SAND2022-12773C

A Probabilistic Future for Neuromorphic Computing

Brad Aimone

Distinguished Member of Technical Staff
Sandia National Laboratories

jbaimon@sandia.gov

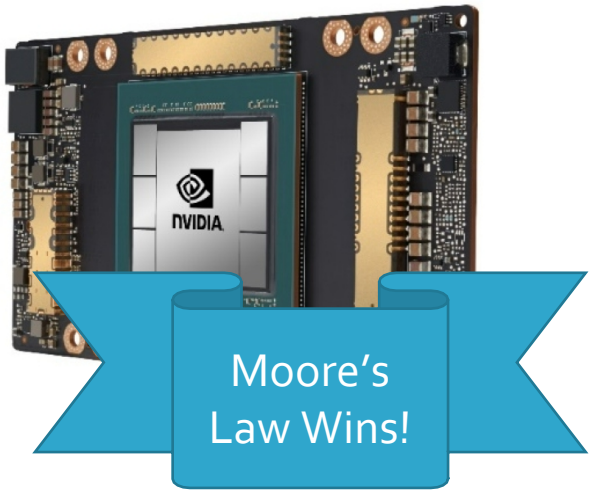
9/21/2022

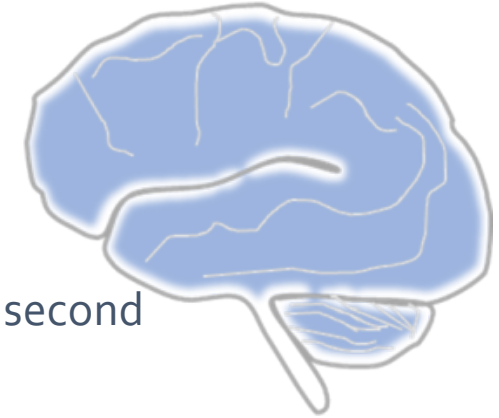
Has the tremendous success of deterministic computing left probabilistic applications behind?

COINFLIPS

Which approach is best to interpret a clear input?

~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic





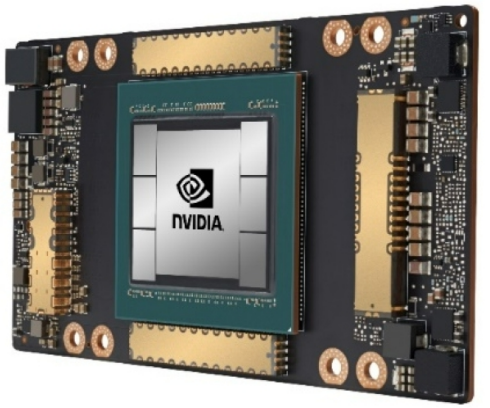
~20 W
~10^{15} synaptic events / second
Fully stochastic

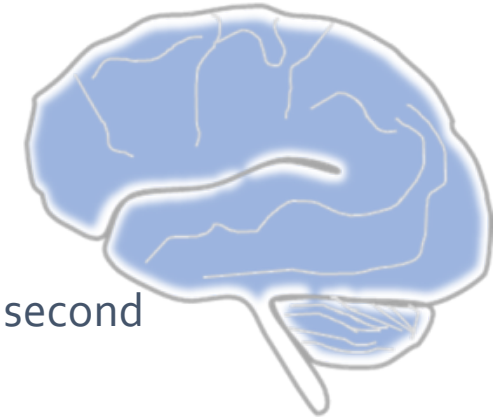
Has the tremendous success of deterministic computing left probabilistic applications behind?

COINFLIPS

Which approach is best to interpret an ambiguous input?

~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic





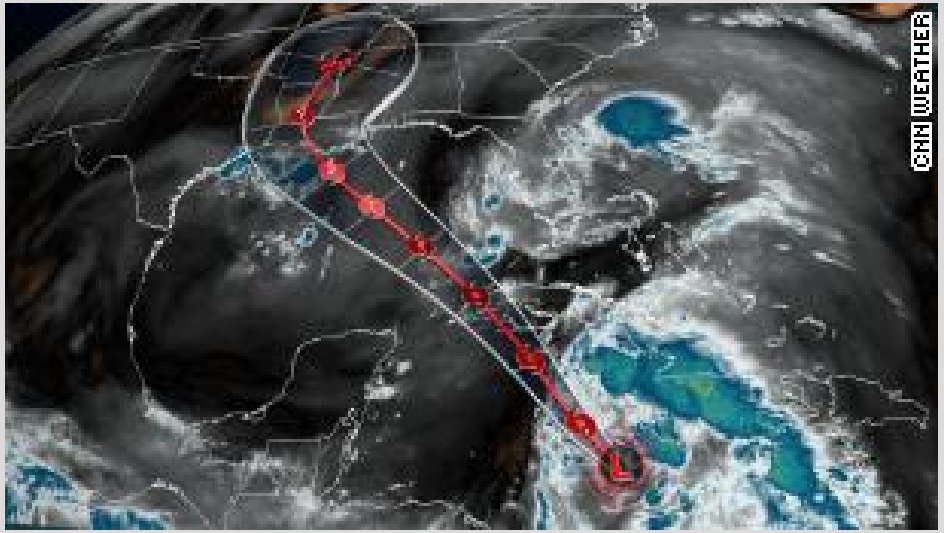
~20 W
~10^{15} synaptic events / second
Fully stochastic

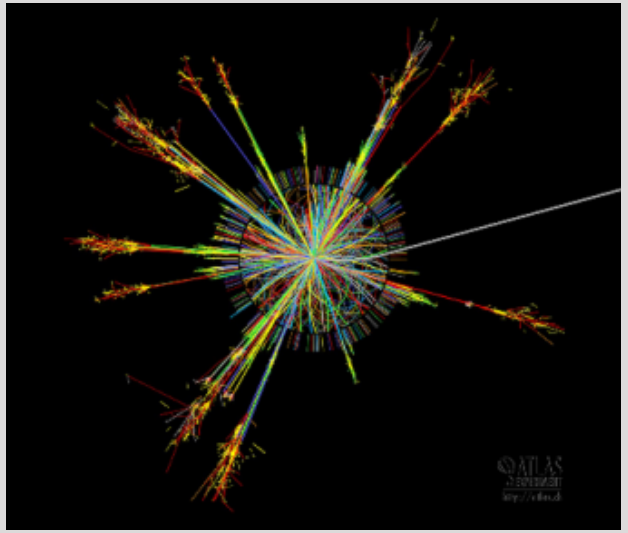
Computing applications face challenges in uncertainty

COINFLIPS

Artificial Intelligence

- Bayesian neural networks are appealing yet often computationally intractable





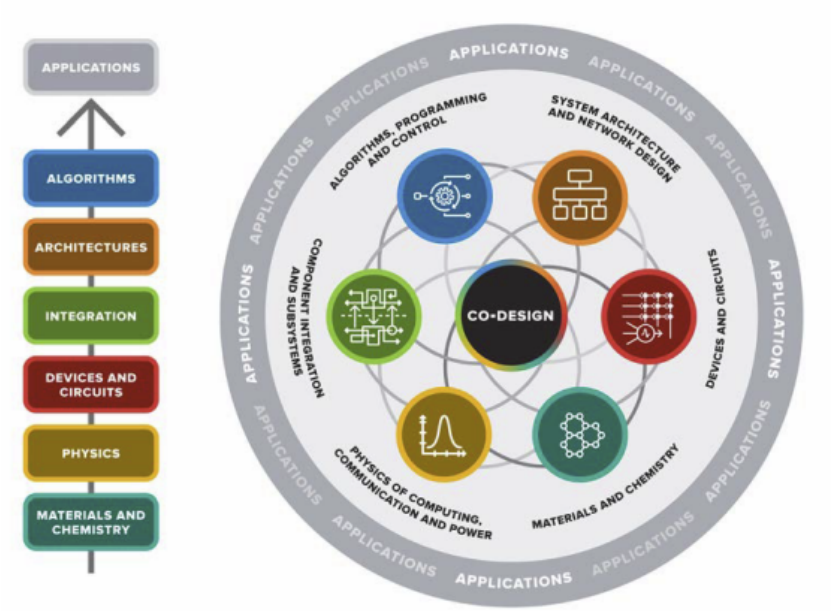
Modeling and Simulation

- Modeling uncertainties is critical in the use of even fully deterministic simulations
- Many applications are inherently stochastic in their physics and are best modeled using probabilistic methods

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

COINFLIPS

- Office of Science Co-Design in Microelectronics program
  - Co-funded through ASCR and BES, participation by NP, HEP, and FES



To enable new generations of energy-efficient computing systems over the next decade, ***a complete reconceptualization of the science and technology underlying the microelectronics co-design approach is needed*** *to integrate emerging devices, materials, interconnects, and non-linear phenomena with the needs of scientific computing applications.*

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories

NYU

The University of Texas at Austin

UT

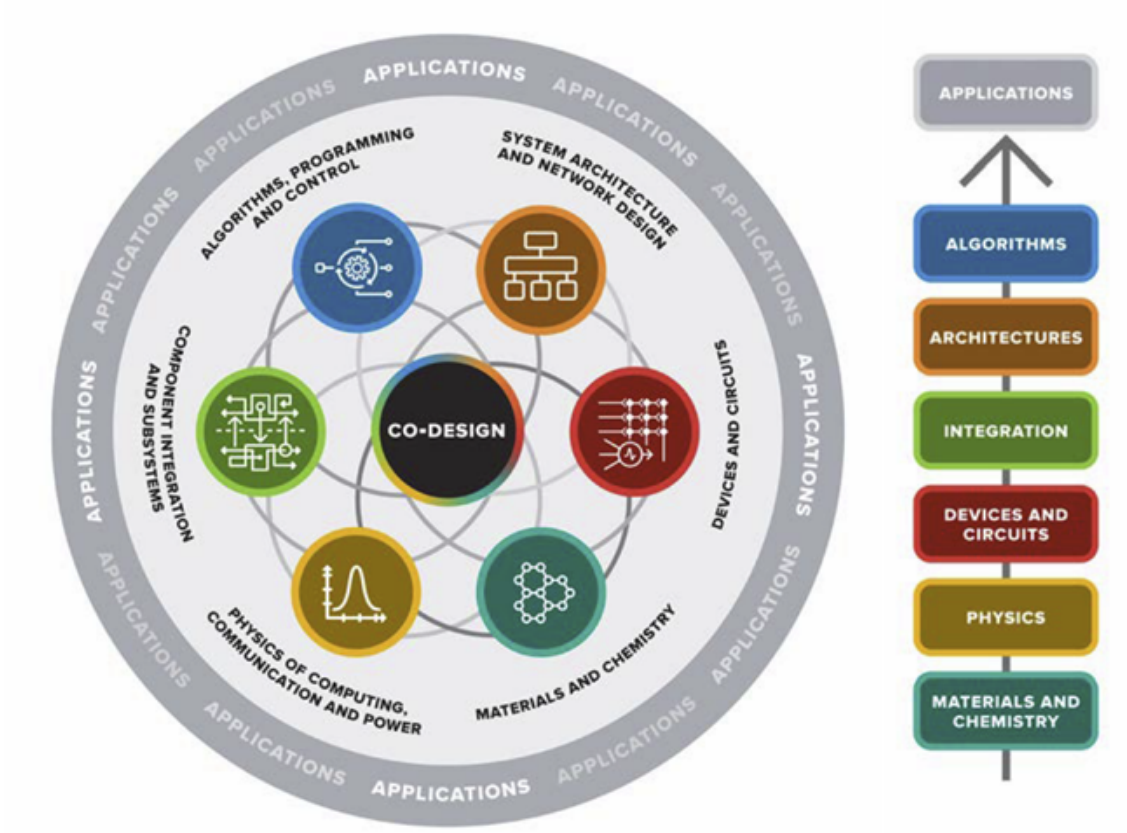
OAK RIDGE National Laboratory

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

- Office of Science Co-Design in Microelectronics program
  - Co-funded through ASCR and BES, participation by NP, HEP, and FES
- ~COINFLIPS is partnering with a growing number of organizations
  - Andy Kent @ New York University
  - Jean Anne Incorvia @ University of Texas Austin
  - Katie Schuman @ University of Tennessee
  - Prasanna Date @ Oak Ridge National Laboratory
  - Les Bland @ Temple University

Postdoc opportunities available!

COINFLIPS Today and in the Future



Today		*Future*
Random Number Generation Graph Analytics	→	*Bayesian Neural Networks* *Neuroscience-inspired Algorithms*
Hand-tuned Neural Circuits Evolution-optimized p-bits	→	*AI-Optimized Circuits* *In situ learning*
Magnetic Tunnel Junctions Tunnel Diodes	→	*Memristors* *…*

COINFLIPS

We are benefitting from 70 years of microelectronics that embrace *deterministic* components to solve *deterministic* problems

COINFLIPS sees an opportunity to embrace *stochastic* computing to solve *uncertainty* problems

Today's computers emulate uncertainty by using pseudo-random number generation

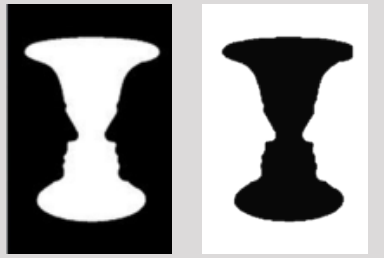
"Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin."

