How to Efficiently Learn On-Device?

PRIYA PANDA

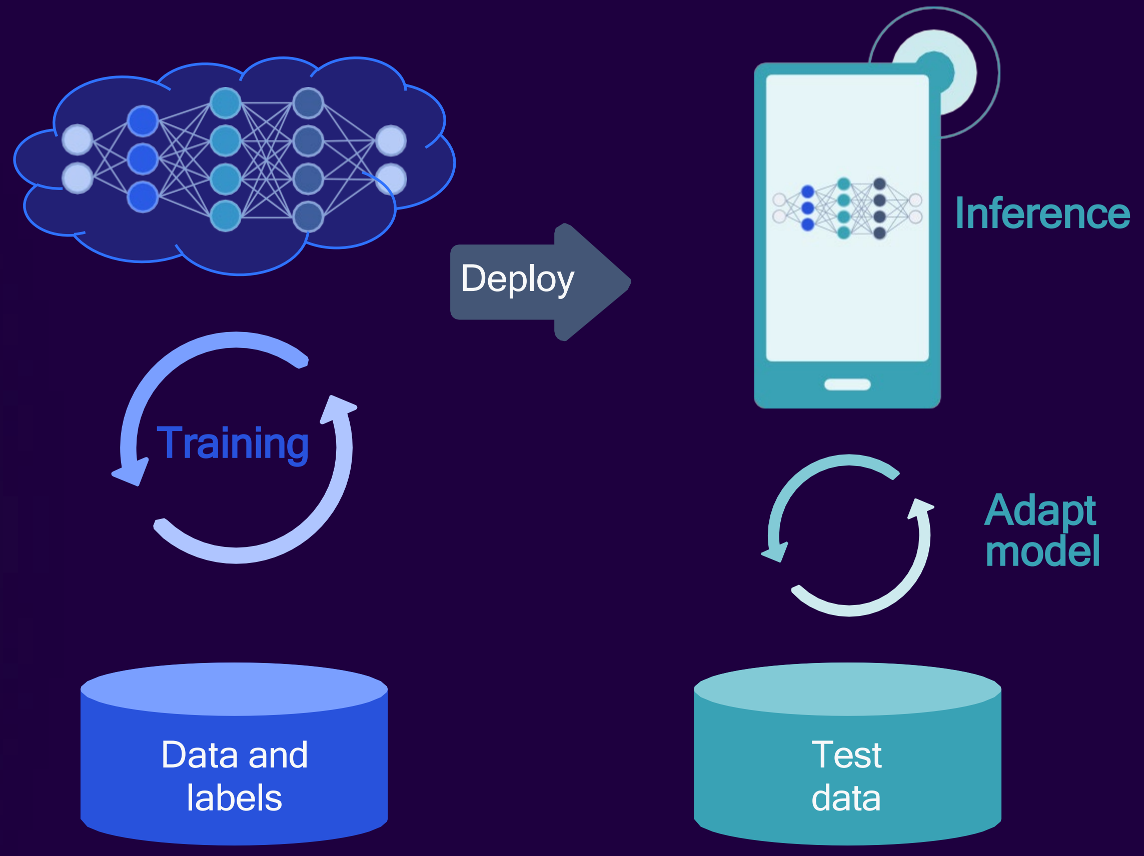
Yale University, USA

https://intelligentcomputinglab.yale.edu/

priya.panda@yale.edu

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

What is on-device learning?



Offline training
A model is trained in the cloud with data reflecting the target application

On-device learning
Modifying model after deployment based on the test environment

Important considerations for on-device learning

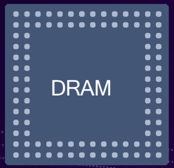
Benefits

- Better examples than training dataset for personalization
- Ability to run with smaller models that adapt to the target data
- Preservation of privacy during model development

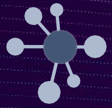
Challenges

- Learning with Backprop is computationally demanding
- Limited compute, storage, and/or power
- Local data can be limited, e.g., noisy labels and class imbalance
- Adversarial attacks to training
- Overfitting or catastrophic forgetting

