Lessons from Loihi for the Future of Neuromorphic Computing

Mike Davies, Director of Intel's Neuromorphic Computing Lab
March 16, 2021

Neuro-Inspired Computing Elements (NICE) 2021

Consider autonomous drone racing

2018 IROS Drone Racing Competition

Brains remain unrivaled computing devices

COCKATIEL PARROT

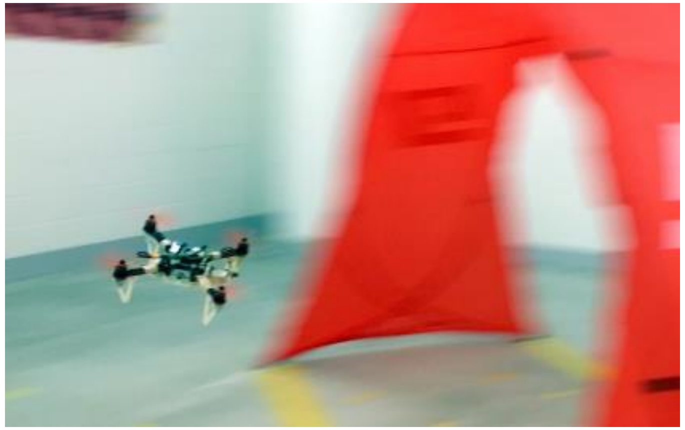
Brain
Power: 50 mW
Mass: 2.2 grams

Navigates and learns unknown environments at 35 km/h

Can learn to speak English words

Can learn to manipulate cups for drinking

AUTONOMOUS DRONE



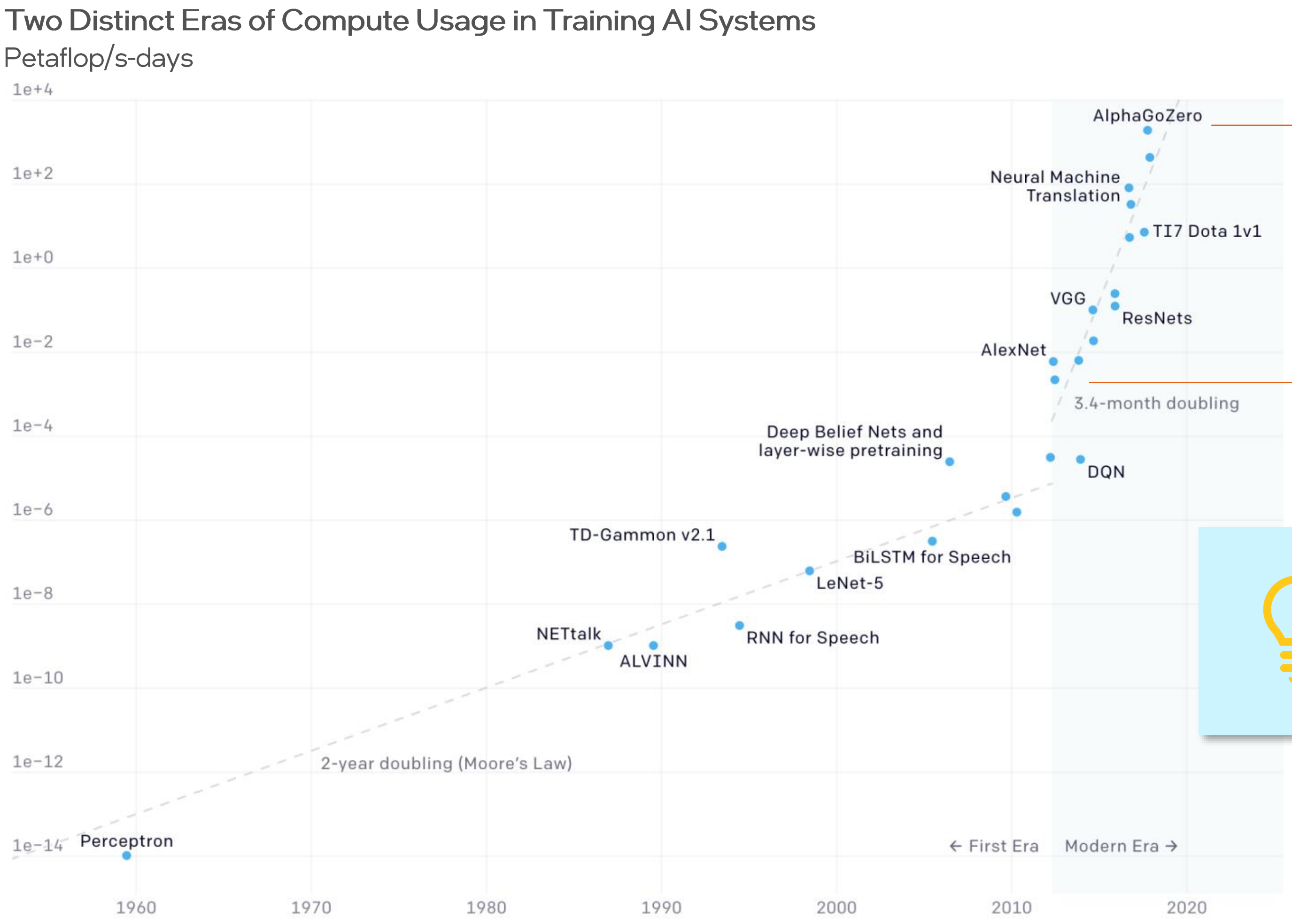
CPU/GPU controller
Power: 18,000 mW
Mass: ~40 grams

Pre-trained to fly between known gates at walking pace

Can't learn anything online

Sources: PNAS, June 13, 2016; https://link.springer.com/article/10.1007/s00360-011-0603-1; Davide Scaramuzza, ETH Zurich and A. Loquercio et al, "Deep Drone Racing: From Simulation to Reality with Domain Randomization," IEEE Trans. Robotics, 2020.

Deep learning models are increasingly power hungry



Source: OpenAI https://openai.com/blog/ai-and-compute/

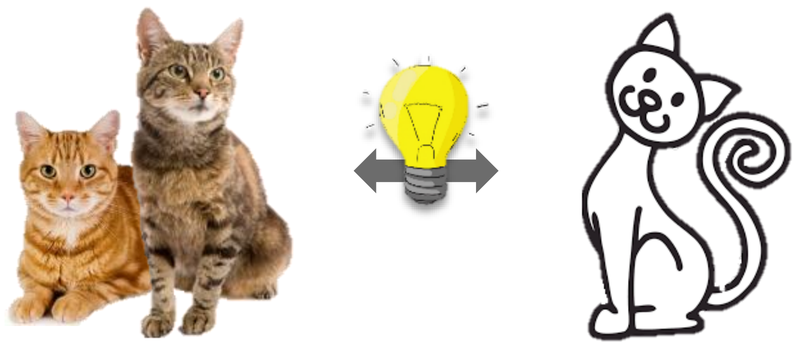
300,000x increase in required training computation over 6 years
… versus **8x** provided by Moore's Law

Not on a trajectory to close the efficiency gap with nature!

