An Adaptive Memory Management Strategy Towards Energy Efficient Machine Inference in Event-Driven Neuromorphic Accelerators

Saunak Saha, Henry Duwe, and Joseph Zambreno

Iowa State University

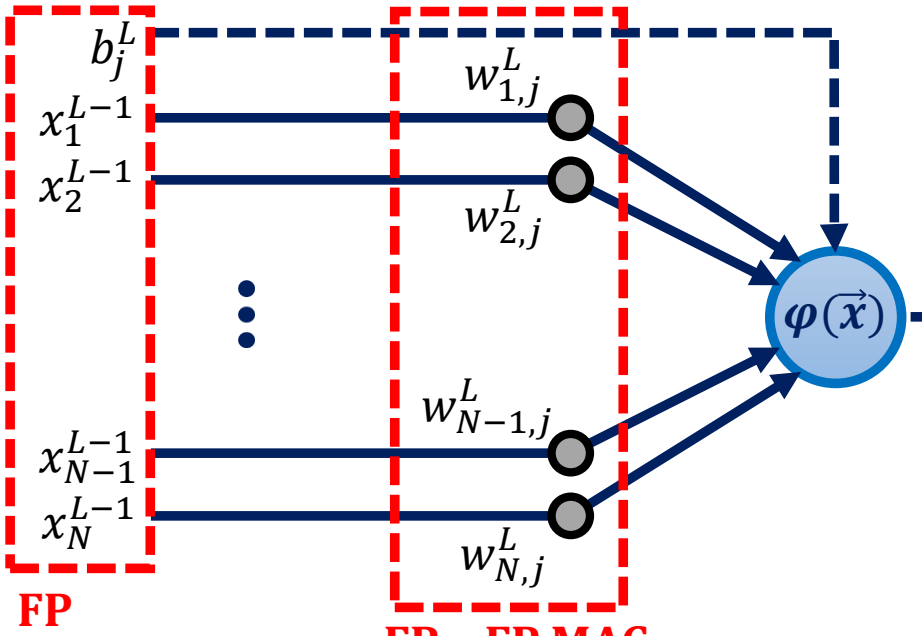
Ames, IA, United States

07/16/2019

International Conference on Application-specific Systems, Architectures and Processors (ASAP) 2019

IOWA STATE UNIVERSITY
Reconfigurable Computing Laboratory

Why SNNs over ANNs?

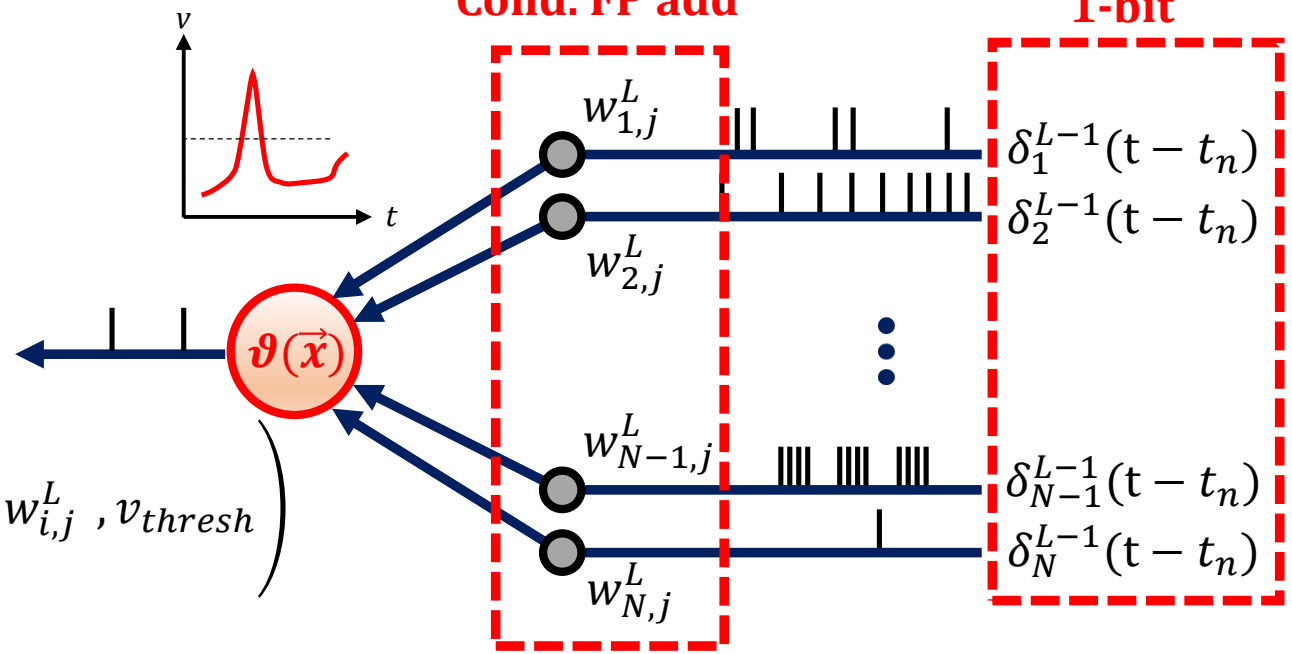


$$\varphi_j^L(\vec{x}) = f_{act}{}_j^L\left(\sum_{i=1}^{N} x_i^{L-1} w_{i,j}^L + b_j^L\right)$$

FP comp.

⇐ ANN

Artificial Neural Network



SNN ⇒

Spiking Neural Network

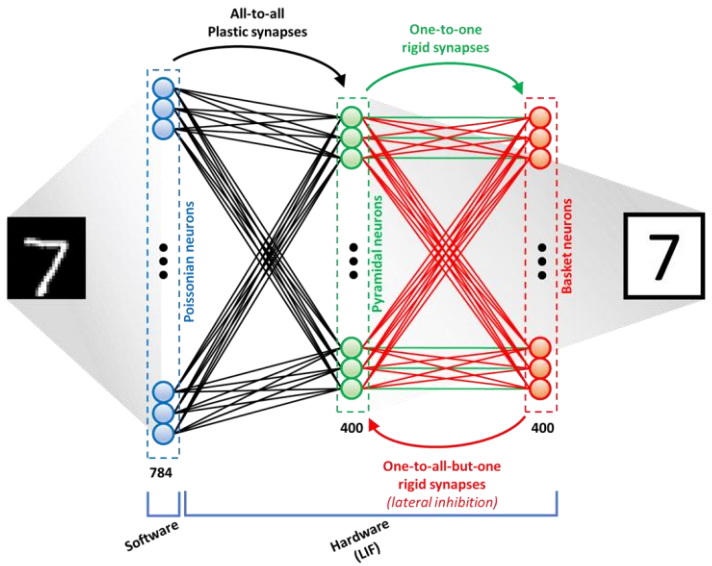
$$\vartheta_j^L(\vec{x}, t_n) = h_j^L\left(\sum_{i=1}^{N} \delta_1^{L-1}(t - t_n) w_{i,j}^L \;, v_{thresh}\right)$$

FP comp.

Representative SNNs

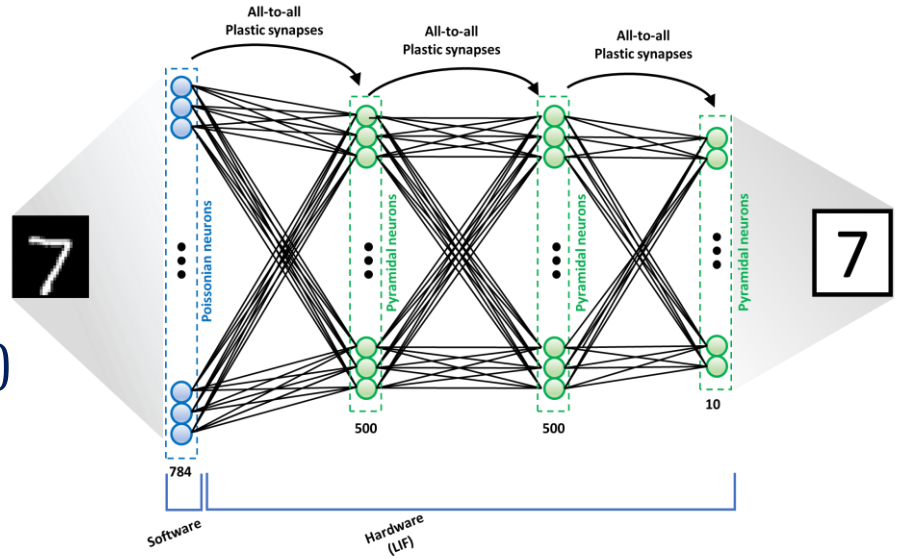
SCWN

(Spiking Competitive Winner-Take-all Network)



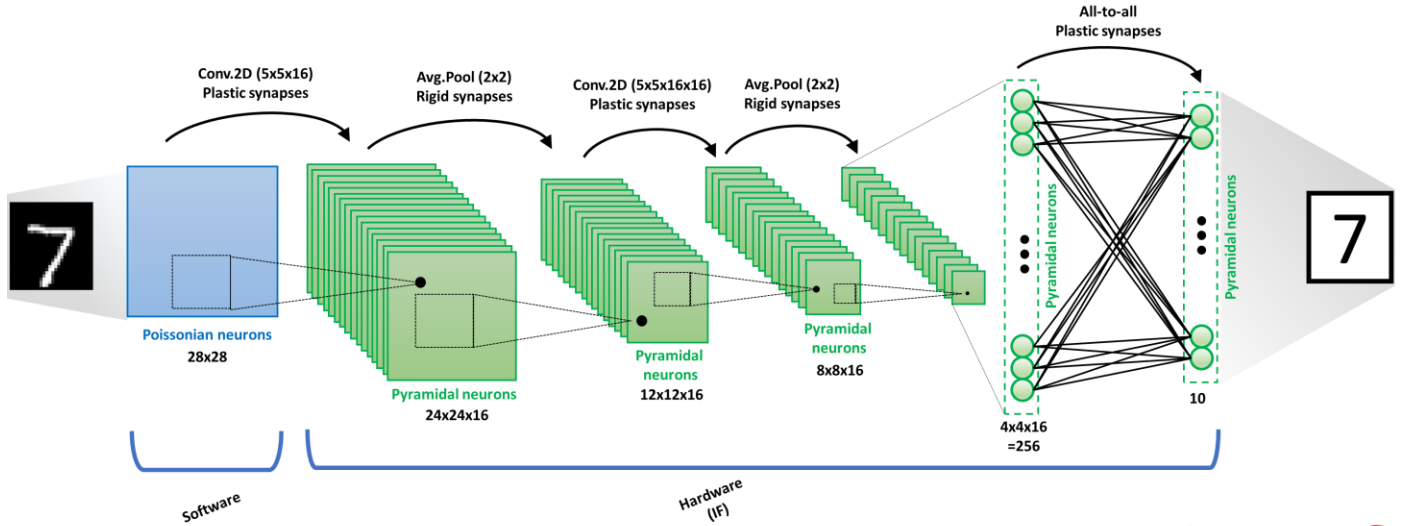
SDBN

(Spiking Deep Belief Network)



