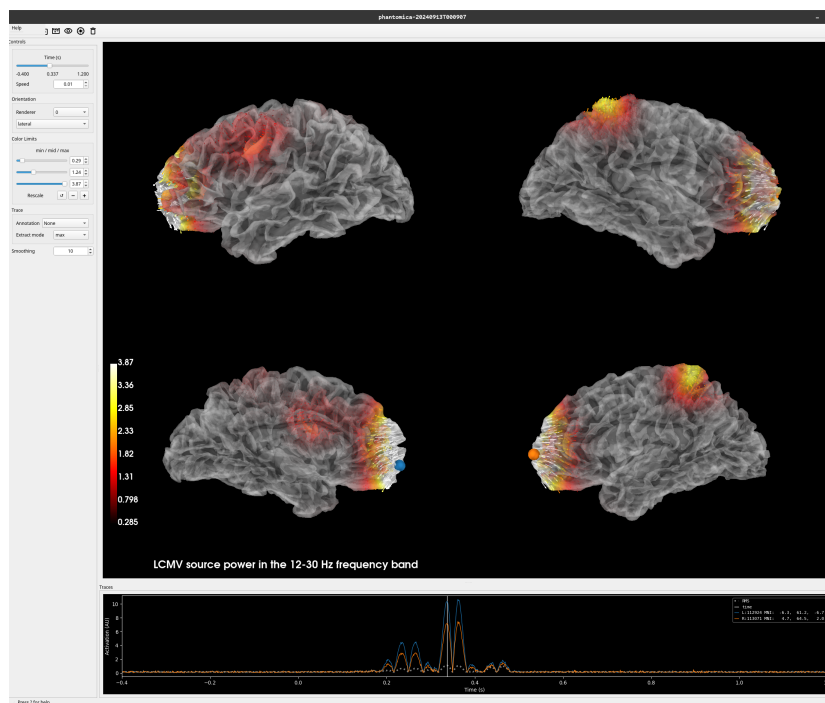
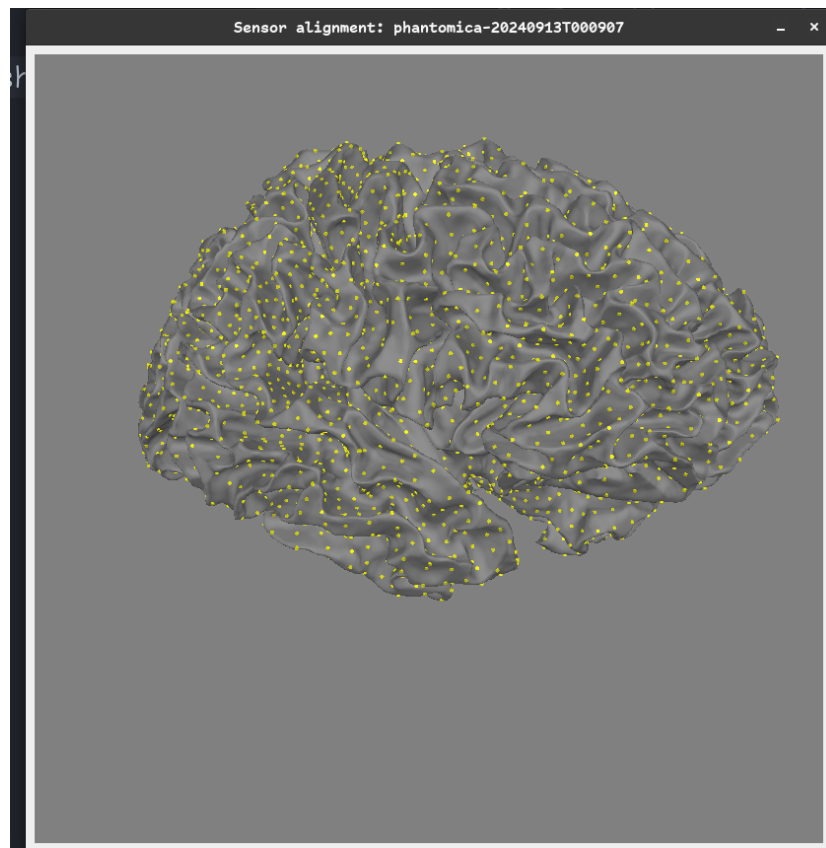
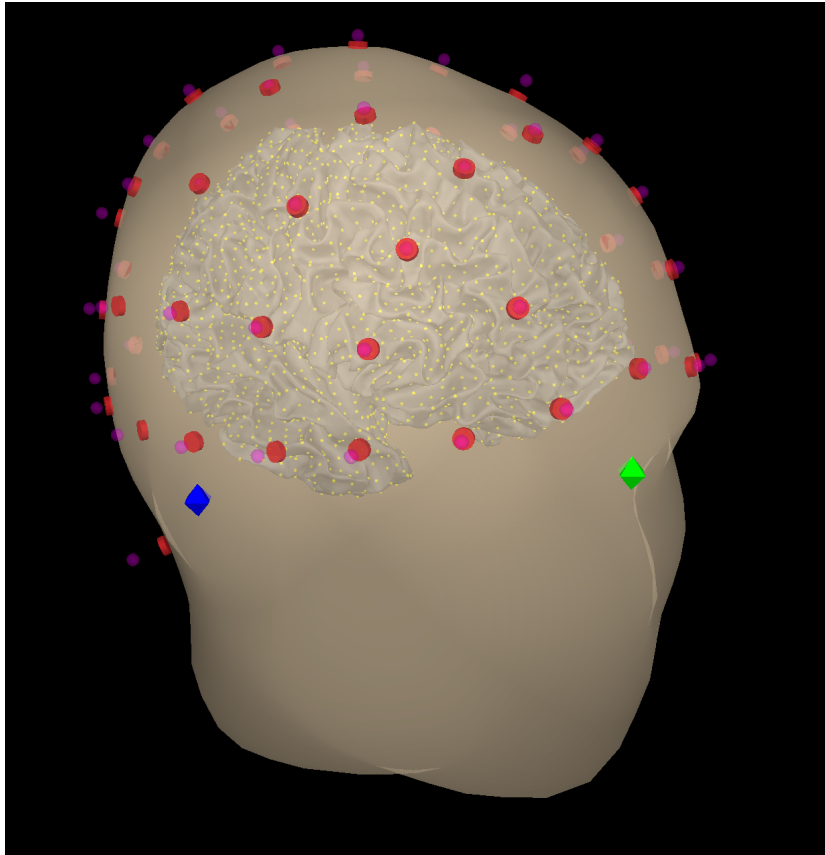


