Tailored Computing: Domain-Specific Architectures for Embodied Autonomous Machines

Zishen Wan

PhD Student @ School of ECE, Georgia Institute of Technology

Web: https://zishenwan.github.io
Email: zishenwan@gatech.edu

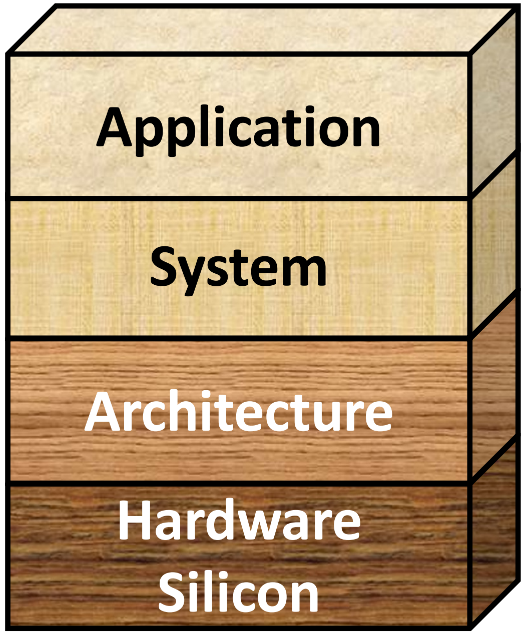
Coordinated Science Laboratory @ University of Illinois Urbana-Champaign (UIUC), Feb. 25, 2025

Bio

Presenter: Zishen Wan
- PhD Student at Georgia Tech
- Advised by Prof. Arijit Raychowdhury and Prof. Tushar Krishna

Webpage: https://zishenwan.github.io

Research Interest
(Cross-Layer Co-Design)



Autonomous systems
Embodied agents
Neuro-symbolic AI

Domain-specific system and architecture

FPGA prototype
ASIC tapeout

Autonomous Machines Era

- Autonomous Machines on the Rise

Self-Driving Cars


Drones

Legged Robot


AR/VR


Embodied AI Robot

- Wide Application Potential

Package Delivery

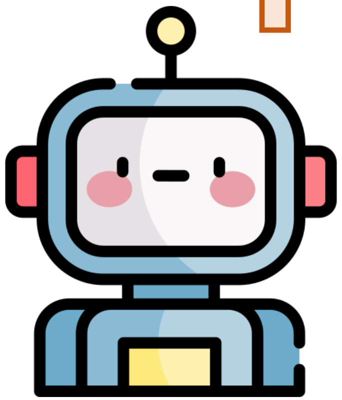

Search & Rescue


Agriculture


Manufacture


Space

Autonomous Machines (Agentic System)



Cognition Capability

Human-like reasoning
Trustworthy decision making
Human-agent interaction

Autonomy Capability

Perception, Localization,
Mapping, Planning, Control,
Learning-based navigation

Requirements

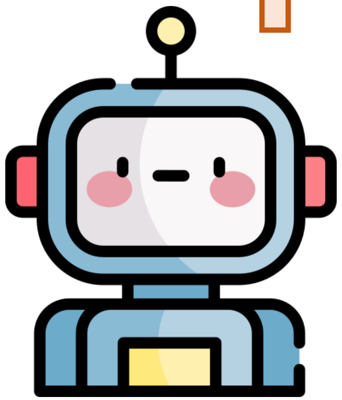
Real-Time ***Performance***

Energy ***Efficiency***

Design ***Scalability***

Safety-Critical ***Reliability***

Autonomous Machines (Agentic System)



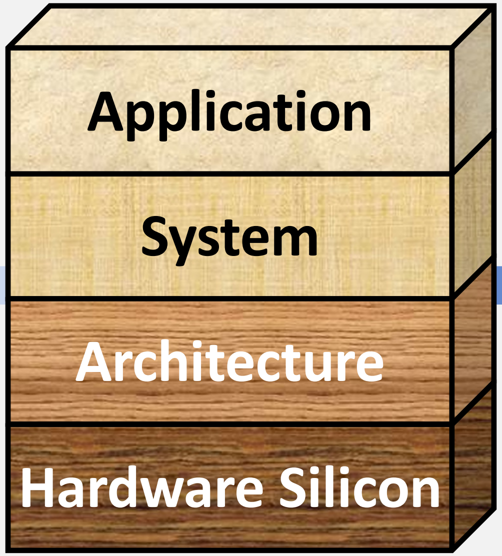
Cognition Capability

Human-like reasoning
Trustworthy decision making
Human-agent interaction

Autonomy Capability

Perception, Localization,
Mapping, Planning, Control,
Learning-based navigation

My Research:
Cross-Layer Approach



Requirements

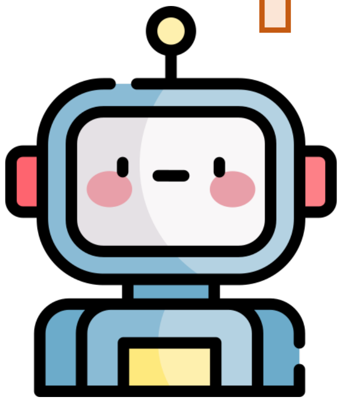
Real-Time ***Performance***

Energy ***Efficiency***

Design ***Scalability***

Safety-Critical ***Reliability***

Autonomous Machines (Agentic System)



Cognition Capability

Human-like reasoning

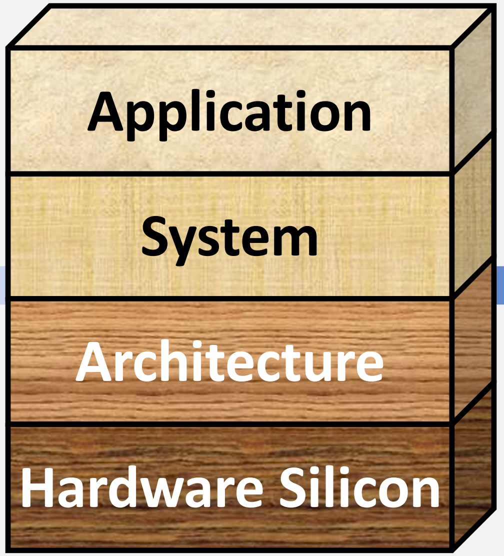
Trustworthy decision making

Human-agent interaction

Autonomy Capability

Perception, Localization,
Mapping, Planning, Control,
Learning-based navigation

My Research:
Cross-Layer Approach



Requirements

Real-Time ***Performance***

Energy ***Efficiency***

Design ***Scalability***

Safety-Critical ***Reliability***

Current Neural Networks in Our Daily Life

Image Recognition

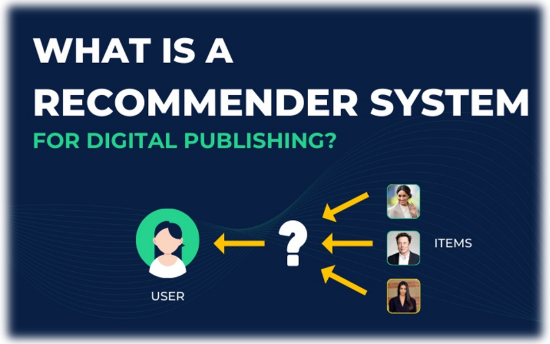
Speech Recognition

Language Translation

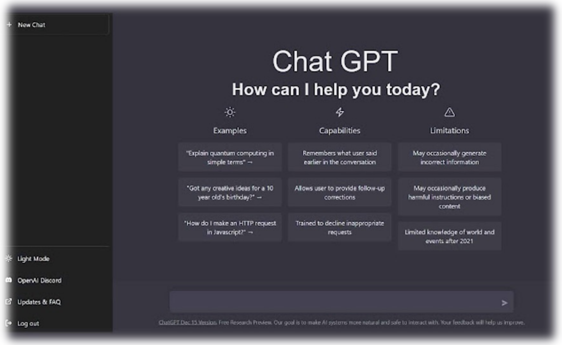
Autonomous Vehicle

Medical Diagnosis

Financial Services

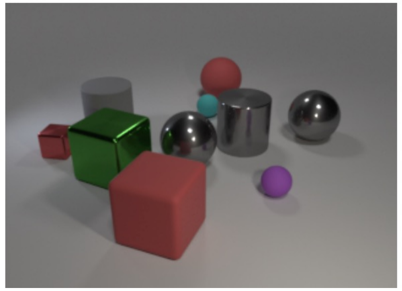


Recommendation Systems



ChatGPT

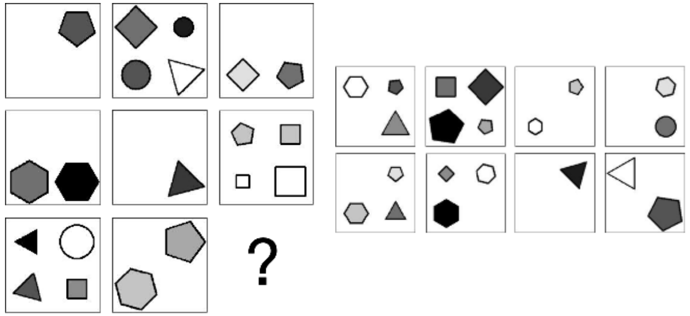
But... Is That Enough?



(i) *Remove all gray spheres. How many spheres are there? (3)*, (ii) *Take away 3 cubes. How many objects are there? (7)*, (iii) *How many blocks must be removed to get 1 block? (2)*

Complex Question Answering

NN accuracy: 50%

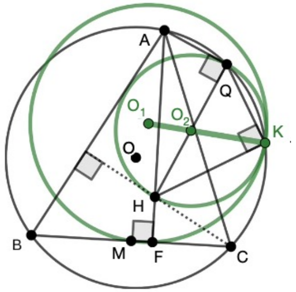


Abstract Reasoning

NN accuracy: 53%

IMO 2015 P3

"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that QH ⊥ QA and let K be the point on (O) such that KH ⊥ KQ. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."



Automated Theorem Proving

NN accuracy: 0%

Interactive Learning

Scenario
Imagine that a stranger will give Hank one thousand dollars to break all the windows in his neighbor's house without his neighbor's permission. Hank carries out the stranger's request.
Imagine that there are five people who are waiting in line to use a single-occupancy bathroom at a concert venue. Someone at the back of the line needs to throw up immediately. That person skips to the front of the line instead of waiting in the back.
At a summer camp, there is a pool. Right next to the pool is a tent where the kids at the camp have art class. The camp made a rule that there would be no cannonballing in the pool so that the art wouldn't get ruined by the splashing water. Today, there is a bee attacking this kid, and she needs to jump into the water quickly. This kid cannonballs into the pool.

Ethical Decision Making

NN accuracy: 65%

Farmer John has N cows ($2 \leq N \leq 10^5$). Each cow has a breed that is either Guernsey or Holstein. As is often the case, the cows are standing in a line, numbered $1 \cdots N$ in this order.

Over the course of the day, each cow writes down a list of cows. Specifically, cow i's list contains the range of cows starting with herself (cow i) up to and including cow E_i ($i \leq E_i \leq N$).

FJ has recently discovered that each breed of cow has exactly one distinct leader. FJ does not know who the leaders are, but he knows that each leader must have a list that includes all the cows of their breed, or the other breed's leader (or both).

Help FJ count the number of pairs of cows that could be leaders. It is guaranteed that there is at least one possible pair.

Problem

Competitive Programming

NN accuracy: 8.7%

But… Is That Enough?

(i) Remove all gray spheres. How many spheres are there? (3), (ii) Take away 3 cubes. How many objects are there? (7), (iii) How many blocks must be removed to get 1 block? (2)

Complex Question Answering

NN accuracy: 50%

Abstract Reasoning

NN accuracy: 53%

IMO 2015 P3

"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that QH ⊥ QA and let K be the point on (O) such that KH ⊥ KQ. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."

Automated Theorem Proving

NN accuracy: 0%

Neuro-Symbolic AI

Interactive Learning

NN accuracy: 71%

Scenario

Imagine that a stranger will give Hank one thousand dollars to break all the windows in his neighbor's house without his neighbor's permission. Hank carries out the stranger's request.

Imagine that there are five people who are waiting in line to use a single-occupancy bathroom at a concert venue. Someone at the back of the line needs to throw up immediately. That person skips to the front of the line instead of waiting in the back.

At a summer camp, there is a pool. Right next to the pool is a tent where the kids at the camp have art class. The camp made a rule that there would be no cannonballing in the pool so that the art wouldn't get ruined by the splashing water. Today, there is a bee attacking this kid, and she needs to jump into the water quickly. This kid cannonballs into the pool.

Ethical Decision Making

NN accuracy: 65%

Farmer John has N cows ($2 \le N \le 10^5$). Each cow has a breed that is either Guernsey or Holstein. As is often the case, the cows are standing in a line, numbered $1 \cdots N$ in this order.

Over the course of the day, each cow writes down a list of cows. Specifically, cow i's list contains the range of cows starting with herself (cow i) up to and including cow E_i ($i \le E_i \le N$).

FJ has recently discovered that each breed of cow has exactly one distinct leader. FJ does not know who the leaders are, but he knows that each leader must have a list that includes all the cows of their breed, or the other breed's leader (or both).

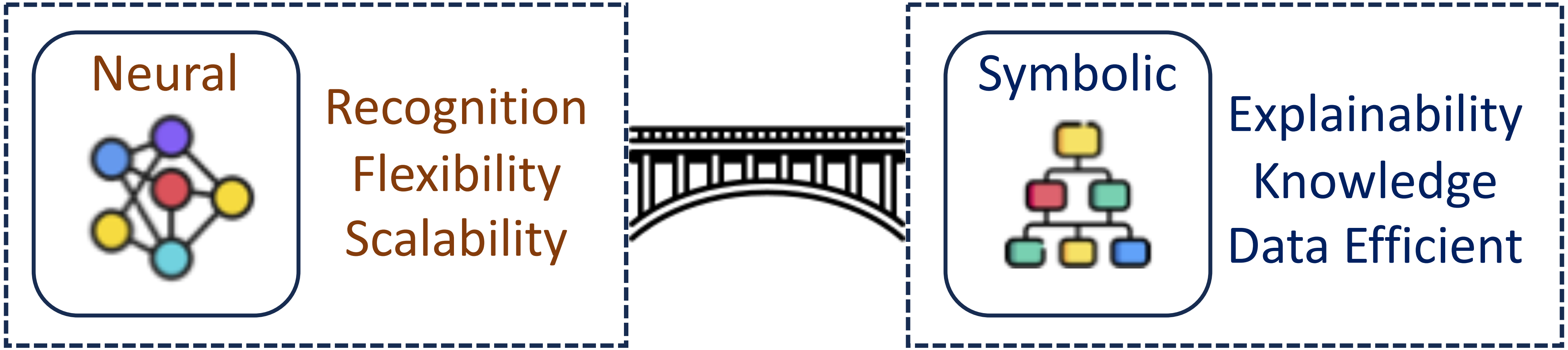
Help FJ count the number of pairs of cows that could be leaders. It is guaranteed that there is at least one possible pair.

Problem

Competitive Programming

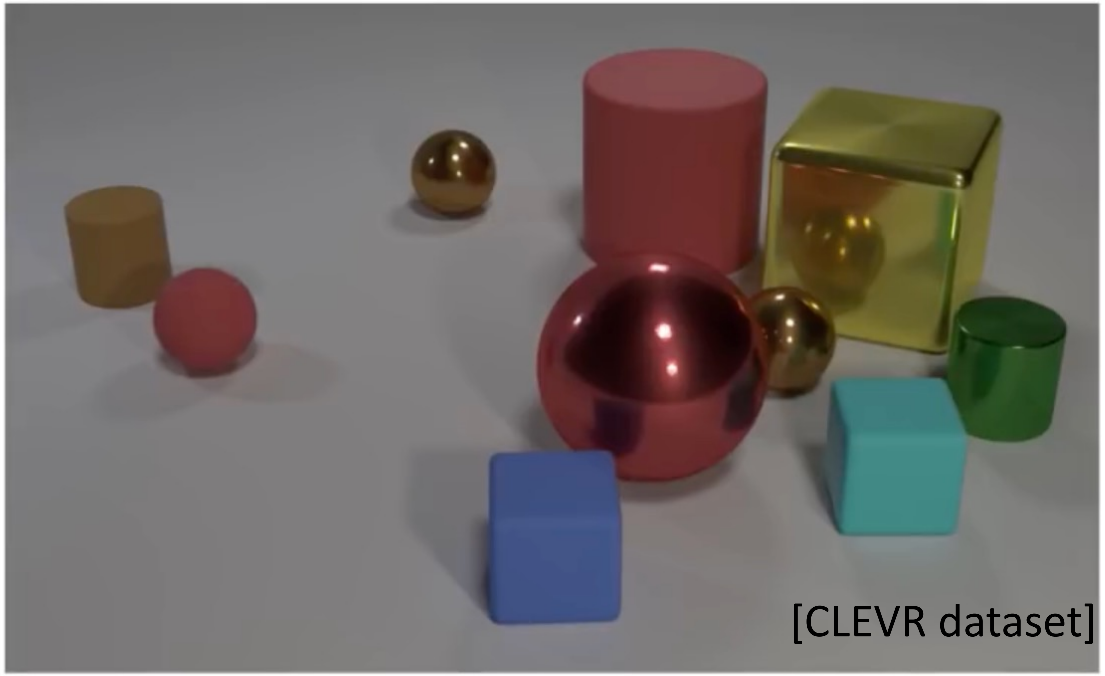
NN accuracy: 8.7%

What is Neuro-Symbolic AI?



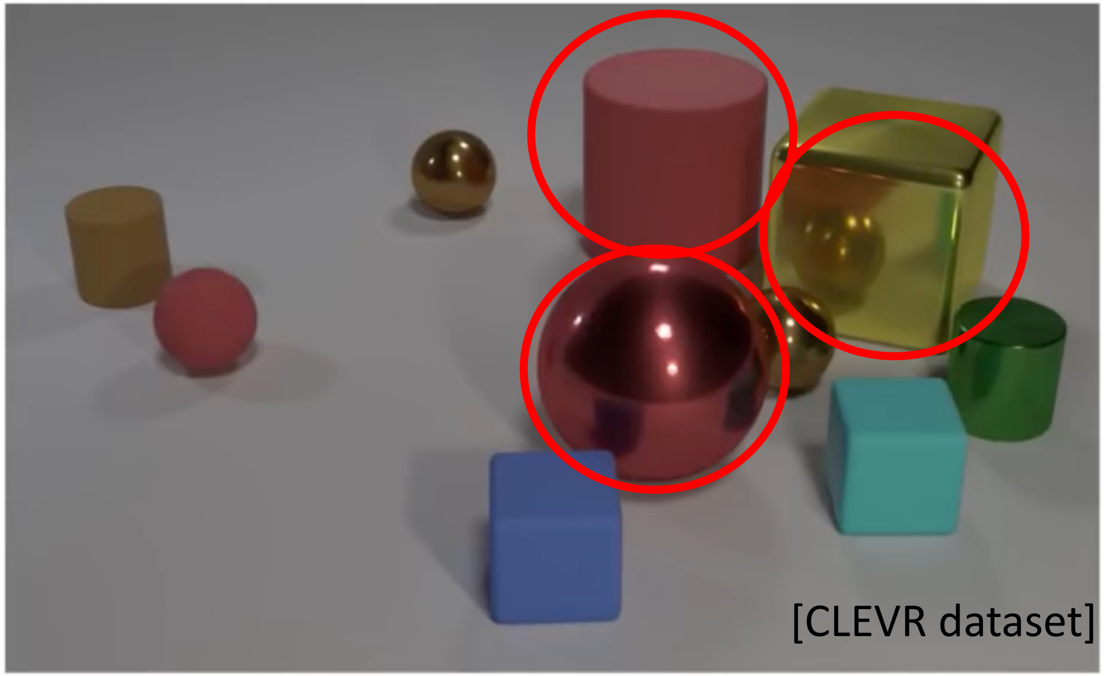
Towards Cognitive and Trustworthy AI Systems

Neuro-Symbolic AI Example: Visual Reasoning



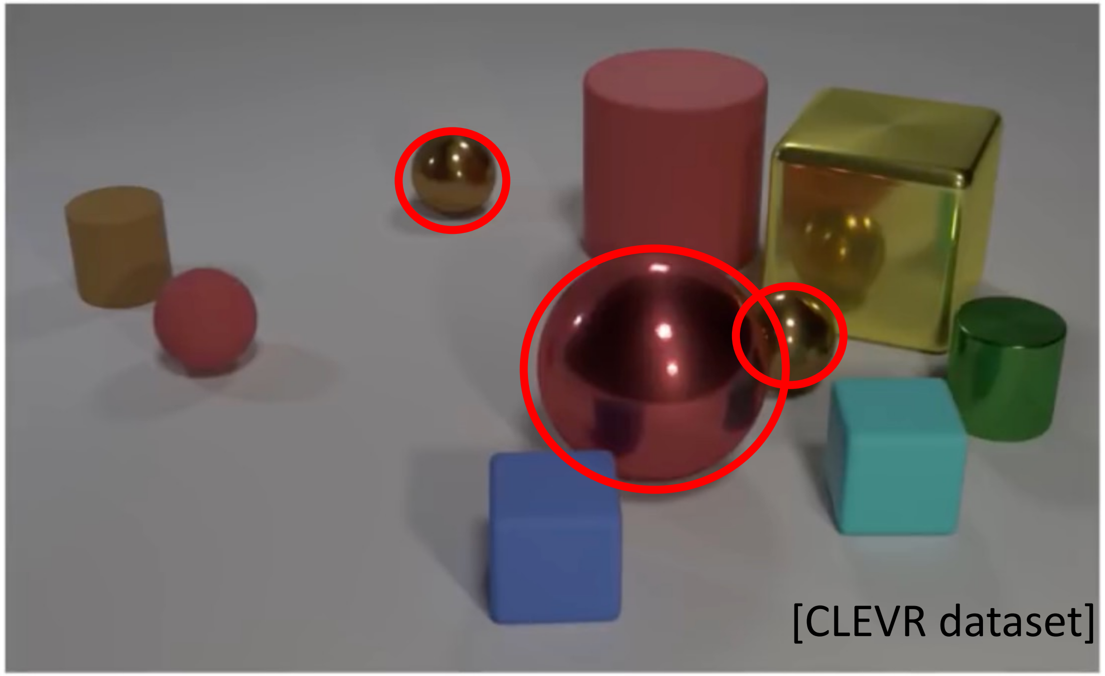
Question: *Are there an* ***equal number*** *of large things and metal spheres?*

Neuro-Symbolic AI Example: Visual Reasoning

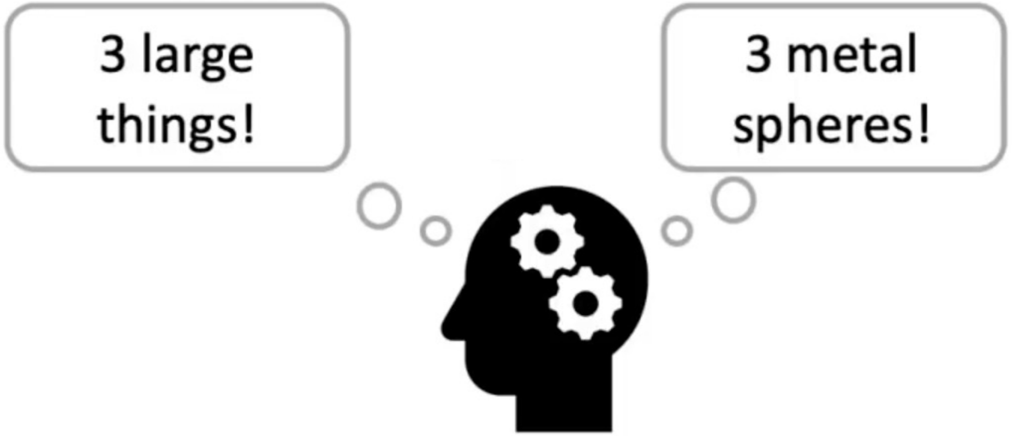


Question: *Are there an **equal number** of large things and metal spheres?*

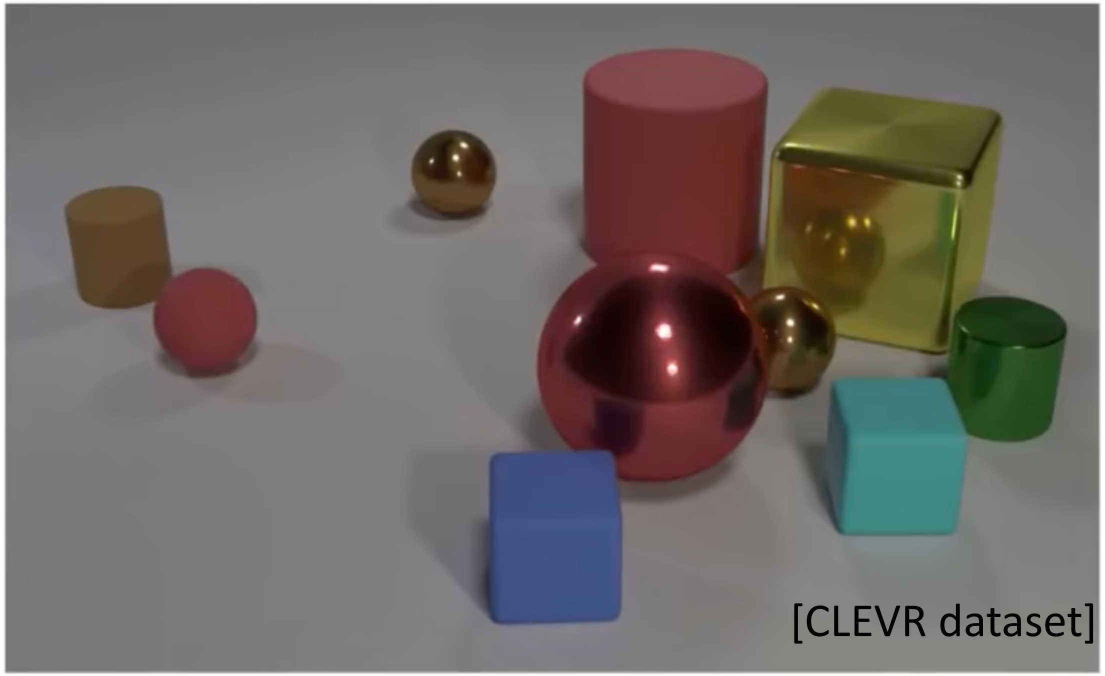
Neuro-Symbolic AI Example: Visual Reasoning

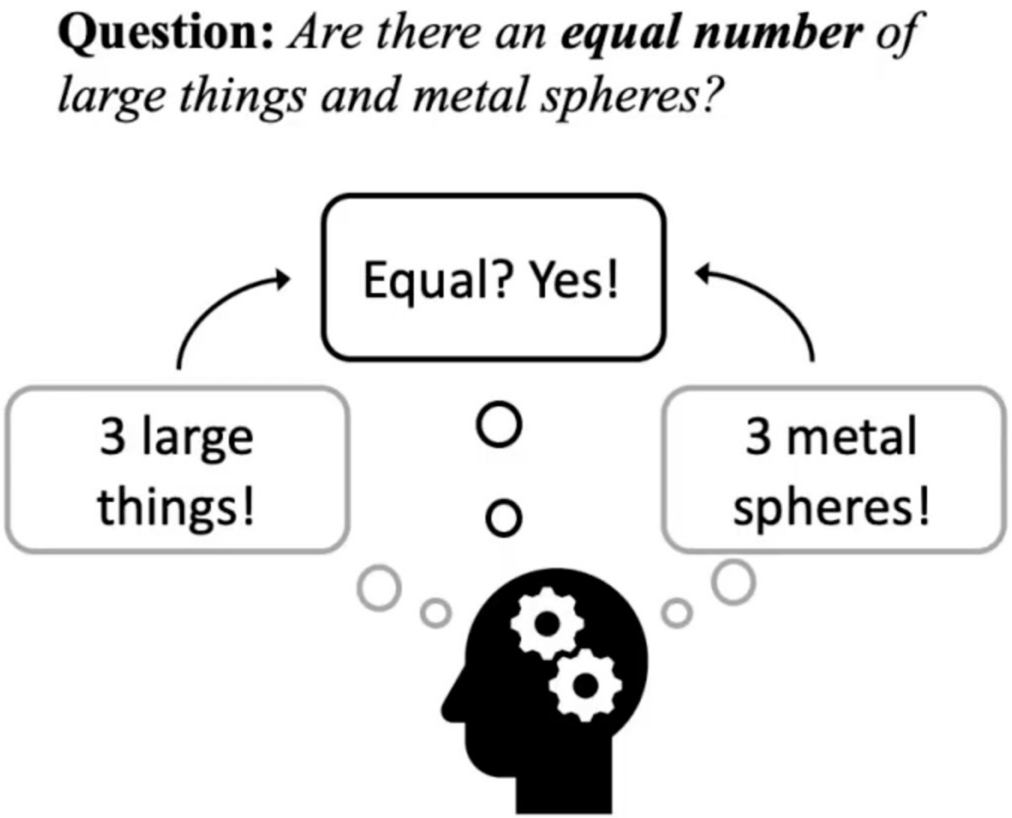


Question: *Are there an **equal number** of large things and metal spheres?*

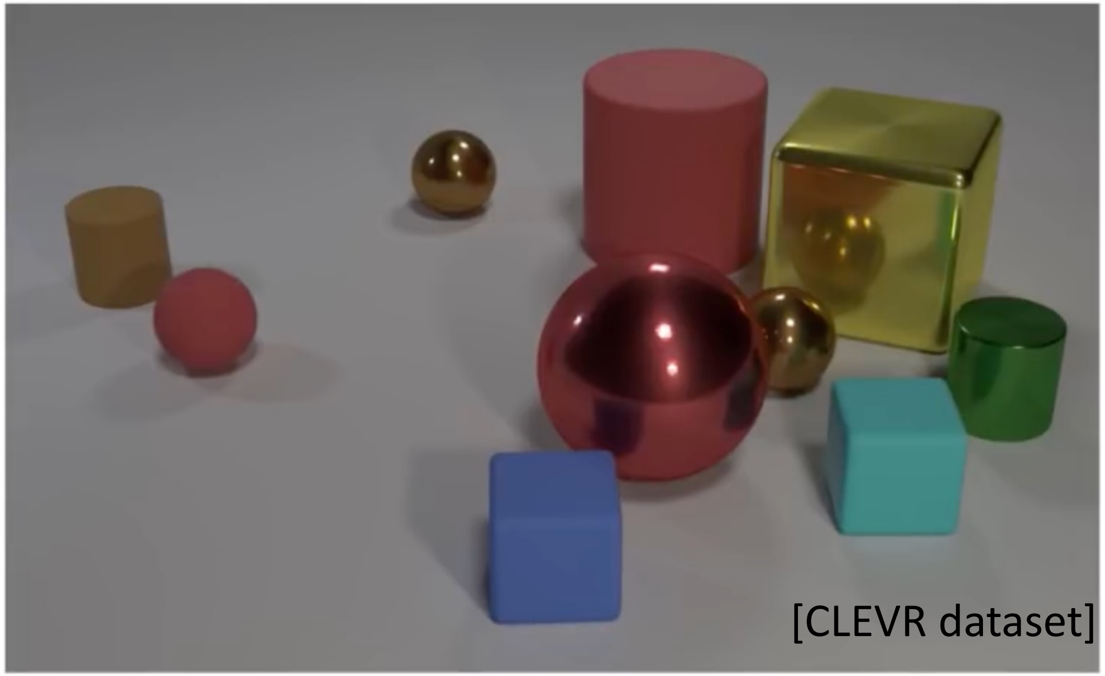


Neuro-Symbolic AI Example: Visual Reasoning





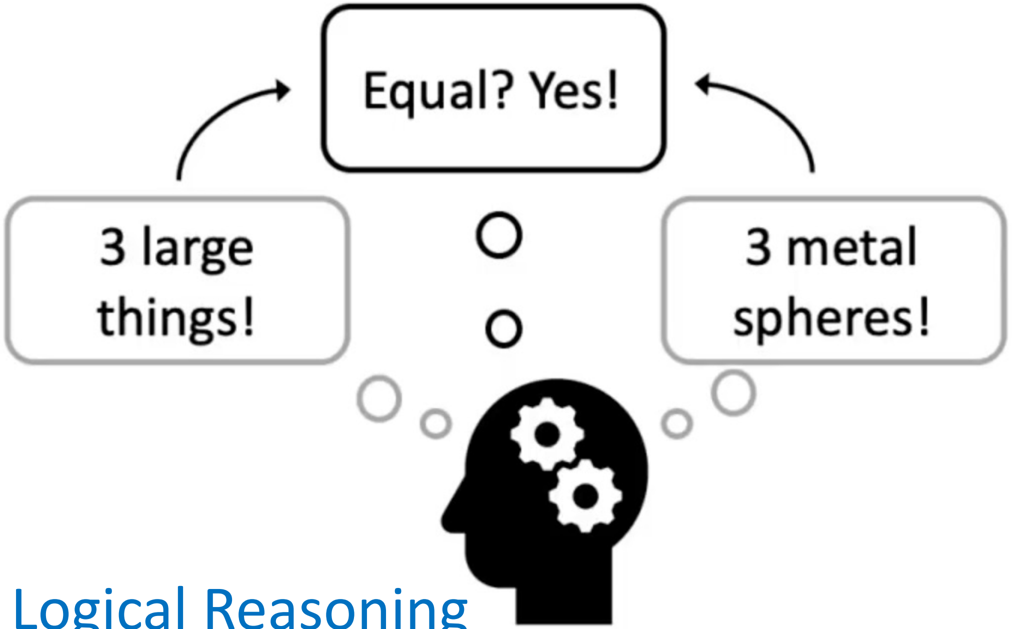
Neuro-Symbolic AI Example: Visual Reasoning



Visual Perception

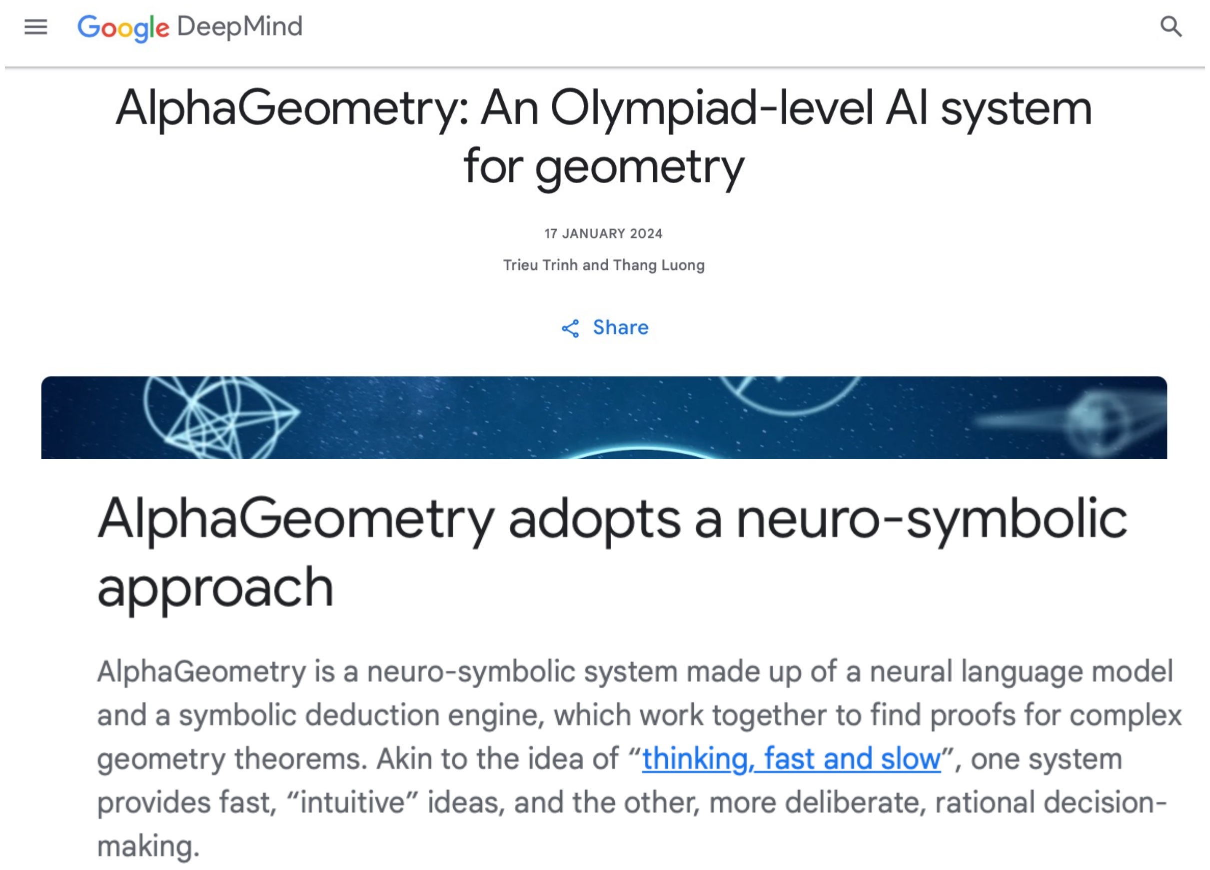
Question Understanding

Question: *Are there an* ***equal number*** *of large things and metal spheres?*



Logical Reasoning

Other Examples


Google DeepMind

AlphaGeometry: An Olympiad-level AI system for geometry

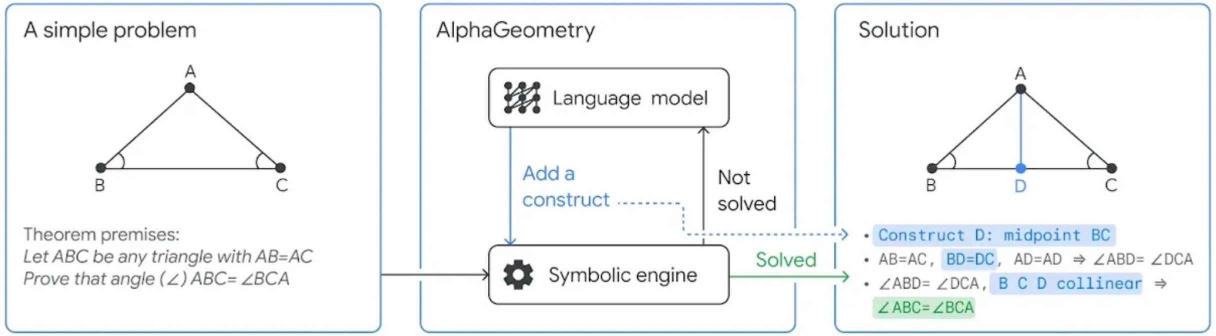
17 JANUARY 2024

Trieu Trinh and Thang Luong

Share

AlphaGeometry adopts a neuro-symbolic approach

AlphaGeometry is a neuro-symbolic system made up of a neural language model and a symbolic deduction engine, which work together to find proofs for complex geometry theorems. Akin to the idea of "thinking, fast and slow", one system provides fast, "intuitive" ideas, and the other, more deliberate, rational decision-making.

