Harvard John A. Paulson School of Engineering and Applied Sciences

WHERE SCIENCE AND ENGINEERING CONVERGE

NeuroBench

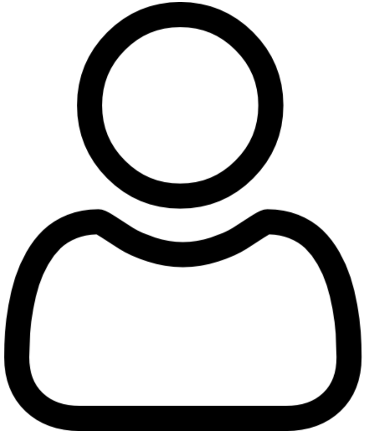
A Framework for Benchmarking Neuromorphic Computing Algorithms and Systems

Jason Yik, Harvard University

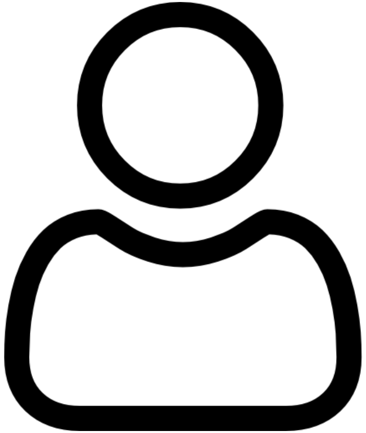
(Representing the work of many contributors)

100+ community members, 50+ represented institutions

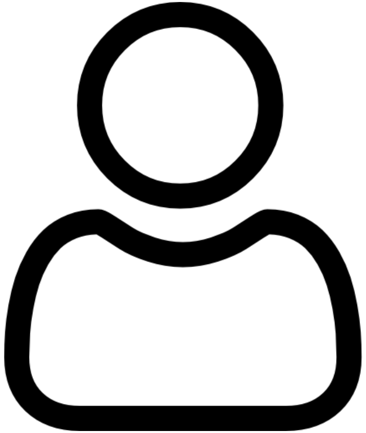
<u>Benchmark</u>



I'm fast



I'm *real* fast

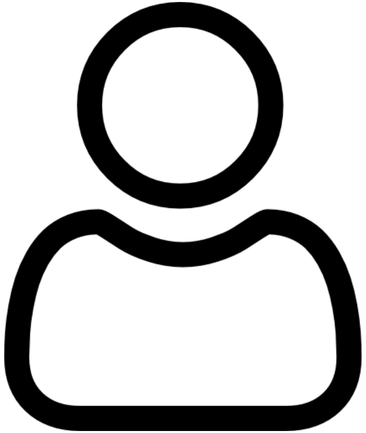


Me most
fastest

What does fast mean?

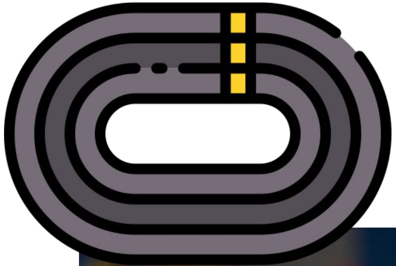
I'm fast

I'm *real* fast



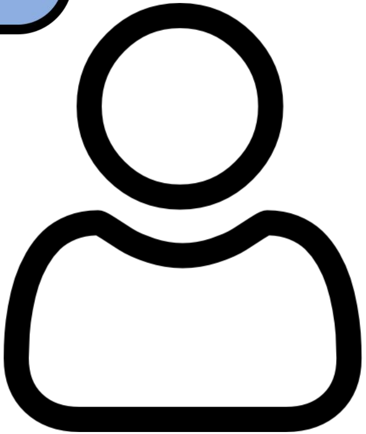
Me most
fastest

Benchmarks standardize what performance means and how it is measured

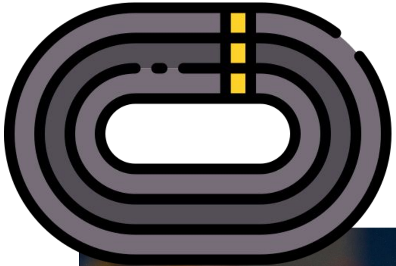


I’m fast

I’m *real* fast



Me most fastest



Benchmarks standardize what performance means and how it is measured

They provide a way to rank / compare performance

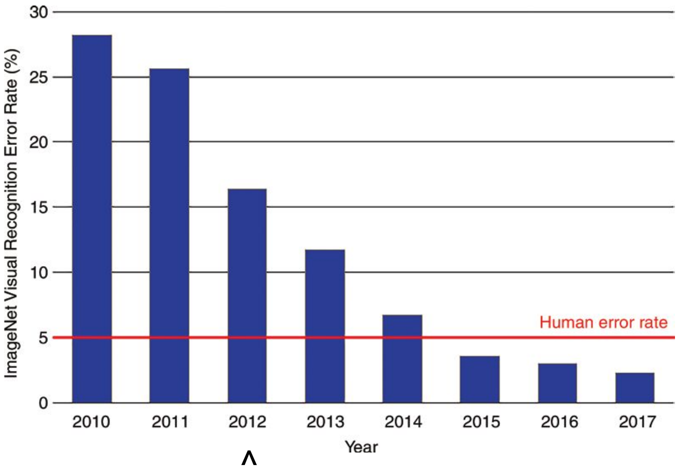
I’m fast

I’m *real* fast

Me most fastest

Benchmark

ImageNet:
A <u>benchmark</u> for deep learning

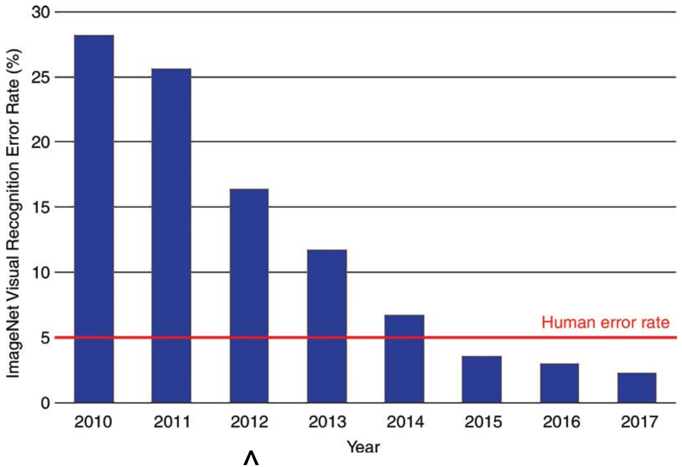


^

AlexNet, sparked deep learning

https://www.researchgate.net/figure/Error-rates-on-the-ImageNet-Large-Scale-Visual-Recognition-Challenge-Accuracy_fig1_332452649

ImageNet:
A <u>benchmark</u> for deep learning

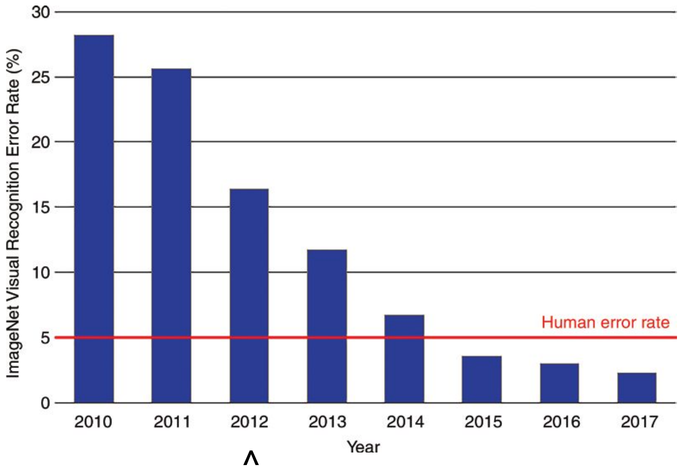


^

AlexNet, sparked deep learning

Task: Image classification
Dataset: Imagenet Corpus
Metric: Top1 Classification-accuracy

ImageNet:
A <u>benchmark</u> for deep learning



^

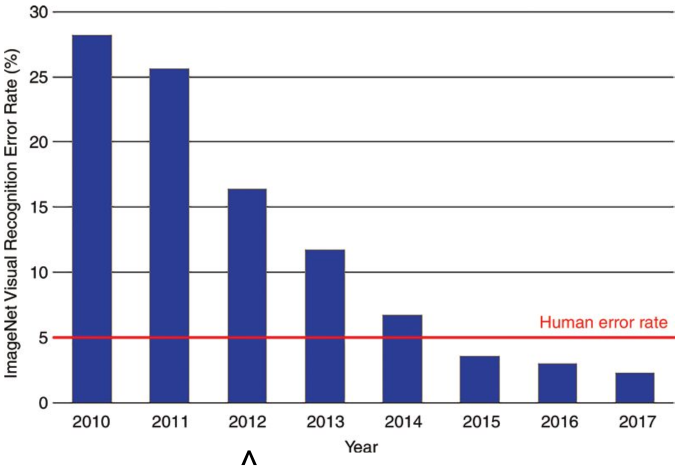
AlexNet, sparked deep learning

Task: Image classification
Dataset: Imagenet Corpus
Metric: Top1 Classification-accuracy

Benchmarks:
- Align on challenges of interest
- Measure SOTA and growth
- Spur research progress

https://www.researchgate.net/figure/Error-rates-on-the-ImageNet-Large-Scale-Visual-Recognition-Challenge-Accuracy_fig1_332452649

ImageNet:
A benchmark for deep learning



^

AlexNet, sparked deep learning

Task: Image classification
Dataset: Imagenet Corpus
Metric: Top1 Classification-accuracy

Benchmarks:
- Align on challenges of interest
- Measure SOTA and growth
- Spur research progress

Standard benchmarks are vital for technological progress.

https://www.researchgate.net/figure/Error-rates-on-the-ImageNet-Large-Scale-Visual-Recognition-Challenge-Accuracy_fig1_332452649

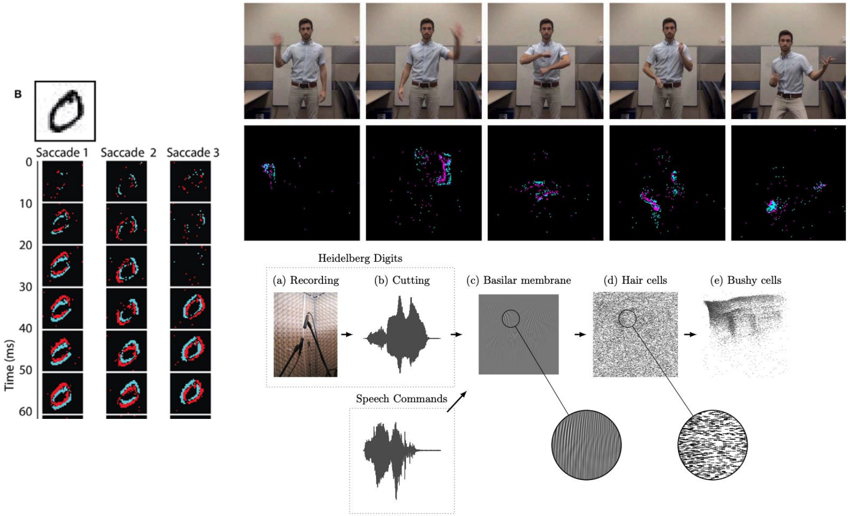
Benchmarks for neuromorphic computing

Deep learning benchmarks?

Lack native temporal dimension

Benchmarks for neuromorphic computing

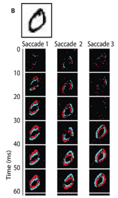
Deep learning benchmarks?

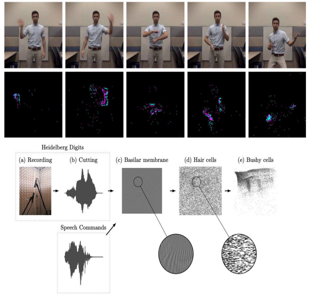


Prior benchmarks
e.g. N-MNIST, DVS Gesture, SHD

Benchmarks for neuromorphic computing

Deep learning benchmarks?