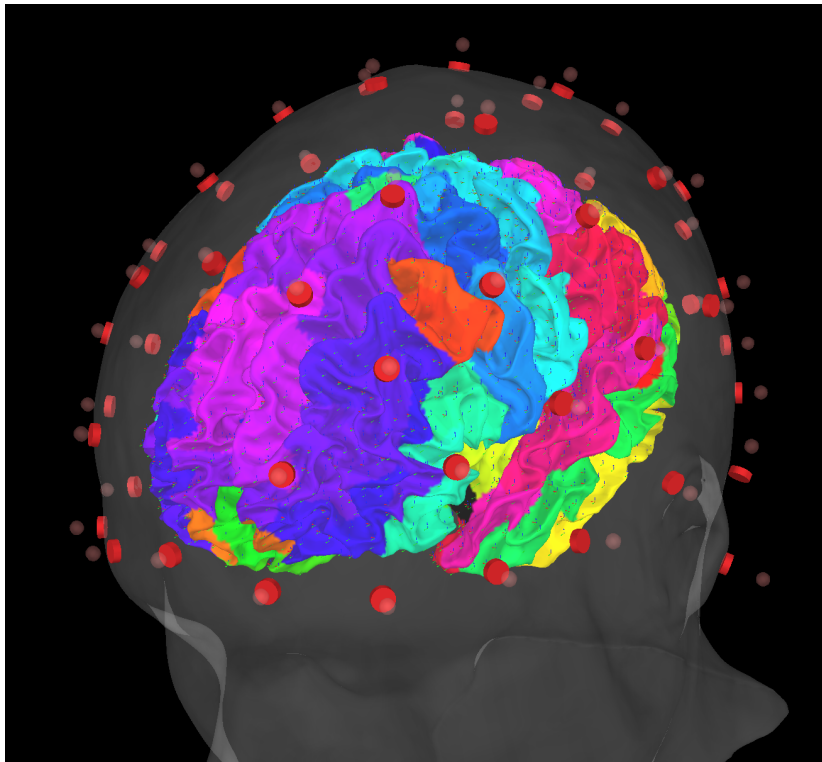
i6001 Gte Package Devel

1 Screenshots









2 Init

2.1 Development

```
import os
# os.environ['QT_NO_GLIB'] = '1'
# os.environ['SESSION_MANAGER'] = ''

devel_mode = False
devel_mode = True
if devel_mode:
    get_ipython().run_line_magic("load_ext", "autoreload")
    get_ipython().run_line_magic("autoreload", "2")
```

The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload

2.2 Imports

```
import mne
import gte
import pathlib
```

```

import numpy as np
import srsly

import matplotlib
from matplotlib import pyplot as plt

from gte import GTE
from gte import Log0
from gte import GTE # noqa: F401
from gte import ddf # noqa: F401
from gte import rdf # noqa: F401

```

2.3 Init

```

get_ipython().run_line_magic("matplotlib", "qt")

logZ = Log0()
log0 = logZ.logger

gte = GTE()
log0.warning(f"{gte = }")
None

```

W: gte = GTE(_subjects_dir=None, _subject=None, _info0=None, _montage0=None, _trans0=None,

2.4 Init: gte.subjects_dir and gte.subject

```

gte.subjects_dir = pathlib.Path().home()/"mount/data/subjects/"
log0.info(f"{gte.subjects_dir = }")

gte.subject = "phantomica-20240913T000907"
log0.info(f"{gte.subject = }")

```

I: gte.subjects_dir = PosixPath('~/.mount/data/subjects')

I: gte.subject = 'phantomica-20240913T000907'

2.5 Init: gte.info0

```

gte.info0 = gte.subjects_dir/gte.subject/f"aux/mne/{gte.subject}-basic-info.fif"
gte.info0 = f"{gte.subject}-basic-info.fif"
log0.info(f"{gte.info0 = }")

```

I: gte.info0 = <Info | 10 non-empty values

 bads: []

 ch_names: Fp1, Fpz, Fp2, AF7, AF3, AF4, AF8, F7, F5, F3, F1, Fz, F2, F4, ...

```

chs: 61 EEG
custom_ref_applied: False
dig: 64 items (3 Cardinal, 61 EEG)
file_id: 4 items (dict)
highpass: 0.0 Hz
lowpass: 500.0 Hz
meas_date: unspecified
meas_id: 4 items (dict)
nchan: 61
projs: []
sfreq: 1000.0 Hz
>

```

2.6 Init: gte.trans0

```

gte.trans0 = f"{gte.subject}-head-mri-trans.fif"
log0.info(f"{gte.trans0 = }")

```

```

I: gte.trans0 = <Transform | head->MRI (surface RAS)>
[[ 0.99917817 -0.0083337  0.03966784  0.00059106]
 [ 0.01419157  0.98861903 -0.14976975  0.00896041]
 [-0.03796825  0.15020962  0.98792481  0.01880399]
 [ 0.          0.          0.          1.          ]]

```

2.7 Init: gte.montage0

```

gte.montage0 = f"{gte.subject}-basic-montage.fif"
log0.info(f"{gte.montage0 = }")

```

```

I: gte.montage0 = <DigMontage | 0 extras (headshape), 0 HPIs, 3 fiducials, 61 channels>

```

2.8 Init: gte.src0

```

""" OPTIONS:
gte.src0 = f"{gte.subject}-src-ico5.fif"
gte.src0 = f"{gte.subject}-src-oct6.fif"
"""
gte.src0 = f"{gte.subject}-src-oct6.fif"
log0.info(f"{gte.src0 = }")

```

```

W: The file '~/mount/data/subjects/phantomica-20240913T000907/aux/mne/phantomica-20240913T
Reading a source space...
Computing patch statistics...
Patch information added...
Distance information added...
[done]

