Pluto and Charon: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

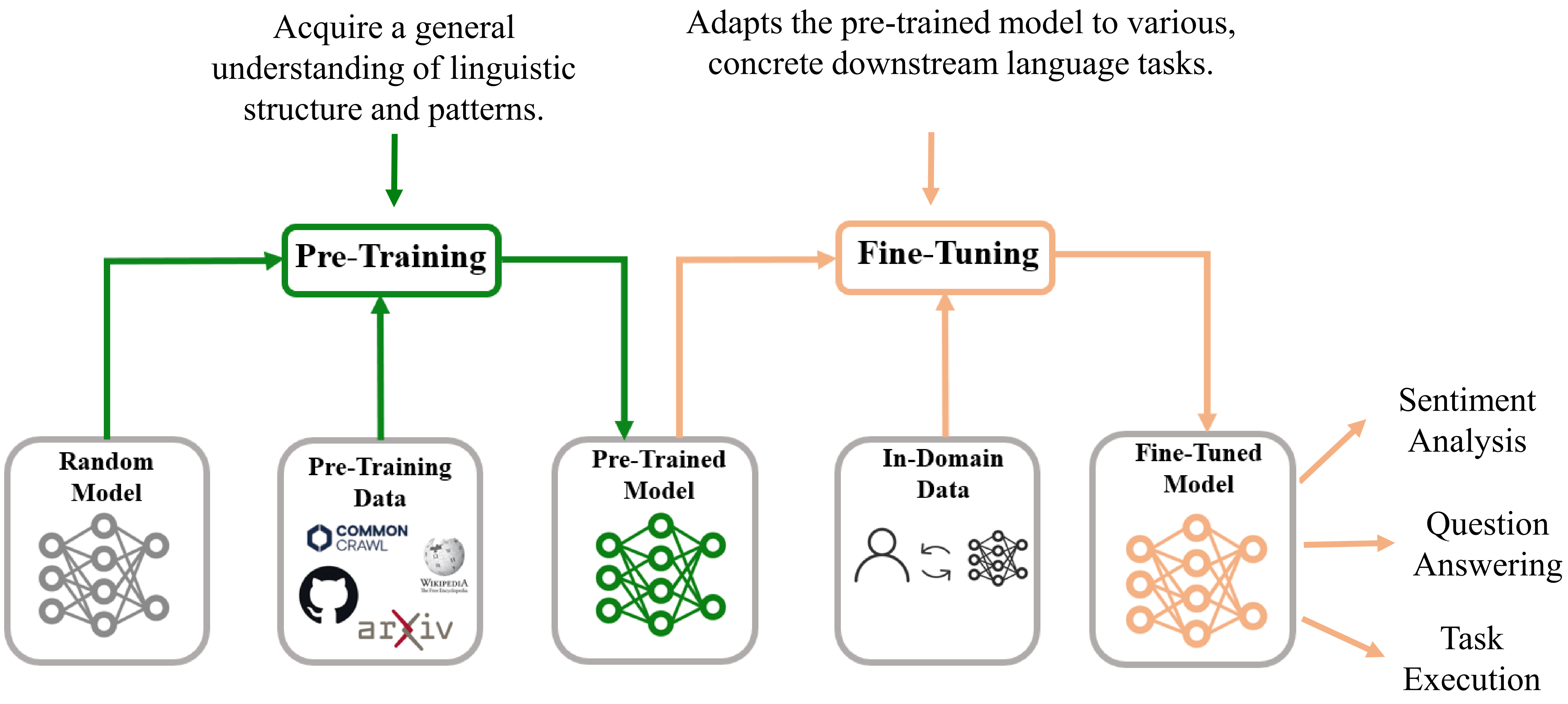
Bei Ouyang[★1], Shengyuan Ye[★1], Liekang Zeng[2], Tianyi Qian[1], Jingyi Li[1], Xu Chen[†1]

[1]**Sun Yat-sen University**, [2]HKUST(GZ)

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)

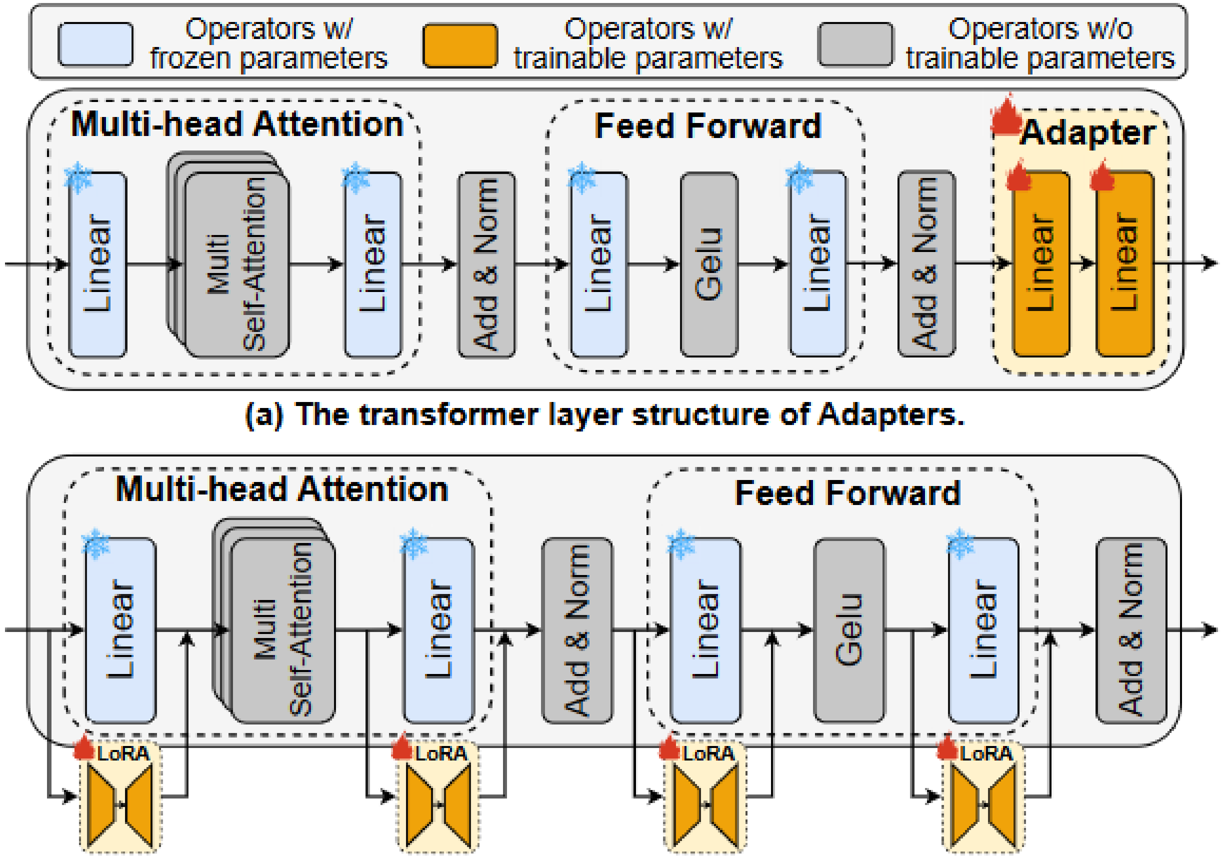
Transformer-Based LLMs and Fine-Tuning



Parameter-efficient finetuning (PEFT) techniques

- **Adapters**: inserts compact bottleneck modules at the end of each transformer layer.

- **LoRA**: injects trainable low-rank matrices into a frozen pre-trained model.

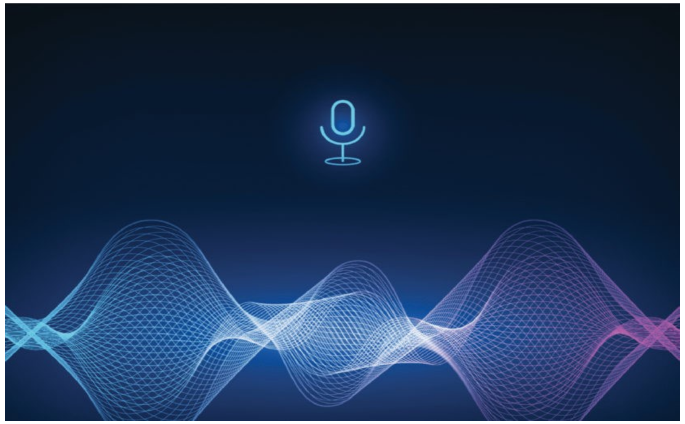


(a) The transformer layer structure of Adapters.

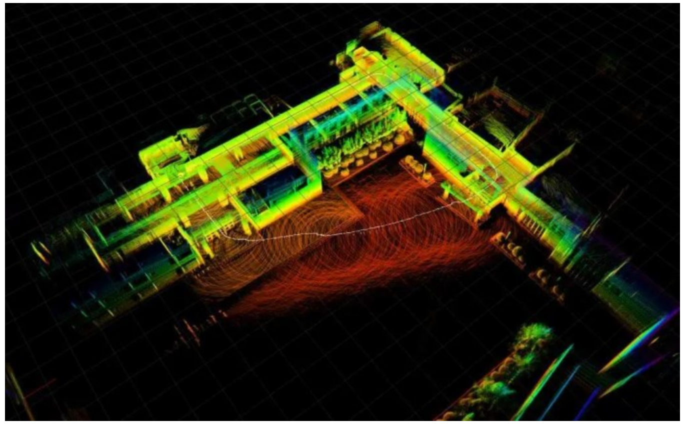
(b) The transformer layer structure of LoRA.

