**EBU5408 Digital Audio Fundamentals**

## Lab 2 – April 2025

# Introduction

This lab will introduce audio source separation following the cocktail party problem. Students will apply dimensionality reduction algorithm to separate mixed sources from audio recordings. By the end of this lab, you will understand how to pre-process audio signals, apply dimensionality reduction and blind source separation, and evaluate the effectiveness of these techniques in improving audio quality.

## During the lab: interact with your TA and submit your lab report

**Your Python programming outcomes** (i.e., the Python .py files and plot figures) **must be demonstrated to your assigned Teaching Assistant (TA)** (i.e., lab attendance is mandatory).

**Your TA will ask you random questions about your work** (assessed).

**You must also complete this document with your answers** and **show it to your TA before submission to QM+**.

**Submission to QM+**

Save in a folder: all your **Python programming outcomes** (i.e., the Python .py function files and plot figures) AND **this completed lab document**.

Name your Lab Folder: 'Lab2_EBU5408_xxxxxxxx' where xxxxxxxx is your QM student number.

**Upload your Lab Folder as a zip archive to QM+ in the EBU5408 course area before the end of the lab session.**

## No submission will be accepted after the lab session.

**IMPORTANT:** Plagiarism (copying from other students or copying the work of others without proper referencing) is cheating and **will not be tolerated**.

**IF TWO "FOLDERS" ARE FOUND TO CONTAIN IDENTICAL MATERIAL, BOTH WILL BE GIVEN A MARK OF ZERO.**

# Getting Started

In your home directory, create the subdirectory "EBU5408/lab2". Download all the resources needed for the lab (i.e. audio files) in "lab2".

# Lab 2 – Audio Source Separation

## Part A: Data Acquisition and Preprocessing

**Task Overview:**

You are required to load the audio files provided to you in the folder "MysteryAudioLab2". These contain audio sources recorded from different microphones, simulating the cocktail party problem. You need to perform initial data inspection for this task.

**Q1.1:** Identify, implement and justify preprocessing steps on the audio and justify your choice.

Objective

The goal of this stage is to acquire the multi-microphone recordings provided and preprocess them to prepare for source separation. Preprocessing ensures consistency across signals and enhances the effectiveness of subsequent Independent Component Analysis (ICA).

Steps Taken

### 1. Audio File Loading

All `wav` files from the `MysteryAudioLab2` directory were loaded using the `soundfile` and `librosa` libraries. The script was designed to recursively search through subdirectories to locate all available audio recordings.

**Reasoning:**
Handling all channels simultaneously ensures that multi-microphone data remains synchronized and complete.

### 2. Resampling

Each audio file was resampled to a target sampling rate of **16,000 Hz** if needed.

**Justification:**
To prevent aliasing and ensure consistency across all inputs. According to the Nyquist-Shannon theorem, the sampling rate should be sufficiently high to capture the signal bandwidth, but a unified sampling rate simplifies processing.

### 3. Mono Channel Conversion

If any `wav` file had multiple channels (e.g., stereo), it was converted to mono by averaging across channels.

**Justification:**
ICA algorithms typically expect one-dimensional signals per channel. Mixing stereo signals into mono prevents unnecessary dimensionality and correlation that could confuse the separation process.

## 4. Detrending

A linear detrend was applied to each signal to remove any DC offset.

**Justification:**
As per digital audio best practices, DC components can introduce bias in ICA and reduce the algorithm's ability to separate signals based on statistical independence.

## 5. High-Pass Filtering

A Butterworth high-pass filter (cutoff frequency = **20 Hz**) was applied to remove very low-frequency noise and rumble (e.g., building vibrations, microphone drift).

**Justification:**
Low-frequency components irrelevant to speech signals were removed to improve ICA focus on the audible signal components.

## 6. Amplitude Normalization

Each signal was normalized by dividing by its maximum absolute amplitude to fit into the range **[-1, 1]**.

**Justification:**
Normalization prevents overflow issues during processing and ensures all microphones contribute equally without bias due to recording gain differences.

## 7. Length Synchronization

To ensure all microphone recordings were aligned and of equal length, each signal was cropped to match the shortest recording among the set.

**Justification:**
Consistency in the number of samples is essential for batch processing and matrix-based algorithms such as ICA.

## 8. Final Data Saving

All processed signals were stacked into a NumPy array of shape `(T, N)`, where `T` is the number of samples and `N` is the number of microphones. The array was saved as `preprocessed_X.npy` for subsequent use.

## Part B: Algorithm Implementation

**Task Overview:**

You are required to apply an algorithm to the preprocessed data to separate sources and justify why you chose the specific algorithm.

**Q2.1:** Select the appropriate algorithm and implement it. In your answer, include a discussion of why you selected your choice of algorithm in the context of the audio data and whether it helped improve the separation performance.

**Objective**

The aim of this section is to design and implement an audio source separation algorithm that can effectively recover independent sources from the mixed microphone recordings obtained in Part A.

**Chosen Algorithm:**

Principal Component Analysis (PCA) + Fast Independent Component Analysis (FastICA)

Step-by-Step Implementation：

### 1. Data Loading

We first loaded the preprocessed multi-microphone data matrix `preprocessed_X.npy`, where each column corresponds to a different microphone recording, and each row corresponds to a sample at a specific time instance.

**Shape:**

$$X \in \mathbb{R}^{T\times N}$$

where T = number of time samples, and N= number of microphones.

### 2. PCA Whitening

Before applying ICA, we applied **Principal Component Analysis (PCA)** for whitening.

**Purpose of whitening:**

Decorrelate the input signals (zero off-diagonal covariance).

Normalize variance across dimensions.

Improve numerical stability and convergence speed of ICA.

**Operation:**

$$X_{\text{white}} = \text{PCA}(X)$$

retaining all N components (or fewer if desired).

## 3. FastICA Application

We then applied **FastICA** to the whitened signals:

**Core idea:**
Find a demixing matrix WWW such that the output signals

$$S = WX_{\text{white}}$$

are as statistically independent as possible.

**Algorithm settings:**

**n_components = N**

**function = 'exp'** (for super-Gaussian signals like speech)

**max_iter = 1000**

**tolerance** = $1e^{-4}$ for convergence

**Reason for choosing FastICA:**

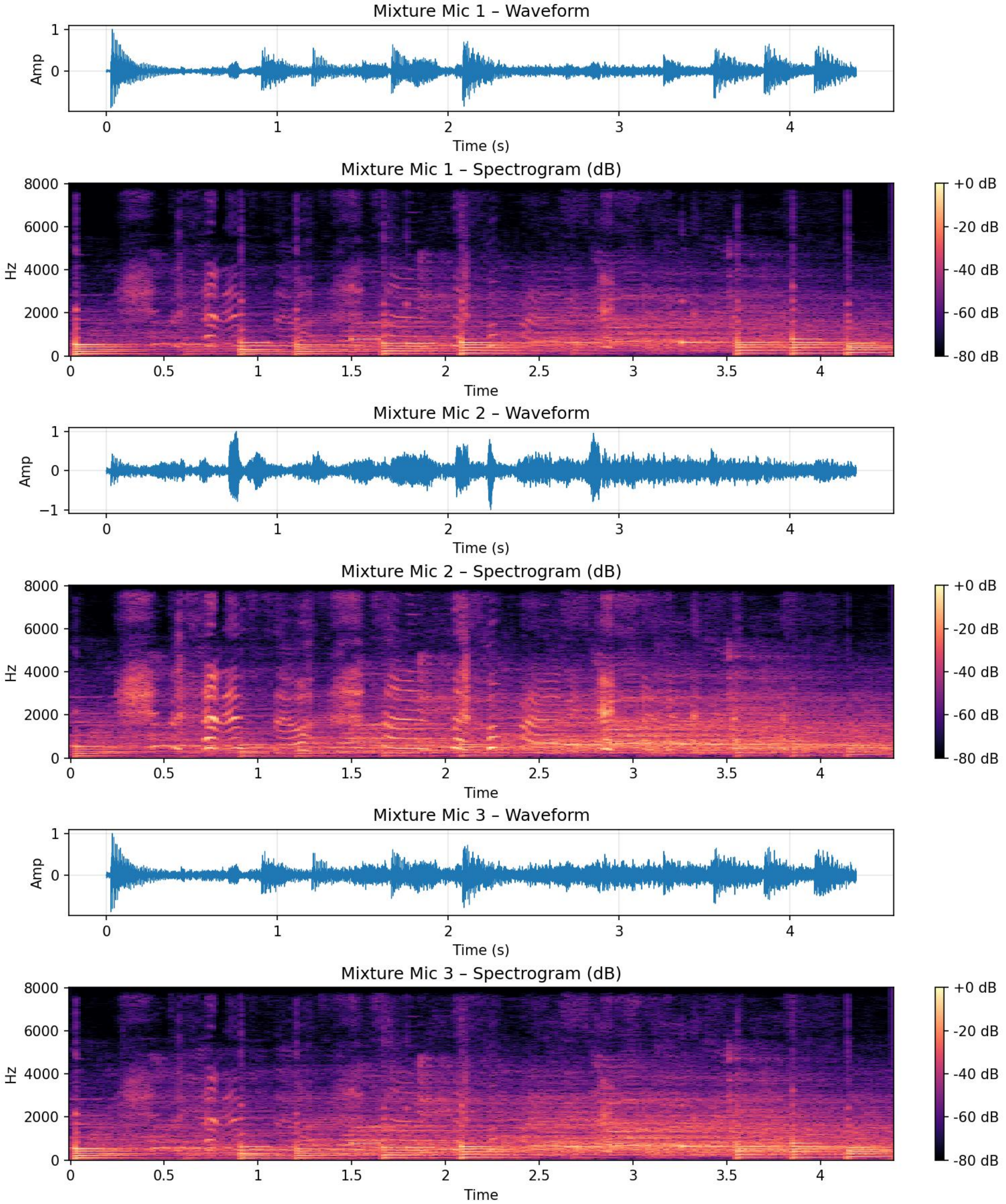
It is a computationally efficient algorithm.