```

```

    Reading a source space...
    Computing patch statistics...
    Patch information added...
~/cc/dev/2024/20240906T200711-gte-python-package/gte/gte/gte.py:433: RuntimeWarning: This
    self._src0 = mne.read_source_spaces(fif_path)
    Distance information added...
    [done]
    2 source spaces read
I: gte.src0 = <SourceSpaces: [<surface (lh), n_vertices=138145, n_used=4098>, <surface (rh

```

2.9 Init: gte.bem_model0

```

""" OPTIONS:
gte.bem_model0 = f"{gte.subject}-bem-model-ico3.fif"
gte.bem_model0 = f"{gte.subject}-bem-model-ico4.fif"
gte.bem_model0 = f"{gte.subject}-bem-model-ico5.fif"
"""

gte.bem_model0 = f"{gte.subject}-bem-model-ico4.fif"
log0.info(f"{type(gte.bem_model0) = }")
log0.info(f"{len(gte.bem_model0) = }")
log0.info(f"{type(gte.bem_model0[0]) = }")
log0.info(f"{len(gte.bem_model0[0]) = }")

```

```

W: The file '~/mount/data/subjects/phantomica-20240913T000907/aux/mne/phantomica-20240913T
    3 BEM surfaces found
    Reading a surface...
[done]
    Reading a surface...
[done]
    Reading a surface...
[done]
    3 BEM surfaces read
I: type(gte.bem_model0) = <class 'list'>
I: len(gte.bem_model0) = 3
I: type(gte.bem_model0[0]) = <class 'dict'>
I: len(gte.bem_model0[0]) = 8

```

2.10 Init: gte.bem_solution0

```

""" OPTIONS:
gte.bem_solution0 = f"{gte.subject}-bem-solution-ico3.fif"
gte.bem_solution0 = f"{gte.subject}-bem-solution-ico4.fif"
gte.bem_solution0 = f"{gte.subject}-bem-solution-ico5.h5"
"""

gte.bem_solution0 = f"{gte.subject}-bem-solution-ico4.fif"
log0.info(f"{gte.bem_solution0 = }")

```

```

W: The file '~/mount/data/subjects/phantomica-20240913T000907/aux/mne/phantomica-20240913T

```

Loading surfaces...

Loading the solution matrix...

Three-layer model surfaces loaded.

Loaded linear collocation BEM solution from ~/mount/data/subjects/phantomica-20240913T0009

I: gte.bem_solution0 = <ConductorModel | BEM (3 layers) solver=mne>

2.11 Init: gte.fwd0

```
""" OPTIONS:
gte.fwd0 = f"{gte.subject}-fwd-src-ico5-bem-solution-ico3.fif"
gte.fwd0 = f"{gte.subject}-fwd-src-ico5-bem-solution-ico4.fif"
gte.fwd0 = f"{gte.subject}-fwd-src-ico5-bem-solution-ico5.fif"
gte.fwd0 = f"{gte.subject}-fwd-src-oct6-bem-solution-ico3.fif"
gte.fwd0 = f"{gte.subject}-fwd-src-oct6-bem-solution-ico4.fif"
gte.fwd0 = f"{gte.subject}-fwd-src-oct6-bem-solution-ico5.fif"
"""

gte.fwd0 = f"{gte.subject}-fwd-src-oct6-bem-solution-ico4.fif"
log0.info(f"{gte.fwd0 = }")
```

W: The file '~/mount/data/subjects/phantomica-20240913T000907/aux/mne/phantomica-20240913T

Reading forward solution from ~/mount/data/subjects/phantomica-20240913T000907/aux/mne/pha

Reading a source space...

Computing patch statistics...

Patch information added...

Distance information added...

[done]

Reading a source space...

Computing patch statistics...

Patch information added...

~/cc/dev/2024/20240906T200711-gte-python-package/gte/gte/gte.py:538: RuntimeWarning: This

self._fwd0 = mne.read_forward_solution(fif_path)

Distance information added...

[done]

2 source spaces read

Desired named matrix (kind = 3523 (FIFF_MNE_FORWARD_SOLUTION_GRAD)) not available

Read EEG forward solution (8196 sources, 61 channels, free orientations)

Source spaces transformed to the forward solution coordinate frame

I: gte.fwd0 = <Forward | MEG channels: 0 | EEG channels: 61 | Source space: Surface with 8

2.12 Init: gte.genuine_noise_cov0

```
gte.genuine_noise_cov0 = f"{gte.subject}-real-noise-cov.fif"

log0.info(f"{gte.genuine_noise_cov0 = }")
gte.genuine_noise_cov0.plot(gte.info0)
```

61 x 61 full covariance (kind = 1) found.

```

I: Real noise covariance matrix loaded from ~/mount/data/subjects/phantomica-20240913T0009
I: gte.genuine_noise_cov0 = <Covariance | kind : full, shape : (61, 61), range : [-3.1e-10
Computing rank from covariance with rank=None
    Using tolerance 1.2e-12 (2.2e-16 eps * 61 dim * 89 max singular value)
    Estimated rank (eeg): 61
    EEG: rank 61 computed from 61 data channels with 0 projectors

```

<Figure size 380x370 with 2 Axes> <Figure size 380x370 with 1 Axes>

2.13 Init: gte.annot0

```

""" OPTIONS:
gte.annot0 = "aparc"
gte.annot0 = "HCPMP1"
gte.annot0 = "aparc_sub"
gte.annot0 = "aparc.a2009s"
gte.annot0 = "aparc_sub_fix"
"""
gte.annot0 = "aparc_sub_fix"
log0.info(f"{gte.annot0 = }")

```

```

I: gte.annot0 = 'aparc_sub_fix'

```

2.14 Init: gte.labels0

```

gte.read_labels_from_annot()
log0.warning(f"{len(gte.labels0) = }")
log0.warning(f"{gte.labels0 = }")
log0.warning(f"{gte.label0_names = }")

```

```

I: Successfully acquired 450 labels from annotation aparc_sub_fix
W: len(gte.labels0) = 450
W: gte.labels0 = [<Label | phantomica-20240913T000907, 'bankssts_1-lh', lh : 352 vertices>
W: gte.label0_names = ['bankssts_1-lh', 'bankssts_2-lh', 'bankssts_3-lh', 'caudalanteriorc

