Tinoosh Mohsenin

E2EdgeAI: Energy Efficient Edge AI for On-device Deployment

Associate Professor, Electrical and Computer Engineering

Secondary appointment in Computer Science

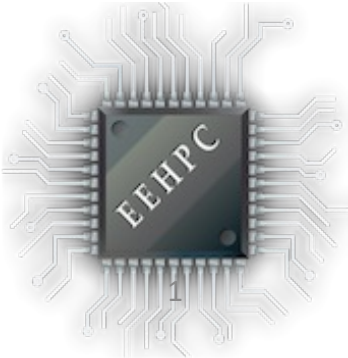
Director of Energy Efficient High Performance Computing Lab (EEHPC)

Institute of Assured Autonomy

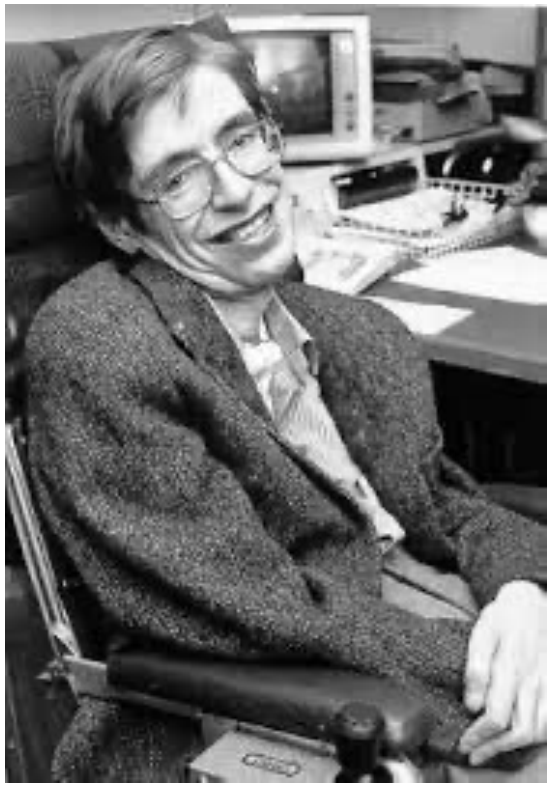
Whiting School of Engineering

EEHPC Lab: https://eehpc.ece.jhu.edu/

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING



AI: Evolution or Revolution?



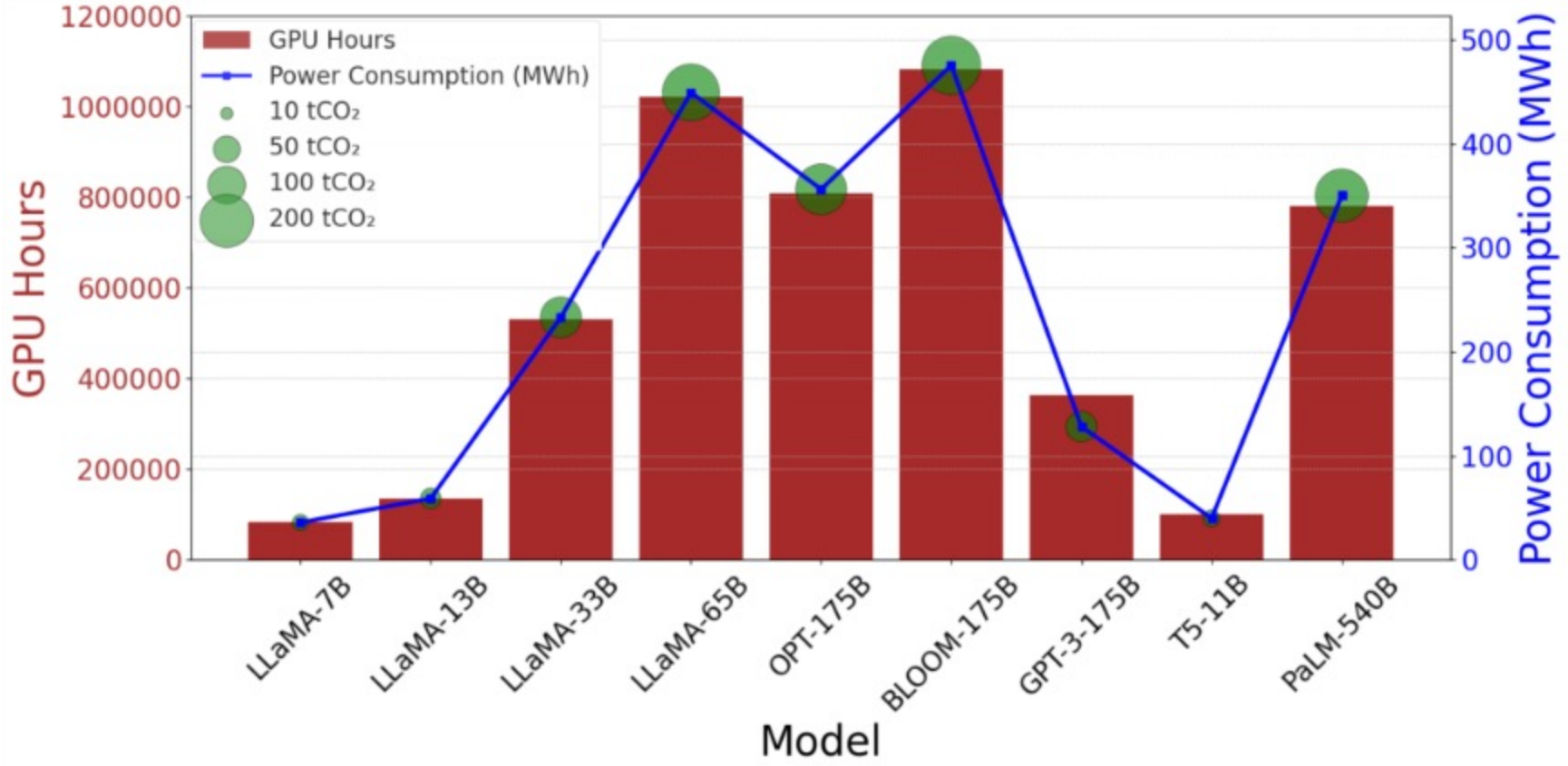
“Success in creating effective AI could be the biggest event in the history of our civilization. Or the worst. So we cannot know if we will be infinitely helped by AI or ignored by it and sidelined, or conceivably destroyed by it.” By Stephen Hawking

Some Challenges and Dark side of AI:

- Environmental, Carbon Footprint and Sustainability
- Energy Inefficiency
- Privacy Issues
- AI Biasness
- Ethics issues

AI Carbon Footprint and Energy Consumption

- The carbon footprint of training LLama 3 with 65 billion parameters can be as high as **173 tonnes of carbon dioxide equivalent (tCO_2).**
- 40 times the lifetime emissions of an average car (~100,000 miles).
- 300 times the emissions of a single flight between New York and San Francisco per passenger.
- For reference:
  - Car emissions: Roughly 1 tCO_2 for every 2,500 miles driven.
  - Airplane emissions: Roughly 0.5 tCO_2 per medium-haul flight (~2,000 miles) per passenger.



Bar plot source: Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." *arXiv preprint arXiv:2302.13971* (2023), Meta AI

For the power consumption of a A100-80GB, we take the thermal design power for NVLink systems, that is 400W. We take a PUE of 1.1 and a carbon intensity factor set at the national US average of 0.385 kg CO2e per KWh.

Challenges: Large Scale and Compute Intensive LLM

A bottle of water per email: the hidden environmental costs of using AI chatbots

A 100-word email generated by an **AI chatbot using** GPT-4 once requires 140 Watt-hours (Wh) of electricity, equal to powering **14 LED light bulbs for 1 hour**.

Washington Post,
September 18, 2024

- It is Crucial to address these questions:
  - **How to improve** the energy efficiency of such models
  - **What are Novel** Algorithms, Circuits and Architectures?
  - **Where** to run these models for real time performance?

Achieving Energy Efficiency Challenges

- Rapidly increasing volume and types of data as well as compute intensive algorithms and large models to attain required accuracy
  - Generative Model sizes from **tens of billions** to p **hundreds of billions** of parameters
  - 41.6 billion connected IoT devices, generating 79.4 zettabytes (ZB) of data
- Memory communication is costly
  - off-chip DRAM costs 3 orders of magnitude more than integer addition
  - off-chip DRAM communication costs nearly 10 to 100 times more than on-ship SRAM communication.

ENERGY TABLE FOR 45NM CMOS PROCESS [1].

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit 32KB SRAM	5	50
32 bit DRAM	**640**	**6400**

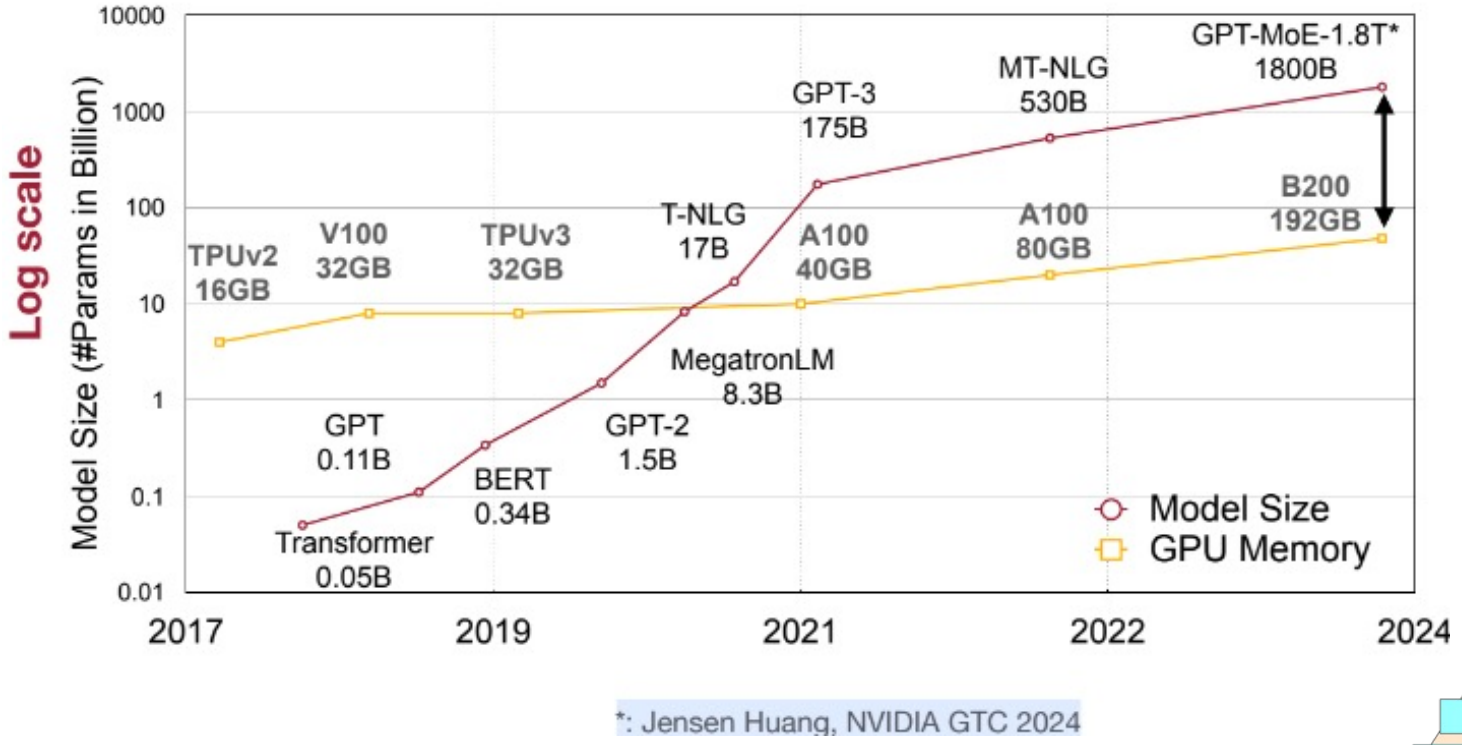
Mark Horowitz, Energy table for 45nm process, Stanford VLSI wiki

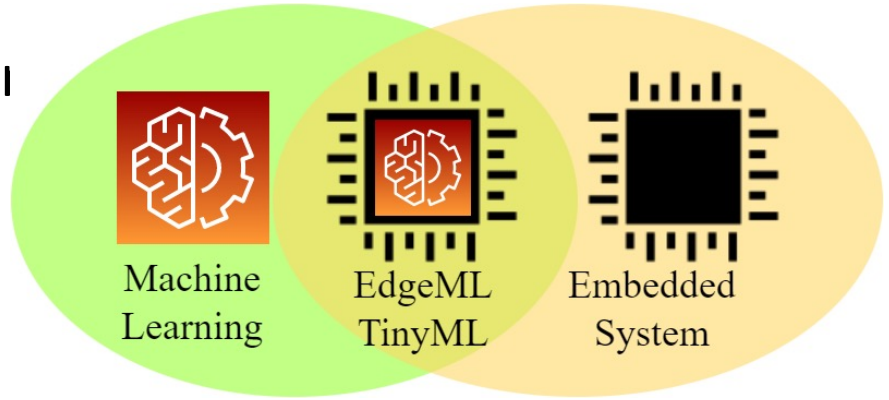
- Energy consumption is proportional to the memory communication and number of operations
- If we bring the computation to on-chip, the challenge is how to fit large model size within on-chip memory and meet the low energy requirements?

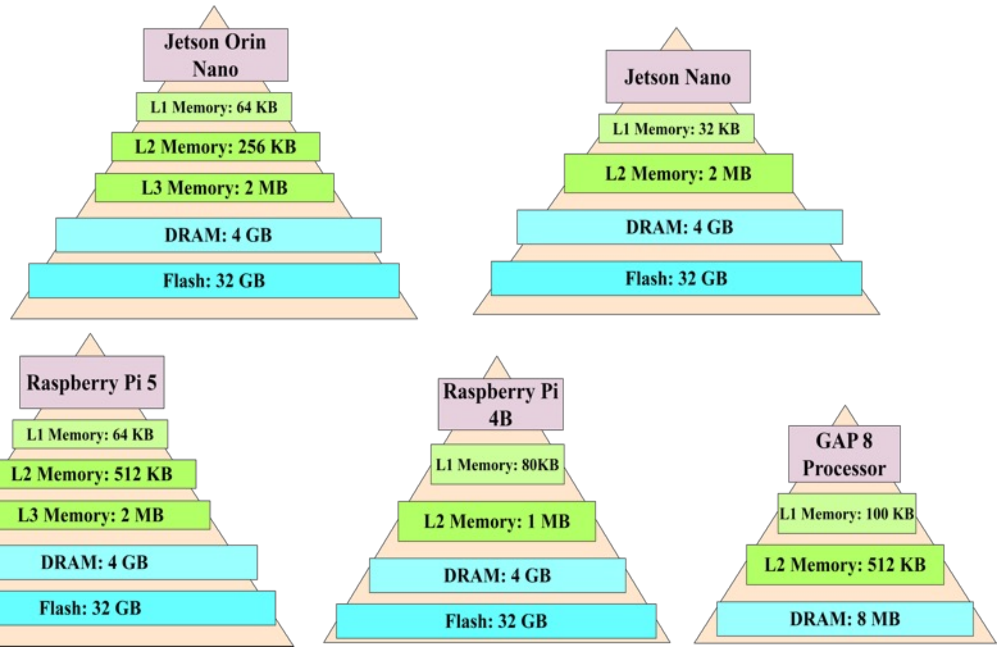
Reducing both model size and #operations can bring down the energy consumption through novel sensing, algorithmic and computing approaches

Challenges: LLM Inference deployment on edge devices

- High parameter count limits LLM use on edge devices.
- Opportunity exists to compress LLM and Transformer models for edge applications.