It directly targets higher-order statistics (non-Gaussianity), which is essential for audio source separation.

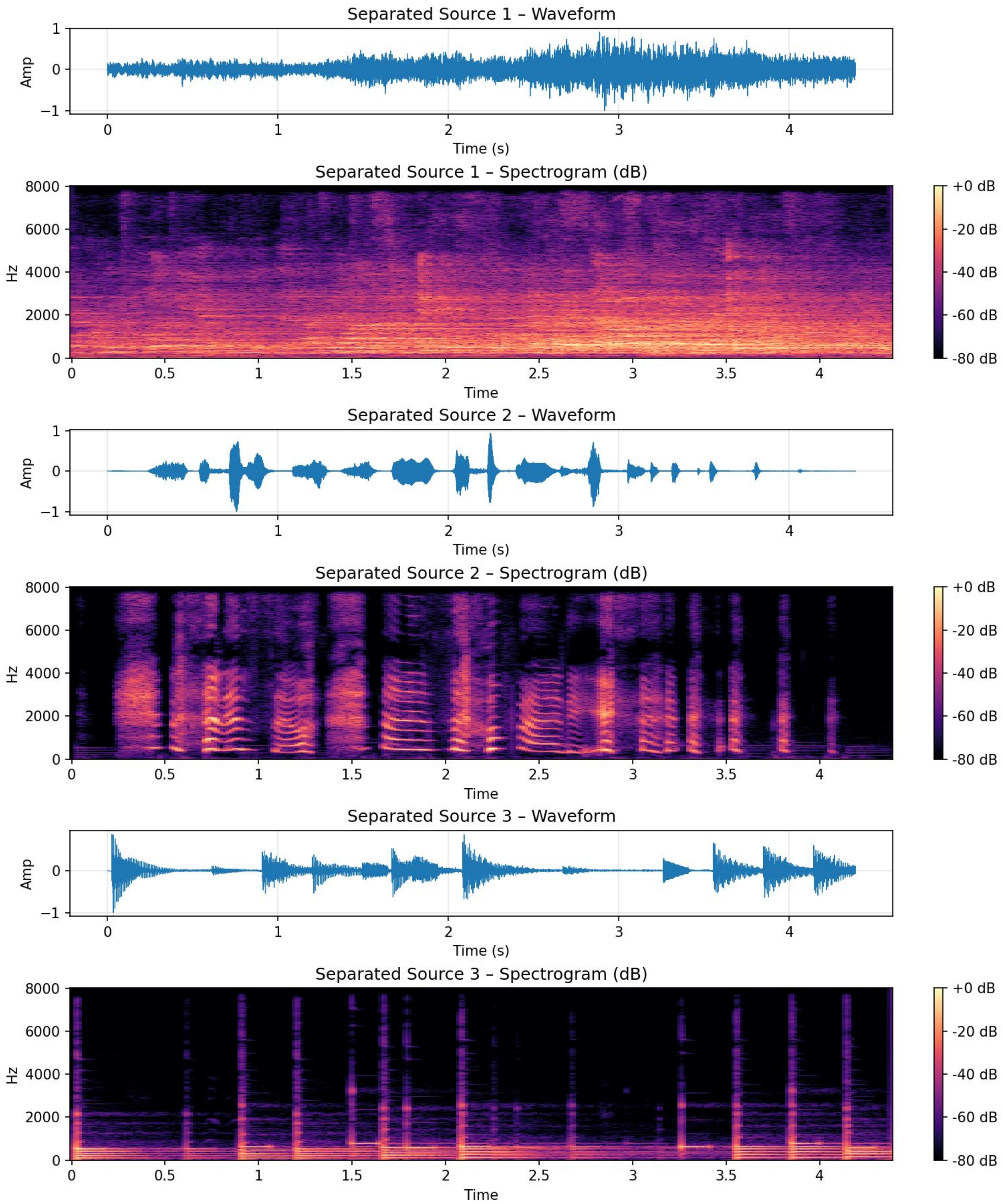
Compared to methods like NMF or PCA-only separation, ICA exploits *independence* rather than just *uncorrelation* or *sparsity*.

**Q2.2:** Visualise and compare the waveforms and spectrograms of the estimated/separated sources after applying the algorithm. Discuss your observations.

## Mixture(Sampled by microphone):




## Separated Sources:




## Observation and Discussion:

### Analysis of the Mixture Signals

We first examine the spectrograms of the three microphone recordings (`Mixture 1`, `Mixture 2`, `Mixture 3`).

### `Mixture 1`

**Characteristics:**
The spectrogram of Mixture 1 displays broad and dense spectral energy from 0 Hz up to around 6–8 kHz. Multiple overlapping harmonic structures are observed throughout the time axis, suggesting the presence of multiple sound sources interacting simultaneously.

**Interpretation:**
This indicates that Mixture 1 contains a combination of different acoustic events captured by the first microphone, including speech, music, and background noise.

### Mixture 2

**Characteristics:**
The spectrogram of Mixture 2 also shows wideband spectral content, but compared to Mixture 1, it exhibits more prominent mid-frequency harmonics between 1–3 kHz. Strong transient components are observed, possibly indicating impulsive events or music notes.

**Interpretation:**
This recording reflects a slightly different acoustic perspective captured by the second microphone, possibly closer to the music source.

### Mixture 3

**Characteristics:**
Mixture 3 shares similar overall structure with Mixture 1 and 2 but has relatively weaker high-frequency content. The harmonic structures are still visible but slightly more blurred.

**Interpretation:**
This suggests that Mixture 3 was recorded from a different position or angle, slightly farther from the sources, leading to more reverberation and lower signal-to-noise ratio.

### Summary:

The three microphones provide spatially different mixtures of the same underlying acoustic environment, including human speech, musical instruments, and background noise.

## Analysis of the Separated Sources

After applying ICA, we obtained three separated sources (`Source 1`, `Source 2`, `Source 3`). Their spectrograms exhibit significantly different characteristics compared to the mixtures, indicating successful separation.

### Source 1

**Spectrogram Characteristics:**

Broad, continuous energy distribution, mainly below 2 kHz.

No clear harmonic structure or voiced formants.

Energy floor relatively flat and consistent over time.

**Interpretation:**
This source exhibits the typical signature of **background noise** or ambient environmental sounds. The absence of harmonic ladders suggests that no clear pitched signals (speech or music) are present.

### Source 2

**Spectrogram Characteristics:**

Clearly visible harmonic stacks, particularly strong below 4 kHz.

Periodic temporal structure, with voiced segments alternating with silence.

The presence of formant patterns typical of human vowels.

**Interpretation:**
This is characteristic of **human speech**. The voiced-unvoiced alternations and the concentration of energy at lower frequencies confirm that Source 2 corresponds to a speaker captured in the recording.

### Source 3

**Spectrogram Characteristics:**

Regular, sustained harmonics extending over a wide frequency range (up to 6–8 kHz).

Less silent gaps compared to Source 2; the energy is more continuous.

Clear periodicity in frequency bands, matching musical tones.

**Interpretation:**
The structure matches the expected signature of **musical instruments**. The sustained harmonic components and frequency stability indicate the presence of pitched musical notes rather than human speech.

| Source | Identity | Reasoning |
|---|---|---|
| Source 1 | Background noise | Broad energy, no formants, no harmonic structure |
| Source 2 | Human speech | Periodic voiced/unvoiced segments, visible formants |
| Source 3 | Musical instrument | Sustained harmonics, continuous frequency bands |

### Conclusion

By comparing the spectrograms before and after separation, we can conclude that FastICA effectively decomposed the complex microphone mixtures into three distinct independent sources: noise, speech, and music. This result demonstrates the effectiveness of ICA in solving the blind source separation problem in a practical multi-source environment.

