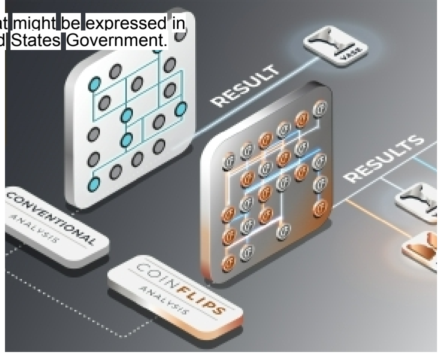
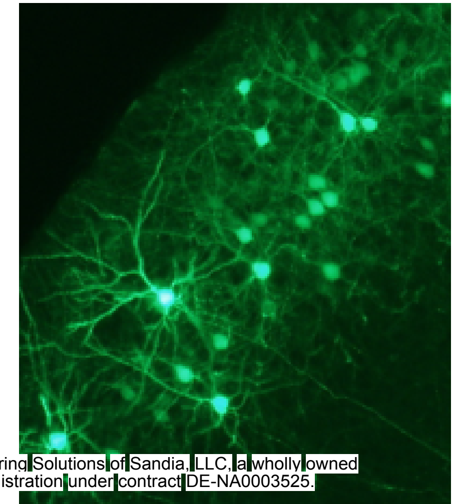
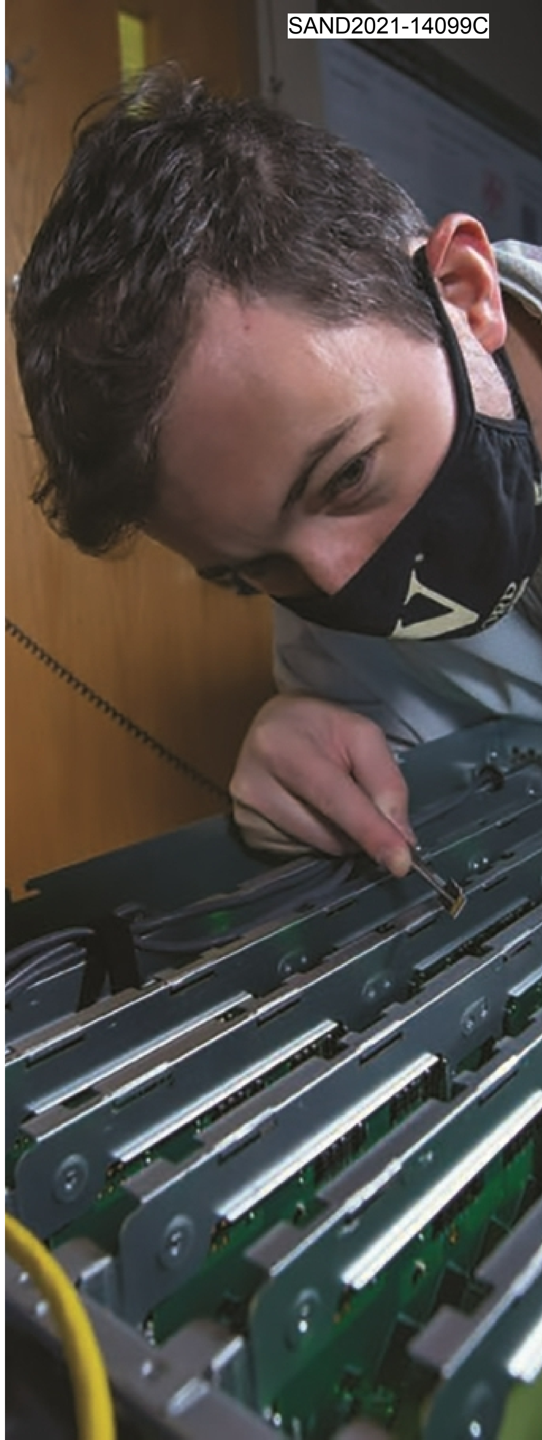
SAND2021-14099C

Sandia National Laboratories

Exceptional service in the national interest







Assessing a Neuromorphic Platform for use in Scientific Stochastic Sampling

J. Darby Smith, William Severa, J. Brad Aimone, and Rich Lehoucq

INTERNATIONAL CONFERENCE ON REBOOTING COMPUTING, 2021, *VIRTUAL*

NOVEMBER 30TH – DECEMBER 2ND

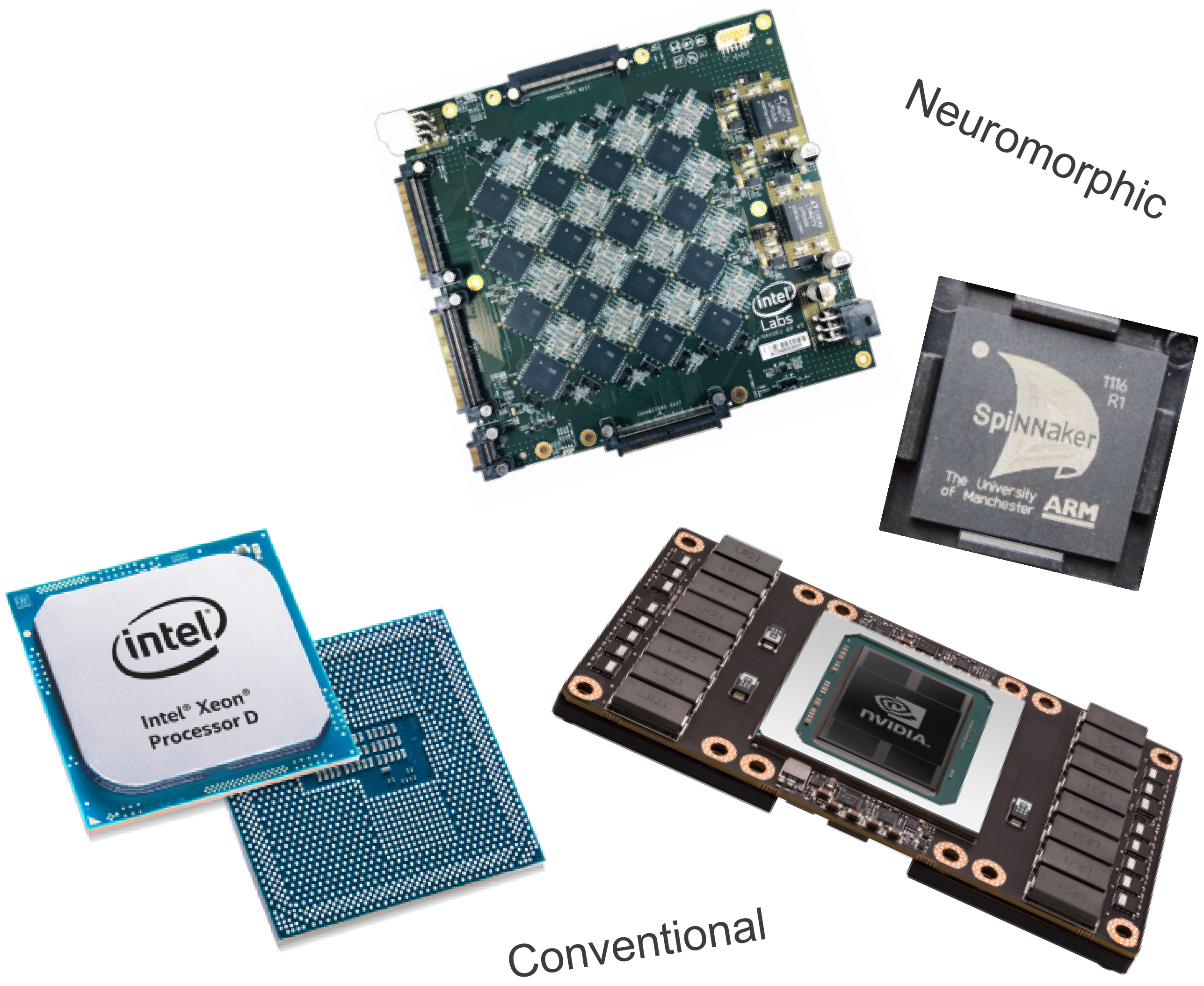
U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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.

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.

A Heterogeneous Future

- We envision a future of computing with neuromorphic hardware working side-by-side with CPUs and GPUs.
- Neuromorphic hardware are often used for AI applications and bio-inspired algorithms.
- The energy efficiency of neuromorphic also has potential for scientific computing applications.



Efficient Markov Chain Sampling

- Our algorithm has been shown to be effective at sampling from discrete time Markov chains (DTMCs).
- A random walk is interpreted as a trajectory from a DTMC.
- Walkers are spikes and are routed through nodes, a cluster of neurons representing the state space of the DTMC.
- For our Loihi algorithm, walkers transition from one node to one of N others through a mutually exclusive probability draw involving $N-1$ probability neurons.

