Efficient simulations of spiking neural networks using NEST GPU

CNS 2025 Software showcase

José Villamar[1,2], **Luca Sergi**[3,4], Tiddia G.[4], Babu P.[5], Lonardo A.[6], Simula F.[6], Pontisso L.[6], Pastorelli E.[6], Paolucci P.S.[6], Golosio B.[3,4], Senk J[1,7].

1. Institute for Advanced Simulation (IAS-6) Jülich Research Centre, Jülich, Germany
2. RWTH Aachen University, Aachen, Germany
3. Dipartimento di Fisica, Università di Cagliari, Monserrato, Italy
4. Istituto Nazionale di Fisica Nucleare, Sezione di Cagliari, Monserrato, Italy
5. Simulation and Data Laboratory Neuroscience, Jülich Supercomputing Centre, Jülich Research Centre, Jülich, Germany
6. Istituto Nazionale di Fisica Nucleare, Sezione di Roma, Roma, Italy
7. Sussex AI, School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom

j.villamar@fz-juelich.de and lsergi@dsf.unica.it

JÜLICH Forschungszentrum

First things first

Materials used during the showcase are available at

Zenodo link: [10.5281/zenodo.15754814](https://doi.org/10.5281/zenodo.15754814)

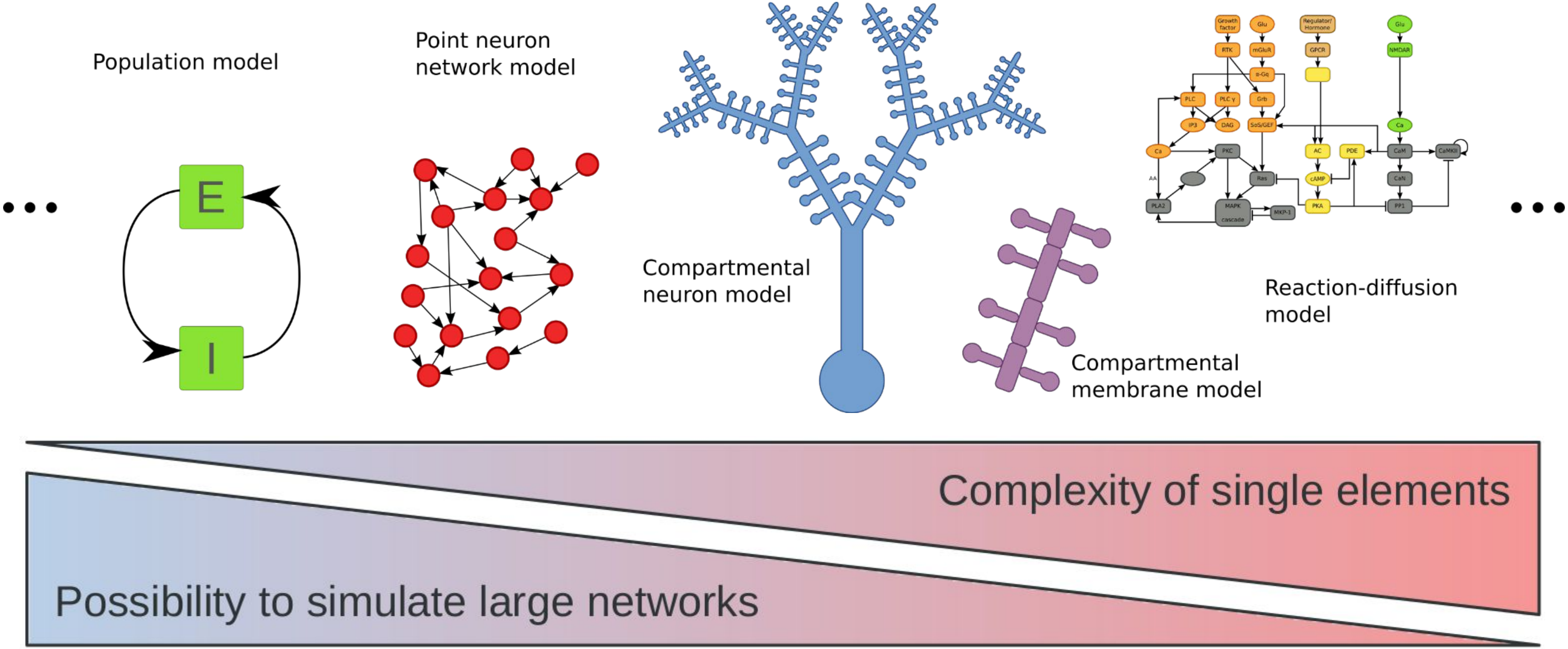
Outline

- What is NEST and what is NEST GPU?
  - Our role in the NEST initiative
- Modeling your first network with NEST GPU
  - An example implementation using the Brunel network
- First steps on spike data analysis
  - Example using first and second order statistics
- Scaling up your network on multiple GPUs
  - A sneak peek on large scale spiking neural network simulations
- Closing remarks
  - Related works and future plans with NEST GPU

Outline

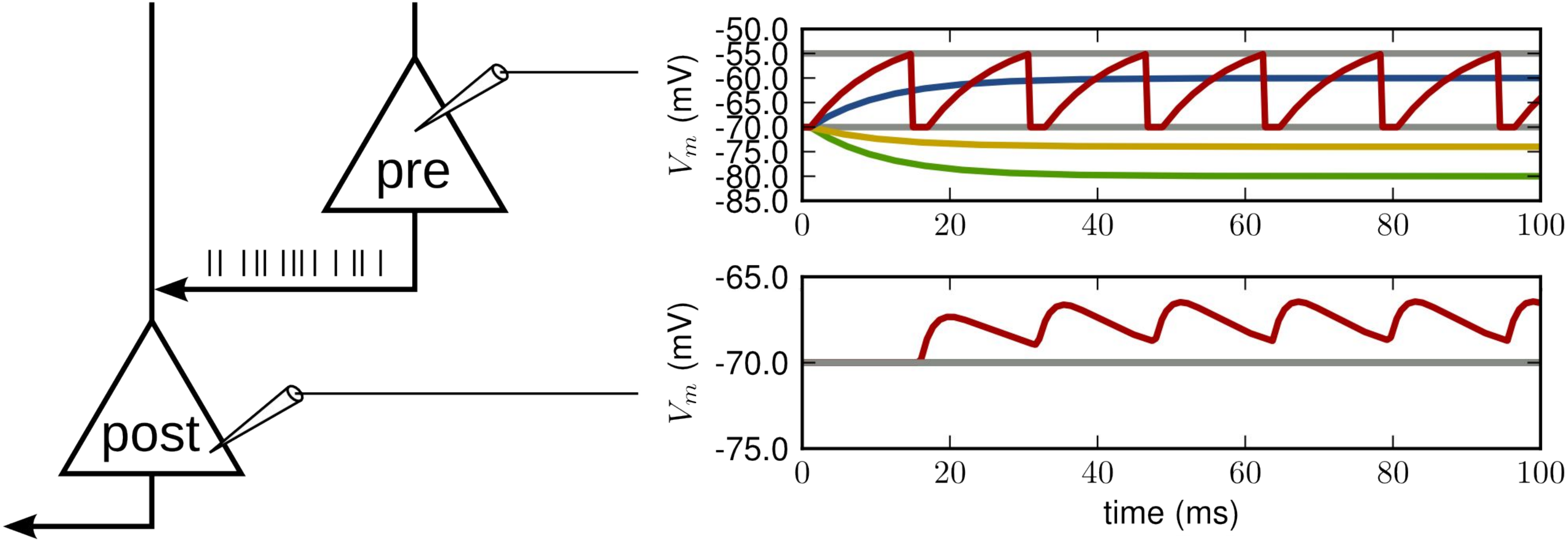
- What is NEST and what is NEST GPU?
  - Our role in the NEST initiative
- Modeling your first network with NEST GPU
  - An example implementation using the Brunel network
- First steps on spike data analysis
  - Example using first and second order statistics
- Scaling up your network on multiple GPUs
  - A sneak peek on large scale spiking neural network simulations
- Closing remarks
  - Related works and future plans with NEST GPU

Different solutions for problems with different sizes



Different solutions for problems with different sizes

Point neuron simulations mimic a neuroscientific experiment



JÜLICH Forschungszentrum

NEST == NEURAL SIMULATION TOOL

nest::

- C++ core, hybrid parallelization (OpenMP+MPI), Python frontend PyNEST
- Same code from laptops to supercomputers → simulation of large-scale models
- Development is driven by scientific needs with a focus on accuracy and flexibility as well as quality assurance

Main website:	https://www.nest-simulator.org
NEST initiative:	https://www.nest-initiative.org
Source code:	https://github.com/nest/nest-simulator
Online documentation:	https://nest-simulator.readthedocs.io
EBRAINS:	https://ebrains.eu/service/nest-simulator

NEST Initiative

Organization behind the NEST software ecosystem

- NESTML: a modeling language for neuron and synapse models
  - https://nestml.readthedocs.io/en/latest/
  - https://github.com/nest/nestml
- NEAT: framework to model and simulate simplified morphological neuron models
  - https://neatdend.readthedocs.io/en/latest/
  - https://github.com/nest/NEAT
- NEST Desktop: server backend and graphical interface for NEST
  - https://nest-desktop.readthedocs.io/en/latest/
  - https://github.com/nest-desktop/nest-desktop

And more recently **NEST GPU**…

What is NEST GPU?

NEST GPU as the GPU backend for the NEST simulator

- CUDA kernels for state update and multi-GPU support using MPI for spike communication
- Similar Python interface
- Many neuron models already implemented, e.g., IAF, conductance and current based, Izhikevich
- Synapse models: standard synapse, nearest-neighbor STDP
- Though NEST GPU is still an experimental project, we welcome new users to test our implementation!

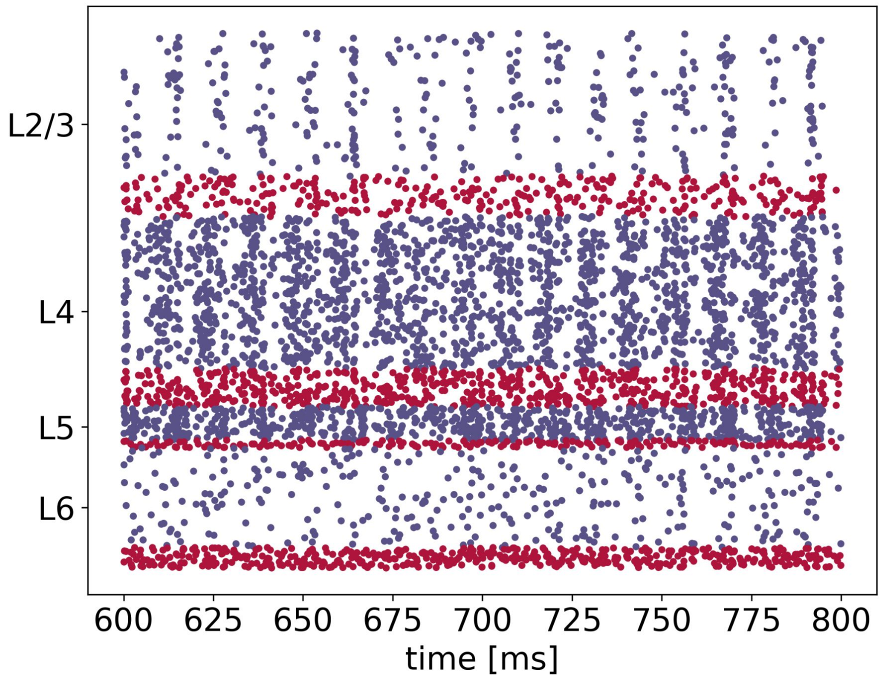
nest::gpu

https://github.com/nest/nest-gpu

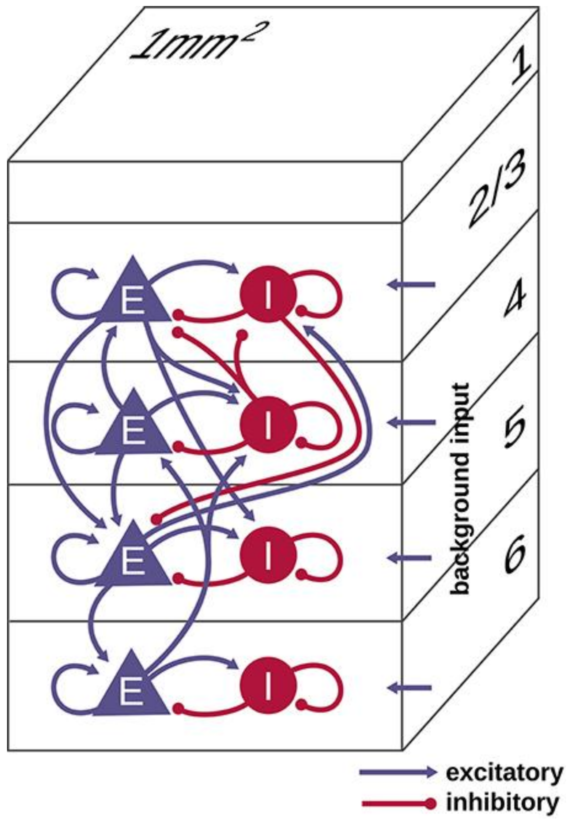
Have a look at the documentation!

Are simulation results compared to NEST CPU the same?

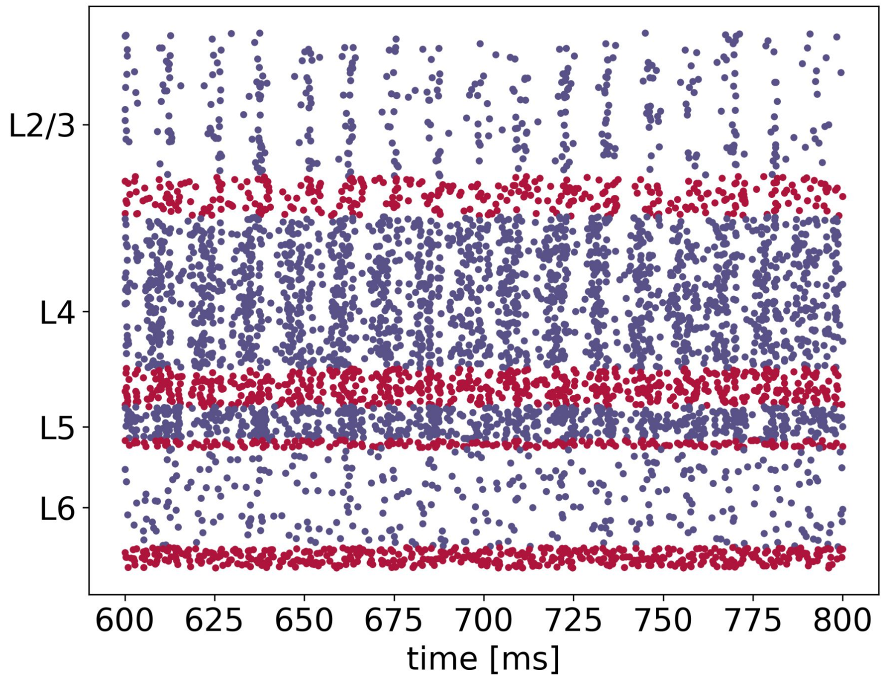
Spiking activity is not identical but statistically equivalent. Golosio et al. (2021)



NEST GPU raster plot of the microcircuit model



Schematic illustration of the Potjans & Diesmann microcircuit model (2014). Adapted from van Albada et al. (2018)