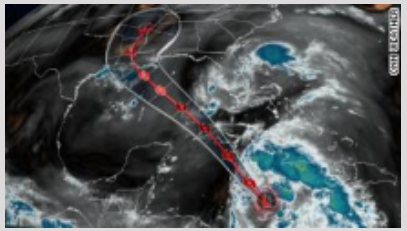
John von Neumann, 1951

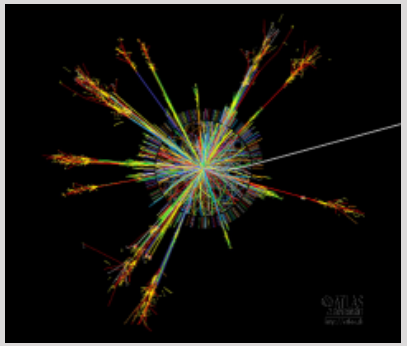
70 years later…

- Pseudo-RNGs can be quite effective, and do offer some advantages in verification, etc.
- But they are expensive, and when they go wrong the implications can be disastrous

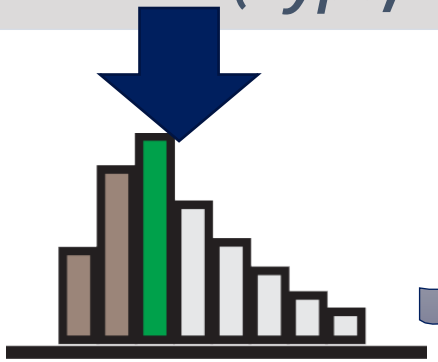
COINFLIPS aims to integrate true random number generators using stochastic devices into neuromorphic architectures





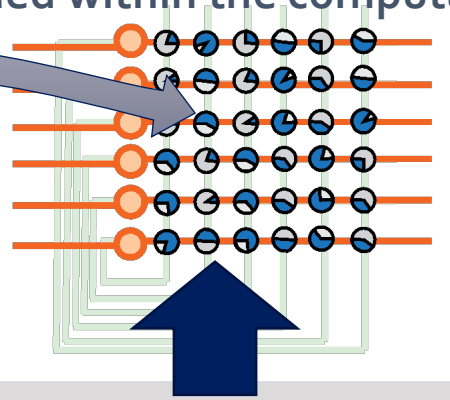


Improved Random Number Generation
(Type, Quantity, Quality)



Sample a random number from the *exact* distribution we require

And sample that number *where* it is needed within the computation



Architecture that integrates ubiquitous stochastic devices with computing and memory

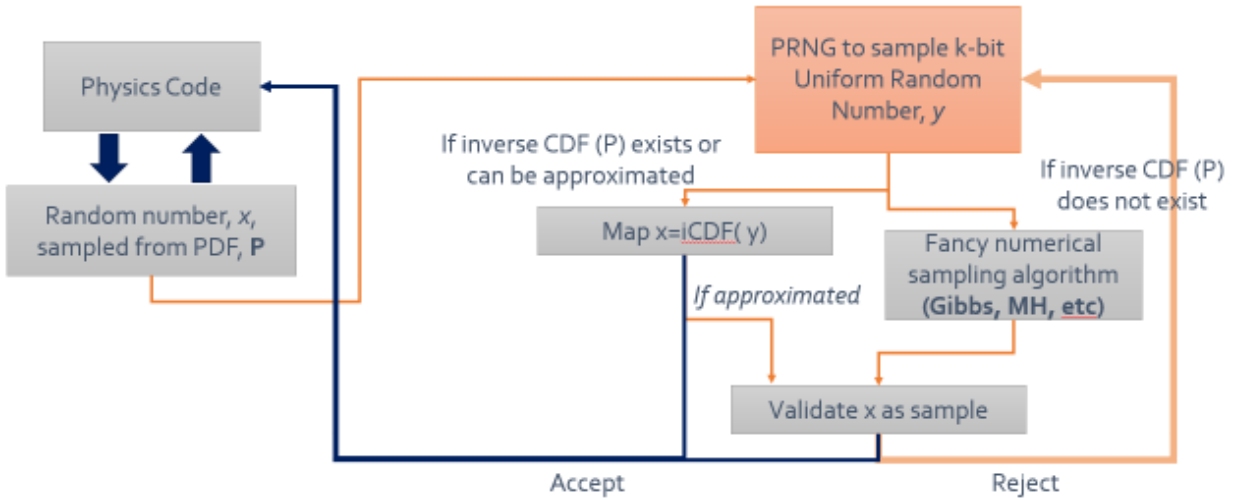
COINFLIPS aims to improve both speed and energy of probabilistic computing applications

Evaluate opportunity of a probabilistic computing paradigm

COINFLIPS

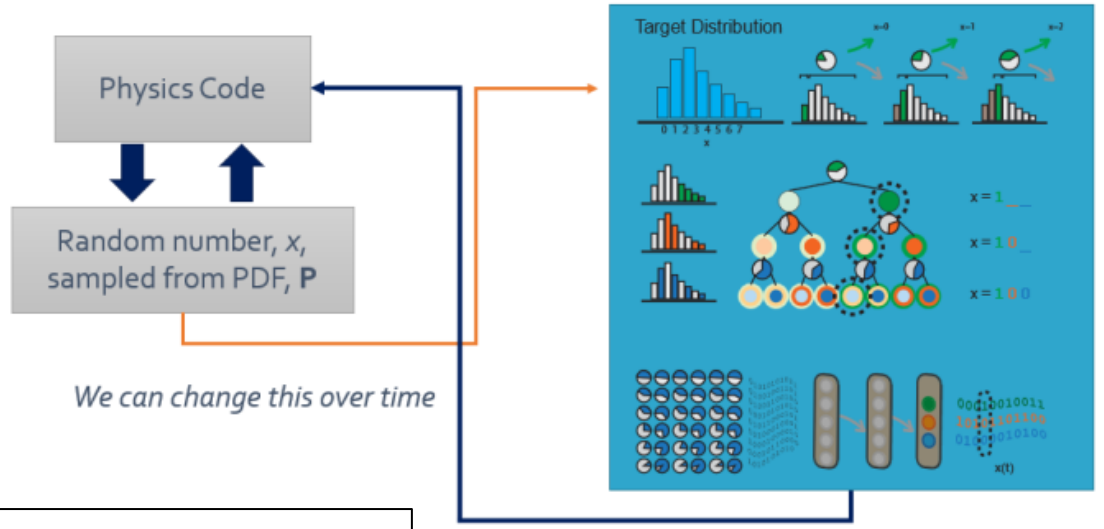
Today

Calculations in model start with uniform random number 0-1



Future?

System dynamically retunes itself to represent the model



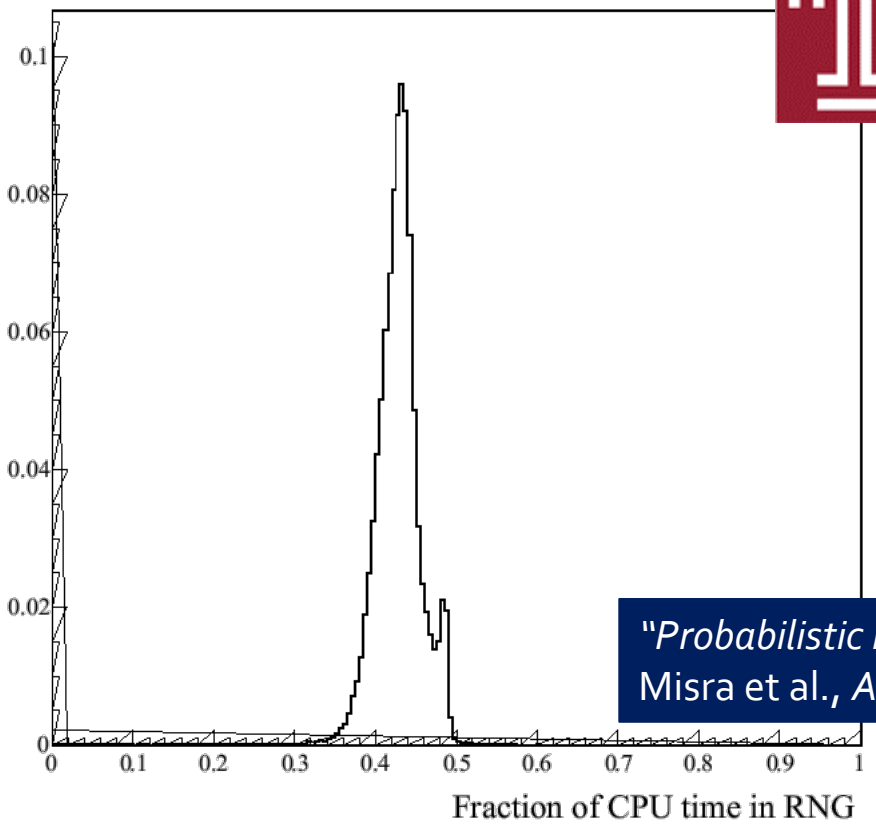
Brain generates 10^{15} stochastic events / sec
Software (Shishua): 200 kW
Hardware (Ring osc): ~ 8 inch wafer

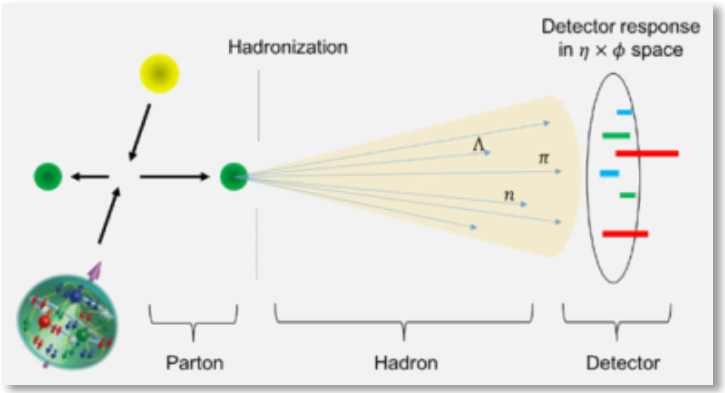
Catch-22: not yet possible to generate 10^{15} random numbers per second, so nobody develops algorithms that can use RNs at that rate

Random numbers are a limiting computational cost for some nuclear physics applications

COINFLIPS

- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration





"Probabilistic neural computing with stochastic devices"
Misra et al., *Advanced Materials*. In Press

Half of computational cost is generating a uniform random number, which then must be transformed

Sampling a uniform distribution (generate random number 0-1)

COINFLIPS

Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

Pseudo-random number generator

$$x \cdot y = \{\text{if } x \geqslant y \text{ then } x - y, \text{ else } x - y + 1\},$$
$$c \circ d = \{\text{if } c \geqslant d \text{ then } c - d,$$
$$\text{else } c - d + 16777213/16777216\}.$$

We require computer instructions that will generate two sequences:

$$x_1, x_2, x_3, \ldots, x_{97}, x_{98}, \ldots,$$
$$\text{with } x_n = x_{n-97} \cdot x_{n-33},$$
$$c_1, c_2, c_3, \ldots,$$
$$\text{with } c_n = c_{n-1} \circ (7654321/16777216).$$

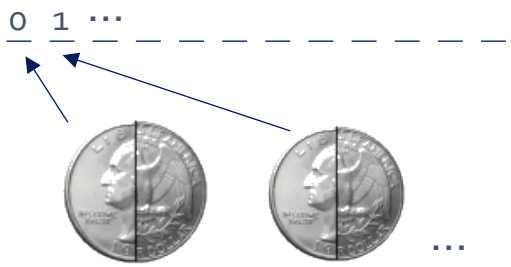
Then produce the combined sequence

$$U_1, U_2, U_3, \ldots, \quad \text{with } U_n = x_n \cdot c_n.$$

Cost:
- Store 100 numbers
- 3 comparisons
- 3 subtractions

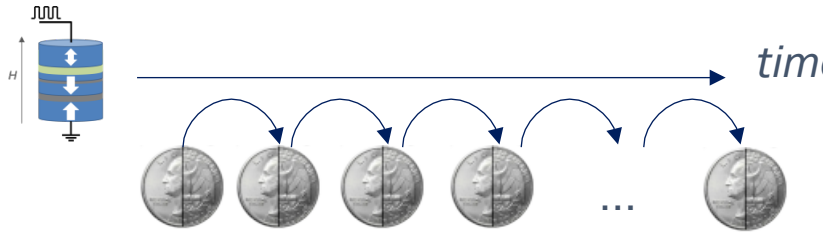
Hardware number generator

k-bit string → single precision number 0-1

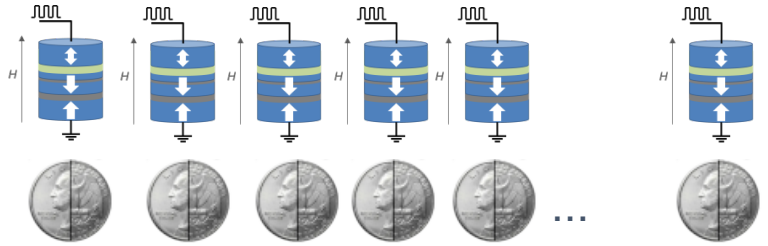


Cost:
- 1 device operation per bit

Two options for true RNG



One device flipping over time



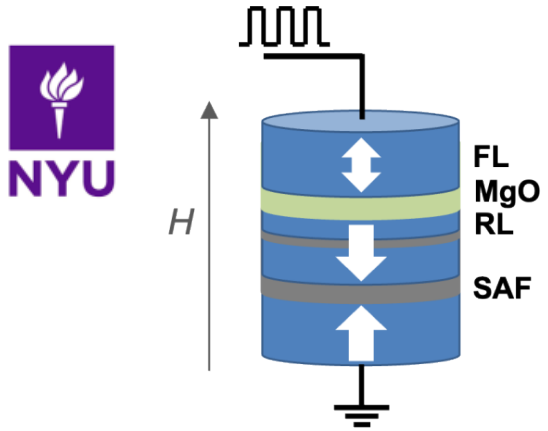
Many devices flipping one time

Fair coinflip device example – Magnetic Tunnel Junction (MTJ)

