

GastroPy: An Open-Source Python Toolbox for Electrogastrography Signal Processing and Gastric-Brain Coupling Analysis

Micah Allen^{1,2}

¹Center of Functionally Integrative Neuroscience, Aarhus University, Denmark

²Cambridge Psychiatry, University of Cambridge, United Kingdom

Corresponding author: `micah@cfin.au.dk`

Abstract

Electrogastrography (EGG) is a non-invasive technique for recording gastric myoelectrical activity via cutaneous abdominal electrodes. Recent advances in concurrent EGG-fMRI have revealed phase synchrony between the gastric rhythm and resting-state brain networks, opening a new window into brain-body interactions, interoception, and mental health. Despite growing interest, the field lacks a dedicated, open-source software toolkit for EGG analysis in Python. Here we present GastroPy, a modular Python package for EGG signal processing and gastric-brain coupling analysis. GastroPy provides a full EGG processing pipeline—spectral analysis, bandpass filtering, phase extraction, cycle-level metrics, and phase-based artifact detection—together with multichannel processing via ICA-based spatial denoising, time-domain preprocessing for spike and movement artifact removal, and time-frequency decomposition. A dedicated neuroimaging module supports the complete EGG-fMRI coupling workflow: scanner trigger alignment, confound regression, voxelwise BOLD phase extraction, and whole-brain phase-locking value (PLV) mapping with surrogate-based statistical testing. The package supports named, citable preprocessing pipelines for reproducibility across labs. Built on NumPy and SciPy with optional MNE-Python and nilearn (Abraham et al., 2014) integration, GastroPy separates core signal processing from neuroimaging workflows and includes bundled sample data for tutorials and testing. GastroPy is freely available under the MIT license at <https://github.com/emodied-computation-group/gastropy>.

1 Introduction

The stomach generates a continuous electrical slow wave at approximately 0.05 Hz (3 cycles per minute) that originates in the interstitial cells of Cajal (ICC) located in the myenteric

plexus of the gastric corpus, and propagates distally through the antrum to coordinate gastric motility (Koch and Stern, 2004; Chen and McCallum, 1991). This rhythm, present in both fasting and postprandial states, serves as the pacemaker signal that triggers smooth muscle contraction and governs the mixing and propulsion of gastric contents. The gastric slow wave can be recorded non-invasively using electrogastrography (EGG), a technique in which cutaneous electrodes placed on the abdominal skin surface capture the far-field projection of the underlying myoelectrical activity (Chang, 2005; Yin and Chen, 2013). The resulting signal, though weak in amplitude and susceptible to respiratory, cardiac, and motion artifacts, provides a continuous measure of gastric rhythmicity that can be characterized through spectral analysis, phase extraction, and cycle-level metrics.

Clinical EGG has been used for decades to assess gastric dysrhythmias (bradygastria, tachygastria), gastroparesis, functional dyspepsia, and other motility disorders (Koch and Stern, 2004). Standard clinical parameters include the dominant frequency, percentage of recording time in the normal 2–4 cycles per minute (cpm) range, the power ratio between pre- and postprandial recordings, and the instability coefficient quantifying rhythm regularity. More recently, the technique has attracted interest from neuroscience and psychophysiology, driven by discoveries of gastric-brain interactions, and from affective computing, where wearable EGG devices have been explored for affect detection (Vujic et al., 2020).

Concurrent EGG-fMRI recordings have revealed that the gastric rhythm is phase-synchronized with resting-state brain activity. Rebollo et al. (2018) identified a distributed “gastric network” spanning somatosensory, motor, visual, and parietal cortices, showing that the phase of the slow wave predicts BOLD fluctuations with a precise temporal sequence within each gastric cycle. Choe et al. (2021) replicated and extended this finding using spatial ICA, identifying at least three phase-locked resting-state networks—cerebellar, dorsal somatosensory-motor, and default mode. Banellis et al. (2025) showed that frontoparietal gastric-brain coupling indexes a dimensional signature of mental health across 243 participants. Most recently, Berther et al. (2026) used high-resolution MEG and high-density EGG to map gastric-brain coupling at rest across multiple frequency bands, revealing broadband phase-amplitude coupling and spatio-spectral fingerprints that overlap with the fMRI-based gastric network.

As the field has grown, methodological work has established foundations for rigorous analysis. Wolpert et al. (2020) provided normative EGG data from a large healthy sample ($N = 117$) and a semi-automated MATLAB pipeline covering electrode placement, quality assessment, and multi-step analysis from raw recording to phase and power extraction. Levakov et al. (2023) characterized the confound sensitivity of gastric-brain coupling, demonstrating that head motion and non-grey-matter signals can inflate the spatial extent of the

gastric network if not controlled. Anisimova et al. (2025) introduced an open-source Python pipeline for standalone EGG with ICA-based denoising and artifact rejection. Yet no existing package covers the full workflow from raw EGG through gastric-brain coupling with fMRI. Researchers typically rely on custom MATLAB scripts (Wolpert et al., 2020; Banellis et al., 2025), ad hoc Python code, or general-purpose toolkits—NeuroKit2 (Makowski et al., 2021) for biosignal processing, MNE-Python (Gramfort et al., 2013) for EEG and MEG—none of which target the gastric slow wave’s low frequency range and long cycle durations. Domain-specific packages such as Systole (Legrand et al., 2022) and HeartPy (van Gent et al., 2019) serve the cardiac community but not gastric physiology.

