The Future Begins Here

intel labs

Labs Day 2020 | December 3

The Future Begins Here

intel labs

Early Benchmarking Results for Neuromorphic Computing

Mike Davies
Senior Principal Engineer and Director,
Neuromorphic Computing Lab

Labs Day 2020

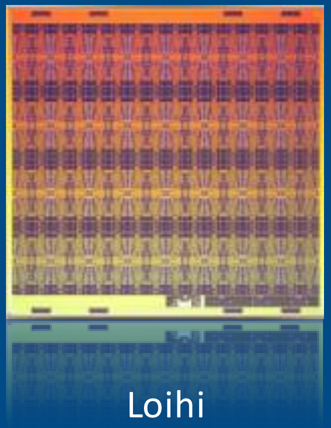
Rethinking Computing Bottom-Up

Compute Efficiency (EDP)
- Compute-memory integration
- Local learning rules
- Sparse temporal activity (aka Spikes)
- Sparse connectivity with fine-grain parallelism
- 3D wiring
- Temporal data coding
- Exploiting material time constants
- Dendritic nonlinearities
- Low precision
- Hybrid analog/digital computation

Algorithmic
- Distributed data representations
- Integration
- Sparse temporal activity (aka Spikes)
- Online causal adaptation
- Very high fanout
- Recurrence and feedback loops
- Oscillatory interaction
- Continuous operation
- Diverse time scales
- Stochasticity
- Parametric Heterogeneity

Resource Efficiency
- Self-organized growth
- Autonomous healing
- Dendritic nonlinearities
- Low precision
- Analog persistent state
- 3D wiring
- Exploiting material time constants

The Brain
1,400,000 mm^3
80B neurons



Loihi
60 mm^3
128K neurons

Loihi Characteristics
Compute and Memory Integrated to spatially embody programmed networks
Temporal Neuron Models (LIF) to exploit temporal correlation
Spike-Based Communication to exploit temporal sparsity
Sparse Connectivity for efficient dataflow and scaling
On-Chip Learning without weight movement or data storage
Digital Asynchronous Implementation for power efficiency, scalability, and fast prototyping
No floating-point numbers, No multiply-accumulators No batching, No off-chip DRAM

Nature Machine Intelligence Reference: Benchmarks for Progress in Neuromorphic Computing

comment

Benchmarks for progress in neuromorphic computing

In order for the neuromorphic research field to advance into the mainstream of computing, it needs to start quantifying gains, standardize on benchmarks and focus on feasible application challenges.

Mike Davies

No one would accuse neuromorphic computing of lacking ambition. With a mission to decipher the multitude of secrets nature deploys to achieve unrivalled efficiency and flexibility in brain-based computing, the field faces one of the most daunting challenges in all of computational science and engineering.

Even with the brain as a guide, reverse engineering such a complex system remains an open-ended and highly unconstrained problem. We must reinvent our understanding of computing from the logic gates up, replacing synchronous sequencing and monolithic memory with millions of parallel dynamical units all communicating with asynchronous spike messages. Familiar programming models are replaced with high-dimensional, temporal and nonlinear computational abstractions yet to be fully comprehended.

In order to deliver on such an ambitious agenda, the neuromorphic field needs to focus more on principles and rigour, less on open-ended exploration and mapping speculative mechanistic features to silicon. Steady progress to real-world value depends on quantitative metrics, discipline and informed prioritization.

Not so fast?

Benchmarking is the traditional methodology for measuring and focusing progress in engineering. In many fields, benchmarks have provided great value, from Dhrystone and SPECint for microprocessor innovation to ImageNet for convolutional deep learning progress more recently. A good benchmark serves to motivate researchers to solve one particular problem chosen as a worthy representative of a broader class of useful problems.

Yet, in a nascent and fragmented field such as neuromorphic computing, some reservation for benchmarking is warranted. An ill-chosen benchmark can focus disproportionate attention on just one piece of the larger puzzle, with the effect of impeding rather than accelerating broader progress.

An example where this is already happening relates to measurements of 'synaptic ops' that are commonly reported for neuromorphic designs. Such microscopic metrics are easy to measure but offer little value for assessing the worth of a particular neuromorphic chip. If a chip hasn't demonstrated even a single meaningful workload, then such an indirect energy measurement should be taken with a grain of salt. On the other hand, if a chip can support meaningful workloads, then it should be measured on those terms. As long as the field has yet to determine the right architectural features and degree of programmability, the emphasis should be on assessing comprehensive workloads, not on microscopic circuit properties. Fixating on readily optimized synaptic op metrics puts the cart before the horse.

This concern about misdirection of focus is also what troubles some who fear 'benchmarking' is code for adopting the particular benchmarks that have guided the deep learning community. While neuromorphic researchers universally use the MNIST dataset to test pattern classification algorithms, consensus disappears on whether more advanced vision datasets such as CIFAR-10 and ImageNet should immediately follow.

At root is unease with the idea that the goal of neuromorphic learning should be the 'training' paradigm that starts from a tabula rasa network state and learns by ingesting a single dataset, in one computationally intensive leap. Many neuromorphic researchers, with human learning in mind, would prefer to develop incremental, hierarchical and factorization-based approaches. Humans don't learn new concepts from hundreds of examples presented in all conceivable contexts; humans learn from single examples by leveraging prior learning. From this perspective, it's unwise to focus on datasets that presuppose a tabula rasa approach.

Another benchmarking challenge arises in the handling of thorny environment modelling issues present in robotics, closed-loop control and active sensing problems. These challenges have stymied standardization even for conventional approaches and can be exacerbated in the neuromorphic domain. Specifically, some neuromorphic systems use analog circuits with real-world time constants and therefore can only run at one particular speed — no slower or faster. These challenges are surmountable, but more thought and care must be devoted to the problem — for example, with a minimal simulation methodology satisfying variability bounds[1].

The need for compelling benchmarks

Despite the challenges and legitimate concerns, the time has nevertheless come for the neuromorphic field to embrace an appropriate set of benchmarks, specifically on two fronts: first, internally oriented, as a way to measure the capabilities of different spiking neuromorphic architectures; and second, in an outward sense, problems that quantify the value of neuromorphic solutions compared to state-of-the-art conventional solutions.

On the first front, the field needs a comprehensive suite of spiking neural network algorithms analogous to SPECint or MLPerf. We might call this SpikeMark. It's important to bear in mind that different neuromorphic architectures are far more varied than the different von Neumann processors for which SPECint was designed in the 1990s, and support a far more varied set of algorithms than the numerous variants of backpropagation training that MLPerf measures. One further hurdle is that there is no standardized language for neuromorphic programming, such as C for SPECint. Nevertheless, these challenges can be addressed with good written specifications and conventionally coded descriptions of ground truth.

The purpose of such a SpikeMark benchmarking suite (Box 1) would be to evaluate the relative features, flexibility, performance and efficiency of different neuromorphic platforms. It would include both applications suitable for real-world

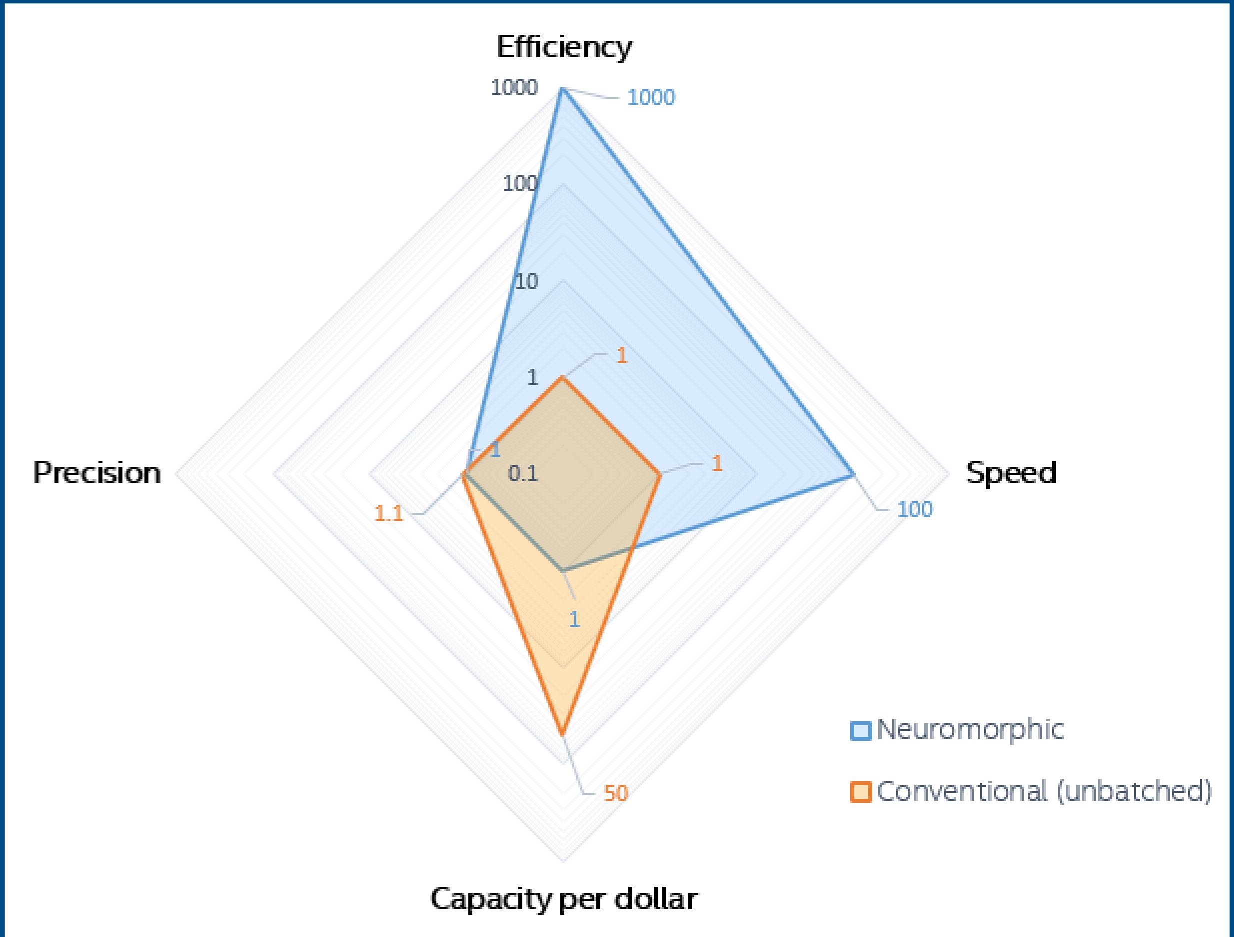
for example, with a minimal simulation
methodology satisfying variability bounds[1].

