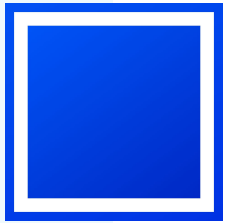
Balancing Pipeline Parallelism with Vocabulary Parallelism

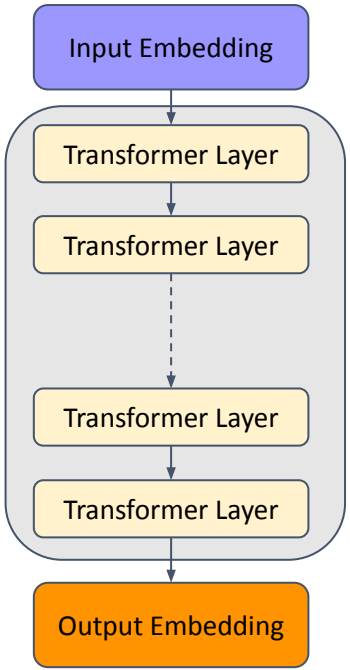
Man Tsung Yeung[1 2], Penghui Qi[1 2], Min Lin[1], Xinyi Wan[1]

[1]Sea AI Lab [2]National University of Singapore



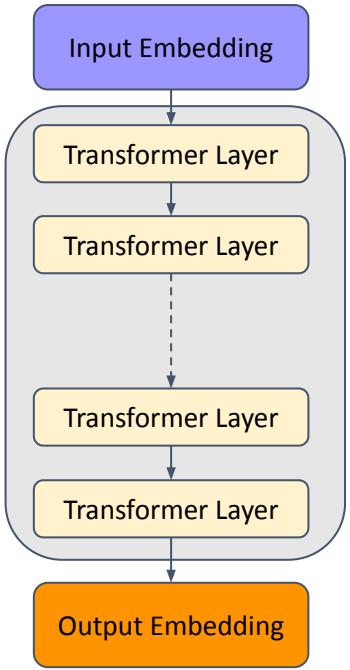
Motivation

Motivation

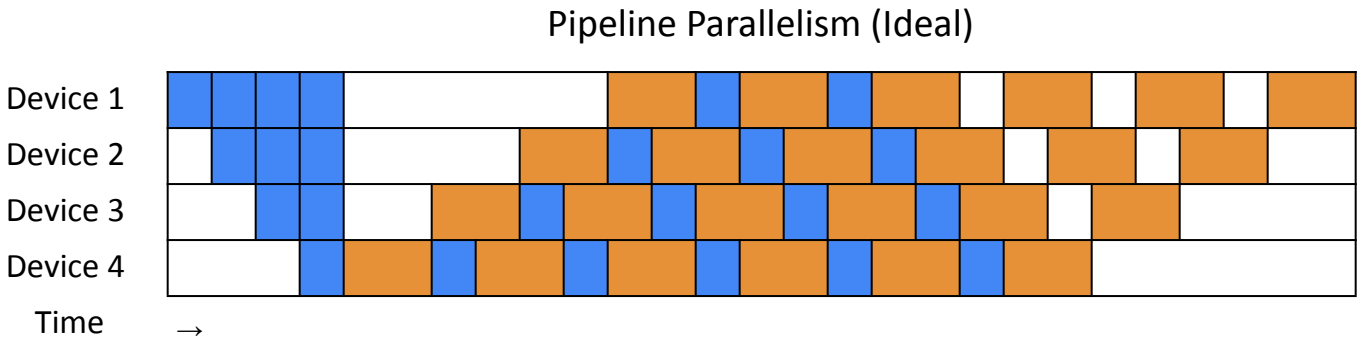


Decoding-only language models

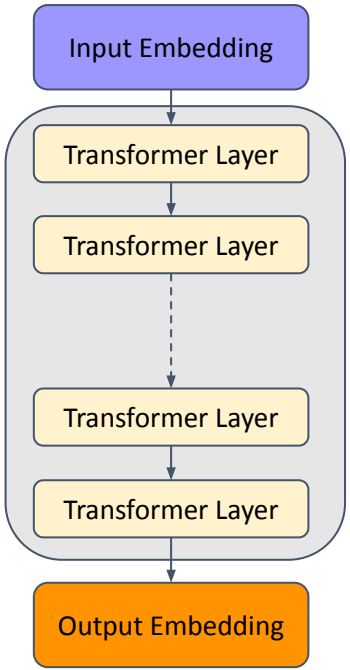
Motivation



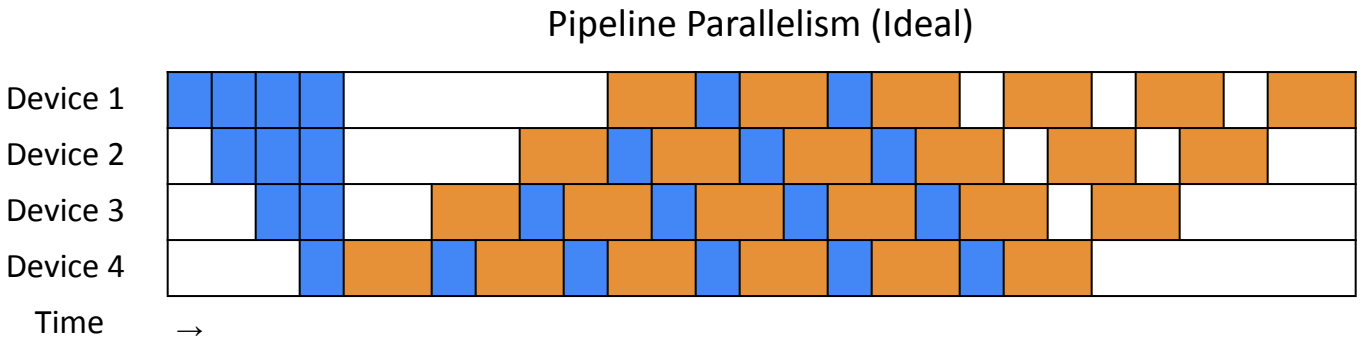
Decoding-only language models

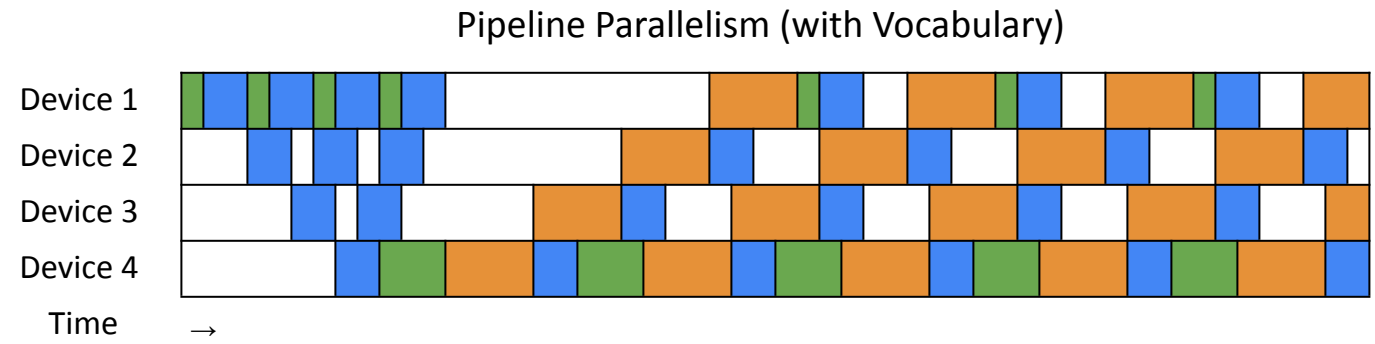


Motivation

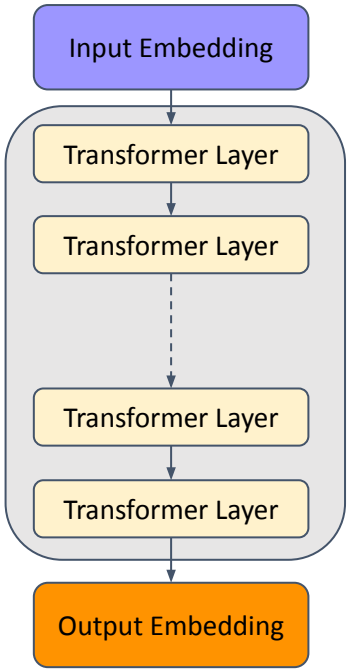


Decoding-only language models

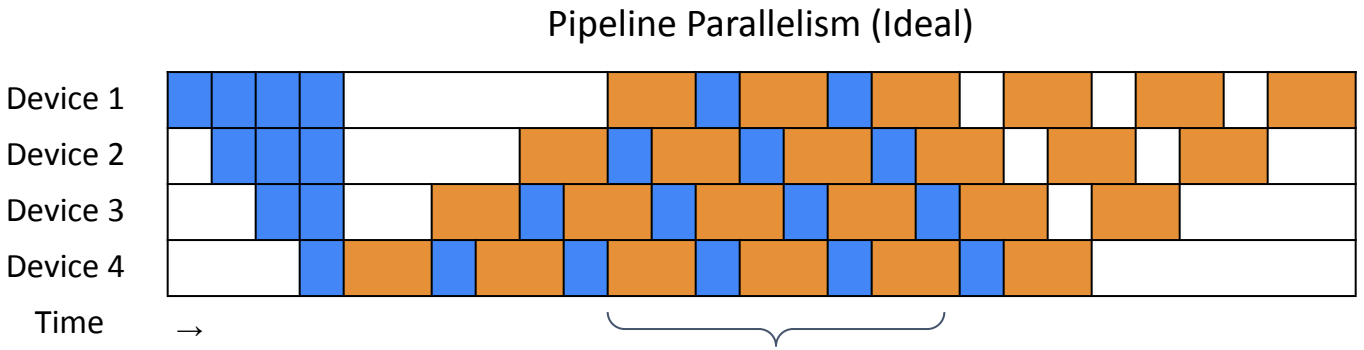


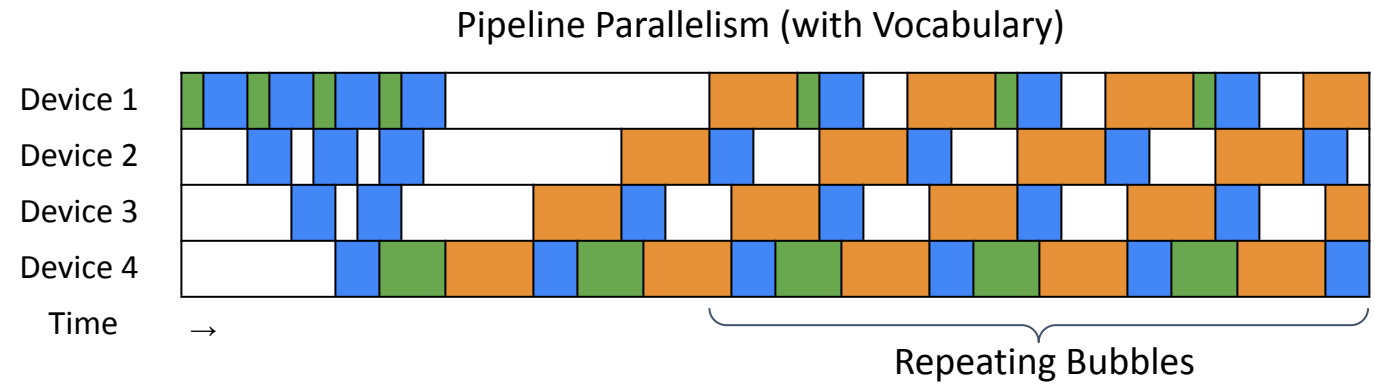


Motivation



Decoding-only language models





Motivation

- Growing vocabulary sizes, especially for multilingual LLMs.

Model	Activated Parameter Per Transformer Layer (M)	Hidden Size	Vocabulary Size	Ratio of activated parameters between output and transformer layer
Gemma2-9B	198	3584	256000	4.63
Gemma2-27B	566	4608	256000	2.08
Deepseek-v3-671B	607	7168	129280	1.53
Qwen2-7B	231	3584	152064	2.36
Qwen2-72B	884	8192	152064	1.41
Llama3-7B	234	4096	128000	2.24
Llama3-70B	862	8192	128000	1.22
Llama3-405B	3198	16384	128000	0.66
Mixtral-8x22B	355	6144	32768	0.57
Mixtral-8x7B	212	4096	32768	0.63

Motivation

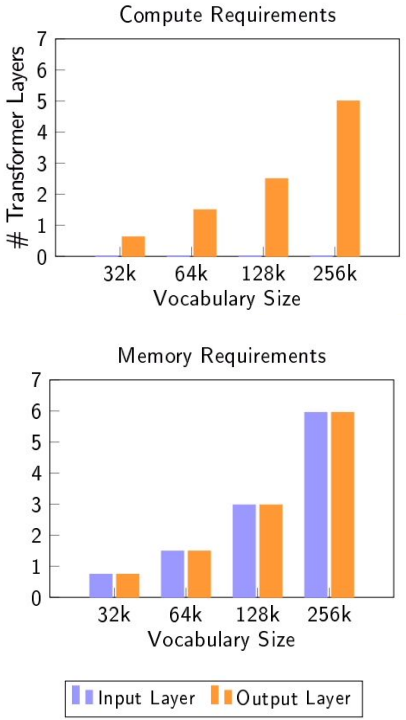
- Growing vocabulary sizes, especially for multilingual LLMs.

