Cost-effective On-device Continual Learning over Memory Hierarchy with Miro

Xinyue Ma[1], Suyeon Jeong[1], Minjia Zhang[2], Di Wang, Jonghyun Choi[3], Myeongjae Jeon[1]

UNIST[1] Microsoft[2] Yonsei University[3]

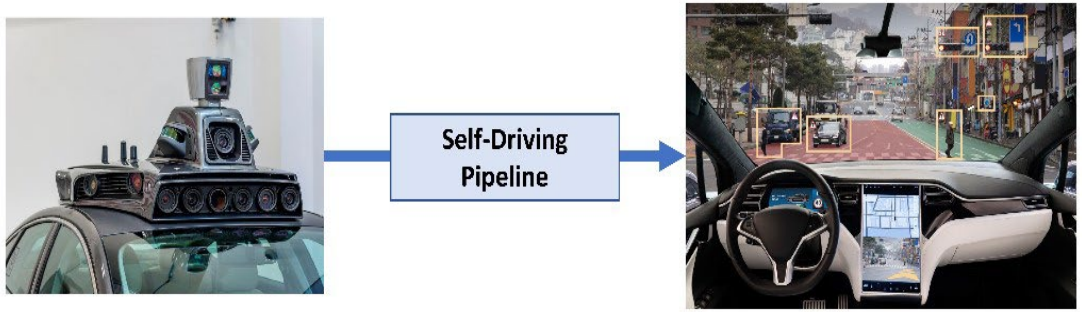
On-device Continual Learning

data drifts: data distribution changes over time, creating unseen tasks

Human activity recognition

New real-life activities and gestures

Video analytics



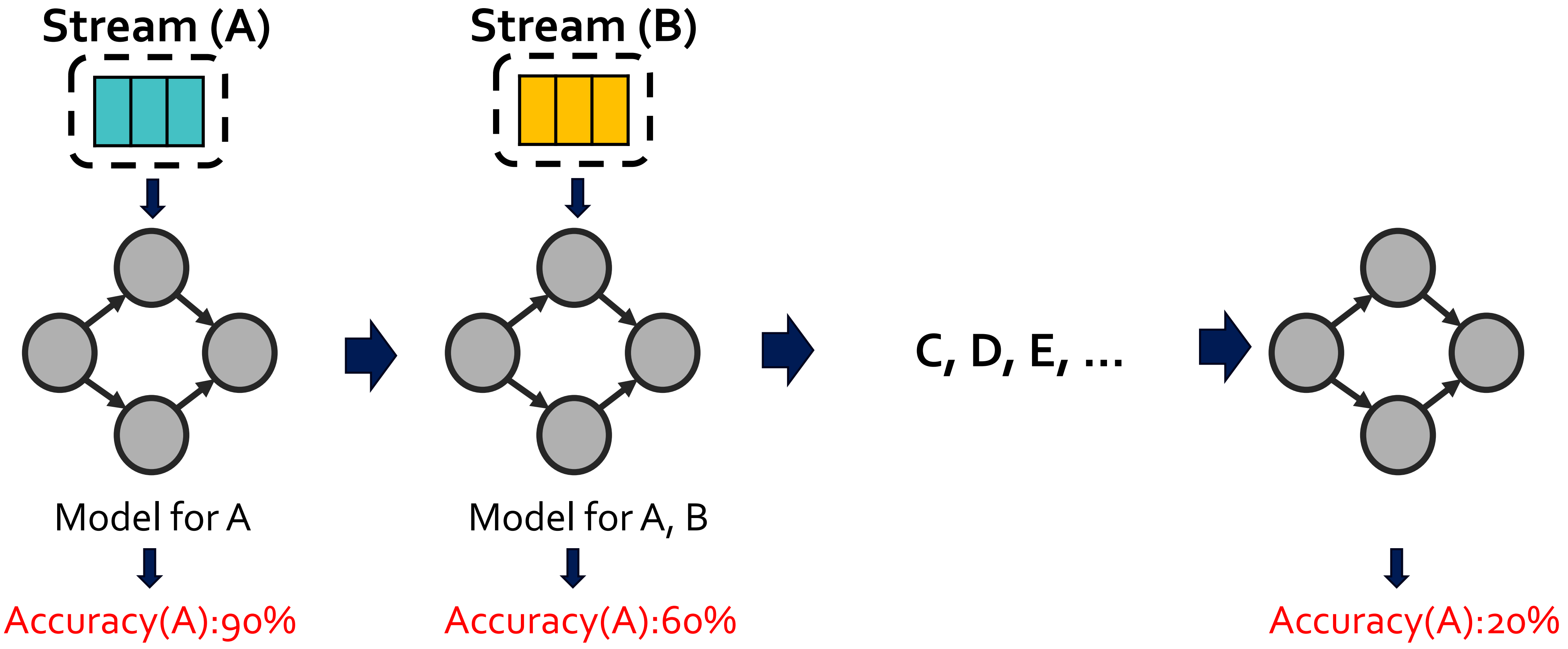
Unseen objects, scenes, and lighting conditions

On-device Learning is essential:

- Protect privacy-sensitive data
- Promptly adapt to new data for customization ➔ Through Continual Learning!

