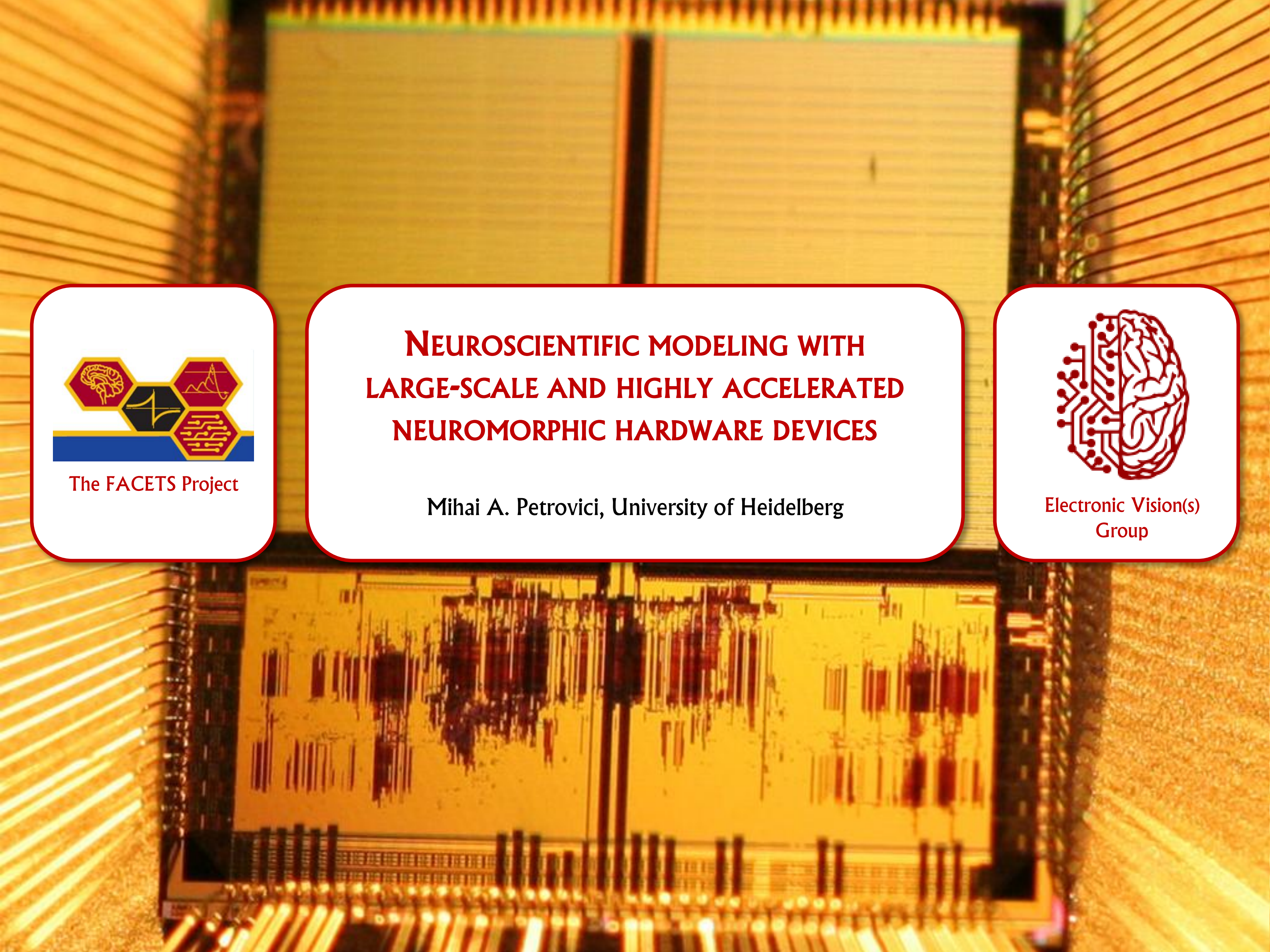


The FACETS Project

NEUROSCIENTIFIC MODELING WITH LARGE-SCALE AND HIGHLY ACCELERATED NEUROMORPHIC HARDWARE DEVICES

Mihai A. Petrovici, University of Heidelberg

Electronic Vision(s)
Group

Part I

An introduction to the FACETS neuromorphic hardware

Limits of numerical approaches

computers use too much resources

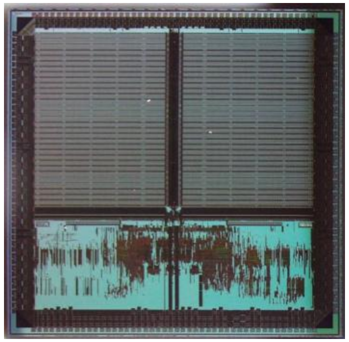
- loss of fault tolerance inherent to neural systems
- power consumption of the simulation layer

main problems of modern microelectronics

biologically inspired architectures preserve the fault tolerance and low power consumption of neural systems at the device level

→ physical model

FACETS neuromorphic hardware

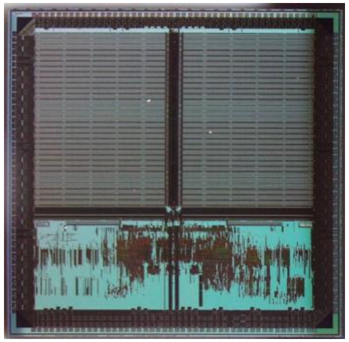


Spikey - 2006:

384 neurons

10^5 synapses

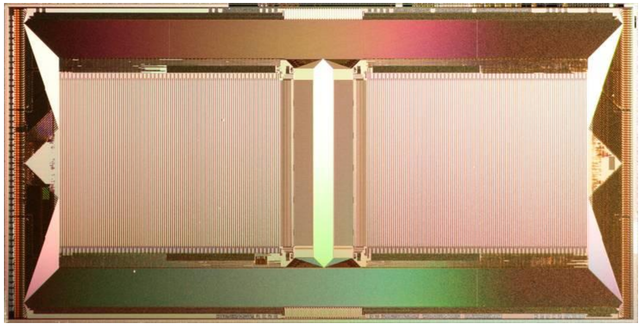
FACETS neuromorphic hardware



Spikey - 2006:

384 neurons

10^5 synapses

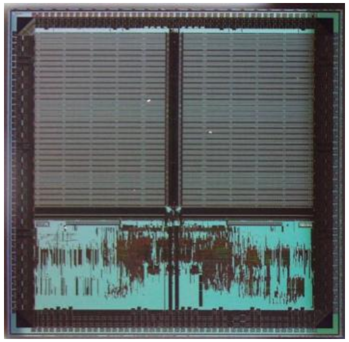


HICANN - 2010

512 neurons

$1.3 \cdot 10^5$ synapses

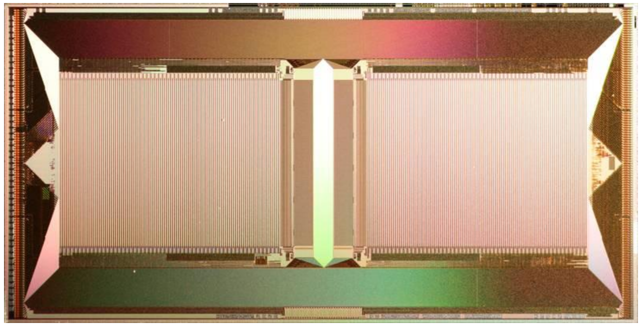
FACETS neuromorphic hardware



Spikey - 2006:

384 neurons

10^5 synapses



HICANN - 2010:

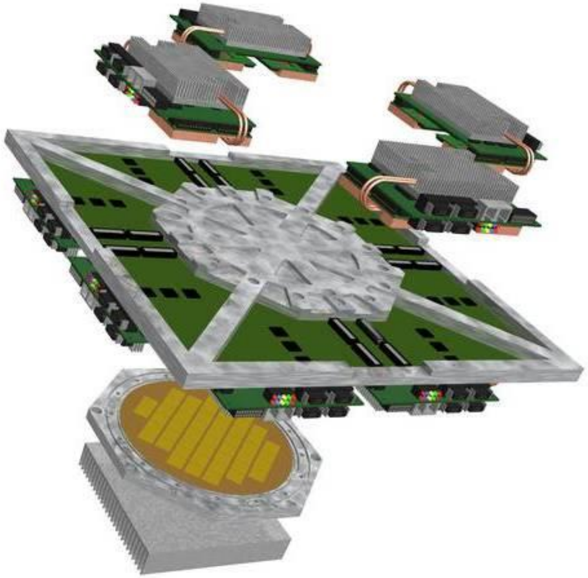
512 neurons

$1.3 \cdot 10^5$ synapses

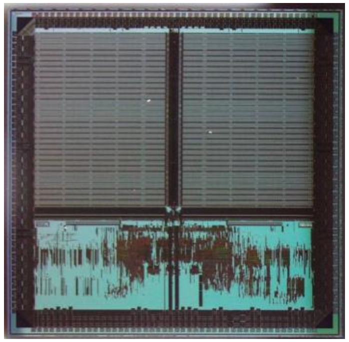
Wafer - 2011:

$16 \cdot 10^4$ neurons

$4 \cdot 10^7$ synapses



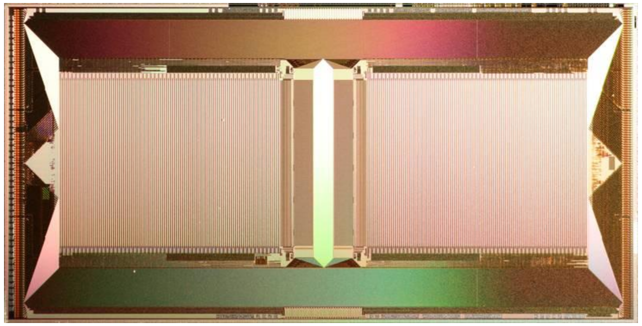
FACETS neuromorphic hardware



Spikey - 2006:

384 neurons

10^5 synapses



HICANN - 2010:

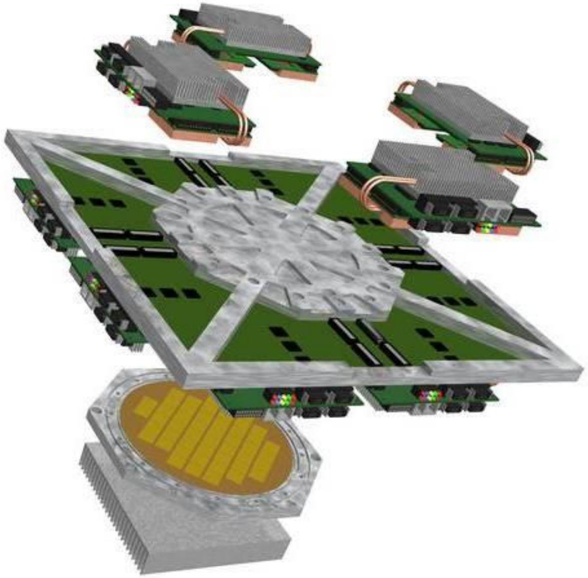
512 neurons

$1.3 \cdot 10^5$ synapses

Wafer - 2011:

$16 \cdot 10^4$ neurons

$4 \cdot 10^7$ synapses

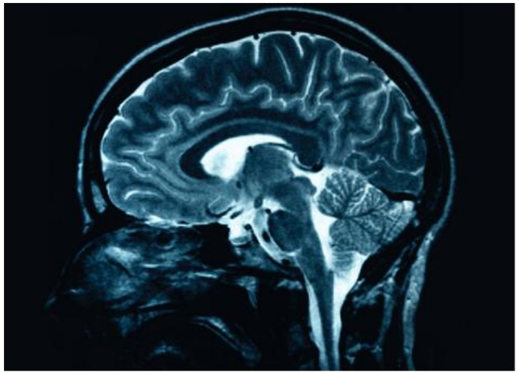


Rack – 20??:

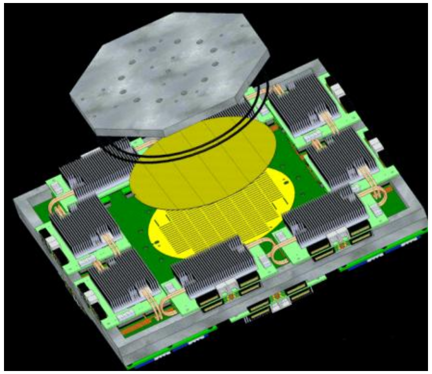
$16 \cdot 10^5$ neurons

$4 \cdot 10^8$ synapses

Hardware vs. biology



up to 10^5 speedup →



	Biological neural computation	FACETS wafer-scale hardware
Connectivity	10^{11} neurons, 10^{15} synapses 10.000 synapses per neuron	10^5 Neurons, 10^7 Synapses arbitrarily configurable
Diversity	vast range of neuron categories and parameters	multi-compartment Adaptive Exponential Integrate and Fire neurons
Plasticity	long term, short term local, global	Short Term Plasticity Spike Timing Dependent Plasticity
Timing	various time constants and delays	adjustable time constants, but no on-wafer delays
Scalability		modular, high bandwidth, low power, fault tolerant

Neuron model of choice

R. Naud et al.: *Firing patterns in the adaptive-exponential integrate-and fire-model, BiolCybern(2008) 99:335–347*

tonic spiking

initial burst

delayed accelerating

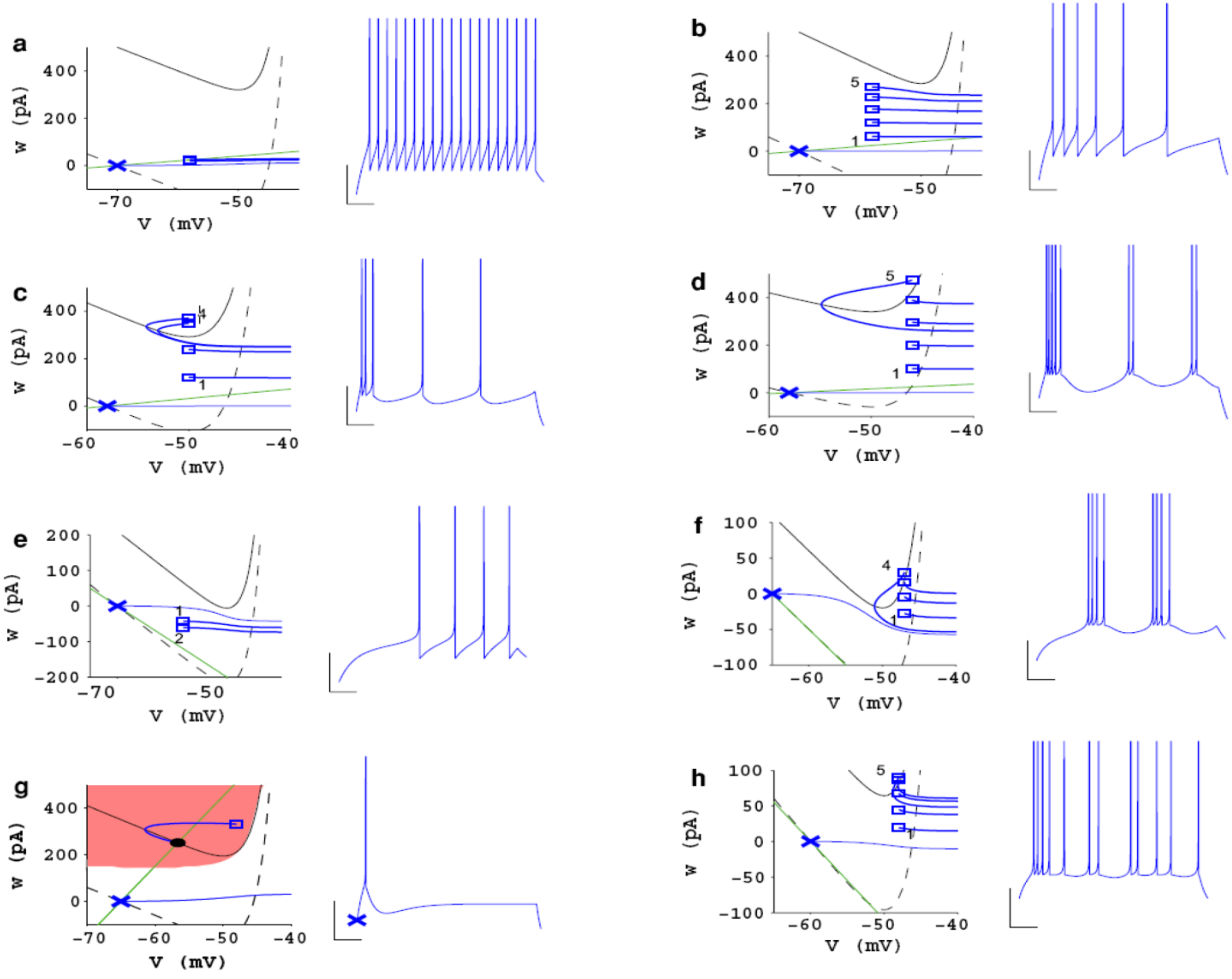
transient spiking

adaptation

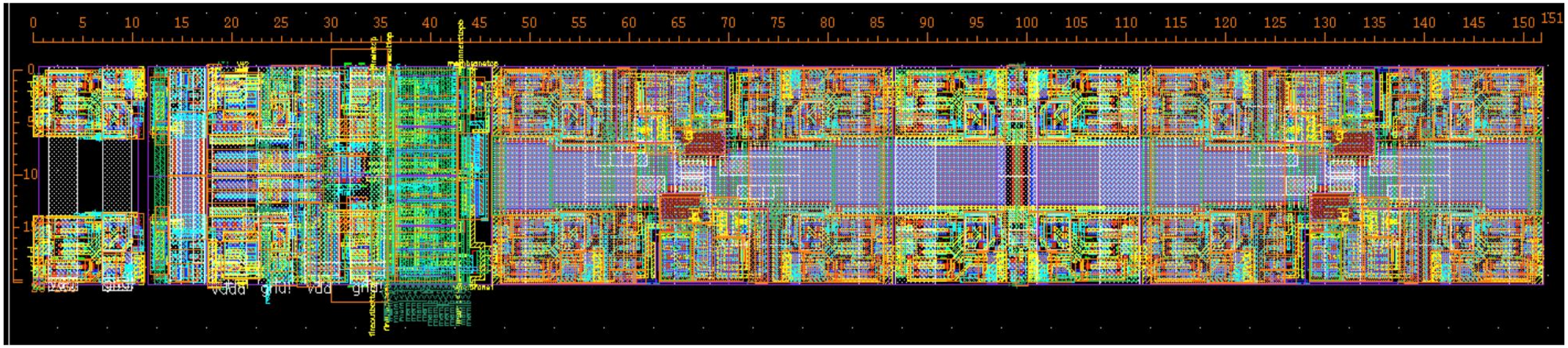
regular bursting

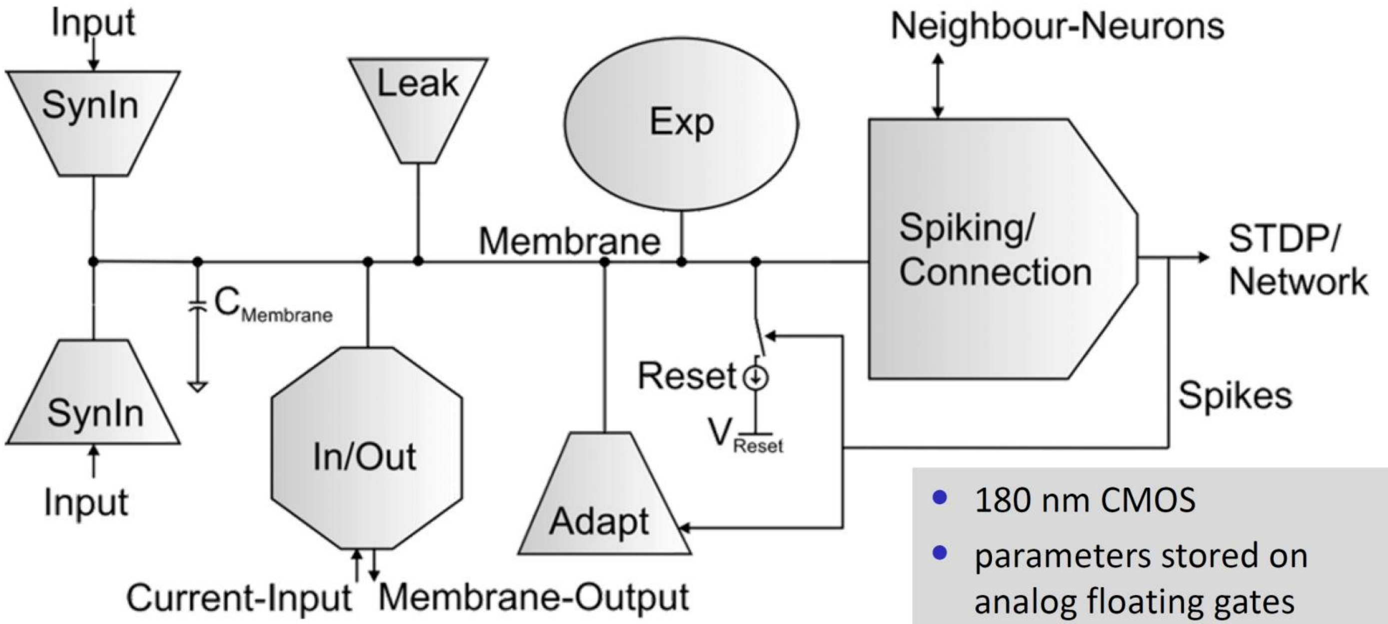
delayed regular bursting

irregular spiking



CMOS implementation of AdEx neuron

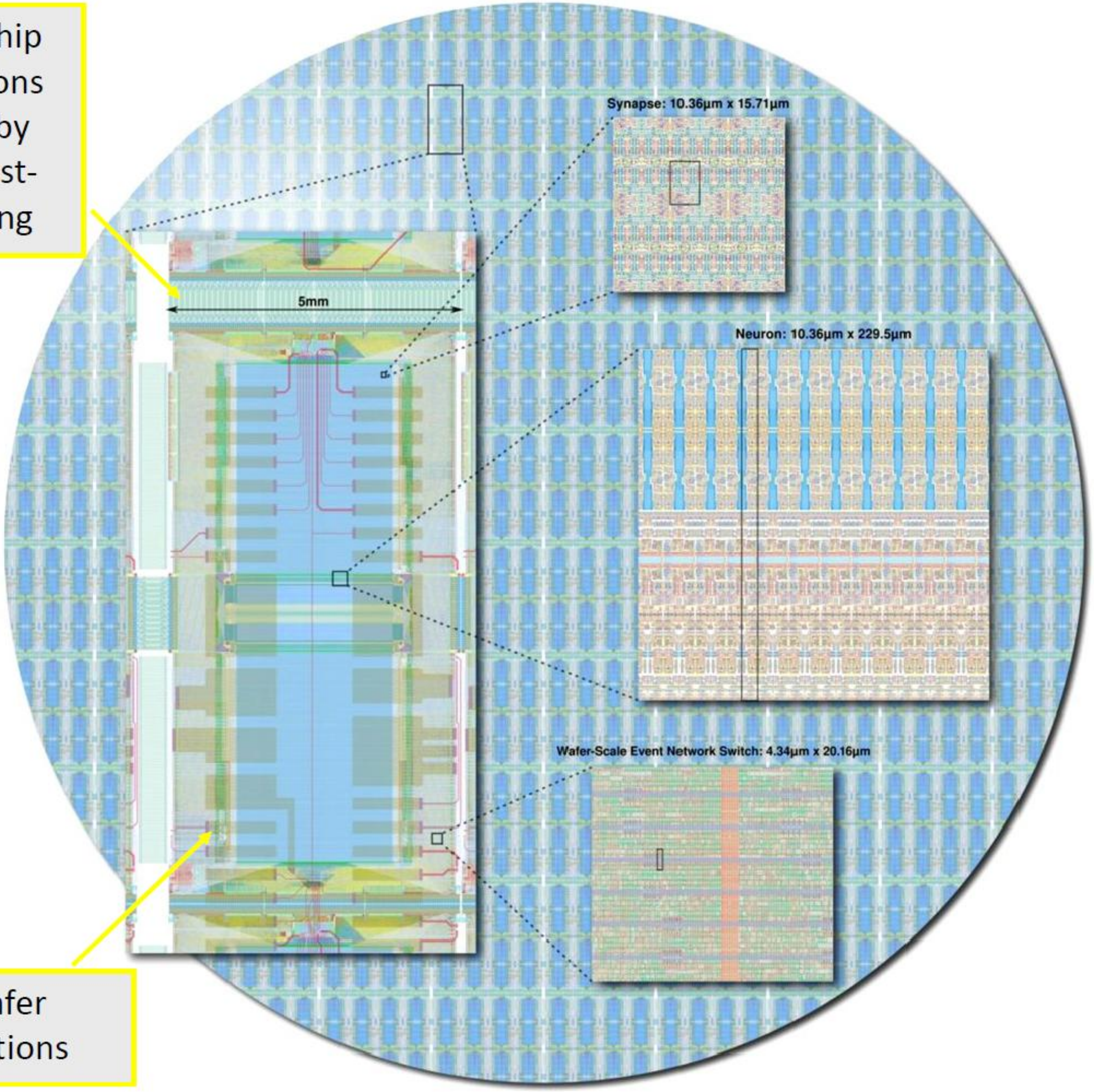




- 180 nm CMOS
- parameters stored on analog floating gates

Wafer-scale integration

chip-to-chip connections formed by wafer post-processing

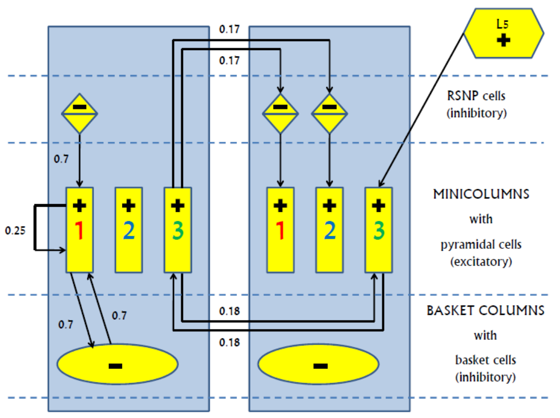


off-wafer connections

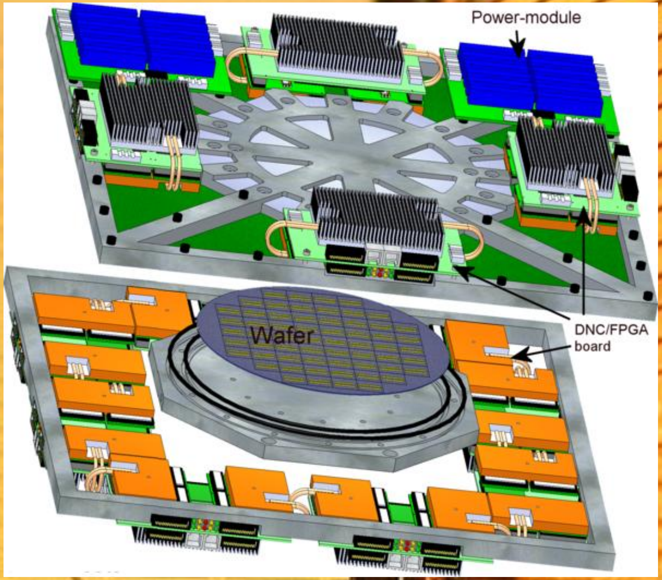
massive configuration space

→ dedicated mapping tools
→ versatile control software
→ distortion analysis and compensation

⇒ complex emulation workflow



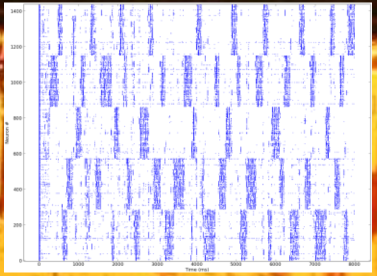
?

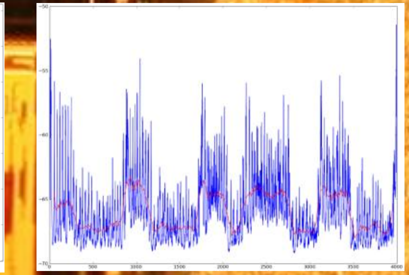


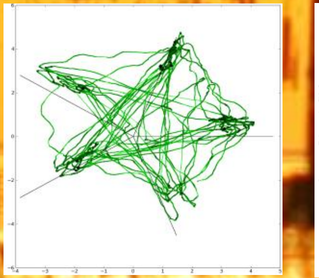
PART II (A)

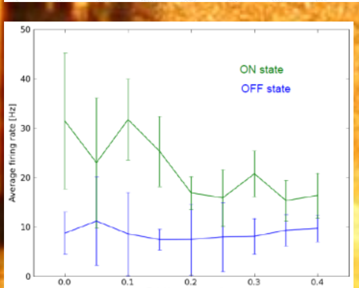
WORKFLOW:
BIOLOGY-TO-HARDWARE MAPPING

?