Intelligent edge applications

- **Transformer-based models driven increasing intelligent applications.**



Intelligent personal assistants

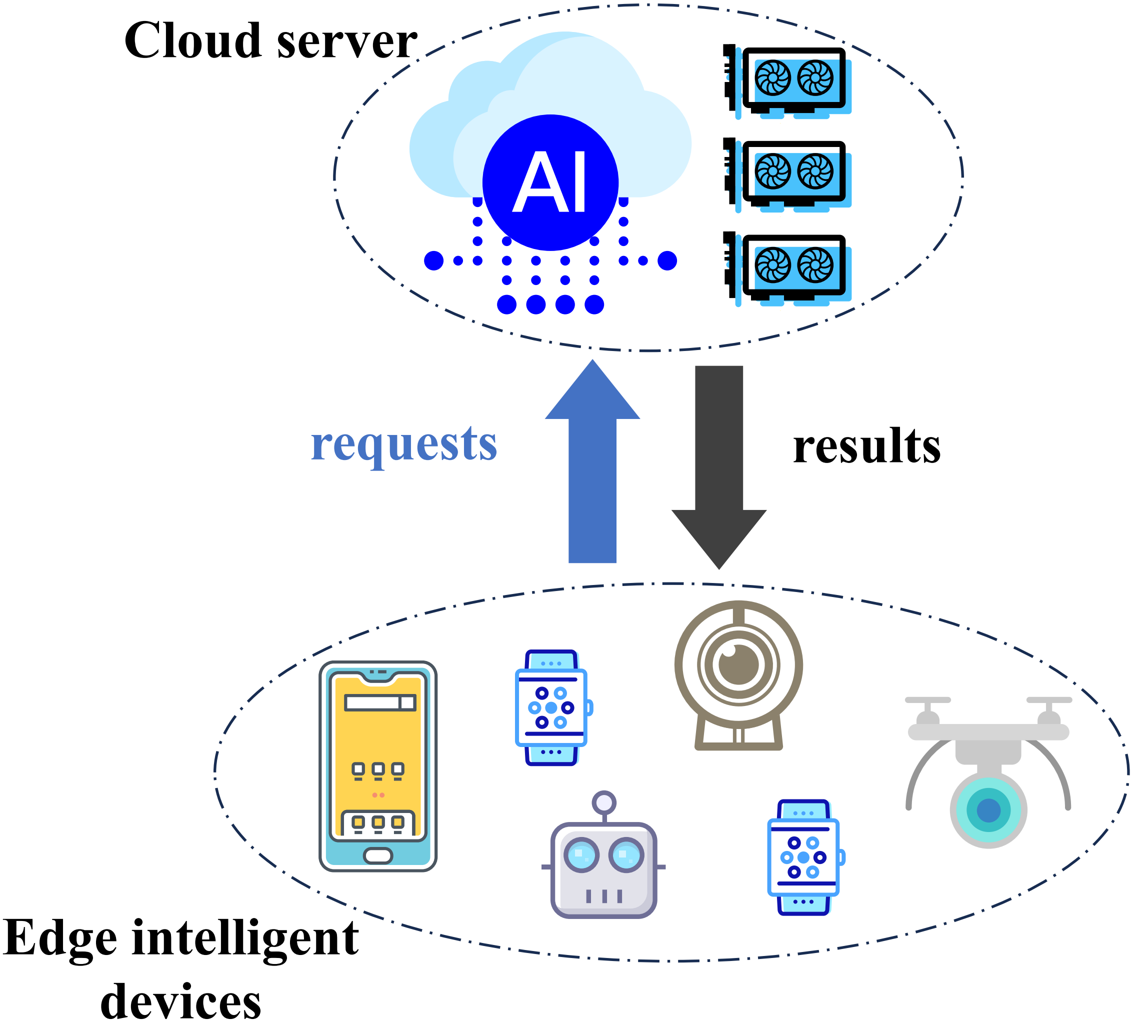


Intelligent robots/aircraft

AR&VR applications

Problems of cloud-assisted approaches

- Current Transformer-based applications heavily depend on cloud services.



The advantage of cloud service

- Powerful and scalable computing resources

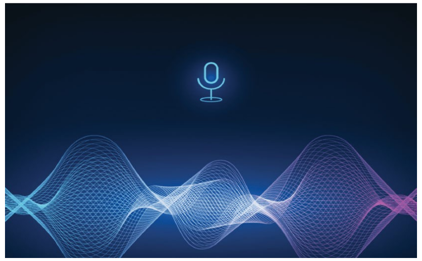
Three issues with cloud service

- Data privacy and security issues.
- Unreliable WAN connections.
- Network and datacenter pressure.

LLMs Fine-Tuning with Resource-Constrained Edge Devices

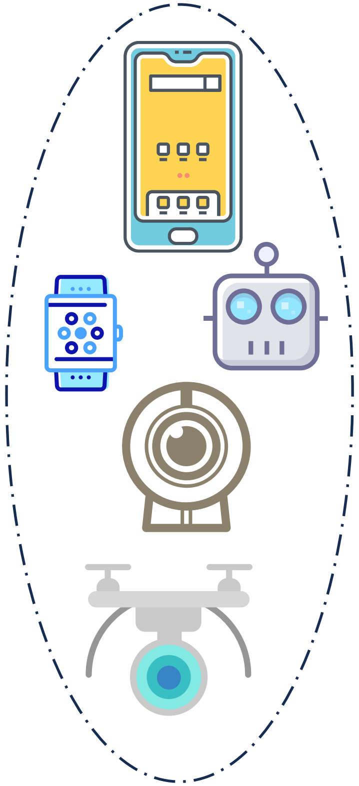
- On-device deployment becomes a promising paradigm for intelligent edge APPs.

Transformer-based intelligent applications



Deploy

Intelligent Edge Devices



Protect date privacy.

Without WAN transmission.

Limited and non-scalable on-board computing resources

Challenges

- PEFT techniques are **not resource-efficient** enough for edge environments.

➢ Adapters and LoRA exhibit a limited reduction in computation (around 30%).

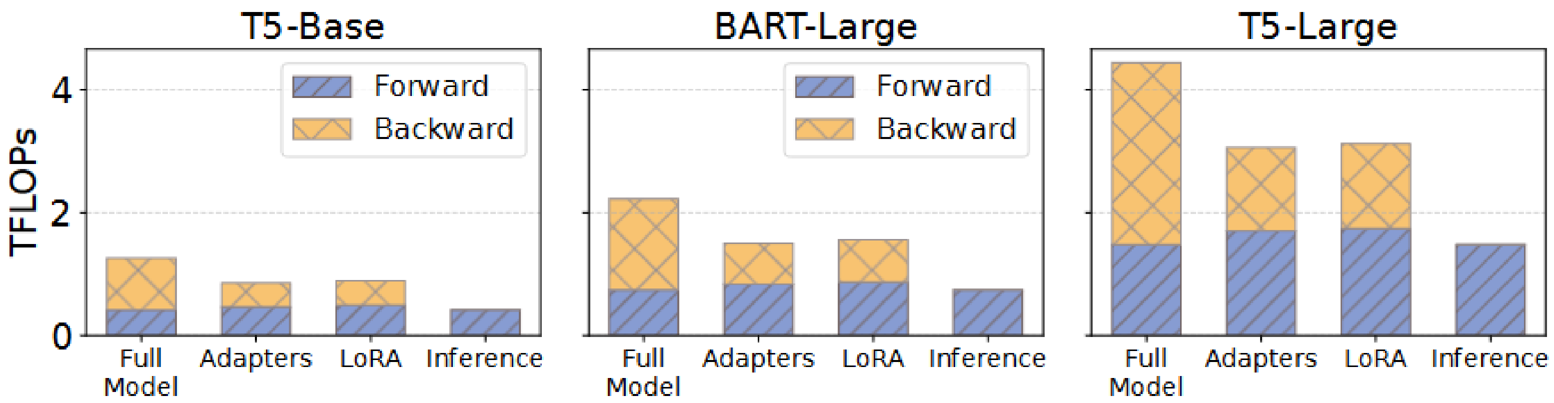
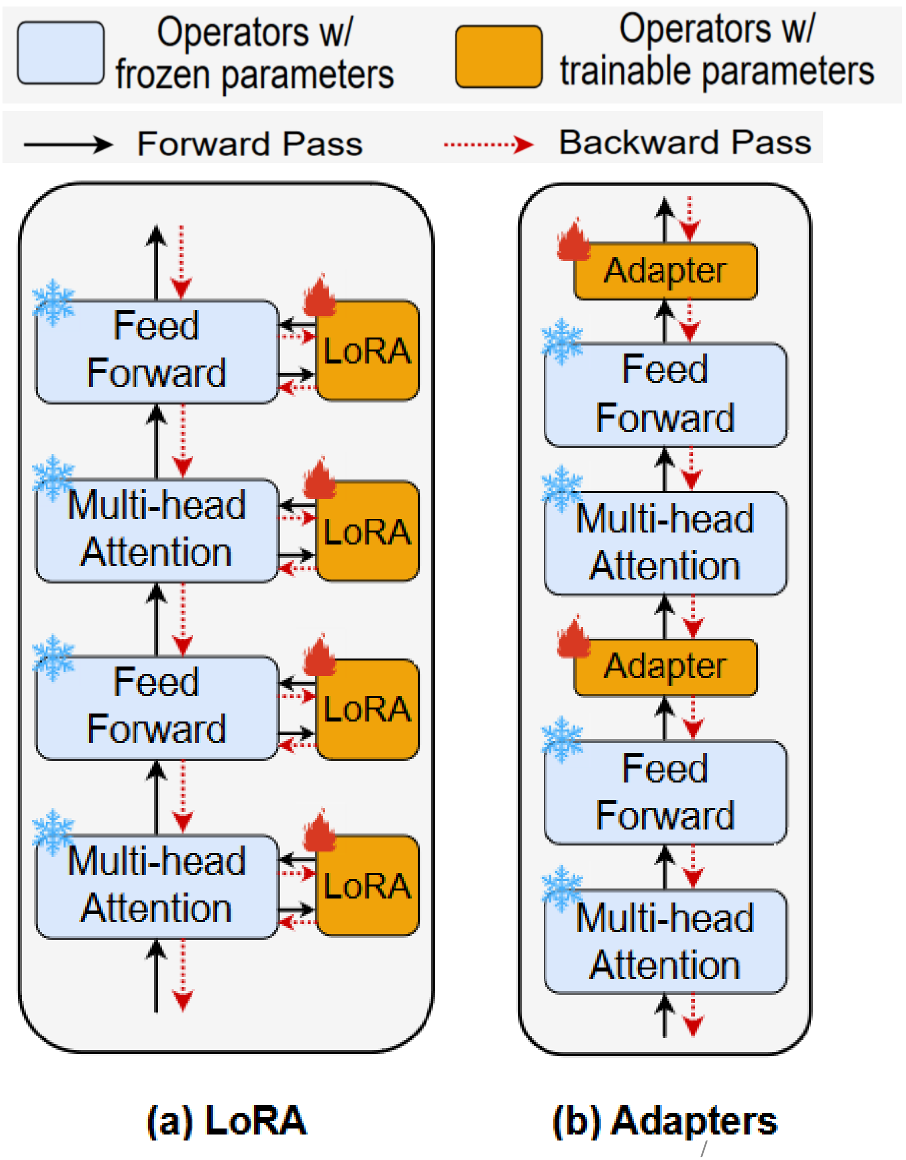