Prior benchmarks

Comment | Published: 11 September 2019

Benchmarks for progress in neuromorphic computing

Mike Davies

Nature Machine Intelligence **1**, 386–388 (2019) | Cite this article

2447 Accesses | **62** Citations | **22** Altmetric | Metrics

Perspective | Published: 31 January 2022

Opportunities for neuromorphic computing algorithms and applications

Catherine D. Schuman, Shruti R. Kulkarni, Maryam Parsa, J. Parker Mitchell, Prasanna Date & Bill Kay

Nature Computational Science **2**, 10–19 (2022) | Cite this article

84k Accesses | **238** Citations | **161** Altmetric | Metrics

Benchmark calls to action

Challenges in Benchmarking Neuromorphics

#1

Lack of a
formal definition

#2

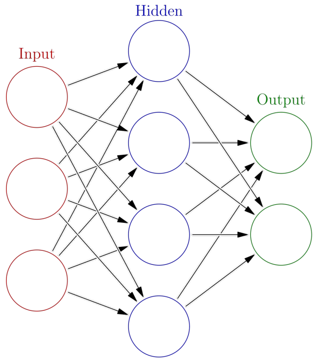
Implementation
diversity

#3

Rapid
research evolution

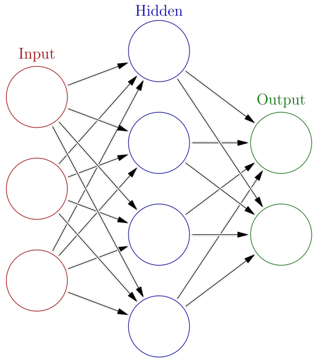
#1: Lack of a formal definition

Neuromorphic == “biologically-inspired”



#1: Lack of a formal definition

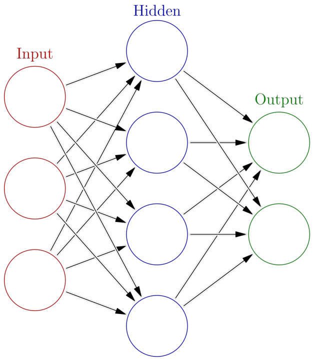
Neuromorphic == “biologically-inspired”



GPT-4

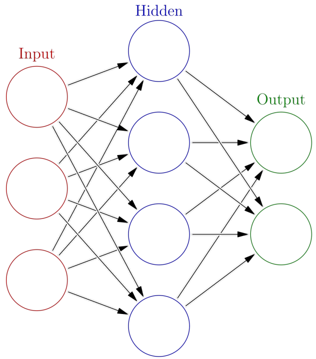
#1: Lack of a formal definition

Neuromorphic == “biologically-inspired”



#1: Lack of a formal definition

Neuromorphic == “biologically-inspired”

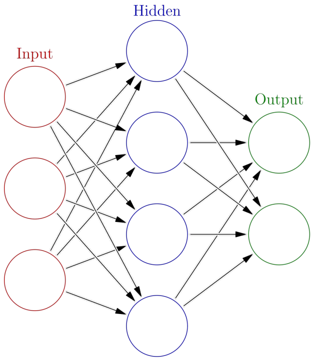


GPT-4

All of deep learning is “neuromorphic”

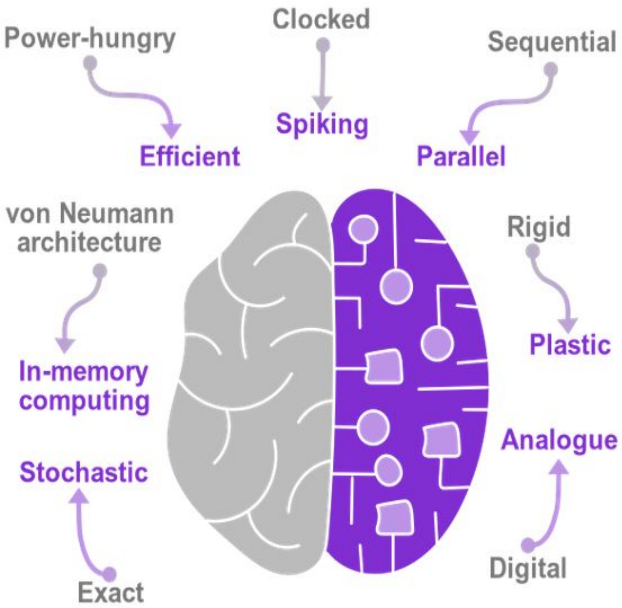
#1: Lack of a formal definition

Neuromorphic == “biologically-inspired”



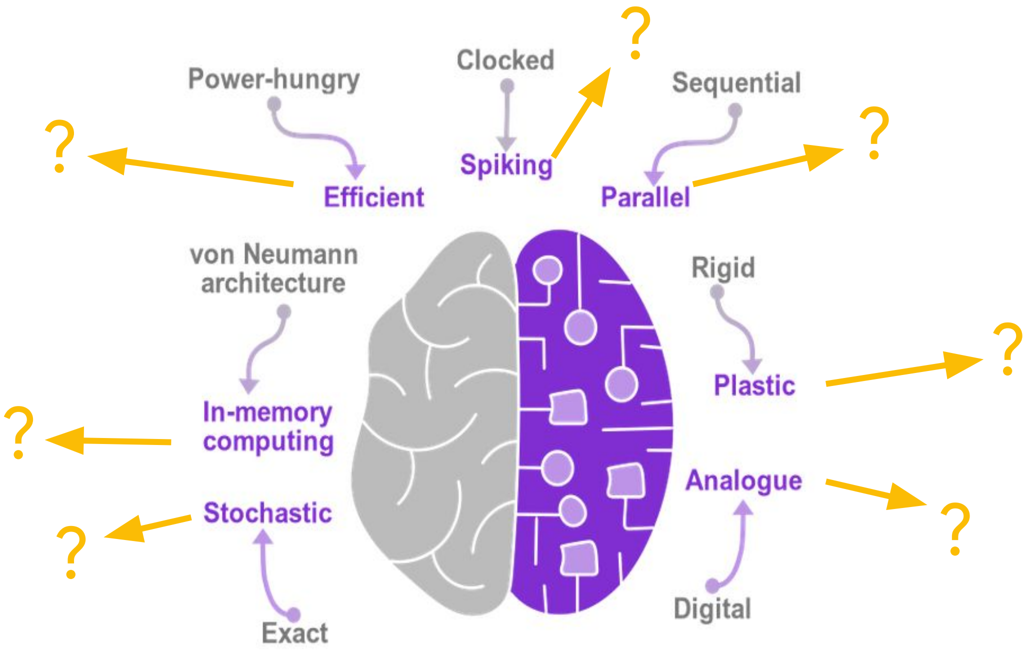
All of deep learning is “neuromorphic”

#1: Lack of a formal definition



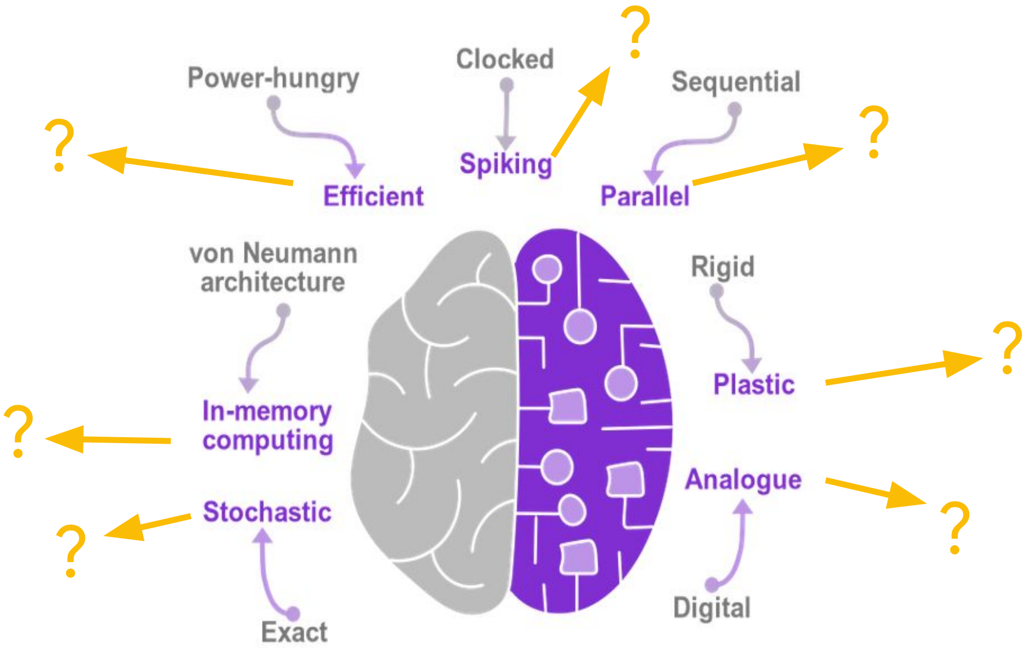
[Example from the NeuroTech EU Consortium]

#1: Lack of a formal definition



[Example from the NeuroTech EU Consortium]

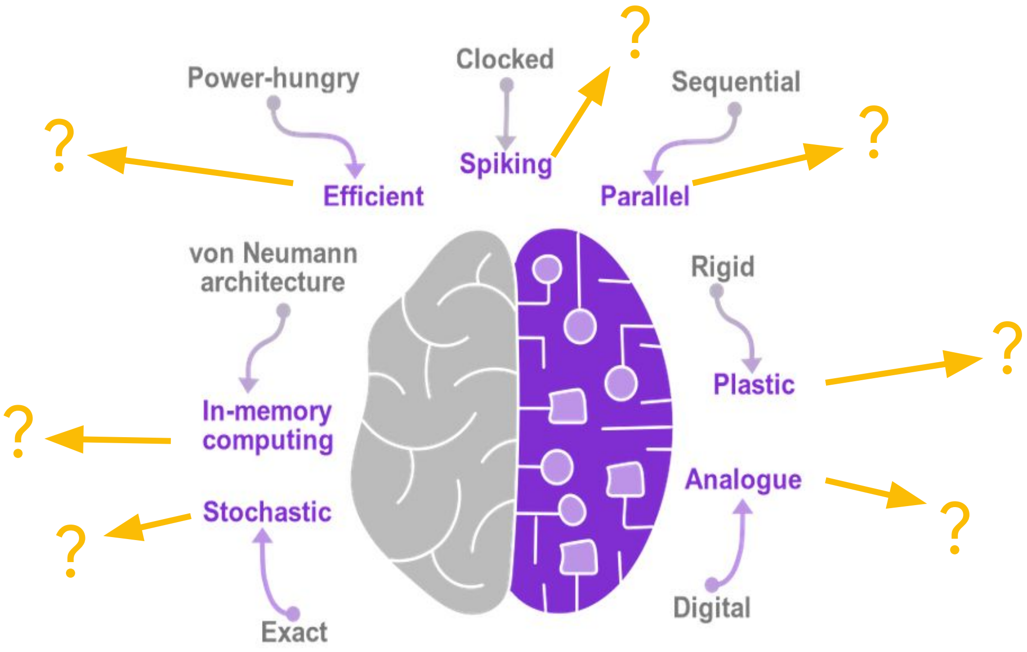
#1: Lack of a formal definition



[Example from the NeuroTech EU Consortium]

What should be benchmarked as “neuromorphic”?

#1: Lack of a formal definition

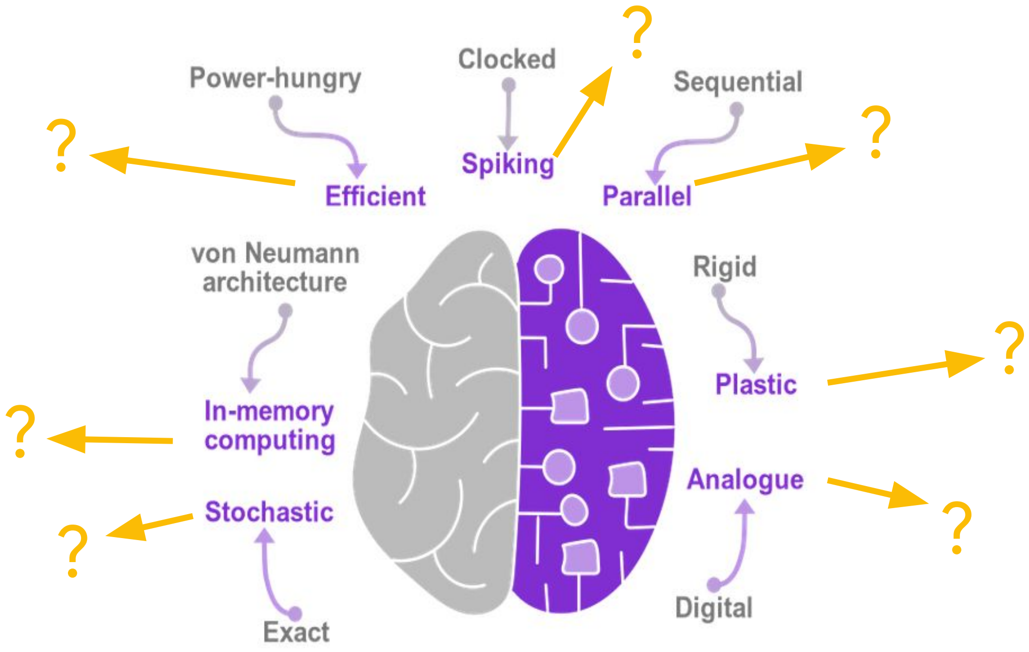


[Example from the NeuroTech EU Consortium]

What should be benchmarked as “neuromorphic”?

Necessitates *inclusive* benchmarks

#1: Lack of a formal definition



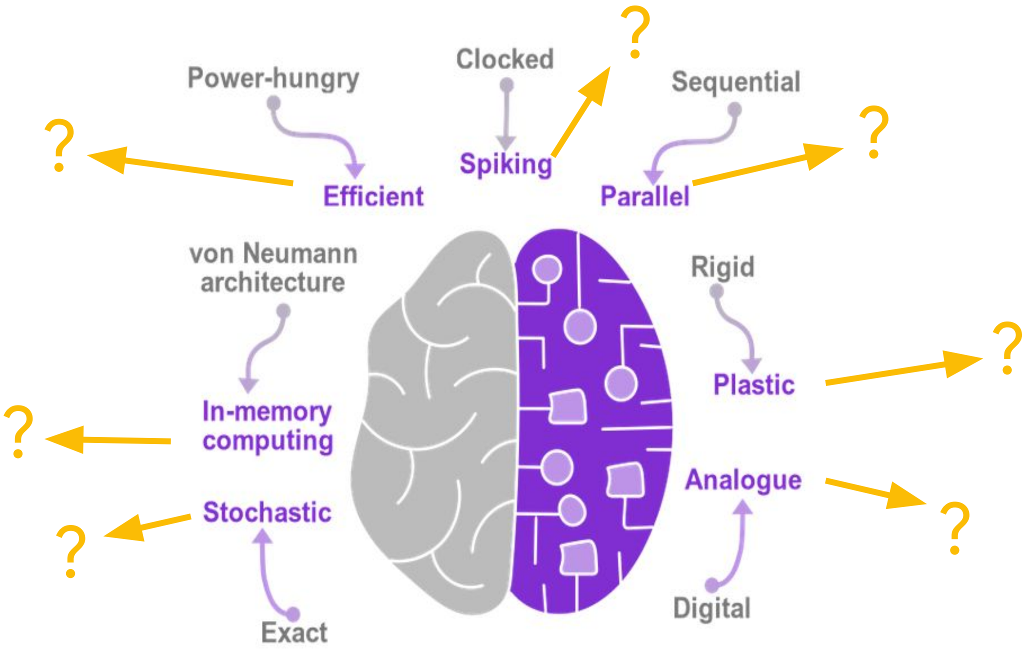
[Example from the NeuroTech EU Consortium]

What should be benchmarked as “neuromorphic”?