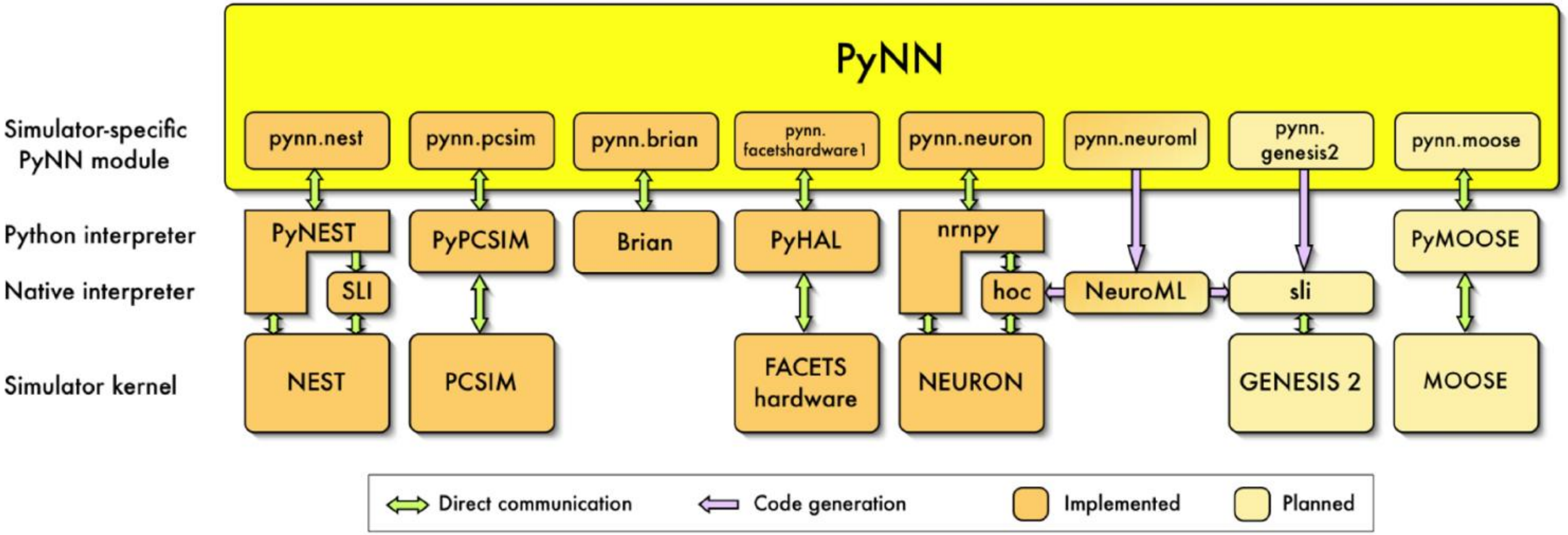




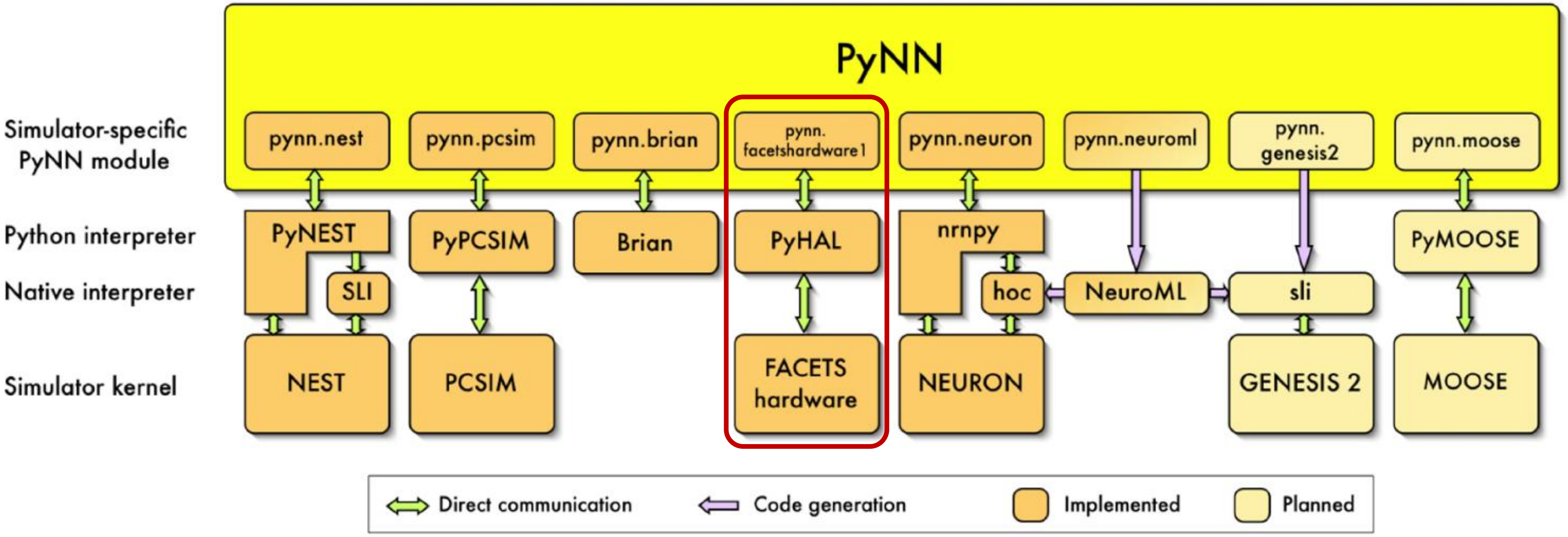




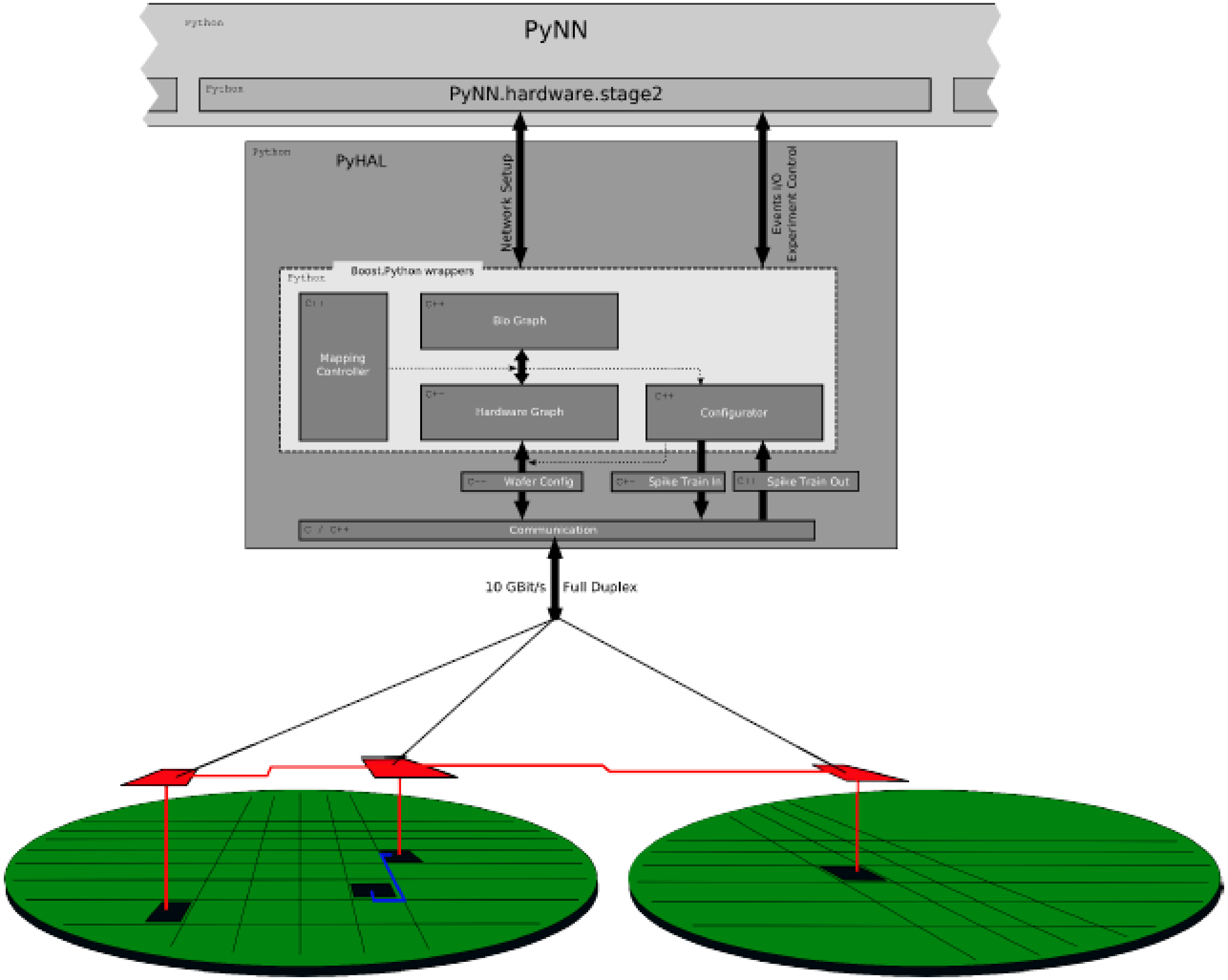
Modeling language



Modeling language



Software and hardware layers



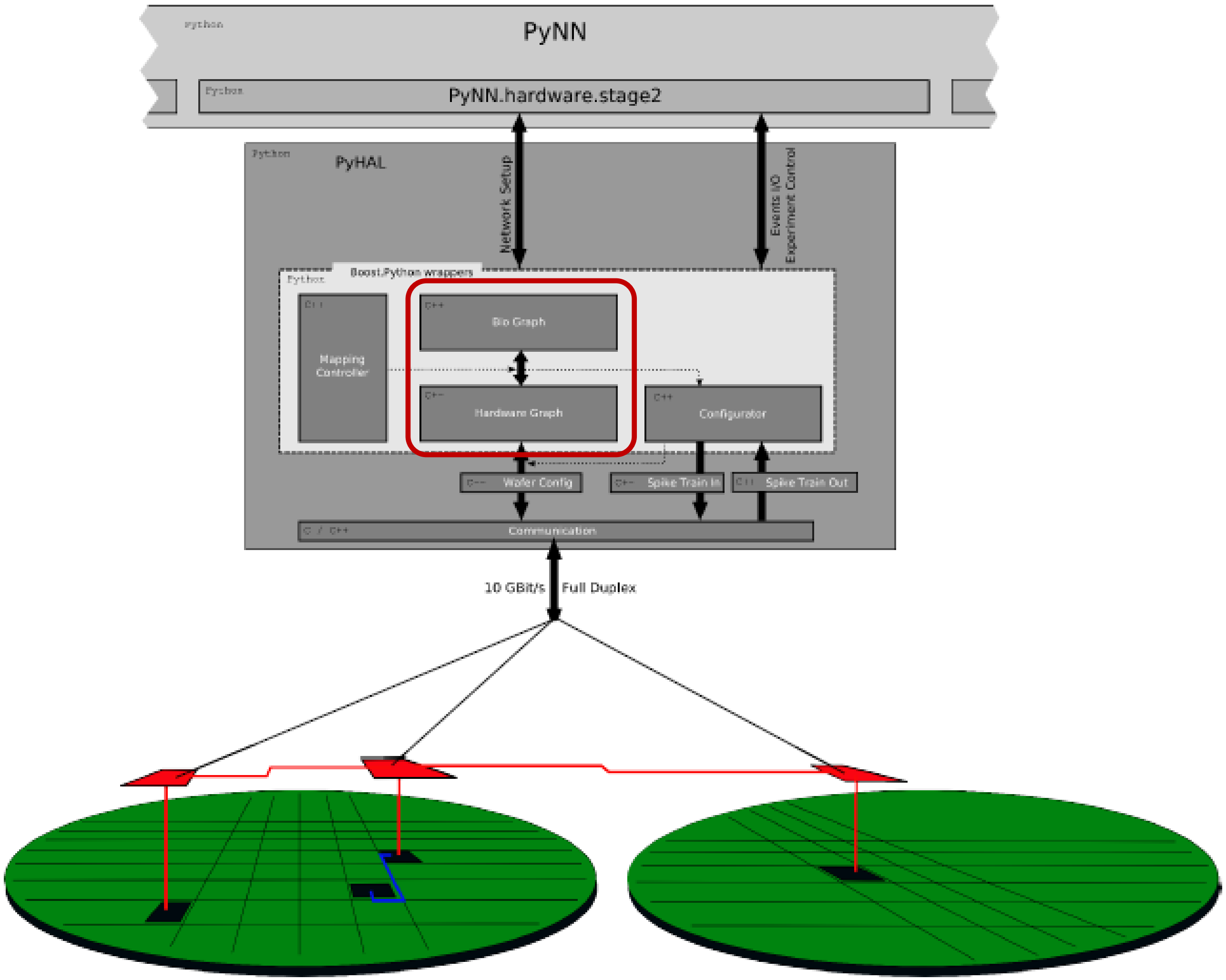
Operating Interface

Mapping Software

Digital (Layer2)

Analog (Layer1)

Software and hardware layers



Operating Interface

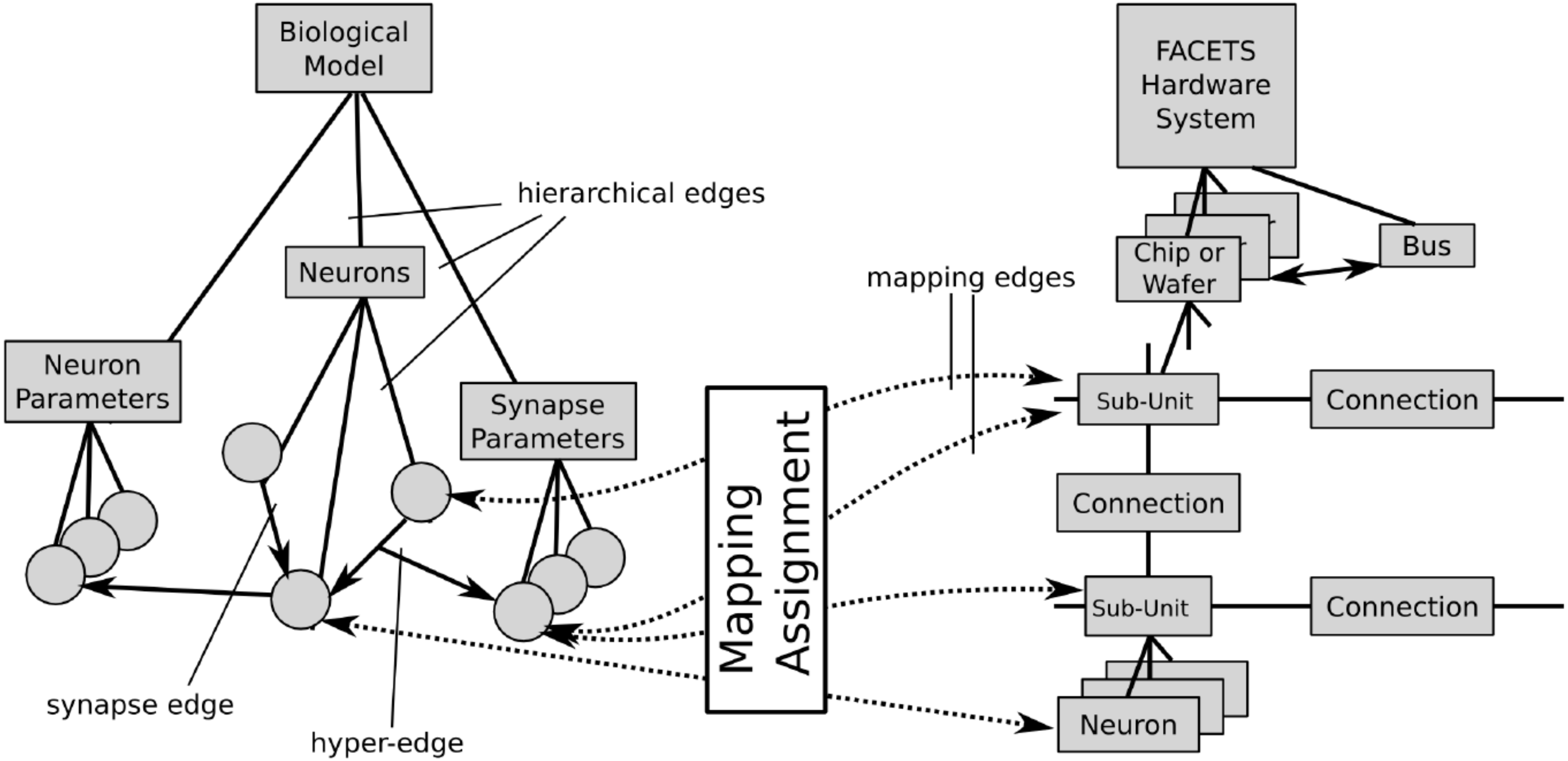
Mapping Software

Digital (Layer2)

Analog (Layer1)

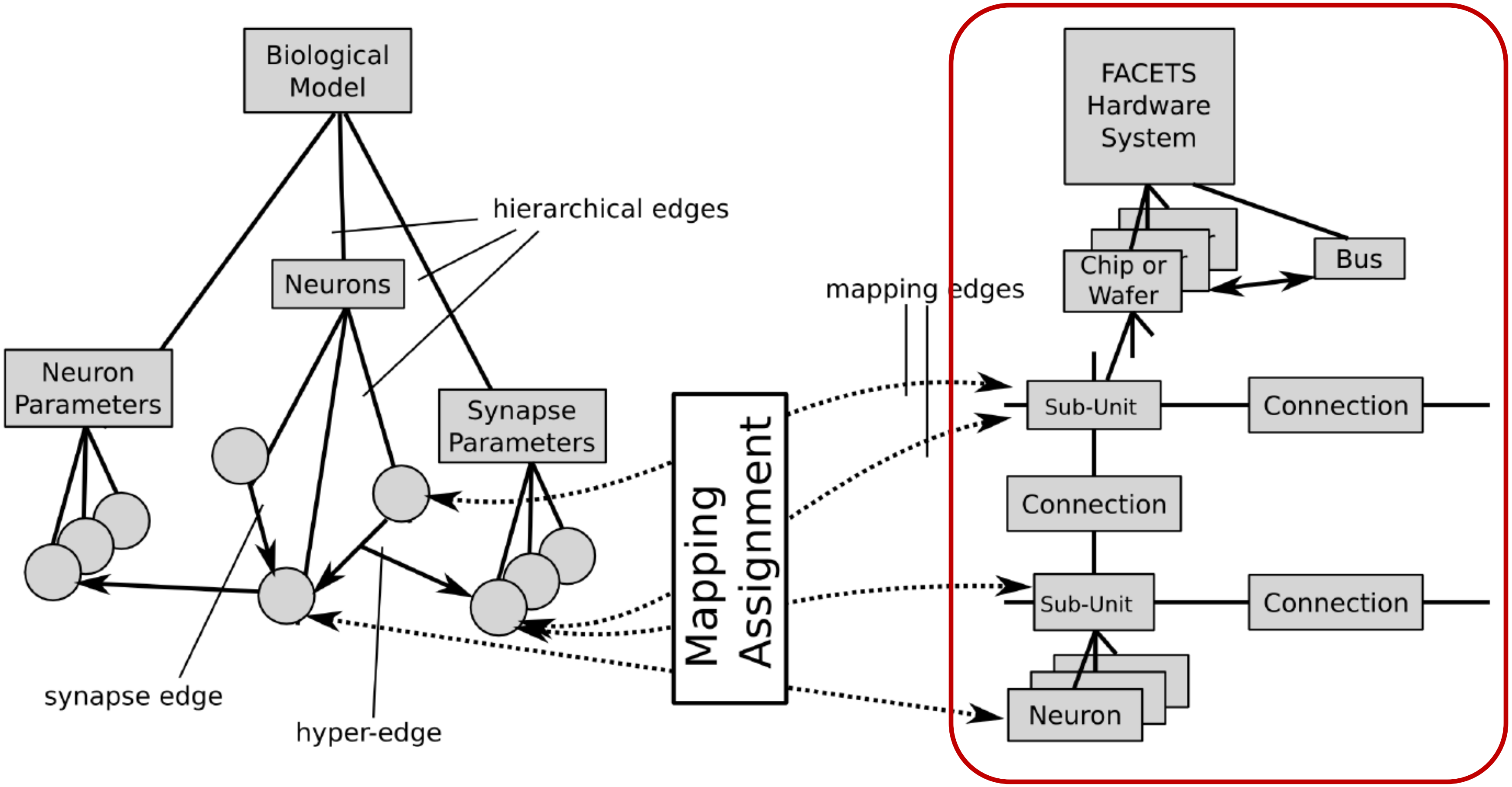
Biology-to-hardware mapping

Graph model (TUD)

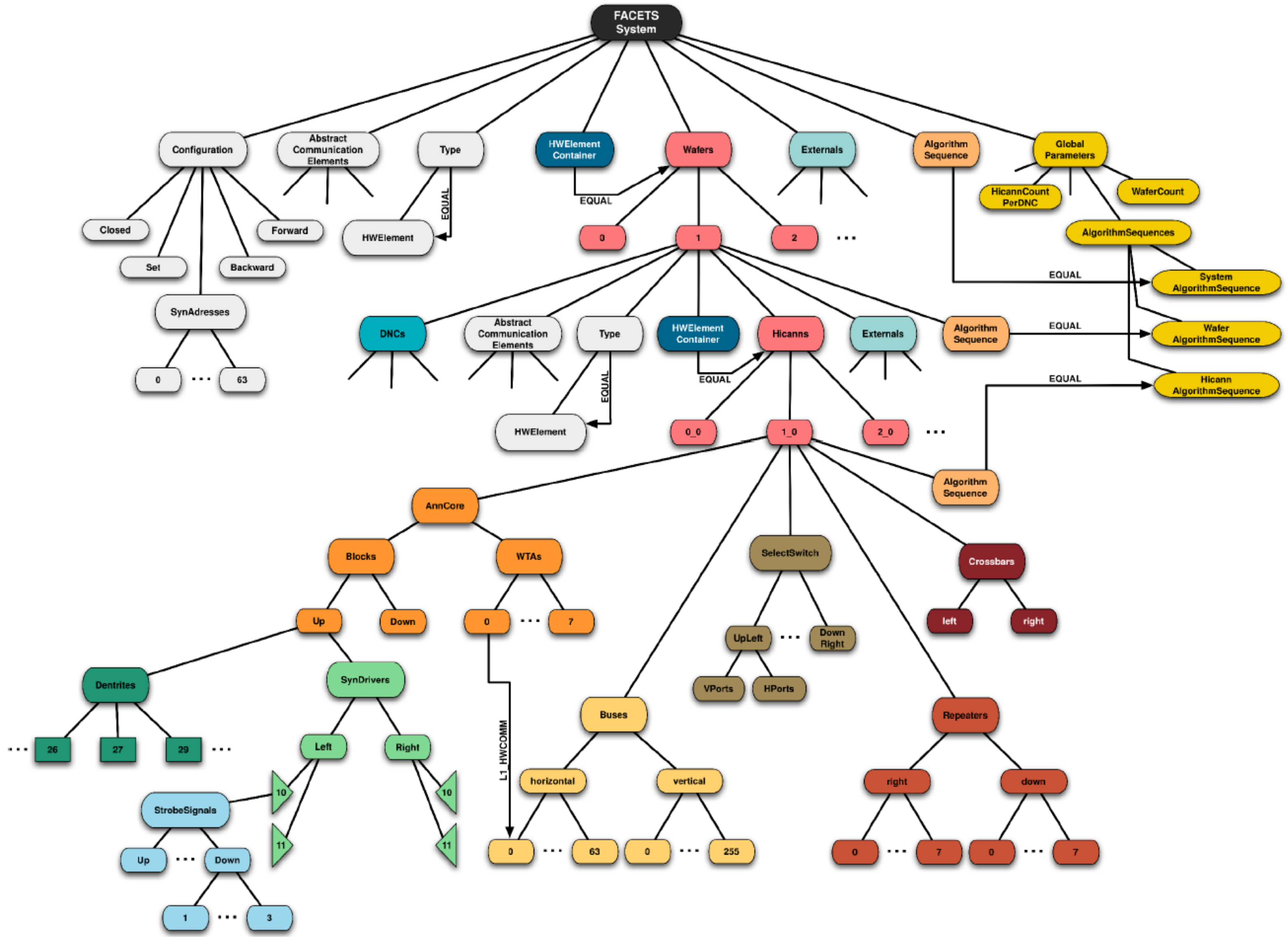


Biology-to-hardware mapping

Graph model (TUD)

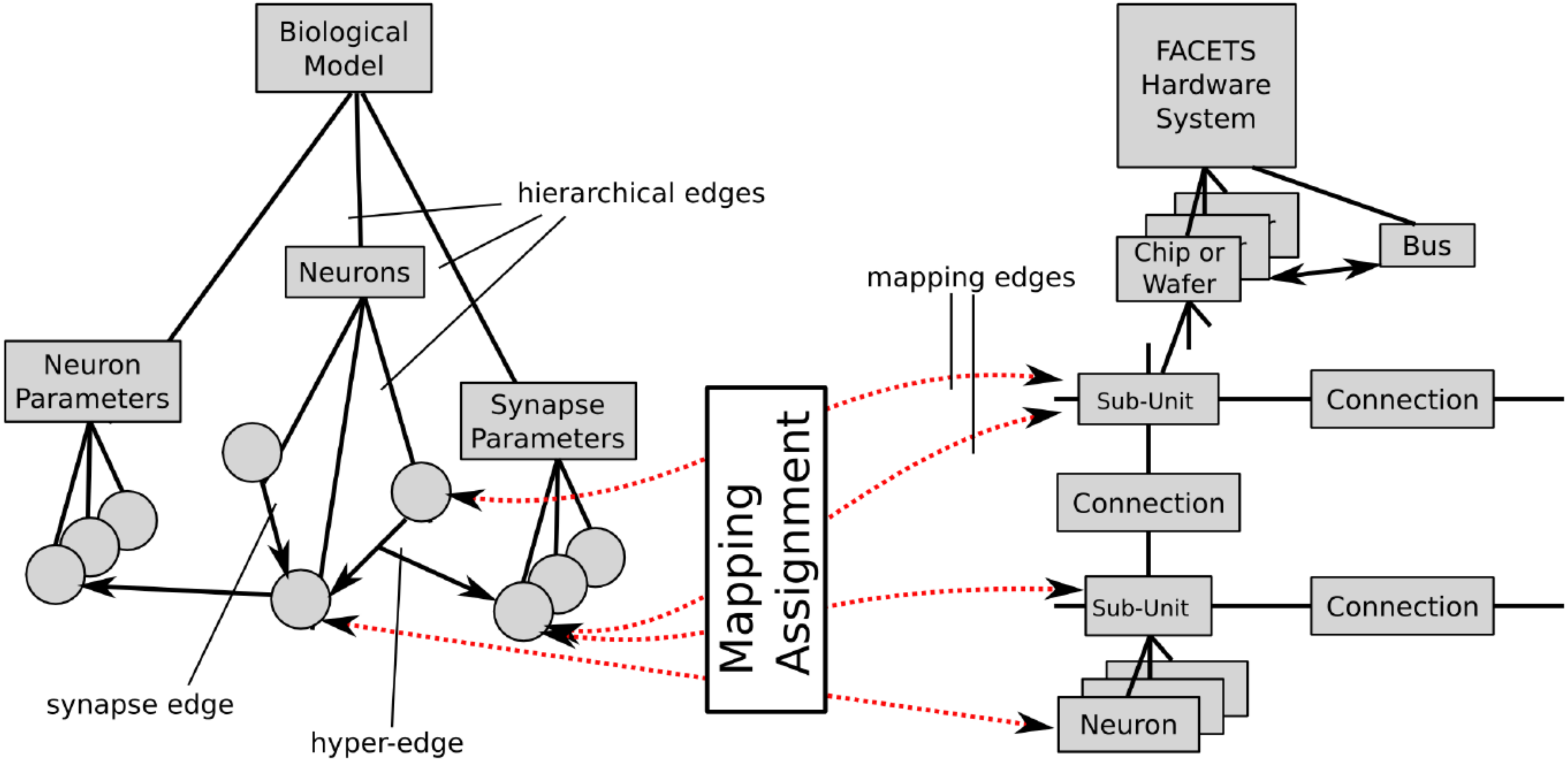


Hardware graph

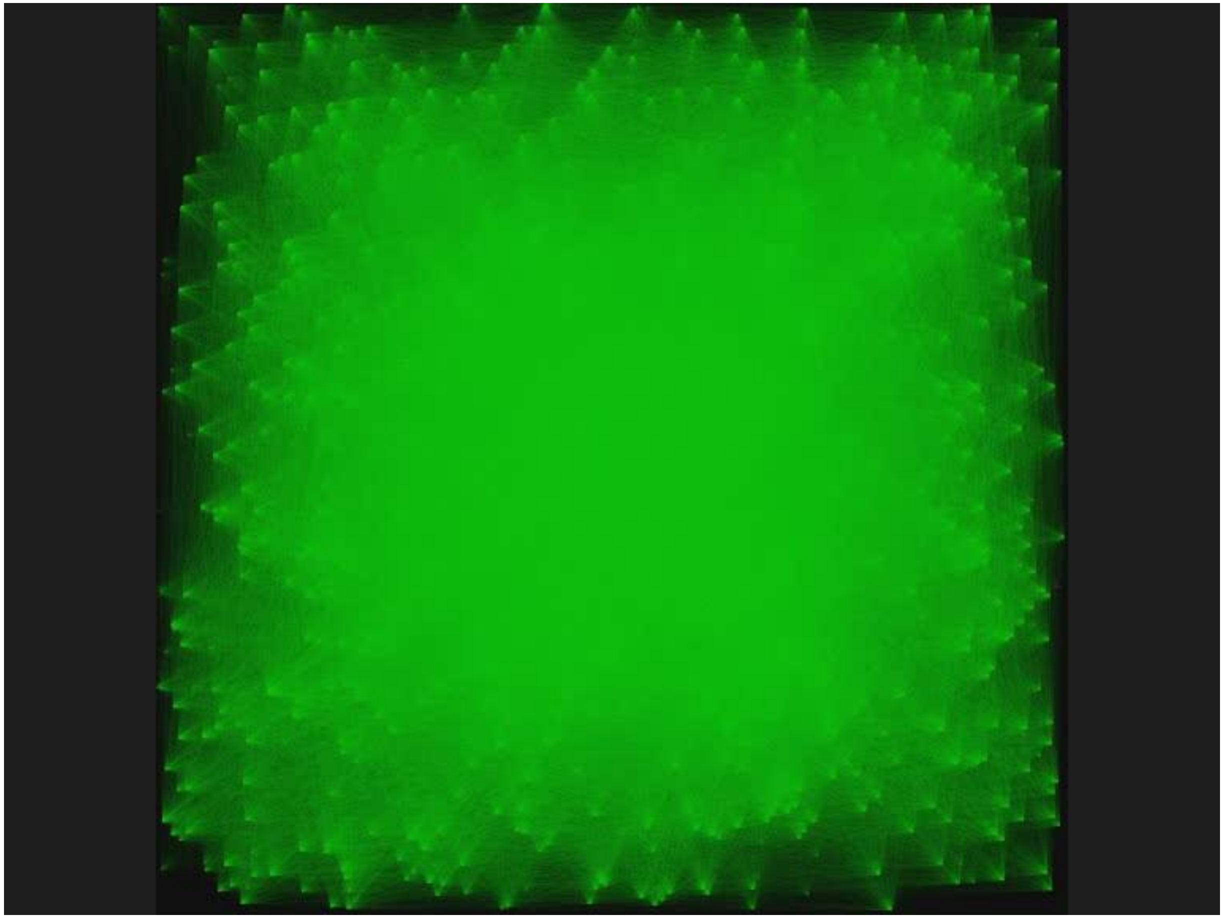


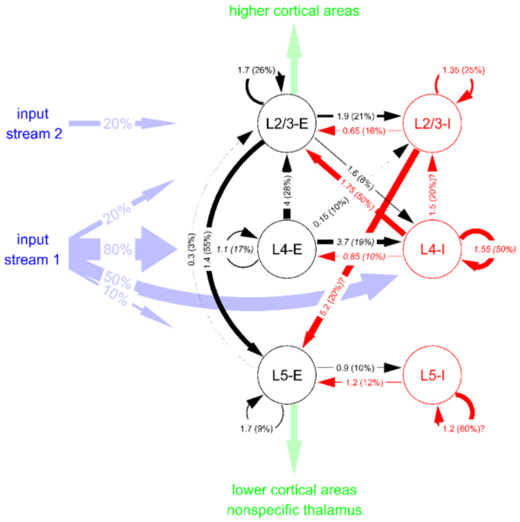
Biology-to-hardware mapping

Graph model (TUD)



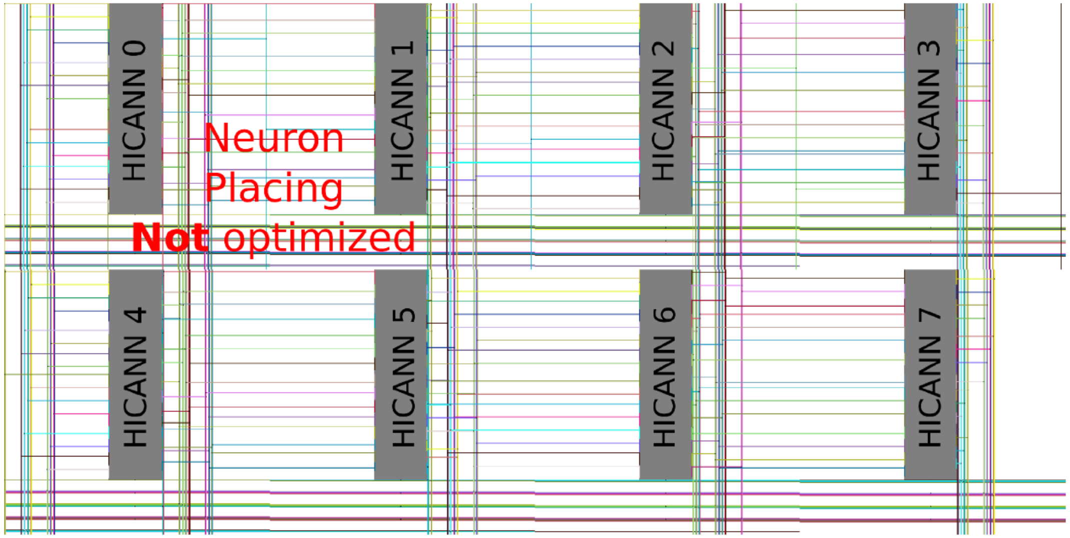
Nforce cluster algorithm



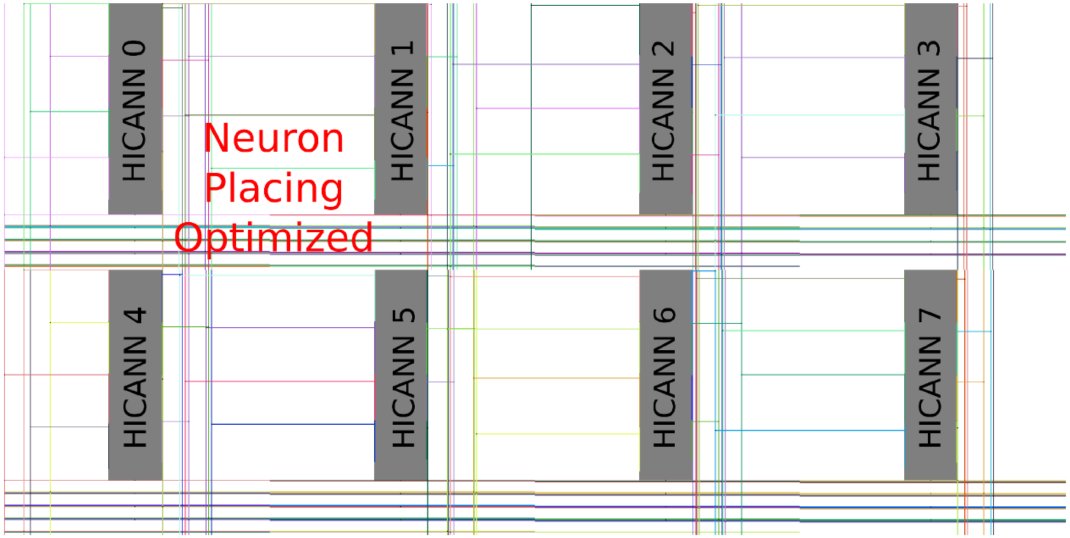


Cortical microcircuit template, Haeussler (2009)

Placing optimization

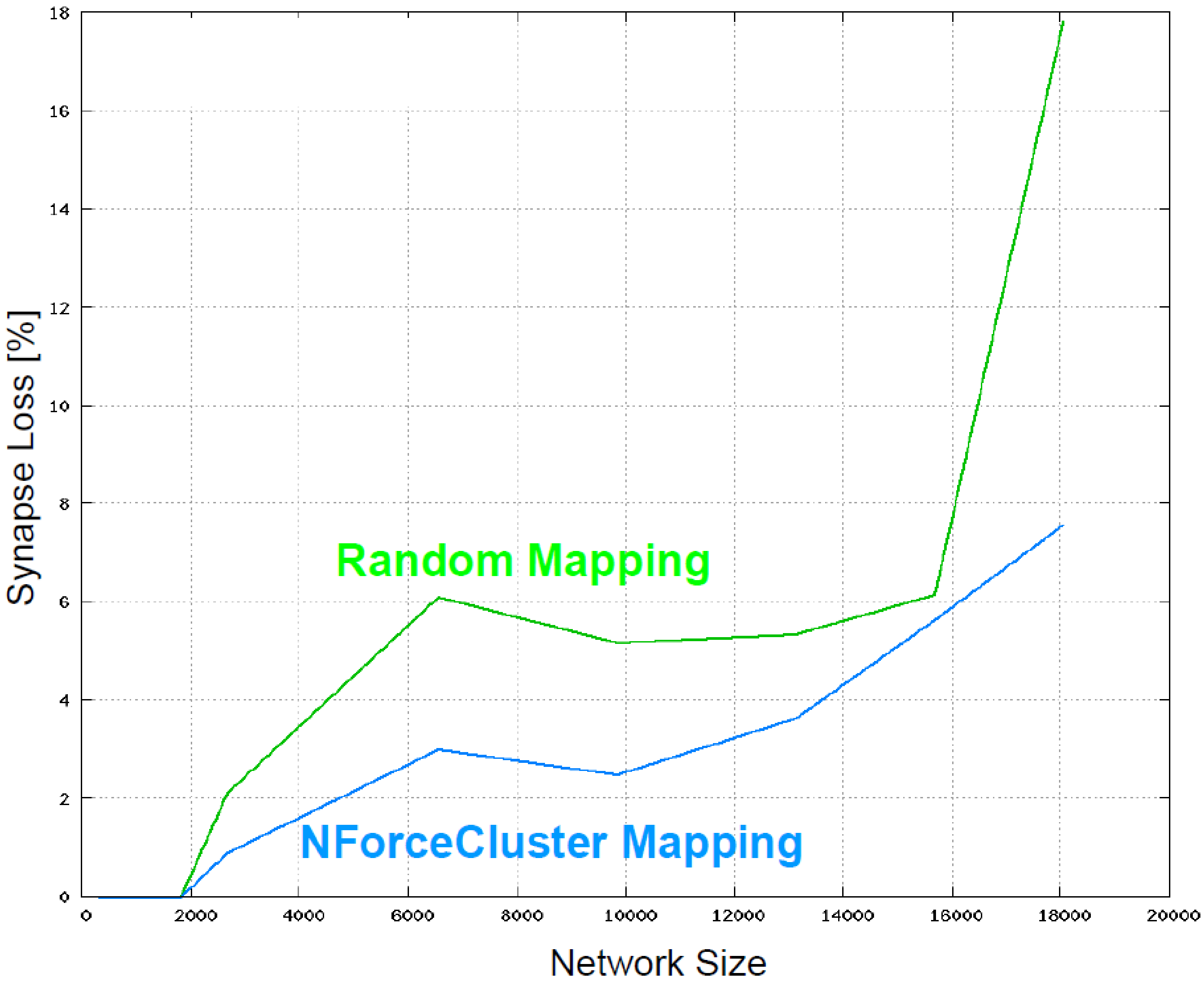


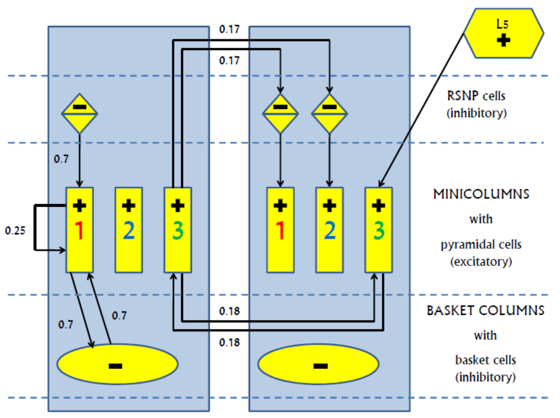
Neuron Placing **Not** optimized



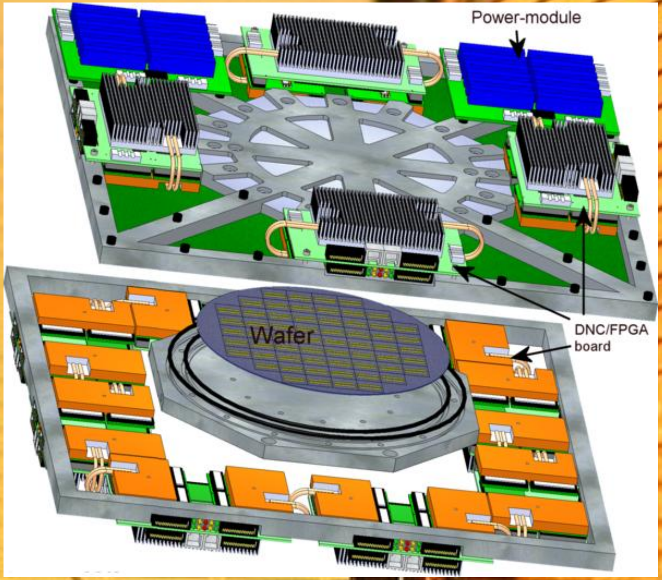
Neuron Placing Optimized

Mapping algorithm performance





?

