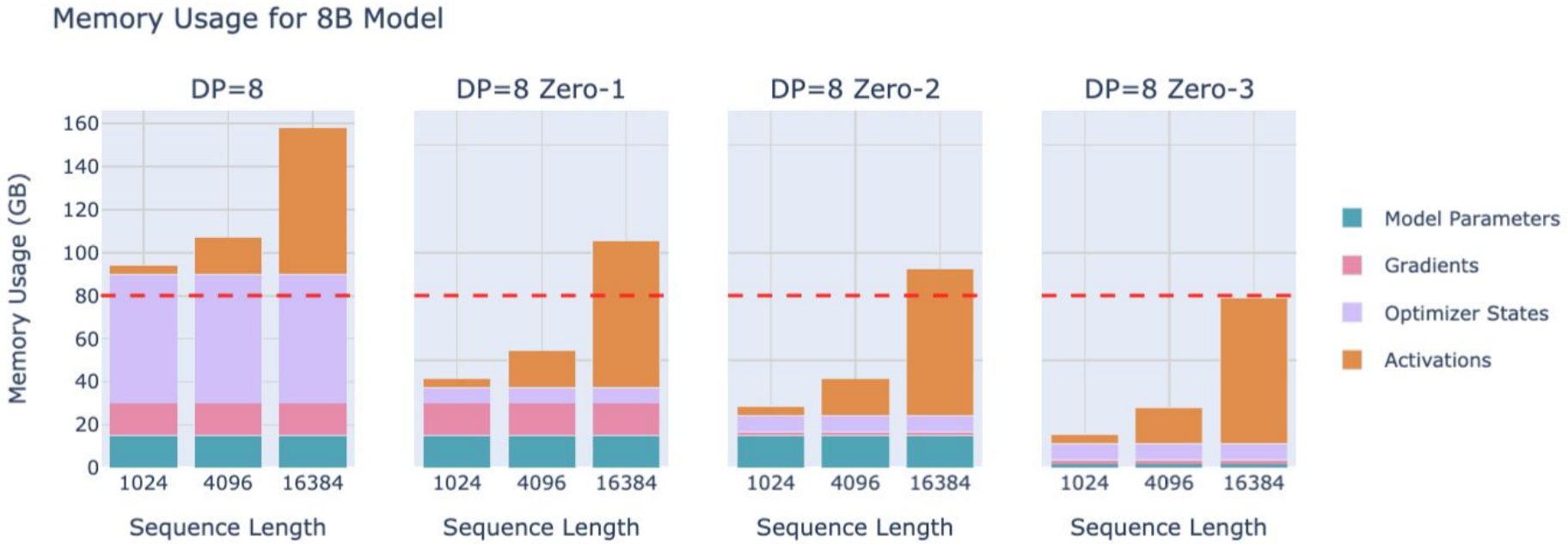
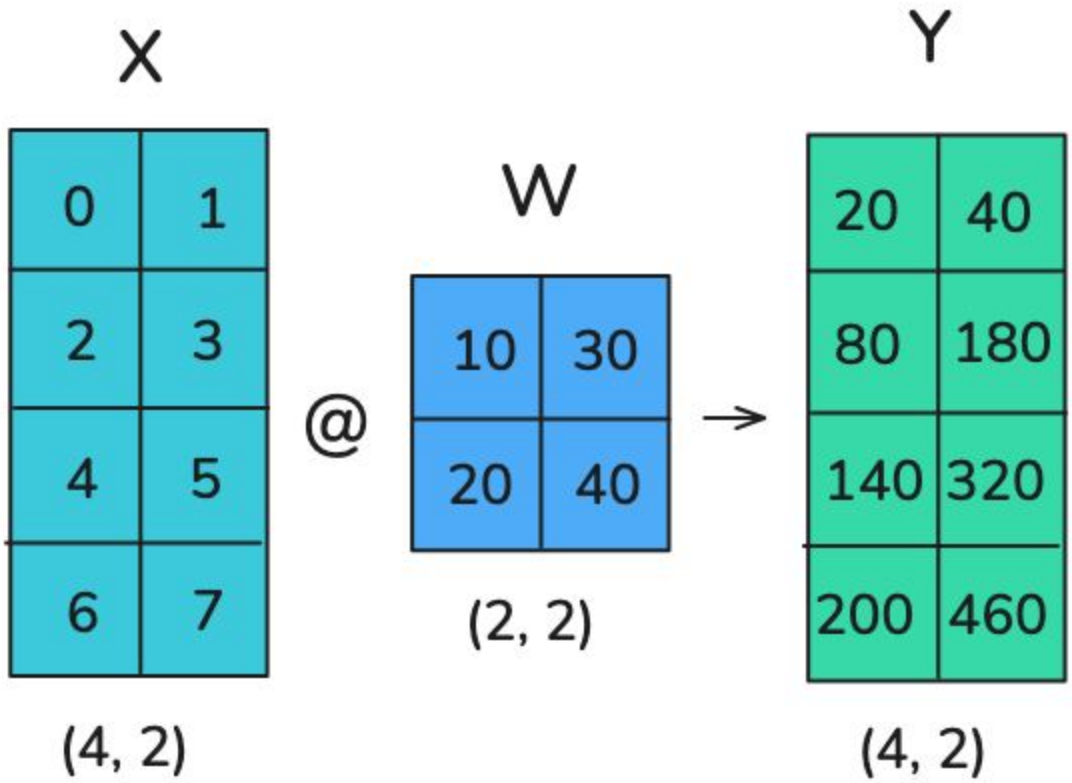
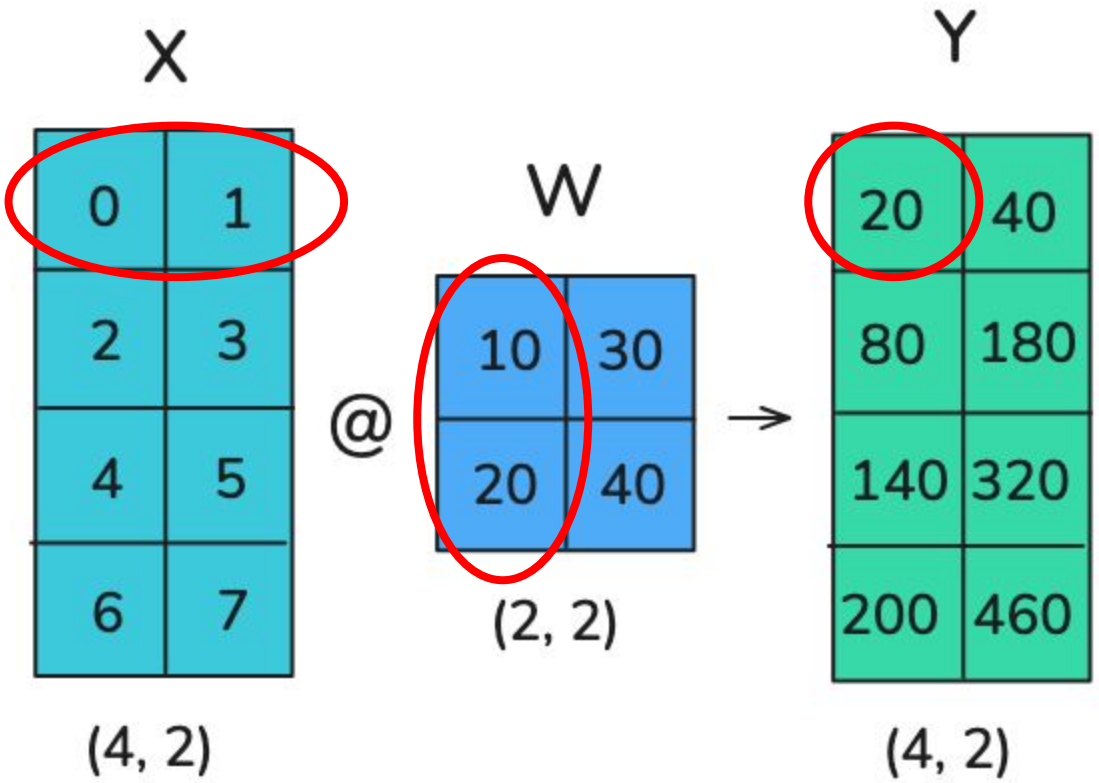
Tensor Parallelism