On-device Continual Learning

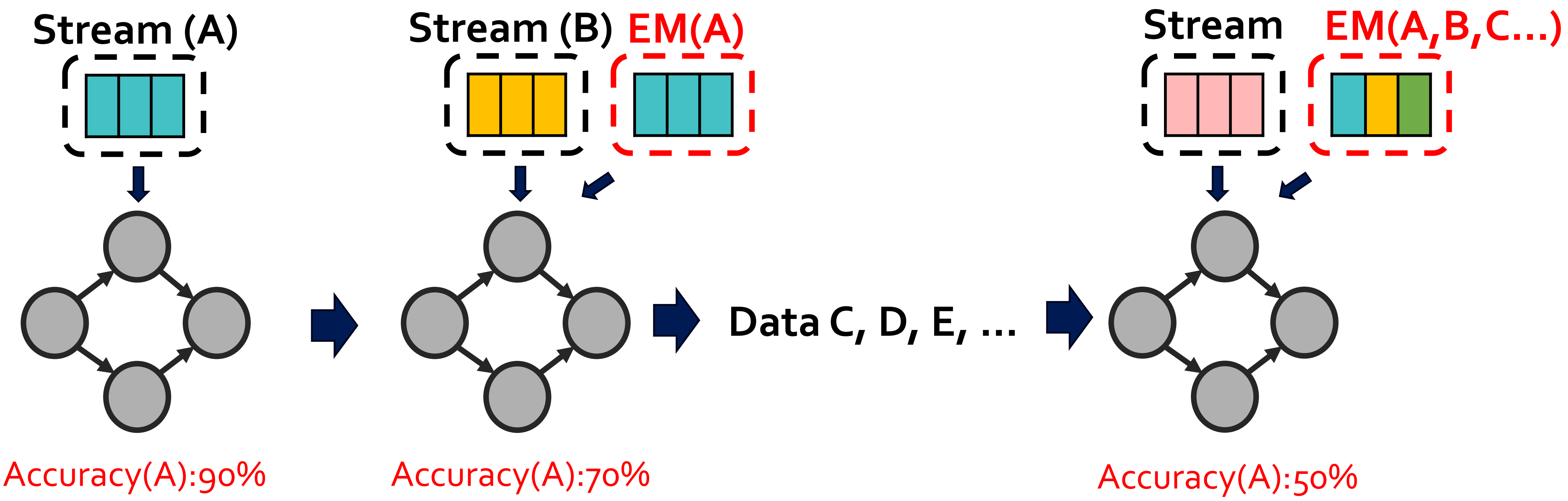
Learning Incrementally as new data becomes available



Forgetting: Previously learned knowledge gradually fades away

Remembering through Episodic Memory (EM)

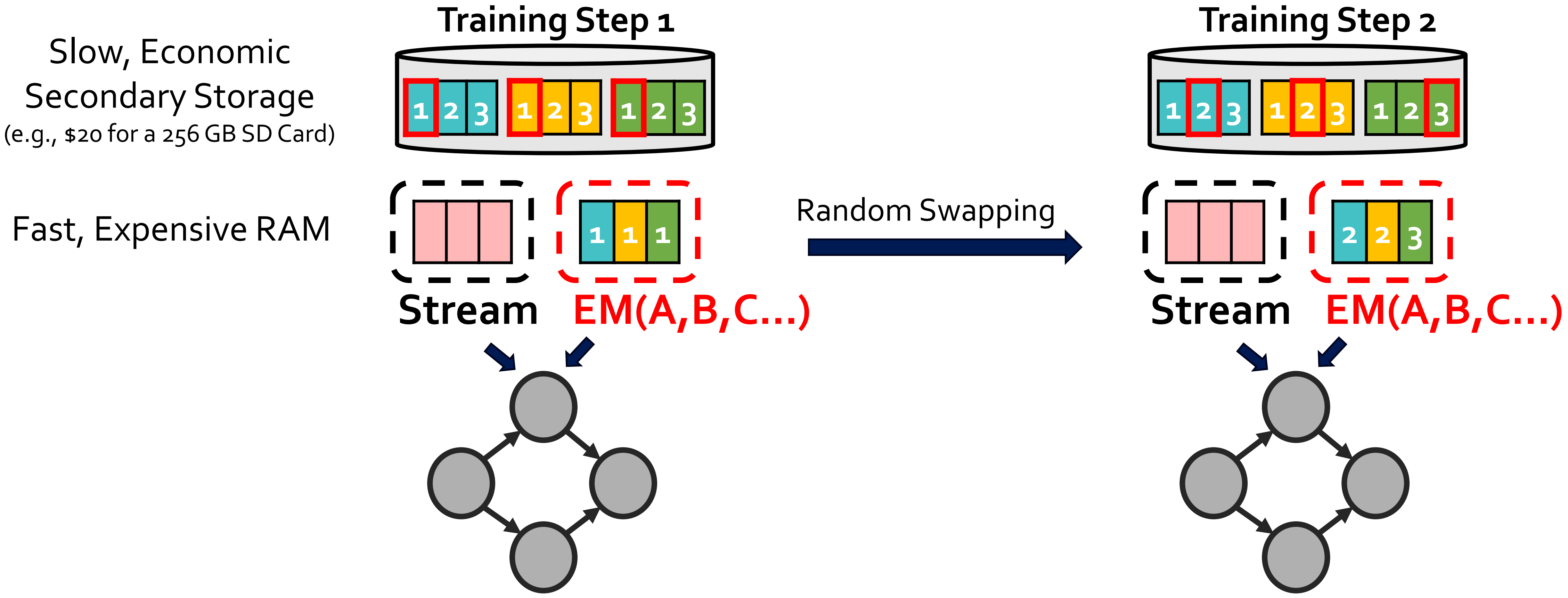
Training on both new and old data



- Designed for **server** computing
- No sufficient consideration on **energy-efficiency**

A System Approach: Hierarchical Episodic Memory (HEM)

Old data stored in RAM and **storage**



Higher *data diversity* ➔ **Higher accuracy**

A System Approach: Hierarchical Episodic Memory (HEM)

