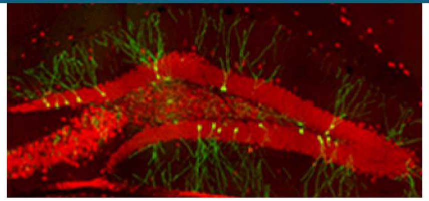
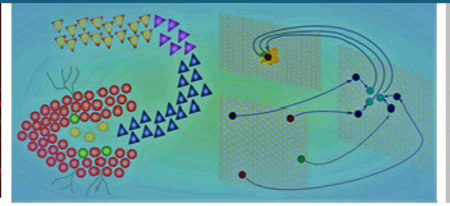
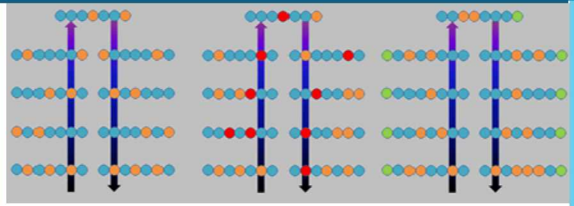
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2019-8594C

Neuromorphic Benchmarking at the Neural Exploration and Research Laboratory



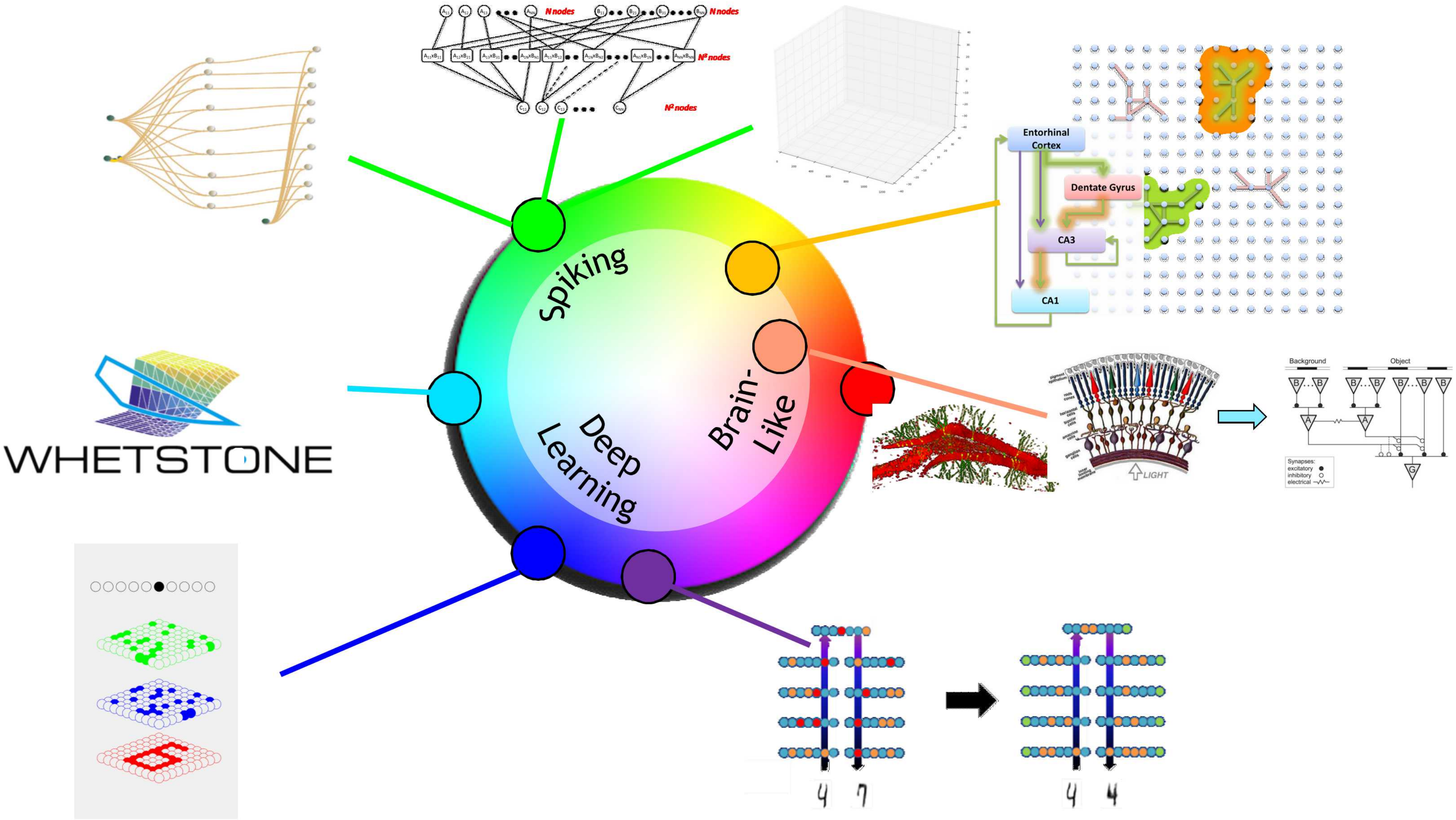




William M. Severa

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.

Introduction: Very low power solutions for a lot more than deep learning!

