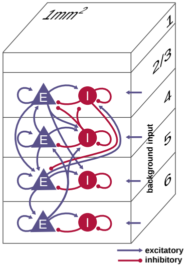
The cortical microcircuit challenge: from neuroscience to neuromorphic computing



Workshop on Theory of Neuromorphic Computing
University of Liverpool, UK
9 June 2026

Johanna Senk

School of Engineering and Informatics,
University of Sussex,
Brighton, United Kingdom

UNIVERSITY OF SUSSEX

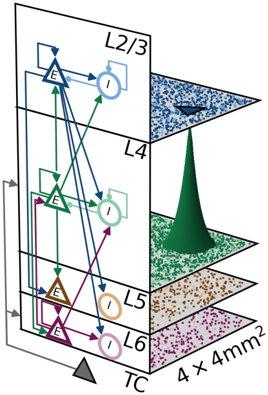
JÜLICH Forschungszentrum

Institute for Advanced Simulation (IAS-6),
Jülich Research Centre,
Jülich, Germany

Mini CV

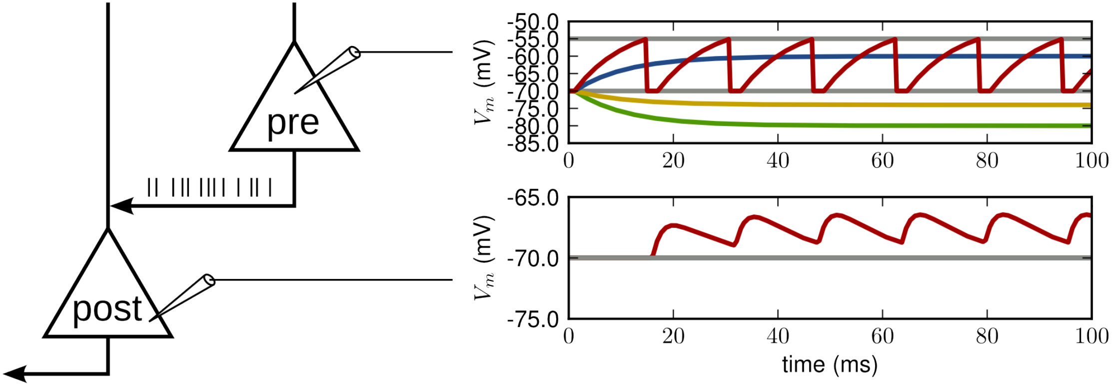
Johanna Senk

2008–2013	**B.Sc., M.Sc. Physics**, RWTH Aachen University (DE)
2011–2012	**Erasmus**, University of Trieste (IT)
since 2014	**PhD → Postdoc → Team Leader "Future Simulation Architectures"**, Jülich Research Centre (DE)
since 2024	**Assistant Professor in Computer Science**, University of Sussex (UK)
Research interests	Large-scale spiking neural network models, simulation technology, neuromorphic computing



Senk et al. (2024) Cereb. Cortex

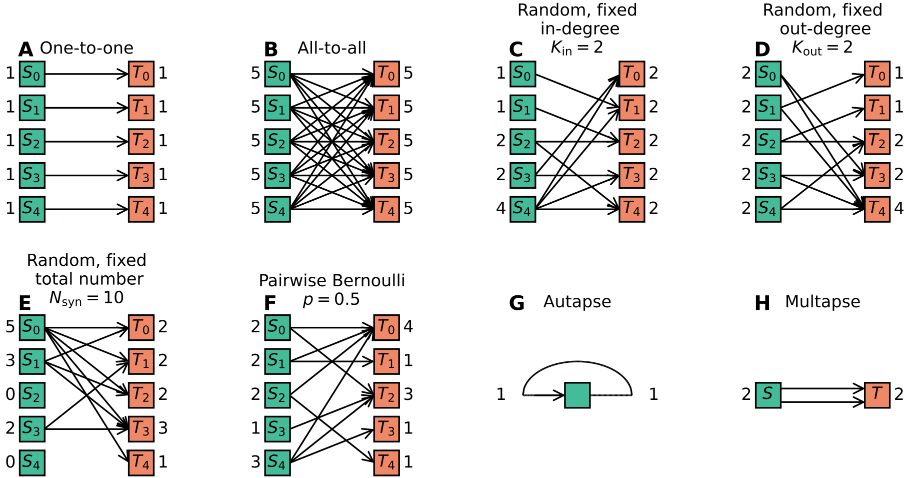
Neuron dynamics



- Simulation mimics a neuroscientific experiment
- Response of a leaky integrate-and-fire neuron to constant current input
- Subthreshold dynamics: $\tau \frac{dV_\mathrm{m}}{dt} = (E_\mathrm{L} - V_\mathrm{m}) + RI$

UNIVERSITY OF SUSSEX JÜLICH Forschungszentrum

Network connectivity



Senk et al. (2022) PLOS CB

NEST = NEural Simulation Tool - `https://www.nest-simulator.org`

- Simulator for **spiking neuronal networks with a focus on the dynamics, size, and structure of neuronal systems**, rather than on the exact morphology of individual neurons
  - Point-neurons or neurons with few electrical compartments

- Development is driven by scientific needs in consideration of accuracy and flexibility
- Same code from laptops to supercomputers → simulation of large-scale models
- C++ core, hybrid parallelization (OpenMP+MPI), Python frontend **PyNEST**

```
nest.Create()
nest.Connect(pre, post, conn_spec, syn_spec)
nest.Simulate()
```

- **NEST GPU**: NeuronGPU became part of NEST Initiative

  Tiddia, Golosio, ..., Senk, ..., van Albada (2022) Front. Neuroinform.

  Golosio, Villamar, Tiddia, ..., Senk (2023) Appl. Sci.

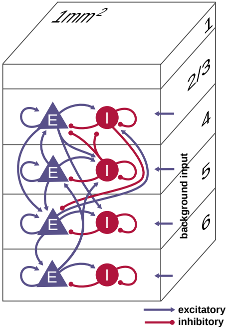
  Golosio, Tiddia, Villamar, ..., Senk (2026) Neuromorph. Comput. Eng.
- **NEST Desktop**: Graphical User Interface (GUI)

  Spreizer, Senk et al. (2021) eNeuro; Senk et al. (2018) Front. Neuroinform.
- **NESTML** Domain-specific language for neuron and synapse models

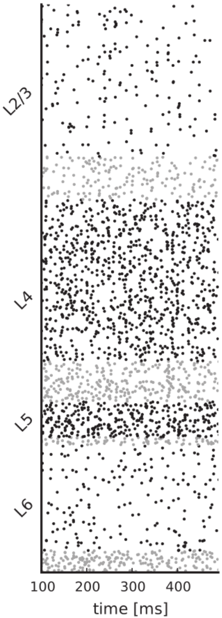
  Linssen et al. (2025) Front. Neuroinform.

...