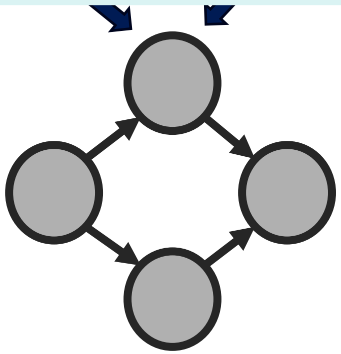
Old data stored in RAM and **storage**

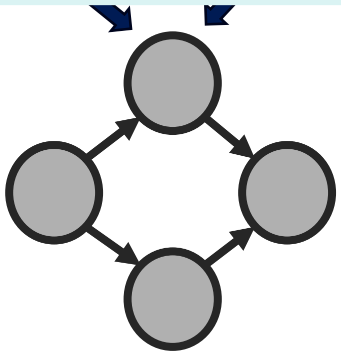
Slow, Economic

Training Step 1

Training Step 2

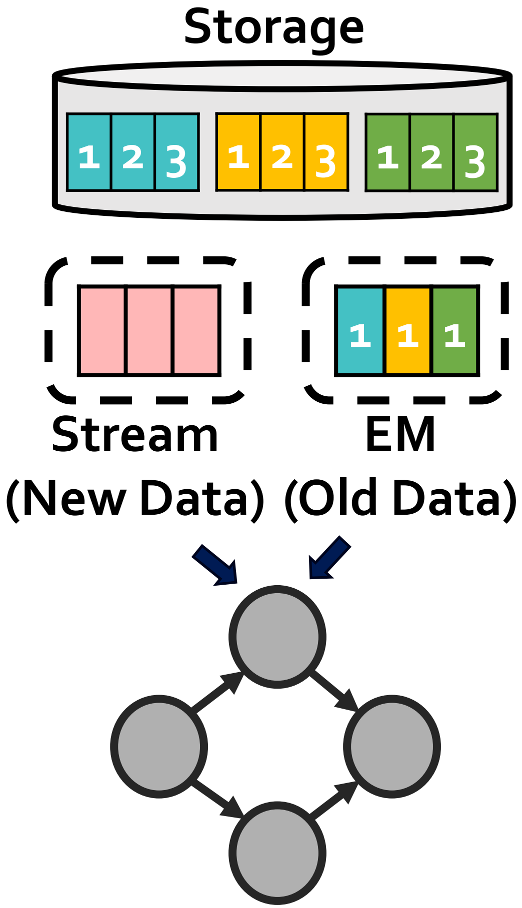
Can we improve HEM at runtime, considering more *system resources* to make it more *cost-effective*?





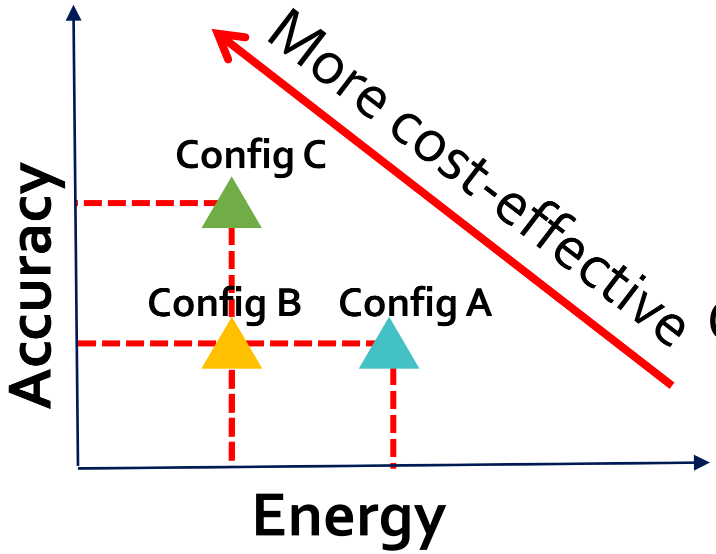
Higher <u>*data diversity*</u> ➔ **Higher accuracy**