Deep learning is fundamentally limited in other respects

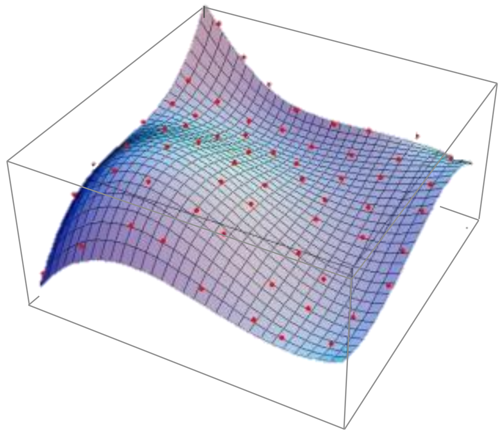
Natural Learning

- Fast generalization with few examples
- Online and incremental
- Automatic abstraction



Deep Learning

- Slow generalization with massive data
- Offline and batched
- “Curve fitting”

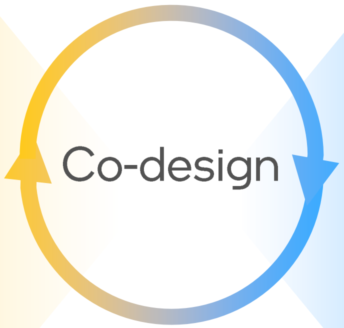


Our Approach: Look to the brain, co-design the architecture and algorithms

Neuro-Inspired Silicon

- **Pursue neuro inspiration?** — No → *Conventional computing and traditional AI approaches*
  - Yes ↓
- **Compute-memory integrated?** — No → *Traditional accelerator-based architectures* GPUs, TPU, Movidius
  - Yes ↓
- **Temporal neuron models?** — No → *Traditional neural network algorithms and other connectionist approaches* Cerebras, Berkeley (Rabaey)
  - Yes (SNNs + derivatives) ↓
- **Standard CMOS or new devices?** — New → *Experimental small-scale designs* RRAM crossbar chips, IBM (A. Sebastian's PCM-based spiking neurons), Rain Neuromorphics
  - CMOS ↓
- **Asynchronous design style?** — No → *Fully standard design methodologies* Tsinghua U (Tianjic), Zheijiang Labs (Darwin), Human Brain Project (SpiNNaker 2), BrainChip, GrAI Matter Labs, IMEC
  - Yes ↓
- **Integrate analog circuits?** — Yes → *"Traditional" neuromorphic engineering* Examples: Stanford (BrainDrop), SynSense/ETHz (DynapSE), Human Brain Project (BrainscaleS)
  - No ↓
- **Support plasticity? (+ other novel features)** — No → IBM (TrueNorth)
  - Yes ↓

Intel Labs (Loihi)

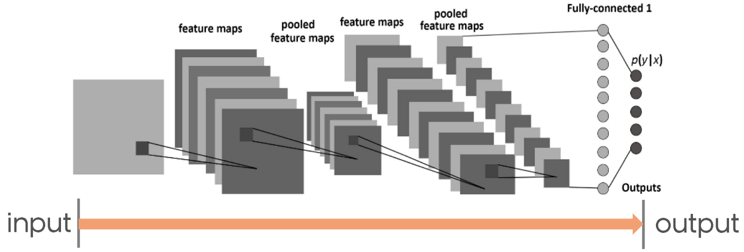


Novel Neuro-Inspired Algorithms

Category	Example applications
Deep learning: backprop-trained event-based DNNs	Object and gesture recognition for event-based vision sensors, slip detection for event-based tactile sensors, ANNs with sparsely changing input data
Deep learning: DNNs with online adaptation	Few-shot new gesture learning, Adaptive control,
Vector Symbolic Architectures (VSA), *aka* **Hyperdimensional Computing (HDC)**	Semantic factorization, relational reasoning, symbolic and analogical reasoning
Neural Engineering Framework (NEF)	Adaptive control systems, state machines
Dynamic Neural Fields (DNF)	SLAM, object tracking, dynamic control, attention
Neural sampling *e.g.* spiking Boltzmann machines	Constraint satisfaction, probabilistic inference
Oscillatory computation	Optimization, event-based spectral transforms, optic flow, audio spectral normalization
Recurrent Excitation/Inhibition-balanced networks	LASSO regression, sparse feature coding
Event-based networks with temporally coded information	Graph search, similarity search

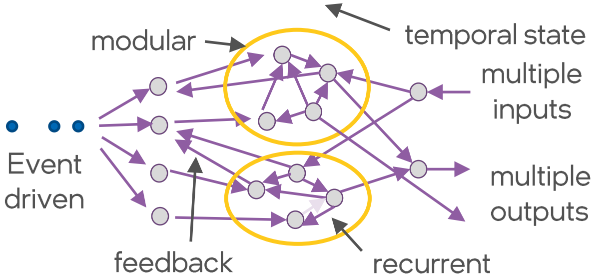
Conventional Deep Networks

$$u_i = \sum_j w_{ij} f\left(u_j\right) + b_i$$



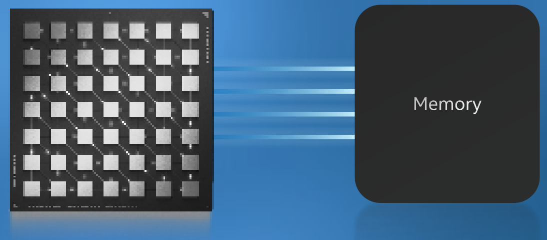
Neuromorphic Networks

$$u_i(t) = \sum_j w_{ij} \left(\delta_j(t) * \alpha_u(t)\right) + b_i$$
$$\tau \dot{v}_i(t) = \left(-v_i(t) + u_i(t)\right) - V_{thr}\delta_i(t)$$