The need for compelling benchmarks
Despite the challenges and legitimate
concerns, the time has nevertheless come
for the neuromorphic field to embrace an
appropriate set of benchmarks, specifically
on two fronts: first, internally oriented, as a
way to measure the capabilities of different
spiking neuromorphic architectures; and

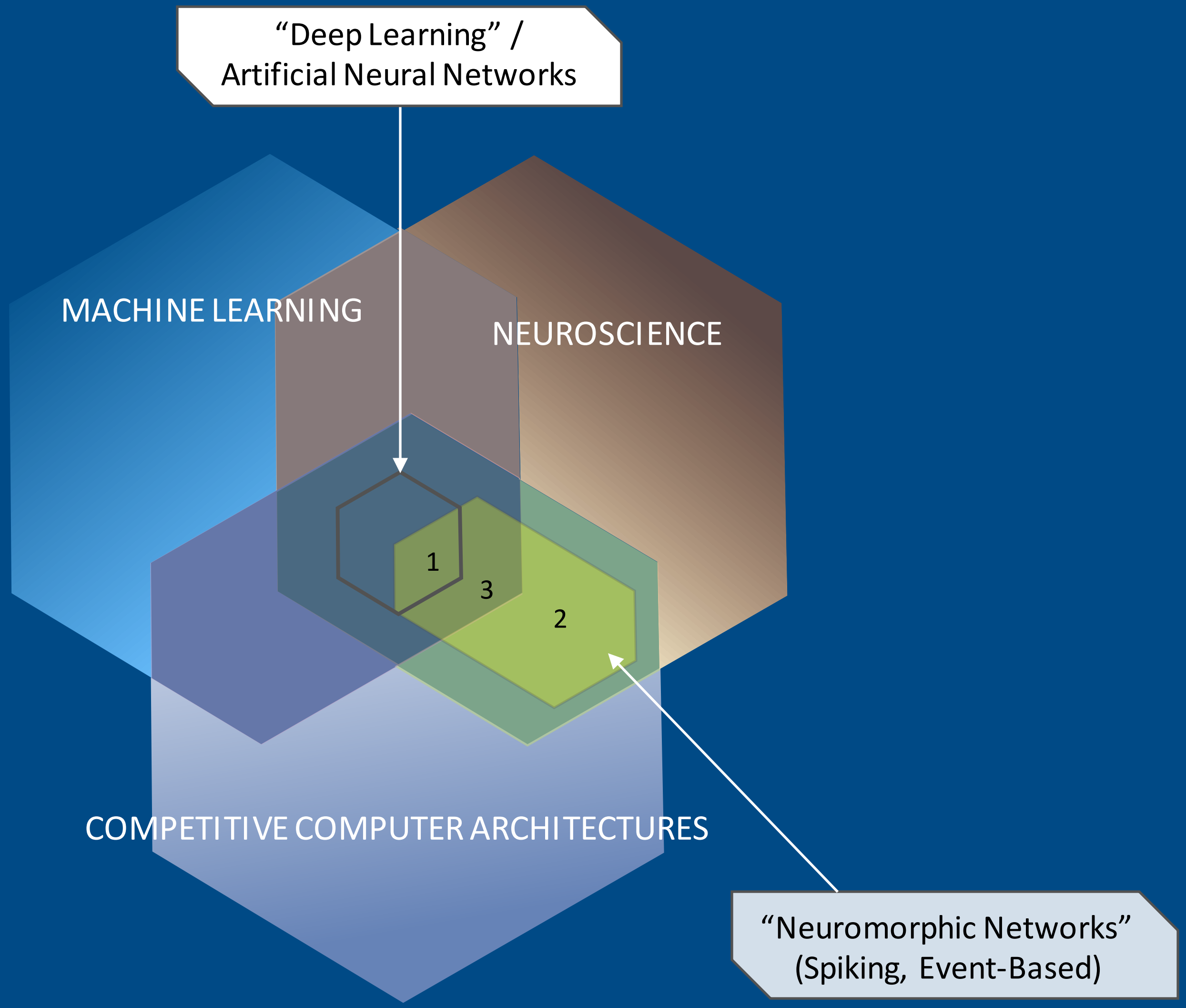
Nature Machine Intelligence, Vol 1, Sept 2019.

Seeking Order of Magnitude Gains

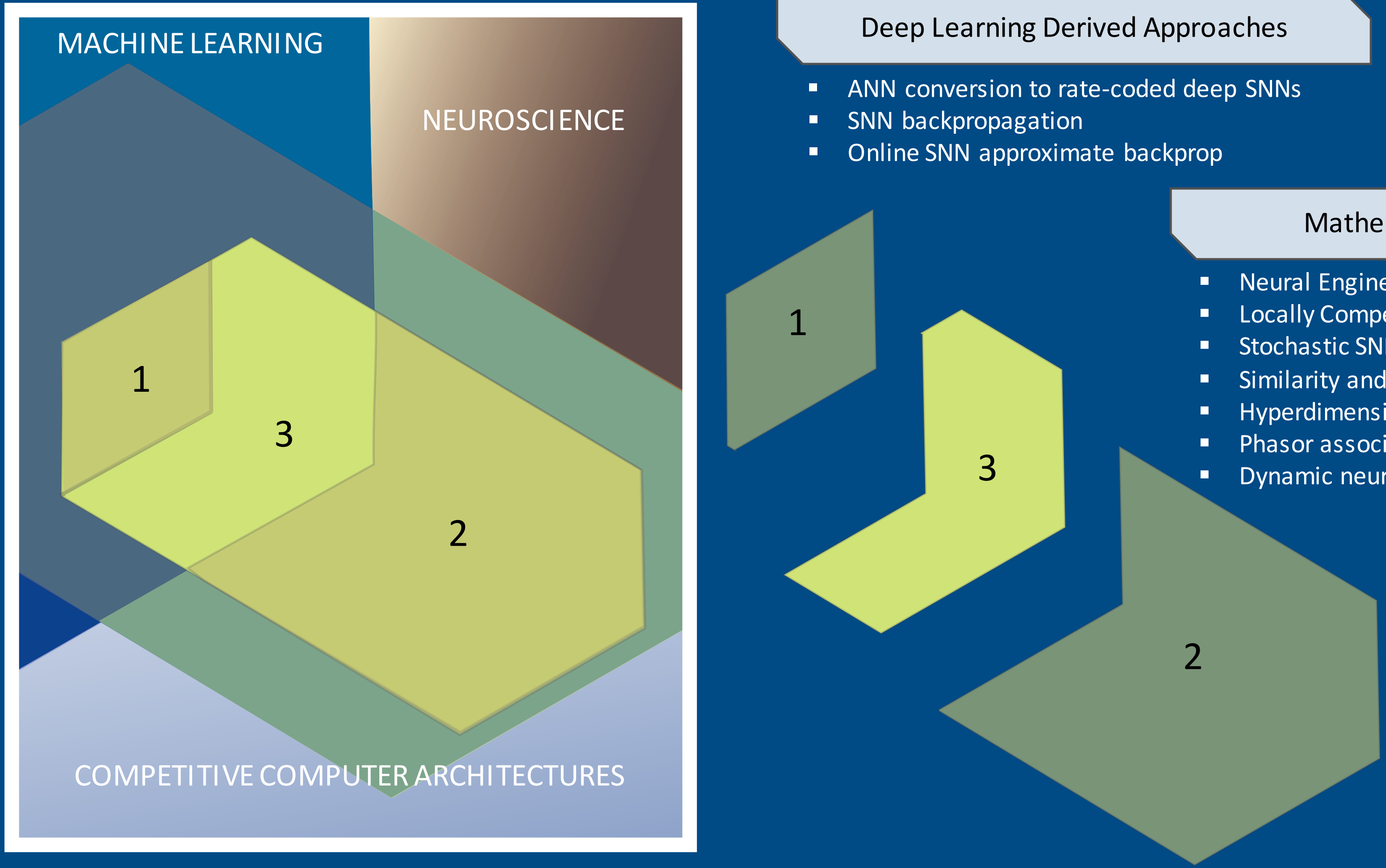
- In energy efficiency
- In speed of processing data – especially signals arriving in real time
- In the data efficiency of learning and adaptation
- With programmability to span a wide range of workloads and scales
- With long-term plans to reduce cost with process technology innovations