COINFLIPS

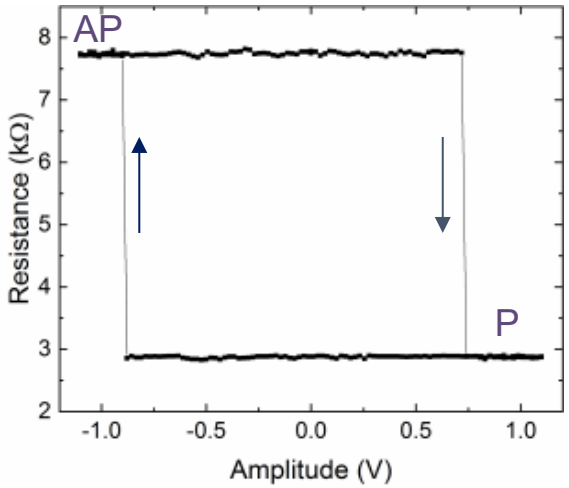
- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration

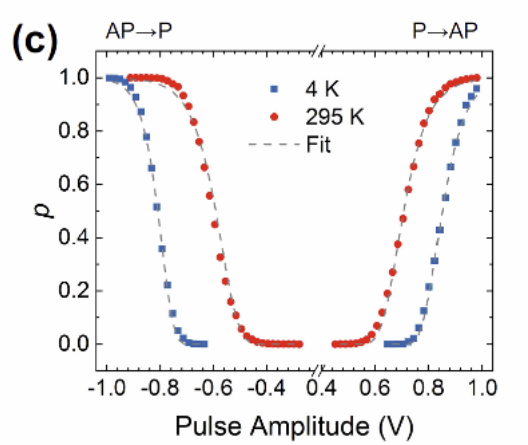
MTJ Coinflip device

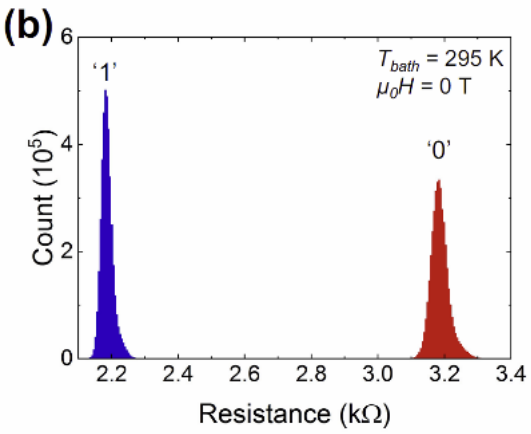


40 nm circular pMTJ with CoFeB/W/CoFeB composite free layer

Reset – set metastable state – read







"*Spin hall effect magnetic tunnel junction coinflips*"
Reim et al., Applied physics letters. Submitted arXiv 2209.01480

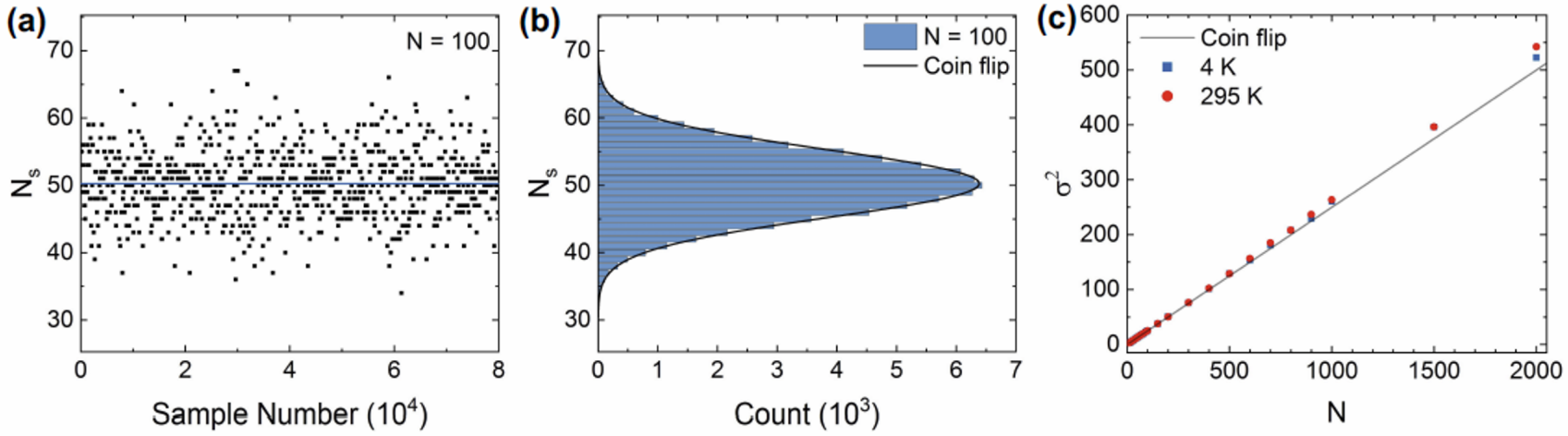
What makes one coinflip device better than another?

Quality of coinflip directly tied to quality of sample

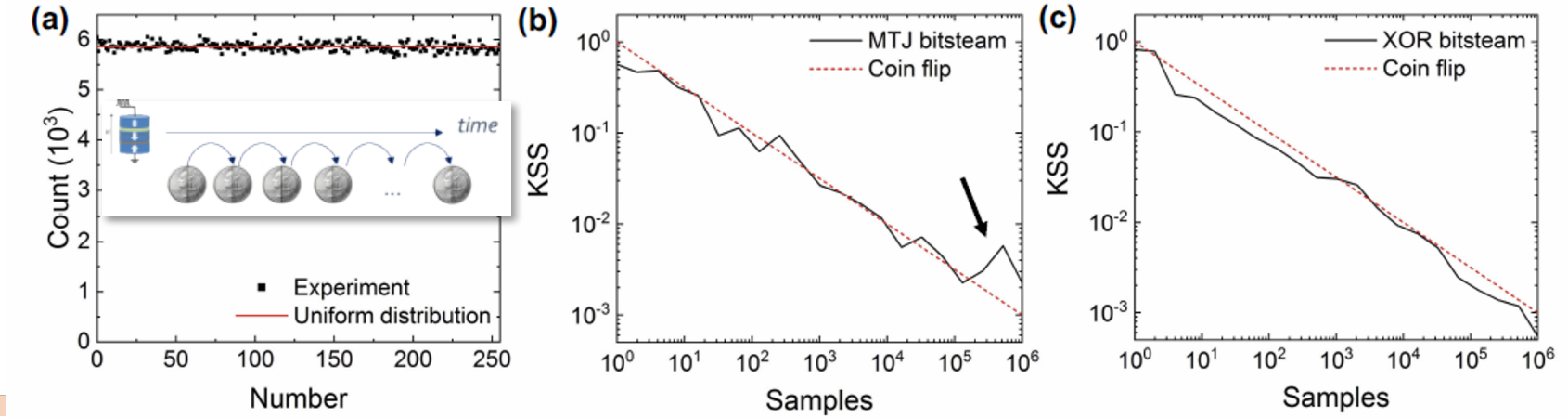
COINFLIPS

- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration

Blocks of 100 random coinflips show expected distribution of random samples



Generating 8-bit (integers from 0 – 255) from coinflips produces good random samples



AI-guided design of neuromorphic circuits – arbitrary distribution

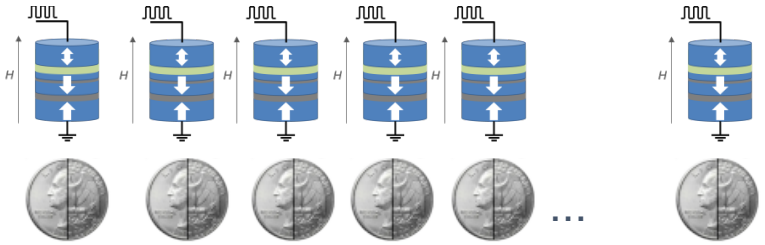
COINFLIPS

Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

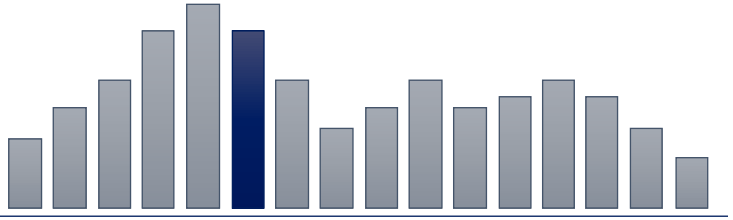
Particle Physics Demonstration



Many devices flipping at one time

Fair coins for *uniform* distribution

Biased coins for *non-uniform* distribution?



AI-guided design of neuromorphic circuits – arbitrary distribution

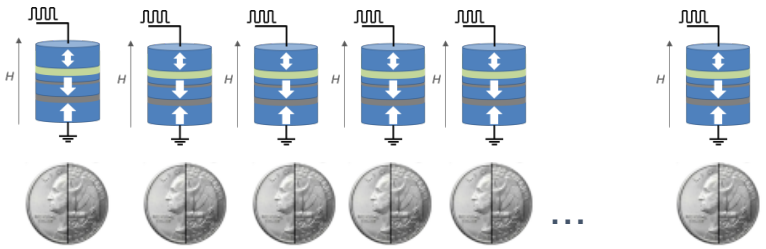
COINFLIPS

Tunable Stochastic Devices

Probabilistic Circuits and Architectures

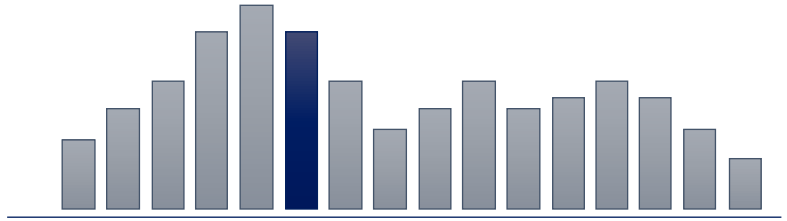
Probabilistic Neural Theory and Algorithms

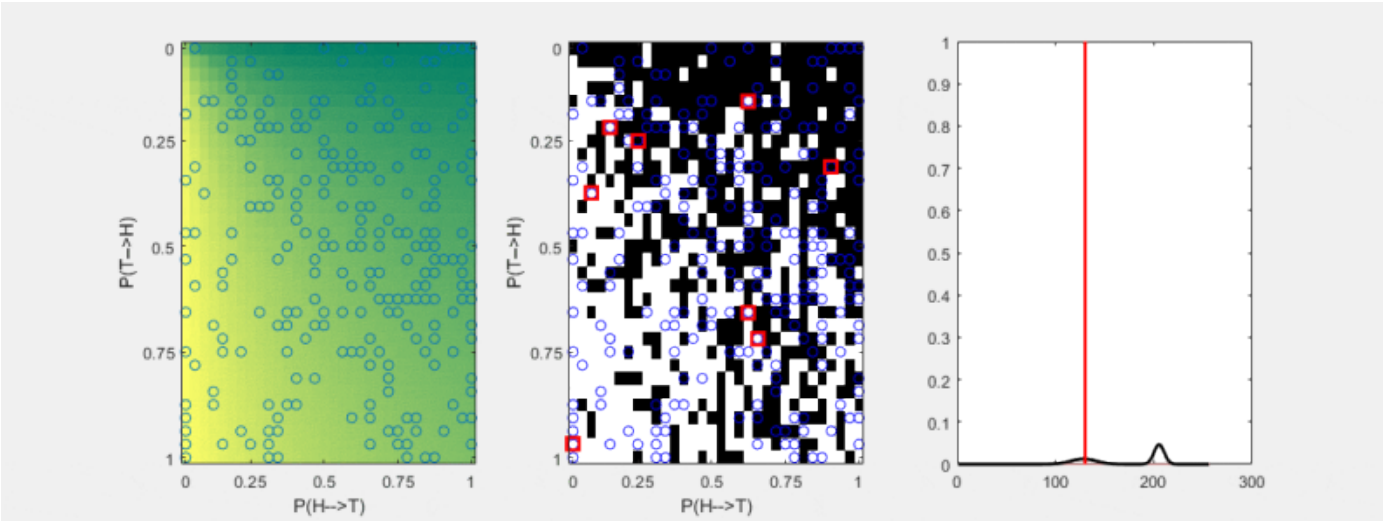
Particle Physics Demonstration



Many devices flipping at one time

Pick and choose biased coins for *non-uniform* distribution?



