INRC Spring 2022 Workshop

New Tools for a New Era of Neuromorphic Computing

Mike Davies
Director, Neuromorphic Computing Lab

April 19, 2022

Intel Neuromorphic Research Community Spring 2022 Workshop

Legal Information

Welcome!
To Our Second Fully Virtual INRC Workshop

Fall Workshop in Graz, Oct 2019	Winter Workshop Feb 2021	Spring Workshop April 2022
Face-to-face	Webex	Teams
100 Attendees	350+	700+
Barely any recorded content	Nearly all content recorded	Nearly all content recorded
Most sessions closed to formally engaged members	Most sessions open to the broader community	Nearly all sessions open to the broader community
PDF agenda	Event website	Event website
Hallway chats, lunches, dinners	Slack	Slack
Acoustics problems	Connection problems	We will see…

(Next one hybrid – fingers crossed)

Workshop Goals

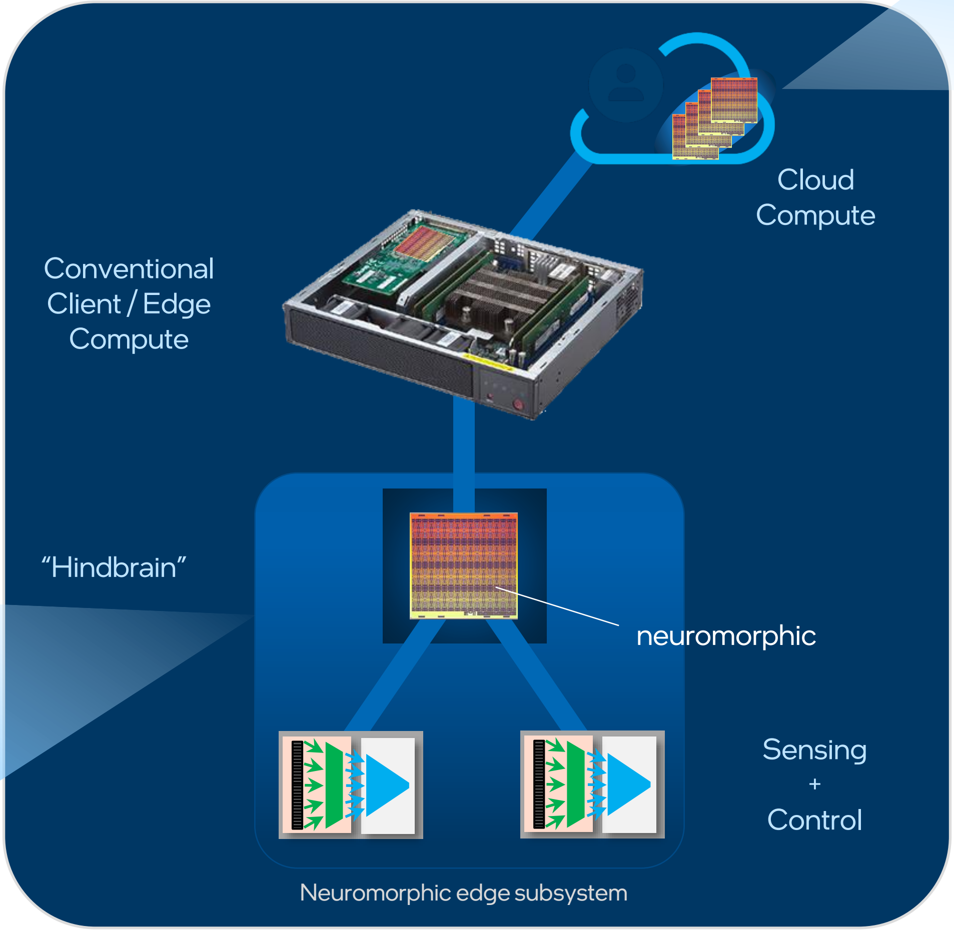
<table>
<tr><th>Long-time INRC Members</th><th>New Neuro Researchers</th><th>New Industry Participants</th></tr>
<tr><td>~40%</td><td>~40%</td><td>~20%</td></tr>
<tr><td colspan="3">Learn about our new tools (Lava, algorithmic libraries, and Loihi 2) and how to start developing</td></tr>
<tr><td colspan="3">Learn and share new results, ideas, and developments with Loihi and the broader community</td></tr>
<tr><td colspan="3">Make new connections for collaboration</td></tr>
<tr><td></td><td colspan="2">Get up to speed on background (“first era” of Loihi research)</td></tr>
<tr><td></td><td></td><td>Share needs and identify compelling business opportunities</td></tr>
</table>

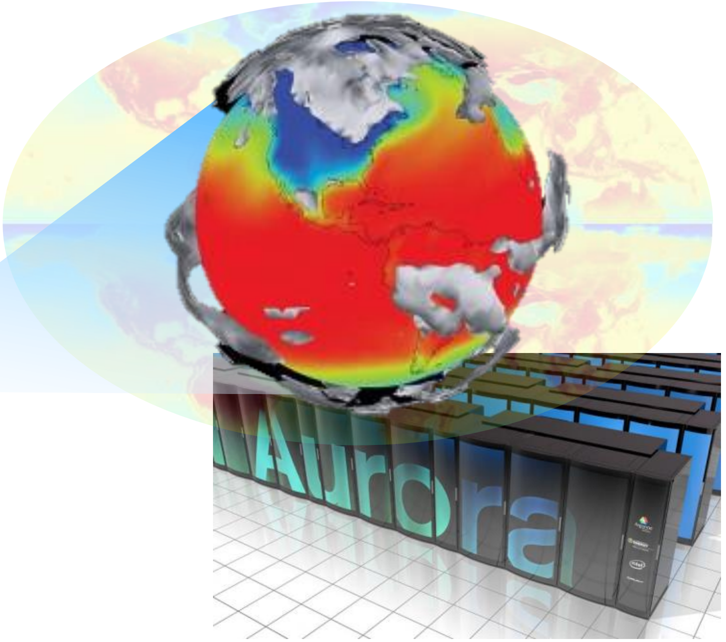
Time (PDT / CET)	Session	Speakers
8:00 / 17:00	**Welcome: Workshop kick-off**	
8:05 / 17:05	**New Tools for a New Era of Neuromorphic Computing**	Mike Davies
8:40 / 17:40	Loihi 2	
8:55 / 17:55	Lava	Mike Davies, Andreas Wild
9:40 / 18:40	**Workshop and Community Orientation**	Tim Shea + WG leads
10:00 / 19:00	Q&A / Break	
10:30 / 19:30	**Featured Community Results**	
10:35 / 19:35	▪ Neuromorphic Tunneling? Comparing Loihi with Quantum Annealing	Garrett Kenyon, LANL
10:52 / 19:52	▪ Monte Carlo Simulations on Loihi	Brad Aimone, Sandia
11:09 / 20:09	▪ Loihi in Orbit: the First 90 Days	Michael Lowry, NASA
11:26 / 20:26	▪ The Backpropagation Algorithm Implemented on Loihi	Alpha Renner, LANL
11:43 / 20:43	▪ VSA With Phasor Neurons on Loihi - Towards Neuromorphic Visual Odometry	Alpha Renner, ETHz/INI
17:00 / 02:00	**Loihi 2 Deep Dive**	Garrick Orchard

Our Goal

Develop a new programmable computing technology inspired by the modern understanding of brain computation

Integrate neuromorphic intelligence into computing products at all scales





Achieve brain-like efficiency, speed, adaptability, and intelligence

What is Neuromorphic Computing?
A Huge Bottom-up Exploration Space

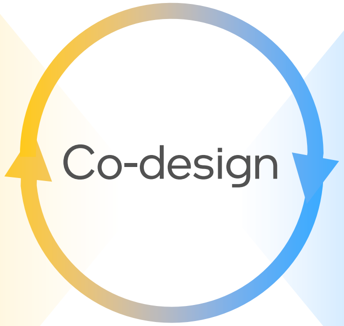
- Self-organized growth
- Autonomous healing
- Exploiting material time constants
- Oscillatory dynamics
- Stochasticity
- Local learning rules
- Very high fanout
- Distributed data representations
- Fine-grain parallelism
- Temporal data coding
- Sparse temporal activity ("Spikes")
- Sparse connectivity
- 3D wiring
- Recurrence and feedback loops
- Compute-memory integration
- Analog-valued persistent state
- Online causal adaptation
- Low precision
- Dynamics on diverse time scales
- Hybrid analog/digital computation
- Continuous time operation
- Parametric Heterogeneity

Increasingly exotic or uncommon properties in conventional computing systems

Our Approach:
Iterative architecture-algorithms co-design

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)



