中山大學

SUN YAT-SEN UNIVERSITY

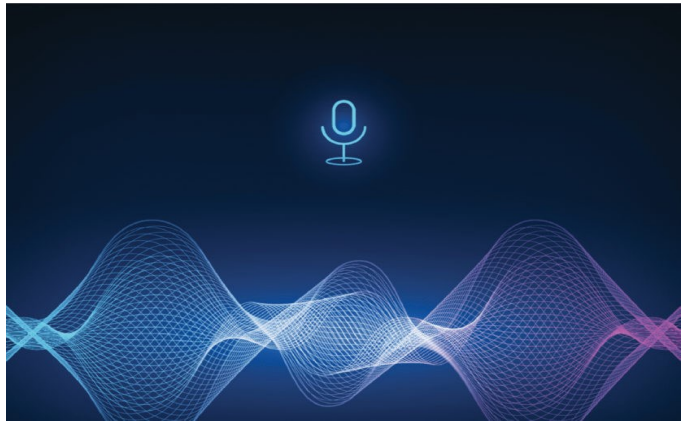
# 泛在算力网络中的协同边缘智能计算系统

## Collaborative Edge AI System in Ubiquitous Computing Network

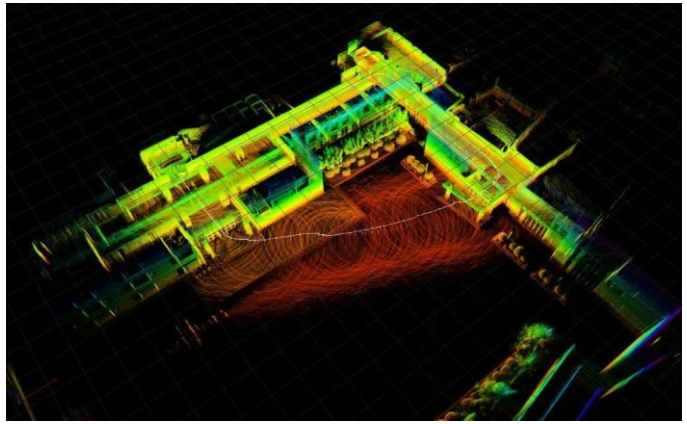
# 智能边缘应用

- 基于深度学习，尤其是大语言模型(LLM)的边缘智能应用受到广泛的关注





个人AI助手

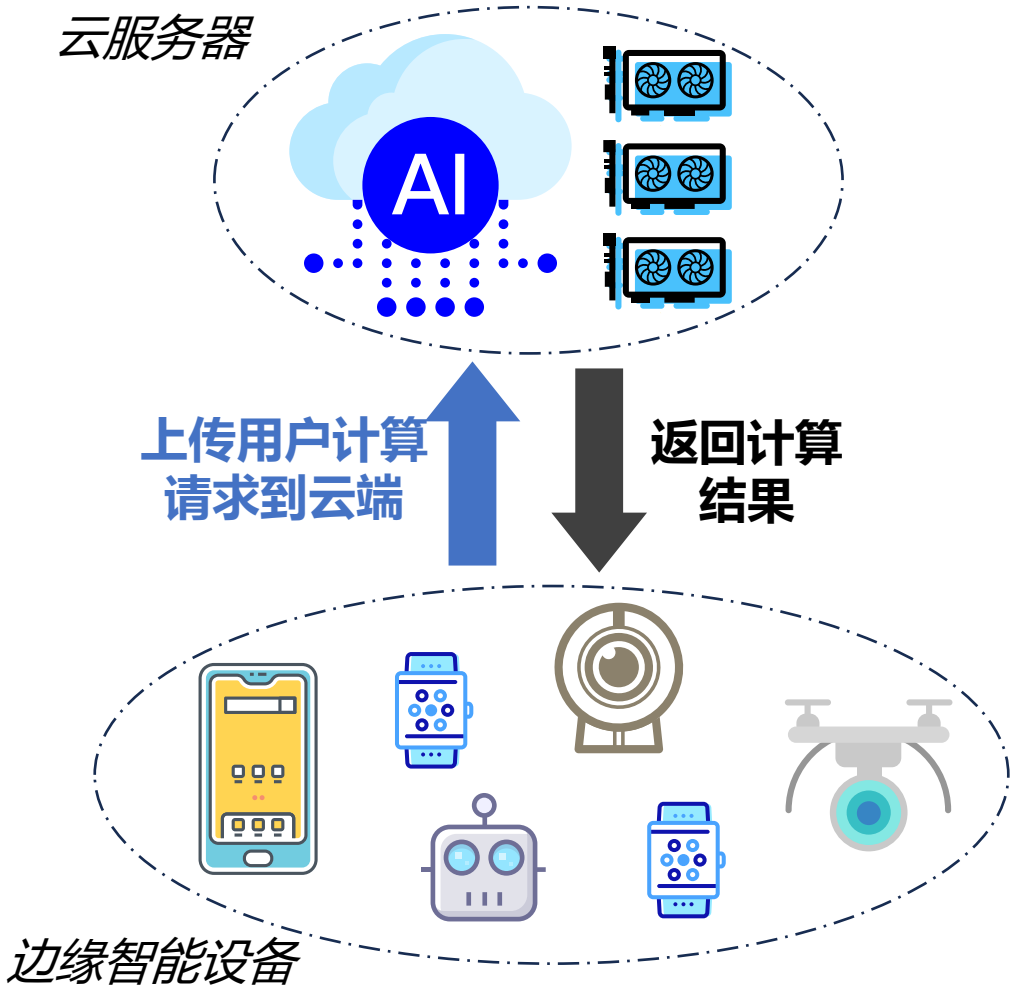


智能机器人/飞行器

AR & VR 应用

# 基于云的边缘智能应用部署

- 当前的基于深度学习的边缘智能应用部署，严重依赖于云服务




## 云服务的优势

- 强大的且可拓展的计算资源

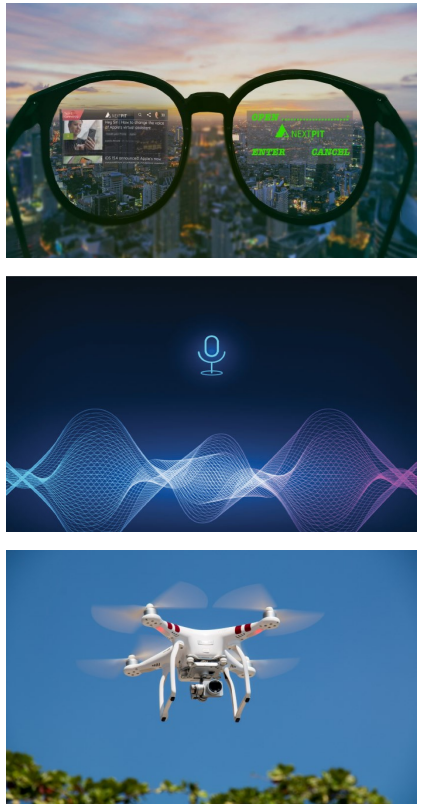
## 云服务的三个致命问题

- 数据隐私安全问题
- 高延迟的广域网连接
- 数据中心和主干网络面临巨大压力