Here we present GastroPy, an open-source Python package for EGG signal processing and gastric-brain coupling analysis. GastroPy is designed around three principles: (1) domain specificity, with frequency bands, filtering parameters, and artifact criteria tailored to the gastric slow wave; (2) modularity, with a layered architecture that separates core DSP from neuroimaging-specific logic; and (3) interoperability, with all functions operating on standard NumPy arrays and pandas DataFrames. We describe the package’s design philosophy and architecture, demonstrate its capabilities through two worked examples, and discuss its place in the emerging ecosystem of brain-body analysis tools.

2 Design Philosophy and Architecture

2.1 Guiding Principles

GastroPy’s design follows four principles drawn from scientific Python software development, particularly the approaches of NeuroKit2 (Makowski et al., 2021) and MNE-Python (Gramfort et al., 2013):

1. **Domain-specific defaults with full parameter control.** All functions expose sensible defaults for gastric rhythm analysis (e.g., normogastric band of 0.033–0.067 Hz, 200-second PSD windows) while allowing complete parameter customization for advanced users.
2. **Layered dependency isolation.** Core signal processing depends only on NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), pandas (McKinney, 2010), and matplotlib (Hunter, 2007). Neuroimaging features (MNE-Python, nilearn, nibabel) are optional extras, keeping the base install lightweight.
3. **Composable, flat API.** Each function operates on NumPy arrays or pandas DataFrames and can be used independently or composed into pipelines. High-level convenience

functions aggregate lower-level operations but never hide them.

4. **Reproducibility by default.** All random operations (e.g., surrogate generation) accept explicit seeds. Bundled sample data and comprehensive documentation ensure that analyses can be reproduced across labs and platforms.

2.2 Package Architecture

GastroPy is organized into seven modules arranged in two layers (Table 1).

Table 1: GastroPy module organization. Core modules require only NumPy, SciPy, pandas, and matplotlib. The neuroimaging layer requires optional dependencies (MNE-Python, Nilearn, Nibabel), installable via `pip install gastropy[neuro]`.

Layer	Module	Description
Core	<code>gastropy.signal</code>	PSD, filtering, phase, preprocessing, ICA, sine fitting
	<code>gastropy.metrics</code>	Band power, instability, cycle statistics
	<code>gastropy.egg</code>	Pipeline, channel selection, multichannel processing
	<code>gastropy.timefreq</code>	Narrowband decomposition, Morlet wavelets
	<code>gastropy.coupling</code>	PLV, circular statistics, surrogates
	<code>gastropy.viz</code>	Publication-ready plotting
	<code>gastropy.data</code>	Bundled sample datasets
Neuro	<code>gastropy.neuro.fmri</code>	Trigger alignment, BOLD phases, PLV maps

The **core layer** handles all EGG-specific signal processing. `gastropy.signal` provides low-level DSP primitives: power spectral density via Welch’s method, FIR and IIR bandpass filter design and application, Hilbert-based instantaneous phase extraction, signal resampling, cycle edge detection, and phase-based artifact detection. It also includes time-domain preprocessing functions—Hampel spike removal, MAD outlier rejection, and LMMSE movement artifact attenuation (Gharibans et al., 2018)—ported from the electrography pipeline of Anisimova et al. (2025), ICA-based spatial denoising via FastICA (Pedregosa et al., 2011) for multi-channel recordings, and least-squares sine fitting for gastric frequency characterization. `gastropy.metrics` computes gastric rhythm metrics including band power (peak frequency, maximum power, proportional power), the instability coefficient (Koch and Stern, 2004), cycle duration statistics, proportion of normogastric cycles, and automated quality assessment. `gastropy.egg` composes these into high-level workflows: `egg_process` runs a complete filter-phase-metrics pipeline in a single call, while `select_best_channel` identifies the channel with the strongest normogastric rhythm in multi-electrode recordings. `egg_process_multichannel` extends this to multi-electrode arrays with three strategies:

independent per-channel processing, automatic best-channel selection, and ICA-based spatial denoising followed by per-channel analysis. The `egg_clean` function supports named method variants, including "dalmaijer2025" (Anisimova et al., 2025), which chains Hampel spike removal, LMMSE movement filtering, and IIR Butterworth bandpass filtering into a citable, reproducible preprocessing pipeline. `gastropy.timefreq` provides narrow-band decomposition via per-band filtering and Morlet wavelet time-frequency representations. `gastropy.coupling` implements modality-agnostic circular statistics (circular mean, resultant length, Rayleigh test) and phase-locking value computation with surrogate-based z-scoring via circular time-shifting. `gastropy.viz` provides plotting functions for every analysis stage, from PSD plots with gastric band shading to comprehensive multi-panel overview figures.