```

2.15 Init: gte.labels2

```

verbose = True
verbose = False
gte.process_labels0(verbose=verbose)

log0.warning(f"{gte.labels2 = }")
log0.warning(f"{gte.label2_names = }")

```

```

I: Successfully processed 450 labels with location 'center' and extent 0.0
W: gte.labels2 = [<Label | phantomica-20240913T000907, 'bankssts_1-lh', lh : 1 vertices>,
W: gte.label2_names = ['bankssts_1-lh', 'bankssts_2-lh', 'bankssts_3-lh', 'caudalanteriorc

```

2.16 Check WaveForm Generation

```
rng = np.random.RandomState(7)

n_samp = 1000
times = np.arange(n_samp, dtype=np.float64) / gte.info0["sfreq"]

tmp_lat = 0.35
tmp_lat = 0.45
tmp_lat = 0.25

tmp_amp = 2
tmp_amp = 3
tmp_amp = 1

tmp_dur = 0.45
tmp_dur = 0.35
tmp_dur = 0.25

tmp_wf = GTE().generate_waveform(times=times, latency=tmp_lat, duration=tmp_dur, amplitude=tmp_amp)
log0.warning(f"{type(tmp_wf) = }")
log0.warning(f"{tmp_wf.shape = }")
log0.warning(f"{times.shape = }")
log0.warning(f"{times[:20] = }")
"""
plt.close('all')
"""
plt.plot(times, tmp_wf)
```

```
W: type(tmp_wf) = <class 'numpy.ndarray'>
W: tmp_wf.shape = (1000,)
W: times.shape = (1000,)
W: times[:20] = array([0.    , 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008,
                    0.009, 0.01 , 0.011, 0.012, 0.013, 0.014, 0.015, 0.016, 0.017,
                    0.018, 0.019])

<matplotlib.lines.Line2D at 0x7f5cb3662ae0>
```

2.17 Check Events Generation

```
event_labels = ["Ev01", "Ev02"]
event_labels = 3

temp_events, temp_event_IDs, temp_event_desc, temp_events_df = GTE().make_dummy_events(
    event_labels=event_labels,
    event_repets=100,
    event_interv=2000,
    event_begins=5000,
)

ddf(temp_events_df.head(n=4))
ddf(temp_events_df.tail(n=4))
```

	sample_num	preceding_val	event_code	event_labels
0	5000	0	2	Ev02
1	7000	0	3	Ev03
2	9000	0	2	Ev02
3	11000	0	2	Ev02

	sample_num	preceding_val	event_code	event_labels
296	597000	0	2	Ev02
297	599000	0	3	Ev03
298	601000	0	3	Ev03
299	603000	0	1	Ev01

2.18 Init: Get Singularity Event(s)

```
gte.make_singularity_events()

ddf(gte.singularity_events_df)
```

I: Singularity events generated and stored.

	sample_num	preceding_val	event_code	event_labels
0	5000	0	1	singularity

2.19 Init: Get Experimental Events

```
event_labels = ["Ev01", "Ev02"]
event_labels = 3
gte.make_experimental_events(event_labels = event_labels)

print(f"{gte.experimental_events_df.shape = }")
ddf(gte.experimental_events_df.event_labels.value_counts().sort_index())
ddf(gte.experimental_events_df.head(n=4))
ddf(gte.experimental_events_df.tail(n=4))
```

I: Experimental events generated and stored.

gte.experimental_events_df.shape = (300, 4)

```
event_labels
Ev01    100
Ev02    100
Ev03    100
Name: count, dtype: int64
```

	sample_num	preceding_val	event_code	event_labels
0	5000	0	2	Ev02
1	7000	0	3	Ev03
2	9000	0	3	Ev03
3	11000	0	1	Ev01

	sample_num	preceding_val	event_code	event_labels
296	597000	0	3	Ev03
297	599000	0	2	Ev02
298	601000	0	3	Ev03
299	603000	0	1	Ev01

2.20 Randomized Activations

```
gte.set_randomized_activations(
    num_labels=9,
    num_labels_per_event=2,
    event_labels=2,
)
print(srsly.yaml_dumps(gte.activ0))
print(srsly.yaml_dumps(gte.activ0_labels))
print(srsly.yaml_dumps(gte.activ0_events))
ddf(gte.activations_to_dataframe())
print("\n===== UPDATED EVENTS 4 =====\n")
print(f"{gte.experimental_events_df.shape = }")
ddf(gte.experimental_events_df.event_labels.value_counts().sort_index())
ddf(gte.experimental_events_df.head(n=4))
ddf(gte.experimental_events_df.tail(n=4))
print("\n===== UPDATED LABELS 4 =====\n")
print(f"{len(gte.labels4) = }")
print(f"{gte.labels4 = }")
print(f"{gte.label4_names = }")
```

I: Experimental events generated and stored.

I: Randomized activations set for 2 events with 9 total labels.

Ev01:

precentral_7-lh:

lat: 0.22

dur: 0.4

amp: 478.95

rostralmiddlefrontal_9-rh:

lat: 0.3

dur: 0.33

amp: 246.59

Ev02:

precentral_11-rh:

lat: 0.36

dur: 0.31

amp: 417.07

precentral_7-lh:

lat: 0.33

dur: 0.38

amp: 459.89

- precentral_11-rh

```

- precentral_7-lh
- rostralmiddlefrontal_9-rh

- Ev01
- Ev02

Event          Region  Latency  Duration  Amplitude
0  Ev01          precentral_7-lh    0.22    0.40    478.95
1  Ev01  rostralmiddlefrontal_9-rh    0.30    0.33    246.59
2  Ev02          precentral_11-rh    0.36    0.31    417.07
3  Ev02          precentral_7-lh    0.33    0.38    459.89
===== UPDATED EVENTS 4 =====

```

```
gte.experimental_events_df.shape = (200, 4)
```

```

event_labels
Ev01    100
Ev02    100
Name: count, dtype: int64