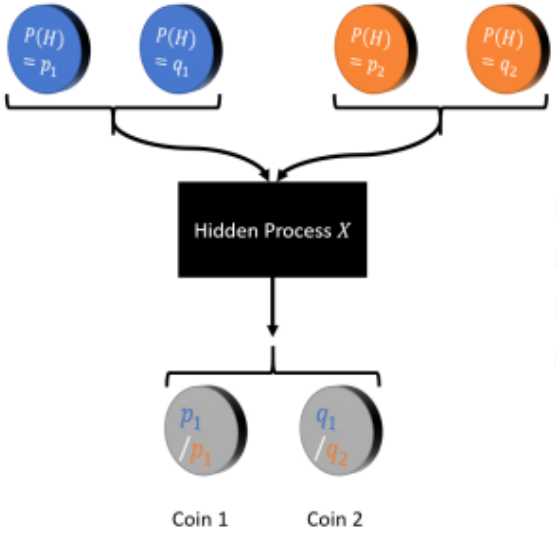


Simulation…

AI-guided design of neuromorphic circuits – arbitrary distribution

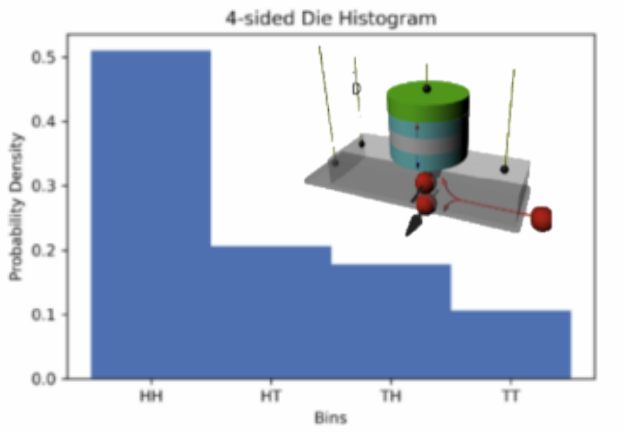
COINFLIPS

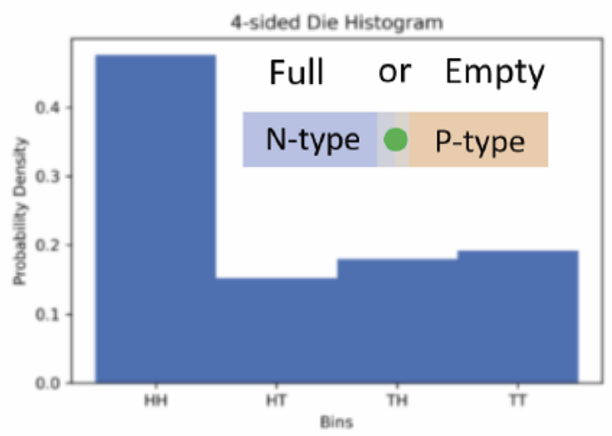
- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration



Target Distribution

$$\mathbb{P}[\text{Coin 1} = H \text{ and Coin 2} = H] = \frac{1}{2}$$
$$\mathbb{P}[\text{Coin 1} = H \text{ and Coin 2} = T] = \frac{1}{6}$$
$$\mathbb{P}[\text{Coin 1} = T \text{ and Coin 2} = H] = \frac{1}{6}$$
$$\mathbb{P}[\text{Coin 1} = T \text{ and Coin 2} = H] = \frac{1}{6}$$





"AI-enhanced Codesign for Probabilistic Neural Circuits"
Cardwell SG et al. In preparation

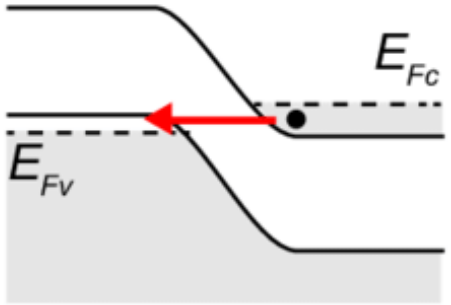
Sampling arbitrary distributions needs *weighted* coinflip devices

Weighted coinflip device – tunnel diode example

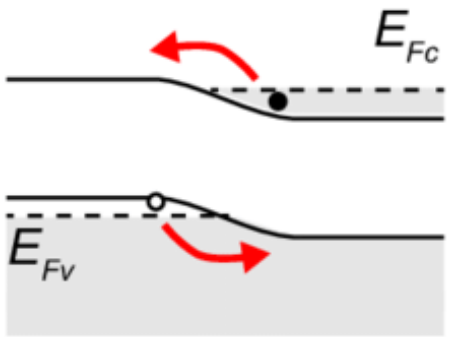
COINFLIPS

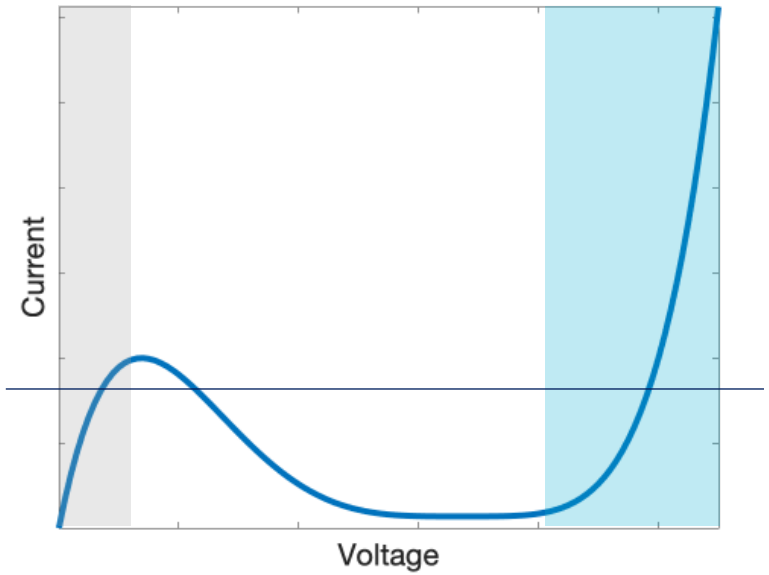
- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration

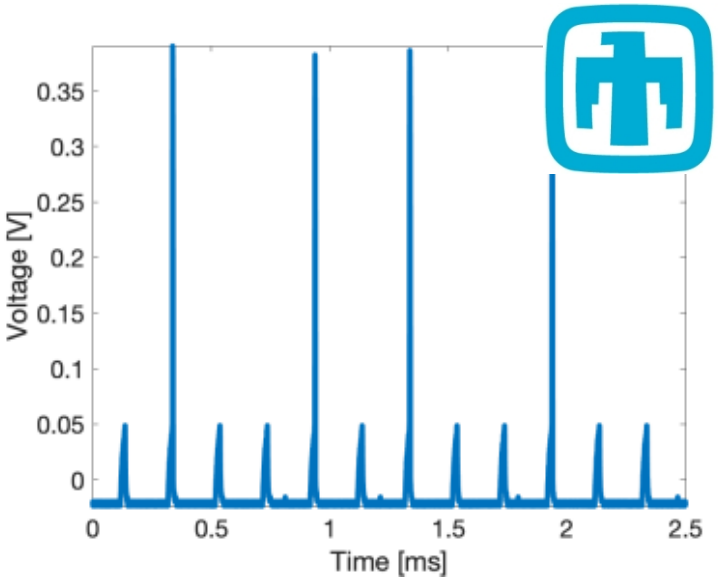
Tunnel Current



Traditional Diode Current







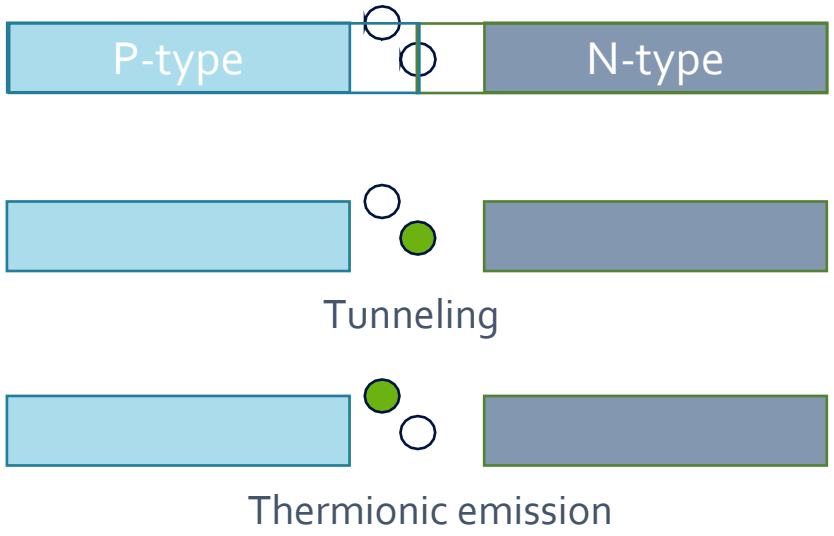
Tunnel branch or thermionic emission branch – acting as a detector

Tuning the tunnel diode

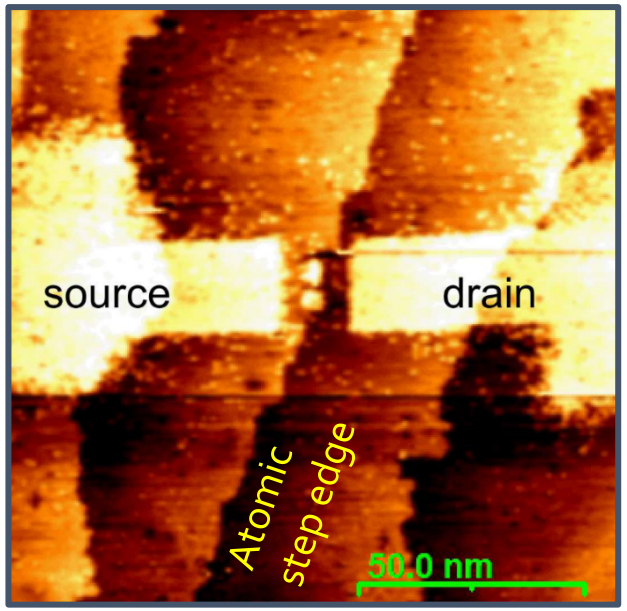
COINFLIPS

- Tunable Stochastic Devices
- Probabilistic Circuits and Architectures
- Probabilistic Neural Theory and Algorithms
- Particle Physics Demonstration

Origin of randomness are defects



Atomic precision advanced manufacturing



Far-reaching Applications, Implications, and Realization of Digital Electronics at the Atomic Limit (SNL GC LDRD)

Control placement of individual defects, manipulate using gates

Develop tunability through deep understanding of the underlying physics

Probabilistic Neuromorphic Algorithms

COINFLIPS

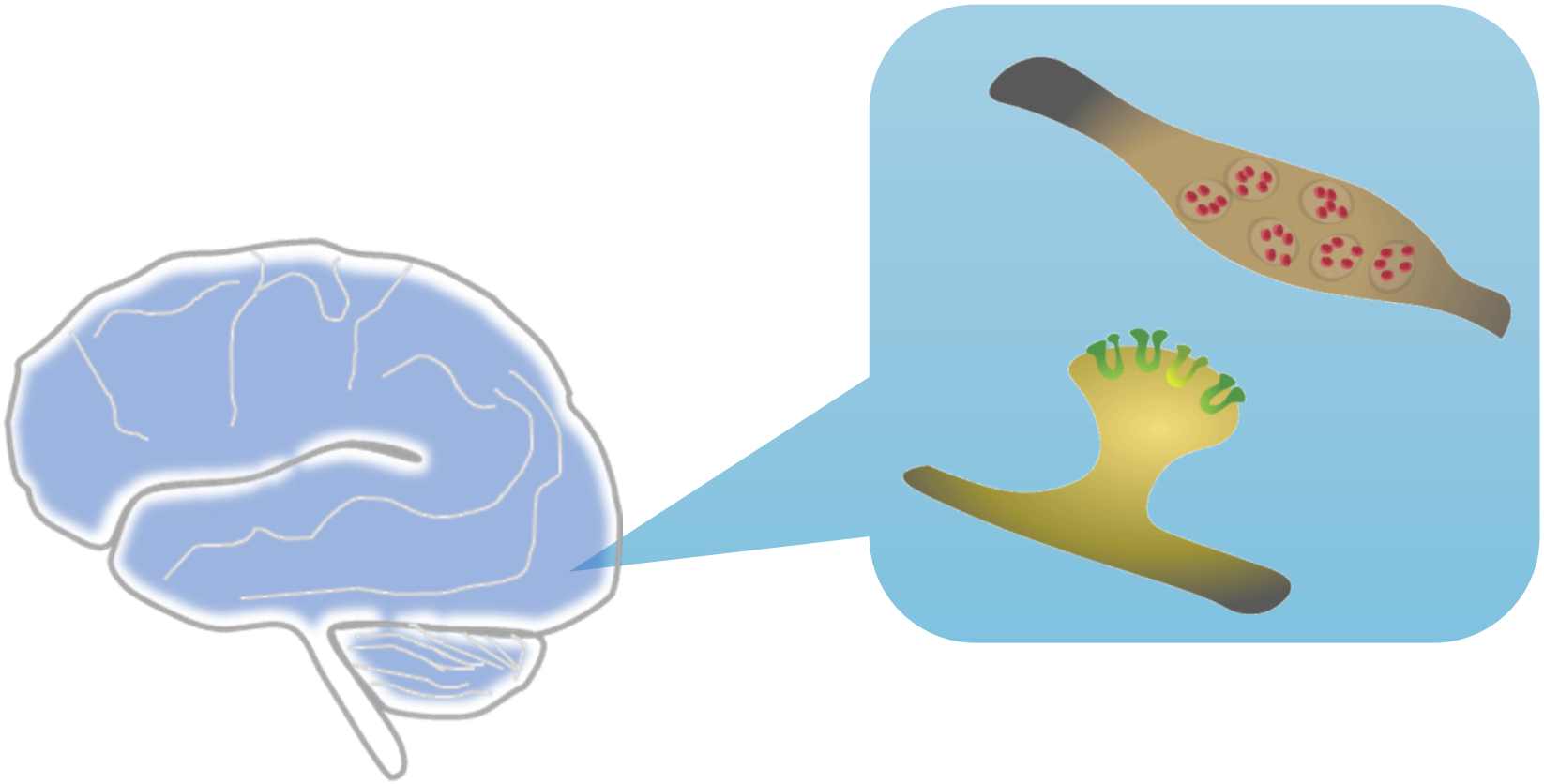
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



