SUN YAT-SEN UNIVERSITY

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

香港中文大學
The Chinese University of Hong Kong

Asteroid: Resource-Efficient Hybrid Pipeline Parallelism for Collaborative DNN Training on Heterogeneous Edge Devices

Shengyuan Ye[1], Liekang Zeng[1], Xiaowen Chu[2], Guoliang Xing[3], Xu Chen[1]

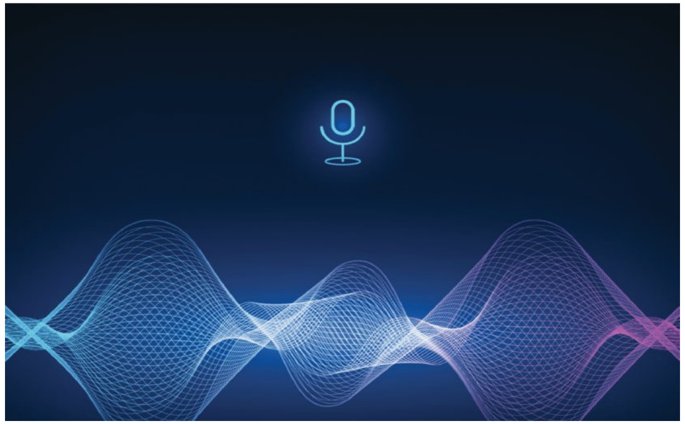
[1] **Sun Yat-sen University**
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] The Chinese University of Hong Kong

acm sigmobile

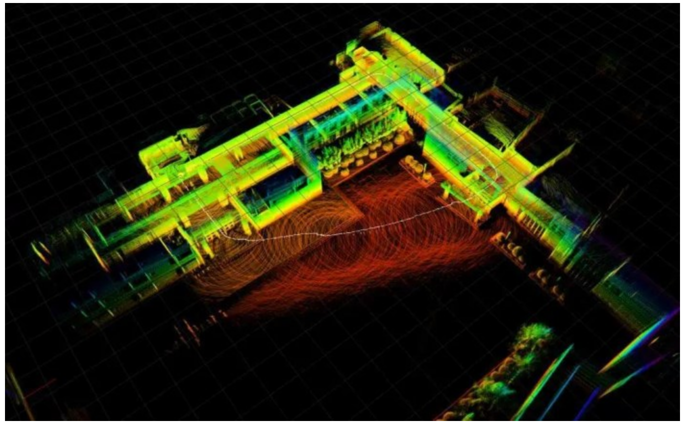
MOBICOM 2024
Nov. 18-22, 2024
Washington, D.C. USA

Intelligent Edge Applications

- **D**eep **N**eural **N**etworks (DNNs) driven increasing intelligent applications.



Personal AI Assistants

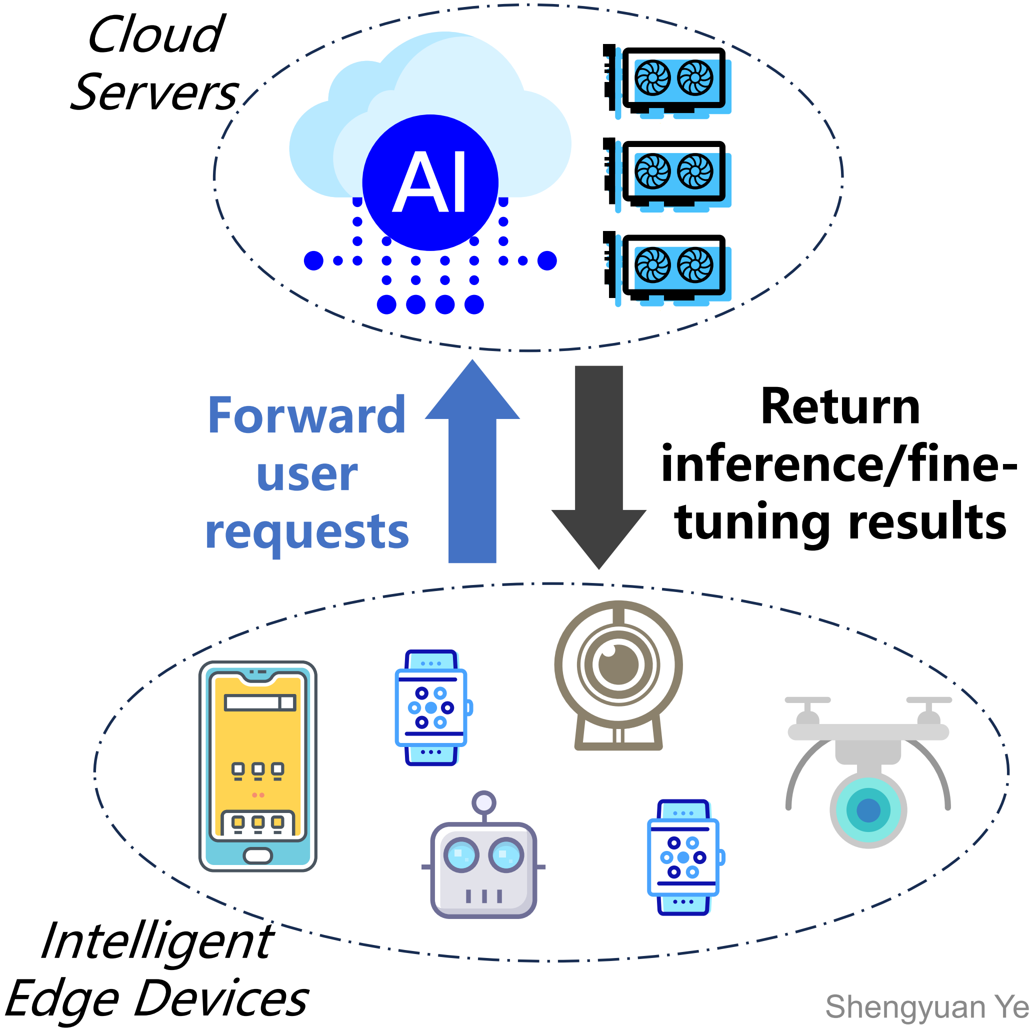


Smart Robotics/UAV

AR & VR APPs

Cloud-Assisted Deployment

- Current DNNs-based applications heavily depend on **cloud services.**



Benefits of cloud deployment:

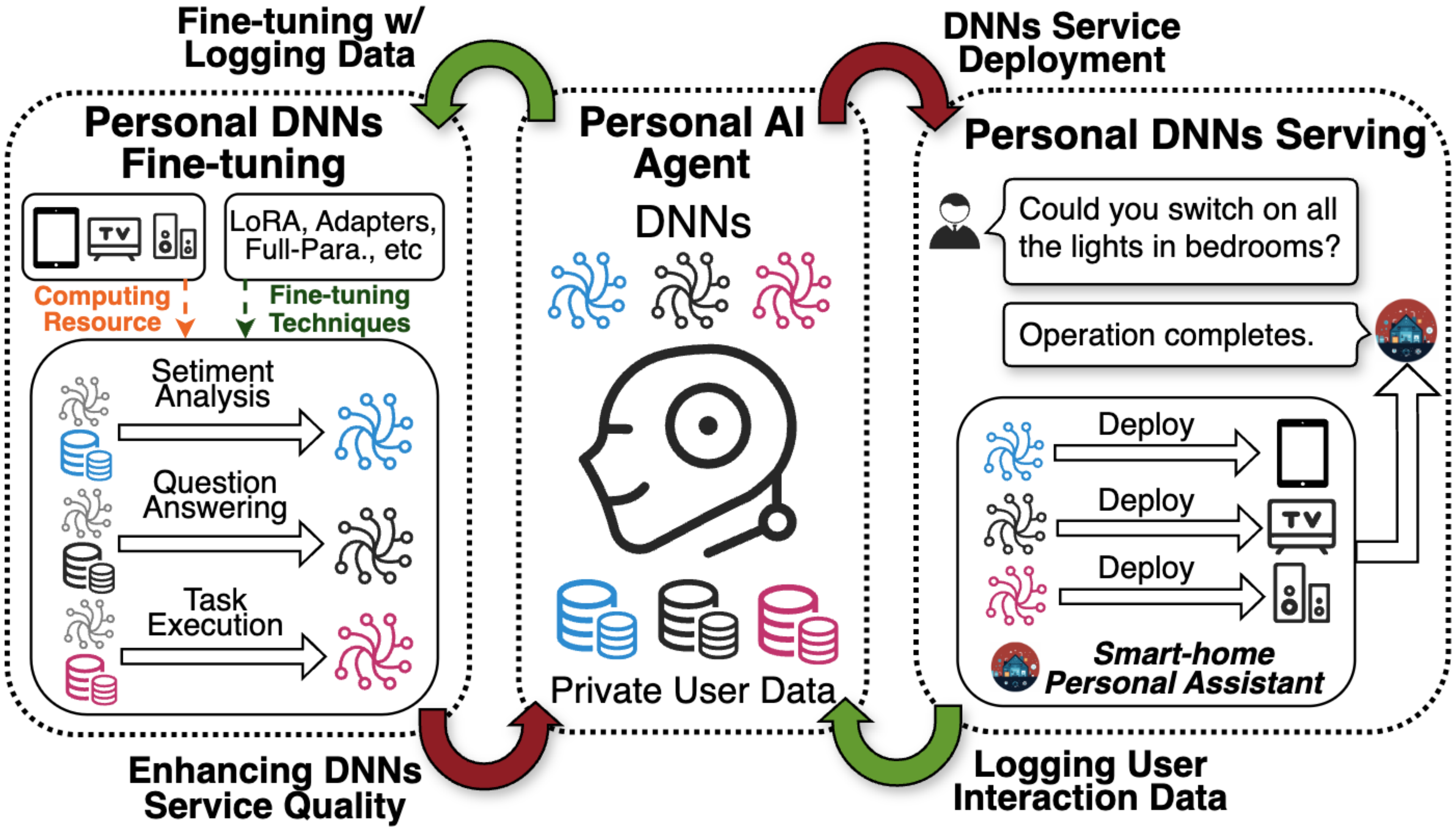
- **Powerful and scalable computing resources.**

Raising three game-stopping problems:

- **Data privacy concerns.**
- **Unreliable WAN connections.**
- **Network and datacenter pressure.**

On-device Deployment

- **On-device deployment** becomes a promising paradigm for intelligent edge APPs.



- **Keep user data in-situ to protect privacy.**
- **Without wide area network transmission.**
- **Alleviating data center pressure by leveraging ubiquitous computing resources.**

Challenges in On-device Training

- **Resource wall** of a single edge device presents challenges for on-device training.

Table 1: Elapsed time of a training epoch on devices.

DNN Model	Average Epoch Time		
	A100	Jetson TX2	Jetson Nano
EfficientNet-B1	10sec	11.2min	26.7min
MobileNetV2	9.4sec	8.5min	22min
ResNet50	65sec	1.14hour	3.48hour

Unbearably prolonged training time

Techniques	Trainable Parameters	Memory Footprint (GB)			
		Weights	Activations	Gradients	Total
Full	737M (100%)	2.75	5.33	2.75	10.83
Adapters	12M (1.70 %)	2.80	4.04	0.05	6.89
LoRA	9M (1.26%)	2.78	4.31	0.04	7.13
Inference	/	2.75	/	/	2.75

Table 1: The breakdown of memory footprint. "Activations" contain the intermediate results and optimizer states. Model: T5-Large; mini-batch size: 16; sequence length: 128.

Memory footprint exceeds typical edge device memory budgets

- **Keep user data in-situ to protect privacy.**
- **Without wide area network transmission.**
- **Alleviating data center pressure by leveraging ubiquitous computing resources.**

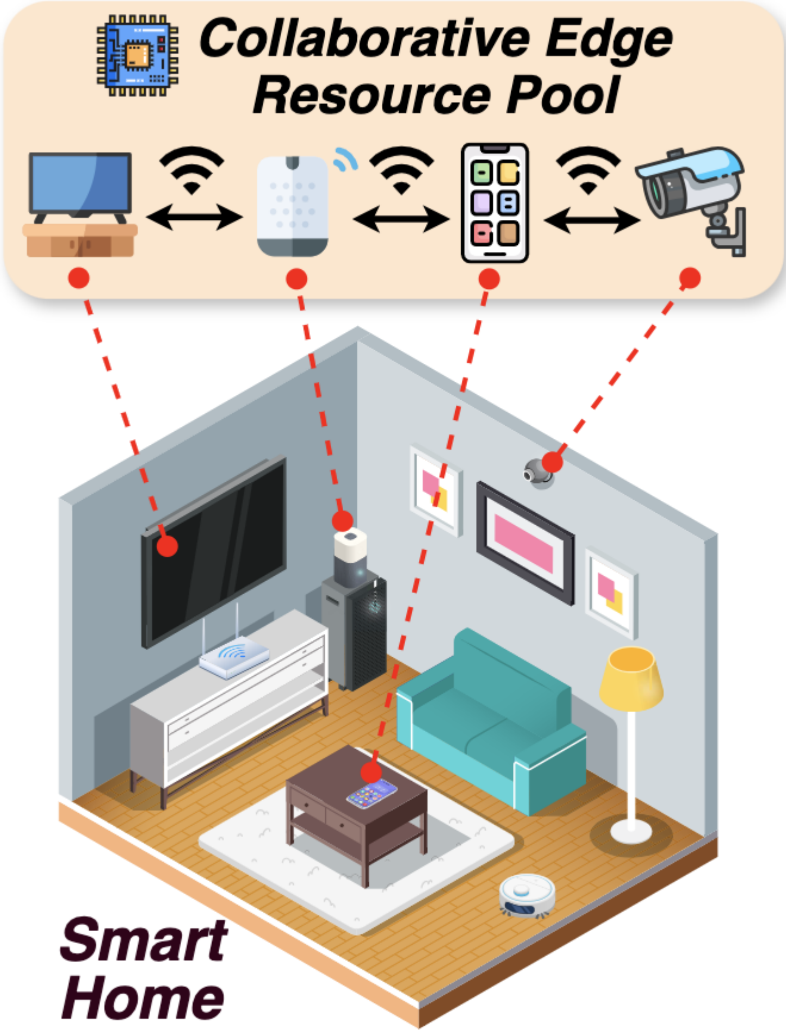
Limited and non-scalable on-board computing resources