The **neuroimaging layer** extends the core with fMRI-specific functionality. `gastropy.neuro.fmri` provides scanner trigger detection, volume window creation, per-volume phase extraction, transient volume removal, confound regression via GLM, voxelwise BOLD phase extraction at the gastric frequency, and whole-brain PLV map generation with surrogate-based null distributions. This module implements the methodology described in Banellis et al. (2025).

2.3 Gastric Frequency Bands

GastroPy defines three standard gastric frequency bands as named constants in `gastropy.metrics`:

- **Bradygastria:** 0.02–0.03 Hz (1.2–1.8 cycles per minute)
- **Normogastria:** 0.033–0.067 Hz (2–4 cycles per minute)
- **Tachygastria:** 0.07–0.17 Hz (4.2–10.2 cycles per minute)

These bands follow clinical conventions (Koch and Stern, 2004; Chang, 2005; Yin and Chen, 2013) and serve as defaults throughout the package, configurable via the `GastricBand` named tuple.

3 Key Algorithms and Methods

3.1 Spectral Analysis

GastroPy computes power spectral density using Welch’s method with a default window length of 200 seconds, chosen to provide sufficient frequency resolution in the gastric range (0.005 Hz resolution) while allowing multiple windows for variance reduction in typical

recording durations (15–30 minutes). Users can control the overlap fraction (default 25%) via the `overlap` parameter. The resulting PSD is used for channel selection, peak frequency identification, and band power computation.

Band power metrics are computed by integrating the PSD within each gastric band. For each band, GastroPy reports the peak frequency (Hz), maximum and mean power, proportional power (band power divided by total power in the 0.01–0.2 Hz analysis range), and the mean power ratio (band power relative to out-of-band power). These metrics follow the conventions established in the clinical EGG literature (Koch and Stern, 2004; Chang, 2005).

3.2 Bandpass Filtering

GastroPy supports both FIR and IIR bandpass filtering with zero-phase application via `scipy.signal.filtfilt`. The default FIR filter uses an adaptive tap count computed from the filter order, sampling frequency, and a configurable transition width (default 20% of the passband). IIR filtering uses a Butterworth design with configurable order (default 4). The zero-phase application ensures no phase distortion, which is critical for subsequent Hilbert-based phase extraction.

For the default normogastric band (0.033–0.067 Hz), the filter passband corresponds to cycle durations of 15–30 seconds. The transition width parameter controls the sharpness of the roll-off and can be adjusted to balance between frequency selectivity and temporal ringing artifacts.

3.3 Phase Extraction and Cycle Detection

Instantaneous phase is extracted via the Hilbert transform applied to the bandpass-filtered signal. The analytic signal $z(t) = x(t) + i\hat{x}(t)$, where $\hat{x}(t)$ is the Hilbert transform of $x(t)$, yields the instantaneous phase as $\phi(t) = \arg(z(t))$ and the instantaneous amplitude as $A(t) = |z(t)|$. The phase is wrapped to $[-\pi, \pi)$.

Gastric cycles are detected by identifying phase wrapping events (transitions from $+\pi$ to $-\pi$). The duration of each cycle is computed as the time between consecutive wrapping events. Cycle-level metrics include the mean and standard deviation of cycle durations, the proportion of cycles falling within the normogastric range (15–30 seconds by default), and the instability coefficient (IC), defined as the standard deviation of cycle durations divided by the mean:

$$\text{IC} = \frac{\sigma_{\text{cycle}}}{\mu_{\text{cycle}}} \quad (1)$$

Lower IC values indicate a more regular gastric rhythm. GastroPy also provides automated quality assessment based on configurable thresholds for minimum cycle count, rhythm stability, and normogastric dominance.

3.4 Artifact Detection

GastroPy implements phase-based artifact detection following the approach of Wolpert et al. (2020). Two types of artifacts are identified at the cycle level: (1) *non-monotonic phase* cycles, where the instantaneous phase does not progress monotonically within a cycle (indicating loss of a stable oscillation), and (2) *duration outlier* cycles, where the cycle duration exceeds a configurable threshold (default: 3 standard deviations from the mean). Artifact cycles are flagged with a boolean mask that can be used to exclude contaminated data from subsequent analyses.