Probabilistic Neuromorphic Algorithms

COINFLIPS

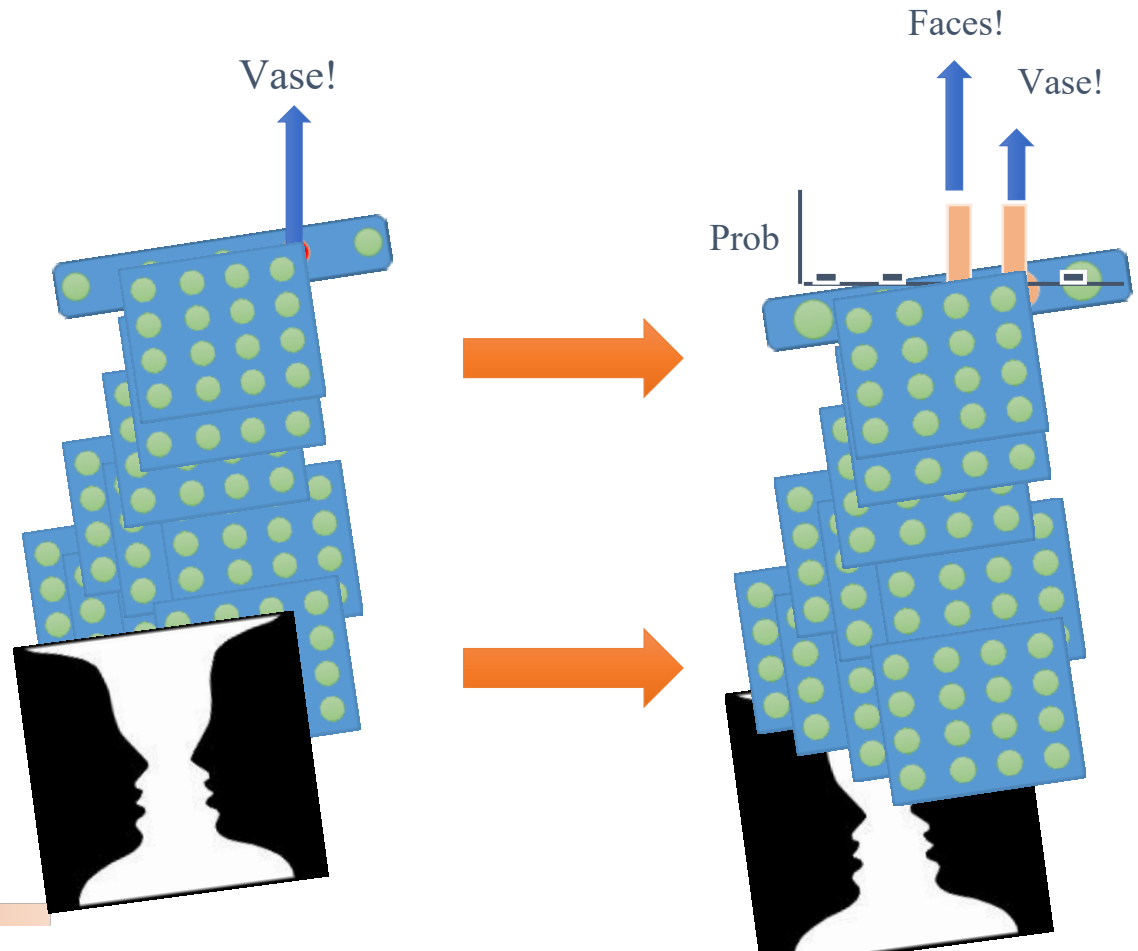
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



We are going to do this! Need to advance stochastic devices, probabilistic circuits, and Bayesian algorithms

Probabilistic Neuromorphic Algorithms

COINFLIPS

Tunable Stochastic Devices

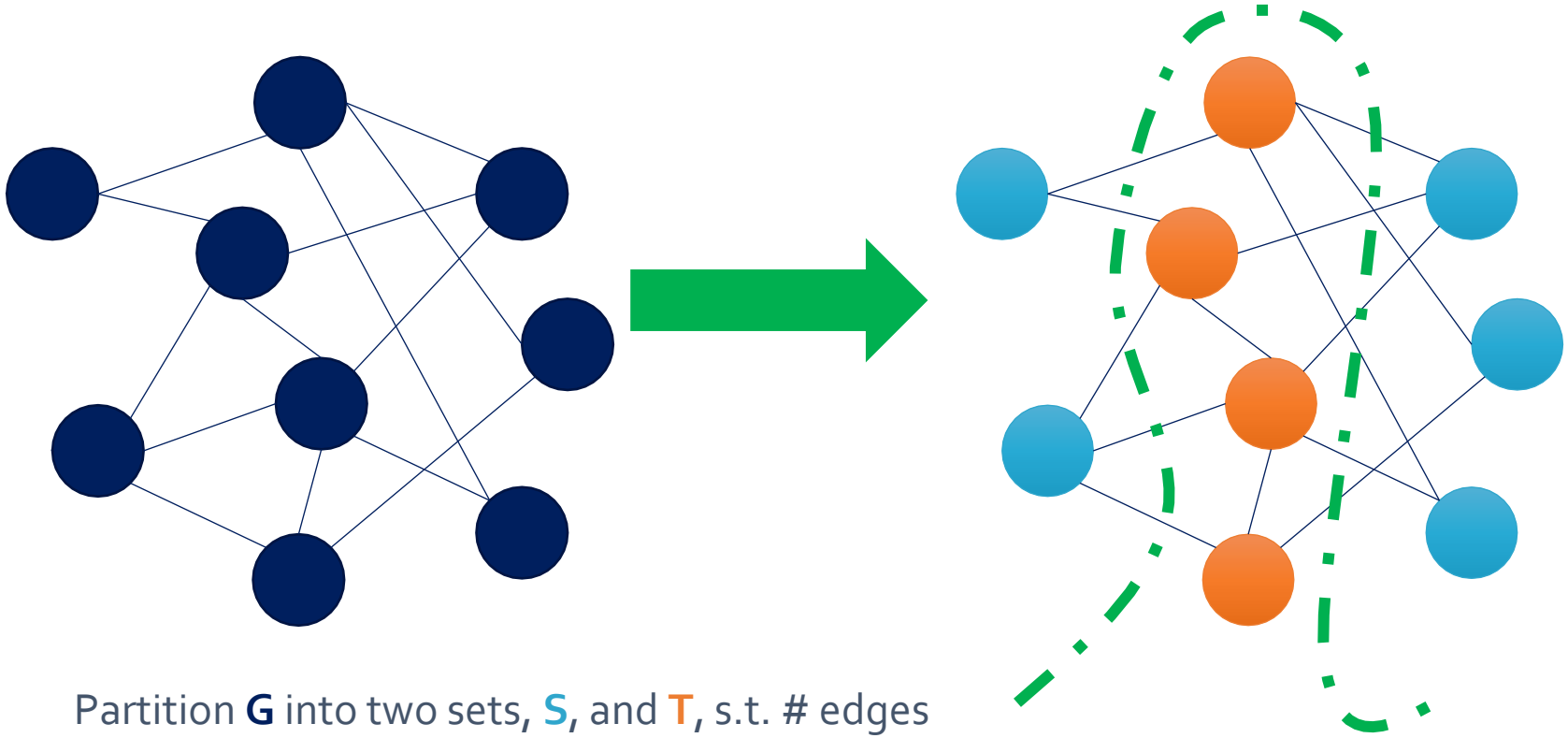
Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?

Approximate Algorithms for "Maximum Cut" of Graphs



Partition **G** into two sets, **S**, and **T**, s.t. # edges between S & T is as large as possible

Probabilistic Neuromorphic Algorithms

COINFLIPS

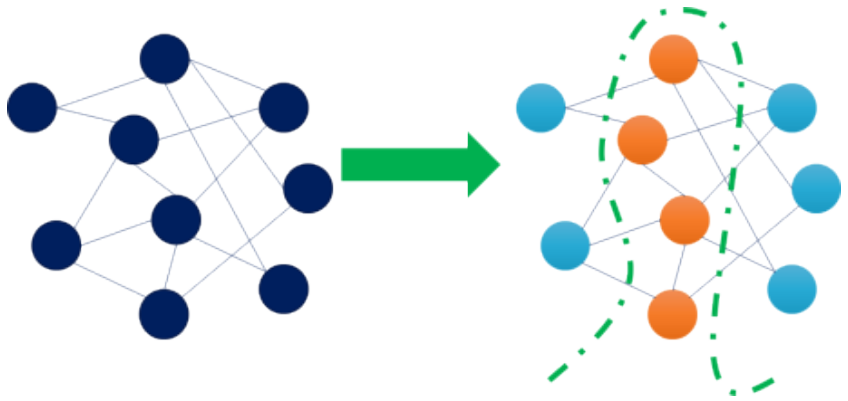
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



Partition **G** into two sets, **S**, and **T**, s.t. # edges between S & T is as large as possible

Algorithms for "Maximum Cut" of Graphs

- Applications to Ising models, VLSI circuit layout design, network design, data analysis, etc.
- Absolute solution is NP-Hard
- In practice, approximate algorithms are effective
  - Naïve algorithm ≈ 50%
  - Goemans Williamson ≈ 87.8%
  - Trevinson > 50%, in practice close to GW

Theilman et al., in preparation

Probabilistic Neuromorphic Algorithms

COINFLIPS

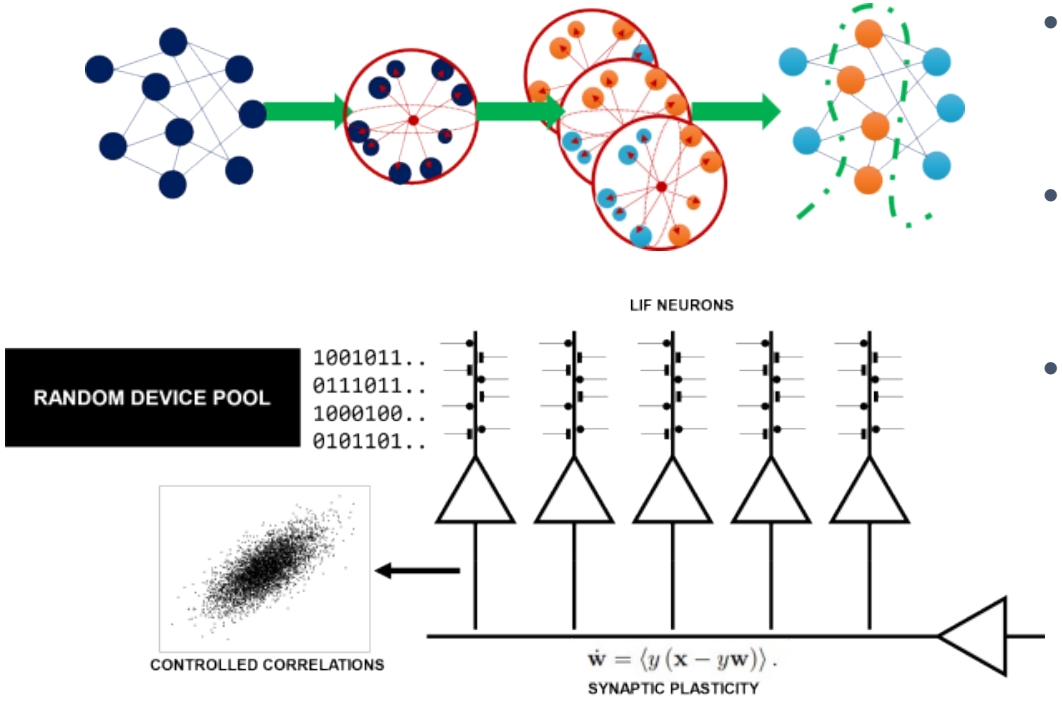
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



Theilman et al., in preparation

Probabilistic Neural Algorithms for Max Cut

- Use large number of COINFLIPS devices as a stochastic device pool
  - Continuous Sampling
- Neural circuits can be constructed to control correlations between output samples
  - Allows samples from algorithmic distributions
- Two neural approaches
  - Goemans Williamson – higher performance, neuromorphic sampling only
  - Trevisan – lower guaranteed performance, whole algorithm in neuromorphic

Probabilistic Neuromorphic Algorithms

COINFLIPS

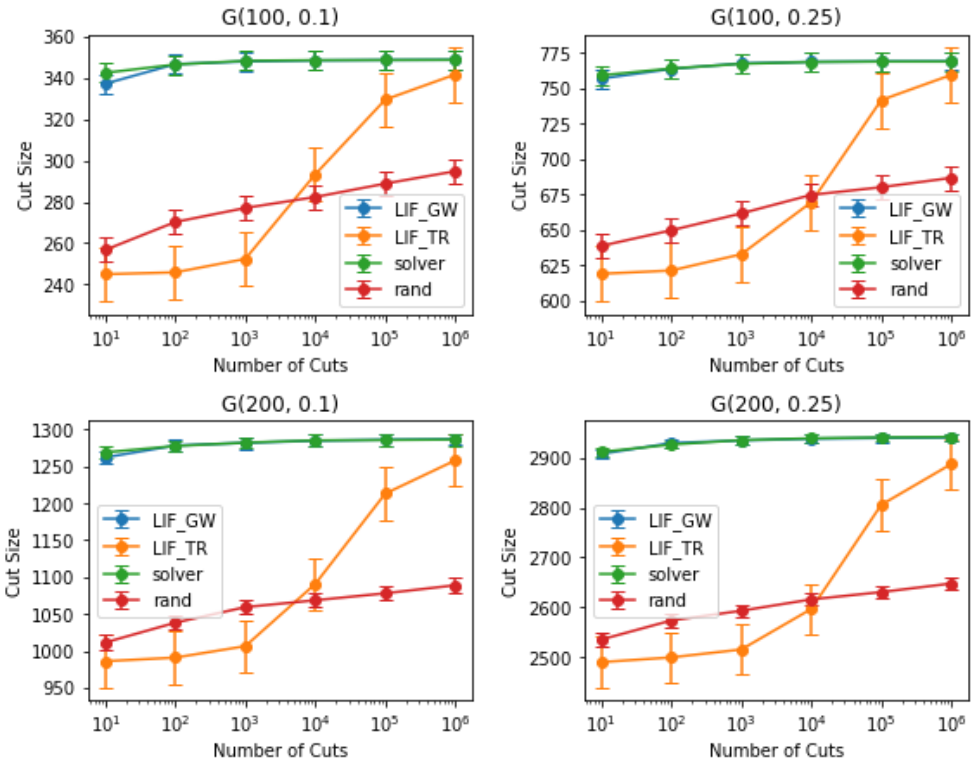
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



Theilman et al., in preparation

Probabilistic Neural Algorithms for Max Cut

- Neural GW algorithm performs as well as conventional solver
  - Only sampling part in neuromorphic
  - Only advantageous if sampling is dominant cost
- Neural Trevisan algorithm approaches conventional solver performance
  - Entire algorithm is in neuromorphic circuit
  - Learning neurons + stochastic devices
- Impact on device and materials
  - Integrating stochastic devices with neurons!
  - Need a lot of stochastic devices in parallel