Opportunities: Collaborative Edge Computing

Opportunities

- ✓ Edge scenarios like smart homes usually comprise **a group of trusted idle devices** *(e.g., mobile phones, laptops, and smart-home devices owned by the same user or family)*

- ✓ These accompanying devices are typically in physical proximity to the primary one running on-device learning tasks and can be associated as a **collaborative resource pool** for in-situ DNNs training acceleration.



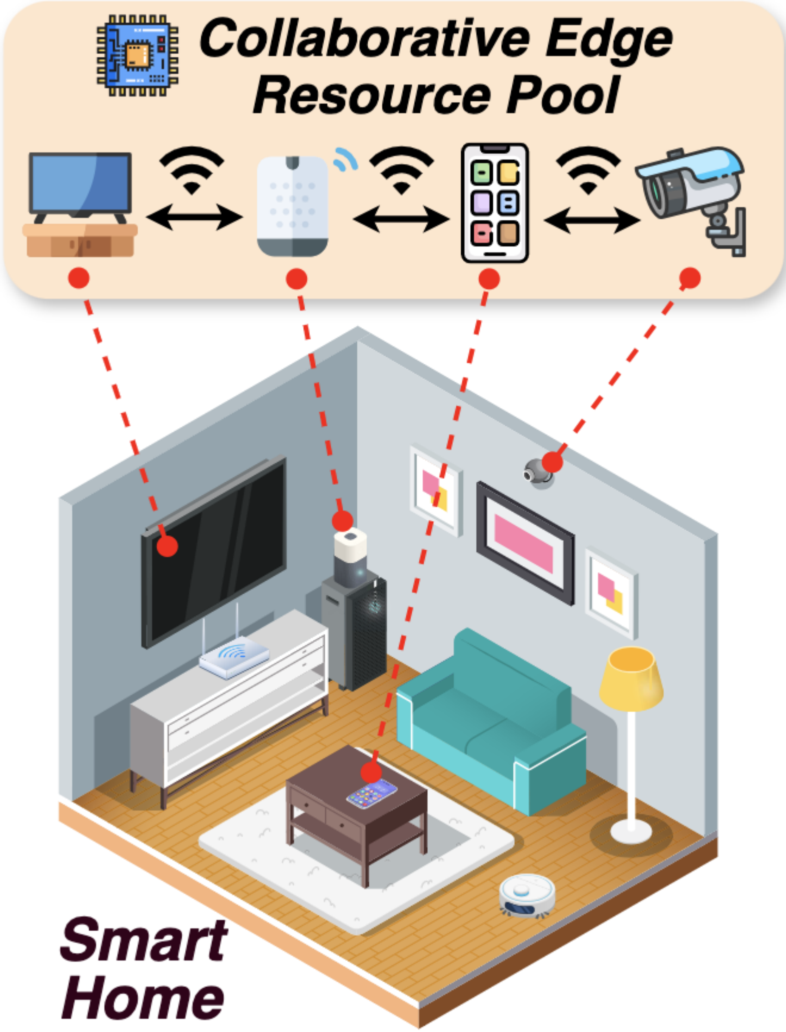
Challenges in Collaborative Edge Training

Challenges

1. How to select the best parallel architecture to orchestrate multiple edge devices?

2. How to tailor parallelism planning to the resource budget of **heterogeneous** edge devices?

3. How to render stable and reliable DNNs training under **dynamic edge environment**.



Challenges in Collaborative Edge Training

Challenges

1. How to select the best parallel architecture to orchestrate multiple edge devices?

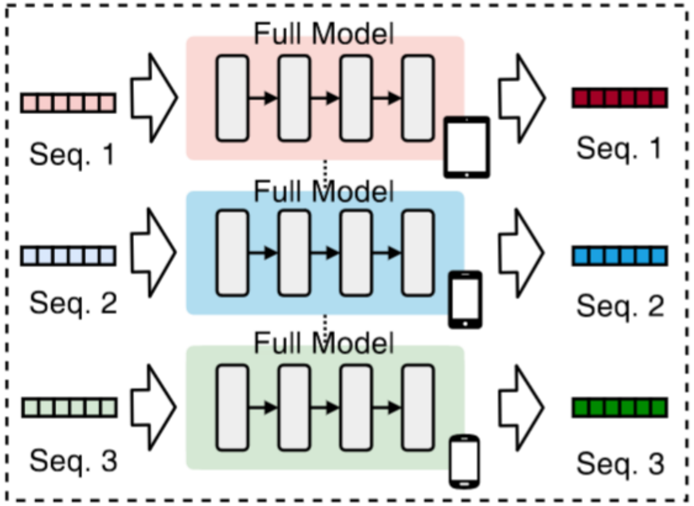
2. How to tailor parallelism planning to the resource budget of **heterogeneous** edge devices?

3. How to render stable and reliable DNNs training under **dynamic edge environment**.

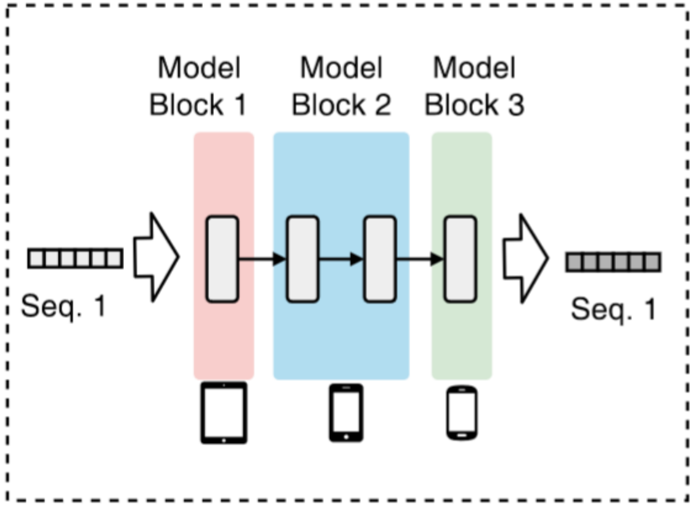
Hybrid Pipeline Parallelism (HPP) in Asteroid

- **Solution to Challenge #1**: Choosing the most suitable parallelism strategy.

