Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

COINFLIPS

How probabilistic neuromorphic computing may impact scientific computing applications

Brad Aimone

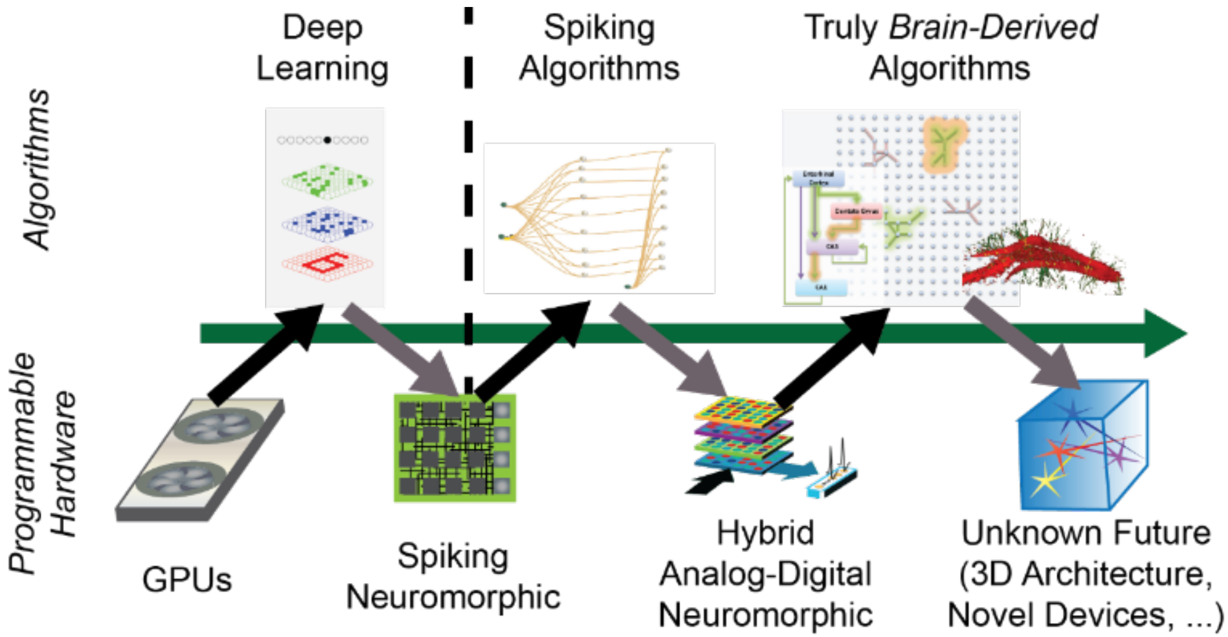
jbaimon@sandia.gov

How probabilistic neuromorphic computing may impact scientific computing applications

Brad Aimone

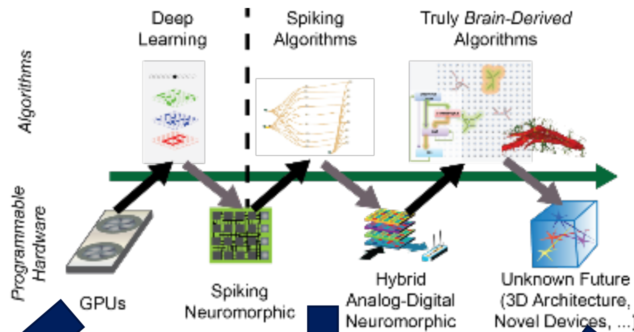
jbaimon@sandia.gov

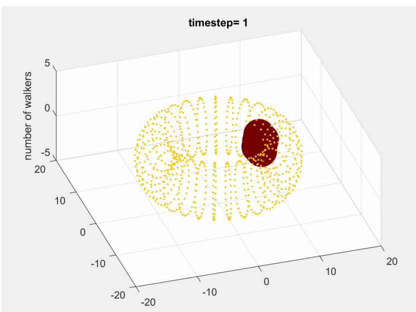
Neuromorphic computing is an evolving field



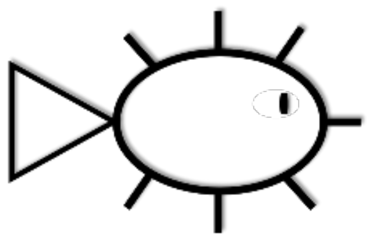
Aimone, Advanced Intelligent Systems, 2021

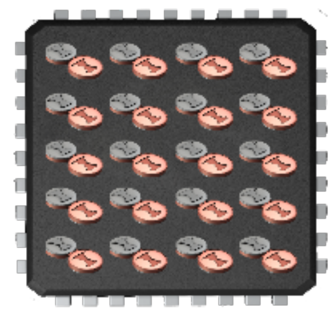
Neuromorphic today and tomorrow: impact of neuromorphic in probabilistic scientific computing





Part 1 – Neuromorphic advantage on a scientific computing task





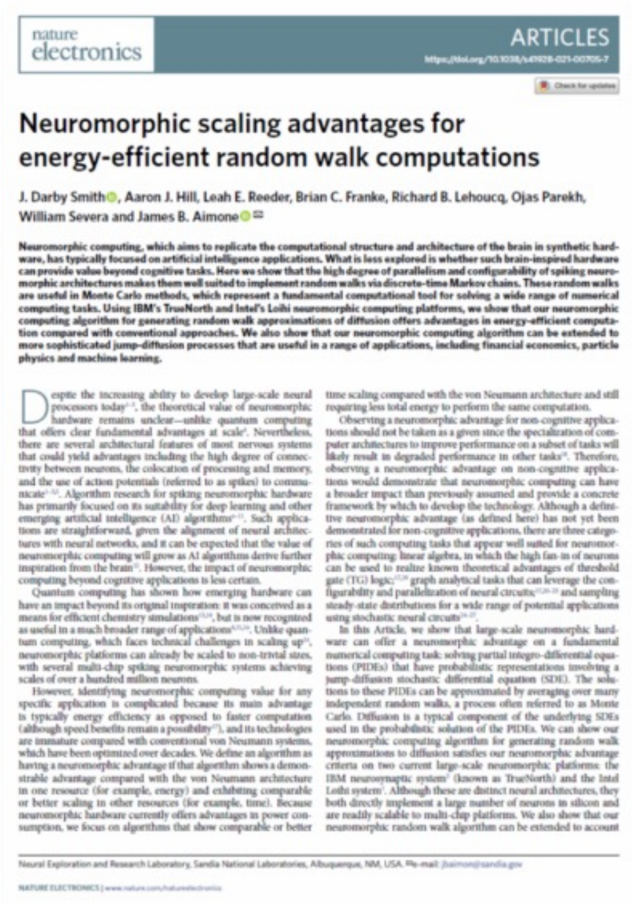
Part 3 – Future Probabilistic COINFLIPS Architecture

Part 2 – Neuromorphic Mini-Apps to understand impact on HPC

Neuromorphic hardware shows advantages for probabilistic algorithms

Part 1

COINFLIPS


nature electronics

ARTICLES

https://doi.org/10.1038/s41928-021-00705-7

Check for updates

Neuromorphic scaling advantages for energy-efficient random walk computations

J. Darby Smith, Aaron J. Hill, Leah E. Reeder, Brian C. Franke, Richard B. Lehoucq, Ojas Parekh, William Severa and James B. Aimone

Neuromorphic computing, which aims to replicate the computational structure and architecture of the brain in synthetic hardware, has typically focused on artificial intelligence applications. What is less explored is whether such brain-inspired hardware can provide value beyond cognitive tasks. Here we show that the high degree of parallelism and configurability of spiking neuromorphic architectures makes them well suited to implement random walks via discrete-time Markov chains. These random walks are useful in Monte Carlo methods, which represent a fundamental computational tool for solving a wide range of numerical computing tasks. Using IBM's TrueNorth and Intel's Loihi neuromorphic computing platforms, we show that our neuromorphic computing algorithm for generating random walk approximations of diffusion offers advantages in energy-efficient computation compared with conventional approaches. We also show that our neuromorphic computing algorithm can be extended to more sophisticated jump-diffusion processes that are useful in a range of applications, including financial economics, particle physics and machine learning.

Despite the increasing ability to develop large-scale neural processors today, the theoretical value of neuromorphic hardware remains unclear—unlike quantum computing that offers clear fundamental advantages at scale. Nevertheless, there are several architectural features of most nervous systems that could yield advantages including the high degree of connectivity between neurons, the colocation of processing and memory, and the use of action potentials (referred to as spikes) to communicate. Algorithm research for spiking neuromorphic hardware has primarily focused on its suitability for deep learning and other emerging artificial intelligence (AI) algorithms. Such applications are straightforward, given the alignment of neural architectures with neural networks, and it can be expected that the value of neuromorphic computing will grow as AI algorithms derive further inspiration from the brain. However, the impact of neuromorphic computing beyond cognitive applications is less certain.

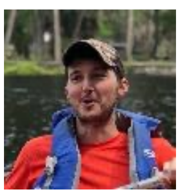
Quantum computing has shown how emerging hardware can have an impact beyond its original inspiration: it was conceived as a means for efficient chemistry simulations, but is now recognized as useful in a much broader range of applications. Unlike quantum computing, which faces technical challenges in scaling up, neuromorphic platforms can already be scaled to non-trivial sizes, with several multi-chip spiking neuromorphic systems achieving scales of over a hundred million neurons.

However, identifying neuromorphic computing value for any specific application is complicated because its main advantage is typically energy efficiency as opposed to faster computation (although speed benefits remain a possibility), and its technologies are immature compared with conventional von Neumann systems, which have been optimized over decades. We define an algorithm as having a neuromorphic advantage if that algorithm shows a demonstrable advantage compared with the von Neumann architecture in one resource (for example, energy) and exhibiting comparable or better scaling in other resources (for example, time). Because neuromorphic hardware currently offers advantages in power consumption, we focus on algorithms that show comparable or better time scaling compared with the von Neumann architecture and still requiring less total energy to perform the same computation.

Observing a neuromorphic advantage for non-cognitive applications should not be taken as a given since the specialization of computer architectures to improve performance on a subset of tasks will likely result in degraded performance in other tasks. Therefore, observing a neuromorphic advantage on non-cognitive applications would demonstrate that neuromorphic computing can have a broader impact than previously assumed and provide a concrete framework by which to develop the technology. Although a definitive neuromorphic advantage (as defined here) has not yet been demonstrated for non-cognitive applications, there are three categories of such computing tasks that appear well suited for neuromorphic computing: linear algebra, in which the high fan-in of neurons can be used to realize known theoretical advantages of threshold gate (TG) logic; graph analytical tasks that can leverage the configurability and parallelization of neural circuits; and sampling steady-state distributions for a wide range of potential applications using stochastic neural circuits.

