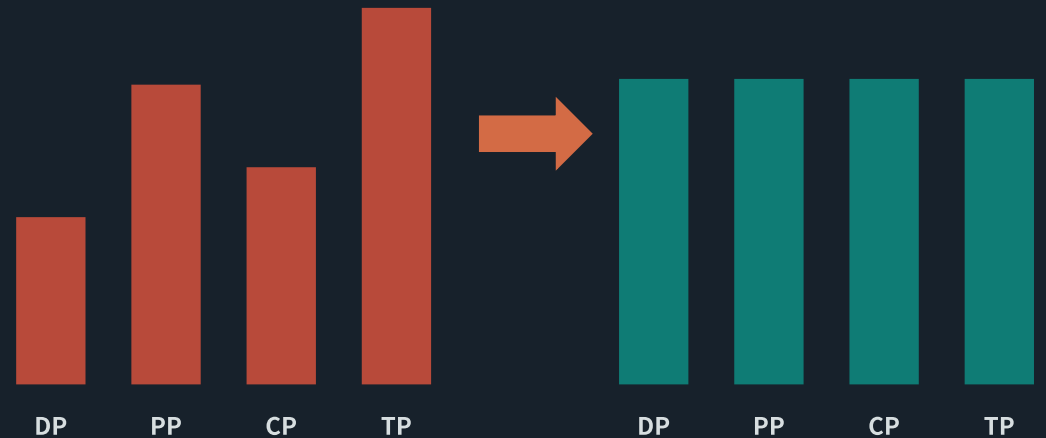


WLB-LLM: Workload-Balanced 4D Parallelism for Large Language Model Training

适用大语言模型训练的 负载均衡型4D并行计算方法

19th USENIX Symposium on Operating
Systems Design and Implementation



1.44×

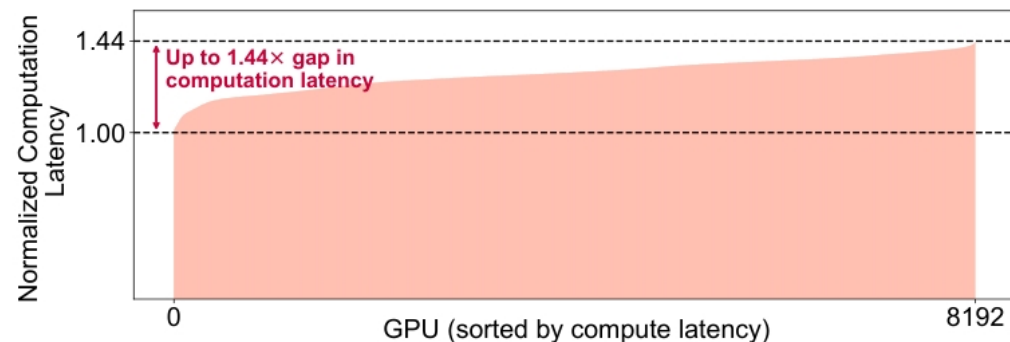
大语言模型训练中 workload 不平衡

1.44x

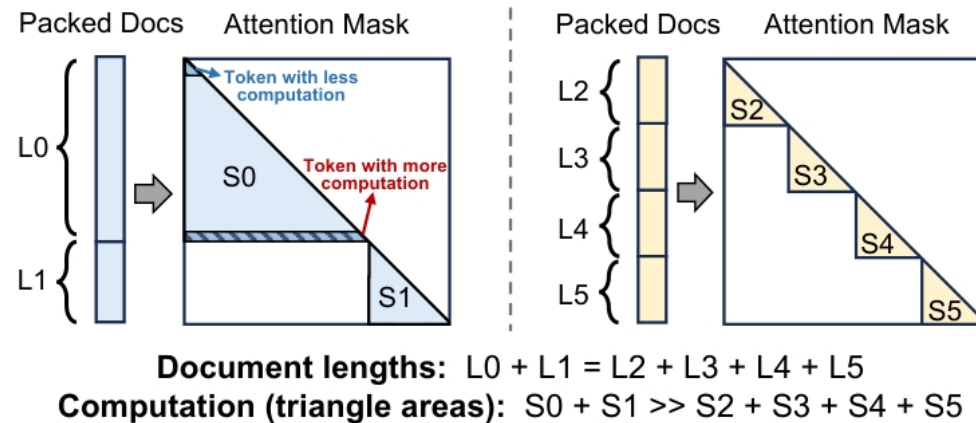
最慢 GPU 的计算延迟

8K H100 • 405B 模型 • 128K 上下文

- 同步训练需要等待最慢 GPU
- 注意力计算具有输入依赖性
- 长文档尾部 token 需关注更多历史



(a) Normalized computation latency in an 8K-GPU 405B LLM training job.



(b) Reason of imbalance: input-dependent nature of attention computation and the varying input document length.

