SAND2023-12678C

Sandia National Laboratories

Exceptional service in the national interest

Codesigning a Probabilistic Computing Future: Applications and Algorithms

Sampling Distributions from Biased Coins

Darby Smith

March 3rd, 2023

SIAM Computer Science and Engineering Minisymposium

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and wholly owned subsidiary of Honeywell International Inc. for the U.S. onal Nuclear Security Administration under contract DE-NA0003525.

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

Stochasticity reveals contrast in computing approaches

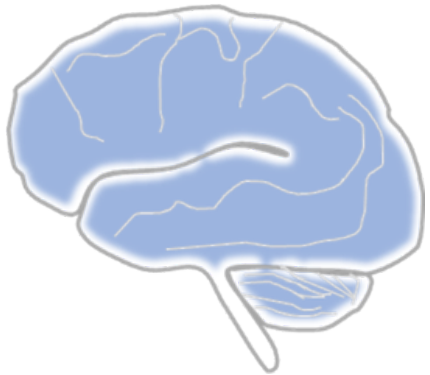
- Modern microelectronics spends tremendous resources in enforcing determinism
- The brain embraces and controls stochasticity across spatial and time scales

Developing probabilistic computing to address probabilistic applications

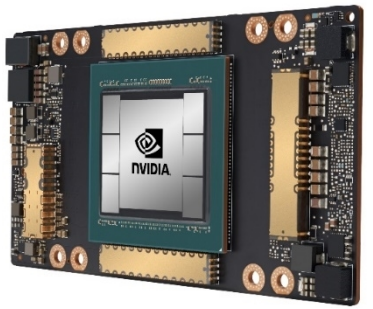
- Co-design is proving invaluable in developing this novel paradigm for microelectronics

Which approach is best to interpret an ambiguous input?

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

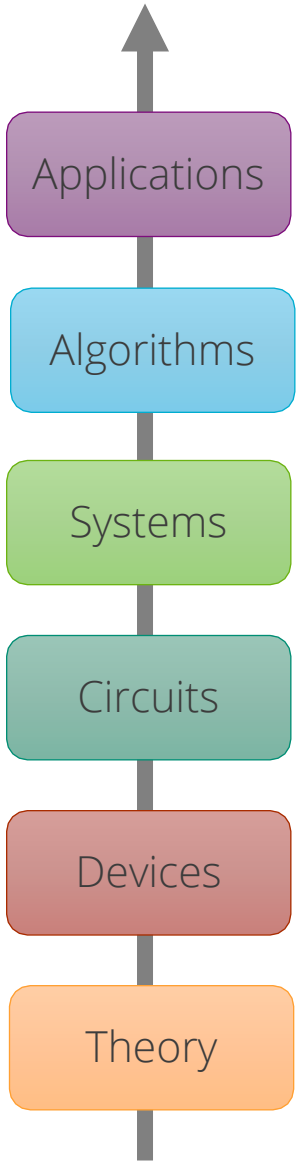


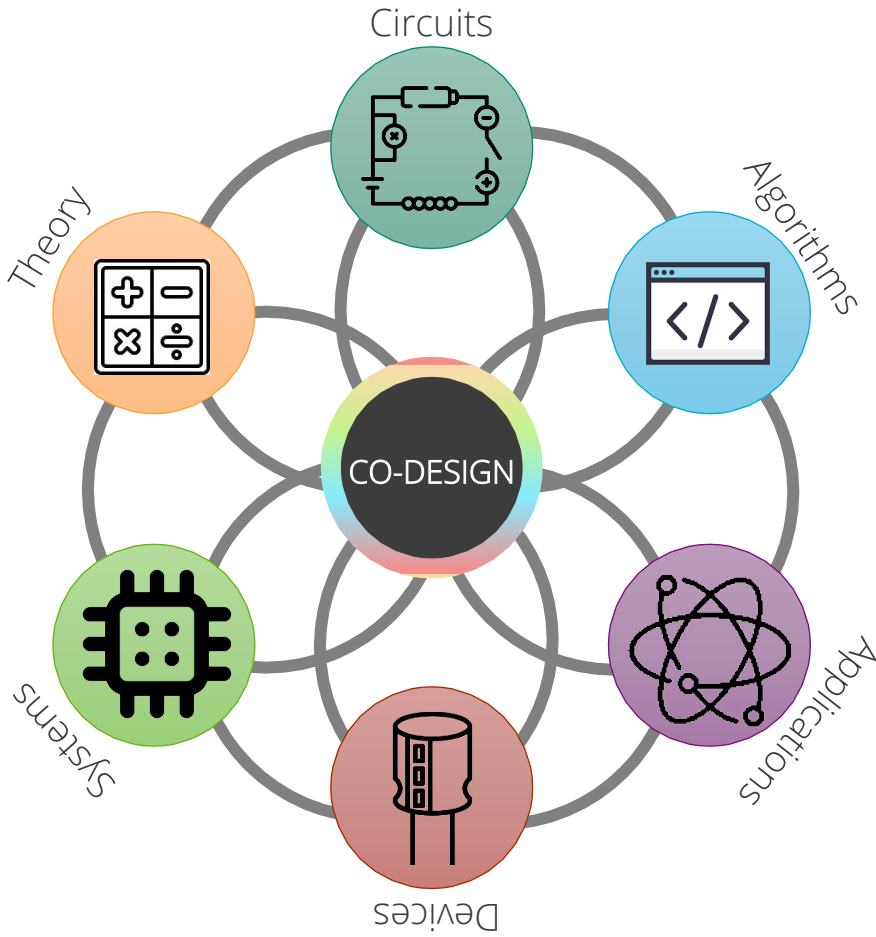
~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic

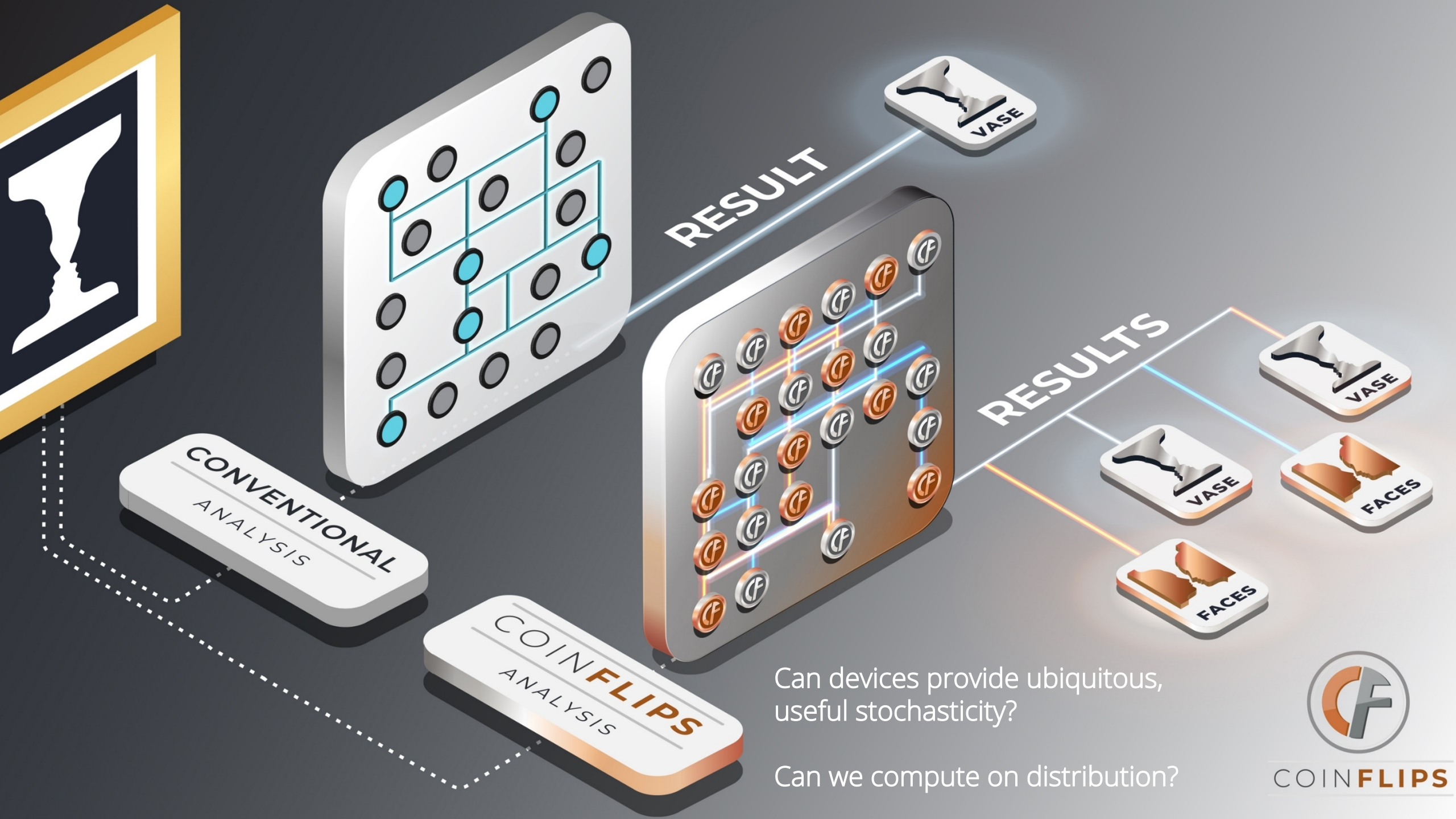


Today's Minisymposium

- What features of probabilistic computing would enable the applications of tomorrow?
- How can we design novel algorithms that utilize probability and stochasticity in relevant applications?
- What new probabilistic, numerical, and mathematical theory are necessary?
- How can design of systems and circuits guide our mathematical choices?

