Xavier Servot

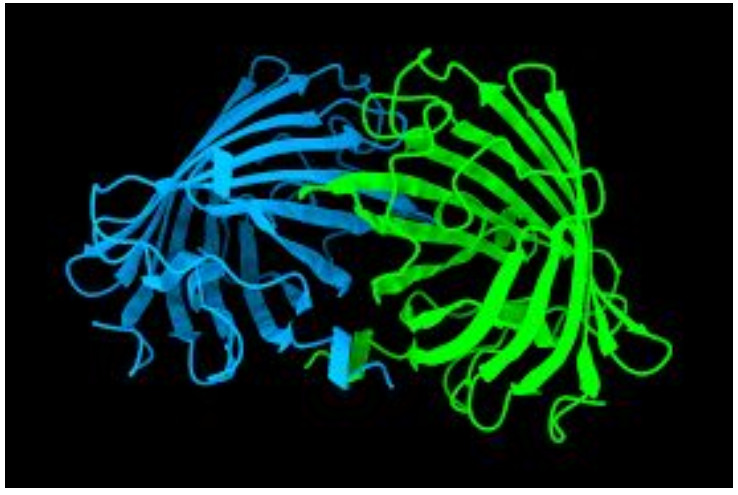
A Thousand Brains on a Thousand Chips

Transformers have **taken over the world of AI** in the past few years

There has been an **explosion** of the **domain of applicability**

***AlphaGeometry*: Gold** medal at the **International Mathematical Olympiads**

***AlphaFold*:** Solving **Protein Folding** and **Biomolecular interactions**



Despite these advances, **current leading artificial intelligence** lack **core attributes** of **biological intelligence**

Rapid learning from limited data[1,2]

[1] Lake, B., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. Science, 350(6266)
[2] Lake, Brenden M., et al. "Building machines that learn and think like people." Behavioral and brain sciences 40 (2017): e253..

Continuous adaptation to new situations[1,2]

[1] Flesch, T., Balaguer, J., Dekker, R., Nili, H., & Summerfield, C. (2018). Comparing continual task learning in minds and machines. Proceedings of the National Academy of Sciences of the USA, 115(44).
[2] McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. Psychology of Learning and Motivation Advances in Research and Theory

Learning while in deployment[1,2,3]

[1] Marshall, James AR, and Andrew B. Barron. "Are transformers truly foundational for robotics?." npj Robotics 3.1 (2025): 9.
[2] Billard, Aude, et al. "A roadmap for AI in robotics." Nature Machine Intelligence (2025): 1-7.
[3] Sünderhauf, Niko, et al. "The limits and potentials of deep learning for robotics." The International journal of robotics research 37.4-5 (2018): 405-420.

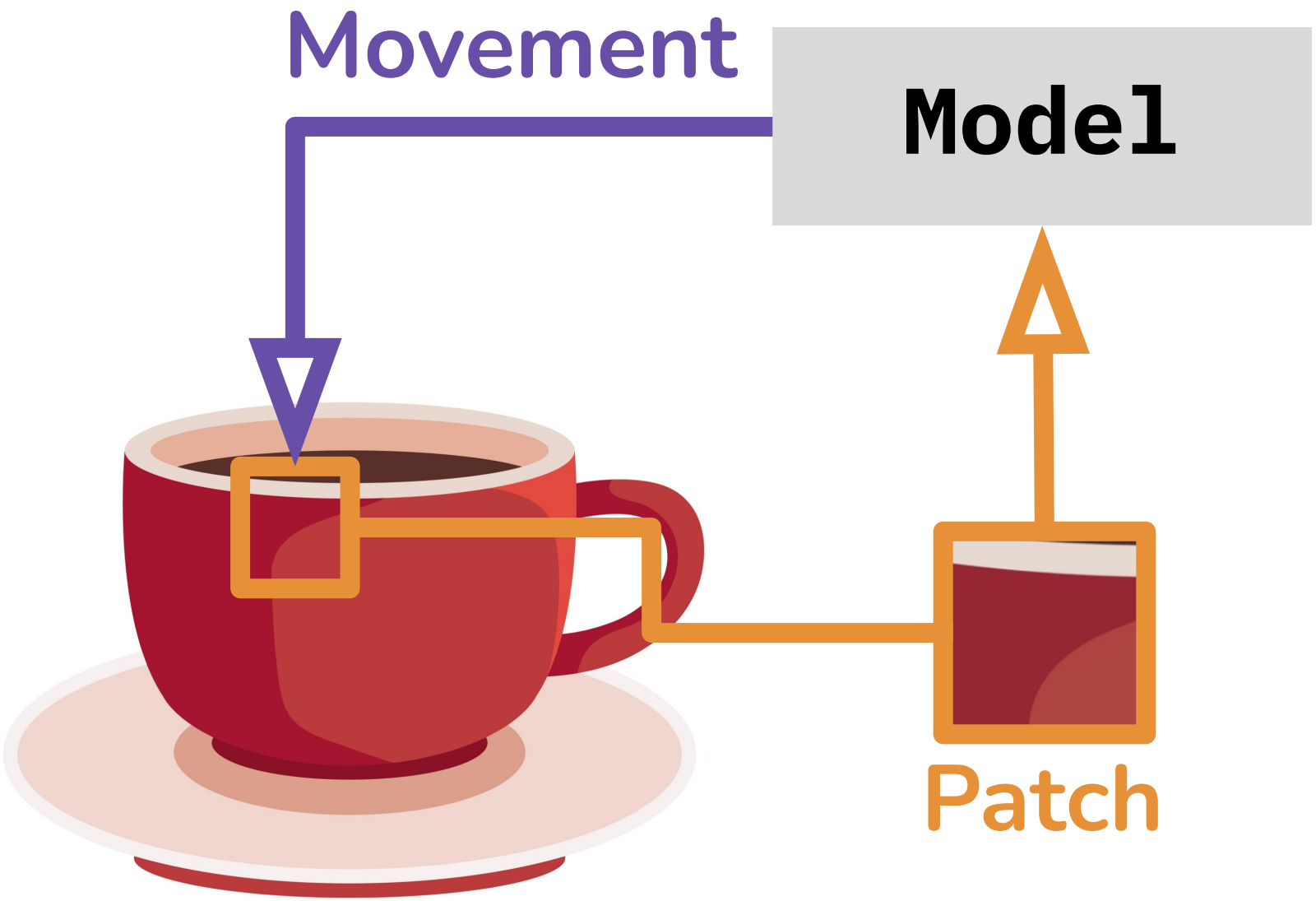
Compute and energy efficient learning[1,2,3]

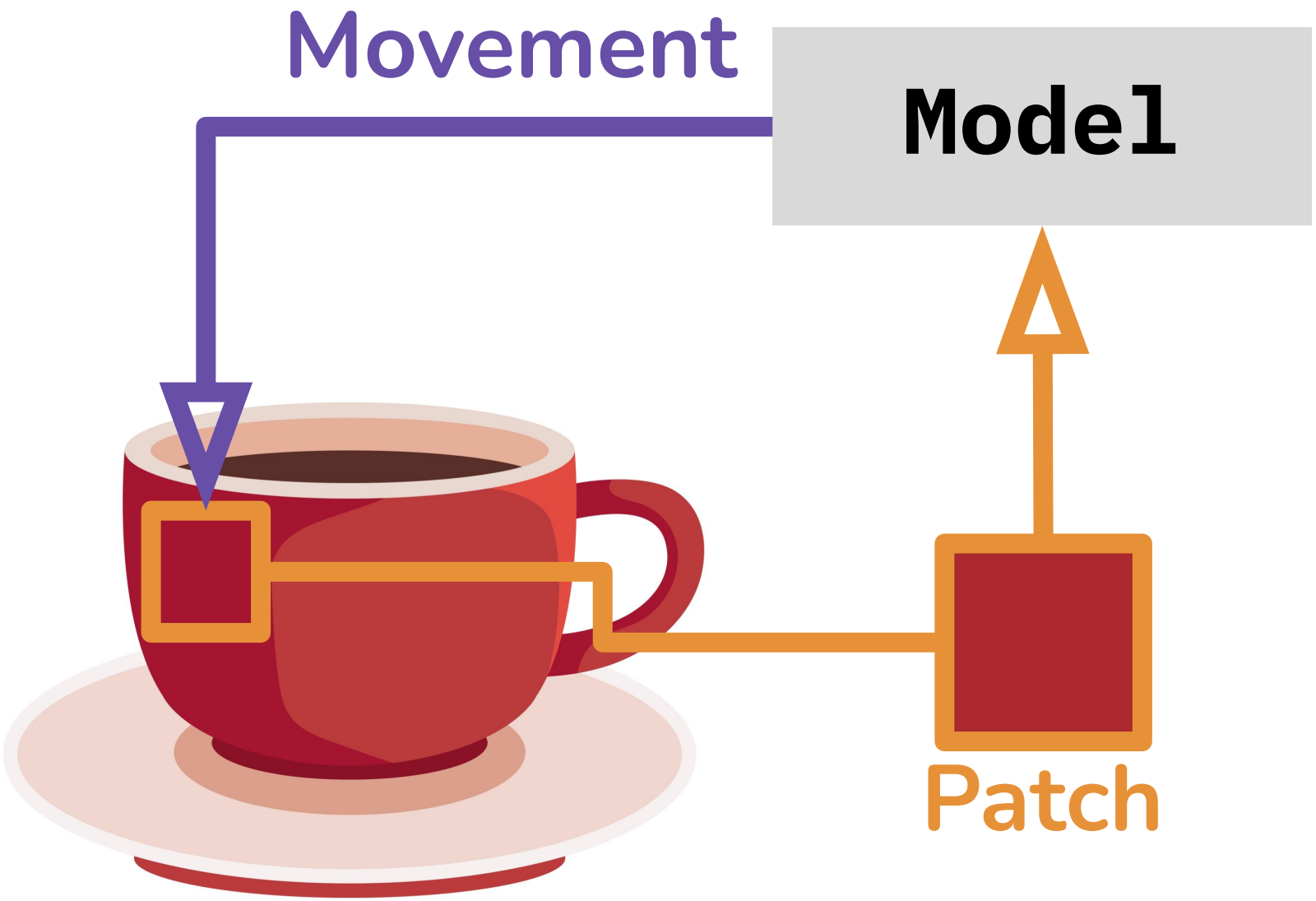
[1] Raichle, Marcus E., and Debra A. Gusnard. "Appraising the brain's energy budget." Proceedings of the National Academy of Sciences 99.16 (2002): 10237-10239.
[2] Thompson, Neil C., et al. "The computational limits of deep learning." arXiv preprint arXiv:2007.05558 10 (2020): 2.
[3] Sevilla, Jaime, et al. "Compute trends across three eras of machine learning." 2022 international joint conference on neural networks (IJCNN). IEEE, 2022.

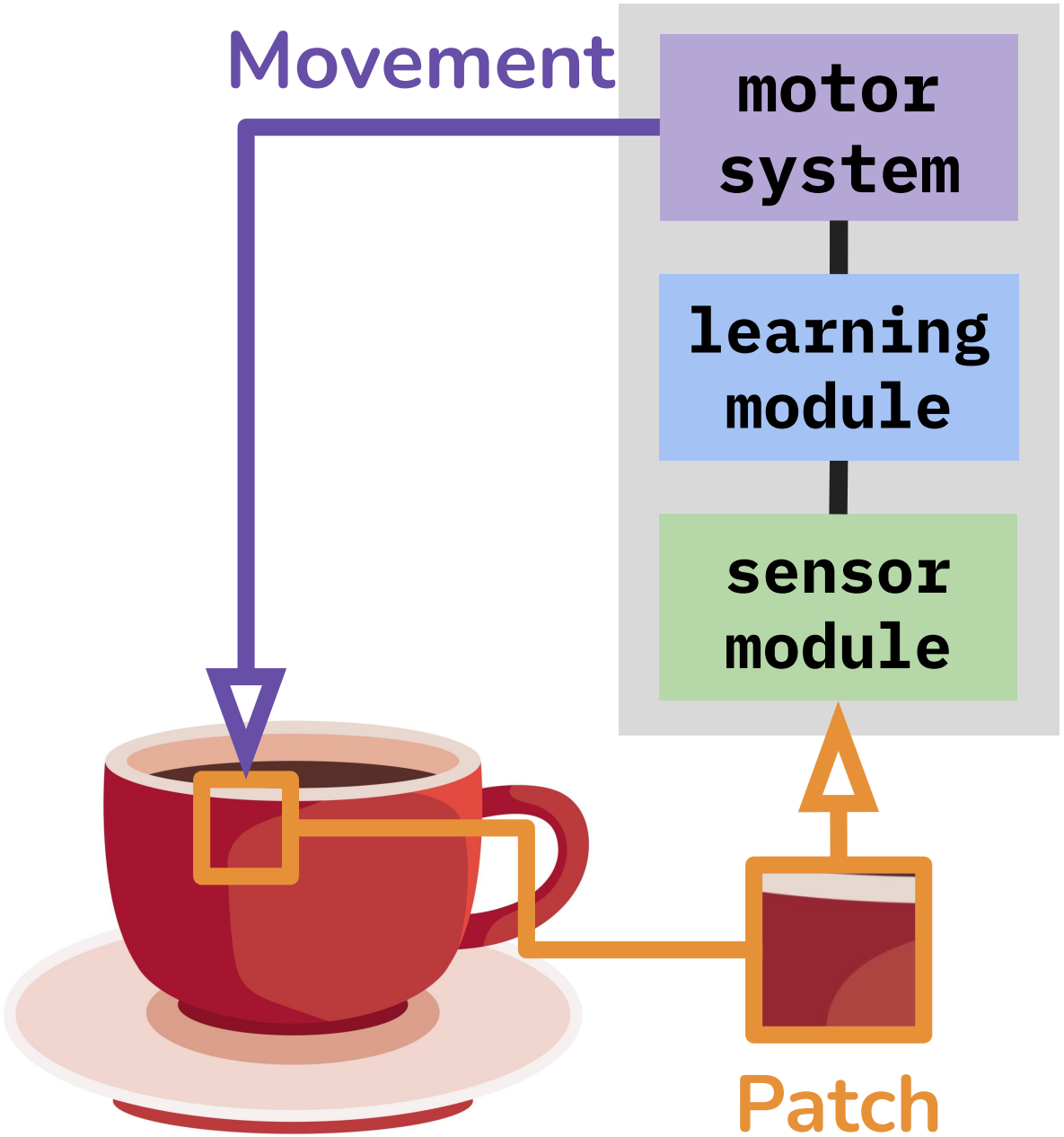
Thousand Brains Systems are a **new** and **emerging** class of **AI models**

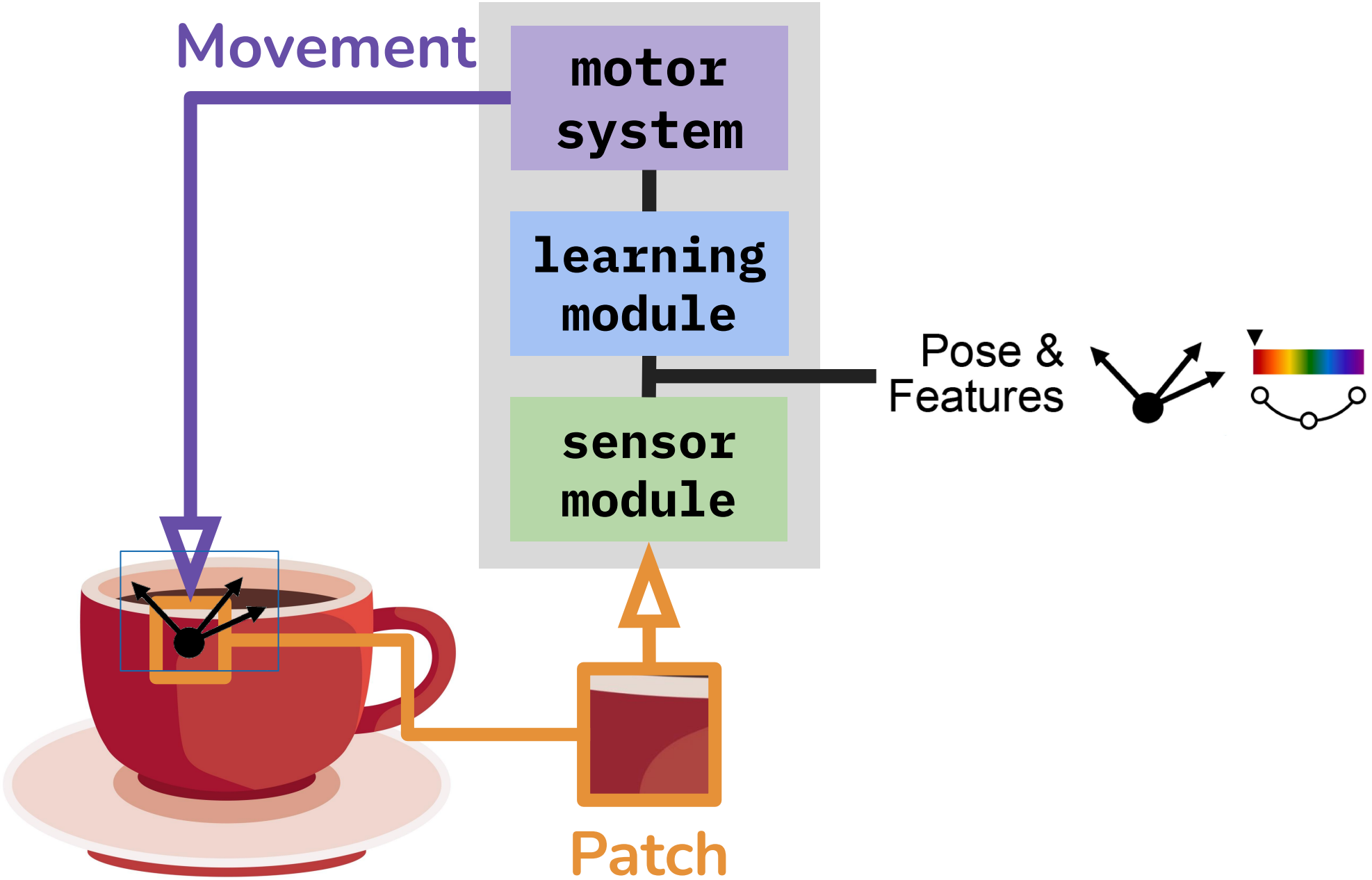
They try to **bridge the gap** between **artificial intelligence** and **biological intelligence**

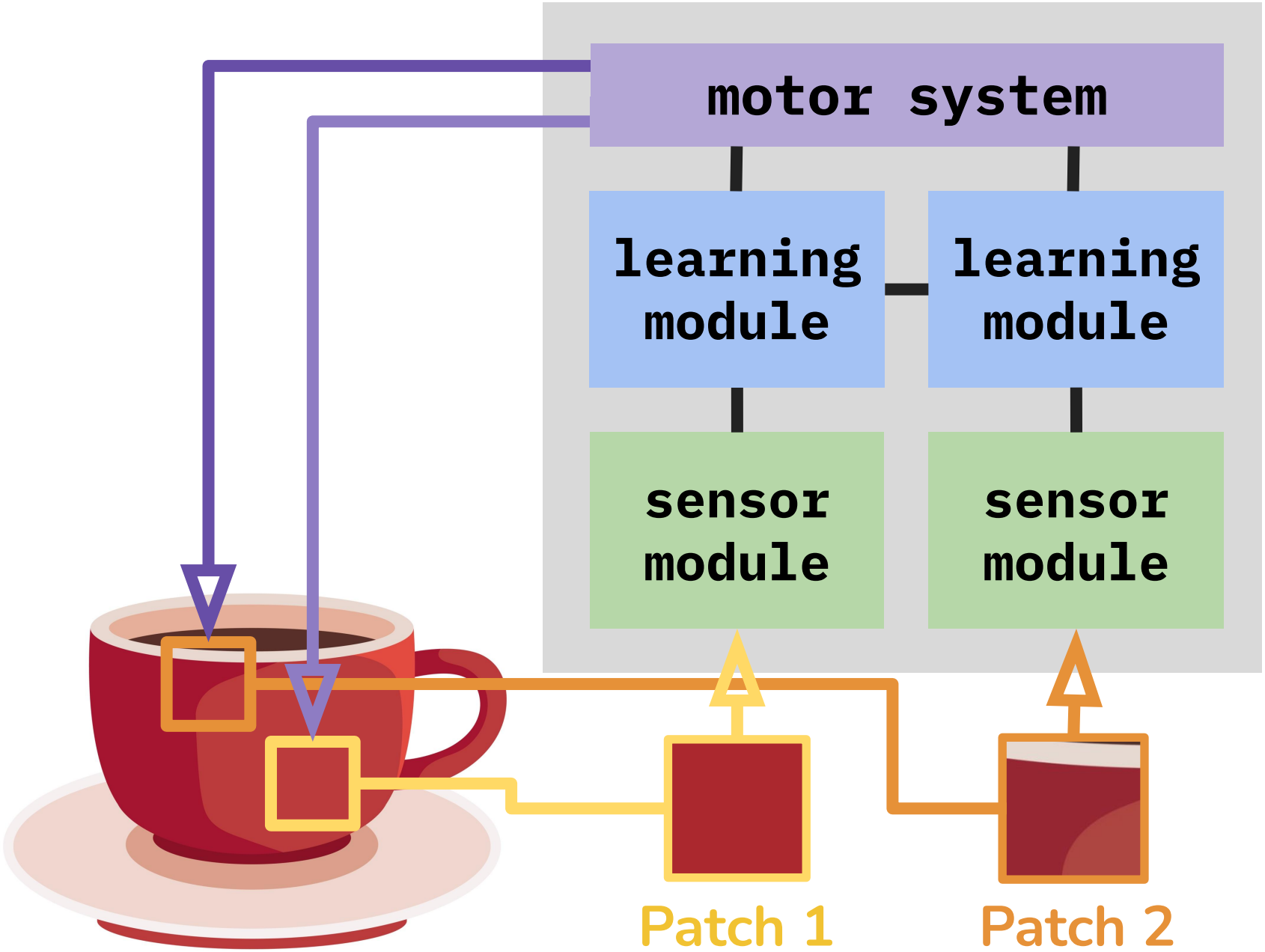
[1] Leadholm, Niels, et al. "Thousand-brains systems: Sensorimotor intelligence for rapid, robust learning and inference." arXiv preprint arXiv:2507.04494 (2025).
[2] Clay, Viviane, Niels Leadholm, and Jeff Hawkins. "The Thousand Brains Project: A New Paradigm for Sensorimotor Intelligence." arXiv preprint arXiv:2412.18354 (2024).

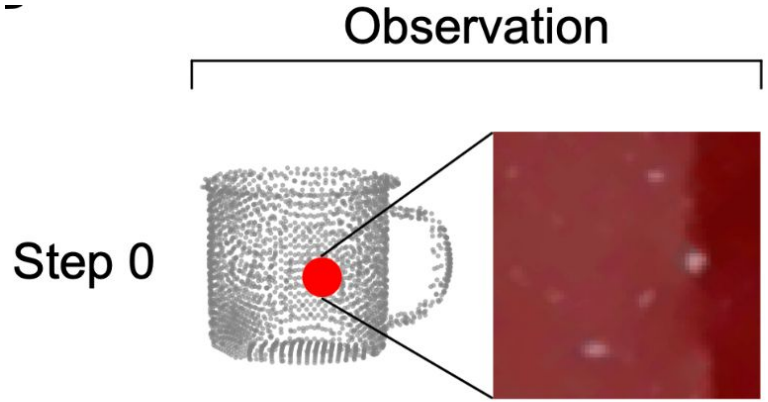

Movement
Model
Patch

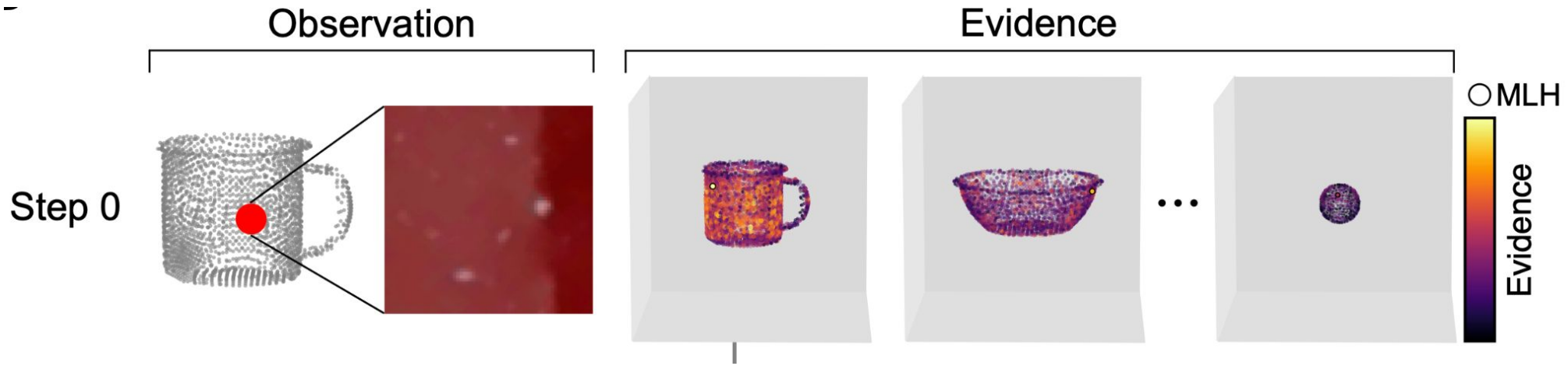

Movement
Model
Patch

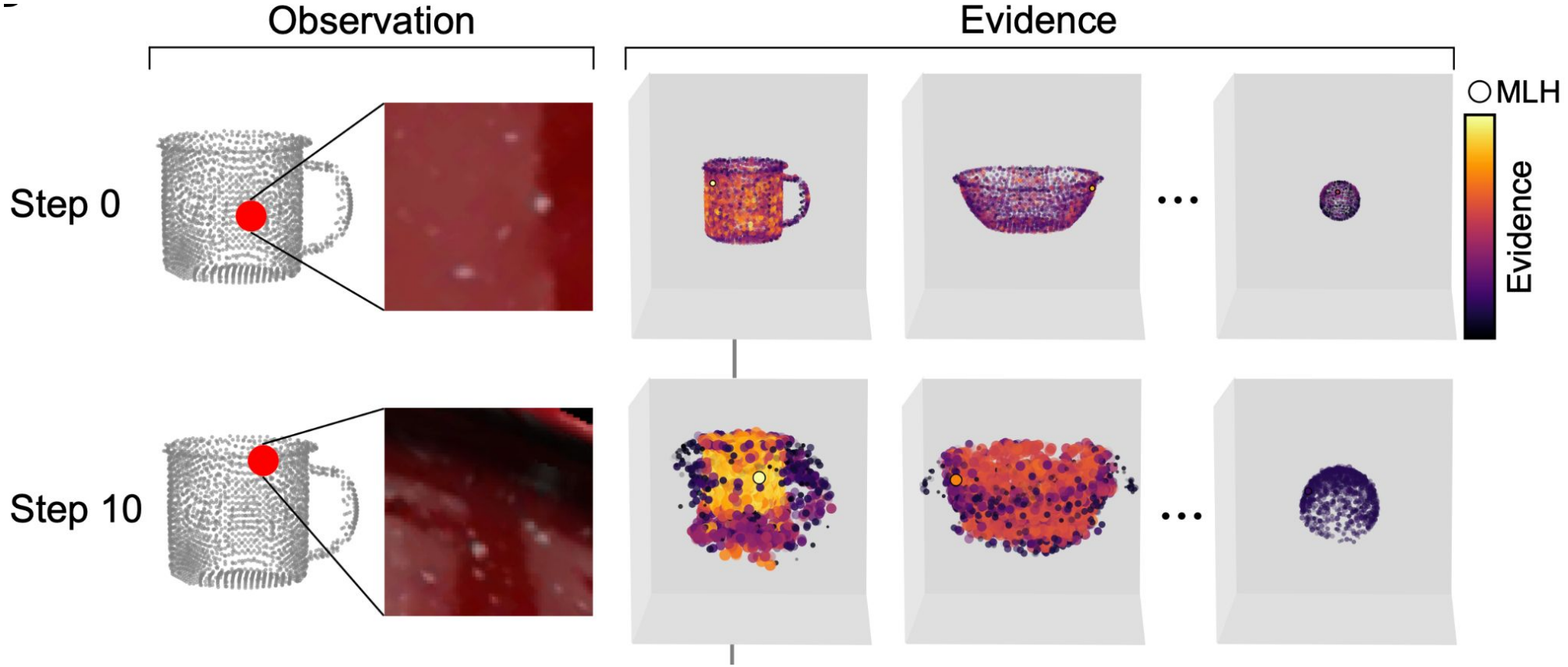

Movement
motor system
learning module
sensor module
Patch

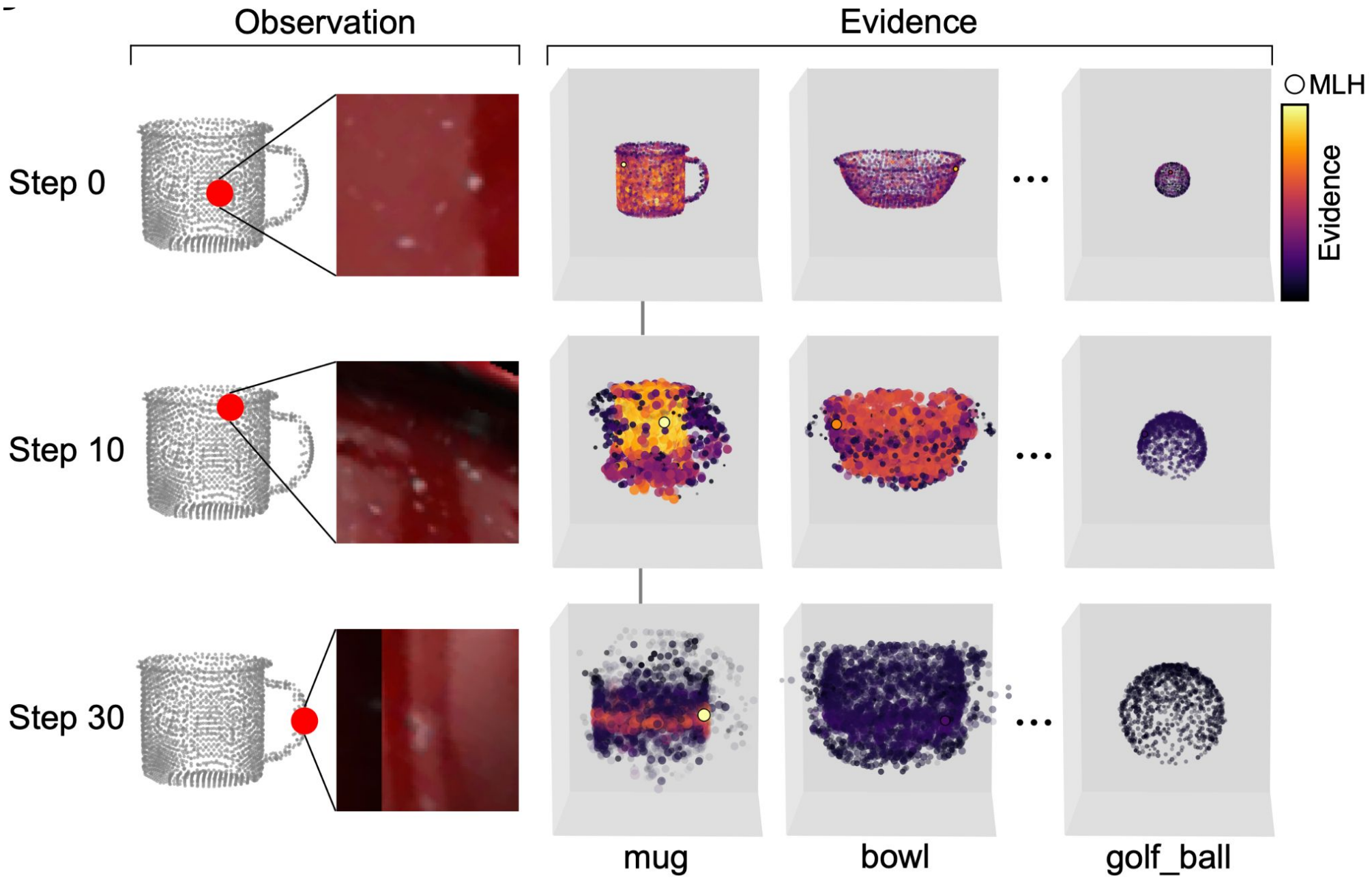

Movement
motor system
learning module
sensor module
Pose & Features
Patch


motor system
learning module
learning module
sensor module
sensor module
Patch 1
Patch 2


Observation
Step 0


Observation
Evidence
Step 0
MLH
Evidence


Observation
Evidence
MLH
Evidence
Step 0
Step 10


Observation
Evidence
MLH
Evidence
Step 0
Step 10
Step 30
mug
bowl
golf_ball

Rapid Learning

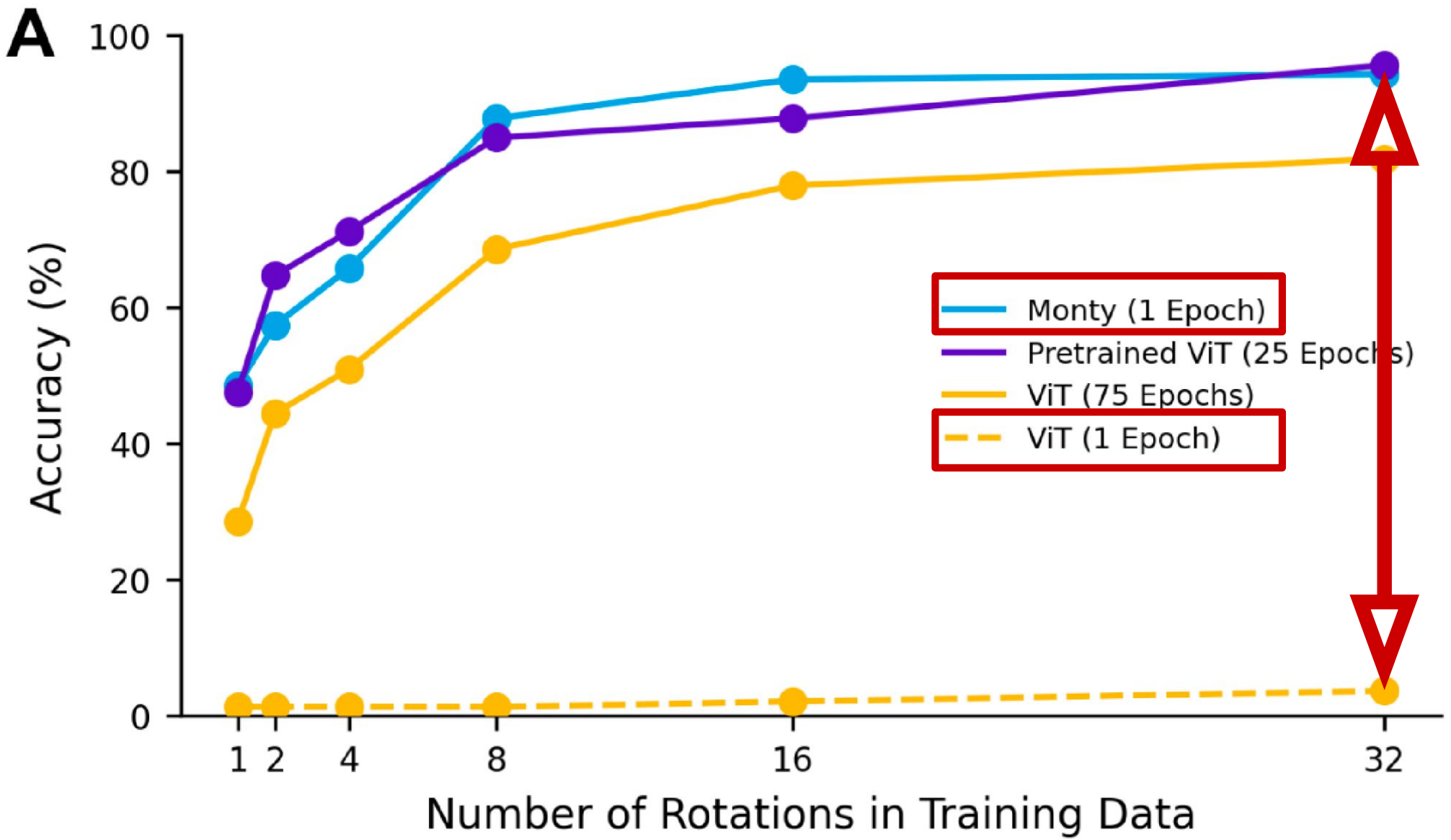


Figure 7A of Leadholm, Niels, et al. "Thousand-Brains Systems: Sensorimotor Intelligence for Rapid, Robust Learning and Inference." *arXiv preprint arXiv:2507.04494* (2025).