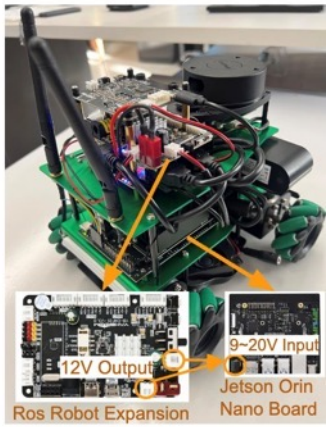


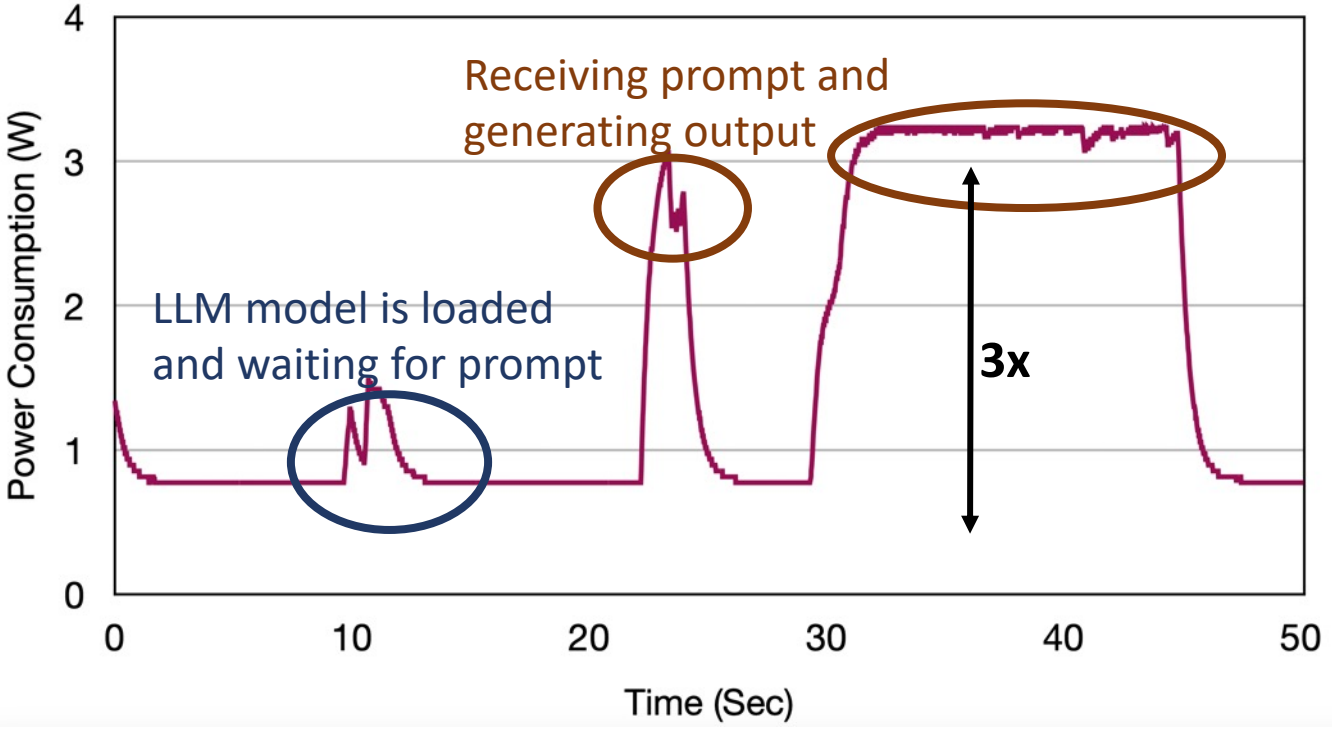


[1] Plot Source AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration, MLsys 2024, https://hanlab.mit.edu/projects/awq

Challenges: Power Consumption of LLM on Edge Devices

- Tiny LLMs on edge devices consume a significant amount of power
- Running LLM (Phi2-4bit) on edge devices such as NVIDIA Jetson Orin leads to up to 3x more power consumption comparing to Idle power
- Reducing both model size and number operations can bring down the power consumption and latency through novel algorithmic and computing approaches





Configuration	Power (W)	Latency (Sec)	Model Size
IDLE	1.1	-	-
YOLO Object detection model	1.8	0.1	Up to 168 MB
Onboard LLM Phi2-4bit	3.3	15 **(150x)**	**1.6 GB**

[1] Plot and Experiments by EEHPC JHU lab, Mozhgan Navardi and Miki Walczak

Background LLMs and Transformers

- LLMs are transformers specifically trained for language modeling and generation tasks.
- LLMs can use Encoder-Decoder or Decoder-only architectures based on the task requirements.
  - Encoder: Analyzes and encodes the input to understand context.
  - Decoder: Generates responses based on the encoded context or previously generated tokens.
- Attention Blocks: Both encoder and decoder contain attention mechanisms for capturing relationships between tokens.
- Some attention-based techniques in Vision Transformers (ViTs) can be adapted for LLMs due to the shared use of attention blocks.

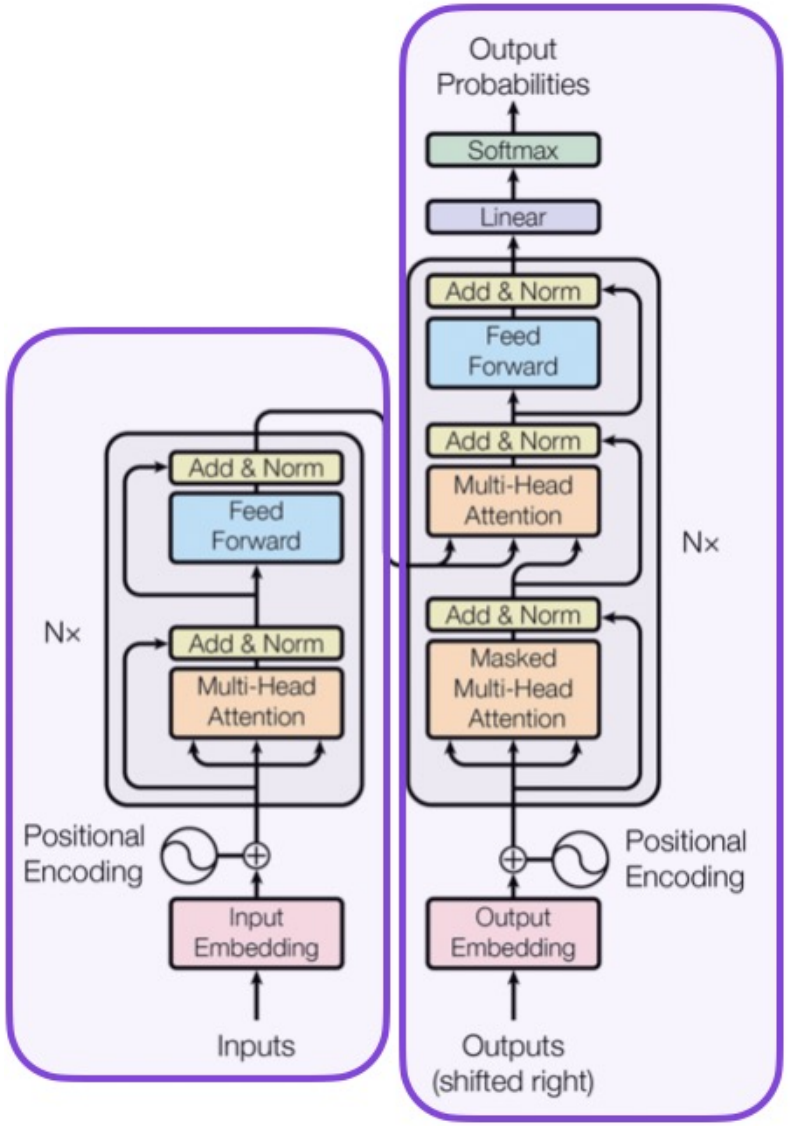


Figure 1: The Transformer - model architecture.

Background: Multihead attention

- Given a set of queries *Q*, keys *K*, and values *V*, the self-attention mechanism can be defined as

- Attention (*Q, K, V*) = softmax $\left(\frac{QK^T}{\sqrt{d_k}}\right)V$ where,
  - $Q = X*W_Q$
  - $K = X*W_K$
  - $V = X * W_V$

- *X* is the input and W_Q, W_K, W_V are the dense weight matrices for the Queries, Keys and Values respectively. d_k is the dimension of the Keys.

- For example Llama 2-7B: Summing up each component:
  - Embedding Layer: ~131 million
  - Multi-Head Attention: ~1.6 billion
  - Feed-Forward Layers: ~4.3 billion
- Although Feed-forward layers generally have more parameters, but attention layers dominate in terms of computational cost.

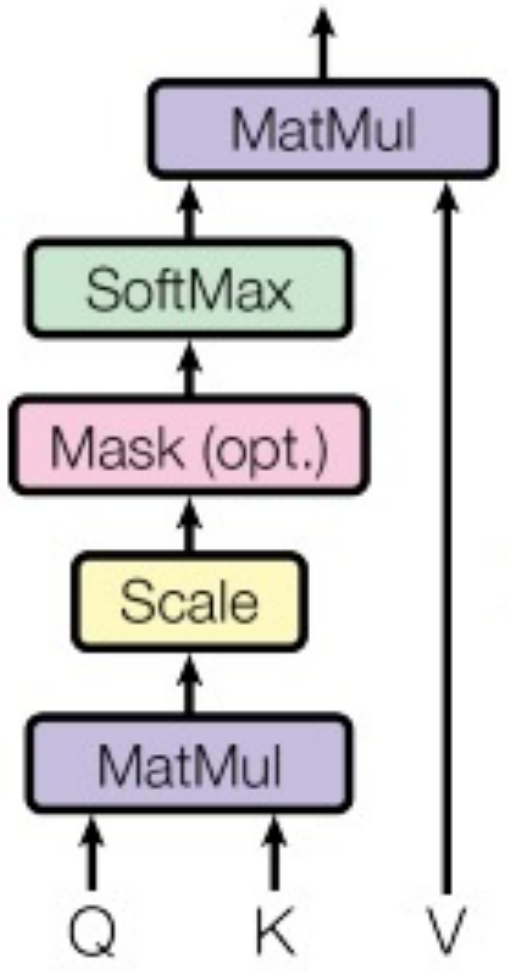
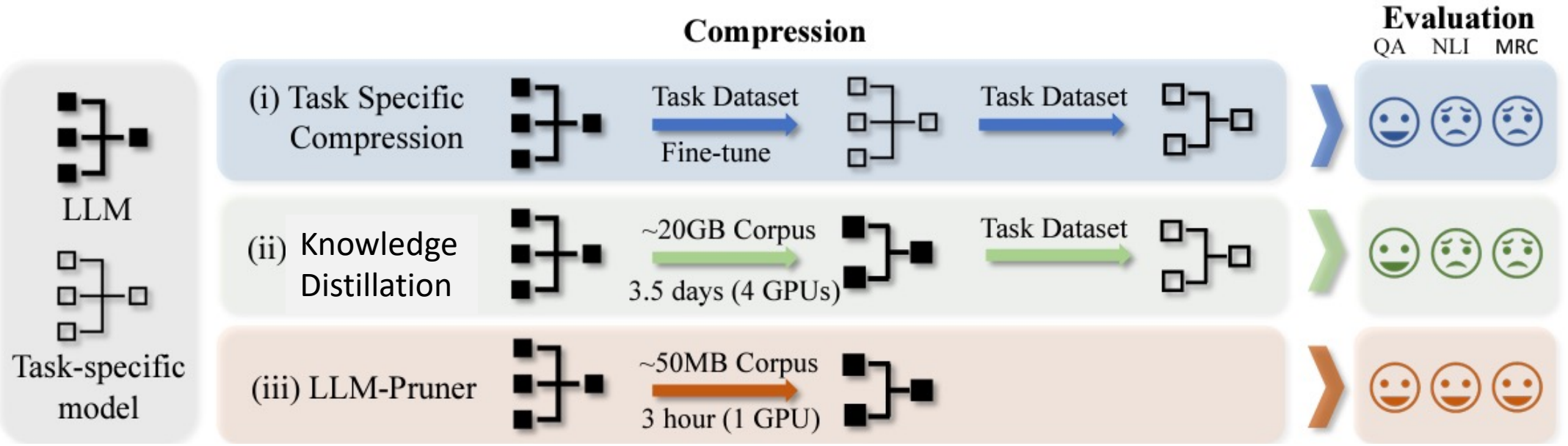


Image source: Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Opportunities for Compressed LLM Models



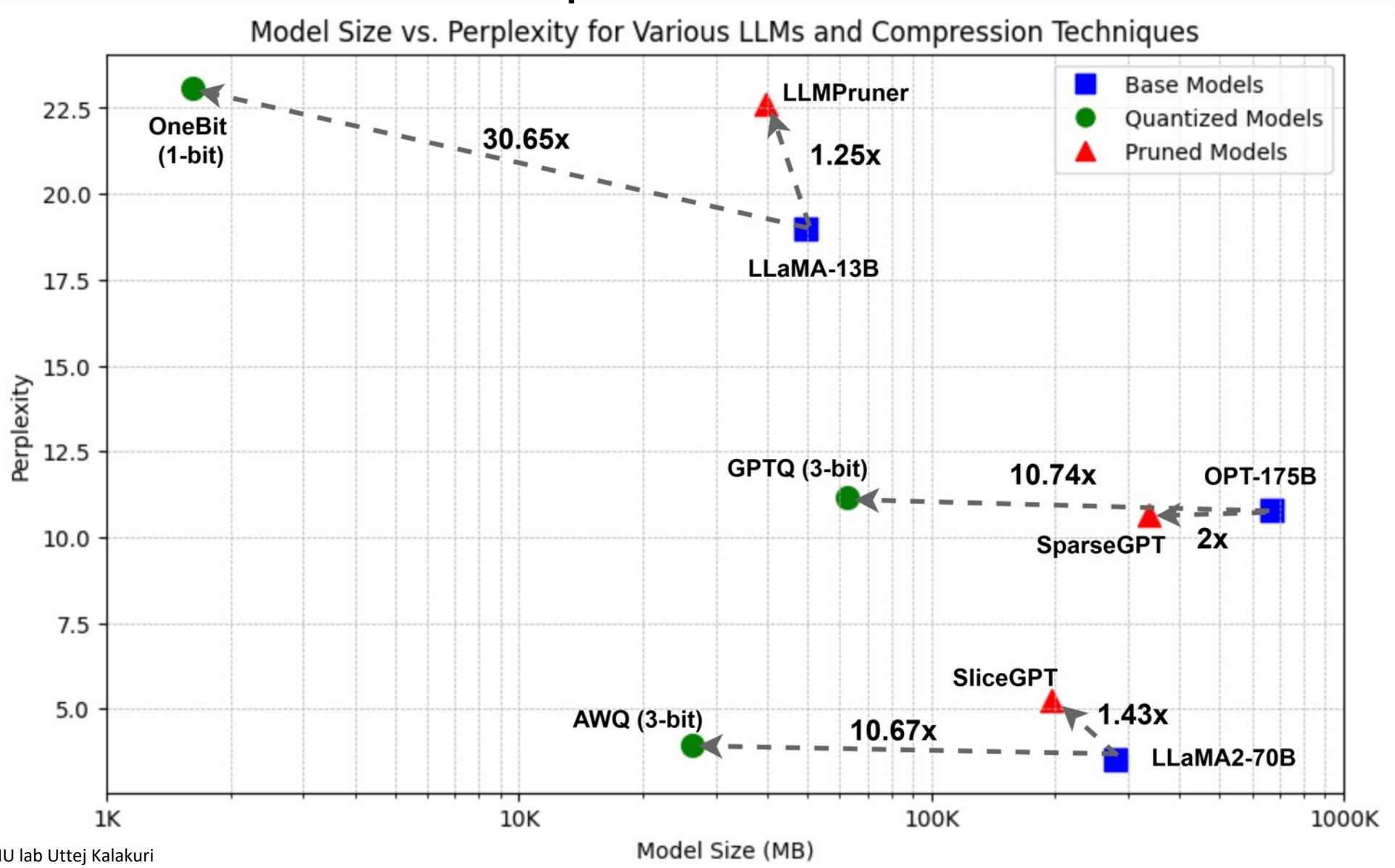
QA: Question Answering
MRC: Machine Reading Comprehension
NLI: Natural Language Inference

Pruning & Sparsification	Ranking neurons according to contribute, and removing low ranking neurons
Knowledge Distillation	Compress large teacher models into compact student models

Quantization	Reducing Bit-Precision
Architecture Search	Regression-Search hyper-parameters for efficient model design

Top figure modified source: Ma, Xinyin, Gongfan Fang, and Xinchao Wang. "Llm-pruner: On the structural pruning of large language models." *Advances in neural information processing systems* 36 (2023): 21702-21720.

