3!!! PRINT THIS AND USE EVERY DAY OF RECORDING !!!

Preparing config to build model.

Create tetrodes config containing tetrodes to be used. Don't include bad channels!

Parameters to be updated:

- animal / day / sessions
- nbinsx, nbinsy
- tetrode number - as in model building tetrode config
- *pack.extr.xyscale* – according to testing, to get same scale as in previous models
- 64/128-channel mode: *channel.num, pack.extr.128mode*
- same lax, nbins in model build and decode
- neighbour distance threshold : 150/300 with/without attenuation
- session_model
- **speed threshold** for model usage !!!

First, run spike dump config. Validate tracking information.
Then, run model build scenario using dumped spikes.

SCENARIOS

Before running an online experiment, test the configuration on the recorded BIN-file, then just remove the BinFileReader processor

1) ONLINE
    a) screening : dump spikes + display spikes
    b) INHIBITION
        i) Synchrony inhibition
        ii) assembly dominance inhibition.
        Config: offline decoding + LPTTrigger with dominance threshold
2) OFFLINE
    a) decoding
        i) params: FR estimation delay - include both environments…

ONLINE ASSEMBLY INHIBITION

PREPARATION

[Get few tetrodes (at at least 4) near or in the layer - not necessary but better for testing**]**
(also , don't include more than 6-7 to avoid long model building)
Create synchrony detection config.
Run online synchrony inhibition.
Adjust high synchrony factor threshold.
Record 2 or 4 exploration sessions (ideally, 2 environments).
Build model, test decoding, tune kernel parameters (also depend on camera settings).
Record 2 environments.
Build model, prepare on-line classification config.
Test on-line decoding. Check decoded environments balance.

FOR ALL SCENARIOS

Check config warnings on default values

CHECK CONFIGS

lpt.trigger.ttpath having timestamp in it

1. Day before - try to build model (offline) on this day data and do assembly detection on-line. Run **gradient descent** for model parameters tuning. Validate # of neighbours distribution (to be ~600).
2. Screen and prepare tetrode config / synchrony tetrodes - 20 minutes after last screening.
3. Define synchrony detection threshold for the rate to be 1 Hz. Use 20 ms window.
4. Record pre-sleep and define baseline threshold for balanced (50/50) environment decoding. Define 10- and 90-percentile for confidence for decision making before timeout.
5. Prepare model building config: day/session/tetrode number/synchrony tetrodes
6. Record learning.
7. Run dump
    a. speed.est.meansn.start - not too early, might be buggy !!!
8. Run the sampling rate calculation.
9. Build model (loading calculated sampling rates).
10. Check all parameters, including AXONA package size.
11. Set the number of spikes parameter (calculated during dump)
12. Start inhibition. After ~5 minutes check event rate and inhibition rate. Adjust frequency if needed.
    a. Check again later because estimations of FR can change
13. Check whether all required output files are created, restart if needed:
    a. log file
    b. timestamps file

POSSIBLE ISSUES

spike buffer rewind because of too many spiked - shouldn't happen during model build

increase buffers size (and waveshape pool)

TODO

Binary classifier

protocol for the model training part: light on? synchrony inhibition? (run together?)

Go through configs and write down all important and changeable parameters

OTHER IMPORTANT PARAMETERS

pc.scale **-** *e.g.* if attenuators are used. Not important for KDE, because features are standardized

30 with attenuators, 60-without

spike.reader.fet.scaling **- DEPRECATED AND SHOULD BE 1.0 (do a warning)**

whl shift / scale - *can later be fixed by …*

BUILD

git pull origin master

Build library in the Visual Studio

./deploy_and_start.sh

PREDICTION OPTIONS

All data / event-based (synchrony)
Fixed time-window / fixed number of spikes with overlap / 1 prediction per event
Spike set - based / continuous update

CHANGES TO LFPO

Test correctness and performance on multiple animals
Update parameters in all configs, both Linux and Windows
Have alternative strategies
Potential bottlenecks - find and test
Test on someone's animal (full pipeline) before starting new experiements
Don't change parameters without a reason !!!