Continual Learning

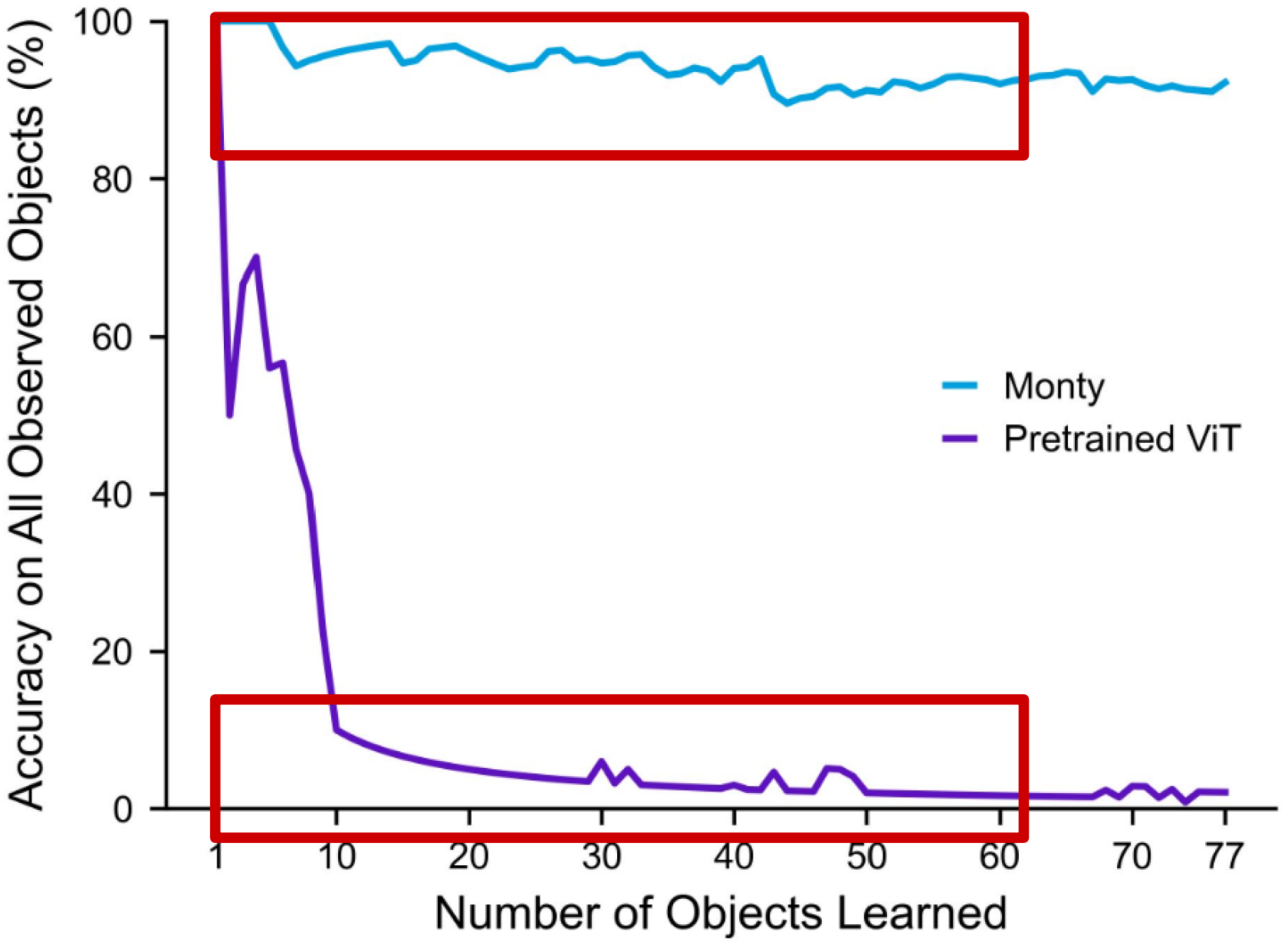


Figure 7CD of Leadholm, Niels, et al. "Thousand-Brains Systems: Sensorimotor Intelligence for Rapid, Robust Learning and Inference." *arXiv preprint arXiv:2507.04494* (2025).

Compute efficient learning

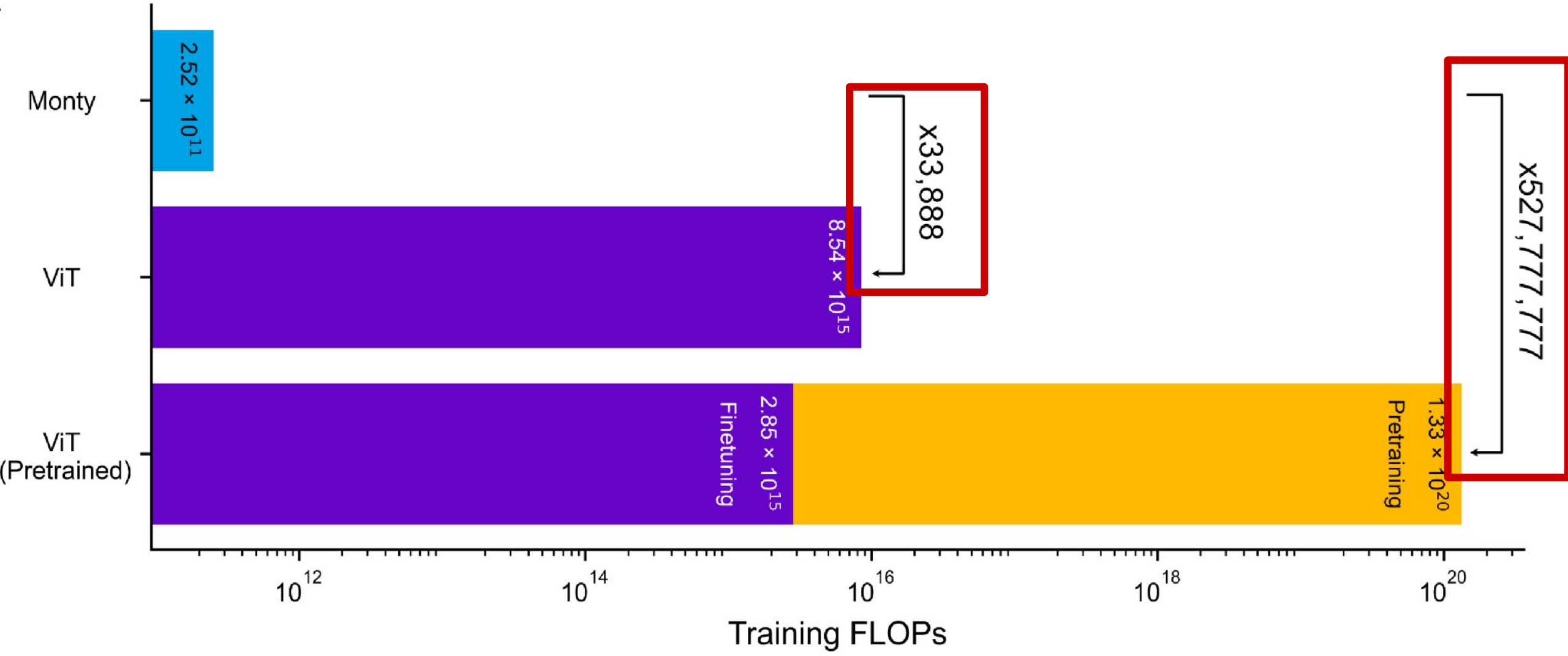
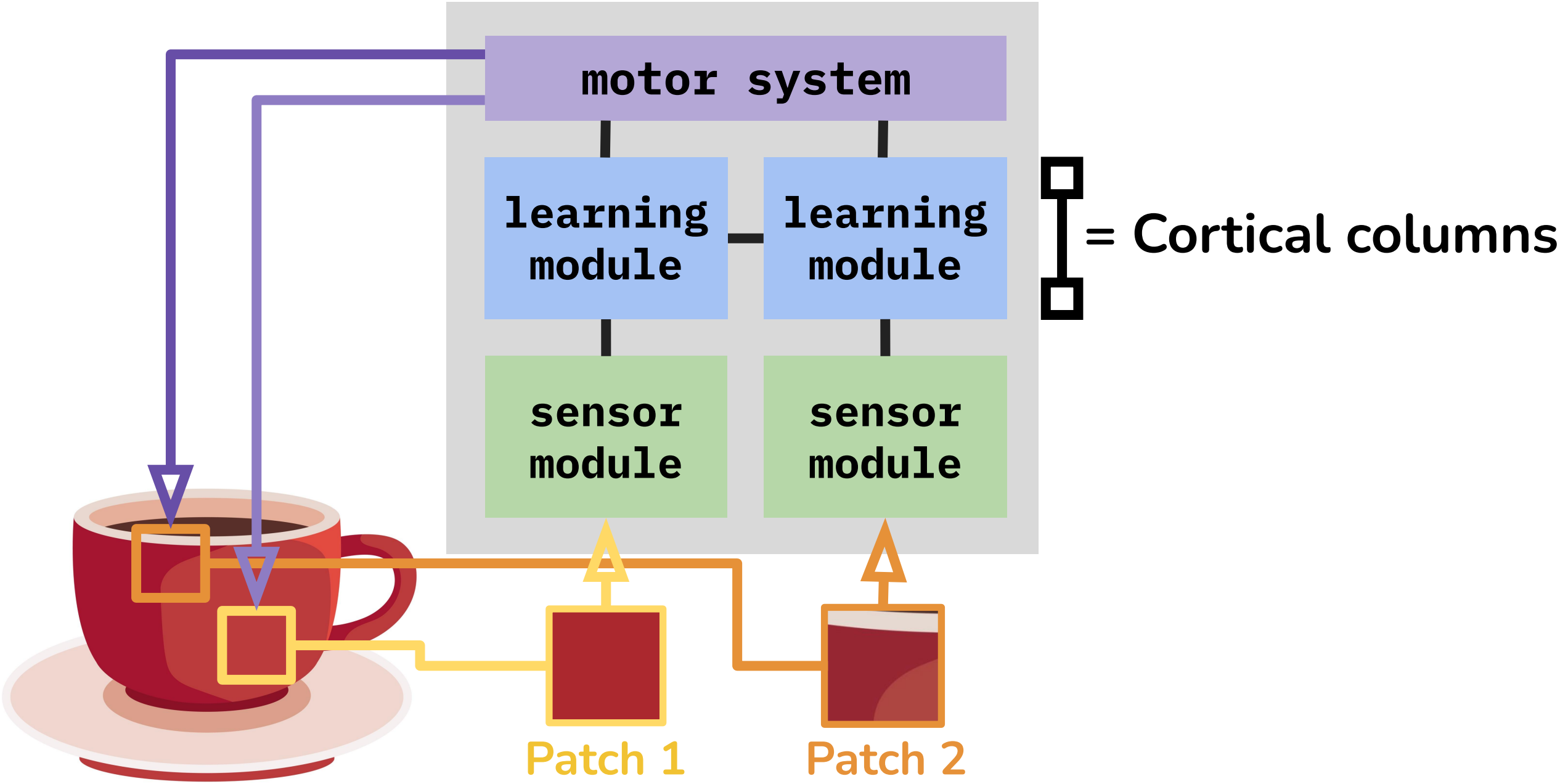
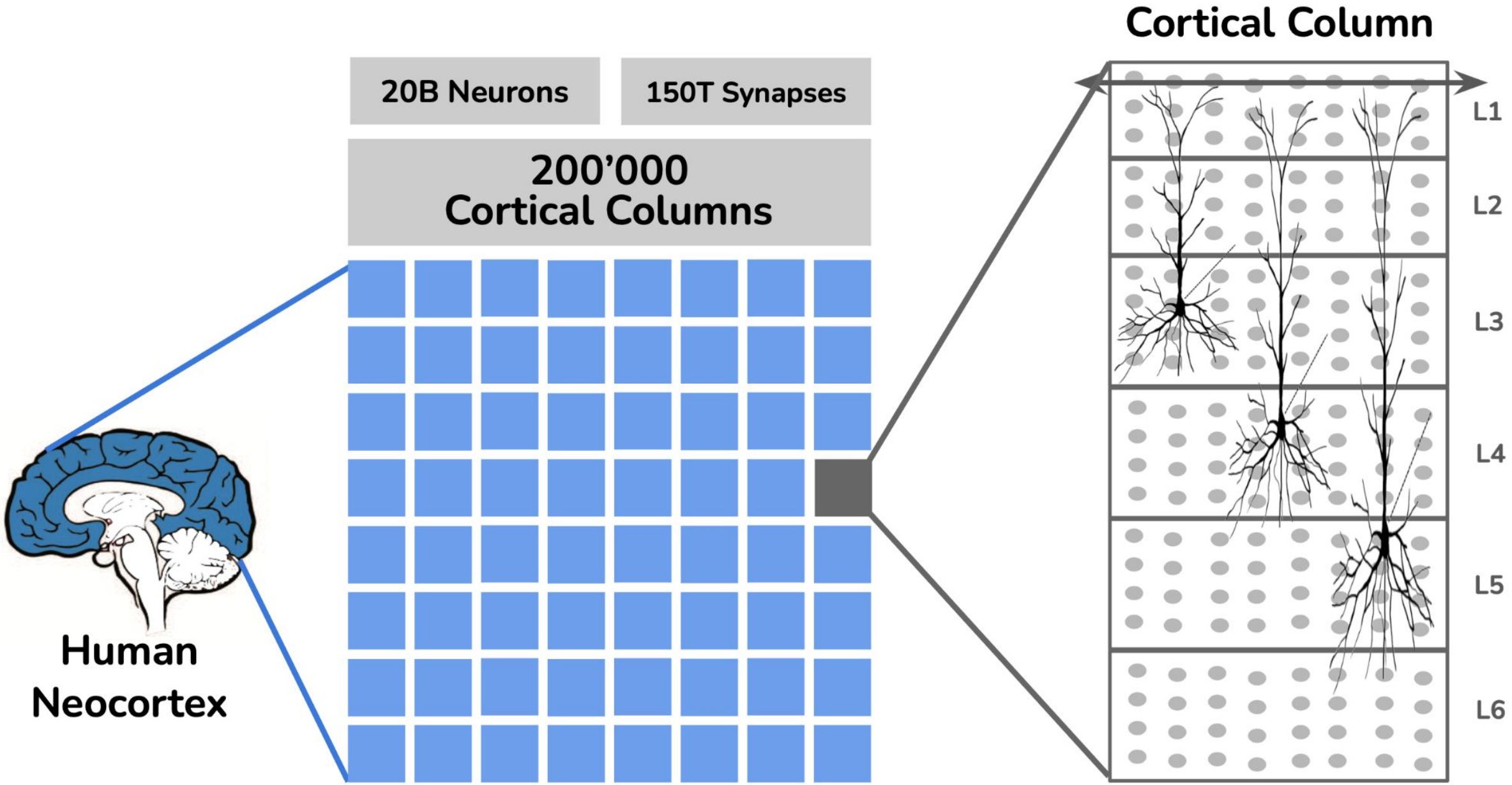


Figure 8 of Leadholm, Niels, et al. "Thousand-Brains Systems: Sensorimotor Intelligence for Rapid, Robust Learning and Inference." *arXiv preprint arXiv:2507.04494* (2025).


motor system
learning module
learning module
sensor module
sensor module
= Cortical columns
Patch 1
Patch 2


Cortical Column
20B Neurons
150T Synapses
200'000
Cortical Columns
Human
Neocortex
L1
L2
L3
L4
L5
L6

[1] Jardim-Messeder, Débora, et al. "Dogs have the most neurons, though not the largest brain: trade-off between body mass and number of neurons in the cerebral cortex of large carnivoran species." *Frontiers in neuroanatomy* 11 (2017): 296229.
[2] Herculano-Houzel, Suzana, Bruno Mota, and Roberto Lent. "Cellular scaling rules for rodent brains." Proceedings of the National Academy of Sciences 103.32 (2006): 12138-12143.
[3] Azevedo, Frederico AC, et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain." Journal of Comparative Neurology 513.5 (2009): 532-541.
[4] Rough estimate, little inter-species variation in column size, see Mountcastle, Vernon B. "The columnar organization of the neocortex." Brain: a journal of neurology 120.4 (1997): 701-722.

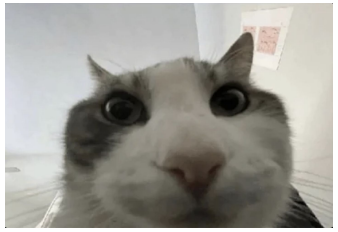
	Neurons	Synapses	Columns [4]
Human [3]	20 B	150 T	200 K

[1] Jardim-Messeder, Débora, et al. "Dogs have the most neurons, though not the largest brain: trade-off between body mass and number of neuron … *Frontiers in neuroanatomy* 11 (2017): 296229.
[2] Herculano-Houzel, Suzana, Bruno Mota, and Roberto Lent. "Cellular scaling rules for rodent brains." Proceedings of the National Academy of Sc…
[3] Azevedo, Frederico AC, et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain … 32-541.
[4] Rough estimate, little inter-species variation in column size, see Mountcastle, Vernon B. "The columnar organization of the neocortex." Brain: a jo…

	Neurons	Synapses	Columns [4]
Human [3]	20 B	150 T	200 K
Cat [1]	250 M	1.9 T	2.5 K

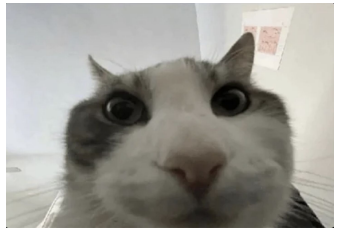
[1] Jardim-Messeder, Débora, et al. "Dogs have the most neurons, though not the largest brain: trade-off between body m… …rs *in neuroanatomy* 11 (2017): 296229.
[2] Herculano-Houzel, Suzana, Bruno Mota, and Roberto Lent. "Cellular scaling rules for rodent brains." Proceedings of the…
[3] Azevedo, Frederico AC, et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically…
[4] Rough estimate, little inter-species variation in column size, see Mountcastle, Vernon B. "The columnar organization of …

	Neurons	Synapses	Columns [4]
Human [3]	20 B	150 T	200 K
Cat [1]	250 M	1.9 T	2.5 K
Guinea Pig [2]	43.5 M	326 B	435



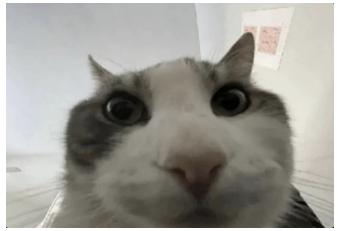
[1] Jardim-Messeder, Débora, et al. "Dogs have the most neurons, though not the largest brain: trade-off between body mass and number of neuron ...iers in *neuroanatomy* 11 (2017): 296229.
[2] Herculano-Houzel, Suzana, Bruno Mota, and Roberto Lent. "Cellular scaling rules for rodent brains." Proceedings of the National Academy of Sci...
[3] Azevedo, Frederico AC, et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain." ...1.
[4] Rough estimate, little inter-species variation in column size, see Mountcastle, Vernon B. "The columnar organization of the neocortex." Brain: a jou...

	Neurons	Synapses	Columns [4]
Human [3]	20 B	150 T	200 K
Cat [1]	250 M	1.9 T	2.5 K
Guinea Pig [2]	43.5 M	326 B	435
Mouse [2]	13.5 M	102 B	135

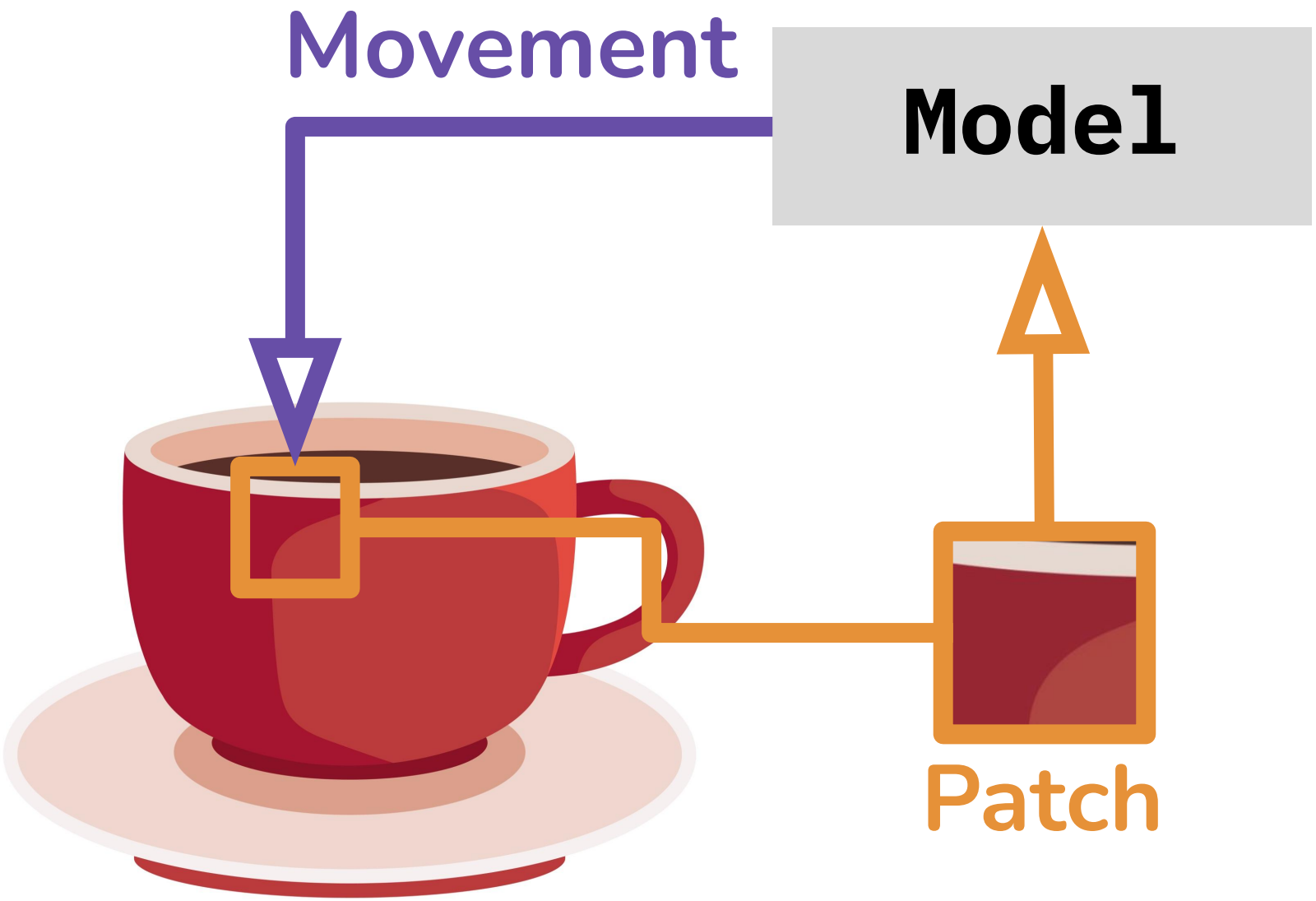



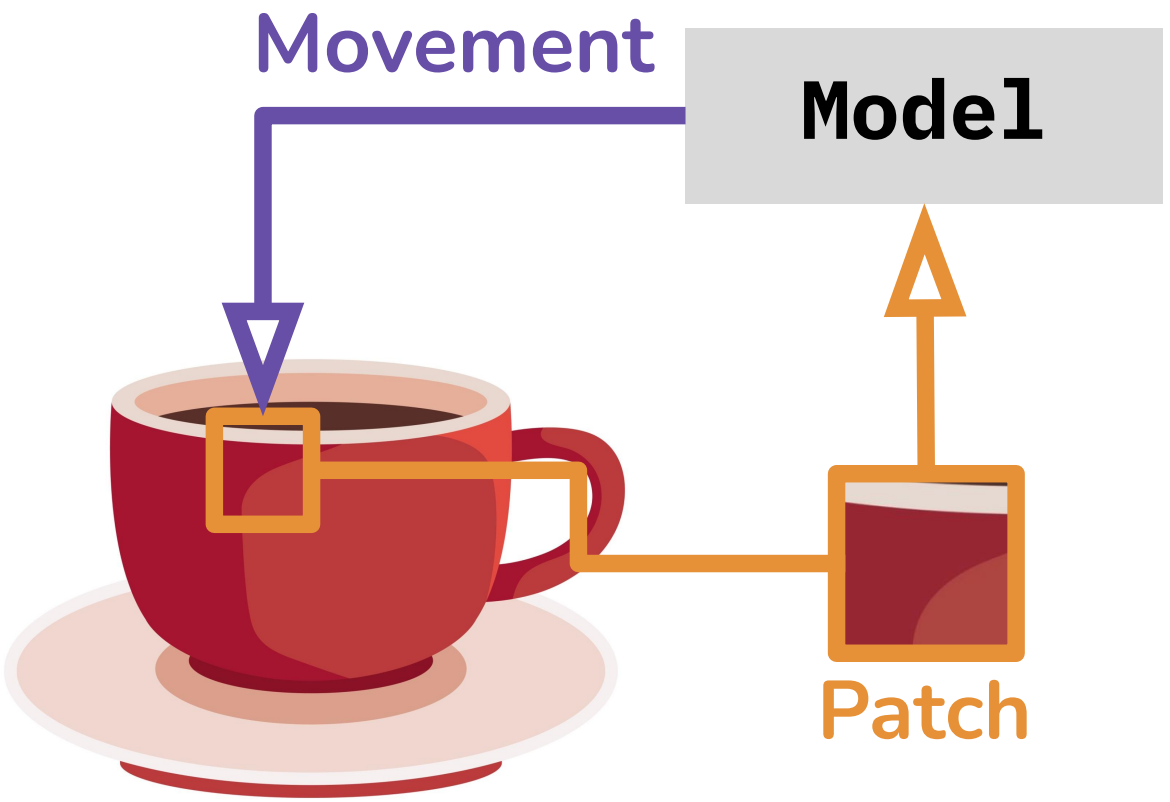
[1] Jardim-Messeder, Débora, et al. "Dogs have the most neurons, though not the largest brain: trade-off between body mass and number of neurons in the cerebral cortex of large carnivoran species." *Frontiers in neuroanatomy* 11 (2017): 296229.
[2] Herculano-Houzel, Suzana, Bruno Mota, and Roberto Lent. "Cellular scaling rules for rodent brains." Proceedings of the National Academy of Sciences 103.32 (2006): 12138-12143.
[3] Azevedo, Frederico AC, et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain." Journal of Comparative Neurology 513.5 (2009): 532-541.
[4] Rough estimate, little inter-species variation in column size, see Mountcastle, Vernon B. "The columnar organization of the neocortex." Brain: a journal of neurology 120.4 (1997): 701-722.

	Neurons	Synapses	Columns [4]
Human [3]	20 B	150 T	200 K
Cat [1]	250 M	1.9 T	2.5 K
Guinea Pig [2]	43.5 M	326 B	435
Mouse [2]	13.5 M	102 B	135



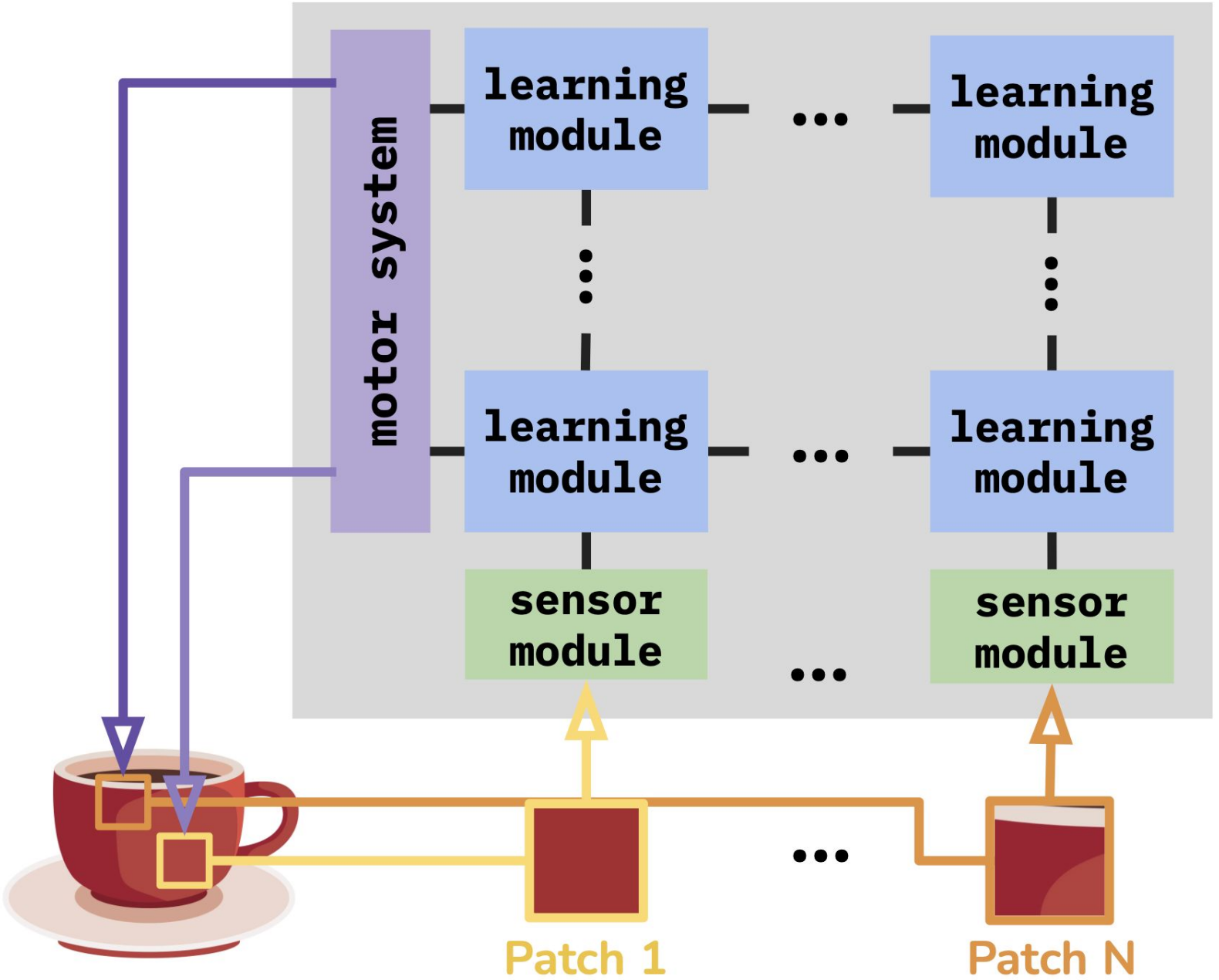
Consequences on **computing systems**?