3.5 Time-Domain Preprocessing

GastroPy provides three composable preprocessing functions that operate on raw EGG signals before bandpass filtering, ported from the electrography pipeline of Anisimova et al. (2025):

1. **Hampel filter:** A sliding-window median filter that replaces samples exceeding a configurable MAD threshold (default: 3) with the local median. This removes isolated spikes while preserving the underlying slow wave.
2. **MAD filter:** A global median absolute deviation filter that replaces samples exceeding a configurable threshold (default: 3 MAD) with the signal median. This addresses gross outliers spanning multiple samples.
3. **LMMSE movement filter:** A linear minimum mean-square error (Wiener) filter that attenuates movement artifacts by estimating and subtracting correlated noise in the frequency domain (Gharibans et al., 2018). The filter operates by computing a transfer function from the signal’s power spectral density and a noise estimate, preserving gastric-band energy while suppressing broadband movement contamination.

All preprocessing functions accept both 1D (single channel) and 2D (`n_channels`, `n_samples`) arrays, enabling batch processing of multi-electrode recordings. These functions are composed into the named method variant `egg_clean(method="dalmaijer2025")`, which applies Hampel filtering, LMMSE movement removal, and IIR bandpass filtering in sequence.

3.6 ICA-Based Spatial Denoising

For multi-channel EGG recordings, GastroPy provides ICA-based spatial denoising via FastICA decomposition (Pedregosa et al., 2011). The algorithm decomposes the multi-channel signal into independent components, computes the signal-to-noise ratio (SNR) of each component within the gastric frequency band, retains only components whose gastric SNR exceeds a configurable threshold, and reconstructs the cleaned signal in channel space. This approach recomposes the electrogastragram from all recorded channels after automatic rejection of nuisance components, reducing the influence of researcher bias inherent in manual single-channel selection (Anisimova et al., 2025). ICA denoising is available both as a standalone function (`gastropy.signal.ica_denoise`) and as one of the three strategies in `egg_process_multichannel`.

3.7 Phase-Locking Value

The phase-locking value (PLV) quantifies the consistency of the phase difference between two signals across time (Rebollo et al., 2018). For two phase time series $\phi_a(t)$ and $\phi_b(t)$, the PLV is defined as:

$$\text{PLV} = \left| \frac{1}{N} \sum_{t=1}^N e^{i(\phi_a(t) - \phi_b(t))} \right| \quad (2)$$

where N is the number of time points. PLV ranges from 0 (no phase consistency) to 1 (perfect phase-locking). GastroPy also computes the complex PLV, which preserves the preferred phase lag direction.

Statistical significance of observed PLV values is assessed using a surrogate-based approach. Surrogate distributions are generated by circularly shifting one phase time series by random offsets (drawn uniformly from the full signal length, with a configurable buffer to avoid trivially similar shifts). The empirical PLV is then z-scored against the surrogate distribution:

$$z = \frac{\text{PLV}_{\text{emp}} - \text{median}(\text{PLV}_{\text{surr}})}{\text{MAD}(\text{PLV}_{\text{surr}})} \quad (3)$$

where MAD is the median absolute deviation. This non-parametric approach avoids distributional assumptions and is robust to non-stationarity in the signals.

3.8 Voxelwise BOLD Phase Extraction

For fMRI coupling analysis, GastroPy extracts the instantaneous phase of BOLD signal fluctuations at the gastric frequency for each brain voxel. The BOLD time series (after confound regression and z-scoring) is bandpass-filtered at the participant’s peak gastric fre-

quency using either an IIR Butterworth filter (default) or a FIR filter, with a configurable half-width at half-maximum (HWHM, default 0.015 Hz) defining the filter bandwidth. The Hilbert transform is then applied to extract the instantaneous phase. Transient volumes at the beginning and end of the filtered time series are removed (default: 21 volumes each) to avoid edge artifacts from the filtering and Hilbert transform operations.

4 Installation and Dependencies

GastroPy is available on the Python Package Index (PyPI) and can be installed with pip:

```
pip install gastropy
```

This installs the core package with its minimal dependencies: NumPy, SciPy, pandas, and matplotlib. For neuroimaging workflows that require MNE-Python, nilearn, and nibabel:

```
pip install gastropy[neuro]
```

For development, including testing and documentation tools:

```
pip install gastropy[dev]
```

GastroPy requires Python 3.10 or later and is tested on Linux, macOS, and Windows via continuous integration. The package uses Hatch as its build system, Ruff for linting and formatting, and Sphinx with the sphinx-book-theme for documentation, which is hosted at <https://embodied-computation-group.github.io/gastropy>.

5 Example 1: Standalone EGG Processing

This example demonstrates a complete EGG analysis workflow using a standalone recording from the Wolpert et al. (2020) normative dataset, bundled with GastroPy.

