Time-series ML-regression on Graphcore IPU-M2000 and Nvidia A100

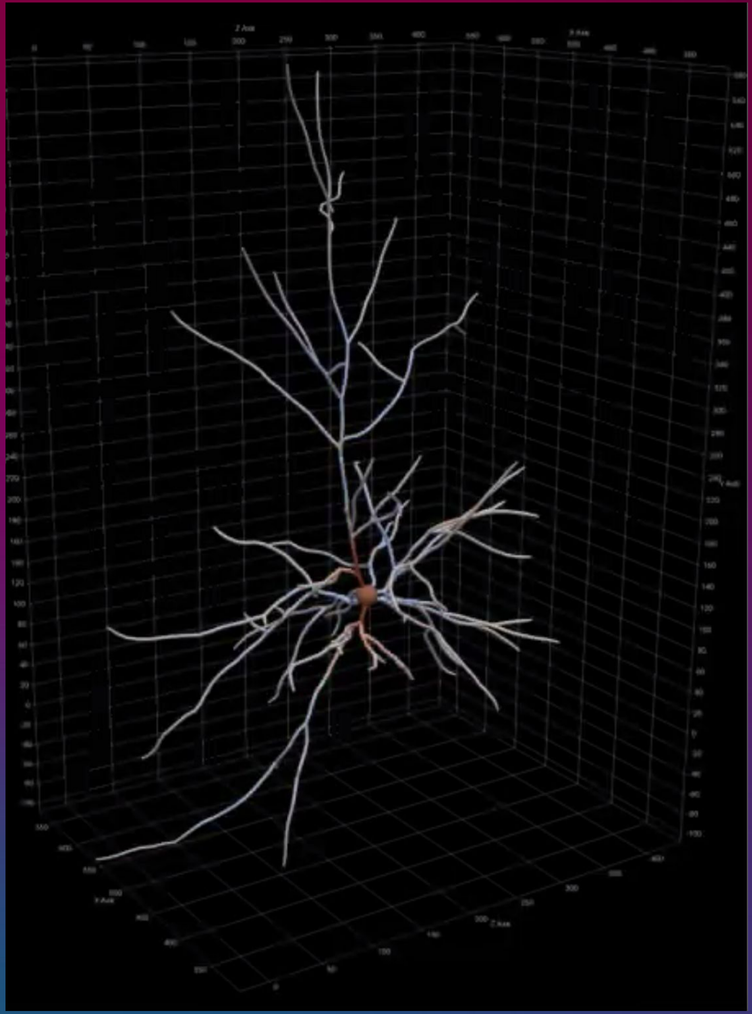
J.Balewski[1] , Z. Liu[2] , A.Tsyplikhin[2] , M. L. Roland[2] , K. Bouchard[3,4,5]

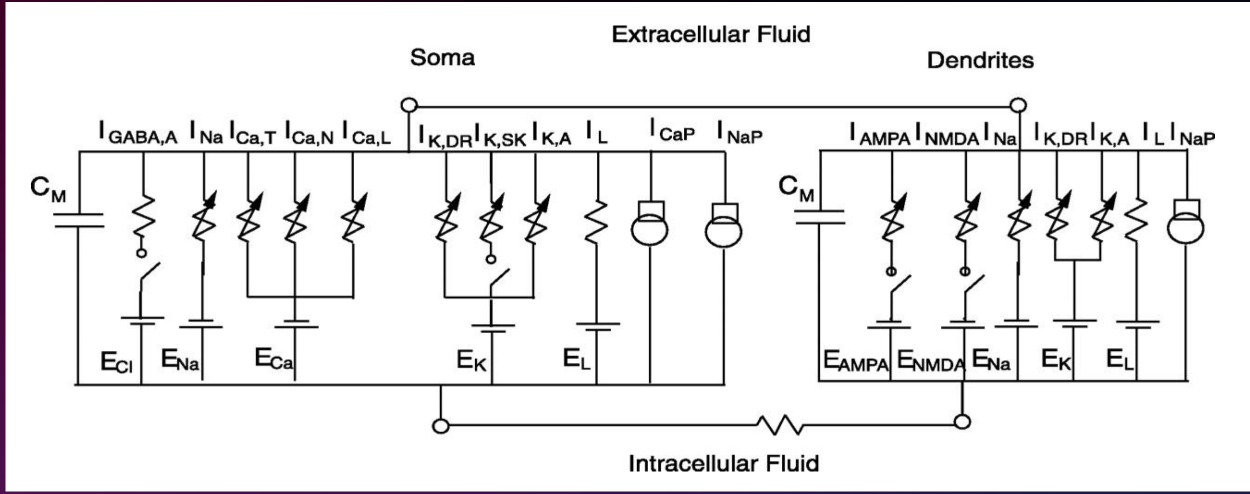
NERSC[1], SD[3] and BSE[4] Divisions, Lawrence Berkeley National Laboratory
Graphcore[2], Palo Alto, CA
Redwood Center for Theoretical Neuroscience[5], UC Berkeley

SC22
Dallas, TX | hpc accelerates.

Neuron-inverter problem

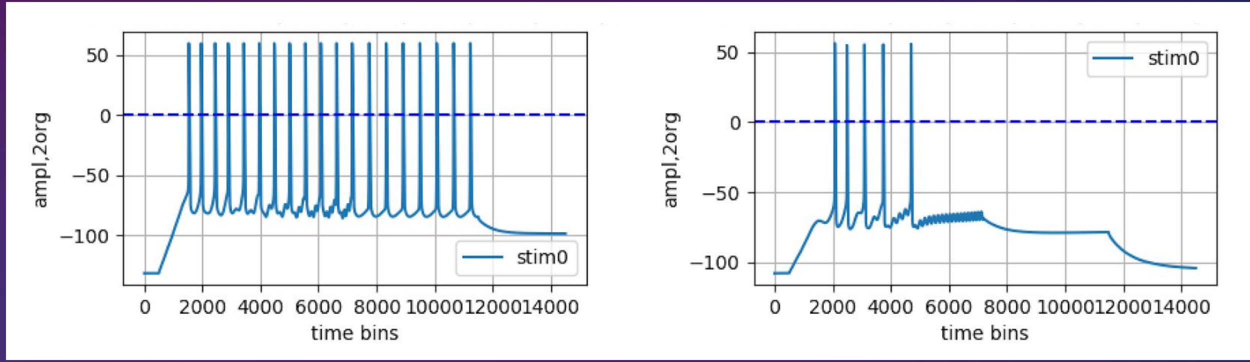
Cell morphology modeled by **lumped circuit representation**





