \Pi_{\{\mathbf{s}_j\}} \\ \mathbf{P}_{\mathbf{s}} &:= \Pi_{\{\mathbf{s}_{j'}\}_{1 \leq j' \leq n}} \\ \mathbf{P}_{\mathbf{s},\mathbf{n}} &:= \Pi_{\{\mathbf{s}_{j'}\}_{1 \leq j' \leq n}, \{\mathbf{n}_i\}_{1 \leq i \leq m}} \end{aligned} \tag{2}$$

and the following components defined above become :

$$\begin{aligned} \mathbf{s}_{\text{target}} &:= \mathbf{P}_{\mathbf{s}_j}\, \hat{\mathbf{s}}_j \\ \mathbf{e}_{\text{interf}} &:= \mathbf{P}_{\mathbf{s}}\, \hat{\mathbf{s}}_j - \mathbf{P}_{\mathbf{s}_j}\, \hat{\mathbf{s}}_j \\ \mathbf{e}_{\text{noise}} &:= \mathbf{P}_{\mathbf{s},\mathbf{n}}\, \hat{\mathbf{s}}_j - \mathbf{P}_{\mathbf{s}}\, \hat{\mathbf{s}}_j \\ \mathbf{e}_{\text{artif}} &:= \hat{\mathbf{s}}_j - \mathbf{P}_{\mathbf{s},\mathbf{n}}\, \hat{\mathbf{s}}_j \end{aligned} \tag{3}$$

Finally , those definitions allow us to define the three metrics as follow :

$$\text{SAR} = 10 \log_{10} \left(\frac{||\mathbf{s}_{\text{target}} + \mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}}||^2}{||\mathbf{e}_{\text{artif}}||^2} \right) \tag{4}$$

$$\text{SIR} = 10 \log_{10} \left(\frac{||\mathbf{s}_{\text{target}}||^2}{||\mathbf{e}_{\text{interf}}||^2} \right) \tag{5}$$

$$\text{SDR} = 10 \log_{10} \left(\frac{||\mathbf{s}_{\text{target}}||^2}{||\mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{artif}}||^2} \right) \tag{6}$$

Since SDR is sensitive to amplitude scaling of the signal, it can artificially inflate the SNR value without any perceptual improvement. To address this issue, Le Roux et al. [15] introduced the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), which removes the dependency on scaling. This is achieved by rescaling the target signal such that the residual error is orthogonal to it, which corresponds to projecting the estimated signal $\hat{s}$ onto the subspace spanned by the target signal s. Formally, this amounts to define

$$\text{SI-SDR} = 10 \log_{10} \left(\frac{||\mathbf{e}_{\text{target}}||^2}{||\mathbf{e}_{\text{res}}||^2} \right) \tag{7}$$

where $e_{\text{target}} = \alpha s$, with $\alpha = \arg\min_{\alpha} |\alpha s - \hat{s}|^2$, and $e_{res} = \hat{s} - e_{target}$

More generally, objective measures remain limited, as they struggle to capture the full richness of human auditory perception. Many perceptual aspects of sound quality are difficult to model computationally. Furthermore, these metrics require access to ground-truth signals, which is typically unavailable in our BSS scenarios. However, in the case of the CLASS-net, the four metrics are useful to evaluate the model base on our data augmentation (detailed in Section 4). A higher number indicates a better performance.

The second type of evaluation relies on subjective metrics. In particular, we consider the MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA) protocol for assessing intermediate audio quality. This method, described in [16], follows the recommendations of the International Telecommunication Union (ITU). It consists of a listening test in which participants are presented with a reference signal (e.g., the original mixture) alongside several test samples (e.g., outputs of a source separation algorithm). Among these samples, one corresponds to a hidden reference, while others include anchors, which are degraded versions of the reference signal. These anchors typically consist of low-pass filtered versions of the reference (e.g., at 7 kHz and 3.5 kHz), providing perceptual baselines for evaluation.

However, due to time constraints, we were not able to conduct such a subjective evaluation in this work.