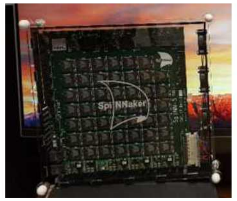


Neuromorphic Hardware @ Neural Exploration & Research Lab (NERL)

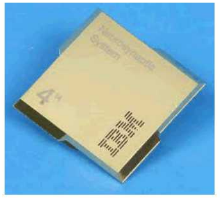
- Enables researchers to explore the boundaries of neural computation
- Consists of a variety of neuromorphic hardware & neural algorithms providing a testbed facility for comparative benchmarking and new architecture exploration

Intel Loihi

SpiNNaker 48 Node Board



IBM TrueNorth*



IBM TrueNorth NS16e*

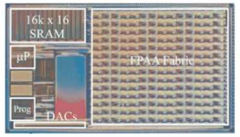
Intel Neural Compute Stick

Google Coral

Google EdgeTPU

Inilabs DAVIS 240C DVS

Georgia Tech FPAA



Intel Loihi

SNL STPU on FPGA

Xilinx PYNQ FPGA

Nengo FPGA

Nvidia Jetson TX1

Nvidia Jetson Nano

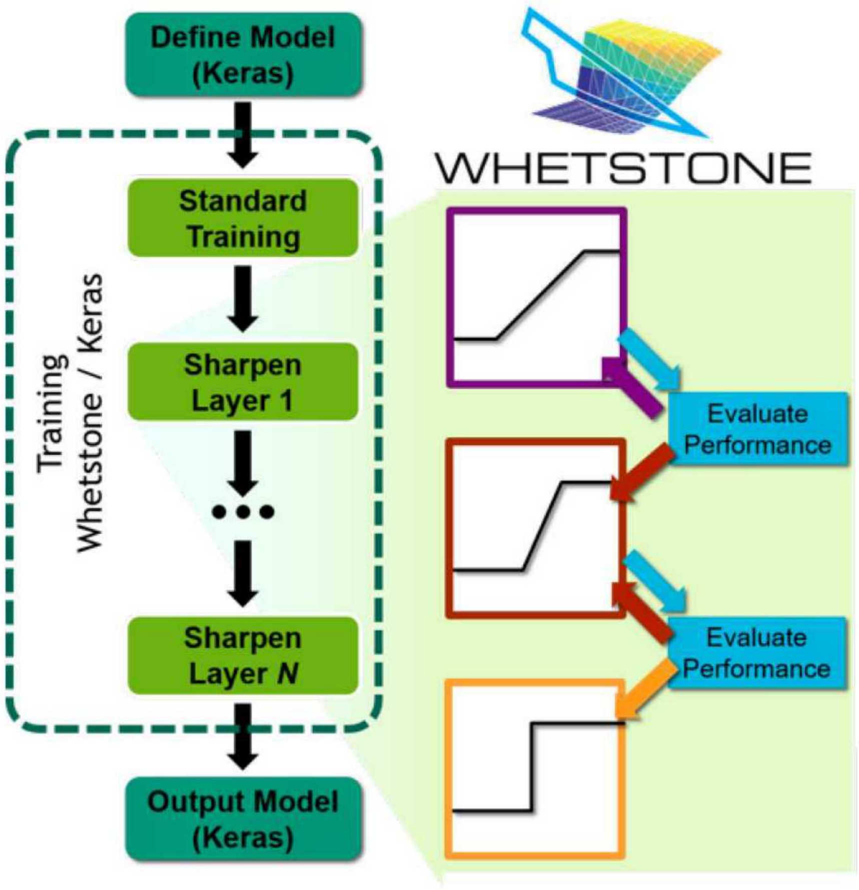
GPU Workstations

Cognimem CM1K

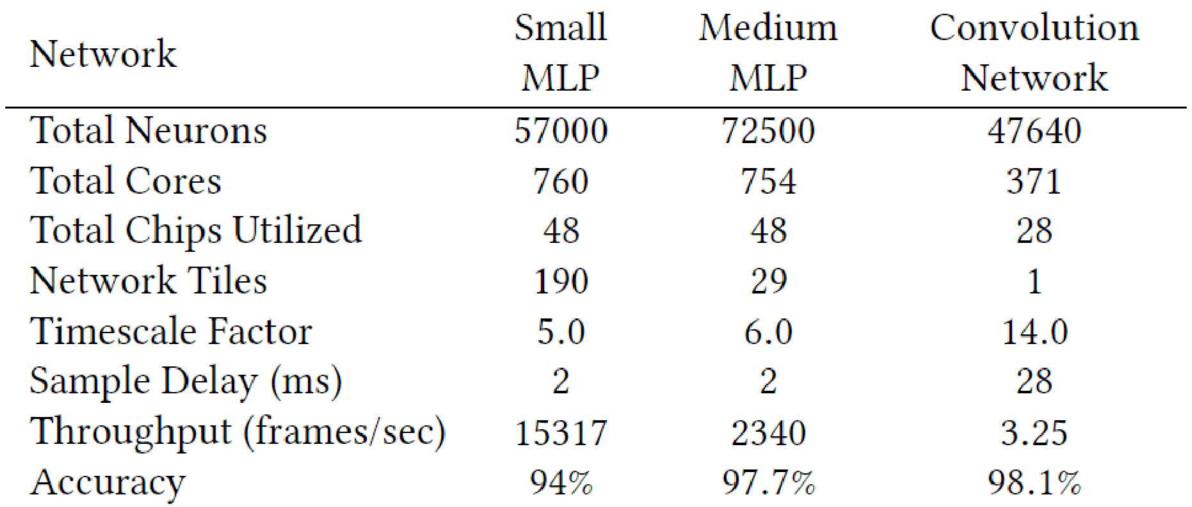
KnuPath Hermosa

*Remote access

Whetstone Overview



- Automatically converts deep learning networks from continuous valued neurons to binary activations, making them compatible with neuromorphic hardware
- Open sourced
- Published in February
- Beginning to port onto neuromorphic platforms
  - SpiNNaker Results look great