Cost-effectiveness in HEM



More resources used ➔ More energy spent

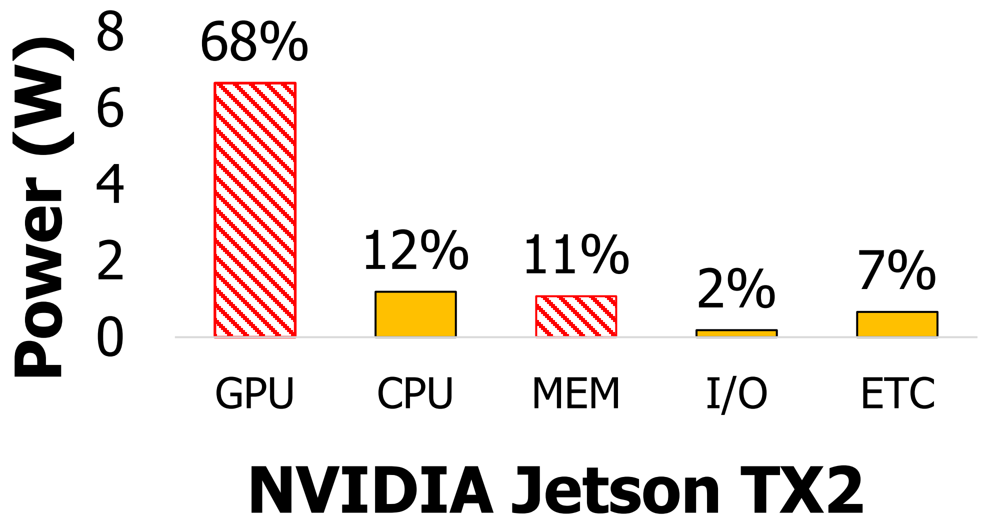
➔ Higher accuracy



Config C > Config B > Config A

The Memory Buffers

Memory Buffers (Stream + EM) is related to GPU and Memory

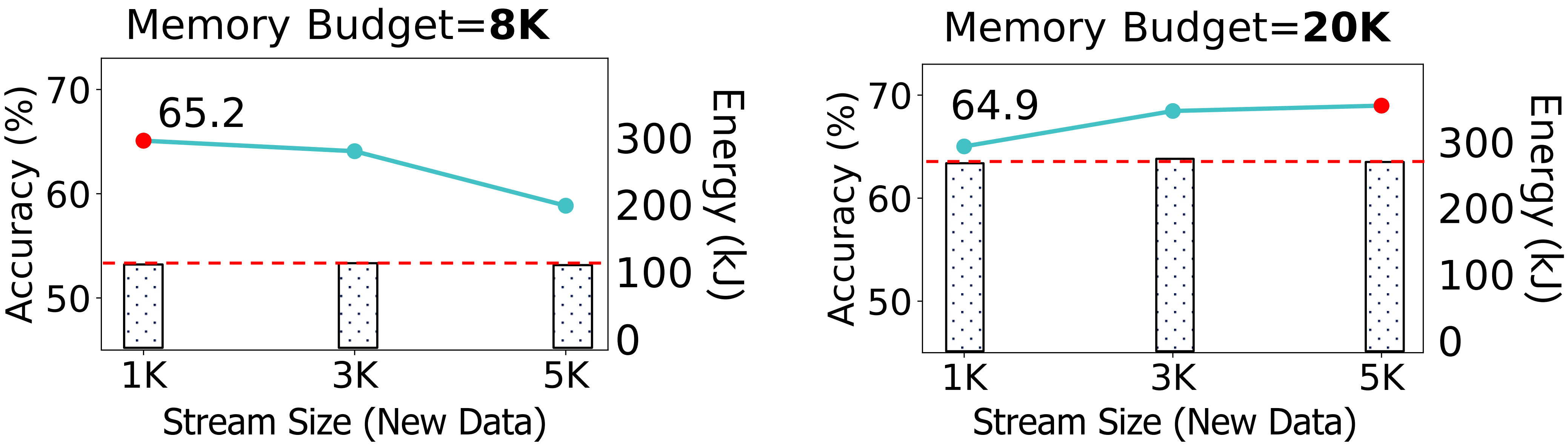


Energy ∝ Memory Buffer Sizes

↳ **different allocations** to Stream and EM for **a fixed budget.**

*Without modifying the model

Better Allocation for a Fixed Budget

