Fazlollahnezhad_Shivaei

Title: Independent Component Analysis (ICA): A Comprehensive Report

Abstract This report explores Independent Component Analysis (ICA), detailing its application, theoretical foundation, and evaluation in signal processing. Using synthetic signals, the methodology applies preprocessing techniques such as PCA and ICA (via the JADE algorithm) to achieve blind source separation. Comprehensive distribution analysis is integrated into the evaluation, with visualizations comparing original, mixed, and recovered signals. Results confirm ICA's potential to separate mixed signals into statistically independent components.

1. Introduction Independent Component Analysis (ICA) is a statistical technique for separating mixed signals into their independent sources. Widely applied in fields such as biomedical engineering and audio signal processing, ICA is uniquely suited for extracting non-Gaussian independent components from complex datasets. This study demonstrates ICA's utility through synthetic data generation, preprocessing with PCA, and implementing the JADE algorithm for source separation. The inclusion of enhanced distribution analysis and detailed visualizations offers deeper insight into the methodology and results.

2. Methodology

2.1 **Data Import and Setup** The experiment begins with setting up the computational environment using NumPy for numerical computations and Matplotlib for plotting. A custom implementation of the JADE algorithm facilitates ICA. This structured setup ensures reproducibility and computational efficiency.

2.2 **Synthetic Signal Generation** Synthetic signals, designed with distinct non-Gaussian features such as sine waves and uniform distributions, are created to simulate real-world independent sources. These signals are combined via a mixing matrix to generate observed mixtures, mirroring the complexity of real-world data.

2.3 **Visualizing Original and Mixed Signals** Line plots display the independent original signals and their mixed counterparts, highlighting the challenges of source separation. The original signals exhibit clear independence, whereas the mixed signals reveal overlapping patterns requiring ICA for separation.

2.4.1 **Analysis** Histograms and density plots of original and mixed signals are analyzed to explore their statistical properties. The original signals exhibit non-Gaussian distributions, essential for ICA's success, while mixed signals approached Gaussian due to the central limit theorem.

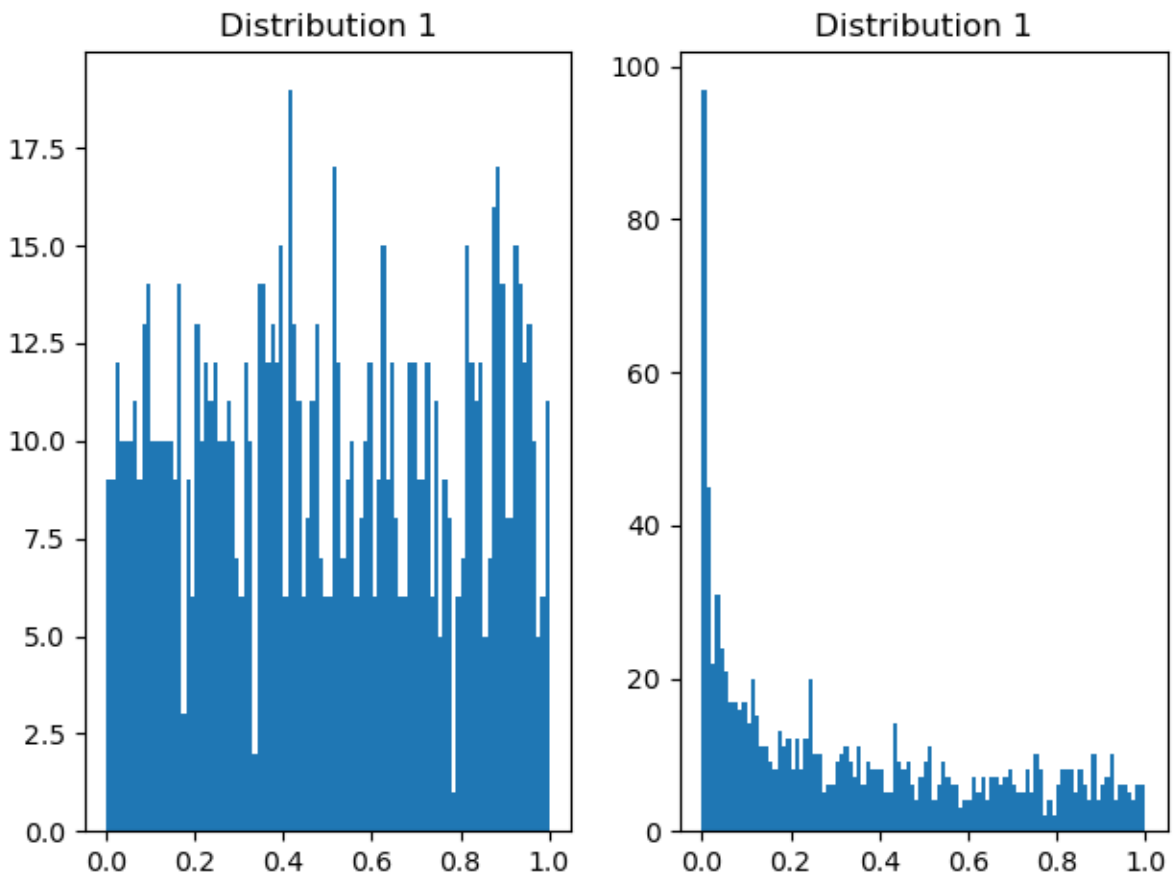

Figure 1: Original Independent Signals (Line plot showcasing the individual original signals)

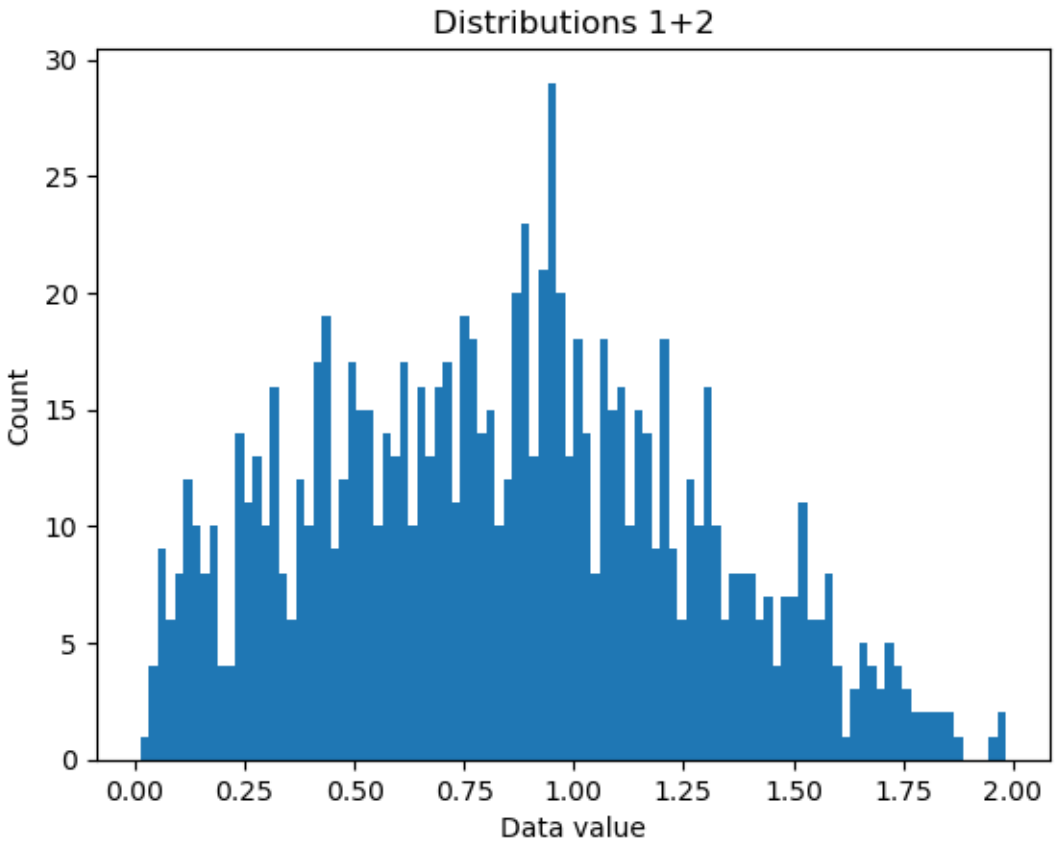

Figure 2: Mixed Signals (Line plot of the signals after being mixed)

2.4.2 **Analysis** This plot reveals the distinct non-Gaussian nature of the original signals. Each signal demonstrates unique statistical characteristics, which are crucial for successful source separation using ICA.

