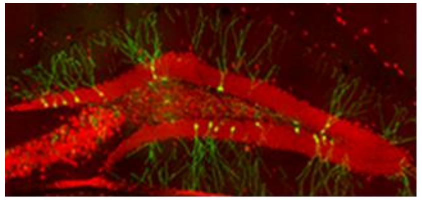
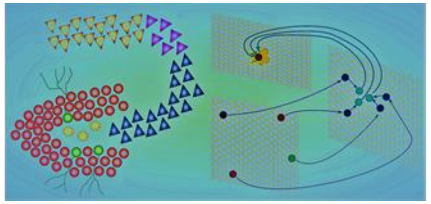
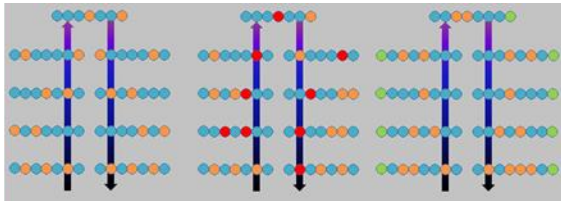
Scaling up Markov Chain Monte Carlo on Loihi



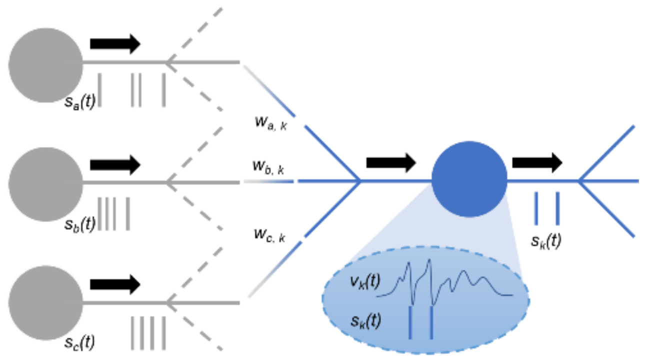




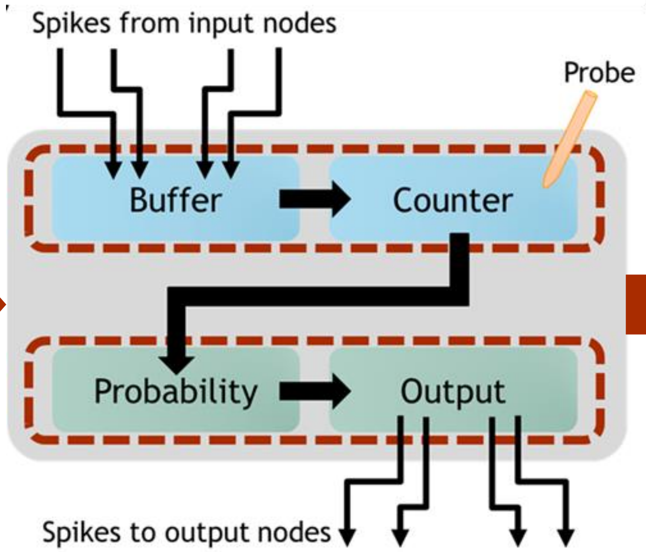
PRESENTED BY

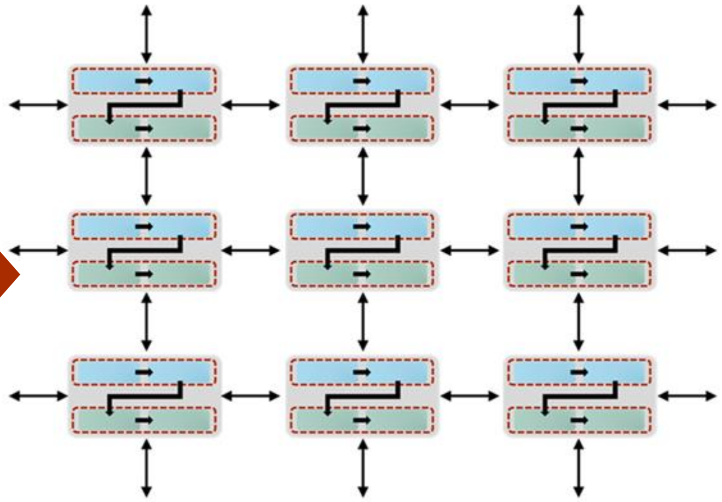
Brad Aimone

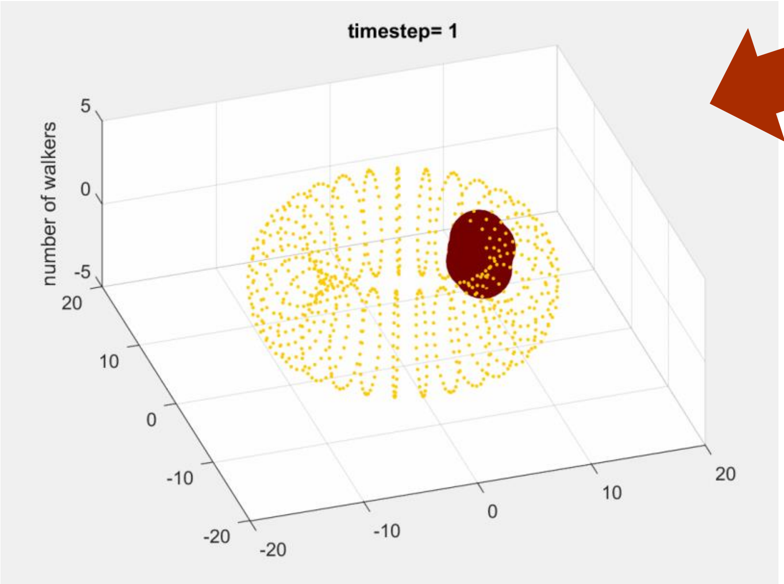
Neuromorphic algorithm can simulate random walks