Model	Activated Parameter Per Transformer Layer (M)	Hidden Size	Vocabulary Size	Ratio of activated parameters between output and transformer layer
Gemma2-9B	198	3584	256000	4.63
Gemma2-27B	566	4608	256000	2.08
Deepseek-v3-671B	607	7168	129280	1.53
Qwen2-7B	231	3584	152064	2.36
Qwen2-72B	884	8192	152064	1.41
Llama3-7B	234	4096	128000	2.24
Llama3-70B	862	8192	128000	1.22
Llama3-405B	3198	16384	128000	0.66
Mixtral-8x22B	355	6144	32768	0.57
Mixtral-8x7B	212	4096	32768	0.63

Motivation

- Growing vocabulary sizes, especially for multilingual LLMs.

Model	Activated Parameter Per Transformer Layer (M)	Hidden Size	Vocabulary Size	Ratio of activated parameters between output and transformer layer
Gemma2-9B	198	3584	256000	4.63
Gemma2-27B	566	4608	256000	2.08
Deepseek-v3-671B	607	7168	129280	1.53
Qwen2-7B	231	3584	152064	2.36
Qwen2-72B	884	8192	152064	1.41
Llama3-7B	234	4096	128000	2.24
Llama3-70B	862	8192	128000	1.22
Llama3-405B	3198	16384	128000	0.66
Mixtral-8x22B	355	6144	32768	0.57
Mixtral-8x7B	212	4096	32768	0.63

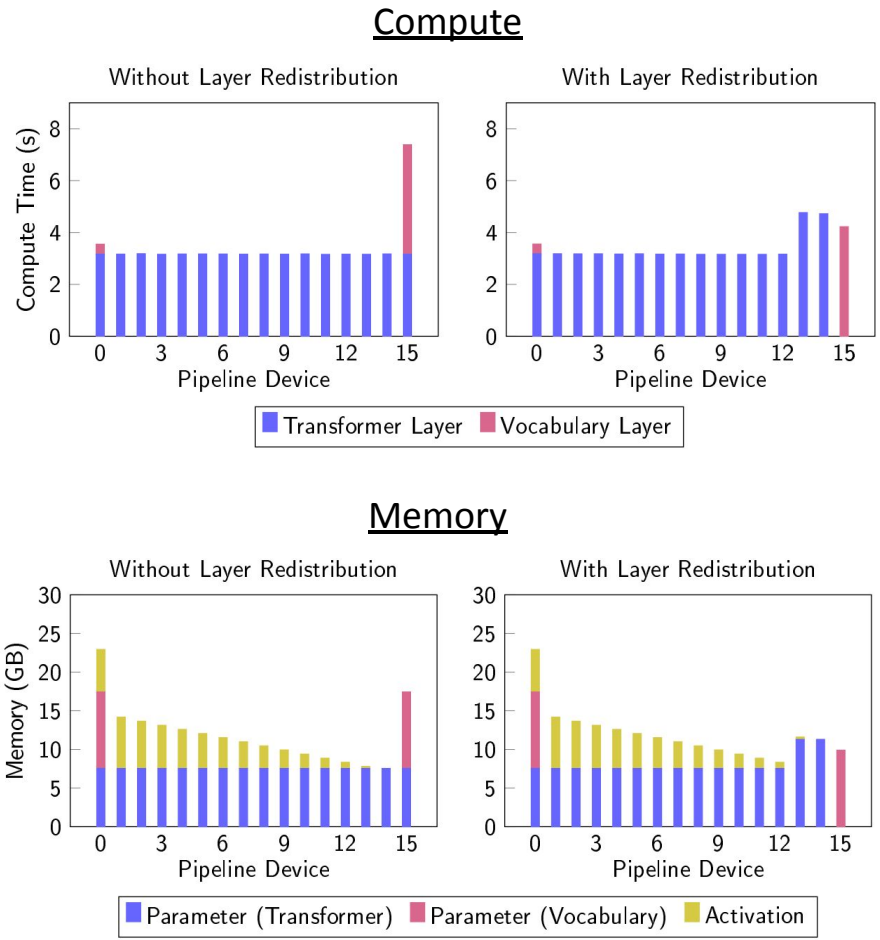


Ratio for Gemma2-9B

Naive Methods

Layer-level Redistribution: DeepSpeed (Smith et al., 2022), Skywork-MoE (Wei et al., 2024).

➔ Unable to perfectly balance both compute and memory.



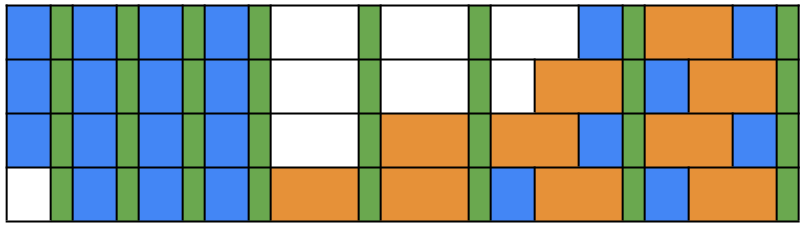
Naive Methods

Layer-level Redistribution: DeepSpeed (Smith et al., 2022), Skywork-MoE (Wei et al., 2024).

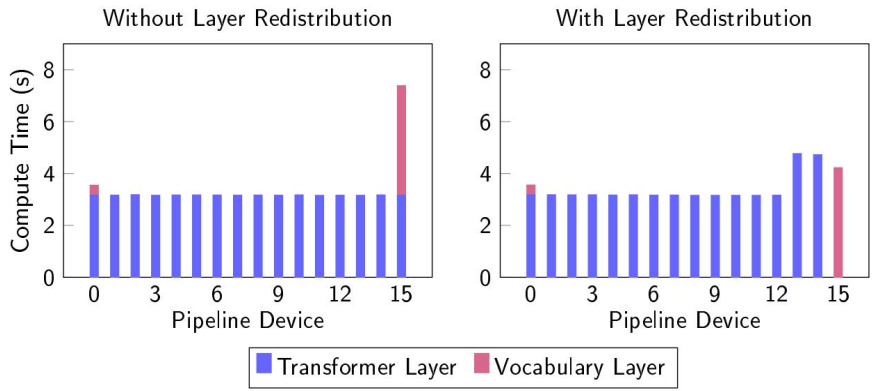
➔ Unable to perfectly balance both compute and memory.

Interlaced Pipeline (Lin et al., 2024).

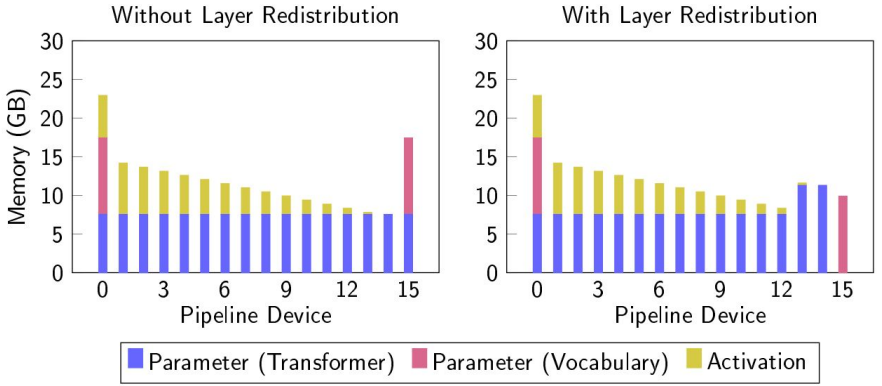
➔ Increases the peak activation memory to 1.5 times of its original value.

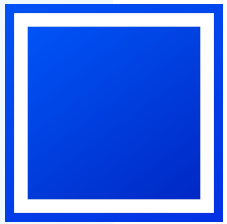


Compute



Memory

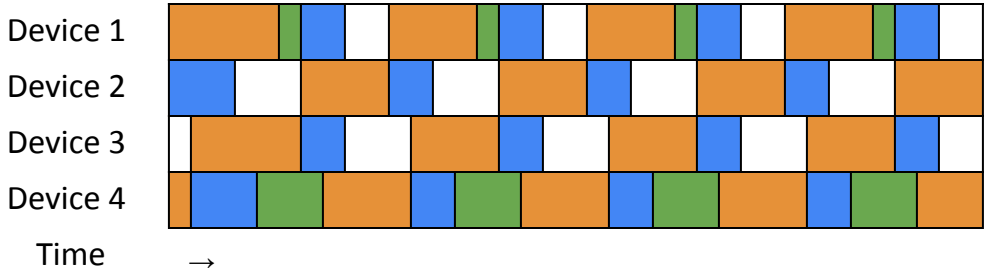




Vocabulary Parallelism

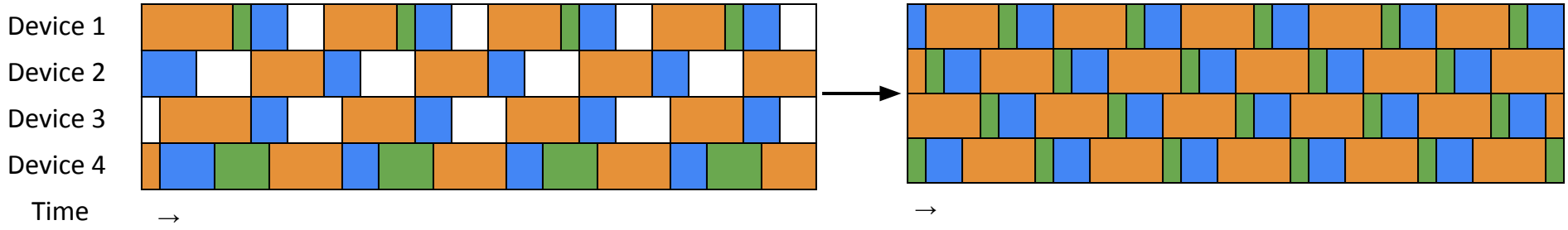
Vocabulary Parallelism: Overview

- Partition the vocabulary layers, distribute them equally across all pipeline devices.



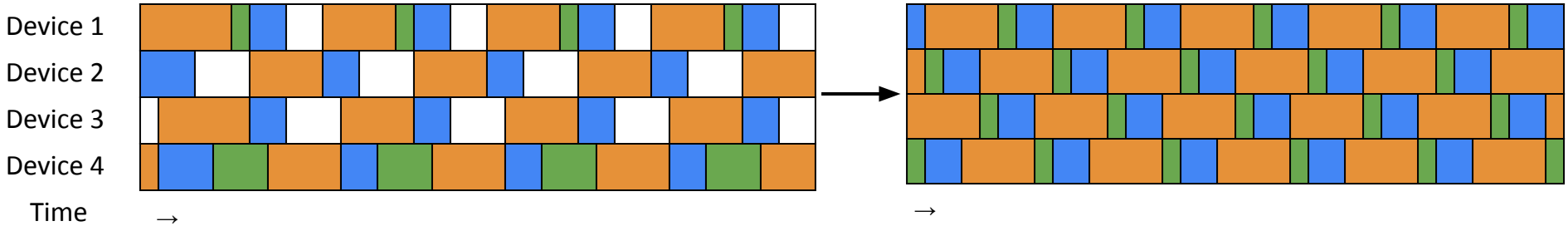
Vocabulary Parallelism: Overview

- Partition the vocabulary layers, distribute them equally across all pipeline devices.



Vocabulary Parallelism: Overview

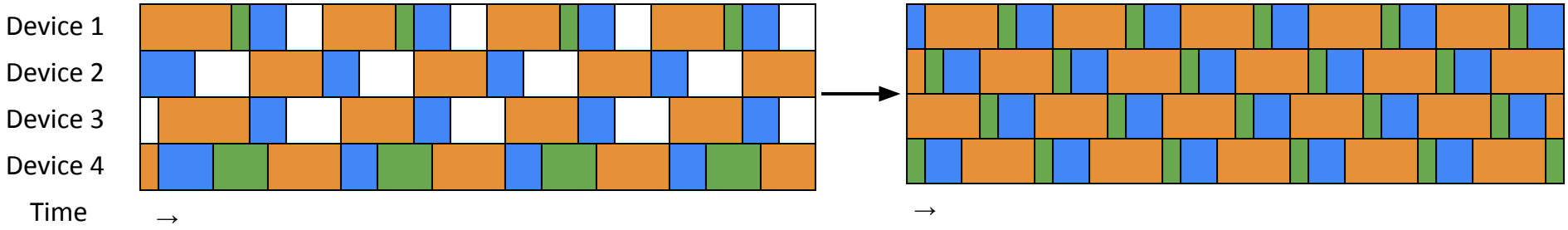
- Partition the vocabulary layers, distribute them equally across all pipeline devices.



✓ Balances out the computation and memory of vocabulary layers

Vocabulary Parallelism: Overview

- Partition the vocabulary layers, distribute them equally across all pipeline devices.