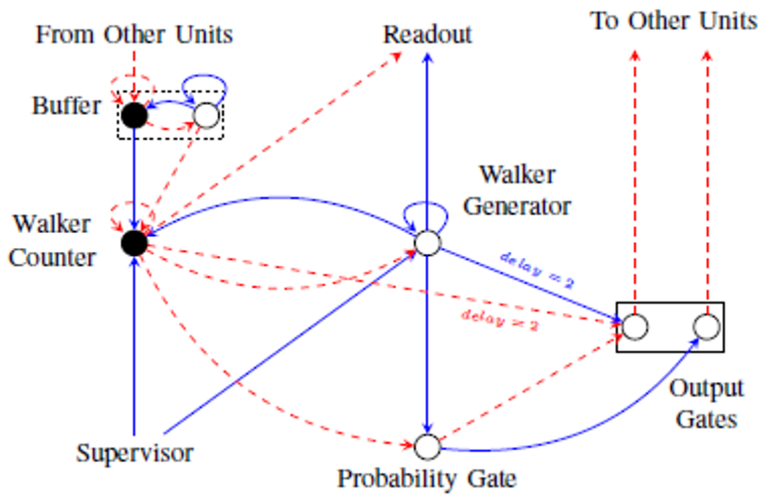


Image: *Severa et al. IJCNN 2018.*

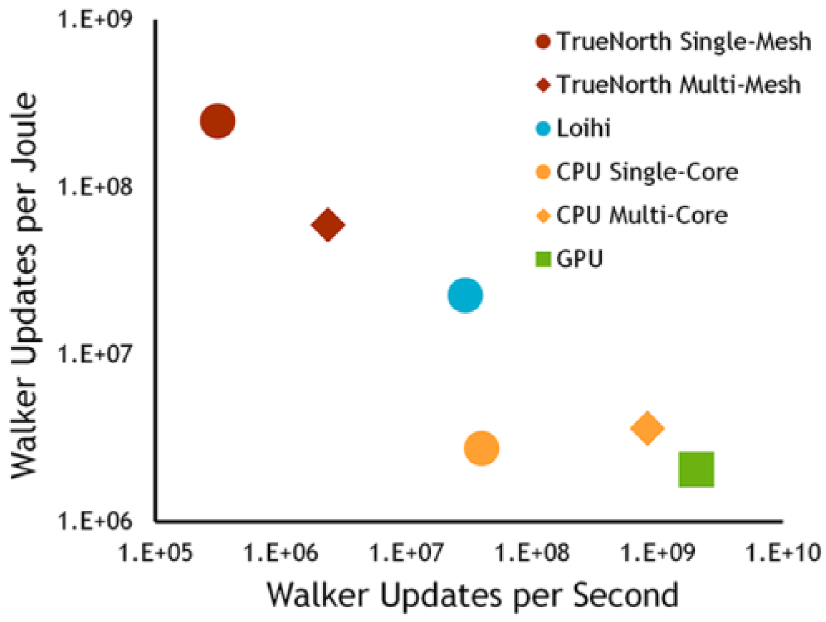
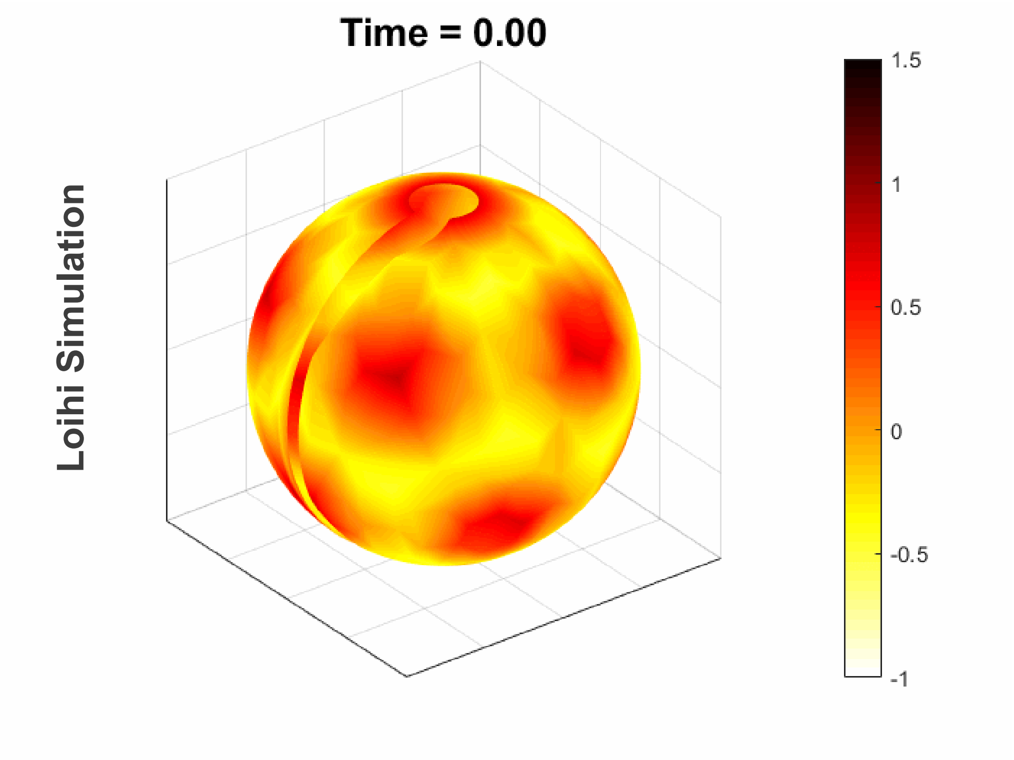
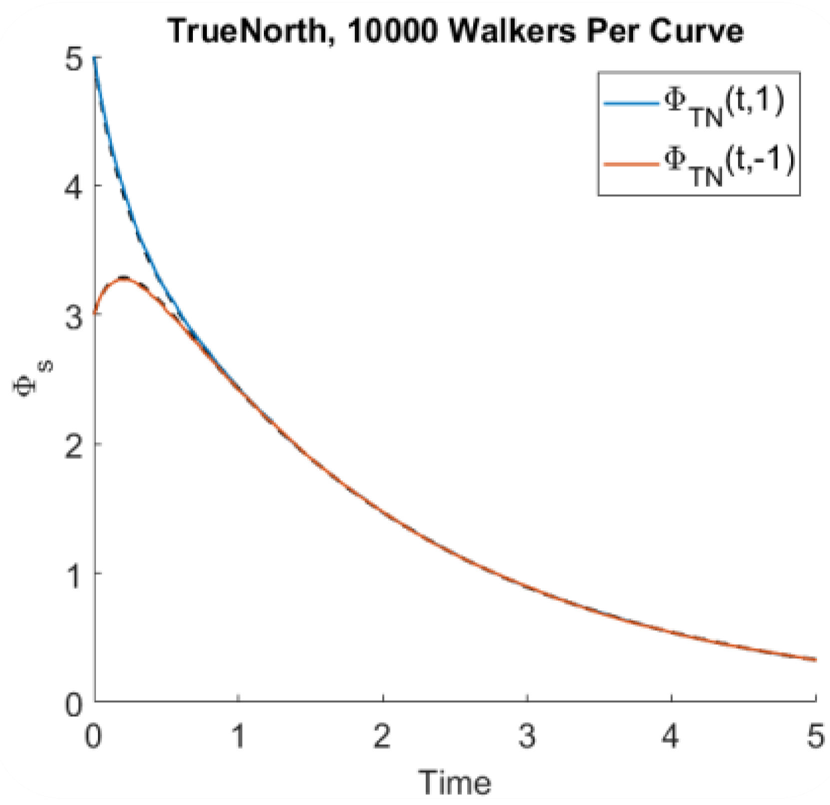


Image: *Smith et al., in review 2021.*

DTMC Sampling Can Solve Problems!