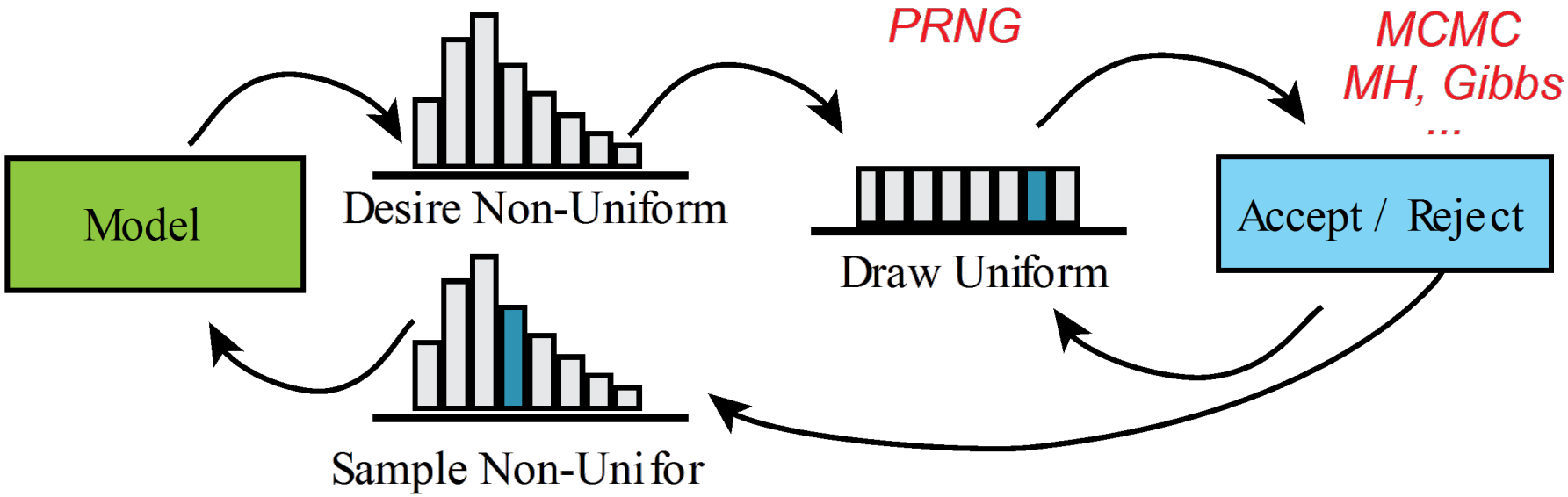





RESULT
VASE
RESULTS
VASE
VASE
FACES
FACES
CONVENTIONAL
ANALYSIS
COINFLIPS
ANALYSIS
Can devices provide ubiquitous, useful stochasticity?
Can we compute on distribution?
COINFLIPS

A New Probabilistic Computing Paradigm

Today

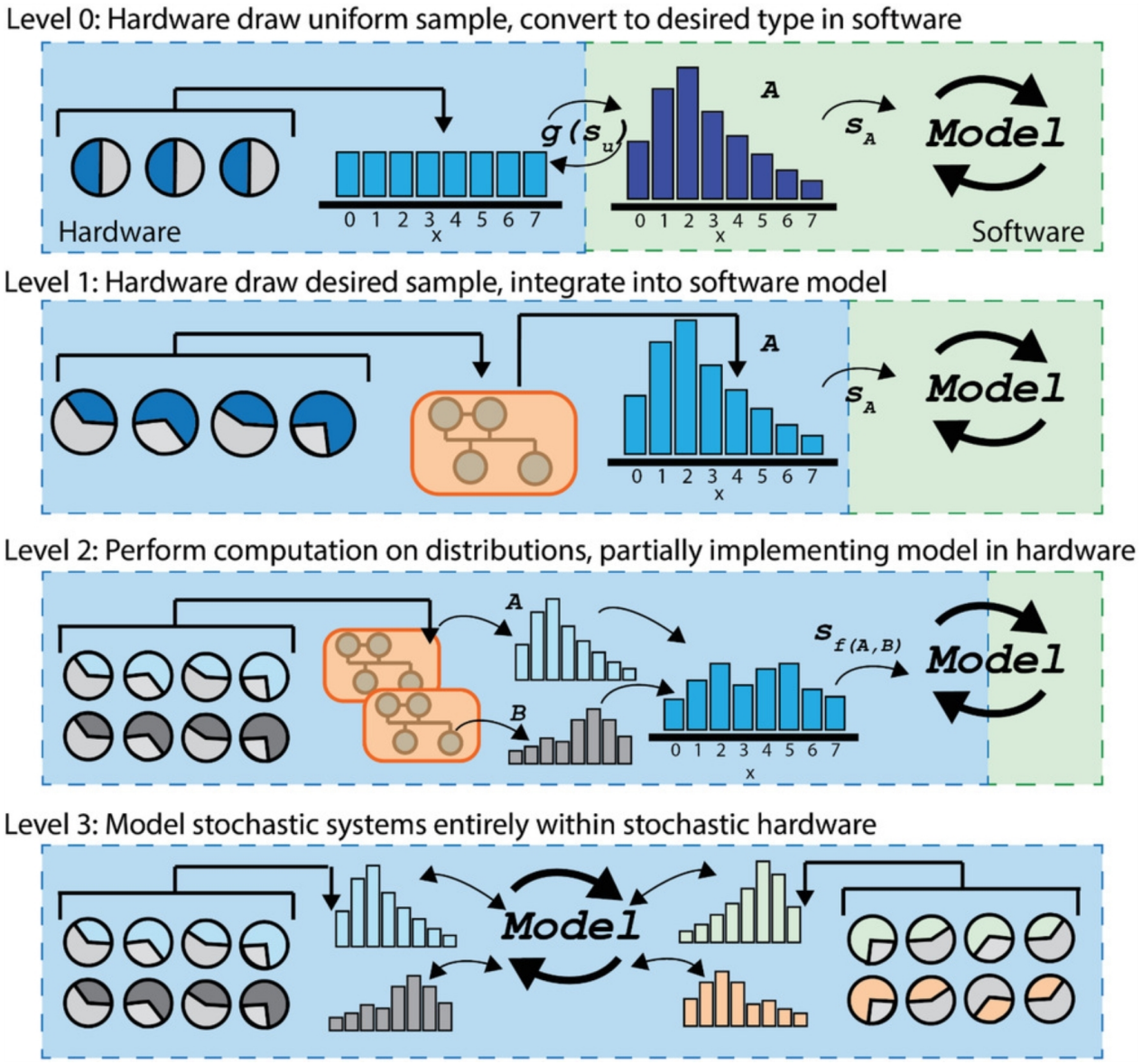


Future

Model stochastic systems entirely within stochastic hardware, no software.

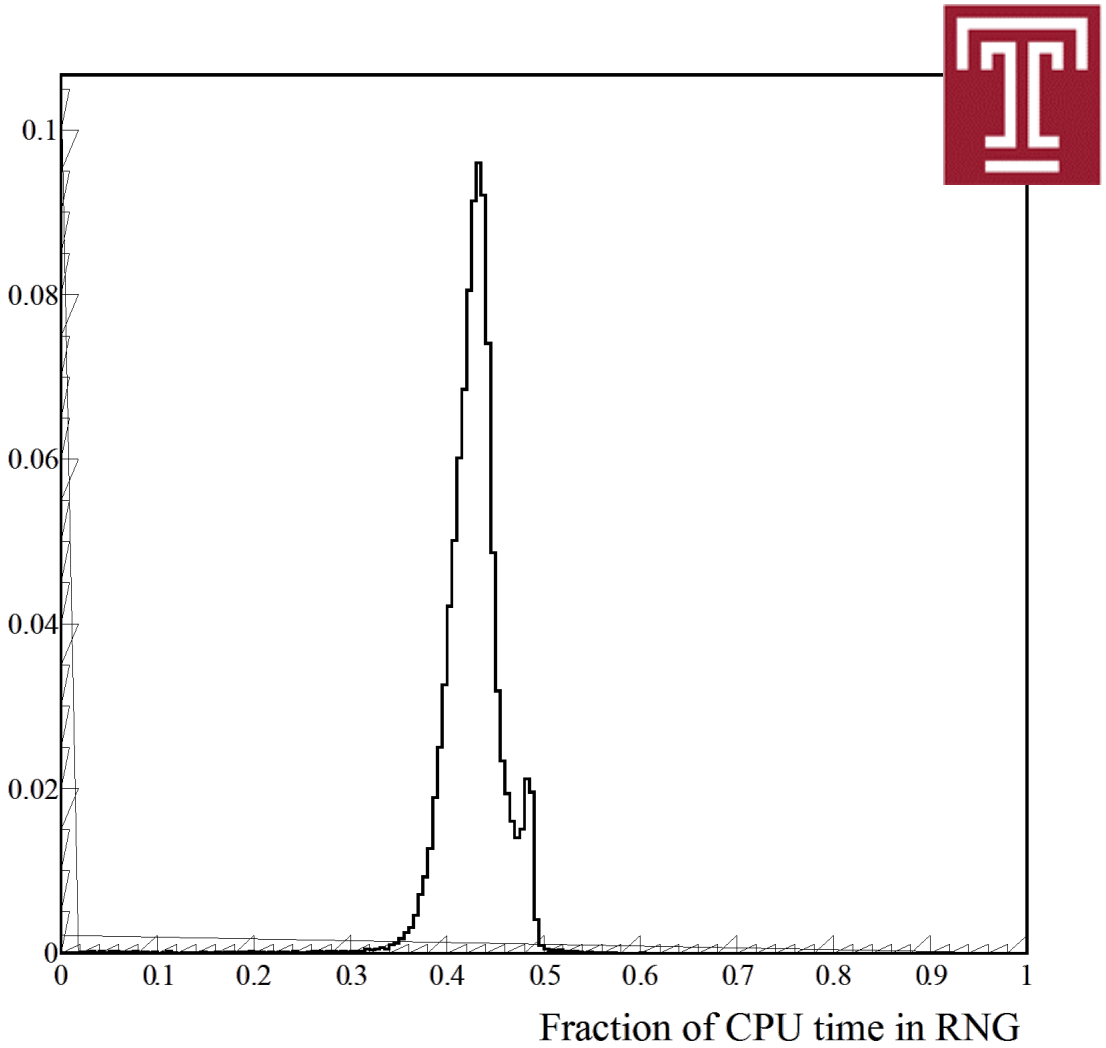
A New Probabilistic Computing Paradigm

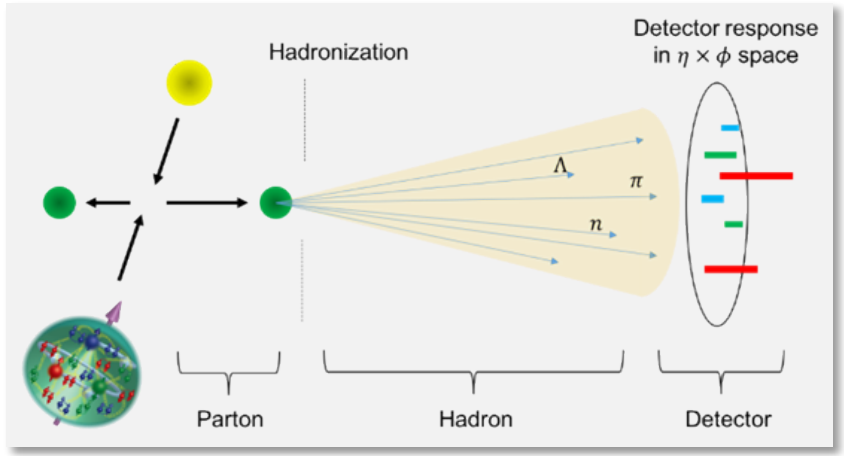
- Use TRNG Bernoulli devices to push random computation today onto hardware.
- Build devices with use case in mind while improving **speed and energy** of probabilistic computing applications.
- First – perform uniform TRNG draw.