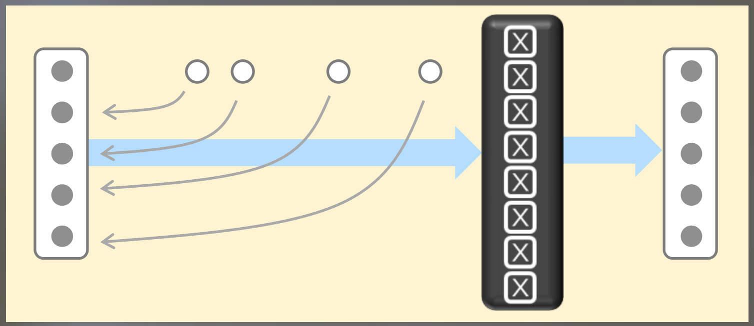


Motivates a fundamentally different kind of computing

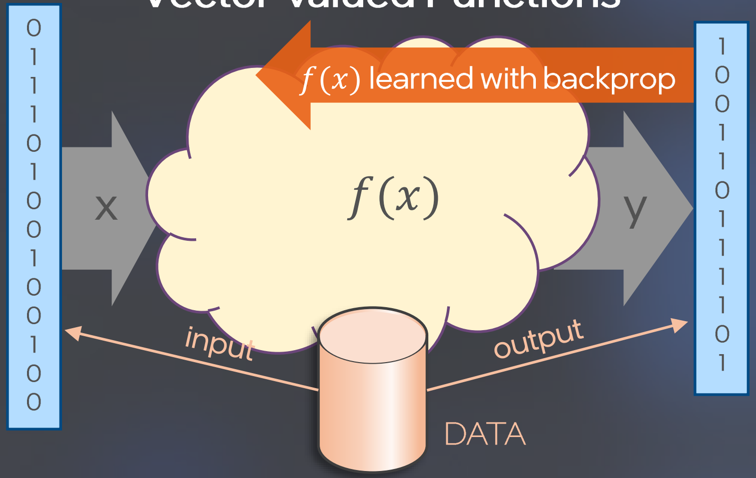
Parallel Computing



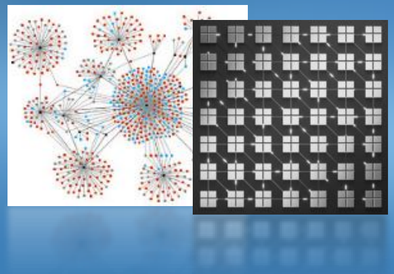
Batched + Vectorized Processing



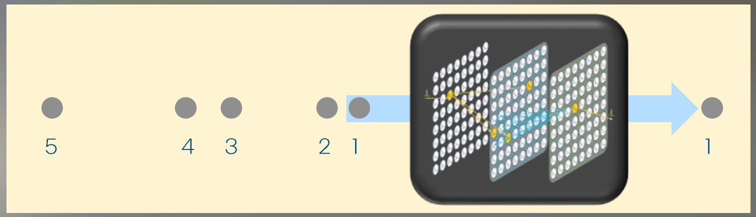
Vector-valued Functions



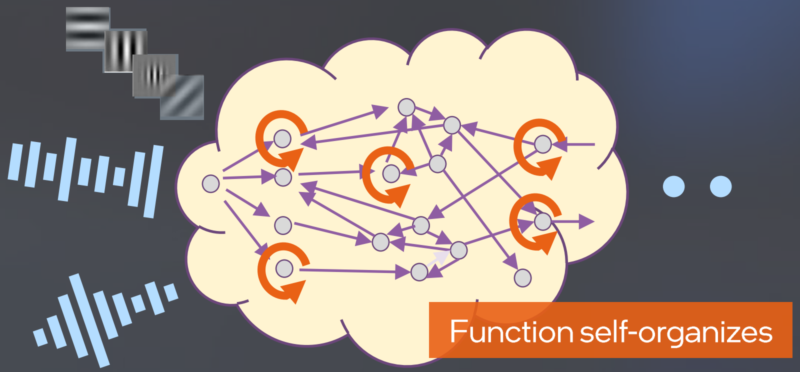
Neuromorphic Computing



Unbatched + Sparse Processing



Event-Driven Dynamical Systems



Our Loihi chip

KEY PROPERTIES

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 scalability
On-chip learning without weight movement or data storage
Digital asynchronous implementation for power efficiency, scalability, and fast prototyping

Yet...

No floating-point numbers No multiply-accumulators No off-chip DRAM

Fundamental to
deep learning hardware

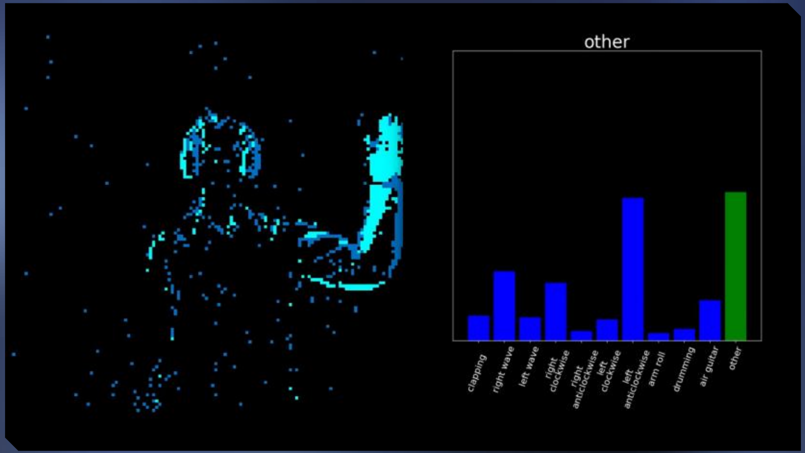
Intel Neuromorphic Research Community

Collaborating to Accelerate the Research

INRC includes over 120 groups

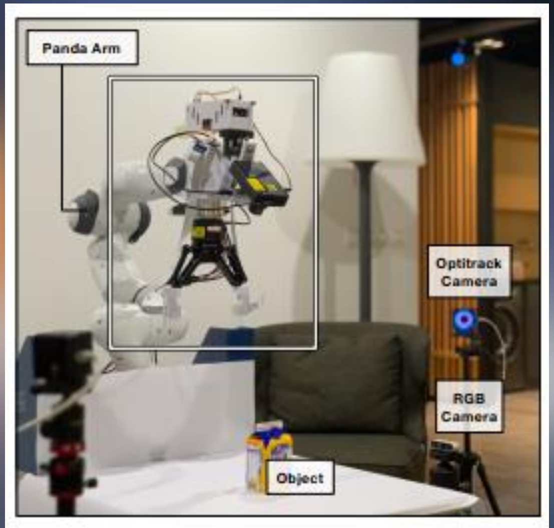
Loihi Results

Efficient Sensing



Gesture recognition + learning

Loihi + DAVIS 240C camera
60 mW total power, 15 mW dynamic [Task 5]
G. Orchard and SB Shrestha,
with K. Stewart, E. Neftci (UCI)

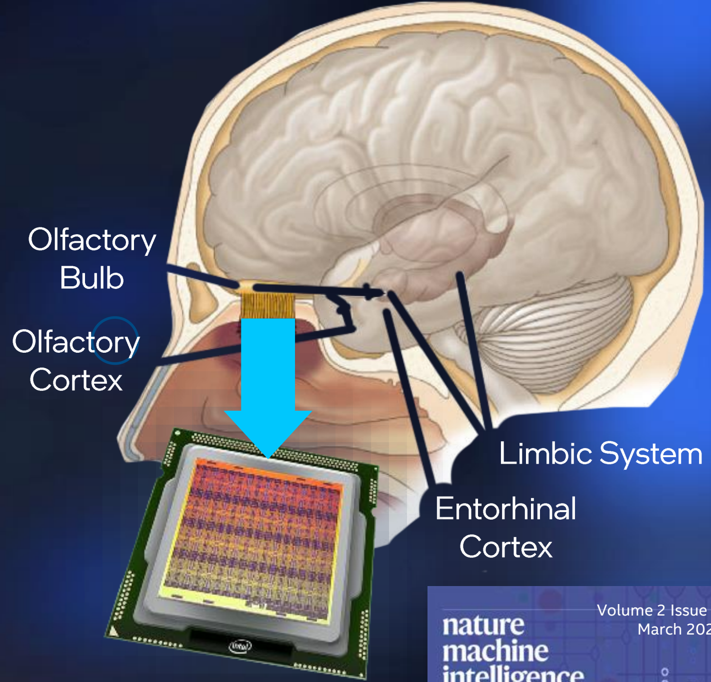


Visual-Tactile Sensing

45x lower power
20% faster vs GPU [Task 6]
T. Taunyazov et al (NUS)

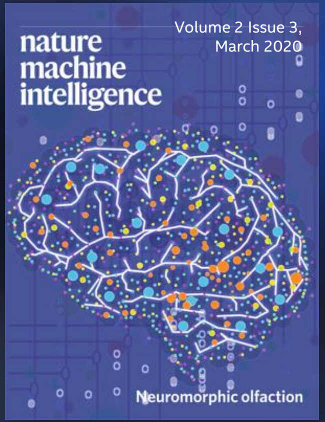
Audio keyword spotting

>100x lower energy per inference vs GPU [Task 1]
P. Blouw et al (ABR)



