东南大学 SOUTHEAST UNIVERSITY
计算机科学与工程学院 School of computer science and engineering

EDGE-LLM: Enabling Efficient Large Language Model Adaptation on Edge Devices via Layerwise Unified Compression and Adaptive Layer Tuning & Voting

Zhongzhi Yu[1], Zheng Wang[1], Yuhan Li[1], Haoran You[1], Ruijie Gao[1],
Xiaoya Zhou[3], Sreenidhi Reedy Bommu[1], Yang (Katie) Zhao[2], Yingyan (Celine) Lin[1]
[1]*Georgia Institute of Technology*, [2]*University of Minnesota, Twin Cities*, [3]*University of California, Santa Barbara*
{zyu401, zwang2478, yli3326, hyou37, eiclab.gatech, sbommu3, celine.lin}@gatech.edu,
yangzhao@umn.edu, xiaoyazhou@umail.ucsb.edu

【DAC'24】

分享人：夏天歌

提纲

1.1 背景

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

开发高效

持续性
保护隐私
……

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

挑战 LLM的模型规模阻碍了其在边缘设备（edge GPU；智能手机等）上的直接适配

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

挑战 LLM的模型规模阻碍了其在边缘设备（edge GPU；智能手机等）上的直接适配

1 在计算 LLM的正向和反向传播时的过量**计算开销**

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

挑战 LLM的模型规模阻碍了其在边缘设备（edge GPU；智能手机等）上的直接适配

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

挑战 LLM的模型规模阻碍了其在边缘设备（edge GPU；智能手机等）上的直接适配

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

- LLM通常在尖端GPU（40GB/80GB显存）上tuning， 且一次tuning需要花费超过1GPU×1天的计算时间

1.1 背景

- 大型语言模型(LLM)的兴起与优越性能，使得对LLM在边缘设备上的tuning技术的需求越来越大

挑战 LLM的模型规模阻碍了其在边缘设备（edge GPU；智能手机等）上的直接适配

1 在计算 LLM的正向和反向传播时的过量**计算开销**

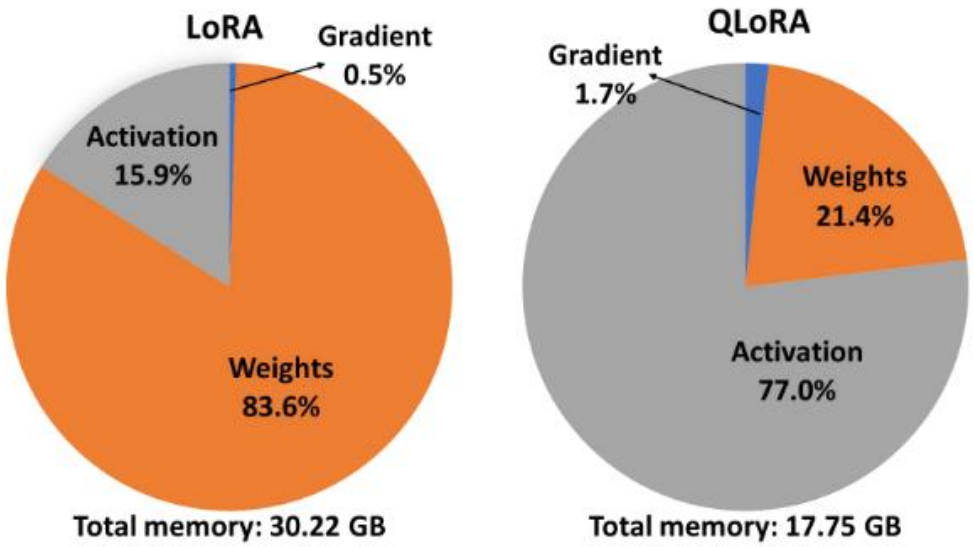
2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

- LLM通常在尖端GPU（40GB/80GB显存）上tuning， 且一次tuning需要花费超过1GPU×1天的计算时间

- 即使是SOTA的高效tuning方法，也不能在边缘设备上有效地tuning相对较小的LLM（LLAMA-7B）

1.1 背景

Figure 1. Profiling results on the memory footprint when tuning LLaMA-7B with LoRA [10] and QLoRA [2] on the Alpaca [20] dataset.



1 在计算 LLM的正向和反向传播时的过量**计算开销**

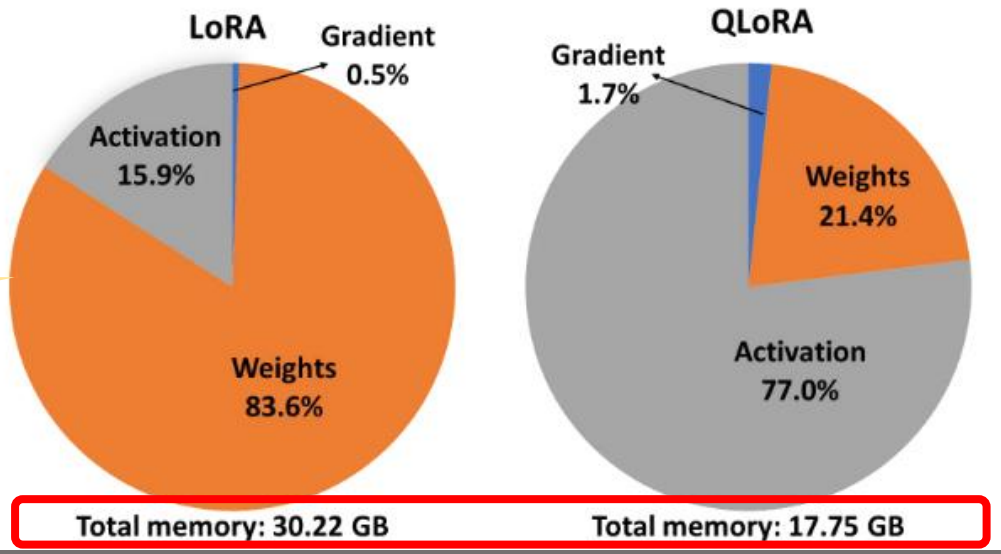
2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

- LLM通常在尖端GPU（40GB/80GB显存）上tuning，且一次tuning需要花费超过1GPU×1天的计算时间

- 即使是SOTA的高效tuning方法，也不能在边缘设备上有效地tuning相对较小的LLM（LLAMA-7B）

1.1 背景

Figure 1. Profiling results on the memory footprint when tuning LLaMA-7B with LoRA [10] and QLoRA [2] on the Alpaca [20] dataset.



TX2只有8 GB显存
Quest Pro只有12 GB显存

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

- LLM通常在尖端GPU（40GB/80GB显存）上tuning，且一次tuning需要花费超过1GPU×1天的计算时间

- 即使是SOTA的高效tuning方法，也不能在边缘设备上有效地tuning相对较小的LLM（LLAMA-7B）

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

缩短反向传播的深度

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

缩小了LLM更新的范围，限制了其可实现的性能

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

1、分层联合压缩（Layer-wise Unified Compression）

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

缩小了LLM更新的范围，限制了其可实现的性能

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

1、分层联合压缩（Layer-wise Unified Compression）

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

缩小了LLM更新的范围，限制了其可实现的性能

2、自适应层tuning（Adaptive Layer Tuning and Voting）

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

缩小了LLM更新的范围，限制了其可实现的性能

1、分层联合压缩（Layer-wise Unified Compression）

2、自适应层tuning（Adaptive Layer Tuning and Voting）

软件层面

1.2 现有工作

1 在计算 LLM的正向和反向传播时的过量**计算开销**

2 Tuning过程中存储大量模型权重和激活信息，带来的大量**内存开销**

压缩目标LLM，减少其模型大小

Qlora [arXiv 23]

Kim et al. [NeurIPS'24]

缩短反向传播的深度

Lst [NeurIPS'22]

Llama-adapter [arXiv 23]

不能很好的达成**有效减少LLM的冗余**与**保持其适应性**之间的均衡

缩小了LLM更新的范围，限制了其可实现的性能

1、分层联合压缩（Layer-wise Unified Compression）

2、自适应层tuning（Adaptive Layer Tuning and Voting）

软件层面

3、互补的硬件调度模块

硬件层面

提纲

2 问题描述

2 问题描述

- 如何优化LLM在边缘设备上tuning时的表现？

2 问题描述

- 如何优化LLM在边缘设备上tuning时的表现？
- 系统输入：

  待tuning的LLM模型　　搭载模型的边缘设备
- 系统输出：

  模型tuning后更新完的权重

2 问题描述

- 如何优化LLM在边缘设备上tuning时的表现？
- 系统输入：

  待tuning的LLM模型　　搭载模型的边缘设备
- 系统输出：

  模型tuning后更新完的权重
- 考察指标：

  计算开销（计算用时ms）　　内存开销（使用内存MB）

2 问题描述

- 如何优化LLM在边缘设备上tuning时的表现？
- 系统输入：

  待tuning的LLM模型　　搭载模型的边缘设备
- 系统输出：

  模型tuning后更新完的权重
- 考察指标：

  计算开销（计算用时ms）　　内存开销（使用内存MB）

降低LLM冗余 ⟷ 保留LLM推理精度　　减少反向传播深度冗余 ⟷ 保留LLM调整能力

2 问题描述

- 如何优化LLM在边缘设备上tuning时的表现？
- 系统输入：

  待tuning的LLM模型　　搭载模型的边缘设备
- 系统输出：

  模型tuning后更新完的权重
- 考察指标：

  计算开销（计算用时ms）　　内存开销（使用内存MB）

降低LLM冗余 ⟷ 保留LLM推理精度　　减少反向传播深度冗余 ⟷ 保留LLM调整能力

1、分层联合压缩（Layer-wise Unified Compression）

2、自适应层tuning（Adaptive Layer Tuning and Voting）

3、互补的硬件调度模块

提纲

3.1 分层联合压缩（Layer-wise Unified Compression）

3.1 分层联合压缩（Layer-wise Unified Compression）

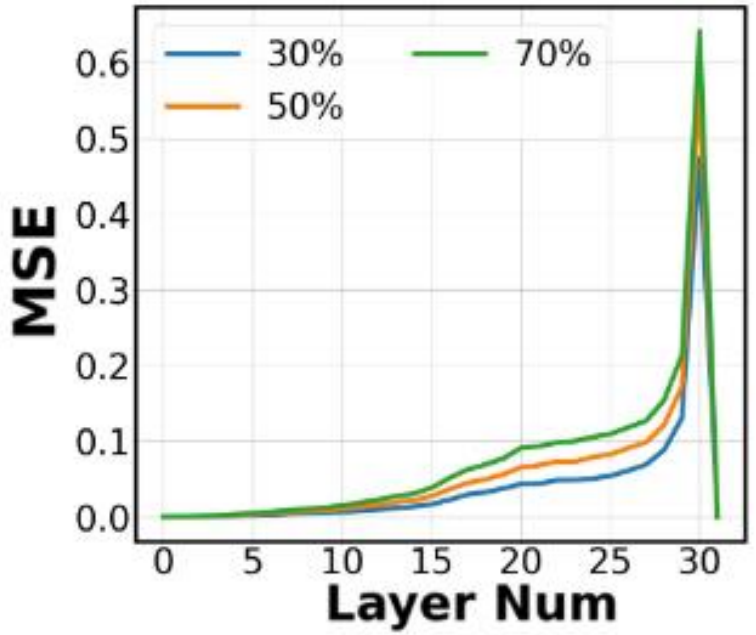


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

3.1 分层联合压缩（Layer-wise Unified Compression）

输入WikiText数据集，修剪或量化后的LLM目标层与原始层的输出间的平均MSE（即均方误差）

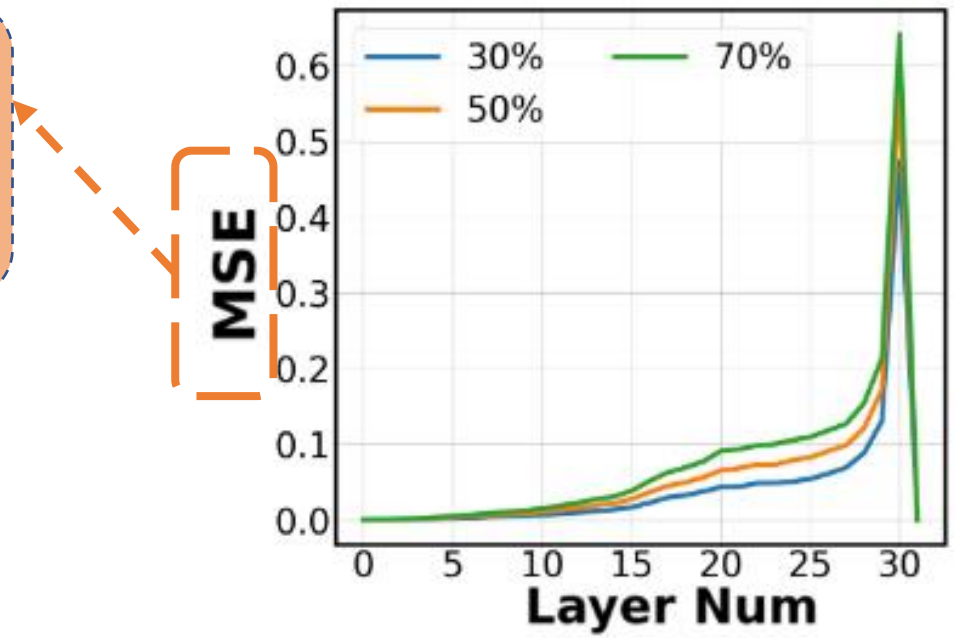


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

3.1 分层联合压缩（Layer-wise Unified Compression）

输入WikiText数据集，修剪或量化后的LLM目标层与原始层的输出间的平均MSE（即均方误差）

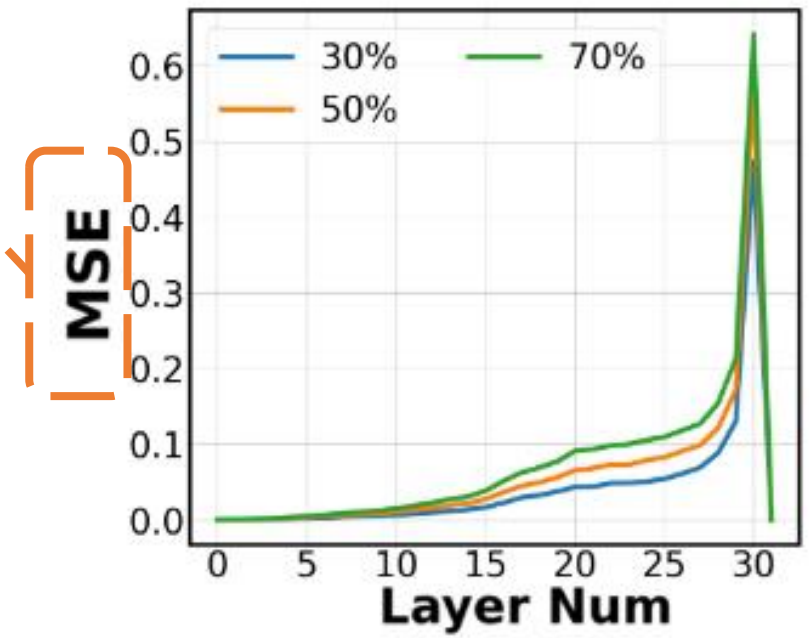
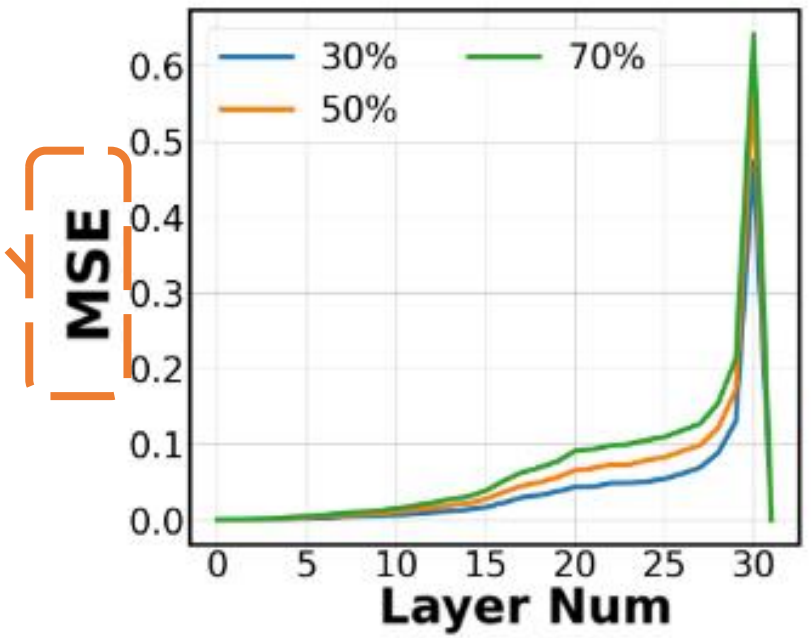