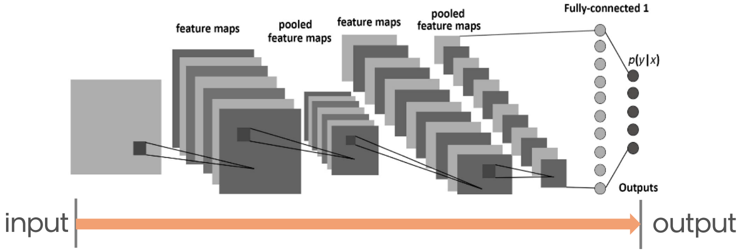
Rigorous Benchmarking

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

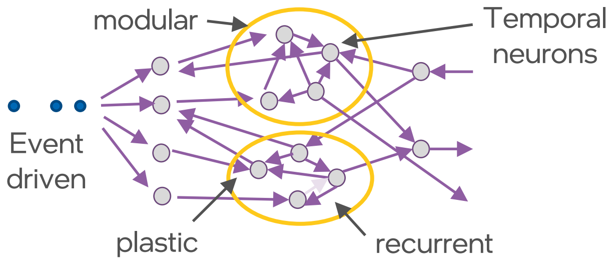
Conventional Deep Networks

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



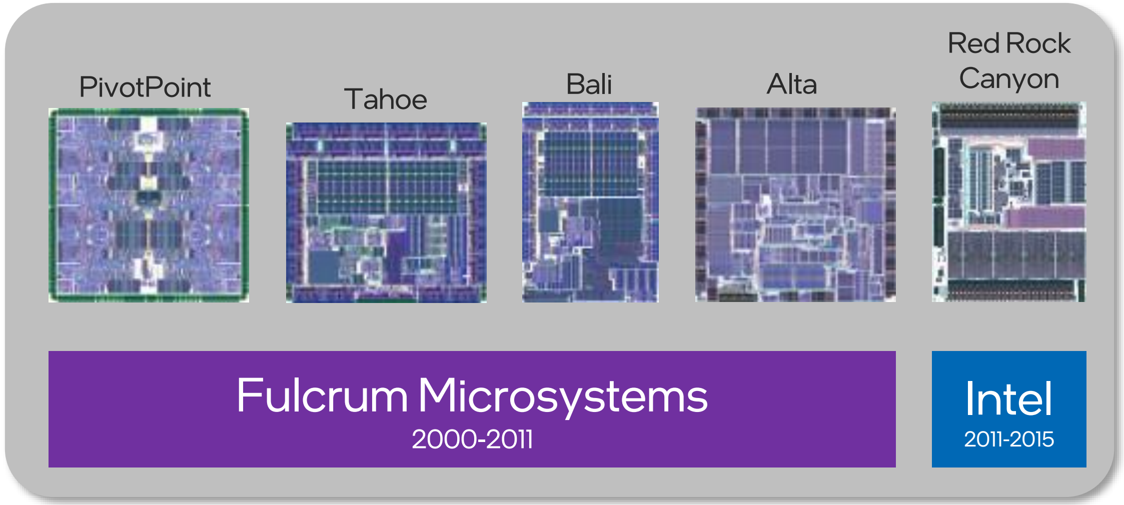
Neuromorphic Networks

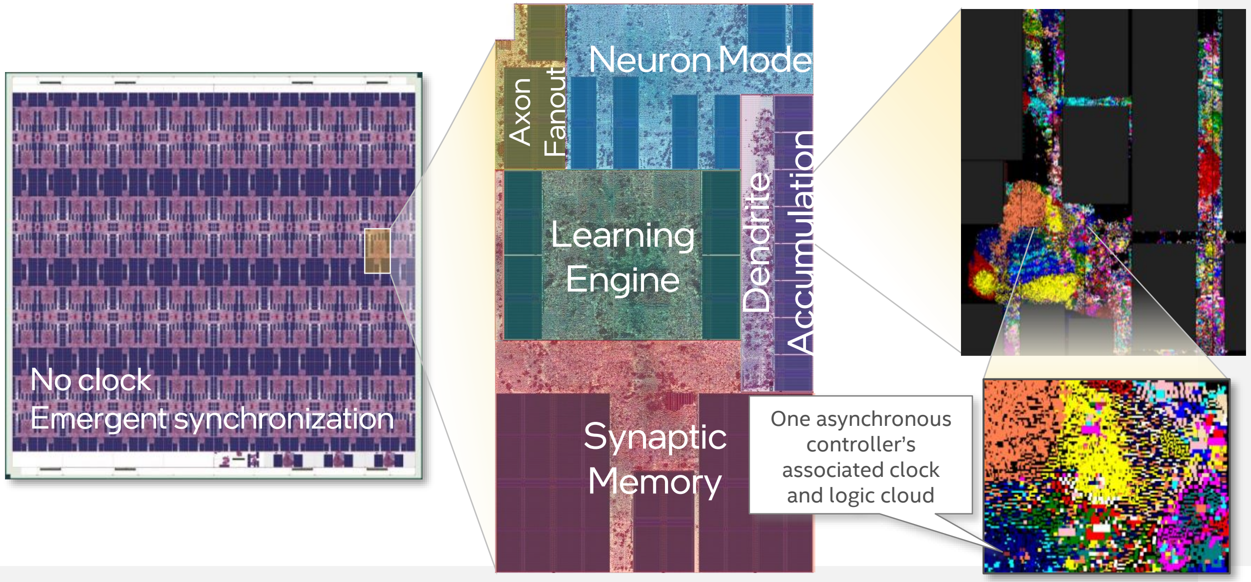
$$u_i(t) = \sum_j w_{ij}\left(\delta_j(t) * \alpha_u(t)\right) + b_i$$
$$\tau \dot{v}_i(t) = (-v_i(t) + u_i(t)) - V_{thr}\delta_i(t)$$



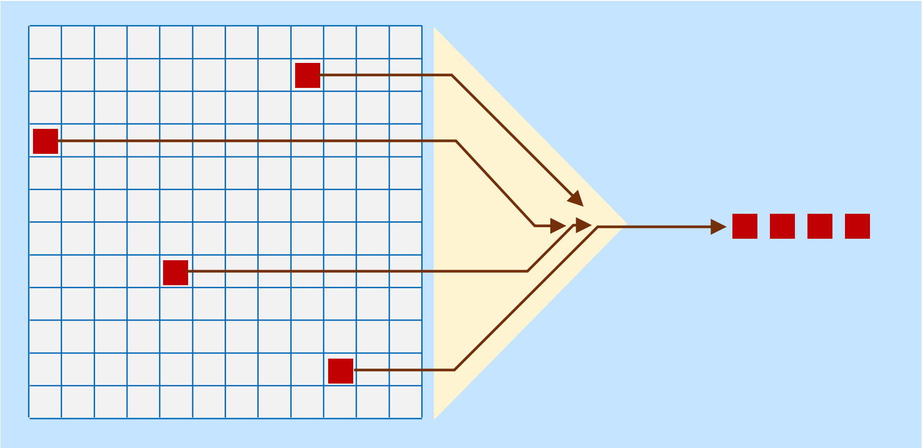
Chip implementation with asynchronous design

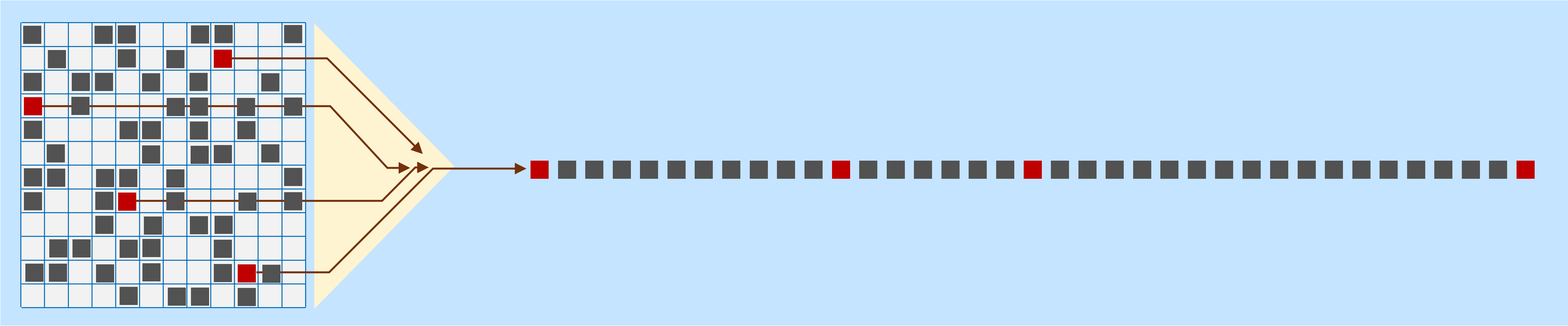
- Commercially developed over 2000-2015 at Fulcrum Microsystems and Intel
- Applied to five generation of commercial Ethernet switches
- Power consumption scales with activity – matched for spiking neurons and sparse interconnect
- Low latency communication – enables scaling to large systems
- Allows neuromorphic mesh to operate over 1000x faster than biological speeds using emergent synchronization
- Supports low energy, highly-ported SRAM arrays



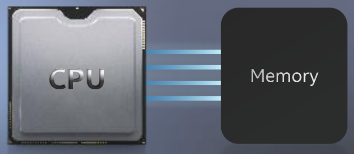
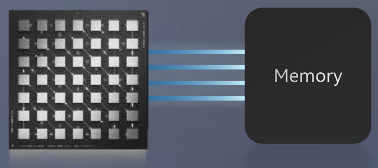
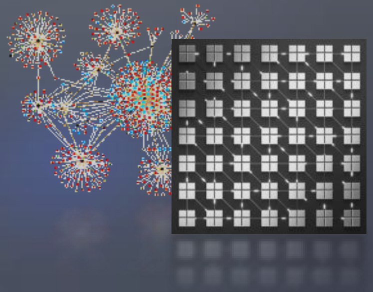
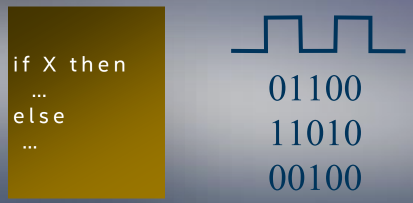
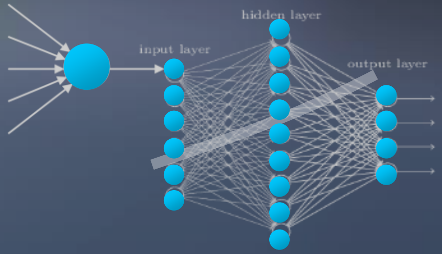
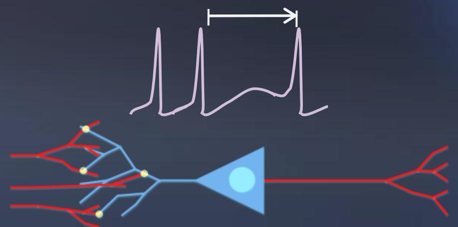


Sparse, asynchronous communication is fast





Leads us to a new class of computer architecture

Standard Computing	Parallel Computing	Neuromorphic Computing
		
PROGRAMMING BY ENCODING ALGORITHMS	OFFLINE TRAINING USING LABELED DATASETS	LEARN ON THE FLY THROUGH NEURON FIRING RULES
SYNCHRONOUS CLOCKING	SYNCHRONOUS CLOCKING	ASYNCHRONOUS EVENT-BASED SPIKES
SEQUENTIAL THREADS OF CONTROL	PARALLEL DENSE COMPUTE	PARALLEL SPARSE COMPUTE
		

Realized in Loihi

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

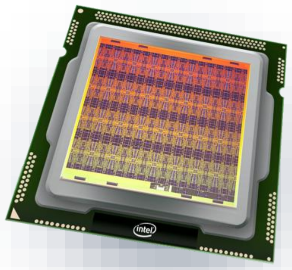
Davies et al, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning." IEEE Micro, Jan/Feb 2018.

Significant progress over three years of Loihi research

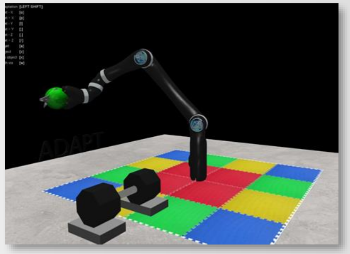
Loihi

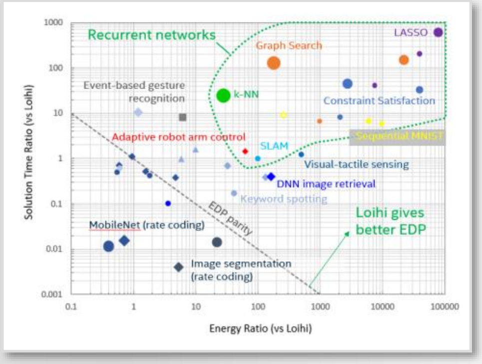
NxSDK

INRC



Asynchronous Design



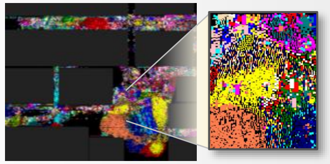


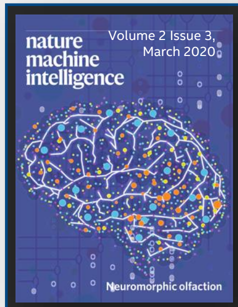
CIO JOURNAL

Intel to Release Neuromorphic-Computing System

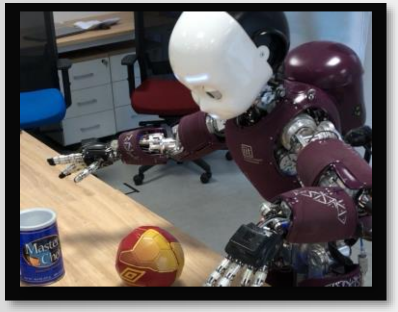
Pohoiki Springs, an experimental system to be rolled out this month, mimics the way human brains work to do computations faster with less energy

THE WALL STREET JOURNAL.

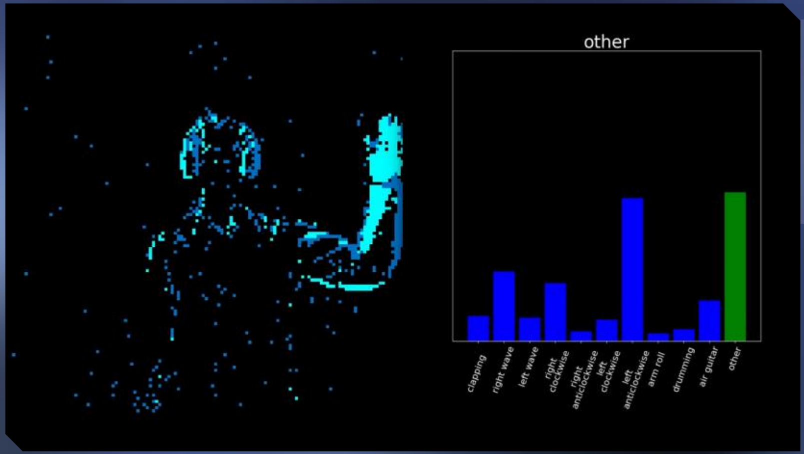




Neuro-inspired algorithm for odor recognition and learning demonstrated on Loihi, able to learn 3000x more data efficiently compared to DNN solution.