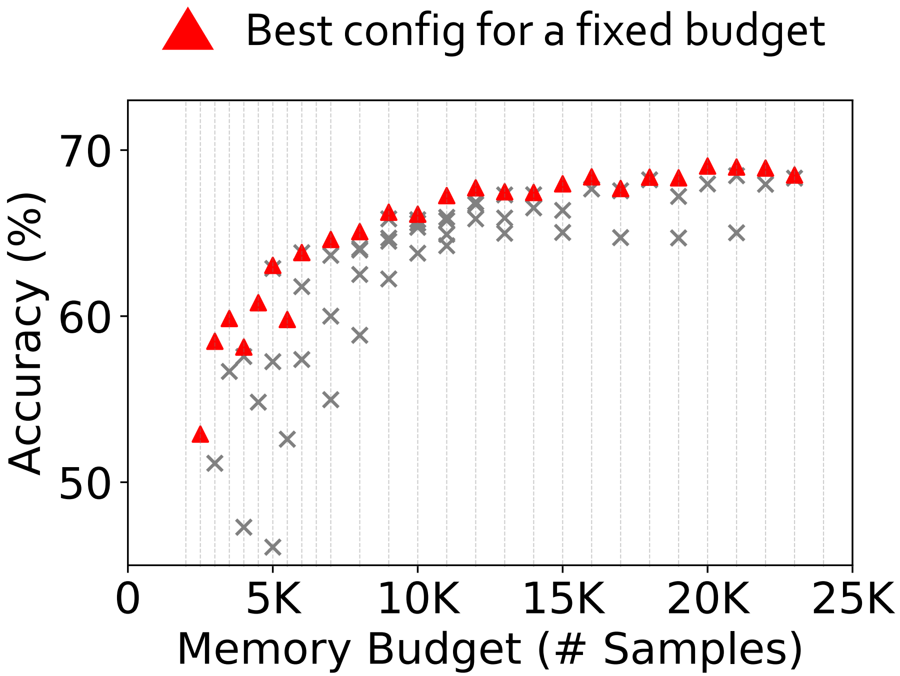


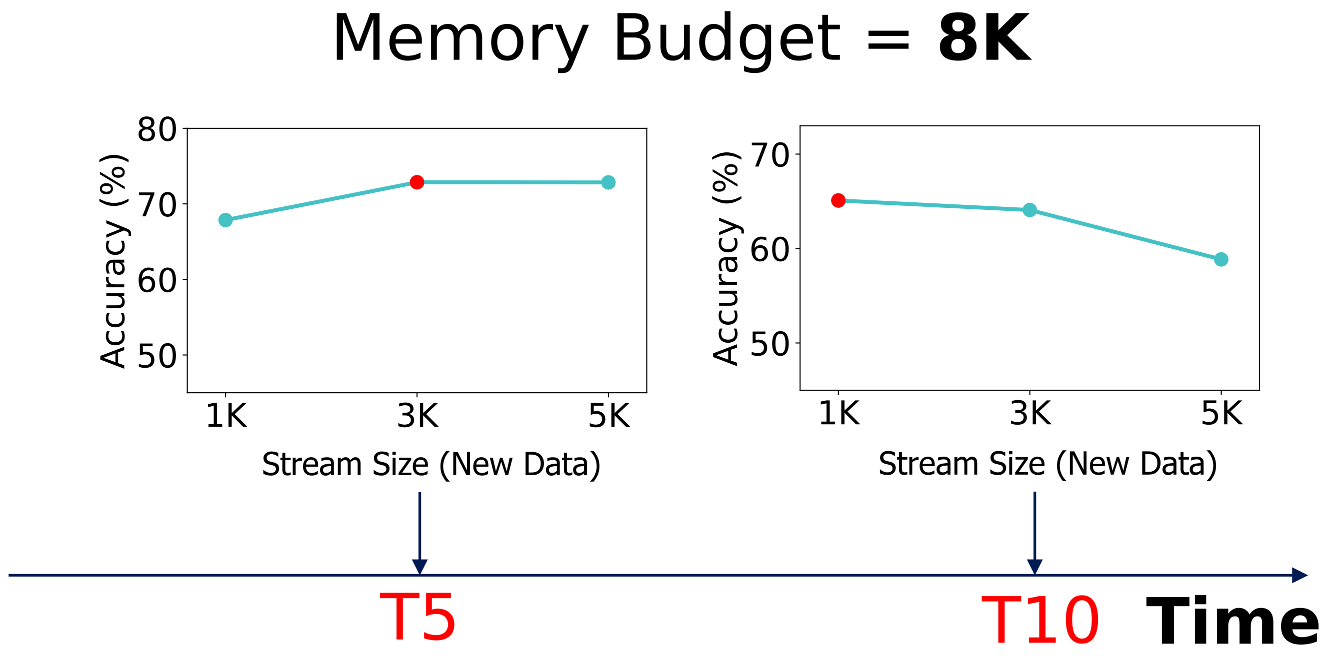
Best Stream Size = 1K

Best Stream Size = 5K

- Best allocation size differs across memory budgets.
- **Larger memory budget does not guarantee better accuracy.**

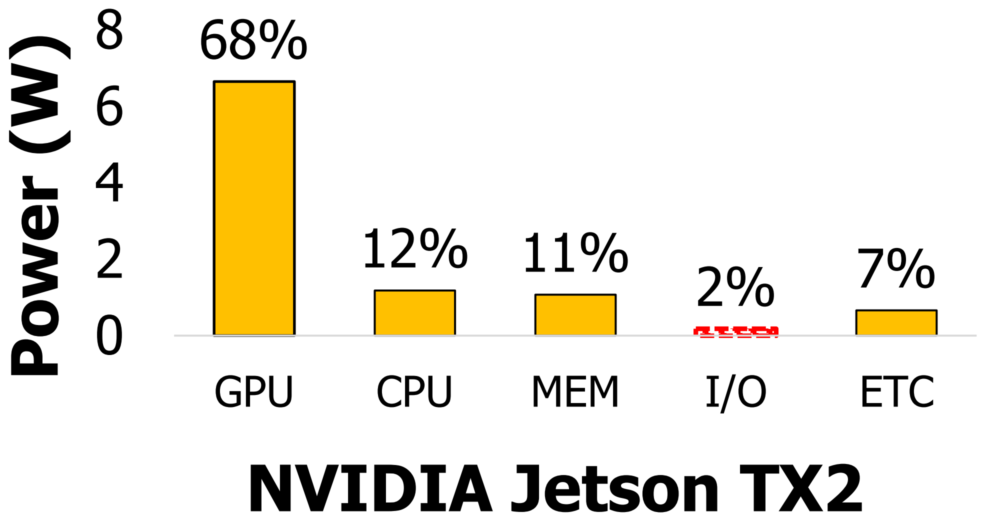
Best Allocation Among All Possible Budgets

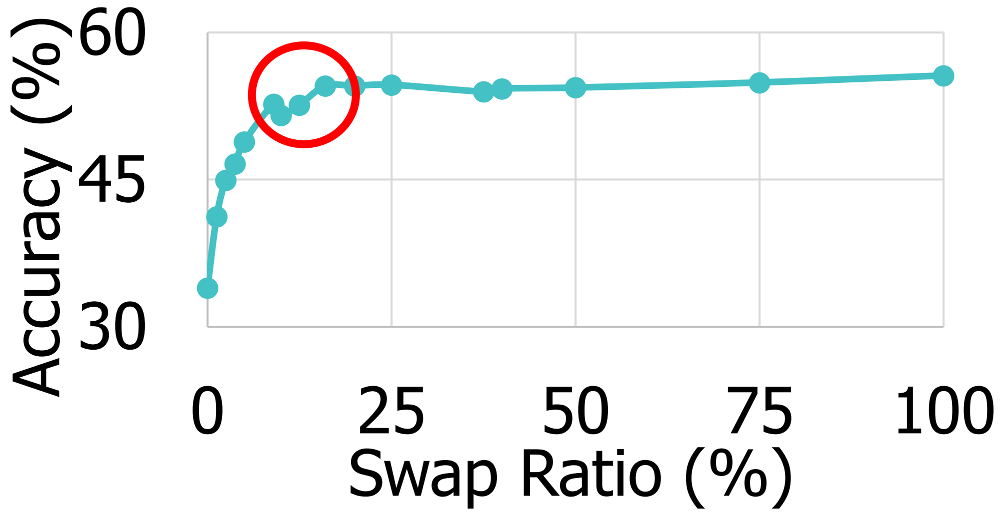




- Some configurations (▲)are better than the others.
- The best configuration changes over time.