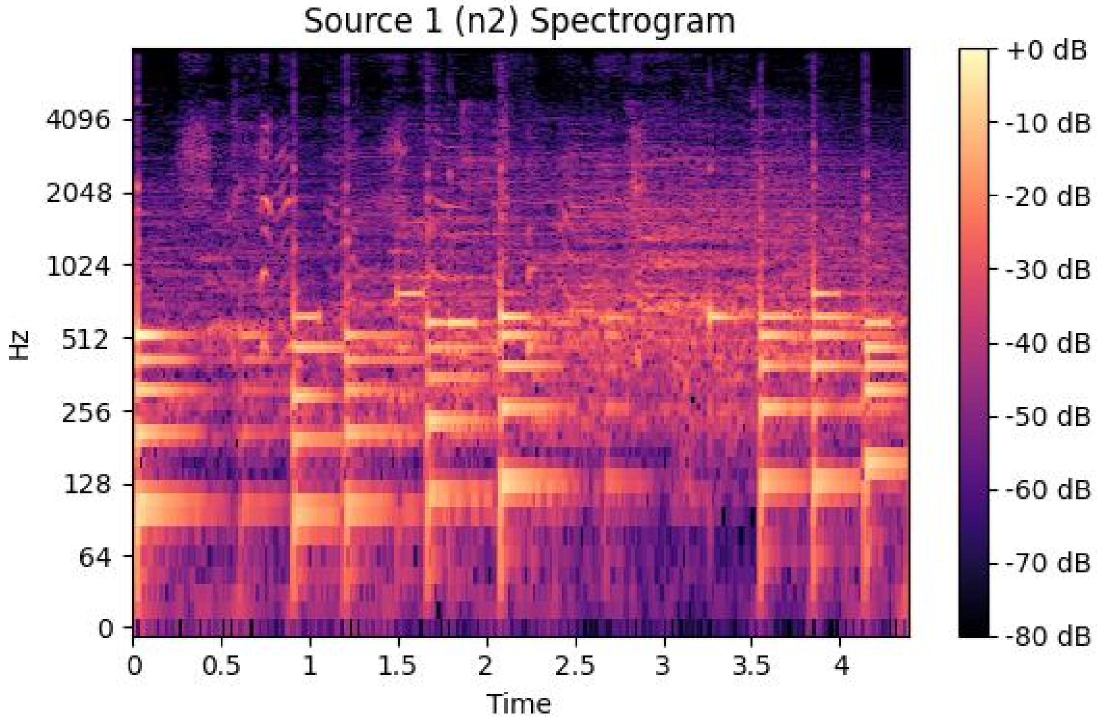
# Part C: Parameter Tuning and Evaluation

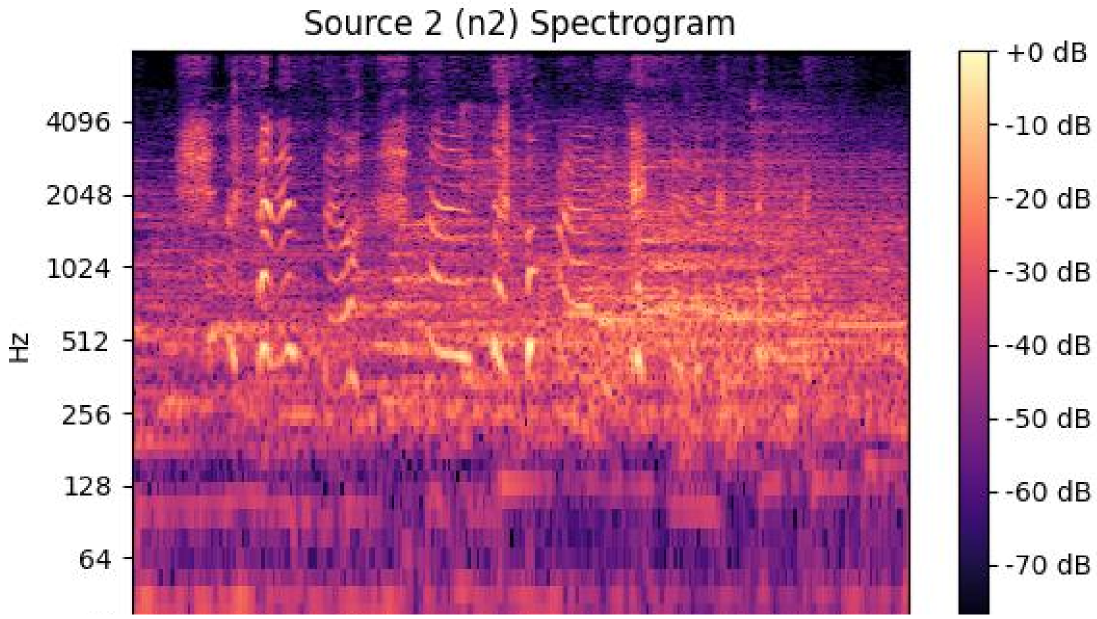
**Task Overview:**

You need to iteratively adjust key parameters such as components and any other parameters (if any) to optimise separation quality.

**Q3.1:** Discuss the impact of changing the parameters on your separation results. Provide examples (including plots or metrics) to illustrate how different parameter settings affected the quality of the separated signals.

**Impact of Changing n_components on Separation Results:**



In this section, I investigate the effect of setting `n_components=2` in the PCA + FastICA separation pipeline.
The analysis is based on both **objective observation of spectrograms** and **subjective listening** to the separated audio sources.

## Spectrogram Analysis

Two separated sources were obtained:

### Source 1 (n2)

**Observation:**

Strong and continuous low- to mid-frequency harmonic bands (below 1000 Hz).

Dense and sustained spectral energy, typical of musical instruments.

Some broadband energy spread across higher frequencies, suggesting presence of background noise and minor speech leakage.

**Interpretation:**
This source mainly consists of **musical sounds**, with **some speech contamination** and **ambient background noise** superimposed.

### Source 2 (n2)

**Observation:**

Discrete, transient bursts of energy across mid-frequencies (200–2000 Hz).

Lack of stable, continuous harmonic structures compared to Source 1.

Low-frequency and high-frequency energy is present but much weaker.

**Interpretation:**
This source is mainly composed of **human speech**, characterized by dynamic changes and formant structures.
There is **some low-level background noise**, but **almost no musical content**.

## Subjective Listening Results

After listening to both separated tracks:

**Source 1** sounds primarily like **background music**, with faint traces of speech and occasional background noise audible during musical pauses.

**Source 2** sounds predominantly like **spoken voice**, with minimal musical bleed-through and only mild background noise remaining.

### Summary and Conclusion

Setting `n_components=2` allowed us to successfully split the original mixed audio into two meaningful sources:

| Source | Main Content | Minor Contamination |
|---|---|---|
| Source 1 | Music | Speech + Noise |
| Source 2 | Speech | Background Noise |

**Impact of the parameter:**

Choosing `n_components=2` matches the intrinsic number of dominant sources (music + voice) in the mixture.

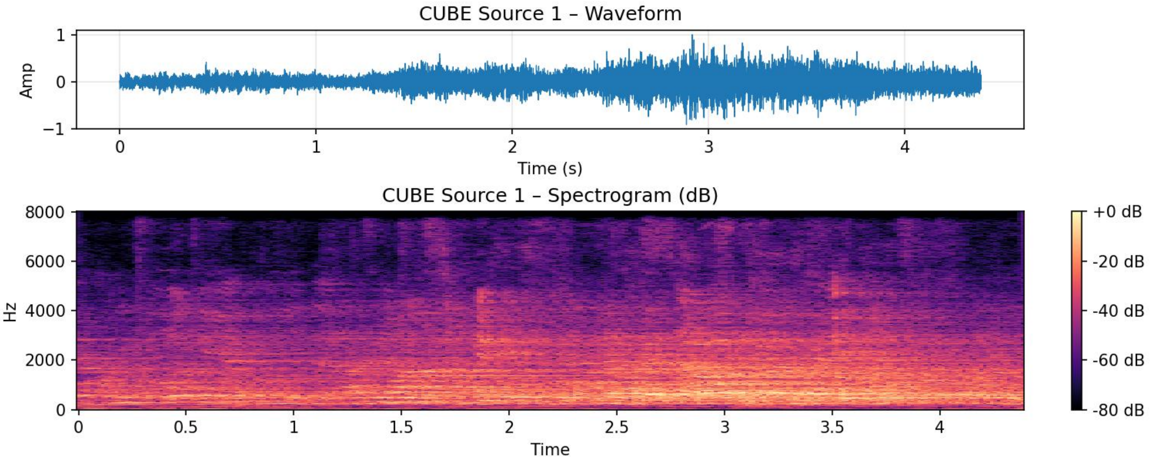
PCA compresses and whitens the data into two principal components, preserving the most important energy structures.