Loihi Has Confirmed the Value of This Direction

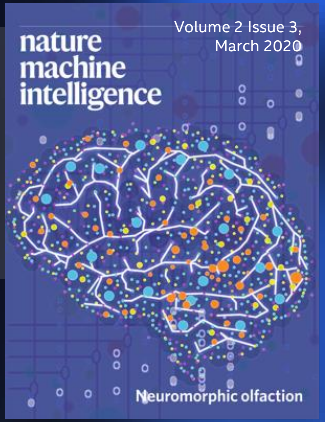


Gesture recognition + learning

Loihi + DAVIS 240C camera
60 mW total power, 15 mW dynamic[1]

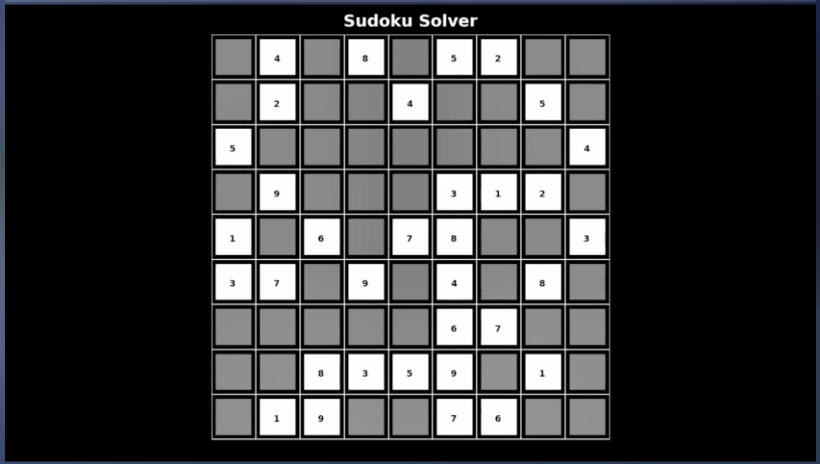
Olfaction-inspired odor recognition and learning

3000x more data efficient learning than a deep autoencoder



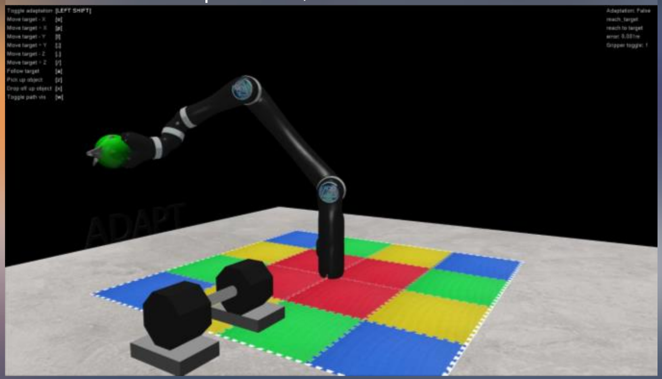
Combinatorial optimization (CSP, SAT, ILP, QP)

2,800x lower energy and 44x faster vs CPU[1]



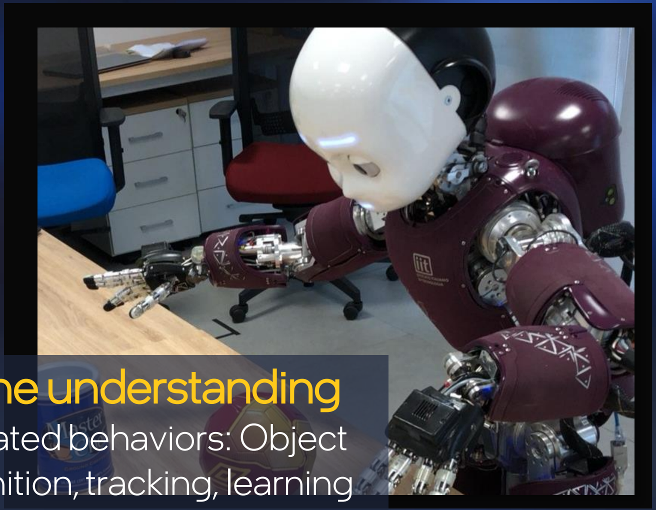
Adaptive robotic arm control

40x lower power, 50% faster vs GPU[1]



Scene understanding

Integrated behaviors: Object recognition, tracking, learning
100x lower power SLAM vs CPU[1]

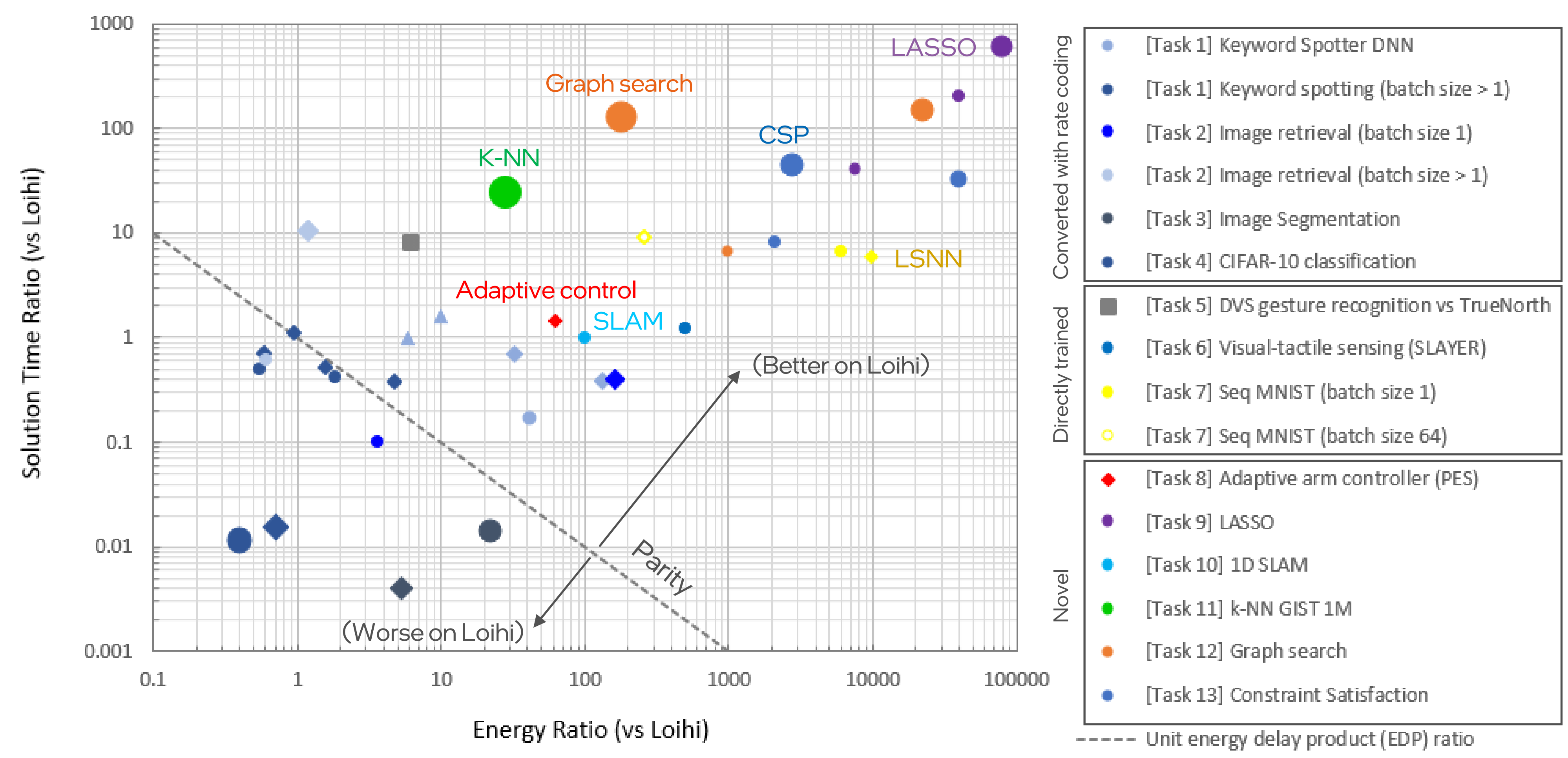


[1] *M. Davies et al, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," Proc. IEEE, 2021. Results may vary.*

For the right workloads, orders of magnitude gains in latency and energy efficiency are achievable