Olfaction-inspired odor recognition and learning

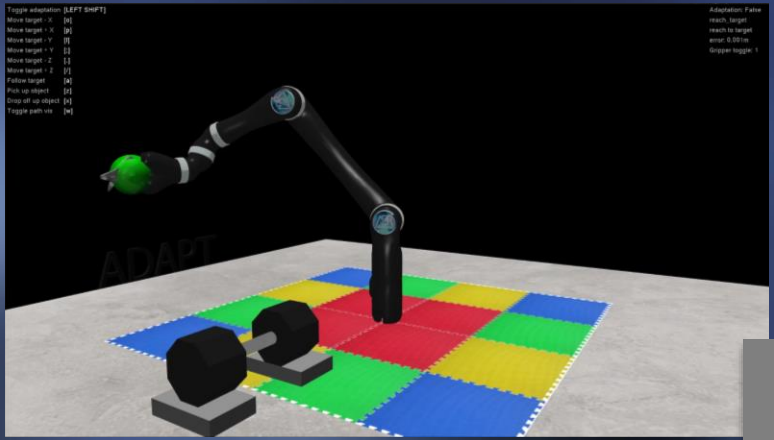
3000x more data efficient learning than a deep autoencoder



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

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

Compelling results for robotic and drone workloads



Adaptive robotic arm control

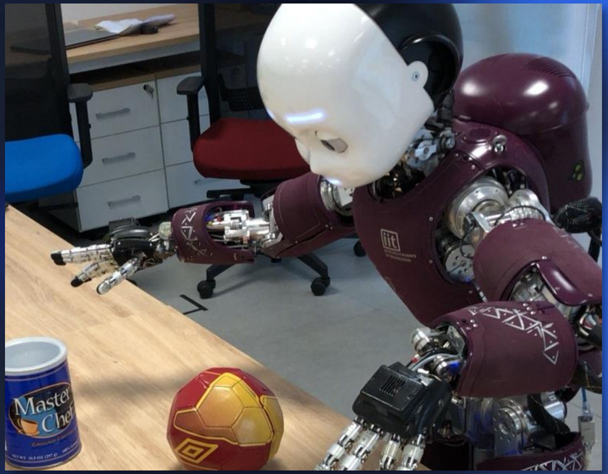
40x lower power, 50% faster vs GPU [Task 8]

Applied Brain Research

iCub scene understanding

Integrated behaviors: Object recognition, tracking, learning

with A. Glover, C. Bartolozzi (IIT)



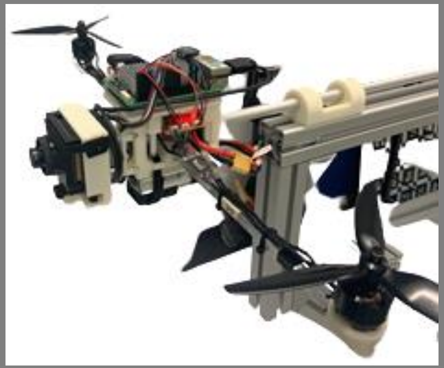
Head Direction Localization and Learning

- 100x lower power vs CPU [Task 10]

G. Tang, K. Michmizos (Rutgers)
Y. Sandamirskaya et al (Intel/ETHz/INI)

Event-based UAV horizon tracking

DVS Hough transform
+
Adaptive PID controller
2ms latency,
22x faster vs CPU
[Task 14]
Intel/ETHz



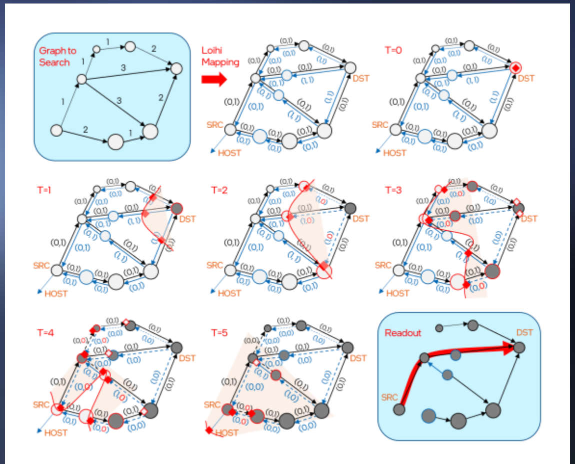
Micro Aerial Vehicle Landing

Evolutionary design of a 35-neuron network that achieves smooth MAV landings with Loihi on board

J. Dupeyroux et al, arXiv:2011.00534v1 (TU Delft)

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

Even greater gains for sparse computational studies



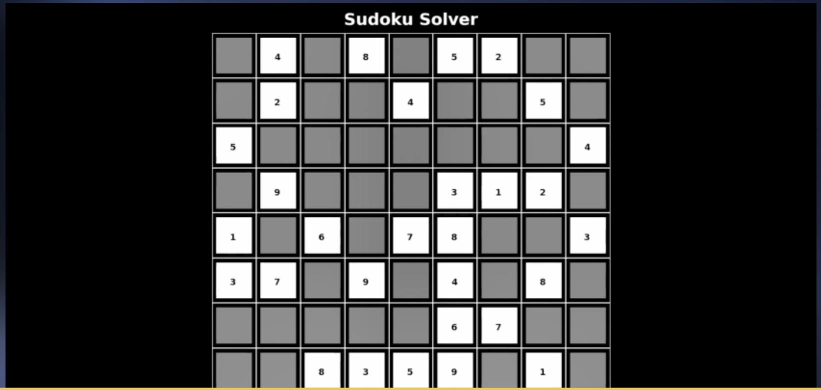
Graph Search

With temporally coded spike wavefronts

100x faster vs CPU [Task 12]

Combinatorial optimization (CSP, SAT, sudoku, train scheduling)

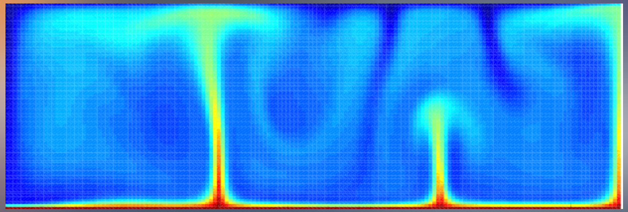
2,000x lower energy and 40x faster vs CPU [Task 13]



Hear more at the Loihi tutorial!