DP、PP、CP、TP

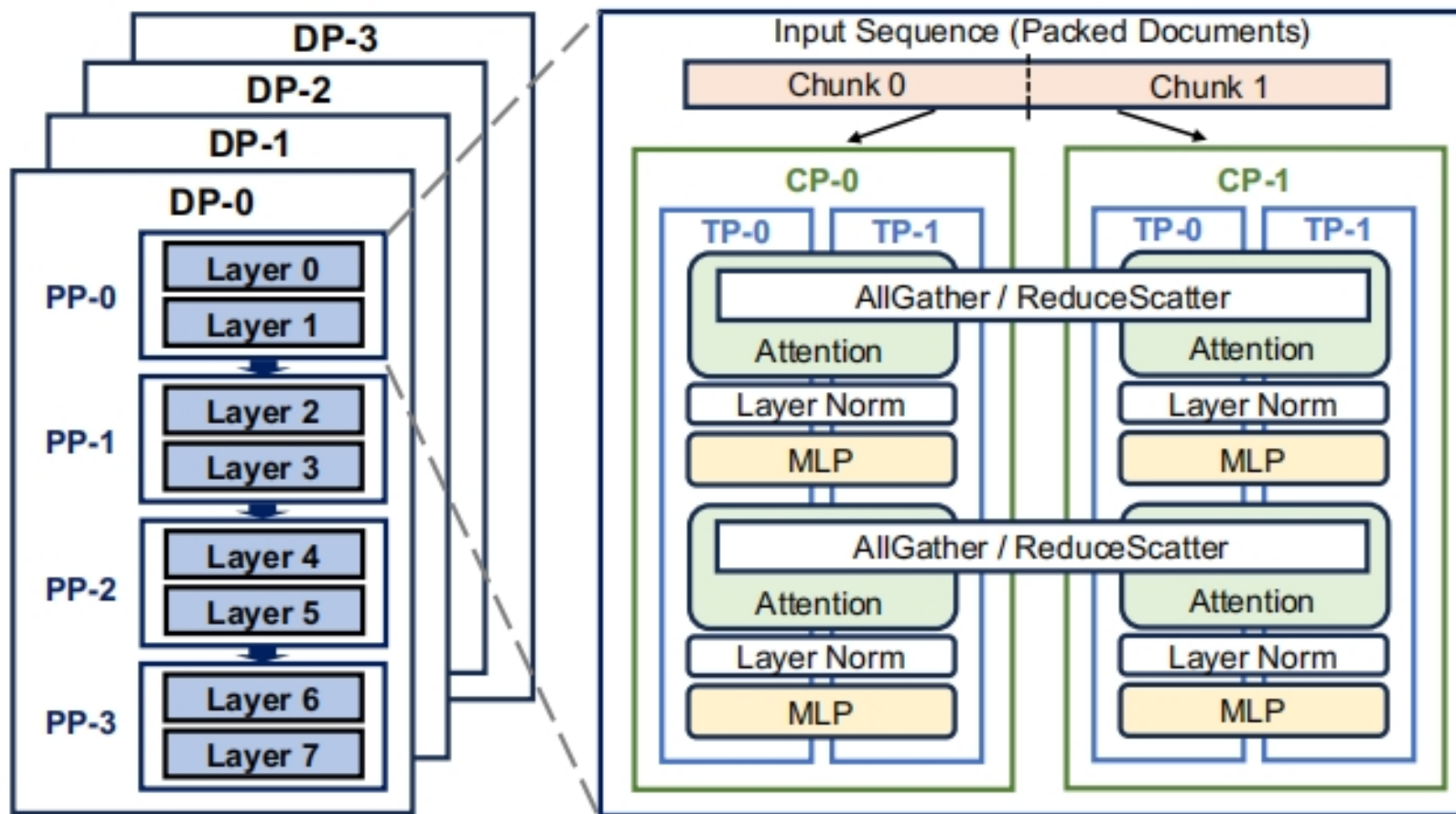
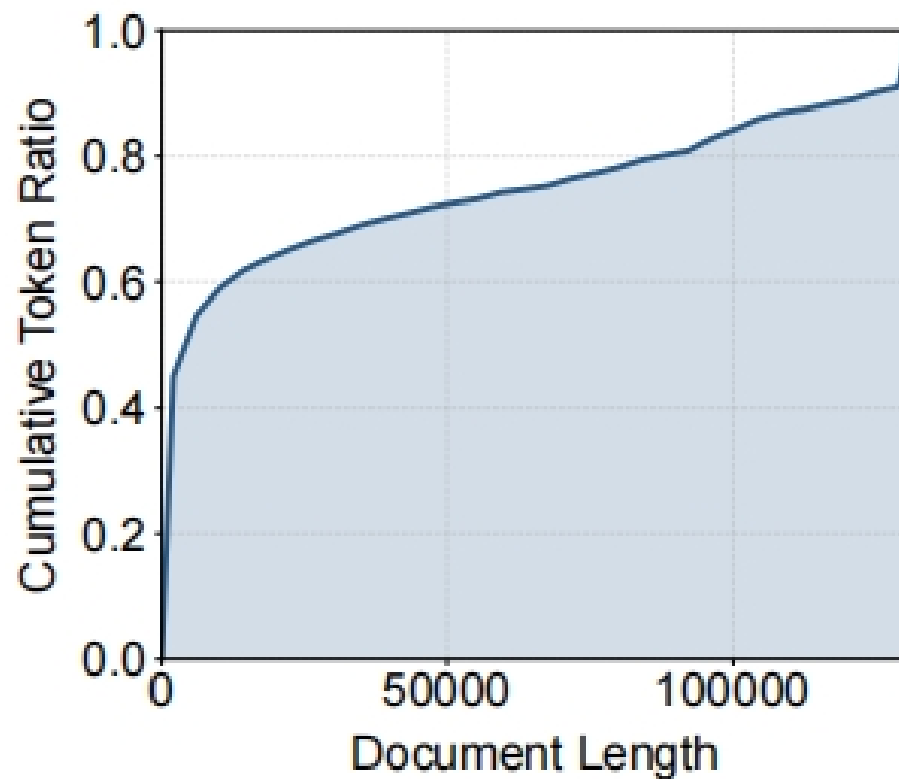
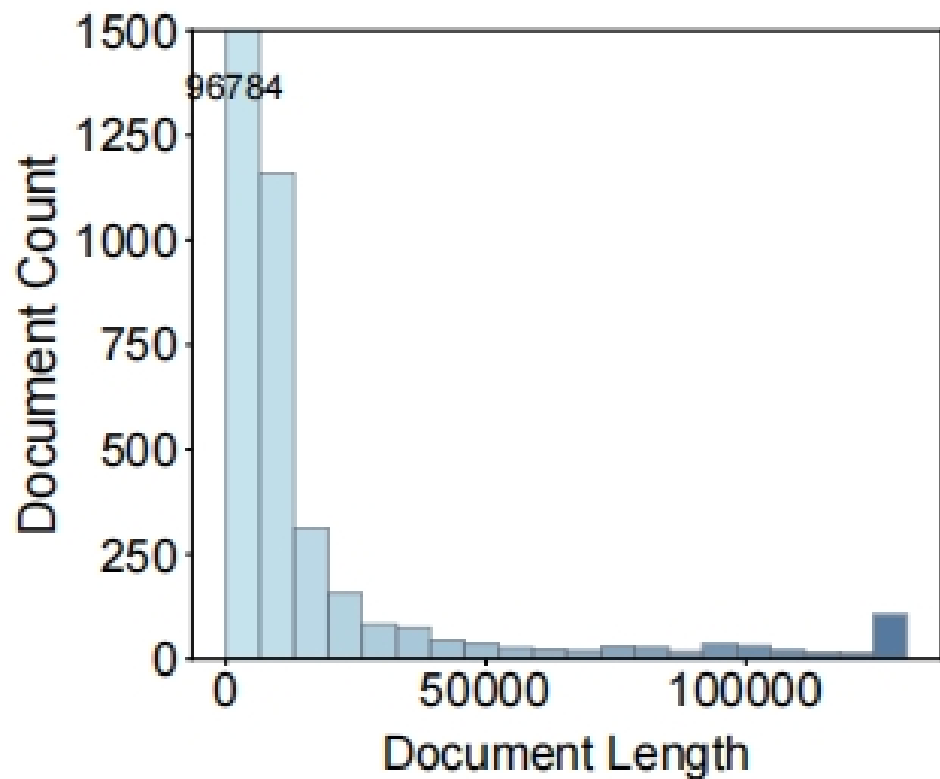


Figure 2: Overview of 4D parallelism for LLM training.