5.1 Loading Data

GastroPy includes several sample datasets accessible via the `gastropy.data` module. The standalone EGG recording contains 7 channels sampled at 10 Hz from a healthy participant:

```
import gastropy as gp
```

```

# Load standalone EGG recording (Wolpert et al., 2020)
rec = gp.load_egg()
signal = rec["signal"]    # shape: (n_channels, n_samples)
sfreq = rec["sfreq"]      # 10.0 Hz

print(f"Channels: {rec['ch_names']}")
print(f"Duration: {rec['duration_s']:.0f} s")

```

5.2 Channel Selection

For multi-channel recordings, GastroPy identifies the channel with the strongest normogastric peak using spectral analysis:

```

best_idx, peak_freq, freqs, psd = gp.select_best_channel(
    signal, sfreq
)
print(f"Best channel: {rec['ch_names'][best_idx]}")
print(f"Peak frequency: {peak_freq:.4f} Hz "
      f"({peak_freq * 60:.1f} cpm)")

```

The channel selection algorithm computes the PSD for each channel using Welch's method, identifies spectral peaks within the normogastric band, and selects the channel whose peak has the highest power. The PSD can be visualized with gastric band shading:

```

fig, ax = gp.plot_psd(
    freqs, psd,
    band=gp.NORMOGASTRIA,
    ch_names=rec["ch_names"],
    best_idx=best_idx,
    peak_freq=peak_freq,
)

```

5.3 Signal Processing Pipeline

The `egg_process` function runs the full analysis pipeline on the selected channel: band-pass filtering in the normogastric band, Hilbert-based instantaneous phase and amplitude extraction, cycle detection, and metric computation:

```

signals_df, info = gp.egg_process(
    signal[best_idx], sfreq
)

# signals_df columns: raw, filtered, phase, amplitude
# info dict contains metrics:
print(f"Cycles detected: {info['n_cycles']}")
print(f"Mean cycle duration: "
      f"{info['mean_cycle_dur_s']:.1f} s")
print(f"Instability coefficient: "
      f"{info['instability_coefficient']:.3f}")
print(f"Proportion normogastric: "
      f"{info['proportion_normogastric']:.1%}")

```

The returned `signals_df` DataFrame contains aligned time series of the raw signal, bandpass-filtered signal, instantaneous phase (in radians), and analytic amplitude. The `info` dictionary contains all computed metrics, including the instability coefficient (Koch and Stern, 2004), cycle duration statistics, band power, and an automated quality assessment flag.

5.4 Artifact Detection

Phase-based artifact detection, following the approach of Wolpert et al. (2020), identifies cycles with non-monotonic phase progression or anomalous durations:

```

times = signals_df.index.values / sfreq
artifact_info = gp.detect_phase_artifacts(
    signals_df["phase"].values, times
)

print(f"Artifact cycles: "
      f"{artifact_info['n_artifact_cycles']}")

fig, ax = gp.plot_artifacts(
    signals_df["phase"].values,
    times,
    artifact_info,
)

```

5.5 Visualization

GastroPy provides several plotting functions for visualizing results. The `plot_egg_overview` function generates a four-panel figure showing the raw signal, filtered signal, instantaneous phase, and amplitude envelope:

```
fig, axes = gp.plot_egg_overview(signals_df, sfreq)
```

Cycle duration distributions can be visualized as histograms with the normogastric range highlighted:

```
fig, ax = gp.plot_cycle_histogram(  
    info["cycle_durations_s"]  
)
```

5.6 Time-Frequency Analysis

For more detailed spectral dynamics, the `multiband_analysis` function decomposes the signal across all three gastric bands simultaneously:

```
bands = gp.multiband_analysis(signal[best_idx], sfreq)  
  
for name, result in bands.items():  
    bp = result["band_power"]  
    print(f"{name}: peak {bp['peak_freq_hz']:.4f} Hz, "  
          f"power {bp['prop_power']:.1%}")
```

6 Example 2: Gastric-Brain Coupling with fMRI

This example demonstrates the gastric-brain coupling pipeline using concurrent EGG-fMRI data, implementing the methodology described in Banellis et al. (2025). The workflow proceeds from EGG phase extraction through voxelwise BOLD phase computation to whole-brain PLV mapping.

6.1 Loading Concurrent EGG-fMRI Data

GastroPy bundles three sessions of concurrent EGG-fMRI recordings, each containing 8-channel EGG at 10 Hz with scanner trigger timing:

```
import gastropy as gp
from gastropy.neuro import fmri
import numpy as np

# Load EGG recording with trigger info
rec = gp.load_fmri_egg(session="0001")
signal = rec["signal"]    # (8, n_samples)
sfreq = rec["sfreq"]      # 10.0 Hz
tr = rec["tr"]            # 1.856 s

# Select best channel and get peak frequency
best_idx, peak_freq, freqs, psd = (
    gp.select_best_channel(signal, sfreq)
)
```

6.2 EGG Phase Extraction

The EGG signal is bandpass-filtered and the instantaneous phase is extracted via the Hilbert transform:

```
# Process EGG on best channel
signals_df, info = gp.egg_process(
    signal[best_idx], sfreq
)
egg_phase = signals_df["phase"].values
```