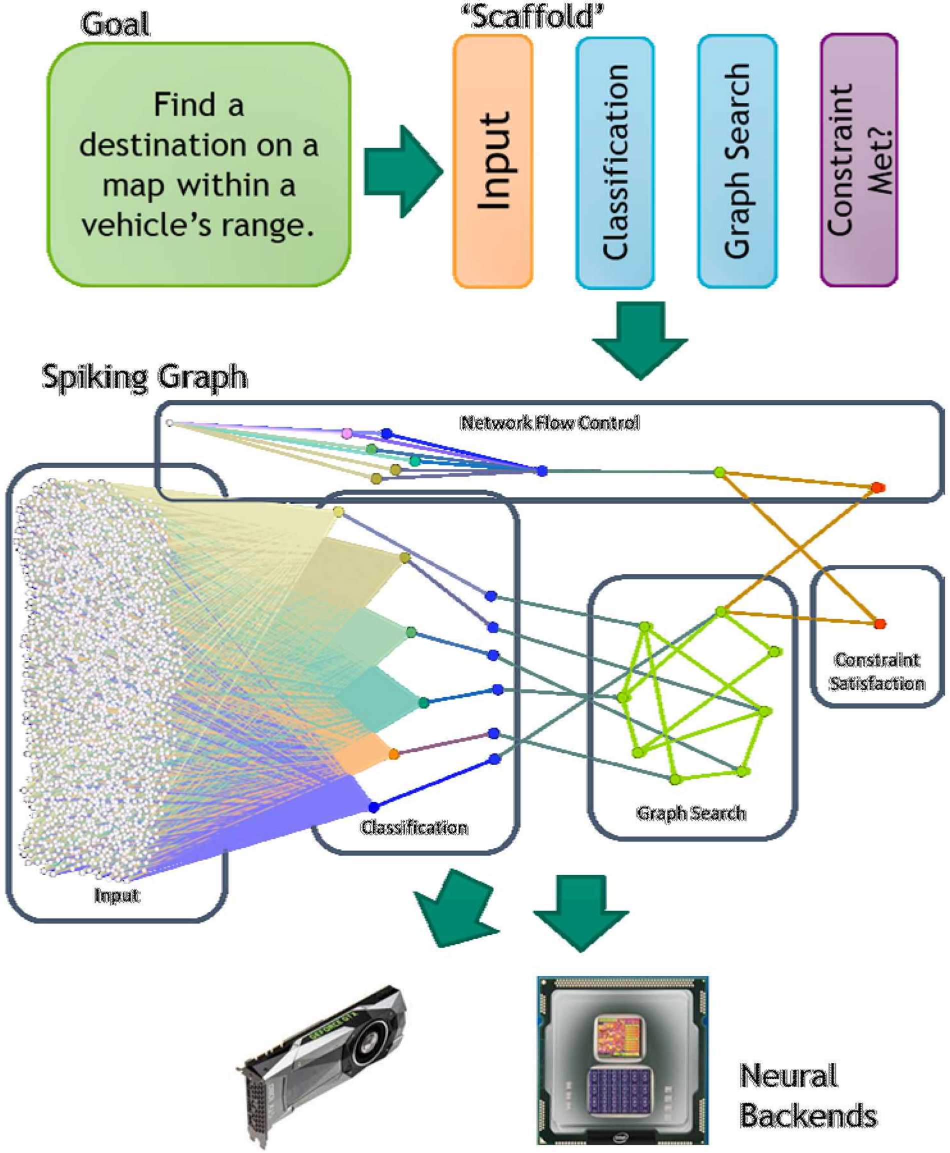


Network	Small MLP	Medium MLP	Convolution Network
Total Neurons	57000	72500	47640
Total Cores	760	754	371
Total Chips Utilized	48	48	28
Network Tiles	190	29	1
Timescale Factor	5.0	6.0	14.0
Sample Delay (ms)	2	2	28
Throughput (frames/sec)	15317	2340	3.25
Accuracy	94%	97.7%	98.1%