2.2.3. *Deep Learning methods*

One of the first successful applications of deep models into source separation was proposed by Nugraha et al. [17]. In their work, deep neural networks are used to model the source spectra and combined with the classical multichannel Gaussian model to exploit the spatial information. This hybrid approach allows the exploitation of spatial information while benefiting from the representational power of neural networks, marking a key transition from purely statistical models to learned models. Another influential paradigm is deep clustering [18], which reformulates source separation as an embedding learning problem. Instead of directly estimating signals or masking functions, the network learns embeddings for each time-frequency bin such that components belonging to the same source are close in the embedding space. Source separation is then performed by clustering these embeddings. This approach addresses the permutation ambiguity inherent in multi-source separation and has inspired many subsequent methods. Recent methods focus on end-to-end neural architectures trained to directly estimate source signals or time-frequency masks. Rouard et al. [2] propose a Hybrid Transformer Demucs model that combines two U-Net architectures: one operating in the time domain using temporal convolutions, and another operating in the frequency domain using spectrogram representations. The integration of Transformer layers enables long-range temporal modeling, resulting in state-of-the-art performance on standard music separation benchmarks. However, these models generally assume direct mixing conditions and may struggle to generalize to reverberant acoustic environments. To address the intrinsic ambiguity of source separation, generative approaches have recently gained attention. Scheibler et al. [19] propose a diffusion-based framework for single-channel speech source separation, modeling the separation process as the reverse of a stochastic differential equation (SDE). By explicitly modeling uncertainty, diffusion models can generate multiple plausible source estimates and demonstrate increased robustness to noise and reverberation compared to deterministic methods. Speaking of methods, Wisdom et al. [20] propose an unsupervised method, mixture invariant training (MixIT), that requires only single-channel acoustic mixtures. They improve reverberant speech separation performance by incorporating mixtures that depend on the acoustics of the room.

More recently, CLAP [21] extends this idea by learning a joint embedding space between encoded audio and encoded text through contrastive learning. originally not designed for source separation, CLAP enables semantic conditioning of audio models, allowing text-guided source separation. Conditioning mechanisms such as Feature-wise Linear Modulation (FiLM) [22] provide a flexible way to integrate such auxiliary information into generative models sush as Latent Diffusion conditioned models. These approaches are particularly promising for complex acoustic mixtures, where semantic and contextual information can help disambiguate sources. Based on this approach, Liu et. al. in their paper "Language-Queried Audio Source Separation" [1], propose a way to separate a source from a mixture based on textual prompt.

3. METHODOLOGY

In this section, we describe our hybrid pipeline, covering the entire process from the sound recording setup to the final algorithmic design.

3.1. FastMNMF2 Framework

We propose a source separation framework that integrates a semi-supervised spectral dictionary and a physics-guided spatial initialization into the Fast Multichannel Non-negative Matrix Factorization 2 (FastMNMF2) algorithm [**?**].

3.1.1. Generative Model

Let $\mathbf{x}_{ft} \in \mathbb{C}^M$ denote the Short-Time Fourier Transform (STFT) coefficients of the mixture recorded by M microphones at frequency f and time index t. Following the Weight-Shared Jointly-Diagonalizable (WJD) spatial model, the mixture is assumed to follow a multivariate complex Gaussian distribution:

$$\mathbf{x}_{ft} \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \sum_{n=1}^{N} \lambda_{n,ft}\mathbf{R}_{n,f}\right). \tag{8}$$

The model parameters are structured as follows:

1. **Spectral Model:** The source Power Spectral Densities (PSDs), $\lambda_{n,ft}$, are factorized via NMF:

$$\lambda_{n,ft} = \sum_{k=1}^{K} w_{nkf} h_{nkt}, \tag{9}$$

   where $w_{nkf} \geq 0$ and $h_{nkt} \geq 0$. In our proposed semi-supervised approach, $\mathbf{W}$ is fixed to a pre-learned dictionary.

2. **Spatial Model:** The Spatial Covariance Matrices (SCMs) are modeled using a frequency-dependent joint diagonalizer $\mathbf{Q}_f \in \mathbb{C}^{M\times M}$ and frequency-invariant spatial weights $\tilde{\mathbf{g}}_n = [\tilde{g}_{n1}, \ldots, \tilde{g}_{nM}]^{\mathsf{T}} \in \mathbb{R}^M_+$:

$$\mathbf{R}_{n,f} = \mathbf{Q}_f^{-1}\,\mathrm{Diag}(\tilde{\mathbf{g}}_n)\mathbf{Q}_f^{-\mathsf{H}}. \tag{10}$$

   To ensure identifiability, we impose the normalization constraints $\sum_{m=1}^{M} \tilde{g}_{nm} = 1$ and $\mathrm{tr}(\mathbf{Q}_f\mathbf{Q}_f^{\mathsf{H}}) = M$.

3.1.2. Initialization Strategy

We introduce a robust initialization scheme to mitigate the sensitivity of MNMF to initial conditions.