State of the art Compressed LLM models

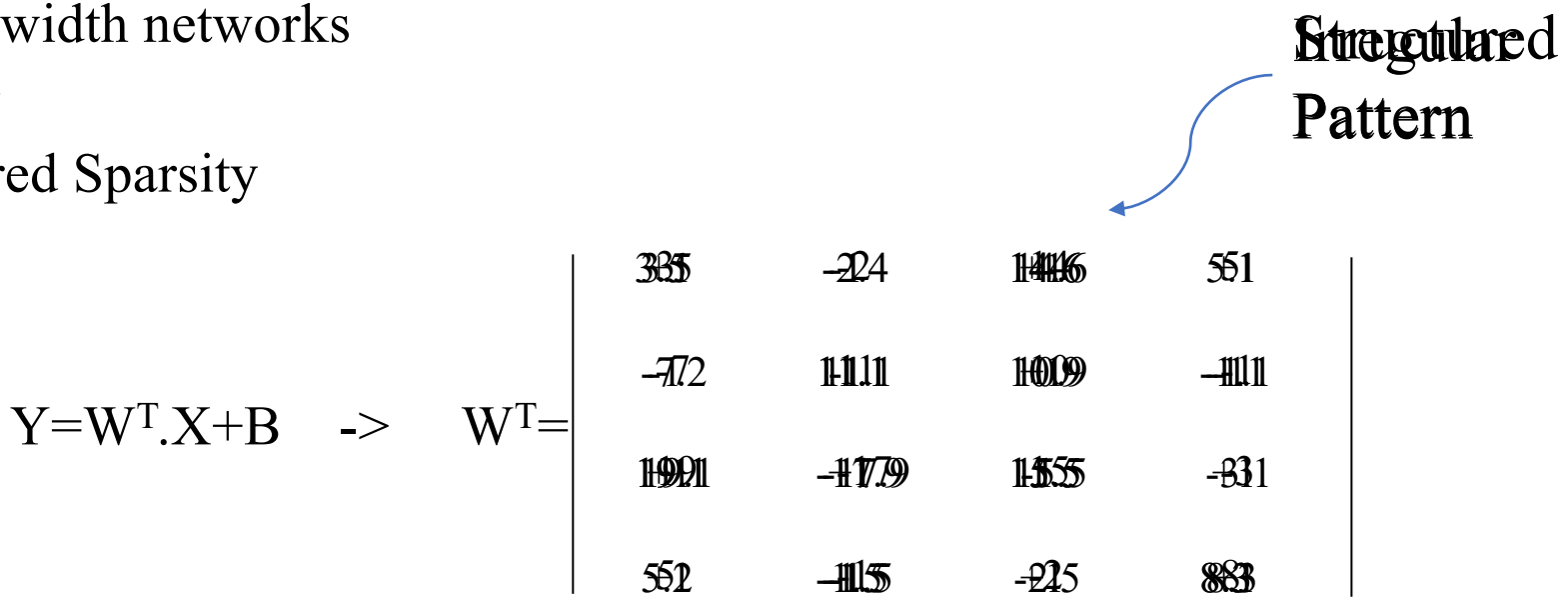


Plot drawn by EEHPC JHU lab Uttej Kalakuri

Data from Zhu, Xunyu, et al. "A survey on model compression for large language models." *arXiv preprint arXiv:2308.07633* (2023).

Review: Model Compression Reduction Techniques

- Neural Network Model Compression and Computation Reduction
  - Quantization
  - Low bitwidth networks
  - Pruning
  - Structured Sparsity



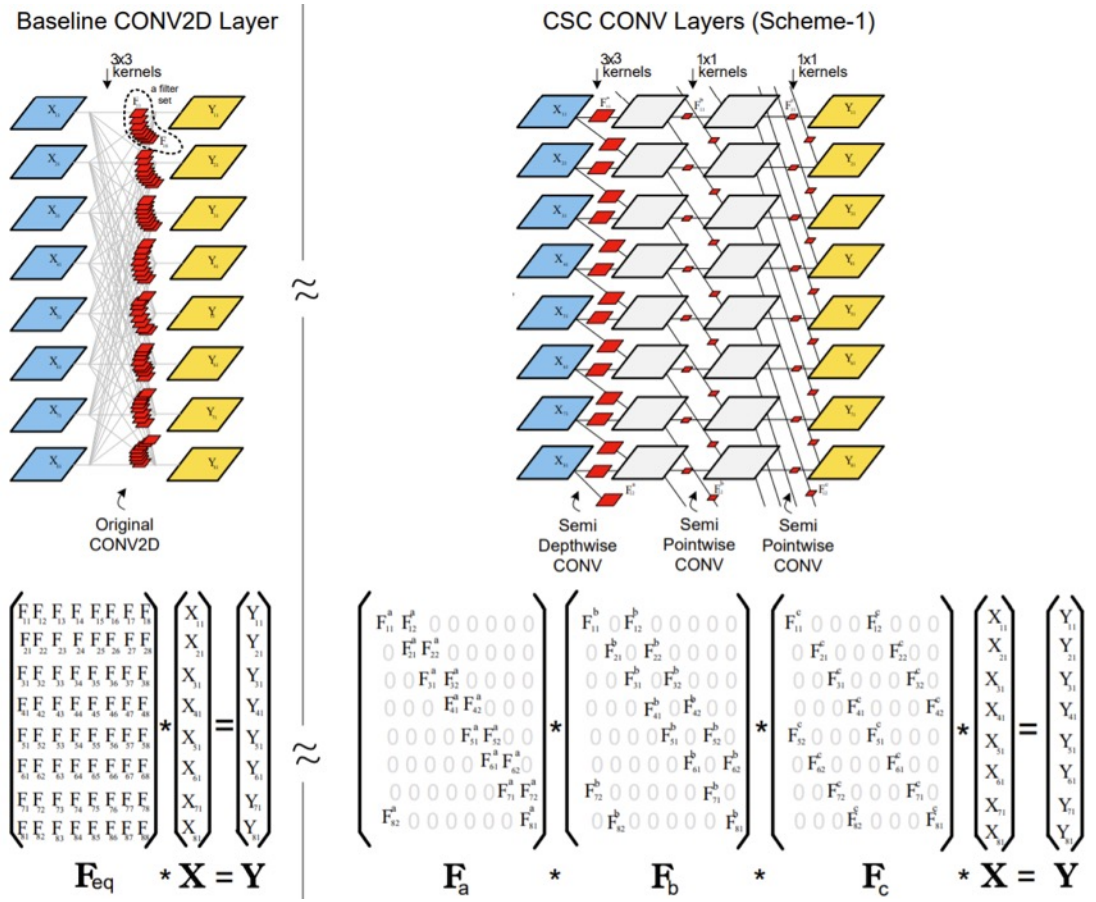
Opportunities for Extreme Model Compression: Ternarized Cyclic Sparsely Connected Layers

Morteza Hosseini, and Tinoosh Mohsenin” Cyclic Sparsely Connected Architectures – from Foundations to Applications ” IEEE Solid-State Circuits Magazine

Morteza Hosseini,and Tinoosh Mohsenin” Cyclic Sparsely Connected Architectures for Compact Deep Convolutional Neural Networks ” IEEE Transactions on Very Large Scale Integration Systems (TVLSI)

Morteza Hosseini and Tinoosh Mohsenin” Binary Precision Neural Network Manycore Accelerator ” ACM Journal on Emerging Technologies in Computing Systems (JETC)

Cyclic Sparsely Connected (CSC) Architectures



- 2D Convolution

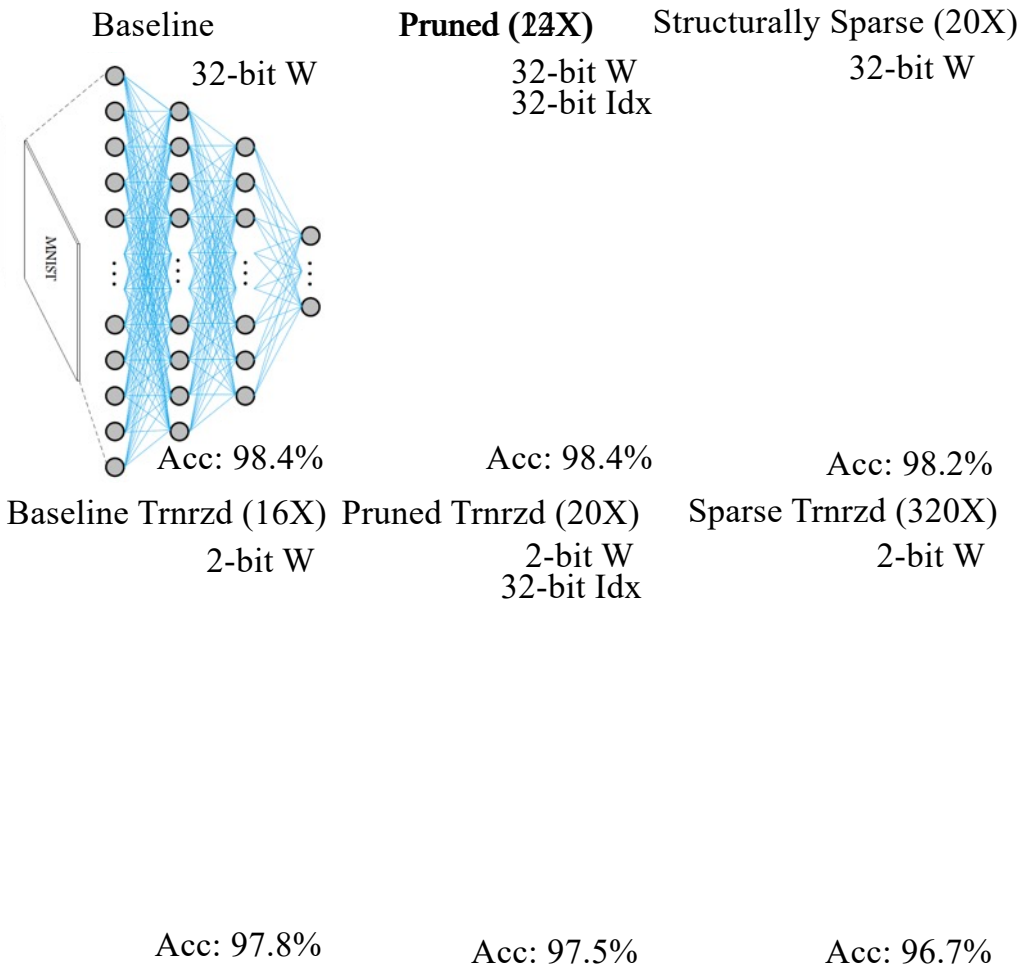
$$\mathbb{Y}_l[i,u,v] = \sum_{j=0}^{N-1} \sum_{w=0}^{W_F-1} \sum_{h=0}^{H_F-1} \mathbb{F}_l[i,j,u,v] \mathbb{X}_l[j,\ u+w, v+h]$$

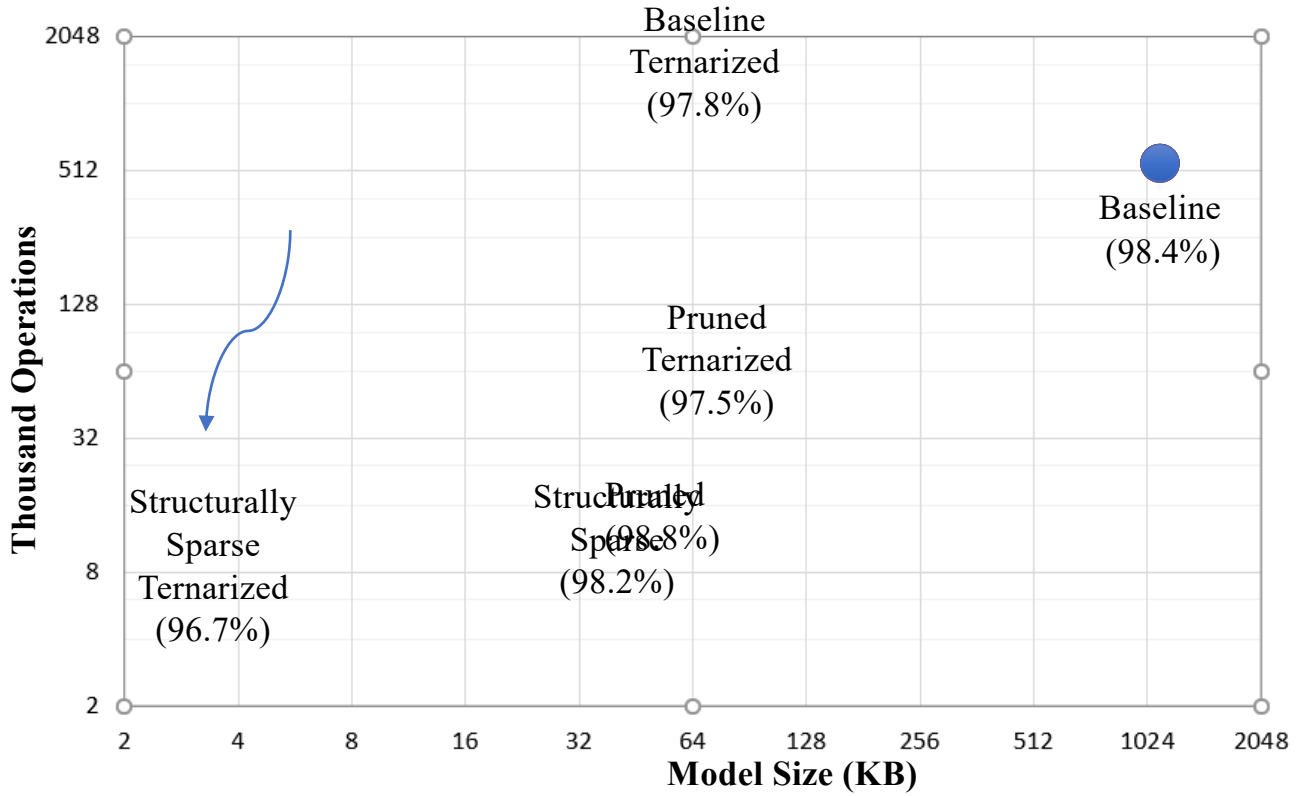
- **Cyclic Channel Dilated 2D Convolution**
- With parameters **N**, **F**, **D**

$$\mathbb{Y}_l[i,u,v] = \sum_{j=0}^{F-1} \sum_{w=0}^{W_F-1} \sum_{h=0}^{H_F-1} \hat{\mathbb{F}}_l[i,j,u,v] \mathbb{X}_l[(i+jD_l) \bmod N, u+w, v+h]$$