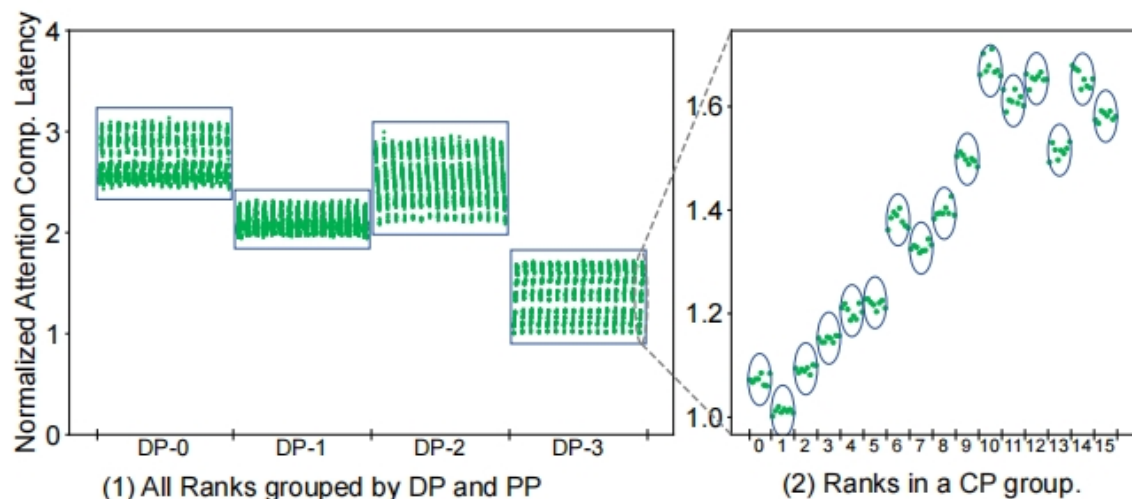
输入文档特征分析



TAKEAWAY

需要根据文档长度变化动态平衡工作负载

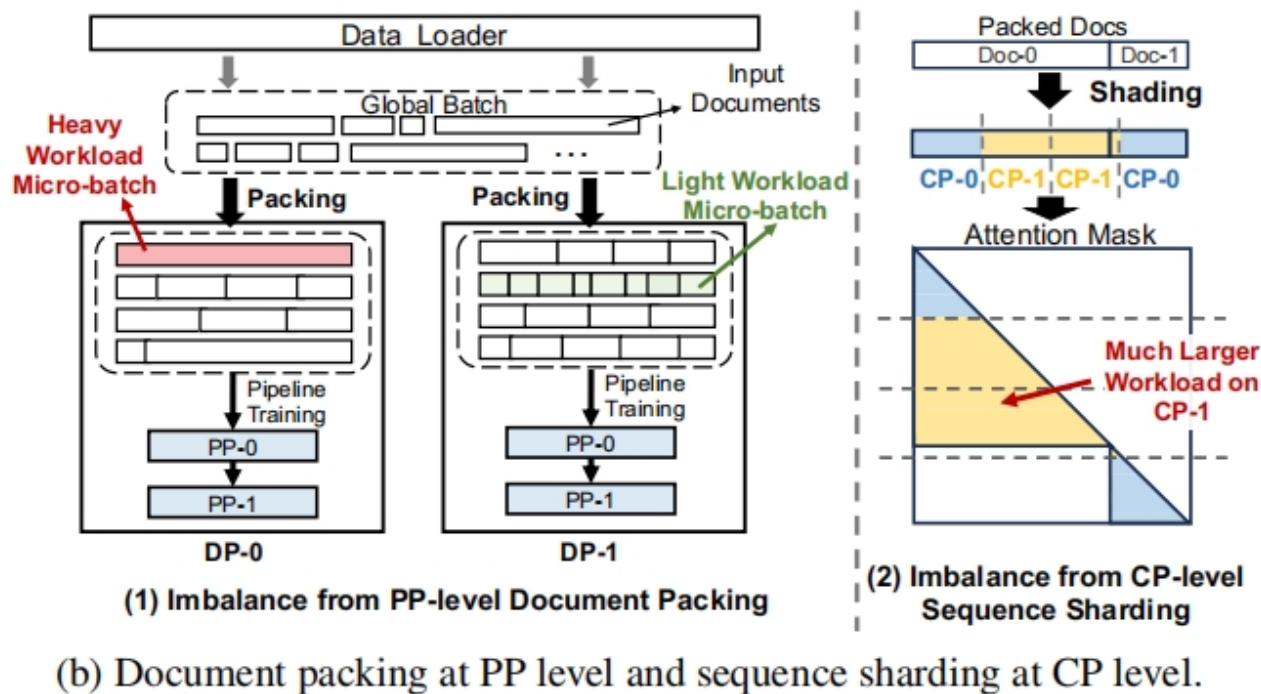
不平衡在 PP 与 CP 两层产生



(a) Imbalance Analysis (TP=8, CP=16, PP=16, DP=4): (1) Normalized computation latency (group by DP and PP); (2) Normalized computation latency in a CP group.

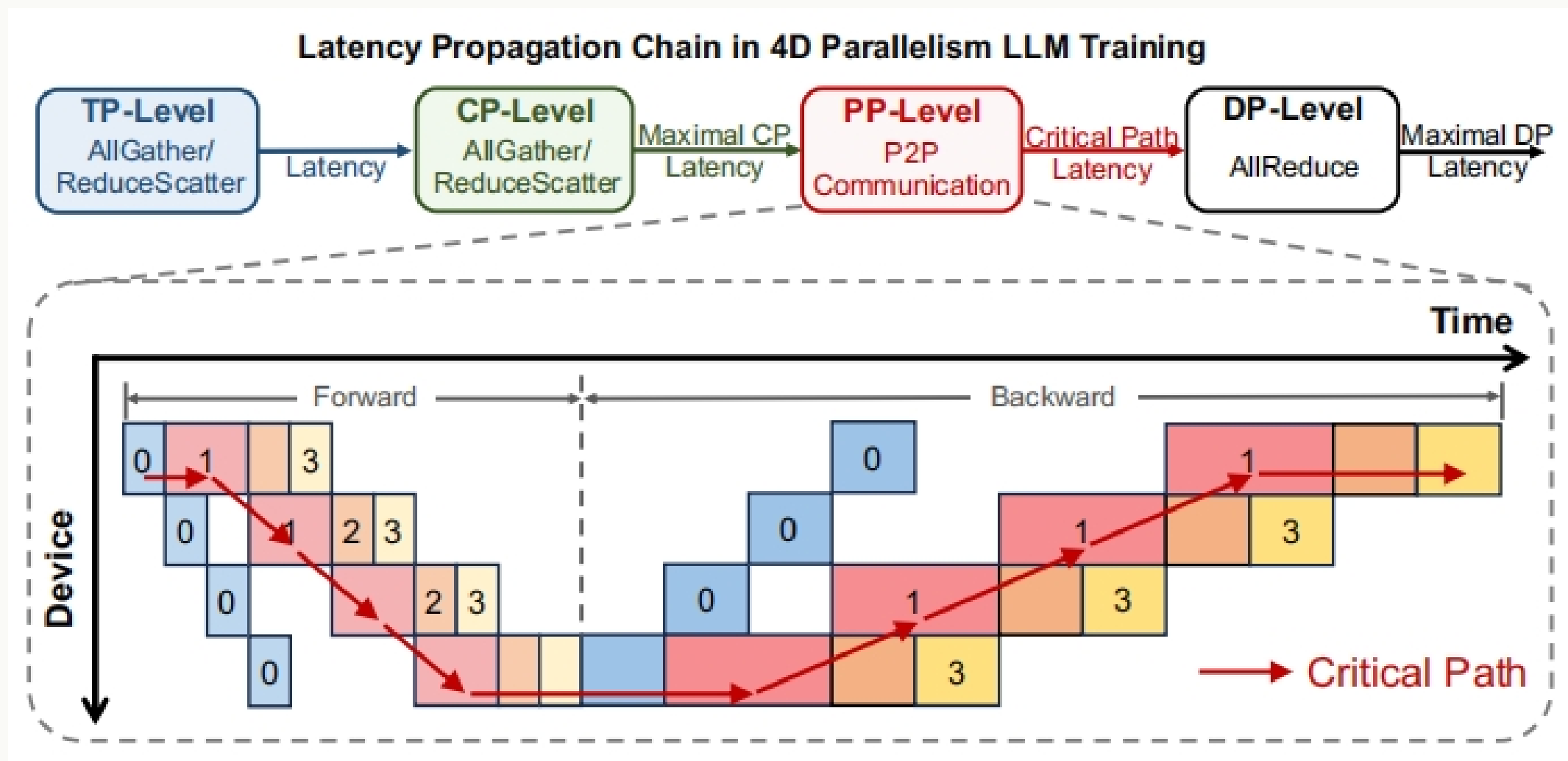
PP • micro-batch

固定长度打包无法对齐 attention 工作量



PP 决定跨 micro-batch 的最大延迟，CP 决定单个 micro-batch 内的最慢 shard。

不均衡沿同步链路放大



4D并行LLM训练延迟传播的过程，工作负载不平衡的影响在PP层面尤为显著。

基准/现有解决方案--优化输入文档的打包方式

以因果掩码为例：

$$\begin{aligned} &\text{minimize} && \max\left(\sum_{i=1}^N x_{ij} \cdot d_i^2\right), j = 1, \dots, M \\ &\text{subject to} && \sum_{j=1}^M x_{ij} = 1, \quad i = 1, \dots, N \\ &&& \sum_{i=1}^N x_{ij} \cdot d_i \leq S, \quad j = 1, \dots, M \\ &&& x_{ij} \in \{0, 1\} \end{aligned}$$