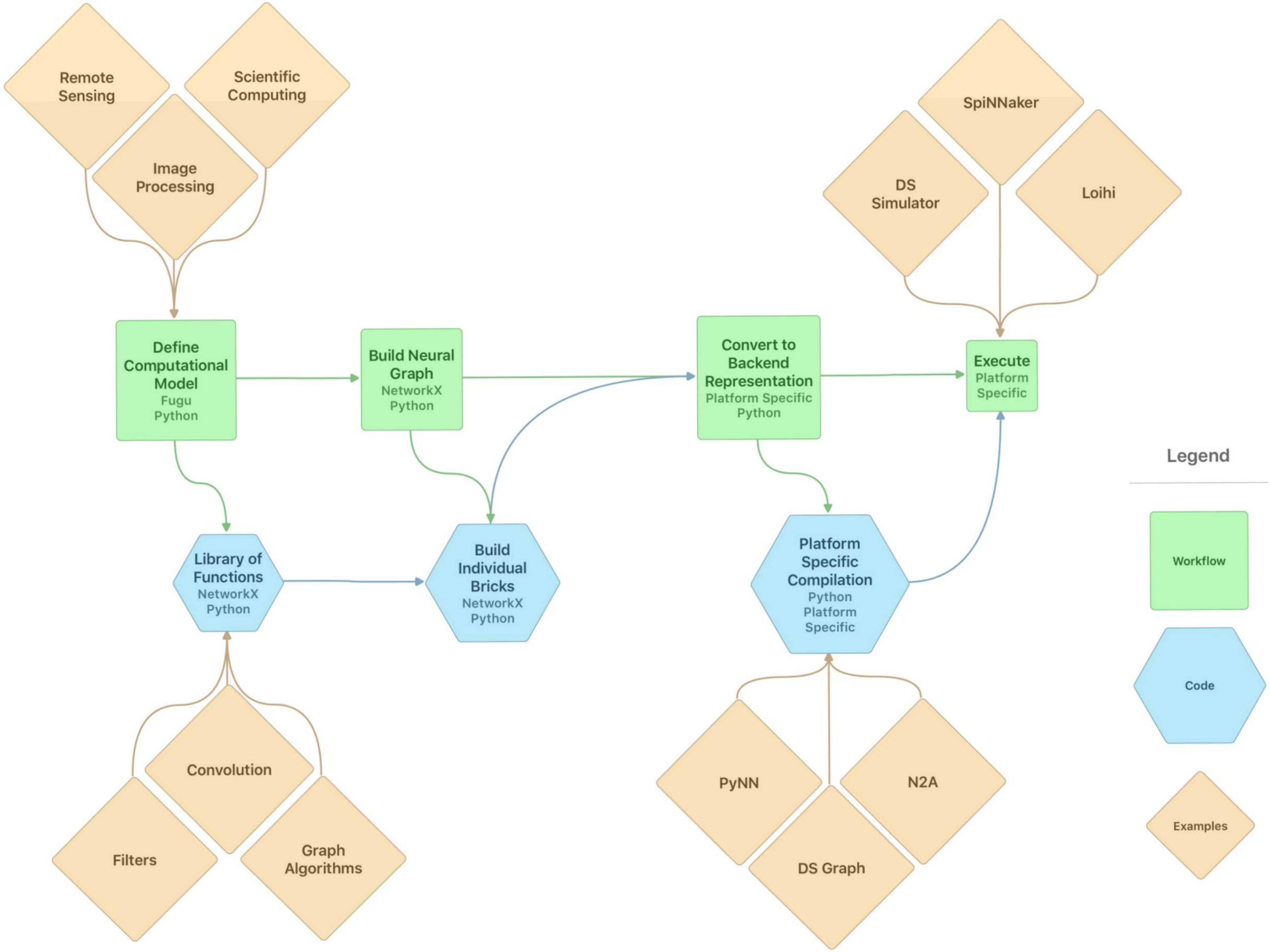
Fugu Overview

- **Problem**
  - Neuromorphic platforms remain a challenge to program
  - A unified, specific-hardware agnostic, framework to enable algorithm development are needed
- **Technical Approach**
  - Developing Fugu framework for linking existing spiking neural networks and expanding to solve scientific computing problems
  - Independent of the hardware that runs the neuron computation
  - Standard neural environment means easy interoperability
- **Results/Impact**
  - Working prototype is finished with basic 'bricks' in place
  - Multiple software backends completed; several hardware backends in development

We hope that Fugu will become a benchmarking **task** and a benchmarking **tool.**