Backprop training requirements

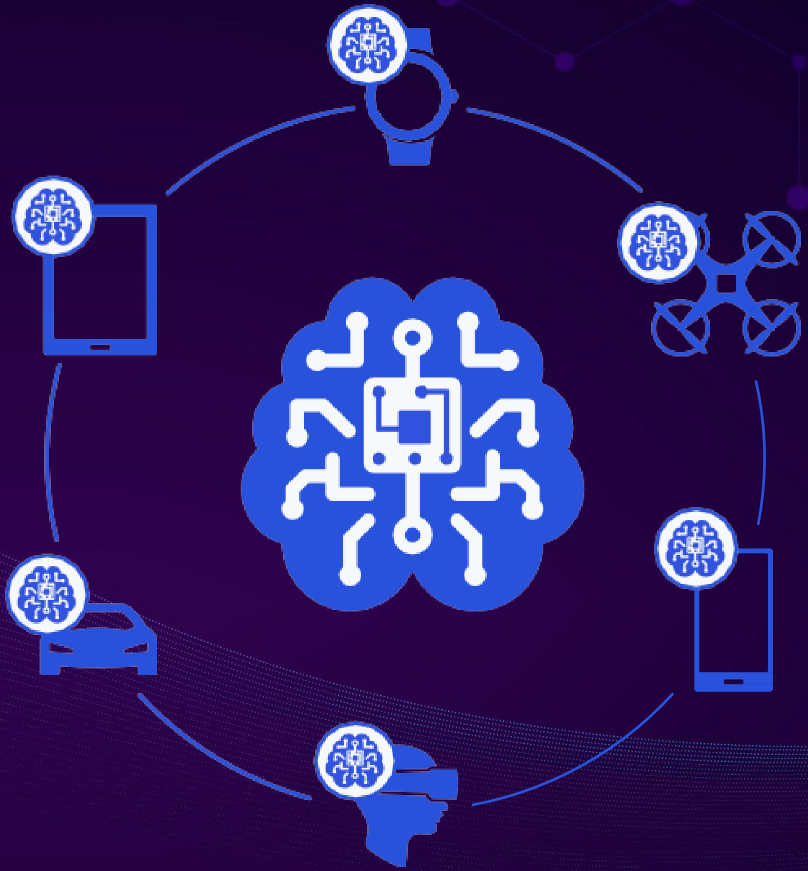


Large memory



Training runtime

High precision



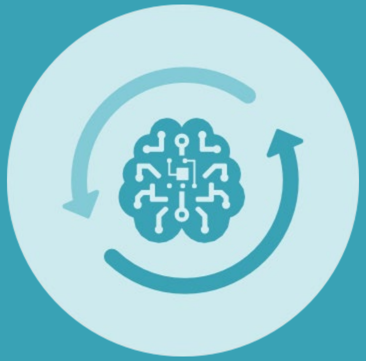
- These requirements cannot be met by battery and memory limited edge devices

Our research aims at addressing the key challenges of on-device learning

[Yale Univ.]

Model-aware learning

How to use learnt model's information to learn new data



Data-aware Learning with Spikes

How to use Spiking Neural Networks to learn based on input data difficulty

Hardware-aware learning

How to implement on-device learning to improve efficiency of hardware resources

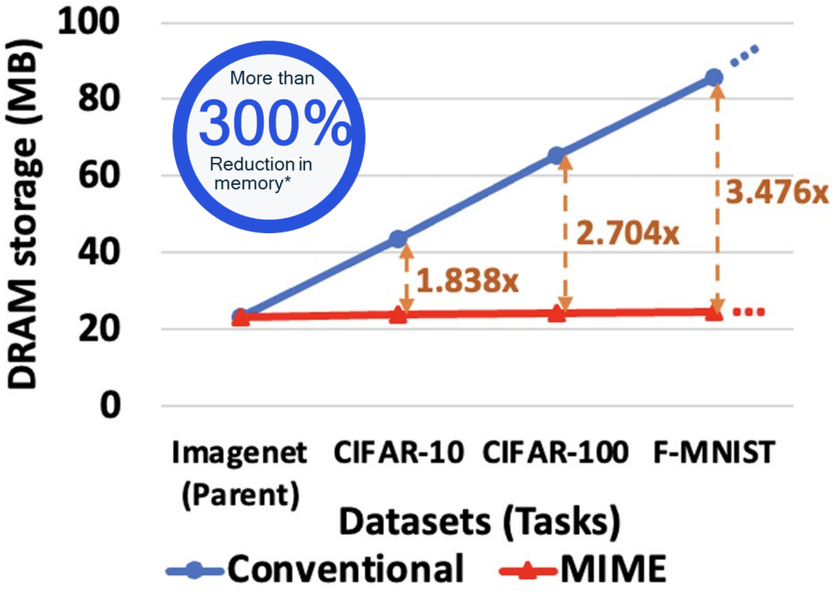
Reduce backprop complexity with model aware learning