PD14 model



Potjans & Diesmann (2014)
Cereb. Cortex



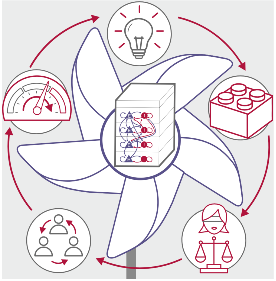
- **Canonical microcircuit**: basic architecture is conserved across species and modalities
  $\rightarrow$ generic computing principles of the brain

- **PD14**: all neurons (ca. 80,000) and synapses (ca. 300 million) below 1 mm^2 of cortical surface; reproduces basic features of cortical activity such as asynchronous and irregular spiking at biologically plausible population-specific rates

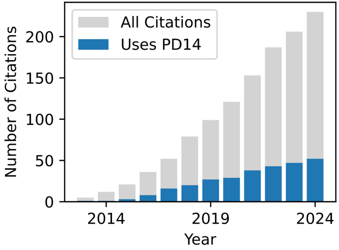
Building on models—a perspective for computational neuroscience

Hans Ekkehard Plesser[1,2,3], Andrew P. Davison[4], Markus Diesmann[2,5,6,7,*], Tomoki Fukai[8], Tobias Gemmeke[9], Padraig Gleeson[10], James C. Knight[11], Thomas Nowotny[11], Alexandre René[12], Oliver Rhodes[13], Antonio C. Roque[14], Johanna Senk[2,11], Tilo Schwalger[15,16], Tim Stadtmann[9], Gianmarco Tiddia[17], and Sacha J. van Albada[2,18]

Cereb. Cortex (2025)



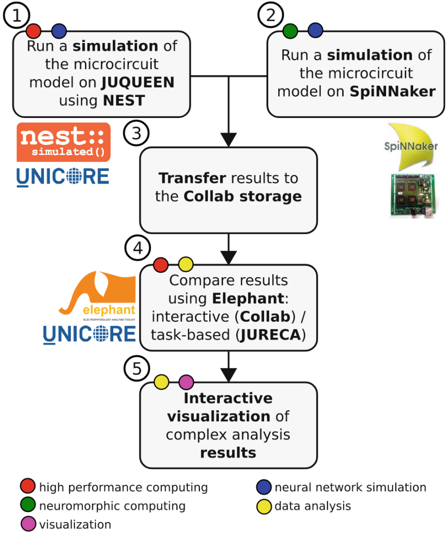
- The model was conceived to provide **neuroscience insight**,
- Became a **building block** of more advanced models,
- Served as **reference** for the validation of mean-field theories,
- Drove the development of methods for **model sharing**, and
- Became a standard for **benchmarking** of neuromorphic and conventional systems.



A Collaborative Simulation-Analysis Workflow for Computational Neuroscience Using HPC

Johanna Senk[1(✉)], Alper Yegenoglu[1(✉)], Olivier Amblet[2], Yury Brukau[2], Andrew Davison[3], David Roland Lester[4], Anna Lührs[5], Pietro Quaglio[1], Vahid Rostami[1], Andrew Rowley[4], Bernd Schuller[5], Alan Barry Stokes[4], Sacha Jennifer van Albada[1], Daniel Zielasko[6,7], Markus Diesmann[1,8,9], Benjamin Weyers[6,7], Michael Denker[1], and Sonja Grün[1,10]

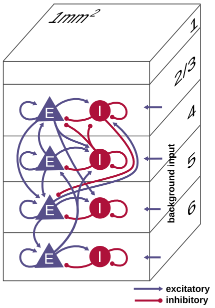
High-Performance Scientific Computing, Springer (2017)



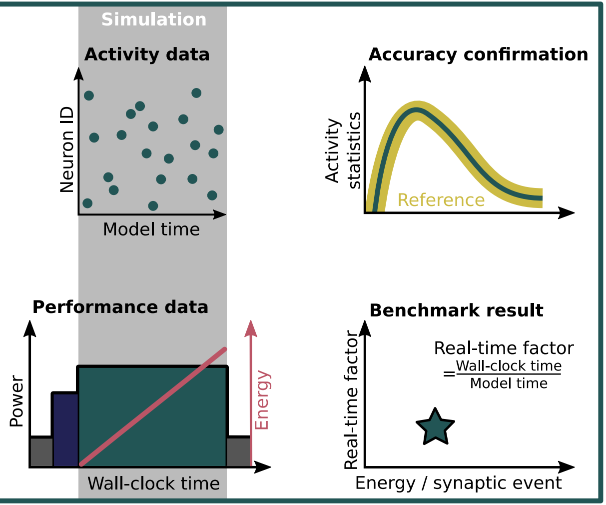
A naive question:

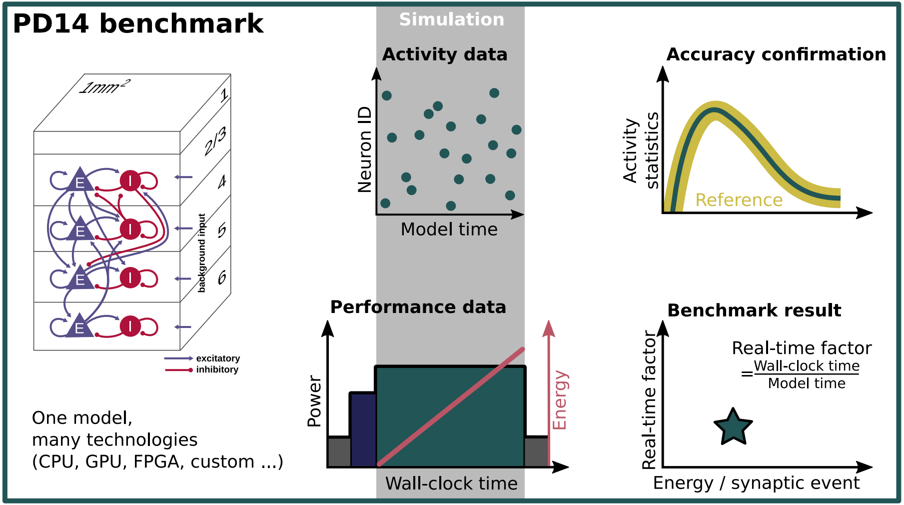
"If a simulation of the same neural network model is run both on an HPC system using NEST and on the neuromorphic hardware system SpiNNaker, are the results the same?"

PD14 benchmark



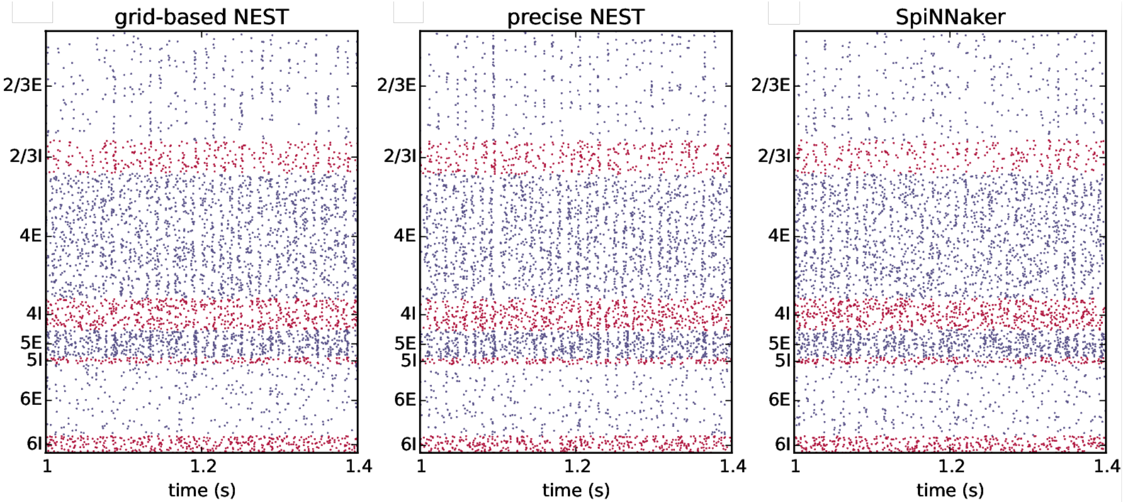
One model,
many technologies
(CPU, GPU, FPGA, custom ...)