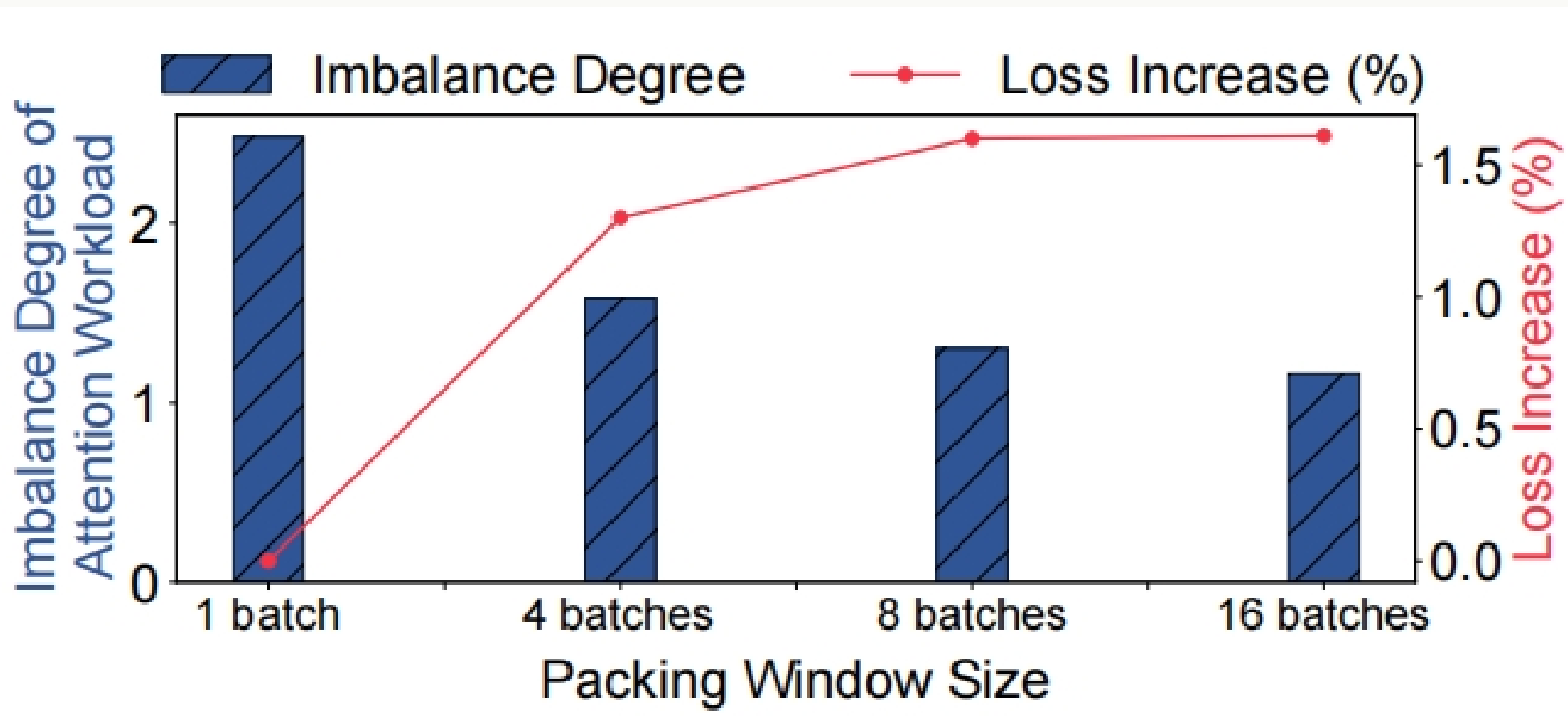
$x_{ij} = 1$ ：表示第 i 个文档放入第 j 个 micro-batch；
每个文档只能放一次；
每个 micro-batch 的总长度不能超过固定长度 S 。

问题一：固定长度限制太强

问题二：跨越太多 batch 会破坏训练数据随机性

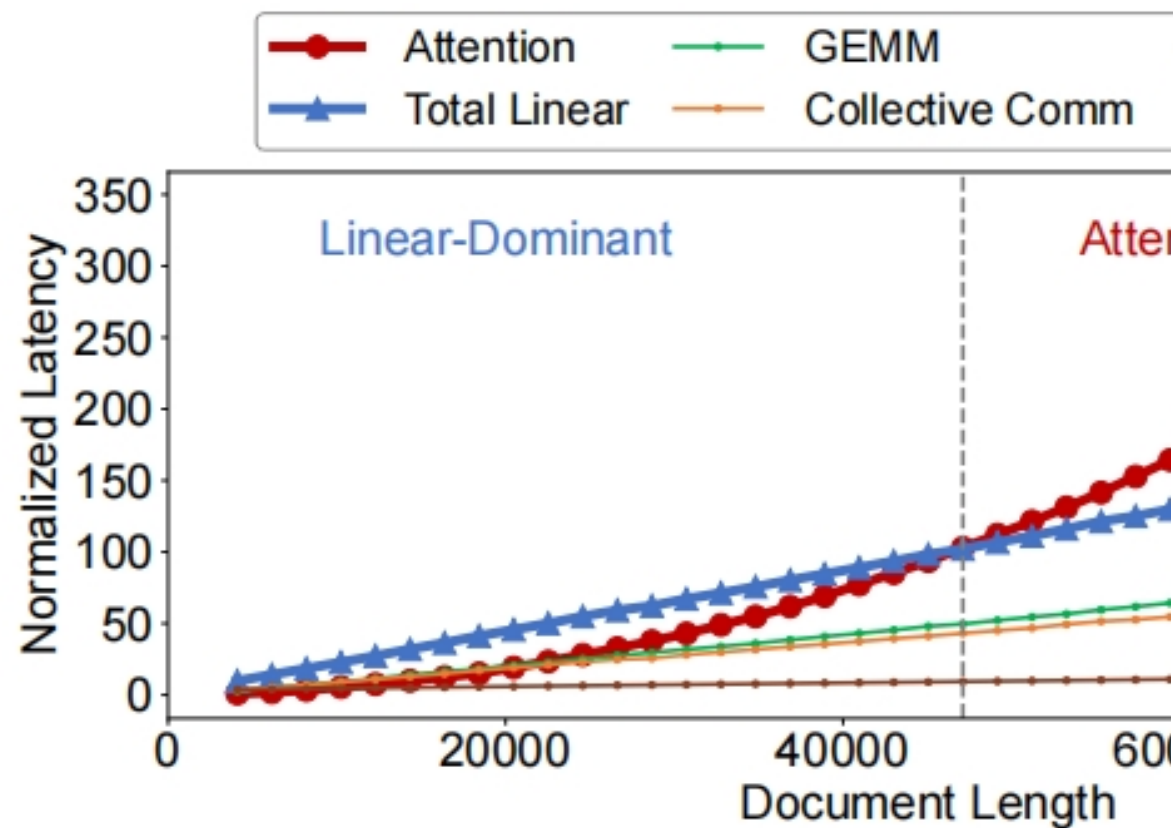
打包平衡与模型质量之间的权衡关系

$$\frac{Max_Attn}{Avg_Attn}$$



打包窗口越大，负载越均衡，但模型训练损失可能越高。

PP1:基于工作负载的可变长度打包



minimize $\max(\sum_{i=1}^N (W_a(x_{ij} \cdot d_i) + W_l(x_{ij} \cdot d_i))),$
 $j = 1, \dots, M$