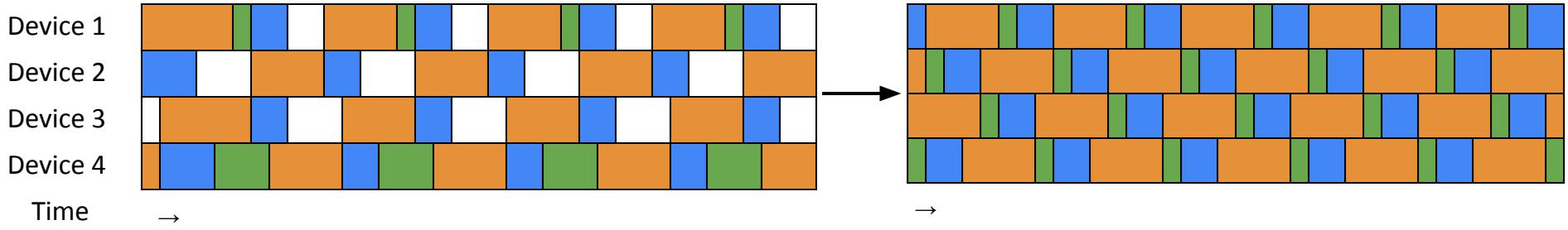
✓ Balances out the computation and memory of vocabulary layers

❓ Should not increase activation memory usage significantly

❓ Should not introduce new (recurring) pipeline bubbles

Vocabulary Parallelism: Overview

- Partition the vocabulary layers, distribute them equally across all pipeline devices.



✓ Balances out the computation and memory of vocabulary layers

❓ Should not increase activation memory usage significantly

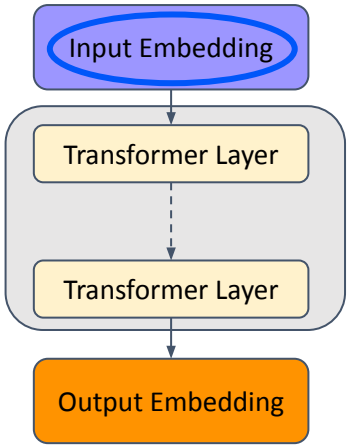
❓ Should not introduce new (recurring) pipeline bubbles

} Main challenge

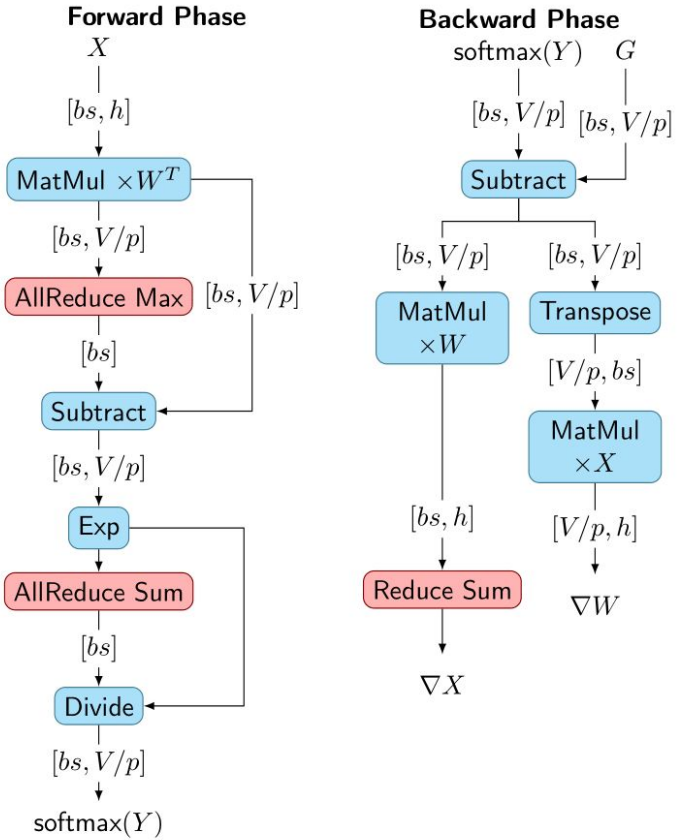
Vocabulary Parallelism: Overview

- Partitioning the **input embedding layer** is relatively trivial.
  - Can be piggybacked to the output embedding passes.

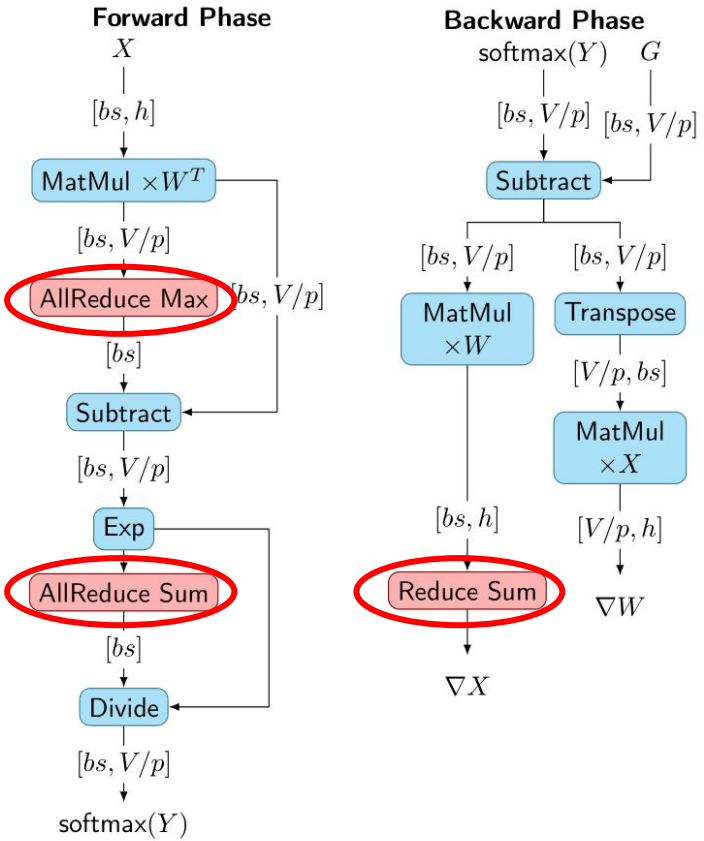
- We focus on the **output embedding layer** for the rest of this presentation.