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

3.1 分层联合压缩（Layer-wise Unified Compression）

输入WikiText数据集，修剪或量化后的LLM目标层与原始层的输出间的平均MSE（即均方误差）



- LLM中只有一小部分层对压缩具有较高的敏感性

Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

3.1 分层联合压缩（Layer-wise Unified Compression）

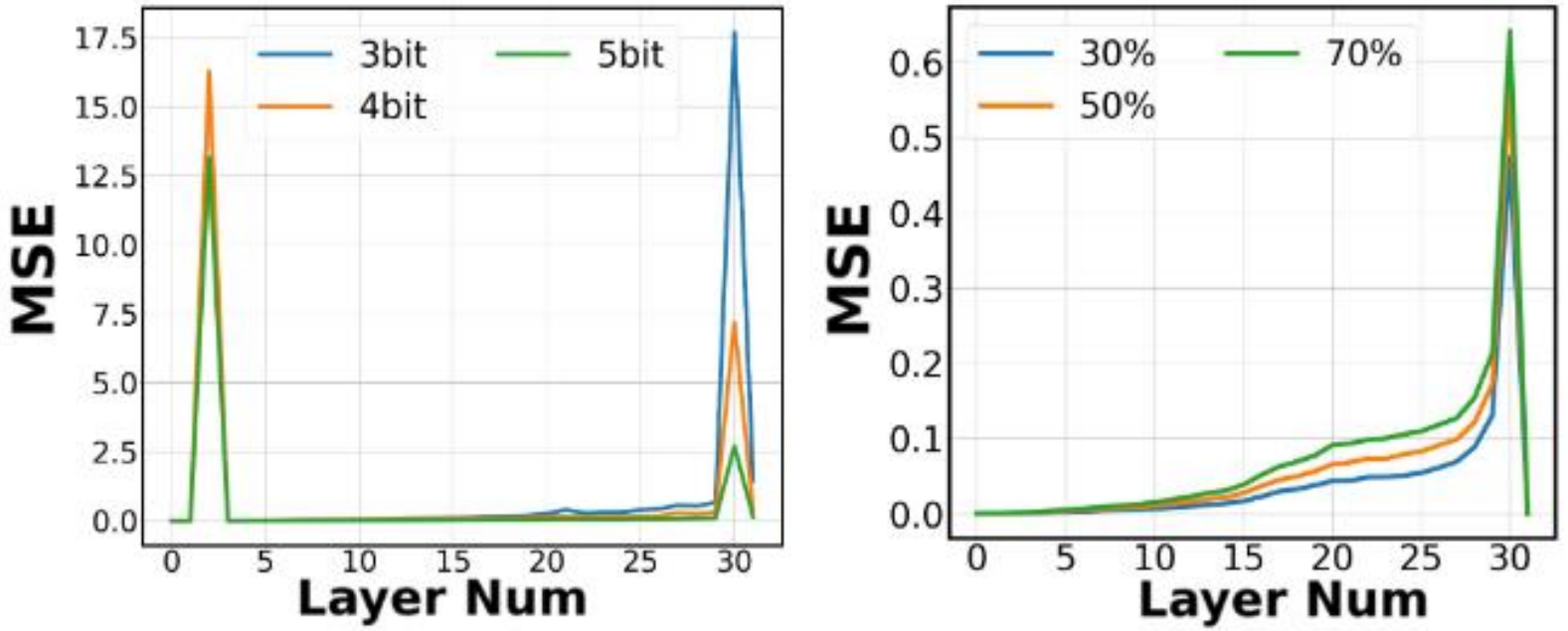


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

- LLM中只有一小部分层对压缩具有较高的敏感性

3.1 分层联合压缩（Layer-wise Unified Compression）

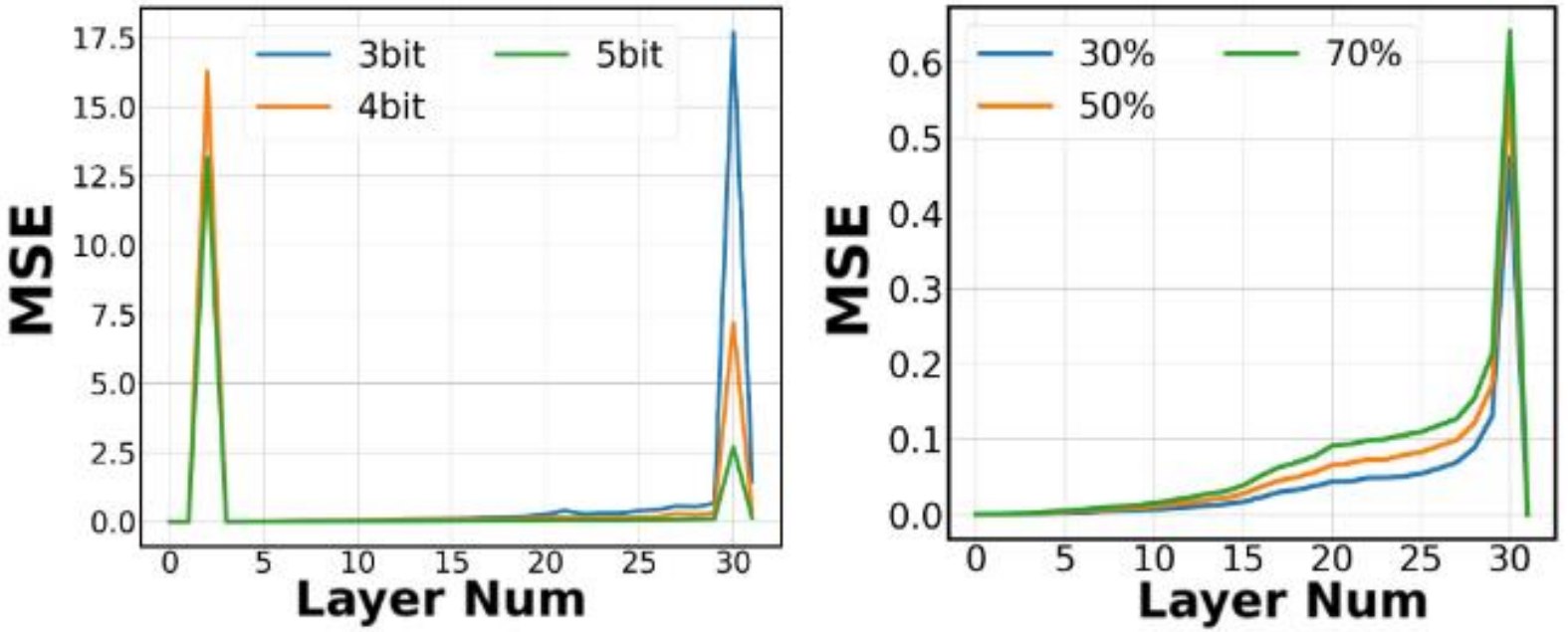


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

- LLM中只有一小部分层对压缩具有较高的敏感性
- LLM的层对于量化位宽和修剪稀疏度的敏感性位于不同维度

3.1 分层联合压缩（Layer-wise Unified Compression）

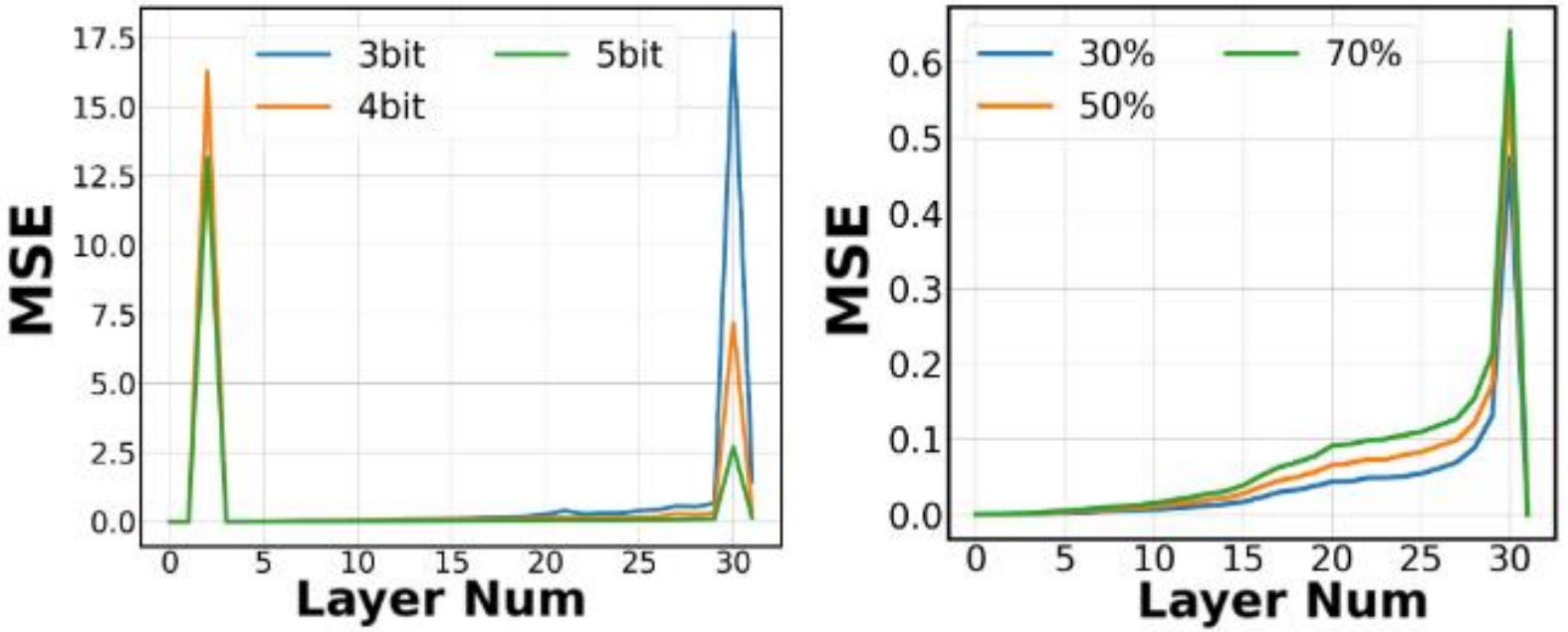
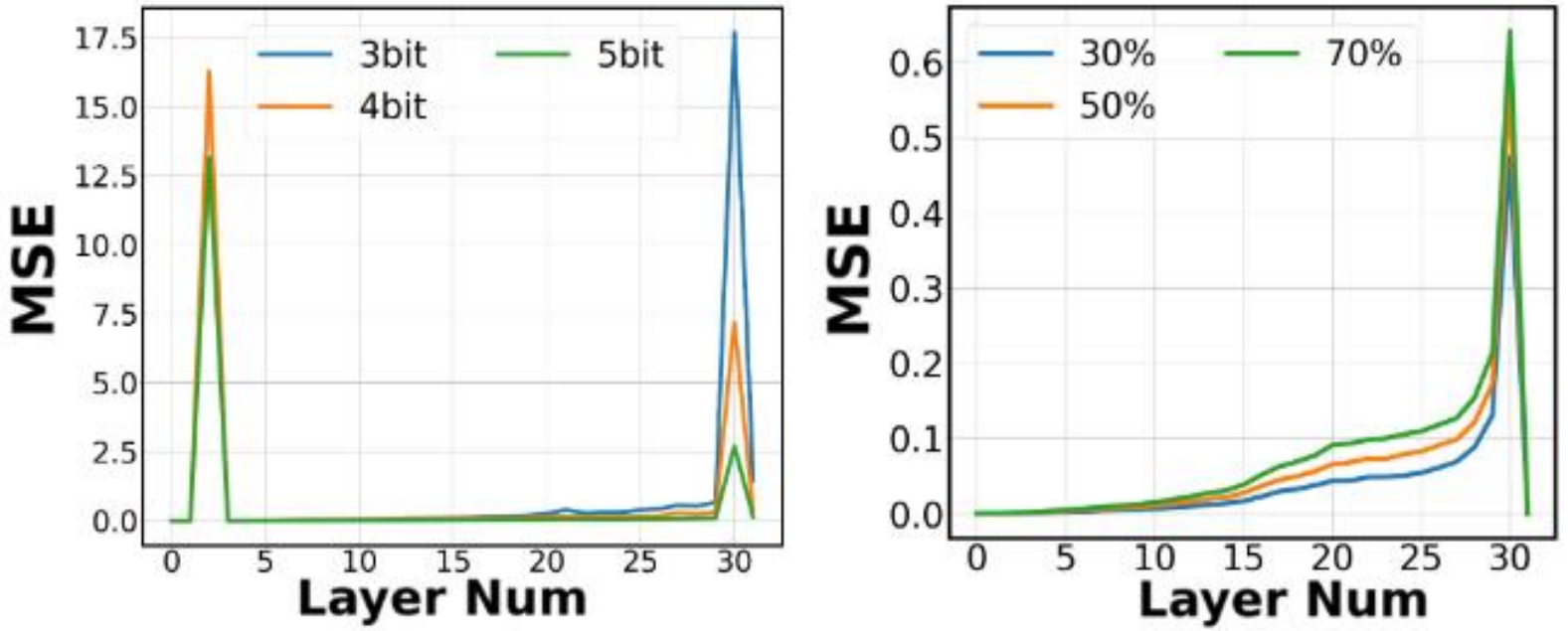


Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

$$b_j = B + \mathbb{1}(s^j_{quant} \geq \frac{\sum_{i=0}^{L-1} s^i_{quant}}{L}), \qquad (1)$$

- LLM中只有一小部分层对压缩具有较高的敏感性
- LLM的层对于量化位宽和修剪稀疏度的敏感性位于不同维度

3.1 分层联合压缩（Layer-wise Unified Compression）



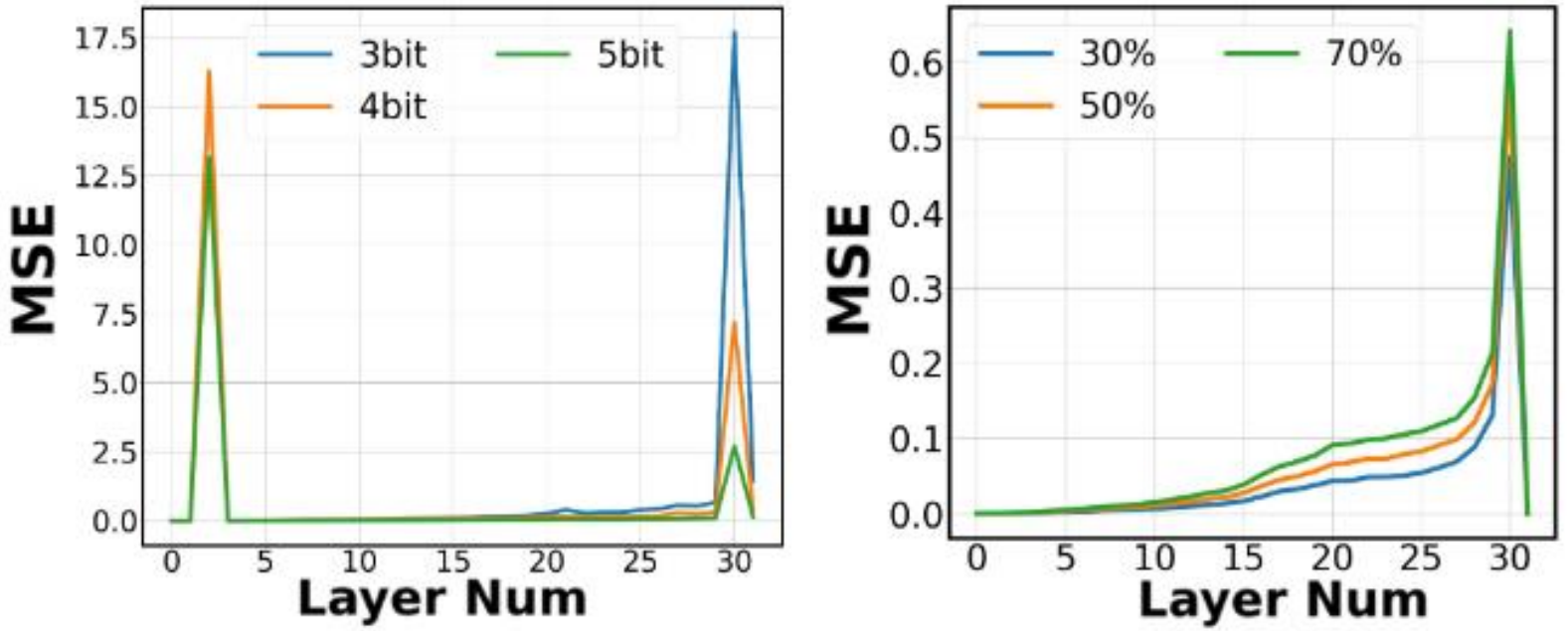
- LLM中只有一小部分层对压缩具有较高的敏感性
- LLM的层对于量化位宽和修剪稀疏度的敏感性位于不同维度

Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

$$b_j = B + \mathbb{1}(s^j_{quant} \geq \frac{\sum_{i=0}^{L-1} s^i_{quant}}{L}), \qquad (1)$$

b_j 对第 j 层采用的量化位宽　　B 目标量化位宽
S_{quant} 第 j 层预统计的b_j下的MSE　　L 模型总层数
$\mathbb{1}(.)$ 指示函数，True为1，False为0

3.1 分层联合压缩（Layer-wise Unified Compression）



- LLM中只有一小部分层对压缩具有较高的敏感性
- LLM的层对于量化位宽和修剪稀疏度的敏感性位于不同维度

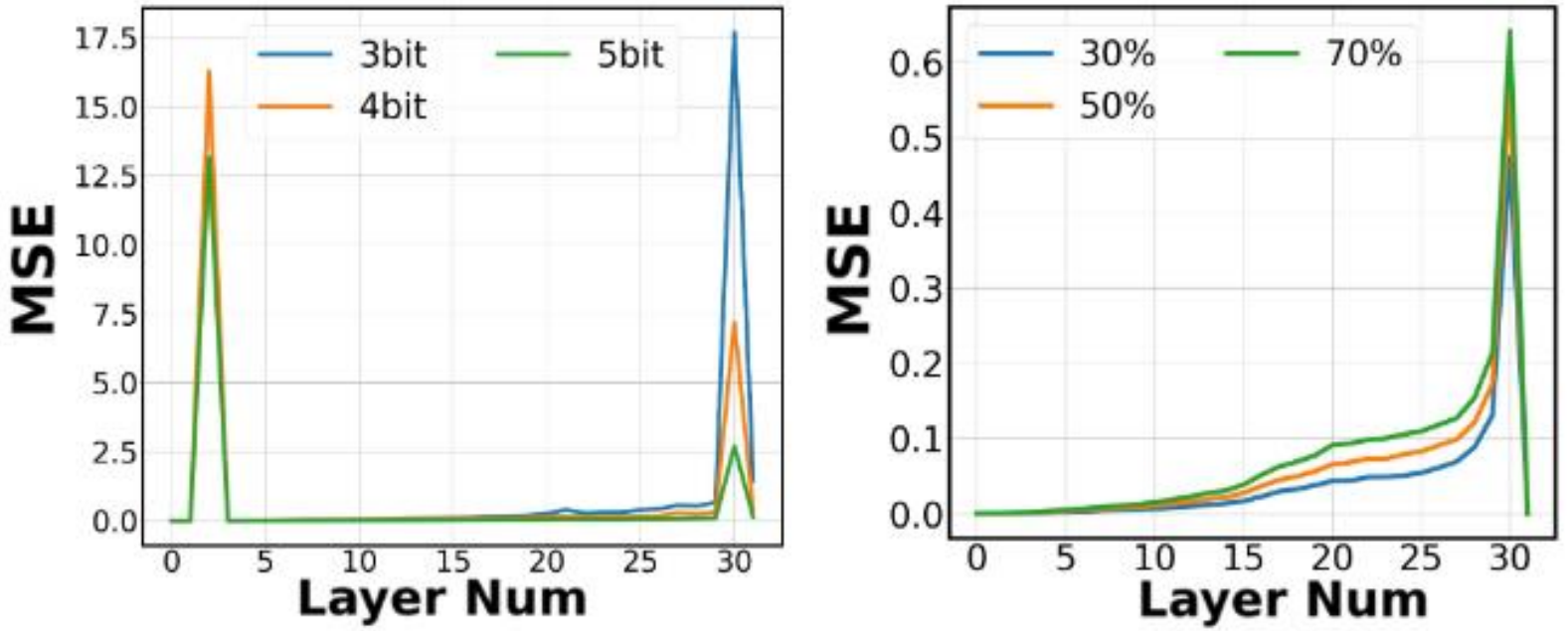
Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

$$b_j = B + \mathbb{1}(s^j_{quant} \geq \frac{\sum_{i=0}^{L-1} s^i_{quant}}{L}), \quad (1)$$

$$p_j = P \times L \times \frac{s^j_{prune}}{\sum_{i=1}^{L-1} s^i_{prune}}, \quad (2)$$

b_j 对第 j 层采用的量化位宽　　B 目标量化位宽
S_{quant} 第 j 层预统计的 b_j 下的MSE　　L 模型总层数
$\mathbb{1}(.)$ 指示函数，True为1，False为0

3.1 分层联合压缩（Layer-wise Unified Compression）



- LLM中只有一小部分层对压缩具有较高的敏感性
- LLM的层对于量化位宽和修剪稀疏度的敏感性位于不同维度

Figure 3. Visualization of LLaMA-7B's layer-wise sensitivity to (a) quantization and (b) pruning.

$$b_j = B + \mathbb{1}(s^j_{quant} \geq \frac{\sum_{i=0}^{L-1} s^i_{quant}}{L}), \quad (1)$$

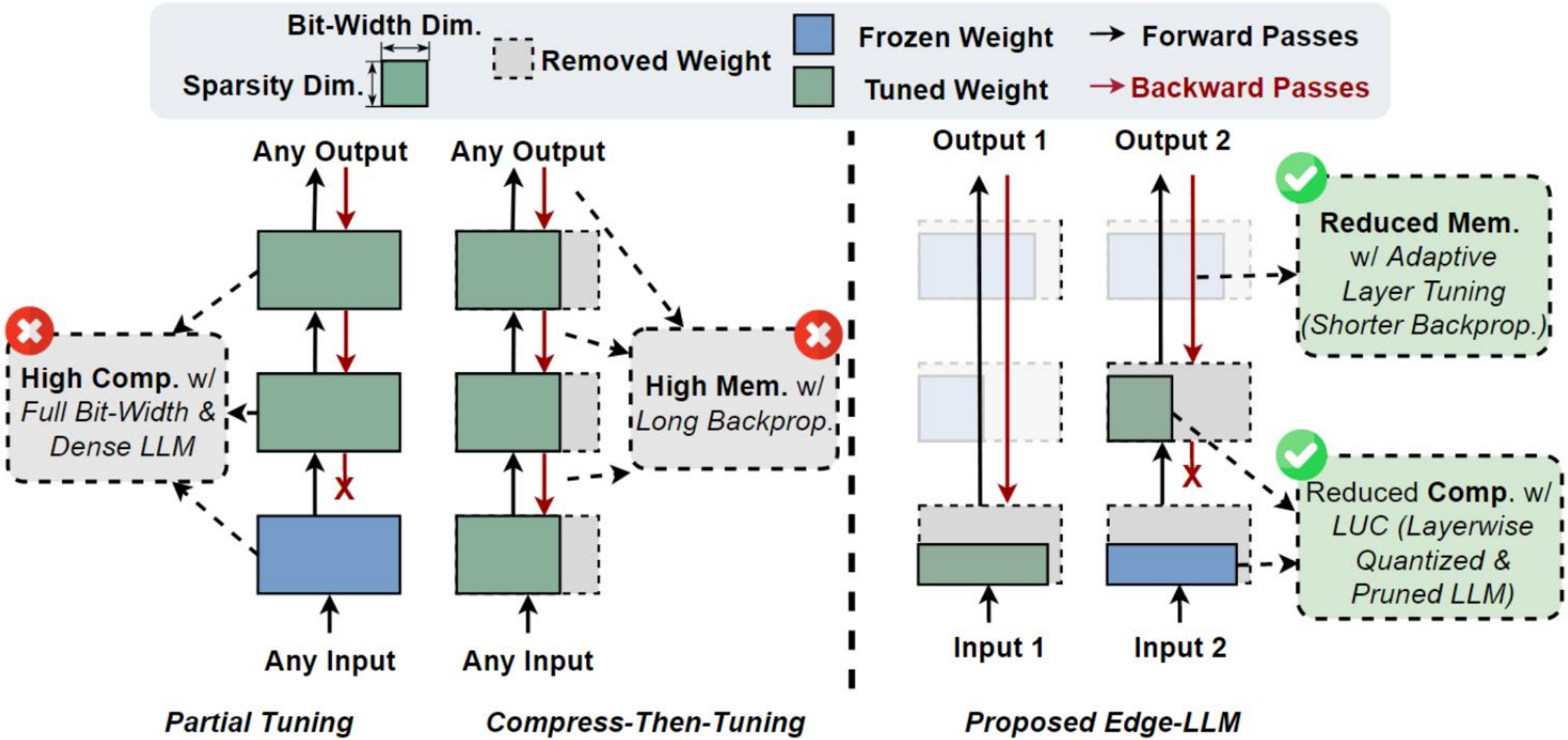
$$p_j = P \times L \times \frac{s^j_{prune}}{\sum_{i=1}^{L-1} s^i_{prune}}, \quad (2)$$

b_j 对第 j 层采用的量化位宽　　B 目标量化位宽
S_{quant} 第 j 层预统计的 b_j 下的MSE　　L 模型总层数
$\mathbb{1}(.)$ 指示函数，True为1，False为0

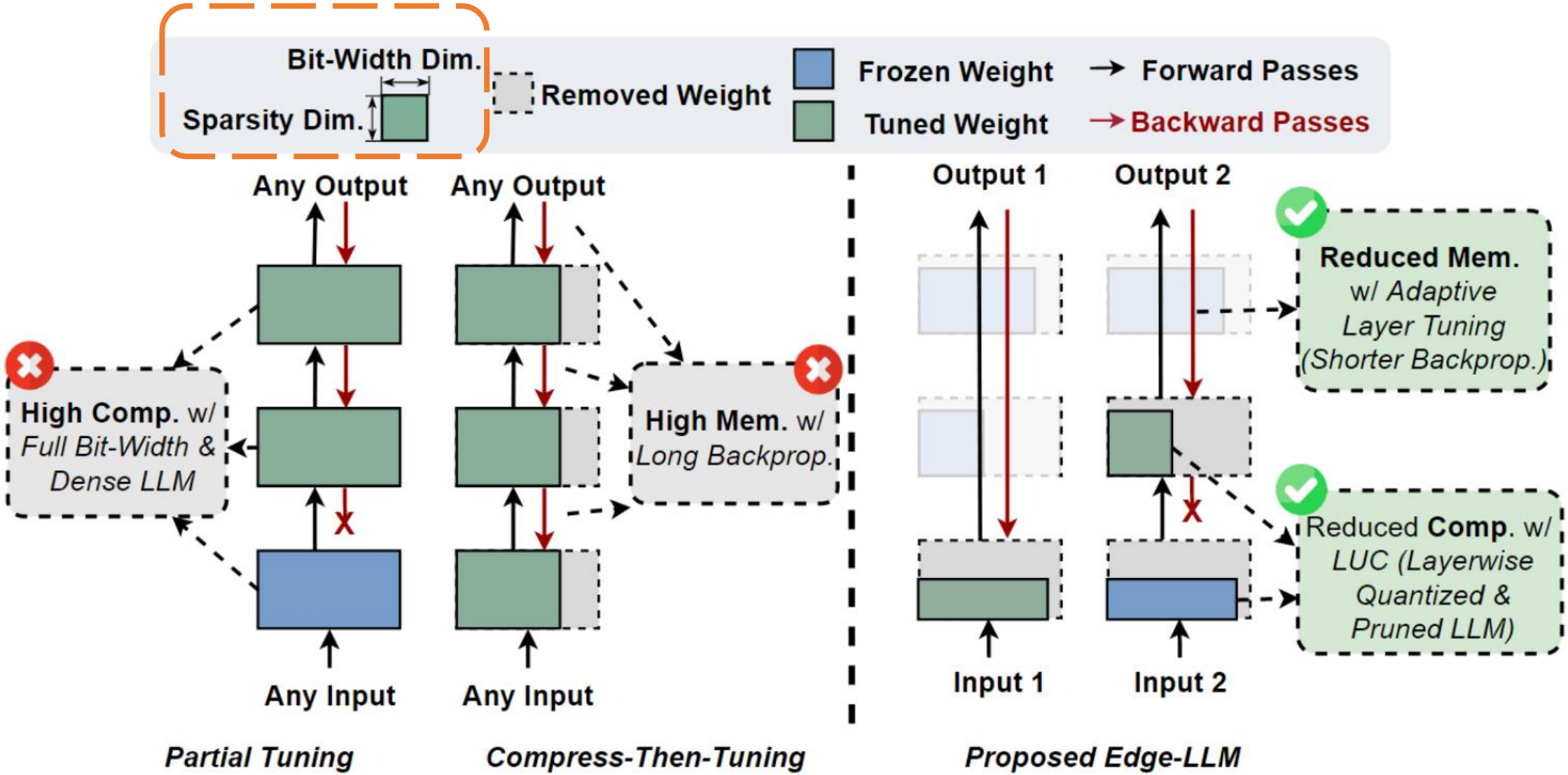
p_j 对第 j 层采用的修剪稀疏度　　P 目标修剪稀疏度
S_{prune} 第 j 层预统计的 p_j 下的MSE　　L 模型总层数

3.2 自适应层tuning（Adaptive Layer Tuning and Voting）

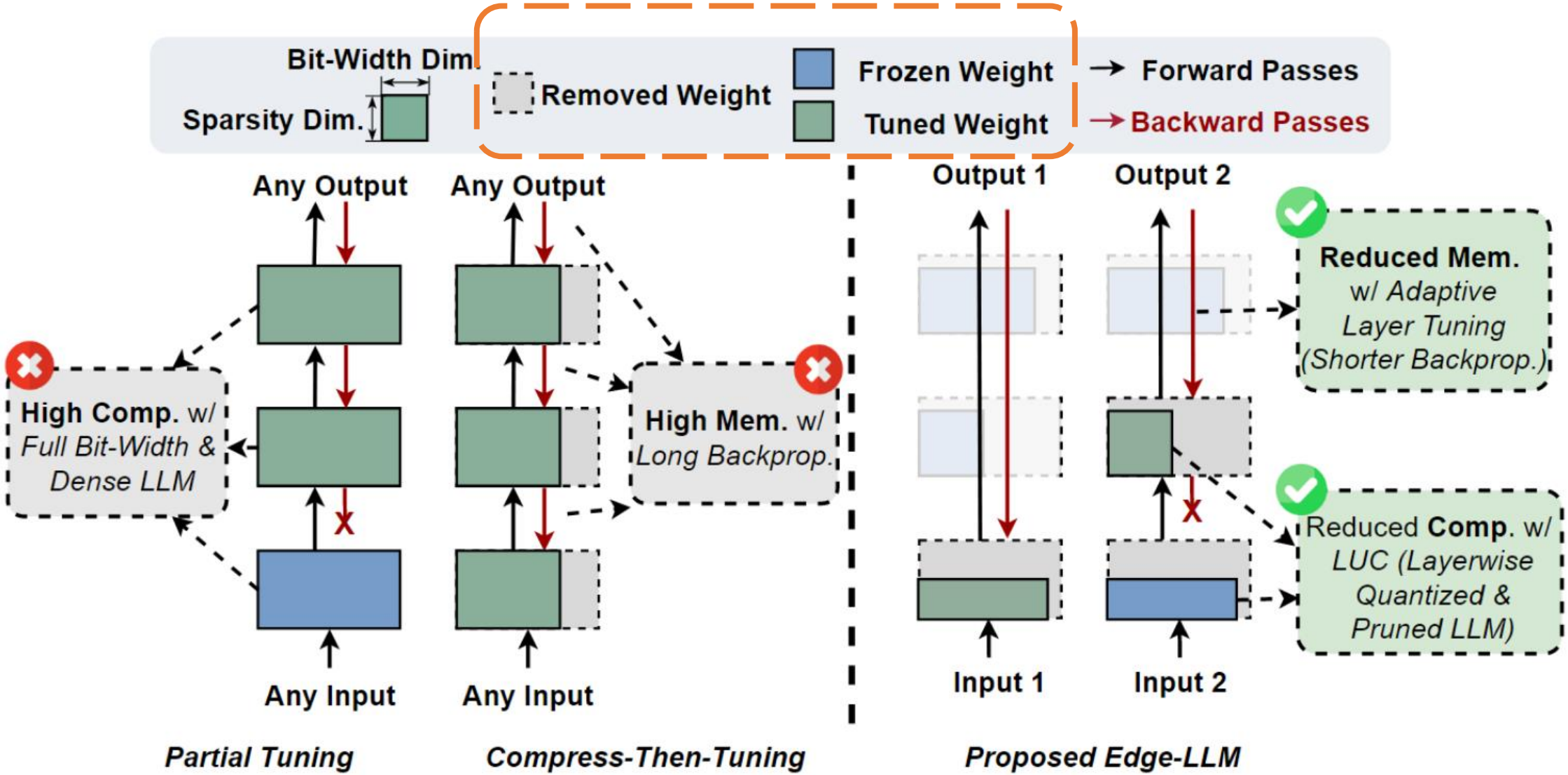
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