The histograms of the mixed signals show a shift towards Gaussian distributions. This transformation arises from the linear combination of original signals.
Although the different scale of the histograms of ICs, they looks pretty similar to Original Data which represents non-gaussian pattern.

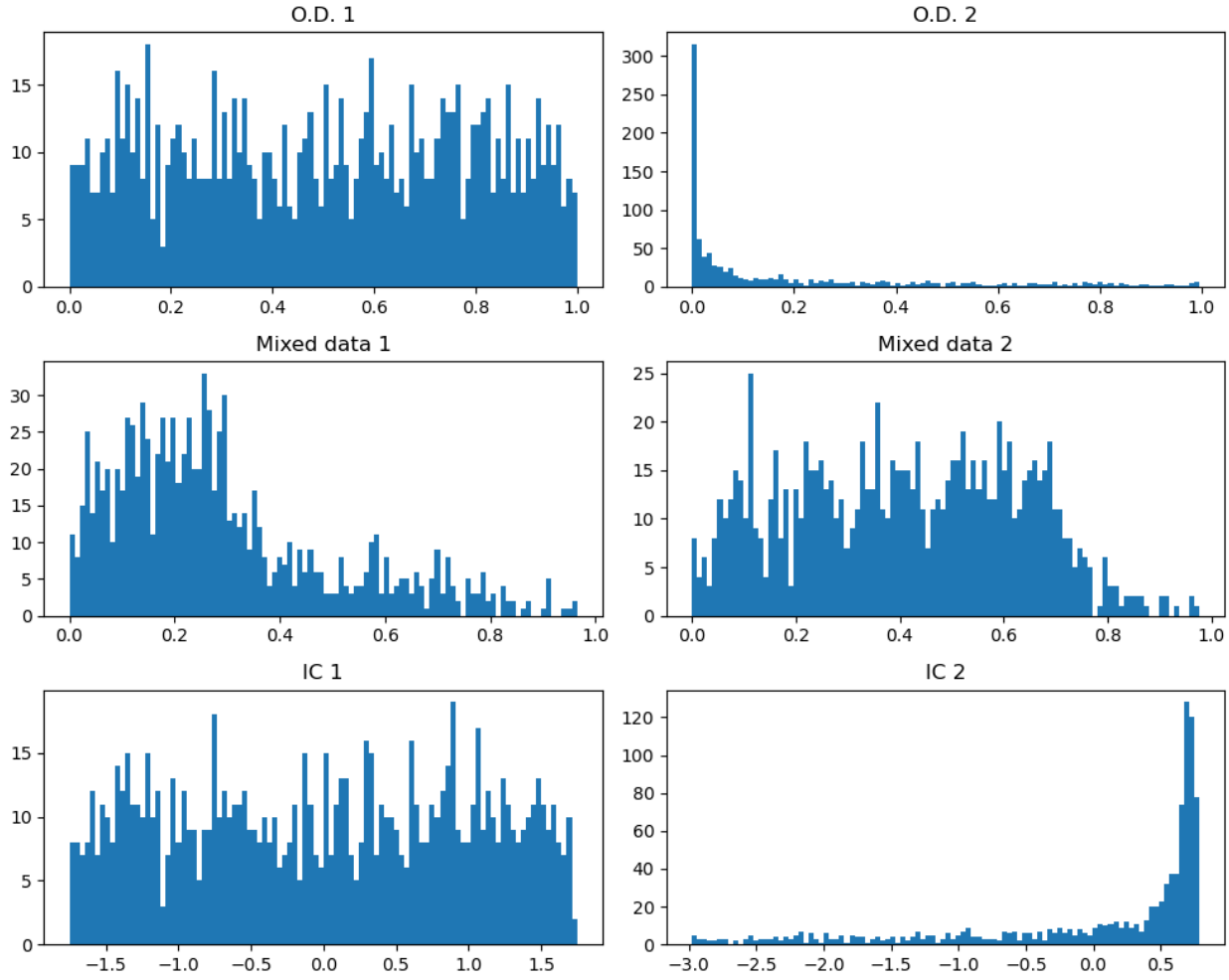

Figure 3: Histograms of Original Data, Mixed Signals and Independent Components.

2.4.3 **Analysis** The density plots reinforce the observations from the histograms, highlighting the non-Gaussian nature of the original signals. These distributions are pivotal for the ICA algorithm and it is evident that Independent Components are correlating to original data.

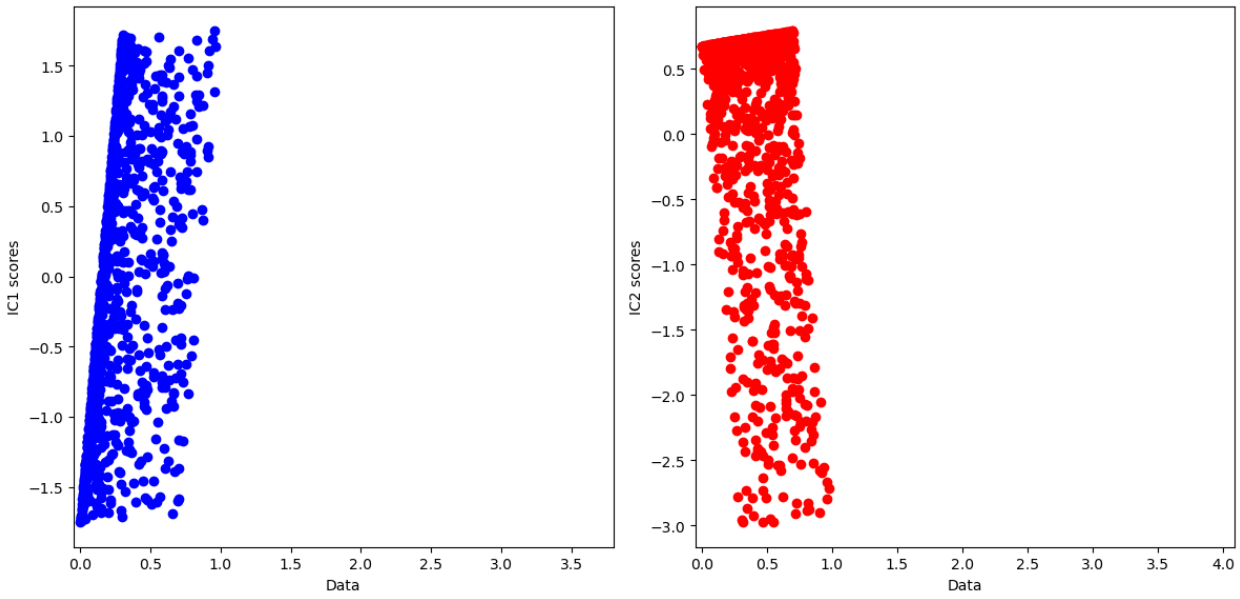

Figure 4: Density Plots of Original Signals (Kernel density estimate plots of original signals)

2.4.4 **Analysis** Similar to the histograms, the density plots show the Gaussian-like characteristics of the mixed signals, underscoring the challenge of recovering the original independent sources because they have really tight correlation between the IC1 and Data stream 1, IC2 and Data Stream 2.