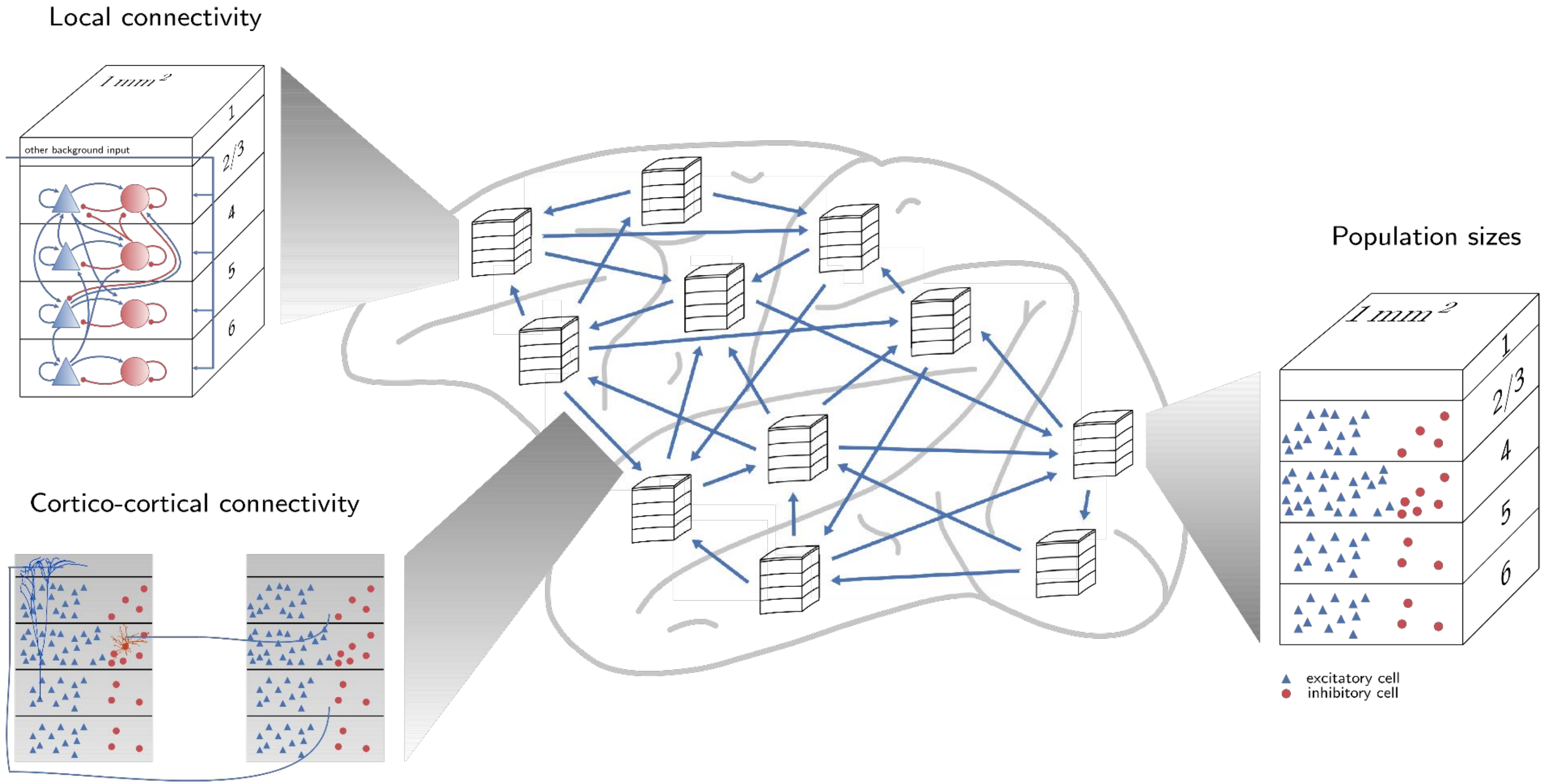


NEST raster plot of the microcircuit model

JÜLICH Forschungszentrum

Support for larger network models

Multi-area model of the macaque visual cortex simulated on a multi-GPU cluster. Tiddia et al. (2022)



Schematic illustration of the multi-area model.
Adapted from Schmidt et al. (2018)

First steps with NEST GPU

Speaker: Luca Sergi

nest::gpu

- What is NEST and what is NEST GPU?
  - Our role in the NEST initiative
- Modeling your first network with NEST GPU
  - An example implementation using the Brunel network
- First steps on spike data analysis
  - Example using first and second order statistics
- Scaling up your network on multiple GPUs
  - A sneak peek on large scale spiking neural network simulations
- Closing remarks
  - Related works and future plans with NEST GPU

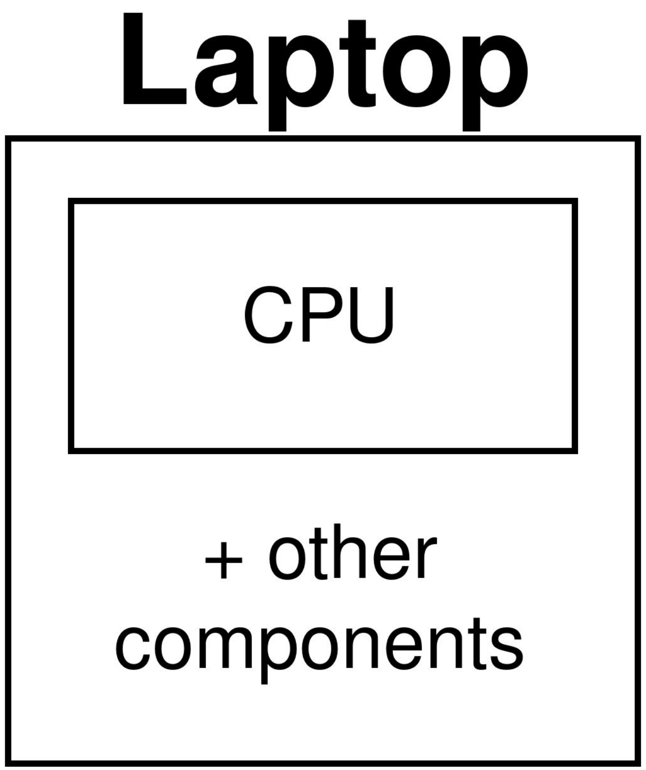
Going large scale with NEST GPU

Speaker: José Villamar

- What is NEST and what is NEST GPU?
  - Our role in the NEST initiative
- Modeling your first network with NEST GPU
  - An example implementation using the Brunel network
- First steps on spike data analysis
  - Example using first and second order statistics
- Scaling up your network on multiple GPUs
  - A sneak peek on large scale spiking neural network simulations
- Closing remarks
  - Related works and future plans with NEST GPU