Probabilistic Neuromorphic Algorithms

COINFLIPS

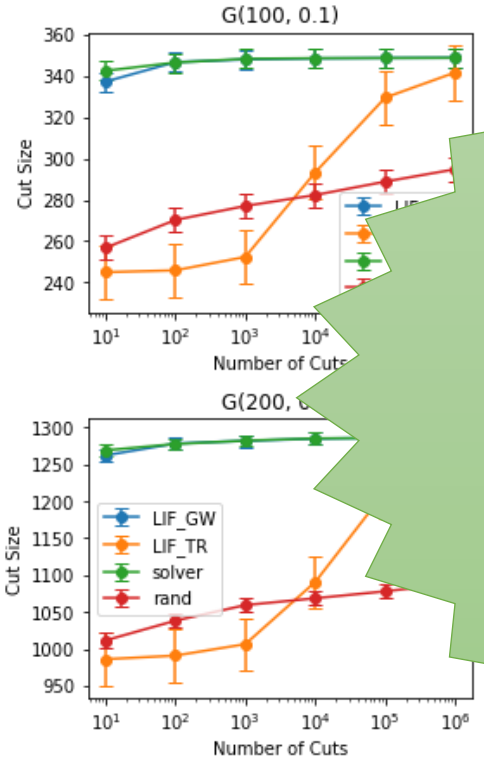
Tunable Stochastic Devices

Probabilistic Circuits and Architectures

Probabilistic Neural Theory and Algorithms

Particle Physics Demonstration

So what happens if we put stochastic devices with neurons?



hms for Max Cut

as

This could be a huge deal…

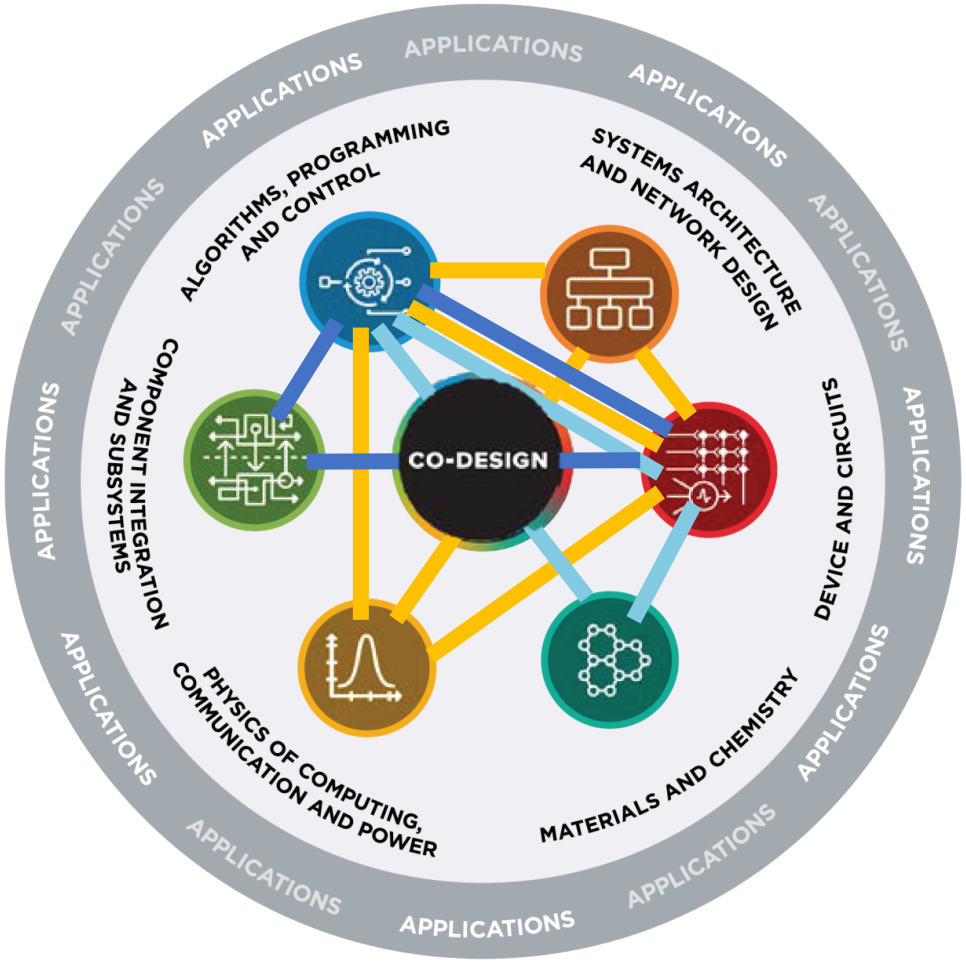
- Neuromorphic hardware & devices offer a lot of potential =>
  - Always missing big impact application
- Stochastic devices + neuromorphic systems =>
  - Impact real scalable computer science problem
- True co-design across materials, devices, circuits, and algorithms =>
  - No one in any one of these fields would think of doing this

n neurons!

ices in parallel

Theilman et al., in preparation

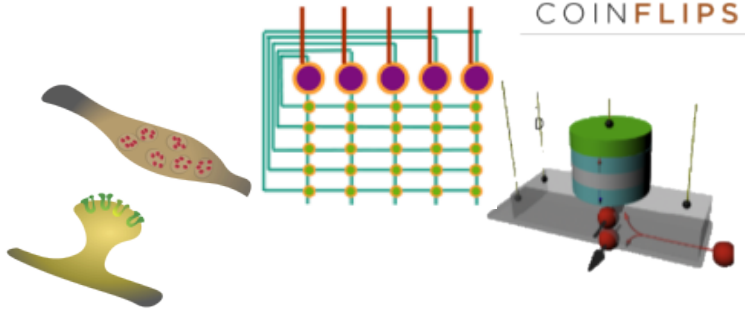
Path to Impact: Demonstrable progress using co-design as a microelectronics research approach

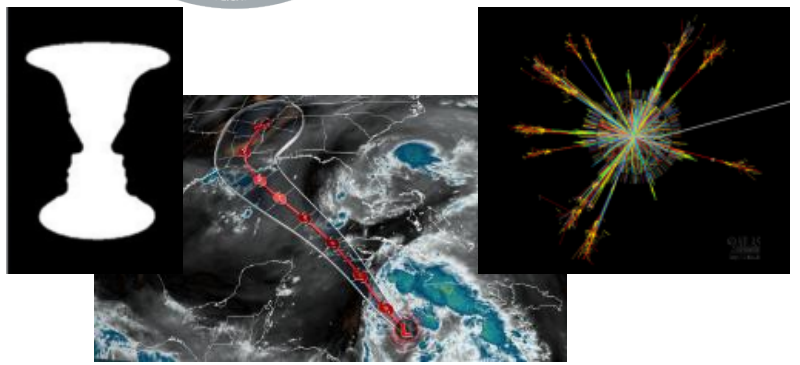


- **Theory research** is seeking to co-design algorithms with physics of stochastic devices for improved random number generation
- **AI-Guided circuit design** seeks to co-optimize stochastic materials and devices with algorithms
- **Bayesian neural networks** for in situ nuclear physics probabilistic jet detection

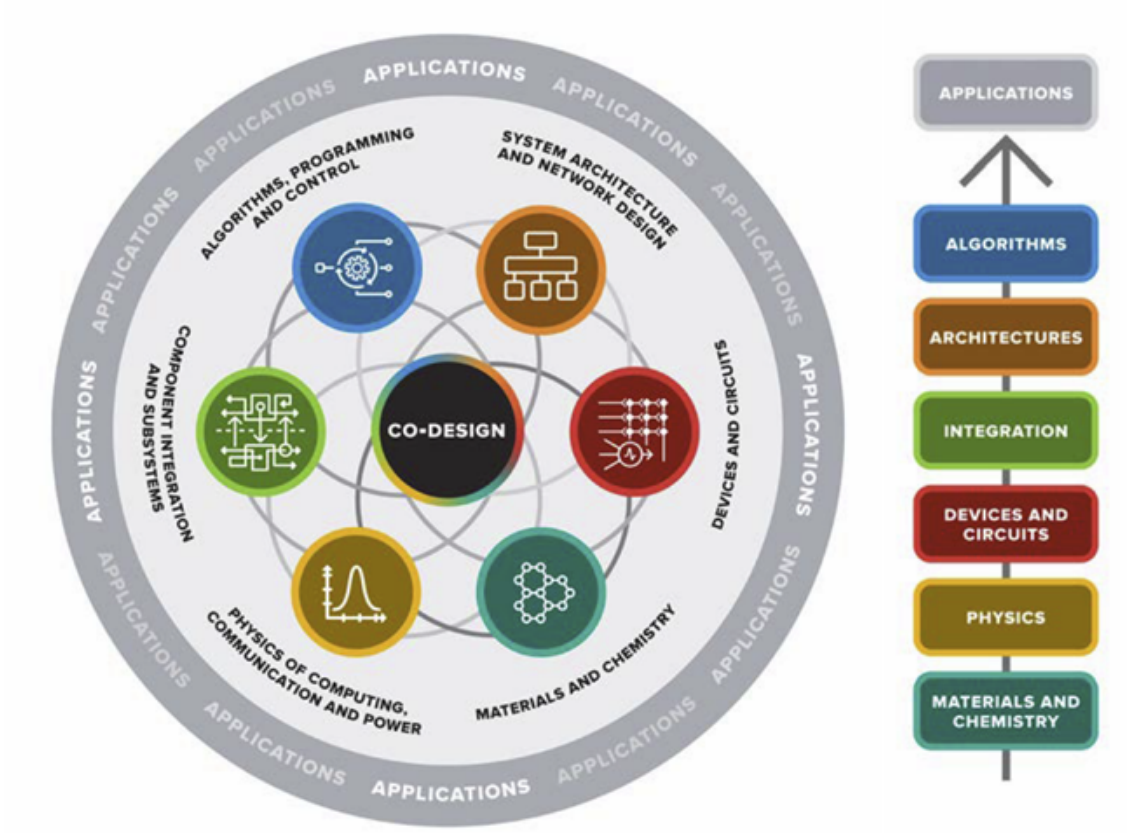
Summary: Probabilistic computing is perhaps an ideal target for exploring potential for microelectronics co-design

- All aspects of microelectronics (from materials to applications) have something to contribute
  - *Can show benefits from innovation at all scales*
- Stochastic devices
  + neuromorphic parallelism
  = broad application impact
  - *Both Mod-Sim and AI stand to benefit*
- Opportunity to consider important aspects of computing up front
  - *Address issues such as I/O, programmability, and theory from the onset, as opposed to after-the-fact*





COINFLIPS Today and in the Future



Today		*Future*
Random Number Generation Graph Analytics	→	*Bayesian Neural Networks* *Neuroscience-inspired Algorithms*
Hand-tuned Neural Circuits Evolution-optimized p-bits	→	*AI-Optimized Circuits* *In situ learning*
Magnetic Tunnel Junctions Tunnel Diodes	→	*Memristors* *…*

Thank You!

COINFLIPS

- Sandia National Laboratories:
  - **Brad Aimone (PI), Shashank Misra, Conrad James, Darby Smith, Suma Cardwell**, Brad Theilman, Ojas Parekh, Yipu Wang, Chris Allemang, William Severa
- Oak Ridge National Laboratory:
  - **Prasanna Date**
- New York University:
  - **Andy Kent**, Laura Reim
- Temple University:
  - **Les Bland**, Bernd Surrow, Jae Nam
- University of Texas Austin:
  - **Jean Anne Incorvia**, Jaesuk Kwon, Sam Liu
- University of Tennessee:
  - **Katie Schuman**, Karan Patel

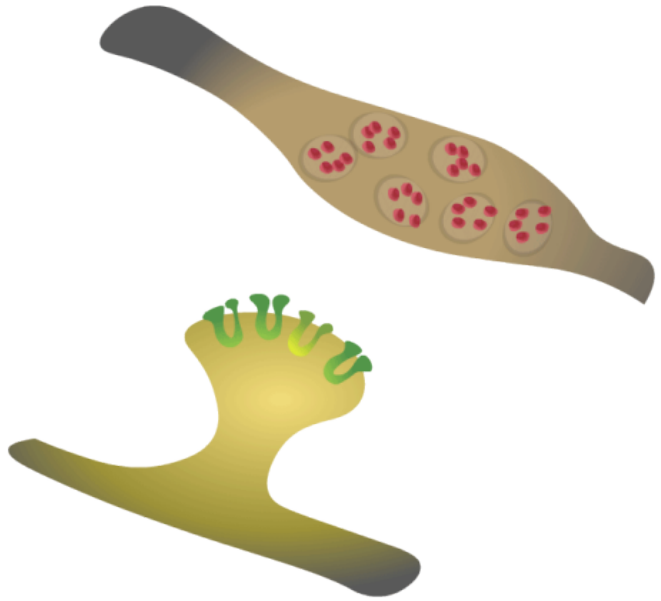
U.S. DEPARTMENT OF ENERGY | Office of Science