Necessitates *inclusive* benchmarks

- General tasks of interest

#1: Lack of a formal definition



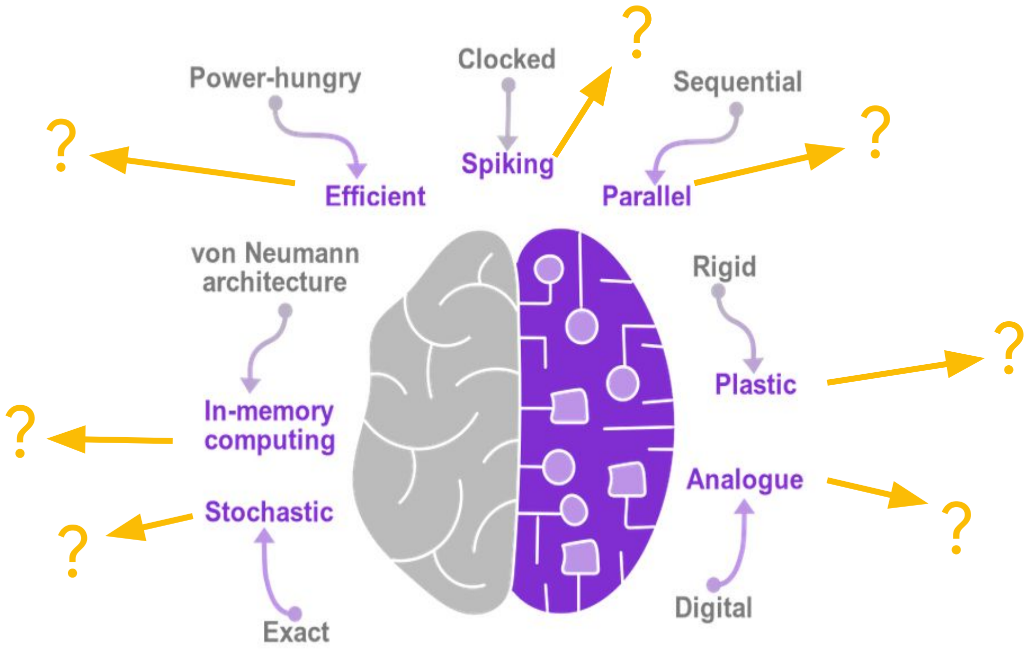
[Example from the NeuroTech EU Consortium]

What should be benchmarked as “neuromorphic”?

Necessitates *inclusive* benchmarks

- General tasks of interest
- General metrics

#1: Lack of a formal definition



[Example from the NeuroTech EU Consortium]

What should be benchmarked as “neuromorphic”?

Necessitates *inclusive* benchmarks

- General tasks of interest
- General metrics
- Direct comparisons with conventional approaches

Challenges in Benchmarking Neuromorphics

#1

Lack of a
formal definition

inclusive

#2

Implementation
diversity

#3

Rapid
research evolution

#2: Implementation diversity

nest::

LAVA

BRIAN

snnTorch

Norse

rockpool

#2: Implementation diversity

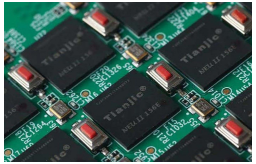
nest::
LAVA
BRIAN
snnTorch
Norse
rockpool

Software Frameworks:
Neuroscience simulation,
hardware interfacing,
automatic SNN configuration

#2: Implementation diversity

nest::
LAVA
BRIAN
snnTorch
rockpool
Norse

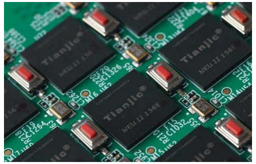
SpiNNaker



BrainScaleS

Software Frameworks:
Neuroscience simulation,
hardware interfacing,
automatic SNN configuration

#2: Implementation diversity



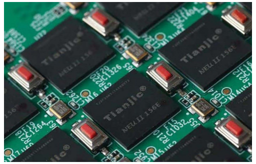
Software Frameworks:
Neuroscience simulation,
hardware interfacing,
automatic SNN configuration

Hardware platforms:
Circuitry, scale,
flexibility, programmability

#2: Implementation diversity

nest::
LAVA
snnTorch
BRIAN
rockpool
Norse

SpiNNaker



BrainScaleS

Software Frameworks:
Neuroscience simulation,
hardware interfacing,
automatic SNN configuration

Hardware platforms:
Circuitry, scale,
flexibility, programmability

Benchmarks require *actionable* portability and standardization

Challenges in Benchmarking Neuromorphics

#1

Lack of a
formal definition

inclusive

#2

Implementation
diversity

actionable

#3

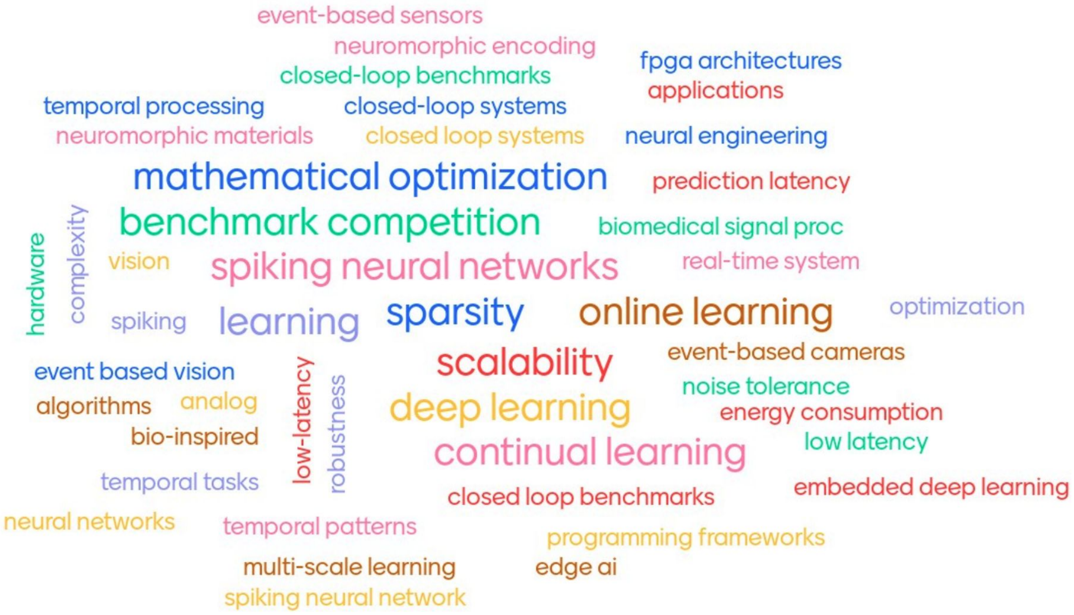
Rapid
research evolution

#3: Rapid research evolution

What topics should a neuromorphic benchmarking workshop include?

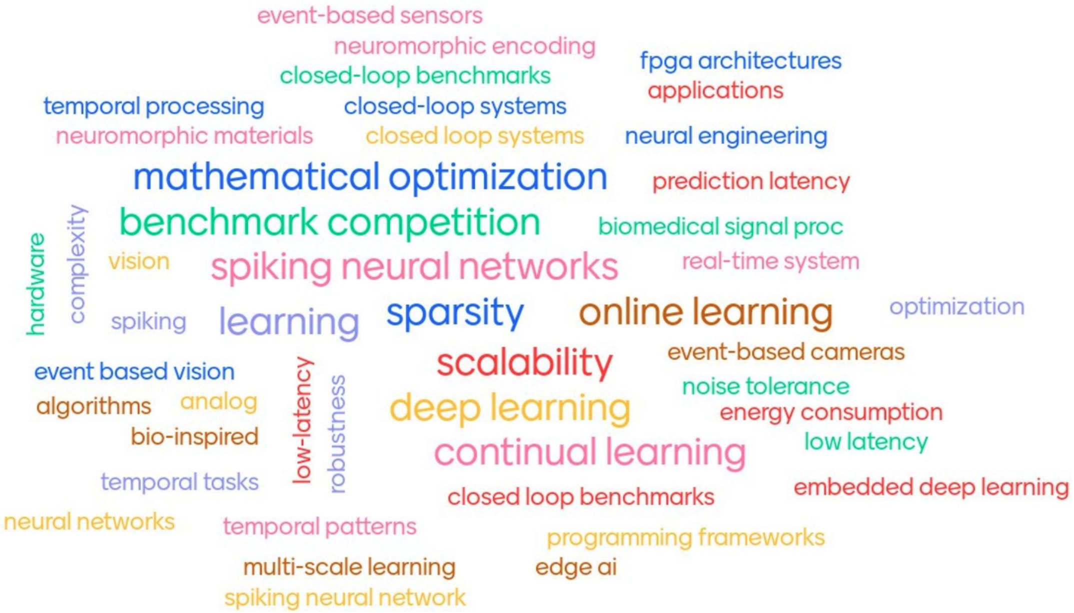
#3: Rapid research evolution

What topics should a neuromorphic benchmarking workshop include?



#3: Rapid research evolution

What topics should a neuromorphic benchmarking workshop include?



Benchmarks must be *iterative*

- Capture features of interest
- Evolve with the SOTA

spec

MLPerf

Challenges in Benchmarking Neuromorphics

#1	#2	#3
Lack of a formal definition	Implementation diversity	Rapid research evolution
inclusive	***actionable***	***iterative***

Challenges in Benchmarking Neuromorphics

#1	#2	#3
Lack of a formal definition	Implementation diversity	Rapid research evolution
inclusive	***actionable***	***iterative***

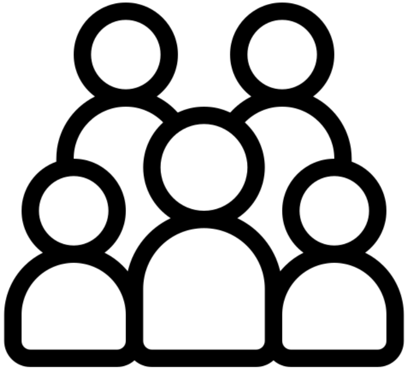
→ NeuroBench: A *framework* for benchmarking neuromorphics

NeuroBench

Goals: inclusive, actionable, iterative benchmarking

NeuroBench

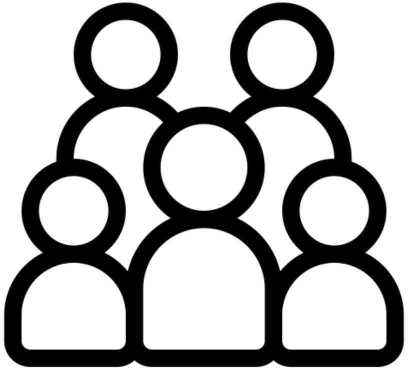
Goals: inclusive, actionable, iterative benchmarking



An open community of
neuromorphic researchers

NeuroBench

Goals: inclusive, actionable, iterative benchmarking



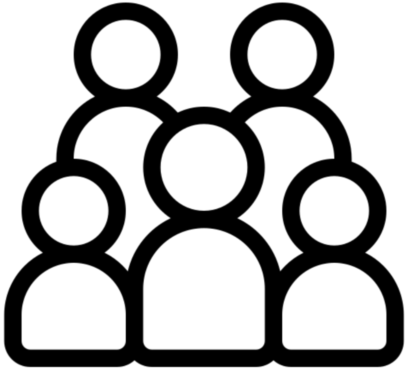
An open community of
neuromorphic researchers

What are the …

- Benchmarks of interest to drive research?

NeuroBench

Goals: inclusive, actionable, iterative benchmarking



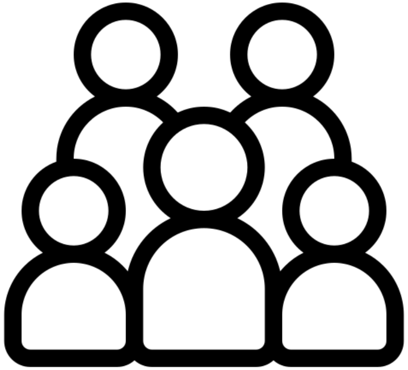
An open community of neuromorphic researchers

What are the …

- Benchmarks of interest to drive research?
- Common tools to be developed?

NeuroBench

Goals: inclusive, actionable, iterative benchmarking



An open community of
neuromorphic researchers

What are the …

- Benchmarks of interest to drive research?
- Common tools to be developed?
- Initial set of baseline approaches?

Dual-track structure

How to approach benchmarking algorithmic / deployed methods?