'Natural' problem
given conductances
solve PDE for cell spiking
(aka action potential)

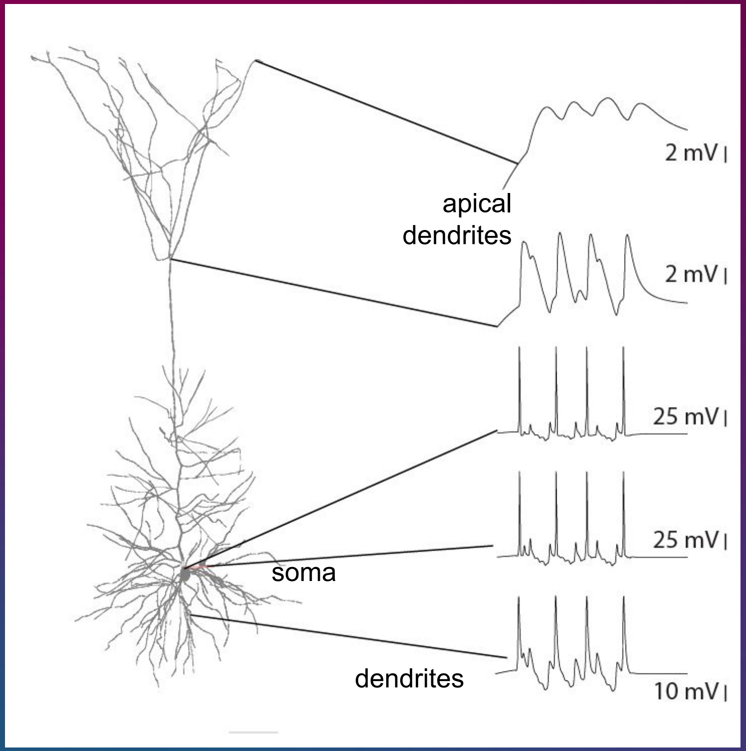
Inverse problem
given spikes
infere conductances

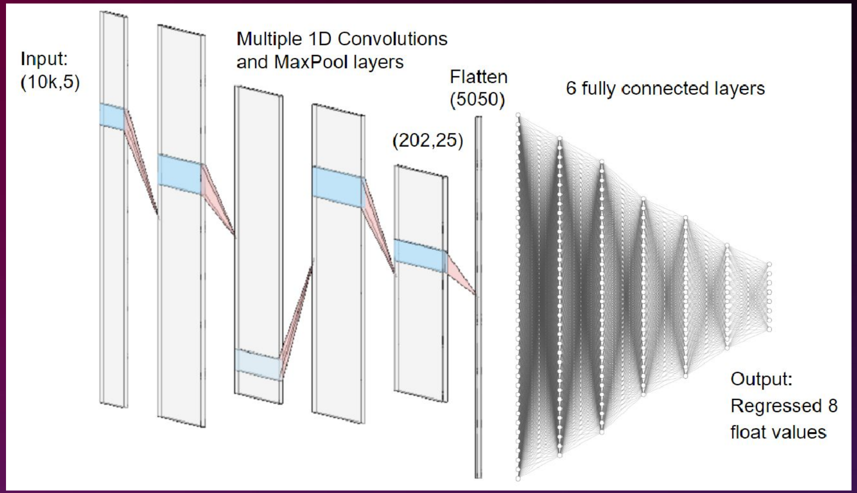


Neuron-inverter ML approach

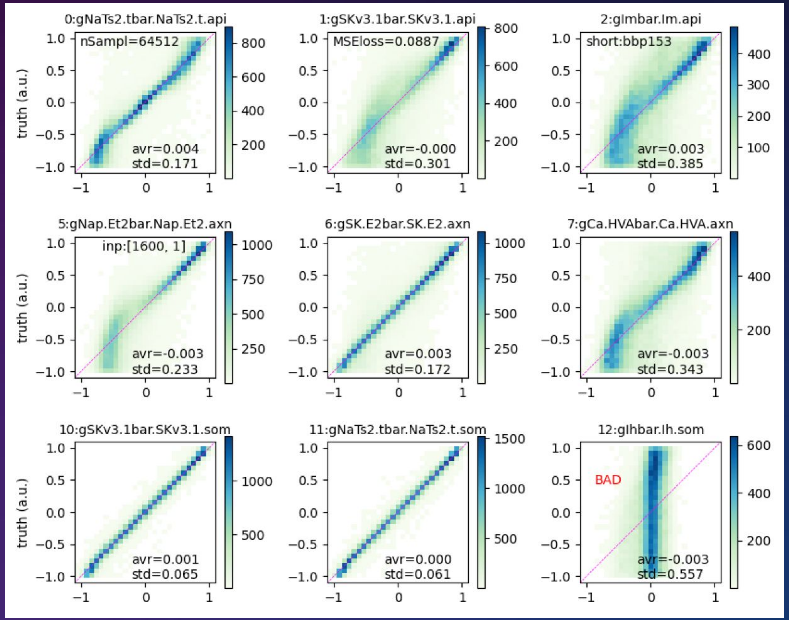
Inferring the mechanisms of neuronal input-output functions

ML Input: Neuron simulated spikes measured at multiple location as 1D time-series





Output: Electrical properties (conductances) determined for different compartments of neuron