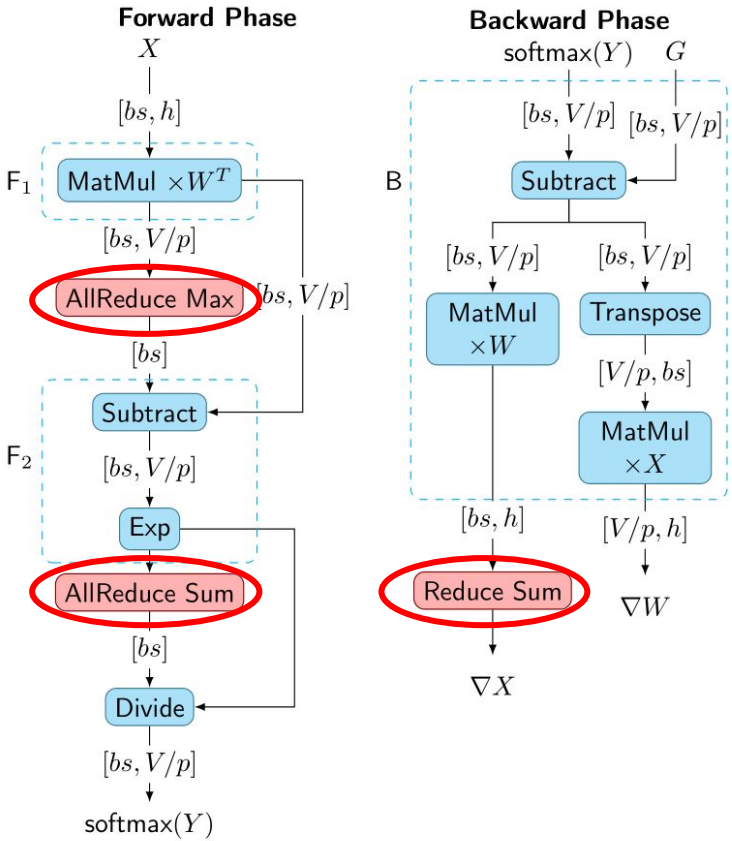
Vocabulary Parallelism: Overview



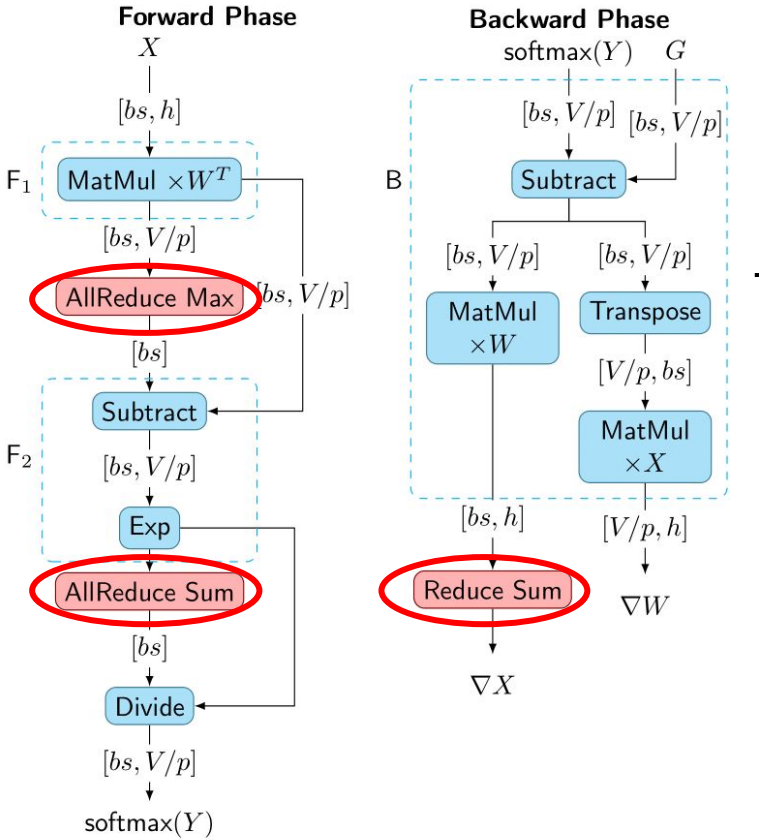
Vocabulary Parallelism: Overview

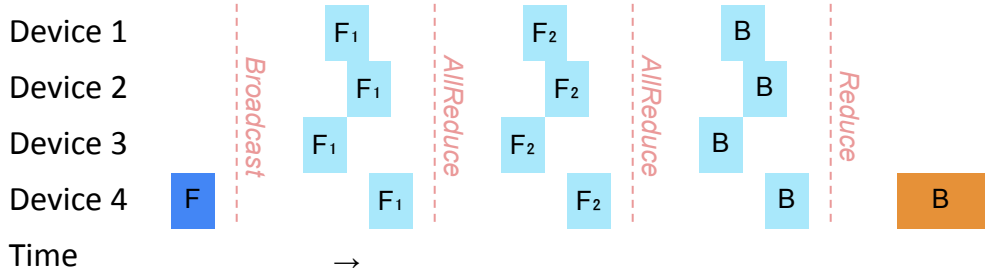


Vocabulary Parallelism: Overview

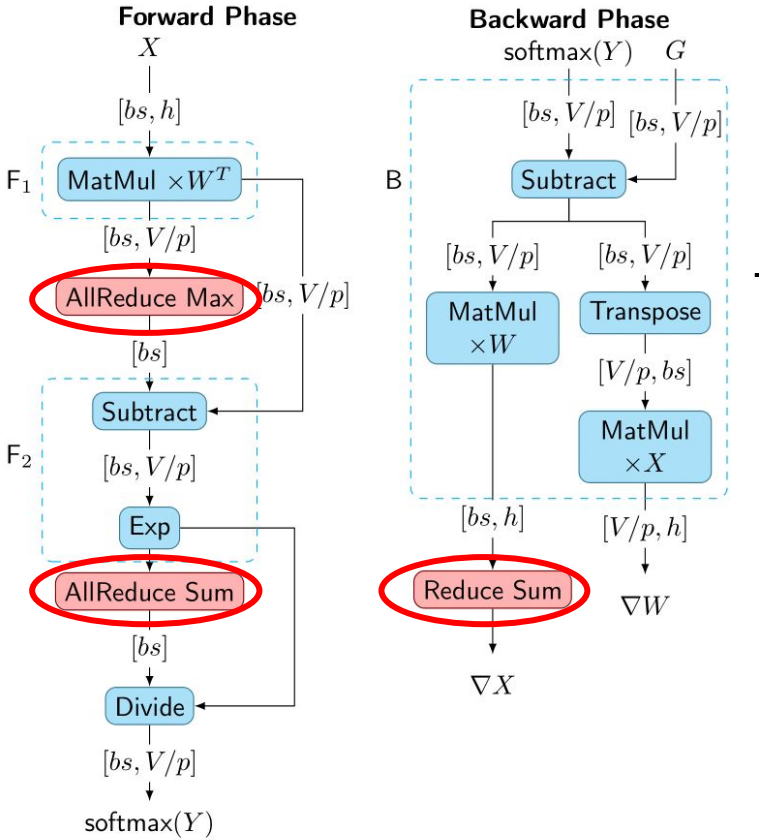


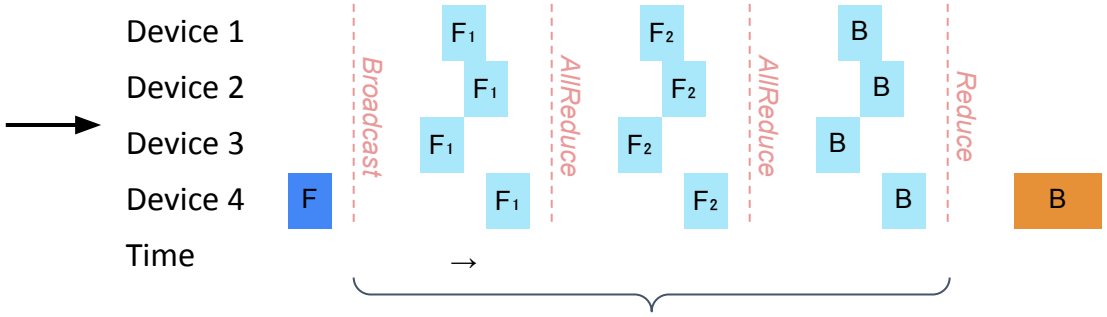
Vocabulary Parallelism: Overview





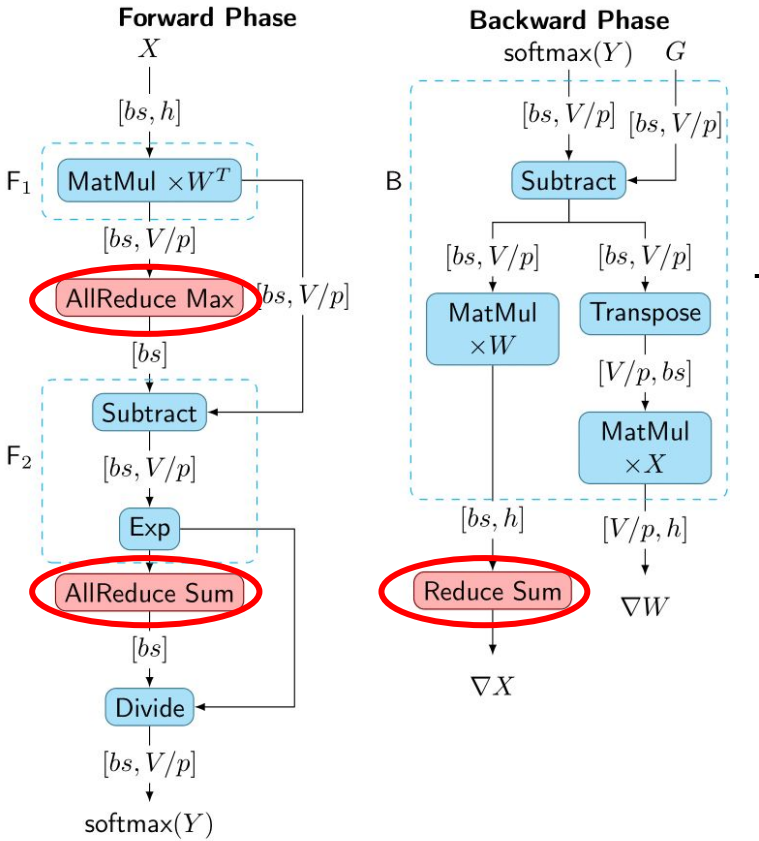
Vocabulary Parallelism: Overview

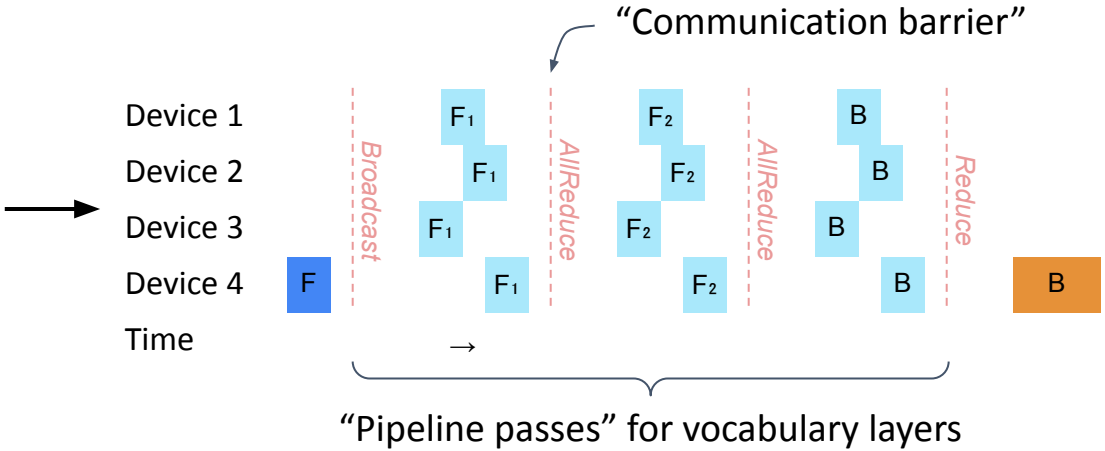




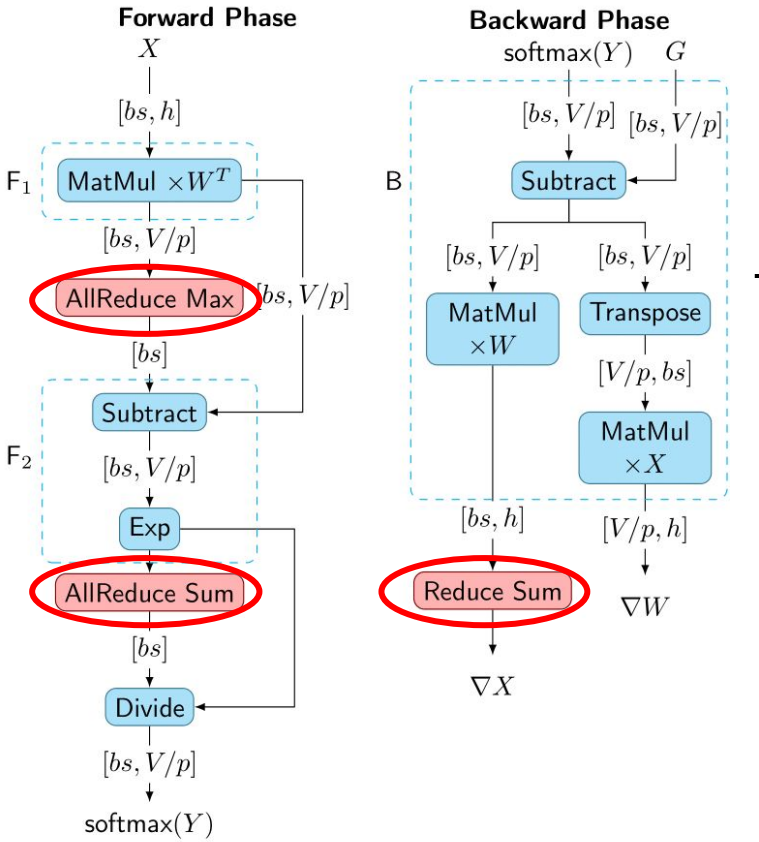
"Pipeline passes" for vocabulary layers

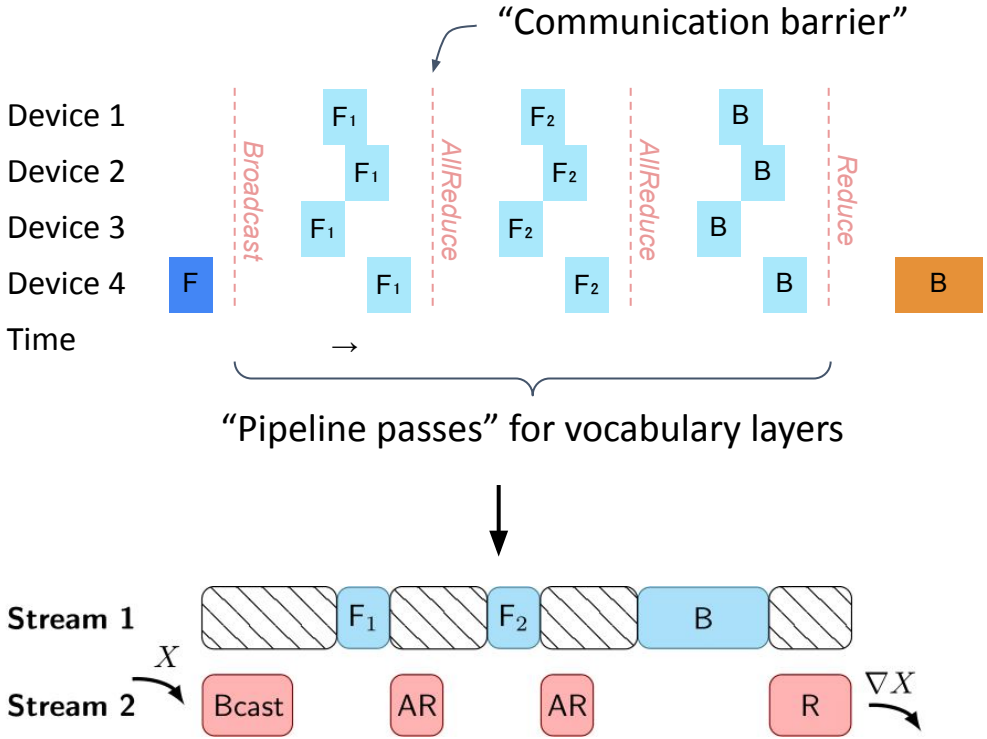
Vocabulary Parallelism: Overview





Vocabulary Parallelism: Overview



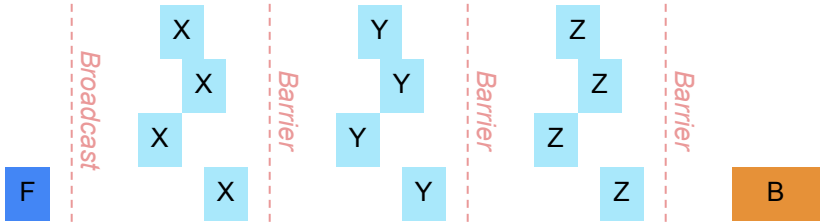


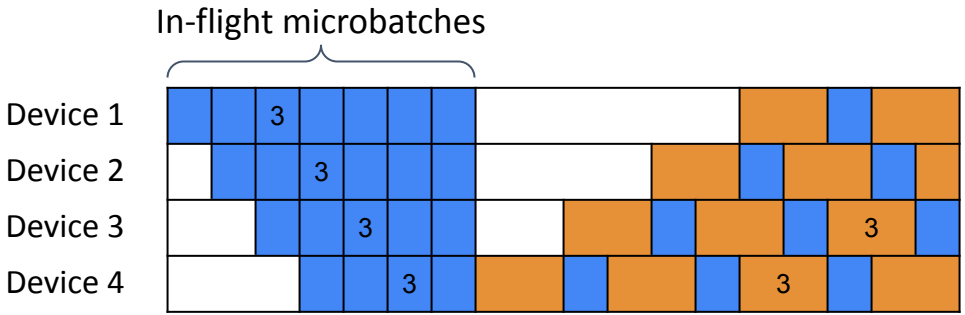
Vocabulary Parallelism: Overview

Number of communication **barriers** ⇔ Activation **memory** usage

Vocabulary Parallelism: Overview

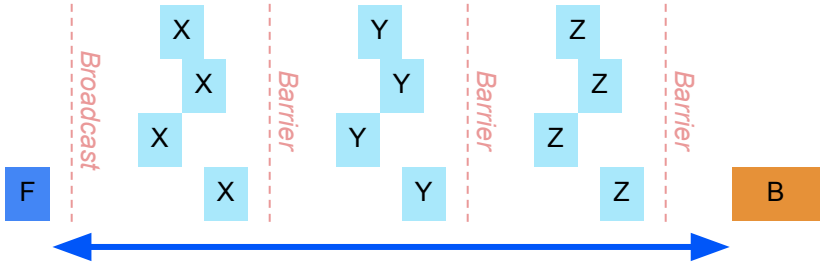
Number of communication **barriers** ⇔ Activation **memory** usage