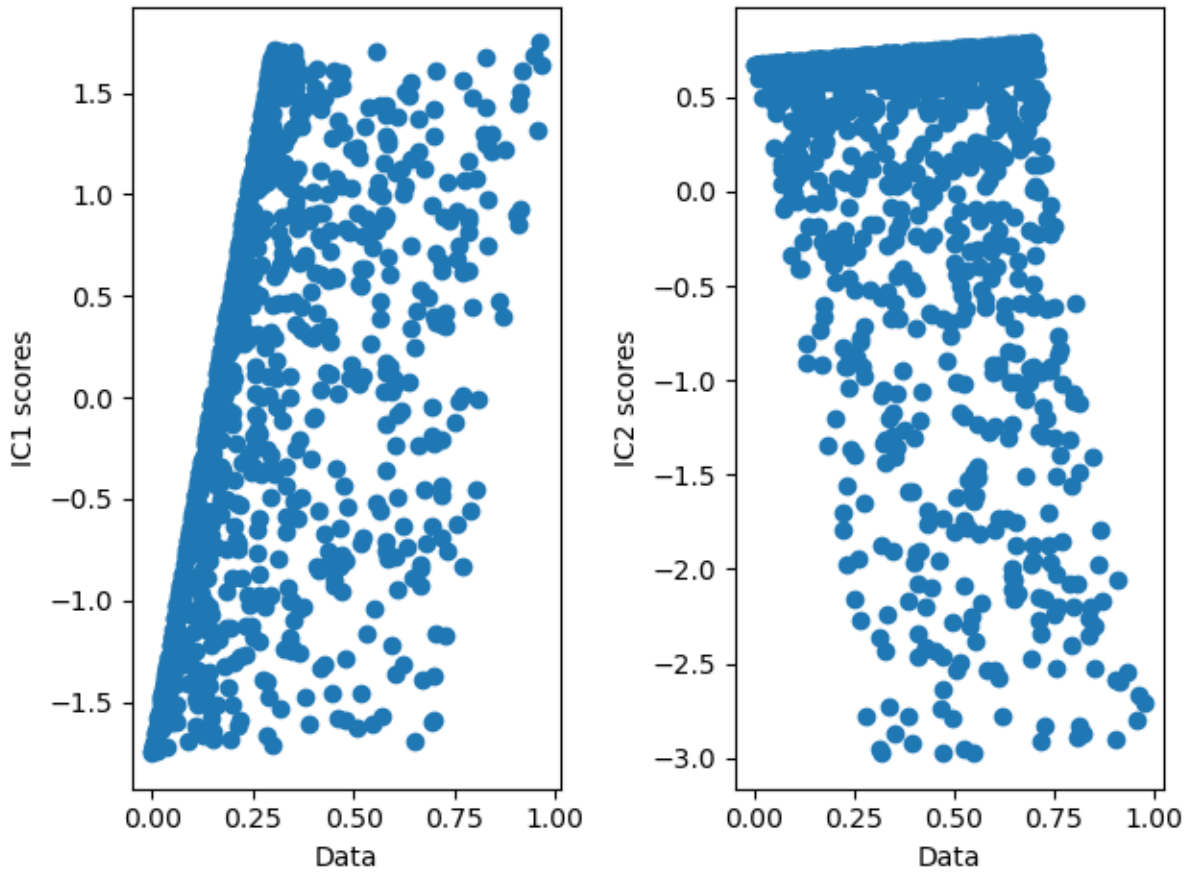

Figure 5: Density Plots of Mixed Signals (Kernel density estimate plots of mixed signals)

2.4.5 **Analysis** A Gaussian curve is applied as a filter in the frequency domain, with its width controlled by the Full Width at Half Maximum (FWHM) parameter. The FWHM is varied logarithmically across a range of values (0.1 Hz to 10 Hz) to illustrate its effect on the frequency content of the signal. Random Fourier coefficients are generated and tapered by the Gaussian filter, transforming the resulting frequency-domain signal back into the time domain using the inverse Fast Fourier Transform (FFT). Narrow FWHM results in a signal with concentrated frequency content, producing a time-domain signal that resembles a nearly pure sinusoid. In contrast, wide FWHM leads to a signal with broader frequency content and increased noise-like behavior in the time domain.

This noise-like behavior is essential for simulating EEG signals, which naturally exhibit complex and dynamic frequency patterns. The resemblance to EEG signals makes wider FWHM filtering more suitable for capturing the realistic characteristics of neural activity, where diverse frequency content and noise-like behavior are prevalent.

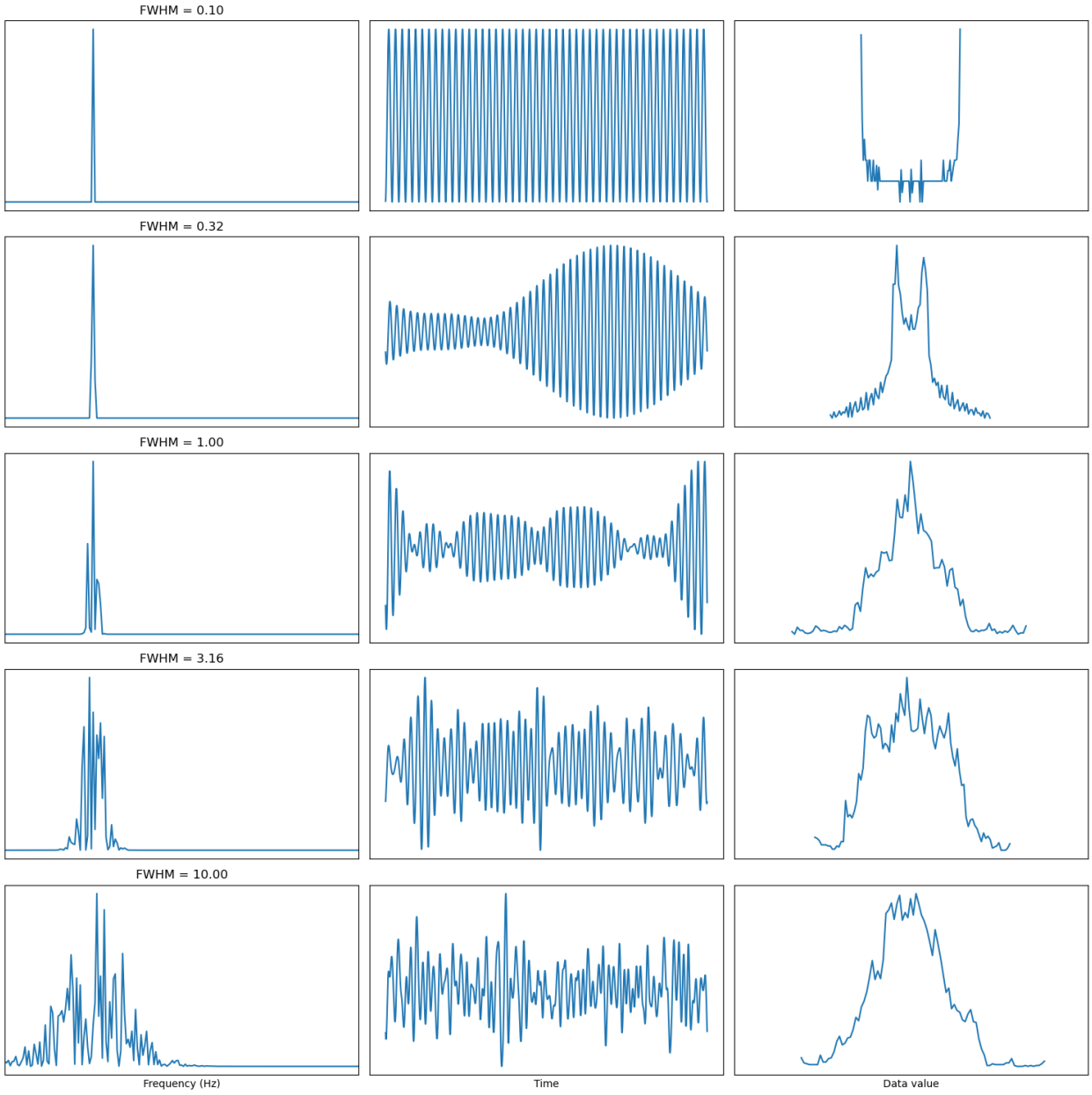

Figure 6: visualization of how the Full Width at Half Maximum (FWHM) of a Gaussian distribution in the frequency domain influences signals in both the time and data-value domains.