- **F=N** → 2D CONV
- **F=1** → DW CONV
- **1<F<N** → CSC-CONV
- **w_F=H_F=1** → CSC-FC
- **w_F=H_F=1, F=N** → FC

Opportunities for Extreme Model Compression: Ternarized Cyclic Sparsely Connected Layers



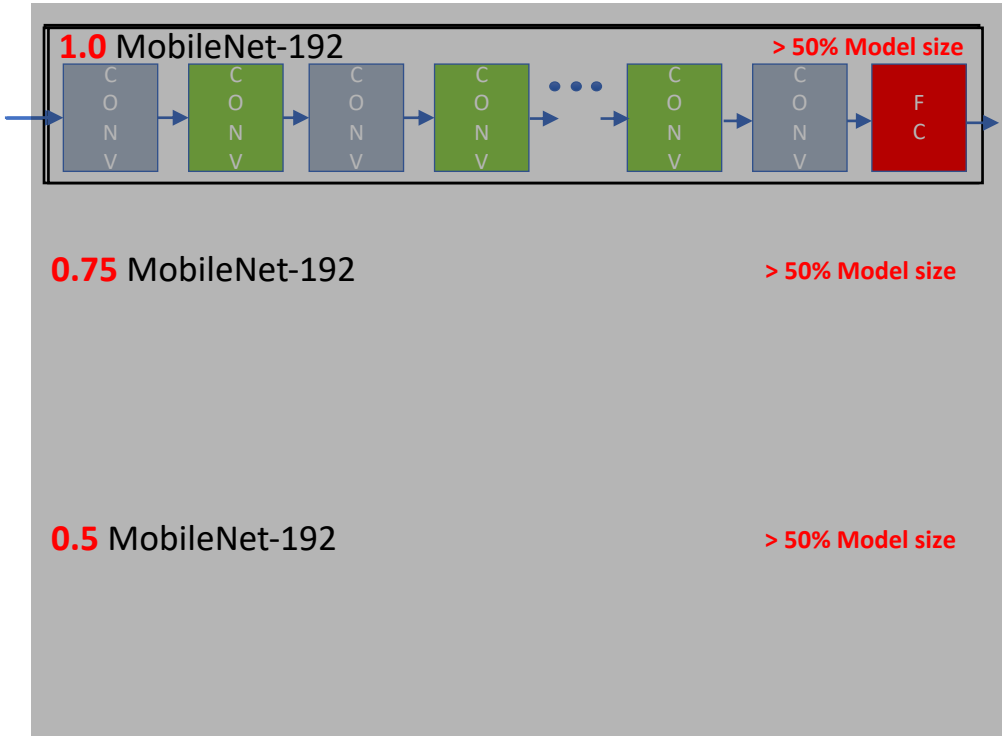


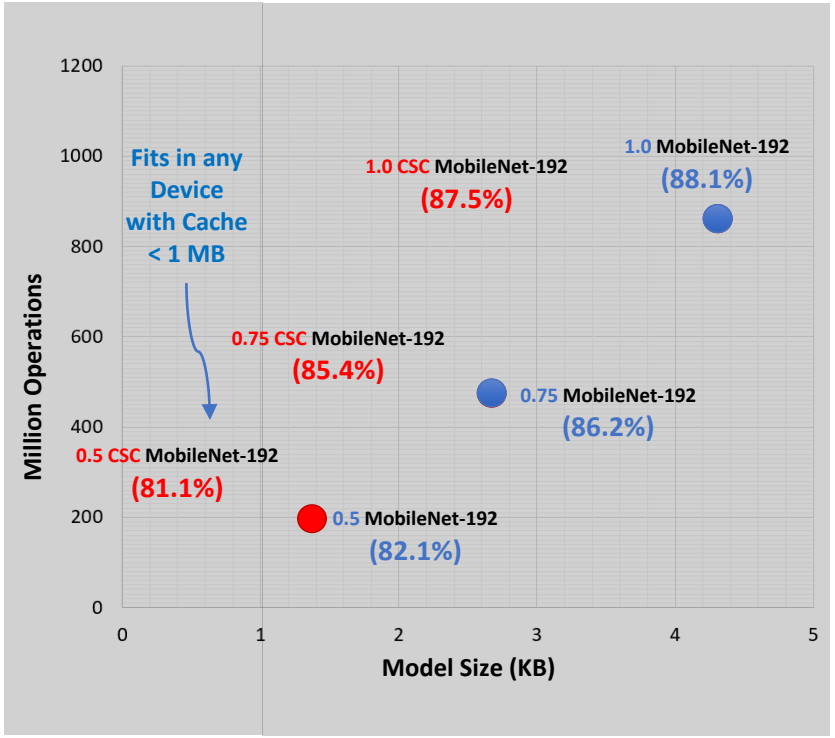
LeNet-300-100 with 32-bit floating-point model as the baseline, its ternarized, and their CSCI compressed models, compared to a related work

LeNet-300-100 (MNIST)	Model Precision	All Layers #Params	#Operations K Ops	Accuracy %	Model Size (Compression)
Baseline	32-bit	266.6K	533.2	98.4	1.06MB (1×)
Compressed with CSCI	**32-bit**	**13.3K**	**26.7**	**98.2**	**53.2KB (20×)**
Ternarized Baseline	2-bit	266.6K	533.2	97.8	66.7KB (16×)
Compressed with CSCI	**2-bit**	**13.3K**	**26.7**	**96.7**	**3.4KB (320×)**
Fine-grain Pruning [10]	32-bit	22.0K	44.0	98.4	88KB (12×)

320X

Opportunities for Model compression: Cyclic Sparsely Connected Architecture



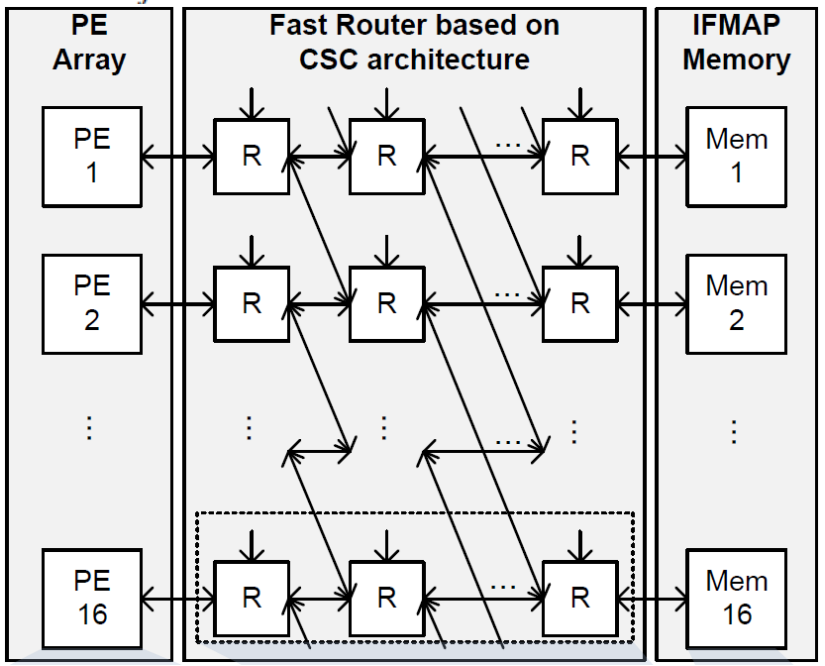


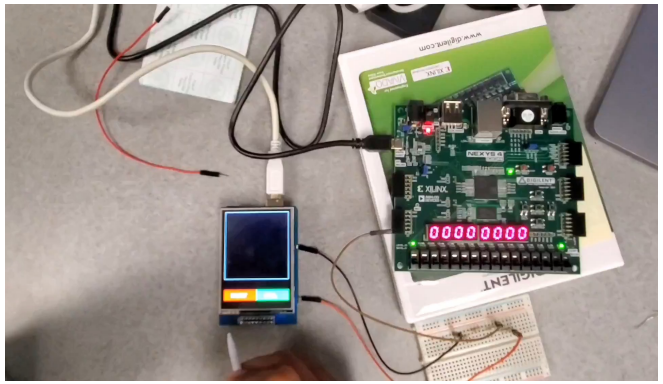
0.5 MobileNetV1-192 (ImageNet)	Model Precision	Total Params	#Ops (M)	Top-5 Acc.(%)	Model Size (Compression)
Baseline	32-bit	1320K	220	83.6	5.3 MB (1.0×)
TFlite Model	8-bit	1320K	220	82.2	1.4 MB (4.0×)
CSCII for last 2 layers	**8-bit**	**873K**	**210**	**81.1**	**0.9 MB (6.0×)**

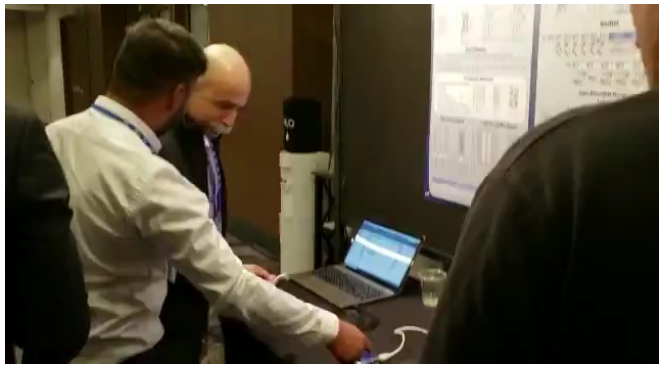
ASIC Chip Fabrication

- TSMC CMOS 65nm 0.6v to 1.0v
- 7.0 mm^2 ,16 PEs – 8-bit datawidt
- 1MHz to 100MHz
- 32KB on chip

$$\mathbf{Y}_l[i] = \sum_{j=0}^{F-1} \mathbf{W}_l[i, (i+jD_l) \bmod N]\mathbf{X}_l[(i+jD_l) \bmod N]$$

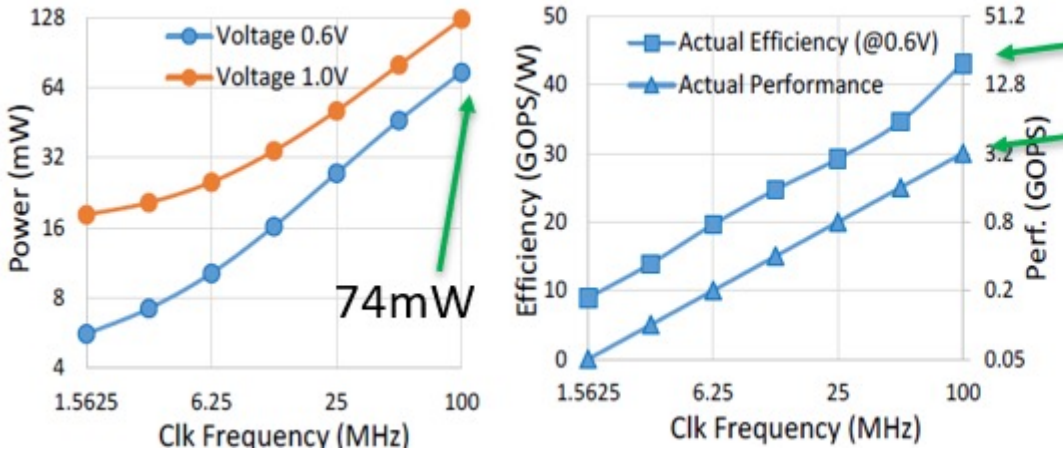






Funding by LPS and Airforce Research Lab
Best Demo Award IEEE ISSCC-SRP
Morteza Hosseini, Nitheesh Manjunath, and Tinoosh Mohsenin, "A 0.9 TOP/S/W Accelerator for Structurally Compressed DNNs Featuring Cyclic Sparsely Connected Layers"