Similarity Search

24x faster and 30x lower energy (vs CPU) [Task 11]



Source: Wikipedia, H. Schmeling, Uni Frankfurt

Heat diffusion modeling

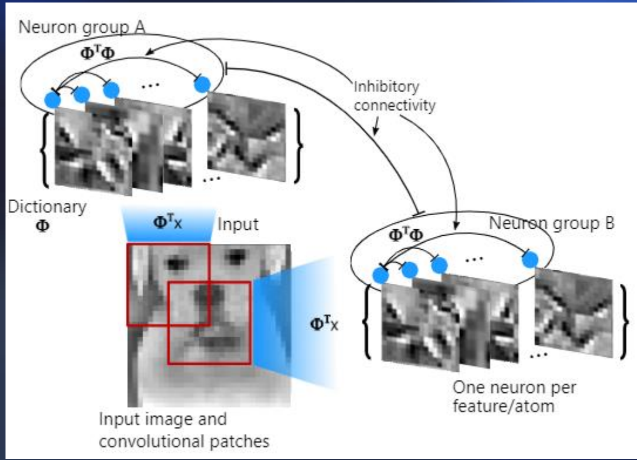
Scaled to 100+ chips and 300k mesh points

B. Aimone et al (Sandia)

LASSO / sparse reconstruction

(Locally Competitive Algorithm)

10^3x faster, 10^4x lower energy vs CPU [Task 9]

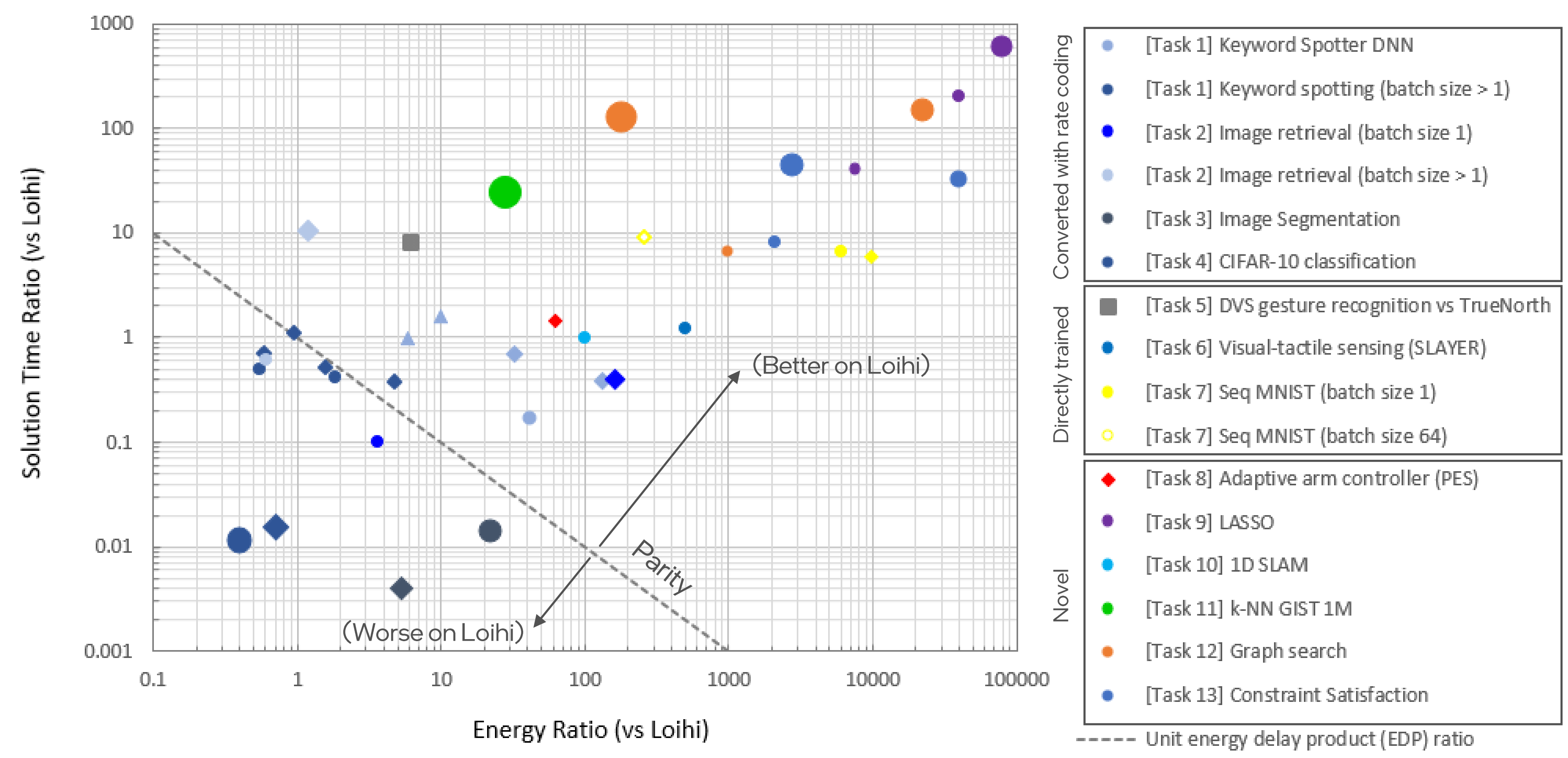


See backup for references and configuration details. Results may vary.

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

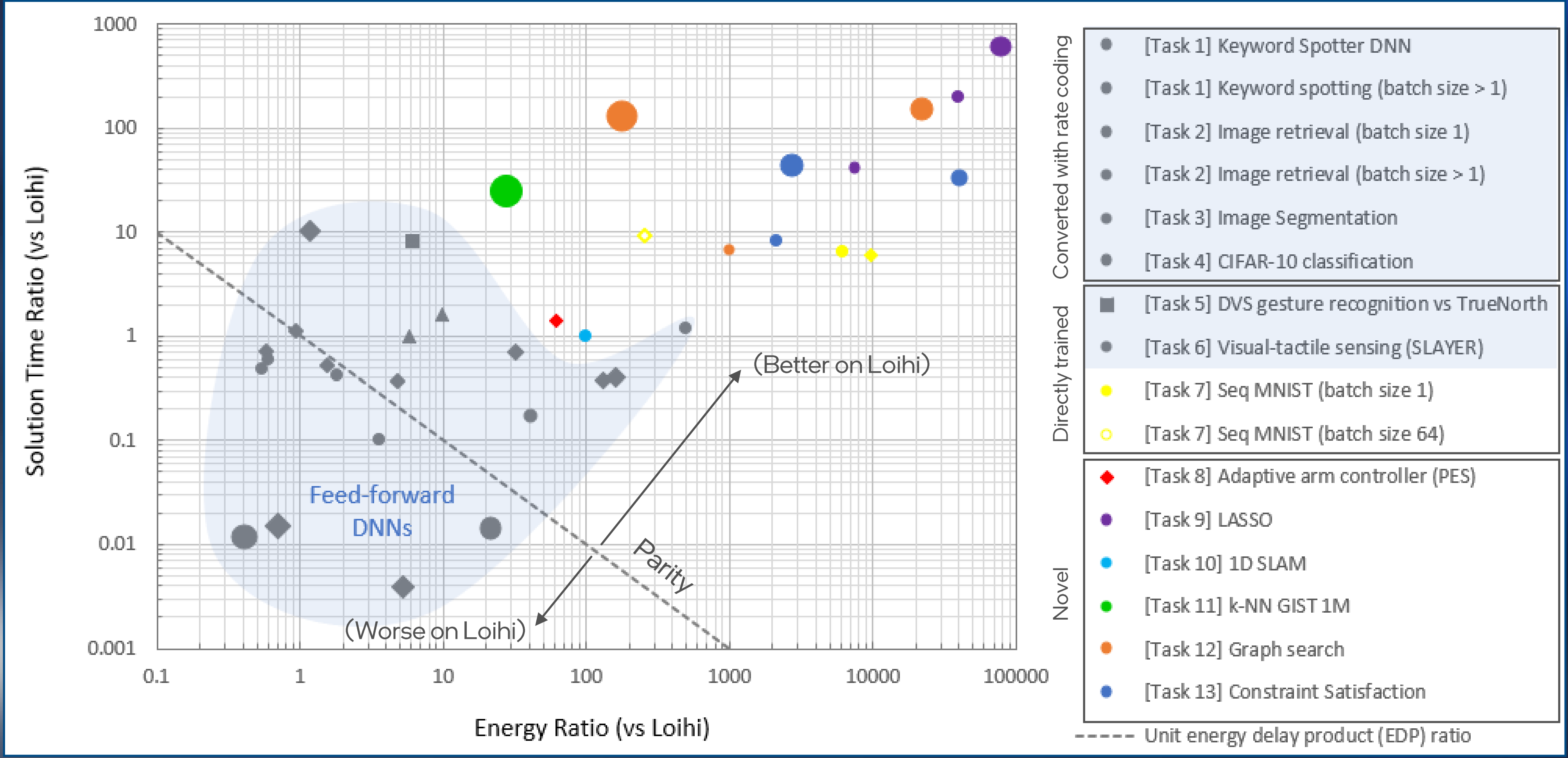


See backup for references and configuration details. Results may vary.