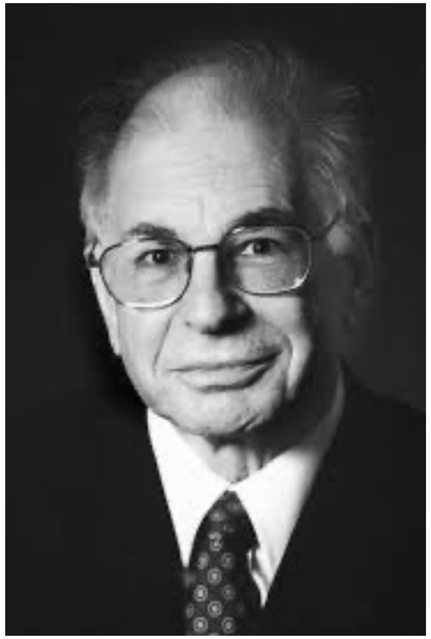


LLM: construct generation
Symbolic: deductive reasoning

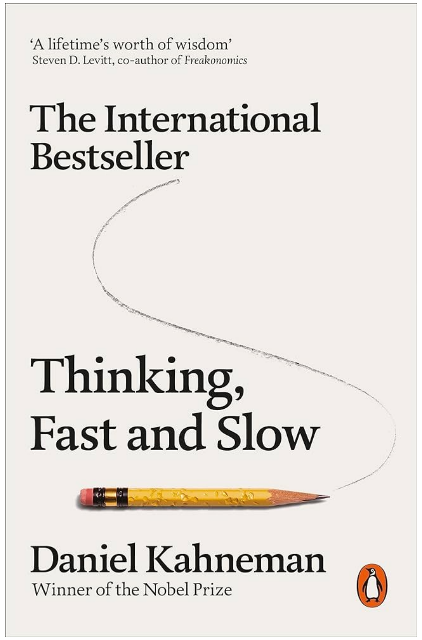
Eval on 30 Int. Math Olympics (IMO) problems:
- GPT-4: 0/30
- AlphaGeometry (Neuro-Symbolic): 25/30
- Human Gold Medalist: 26/30

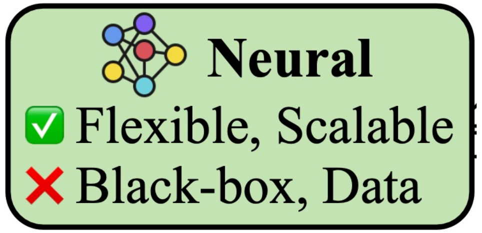
Trinh et al, "Solving Olympiad Geometry without Human Demonstrations", Nature 2024

Relationship to Human Minds



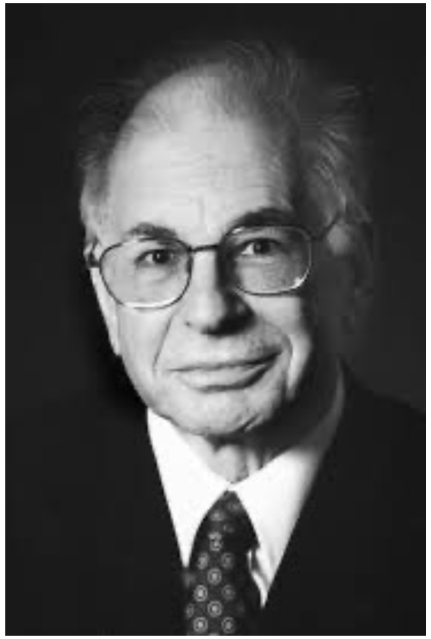
Daniel Kahneman
(1934-2024)



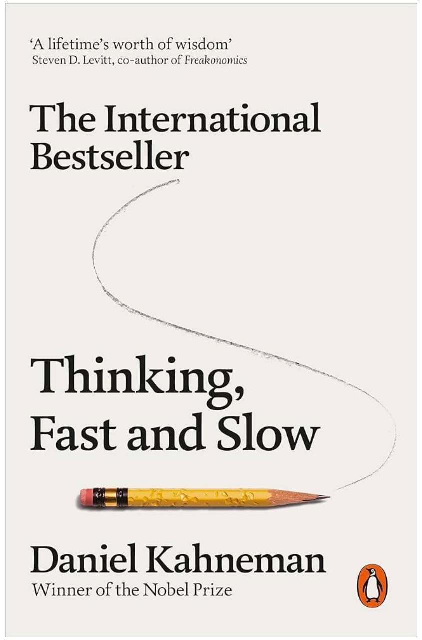


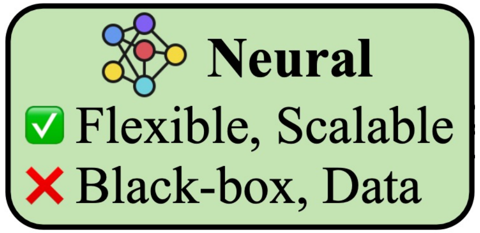
*System 1: thinking **fast***
(intuitive perception)

Relationship to Human Minds

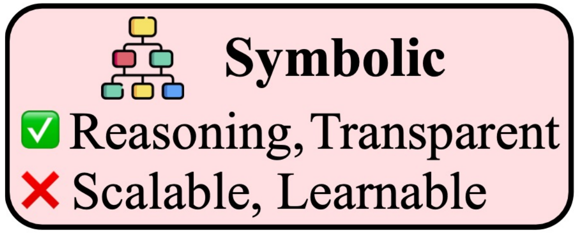


Daniel Kahneman
(1934-2024)



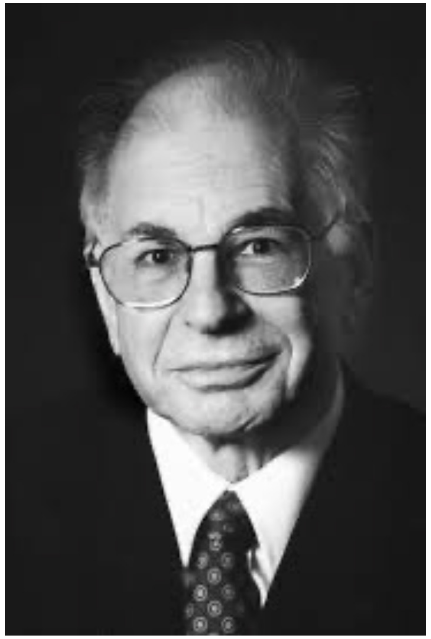


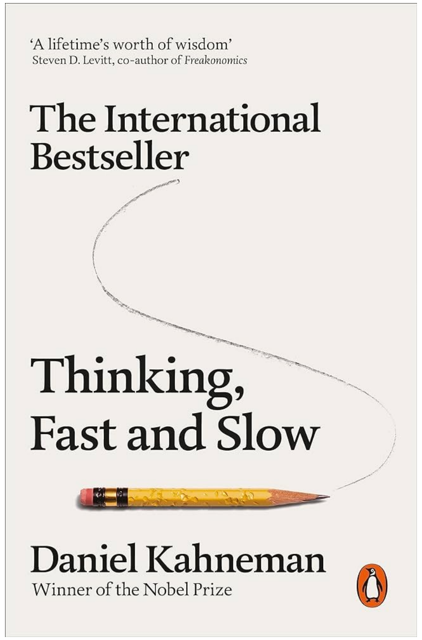
System 1: thinking ***fast***
(intuitive perception)

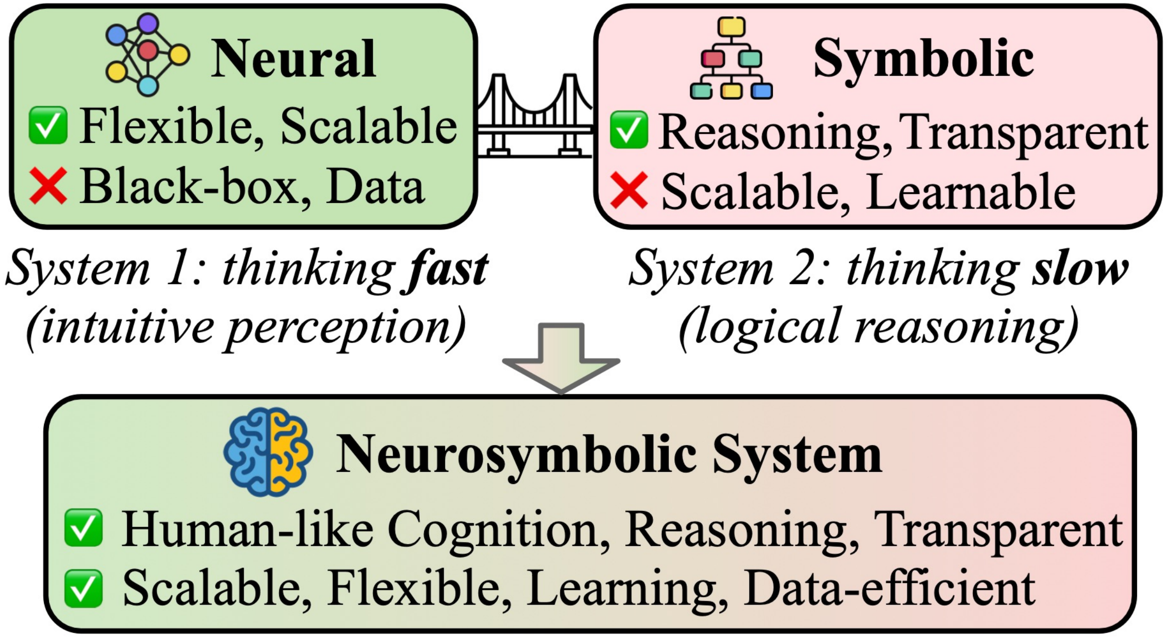


System 2: thinking ***slow***
(logical reasoning)

Relationship to Human Minds

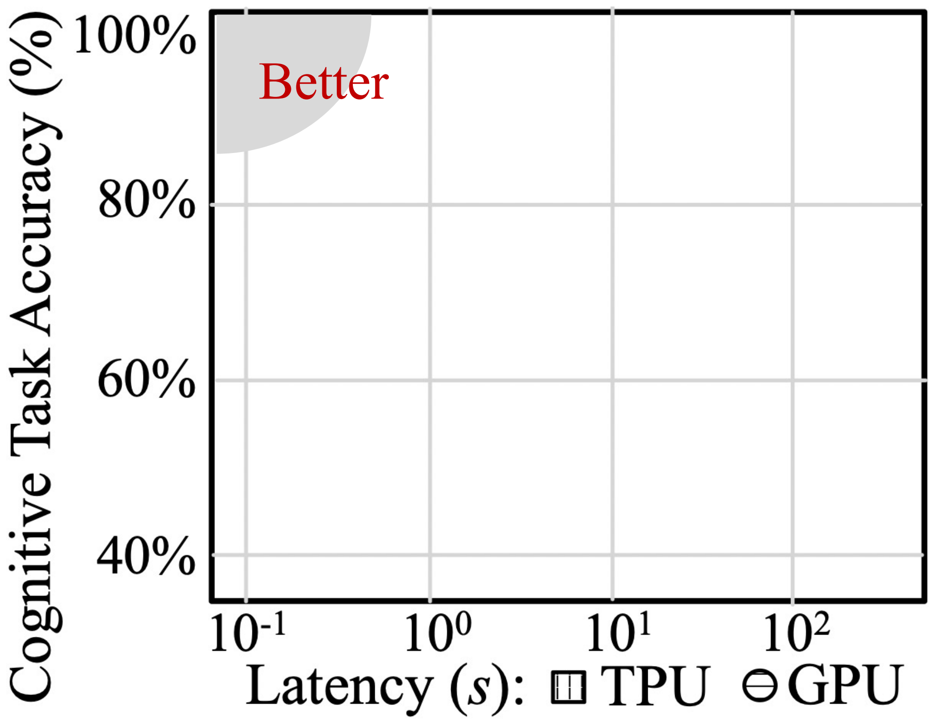


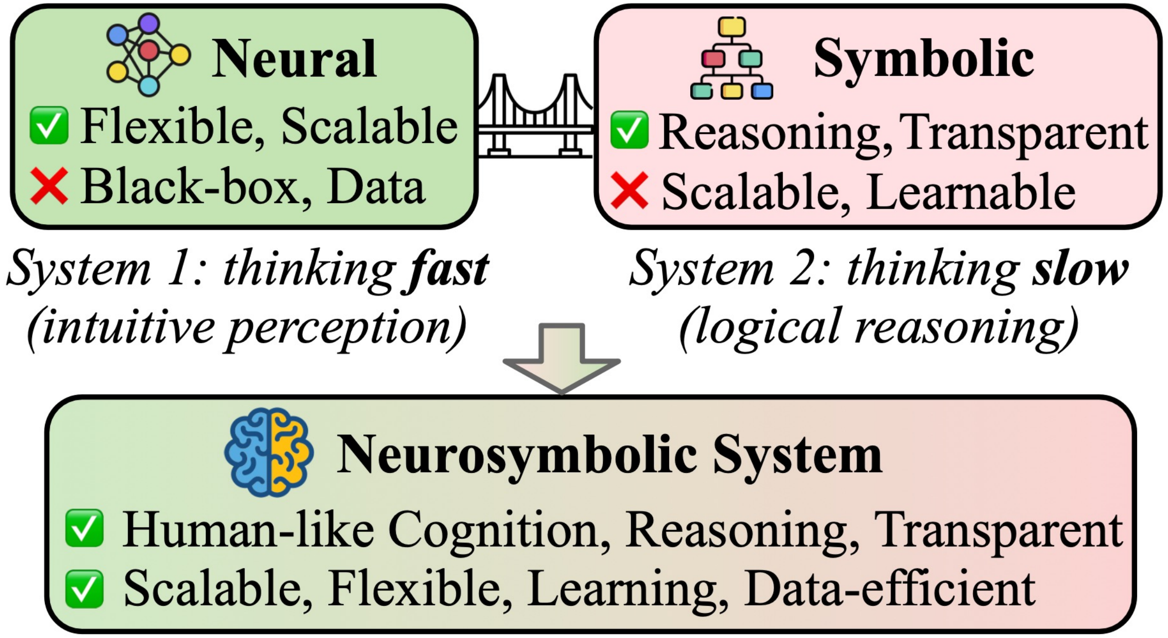




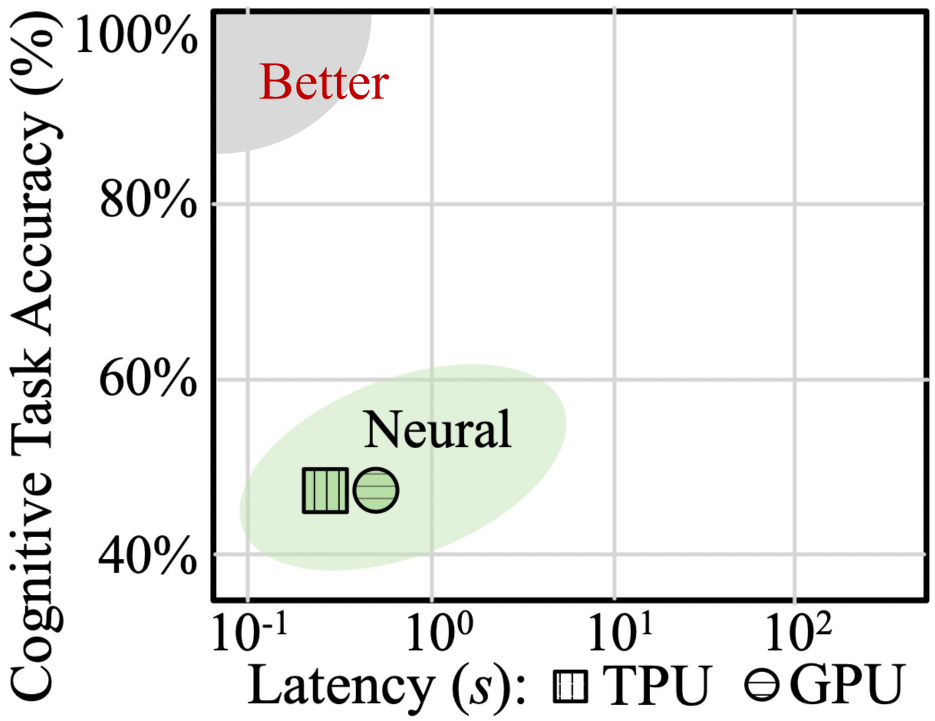
Daniel Kahneman
(1934-2024)

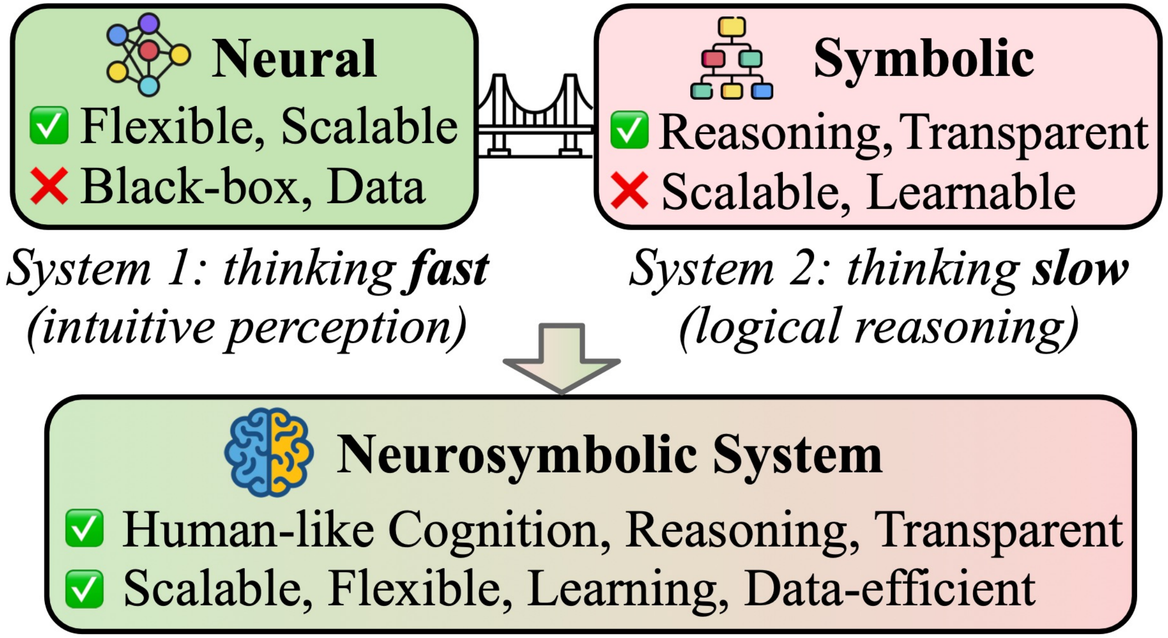
However.. From Computing Perspective



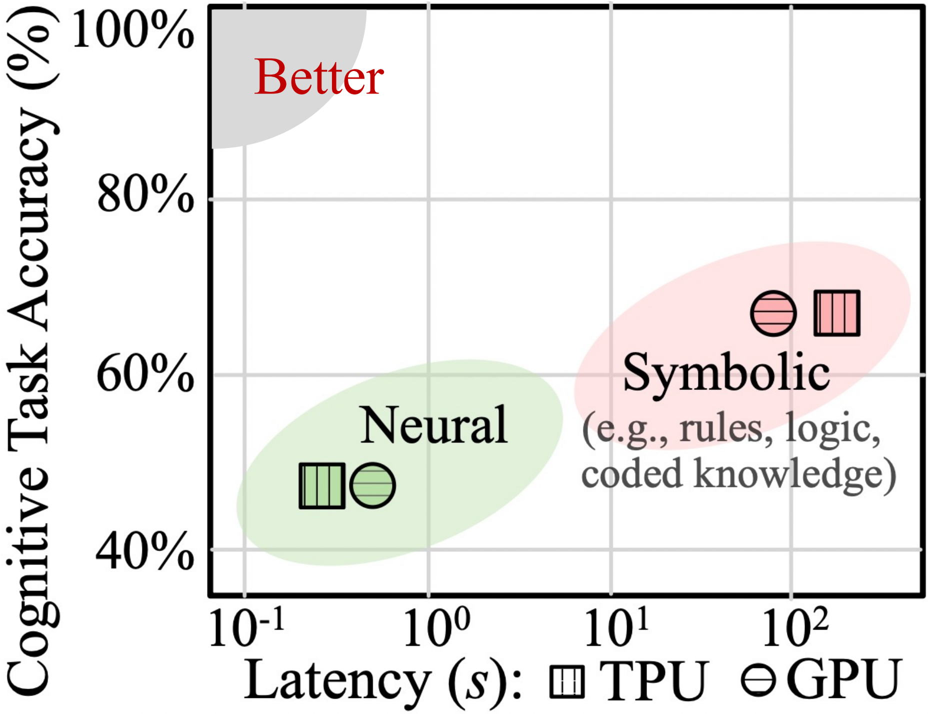


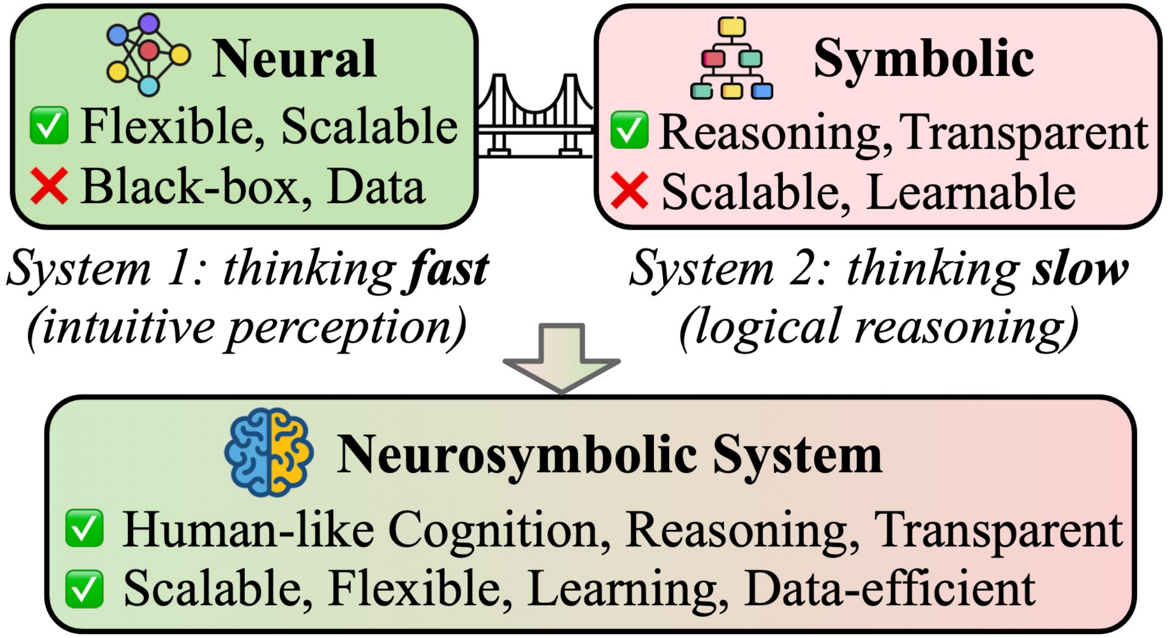
However.. From Computing Perspective



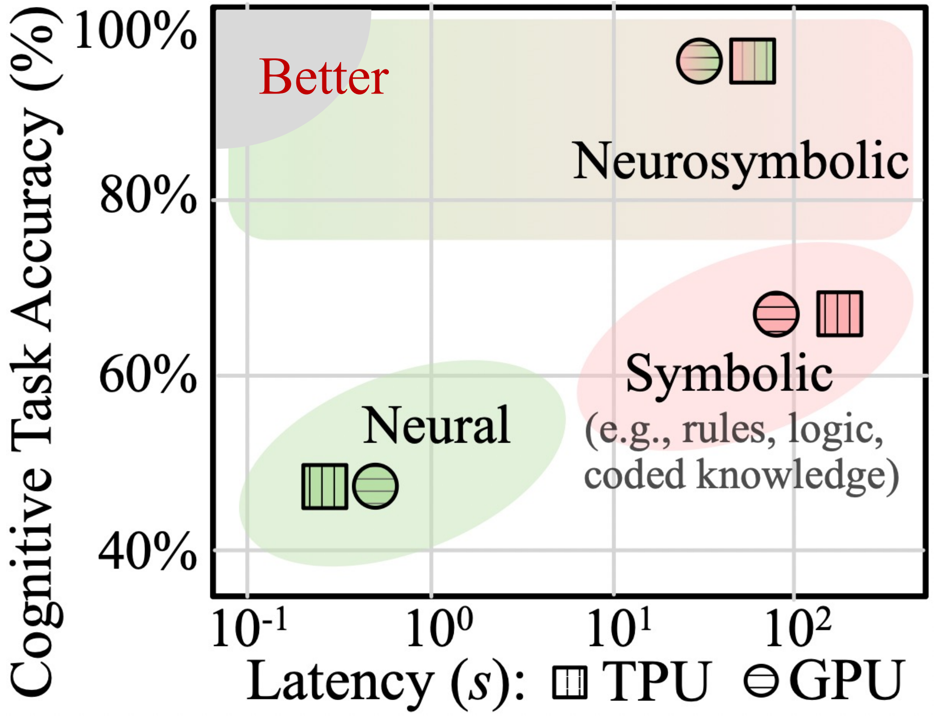


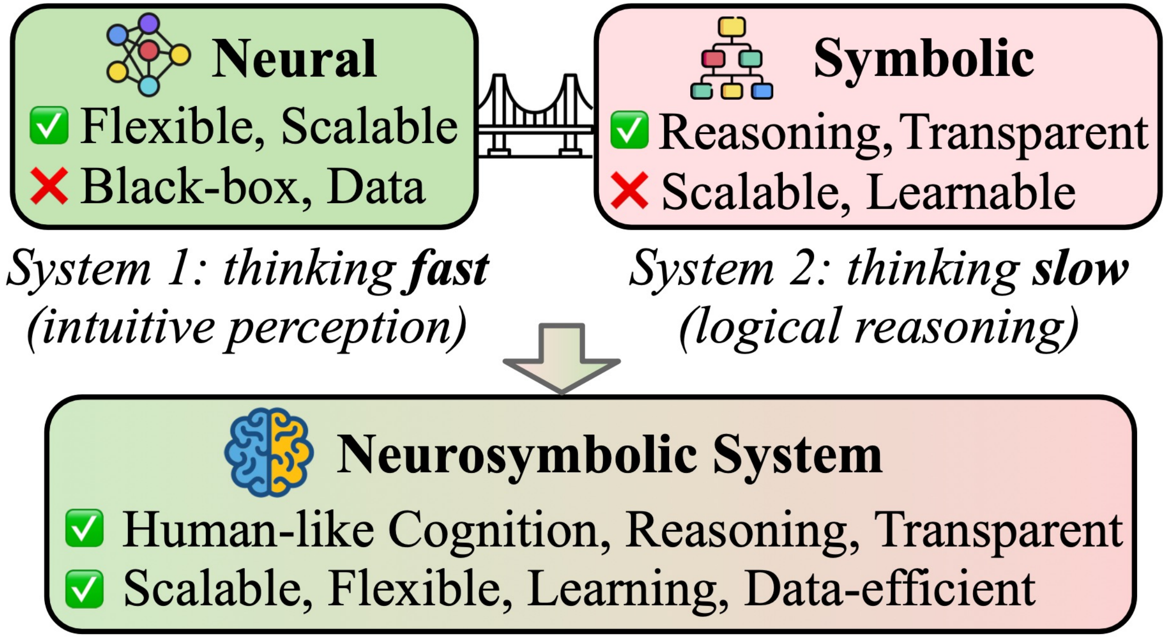
However.. From Computing Perspective



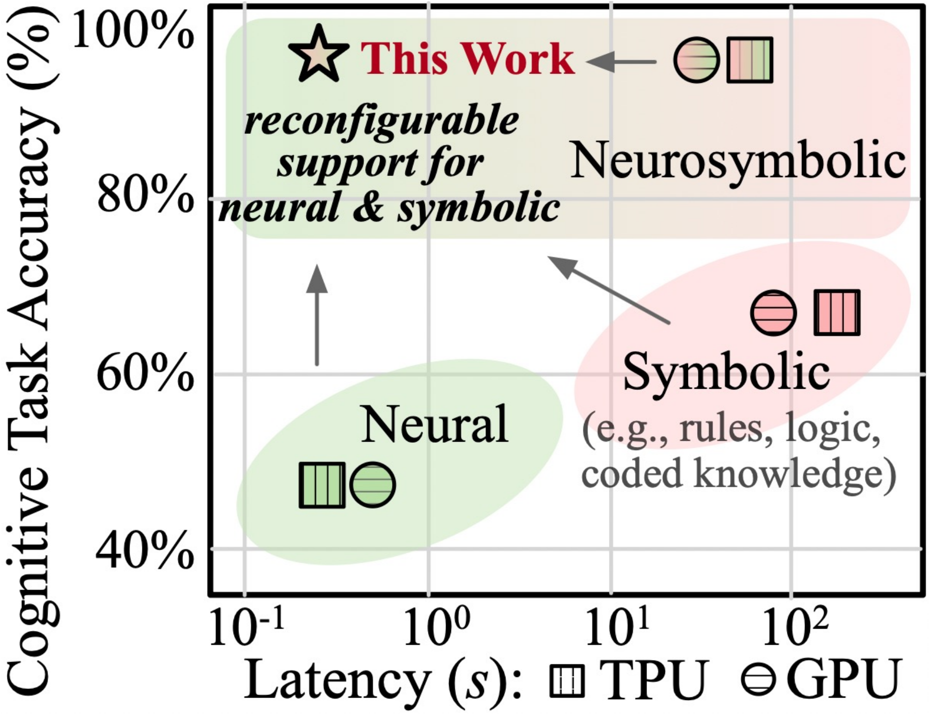


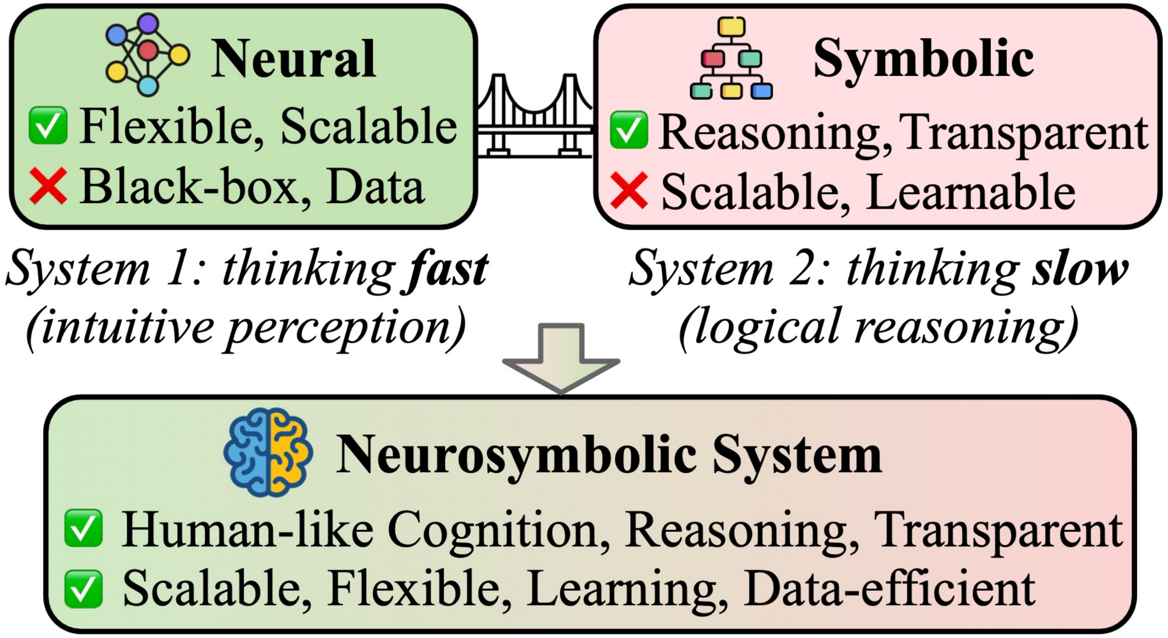
However.. From Computing Perspective





However.. From Computing Perspective



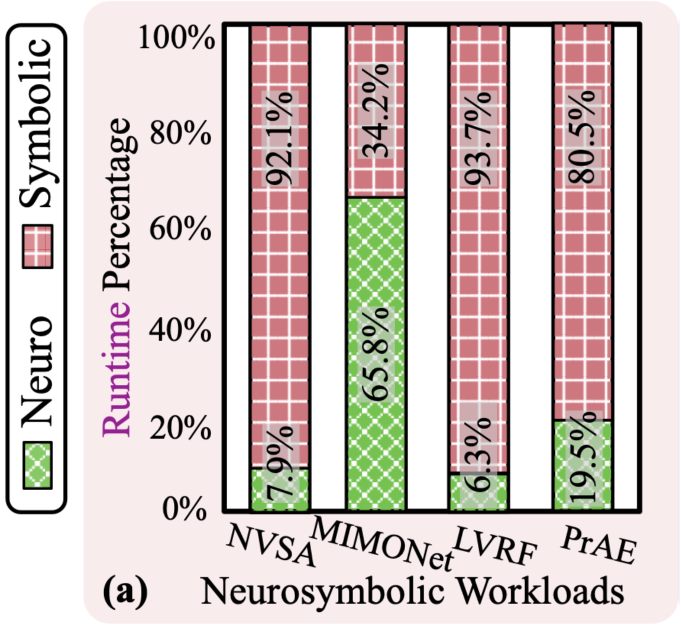


Research Question:

What's the **system implications** of neuro-symbolic workloads?