A deeper look into computing hardware

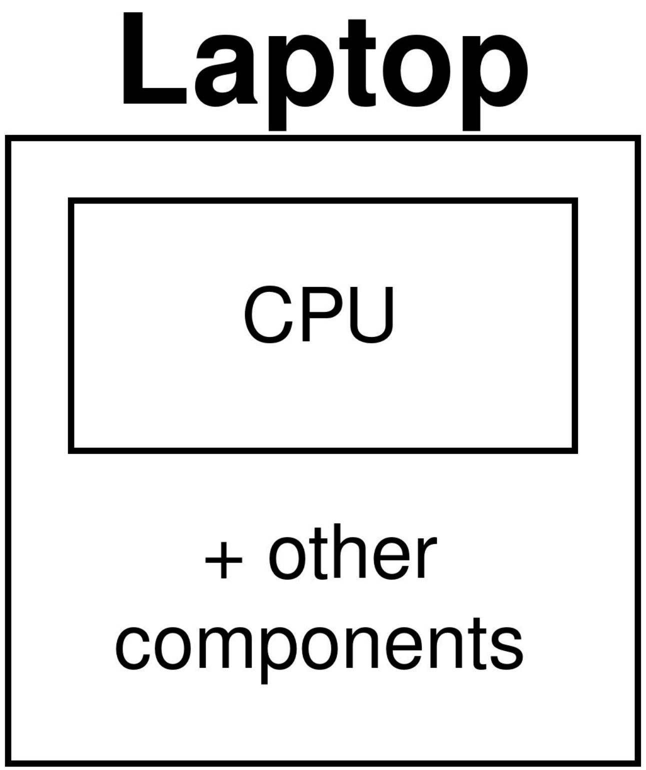
From laptops to clusters

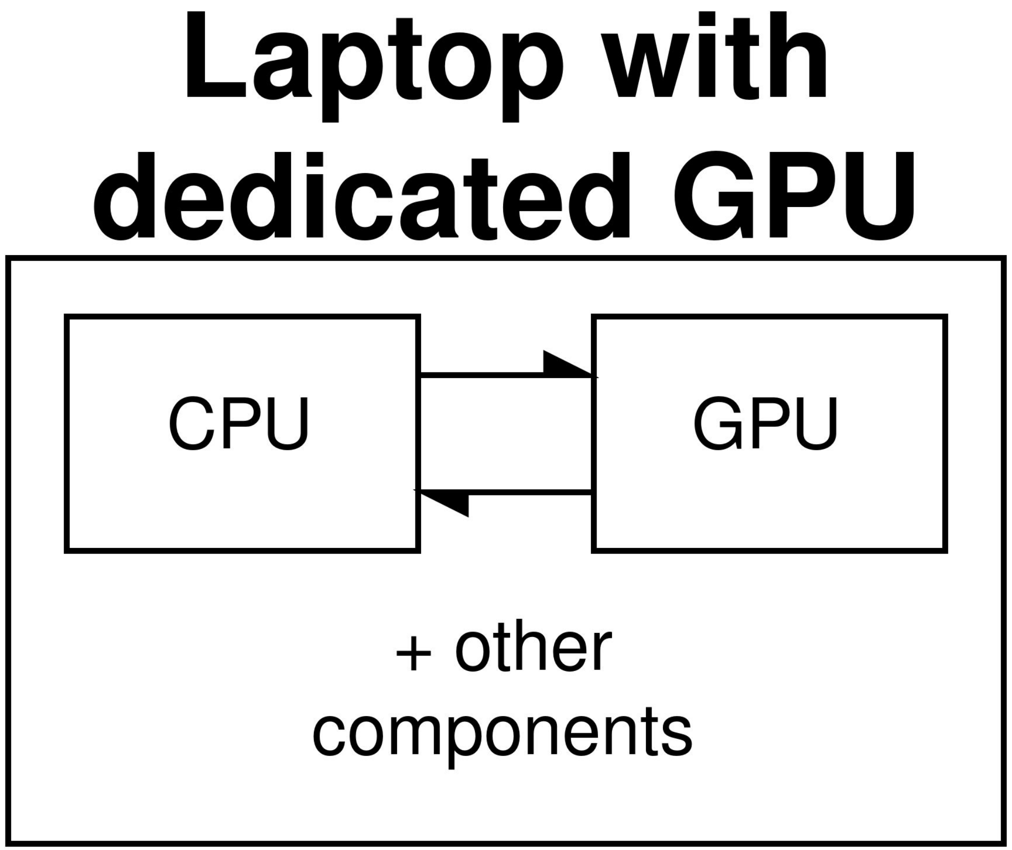


JÜLICH Forschungszentrum

A deeper look into computing hardware

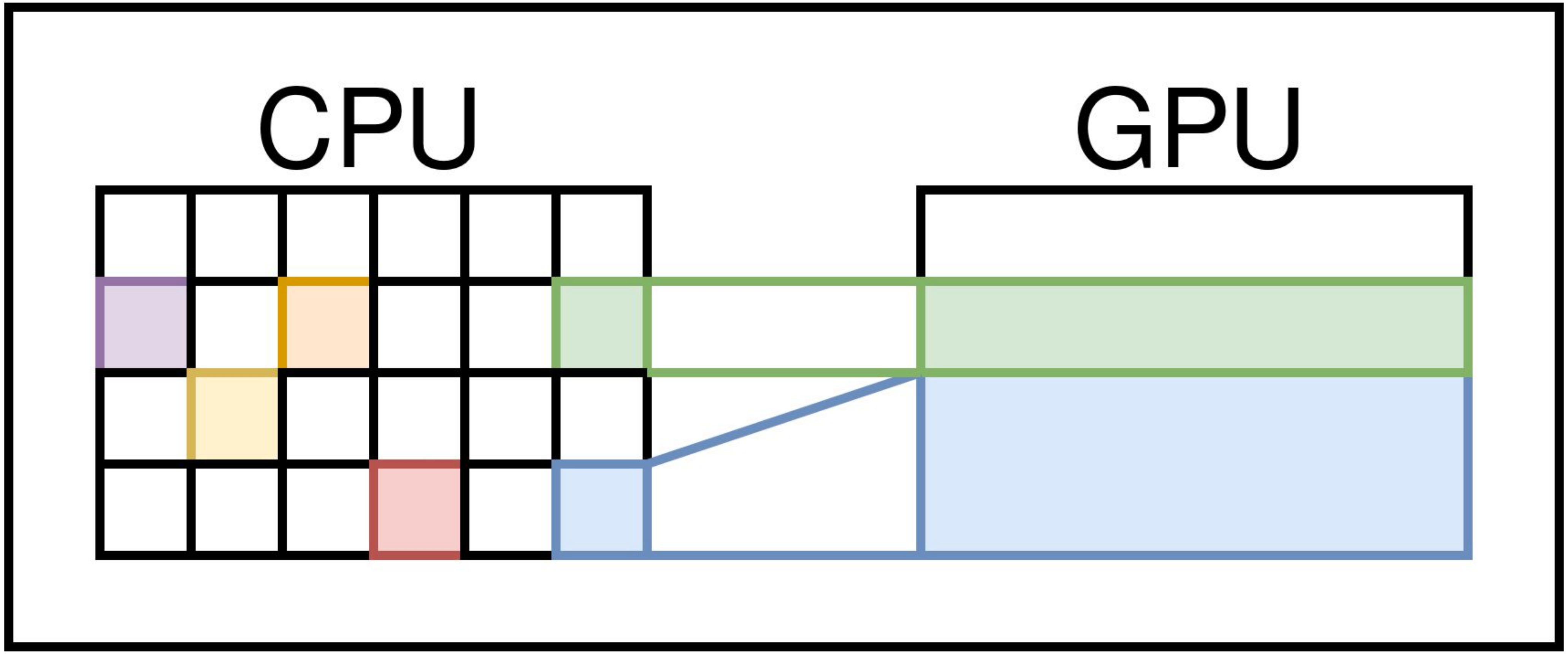
From laptops to clusters





A deeper look into computing hardware

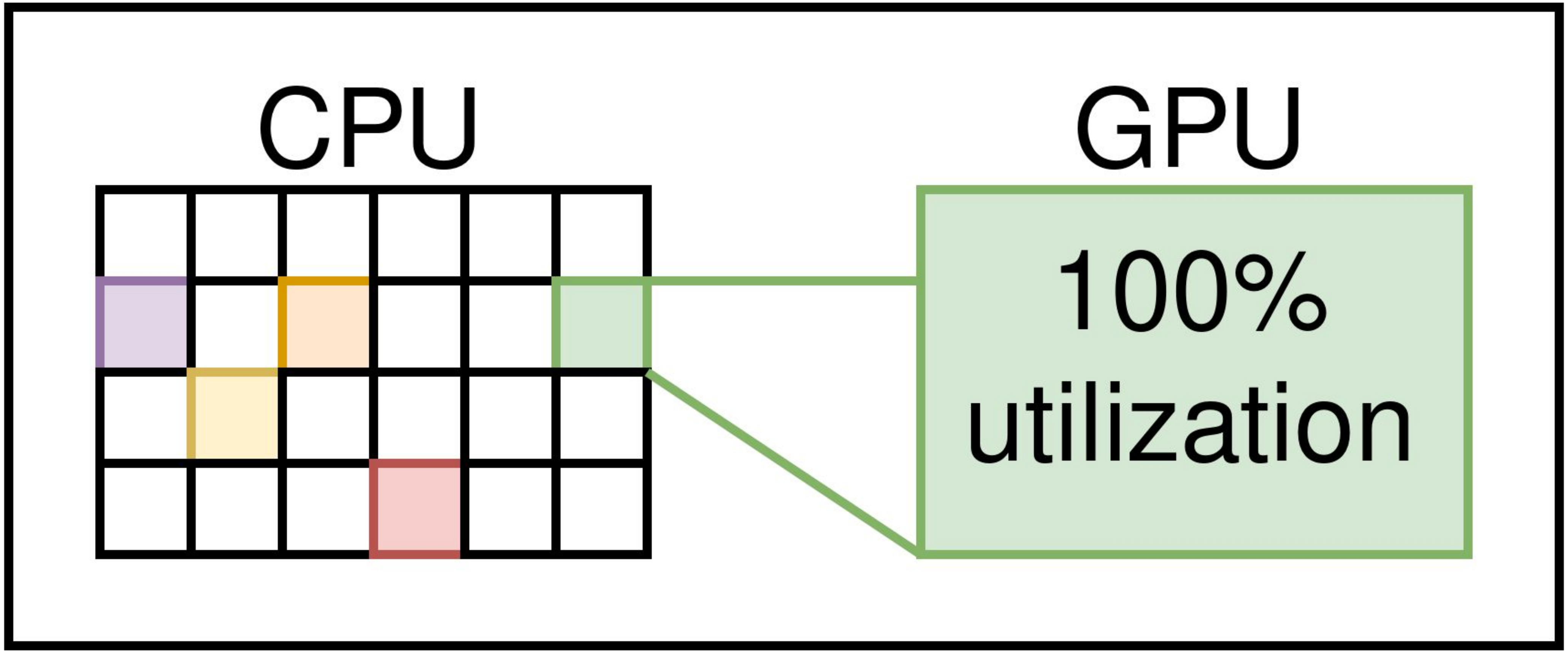
From laptops to clusters



JÜLICH Forschungszentrum

A deeper look into computing hardware

From laptops to clusters

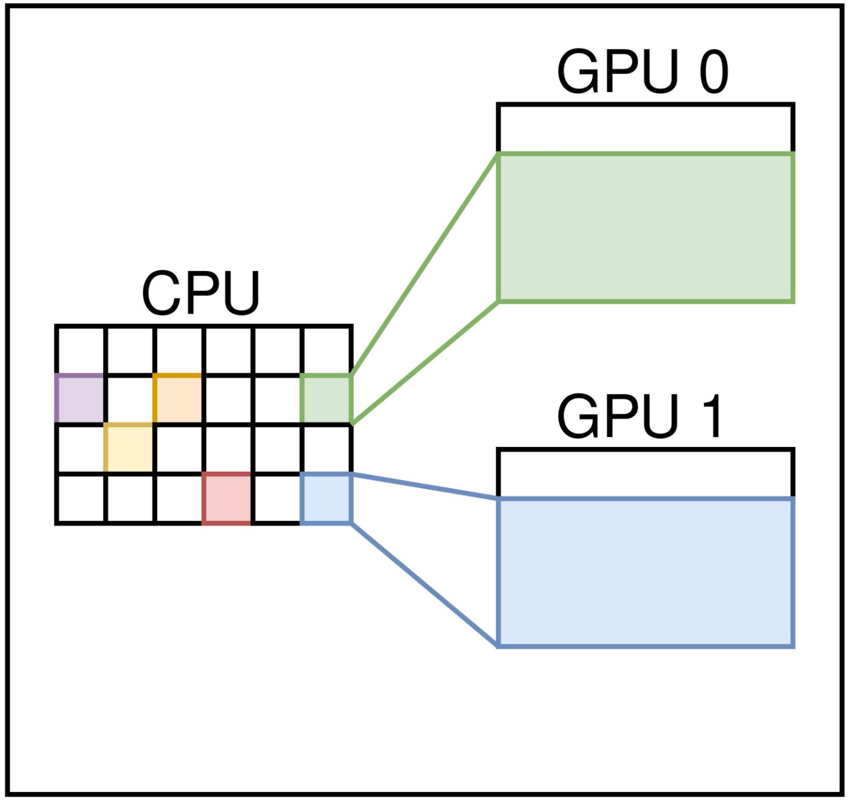


JÜLICH
Forschungszentrum

A deeper look into computing hardware

From laptops to clusters

Workstation with 2 dedicated GPUs

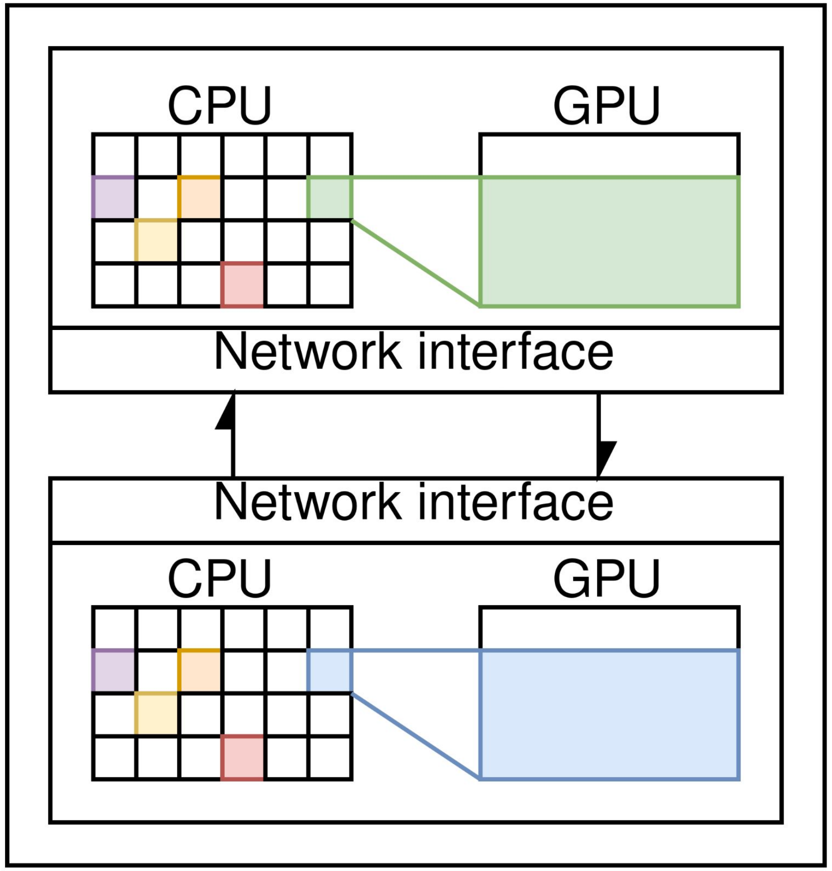


JÜLICH Forschungszentrum

A deeper look into computing hardware

From laptops to clusters

Compute cluster equipped with GPUs

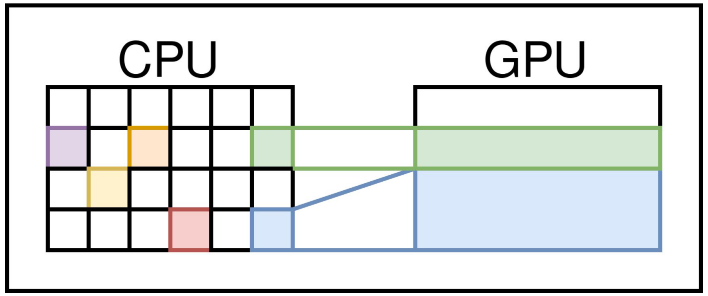


JÜLICH Forschungszentrum

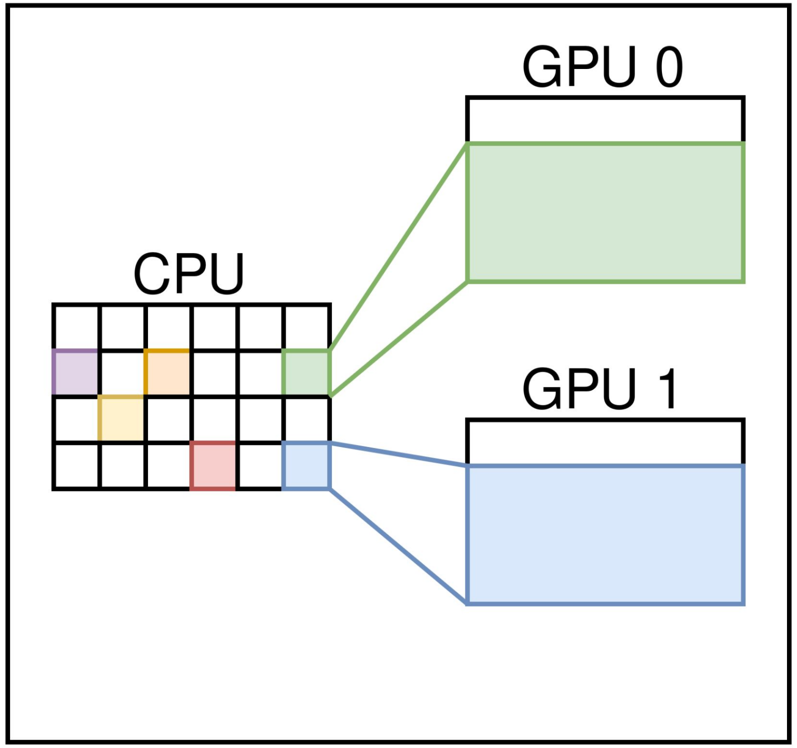
A deeper look into computing hardware

From laptops to clusters

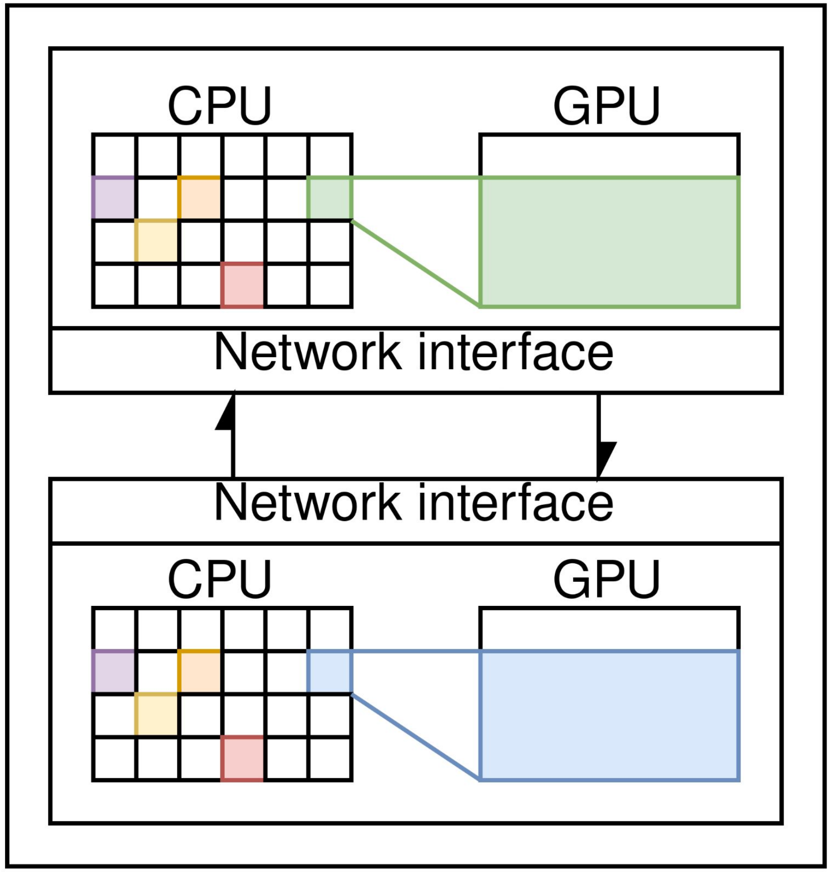
Laptop with dedicated GPU



Workstation with 2 dedicated GPUs



Compute cluster equipped with GPUs



JÜLICH Forschungszentrum

Q: Why does the computing architecture matter?

A: It shouldn't, we take care of it by parallelizing through MPI :)

- The interface of NEST GPU does not globally control all GPUs at once
- The user needs to specify which parts of the network go to each GPU
- Each GPU is controlled by a single MPI process
- The interface is designed so that users define networks in an MPI aware structure

MPI specific interface functions

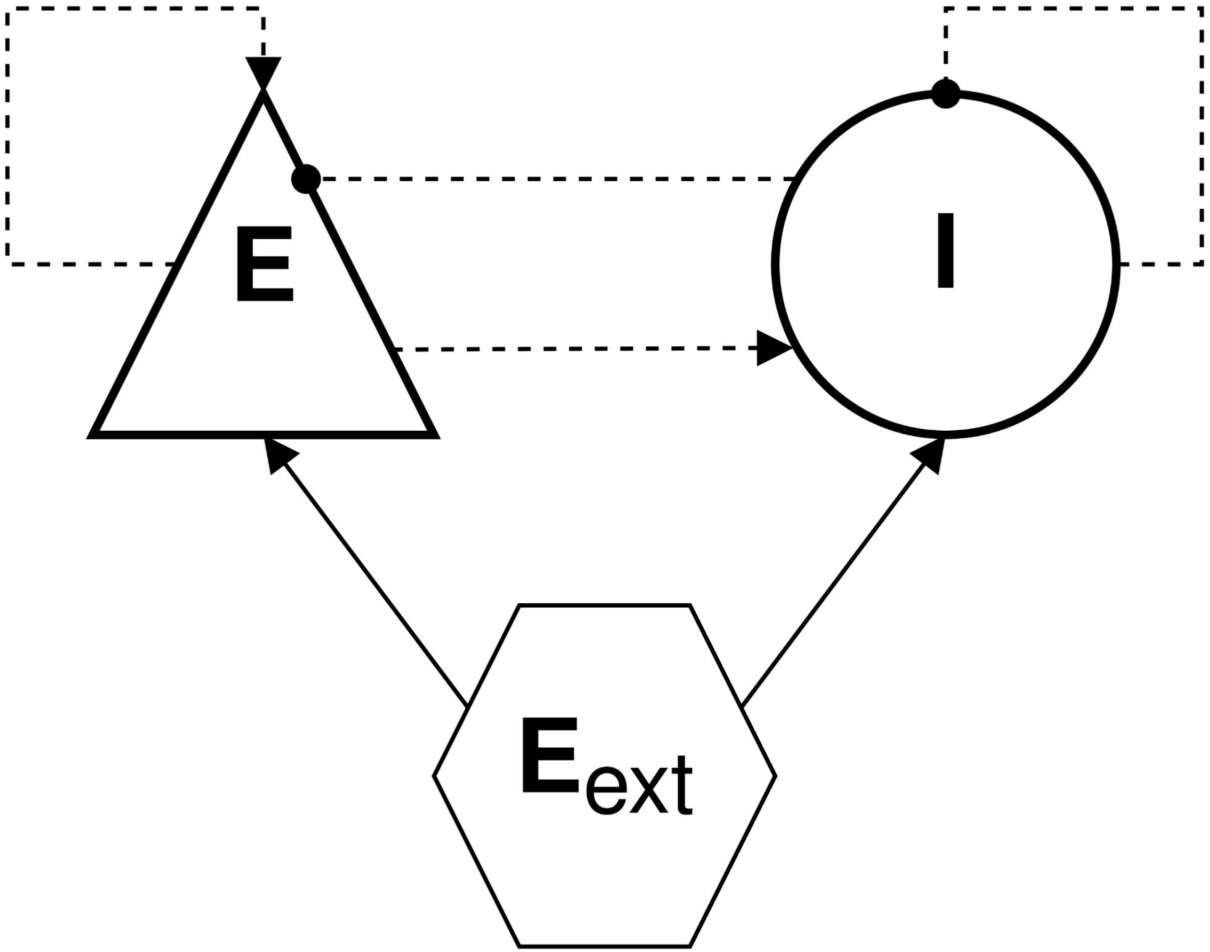
Users instantiate networks either locally without MPI interaction or "remotely" through MPI coordination:

```
nestgpu.Create( neuron model, neuron count, *params )
nestgpu.Connect( source neuron population, target neuron population,
                  *params )
```

```
nestgpu.RemoteCreate( MPI rank, neuron model, neuron count, *params )
nestgpu.RemoteConnect( source MPI rank, source neuron population,
                        target MPI rank, target neuron population, *params )
```