2.5.1 **Explore IC Distributions in Real Data** This section applies ICA to real-world data, analyzing the independent components extracted. Each visualization is described and interpreted below:

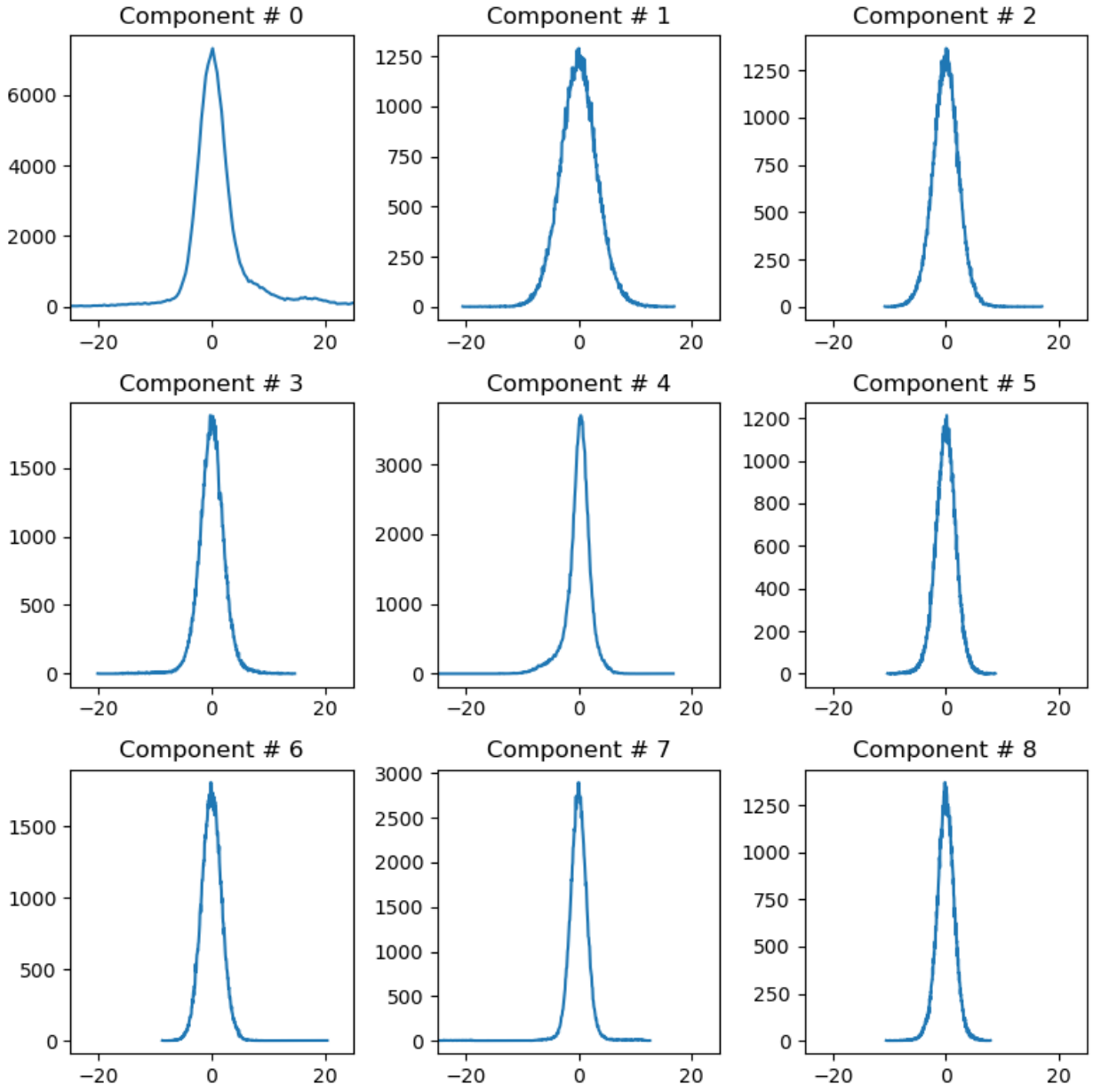

Figure 7: Histograms of Independent Components in Real Data (Illustrates the non-Gaussian nature of ICs)

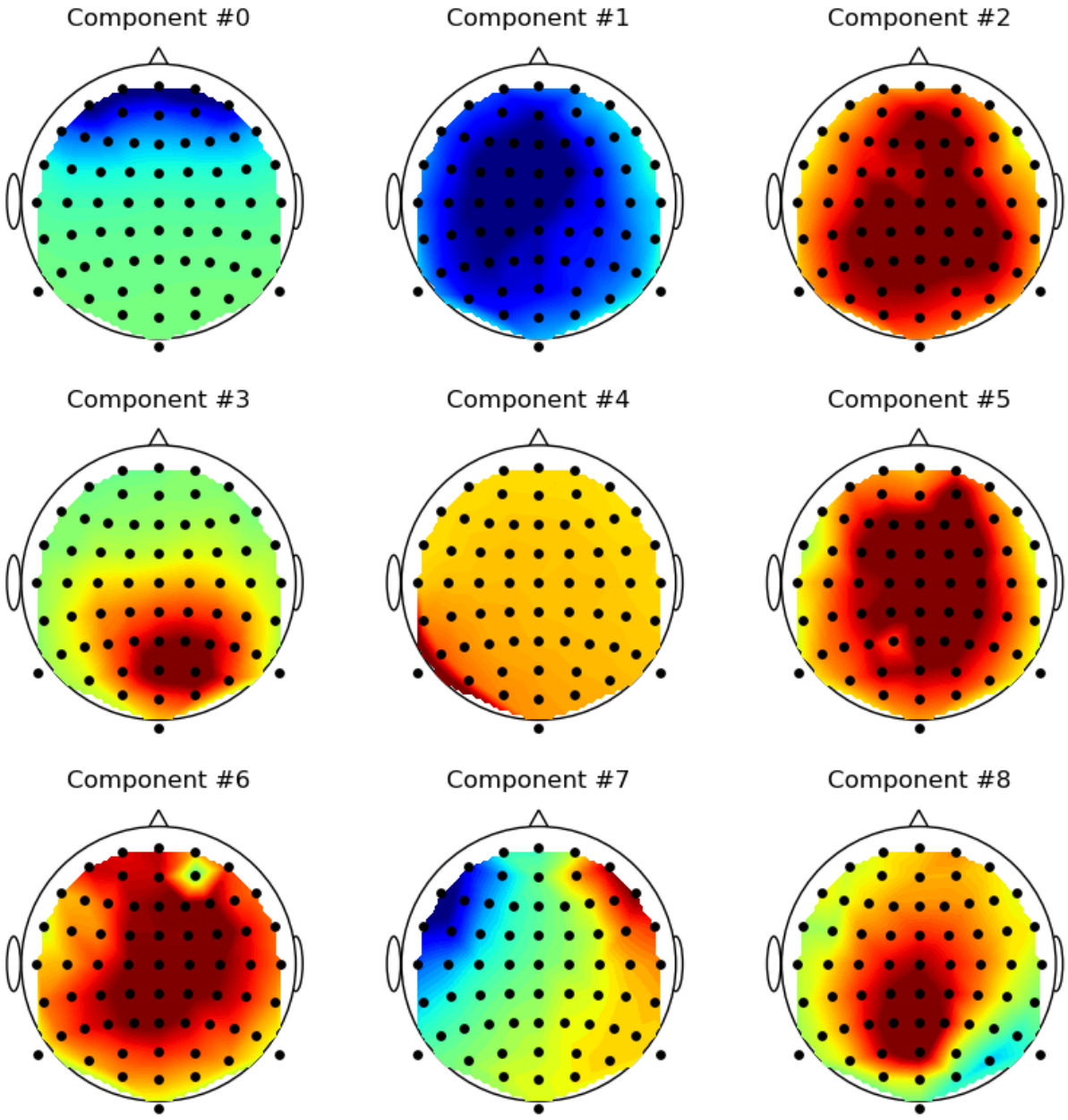

Figure 7.1: topographical maps for nine independent components *in Real Data.*

2.5.2 The histograms show the distributions of the extracted independent components (ICs). The observed non-Gaussian shapes indicate that the ICA algorithm has successfully separated the components, recovering their statistical independence.