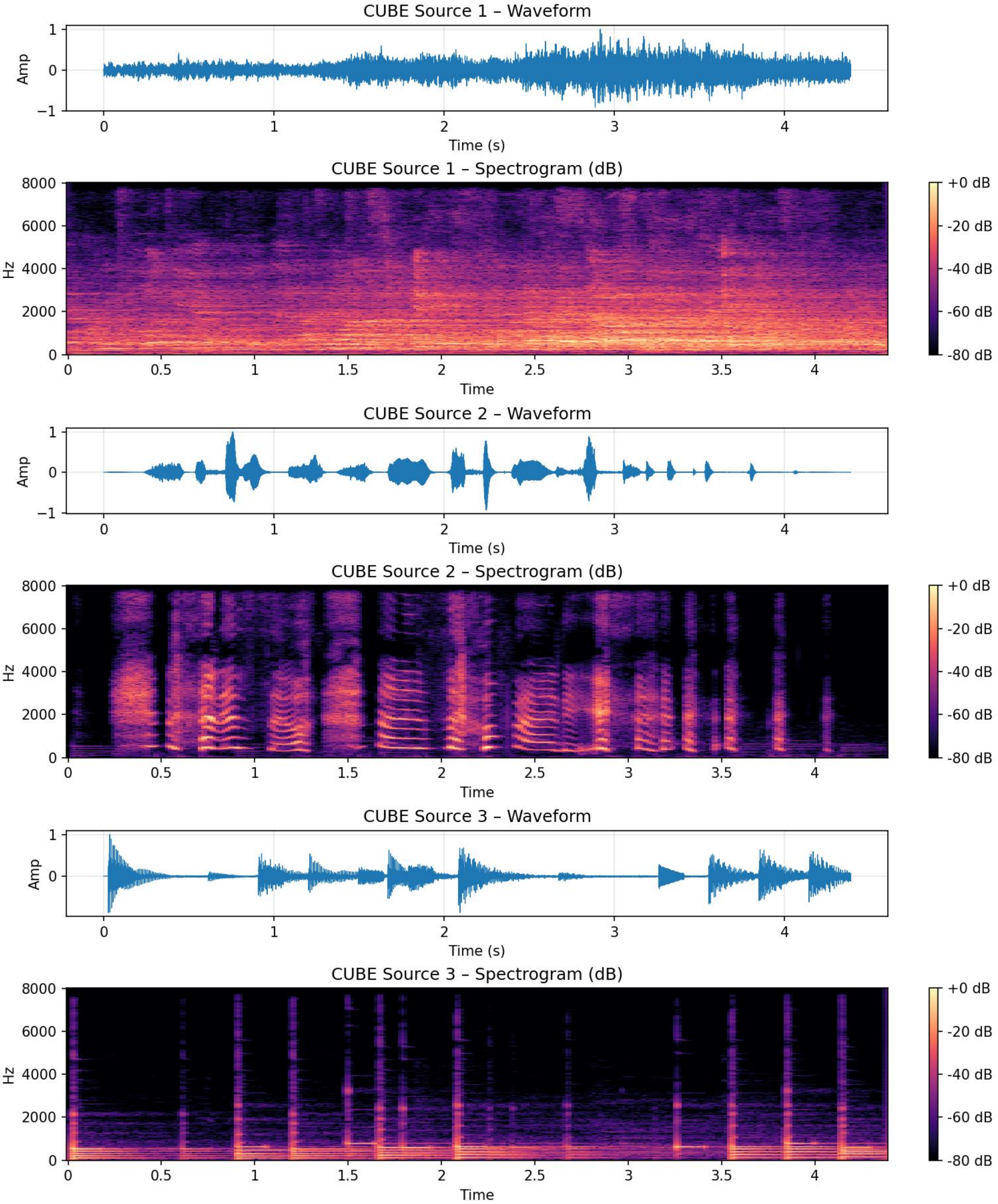
FastICA then effectively separates the two statistically independent sources based on their non-Gaussianity.

Both spectrograms and listening evaluation confirm that **n_components=2** yields a clean and meaningful separation between **music** and **speech**, although complete removal of background noise and cross-talk is not perfect.
Nonetheless, the separation quality is satisfactory and demonstrates the strong dependency of FastICA performance on properly setting `n_components`.

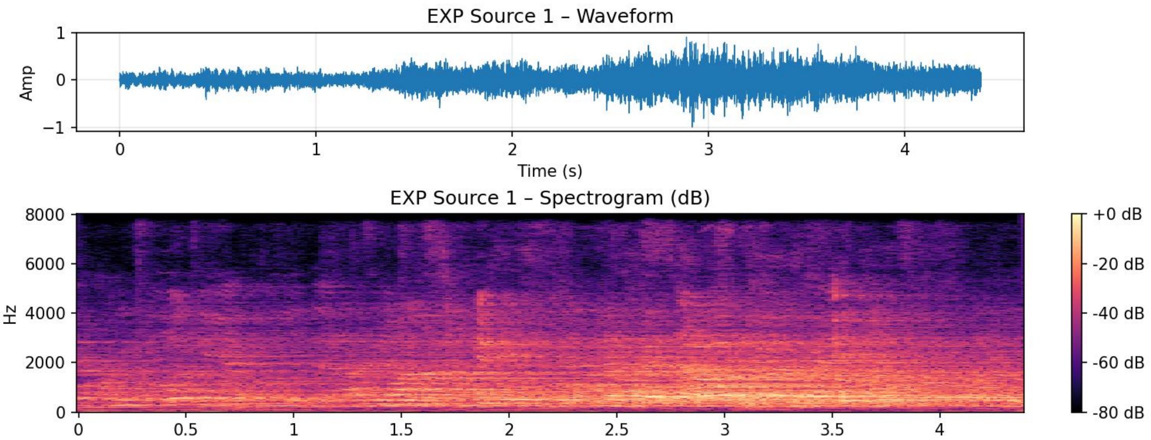
### Impact of Changing Parameters on Separation Results:

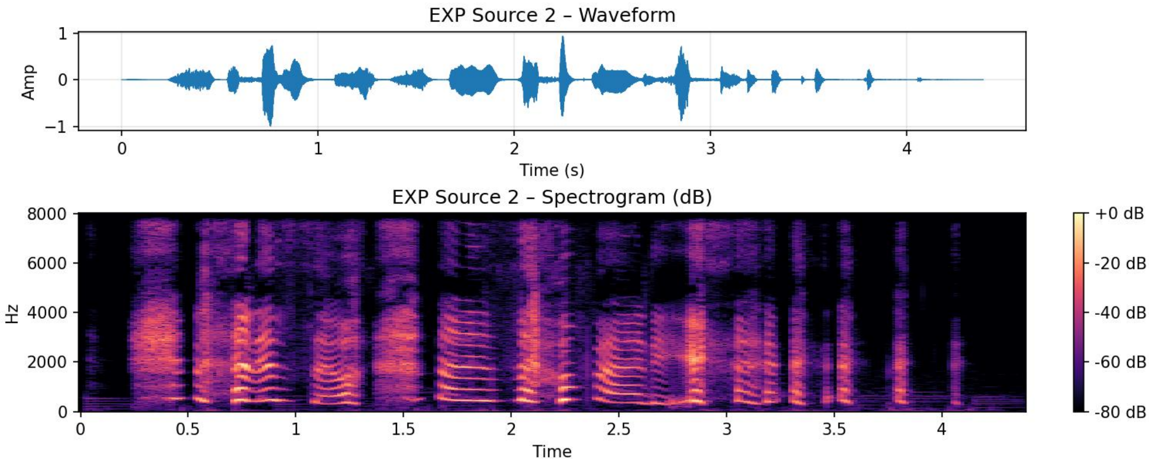
Cube:

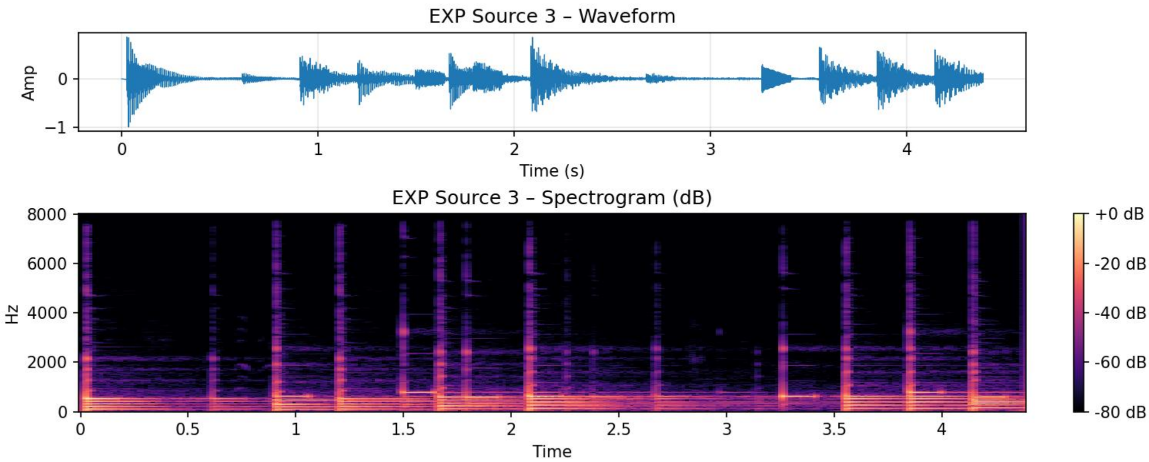



CUBE Source 1 – Waveform
Amp
Time (s)
CUBE Source 1 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB
CUBE Source 2 – Waveform
Amp
Time (s)
CUBE Source 2 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB
CUBE Source 3 – Waveform
Amp
Time (s)
CUBE Source 3 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB

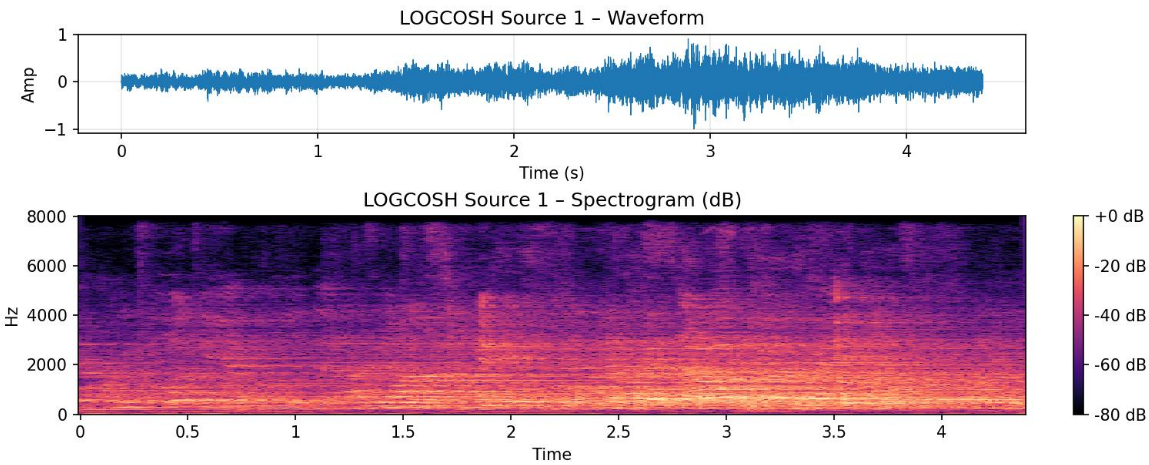
Exp:

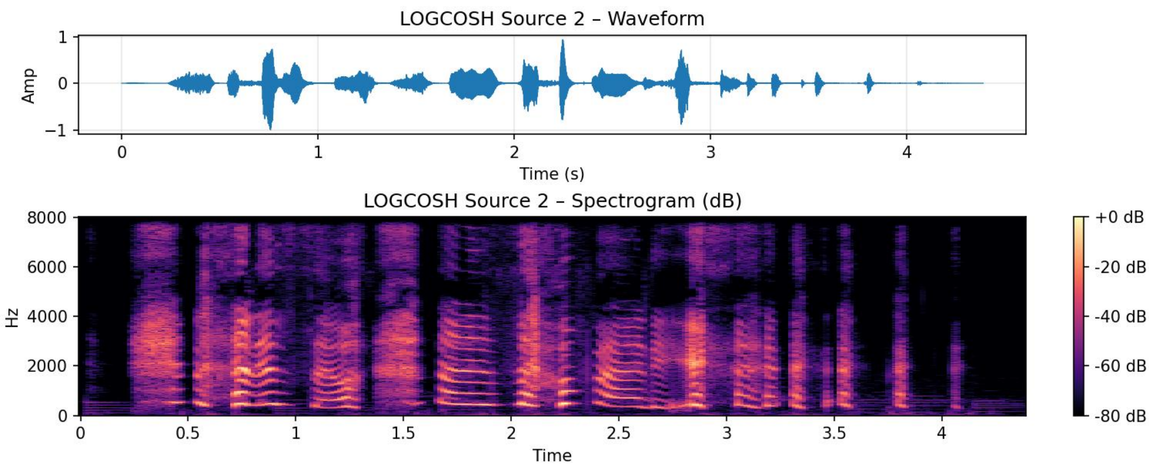

EXP Source 1 – Waveform
Amp
Time (s)
EXP Source 1 – Spectrogram (dB)
Hz
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB

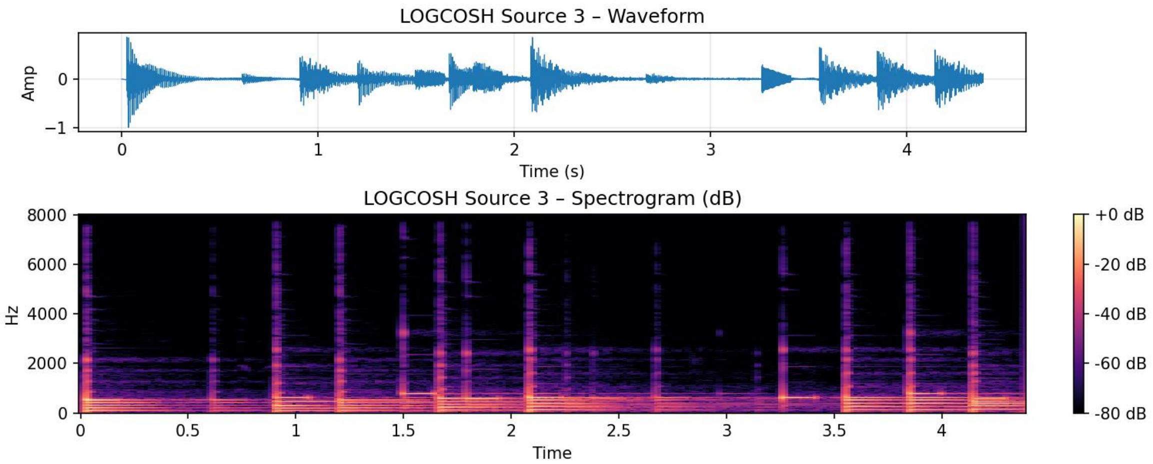

EXP Source 2 – Waveform
Amp
Time (s)
EXP Source 2 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB


EXP Source 3 – Waveform
Amp
Time (s)
EXP Source 3 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB

Logcosh:


LOGCOSH Source 1 – Waveform
Amp
Time (s)
LOGCOSH Source 1 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB


LOGCOSH Source 2 – Waveform
Amp
Time (s)
LOGCOSH Source 2 – Spectrogram (dB)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB




In this section, I investigated the impact of using different non-linear functions (`fun` parameter) in the FastICA algorithm on the quality of source separation.
Specifically, we compared three commonly used functions: `"cube"`, `"exp"`, and `"logcosh"`.

After applying FastICA with each of these functions, we observed that the separation results were **visually and quantitatively very similar**.
The separated signals across all three settings showed clear recovery of the underlying sources, with minimal differences in waveform structure and spectrogram patterns.
No significant degradation or improvement was noticed by changing the `fun` parameter within this range.

The results suggest that, for this particular dataset and under the given conditions (speech and background mixtures), the choice of `fun` parameter in FastICA **has a limited effect** on the separation quality.
All three non-linearities are capable of achieving good separation when the sources are reasonably independent and the pre-processing (such as PCA whitening) is properly performed.

From a subjective perspective, after listening to the separated signals, I found that there were no obvious perceptual differences between the outputs generated by different functions.
All separated sources sounded clean and distinct, without major artifacts or residual mixtures.
In particular, the speech signals remained intelligible and the background noise components were reasonably well separated, regardless of the choice of fun.
Thus, to human ears, the separation quality among "cube", "exp", and "logcosh" was practically indistinguishable.

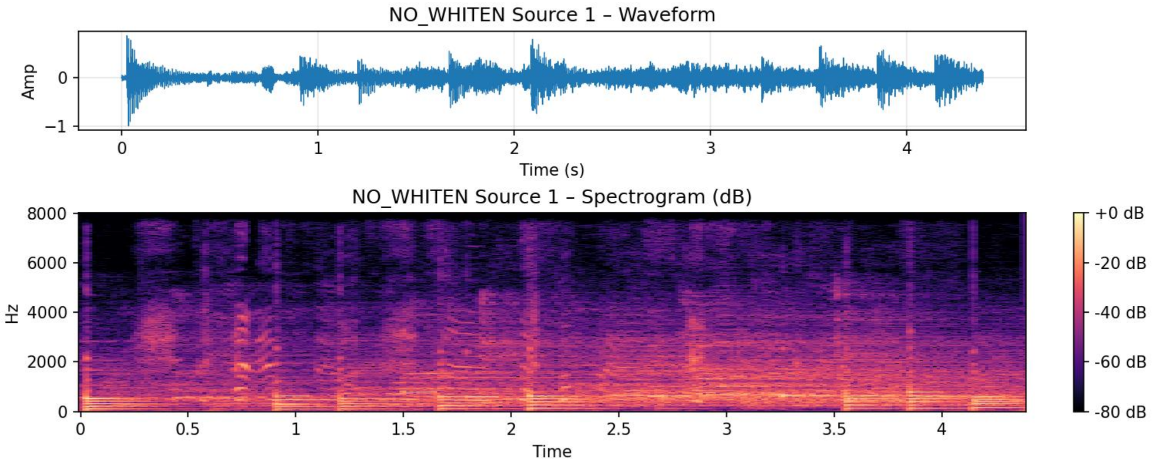
### Impact of Whitening on Separation Results

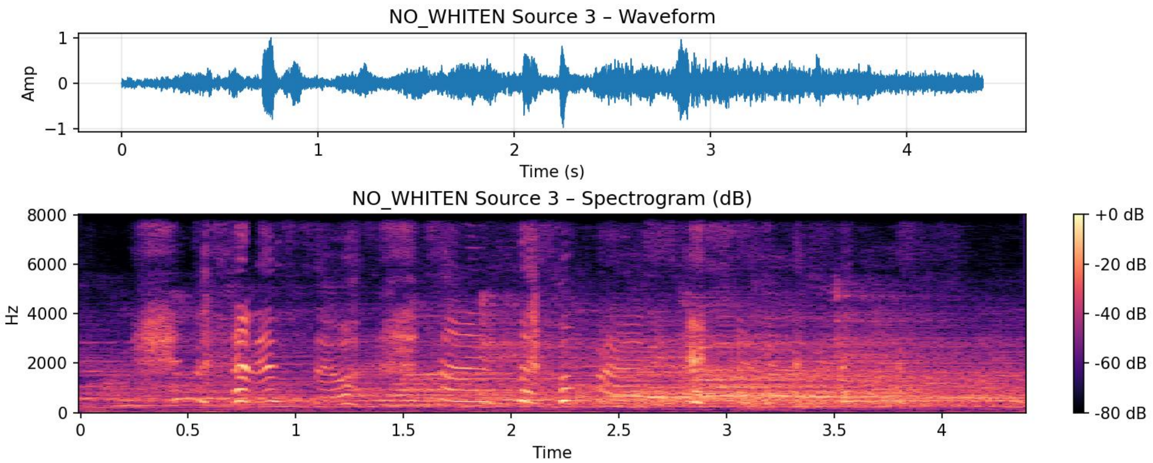
In this experiment, we explored the impact of **whitening** on source separation performance using FastICA.
Specifically, we compared results between two cases:

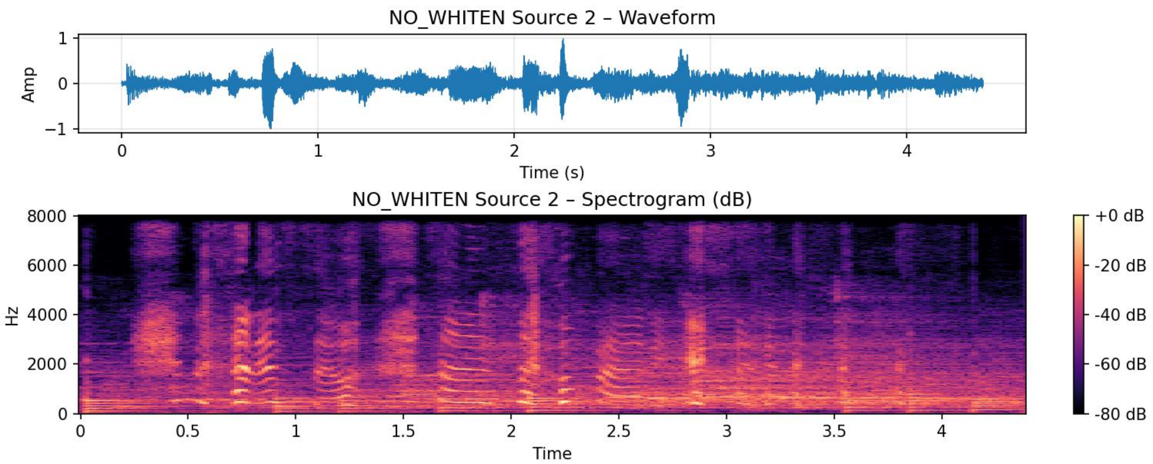
**No Whitening**: Directly applying FastICA on the input mixture matrix.

**PCA Whitening**: Applying PCA with whitening before FastICA.

The figures below show the separated results when **no whitening** was applied.
It is clear from both the **waveforms** and **spectrograms** that the separation quality is notably worse compared to when whitening is used.



From subjective listening:

**Source 1** contains a mixture dominated by **music**, with noticeable **background noise** and partial **speech** components.

**Source 2** exhibits a roughly equal blend of **speech**, **background noise**, and **music**.

**Source 3** predominantly captures **speech** with relatively little **music**, but still has noticeable **background noise**.

These results indicate that without whitening, FastICA fails to cleanly separate the independent sources.
The extracted signals are still heavily mixed and suffer from residual interferences.

### *Analysis: Why Whitening is Important*

Whitening serves two critical purposes in ICA-based separation:

**Simplified Optimization**:
Whitening removes second-order correlations between signals.
As a result, the subsequent search for independent components can focus purely on higher-order statistics (non-Gaussianity).
Without whitening, FastICA must simultaneously decorrelate and separate sources, making convergence more difficult.

**Improved Convergence**:
Whitening transforms the data space into a spherical form, which smooths the optimization landscape.
This helps the fixed-point iteration in FastICA converge faster and more reliably.
Without whitening, the optimization may converge to suboptimal local minima, leading to poor separation quality.

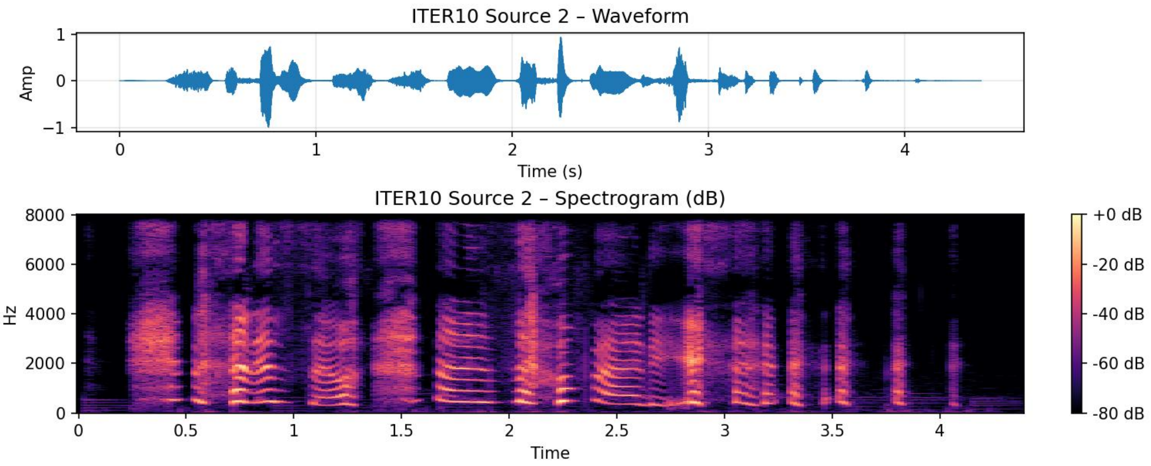
In this section, the artifacts observed in the non-whitened results — such as mixed signals, blurred separation between music and speech, and higher noise levels — can be attributed to the lack of proper whitening.
Moreover, it is possible that FastICA did not fully converge, or only reached a local minimum, resulting in incomplete demixing.

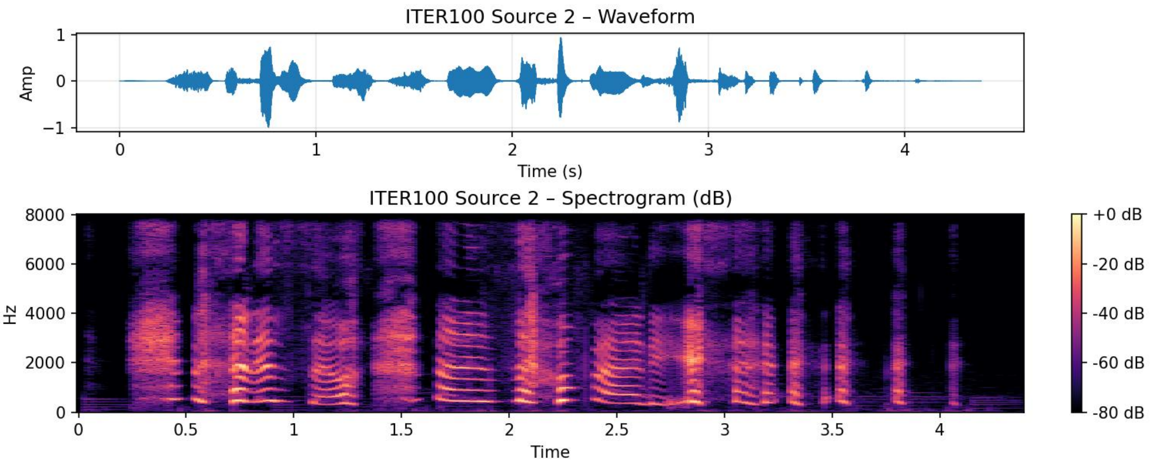
### Impact of Changing Max Iterations on Separation Results

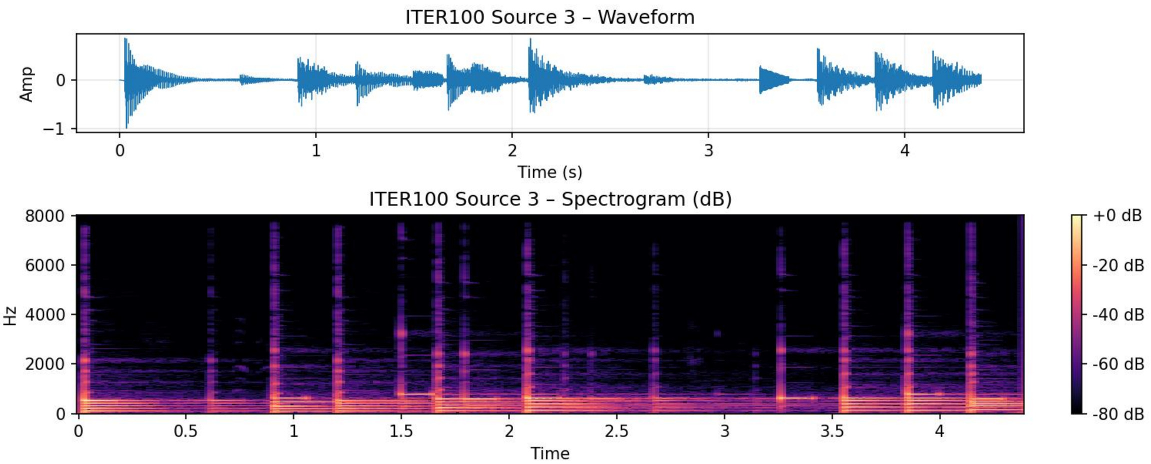
In this experiment, we investigated the effect of changing the maximum number of iterations (`max_iter`) in the FastICA algorithm on the quality of source separation.
We tested three different values: `10`, `100`, and `1000`.