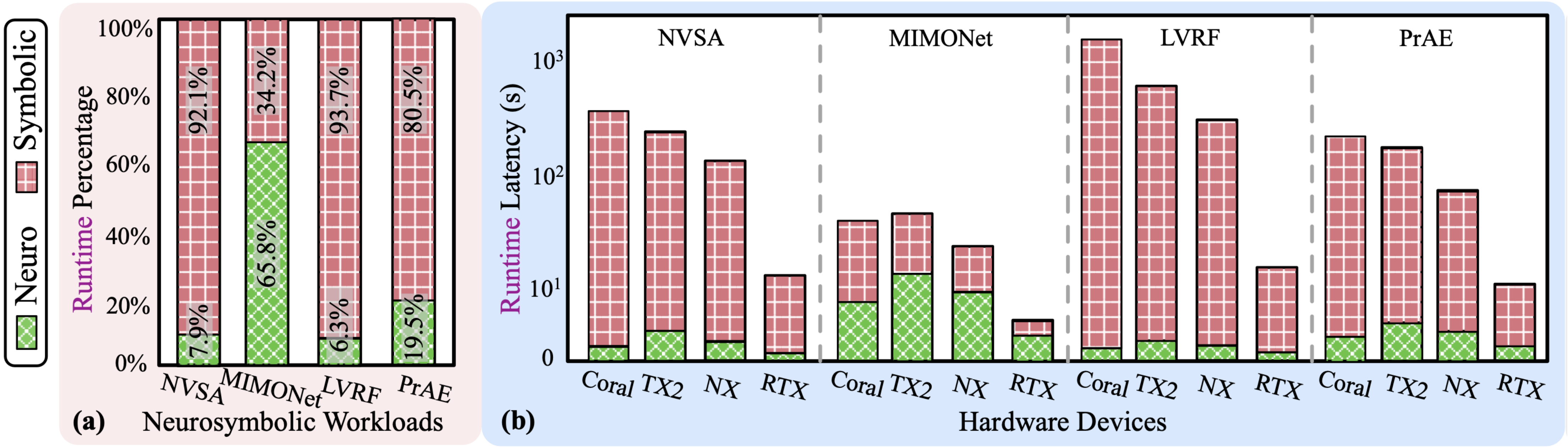
Why neuro-symbolic workloads are **inefficient** on off-the-shelf hardware?

Workload Profiling – Runtime



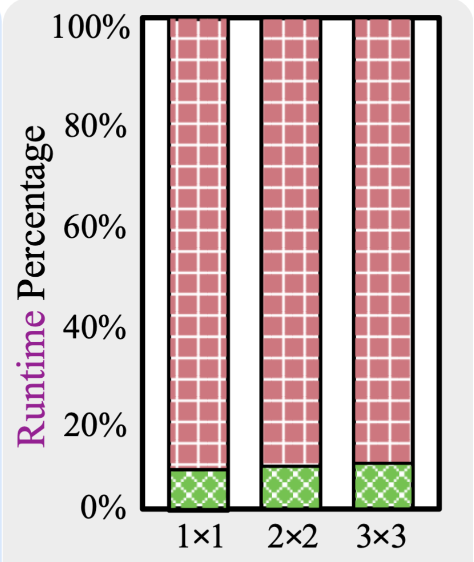
Neuro-symbolic workload exhibits **high latency** compared to neural models;

Workload Profiling – Runtime

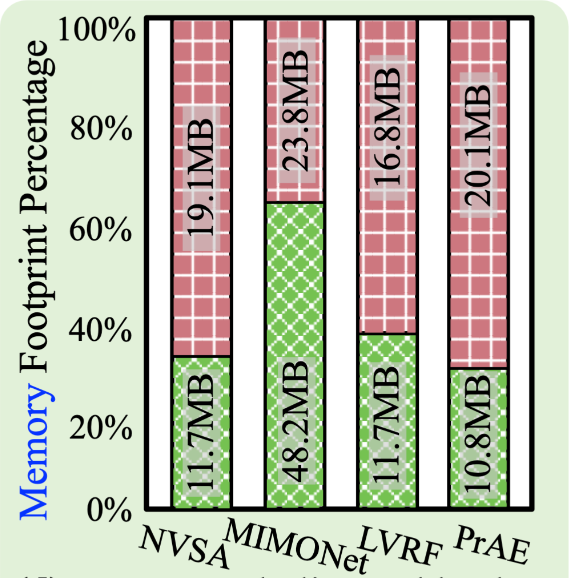


Neuro-symbolic workload exhibits **high latency** compared to neural models; Symbolic component is executed **inefficiently** across off-the-shelf CPU/GPUs

Workload Profiling – Memory & Operator



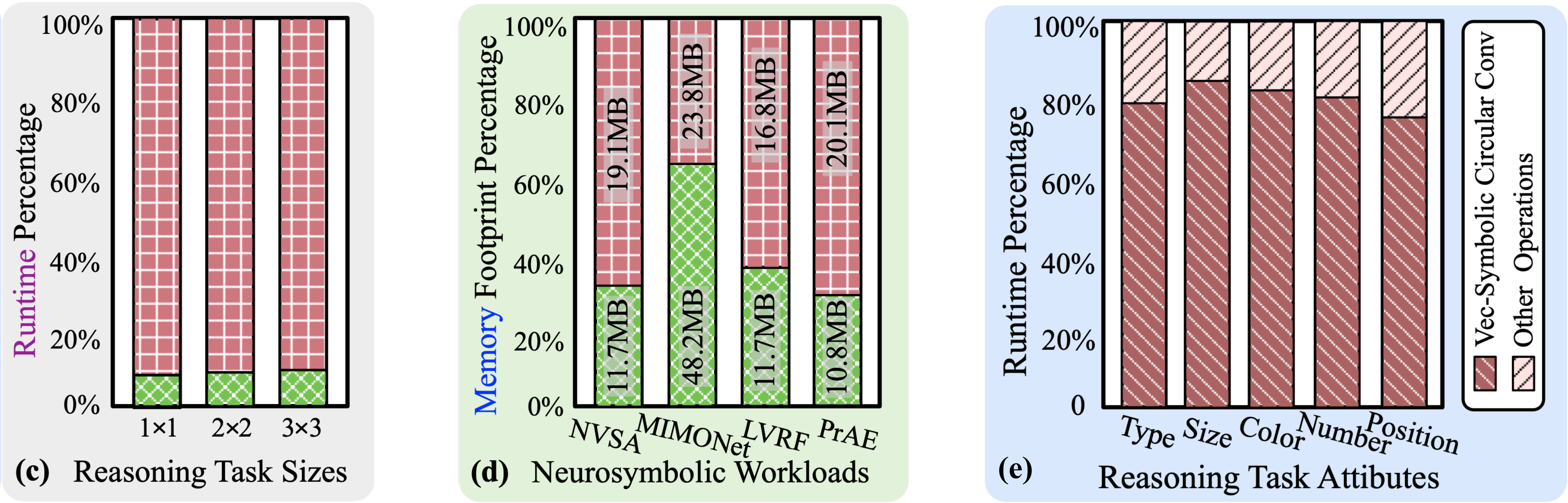
(c) Reasoning Task Sizes



(d) Neurosymbolic Workloads

Symbolic components exhibit **large memory footprint**;

Workload Profiling – Memory & Operator



Symbolic components exhibit **large memory footprint**;
Symbolic operations are dominated by **vector-symbolic circular convolutions**

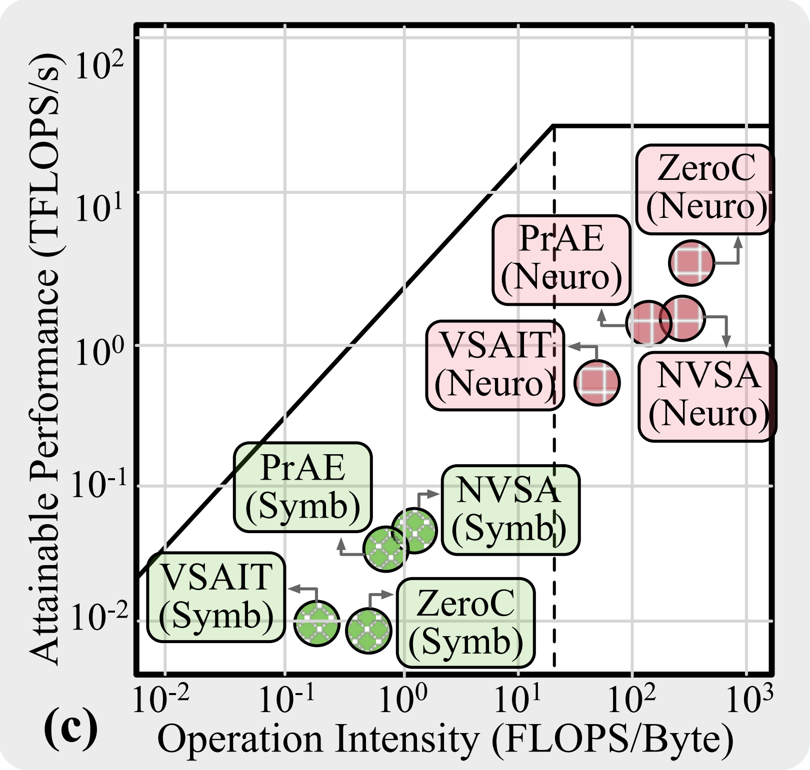
Workload Profiling – Kernel Behavior

	Neuro Kernel		Symbolic Kernel	
	segmm_nn	relu_nn	vectorized	elementwise
Runtime Percentage (%)				
Compute Throughput (%)				
ALU Utilization (%)				
L1 Cache Hit Rate (%)				
L2 Cache Hit Rate (%)				
L1 Cache Throughput (%)				
L2 Cache Throughput (%)				
DRAM BW Utilization (%)				

Why system Inefficiency?

Workload Profiling – Roofline Analysis

	Neuro Kernel		Symbolic Kernel	
	segmm_nn	relu_nn	vectorized	elementwise
Runtime Percentage (%)	18.2	10.4	37.5	12.4
Compute Throughput (%)	95.1	92.9	3.0	2.3
ALU Utilization (%)	90.1	48.3	5.9	4.5
L1 Cache Hit Rate (%)	1.6	51.6	29.5	33.3
L2 Cache Hit Rate (%)	86.8	65.5	48.6	34.3
L1 Cache Throughput (%)	79.7	82.6	28.4	10.8
L2 Cache Throughput (%)	19.2	17.5	29.8	22.8
DRAM BW Utilization (%)	14.9	24.2	90.9	78.4



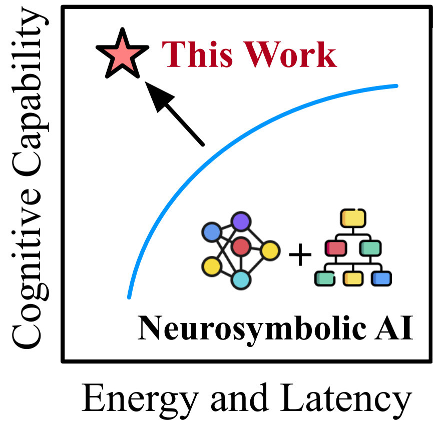
Symbolic exhibits low ALU utilization, low cache hit rate, massive data transfer, low data reuse, resulting in hardware underutilization and inefficiency

Research Question:

How to enhance the **efficiency and scalability** of neuro-symbolic systems?