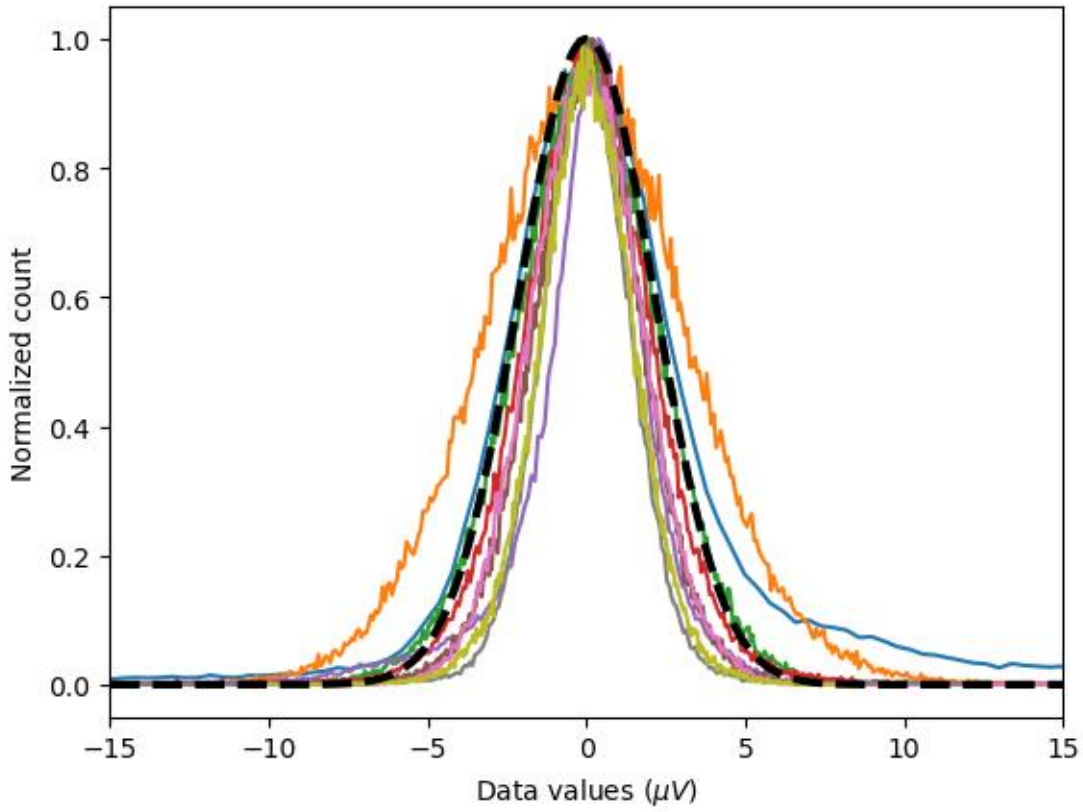

Figure 8: Histogram of merged Independent Components in Real Data (Illustrates the non-Gaussian nature of ICs)

2.6 **Principal Component Analysis (PCA)** PCA preprocesses the mixed signals, reducing dimensionality and decorrelating the data. This simplifies ICA's task by transforming the data into orthogonal components, aligning them along principal axes of variance.

The XY plot illustrates how to streams of data different from each other, the red lines represents two eigen factors and of course they are orthogonal to each others, the PCA represents that the two variance directions which means PCA code is correct but this is not the answer we are investigating.
The black lines are related to jade algorithm which are the independent components because they are not orthogonal to each others which the third plots represents the XY data in IC space.
In conclusion PC tells tell us which direction has the most covariance in the entire dataset and the IC full seperates the two steams.

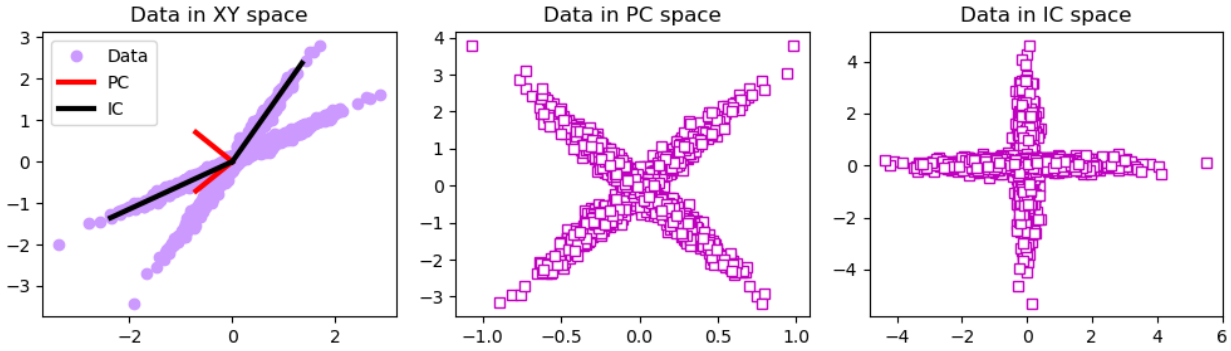

Figure 9: Density Plots of Independent Components in Real Data (Kernel density estimate plots of ICs)

2.7 **ICA, PCA, and GED on Simulated Data** This section compares the performance of ICA, PCA, and Generalized Eigenvalue Decomposition (GED) on simulated data. Visualizations and explanations are provided for each method:

2.7.1 This code simulates EEG data based on a lead field model and visualizes brain dipoles, their scalp projections, and corresponding time-series signals. It begins by loading a .mat file containing EEG data and a lead field model that maps dipole sources in the brain to scalp electrodes. A matrix representing the normal dipole orientations (aligned with the cortical surface) is calculated. Random dipole activity is generated, with a strong sinusoidal signal (10 Hz) added to a specific dipole to simulate a localized source of neural activity. This dipole activity is projected to scalp electrodes to create EEG data.

The code then visualizes the setup in three ways: a 3D plot showing the dipole grid in the brain with the active dipole highlighted, a scalp topography of the active dipole's projection, and a time-series plot comparing the original dipole signal to the corresponding electrode signal. This approach combines simulation and visualization to study how brain activity projects onto EEG electrodes and how specific neural sources influence scalp recordings.

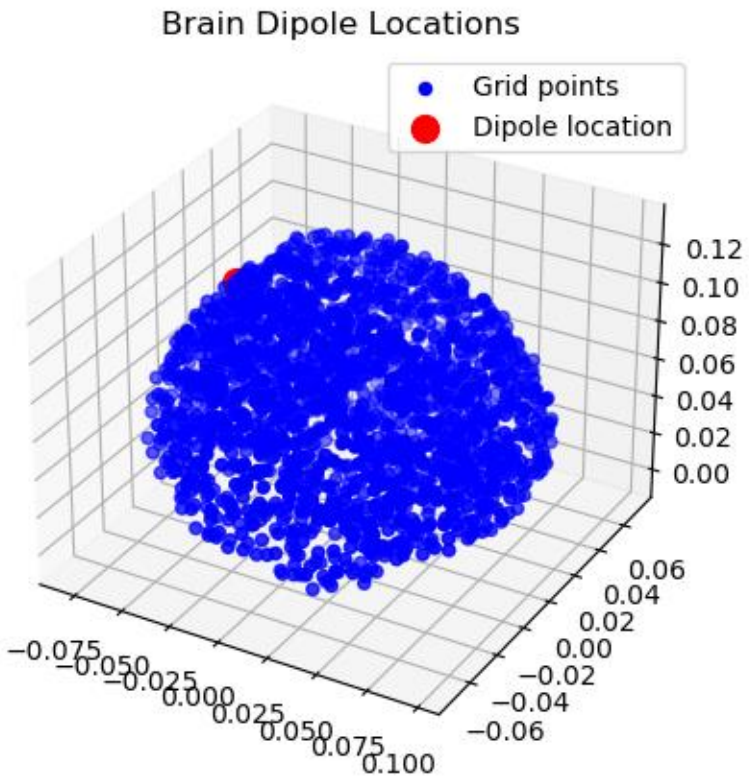

Figure 9: Simulation of a signal in 1 dipole(red dot) and noise in another dipoles.