```

```

sample_num  preceding_val  event_code  event_labels
0          5000             0            2          Ev02
1          7000             0            2          Ev02
2          9000             0            2          Ev02
3         11000             0            2          Ev02

sample_num  preceding_val  event_code  event_labels
196        397000           0            1          Ev01
197        399000           0            1          Ev01
198        401000           0            1          Ev01
199        403000           0            1          Ev01
===== UPDATED LABELS 4 =====

```

```

len(gte.labels4) = 3
gte.labels4 = [<Label | phantomica-20240913T000907, 'precentral_7-lh', lh : 1 vertices>, <
gte.label4_names = ['precentral_11-rh', 'precentral_7-lh', 'rostralmiddlefrontal_9-rh']

```

2.21 Predefined Activations

```

gte.set_predefined_activations()
print(srsly.yaml_dumps(gte.activ0))
print(srsly.yaml_dumps(gte.activ0_labels))
print(srsly.yaml_dumps(gte.activ0_events))
ddf(gte.activations_to_dataframe())
print("\n===== UPDATED EVENTS =====\n")
print(f"{gte.experimental_events_df.shape = }")
ddf(gte.experimental_events_df.event_labels.value_counts().sort_index())
ddf(gte.experimental_events_df.head(n=4))
ddf(gte.experimental_events_df.tail(n=4))
print("\n===== UPDATED LABELS 4 =====\n")
print(f"{len(gte.labels4) = }")

```

```
print(f"{gte.labels4 = }")
print(f"{gte.label4_names = }")
```

I: Experimental events generated and stored.

Ev01:

```
frontalpole_1-lh:
  lat: 0.35
  dur: 0.2
  amp: 300
superiorfrontal_1-lh:
  lat: 0.25
  dur: 0.4
  amp: 320
rostralmiddlefrontal_7-lh:
  lat: 0.45
  dur: 0.2
  amp: 320
superiorparietal_3-rh:
  lat: 0.25
  dur: 0.6
  amp: 320
```

Ev02:

```
superiorfrontal_1-lh:
  lat: 0.35
  dur: 0.2
  amp: 320
superiorparietal_1-rh:
  lat: 0.45
  dur: 0.2
  amp: 440
superiortemporal_1-lh:
  lat: 0.25
  dur: 0.4
  amp: 320
precentral_13-lh:
  lat: 0.35
  dur: 0.2
  amp: 320
lateraloccipital_2-rh:
  lat: 0.45
  dur: 0.3
  amp: 440
lateraloccipital_4-lh:
  lat: 0.25
  dur: 0.2
  amp: 320
```

```
- frontalpole_1-lh
- lateraloccipital_2-rh
```

```

- lateraloccipital_4-lh
- precentral_13-lh
- rostralmiddlefrontal_7-lh
- superiorfrontal_1-lh
- superiorparietal_1-rh
- superiorparietal_3-rh
- superiortemporal_1-lh

- Ev01
- Ev02

Event          Region  Latency  Duration  Amplitude
0  Ev01      frontpole_1-lh    0.35      0.2      300
1  Ev01      superiorfrontal_1-lh    0.25      0.4      320
2  Ev01  rostralmiddlefrontal_7-lh    0.45      0.2      320
3  Ev01      superiorparietal_3-rh    0.25      0.6      320
4  Ev02      superiorfrontal_1-lh    0.35      0.2      320
5  Ev02      superiorparietal_1-rh    0.45      0.2      440
6  Ev02      superiortemporal_1-lh    0.25      0.4      320
7  Ev02      precentral_13-lh    0.35      0.2      320
8  Ev02      lateraloccipital_2-rh    0.45      0.3      440
9  Ev02      lateraloccipital_4-lh    0.25      0.2      320
===== UPDATED EVENTS =====

```

```
gte.experimental_events_df.shape = (200, 4)
```

```

event_labels
Ev01      100
Ev02      100
Name: count, dtype: int64

```

```

sample_num  preceding_val  event_code  event_labels
0          5000             0           1          Ev01
1          7000             0           2          Ev02
2          9000             0           1          Ev01
3         11000             0           2          Ev02

sample_num  preceding_val  event_code  event_labels
196        397000           0           1          Ev01
197        399000           0           2          Ev02
198        401000           0           1          Ev01
199        403000           0           1          Ev01
===== UPDATED LABELS 4 =====

```

```

len(gte.labels4) = 9
gte.labels4 = [<Label | phantomica-20240913T000907, 'frontpole_1-lh', lh : 1 vertices>,
gte.label4_names = ['frontpole_1-lh', 'lateraloccipital_2-rh', 'lateraloccipital_4-lh',

```

2.22 Select Activity Labels

```
gte.labels2[0].name
```

```
bankssts_1-lh
```

2.23 Number of Samples in Trial

```
print(f"{gte.activ0_trial_num_samp = }")
```

```
gte.activ0_trial_num_samp = 1000
```

2.24 Times

```
print(f"{len(gte.times0) = }")  
print(f"{gte.times0[:5] = }")
```

```
len(gte.times0) = 1000  
gte.times0[:5] = array([0.    , 0.001, 0.002, 0.003, 0.004])
```

2.25 Initialize Source Simulator

```
gte.initialize_source_simulator()  
print(gte.source_simulator)
```

```
I: SourceSimulator initialized with timestep=0.001000s  
<mne.simulation.source.SourceSimulator object at 0x7f5cb3631af0>
```