3.1.2.1. Spectral Initialization

The temporal activations $\mathbf{H}$ are initialized by projecting initial source estimates (obtained via a pre-trained Deep Neural Network) onto the fixed dictionary $\mathbf{W}_{\text{dict}}$. Specifically, for each source n, $\mathbf{H}_n^{(init)}$ is obtained by solving a Non-negative Least Squares (NNLS) problem.

3.1.2.2. Physics-Guided Spatial Initialization

We initialize the spatial parameters based on the geometry of the microphone array. We compute the steering vectors $\mathbf{a}_{n,f} \in \mathbb{C}^M$ for all sources using a spherical wave propagation model. The theoretical mixing matrix is formed as $\mathbf{A}_f = [\mathbf{a}_{1,f}, \ldots, \mathbf{a}_{N,f}]$.

The joint diagonalizer $\mathbf{Q}_f$ is initialized as the Moore-Penrose pseudo-inverse of the mixing matrix:

$$\mathbf{Q}_f^{(0)} = \begin{bmatrix} \mathbf{A}_f^{\dagger} \\ \mathbf{U}_{\text{null}} \end{bmatrix}, \tag{11}$$

where $\mathbf{U}_{\text{null}}$ spans the null space if $M > N$. Accordingly, the spatial weights $\tilde{\mathbf{G}}$ are initialized to be diagonally dominant ($\tilde{g}_{nm} \approx \delta_{nm}$ for $n \leq N$), ensuring the n-th output of the diagonalizer corresponds to the n-th source.

3.1.3. Parameter Estimation

The parameters $\Theta = \{\mathbf{H}, \tilde{\mathbf{G}}, \mathbf{Q}\}$ are estimated by maximizing the log-likelihood function. Let $\tilde{x}_{ftm} = |\mathbf{q}_{fm}^{\mathsf{H}}\mathbf{x}_{ft}|^2$ denote the power of the decorrelated components, and $y_{ftm} = \sum_{n=1}^{N} \lambda_{n,ft}\tilde{g}_{nm}$ be the model variance.

The update rules, derived for FastMNMF2, are applied iteratively:

- **Temporal Activations (H):**

$$h_{nkt} \leftarrow h_{nkt}\sqrt{\frac{\sum_{f,m} w_{nkf}\tilde{g}_{nm}\tilde{x}_{ftm}y_{ftm}^{-2}}{\sum_{f,m} w_{nkf}\tilde{g}_{nm}y_{ftm}^{-1}}}. \tag{12}$$

- **Spatial Weights ($\tilde{\mathbf{G}}$):** Since $\tilde{\mathbf{g}}_n$ is shared across frequencies, the update aggregates information over all frequency bins:

$$\tilde{g}_{nm} \leftarrow \tilde{g}_{nm}\sqrt{\frac{\sum_{f,t} \lambda_{n,ft}\tilde{x}_{ftm}y_{ftm}^{-2}}{\sum_{f,t} \lambda_{n,ft}y_{ftm}^{-1}}}. \tag{13}$$

  Note that $\lambda_{n,ft} = \sum_k w_{nkf}h_{nkt}$ allows efficient computation.

- **Joint Diagonalizer (Q):** We use the Iterative Projection (IP) method. For each channel m, the row $\mathbf{q}_{fm}$ is updated by solving the linear system:

$$\mathbf{q}_{fm}^{\mathsf{H}} \leftarrow (\mathbf{Q}_f\mathbf{V}_{fm})^{-1}\mathbf{e}_m, \tag{14}$$

  where $\mathbf{V}_{fm} = \frac{1}{T}\sum_{t=1}^{T} y_{ftm}^{-1}\mathbf{x}_{ft}\mathbf{x}_{ft}^{\mathsf{H}}$ is the weighted empirical covariance matrix.

Scale ambiguities are resolved at each iteration by normalizing $\mathbf{Q}$ and $\tilde{\mathbf{G}}$ according to the constraints defined in the generative model section.

Table 1. Coordinates of the sources in the relation to the ambiance cardioid

Sources	X	Y	Z
Piano	6.0	0.0	0.6
Saxophone	4.5	1.0	0.6
Violin	4.5	-1.0	0.6

Table 2. Coordinates of the microphones in the relation to the ambiance cardioid

Microphone	X	Y	Z	Type
1-2-3	3.0	0.0	1.5	cardioid
4-Ambiant	0.0	0.0	1.8	cardioid
5-Piano	6.0	0.0	1.9	bidirectional
6-Saxo	4.5	0.5	1.7	bidirectional
7-Violin	4.5	-0.5	1.7	bidirectional

3.2. Sound recording

3.3. Dataset augmentation

A fundamental challenge in supervised music source separation is the scarcity of high-quality, multi-instrument recordings with isolated stems. While large-scale datasets exist for certain instruments or mixing conditions, recordings that match the specific acoustic characteristics of our target scenario—a violin, saxophone, and piano trio captured in a reverberant studio environment—are extremely limited. To address this limitation and enable robust training of the CLASS-net model, we developed a hybrid data augmentation strategy that combines synthetic mixtures from the Slakh2100 dataset [4] with our own multi-instrument recordings acquired at IRCAM.