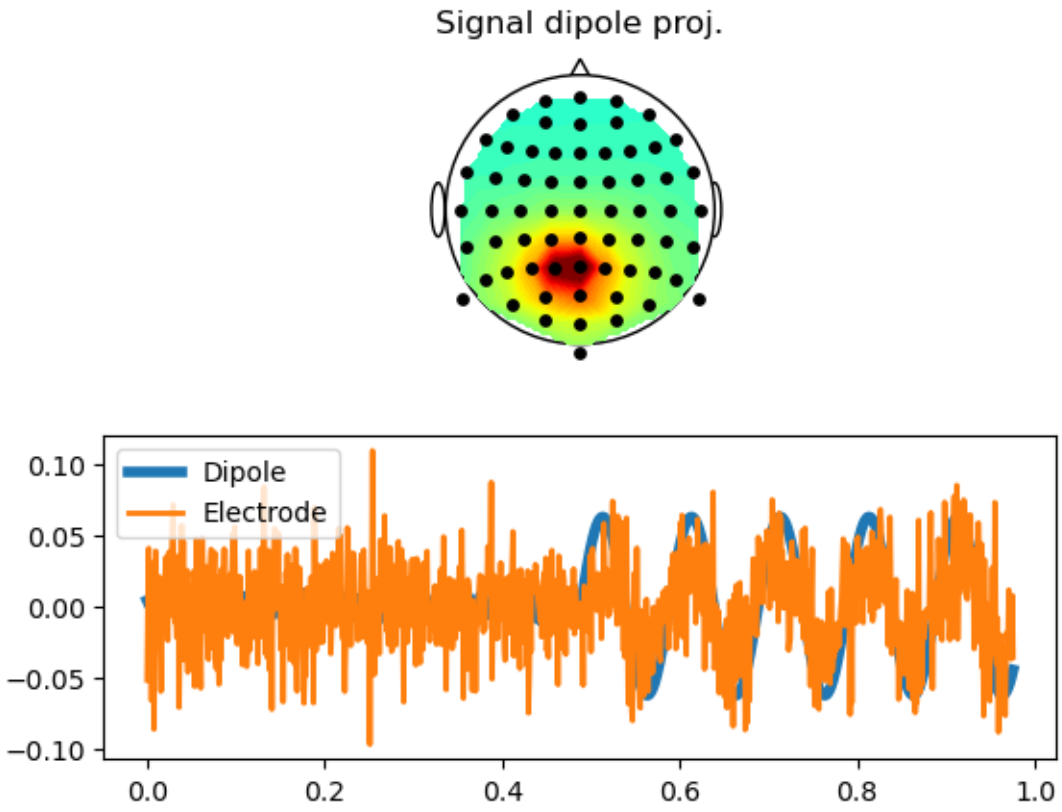

Figure 9.1: *representing signal in dipole location.*

2.7.2 This plot displays the signals extracted by ICA, PCA, and GED. ICA successfully recovers the independent sources, while PCA outputs decorrelated but dependent components. GED exhibits strengths in specific tasks but cannot achieve full independence like ICA.

2.7.3 covariance matrices from two halves of EEG data to compare the statistical relationships between electrodes over time, helping to identify patterns and changes in brain activity. The first half of the data is used to compute the covariance matrix RR, while the second half computes SS. By comparing these matrices, the analysis reveals how the statistical dependencies (or correlations) between electrode signals evolve, which is useful for detecting shifts in brain states or responses to stimuli.

Additionally, the code computes and visualizes R−1SR^{-1}S, a transformation that highlights how the covariance structure in the second period relates to the first. This step is common in multivariate signal processing to extract dominant patterns or features, aiding in tasks like brain state classification or artifact detection. Overall, this process is essential for understanding temporal dynamics and structural changes in EEG data.

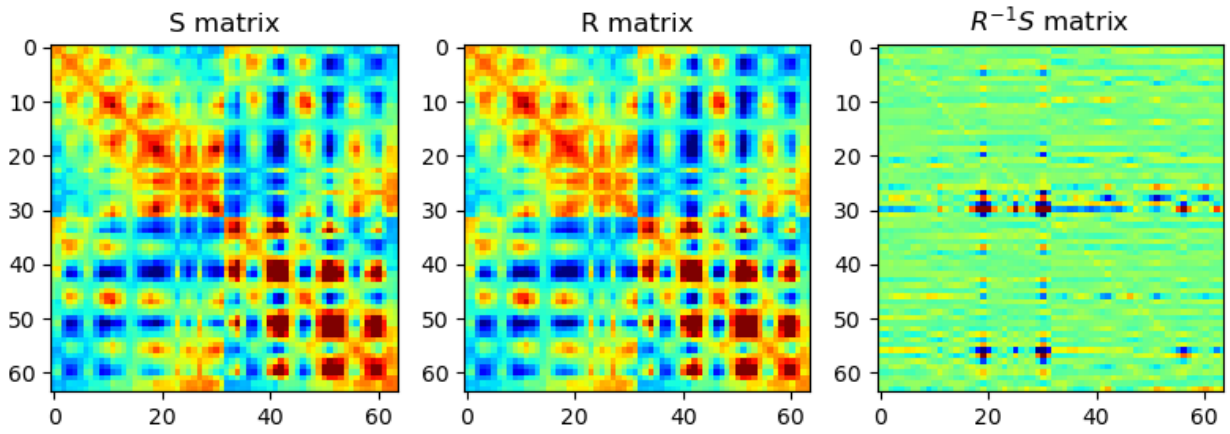

Figure 10: Covariance matrices.

2.7.4 **Principal Component Analysis (PCA)** is to compress the EEG data's dimensionality and extract key spatial and temporal features. Using the covariance matrices (RR and SS), the eigenvalues and eigenvectors are computed to identify the principal components (PCs). The eigenvalues represent the variance explained by each PC, while the eigenvectors define spatial patterns associated with those components.

The first principal component (associated with the largest eigenvalue) is analyzed in detail. Its topography (spatial distribution across electrodes) is compared to the true dipole's projection on the scalp, showing how well PCA identifies the underlying source. The PCA spatial filter is also applied to the EEG data to extract a time series, which is compared to the true dipole signal and an individual electrode's signal, demonstrating PCA's ability to reconstruct dominant patterns in the data.

Overall, this code uses PCA as a tool to reduce dimensionality, analyze signal variability, and highlight the spatial and temporal correspondence between the true dipole signal and the EEG data, building on the covariance analysis to reveal key neural sources.

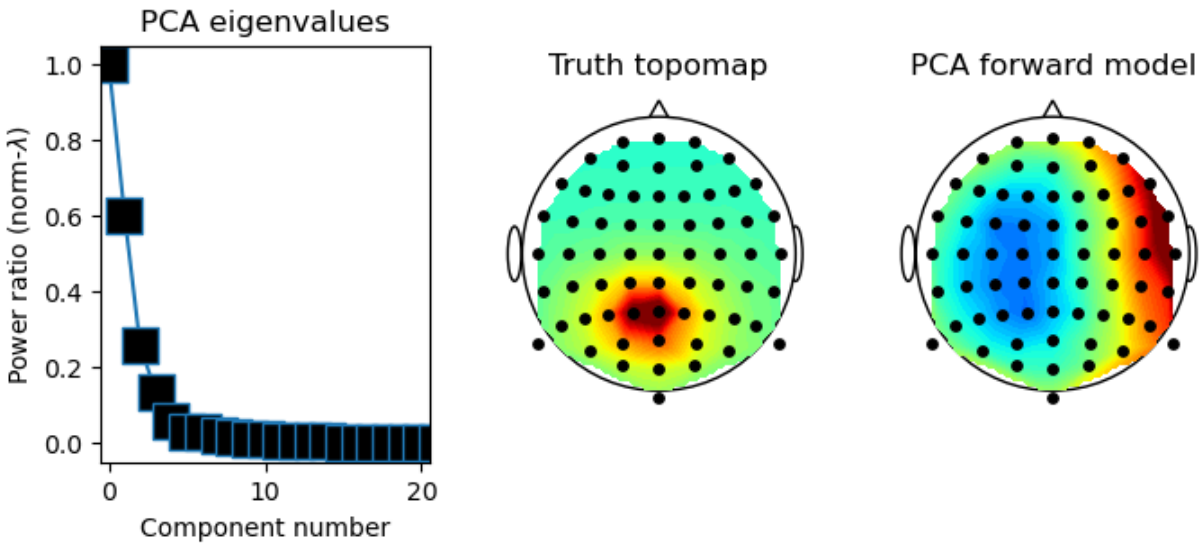

Figure 10.1: PCA eigenvalues and topographical maps of Components Extracted Using PCA