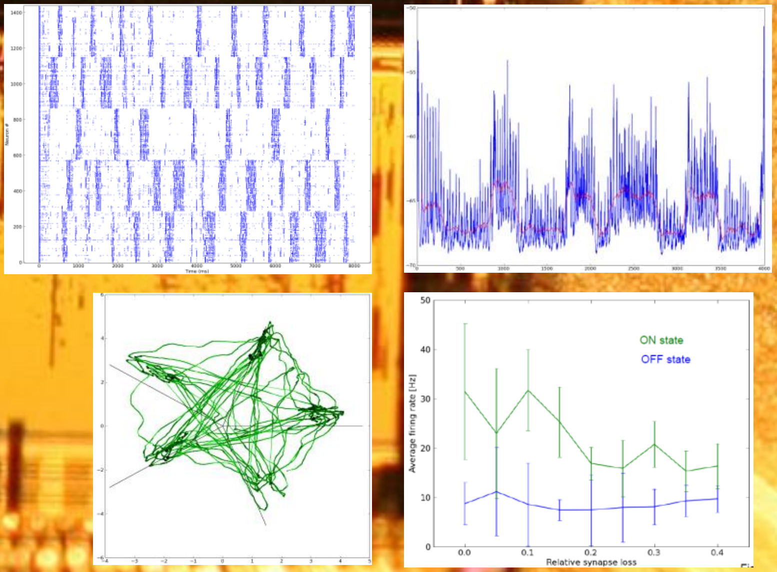


?

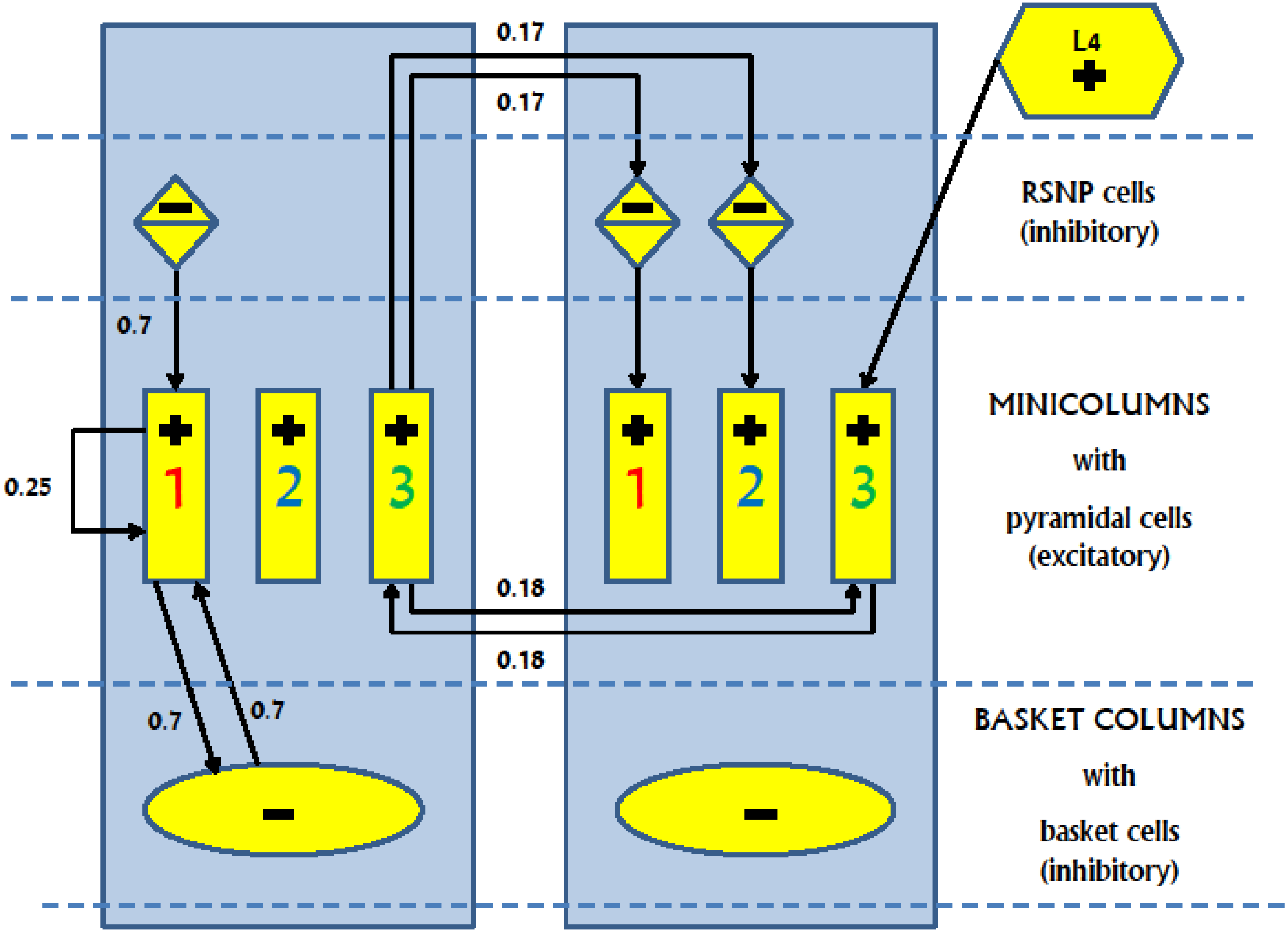
PART II (B)

WORKFLOW:

DISTORTION EVALUATION AND COMPENSATION



Attractor memory schematic

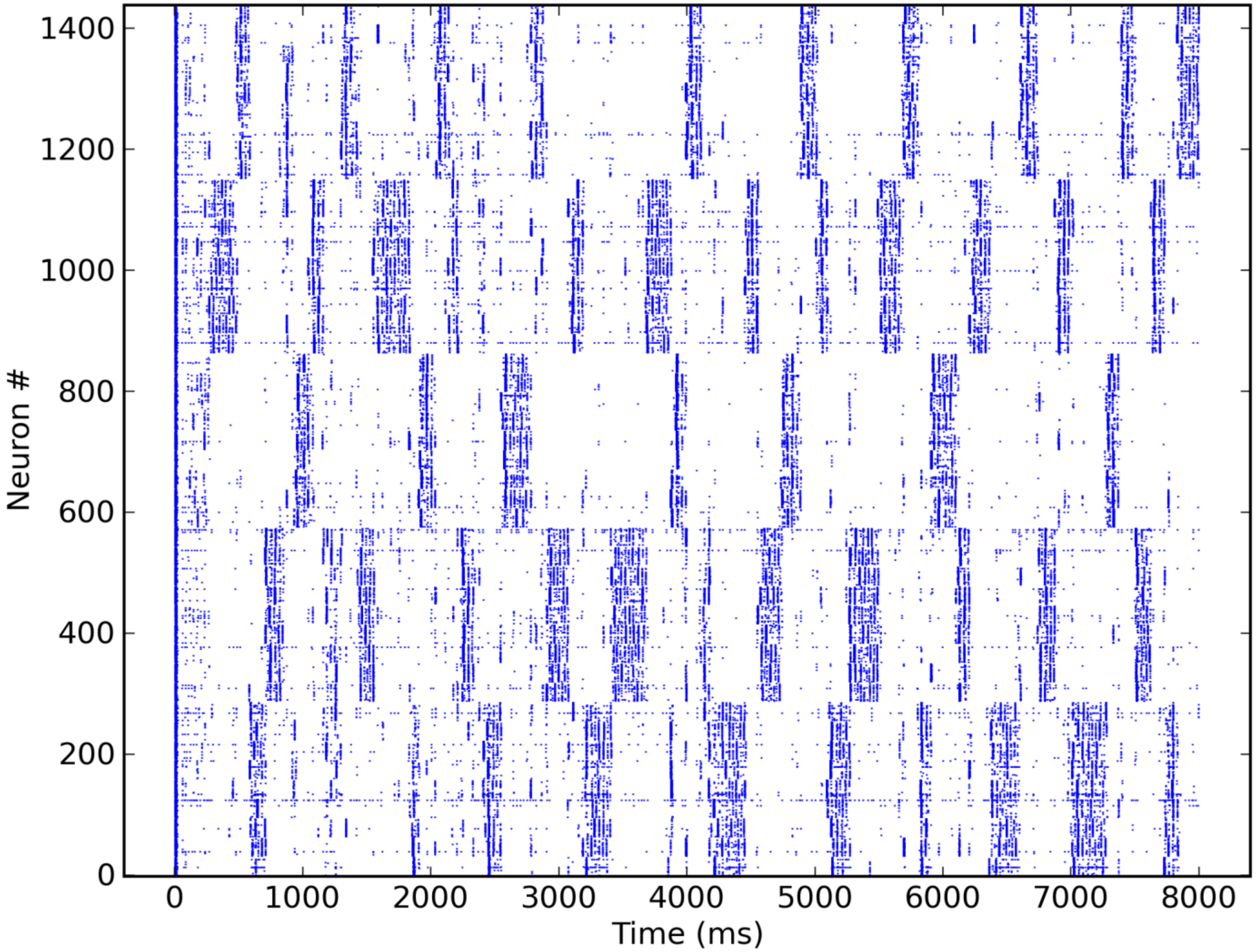


RSNP cells
(inhibitory)

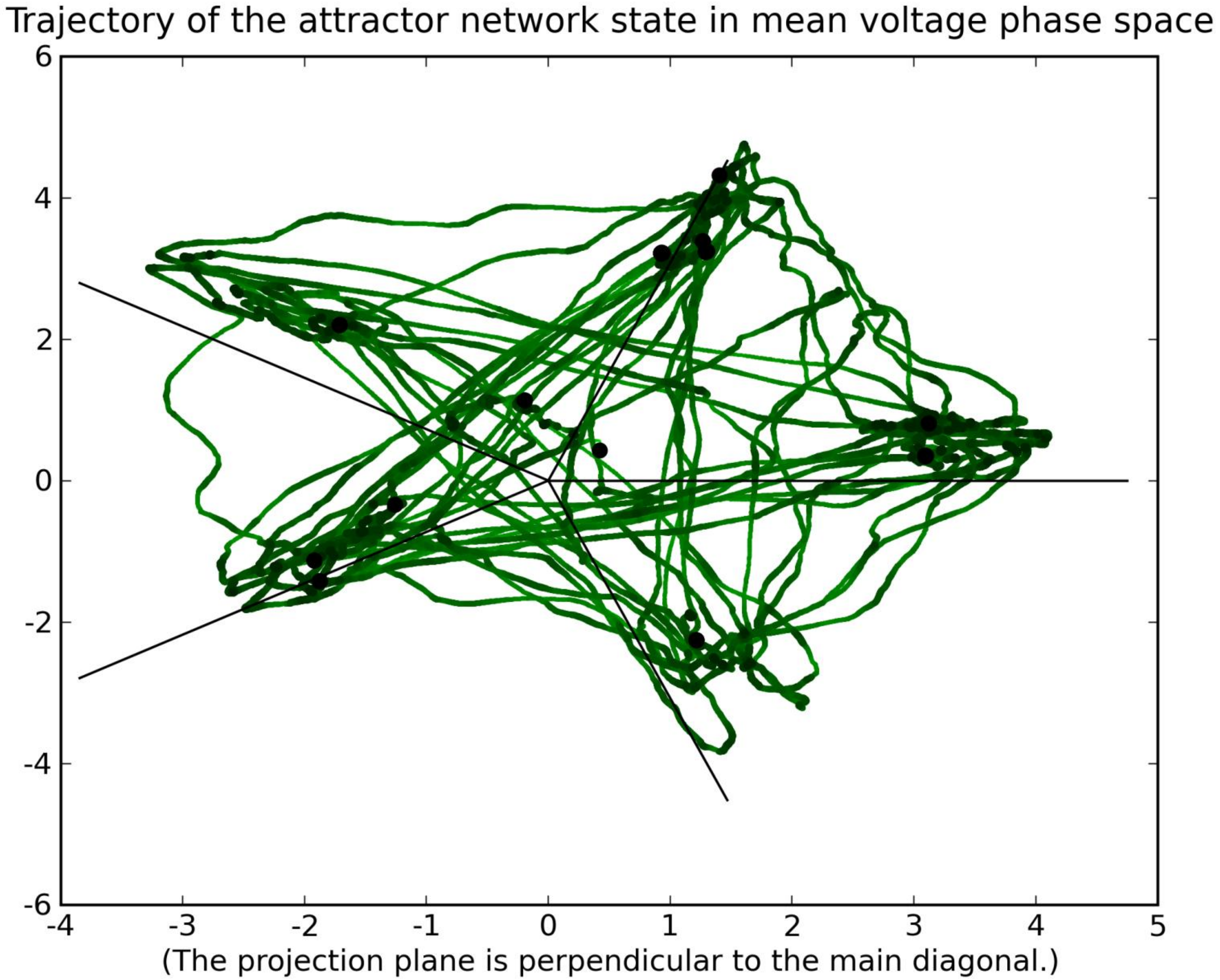
MINICOLUMNS
with
pyramidal cells
(excitatory)

BASKET COLUMNS
with
basket cells
(inhibitory)

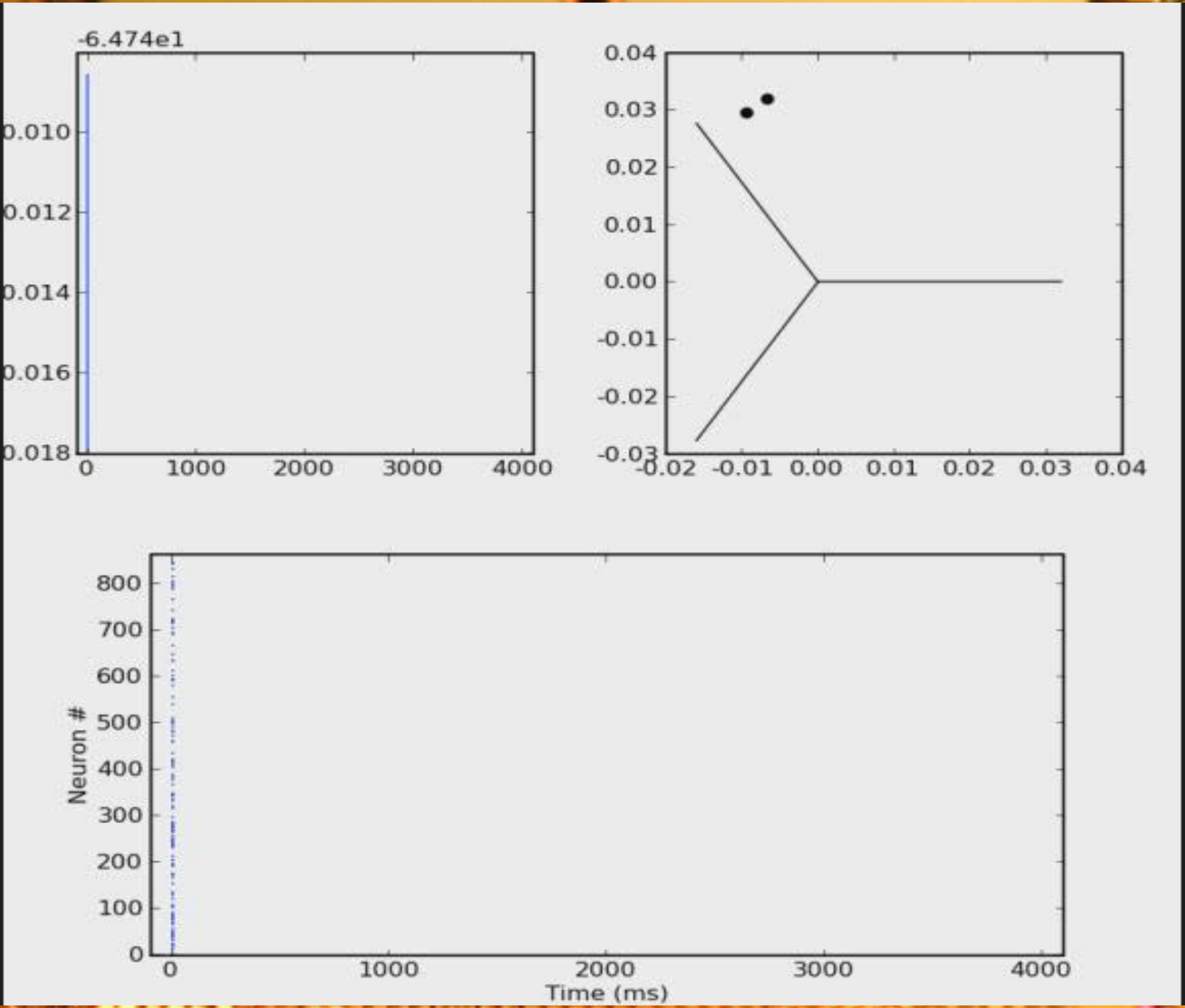
Spiking patterns



Trajectories in voltage space



Network dynamics



Network dynamics

Motivation

- hardware imperfections
- nonisomorphic simulation/emulation environments
  e.g. neuron model, digitized weights, …
- mapping/routing losses

robustness is an essential characteristic of biological neural networks
→ hardware independent research

Relevant parameters

model-independent

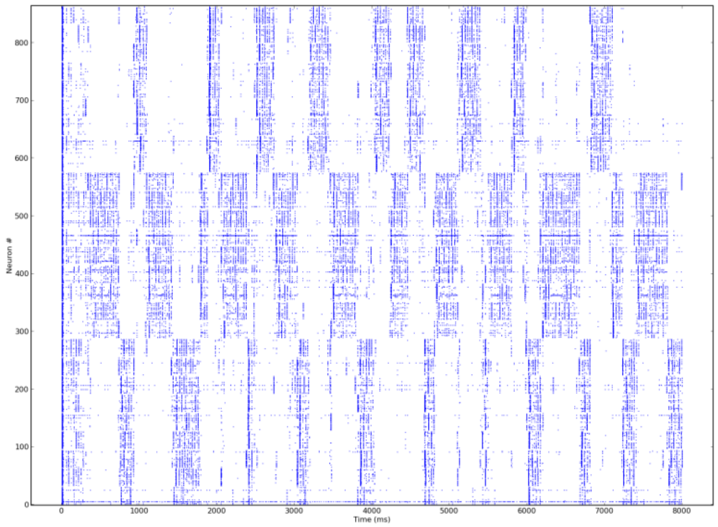
- STP
- adaptation
- delays
- synaptic weights
- neuron loss
- synapse loss

model-specific

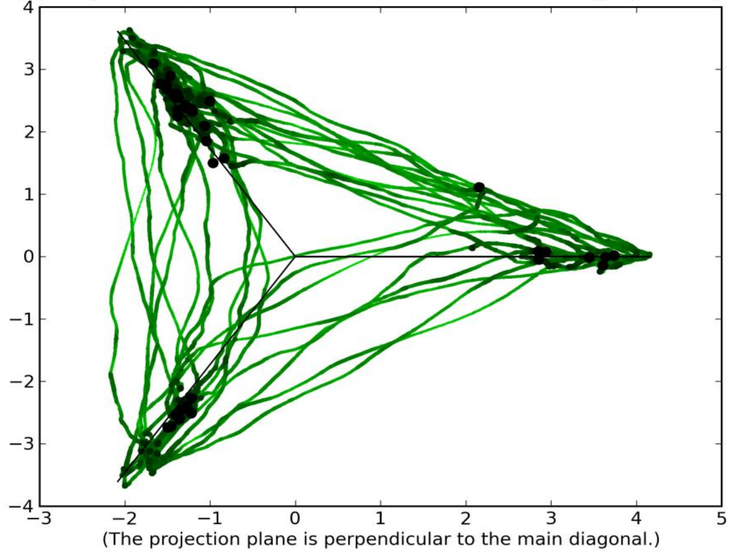
- number of MC per HC
- number of HC
- total number of MC
  (≡ network size)

The importance of STP

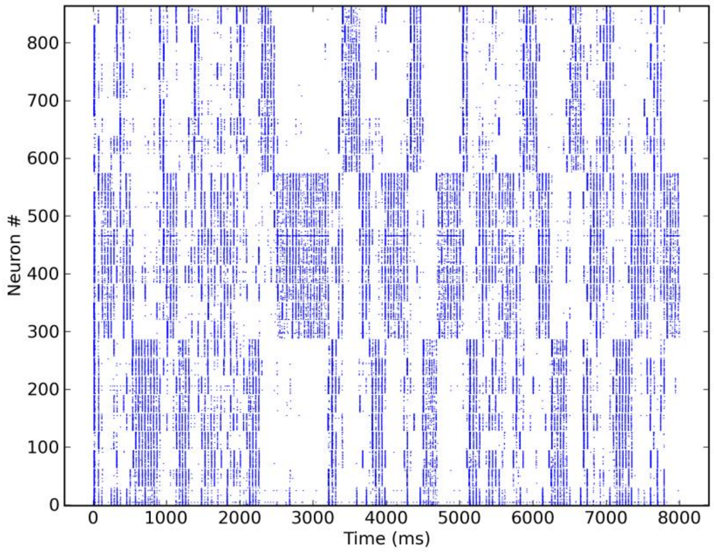
with STP (Poisson input: 4 kHz)



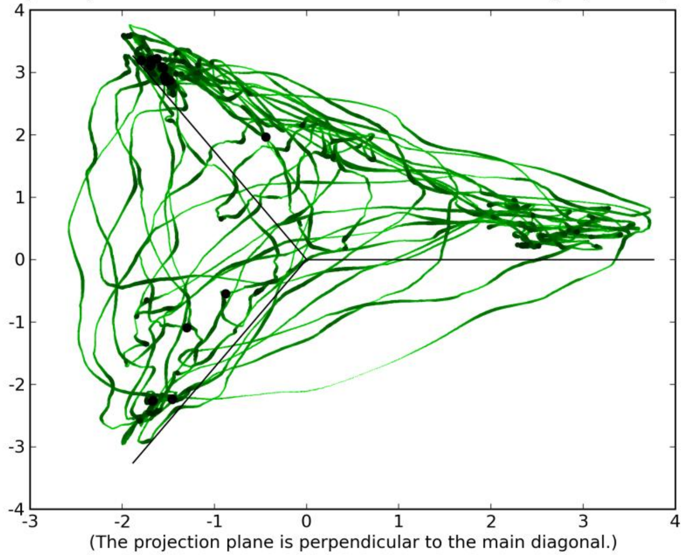
Trajectory of the attractor network state in mean voltage phase space



without STP (Poisson input: 1 kHz)



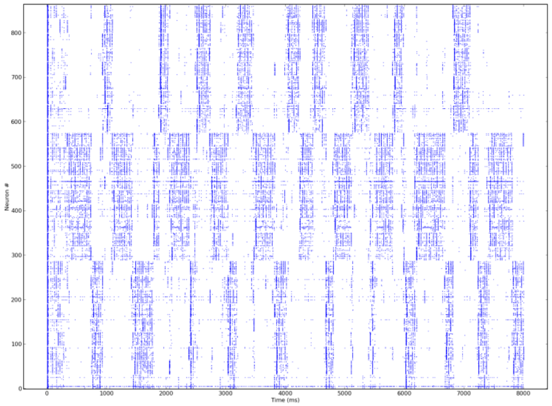
Trajectory of the attractor network state in mean voltage phase space