- Sampling can be used for many problems, including solving PDEs.





$$\frac{\partial}{\partial t}u(t,x,y,z) = \alpha\nabla u(t,x,y,z), \qquad (x,y,z) \in \mathcal{S}^2$$

$$u(0,x,y,z) = g(x,y,z)$$

$$\frac{\partial}{\partial t}\Phi(t,\Omega) = -(\sigma_a + \sigma_s)\Phi(t,\Omega) + \int \sigma_s \Phi(t,\Omega')\mathbb{P}(\Omega' \to \Omega)d\Omega'$$

$$\Phi(\Omega,0) = g(\Omega) = \begin{cases} 5 & \text{if } \Omega = 1 \\ 3 & \text{if } \Omega = -1 \end{cases}$$

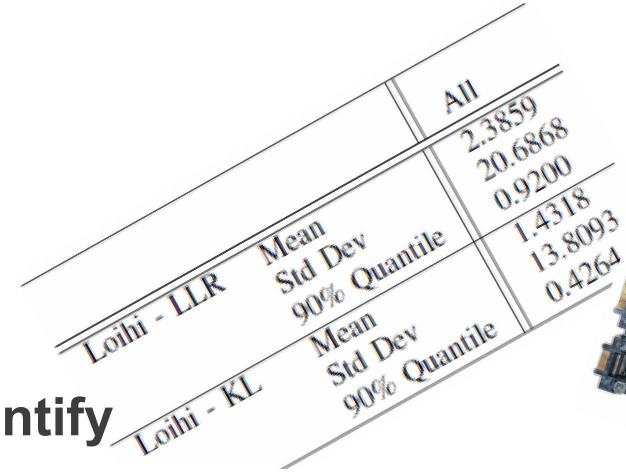
Images: *Smith et al., in review 2021.*

Evaluating a Sampling Algorithm on Loihi

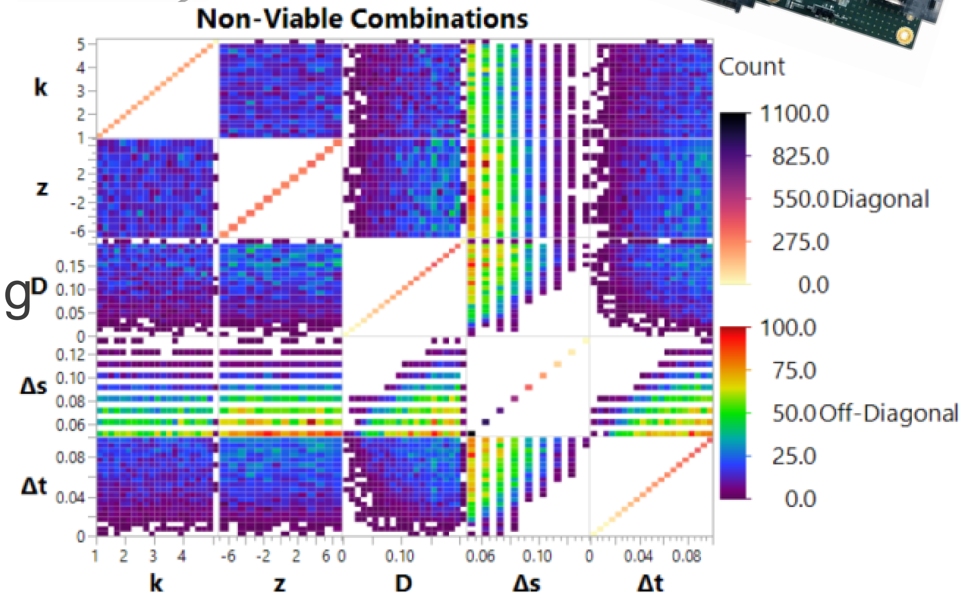
- Loihi has 8-bit limited PRNG. Our algorithm approximates a stochastic process with a DTMC.

- **How do we assess stochastic sampling and identify where hardware or algorithm limitations exist?**
  - Code verification and Hardware validation

- We test Loihi and our algorithm by statistically assessing the samples generated and exploring the parameter space where the algorithm may fail.



		All
Loihi - LLR	Mean	2.3859
	Std Dev	20.6868
	90% Quantile	0.9200
Loihi - KL	Mean	1.4318
	Std Dev	13.8093
	90% Quantile	0.4264



The Ornstein-Uhlenbeck Equation

- We choose to assess our Loihi algorithm on the Ornstein-Uhlenbeck Equation:

$$dX(t) = -k(X(t) - z)dt + \sqrt{2D}dW(t)$$

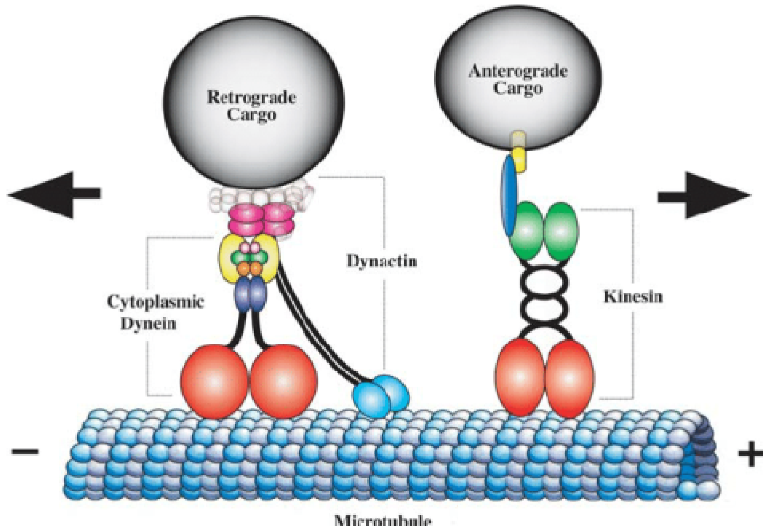


Image: *Duncan & Goldstein, PLoS genetics 2006.*

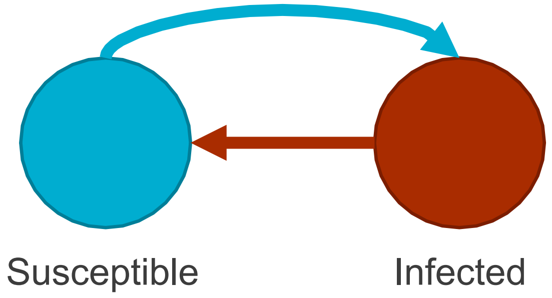
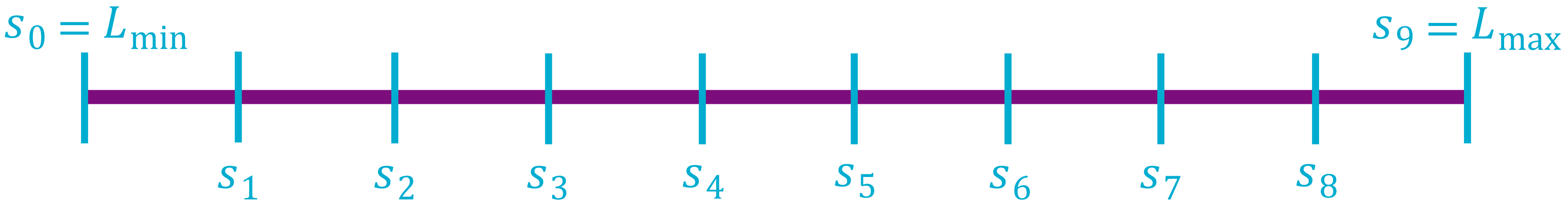


Image: *Creative Commons BY-SA-NC*

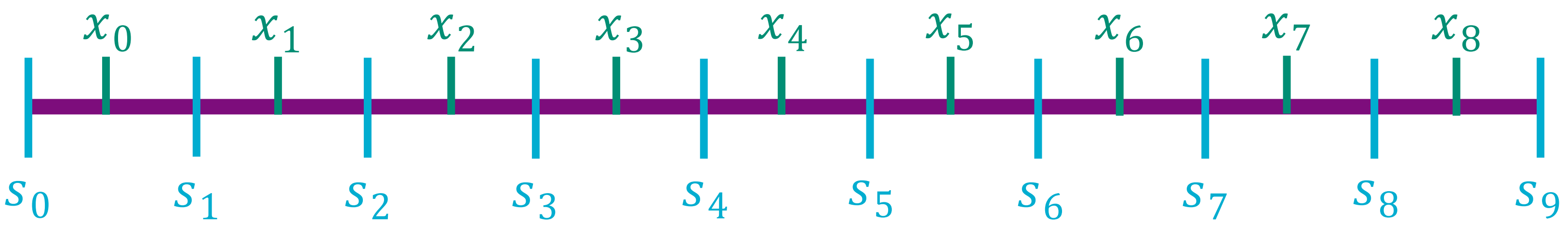
Making a DTMC from the OU Process

- Divide the real line into bins of size Δs.

- Truncate the line into an interval of interest with minimum $L_{\min}$ and maximum $L_{\max}$.



- Define the state space $\{x_i\}$ to consist of the midpoints of the bins.