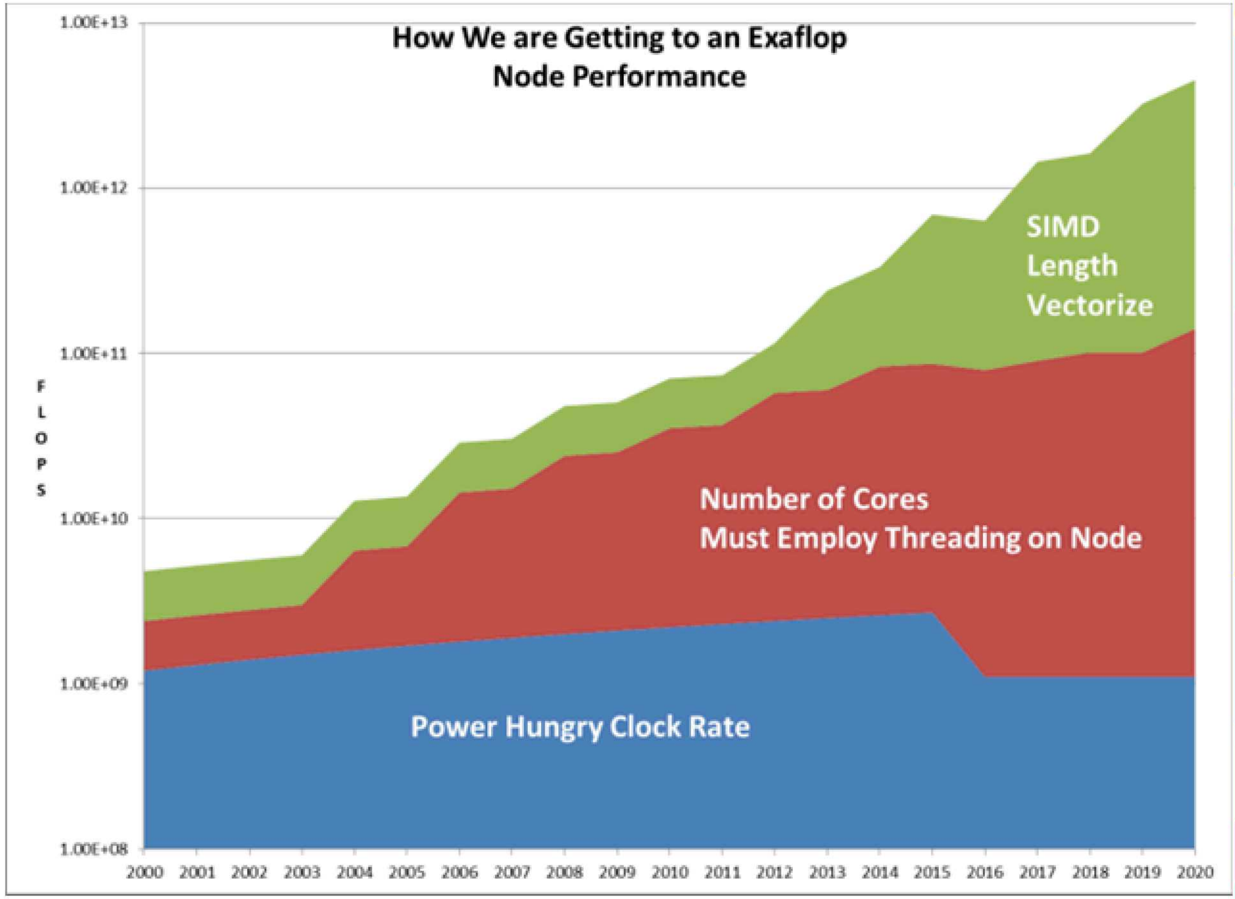


Fugu Overview



Good Benchmarking is a necessity!

- Performance of neuromorphic systems needs to be quantifiable for any large-scale adoption
- High-quality benchmarks allow customers to
  - Make informed decisions
  - Have reasonable expectations
  - Get excited about new technologies
- Within the field, benchmarks
  - Motivate development directions
  - Allow for honest comparisons
  - Measure progress
- Developing good benchmarks is hard!