The Swap Ratio





Swapping has minimum impact on energy consumption and **a low knee point.**

Turning Observations to Design Principles

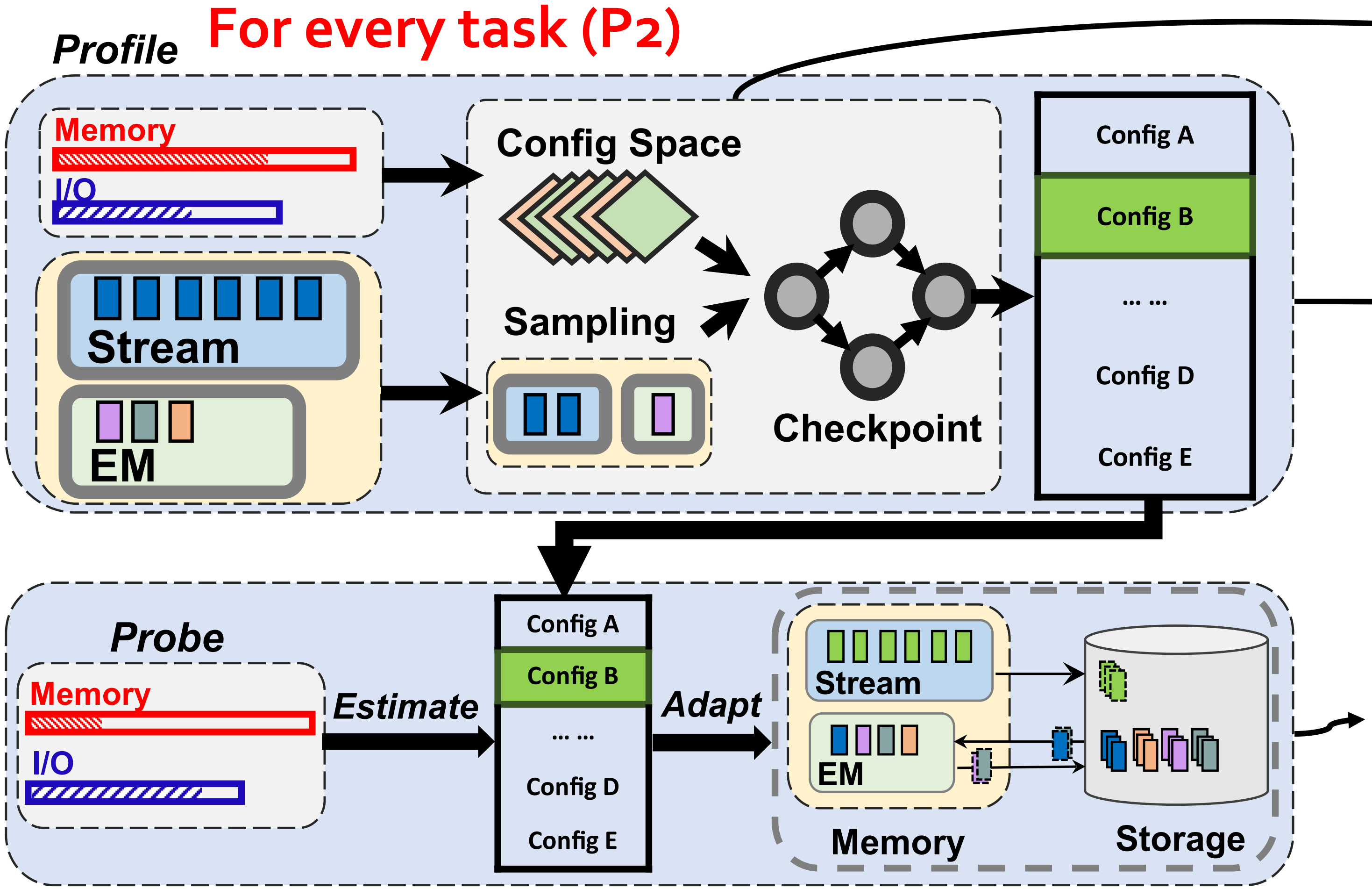
P1. Identifies the best configuration in terms of cost-effectiveness.

P2. Updates the configuration when new tasks arrive.

P3. Configures the swap ratio as high as possible.

+P4. Adapts to changes in resource states in the system.

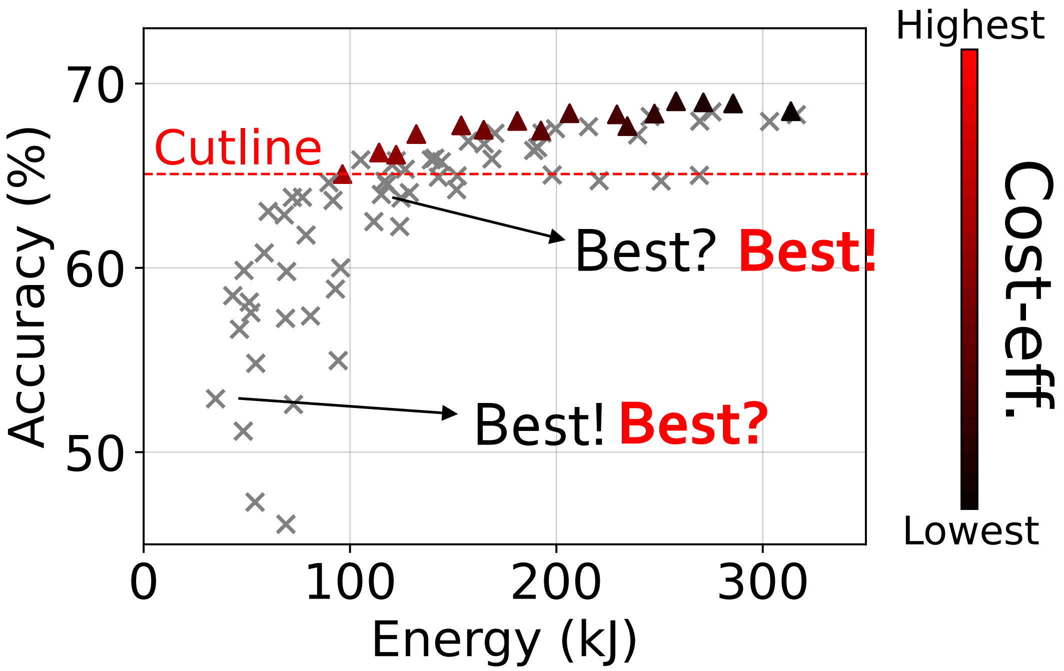
Overall Architecture of Miro



7-10% Overhead

1. Profiling at Low Overhead (P4)
2. Find the best configuration (P1)
3. Configuring Swap Ratio (P3)
4. Adapting to changing resources states (P4)