subject to $\sum_{j=1}^M x_{ij} = 1,$
 $i = 1, \dots, N$

$\sum_{i=1}^N x_{ij} \cdot d_i \leq S_{max},$
 $j = 1, \dots, M$

$x_{ij} \in \{0, 1\}$

Llama2 7B模型在16个H100 GPU上执行训练，以文档长度为4096时的注意力计算延迟作为基准进行归一化。

PP2:自适应延迟处理异常(极长)文档

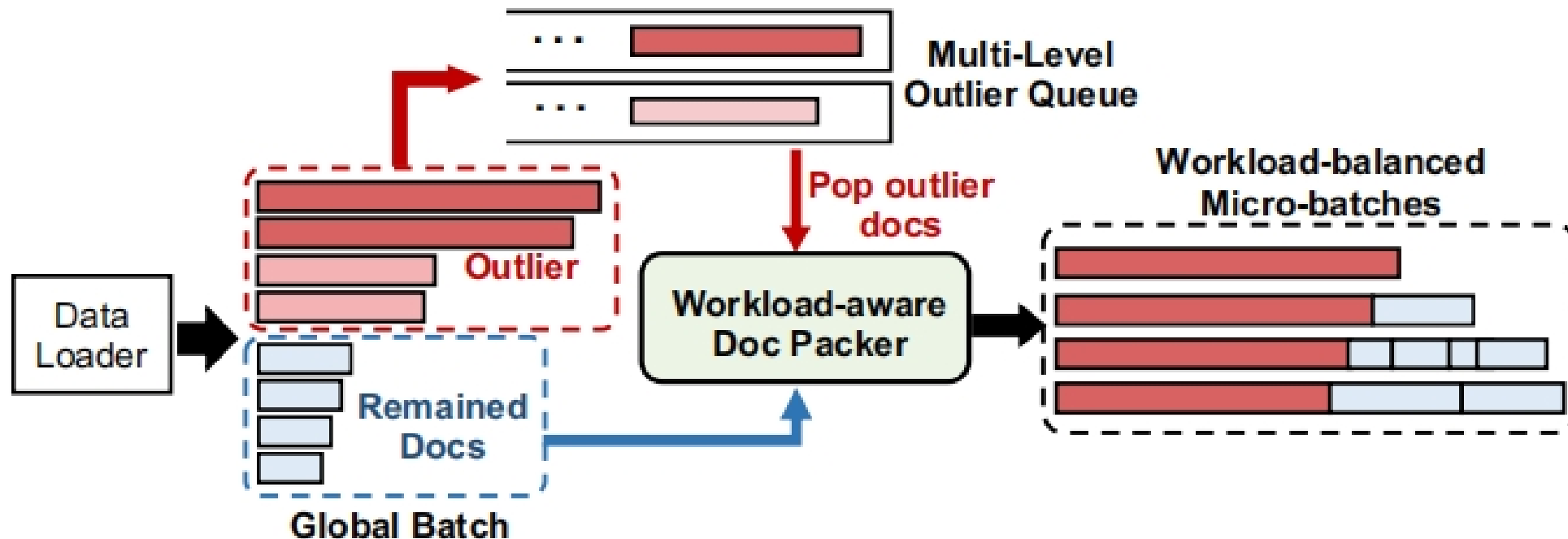
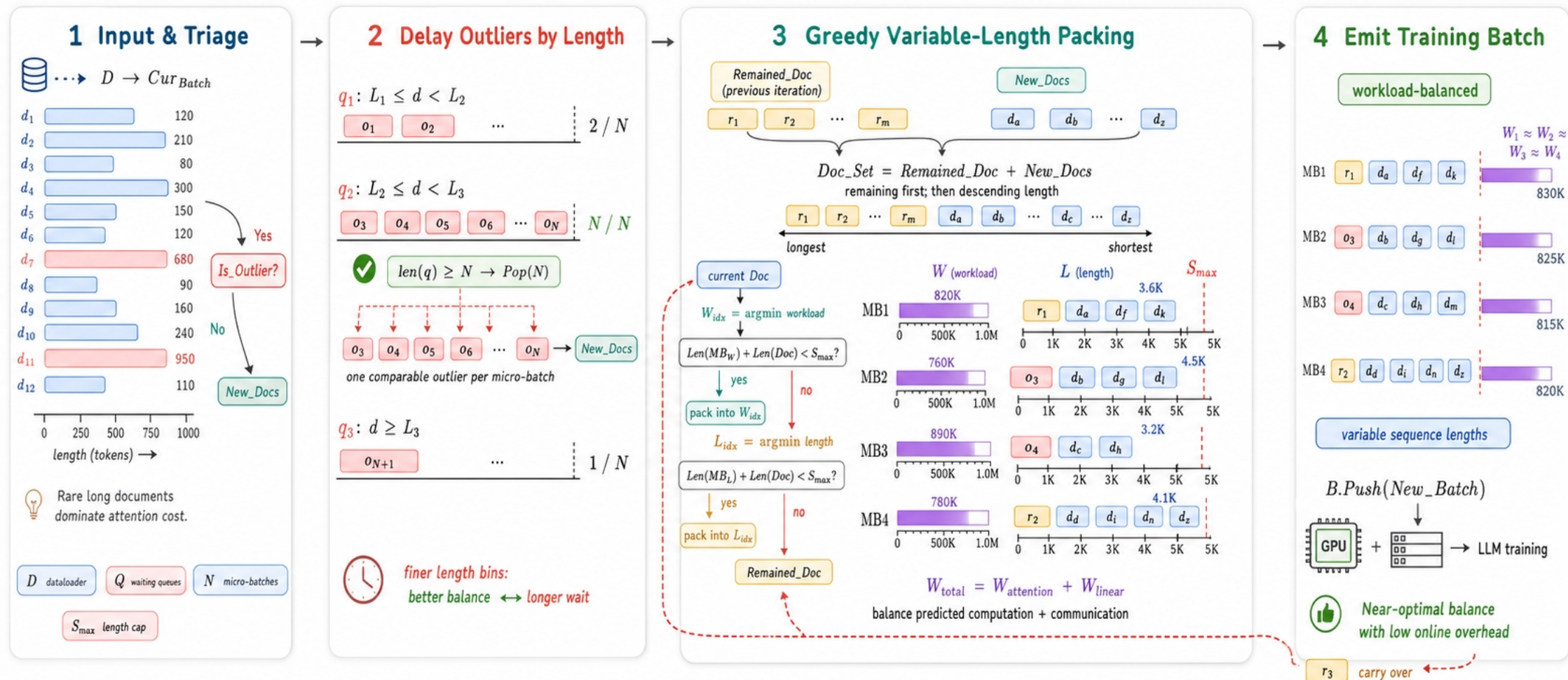


Figure 8: The process of outlier document delay combined with var-length packing.