Dual-track structure

How to approach benchmarking algorithmic / deployed methods?

Algorithm track: System-independent complexity analysis

System track: For measuring methods deployed on hardware

Dual-track structure

How to approach benchmarking algorithmic / deployed methods?

Algorithm track: System-independent complexity analysis

- General metrics for model complexity, proxy hardware performance

System track: For measuring methods deployed on hardware

Dual-track structure

How to approach benchmarking algorithmic / deployed methods?

Algorithm track: System-independent complexity analysis

- General metrics for model complexity, proxy hardware performance
- Exploration and prototyping, without implementation to neuromorphic HW

System track: For measuring methods deployed on hardware

Dual-track structure

How to approach benchmarking algorithmic / deployed methods?

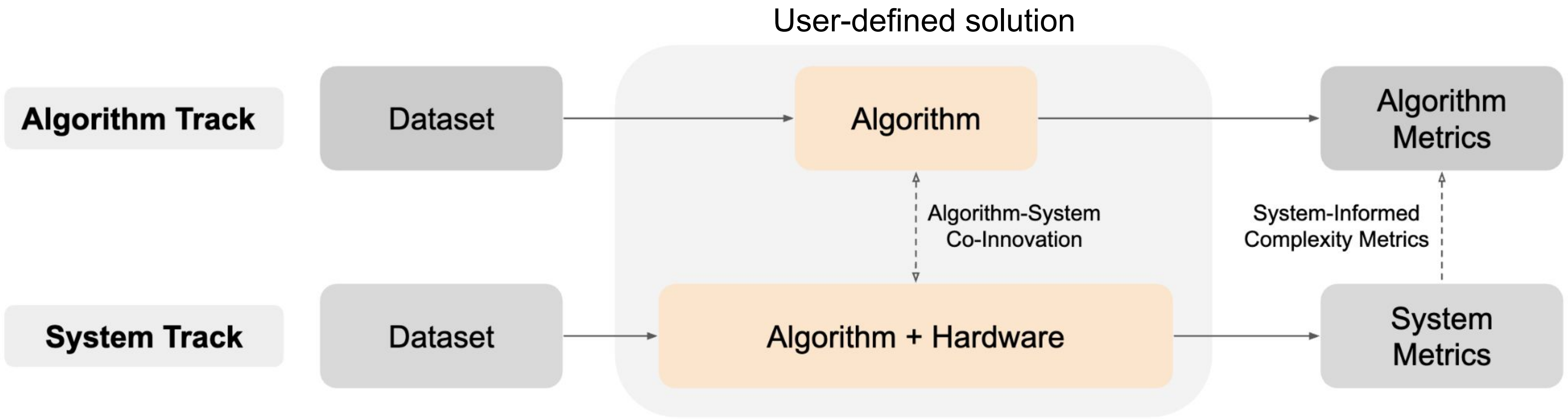
Algorithm track: System-independent complexity analysis

- General metrics for model complexity, proxy hardware performance
- Exploration and prototyping, without implementation to neuromorphic HW

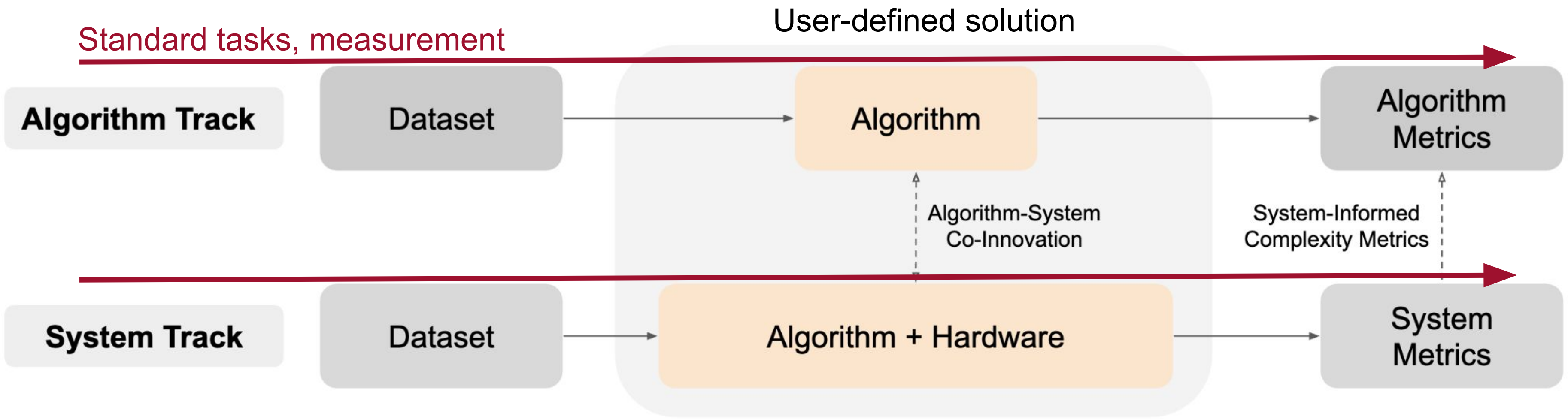
System track: For measuring methods deployed on hardware

- Evaluate deployed latency, throughput, energy efficiency

Dual-track structure

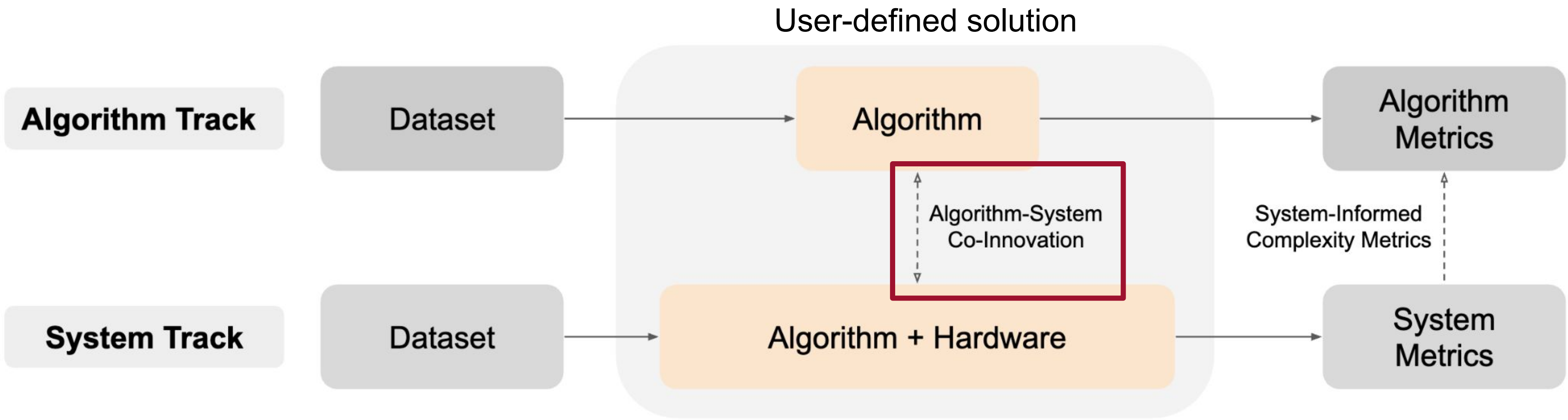


Dual-track structure



Dual-track structure

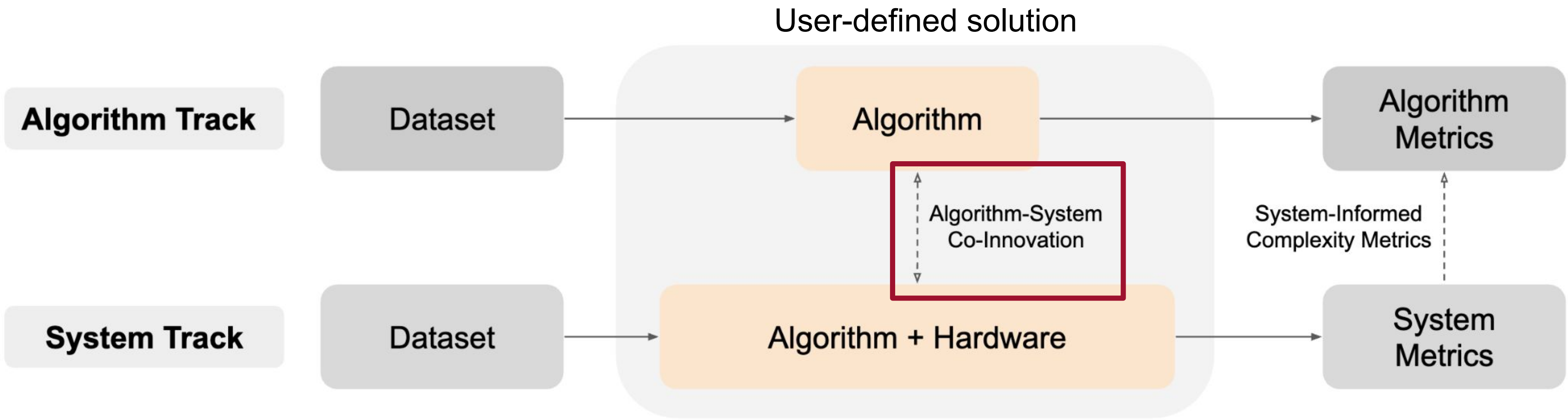
→ Highly-effective solutions can motivate future solutions in the other benchmark track



Dual-track structure

e.g., best-performing algorithms are target workloads for future hardware, hardware influences network topologies, etc.

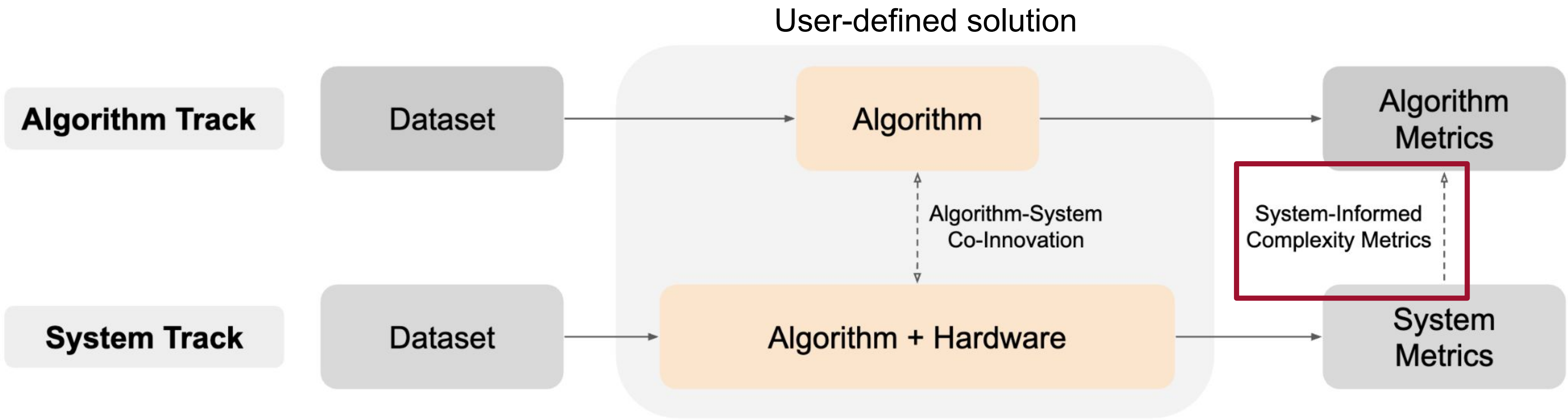
→ Highly-effective solutions can motivate future solutions in the other benchmark track



Dual-track structure

→ Highly-effective solutions can motivate future solutions in the other benchmark track

→ Deployed performance will inform hardware models of algorithm complexity metrics

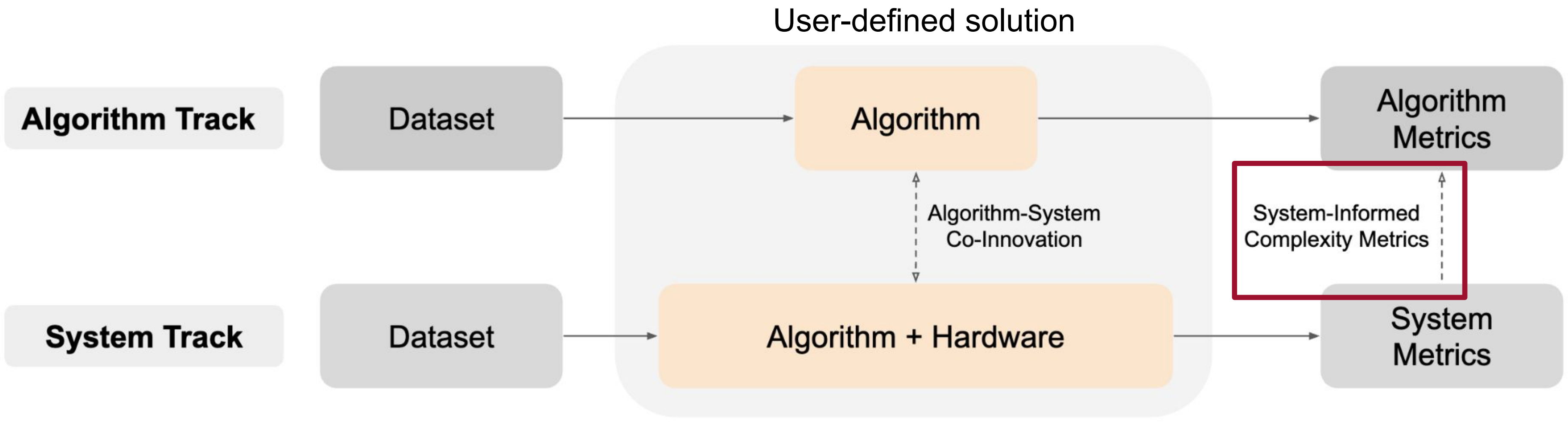


Dual-track structure

e.g., determine operation costs, compute / memory resource costs

→ Highly-effective solutions can motivate future solutions in the other benchmark track

→ Deployed performance will inform hardware models of algorithm complexity metrics



Dual-track structure

Algorithm track: System-independent complexity analysis

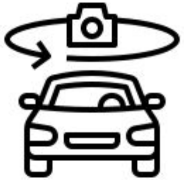
- General metrics for model complexity, proxy hardware performance
- Exploration and prototyping, without implementation to neuromorphic HW

System track: For measuring methods deployed on hardware

- Evaluate deployed latency, throughput, energy efficiency

Algorithm Track v1.0 Benchmarks

Keyword Few-shot, Continual Learning



Event Camera Object Detection

Primate Motor Prediction

Chaotic Function Prediction

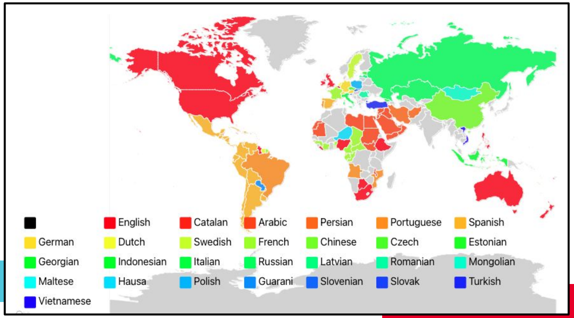
Keyword Few-shot Continual Learning

Application
Continual expansion of multilingual keyword dictionary using few training examples.