2.26 Add data to Source Simulator

```
gte.add_data_to_source_simulator()
```

```
W: 0: Ev01 [1] (100, 3)  
W: - 0: frontpole_1-lh --- {'lat': 0.35, 'dur': 0.2, 'amp': 300}  
W:   - tmp_label = <Label | phantomica-20240913T000907, 'frontpole_1-lh', lh : 1 vertice  
W:   - tmp_lat = 0.35, tmp_dur = 0.2, tmp_amp = 300  
W:   - data added source waveform to source simulator for frontpole_1-lh  
W: - 1: superiorfrontal_1-lh --- {'lat': 0.25, 'dur': 0.4, 'amp': 320}  
W:   - tmp_label = <Label | phantomica-20240913T000907, 'superiorfrontal_1-lh', lh : 1 ver  
W:   - tmp_lat = 0.25, tmp_dur = 0.4, tmp_amp = 320  
W:   - data added source waveform to source simulator for superiorfrontal_1-lh  
W: - 2: rostralmiddlefrontal_7-lh --- {'lat': 0.45, 'dur': 0.2, 'amp': 320}  
W:   - tmp_label = <Label | phantomica-20240913T000907, 'rostralmiddlefrontal_7-lh', lh :  
W:   - tmp_lat = 0.45, tmp_dur = 0.2, tmp_amp = 320
```

```

W: - data added source waveform to source simulator for rostralmiddlefrontal_7-lh
W: - 3: superiorparietal_3-rh --- {'lat': 0.25, 'dur': 0.6, 'amp': 320}
W: - tmp_label = <Label | phantomica-20240913T000907, 'superiorparietal_3-rh', rh : 1 ve
W: - tmp_lat = 0.25, tmp_dur = 0.6, tmp_amp = 320
W: - data added source waveform to source simulator for superiorparietal_3-rh
W: 1: Ev02 [2] (100, 3)
W: - 0: superiorfrontal_1-lh --- {'lat': 0.35, 'dur': 0.2, 'amp': 320}
W: - tmp_label = <Label | phantomica-20240913T000907, 'superiorfrontal_1-lh', lh : 1 ver
W: - tmp_lat = 0.35, tmp_dur = 0.2, tmp_amp = 320
W: - data added source waveform to source simulator for superiorfrontal_1-lh
W: - 1: superiorparietal_1-rh --- {'lat': 0.45, 'dur': 0.2, 'amp': 440}
W: - tmp_label = <Label | phantomica-20240913T000907, 'superiorparietal_1-rh', rh : 1 ve
W: - tmp_lat = 0.45, tmp_dur = 0.2, tmp_amp = 440
W: - data added source waveform to source simulator for superiorparietal_1-rh
W: - 2: superiortemporal_1-lh --- {'lat': 0.25, 'dur': 0.4, 'amp': 320}
W: - tmp_label = <Label | phantomica-20240913T000907, 'superiortemporal_1-lh', lh : 1 ve
W: - tmp_lat = 0.25, tmp_dur = 0.4, tmp_amp = 320
W: - data added source waveform to source simulator for superiortemporal_1-lh
W: - 3: precentral_13-lh --- {'lat': 0.35, 'dur': 0.2, 'amp': 320}
W: - tmp_label = <Label | phantomica-20240913T000907, 'precentral_13-lh', lh : 1 vertice
W: - tmp_lat = 0.35, tmp_dur = 0.2, tmp_amp = 320
W: - data added source waveform to source simulator for precentral_13-lh
W: - 4: lateraloccipital_2-rh --- {'lat': 0.45, 'dur': 0.3, 'amp': 440}
W: - tmp_label = <Label | phantomica-20240913T000907, 'lateraloccipital_2-rh', rh : 1 ve
W: - tmp_lat = 0.45, tmp_dur = 0.3, tmp_amp = 440
W: - data added source waveform to source simulator for lateraloccipital_2-rh
W: - 5: lateraloccipital_4-lh --- {'lat': 0.25, 'dur': 0.2, 'amp': 320}
W: - tmp_label = <Label | phantomica-20240913T000907, 'lateraloccipital_4-lh', lh : 1 ve
W: - tmp_lat = 0.25, tmp_dur = 0.2, tmp_amp = 320
W: - data added source waveform to source simulator for lateraloccipital_4-lh

```

2.27 Get Source Time Course

```

gte.extract_activ0_stc()
gte.activ0_stc

```

```
<SourceEstimate | 9 vertices, subject : phantomica-20240913T000907, tmin : 0.0 (ms), tmax
```

2.28 Check Source Time Course

```
gte.activ0_stc
```

```
<SourceEstimate | 9 vertices, subject : phantomica-20240913T000907, tmin : 0.0 (ms), tmax
```

2.29 Generate Raw Data

```
gte.extract_activ0_raw()
```

Setting up raw simulation: 1 position, "cos2" interpolation

Event information not stored

Interval 0.000-1.000 s

Setting up forward solutions

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

Interval 0.000-1.000 s

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

Interval 0.000-1.000 s
Interval 0.000-1.000 s
Interval 0.000-1.000 s
Interval 0.000-1.000 s
Interval 0.000-1.000 s
404 STC iterations provided
[done]
EEG channel type selected for re-referencing
Adding average EEG reference projection.
1 projection items deactivated
Average reference projection was added, but has not been applied yet. Use the apply_proj m

```

2.30 Plot Raw

```
gte.activ0_raw.plot(duration=10.0, start=0.0)
```

```

Using qt as 2D backend.
<mne_qt_browser._pg_figure.MNEQtBrowser(0x5625b54714c0) at 0x7f5cb3178bc0>

```

2.31 Add Noise to Raw

```
gte.extract_activ2_raw() # TODO add noise parameters
```

```

Adding noise to 61/61 channels (61 channels in cov)
Sphere                : origin at (0.0 0.0 0.0) mm
                      radius   : 0.1 mm
Source location file  : dict()
Assuming input in millimeters
Assuming input in MRI coordinates

Positions (in meters) and orientations
2 sources
blink simulated and trace not stored
Setting up forward solutions

```

2.32 Plot Noisy Raw

```
gte.activ2_raw.plot(duration=10.0, start=0.0)
```

```

<mne_qt_browser._pg_figure.MNEQtBrowser(0x5625b7073880) at 0x7f5cb2a77080>Channels marked
none
~/anaconda3/envs/mne0/lib/python3.12/site-packages/mne_qt_browser/_pg_figure.py:3061: RuntimeWarning:
  sig.disconnect()

```

2.33 Add Evoked and Epoched Data

```
gte.extract_activ2_epochs_and_evoked()
```

```
Not setting metadata
200 matching events found
Setting baseline interval to [-0.4, 0.0] s
Applying baseline correction (mode: mean)
Created an SSP operator (subspace dimension = 1)
1 projection items activated
```

2.34 Plot Epochs

```
print(gte.activ2_epochs.event_id.keys())
gte.activ2_epochs.plot()
```

```
Using data from preloaded Raw for 20 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 20 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 20 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
{'Ev01': 1, 'Ev02': 2}
```

```
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 19 events and 1601 original time points ...
Using data from preloaded Raw for 28 events and 1601 original time points ...
Dropped 0 epochs:
```

The following epochs were marked as bad and are dropped:

```
[]
```

Channels marked as bad:

```
none
```

```
~/anaconda3/envs/mne0/lib/python3.12/site-packages/mne_qt_browser/_pg_figure.py:3061: RuntimeWarning:
  sig.disconnect()
```

2.35 Plot evoked

```
ev = list(gte.activ2_epochs.event_id.keys())[0]
gte.activ2_evoked[ev].plot(spatial_colors=True)
gte.activ2_evoked[ev].plot_image()
```

<Figure size 640x300 with 2 Axes>

2.36 Compute Covariances

```
gte.compute_covariances(  
    data_tmin = 0.01,  
    data_tmax = 0.60,  
    noise_tmin = None,  
    noise_tmax = 0,  
    method = "empirical")
```

```
Using data from preloaded Raw for 99 events and 1601 original time points ...  
Using data from preloaded Raw for 100 events and 1601 original time points ...  
    Created an SSP operator (subspace dimension = 1)  
    Setting small EEG eigenvalues to zero (without PCA)  
Reducing data rank from 61 -> 60  
Estimating covariance using EMPIRICAL  
Done.  
Number of samples used : 117609  
[done]  
Using data from preloaded Raw for 99 events and 1601 original time points ...  
Using data from preloaded Raw for 100 events and 1601 original time points ...  
    Created an SSP operator (subspace dimension = 1)  
    Setting small EEG eigenvalues to zero (without PCA)  
Reducing data rank from 61 -> 60  
Estimating covariance using EMPIRICAL  
Done.  
Number of samples used : 79799  
[done]  
I: Data, noise, and common covariances computed and set.
```

2.37 Plot Covariances

```
gte.activ2_data_cov.plot(gte.activ2_epochs.info)  
gte.activ2_noise_cov.plot(gte.activ2_epochs.info)  
gte.activ2_common_cov.plot(gte.activ2_epochs.info)
```

```
Computing rank from covariance with rank=None  
    Using tolerance 1.7e-12 (2.2e-16 eps * 61 dim * 1.3e+02 max singular value)  
    Estimated rank (eeg): 60  
    EEG: rank 60 computed from 61 data channels with 0 projectors  
Computing rank from covariance with rank=None  
    Using tolerance 1.2e-12 (2.2e-16 eps * 61 dim * 88 max singular value)  
    Estimated rank (eeg): 60  
    EEG: rank 60 computed from 61 data channels with 0 projectors  
Computing rank from covariance with rank=None  
    Using tolerance 1.5e-12 (2.2e-16 eps * 61 dim * 1.1e+02 max singular value)  
    Estimated rank (eeg): 60  
    EEG: rank 60 computed from 61 data channels with 0 projectors
```

<Figure size 380x370 with 2 Axes> <Figure size 380x370 with 1 Axes>

2.38 Filters

```
gte.compute_lcmv_filters()
```

```
Computing rank from covariance with rank=None
  Using tolerance 1.7e-12 (2.2e-16 eps * 61 dim * 1.3e+02 max singular value)
  Estimated rank (eeg): 60
  EEG: rank 60 computed from 61 data channels with 1 projector
Computing rank from covariance with rank=None
  Using tolerance 1.2e-12 (2.2e-16 eps * 61 dim * 88 max singular value)
  Estimated rank (eeg): 60
  EEG: rank 60 computed from 61 data channels with 1 projector
Making LCMV beamformer with rank {'eeg': 60}
Computing inverse operator with 61 channels.
  61 out of 61 channels remain after picking
Selected 61 channels
Whitening the forward solution.
  Created an SSP operator (subspace dimension = 1)
Computing rank from covariance with rank={'eeg': 60}
  Setting small EEG eigenvalues to zero (without PCA)
Creating the source covariance matrix
Adjusting source covariance matrix.
Computing beamformer filters for 8196 sources
Filter computation complete
Computing rank from covariance with rank=None
  Using tolerance 1.7e-12 (2.2e-16 eps * 61 dim * 1.3e+02 max singular value)
  Estimated rank (eeg): 60
  EEG: rank 60 computed from 61 data channels with 1 projector
Computing rank from covariance with rank=None
  Using tolerance 1.2e-12 (2.2e-16 eps * 61 dim * 88 max singular value)
  Estimated rank (eeg): 60
  EEG: rank 60 computed from 61 data channels with 1 projector
Making LCMV beamformer with rank {'eeg': 60}
Computing inverse operator with 61 channels.
  61 out of 61 channels remain after picking
Selected 61 channels
Whitening the forward solution.
  Created an SSP operator (subspace dimension = 1)
Computing rank from covariance with rank={'eeg': 60}
  Setting small EEG eigenvalues to zero (without PCA)
Creating the source covariance matrix
Adjusting source covariance matrix.
Computing beamformer filters for 8196 sources
Filter computation complete
I: LCMV beamformer filters computed for 2 conditions with pick_ori='vector' and weight_norm='vector'
```

2.39 Apply LCMV Filters

```
gte.apply_lcmv_filters()
```

I: LCMV beamformer applied to 2 conditions. Source estimates stored in stcs property.

2.40 Plot Activity

```
idx0 = 1
idx0 = 0
ev = list(gte.activ2_epochs.event_id.keys())[idx0]

log0.warning(f"{gte.stcs[ev].shape = }")
gte.stcs[ev].plot(
    # hemi="rh",
    hemi="split",
    subjects_dir=gte.subjects_dir,
    subject=gte.subject,
    views=["lat", "med"],
    time_label="LCMV source power in the 12-30 Hz frequency band",
)
```

W: gte.stcs[ev].shape = (8196, 3, 1601)
Using control points [0.36756688 0.47124662 3.86961154]
<mne.viz._brain._brain.Brain at 0x7f5bf43d5100>

2.41 Final Checkups