Chip Evaluation: Hardware Implementation and Evaluation



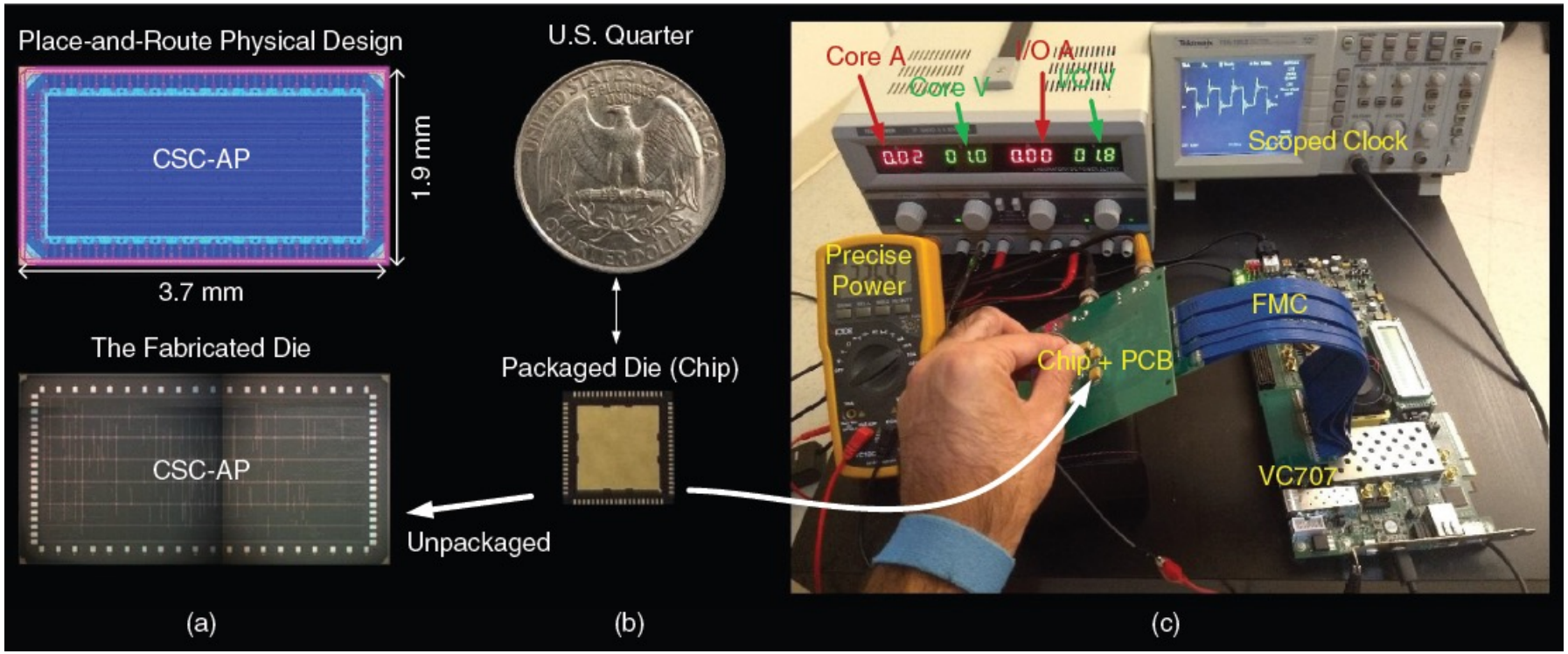
43 GOPJ

3.2 GOPS

$$Power = C\,V^2\,F$$

$$Performance_{true} = 2 \times \#MAC \times freq$$

$$Efficiency_{true} = \frac{Performance_{true}}{Power}$$



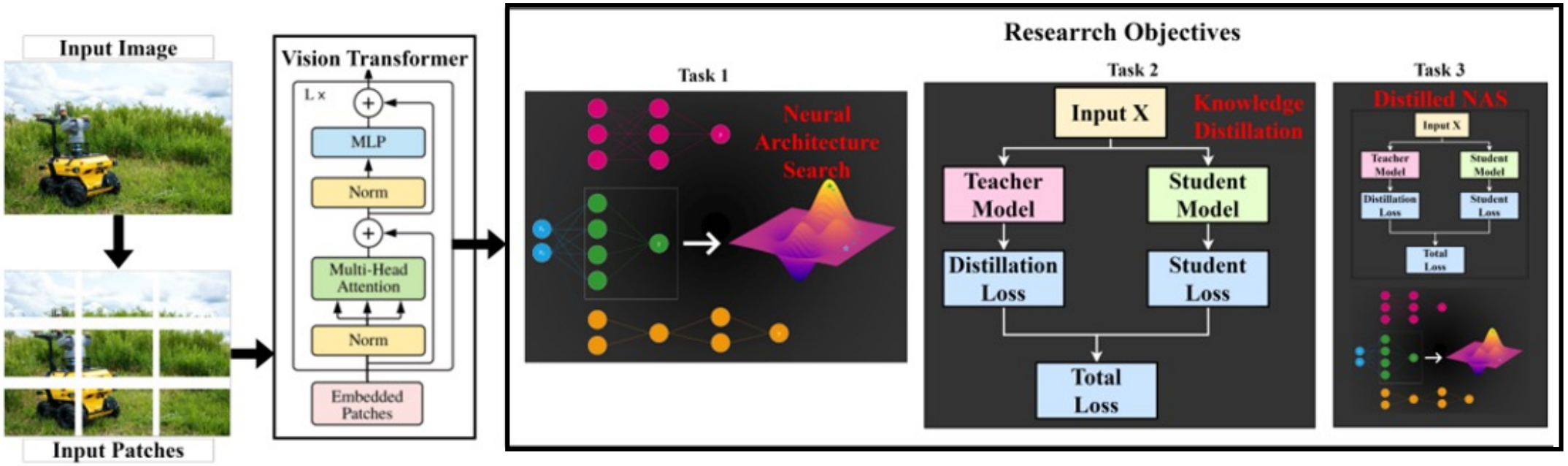
Regression-Based Neural Architecture Search for Optimal Transformers: A Fast Hardware-Software Aware Deterministic Approach

Morteza Hosseini, and Tinoosh Mohsenin” QS-NAS: Optimally Quantized Scaled Architecture Search to Enable Efficient On-Device Micro-AI ” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, (JETCAS)

Arnab Neelim Mazumder and Tinoosh Mohsenin “Reg-TuneV2: Hardware-Aware and Multi-Objective Regression-Based Fine-Tuning Approach for DNNs on Embedded Platforms” IEEE Micro Magazine

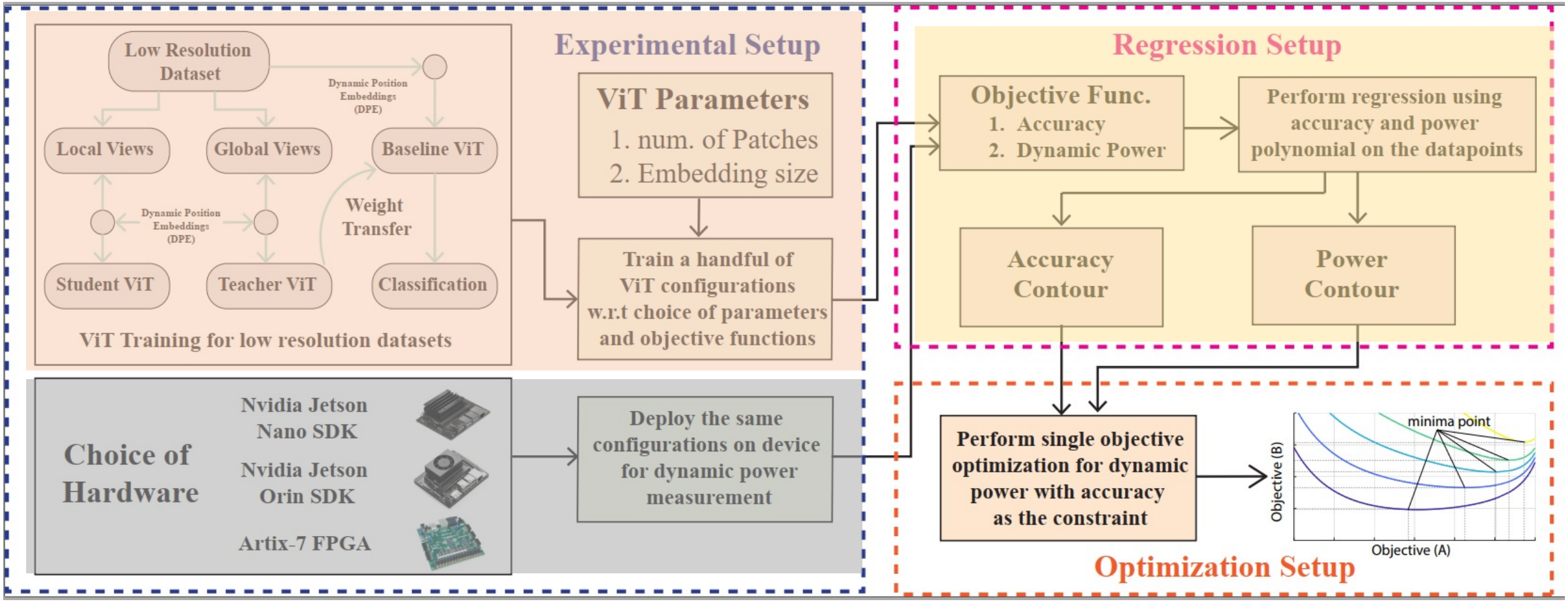
Arnab Neelim Mazumder, Farshad Safavi, Maryam Rahnemoonfar and Tinoosh Mohsenin “Reg-Tune: A Regression-Focused Fine-Tuning Approach for Profiling Low Energy Consumption and Latency” ACM Transactions on Embedded Computing Systems (TECS)

A Fast Hardware-Software Aware Deterministic Approach for Optimal Transformer Models



- **High Computational Cost**: Traditional architecture search methods like grid search and Neural Architecture Search (NAS) are resource-intensive.
- **Lightweight Regression-Based Approach**: The proposed regression-based exploration provides a more efficient alternative to brute-force searches.
- **Efficient Design Space Exploration**: This method predicts power consumption and accuracy using polynomial models for effective exploration.
- **Hardware-Aware Optimization**: The approach integrates hardware considerations, enabling optimization for edge applications.
- What **are the most effective techniques for knowledge distillation** in Transformers to reduce computational and memory requirements while maintaining accuracy?

High Level Overview Regression-Based Architecture Search for Vision Transformers (ViTs)



$$minimize\ NN\ (Dynamic\ Power) for\ config\ \in \{d, n\}$$

$$such\ that\ accuracy\ \geq desired\ accuracy$$

$$Accuracy\ (NN(d,n)) \approx \frac{\boldsymbol{A_6}.d.n + \boldsymbol{A_5}.n + \boldsymbol{A_4}.d + \boldsymbol{A_3}}{d.n + \boldsymbol{A_2}.n + \boldsymbol{A_1}.d + \boldsymbol{A_0}}$$

$$Dynamic\ Power\ \approx\ \hat{B}_2.d.n + \hat{B}_1.\sqrt{n} + \hat{B}_0$$

d: Embedding dimension n: Number of patches

Metareasoning for Optimal Model Selection During Runtime

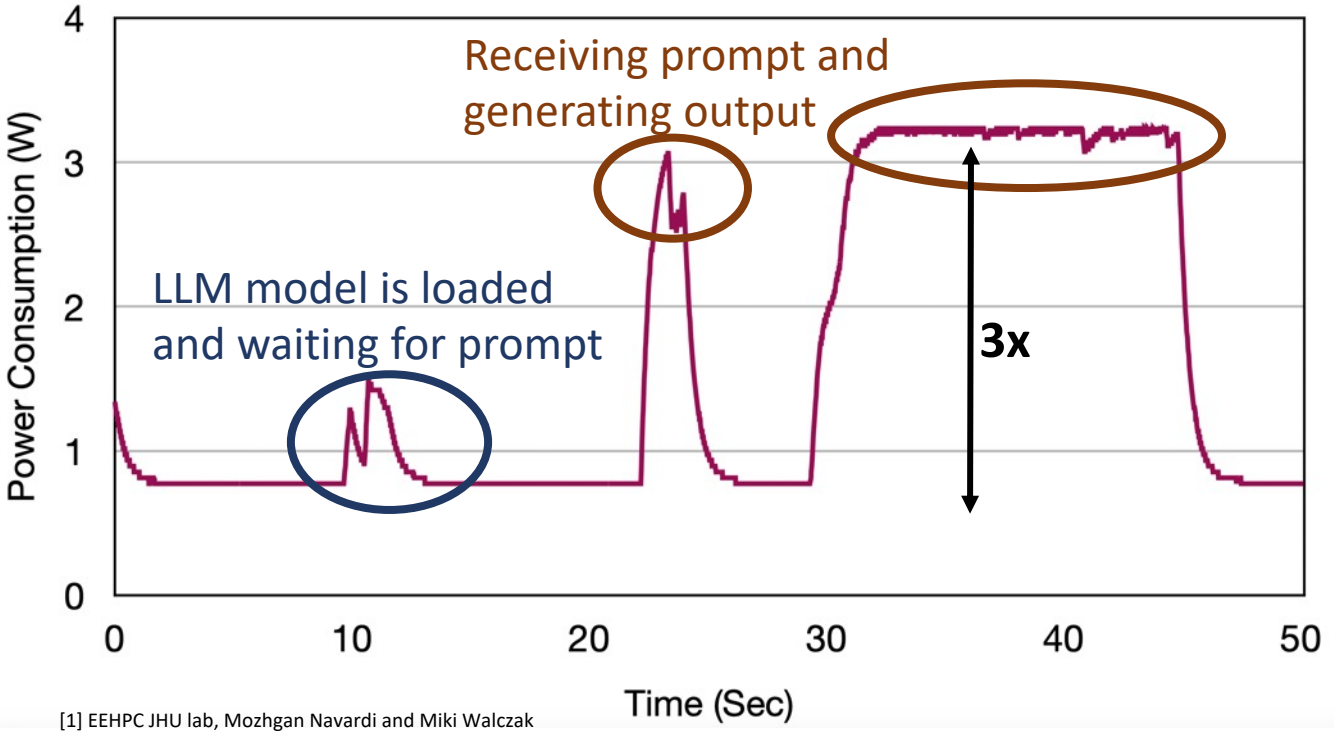
Mozhgan Navardi, Edward Humes, Tinoosh Mohsenin, “MetaTinyML: End-to-End Metareasoning Framework for TinyML Platforms“, IEEE Embedded Systems Letters (ESL), 2024

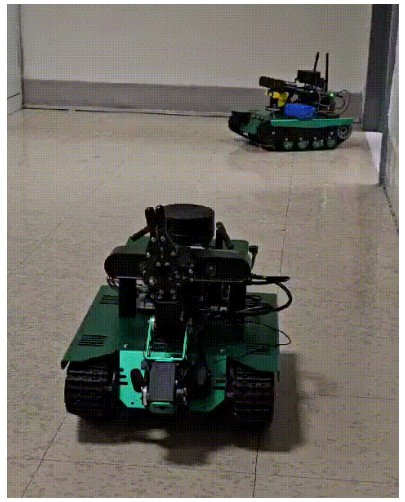
Mozhgan Navardi, Edward Humes, Tejaswini Manjunath, Tinoosh Mohsenin, “MetaE2RL: Toward Metareasoning for Energy-Efficient Multi-Goal Reinforcement Learning with Squeezed Edge YOLO“, IEEE Micro Magazine, 2023

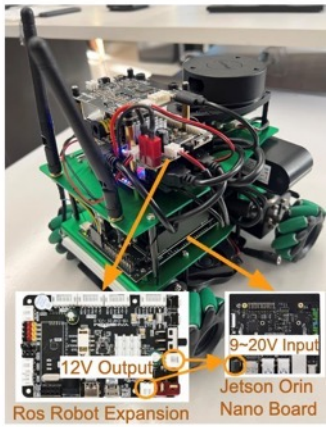
Mozhgan Navardi, Tinoosh Mohsenin
“ MLAE2: Metareasoning for Latency-Aware Energy-Efficient Autonomous Nano-Drones “
The IEEE International Symposium on Circuits and Systems (ISCAS), 2023

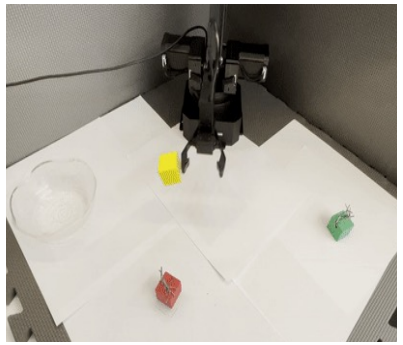
Latency and Power Consumption of LLM for Autonomous Systems

- In Autonomous systems real time reaction to environment changes is critical.
- Tiny LLMs on edge devices consume a significant amount of power and are slow
- LLM Phi2-4bit on NVIDIA Jetson Orin
  - Is 3x more power consumption comparing to Idle power
  - Takes 15 second for generating 170 tokens (a paragraph)
  - 150x times slower than the Yolo Object detection









Configuration	Power (W)	Latency (Sec)	Model Size
IDLE	0.9	-	-
YOLO Object detection model	1.8	0.1	168 MB
Onboard LLM Phi2-4bit	3.3	15 **(150x)**	**1.6 GB**

[1] EEHPC JHU lab, Mozhgan Navardi and Miki Walczak

Current Quantized tiny LLMs on Edge Devices

The table shows a comprehensive TinyLLMs profiling on the 4-bit quantized version of