SCNN

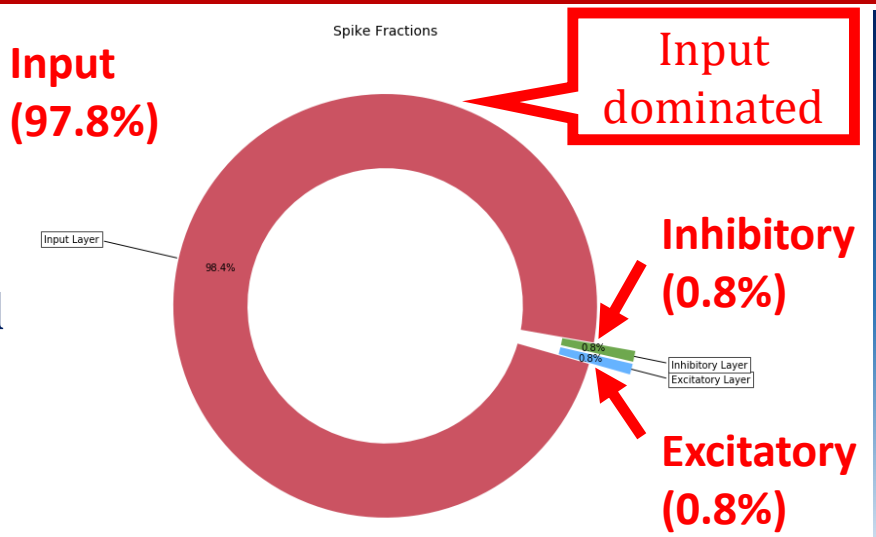
(Spiking Convolutional Neural Network)



Representative SNNs

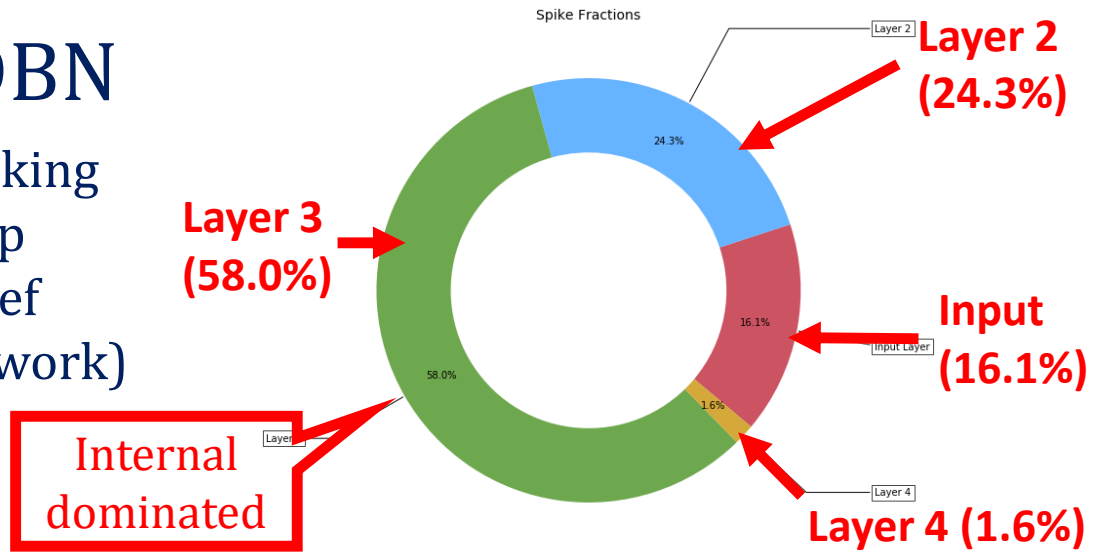
SCWN

(Spiking Competitive Winner-Take-all Network)



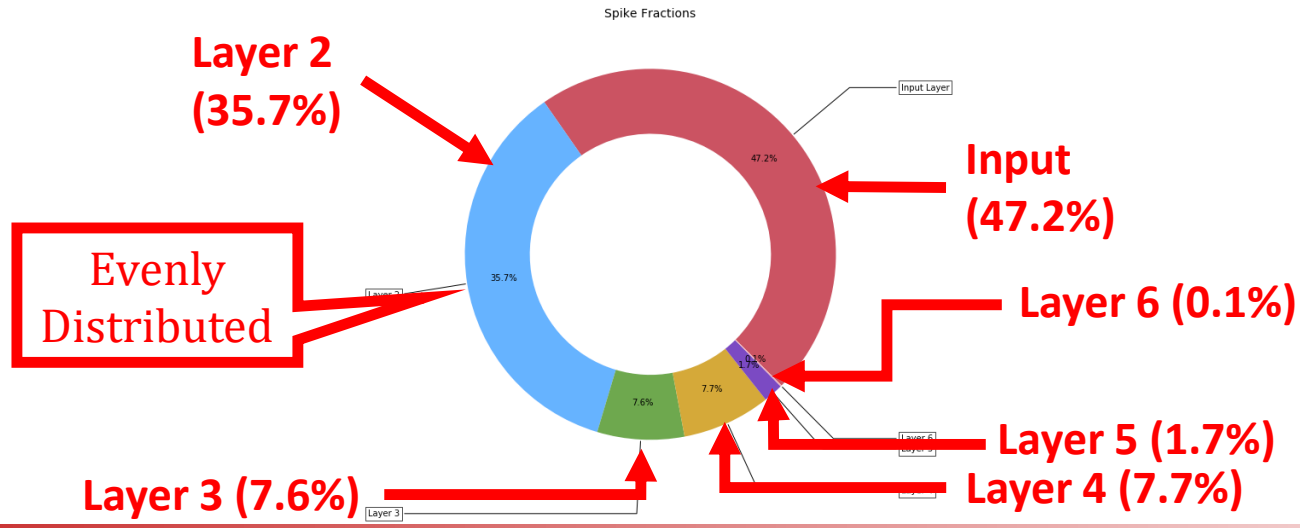
SDBN

(Spiking Deep Belief Network)

