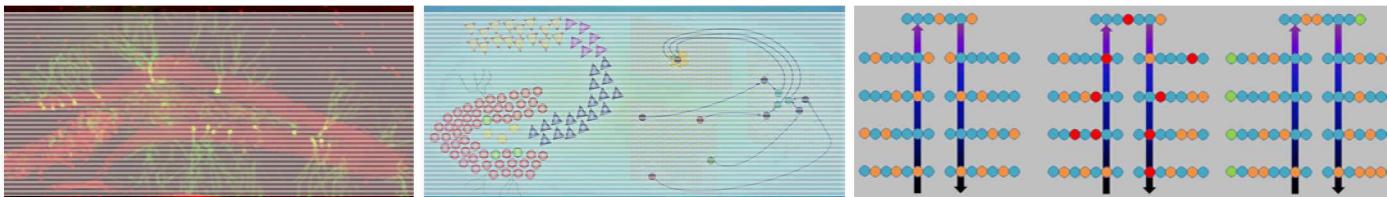


# Computing with Spikes

*Everything from the deep learning to numerical applications*



PRESENTED BY

Brad Aimone

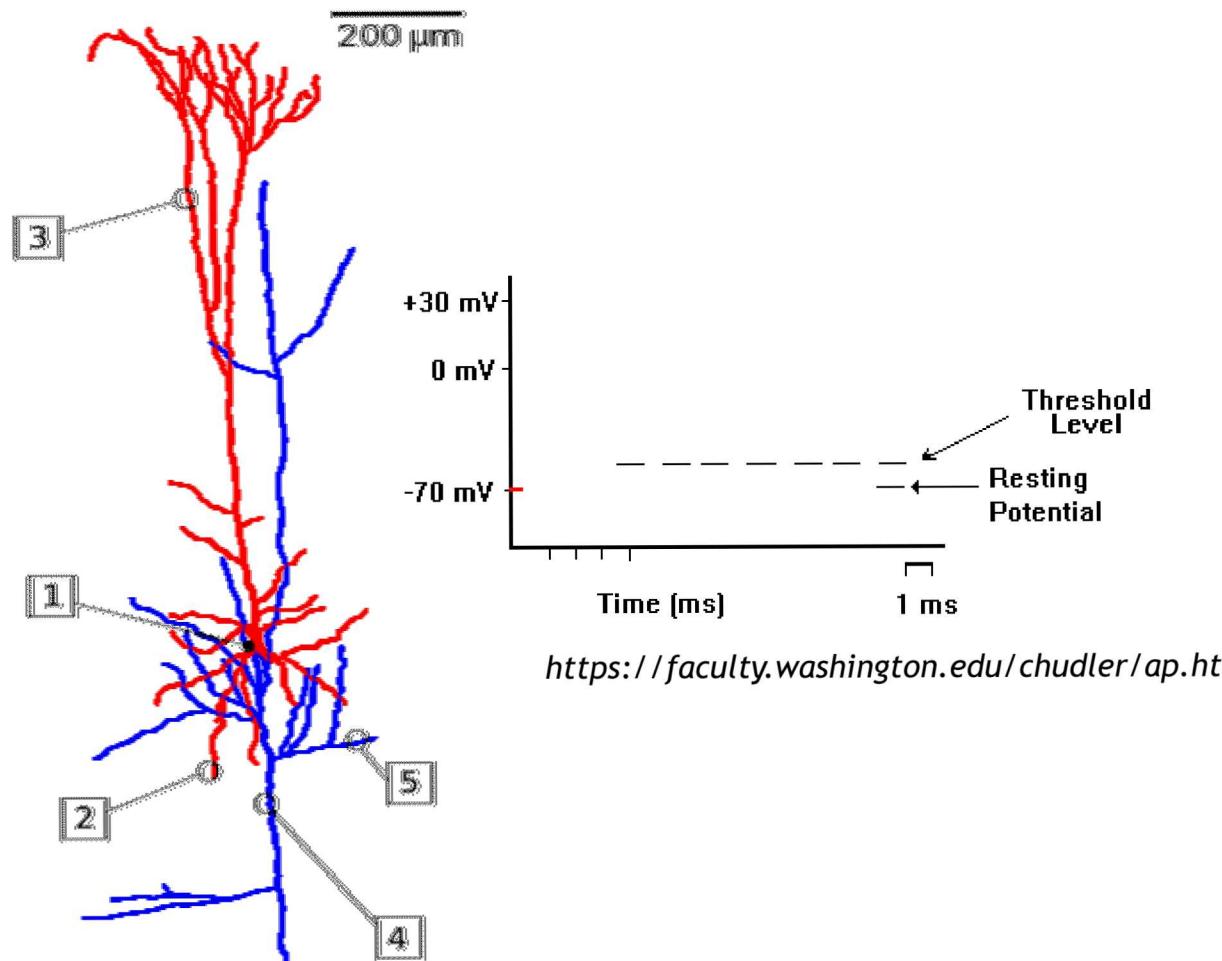


SAND2019-12692PE

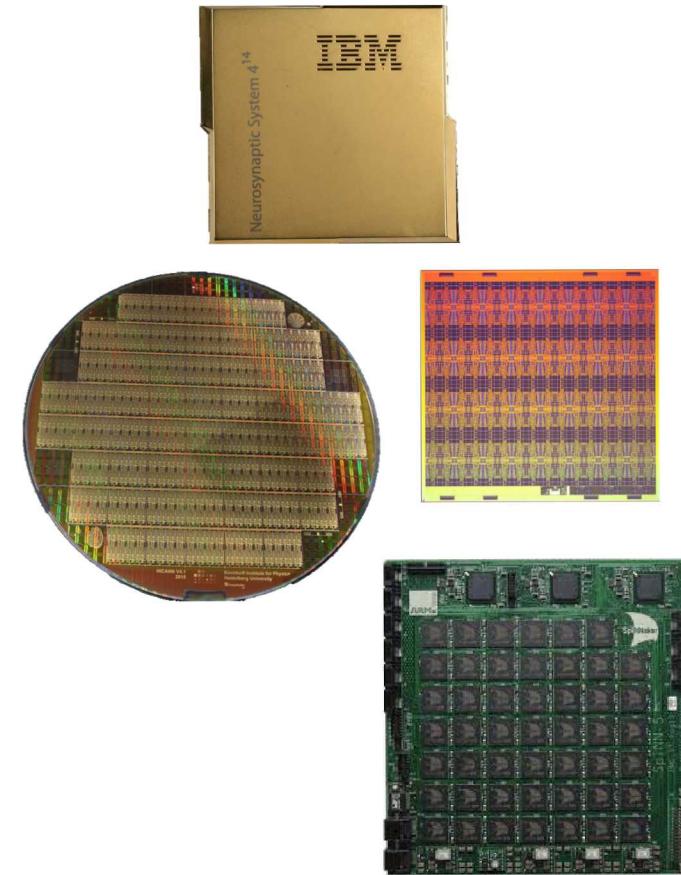


Sandia National Laboratories is a  
multimission laboratory managed and  
operated by National Technology and  
Engineering Solutions of Sandia LLC, a wholly  
owned subsidiary of Honeywell International  
Inc. for the U.S. Department of Energy's  
National Nuclear Security Administration  
under contract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