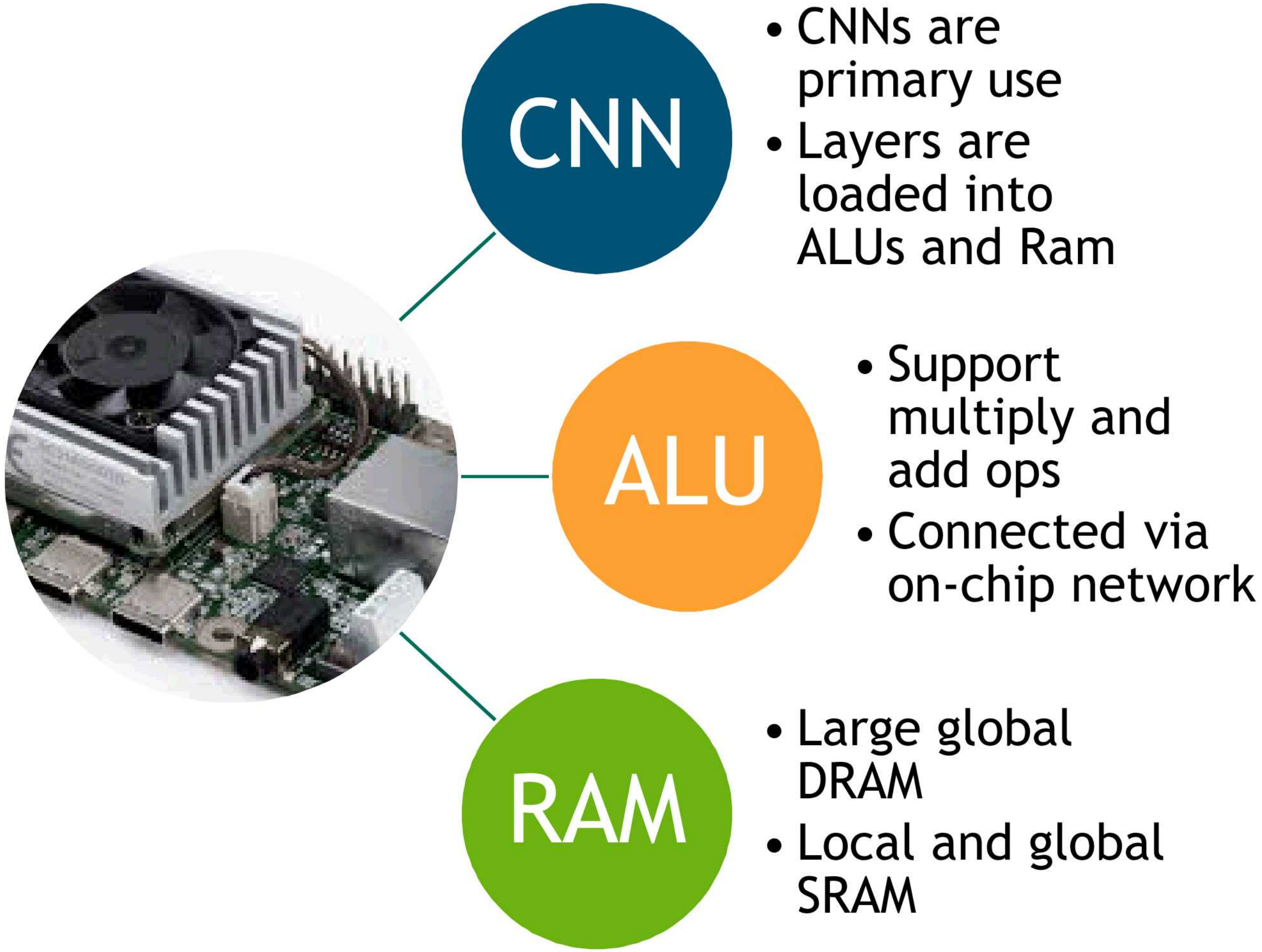


http://www.cray.com/blog

Let’s Learn from Existing Methods

Energy Models of DNN Acceleration Hardware

- Current software exists to estimate energy use
  - Tools are available *now*
    - Working ”out-of-the-box” with some catches
  - Currently focused on energy estimates of CNN acceleration hardware
    - (Google Coral, Systolic Arrays, etc.)
- State of the art uses properties of DNN hardware:
  - CNNs are primary use
  - CNN layers are loaded into ALUs & RAM
  - ALUs support multiply / add ops
  - Hardware has local and global SRAM cache
  - Global large DRAM
  - ALUs are connected via on-chip network



- CNNs are primary use
- Layers are loaded into ALUs and Ram

- Support multiply and add ops
- Connected via on-chip network

- Large global DRAM
- Local and global SRAM

Let's Learn from Existing Methods

Analytical Models

(Very fast; less accurate)

MAESTRO

Provides tool to estimate energy use given hardware & dataflow

Given a single layer of a CNN will generate estimates of:

- Latency
- Memory access stats
- Memory reuse
- And more hardware stats

Must have existing dataflow description and hardware design

Timeloop

Provides tool to find optimal dataflow given hardware

- Useful to benchmark different hardware designs

Given multiple layers of a CNN, will generate estimates of:

- Latency
- Memory accesses
- Memory reue
- A optimal or near optimal dataflow for the hardware

Needs an existing hardware design

Hardware Simulation Models

(Potentially more accurate; slower and more nuanced)

SCALE-Sim

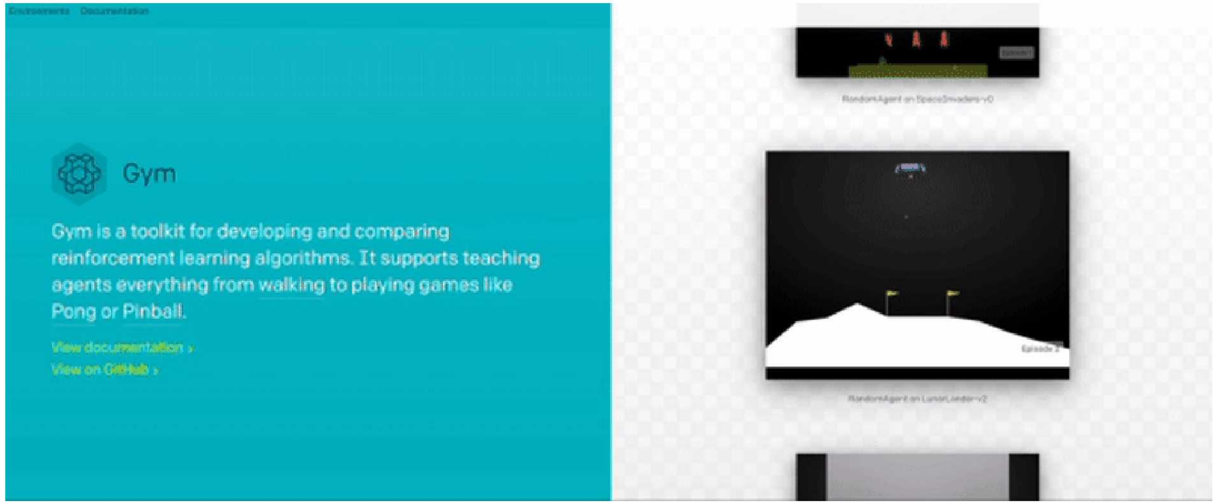
Simulates systolic array hardware at a cycle-accurate level

Given a CNN, produces

- DRAM read and write bandwidth access patterns
- Latency
- Cache access patterns at a cycle accurate level

Potential for integration into a system-level model

Let's Learn from Existing Methods



Gym

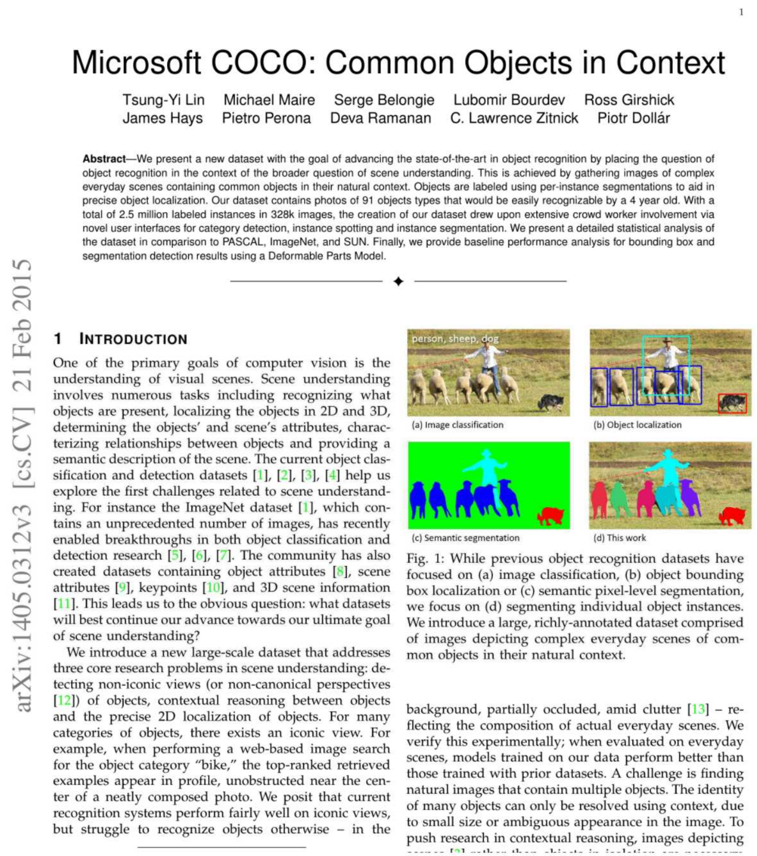
Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