The Challenge: SNN Algorithm Discovery



The Challenge: Algorithm Discovery



Deep Learning Derived Approaches

- ANN conversion to rate-coded deep SNNs
- SNN backpropagation
- Online SNN approximate backprop

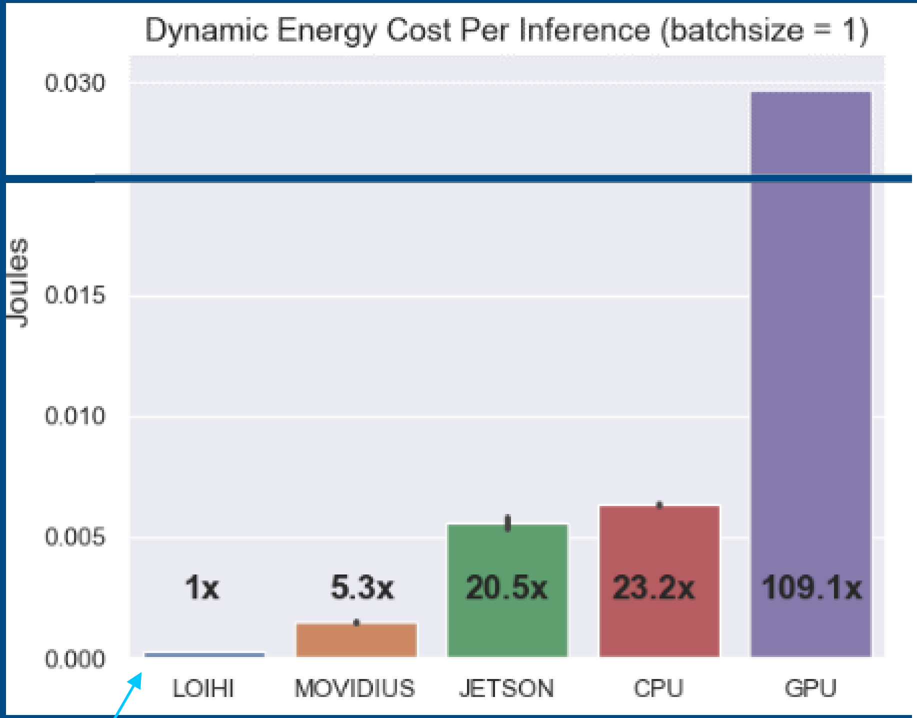
Mathematically Formalized

- Neural Engineering Framework (NEF)
- Locally Competitive Algorithm for LASSO
- Stochastic SNNs for solving CSPs
- Similarity and graph search with temporal spike codes
- Hyperdimensional computing
- Phasor associative memories
- Dynamic neural fields and continuous attractor networks

New Ideas Guided by Neuroscience

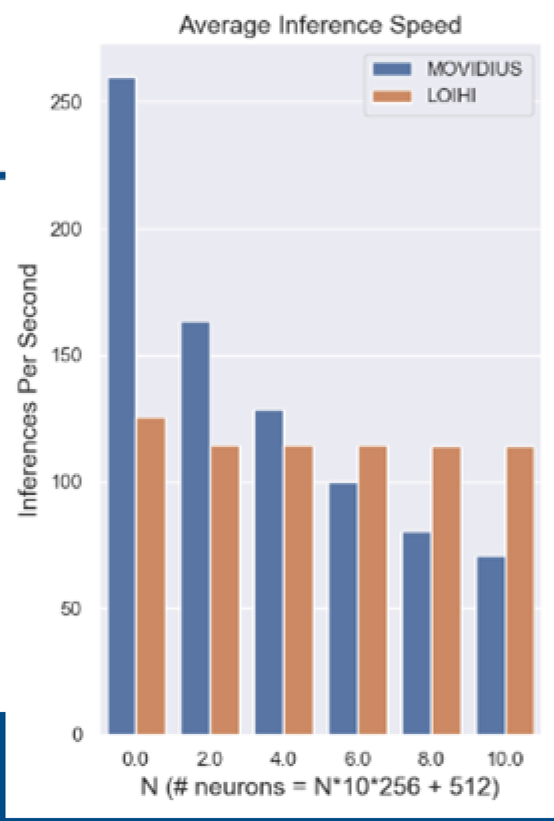
- Olfaction-inspired rapid learning
- "RatSLAM" for mapping and navigation
- Cortical microcircuit models
- Evolutionary optimization of SNNs

Deep Network Conversion for Keyword Spotting

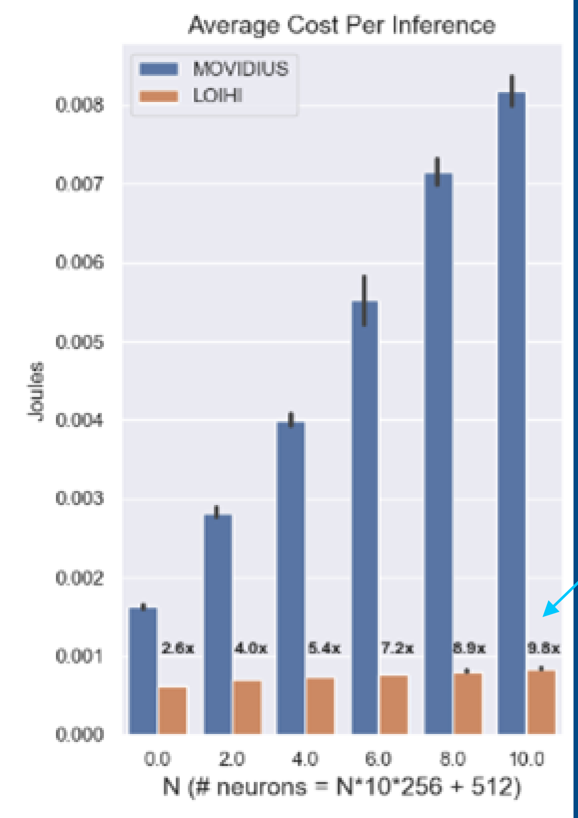


(Lower is better)

Loihi is the most energy-efficient architecture for real-time inference (batchsize=1 case)



(Higher is better)



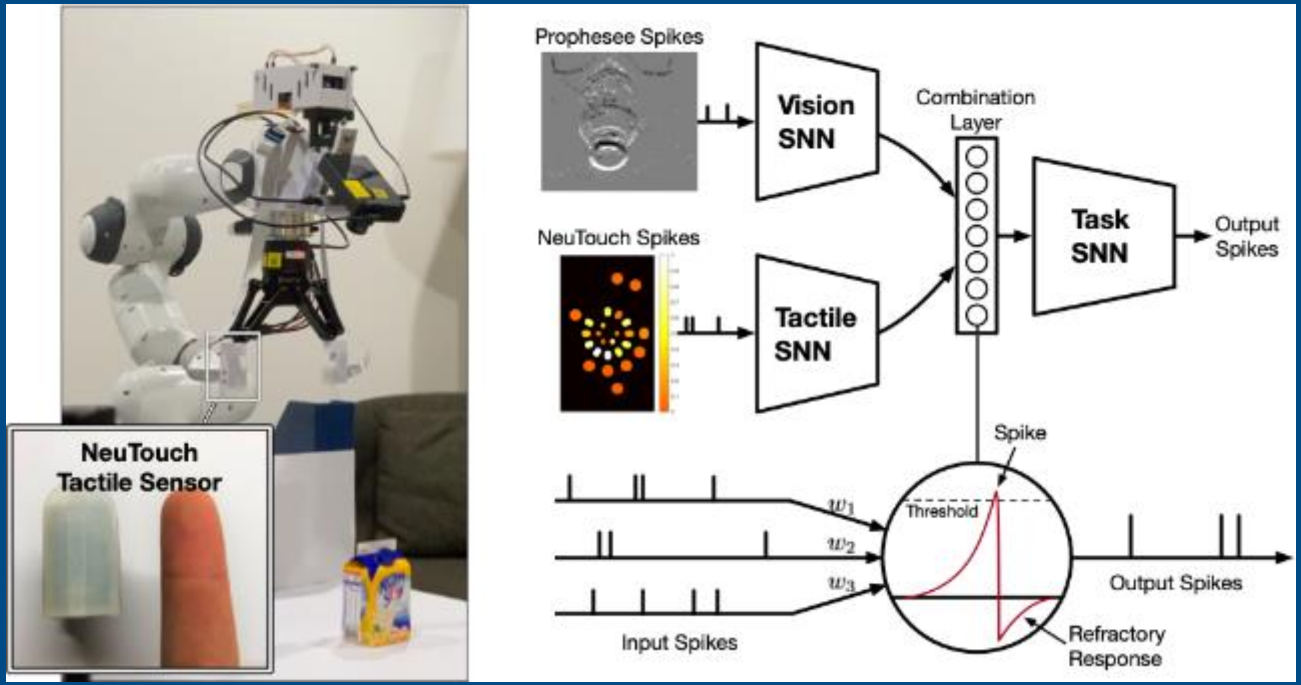
(Lower is better)