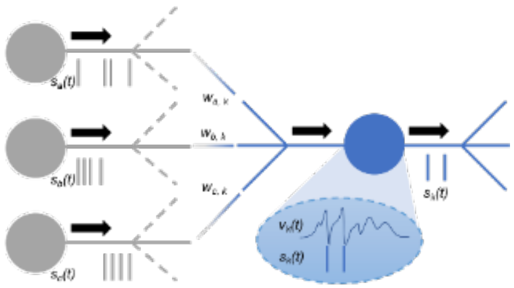
In this Article, we show that large-scale neuromorphic hardware can offer a neuromorphic advantage on a fundamental numerical computing task: solving partial integro-differential equations (PIDEs) that have probabilistic representations involving a jump-diffusion stochastic differential equation (SDE). The solutions to these PIDEs can be approximated by averaging over many independent random walks, a process often referred to as Monte Carlo. Diffusion is a typical component of the underlying SDEs used in the probabilistic solution of the PIDEs. We can show our neuromorphic computing algorithm for generating random walk approximations to diffusion satisfies our neuromorphic advantage criteria on two current large-scale neuromorphic platforms: the IBM neurosynaptic system (known as TrueNorth) and the Intel Loihi system. Although these are distinct neural architectures, they both directly implement a large number of neurons in silicon and are readily scalable to multi-chip platforms. We also show that our neuromorphic random walk algorithm can be extended to account

Neural Exploration and Research Laboratory, Sandia National Laboratories, Albuquerque, NM, USA. e-mail: jbaimon@sandia.gov

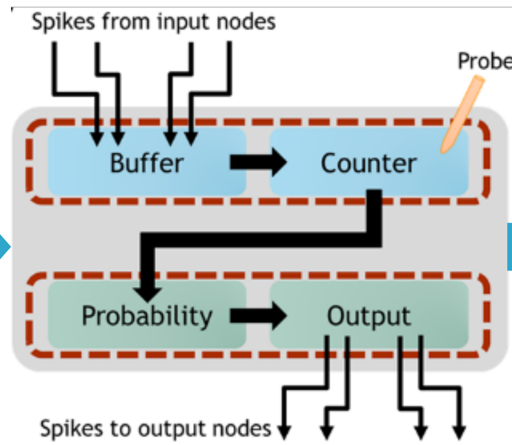


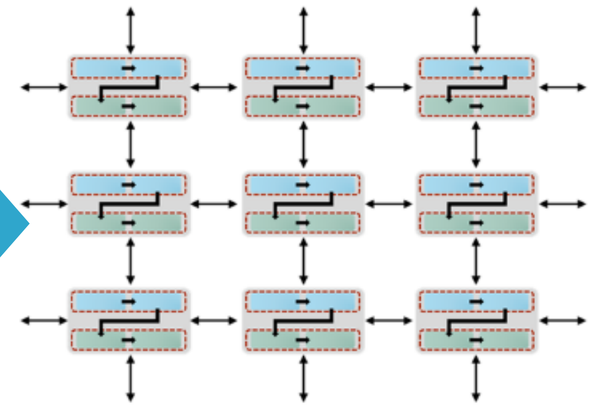
Darby Smith

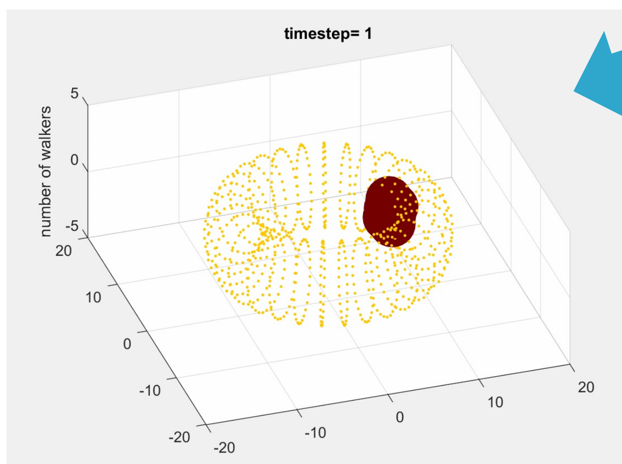
Neuromorphic algorithm can simulate random walks



Leaky Integrate and Fire Neuron

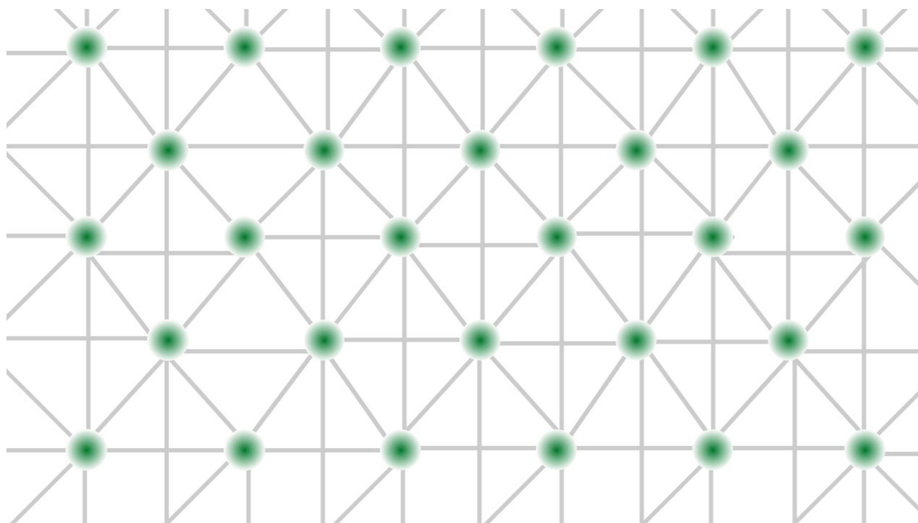






Smith et al., Nature Electronics 2022

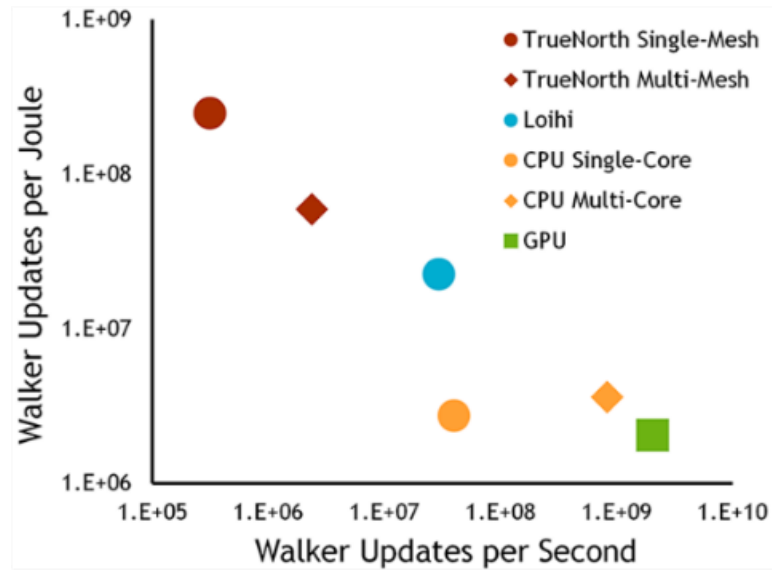
COINFLIPS

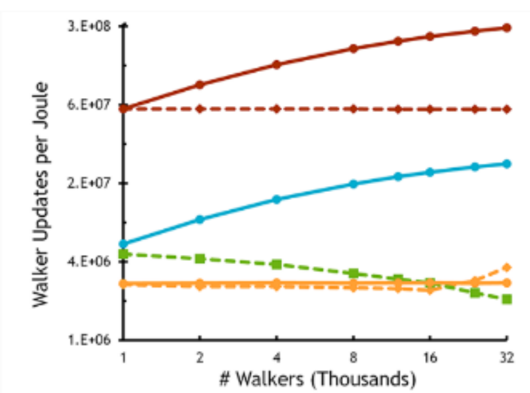


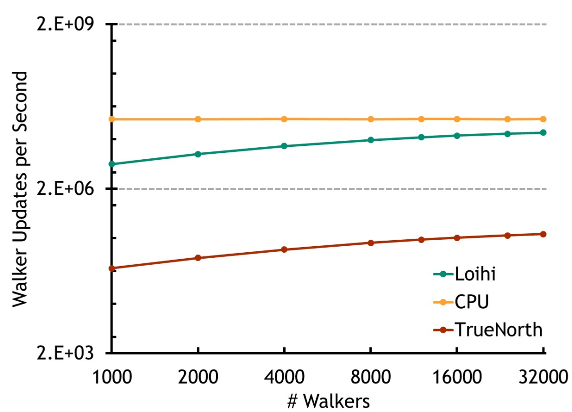
We can identify a neuromorphic advantage for simulating random walks

COINFLIPS

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).





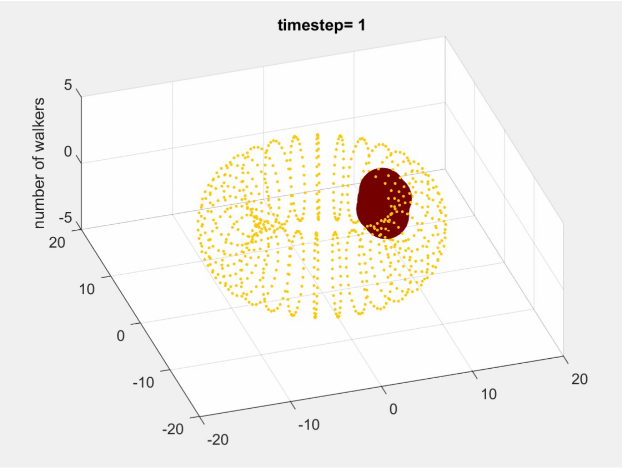


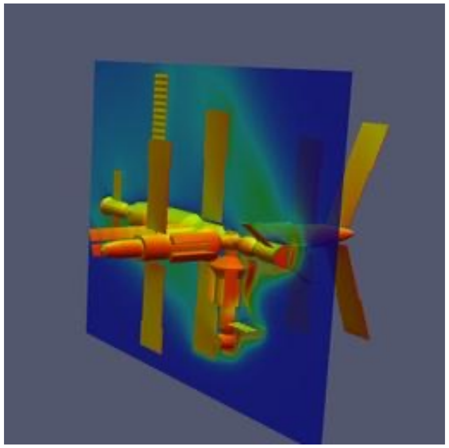
Smith et al., Nature Electronics 2022

Neuromorphic Mini-Apps to understand long-term value for HPC

Part 2

Will this translate to real world impact?