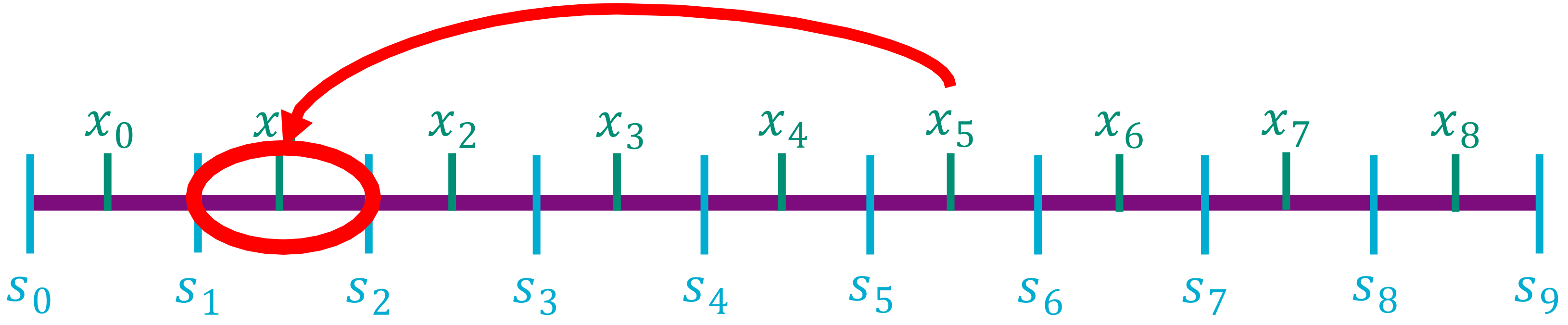
Making a DTMC from the OU Process

- Next, we choose a time step size Δt.

$$X(t+\Delta t) \sim \mathcal{N}(\underbrace{X(t) - k(X(t) - z)\Delta t}_{\text{Mean}}, \quad \underbrace{2D\Delta t}_{\text{Variance}})$$

- Using the Euler-Maruyama update scheme, p_{ij} can be calculated.

$$p_{ij} = \mathbb{P}\left[X(\Delta t) \in [s_j, s_{j+1}) \middle| X(0) = x_i\right]$$



Making a DTMC from the OU Process

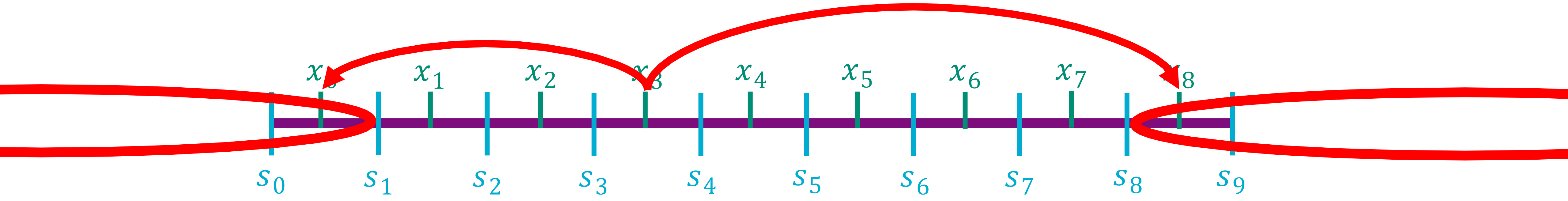
- Next, we choose a time step size Δt.

$$X(t + \Delta t) \sim \mathcal{N}(\underbrace{X(t) - k(X(t) - z)\Delta t}_{\text{Mean}}, \underbrace{2D\Delta t}_{\text{Variance}})$$

- Care must be taken with end cases!

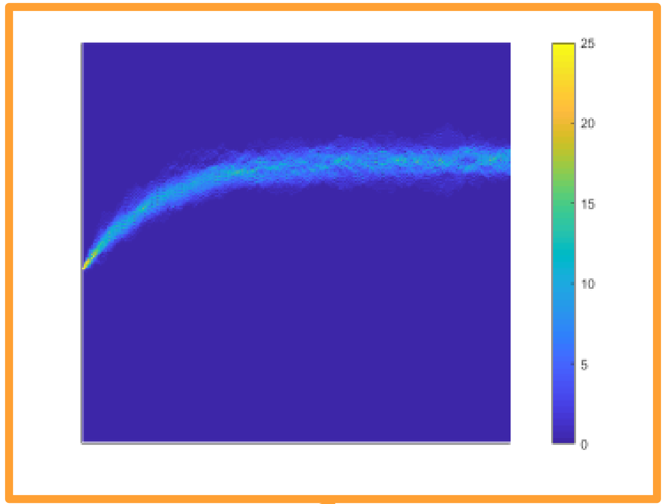
$$p_{i0} = \mathbb{P}[X(\Delta t) \in (-\infty, s_1) | X(0) = x_i]$$

$$p_{i8} = \mathbb{P}[X(\Delta t) \in [s_8, \infty) | X(0) = x_i]$$



Data Generation

- We generated 29,087 observations on Loihi by sampling observations from a DTMC given a collection of parameters $L_{\min}, L_{\max}, k, z, D, \Delta s$, and Δt.
- Observations are of M random walkers starting at location $X(0)$.



$X(0)$

M

$k, z, D, \Delta s, \Delta t, L_{\min}, L_{\max}$

$$P = \begin{bmatrix} p_{11} & \cdots & p_{18} \\ \vdots & \ddots & \vdots \\ p_{81} & \cdots & p_{88} \end{bmatrix}$$

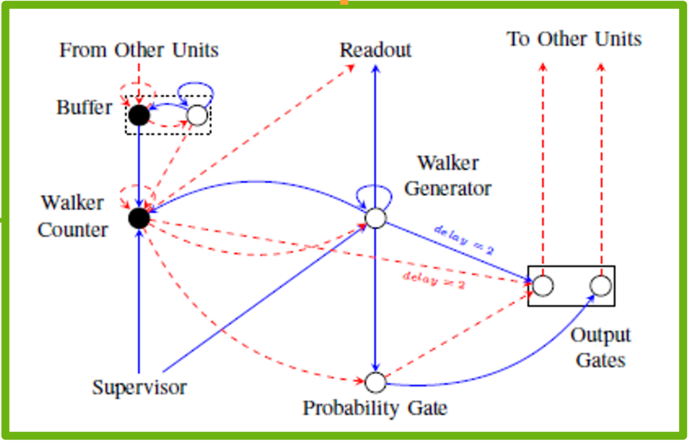


Image: *Severa et al. IJCNN 2018.*

Relative Entropy

- Relative entropy provides a way to measure the difference between two probability distributions p and q.
- The Kullback-Leibler (KL) Divergence:

$$\kappa(p, q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$$

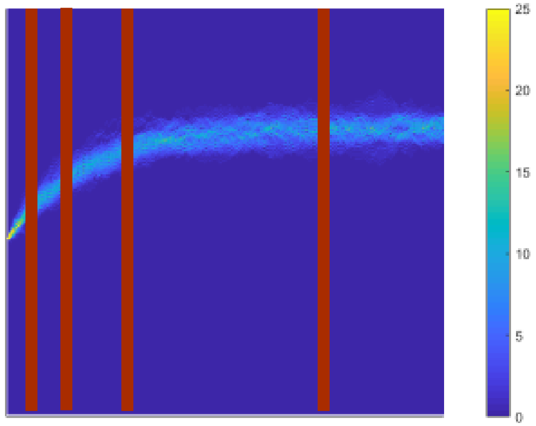
- We will let p represent the distribution of samples from Loihi and q represent the distribution of the OU process.
- We approximate κ in two ways:
  1. Approximate p through a histogram probability density and compare to q.
  2. Discretize q and compare to the Loihi walker counts.

Relative Entropy

- Note that the distribution is time dependent. For each t we have a different distribution.
- Therefore we will get a relative entropy measurement for each time point we simulate for each set of trajectories.