"Probabilistic neural computing with stochastic devices" Misra et al., *Advanced Materials*. 2023

Random Numbers Are a Limiting Cost in Applications





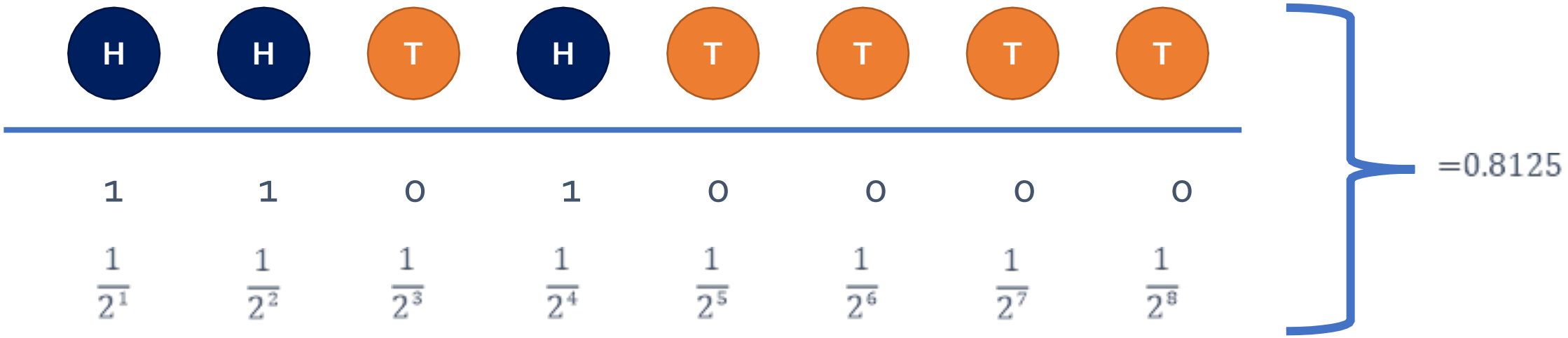
Up to half of compute time can be spent just generating uniform random numbers.

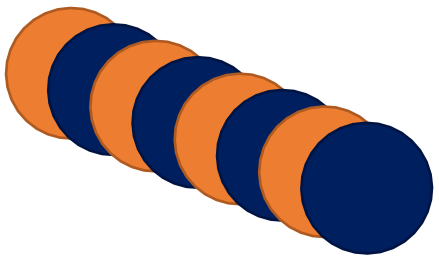
"Probabilistic neural computing with stochastic devices" Misra et al., *Advanced Materials*. 2023

From COINFLIPS to Distributions

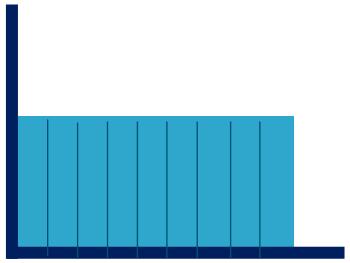
Drawing a Uniform Random Number

Suppose we have a fair coin.





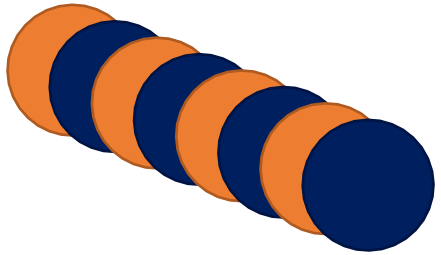
$\sim$



→ Uniform in ∞-coin limit.

Drawing a Random Number from an Unfair Coin

What if your coin is biased?



- Let $x_i \in \{0,1\}$ with appropriate probability.
- Set

$$X_n = \sum_{i=1}^{n} \frac{x_i}{2^i}$$

- Then, the probability mass function is

$$g_n(X_n) = p_0^{n-\sum_{i=1}^{n} x_i} p_1^{\sum_{i=1}^{n} x_i}$$

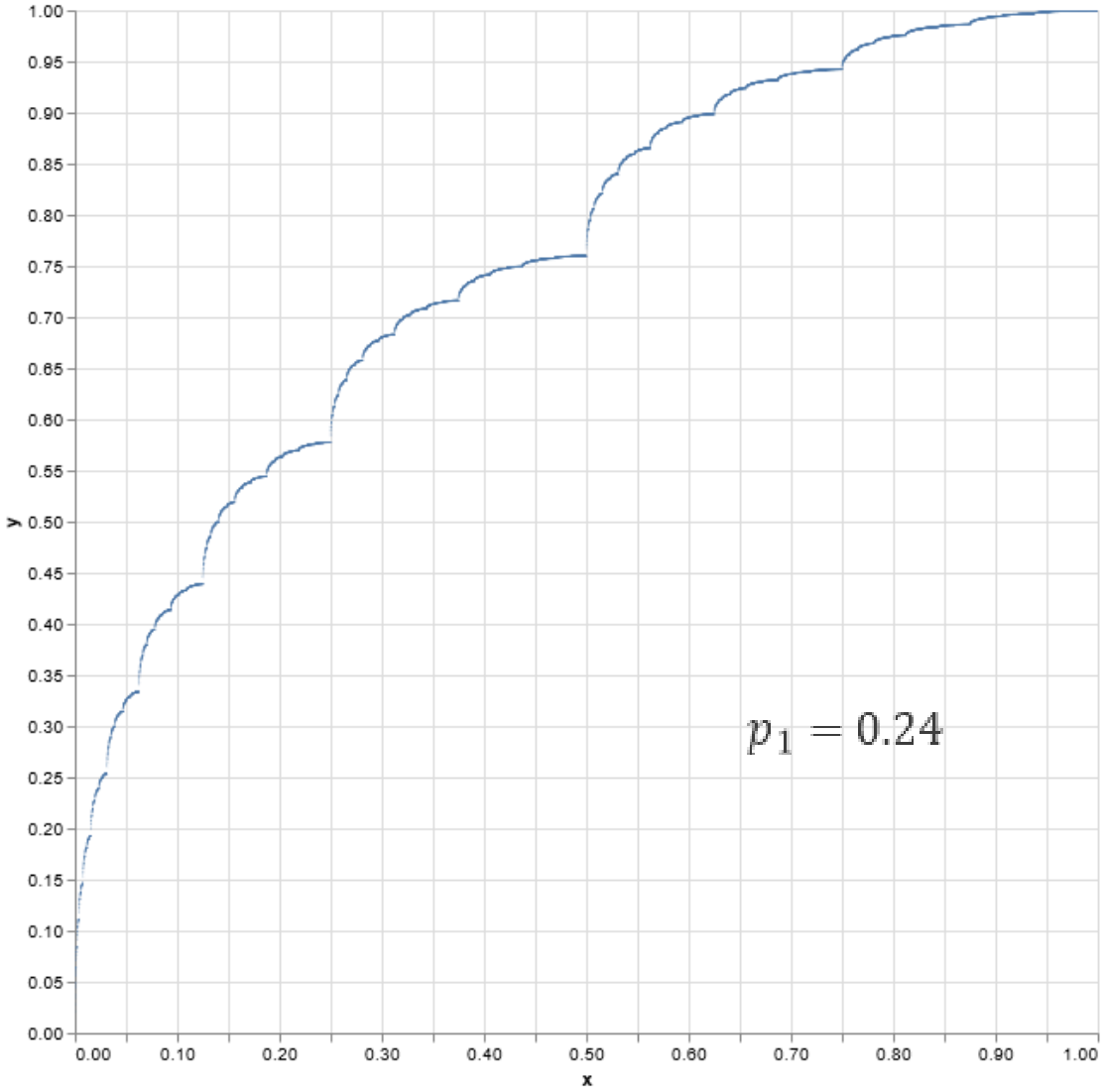
- Extend to a pdf on the real line.
- Let $x \in [0,1]$ and let $b_{n,x}$ be the first n digits in the binary expansion of x.
- Set $\sum b_{n,x}$ to be the number of 1's in $b_{n,x}$.

$$f_n(x) = 2^n p_0^{n-\sum b_{n,x}} p_1^{\sum b_{n,x}}$$

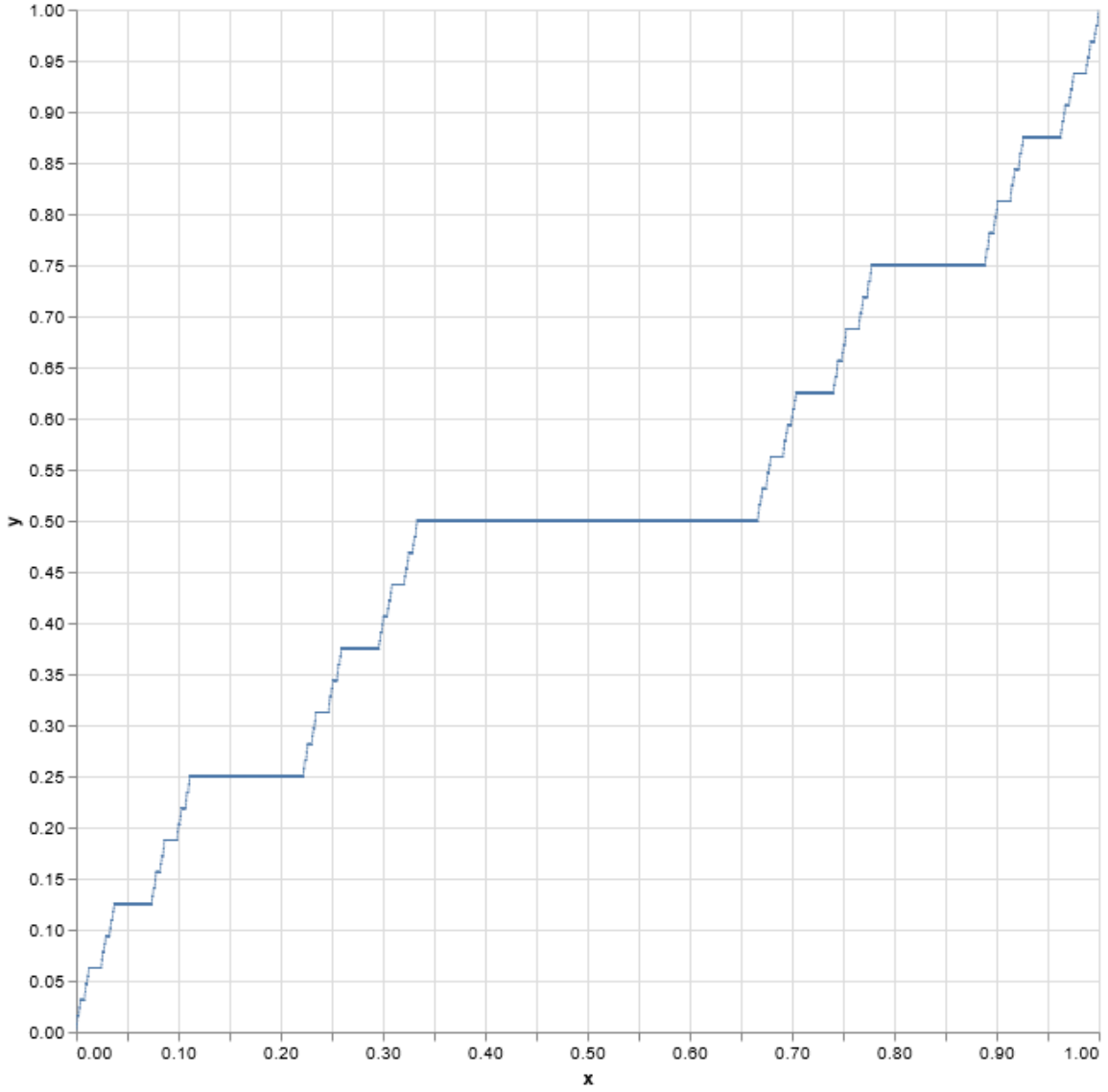
$$p_0 \neq \frac{1}{2} \Rightarrow f_n \to 0$$