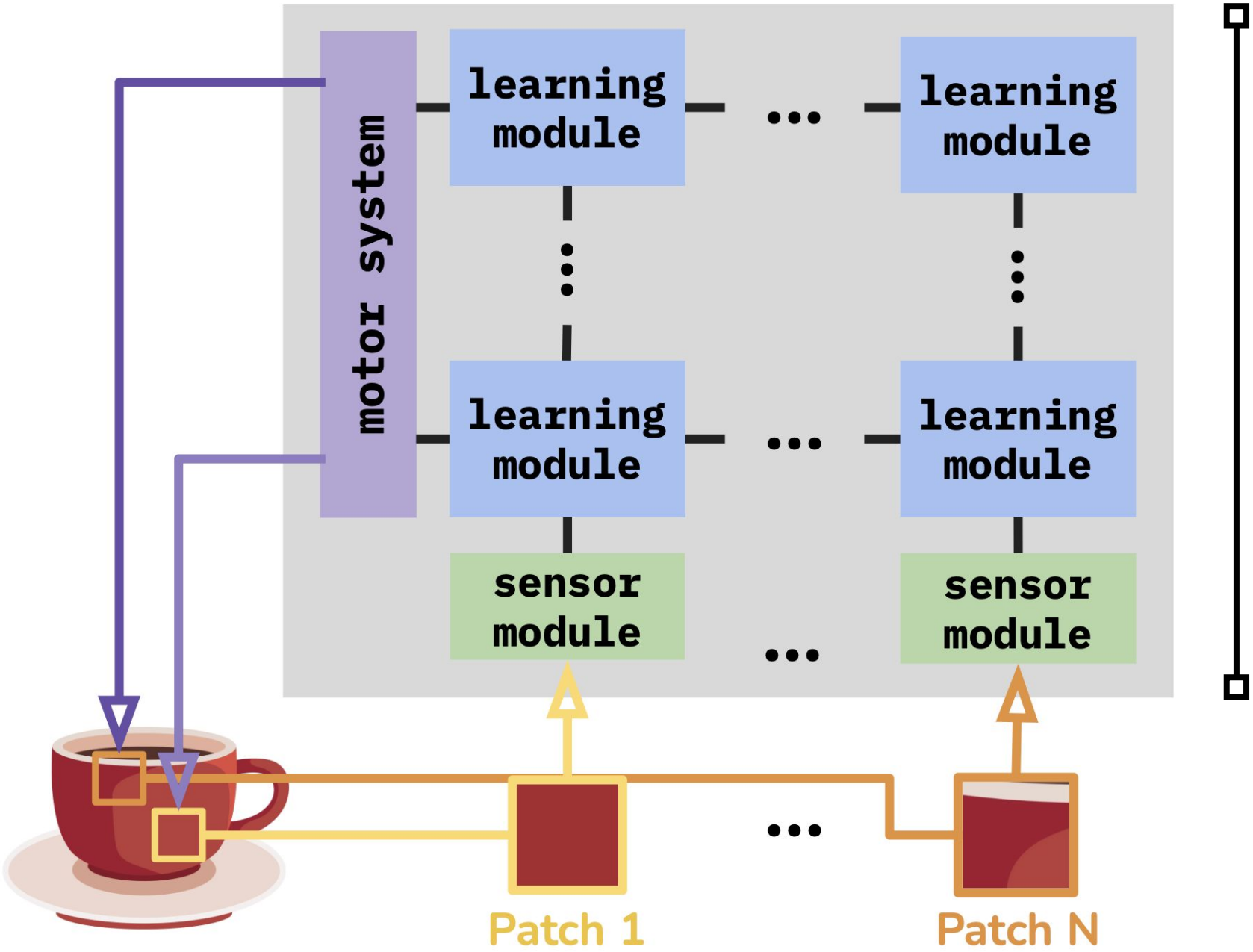

Movement
Model
Patch



Auto-regressive loop: cannot **parallelize** **across** steps

→ **Parallelize** **within** a step


motor system
learning module
learning module
learning module
learning module
sensor module
sensor module
Patch 1
Patch N

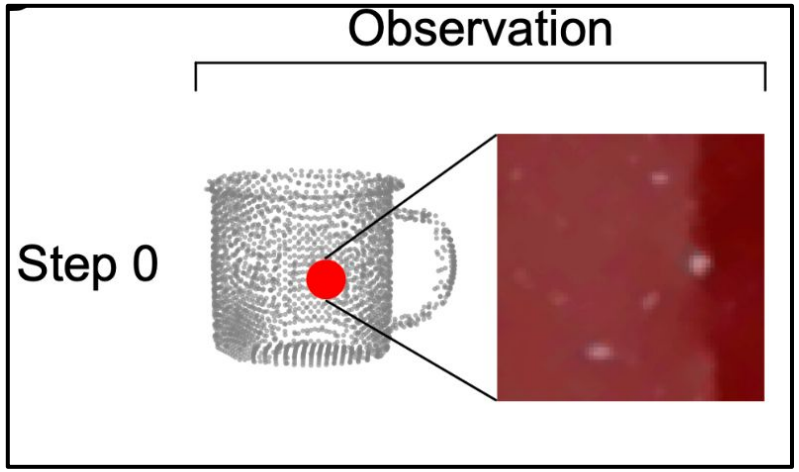


Parallelization opportunity

We investigate the **scalability** of **Thousand Brains Systems** on ❶ **GPUs**, ❷ **CPUs** and ❸ **Processing-in-Memory (PiM)**

Monty

Released in 2024, it is written in Python and builds models of the world using **explicit graphs** in 3d space



[1] Leadholm, Niels, et al. "Thousand-Brains Systems: Sensorimotor Intelligence for Rapid, Robust Learning and Inference." arXiv preprint arXiv:2507.04494 (2025).
[2] Clay, Viviane, Niels Leadholm, and Jeff Hawkins. "The Thousand Brains Project: A New Paradigm for Sensorimotor Intelligence." arXiv preprint arXiv:2412.18354 (2024).
[3] https://github.com/thousandbrainsproject/tbp.monty

Montyll

We introduce **Montyll**, a **novel** Thousand Brains Systems.

Montyll² is implemented using elements of **low-level cortical processing** (e.g. **accurate neuron models**)

Montyll was designed to align with the **long terms goals** of the Thousand Brains Project[1] and represent them computationally

[1] Clay, Viviane, Niels Leadholm, and Jeff Hawkins. "The Thousand Brains Project: A New Paradigm for Sensorimotor Intelligence." arXiv preprint arXiv:2412.18354 (2024).
[2] https://github.com/Xavier0301/cmontyll

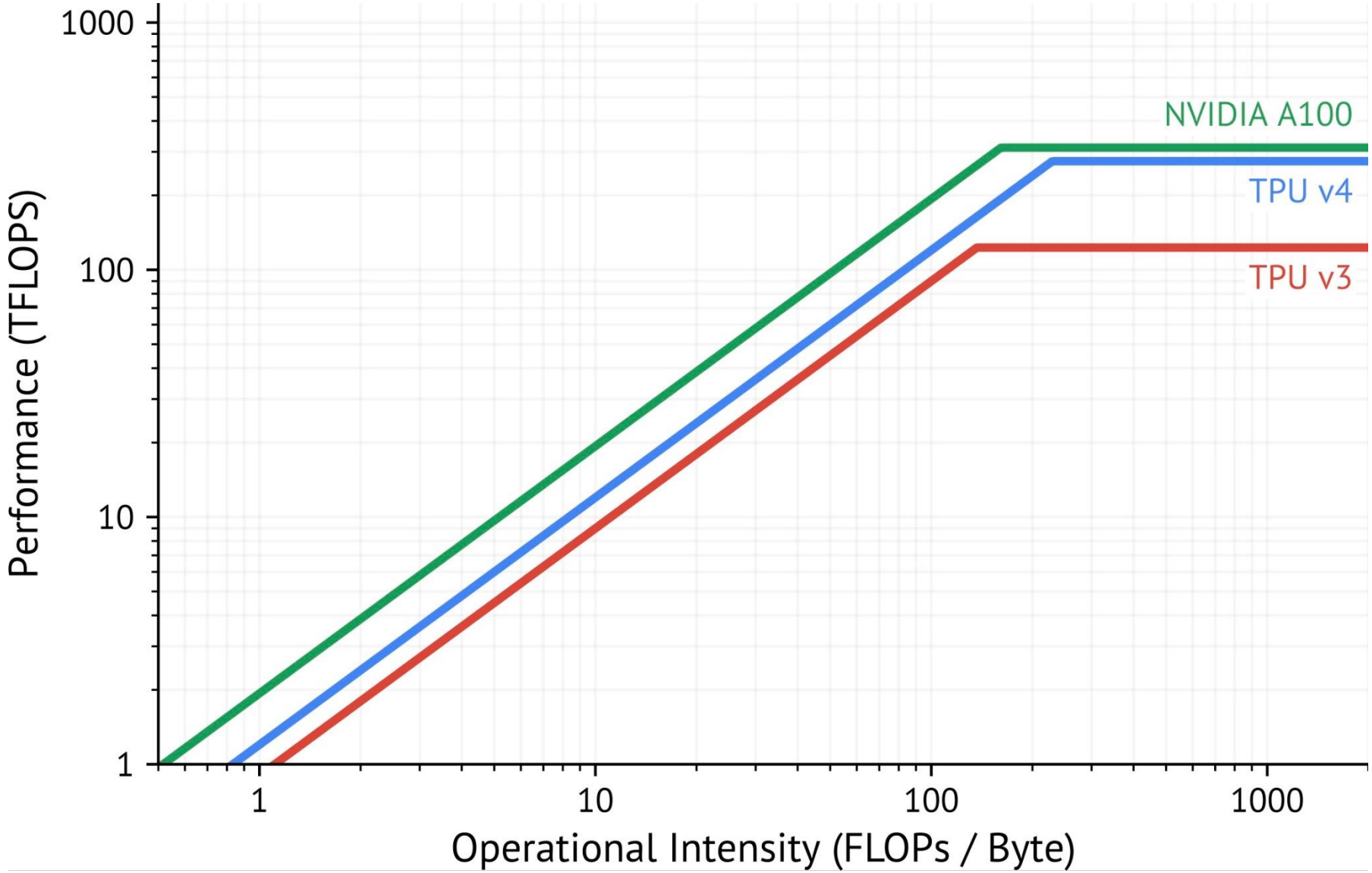
❶ GPUs

Operational Intensity = Operations / Byte

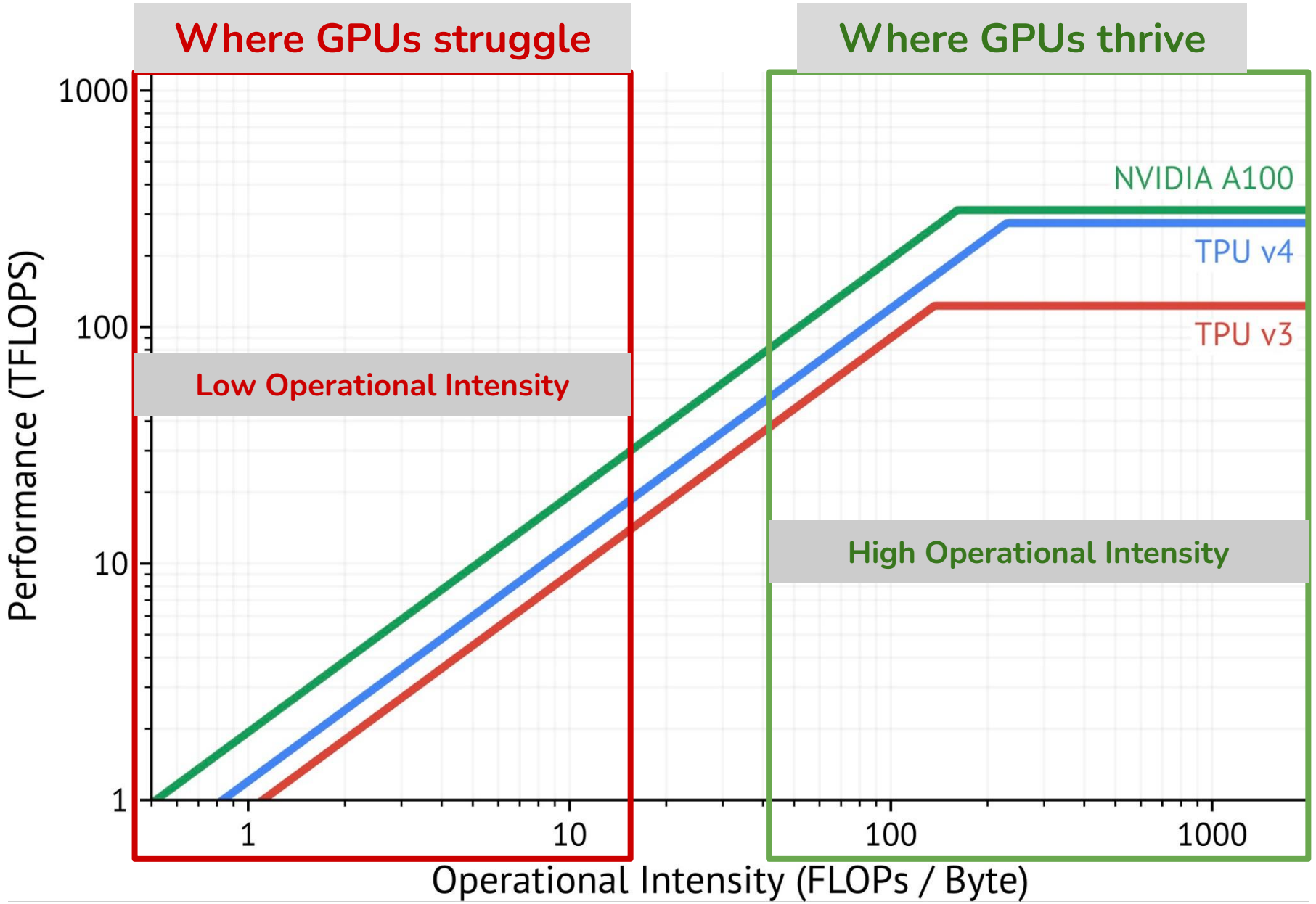
Measures how much **data reuse** a program exhibits.

Thousand Brains Systems

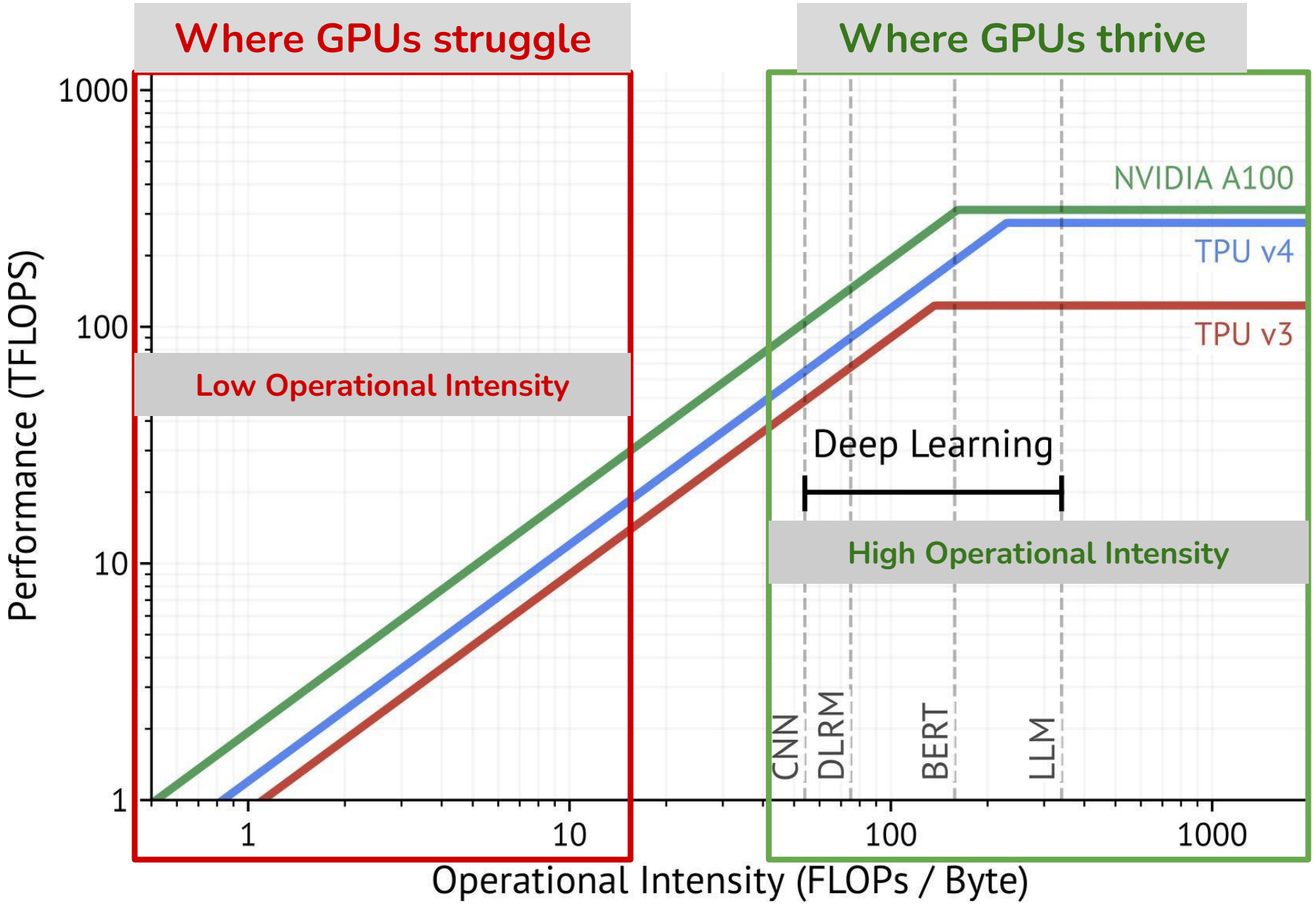
→ **Low Operational Intensity**



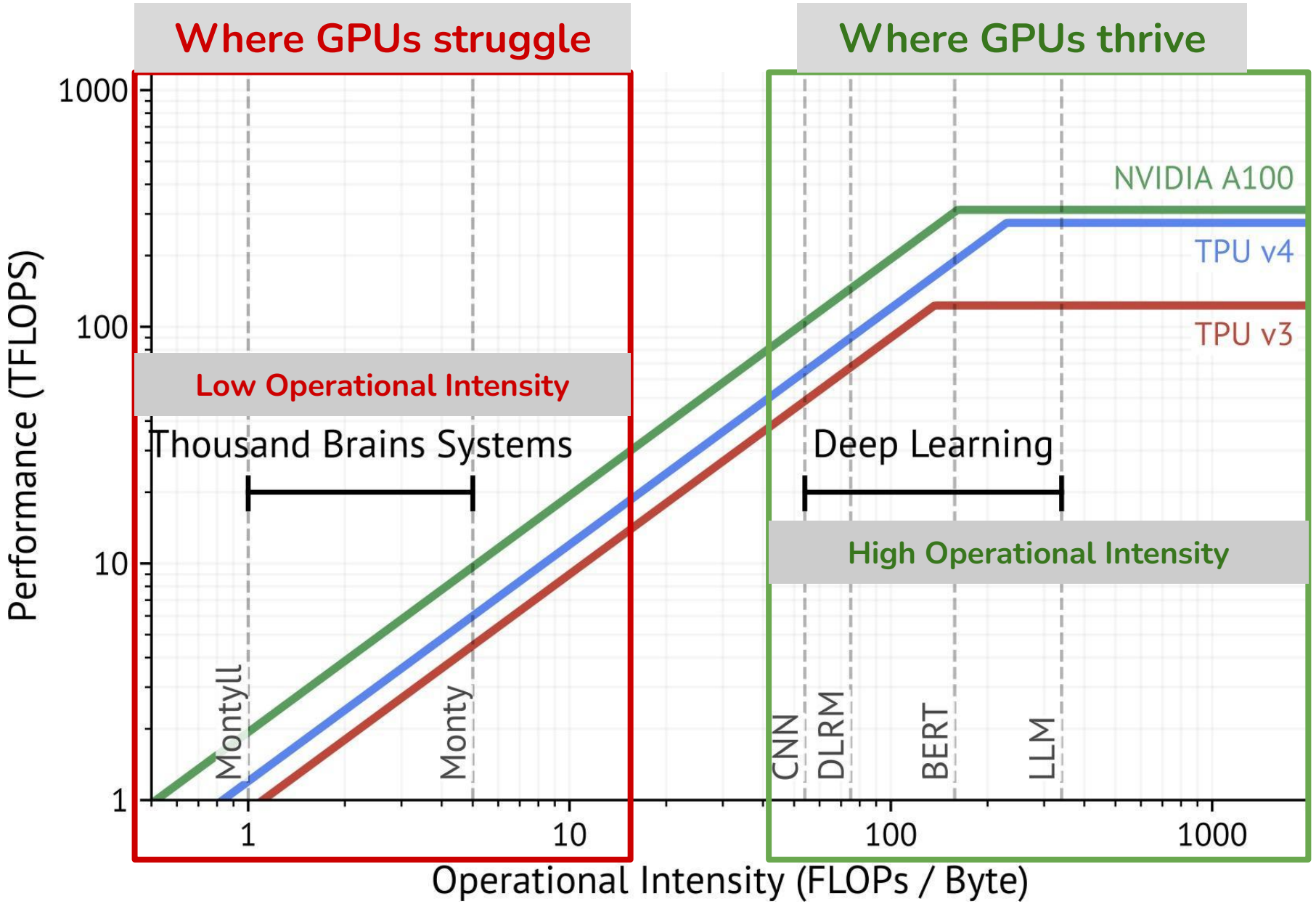
Jouppi, Norm, et al. "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings." Proceedings of the 50th ISCA 2023.



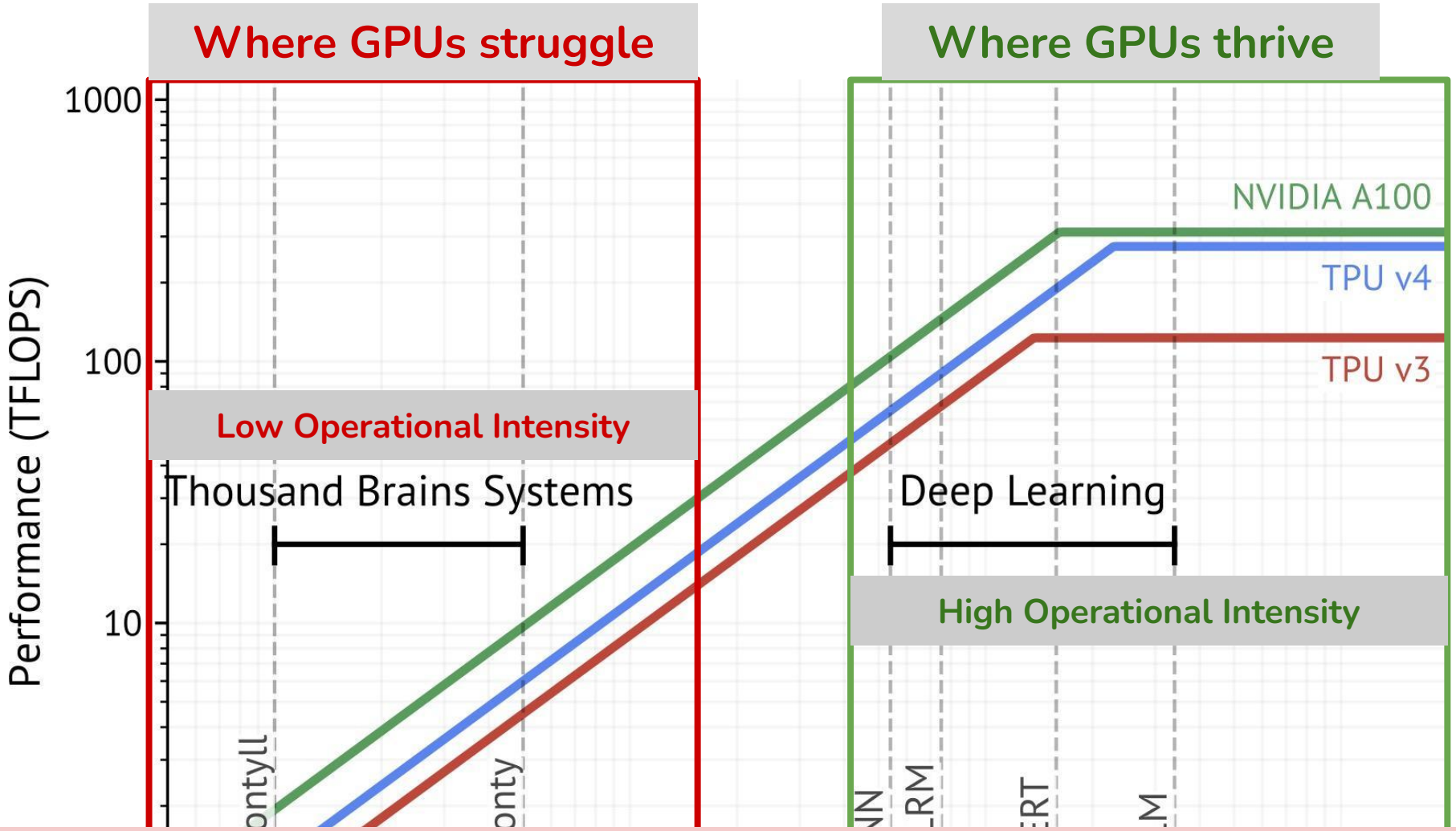
Jouppi, Norm, et al. "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings." Proceedings of the 50th ISCA 2023.



Jouppi, Norm, et al. "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings." Proceedings of the 50th ISCA 2023.

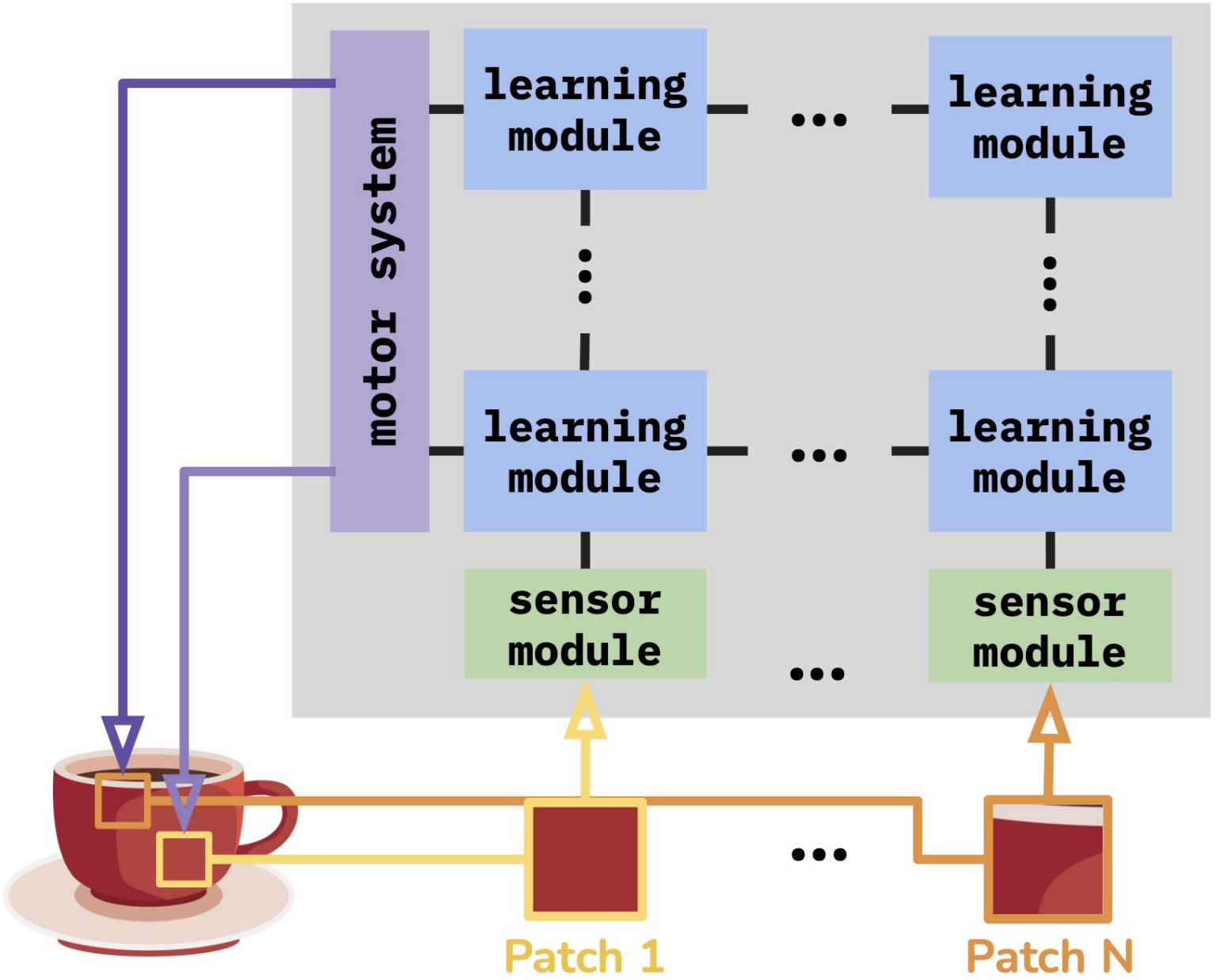


Jouppi, Norm, et al. "Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings." Proceedings of the 50th ISCA 2023.

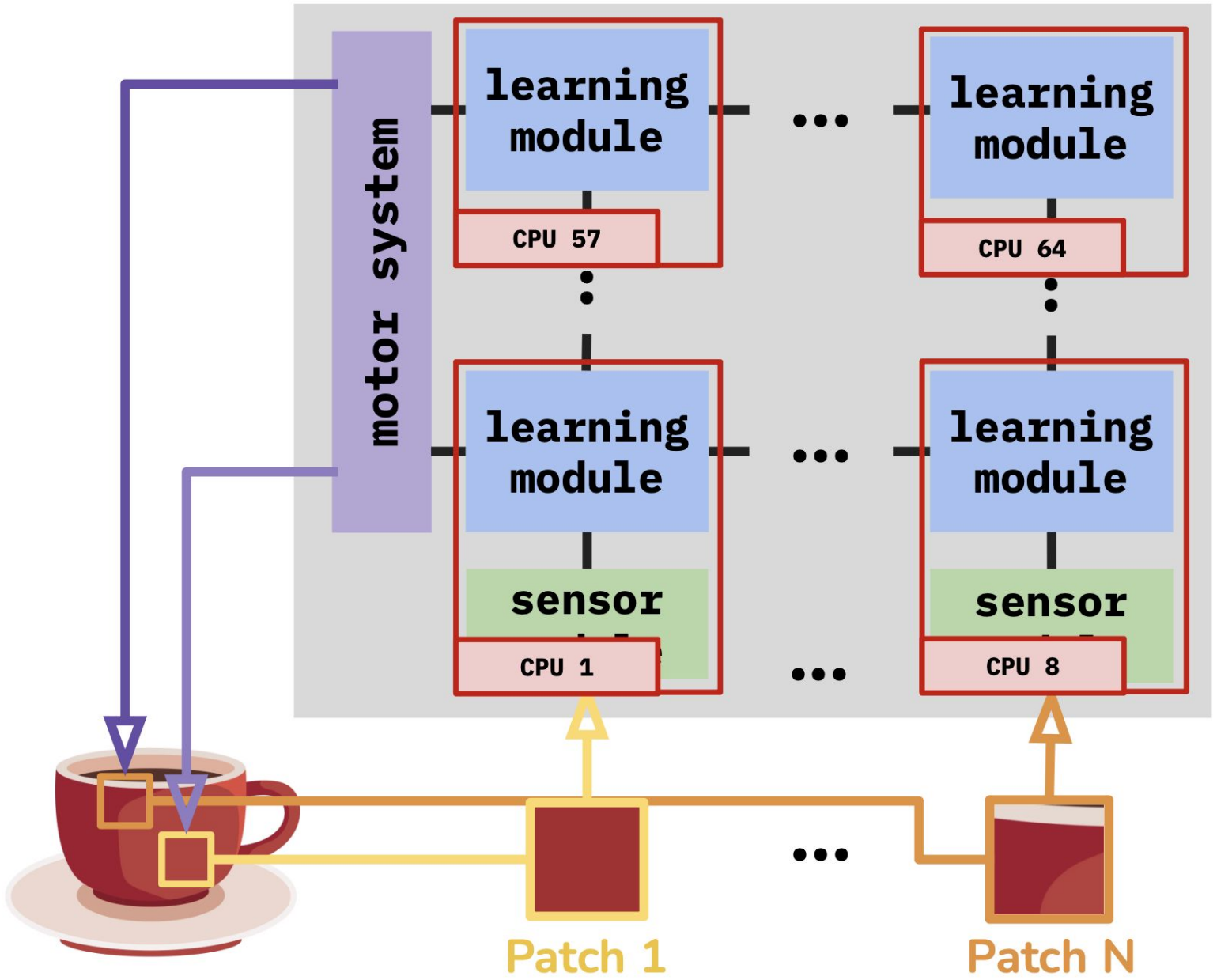


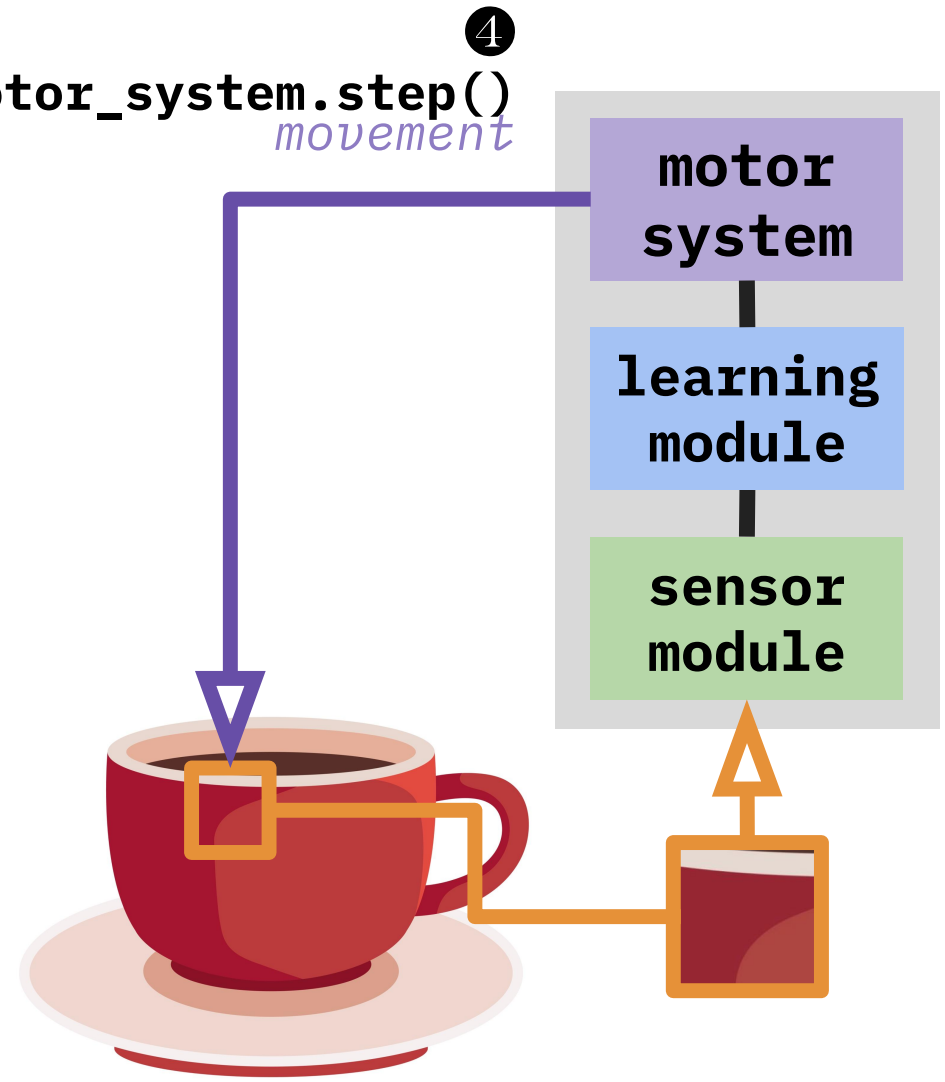
→ GPUs are not a good fit to scale up Thousand Brains Systems