Figure 3: The comparison of floating point of operations (FLOPs). Mini-batch size: 16; sequence length: 128.



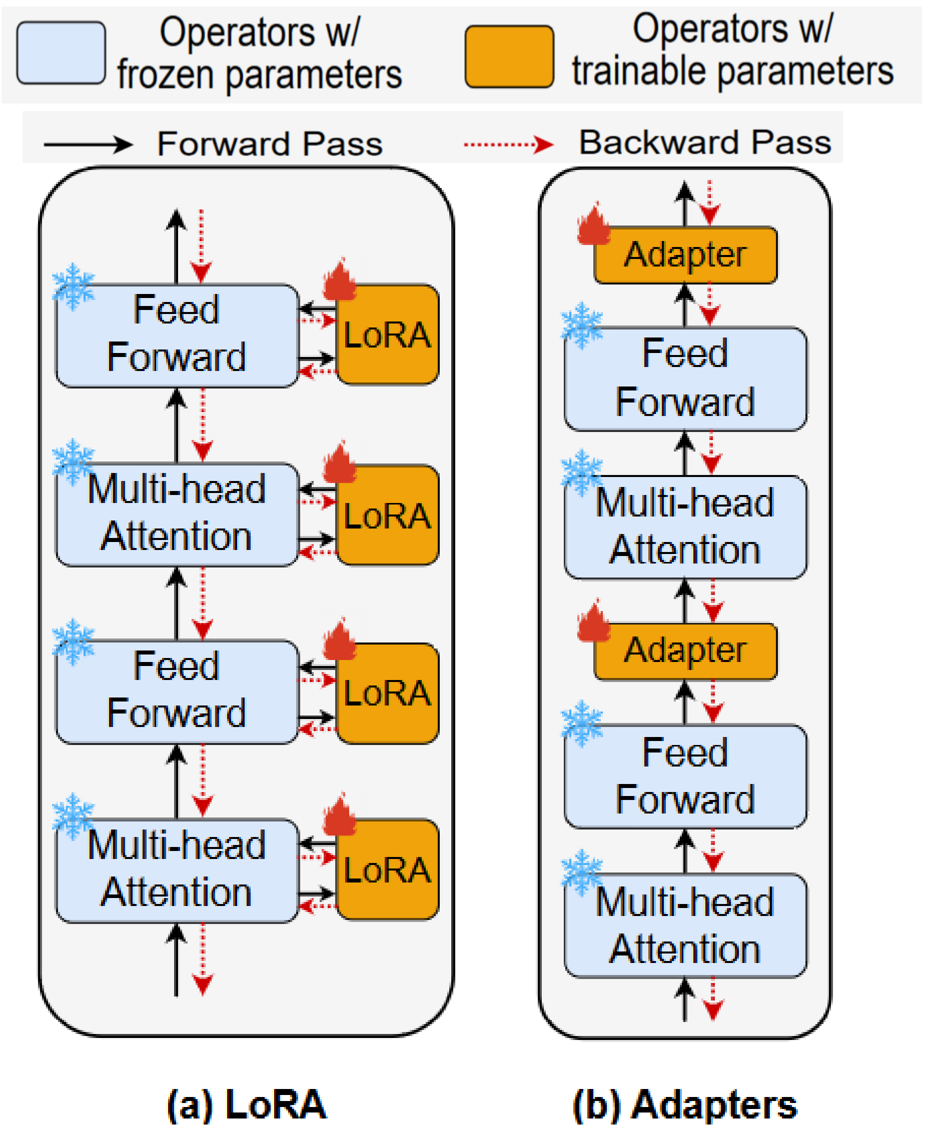
Challenges

- PEFT techniques are **not resource-efficient** enough for edge environments.

➢ Adapters and LoRA exhibit a with a maximum reduction of only 36% in memory.

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.



Challenges

- The fundamental contradiction between intensive LLM fine-tuning workload and constrained on-board resources.

➢ The computational capabilities of edge devices are constrained.

TABLE I
INFERENCE LATENCY AND MEM. FOOTPRINT OF TRANSFORMER MODELS

Model	DistilBert	Bert-L	GPT2-L	OPT-L	OPT-XL
Nano-M	0.37s	2.43s	OOM	OOM	OOM
Nvidia A100	5ms	20ms	29ms	27ms	38ms
Memory Footprint	130MB	680MB	1.6GB	2.6GB	5.4GB

Device	AI Performance
Jetson Nano	472 GFLOPS
NVIDIA A100	312 TFLOPS

121x performance gap.

Challenges

- The fundamental contradiction between intensive LLM fine-tuning workload and constrained on-board resources

➢ On-device fine-tuning is hindered by the memory wall.

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.

Device	Memory
Jetson Nano	4 GB
NVIDIA A100	40 GB/ 80 GB

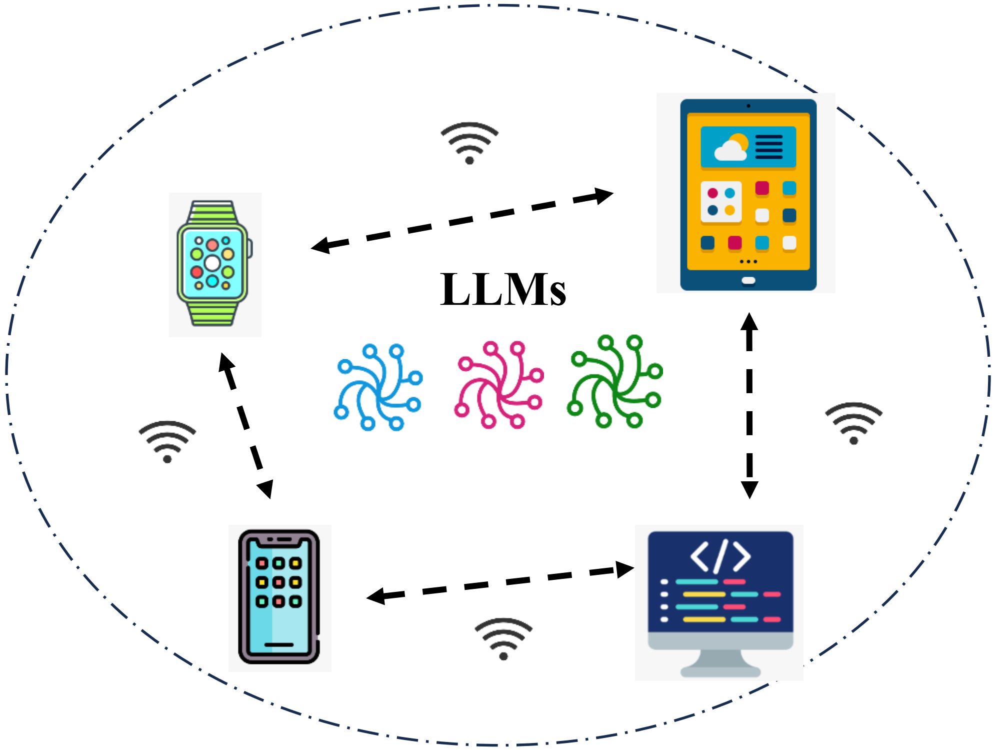
Incurs a peak memory footprint that is often unaffordable for edge devices.