Dataset
Multilingual Spoken Word Corpus (MSWC) keyword dataset (50 languages, over 6000 hours).

Task
Model base-trains on 100 keywords across 6 languages. Then, it successively undergoes 10-way, 5-shot learning sessions of 100 total new keywords from 10 new languages.

Correctness
Classification accuracy is measured after each session, on all previously learned classes.



Event Camera Object Detection

Application

Real-time, energy-efficient / always-on automotive object detection, autonomous driving.

Dataset

Prophesee 1MP Gen 4 Automotive Detection dataset (14.65 hours, 3.5TB uncompressed).

Task

Detect car, two-wheeler, pedestrian. [train / val / test] split of [11.2 / 2.2 / 2.2] hours.

Correctness

COCO mean average precision (mAP).

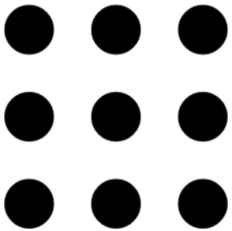
Primate Motor Decoding

Application
Sensorimotor biophysiological emulation, for prosthetics and brain-computer interfaces.

Dataset
Motor cortex recordings of two non-human primates engaged in reaching tasks (touch screen).

Task
Use cortical recording time-series to predict fingertip reach velocity in X and Y dimensions.

Correctness
R^2 of predicted velocities against ground truth.



Chaotic Function Prediction

Application

Dynamic time-series forecasting, (markets, climate, signals, etc.). Also a small dimensional problem useful for prototyping emerging resource-constrained hardware (i.e., mixed-signal).

Dataset

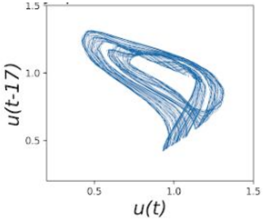
Mackey-Glass time series, one-dimensional non-linear time delay differential equation.

Task

Train using the first half of the generated time series, then autonomously forecast the second half.

Correctness

Symmetric mean absolute percentage error (sMAPE).



$$\frac{dx}{dt} = \beta \frac{x(t-\tau)}{1 + x(t-\tau)^n} - \gamma x(t).$$

Algorithm Track Metrics

Solution-agnostic metrics (primary)

- Correctness (defined per task)
- Complexity
  - General metrics which reflect the architectural cost of the algorithm

Solution-specific metrics can be added:

- Complexity of neuron dynamics
- Robustness to noise (e.g. for methods aimed towards analog hardware)

Algorithm Track Complexity Metrics

Footprint

Memory usage accounting for quantization, parameters and buffer requirements.

Connection Sparsity

Sparsity of model synaptic connections. Accounts for sparse initialization and pruning.

Activation Sparsity

Sparsity of neuron activations during execution. Insight into deployed communication requirement.

Algorithm Track Complexity Metrics

Footprint, Connection Sparsity, Activation Sparsity

Synaptic Operations

Number of synaptic operations per prediction.

→ Dense SynOps account for all operations.

→ Effective SynOps count only non-zero operations.

→ Multiply-Accumulates (MACs) for valued activations and Accumulates (ACs) for binary spikes.

Model Execution Rate*

Throughput of model output, reflects responsiveness and deployed compute requirement.

→ Critical algorithmic feature, not necessarily a metric to be calculated.

Benchmark case studies

Algorithmic baselines

Keyword FSCIL - Base Metrics

Baseline	Accuracy (Base / Session Avg)	Footprint (bytes)	Model Exec. Rate (Hz)	Connection Sparsity	Activation Sparsity	SynOps (per model exec.)		
						Dense	Eff_MACs	Eff_ACs
M5 ANN	(97.09% / 89.27%)	6.03×10^6	1	0.0	0.783	2.59×10^7	7.85×10^6	0
SNN	(93.48% / 75.27%)	1.36×10^7	200	0.0	0.916	3.39×10^6	0	3.65×10^5

Keyword FSCIL - Base Metrics

Baseline	Accuracy (Base / Session Avg)	Footprint (bytes)	Model Exec. Rate (Hz)	Connection Sparsity	Activation Sparsity	SynOps (per model exec.)		
						Dense	Eff_MACs	Eff_ACs
M5 ANN	(97.09% / 89.27%)	6.03×10^6	1	0.0	0.783	2.59×10^7	7.85×10^6	0
SNN	(93.48% / 75.27%)	1.36×10^7	200	0.0	0.916	3.39×10^6	0	3.65×10^5

- Footprint vs Dense SynOps
- Model execution rate: real-time data processing
- Activation sparsity of SNN baseline

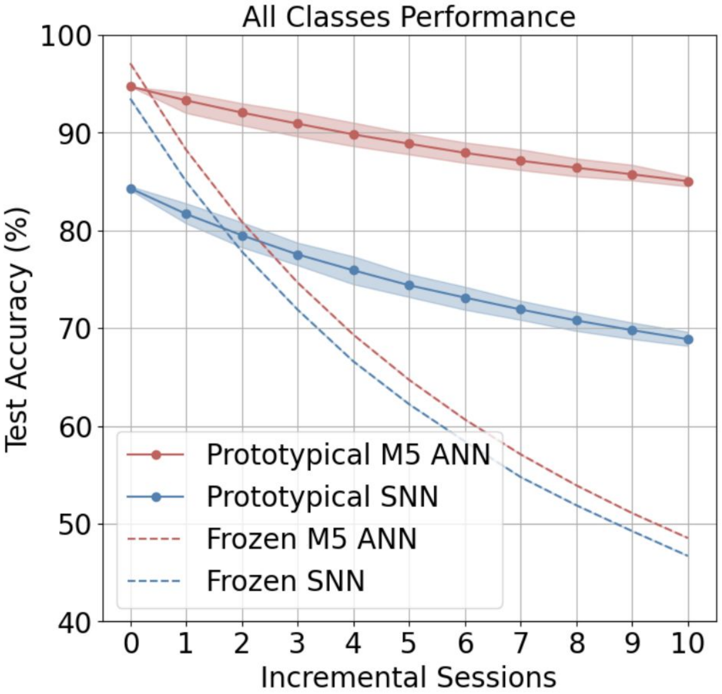
Keyword FSCIL - Continual Learning

Prototypical Networks

Using pre-trained feature extractor, predicted class is based on distance to mean feature embeddings.

Large initial drop for SNN

→ Opportunities towards general SNN feature extraction.



Event Camera Object Detection

Baseline	mAP	Footprint (bytes)	Model Exec. Rate (Hz)	Connection Sparsity	Activation Sparsity	SynOps (per model exec.)		
						Dense	Eff_MACs	Eff_ACs
RED ANN	0.429	9.13×10^7	20	0.0	0.634	2.84×10^{11}	2.48×10^{11}	0
Hybrid	0.271	1.21×10^7	20	0.0	0.613	9.85×10^{10}	3.76×10^{10}	5.60×10^8

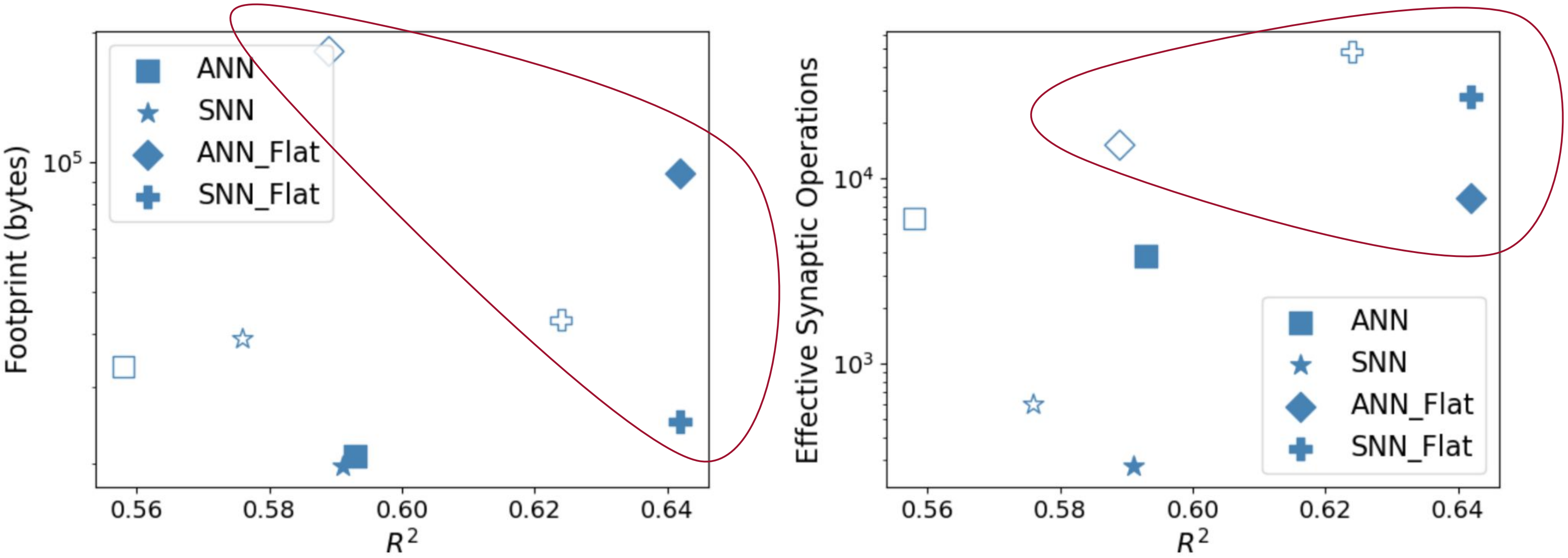
- Footprint: not much different
- Activation sparsity
- Ratio of effective SynOps for Hybrid model

Primate Motor Decoding

Baseline	R^2	Footprint (bytes)	Model Exec. Rate (Hz)	Connection Sparsity	Activation Sparsity	SynOps (per model exec.)		
						Dense	Eff_MACs	Eff_ACs
ANN	0.593	20824	250	0.0	0.683	4704	3836	0
	0.558	33496	250	0.0	0.668	7776	6103	0
SNN	0.593	19648	250	0.0	0.997	4900	0	276
	0.568	38848	250	0.0	0.999	9700	0	551