Random walks on neuromorphic (Smith et al., 2022)

- Brownian motion
- 1000's of particles
- 100's of cells
- 100's of timesteps
- 1 neuromorphic chip

SPARTA simulation of Mir space station (Michael Gallis, Sandia)

- Gas physics
- 1.6 Billion particles
- 10 million cells
- 500,000 timesteps
- 2048 Xeon cores

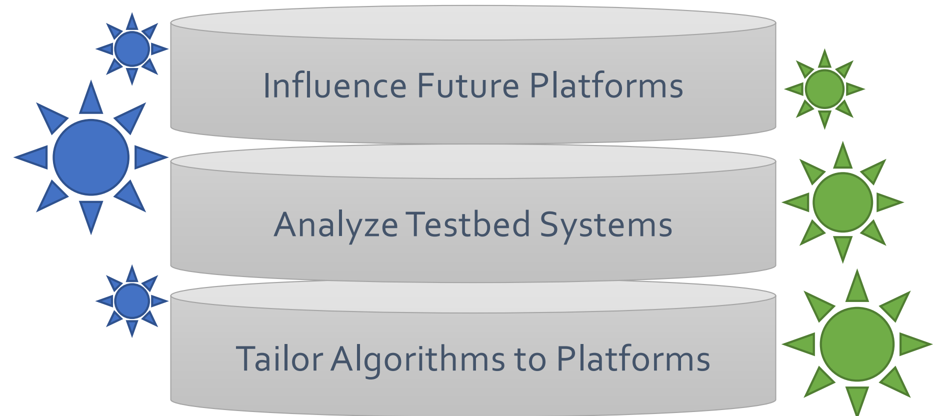
Why Mini-Apps for Neuromorphic?

- From Heroux et al., 2009: "there is a middle ground for small, self-contained programs that, like benchmarks, contain the performance-intensive computations of a large-scale application, but are large enough to also contain the context of those computations."

These Mini-Apps would enable:

- Interaction with external research communities
- Simulators
- Early node architecture studies
- Network scaling studies
- New language and programming models
- Compiler tuning

NMC systems have considerable uncertainty at algorithm level



Conventional systems have less uncertainty at algorithm and programming level

Conventional and Neuromorphic Mini-Apps

Conventional Mini-Apps

Mini-Apps

Neuromorphic Mini-Apps?

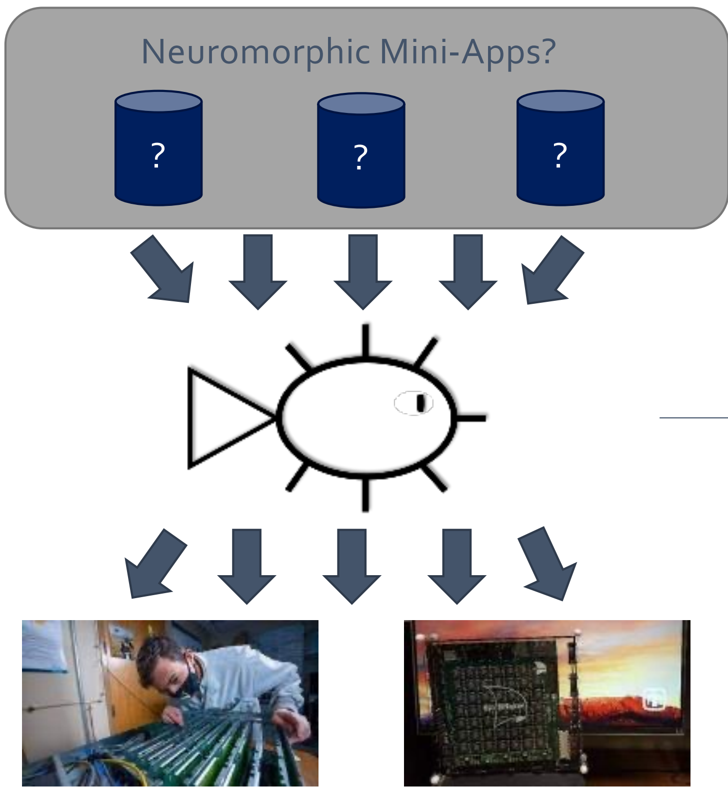
? ? ?

Platform-Agnostic Programming Layer

FUGU

Back-Ends

Fugu addresses two key challenges of neuromorphic programming



Composability

Deploying applications on neuromorphic hardware requires implementing algorithms within neural circuits

- Need to be able to build applications from well designed kernels
- Need to take advantage of features offered by spiking neuron model

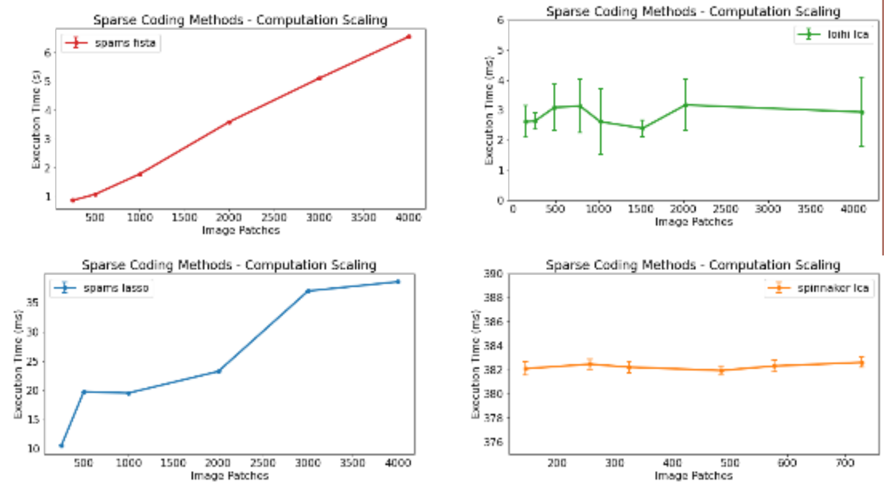
Portability

Programming neuromorphic platforms requires a *graph* of neurons (nodes) and synapses (edges)

- Need to represent neural algorithms in common graph format
- Need ability to translate graph into backend specific constraints

Neural Sparse Coding

Example Results -



Sparse Coding or Sparse Dictionary Learning

- Method of modeling data by decomposing it into sparse linear combinations of elements of a given overcomplete basis set
- On neuromorphic, the LASSO (least absolute shrinkage and selection operator) computation for sparse coding can be approximated with the spike-based algorithm LCA (locally competitive algorithm)
  - Implemented as rate-coded neurons with inhibitory connections between competing dictionary elements

Parameterization	Size of image, Size of image patch, Size of the dictionary, Stride of image patch, Desired sparsity
Scaling	Problem size via # of image patches, Parameters
Metrics	Time for setup, Time for reconstruction, Reconstruction performance, Reconstruction sparsity, Compute resource usage, Energy resource usage

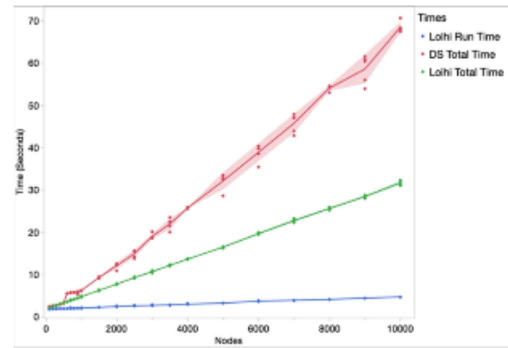
Neural Graph Analysis

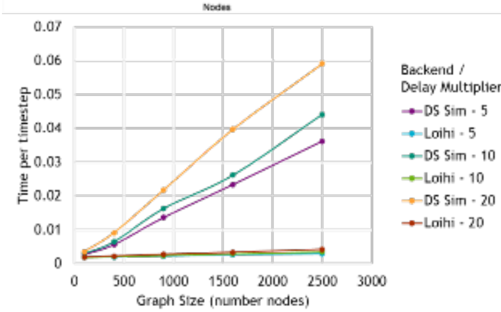
Example Results -

Single Source Shortest Path (SSSP)

- Between a source and target node, what is the shortest path (and path length) that connects the two
- SNN is straightforward – each vertex in the source graph is a neuron, each edge is a synapse between neurons, & graph weights equate to delays
  - The source neuron receives input driving it to spike send ensuing spikes through the SNN
  - Shortest path length is determined when the target spikes & monitoring edges can yield the path

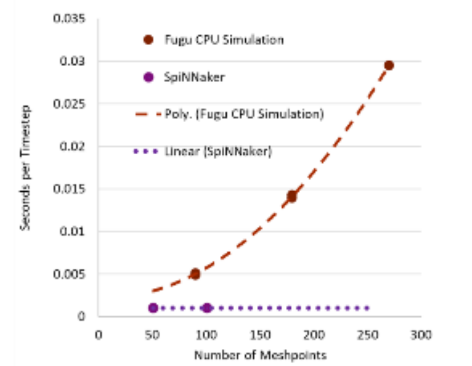
Parameterization	Graph generation (uniformly random tree, small world), Nodes, Weight range, Max runtime, Source, Target
Scaling	Graph scale, Weight/delay range
Metrics	Total time, Time for setup

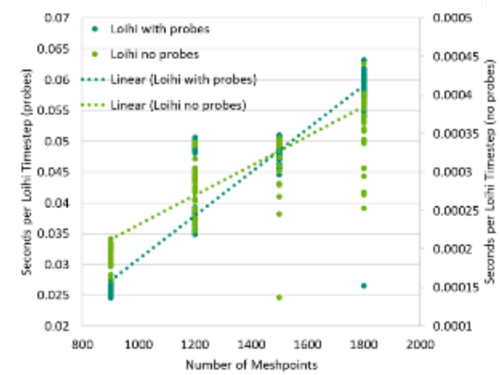




Neuromorphic Random Walk

Example Results -





Discrete time Markov Chain (DTMC)

- Particle Angular Fluence: the time-integrated flux of particles traveling through media given as a function of position and velocity
- Particles travel at a constant speed and experience relative velocity scattering over a small region of space
- Conventional approach models walkers & tracks states – neuromorphic models state & tracks walkers

Smith, J. Darby, et al. "Neuromorphic scaling advantages for energy-efficient random walk computations." Nature Electronics (2022): 1-11.

Parameterization	Number of total walkers, Size of direction/relative velocity/angular discretization, Time step size of simulation, Size of the state space, Size of positional discretization
Scaling	Walkers, Mesh size
Metrics	Energy cost of walkers, Time to run, Space to run

Where does this advantage come from?