Opportunities

Edge environment often comprise a rich set of trusted idle edge devices.

- Prevalent edge environments like smart homes usually comprise a group of trusted idle devices beyond a single terminal (e.g., mobile phones, laptops, and smart-home devices owned by the same user or family).

- These accompanying devices are typically in physical proximity and can be associated as a resource augmentation for in-situ personal LLMs fine-tuning.

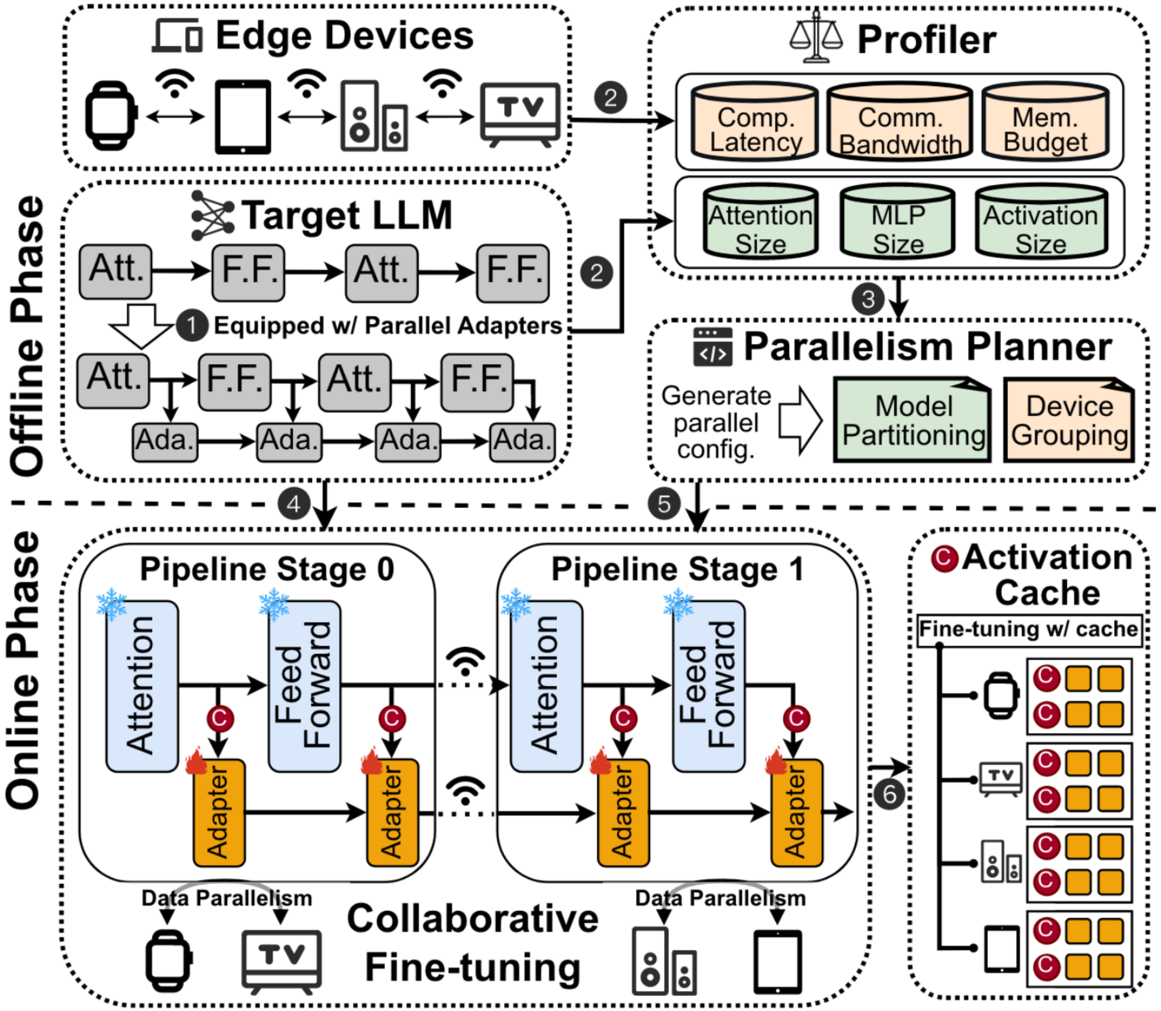


PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ **Algorithm-system codesign**

(Algorithm): In light of the side-tuning techniques, we employ not only parameter but also time and memory-efficient personal LLMs finetuning techniques with Parallel Adapters, which provides a dedicated gradient "highway" for the trainable parameters.

(System): We leverage edge devices in physical proximity and associate them as an edge resource pool for in-situ personal LLMs fine-tuning.



PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ **Algorithm-system codesign**

(Algorithm): In light of the side-tuning techniques, we employ not only parameter but also time and memory-efficient personal LLMs finetuning techniques with Parallel Adapters, which provides a dedicated gradient "highway" for the trainable parameters.

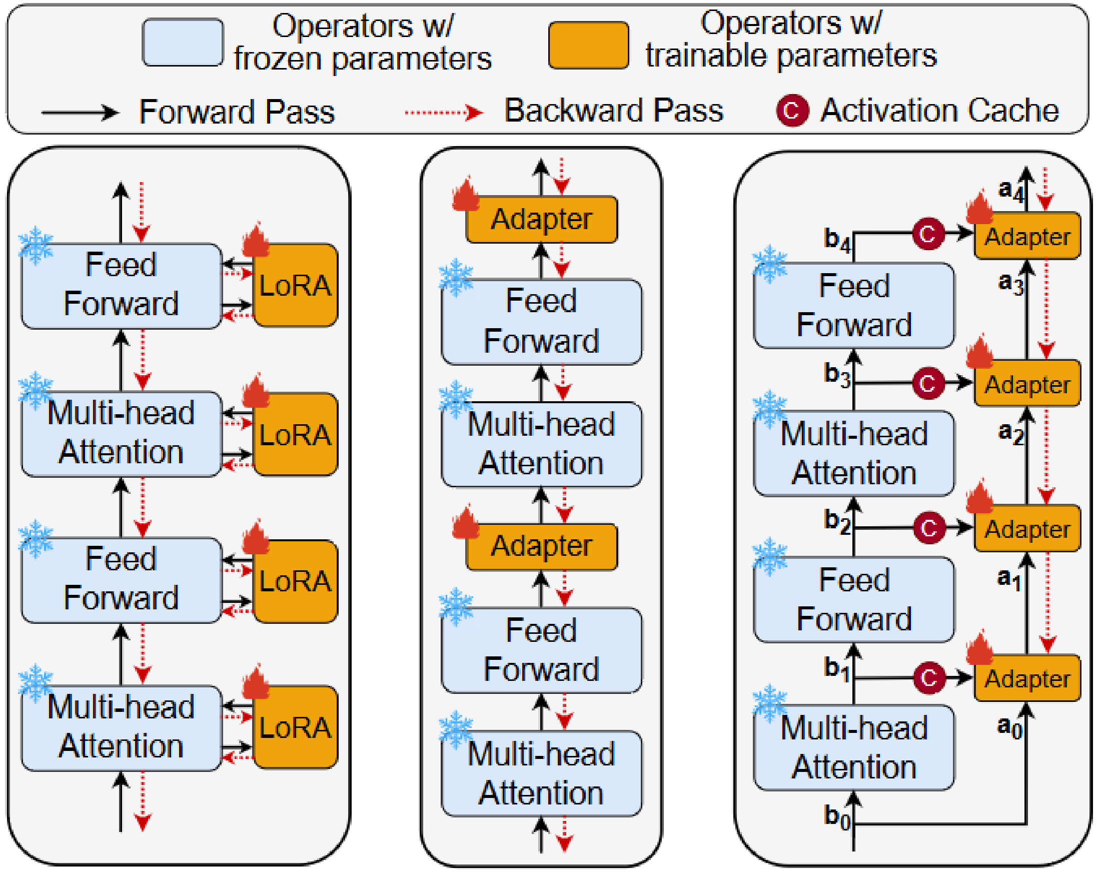
(System): We leverage edge devices in physical proximity and associate them as an edge resource pool for in-situ personal LLMs fine-tuning.

PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Fine-Tuning LLMs with Parallel Adapters

- The parameters of backbone transformer are **frozen**.