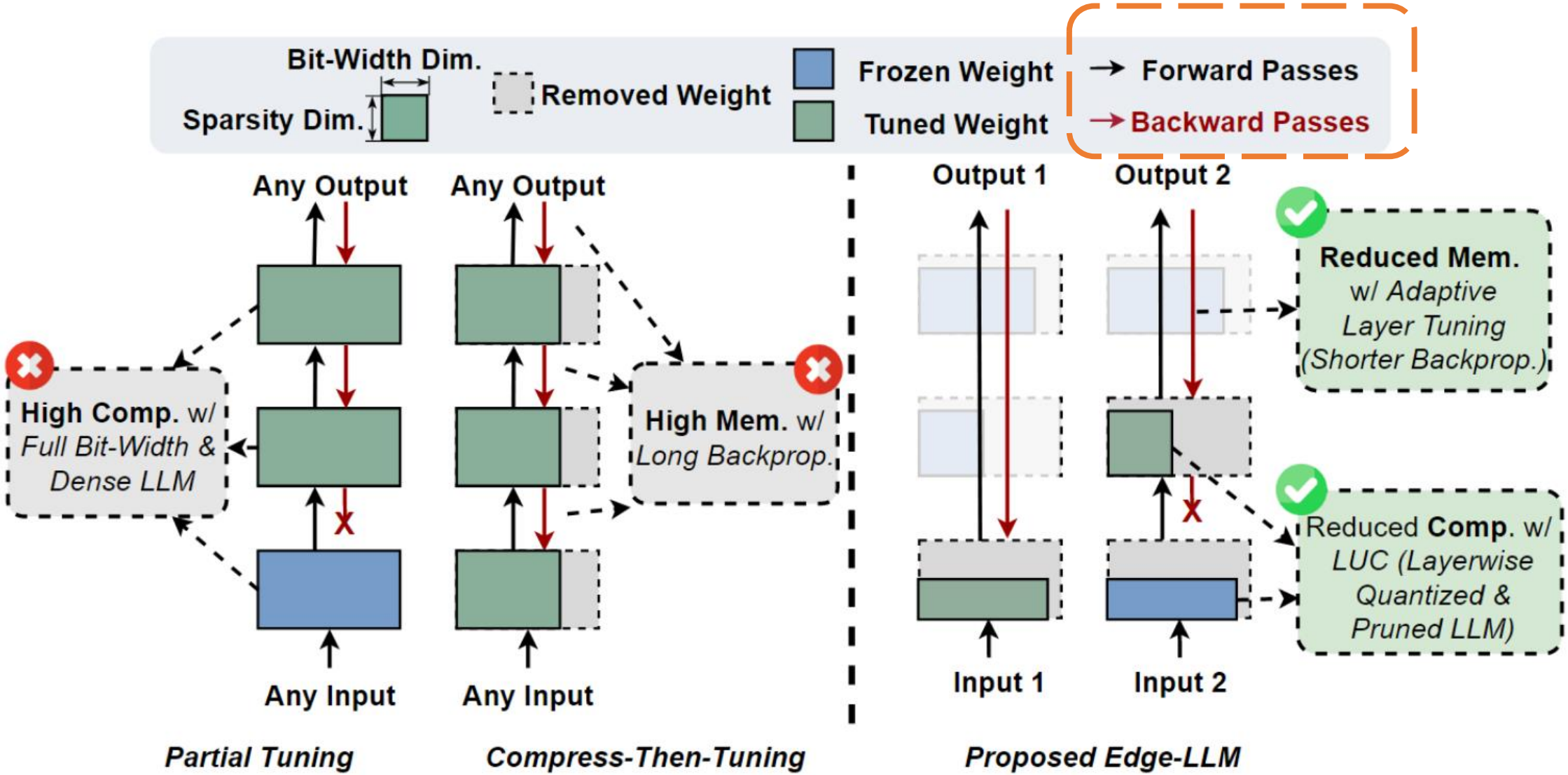
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



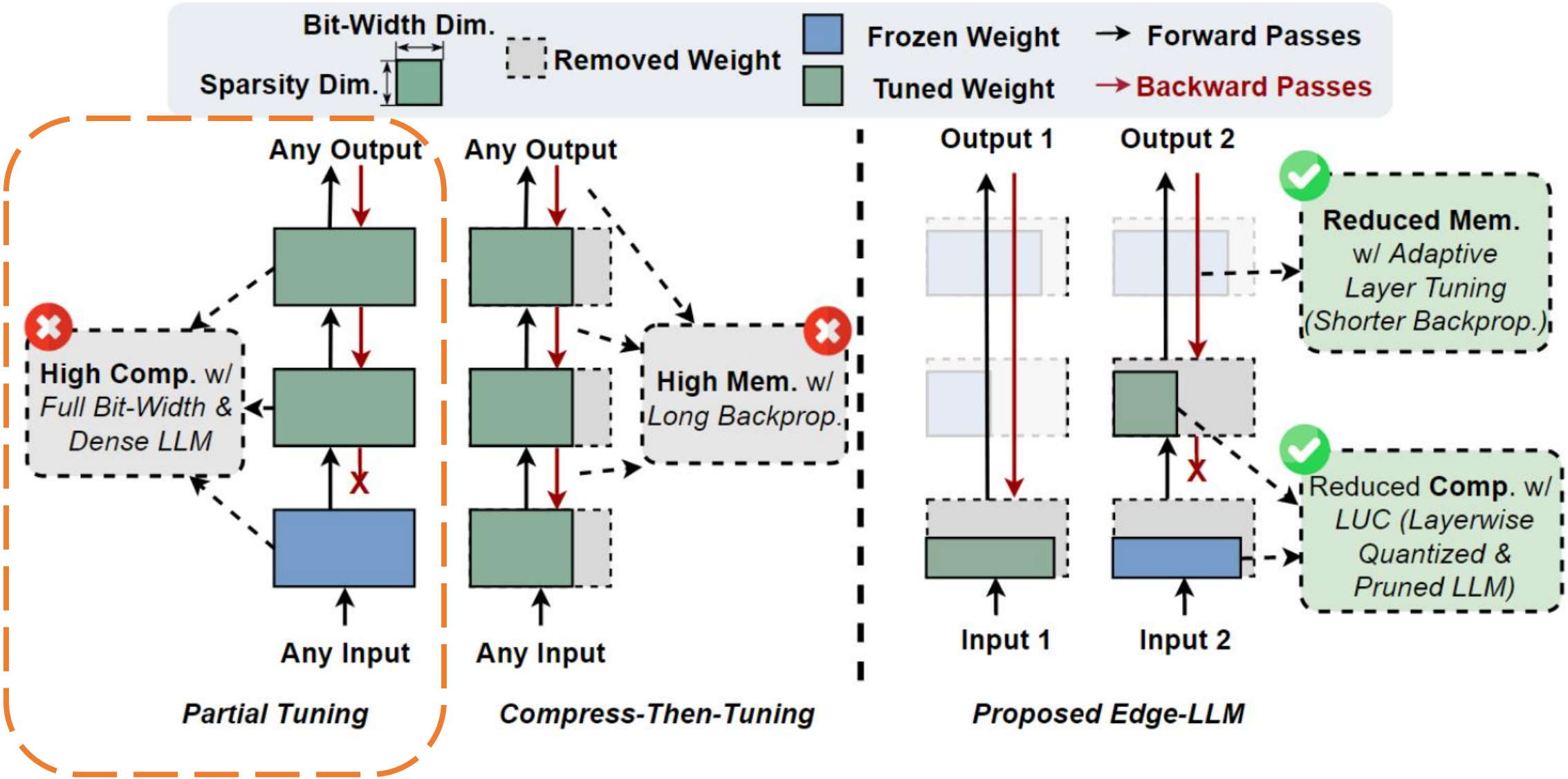
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



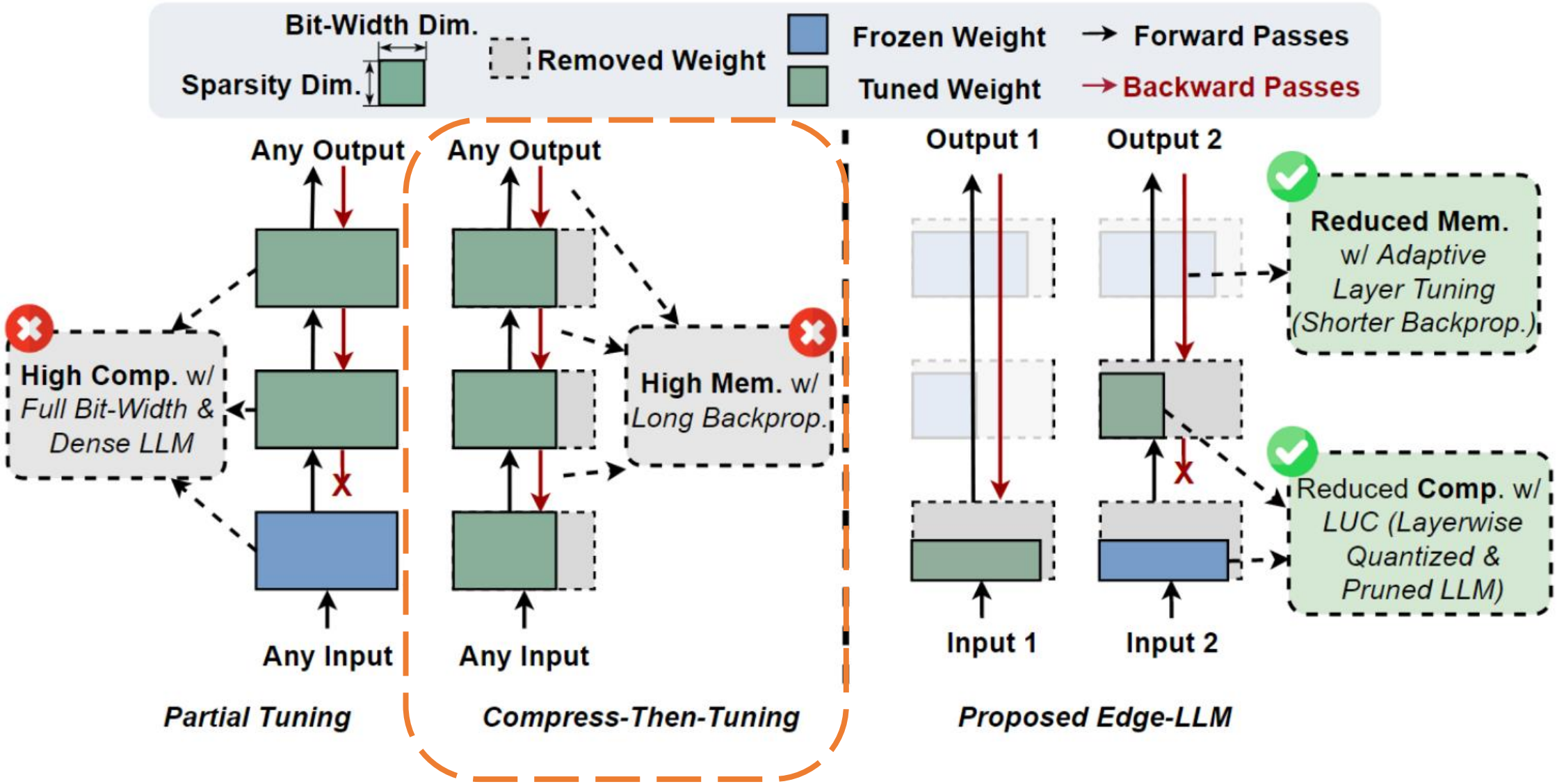
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



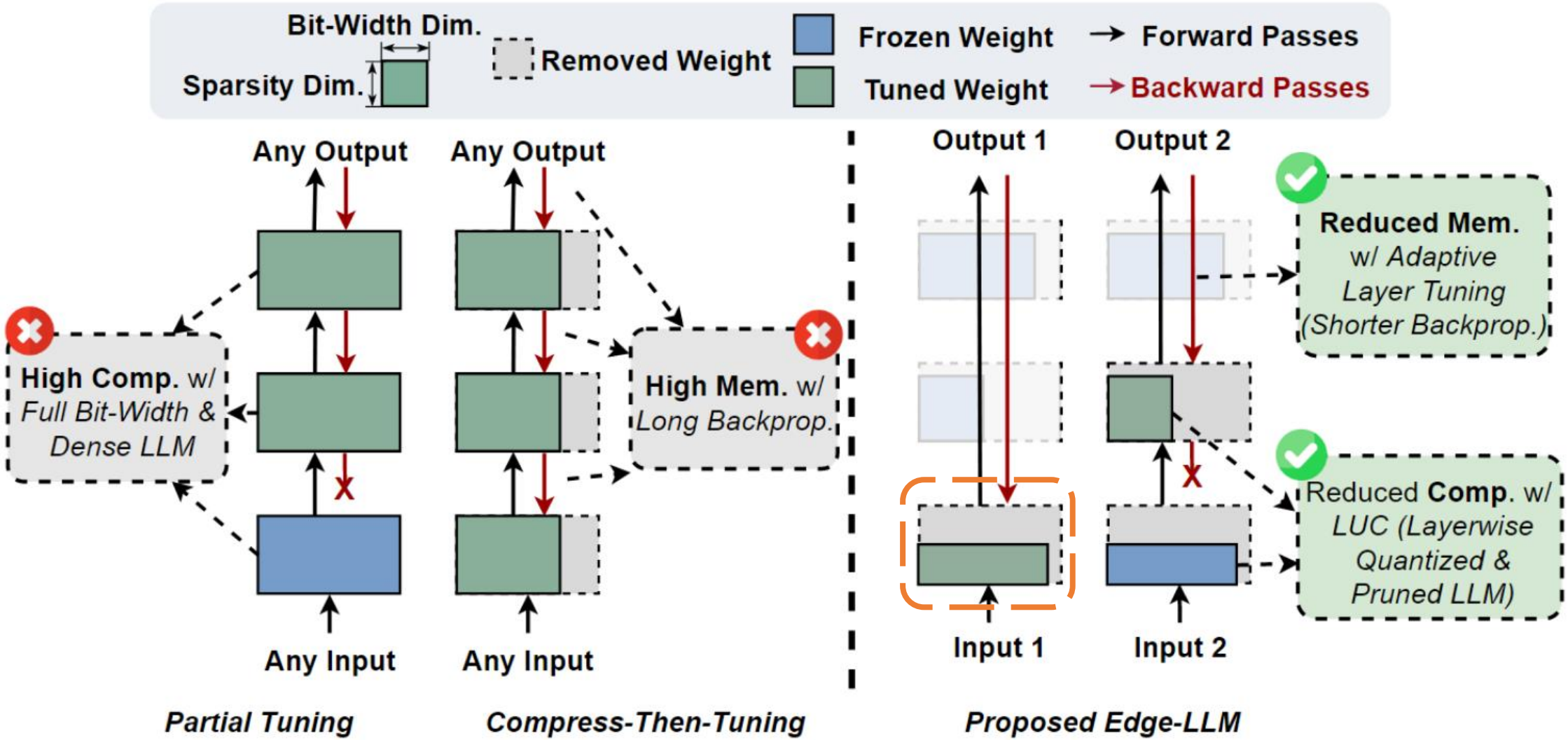
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