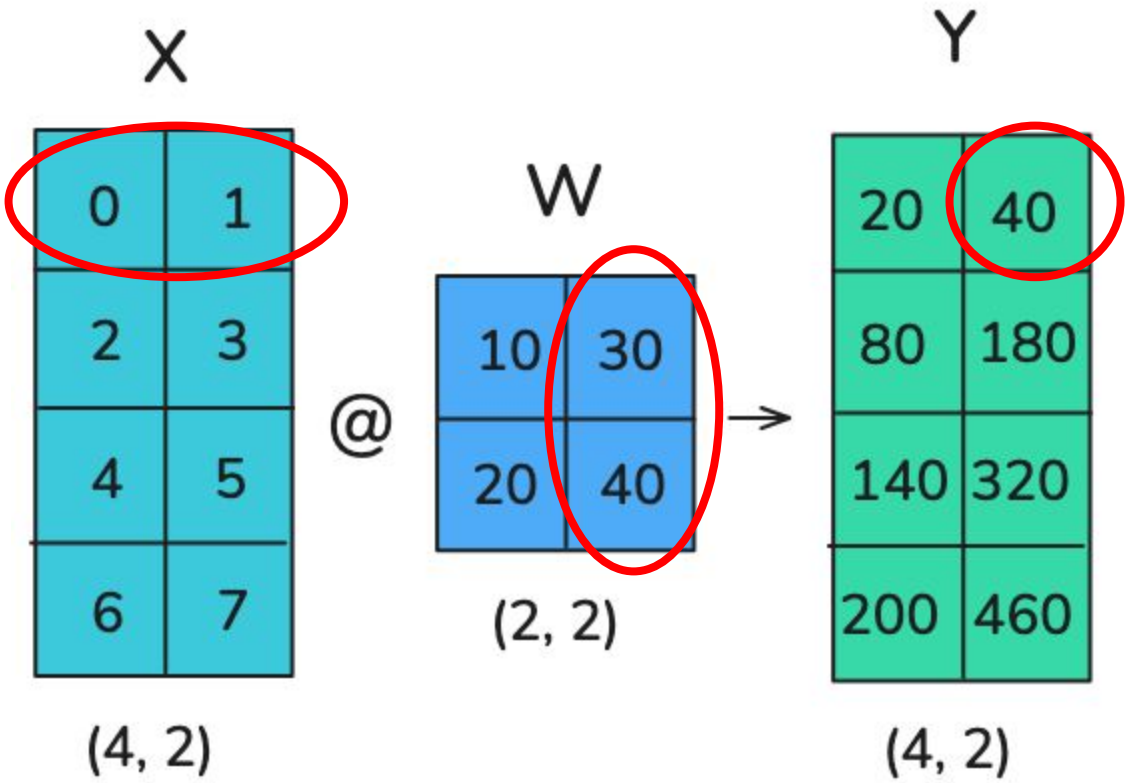
made with ❤️ for “Little ML book club”

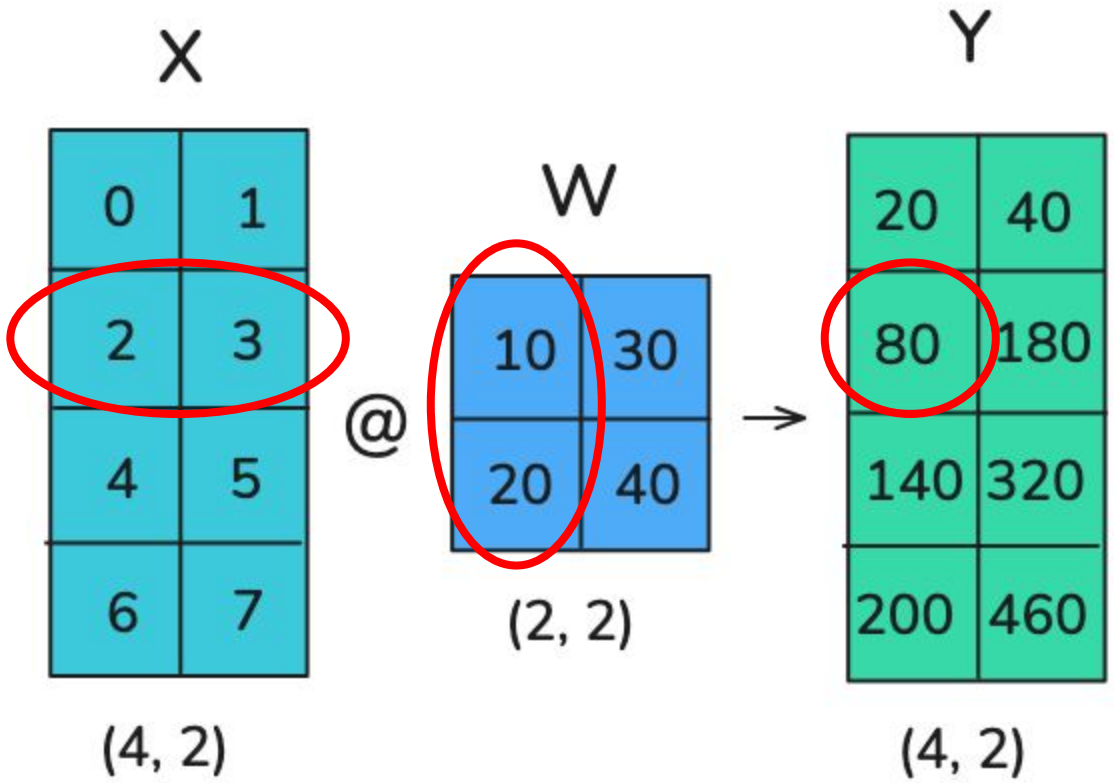

Memory Usage for 8B Model
DP=8