Leaky Integrate and Fire Neuron





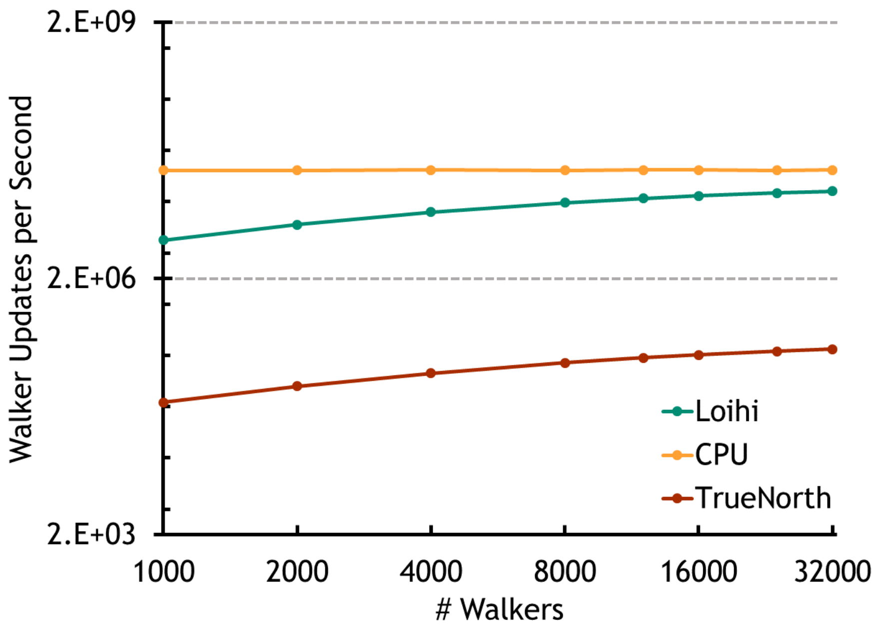


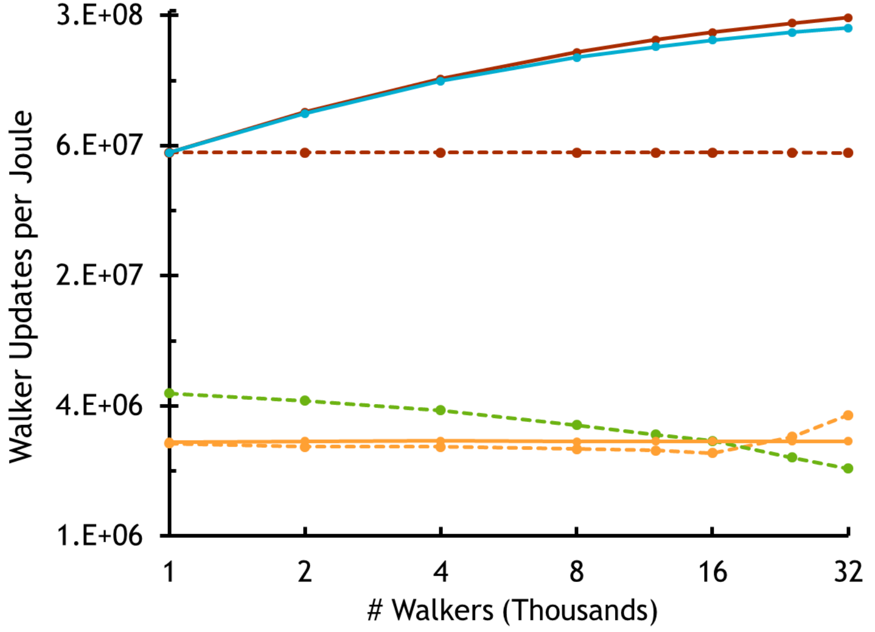
Smith et al., in preparation 2020

We can identify a *neuromorphic advantage* for simulating random walks

We define a *neuromorphic advantage* as an algorithm that shows a demonstrable **advantage** in terms of one resource (e.g., energy) while exhibiting comparable **scaling** in other resources (e.g., time).

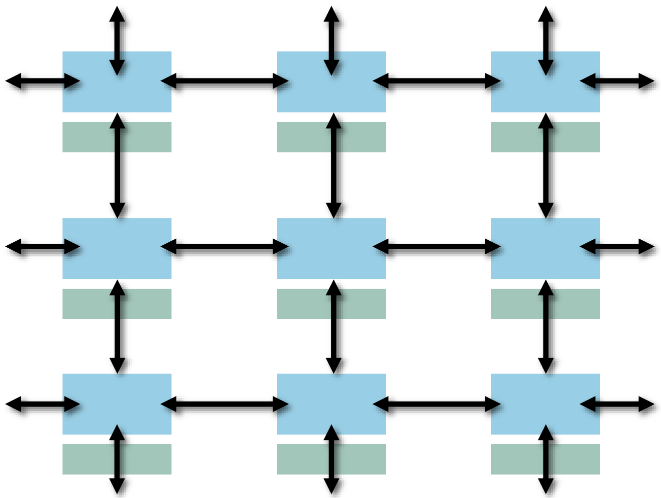
- We show a *neuromorphic advantage* for implementing simple random walks on neuromorphic hardware compared to CPU implementation
  - Same task, architecture specific algorithms
  - TrueNorth and Loihi are slower, but NMC algorithm time scales better
  - **Overall energy consumption (speed / power) is markedly better (20x-100x) on NMC**