# 基于边缘端的本地边缘智能应用部署

- **在边缘设备上进行本地部署成为了边缘智能应用的新范式**

基于深度学习的边缘智能应用



部署

边缘智能设备

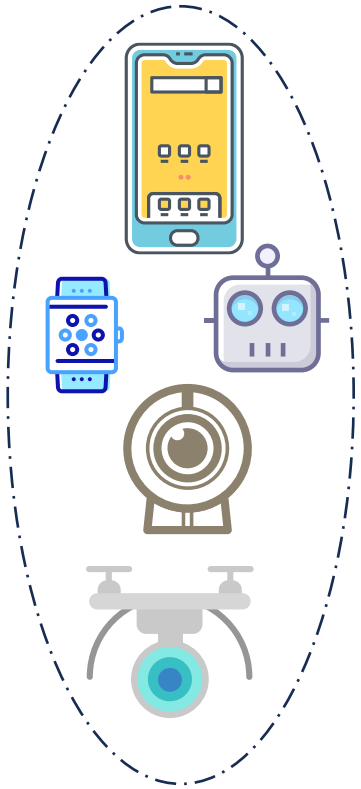


TABLE I
INFERENCE LATENCY AND MEM. FOOTPRINT OF TRANSFORMER MODELS

| 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 |

LLM推理与A100有121x 的时延差距.

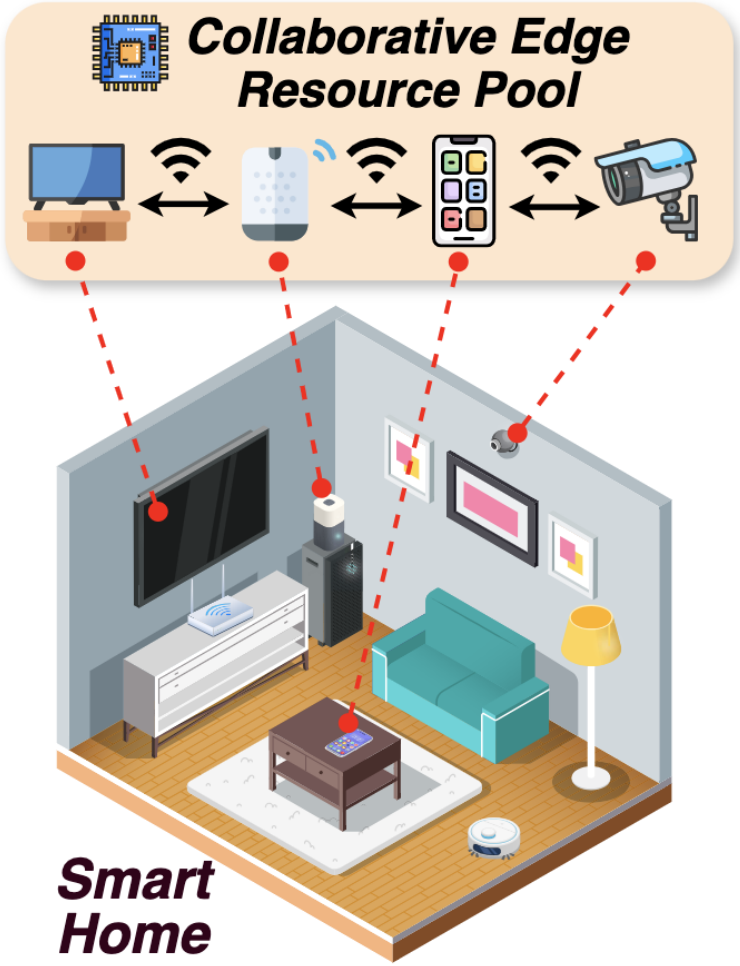
部署LLMs模型发生内存溢出.

- **保护数据隐私安全**
- **避免了通过广域网传输**
- **有限且无法拓展的计算资源(算力，内存，IO带宽...)**

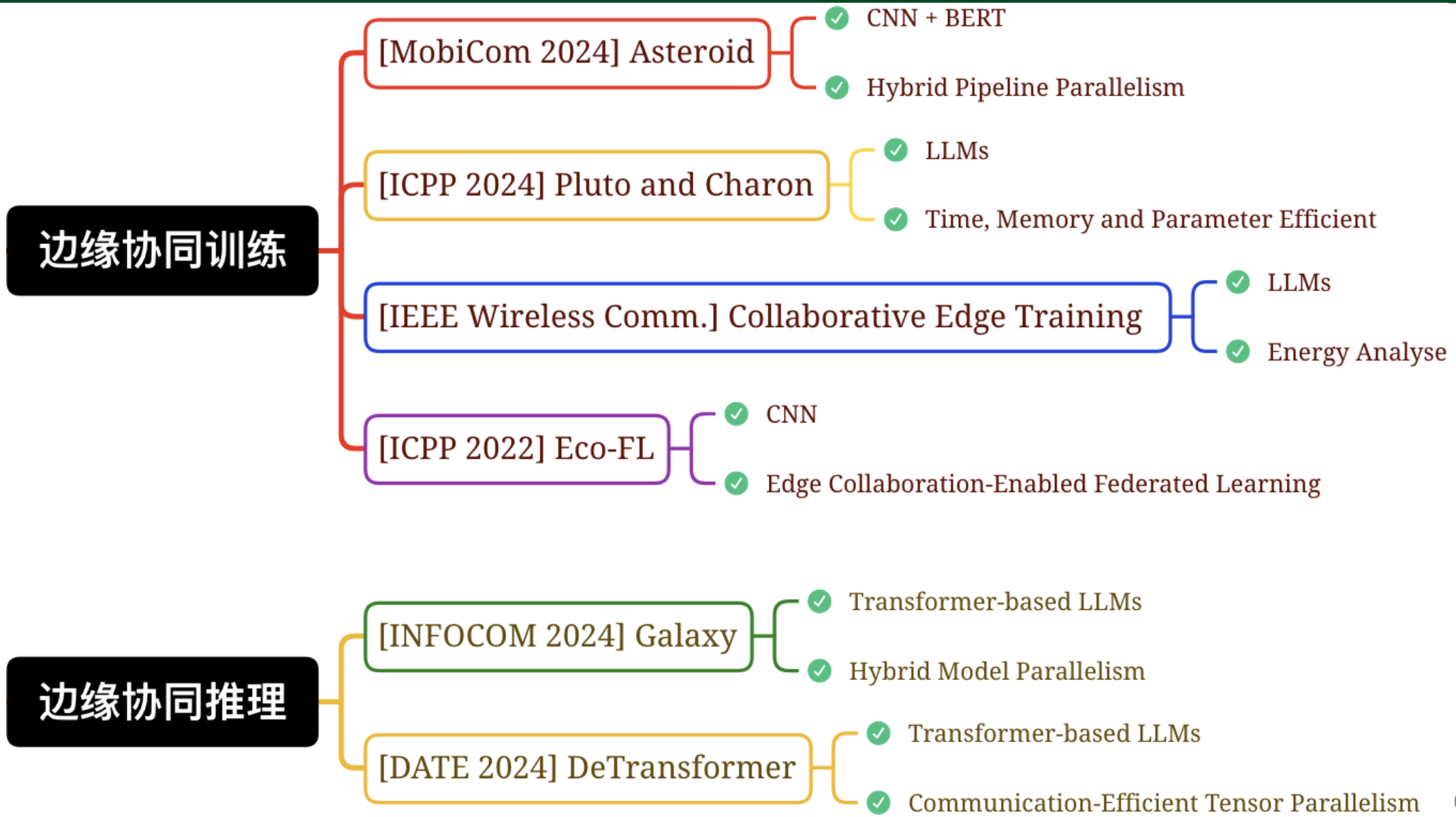
# 协同边缘智能计算 (Collaborative Edge Computing)