❷ CPUs


motor system
learning module
learning module
learning module
learning module
sensor module
sensor module
Patch 1
Patch N

❷ CPUs


motor system
learning module
CPU 57
learning module
CPU 64
learning module
sensor
CPU 1
learning module
sensor
CPU 8
Patch 1
Patch N



object_id, pose

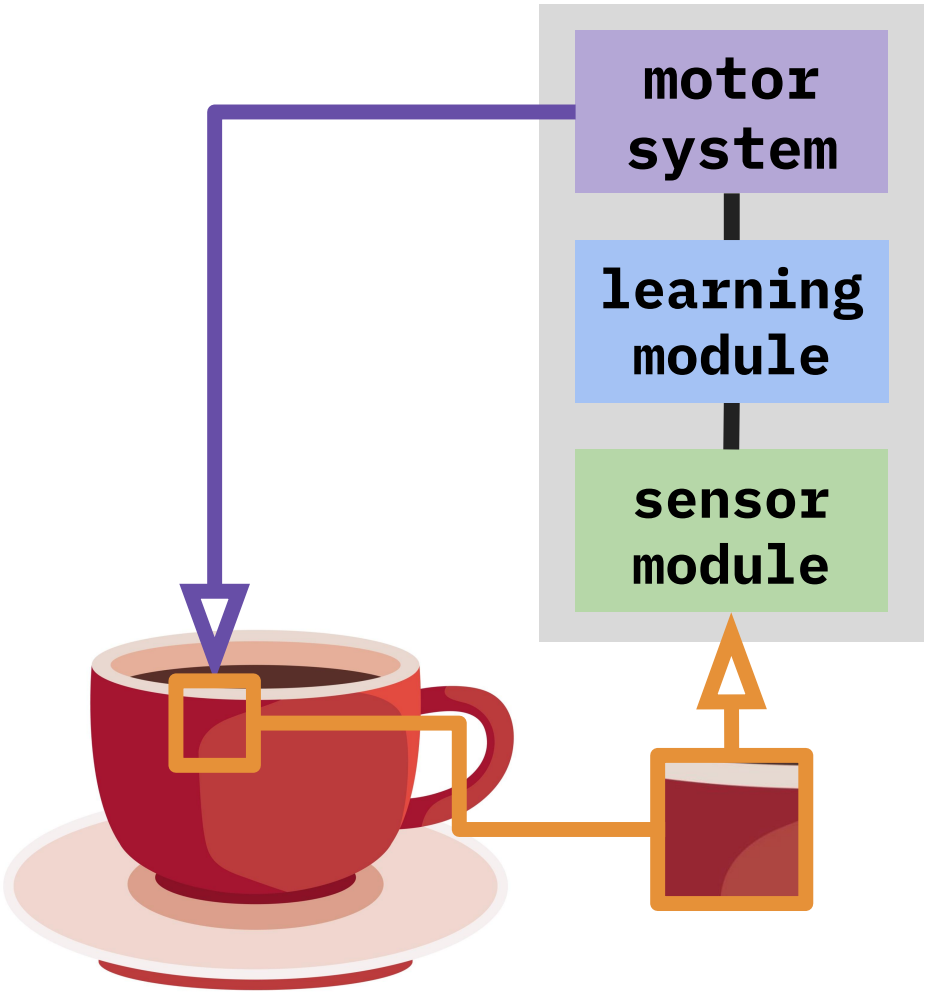
❸ **learning_module.step()**

feature, pose

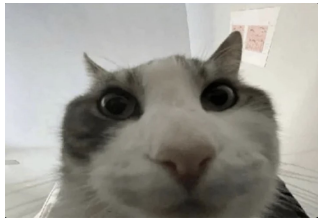
❷ **sensor_module.step()**

patch

❶ **environment.get_patch()**



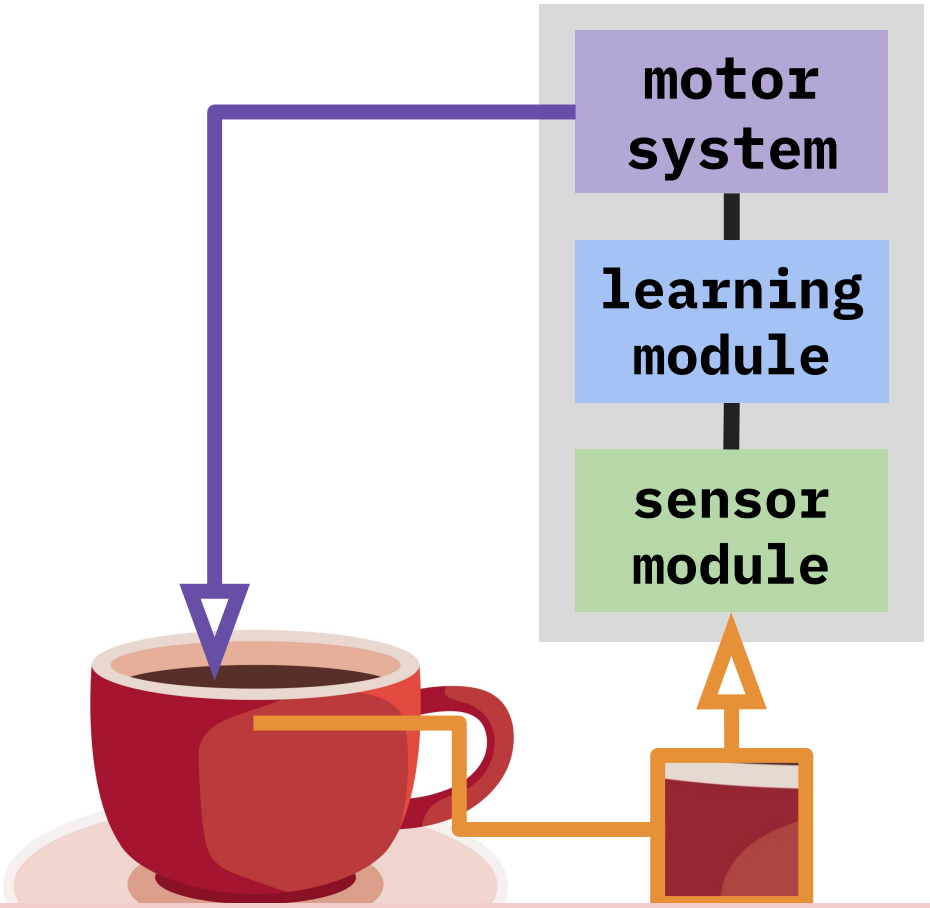
Cat *cortex*-scale system
2500 learning modules:



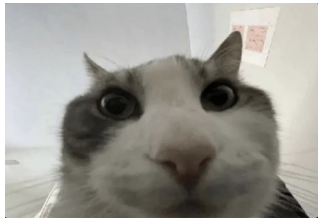
Monty[1] data movement:
12.5 - 125 GiB

Montyll[2] data movement:
87.5-175 GiB

1: https://github.com/thousandbrainsproject/tbp.monty
2: https://github.com/Xavier0301/montyll



***Cat cortex**-scale system*
2500 learning modules:



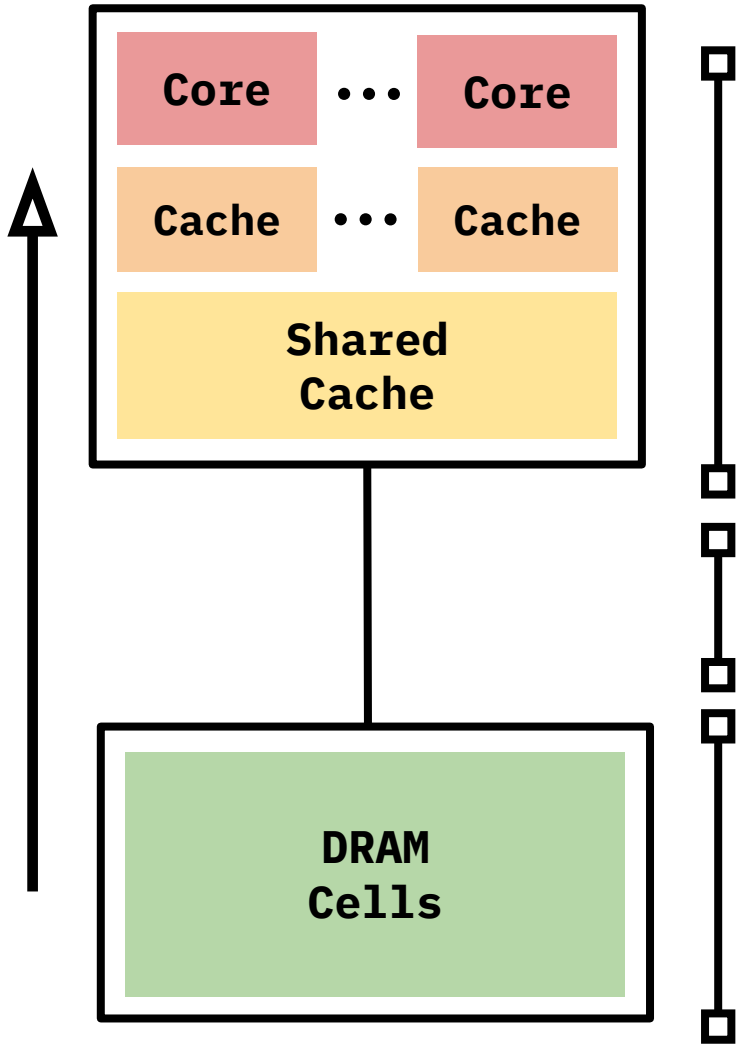
Monty[1] data movement:
12.5 - 125 GiB

Montyll[2] data movement:
87.5-175 GiB

Can multicore **CPUs** **handle** this **data movement**?

2500 Learning Modules:

~100 GiB per step



Logic

~ 100 MB

> 100 GB/s

Memory

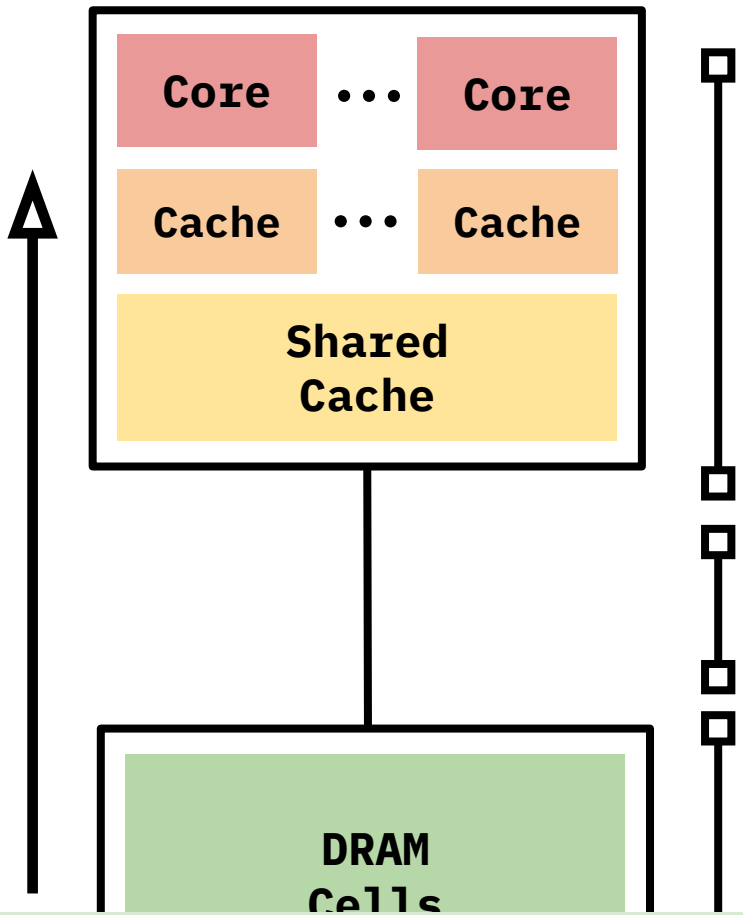
~ 100 GB

Overwhelmed caches: need to fetch data from main memory

Large memory bandwidth and high-performance prefetchers hide the memory latency

2500 Learning Modules:

~100 GiB per step



Logic

~ 100 MB

Overwhelmed caches: need to fetch data from main memory

> 100 GB/s

Large memory bandwidth and high-performance prefetchers hide the memory latency

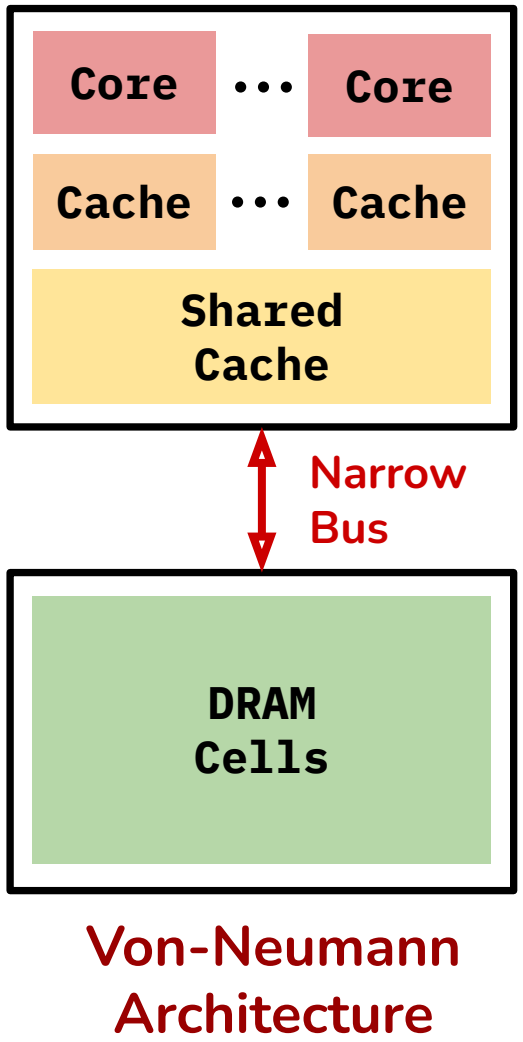
Memory

→ CPUs are a good candidate to scale up Thousand Brains Systems

❸ Processing-in-Memory[1] (PiM)

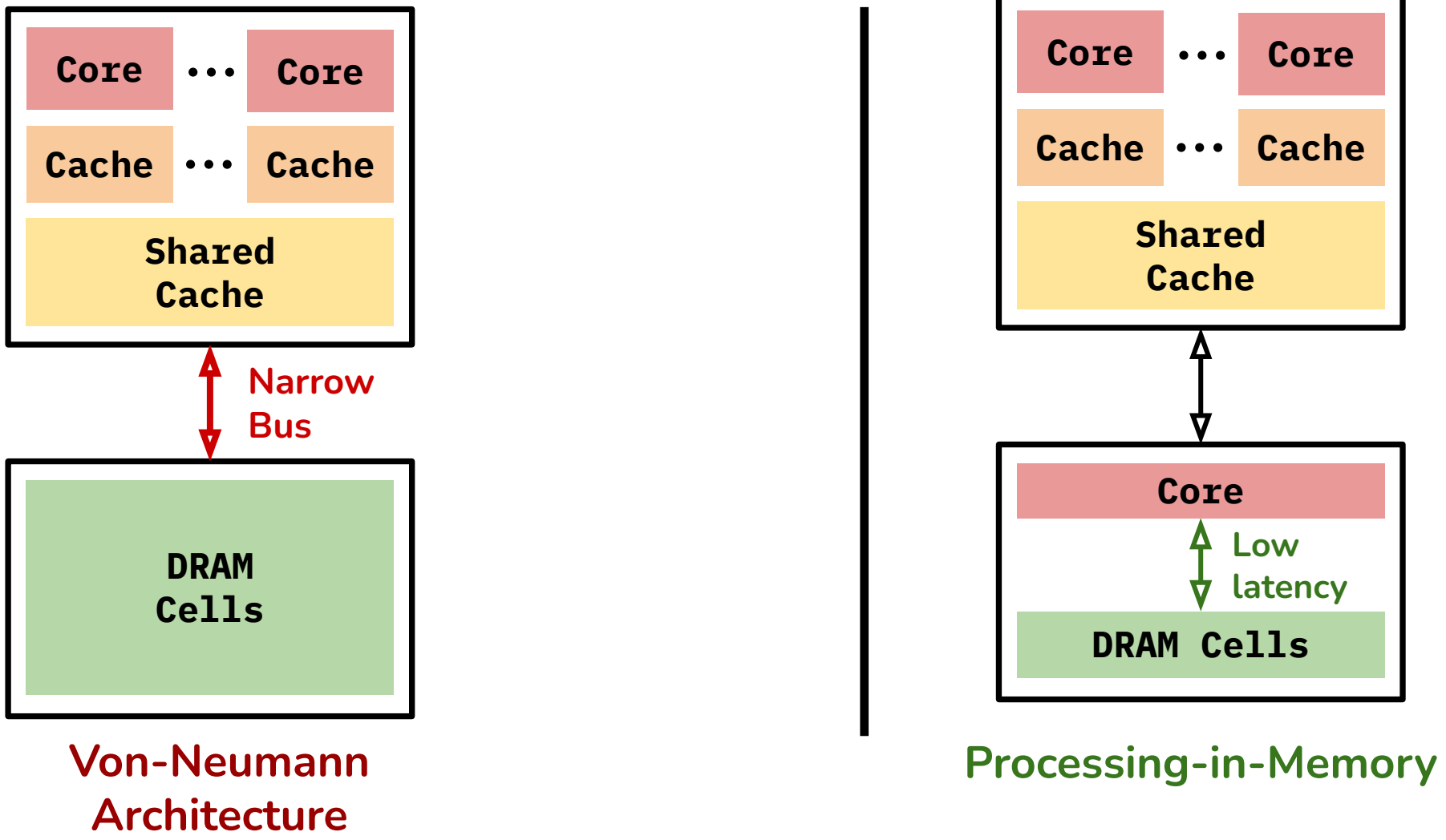
1: Mutlu, Onur, et al. "A modern primer on processing in memory." Emerging computing: from devices to systems: looking beyond Moore and Von Neumann. Singapore: Springer Nature Singapore, 2022. 171-243.

3 Processing-in-Memory[1] (PiM)



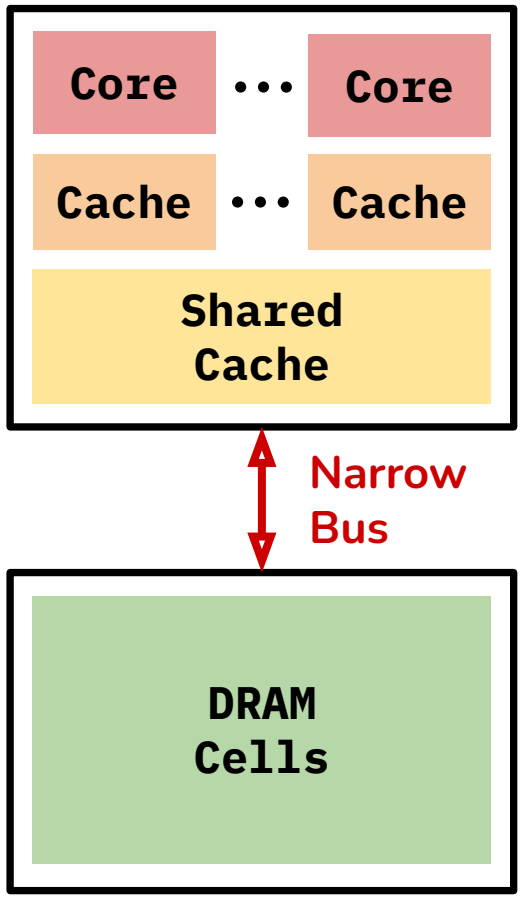
1: Mutlu, Onur, et al. "A modern primer on processing in memory." Emerging computing: from devices to systems: looking beyond Moore and Von Neumann. Singapore: Springer Nature Singapore, 2022. 171-243.

3 Processing-in-Memory[1] (PiM)



1: Mutlu, Onur, et al. "A modern primer on processing in memory." Emerging computing: from devices to systems: looking beyond Moore and Von Neumann. Singapore: Springer Nature Singapore, 2022. 171-243.

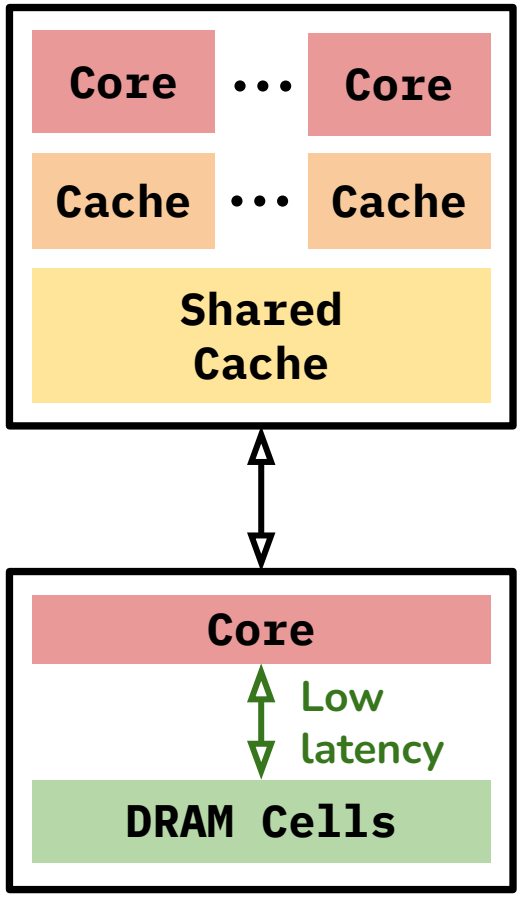
3 Processing-in-Memory[1] (PiM)



Von-Neumann Architecture

2500 Learning Modules:

100 GiB per step

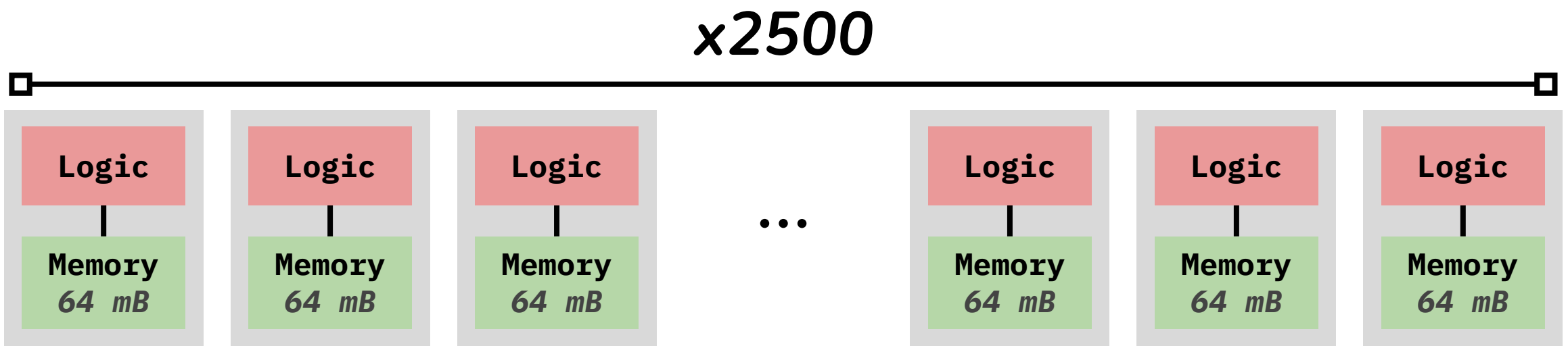


Processing-in-Memory

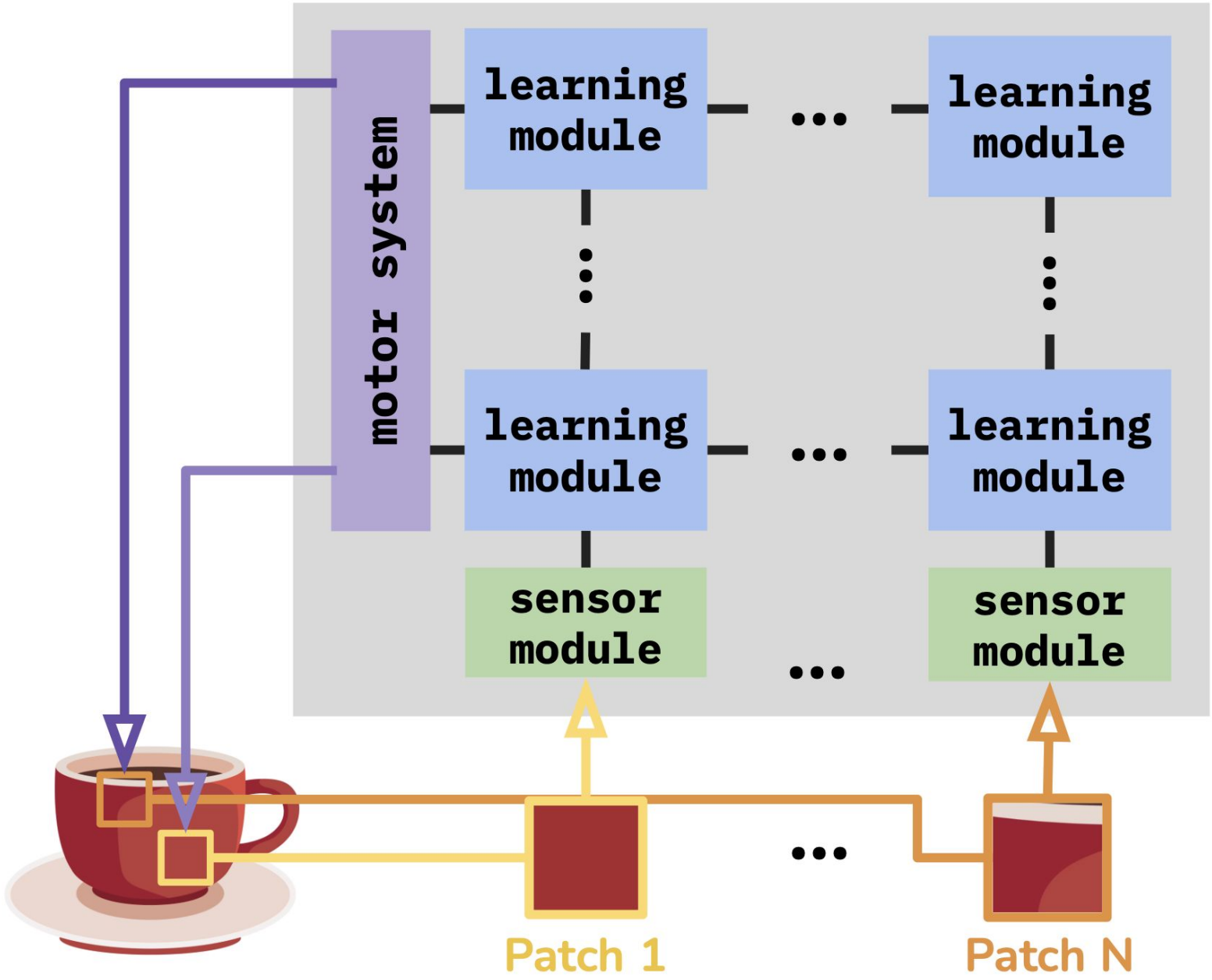
Constant cost:

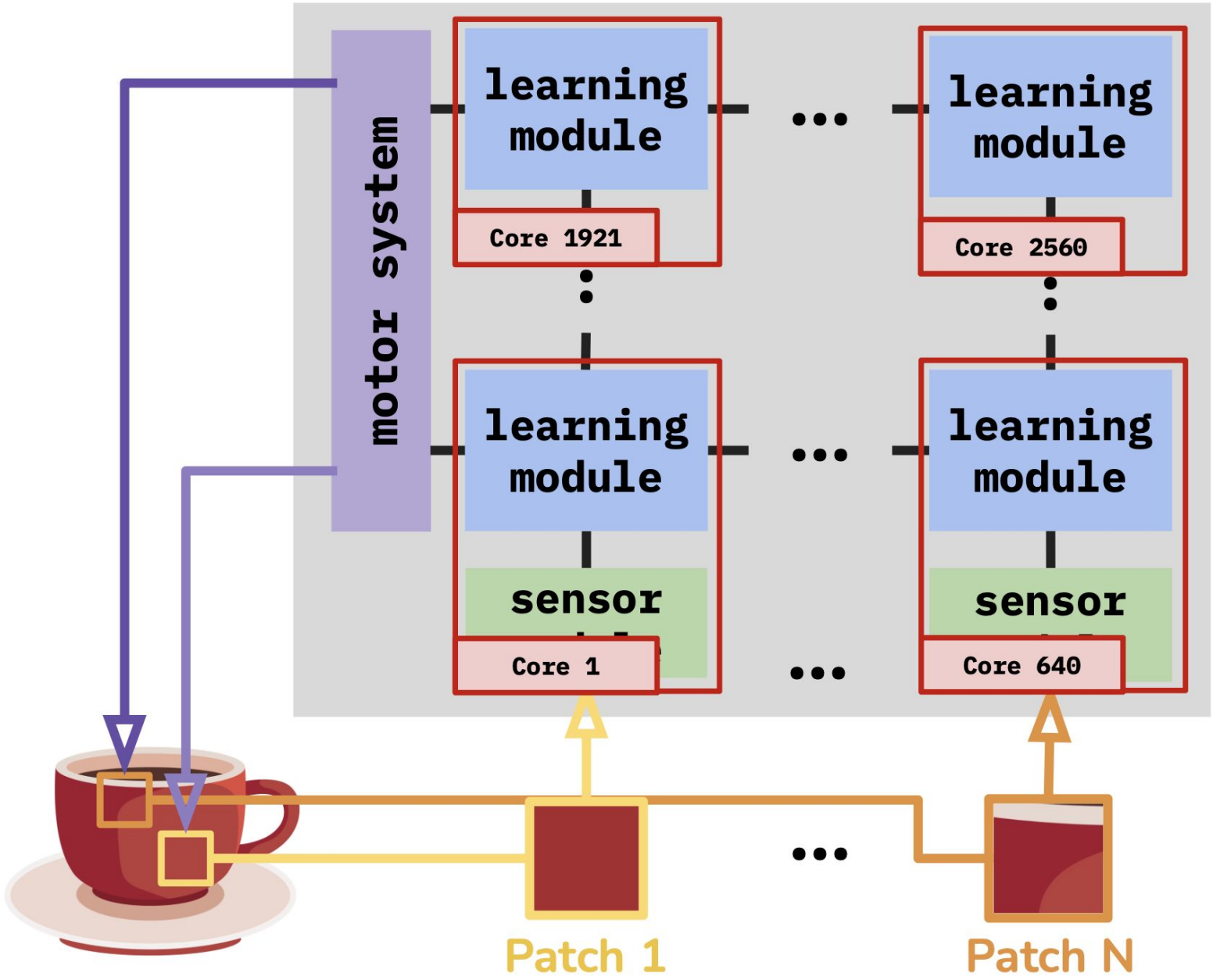
50 MiB per PiM Core per step

1: Mutlu, Onur, et al. "A modern primer on processing in memory." Emerging computing: from devices to systems: looking beyond Moore and Von Neumann. Singapore: Springer Nature Singapore, 2022. 171-243.