START TESTING (DUMP, BUILD, INHIBIT) EARLY AND DON'T CHANGE CONFIG FILES AFTER

PROTOCOL

5-ARM: hide myself

FULL AXONA + LFPO+LASER TESTING PROTOCOL (BEFORE RECORDING DAYS)
Date: _____ Animal:_____

1. ▢ (Optionally) Laser power: _____
2. ▢ Prepare tetrodes config with valid channels. # of tetrodes: _____
3. ▢ Spike dump / display; Dump processing time w/o PCA: 1 minute: _____
4. ▢ *Regular flashes :*
    a. ▢ *Without laser offline*
    b. ▢ *With laser offline*
    c. ▢ *With laser online*
5. ▢ Durability crash-test: 2h+ inhibition session, may be without laser.
6. ▢ Synchrony inhibition
    a. ▢ Offline: tetrodes, threshold : _____, time window : _____
    b. ▢ Online; Rate : _____
7. ▢ Model build + assembly inhibition
    a. ▢ Record 2 x 7.5 minutes sessions of running with tuning in-between (to get difference)
    b. ▢ Select tetrodes having enough units
    c. ▢ Run FR / N spikes calculation and dump
        i. N spikes: _____
    d. ▢ Run model build, model ID: _____
        i. ▢ Check model files (gnuplot) :occupancy, LX, LAX
    e. ▢ Run assembly inhibition :
        i. ▢ Offline on previous sleep
        ii. ▢ online
8. ▢ Test model build, online decoding and assembly inhibition on recorded control data.

FULL SETUP + AXONA + LFPO + LASER TESTING AND RECORDING PROTOCOL (RECORDING DAYS):

1. ▢ Day before: check spike dump + model + … and don't change code + configs without testing
2. ▢ Setup preparation
    a. ▢ Towels on the doors
    b. ▢ "Quiet" signs
    c. ▢ Environment changed to be **novel** (external cues, objects, floor, different linear tracks), nothing that could scare animal:

       _____
    d. ▢ New batteries installed
    e. ▢ Enough palettes
    f. ▢ Tracking borders checked
    g. ▢ Photos of environment taken
    h. ▢ Cable mobility checked: loose when animal in the sleep box and not too tight on edges
    i. ▢ No reflections and other tracking artefacts (check moving headstage)
    j. ▢ Enough disk space

3. ▢ (Optionally) Laser power: _____
4. ▢ (For inhibition days) After last screen: prepare tetrodes config. Tetrodes #: _____ (copy from all tetrodes and remove bad ones)
5. ▢ Update and check configuration files:
    a. ▢ EXPERIMENTAL_spike_dump:
        i. ▢ Date, sessions
        ii. ▢ bin.path.* : 4 files
        iii. ▢ bin.x.shift = 170
        iv. ▢ tetr.conf.path - day-specific,
        v. ▢ buf.fr.save, buf.frest.delay, buf.fr.path
        vi. ▢ channel.num
        vii. ▢ pack.extr.xyscale=0.31
        viii. ▢ pack.extr.128mode
        ix. ▢ pca.num.pc, pca.num.pc.* = 2
        x. ▢ pca.scale (**60 always**)
    b. ▢ EXPERIMENTAL_build_model:
        i. ▢ date, session (keep the same across days)
        ii. ▢ kd.path (KDE executable)
        iii. ▢ tetr.conf.path to be day-specific
        iv. ▢ model.id : incremental
        v. ▢ buf.fr.load, buf.fr.path
        vi. ▢ channel.num: ▢ 64 ▢ 128
        vii. ▢ pca.num.pc, pca.num.oc.* = 2
        viii. ▢ spike.reader.fet.scaling = 1.0
        ix. ▢ nbinsx=75, nbinsy = 35, bin.size=4
        x. ▢ kd.max.jobs = # processors
        xi. ▢ kd.sigma.a = 0.5, kd.sigma.x=2.0, kd.sigma.xx = 0.06
        xii. ▢ kd.sampling.delay = 21600000
        xiii. ▢ kd.sampling.end = unlimited
        xiv. ▢ buf.rewind.guard = ${kd.sampling.delay}
        xv. ▢ kd.speed.thold=5.0