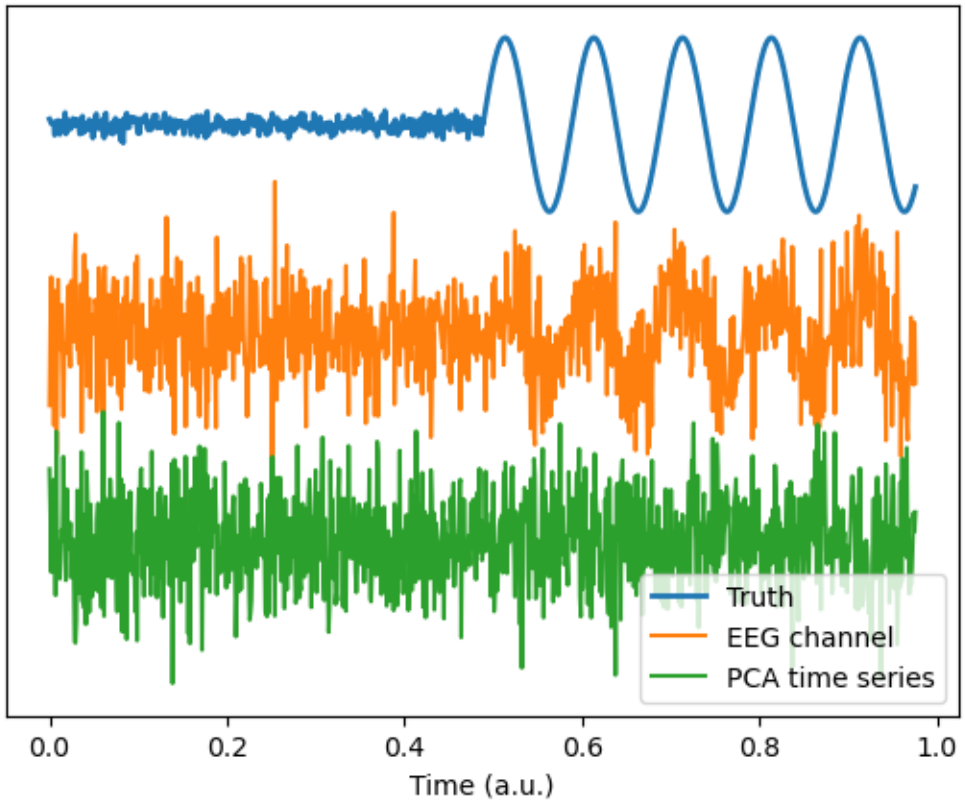

Figure 10.2: Distribution Analysis of Components Extracted Using PCA

Histograms and density plots provide a statistical comparison of the components extracted by each method. ICA's components exhibit non-Gaussian distributions, affirming their independence, whereas PCA and GED produce components with overlapping statistical properties.

2.8 **Independent Component Analysis with JADE** The JADE algorithm applies ICA, leveraging higher-order statistics to separate the mixed signals into statistically independent components. By focusing on kurtosis and joint diagonalization, JADE isolates independent sources with precision.

2.9 **Visualizing Recovered Signals** Recovered signals are compared to original sources using line and scatter plots. The scatter plots demonstrate statistical independence, and the visual similarity confirms the algorithm's success.

2.9.1 **Generalized Eigenvalue Decomposition (GED)** is to identify and analyze the most dominant patterns in EEG data by comparing the relationships between two segments of the data (the first and second halves). GED works by finding components that show the largest differences in variability between these two periods. The strength of each component is measured, and the most important one is analyzed further.

The top spatial pattern (forward model) of the strongest component is compared to the true simulated dipole to see how well GED identifies the underlying source. The code also extracts a time series for this component by applying its spatial filter to the EEG data. This time series is then compared to the true dipole signal and a single electrode's signal to demonstrate how well GED isolates the dominant source in both space and time. Overall, the code helps to separate and highlight key brain activity patterns in the EEG data, focusing on differences between the two data segments.

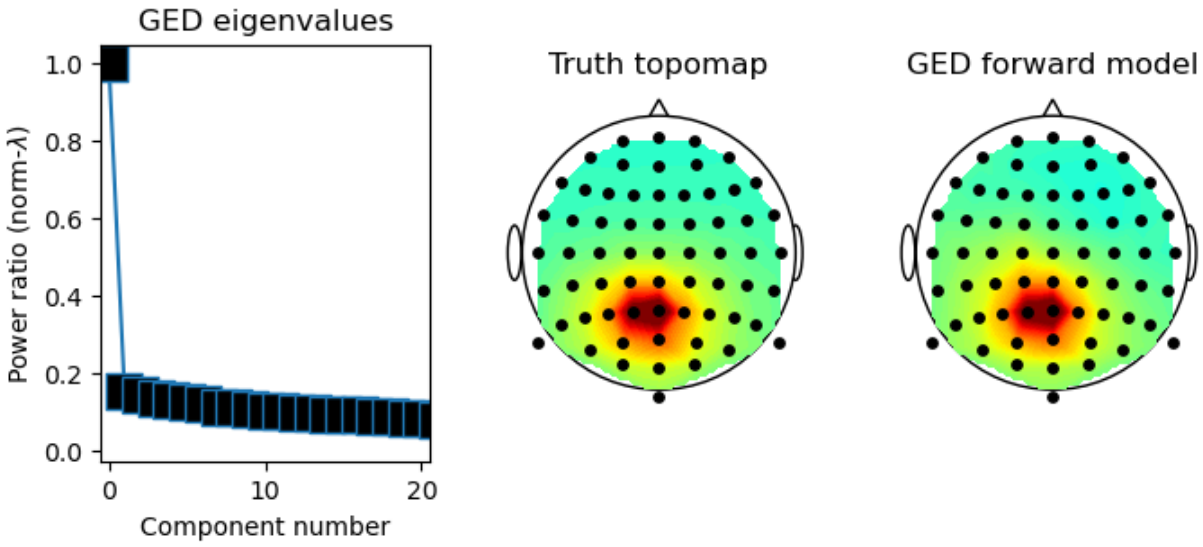

Figure 11.1: GED eigenvalues and topographical maps of Components Extracted Using GED