- Extreme parallelism of neuromorphic hardware
  plus
  Embarrassingly parallel nature of Monte Carlo random walks

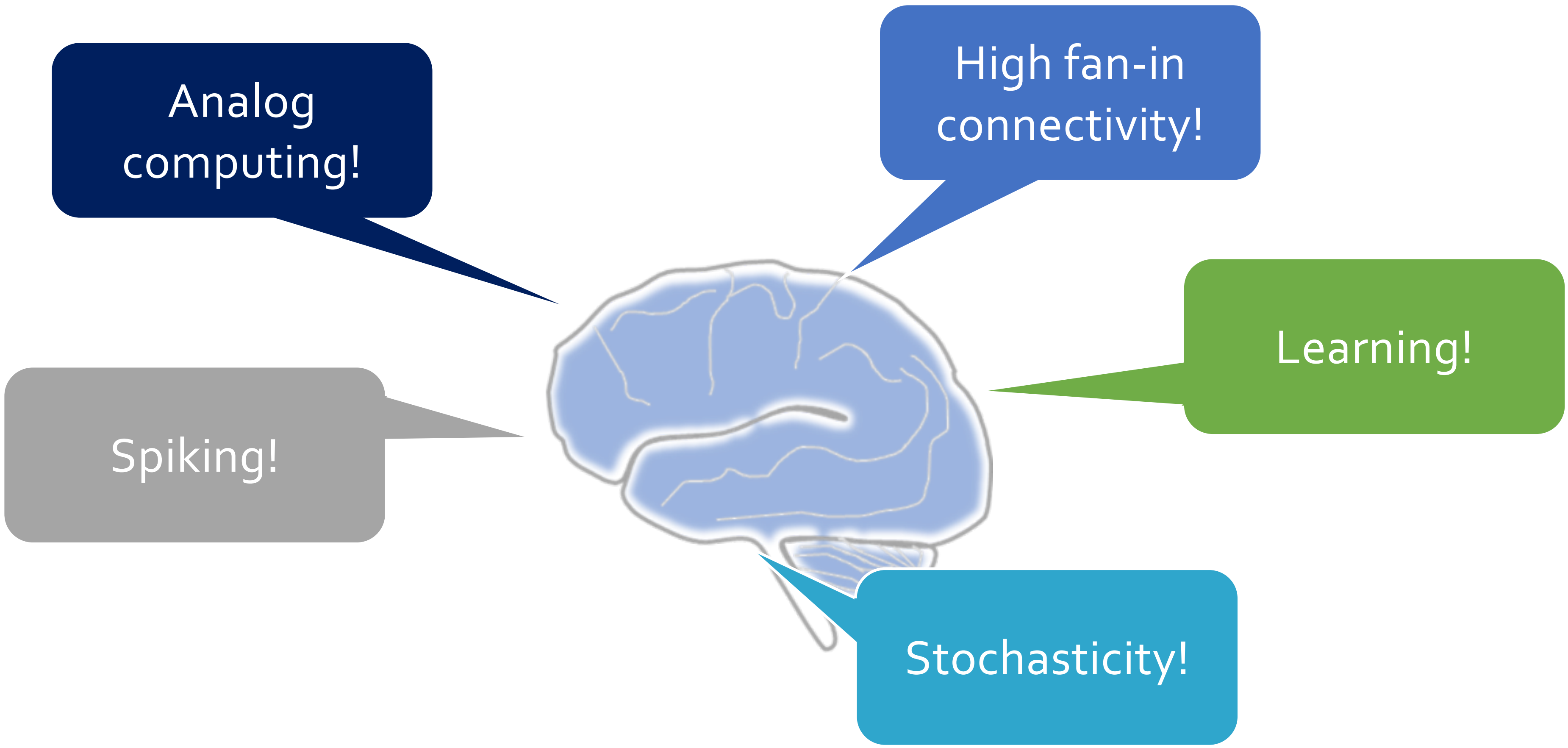
- Many simple calculations in parallel
  vs
  Single complex calculation

- Limiting factor going forward will likely be probabilistic component
  - Quality and form of random numbers
  - Quantity and location of random number generation

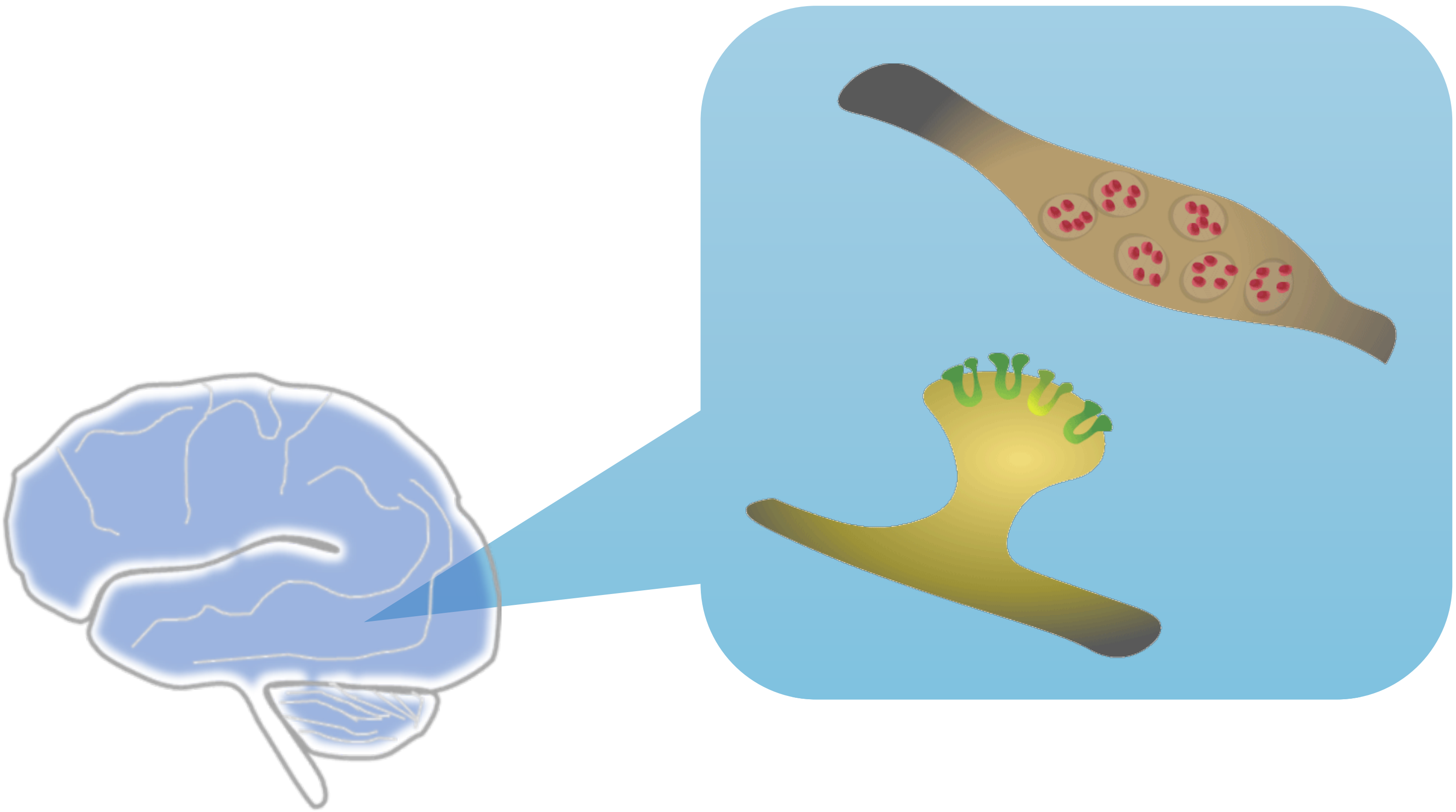
What happens if we build a neuromorphic chip centered on probabilistic sampling?

Part 3

What constitutes brain inspiration?



The brain’s trillions of synapses exhibit considerable stochasticity



The brain appears to use probabilistic sampling of populations

Neuron

Article

Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion

Highlights

- Hippocampal replay can represent Brownian diffusion-like random trajectories
- Reactivated trajectories cover positions over wide ranges of spatiotemporal scales
- Replay event statistics are incompatible with actual behavioral trajectories
- Expression dynamics of replayed assemblies was linked to specific oscillatory bands

Authors

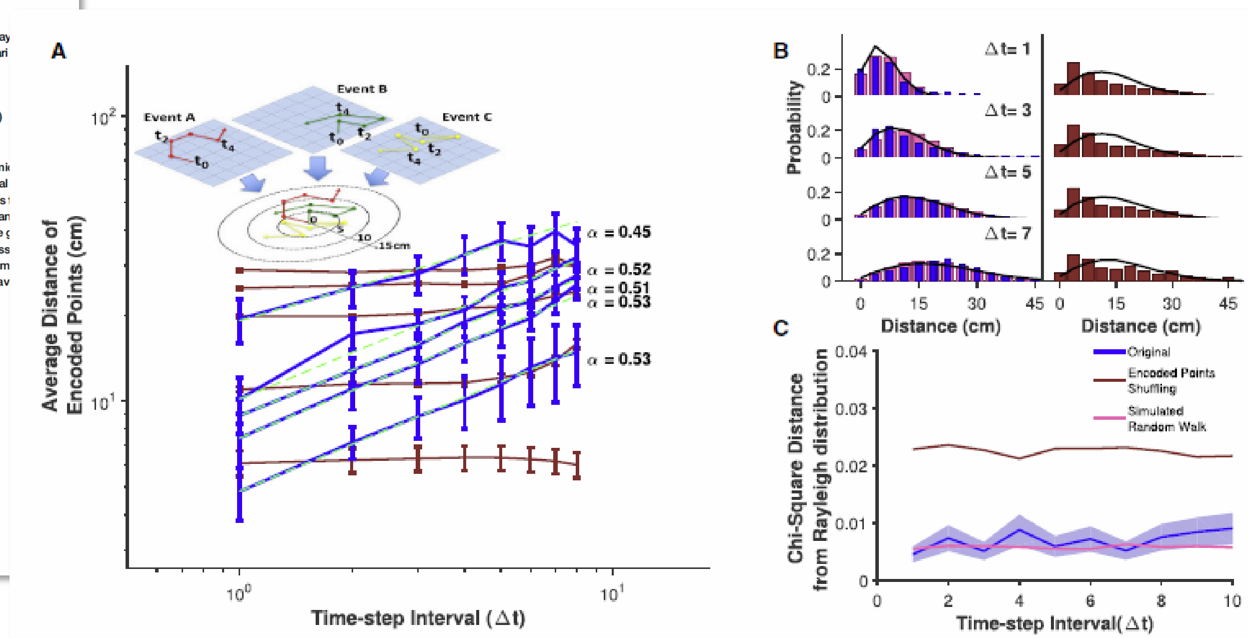
Federico Stella, Peter Baracskay
Joseph O'Neill, Jozsef Csicsvari

Correspondence

federico.stella@ist.ac.at (F.S.),
jozsef.csicsvari@ist.ac.at (J.C.)