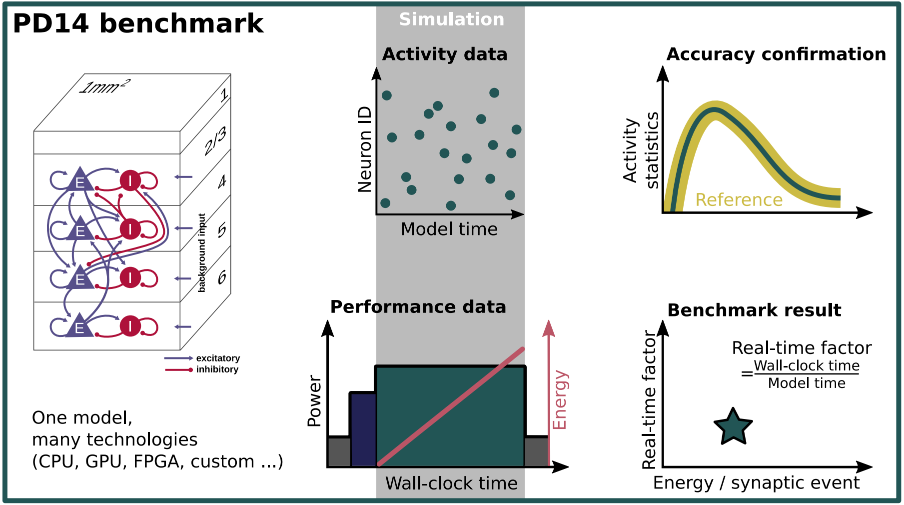


- **Simulation result**: spike times of each neuron over biological model time

Activity data



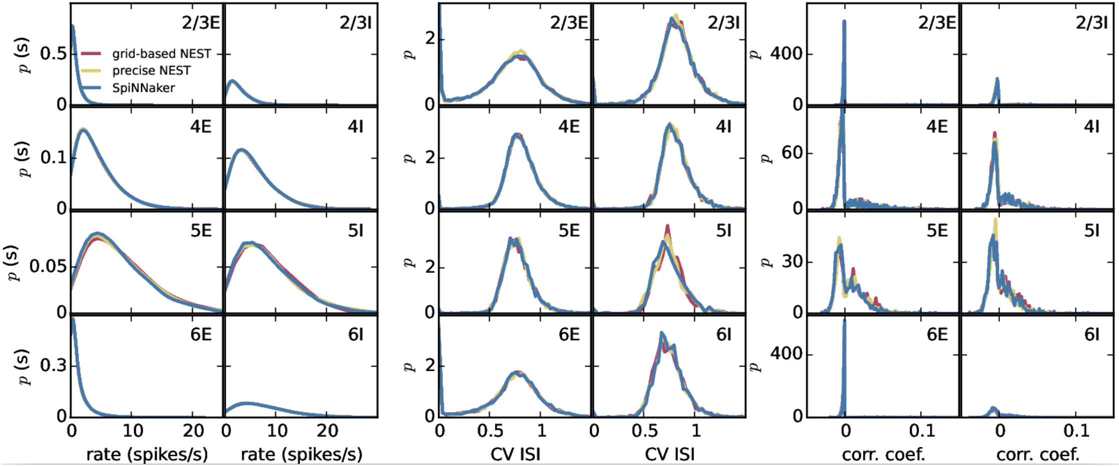
van Albada, Rowley, Senk et al. (2018) Front. Neurosci.



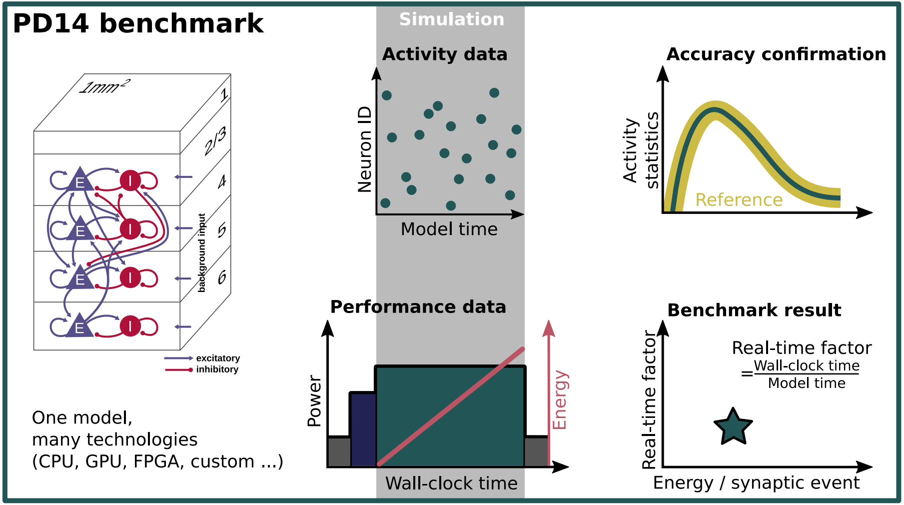
- **Simulation result**: spike times of each neuron over biological model time
- **Accuracy confirmation:** statistics of simulated activity is compared to reference data

UNIVERSITY OF SUSSEX
JÜLICH Forschungszentrum

Accuracy confirmation



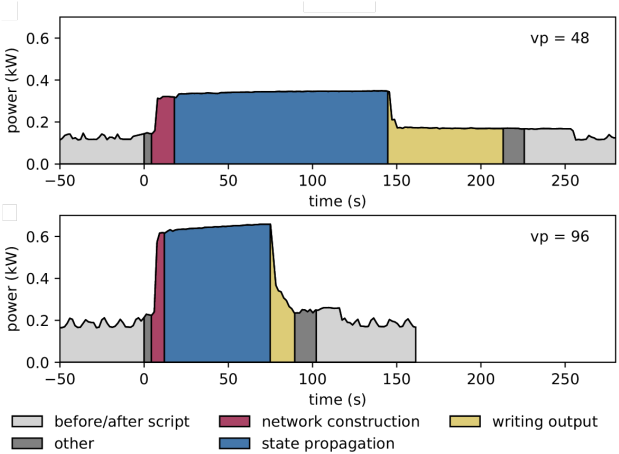
van Albada, Rowley, Senk et al. (2018) Front. Neurosci.



- Most neuromorphic benchmarks: metric is accuracy to solve a particular task, here: accuracy is starting point
- **Performance data:** power consumption and wall-clock time

UNIVERSITY OF SUSSEX JÜLICH Forschungszentrum

Performance data



van Albada, Rowley, Senk et al. (2018) Front. Neurosci.

Energy per synaptic event

$$E_{\mathrm{syn}} = \frac{\int_0^{T_{\mathrm{wall}}} P(t)\,\mathrm{d}t}{T_{\mathrm{model}} \cdot \sum_\alpha N_\alpha \cdot K_{\mathrm{out},\alpha} \cdot \nu_\alpha}$$

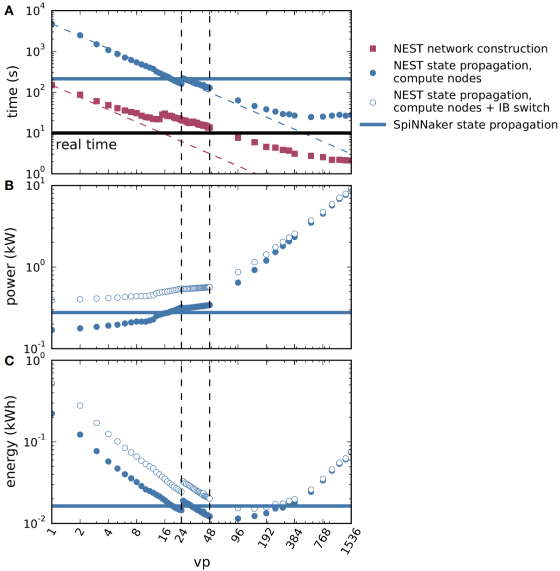
- PD14 model:
  ca. 1 billion synaptic events
  per second of model time