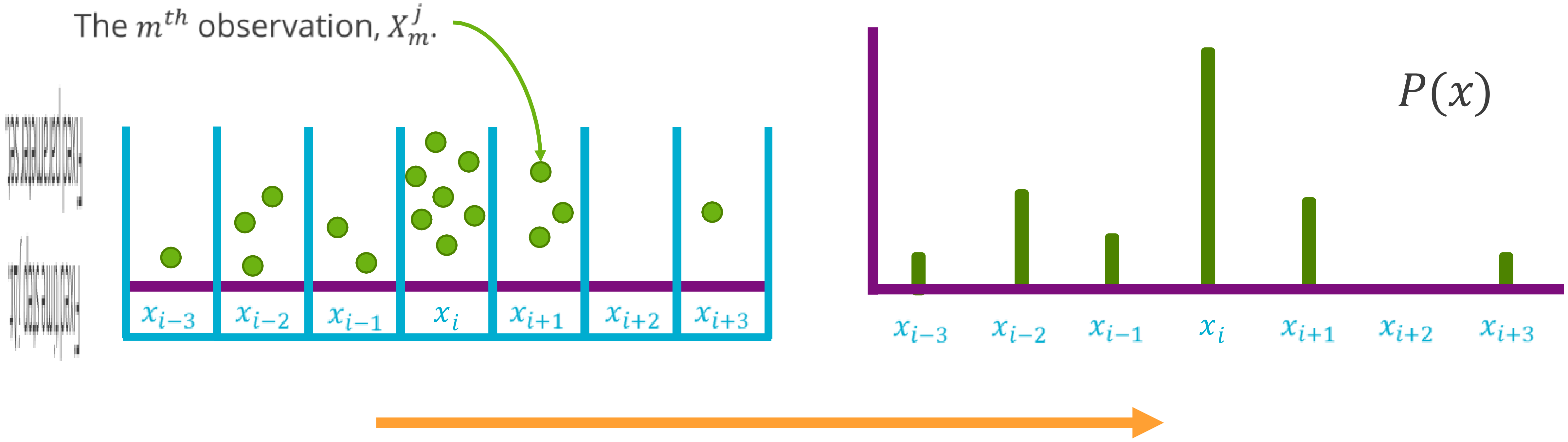
$$X(t) = X(0)e^{-kt} + z\left(1 - e^{-kt}\right) + \sqrt{2D}\int_0^t e^{-k(t-u)}\mathrm{d}W(u)$$

$$X(t) \sim \mathcal{N}\left(\underbrace{X(0)e^{-kt} + z\left(1 - e^{-kt}\right)}_{\text{Mean}}, \underbrace{\frac{D}{k}\left(1 - e^{-2kt}\right)}_{\text{Variance}}\right)$$



The Discrete KL Divergence

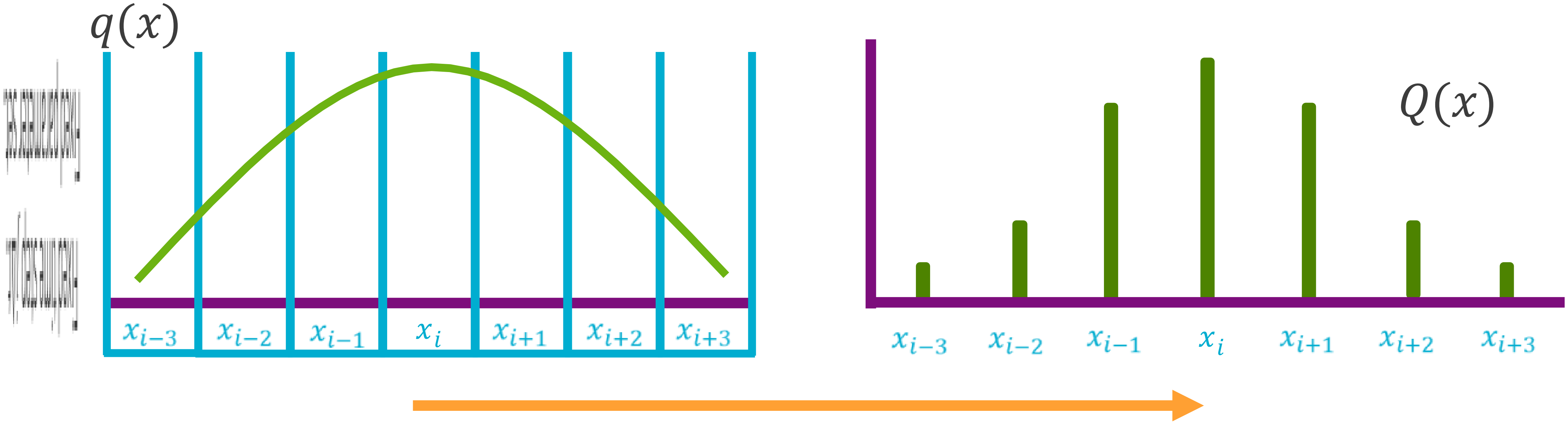
- To employ the discrete KL divergence, we make a discrete pmf $P(x)$ by dividing the number of walkers in each bin by M only.



$P(x)$ is constructed by dividing the number of walkers in each bin by the total number of walkers M

The Discrete KL Divergence

- To employ the discrete KL divergence, make a discrete pmf $P(x)$ by dividing the number of walkers in each bin by M only.
- Then, we construct a discrete $Q(x)$ by integrating $q(x)$ over the bins.



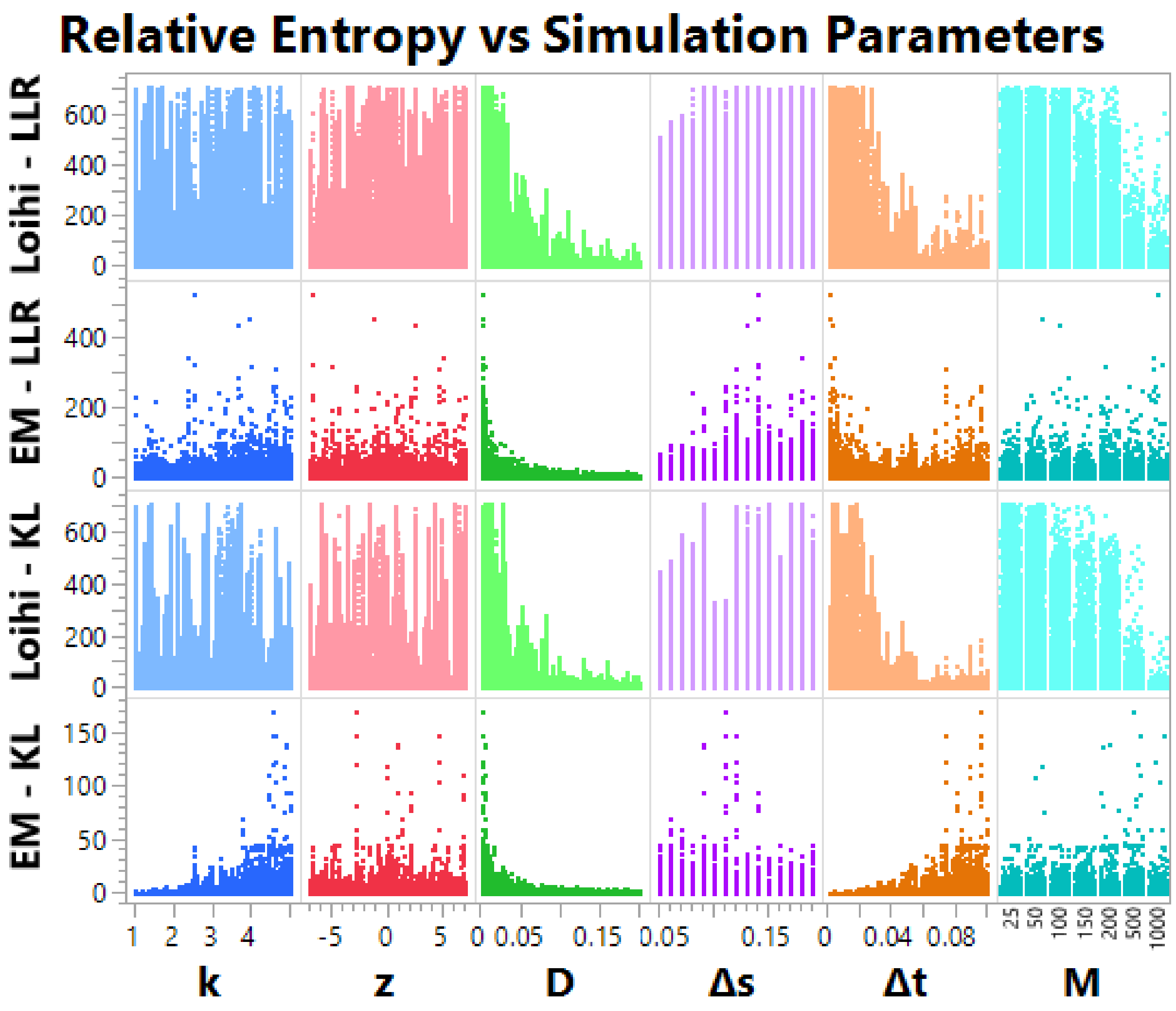
$Q(x)$ is constructed by integrating $q(x)$ over each bin and assigning the value to the new discrete distribution at the midpoints x_i.