```
ddf(gte.story())
```

	Name	Declared Type	Actual Type
0	_subjects_dir	<class 'pathlib.Path'>	PosixPath
1	_subject	<class 'str'>	str
2	_info0	<class 'mne._fiff.meas_info.Info'>	Info
3	_montage0	<class 'mne.channels.montage.DigMontage'>	DigMontage
4	_trans0	<class 'mne.transforms.Transform'>	Transform
5	_src0	<class 'mne.source_space._source_space.S...'>	SourceSpace
6	_bem_model0	<class 'list'>	list
7	_bem_solution0	<class 'mne.bem.ConductorModel'>	ConductorModel
8	_fwd0	<class 'mne.forward.forward.Forward'>	Forward
9	_genuine_noise_cov0	<class 'mne.cov.Covariance'>	Covariance
10	_annot0	<class 'str'>	str
11	_labels0	typing.List[mne.label.Label]	list
12	_label0_names	<class 'str'>	list
13	_labels2	typing.List[mne.label.Label]	list
14	_label2_names	<class 'str'>	list

15	<code>_labels3</code>	<code>typing.List[mne.label.Label]</code>	<code>Non</code>
16	<code>_label3_names</code>	<code><class 'str'></code>	<code>Non</code>
17	<code>_labels4</code>	<code>typing.List[mne.label.Label]</code>	<code>lis</code>
18	<code>_label4_names</code>	<code><class 'str'></code>	<code>lis</code>
19	<code>_singularity_events</code>	<code><class 'numpy.ndarray'></code>	<code>ndarra</code>
20	<code>_singularity_event_IDs</code>	<code>typing.Dict[str, int]</code>	<code>dic</code>
21	<code>_singularity_events_desc</code>	<code>typing.Dict[int, str]</code>	<code>dic</code>
22	<code>_singularity_events_df</code>	<code><class 'pandas.core.frame.DataFrame'></code>	<code>DataFram</code>
23	<code>_experimental_events</code>	<code><class 'numpy.ndarray'></code>	<code>ndarra</code>
24	<code>_experimental_event_IDs</code>	<code>typing.Dict[str, int]</code>	<code>dic</code>
25	<code>_experimental_events_desc</code>	<code>typing.Dict[int, str]</code>	<code>dic</code>
26	<code>_experimental_events_df</code>	<code><class 'pandas.core.frame.DataFrame'></code>	<code>DataFram</code>
27	<code>_activ0</code>	<code>typing.Dict[str, typing.Dict[str, typing...</code>	<code>dic</code>
28	<code>_activ0_labels</code>	<code>typing.List[str]</code>	<code>lis</code>
29	<code>_activ0_events</code>	<code>typing.List[str]</code>	<code>lis</code>
30	<code>_activ0_trial_num_samp</code>	<code><class 'int'></code>	<code>in</code>
31	<code>_times0</code>	<code><class 'numpy.ndarray'></code>	<code>Non</code>
32	<code>_source_simulator</code>	<code><class 'mne.simulation.source.SourceSimu...</code>	<code>SourceSimulato</code>
33	<code>_activ0_stc</code>	<code><class 'mne.source_estimate.SourceEstima...</code>	<code>SourceEstimat</code>
34	<code>_activ0_raw</code>	<code><class 'mne.io.fiff.raw.Raw'></code>	<code>RawArra</code>
35	<code>_activ2_raw</code>	<code><class 'mne.io.fiff.raw.Raw'></code>	<code>RawArra</code>
36	<code>_activ2_epochs</code>	<code><class 'mne.epochs.Epochs'></code>	<code>Epoch</code>
37	<code>_activ2_evoked</code>	<code>typing.Dict[str, mne.evoked.Evoked]</code>	<code>dic</code>

2.42 Publish

```
gte.publish()
```

3 Checkups

3.1 MNE Plot: Source Space

```
gte.src0.plot(
    head=False,
    brain=None,
    skull=False,
    trans=gte.trans0,
    subjects_dir=gte.subjects_dir)
```

```
<mne.viz.backends._pyvista.PyVistaFigure at 0x7f5bf42b76b0>
```

3.2 MNE Plot: Alignment

```
mne.viz.close_all_3d_figures()
fig = mne.viz.create_3d_figure(size=(600, 400), bgcolor=(0.00, 0.00, 0.00))
src = None
```

```

src = gte.src0 # get source positions
fwd = gte.fwd0
fwd = None # get no quivers
surfaces = ["white", "head"]
mne.viz.plot_alignment(
    info=gte.info0,
    trans=gte.trans0,
    subject=gte.subject,
    subjects_dir=gte.subjects_dir,
    surfaces=surfaces,
    coord_frame="mri",
    meg=(),
    eeg=dict(original=0.2, projected=0.8),
    fwd=fwd,
    dig=False,
    ecog=False,
    src=src,
    bem=gte.bem_model0,
    mri_fiducials=True,
    seeg=False,
    fnirs=False,
    show_axes=True,
    dbs=False,
    fig=fig,
    interaction="terrain",
    sensor_colors="magenta",
    verbose=True)

```

Channel types:: eeg: 61

Projecting sensors to the head surface

<mne.viz.backends._pyvista.PyVistaFigure at 0x7f5be7761b80>

3.3 MNE Plot: Brain, Head, Montage and Sources

```

hemi = "lh"
hemi = "rh"
hemi = "split"
hemi = "both"

surf = "inflated"
surf = "pial"
surf = "white"

cortex = "high_contrast"
cortex = "low_contrast"
cortex = "classic"

Brain = mne.viz.get_brain_class()
brain = Brain(
    subject=gte.subject,
    hemi=hemi,

```

```

    surf=surf,
    cortex=cortex,
    subjects_dir=gte.subjects_dir,
    alpha=0.4,
    size=(800, 600),
)
brain.add_annotation(gte.annot0, borders=False, alpha=1.0)
brain.add_sensors(info=gte.info0, trans=gte.trans0, eeg=dict(original=0.2, projected=1.0))
brain.add_forward(fwd=gte.fwd0, trans=gte.trans0)
brain.add_head(dense=True, color="white", alpha=0.15)
type(brain)

```

```

Using phantomica-20240913T000907-head-dense.fif for head surface.
    1 BEM surfaces found
    Reading a surface...
[done]
    1 BEM surfaces read
Channel types:: eeg: 61
Projecting sensors to the head surface
Using phantomica-20240913T000907-head-dense.fif for head surface.
    1 BEM surfaces found
    Reading a surface...
[done]
    1 BEM surfaces read

mne.viz._brain._brain.Brain

```