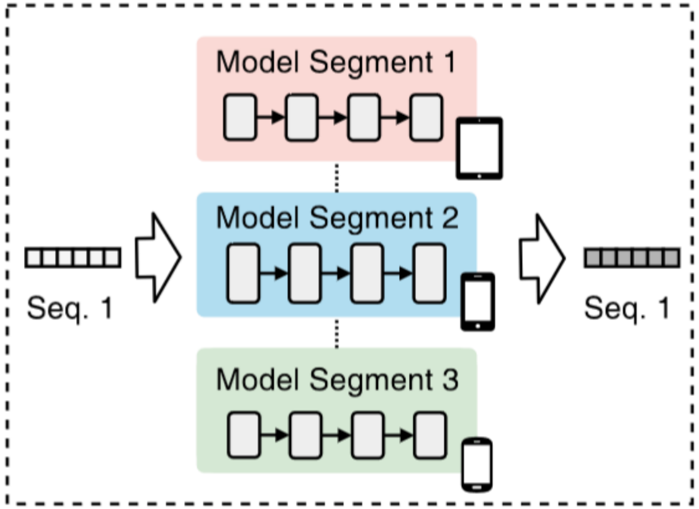
Data Parallelism



Pipelined Model Parallelism



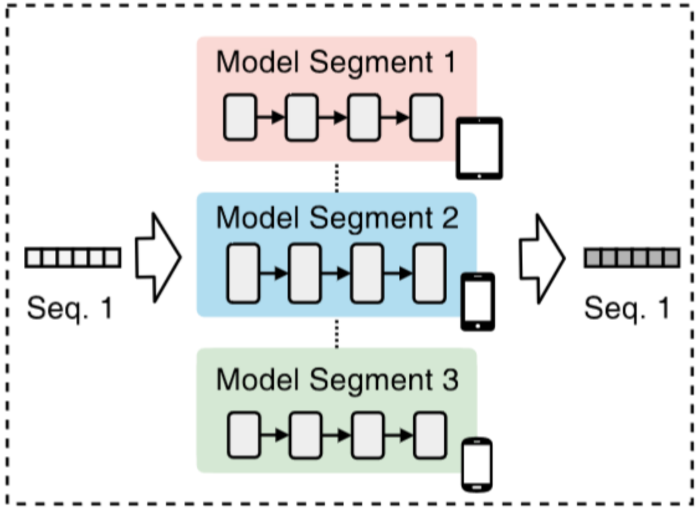
Intra-Operator Model Parallelism



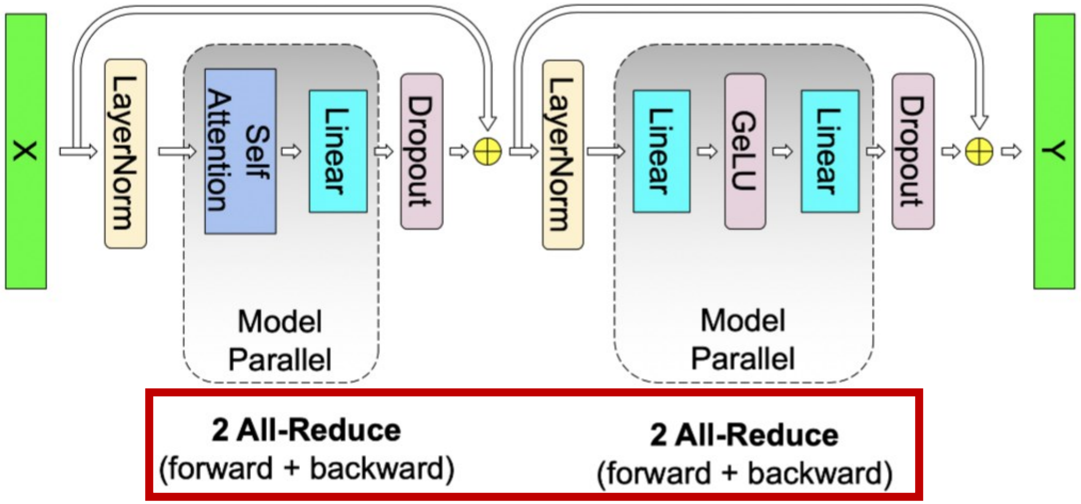
Hybrid Pipeline Parallelism (HPP) in Asteroid

- **Solution to Challenge #1**: Choosing the most suitable parallelism strategy.

Intra-Operator Model Parallelism



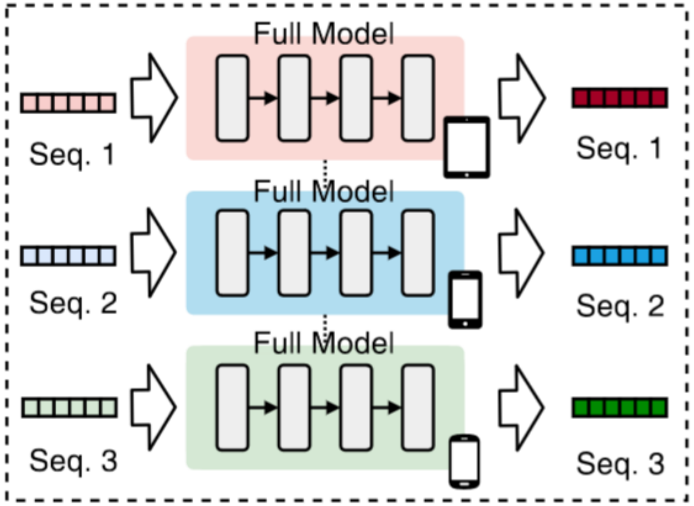
- Operator-level model parallelism encounters data dependency issues
- Necessitating extensive synchronization of intermediate tensors at each DNN layer.



Hybrid Pipeline Parallelism (HPP) in Asteroid

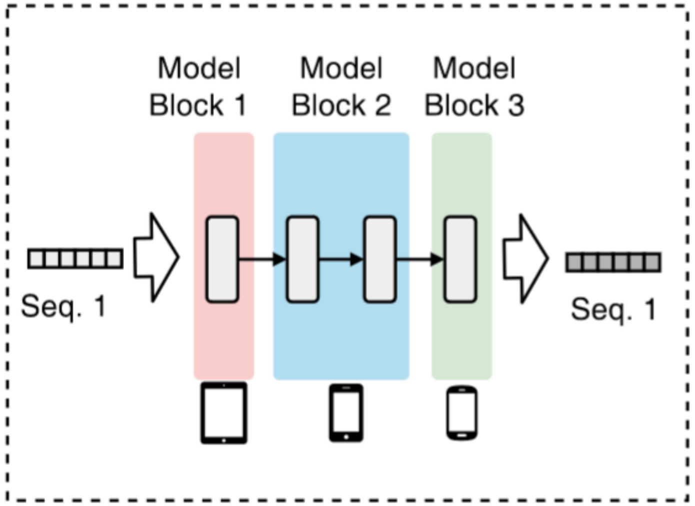
- **Solution to Challenge #1**: Choosing the most suitable parallelism strategy.