6.3 Volume Windowing and Phase-per-Volume

Scanner trigger times are used to create volume windows that map each fMRI volume to the corresponding EGG time points. The mean EGG phase within each volume window is then computed:

```
# Create volume windows from trigger onsets
```

```

trigger_times = rec["trigger_times"]
n_volumes = len(trigger_times)
windows = fmri.create_volume_windows(
    trigger_times, tr, n_volumes
)

# Extract analytic signal for phase computation
_, analytic = gp.instantaneous_phase(
    gp.apply_bandpass(signal[best_idx], sfreq,
                      gp.NORMOGASTRIA.low,
                      gp.NORMOGASTRIA.high)
)

# Compute mean phase per fMRI volume
phase_per_vol = fmri.phase_per_volume(
    analytic, windows
)

```

6.4 BOLD Preprocessing and Phase Extraction

Preprocessed BOLD data (e.g., from fMRIPrep) is loaded, confound signals are regressed out, and the instantaneous phase at the gastric frequency is extracted for each voxel:

```

# Load preprocessed BOLD data
bold_data = gp.fetch_fmri_bold(session="0001")
bold_2d = bold_data["bold"] # (n_voxels, n_vols)
confounds = bold_data["confounds"] # DataFrame

# Regress out motion and physiological confounds
bold_clean = fmri.regress_confounds(bold_2d, confounds)

# Apply volume cuts to remove edge transients
begin_cut, end_cut = 21, 21
bold_cut = fmri.apply_volume_cuts(
    bold_clean, begin_cut, end_cut
)
egg_cut = fmri.apply_volume_cuts(
    phase_per_vol.reshape(1, -1),

```

```

    begin_cut, end_cut
).ravel()

# Extract BOLD phase at gastric frequency
bold_sfreq = 1.0 / tr
bold_phases = fmri.bold_voxelwise_phases(
    bold_cut, peak_freq, bold_sfreq
)

```

6.5 PLV Map Computation

The phase-locking value between the EGG phase and each voxel's BOLD phase is computed to generate a whole-brain PLV map. Statistical significance is assessed via comparison to a surrogate distribution created by circular time-shifting:

```

# Compute empirical PLV map
plv_map = fmri.compute_plv_map(egg_cut, bold_phases)

# Compute surrogate PLV for statistical testing
surr_plv = fmri.compute_surrogate_plv_map(
    egg_cut, bold_phases,
    n_surrogates=1000, seed=42
)

# Z-score the empirical PLV against surrogates
z_map = gp.coupling_zscore(plv_map, surr_plv)
print(f"Max PLV: {plv_map.max():.3f}")
print(f"Max z-score: {z_map.max():.2f}")

```

The resulting PLV and z-score maps can be reshaped to 3D volumes and saved as NIfTI images for visualization in standard neuroimaging viewers, or plotted directly using GastroPy's Nilearn-based visualization functions.

6.6 Coupling Statistics

GastroPy's coupling module also provides functions for testing the significance of individual PLV values using the Rayleigh test for circular uniformity:

```

# Test whether EGG-BOLD coupling is significant
# for a region of interest
roi_phase = bold_phases[roi_mask].mean(axis=0)
phase_diff = egg_cut - roi_phase

z_stat, p_value = gp.rayleigh_test(phase_diff)
print(f"Rayleigh z = {z_stat:.2f}, p = {p_value:.4f}")

```

7 Relationship to Existing Tools

GastroPy is, to our knowledge, the first dedicated Python package for EGG signal processing and gastric-brain coupling analysis. Existing tools address adjacent needs: NeuroKit2 (Makowski et al., 2021) provides general biosignal processing with basic EGG support, MNE-Python (Gramfort et al., 2013) is the standard for EEG and MEG analysis, and domain-specific packages such as Systole (Legrand et al., 2022) and HeartPy (van Gent et al., 2019) serve the cardiac community. The electrography tutorial of Anisimova et al. (2025) provides a standalone EGG processing pipeline with ICA-based denoising, elements of which are integrated into GastroPy as the "dalmaijer2025" named method variant. GastroPy complements these tools by providing a pipeline from raw EGG to gastric-brain coupling maps, with domain-specific defaults, multi-channel processing strategies, and a layered architecture separating core signal processing from neuroimaging workflows.

The gastric slow wave occupies a unique frequency range (0.03–0.07 Hz) that falls below the default settings of most electrophysiology toolkits, and its coupling with brain signals requires fMRI-specific preprocessing (trigger alignment, confound regression) outside the scope of general biosignal packages. GastroPy’s standard array-based API ensures interoperability with the broader scientific Python ecosystem, and its modular design allows researchers to use individual functions as building blocks within custom pipelines.