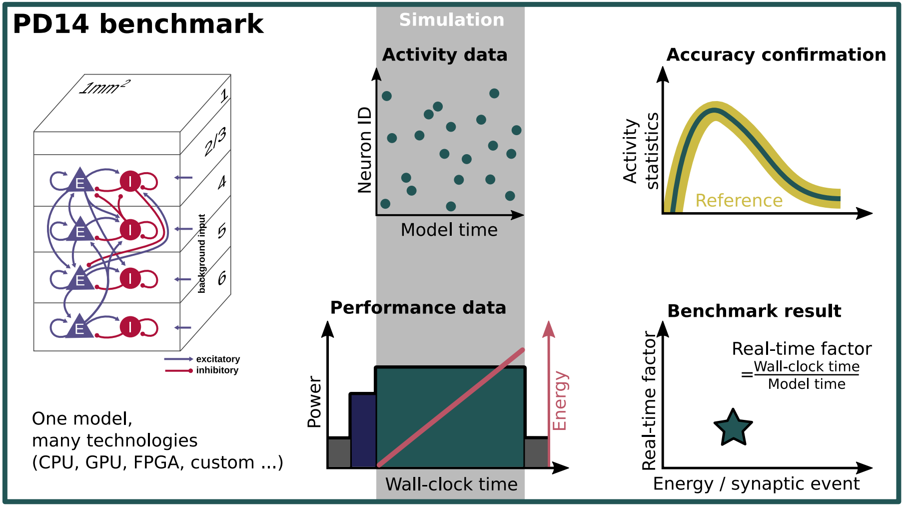
Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model

Sacha J. van Albada[1], Andrew G. Rowley[2], Johanna Senk[1], Michael Hopkins[2], Maximilian Schmidt[1,3], Alan B. Stokes[2], David R. Lester[2], Markus Diesmann[1,4,5] and Steve B. Furber[2]*

Front. Neurosci. (Section: Neuromorphic Engineering) (2018)

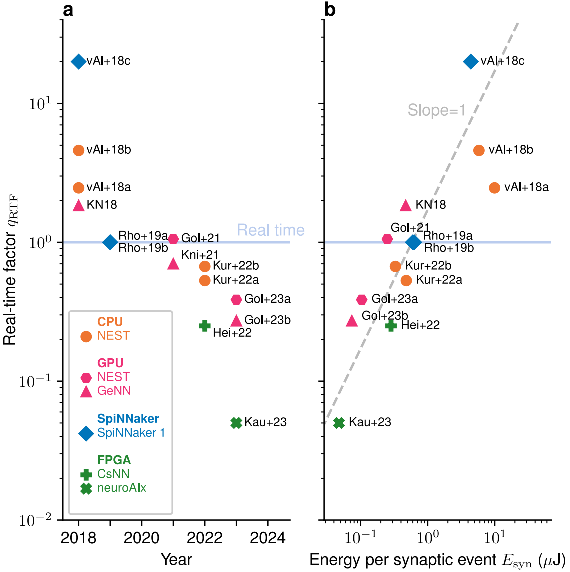
- NEST ca. 2.5 (4.6) times slower than real time
- Energy per synaptic event: 10.0 (5.8) μJ





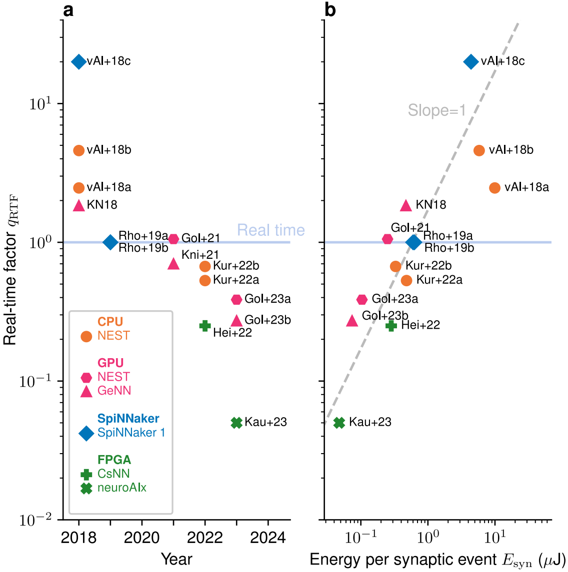
- Most neuromorphic benchmarks: metric is accuracy to solve a particular task, here: accuracy is starting point
- **Performance data:** power consumption and wall-clock time
- **Benchmark result:** time and energy to solution

Benchmark results

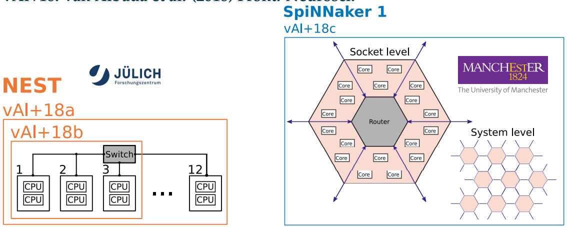


- In only **a few years**:
  simulations became
  two orders of magnitude **faster** and
  two orders of magnitude more **energy efficient**
- Authors of independent research studies came together to write a review article

Benchmark results: Initial data



vAl+18: van Albada et al. (2018) Front. Neurosci.

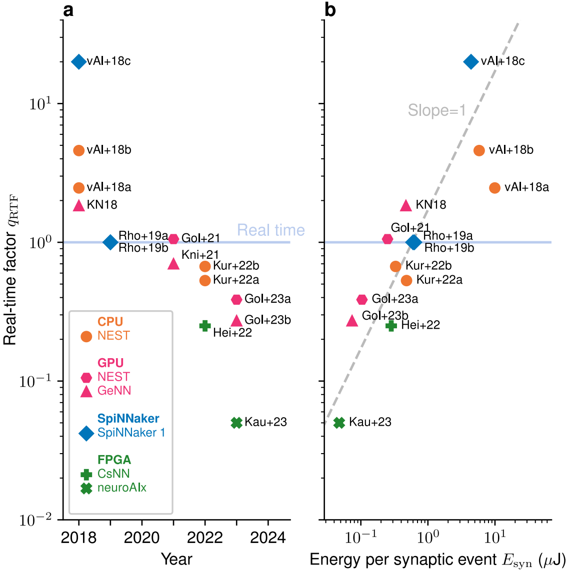


- Established CPU code shows highest energy cost
- SpiNNaker meets accuracy, but with slowdown

KN18: Knight & Nowotny (2018) Front. Neurosci.

- Emerging GPU code outperforms

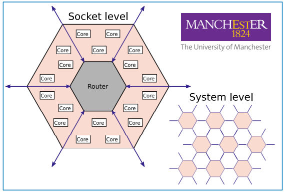
Benchmark results: Real-time barrier



Rho+19: Rhodes et al. (2019) Philos. Trans. R. Soc. A.

SpiNNaker 1

Rho+19

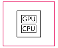


- SpiNNaker reaches its real-time specification

Gol+21: Golosio et al. (2021) Front. Comput. Neurosci.; Kni+21: Knight et al. (2021) Front. Neuroinform.; Gol+23: Golosio, Villamar, Tiddia et al. (2023) Appl. Sci.

NEST GPU, GeNN

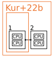
Gol+21, Kni+21, Gol+23



NEST CPU

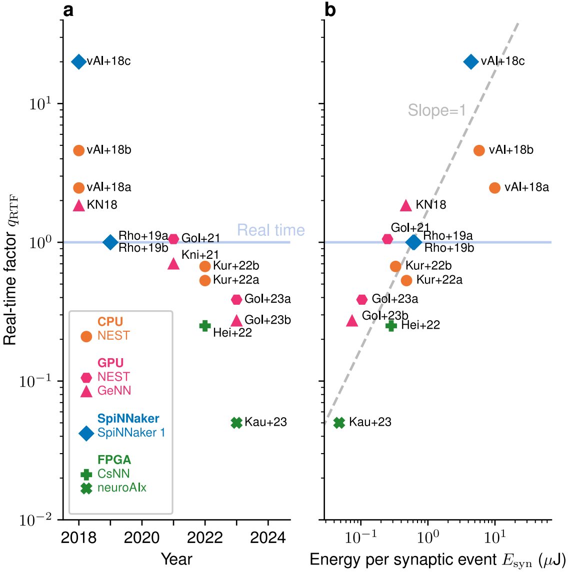
Kur+22a

Kur+22b

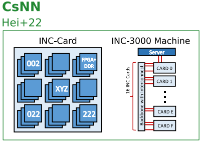


- Conventional technologies approach and surpass real-time: new GPU code, sub-realtime on one commodity CPU

Benchmark results: FPGAs

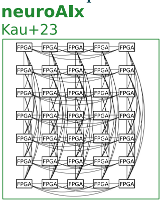


Hei+22: Heittmann et al. (2022) Front. Neurosci.