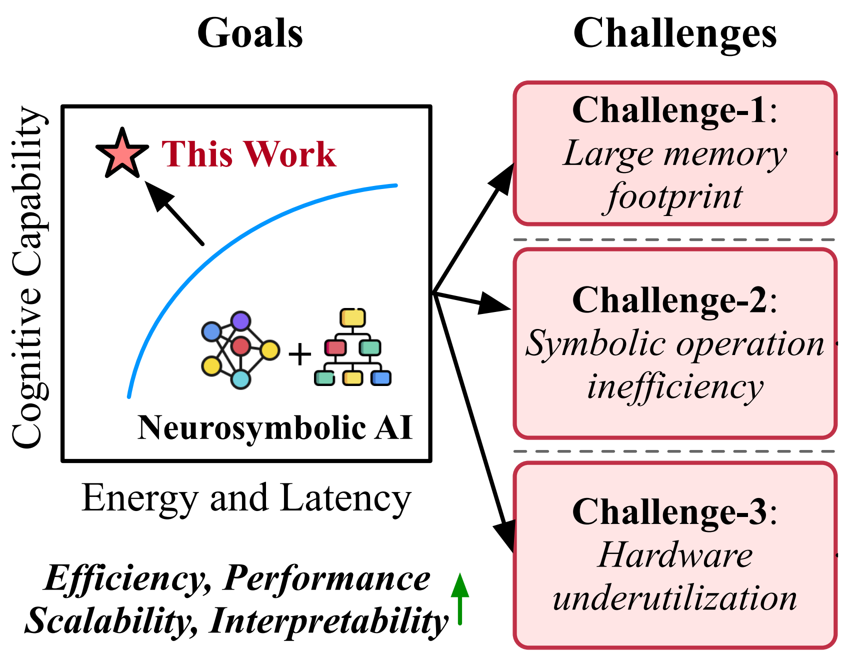
Our Methodology

Goals

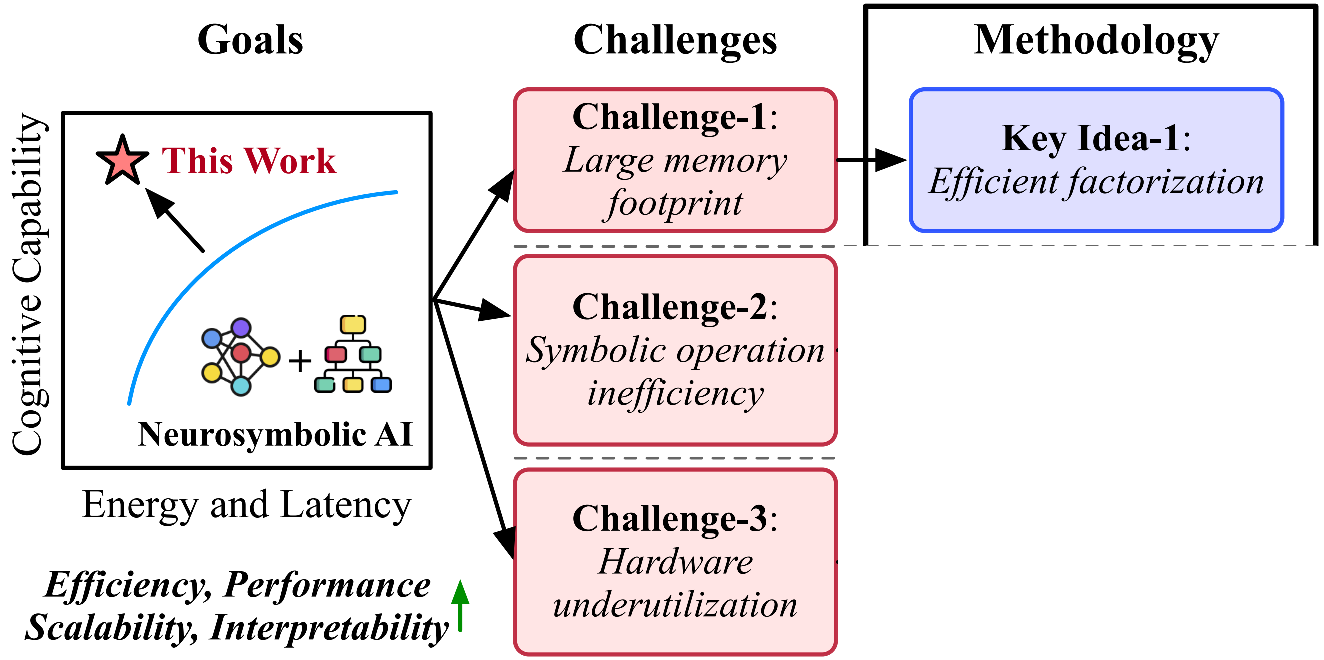


Efficiency, Performance
Scalability, Interpretability

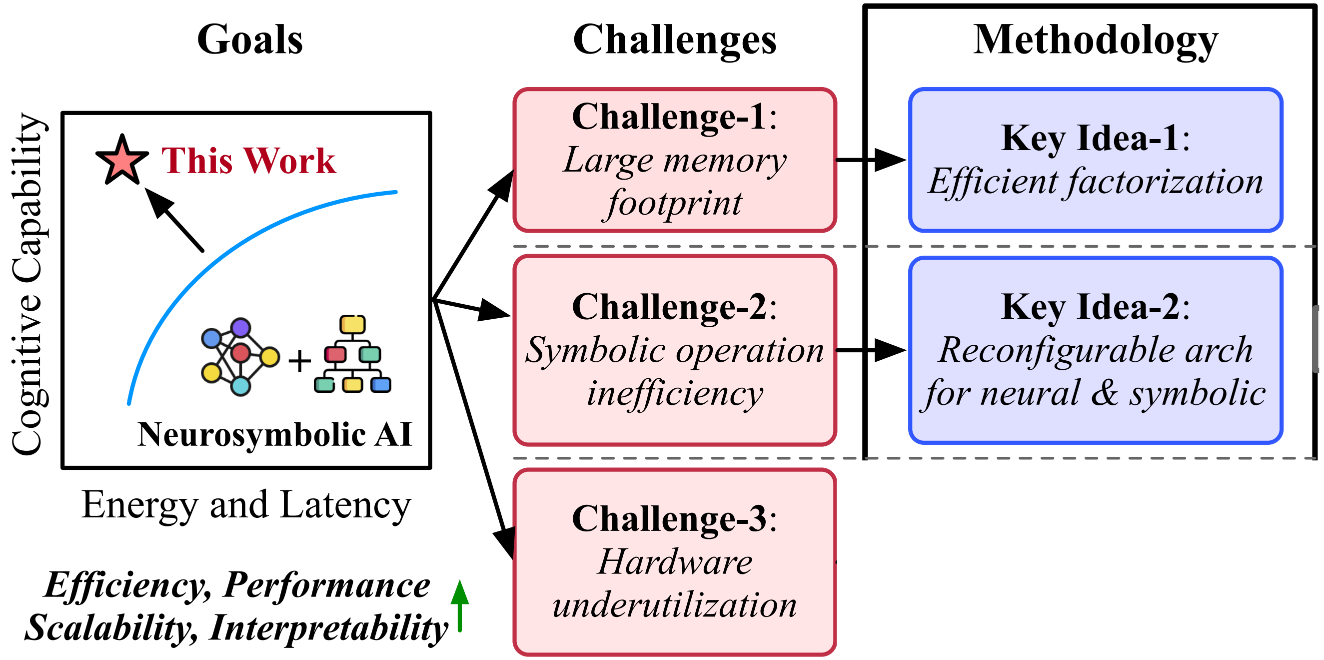
Our Methodology



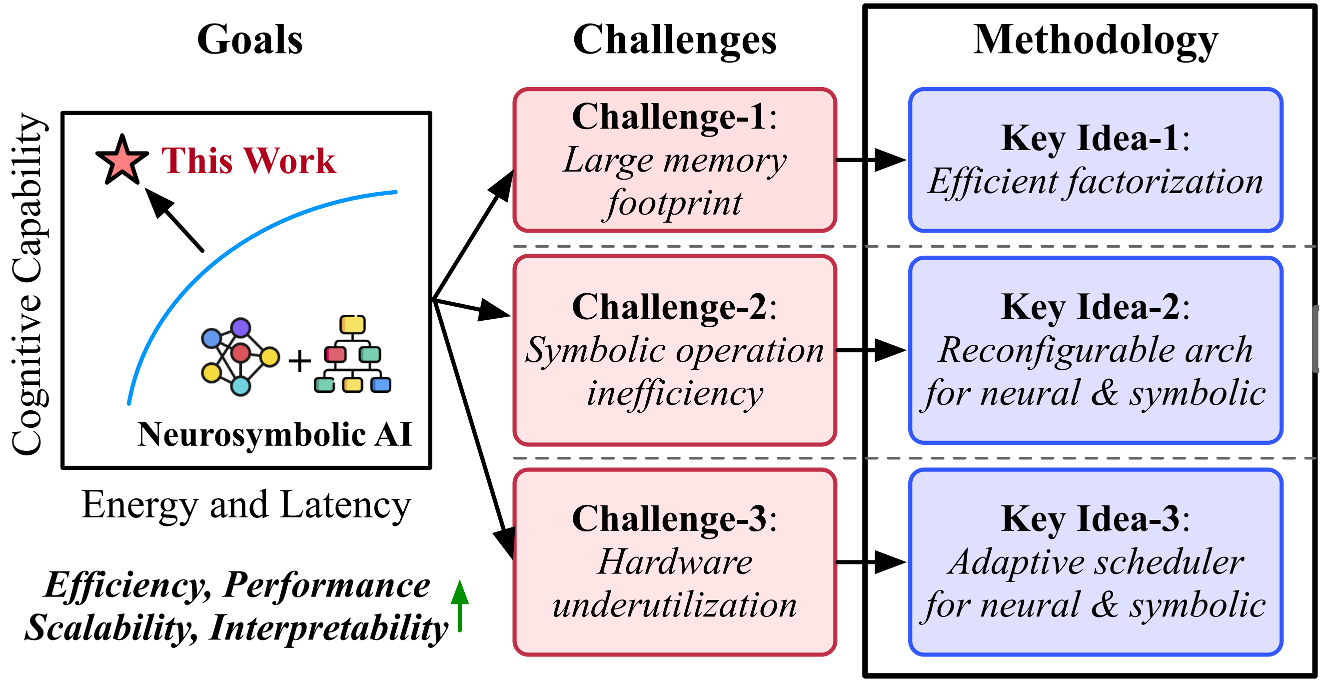
Our Methodology



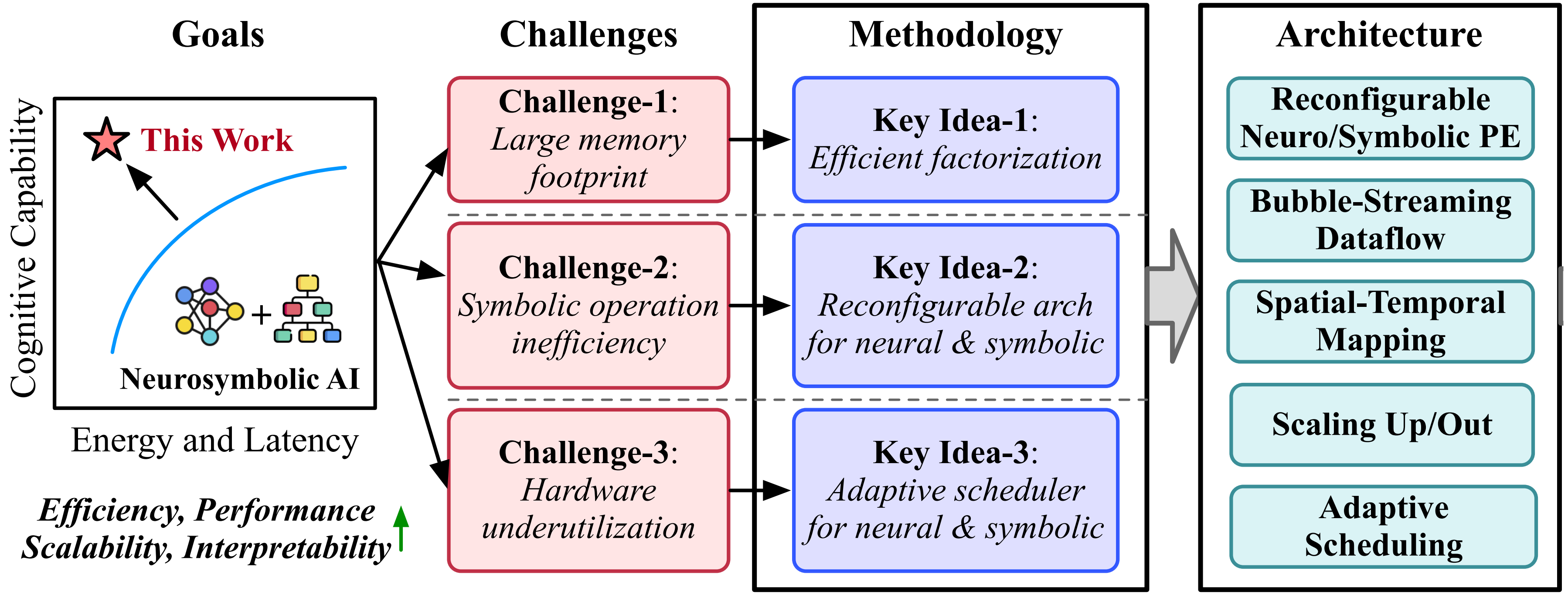
Our Methodology



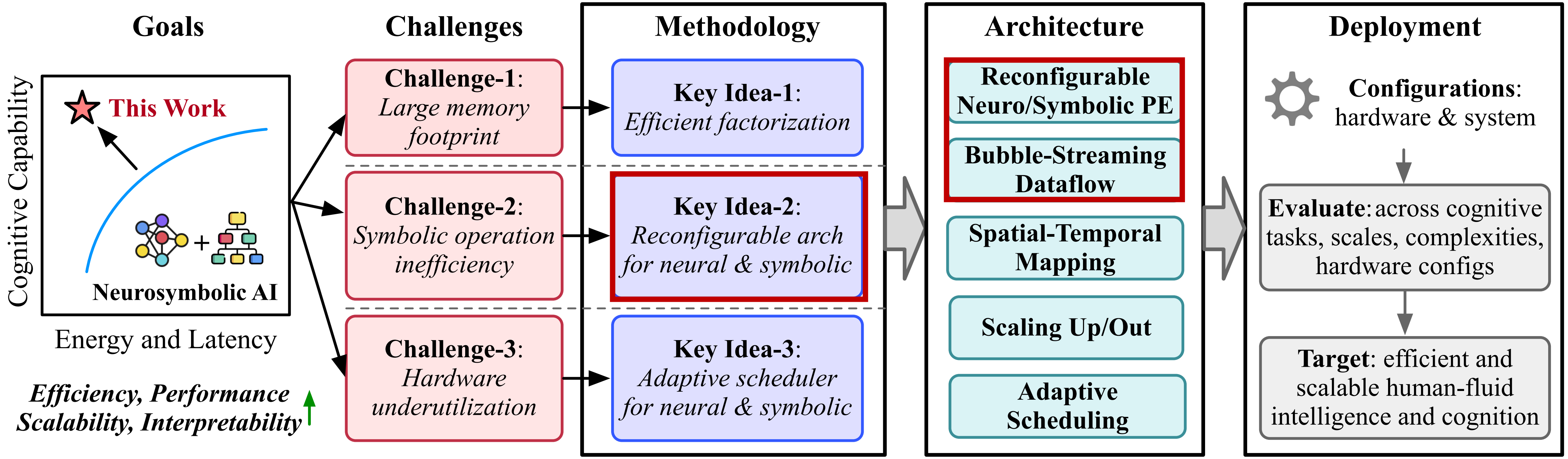
Our Methodology



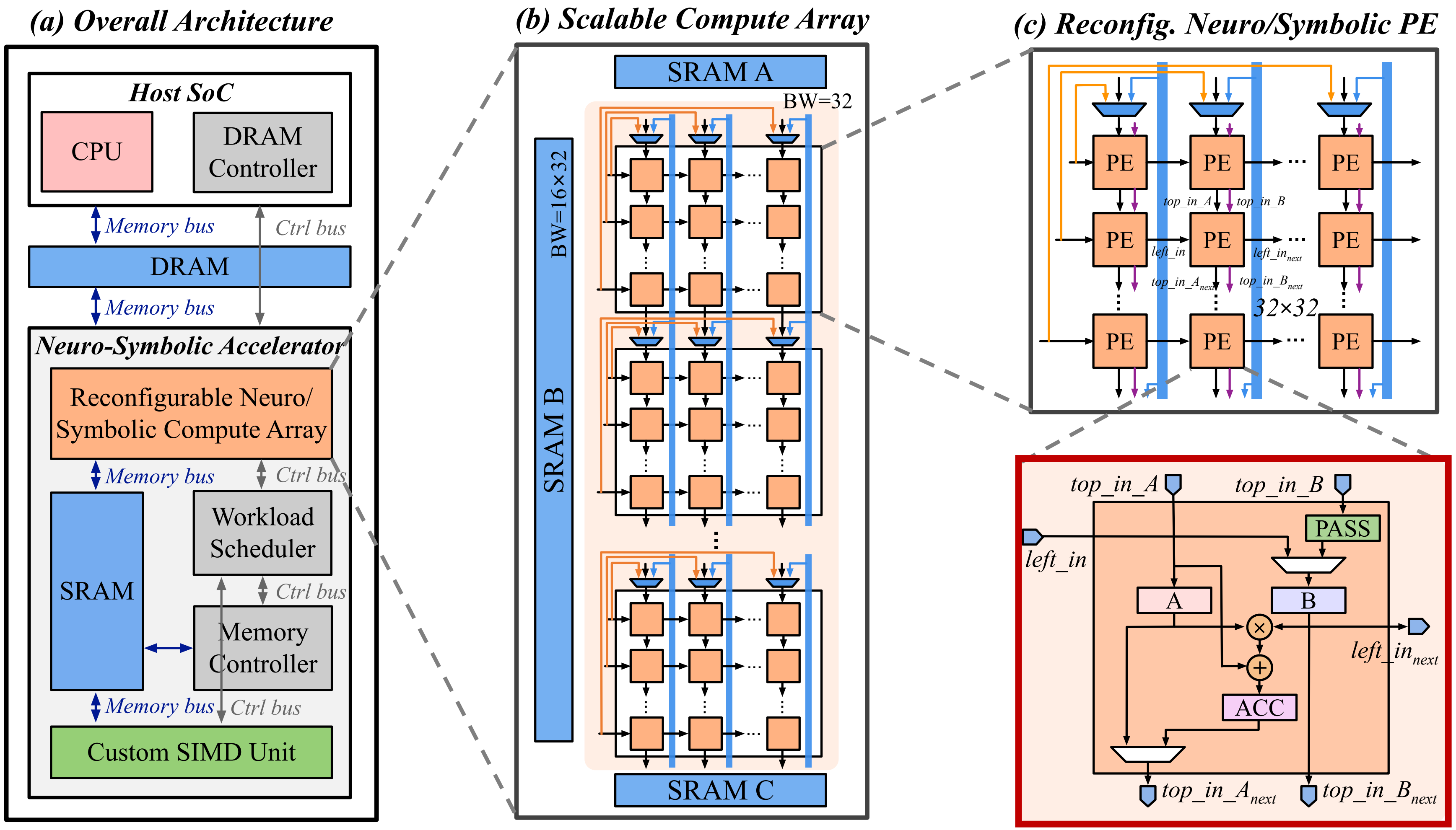
Our Methodology



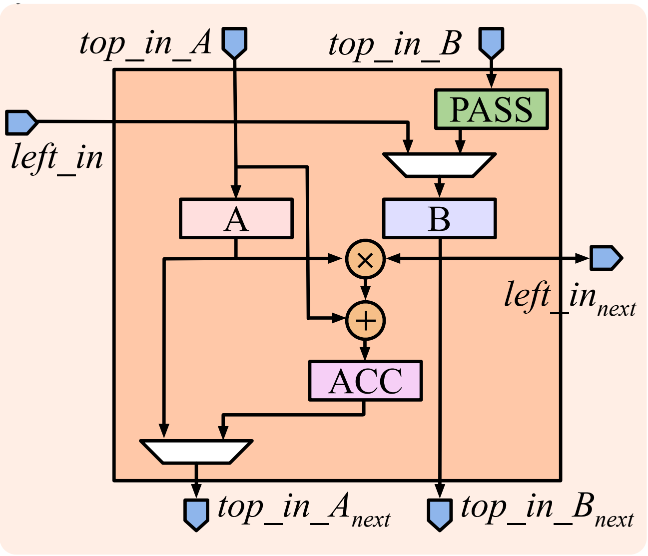
Our Methodology



Hardware Architecture Overview



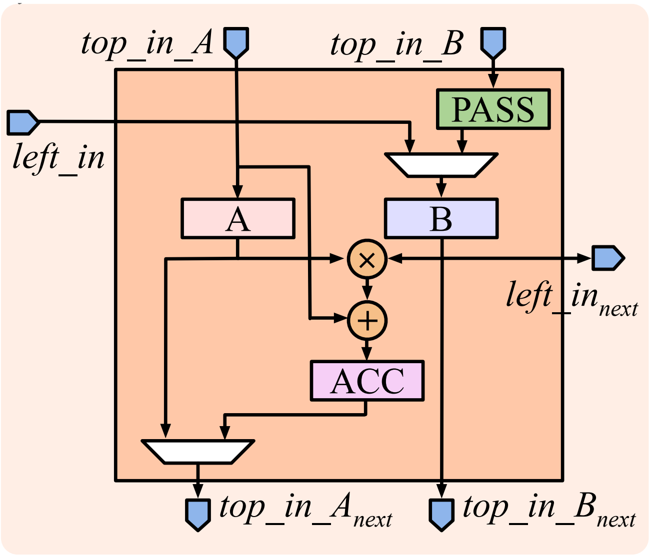
Reconfigurable Neuro/Symbolic PE



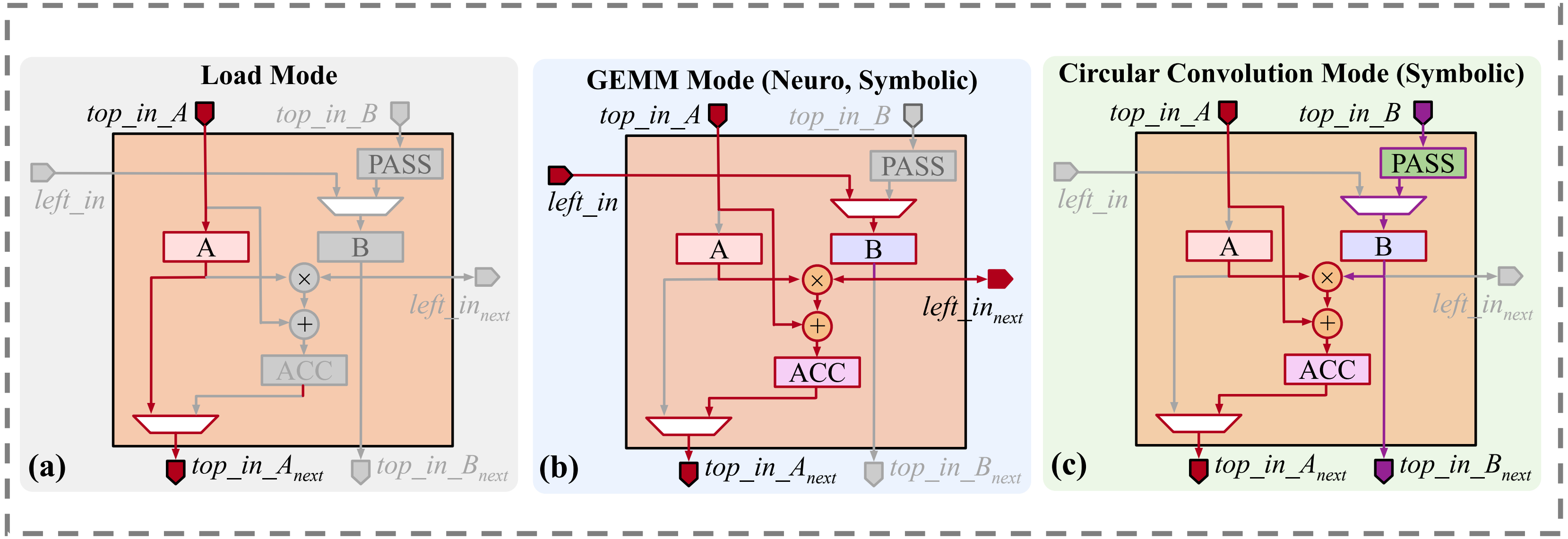
Micro-architecture of reconfigurable neuro/symbolic PE

Reconfigurable neuro/symbolic PE incurs **low area overhead** compared to systolic array PE;

Reconfigurable Neuro/Symbolic PE



Micro-architecture of reconfigurable neuro/symbolic PE

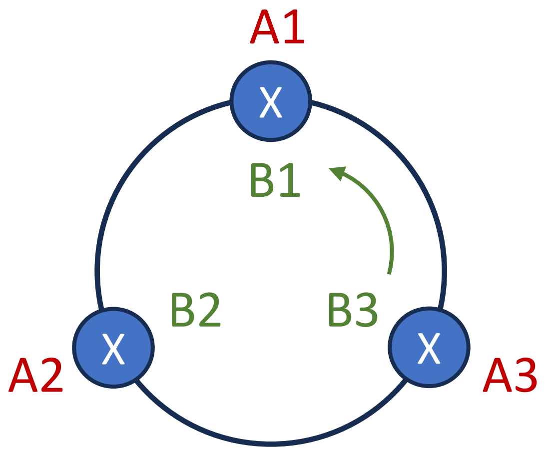


Operation mode of reconfigurable neuro/symbolic PE

Reconfigurable neuro/symbolic PE incurs **low area overhead** compared to systolic array PE; The PE is reconfigurable for **three operation modes**: load, neuro, symbolic

What is Circular Convolution?

$$\begin{bmatrix} A1 \\ A2 \\ A3 \end{bmatrix} \odot \begin{bmatrix} B1 \\ B2 \\ B3 \end{bmatrix} = \begin{pmatrix} A1B1+A2B2+A3B3 \\ A1B3+A2B1+A3B2 \\ A1B2+A2B3+A2B1 \end{pmatrix}$$

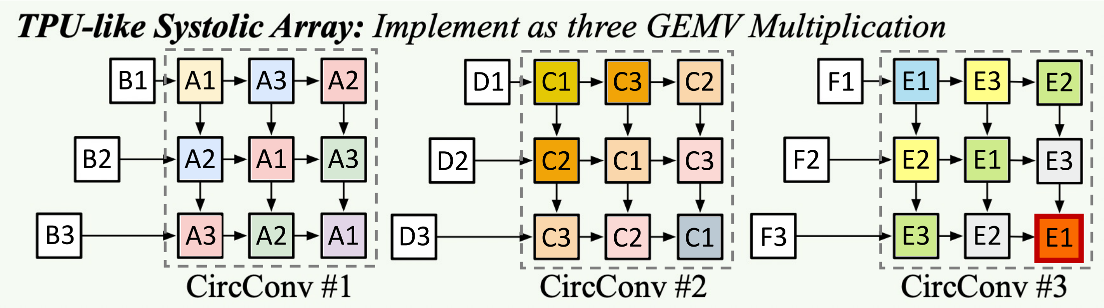


A1B1+A2B2+A3B3 A1B3+A2B1+A3B2 A1B2+A2B3+A2B1

Bubble Streaming Dataflow

Vector-Symbolic Circular Convolution Example (3 CircConv):
CircConv #1: (A1, A2, A3) ⊙ (B1, B2, B3)
CircConv #2: (C1, C2, C3) ⊙ (D1, D2, D3)
CircConv #3: (E1, E2, E3) ⊙ (F1, F2, F3)

CircConv #1 Computation:
(A1, A2, A3) ⊙ (B1, B2, B3) =
(A1B1+A2B2+A3B3, A1B3+A2B1+A3B2, A1B2+A2B3+A2B1)



Cycles:

TPU: Finish at (3n+15) = 24 cycles

For symbolic operation:
- TPU-like array **suffers from** low parallelism & high memory access;

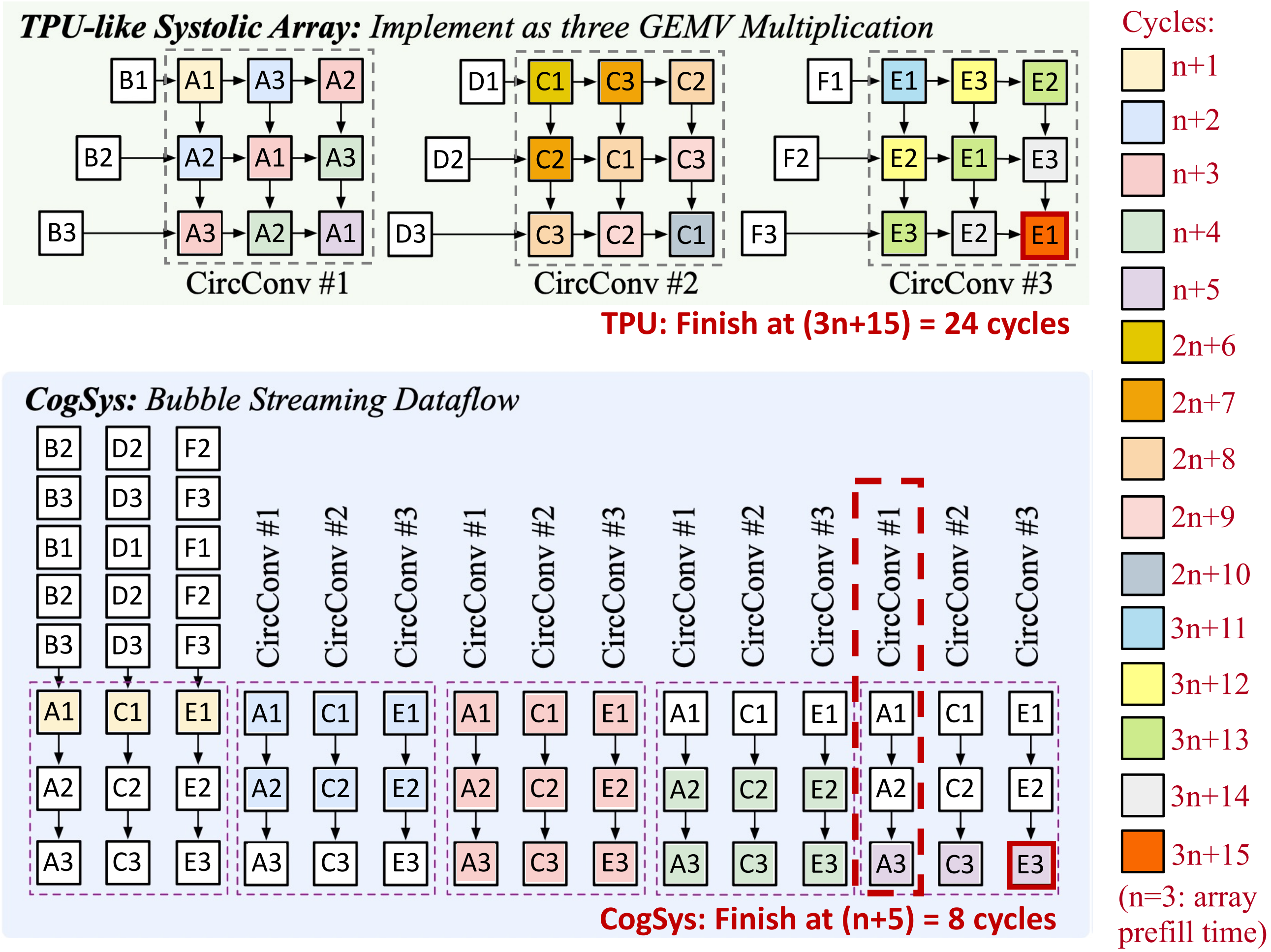
Bubble Streaming Dataflow

Vector-Symbolic Circular Convolution Example (3 CircConv):
CircConv #1: (A1, A2, A3) ⊙ (B1, B2, B3)
CircConv #2: (C1, C2, C3) ⊙ (D1, D2, D3)
CircConv #3: (E1, E2, E3) ⊙ (F1, F2, F3)

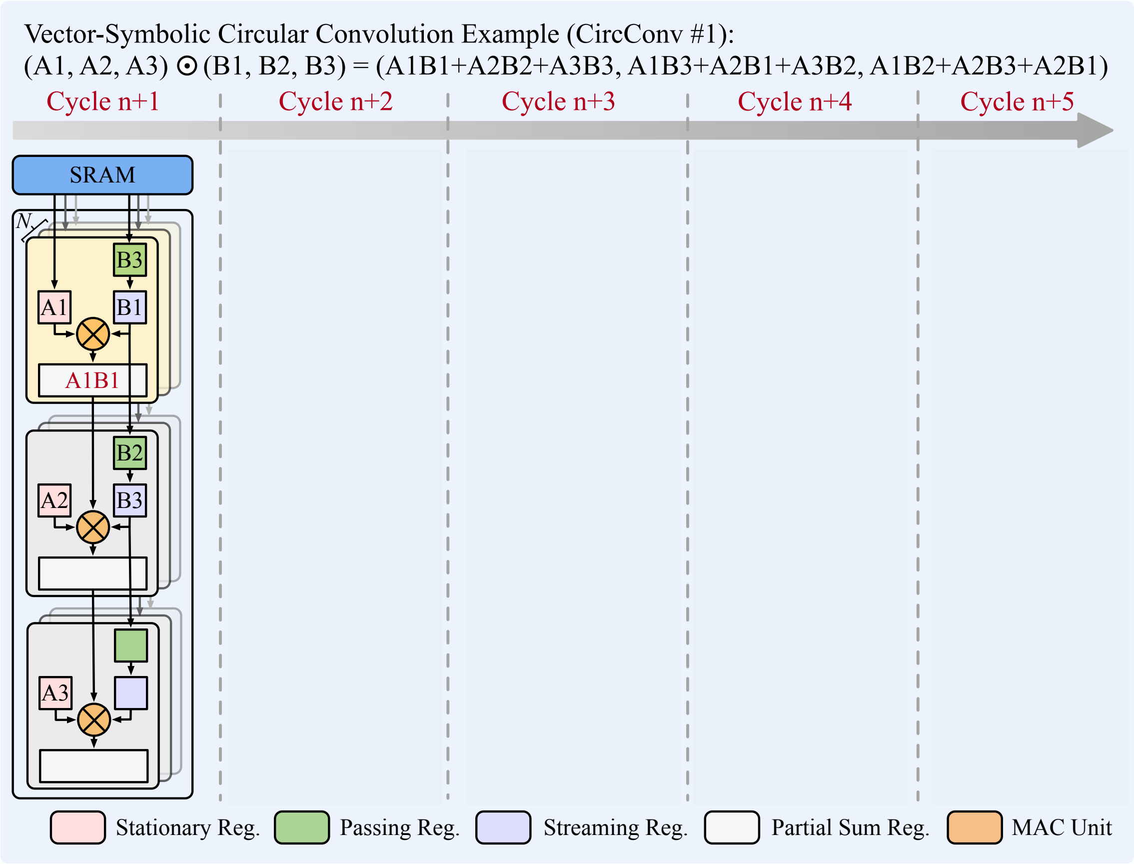
CircConv #1 Computation:
(A1, A2, A3) ⊙ (B1, B2, B3) =
(A1B1+A2B2+A3B3, A1B3+A2B1+A3B2, A1B2+A2B3+A2B1)

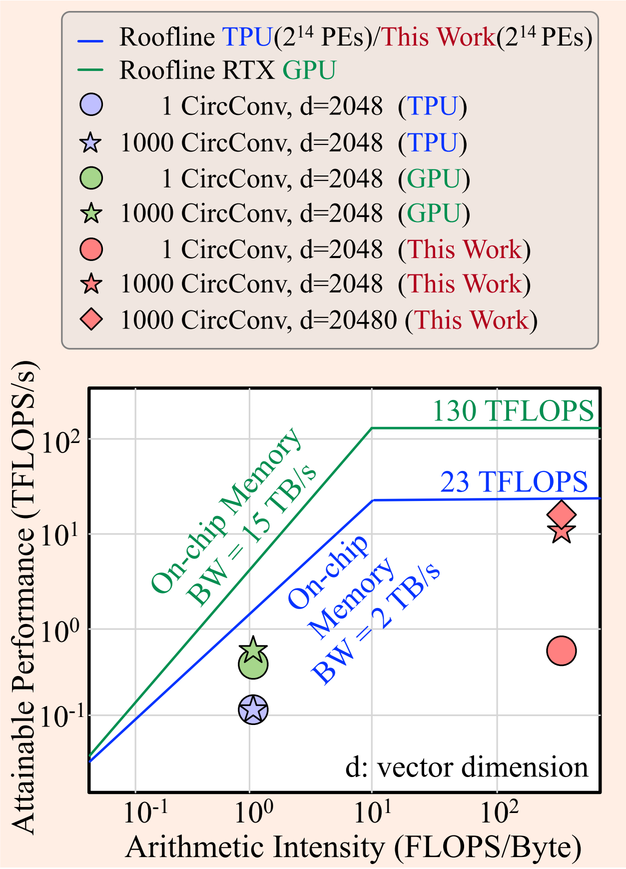
For symbolic operation:

- TPU-like array **suffers from** low parallelism & high memory access;
- Bubble streaming dataflow **improve parallelism, arithmetic intensity, and data reuse.**



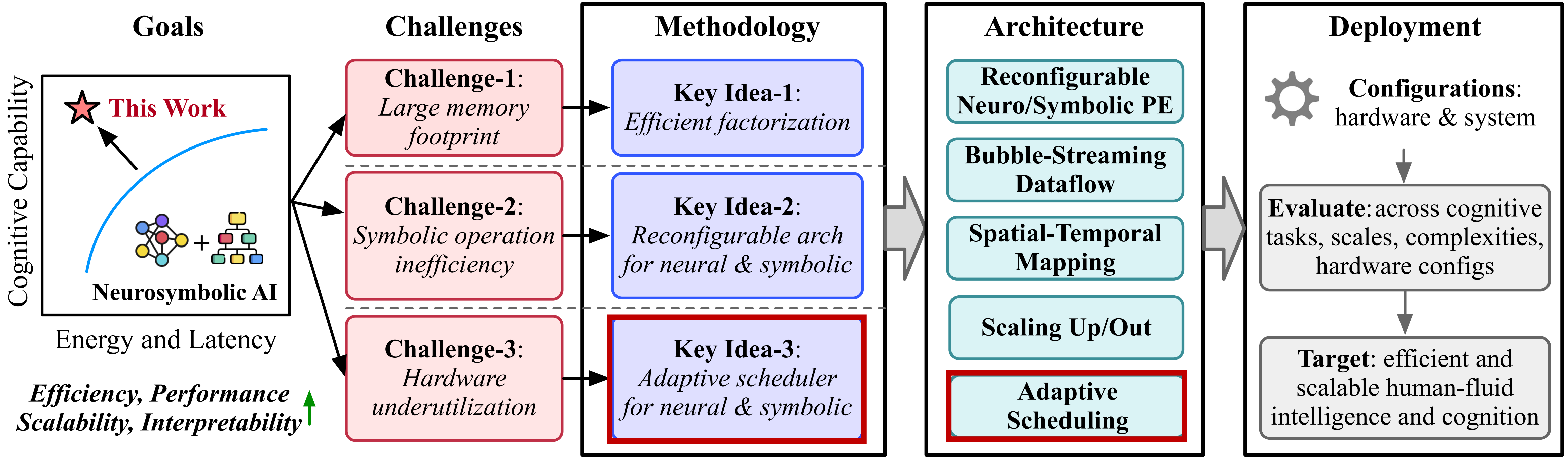
Bubble Streaming Dataflow



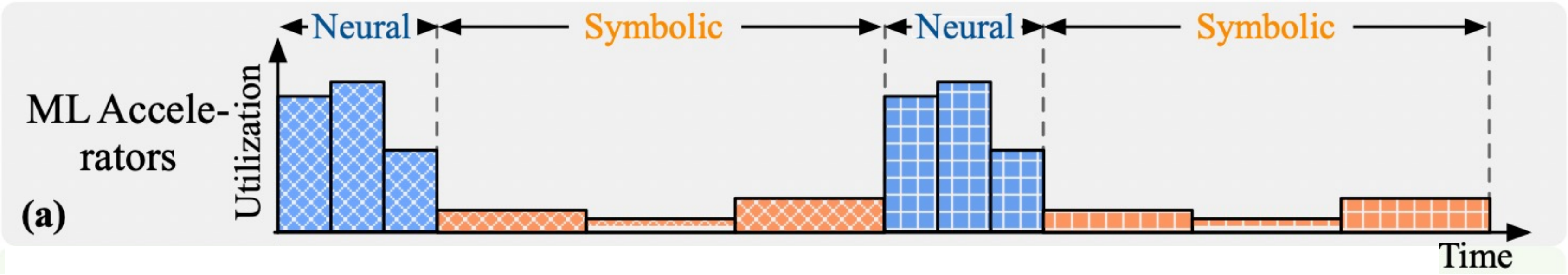


Bubble streaming dataflow flow improve parallelism, arithmetic intensity, and data reuse

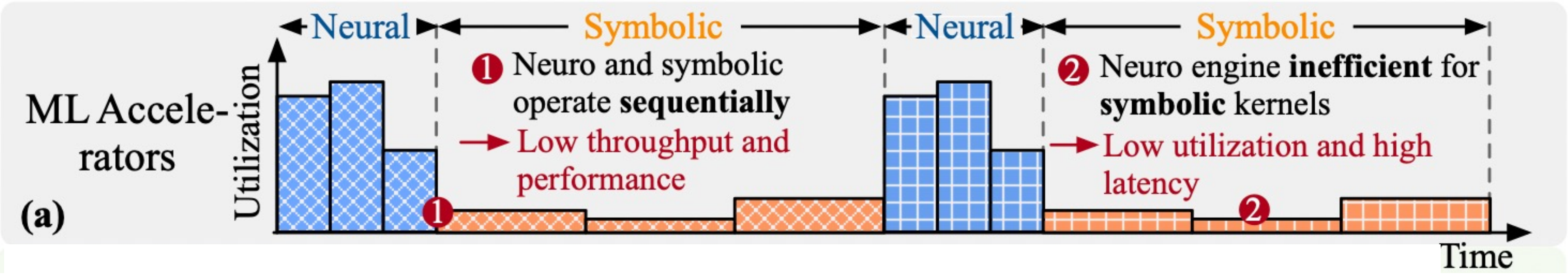
Our Methodology



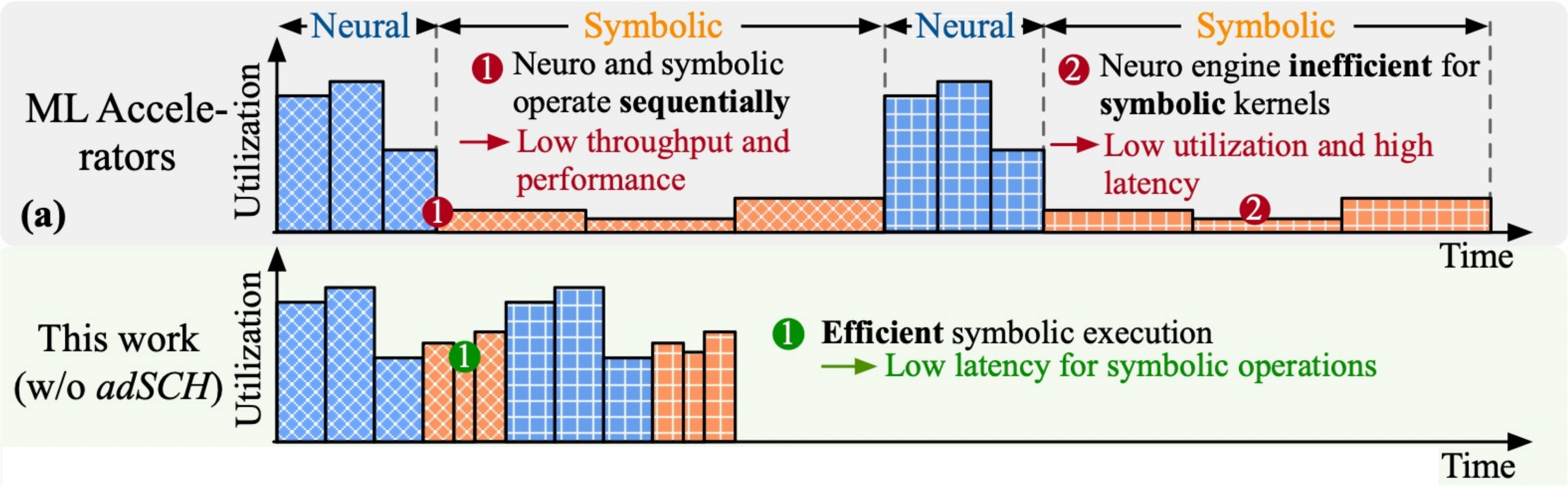
System Optimization - Adaptive Scheduling



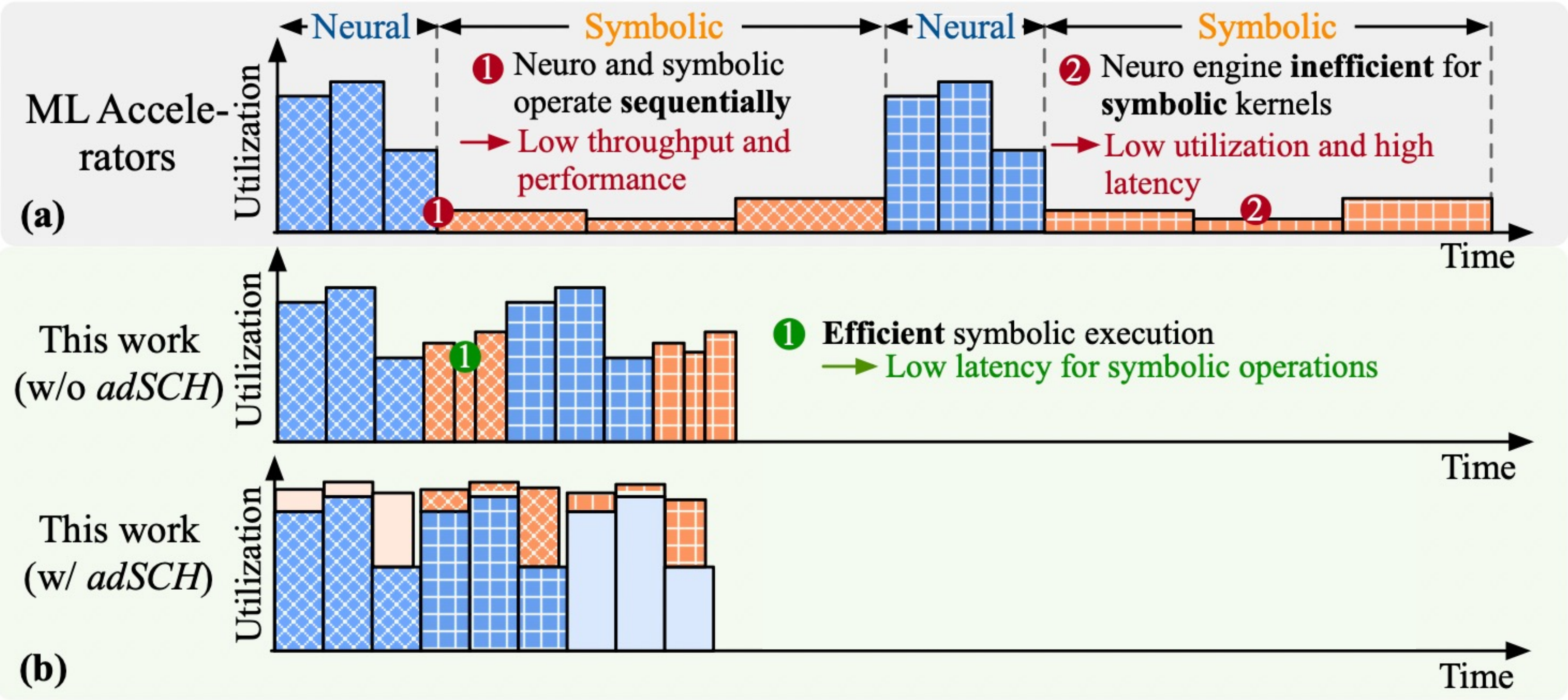
System Optimization - Adaptive Scheduling



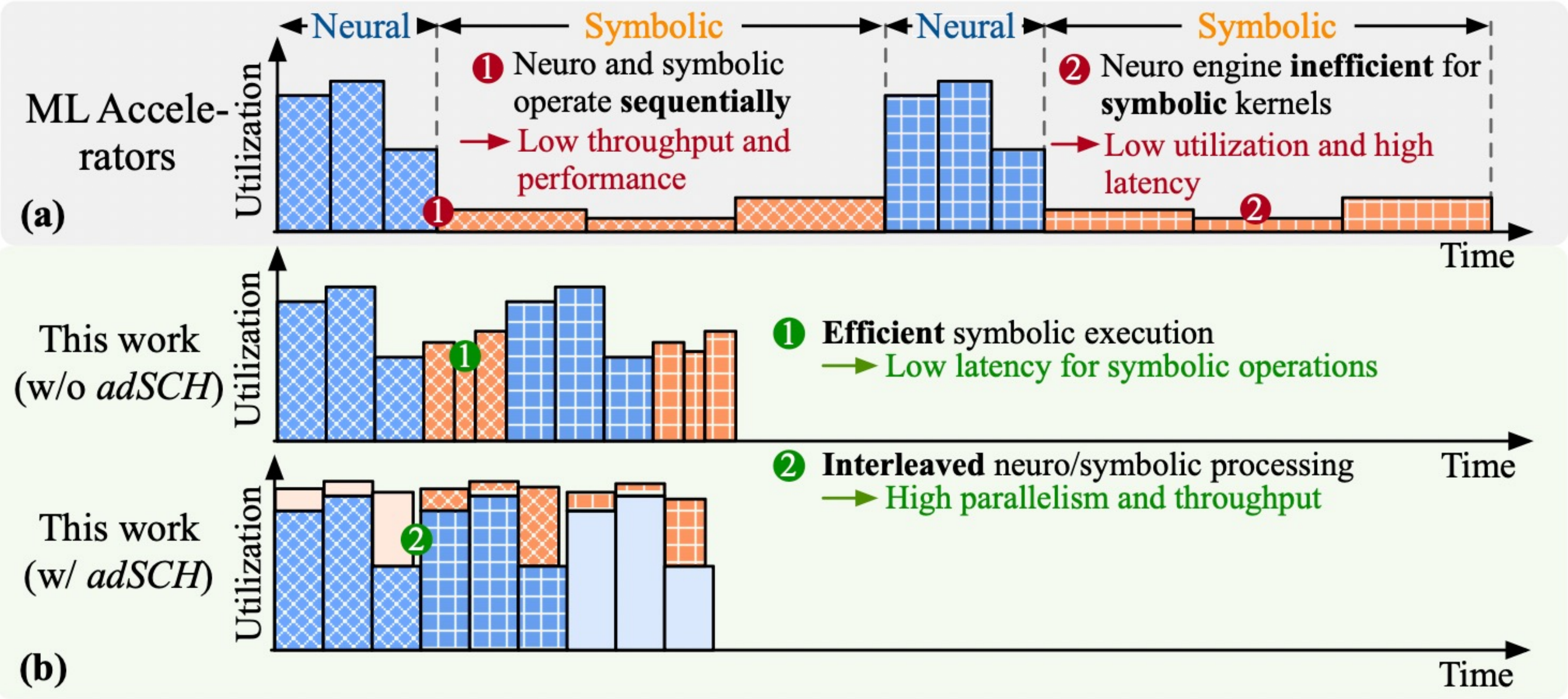
System Optimization - Adaptive Scheduling