- Activation sparsity (!)
- Effective SynOps

Correctness-Complexity Trade-off

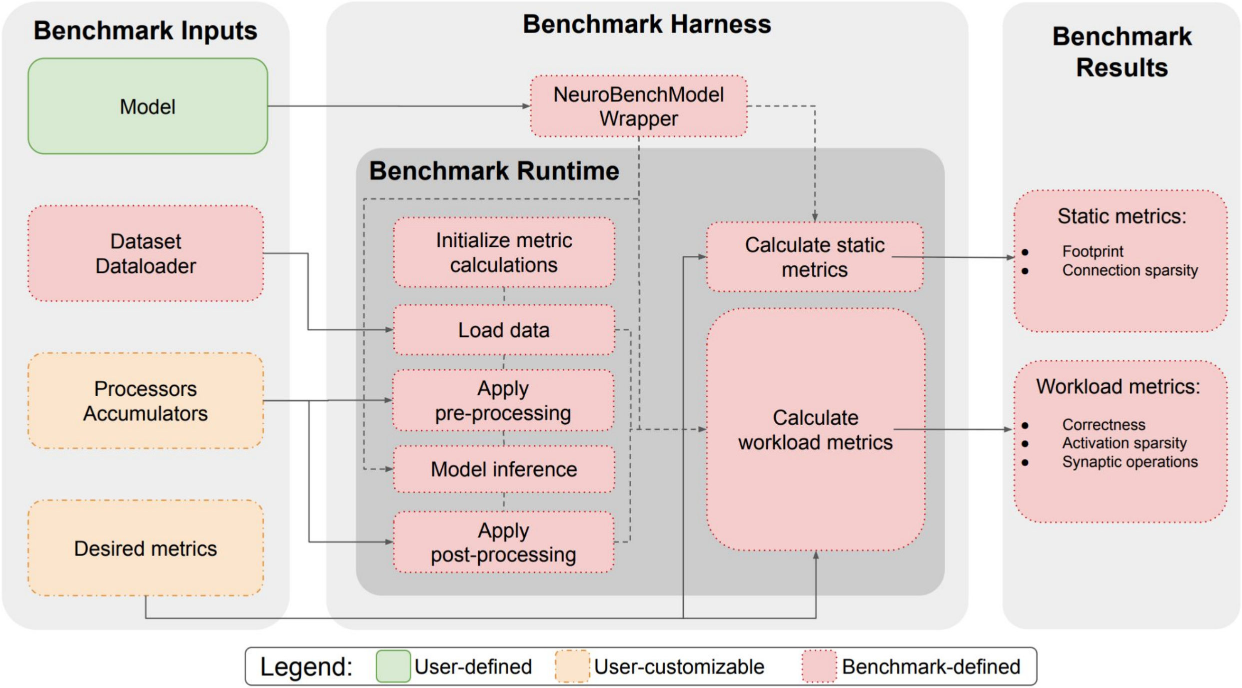


Chaotic Function Prediction

Baseline	sMAPE	Footprint (bytes)	Model Exec. Rate (Hz)	Connection Sparsity	Activation Sparsity	SynOps (per model exec.)		
						Dense	Eff_MACs	Eff_ACs
ESN	14.79	2.81×10^5	-	0.876	0.0	3.52×10^4	4.37×10^3	0
LSTM	13.37	4.90×10^5	-	0.0	0.530	6.03×10^4	6.03×10^4	0

- Reservoir computing solution
- Connection sparsity

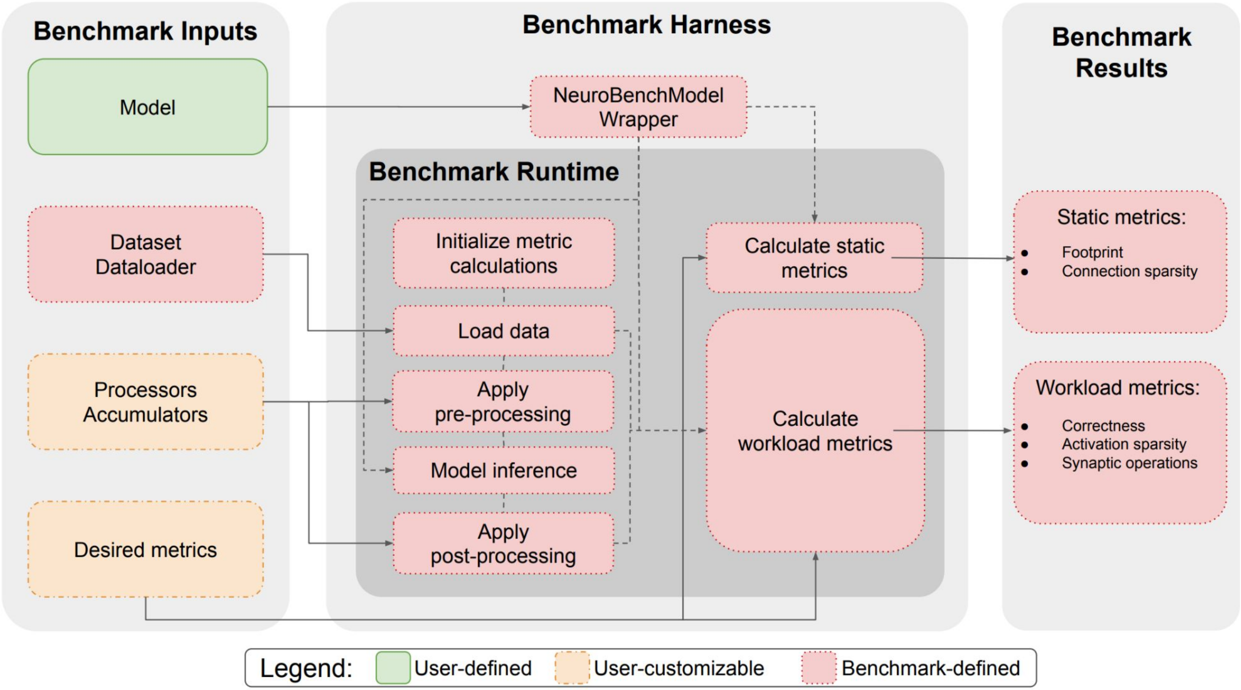
Algorithm Track Harness



Extensible to new tasks, metrics, frameworks.

Modular.

Designed for easy development.



Using the harness

```
pip install neurobench
```

1. Install the harness
2. Pick the dataset
3. Train the model
4. Define preprocessors, postprocessors, and metrics
5. Run the benchmark

```python
import torch
from torch.utils.data import DataLoader

from neurobench.datasets import SpeechCommands
from neurobench.preprocessing import S2SPreProcessor
from neurobench.postprocessing import choose_max_count
from neurobench.models import SNNTorchModel
from neurobench.benchmarks import Benchmark

from SNN import net

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

test_set = SpeechCommands(path="../../../data/speech_commands/", subset="testing")
test_set_loader = DataLoader(test_set, batch_size=500, shuffle=True)

net.load_state_dict(torch.load("./model_data/s2s_gsc_snntorch", map_location=device))
model = SNNTorchModel(net)

preprocessors = [S2SPreProcessor(device=device)]
postprocessors = [choose_max_count]

static_metrics = ["footprint", "connection_sparsity"]
workload_metrics = ["classification_accuracy", "activation_sparsity", "synaptic_operations"]

benchmark = Benchmark(model, test_set_loader, preprocessors, postprocessors, [static_metrics, workload_metrics])
results = benchmark.run(device=device)
print(results)
```

Simplicity/Modularity - SynOps

Example:

We calculate synaptic operations exactly, so it can be slow

```
workload_metrics = ["classification_accuracy", "activation_sparsity", "synaptic_operations"]
```

```
gsc(main)$ python benchmark_snn.py
Running benchmark
 39%|█████████████████                                    | 9/23 [00:19<00:30,  2.17s/it]
```

```
workload_metrics = ["classification_accuracy"]
```

```
gsc(main*)$ python benchmark_snn.py
Running benchmark
 43%|███████████████████                                  | 10/23 [00:08<00:10,  1.18it/s]
```

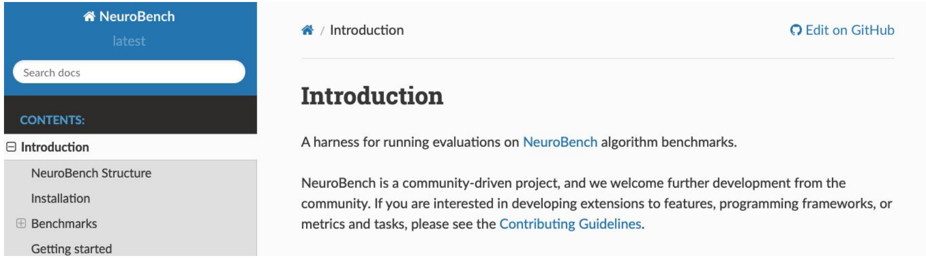
Open-Source

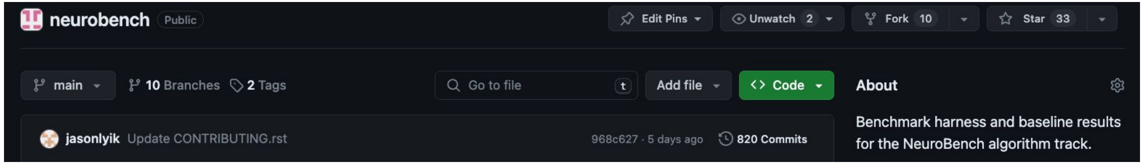
Tutorial notebooks, baseline reproduction scripts, documentation

https://pypi.org/project/neurobench/

https://neurobench.readthedocs.io/en/latest/readme.html

https://github.com/NeuroBench/neurobench





Dual-track structure

Algorithm track: System-independent complexity analysis

- General metrics for model complexity, proxy hardware performance
- Exploration and prototyping, without implementation to neuromorphic HW

System track: For measuring methods deployed on hardware

- Evaluate deployed latency, throughput, energy efficiency

Dual-track structure

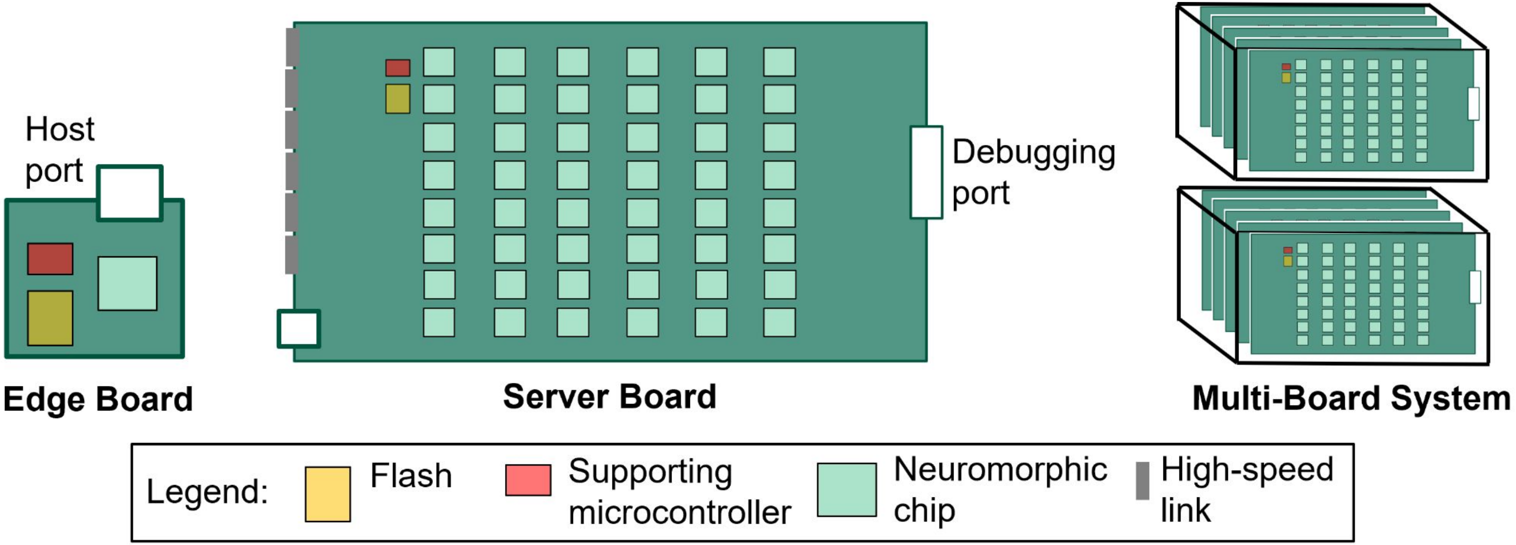
Algorithm track: System-independent complexity analysis

- General metrics for model complexity, proxy hardware performance
- Exploration and prototyping, without implementation to neuromorphic HW

System track: For measuring methods deployed on hardware

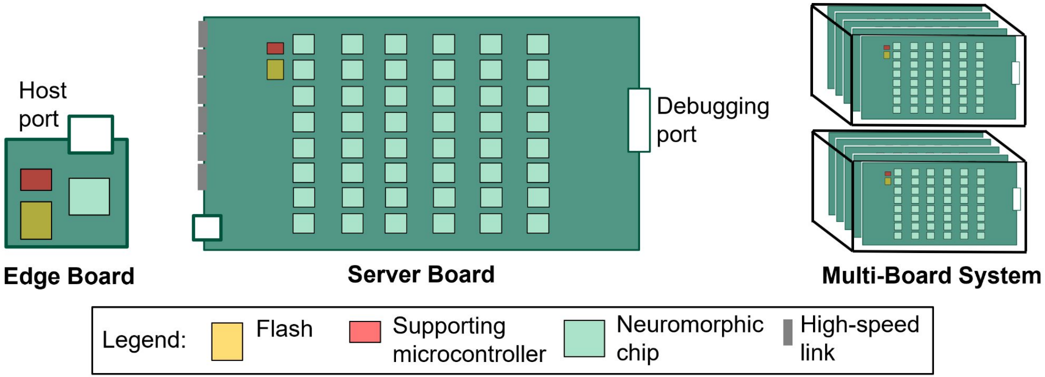
- Evaluate deployed latency, throughput, energy efficiency