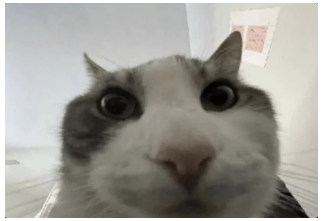

x2500
Logic
Memory
64 mB
Logic
Memory
64 mB
Logic
Memory
64 mB
...
Logic
Memory
64 mB
Logic
Memory
64 mB
Logic
Memory
64 mB

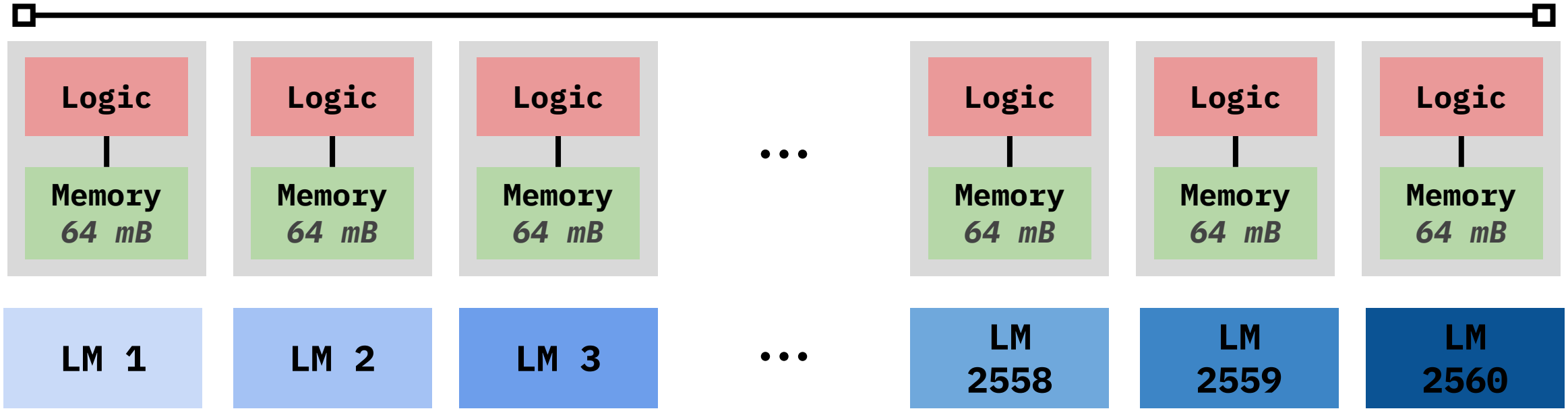
50


motor system
learning module
learning module
learning module
learning module
learning module
learning module
sensor module
sensor module
sensor module
Patch 1
Patch N


motor system
learning module
learning module
learning module
learning module
learning module
learning module
Core 1921
Core 2560
sensor
sensor
Core 1
Core 640
Patch 1
Patch N

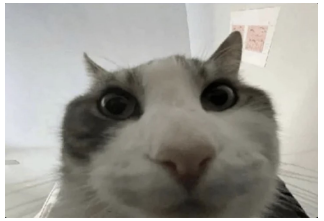
Cat cortex-scale system
2500 learning modules

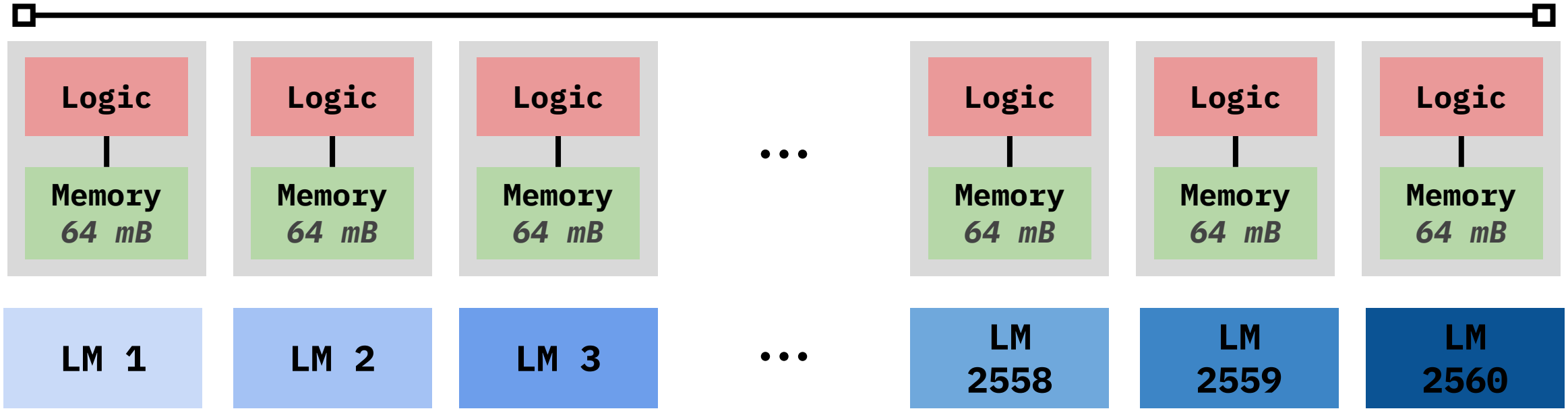




Gómez-Luna, Juan, et al. "Benchmarking a new paradigm: An experimental analysis of a real processing-in-memory architecture." arXiv preprint arXiv:2105.03814 (2021).

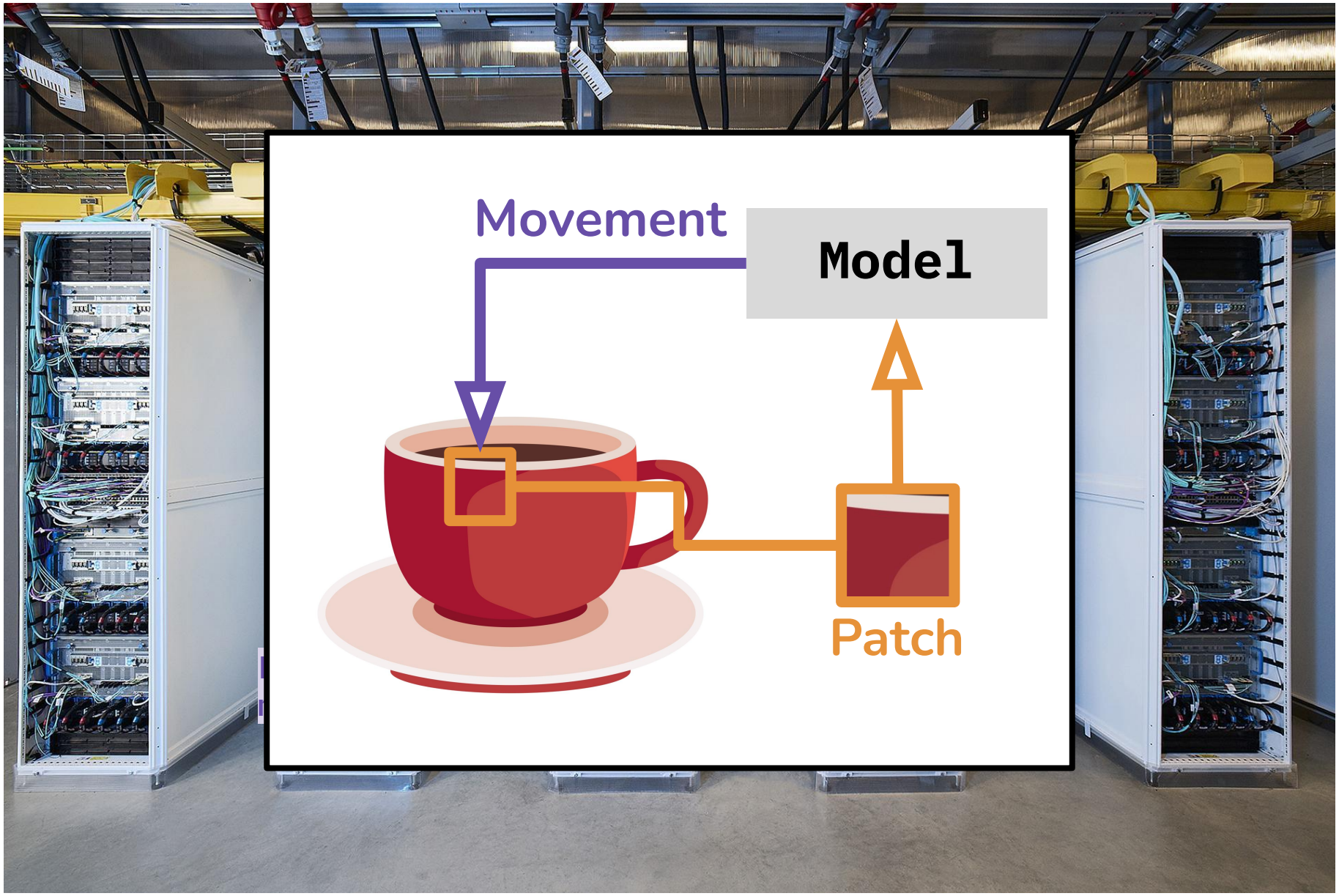
Cat cortex-scale system
2500 learning modules



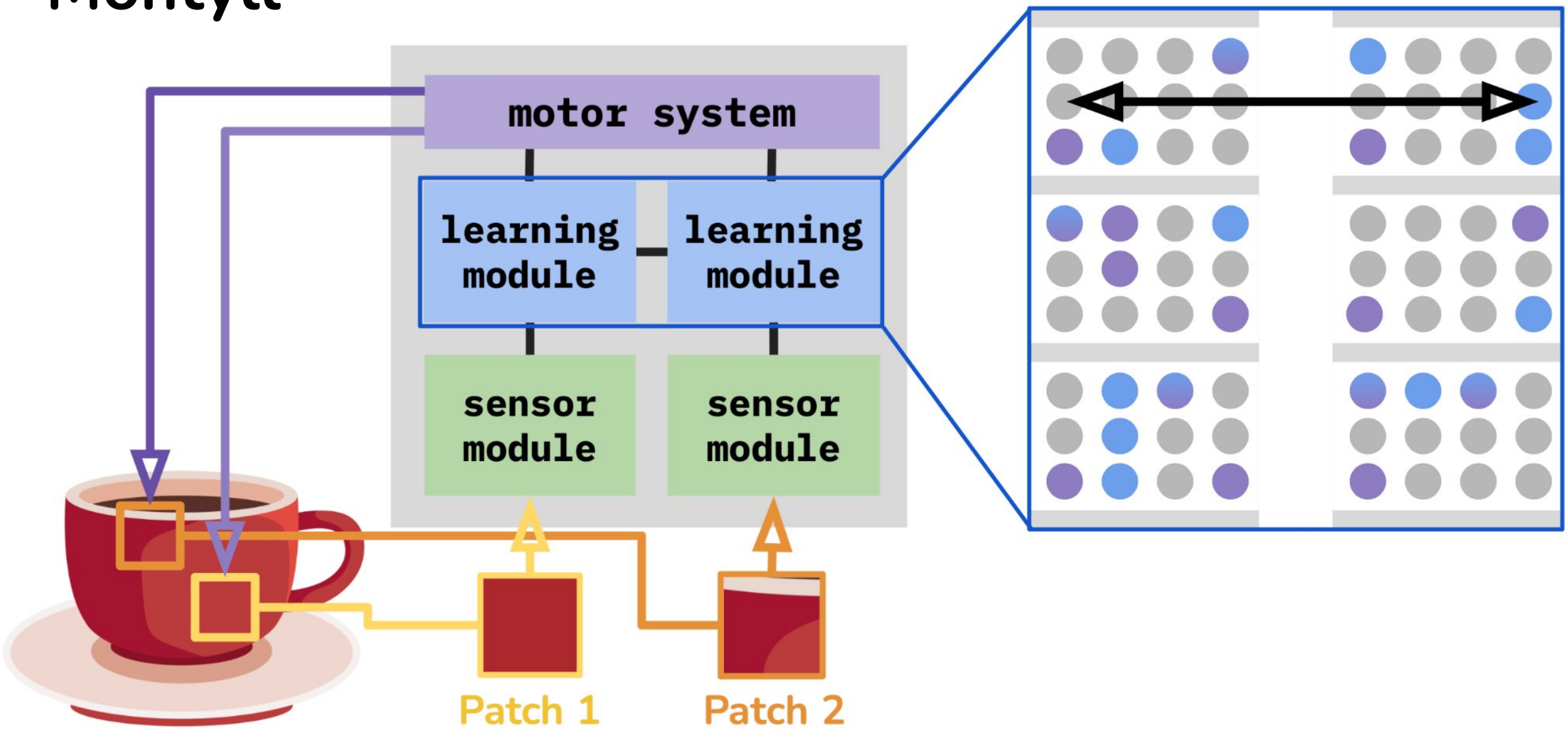


→ PiM is a good candidate to scale up Thousand Brains Systems

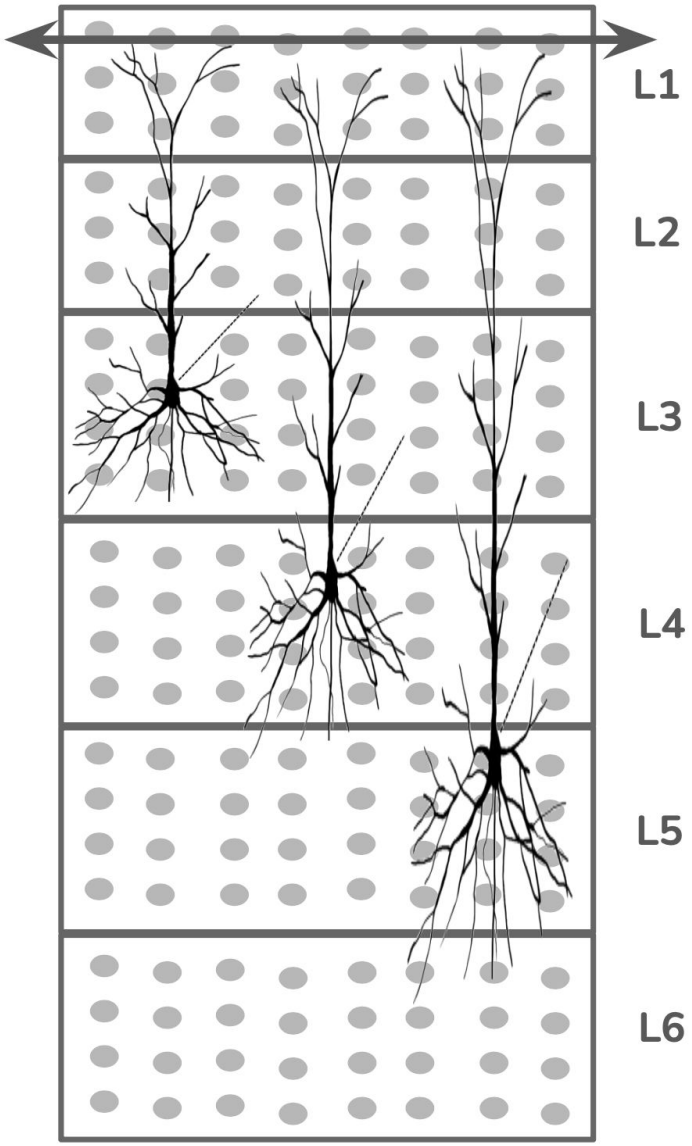

Learning module 1
Learning module 2
Learning module 3
Learning module 4
Learning module 5
...


Movement
Model
Patch

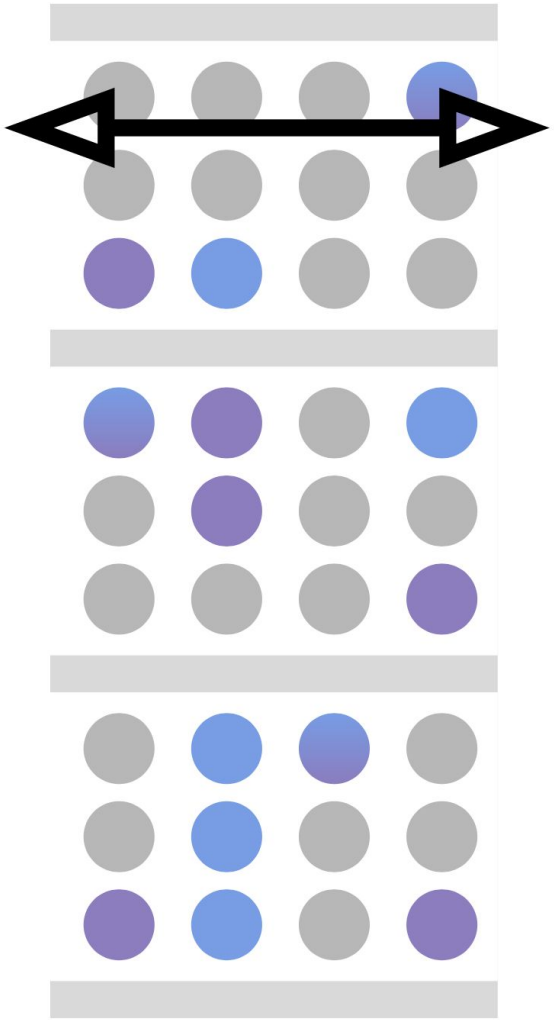
Montyll


motor system
learning module
learning module
sensor module
sensor module
Patch 1
Patch 2

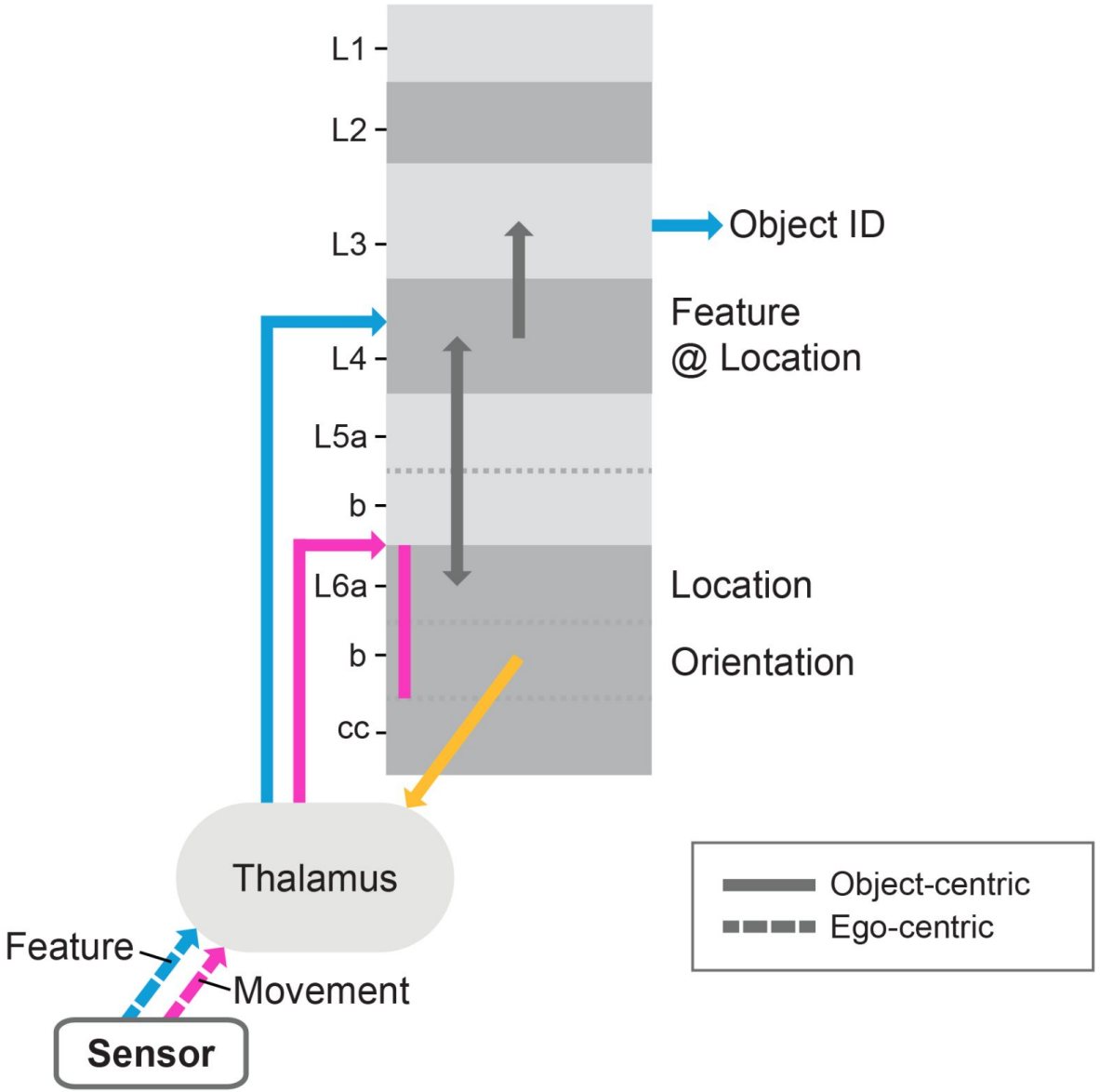
Cortical Column



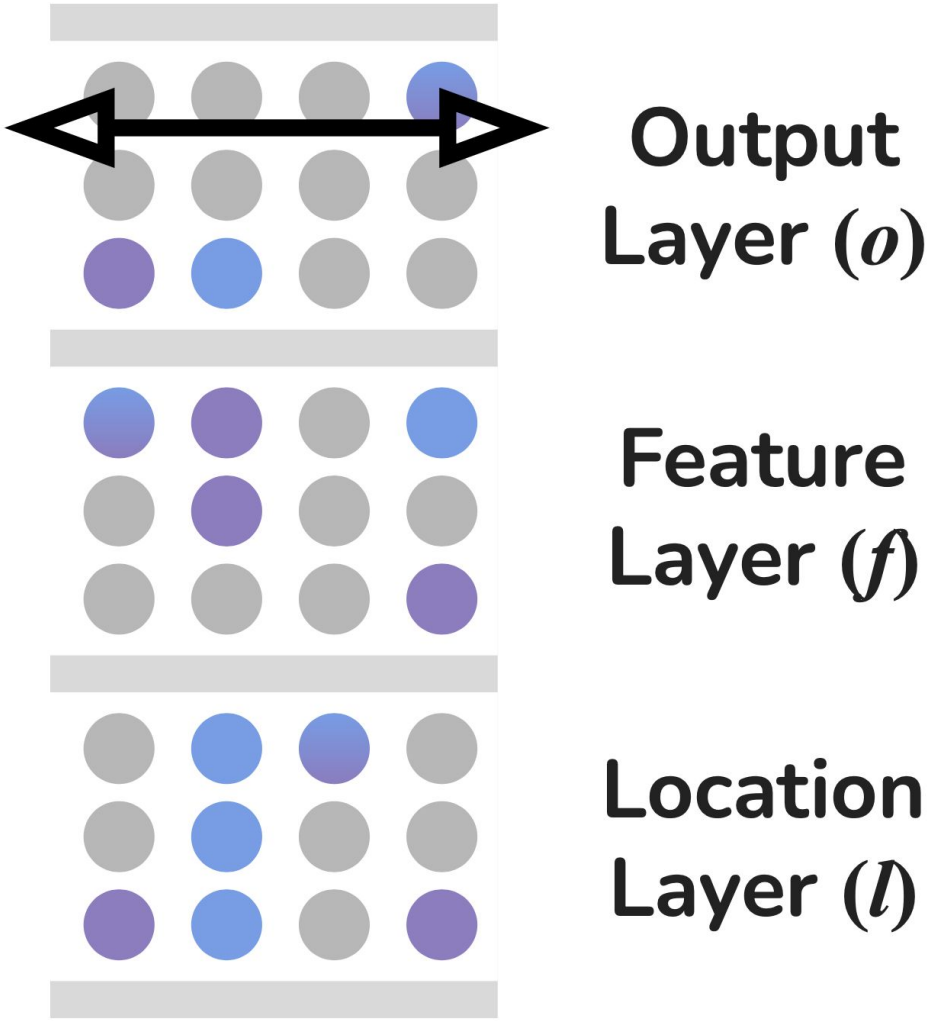
Learning Module



Cortical Column

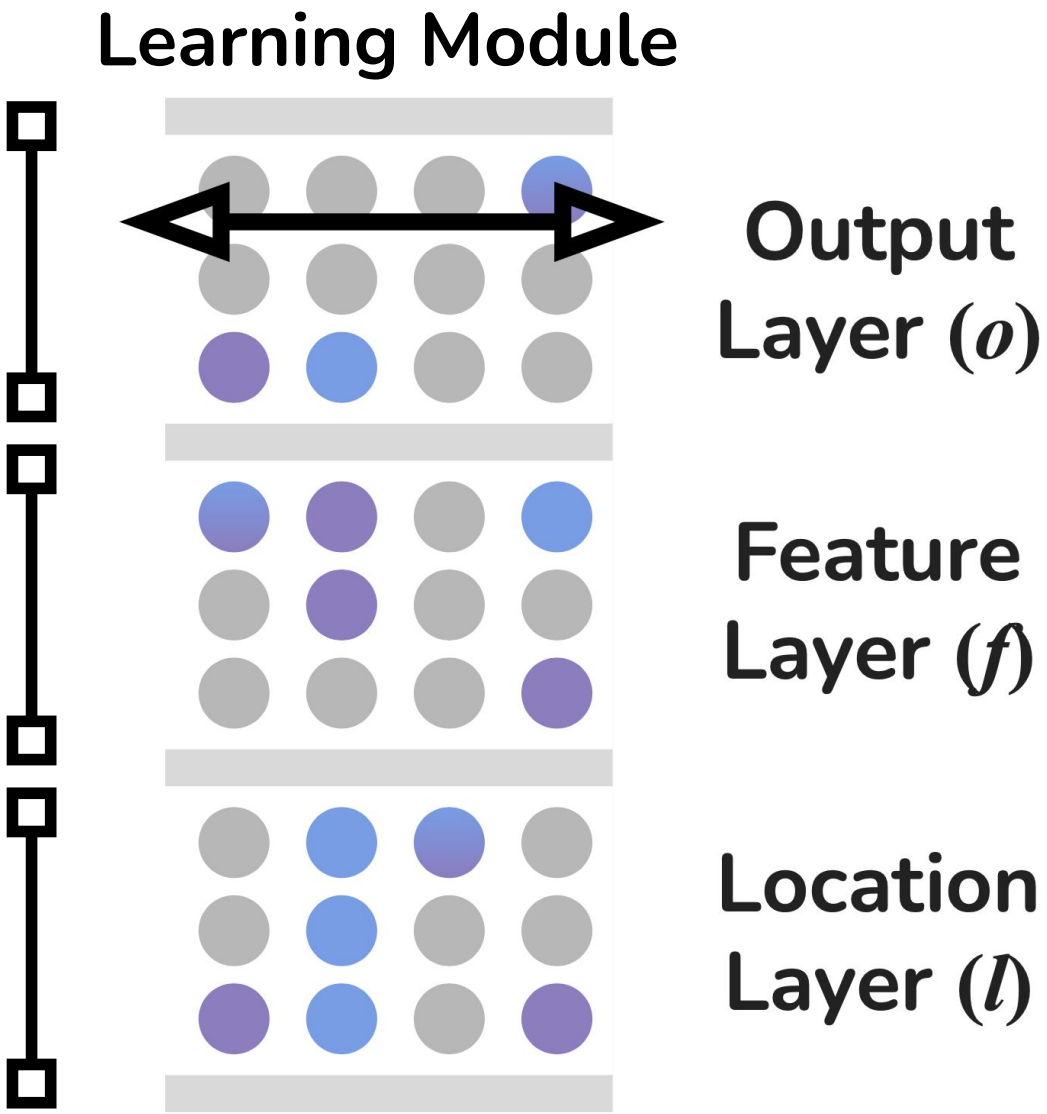


Learning Module

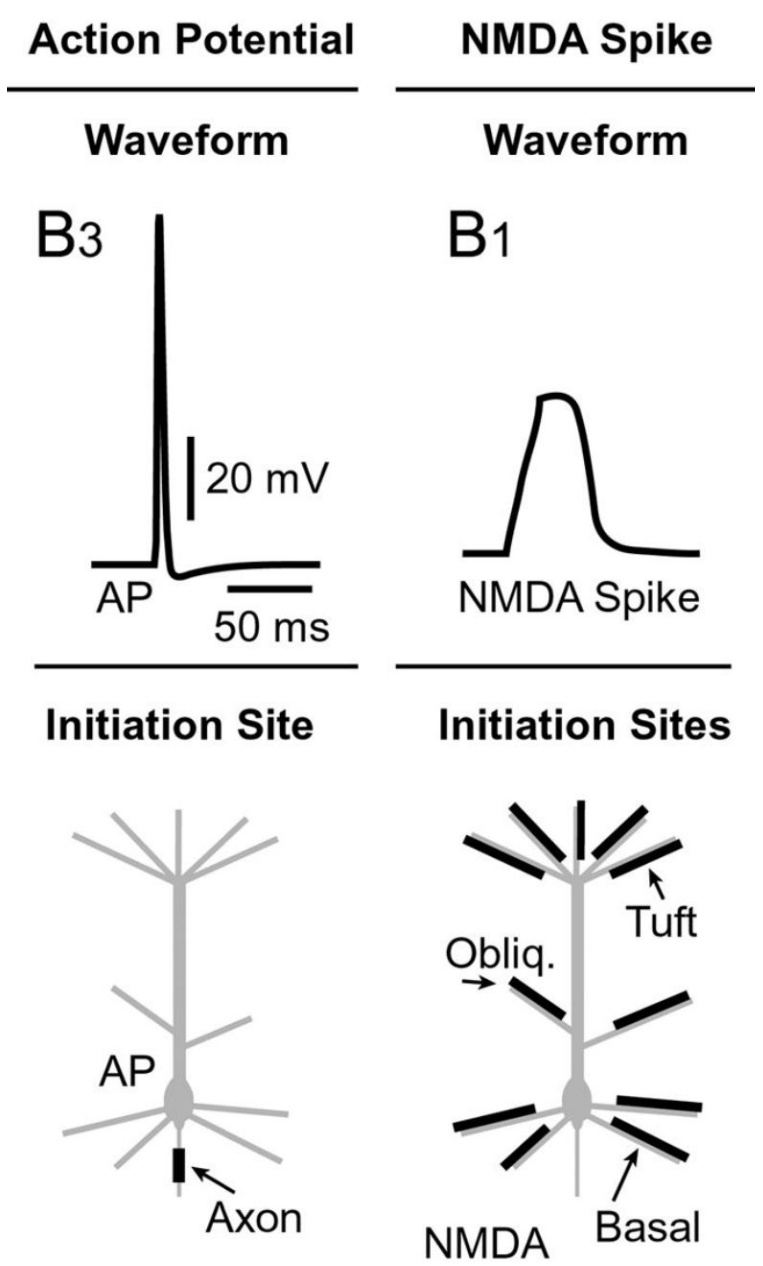


[1] Hawkins, Jeff, Niels Leadholm, and Viviane Clay. "Hierarchy or Heterarchy? A Theory of Long-Range Connections for the Sensorimotor Brain." arXiv preprint arXiv:2507.05888 (2025).

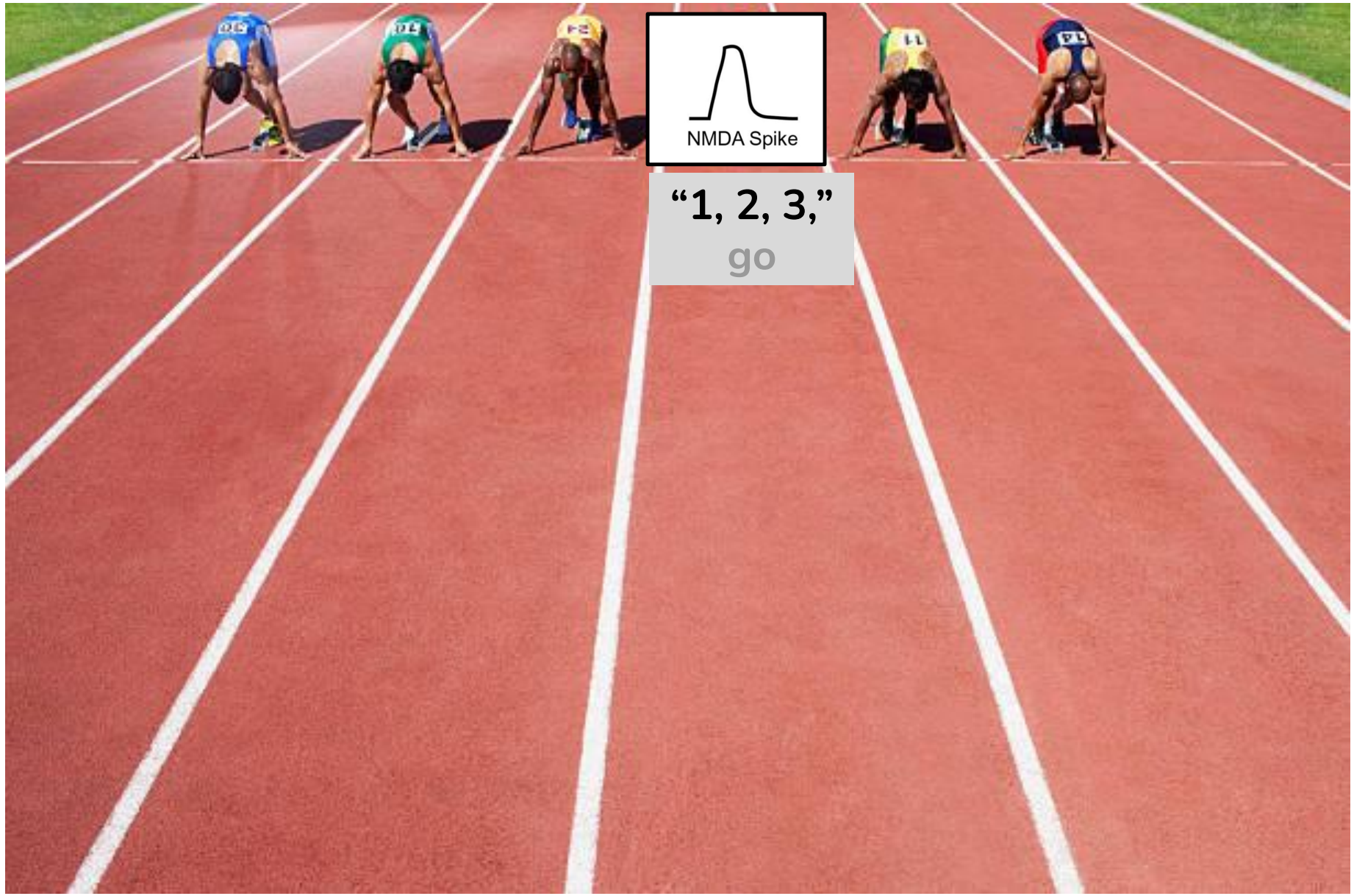
Hierarchical-Temporal Memory (HTM) Networks[1'2'3'4]



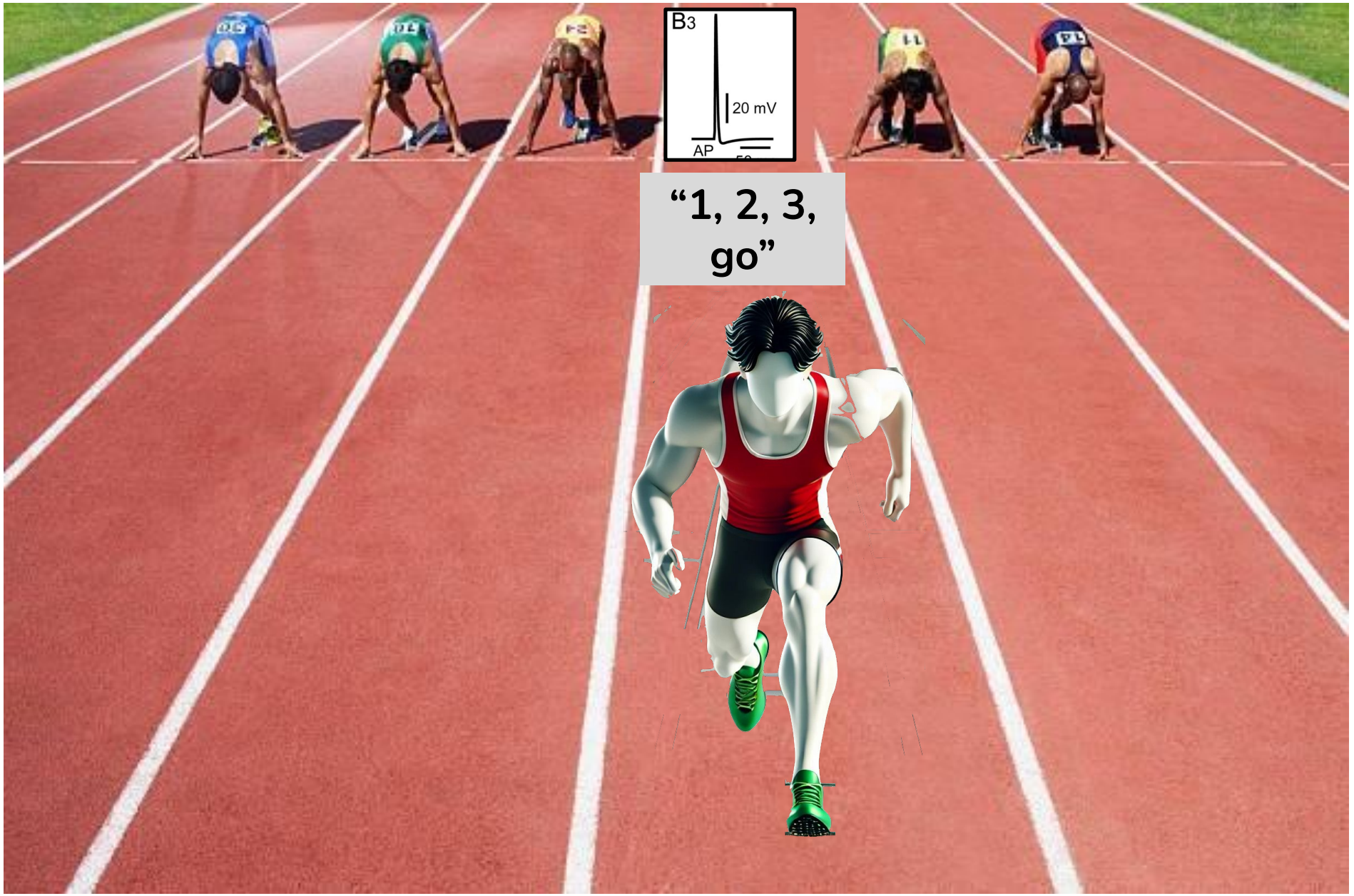
[1] Hawkins, Jeff, and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *Frontiers in neural circuits* 10 (**2016**): 174222.
[2] Cui, Yuwei, Subutai Ahmad, and Jeff Hawkins. "The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding." Frontiers in computational neuroscience 11 (2017): 272195.
[3] Ahmad, Subutai, et al. "Unsupervised real-time anomaly detection for streaming data." Neurocomputing 262 (2017): 134-147.
[4] Cui, Yuwei, Subutai Ahmad, and Jeff Hawkins. "Continuous online sequence learning with an unsupervised neural network model." Neural computation 28.11 (2016): 2474-2504.