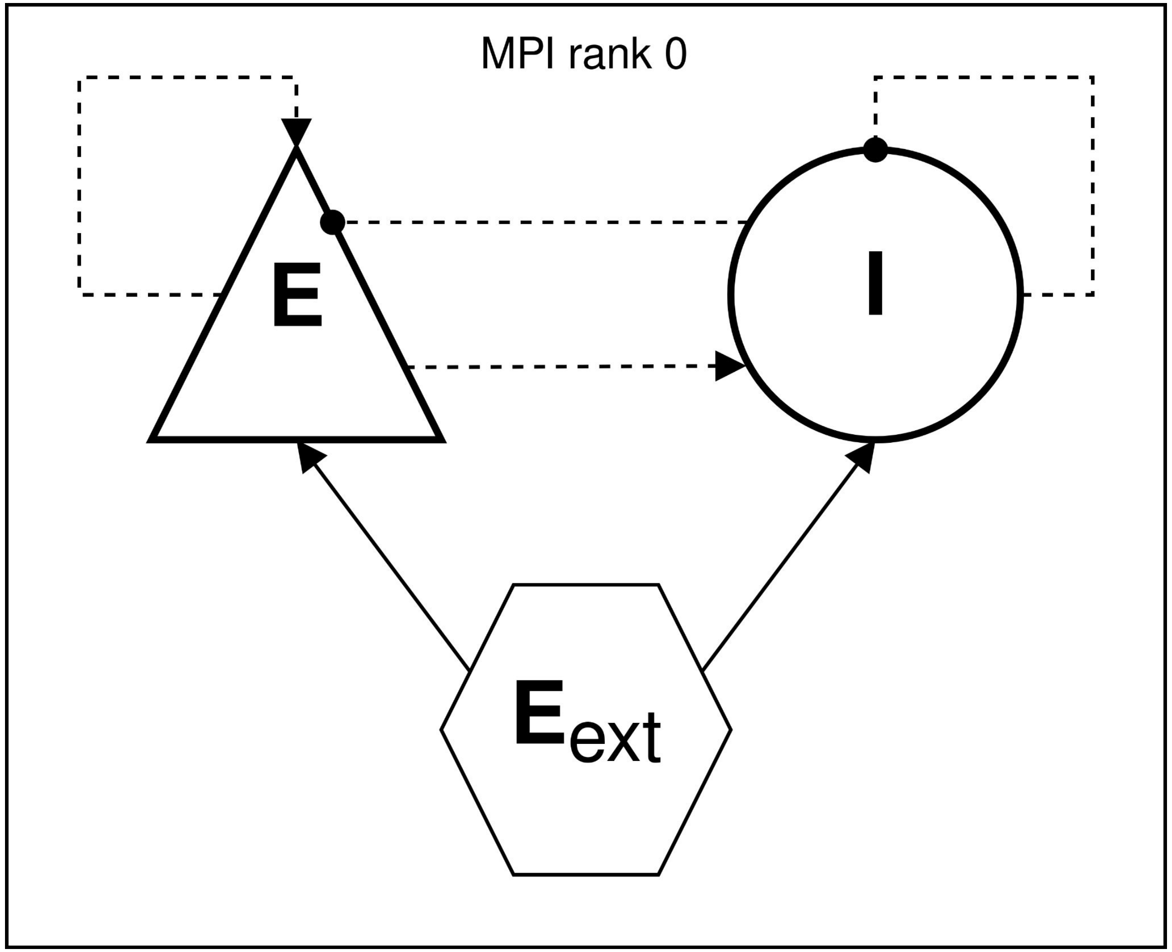
Putting things in practice

Starting from the previously viewed Brunel network



Putting things in practice

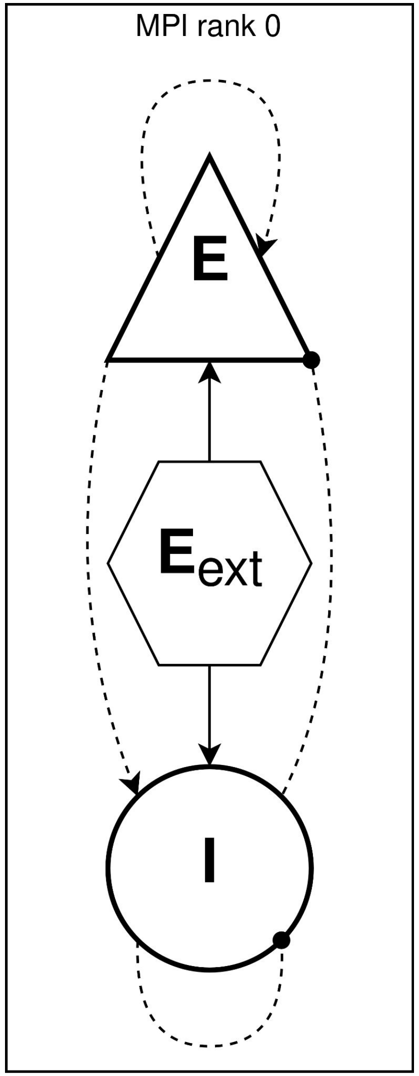
Starting from the previously viewed Brunel network