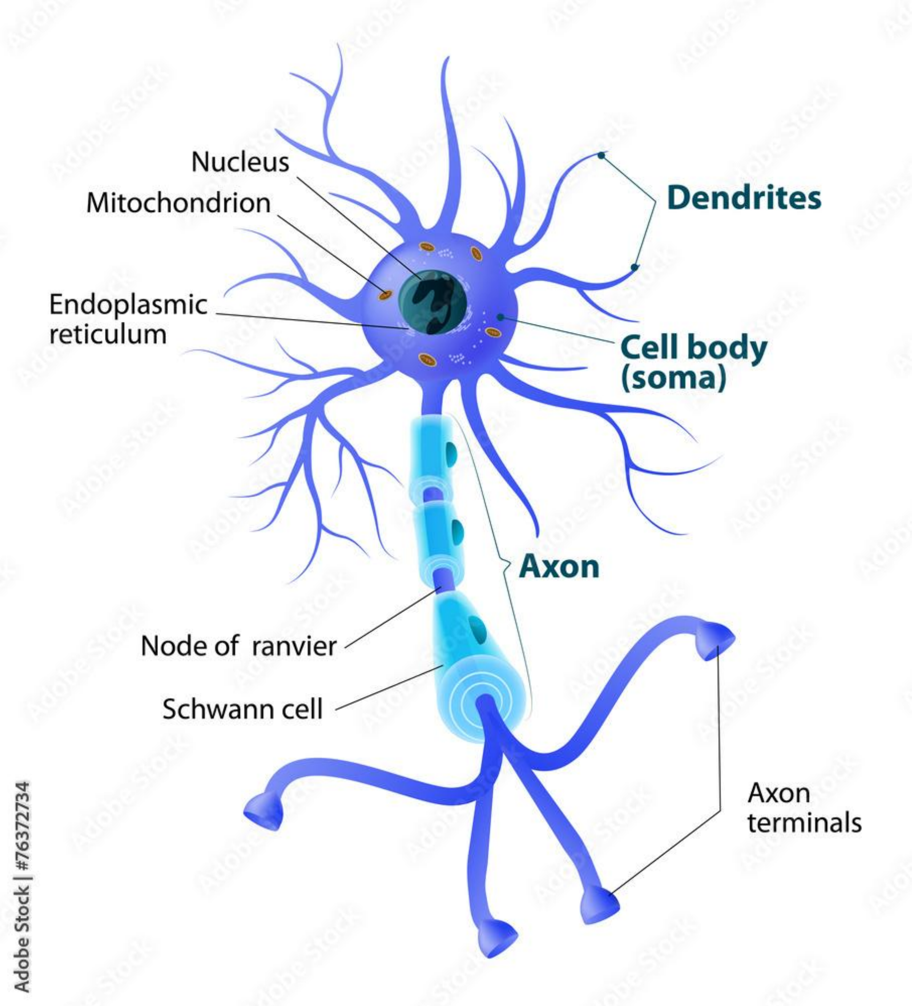
ML objective: regression
INPUT shape (N,1600,4) float
OUTPUT shape (N,15) float
Loss: MSE