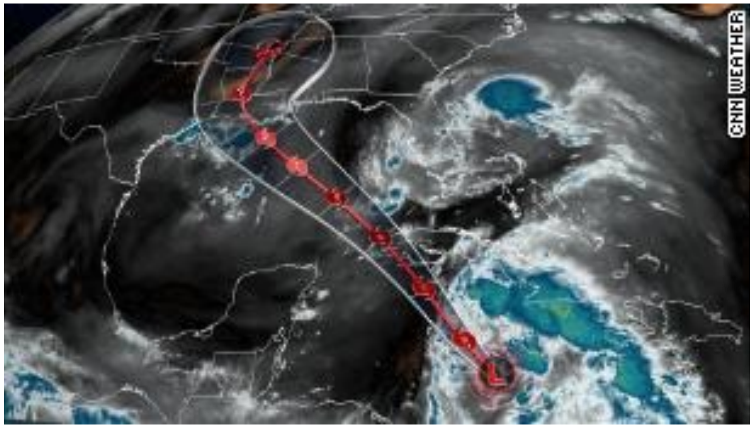
In Brief

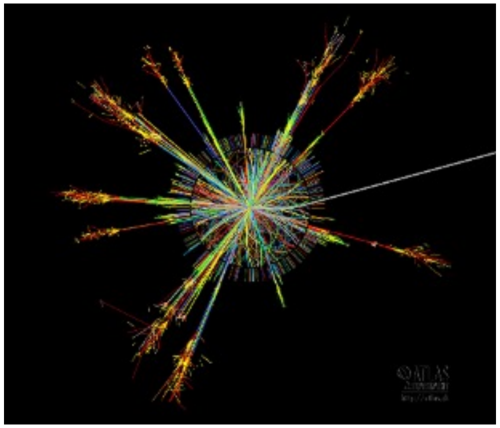
Stella et al. examine the dynami
properties of reactivated spatial
trajectories in the hippocampus
non-stereotypical exploration an
that reactivated trajectories are
by a Brownian diffusion process
occur at varying lengths and tim
without directly reflecting behav



Many applications of computing have inherent uncertainty

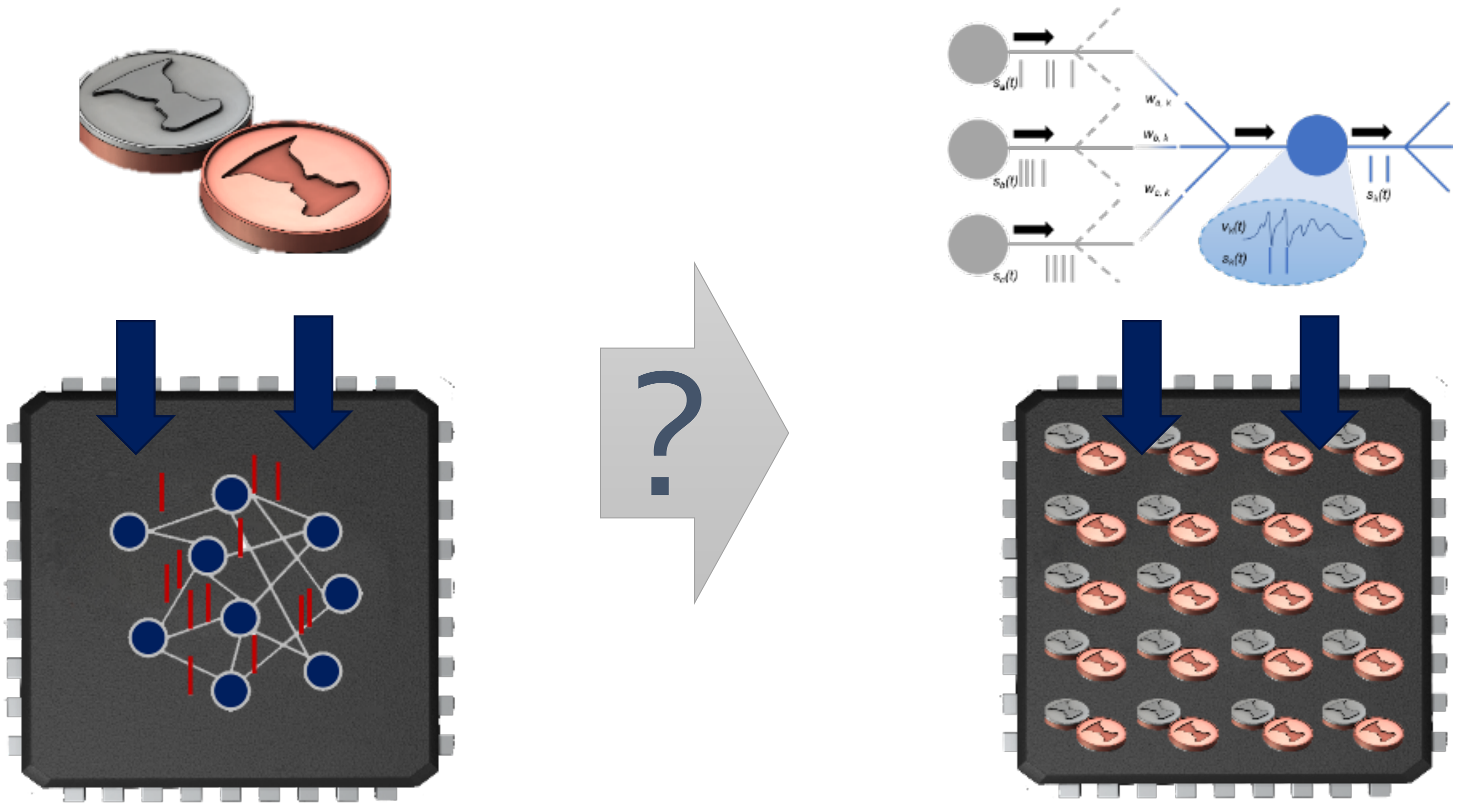
Many applications of computing have inherent uncertainty





Two main use cases:
- ❖ Mod-Sim --- Generating the random number *you need*
- ❖ Artificial Intelligence --- Effective and efficient sampling of algorithms

Making stochasticity ubiquitous may require that we revisit how we design neuromorphic hardware



COINFLIPS

ENERGY.GOV SCIENCE & INNOVATION ENERGY ECONOMY SECURITY & SAFETY SAVE ENERGY, SAVE MONEY

Department of Energy

DOE Announces $54 Million for Microelectronics Research to Power Next-Generation Technologies

MARCH 24, 2021

Energy.gov » DOE Announces $54 Million for Microelectronics Research to Power Next-Generation Technologies

National Labs Will Lead Transformation of Smart Devices, Clean Energy Technologies, and Semiconductor Manufacturing

WASHINGTON, D.C. — The U.S. Department of Energy (DOE) today announced up to $54 million in new funding for the agency's National Laboratories to advance basic research in microelectronics. Microelectronics are a fundamental building block of modern devices such as laptops, smartphones, and home appliances, and hold the potential to power innovative solutions to challenges like the climate crisis and national security. Watch this video to learn more about microelectronics.

"Thanks to microelectronics, transformational technologies that used to swallow up entire buildings now fit in the palms of our hands—and it's time to take this work to the next level," said **Secretary of Energy Jennifer M. Granholm**. "Microelectronics are the key to the technologies of tomorrow, and with DOE's world-class scientists leading the charge, they can help bring our clean energy future to life and put America a step ahead of our economic competitors."

Microelectronics were originally developed as a powerful capability for miniaturizing transistors and electronic circuits. Since then, they have fueled a digital revolution, making devices like computers and phones more powerful, compact, and convenient for everyday use.

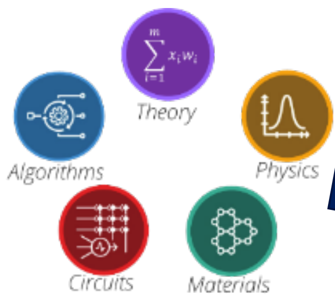
More microelectronics research is needed to pave the way for the next generation of revolutionary technologies. Potential applications include clean energy technologies that will help America combat the climate crisis, such as developments to make the nation's grid more efficient, more responsive to fluctuations in energy demand, and more resilient to extreme weather events.

New research could also help revive American production of semiconductors—critical computer

COINFLIPS

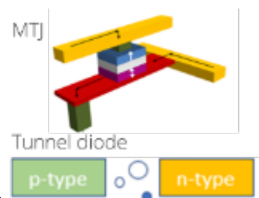
CO-designed Improved Neural Foundations Leveraging Inherent Physics Stochasticity (COINFLIPS)

COINFLIPS

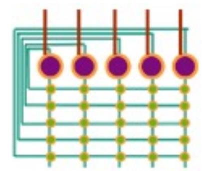


Approach

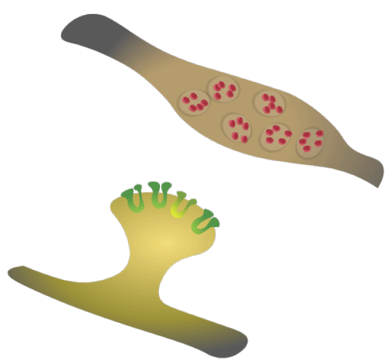
Co-design approach on new energy-efficient microelectronics



Tunable Stochastic Devices

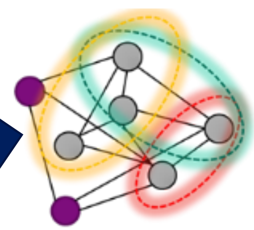


Probabilistic Circuits and Architectures

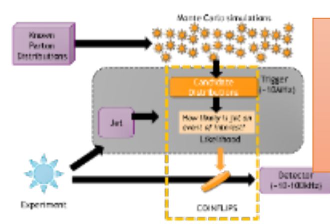


Every synapse in the brain is a stochastic "coinflip"

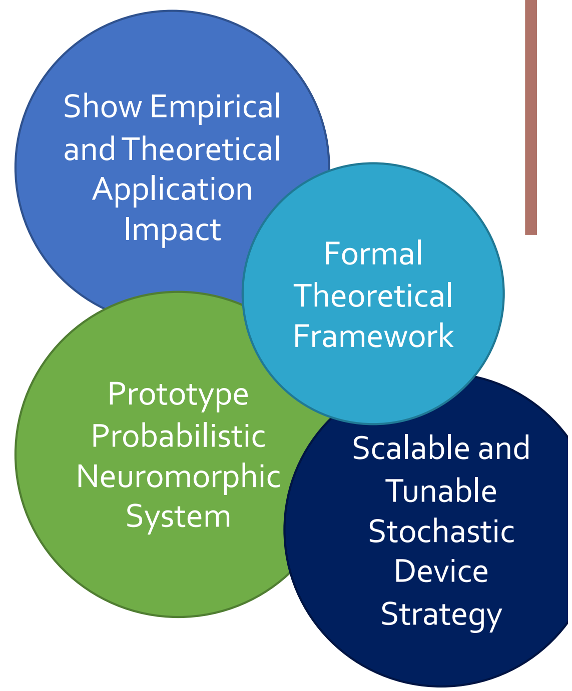
Inspiration



Probabilistic Neural Theory and Algorithms



Particle Physics Demonstration



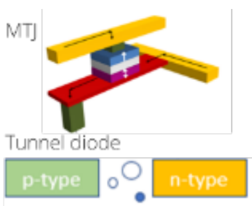
COINFLIPS devices

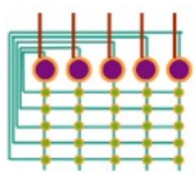
Tunable Stochastic Devices

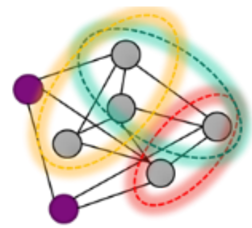
Probabilistic Circuits and Architectures