Reference architecture

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

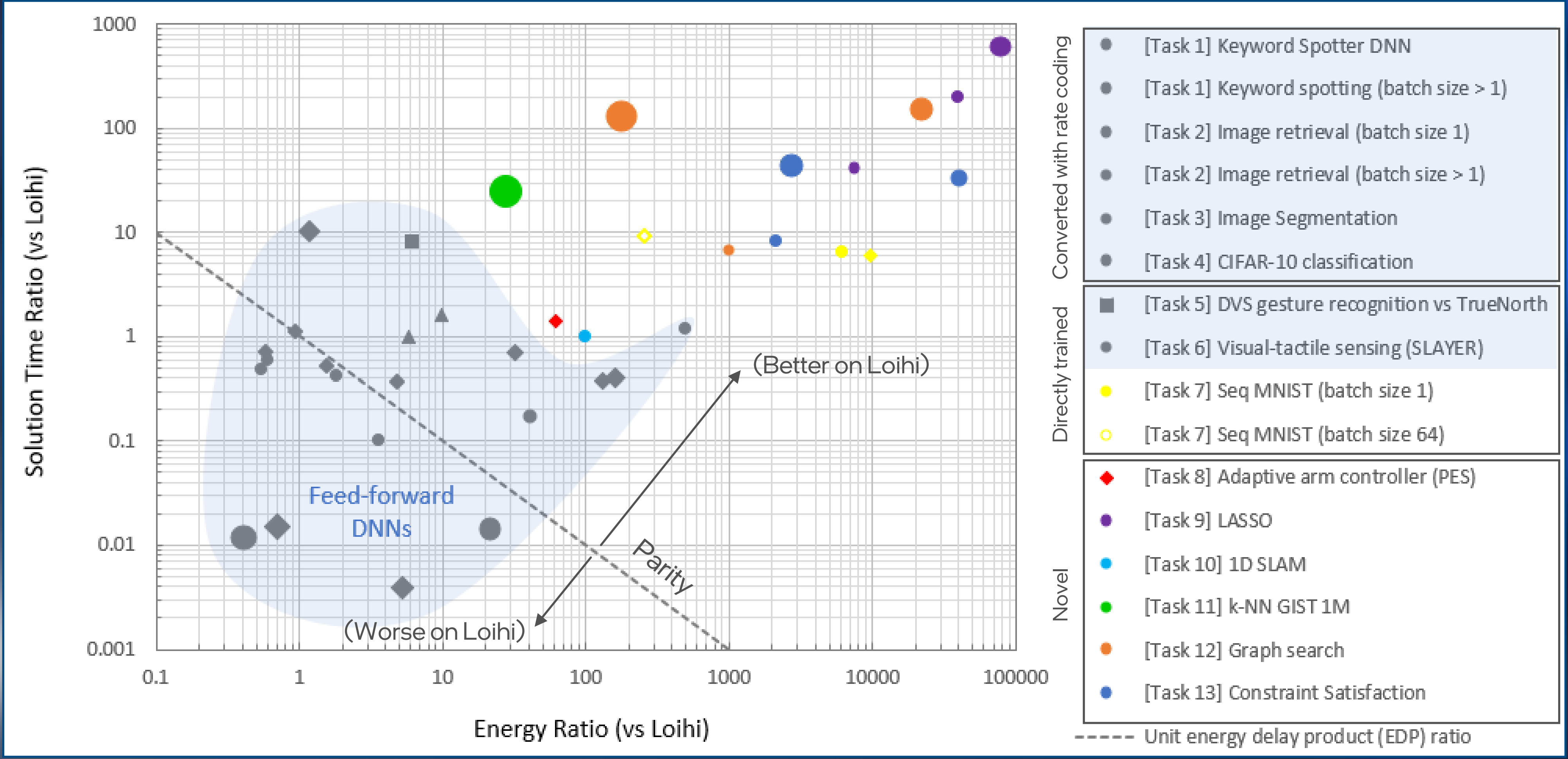


M. Davies et al, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," Proc. IEEE, 2021. Results may vary.

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

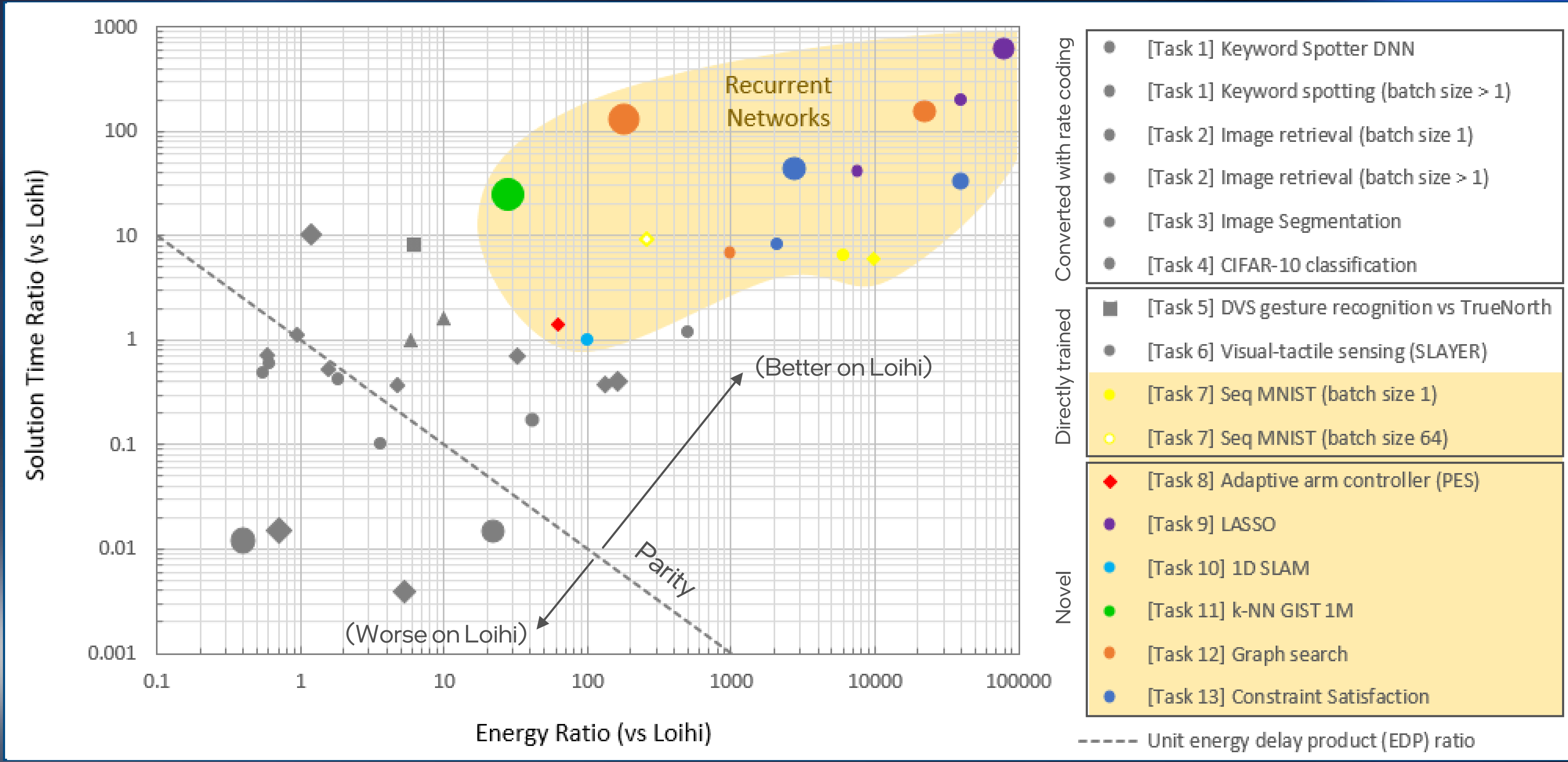


M. Davies et al, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," Proc. IEEE, 2021. Results may vary.

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

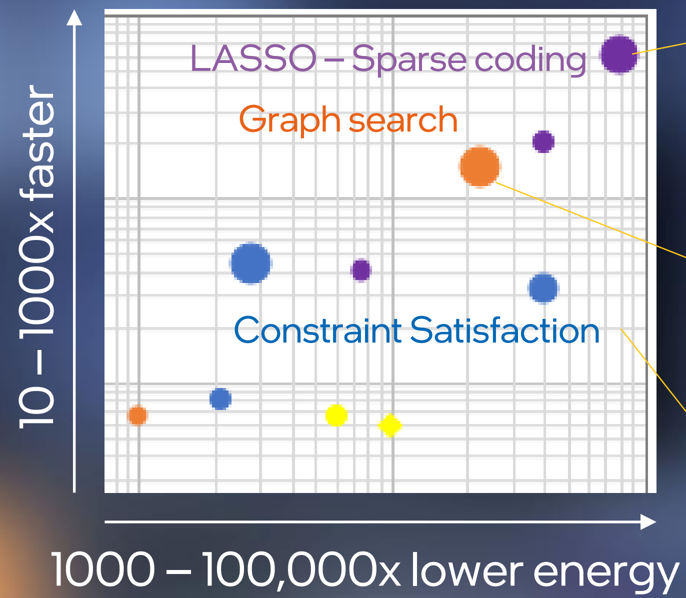
Reference architecture

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



M. Davies et al, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," Proc. IEEE, 2021. Results may vary.

Zooming in on the best examples: Optimization problems



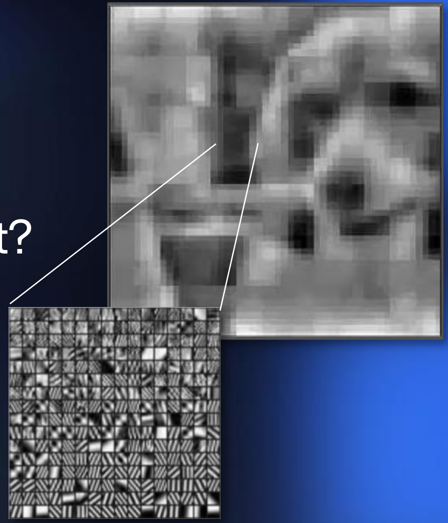
What features best explain the sensory input?

$$\underset{z}{\operatorname{argmin}} \|x - Dz\|_2^2 + \lambda\|z\|_1$$

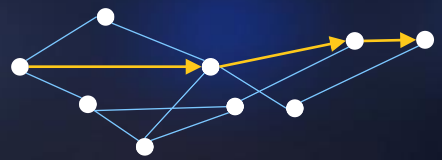
Input

Reconstruction

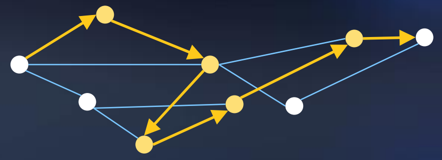
Sparse regularization



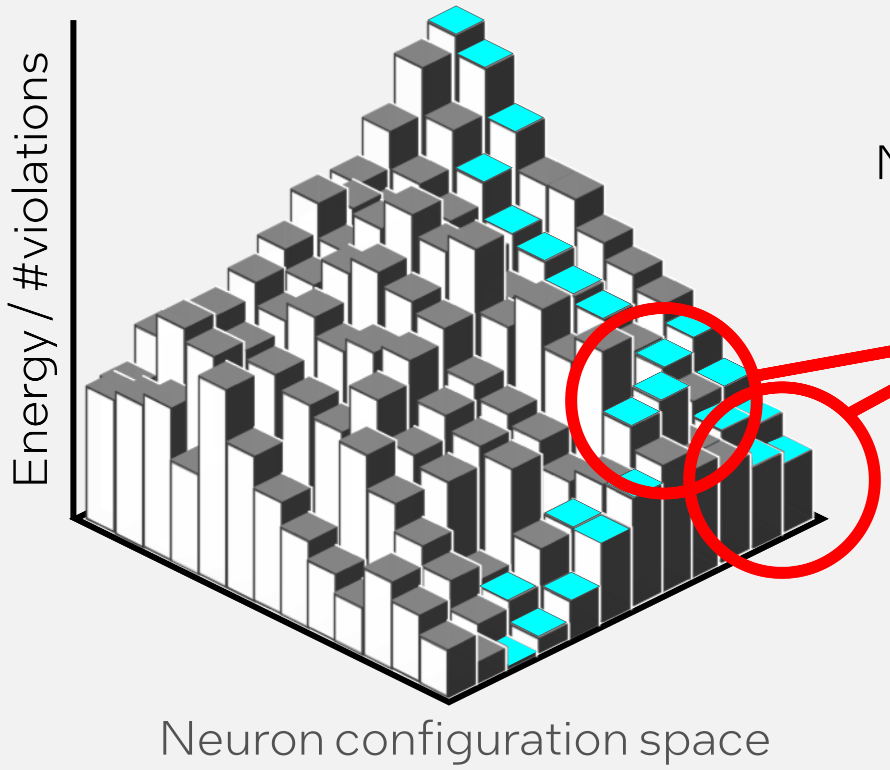
What is the shortest path to my goal?



What is the shortest path while visiting each waypoint exactly once?



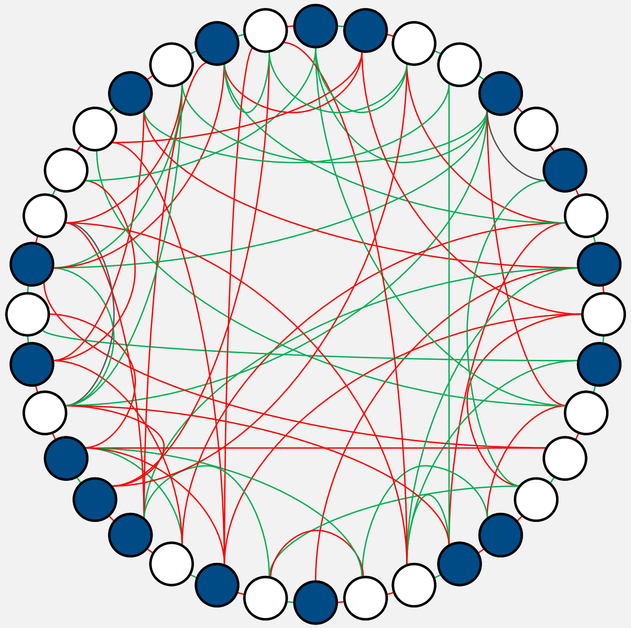
SNNs efficiently optimize via stochastic gradient descent



Neural dynamics descend the gradient

Local minimum
escaped by stochastic
spiking dynamics

Efficient descent due to massively
parallel, asynchronous neuromorphic
computing architecture

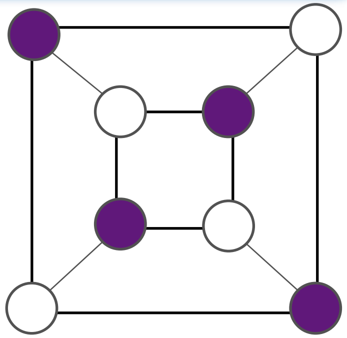


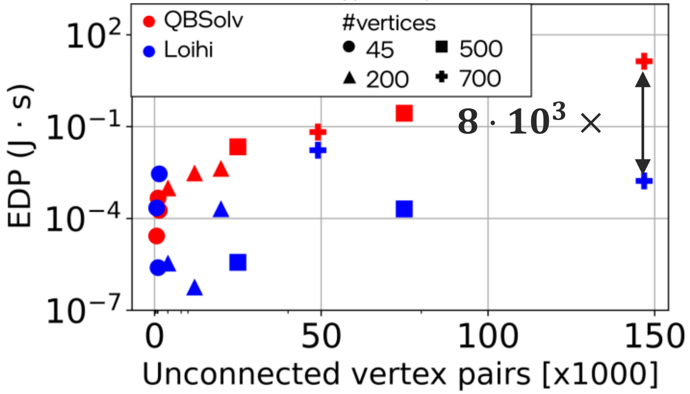
Loihi outperforms leading optimization solvers by orders of magnitude

QUBO (Maximum Independent Set)

Workload:
Find largest set of unconnected vertices

Relevance:
- Target of SOTA quantum annealing approaches
- NP hard



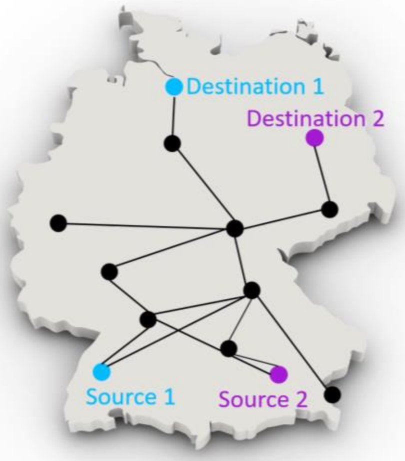


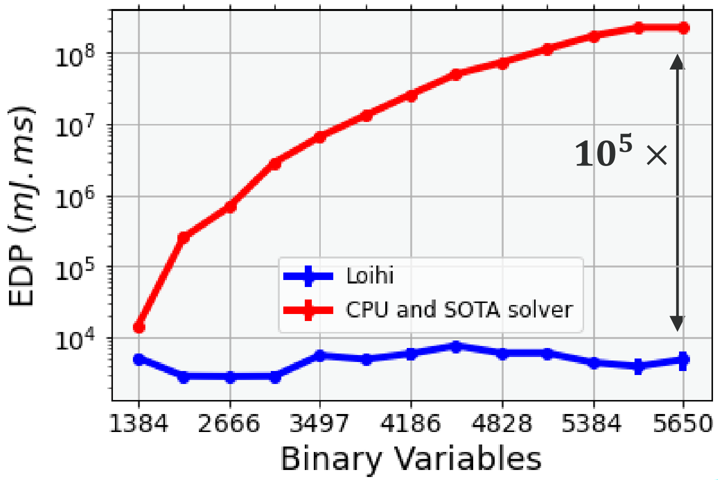
Integer Linear Programing (Train Scheduling)

In collaboration with:
DB

Workload:
Find the largest possible set of route assignments, given customer requests and railway, time and train constraints.

