Energy

ENERGY CONSIDERATIONS AND (REAL-TIME) AI FOR IOT

Energy Management: The Background

A key bottleneck in IoT device deployment

Why Consider Energy?

- Energy is one of the key bottlenecks in AI deployment at the edge
- IoT applications often use energy-limited (e.g., battery-operated) devices (wireless sensors, phones, etc, …)
- Processor speed grows faster than battery capacity: energy becomes a bottleneck

Background: Time and Power

Energy = Power x Time

Trade -off:

- Work more quickly (higher power, less time)
  - Example 1: Increase CPU clock speed
  - Example 2: Use more cores in parallel
- Pace yourself (lower power, more time)
- What's better from an *energy* perspective?

Background: The Power of Computation

Terminology

- R : Power spent on computation
- V : Processor voltage
- f : Processor clock frequency
- R_0 : Leakage power

Power spent on computation is:

- $R = k_v V^2 f + R_0$

  where k_v is a constant

Background: The Energy of Computation

Power spent on computation is:

- $R = k_v V^2 f + R_0$

Consider a task of length C clock cycles and a processor operating at frequency f

The execution time is $t = C/f$

Energy spent is:

- $E = R\,t = (k_v V^2 f + R_0)(C/f)$

Energy and Reducing Processor Frequency

Power spent on computation is:

- $R = k_v V^2 f + R_0$

Energy spent is:

- $E = R\,t = (k_v V^2 f + R_0)(C/f)$

Question:

- Does it make sense to operate the processor at a reduced speed to save energy? Why or why not?

Reducing Processor Frequency
Good or Bad?

Does it make sense to operate the processor at a reduced speed to save energy? Why or why not?

Possible Answer:

$$E = R\,t = (k_v\,V^2 f + R_0)(C/f) = k_v\,V^2 C + R_0 C/f$$

- Conclusion: E is minimum when f is maximum.

  → Operate at top speed

- Is this really true? What are the underlying assumptions?

Dynamic Voltage Scaling (DVS):

Reducing Voltage and Frequency

Processor voltage can be decreased if clock frequency is decreased

- Voltage and frequency can be decreased roughly proportionally.
- In this case (where $V \sim f$):

$$R = k_f f^3 + R_0$$

$$E = (k_f f^3 + R_0)(C/f) = k_f f^2 C + R_0 C/f$$

Dynamic Voltage Scaling (DVS):

Reducing Voltage and Frequency

Processor voltage can be decreased if clock frequency is decreased

- Voltage and frequency can be decreased roughly proportionally.

$$R = k_f f^3 + R_0$$

$$E = (k_f f^3 + R_0)(C/f) = k_f f^2 C + R_0 C/f$$

- Question: Does reducing frequency (and voltage) increase or decrease total energy spent on a task?

Dynamic Voltage Scaling (DVS):

The Critical Frequency

There exists a minimum frequency below which no energy savings are achieved

$$E = k_f f^2 C + R_0 C/f$$

$$dE/df = 2k_f f\, C - R_0 C/f^2 = 0$$

$$f = \sqrt[3]{\frac{R_0}{2k_f}}$$

Example

Processor energy consumption is:

$$E = 4f^2 + 1/f$$

What is the optimal frequency?

A Simple DVS Algorithm for Green Real-time Inference at the Edge

1. Calculate the energy optimal frequency
2. Calculate the minimum frequency at which the task set remains schedulable
   - Example: If EDF is used and the utilization is 60% at the maximum frequency f_{max}, then the frequency can be decreased to $0.6\, f_{max}$.
3. Let f_{opt} be the larger of the above two
4. Operate the system at the smallest frequency at or above f_{opt}.

Practical Consideration: Accounting for Off-chip Overhead

In the preceding discussion, we assumed that task execution *time* at frequency f is C/f, where C is the total cycles needed

In reality some cycles are lost waiting for memory access and I/O (Off-chip cycles).

- Let the number of CPU cycles used be C_{cpu} and the time spent off-chip be $C_{off\text{-}chip}$
- Execution time at frequency f is given by

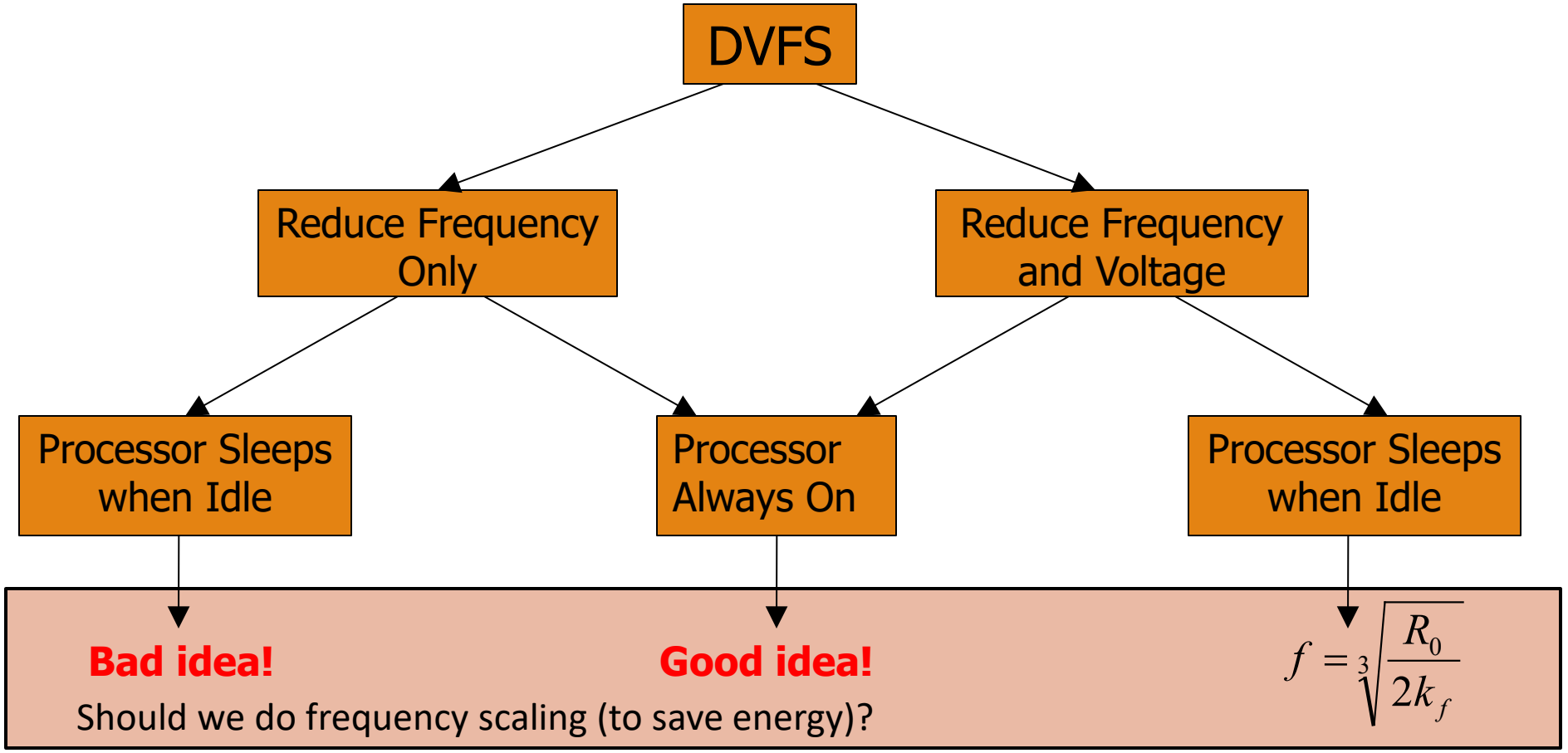
$$C_{cpu}/f + C_{off\text{-}chip}$$

Energy

$$E = (k_f f^3 + R_0)(C_{cpu}/f + C_{off\text{-}chip})$$

To minimize energy, set $dE/df = 0$, then solve for f.

Recap (CPU-centric Tasks)



Memory-centric Tasks

Do the same results apply?

Memory-centric Tasks

Note: reducing CPU frequency reduces power but does not increase the time it takes to run the task.

- Lower frequencies are always better.

DVS on Multicore

From the perspective of minimizing energy, is it always a good idea to use up all cores?

How Many Cores to Use?

Consider using one core at frequency f versus two at frequency $f/2$

Case 1: Total power for one core

- $k_f f^3 + R_0$

Case 2: Total power for two cores

- $2\ \{k_f (f/2)^3 + R_0\} = k_f f^3/4 + 2\ R_0$

How Many Cores to Use?

Consider using one core at frequency f versus two at frequency $f/2$

Case 1: Total power for one core

- $k_f f^3 + R_0$

Case 2: Total power for two cores

- $2\ \{k_f (f/2)^3 + R_0\} = k_f f^3/4 + 2\ R_0$

The general case: n cores

- $n\ \{k_f (f/n)^3 + R_0\} = k_f f^3/n^2 + n\ R_0$

How Many Cores to Use?

The general case: n cores

- $Power = n\ \{k_f (f/n)^3 + R_0\} = k_f f^3/n^2 + n\ R_0$
- $dPower/dn = -2\ k_f f^3/n^3 + R_0 = 0$

$$n = \sqrt[3]{\frac{2k_f f^3}{R_0}}$$

Turning Processors Off
The Cost of Wakeup

Energy expended on wakeup, E_{wake}

To sleep or not to sleep?

Turning Processors Off
The Cost of Wakeup

Energy expended on wakeup, E_{wake}

To sleep or not to sleep?