PP3:启发式打包算法

WLB-LLM Heuristic Variable-Length Packing

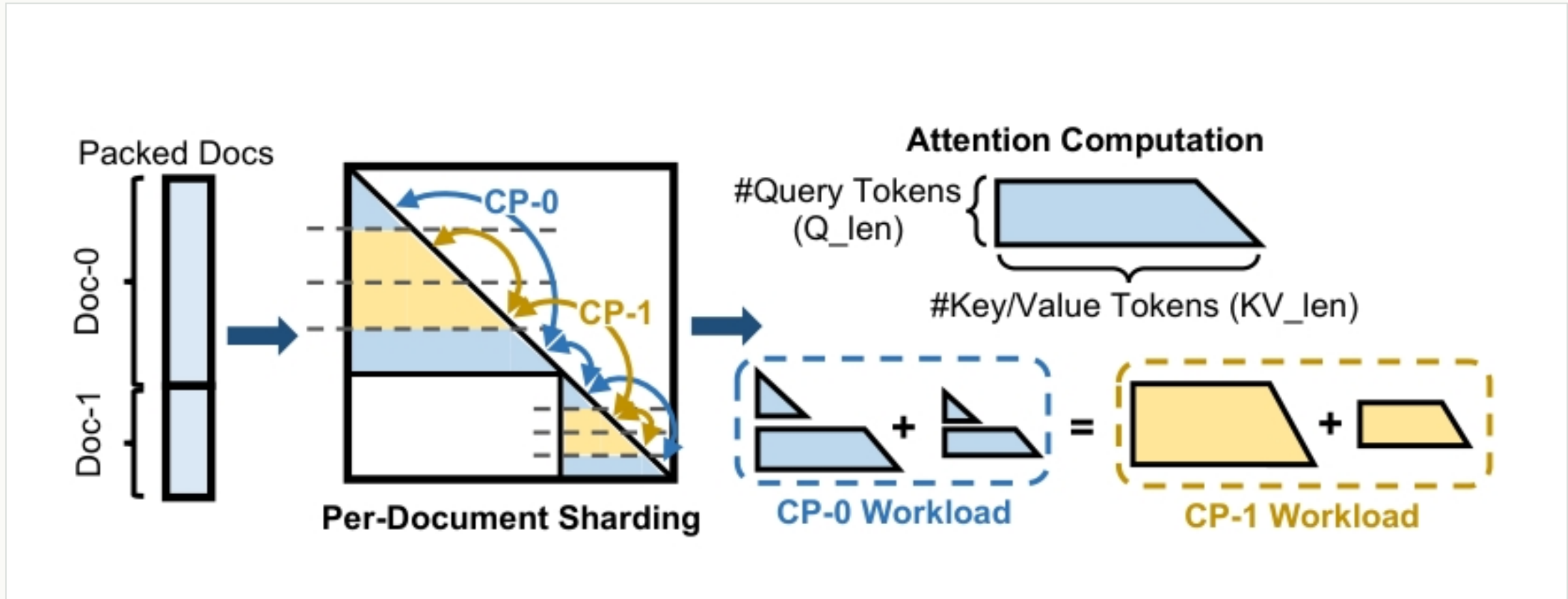
Delay rare outliers, then balance total workload—not token count.



normal doc outlier carry-over packed/released workload

Outliers wait in length-aware FIFO queues; every other document is greedily assigned under $S_{\max}$.