Relevance:
- Large-scale, real-world use case
- Applicable to resource allocation in warehouses and production lines.





Loihi: Nahuku board running NxSDK 0.95 with an Intel Core i7-9700K host with 128GB RAM, running Ubuntu 16.04.6 LTS
QUBO-QBSolv/CPU: benchmarks ran on an Intel Xeon CPU E5-2699 v3 @ 2.30GHz with 32GB DRAM (https://github.com/dwavesystems/qbsolv)
ILP-CPU: Xeon-based commercial cloud service as used operationally by DB. Solver runtime was measured; energy consumption estimated based on a 100W TDP estimate.

Performance results are based on testing as of September 2021 and may not reflect all publicly available security updates. Results may vary.

Generalizing neuromorphic optimization

Example Applications

Train scheduling	CSP
Route optimization	QUBO
Supply chain design	MILP
Job-shop scheduling	CSP
Flight gate assignment	QUBO

Prototype design	MILP
Material design	LP
Particle jet reconstruction	QUBO
Molecule structure prediction	QUBO

Trajectory optimization	QP
Coordinating mobile robots	MIQP
Model predictive control	QP
Image compression	CSP

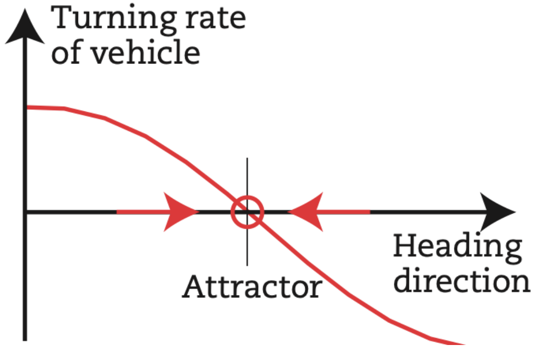
Optimization Problem Class

	Problem	Domain	Constraints	Cost
CSP	constraint satisfaction problems	$\mathbb{Z}^n$	≥ , = , …	Constant
ILP	integer linear programming	$\mathbb{Z}^n$	≥ , =	
LP	linear programming	$\mathbb{R}^n$	≥ , =	Linear
MILP	mixed-integer linear programming	$\mathbb{Z}^n \cup \mathbb{R}^n$	≥ , =	
QUBO	quadratic unconstrained binary optimization	$\{0,1\}^n$	/	
QP	quadratic programming	$\mathbb{R}^n$	≥ , =	Nonlinear: Quadratic
MIQP	mixed-integer quadratic programming	$\mathbb{Z}^n \cup \mathbb{R}^n$	≥ , =	

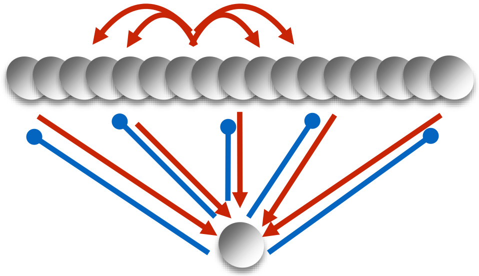
Available on Loihi
Work in progress
= Equality constraints
≥ Inequality constraints

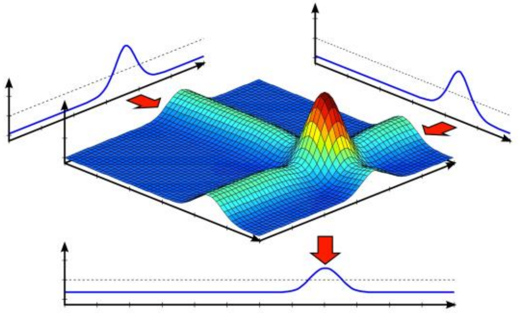
Programming with Attractor Networks: Dynamic Neural Fields

Neural dynamics designed to create attractor states



$$\tau\dot{\phi}(t) = -\phi(t) + A(t)$$

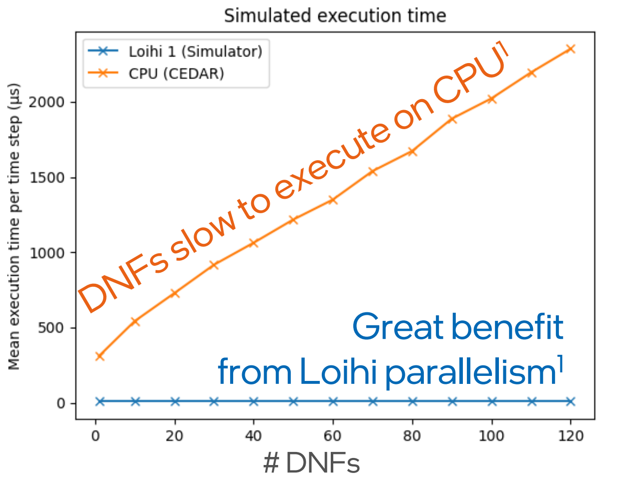




Attractor Networks as State Variables

- Memory – capture and hold inputs
- Attention – ignore distractors
- Defined state transitions

Execution time



[1]*See backup for characterization details.. Results may vary.*

Into a New Era of Neuromorphic Computing

Computational value is proven
(using today's manufacturing tech)

Motivates a new computational paradigm
(cheap, continuous optimization)

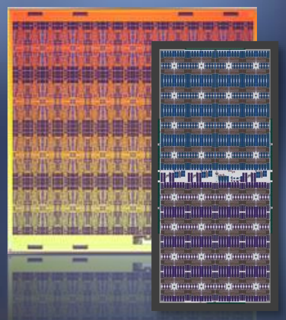
Many successful learning algorithms
(albeit shallow so far, not deep)

Properties of suitable applications:

- Power constrained
- Latency constrained
- Process real-time signals
- Slowly evolving structure
- Benefit from shallow online learning
- Apply deep learning for offline trainingc

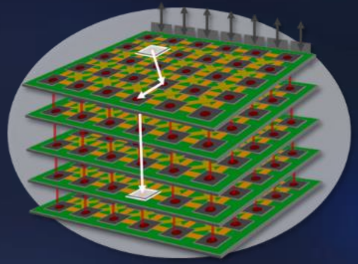
Outlook to Commercial Value

Today:



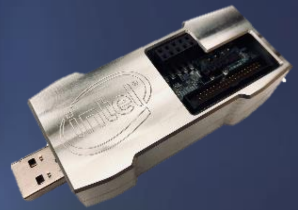
General-purpose research chips and software framework

Scaled up systems



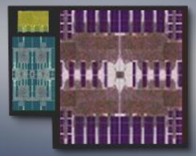
- Acceleration for datacenter optimization workloads
- Recommendation systems
- Scientific computing, HPC

Intelligent Extreme Edge Co-Processors

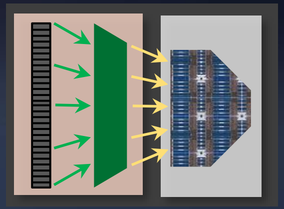


- Aerospace and robotics devices
- Scene awareness and localization
- Model predictive control
- Navigation and planning
- Consumer devices (longer term)

Specialized Designs



- Audio and other signal processing functions in SoCs
- Sensor integration (e.g. event-based cameras, electronic skins)
- Wireless signal processing and channel optimization
- IP and embedded accelerators for Intel Foundry customers



Challenges and headwinds

High cost due to on-chip memory integration

Algorithms and Programming models

Software convergence

A greatly improved Loihi 2 chip

10x Faster

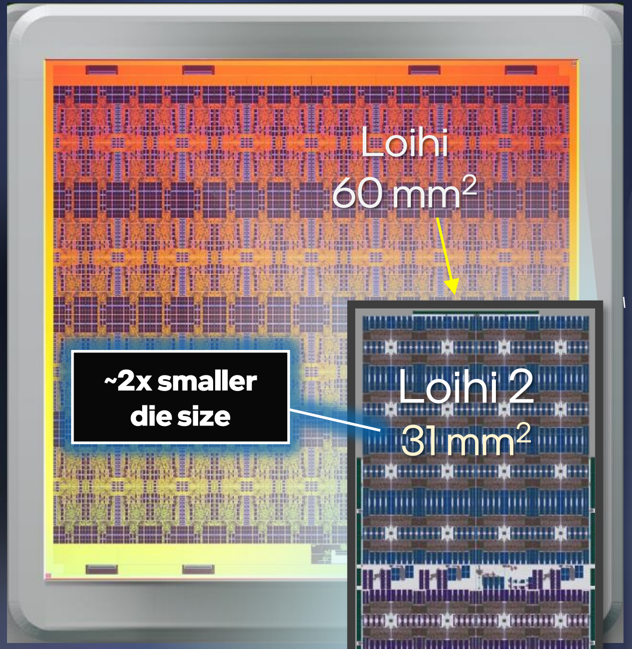
2-10x faster circuits[2] and design optimizations speed up workloads by up to 10x[3]

8x More Neurons

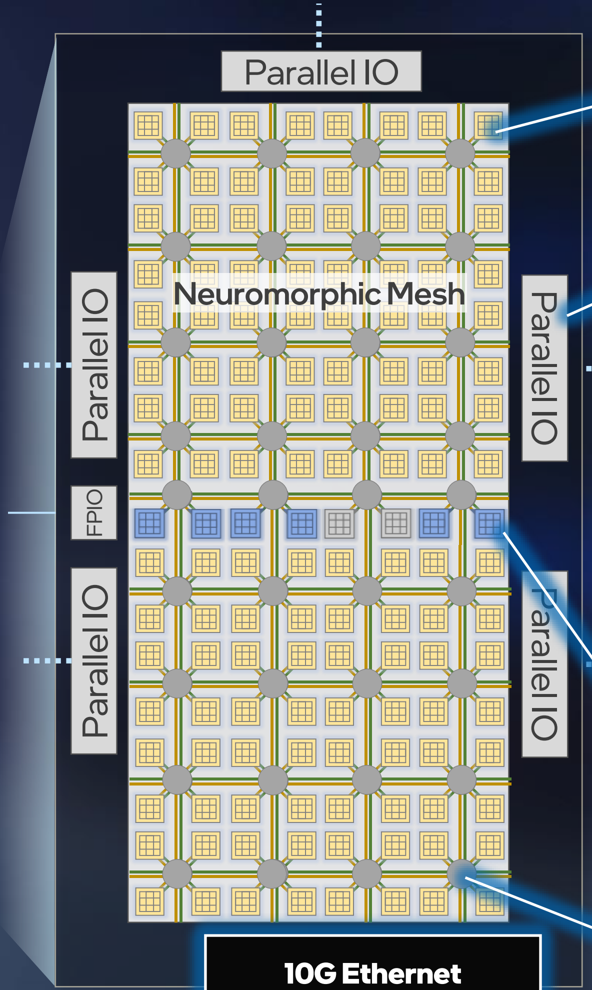
Up to 1 million neurons per chip with up to 80x better synaptic utilization, in 1.9x smaller die

Better Scaling and Integration

3D scaling with 4x more bandwidth per link[4], >10x compression[5] with standard interfaces

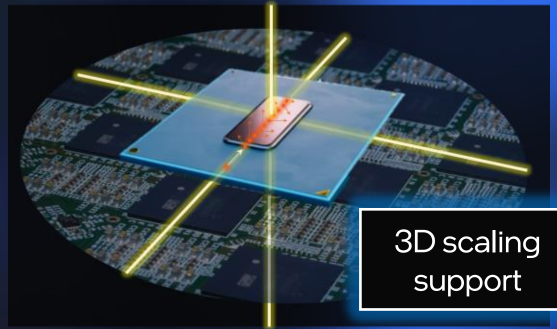


	Loihi 1	Loihi 2
Neuron cores:	128	128
Max neurons:	130K	→ 1M
Max synapses:	128M	123M
Max μP cores:	3	→ 6



Neuromorphic cores (128)
Fully redesigned with up to 8192 neurons

Parallel off-chip interfaces (6)
Async wave pipelined at 10 Gb/s with multicast compression



3D scaling support

Microprocessor cores (6)
Asynchronous x86 and RISC-V

Optimized Async NoC
<200ns min timesteps possible[6] with accelerated barrier synchronization

10G Ethernet
Accelerated host message + spike I/O

[2,3,4,5,6] See backup for characterization details.. Results may vary.