Loihi provides extremely good scaling vs conventional architectures as network size grows by 50x

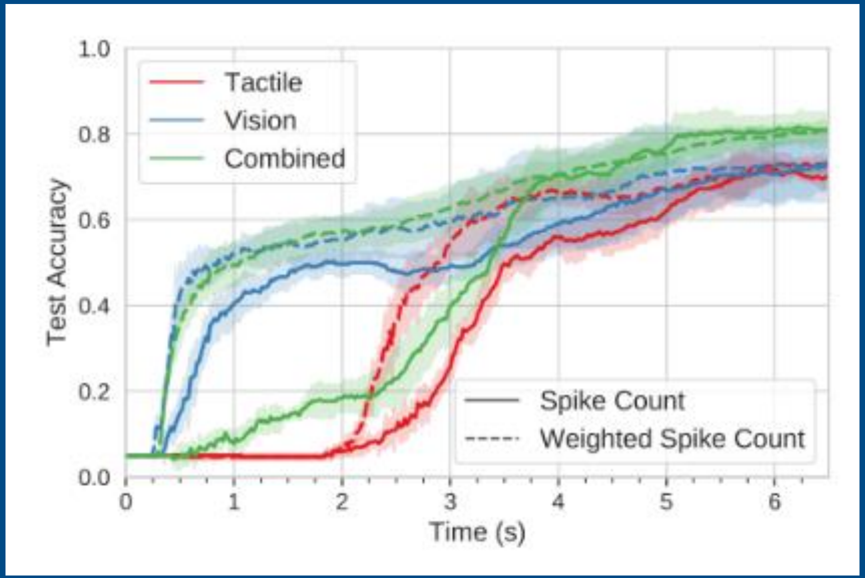
Loihi consumes 5-10x lower energy than closest conventional DNN architecture

For workloads, configurations, and results, see Blouw et al, "Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware." arXiv:1812.01739. Results May Vary.

Directly Trained SNNs for Event-based Vision + Tactile Sensing



Object Classification



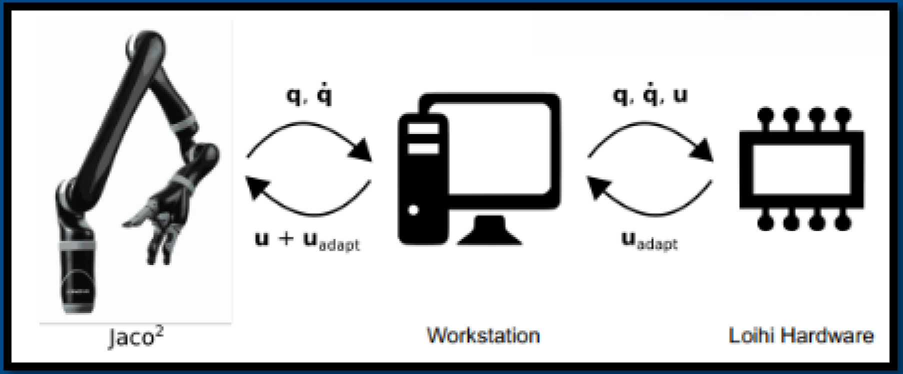
Loihi outperforms on all metrics vs GPU[1]:

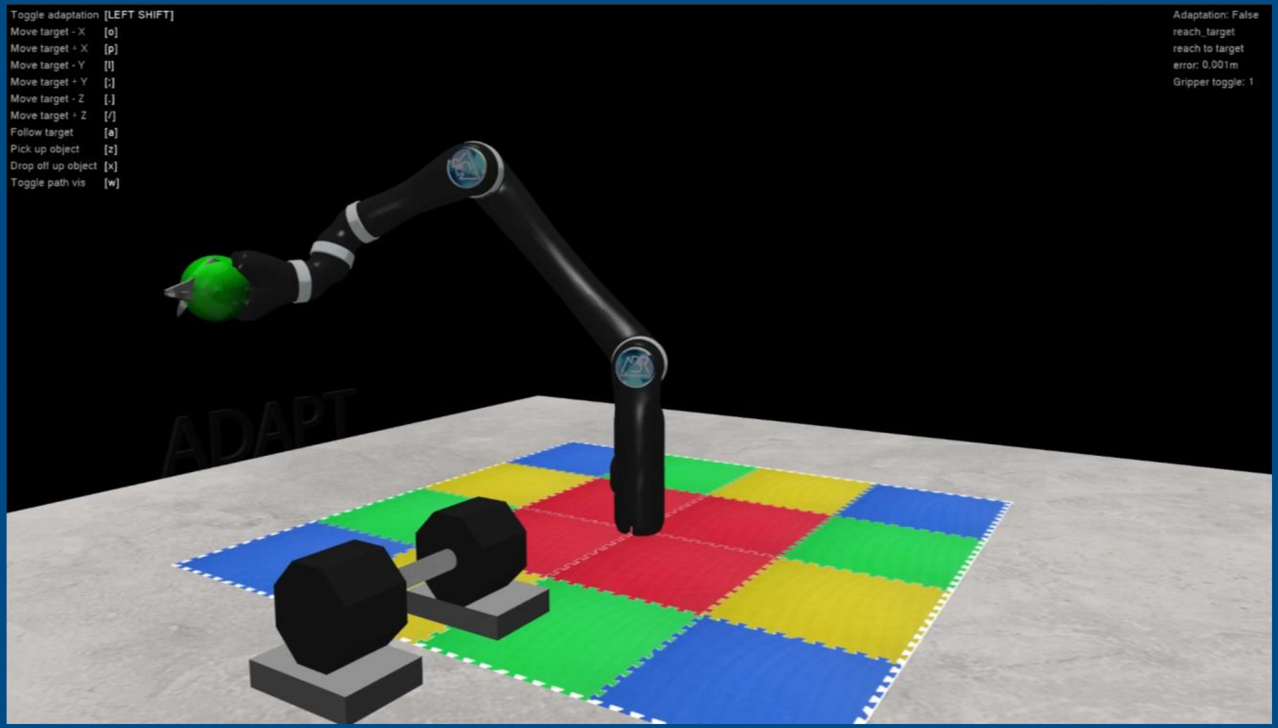
- 20% faster
- 45x lower power

[1] For workloads, configurations, and results, see Event-Driven Visual-Tactile Sensing and Learning for Robots Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin Tee, and Harold Soh Robotics: Science and Systems Conference (RSS), 2020. Results may vary.

Adaptive Control of a Robot Arm Using Loihi

- SNN adaptive dynamic controller implemented on Loihi allows a robot arm to adjust in real time to nonlinear, unpredictable changes in system mechanics[1, 2]
- Loihi outperforms with **40x lower power**, **2x faster control rate** compared to a GPU[3]



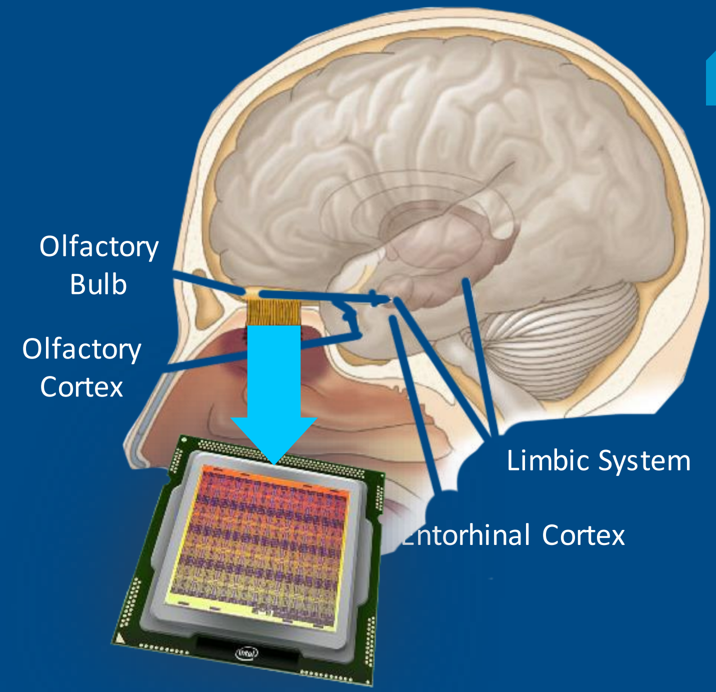


[1] DeWolf, T., Stewart, T. C., Slotine, J. J., & Eliasmith, C. (2016, November). A spiking neural model of adaptive arm control. In *Proc. R. Soc. B* (Vol. 283, No. 1843, p. 20162134). The Royal Society.