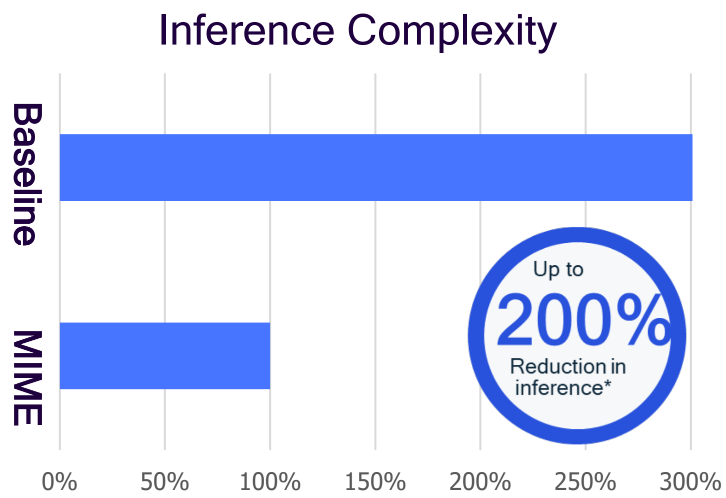
- Correlation between pre-trained model and on-device data determines computation
- Dynamic gradient computation adjustment

DARPA ShELL Accomplishments:

- >300% reduction in memory during training
- ~200% reduction in inference complexity
- Competitive accuracy with baseline

[Yale Univ.]

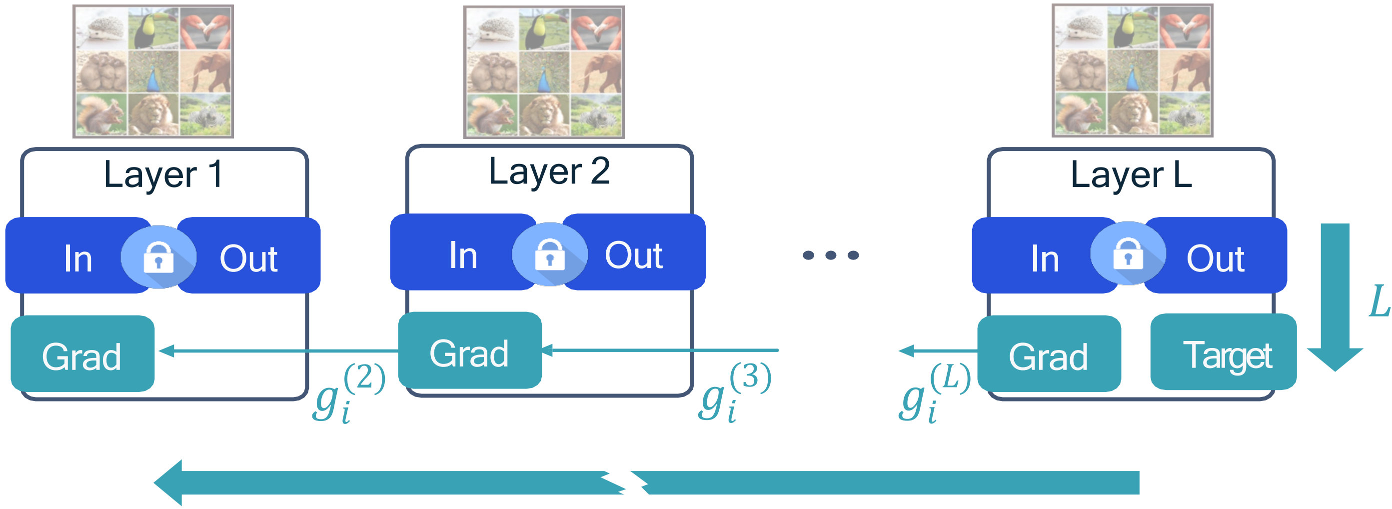




[MIME: Bhattacharjee et al., DAC 2022]