Generalized and optimized neuromorphic core

Generalized Spikes

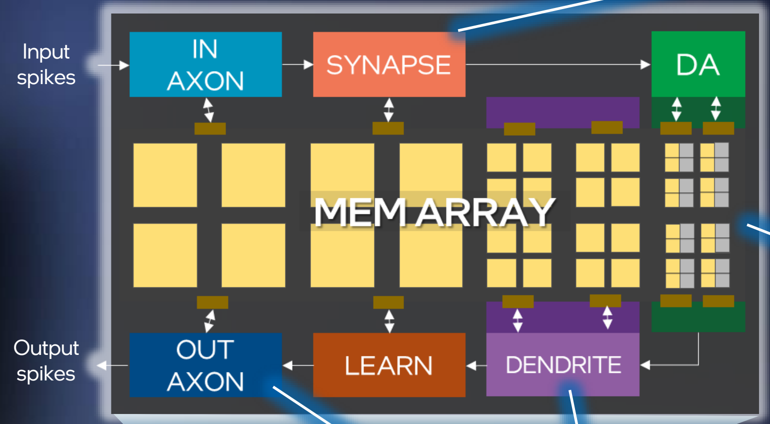
Spikes carry int8 magnitudes for greater workload precision

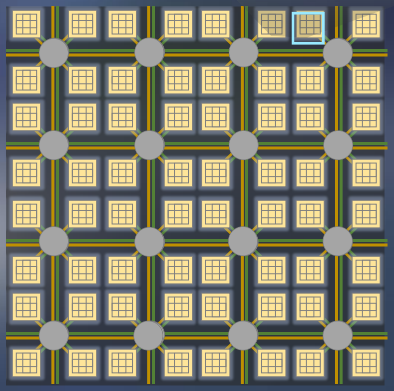
Programmable Neurons

Neuron models described by microcode instructions

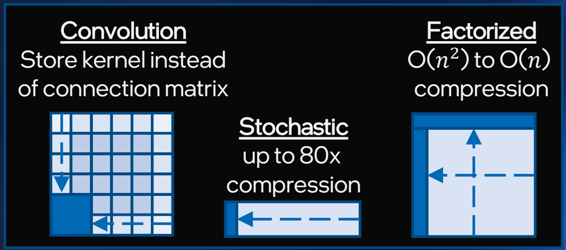
Enhanced Learning

Support for powerful new “three factor” learning rules from neuroscience



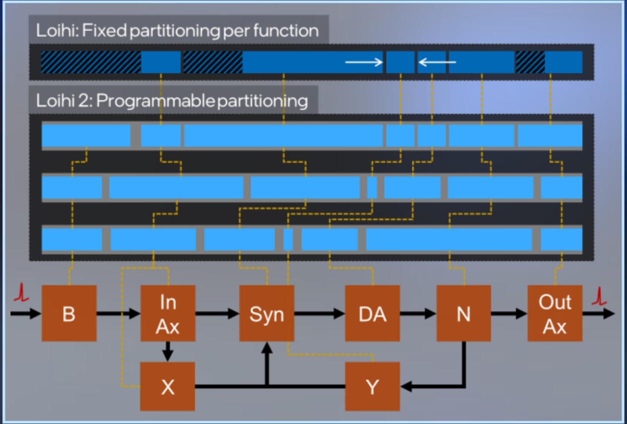


Better Synaptic Compression

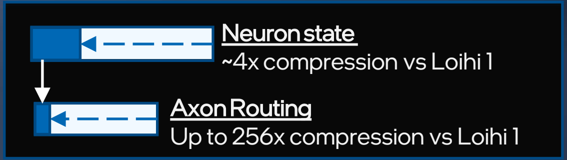


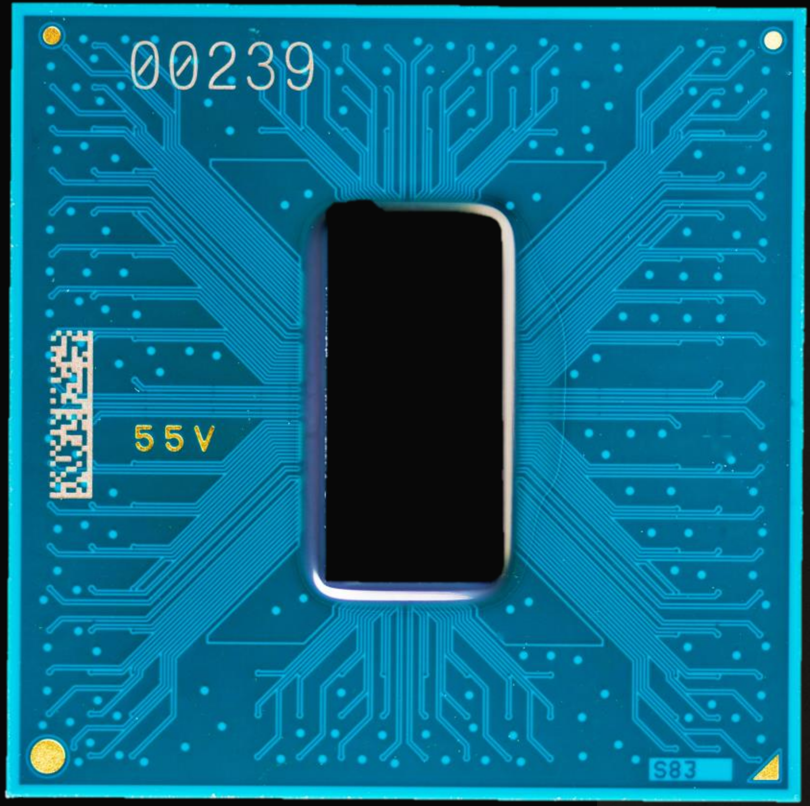
Better Utilization of Core Memory

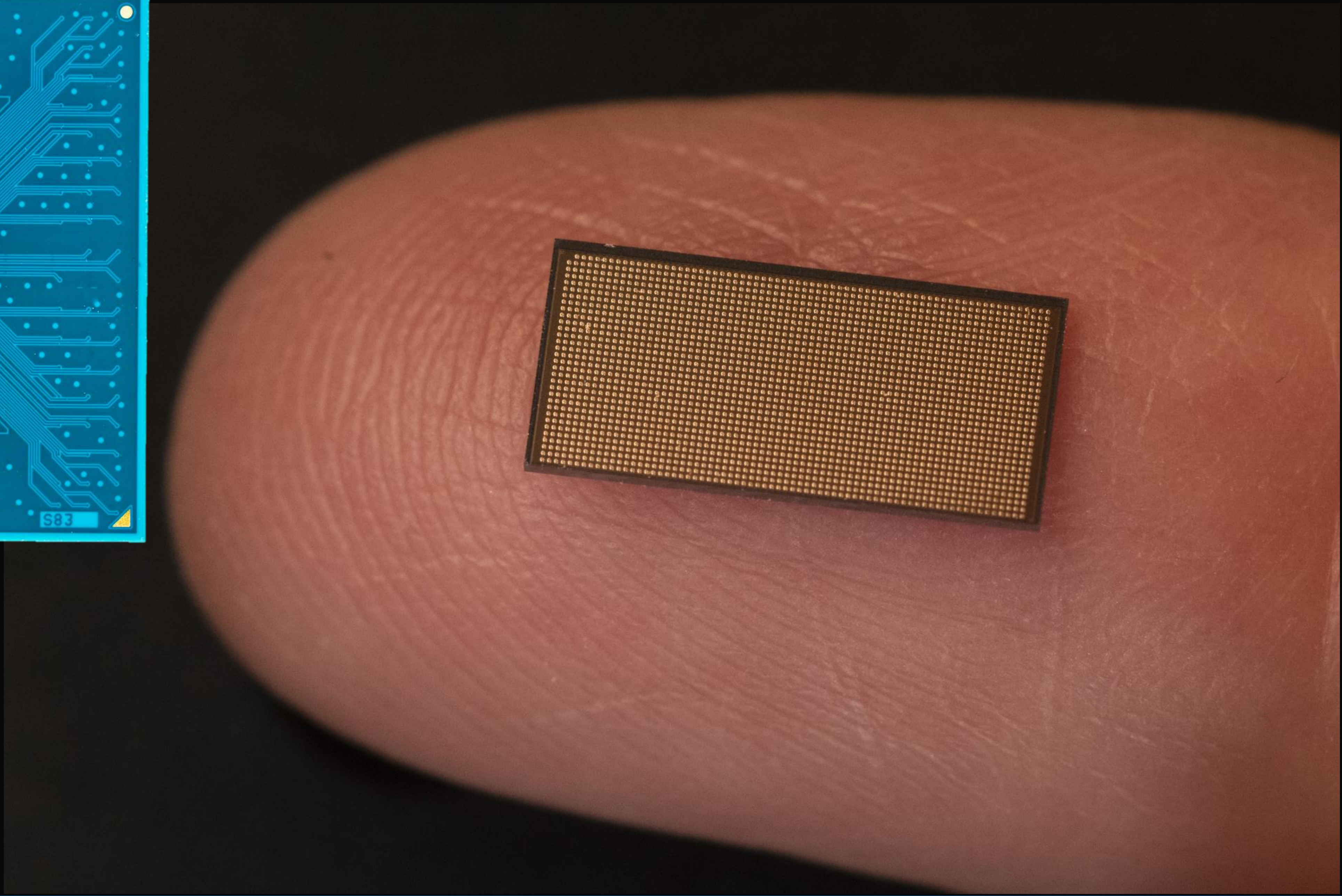
Highly ported centralized async memory array provides resource allocation flexibility



Better Neuron and Routing State Compression




00239
55V
S83



Loihi 2 systems and characterization

Oheo Gulch
Single-chip system

Kapoho Point
Stackable 8-chip board

Selected chip measurements

	Loihi 1 [7]	Loihi 2 [6]	Improvement
Neuron update time (ns)	9.6	4.4	2.2x faster
Synaptic Op time (ns)	4.0	0.66	6x faster
Minimum timestep (us)	1.57	0.19	8.3x faster
Neuron update energy (pJ)	70	56	25% lower
Synaptic Op energy (pJ)	21	7.8	2.7x lower

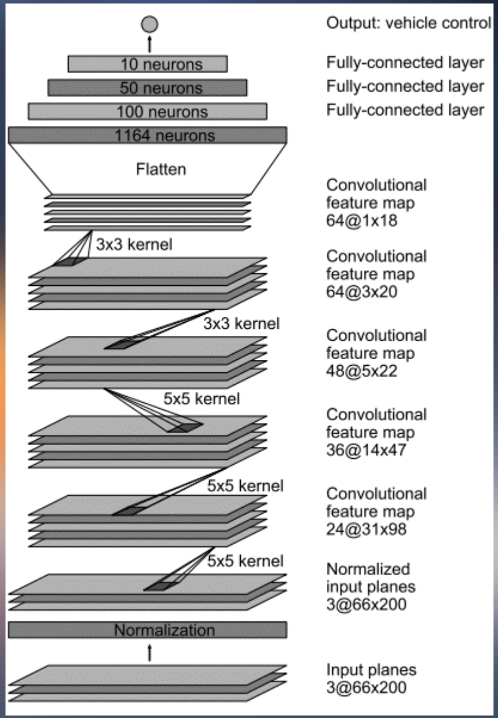
[6, 7] *See backup for characterization details.. Results may vary.*

Deep Learning with Loihi 2

PilotNet: Predict steering wheel angle from dashboard video

Bojarski, Mariusz et al. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316 (2016).