ML model: CNN+FC, 2M params
N=500k training samples

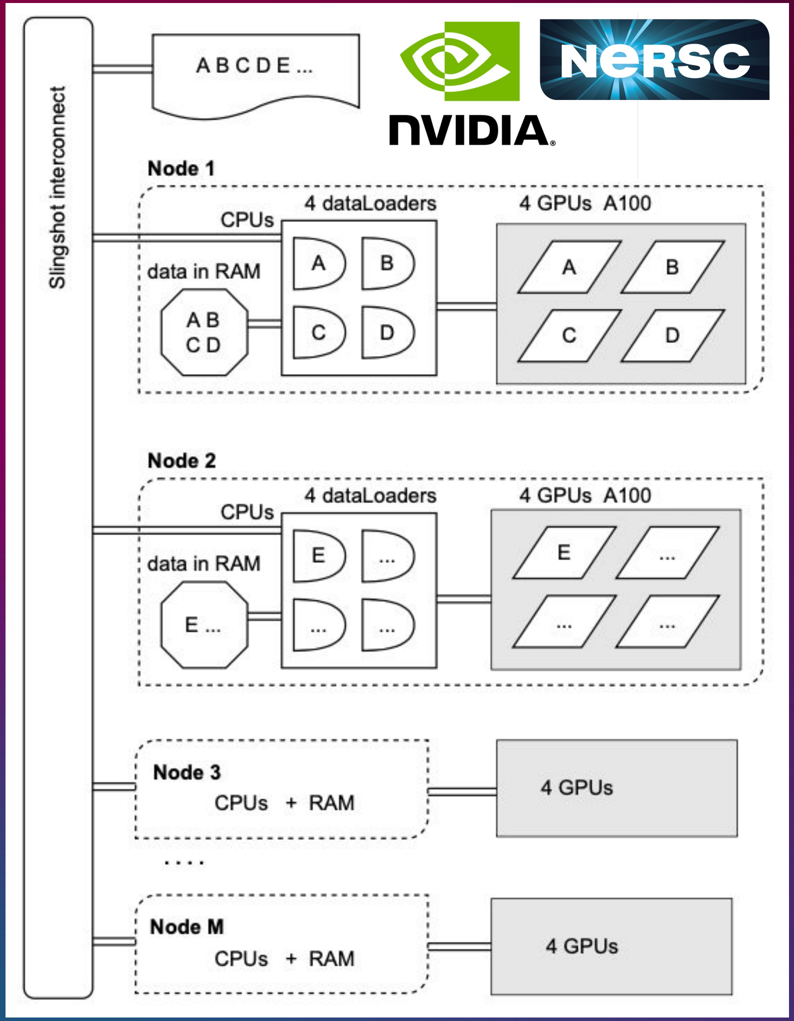
Simulated data
→ ground truth is known

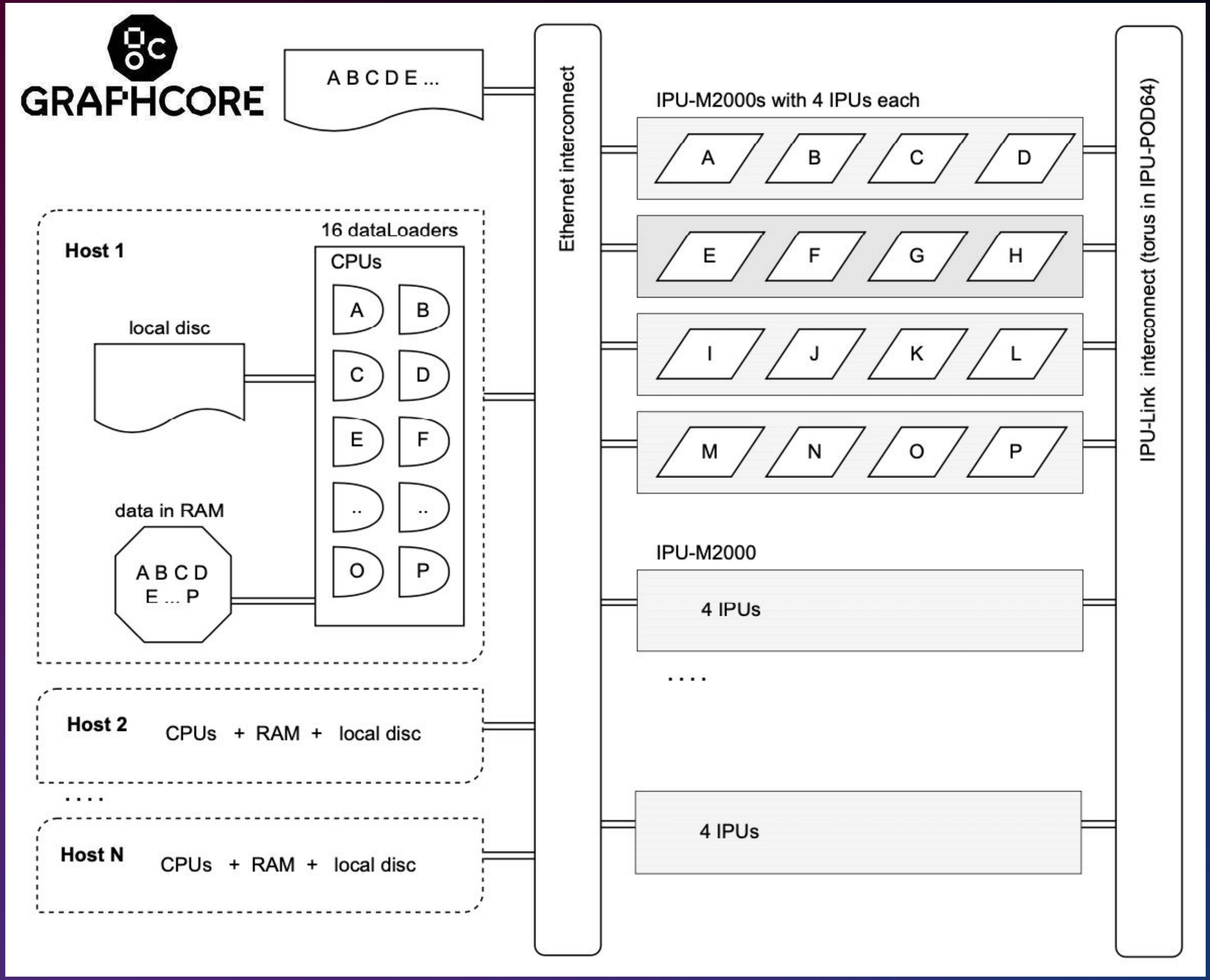
Neuron-inverter ML benchmark

- **Dataset**:
  - simulated spikes (as time series) measured for random conductances
  - 7M training and 700k validations samples
- ML-model : 3M trainable parameters, PyTorch implementation
  - ML-layers: 3 convolutional, 1 batch normalization, 5 fully connected
  - regression loss: MSE
  - optimizer: AdamW
- Training schedule: **fixed** training data, same number of epochs
  - training data distributed in CPUs RAM to avoid any disc-CPU IO cost
  - **constant** local batch size when scaling number of accelerators
  - val-loss used to reduce LR on plateau
    - for GC pseudo-validation loss used instead to avoid graph switching cost
    - true val-loss computed once at the end the whole training (not included in the time-budget)
- Benchmark criteria: end-loss, training time, used energy