The Discrete KL Divergence

- To employ the discrete KL divergence, we make a discrete pmf $P(x)$ by dividing the number of walkers in each bin by M only.
- Then, we construct a discrete $Q(x)$ by integrating $q(x)$ over the bins.
- With these in hand, we can define the discrete KL Divergence:

$$\tilde{\kappa}(P, Q) = \sum_{i=1}^{N} P(x_i) \log \frac{P(x_i)}{Q(x_i)}$$

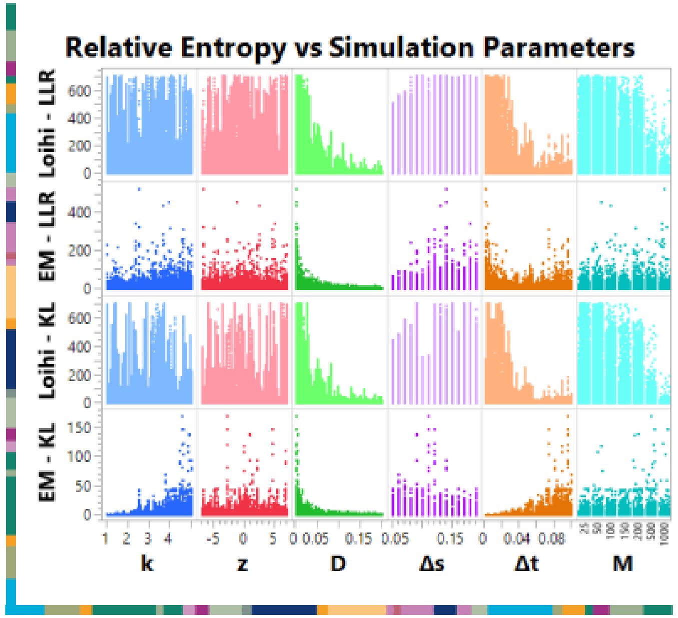
Relative Entropy Results



- We also simulated trajectories in a traditional manner (Euler-Maruyama) for the exact same parameters as Loihi trajectories for the exact same number of time steps.
- Each square subplot contains one dot for each time step in each of the 29,087 trajectories. This is a total of 7,715,213 total measurements per subplot.

Relative Entropy Results

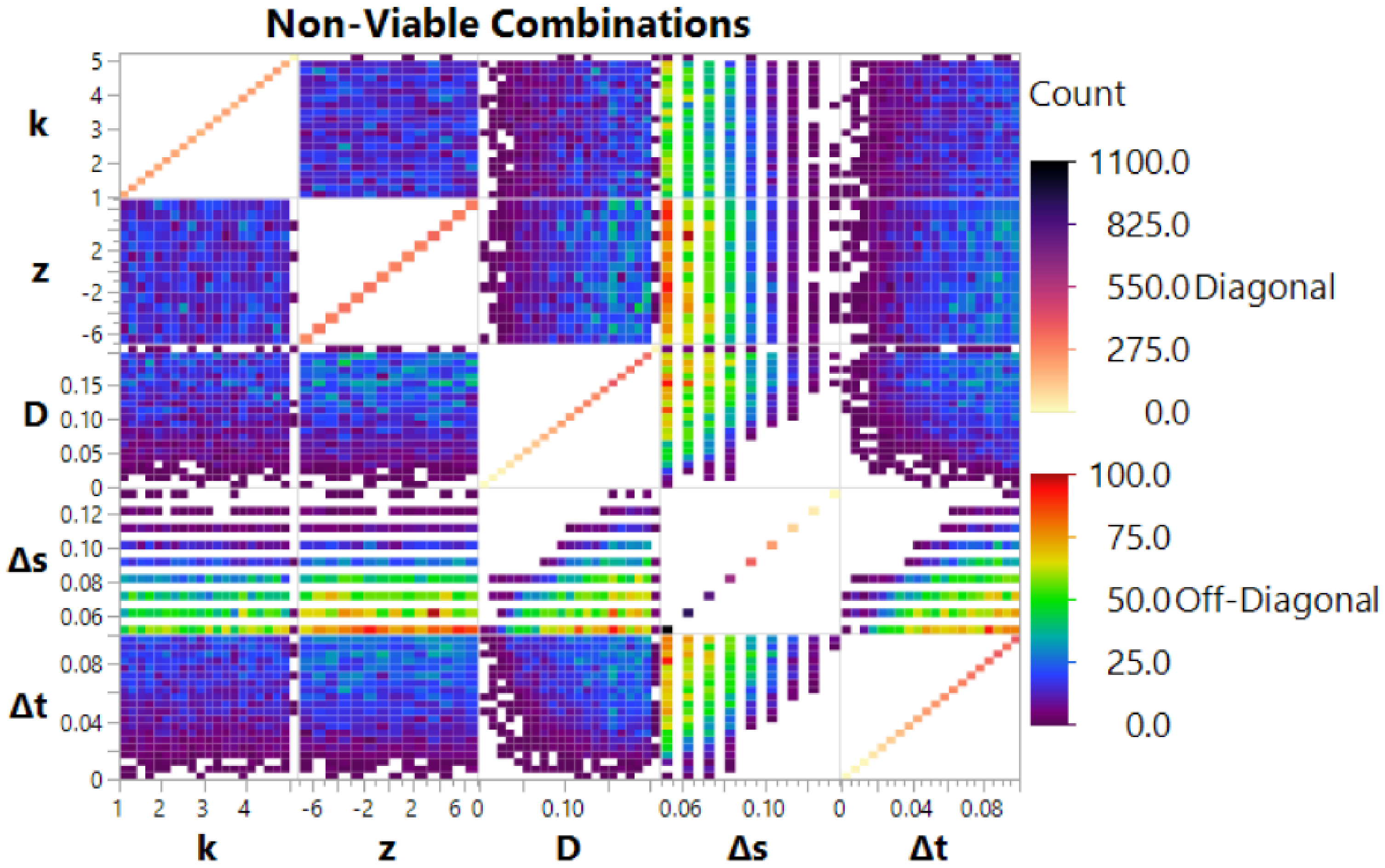
- The majority of time steps observed on Loihi have a relative entropy value less than 1.
- The discrete KL measure is slightly better as the comparison is to a distribution more similar to that of the DTMC.



		M							
		25	50	100	150	200	500	1000	All
Loihi - LLR	Mean	3.0348	2.6378	1.9470	1.7794	1.4786	0.8407	0.6270	2.3859
	Std Dev	24.1788	21.8369	18.8534	15.9938	14.4001	6.9788	5.7801	20.6868
	90% Quantile	1.0880	0.8682	0.7646	0.8320	0.6899	0.7300	0.7626	0.9200
EM - LLR	Mean	0.3517	0.2719	0.2097	0.2017	0.2128	0.2535	0.3536	0.2790
	Std Dev	0.7112	1.0117	1.1081	0.9558	1.5374	1.8285	3.4872	1.1118
	90% Quantile	0.5268	0.3615	0.2853	0.2810	0.2669	0.3348	0.4801	0.4407
Loihi - KL	Mean	1.8094	1.6362	1.1634	1.0352	0.8980	0.4350	0.2031	1.4318
	Std Dev	15.4816	15.8599	11.9741	10.1031	10.5996	5.0513	1.9176	13.8093
	90% Quantile	0.5350	0.3383	0.2463	0.2503	0.2118	0.2387	0.2243	0.4264
EM - KL	Mean	0.1917	0.1105	0.0665	0.0529	0.0477	0.0545	0.0763	0.1187
	Std Dev	0.1796	0.2248	0.2020	0.2809	0.4522	0.8234	0.8567	0.3086
	90% Quantile	0.3588	0.1996	0.1118	0.0805	0.0648	0.0442	0.0837	0.2593

Algorithm’s Viable Simulation Space

- We examine 4229 combinations of $\Delta s, \Delta t, k, z,$ and D that fail to simulate using our Loihi algorithm.