[2] Eliasmith, "Building applications with next generation neuromorphic hardware." *NICE Workshop 2018*

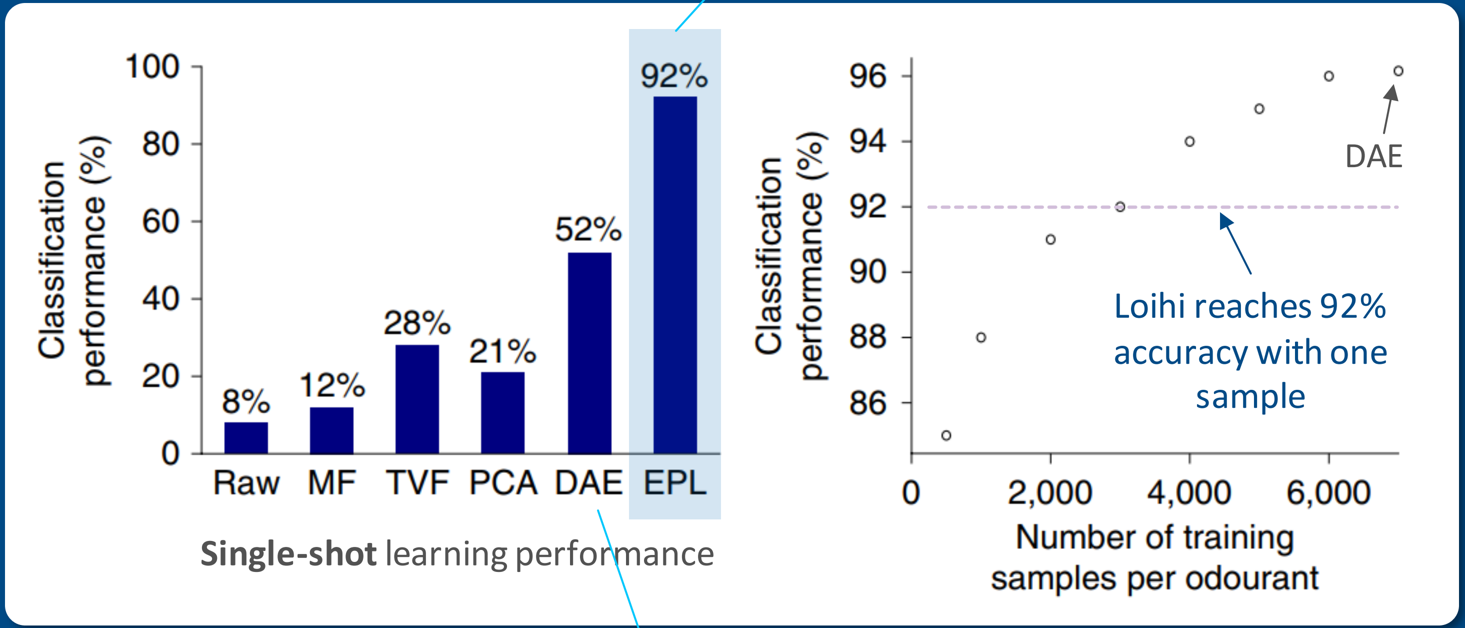
[3] DeWolf, T., Jaworski, P., Eliasmith, C. (2020). Nengo and Low-Power AI Hardware for Robust, Embedded Neurorobotics. Frontiers in Neurorobotics. Results may vary.

An Example 3000x More Data Efficient than DL



Bio-inspired odor learning and recognition

Loihi's bio-inspired algorithm



Deep Learning solution (deep autoencoder)

Nabil Imam and Thomas Cleland, Nature Machine Intelligence, March 2020

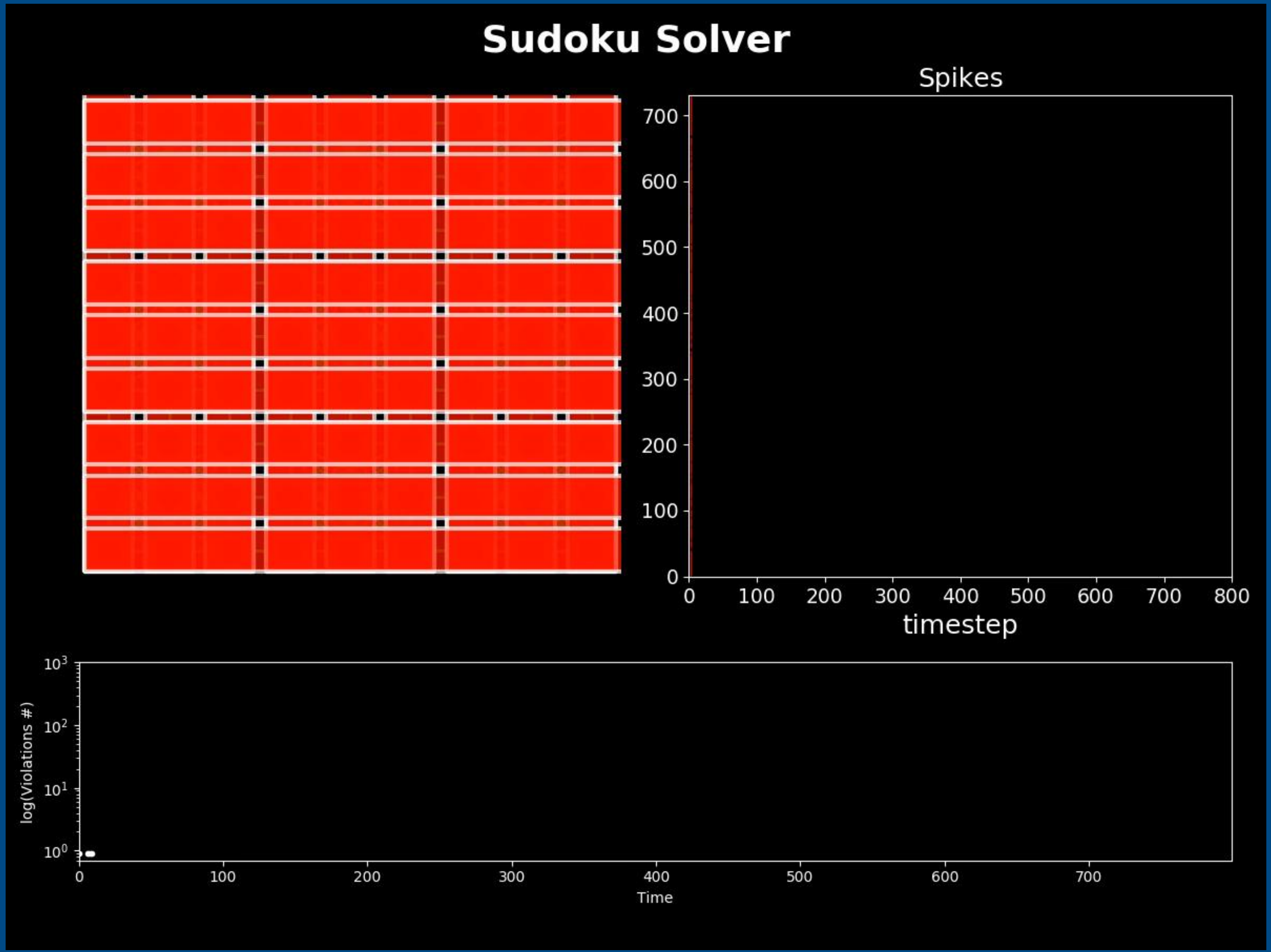
Optimization, Planning, Constraint Satisfaction

Problems solved by Loihi to date:

- LASSO regression
- Graph search (Dijkstra)
- Constraint Satisfaction (CSP)
- Boolean satisfiability (SAT)

Benefits:

- Over 10^5 times lower energy-delay-product for solving constraint satisfaction problems vs CPU[1]
- Up to 100x faster graph search[2]
- Even greater gains for LASSO

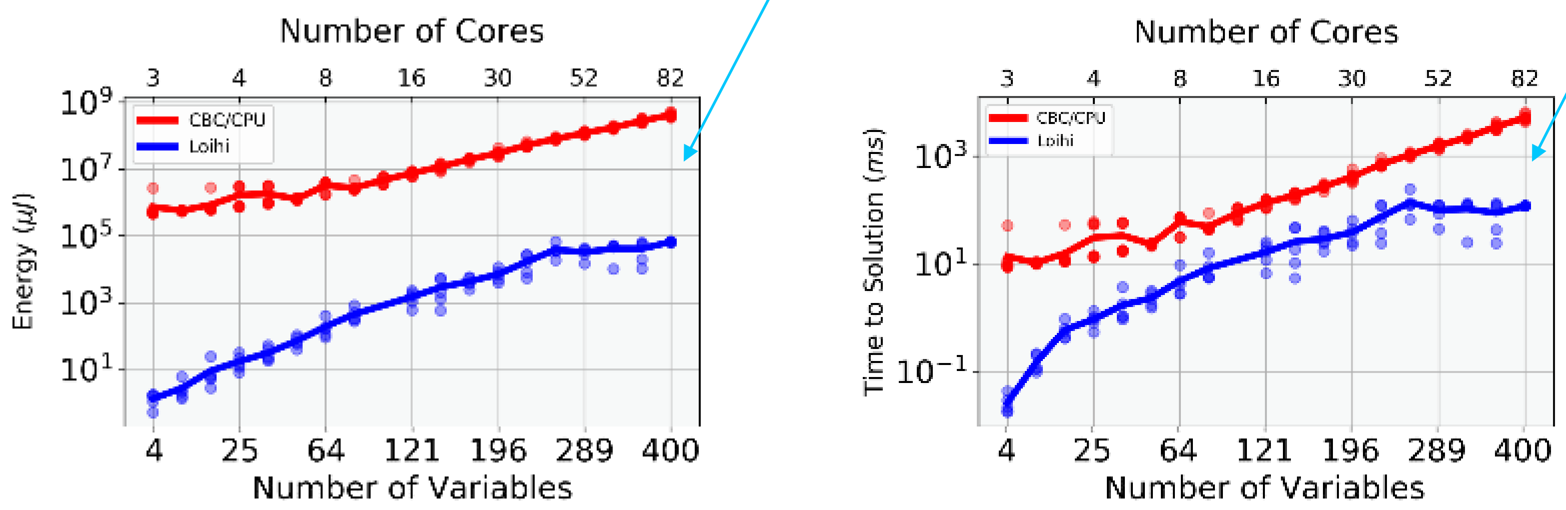


Loihi: Nahuku 32-chip system with NxSDK 0.98
CPU: Core i7-9700K w/ 32GB RAM running [1] [Task 13] Coin-or branch and cut (https://github.com/coin-or/Cbc) or [2] [Task 12] NetworkX (for graph search)
See backup for additional test configuration details. Performance results are based on testing as of July 2020 and may not reflect all publicly available security updates. Results may vary.