Correlated

Un-Correlated



Backward pass with weights frozen and activation threshold learnt with gradient computation

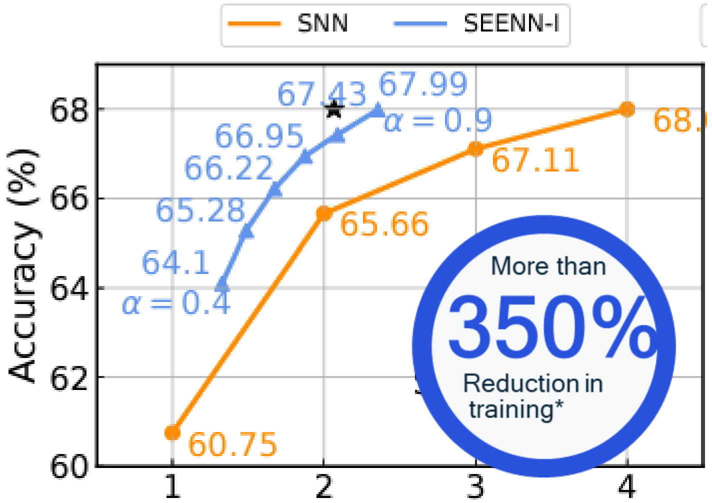
Reduce backprop complexity with data aware learning

- Easy vs. Difficult data determine the temporal compute effort required during training of neuromorphic spiking neural networks

Ongoing JUMP2.0 (CoCoSys-T4):

- >350% reduction in training latency per image
- >70% of inputs can be classified early (Easy inputs are in larger concentration in real-world datasets)
- Iso (or higher) accuracy than baseline

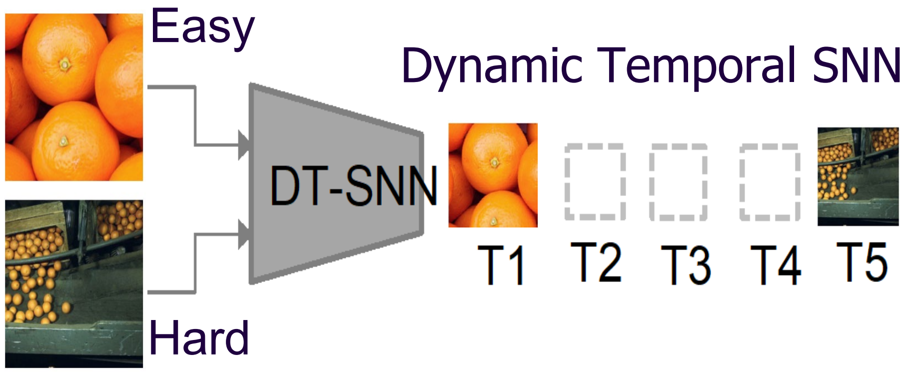
[Yale Univ.]



[Li et al., DAC 2023; Li et al., arXiv:2304.01230v1]