2. Find the Best Configuration



$$\underline{Cost\text{-}effectiveness} = \frac{Accuracy\ Gain}{Energy\ Spent}$$

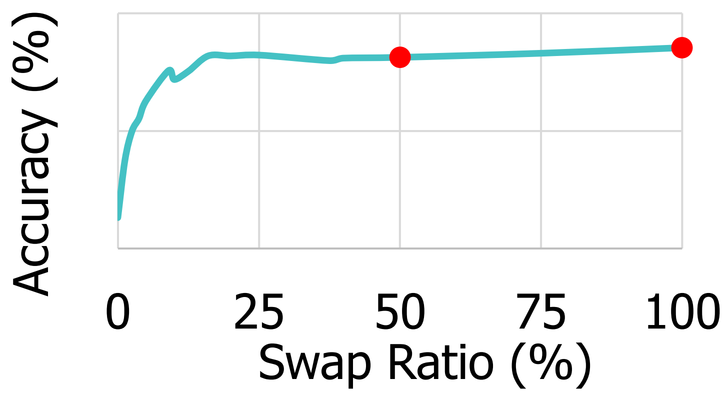
+

$\underline{Cutline}$: Filtering out unpromising configurations

3 & 4. Adapting to Changes in Resource States

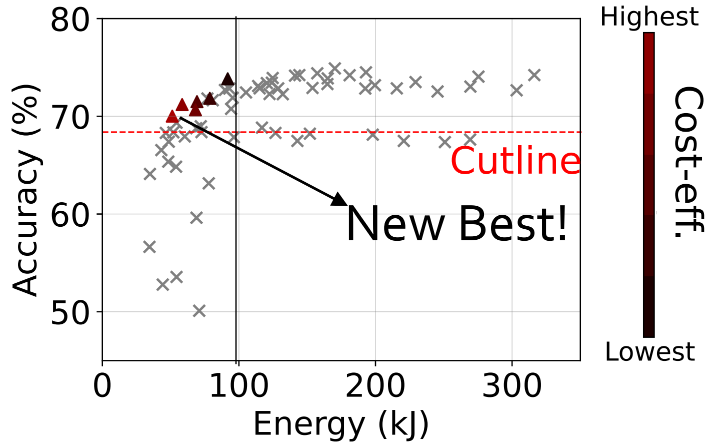
Swap Ratio

I/O Congestion → Halving the swap ratio →



Memory Buffers (SB+EM)

Lower Memory Budget → Blacklist the over-budget configurations



Evaluation - Setup

← Generalizability → ← Scalability →

DATASETS

10 Tasks			20 Tasks	50 Tasks
Image Classification	Audio Classification	HAR	Image Classification	Image Classification
CIFAR100 (50k / 197MB)	UrbanSound8K (8.7k / 184MB)	DSADS (9k / 163MB)	Tiny-ImageNet (100k / 1.1GB)	ImageNet-1K (1.28m / 146GB)