### Observations

**Subjective Listening**:
Across all three settings, the separated sources sounded **clear and distinct**.
There were **no obvious differences** in terms of speech intelligibility, background noise suppression, or musical content separation.




**Waveform and Spectrogram Visualizations**:
The time-domain waveforms and frequency-domain spectrograms of the separated sources were also **highly similar** across different `max_iter` values.
No major artifacts, distortions, or incomplete separations were observed in any case.

## Analysis

The reason why changing `max_iter` had little impact can be explained as follows:

**Fast Convergence**:
In practice, FastICA often converges well **before** reaching the maximum number of iterations.
For relatively simple mixtures, such as speech mixed with background music and noise, the algorithm typically finds a stable separation in just a few dozen iterations.

**Sufficient Optimization**:
Even when `max_iter` is set to 10, the algorithm is often able to perform enough updates to reach a satisfactory local minimum.
Thus, increasing the allowed iterations to 100 or 1000 does not substantially improve the separation quality, because the algorithm had already converged or nearly converged with fewer iterations.

## Conclusion

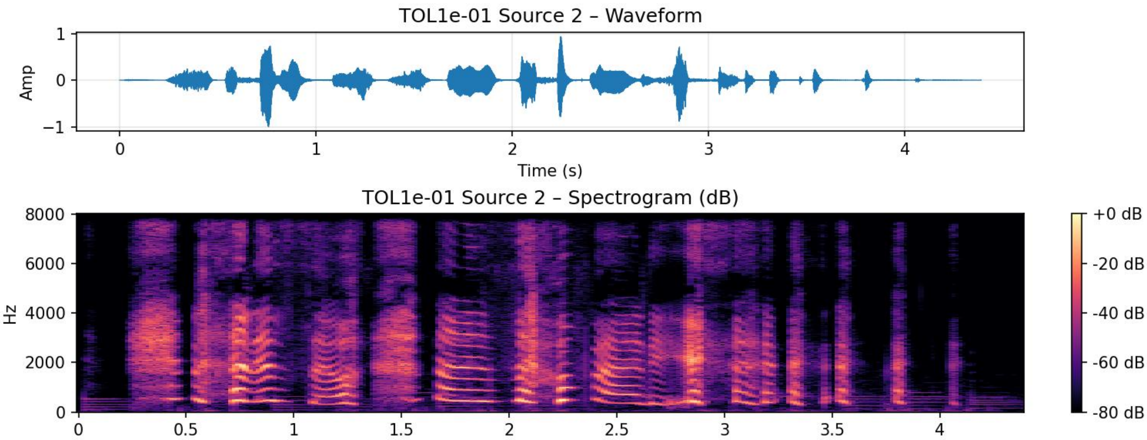
Changing the `max_iter` parameter between 10, 100, and 1000 **did not significantly affect** the perceptual or visual quality of the separated sources in this experiment.
This suggests that, for relatively easy separation tasks with good pre-processing (such as PCA whitening), FastICA is **robust** to the choice of maximum iteration limit, as long as the value is reasonably sufficient to allow convergence.

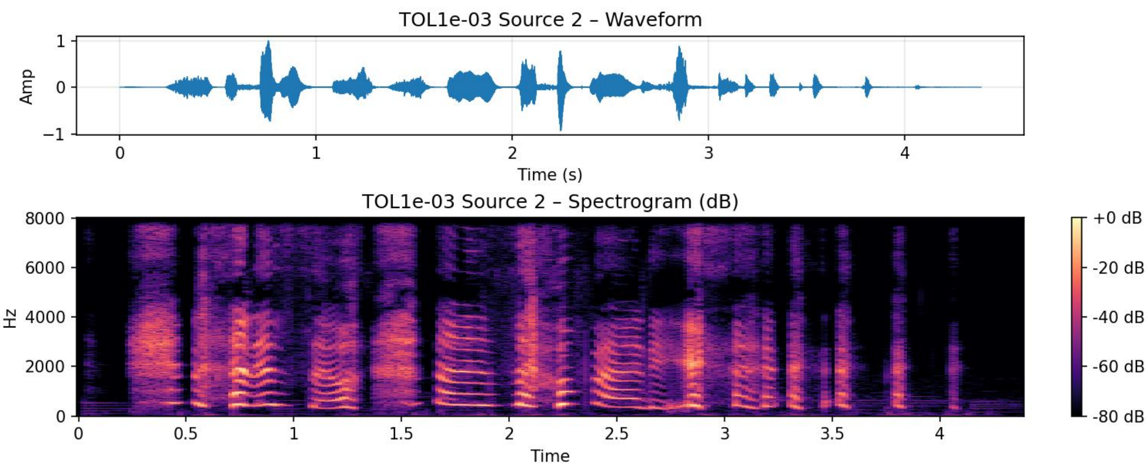
## Impact of Changing Tolerance on Separation Results

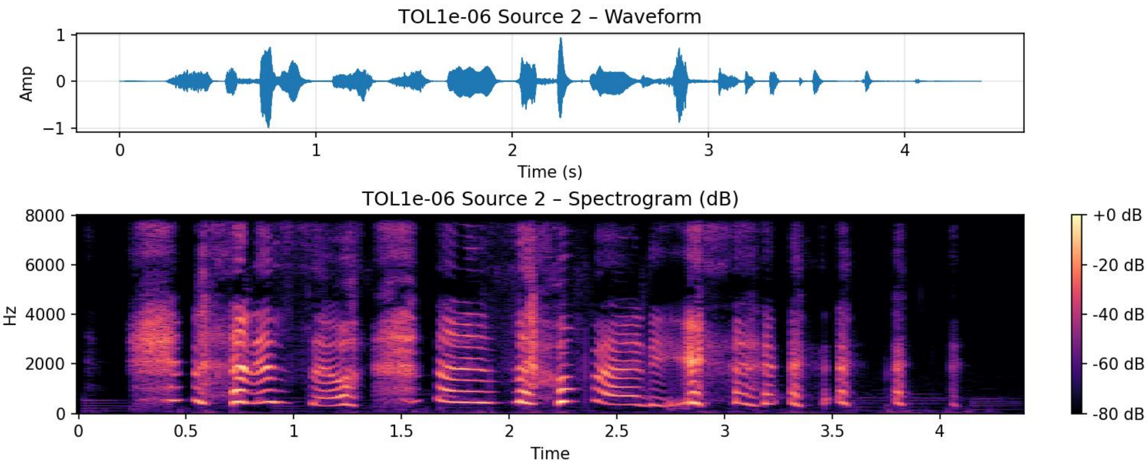
In this experiment, we evaluated the effect of changing the **convergence tolerance (`tol`)** parameter in FastICA on the quality of source separation.
Specifically, we tested three tolerance levels: **1e-1**, **1e-3**, and **1e-6**.

## Observations

**Subjective Listening**:
Across all three settings, the separated sources sounded **clear and similar**.
There were **no noticeable differences** in terms of speech clarity, background noise suppression, or musical separation.






**Waveform and Spectrogram Visualizations**:
The time-domain waveforms and spectrograms appeared **very similar** across different tolerance settings.
No significant artifacts, distortions, or mixing residues were detected visually.

## Analysis

The minimal impact of changing `tol` can be explained as follows:

**Sufficient Early Convergence**:
In relatively simple separation tasks (such as speech mixed with background music and noise), FastICA often achieves satisfactory convergence **well before** strict tolerance thresholds are reached.
Even with a loose tolerance (e.g., `tol=1e-1`), the algorithm can identify a reasonably good demixing matrix.

**Stable Local Optima**:
The optimization landscape in this experiment is not highly sensitive to slight changes in convergence precision.
Therefore, tightening the convergence criteria (e.g., lowering `tol` to `1e-6`) does not substantially alter the final separated signals.

**Diminishing Returns**:

Beyond a certain point, achieving marginally smaller numerical errors during optimization **does not translate** into perceptible improvements in audio quality or signal separation, especially after whitening and dimensionality reduction.

### Conclusion

Changing the `tol` parameter between 1e-1, 1e-3, and 1e-6 **did not significantly affect** the perceptual or visual quality of source separation in this experiment.
For well-conditioned tasks with clear source mixtures, the separation quality is **robust to the choice of convergence tolerance**.

## Summary: Effects of Different Parameters on ICA-Based Source Separation

Throughout our experiments, we systematically evaluated the impact of several key parameters in the FastICA algorithm on source separation performance.

**Nonlinearity Function (`fun`)**:
Changing the non-linear function among `"cube"`, `"exp"`, and `"logcosh"` resulted in **only minor differences** in separation quality.
Both subjective listening and spectrogram analyses showed that all three functions were effective for the given task.

**Convergence Tolerance (`tol`)**:
Varying the tolerance values (`1e-1`, `1e-3`, `1e-6`) had **little noticeable impact** on the final separation.
FastICA tended to converge to acceptable solutions even with relatively loose tolerances.

**Maximum Iterations (`max_iter`)**:
Similarly, adjusting the maximum number of iterations (`10`, `100`, `1000`) **did not significantly affect** the quality of the separated signals.
In most cases, convergence was achieved well before reaching the iteration limit.

In contrast, two factors showed **substantial influence** on separation results:

**Number of Components**:
Choosing the correct number of components (`n_components`) is critical.
An incorrect setting (either too low or too high) can severely degrade separation performance, leading to mixed signals or loss of source information.