Data Parallelism



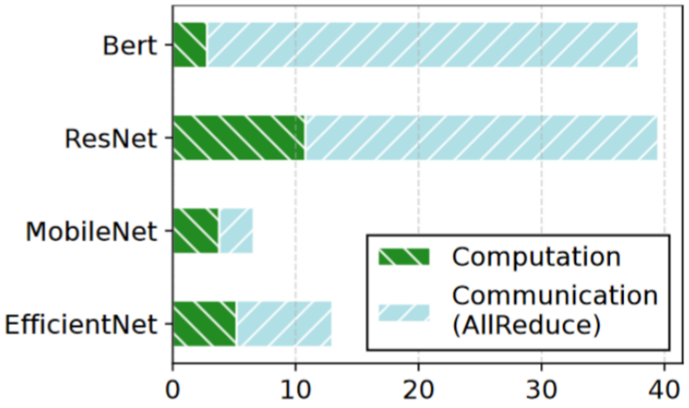
✔ Convolutional layers

Pipelined Model Parallelism

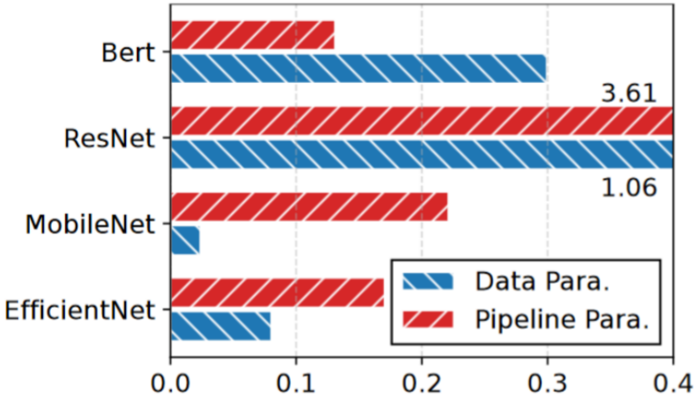


✔ Fully connected layers

✔ Multi-Attention layers



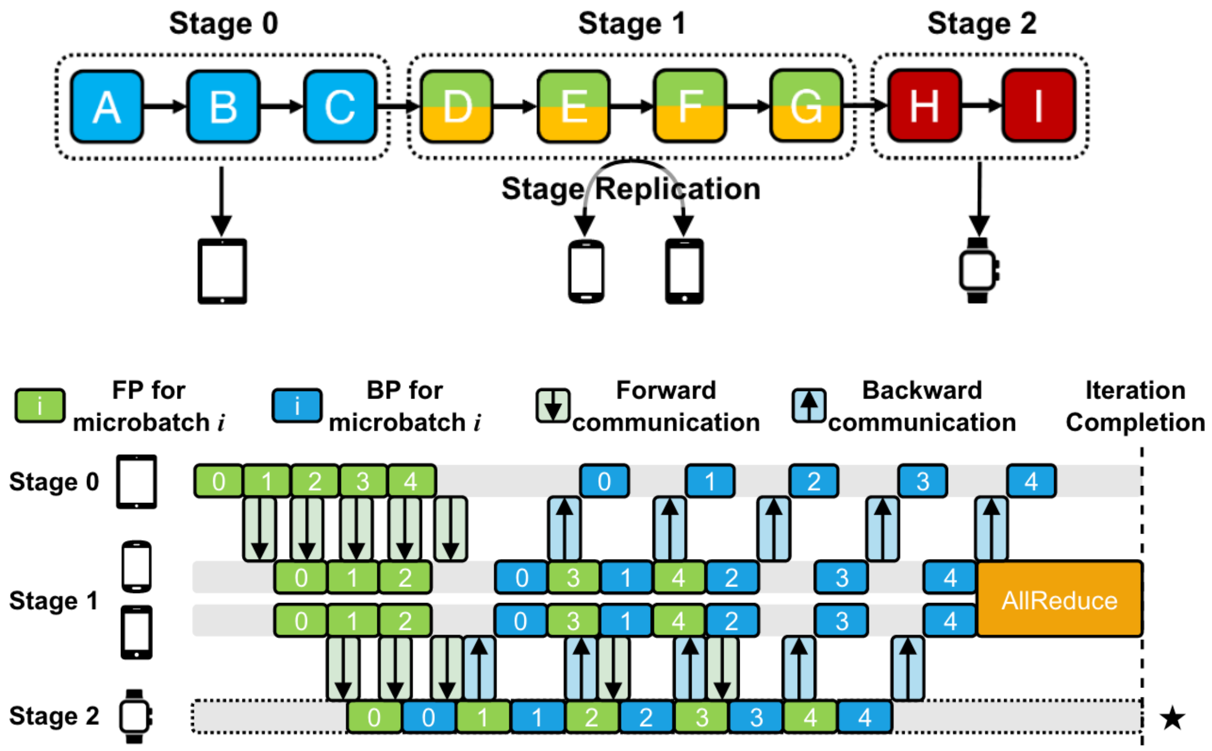
Latency Breakdown in a DP round (s)



Communication Volume (MB/sample)

Hybrid Pipeline Parallelism (HPP) in Asteroid

- **Solution to Challenge #1**: Utilizing hybrid pipeline parallelism for orchestration.
  - Step 1: **Divide the DNNs** into multiple pipeline stages, with each stage containing a sub-model.
  - Step 2: **Group the edge devices** and assign each group to a different pipeline stage.
  - Step 3: Each mini-batch of fine-tuning data is split into multiple micro-batches and injected into the pipeline, enabling **inter-group pipeline parallelism** and **intra-group data parallelisms.**