System Optimization - Adaptive Scheduling

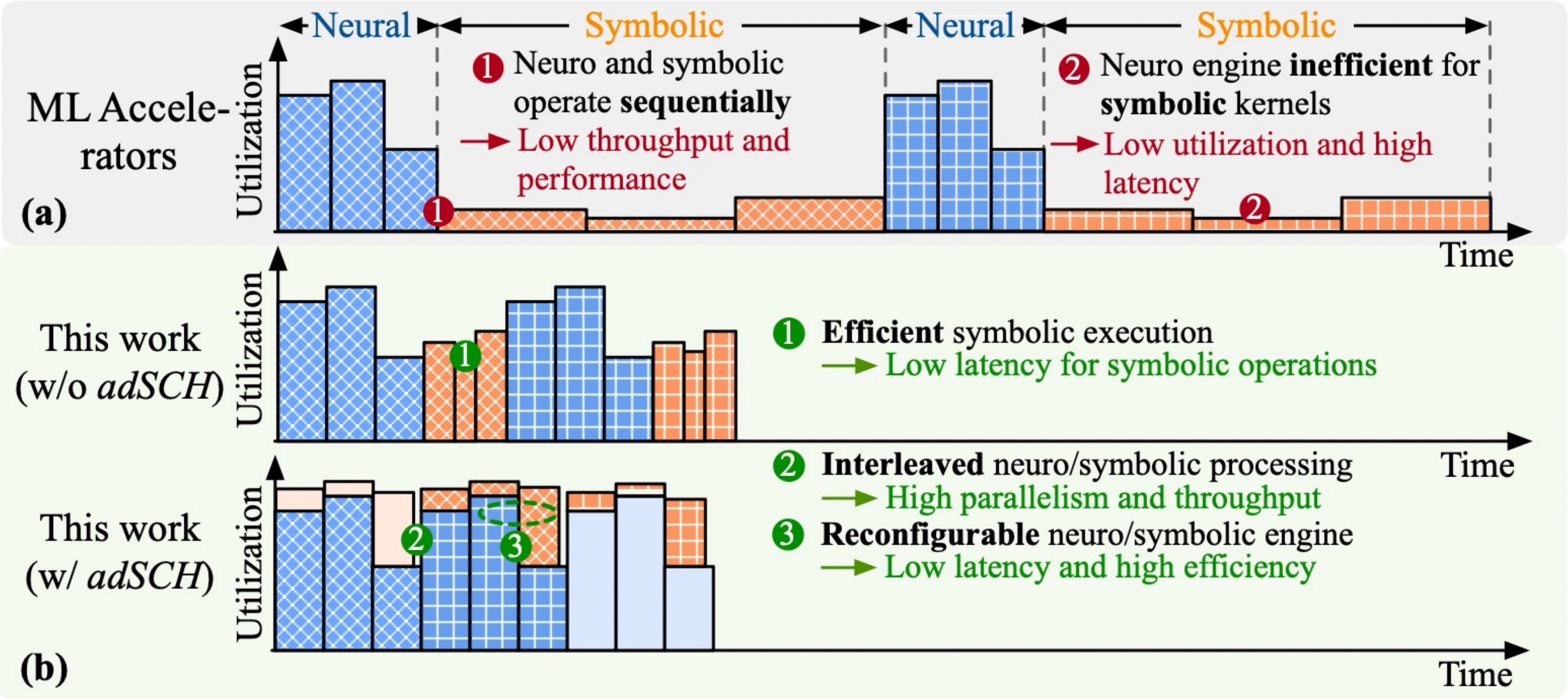


System Optimization - Adaptive Scheduling



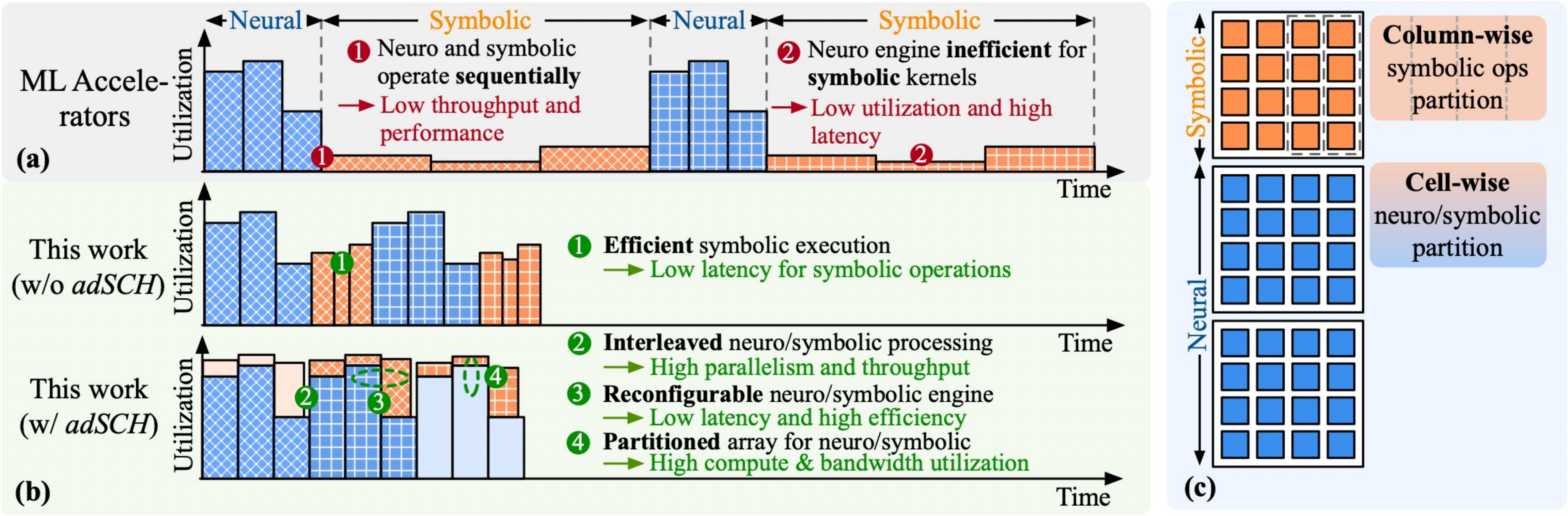
Adaptive scheduling enables **interleaved**

System Optimization - Adaptive Scheduling



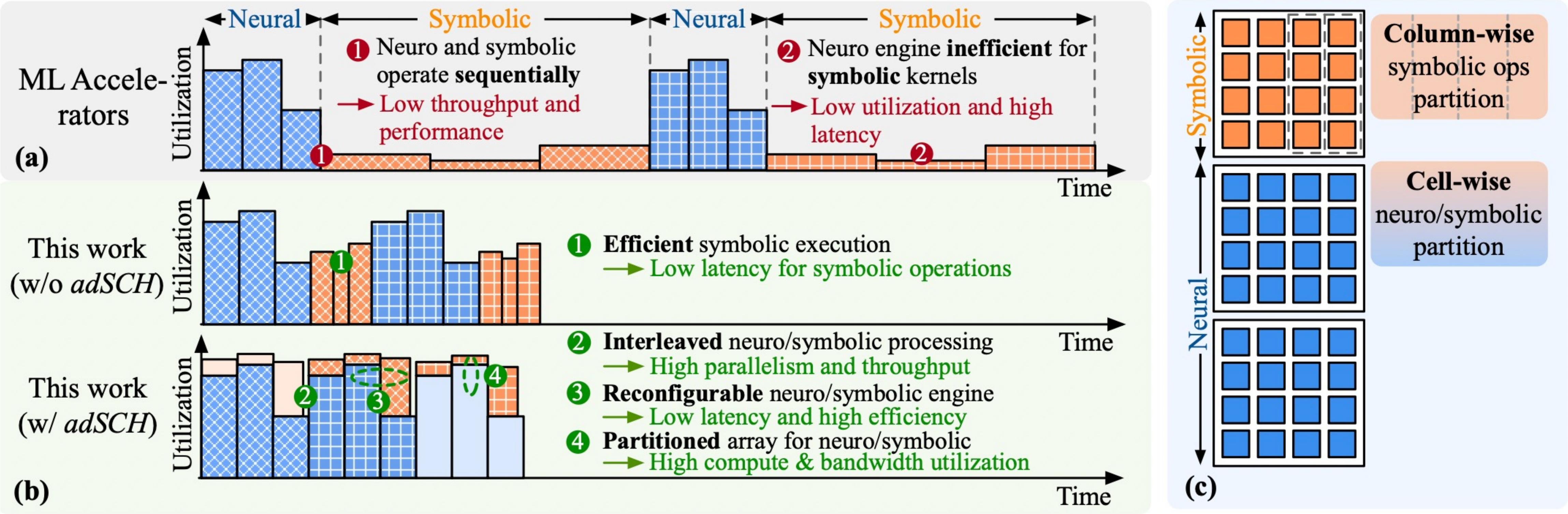
Adaptive scheduling enables **interleaved** and **reconfigurable** neuro/symbolic processing

System Optimization - Adaptive Scheduling



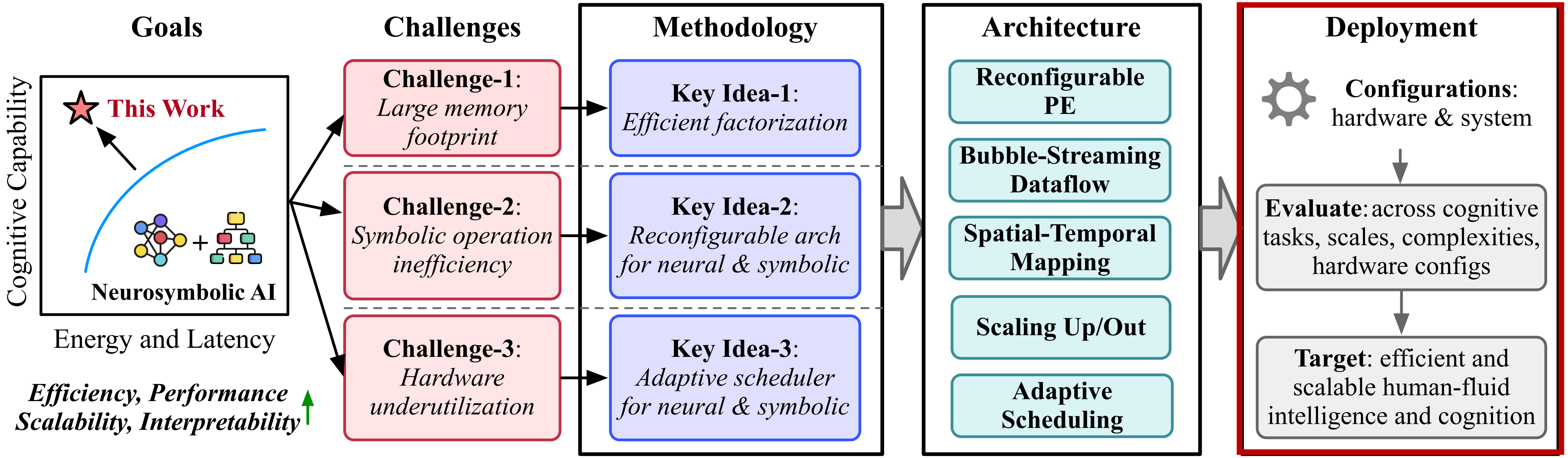
Adaptive scheduling enables **interleaved** and **reconfigurable** neuro/symbolic processing with **partitioned array**

System Optimization - Adaptive Scheduling

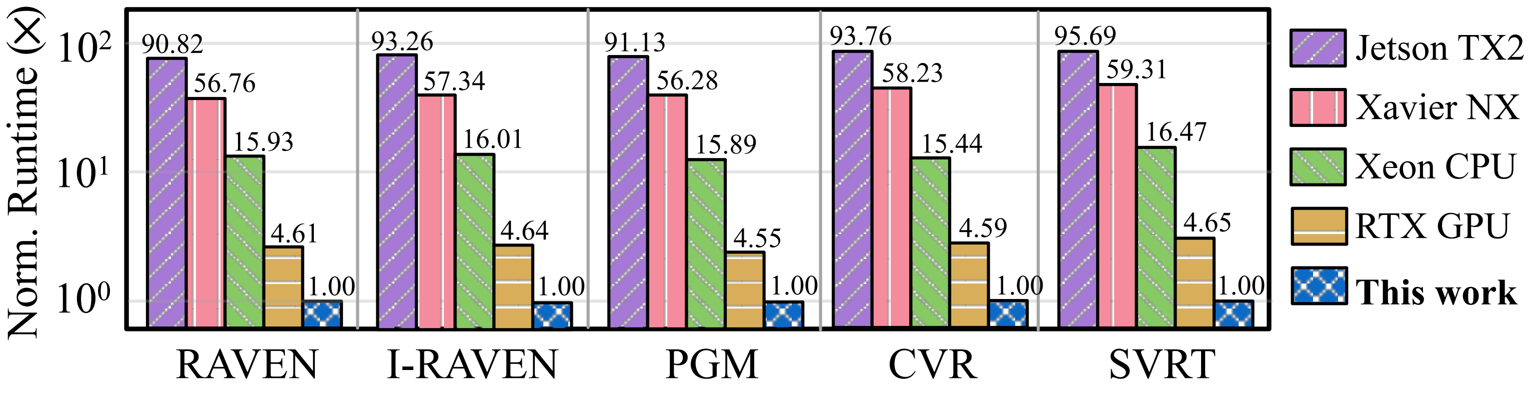


Adaptive scheduling enables **interleaved** and **reconfigurable** neuro/symbolic processing with **partitioned array**, improving parallelism, latency, efficiency, and utilization

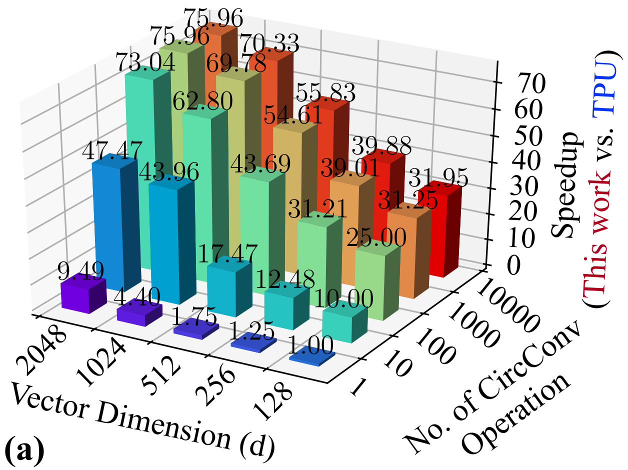
Our Methodology

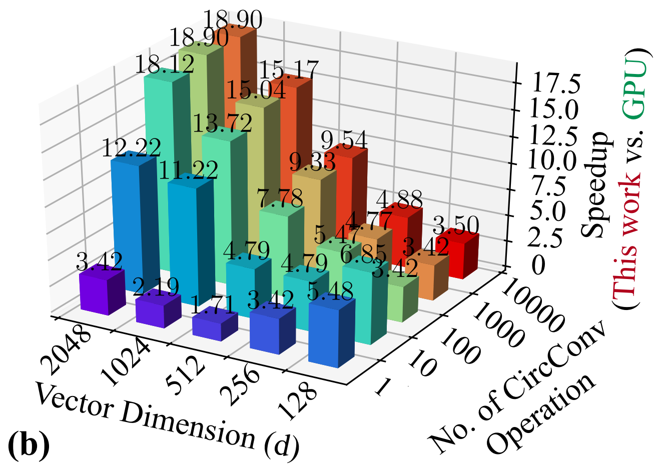


Evaluation – Hardware Performance



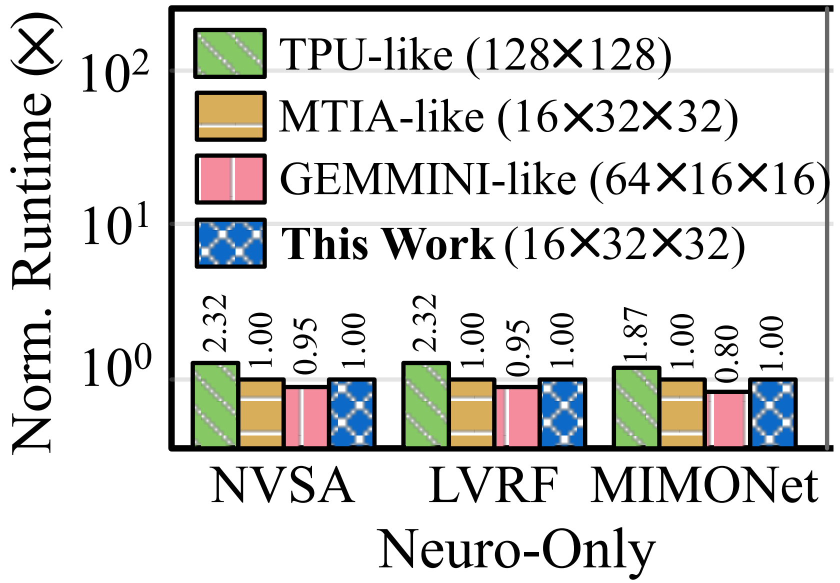
4x - 90x speedup compared to CPU/GPU





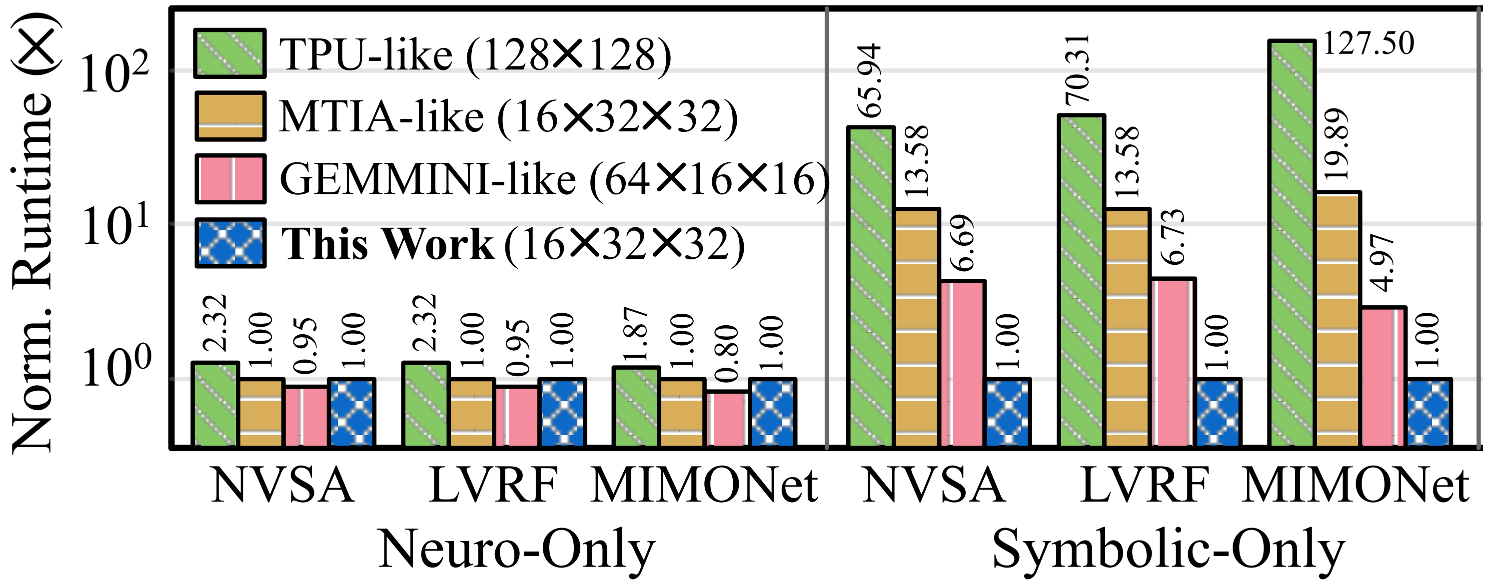
Symbolic operation:
75x speedup to TPU
18x speedup to GPU

Evaluation – Hardware Performance



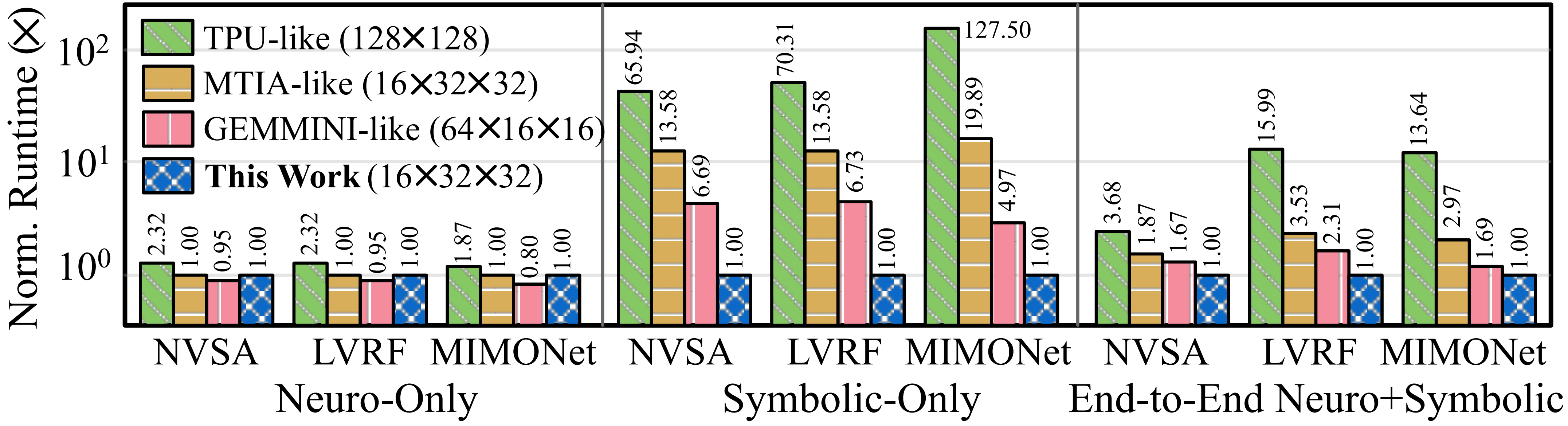
Compared with ML accelerators: similar neuro latency,

Evaluation – Hardware Performance



Compared with ML accelerators: similar neuro latency, **7-120x symbolic** speedup,

Evaluation – Hardware Performance



Compared with ML accelerators: similar neuro latency, **7-120x symbolic** speedup, **2-16x end-to-end neuro-symbolic** speedup

Key Observations:

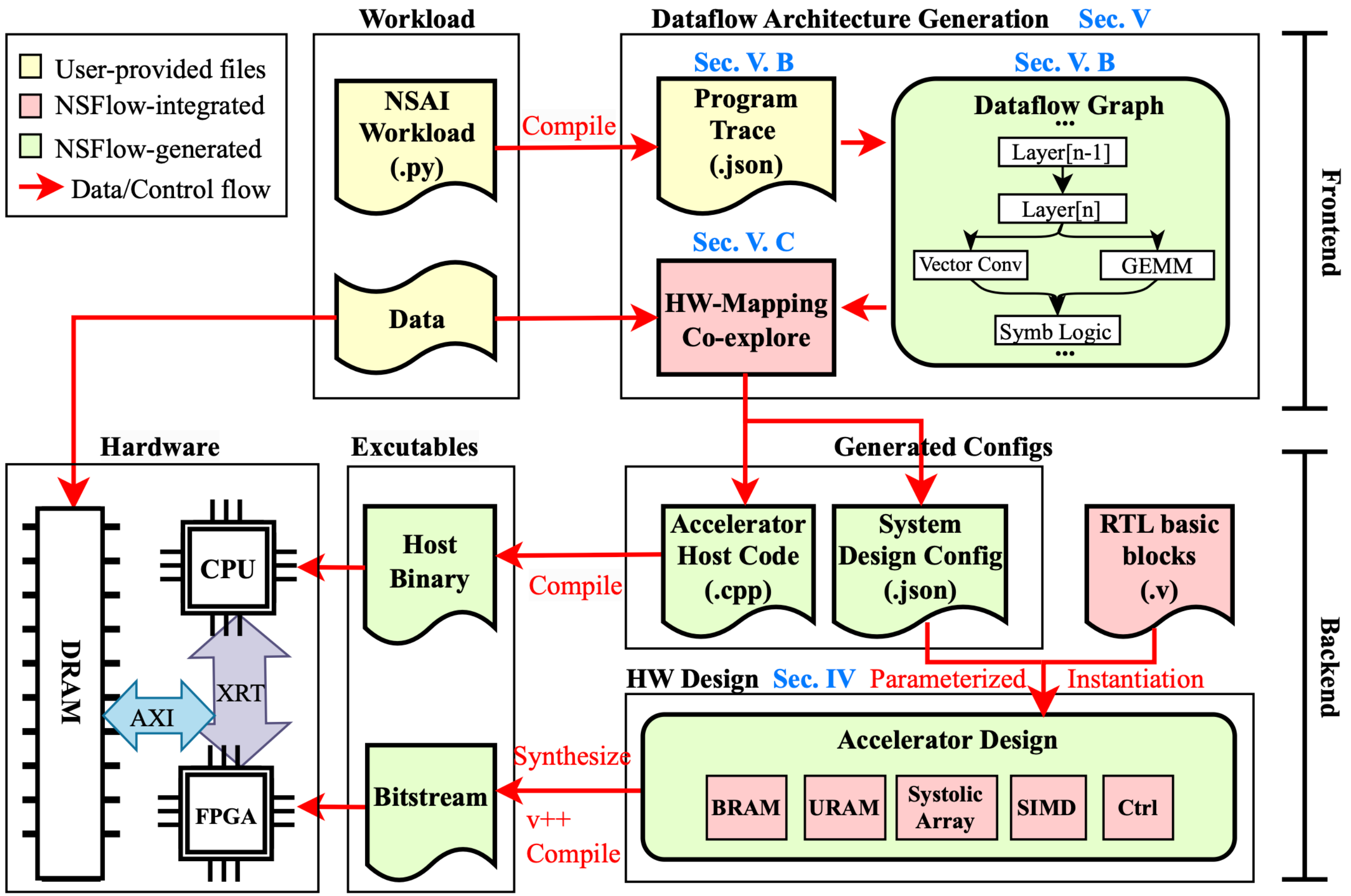
Compared with systolic arrays that only support neural, our design provides **reconfigurable support for neural and symbolic** operations with **only 4.8% area overhead**.

Our design achieves **0.3s latency** per cognition task, with **1.18W power** consumption.

Research Question:

How to **automate** this neuro-symbolic architecture **design** process?

Automated End-to-End FPGA Deployment



Frontend: dataflow arch generator

- Step 1: Extract execution trace
- Step 2: Generate dataflow graph
- Step 3: HW-mapping co-exploration

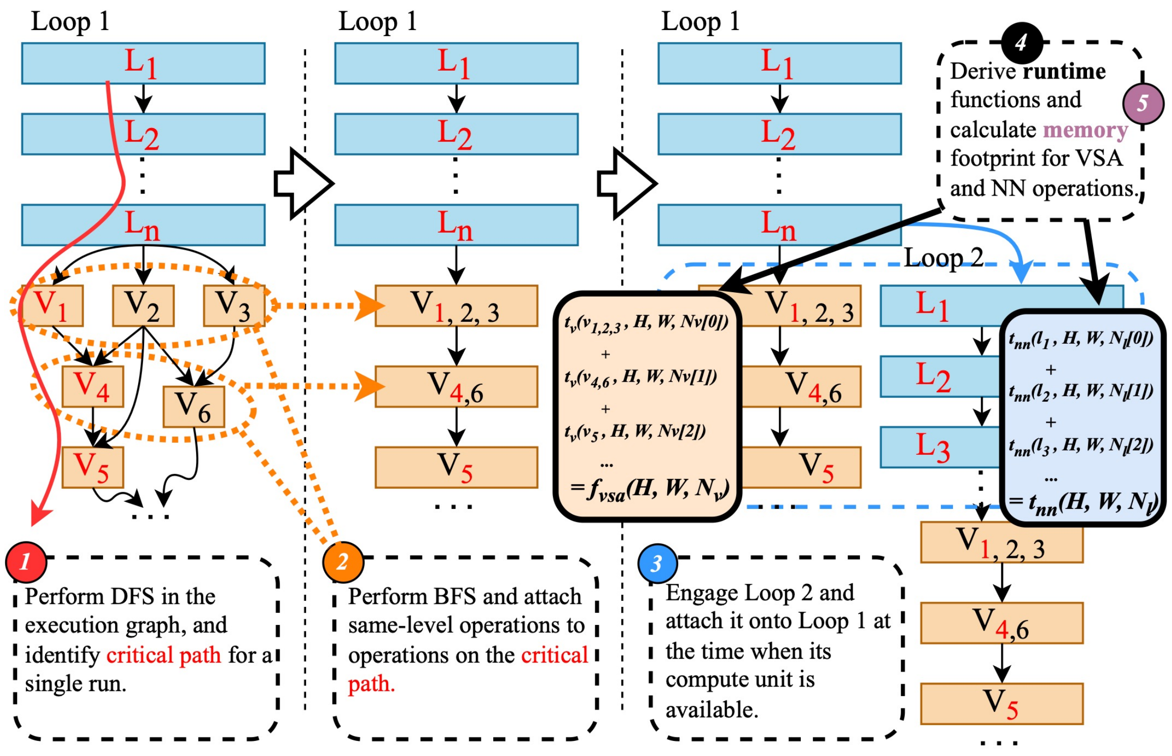
Backend: FPGA deployment

- Step 1: Pre-define hardware template
- Step 2: Configure design parameters
- Step 3: Synthesize and compile RTL

Frontend – Dataflow architecture Generation

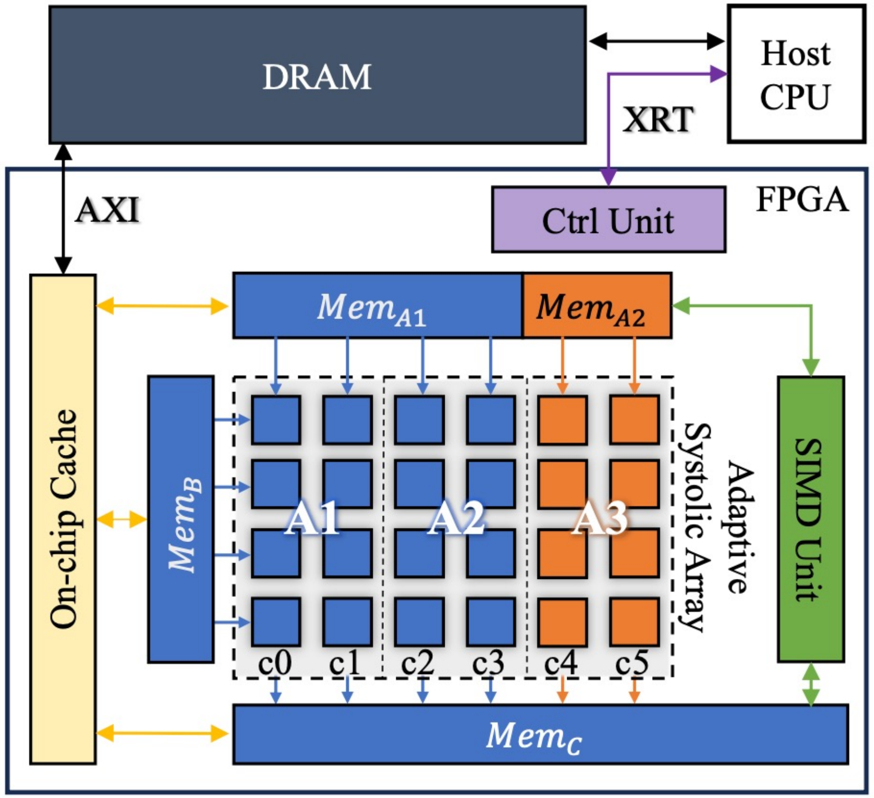
```
graph():
    ...
    // Neuro Operation - CNN (Resnet18)
    %relu_1[16,64,160,160] : call_module[relu](args = (%bn1
         [16,64,160,160]))
    %maxpool_1[16,64,160,160] : call_module[maxpool](args =
         (%relu_1[16,64,160,160]))
    %conv2d_1[16,64,160,160] : call_module[conv2d](args =
         (%maxpool_1[16,64,160,160]))
    ...
    // Symbolic Operations
    // Inverse binding of two block codes vectors by
         blockwise cicular correlation
    %inv_binding_circular_1[1,4,256] : call_function[nvsa.
         inv_binding_circular](args = (%vec_0[1,4,256], %
         vec_1[1,4,256]))
    %inv_binding_circular_2[1,4,256] : call_function[nvsa.
         inv_binding_circular](args = (%vec_3[1,4,256], %
         vec_4[1,4,256]))
    // Compute similarity between two block codes vectors
    %match_prob_1[1] : call_function[nvsa.match_prob](args
         = (%inv_binding_circular_1[1,4,256], %vec_2
         [1,4,256]))
    // Compute similarity between a dictionary and a batch
         of query vectors
    %match_prob_multi_batched_1[1]: call_function[nvsa.
         match_prob_multi_batched](args = (%
         inv_binding_circular_2[1,4,256], %vec_5[7,4,256]))
    %sum_1[1] : call_function[torch.sum](args = (%
         match_prob_multi_batched_1[1]))
    %clamp_1[1] : call_function[torch.clamp](args = (%sum_1
         [1]))
    %mul_1[1] : call_function[operator.mul](args = (%
         match_prob_1[1], %clamp_1[1]))
    ...
```

Extract workload execution trace

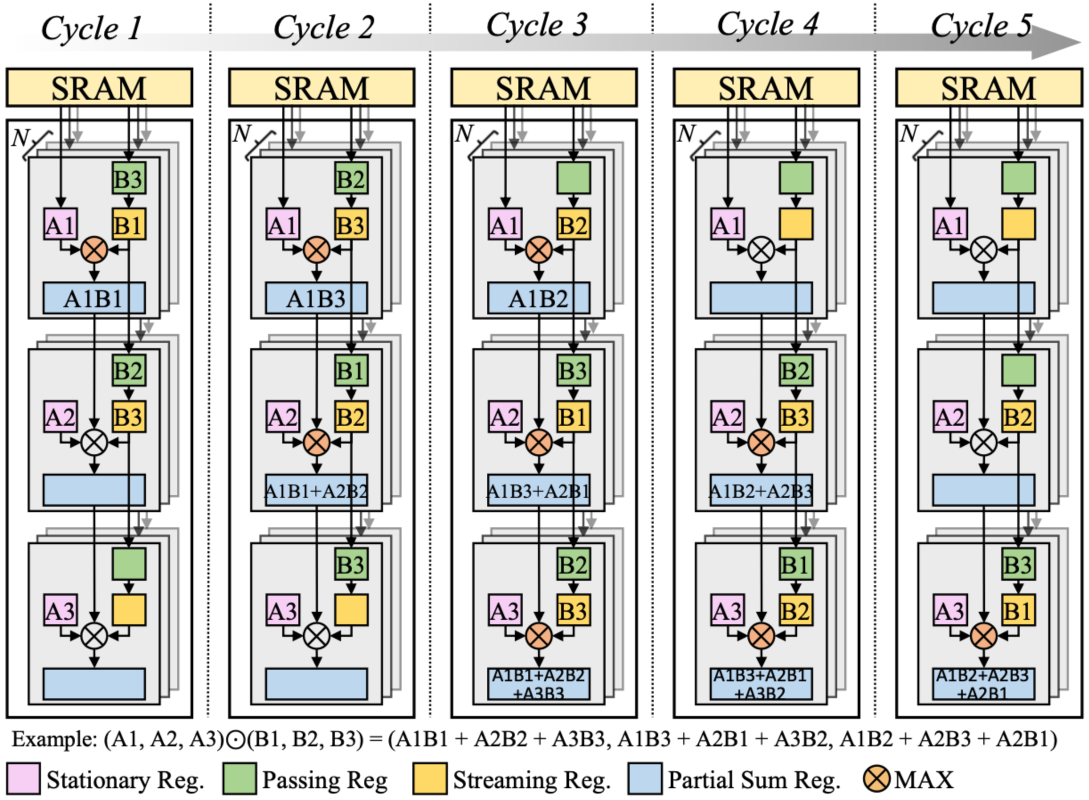


Generate dataflow graph &
two-stage HW-mapping co-exploration

Backend – FPGA Deployment



Pre-defined architecture template

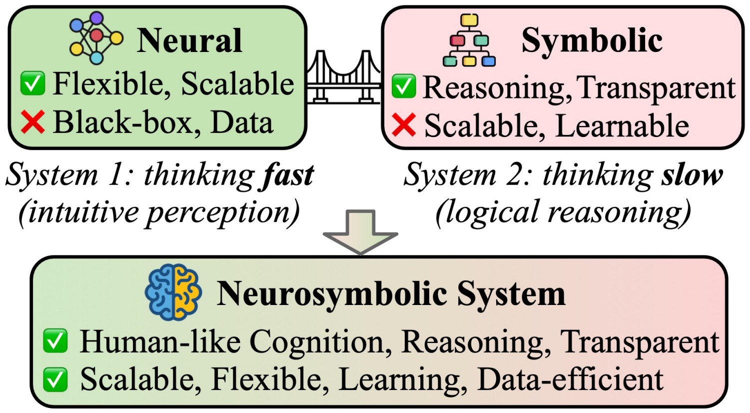


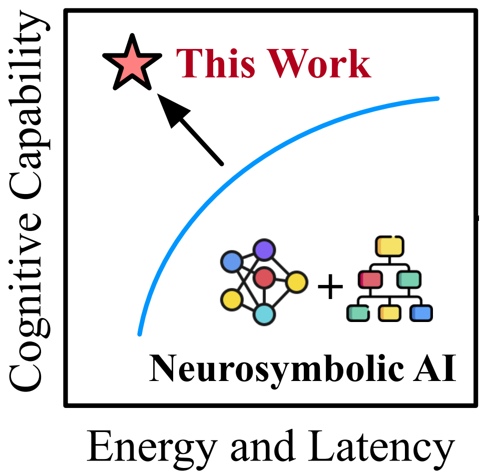
Dataflow & configure design parameters

Summary

- **Neuro-symbolic AI** is a compositional method to improve agent reasoning and interpretability.