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

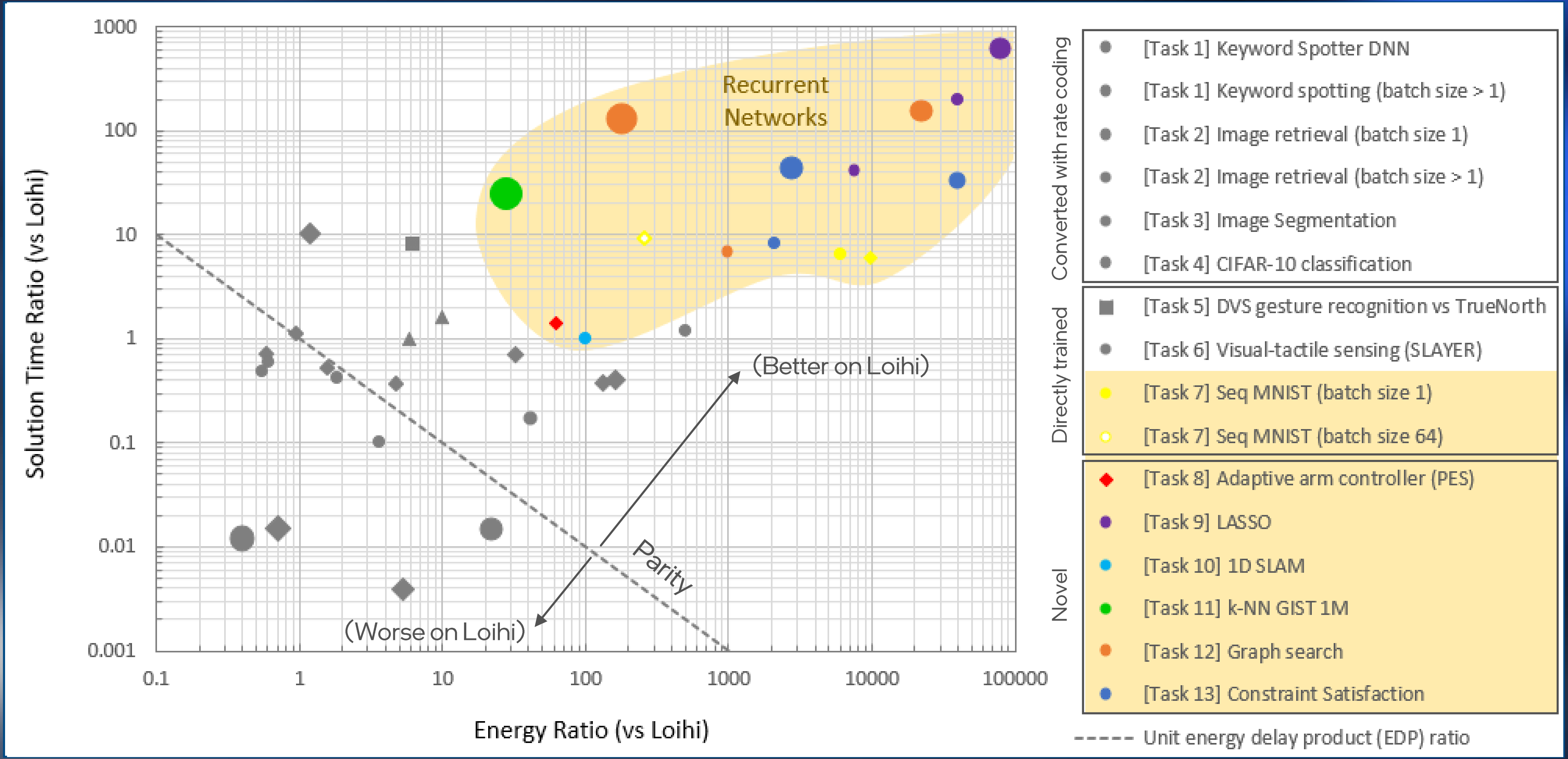


See backup for references and configuration details. Results may vary.

