EEG-Kuramoto Neural Dynamics Framework

Modular Implementation Report and Documentation

Yehia Said Gewily

April, 2025

Overview

The EEG-Kuramoto Neural Dynamics Framework has been comprehensively restructured into a modular and scalable Python package. This transformation enhances maintainability, usability, and extensibility, making the framework suitable for both research and production use in neuroscience and computational modeling.

Key Changes and Enhancements

Modular Package Structure

The codebase has been reorganized into logical submodules for clarity and separation of concerns:

```
eeg_kuramoto/
          eeg/              # EEG data preprocessing and handling
          models/           # Kuramoto model implementations
          analysis/         # Closed-loop analysis and tuning tools
          utils/            # Utility functions and visualization helpers
          examples/         # Example usage scripts
          __init__.py       # Root-level imports
          setup.py          # Package installation configuration
```

Documentation and Readability

- Added comprehensive docstrings to all modules, classes, and functions.
- Included type hints and parameter descriptions.
- Created a detailed `README.md` with setup and usage instructions.

Robust Error Handling

- Validated all input arguments with informative error messages.
- Implemented defensive programming practices for edge cases.

Installation and Setup

A `setup.py` file is included to allow pip installation. Proper `__init__.py` files are used for import consistency.

```
pip install -e .
```

Example Code and Demonstrations

To run a complete demonstration pipeline:

```
python -m eeg_kuramoto.examples.run_pipeline
```

This example includes:

- EEG signal preprocessing
- Time-varying Kuramoto model simulation
- Real-time feedback and closed-loop refinement
- Visualization of output dynamics

Benefits of the New Architecture

Benefit	Description
Maintainability	Easier to test, debug, and modify each module independently
Extensibility	New components (models, preprocessors, visualizations) can be added easily
Usability	Clean API with logical imports and example scripts
Readability	Fully documented and structured code
Collaboration	Multiple developers can contribute without conflicts

Getting Started

1. Installation

```
git clone https://github.com/your-repo/eeg-kuramoto
cd eeg-kuramoto
pip install -e .
```

2. Basic Usage

```
from eeg_kuramoto.eeg import EEGDataInversion, EEGGenerator
from eeg_kuramoto.models import TimeVaryingKuramoto
from eeg_kuramoto.analysis import ClosedLoopRefiner
```

3. Run Example Pipeline

```
python -m eeg_kuramoto.examples.run_pipeline
```

Future Directions

- Integration with real-time EEG hardware
- GUI for visualizing neural dynamics
- Expansion of model library beyond Kuramoto
- Unit tests and CI/CD integration