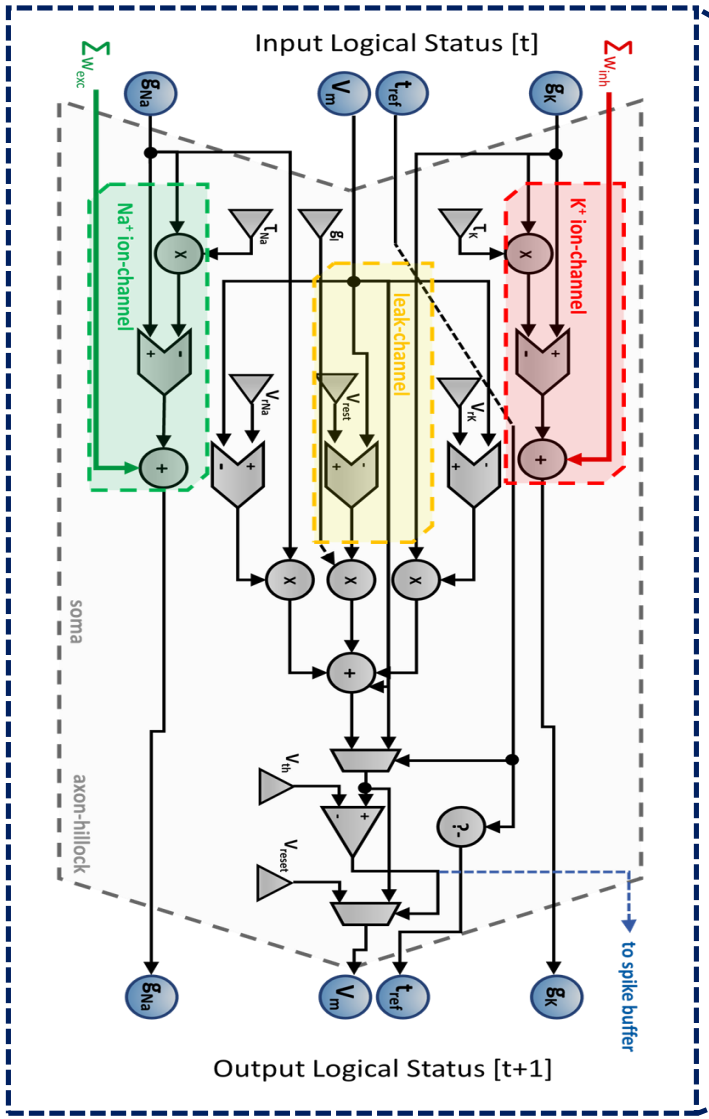


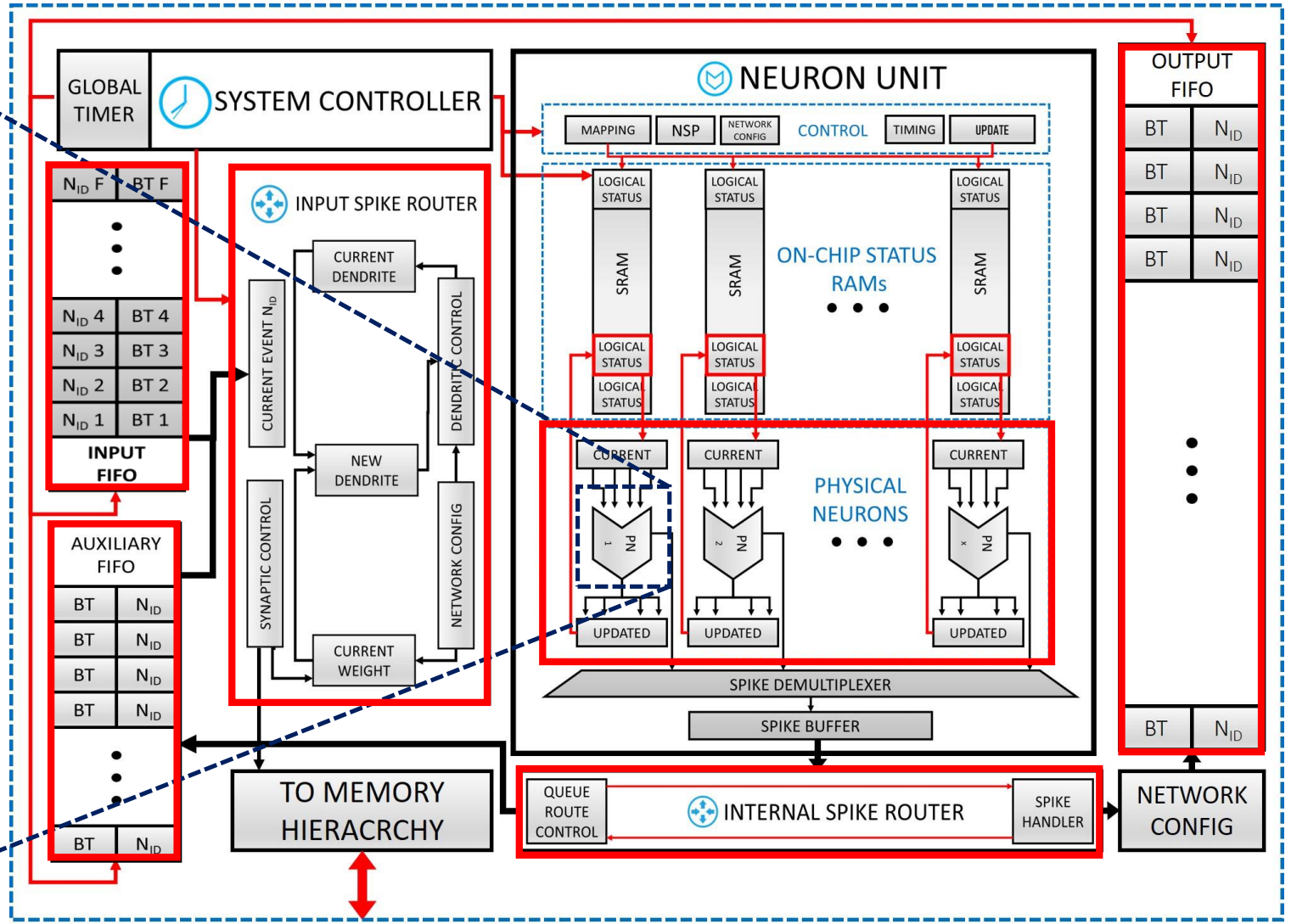
SCNN

(Spiking Convolutional Neural Network)

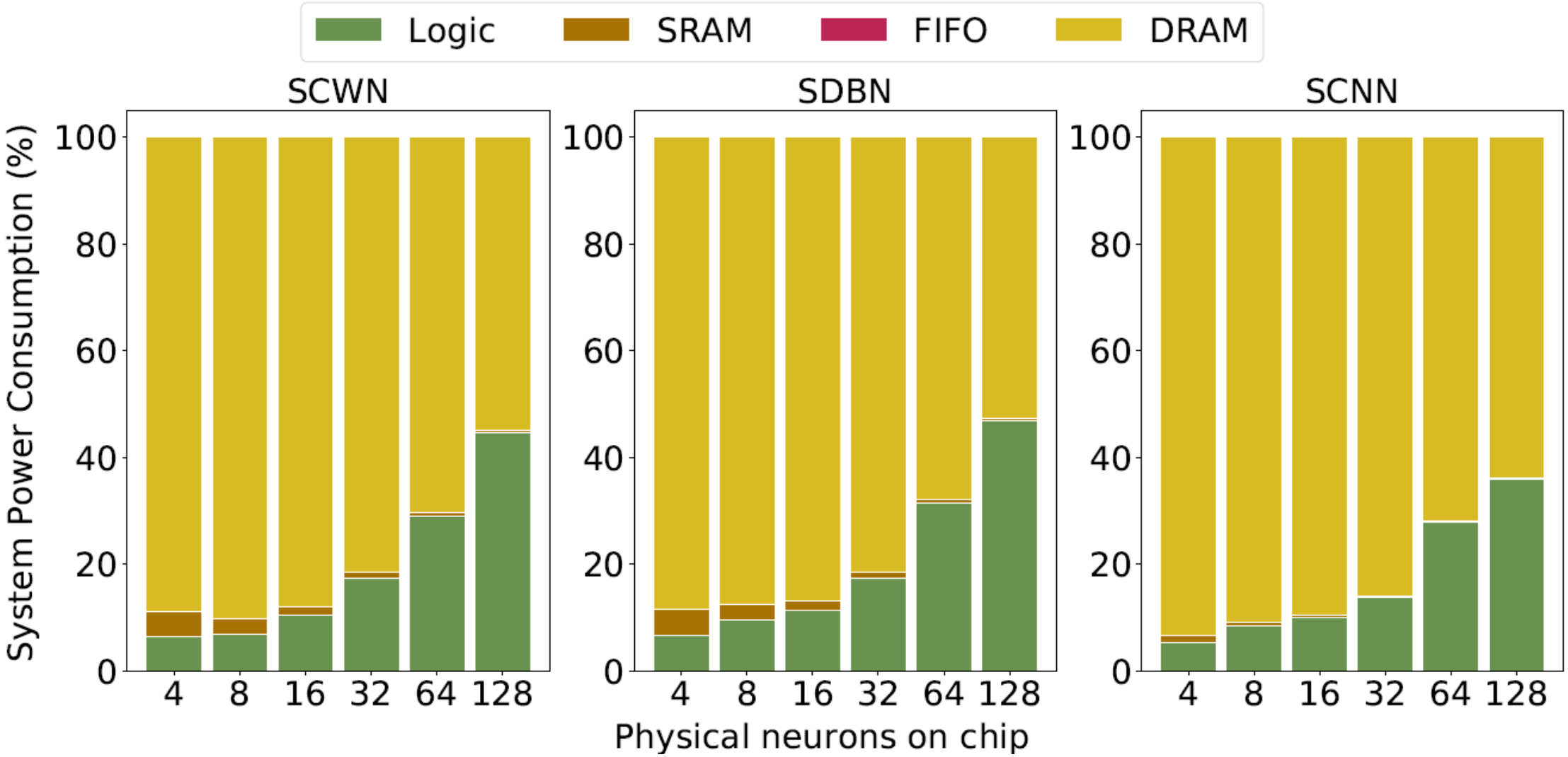


The CyNAPSE microarchitecture





Baseline power consumption



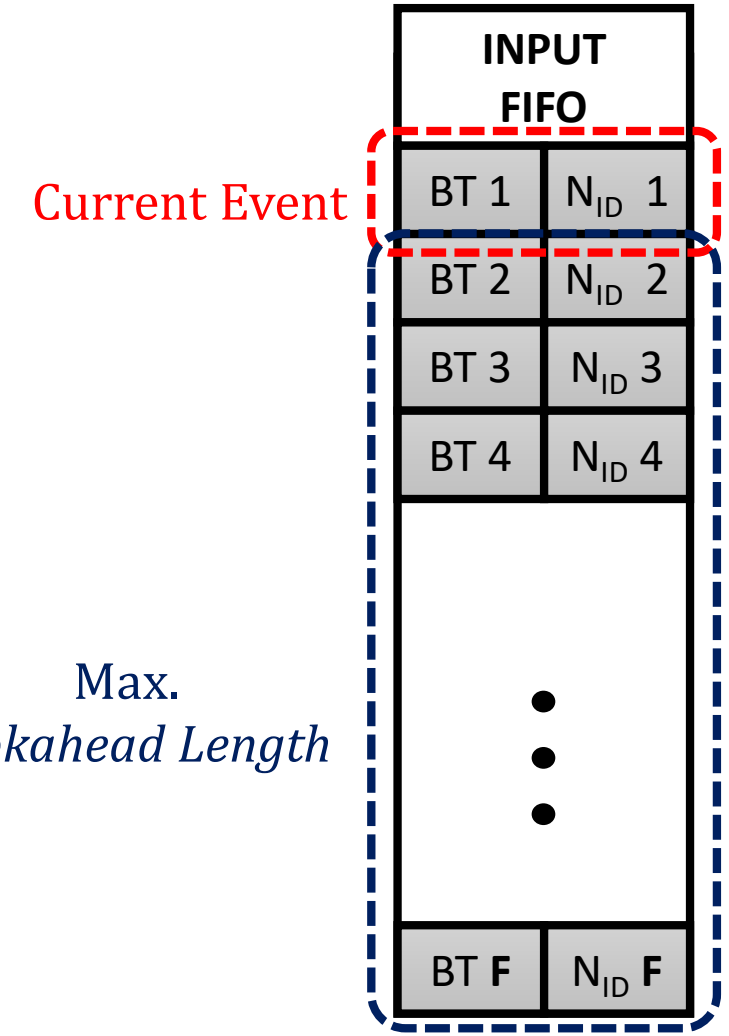
Energy-efficient memory management techniques

General purpose computing:

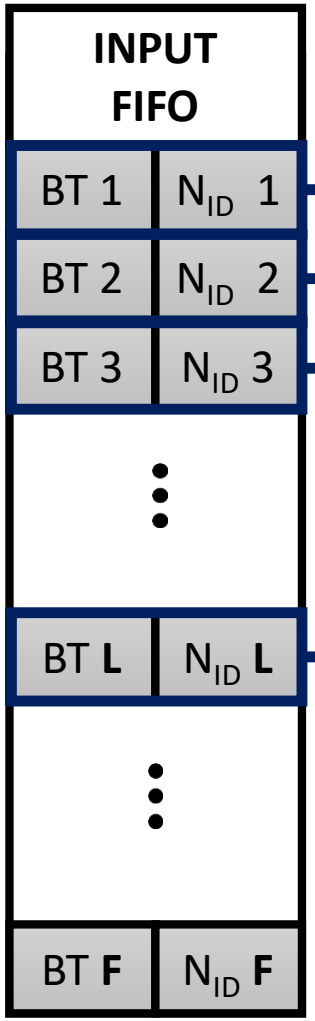
- Memory hierarchy and caches
- LRU and Random
- Belady's OPT: *Infeasible* [3]
- DIP[4], RRIP[5], LIRS[6] : *Speculative*

CyNAPSE:

- Input queue
- Event-driven simulation: *Inherent forward visibility*
- Depending on *Route latency, Queue length* and steady-state *Memory bandwidth*



Proposed management scheme



Warm Up

Read: Allocate and set **reuse score**

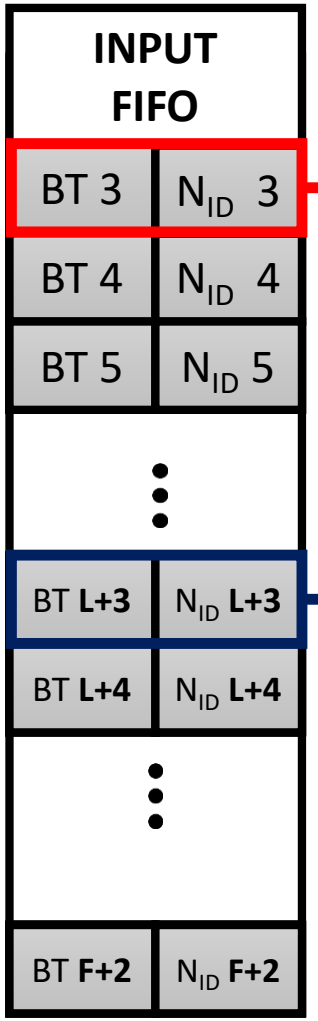
Read: Allocate and set reuse score

Read: Allocate and set reuse score

Read: Allocate and set reuse score

Reuse score is a field associated with every cache line that denotes the number of times that line is expected to be reused in the future.