# What is spiking?



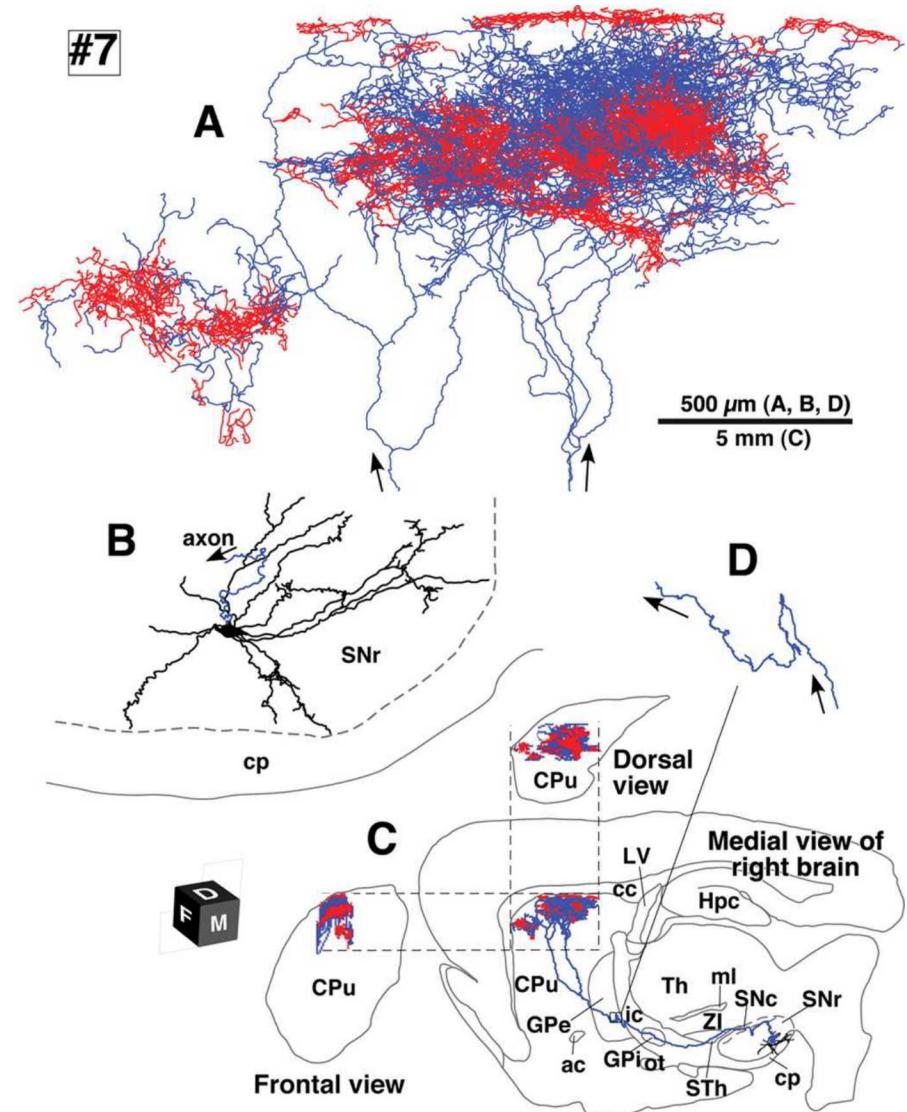
Pyramidal Cell -- Wikipedia



Clockwise from top: IBM TrueNorth,  
Intel Loihi, SpiNNaker, BrainScales

# Why spiking?

- Event-driven
  - Only expend energy when neuron crosses threshold
- Reliable and efficient over long distances
  - Neurons often project across brain or whole body...
- Robust to noise
  - Away from threshold, biophysical noise should not accidentally cause spikes