1: London, Michael, and Michael Häusser. "Dendritic computation." *Annu. Rev. Neurosci.* 28.1 (2005): 503-532.
2: Antic, Srdjan D., et al. "The decade of the dendritic NMDA spike." *Journal of neuroscience research* 88.14 (2010): 2991-3001.
3: Major, Guy, Matthew E. Larkum, and Jackie Schiller. "Active properties of neocortical pyramidal neuron dendrites." *Annual review of neuroscience* 36.1 (2013): 1-24.
4: Hawkins, Jeff, and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *Frontiers in neural circuits* 10 (**2016**): 174222.



Hawkins, Jeff, and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *Frontiers in neural circuits* 10 (**2016**): 174222.



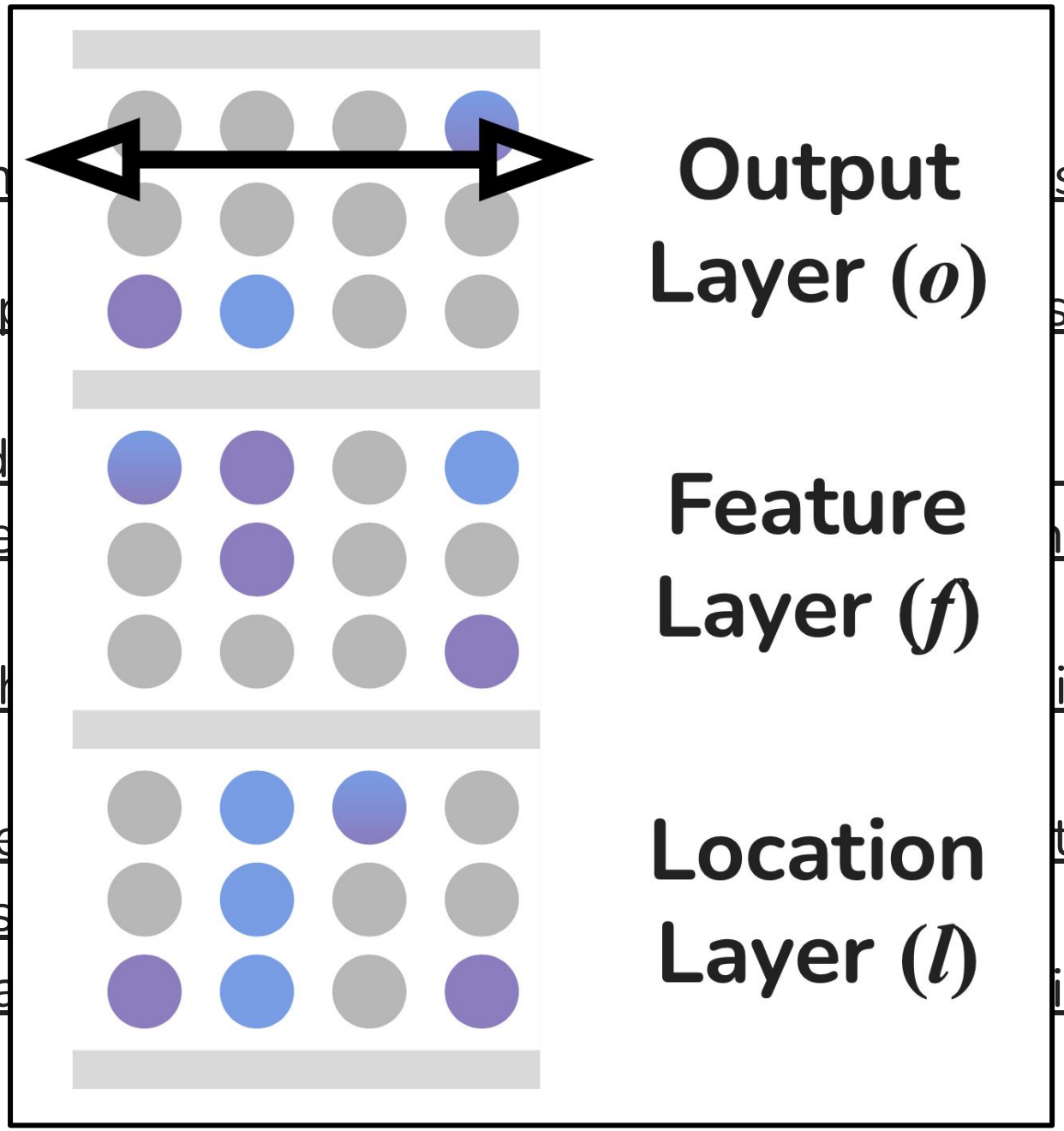
Hawkins, Jeff, and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *Frontiers in neural circuits* 10 (**2016**): 174222.

HTM Networks

2016: Hawkins et al. "Why neurons have thousands of synapses, a theory of sequence memory in the neocortex."
2017: Cui et al. "The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding."
2017: Ahmad et al. "Unsupervised real-time anomaly detection for streaming data."
2016: Cui et al. "Continuous online sequence learning with an unsupervised neural network model."
2017: Hawkins et al. "A theory of how columns in the neocortex enable learning the structure of the world."
2019: Lewis et al. "Locations in the neocortex: a theory of sensorimotor object recognition using cortical grid cells."
2020: Bennett, Max. "An attempt at a unified theory of the neocortical microcircuit in sensory cortex."

HTM Networks

2016: Hawkins et al. "Why neuron sequence memory in the neocortex."
2017: Cui et al. "The HTM spatial p sparse distributed coding."
2017: Ahmad et al. "Unsupervised data."
2016: Cui et al. "Continuous online neural network model."
2017: Hawkins et al. "A theory of h ing the structure of the world."
2019: Lewis et al. "Locations in the t recognition using cortical grid cells"
2020: Bennett, Max. "An attempt a ircuit in sensory cortex."



Property	**Feature (f)**	**Location (l)**	**Output (o)**	**Pooler (p)**
Layout	$M \times N$ cells	$M \times N$ cells	M cells	M cells
States				
Activation	$a_{ij}^{t[f]} \in \{0,1\}$	$a_{ij}^{t[l]} \in \{0,1\}$	$a_{i}^{t[o]} \in \{0,1\}$	$a_{i}^{t[p]} \in \{0,1\}$
Prediction	$\pi_{ij}^{t[f]} \in \{0,1\}$	$\pi_{ij}^{t[l]} \in \{0,1\}$	$\pi_{i}^{t[o]} \in \{0,1\}$	—
Spiking Segments	$\tau_{ij}^{t[f]}$	$\tau_{ij}^{t[l]}$	$\tau_{i}^{t[o]}$	—
Connections				
Feedforward	—	—	$\mathbf{F}_{i}^{t}$	$\mathbf{F}_{i}^{t}$
Contextual	$\mathbf{D}_{ijd}^{t}$	$\mathbf{D}_{ijd}^{t}$	$\mathbf{D}_{id}^{t}$	—
Incident Index (I)	(f,i,j)	(l,i,j)	(o,i)	(p,i)
Overlap	$\mu^{t_0}(\mathbf{G},p^*)$ counts active connections with concomitantly active presynaptic cells on connection structure $\mathbf{G}$ (where $\mathbf{G}$ is $\mathbf{F}$ or $\mathbf{D}$).			

Activation and Learning Rules

Mechanism	Mathematical Formulation	Remarks
Synaptic Overlap	$\mu^t(\mathbf{G}, p^*) = \lvert\{(I, p_t) \in \mathbf{G} \mid p_t \geq p^*, a_I^{t[x]} = 1\}\rvert$	$\mathbf{G}$ can be proximal ($\mathbf{F}$) or distal ($\mathbf{D}$). Counts active connections with active cells.
Segment NMDA Spike	$\tau_{ij}^t = \sum_d \mathbb{I}\{\mu^{t-1}(\mathbf{D}_{ijd}^t, p^*) \geq \theta_d\}$	**Distal Context:** Occurs locally on a segment when the contextual overlap μ exceeds threshold θ_d.
Cell Depolarization	$\pi_I^t = 1$ if Condition(τ_I^t)	**Feature/Location:** Primed if ≥ 1 segment spikes. **Output:** Primed if in Top-k cells by segment spike count.
Somatic Action Potential	$a_I^t = \text{Decision}(\mu^t(\mathbf{F}_I^t, p^*), \pi_I^{t-1})$	**Pooler:** Enough ffw overlap and competitive Top-k overlap ranking. **Feature/Location:** Bursting logic if no cell is depolarized. No ffw connection $\mathbf{F}^t$. **Output:** Enough ffw overlap and predicted.
Learning Rule	$p_{t+1} = p_t + p^+$ if incident $a_I^{t[x]} = 1$, else $p_t - p^-$. This Hebbian update is applied to all segments in the update set X^t (correctly predicting segments or best-match segments during a burst).	

Algorithmic Complexity

Layer	Overlap (Prediction)	Weight Update (Learning)	Total Step Complexity
SM: Pooler	$O(MC)$	$O(MCs)$	$MC(1+s)$
LM: Feature	$O(MNSC)$	$O(MNSCs)$	$O(MNSC(1+s))$
LM: Location	$O(MNSC)$	$O(MNSCs)$	$O(MNSC(1+s))$
LM: Output	$O(MSC)$	$O(MSCs)$	$O(MSC(1+s))$

Space Complexity

Component	Pooler (p)	Feature (f)	Location (l)	Output (o)
States				
Activation (a)	$O(M)$	$O(MN)$	$O(MN)$	$O(M)$
Prediction (π)	—	$O(MN)$	$O(MN)$	$O(M)$
Connections				
Feedforward (F)	$O(MC)$	—	—	$O(MC)$
Contextual (D)	—	$O(MNSC)$	$O(MNSC)$	$O(MSC)$

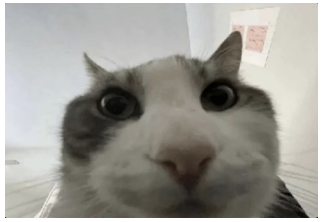
Montyll: Implementation

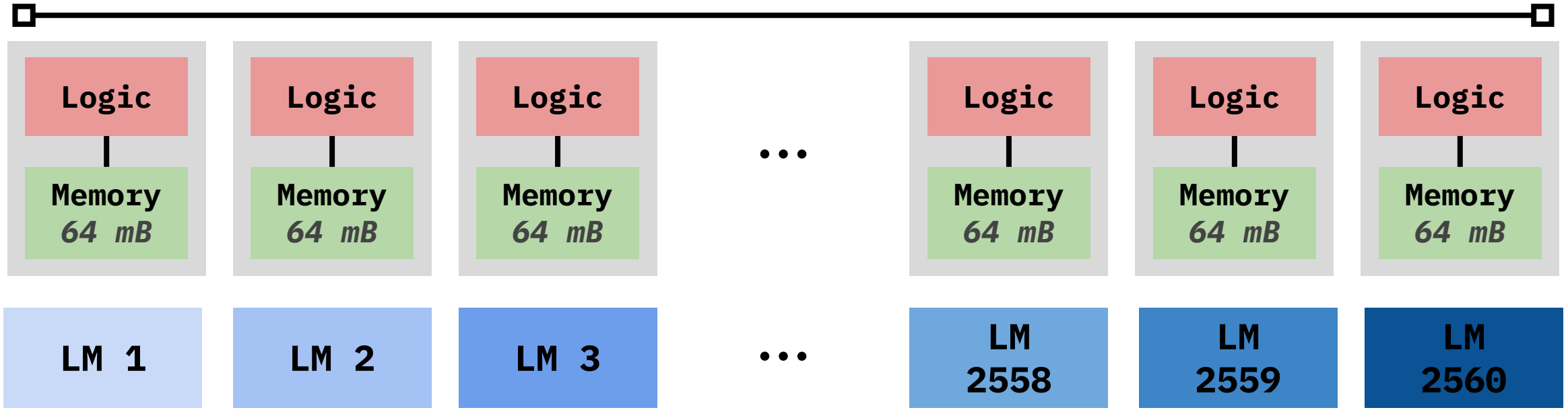
We write a high-performance **C implementation** of **Montyll** called `cmontyll`

Serves as the backbone of both the **CPU** and **PiM implementations**

https://github.com/Xavier0301/cmontyll

Cat cortex-scale system
2500 learning modules

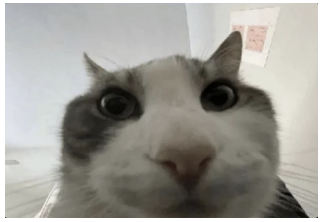


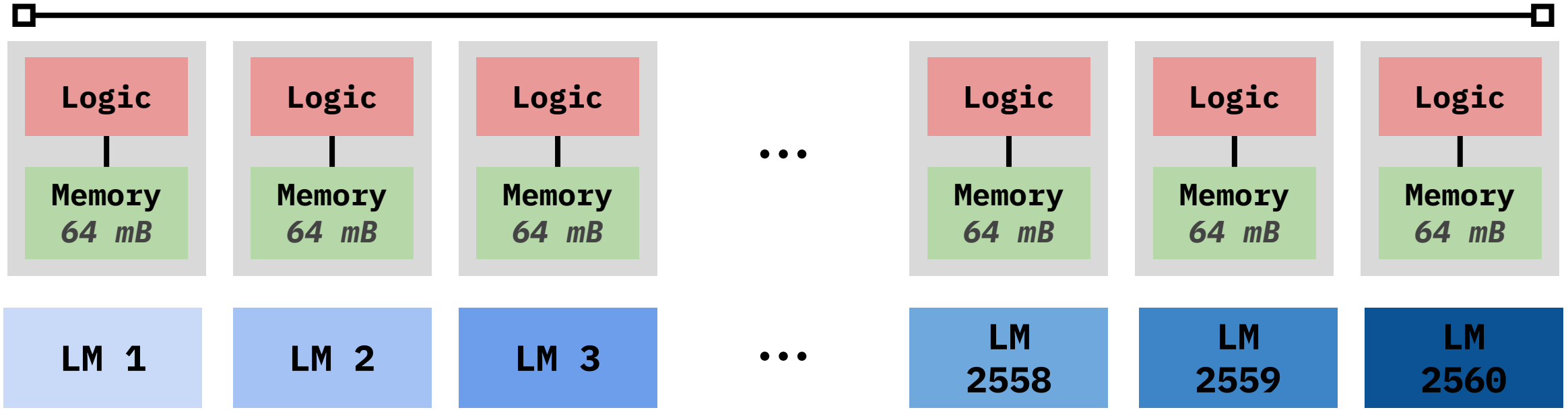


[1][2]Gómez-Luna, Juan, et al. "Benchmarking a new paradigm: An experimental analysis of a real processing-in-memory architecture." arXiv preprint (2021).
[1] **assuming big enough transfer chunks,** which is the case for this workload because of the underlying access pattern

71

Cat cortex-scale system
2500 learning modules



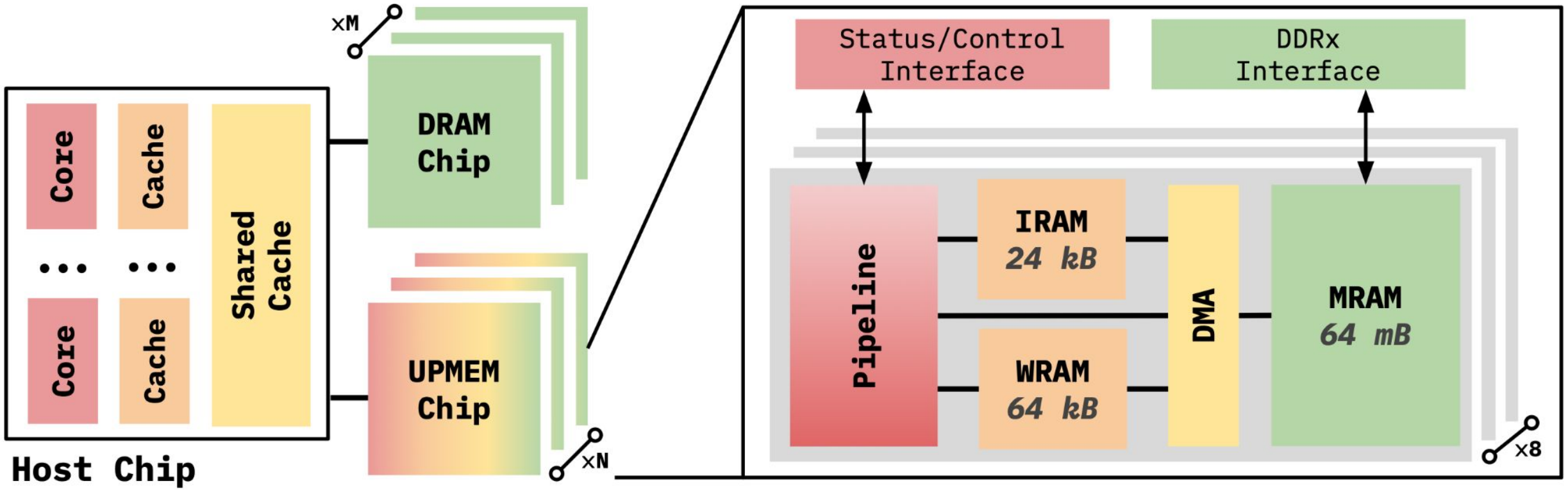


logic ↔ bank bandwidth[1]:
~500 MiB/s

logic frequency[2]:
400 MHz

[1][2]Gómez-Luna, Juan, et al. "Benchmarking a new paradigm: An experimental analysis of a real processing-in-memory architecture." arXiv preprint (2021).
[1] **assuming big enough transfer chunks,** which is the case for this workload because of the underlying access pattern

72



32-bit **integer** arithmetic

8-bit **integer** multiplication **only**

explicit **WRAM-MRAM** transfers

Mechanism	Mathematical Formulation	Remarks
Synaptic Overlap	$\mu^t(\mathbf{G}, p^*) = \lvert\{(I, p_t) \in \mathbf{G} \mid p_t \geq p^*, a_I^{t[x]} = 1\}\rvert$	$\mathbf{G}$ can be proximal ($\mathbf{F}$) or distal ($\mathbf{D}$). Counts active connections with active cells.
Segment NMDA Spike	$\tau_{ij}^t = \sum_d \mathbb{I}\{\mu^{t-1}(\mathbf{D}_{ijd}^t, p^*) \geq \theta_d\}$	**Distal Context:** Occurs locally on a segment when the contextual overlap μ exceeds threshold θ_d.
Cell Depolarization	$\pi_I^t = 1$ if Condition(τ_I^t)	**Feature/Location:** Primed if ≥ 1 segment spikes. **Output:** Primed if in Top-k cells by segment spike count.
Somatic Action Potential	$a_I^t = \text{Decision}(\mu^t(\mathbf{F}_I^t, p^*), \pi_I^{t-1})$	**Pooler:** Enough ffw overlap and competitive Top-k overlap ranking. **Feature/Location:** Bursting logic if no cell is depolarized. No ffw connection $\mathbf{F}^t$. **Output:** Enough ffw overlap and predicted.
Learning Rule	$p_{t+1} = p_t + p^+$ if incident $a_I^{t[x]} = 1$, else $p_t - p^-$. This Hebbian update is applied to all segments in the update set X^t (correctly predicting segments or best-match segments during a burst).	

https://github.com/Xavier0301/montyll-pim

WRAM
64 kB

States

Activation $a_{ij}^{t[f]} \in \{0, 1\}$

Prediction $\pi_{ij}^{t[f]} \in \{0, 1\}$

Activations (a) and **predictions (π)** in the network are packed into `32-bit` integers (e.g. `0110010110`) and accessed through bit-serial computation

https://github.com/Xavier0301/cmontyll

MRAM
64 mB

Connections	
Feedforward	$\mathbf{F}_i^t$
Contextual	$\mathbf{D}_{ijd}^t$
Incident Index (I)	(f, i, j)

Context (D) and **Feedforward (F)** connections, which are highly sparse matrices, are compressed using a sparse representation, where each entry is packed into a `32-bit struct` storing both the **incident cell index (I)** and the **permanence (p)**

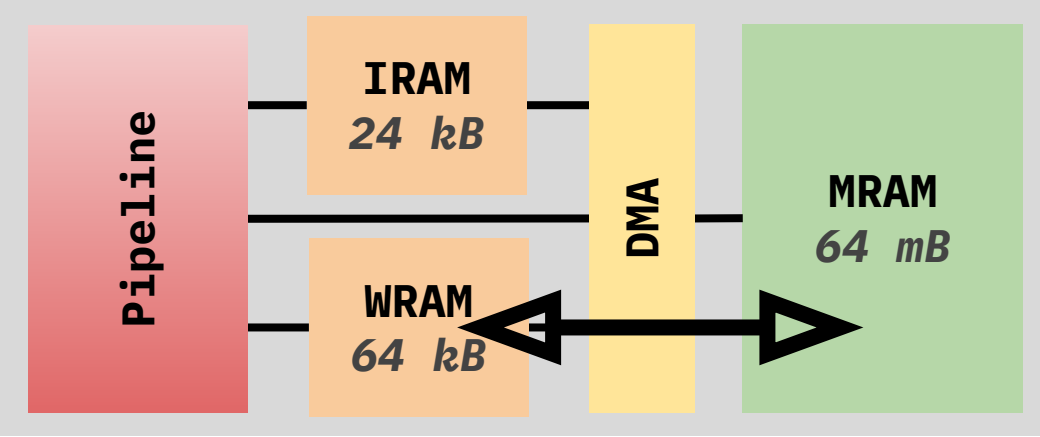
Boosting $b_i = e^{-\beta(A_i^t - \overline{A^t})}$

No **floating-point unit**, therefore the **boosting factors** are calculated using a combination of **fixed-point arithmetic** and **hybrid dynamic-precision lookup tables**

https://github.com/Xavier0301/montyll-pim

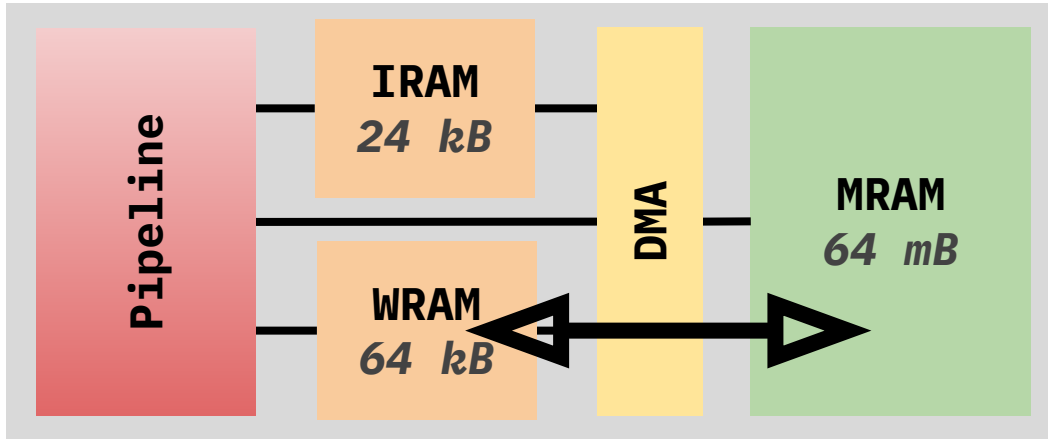
Connections transfer

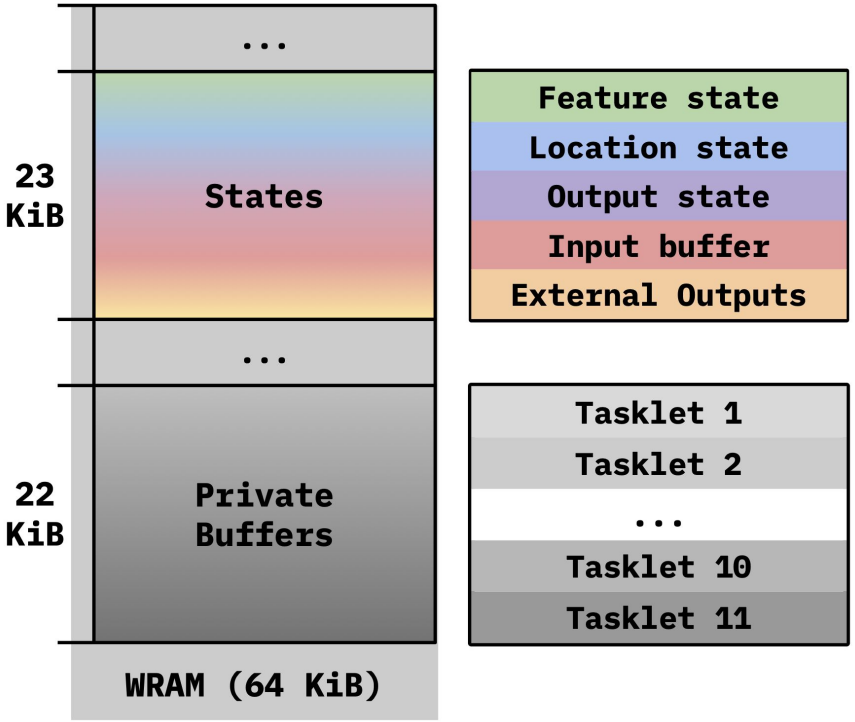
Connections are brought from **MRAM** to **WRAM** in blocks of ~2KB for **maximal transfer bandwidth**



https://github.com/Xavier0301/montyll-pim

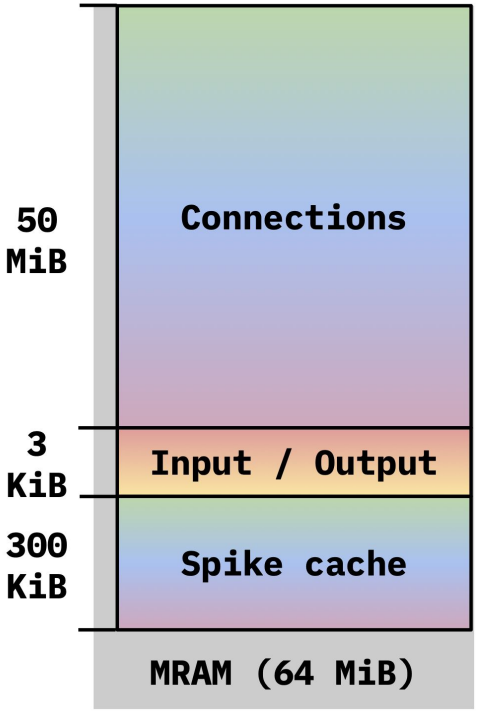
Connections transfer





Feature state
Location state
Output state
Input buffer
External Outputs

Tasklet 1
Tasklet 2
...
Tasklet 10
Tasklet 11



Feature ctx
Location ctx
Output ctx
Output ffw

Input movement
Input feature
External output layer activity

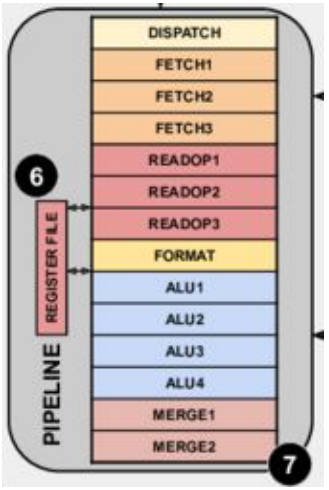
Feature spike cache
Location spike cache
Output spike cache

https://github.com/Xavier0301/montyll-pim

Tasklet-level parallelism

To fill the **fine-grained multi-threaded** (**FGMT**) **pipeline**, we need to **parallelize** the work into at least **11 tasklets** (**threads**)

Pipeline

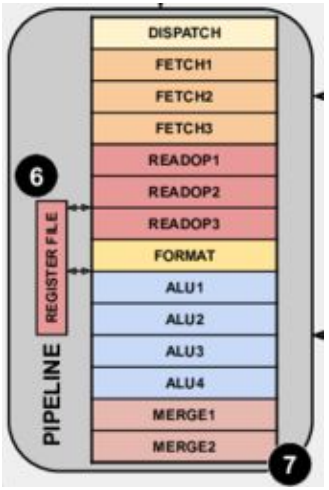


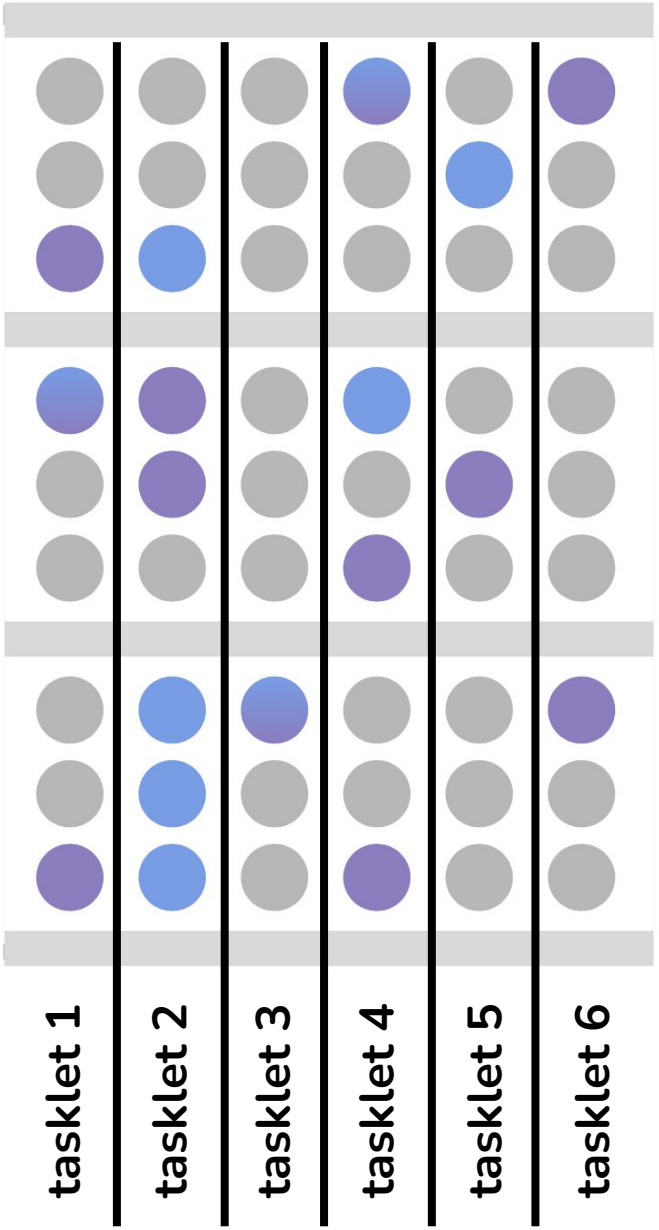
https://github.com/Xavier0301/montyll-pim

Tasklet-level parallelism

To fill the **fine-grained multi-threaded (FGMT) pipeline**, we need to **parallelize** the work into at least **11 tasklets (threads)**

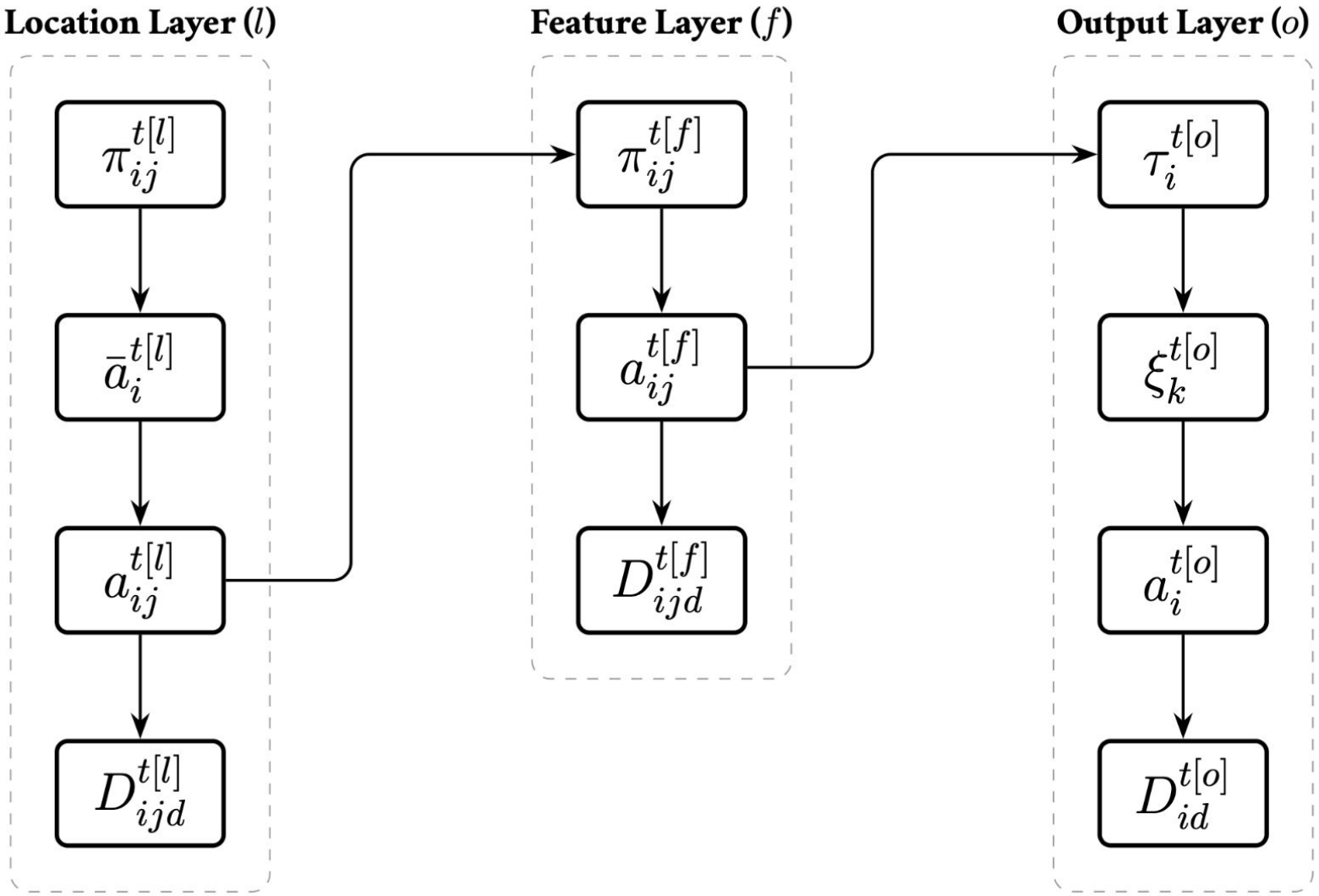
Pipeline





https://github.com/Xavier0301/montyll-pim

Barriers & synchronization



https://github.com/Xavier0301/montyll-pim

Methodology

Computing systems

Platform	Year	Cores	Freq.	LLC	Mem. BW	Mem. Cap.
CPU1 [Low] M2 Pro 12-CPU	2023	12 (8P + 4E)	2.8/3.5† GHz	36 MB (L2)	200 GB/s	16 GB
CPU2 [High] AMD EPYC 7763	2021	48‡	2.45 GHz	256 MB (L3)	204.8 GB/s	128 GB
PiM UPMEM v1B	2021	2,560 DPUs	400 MHz	–	600 MB/s per DPU	64 MB per DPU

† 8 P Cores at 3.5 GHz, 4 E Cores at 2.8 GHz. ‡ We use 48 of the 64 cores on the machine.