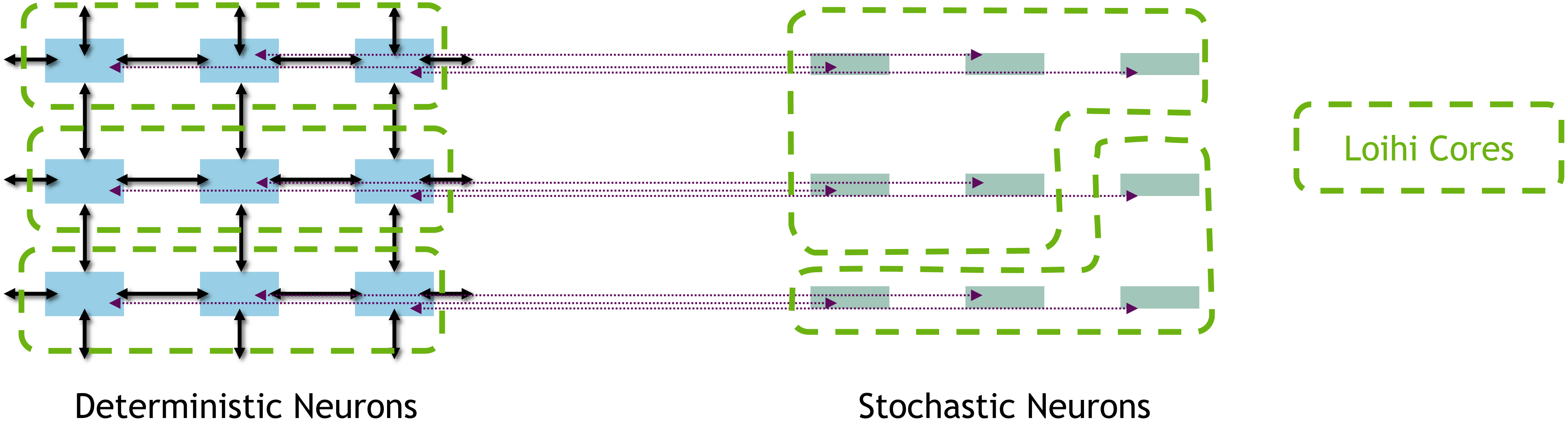
Smith et al., in review 2020

Loihi Mapping