## 机会

- ✓ 常见的边缘环境（如：智能家庭），通常包含多个受信任的设备通过局域网相互连接，例如同一家人拥有的手机、笔记本电脑和智能家居设备等。
- ✓ 利用统一边缘环境中通过局域网高速连接的边缘设备进行资源扩展与池化，从而提高基于深度学习的边缘智能系统在本地的服务性能与质量。



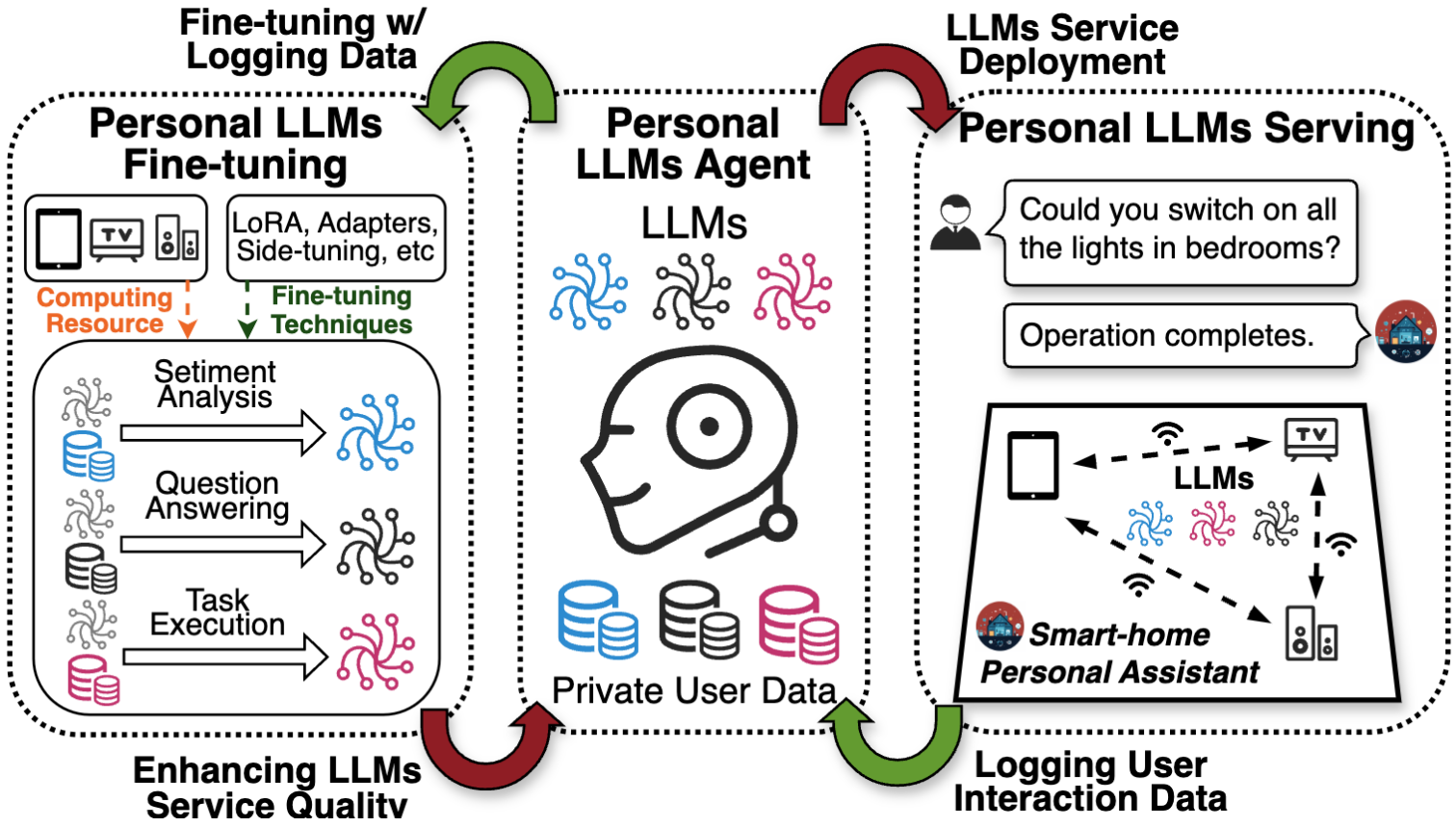
# 协同边缘智能计算 (Collaborative Edge Computing)