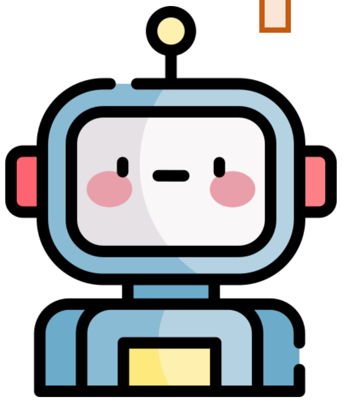
- In these work,
  - **Model**: Characterize workload implications
  - **Architecture**: Reconfigurable neuro-symbolic PE, dataflow, mapping
  - **System**: adaptive workload scheduling
  - **FPGA**: automated end-to-end FPGA deployment
  - **ASIC SoC**: programmable neuro-symbolic SoC
  - Achieve **efficient and scalable neuro-symbolic** execution across agentic reasoning tasks





Efficiency, Performance
Scalability, Interpretability

Autonomous Machines (Agentic System)



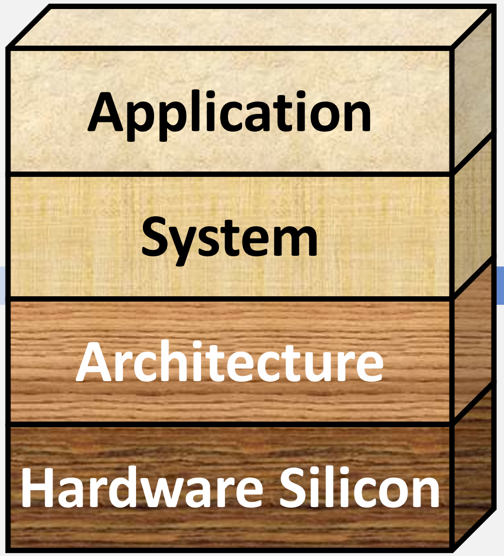
Cognition Capability

Human-like reasoning
Trustworthy decision making
Human-agent interaction

Autonomy Capability

Perception, Localization,
Mapping, Planning, Control,
Learning-based navigation

My Research:
Cross-Layer Approach



Requirements

Real-Time ***Performance***

Energy ***Efficiency***

Design ***Scalability***

Safety-Critical ***Reliability***

Research Question:

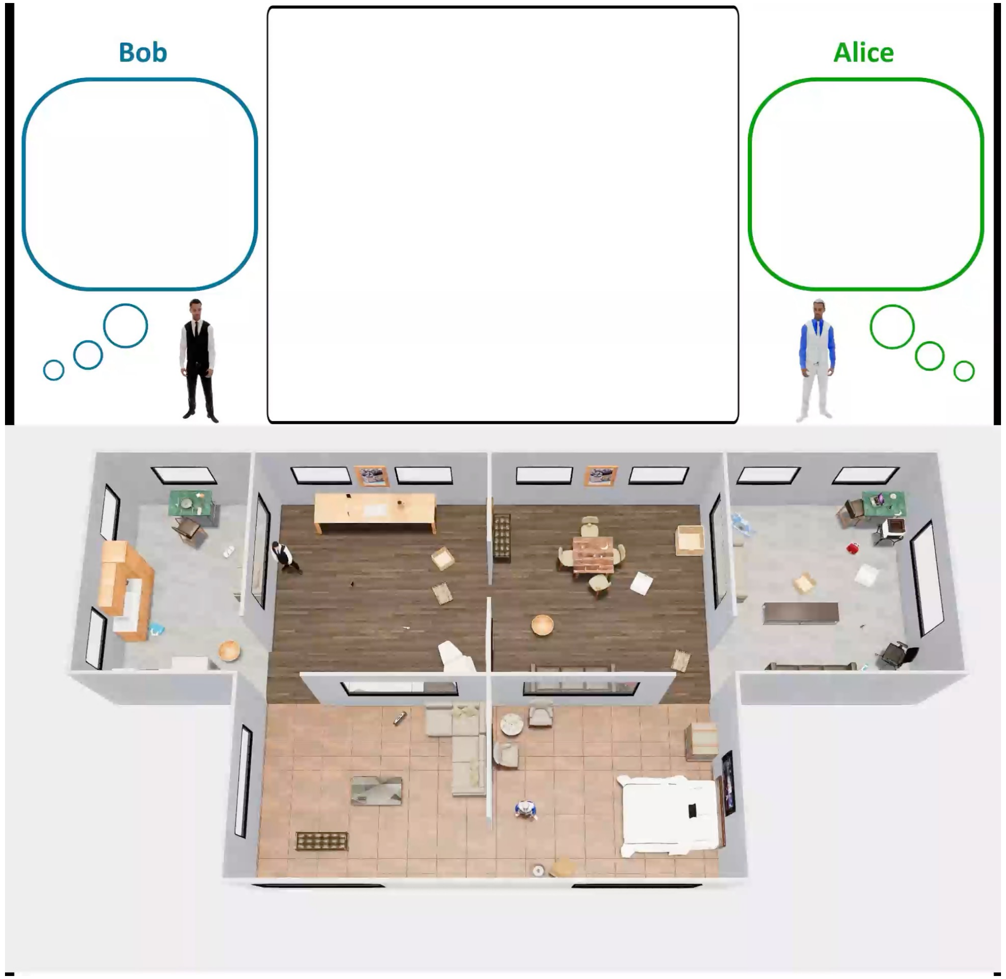
Can autonomous agents collaboratively conduct complex **long-horizon multi-objective tasks**?

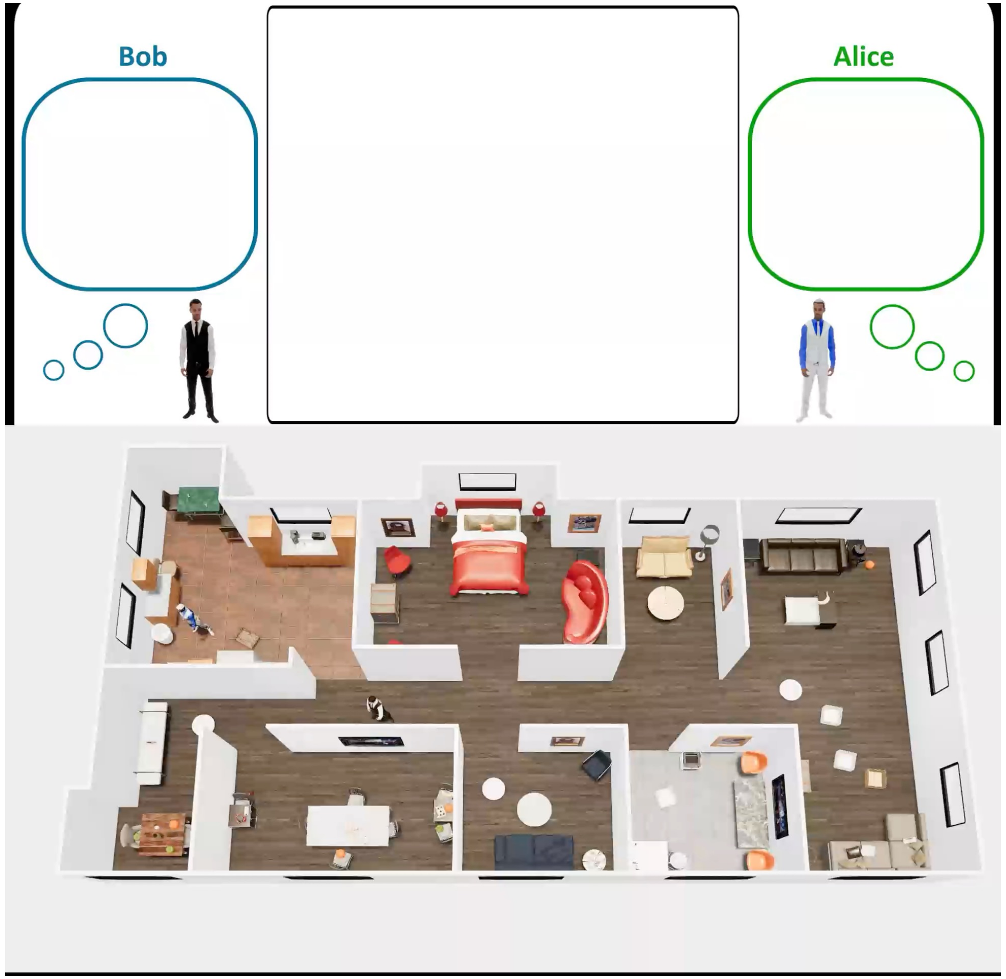
What's **system characteristics** of embodied agents?
How to improve **system efficiency**?

Embodied Autonomous Agent System

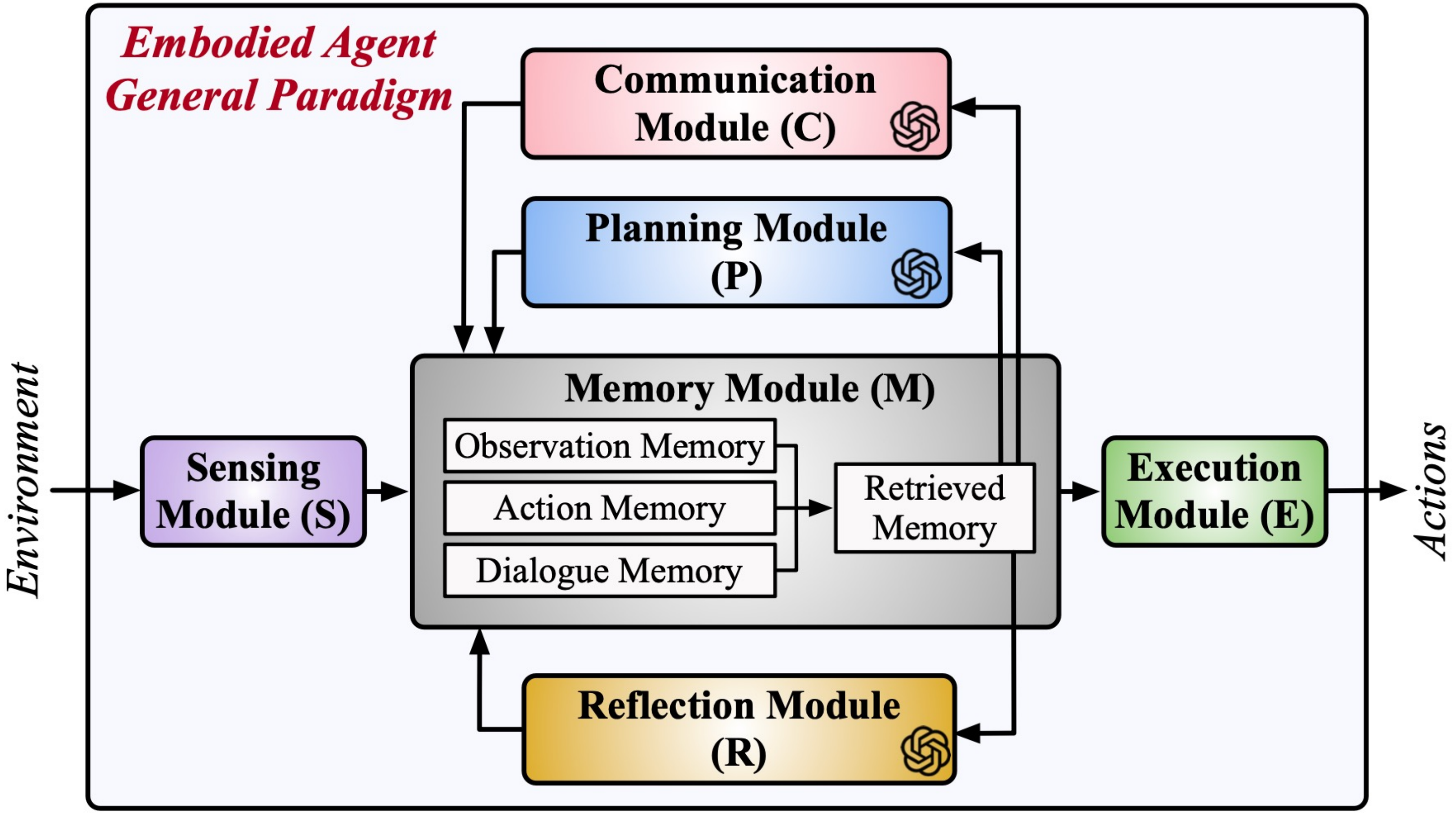
- **Task**: long-horizon multi-objective task and motion planning
  - Examples: household tasks, transport objects, make meal, set up table, cook…

Demo: Long-Horizon Multi-Objective Planning

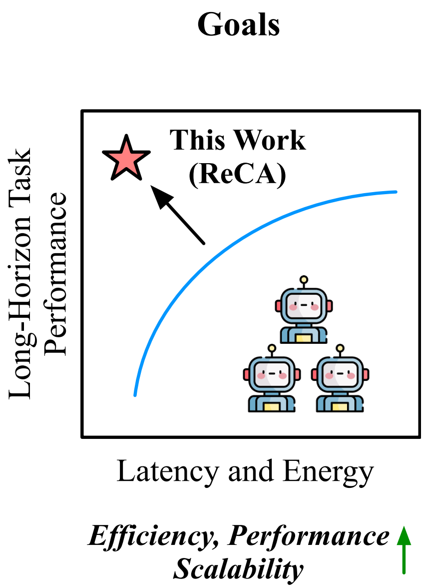




Embodied AI Agent Workflow

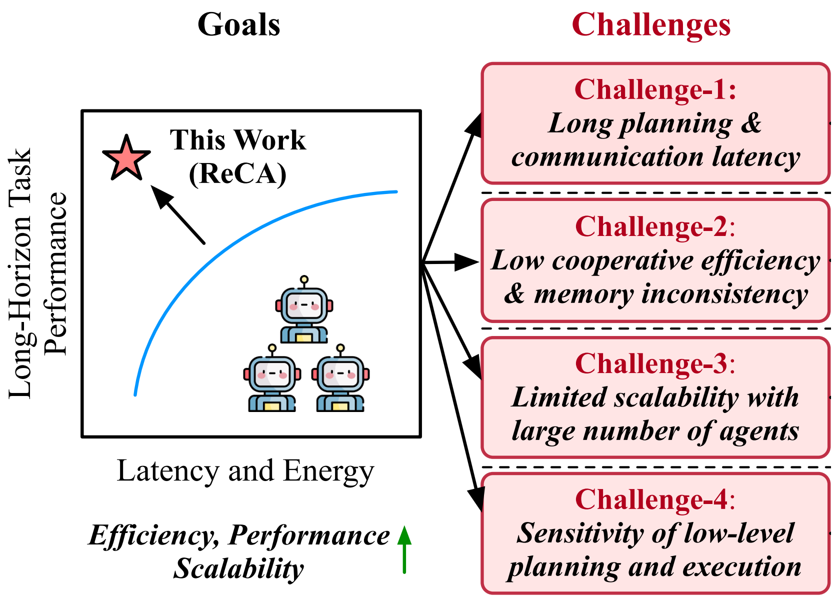


Embodied AI Agent System Characterization



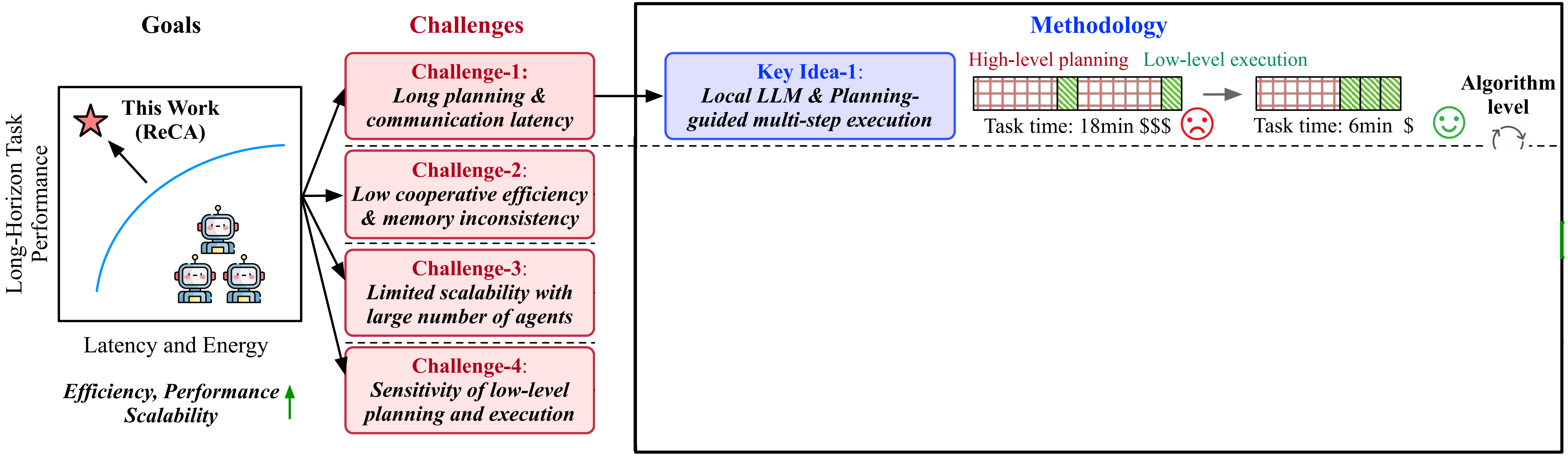
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Characterization



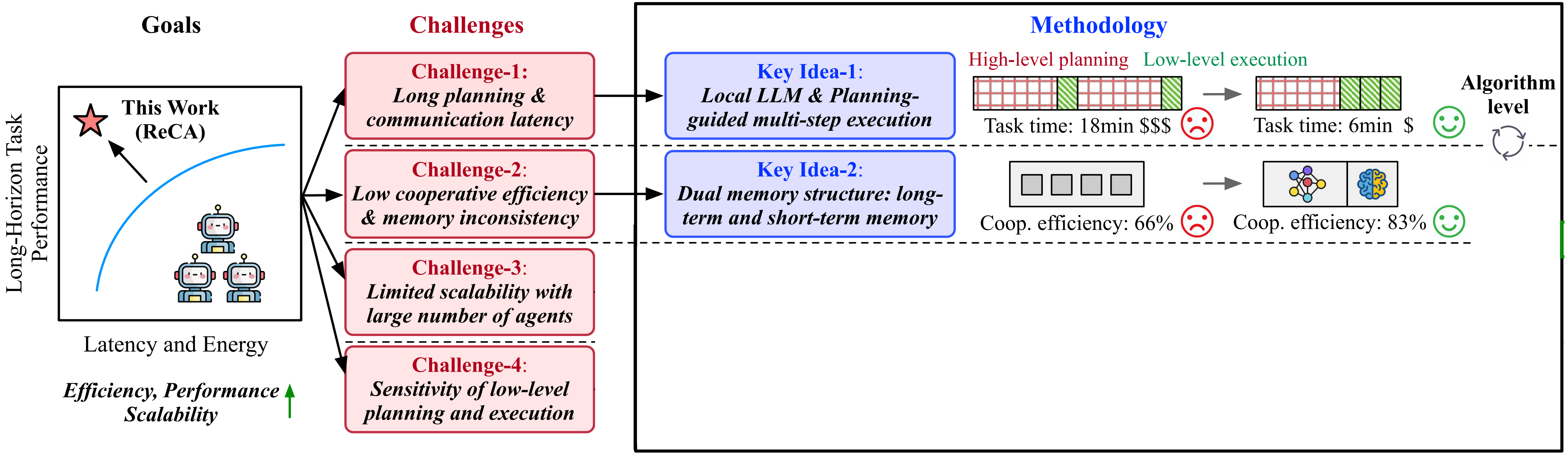
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Optimization



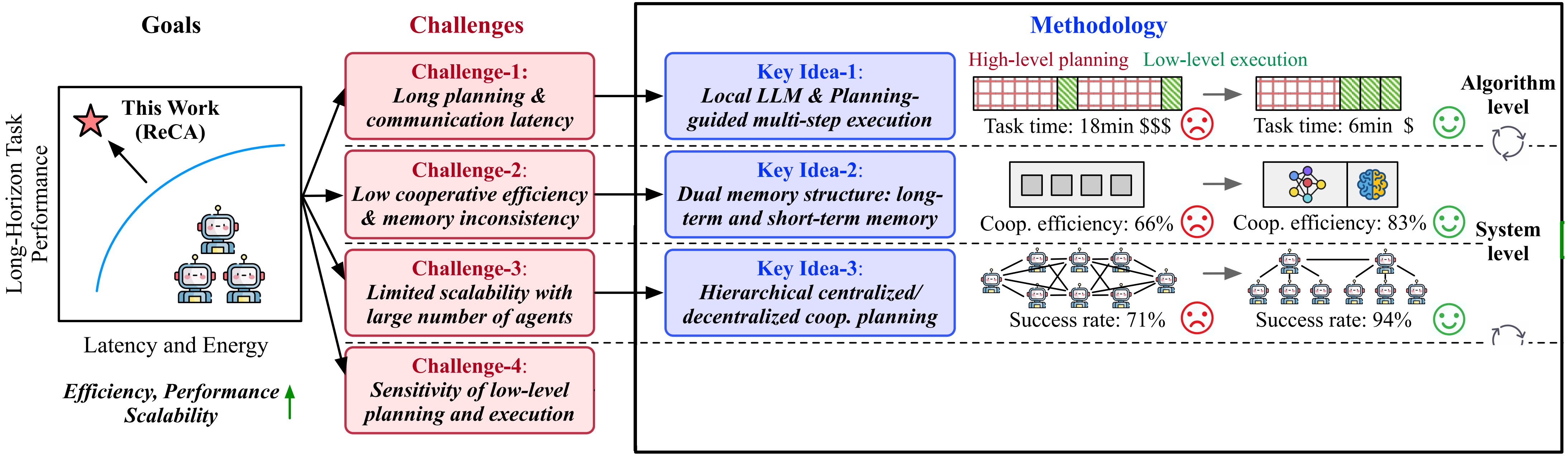
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Optimization



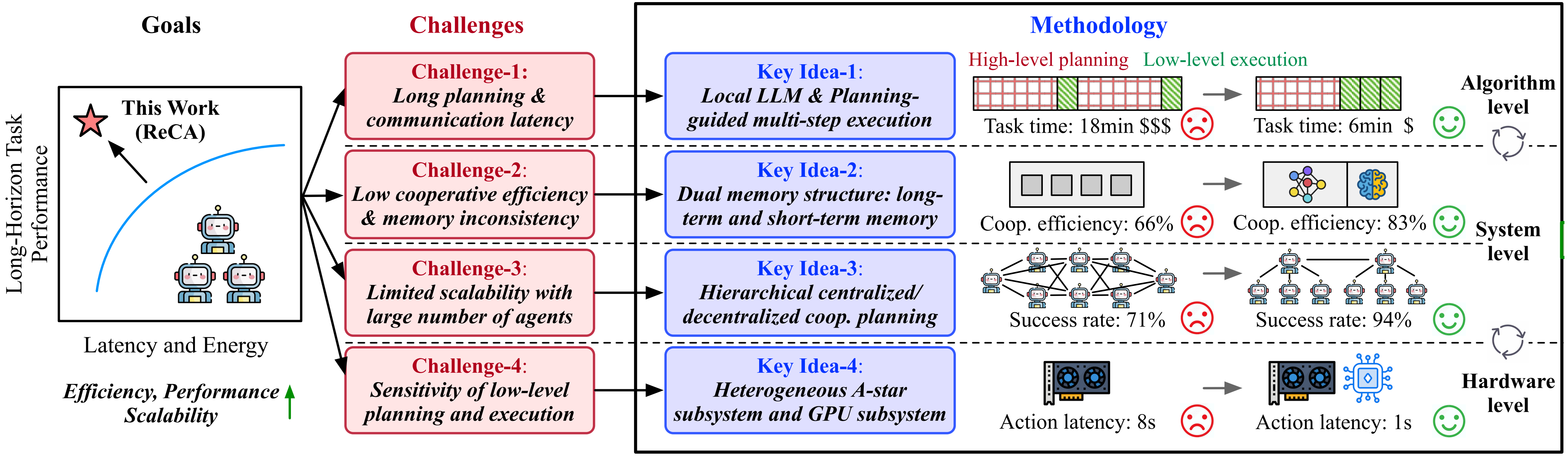
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Optimization



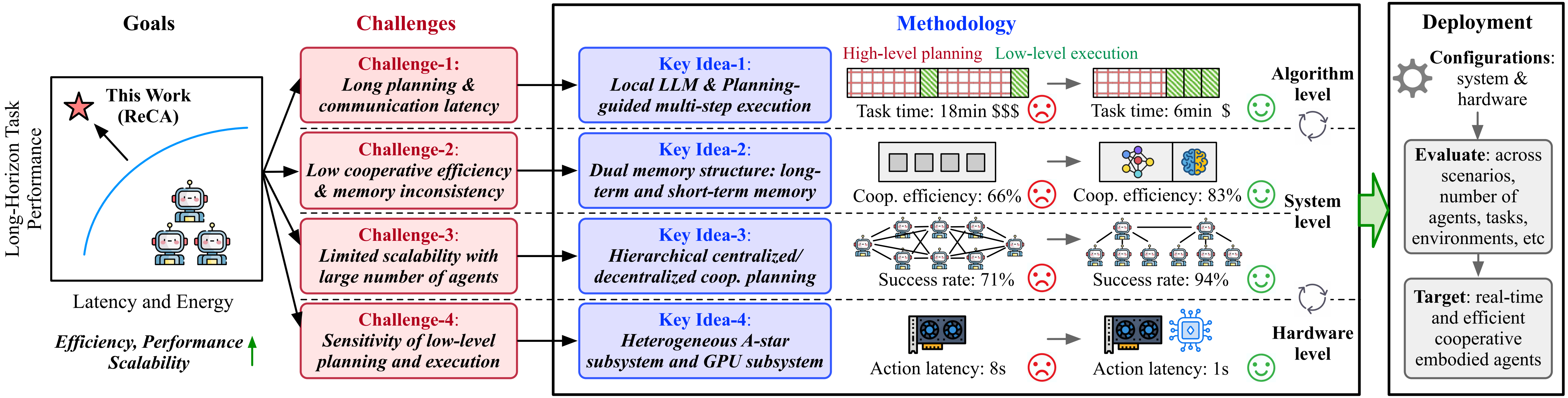
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Optimization



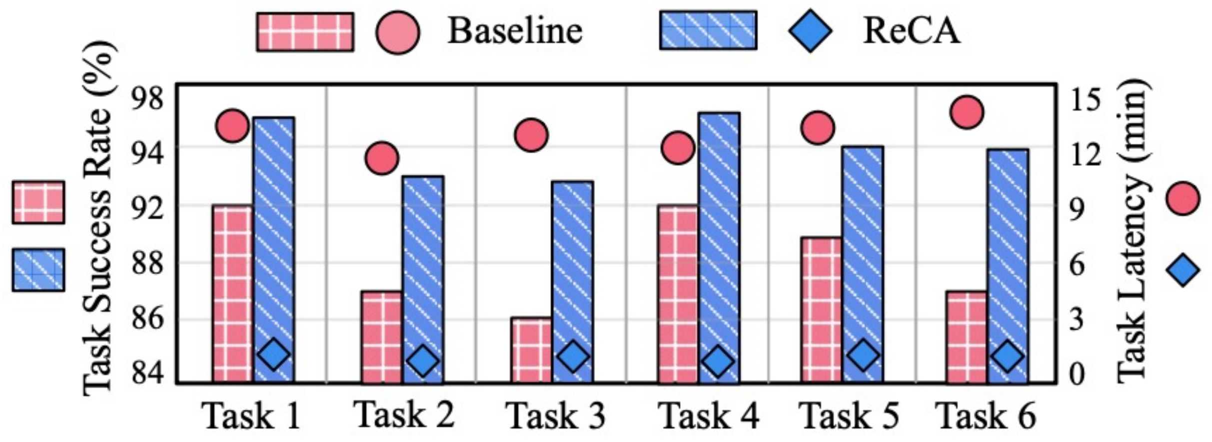
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Embodied AI Agent System Optimization



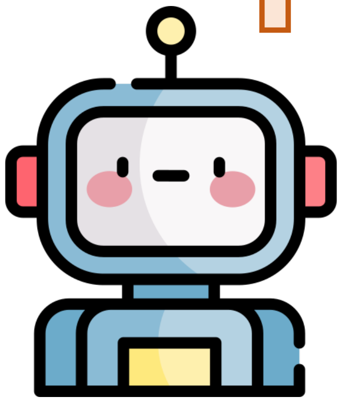
Goal: Improve runtime efficiency, performance, and scalability of cooperative embodied AI agent systems

Evaluation Results



Task	Descriptions
1	Find and place 3 forks and 1 plate into the dishwasher
2	Find and place 1 bottle of wine, 1 pancake, 1 pound cake, 1 juice, and 1 apple on the kitchen table
3	Find and place 3 forks into the dishwasher
4	Find and place 1 pudding, 1 juice, 1 apple, and 2 cupcakes on the coffee table
5	Find and place 1 bottle of wine, 2 cupcakes, and 1 pudding on the coffee table
6	Find and place 1 bottle of wine, 1 juice, 1 apple, 1 cupcake, and 1 pound cake on the kitchen table

Autonomous Machines (Agentic System)



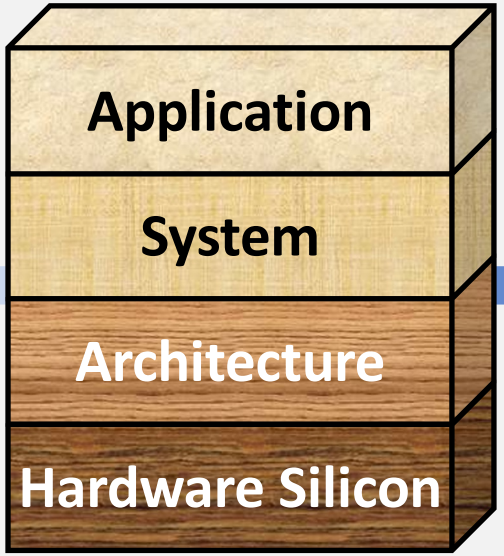
Cognition Capability

Human-like reasoning
Trustworthy decision making
Human-agent interaction

Autonomy Capability

Perception, Localization, Mapping, Planning, Control, Learning-based navigation

My Research:
Cross-Layer Approach



Requirements

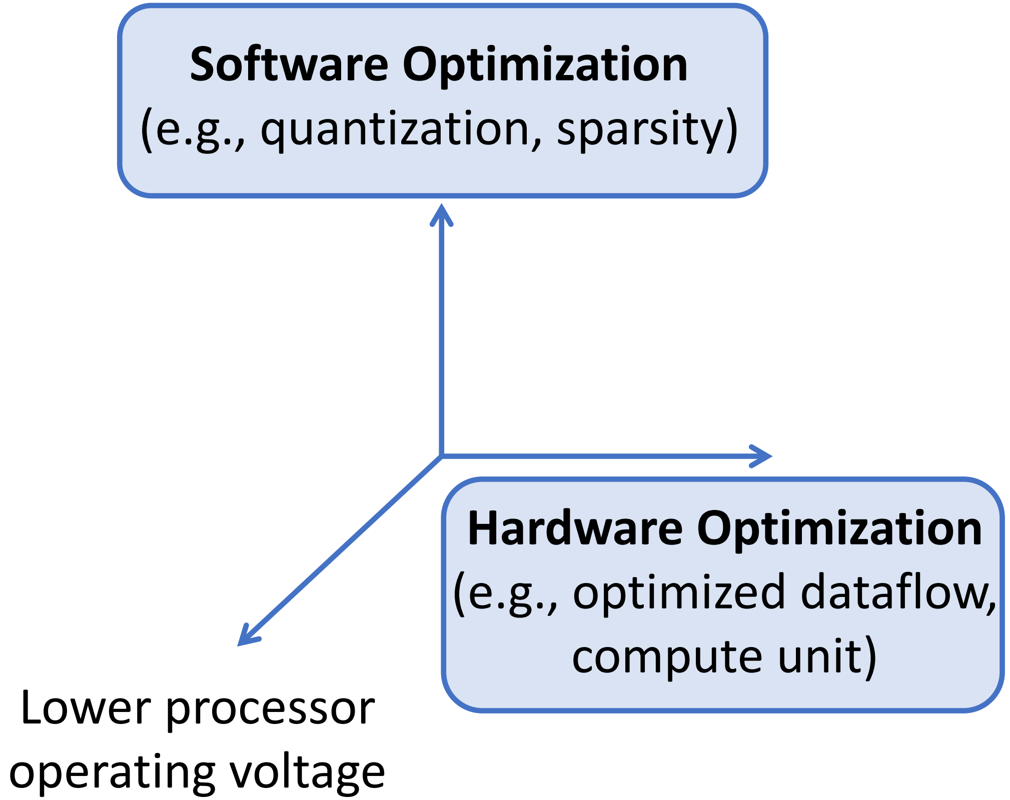
- Real-Time ***Performance***
- Energy ***Efficiency***
- Design ***Scalability***
- Safety-Critical ***Reliability***

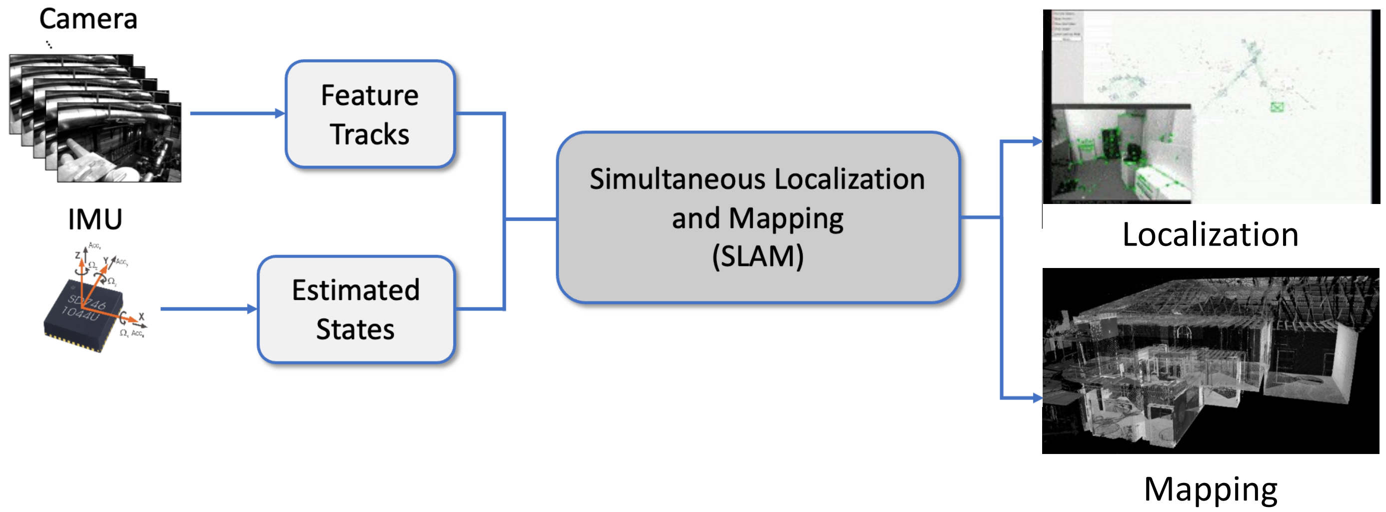
[@CICC’22]

Research Question:

How can we improve robotics autonomy’s **real-time performance** and **energy efficiency**?