- Microsoft Phi2-2.7B,
- Meta Llama2-7B,
- Llama3-8B

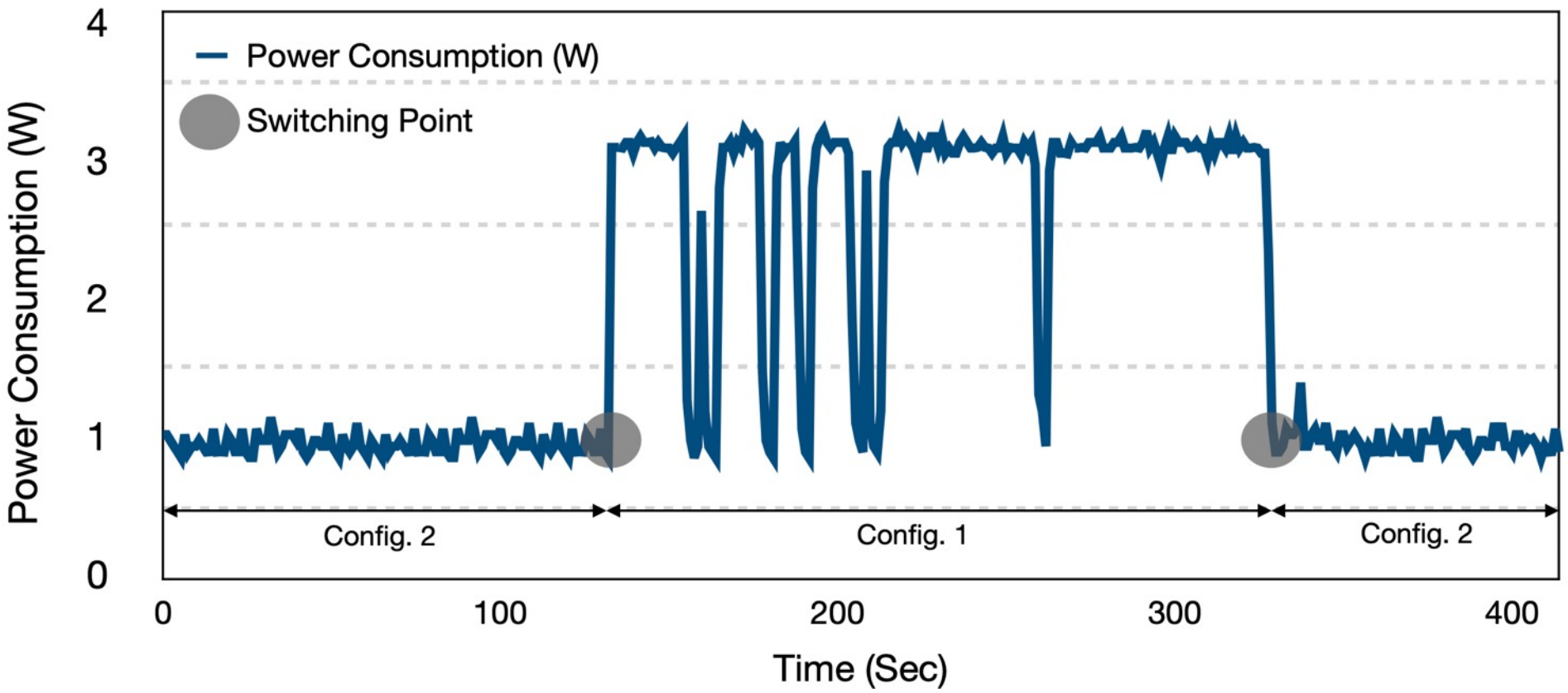
- The result shows by increasing the computation capacity of the platforms from 7W to 50W the performance of the TinyLLMs will increase.
- Microsoft Phi2 consumes less power due to the less amount of computation.

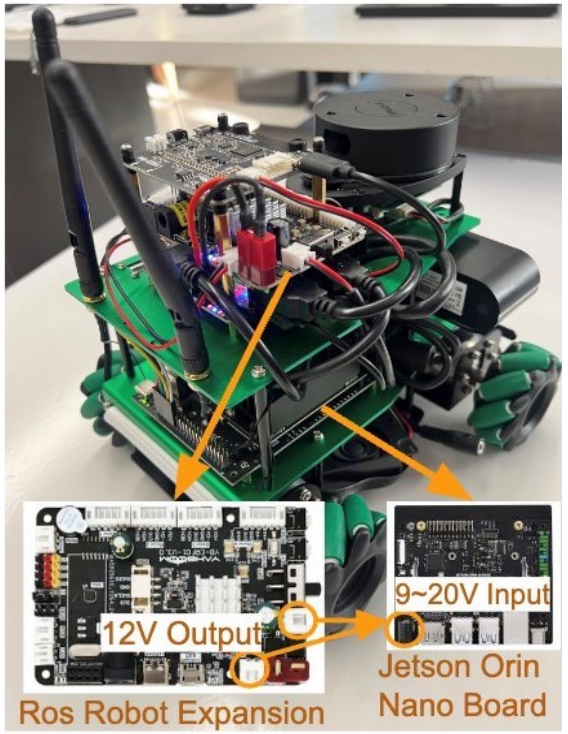
TinyLLM				Tokens/Seconds						Power Consumption (W)				
Name	Interface	Size (GB)		Orin Nano		Orin AGX			AMD Ryzen 9	Orin Nano		Orin AGX		
		Org.	Quant.	7W	15W	15W	30W	50W		7W	15W	15W	30W	50W
Llama2-7B (4q)	TinyChatEngine	13	3.2	2.2	6.1	2.7	5	9	5.9	1.8	4.2	6	7.6	10.8
Llama3-8B (4q)	TinyChatEngine	19.2	4.7	-	-	-	-	-	6.8	-	-	-	-	-
Phi2-2.7B (4q)	llama.cpp	6	1.6	2.4	6.3	2.5	8.3	10.6	-	1.2	2.9	3.2	6.6	8.4
IDLE	N/A									0.4	0.6	2.4	2.4	2.4

Models are from: https://github.com/mit-han-lab/TinyChatEngine

[M. Navardi et al, “Metareasoning for Efficient TinyLLM-based Autonomous Navigation", In progress]

When to use LLMs on Edge Devices- Power analysis on Edge Jetson Orin



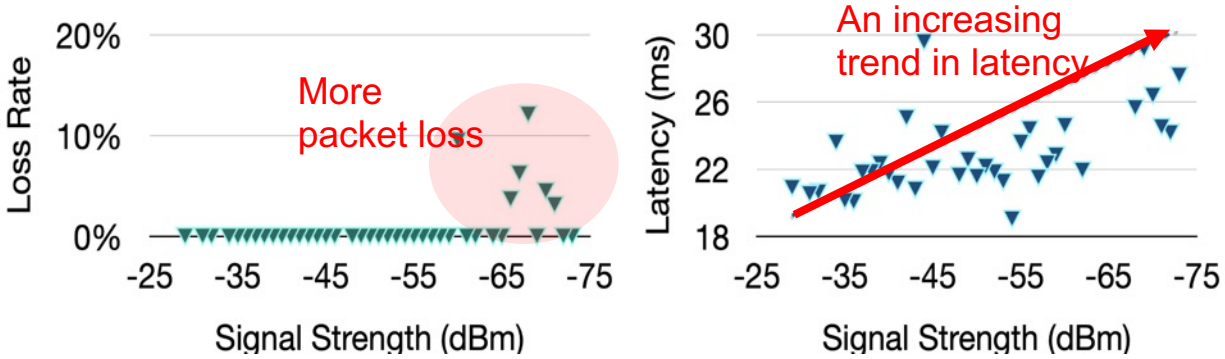


Config. 1	LLM on edge device
Config. 2	LLM on cloud

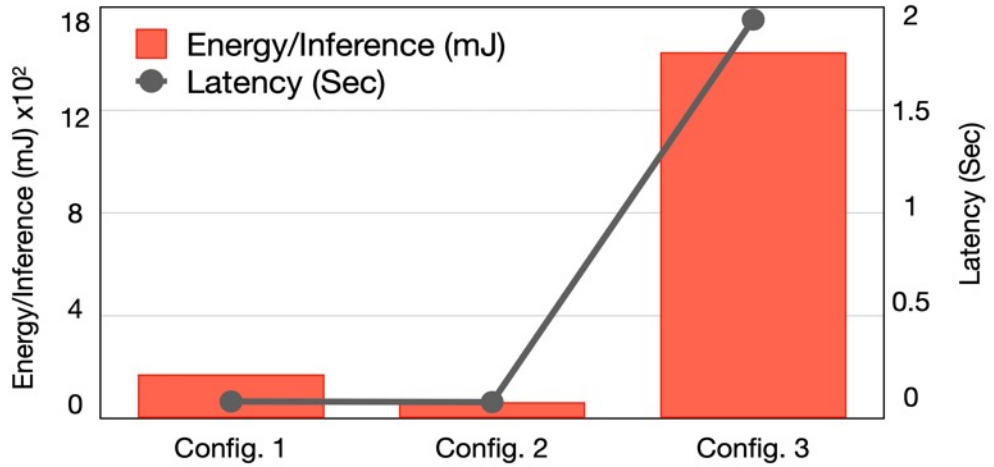
[M. Navardi et al, “Metareasoning for Efficient TinyLLM-based Autonomous Navigation", In progress]

Where to use LLMs Edge or Cloud Devices- Power analysis on Edge Jetson Orin

- Config. 1: onboard, all processes including **LLM, Reinforcement learning (RL), and Object detector (YOLO) run on Jetson Orin.**
- Config. 2: mixed onboard and cloud, where **LLM is on the cloud, RL and Yolo**
- Config 3: on cloud, all processes **run on cloud and the signal is weak, latency high**

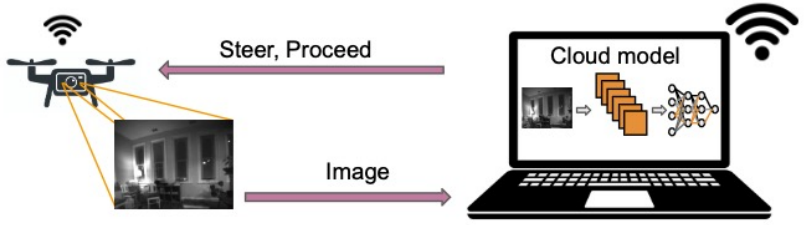


Config. 1	LLM on edge device
Config. 2	LLM on cloud with strong signal
Config. 3	LLM on cloud with weak signal

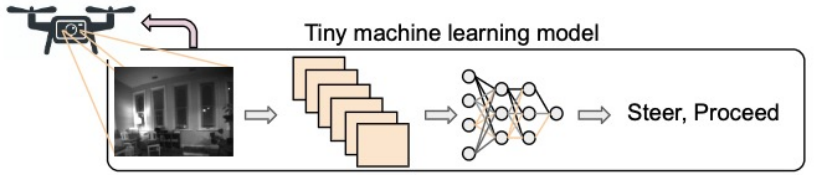


Real time Guarantee in Autonomous systems

Cloud Computing	Edge Computing
▪ Communication power ▪ Security concerns ▪ Latency due to communication	▪ Power consumption limitation ▪ Memory limitation ▪ Latency due to computation

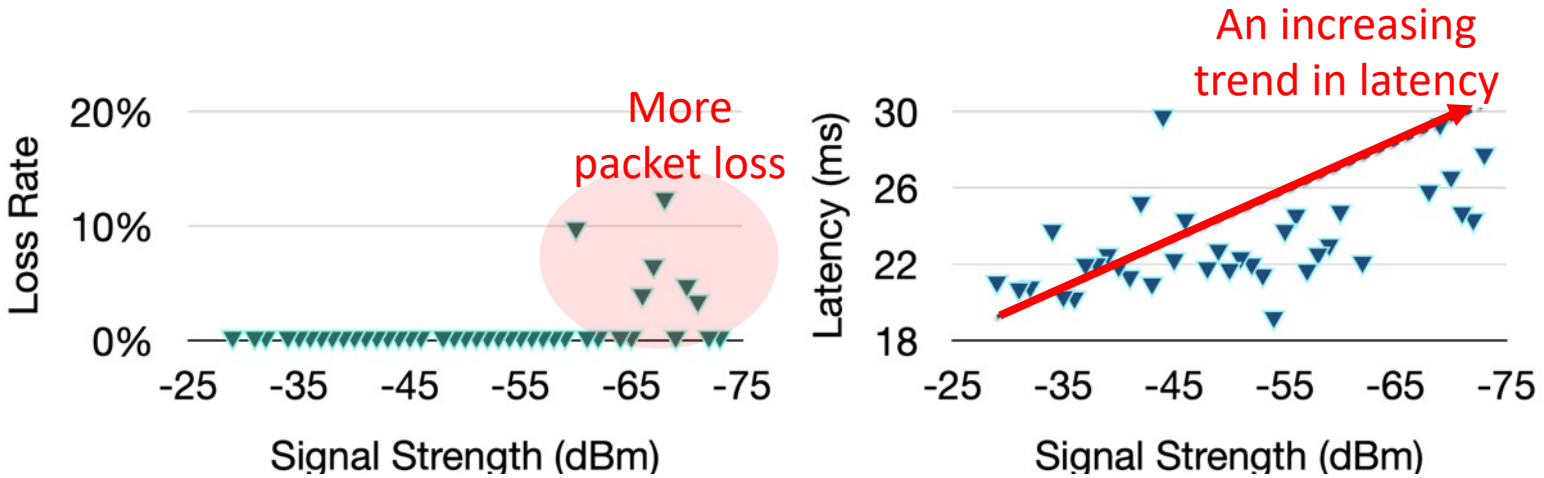


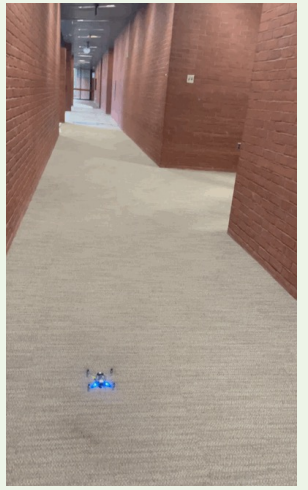
(a) Cloud-based Computing



(b) Edge Computing

- Packet loss rate and average delay are measured based on the signal strength calculated using the equation dBm=10*log10(Power).



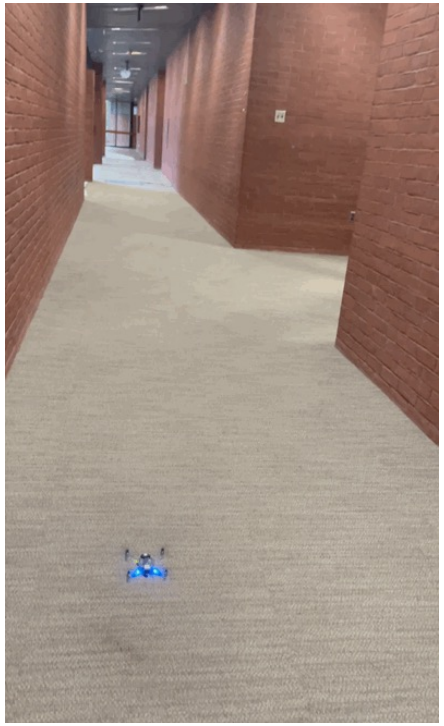


Video Streaming (Drone View)

Cloud Computing Demo

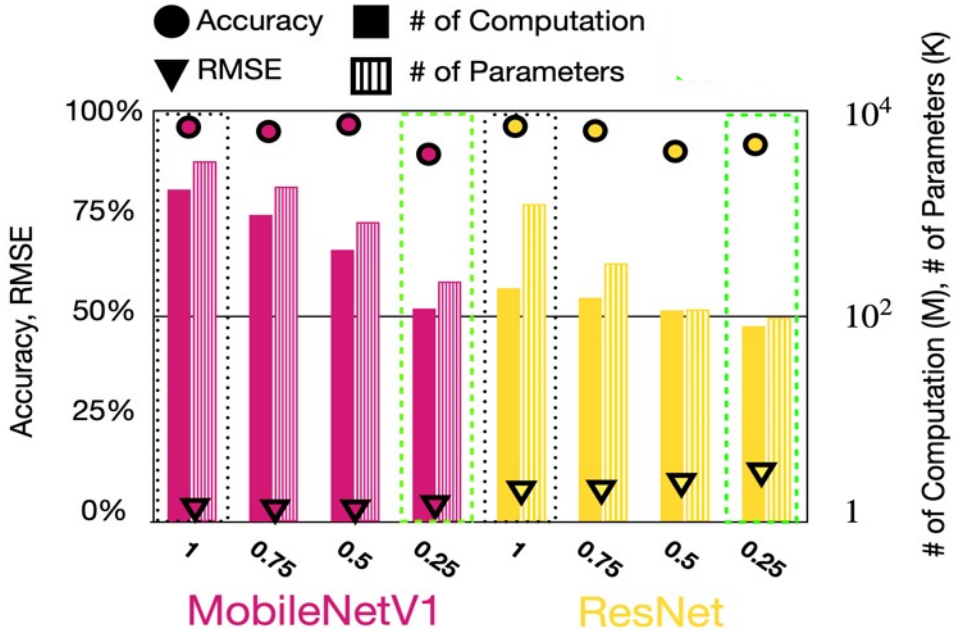
Real time Guarantee for UAV Obstacle Detection: Cloud vs on-board Processing