Proposed management scheme



Post-warm up

Route: Route, allocate if required, and set reuse score

Read: Allocate and set reuse score

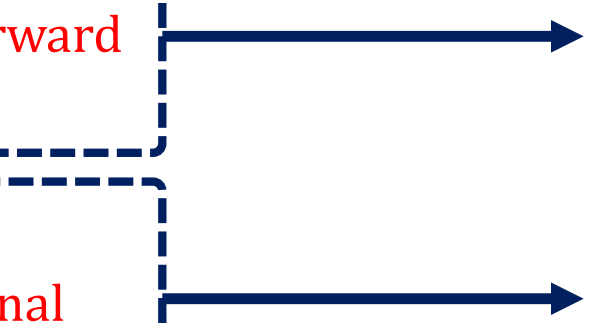
Multiple exclusive R/W ports

Network-adaptive enhancements

- Works for Input Events ONLY. Internal Events?
  - -> Routed in immediately next timestep
  - -> (Internal/Input) activity is significant in some networks (>1 event)

Statistics on a *layer-by-layer* granularity saves both storage and processing overheads

- Topological hints
  - -> E.g. Output layer neurons in feed-forward networks
- Simulation hints
  - -> E.g. Sparse activations in convolutional layers



Static Kernel Information

- Layer types (conv2D, pool, dense)
- Pyramidal/Basket
- Ranges
- Connectivity

Dynamic Kernel Statistics

- Input/Output Layer activity fraction
- Layer mean reuse distance

Need network -adaptive enhancements to the scheme.

Network-adaptive enhancements

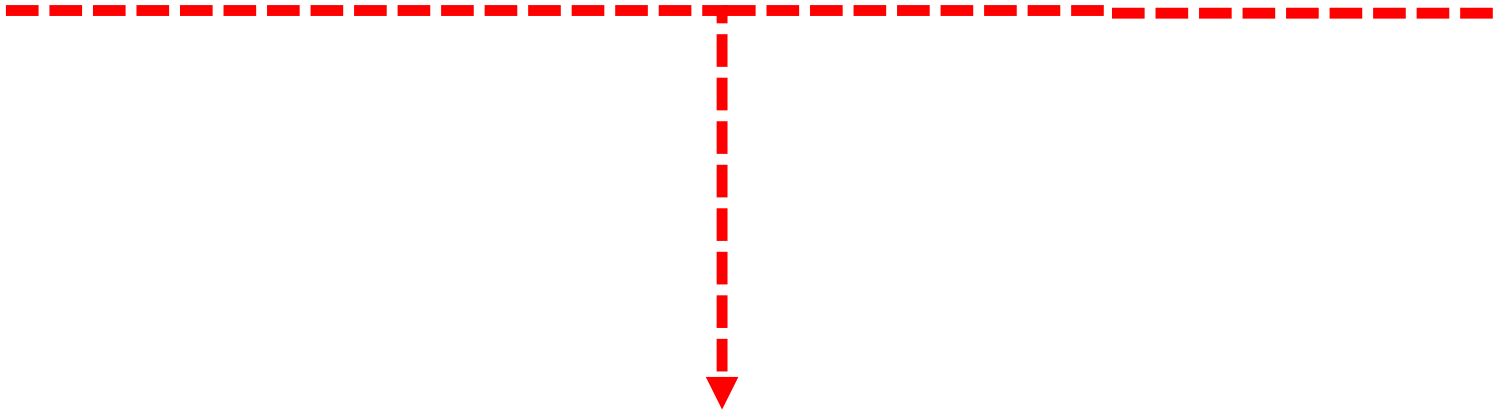
Static Kernel Information

- Layer types (conv2D, pool, dense)
- Pyramidal/Basket ranges
- Connectivity
- Input/Output

Dynamic Kernel Statistics

- Layer activity fraction
- Layer mean reuse distance

Low Activity Layers



Cache Bypassing

Disallowing allocation of low activity neurons preventing them from thrashing high reuse input neurons

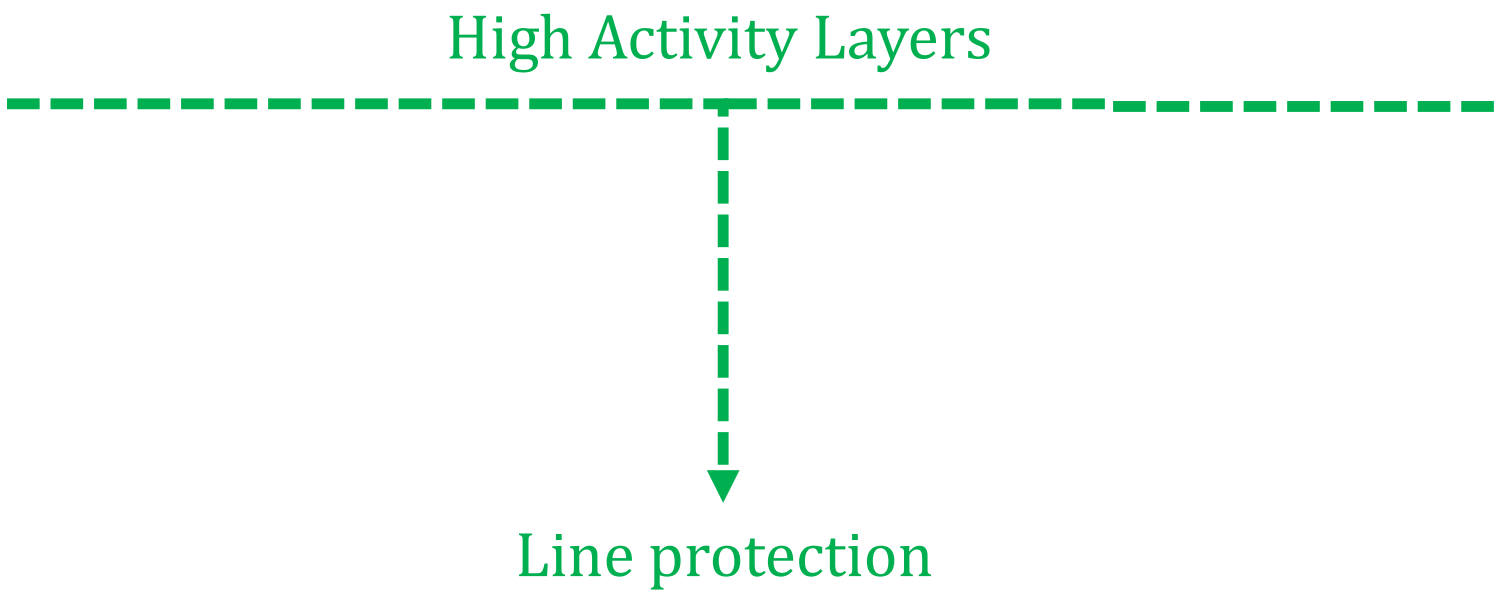
Network-adaptive enhancements

Static Kernel Information

- Layer types (conv2D, pool, dense)
- Pyramidal/Basket ranges
- Connectivity
- Input/Output