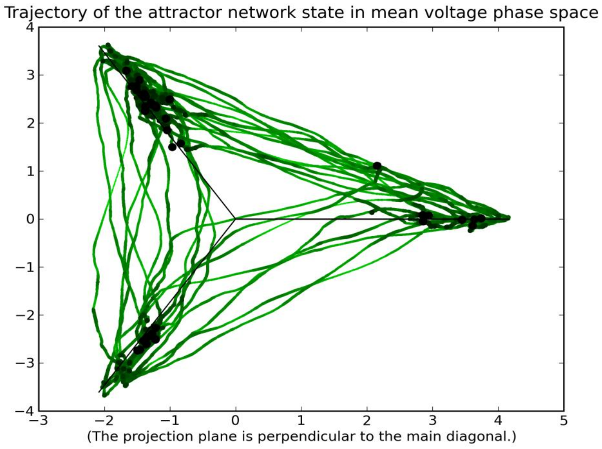


The importance of adaptation and delays

+ adaptation + delays

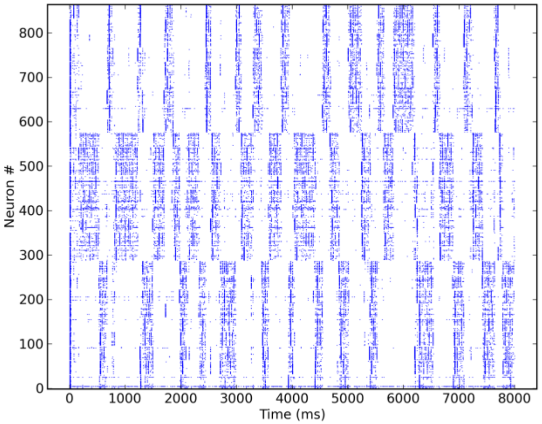
mean firing rate in ON state:
30 Hz

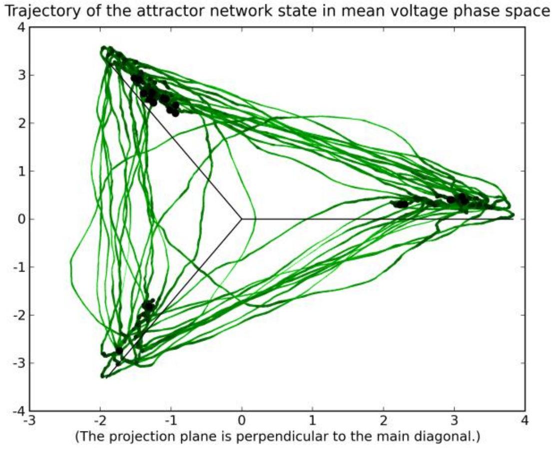



Trajectory of the attractor network state in mean voltage phase space

+ adaptation - delays

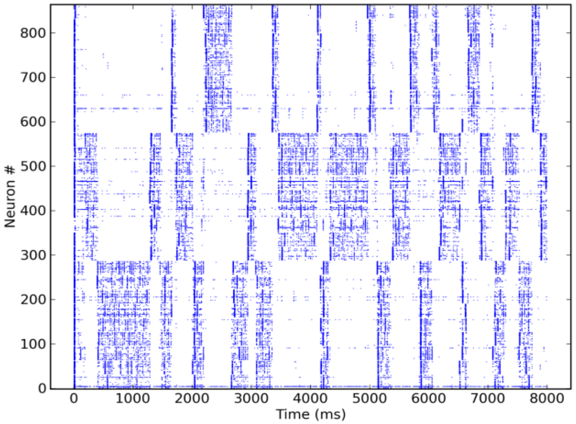
mean firing rate in ON state:
28 Hz

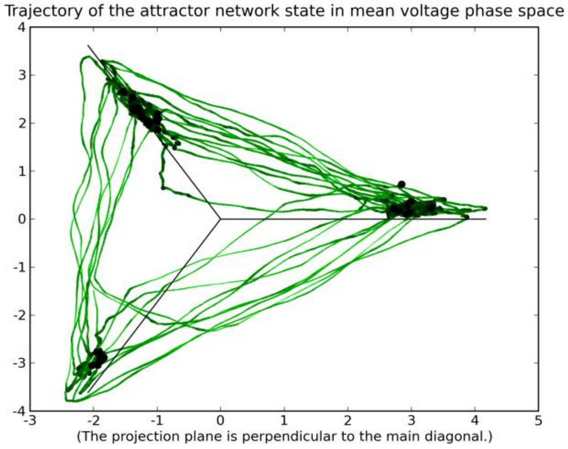



Trajectory of the attractor network state in mean voltage phase space

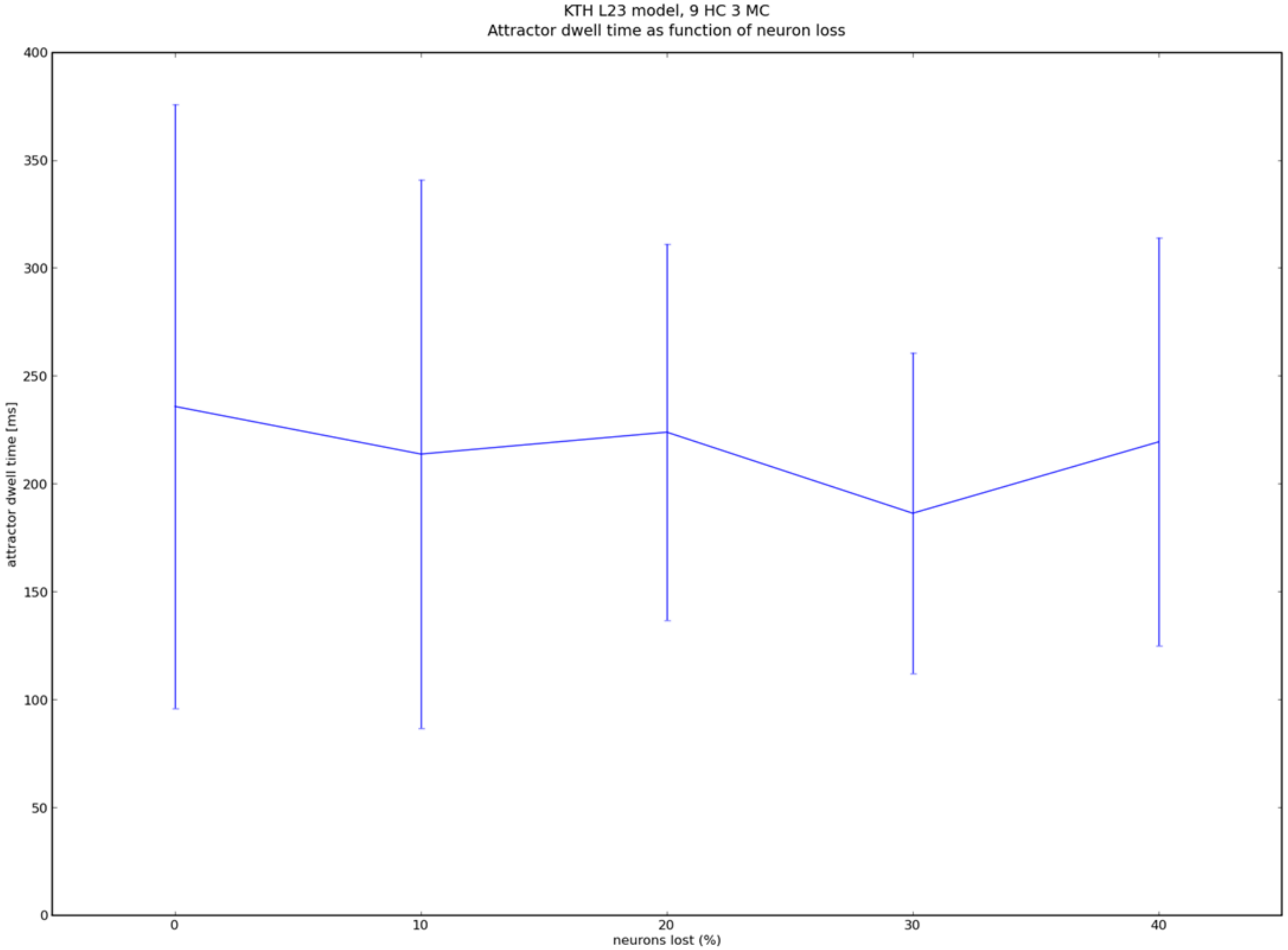
- adaptation - delays

mean firing rate in ON state:
116 Hz

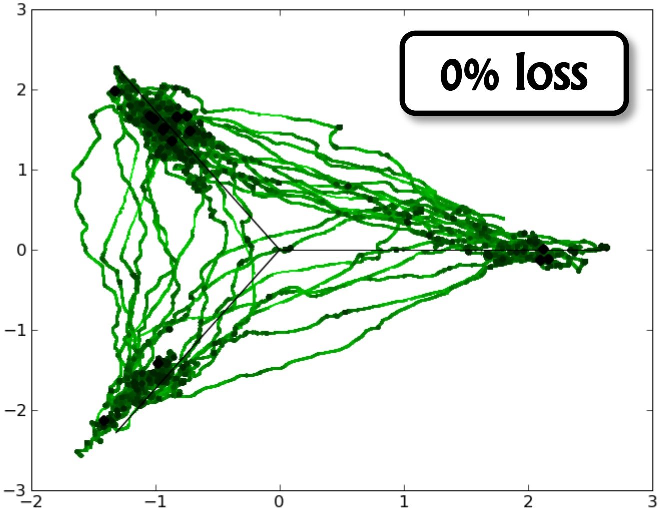



Trajectory of the attractor network state in mean voltage phase space

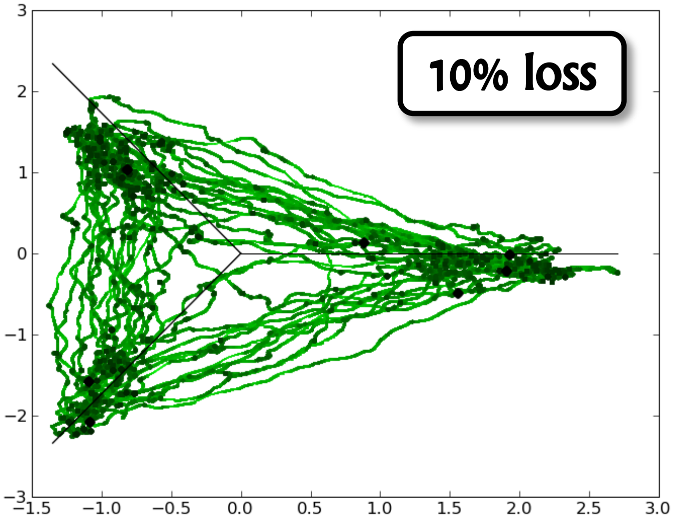
Dwell times and neuron loss



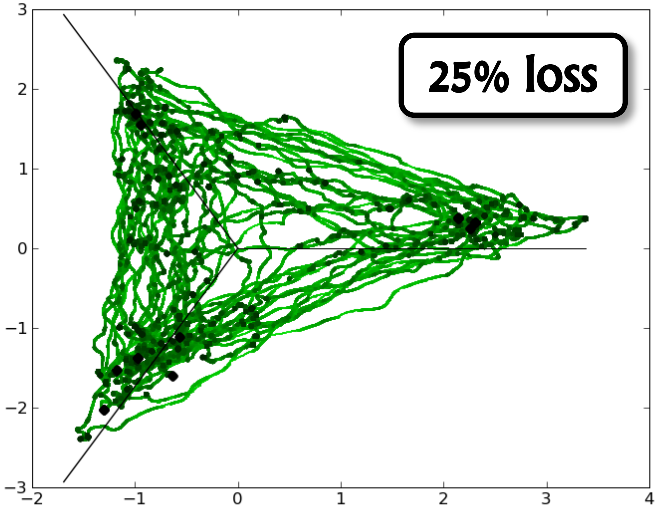
Synapse loss



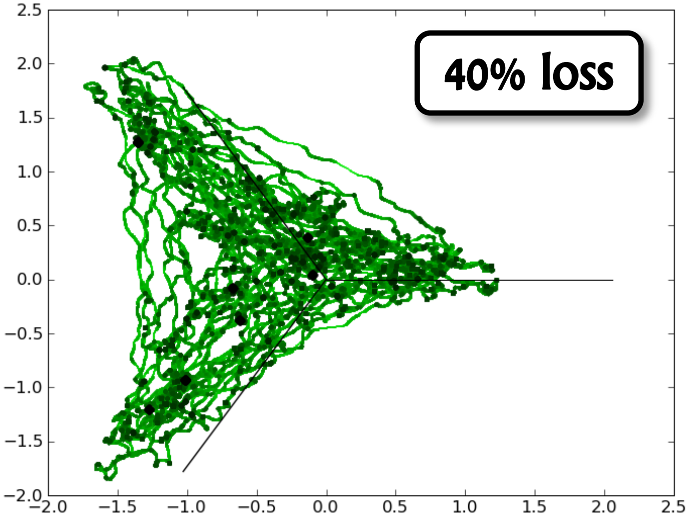
0% loss



10% loss

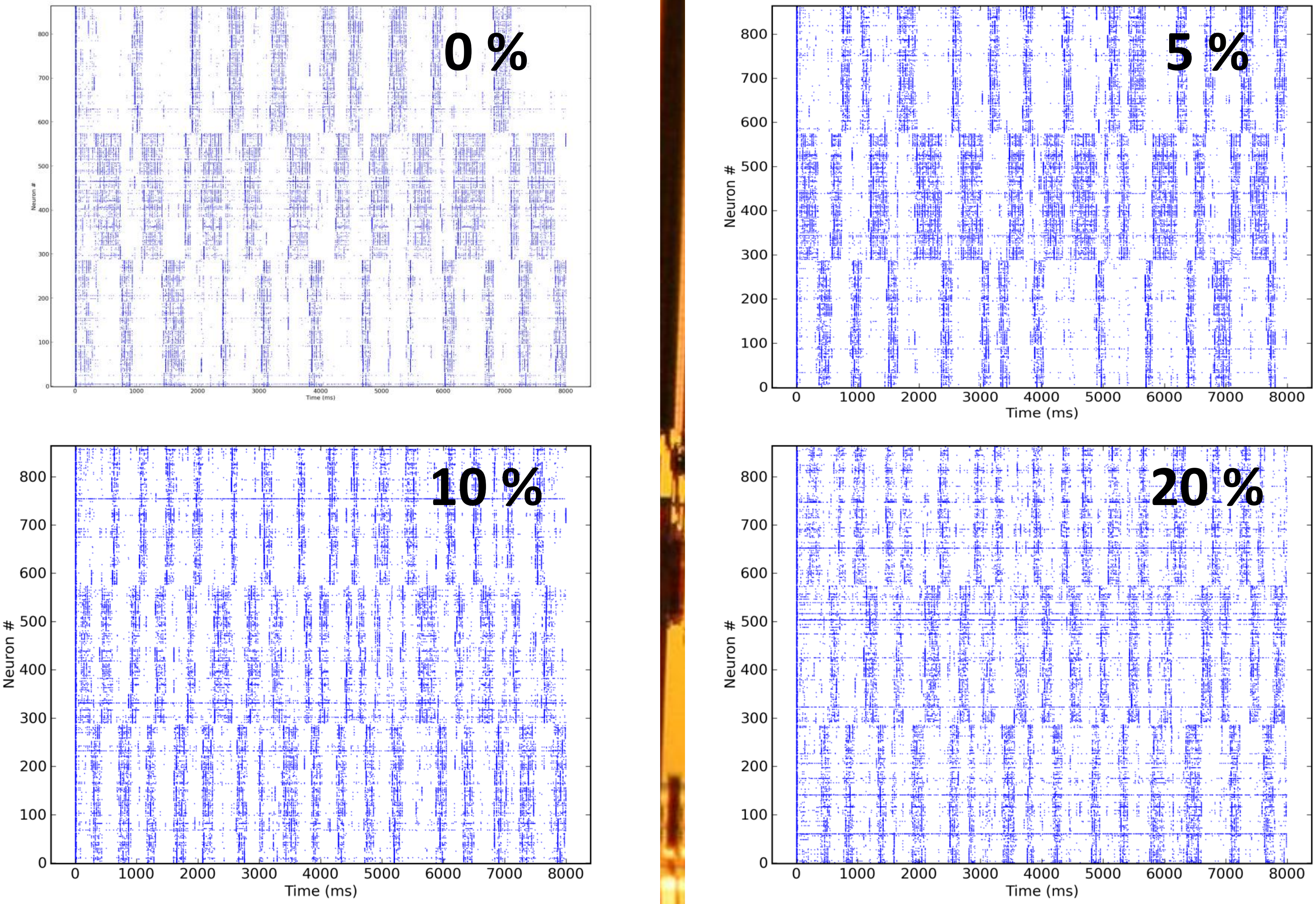


25% loss

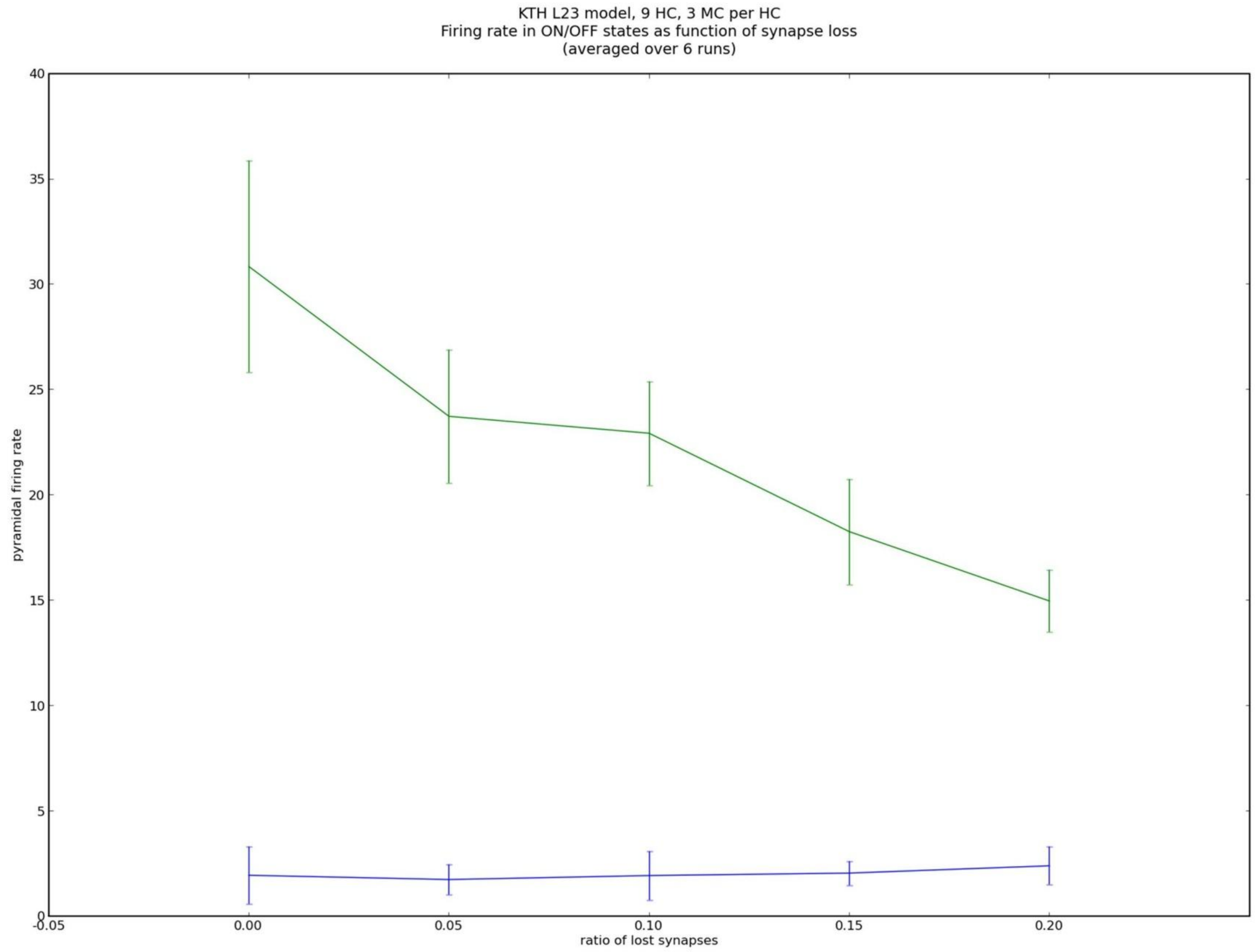


40% loss

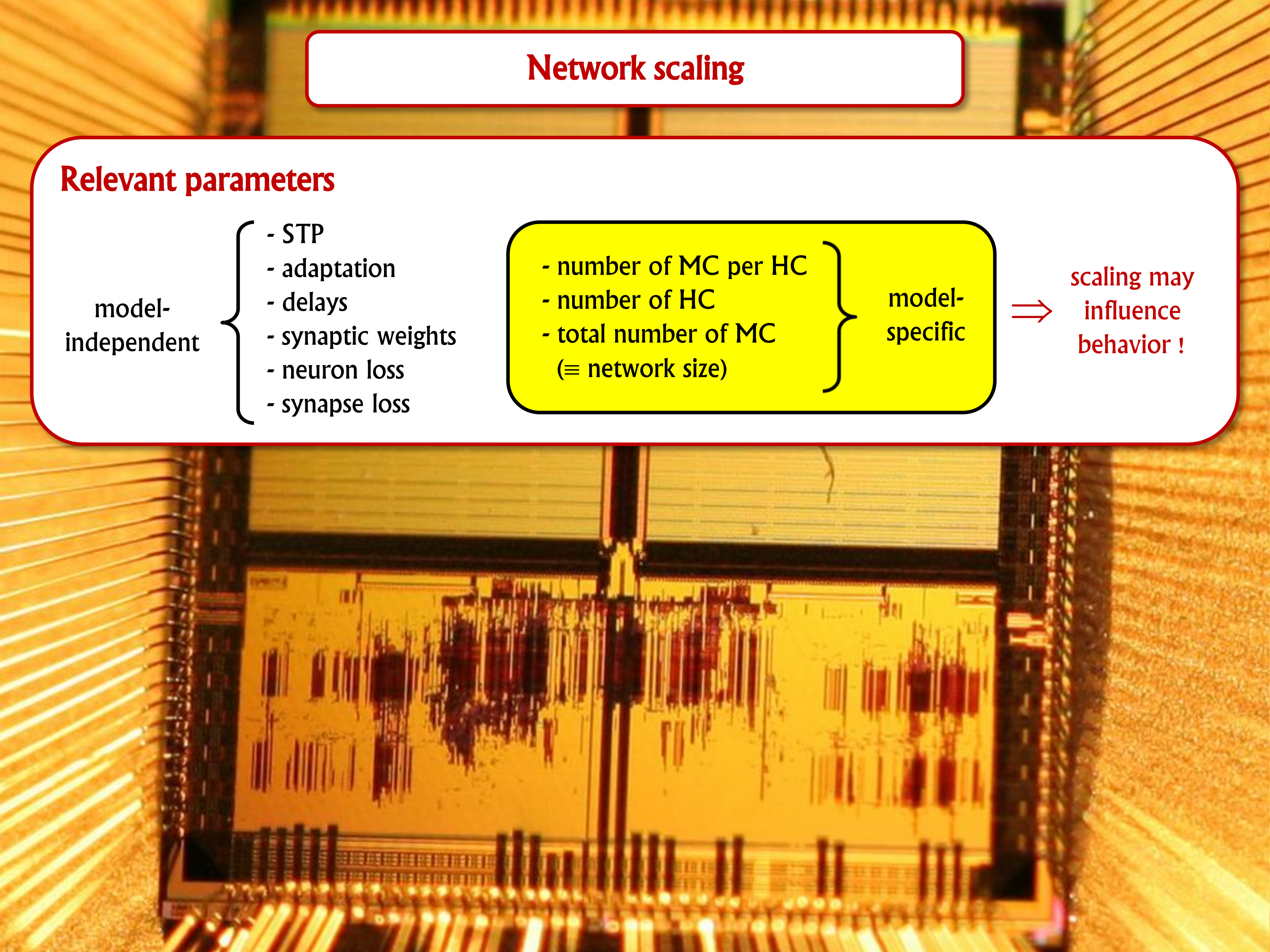
Dwell times and synapse loss



Firing rates and synapse loss



Network scaling



Relevant parameters

model-independent

- STP
- adaptation
- delays
- synaptic weights
- neuron loss
- synapse loss

- number of MC per HC
- number of HC
- total number of MC
  (≡ network size)

model-specific

⇒ scaling may influence behavior !

Network scaling

Relevant parameters

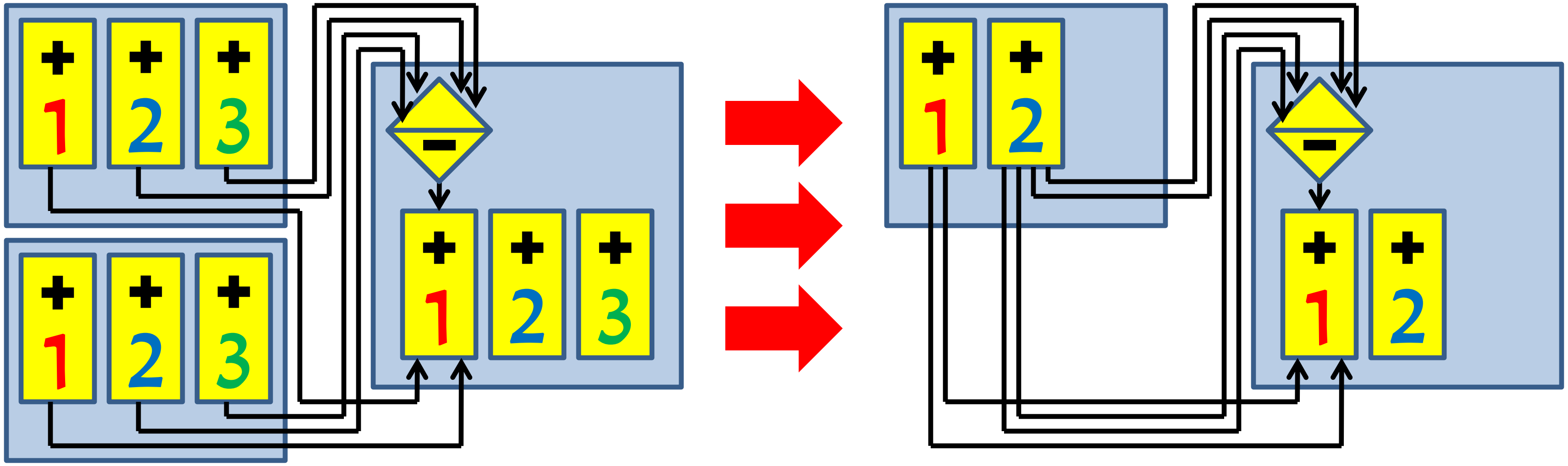
model-independent
- STP
- adaptation
- delays
- synaptic weights
- neuron loss
- synapse loss

- number of MC per HC
- number of HC
- total number of MC (≡ network size)

model-specific

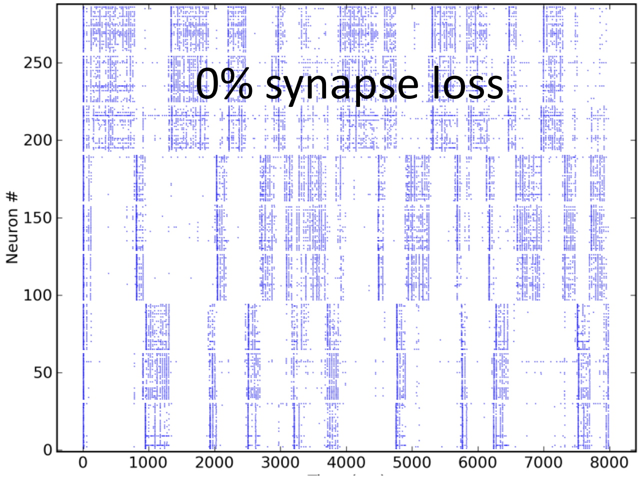
⇒ scaling may influence behavior !

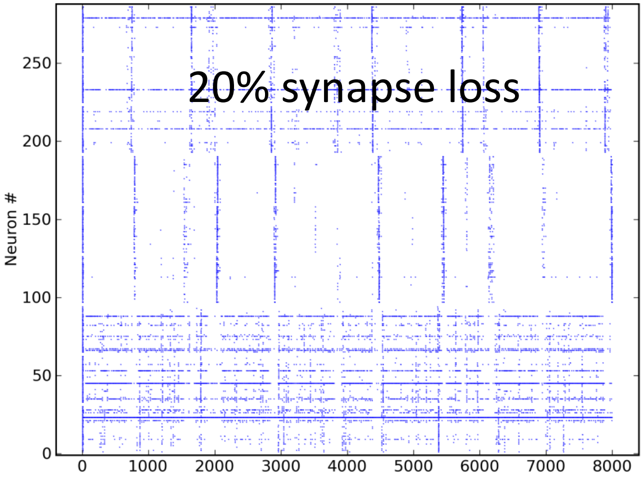
Scaling through modification of connection probabilities



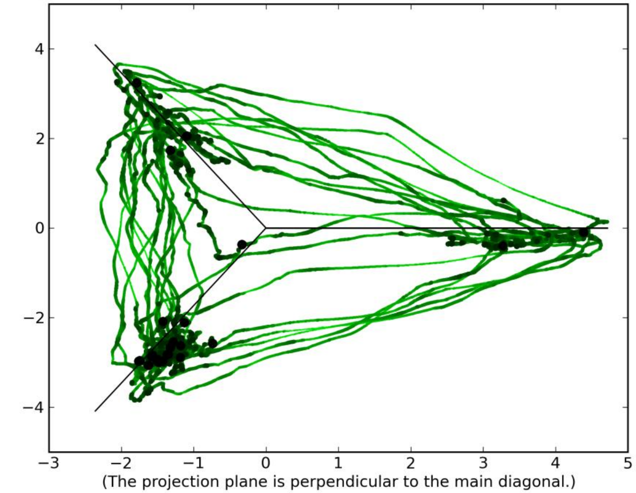
Scaling and robustness

3 HC 3 MC

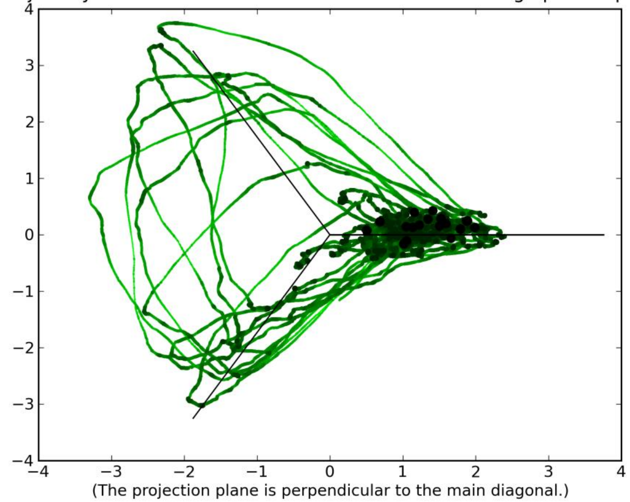




Trajectory of the attractor network state in mean voltage phase space



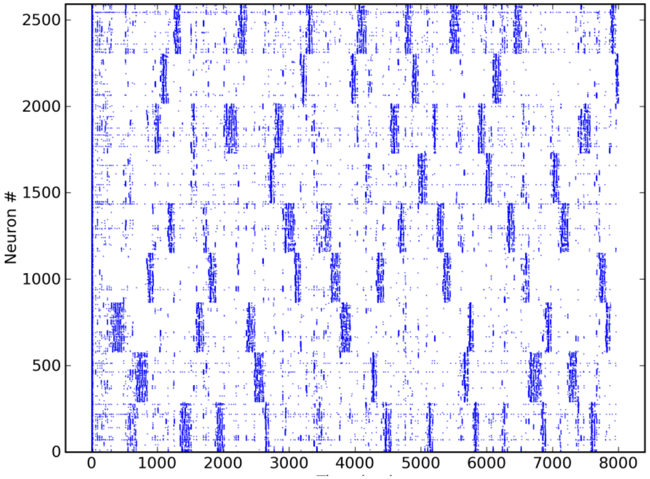
Trajectory of the attractor network state in mean voltage phase space



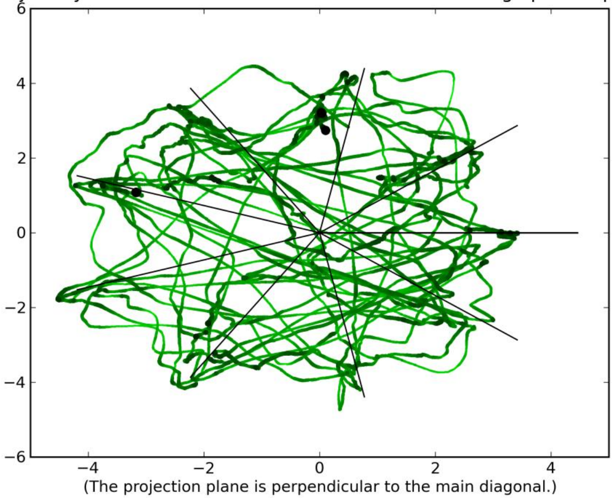
Scaling and robustness

9 HC 9 MC

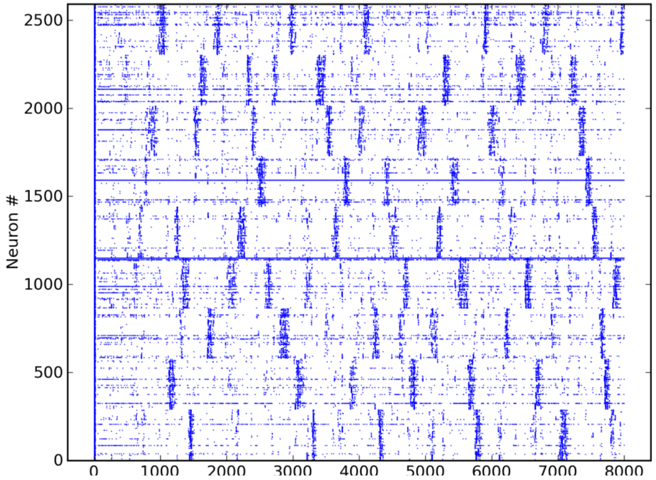
0% synapse loss



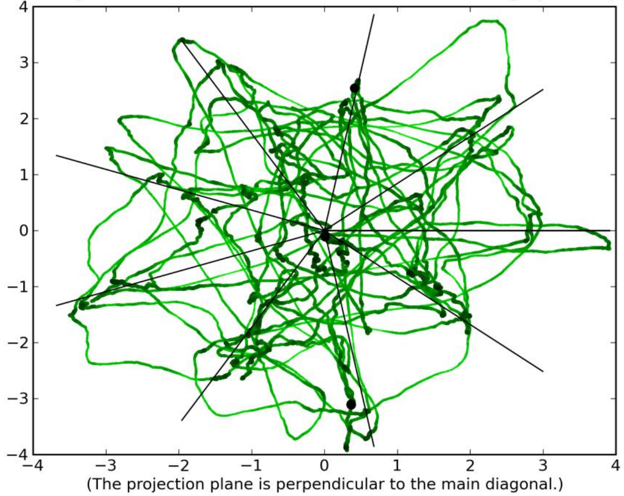
Trajectory of the attractor network state in mean voltage phase space