Loihi 2 greatly improves Loihi 1's weakest results
(Feed-forward DNNs)



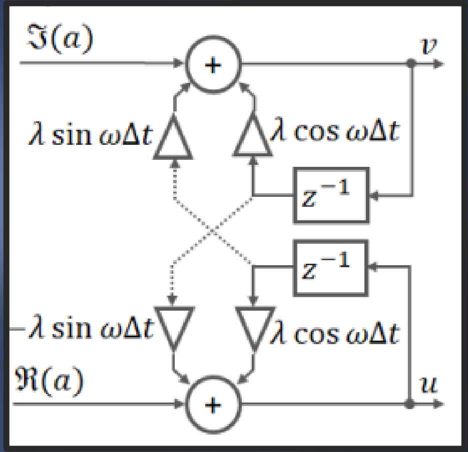
9-layer Convolutional network

	Loihi 1 LIF SNN [7]	Loihi 2 LIF SNN [6]		Loihi 2 - SDNN [3]	
Mean-Square-Error	0.049	0.049	1	0.037	32% lower
Neuron cores	368	70	5x smaller	70	5x smaller
Latency (ms)	15.5	2.56	6x faster	1.22	9-12x faster
Throughput (fps)	808	4877		7404	
TOPS (DNN equiv)	0.05	0.166		0.25	
Energy (uJ/frame)	1770	270	6.5x better	120	15x more efficient
TOPS/W (DNN equiv)	0.02	0.13		0.28	

LIF SNNs are rate-coded and direct trained. SDNN is a sigma-delta coded ReLU network
All networks are trained with lava-dl. Unbatched data processing

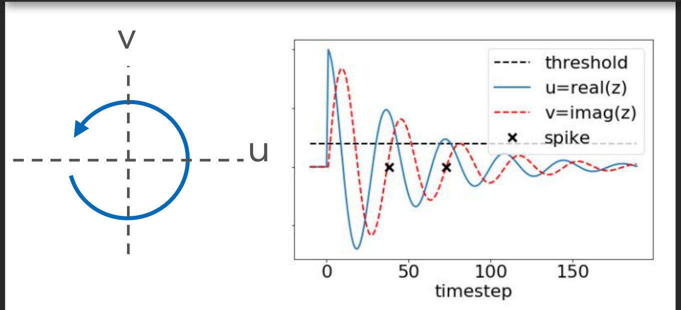
[3,6,7] See backup for characterization details.. Results may vary.

Other new Loihi 2 networks: Resonate-and-Fire neurons



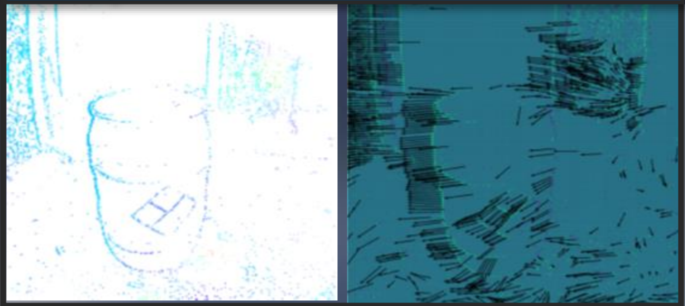
Resonate and Fire neurons compute optical flow for event-cameras with higher accuracy and 90x fewer ops than leading DNN solution

Resonate & Fire Neuron Behavior

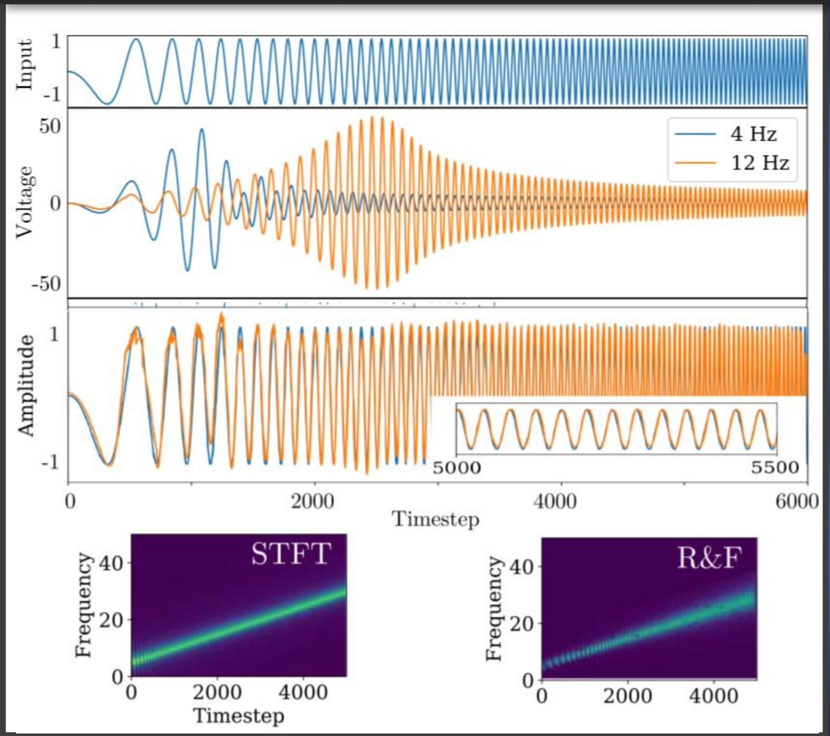


$$z_k[t] = \lambda e^{i\omega\Delta t} z_k[t-1] + a_k[t]$$

Optical Flow for Event-Cameras



Spectral Decomposition of an Audio Chirp



50x sparser output than conventional Short Time Fourier Transform

G. Orchard et al, "Efficient Neuromorphic Signal Processing with Loihi 2" IEEE International Workshop on Signal Processing Systems, Coimbra, Portugal, Oct 2021

LAVA

a new software framework for neuromorphic computing

Event-based communication
between simple parallel processes

Multi-Paradigm

Multi-Abstraction

Multi-Platform

Open source with permissive licensing of all core components

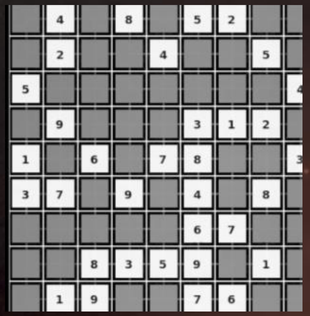
Today's SW for neuromorphic computing

	TensorFlow	PyTorch	Nengo	PyNN	NxSDK	BRIAN	ROS	Lava
Asynchronous message passing	☒	☒	☒	☒	☒	☒	☑	☑
CPU and GPU support	☑	☑	☑	☒	☒	☑	☑	☑
HW acceleration	☑	☑	☑	☑	☑	☒	☒	☑
Direct Backprop	☑	☑	☒	☒	☒	☒	☒	☑
Behavioral abstraction	☒	☒	☑	☒	☒	☒	☒	☑
Spiking neuron modeling	☒	☒	☑	☑	☑	☑	☒	☑
Permissive open source licensing	☑	☑	☒	☒	☒	☒	☑	☑

See https://github.com/lava-nc

Multi-Paradigm

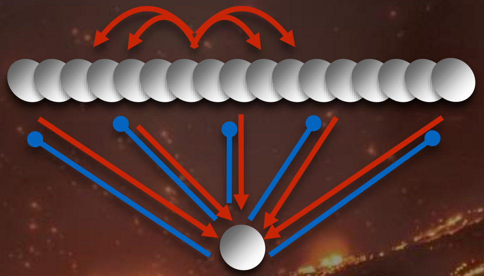
Optimization



LCA, Stochastic SNNs
LASSO, QP,
CSP, ILP, QUBO

+ model learning

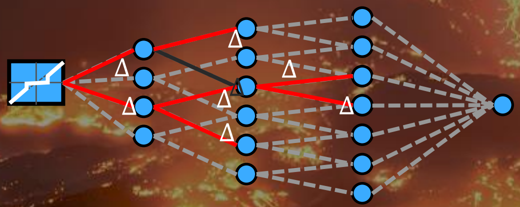
Neural Attractors



Dynamic Neural Fields,
Continuous Attractor NNs,
WTA

+ associative learning

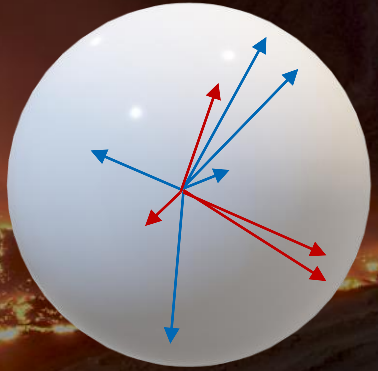
Deep Learning



ANN->SNN rate-coded conversion,
Directly trained SNN ConvNets
Sigma-Delta Neural Networks
TTFS- and Phase-coded SNNs