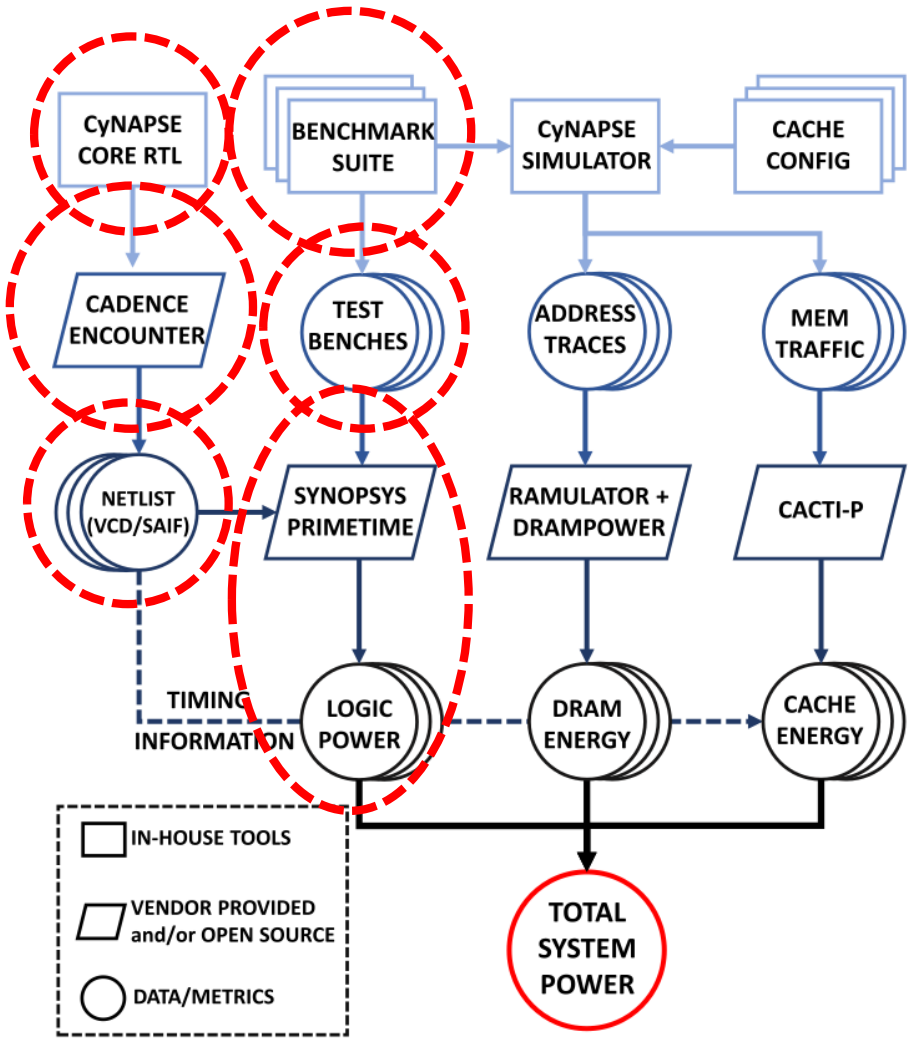
Dynamic Kernel Statistics

- Layer activity fraction
- Layer mean reuse distance

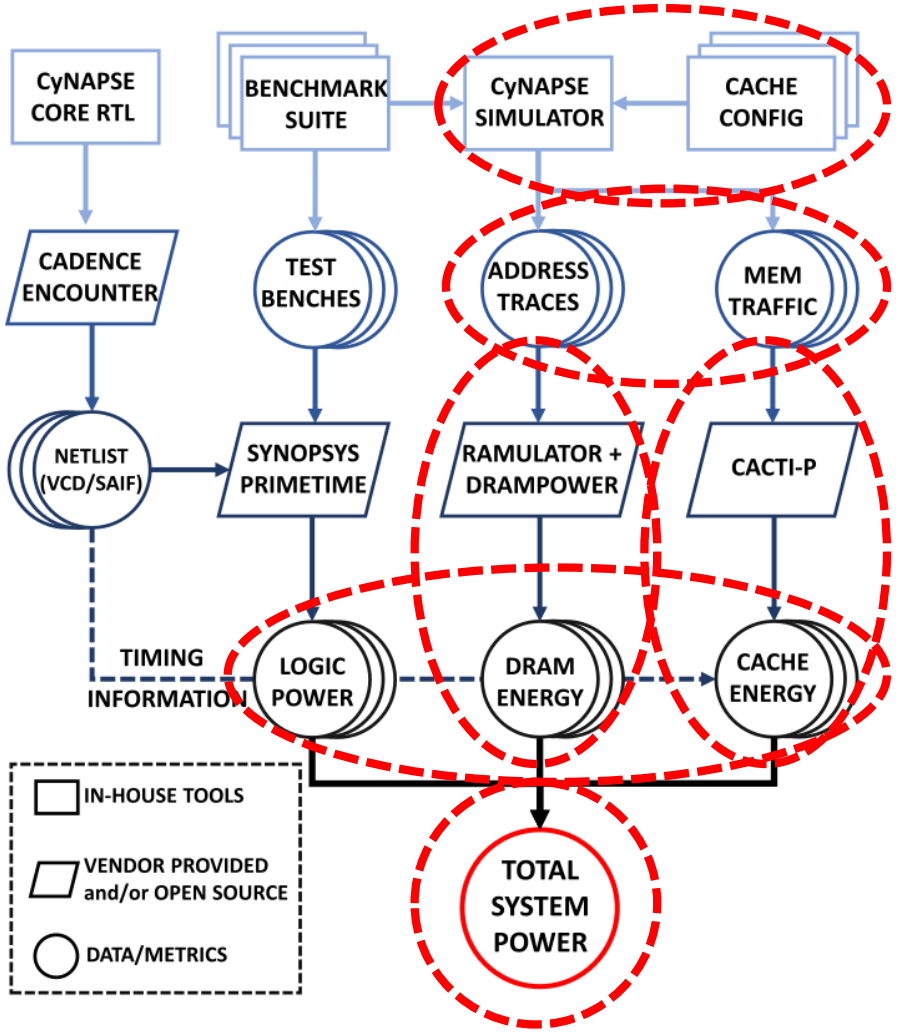


Arming high-activity neurons with an *probable reuse score* based on their reuse distances to prevent being thrashed by low-reuse input neurons

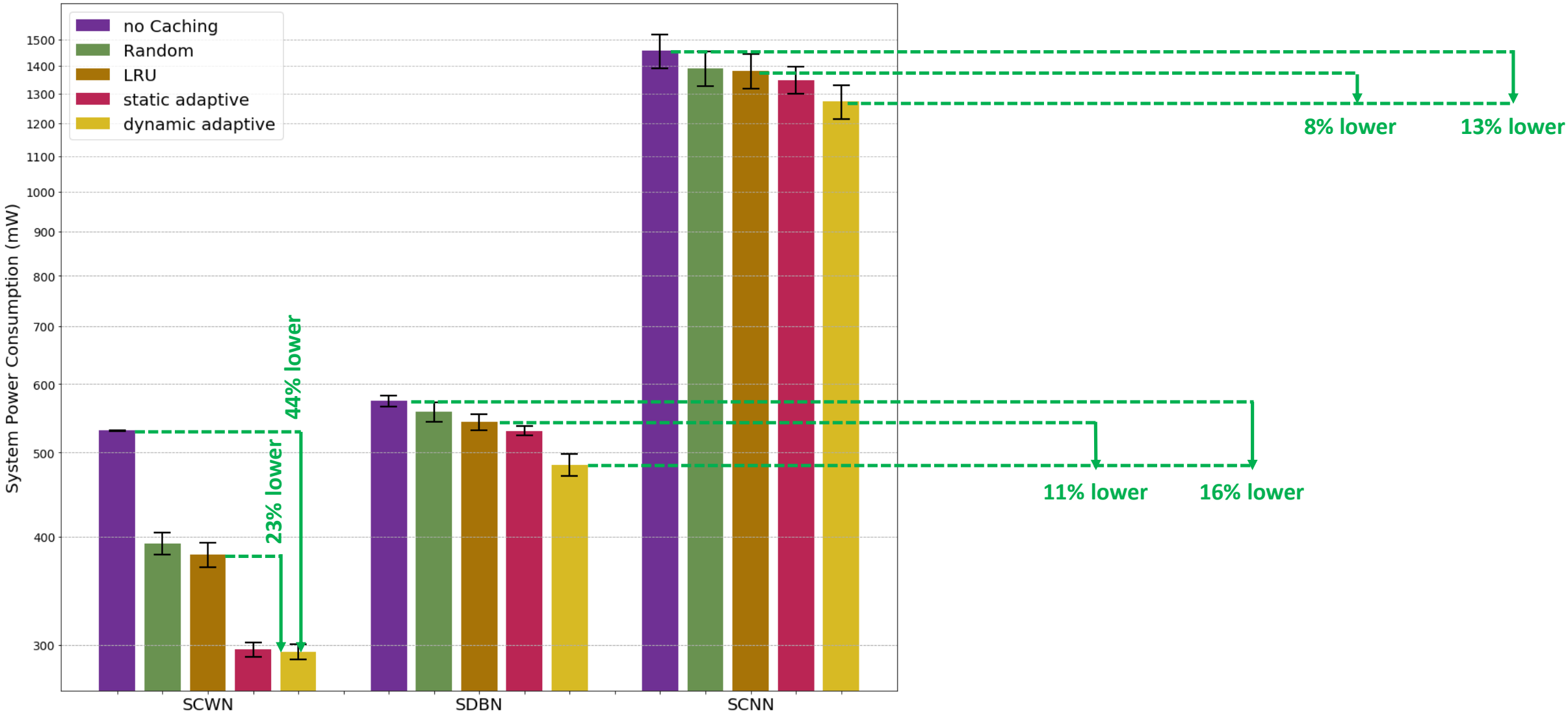
Experimental infrastructure



Experimental infrastructure



Results


no Caching
Random
LRU
static adaptive
dynamic adaptive
System Power Consumption (mW)
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
SCWN
SDBN
SCNN
44% lower
23% lower
8% lower
13% lower
11% lower
16% lower

Summary

- **SNNs** -> High efficiency, inherently temporal, hybridized for better accuracy
- **CyNAPSE** -> Reconfigurable neural dynamics, reconfigurable topology
- **Event-driven framework** -> forward visibility of memory accesses exploited
- **Power consumption** -> reduced by up to 44% over baseline and 23% over conventinal policies

References

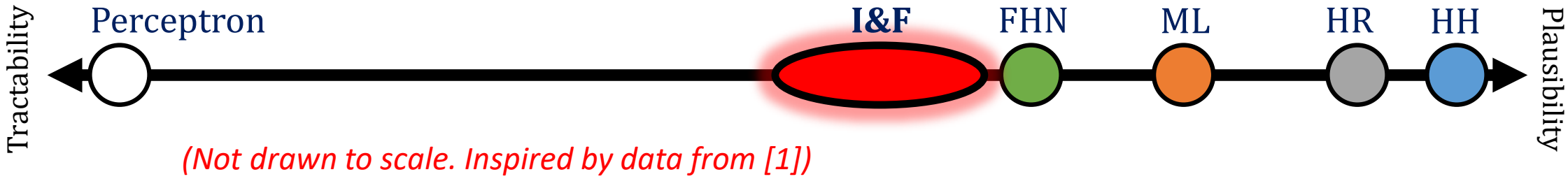
[1] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" IEEE transactions on neural networks, vol. 15, no. 5, pp. 1063–1070, 2004.
[2] F. Jug, "On competition and learning in cortical structures," Ph.D. dissertation, ETH Zurich, 2012.
[3] L. A. Belady, "A study of replacement algorithms for a virtual-storage computer," IBM Systems journal, vol. 5, no. 2, pp. 78–101, 1966.
[4] M. K. Qureshi, A. Jaleel, Y. N. Patt, S. C. Steely, and J. Emer, "Adaptive insertion policies for high performance caching," ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp. 381–391, 2007.
[5] S. M. Khan, Y. Tian, and D. A. Jimenez, "Sampling dead block prediction for last-level caches," in Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2010, pp. 175–186.
[6] S. Jiang and X. Zhang, "Lirs: an efficient low inter-reference recency set replacement policy to improve buffer cache performance," ACM SIGMETRICS Performance Evaluation Review, vol. 30, no. 1, pp. 31–42, 2002.
[7] D. F. Goodman and R. Brette, "The brian simulator," Frontiers in neuroscience, vol. 3, p. 26, 2009
[8] Y. Kim, W. Yang, and O. Mutlu, "Ramulator: A fast and extensible dram simulator," IEEE Computer architecture letters, vol. 15, no. 1, pp. 45–49, 2015.
[9] K. Chandrasekar, C. Weis, Y. Li, B. Akesson, N. Wehn, and K. Goossens, "Drampower: Open-source dram power & energy estimation tool," URL: http://www. drampower. info, vol. 22, 2012.
[10] S. Li, K. Chen, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "Cacti-p: Architecture level modeling for sram-based structures with advanced leakage reduction techniques," in Proceedings of the International Conference on Computer-Aided Design. IEEE Press, 2011, pp. 694–701.