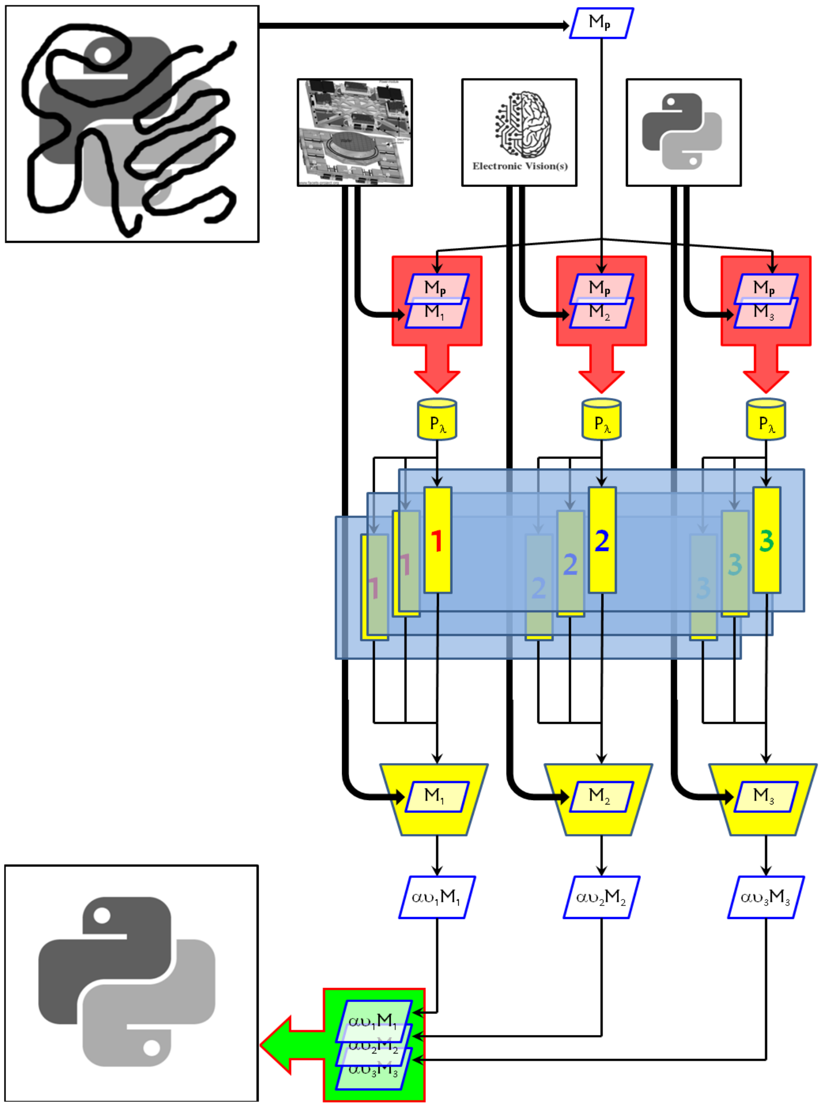
20% synapse loss



Trajectory of the attractor network state in mean voltage phase space

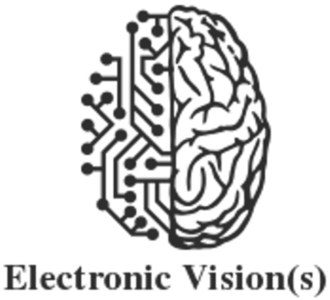


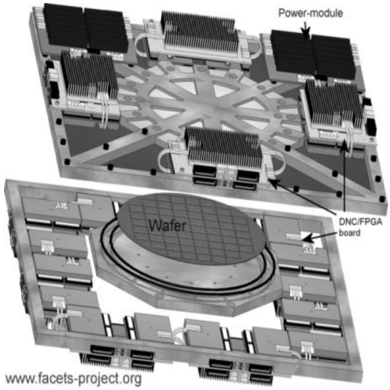
Pattern completion



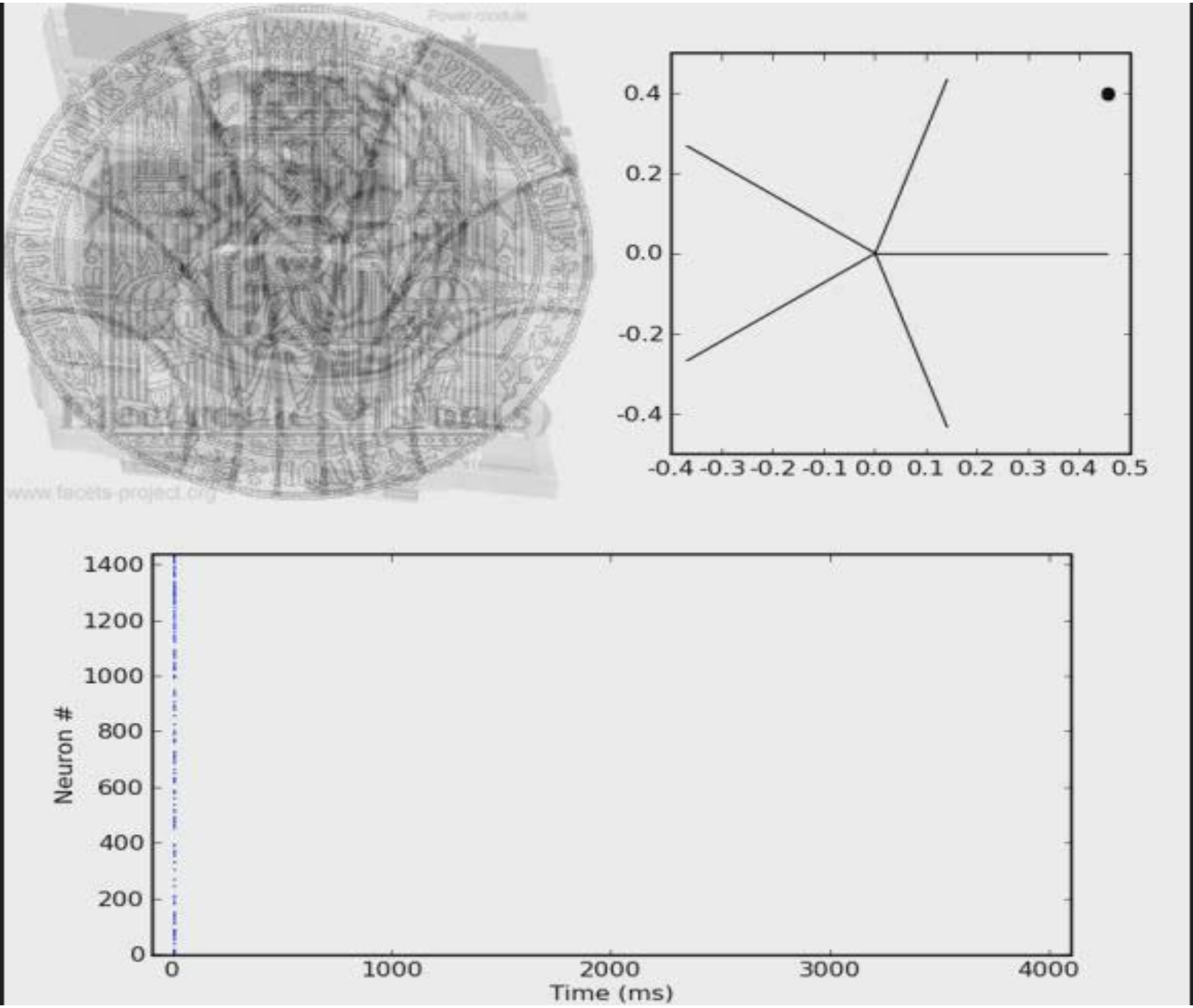
stored

images



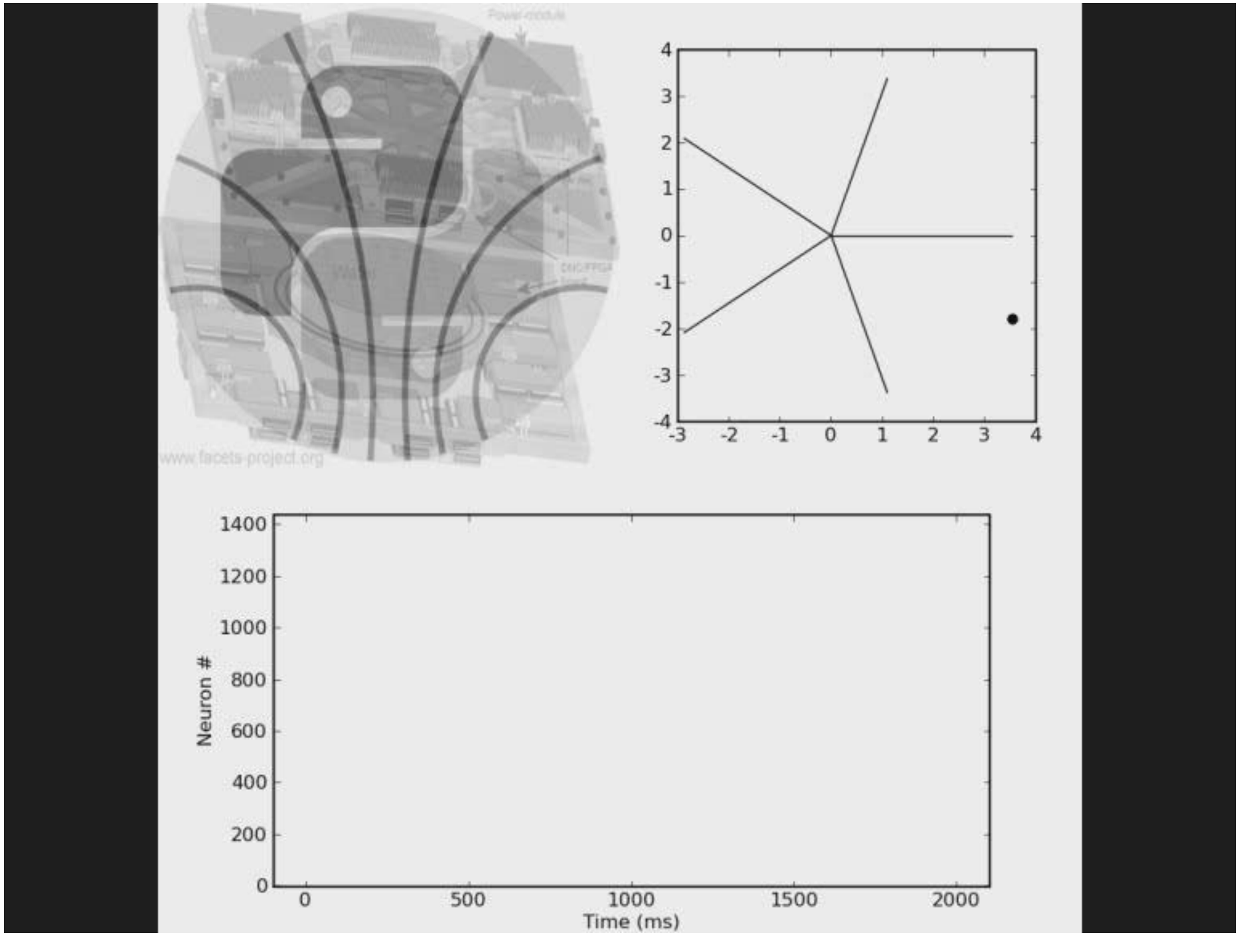


Spontaneous pattern generation



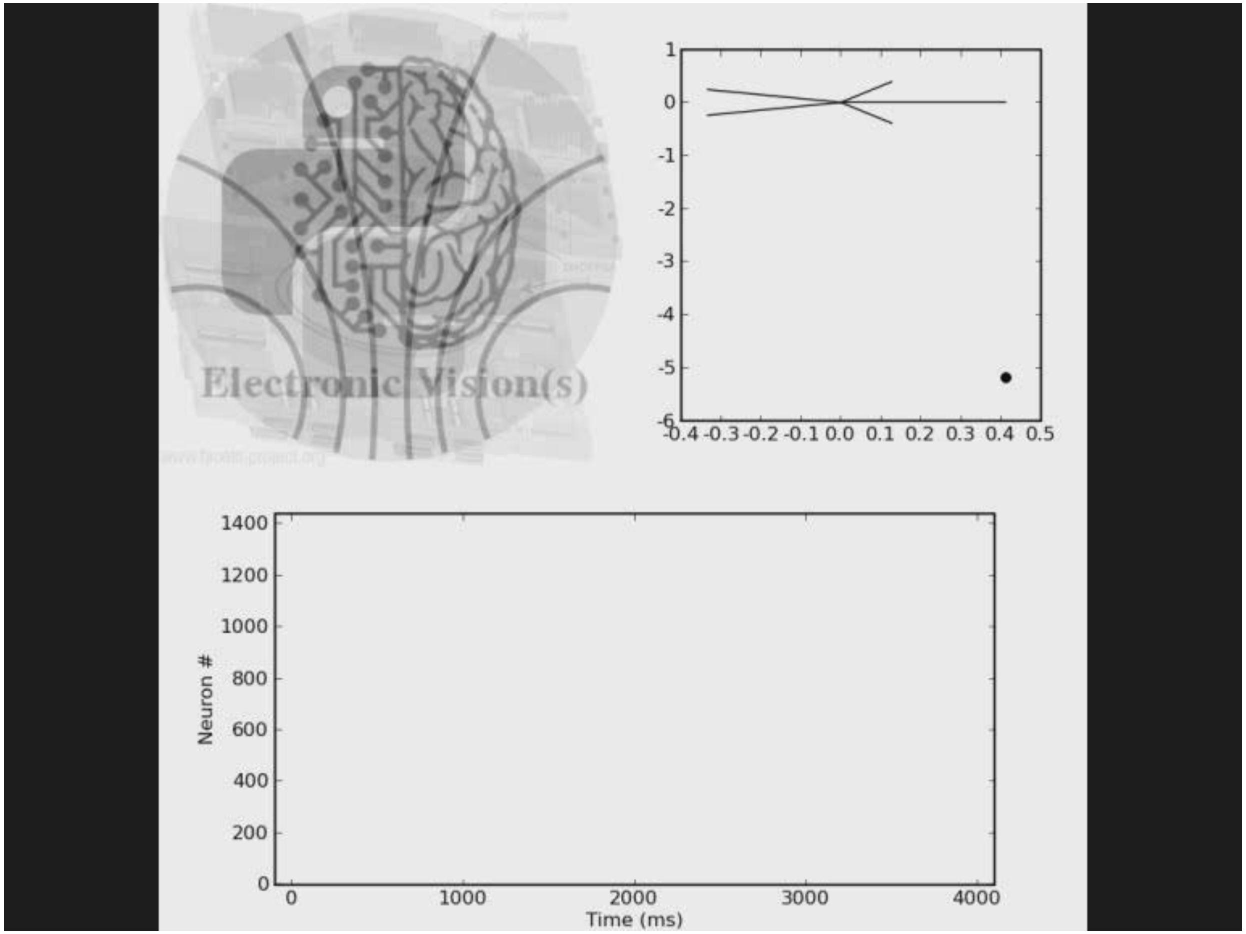
Pattern completion: small distortion

input image



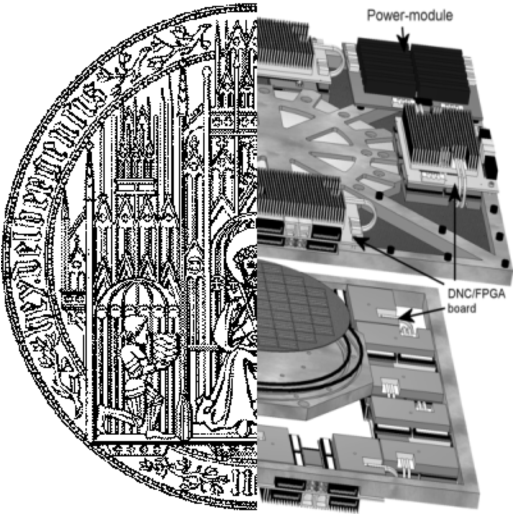
Pattern completion: large distortion

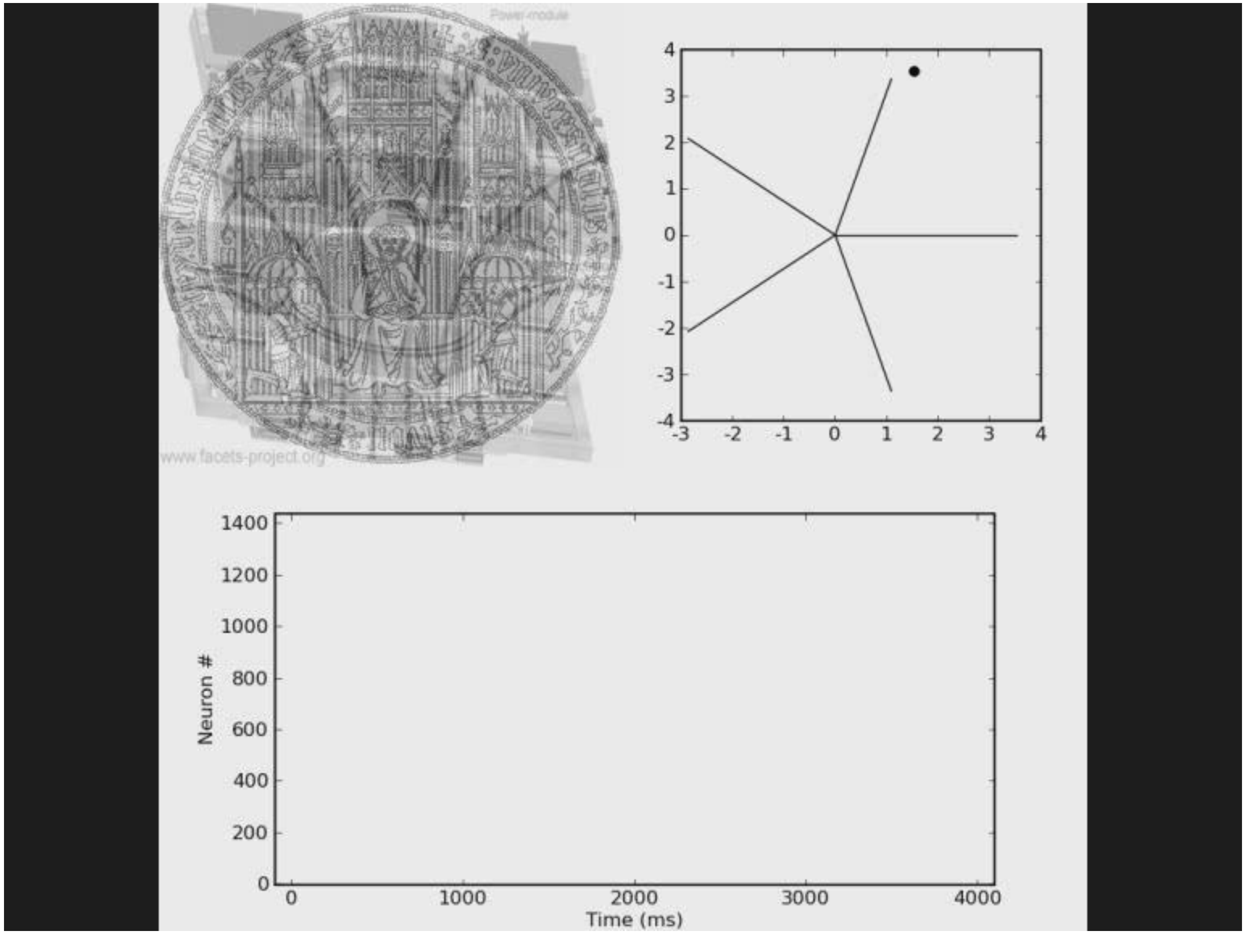
input image



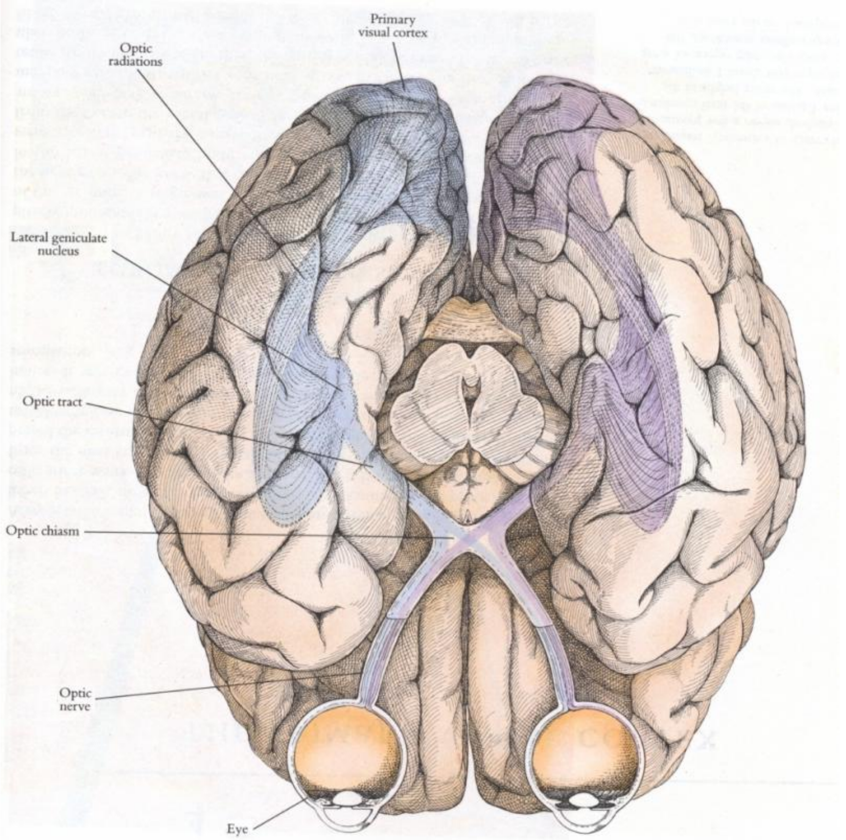
Pattern completion: two patterns

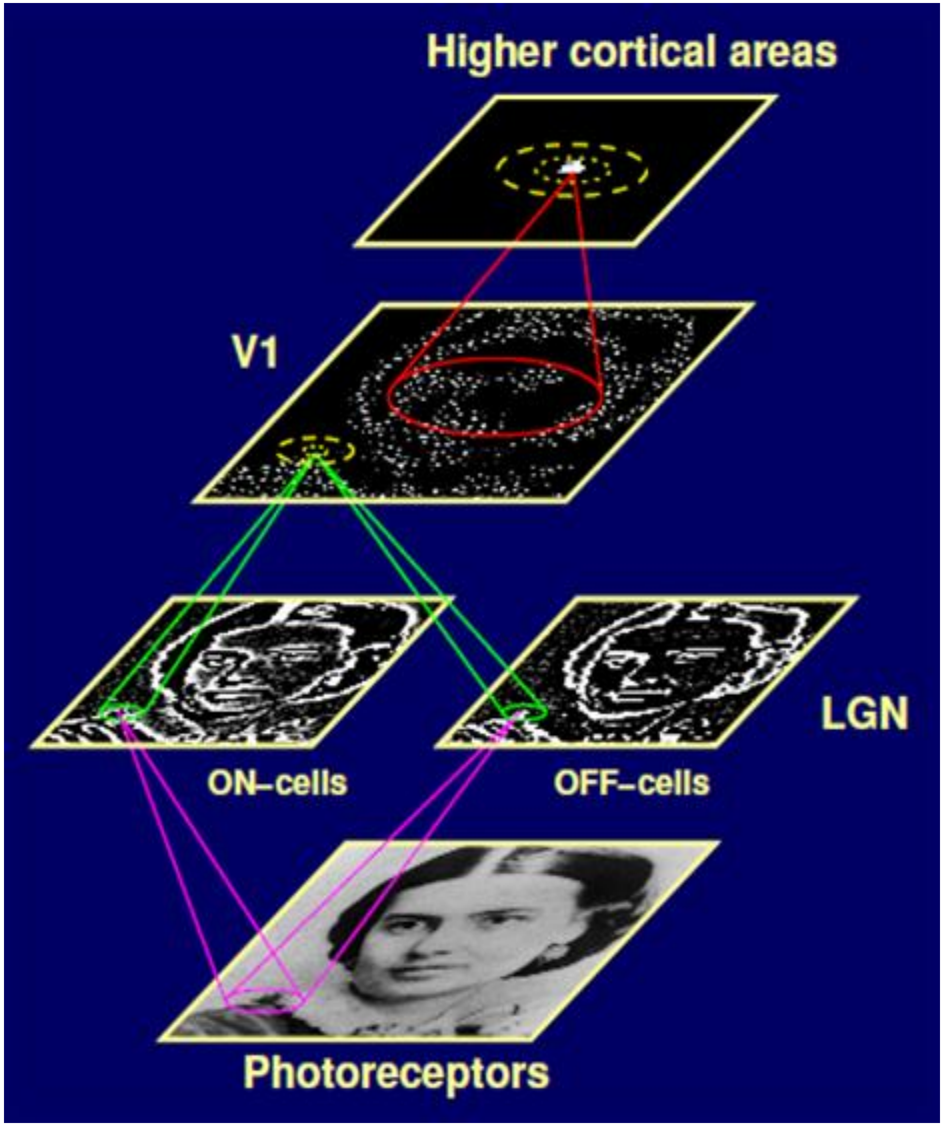
input image



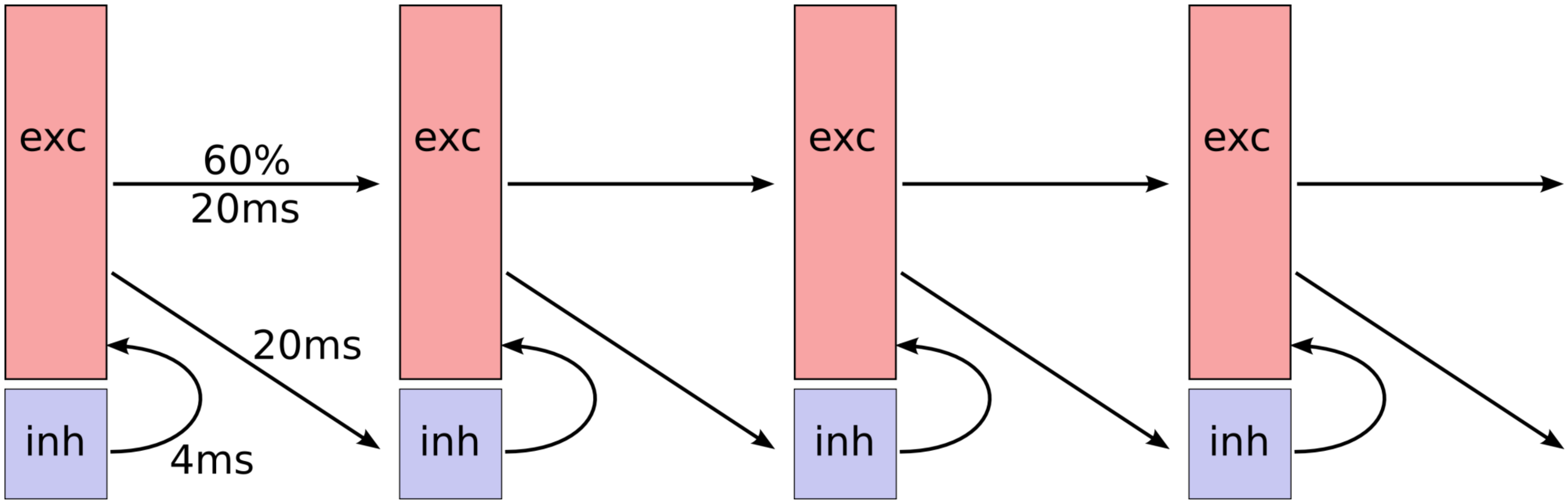


Pattern completion: a more biological approach





Synfire chain schematic

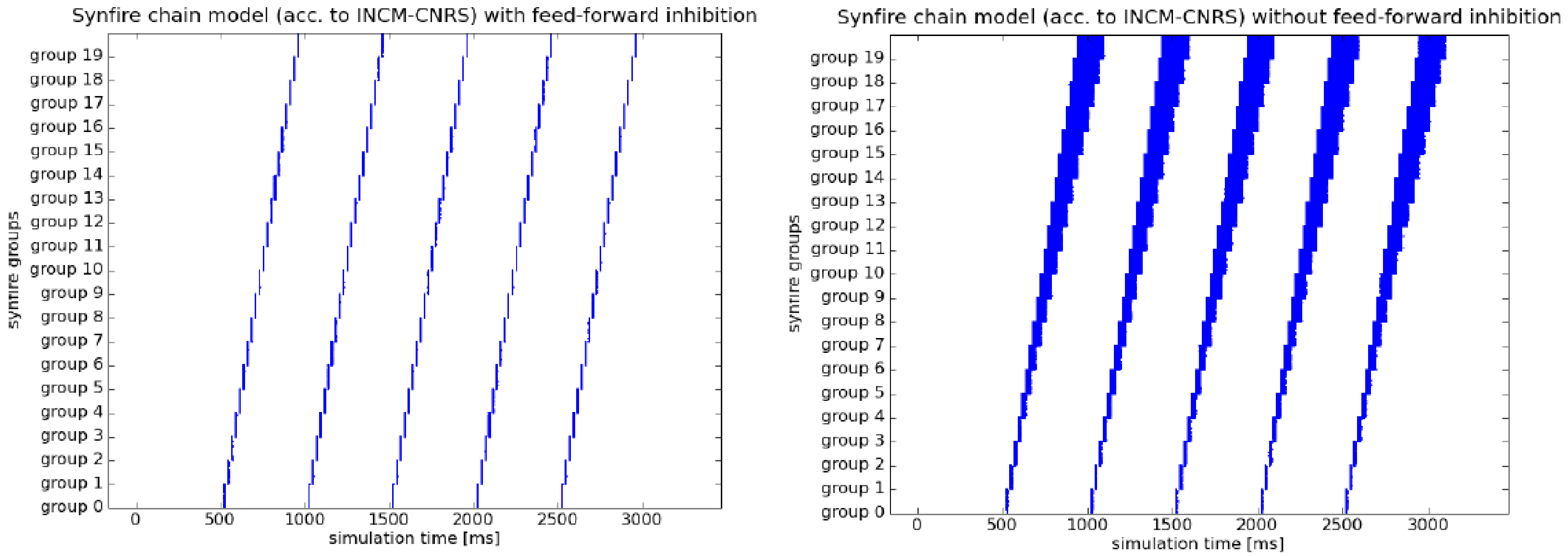


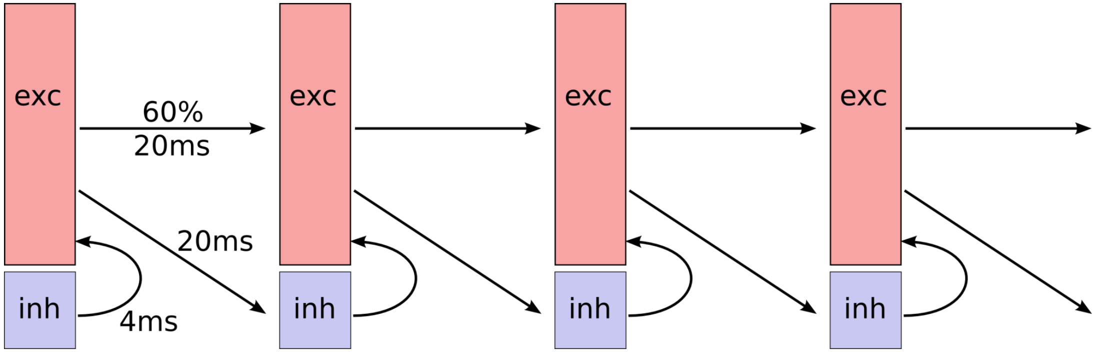
exc 100 regular spiking neurons

inh 25 fast spiking neurons

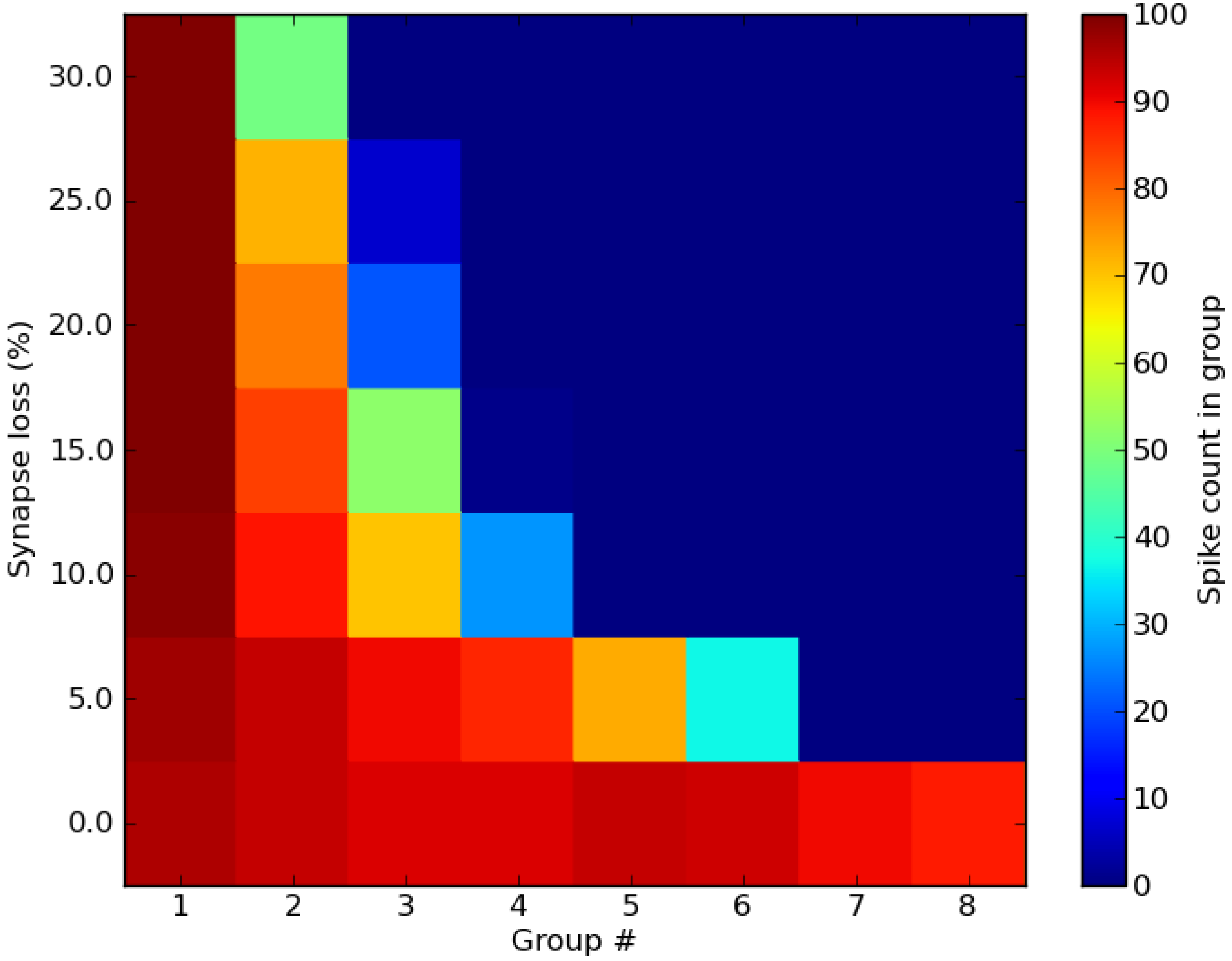
} same parameters in our model

Synfire chain simulations

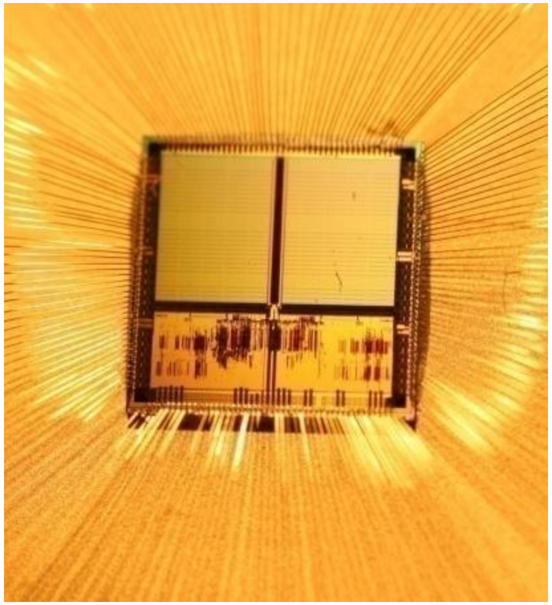




Synapse loss

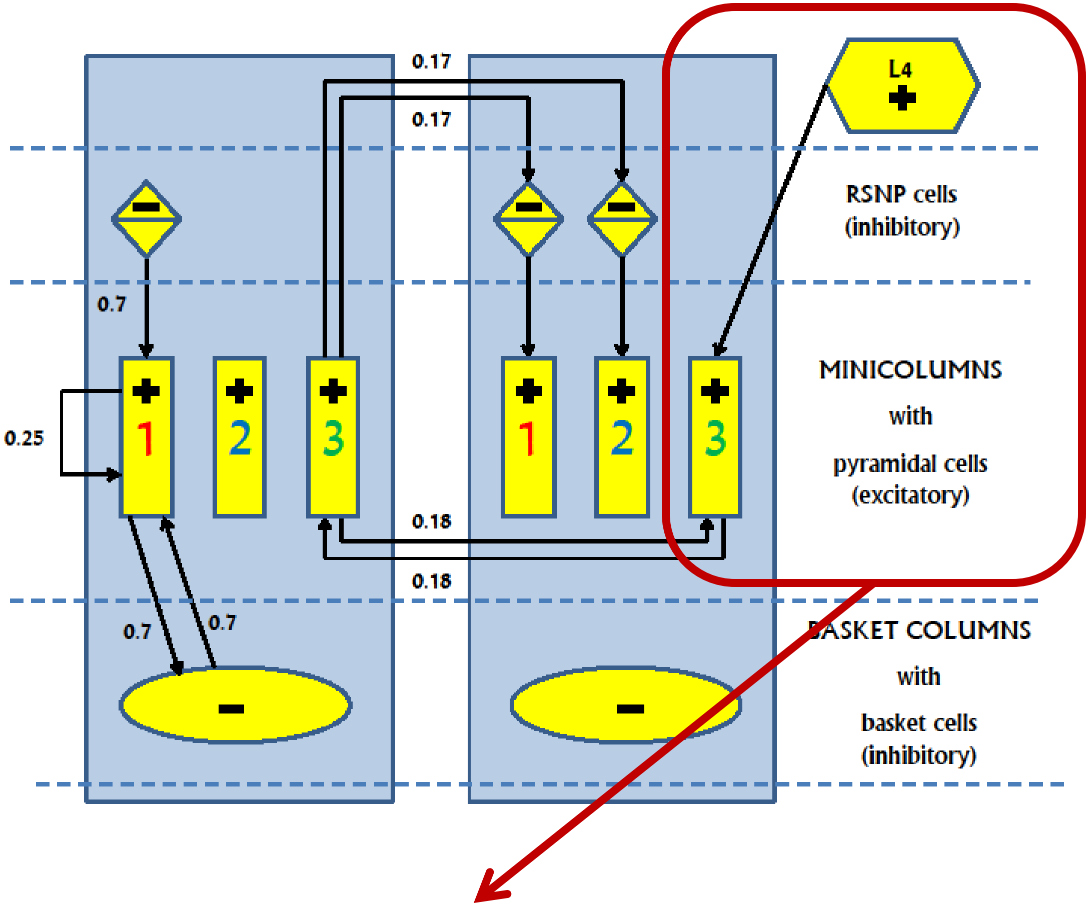


The problem of limited input



only 64 external inputs
with max. 100 Hz / channel

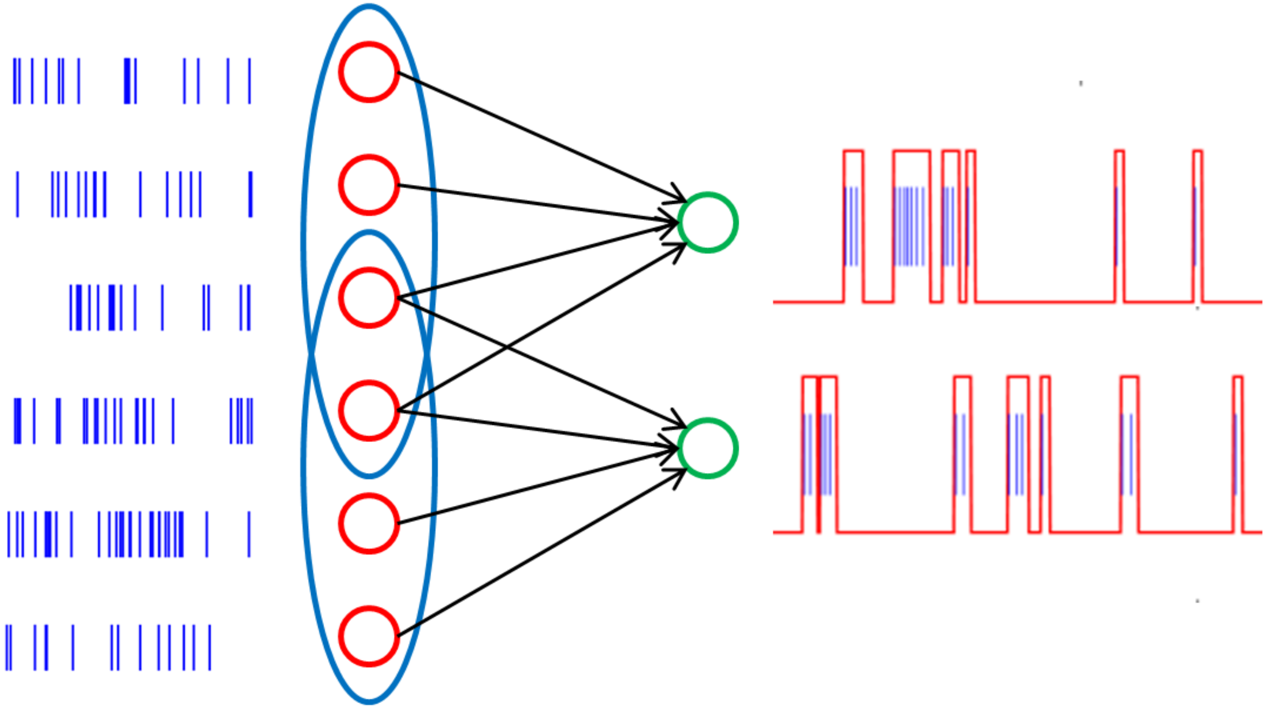
for 192 neurons



4000 Hz independent
Poisson input

per neuron

The problem of limited input



Problem II

Problem I

given a limited set of input channels and a minimum requirement for inputs per neuron, can we find a corresponding mapping ?

how to quantify and predict correlations which arise from shared inputs ?