SYSTEM PLATFORMS

NVIDIA Jetson TX2

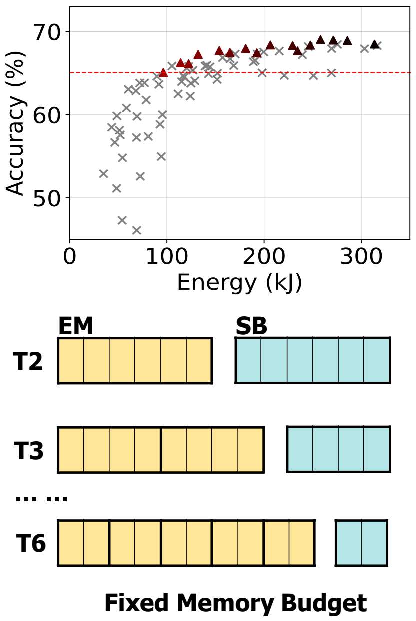
NVIDIA Jetson Xavier NX

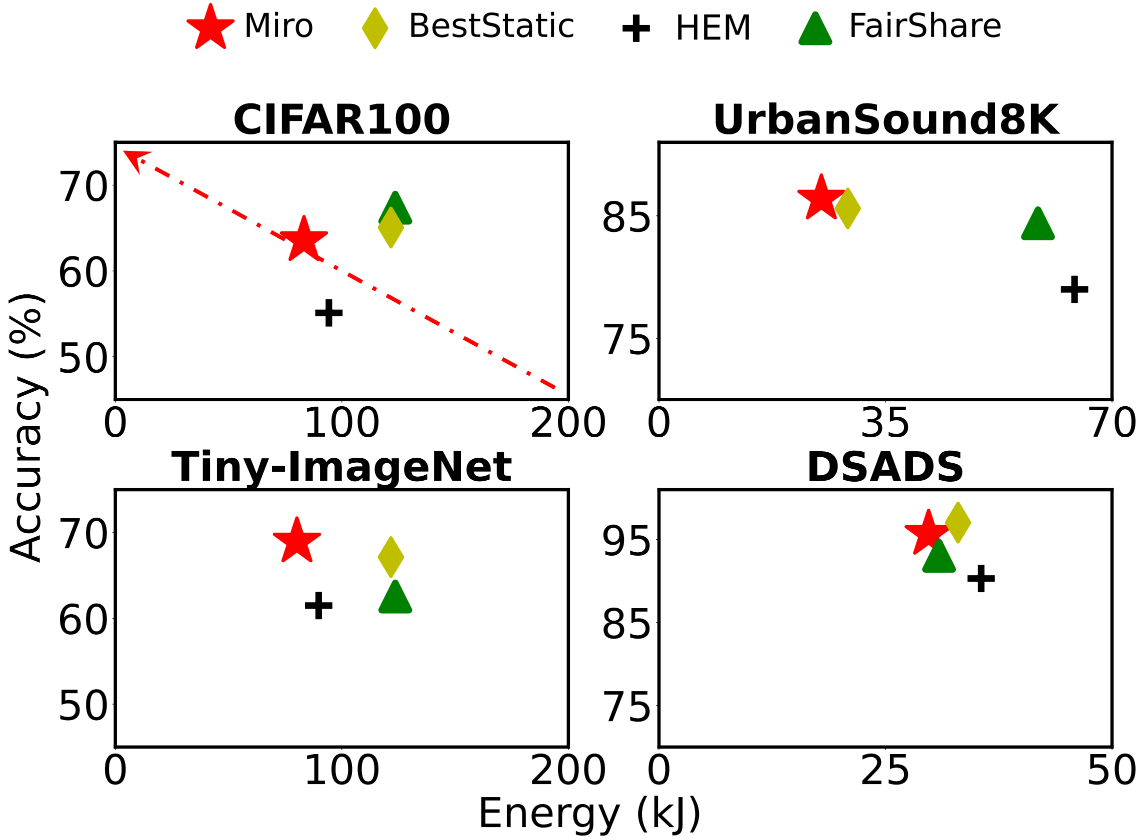
Evaluation – Generalizability

1.HEM Using the default config

2.BestStatic

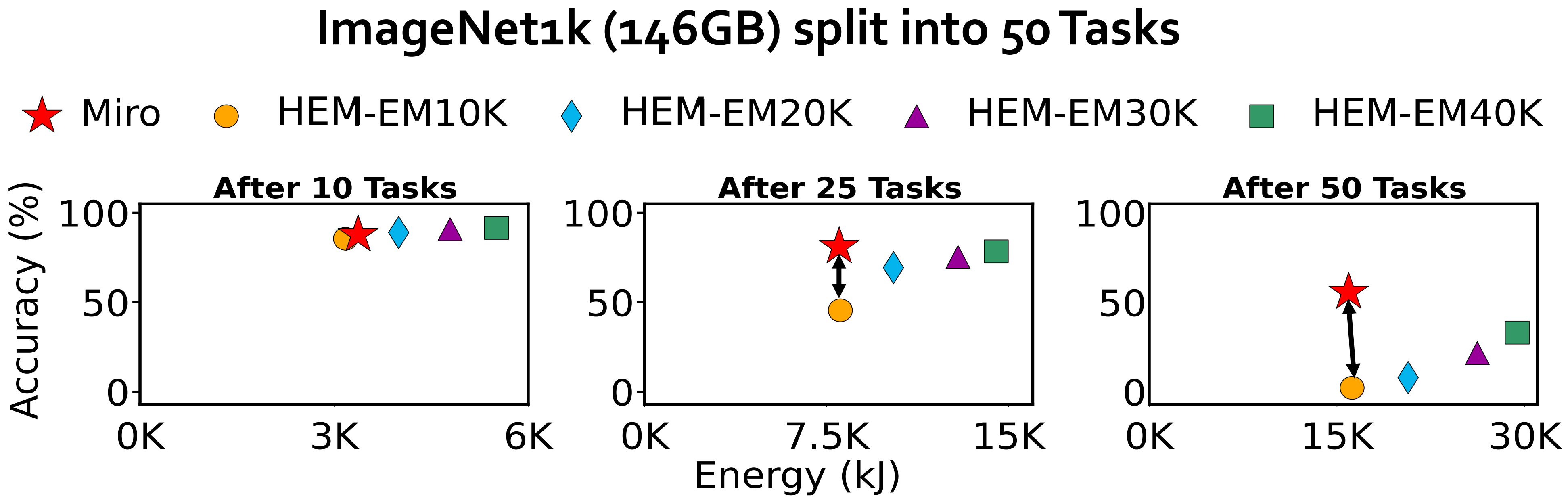
3. FairShare





Miro is more *cost-effective* across the task domains.

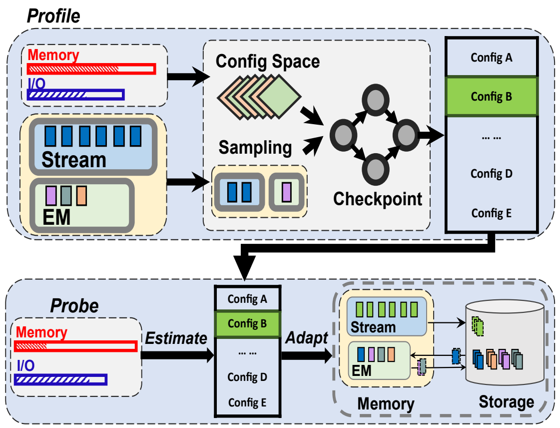
Evaluation – Scalability

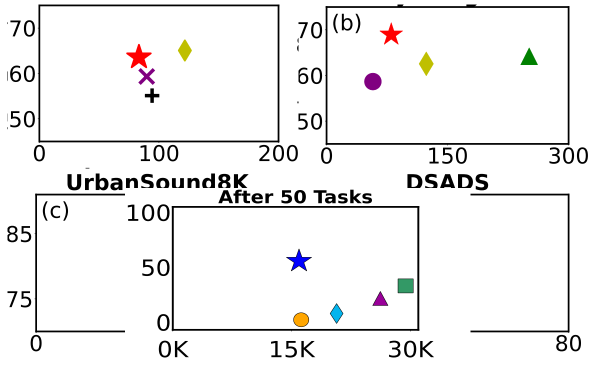


- **Consistently higher cost-effectiveness**
- **Scalability for longer and larger workloads**

Conclusion

- We explored the design parameters of HEM and analyzed their impacts on cost-effectiveness.
- We built Miro, a system runtime that **dynamically** re-configures CL tasks, to achieve **cost-effective** on-device continual learning.
- Miro shows **higher cost-effectiveness** on 5 datasets over 3 task domains, and great **scalability** for longer and larger workloads.





Full Paper

Q&A

Thank you for listening!