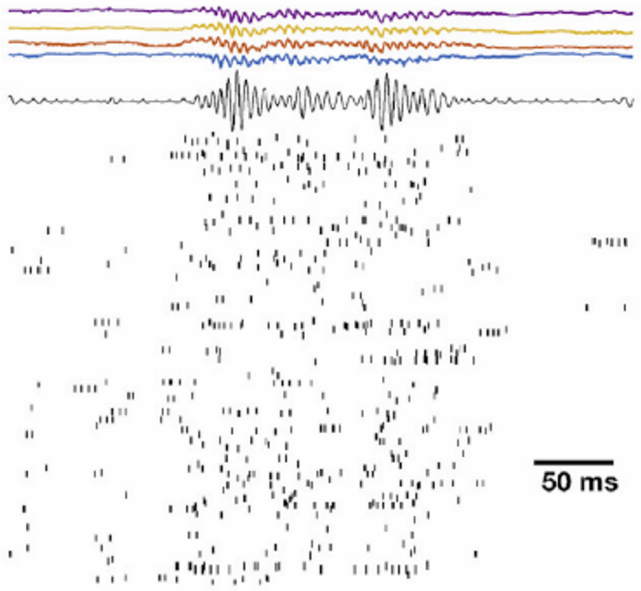
jbaimon@sandia.gov

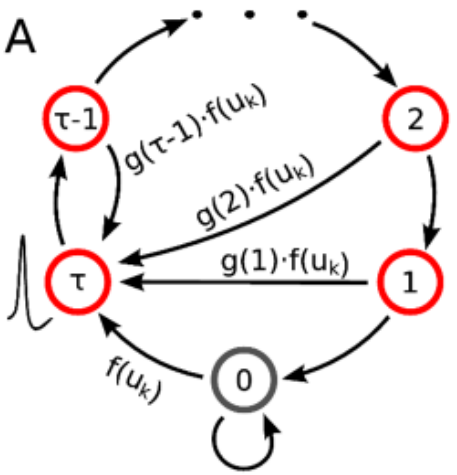
Backup: Neuroscience and stochastic computation

COINFLIPS

- There is a long history of viewing neuroscience through a stochastic computation perspective. Most of this history is independent of envisioned computing applications







"Independent sources of quantal variability at single glutamatergic synapses" Franks KM, Stevens CF, Sejnowski TJ J. Neurosci. 2003

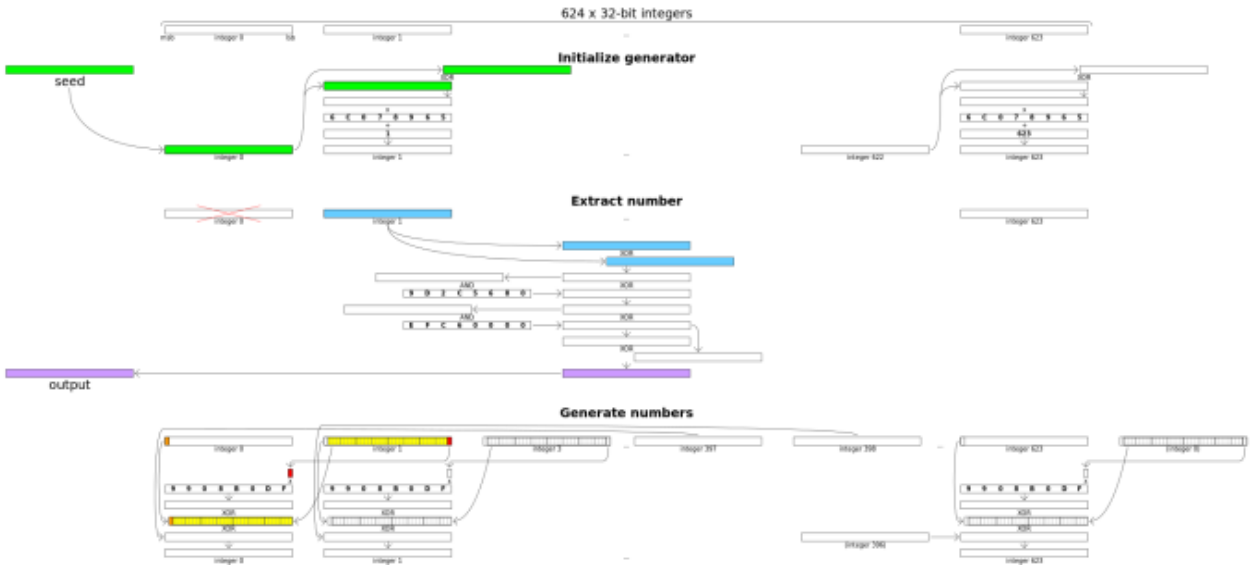
"Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion" Stella F et al., Neuron. 2017

"Neural Dynamics as Sampling: A model of stochastic computation …" Buesing L et al., PLOS Computational Biology. 2011

Backup: Pseudo Random Numbers vs True Random Numbers (Part 1: Strengths and limitations of PRNGs)

COINFLIPS

- Pseudo random number generators use deterministic procedures to generate a series of unpredictable and well-distributed, series of (typically uniformly distributed) numbers
- Strengths
  - Very well understood and established community of practice
  - Modern PRNGs are highly effective
  - Seeds allow verification and repeatability
- Weaknesses
  - Ultimately deterministic
  - Incur a non-trivial computational cost
  - Require care in parallelization
  - Require additional computation to convert to desired distribution



Visualization of Mersenne Twister PRNG algorithm (*Wikipedia)*

"*Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator.*" Matsumoto, M and Nishimura T. ACM Transactions on Modeling and Computer Simulation 1998

Backup: Pseudo Random Numbers vs True Random Numbers (Part 2: Strengths and limitations of tRNGs)

- True random number generators use some external source of randomness (cosmic rays, mouse movements, lava lamps, stochastic devices, etc), to provide a series of unpredictable inputs
- Strengths
  - True randomness ensures unpredictability and non-repeatability
  - Can provide a range of distributions
  - Energy-efficient (observing existing noise)
- Weaknesses
  - Typically not well understood
  - Some methods are very slow
  - No repeatability (seed setting) makes verification difficult
  - Most methods require additional computation to convert to desired distribution
  - Difficult to parallelize

Visualization of Lavarand true random number generator at Silicon Graphics (*Wikipedia)*

Backup: Pseudo Random Numbers vs True Random Numbers (Part 3: COINFLIPS aims to make tRNGs a more natural fit to computing)

- True random number generators use some external source of randomness (cosmic rays, mouse movements, lava lamps, stochastic devices, etc), to provide a series of unpredictable inputs
- Strengths
  - True randomness ensures unpredictability and non-repeatability
  - Can provide a range of distributions
  - Energy-efficient (observing existing noise)
- Weaknesses
  - Typically not well understood
  - Some methods are very slow
  - No repeatability (seed setting) makes verification difficult
  - Most methods require additional computation to convert to desired distribution
  - Difficult to parallelize

Stochastic behavior of well-understood COINFLIPS devices (TDs, MTJs) can be well characterized and controlled

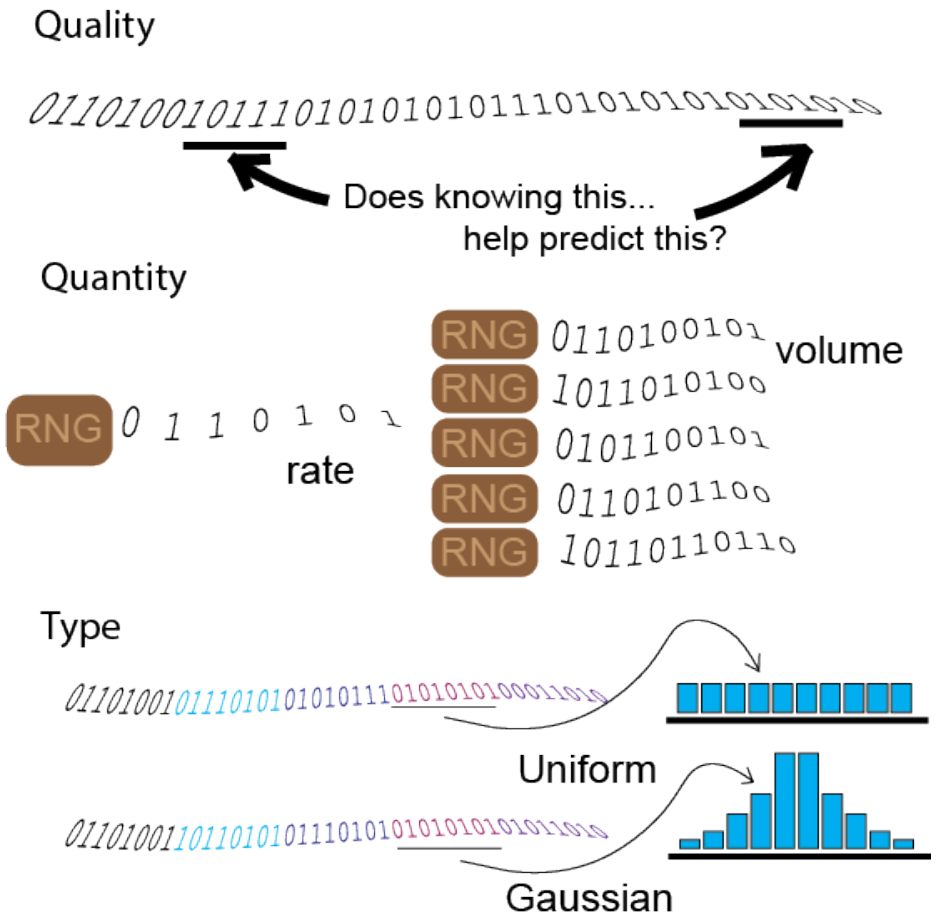
MTJs and TDs can be fabricated in situ with CMOS and run at high speeds

Treating RNG source as a tunable Bernoulli coinflip allows circuits to sample directly from distributions we desire

Embedding coinflips devices within neuromorphic architectures provides mechanism for parallel sampling

Backup: Generation of Random Numbers can be evaluated according to several metrics

The ***Quality***, ***Quantity***, and ***Type*** of random numbers all matter



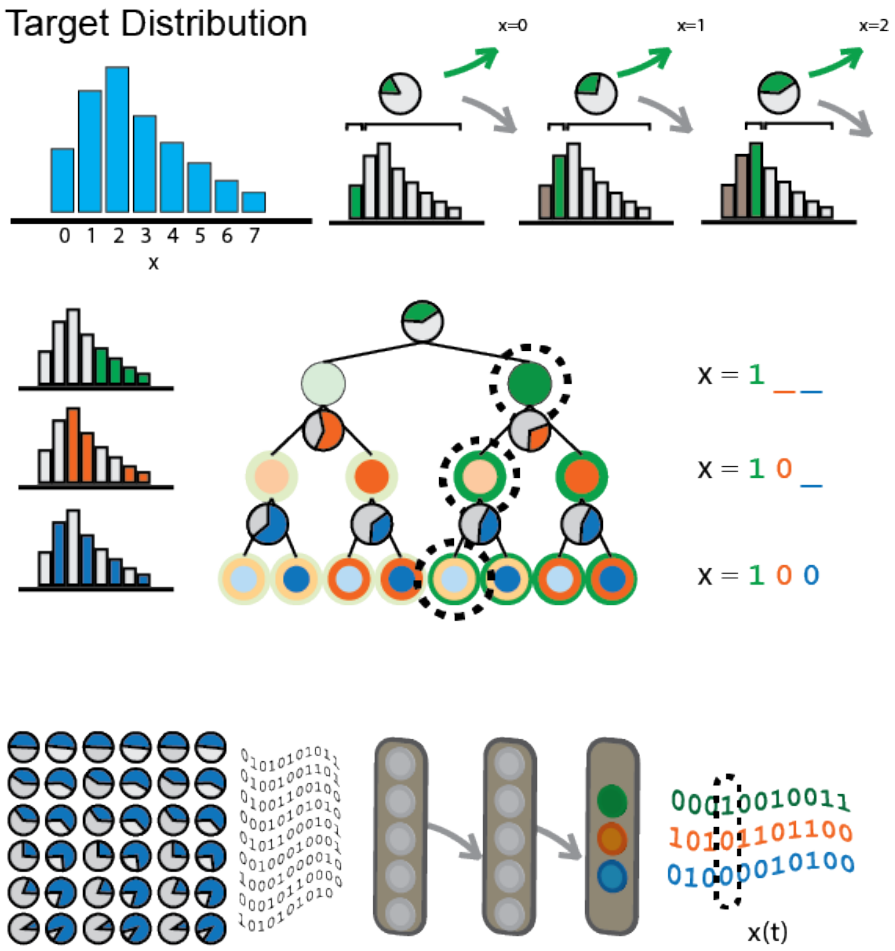
Quality of random numbers is measured by well defined criterion (e.g., NIST SP800 test suite)

Quantity of random numbers can be measured at both the device level (coinflips per second) or at the circuit output (random numbers per second)

Type of random numbers is measured by well established statistical criteria for comparing samples to target distributions (e.g., KL divergence)

"Probabilistic neural computing with stochastic devices"
Misra S et al. Advanced Materials. Submitted