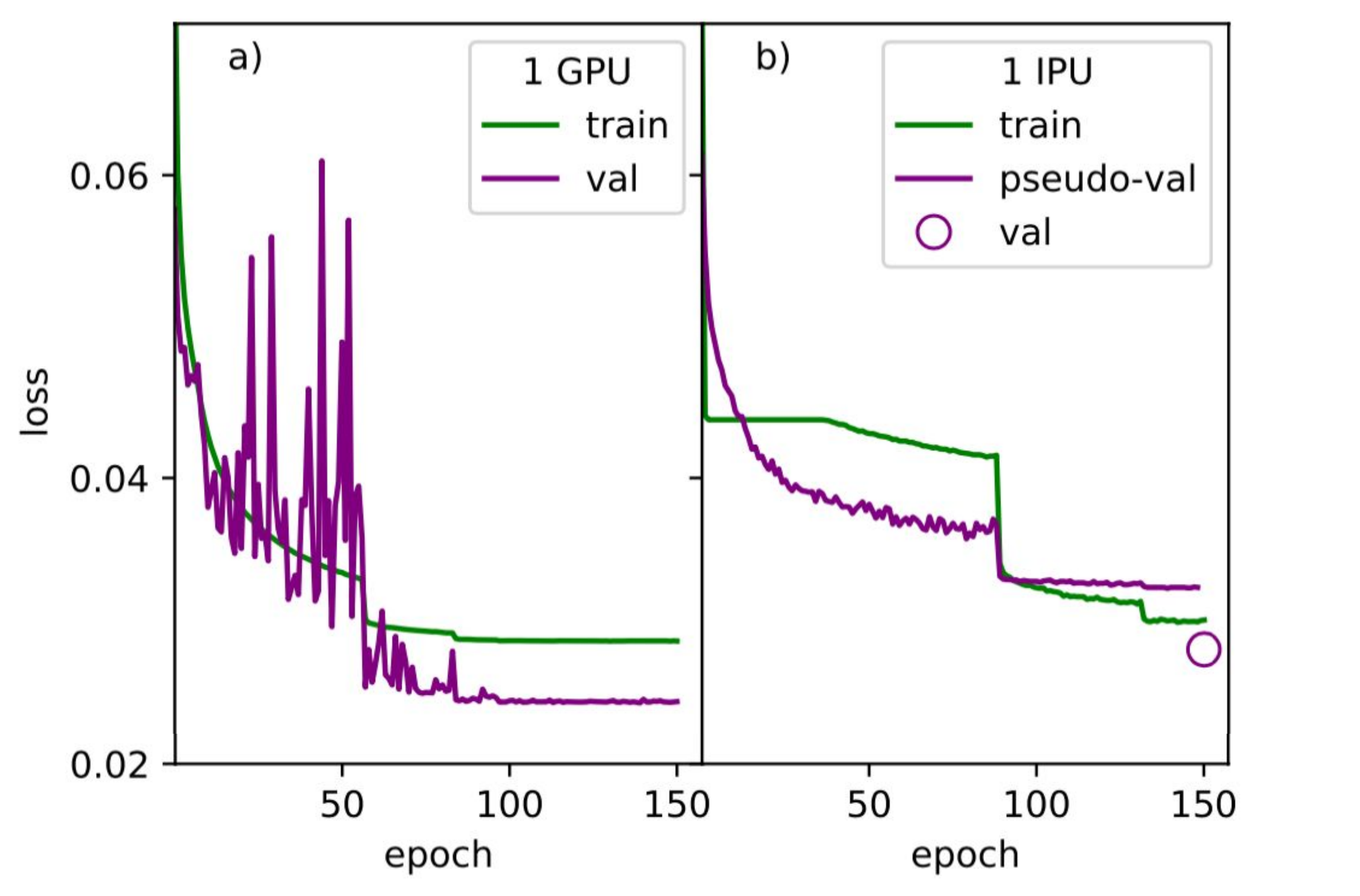


Systems architecture

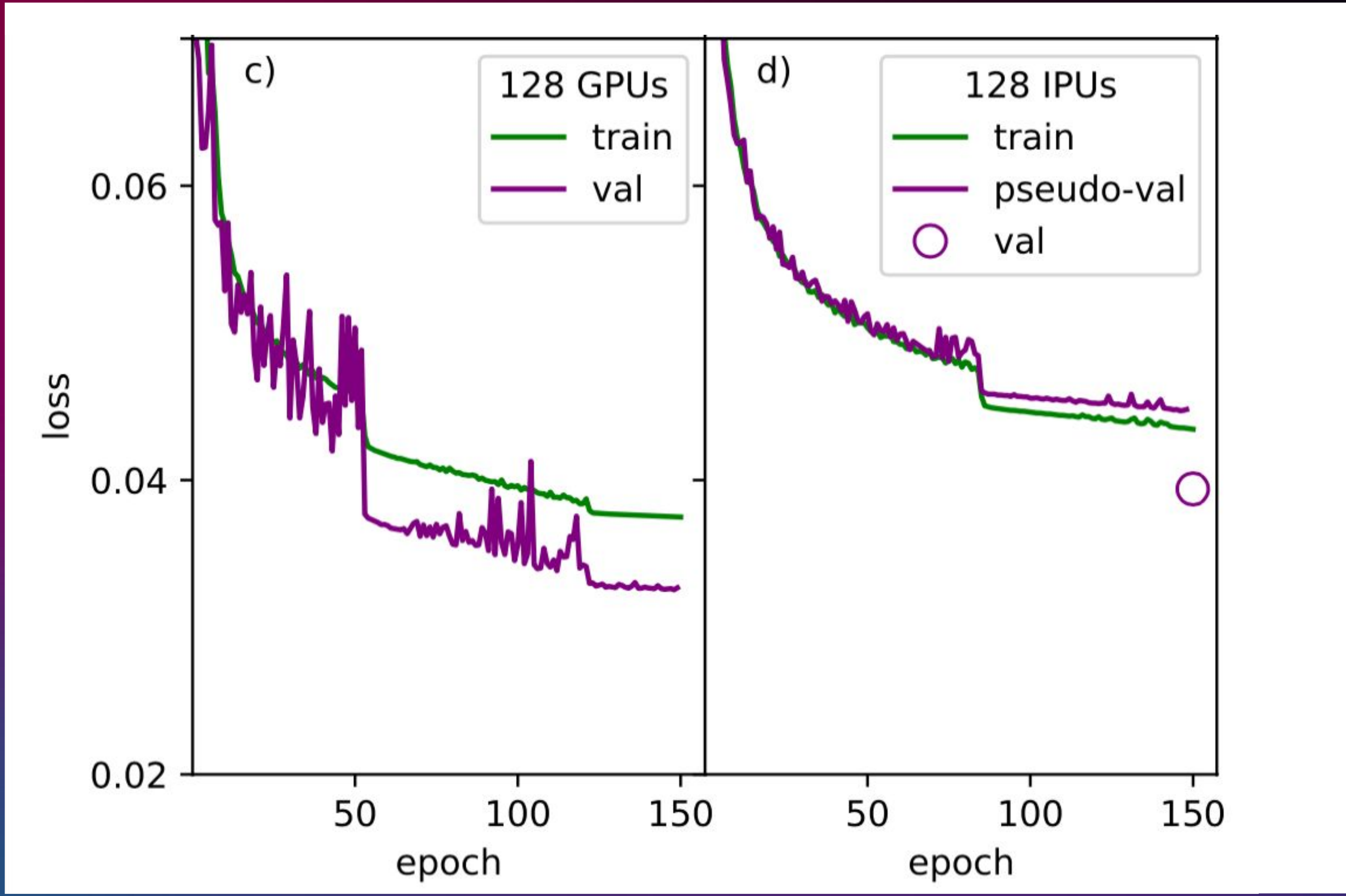