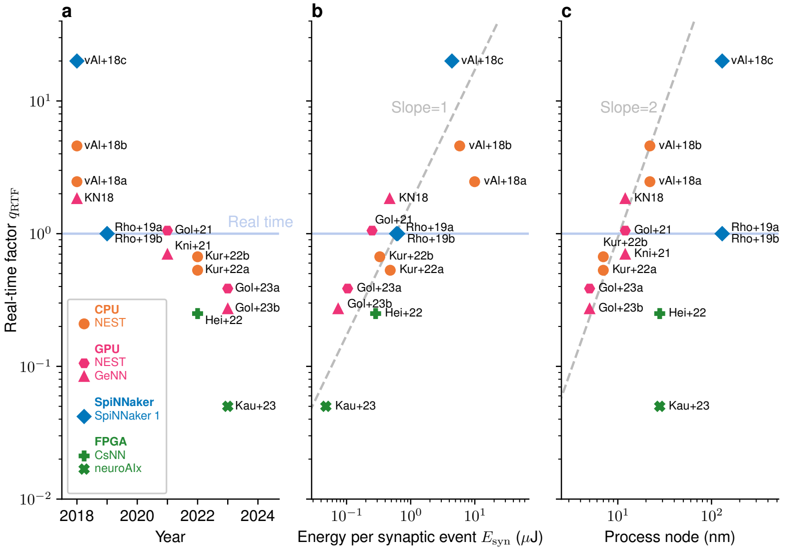
- Procedural connectivity

Kau+23: Kauth et al. (2023), Front. Comput. Neurosci.



- Jump to 20× faster than real-time

Semiconductor technology



- Separation of conventional and dedicated hardware (system effects not resolved)
- 100× faster than real-time seems plausible with today's technologies and custom software-hardware designs!

UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

IOP Publishing Neuromorph. Comput. Eng. 6 (2026) 012001 https://doi.org/10.1088/2634-4386/ae379a

NEUROMORPHIC
Computing and Engineering

Check for updates

TOPICAL REVIEW

OPEN ACCESS

RECEIVED
5 August 2025

REVISED
16 December 2025

ACCEPTED FOR PUBLICATION
13 January 2026

PUBLISHED
16 February 2026

Constructive community race: full-density spiking neural network model drives neuromorphic computing

Johanna Senk[1,2,*], Anno C Kurth[2,15], Steve Furber[3], Tobias Gemmeke[4], Bruno Golosio[5,6], Arne Heittmann[7], James C Knight[1], Eric Müller[8], Tobias Noll[9], Thomas Nowotny[1], Gorka Peraza Coppola[2,9], Luca Peres[3], Oliver Rhodes[3], Andrew Rowley[3], Johannes Schemmel[10], Tim Stadtmann[4], Tom Tetzlaff[2], Gianmarco Tiddia[6], Sacha J van Albada[2,11], José Villamar[2,9] and Markus Diesmann[2,12,13,14]

1 Sussex AI, School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom
2 Institute for Advanced Simulation (IAS-6), Jülich Research Centre, Jülich, Germany
3 Department of Computer Science, University of Manchester, Manchester, United Kingdom
4 Lehrstuhl für Integrierte Digitale Systeme und Schaltungsentwurf (IDS), RWTH Aachen University, Aachen, Germany
5 Department of Physics, University of Cagliari, Monserrato, Italy
6 Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Cagliari, Monserrato, Italy
7 Neuromorphic Software Ecosystems (PGI-15), Jülich Research Centre, Jülich, Germany
8 Kirchhoff Institute for Physics, Heidelberg University, Heidelberg, Germany
9 RWTH Aachen University, Aachen, Germany
10 Institute of Computer Engineering, Heidelberg University, Heidelberg, Germany
11 Institute of Zoology, University of Cologne, Cologne, Germany
12 JARA-Institute Brain Structure-Function Relationships (INM-10), Jülich Research Centre, Jülich, Germany
13 Department of Psychiatry, Psychotherapy and Psychosomatics, School of Medicine, RWTH Aachen University, Aachen, Germany
14 Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany
15 Current address: Hierarchical Neural Computation RIKEN ECL Research Unit, RIKEN Center for Brain Science, Wako, Japan.
* Author to whom any correspondence should be addressed.

E-mail: j.senk@sussex.ac.uk

UNIVERSITY OF SUSSEX JÜLICH Forschungszentrum

Lessons learned: Benchmarking recipe

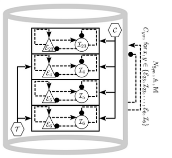
Model reference	`https://github.com/INM-6/microcircuit-PD14-model`
External drive	State whether DC or Poisson is used
Initial conditions	Amended initial conditions: distribute membrane potentials normally with population-specific mean and variance
Warm-up time	Discard the initial 500 ms of model time from the data to be analyzed
Simulation duration	Accuracy: $T_{\text{model}} = 15\,\text{min}$, performance: $T_{\text{model}} \geq 10\,\text{s}$
Repeated simulations	Average across ten random seeds
Spike recording	Accuracy: yes, performance: no
Accuracy	Compute distributions of 1) single-neuron *firing rate* (FR), 2) *coefficient of variation* (CV) of the inter-spike intervals (ISI), and 3) short-term spike-count *correlation coefficients* (CC), and compare with reference data
Performance	Measure real-time factor q_{RTF} and the energy per synaptic event E_{syn} (include all contributions necessary for running the simulations at the power outlet)

UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

Model reference - https://github.com/INM-6/microcircuit-PD14-model

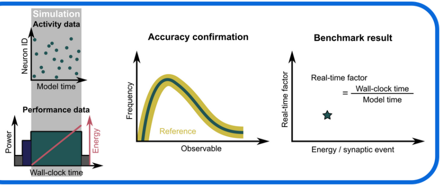
Model description



$\mathcal{I}_5$	0.0548	0.0269	0.0257	0.0022	0.06
$\mathcal{E}_6$	0.0156	0.0066	0.0211	0.0166	0.05
$\mathcal{I}_6$	0.0364	0.0010	0.0034	0.0005	0.02

Neuron		
Name	**Value**	**Description**
θ	-50 mV	spike threshold
E_L	-65 mV	resting potential
τ_m	10 ms	membrane time constant
C_m	250 pF	membrane capacitance
R_m	$\tau_m/C_m = 40\,\mathrm{M\Omega}$	membrane resistance

Performance benchmarking



Model implementation

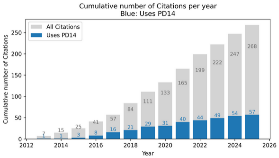
```
pip install .
```

```python
from microcircuit import network
from microcircuit.network_params import default_net_dict as net_pars
from microcircuit.sim_params import default_sim_dict as sim_pars
from microcircuit.stimulus_params import default_stim_dict as stim_pars

model = network.Network(sim_pars, net_pars, stim_pars)
model.create()
model.connect()
model.simulate(1000)
```

nest::simulated()

Publications

```
@article{Sinha_2025,
  title={The NeuroML ecosystem for standardized multi-scale
  volume={13},
  ISSN={2050-084X},
  url={http://dx.doi.org/10.7554/elife.95135.3},
  DOI={10.7554/elife.95135.3},
  journal={eLife},
  publisher={eLife Sciences Publications, Ltd},
  author={Sinha, Ankur and Gleeson, Padraig and Marin, Bóris
  year={2025},
  month=jan,
  uses_PD14 = {no}
}

@article{Jiang_2024,
  title={A layered microcircuit model of somatosensory corte
  volume={34},
  ISSN={1460-2199},
  url={http://dx.doi.org/10.1093/cercor/bhae378},
  DOI={10.1093/cercor/bhae378},
  number={9},
  journal={Cerebral Cortex},
  publisher={Oxford University Press (OUP)},
  author={Jiang, Han-Jia and Qi, Guanxiao and Duarte, Renato
  year={2024},
  month=sep,
  uses_PD14 = {yes}
}
```



New data points of performance benchmarks are welcome!

UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

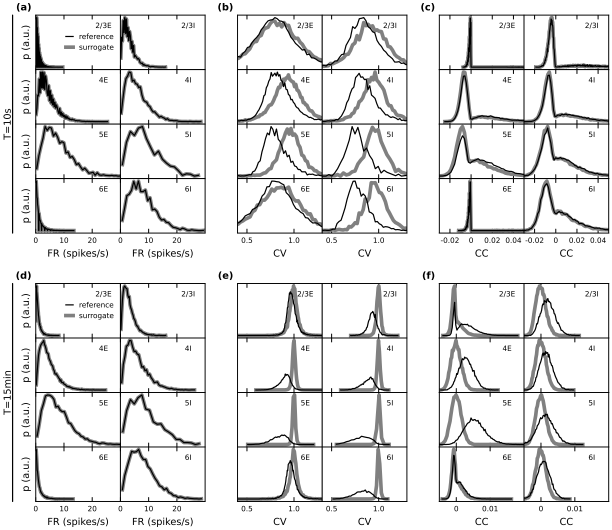
Lessons learned: Benchmarking recipe

Model reference	`https://github.com/INM-6/microcircuit-PD14-model`
External drive	State whether DC or Poisson is used
Initial conditions	Amended initial conditions: distribute membrane potentials normally with population-specific mean and variance
Warm-up time	Discard the initial 500 ms of model time from the data to be analyzed
Simulation duration	Accuracy: $T_{\mathrm{model}} = 15\,\mathrm{min}$, performance: $T_{\mathrm{model}} \geq 10\,\mathrm{s}$
Repeated simulations	Average across ten random seeds
Spike recording	Accuracy: yes, performance: no
Accuracy	Compute distributions of 1) single-neuron *firing rate* (FR), 2) *coefficient of variation* (CV) of the inter-spike intervals (ISI), and 3) short-term spike-count *correlation coefficients* (CC), and compare with reference data
Performance	Measure real-time factor q_{RTF} and the energy per synaptic event E_{syn} (include all contributions necessary for running the simulations at the power outlet)

UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

Specificity of validation measures



- Surrogate data by randomizing spike times
- Role of observation duration (convergence of distributions)

Dasbach, Tetzlaff, Diesmann & Senk (2021) Front. Neurosci.

Lessons learned: Benchmarking recipe

Model reference	`https://github.com/INM-6/microcircuit-PD14-model`
External drive	State whether DC or Poisson is used
Initial conditions	Amended initial conditions: distribute membrane potentials normally with population-specific mean and variance
Warm-up time	Discard the initial 500 ms of model time from the data to be analyzed
Simulation duration	Accuracy: $T_{\mathrm{model}} = 15\,\mathrm{min}$, performance: $T_{\mathrm{model}} \geq 10\,\mathrm{s}$
Repeated simulations	Average across ten random seeds
Spike recording	Accuracy: yes, performance: no
Accuracy	Compute distributions of 1) single-neuron *firing rate* (FR), 2) *coefficient of variation* (CV) of the inter-spike intervals (ISI), and 3) short-term spike-count *correlation coefficients* (CC), and compare with reference data
Performance	Measure real-time factor q_{RTF} and the energy per synaptic event E_{syn} (include all contributions necessary for running the simulations at the power outlet)

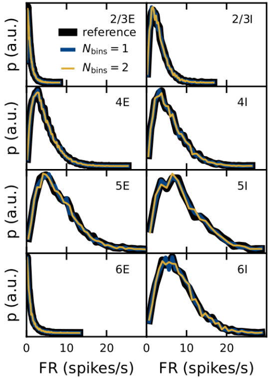
UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

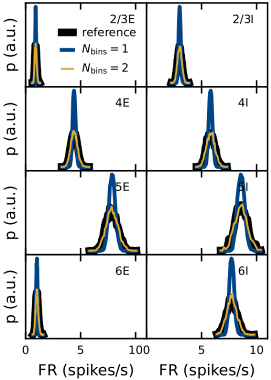
Connectivity vs. synapse parameters

Sufficiently heterogeneous in-degrees → firing statistics can be preserved even if all synaptic weights are replaced by the mean of the weight distribution

Fixed total number



Fixed in-degree

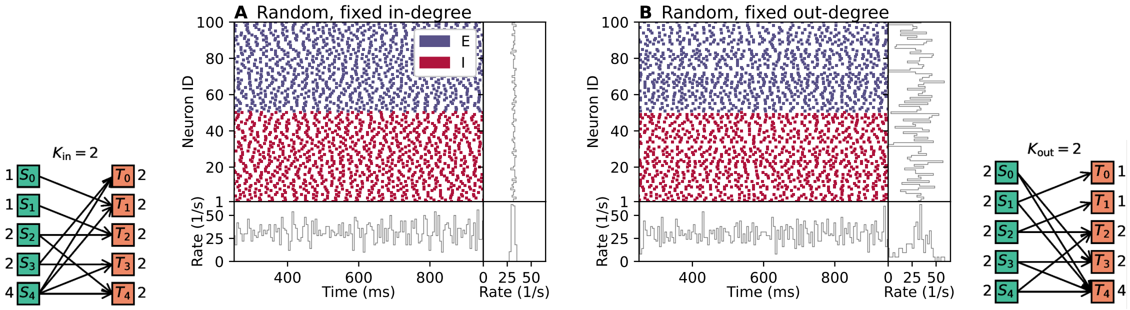


Dasbach, Tetzlaff, Diesmann & Senk (2021) Front. Neurosci

Are the accuracy metrics sufficient?

In the spirit of the model, all synapses need to be represented (to allow for plasticity).

Connectivity details matter for dynamics!



- Higher inter-neuron variability in networks connected with fixed out-degree compared to fixed in-degree

Senk et al. (2022) PLOS CB

RESEARCH ARTICLE

Connectivity concepts in neuronal network modeling

Johanna Senk[1 *], Birgit Kriener[2], Mikael Djurfeldt[3], Nicole Voges[4], Han-Jia Jiang[1,5], Lisa Schüttler[6], Gabriele Gramelsberger[6], Markus Diesmann[1,7,8], Hans E. Plesser[1,9], Sacha J. van Albada[1,5]

PLOS CB (2022)

- **Review:**
  Which **connectivity structures** do computational neuroscientists use in their models?
  Which **terminology** do they use to describe these models?
- **Ambiguity**:
  Left-out details about mathematical concepts and assumptions, algorithmic implementations, or parameterizations are common and **hinder reproducibility**.

Systematic description of connectivity concepts

Random, fixed in-degree with multapses

Symbol: K_{in}, M

CSA: $\rho_1(K)\mathbf{M}(\mathbb{N}_S \times \mathbb{N}_T)$

Definition: Each target node in $\mathcal{T}$ is connected to K_{in} nodes in $\mathcal{S}$ randomly chosen with replacement.

N_s is the number of source nodes from which exactly K_{in} connections are drawn with equal probability $p = 1/N_s$ for each of the N_t target nodes $t_i \in \mathcal{T}$. The in-degree distribution is by definition $P(K) = \delta_{K,K_{\text{in}}}$. To obtain the out-degree distribution, we observe that because multapses are allowed, drawing N_t times $K_{\text{in},i} = K_{\text{in}}$ from $\mathcal{S}$ is equivalent to drawing $N_t K_{\text{in}}$ times with replacement from $\mathcal{S}$. This procedure yields a multinomial distribution of the out-degrees $K_{\text{out},j}$ of source nodes $s_j \in \mathcal{S}$ [75], i.e.,

$$P(K_{\text{out},1} = K_1, \ldots, K_{\text{out},N_s} = K_{N_s}) = \begin{cases} \frac{(N_t K_{\text{in}})!}{K_1! \ldots K_{N_s}!} p^{N_t K_{\text{in}}} & \text{if } \sum_{j=1}^{N_s} K_j = N_t K_{\text{in}} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The marginal distributions are binomial distributions

$$P(K_{\text{out},j} = K) = \mathcal{B}(K | N_t K_{\text{in}}, 1/N_s)\,. \quad (15)$$

Senk et al. (2022) PLOS CB

Living reference

Connectivity concepts in NEST simulator documentation

Connectivity concepts   NEST Simulator "600 Stars · 387 Forks"

NEST Simulator Documentation

USAGE

- Install
- Tutorials and Guides
- Examples
- Models
- Python API
- Glossary
- NEST performance benchmarks
- Cite NEST
- License

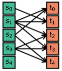
COMMUNITY

- Contact us
- Contribute
- What's new?
- NEST Homepage
- Acknowledgments

RELATED PROJECTS

- NEST Desktop
- NESTML
- NESTGPU
- NEST NEAT

Random, fixed in-degree



The nodes in `S` are randomly connected with the nodes in `T` such that each node in `T` has a fixed `indegree` of `N`.