Thank you!

Questions?

Backup Slides

Spiking Neuron model



(Not drawn to scale. Inspired by data from [1])

Generalized Leaky Integrate and Fire (LIF) model:

- Only **7 parameters** need fitting (τ_m, τ_{Na}, τ_K, g_l, V_{rest}, V_{reset} and t_{ref}) **instead of 20** for HH model.
- Biologically plausible parameters available in-vitro or in-vivo [2]
- Reconfigurable:
  -> For conversion to ***direct current-integration LIF***: use very small τ_{Na}, τ_K and skip voltage-gated ion-channels
  ->For conversion to ***perfect IF***: use above and arbitrarily large τ_m and/or zero g_l

Generalized LIF Neuron

Membrane time-constant

Constant Leak conductance

Na^+ ion channel conductance

K^+ ion channel conductance

$$\tau_m \frac{dV_m(t)}{dt} = -g_l(V_m(t) - V_{rest}) - g_{Na}(t)(V_m(t) - V_{r_{Na}}) - g_K(t)(V_m(t) - V_{r_K})$$

Na^+ time-constant

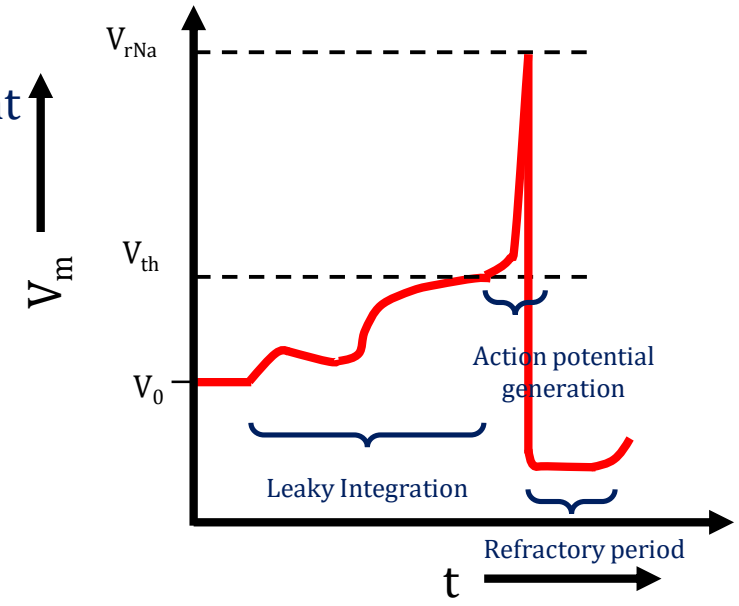
$$\tau_{Na} \frac{dg_{Na}(t)}{dt} = -g_{Na}(t) + I_{syn-exc}(t)$$

Excitatory synaptic current

K^+ time-constant

$$\tau_K \frac{dg_K(t)}{dt} = -g_K(t) + I_{syn-inh}(t)$$

Inhibitory synaptic current



Neuron activity (all-or-nothing/digital)

$$S_i(t) = \begin{cases} 0, & V_m(t) < V_{th} \\ 1, & V_m(t) \geq V_{th} \end{cases}$$

Reset (hyperpolarized) voltage

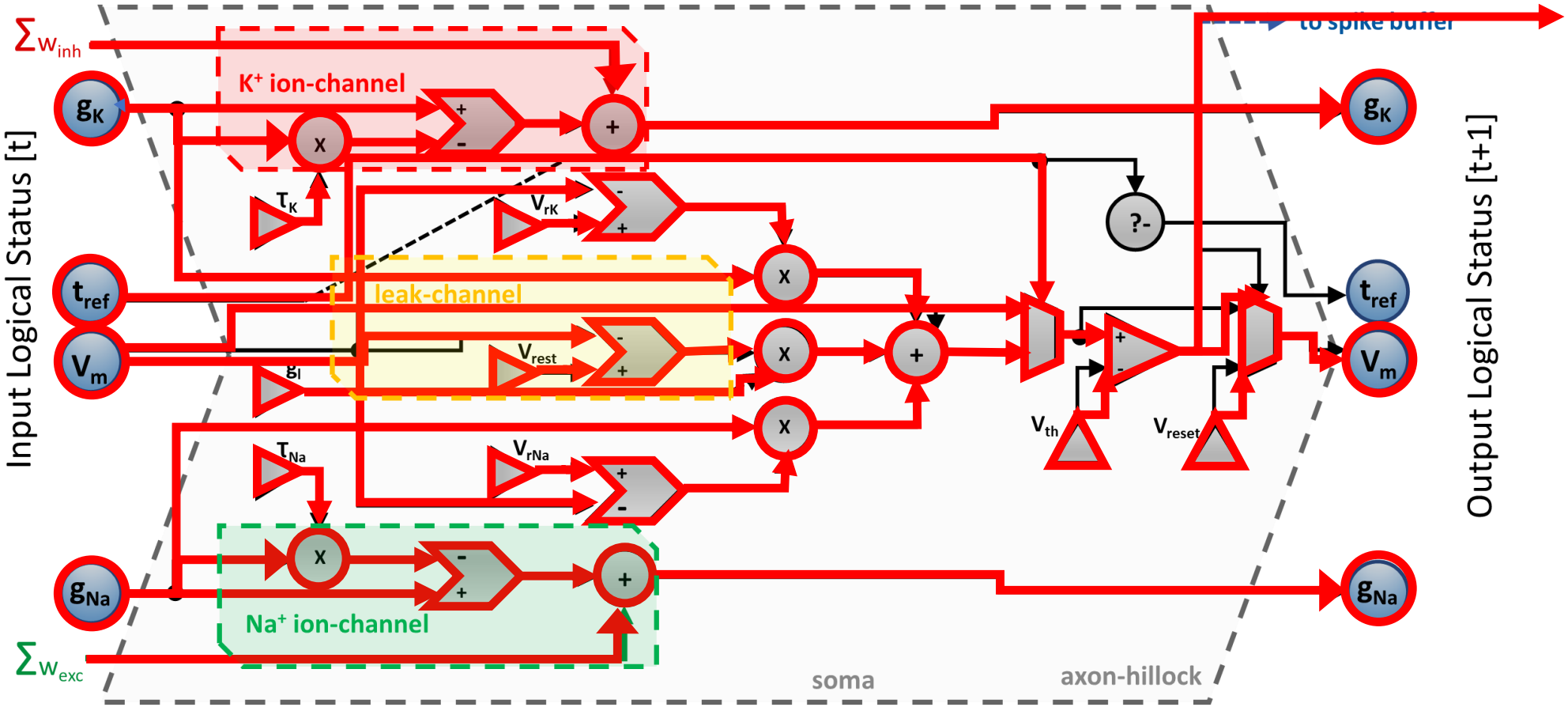
$$V_m(t) = \begin{cases} V_{reset} \\ V_m(t), \end{cases} \quad t_n \leq t \leq (t_n + t_{ref}) \quad otherwise$$

Refractory period

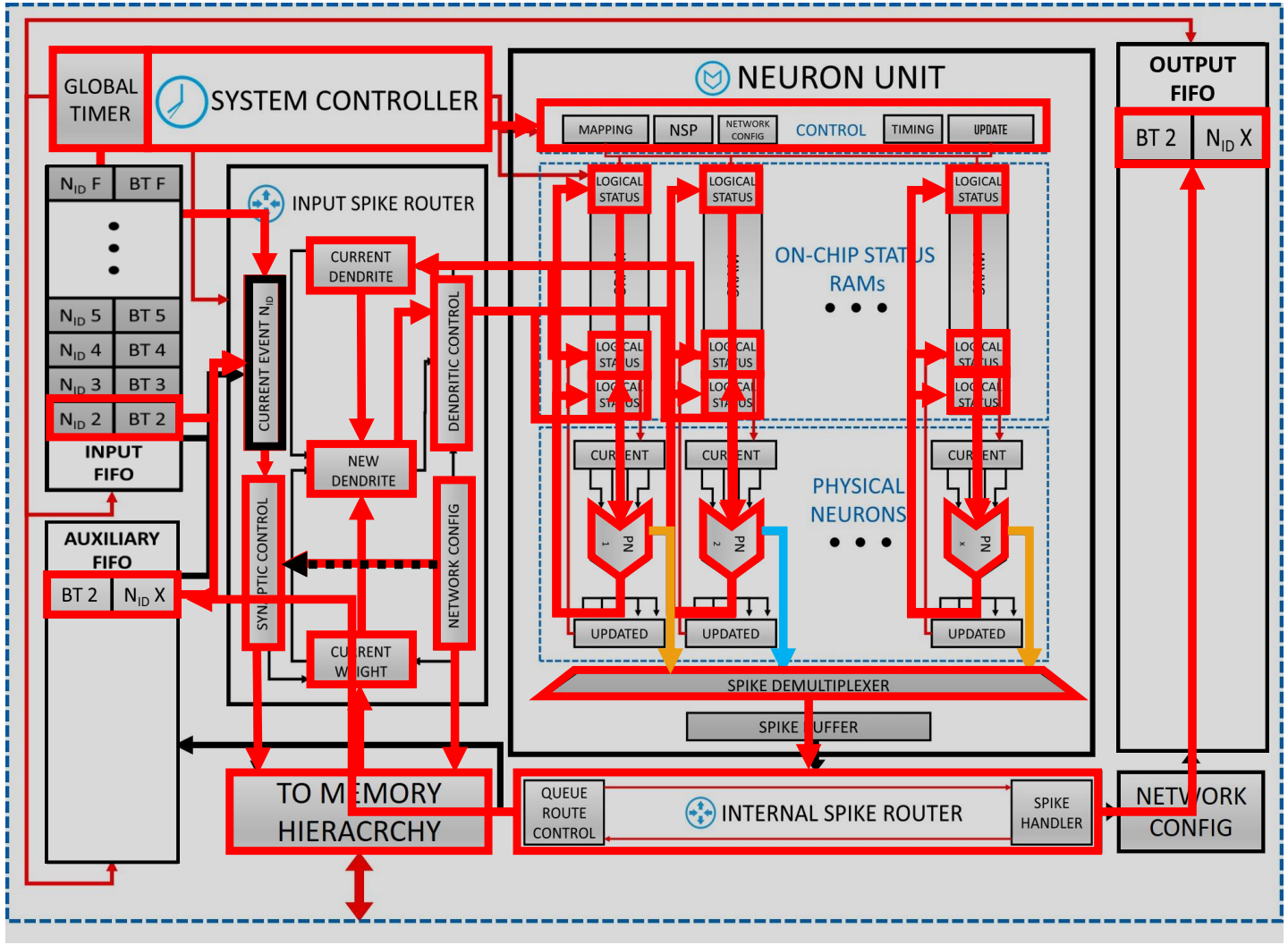
$$[for\ a\ spike\ train\ S_i(t) = \sum_n \delta(t - t_n)\ ]$$