JÜLICH Forschungszentrum

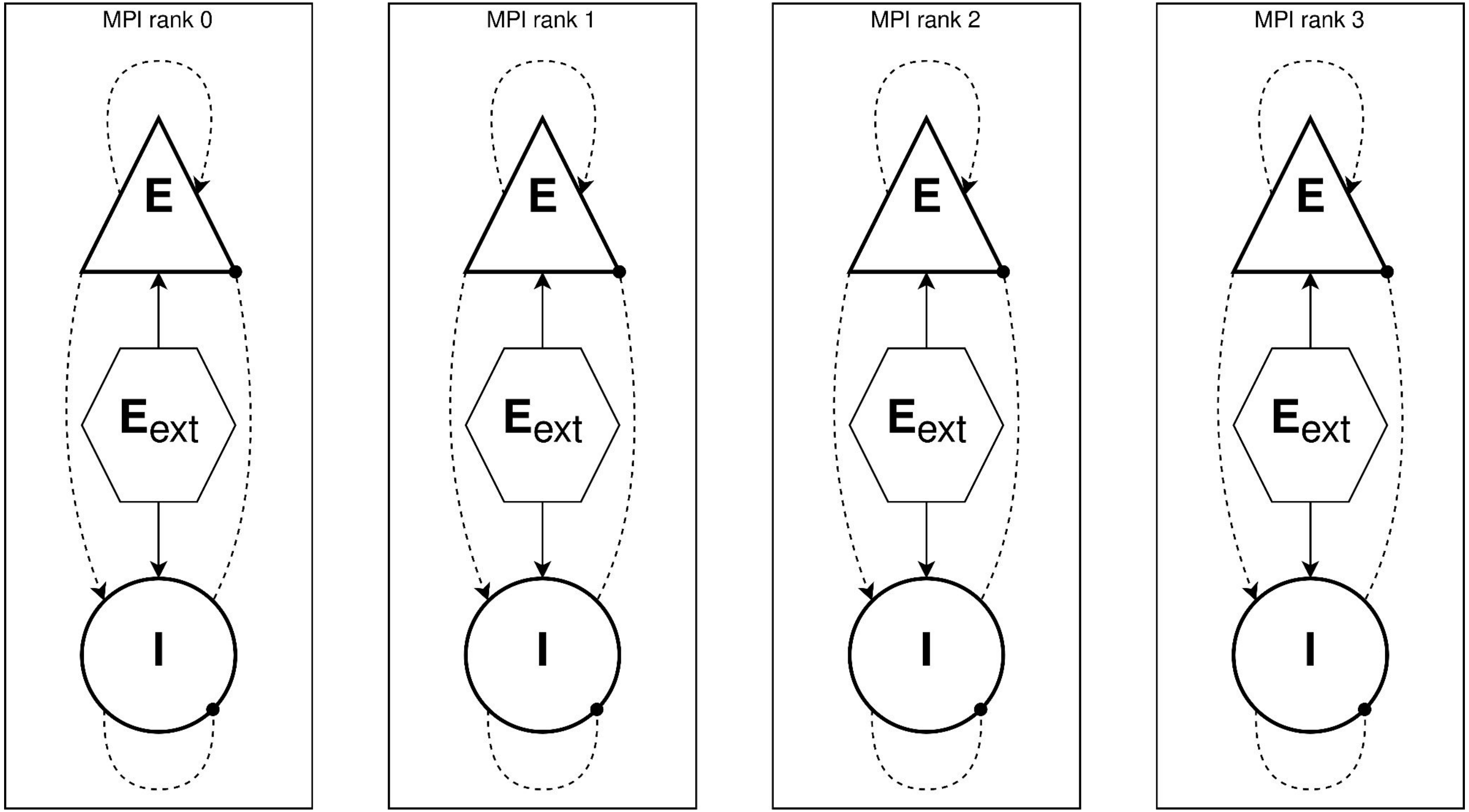
Putting things in practice

Starting from the previously viewed Brunel network



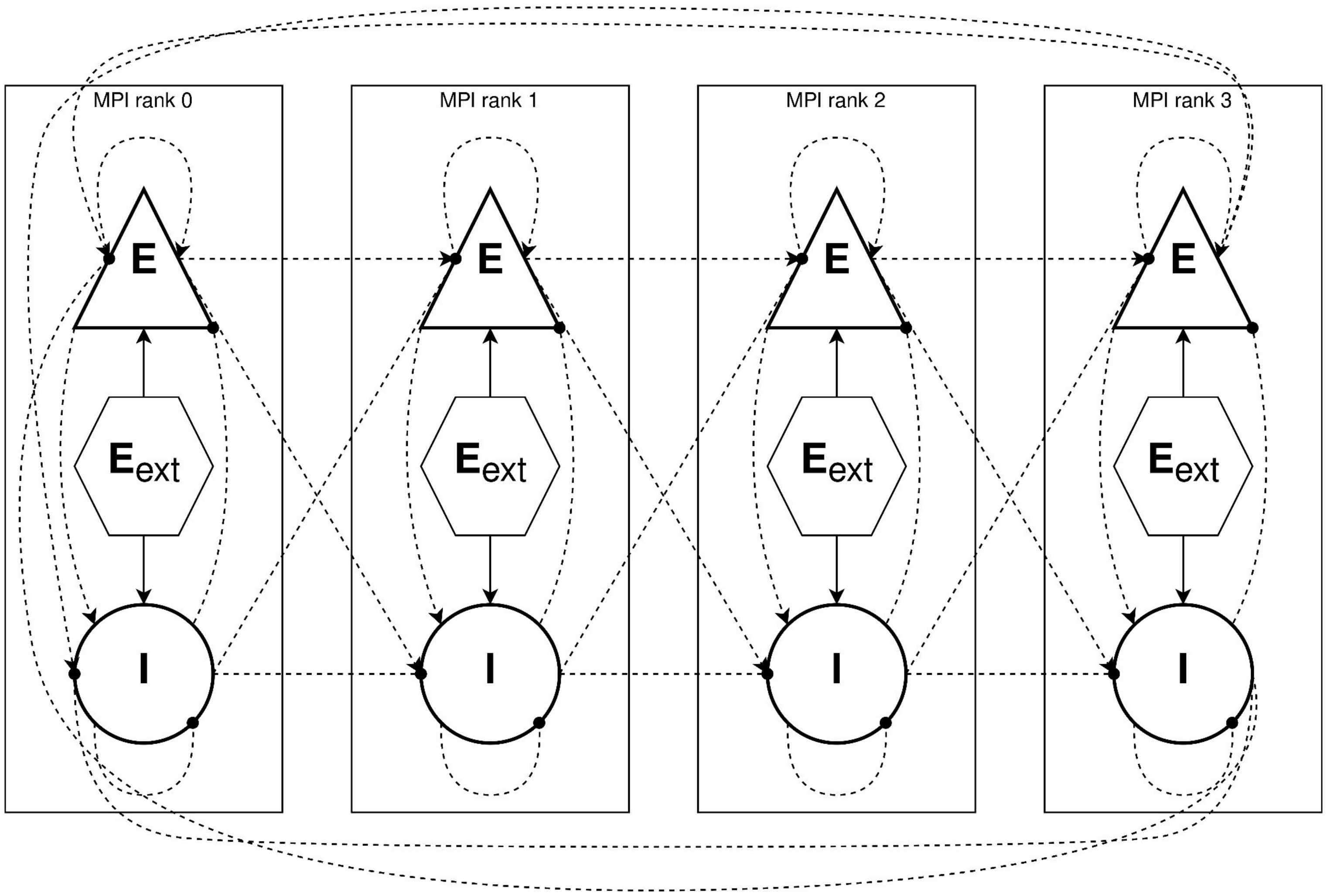
Putting things in practice

Going to multiple Brunel networks across MPI processes



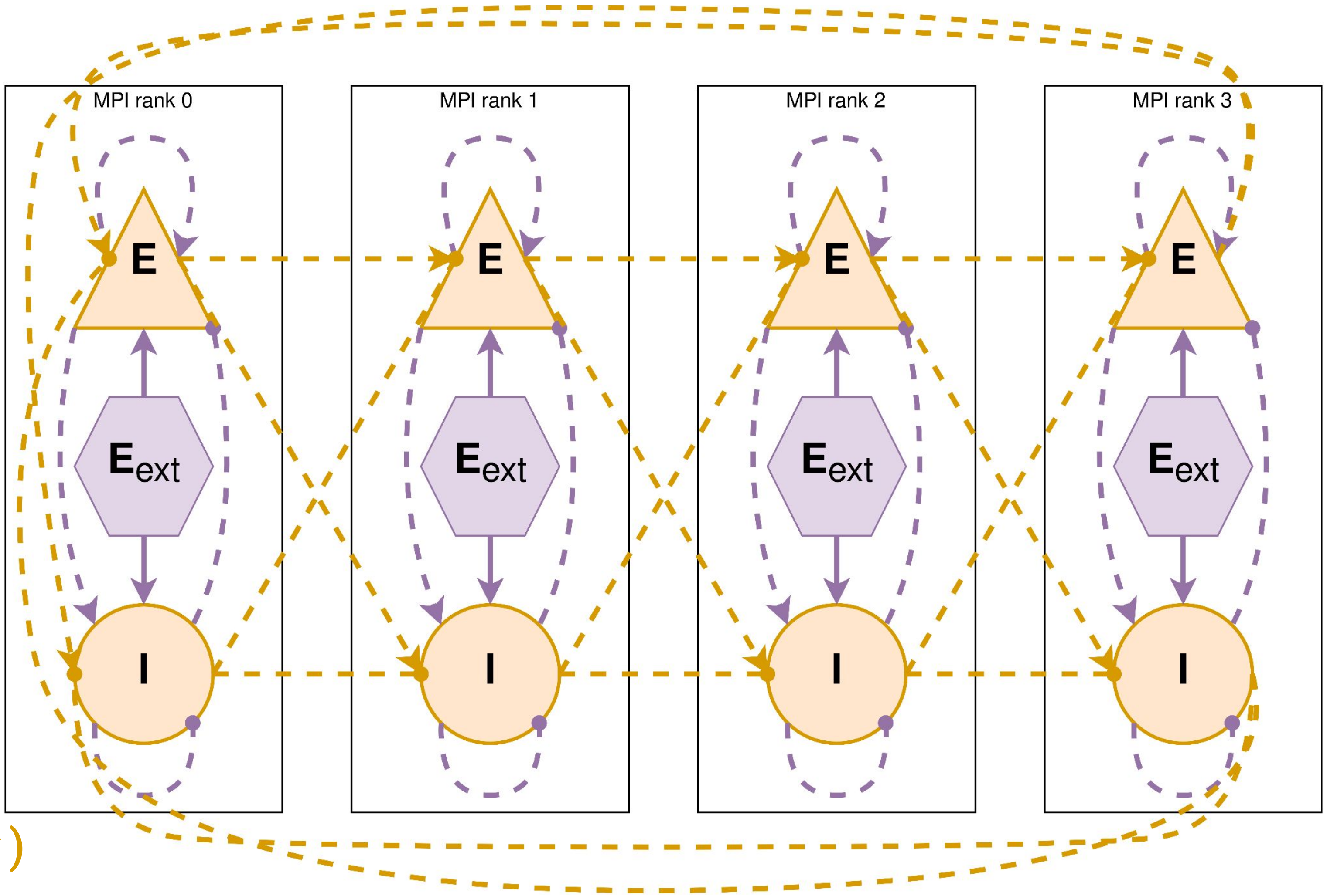
Putting things in practice

Interconnecting each Brunel network in a feed-forward ring topology



Putting things in practice

Interconnecting each Brunel network in a feed-forward ring topology



`Create(*)`

`Connect(*)`

`RemoteCreate(*)`

`RemoteConnect(*)`

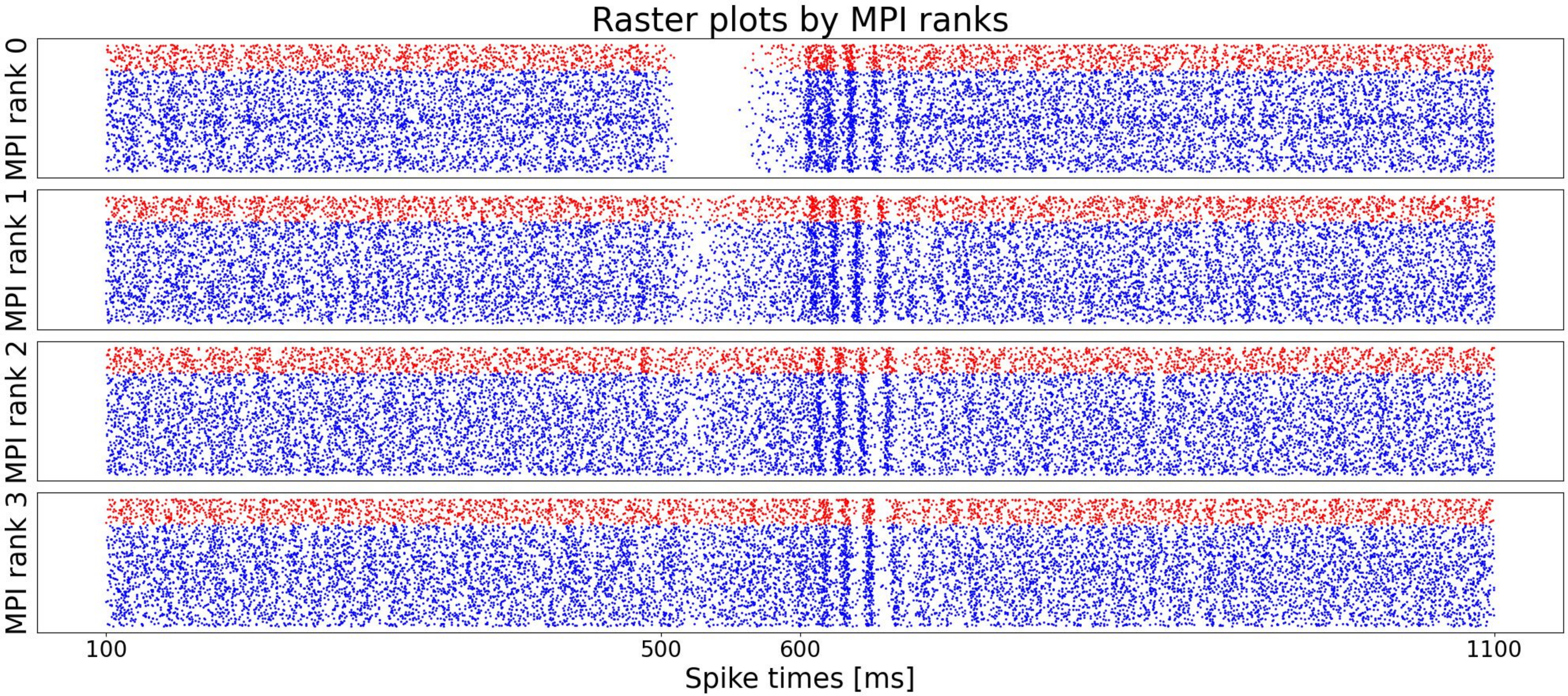
Putting things in practice

Dummy experiment with Brunel ring network

- Each MPI process hosts an instance of the Brunel network
- Each network instance is driven by an individual Poisson generator
- Total number of incoming connections per neurons remain constant regardless of the number of network instances
- Incoming connections are split into remote sources and local sources
    - 60% of excitatory connections come from remote sources
    - 10% of inhibitory connections come from remote sources
- Each network instance receives connections from a neighbor instance and sends connections to another instance
    - By creating connections between "left" and "right" neighbors, all instances form a feed-forward ring topology
- After 500ms the rate of a single Poisson generator is reduced by 25% for 100 ms

Putting things in practice

Getting some interesting(?) results



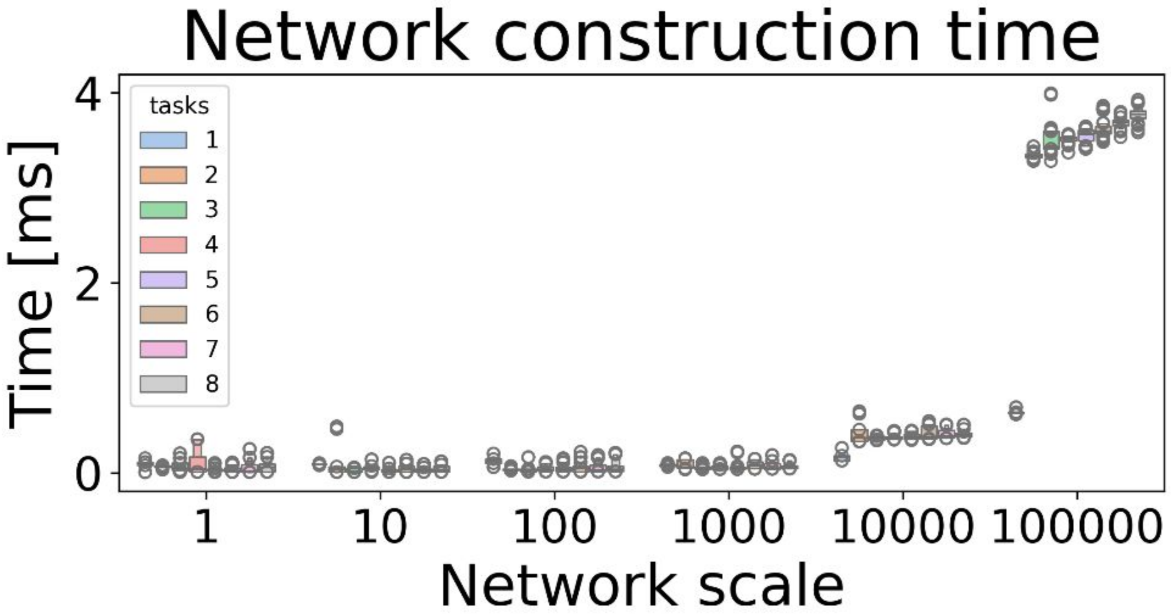
What about performance?

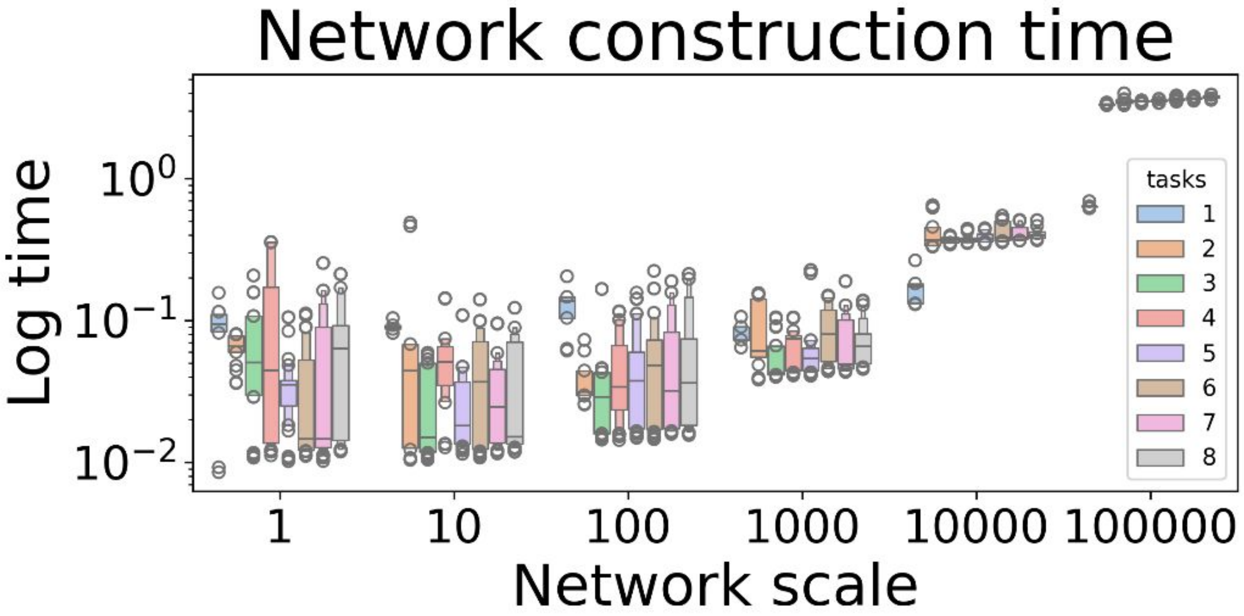
Benchmarking the ring network

- Benchmarks performed at LEONARDO Booster using 1 and 2 nodes
- Each node is equipped with:
  - 1x 32 cores Intel Ice Lake Intel Xeon Platinum 8358
  - 4x NVIDIA Ampere100 custom, 64GiB HBM2e NVLink 3.0 (200 GB/s)
- Simulation run configurations:
  - Using 1 to 8 GPUs
  - Networks instantiated at 1, 10, 100, 1000, 10000, 100000 scale
  - Values averaged across 10 different simulation seeds

What about performance?

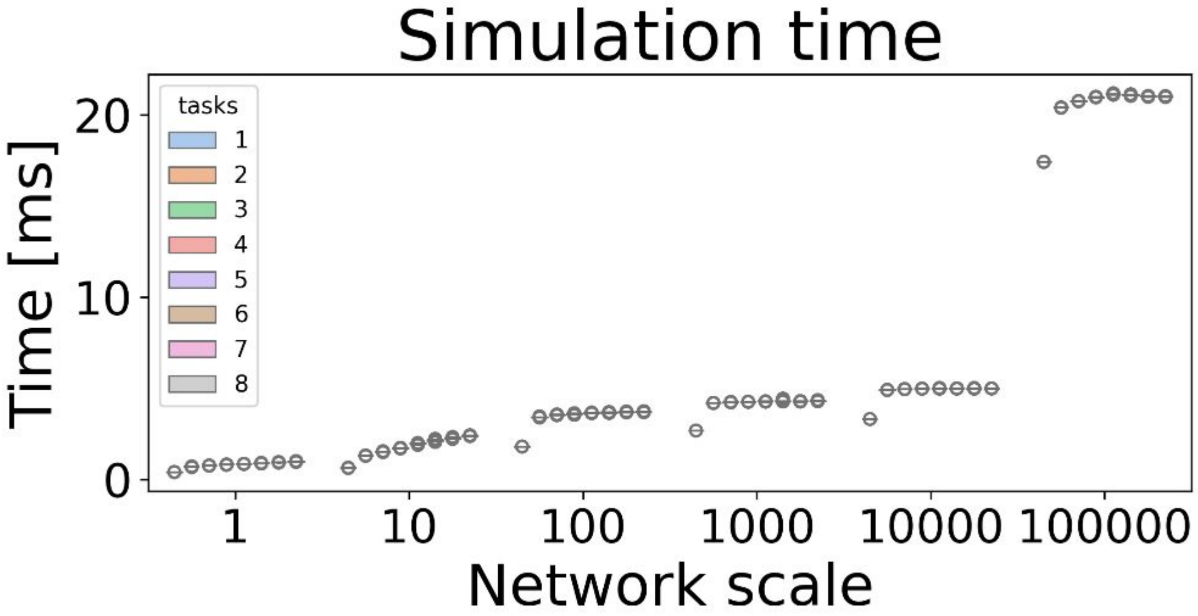
Benchmarking the ring network

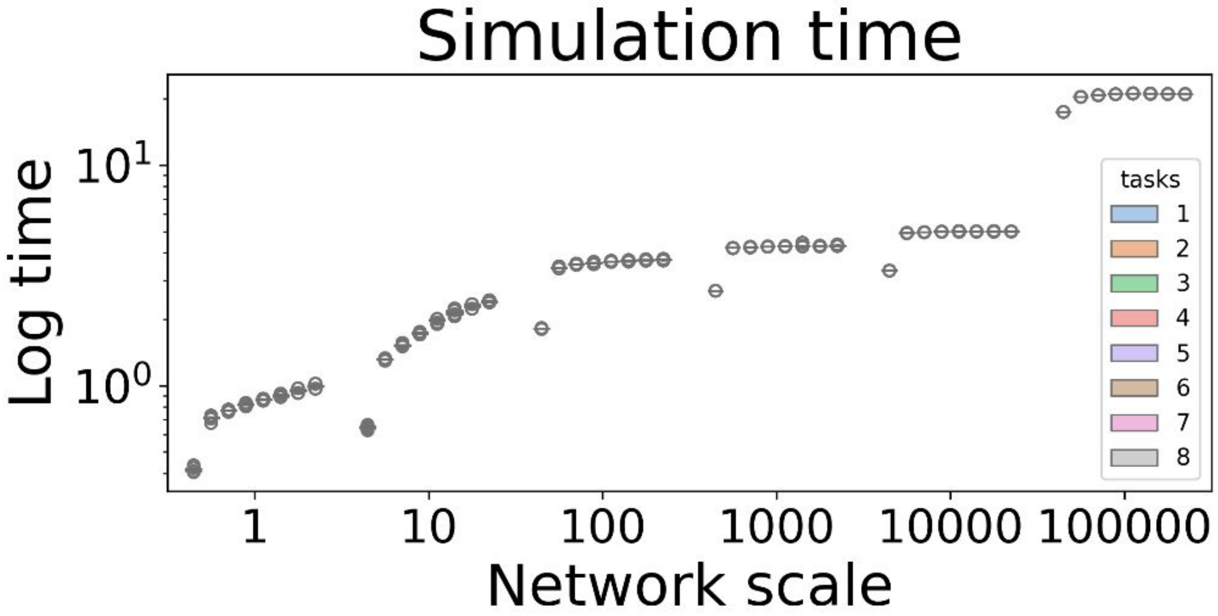




What about performance?