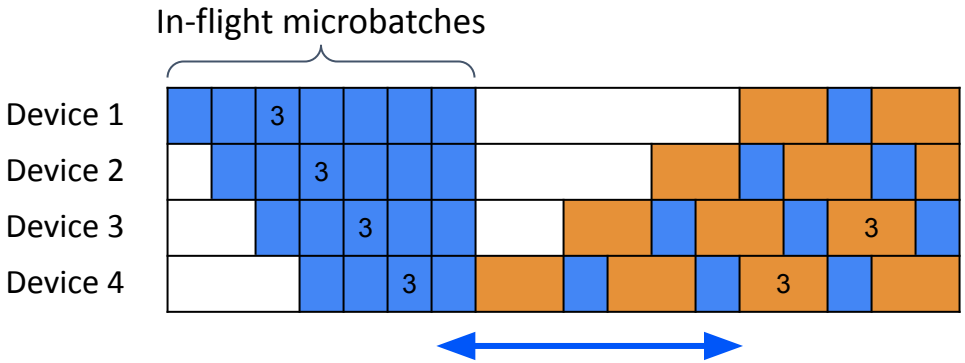




Vocabulary Parallelism: Overview

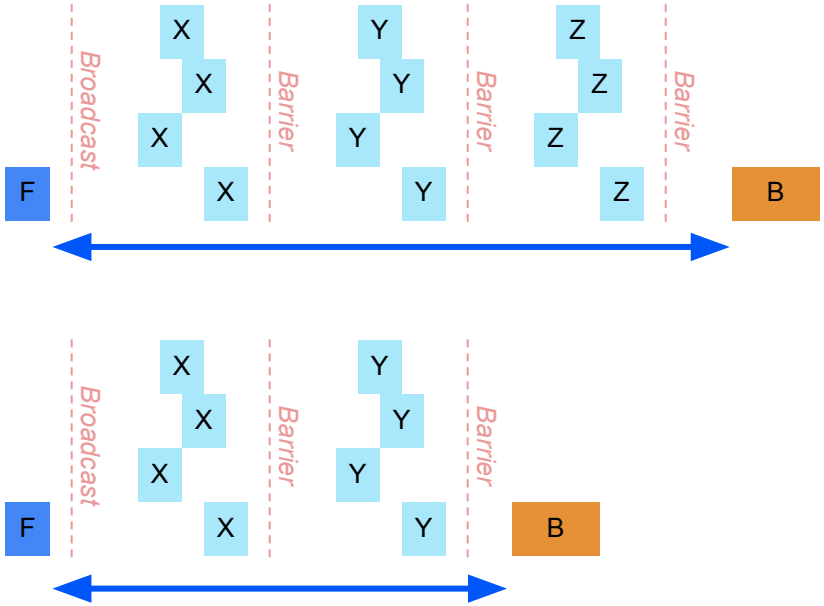
Number of communication **barriers** ⇔ Activation **memory** usage

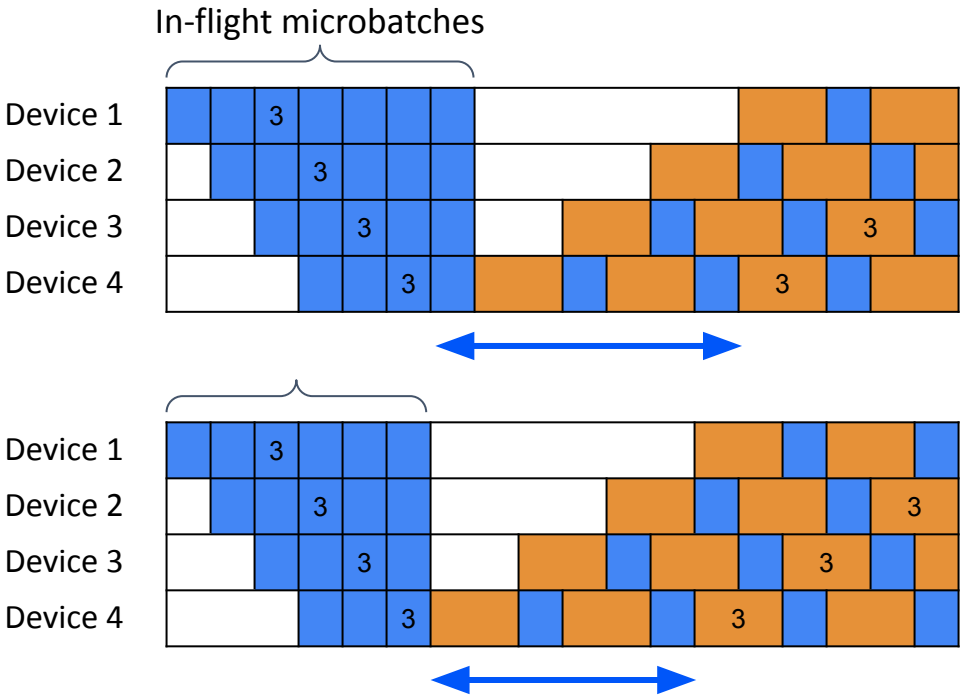




Vocabulary Parallelism: Overview

Number of communication **barriers** ⇔ Activation **memory** usage





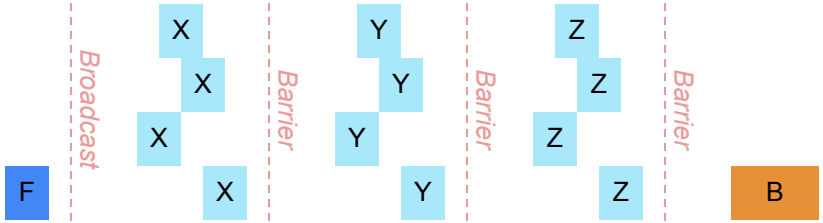
Vocabulary Parallelism: Overview

Our Goals:

Vocabulary Parallelism: Overview

Our Goals:

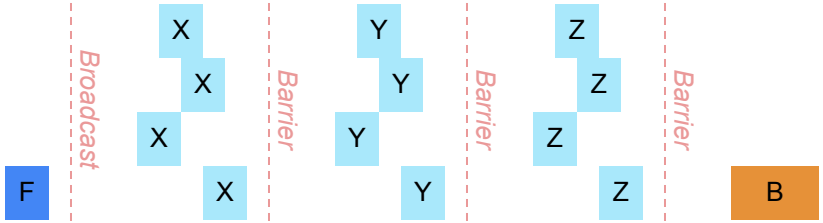
1. Reduce the number of **communication barriers**.



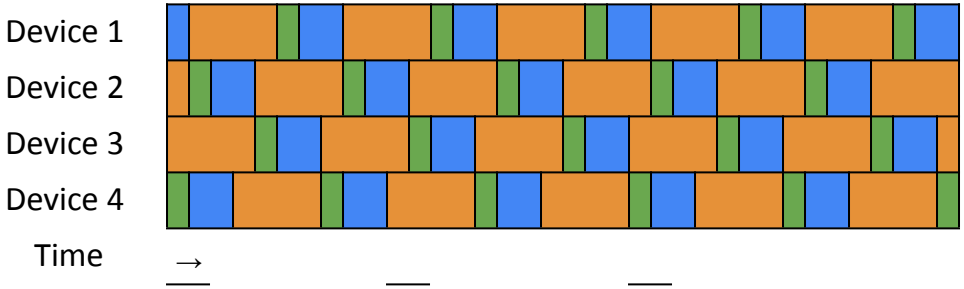
Vocabulary Parallelism: Overview

Our Goals:

1. Reduce the number of **communication barriers**.



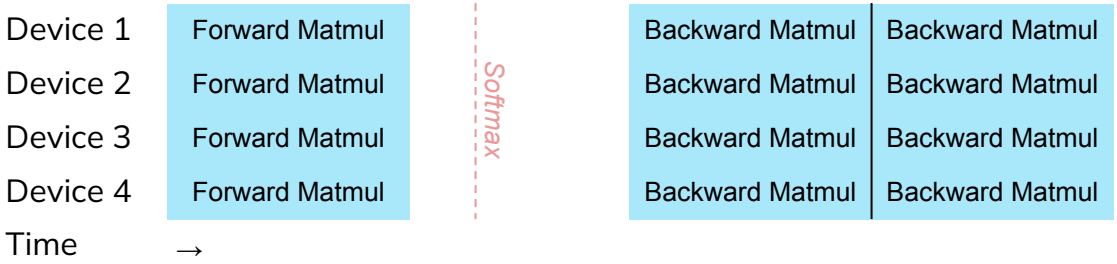
2. Integrate these vocabulary passes into existing pipeline **schedules**.



Vocabulary Parallelism: (1) Reducing Communication Barriers

Online-softmax (Milakov & Gimelshein, 2018):

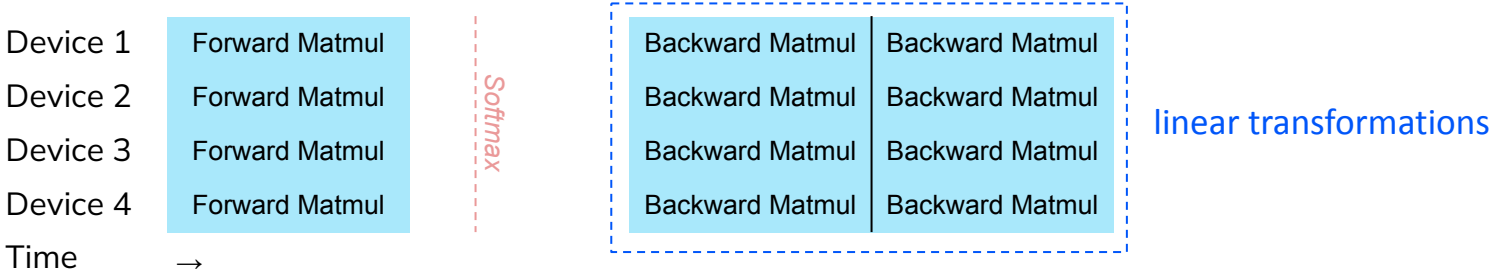
- Moves the synchronization point out of the computation critical path.



Vocabulary Parallelism: (1) Reducing Communication Barriers

Online-softmax (Milakov & Gimelshein, 2018):

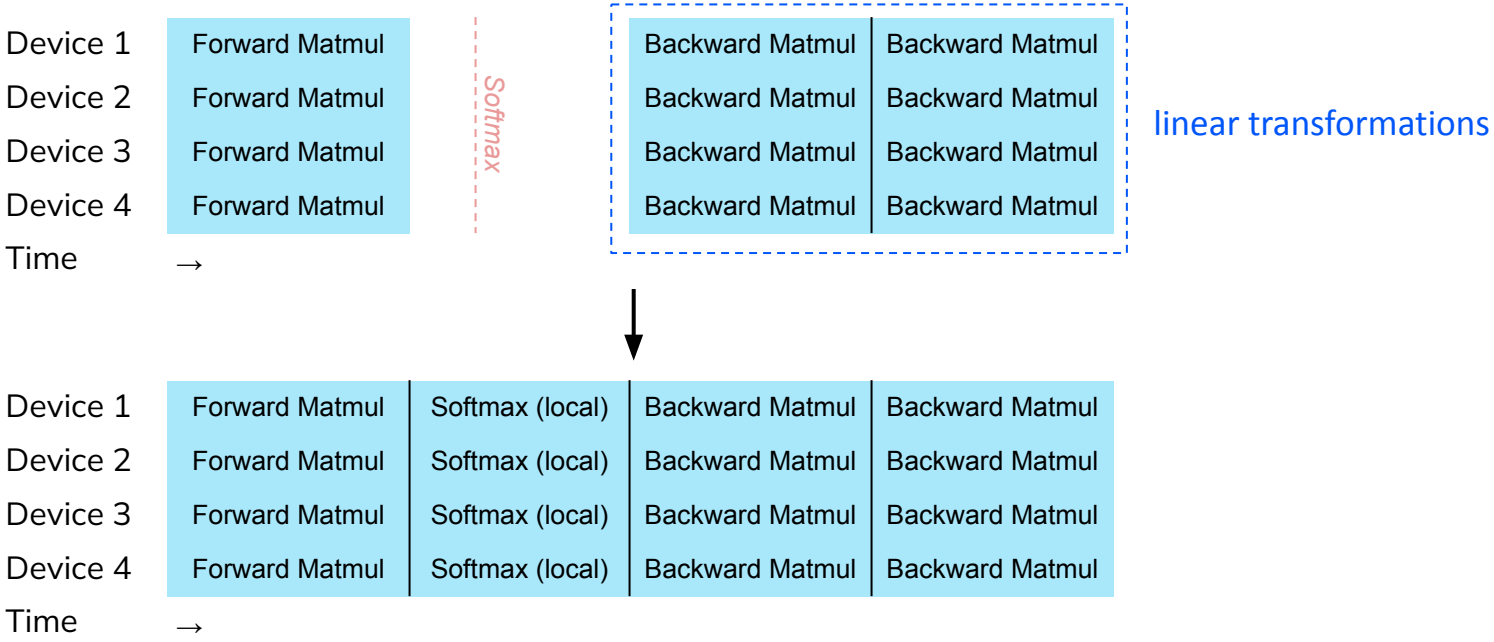
- Moves the synchronization point out of the computation critical path.