Draw a Random Number from an Unfair Coin

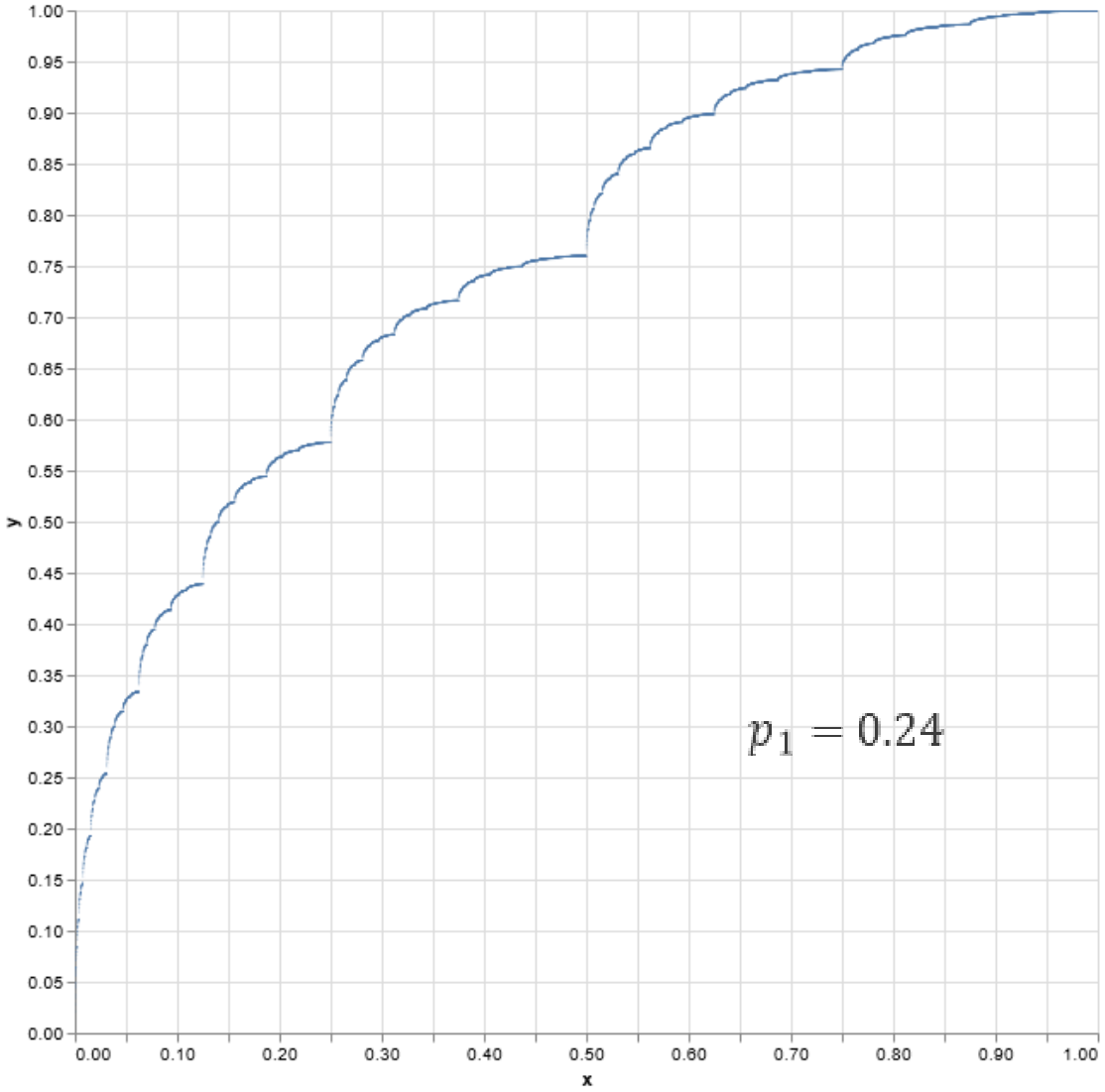
$$F(x) = \begin{cases} p_0 F(2x) & 0 \le x \le \frac{1}{2} \\ p_0 + p_1 F(2x-1) & \frac{1}{2} \le x \le 1 \end{cases}$$

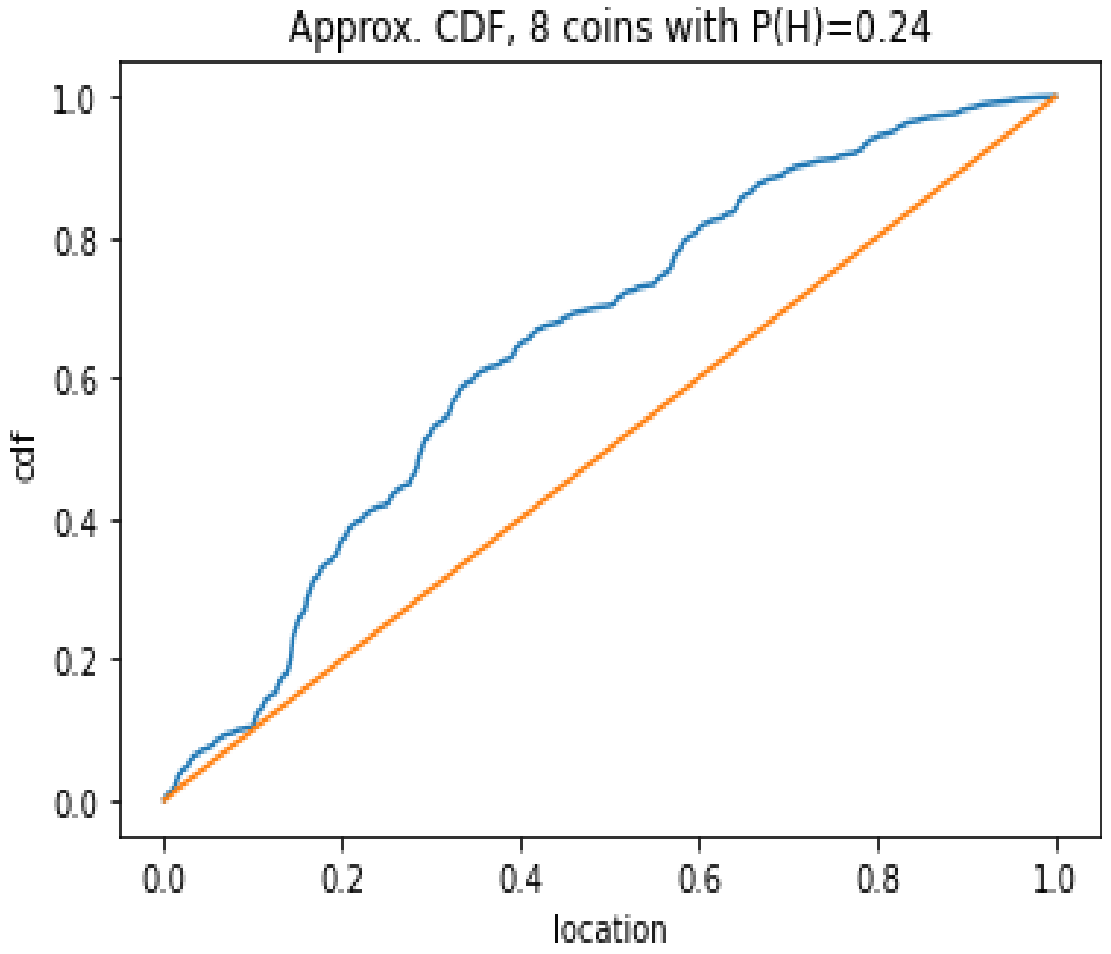


Quick Fun Fact – The Devil's Staircase



Draw a Random Number from an Unfair Coin

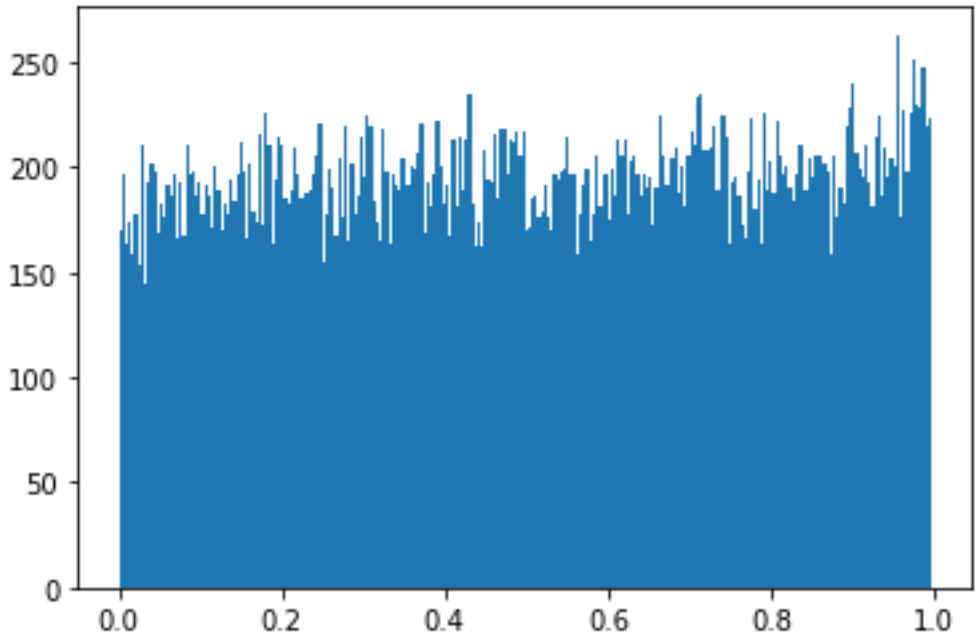




How safe are biased coins?