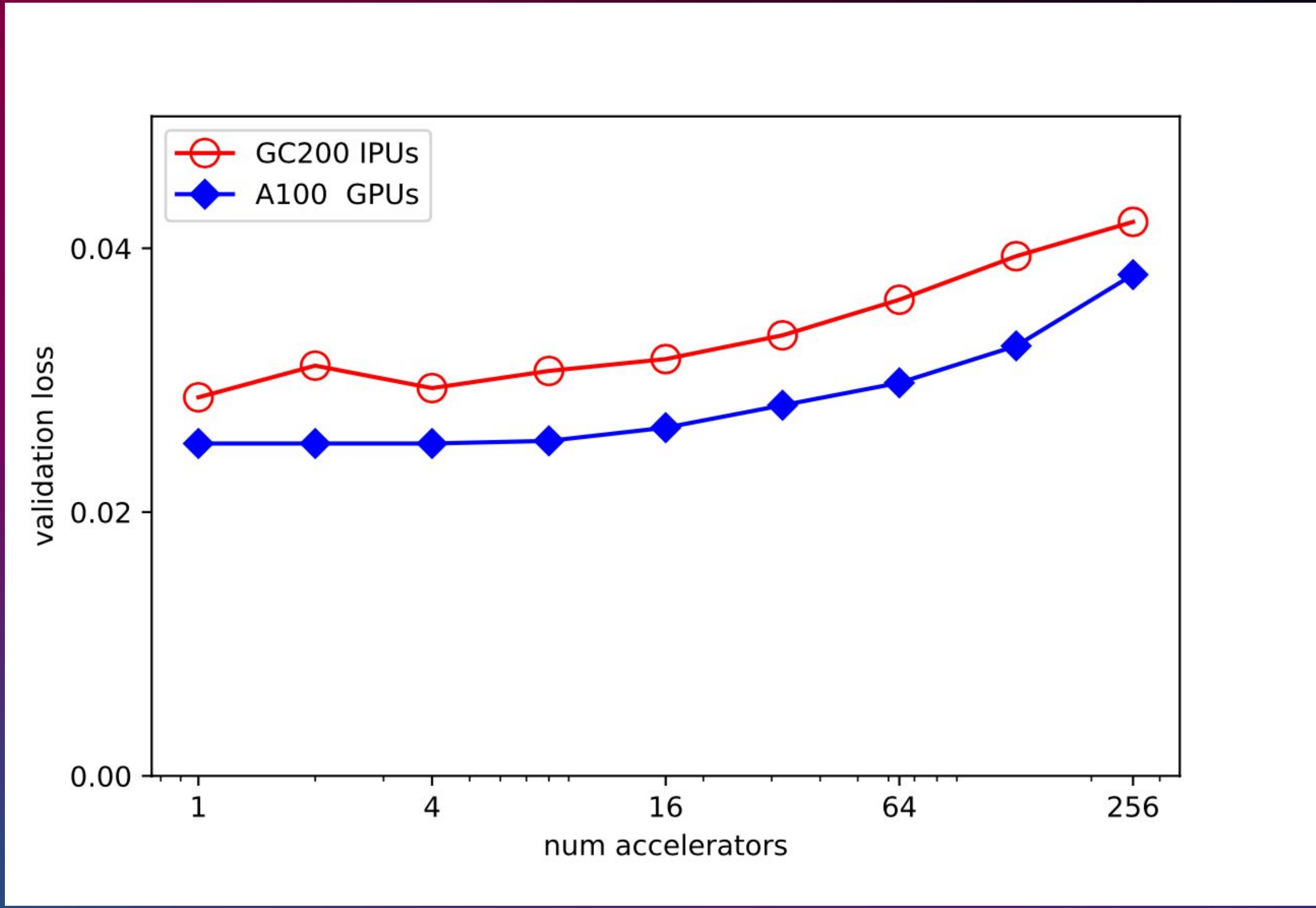
Convergence of **one**-accelerator training