3.3.1. Source Datasets

3.3.1.1. Slakh2100 Dataset

The Slakh2100 dataset provides a large-scale collection of automatically mixed multi-track audio derived from the Lakh MIDI dataset. Each track in Slakh2100 contains isolated stems for individual instruments rendered using high-quality virtual instruments, along with metadata specifying the MIDI program names. We utilize the training split of the Slakh2100 dataset resampled to 16 kHz, which contains diverse instrumental combinations across multiple musical genres.

To focus on acoustically similar instruments and avoid domain mismatch, we apply a keyword-based filtering strategy. Specifically, we retain only stems corresponding to instruments in our target set: violin, piano, and saxophone. Furthermore, we explicitly reject stems containing the keywords ”electric” or ”synth,” as these exhibit fundamentally different spectral characteristics from their acoustic counterparts.

3.3.1.2. IRCAM Studio Recordings

To improve the model's performance on our particular source separation task, we augmented the Slakh2100 data with recordings from our own studio session. As described in Section 3, we recorded a trio consisting of violin, tenor saxophone, and piano performing three popular songs in a controlled studio environment at IRCAM, along with isolated performances of each instrument under identical acoustic conditions. These recordings provide in-domain data that will allow our model to be fine-tuned to the particular instruments that were used during the test recordings.

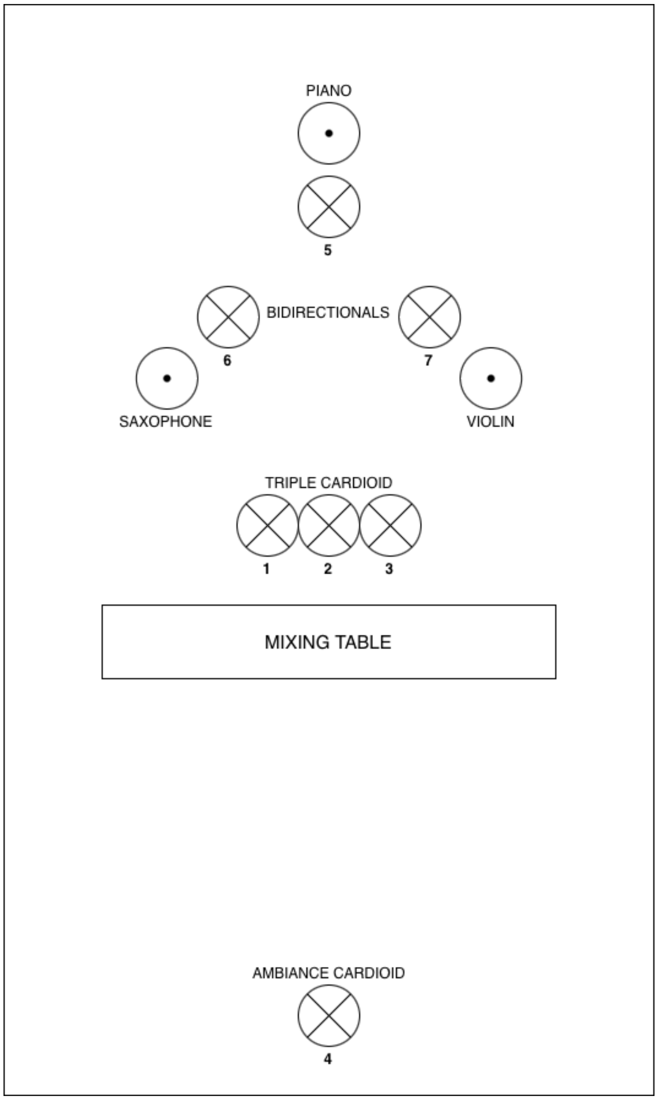

Fig. 2. Setting of the 3 sources and the 7 microphones in the studio

While we tried to record every instrument separately with dedicated microphones during the song recordings, the presence of acoustic reflectors in the studio lead to abundant cross-channel bleeding despite the microphones being directive and setup to minimize direct interference from other instruments. Consequently, these stems were not usable as targets to fine-tune the model, and only the solo performances were used. We integrated the remaining files into our dataset under the labels `violin_rec`, `saxophone_rec`, and `piano_rec` to distinguish them from the Slakh2100 sources.