# 基于协同边缘计算的智能家庭助理本地部署框架

## 工作流程

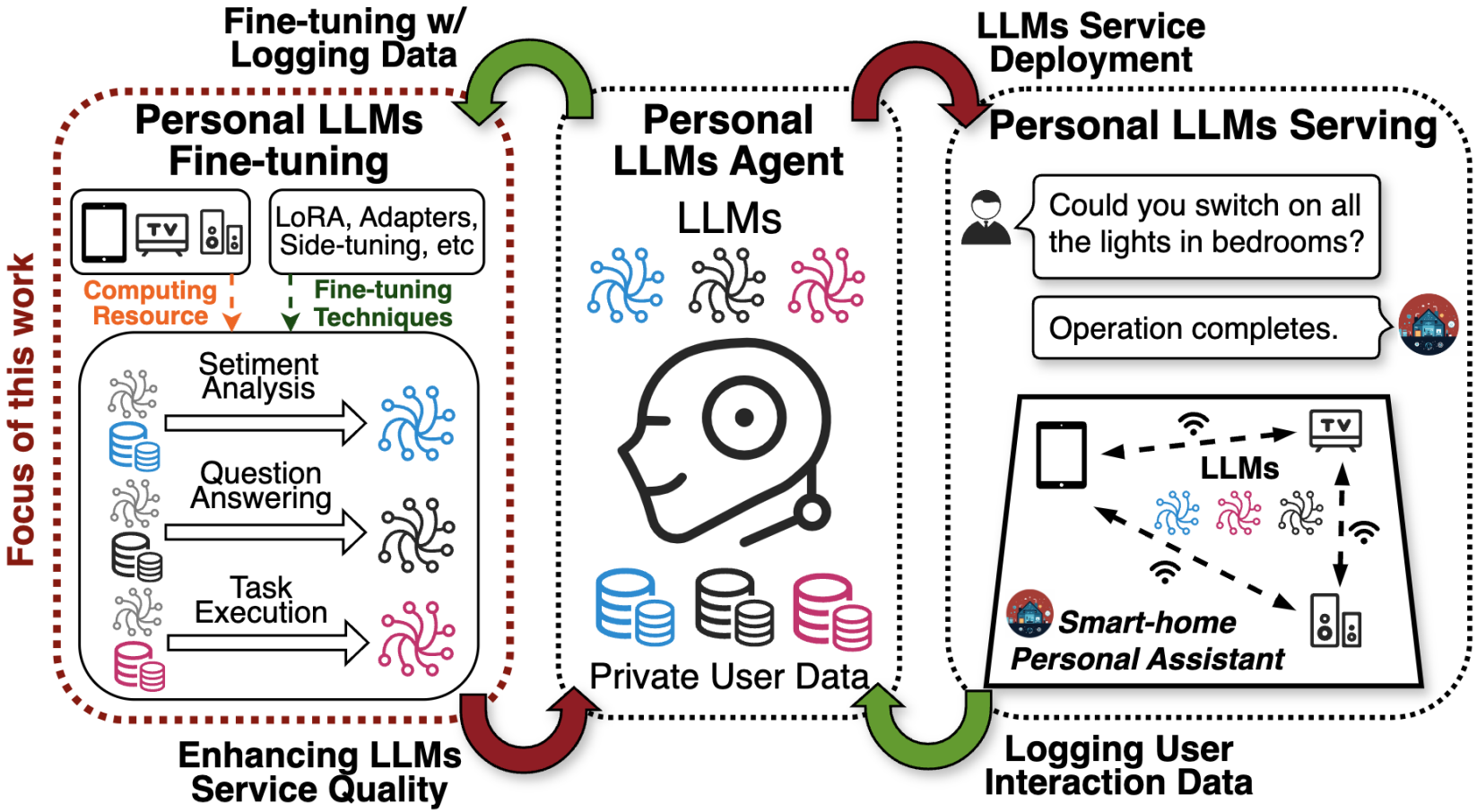
- ✓ 使用协同边缘设备共同支撑大语言模型的部署。
- ✓ 利用收集的用户本地数据对大语言模型进行**个性化微调**。
- ✓ 部署个性化大语言模型**推理服务**，为家庭成员提供智能应用。同时收集服务过程中产生的用户数据，周期性的对大语言模型进行微调以提升性能。



# 边缘协同资源高效的大语言模型微调

## 研究问题

- ✓ **问题一：** 如何使用资源受限的边缘设备进行时间和内存高效的大语言模型训练？
- ✓ **问题二：** 如何编排多个可用的边缘计算设备以及划分工作负载，以实现资源高效的分布式协同大语言模型微调？



# 边缘协同资源高效的大语言模型微调

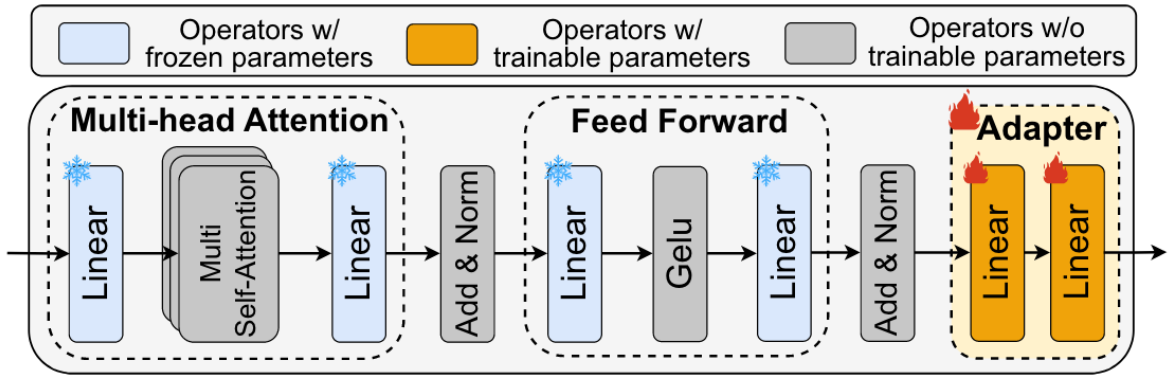
## 研究问题

- ✓ **问题一：** 如何使用资源受限的边缘设备进行时间和内存高效的大语言模型微调？
- ✓ **问题二：** 如何编排多个可用的边缘计算设备以及划分工作负载，以实现资源高效的分布式协同大语言模型微调？

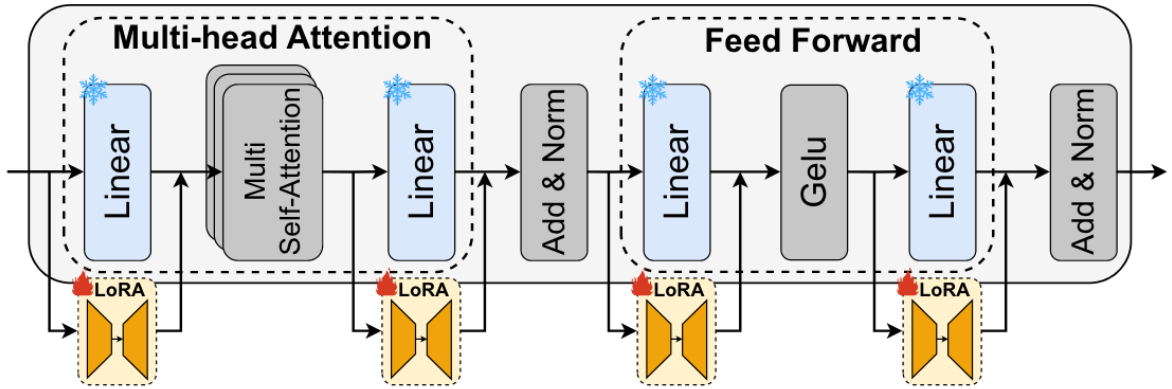
**要同时克服这两个研究问题，我们需要探索系统和算法的协同设计**

# 算法角度

✓ 目前最主流的参数高效微调技术（例如：LoRA, Adapters），对于资源受限的边缘设备**并不足够**的时间与内存高效。




(a) The transformer layer structure of Adapters.




(b) The transformer layer structure of LoRA.