# Correcting some common misconceptions about spiking

## ➤ *Spiking is necessary for brain-like computation*

- Reality: The advantage of spiking is efficiency and reliability over distance, not computability.

It changes the tradeoffs between time, power, and space

## ➤ *Spiking does not offer anything for algorithms*

- Reality: Spiking facilitates developing algorithms that more directly leverage time in computing

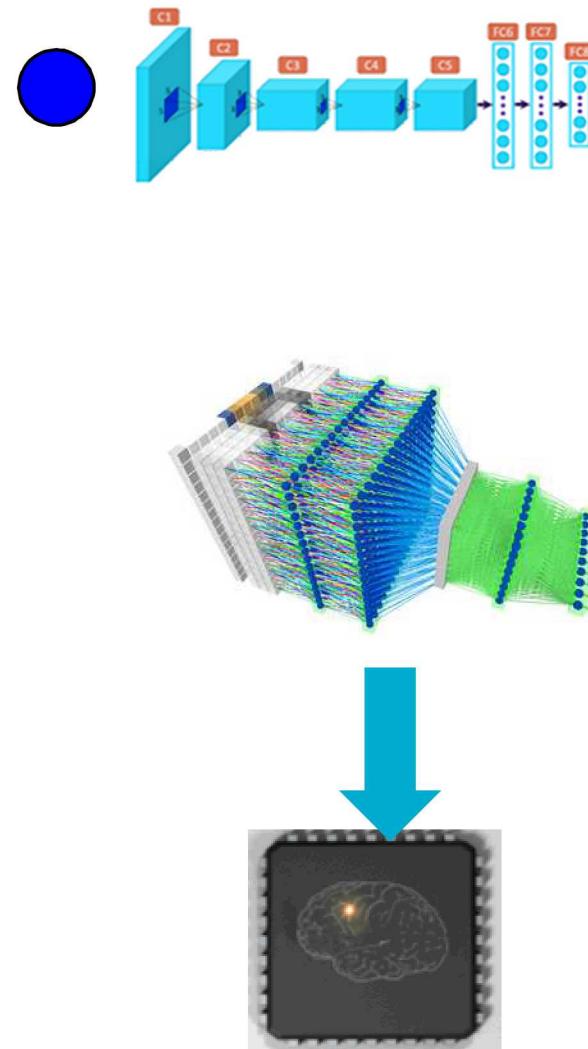
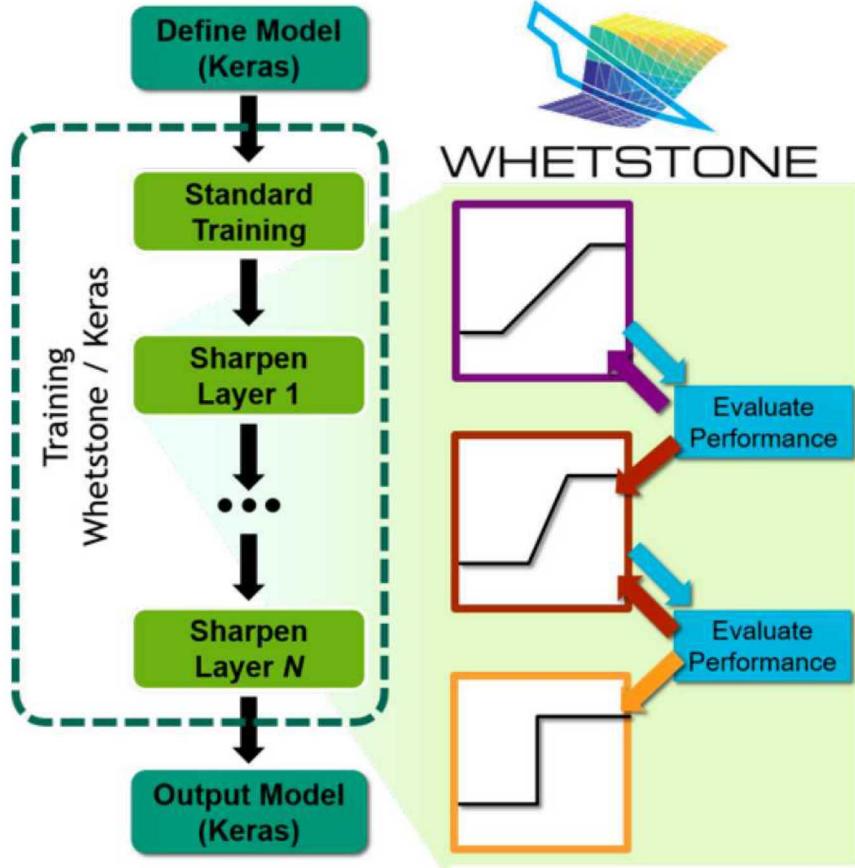
## ➤ *Spiking reduces the accuracy of algorithms*