3.3.2. Mixture Synthesis Procedure

Our data augmentation pipeline generates synthetic training examples by constructing controlled mixtures that simulate the separation task. Each training example consists of a triplet: a reference audio segment, a target source, and a mixture containing the target source embedded within a background of interfering instruments.

3.3.2.1. Instrument Selection and Sampling

In the following section, random sampling was always done using a uniform distribution.

For each generated datapoint, we first randomly select a target instrument from the filtered pool containing violin, piano, and saxophone. This inclues both Slakh and recorded variants, with the recorded variants being chosen 50% of the time. This target instrument represents the source that the model must learn to extract. A corresponding stem is randomly sampled from the available paths for that instrument class. If the target originates from our recorded data (e.g., "`saxophone_rec`"), the prompt is mapped to its canonical form (e.g., "`saxophone`") to ensure consistency with the CLAP text encoder.

To create a realistic mixture, we select between 1 and 3 additional instruments to serve as interferers. These background instruments are sampled uniformly from the entire Slakh2100 dataset combined with our recordings, subject only to the constraint that they must not belong to the same instrument class as the target. This ensures that the model learns to discriminate between the target and acoustically diverse distractors. The dataset is sanitized dynamically at this step, preventing unrendered stems and missing instruments from being chosen again, thus speeding up the data generation process.

3.3.2.2. Temporal Scattering and Active Audio Extraction

To increase temporal diversity and prevent the model from relying on fixed temporal patterns, we apply a scattering procedure that randomly places audio segments throughout the mixture duration, implementing the following steps.

1. The mixture timeline is divided into 5 non-overlapping temporal bins of equal duration of 2 seconds each.
2. For each bin, for each instrument, with a probability of 80%, a random audio segment of at most 3 seconds is inserted at a random position within that bin.
3. If an audio segment s of the same instrument overlaps the current bin, the possible insertion positions for a new audio segment restricted from the end of s to the end of the current bin.

Before scattering, all audio is preprocessed in order to keep only portions of the stems where audio is present. This is done by applying a low-pass filter on the absolute value of the waveform w to obtain an array y, and keeping only the values $w(n)$ such that $y(n) > T$ where T is a threshold value. This ensures that the mixture does not contain extended silent regions that would be trivial to separate.

This whole process guarantees medium to hard training examples for the model, and allows the generation of large amounts of varied training examples from relatively little recorded data.

3.3.2.3. Final Mixture Construction

The complete training example is constructed as follows:

- **Reference:** A clean segment of the target instrument class, randomly sampled from the filtered Slakh2100 stems and recordings. The reference is preprocessed as above and normalized to ensure consistent amplitude. It is then truncated to a 10 second clip, and scattering is *not* applied.
- **Target:** The selected target stem is preprocessed, normalized and scattered over a duration of 10 seconds.
- **Mixture:** The sum of the scattered target and all scattered background instruments. The final mixture is normalized, and silences are kept if they occur, for a final duration of 10 seconds.

Additionally, we generate a metadata JSON file for each training example, recording the prompt keyword (target instrument), for use in CLAP.

3.3.3. Implementation Details

All audio is processed at a sampling rate of 16 kHz to match the input requirements of the CLAP encoder and to reduce computational cost. The STFT parameters used in CLASS-net (frame size of 1024, hop size of 256) are consistent with this sampling rate and yield a time-frequency representation of dimension 513×626 for a 10 second audio clip.

Dataset generation is performed offline, and the resulting triplets (reference, target, mixture) are saved as WAV files in separate directories, along with the JSON metadata contatining the prompt. Overall, 1000 examples (2 hours and 47 minutes of audio) were generated for training, 200 (34 minues) for testing and 300 (50 minues) for evaluation.

3.4. CLASS-net model

Prior to the initialization of the FastMNMF algorithm, we propose an alternative model inspired by the work of Liu et al. As shown in Fig. 3, our approach, termed CLASS (CLAP-based Language-Queried Audio Source Separation), replaces the text encoder used in LASS-net with a pre-trained CLAP model [21].