- Not to sleep (for time t):

$$E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$$

- To sleep (for time t) then wake up:

$$E_{sleep} = P_{sleep}\, t + E_{wake}$$

Turning Processors Off
The Cost of Wakeup

Energy expended on wakeup, E_{wake}

To sleep or not to sleep?

- Not to sleep (for time t):

$$E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$$

- To sleep (for time t) then wake up:

$$E_{sleep} = P_{sleep}\, t + E_{wake}$$

- To save energy by sleeping: $E_{sleep} < E_{no\text{-}sleep}$

$$t > \frac{E_{wake}}{k_v V^2 f + R_0 - P_{sleep}}$$

Turning Processors Off
The Cost of Wakeup

Energy expended on wakeup, E_{wake}

To sleep or not to sleep?

- Not to sleep (for time t):

$$E_{no\text{-}sleep} = (k_v V^2 f + R_0)\, t$$

- To sleep (for time t) then wake up:

$$E_{sleep} = P_{sleep}\, t + E_{wake}$$

- To save energy by sleeping: $E_{sleep} < E_{no\text{-}sleep}$

$$t > \frac{E_{wake}}{k_v V^2 f + R_0 - P_{sleep}}$$

Minimum sleep interval

Dynamic Power Management

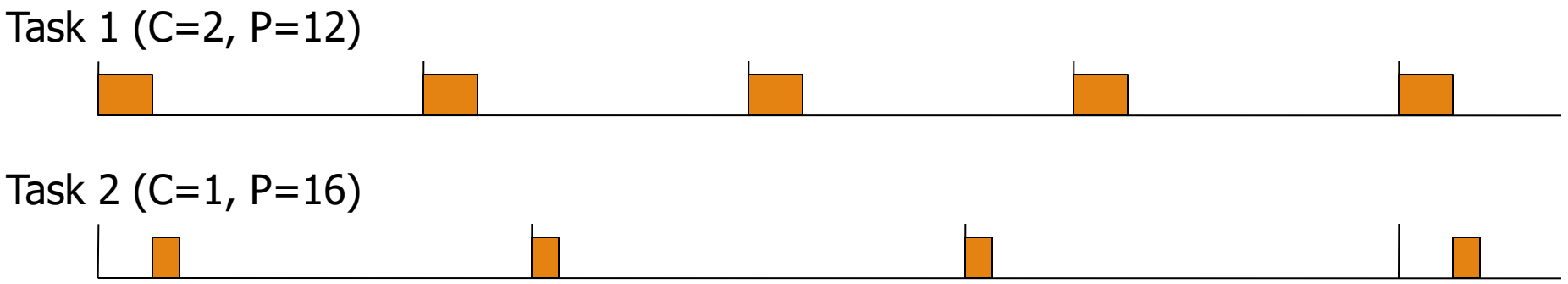
DPM refers to turning devices off (or putting them in deep sleep modes)

Device wakeup has a cost that imposes a minimum sleep interval (a breakeven time)

DPM must maximize power savings due to sleep while maintaining schedulability

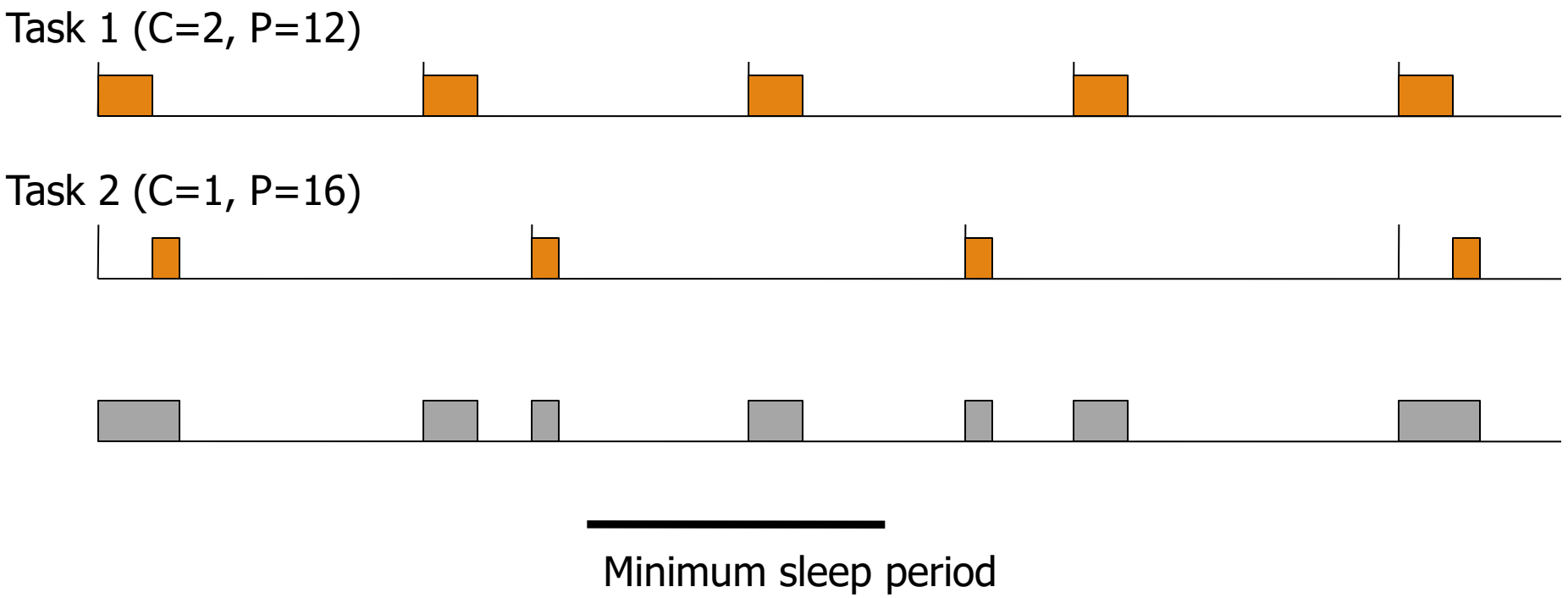
DPM and the Problem with Work-conserving Scheduling

Example:



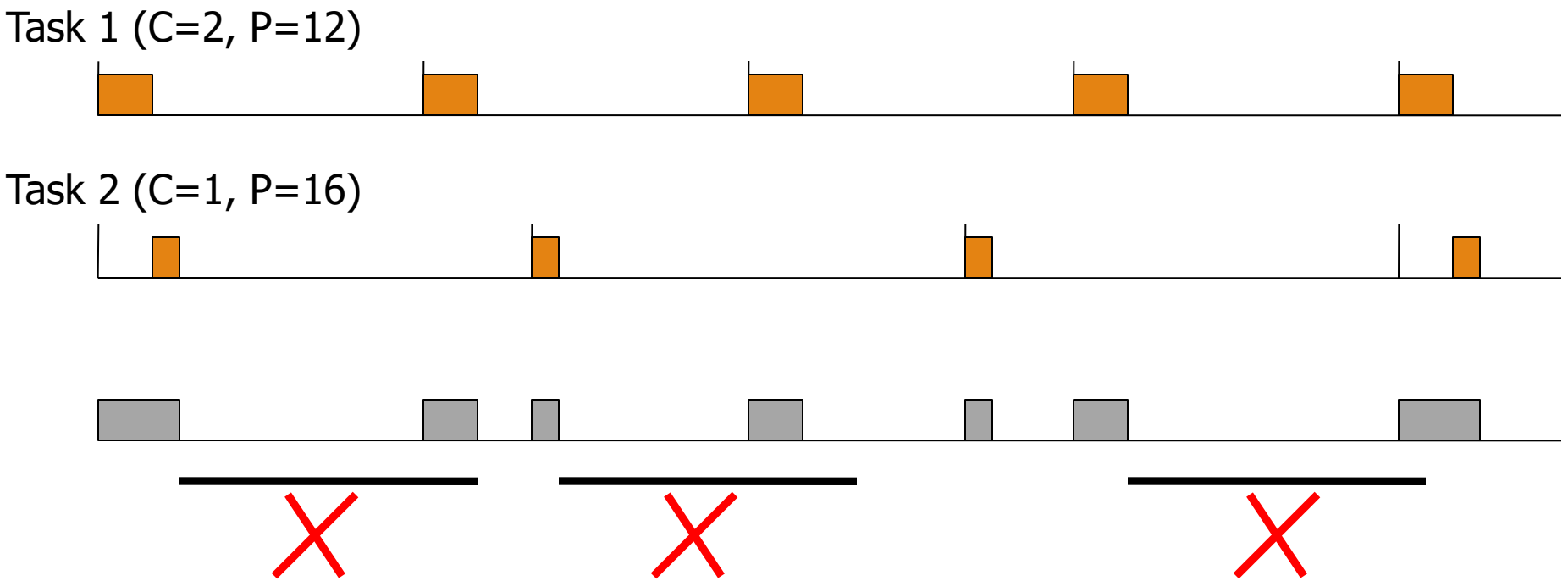
DPM and the Problem with Work-conserving Scheduling

Example:



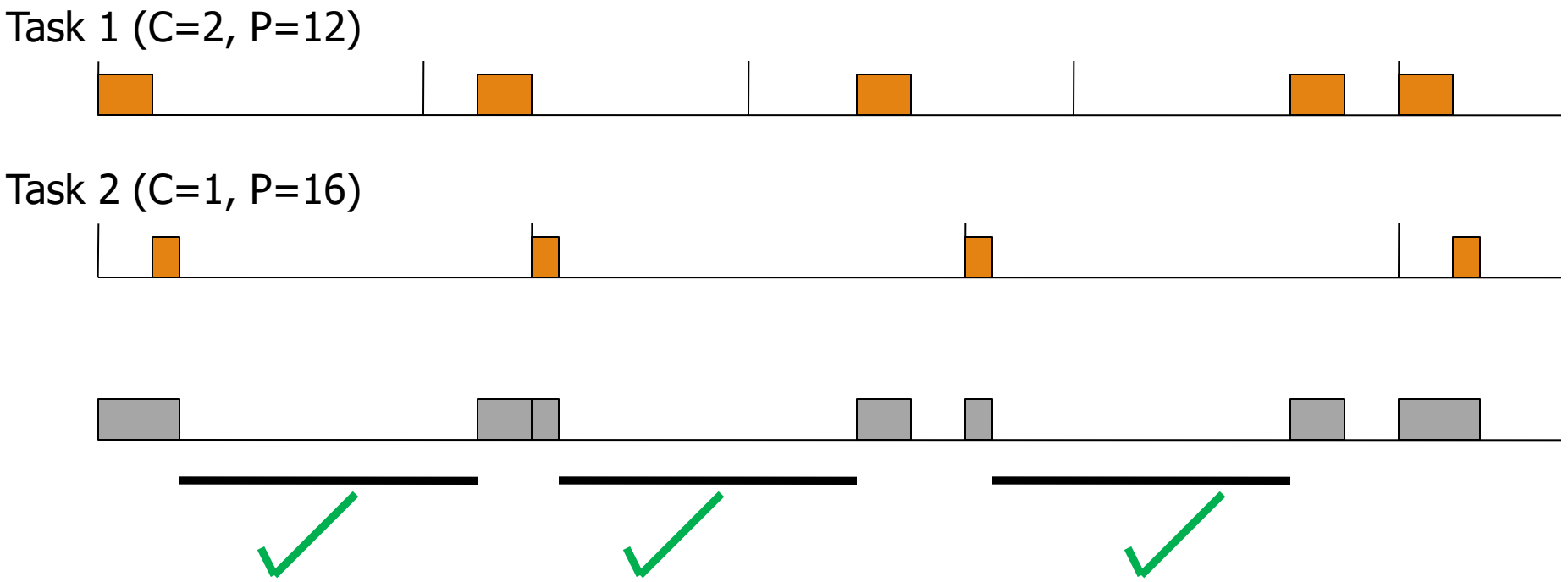
DPM and the Problem with Work-conserving Scheduling

No opportunity to sleep ☹



DPM and the Problem with Work-conserving Scheduling

Must batch! ☺



Temperature: Relation of Temperature and Power

The rate of change of temperature is proportional to the difference between input power and output power (via cooling)

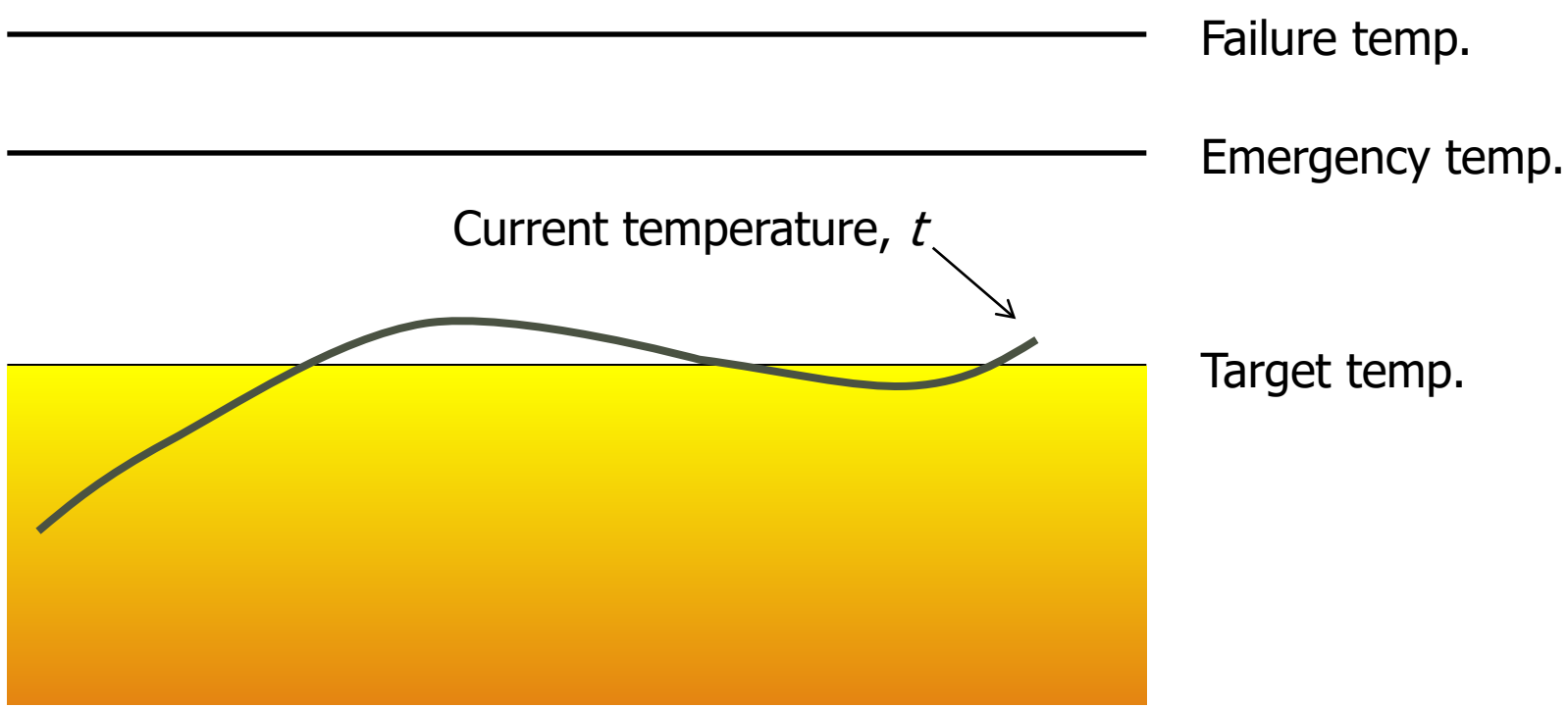
$$\frac{dT}{dt} = P_{in} - P_{out}$$

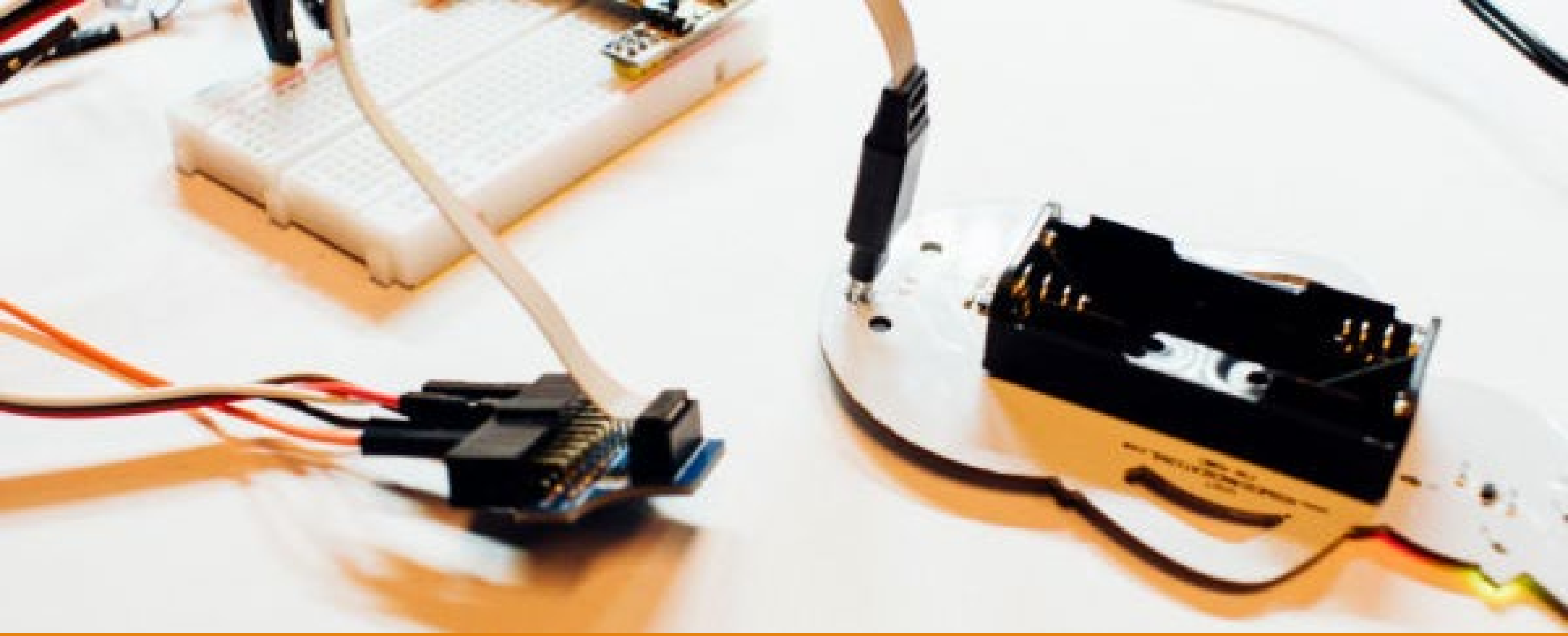
$$P_{in} = f(DVS, sleep)$$

$$P_{out} = g(T)$$

An Extra Constraint: Thermal Throttling

Target temperature, emergency temperature, and meltdown temperature:



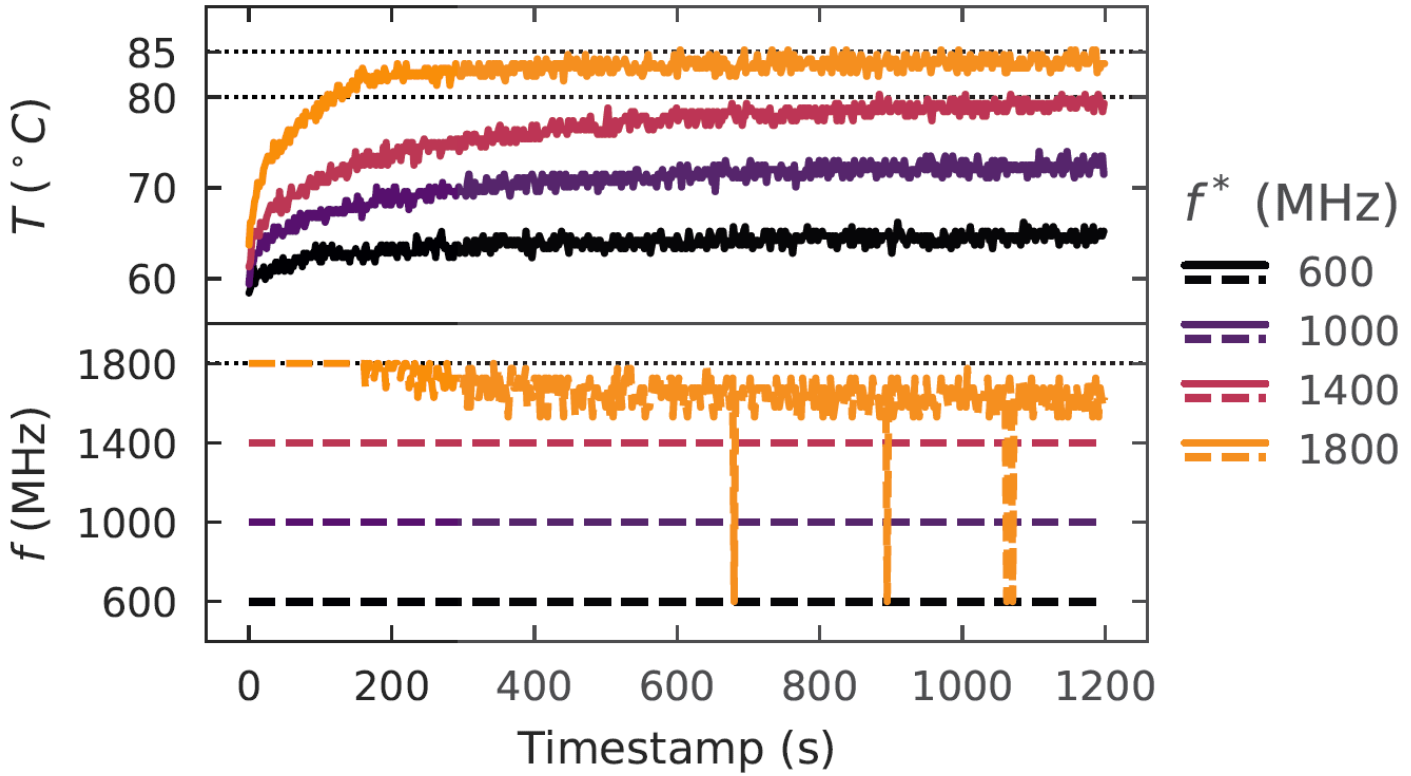


Real-time AI, Power, and Thermal Management

Putting it all together

Temperature Issues and Thermal Throttling

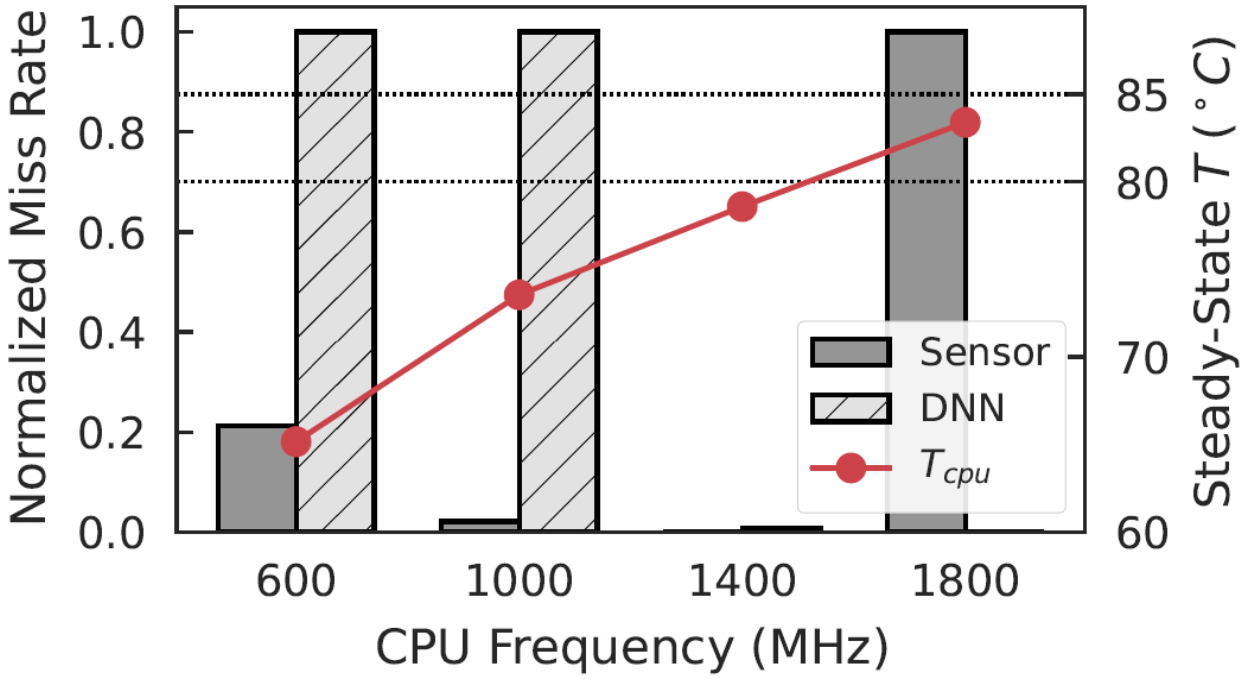
When the CPU is set to a higher frequency, firmware-level thermal throttling reduces its availability for sensor board interrupts, leading to deadline misses. The 1800 MHz curve illustrates this throttling



Timely Execution, Temperature, and Energy Issues

Normalized deadline misses for DNN inference and sensor tasks (bar plots), with corresponding steady-state CPU temperature (red curve).

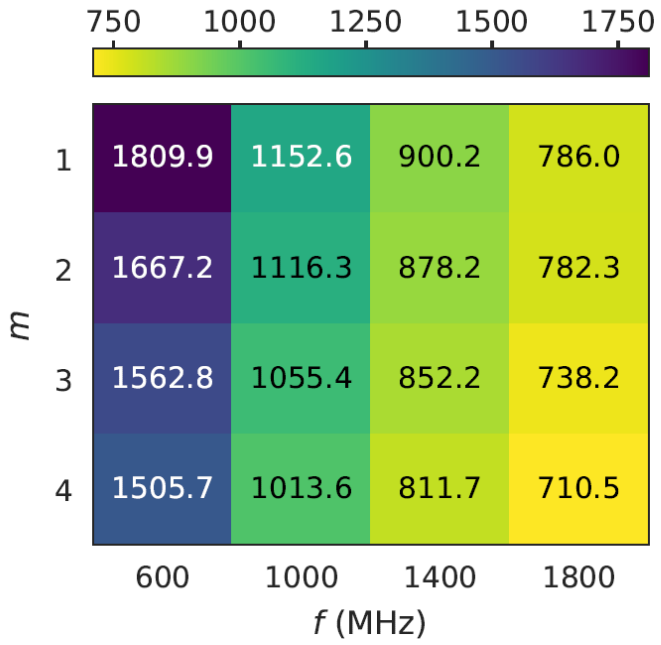
- At low frequencies (600 MHz, 1000 MHz), DNN inference always exceeds its 1s deadline and sensor tasks may miss deadlines due to slow interrupt servicing
- At moderate frequency (1400 MHz), both see fewer misses.
- At the highest frequency, thermal throttling kicks in, causing an uptick in sensor deadline misses.