Latin Squares Solver: Quantitative Results

Over 2,500x lower energy

Over 40x faster



(Lower is better)

(Lower is better)

[Task 13]
CBC/CPU: Core i7-9700K w/ 32 GB RAM running Coin-or branch and cut (https://github.com/coin-or/Cbc)
Loihi: Nahuku 32-chip system with NxSDK 0.98
See backup for additional test configuration details. Performance results are based on testing as of July 2020 and may not reflect all publicly available security updates. Results may vary.

SLAM (Simultaneous Localization and Mapping)

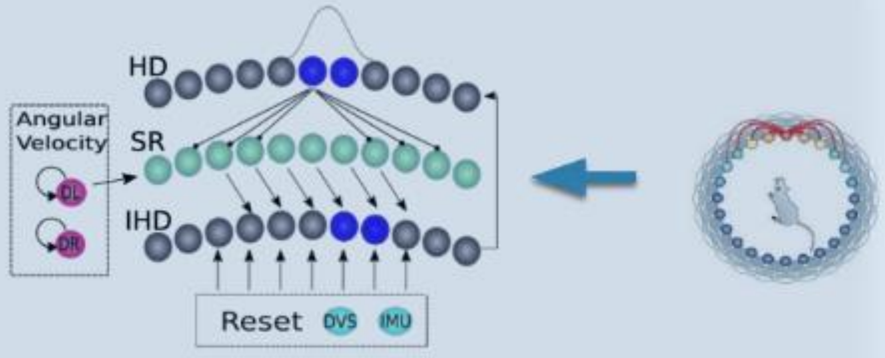
Fundamental task for any device (robot, AR glasses)
that needs to autonomously acquire spatial awareness

Neuromorphic components:

- 1D attractor ring(s) for pose estimation
- 2D position network ("place cells")
- Map learning
- Loop closure

Demonstrated on Loihi to date:

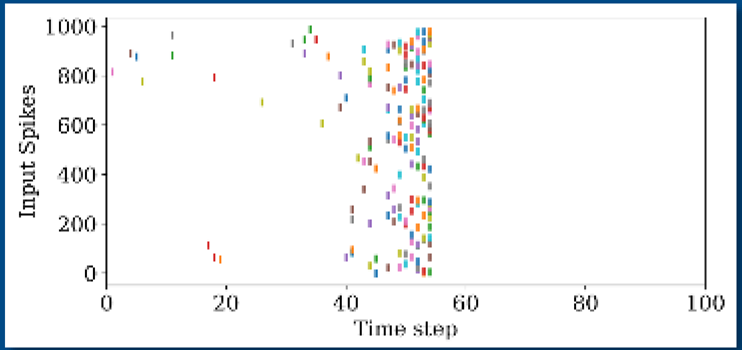
- Basic proof-of-concept functionality
- 100x lower dynamic power vs GMapping library on CPU[1]



[1] [Task 10] For workloads, configurations, and results, see Tang, G., Shah, A., & **Michmizos, K. P.** (2020). *Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM*. 4176–4181. https://doi.org/10.1109/iros40897.2019.8967864. Results may vary.

Nearest Neighbor Search on Pohoiki Springs

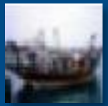
Input image:



Output(s):

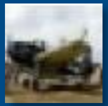
1

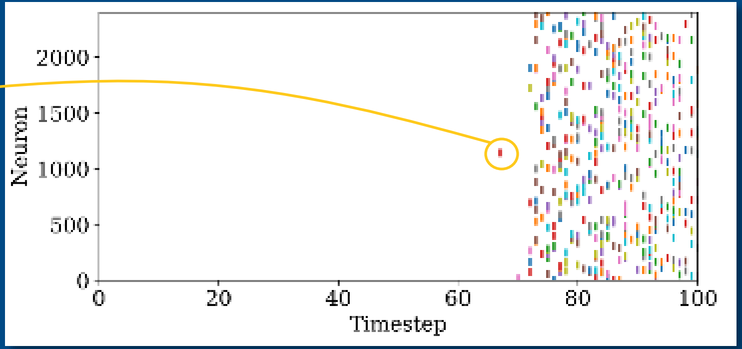
2



3

4





Lesser matches indicated by later spikes

k-NN on Loihi:

- Novel use of fine-grain parallelism and sparse temporal matching and searching
- 1+ million pattern datasets
- Up to 1k search key dimensionality

Benefits:

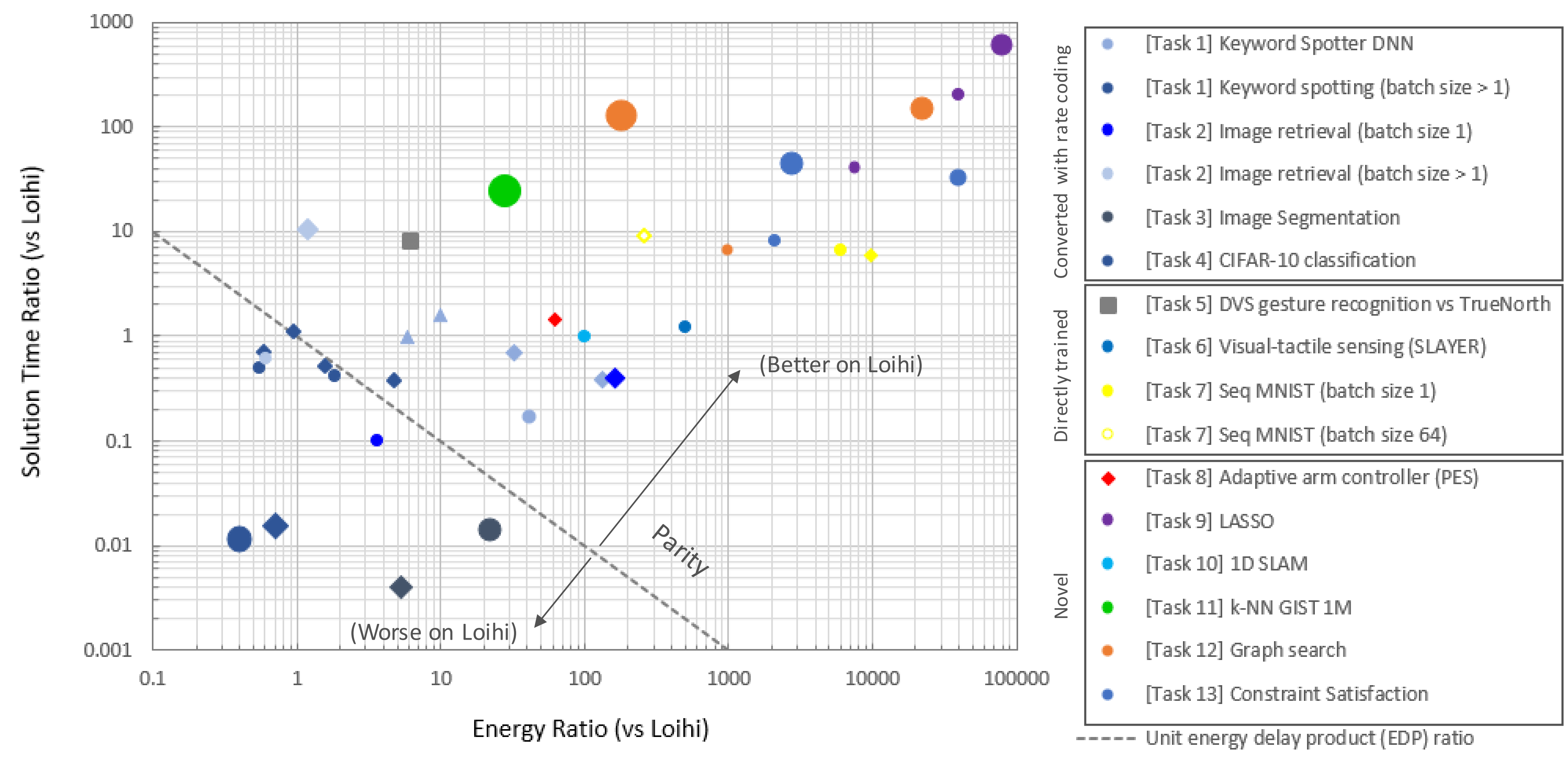
- Up to **4x faster latency** or **80-300x faster index generation** than state-of-the-art CPU implementations
- Supports adding new patterns online in **milliseconds**
- **650x better energy-delay-product** compared to CPU implementation

[Task 11] For workloads, configurations, and results, see EP Frady et al, "Neuromorphic Nearest-Neighbor Search Using Intel's Pohoiki Springs." arXiv:2004.12691. Results may vary.