- Reality: Not necessarily! Spiking does lower the precision of communication, but often this precision is unnecessary or can be compensated for in other ways.

## ➤ *Spiking requires paying a time penalty*

- Reality: Not always! Some coding schemes are actually time advantageous – e.g., you can implement very fast threshold gate circuit algorithms on spiking hardware

# What can you do with spiking neurons?



## Spiking deep neural networks

- Whetstone allows us to use spiking communication with *no time penalty* and minimal accuracy reduction

ARTICLES  
<https://doi.org/10.1038/s43256-019-0015-y>

## Training deep neural networks for binary communication with the Whetstone method

William Severa<sup>◎</sup>, Craig M. Vineyard<sup>◎</sup>, Ryan Dellana<sup>◎</sup>, Stephen J. Verzi<sup>◎</sup> and James B. Aimeone<sup>◎</sup>  
The computational cost of deep neural networks presents challenges to broadly deploying these algorithms. Low-power embedded neuromorphic processors afford potentially dramatic performance-per-watt improvements over traditional processor architectures. However, programming these brain-inspired platforms generally requires platform-specific expertise. It is therefore crucial to achieve state-of-the-art performance on these platforms, limiting their applicability. Here we present Whetstone, a deep learning framework that is designed to be highly portable and easy to use. Whetstone is a spiking neural network framework that uses a spiking neural network to implement a feed-forward neural network. The spiking neural network is trained using a backpropagation-like process, the activations of which at each layer are progressively sharpened through binomial activation, with limited loss of performance. Whetstone networks do not require a rate code or other spike-based coding scheme, thus producing

networks comparable in timing and size to conventional artificial neural networks. We demonstrate Whetstone on a number of architectures and tasks such as image classification, autoencoders and semantic segmentation. Whetstone is currently implemented within the Keras wrapper for TensorFlow and is freely extendable.

algorithms that simply require the deployment of an instance architecture. In contrast, the *ANN-based approach* uses learned kernels of ANNs in application-specific, integrated circuits (ASICs). However, while these ASICs can provide substantial performance improvements over software-implemented applications and offer lack flexibility for implementing changing ANN architectures.

ingly these approaches leverage spiking to achieve substantial energy reduction between nodes. With this shift of the optimization target, the

---

Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA. 'e-mail: wmsauer@sandia.gov

NATURE MACHINE INTELLIGENCE | VOL. 1 | FEBRUARY 2019 | 88–94 | www.nature.com/natmachint/

## lecture Machine Intelligence

# Large Machine Learning

NICE Proceedings 20

NICE Proceedings, 20

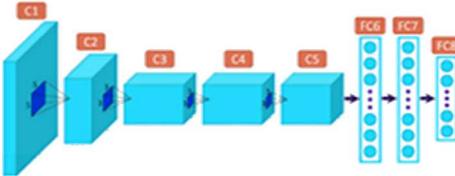
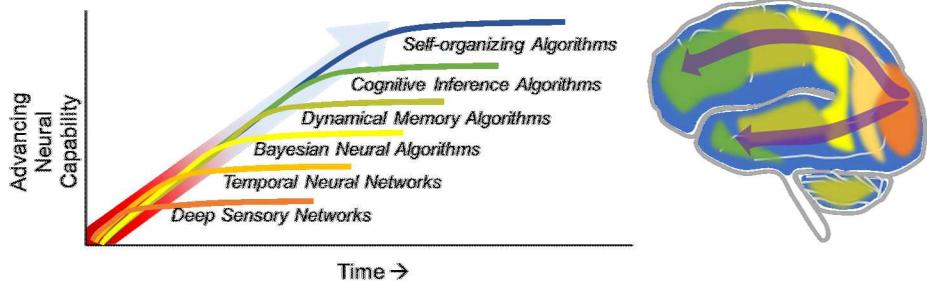
THE PROSECUTOR, 23