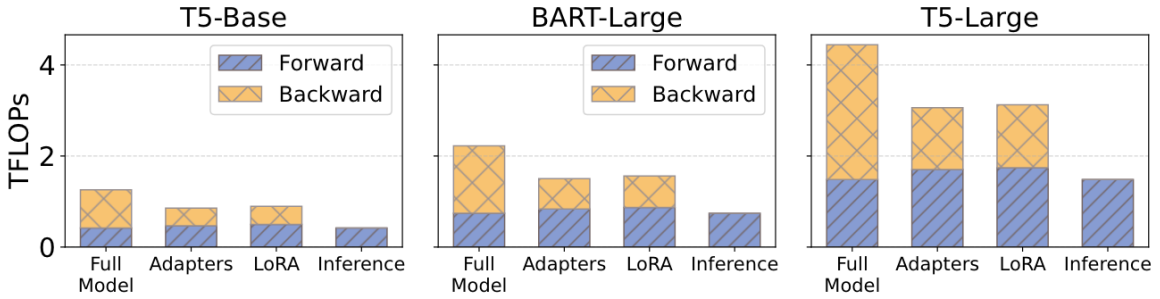



**Figure 3: The comparison of floating point of operations (FLOPs). Mini-batch size: 16; sequence length: 128.**

| 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 |

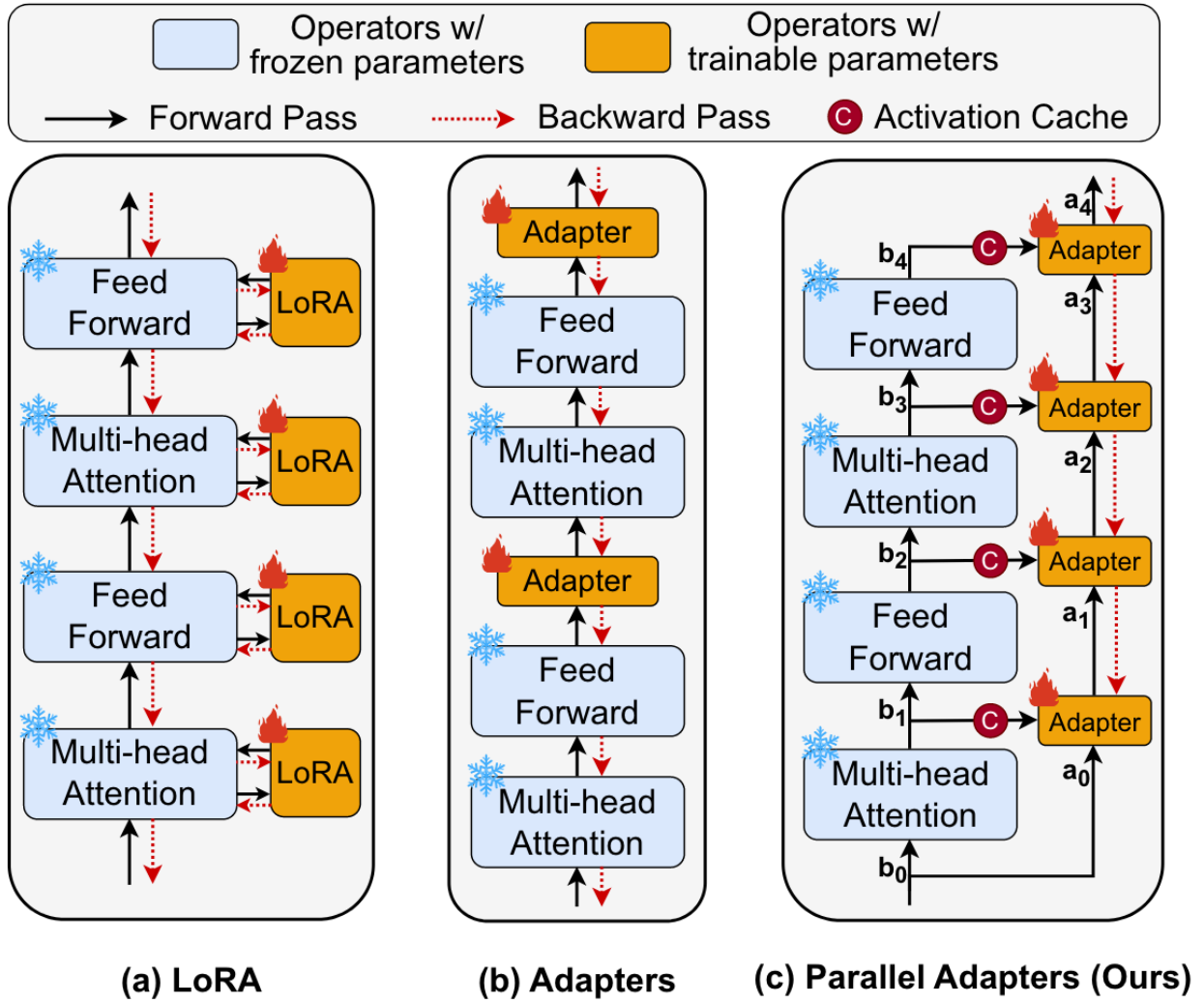
**Table 1: The breakdown of memory footprint. "Activations" contain the intermediate results and optimizer states. Model: T5-Large; mini-batch size: 16; sequence length: 128.**

# 算法角度

## ✓ Parallel Adapters LLMs微调算法

- 大语言模型主干网络参数被冻结。
- 在主干网络旁构建一个并行的轻量级子网络，用于参数高效的微调**（可以避免对主干网络进行反向传播！）**
- 主干网络推理产生的激活结果可以被缓存下来以重用**（可以避免对主干网络进行前向传播！）**