- Parallel adapters are a lightweight, **separate network** that takes the intermediate activations from the backbone LLM as input and generates predictions. **（Skip the backward propagation from the LLM backbone!）**



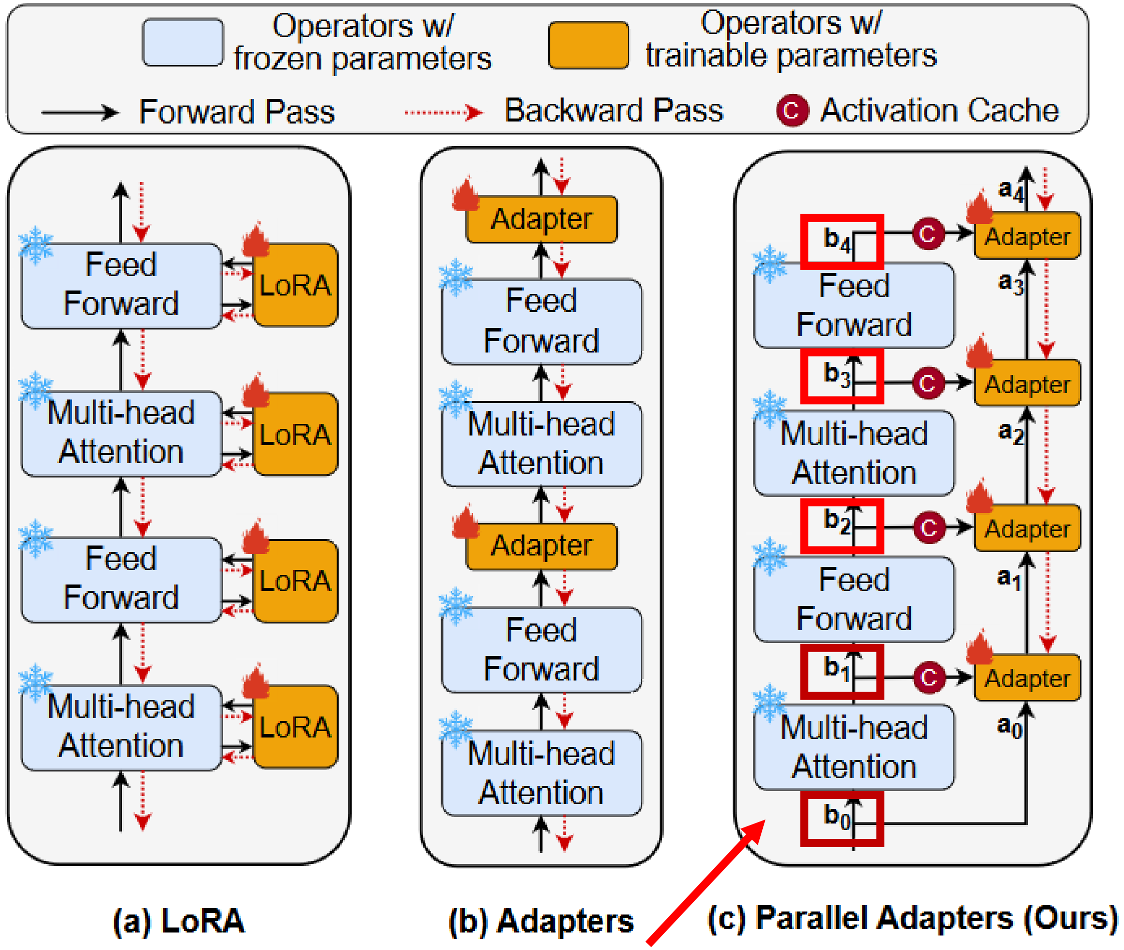
(a) LoRA (b) Adapters (c) Parallel Adapters (Ours)

PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ PAC Activation Cache for Parallel Adapters

- During the first epoch, when processing a new input sequence, cache all the input activations required by the Parallel Adapters that are obtained from the LLM backbone.

- In subsequent fine-tuning epochs using the same input sequence, we can reuse cached activations. **（Skip the forward propagation from the LLM backbone!）**



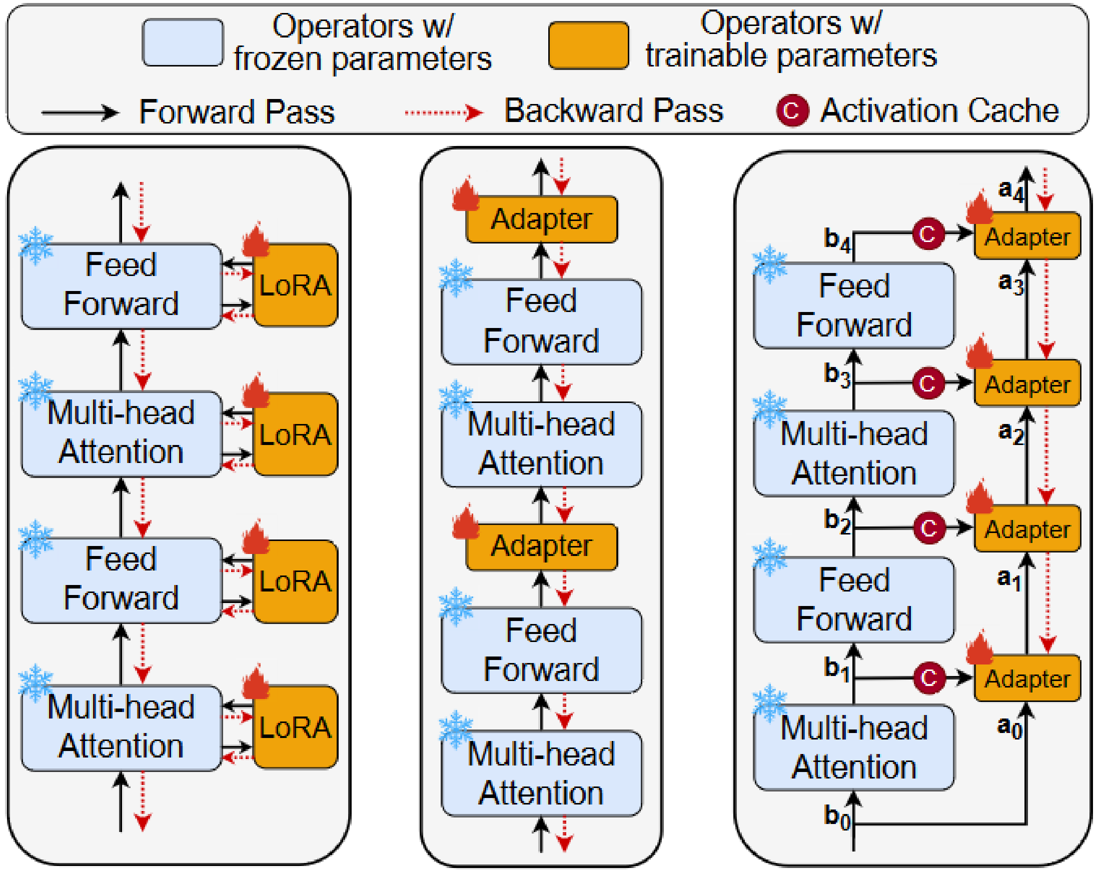
Invariant activations

PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ PAC Activation Cache for Parallel Adapters

Skip both the forward and backward propagation through the LLM backbone entirely !

- Significantly accelerating the fine-tuning process.
- Reducing the memory footprint by allowing the release of the memory space occupied by the LLM parameters.



(a) LoRA (b) Adapters (c) Parallel Adapters (Ours)

PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

- **Algorithm-system codesign**

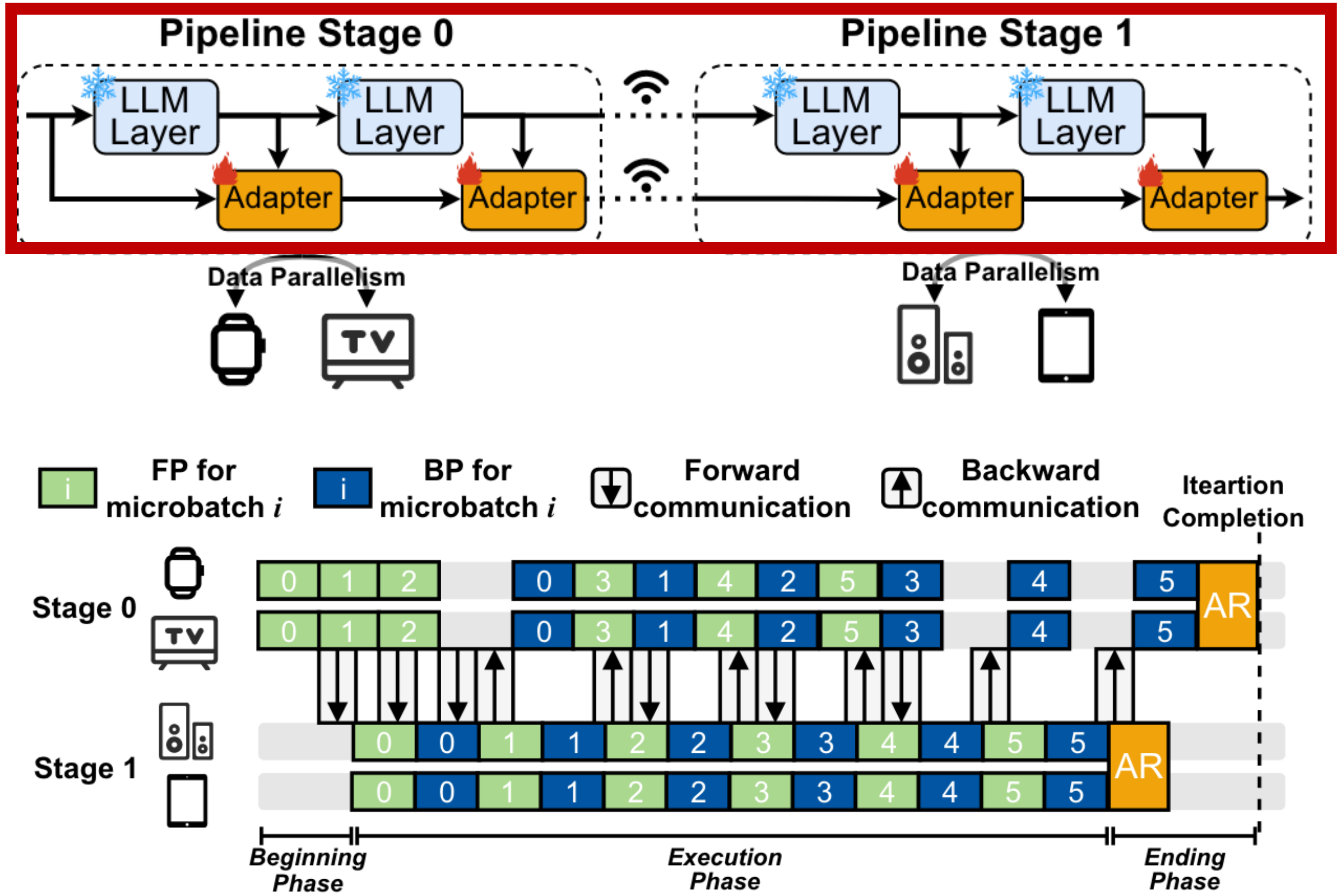
(Algorithm): In light of the side-tuning techniques, we employ not only parameter but also time and memory-efficient personal LLMs finetuning techniques with Parallel Adapters, which provides a dedicated gradient "highway" for the trainable parameters.