Computing systems

For the **PiM system**, we use the **uPIMulator** [1] **cycle-level simulator**, with the following parameters

DRAM System	
MRAM size	64 MB
DDR specification	DDR4-2400 [121]
Memory scheduling policy	FR-FCFS
Row buffer size	1 KB
Timings ($t_{RCD}, t_{RAS}, t_{RP}, t_{CL}, t_{BL}$)	16, 39, 16, 16, 4 cycles

We test our implementation using the **UPMEM functional simulator** [2]

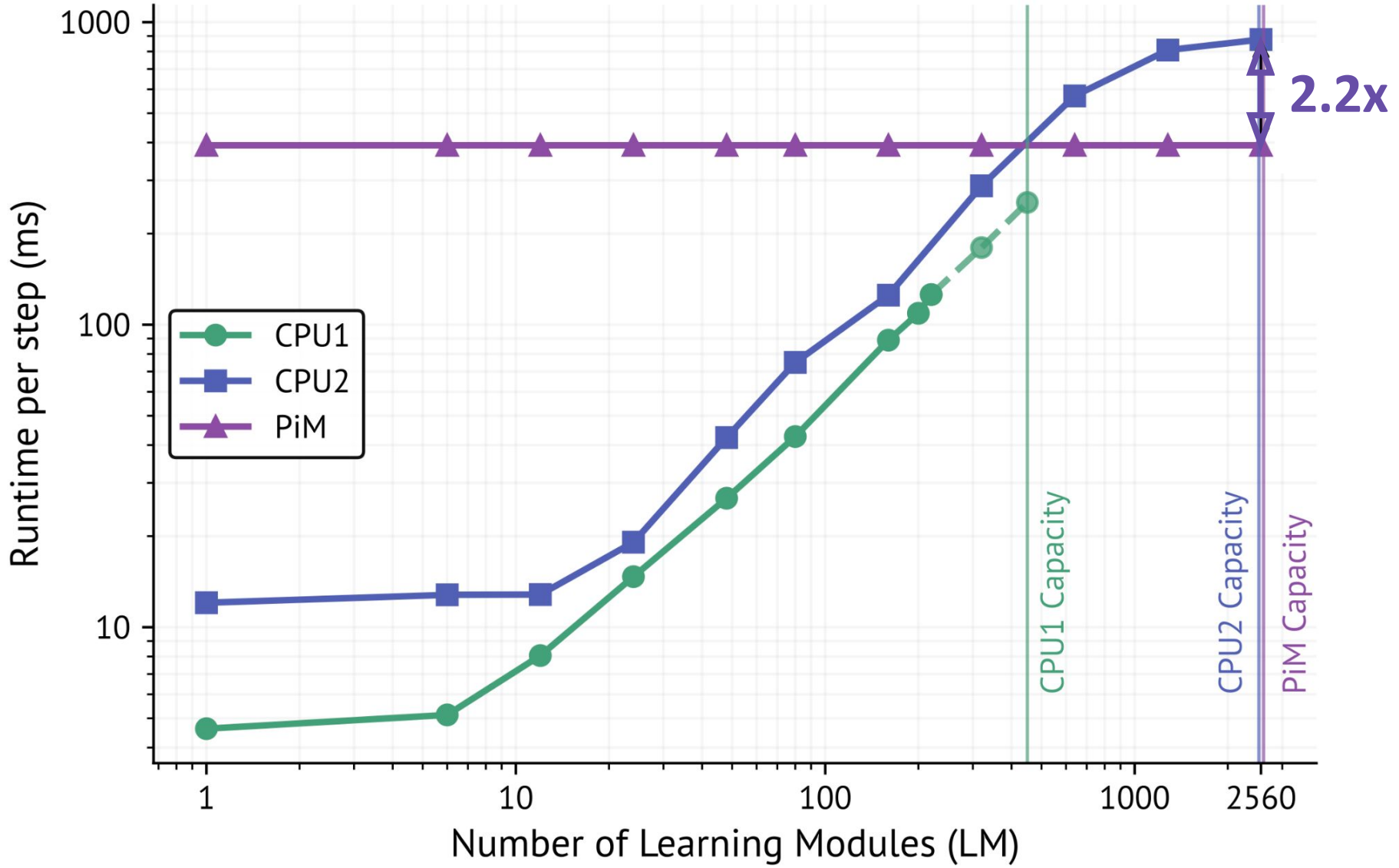
[1] Hyun, Bongjoon, et al. "Pathfinding future pim architectures by demystifying a commercial pim technology." *2024 HPCA*. IEEE, 2024.
[2] sdk.upmem.com
[121] Samsung Electronics, 8Gb C-die DDR4 SDRAM x16, K4A8G165WC, Rev. 1.5, Samsung Semiconductor, Apr. 2017

Results

Montyll: Scaling

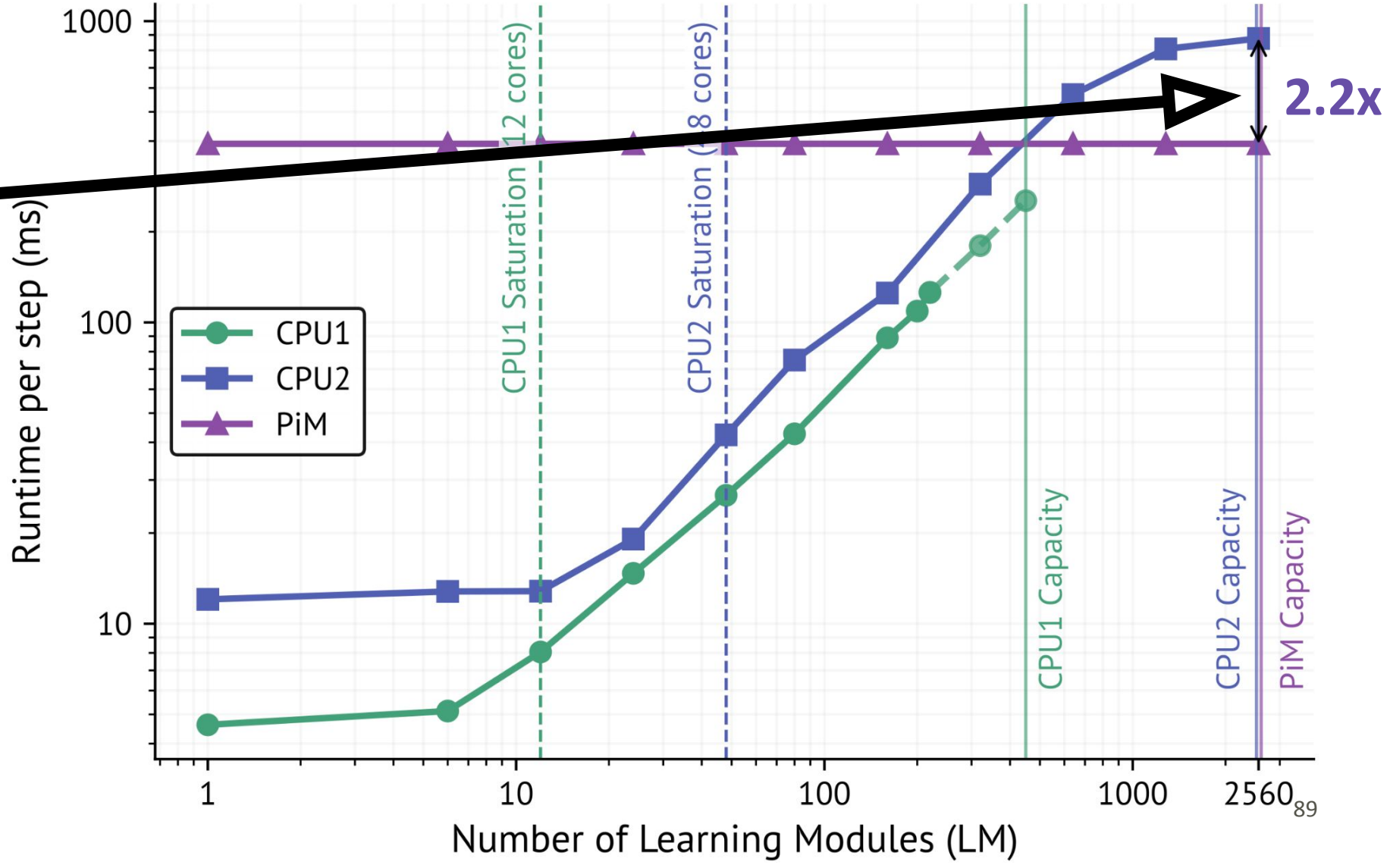
We **successfully scale Montyll** to **2560 learning modules** on **CPU2** and **PiM**, representing **44.5M neurons** and **16.1B synapses**

Montyll: Time per Step



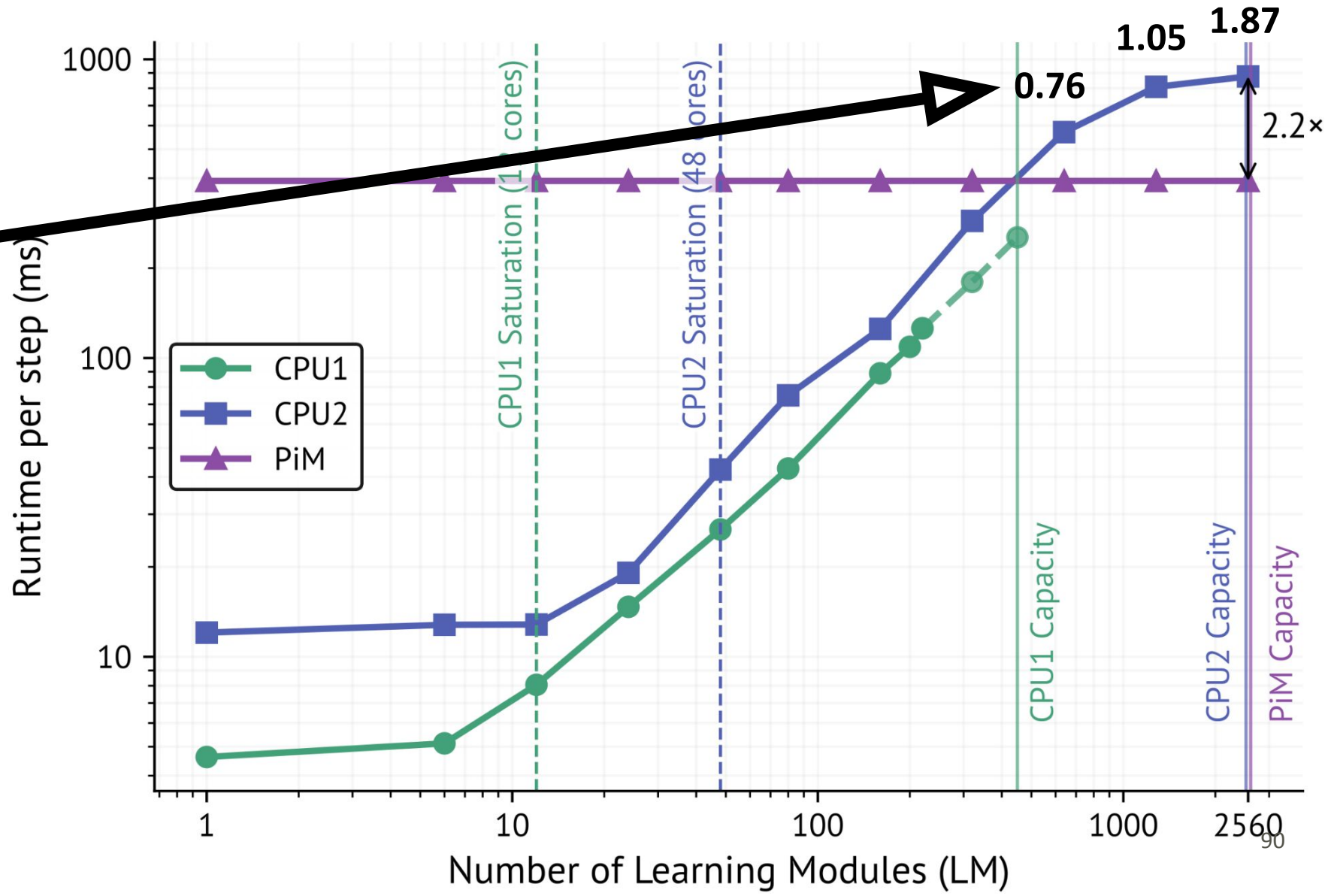
Montyll: Time per Step

PiM outperforms CPU2 by **2.2x** at 2560 LMs, with **2.55 vs. 1.14 steps/second**



Montyll: Time per Step

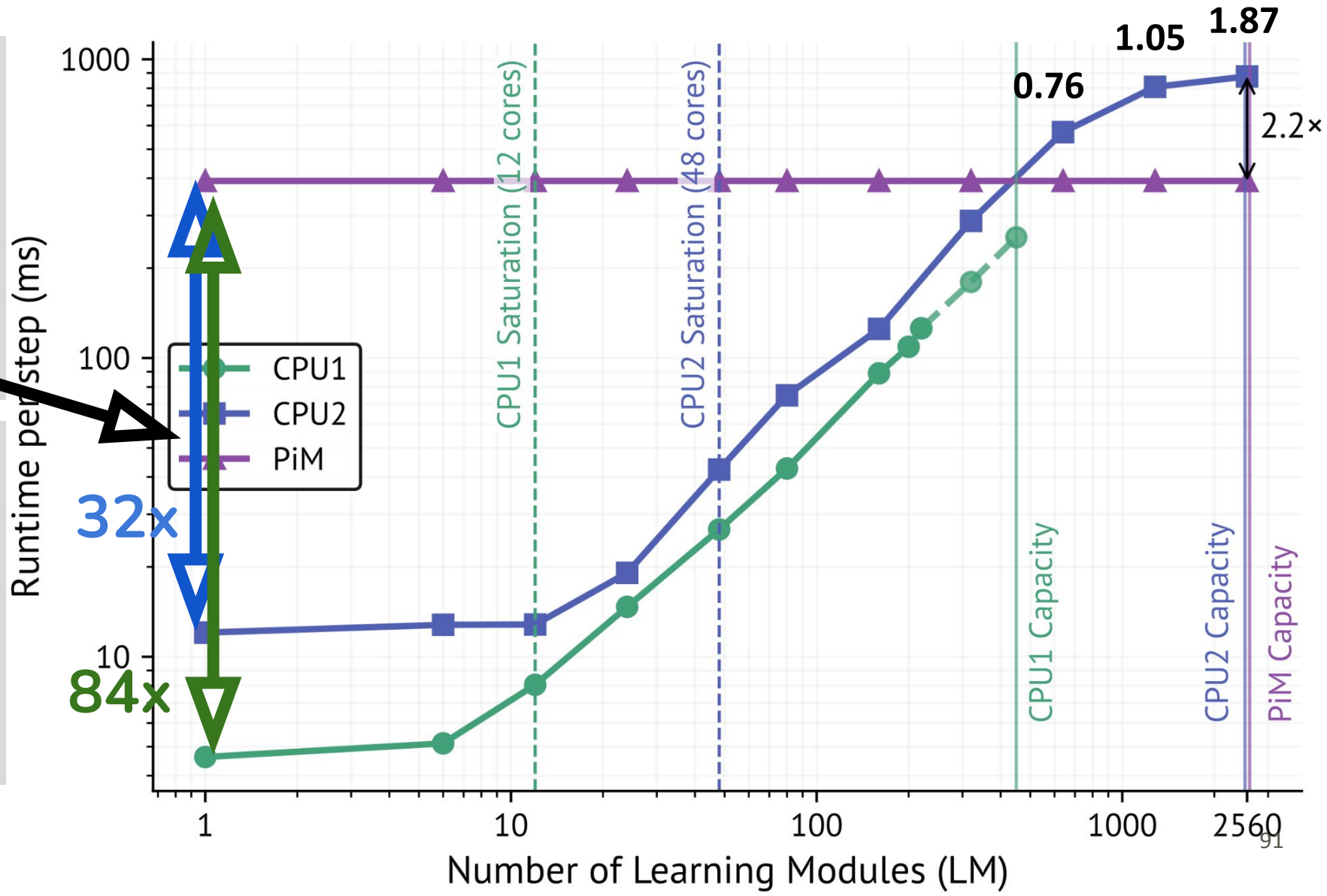
CPU2 benefits from **higher IPC** after as the **number of LMs increases,** explaining the **sub-linear scaling**

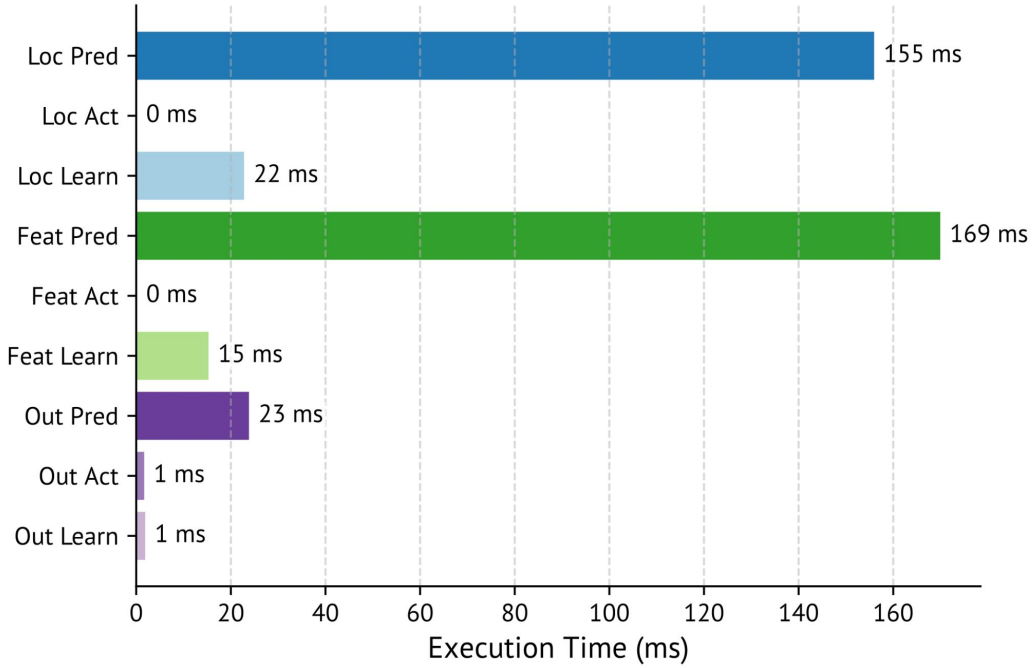


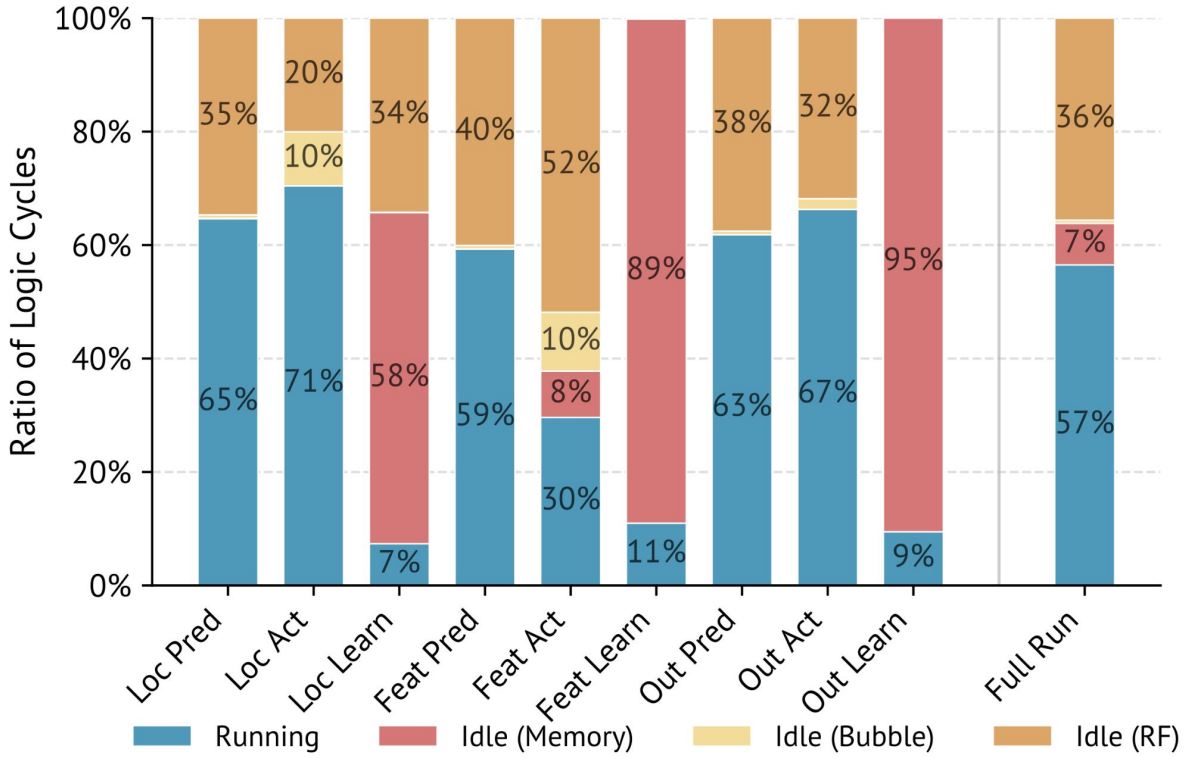
Montyll: Time per Step

CPU1 and **CPU2** massively **outperform** **PiM** for **1 LM**

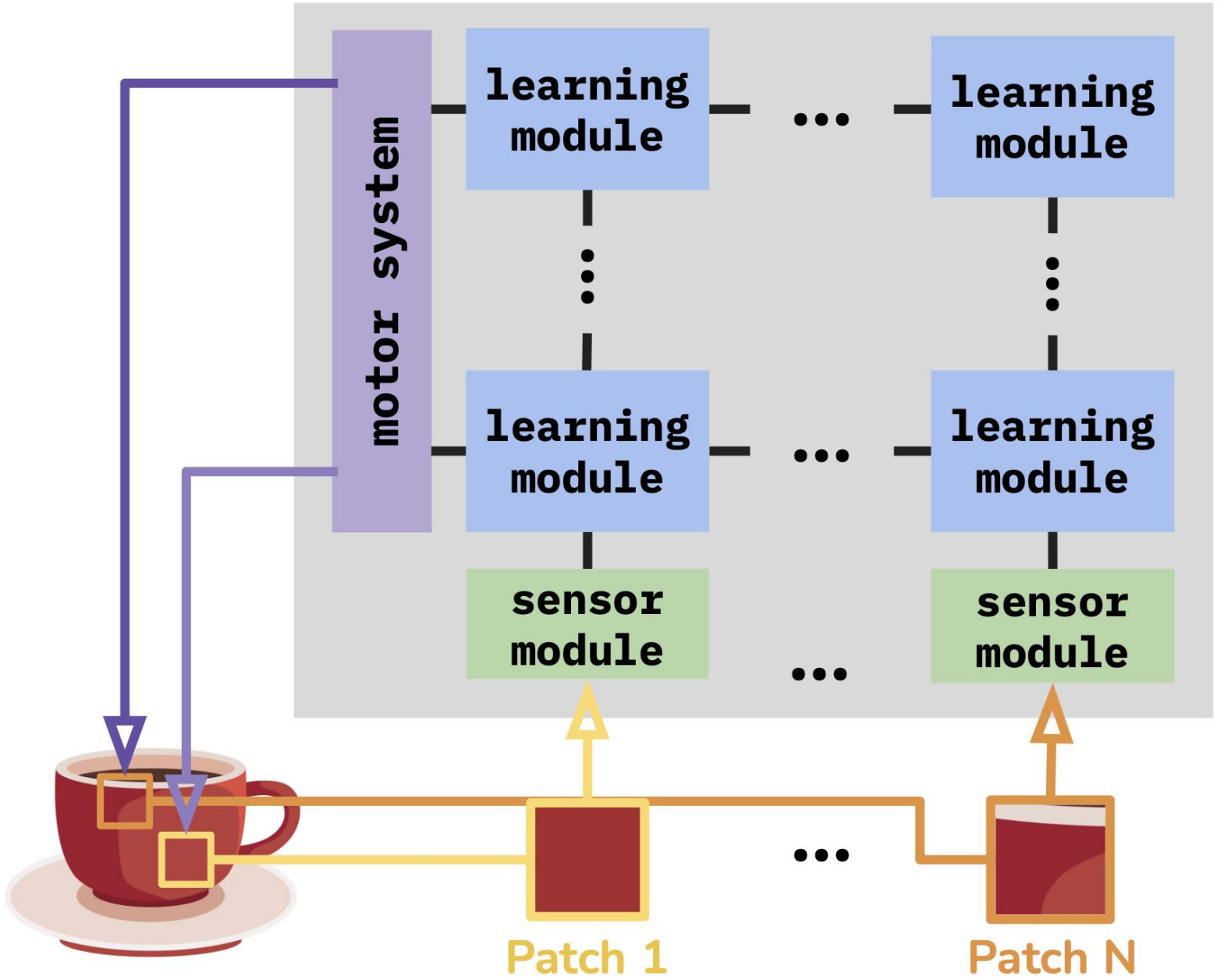
CPU1 has **2x** better **single-core latency** than **CPU2**

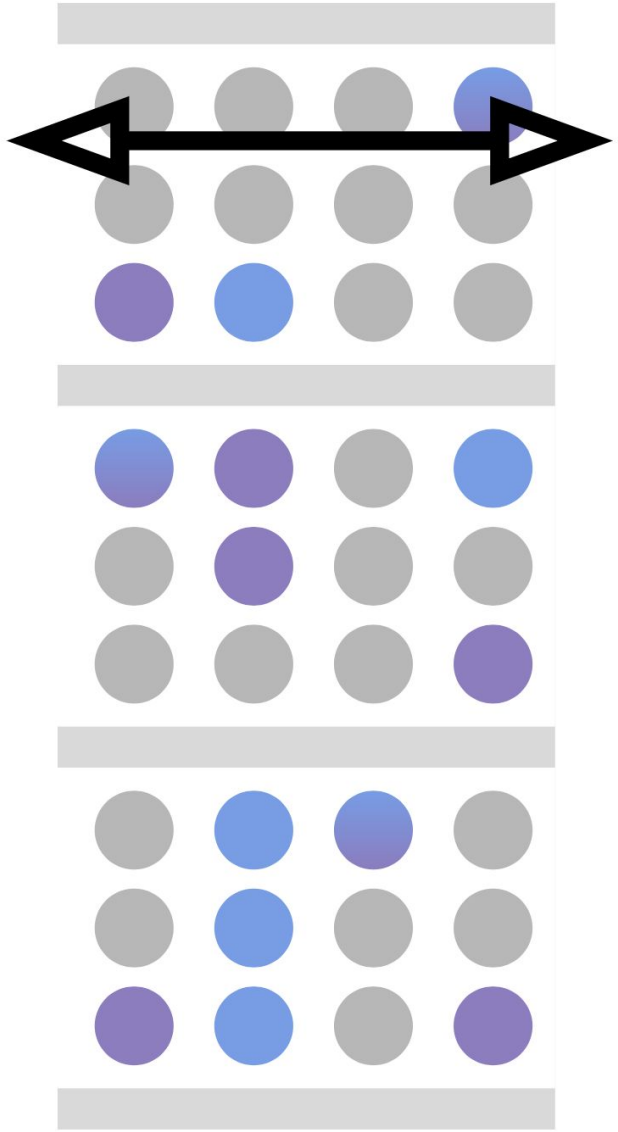



Loc Pred
Loc Act
Loc Learn
Feat Pred
Feat Act
Feat Learn
Out Pred
Out Act
Out Learn
155 ms
0 ms
22 ms
169 ms
0 ms
15 ms
23 ms
1 ms
1 ms
0
20
40
60
80
100
120
140
160
Execution Time (ms)


Ratio of Logic Cycles
100%
80%
60%
40%
20%
0%
35%
65%
20%
10%
71%
34%
58%
7%
40%
59%
52%
10%
8%
30%
89%
11%
38%
63%
32%
67%
95%
9%
36%
7%
57%
Loc Pred
Loc Act
Loc Learn
Feat Pred
Feat Act
Feat Learn
Out Pred
Out Act
Out Learn
Full Run
Running
Idle (Memory)
Idle (Bubble)
Idle (RF)

Conclusion


motor system
learning module
learning module
learning module
learning module
learning module
learning module
sensor module
sensor module
Patch 1
Patch N