避免了对LLMs主干模型进行前向传播和反向传播，显著降低了计算资源的需求！



# 系统角度

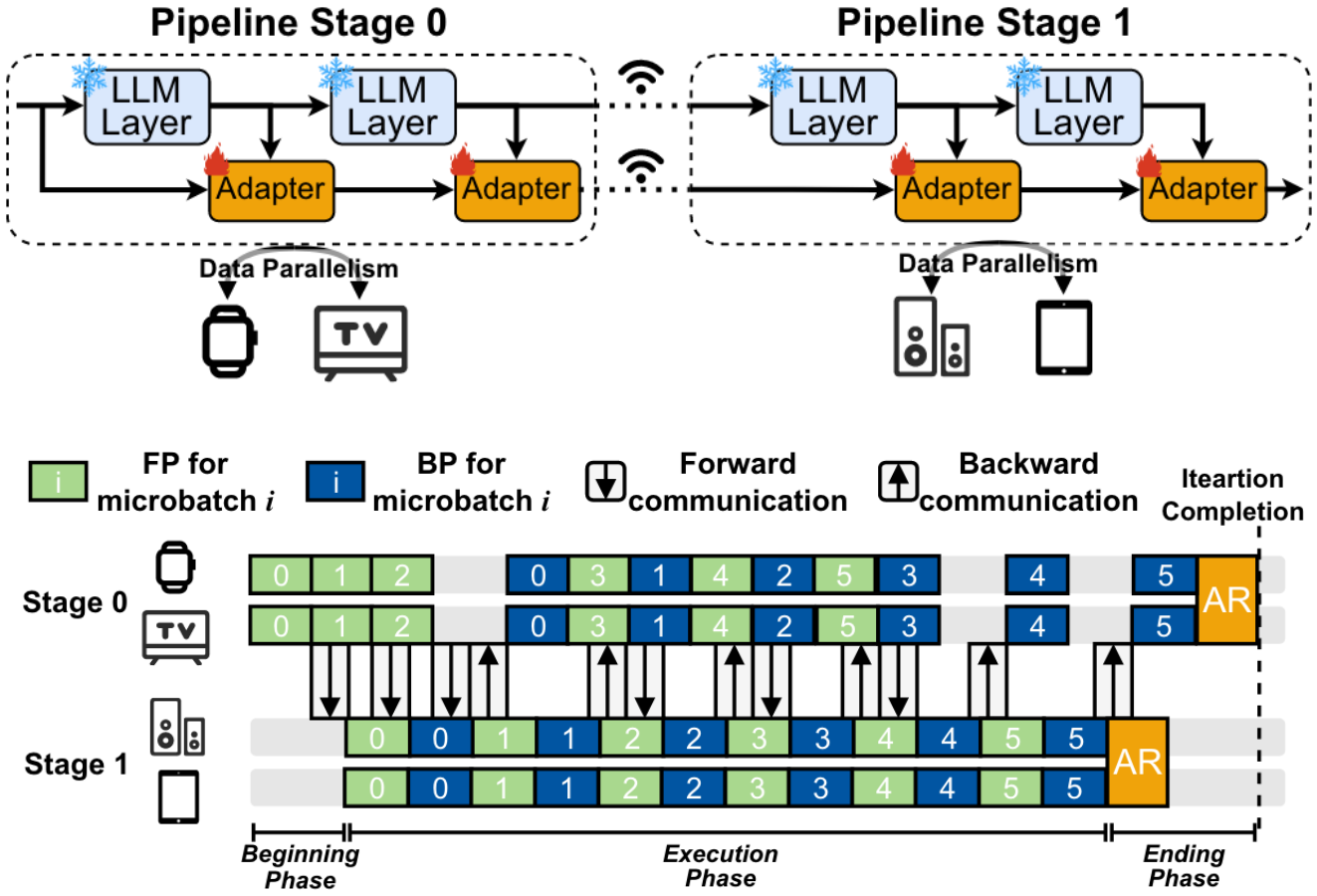
## ✓ 数据&流水线混合并行LLMs微调

1. 步骤一：将LLMs切割为多个流水线阶段，每个阶段包含一个阶段子模型。

2. 步骤二：将边缘设备分组，并将每一组关联到不同的流水线阶段上。

3. 步骤三：向流水线中注入微调数据。在设备组间进行流水线并行训练，在设备组内进行数据并行训练。

采用动态规划算法来搜索最优的大语言模型划分方式和设备分组方法



# 微调框架性能展示

✓ 端到端框架性能对比基线方法有最高**8.64倍**的微调加速

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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** |

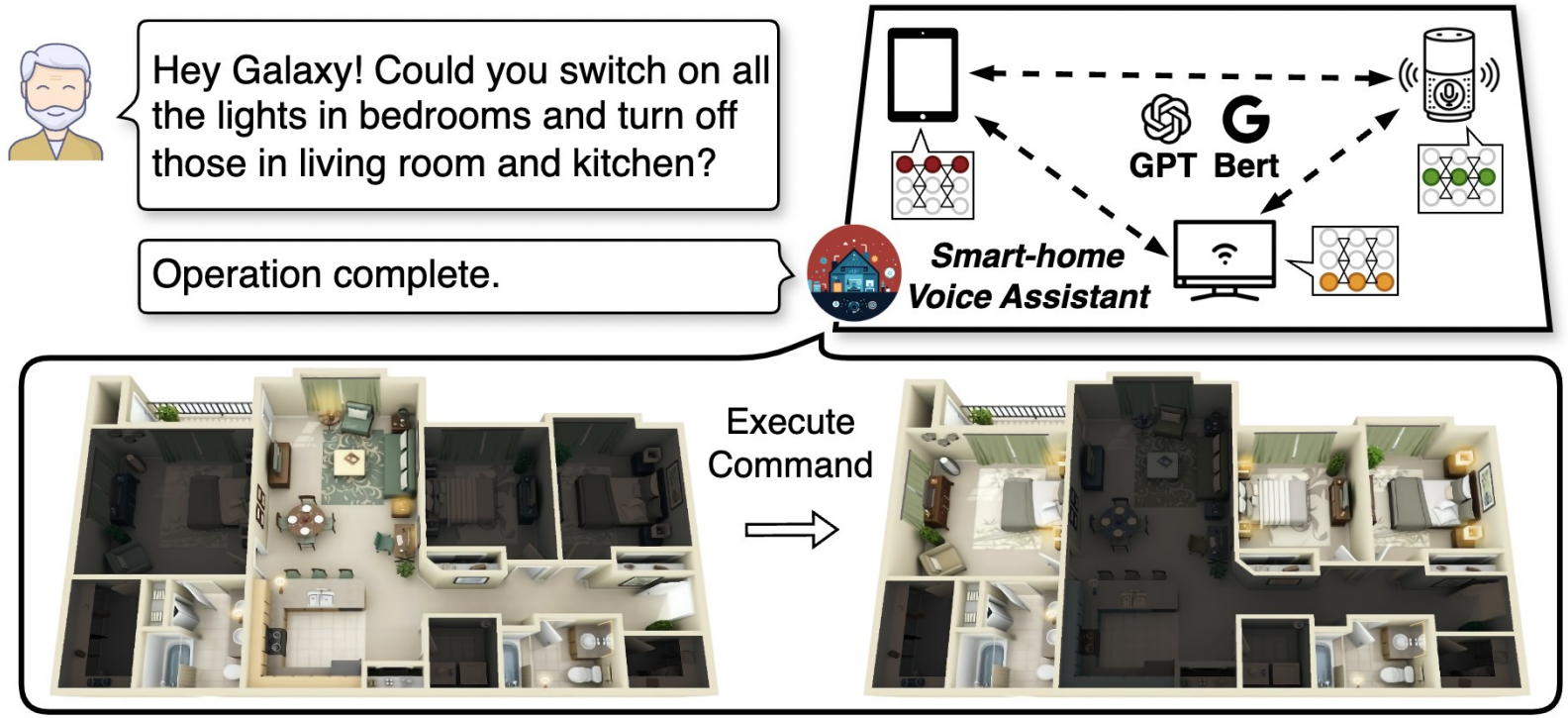
✓ 对比目前主流的LLMs微调算法，可以取得**非常相近甚至更优**的微调后模型性能。

| 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 |

# 基于混合模型并行的协同边缘LLMs推理框架

## 研究问题

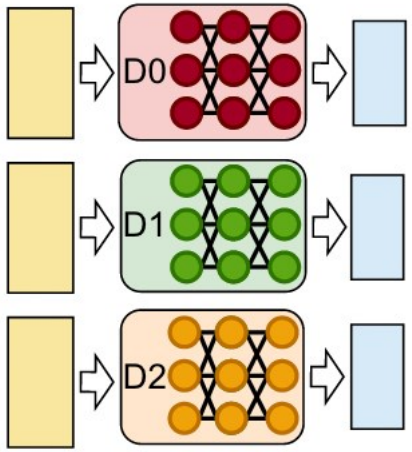
- ✓ **问题一：**如何选择并行计算架构，来协同多个边缘设备进行资源高效的分布式Transformer大语言模型推理。
- ✓ **问题二：**如何在带宽受限的边缘网络环境下进行资源高效的分布式Transformer大语言模型推理。