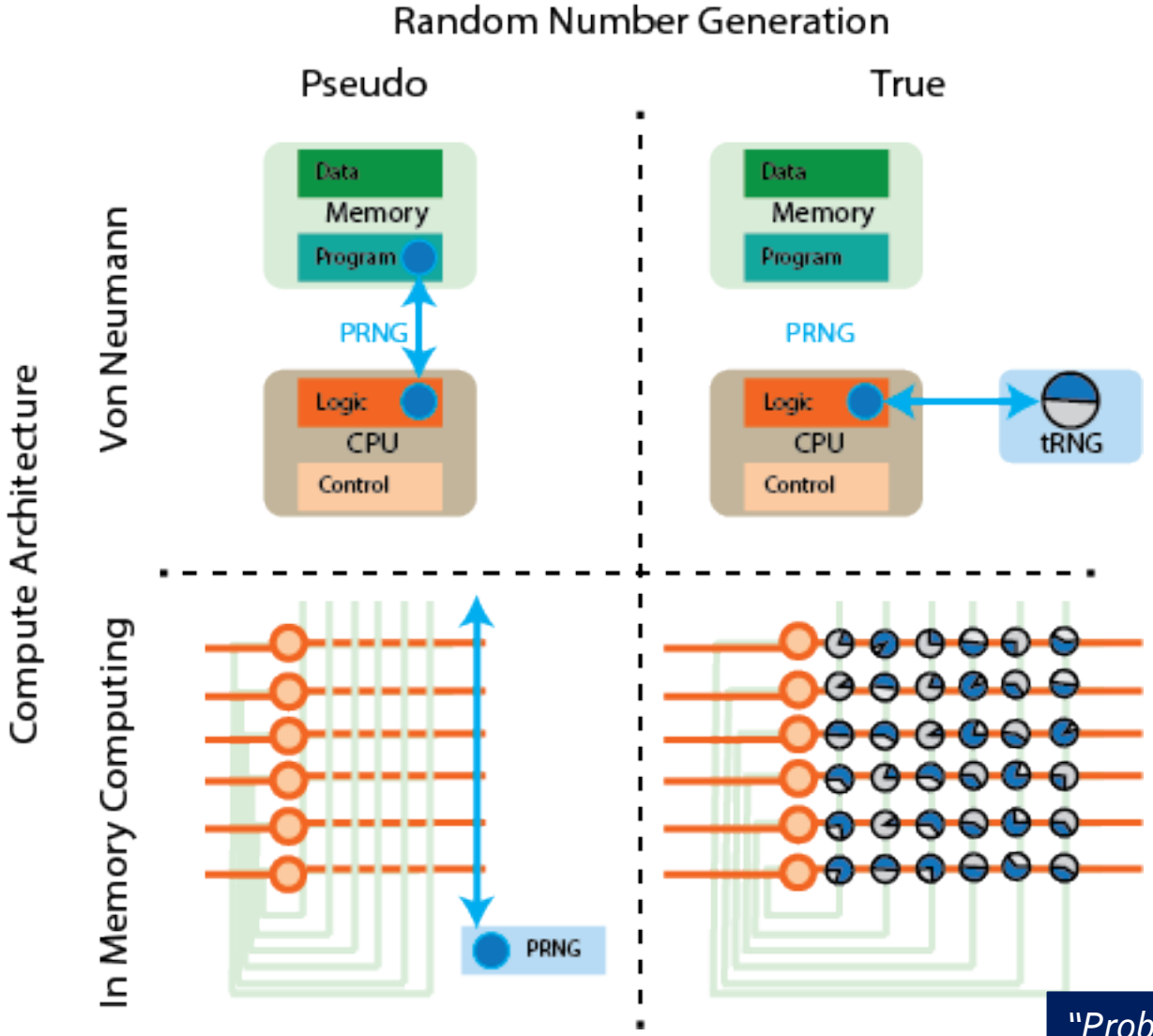
Backup: COINFLIPS is exploring several strategies to convert Bernoulli coinflips to samples from targeted distributions



- The output of any RNG is a binary encoded number from a given distribution (defined from a probability density function) at a certain precision
  - Example: target distribution on left
    - The precision of the RNG is k=3 (3-bits)
    - The PDF may be defined continuously (like a gamma distribution), but is realized as a k-bit precise number
- A single coin can do this by walking through each bin, retuning the P(H) to equal the marginal strength of that bin
  - Cost – O(1) coins; up to $O(2^k)$ timesteps
- We can expand out the PDF as a binary try, with each layer having a coin representing a place value of the binary output. The coin is tuned for the probability of that branch of the tree
  - Cost – ~O(k) tunable coins; up to O(k) timesteps
  - Cost – ~$O(2^k)$ fixed tuned coins; up to O(k) timesteps
- We can also train a neural network to learn how to convert arbitrarily tuned coins to a sample of a desired distribution
  - Cost – ~O(???) tunable coins; possibly O(1) timesteps

"Probabilistic neural computing with stochastic devices"
Misra S et al. Advanced Materials. Submitted

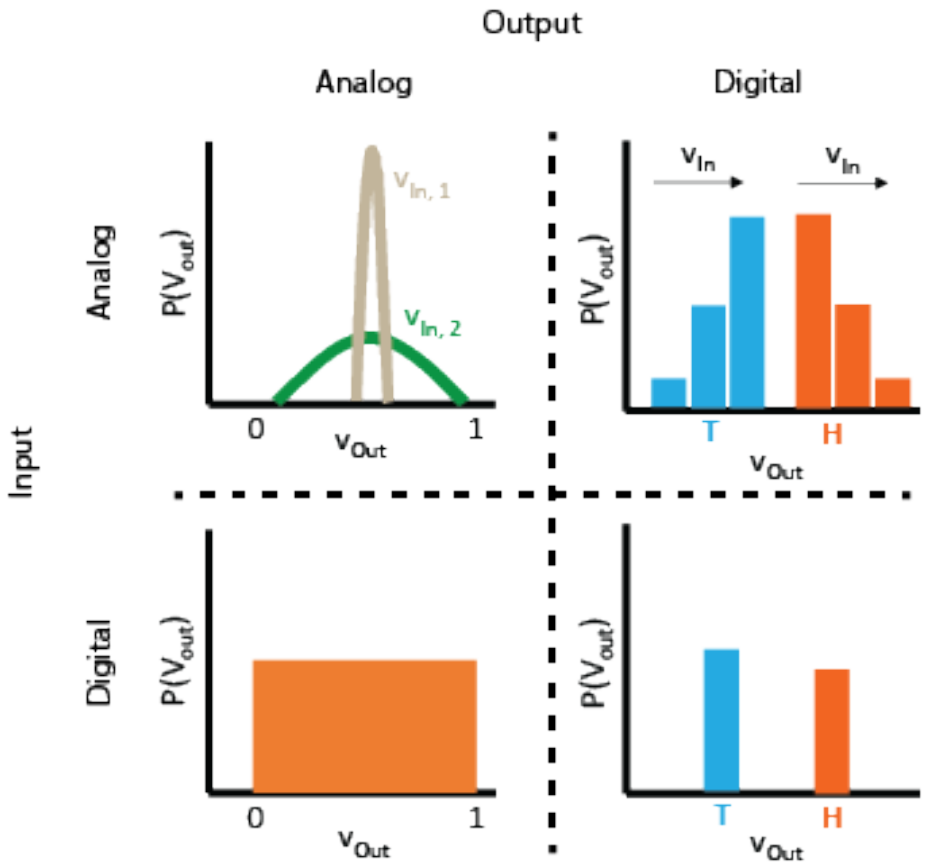
Backup: Avoiding a random number bottleneck



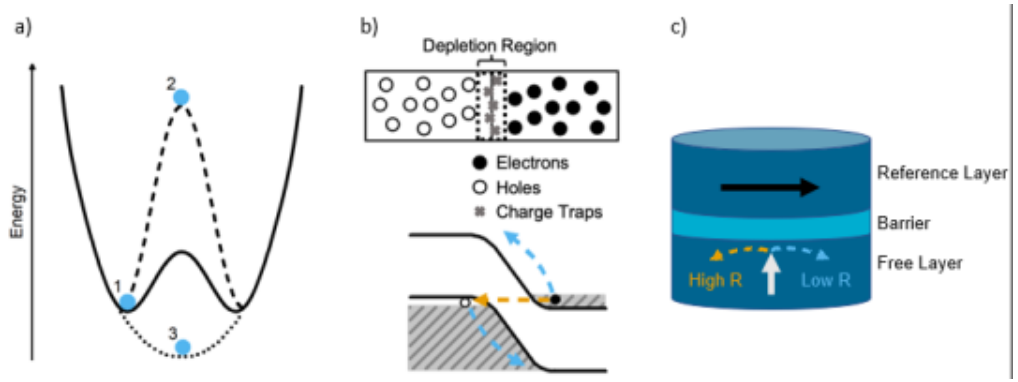
- In serial systems, PRNG and tRNG improvements will both be limited by von Neumann bottleneck
- Depending on implementation, multiple PRNGs are required (which requires care in seed setting) or a shared PRNG can become a bottleneck in parallel systems
- In envisioned probabilistic neuromorphic systems, the generation of tRNGs in situ with computing and memory will eliminate bottlenecks.

"Probabilistic neural computing with stochastic devices"
Misra S et al. Advanced Materials. Submitted

Backup: COINFLIPS devices target the best of analog and digital



- Most devices can be operated in an analog or digital manner
- We desire well-defined outputs (e.g., a digital 'Heads' or 'Tails')
- We can either adjust the probability of an output by a stochastic read or write



"*Probabilistic neural computing with stochastic devices*"
Misra S et al. Advanced Materials. Submitted

Has the tremendous success of deterministic computing left probabilistic applications behind?

COINFLIPS

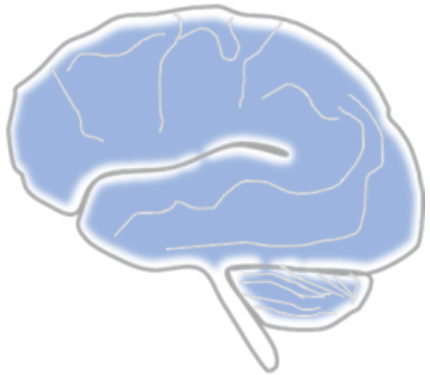
- *Stochasticity reveals contrast in computing approaches*
- Modern microelectronics spends tremendous resources in enforcing determinism
- The brain embraces and controls stochasticity across spatial and time scales
- *Developing probabilistic computing to address probabilistic applications*
- ***COINFLIPS*** is combining stochastic devices with neuromorphic architectures
- Co-design is proving invaluable in developing this novel paradigm for microelectronics

Which approach is best to interpret an ambiguous input?

~20 W

~10^{15} synaptic events / second

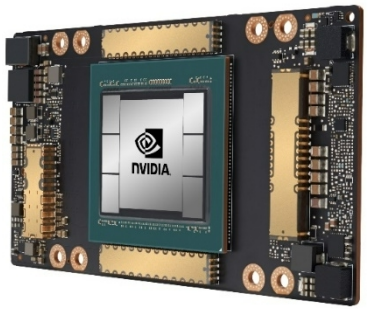
Fully stochastic



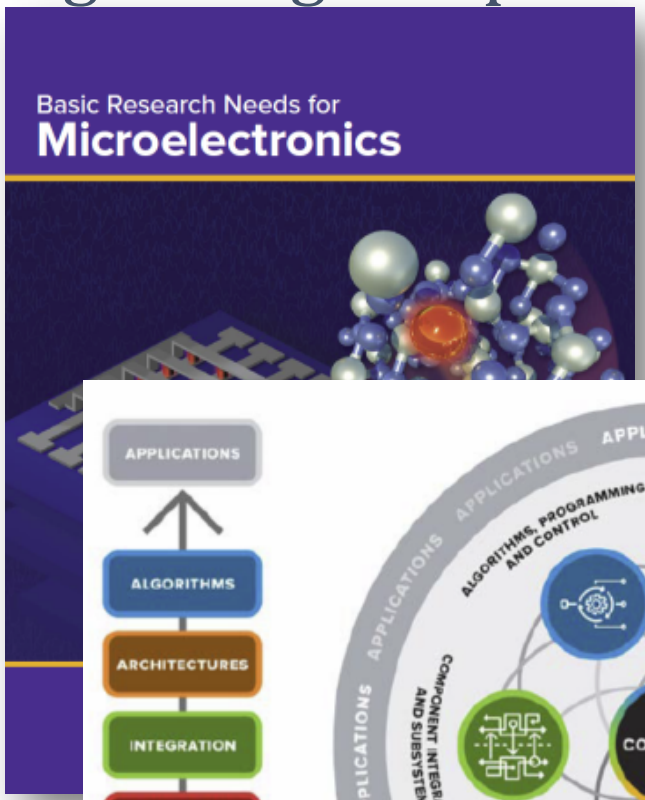
~400 W

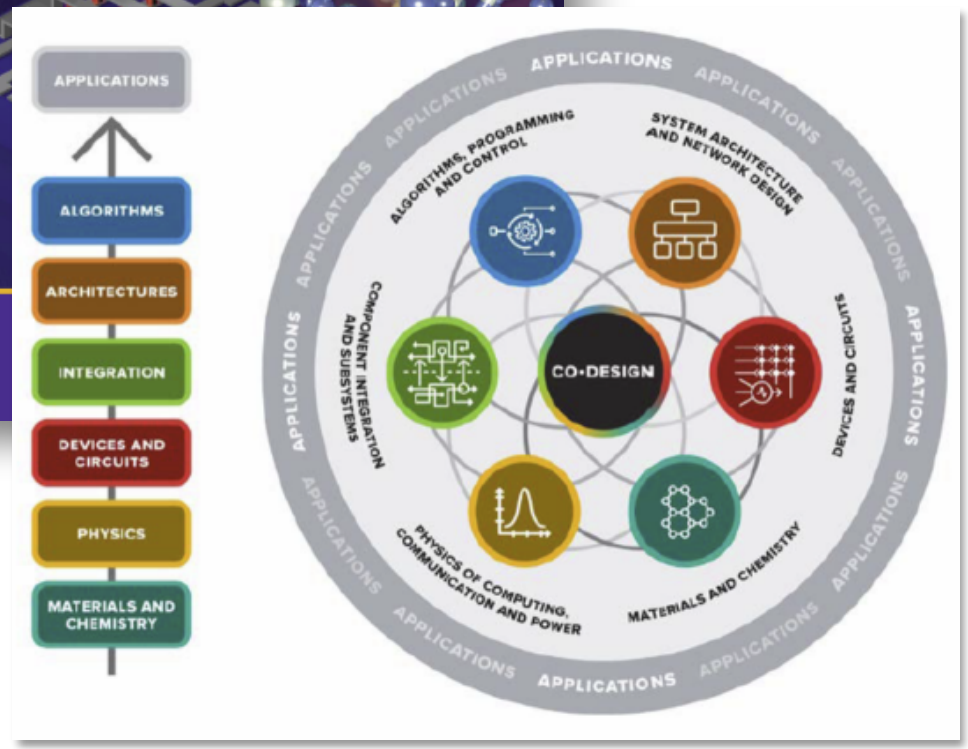
~10^{13}-10^{14} FLOPS

Fully deterministic



Backup: *Co-design* is increasingly hypothesized as a solution to growing competitive challenges in microelectronics





To enable new generations of energy-efficient computing systems over the next decade, ***a complete reconceptualization of the science and technology underlying the microelectronics co-design approach is needed*** *to integrate emerging devices, materials, interconnects, and non-linear phenomena with the needs of scientific computing applications.*