Vocabulary Parallelism: (1) Reducing Communication Barriers

Online-softmax (Milakov & Gimelshein, 2018):

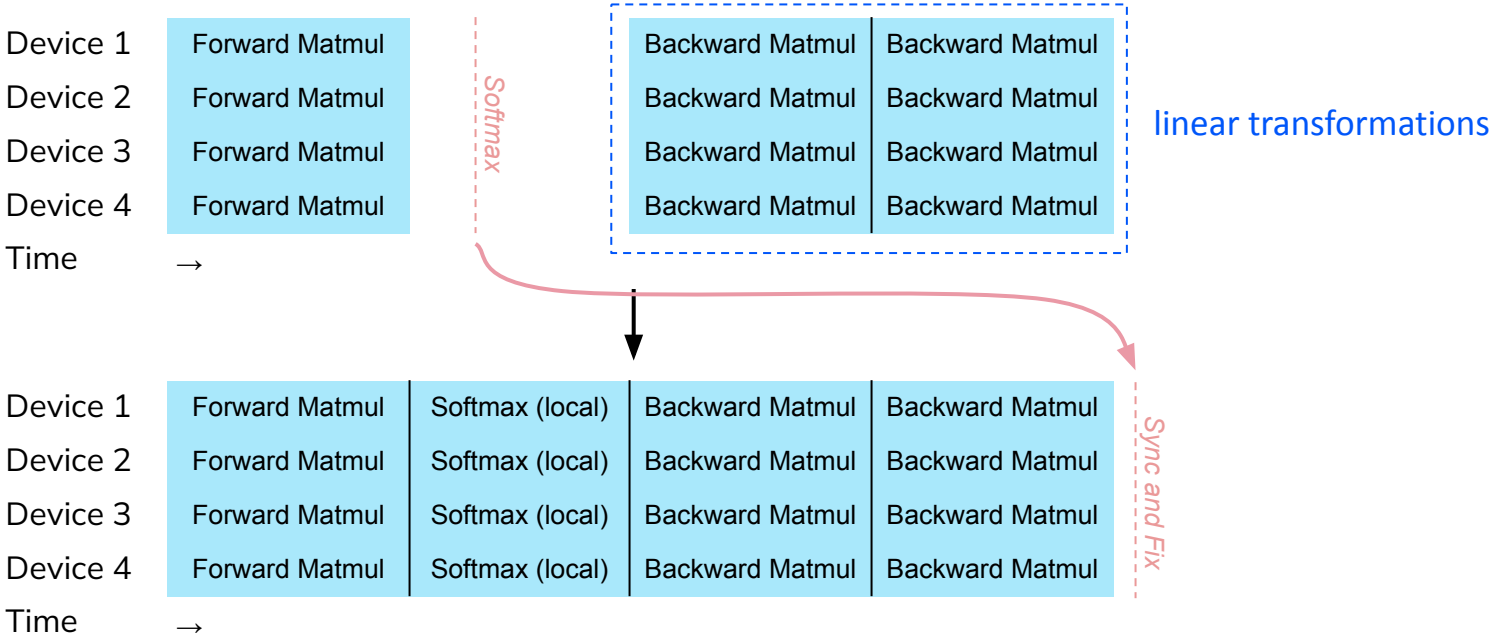
- Moves the synchronization point out of the computation critical path.



Vocabulary Parallelism: (1) Reducing Communication Barriers

Online-softmax (Milakov & Gimelshein, 2018):

- Moves the synchronization point out of the computation critical path.



Vocabulary Parallelism: (1) Reducing Communication Barriers

Using a similar approach, reduce the number of communication barriers **from 3 → 1**.

Algorithm 2 Output layer with 1 communication barrier

function forward_and_backward(W)

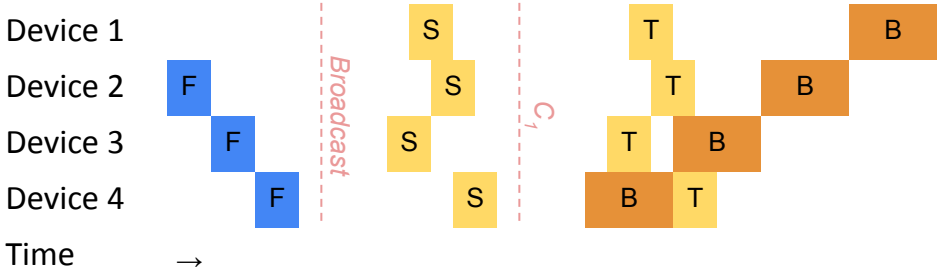
$X \leftarrow$ Receive Broadcast ◁ C_0

$Y \leftarrow XW^T$
$m'_i \leftarrow \max_{k=1}^{V/p} Y_{ik}$
$\text{sum}'_i \leftarrow \sum_{k=1}^{V/p} e^{Y_{ik} - m'_i}$
$\text{softmax}'(Y_{ij}) \leftarrow \dfrac{e^{Y_{ij} - m'_i}}{\text{sum}'_i}$ ◁ S
$A \leftarrow \text{softmax}'(Y)W$
$B \leftarrow GW$

$m_i \leftarrow$ AllReduce m'_i
$\text{sum}'_i \leftarrow \text{sum}'_i \times e^{m'_i - m_i}$
$\text{sum}_i \leftarrow$ AllReduce sum'_i
$\nabla X \leftarrow$ Reduce $A \times \frac{\text{sum}'_i}{\text{sum}_i} - B$ ◁ C_1

$\text{softmax}(Y) \leftarrow \text{softmax}'(Y) \times \dfrac{\text{sum}'_i}{\text{sum}_i}$
$\nabla W \leftarrow (\text{softmax}(Y) - G)^T X$ ◁ T

end function



Vocabulary Parallelism: (1) Reducing Communication Barriers

Using a similar approach, reduce the number of communication barriers **from 3 → 1**.

Algorithm 2 Output layer with 1 communication barrier

function forward_and_backward(W)

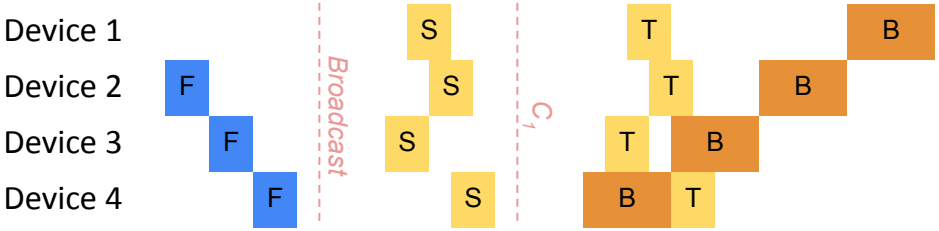
$X \leftarrow$ Receive Broadcast ◁ C_0

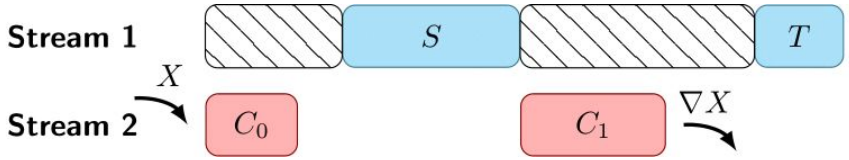
$Y \leftarrow XW^T$
$m'_i \leftarrow \max_{k=1}^{V/p} Y_{ik}$
$\text{sum}'_i \leftarrow \sum_{k=1}^{V/p} e^{Y_{ik} - m'_i}$
$\text{softmax}'(Y_{ij}) \leftarrow \dfrac{e^{Y_{ij} - m'_i}}{\text{sum}'_i}$ ◁ S
$A \leftarrow \text{softmax}'(Y)W$
$B \leftarrow GW$

$m_i \leftarrow$ AllReduce m'_i
$\text{sum}'_i \leftarrow \text{sum}'_i \times e^{m'_i - m_i}$
$\text{sum}_i \leftarrow$ AllReduce sum'_i
$\nabla X \leftarrow$ Reduce $A \times \frac{\text{sum}'_i}{\text{sum}_i} - B$ ◁ C_1

$\text{softmax}(Y) \leftarrow \text{softmax}'(Y) \times \dfrac{\text{sum}'_i}{\text{sum}_i}$
$\nabla W \leftarrow (\text{softmax}(Y) - G)^T X$ ◁ T

end function

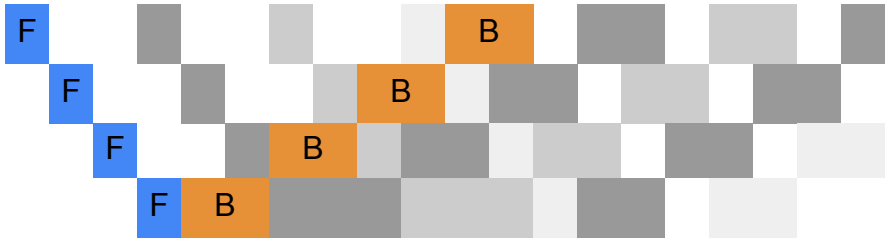




Vocabulary Parallelism: (2) Pipeline Scheduling

Building block framework (Qi et al., 2024):

- Most pipeline schedules can be represented in a **building block** (repeating pattern).



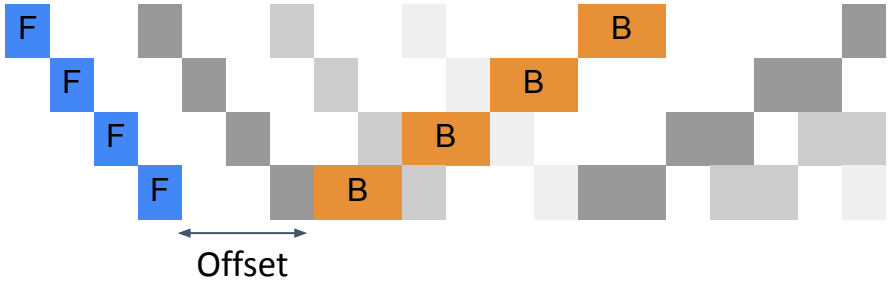
Vocabulary Parallelism: (2) Pipeline Scheduling

We modify the building block as follows:

Vocabulary Parallelism: (2) Pipeline Scheduling

We modify the building block as follows:

1

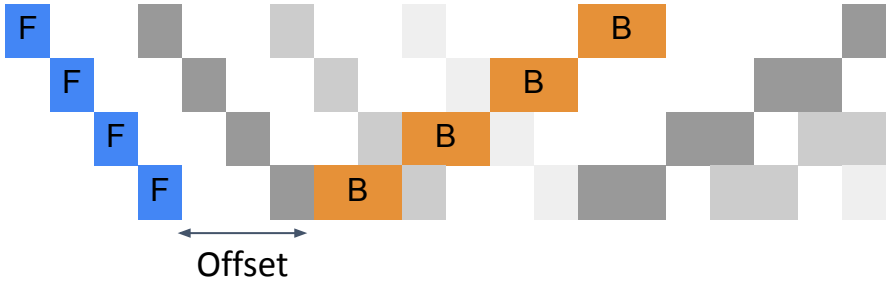


Increase the **offset** between F and B passes.

Vocabulary Parallelism: (2) Pipeline Scheduling

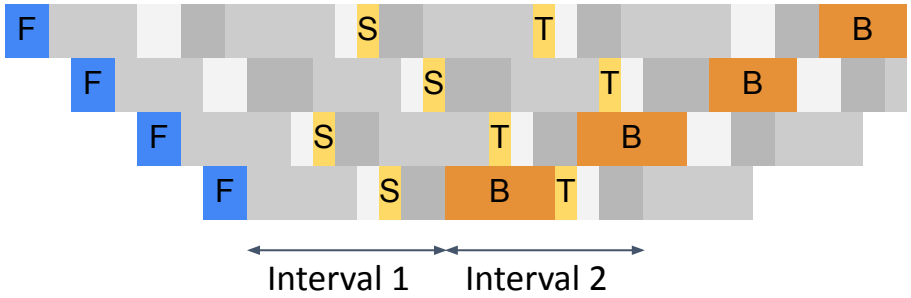
We modify the building block as follows:

1



Increase the **offset** between F and B passes.

2



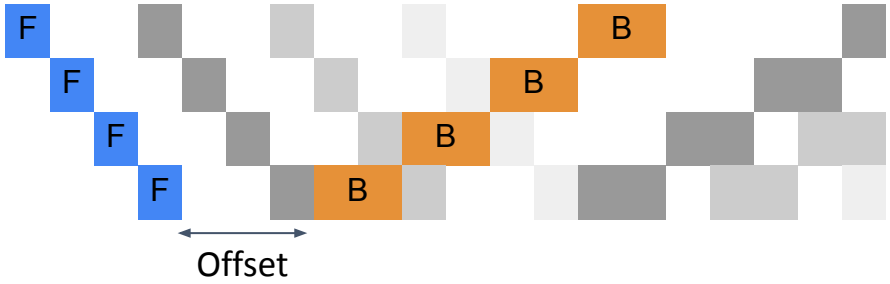
For each **interval**, insert an additional pipeline pass in each pipeline stage.