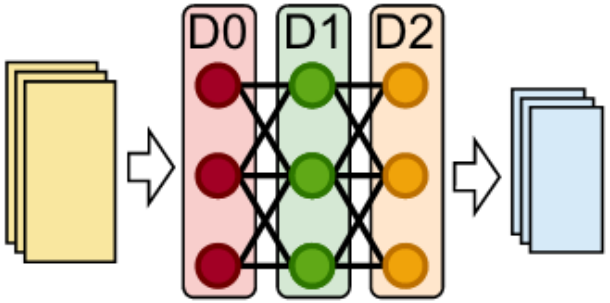
# 基于混合模型并行的协同边缘LLMs推理框架

**研究问题一：**如何选择最合适的模型并行架构？

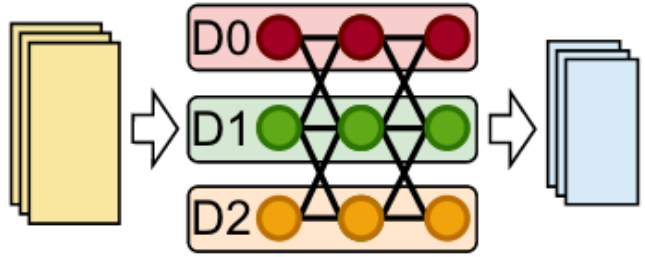
- ✓ 边缘端的推理任务经常是包含**单样本**的用户请求，例如：一句打开开关的指令。
  - 数据并行和流水线并行在处理多样本请求时才能并行利用多台计算设备资源。
- ✓ 模型并行是intra-operator级别切分，支持多台边缘设备共同推理单个样本。
  - 主流的Transformer模型并行技术包括：**张量模型并行**和**序列模型并行**。