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

Reference architecture

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

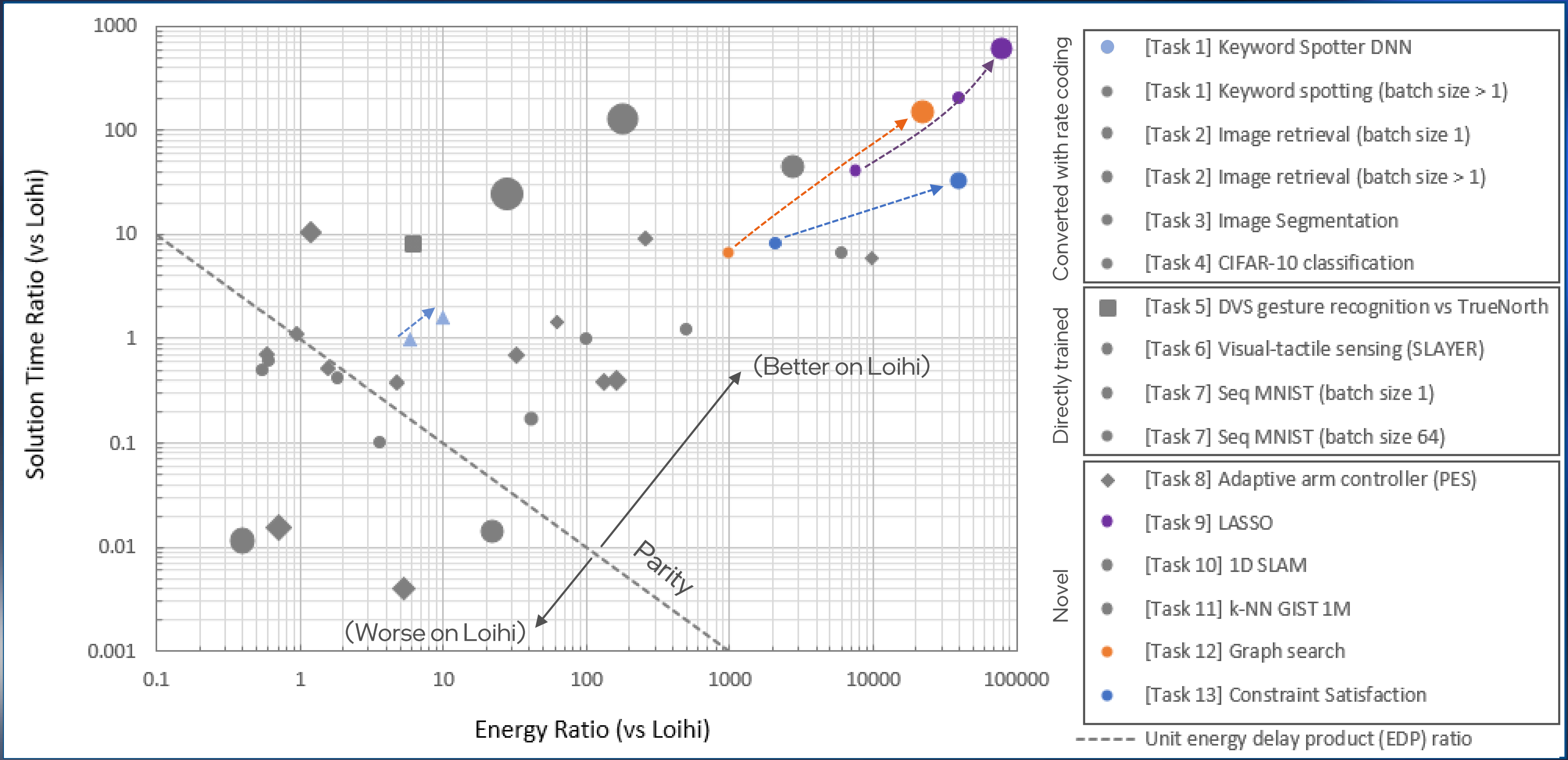


See backup for references and configuration details. Results may vary.

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.

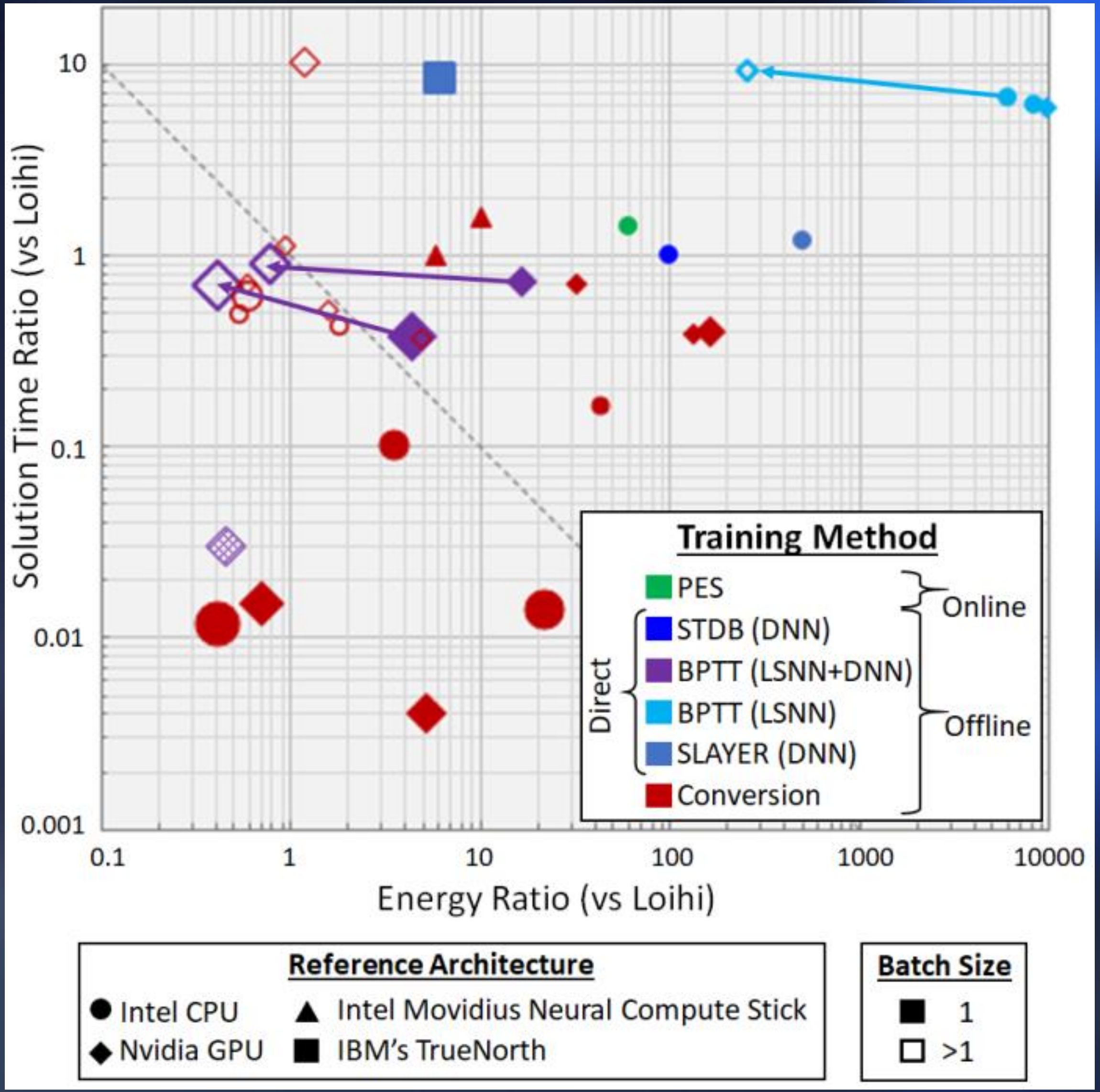
Deep Learning on Loihi

Red: ANNs converted with rate coding
→ Low energy but high latency
→ Poor scaling
→ Not very promising

Blue + purple: Offline backprop-trained spike timing
→ Low energy and low latency
→ Compute intensive to train (and scale)