EASY
Less time, and effort

DIFFICULT
More time, and effort



Reduce backprop complexity with hardware aware learning

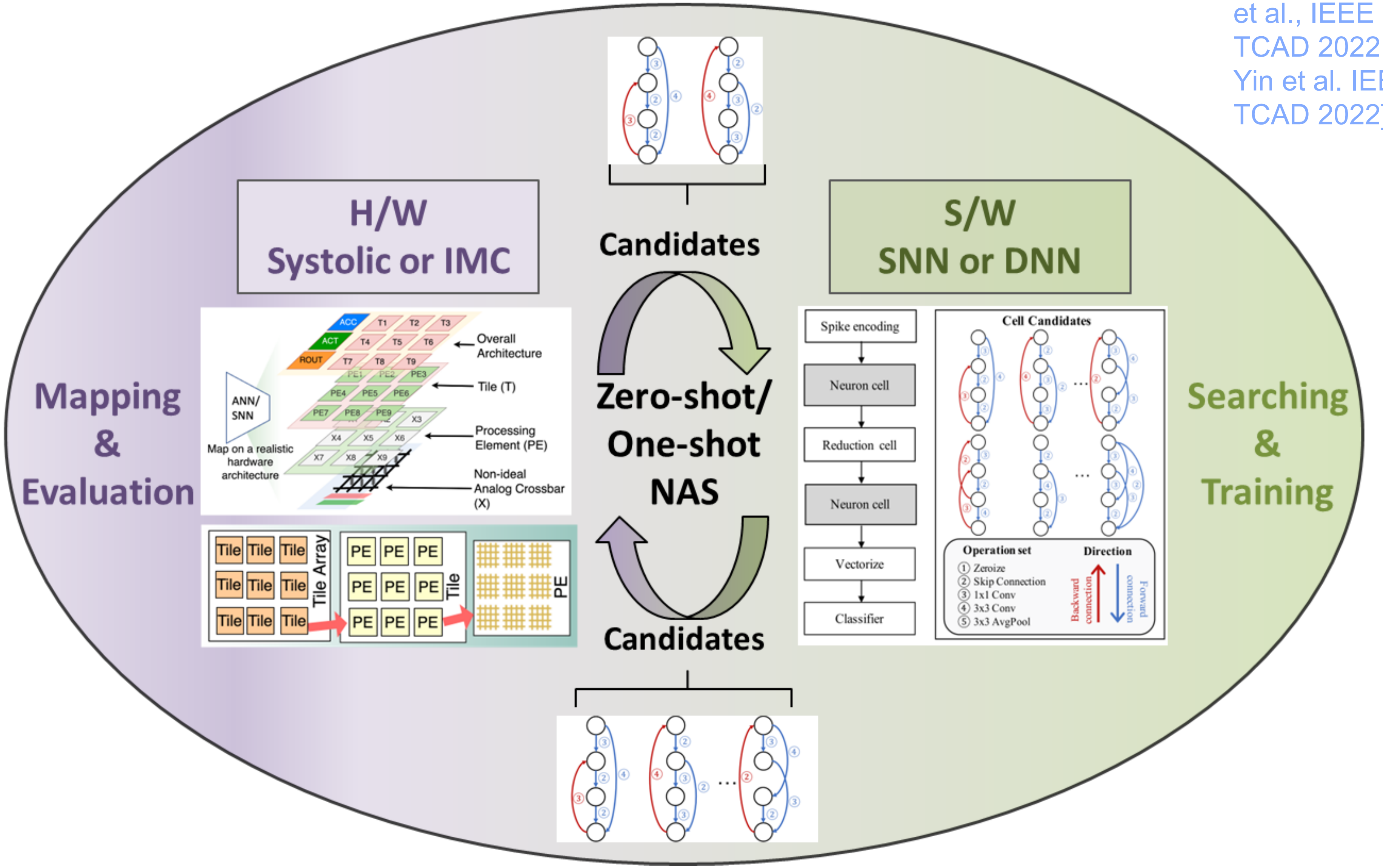
- Hardware and Algorithm Co-exploration with Neural Architecture Search (NAS)

DARPA YFA 2023:
Prelim evaluations on IMC

- >150% higher GOPS/s
- ~400% lower power than state-of-the-art IMC accelerators

[Yale Univ.]

[Moitra et al., DAC 2023; Bhattacharjee et al., IEEE TCAD 2022; Yin et al. IEEE TCAD 2022]



Distribution Statement A – Approved for public release. Distribution unlimited.