xvi. ▢ kd.min.spikes = 20000
xvii. ▢ kd.spike.graph.cover.distance.threshold=800
xviii. ▢ kd.spike.graph.cover.nneighb=2000
xix. ▢ kd.wait.speed = 1
xx. ▢ kd.dump.delay=huge
xxi. ▢ kd.wait.speed=0
xxii. ▢ kd.neighb.num=3
xxiii. ▢ kd.swr.swithc=1
xxiv. ▢ kd.pred.dump=0
xxv. ▢ kd.fixed.spike.number = see output of the build model
xxvi. ▢ kd.single.pred.per.swr=1
xxvii. ▢ kd.pred.continuous=0
xxviii. ▢ speed.est.meansn.thold=5.0 (same as for model building)
xxix. ▢ speed.est.meansn.start = 21600000 (2nd half)
xxx. ▢ speed.est.meansn.end = 43000000 (before end)

c. ▢ EXPERIMENTAL_assembly_inhibition:
i. ▢ Day/session/session_model
ii. ▢ model.id same as in build model
iii. ▢ pca.path.base = ../${session_model}/pca/pca_
iv. ▢ kd.path.base =
v. ▢ buf.fr.load, buf.fr.path=../${session_model/frates.}
vi. ▢ channel.num
vii. ▢ synchrony = # of tetrodes and full list
viii. ▢ pop.vec.win.len.ms = 10/15 ms
ix. ▢ high.synchrony.factor = 3.5 (should be adaptive!)
x. ▢ pack.extr.128mode
xi. ▢ spike.detection.nstd = 6.0 (to match dump)
xii. ▢ nbinsx=75, nbinsy=25, bin.size=4
xiii. ▢ pca.num.pc=2, pca.load=1, pca.scale (**60 always**)
xiv. ▢ kd.sampling.delay=24000. kd.prediction.delay=${kd.sampling.delay}
xv. ▢ lpt.trigger.cooldown = 3600 (including inhibition window)
xvi. ▢ lpt.trigger.sync.max.duration=0
xvii. ▢ lpt.trigger.confidence.average = 25 ???
xviii. ▢ lpt.trigger.pulse.length=2400
xix. ▢ lpt.trigger.inhibit.nonconf=0
xx. ▢ lpt.trigger.ttpath=${out.path.base}${session}_${timestamp}.timestamps
xxi. ▢ kd.ignore.lx=1

6. ▢ Make sure decision boundary (x) is correct
7. ▢ Check configs with diff and make sure no parameters are changed except for date and model.
8. ▢ Record regular pulses
9. ▢ Record pre-sleep
10. ▢ Copy pre-sleep to Linux, give a break
11. ▢ Run dump on pre-sleep (for faster assembly inhibition test and frates.txt for sync inhibition)
12. ▢ Copy frates.txt to Windows (to use for sync inhibition)
13. ▢ Record learning sessions, make sure animal drinks during the break
    a. ▢ check cable
    b. ▢ record session
    c. ▢ copy session data to Linux
14. ▢ Give animal short break while copying files to Linux computer

15. ▢ Strat full synchrony inhibition session
16. ▢ Run spike_dump_128 on Linux
17. ▢ Run model buildon Linux
18. ▢ Test model by runinng assembly inhibition on the pre-sleep file
19. ▢ Set mean spike number parameter to assembly inhibition
20. ▢ (Before every start) AXONA package number: _____
21. ▢ Run for couple of minutes and adjust threshold parameters
22. ▢ Run assembly inhibition for 2-3+ hours.
23. ▢ Run regular pulses
24. ▢ After experiment:
    a. ▢ Batteries put to charge
    b. ▢ Animal fed, weighted, put back to animal room
    c. ▢ Recorded data set to copy
    d. ▢ AXONA and lasers turned off
    e. ▢ Headstage protected