Full-custom Silicon Neuron

$$V_m[t+1] = V_m[t] + g_{Na}[t] \times (V_m[t] - V_{r_{Na}}) \quad \text{(unless refractory)}$$



High-level system operation

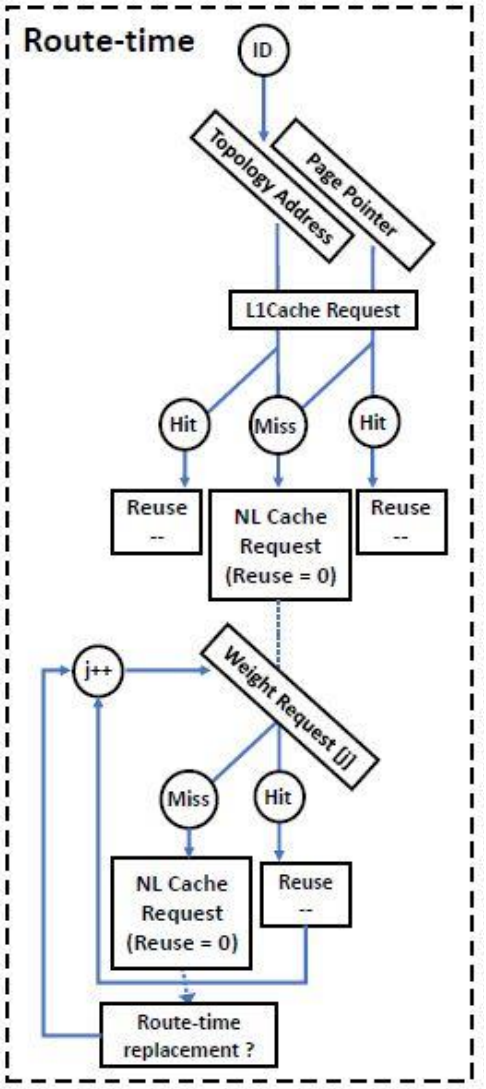


Proposed management scheme

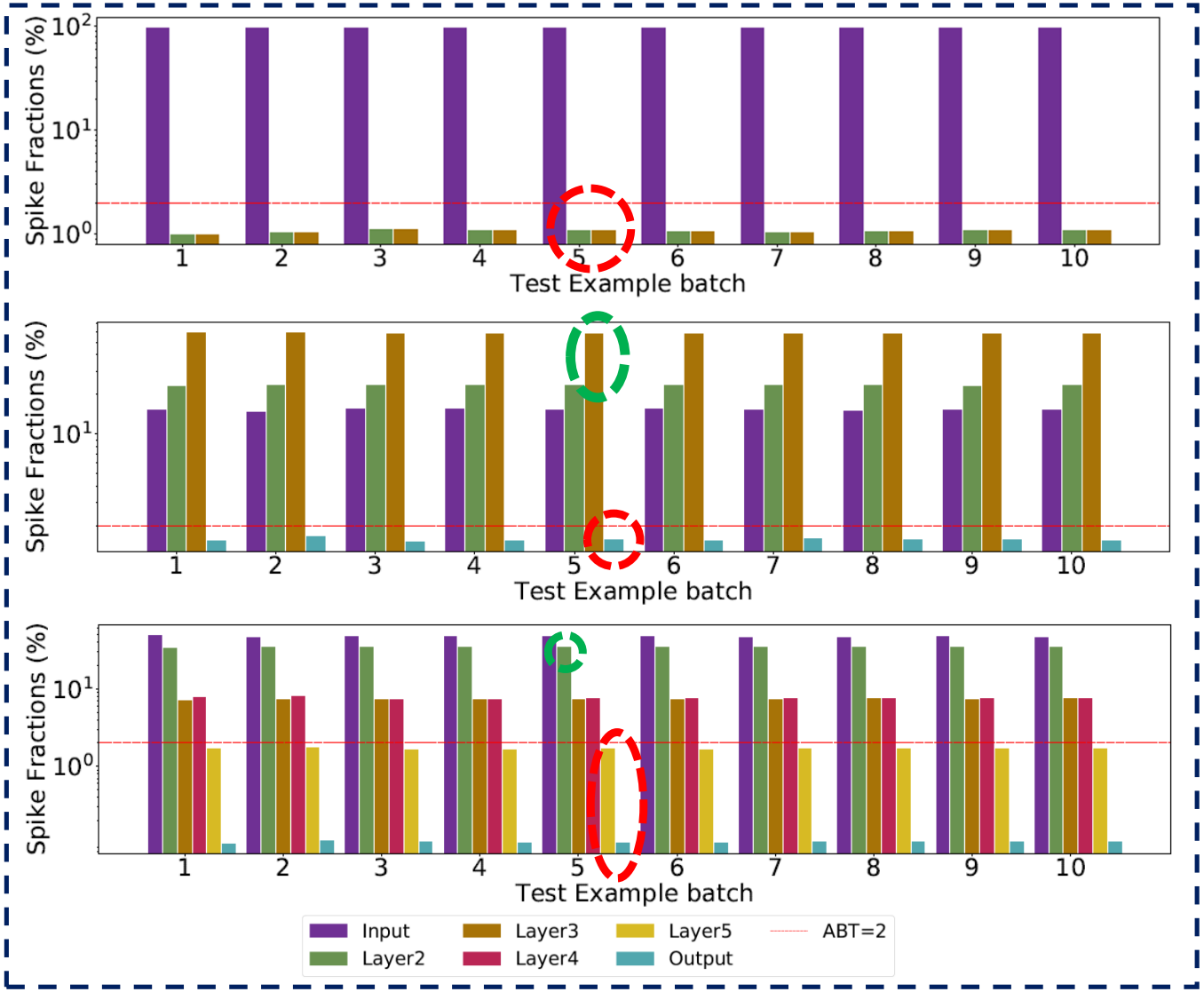
Cache replacement scenarios:

- Compulsory miss at warm-up and event read-time
- Hit at read-time
- Capacity/Conflict miss at read-time (read-time replacements)
  - -> Conservative Approach
  - -> Aggressive Approach
  - -> Intelligent Approach
    (reuse threshold)

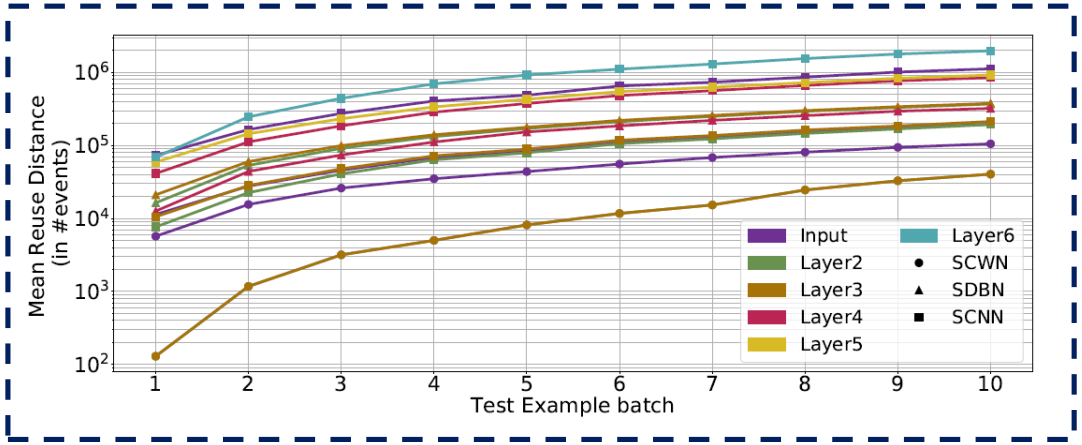
- Compulsory miss at route-time
- Hit at route-time
- Policy miss at route-time