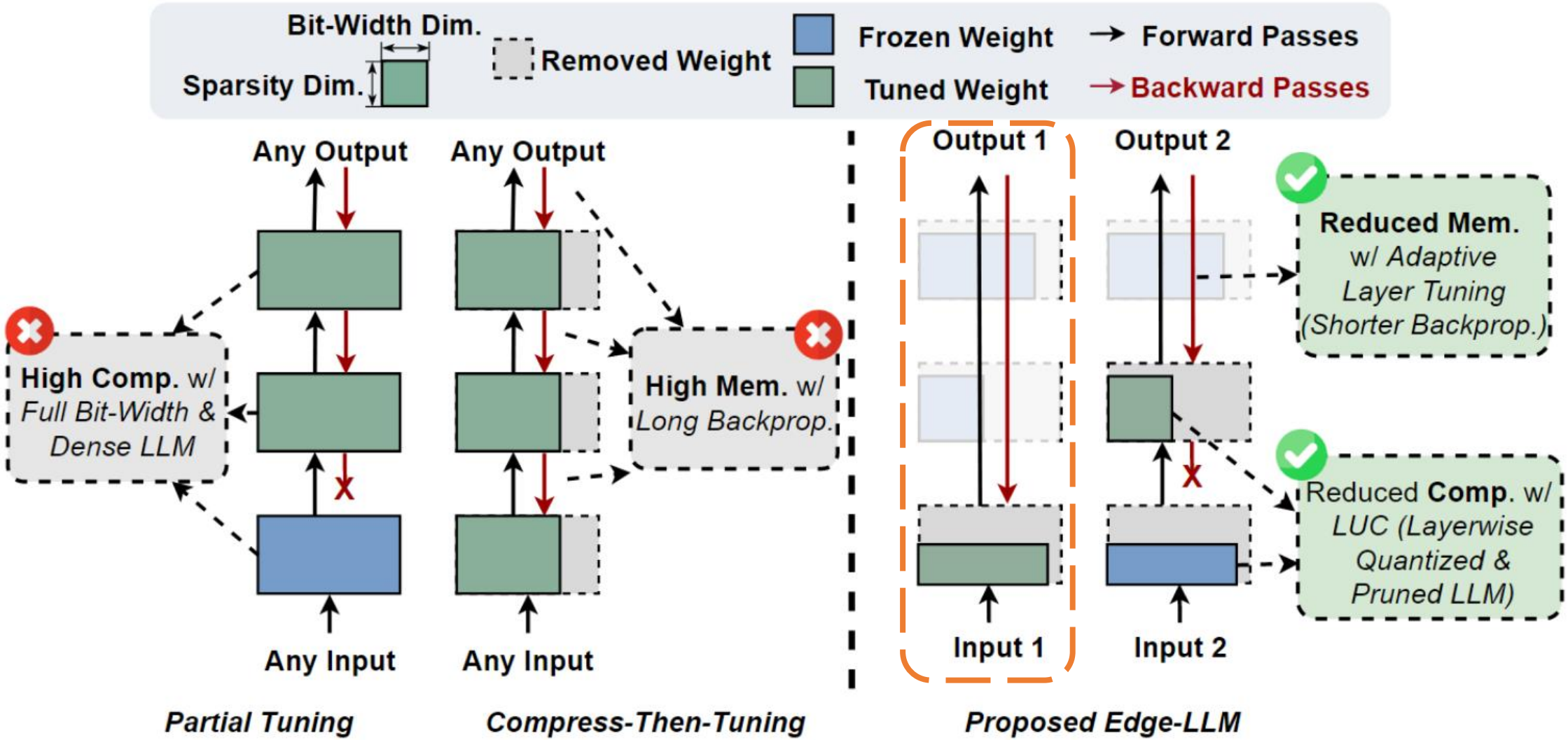
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



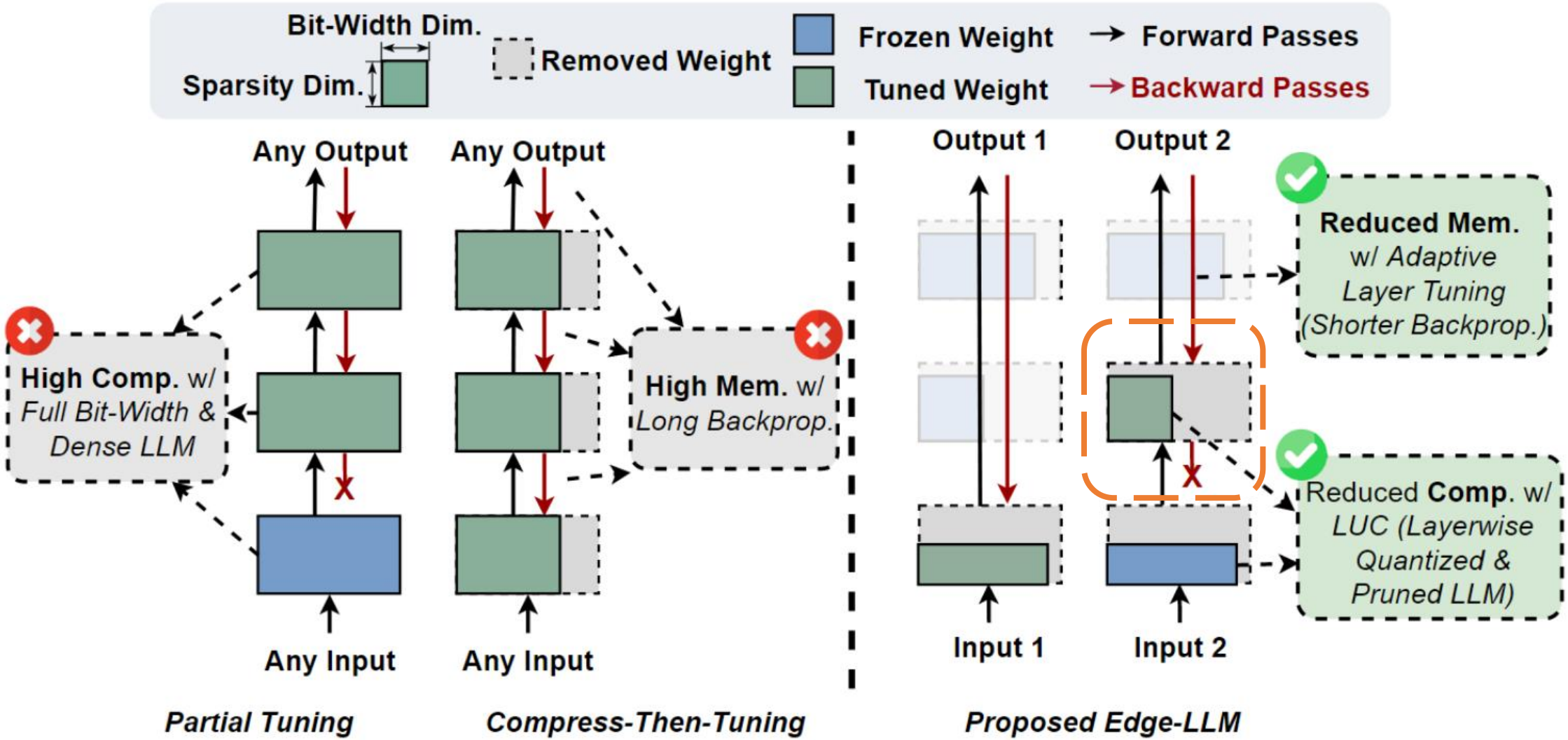
3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



3.2 自适应层tuning（Adaptive Layer Tuning and Voting）

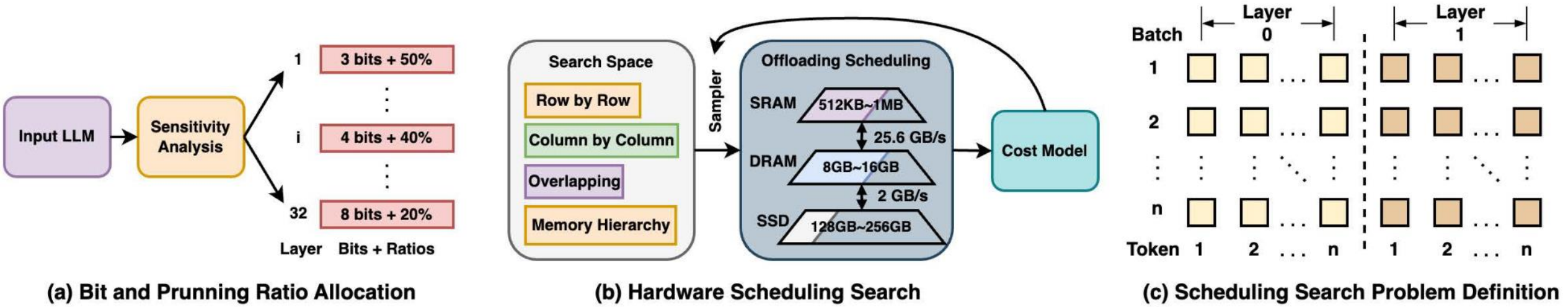


3.2 自适应层tuning（Adaptive Layer Tuning and Voting）



3.3 互补的硬件调度模块

3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

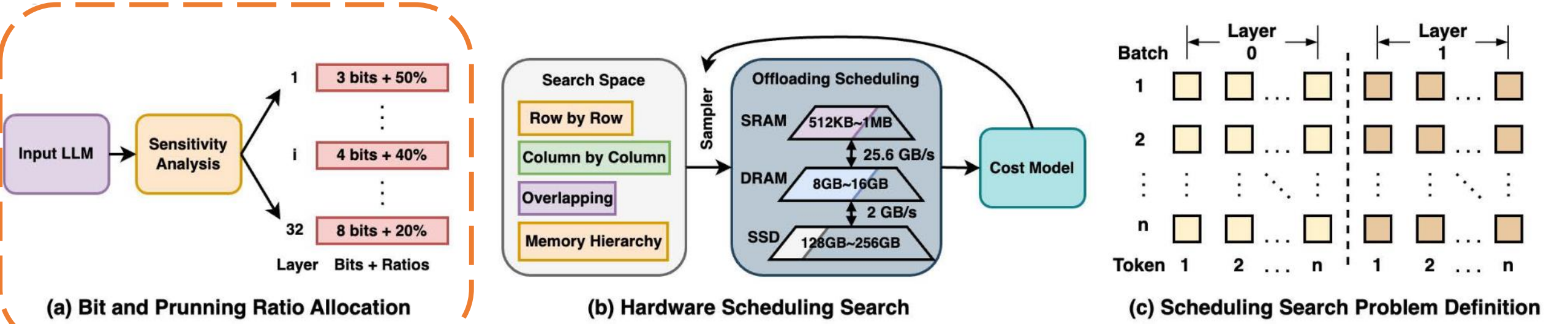


Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

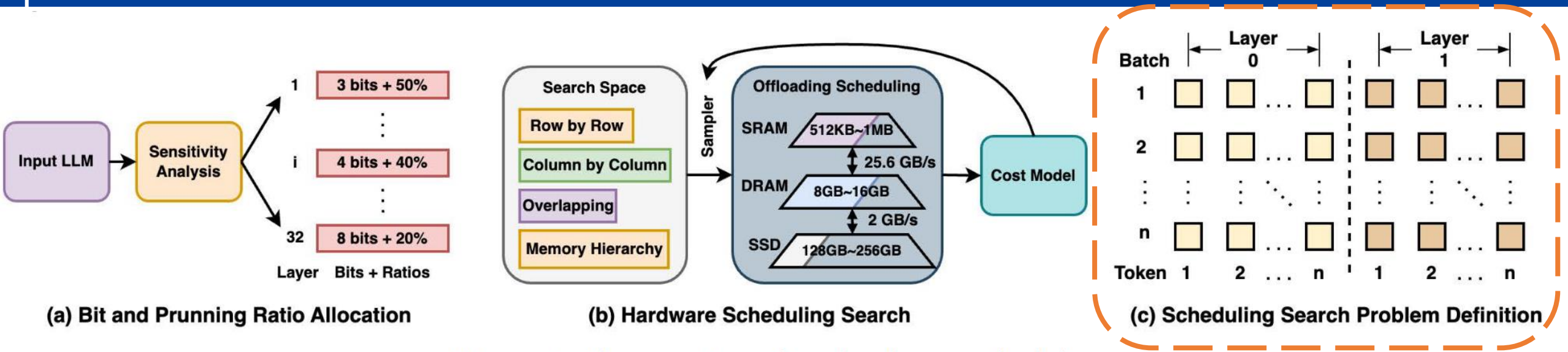
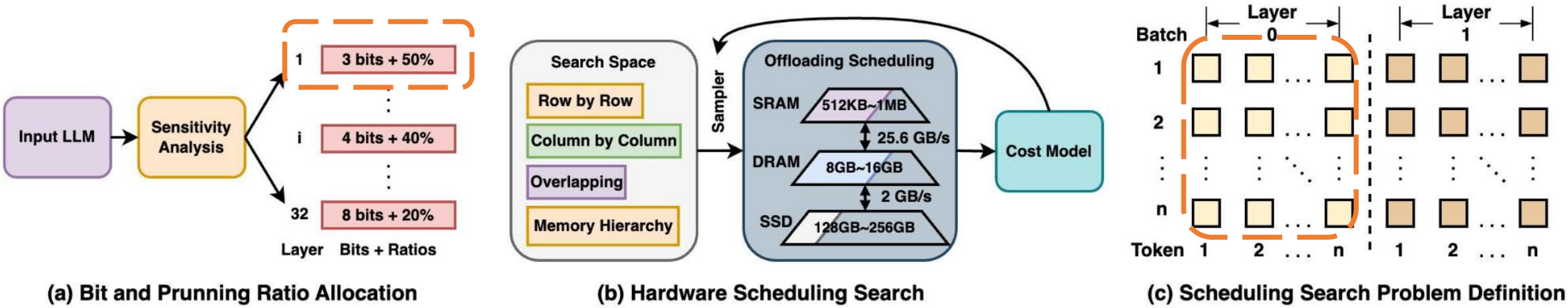


Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

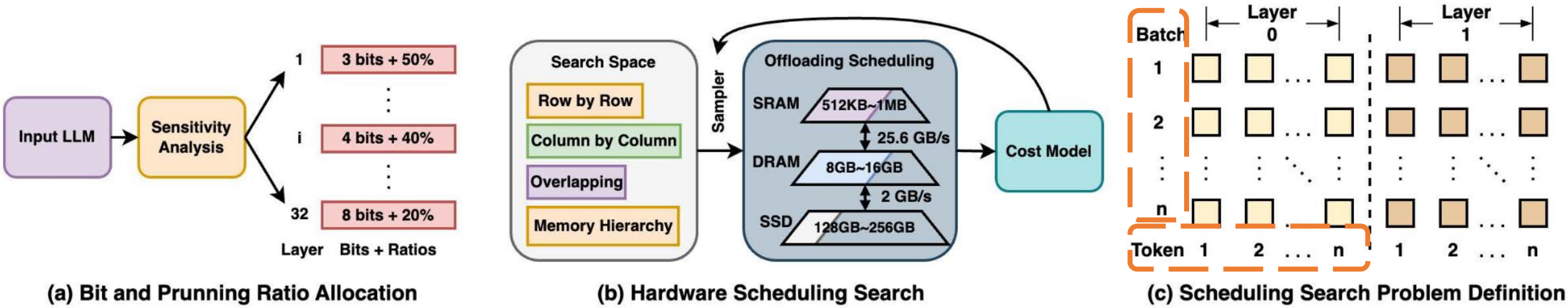


Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

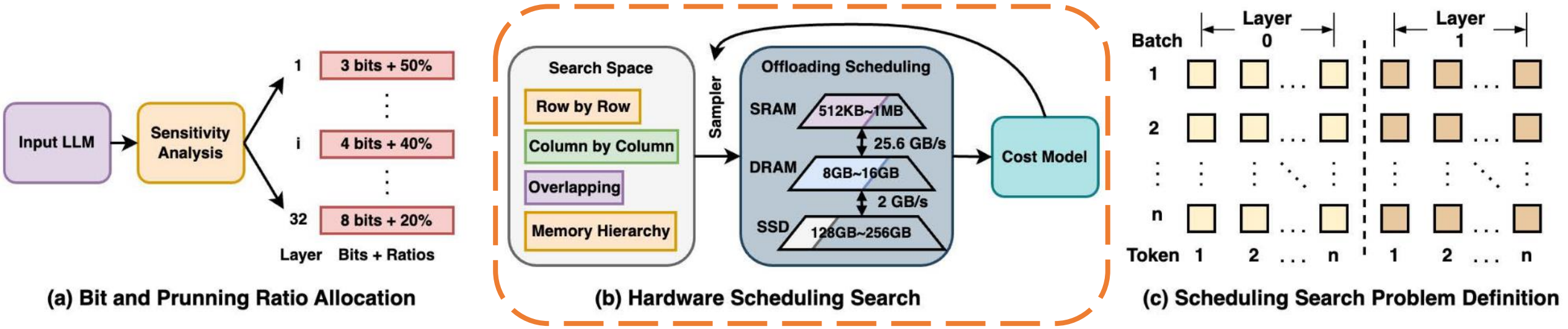


Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

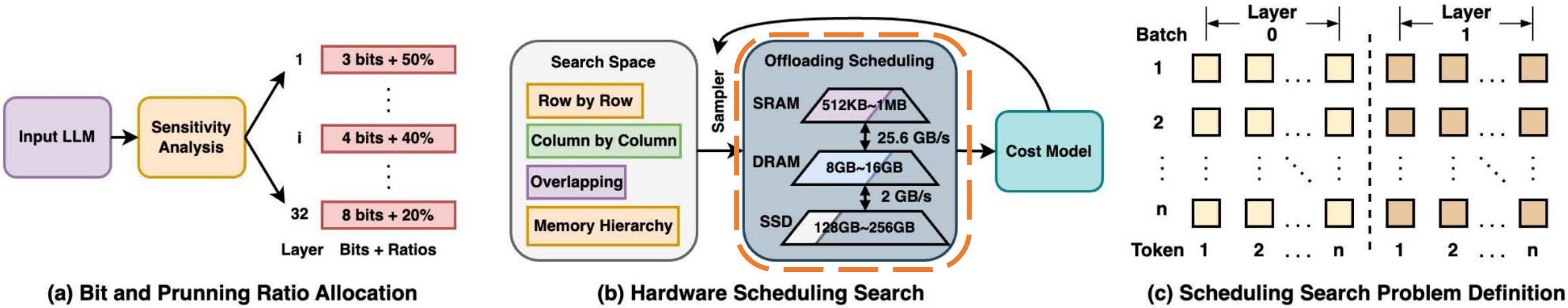
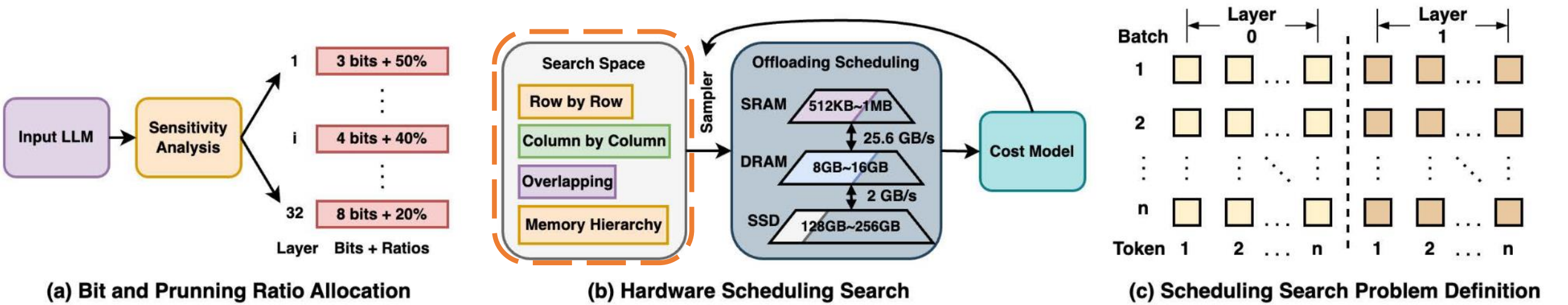


Figure 4. The overview of our hardware scheduling.

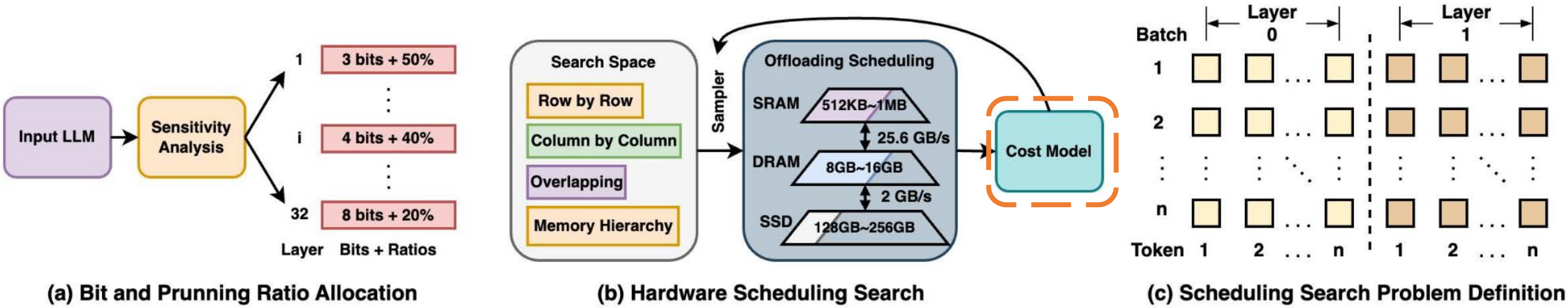
3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块

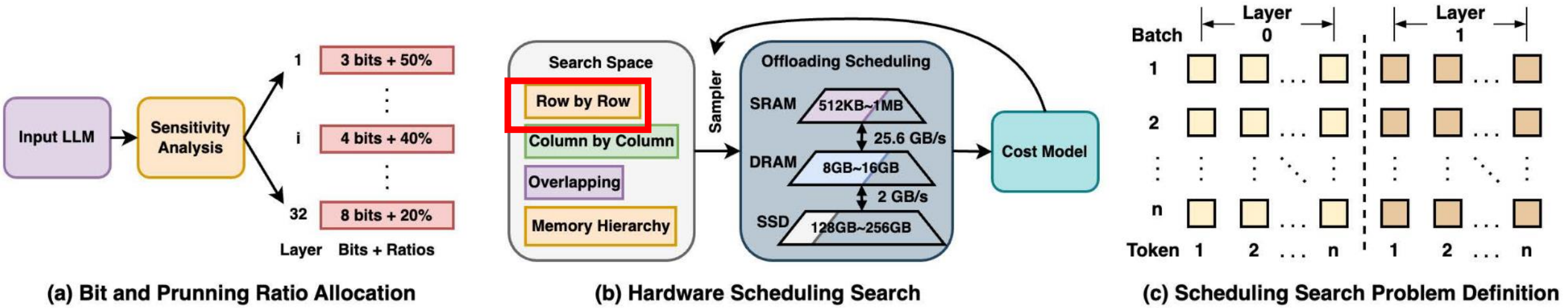
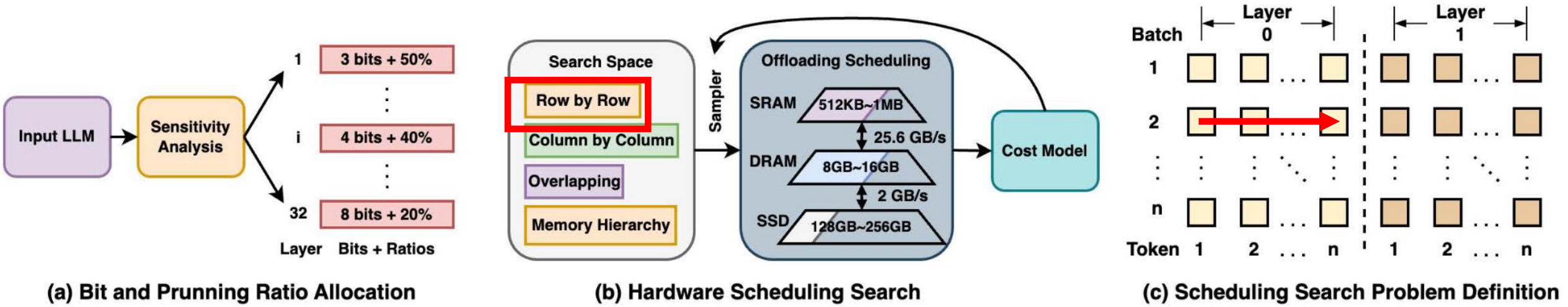


Figure 4. The overview of our hardware scheduling.

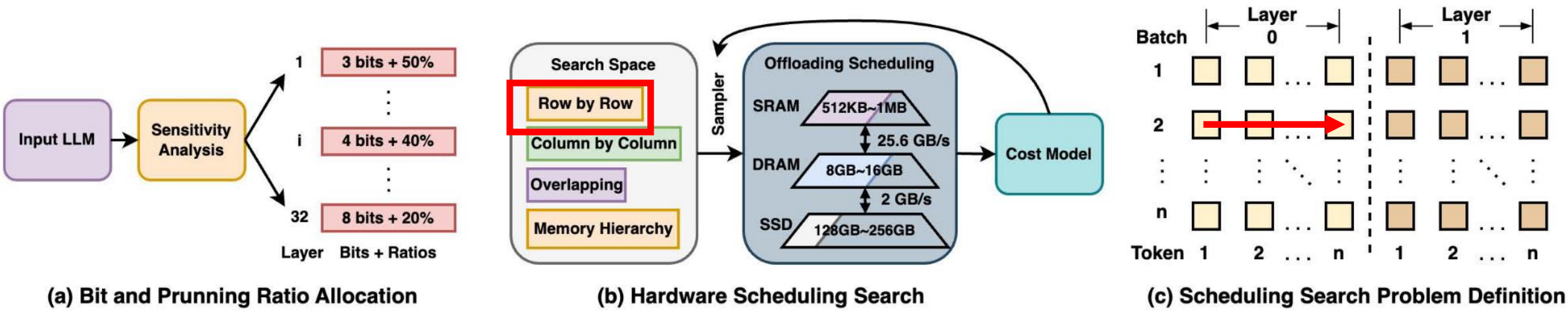
3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

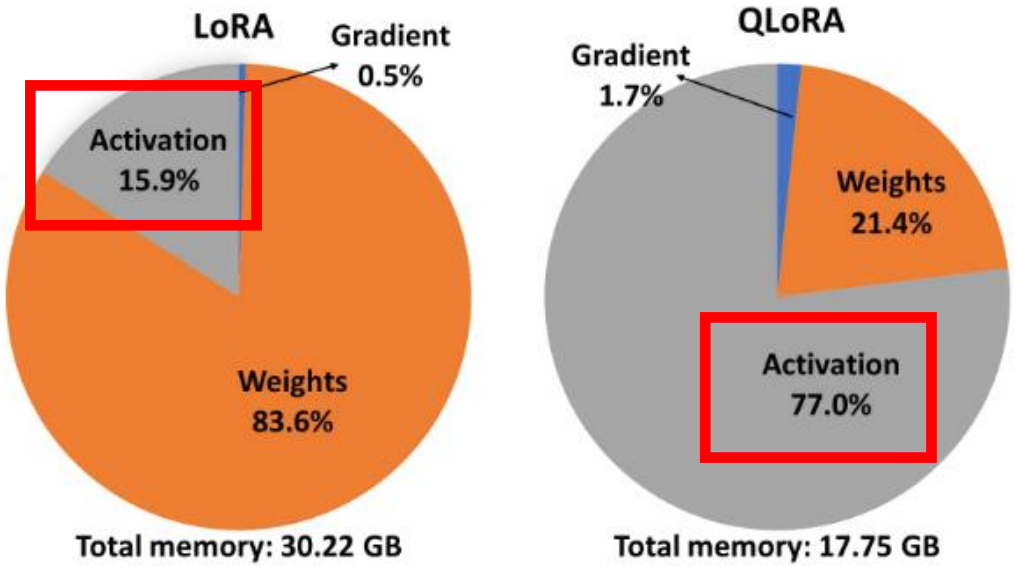
Figure 4. The overview of our hardware scheduling.

3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.



3.3 互补的硬件调度模块

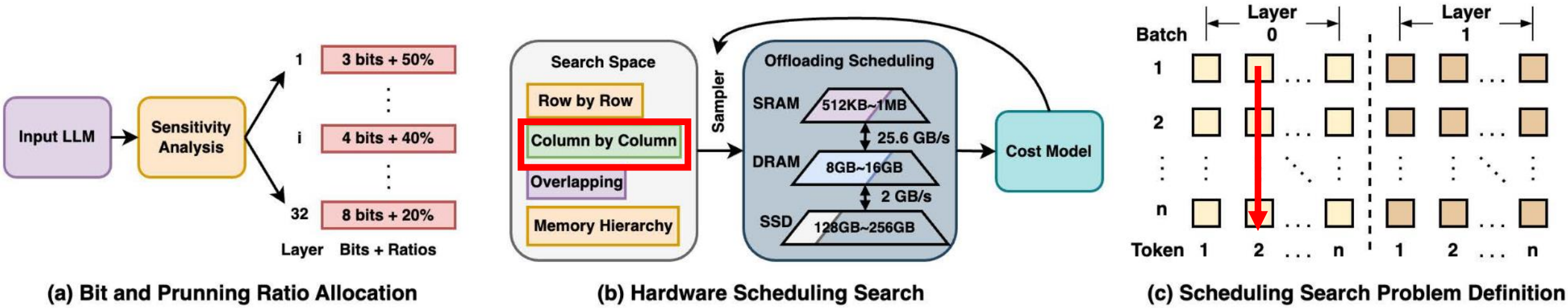
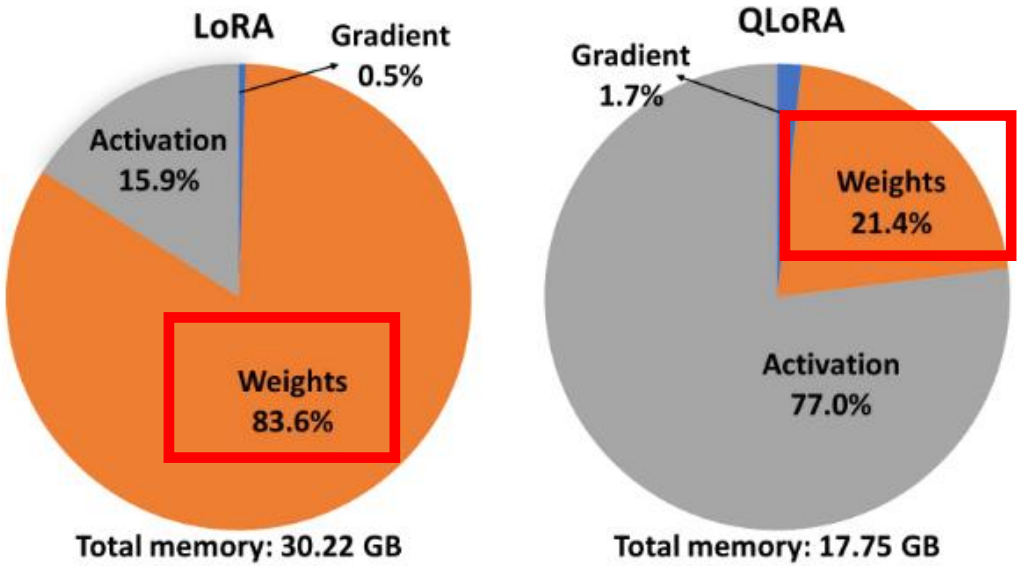
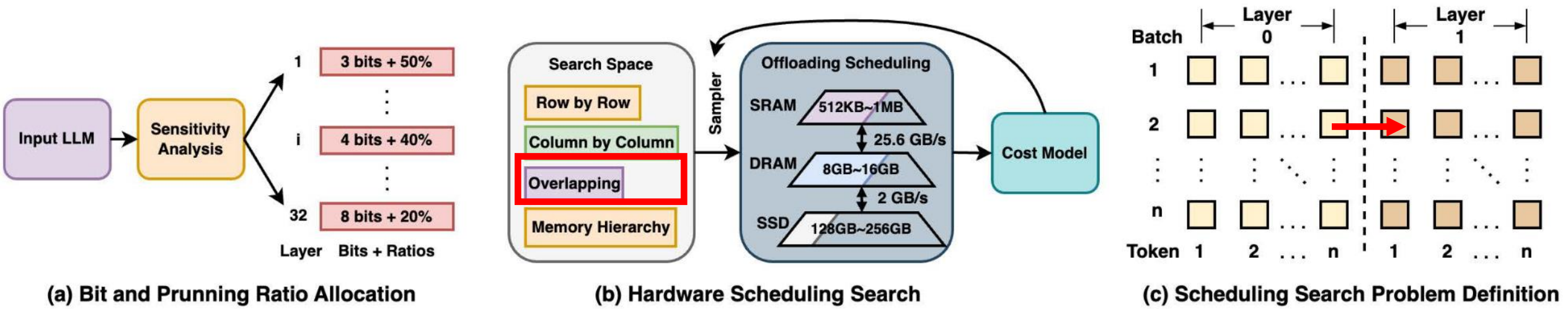


Figure 4. The overview of our hardware scheduling.