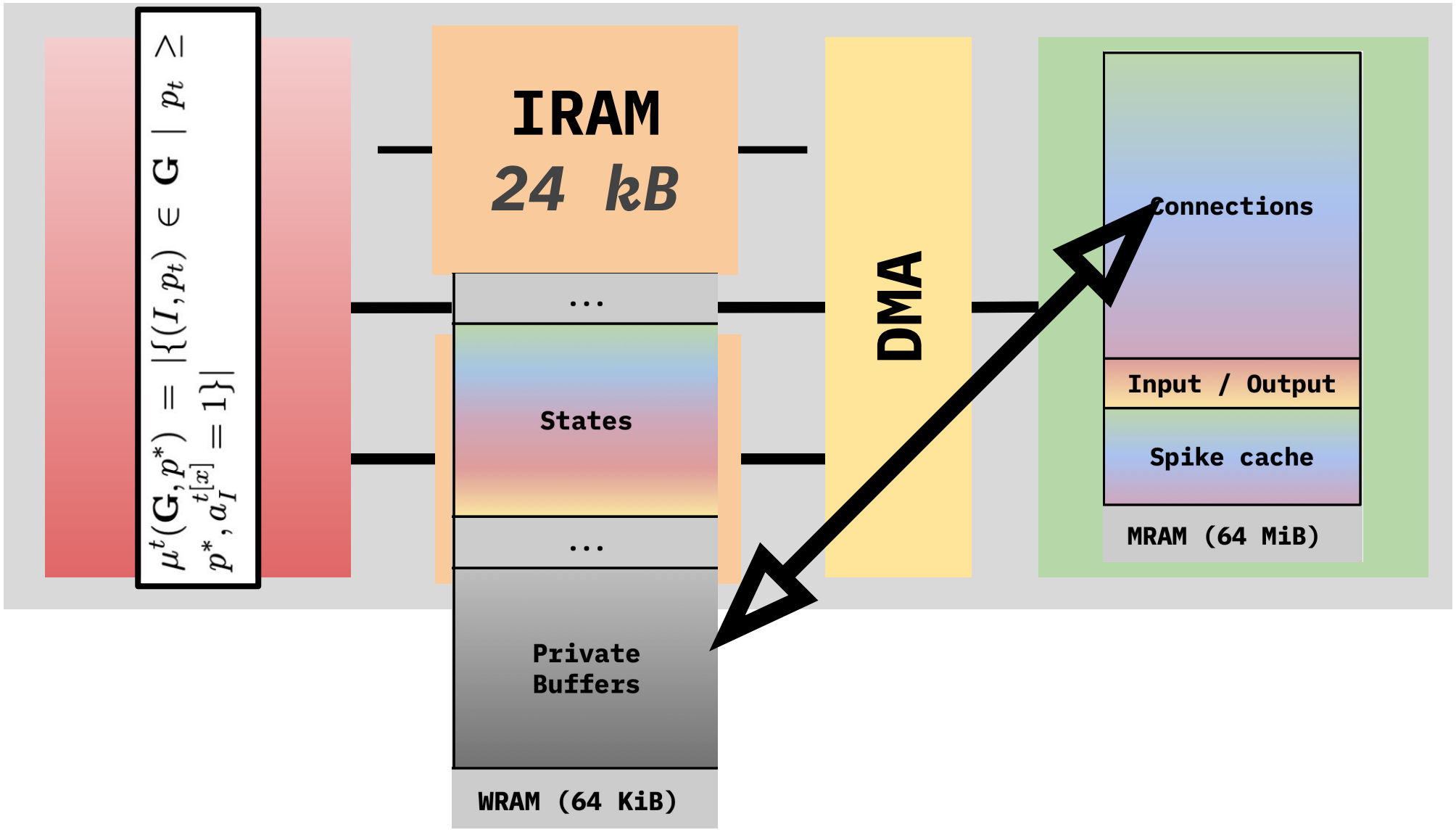


Output Layer (o)

Feature Layer (f)

Location Layer (l)

Mechanism	Mathematical Formulation
Synaptic Overlap	$\mu^t(\mathbf{G}, p^*) = \lvert\{(I, p_t) \in \mathbf{G} \mid p_t \geq p^*, a_I^{t[x]} = 1\}\rvert$
Segment NMDA Spike	$\tau_{ij}^t = \sum_d \mathbb{I}\{\mu^{t-1}(\mathbf{D}_{ijd}^t, p^*) \geq \theta_d\}$
Cell Depolarization	$\pi_I^t = 1$ if Condition(τ_I^t)
Somatic Action Potential	$a_I^t = \text{Decision}(\mu^t(\mathbf{F}_I^t, p^*), \pi_I^{t-1})$


$\mu^t(\mathbf{G}, p^*) = |\{(I, p_t) \in \mathbf{G} \mid p_t \geq p^*, a_I^{t[x]} = 1\}|$
IRAM
24 kB
...
States
...
Private Buffers
WRAM (64 KiB)
DMA
Connections
Input / Output
Spike cache
MRAM (64 MiB)

We successfully scale **Montyll** to **2560 LMs** on a **CPU system** and a **PiM system**, for a total of **44.5M neurons** and **16.1B synapses**

The **PiM system** outperforms the **CPU system** by **2.2x**, running at **2.55** vs. **1.14** **steps/second**

We open source our code

Montyll: High-performance C library

github.com/Xavier0301/cmontyll

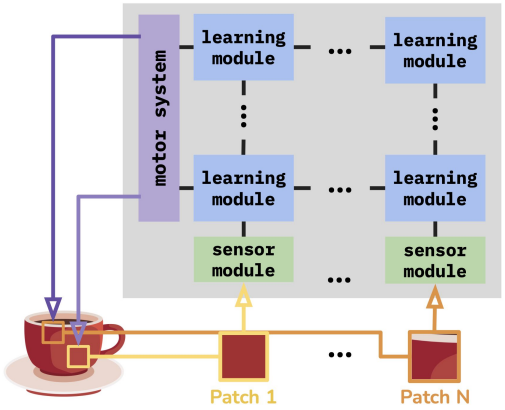
Montyll-PiM: Implemented on Functional simulator

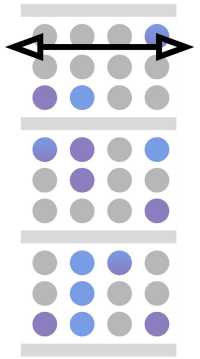
github.com/Xavier0301/montyll-pim

Montyll-PiM: Implemented on cycle-level simulator

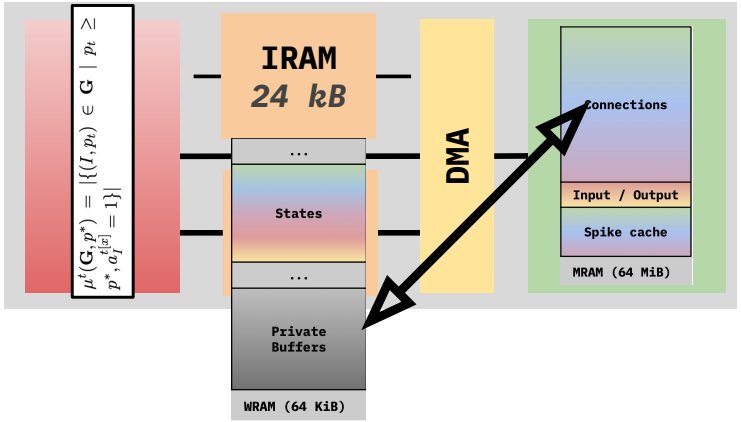
github.com/Xavier0301/montyll-upimulator

Conclusions





Mechanism	Mathematical Formulation
Synaptic Overlap	$\mu^t(\mathbf{G}, p^*) = \lvert\{(I, p_t) \in \mathbf{G} \mid p_t \geq p^*, a_I^{t[x]} = 1\}\rvert$
Segment NMDA Spike	$\tau_{ij}^t = \sum_d \mathbb{I}\{\mu^{t-1}(\mathbf{D}_{ijd}^t, p^*) \geq \theta_d\}$
Cell Depolarization	$\pi_I^t = 1$ if Condition(τ_I^t)
Somatic Action Potential	$a_I^t = \text{Decision}(\mu^t(\mathbf{F}_I^t, p^*), \pi_I^{t-1})$



We successfully scale Montyll to **2560 LMs** on a CPU system and a PiM system, for a total of **44.5M neurons** and **16.1B synapses**

The PiM system outperforms the CPU system by **2.2x**, running at 2.55 vs. 1.14 **steps/second**

More of SPCL's research:

youtube.com/@spcl — 210+ Talks

twitter.com/spcl_eth — 1.6K+ Followers

github.com/spcl — 5.6K+ Stars

… or spcl.ethz.ch

We open source our code

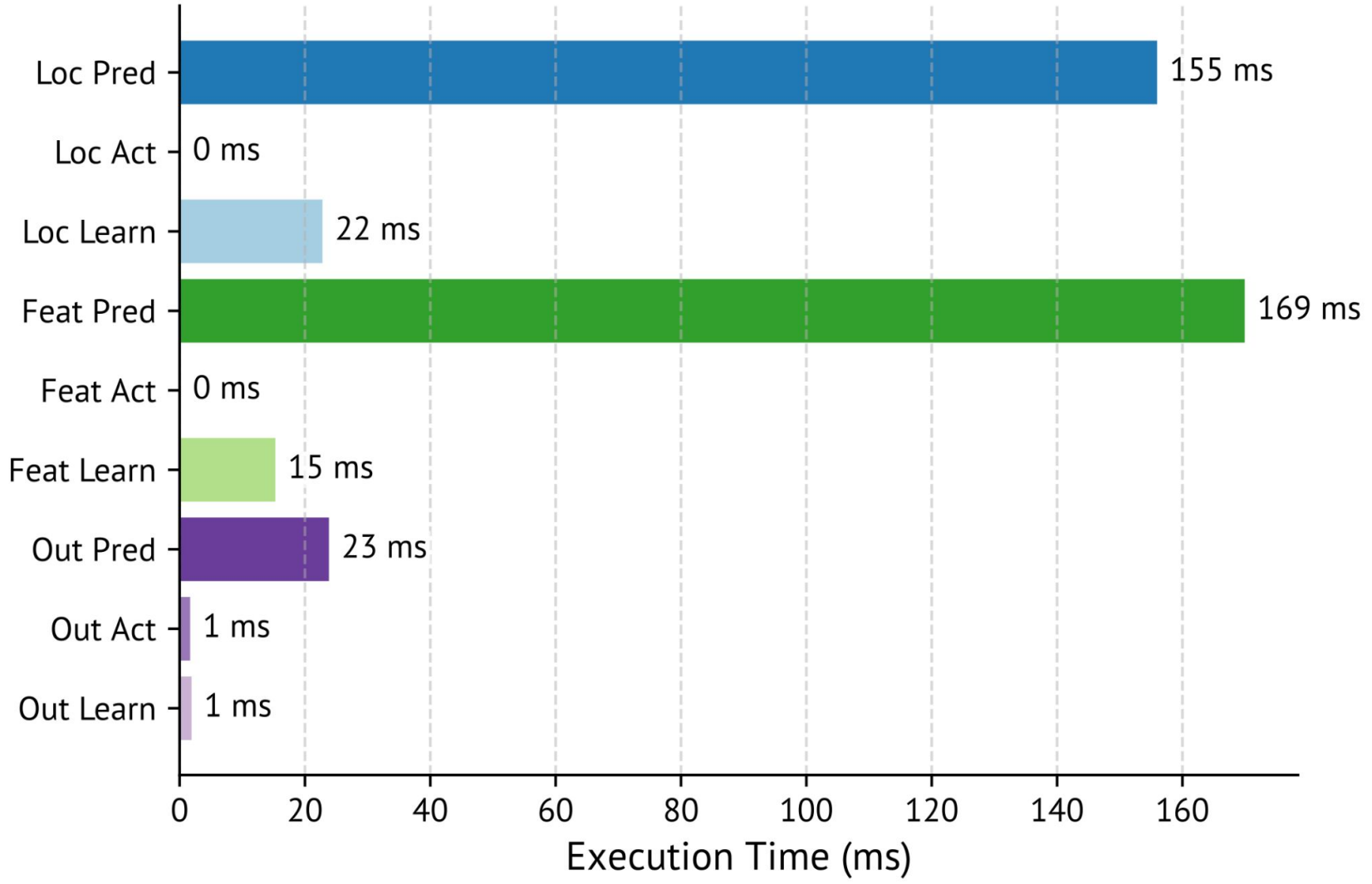
Montyll: High-performance C library
github.com/Xavier0301/cmontyll

Montyll-PiM: Implemented on Functional simulator
github.com/Xavier0301/montyll-pim

Montyll-PiM: Implemented on cycle-level simulator
github.com/Xavier0301/montyll-upimulator

Backup Slides

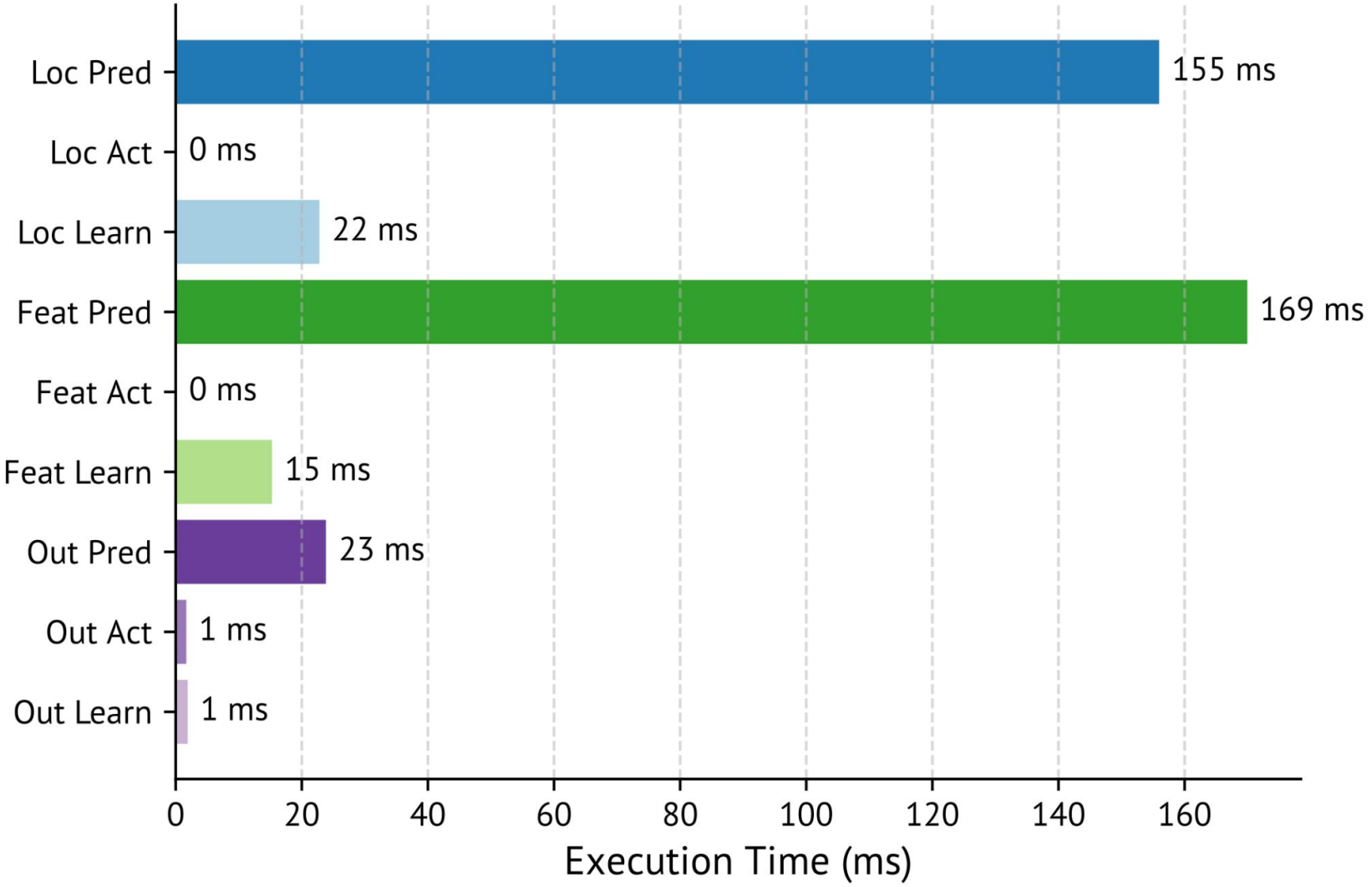
Montyll: Time per Step Breakdown on PiM



Montyll: Time per Step Breakdown on PiM

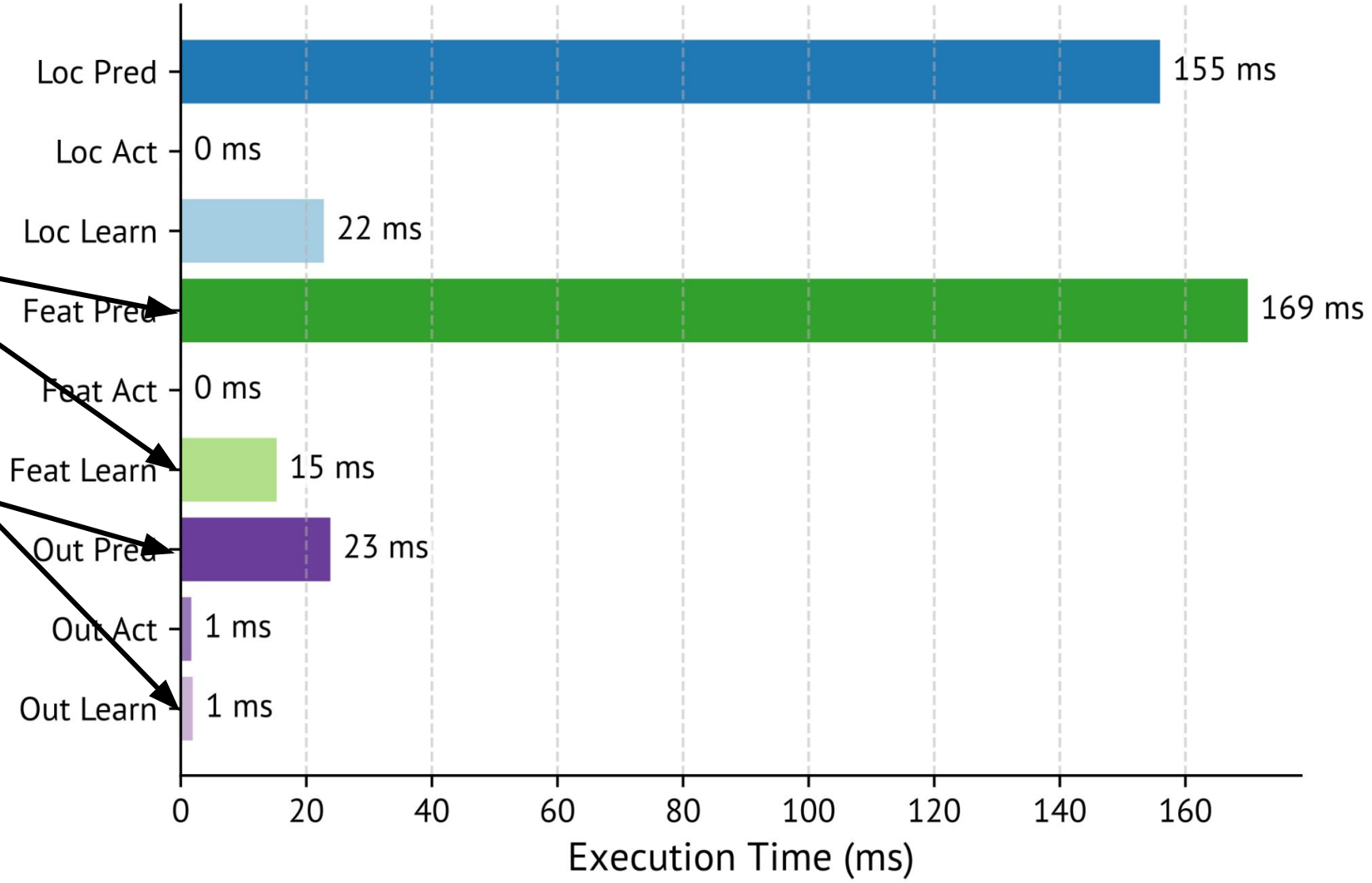
82.8% of the **per step runtime** is spent **computing predictions**

Activations are **negligible,** with **0.18%** of **runtime** spent



Montyll: Time per Step Breakdown on PiM

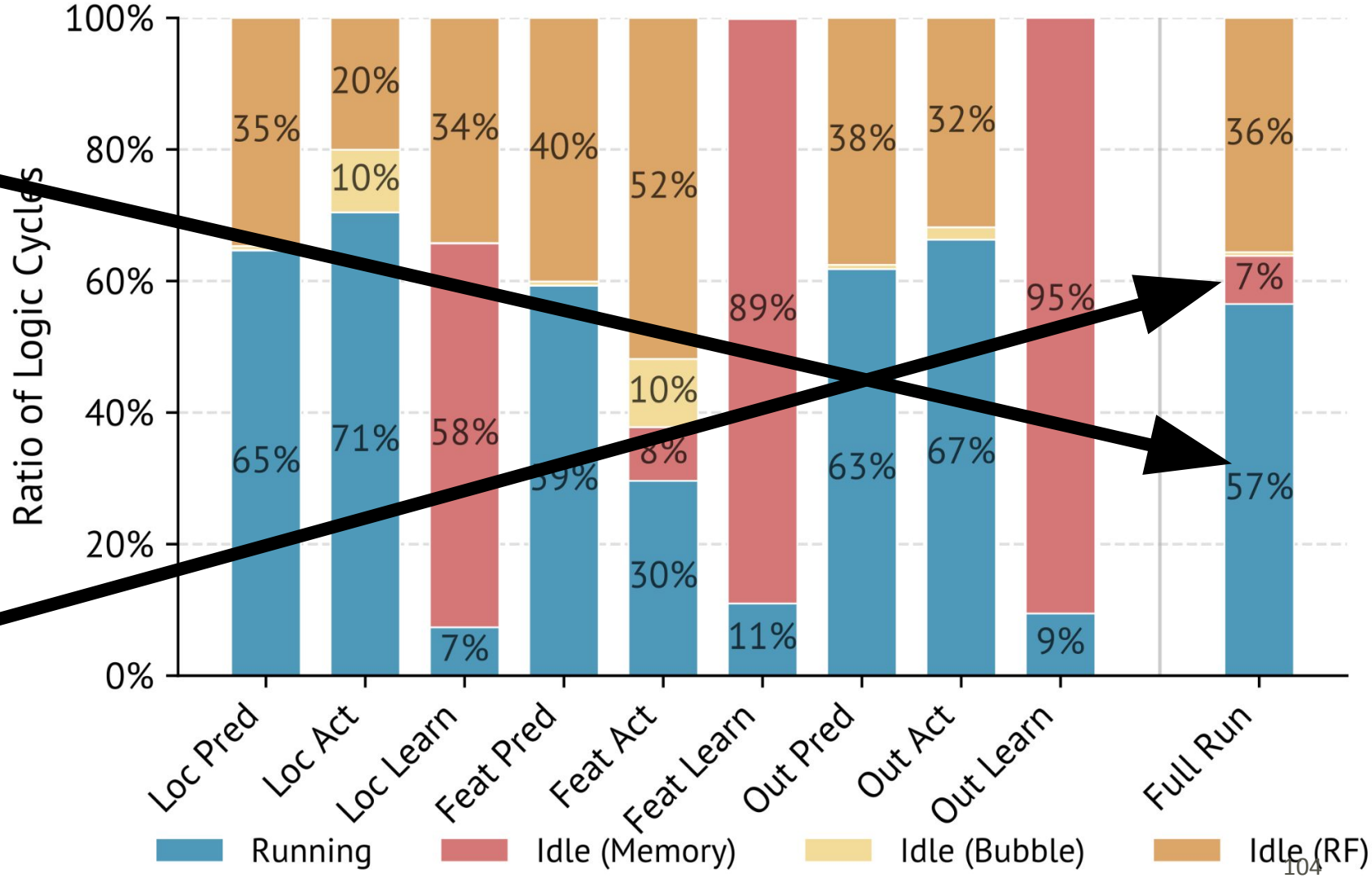
	Overlap (Prediction)	Weight Update (Learning)
LM: Feature	$O(MNSC)$	$O(MNSCs)$
LM: Location	$O(MNSC)$	$O(MNSCs)$
LM: Output	$O(MSC)$	$O(MSCs)$



Montyll: PiM pipeline activity

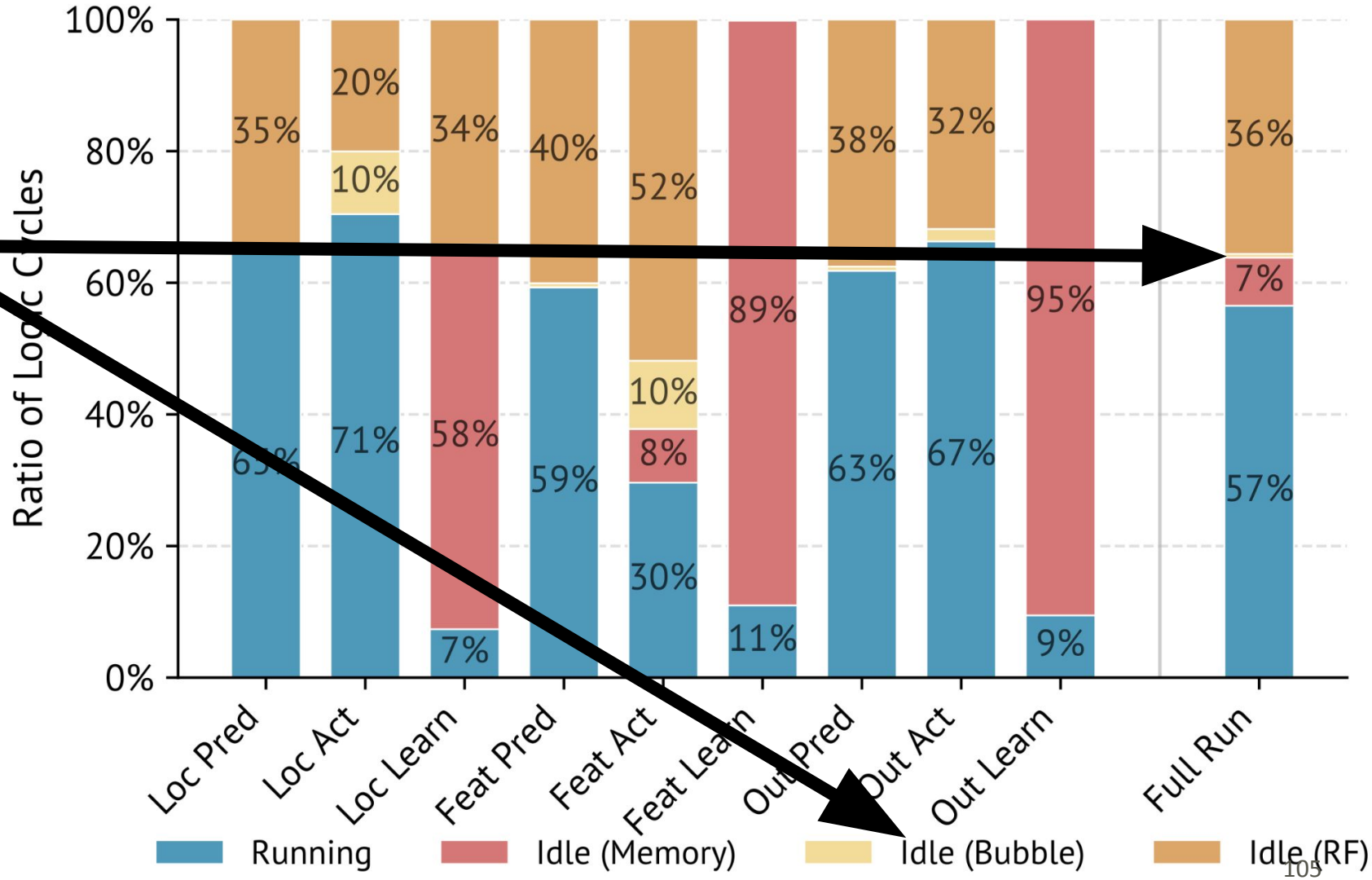
IPC is 0.57, (theoretical maximum is 1.0)

Workload is heavily **compute-bound** on PiM, **memory accesses** account for **7% of runtime**



Montyll: PiM pipeline activity

Work is **distributed very well**, almost no stalls are caused by **in-issuable tasklets** in the **FGMT pipeline**



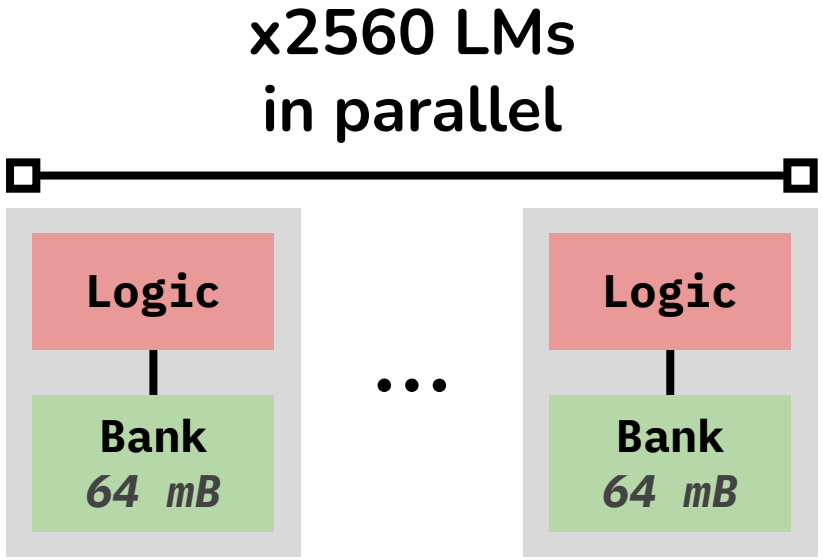
Time per step
for 2560 Learning Modules (LMs)

Data movement per step: ~70 MiB / LM

logic ↔ bank access bandwidth[1]: ~500 MiB/s

Operations per step: ~70M / LM

logic frequency[2]: 400 MHz

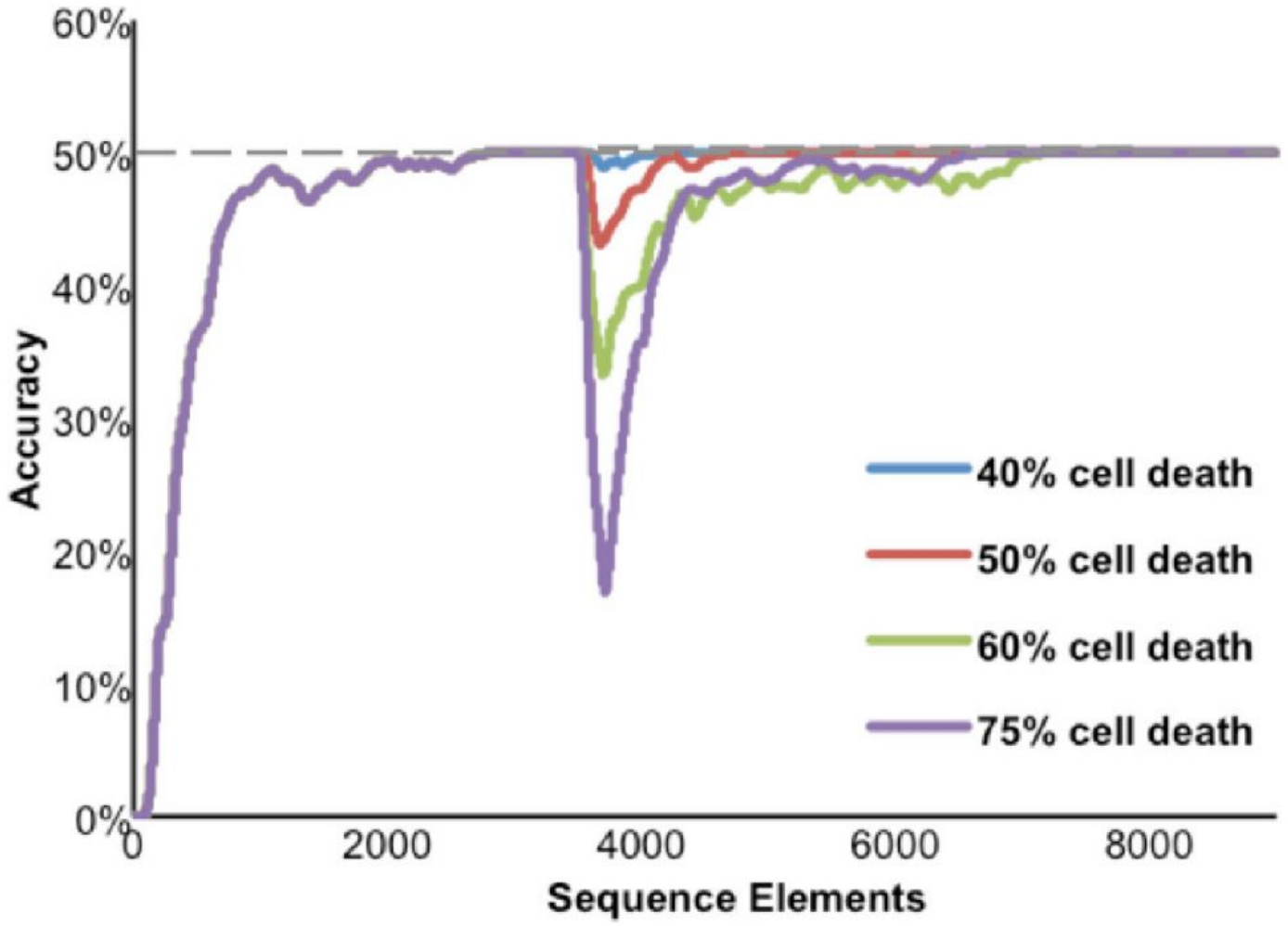


One step takes 70/400 + 70/600 = 0.175 + 0.12 = 0.295, assuming we **cannot** hide mram-wram transfer latency
steps per second is 1/0.295 = 3.39

→ Optimistically, we have **enough internal bandwidth** and a **big enough core frequency** to process **3.39 steps/second**

[1][2]Gómez-Luna, Juan, et al. "Benchmarking a new paradigm: An experimental analysis of a real processing-in-memory architecture." arXiv preprint (2021).
[1] **assuming big enough transfer chunks,** which is the case for this workload because of the underlying access pattern

HTM Networks: Robustness to noise



Hawkins, Jeff, and Subutai Ahmad. "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *Frontiers in neural circuits* 10 (**2016**): 174222.

Boosting $b_i = e^{-\beta(A_i^t - \overline{A^t})}$

No **floating-point unit**, therefore the **boosting factors** are calculated using a combination of **fixed-point arithmetic** and **hybrid dynamic-precision lookup tables**

$$g: \begin{aligned} [\![0; B]\!] \times \{0; 1\} &\to [\![0; B]\!] \\ (X, Y) &\mapsto Bf(X/B, Y) = X + (YB - X)/T \end{aligned}$$

https://github.com/Xavier0301/montyll-pim

Boosting $b_i = e^{-\beta(A_i^t - \overline{A^t})}$

No **floating-point unit**, therefore the **boosting factors** are calculated using a combination of **fixed-point arithmetic** and **hybrid dynamic-precision lookup tables**

- For columns which response needs to be suppressed ($b_i \leq 0.5$), the LUT stores a negative value representing a logarithmic shift magnitude ($= \log_2(b_i)$). The runtime applies this as a right-shift operation (`>> boosting_factor`), effectively dividing the response.
- Otherwise if $b_i > 0.5$, the LUT stores the direct integer multiplier. The runtime applies this using the available 8x8 bit multiplication (`* boosting_factor`).

https://github.com/Xavier0301/montyll-pim