As described in Section 2, CLAP is trained on large-scale audio–text pairs and learns a joint embedding space that captures the relationship between audio content and its natural language description. This modification enables the incorporation of an additional conditioning signal in the form of an audio reference, guiding the model to extract the desired source from the mixture. Specifically, CLAP generates a joint embedding from both an audio reference (e.g. a violin sound) and a textual tag (e.g. "violin"), which is then used as a conditional input to train the separation network. The CLAP embedding z_{CLAP}, defined as

$$\mathbf{z}_{\text{CLAP}} = \frac{\frac{1}{2}\left(\mathbf{z}_{\text{audio}} + \mathbf{z}_{\text{text}}\right)}{\left\|\frac{1}{2}\left(\mathbf{z}_{\text{audio}} + \mathbf{z}_{\text{text}}\right)\right\|_2} \tag{15}$$

where z_{audio} and z_{text} are the audio reference encoded through the HT-SAT encoder [23], and the tag prompt through the RoBERTa encoder [24] respectively, is injected into the network via FiLM (Feature-wise Linear Modulation) [22], applied to the output of each convolutional block in both the encoder and decoder. The separation backbone is based on the ResUNet architecture proposed by Kong et al. [25]. The overall model is fine-tuned from weights pre-trained on LASS-net. The predicted spectrogram mask is first multipled to the mixture to compute the time-frequency predicted source $\hat{S}$, which is then compared to the target source S through the Mean Absolute Error (MAE) given by

$$\mathcal{L}_{\text{MAE}} = \left\| |\mathbf{S}| - |\hat{\mathbf{S}}| \right\|_1 \tag{16}$$

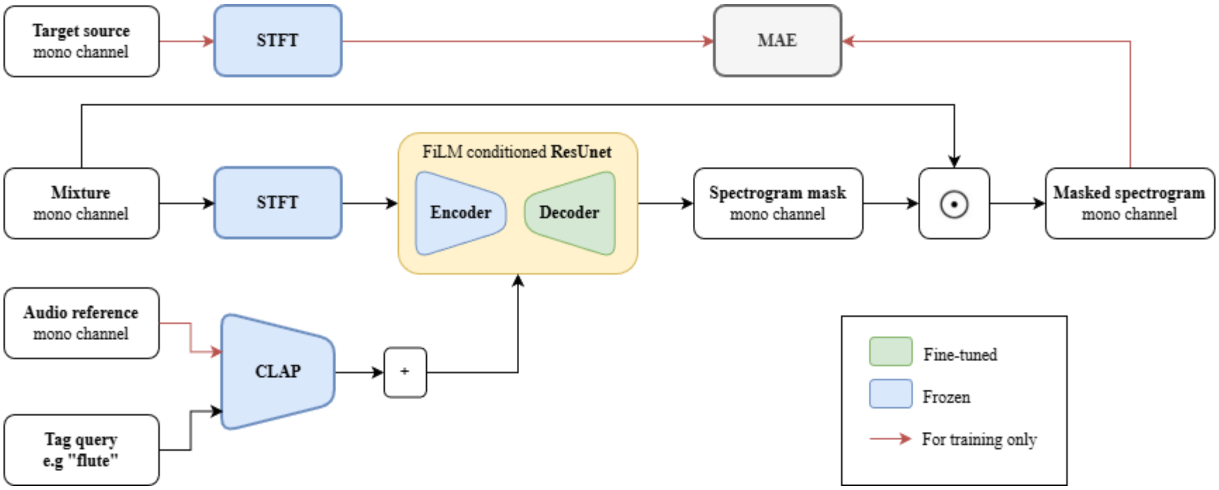

Fig. 3. Framework of the proposed CLASS-Net

In practice, we freeze the CLAP encoder, as it has already been trained on large-scale musical data, and restrict fine-tuning to the decoder part of the ResUNet. This strategy stabilizes training while allowing the separation network to adapt to our specific task. Since CLASS-net processes only monaural audio, we select one microphone from our multichannel recording setup at inference time to predict the desired source given the text prompt. To properly initialize the FastMNMF algorithm, we use CLASS-net to estimate the separated mono time–frequency representation V_n, where $n = 3$ is the number of sources/instruments. We then estimate the note activations H using a Non-negative Least Squares (NNLS) procedure, described in Section 3.1.2.

4. EXPERIMENTS AND RESULTS

In this section, we provide the training and evaluation procedure regarding the CLASS-net module and the FastMNMF2 algorithm.

4.1. CLASS-net training and evaluation

4.1.1. Data processing

For training simplicity and with respect to the specific Slakh dataset from Chang et al. paper [26] (see Subsection 3.3), we load from our dataset audio signals using 16KHz sampling rates. The STFT is performed using a frame size of 1024 and a hop size of 256. The mixture and the target source of 10 seconds results in a spectrogram with shape of 513 × 626.