http://gym.openai.com

MLPerf

Fair and useful benchmarks for measuring training and inference performance of ML hardware, software, and services.

http://mlperf.org



Microsoft COCO: Common Objects in Context

Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick
James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollár

Abstract—We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

1 Introduction

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects and providing a semantic description of the scene. The current object classification and detection datasets [1], [2], [3], [4] help us explore the first challenges related to scene understanding. For instance the ImageNet dataset [1], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and detection research [5], [6], [7]. The community has also created datasets containing object attributes [8], scene attributes [9], keypoints [10], and 3D scene information [11]. This leads us to the obvious question: what datasets will best continue our advance towards our ultimate goal of scene understanding?

We introduce a new large-scale dataset that addresses three core research problems in scene understanding: detecting non-iconic views (or non-canonical perspectives [12]) of objects, contextual reasoning between objects and the precise 2D localization of objects. For many categories of objects, there exists an iconic view. For example, when performing a web-based image search for the object category "bike," the top-ranked retrieved examples appear in profile, unobstructed near the center of a neatly composed photo. We posit that current recognition systems perform fairly well on iconic views, but struggle to recognize objects otherwise – in the background, partially occluded, amid clutter [13] – reflecting the composition of actual everyday scenes. We verify this experimentally; when evaluated on everyday scenes, models trained on our data perform better than those trained with prior datasets. A challenge is finding natural images that contain multiple objects. The identity of many objects can only be resolved using context, due to small size or ambiguous appearance in the image. To push research in contextual reasoning, images depicting

(a) Image classification (b) Object localization

(c) Semantic segmentation (d) This work

Fig. 1: While previous object recognition datasets have focused on (a) image classification, (b) object bounding box localization or (c) semantic pixel-level segmentation, we focus on (d) segmenting individual object instances. We introduce a large, richly-annotated dataset comprised of images depicting complex everyday scenes of common objects in their natural context.

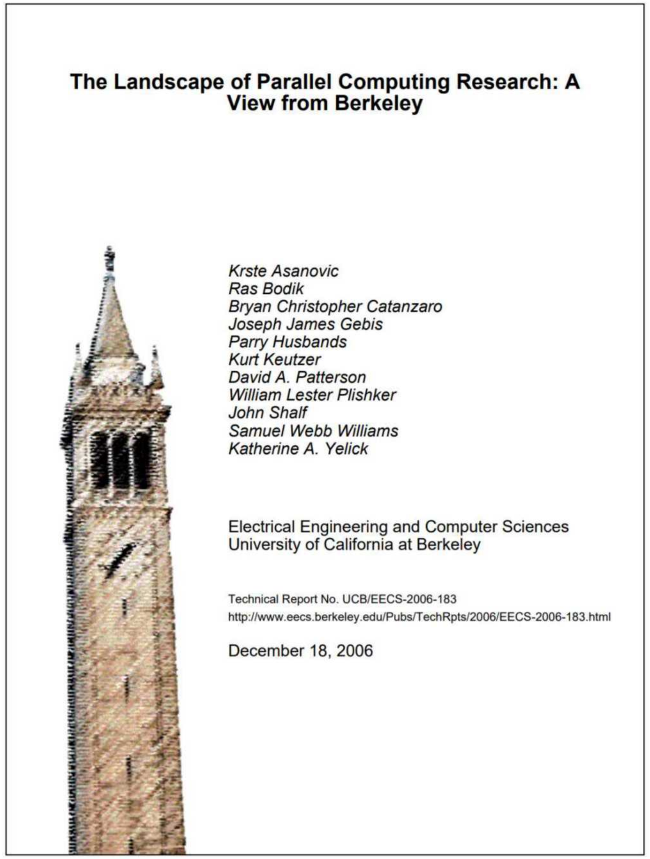
http://cocodataset.org

Seven Motifs of Scientific Computing

Delineate the seven basic workloads of scientific computing:

- Structured grids
- Unstructured grids
- Dense linear algebra
- Sparse linear algebra
- Fast Fourier transforms
- Particles
- Monte Carlo

Phil Colella. 2004. Defining Software Requirements for Scientific Computing



The Landscape of Parallel Computing Research: A View from Berkeley

Krste Asanovic
Ras Bodik
Bryan Christopher Catanzaro
Joseph James Gebis
Parry Husbands
Kurt Keutzer
David A. Patterson
William Lester Plishker
John Shalf
Samuel Webb Williams
Katherine A. Yelick