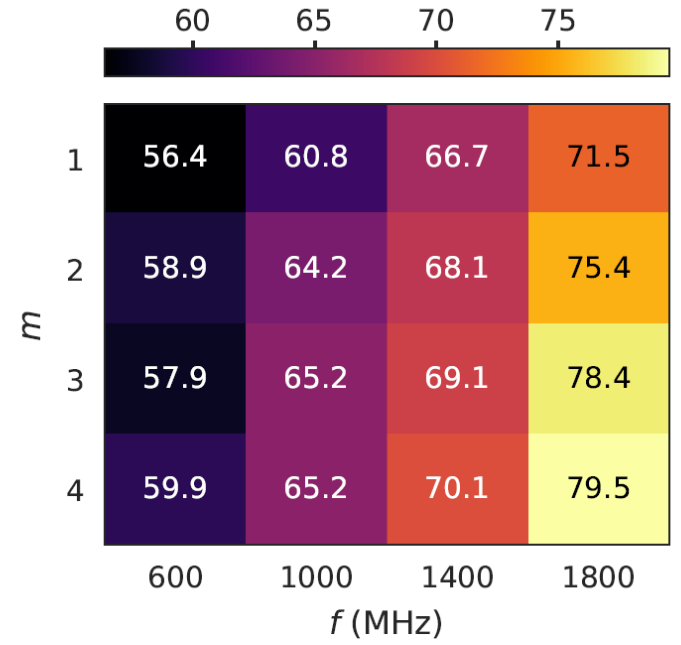
Challenge: Meet Both Time and Thermal Constraints

Two intertwined “knobs” impact timing and power:

- ***Frequency scaling:***
  - Lower frequencies → less power consumption but slower (possible deadline misses)
  - Higher frequencies → faster but more power consumptions (possible thermal throttling and therefore deadline misses)
- ***Number of used cores:***
  - Fewer cores → more deadline misses
  - More cores → more power consumption, hotter (possible thermal throttling)



(a) Execution Time (ms)



(b) CPU Temperature ($^{\circ}C$)

AI Inference in the Heat*

Problem statement:

Given a task set and a multicore processor, jointly determine the optimal number of cores to use for each AI task, together with the frequency settings for the task in order to meet all deadlines and thermal constraints.

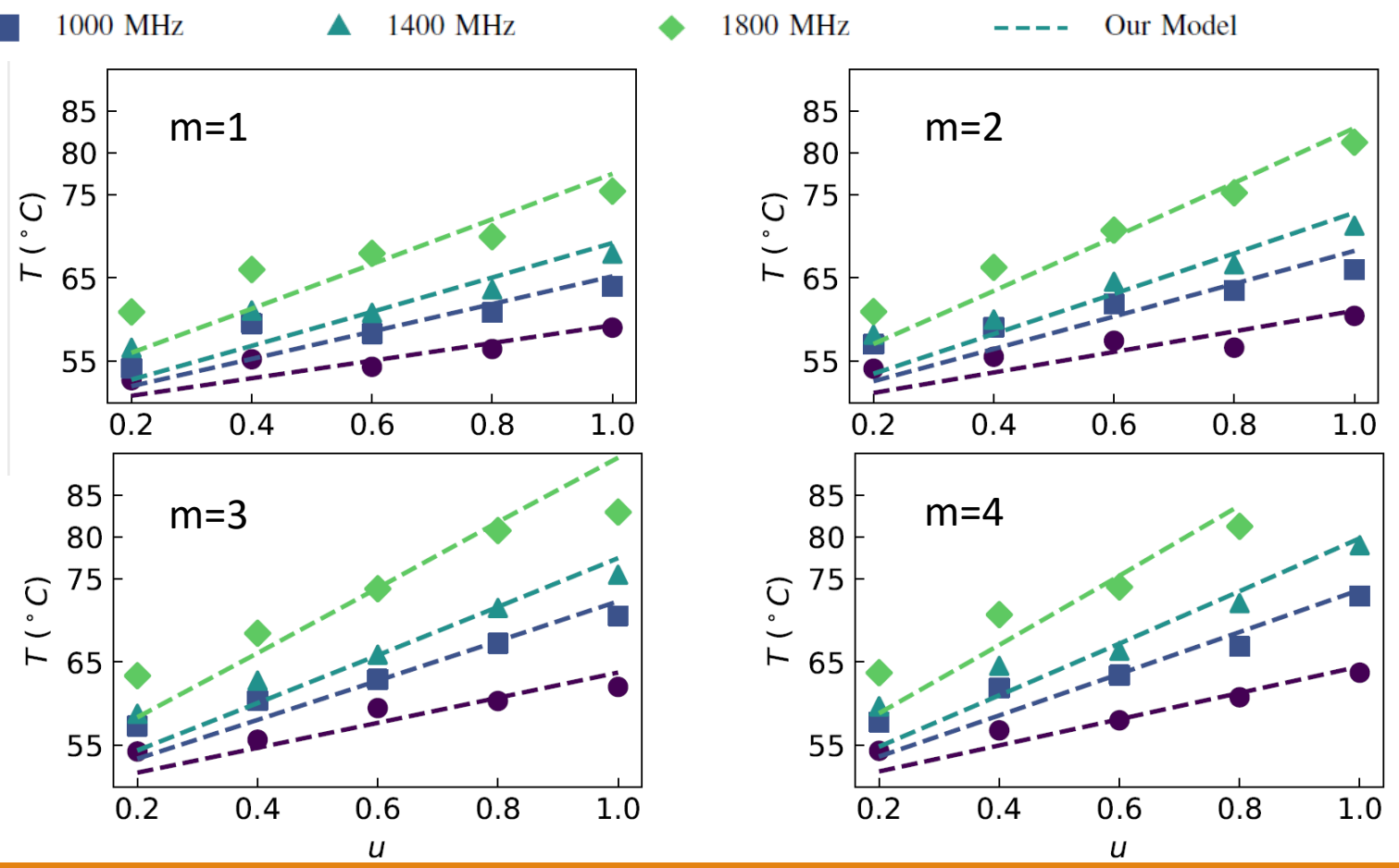
Scheduling assumptions:

- Partitioned scheduling (not global)
- EDF within each partition

(*) Binqi Sun, Jinyang Li, Tomasz Kloda, Tarek Abdelzaher, Marco Caccamo, “AI Inference in the Heat: Thermal-Aware Strict Partitioning for Configurable Real-Time Gang Tasks,” In Proc. IEEE RTAS, Saint Malo, France, May 2026.

Modeling Thermal Properties

Empirical measurements of the effect of the number of cores, m, the frequency, f, and the utilization, u, on chip temperature.



Integer Linear Programming (ILP) Formulation

Minimize energy subject to task partitioning and frequency selection constraints:

$$\sum_{m=1}^{M} x_{k,m} \leq 1, \qquad \forall k \in [M],$$

$$\sum_{k=1}^{M} \sum_{m=1}^{M} y_{i,k,m} = 1, \qquad \forall i \in [N],$$

$$y_{i,k,m} \leq x_{k,m}, \qquad \forall i \in [N], k \in [M], m \in [M],$$

$$\sum_{k=1}^{M} \sum_{m=1}^{M} m \cdot x_{k,m} \leq M,$$

$$\sum_{i=1}^{N} \sum_{m=1}^{M} y_{i,k,m} \cdot U_{i,m,f} \leq 1, \qquad \forall k \in [M],$$

$$\sum_{i=1}^{N} \sum_{m=1}^{M} y_{i,k,m} \cdot \theta_{i,m,f} \leq 1, \qquad \forall k \in [M].$$

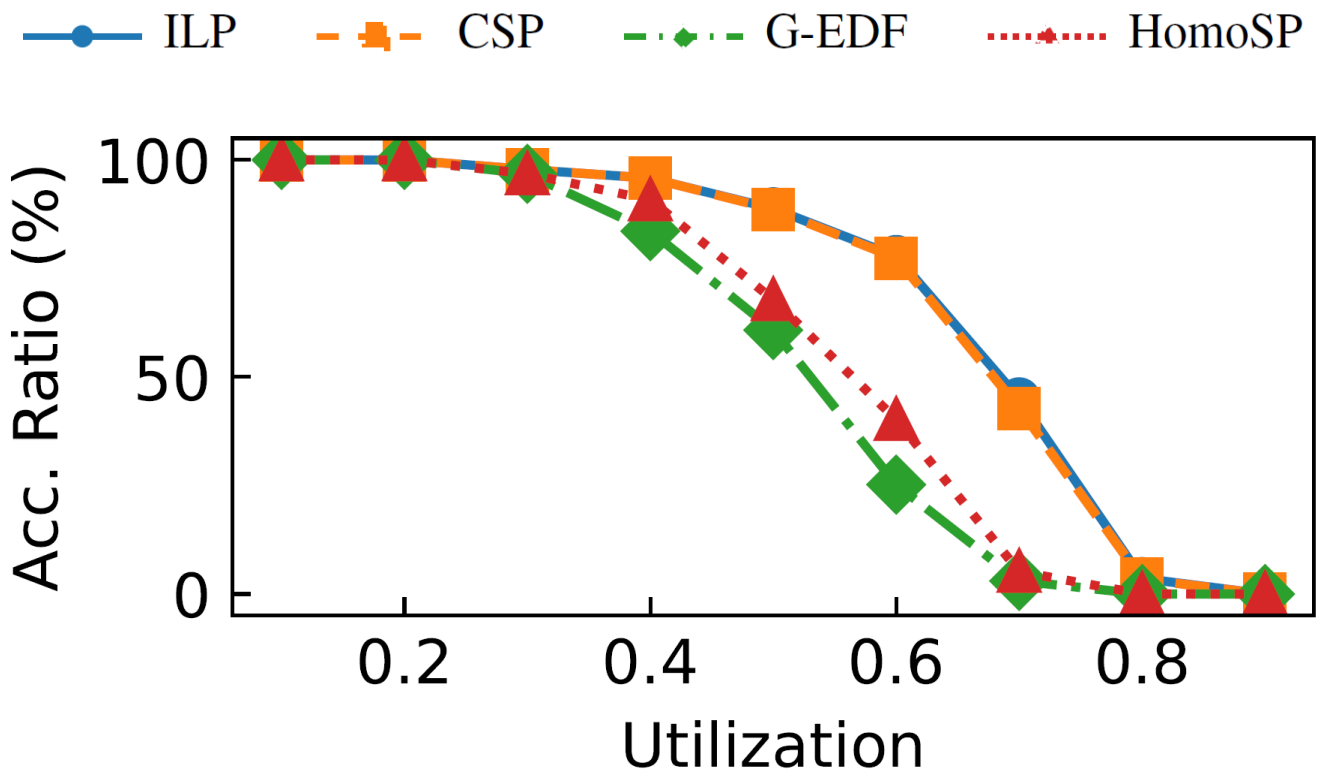
Task Acceptance Ratio

ILP: Integer-Linear Programming
(optimal but computationally heavy)