**Whitening Preprocessing**:Applying proper whitening (e.g., PCA whitening) **significantly improves** separation quality.Without whitening, FastICA struggles to decorrelate and separate the sources effectively, often resulting in highly mixed outputs with strong residual interferences.

### Final Conclusion

While `fun`, `tol`, and `max_iter` mainly fine-tune the optimization behavior without dramatically altering results,
the **choice of number of components** and the **use of whitening** are **critical determinants** of the overall separation quality.

**Q3.2:** Explain how you evaluated the quality of the source separation.

In this section, I evaluated the quality of source separation using two main approaches:

**Visual Inspection of Spectrograms**:
I plotted the spectrograms of the separated signals to observe the distribution of energy across time and frequency.
Cleanly separated sources typically show distinct patterns, with minimal overlaps between speech, music, and background noise components.

**Subjective Listening**:
I performed subjective evaluation by listening to the separated signals and judging their perceptual quality.
This allowed me to assess aspects such as speech intelligibility, background noise suppression, and the presence of musical content.

## Reasons for Not Using Objective SNR Metrics:

Objective metrics like **Signal-to-Noise Ratio (SNR)** typically require access to the **ground truth clean sources** for comparison.
However, this experiment is a **blind source separation** task, meaning that the original unmixed sources are unknown and unavailable.
Therefore, SNR-based evaluations were **not applicable** for me in this setting.

## Reasons for Not Using Speech-Specific Scoring:

I also chose **not to use speech-specific evaluation metrics** (e.g., PESQ, MOS, or human speech intelligibility scores) because the separated signals include **more than just speech**.
The sources contain a mixture of **speech**, **music**, and **background noise** components.
Thus, a speech-focused metric would only partially reflect the overall separation quality and would **ignore the non-speech components**, leading to an incomplete and potentially misleading assessment.

## Conclusion:

By combining spectrogram visualization and subjective listening, I was able to comprehensively and appropriately evaluate the quality of source separation under the constraints of blind separation and multi-type signal content.

## Part D: Critical Analysis of Generative AI Assistance

**Task Overview:**

If you choose to use generative AI (GenAI) tools for code generation or parameter suggestions, they must clearly document which parts were assisted by AI, and critically analyse the generated outputs.

For example,

1. Which parts of your code or parameter selection were assisted by generative AI?
2. Describe the areas where GenAI provided helpful insights and where you had to make modifications.
3. What did you learn from this process about the limitations and strengths of AI-generated solutions?

**1.** Almost all of the code used in the experiment was generated with the assistance of AI. I needed to provide the AI with my processing ideas based on what I had learned, specify the methods I wanted to apply, and guide it to generate appropriate code.
For areas I was less familiar with, I asked the AI questions and also reviewed lecture slides to better instruct the AI on how to proceed.
Regarding parameter selection, I independently designed the experiments by setting different values for FastICA parameters such as t**he number of components, fun, tol, whiten, and max_iter.**
The AI mainly assisted me in **analyzing which parameters were likely to have the most significant impact** on the separation results, and helped me **summarize and interpret the experimental findings.**

2.The AI was extremely helpful in explaining experimental concepts and generating highly accurate code, which **greatly improved my efficiency** during the experiments.
It also **helped me review key concepts from the course**, especially the workflows and implementation details of **PCA** and **ICA**.

However, there were also limitations.
For instance, in the initial design of Part C, I correctly identified that the **number of components** would have a major impact on the results, and designed experiments ranging from 2 to 7 components.
However, the AI did not remind me that **the ICA algorithm is limited** by the number of sources and cannot separate more sources than sensors.
This oversight caused me to spend a significant amount of time troubleshooting Part C.
Additionally, the AI was **unable to provide a complete working version of the Part C code** all at once, requiring me to manually **integrate and adjust** different code segments with AI assistance.

2.

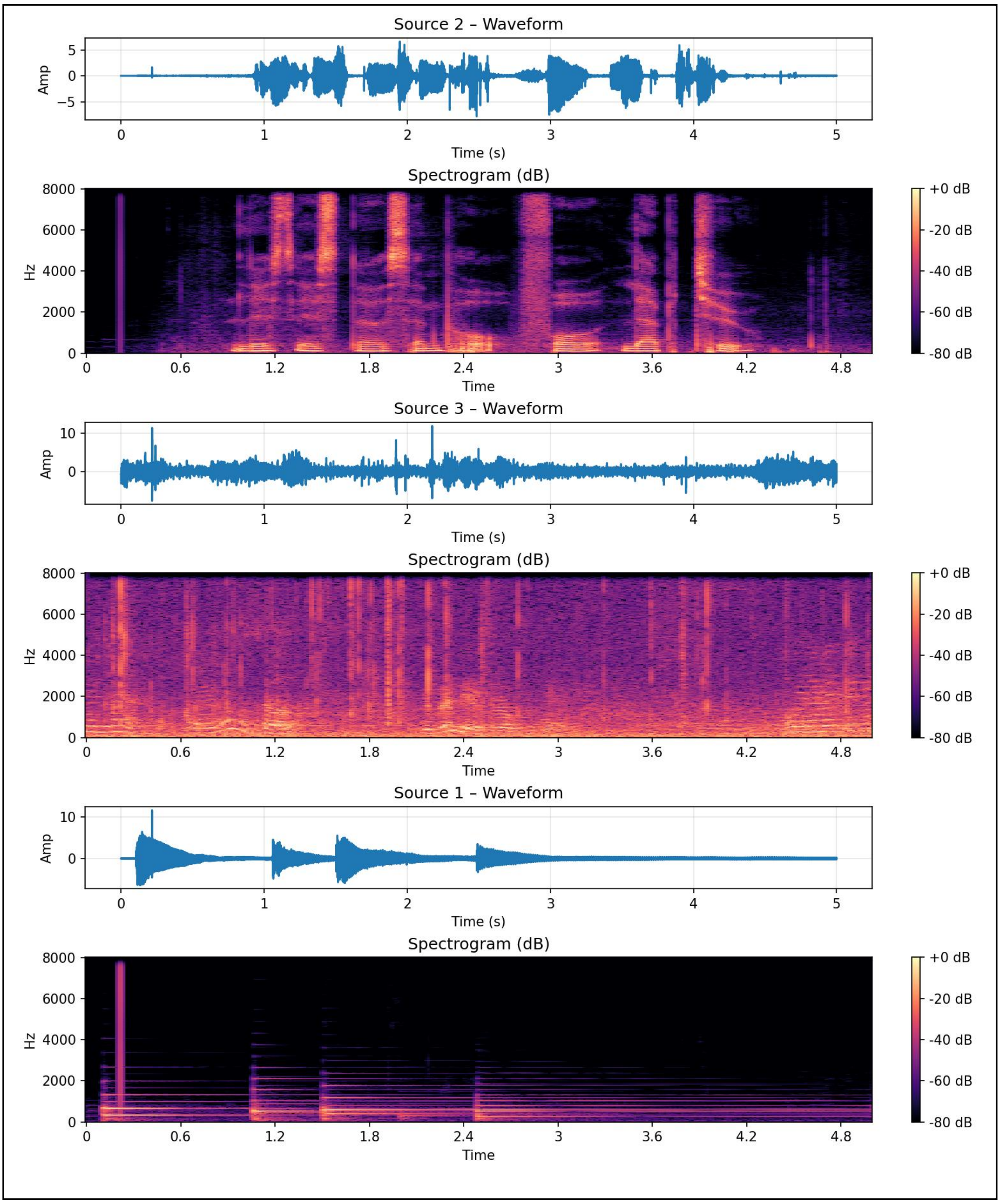
3.Through this process, I realized that the biggest limitations of AI-generated solutions are **hallucination** and **context understanding**.
Hallucination — the AI’s tendency to generate plausible but incorrect information — can easily mislead users if they are not cautious.
Furthermore, the AI's limited ability to maintain and manage **long or multi-file contexts** makes it difficult to solve complex problems that require a broad view across multiple scripts or files.Despite these limitations, AI is extremely powerful:
it can **generate complete code**, **answer targeted technical questions**, and **significantly reduce the psychological barrier to seeking help**.
I believe we should treat AI as a **tool** — by learning how to use it effectively and critically, we can better adapt to a world where generative AI plays an important role.

## Part E: Create your own Audio (Optional)

This section is optional, but completing it can earn you additional marks.

**Task**: Record your own audio clip with different sources, such as other people talking. Apply source separation techniques to separate your voice from the other sources. Successfully doing so will earn you extra credit.

The audio must be complex, with multiple sources (e.g. multiple people talking in the background with music or any other noise). Simple recordings with no or minimal background noise will not be accepted for this task. This is to experiment creatively while demonstrating your understanding of audio processing techniques.



For this task, I recorded a complex audio mixture that contained multiple sources, including background noise from a crowded environment, instrumental music, and my own voice.
The recording fulfilled the requirement of complexity, as it featured overlapping speech, environmental noise, and additional audio elements.

I then applied **source separation techniques** using **FastICA**.
The separation results were as follows:

**Source 1**: Instrumental music

**Source 2**: My own voice

**Source 3**: Background noise from the environment

Through this process, I was able to successfully isolate my voice from the other audio sources.
The results demonstrate my understanding of both the challenges and techniques involved in blind source separation for real-world, multi-source audio recordings.