Severa et al., *Nature Machine Intelligence*, Feb 2019  
Vineyard et al., *NICE Proceedings*, 2019

# What can you do with spiking neurons?



Algorithm Class	Current Algorithms	Inspiration	Application
Deep Vision Processing	Deep Convolutional Networks (VGG, AlexNet, GoogleNet, etc.), HMax, Neocognitron	Hierarchy of sensory nuclei and early sensory cortices	Static feature extraction (e.g., images) & pattern classification
Temporal Neural Networks	Deep Recurrent Networks (e.g., long short-term memory), Hopfield Networks	Local recurrence of most biological neural circuits, especially higher sensory cortices	Dynamic feature extraction (e.g., videos, audio) & classification
Bayesian Neural Algorithms	Predictive Coding, Hierarchical Temporal Memory, Recursive Cortical Networks	Substantial reciprocal feedback between "higher" and "lower" sensory cortices	Inference across spatial and temporal scales
Dynamical Memory and Control Algorithms	Liquid State Machines, Echo State Networks, Neural Engineering Framework	Continual dynamics of hippocampus, cerebellum, and prefrontal and motor cortices	Online learning content-addressable memory & adaptive motor control
Cognitive Inference Algorithms	Reinforcement learning (e.g., Deep Q-learning) Neural Turing Machines	Integration of multiple modalities and memory into prefrontal cortex, which provides top-down influence on sensory processing	Context and experience dependent information processing and decision making
Self-organizing Algorithms	Neurogenesis Deep Learning	Initial development and continuous refinement of neural circuits to specific input and outputs	Automated neural algorithm development for unknown input and output transformations

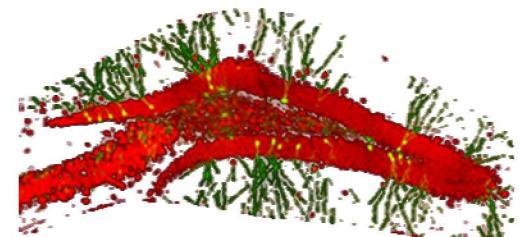


## Spiking deep neural networks

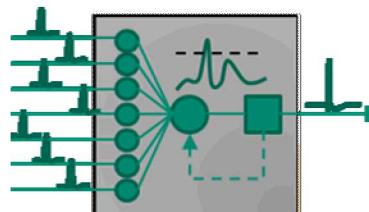
- Whetstone allows us to use spiking communication with *no time penalty* and minimal accuracy reduction

## Neuroscience-constrained algorithms

- Computation incorporates broad range of neural plasticity and dynamics
- Generally still unexplored from algorithms perspective*