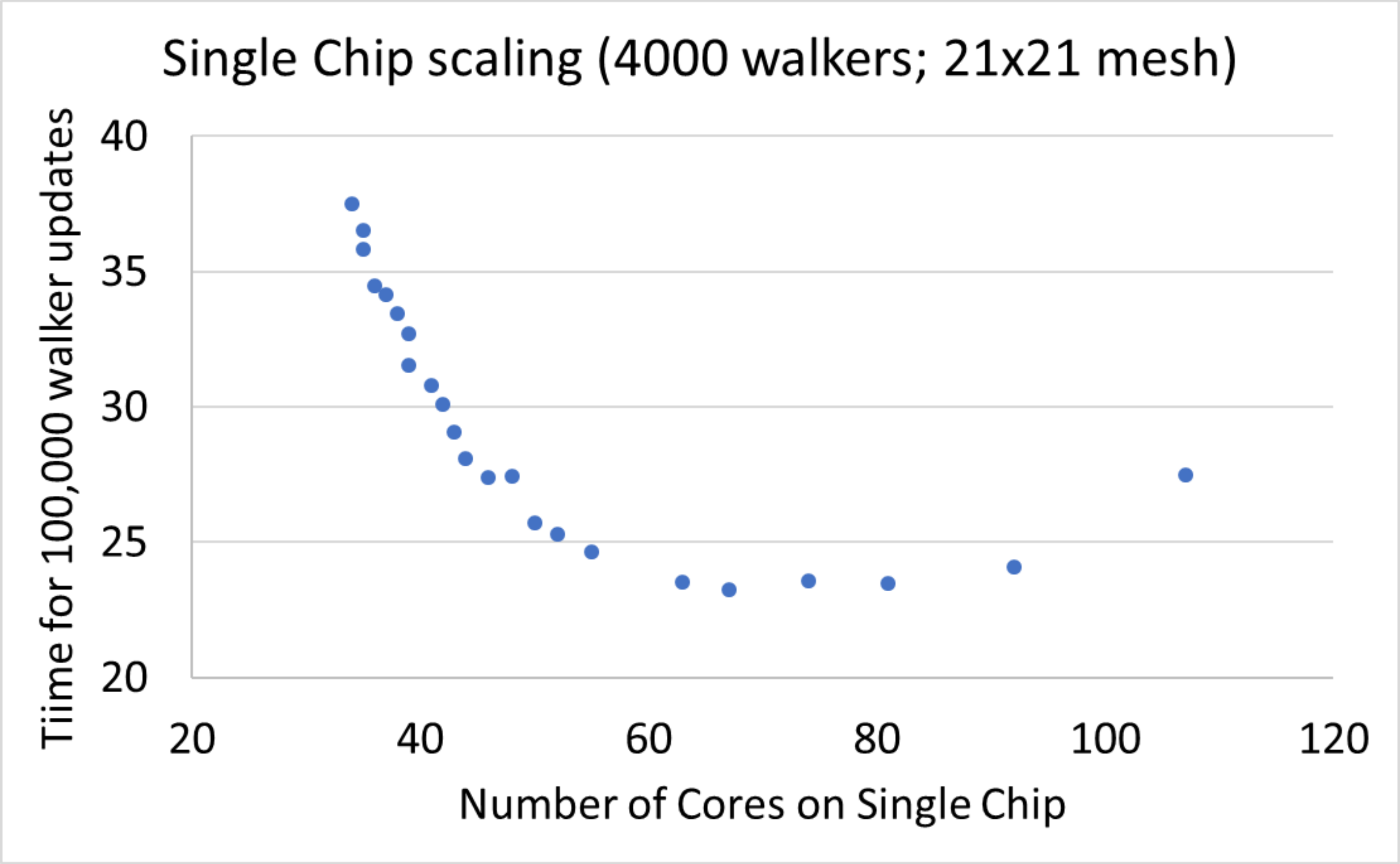
Neural algorithm mixes stochastic and deterministic neurons within each mesh point