As multapses are per default allowed and possible with this rule, you can disallow them by adding `'allow_multapses': False` to the `conn_dict`.

```
n, m, N = 5, 5, 2
S = nest.Create('iaf_psc_alpha', n)
T = nest.Create('iaf_psc_alpha', m)
conn_spec = {'rule': 'fixed_indegree', 'indegree': N}
nest.Connect(S, T, conn_spec)
```

Mathematical details: Random, fixed in-degree with multapses

Symbol: K_{in}, M
CSA: $\rho_1(K)\mathbf{M}(\mathbb{N}_S \times \mathbb{N}_T)$
Definition: Each target node in $\mathcal{T}$ is connected to K_{in} nodes in $\mathcal{S}$ randomly chosen with replacement.
N_s is the number of source nodes from which exactly K_{in} connections are drawn with equal probability $p = 1/N_s$ for each of the N_t target nodes $t_i \in \mathcal{T}$. The in-degree distribution is by definition $P(K) = \delta_{K,K_{\text{in}}}$. To obtain the out-degree distribution, we observe that because multapses are allowed, drawing N_t times $K_{\text{in},i} = K_{\text{in}}$ from $\mathcal{S}$ is equivalent to drawing $N_t K_{\text{in}}$ times with replacement from $\mathcal{S}$. This procedure yields a multinomial distribution of the out-degrees $K_{\text{out},j}$ of source nodes $s_j \in \mathcal{S}$ [5], i.e.,

$$P(K_{\text{out},1} = K_1, \ldots, K_{\text{out},N_s} = K_{N_s}) = \begin{cases} \frac{(N_t K_{\text{in}})!}{K_1! \ldots K_{N_s}!} p^{N_t K_{\text{in}}} & \text{if } \sum_{j=1}^{N_s} K_j = N_t K_{\text{in}} \\ 0 & \text{otherwise} \end{cases}$$

The marginal distributions are binomial distributions

$$P(K_{\text{out},j} = K) = \mathcal{B}(K|N_t K_{\text{in}}, 1/N_s)\,.$$

Page Contents

https://nest-simulator.readthedocs.io/en/stable/synapses/connectivity_concepts.html#connectivity-concepts

UNIVERSITY OF SUSSEX   JÜLICH Forschungszentrum

Graphical notation for network diagrams

Network node

Node class	
Individual unit	
Population	
Node type	
Generic node	
Excitatory neural node	
Inhibitory neural node	
Stimulating device node	
Recording device node	

Network edge

Determinism	
Deterministic	
Probabilistic	
Edge type	
Generic edge	
Excitatory edge	
Inhibitory edge	
Directionality	
Unidirectional	
Bidirectional	

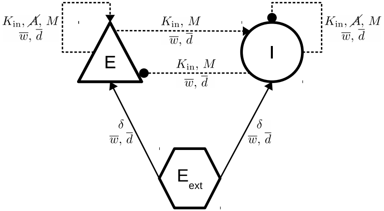
Senk et al. (2022) PLOS CB

Annotation

Connectivity concept	
Concept	
One-to-one	δ
All-to-all	Ω
Explicit connections	X
Pairwise Bernoulli	p
Random, fixed total number	N_{syn}
Random, fixed in-degree	K_{in}
Random, fixed out-degree	K_{out}
Constraint	
Autapses allowed	A
Multapses allowed	M
Prohibited	$\not{A}, \not{M}$
Parameterization	
Constant parameter	$\overline{w}$
Distributed parameter	$w \sim \mathcal{D}$
Further specification	
Functional dependence	$f(\cdot)$

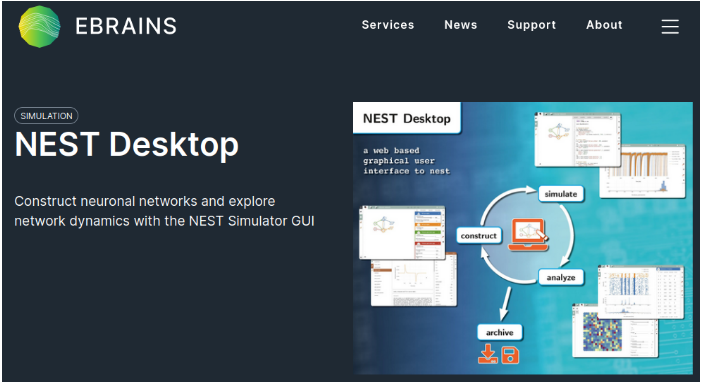
Example

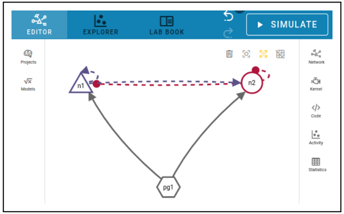
$\mathcal{S}$ <Concept> <Constraint> <Parameterization> $\mathcal{T}$



NEST Desktop

NEST Simulator GUI adopts proposed graphical notation





Spreizer, Senk, Rotter, Diesmann & Weyers (2021) eNeuro

Living reference:

`https://nest-desktop.readthedocs.io/en/latest/user/usage-advanced/network-graph.html`

UNIVERSITY OF SUSSEX

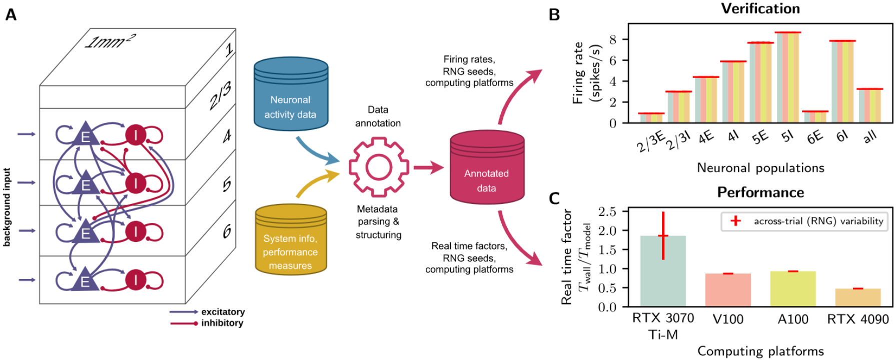
JÜLICH Forschungszentrum

OPEN ARTICLE

Metadata practices for simulation workflows

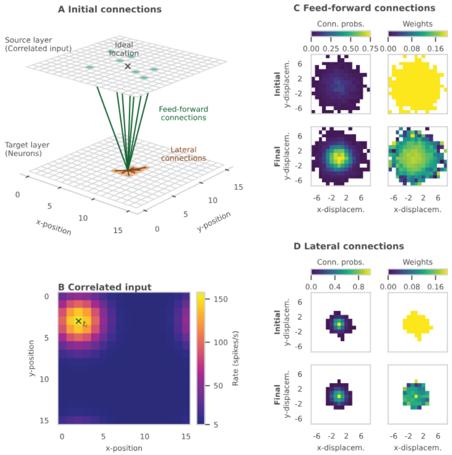
José Villamar[1,2] ✉, Matthias Kelbling[3], Heather L. More[1,4], Michael Denker[1], Tom Tetzlaff[1], Johanna Senk[1,5] & Stephan Thober[3]