8 Testing and Quality Assurance

GastroPy maintains an automated test suite of 297 tests covering all public functions across every module. Tests verify signal processing correctness (e.g., that bandpass filtering preserves in-band energy while attenuating out-of-band frequencies), metric computation accuracy (e.g., that the instability coefficient increases for irregular rhythms), coupling statistics (e.g., that PLV equals 1.0 for identical phase series and approaches 0 for uncorrelated se-

ries), and visualization outputs (e.g., that plotting functions return valid matplotlib figures without errors).

The test suite runs automatically via GitHub Actions on every push and pull request, with separate workflows for testing (pytest), linting (Ruff check and format verification), and documentation building (Sphinx). Tests are executed on the latest Ubuntu runner with Python 3.13. All sample data used in tests is bundled with the package, ensuring that tests are self-contained and reproducible without network access.

9 Discussion

GastroPy is the first dedicated, open-source Python package for electrogastrography signal processing and gastric-brain coupling analysis. By consolidating methods previously scattered across lab-specific MATLAB scripts and ad hoc Python code into a tested, documented, and pip-installable package, it aims to lower the barrier to rigorous EGG analysis and accelerate research on the brain-gut axis.

9.1 Current Capabilities and Use Cases

The package currently supports the full standalone EGG processing workflow (spectral analysis, time-domain preprocessing, ICA-based spatial denoising, bandpass filtering, phase extraction, cycle metrics, artifact detection, quality assessment) and the complete EGG-fMRI coupling pipeline (trigger alignment, BOLD preprocessing, voxelwise phase extraction, PLV mapping with surrogate testing). Named method variants (e.g., "dalmaijer2025") provide citable, reproducible preprocessing pipelines that chain specific combinations of artifact removal and filtering steps. The multichannel processing module offers three strategies for multi-electrode recordings, including ICA-based denoising that reduces researcher bias inherent in manual channel selection (Anisimova et al., 2025). The modular architecture allows each function to be used independently, so researchers can build custom pipelines tailored to their experimental designs while relying on tested implementations of standard analysis steps.

GastroPy serves several overlapping user communities. Neuroscientists studying brain-body interactions can use the EGG-fMRI coupling pipeline to generate whole-brain PLV maps. Psychophysiologicals conducting standalone EGG studies can use the core processing modules for spectral analysis, rhythm characterization, and artifact detection without neuroimaging dependencies. Clinical researchers investigating motility disorders can use the automated quality assessment and multi-band decomposition to characterize dysrhythmias.

The composable function-level API also supports methods developers who need individual operations (e.g., phase extraction, PLV computation, surrogate generation) as building blocks within custom pipelines.

Bundled sample data (three fMRI-EGG sessions from a concurrent recording study and one standalone EGG recording from the Wolpert et al. (2020) normative dataset) provides immediate access to realistic data for tutorials, testing, and benchmarking without requiring researchers to acquire or share their own recordings. Larger preprocessed fMRI BOLD datasets are available via the `fetch_fmri_bold` function, which downloads data from GitHub Releases using the Pooch library for reliable, cached retrieval.

9.2 Reproducibility Considerations

A core motivation for GastroPy is improving reproducibility in EGG research. Many labs use custom analysis scripts that are not publicly available, contain undocumented parameter choices, and are difficult to verify or replicate. A tested, documented, and version-controlled implementation of standard methods enables researchers to report their analysis parameters precisely (e.g., by referencing specific function names and parameter values) and to share executable analysis pipelines alongside their publications.

All stochastic operations in GastroPy (surrogate generation for PLV testing) accept explicit random seeds, ensuring bitwise reproducibility. The package pins minimum dependency versions and is tested on multiple platforms via GitHub Actions. The 297-test suite provides regression protection against inadvertent changes to algorithm behavior across releases.

9.3 Limitations and Future Directions

The current release has several limitations. The coupling module supports only the PLV metric; future releases will add amplitude-phase coupling, coherence, and information-theoretic metrics, along with group-level statistical testing (e.g., permutation-based cluster correction for PLV maps). The neuroimaging layer supports only fMRI; planned extensions include EEG-EGG and MEG-EGG coupling via `gastropy.neuro.eeg` and `gastropy.neuro.meg`, using MNE-Python’s infrastructure for high-frequency electrophysiology data. The package does not yet include fully BIDS-compatible data loading, though preliminary I/O support for BrainVision, EDF, and CSV formats is available in `gastropy.io`. Finally, while the current release includes ICA-based spatial denoising and three composable time-domain preprocessing filters, artifact detection operates at the cycle level; future work may incorporate continuous artifact scoring and automated channel interpolation.

We also plan to expand the sample data with recordings from additional experimental conditions (fasting vs. postprandial, clinical populations, pharmacological challenges), develop Jupyter notebook tutorials for common workflows, and establish cross-validation benchmarks against existing MATLAB implementations. Integration with Systole (Legrand et al., 2022) for cardiac-gastric interaction analysis is another planned direction.