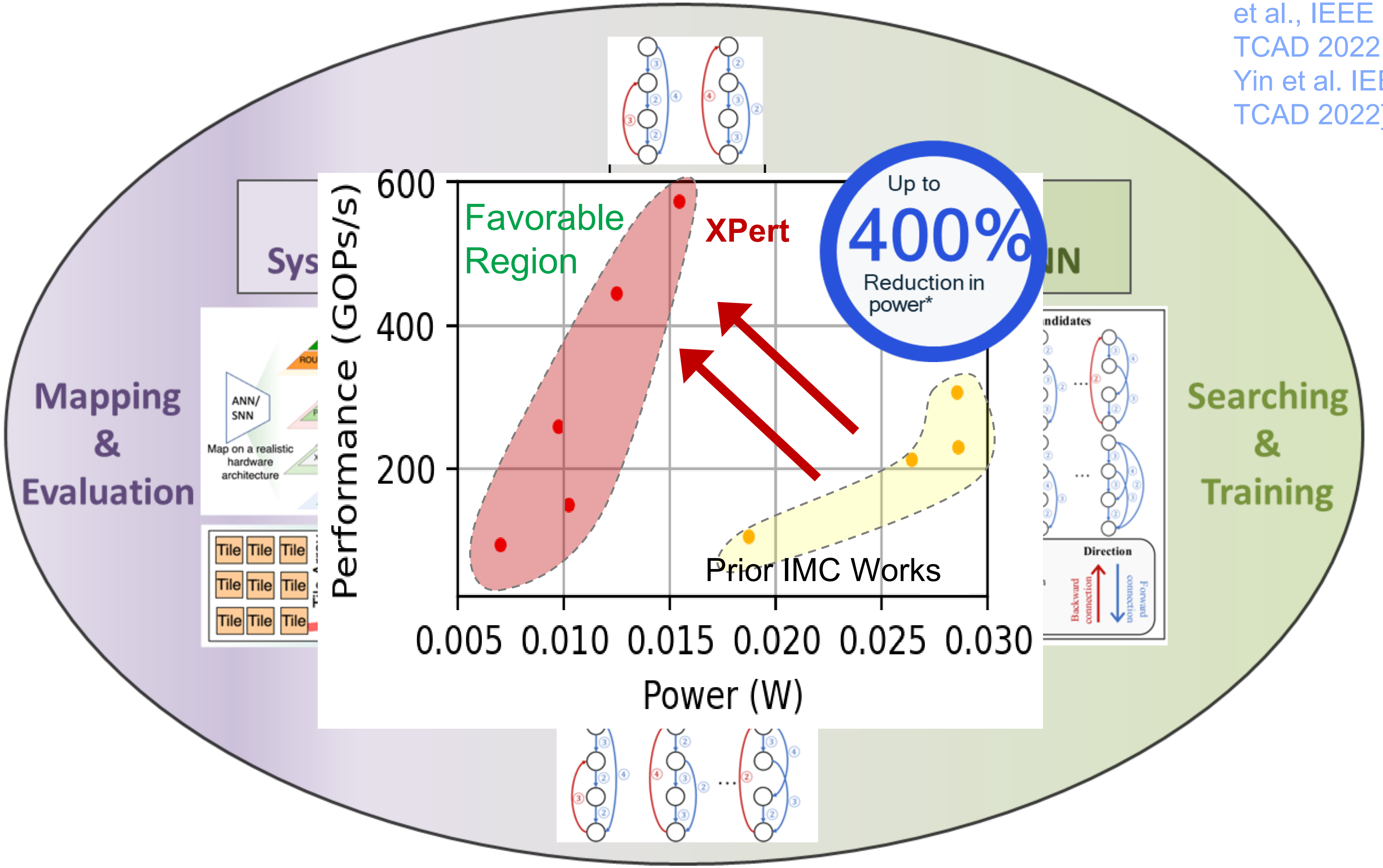
Reduce backprop complexity with hardware aware learning

- Hardware and Algorithm Co-exploration with Neural Architecture Search (NAS)

DARPA YFA 2023:
Prelim evaluations on IMC

- >150% higher GOPS/s
- ~400% lower power than state-of-the-art IMC accelerators

[Yale Univ.]



[Moitra et al., DAC 2023; Bhattacharjee et al., IEEE TCAD 2022; Yin et al. IEEE TCAD 2022]

Summary

Algorithm-Hardware Co-Design combining model awareness, data awareness and hardware awareness will substantially impact on-device learning.

Model-aware learning

More than
300%
Reduction in memory*

Data-aware learning with Spikes

More than
350%
Reduction in training*

Hardware-aware learning

Up to
400%
Reduction in power*

[Yale Univ.]