Neuromorphic Systems



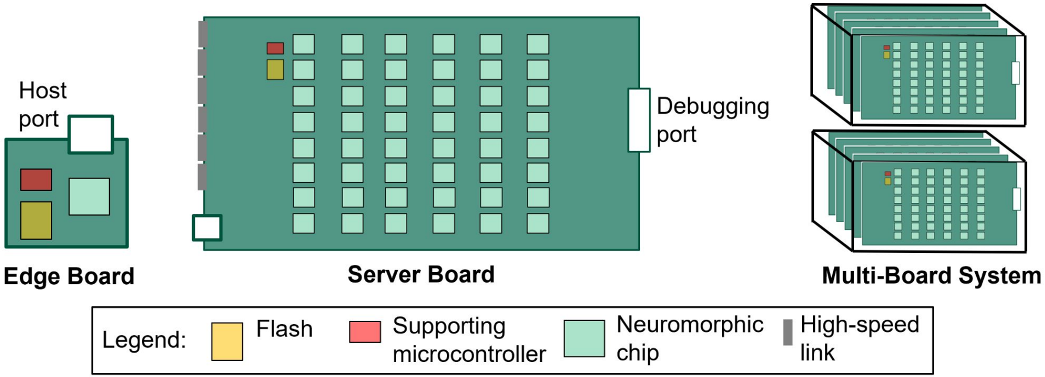
Neuromorphic System Variation

- Scale: mW to kW machines



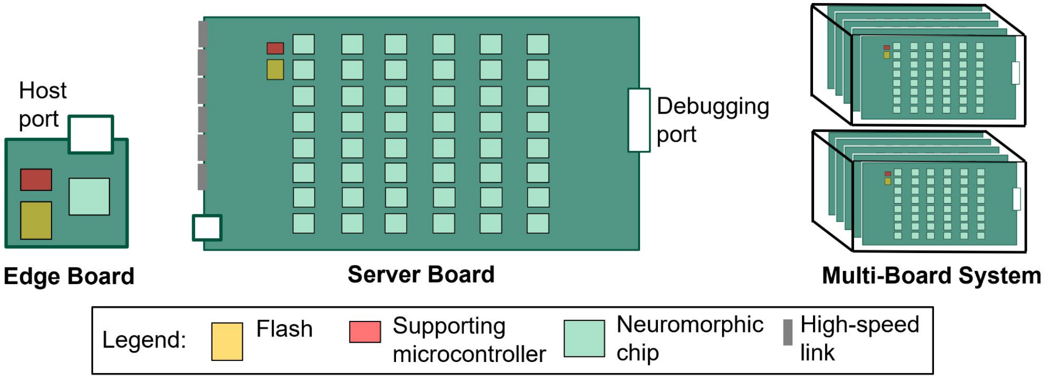
Neuromorphic System Variation

- Scale: mW to kW machines
- Form factor: standalone and accelerator boards



Neuromorphic System Variation

- Scale: mW to kW machines
- Form factor: standalone and accelerator boards
- Maturity: development, prototype, commercial



System Track Goals

Shared guidelines across wide system variation

System Track Goals

Shared guidelines across wide system variation

- Application-level benchmarks

System Track Goals

Shared guidelines across wide system variation

- Application-level benchmarks
- Performance and efficiency measurements

System Track Goals

Shared guidelines across wide system variation

- Application-level benchmarks
- Performance and efficiency measurements
- Open algorithm and pre-/post-processing
  - Algorithm can be tailored to hardware features

System Track Goals

Shared guidelines across wide system variation

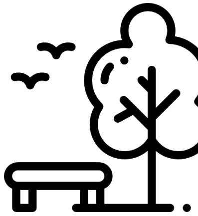
- Application-level benchmarks
- Performance and efficiency measurements
- Open algorithm and pre-/post-processing
- Component-granularity results
    - e.g. host CPU, pre-/post-processing units, memory, etc.

System Track Goals

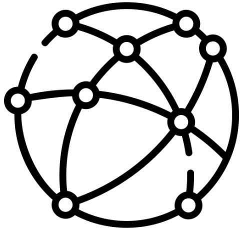
Shared guidelines across wide system variation

- Application-level benchmarks
- Performance and efficiency measurements
- Open algorithm and pre-/post-processing
- Component-granularity results
- Transparency then consistency
    - Enable intuitive analysis of widely varying systems

System Track v1.0 Benchmarks



Acoustic Scene Classification



Quadratic Unconstrained Binary Optimization

Acoustic Scene Classification

Application

Always-on audio smart sensor for various environments, mW power range.

Dataset

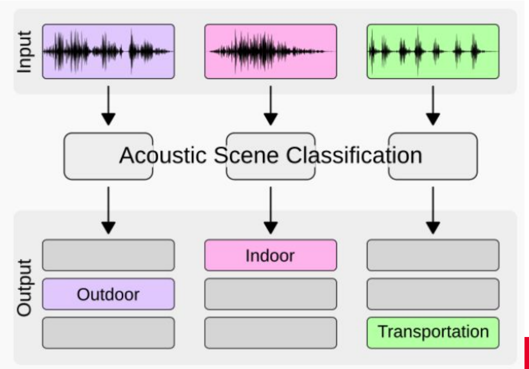
TAU Urban Acoustic Scenes (DCASE 2020 challenge).

Benchmark scenario

Single stream, batch-size 1 processing.

Metrics

Classification accuracy, latency per sample, energy per sample.



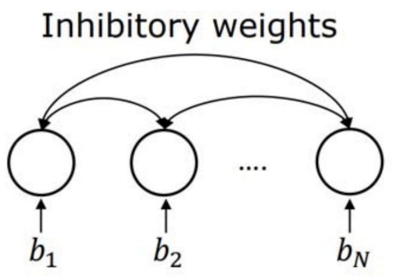
QUBO (Max Independent Set)

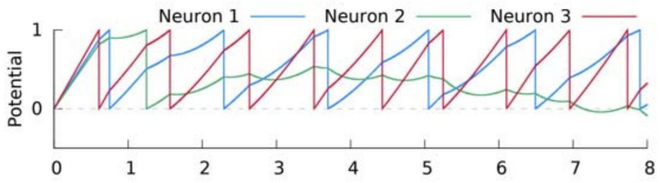
Application
Scalable optimization for finance, routing, scheduling in mW to kW range.

Dataset
Synthetic graphs of various sizes and connectivity.

Benchmark scenario
Optimization, the solution improves over time.

Metrics
Most optimal solution found, latency and energy to reach various threshold optimalities.





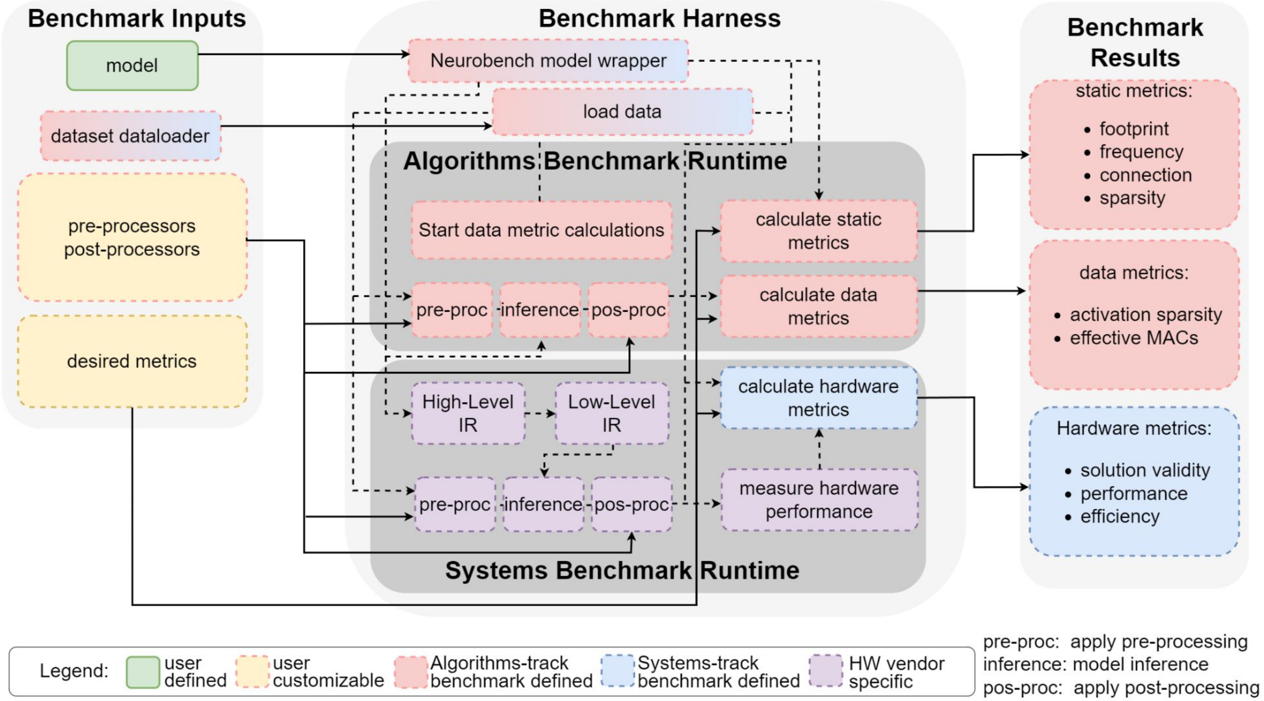
Davies et al., 2018: https://redwood.berkeley.edu/wp-content/uploads/2021/08/Davies2018.pdf

System Track v1.0: Ongoing!

→ Transparent, rigorous comparisons of mature, optimized neuromorphic hardware systems on identical tasks.

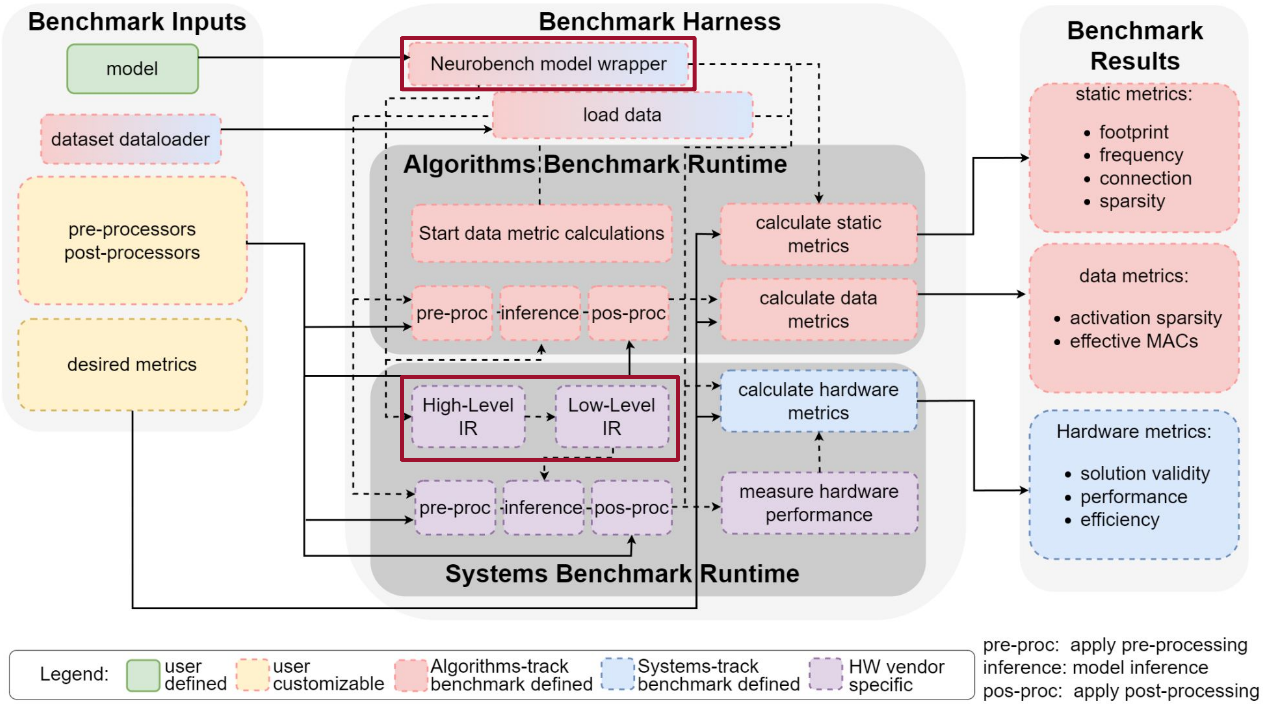
→ Provide the foundation for understanding and measuring further neuromorphic systems.

System track benchmark tooling



* System track tools (blue/purple) currently under development.