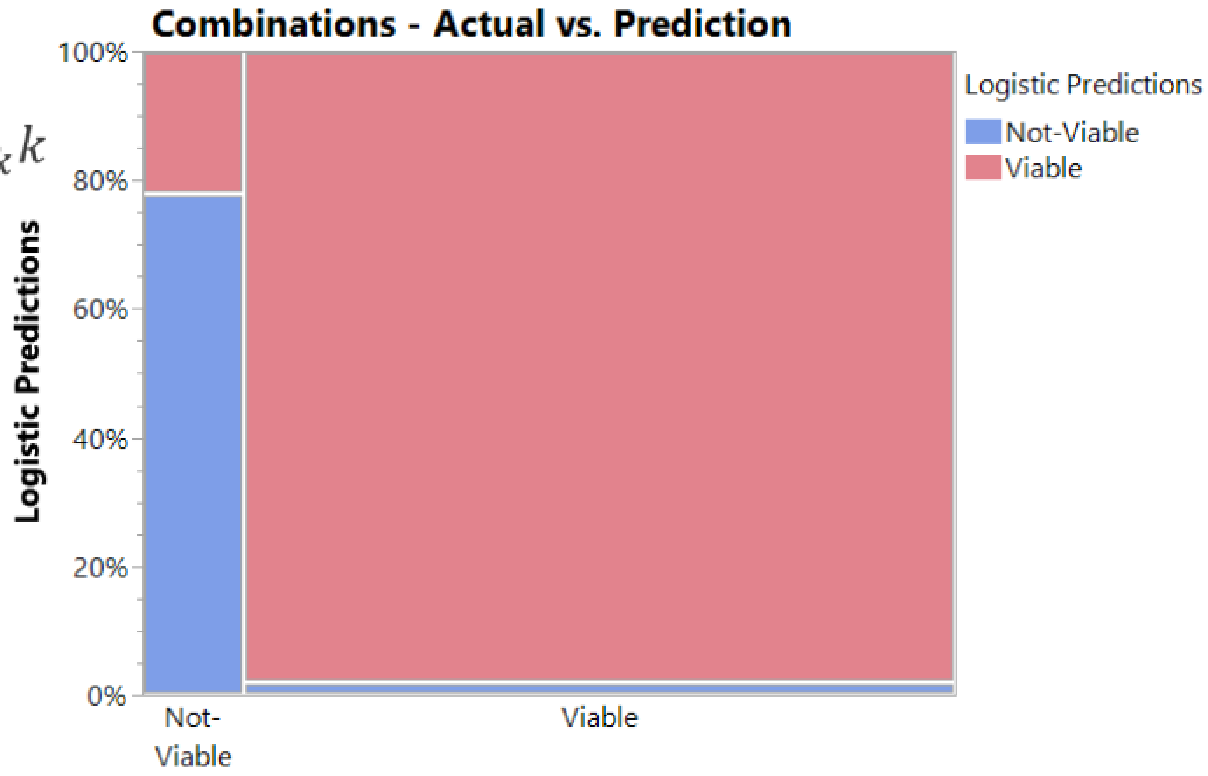
Predicting Viable Combinations

- We use a logistic model in four variables $\Delta s, D, \Delta t, k$ to model the probability of successful simulation.

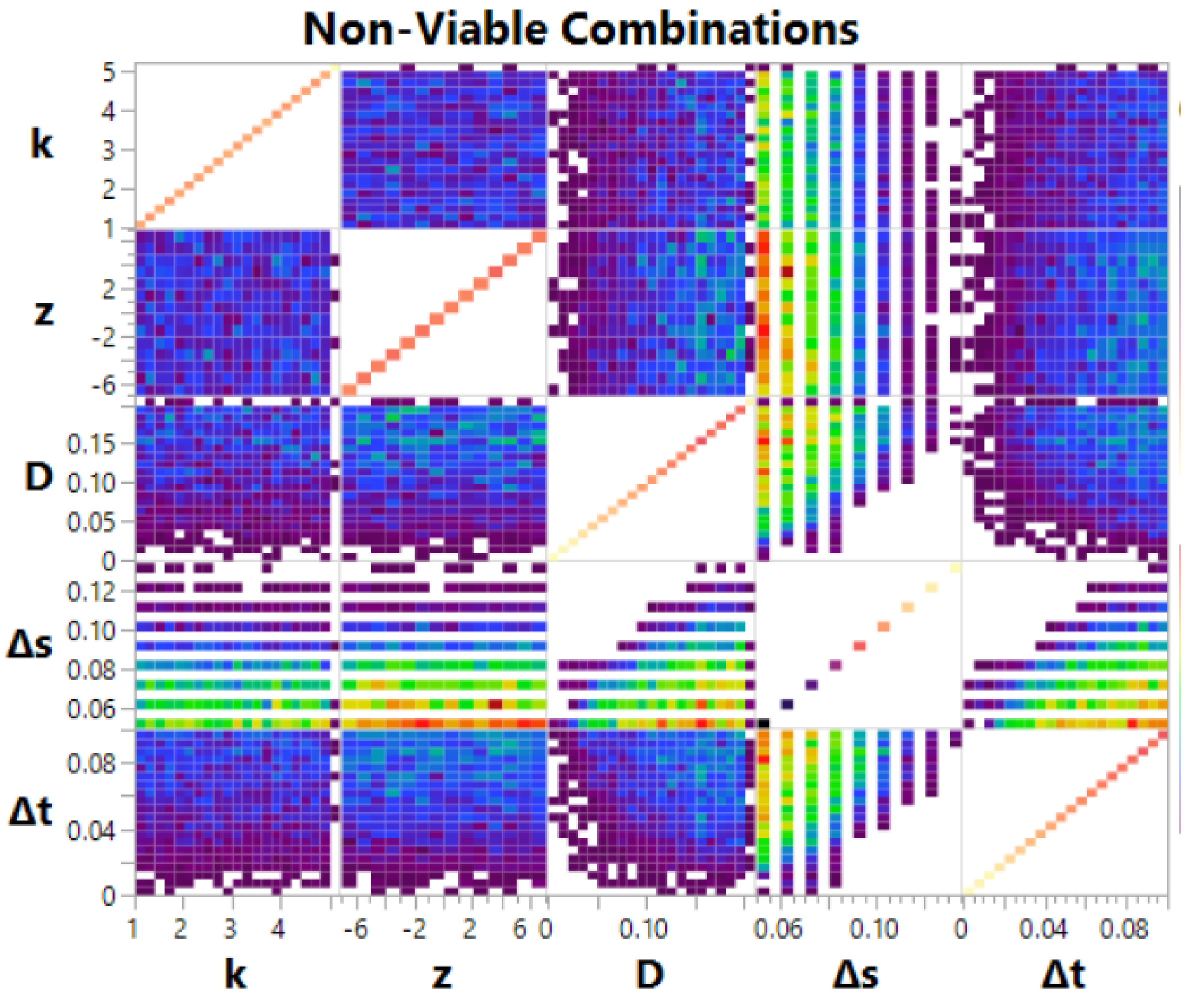
$$\mathbb{P}[\text{viable} \mid \Delta s, D, \Delta t, k] = \frac{1}{1 + \exp\left(-f(\Delta s, D, \Delta t, k)\right)},$$

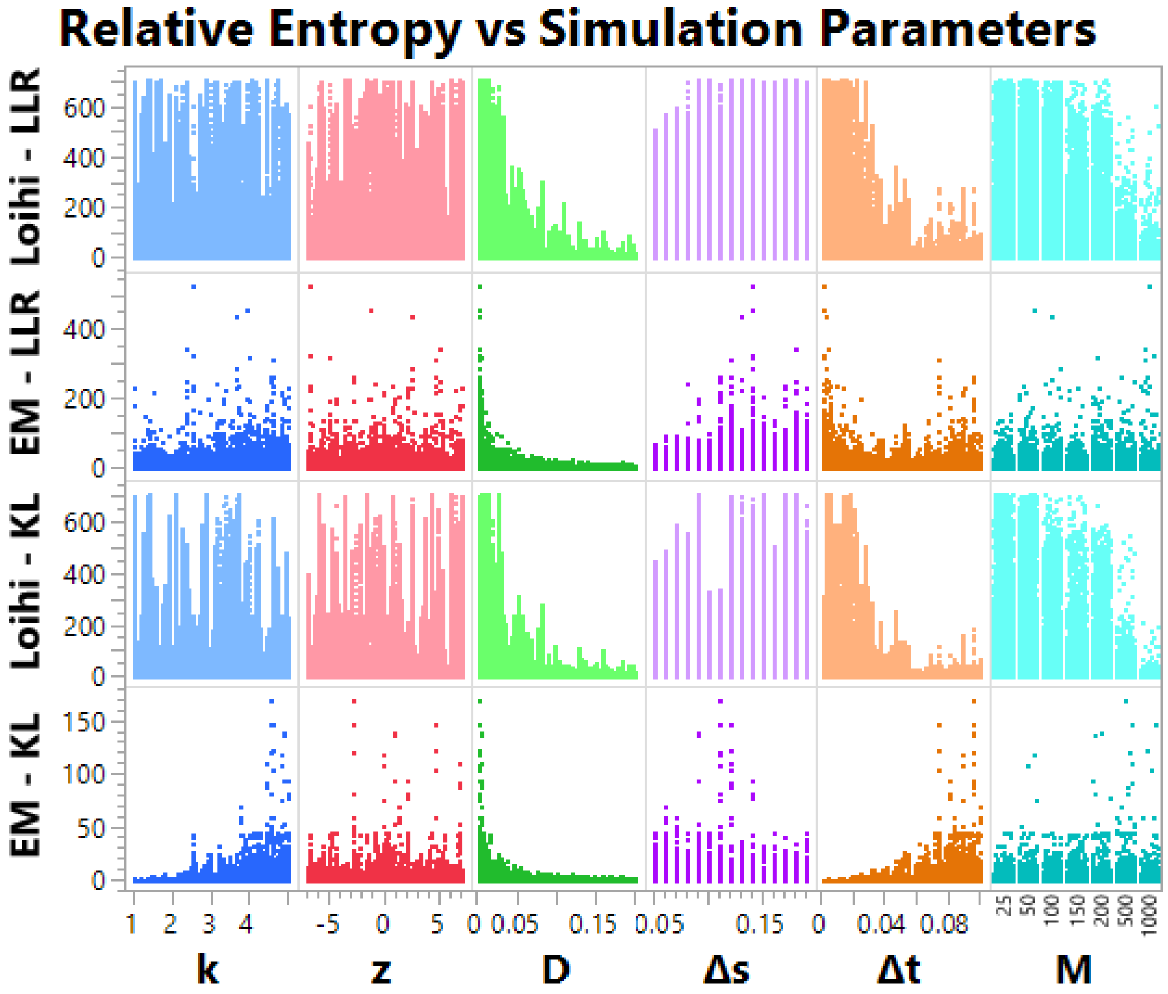
$$f(\Delta s, D, \Delta t, k) = \beta_0 + \beta_{\Delta s}\Delta s + \beta_D D + \beta_{\Delta t}\Delta t + \beta_k k$$