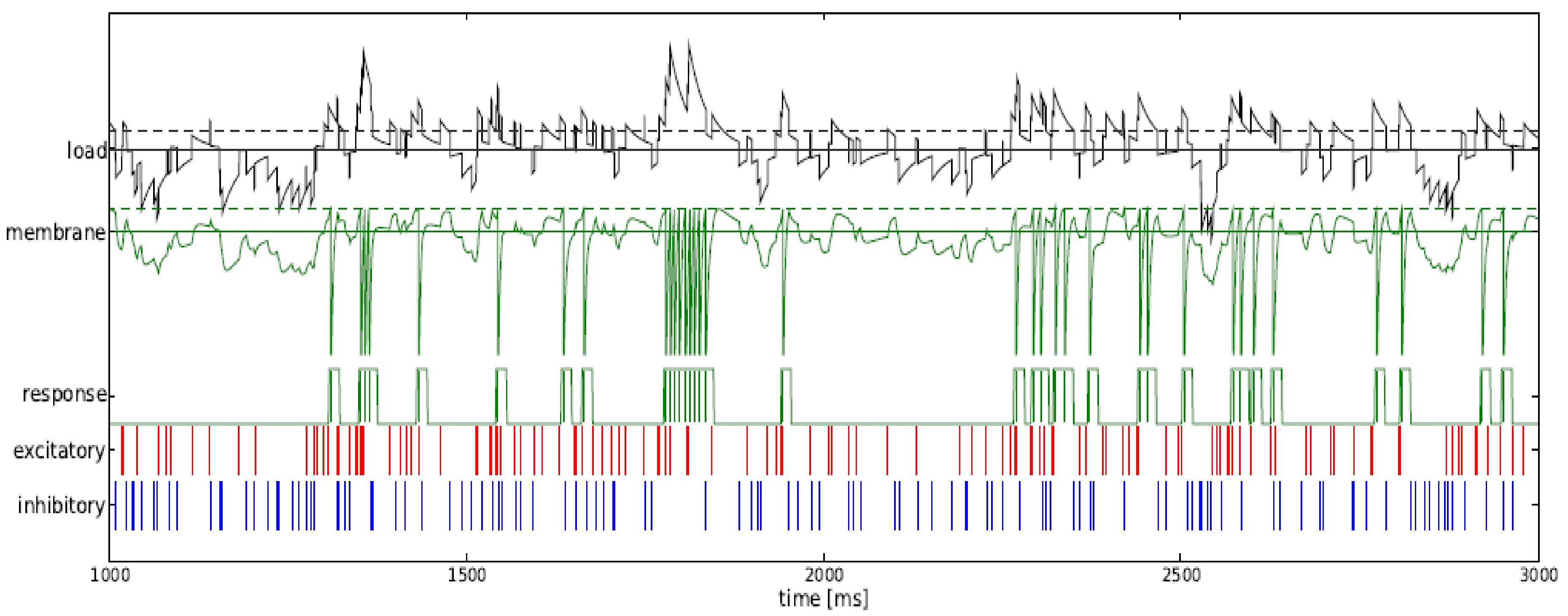
Single neuron behavior

The Load Function

$$\mathcal{L}(t=0) = \sum_{\text{spikes}\,i} w_i \cdot \Theta(-\mathrm{t}_\mathrm{i}) \cdot \exp t_i/\tau$$

with $\tau = \max(\tau_{syn}, \tau_{mem})$

the neuron fires if $\mathcal{L} > \mathcal{L}_{\text{thresh}}$



Statistical treatment of neural activity

Gaussian distribution: $\mathcal{N}_{\mathrm{M}}(\mu, \Sigma)$, for example $\mathcal{N}_1\left(\overline{\mathcal{L}}, \sigma^2\right)$

two channels: shared $\left(\mathcal{L}_s\right)$ and private $\left(\mathcal{L}_p\right)$

$$P_0(\mathcal{L}_A = a) = \int_{-\infty}^{\infty} P_0(\mathcal{L}_s = x) P_0(\mathcal{L}_p = a - x) dx = \mathcal{N}_1\left(\overline{\mathcal{L}}_s + \overline{\mathcal{L}}_p, \sigma_{\mathcal{L}_s}^2 + \sigma_{\mathcal{L}_p}^2\right)$$

two neurons sharing inputs:

$$P_0(\mathcal{L}_A = a, \mathcal{L}_B = b) = \int_{-\infty}^{\infty} P_0(\mathcal{L}_s = x) P_0(\mathcal{L}_p = a - x) P_0(\mathcal{L}_p = b - x) dx$$

$\rightarrow$ multivariate normal distributions

numerical integration: $P(A, \neg B) := P(a > \mathcal{L}_{\text{thresh}}, b < \mathcal{L}_{\text{thresh}})$

conditional probability: $P(A \mid B) = P(A, B) / P(B)$

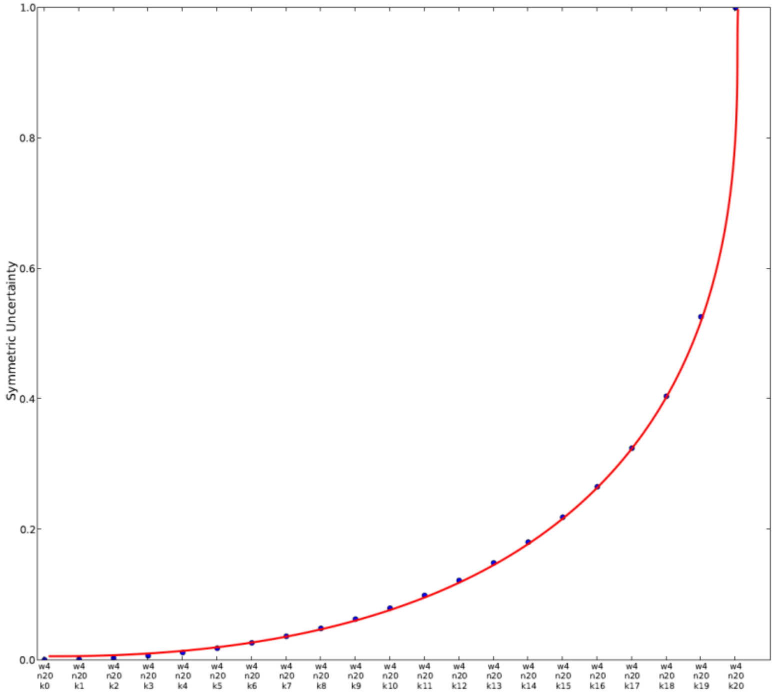
Symmetric Uncertainty

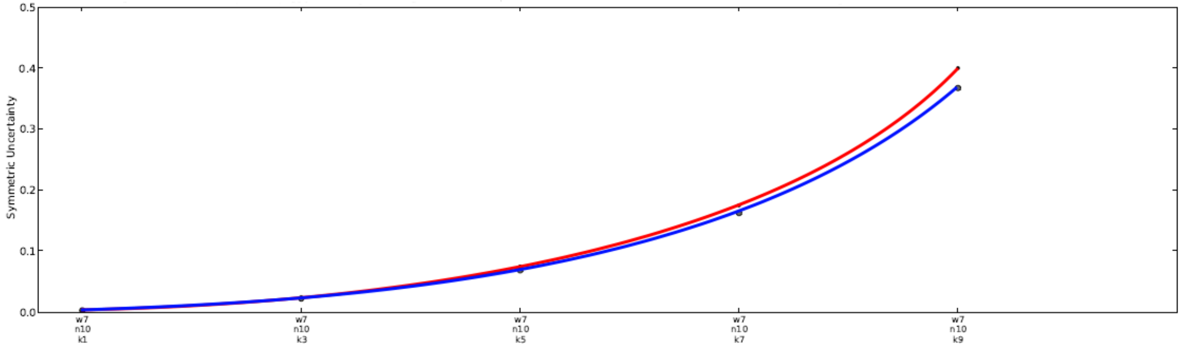
$$SU(X,Y)=2R=2\frac{I(X;Y)}{H(X)+H(Y)}$$

$$I(A;B)=\sum_{A\in\{0,1\}}\sum_{B\in\{0,1\}}p(A\cap B)\log\frac{p(A\cap B)}{p(A)p(B)}$$

features:

- symmetric in X and Y
- pure information theory → highly general
- normalized: SU $\in[0,1]$ ⇒ allows comparison over a wide range of spike train parameters
- no free parameters !
- more than just synchrony

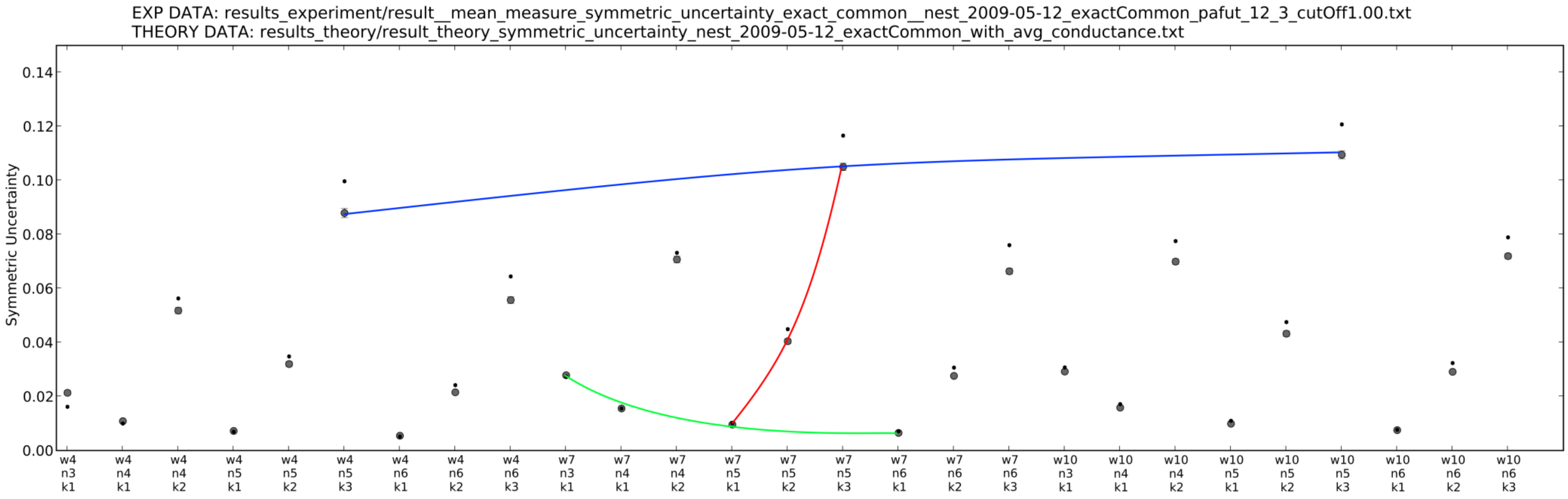




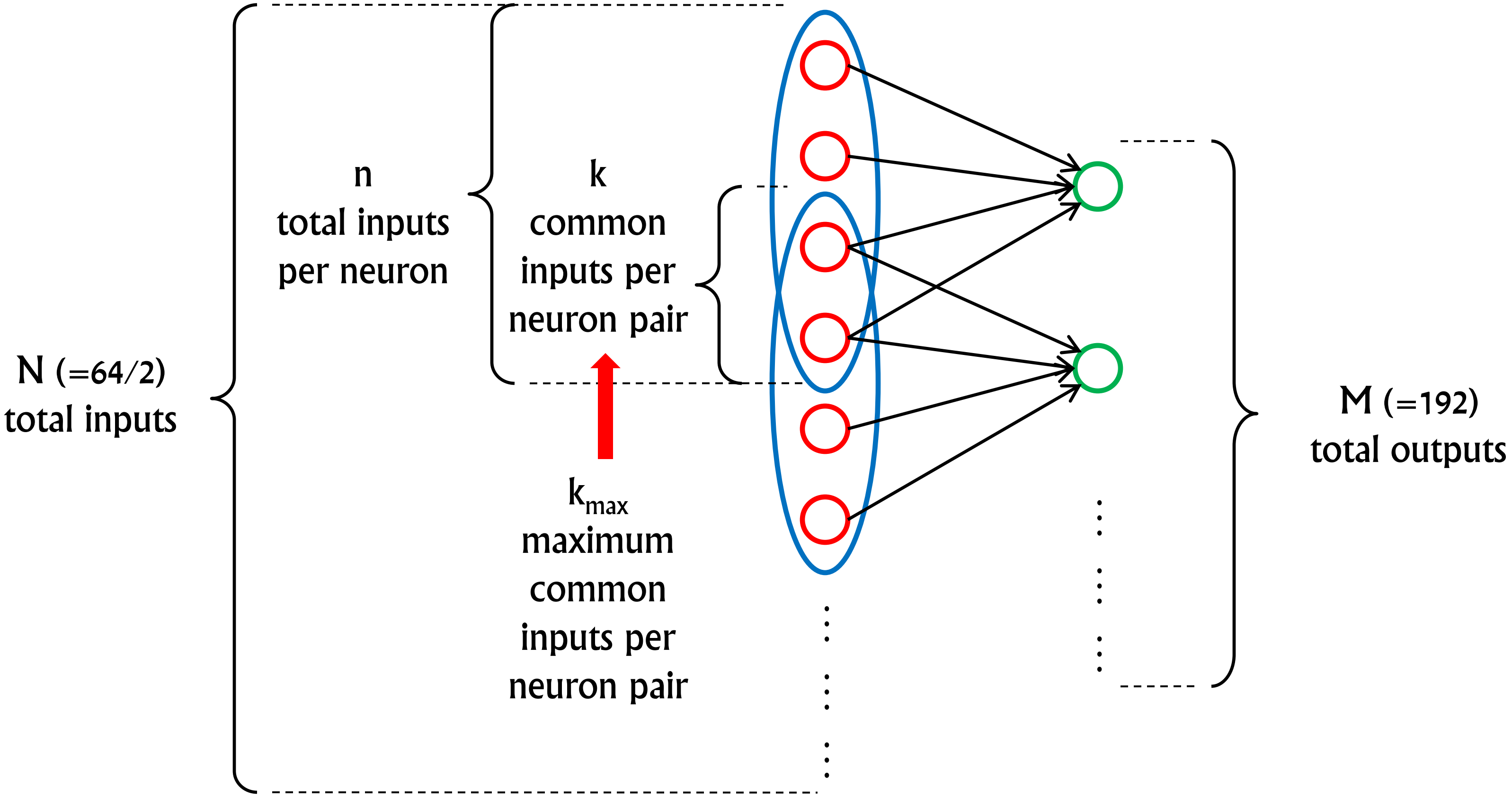
Partial derivatives

V_{thresh} = -55 mV
V_{rest} = -59 mV
τ_{mem} = 5 ms

simtime = 20 s
w_{exc}= w · 0,5 nS



The mapping problem



N (=64/2)
total inputs

n
total inputs
per neuron

k
common
inputs per
neuron pair

k_{max}
maximum
common
inputs per
neuron pair

M (=192)
total outputs

for given N, n minimize k
while keeping M ≥ 192

or

for given N, n (large), k_{max} (small)
can we find enough subsets (M)?

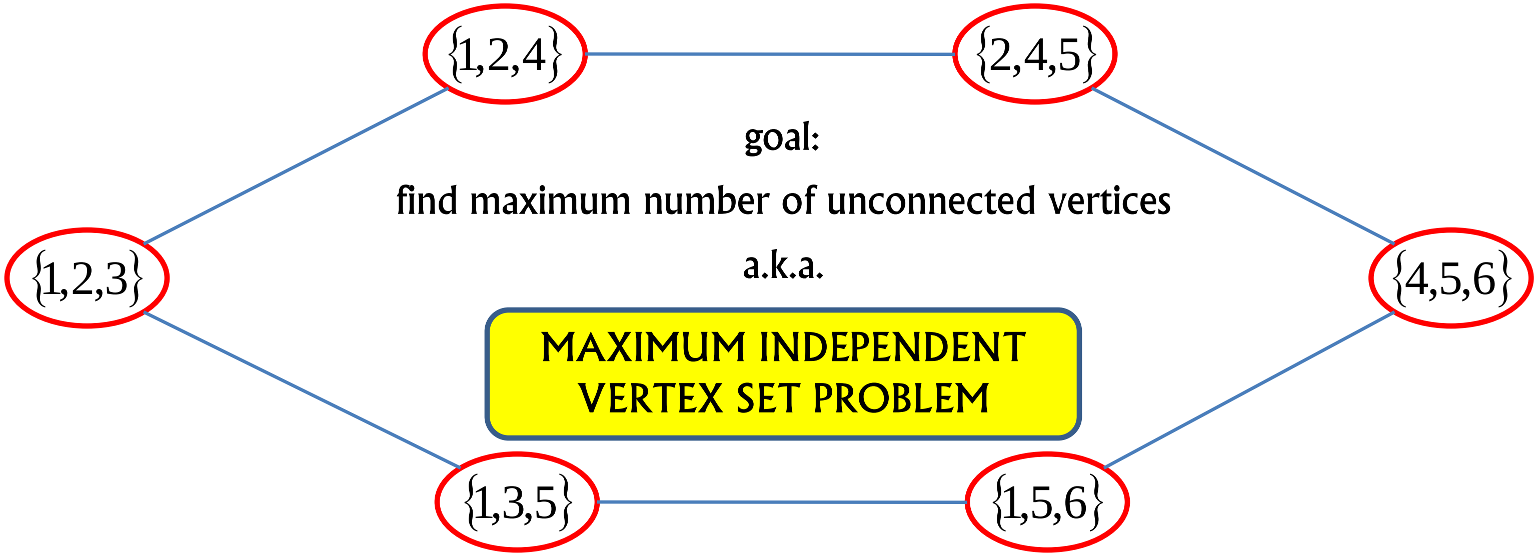
A graph theoretical approach

vertices ← subsets
edges ← overlap between subsets

two subsets are connected if they have more than k_{max} elements in common

$$\Omega = \{\{1,2,3\}, \{1,2,4\}, \{1,3,5\}, \{4,5,6\}, \{2,4,5\}, \{1,5,6\}\}$$

$$k_{\max} = 1$$



goal:

find maximum number of unconnected vertices

a.k.a.

MAXIMUM INDEPENDENT VERTEX SET PROBLEM

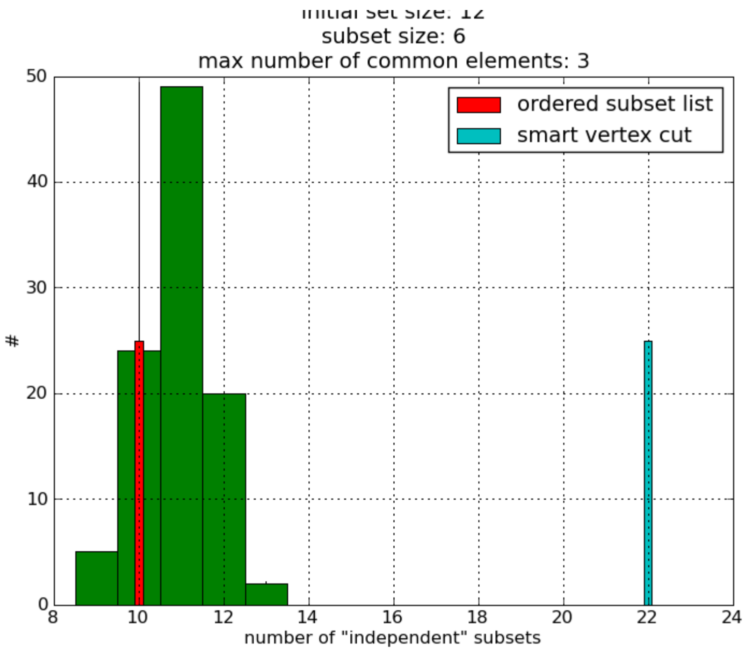
Results

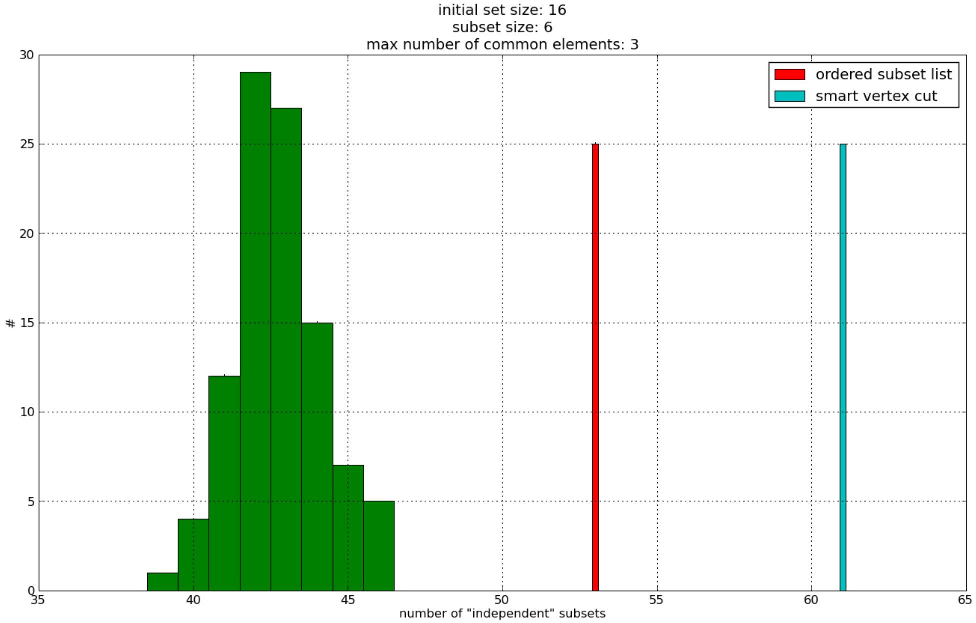
The hybrid algorithm

idea: 1) use greedy algorithm until $\mathrm{card}(\Omega) \leq \mathrm{smart\_barrier} \approx 40000$

2) use "smart" (vertex-cut) algorithm from that point onward

Results



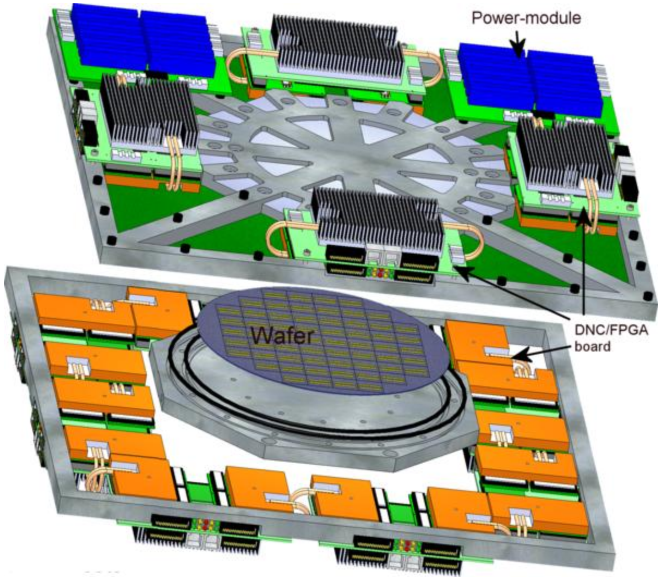


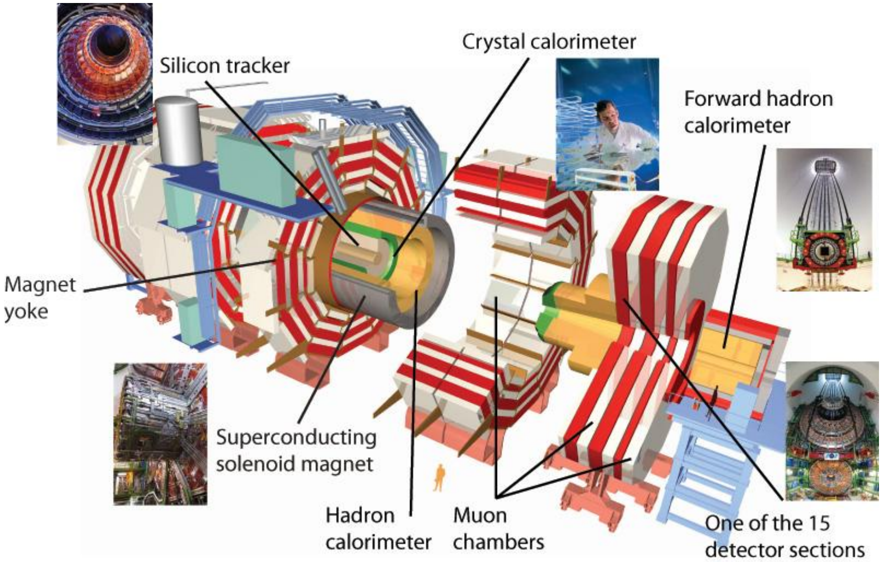
$N=32,\ M \geq 192$

$\min(k_{max})$

- $n=4,\ k_{max}=2 \rightarrow M=1240$
- $n=5,\ k_{max}=2 \rightarrow M=348$
- $n=6,\ k_{max}=3 \rightarrow M=1357$
- $n=7,\ k_{max}=3 \rightarrow M=412$

Limited output





on-wafer bandwidth:
2 Tbps (Layer 1)

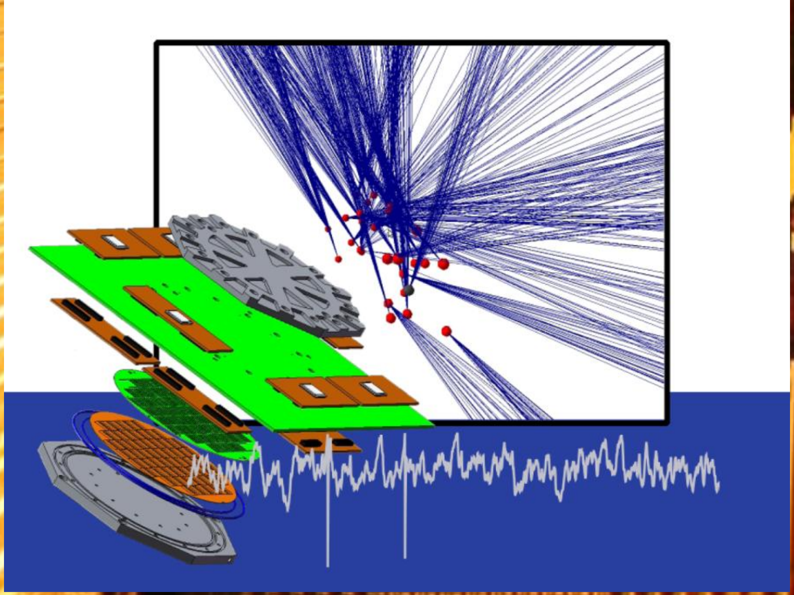
front-end data volume @CMS:
2 Tbps

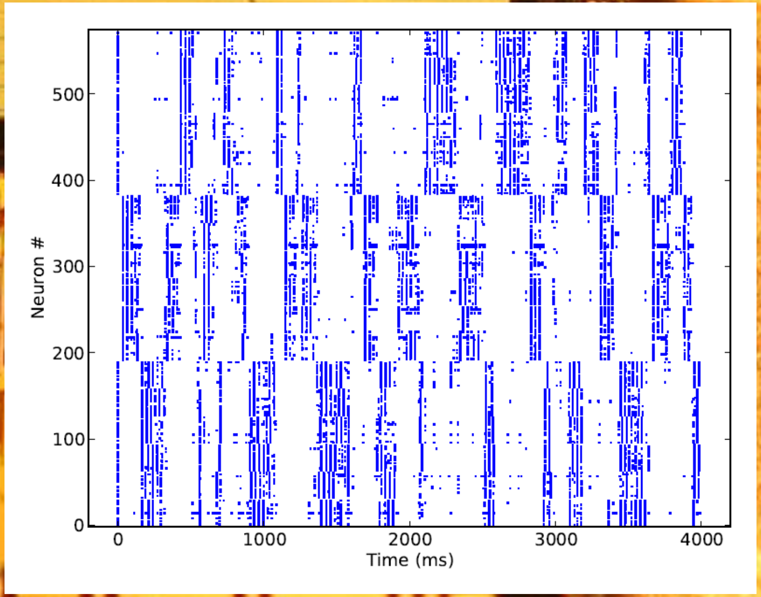
only 1% of this data can be read out

voltages: 2 per chip, 384 chips
20 MB/s for one channel

PART III

THE FACETS DEMONSTRATOR





The FACETS Demonstrator…

… integrates techniques and tools developed within FACETS …

… into a complete workflow …

… that allows to use the FACETS wafer-scale hardware system …

(currently: a virtual version of it)

… for the emulation of benchmark cortical neural network models …

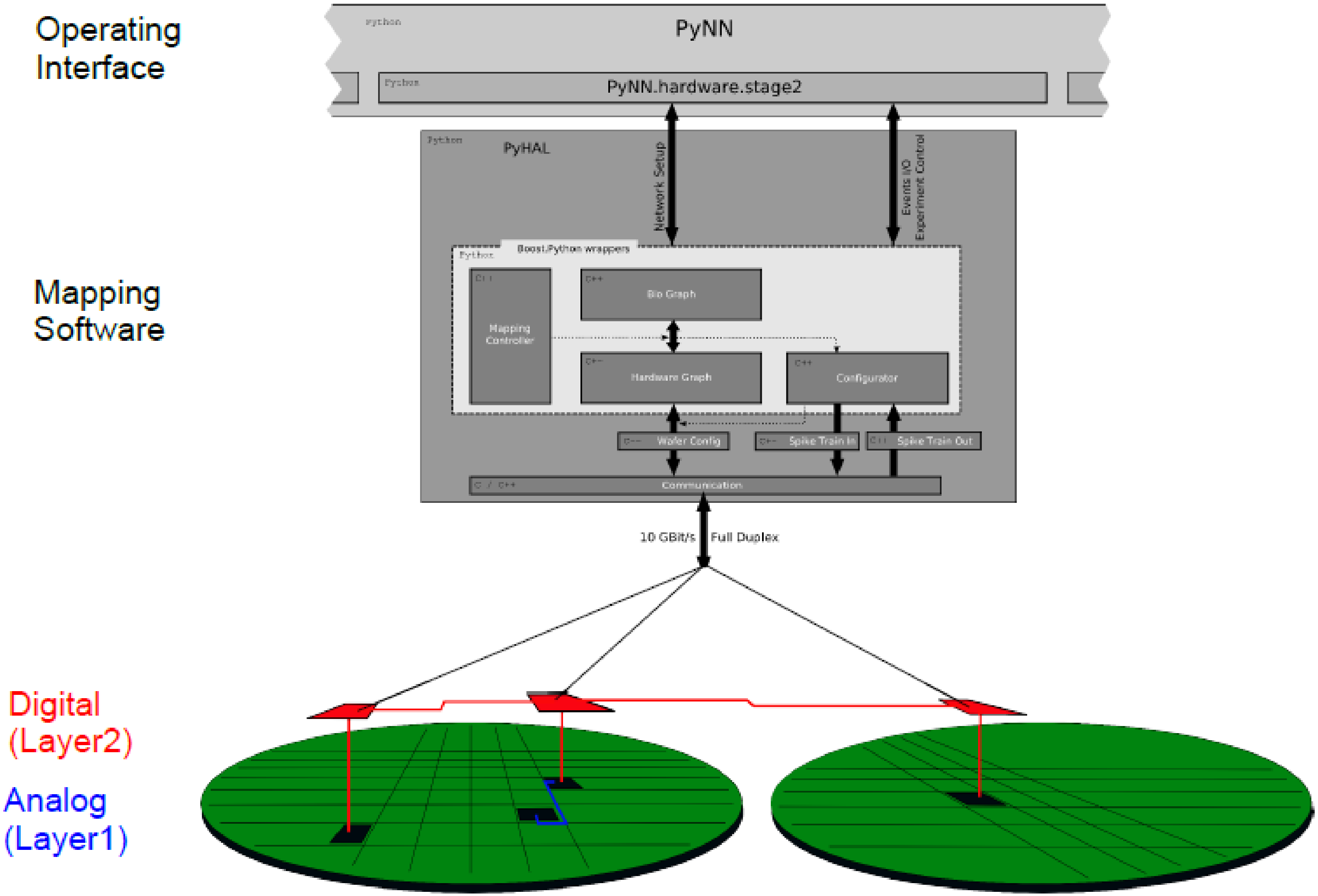
… which exhibit functionality that can be demonstrated …

… which are written in PyNN …

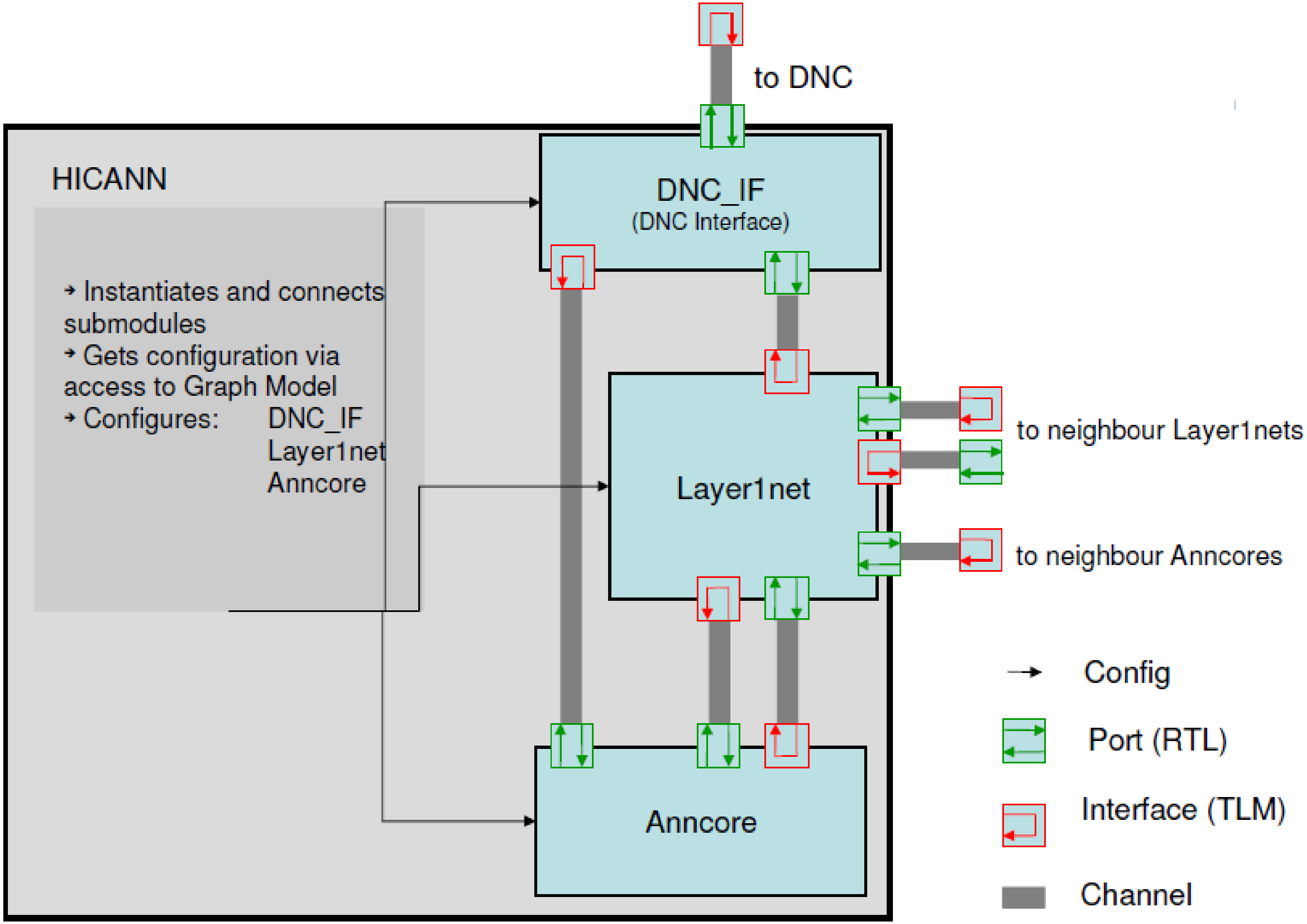
… and therefore can be computed with established software simulators

(for verification, performance evaluation etc.)