Benchmarking the ring network





JÜLICH
Forschungszentrum

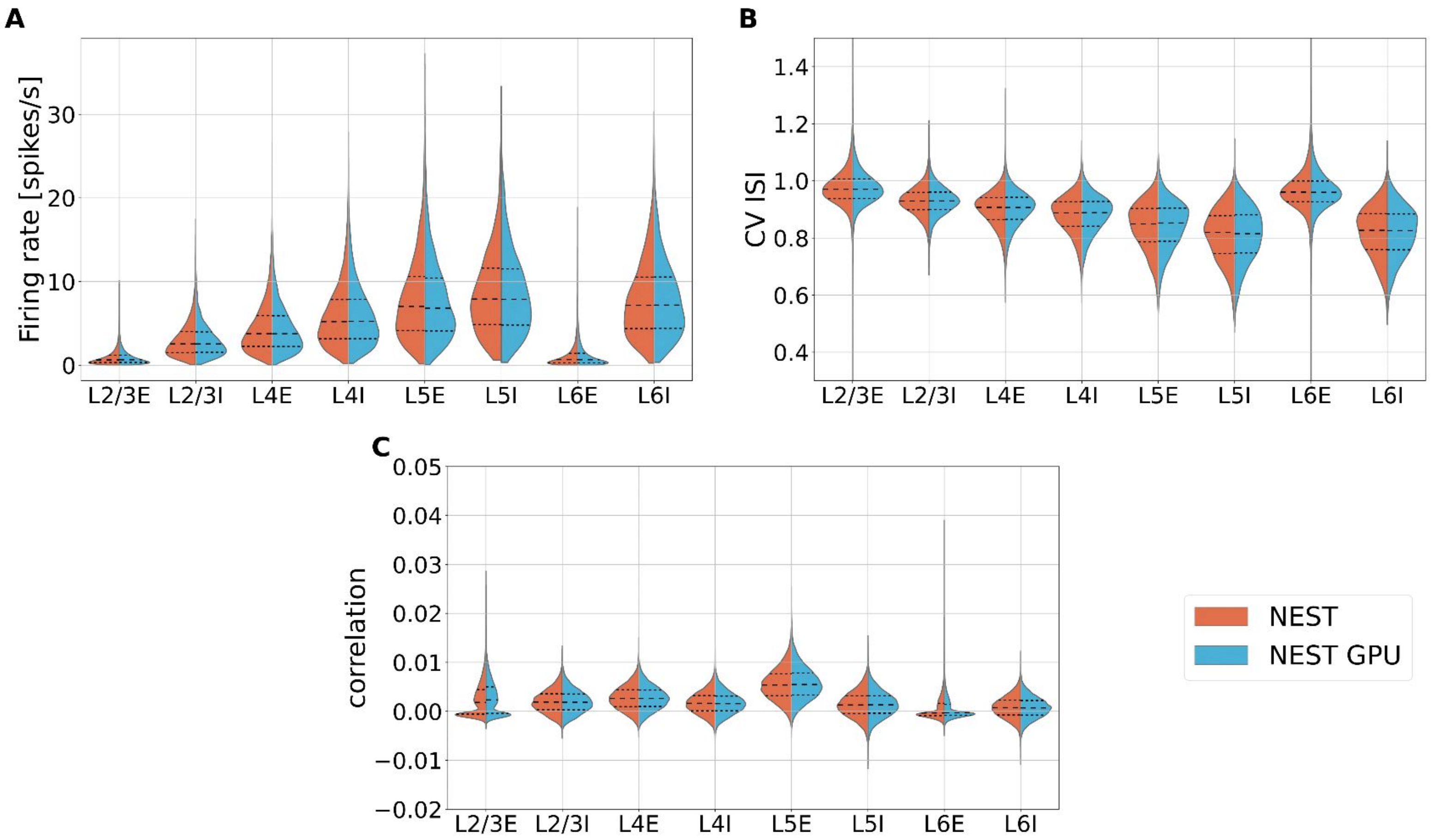
To wrap things up

- What is NEST and what is NEST GPU?
  - Our role in the NEST initiative
- Modeling your first network with NEST GPU
  - An example implementation using the Brunel network
- First steps on spike data analysis
  - Example using first and second order statistics
- Scaling up your network on multiple GPUs
  - A sneak peek on large scale spiking neural network simulations
- Closing remarks
  - Related works and future plans with NEST GPU

Related works

- Simulation of Potjans & Diesmann microcircuit model on a single GPU
  - Golosio et al. (2021) 10.3389/fncom.2021.627620
- Multi-area model of the macaque visual cortex simulated on a multi-GPU cluster
  - Tiddia et al. (2022) 10.3389/fninf.2022.883333
- Improvements on network construction methods
  - Golosio, Villamar, Tiddia et al. (2023) on a single GPU 10.3390/app13179598
  - Golosio, Villamar, Tiddia et al. (in preparation) on multiple GPUs
- Potjans & Diesmann microcircuit model comparison with neuromorphic systems
  - Senk et al. (2025) 10.48550/arXiv.2505.21185

Related works: Spiking data comparison using PD14

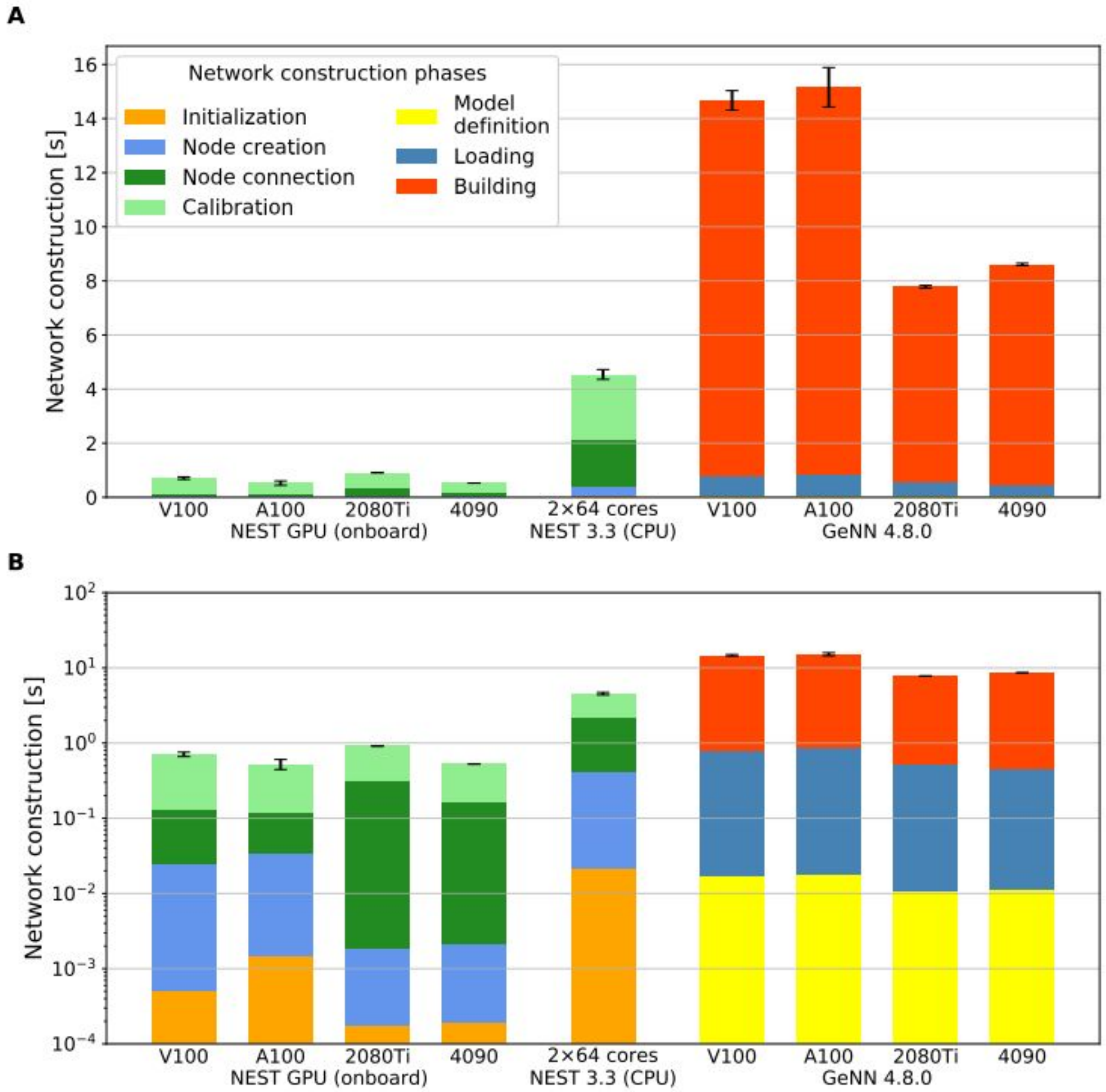


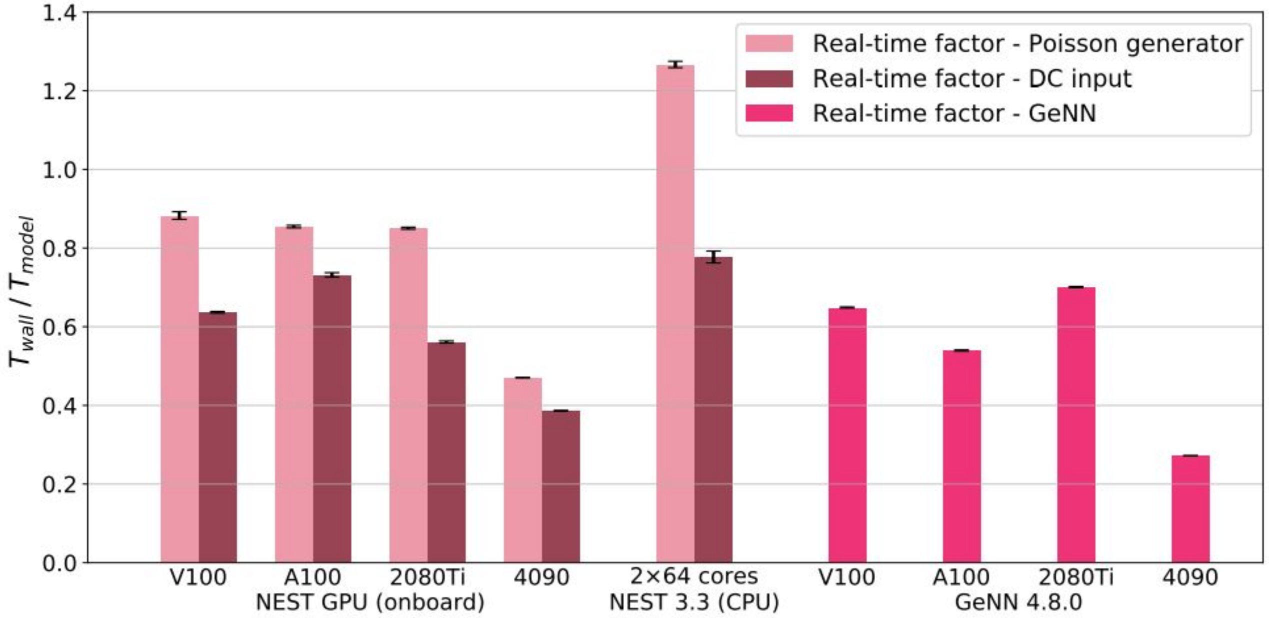
Violin plots of the distributions of **firing rate** (**A**), **CV ISI** (**B**), and **Pearson correlation** (**C**) of simulations of the populations of the microcircuit model using NEST GPU and NEST 3.3.

Spiking activity was recorded for 600s of simulation time after 500ms of pre-simulation time using a resolution of 0.1ms.

Golosio, Villamar, Tiddia et al. (2023)

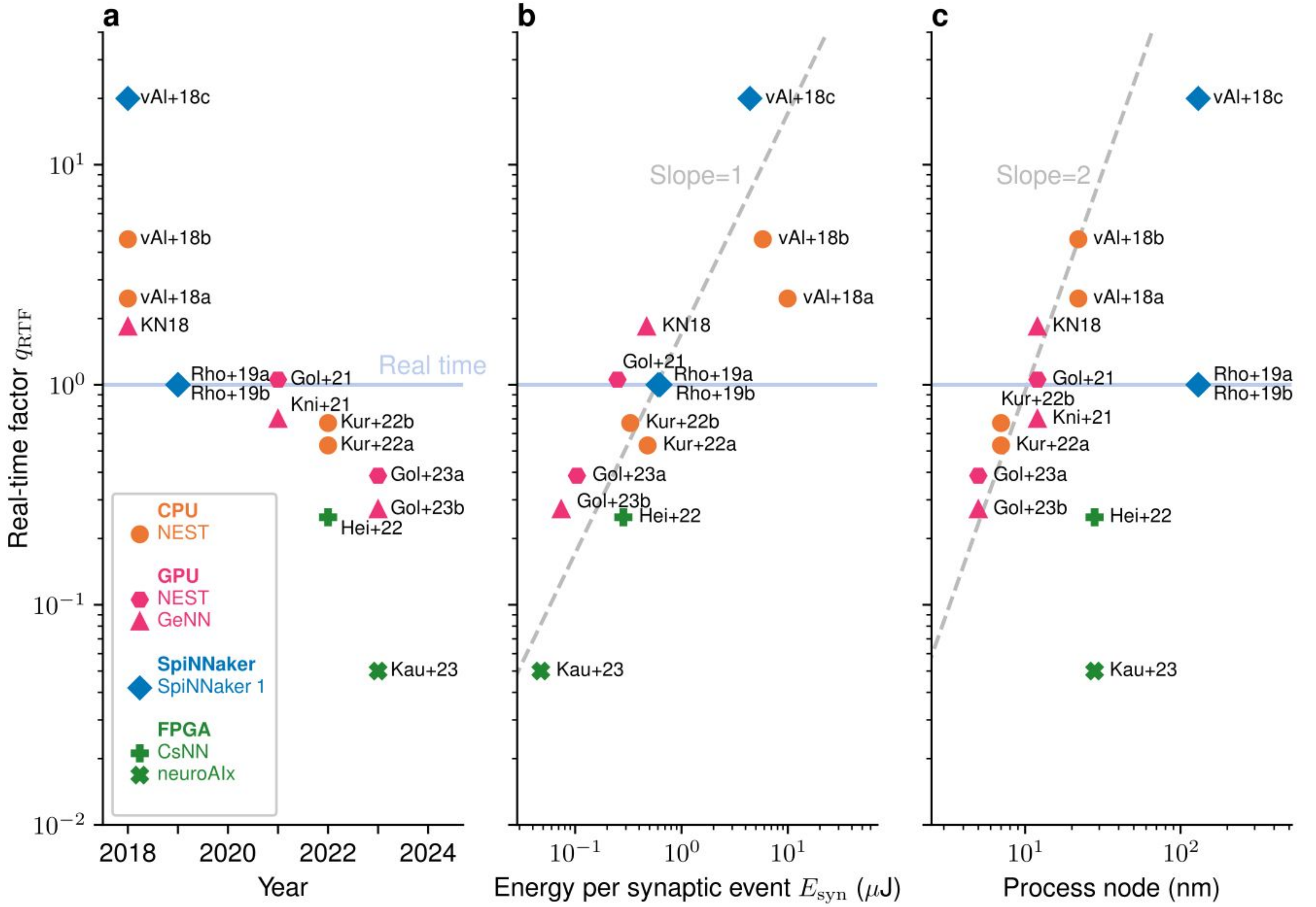
Related works: Performance comparison using PD14





- Golosio, Villamar, Tiddia et al. (2023)
- Network construction in ~0.5 s
- Real-time factor:
  - Poisson generator: ~0.5
  - DC input: ~0.4

Related works: Comparison with neuromorphic systems



The PD14 model has been implemented and simulated across a variety of software and hardware.

Senk et al. (2025) show a comparison of these implementations through simulation speed, energy consumption, and process node size.