Vocabulary Parallelism: (2) Pipeline Scheduling

We modify the building block as follows:

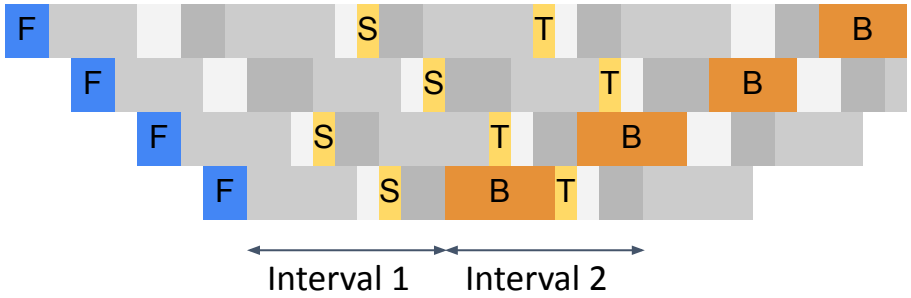
In-flight microbatches increases by 1 (a constant)

1



Increase the **offset** between F and B passes.

2



For each **interval**, insert an additional pipeline pass in each pipeline stage.

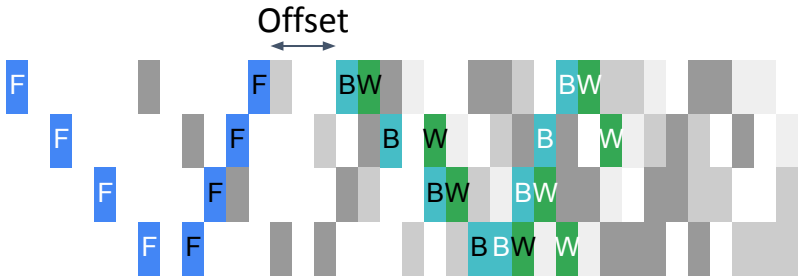
Vocabulary Parallelism: (2) Pipeline Scheduling

Generalizable to all pipeline schedules under the building block framework.

Vocabulary Parallelism: (2) Pipeline Scheduling

Generalizable to all pipeline schedules under the building block framework.

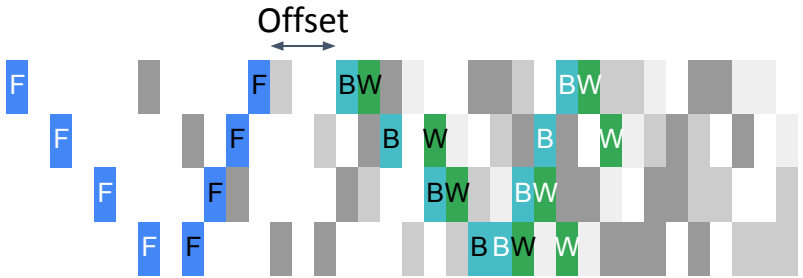
Example: V-Half (Qi et al., 2024):

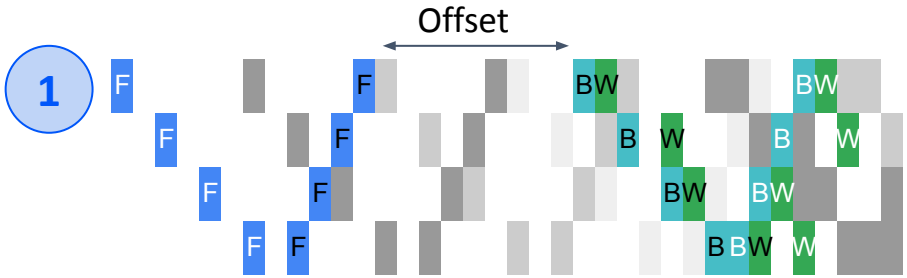


Vocabulary Parallelism: (2) Pipeline Scheduling

Generalizable to all pipeline schedules under the building block framework.

Example: V-Half (Qi et al., 2024):

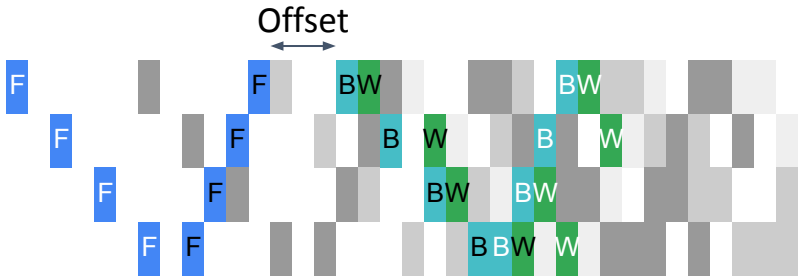




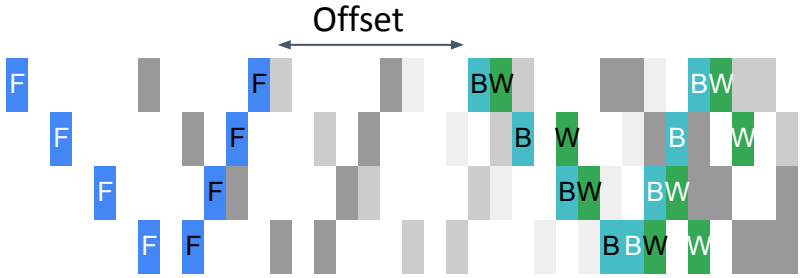
Vocabulary Parallelism: (2) Pipeline Scheduling

Generalizable to all pipeline schedules under the building block framework.

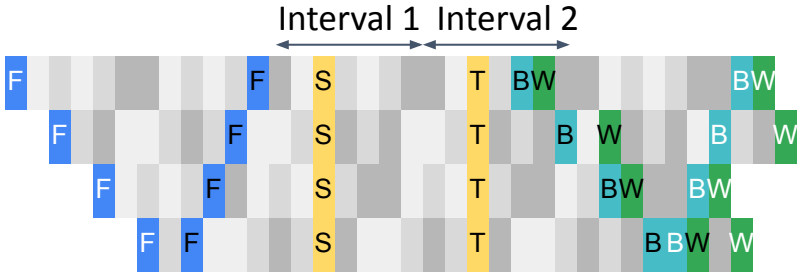
Example: V-Half (Qi et al., 2024):

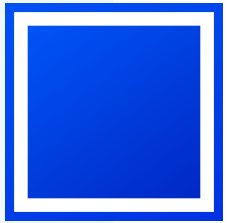


1



2





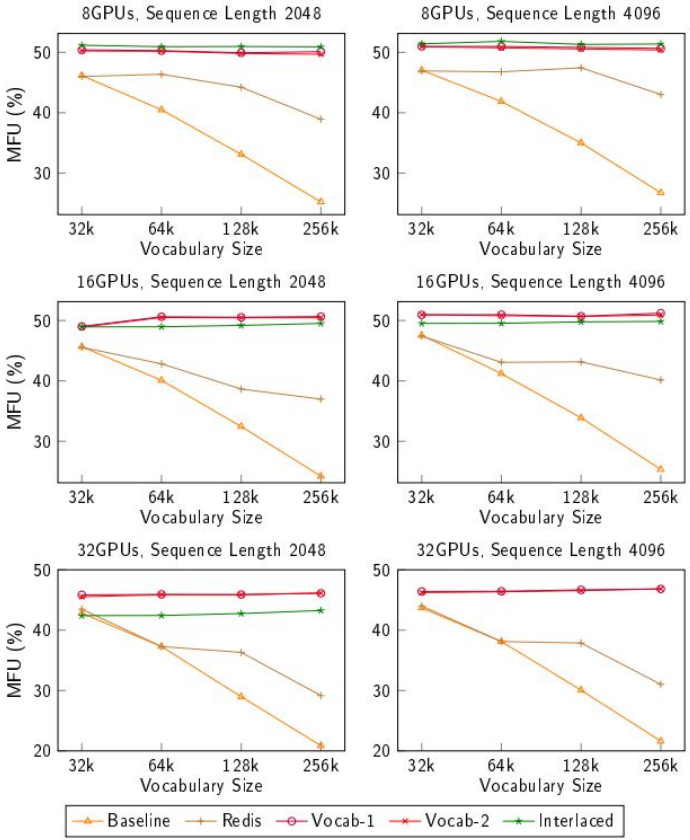
Evaluations

Evaluations: Comparison of Methods

Schedule	Method	Activation Memory	Bubble Rate	Balanced Compute	Balanced Memory
1F1B	–	d	B	❌	❌
	Redistribution	d	B	❌	❌
	Interlaced	1.5d	B	✓	Vocabulary only
	Vocab Parallel	d + 1	B	✓	Vocabulary only

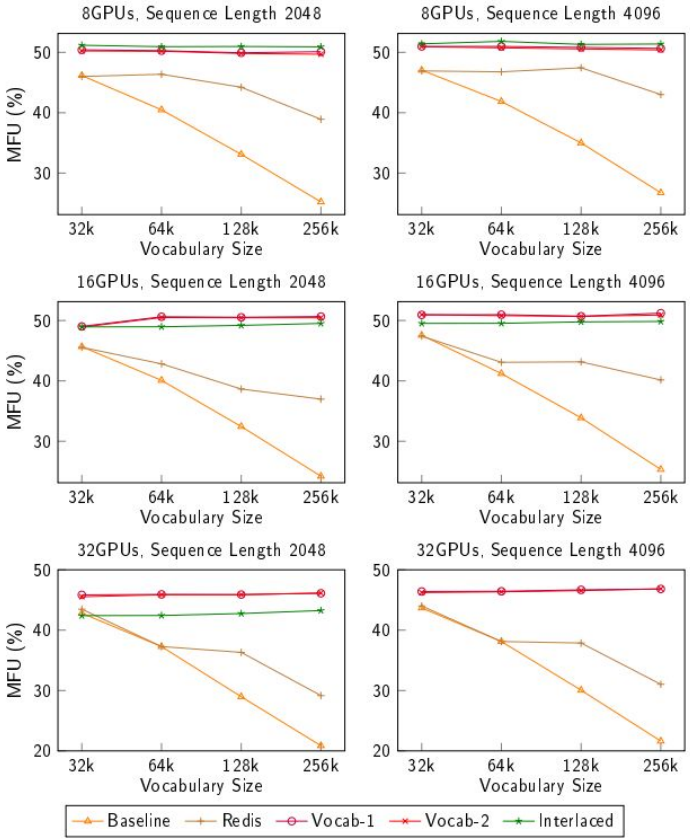
Evaluations: Comparison of Methods

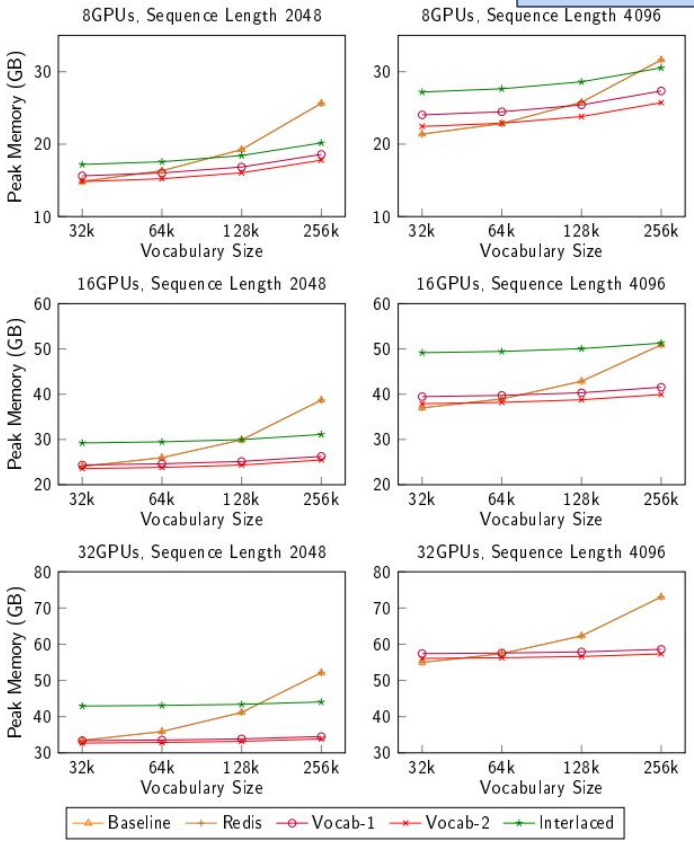
Throughput (MFU) 5 - 51% improvement



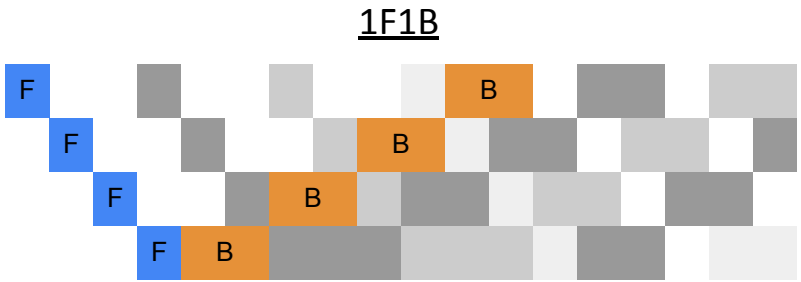
Evaluations: Comparison of Methods

Throughput (MFU) | 5 - 51% improvement | Memory Usage | improves on large vocabulary scenario