CSP: Greedy strict partitioning
heuristic proposed in the paper

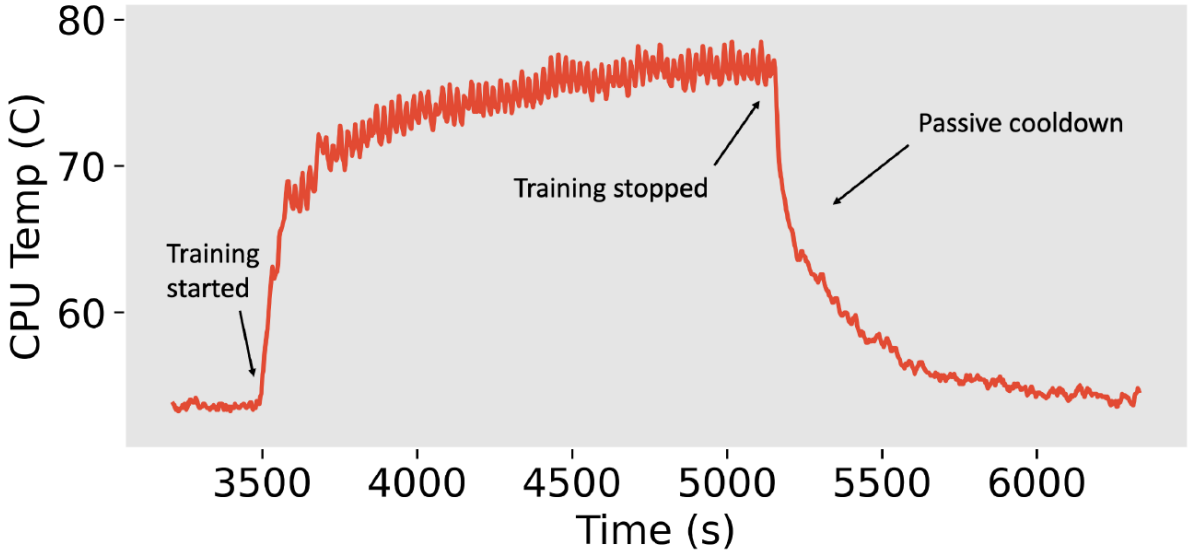
G-EDG: Global EDF

HomoSP: Homogeneous strict
partitioning

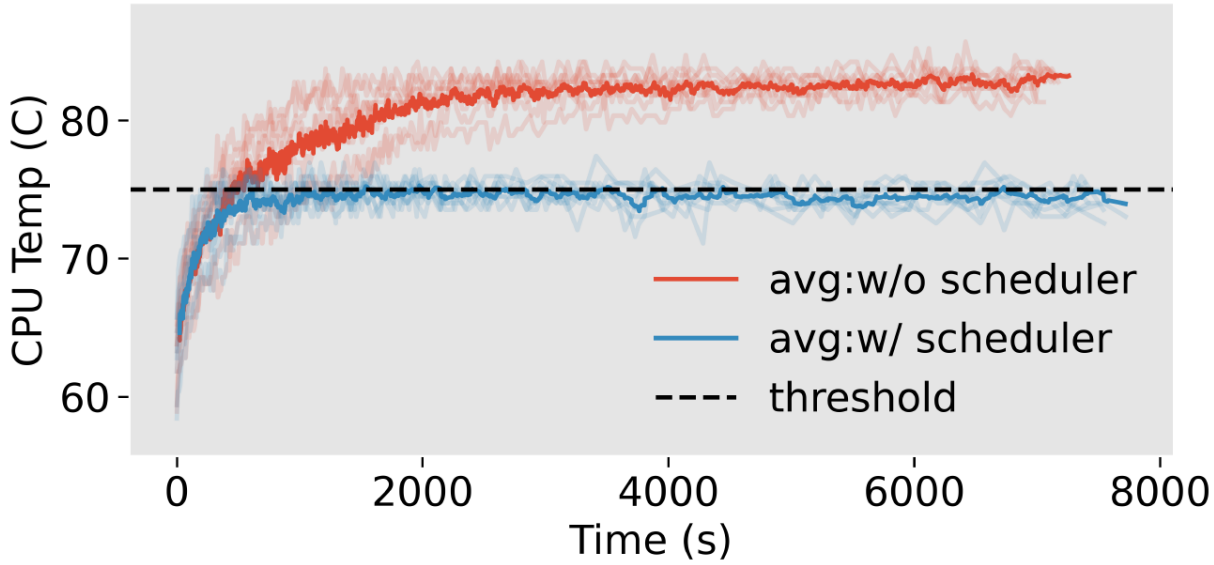


Thermal Effects of an AI Module (on a Raspberry Pi)

The need to perform DVPS on the board creates latency/quality/temperature tradeoffs



Overheating may trigger an emergency shutdown



Temperature control prevents shutdown but increases latency, offering a novel trade-off space

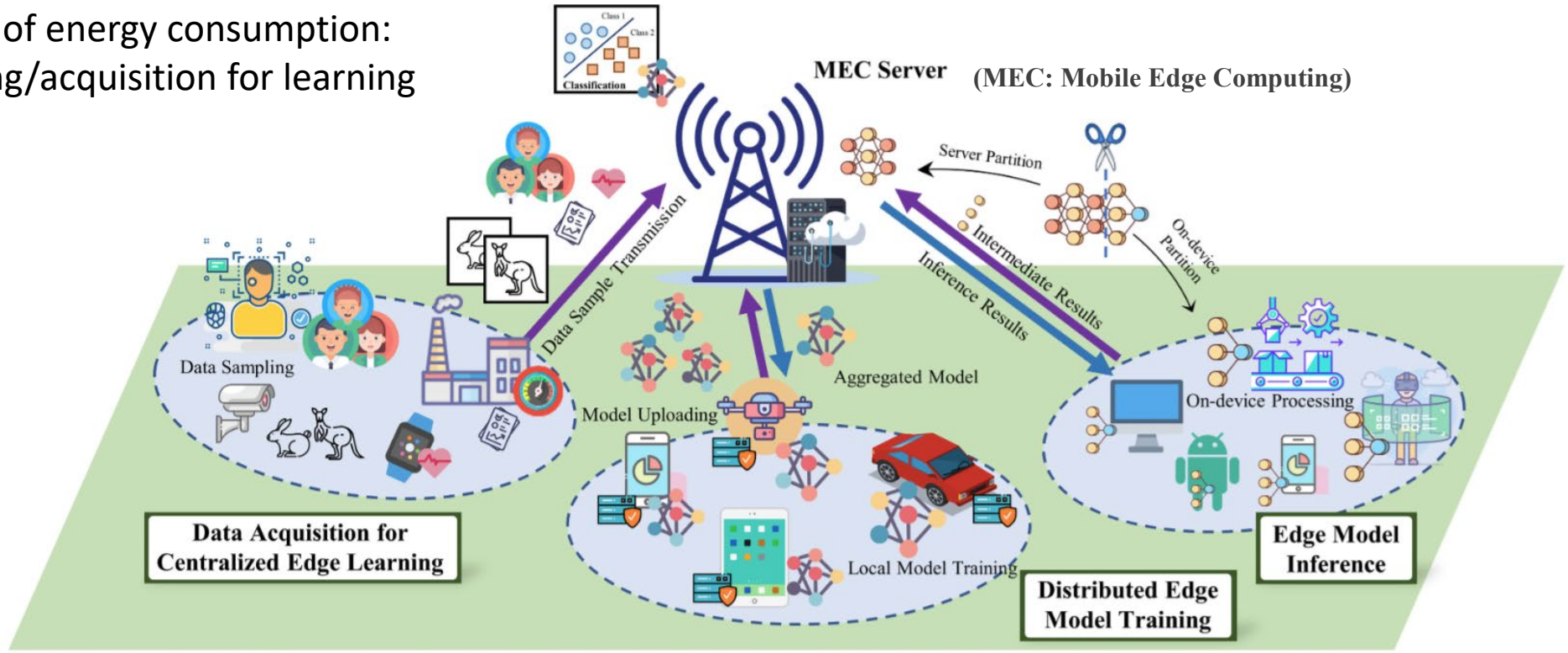
The Problem Space of Energy Saving in AI

Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

A More General Look at Energy in Edge AI

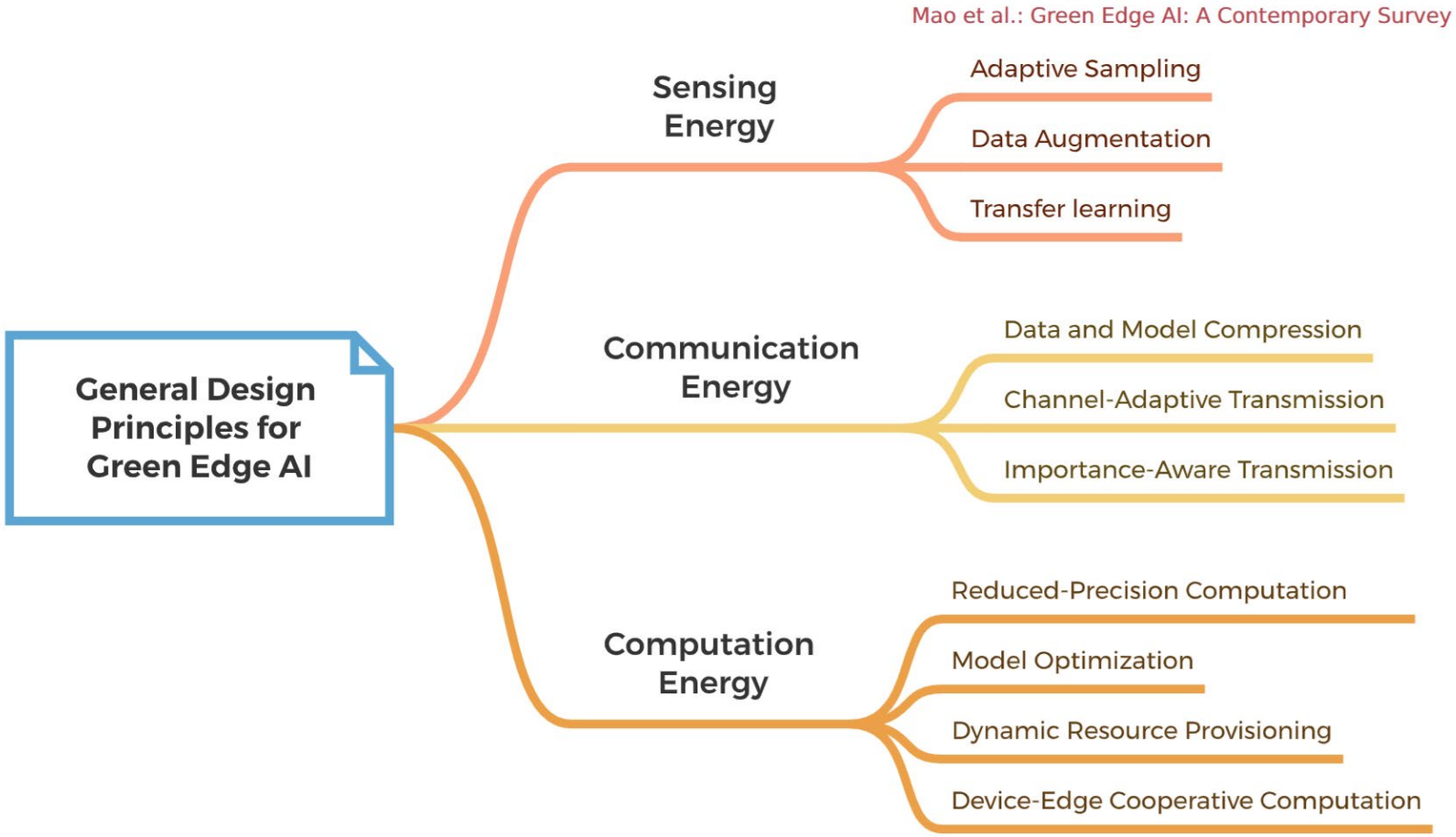
Three sources of energy consumption:

- Data sensing/acquisition for learning
- Training
- Inference



Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

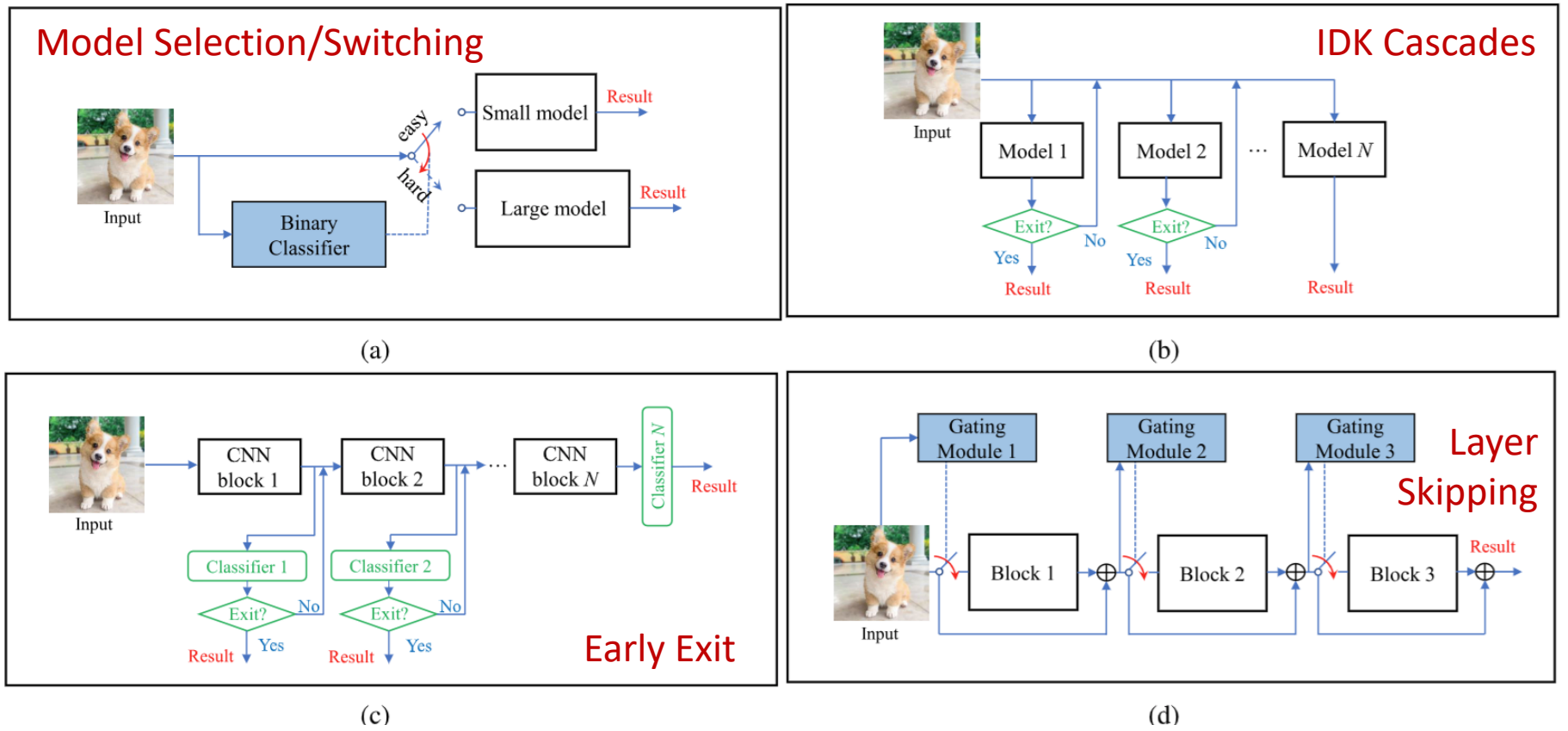
Approaches to Save Energy



Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

On Device Inference Energy Adaptation

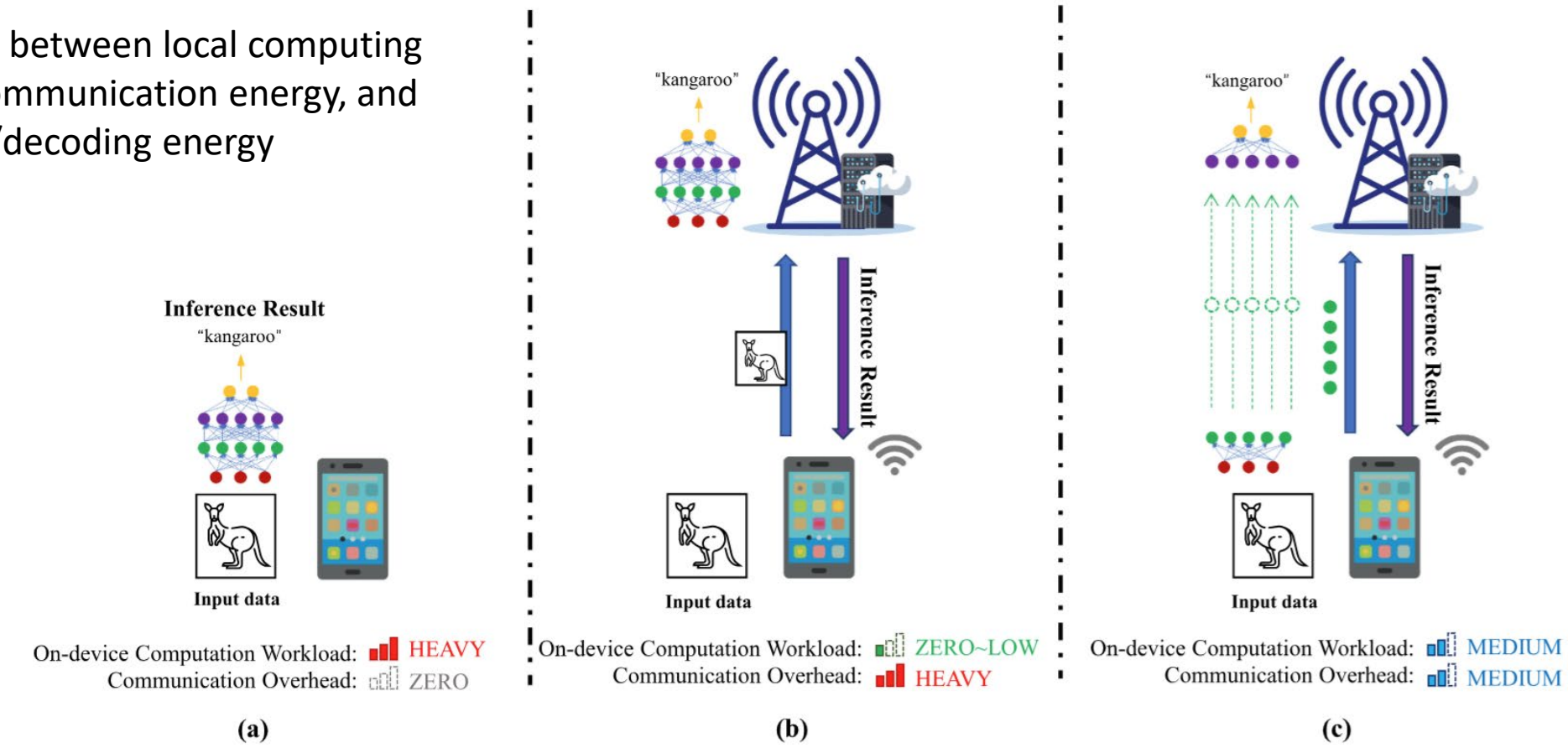
Mao et al.: Green Edge AI: A Contemporary Survey



Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

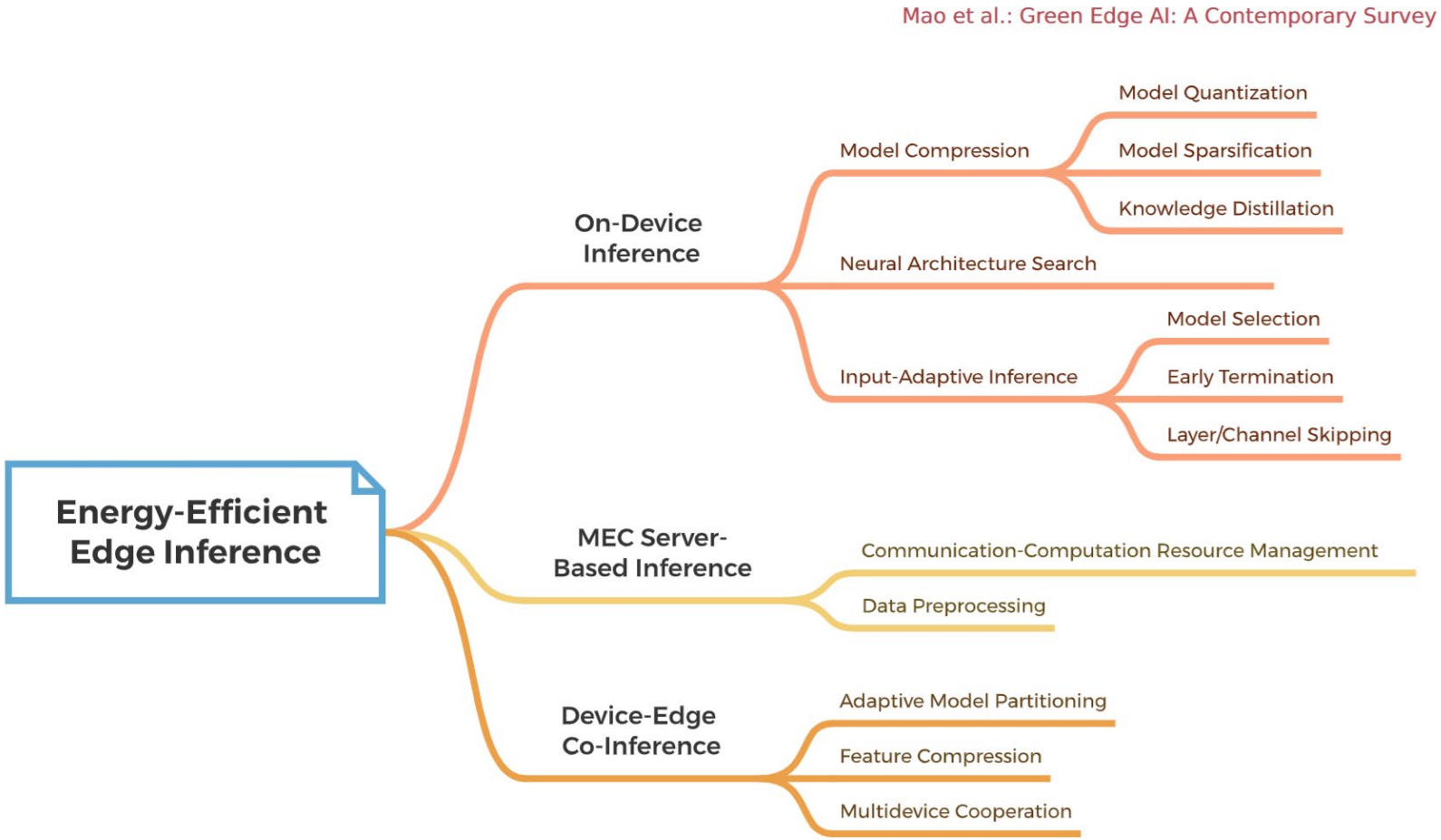
Energy Considerations in Inference Offloading

Trade-offs between local computing energy, communication energy, and encoding/decoding energy



Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

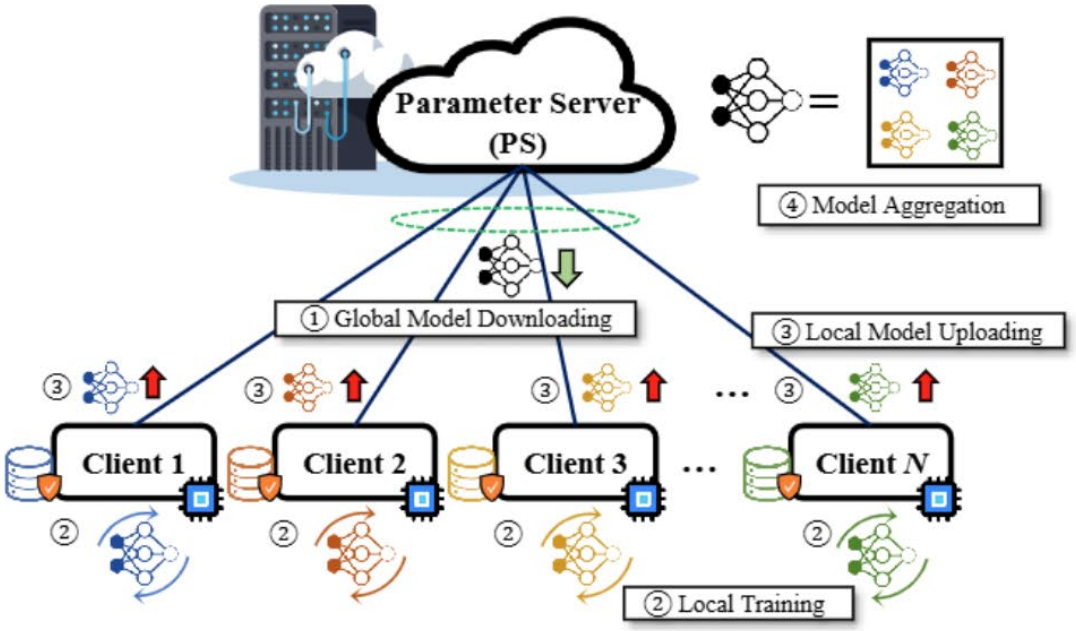
End-to-End Energy Efficient Inference

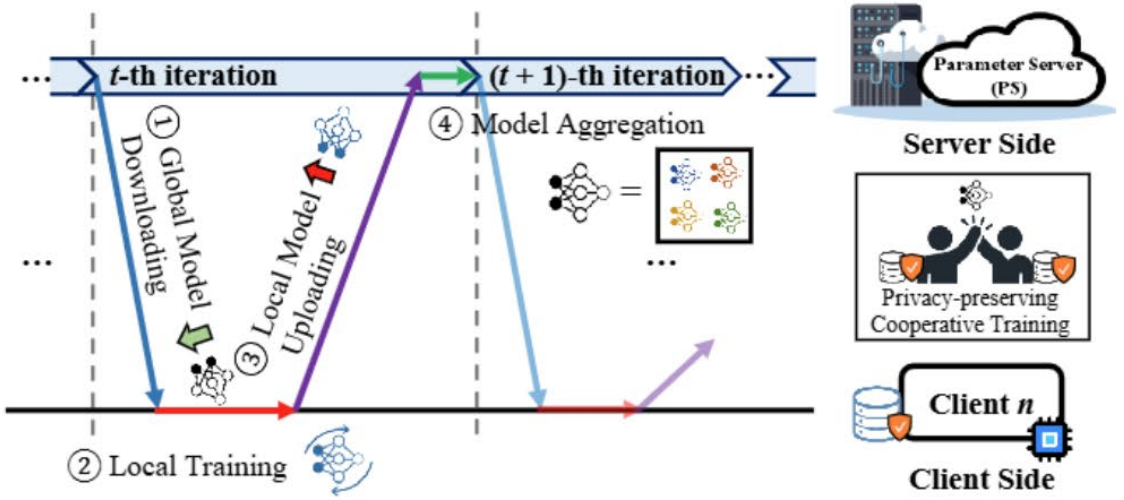


Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

Training Energy (in Federated Learning)

An architecture for federated learning:





Mao, Yuyi, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. "Green edge AI: A contemporary survey." *Proceedings of the IEEE* 112, no. 7 (2024): 880-911.

Energy Considerations in (Federated) Training