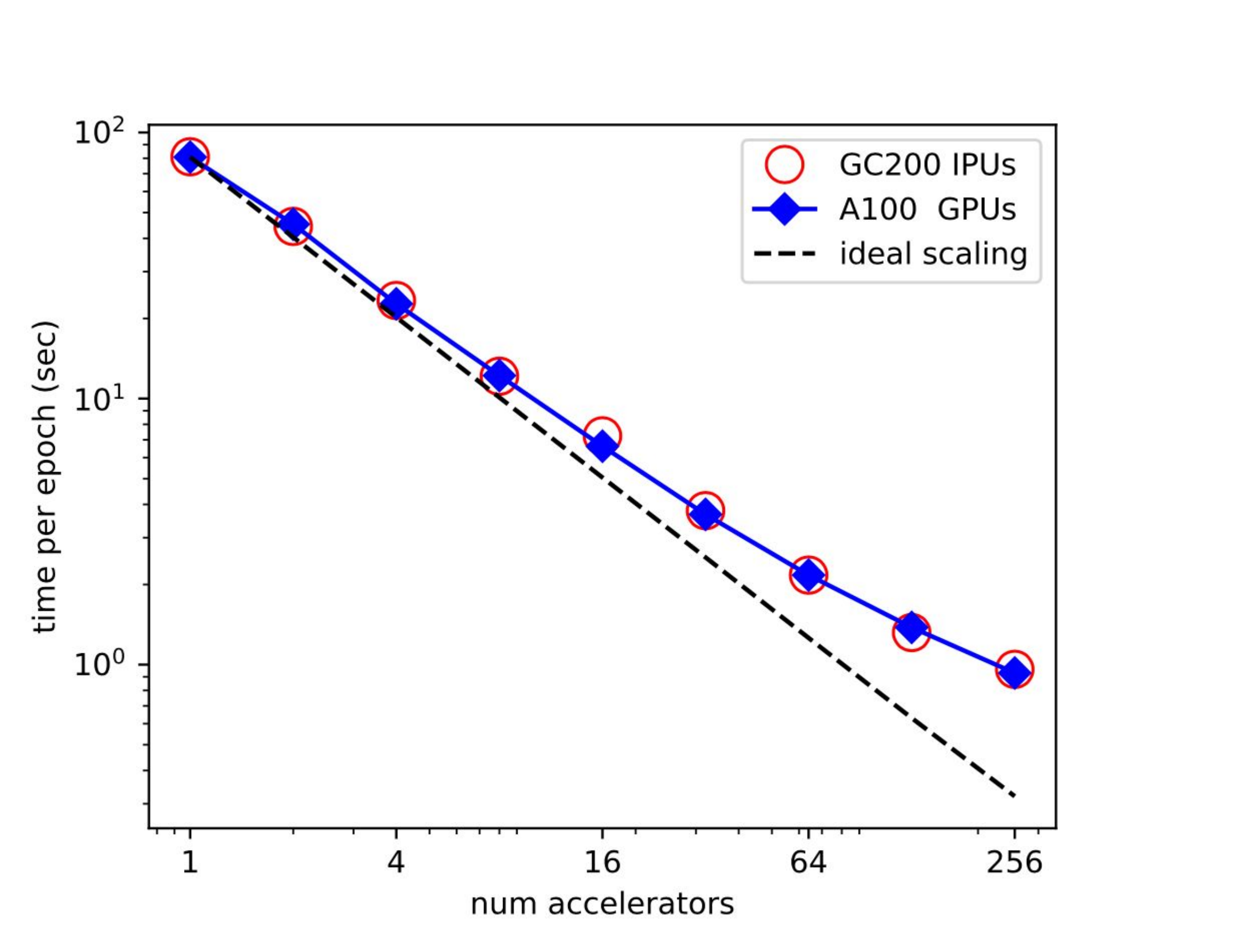
Convergence of **128**-accelerators training