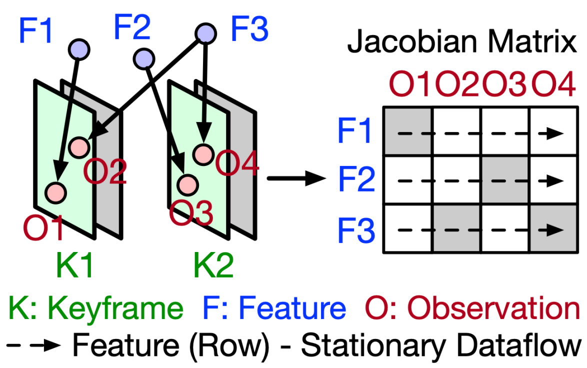
Autonomy: Localization and Mapping



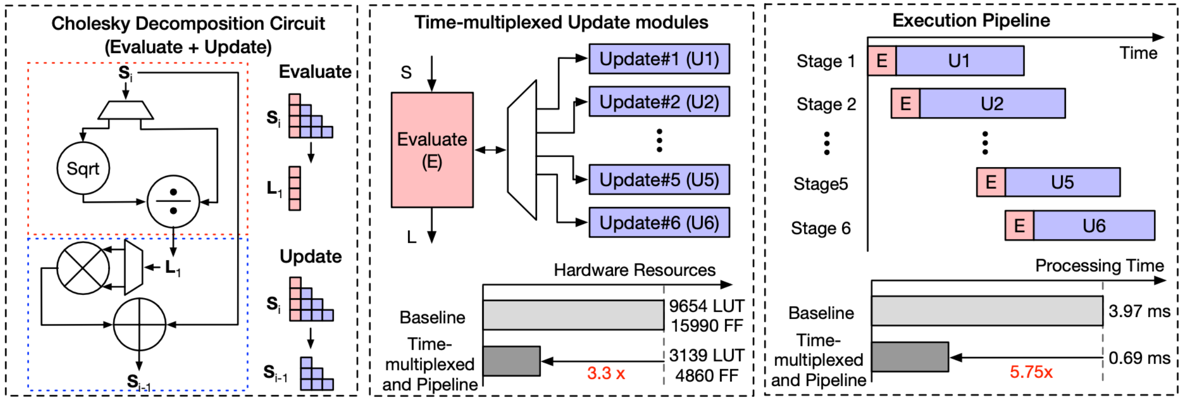


Example: Simultaneous Localization and Mapping (SLAM)

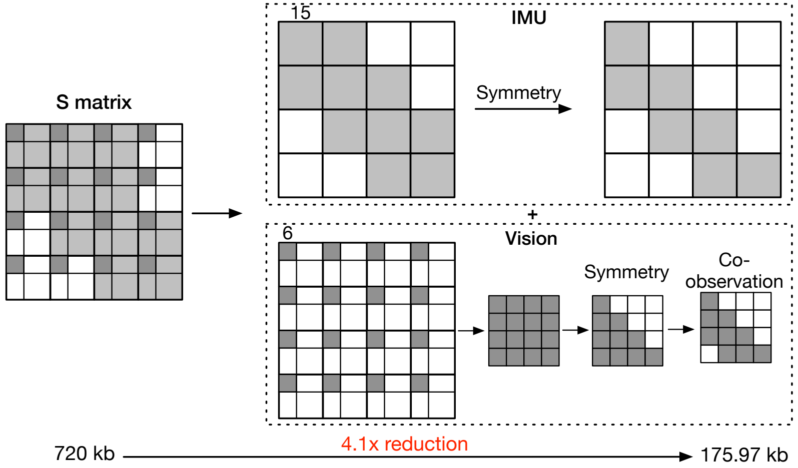
Key Domain-Specific Arch Design Techniques



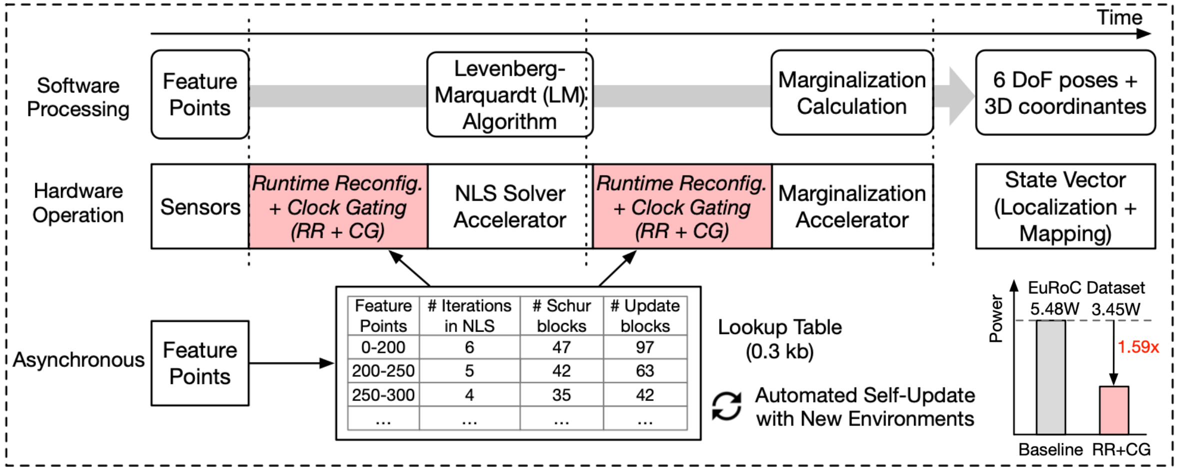
Data reuse and dataflow



Time-multiplexing and pipelining



Memory layout, symmetry, sparsity



Runtime reconfigurability and clock gating

Key Takeaways:

Domain-specific co-design and **design-technology co-optimization** unlock system performance and efficiency

Autonomous machines need **spatial-aware computing** that consider environment dynamics and heterogeneity

Low-Voltage Processing Reduces Energy

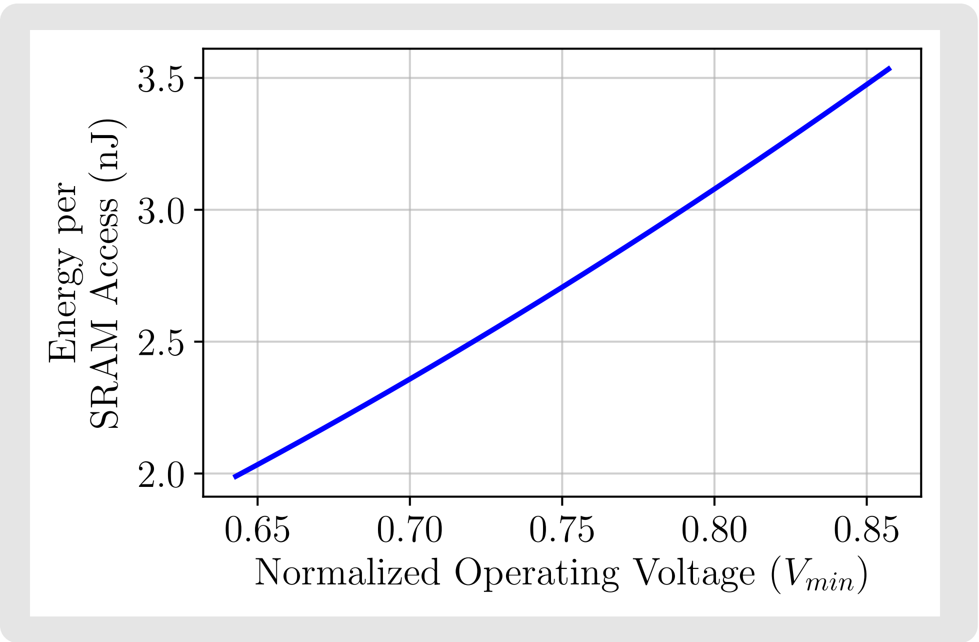
Software Optimization
(e.g., quantization, sparsity)

Hardware Optimization
(e.g., optimized dataflow, specialized compute unit)

Lower processor operating voltage

Energy ∝ Voltage ^{2}

SRAM Access Energy vs. Operating Voltage



Data measured from 14nm FinFET SRAM chips

Lower operating voltage quadratically reduces energy

Low-Voltage Processing Bring Variations

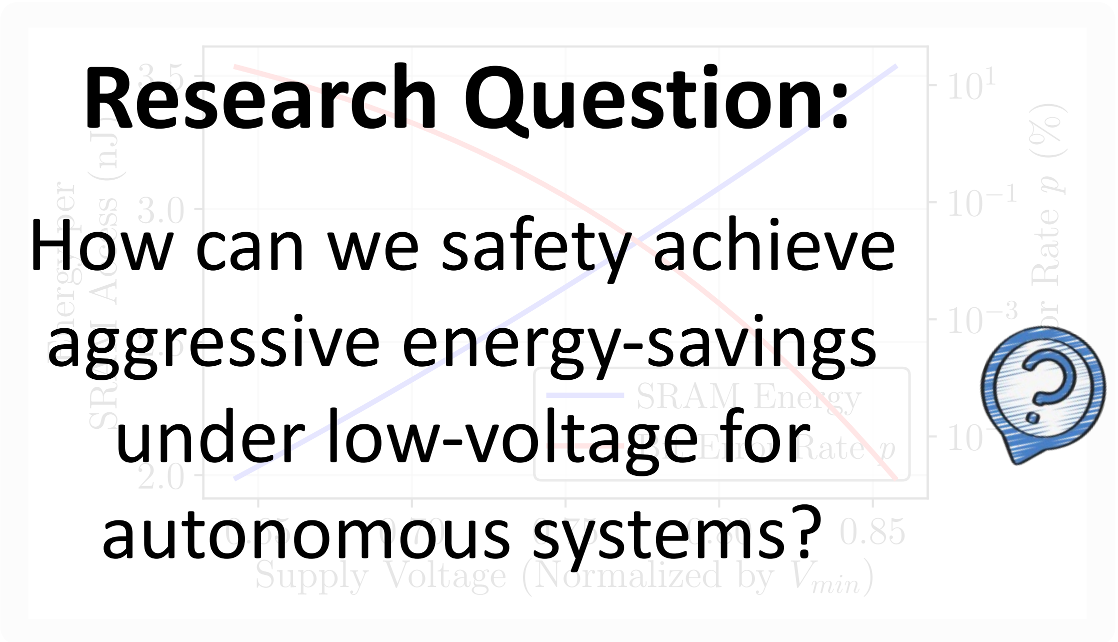
Software Optimization
(e.g., quantization, sparsity)

Hardware Optimization
(e.g., optimized dataflow, specialized compute unit)

Lower processor operating voltage

Energy ∝ Voltage ^{2}

SRAM Access Energy / Bit Error Rate vs. Operating Voltage



Data measured from 14nm FinFET SRAM chips

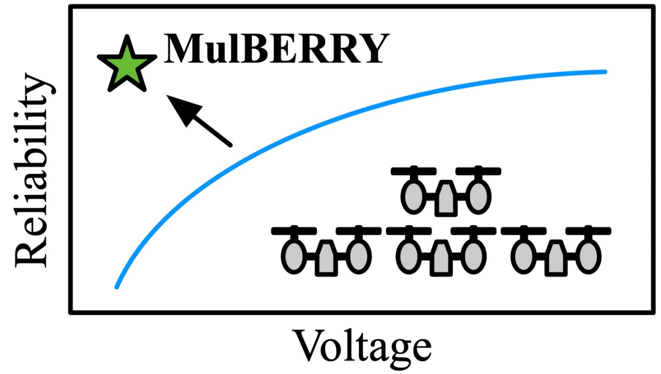
Lower operating voltage bring chip variations/errors

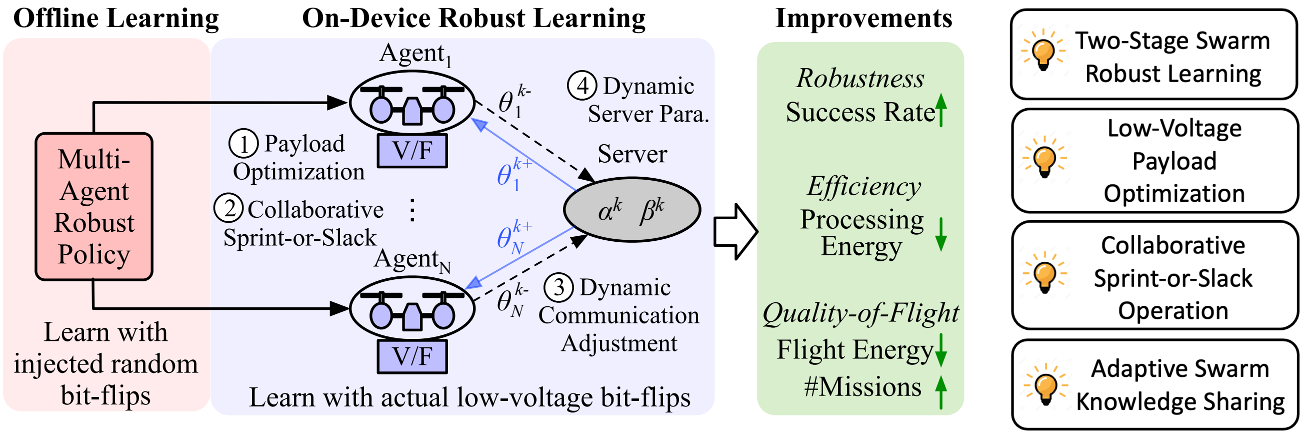
Research Question:

How can we safety achieve aggressive energy-savings under low-voltage for autonomous systems?

MulBERRY: Low-Vol Efficient Auto. Machines

- **Design Objective**: Aggressive *energy-savings* under *low-voltage operation*, yet *computationally-resilient* for swarm autonomous drone systems.
- **Design Principle**: Cross-layer swarm robust learning framework, integrates *algorithm-level* error-aware learning with *system-level* collaborative optimization and *hardware-level* thermal-voltage adaptive adjustment.

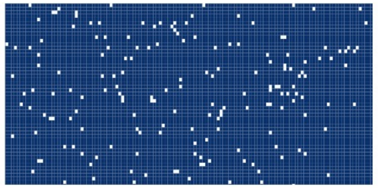


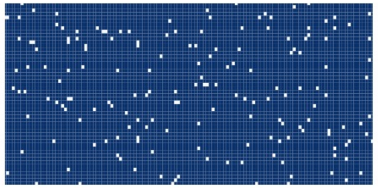


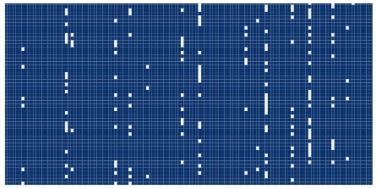
Evaluation: Efficiency Improve Across Scenarios

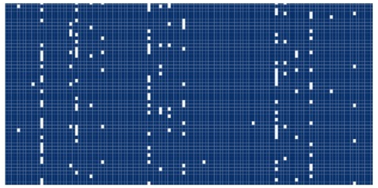

Crazyflie UAV


DJI Tello UAV


UAV 1 (Chip 1)


UAV 2 (Chip 2)


UAV 3 (Chip 3)


UAV 4 (Chip 4)


Sparse Obstacle


Medium Obstacle


Dense Obstacle

Environment	Sparse		Medium		Dense	
	Flight Energy (J)	Num. of Missions	Flight Energy (J)	Num. of Missions	Flight Energy (J)	Num. of Missions
Baseline @1V	52.41	58.56	75.80	40.15	102.4	28.04
MulBERRY (optimal)	42.02	71.63	61.42	49.01	85.77	33.79
	@0.69V_{min}		@0.70V_{min}		@0.73V_{min}	

MulBERRY is adaptive across drones, hardware chips, environments, agent numbers, tasks, and consistently improves mission efficiency

Key Takeaways:

Low-voltage operation leads to **energy savings** in both **compute** and **end-to-end mission energy**

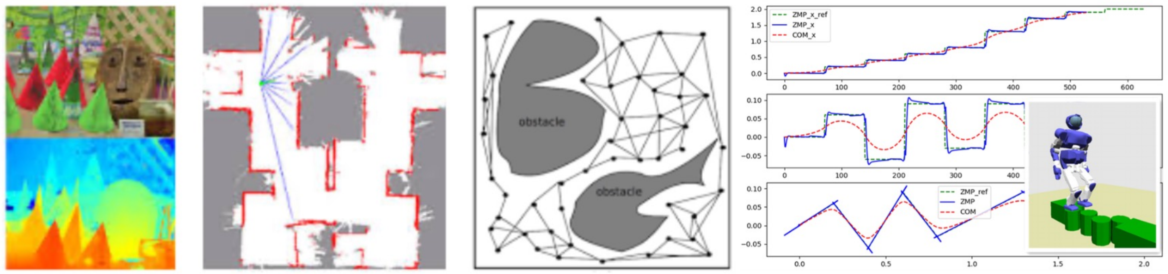
Optimizing autonomous system **cyber** components (compute) impacts its **physical** performance

Design Accelerators for Each Algorithm?

Robot Applications

sweeper | Manipulator | Vehicle | Drone

Robot Algorithms



Localization | Planning | Control

Solution 1

Programmable Accelerators

Strength: High generality

Weakness: Less effective in exploiting specific sparse structures

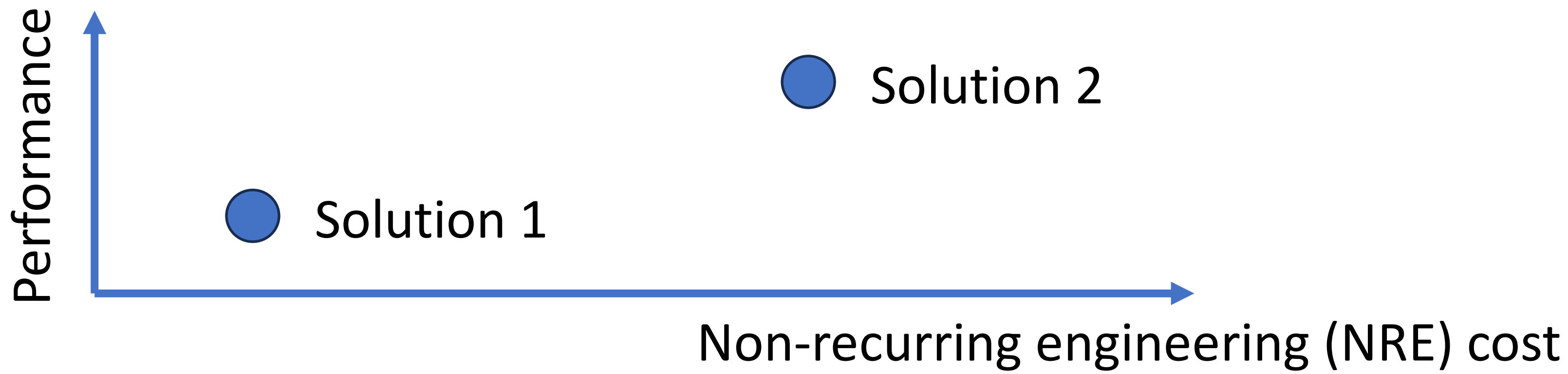
Solution 2

Dedicated Accelerators

Strength: High performance

Weakness: High NRE costs; Stacking accelerators requires large chip area

Design Accelerators for Each Algorithm?



Solution 1

Programmable Accelerators

Strength: High generality
Weakness: Less effective in exploiting specific sparse structures

Solution 2

Dedicated Accelerators

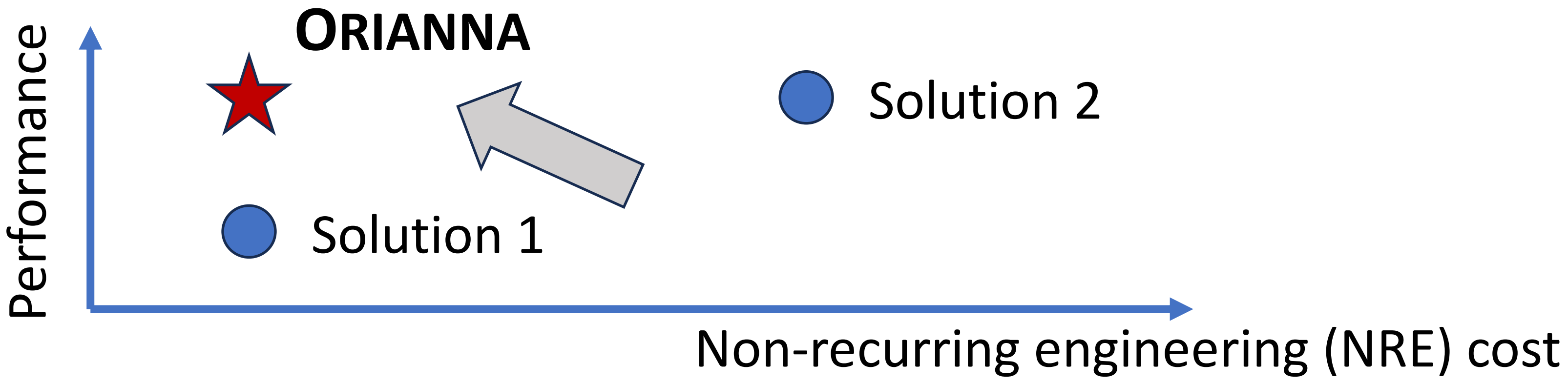
Strength: High performance
Weakness: High NRE costs; Stacking accelerators requires large chip area

[@ASPLOS’24b]

Research Question:

How can we improve robotics domain-specific accelerators **design adaptability and scalability**?

Orianna: Accelerator Generation Framework for Optimization-Based Robotic Applications



Solution 1

Programmable Accelerators

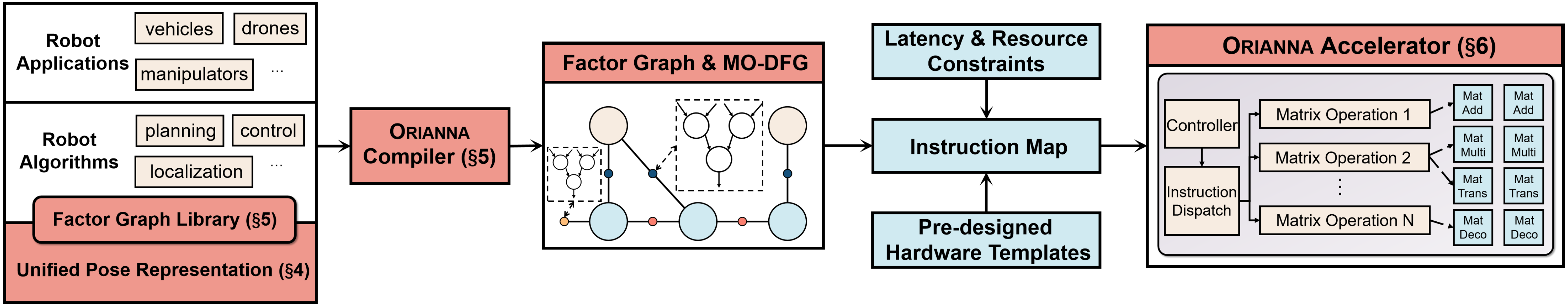
Strength: High generality
Weakness: Less effective in exploiting specific sparse structures

Solution 2

Dedicated Accelerators

Strength: High performance
Weakness: High NRE costs; Stacking accelerators requires large chip area

Orianna Framework



ORIANNA Software

Use **factor graph** as unified abstraction; Build flexible **factor graph software library**

Performance ↑

ORIANNA Compiler

Compiler transforms diverse algorithms into **unified factor graph instructions** and **data flow graph**

Generality & Scalability ↑

ORIANNA Accelerator

Different robotic algorithms can be executed **concurrently** on the **Factor Graph Accelerator**

Resource Utilization ↑

Evaluation - Benchmark

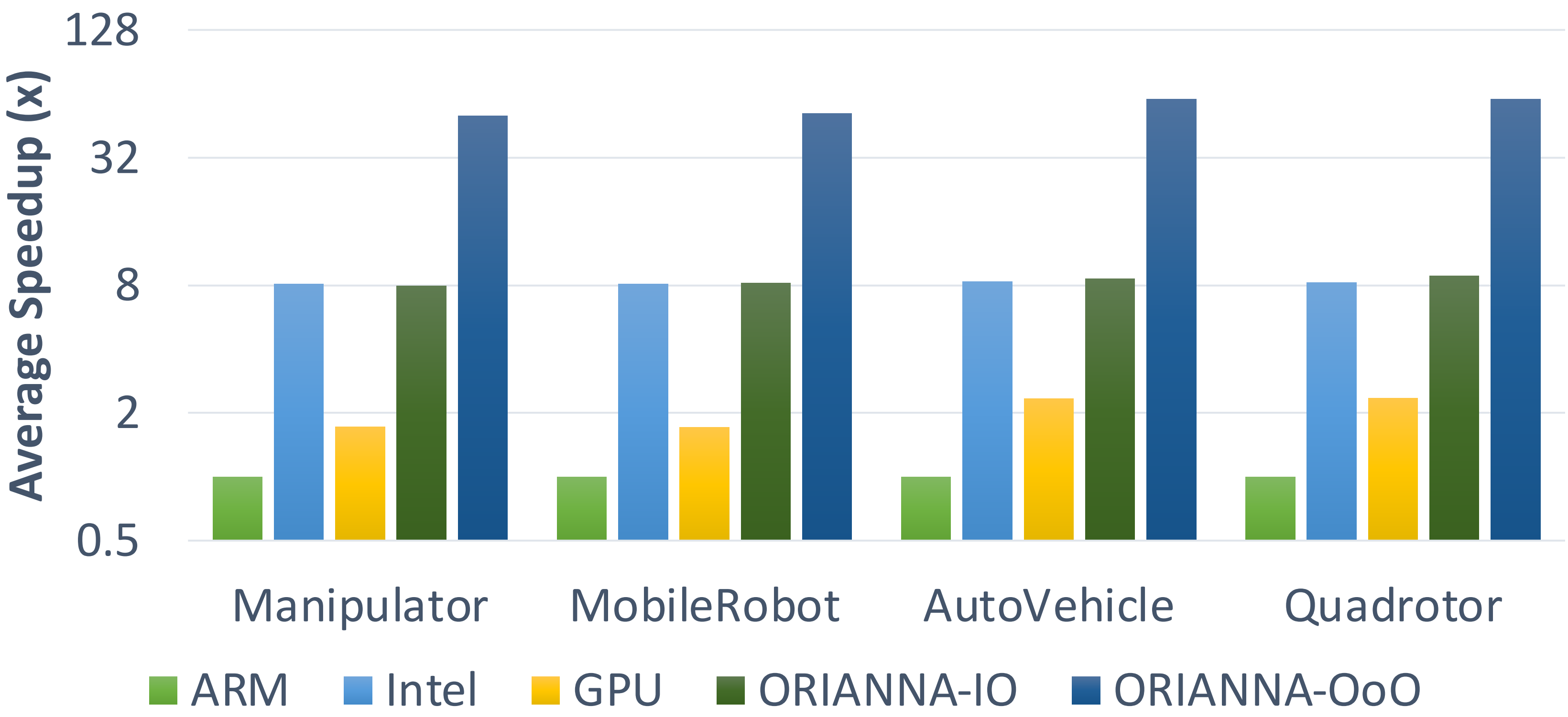
		Localization	Planning	Control
Mobile Robot	Variable dim	3	6	3, 2
	Factor	LiDAR GPS	Collision-free Smooth	Dynamics
Manipulator	Variable dim	2	4	2, 2
Auto Vehicle	Variable dim	3	6	5, 2
Quadrotor	Variable dim	6	12	12, 5

Evaluation - Setup and Baseline

Hardware Setup	Clock Frequency
Xilinx ZC706 FPGA	167 MHz

Baseline		Detailed Information	Short Title
Processors	High-end desktop CPU	16-core Intel 11th i7-11700 CPU	<u>Intel</u>
	Lower power mobile CPU	4-core ARM Cortex-A57	<u>ARM</u>
	Embedded GPU	256-core NVIDIA Maxwell GPU	<u>GPU</u>
Accelerators	Accelerator for dense matrix operations	Directly accelerates matrix operations used in optimization problems	<u>VANILLA-HLS</u>
	Accelerator utilizing factor graphs to accelerate individual algorithms	Simple integration of three accelerators	<u>STACK</u>

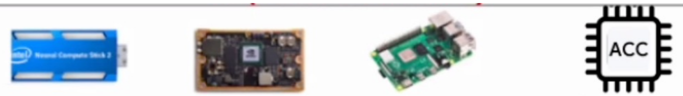
Evaluation - Performance vs Processor



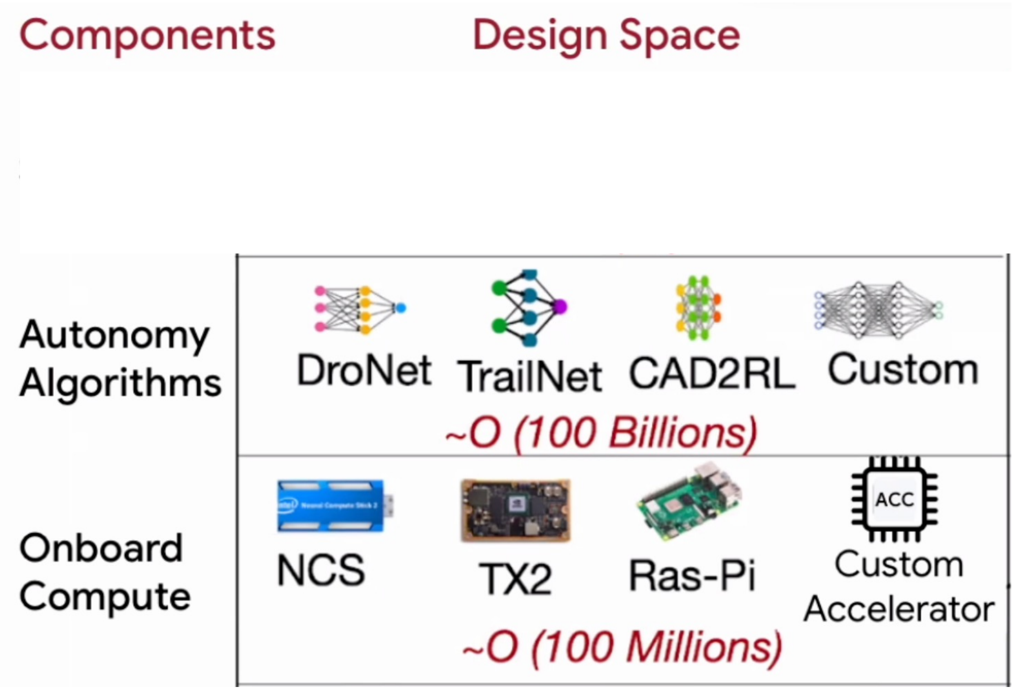
ORIANNA demonstrates a signicant speedup of 53.5× over ARM, 6.5× over Intel and 28.6× over GPU.