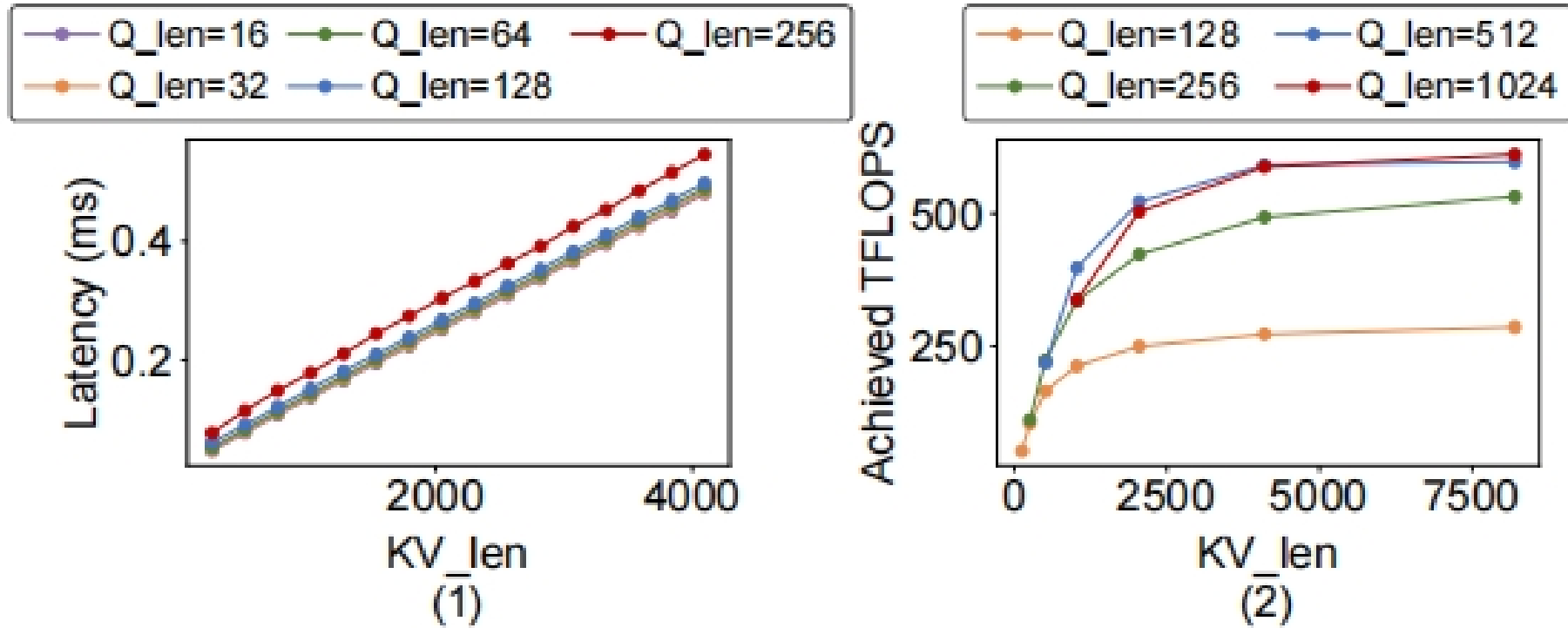
CP1: 按文档切分设计，消除切片间工作量差



Per-Seq token 数相等
attention 工作量不等

Per-Doc token 数相等
attention 工作量相等

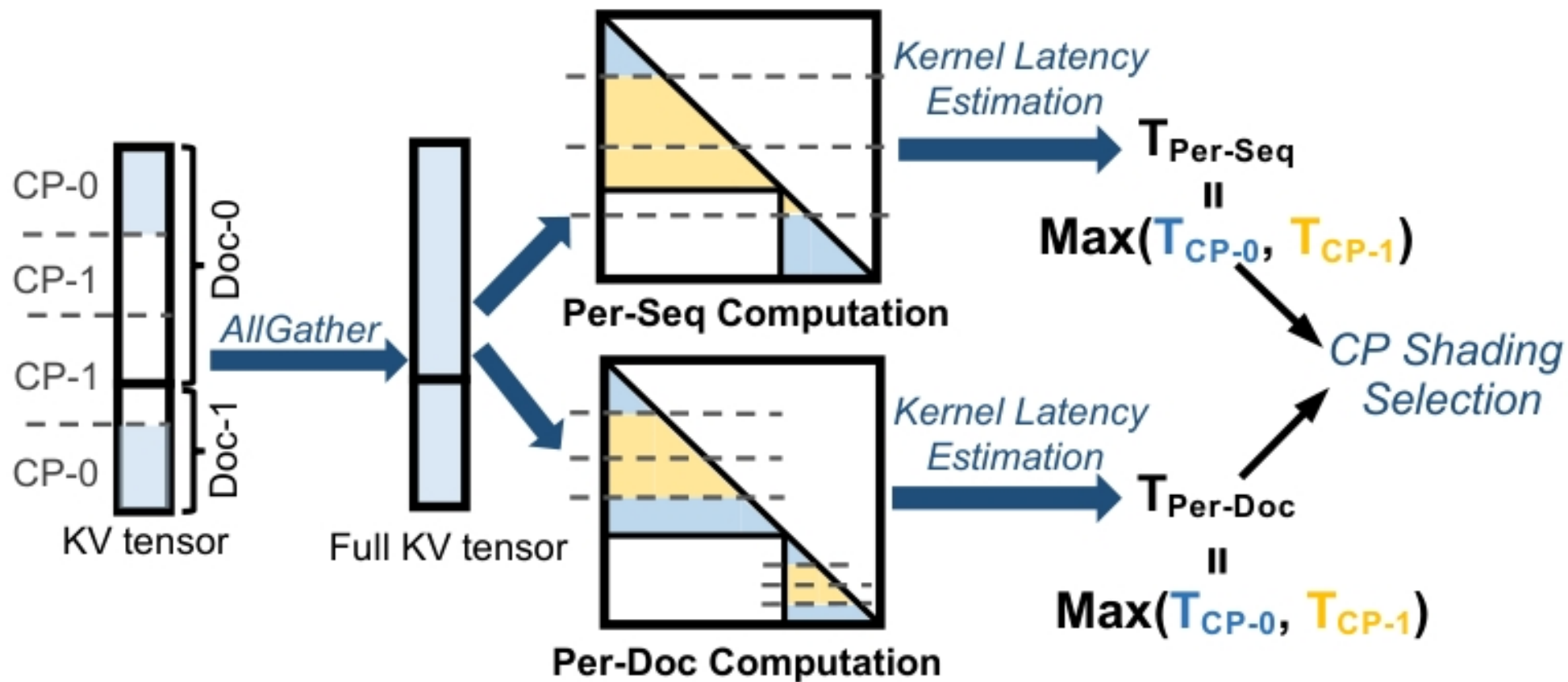
CP2: 自适应分片选择

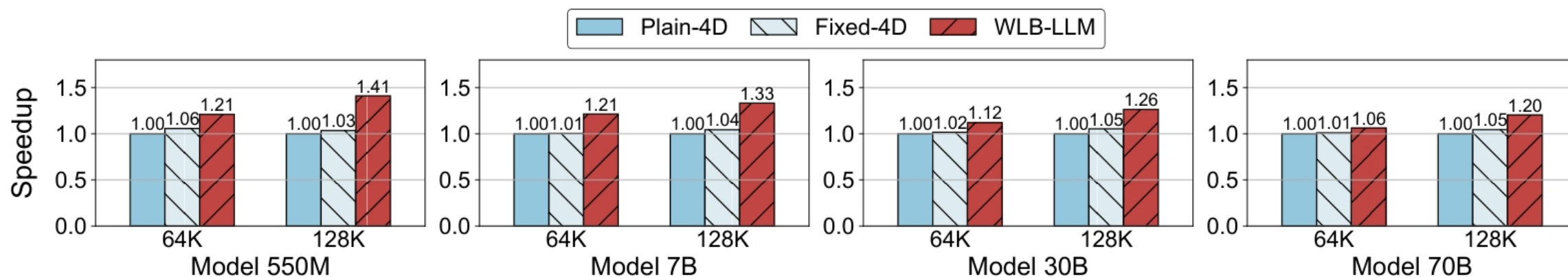


TAKEAWAY

更均衡的分片vs.更高效的注意力核

CP2: 自适应分片选择





TAKEAWAY

在不同LLaMA类模型下，WLB-LLM与Fixed-4D相较于Plain-4D的训练性能提升效果

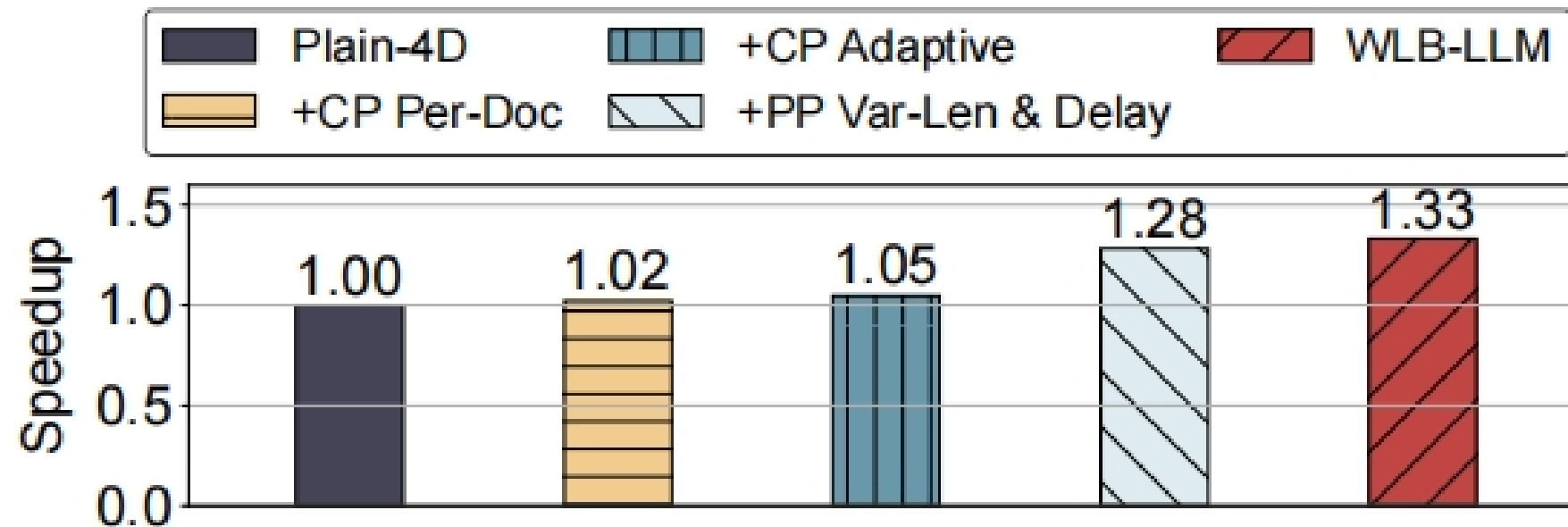


Figure 13: Performance breakdown of *WLB-LLM* on the 7B model with a 128K context window.

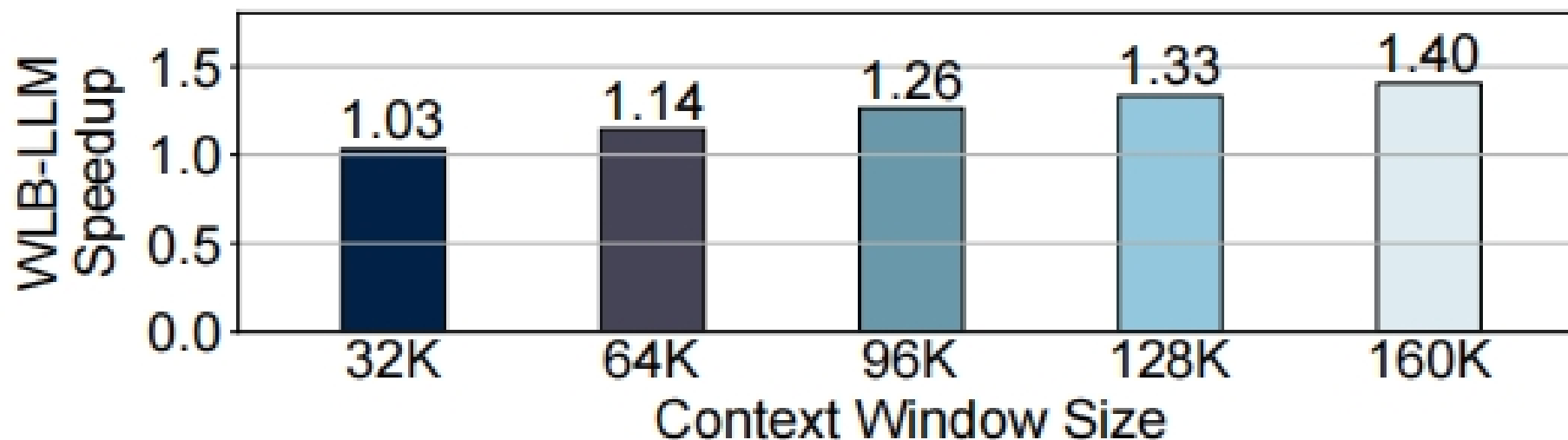


Figure 14: Speedups of *WLB-LLM* on the 7B model across context window sizes.