Challenges in Collaborative Edge Training

Challenges

1. How to select the best parallel architecture to orchestrate multiple edge devices?

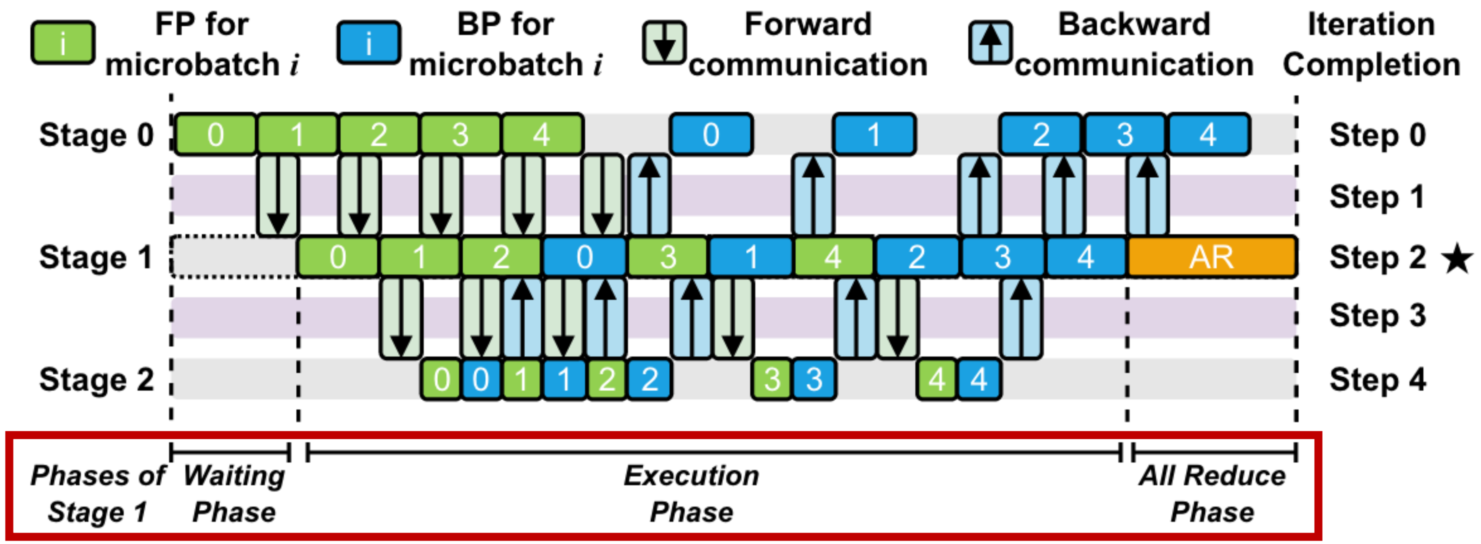
2. How to tailor parallelism planning to the limited resource budget of **heterogeneous** edge devices?

3. How to render stable and reliable DNNs training under **dynamic edge environment**.

Parallelism Planning for HPP

- **Solution to Challenge #2**: Optimize workload partitioning and device Orchestration.
  - *Optimization Objective:* $\text{HPP-Round Latency} = \max_{s\in\{0,1,\ldots,S-1\}} \left(T_w^s + T_e^s + T_a^s\right),$

$$T_w^s = \sum_{i=0}^{s-1} E_f^i, \quad T_a^s = \frac{2\left(|\mathcal{G}_s| - 1\right) \cdot \sum_{l\in\mathcal{D}_s} w_l}{|\mathcal{G}_s| \cdot \min_{d,d'\in\mathcal{G}_s} b_{d,d'}}. \quad T_e^s = M \times \left(E_f^{dm} + E_b^{dm}\right) + \begin{cases} \sum_{i=s}^{dm-1}\left(E_f^i + E_b^i\right), & s < dm, \\ -\sum_{i=dm}^{s-1}\left(E_f^i + E_b^i\right), & s \ge dm. \end{cases}$$



Parallelism Planning for HPP

- **Solution to Challenge #2**: Optimize workload partitioning and device Orchestration.

  A novel **dynamic programming algorithm** is devised to facilitates optimal parallelism planning.

Algorithm 2: Dynamic Programming HPP Planning

for p ***from*** 1 ***to*** $min(L, N)$ **do**
    for n ***from*** 1 ***to*** N **do**
        for l ***from*** 1 ***to*** L **do**
            for n' ***from*** 0 ***to*** n **do**
                for l' ***from*** 0 ***to*** l **do**
                    Get E^s_f and E^s_b with Alg. 1 and Eq. (8);
                    Update Dominant Step with Eq. (11);
                    Get T^s_w, T^s_e and T^s_a with Eq. (5) and (6);
                    Get HPP-Round Latency with Eq. (4);
                Update $Q(l, n, p)$ with Eq. (10);

Properties	PipeDream	Dapple	Alpa	HetPipe	Asteroid
Combining DP with PP?	✓	✓	✓	✓	✓
Resource Heterogeneous Awareness?				✓	✓
Memory Constraint Awareness?			✓		✓
Communication Modeling & Optimization?		✓			✓

Challenges in Collaborative Edge Training

Challenges

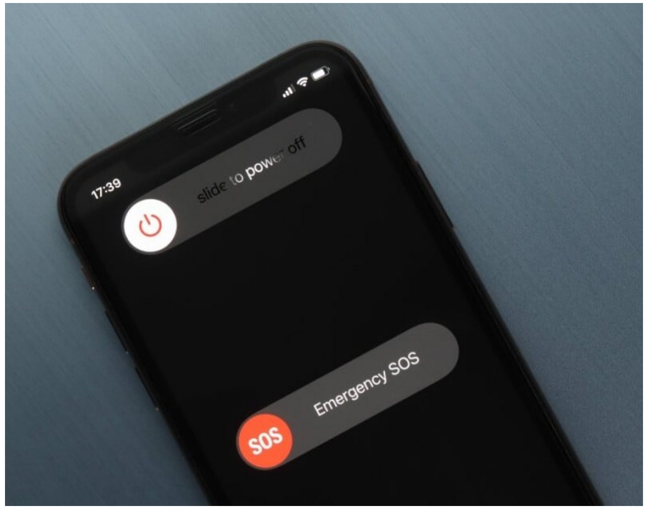
1. How to select the best parallel architecture to orchestrate multiple edge devices?

2. How to tailor parallelism planning to the limited resource budget of **heterogeneous** edge devices?

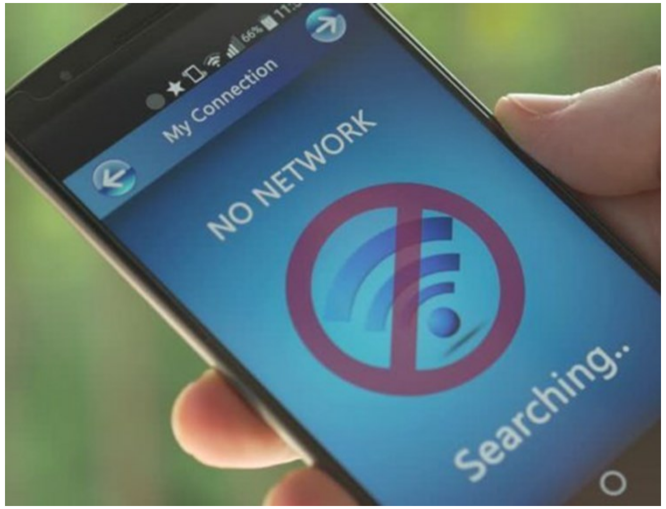
3. How to render stable and reliable DNNs training under **dynamic edge environment**.