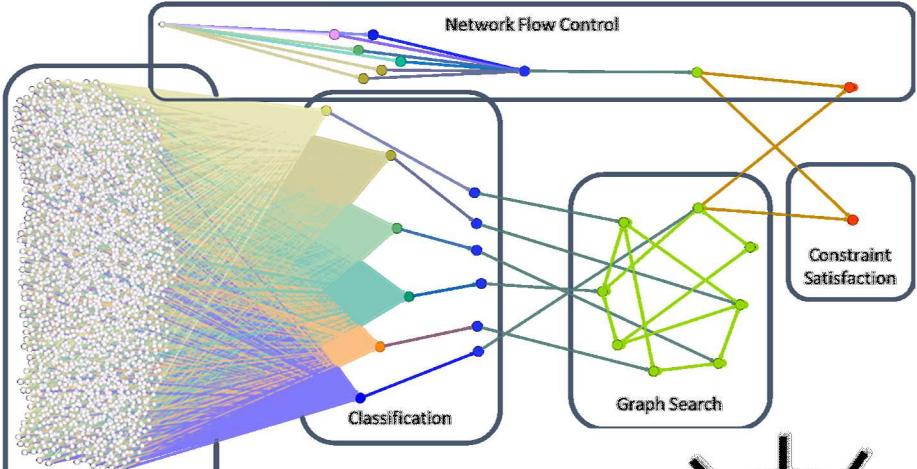


# What can you do with spiking neurons?

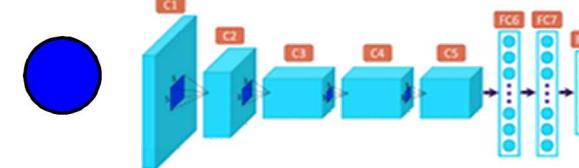


*Treat neurons as powerful logic gates*

*Algorithms are circuits...*

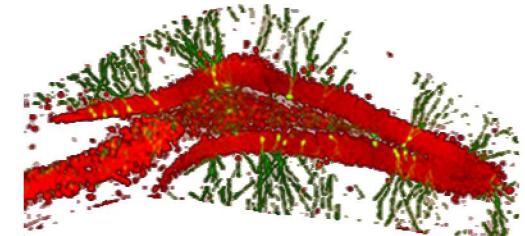


*Aimone et al, ICONS 2019*



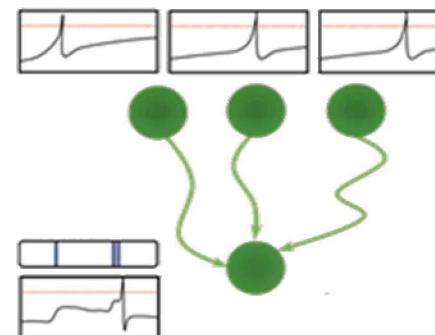
## Spiking deep neural networks

- Whetstone allows us to use spiking communication with *no time penalty* and minimal accuracy reduction



## Neuroscience-constrained algorithms

- Computation incorporates broad range of neural plasticity and dynamics
- Generally still unexplored from algorithms perspective*



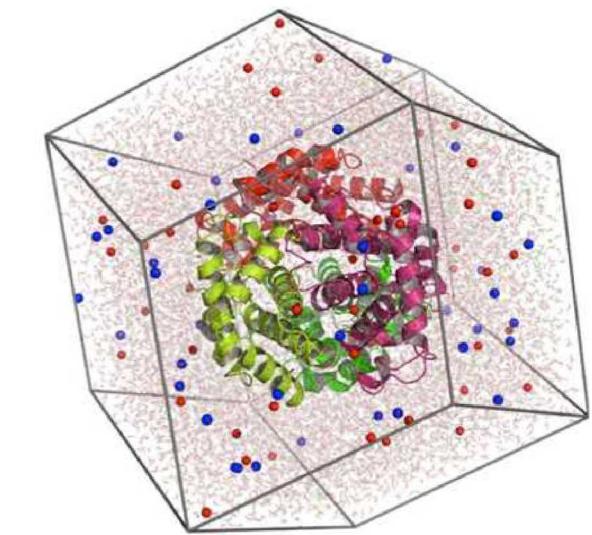
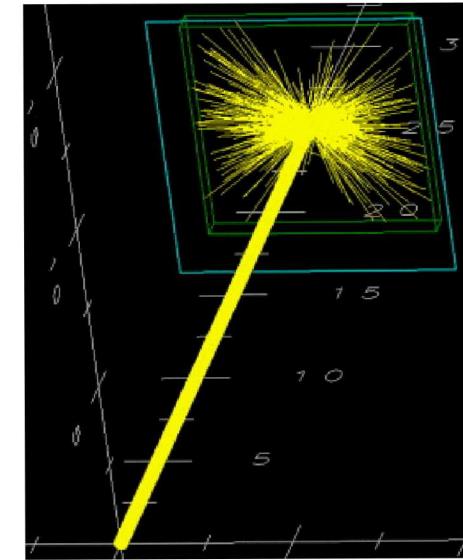
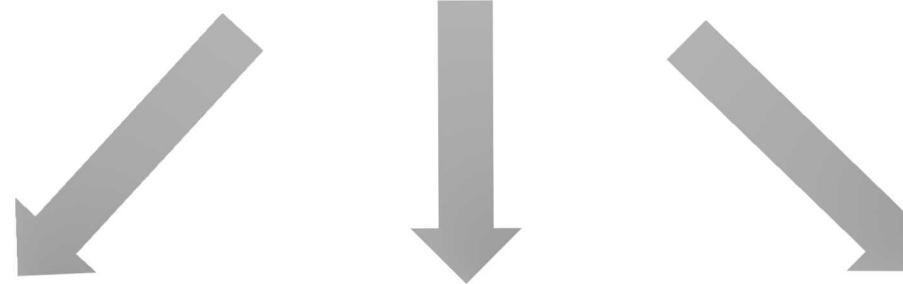
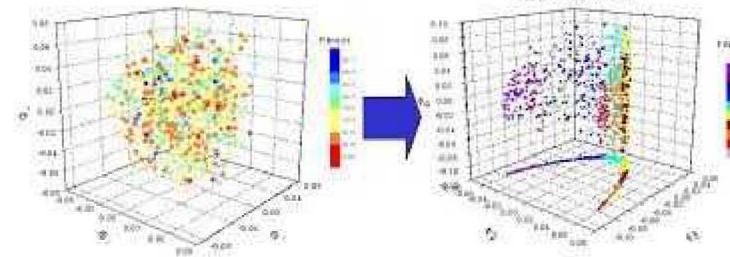
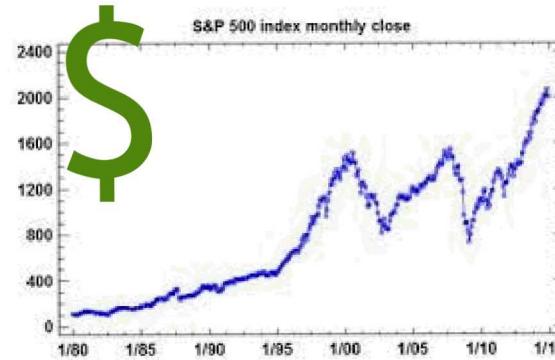
## Spiking neural algorithms

- Hand-crafted circuits of spiking neurons
- Model of parallel computation
- Energy efficiency through event-driven communication and high fan-in logic

Can spiking really be used to solve non-cognitive tasks efficiently?