8-bit draws	*p*-value	Pass/Fail at 5-σ
1000	0.853	Pass
5000	0.631	Pass
10,000	0.119	Pass
20,000	0.010	Pass
50,000	1.07e-4	Pass
50,000	2.17e-15	Fail

$p_1 = 0.51$



How many samples can I draw?

$$\mathbb{E}[N] = \frac{\chi_{crit} + 2^b \sum_{j=0}^{b} \binom{b}{j} \left(p_1^j p_0^{b-j} - \frac{1}{2} \right)^2 - \frac{2^{2b}}{4}}{2^b \sum_{j=0}^{b} \binom{b}{j} \left(p_1^j p_0^{b-j} - \frac{1}{2^b} \right)^2}$$

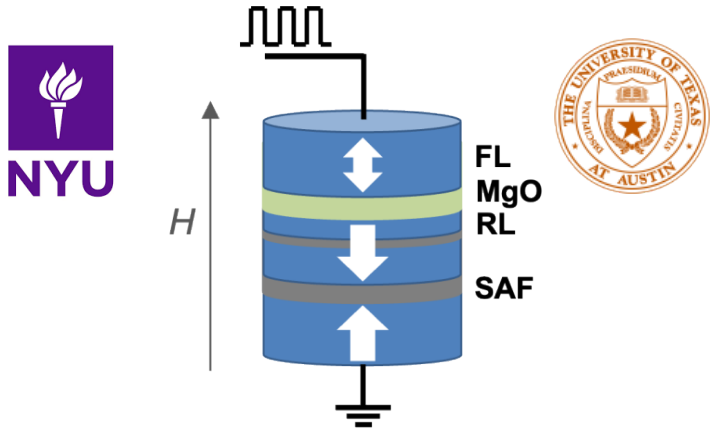
- χ_{crit} is dependent on your significance level and bits.
- b is the number of bits.
- p_1 is the probability of heads.

$$\mathbb{V}ar[N] \Rightarrow c = \mathbb{E}\left[\left(\sum_{j=1}^{2^b} \frac{\left(\sum_{k=1}^{N} 1_{x_j}(\tau_k) - N/2^b \right)^2}{\left(N/2^b \right)^2} \right)^2 \right]$$

p_1	$\mathbb{E}[N]$
0.55	1558
0.51	40,275
0.501	4,033,069
0.5001	403,312,444
0.50001	40,331,249,944
0.5	∞

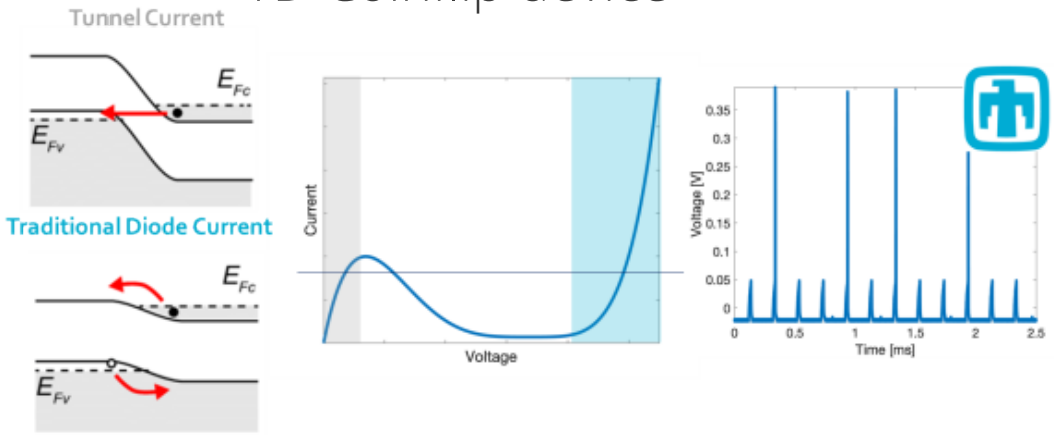
Devices and Uncertain Bias

MTJ Coinflip device



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

TD Coinflip device



- Devices may not fit into an idealized setting.
- Can we categorize a device with random effects that can make use of our developed methodologies?

Coins with Uncertain Bias

Coins with Uncertain Bias

$+ \ \varepsilon$ $\quad (\varepsilon \neq f(t))$

Two cases:

- Realization of ε is fixed, but random across devices/coins.
- Realization of ε changes for a single coin from use to use.

Flipping a single coin repeatedly:

- For fixed ε, we gain the bias of ε.
- For random, we gain the mean of ε on average.

$\mathbb{E}[N]$

$\mathbb{V}ar[N]$

Flipping a different coin every time:

- For fixed ε per coin, you get 2^b unique bin probabilities.
- For iid, we gain the mean of ε on average.

Reducing Mean Bias

Coin	Probability
0	$0.5 + \varepsilon$
1	$0.5 - \varepsilon$

Coins	Probability	XOR	Probability
0 1	$0.25 - \varepsilon^2$	1	$0.5 - 2\varepsilon^2$
1 0	$0.25 - \varepsilon^2$		
0 0	$0.25 + 2\varepsilon + \varepsilon^2$	0	$0.5 + 2\varepsilon^2$
1 1	$0.25 - 2\varepsilon + \varepsilon^2$		

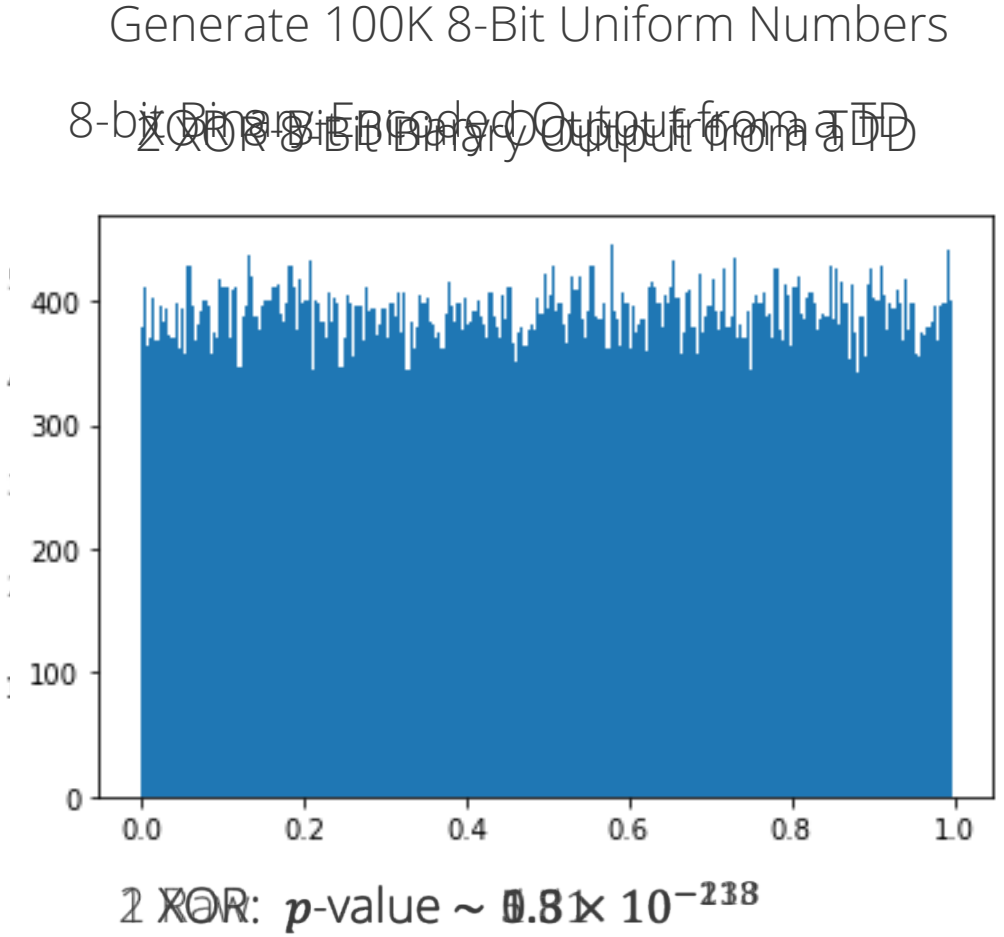
Two XORs!

p_1	$_m[N]$
0.55	1558
0.505	161,269
0.50005	1,613,249,944

- Whether using average bias contribution or a fixed repeated bias, a single XOR reduces bias.
- For fixed coin-to-coin biases, more analysis is needed.

Real Data from a Tunnel Diode Device

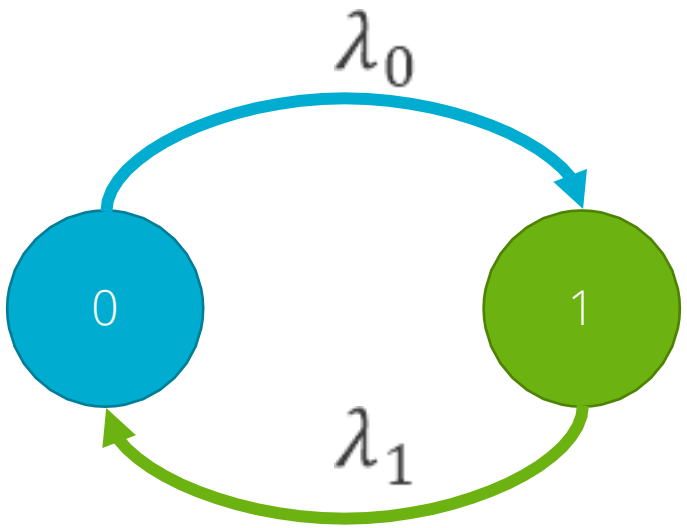
- Real devices probably do not behave in those nice modes.
- Nonetheless, an XOR application can still remove or lessen average deviations from fair.