Fault-Tolerant Pipeline Replay

- **Devices at the edge exhibit strong dynamics.**
  - The device departing can result in the loss of the trained weights.
  - An abnormal device in the pipeline can cause training to stop.

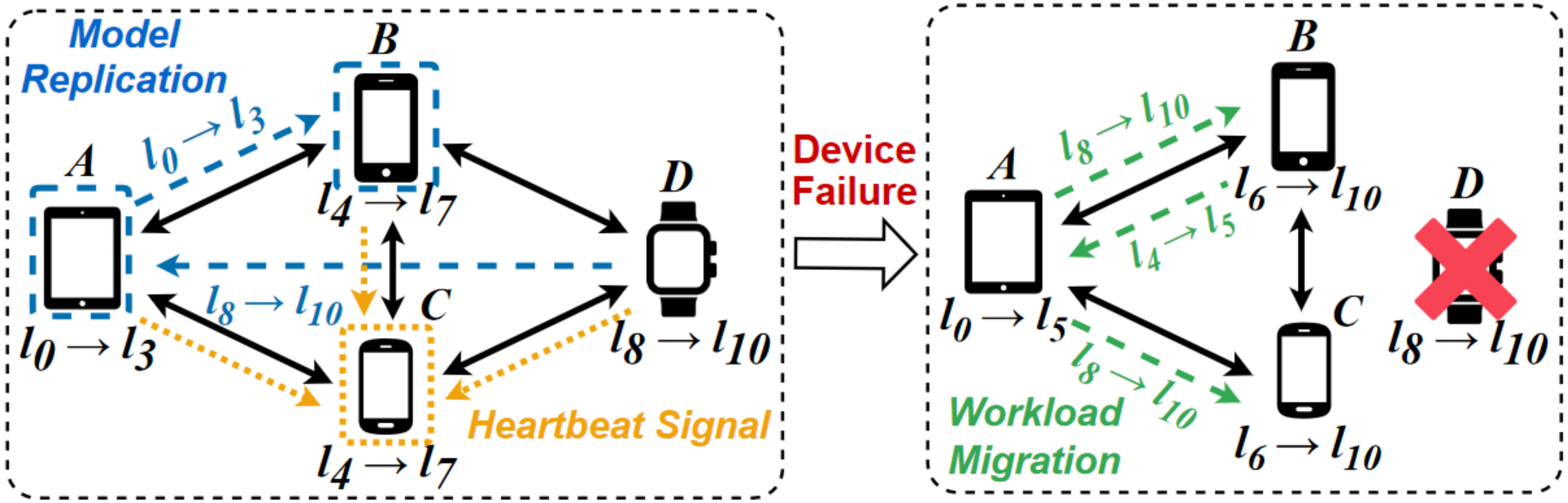


Energy Depletion



Network Anomalies

Fault-Tolerant Pipeline Replay



- Heartbeat-guided Failure Detection.
- Topology-driven Model Replication.
- Layer-wise Lightweight Pipeline Re-planning.

Evaluation

- **Testbeds**
  - ➢ Using these 3 heterogeneous devices, we simulated **4 different edge clusters**, including both homogeneous and heterogeneous clusters.

Table 5: Specifications of edge devices in experiments.

Edge Device	GPU Processor	Memory
Jetson Nano [2]	128-core NVIDIA Maxwell	4GB
Jetson TX2 [1]	256-core NVIDIA Pascal	8GB
Jetson NX [3]	384-core NVIDIA Volta	8GB

Table 6: Heterogeneous edge env. used in experiments.

ID	Devices	ID	Devices
A	$5 \times$ Nano	C	$1 \times$ NX, $2 \times$ TX2, $3 \times$ Nano
B	$3 \times$ NX, $2 \times$ TX2	D	$1 \times$ TX2, $3 \times$ Nano

- **Models and datasets**
  - ➢ 4 typical DNNs models widely used in CV and NLP areas: EfficientNet, MobileNet, ResNet and BERT.
  - ➢ Evaluate with the CIFAR-10, Mini-ImageNet and GLUE dataset.

Evaluation

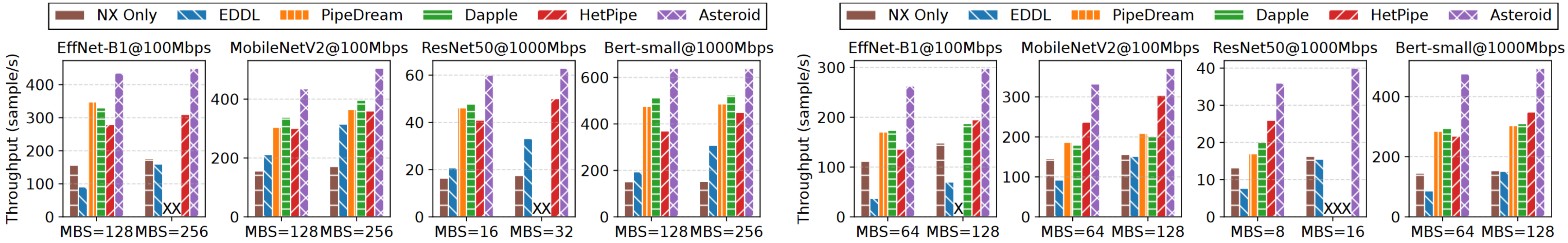
Maintained high performance **across various edge environment and network conditions**, with up to **12.8x** training acceleration compared to DP and PP!!!

Table 4: Summary of throughput results comparing Asteroid with on-device training, data parallelism (DP), and pipeline parallelism (PP). The pipeline configuration generated by Asteroid is visualized in Fig. 12. We select the most powerful device in each edge environment as the platform for on-device training.

Task	Model	Dataset	Input Size	Edge Environment	Asteroid Config.	Speedup over Device	Speedup over DP	Speedup over PP
Image Classification	EfficientNet-B1 [49]	Cifar-10 [15]	$3 \times 32 \times 32$	A (100Mbps)	❶	4.4×	2.1×	2.8×
				B (100Mbps)	❹	3.0×	4.8×	9.7×
				B (1000Mbps)	❹	3.7×	2.1×	1.4×
	MobileNetV2 [45]	Cifar-10 [15]	$3 \times 32 \times 32$	A (100Mbps)	❷	4.5×	1.5×	3.5×
				B (100Mbps)	❺	3.2×	2.3×	11.2×
				B (1000Mbps)	❻	3.8×	1.2×	1.3×
	ResNet50 [20]	Mini-ImageNet [52]	$3 \times 224 \times 224$	A (100Mbps)	❷	3.4×	3.6×	5.8×
				B (100Mbps)	❻	1.5×	6.1×	12.2×
				B (1000Mbps)	❹	3.7×	2.9×	3.1×
Language Model	Bert-small [14]	Synthetic Data	32×512	A (100Mbps)	❸	3.5×	6.4×	1×
				B (100Mbps)	❼	1.3×	6.8×	1×
				B (1000Mbps)	❼	3.9×	4.2×	1.3×

Evaluation

When compared with SOTA system for cloud, Asteroid achieves up to **86% latency reduction** compared to these baseline methods!!!