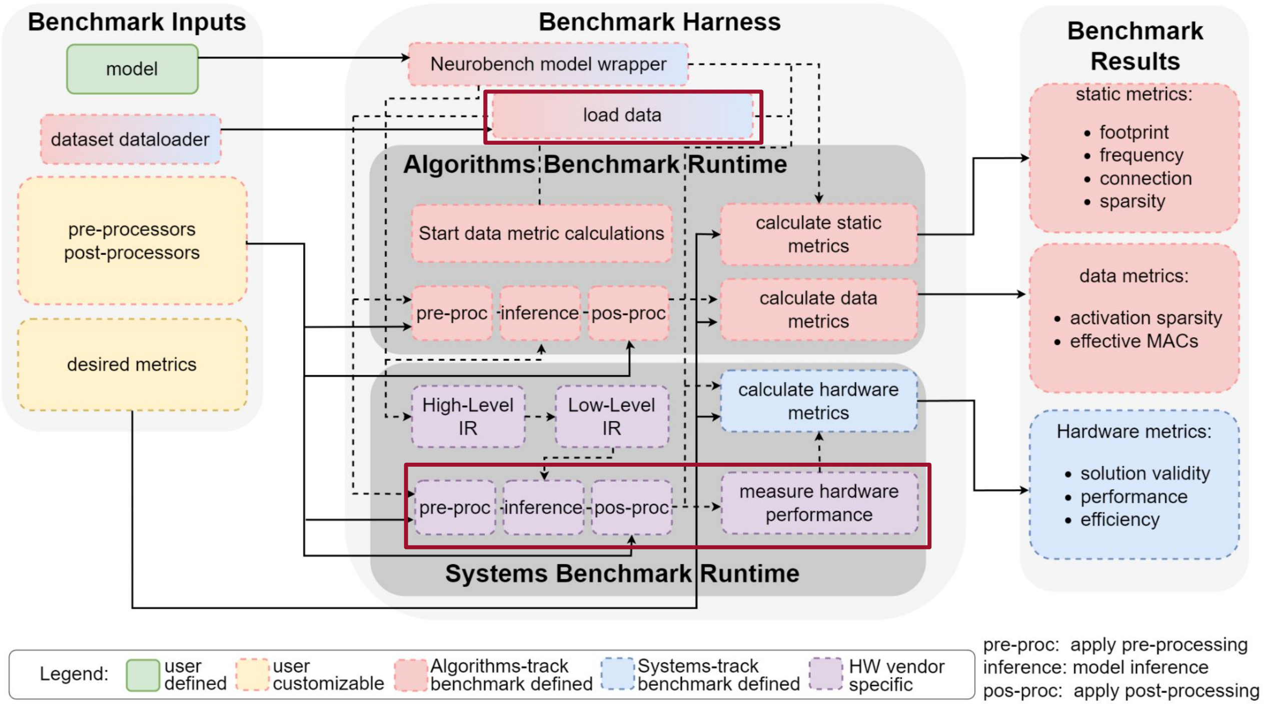
System track benchmark tooling



* System track tools (blue/purple) currently under development.

1. Compiling + Mapping

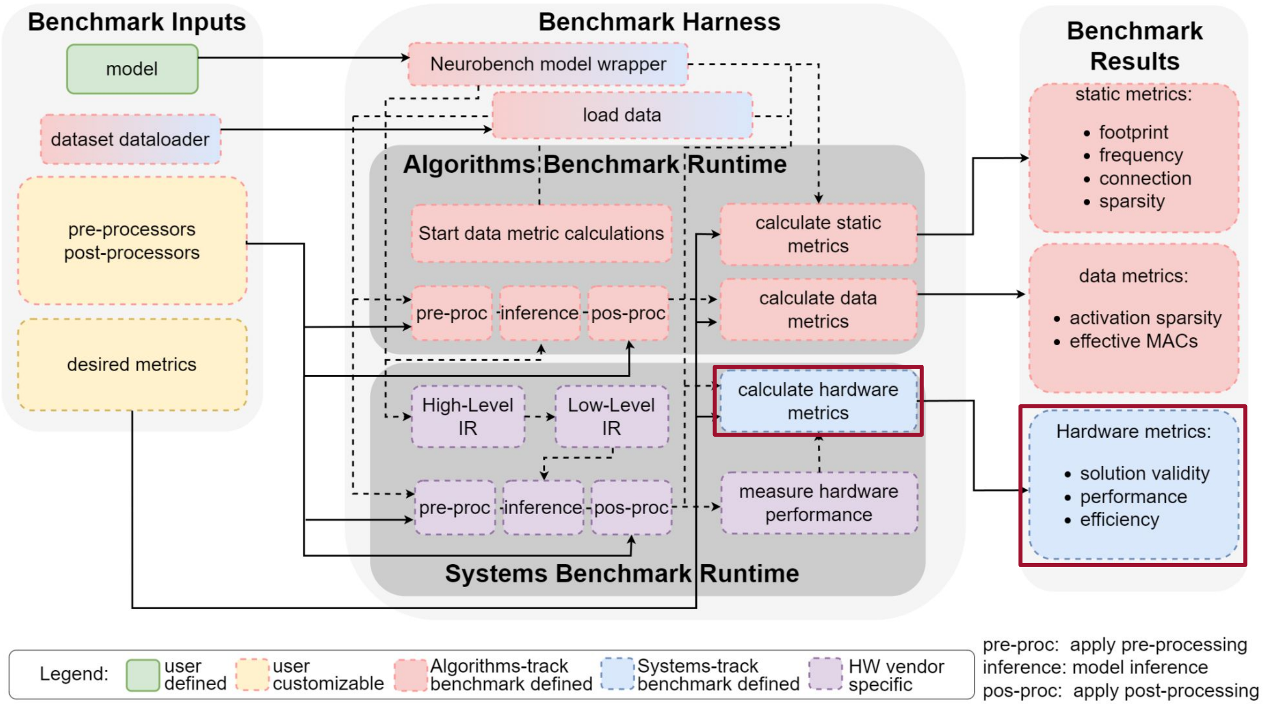
System track benchmark tooling



* System track tools (blue/purple) currently under development.

1. Compiling + Mapping
2. Execution + Measurement

System track benchmark tooling

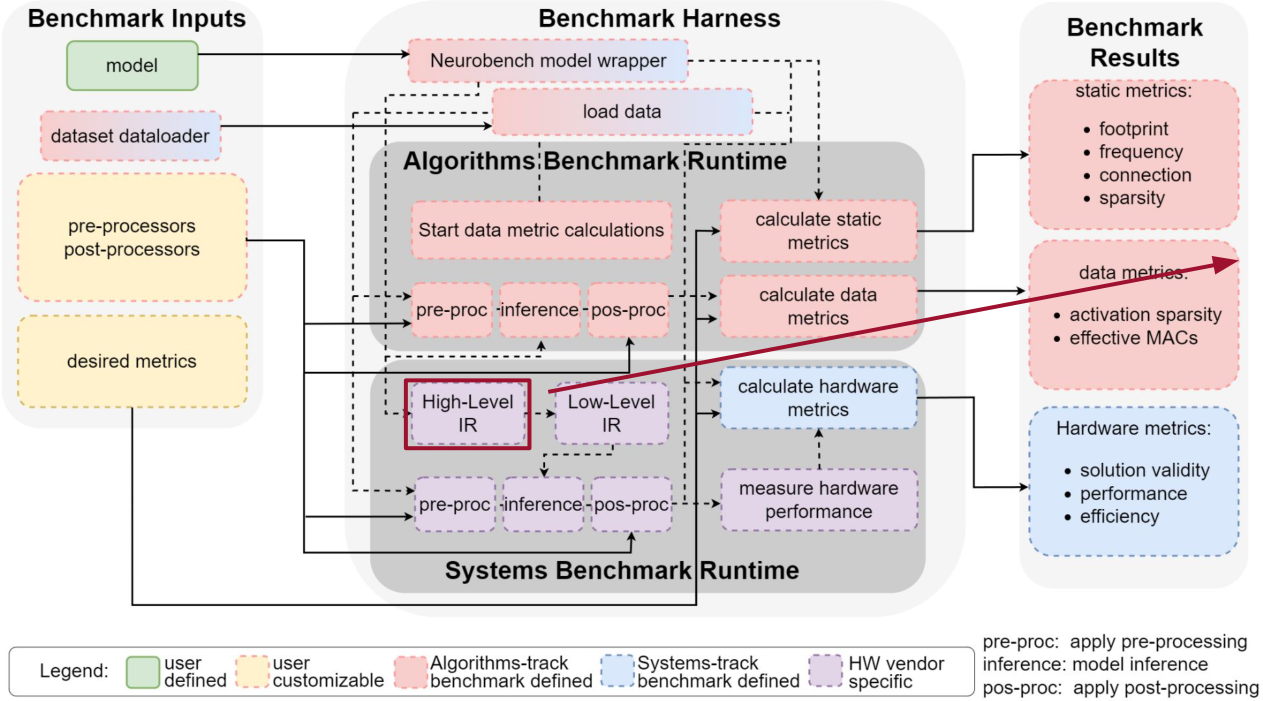


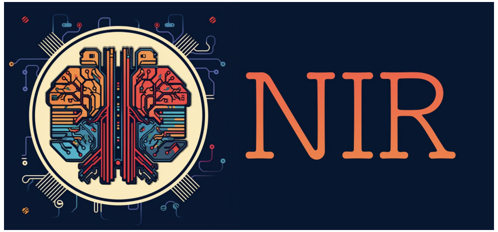
pre-proc: apply pre-processing
inference: model inference
pos-proc: apply post-processing

* System track tools (blue/purple) currently under development.

1. Compiling + Mapping
2. Execution + Measurement
3. Report results

Integration with other tools





Neuromorphic Intermediate Representation

https://github.com/neuromorphs/NIR

Ongoing Work

- Repository active maintenance
- System track v1.0 results
- Closed-loop extension, classic RL and neural decoding tasks
- BioCAS 2024 motor decoding challenge
- Common leaderboards
- Telluride talk + tutorial

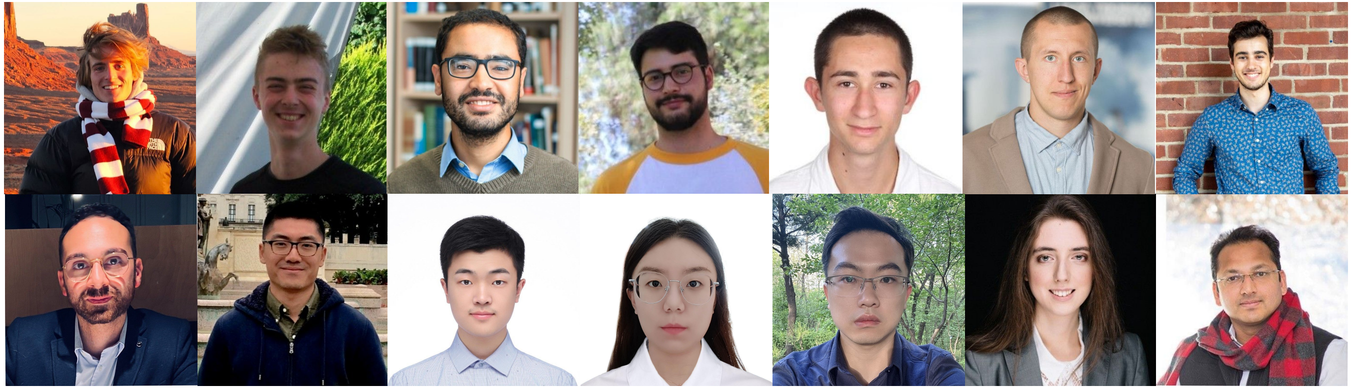
Other Future Work

- Datasets, e.g. integration with Tonic library
- Neuron dynamics metrics
- Continuous-time benchmarking
- Open-source hardware benchmark infrastructure

Acknowledgement

NeuroBench: A Framework for Benchmarking Neuromorphic Computing Algorithms and Systems

Jason Yik, Korneel Van den Berghe, Douwe den Blanken, Younes Bouhadjar, Maxime Fabre, Paul Hueber, Denis Kleyko, Noah Pacik-Nelson, Pao-Sheng Vincent Sun, Guangzhi Tang, Shenqi Wang, Biyan Zhou, Soikat Hasan Ahmed, George Vathakkattil Joseph, Benedetto Leto, Aurora Micheli, Anurag Kumar Mishra, Gregor Lenz, Tao Sun, Zergham Ahmed, Mahmoud Akl, Brian Anderson, Andreas G. Andreou, Chiara Bartolozzi, Arindam Basu, Petrut Bogdan, Sander Bohte, Sonia Buckley, Gert Cauwenberghs, Elisabetta Chicca, Federico Corradi, Guido de Croon, Andreea Danielescu, Anurag Daram, Mike Davies, Yigit Demirag, Jason Eshraghian, Tobias Fischer, Jeremy Forest, Vittorio Fra, Steve Furber, P. Michael Furlong, William Gilpin, Aditya Gilra, Hector A. Gonzalez, Giacomo Indiveri, Siddharth Joshi, Vedant Karia, Lyes Khacef, James C. Knight, Laura Kriener, Rajkumar Kubendran, Dhireesha Kudithipudi, Yao-Hong Liu, Shih-Chii Liu, Haoyuan Ma, Rajit Manohar, Josep Maria Margarit-Taulé, Christian Mayr, Konstantinos Michmizos, Dylan Muir, Emre Neftci, Thomas Nowotny, Fabrizio Ottati, Ayca Ozcelikkale, Priyadarshini Panda, Jongkil Park, Melika Payvand, Christian Pehle, Mihai A. Petrovici, Alessandro Pierro, Christoph Posch, Alpha Renner, Yulia Sandamirskaya, Clemens JS Schaefer, André van Schaik, Johannes Schemmel, Samuel Schmidgall, Catherine Schuman, Jae-sun Seo, Sadique Sheik, Sumit Bam Shrestha, Manolis Sifalakis, Amos Sironi, Matthew Stewart, Kenneth Stewart, Terrence C. Stewart, Philipp Stratmann, Jonathan Timcheck, Nergis Tömen, Gianvito Urgese, Marian Verhelst, Craig M. Vineyard, Bernhard Vogginger, Amirreza Yousefzadeh, Fatima Tuz Zohora, Charlotte Frenkel, Vijay Janapa Reddi



NeuroBench: Summary

- Algorithm and system benchmarking framework
- Novel benchmark tasks
- Extensible open-source benchmark platform

Listen in on NICE 2025

- *We demonstrate top-of-the-leaderboard accuracy / SynOps on <task>.*
- *We combine our method with <project> and show an improvement of …*
- *We open-source this new task through the NeuroBench harness.*
- *Compared to <project> from last year's NICE, we have improved by …*

Call to Action

NeuroBench is a community-driven benchmark framework.

v1.0 is ready and still being actively extended.

Engage with the project!

neurobench.ai