(System): We leverage edge devices in physical proximity and associate them as an edge resource pool for in-situ personal LLMs fine-tuning.

PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Data & Pipeline Hybrid Parallelism for LLMs Fine-Tuning

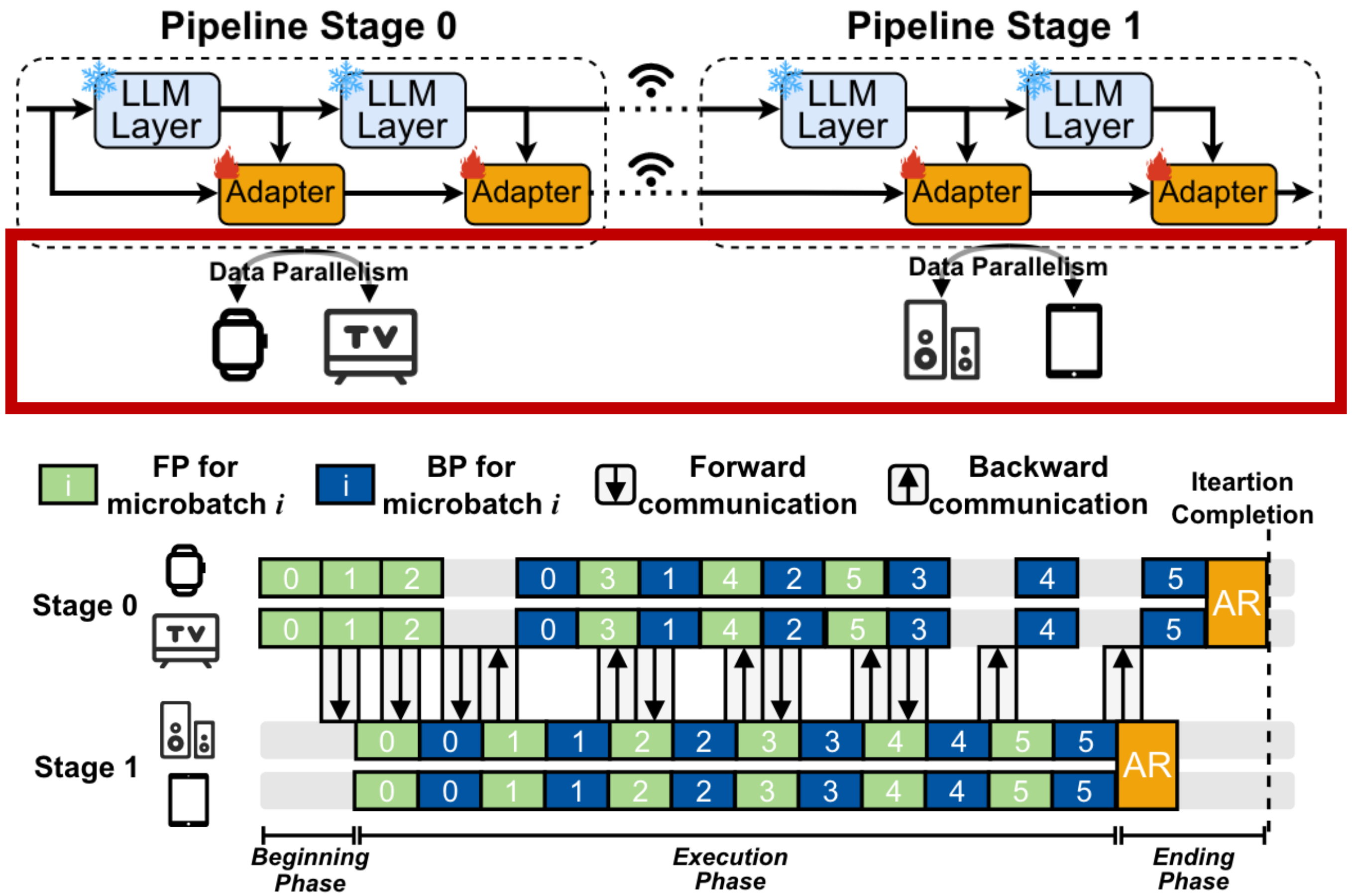
Step1 ：PAC first divides an LLM into multiple stages where each contains a stage model composed of a set of consecutive transformer layer.



PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Data & Pipeline Hybrid Parallelism for LLMs Fine-Tuning

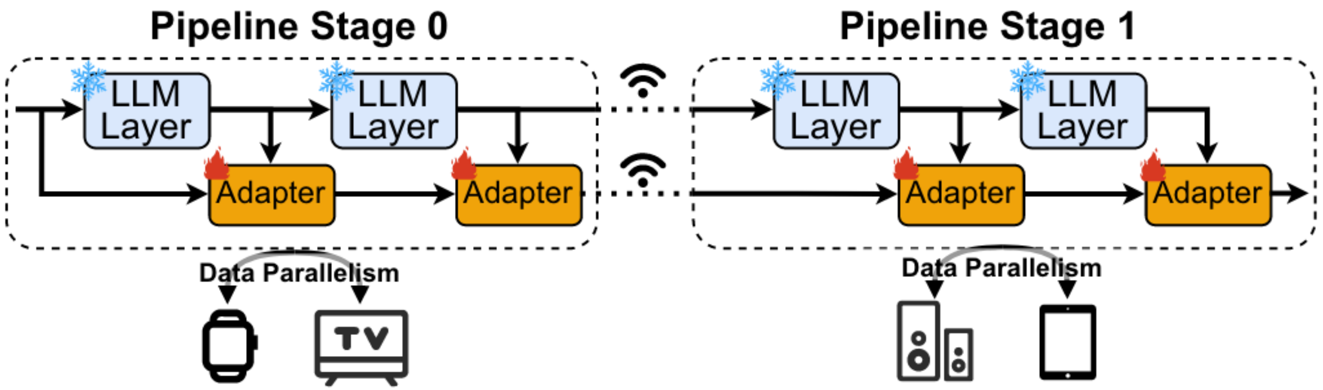
Step2： Edge devices are allocated into several device groups, each comprising one or more devices. PAC maps each stage to a group, with the stage model replicated across all devices within that group.

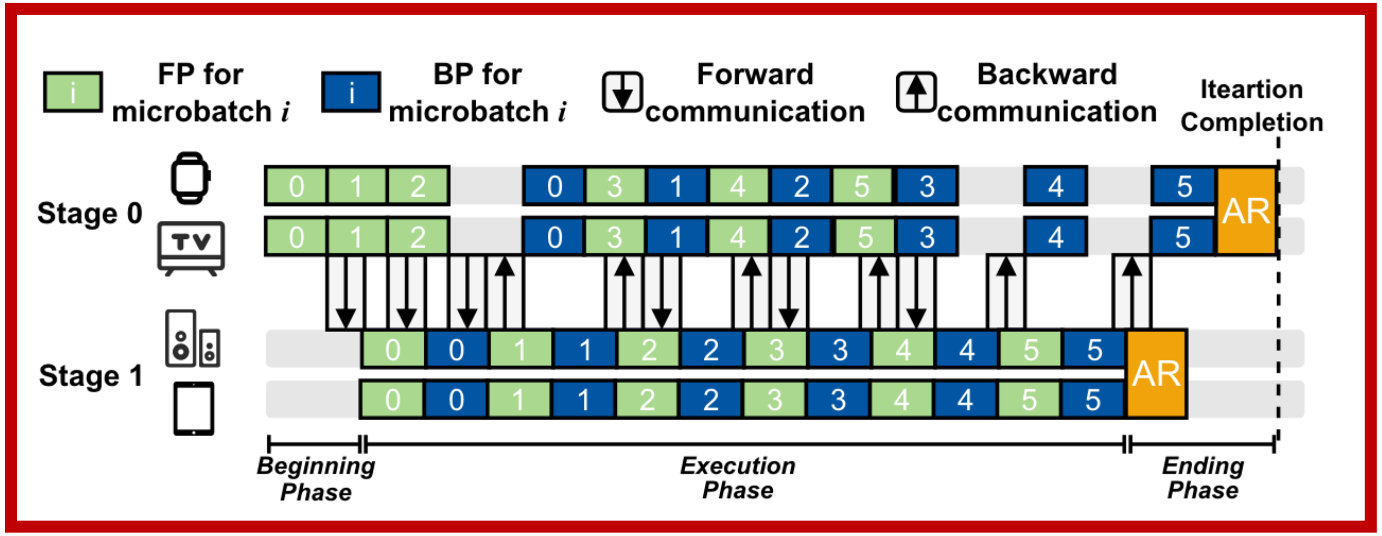


PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Data & Pipeline Hybrid Parallelism for LLMs Fine-Tuning

Step3 ： A mini-batch is divided into several micro-batches for concurrent processing to enhance parallelism. If a device cluster hosts multiple devices, micro-batches are further subdivided.