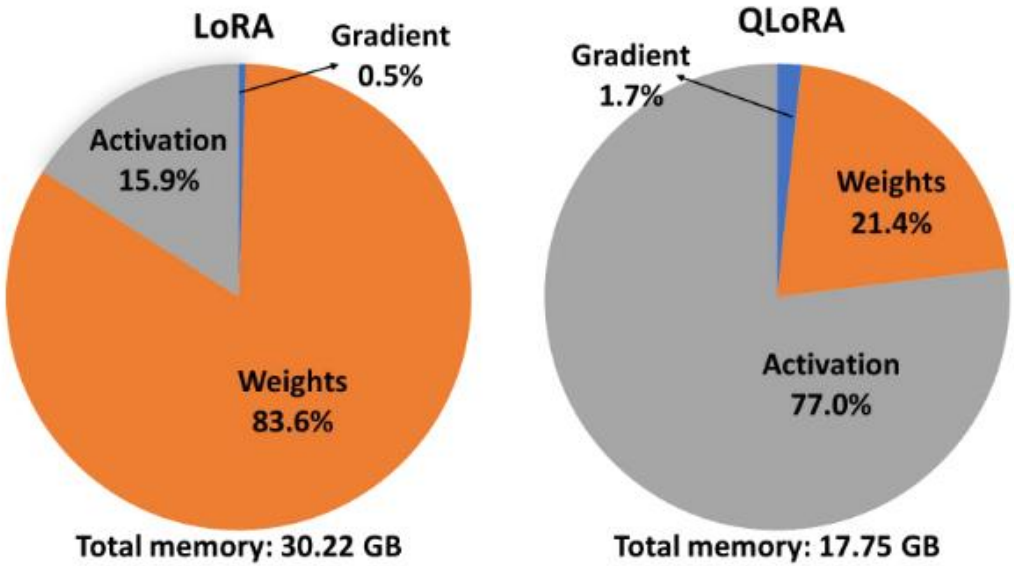


3.3 互补的硬件调度模块

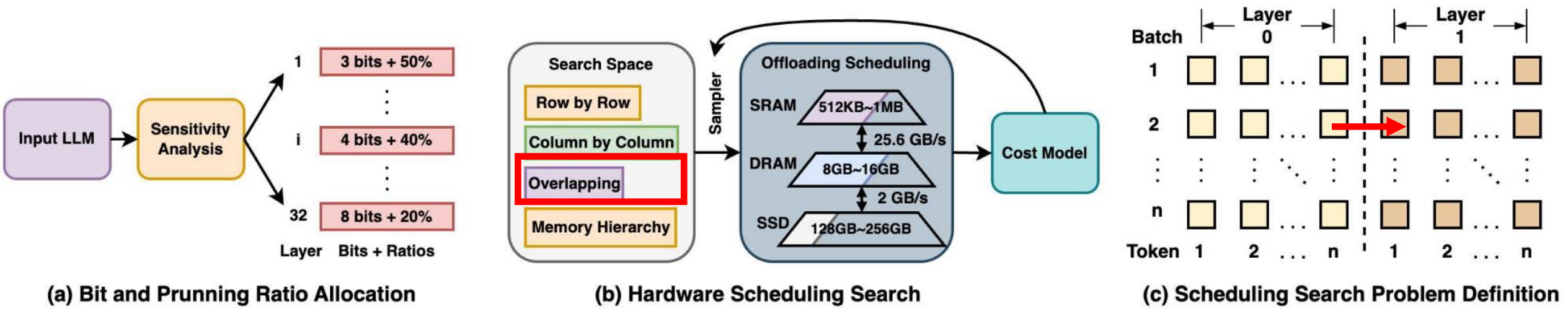


(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.

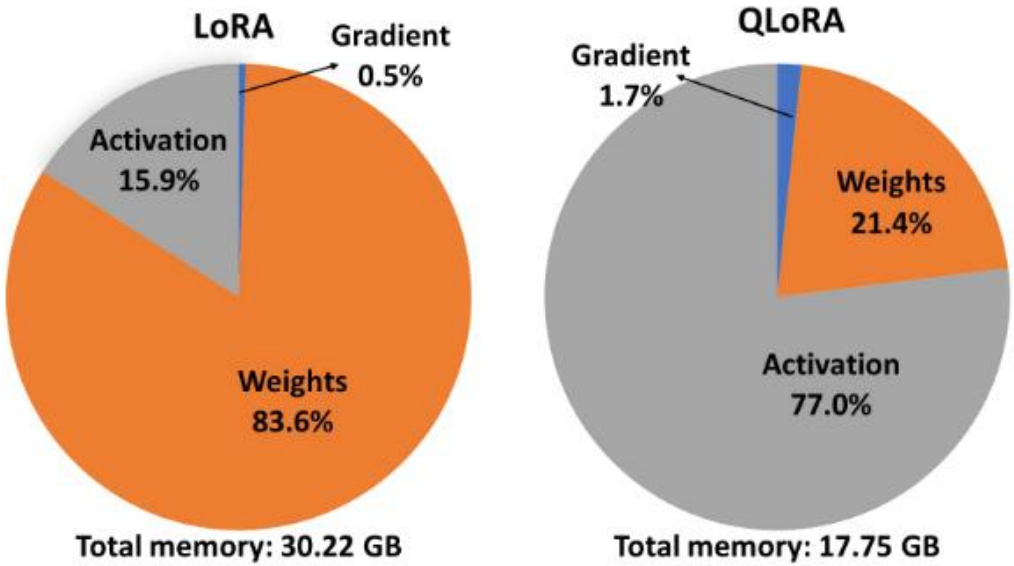


3.3 互补的硬件调度模块



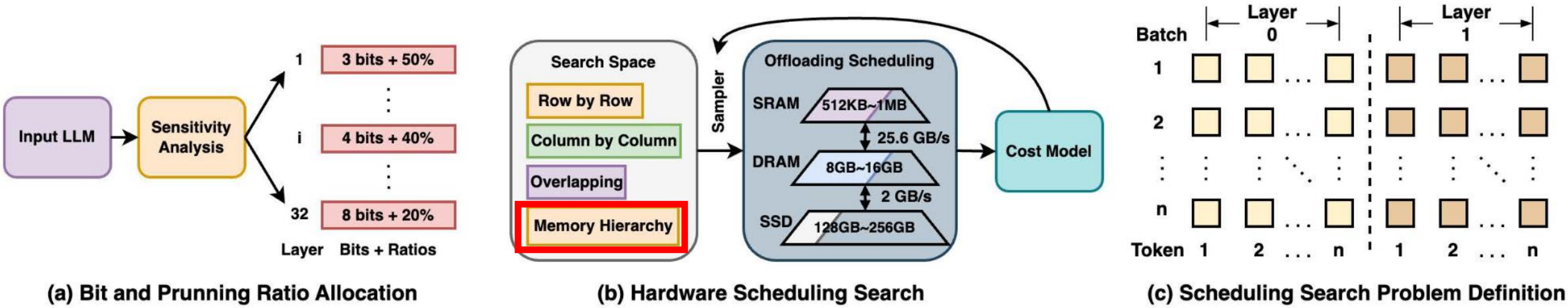
(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.



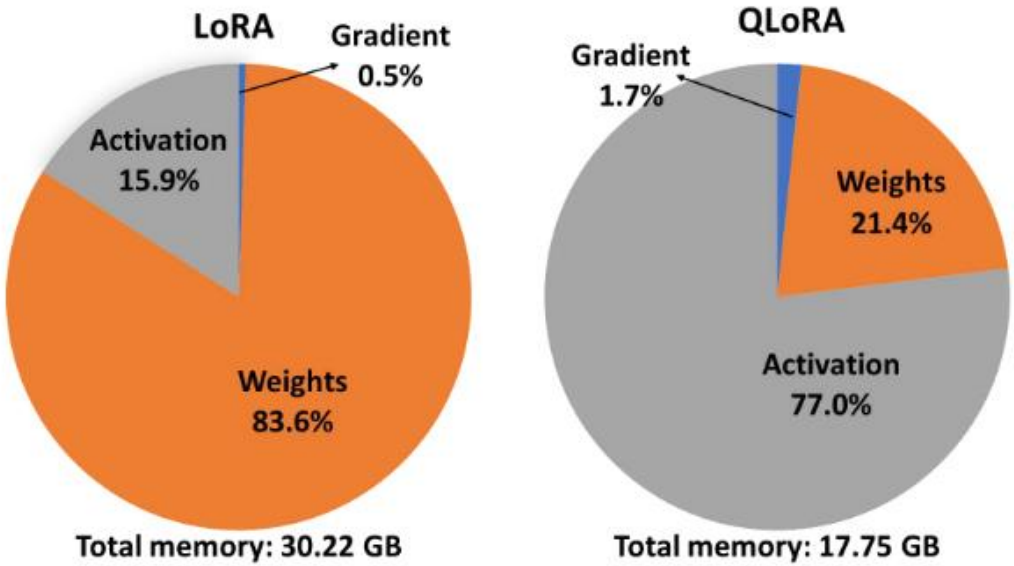
- 下一层的权重载入
- 后续批次的激活信息载入
- 前一批次的激活信息存储
- 当前批次的计算

3.3 互补的硬件调度模块



(a) Bit and Prunning Ratio Allocation (b) Hardware Scheduling Search (c) Scheduling Search Problem Definition

Figure 4. The overview of our hardware scheduling.



3.3 互补的硬件调度模块

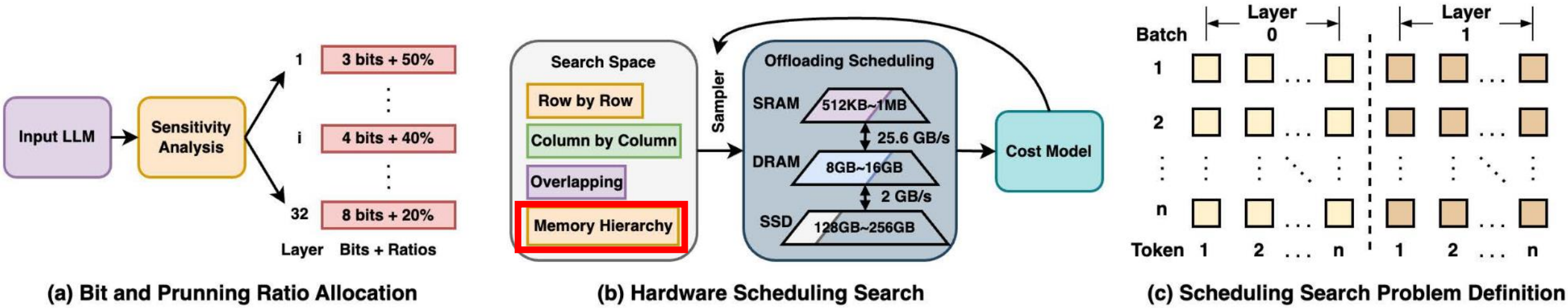
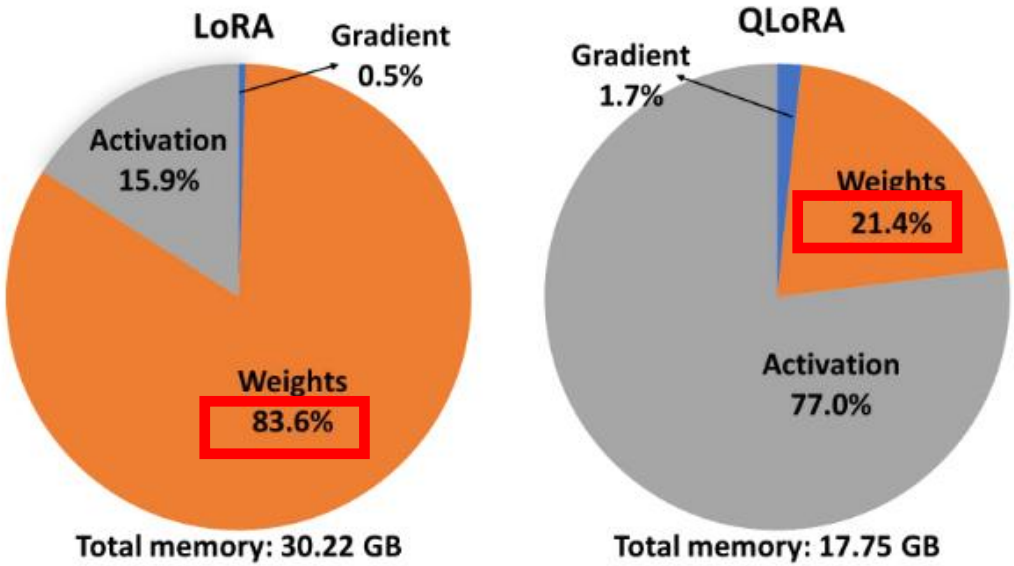


Figure 4. The overview of our hardware scheduling.



w_{sram} w_{dram} w_{ssd} SRAM/DRAM/SSD中权重所占比例

3.3 互补的硬件调度模块

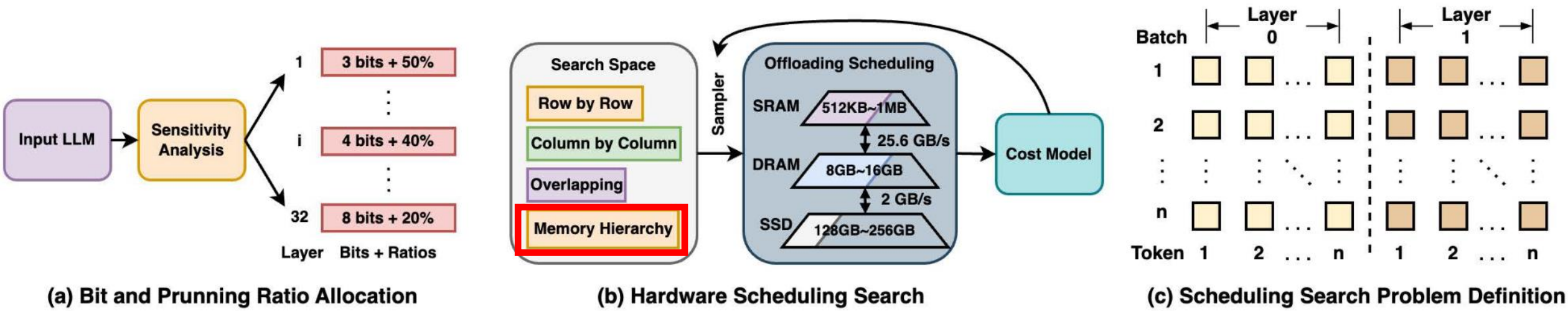
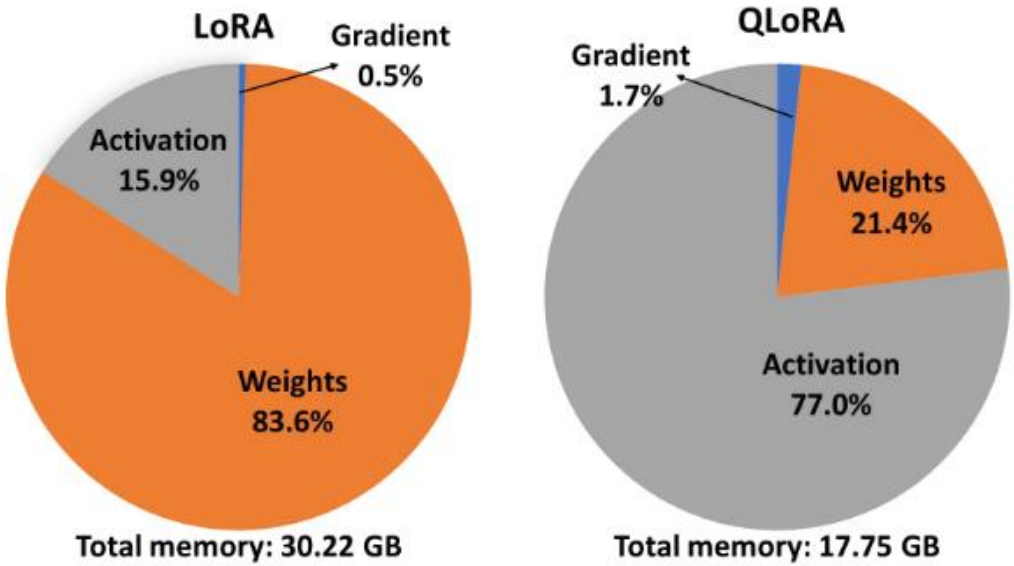