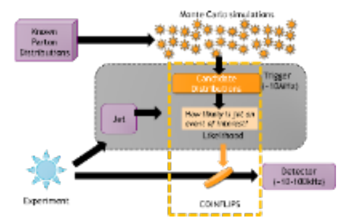
Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration





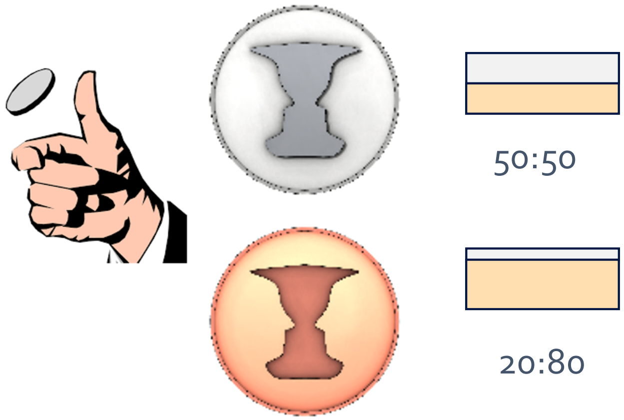




Tunable RNG – magnetic tunnel junctions & tunnel diodes

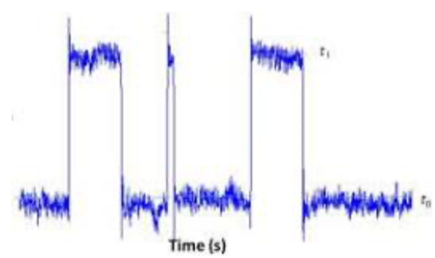
COINFLIPS

Tunable random number generator

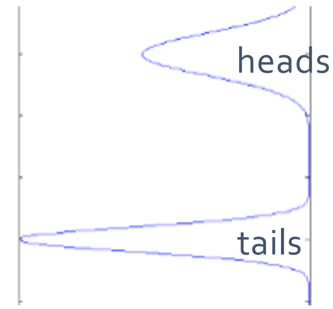


Why did we pick the devices we picked?

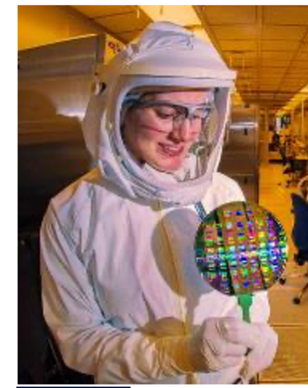
Large signals



Tunable



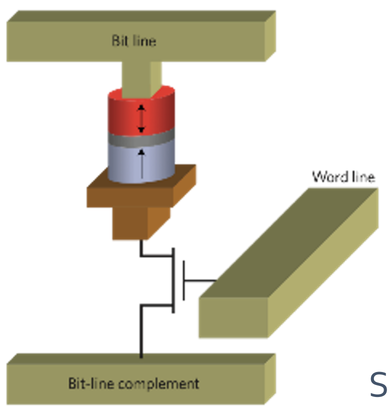
Integration



I. Magnetic tunnel junctions

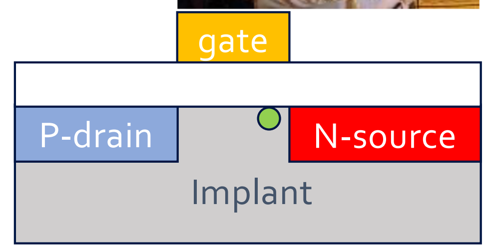
Jean Anne Incorvia

Andy Kent



II. Tunnel diodes

Shashank Misra & Tzu-Ming Lu



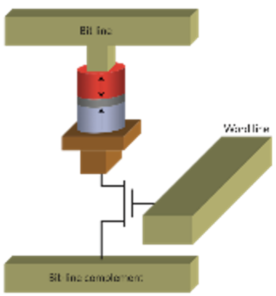
COINFLIPS motivating application

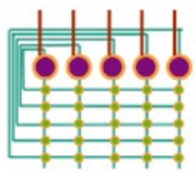
Tunable Stochastic Devices

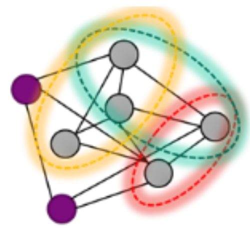
Probabilistic Circuits and Architectures

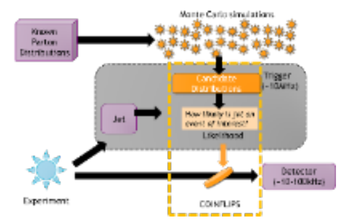
Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration



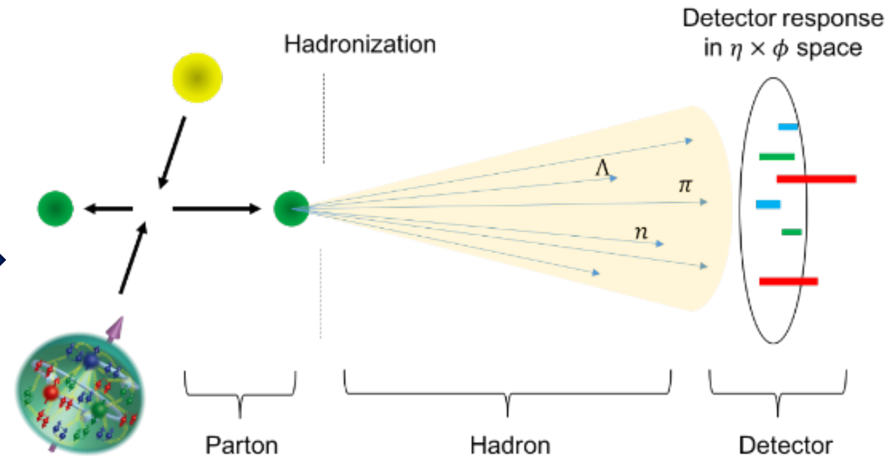




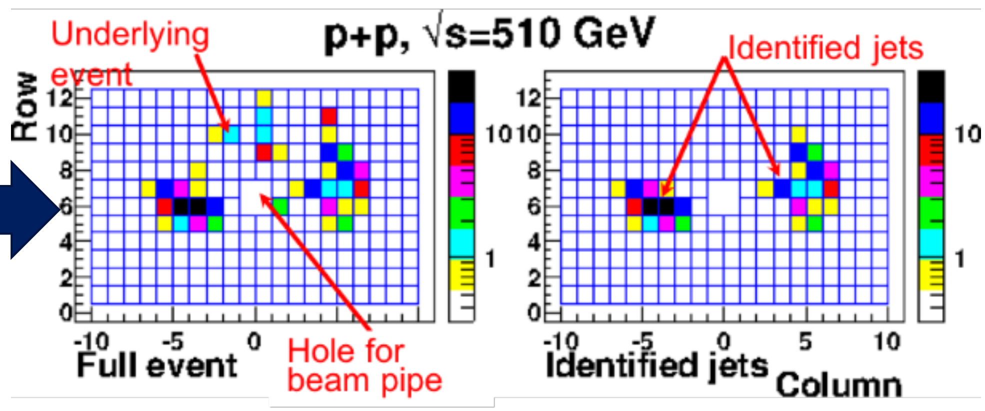


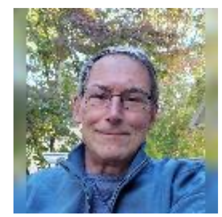
Jet detection in particle physics

Improved Monte Carlo modeling of particle physics



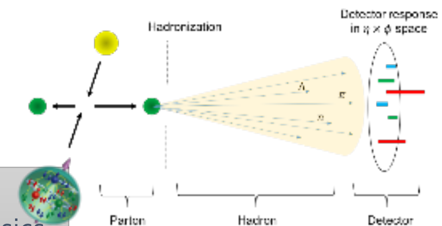
Bayesian Neural Networks to process sensor readings in real-time





Les Bland, Bernd Surrow, Jae Nam

COINFLIPS algorithms – random number generation

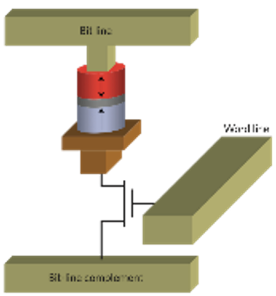


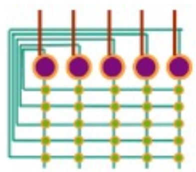
Tunable Stochastic Devices

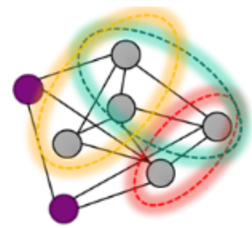
Probabilistic Circuits and Architectures

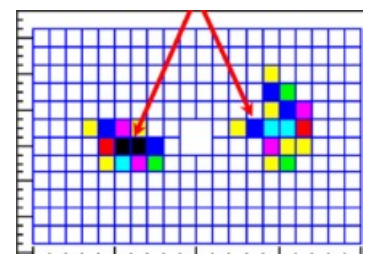
Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

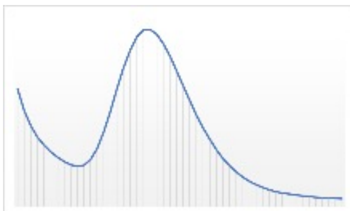








Random numbers are a non-trivial computational cost today



Draw uniform RNG

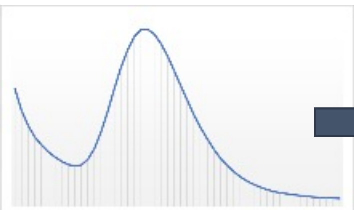
Convert to desired PDF

We want a RN pulled from some physics distribution

Software uses pseudo-RNG to pull uniform random number
- This is simple, but can be costly for volume and quality