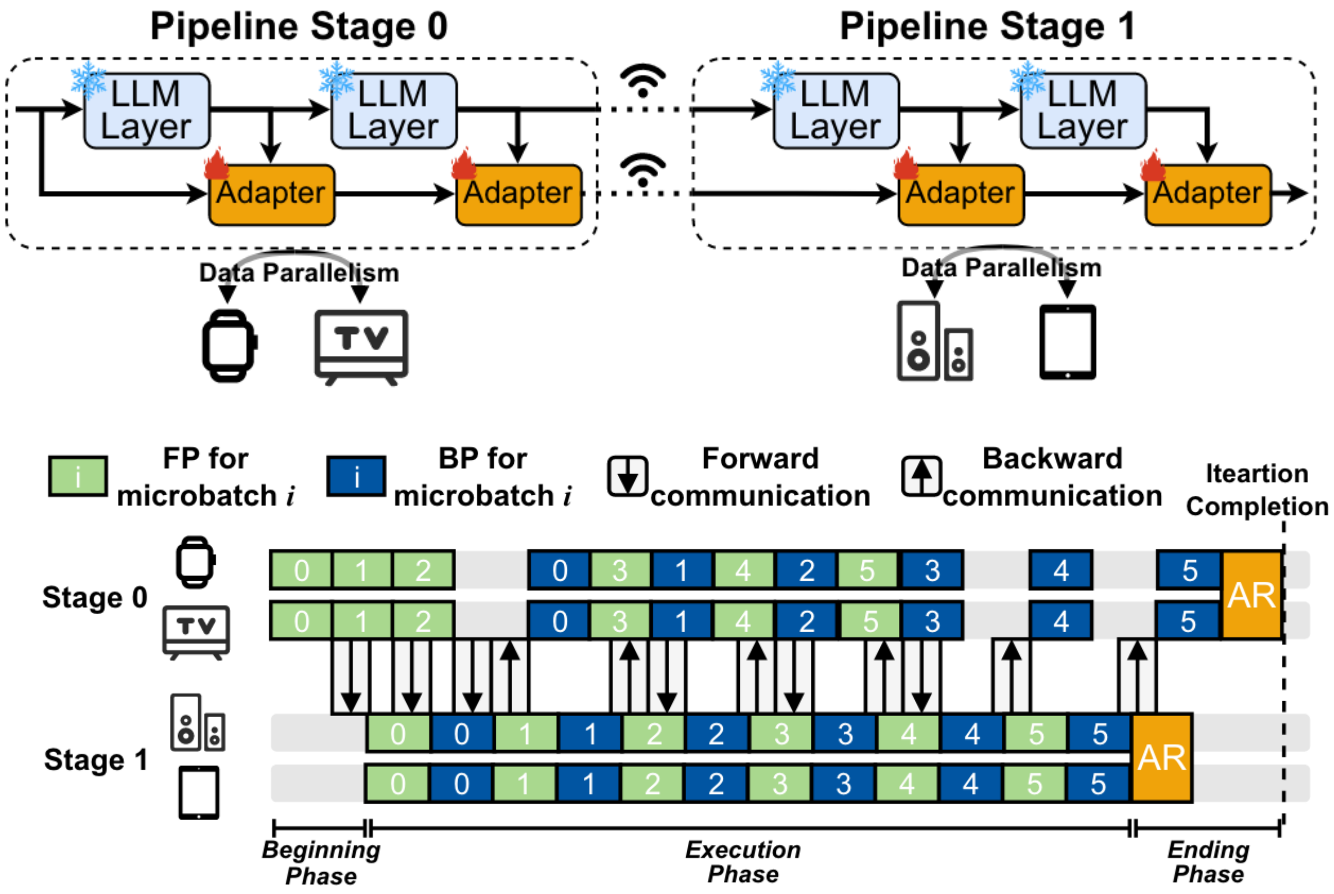




PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Data & Pipeline Hybrid Parallelism for LLMs Fine-Tuning

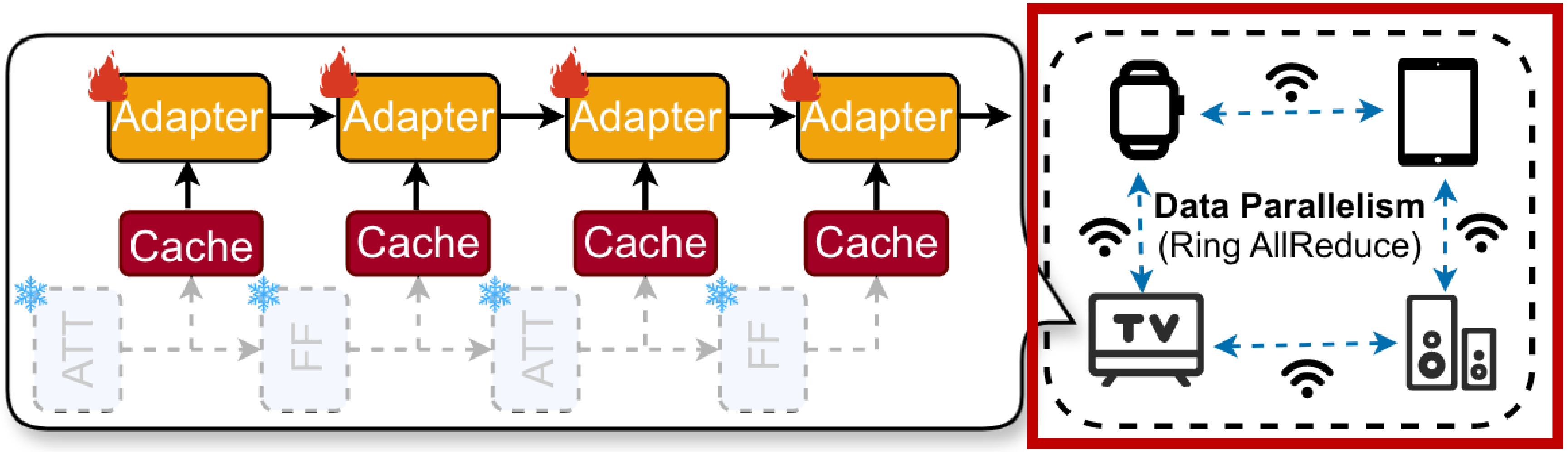
We design a dynamic programming algorithm to search for the optimal partitioning method and device grouping method for LLMs.



PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Cache-Enabled Collaborative Edge Fine-Tuning of Parallel Adapters

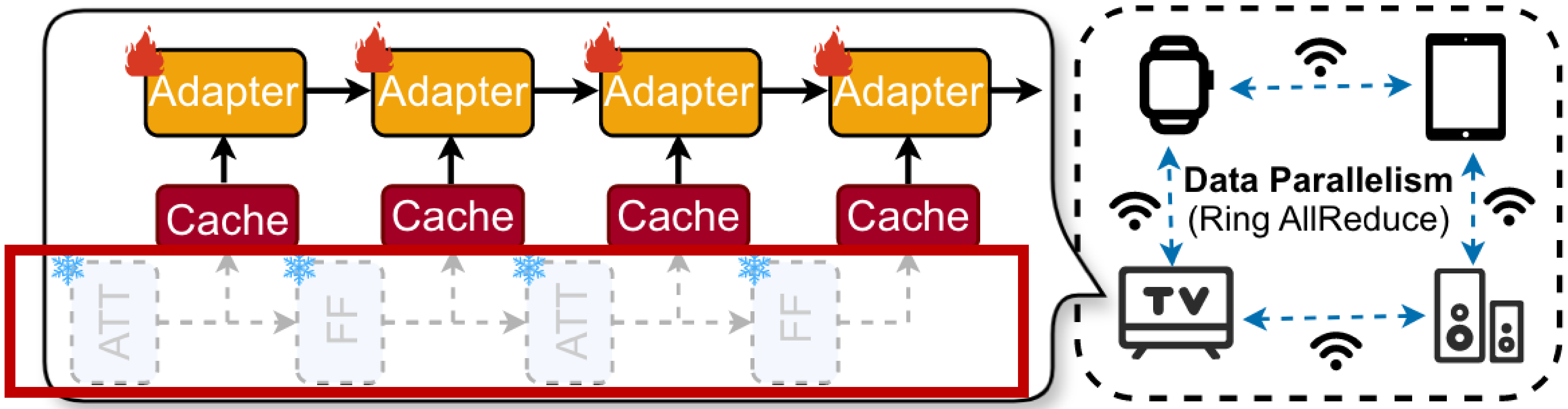
Step 1: Perform collective communication to redistribute the Parallel Adapters' parameters and locally cached activations across all devices.



PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Cache-Enabled Collaborative Edge Fine-Tuning of Parallel Adapters

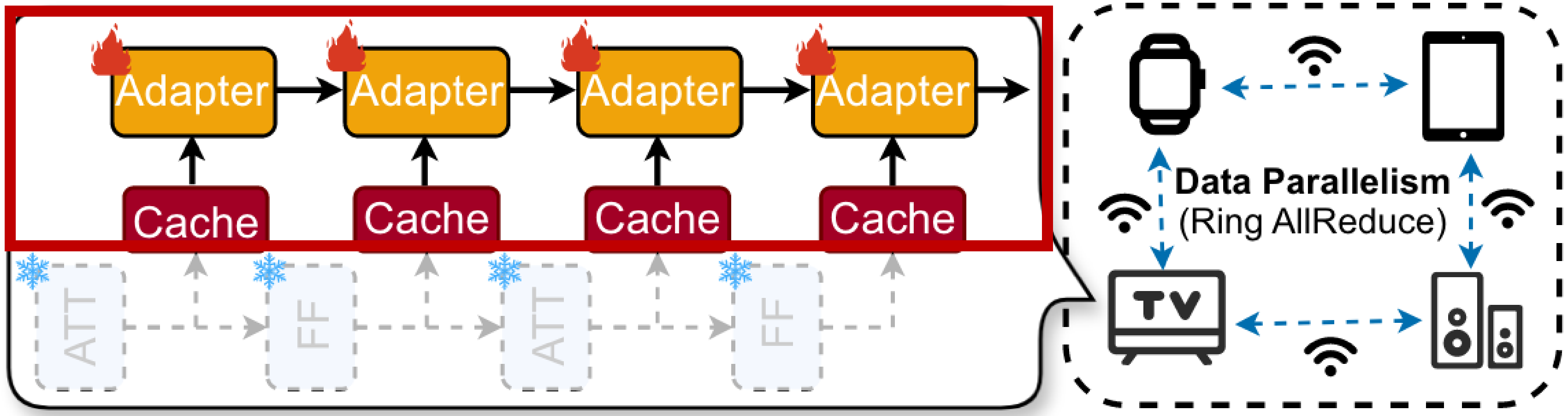
Step 2: Release the memory usage of the backbone model by simply loading parallel adapters for fine-tuning.



PAC: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-Tuning

➢ Cache-Enabled Collaborative Edge Fine-Tuning of Parallel Adapters

Step 3: The devices then utilize cached activations to fine-tune the parallel adapters in a data-parallel manner.



EVALUATION

➢ Implementation and Setups

- Models:

Model	Structure	Layers	Heads	Hidden Size	Param. Count
T5-Base [20]	en-de	12	12	768	0.25B
BART-Large [13]	en-de	12	16	1024	0.41B
T5-Large [20]	en-de	24	16	1024	0.74B

- Edge Environment Setup:
  - 8 NVIDIA Jetson Nanos
  - network bandwidth: 1000Mbps

EVALUATION

➢ Implementation and Setups

- Baseline Methods:
  - Standalone + Full model fine-tuning/Adapters/LoRA
  - Eco-FL (ICPP 2022) + Full model fine-tuning/Adapters/LoRA
  - EDDL (SEC 2021) + Full model fine-tuning/Adapters/LoRA

EVALUATION

➢ End-to-end Performance

- PAC accelerates fine-tuning up to **8.64×** faster than existing state-of-the-art methods.

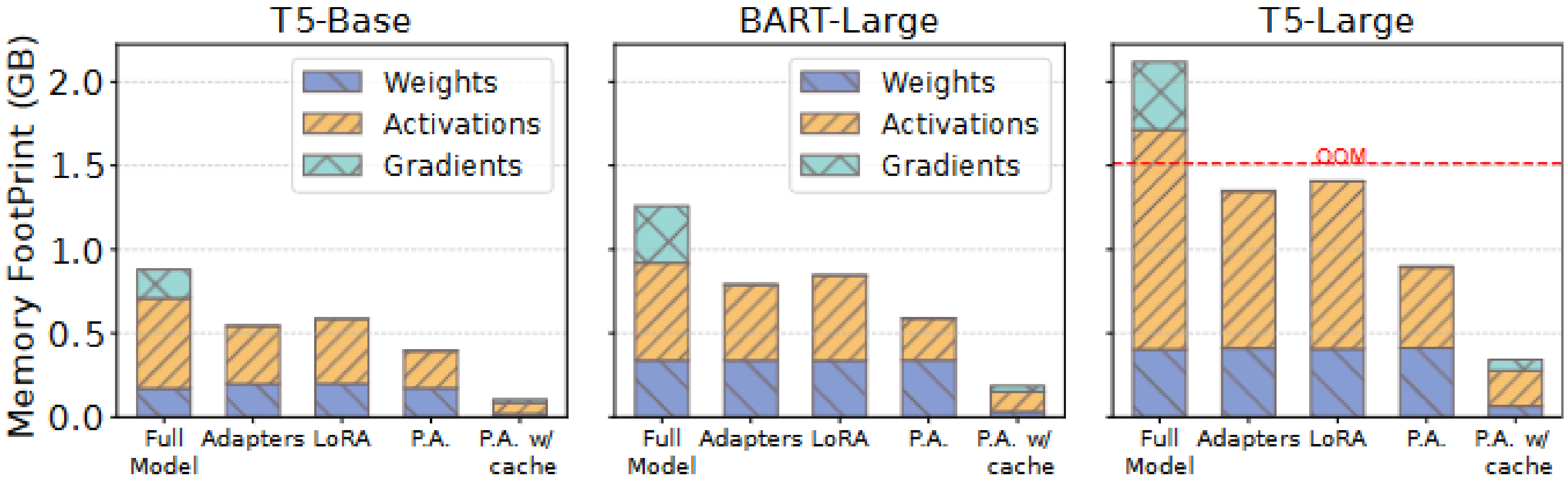
Table 2: Training durations (in hours) for different methods: 3 epochs for MRPC and STS-B, and 1 epoch for SST-2 and QNLI.

Fine-tuning Techniques	Baseline Methods	T5-Base				BART-Large				T5-Large			
		MRPC	STS-B	SST-2	QNLI	MRPC	STS-B	SST-2	QNLI	MRPC	STS-B	SST-2	QNLI
Full Model	Standalone	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	Eco-FL	0.45	0.71	2.74	4.32	2.41	3.78	14.56	22.98	OOM	OOM	OOM	OOM
	EDDL	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
Adapters	Standalone	1.21	1.9	7.29	11.51	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	Eco-FL	0.39	0.61	2.35	3.71	0.54	0.85	3.27	5.16	2.75	4.31	16.59	26.19
	EDDL	0.34	0.53	2.06	3.25	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
LoRA	Standalone	1.21	1.89	7.28	11.49	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	Eco-FL	0.41	0.64	2.45	3.87	0.55	0.87	3.33	5.26	2.73	4.28	16.48	26.02
	EDDL	0.31	0.48	1.86	2.94	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
Parallel Adapters	PAC (Ours)	**0.14**	**0.22**	**1.34**	**2.12**	**0.29**	**0.45**	**2.69**	**4.25**	**0.69**	**1.09**	**8.88**	**14.02**

EVALUATION

➢ End-to-end Performance

- PAC decrease the peak memory up to **88.16%** compared to baselines.



EVALUATION

➢ End-to-end Performance

- PAC can achieve **comparable or even superior** fine-tuned model performance.

Fine-tuning Techniques	T5-Base				BART-Large				T5-Large			
	MRPC	STS-B	SST-2	QNLI	MRPC	STS-B	SST-2	QNLI	MRPC	STS-B	SST-2	QNLI
Full Model	89.71	90.94	94.03	93.08	88.16	91.10	95.64	94.40	92.78	91.08	95.30	93.30
Adapters	88.73	90.51	93.58	93.04	86.63	90.24	94.93	93.27	91.86	90.58	96.10	94.07
LoRA	86.27	90.73	93.69	93.30	87.46	90.36	95.23	94.48	90.27	92.08	95.53	94.18
Mean Value	88.24	90.73	93.77	93.14	87.42	90.57	95.27	94.05	91.64	91.25	95.64	93.85
Parallel Adapters (Ours)	88.24	90.43	93.46	93.25	87.71	90.54	95.25	93.68	91.7	91.57	95.76	93.7
Difference from Mean	+0.00	-0.30	-0.31	+0.11	+0.29	-0.03	-0.02	-0.37	+0.06	+0.32	+0.12	-0.15

Thanks for listening

Bei Ouyang[★1], Shengyuan Ye[★1], Liekang Zeng[2], Tianyi Qian[1], Jingyi Li[1], Xu Chen[†1]

[1]**Sun Yat-sen University**, [2]HKUST(GZ)

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)