(a) 数据并行推理 ✗




(b) 流水线并行推理 ✗



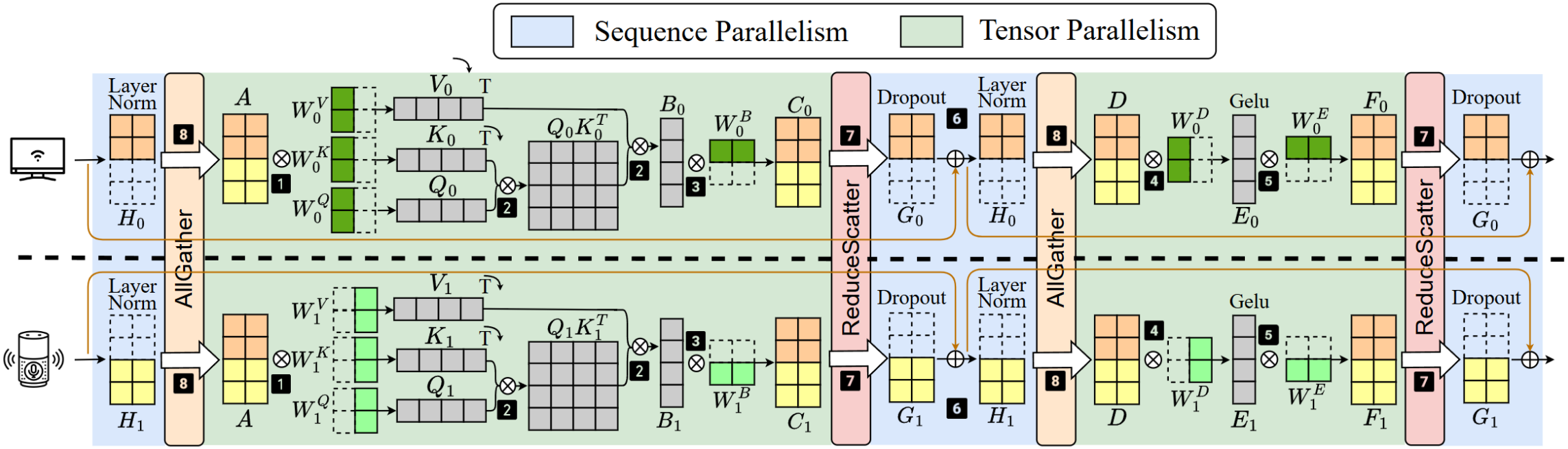

(c) 模型并行推理 ✓

# 基于混合模型并行的协同边缘LLMs推理框架

✓ 使用张量模型并行+序列模型并行的**混合模型并行技术**：

- 对Attention模块和MLP模块采用张量并行。
- 对于两者之间的连接部分采用序列并行

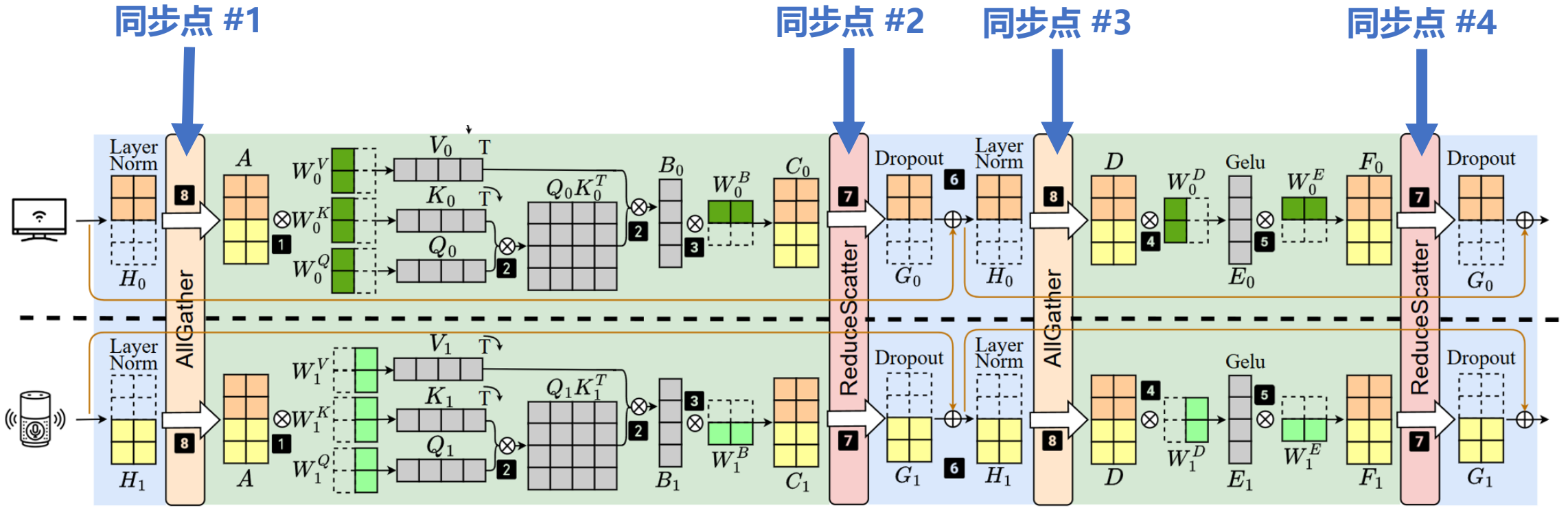
采用异构资源感知的负载分配算法
均匀分配计算任务到异构设备上




两个设备的混合模型并行架构示意图

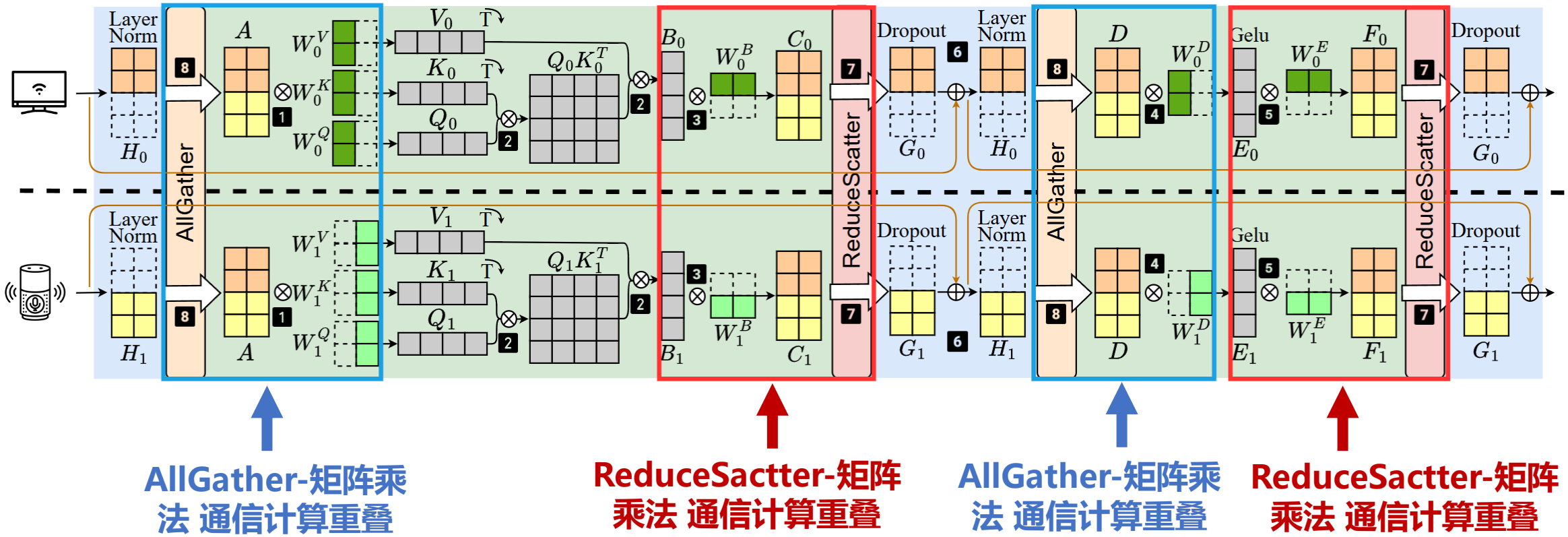
# 基于混合模型并行的协同边缘LLMs推理框架

✓ Transformer混合模型并行推理在每一层都需要进行大量的设备间张量同步通信，这成为了推理系统的性能瓶颈。



# 基于混合模型并行的协同边缘LLMs推理框架

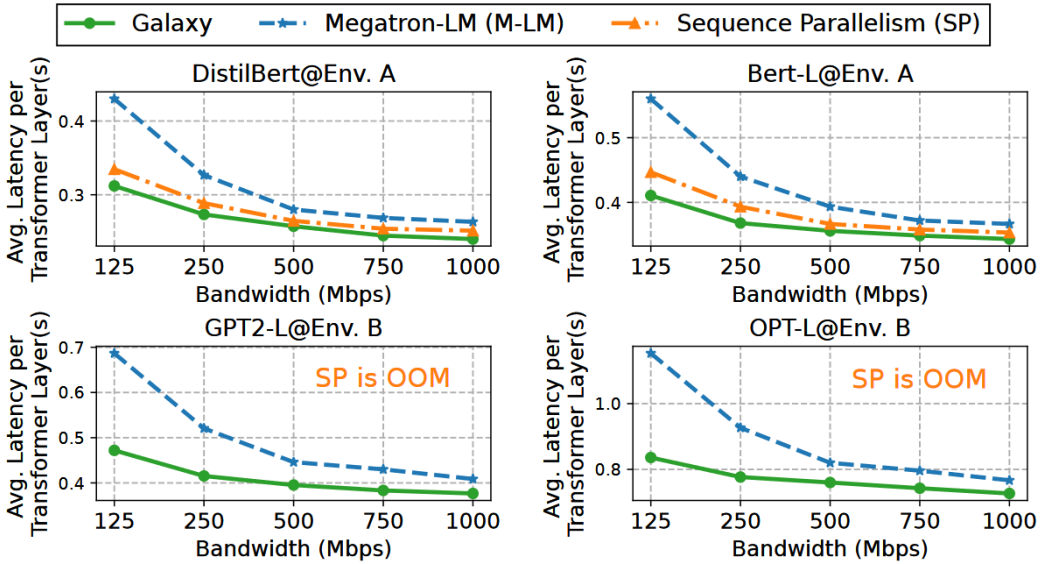
使用通信计算重叠技术来优化通信开销



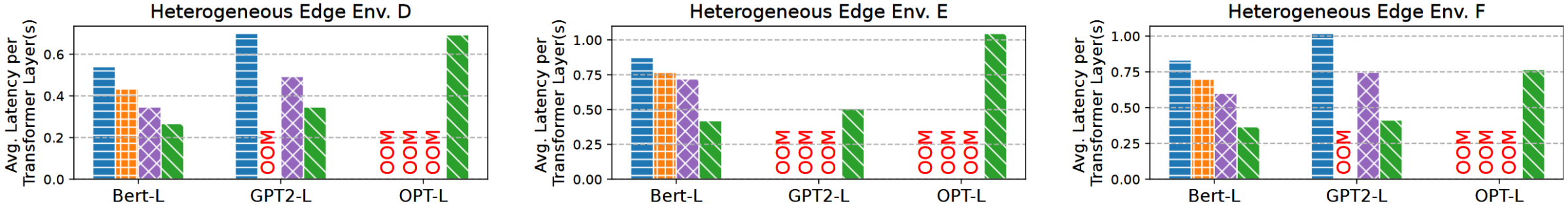
# 协同边缘推理框架性能展示

在不同的网络带宽环境下始终保持着很高的性能，与基准方法相比最高可以降低**46%**的推理延迟

| Model | Layers | Heads | Hidden Layer | Edge Env. | Speedup Over M-LM | Speedup Over SP |
|---|---|---|---|---|---|---|
| DistilBert [33] | 6 | 12 | 768 | A | 1.37× | 1.08× |
| Bert-L [1] | 24 | 16 | 1024 | A | 1.36× | 1.09× |
| | | | | B | 1.38× | 1.11× |
| GPT2-L [12] | 36 | 20 | 1280 | A | 1.31× | OOM |
| | | | | B | 1.46× | OOM |
| OPT-L [34] | 24 | 16 | 2048 | A | 1.26× | OOM |
| | | | | B | 1.40× | OOM |
| | | | | C | 1.43× | OOM |
| OPT-XL [34] | 32 | 32 | 2560 | A | OOM | OOM |
| | | | | B | OOM | OOM |
| | | | | C | 1.28× | OOM |




Galaxy在异构边缘环境中表现出色，相比于其他基准并行方法推理延迟减少了最高**2.5倍**




中山大學
SUN YAT-SEN UNIVERSITY

# 感谢！