Dynamic kernel statistics

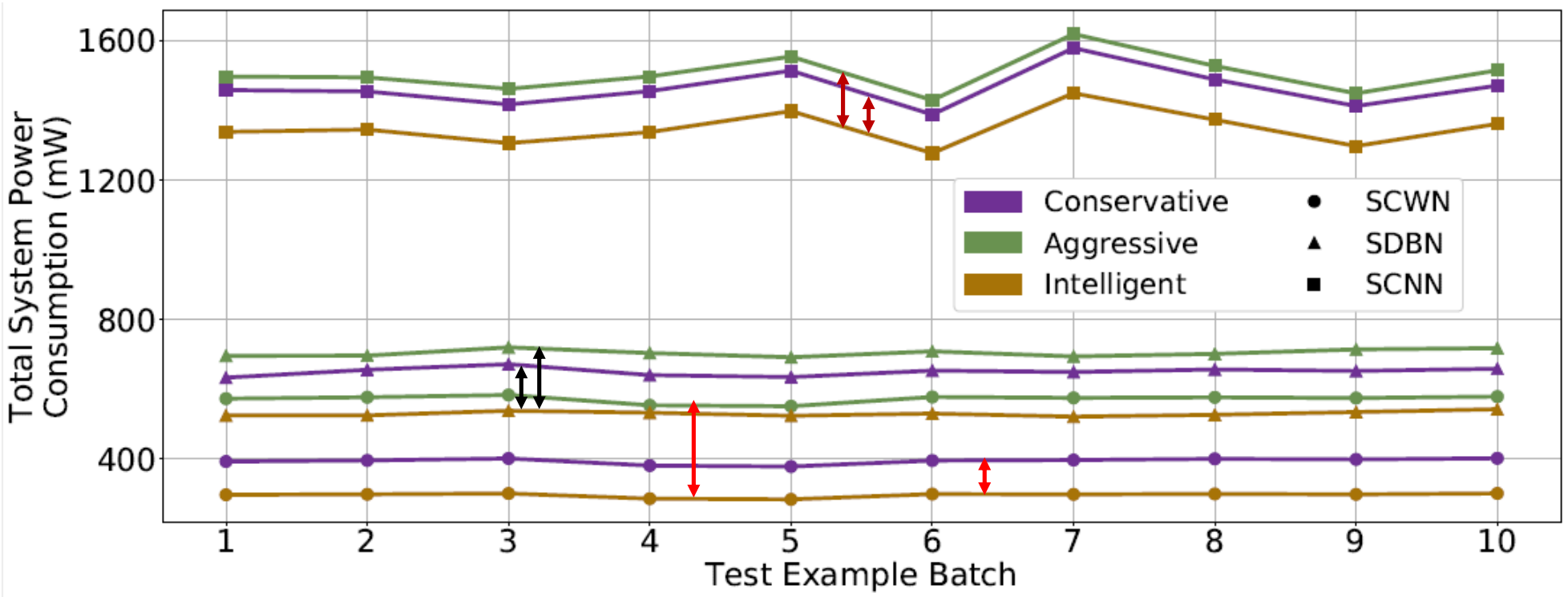


◀ **Layer-wise spike fractions** of (from Top) SCWN, SDBN and SCNN



▲ **Layer-wise mean reuse distance** of neurons in the benchmarks

Read-time replacements



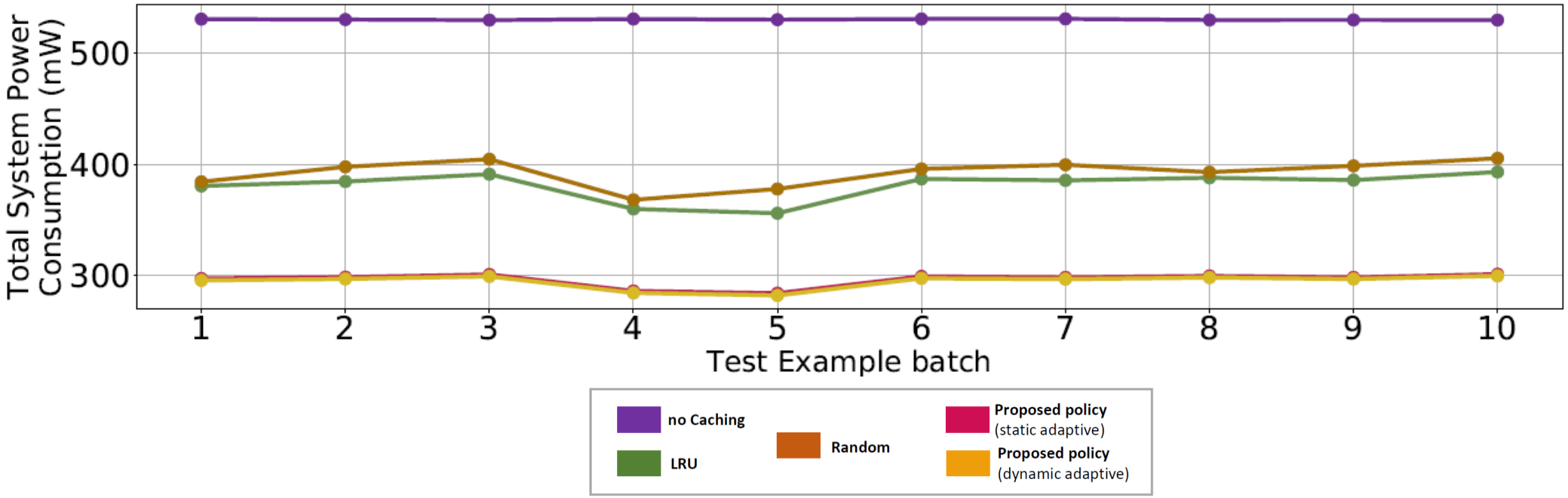
System Power Consumption: **Intelligent < Conservative < Aggressive** (for all three benchmarks)

System Power Consumption loss: **SCWN > SDBN > SCNN**

Verdict: Use Intelligent for all benchmarks

LRU vs Random vs Proposed policy

SCWN

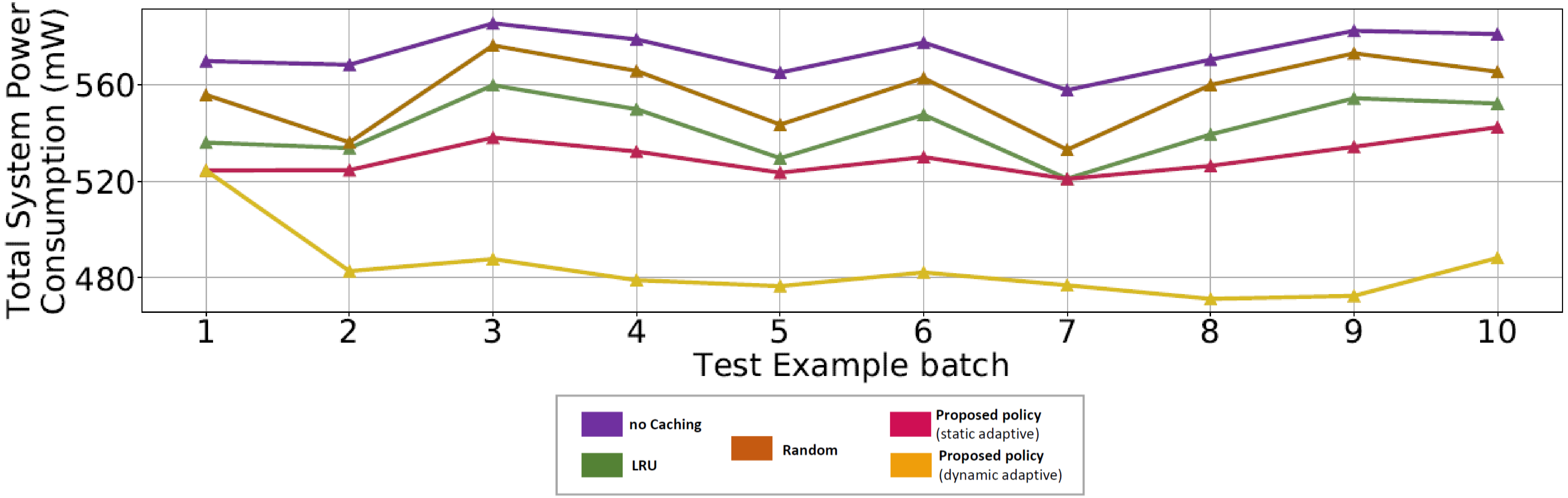


Low activity (2.144/Δt) – high reuse network → Static adaptive scheme performs LRU and Random can exploit intra-reasonably well. Not much gain comes stimulus reuse but not extra-stimulus from dynamic kernel information (only reuse 2.2% activity)

Largely dominated (97.8%) by input events →

LRU vs Random vs Proposed policy

SDBN

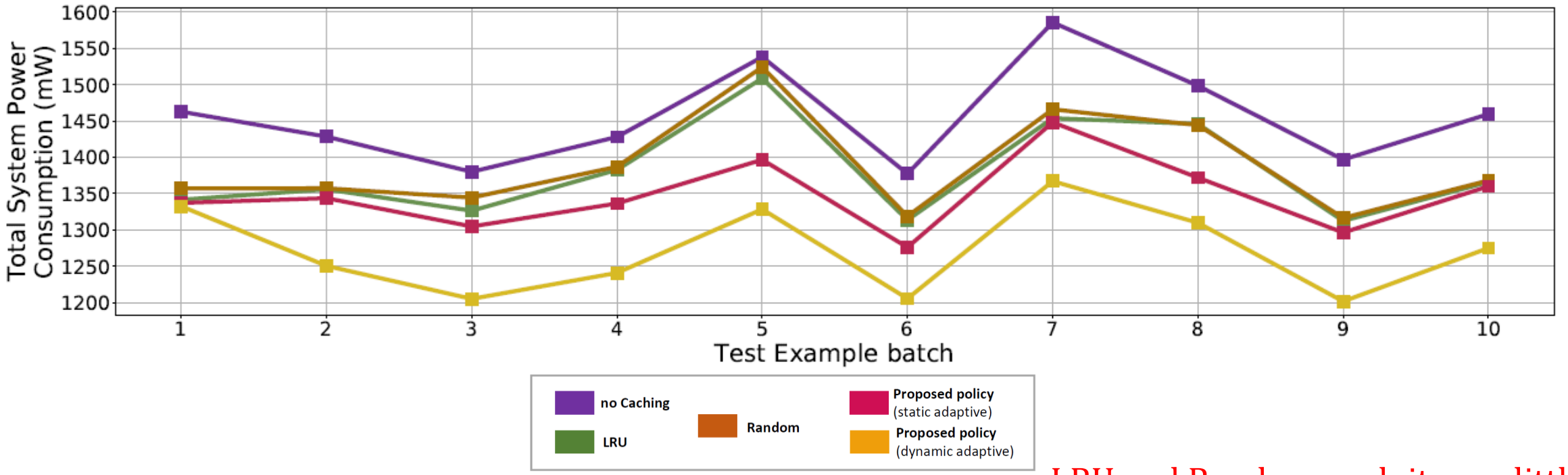


Low input activity (3.99/ Δt) –lower reuse network → LRU and Random can exploit most temporal locality quite well and relative benefits from static adaptive scheme is quite modest.

High internal activity (esp. Layer 3) → Third layer neurons benefit greatly from line protection and therefore marked relative savings in power consumption is observed

LRU vs Random vs Proposed policy

SCNN



Very high input activity (219/Δt) - very low reuse network → LRU and Random exploit very little temporal locality but static-adaptive policy targets a better fraction of input activity

High internal activity (esp. Layer 3) → Third layer neurons benefit greatly from line protection

Relative benefits from Static to Dynamic: **SCNN < SDBN** → Sparse activations in conv2D layers.

LRU vs Random vs Proposed policy

Summary

Benchmark	LRU v/s baseline	Random v/s baseline	Proposed Policy (static adaptive) v/s baseline	Proposed Policy (dynamic adaptive) v/s baseline	Proposed Policy v/s LRU
SCWN	28.13%	25.99%	44.13%	44.45%	22.71%
SDBN	5.46%	2.88%	7.65%	15.55%	10.67%
SCNN	5.12%	4.59%	7.4%	12.61%	7.9%

Scope of future work

Architectural enhancements:

- Multi-core CyNAPSE : interconnects and multi-level memory hierarchy
- Core leakage control techniques
- Compiler driven optimizations/Better dataflow for SNNs
- Is Proposed policy applicable to any event-driven simulation framework?

Learning:

- We are interested in spike driven STDP hardware using memristive devices
- Evolving SNNs: benefits of hardware acceleration is still not clear.
- Extending CyNAPSE stack up to parsers and down to motor control or BCI.