Packing Method		Imbalance Degree	Packing Overhead (ms)
Method	Config		
<i>Original Packing</i>	/	1.44	0
<i>Fixed-Len Greedy</i>	#global batch=1	1.41	4
	#global batch=2	1.22	5
	#global batch=4	1.11	5
	#global batch=8	1.08	5
<i>Fixed-Len Solver</i>	#global batch=1	1.40	467
	#global batch=2	1.18	1488
	#global batch=4	1.09	25313
<i>WLB-LLM</i>	#queue=1	1.24	8
	#queue=2	1.05	20
	#queue=3	1.05	23

Table 2: Packing imbalance degree and overhead analysis.

TAKEAWAY

WLB-LLM能够以几十毫秒以内的开销，把不均衡度从1.44降到约1.05

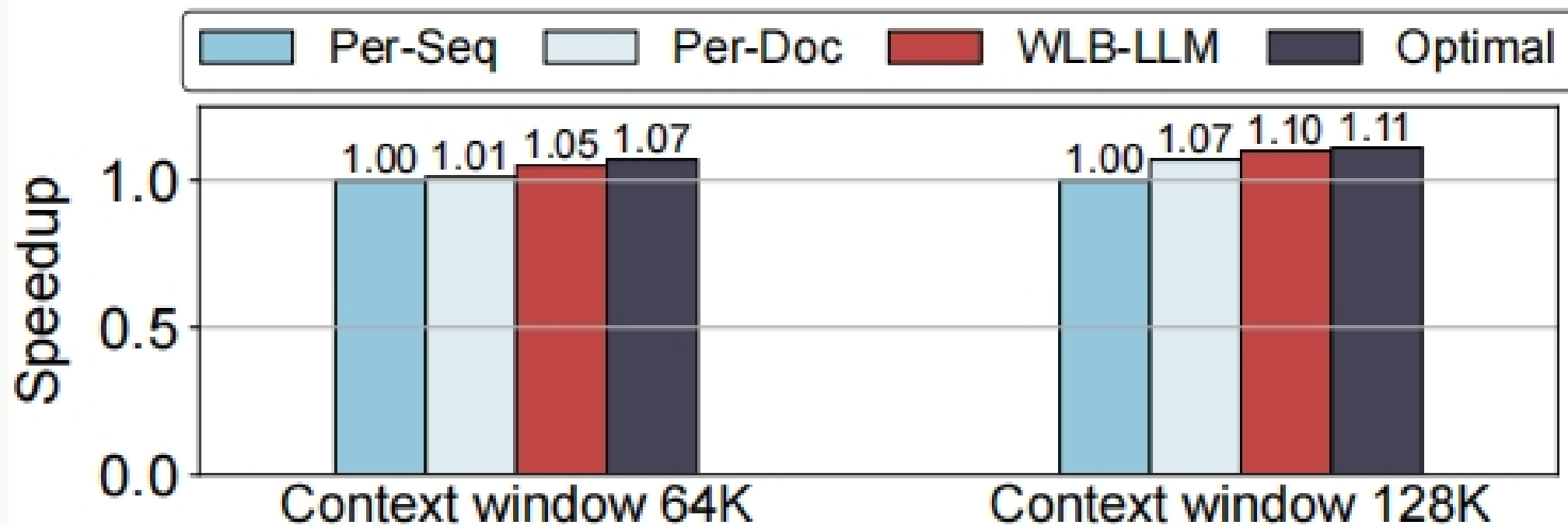


Figure 15: CP sharding performance comparison.

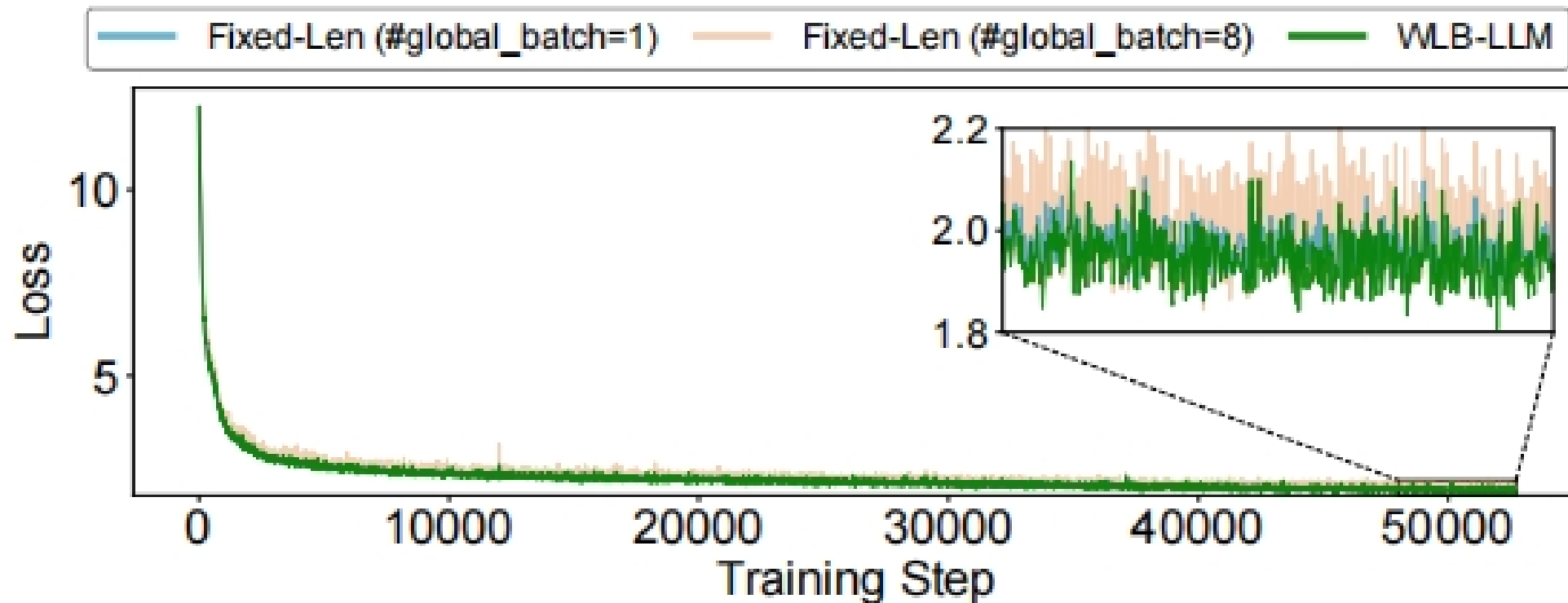


Figure 16: Training loss comparison on a 550M model.

一句话总结

从按 token 数均分， 转向按计算与通信工作量均衡

PP

变长打包 + outlier delay

CP

Per-Doc + 自适应选择

E2E

平均 1.23×，不改收敛趋势

1.23×

平均端到端训练加速

Take-home: 长上下文训练的调度单位不应只是 token，
而应是输入诱导的真实 workload。