JÜLICH Forschungszentrum

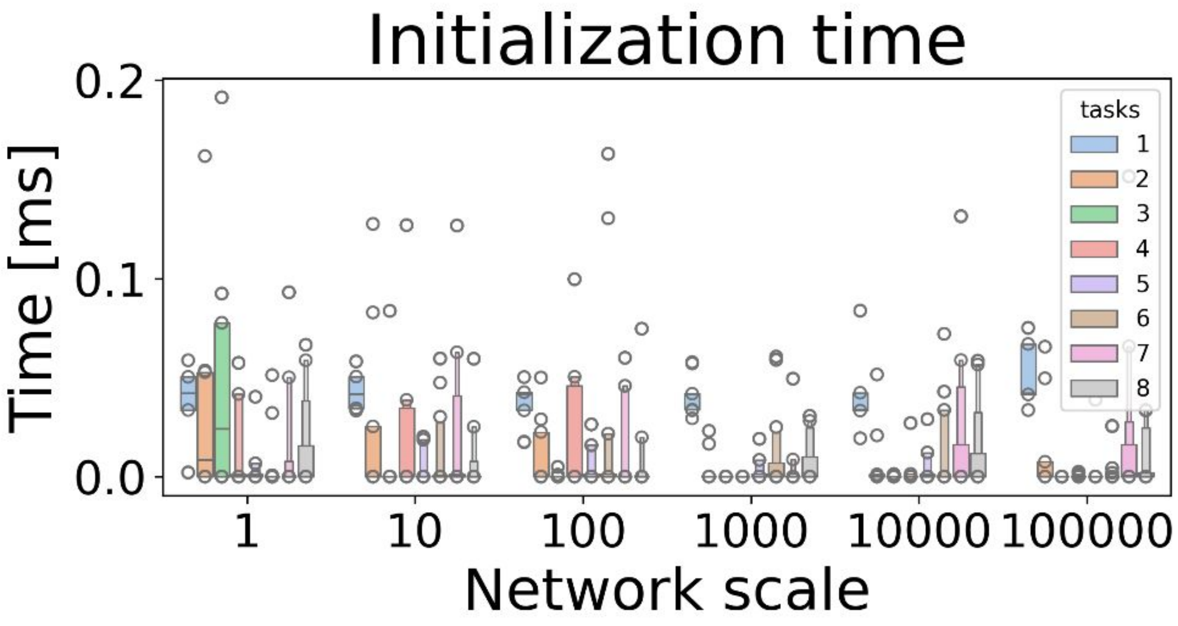
Future plans with NEST GPU

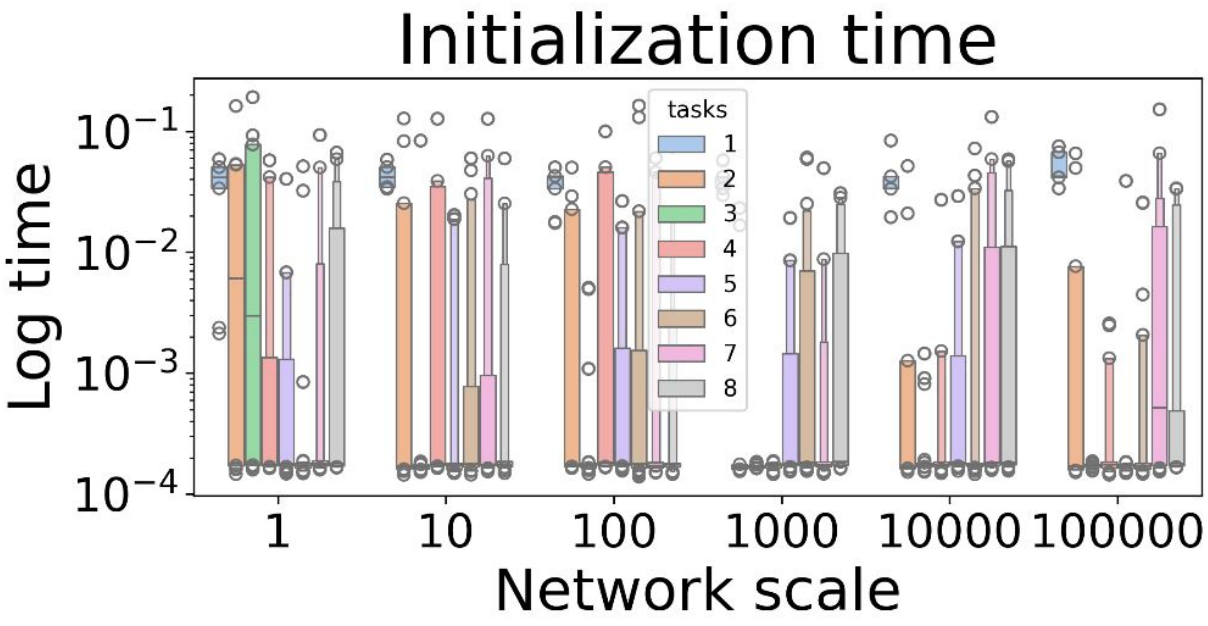
- Kernel optimizations:
  - Optimize single GPU performance
  - Use hardware topology to define machine aware MPI neighborhoods
- Code base alignment:
  - Compatibility with NESTML to expand upon available neuron and synapse models
  - Unify Python interface for easier user interaction between CPU and GPU backends
  - Bring more of the available features of the CPU simulator
    - Next planned feature is spatially defined networks
  - Merge simulation kernel backends to achieve a single code base

Q&A

What about performance?

Benchmarking the ring network

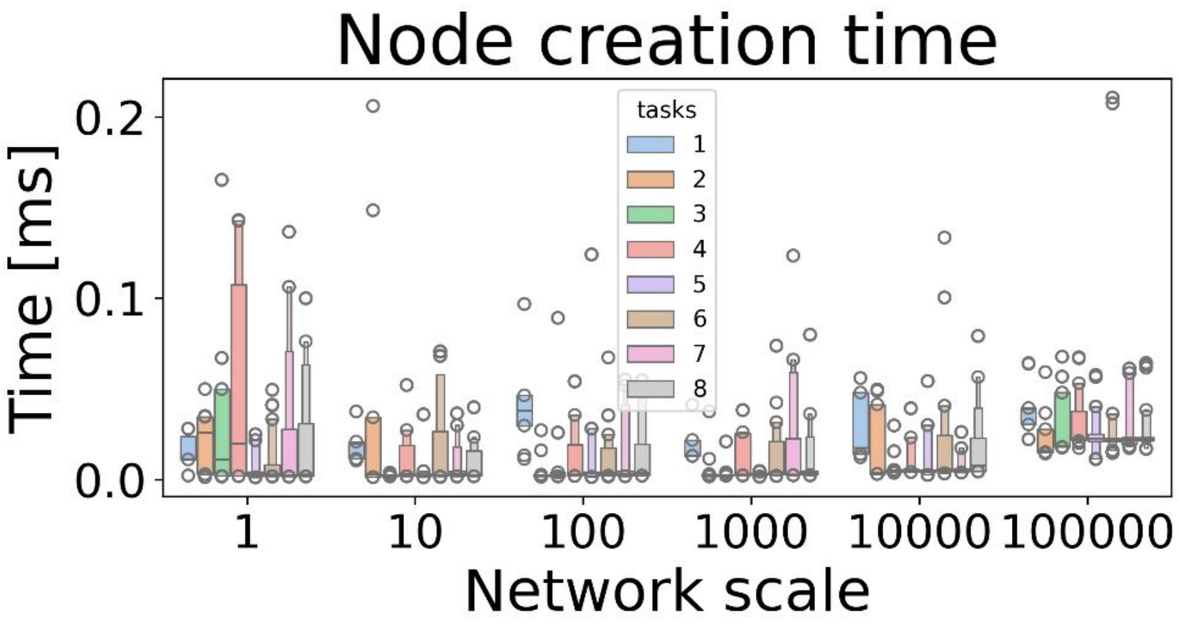


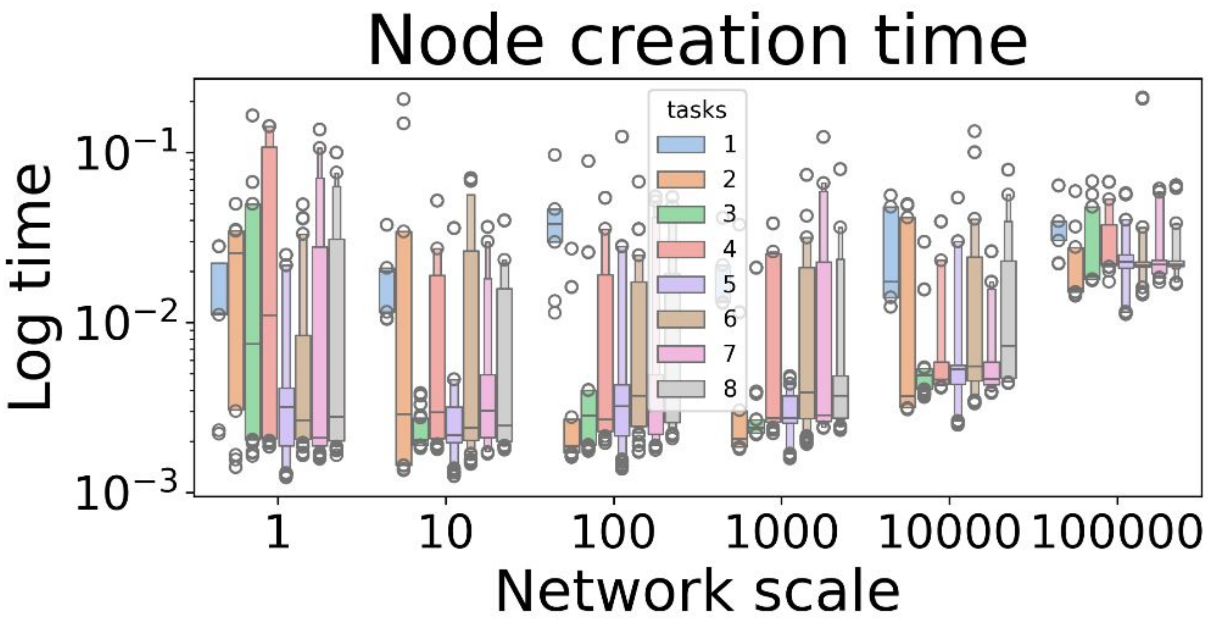


JÜLICH
Forschungszentrum

What about performance?

Benchmarking the ring network

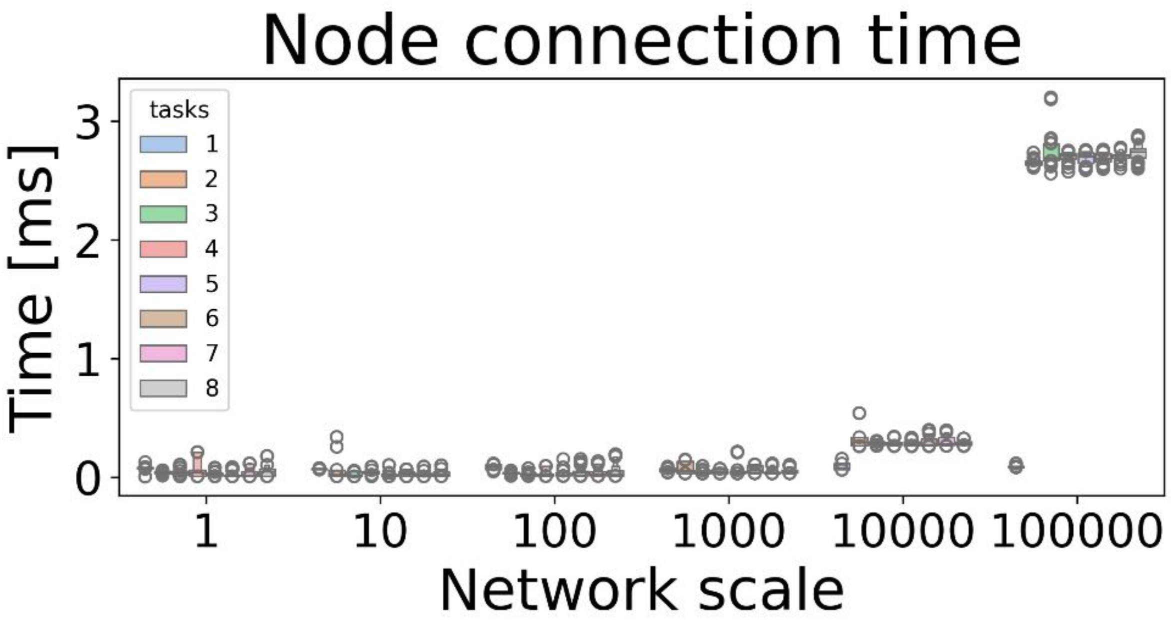


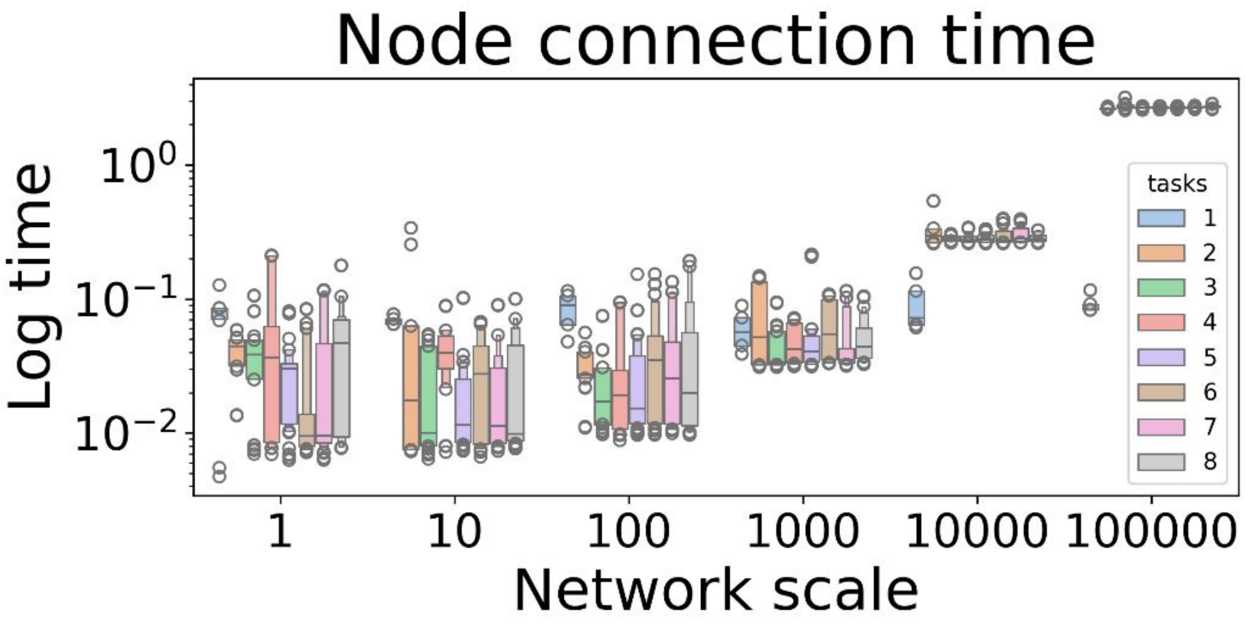


JÜLICH Forschungszentrum

What about performance?

Benchmarking the ring network

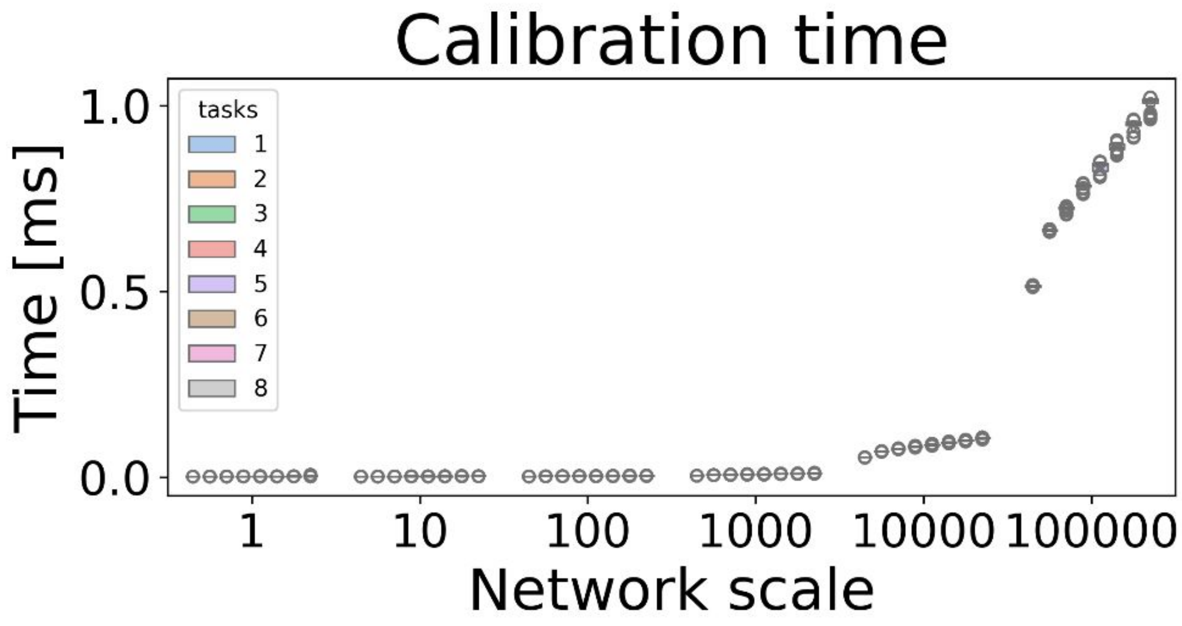


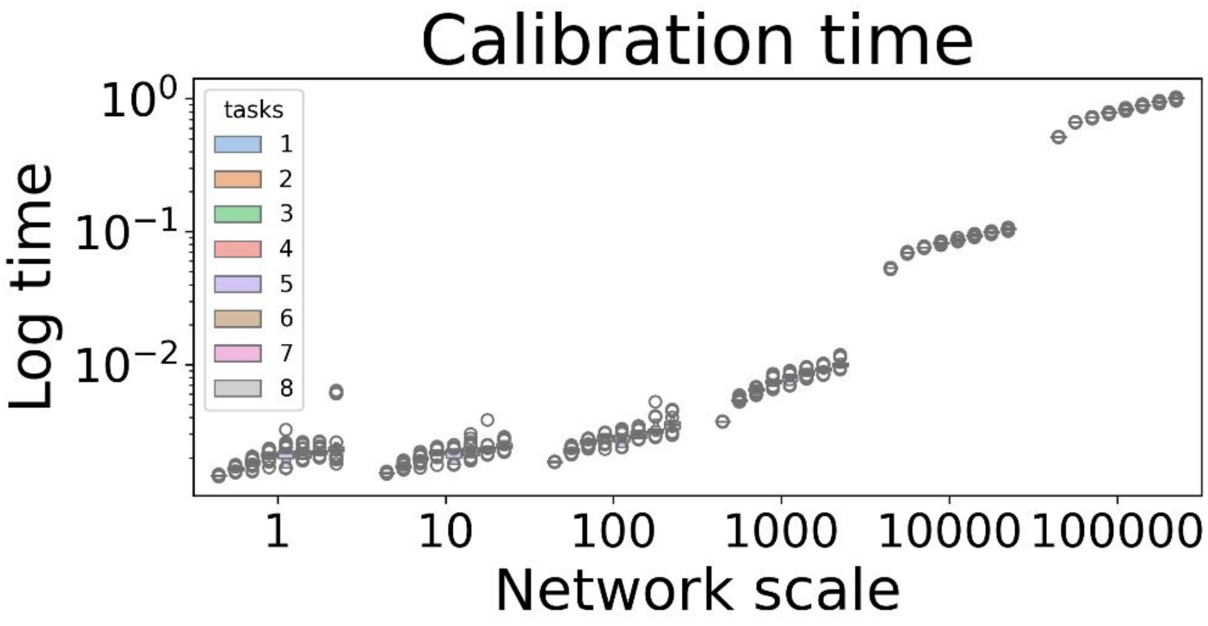


JÜLICH
Forschungszentrum

What about performance?