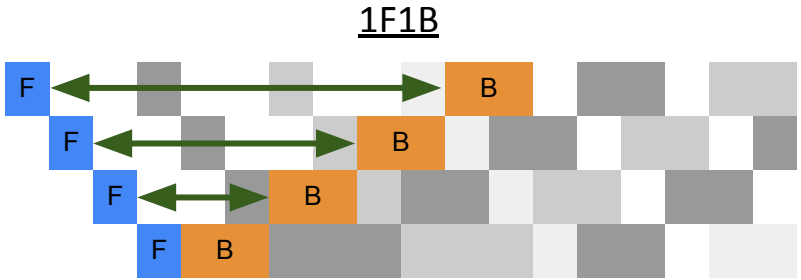


Evaluations: Memory-Balanced Schedule



Non-uniform activation memory usage

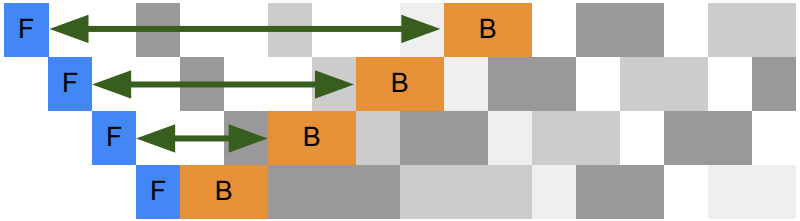
Evaluations: Memory-Balanced Schedule



Non-uniform activation memory usage

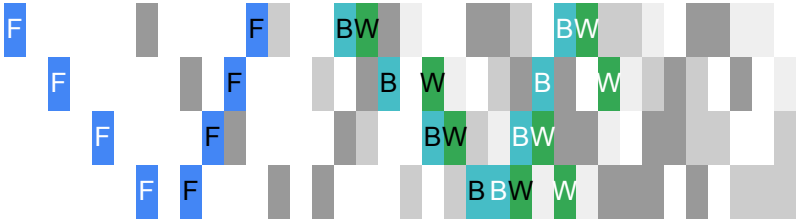
Evaluations: Memory-Balanced Schedule

1F1B



Non-uniform activation memory usage

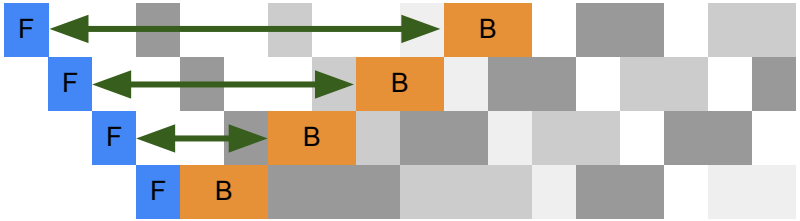
V-Half



Uniform activation memory usage

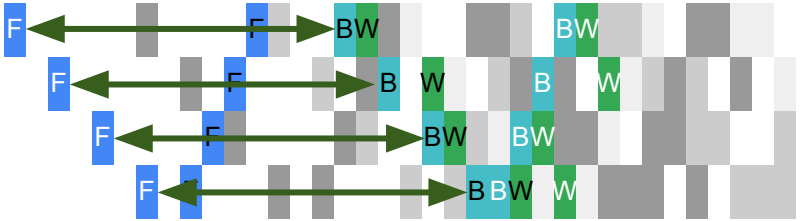
Evaluations: Memory-Balanced Schedule

1F1B



Non-uniform activation memory usage

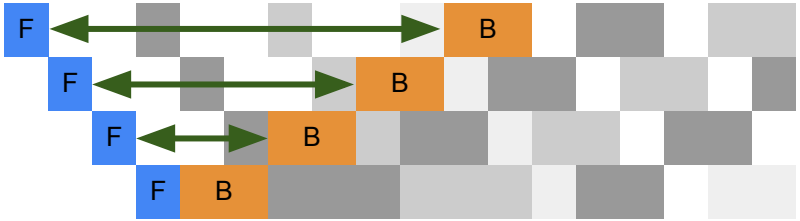
V-Half



Uniform activation memory usage

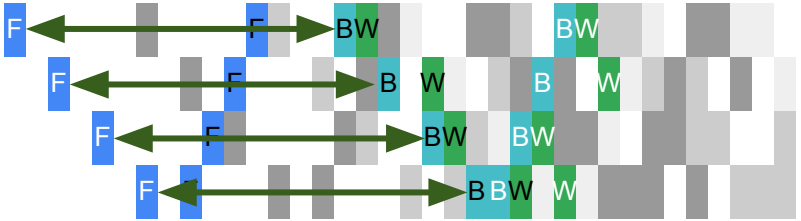
Evaluations: Memory-Balanced Schedule

1F1B



Non-uniform activation memory usage

V-Half

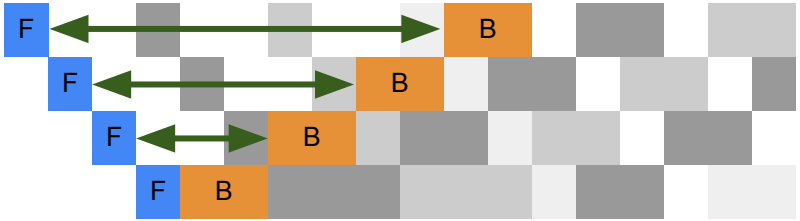


Uniform activation memory usage

Memory-balanced schedule + Vocabulary Parallelism

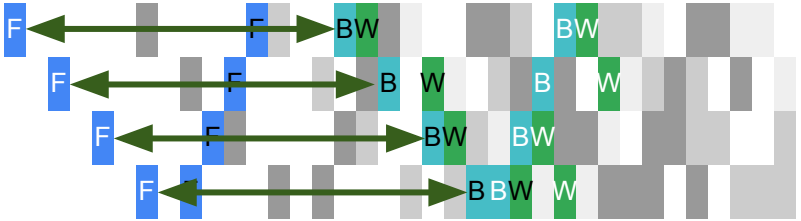
Evaluations: Memory-Balanced Schedule

1F1B



Non-uniform activation memory usage

V-Half

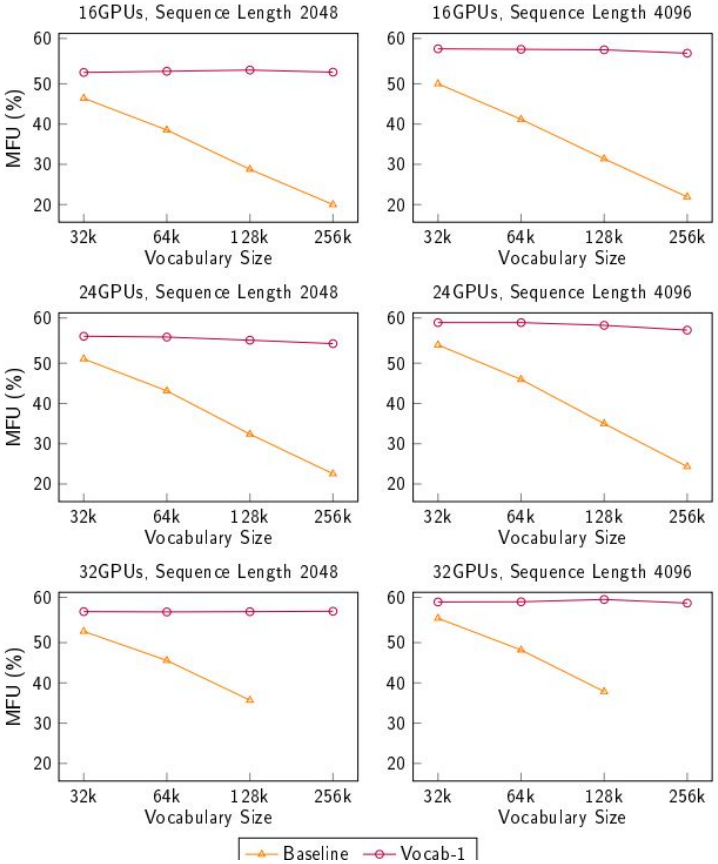


Uniform activation memory usage

Memory-balanced schedule + Vocabulary Parallelism ⇒ **Balanced memory usage**

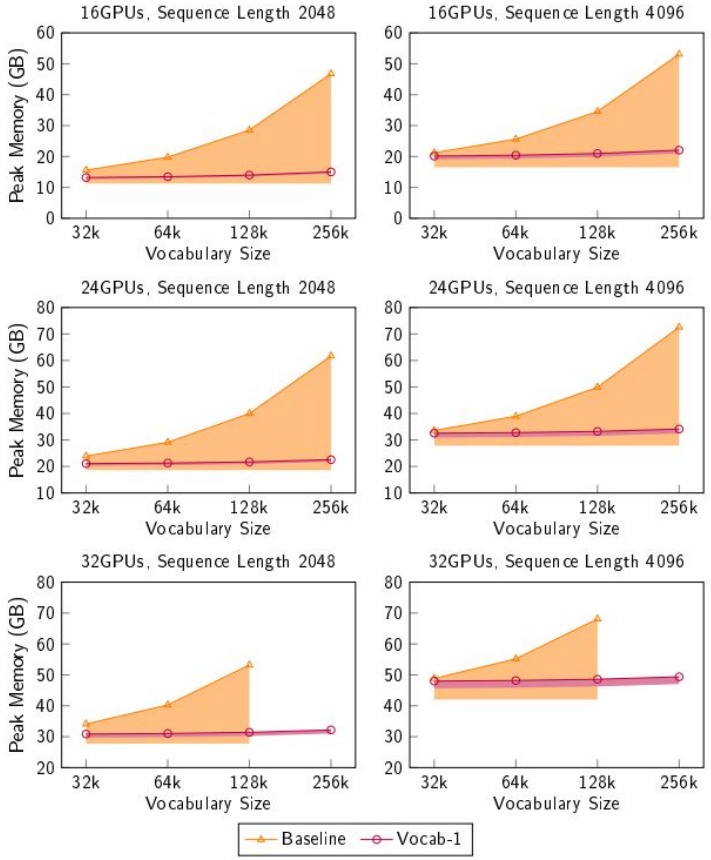
Evaluations: Memory-Balanced Schedule

Throughput (MFU)



Memory Usage

balanced



Evaluations: Summary

Schedule	Method	Activation Memory	Bubble Rate	Balanced Compute	Balanced Memory
1F1B	–	d	B	❌	❌
	Redistribution	d	B	❌	❌
	Interlaced	1.5d	B	✓	Vocabulary only
	Vocab Parallel	d + 1	B	✓	Vocabulary only
V-Half	–	≈ d / 2	≈ B / 2	❌	Transformer only
	Vocab Parallel	≈ d / 2 + 1	≈ B / 2	✓	✓

Evaluations: Scaling Overhead

Parallelizing the vocabulary layers comes with some computation overhead:

- Reduced model FLOPs utilization of individual GPU kernels
- Extra computation (e.g. vocabulary size-independent portions, sync and fix)

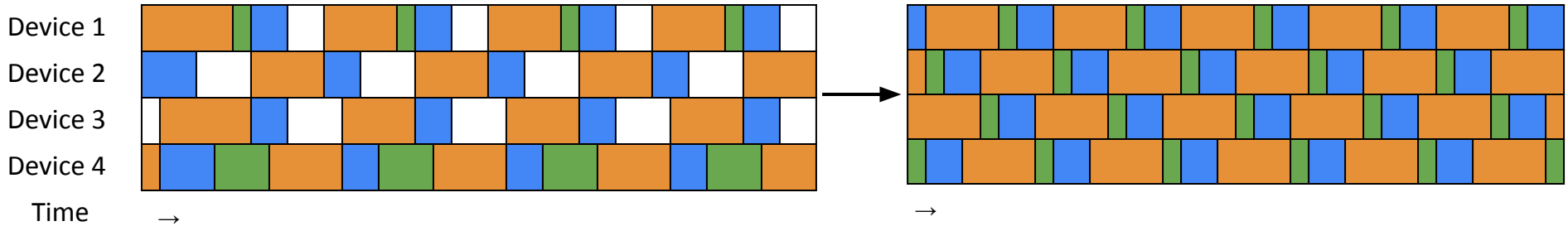
Still brings overall performance gains.

SEQ	LAYER	8GPU	16GPU	32GPU
2048	OUTPUT-VOCAB-1	91.29%	84.22%	80.59%
	OUTPUT-VOCAB-2	86.72%	79.84%	75.93%
	INPUT	39.99%	28.85%	15.18%
4096	OUTPUT-VOCAB-1	93.21%	88.02%	85.24%
	OUTPUT-VOCAB-2	88.36%	83.42%	79.66%
	INPUT	27.69%	15.52%	8.35%

Scaling factor of vocabulary layer computation relative to linear scaling on sequence lengths 2048 and 4096.

Conclusion

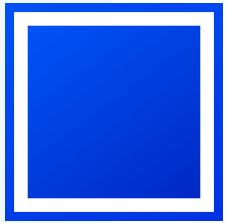
- Proposed **Vocabulary Parallelism** on top of **Pipeline Parallelism** to balance out computation and memory across pipeline devices.



Paper

Code

Playground



Thanks!