4.1.2. Updating LASS-net baseline

Compared to the original LASS-net, the audio reference of 10 seconds and its textual description are directly handled by CLAP. The embedding produced by the encoder z_{CLAP}, has a shape of batch size x 512. We added a fully-connected layer with 256 nodes followed by a ReLU activation to match the rest of the architecture. As detailed in Subsection 3.4, we decided to fine-tune only the decoder part of the ResUnet, taking the best checkpoint model from the LASS-net. The rest of the architecture are completly frozen. We used the *music audioset* best checkpoint model on CLAP. Similar to the LASS-net training, we set batch size to 2. AdamW [27] optimizer is used for training with the learning rate of 1 x 10^{-4}. We train CLASS-net for 25 000 iterations using one Nvidia-Tesla-K80-11GB GPU.

4.1.3. Result on objective metrics

We evaluate our model using four commonly used metrics in source separation, as detailed in Section 2. The evaluation is conducted on a held-out test subset derived from our augmented dataset. We compare our results with the best reported performance of LASS-Net in its original evaluation setting (see Section 5.5 of [1]). To compute those objective metrics, we used a fast implementation proposed by Scheibler [28]. Since metrics such as SDR, SIR, SAR and SI-SDR primarily assess the quality of the separated sources, we do not perform a direct comparison on a shared dataset. Indeed, LASS-Net was not trained on musical data, making such a comparison less meaningful.

Table 3. Models Performance Summary.

Model	**SDR**	**SIR**	**SAR**	**SI-SDR**
LASS-net	5.89	16.70	5.18	4.86
CLASS-net	4.15	-	4.15	2.83

As shown in Table 3, CLASS-Net does not outperform the original LASS-Net, although the results remain respectable. Regarding the SIR metric, it cannot be reliably evaluated in our setting. Indeed, due to our data augmentation procedure, which considers only a single target source, it is not possible to properly assess interference between multiple sources. Furthermore, qualitative listening on an example from the recorded dataset reveals that, while the piano part can generally be distinguished from the mixture, the violin is sometimes confused with the tenor saxophone. Overall, we find it encouraging that the model performs reasonably well given the training constraints. Audio examples are available on our demo page. These results can primarily be attributed to the limited amount of data available for both training and validation. In addition, time constraints prevented a more thorough exploration of the model architecture and training procedure. Additionally, due to GPU memory limitations, only a small batch size (up to 4) was feasible for training the model. One potential direction for improvement lies in the CLAP-based conditioning. During training, the model is exposed to a balanced combination of text and audio inputs, whereas inference may rely solely on textual conditioning, leading to a mismatch that can affect performance. Furthermore, fine-tuning the CLAP model jointly with the training process could potentially improve the quality of the learned representations and, consequently, the overall performance.

In addition, several directions could be explored in future work. First, the use of MAE as a reconstruction metric introduces limitations due to the inherent trade-off between temporal and frequency resolution. As a result, it may fail to accurately capture fast transients or fine harmonic structures. An improved loss formulation could be considered by drawing inspiration from the approach of Kreuk et al. [29]. In their work, the authors combine a time-domain MAE between the target and reconstructed signals with a Multi-Resolution STFT loss, originally proposed by Yamamoto et al. [30]. Such a combination enables a more comprehensive characterization of both temporal and spectral aspects of the audio signal, and could lead to improved reconstruction quality.

4.2. FastMNMF2 training and evaluation

FastMNMF2 nP corresponds to the performance of the algorithm without prior information (the output of CLASS-net) and FastMNMF2 P corresponds to the algorithm initialized with prior information.

Table 4. Algorithms Performance Summary.

Algorithm	**SDR**	**SIR**	**SAR**	**SI-SDR**
FastMNMF2 nP (4 channels)	.	.	.	.
FastMNMF2 nP (7 channels)	.	.	.	.
FastMNMF2 P (7 channels)	.	.	.	.

5. CONCLUSION AND PERSPECTIVES

6. ACKNOWLEDGMENTS

We would first like to thank Benoît Fabre, Professor of Musical Acoustics and Sound Techniques at CRR93 and teacher-researcher at the Institut d'Alembert, Sorbonne University, for his valuable contributions to this project, particularly regarding sound recording procedures and techniques. His expertise greatly guided us in the preparation and realization of the recording session. We also sincerely thank Mathieu Fontaine, Professor and Researcher in the ADASP group at Télécom Paris, for his continuous support throughout this project, especially with respect to the source separation pipeline. His knowledge and experience in the field helped us better understand and implement the model architecture, data augmentation strategies, and the integration of the model within the FastMNMF algorithm. He also provided us an access to one GPU from Télécom Paris to perform the training of the model.