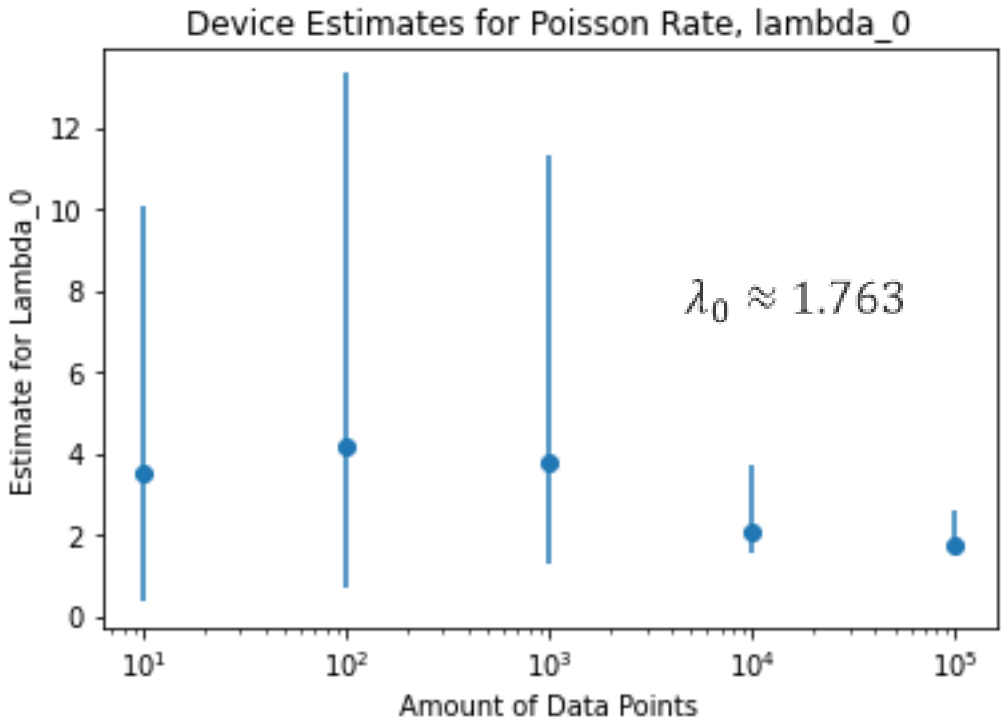


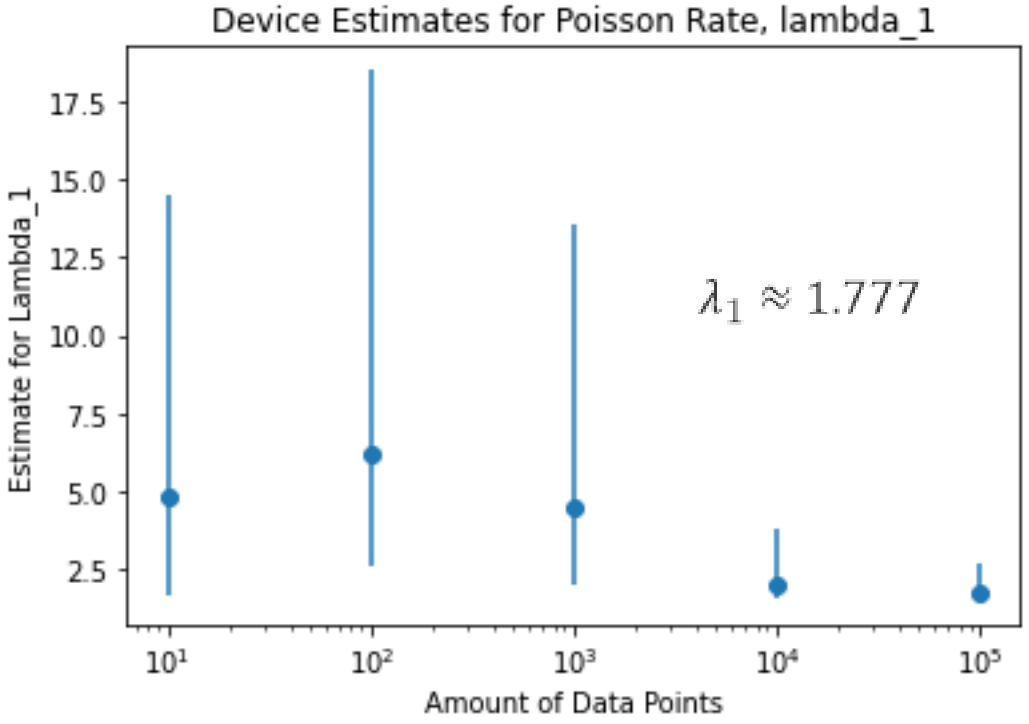
2 XOR Stream at 100K Bits passes our significance test.

Is a Coin the Best Model for the Device?

- Being a reliable coin is probably unrealistic.
- Can we model the device state as a Poisson arrival process?







Is the Coin the Best Model for the Device?

$$\mathbb{P}[0\to 0;\Delta t] = \frac{\lambda_1}{\lambda_0+\lambda_1} + \frac{\lambda_0}{\lambda_0+\lambda_1} e^{-(\lambda_0+\lambda_1)\Delta t}$$

$$\mathbb{P}[1\to 1;\Delta t] = \frac{\lambda_0}{\lambda_0+\lambda_1} + \frac{\lambda_1}{\lambda_0+\lambda_1} e^{-(\lambda_0+\lambda_1)\Delta t}$$

$$\mathbb{P}[\text{same};\Delta t] = \frac{1}{2} + \frac{1}{2} e^{-2\lambda\Delta t}$$

- Device could be described as a random variable that has a preference of staying the same value.

- Given this preference, how can we correct?
- Can we correct with a simple circuit?

Flipping the Output

$$\mathbb{P}[1 \to 1] = r$$
$$\mathbb{P}[1 \to 0] = 1 - r$$
$$\mathbb{P}[0 \to 0] = r$$
$$\mathbb{P}[0 \to 1] = 1 - r$$

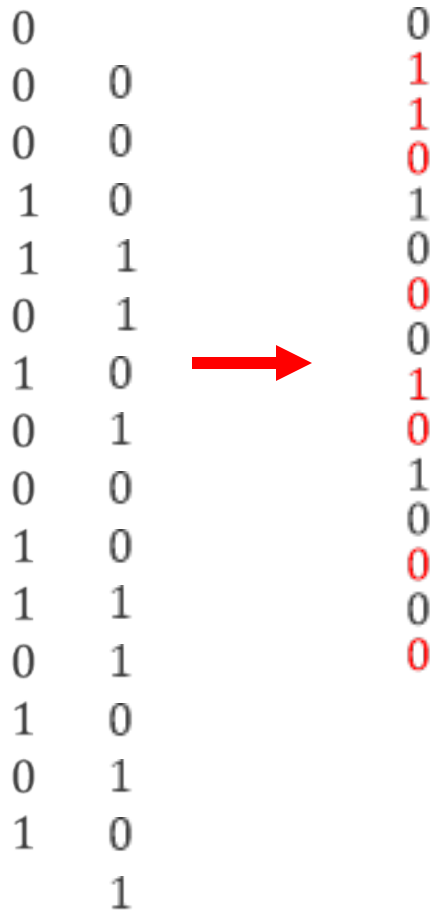
Let x_i represent the device output and o_i represent an observed output.

$$\mathbb{P}[o_i = 1 | x_{i-1} = 1] = \mathbb{P}[x_i = 1 | x_{i-1} = 1] = r$$
$$\mathbb{P}[o_i = 1 | x_{i-1} = 0] = \mathbb{P}[x_i = 0 | x_{i-1} = 0] = r$$

x: 0 0 0 1 1 0 1 0 0 1 1 0 1 0 1

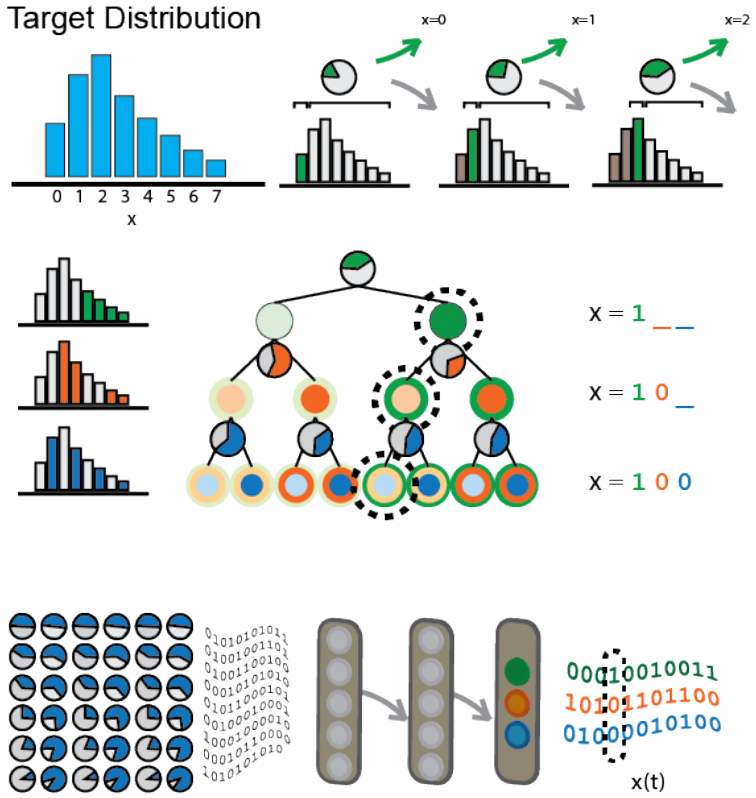
o: 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0

XAND

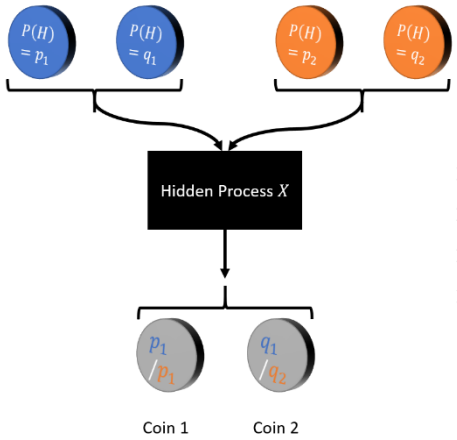


Once XANDed to remove dependence, you can reduce bias through repeated XORs.

Future and Ongoing Work

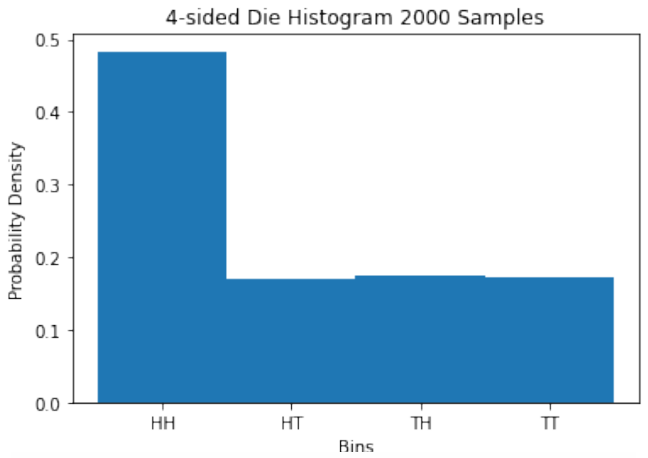


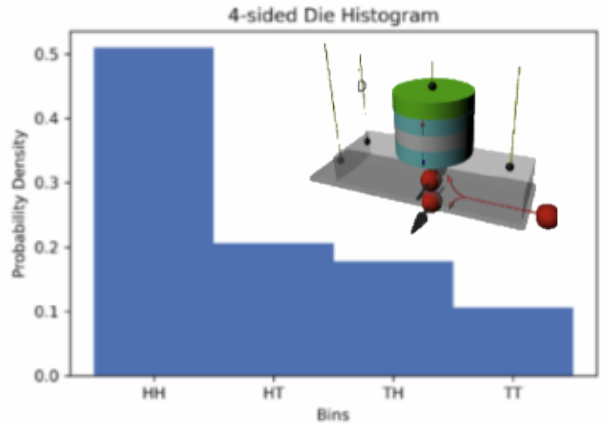
"Probabilistic neural computing with stochastic devices" Misra S et al. Advanced Materials. 2023

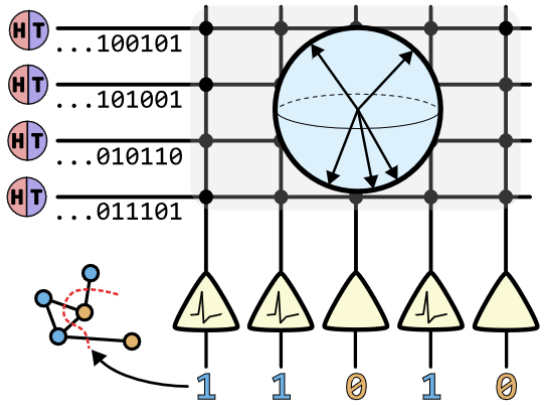


$$\mathbb{P}[\text{Coin 1} = H \text{ and Coin 2} = H] = \frac{1}{2}$$
$$\mathbb{P}[\text{Coin 1} = H \text{ and Coin 2} = T] = \frac{1}{6}$$
$$\mathbb{P}[\text{Coin 1} = T \text{ and Coin 2} = H] = \frac{1}{6}$$
$$\mathbb{P}[\text{Coin 1} = T \text{ and Coin 2} = H] = \frac{1}{6}$$

"AI-enhanced Codesign for Probabilistic Neural Circuits," Cardwell SG et al. International Conference on Rebooting Computing (ICRC) 2022.







"Stochastic Neuromorphic Circuits for Solving MAXCUT" Theilman, BH et al. International Parallel and Distributed Processing Symposium (IPDPS) 2023. Accepted.

Thanks!!

