```
In [3]: %run plot_simulate_evoked.py
Warning: no DISPLAY environment variable.
--No graphics will be displayed.
Joblib will run 2 trial(s) in parallel by distributing trials over 2
jobs.
/opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
parallel_backends.py:508: UserWarning: joblib not installed. Cannot
run in parallel.
  warn('joblib not installed. Cannot run in parallel.')
----------------------------------------------------------------------
-----
FileNotFoundError                        Traceback (most recent call
last)
File ~/Downloads/plot_simulate_evoked.py:96
     93 from hnn_core import JoblibBackend
     95 with JoblibBackend(n_jobs=2):
---> 96     dpls = simulate_dipole(net, tstop=170., n_trials=2)
     98
######################################################################
#########
     99 # Rather than reading smoothing and scaling parameters from
file, we recommend
    100 # explicit use of the :meth:`~hnn_core.dipole.Dipole.smooth`
and
    101 # :meth:`~hnn_core.dipole.Dipole.scale` methods instead. Note
that both methods
    102 # operate in-place, i.e., the objects are modified.
    103 window_len, scaling_factor = 30, 3000

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
dipole.py:100, in simulate_dipole(net, tstop, dt, n_trials,
record_vsec, record_isec, postproc)
     95 if postproc:
     96     warnings.warn('The postproc-argument is deprecated and
will be removed'
     97                   ' in a future release of hnn-core. Please
define '
     98                   'smoothing and scaling explicitly using
Dipole methods.',
     99                   DeprecationWarning)
--> 100 dpls = _BACKEND.simulate(net, tstop, dt, n_trials, postproc)
    102 return dpls

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
parallel_backends.py:558, in JoblibBackend.simulate(self, net, tstop,
dt, n_trials, postproc)
    555 print(f"Joblib will run {n_trials} trial(s) in parallel by "
    556       f"distributing trials over {self.n_jobs} jobs.")
    557 parallel, myfunc = self._parallel_func(_simulate_single_trial)
--> 558 sim_data = parallel(myfunc(net, tstop, dt, trial_idx) for
    559                             trial_idx in range(n_trials))
    561 dpls = _gather_trial_data(sim_data, net=net,
n_trials=n_trials,
    562                           postproc=postproc)
    564 return dpls

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
parallel_backends.py:558, in <genexpr>(.0)
    555 print(f"Joblib will run {n_trials} trial(s) in parallel by "
    556       f"distributing trials over {self.n_jobs} jobs.")
    557 parallel, myfunc = self._parallel_func(_simulate_single_trial)
--> 558 sim_data = parallel(myfunc(net, tstop, dt, trial_idx) for
    559                     trial_idx in range(n_trials))
    561 dpls = _gather_trial_data(sim_data, net=net,
n_trials=n_trials,
    562                           postproc=postproc)
    564 return dpls

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
network_builder.py:43, in _simulate_single_trial(net, tstop, dt,
trial_idx)
     35 def _simulate_single_trial(net, tstop, dt, trial_idx):
     36     """Simulate one trial including building the network
     37 
     38     This is used by both backends. MPIBackend calls this in
mpi_child.py, once
     39     for each trial (blocking), and JoblibBackend calls this
for each trial
     40     (non-blocking)
     41     """
---> 43     neuron_net = NetworkBuilder(net, trial_idx=trial_idx)
     45     global _PC, _CVODE
     47     h.load_file("stdrun.hoc")

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
network_builder.py:305, in NetworkBuilder.__init__(self, net,
trial_idx)
    301     self._expose_imem = True
    303 self._rank = 0
--> 305 self._build()

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
network_builder.py:316, in NetworkBuilder._build(self)
    313 self._rank = _get_rank()
    315 # load mechanisms needs ParallelContext for get_rank
--> 316 load_custom_mechanisms()
    318 if self._rank == 0:
    319     print('Building the NEURON model')

File /opt/anaconda3/envs/hnn/lib/python3.12/site-packages/hnn_core/
```

```
network_builder.py:160, in load_custom_mechanisms()
    157             break
    159 if len(mech_fname) == 0:
--> 160     raise FileNotFoundError(f'No .so or .dll file found in
{mod_dir}')
    162 h.nrn_load_dll(mech_fname[0])
    163 print('Loading custom mechanism files from %s' %
mech_fname[0])

FileNotFoundError: No .so or .dll file found in /opt/anaconda3/envs/
hnn/lib/python3.12/site-packages/hnn_core/mod
```