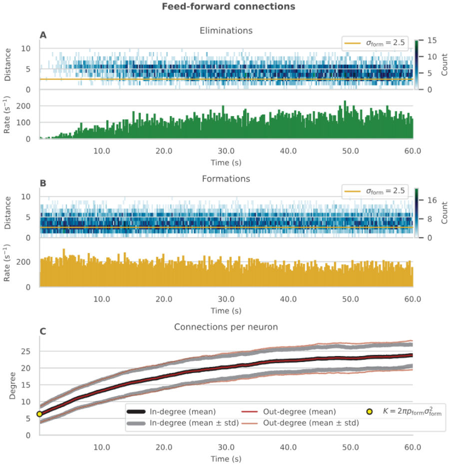
Scientific Data (2025)



UNIVERSITY OF SUSSEX
JÜLICH Forschungszentrum

Next: more complex dynamics and slow processes



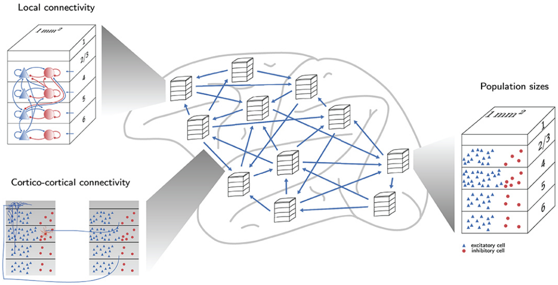


- Refinement of a topographic map using a **structural plasticity rule** with activity-dependent and activity-independent contributions

Knight, Senk & Nowotny (2026) Neuromorph. Comput. Eng.

Next: larger models

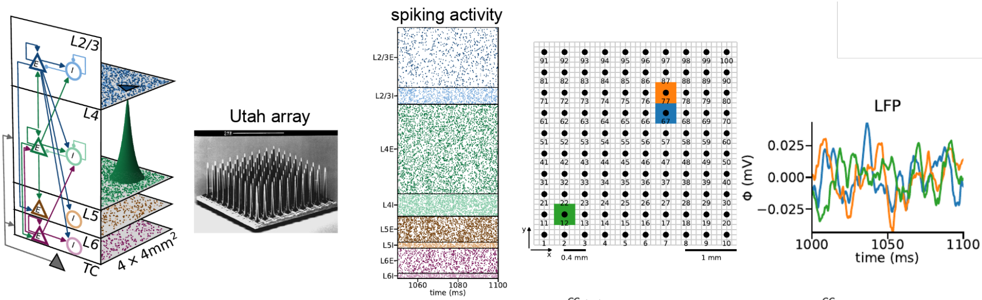
Multi-area model: 32 vision-related areas of macaque monkey



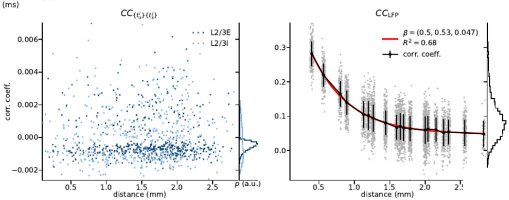
- Model development and NEST simulation:
  Schmidt et al. (2018) PLoS Comput. Biol.
- GeNN (procedural connectivity):
  Knight & Nowotny (2021) Nat. Comput. Sci.
- NEST GPU:
  Tiddia, ..., Senk, et al. (2022) Front. Neuroinform.

Next: larger models

Mesocircuit: spatially organised cortical network model



- 4x4 mm², 3.2 million neurons, 7.7 billion synapses
- Distance-dependent connectivity
- Weak spike-train correlations but highly coherent local field potentials across space

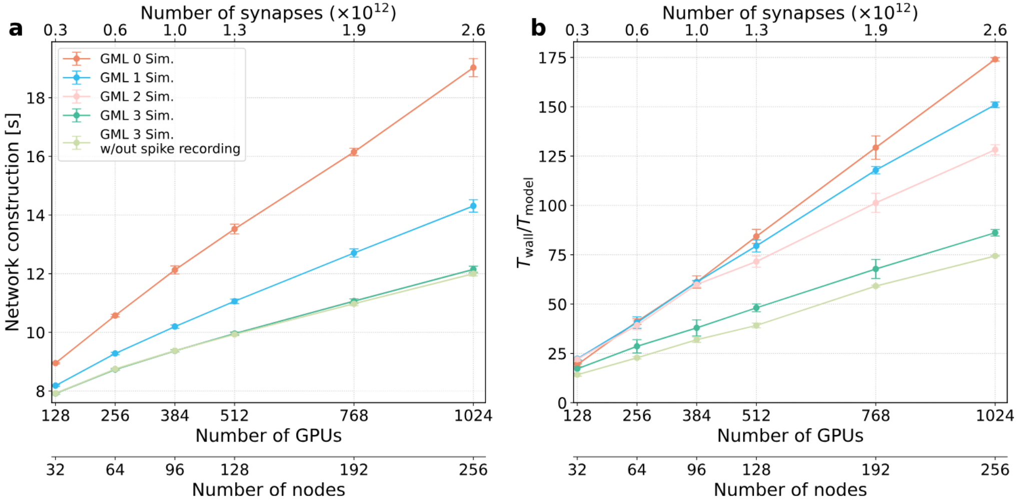


Senk, Hagen, van Albada, Diesmann (2024) Cereb. Cortex

UNIVERSITY OF SUSSEX

JÜLICH Forschungszentrum

NEST GPU on its path to exascale



Golosio, Tiddia, Villamar, ..., Senk (2026) Neuromorph. Comput. Eng.

Golosio, Villamar, Tiddia, ..., Senk (2023) Appl. Sci.

Take home message

- Neuroscience needs
  - the representation of all neurons and synapses to simulate correlation structure and plasticity,
  - significantly faster than real-time simulation for slow processes like learning, and
  - technologies capable of simulating larger and more complex models than PD14.

- Neuromorphic computing needs
  - clear requirements from neuroscience and
  - a strong theoretical foundation on a conceptual and algorithmic level.

- Structuring the unstructured community effort has shown:
  - A model meant to help understand the brain became a de facto standard benchmark
  - **There is a large and complementary potential of different simulation technologies to act as research tools for neuroscience and beyond!**

- Outlook:
  - Downscaled PD14 model: BrainScaleS already reaches $q_{\mathrm{RTF}} = 0.0001$ at $E_{\mathrm{syn}} < 0.012\,\mu\mathrm{J}$
    (Sch+25: Schmidt et al. (2025) NICE)
    but still 5-6 orders of magnitude away from biology...
  - Ongoing efforts at University of Sussex, Jülich Research Centre and many other places to form neuromorphic communities, jointly defining and advancing the field of neuromorphic computing.

UNIVERSITY OF SUSSEX   JÜLICH Forschungszentrum

Thank you!

Anno C. Kurth, Steve Furber, Tobias Gemmeke, Bruno Golosio, Arne Heittmann, James C. Knight, Eric Müller, Tobias Noll, Thomas Nowotny, Gorka Peraza Coppola, Luca Peres, Oliver Rhodes, Andrew Rowley, Johannes Schemmel, Tim Stadtmann, Tom Tetzlaff, Gianmarco Tiddia, Sacha J. van Albada, José Villamar, Markus Diesmann

Sussex AI@University of Sussex,
IAS-6@Jülich Research Centre

EBRAINS
HELMHOLTZ
Helmholtz Portfolio Theme SMHB Supercomputing and Modeling for the Human Brain
DFG Deutsche Forschungsgemeinschaft
ACA
SPONSORED BY THE Federal Ministry of Education and Research
UKRI Engineering and Physical Sciences Research Council
HBP Human Brain Project
Google Summer of Code
US University of Sussex
JÜLICH Forschungszentrum