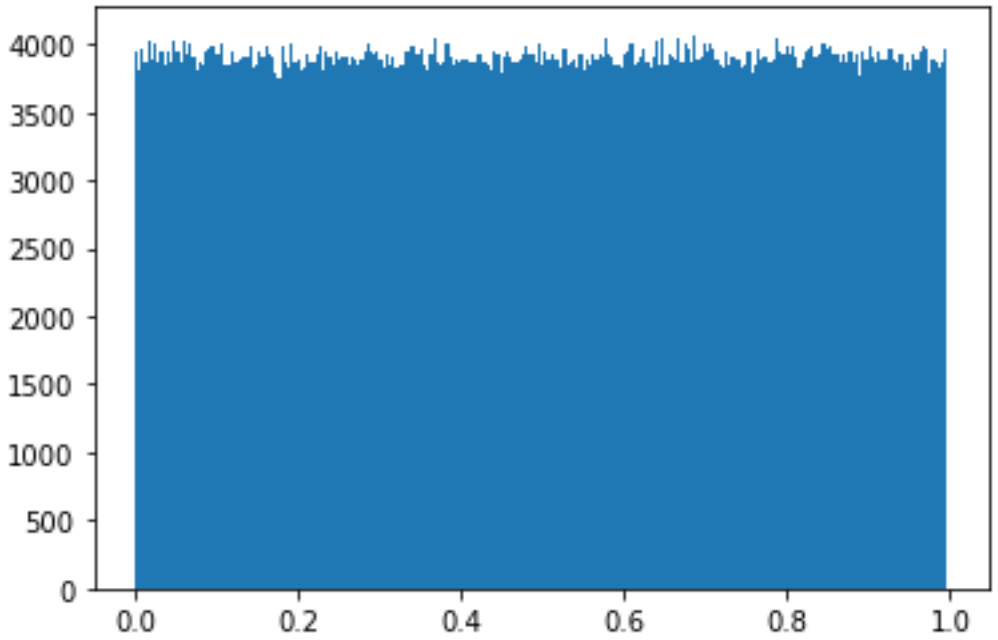
- Office of Science Co-Design in Microelectronics program
  - Co-funded through ASCR and BES, participation by NP, HEP, and FES
- COINFLIPS is partnering with a growing number of organizations
  - Andy Kent @ New York University
  - Jean Anne Incorvia @ University of Texas Austin
  - Katie Schuman @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

- Check us out at https://coinflipscomputing.org

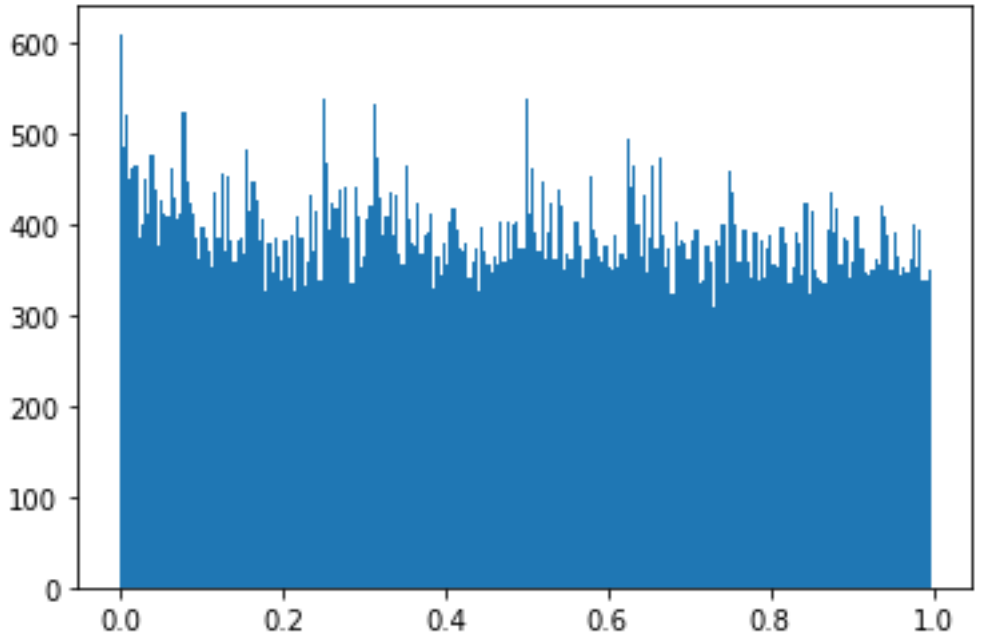
Is it reasonable to think fair coins are unambiguously uniform?

8-bit draws	*p*-value	Pass/Fail at 99.5%
1000	0.770	Fail
10,000	0.382	Fail
100,000	0.490	Fail
1,000,000	.829	Fail

$p_1 = 0.50$



Applying XAND then XOR to TD Data



p-value: 3.83×10^{-134}

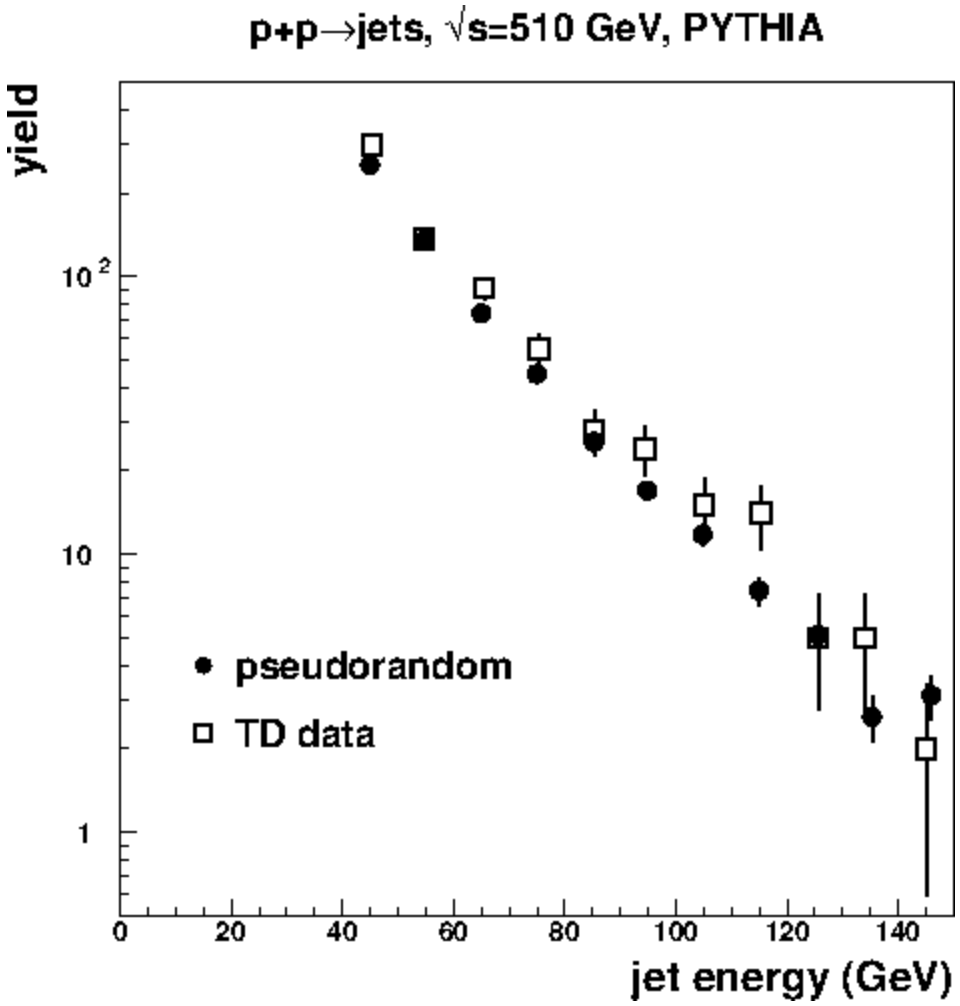
(Raw value was 5.5×10^{-238}

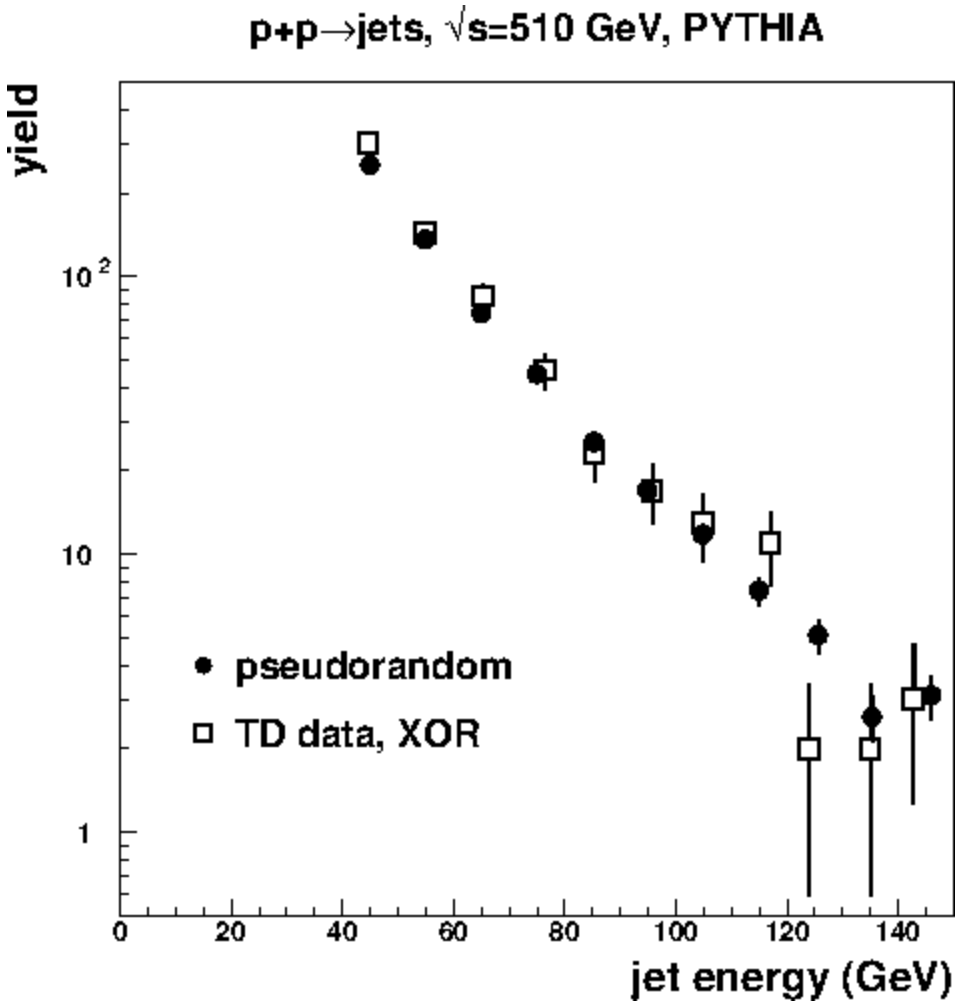
Adder, No Carry

- Treat dependent nature of flips as fair, $\pm \varepsilon$.
- Add three subsequent bits with no carry.
- You can reduce the dependence, but not remove it.
- Iterations will decrease this dependence like repeated XOR reduces bias.