9.4 Conclusion

GastroPy fills a gap in the scientific Python ecosystem: no dedicated toolkit for electro-gastrography signal processing and gastric-brain coupling existed in Python until now. Its modular architecture, domain-specific defaults, test suite, and bundled sample data make it accessible to both newcomers to EGG analysis and experienced researchers transitioning from MATLAB-based workflows. By consolidating validated implementations into a single, documented package, GastroPy aims to reduce the methodological variability that limits cross-study comparisons and to support the adoption of best practices in gastric-brain coupling research. As interest in the brain-gut axis grows across neuroscience, psychiatry, gastroenterology, and human-computer interaction, we hope GastroPy will serve as a foundation for reproducible, collaborative research on gastric-brain interactions.

AI Usage Disclosure

Generative AI tools were used during the development of GastroPy. Specifically, Anthropic Claude (Claude Code CLI, models `claude-sonnet-4-6` and `claude-opus-4-6`) was used for code generation, test writing, documentation drafting, and debugging across all modules. All AI-generated code was reviewed, tested, and validated by the author. Architectural decisions, algorithm selection, and scientific methodology were determined by the author. The automated test suite verifies correctness of all implementations. The author accepts full responsibility for the accuracy, originality, and licensing of all code and documentation.

Acknowledgements

We acknowledge support from the Lundbeck Foundation (R380-2021-1538, R140-2013-13057), the European Research Council (ERC-2020-COG, 101001893, EMBODIED-COMPUTATION), and Aarhus University.

References

- Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., and Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8:14.
- Anisimova, E., Alladin, S. N. B., Tsamaz, S., and Dalmaijer, E. S. (2025). A tutorial on electrogastrography using low-cost hardware and open-source software. arXiv preprint arXiv:2509.17260.
- Banellis, L., Rebollo, I., Nikolova, N., and Allen, M. (2025). Stomach–brain coupling indexes a dimensional signature of mental health. *Nature Mental Health*, 3(8):899–908.
- Berther, T., Saltafossi, M., Fehring, J., Rebollo, I., and Kluger, D. S. (2026). Spatiospectral signatures of stomach-brain synchrony. bioRxiv preprint.
- Chang, F.-Y. (2005). Electrogastrography: Basic knowledge, recording, processing and its clinical applications. *Journal of Gastroenterology and Hepatology*, 20(4):502–516.
- Chen, J. and McCallum, R. W. (1991). Electrogastrography: Measurement, analysis and prospective applications. *Medical and Biological Engineering and Computing*, 29(4):339–350.
- Choe, A. S., Tang, B., Smith, K. R., Honari, H., Lindquist, M. A., Caffo, B. S., and Pekar, J. J. (2021). Phase-locking of resting-state brain networks with the gastric basal electrical rhythm. *PLOS ONE*, 16(1):e0244756.
- Gharibans, A. A., Smarr, B., Kunkel, D. C., Kriegsfeld, L. J., Mousa, H., and Coleman, T. P. (2018). Artifact rejection methodology enables continuous, noninvasive measurement of gastric myoelectric activity in ambulatory subjects. *Scientific Reports*, 8:5019.
- Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., and Hämäläinen, M. S. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7:267.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Krevelen, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- Koch, K. L. and Stern, R. M. (2004). *Handbook of Electrogastrography*. Oxford University Press.
- Legrand, N., Engen, H., Correa, C., Mathiasen, N. K., Nikolova, N., Fardo, F., and Allen, M. (2022). Systole: A Python package for cardiac signal synchrony and analysis. *Journal of Open Source Software*, 7(69):3832.
- Levakov, G., Ganor, S., and Avidan, G. (2023). Reliability and validity of brain-gastric phase synchronization. *Human Brain Mapping*, 44(14):4956–4966.
- Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., and Chen, S. H. A. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods*, 53(4):1689–1696.
- McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Rebollo, I., Devauchelle, A.-D., Béranger, B., and Tallon-Baudry, C. (2018). Stomach-brain synchrony reveals a novel, delayed-connectivity resting-state network in humans. *eLife*, 7:e33321.
- van Gent, P., Farah, H., van Nes, N., and van Arem, B. (2019). HeartPy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation Research Part F: Traffic Psychology and Behaviour*, 66:368–378.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

- Vujic, A., Tong, S., Picard, R., and Maes, P. (2020). Going with our guts: Potentials of wearable electrogastrography (EGG) for affect detection. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 260–268. Association for Computing Machinery.
- Wolpert, N., Rebollo, I., and Tallon-Baudry, C. (2020). Electrogastrography for psychophysiological research: Practical considerations, analysis pipeline, and normative data in a large sample. *Psychophysiology*, 57(9):e13599.
- Yin, J. and Chen, J. D. Z. (2013). Electrogastrography: Methodology, validation and applications. *Journal of Neurogastroenterology and Motility*, 19(1):5–17.