# Spiking circuits can efficiently solve stochastic differential equations

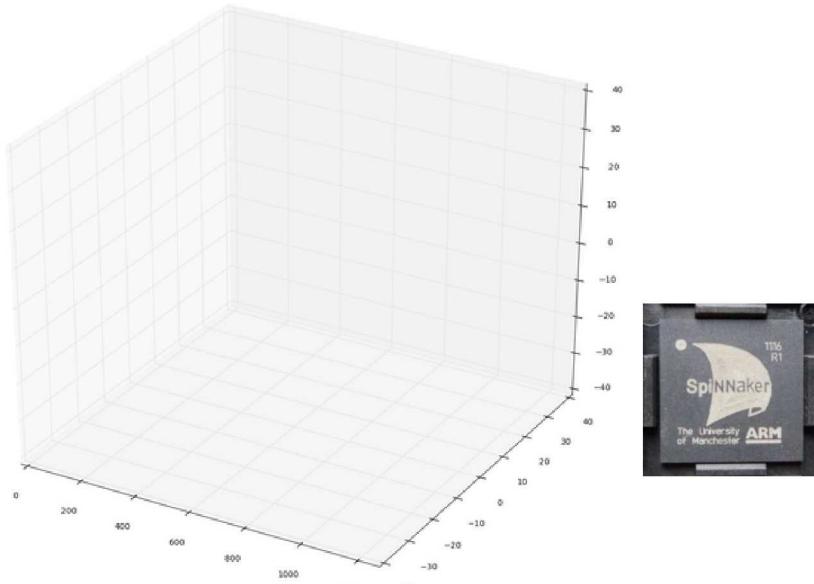
Diffusion:  $\frac{\partial C(x,t)}{\partial t} = D \frac{\partial^2 C(x,t)}{\partial x^2}$



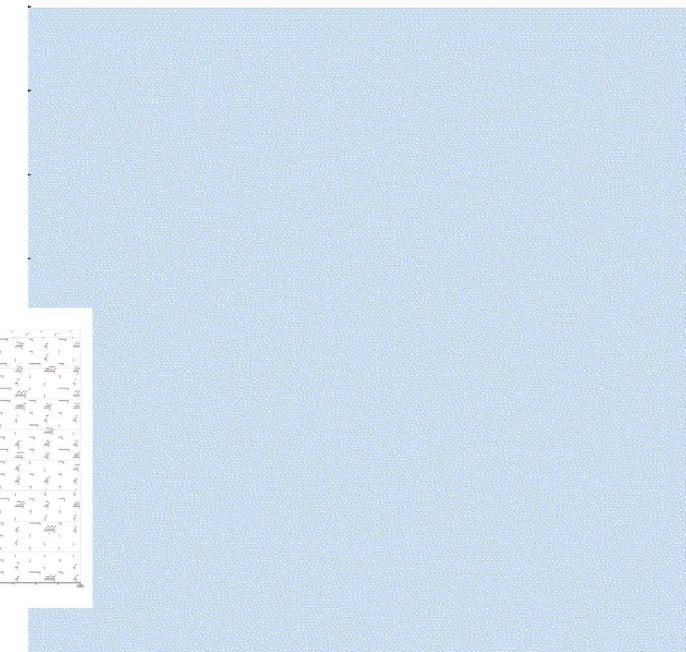
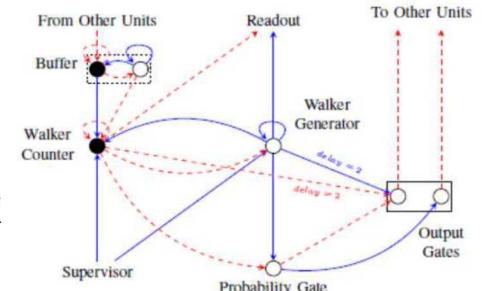
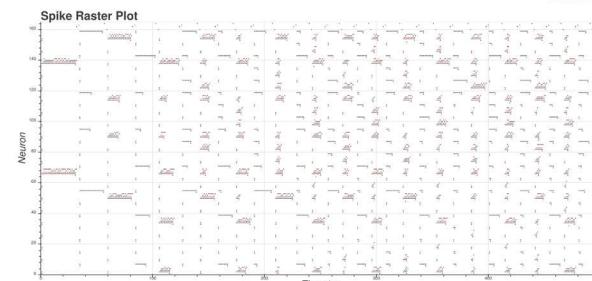
# Spiking circuits can efficiently solve stochastic differential equations