For the Right Workloads, Loihi Provides Orders of Magnitude Gains in Latency and Energy

Reference architecture

- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth

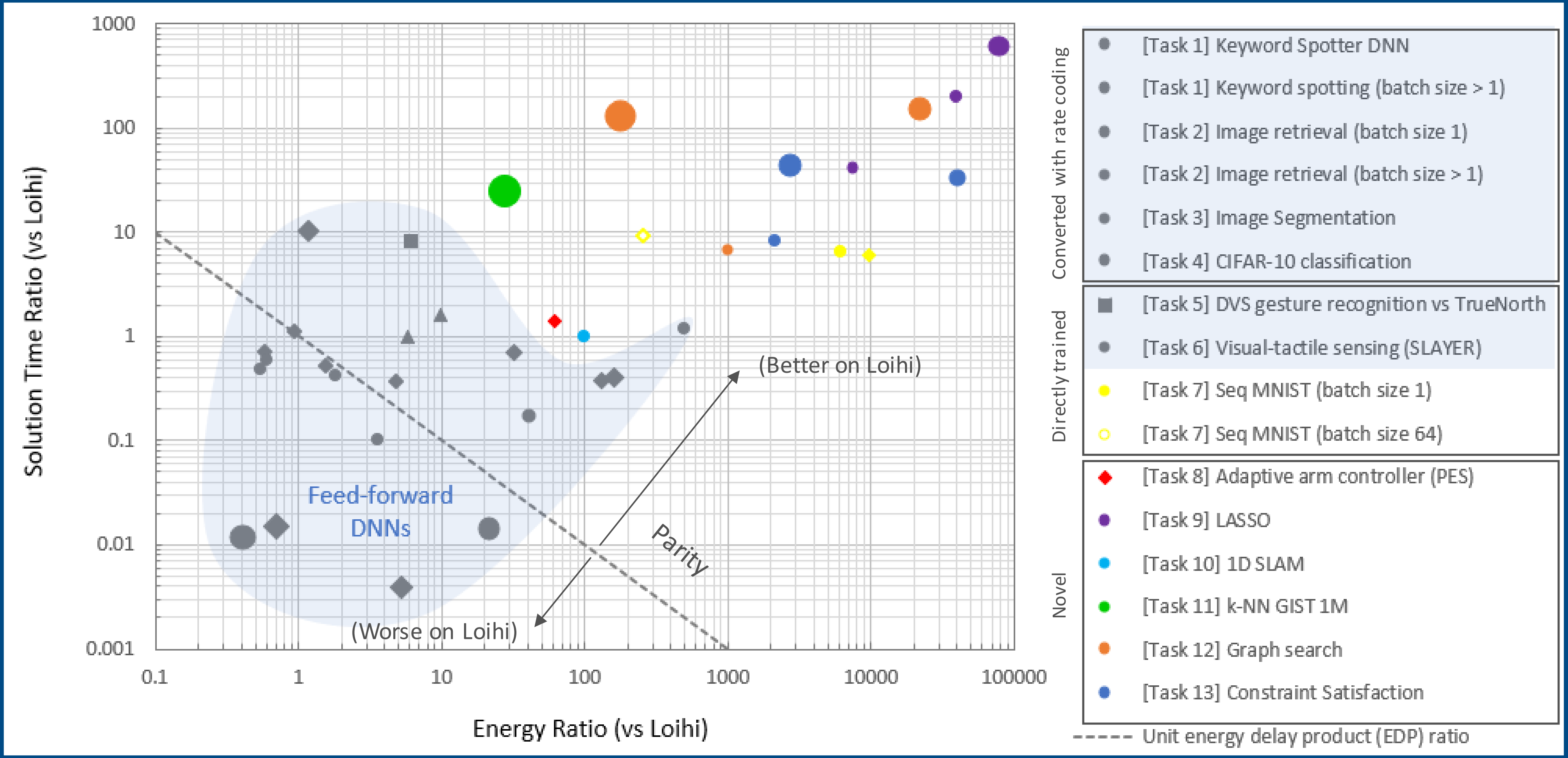


See backup for references and configuration details. Results may vary.

Standard feed-forward deep neural networks give the least compelling gains (if gains at all)

Reference architecture

- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth

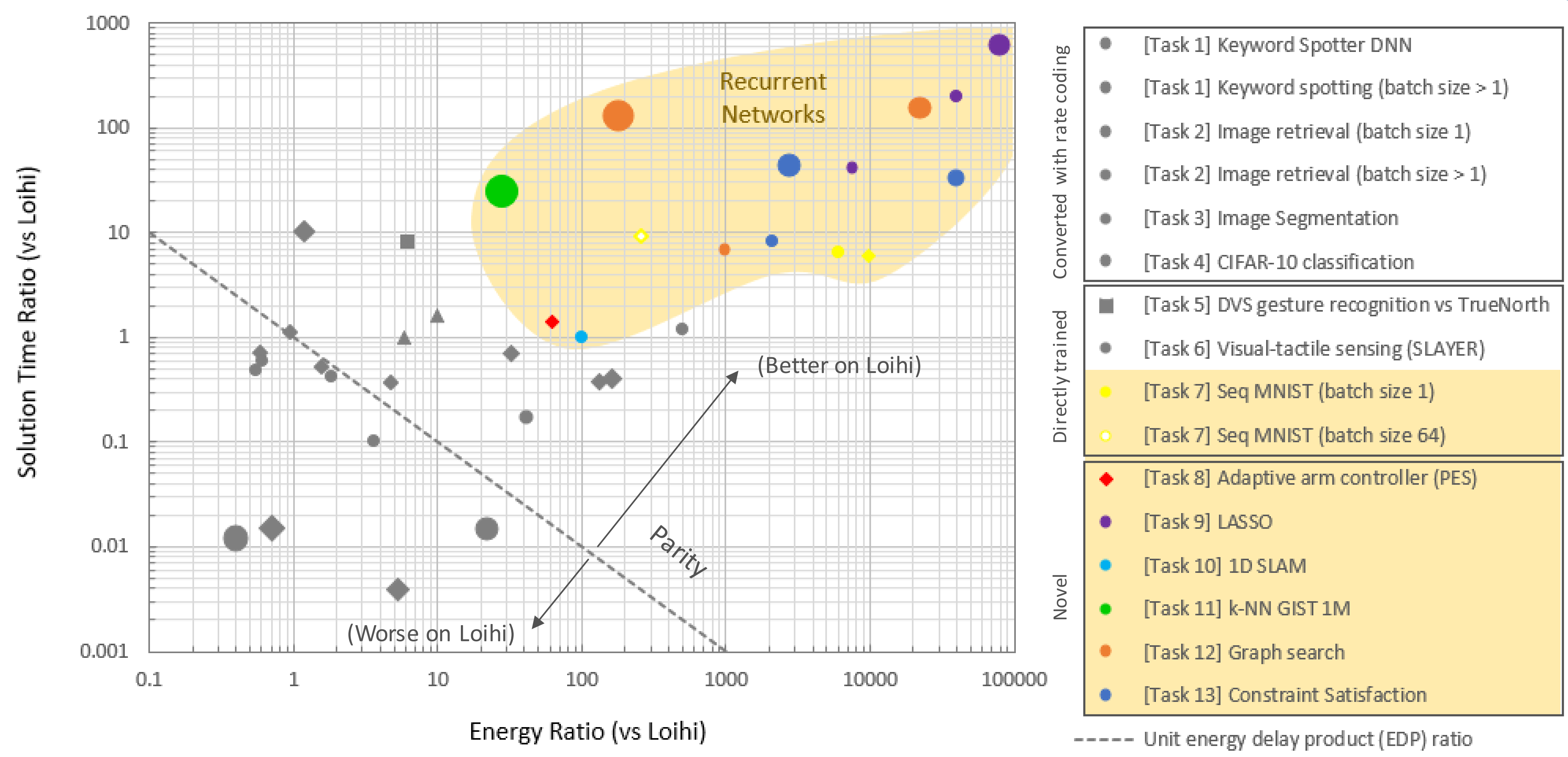


See backup for references and configuration details. Results may vary.