Large Design Space

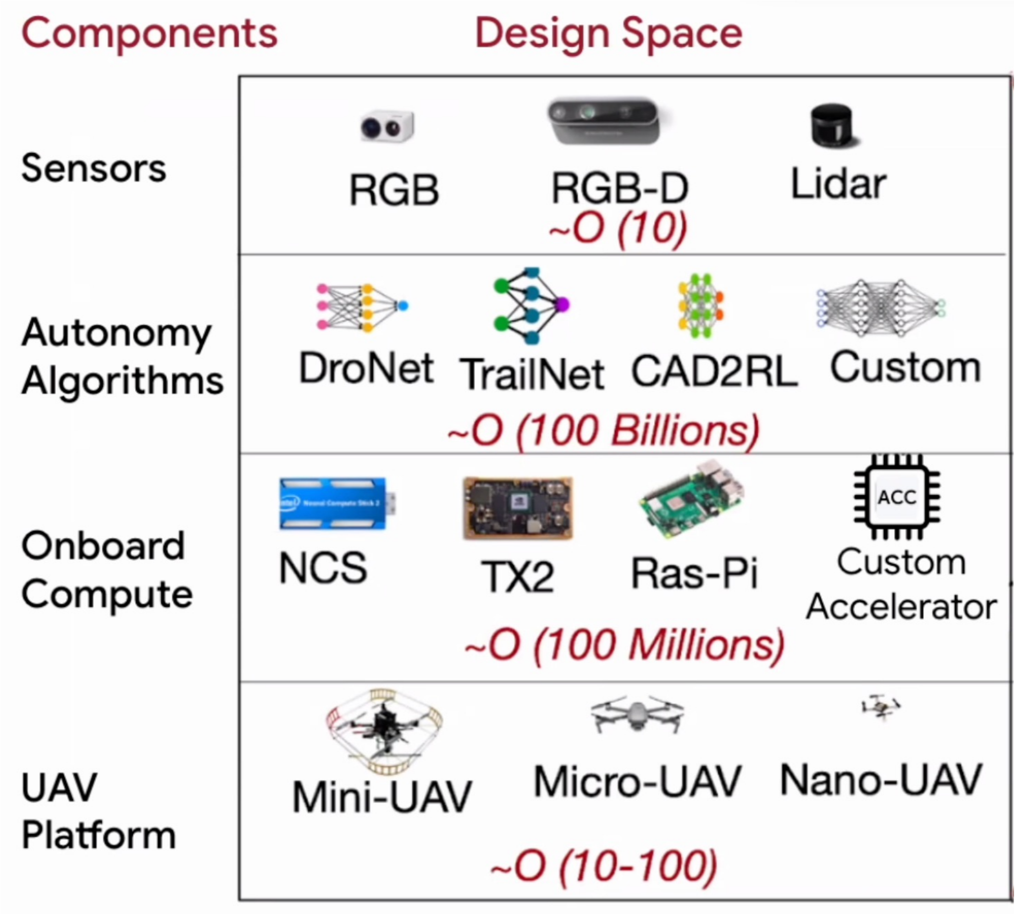
Components	Design Space
Onboard Compute	NCS, TX2, Ras-Pi, Custom Accelerator *~O (100 Millions)*



Large Design Space



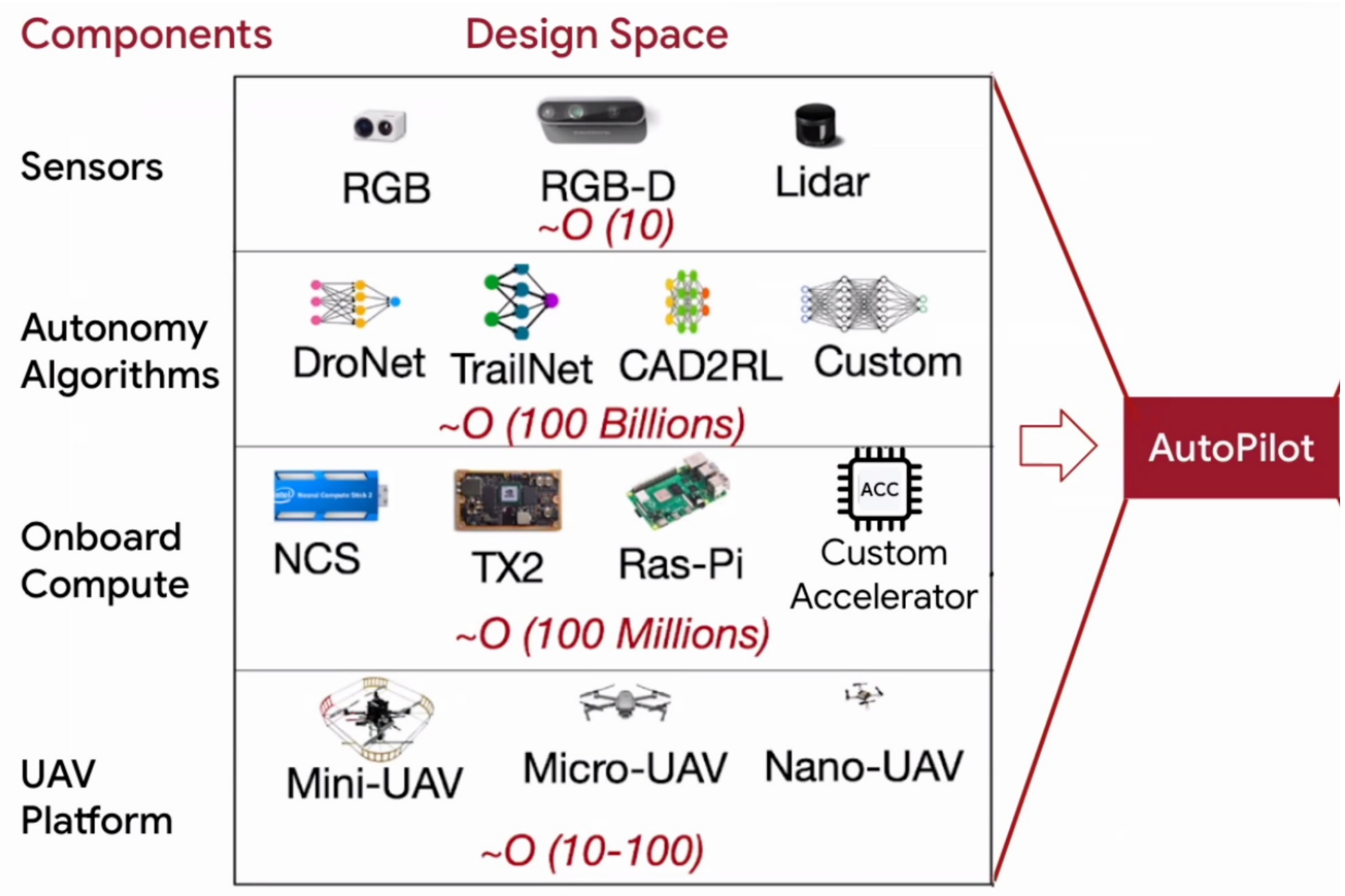
Large Design Space



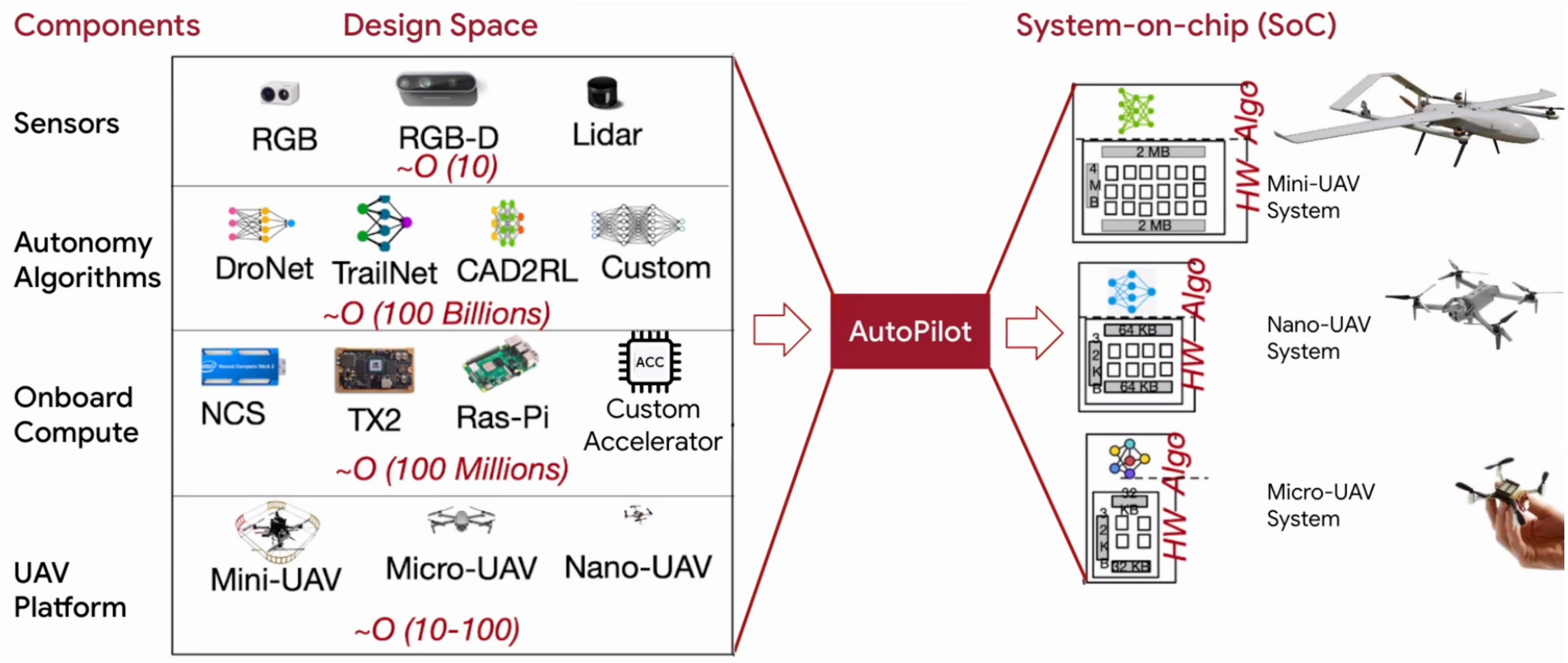
Research Question:

How can we design DSAs to handle the increasing levels of system complexity?

Our Solution: AutoPilot

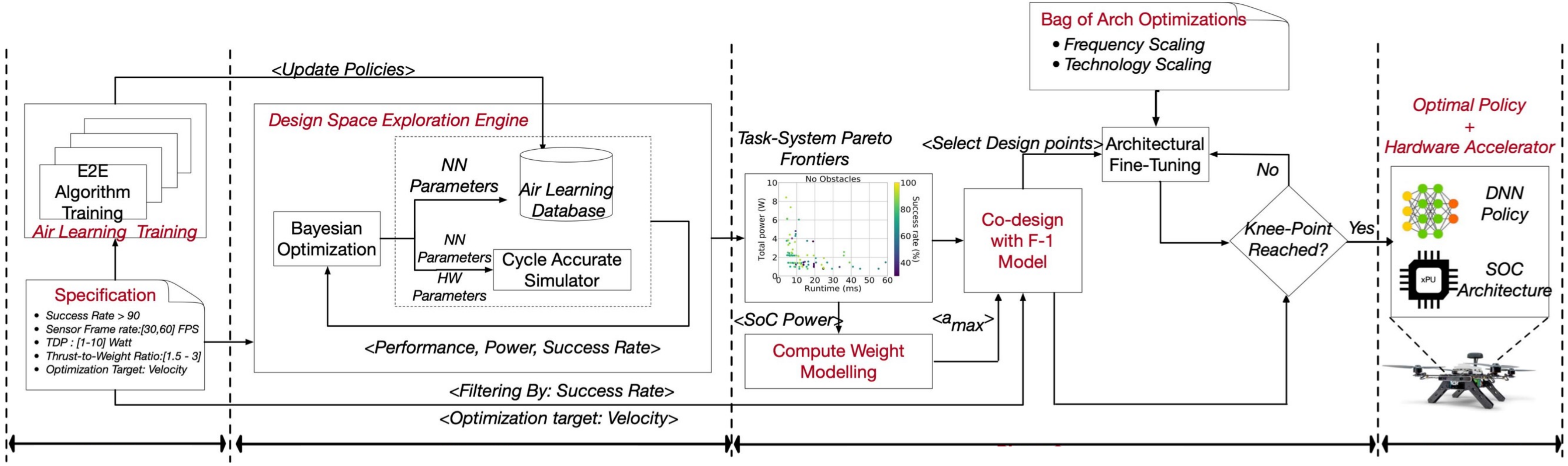


Our Solution: AutoPilot

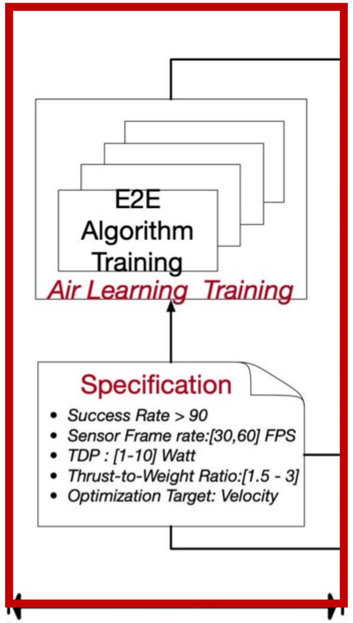


[@MICRO'23]

AutoPilot Framework



Automating SoC Design Space Exploration for Size, Weight, and Power Constrained Autonomous UAVs



Specification Phase

(Domain-specific)
Input the specification

Specification

- Success Rate >90
- Sensor Frame rate: [30, 60] FPS
- TDP: [1-10] Watt
- Thrust-to-Weight Ratio: [1.5-3]
- Optimization Target: Velocity

Sensor configuration

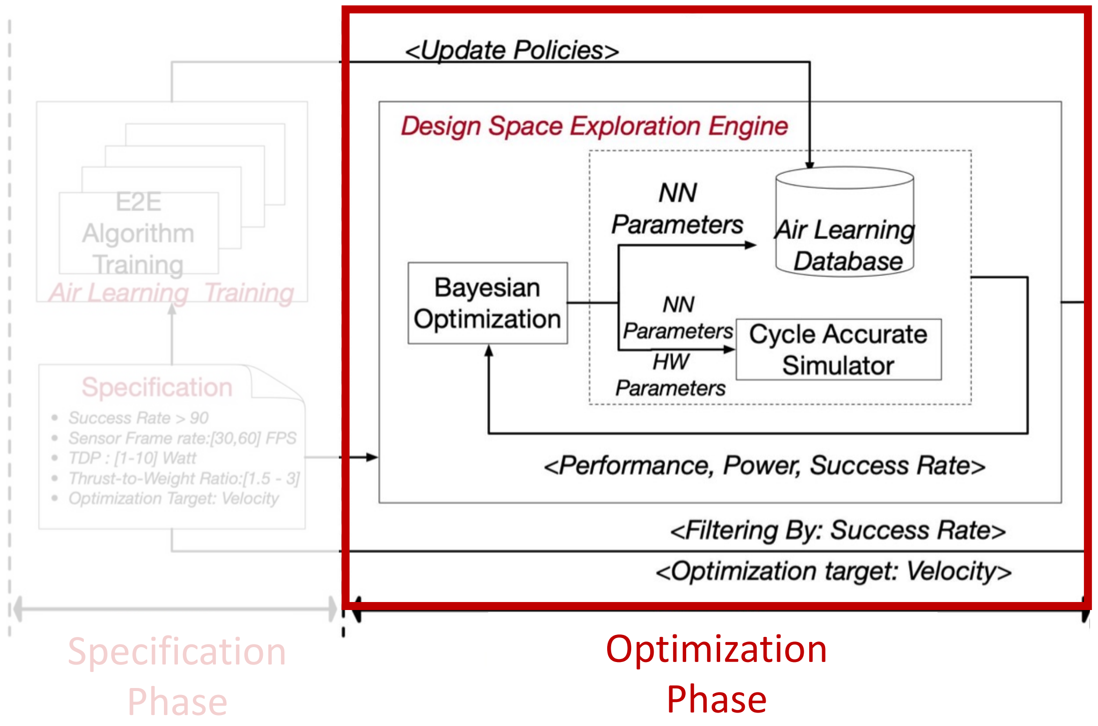
- Frame rate
- RGB
- LIDAR
- …

Compute configuration

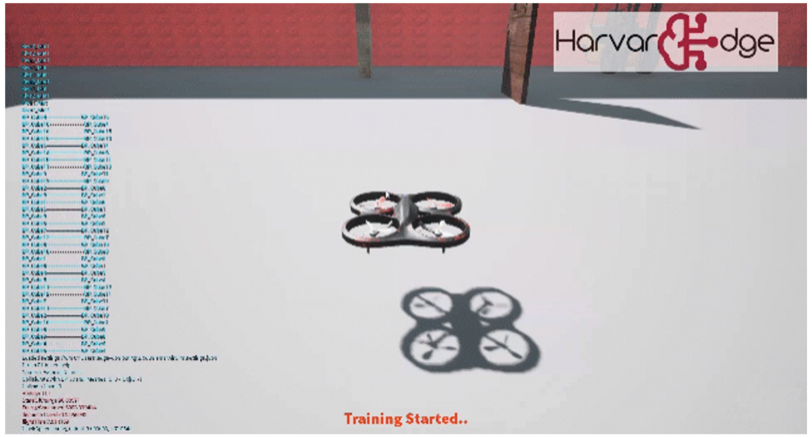
- TDP
- Latency
- Throughput
- …

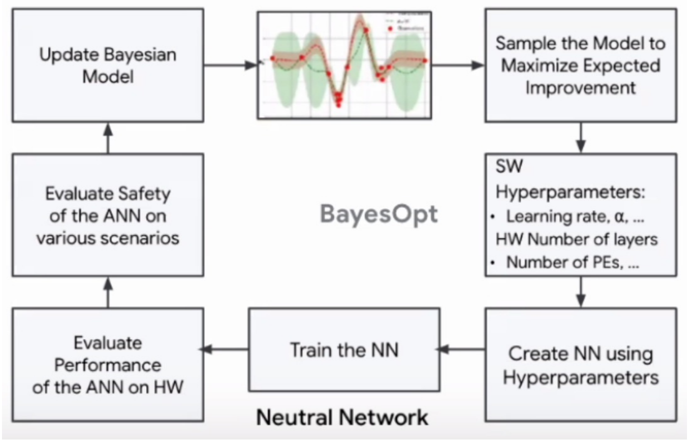
System configuration

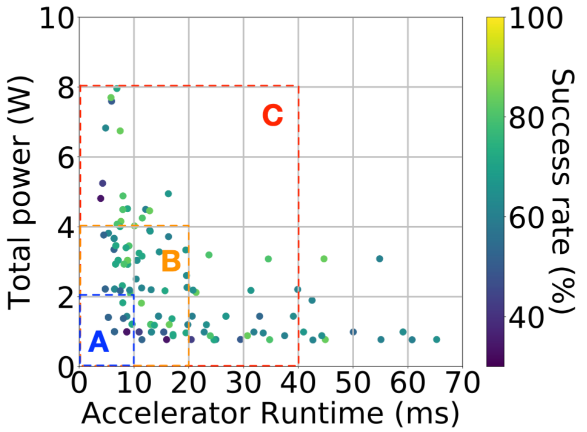
- Weight
- Thrust
- …



(Domain-agnostic) Algorithm-Hardware Co-Design Optimization



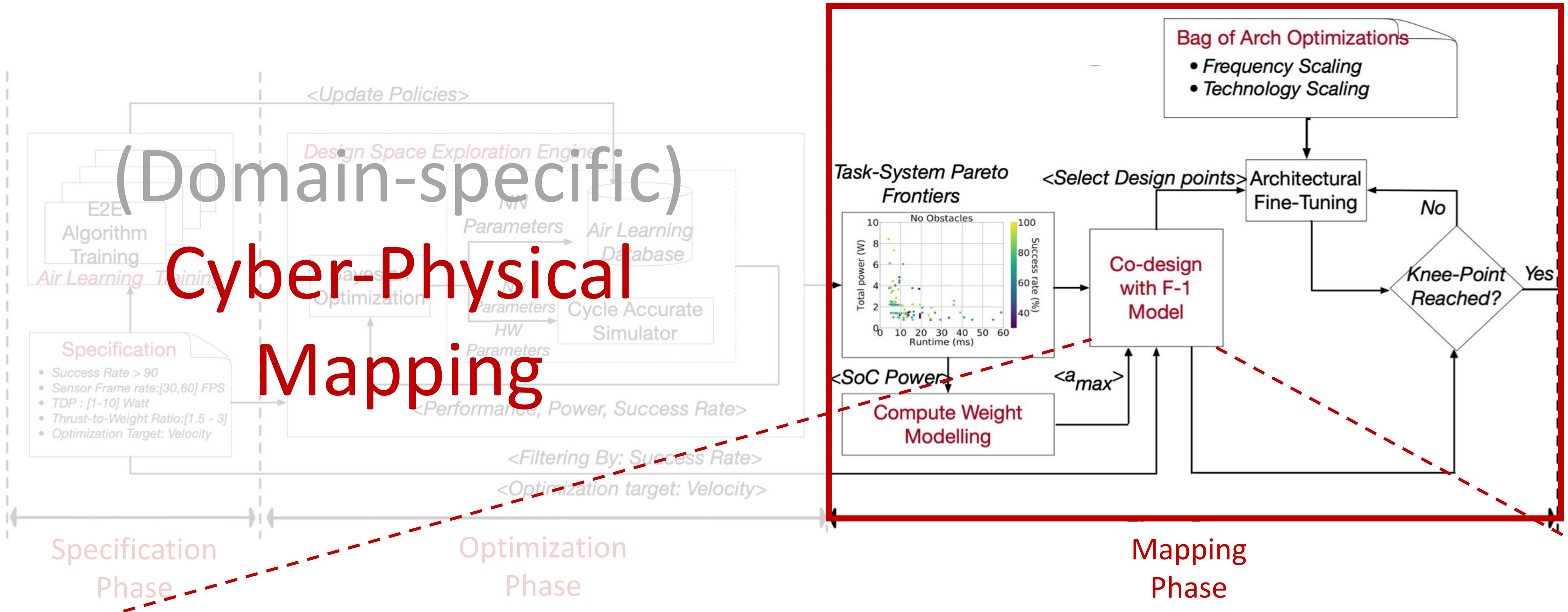


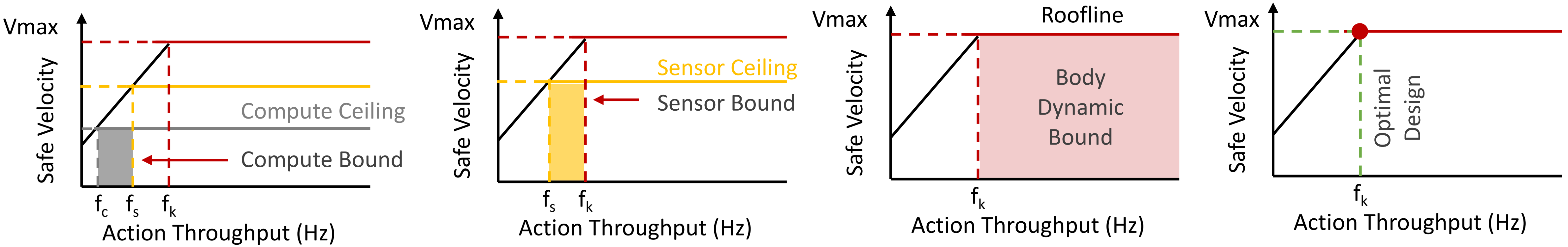


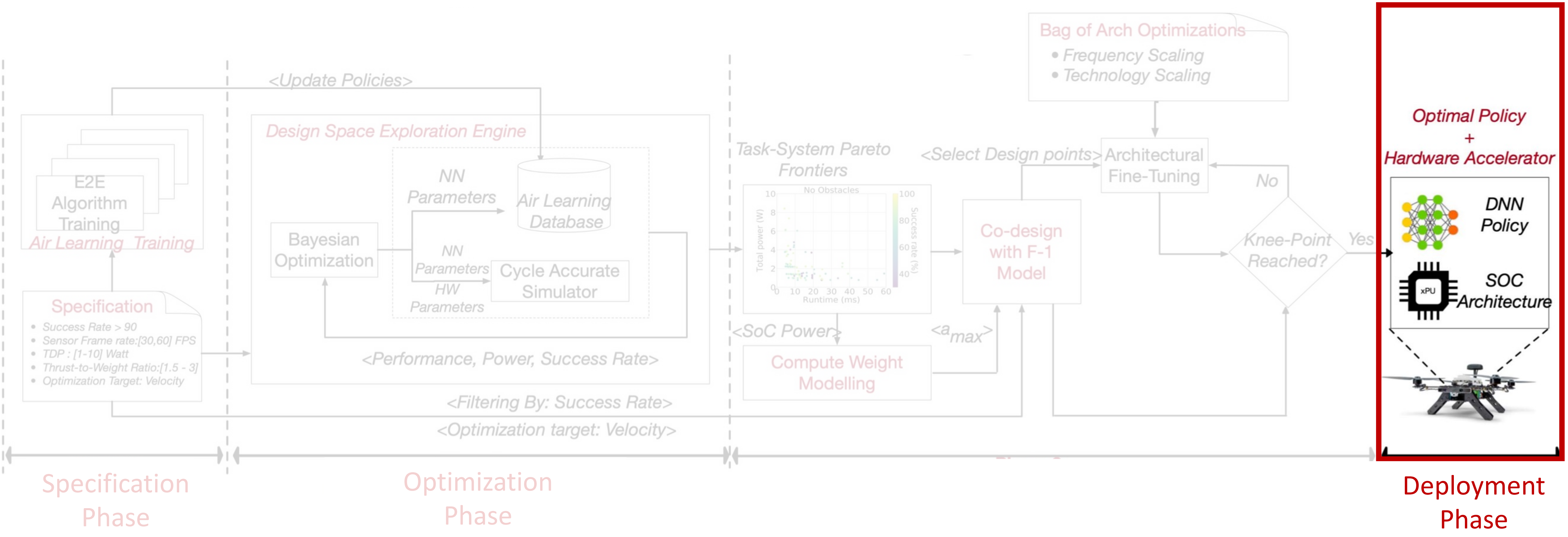
[@ISPASS'23]

[@CAL'20]

(Domain-specific) Cyber-Physical Mapping

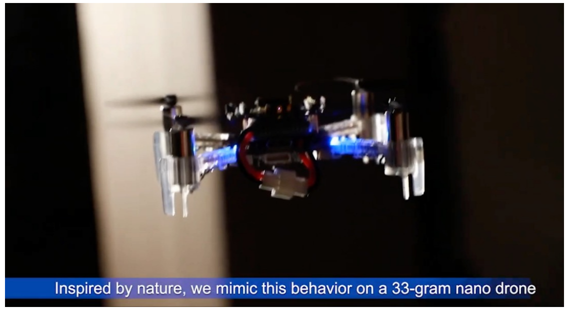






Mini UAV

Micro UAV

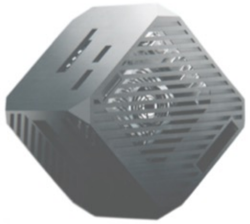


Nano UAV

RobotPerf

"If You Can't Measure It, You Can't Improve it" - Peter Drucker

- A Benchmarking Suite for Evaluating Robotics Computing Performance



(your hardware)

RobotPerf Benchmarks

Perception

Localization

Control

Navigation

Manipulation

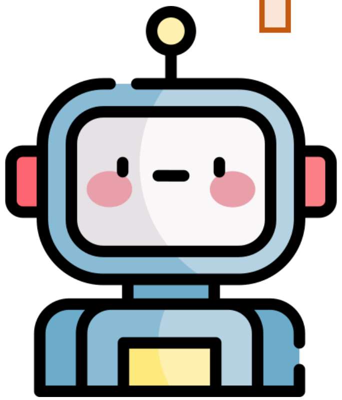
Collaborative efforts across 10+ universities & industries

Key Takeaways:

Focusing on **fragments in isolation** leads to **overlooking interdependencies and system-wide implications**

Prioritizing **system morphology** can lead to greater domain-specific architecture adaptability and efficiency

Autonomous Machines (Agentic System)



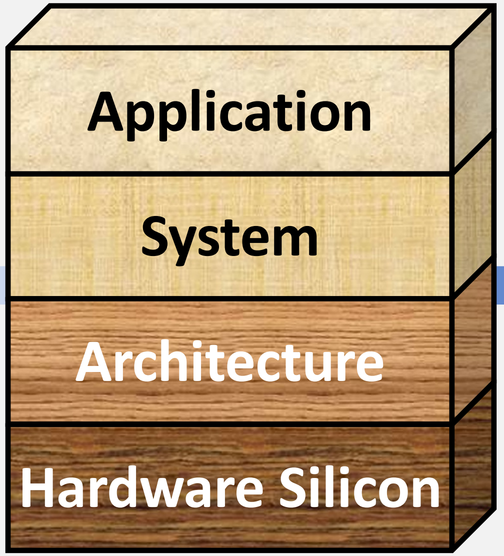
Cognition Capability

Human-like reasoning
Trustworthy decision making
Human-agent interaction

Autonomy Capability

Perception, Localization,
Mapping, Planning, Control,
Learning-based navigation

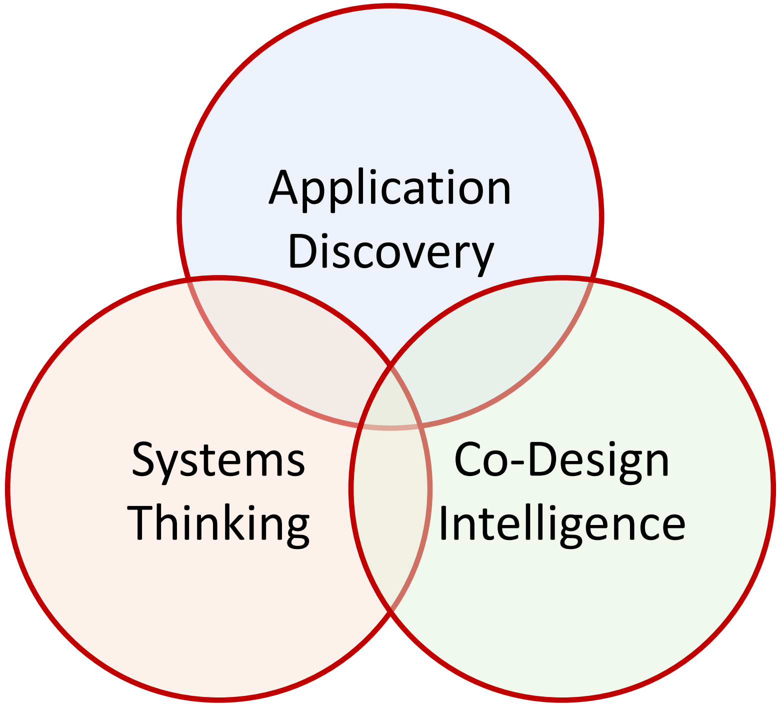
My Research:
Cross-Layer Approach



Requirements

- Real-Time ***Performance***
- Energy ***Efficiency***
- Design ***Scalability***
- Safety-Critical ***Reliability***

Summary: Core Research Methodology



01. Application Discovery: Deeply understanding an application through deep characterization to identify and address the underlying problem space.

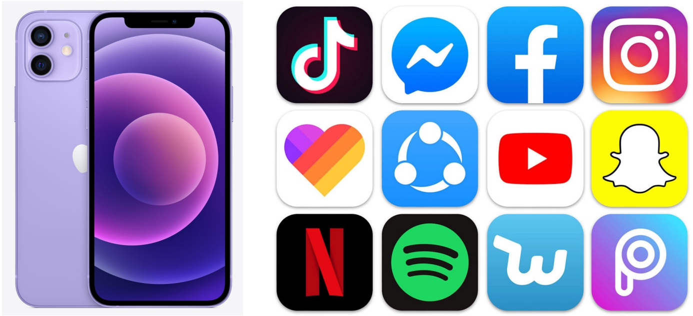
02. Systems Thinking: End-to-end design of complex systems, where every element is considered as interconnected and part of a larger, integrated whole.

03. Co-Design Intelligence: Developing software, architecture, and silicon prototype that incorporate insights from application discovery and systems thinking to automatically design solutions.

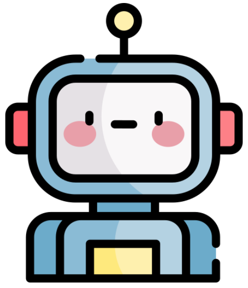
Vision for Future

90% basic functions
10% end-user applications

10% basic functions
90% end-user applications



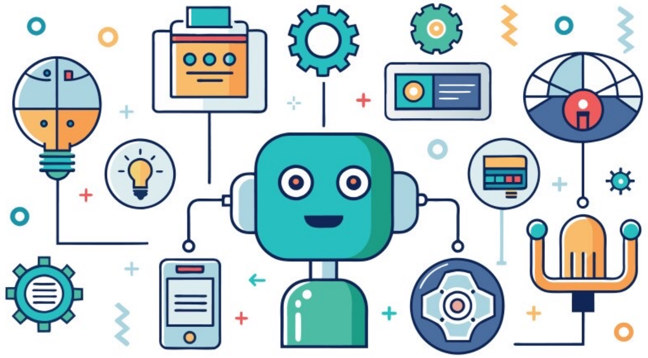
90% basic autonomy functions
10% end-user applications



Our research

software-system-hardware co-design intelligence

10% basic functions (perception/planning/control)
90% AGI (reasoning, cognition, human-AI)



Acknowledgement

- Prof. Arijit Raychowdhury & Georgia Tech ICSRL Lab
- Prof. Tushar Krishna & Georgia Tech Synergy Lab
- Prof. Vijay Janapa Reddi & Harvard Edge Computing Lab
- Dr. Karthik Swaminathan, Dr. Ananda Samajdar (IBM Research)
- Dr. Meng-Fan Chang, Dr. Win-San Khwa (TSMC Research)
- Dr. Katie Zhao (UMN), Dr. Yiming Gan (Rochester), Dr. Shaoshan (PerceptIn)
- JUMP 2.0 CoCoSys Center, JUMP 1.0 CBRIC Center

Tailored Computing: Domain-Specific Architectures for Embodied Autonomous Machines

Zishen Wan

PhD Student @ School of ECE, Georgia Institute of Technology

Web: https://zishenwan.github.io
Email: zishenwan@gatech.edu

Coordinated Science Laboratory @ University of Illinois Urbana-Champaign (UIUC), Feb. 25, 2025