Loihi requires that we separate deterministic and stochastic neurons onto different cores



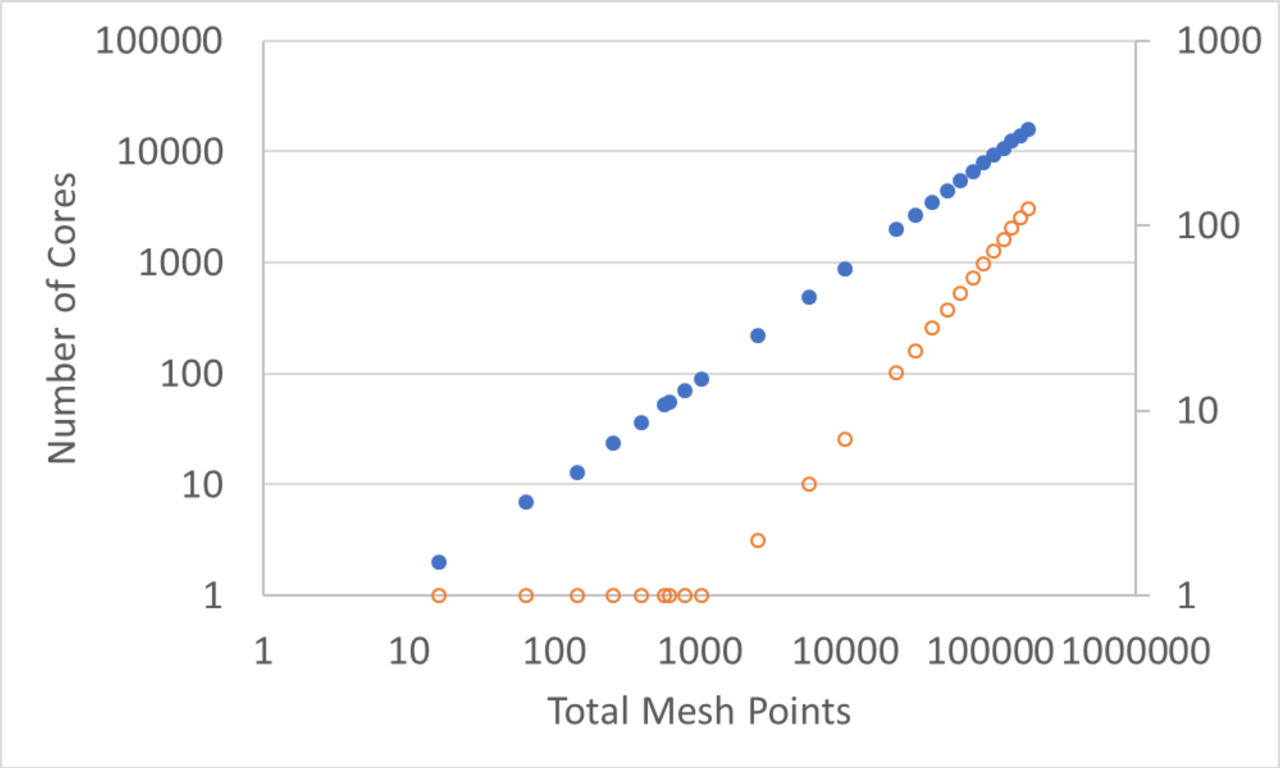
How much of a single chip to use?

- 441 mesh points, ~15 neurons per mesh point
- For models smaller than a single chip, how much of the chip to use?
  - Neural processing is not fully parallel within each core
  - More cores -> more communication
- In a simple study, we see ~50% chip usage to be ideal
  - If model is spread too thin, the benefit of using more cores diminishes



Scaling to many chips

- Sandia has 50 million neuron Loihi testbed system
- First crack at a scaling study
  - Keep # walkers constant
  - Increase mesh size beyond single chip level
- The value of this algorithmic approach really is at large scale
  - No one needs to do random walks over 441 mesh points
  - The ability to run millions of random walks over millions of mesh points quickly is of significant value in a wide range of applications



Early data

- It works pretty well!
  - Scaling up simulation was pretty seamless
    - Build bigger mesh in NetworkX
    - Embed bigger mesh through NxNet onto large-scale Loihi platform
- Not optimized
  - Graph embedding
    - We should keep linked stochastic and deterministic neurons on same chip
    - Connected mesh points should be proximal on system
  - Walker density, etc.