Simulating the emulator



Simulating the emulator



- Instantiates and connects submodules
- Gets configuration via access to Graph Model
- Configures: DNC_IF
  Layer1net
  Anncore

Goals

Testing and evaluation of all involved software layers

Virtual hardware allows to

- test software before hardware is available
- test without possible hardware-specific problems
- provide a preliminary PyNN module for off-line testing of experiments

Verification of possible hardware changes

e.g. optionally insert detailed HICANN model

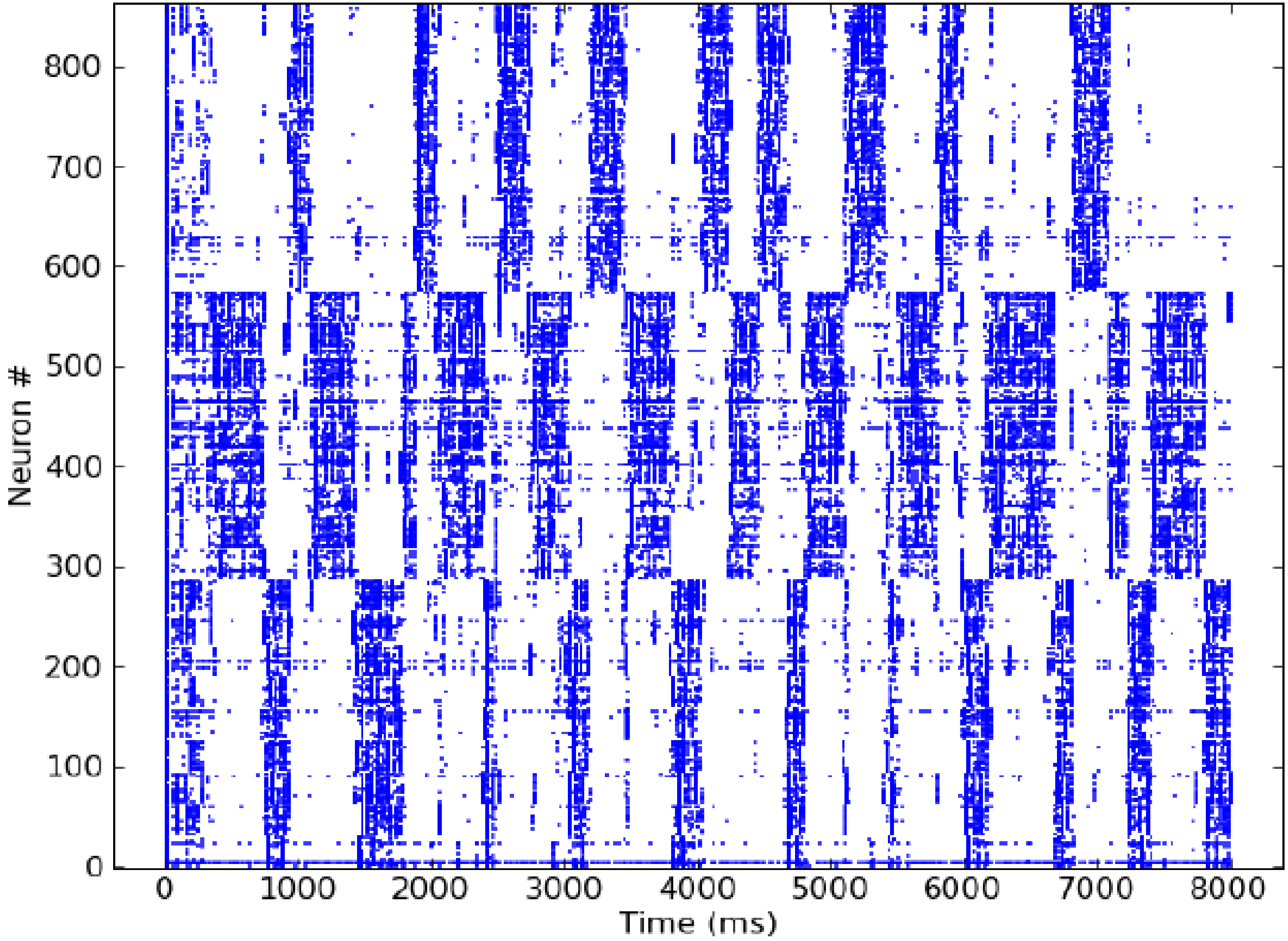
$\Rightarrow$ indispensable framework for preparation and development tasks

The Demonstrator models (so far)

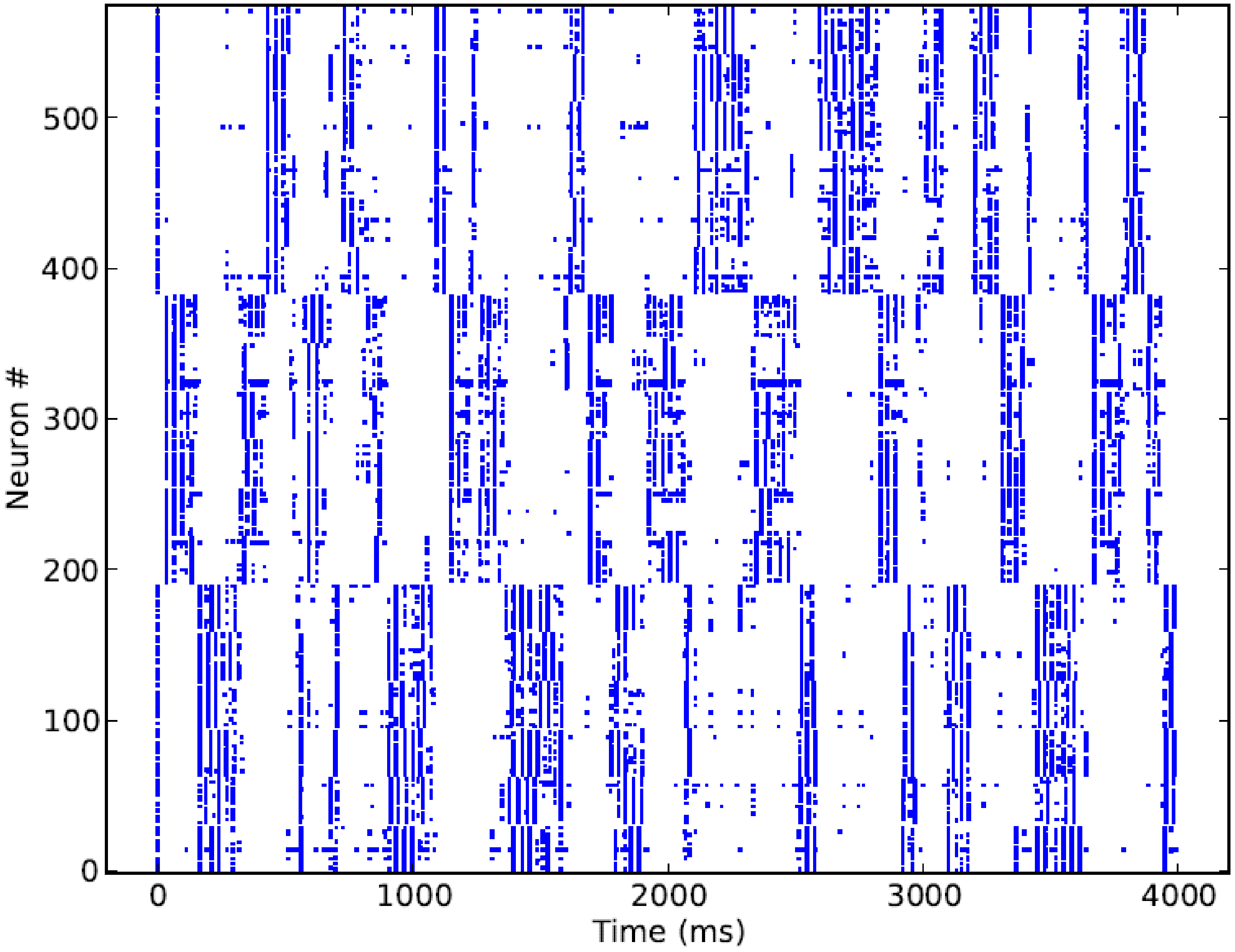
- A layer 2/3 attractor memory
  (by KTH, Krishnamurty / Lansner)
- A synfire chain model
  (by INCM and ALUF, Kremkow / Aertsen / Masson)
- A model of self-sustaining cortical AI states
  (by UNIC, Davison / Destexhe)
- Upcoming: Two-layer model by UNIC

All written in PyNN, all scalable, basic versions can be mapped to hardware without synapse loss

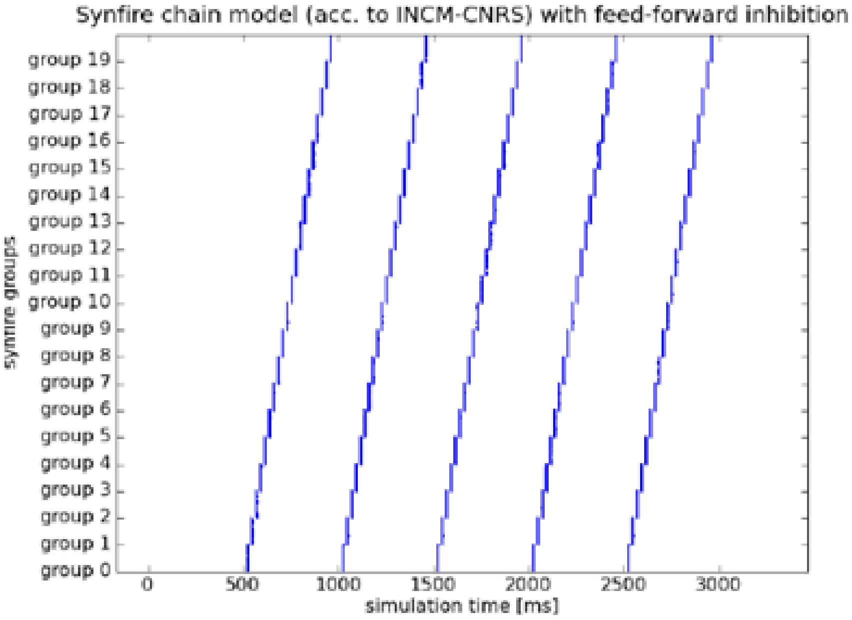
L2/3 cortical attractor memory (NEST)

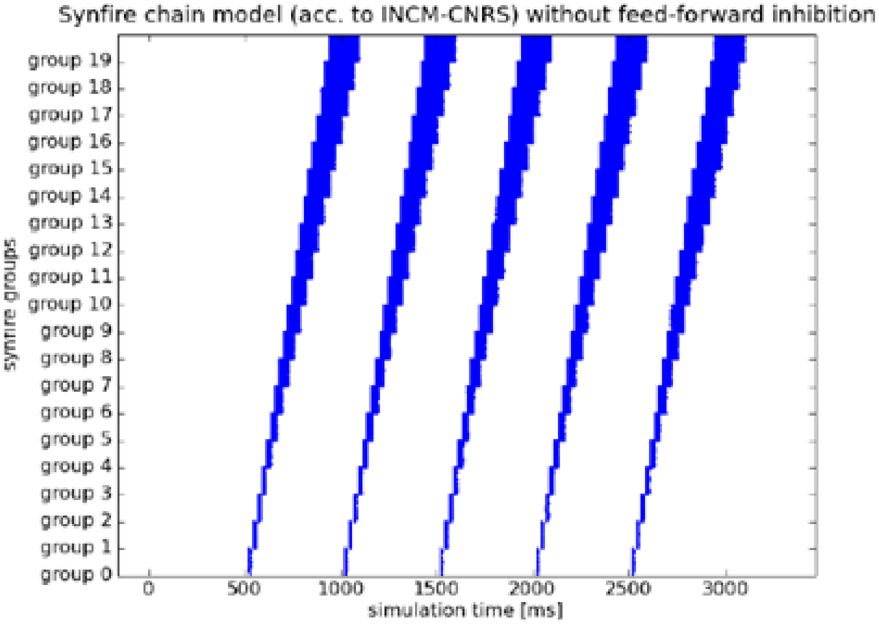


L2/3 cortical attractor memory (virtual HW)

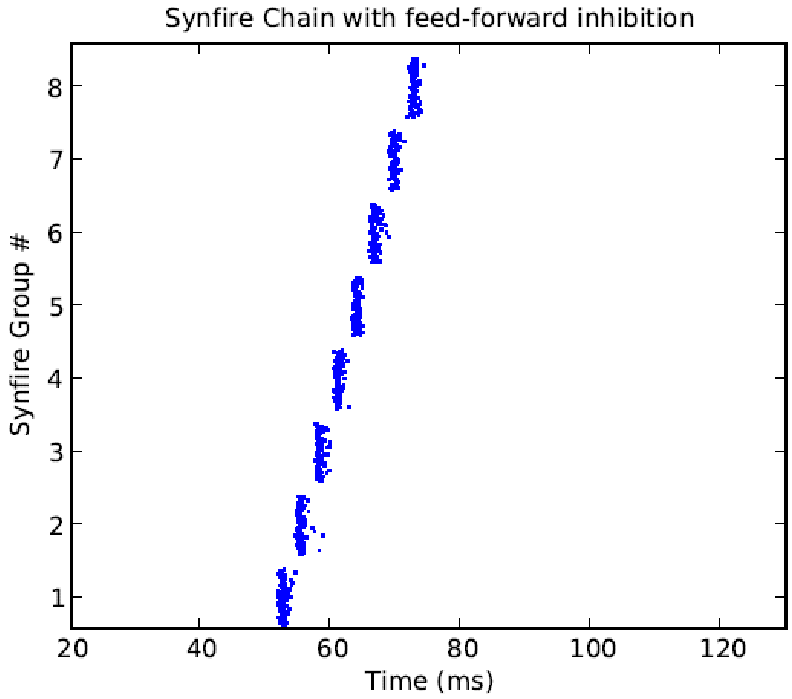


Synfire chain with feedforward inhibition (NEST)

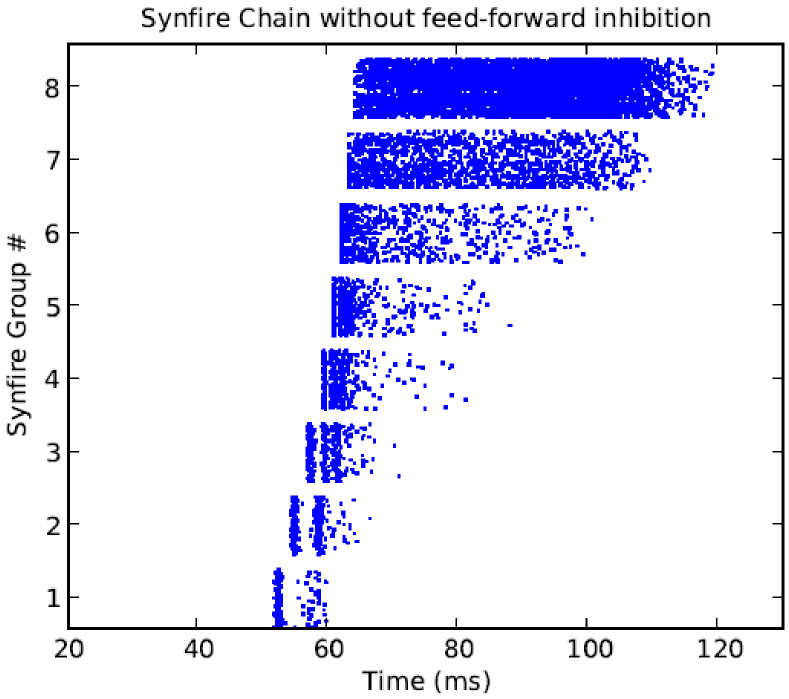




Synfire chain with feedforward inhibition (virtual HW)

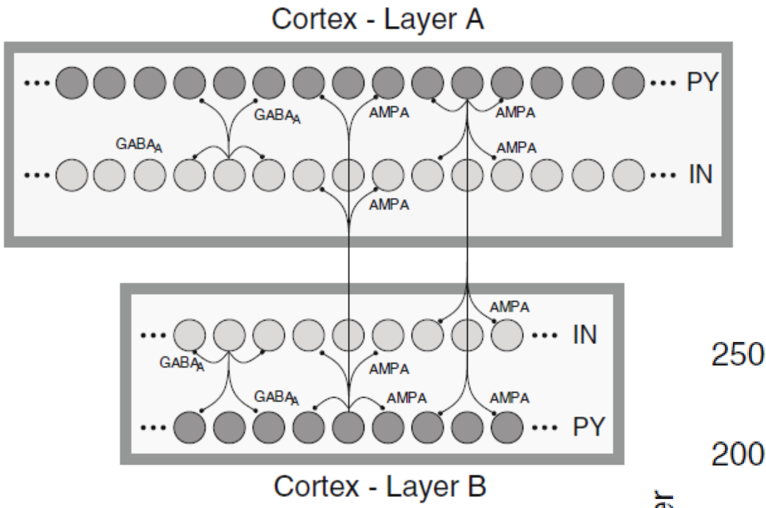


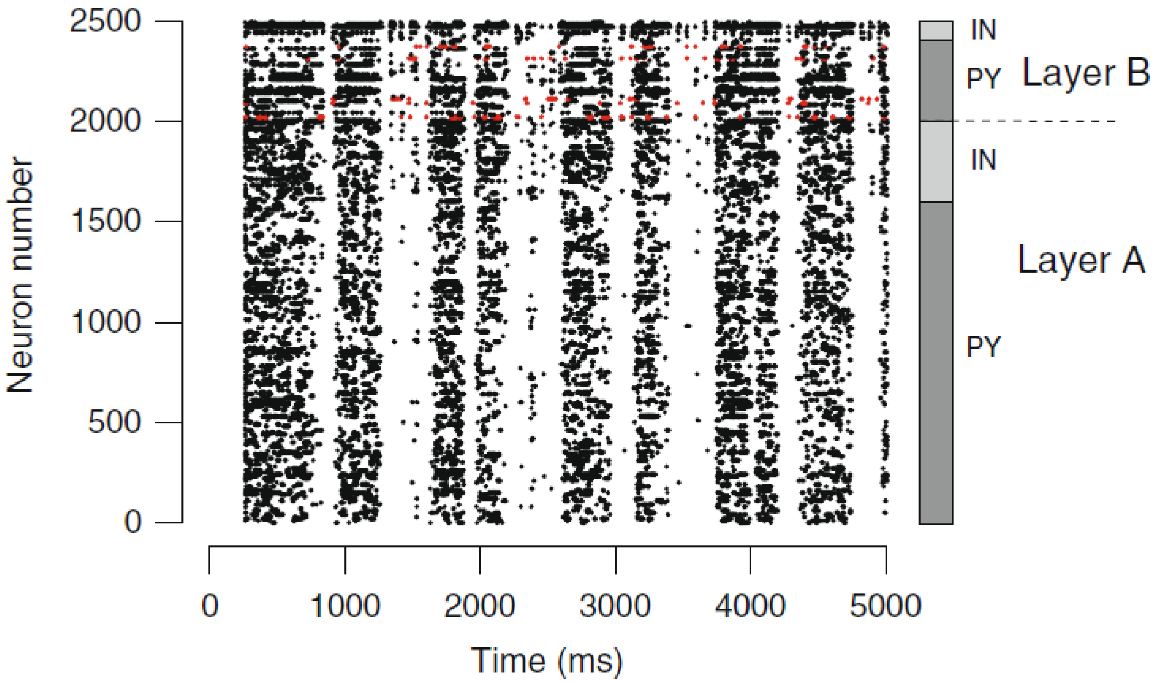
Synfire Chain with feed-forward inhibition



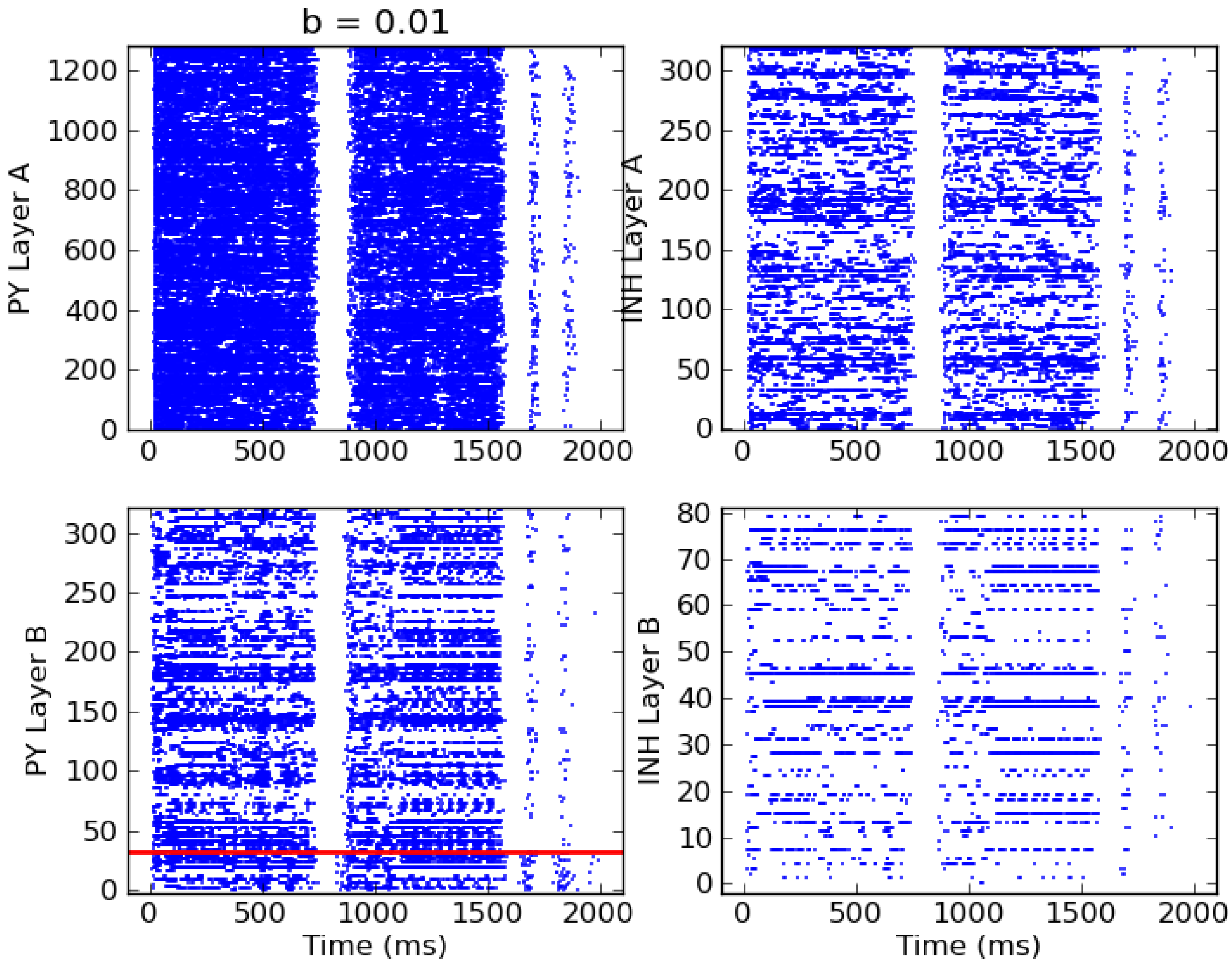
Synfire Chain without feed-forward inhibition

Cortical AI states (NEURON)



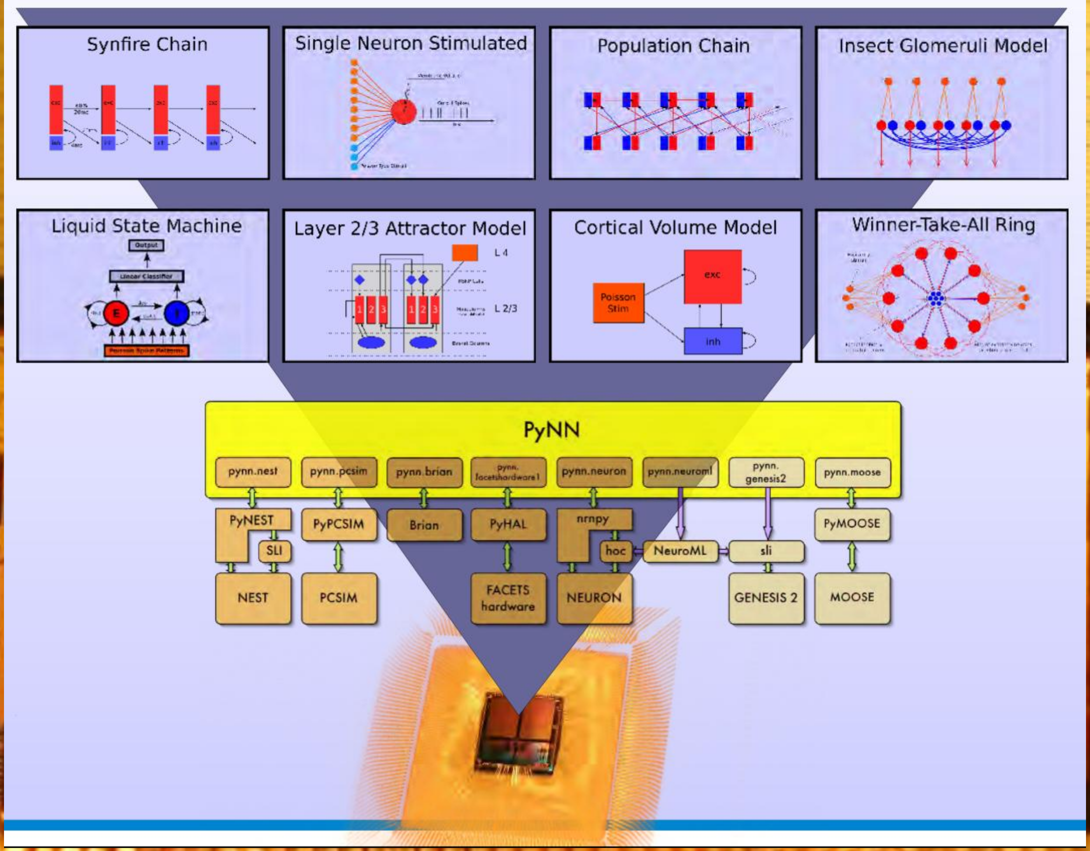


Cortical AI states (virtual HW)

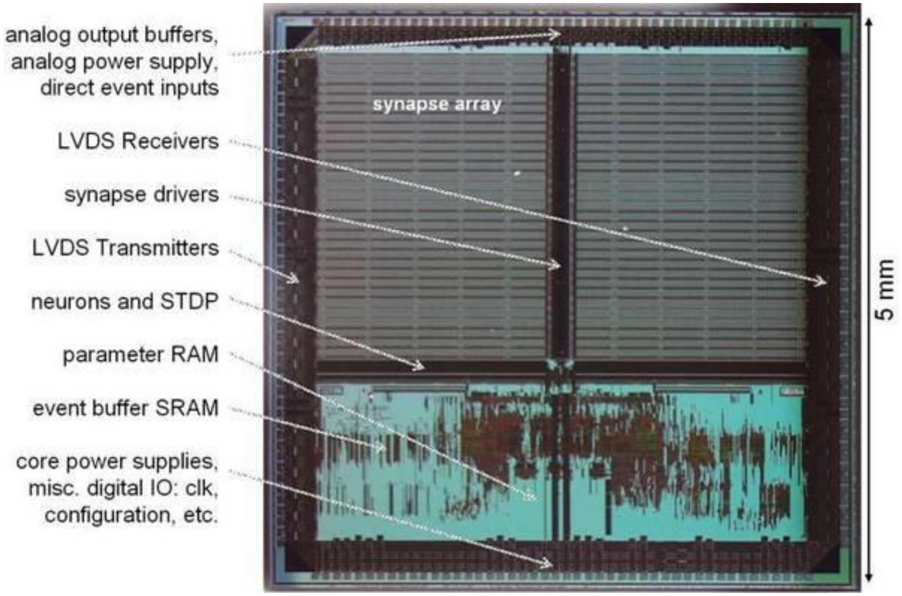


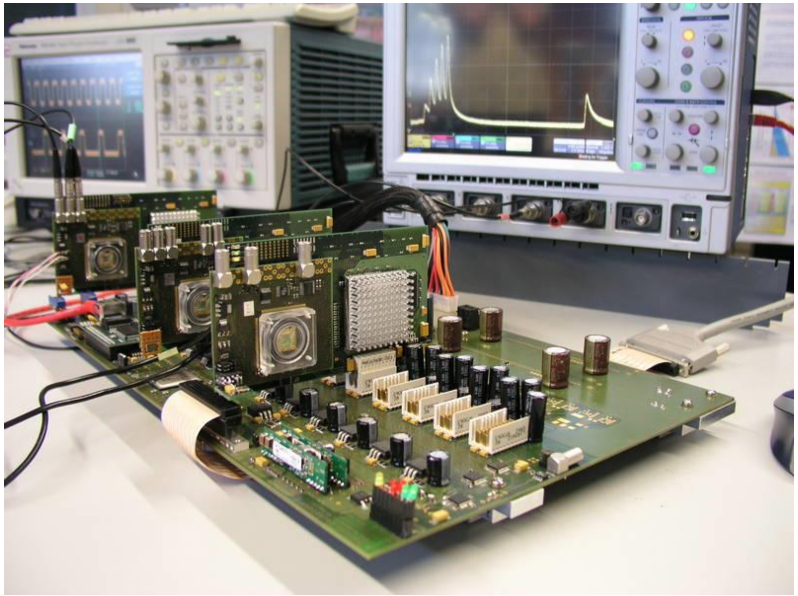
PART IV

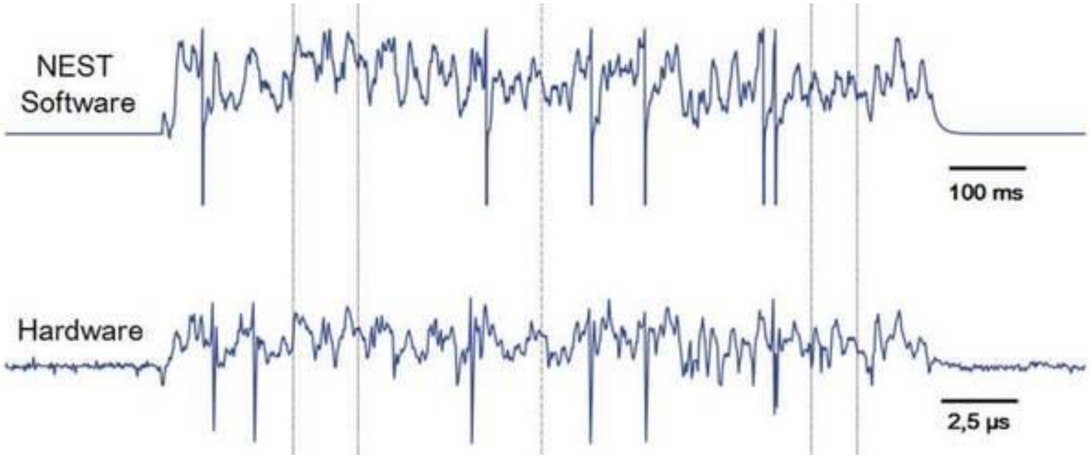
“SPIKEY” - DEMOS



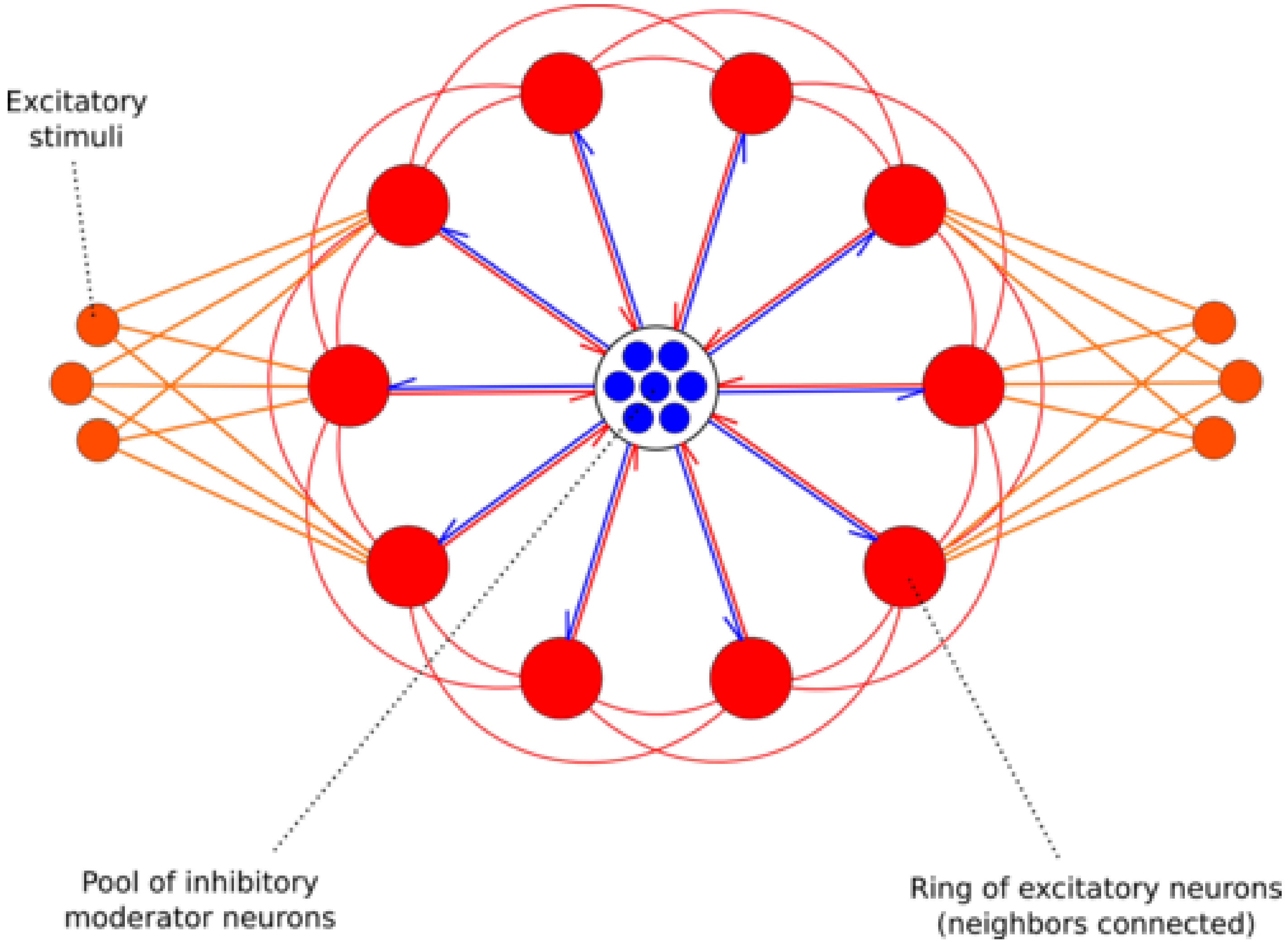
The “Spikey” chip



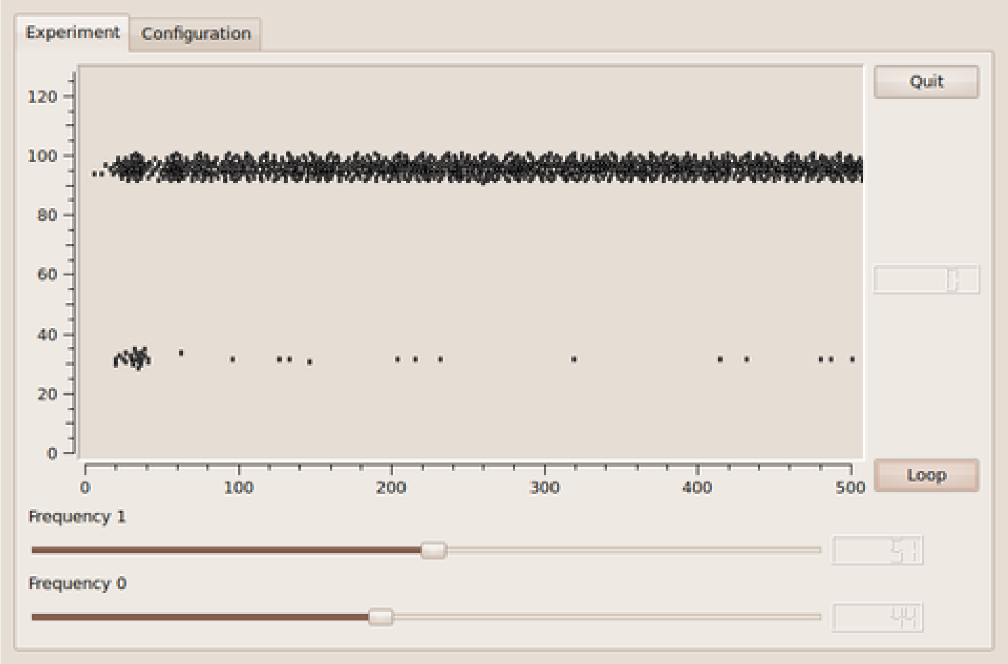




WTA ring

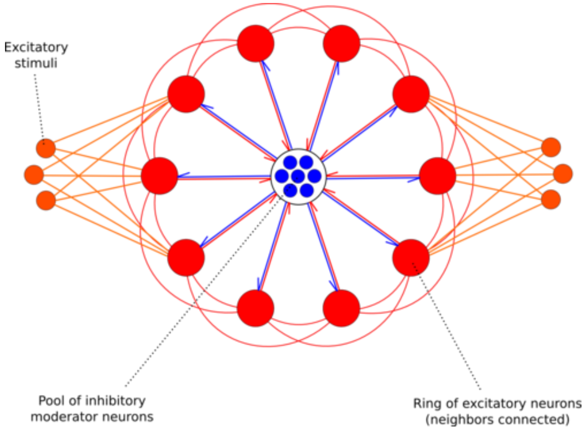

Excitatory
stimuli
Pool of inhibitory
moderator neurons
Ring of excitatory neurons
(neighbors connected)