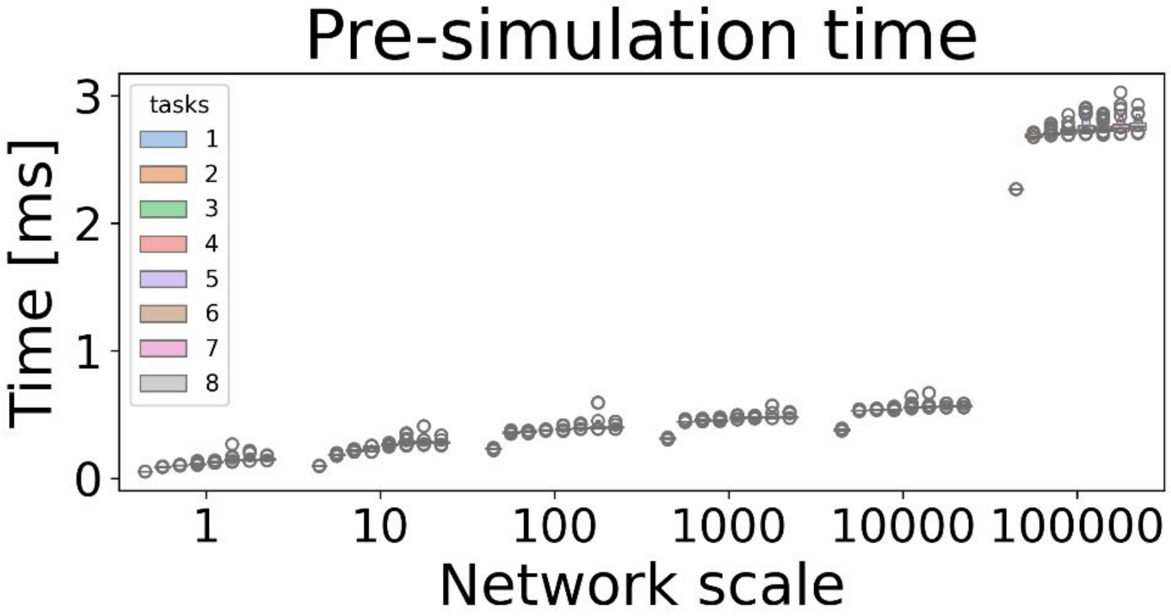
Benchmarking the ring network

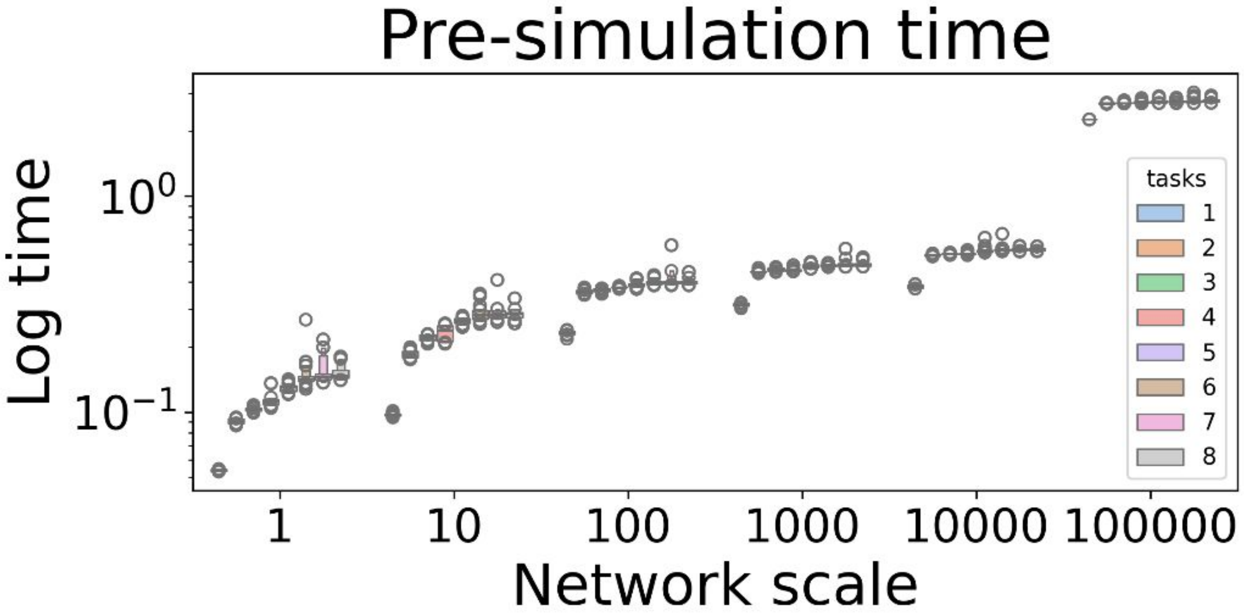




What about performance?

Benchmarking the ring network

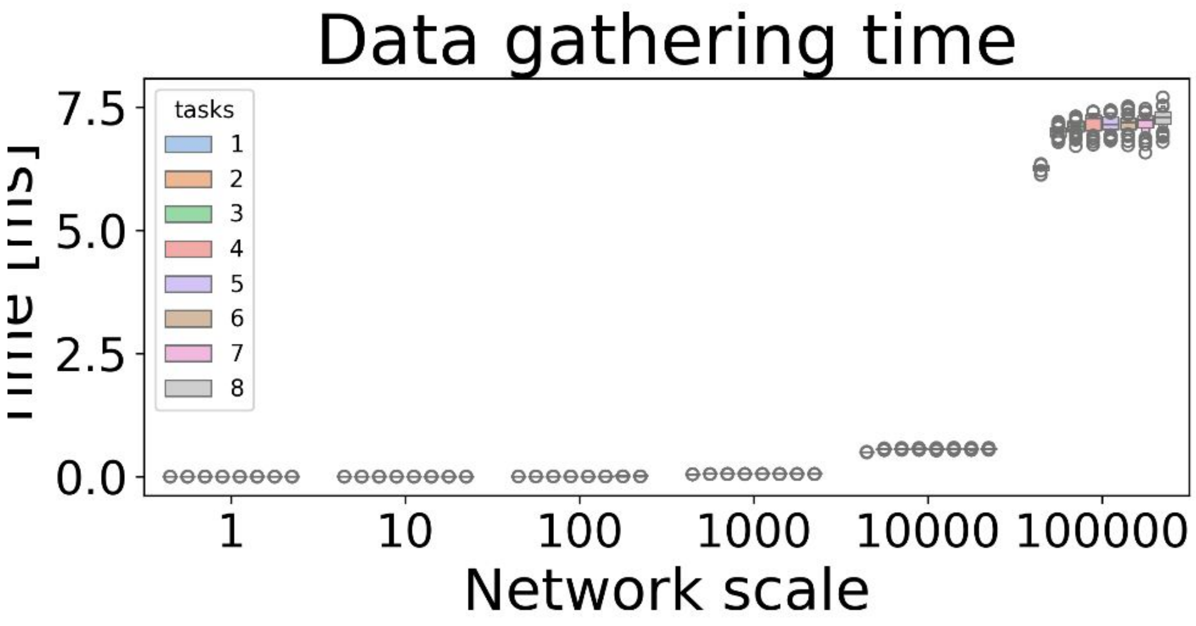


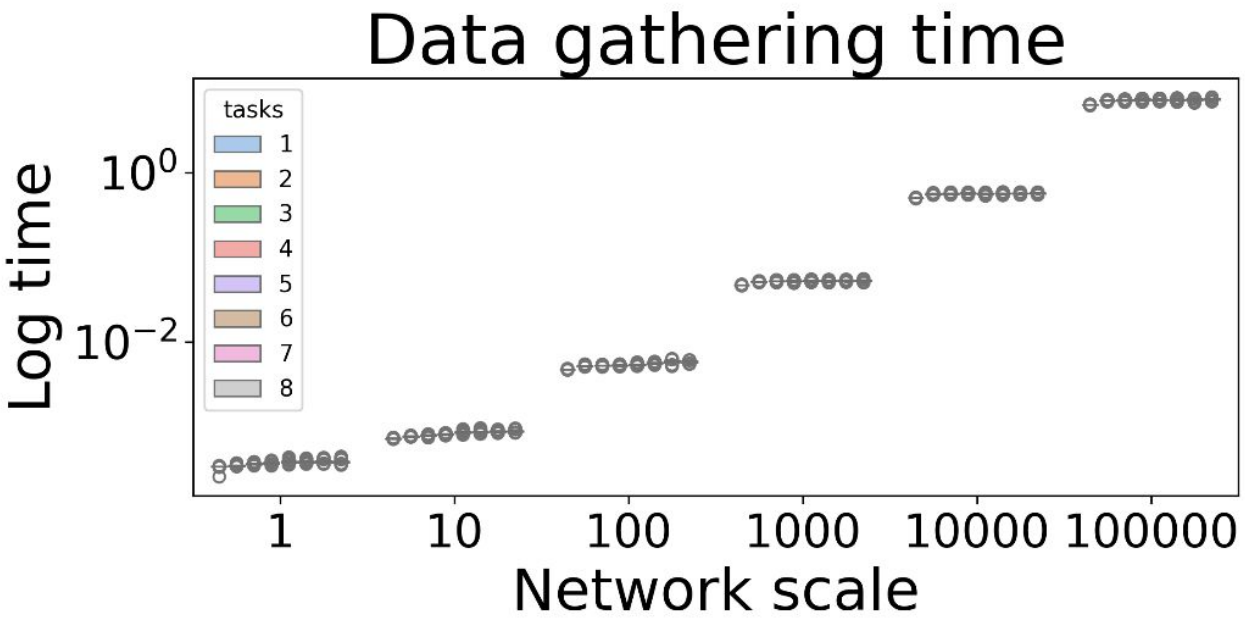


JÜLICH
Forschungszentrum

What about performance?

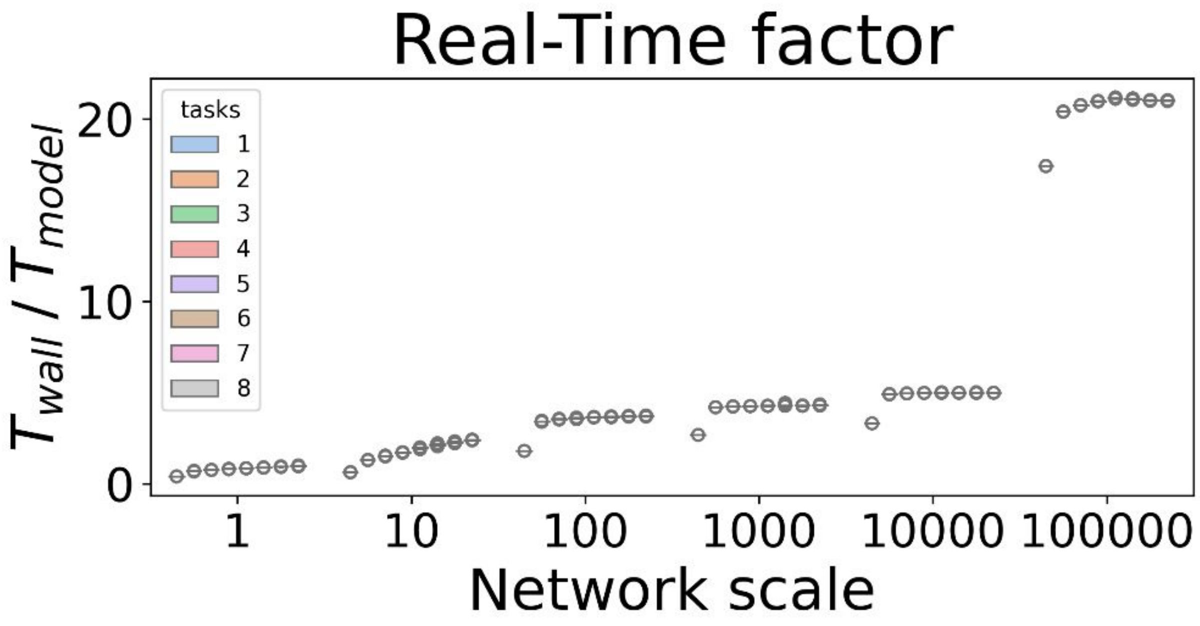
Benchmarking the ring network

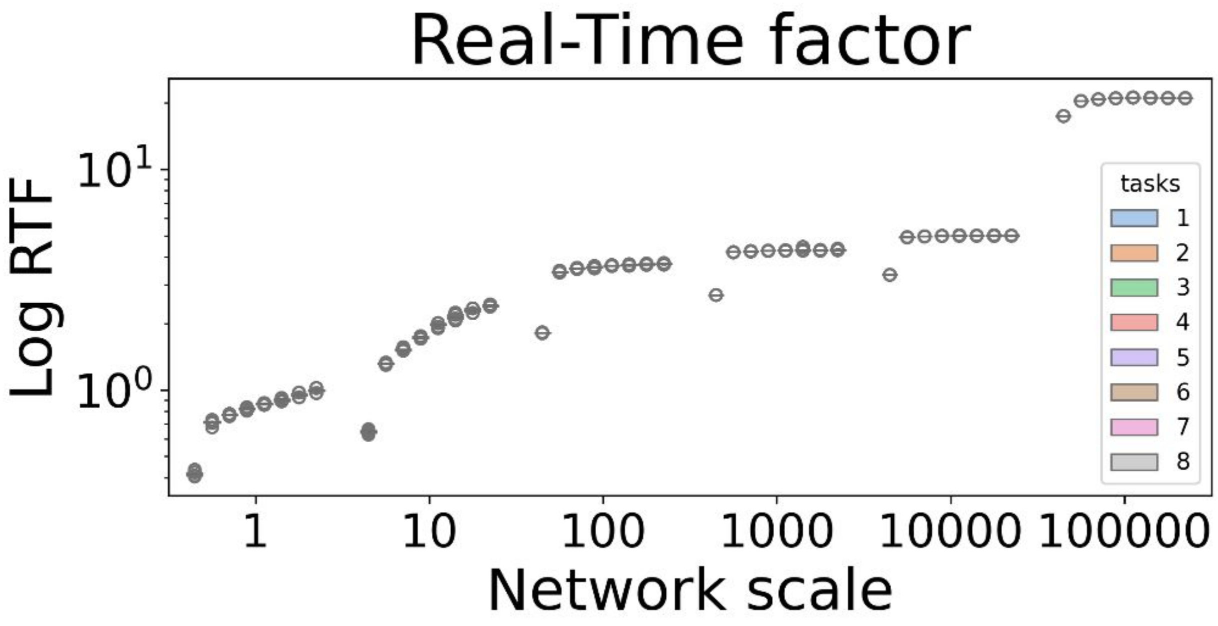




What about performance?

Benchmarking the ring network





JÜLICH Forschungszentrum