Streaming to Cloud for processing



Sensing and Processing On-Board

- Model size is reduced from 1M to 98K Parameters to fit **On-Board Crazyflie GAP8 processor**



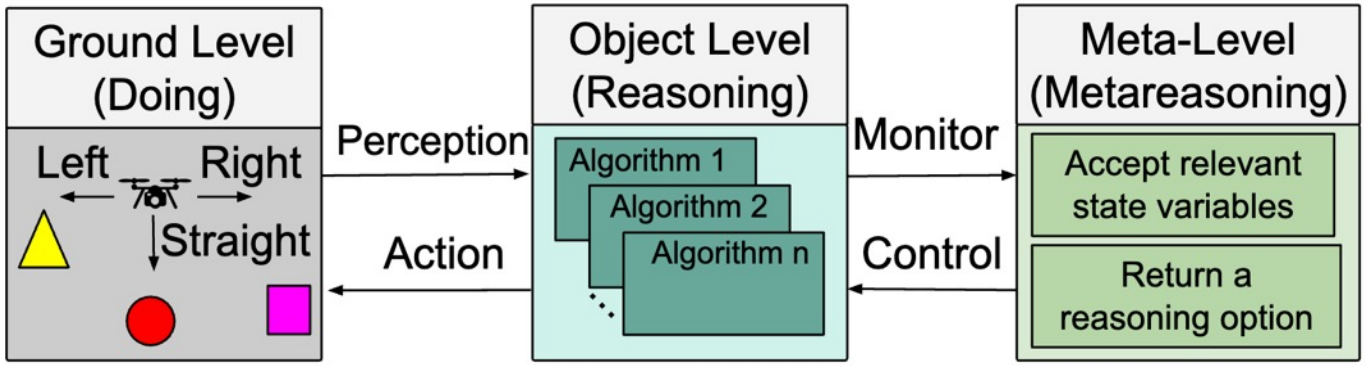
Metric/Platform	Latency Aware Model (m_{edge}, ResNet-25%)			
	Server (M1 Pro)	CPU (ARM)	GPU (Jetson TX2)	On-Board Crazyflie GAP8
Computation Latency (ms)	22	21	13	40.6
Throughput (Inference/Sec)	45	47.6	76.9	**25**
Computation Power (W)	N/A	3.7	4.7	0.85
Performance (GOPS)	3.7	3.8	6.2	2
Energy/Inference (mJ)	N/A	79	70	**34**

Mozhgan Navardi et al, "An Optimization Framework for Efficient Vision-Based Autonomous Drone Navigation", IEEE AICAS 2022

Real time Guarantee: Metareasoning Approach

- **Metareasoning** is a formalization of metacognition, which an intelligent agent does when it thinks about its own thinking
  - determining which algorithm to use to make a decision
  - determining when to stop computing and execute an action

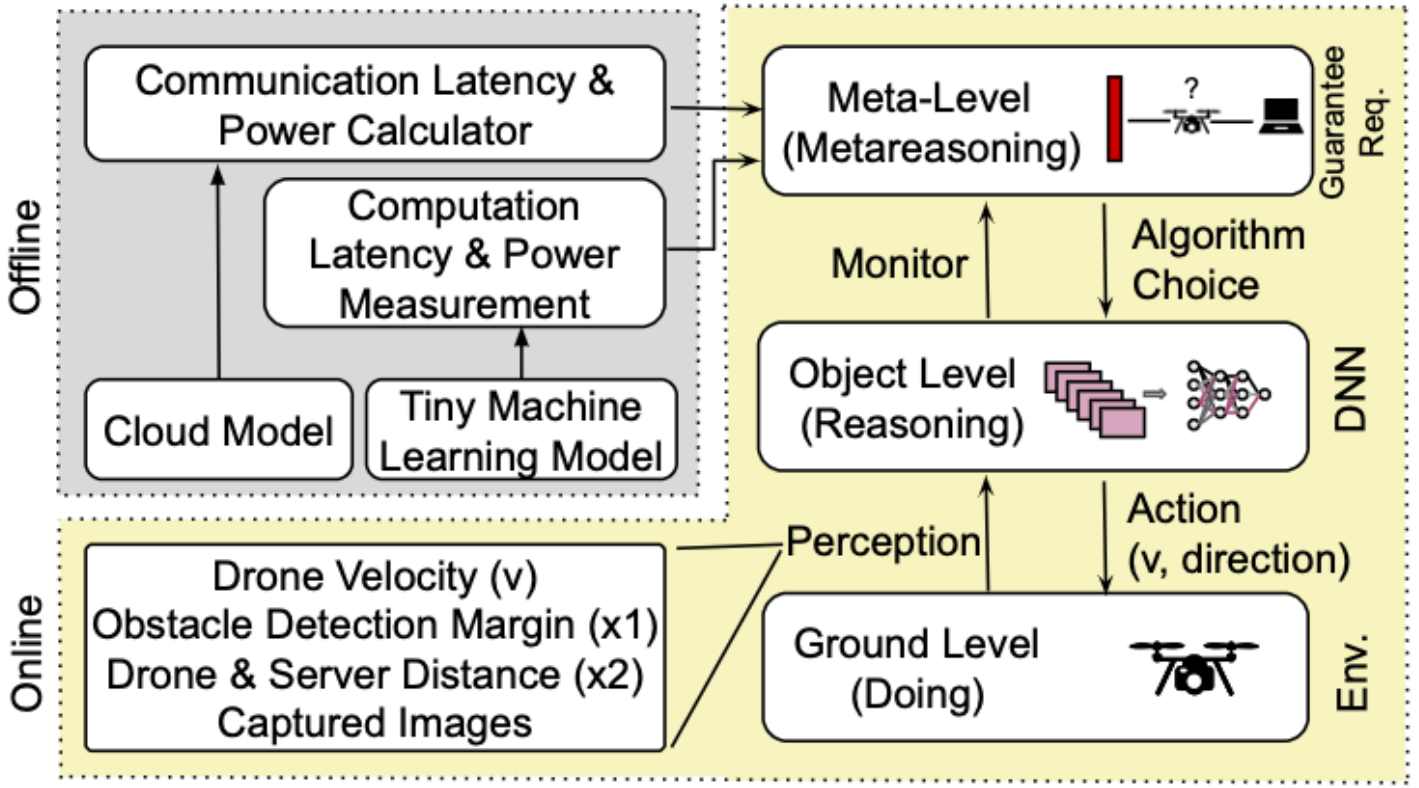
- Three-level model of metareasoning
  - the *ground level* contains the robot's interactions with the environment
  - the *object level* contains the robot's reasoning actions that perceive the environment and select its actions
  - the *meta-level* contains the robot's metareasoning actions that monitor and control the reasoning activities in the object leve



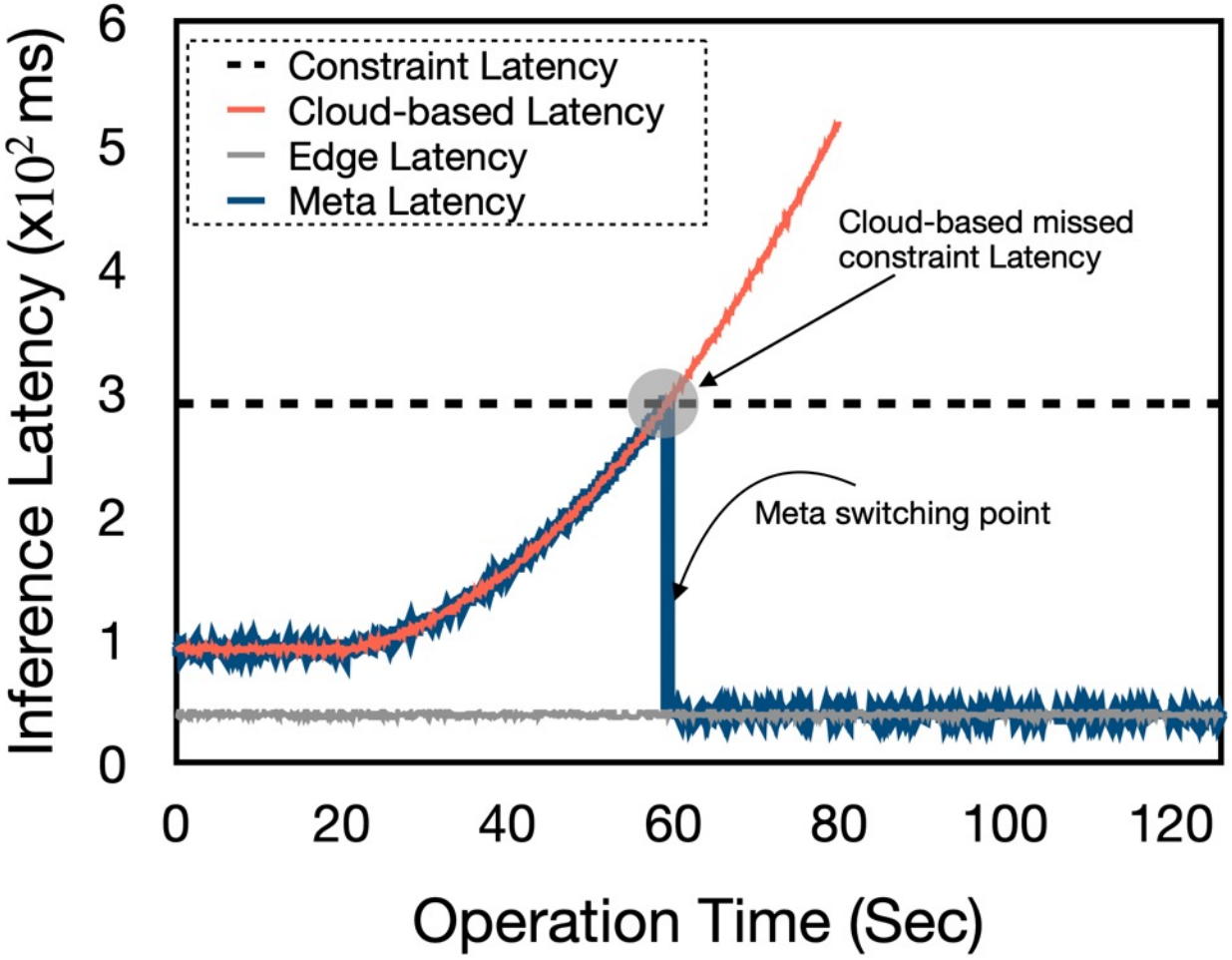
[1] Cox, M. T., and Raja, A., eds. Metareasoning: Thinking About Thinking. MIT Press. OCLC: ocn611551144
[2] Jeffrey W. Hermann, "Metareasoning for Robots"

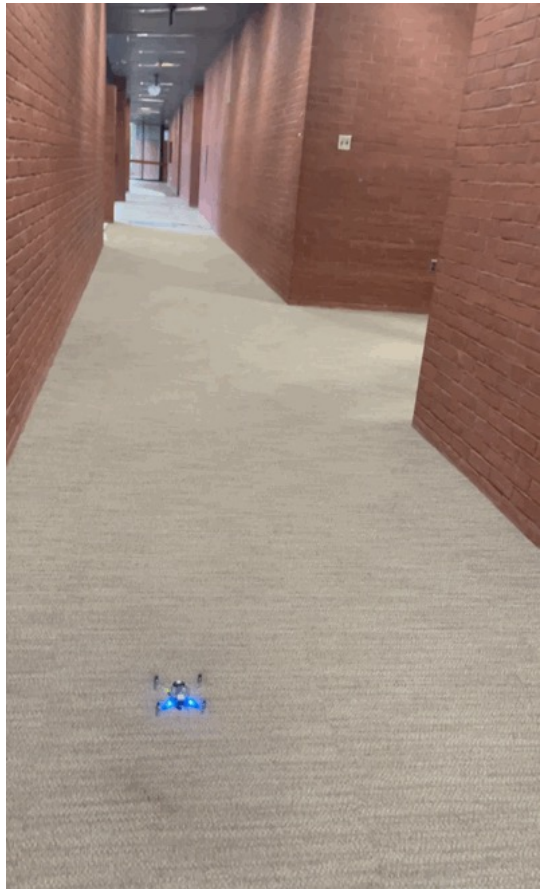
MLAE2: Metareasoning for Latency-Aware Energy-Efficient Autonomous Systems

- For autonomous drone navigation, a **complex DNN** can be applied on the cloud or a **tiny model** can be applied on the edge with fewer resources.
- The Proposed approach includes two online and offline phases. Power consumption and latency for both models have been measured in the offline phase.
- The online phase includes ground, object, and meta levels for reasoning to do the computing on cloud or edge based on the environment changes such as drone velocity, drone and server distance, and obstacle detection margin.
- The drone does the computations on the cloud or the edge based on the metareasonig policy.