+ gradient learning

Vector Symbolic

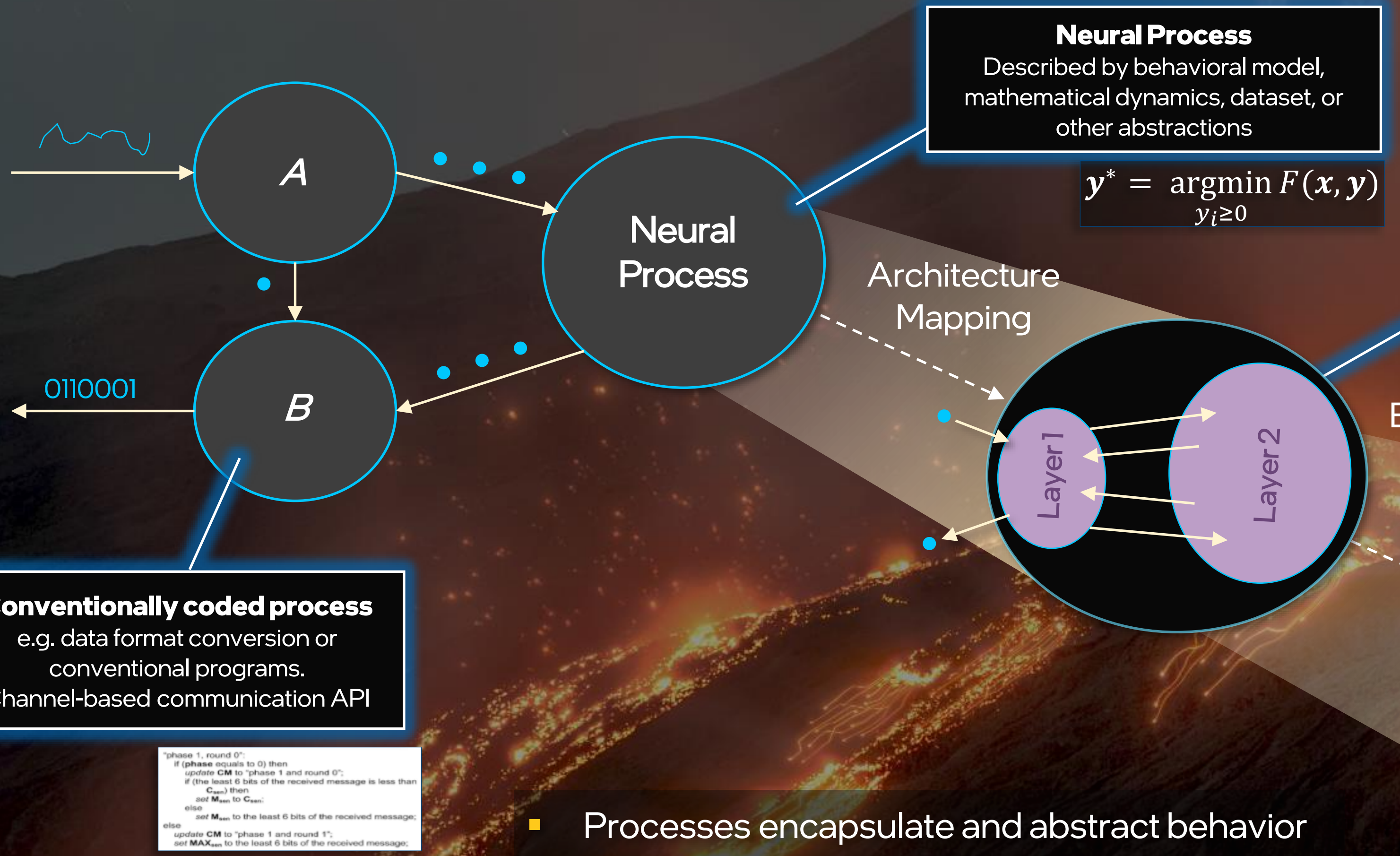


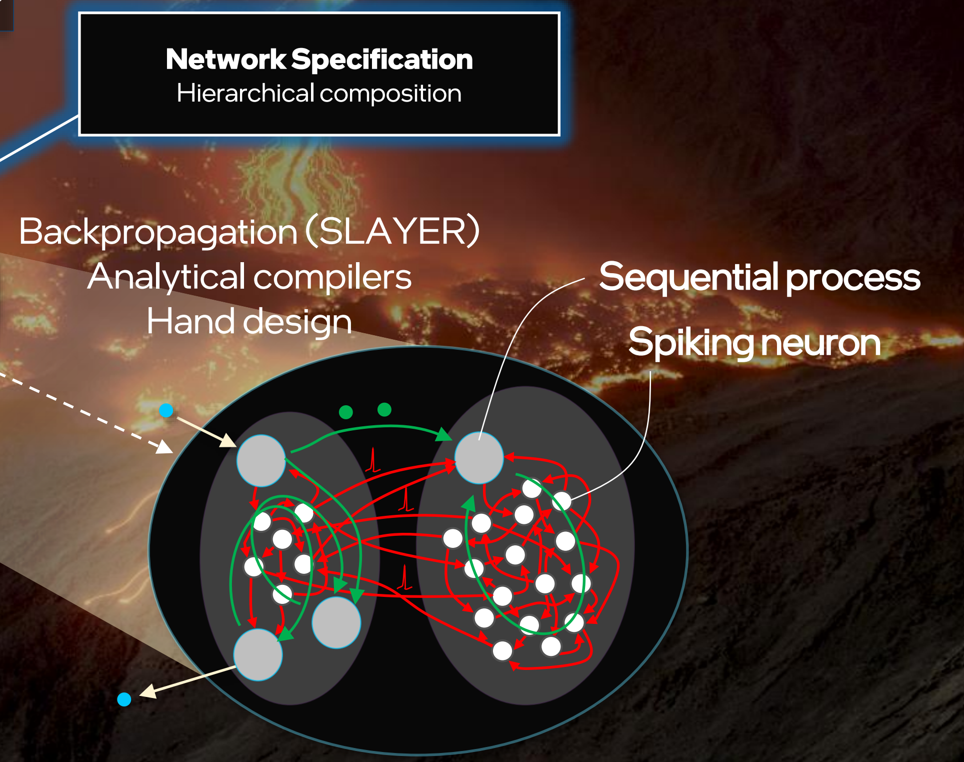
HRRs, MAPs,
Sparse Block Codes,
Associative Memories,
Resonator Networks

+ HD learning

Many others to come: NEF, Reservoir Computing, STICK, Equilibrium Propagation, evolutionary, ...

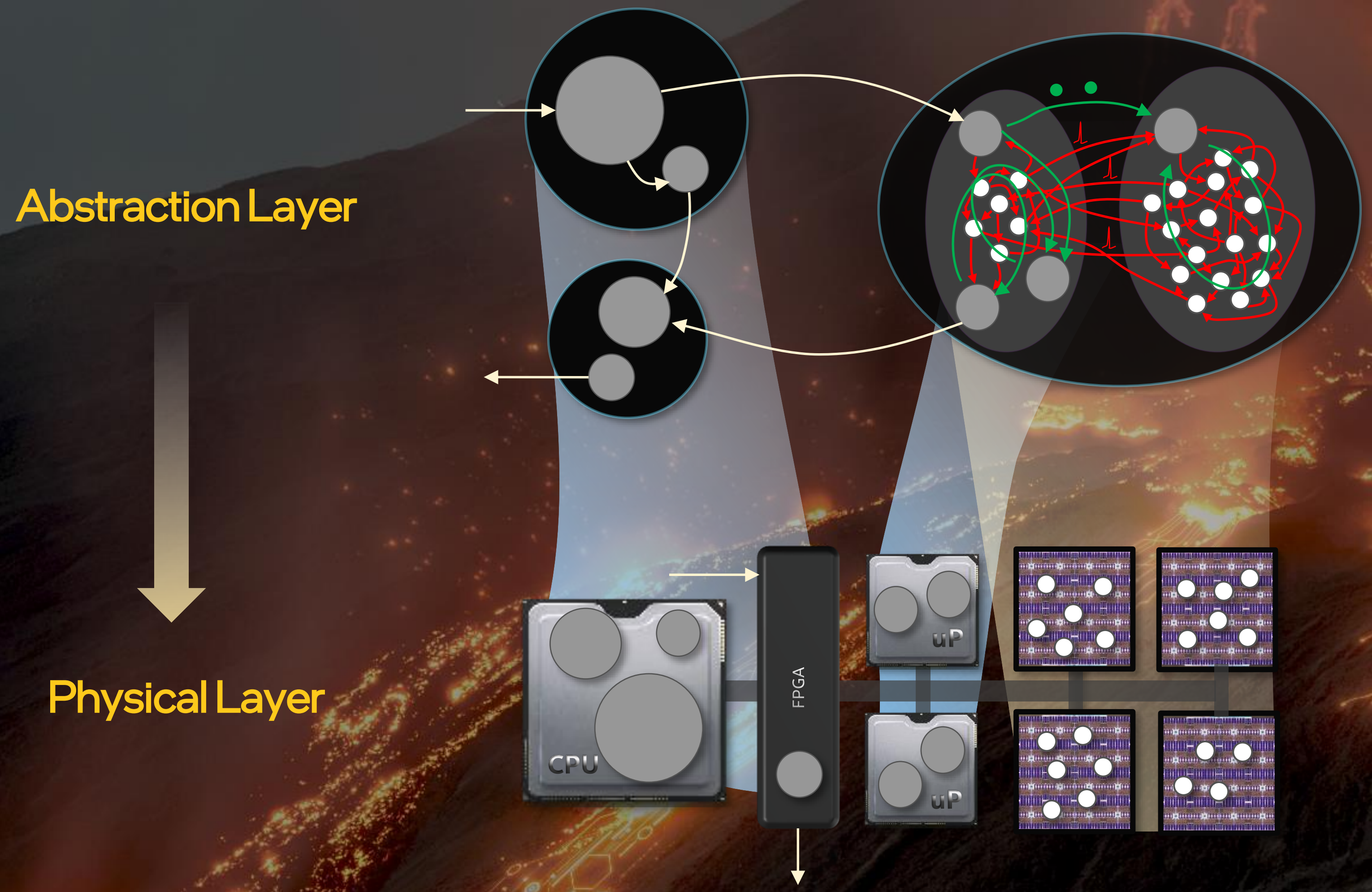
Multi-Abstraction





- Processes encapsulate and abstract behavior
- Hierarchical layers of abstraction
- Models describe *emergent behavior*

Multi-Platform

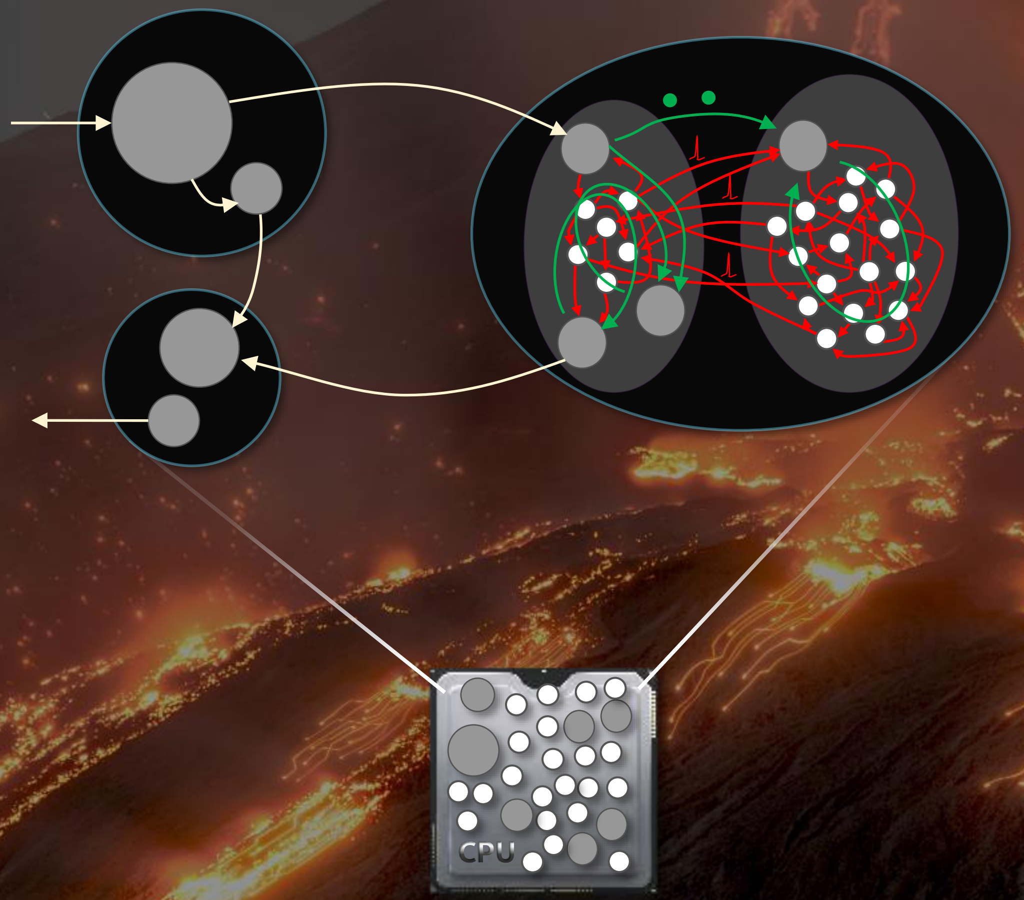


- Heterogeneous system architecture
- Multi-backend execution + profiling
- Fast compilation and execution
- Performant real-time operation

- CPU
- GPU
- FPGA
- Loihi 1
- Loihi 2
- Others…

CPU-only Execution for Exploration and Prototyping

Abstraction Layer



Physical Layer

- Heterogeneous system architecture
- Multi-backend execution + profiling
- Fast compilation and execution
- Performant real-time operation

- CPU
- GPU
- FPGA
- Loihi 1
- Loihi 2
- Others...

Performance Analysis Details

[1] CPU dynamic neural field measurements obtained using repo version of Cedar (https://cedar.ini.rub.de/) as of October 2021 running on an Intel Core i7-4720HQ CPU with four threads, 128GB RAM, with Ubuntu 18.04 OS. Loihi 1 simulation measurements obtained using a silicon-calibrated Lava profiling model (unreleased) as of September 2021. Each DNF is a 2D mesh attractor with 27x27 neurons, with one input DNF fanning out to all other DNFs operating in parallel.

[2] Based on comparisons between barrier synchronization time, synaptic update time, neuron update time, and neuron spike times between Loihi 1 and 2. Loihi 1 parameters measured from silicon characterization (see below); Loihi 2 parameters measured from both silicon characterization with N3B1 revision and pre-silicon circuit simulations using back-annotated timing for Loihi 2.

[3] Based on Lava simulations in September, 2021 of a nine-layer variant of the PilotNet DNN inference workload implemented as a sigma-delta neural network on Loihi 2 compared to the same network implemented with SNN rate-coding on Loihi. The Loihi 2 SDNN implementation gives better accuracy than the Loihi 1 rate-coded implementation. Equivalent DNN op counts calculated from a conventional DNN implementation with the same topology and same number of 8-bit parameters.

See Bojarski, Mariusz et al. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316 (2016).

[4] Circuit simulations of Loihi 2's wave pipelined signaling circuits show 800 Mtransfers/s compared to Loihi 1's measured performance of 185 Mtransfers/s.

[5] Based on analysis of 3-chip and 7-chip Locally Competitive Algorithm examples.

[6] Loihi 1 measurements were obtained on Oheo Gulch FMC board ncl-og-06 using an internal version of NxSDK advanced from v1.0.0

[7] Loihi 2 measurements were obtained on Nahuku 32 board ncl-ghrd-01 using NxSDK v1.0.0

The Lava performance model for both chips is based on silicon characterization in September 2021 using the Nx SDK release 1.0.0 with an Intel Xeon E5-2699 v3 CPU @ 2.30 GHz, 32GB RAM, as the host running Ubuntu version 20.04.2. Loihi results use Nahuku-32 system ncl-ghrd-04. Loihi 2 results use Oheo Gulch system ncl-og-04.

Results may vary.

Thank You!

Email inrc_interest@intel.com for more information
Visit https://github.com/lava-nc to get started with Lava