Recurrent networks with novel bio-inspired properties give the best gains

Reference architecture

- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth

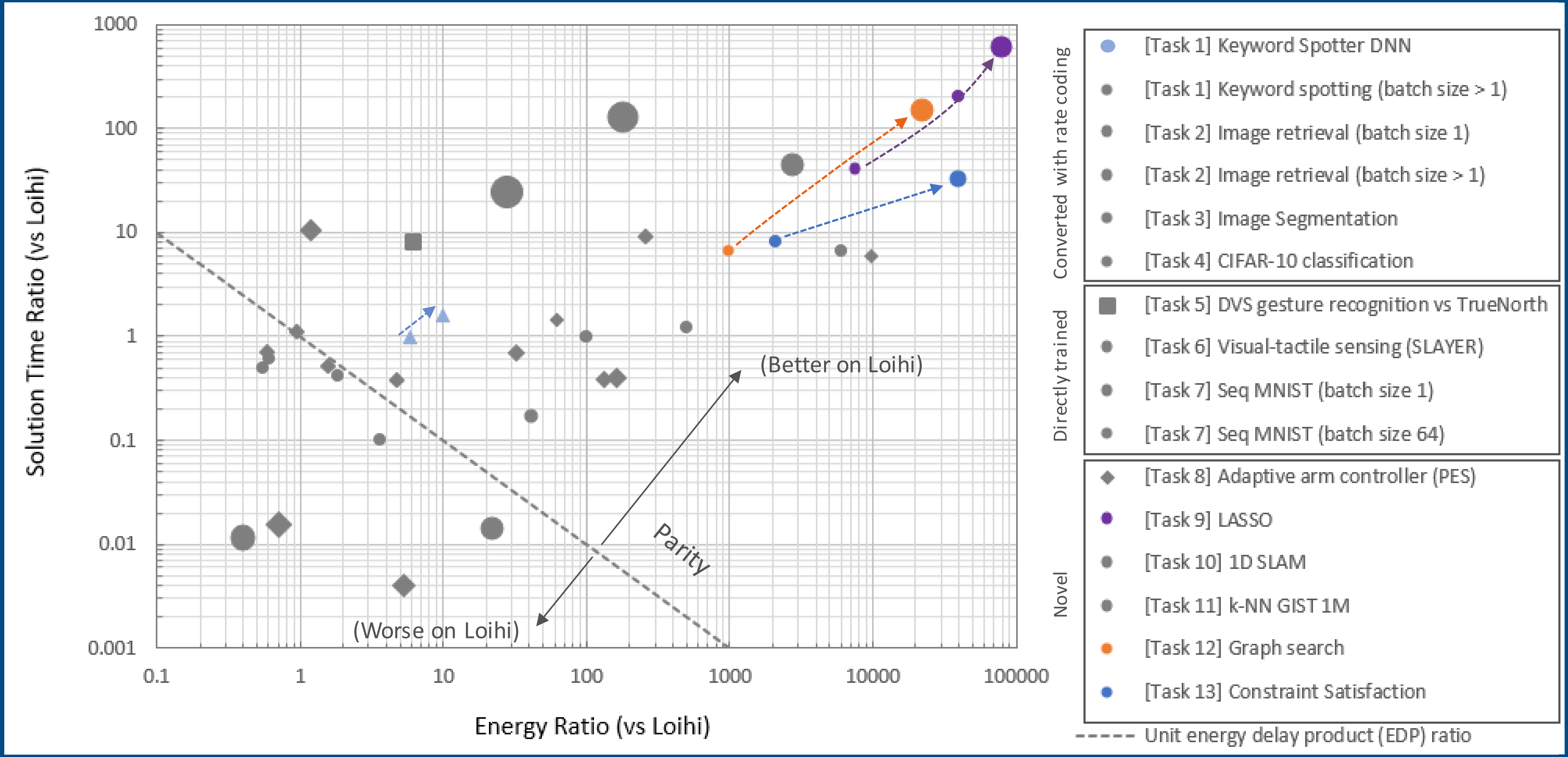


See backup for references and configuration details. Results may vary.

Compelling scaling trends:
Larger networks give greater gains

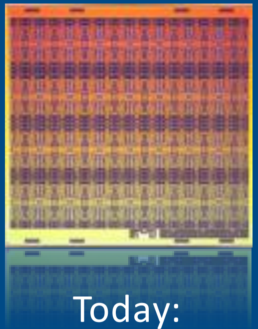
Reference architecture

- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth



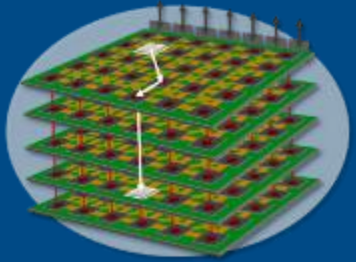
See backup for references and configuration details. Results may vary.

What this Implies for the Technology Outlook



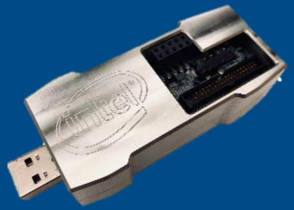
Today:
General-purpose
neuromorphic
architecture

Scaled up systems



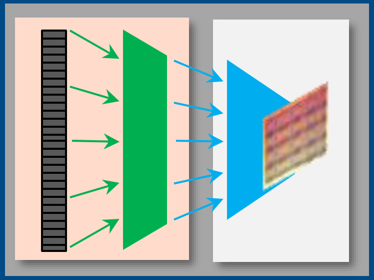
- Solve hard problems quickly
- Real-time pattern matching
- Recommendation systems
- Graph analytics
- Scientific computing, HPC

Edge Computing



- Enables novel AI algorithms
- Online adaptation + learning
- Real-time temporal data processing
- Low power + low latency

Event-Based Sensing



- Orders of magnitude lower latency and power
- Re-thinking visual sensing – electronic retina
- Tactile sensing – electronic skin
- Active sensing
- *Calls for sensor-level integration with neuromorphic processing*

Legal Information

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Results have been estimated or simulated.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

intel®

References and System Test Configuration Details

[Task 1] P Blouw et al, 2018. arXiv:1812.01739

[Task 2] TY Liu et al, 2020, arXiv:2008.01380

[Task 3] KP Patel et al, "A spiking neural network for image segmentation," *submitted, in review*, Aug 2020.

[Task 4] **Loihi**: Nahuku system running NxSDK 0.95. CIFAR-10 image recognition network trained using the SNN-Toolbox (code available at https://snntoolbox.readthedocs.io/en/latest). **CPU**: Core i7-9700K with 32GB RAM, **GPU**: Nvidia RTX 2070 with 8GB RAM. OS: Ubuntu 16.04.6 LTS, Python: 3.5.5, TensorFlow: 1.13.1. Performance results are based on testing as of July 2020 and may not reflect all publicly available security updates.

[Task 5] **Loihi**: Nahuku system running NxSDK 0.95. Gesture recognition network trained using the SLAYER tool (code available at https://github.com/bamsumit/slayerPytorch). Performance results are based on testing as of July 2020 and may not reflect all publicly available security updates. **TrueNorth:** Results and DVS Gesture dataset from A. Amir et al, "A low power, fully event-based gesture recognition system," in IEEE Conf. Comput. Vis. Pattern Recog. (CVPR), 2017.

[Task 6] T. Taunyazov et al, 2020. RSS 2020

[Task 7] Bellec et al, 2018. arXiv:1803.09574. **Loihi**: Loihi: Wolf Mountain system running NxSDK 0.85. **CPU**: Intel Core i5-7440HQ, with 16GB running Windows 10 (build 18362), Python: 3.6.7, TensorFlow: 1.14.1. **GPU**: Nvidia Telsa P100 with 16GB RAM. Performance results are based on testing as of December 2018 and may not reflect all publicly available security updates.

[Task 8] T. DeWolf et al, "Nengo and Low-Power AI Hardware for Robust, Embedded Neurorobotics," Front. in Neurorobotics, 2020.

[Task 9] Loihi Lasso solver based on PTP Tang et al, "Sparse coding by spiking neural networks: convergence theory and computational results," arXiv:1705.05475, 2017. **Loihi**: Wolf Mountain system running NxSDK 0.75. **CPU**: Intel Core i7-4790 3.6GHz w/ 32GB RAM running Ubuntu 16.04 with HyperThreading disabled, SPAMS solver for FISTA, http://spams-devel.gforge.inria.fr/.

[Task 10] G Tang et al, 2019. arXiv:1903.02504

[Task 11] EP Frady et al, 2020. arXiv:2004.12691

[Task 12] Loihi graph search algorithm based on *Ponulak F., Hopfield J.J. Rapid, parallel path planning by propagating wavefronts of spiking neural activity. Front. Comput. Neurosci. 2013.* **Loihi**: Nahuku and Pohoiki Springs systems running NxSDK 0.97. **CPU**: Intel Xeon Gold with 384GB RAM, running SLES11, evaluated with Python 3.6.3, NetworkX library augmented with an optimized graph search implementation based on Dial's algorithm. See also http://rpg.ifi.uzh.ch/docs/CVPR19workshop/CVPRW19_Mike_Davies.pdf

[Task 13] **Loihi**: constraint solver algorithm based on *G.A. Fonseca Guerra and S.B. Furber, Using Stochastic Spiking Neural Networks on SpiNNaker to Solve Constraint Satisfaction Problems. Front. Neurosci. 2017*. Tested on the Nahuku 32-chip system running NxSDK 0.98. **CPU**: Core i7-9700K with 32GB RAM running Coin-or Branch and Cut (https://github.com/coin-or/Cbc). Performance results are based on testing as of July 2020 and may not reflect all publicly available security updates.