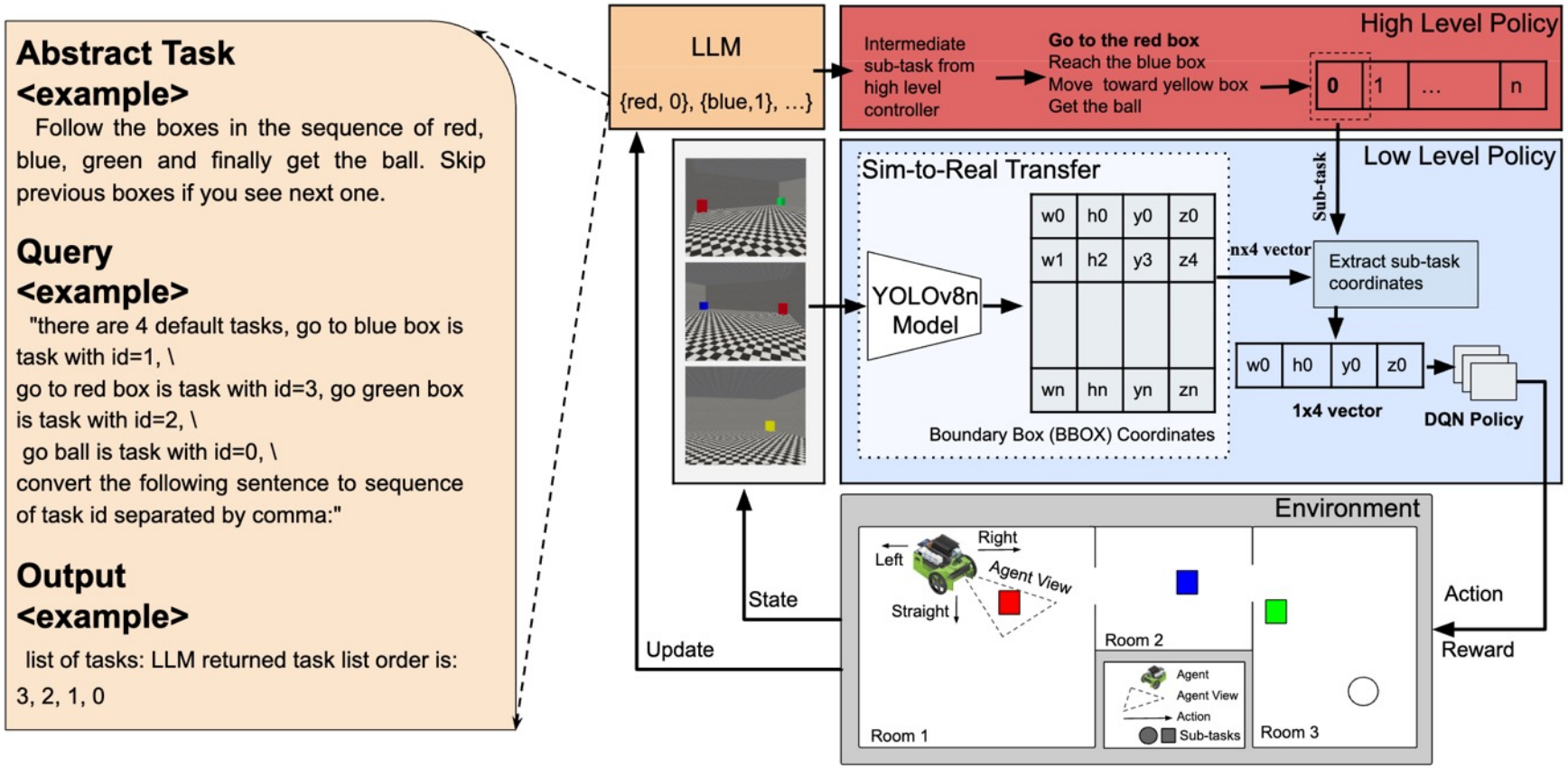
How Metareasoning Guarantee Real-time Performance





Example: LLMs for Multi-task Autonomous Navigation

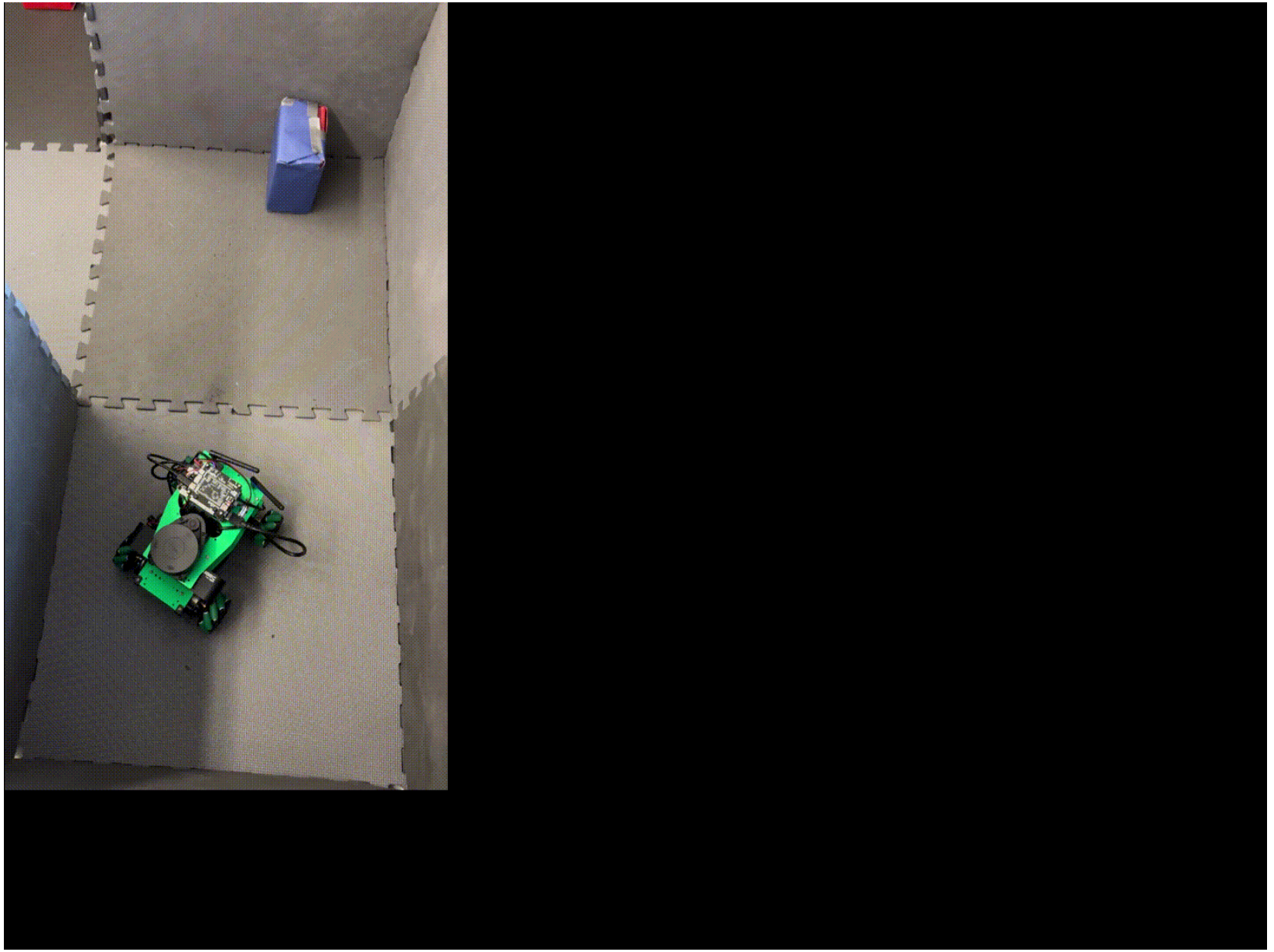
- Proposed LLM-based dynamic planning framework
- An abstract task is passed to the LLM and it dynamically plans the next task to reach the goal.
- Each sub-task is trained with Reinforcement Learning (RL) to reach the goal using a DQN policy.
- Case Study: In the environment, there are four boxes and reaching the ball is the final task,
- The proposed framework can be extended to various types of tasks.



[M. Navardi et al, “Metareasoning for Efficient TinyLLM-based Autonomous Navigation", In progress]

LLMs Planner for Autonomous Navigation - Sim to Real Transfer

- Deployed the trained model on the Rosmaster with NVIDIA Jetson Orin 8GB.
- In the target demo, the LLM-based RL agent successfully reaches the goal in sequence by following the boxes while LLM plans the sub-tasks for the agent.
- A video demonstration of the proposed approach is given below:



[M. Navardi et al, “Metareasoning for Efficient TinyLLM-based Autonomous Navigation", In progress]

Conclusion

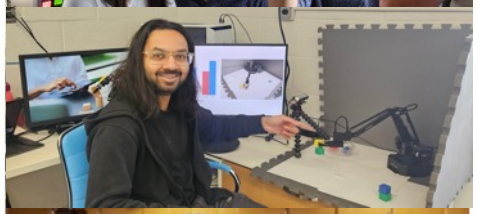
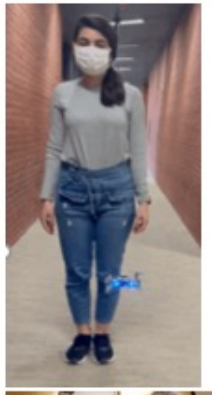
- AI Energy Consumption, Real Time Operation, and Carbon Footprint is a major concern for AI compute.
- Cross Layer design across Algorithms, Modeling, Circuits, and Hardware Architecture needed for two orders of magnitude energy reduction.
- Opportunities for extreme model compression
  - Cyclic Sparsely Connected with extreme quantization provides 2 orders of magnitude model reduction and energy improvement
  - Fast Analytical search and hardware modeling provides a deterministic approach for least energy efficiency
  - Metareasoning for optimal model selection provides a scalable and efficient approach for large scale models for real time performance guarantee.

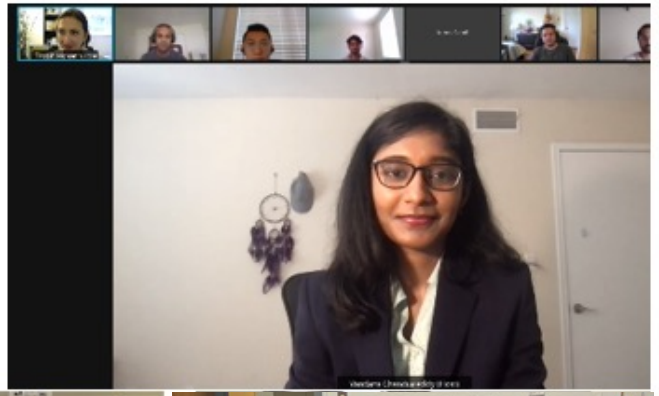
Eco Edge - A New Vision for a Sustainable Edge AI Ecosystem

Tinoosh Mohsenin, Johns Hopkins University
Vijay Reddi, Harvard University

As an expert in the field of edge computing and AI, **we would very much like you to join us for this private workshop**. The event will bring together select leading researchers from academia, the technology industry, NGOs and policy makers to hear presentations and develop workgroup-based output that prescribes the challenges and opportunities that drive a reference for next steps in research, education and policy making.

Thanks to EEHPC Lab members and Current Funding Agencies



Energy Efficient High Performance Computing Lab

https://eehpc.ece.jhu.edu/

Wearable Multi-Physiological Machine Learning for IoT Health

1. COVID-Matter: Multimodal Machine Learning for Respiratory Symptoms Detection
2. EEG Artifact Detection and Removal
3. Epileptic EEG-based Seizure Detection
4. Tongue Drive Assistive Device
5. Multi-modal e S

ARL NSF University of Maryland Baltimore Institute for Clinical & Translational Research

Energy Efficient Autonomous Systems and Robotics

1. Low Power Multi-Agent Reinforcement Learning for UAVs and UGVs
2. Human-Agent Teaming, Hardware Accelerator for Language Guided Reinforcement Learning
3. Context Reasoning and Relational Reinforcement Learning for Multi-Agent Teaming
4. Resource efficient AI for Autonomy
5. A Binarized and Terrorized Edge-AI Autoencoder Accelerator for Reinforcement Learning

ARL DEVCOM Army Research Laboratory DARPA ArtIAMAS

Compressive Sensing and Optical Communincation Recovery for Secured Platforms

NASA U.S. Army

Domain Specific Many-core Platform for Machine Learning

NSF

Edge Hardware Accelerators

1. SensorNet: An Educational Deep Neural Network Framework for Low Power Multi-Modal Data Classification
2. CSC: Cyclic Sparsely Connected Architectures for Compact Deep Convolutional Neural Networks
3. QS-NAS: A Systematic Approach to Optimize Ener Consumption of Low Bit Width Neural Networks

NVIDIA

NSF CAREER AWARDS Northrop Grumman Boeing Xilinx Booz | Allen | Hamilton

Experimental Setup: Hardware Implementation and Power Measurement

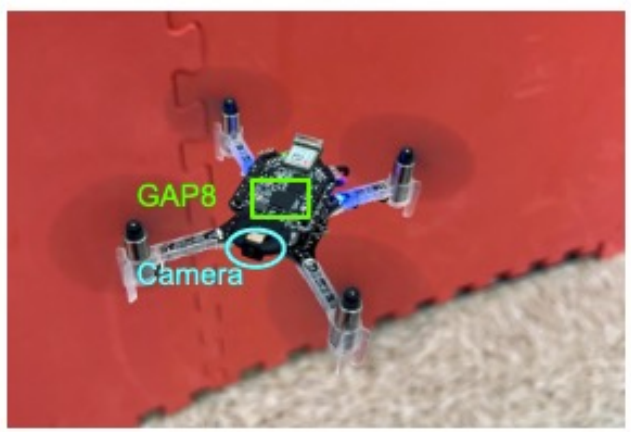
A drone named Crazyflie equipped with a gray-scale camera and an AI-deck is used for the experimental result

AI-Deck features:

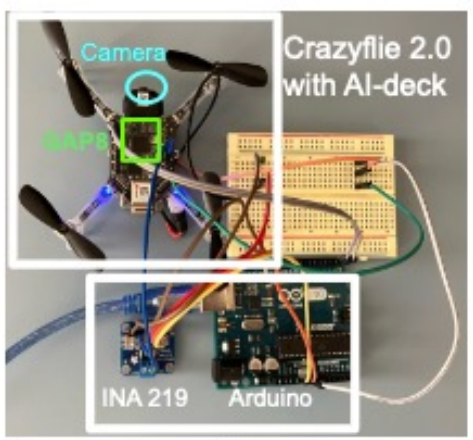
- low power camera
- Wi-Fi module
- Ultra low power RISC-V processor GAP8 for doing edge computing

Power measurement technique:

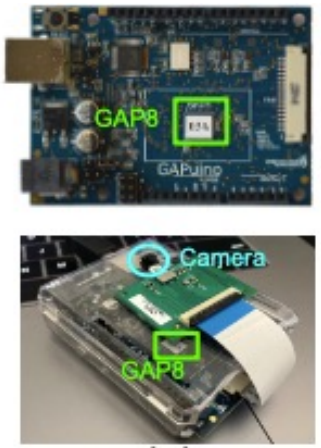
- Setup Arduino and INA 219 to measure



(a)



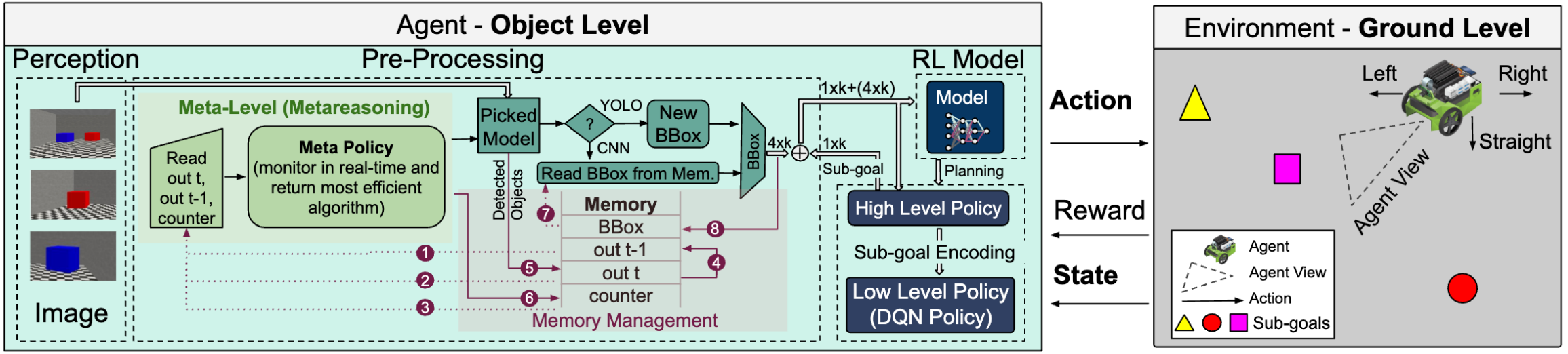
(b)



(c)

GreenWavesTechnologies, “Greenwaves technologies, gapuino development board, https://greenwaves-

MetaTinyML: End-to-End Metareasoning Framework for TinyML Platforms

- End-to-end deployment of Metareasoning approach for an efficient goal-oriented RL-based navigation
- The goal is to search for an object in an environment
- Instead of giving the whole image to RL model, input is shrunk to a small vector including boundary box coordinates and sub-goal: *1x4+(4xk)* which *k* is the number of goals

Proposed Framework