WTA ring



Frequency 1
51

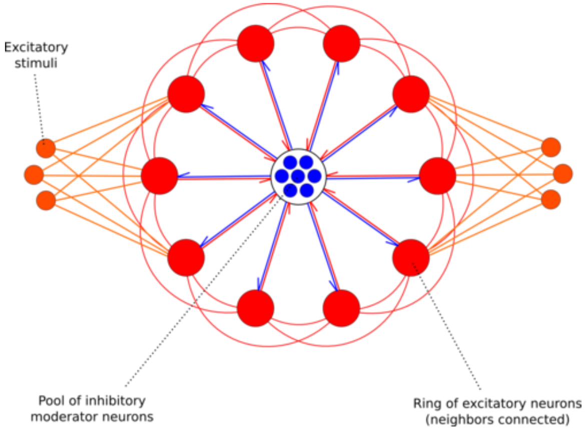
Frequency 0
44



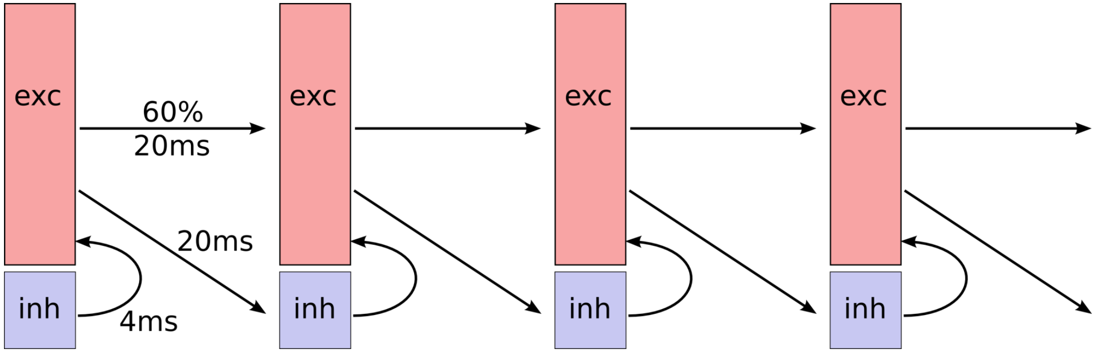
WTA ring

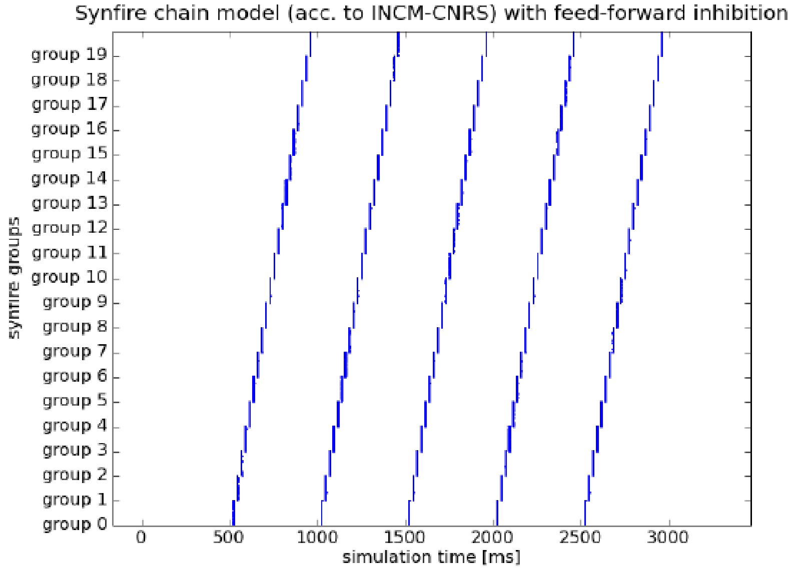

Experiment
Configuration
Quit
120
100
80
60
40
20
0

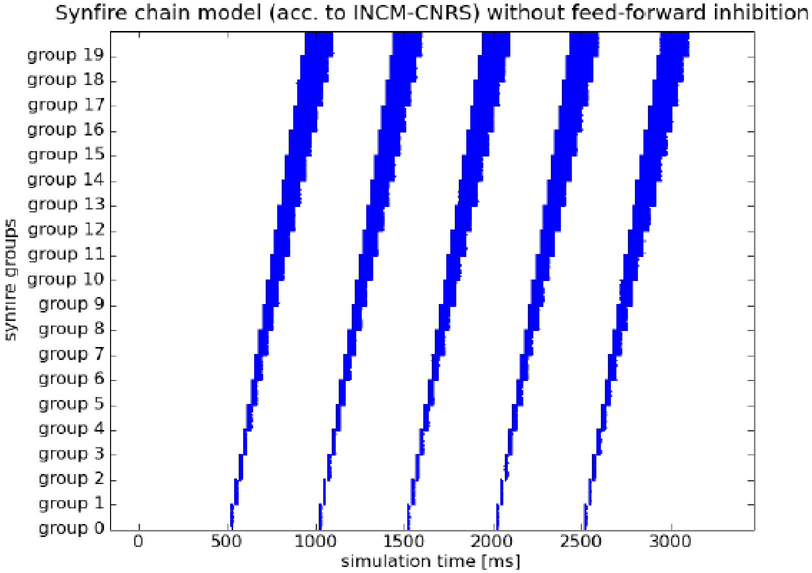

0
100
200
300
Frequency 1
Frequency 0
Experiment
Configuration
Weight for Stimulus 1
Weight for Stimulus 0
Position of Stimulus 1
Position of Stimulus 0
Ring Neighborhood Width
Excitatory-to-Excitatory Weight
Excitatory-to-Inhibitory Weight
Inhibitory-to-Excitatory Weight
Duration


Excitatory stimuli
Pool of inhibitory moderator neurons
Ring of excitatory neurons (neighbors connected)

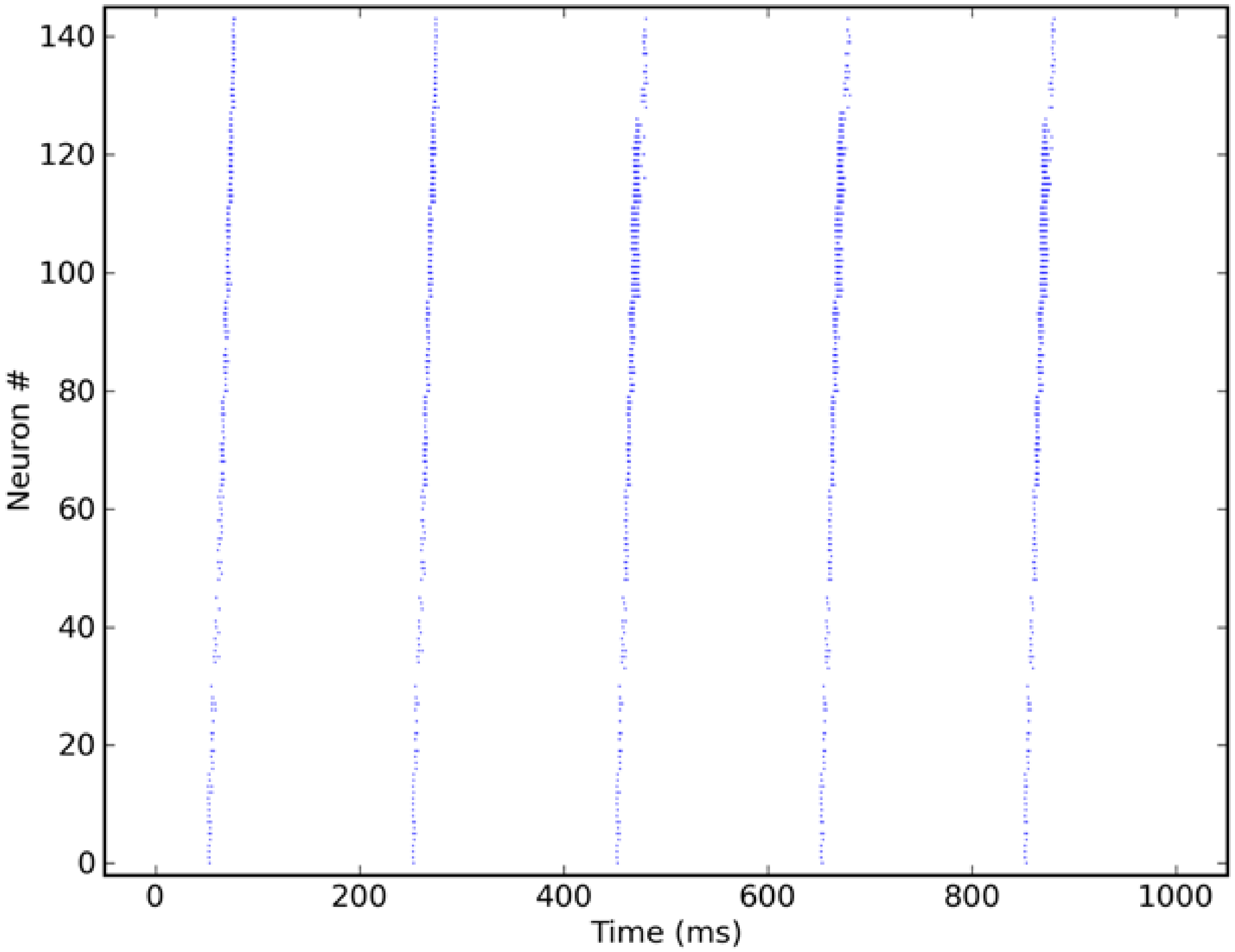
Synfire chain with feedforward inhibition



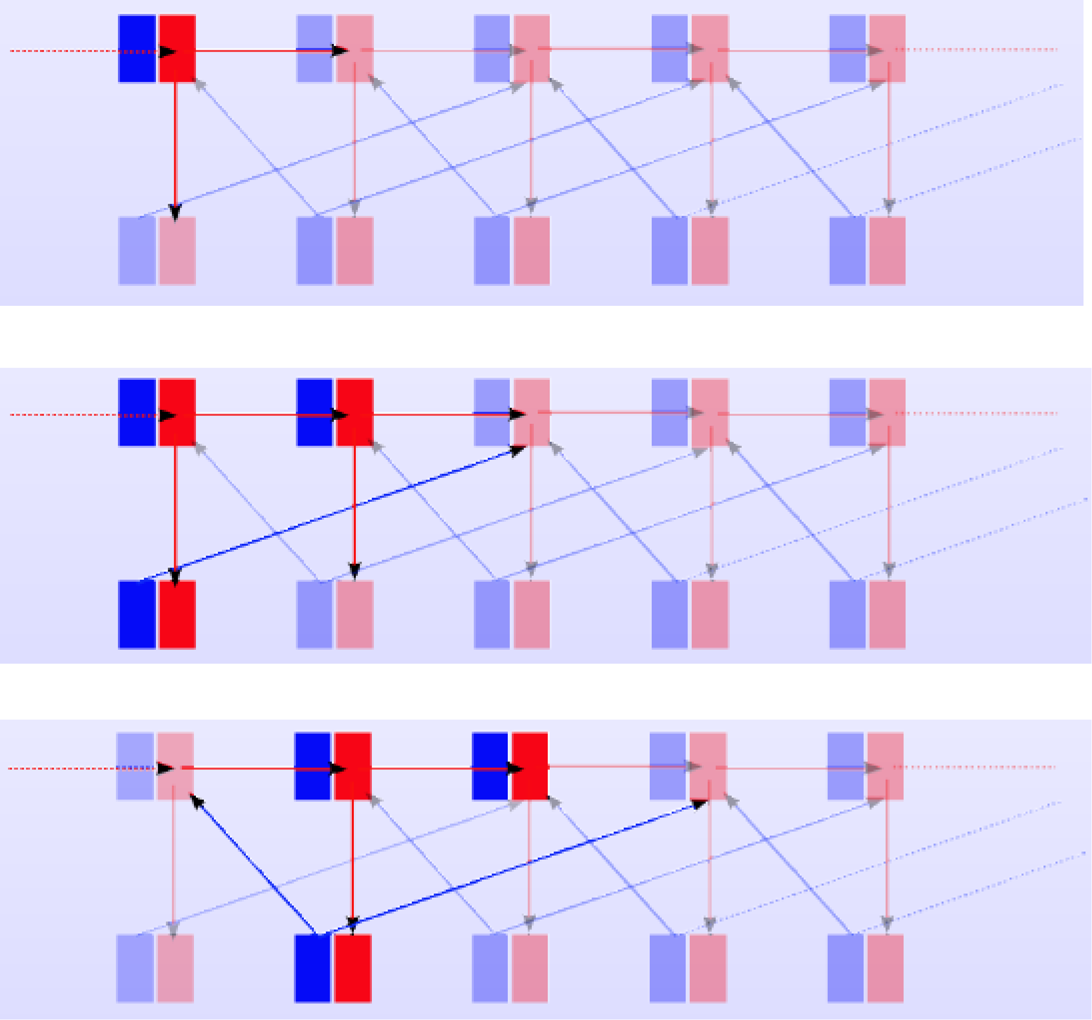




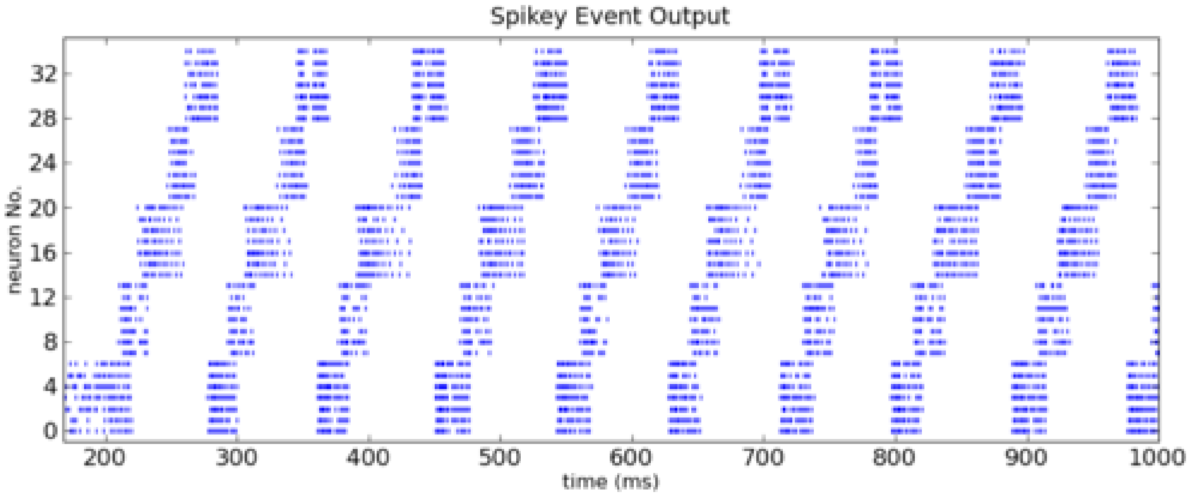
Synfire chain with feedforward inhibition

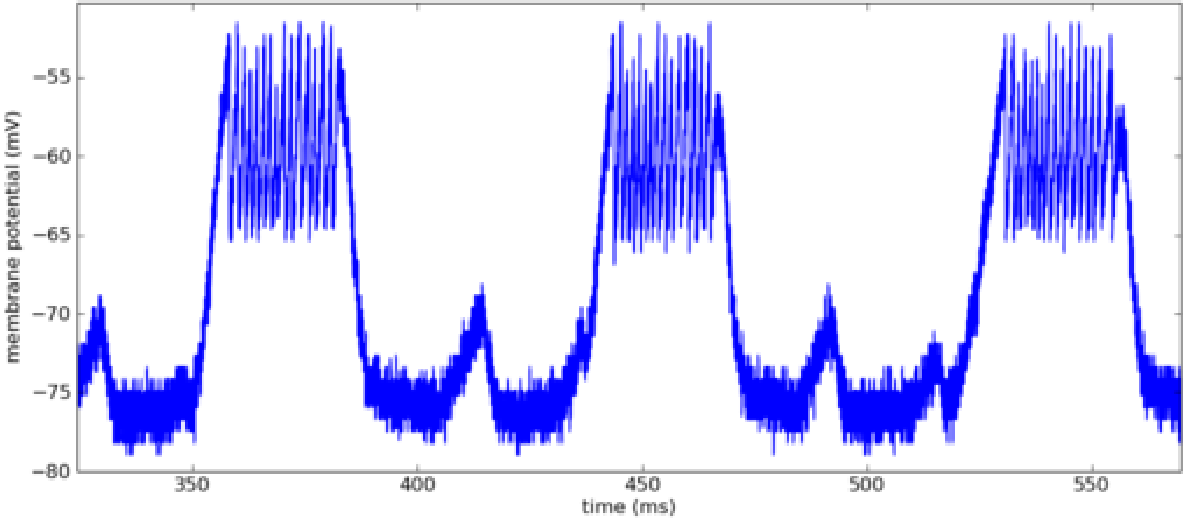


“Hellfire chain”

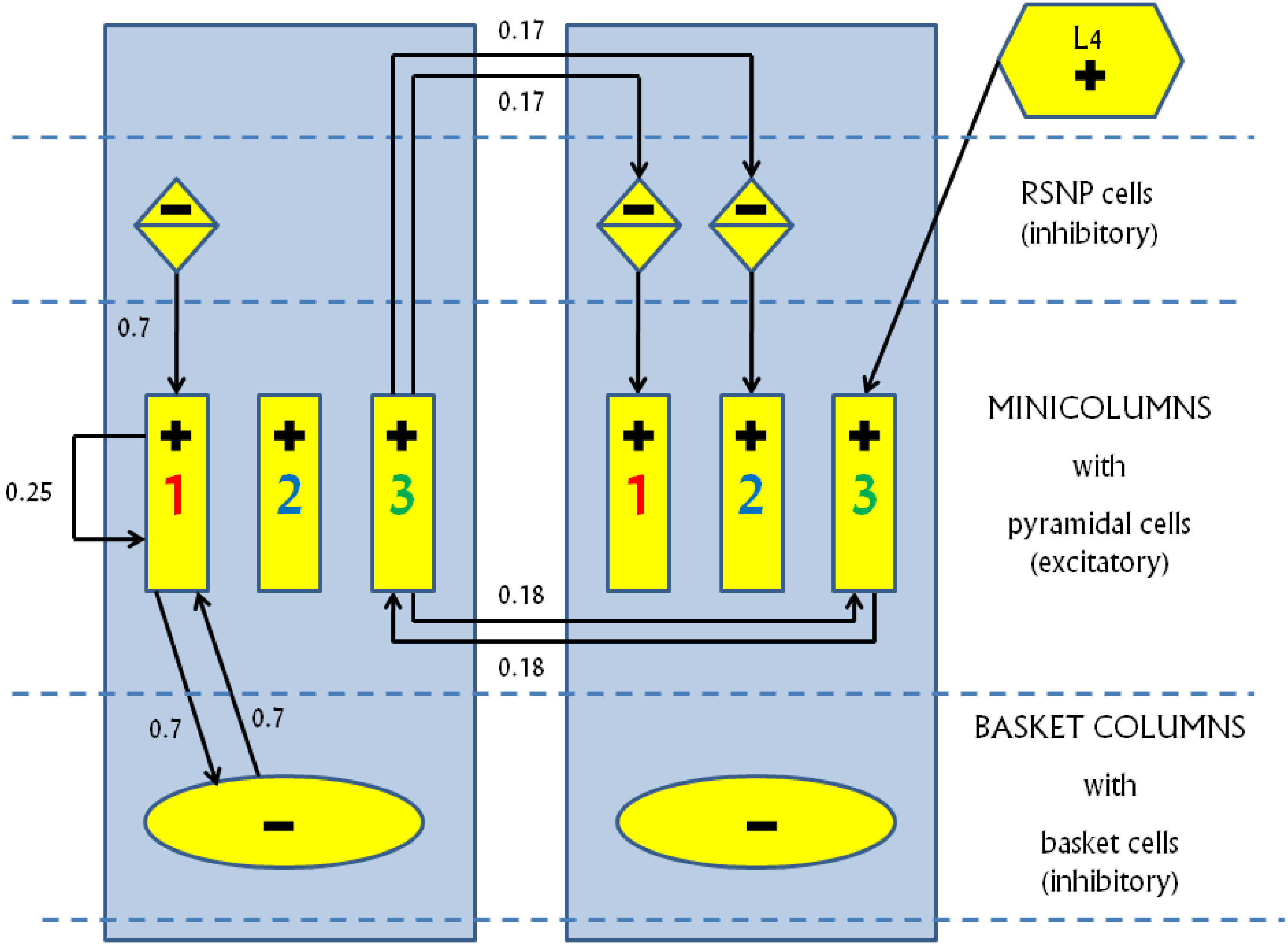


“Hellfire chain”





L2/3 cortical attractor memory



RSNP cells
(inhibitory)

MINICOLUMNS

with

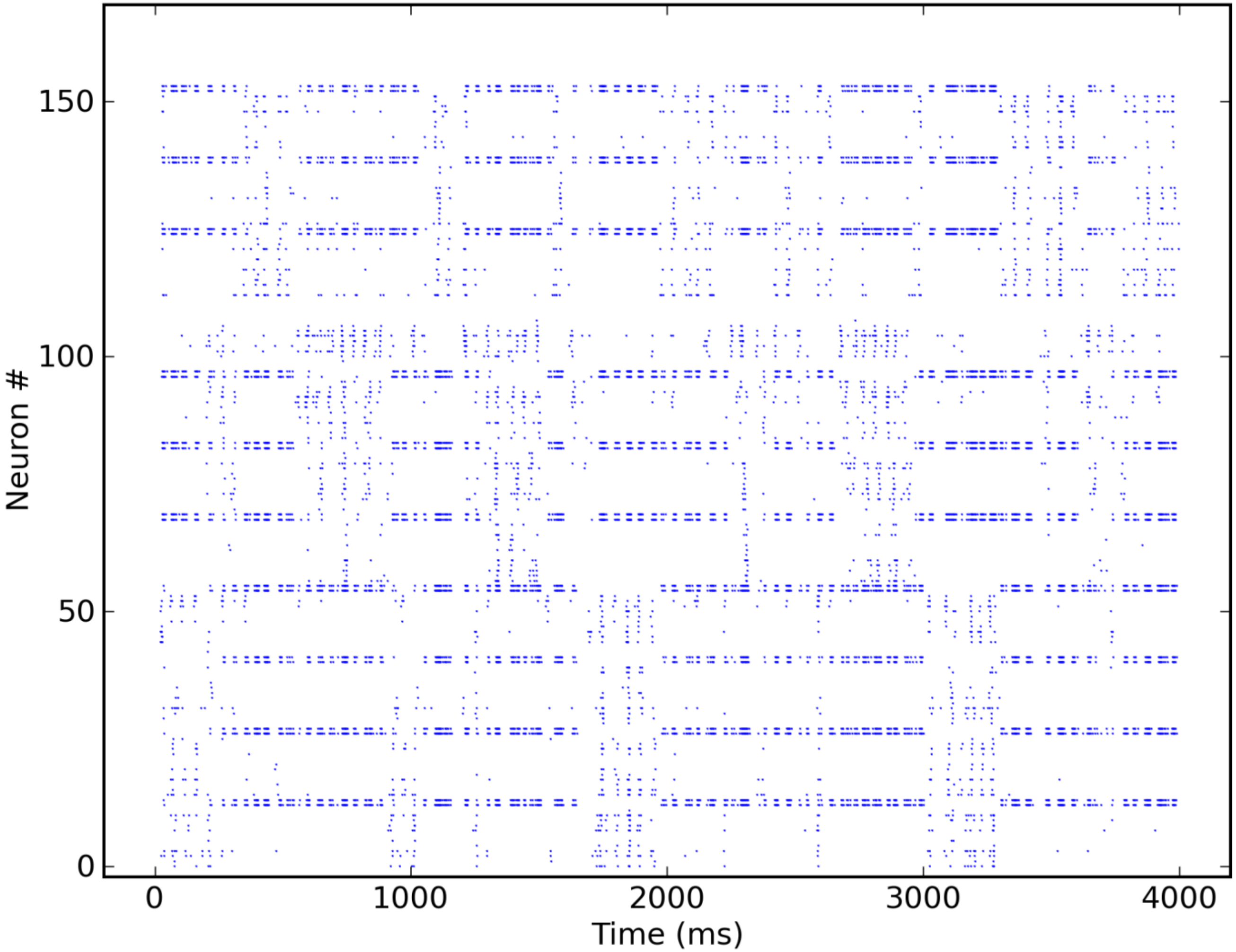
pyramidal cells
(excitatory)

BASKET COLUMNS

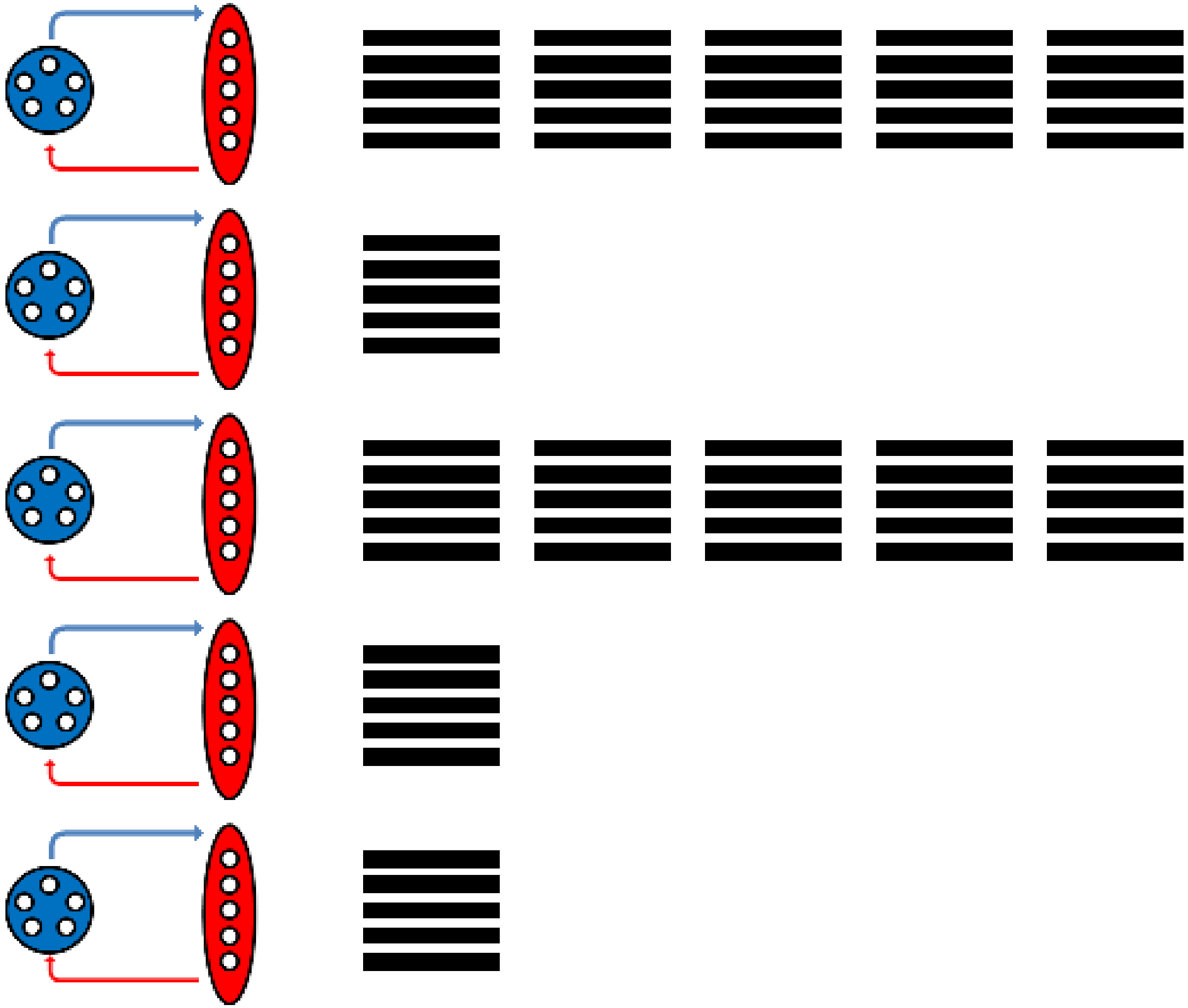
with

basket cells
(inhibitory)

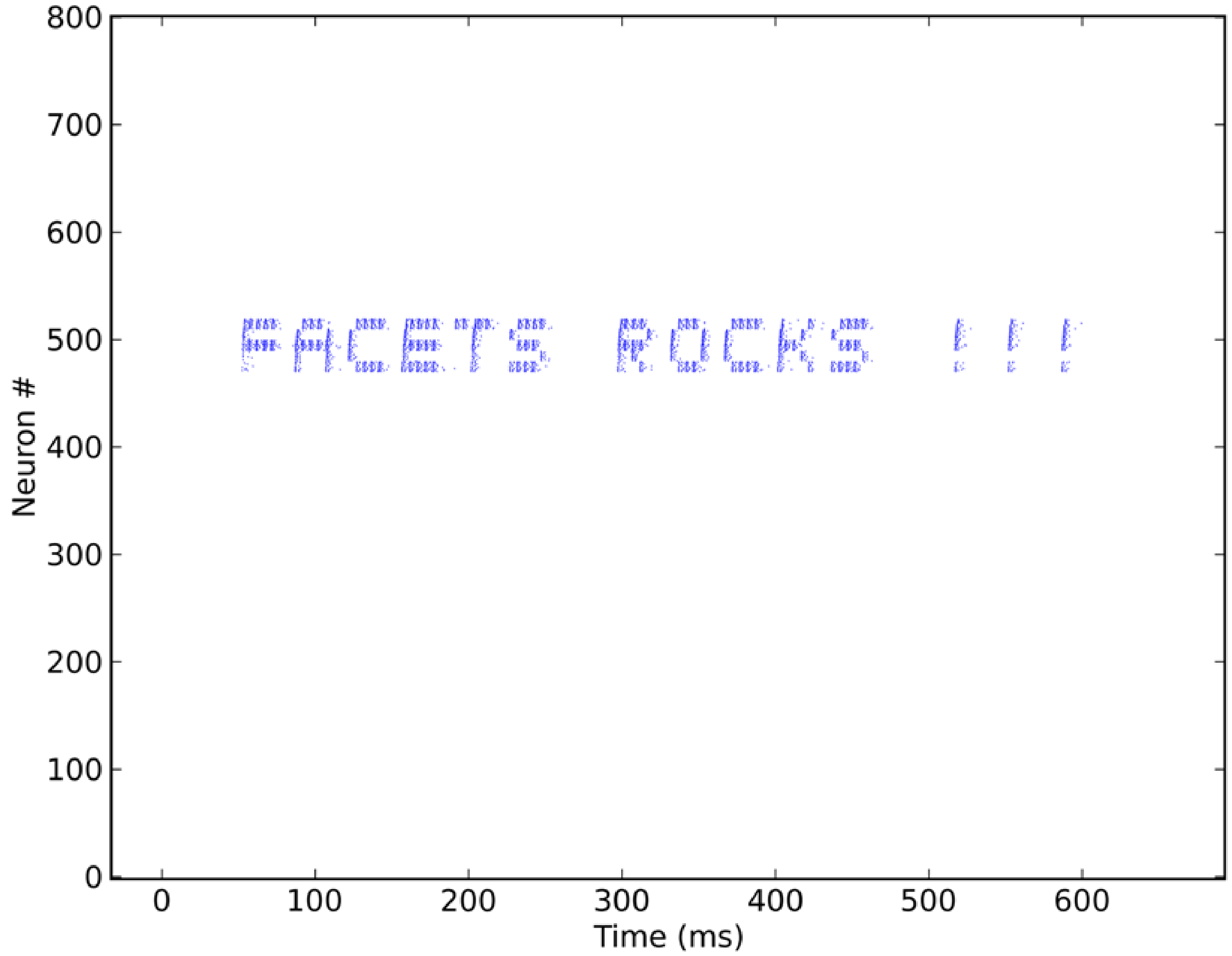
L2/3 cortical attractor memory



Talking Spikey



Talking Spikey



PART V

SUMMARY & TO-DO-LIST

Summary

well-established workflow:

1. write model in PyNN
2. run!

Summary

well-established workflow:

1. **write model in PyNN**
2. **run!**

3.1 mapping tool chooses optimal placing and routing

3.2 graph model used for parameter space configuration

3.3 complex, custom-designed software takes care of communication

$\Bigg\}$ this is done automatically…

Summary

0.1 evaluate model – check if suitable for HW

0.2 analyze influence of distortions on dynamics

0.3 find (if possible !) suitable compensation mechanisms

0.4 investigate scaling properties, if necessary

0.5 think about input-to-network mapping

0.6 think about readout issues

however, you still need to use your brain…

well-established workflow:

1. **write model in PyNN**
2. **run!**

3.1 mapping tool chooses optimal placing and routing

3.2 graph model used for parameter space configuration

3.3 complex, custom-designed software takes care of communication

this is done automatically…

To-do list

Software and modeling

- Demonstrator benchmark models:
  find suitable compensation mechanisms for hardware-specific distortions
- embed input-to-network mapping optimization in mapping algorithm

Hardware and low-level software

- implement multi-Spikey environment
- get a fully functioning wafer-scale system (huge R&D effort for hardware people)
  investigate the interplay between software and actual hardware

Long-term perspectives

- multi-wafer neuromorphic computation facility

Acknowledgements

Electronic Vision(s)
Group

Acknowledgements

The FACETS Project

Links

The FACETS Project

www.facets-project.org

The Electronic Vision(s) group

www.kip.uni-heidelberg.de/cms/groups/vision/home/

PyNN

neuralensemble.org/trac/PyNN/