	Estimate	Std. Error
β_0	-0.3609	0.1442
$\beta_{\Delta s}$	134.1436	2.4284
β_D	-43.0086	0.8776
$\beta_{\Delta t}$	-84.2552	1.7536
β_k	0.0866	0.0262



A Trade-Off in Simulation





Where do hardware limitations end and algorithm limitations begin?

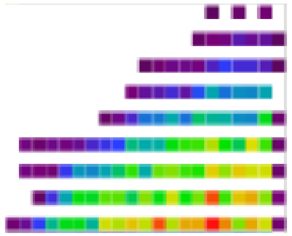
Hardware

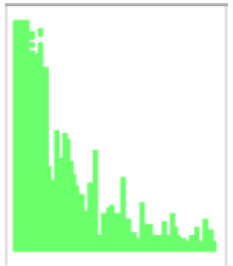
- Limited PRNG availability and precision.

$$\frac{1}{256}$$

0	0	1	0	0	1	1	1

- Viable parameter space limitations and accuracy implications.



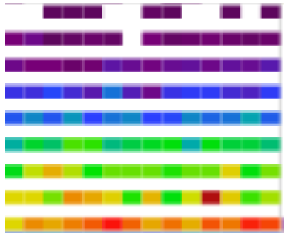


Algorithm

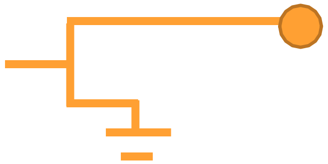
- More resource-intensive probability circuits.

0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1

- Further limited parameter space.



- Clever efficient circuits?



Conclusions and Future Questions

- Neuromorphic can reasonably sample from DTMCs today.

- We have provided a methodology to assess the outcome of sampling a stochastic process given a neuromorphic technology and a sampling algorithm.

- A trade-off exists in hardware and algorithm limitations when applying emerging technologies to existing numerical and scientific computing tasks.

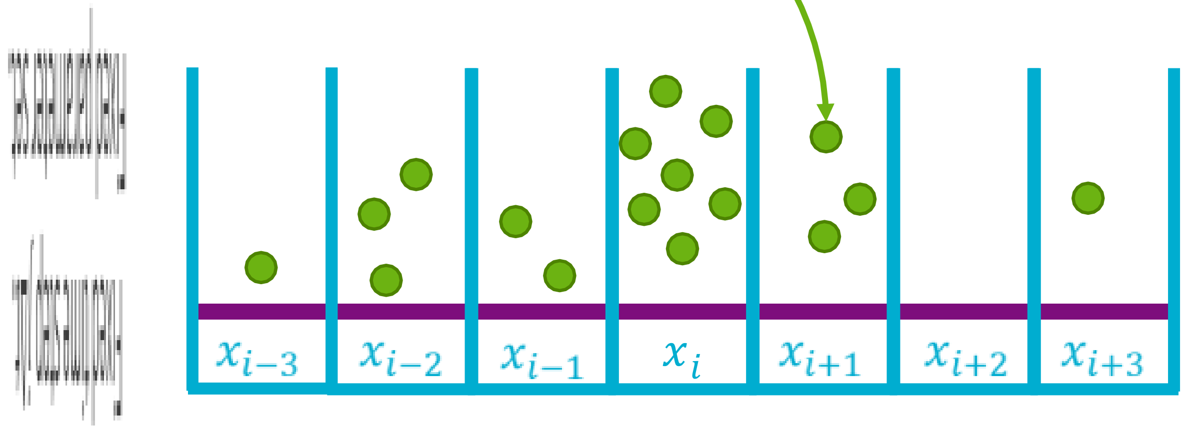
- The existing limitations are an opportunity to discover new, efficient, and optimized algorithms for scientific applications.

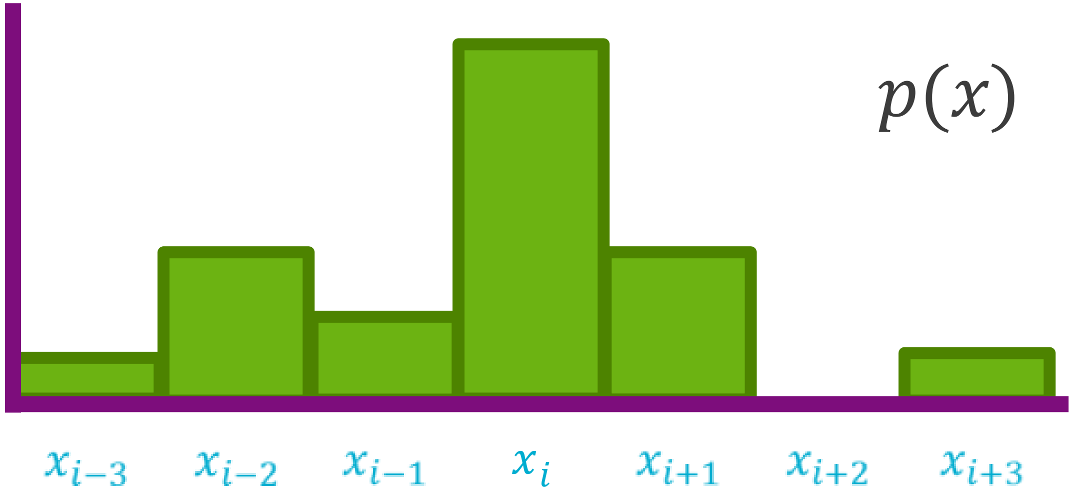
Thanks!

The Log-Likelihood Ratio (LLR)

- To use our samples in a direct comparison to the distribution q, we can generate a histogram probability density.

The m^{th} observation, X_m^j.





Divide the number of walkers in each bin by $M\Delta s$,
the total number of walkers times the bin size.

The Log-Likelihood Ratio (LLR)

- Using the previous notation, we then define the normalized log-likelihood ratio:

$$\widetilde{\Lambda}_M(p,q) = \frac{1}{M}\sum_{m=1}^{M} \log \frac{p\left(X_j^m = x_i\right)}{q(x_i)}$$

- For a fixed p, $\widetilde{\Lambda}_M \to \kappa$ as $M \to \infty$.
- This allows us a direct comparison to the known OU distribution q.

Relative Entropy Results

- We also simulated a DTMC implementation on MATLAB and artificially reduced the probabilities of transition.

		All
Loihi - LLR	Mean	2.3859
	Std Dev	20.6868
	90% Quantile	0.9200
Loihi - KL	Mean	1.4318
	Std Dev	13.8093
	90% Quantile	0.4264

		Bit			
		6	7	8	Full
DTMC - LLR	Mean	1.7200	1.7071	1.7092	1.7034
	Std Dev	18.2399	18.1474	18.1465	18.0668
	90% Quantile	0.4772	0.4825	0.4771	0.4784
DTMC - KL	Mean	1.0413	1.0312	1.0328	1.0343
	Std Dev	11.3536	11.2774	11.2446	11.2836
	90% Quantile	0.1917	0.1894	0.1906	0.1913