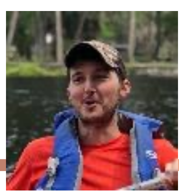
Numerical methods convert uniform RN to desired distribution
- Some distributions are easy (simple inverse CDF)
- Some distributions are challenging

It is possible to generate a random number from a desired statistical distribution



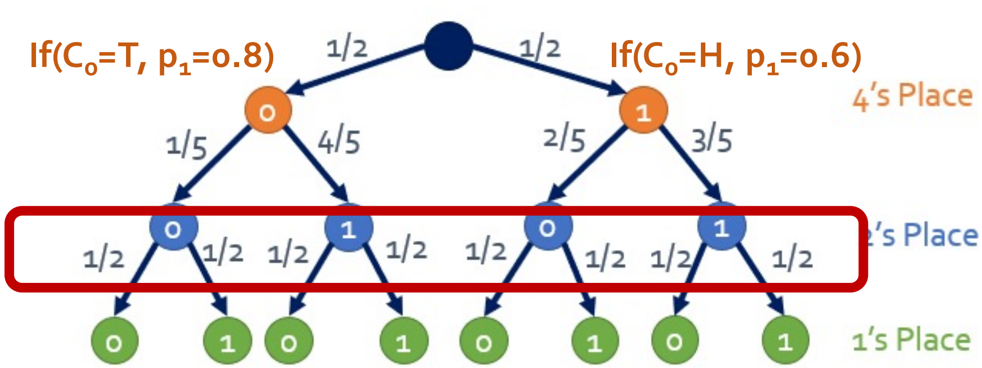
Draw uniform RNG

Convert to desired PDF



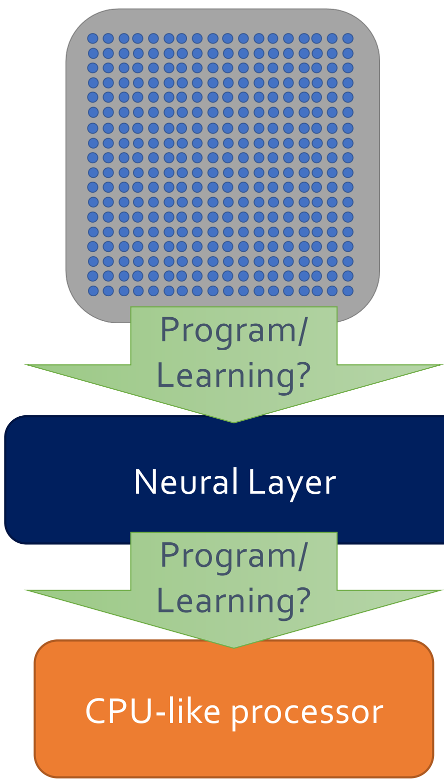
Darby Smith

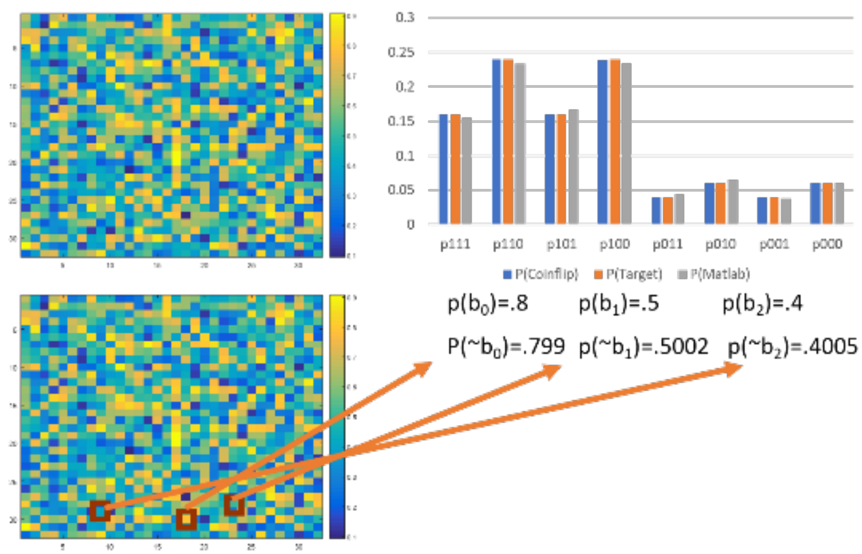
Expand Boolean tree of PDF and flip many coins for all branches in parallel



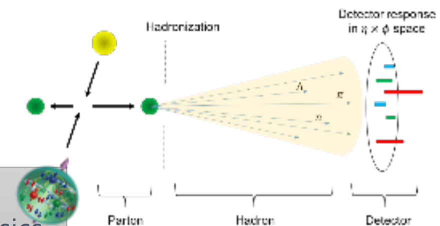
- Worst case, this is a exponentially large number of coins
- PDFs have structure and redundancies that can be leveraged
- Correlations from devices or built into neural circuits can similarly compress tree

A potential COINFLIPS architecture for generating random numbers





COINFLIPS algorithms – artificial intelligence

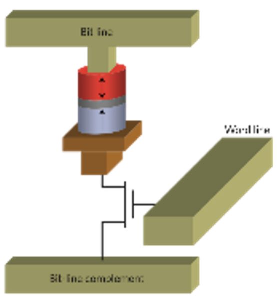


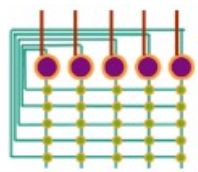
Tunable Stochastic Devices

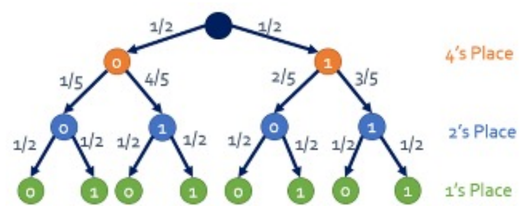
Probabilistic Circuits and Architectures

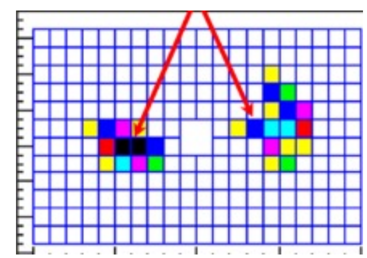
Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration



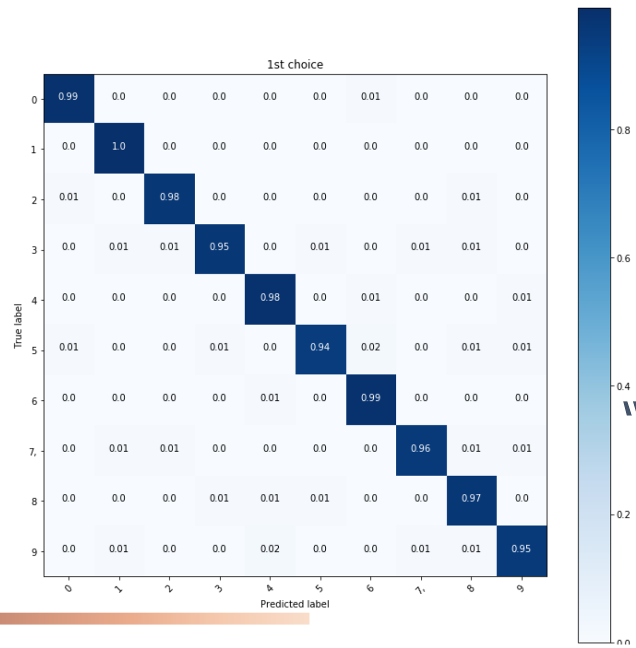


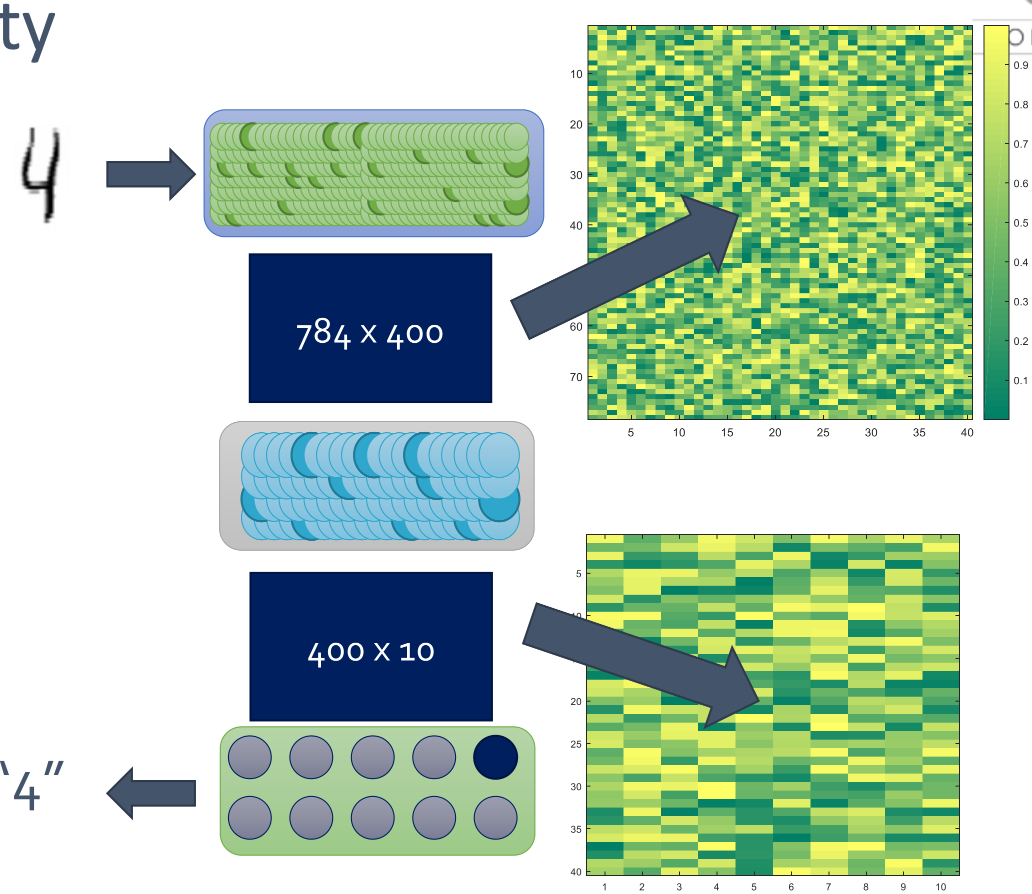




Sampling ANNs with stochastic synapses provides estimate of uncertainty

- Approach
  - Train simple neural network with only minor modifications
  - Simple network can achieve decent performance