Hear more about SLAYER in the Loihi tutorial

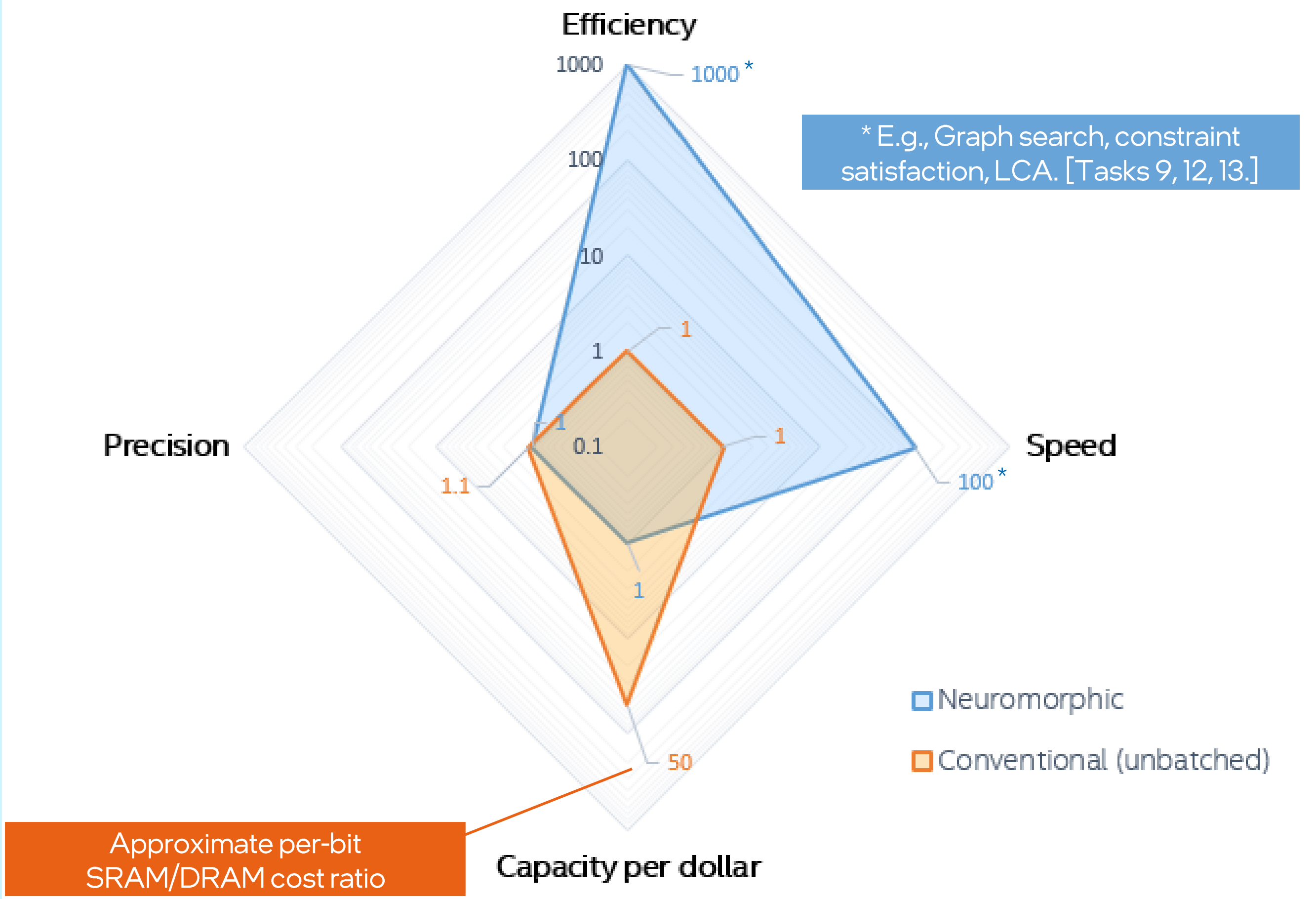
Green: Online backprop
→ Well suited for continuous adaptation



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

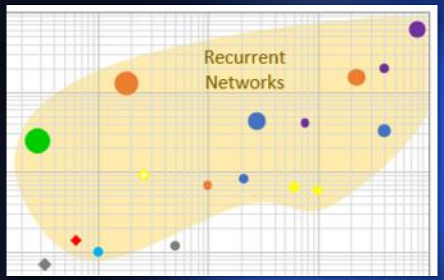
Loihi shows order of magnitude gains are possible

- 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
- Long term, we will need to **reduce cost with process technology innovations**



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

Computing with Collective Dynamics



	Gradient Descent	Non-Gradient Based Approaches
Plastic Weights	Backprop (offline) Online Backprop approximations	Olfaction-inspired learning Associative learning (e.g. SLAM) Graph Search
Static Weights	Locally Competitive Algorithm Winner Take All Dynamic Neural Fields	Combinatorial optimization Nearest Neighbor Search

Neuromorphic Learning Perspectives

Gradient-Based Learning

- DNN scaling possible(?), not yet proven
- Data hungry – slow to learn
- Data samples need to be uniformly distributed during learning
- Learning activity is not sparse

Limited today to shallow networks that run relatively slowly

Examples: feedback alignment, e-prop, delta

Good for fine-tuning and adapting

Non-Gradient Based Learning

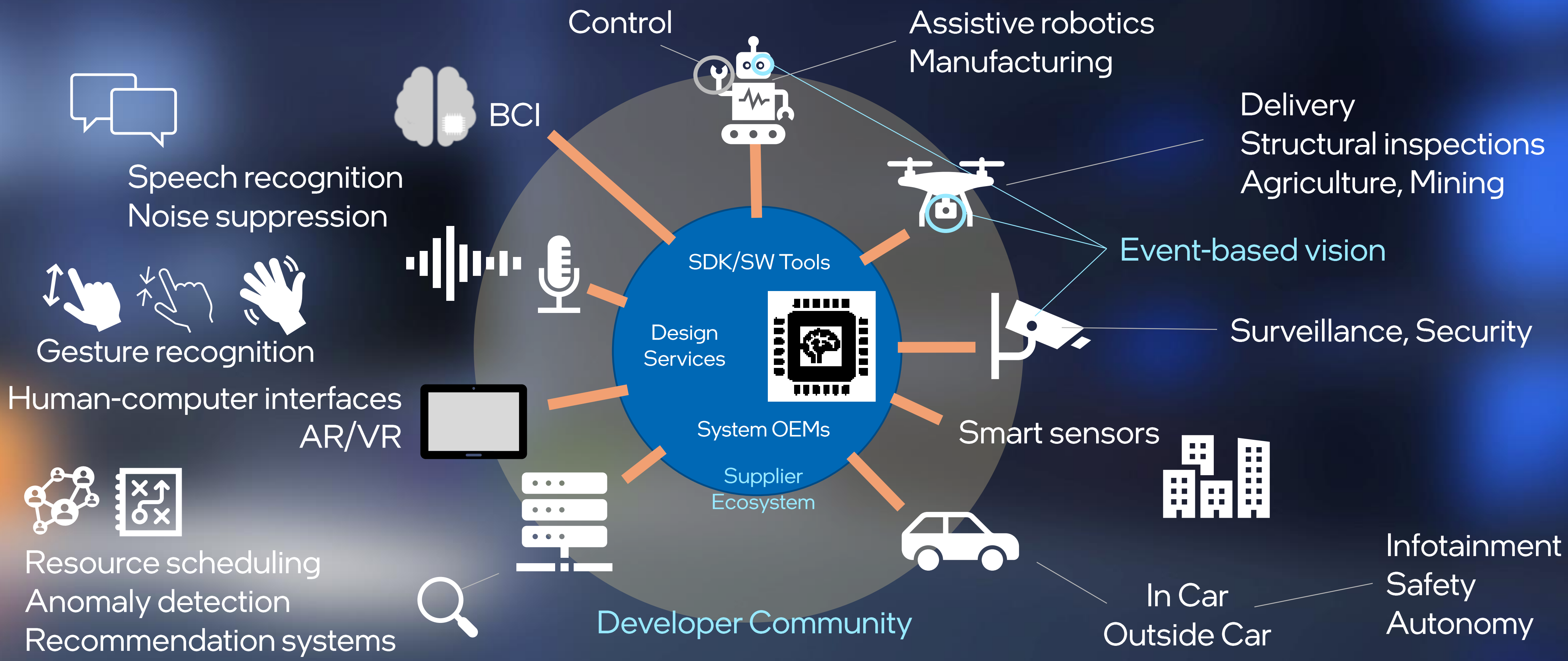
- No “deep” examples to date
- Fast to learn from few examples
- Networks mostly need to be hand engineered and tuned
- Learning activity is sparse

Limited today to interesting examples, but with narrow scope

Example: olfactory model

Good for associative learning

Outlook to Commercialization



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.

Thank You!

Learn more at the Loihi tutorial tomorrow

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.