(a) Training throughput compared with existing approaches on Env. B.

(b) Training throughput compared with existing approaches on Env. C.

Figure 13: Training throughput comparison under various settings. × means out-of-memory error.

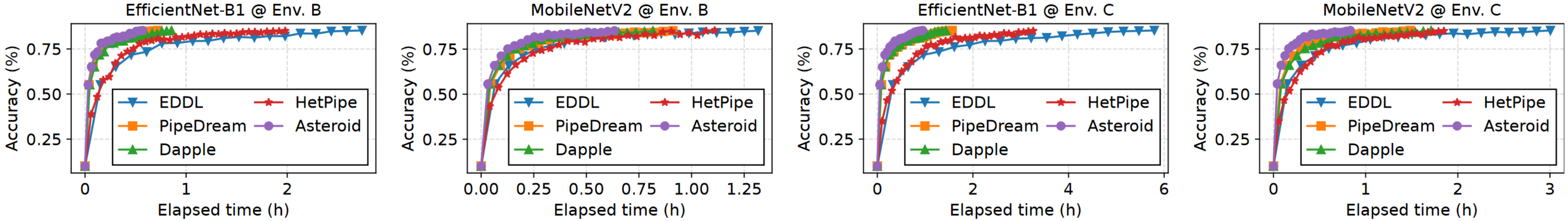


Figure 14: Training convergence of EfficientNet-B1 and MobileNetV2 on Env. B and C compared with baselines.

Evaluation

Our design enables efficient replay of training within several seconds, while simultaneously maintaining a high training throughput by rebalancing the pipeline.

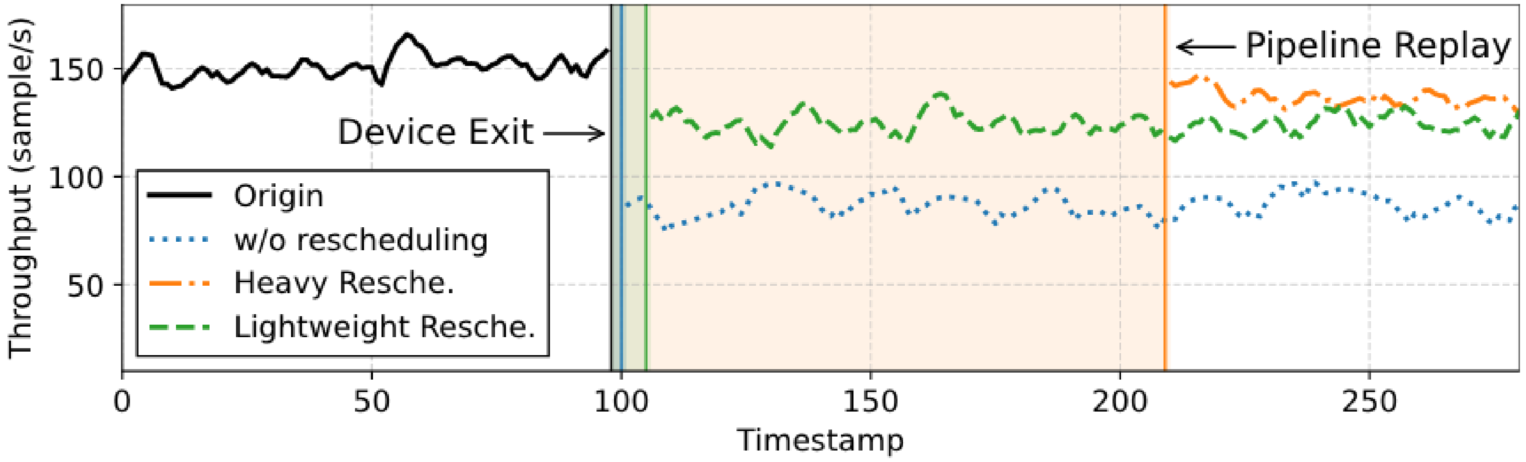
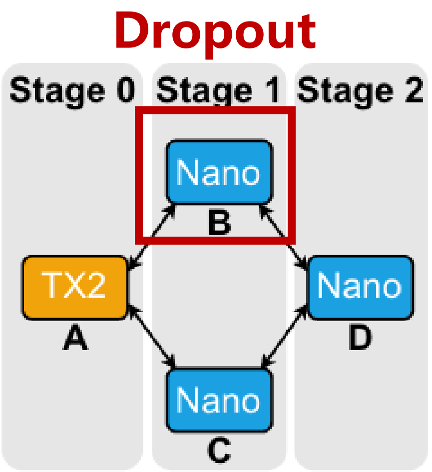
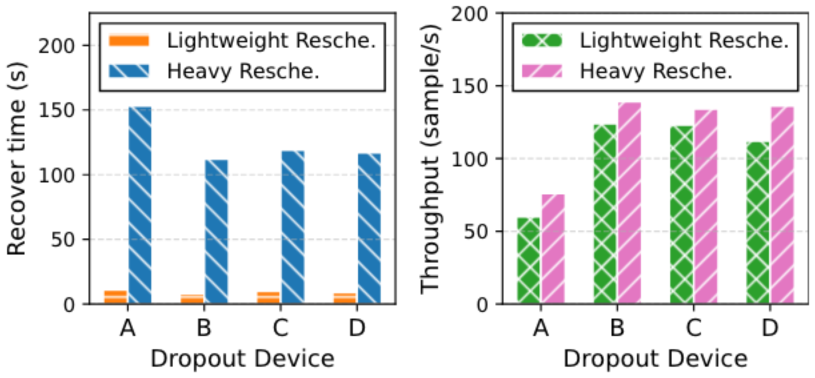


Figure 17: Throughput variation of different scheduling strategies when device B exits the training pipeline.





Takeaway

Eco: An **E**dge **CO**llaborative AI framework for serving miscellaneous AI model at the edge.

💡 We aim to design affordable, accessible, and adaptive AI with your private group of mobile and edge devices.

https://collaborative-edge-ai.github.io/

Eco Project Page

Features

😊 Optimized Computation

- Language models
- Vision perceptrons
- Graph nets

⚒ Heterogeneity Awarenss

- Mobile phones
- Embedded devices
- Edge servers

🏄 Resilient Elasticity

- Device breakdown
- Load variation
- Bandwidth fluctution

Thanks for listening

Shengyuan Ye[1], Liekang Zeng[1], Xiaowen Chu[2], Guoliang Xing[3], Xu Chen[1]

[1] **Sun Yat-sen University**
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] The Chinese University of Hong Kong

中山大學 SUN YAT-SEN UNIVERSITY
香港科技大学(广州) THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)
香港中文大學 The Chinese University of Hong Kong