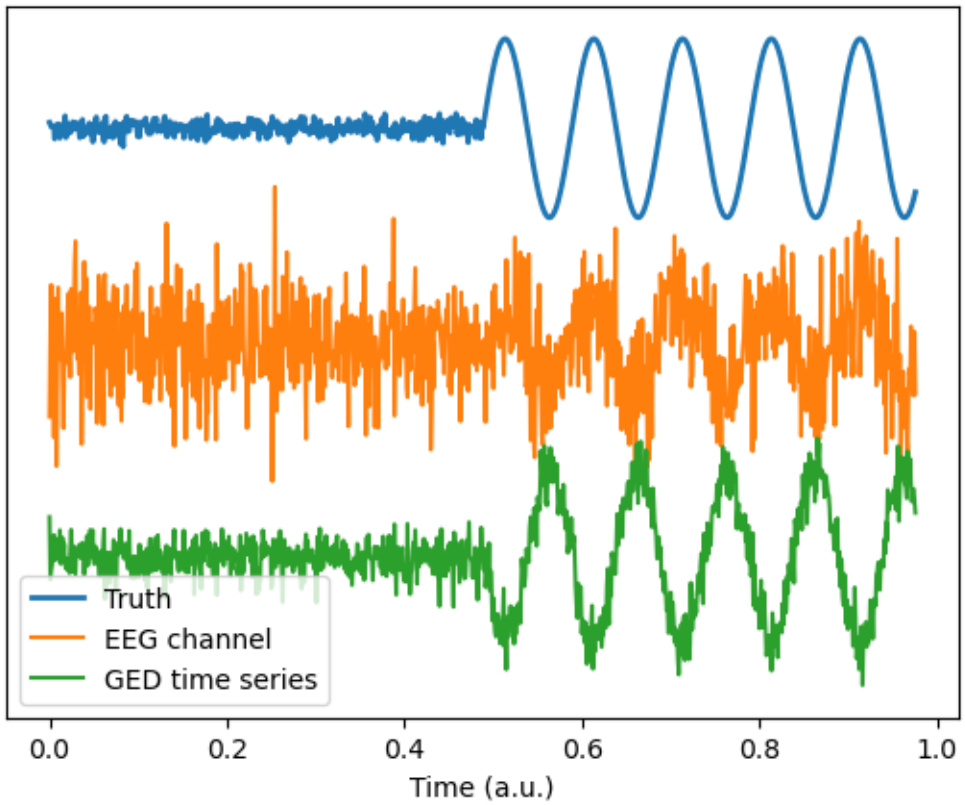

Figure 11.2: Distribution Analysis of Components Extracted Using GED

2.9.2 **Independent Component Analysis (ICA)** is to separate EEG data into independent sources, isolating distinct patterns of brain activity or noise. ICA identifies spatial maps that show how each source contributes to different electrodes and corresponding time series that represent the activity of these sources over time. The most dominant component is analyzed by comparing its spatial map to the true dipole map and its time series to the true dipole signal and a single electrode's signal. This demonstrates how well ICA can identify and reconstruct the dominant source in both space and time, making it a powerful tool for analyzing and cleaning EEG data.

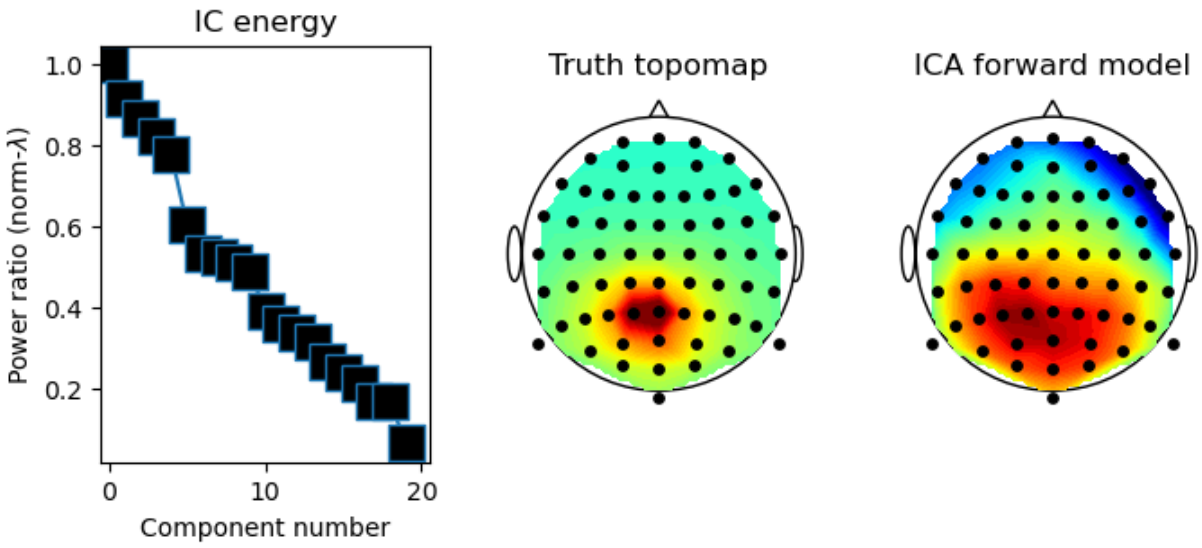

Figure 12.1: Distribution Analysis of Components Extracted Using ICA

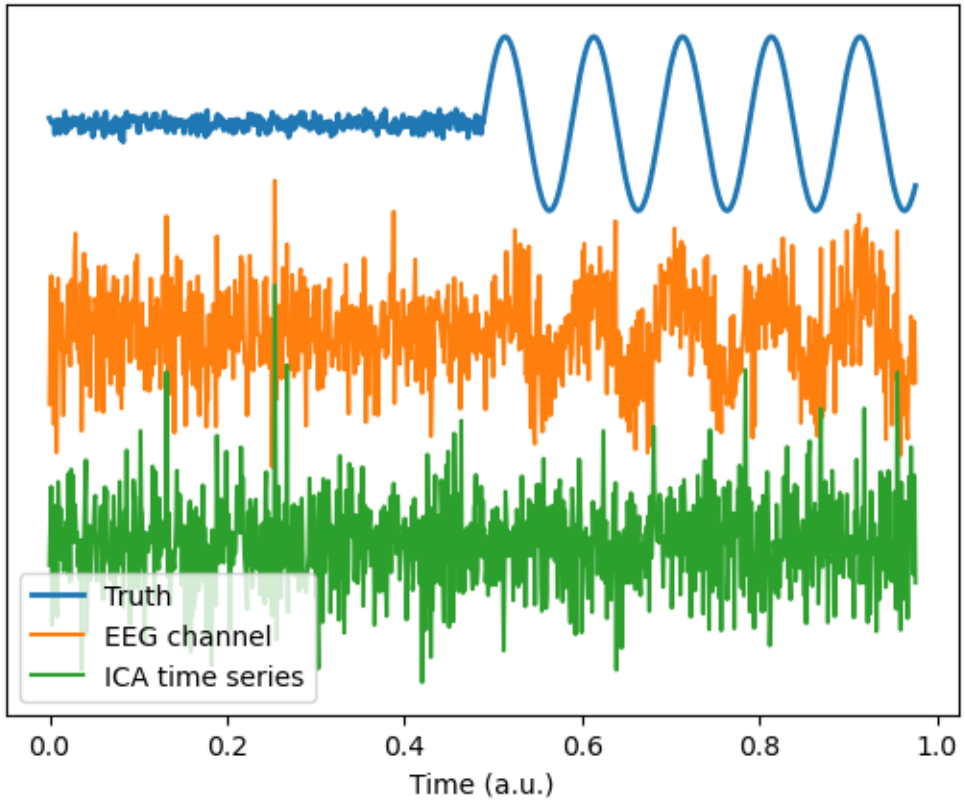

Figure 12.2: Distribution Analysis of Components Extracted Using ICA

The scatter plots illustrate the statistical independence of the recovered signals. The lack of linear relationships between pairs confirms that ICA has successfully separated the sources.

3. Results

3.1 **Original and Mixed Signals** The synthetic signals demonstrate distinct patterns, while their mixtures represent realistic, overlapping datasets. These mixtures provide a challenging input for ICA.

3.2 **Distribution Analysis** Histograms and density plots illustrate the non-Gaussian nature of the original signals and the mixed signals' transformation due to linear combinations. These insights reinforce the necessity of ICA for separating the sources.

3.3 **Recovered Signals** The recovered signals align closely with the original sources, and scatter plots confirm their statistical independence. This demonstrates the JADE algorithm's efficiency in performing source separation.

3.4 **Comparison of ICA, PCA, and GED** The evaluation highlights the unique strengths of each method. PCA efficiently decorrelates data, providing a simplified signal space, but it cannot achieve full independence. GED effectively addresses specific tasks requiring eigenvalue decomposition but lacks ICA's generality. ICA, particularly with JADE, emerges as the most effective approach for blind source separation, achieving high independence and accuracy.

4. Conclusion This study highlights the complementary roles of PCA and ICA in data analysis and blind source separation. PCA is effective for decorrelating data and simplifying signal space through dimensionality reduction, while ICA, particularly using the JADE algorithm, separates independent components by leveraging higher-order statistics. ICA proves essential for isolating non-Gaussian sources, making it invaluable for tasks requiring detailed signal separation. The incorporation of distribution analysis deepens the understanding of the data’s statistical properties, validating the effectiveness of these methods. However, limitations such as sensitivity to noise and the reliance on non-Gaussianity are acknowledged. Future research should focus on enhancing ICA for noisy or nonlinear datasets, applying these methods to real-world data, and exploring hybrid approaches for improved performance and broader applicability.