Diffusion:  $\frac{\partial C(x,t)}{\partial t} = D \frac{\partial^2 C(x,t)}{\partial x^2}$

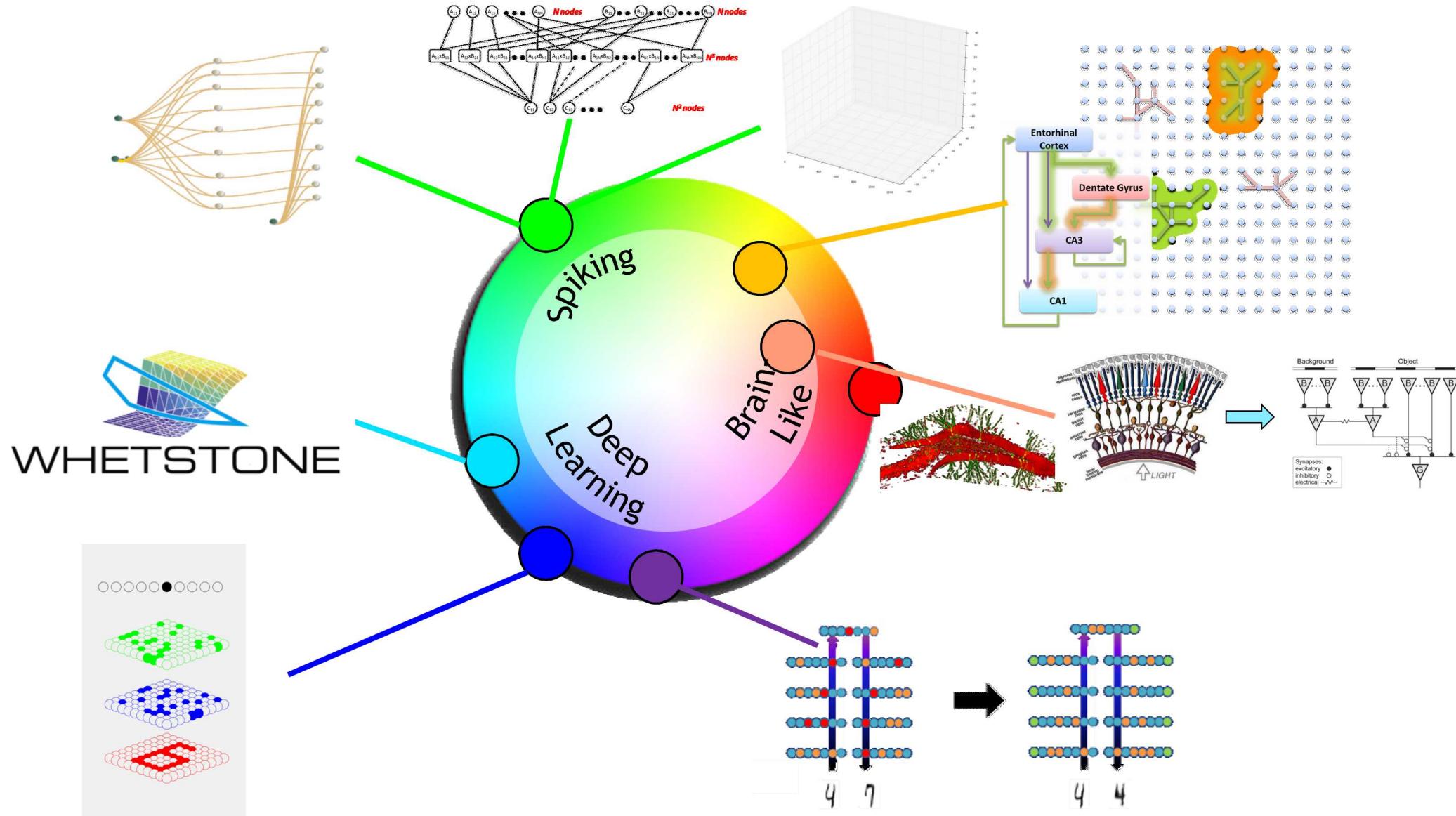
Modular circuit of spiking neurons per random walk particle



RW counting circuit of spiking neurons per simulation mesh vertex



# There is an increasing diversity of spiking algorithms and applications



A brief plug...



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

## NICE 2020

March 17-20<sup>th</sup>, 2020

Neuro-Inspired  
Computational Elements  
Workshop

Abstracts due  
November 1st, 2020



Im Neuenheimer Feld 227  
D-69120 Heidelberg  
Germany

Workshop: March 24-26<sup>th</sup> 2020  
Tutorials: March 27<sup>th</sup> 2020

## Heidelberg - Germany



Picture: fotolia.com / Sergey Borisov

## Kirchhoff Institute for Physics