Figure 4. The overview of our hardware scheduling.



w_{sram} w_{dram} w_{ssd} SRAM/DRAM/SSD中权重所占比例

a_{sram} a_{dram} a_{ssd} SRAM/DRAM/SSD中激活信息所占比例

g_{sram} g_{dram} g_{ssd} SRAM/DRAM/SSD中梯度信息所占比例

3.3 互补的硬件调度模块

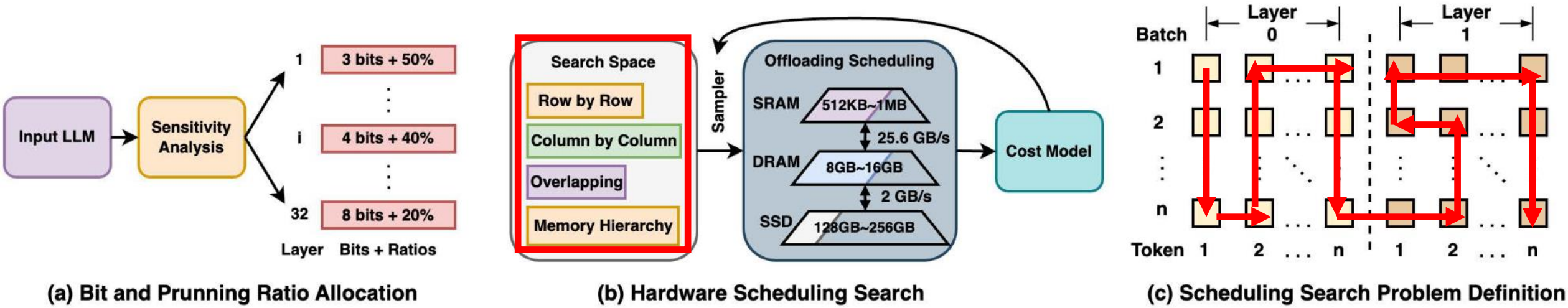
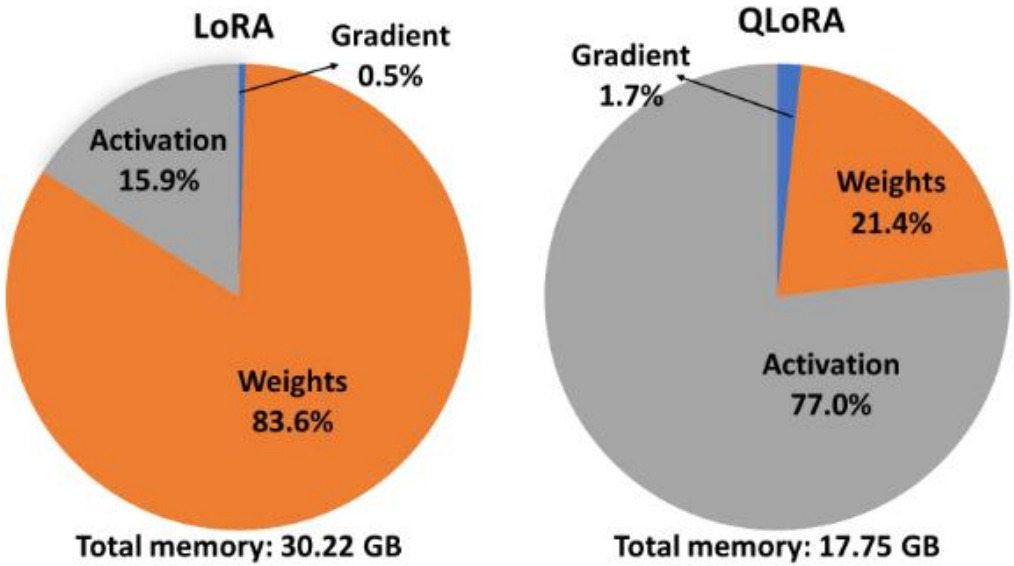


Figure 4. The overview of our hardware scheduling.



3.3 互补的硬件调度模块

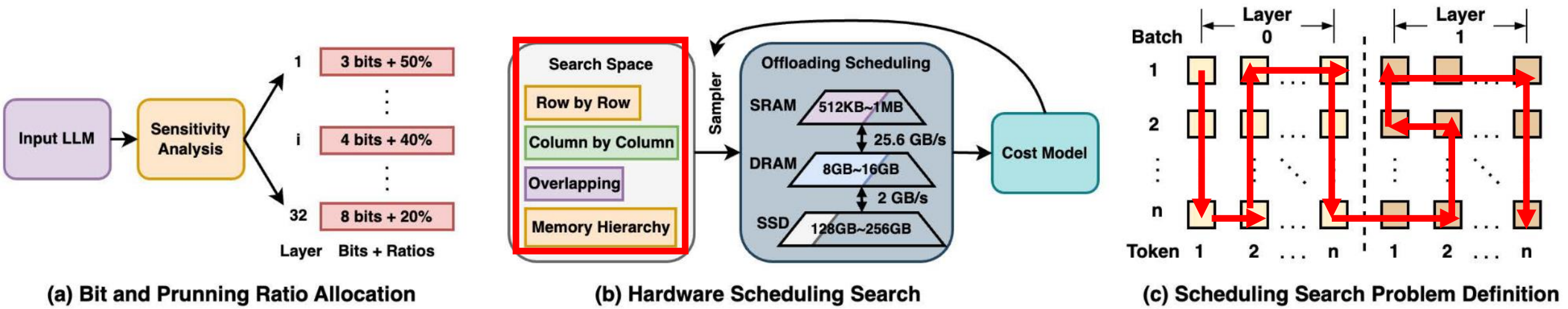
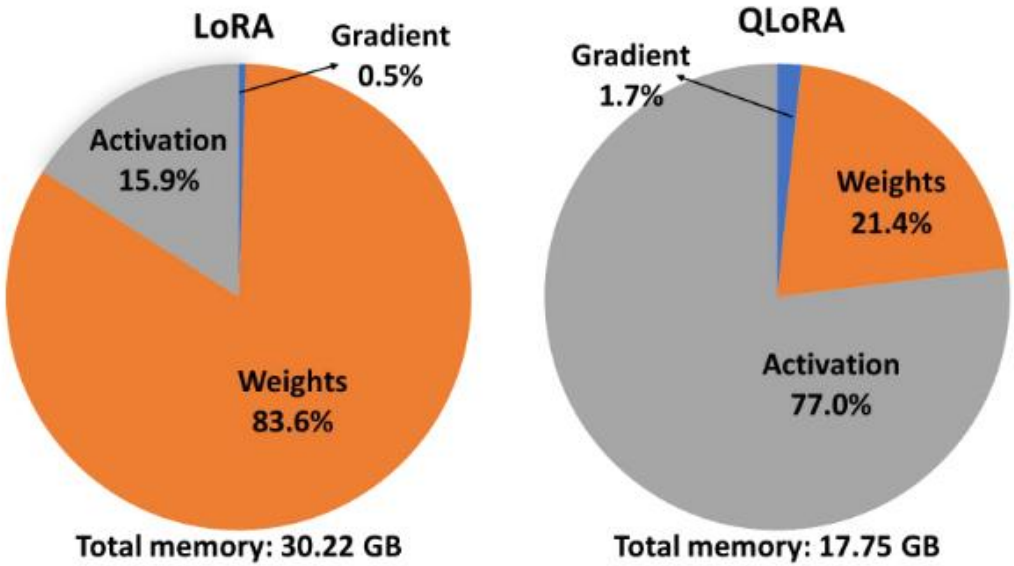


Figure 4. The overview of our hardware scheduling.



约束：

- 在LLM正向或反向传播过程中，正方形的计算分别取决于其行中的左层或右层。
- 为了计算当前正方形，其所有依赖的输入(权重、激活、缓存)都必须加载到片上SRAM。
- 在任何给定的时间，存储在各级存储中的张量的总大小不得超过其总容量。

3.3 互补的硬件调度模块

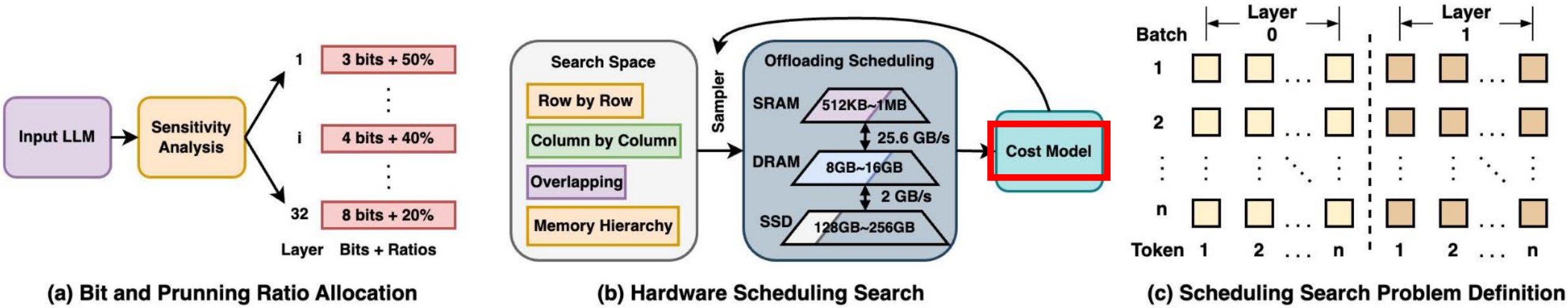
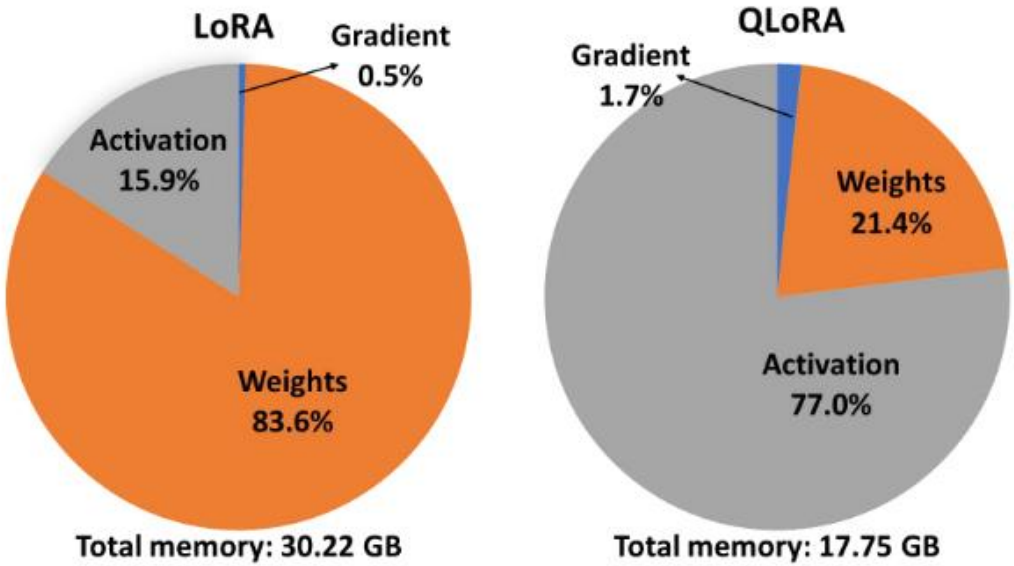


Figure 4. The overview of our hardware scheduling.



约束：

- 在LLM正向或反向传播过程中，正方形的计算分别取决于其行中的左层或右层。
- 为了计算当前正方形，其所有依赖的输入(权重、激活、缓存)都必须加载到片上SRAM。
- 在任何给定的时间，存储在各级存储中的张量的总大小不得超过其总容量。

$$T_{\text{dec}} = \max(r_{\text{to\_sram}}, w_{\text{to\_dram}}, r_{\text{to\_dram}}, w_{\text{to\_ssd}}, T_{\text{comp}}) \quad (3)$$

提纲

4.1 实验配置

4.1 实验配置

- 数据集

MLU、WikiText

4.1 实验配置

- 数据集

MLU、WikiText

- 模型

LLaMA-7B

4.1 实验配置

- 数据集

MLU、WikiText

- 模型

LLaMA-7B

- 算法Baseline

LoRA [arXiv 21]、LST [NeurIPS'22]、 Sparse-GPT [ICML'23] 、 LLM-QAT [arXiv 23]

论文方法的7种变体

4.1 实验配置

- 数据集

MLU、WikiText

- 模型

LLaMA-7B

- 算法Baseline

LoRA [arXiv 21]、LST [NeurIPS'22]、 Sparse-GPT [ICML'23] 、 LLM-QAT [arXiv 23]

论文方法的7种变体

- 硬件Baseline

Shao et al. [VLSI'23]

4.1 实验配置

- 数据集

MLU、WikiText

- 模型

LLaMA-7B

- 算法Baseline

LoRA [arXiv 21]、LST [NeurIPS'22]、 Sparse-GPT [ICML'23] 、 LLM-QAT [arXiv 23]

论文方法的7种变体

- 硬件Baseline

Shao et al. [VLSI'23]

- 硬件配置

DRAM：8GB LPDDR4　　SRAM：1MB

4.2 实验结果

Table 1. Benchmarking Edge-LLM on MMLU dataset.

Method	Avg. Bit	Sparsity	Norm. Mem.	MMLU
LoRA	8.0	0%	1.00×	33.60
Partial Tuning	5.0	50%	0.25×	30.94
Ours	5.1	50%	0.25×	**31.64**
LST	4.0	0%	0.29×	29.04
Partial Tuning	4.0	50%	0.25×	28.70
Ours	4.1	50%	0.25×	**29.89**
Partial Tuning	3.0	50%	0.25×	26.61
Ours	3.1	50%	0.25×	**27.68**

Table 2. Ablation on LUC's performance with its variants

Method	Avg. Bit	Sparsity	Perplexity
SparseGPT	8.0	50%	15.88
LLM-QAT	8.0	0%	13.34
Uniform	5.0	50%	17.61
Random	5.1	50%	16.21
Ours	5.1	50%	**15.71**
Uniform	4.0	50%	19.86
Random	4.1	50%	19.81
Ours	4.1	50%	**18.58**
Uniform	3.0	50%	32.52
Random	3.1	50%	31.71
Ours	3.1	50%	**30.03**

而在WikiText-2上，实现了与LoRA tuning相当的预测效果的前提下，达成了2.92×的时延减小与4×的内存开销减小

提纲

5 总结

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

我的看法

- 文章idea可视化非常形象，思路流畅，比较的baseline与数据集较为丰富

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

我的看法

- 文章idea可视化非常形象，思路流畅，比较的baseline与数据集较为丰富
- 硬件优化部分的使用图遍历的方式建模batch与token的计算调度，思路新颖

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

我的看法

- 文章idea可视化非常形象，思路流畅，比较的baseline与数据集较为丰富
- 硬件优化部分的使用图遍历的方式建模batch与token的计算调度，思路新颖
- 软件部分的设计比较简单直接，实验部分的图表呈现不太完整

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

我的看法

- 文章idea可视化非常形象，思路流畅，比较的baseline与数据集较为丰富
- 硬件优化部分的使用图遍历的方式建模batch与token的计算调度，思路新颖
- 软件部分的设计比较简单直接，实验部分的图表呈现不太完整
- 硬件计算调度部分的建模可以迁移到计算资源、计算精度等优化场景，而非仅优化计算时间。寻路方法也可以加入RL等深度学习方法。

5 总结

文章总结

- 实现了一个LLM的tuning框架EdgeLLM，在边缘设备上实现了高效的LLM tuning
- 实验表明，Edge-LLM 实现了与 vanilla 相当的精度的前提下，

  速度提高了 2.92×，内存开销减少了 4×

我的看法

- 文章idea可视化非常形象，思路流畅，比较的baseline与数据集较为丰富
- 硬件优化部分的使用图遍历的方式建模batch与token的计算调度，思路新颖
- 软件部分的设计比较简单直接，实验部分的图表呈现不太完整
- 硬件计算调度部分的建模可以迁移到计算资源、计算精度等优化场景，而非仅优化计算时间。寻路方法也可以加入RL等深度学习方法。
- 引入多处理器（CPU/GPU/NPU）；infering的优化；根据请求动态地tuning

谢谢聆听！