Previous Coin	Next Sequence	Added Value	Probability	Total Probability	$r = 0.5 + \varepsilon$
0	000	0	r^3	$r^2 + (1-r)^2$	$0.5 + 2\varepsilon^2$
	011		$r^2(1-r)$		
	101		$(1-r)^3$		
	110		$r(1-r)^2$		
	001	1	$r^2(1-r)$	$2r(1-r)$	$0.5 - 2\varepsilon^2$
	010		$r(1-r)^2$		
	100		$r(1-r)^2$		
	111		$r^2(1-r)$		
1	000	0	$r^2(1-r)$	$2r(1-r)$	$0.5 - 2\varepsilon^2$
	011		$r(1-r)^2$		
	101		$r(1-r)^2$		
	110		$r^2(1-r)$		
	001	1	$r(1-r)^2$	$r^2 + (1-r)^2$	$0.5 + 2\varepsilon^2$
	010		$(1-r)^3$		
	100		$r^2(1-r)$		
	111		r^3		

Use of TD Data in Real Application





Expected N – Proof, sketch

- Let N be the average number of observations before failing the hypothesis test.
- As N is an (average) number of observations, N must be at least 1. (In all cases an observation must be made before drawing a conclusion, therefore the average number of observations before failure must be at least 1.)
- Let χ_{crit} be the chosen critical value of the χ^2 hypothesis test. This value will be dependent on a desired significance level and on the number of coinflips/number of bits in the expansion. The number of bits determines the degrees of freedom.
- Let $\{\tau_1, \tau_2, \dots\}$ be the outcome string of b flips of a coin with $\mathbb{P}[1] = p_1$, $p_0 = 1 - p_1$. Let $H\left(x_j, \{\tau_i\}, N\right)$ be the value of the histogram in the j^{th} bin after counting the first N draws of $\{\tau_i\}$.
- Letting 1_{x_j} be the indicator function of the j^{th} bin,

$$H(x_k, \{\tau_i\}, N) = \sum_{\ell=1}^{N} 1_{x_j}(\tau_\ell)\,.$$

- Note, $\{\tau_i\}$ is an iid random variable collection independent of N. This makes H a function of many random variables.

Expected N – Proof, sketch (cont)

- To calculate $\mathbb{E}[N]$, we will take the expected value at the moment the χ^2 test statistic equals the chosen critical value.
- For bin value x_j, let n_j represent the number of 1's in the binary decimal representative of the j^{th} bin.
- The proof rests on two lemmas. First:

$$\mathbb{E}\left[H\left(x_j, \{\tau_i\}, N\right) \middle| N\right] = N\left(p_1^{n_j} p_0^{b-n_j}\right)$$

- Second:

$$\mathbb{E}\left[\left(H\left(x_j, \{\tau_i\}, N\right)\right)^2 \middle| N\right] = N\left(p_1^{n_j} p_0^{b-n_j}\right) + N(N-1)\left(p_1^{n_j} p_0^{b-n_j}\right)^2$$

- Observe:

$$\mathbb{E}\left[\frac{\left(H\left(x_j, \{\tau_i\}, N\right) - N/2^b\right)^2}{N/2^b}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{\left(H\left(x_j, \{\tau_i\}, N\right) - N/2^b\right)^2}{N/2^b} \middle| N\right]\right]$$

$$= \mathbb{E}\left[\frac{2^b}{N}\mathbb{E}\left[\left(H\left(x_j, \{\tau_i\}, N\right)\right)^2 \middle| N\right] - 2\mathbb{E}\left[H\left(x_j, \{\tau_i\}, N\right) \middle| N\right] + \frac{N}{2^b}\right]$$

Expected N – Proof, sketch (cont)

- Hence:

$$\mathbb{E}\left[\frac{\left(H\left(x_j,\{\tau_i\},N\right)-N/2^b\right)^2}{N/2^b}\right]=\mathbb{E}\left[2^bN\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2^b}\right)^2-2^b\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2}\right)^2+\frac{2^b}{4}\right]$$

$$=2^b\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2^b}\right)^2\mathbb{E}[N]-2^b\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2}\right)^2+\frac{2^b}{4}$$

- Taking the expected value for the χ^2 test statistic at the critical value yields

$$\chi_{crit}=\sum_{j=1}^{2^b}\mathbb{E}\left[\frac{\left(H\left(x_j,\{\tau_i\},N\right)-N/2^b\right)^2}{N/2^b}\right]$$

$$=\sum_{j=1}^{2^b}2^b\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2^b}\right)^2\mathbb{E}[N]-2^b\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2}\right)^2+\frac{2^b}{4}$$

$$=2^b\mathbb{E}[N]\sum_{j=0}^{b}\binom{b}{j}\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2^b}\right)^2-2^b\sum_{j=0}^{b}\binom{b}{j}\left(p_1^{n_j}p_0^{b-n_j}-\frac{1}{2}\right)^2+\frac{2^{2b}}{4}$$

Expected N – Proof, sketch (cont)

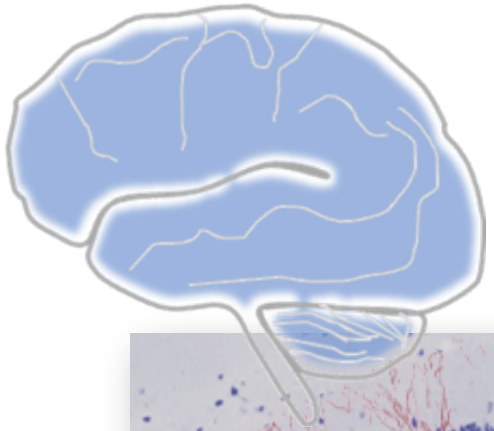
- Ergo, we can represent $\mathbb{E}[N]$ as a function of χ_{crit}:

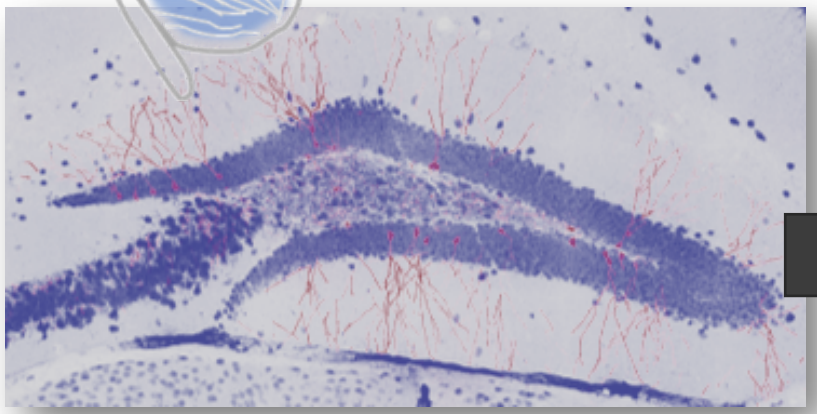
$$\mathbb{E}[N] = \frac{\chi_{crit} + 2^b \sum_{j=0}^{b} \binom{b}{j} \left(p_1^{n_j} p_0^{b-n_j} - \frac{1}{2} \right)^2 - \frac{2^{2b}}{4}}{2^b \sum_{j=0}^{b} \binom{b}{j} \left(p_1^{n_j} p_0^{b-n_j} - \frac{1}{2^b} \right)^2}$$

- Note that when $p_1 = p_0 = \frac{1}{2}$, the denominator collapses to zero. This is expected as when $p_1 = p_0 = \frac{1}{2}$ the histogram generated should be uniform. We are comparing to uniform hence we should never expect to fail the hypothesis test on average.

Lemmas – Idea

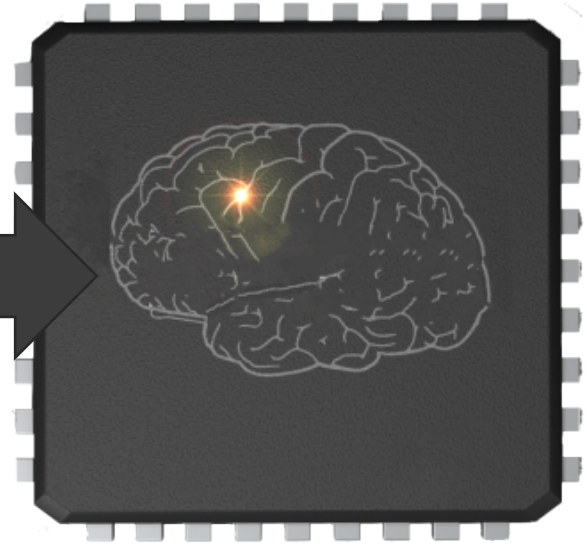
- Both Lemmas are proofs by induction on the value of N.
- The result for $\mathbb{E}[H|N]$ conforms to intuition.
- The proof for $\mathbb{E}[H^2|N]$ requires knowing $\mathbb{E}[H|N-1]$.
- While exceedingly tedious, both are rather straightforward and only require breaking up sums along the first $N-1$ possibilities for a sequence of draws.





Realized Features of Brain Inspiration in Neuromorphic Hardware

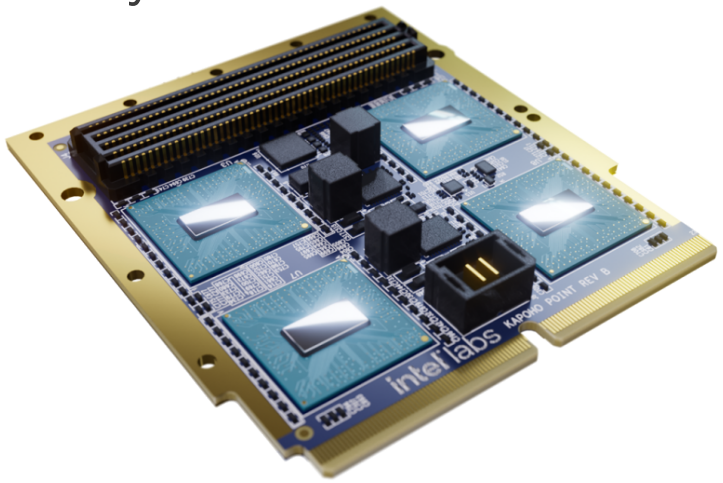
- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity



Realized Features of Brain Inspiration in Neuromorphic Hardware

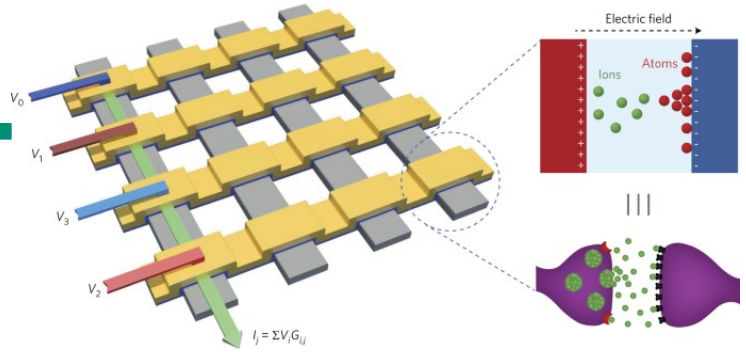
- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity

Today



Intel Loihi 2
Millions of CMOS neurons
Billions of CMOS synapses
~ 1 Watt power

Tomorrow



Zidan et al., 2018

Post-Moore Devices
(ECRAM, Memristors, MTJs, etc)

Scale to human sizes?