We would also like to thank IRCAM, and in particular Cyrielle Fiolet and Grégoire Lavaud, for providing us with access to their studio facilities and the necessary recording equipment. Finally, we warmly thank Simon Luko (tenor saxophonist), Simon Jacquin (violinist), and Emilio Picard (pianist) for their dedication, professionalism, and the quality of their musical performances during the recording sessions.

7. REFERENCES

[1] Xubo Liu, Haohe Liu, Qiuqiang Kong, Xinhao Mei, Jinzheng Zhao, Qiushi Huang, Mark D Plumbley, and Wenwu Wang, "Separate what you describe: Language-queried audio source separation," in *INTERSPEEH*, 2022.

[2] Simon Rouard, Francisco Massa, and Alexandre Defossez, "Hybrid transformers for music source separation," 2022.

[3] Alexey Ozerov and Cédric Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 550 – 563, 04 2010.

[4] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux, "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

[5] R. Streicher and W. Dooley, "Basic stereo microphone perspectives-a review," *AES*, 1985.

[6] F. Salmon, F. Changenet, T. Colas, C. Verron, and M. Paquier, "A comparative study of multichannel microphone arrays used in classical music recording," *AES*, 2023.

[7] Daniel Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds. 2000, vol. 13, MIT Press.

[8] Kouhei Sekiguchi, Yoshiaki Bando, Aditya Arie Nugraha, Kazuyoshi Yoshii, and Tatsuya Kawahara, "Fast multichannel nonnegative matrix factorization with directivity-aware jointly-diagonalizable spatial covariance matrices for blind source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2610–2625, 2020.

[9] Ngoc Duong, Emmanuel Vincent, and Remi Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," 2009.

[10] A Hyvärinen and Erkki Oja, "Oja, e.: Independent component analysis: Algorithms and applications. neural networks 13(4-5), 411-430," *Neural networks : the official journal of the International Neural Network Society*, vol. 13, pp. 411–30, 06 2000.

[11] Ryo MUKAI Shoko ARAKI Shoji MAKINO, Hiroshi SAWADA, "Blind source separation of convolutive mixtures of speech in frequency domain," *IEICE TRANSACTIONS on Fundamentals*, vol. E88-A, no. 7, pp. 1640–1655, July 2005.

[12] Taesu Kim, Intae Lee, and Te-Won Lee, "Independent vector analysis: Definition and algorithms," *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 1393–1396, 2006.

[13] Daichi Kitamura, Nobutaka Ono, Hiroshi Sawada, Hirokazu Kameoka, and Hiroshi Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1626–1641, 2016.

[14] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, 2006.

[15] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey, "Sdr - half-baked or well done?," *ICASSP*, 2018.

[16] "Method for the subjective assessment of intermediate quality level of audio systems," *Radiocommunication Sector of ITU*, 2015.

[17] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent, "Multichannel audio source separation with deep neural networks," *Transactions on Audio, Speech, and Language Processing*, 2016.

[18] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *ICASSP*, 2016.

[19] Robin Scheibler, Youna Ji, Soo-Whan Chung, Jaeuk Byun, and Soyeon Choe Min-Seok Choi, "Diffusion-based generative speech source separation," *ICASSP*, 2023.

[20] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J. Weiss, Kevin Wilson, and John R. Hershey, "Unsupervised sound separation using mixture invariant training," *NeurIPS*, 2020.

[21] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang, "Clap : Learning audio concepts from natural language supervision," *ICASSP 2023*, 2022.

[22] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville, "Film: Visual reasoning with a general conditioning layer," *AAAI 2018*, 2017.

[23] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," *ICASSP 2022*, 2022.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ICLR 2020*, 2020.

[25] Qiuqiang Kong, Yin Cao, Haohe Liu, Keunwoo Choi, and Yuxuan Wang, "Decoupling magnitude and phase estimation with deep resunet for music source separation," *arXiv::2109.05418*, 2021.

[26] Sungkyun Chang, Simon Dixon, and Emmanouil Benetos, "YourMT3: a toolkit for training multi-task and multi-track music transcription model for everyone," Dec. 2022, (Poster) Presented at DMRN+17: Digital Music Research Network One-day Workshop 2022.

[27] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *ICLR 2019*, 2019.

[28] Robin Scheibler, "Sdr — medium rare with fast computations," 2021.

[29] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Defossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi, "Audiogen: Textually guided audio generation," *ICLR 2023*, 2023.

[30] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Ki, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *ICASSP 2020*, 2019.