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-183
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

December 18, 2006

3.1 Seven Dwarfs

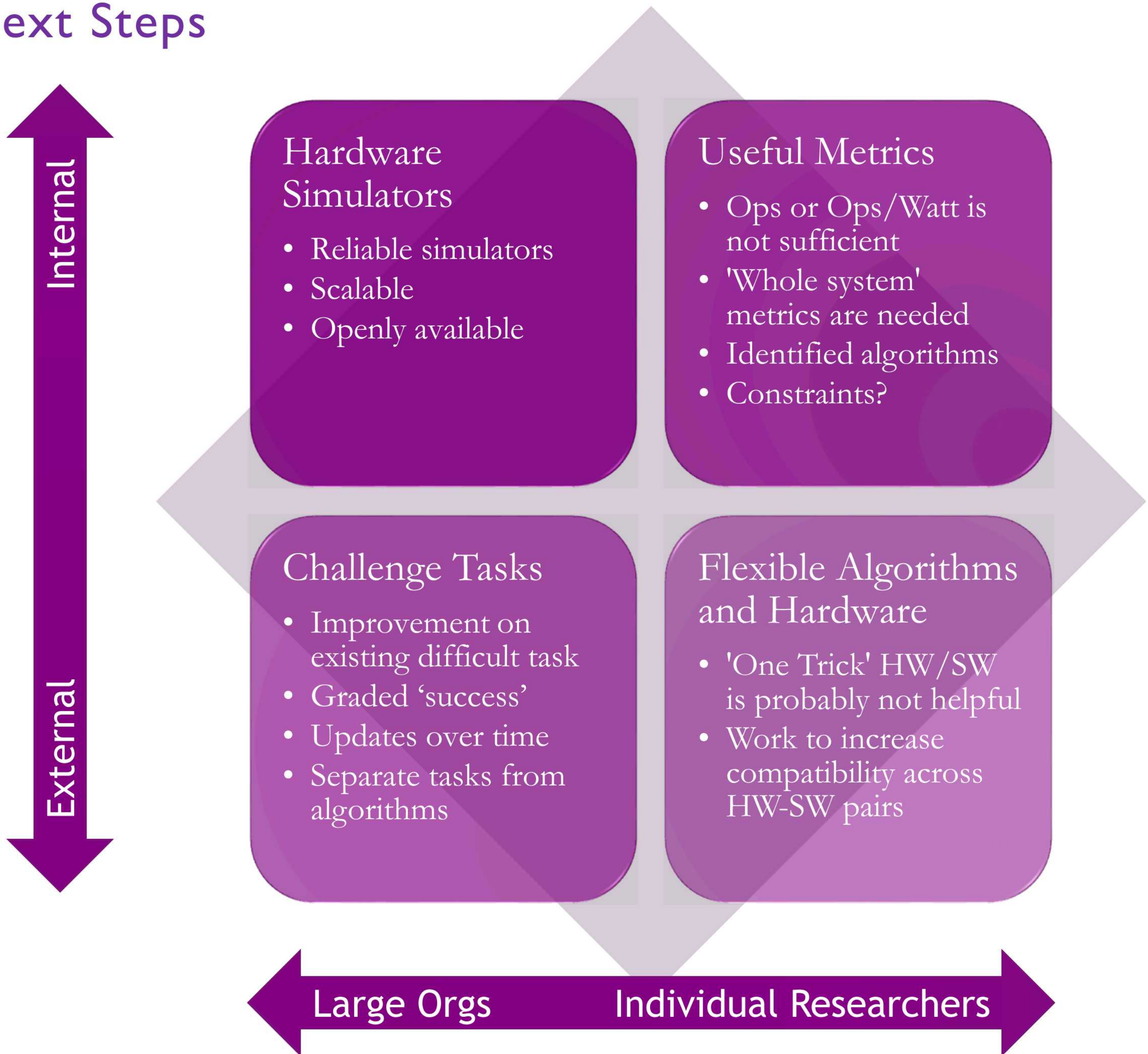
3.2 Finding More Dwarfs

3.3 Composition of Dwarfs

Cognitive Workloads

- Feed-forward sensory processing
  - Classification or regression from potentially multimodal sensory inputs. Modes may include vision, audio, tactile, sonar, radar, or other more abstract data like sales transaction information.
- Time-dependent sensory processing
  - Memory is needed in systems with time-dependent behavior. Simple memory is often achieved through network recurrency. Workloads could exhibit partial observability to enforce time-dependency.
- Bayesian neural algorithms
  - Requires possibly multi-modal sensory processing to efficiently solve tasks with information aggregated and processed over time.
- Dynamical memory and control algorithms
  - High-dimensional control problems. E.g. Hand is able to hold 100 lb. bow and thread a needle.
- Cognitive inference algorithms, self-organizing algorithms and beyond
  - E.g. Using neurogenesis to solve ever-changing tasks, making use of past knowledge and skills.

Ideas for Next Steps



Internal

External

Large Orgs

Individual Researchers

Hardware Simulators

- Reliable simulators
- Scalable
- Openly available

Useful Metrics

- Ops or Ops/Watt is not sufficient
- 'Whole system' metrics are needed
- Identified algorithms
- Constraints?

Challenge Tasks

- Improvement on existing difficult task
- Graded ‘success’
- Updates over time
- Separate tasks from algorithms

Flexible Algorithms and Hardware

- 'One Trick' HW/SW is probably not helpful
- Work to increase compatibility across HW-SW pairs