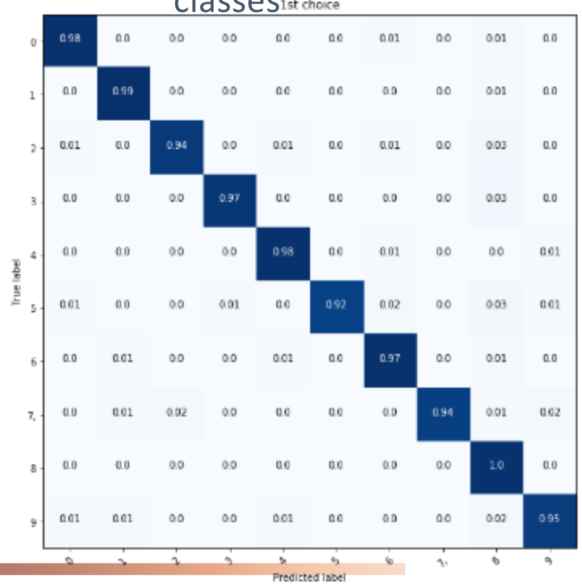


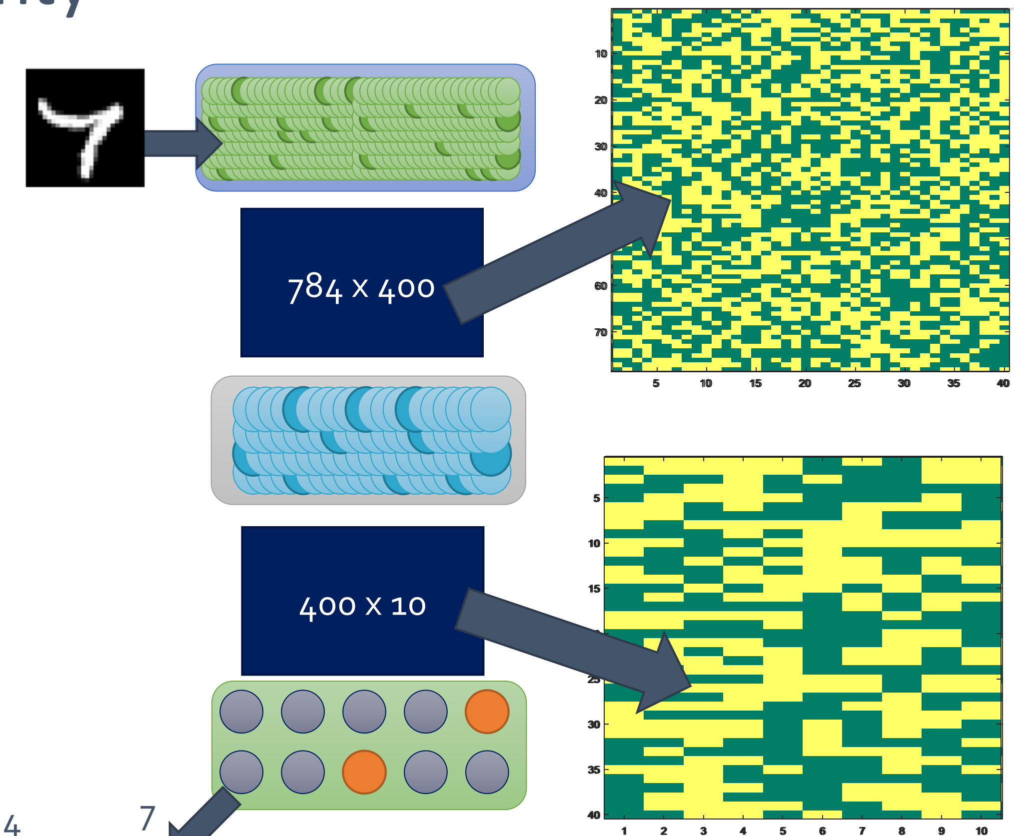


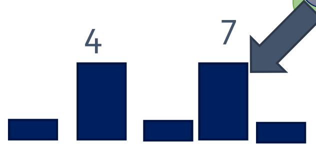
Weights continuous between 0 and 1

Sampling ANNs with stochastic synapses provides estimate of uncertainty

- Approach
  - Train simple neural network with only minor modifications
  - Convert weights to Bernoulli probabilities (weighted coinflips)
  - Sample network to identify what classes





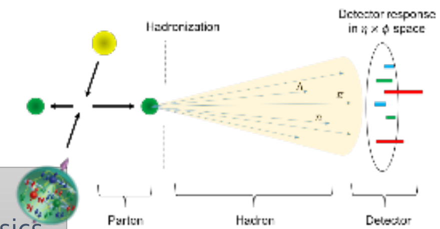


Weights *sampled* as probability

2nd choice of stochastic sampled networks is often the ‘right’ answer for misclassified results

6 – 0.38	9 – 0.31	4 – 0.36	9 – 0.26	3 – 0.23	6 – 0.26	0 – 0.39
5 – 0.17	4 – 0.28	7 – 0.35	2 – 0.20	9 – 0.20	2 – 0.25	6 – 0.27

COINFLIPS circuit design

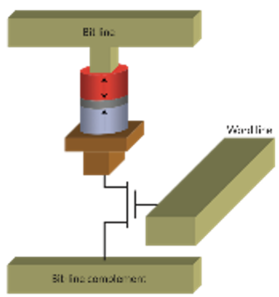


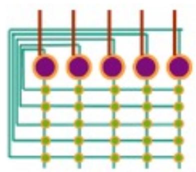
Tunable Stochastic Devices

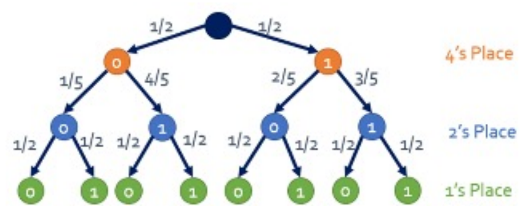
Probabilistic Circuits and Architectures

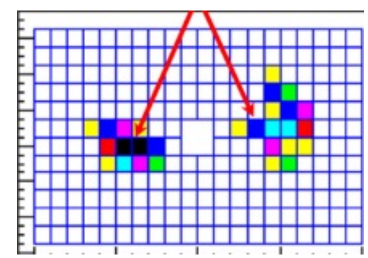
Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration



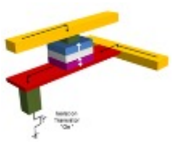




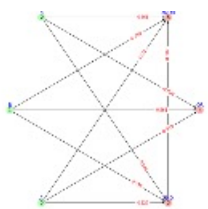


AI-Enhanced Co-Design across Scales

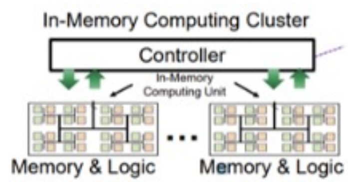
Device Design



Circuit Design



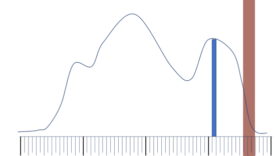
System Design

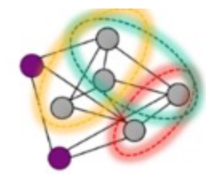


Fan (UCF), 2018

Architecture Design

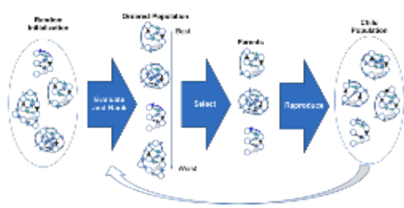
Algorithm Design



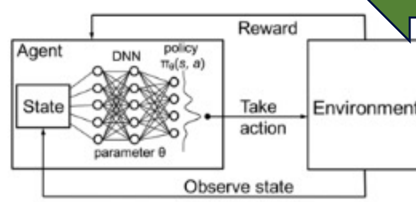


Approach

Can we leverage AI to generate specifications for novel devices?



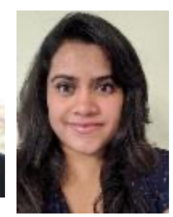
Evolutionary/RL approaches



RL approaches

Analytical and cycle-accurate tools, network simulation tools

Katie Schuman (Tenn)
Suma Cardwell (Sandia)



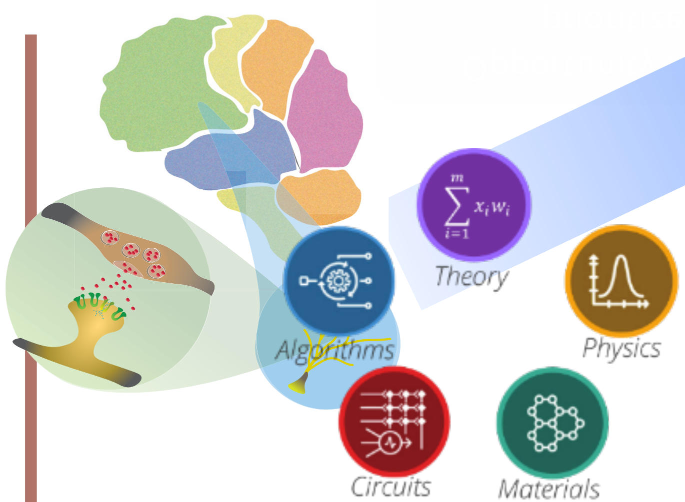
COINFLIPS presents an opportunity to develop a *community of interest* to create a new computing paradigm

Jointly develop a programming model and theoretical framework with an emerging technology

Opportunity for computing to prioritize impact on different classes of applications

Factor in integration and system design from the onset of a new approach

Optimize non-CMOS devices for scalability and cost of reliability



Thanks!

COINFLIPS Team

Sandia: Shashank Misra, Suma Cardwell, Darby Smith, Conrad James, Brad Theilman, William Severa, Ojas Parekh, Yipu Wang, Cale Crowder, Tzu-Ming Lu, Chris Allemang, Xujiao Gao, Juan Pedro Mendez, Scott Schmucker, Deanna Lopez

Tennessee: Katie Schuman

Temple: Les Bland, Bernd Surrow, Jae Nam

Texas: Jean Anne Incorvia, Jaesuk Kwon, Samuel Liu

NYU: Andy Kent, Laura Rehm

DOE Office of Science: ASCR (Robinson Pino PM), BES, HEP, NP, FES

U.S. DEPARTMENT OF ENERGY

Office of Science

Thanks!

Neural Random Walks Team

Darby Smith, William Severa, Rich Lehoucq,
Brian Franke, Leah Reeder, Aaron Hill, Ojas Parekh

Sandia National Laboratories Laboratory Directed Research and Development
DOE NNSA: Advanced Simulation and Computing

Neural Mini-Apps Team

Craig Vineyard, Suma Cardwell, Frances Chance, Srideep Musuvathy, Fred Rothganger, William Severa, Darby Smith, Corinne Teeter, Craig Vineyard, Felix Wang

DOE NNSA: Advanced Simulation and Computing