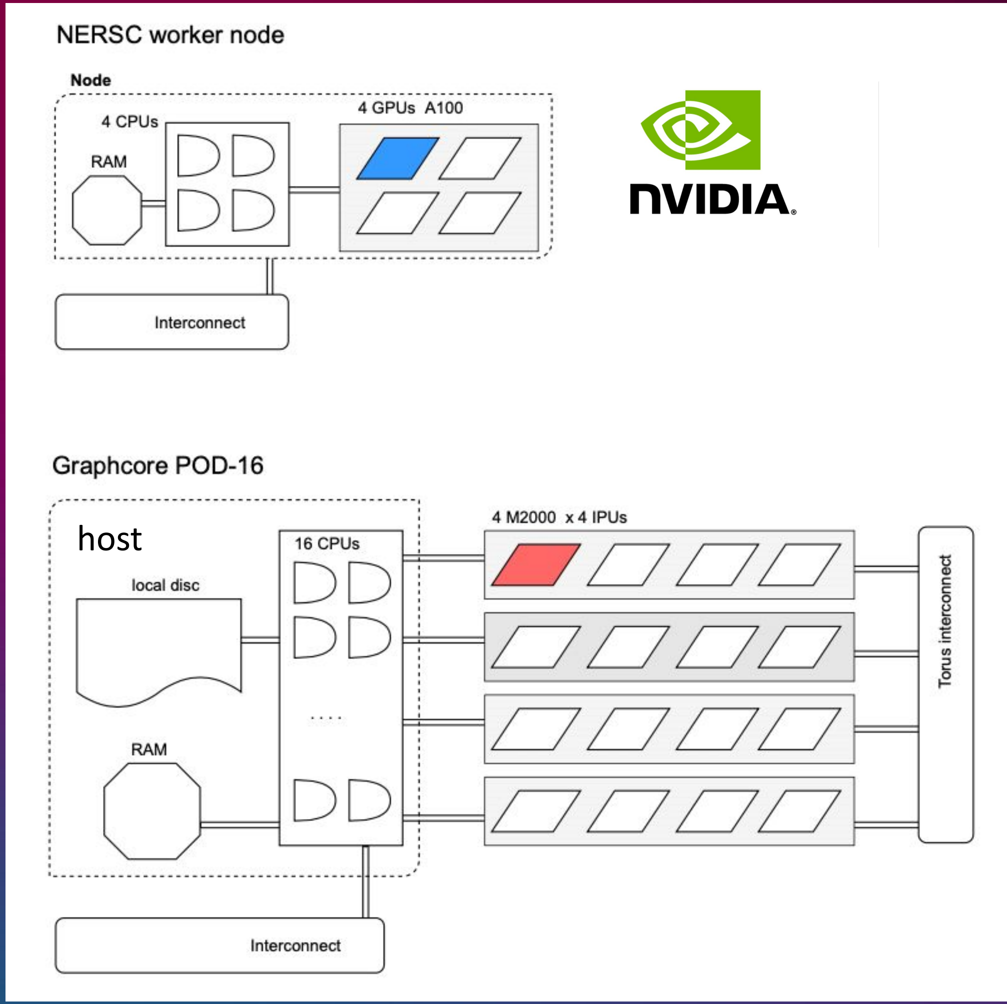
End-loss scaling

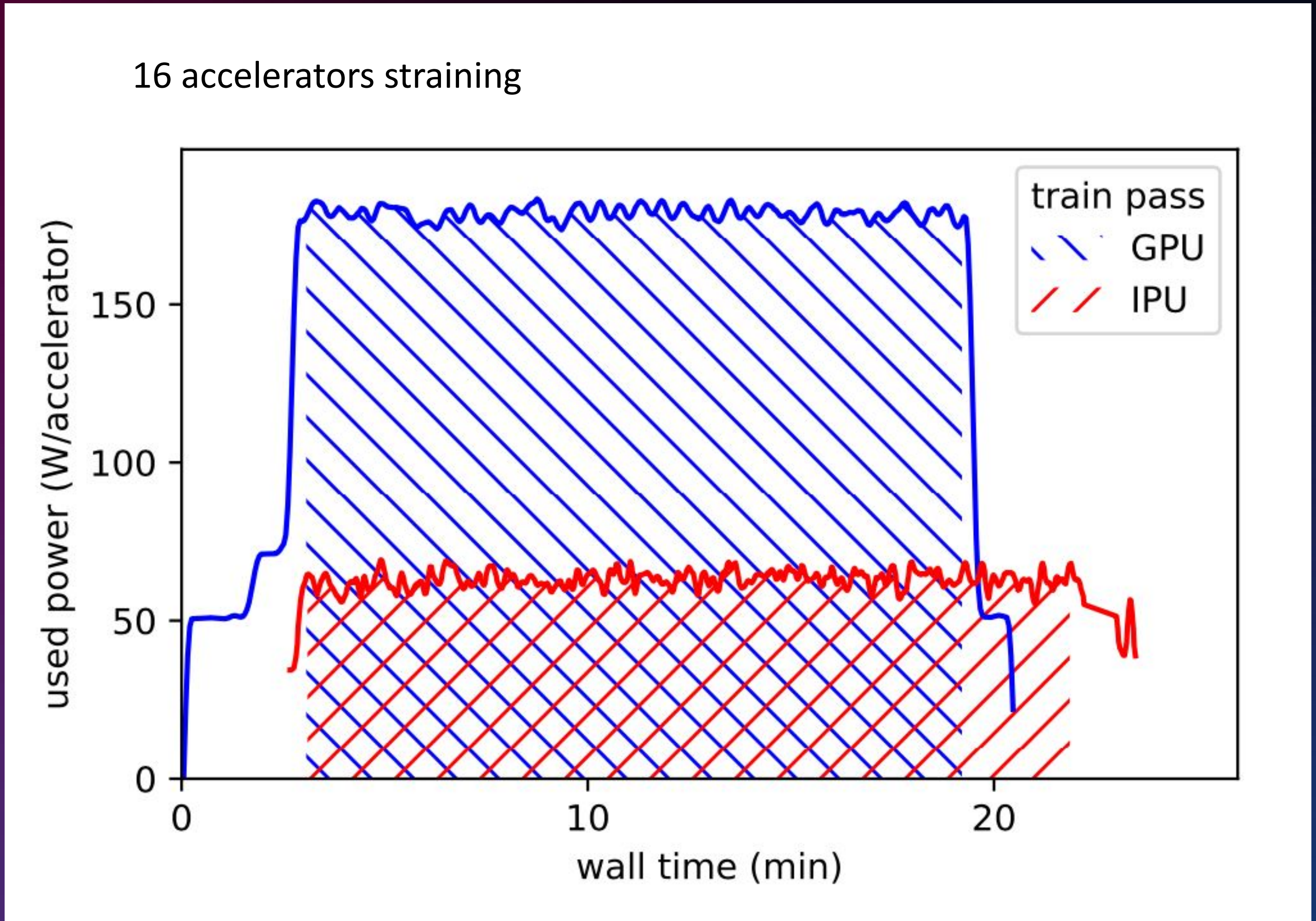


Weak scaling

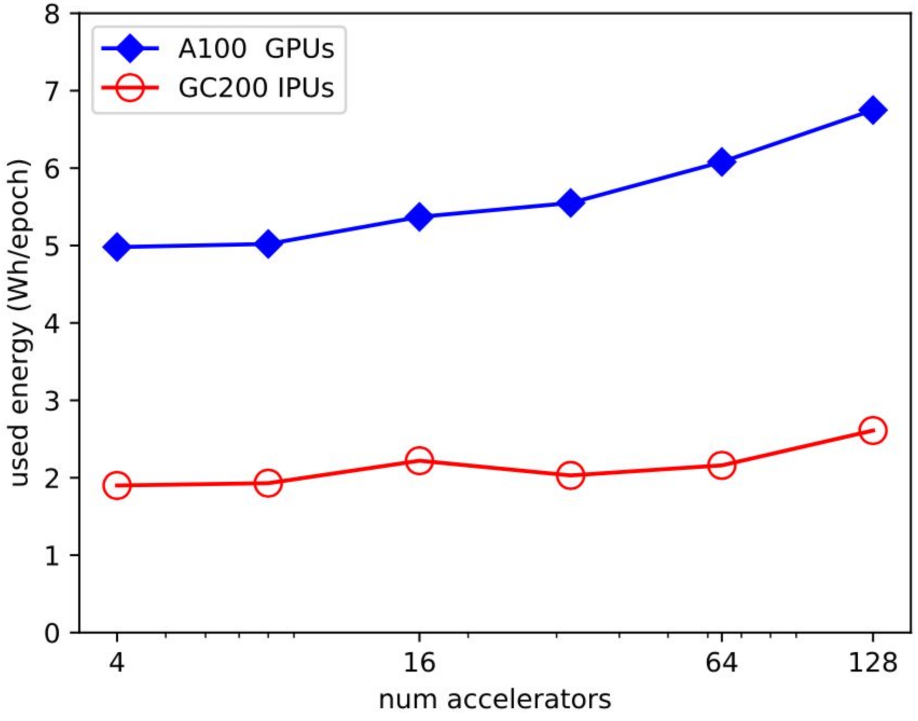


Power consumption profiles





Energy per training usage scaling



Conclusions

- ML is being applied into variety of research projects
  - finding optimal HW is a research topic by itself
- The inversion of PDE has no analytical solution but ML can find the inverse multivalued function
- Neuron-inverter derived from a real neuroscience research project was used as ML benchmark
- The criteria of benchmark were:
  - quality of solution, time to solution, energy consumed until solution was found
- ML benchmark executed on 1 - 256 accelerators from Nvidia (A100) and Graphcore (IPU)
- Results, consisten for any number of accelerators up to 256
  - end-loss achieved on A100s and IPUs were the same within 10-20%
  - total training time was the same within 15%
  - Graphcore chips needed 2.5x less power and used 2x less energy to deliver the above results