DP=8 Zero-1
DP=8 Zero-2
DP=8 Zero-3
Memory Usage (GB)
0
20
40
60
80
100
120
140
160
1024
4096
16384
Sequence Length
Model Parameters
Gradients
Optimizer States
Activations

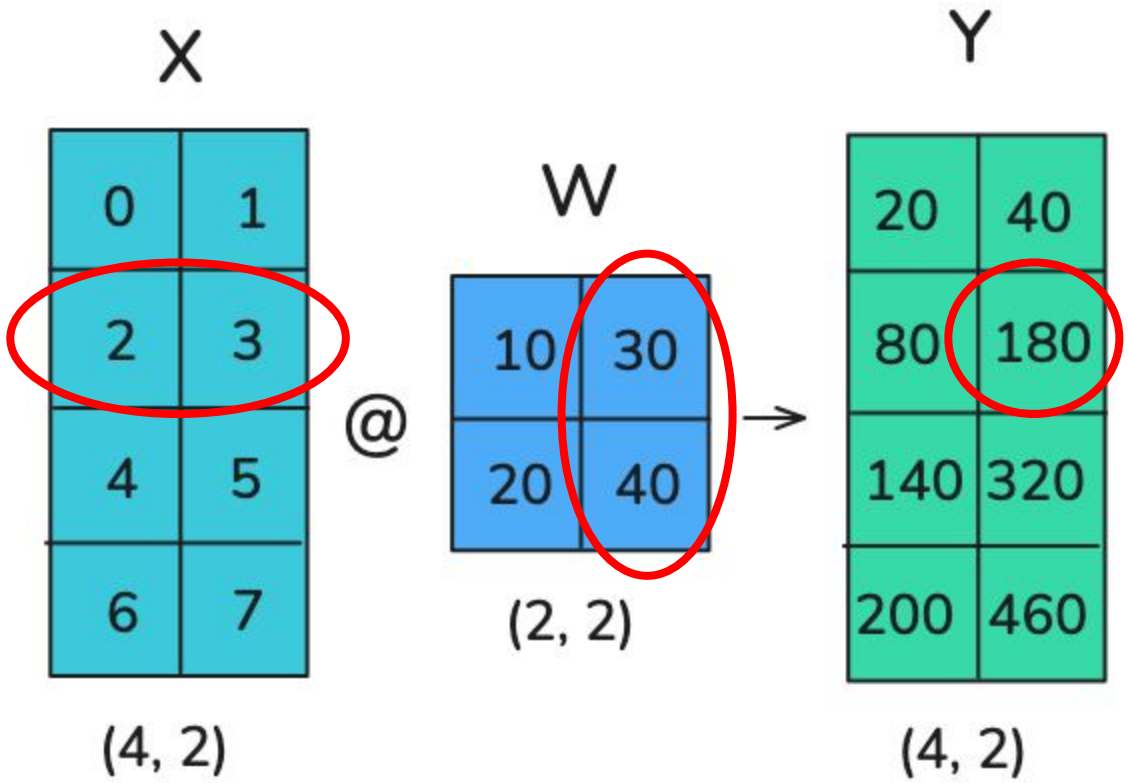
mom,
but I want to play
with veeeeery long sequences
and
veeeeery large models

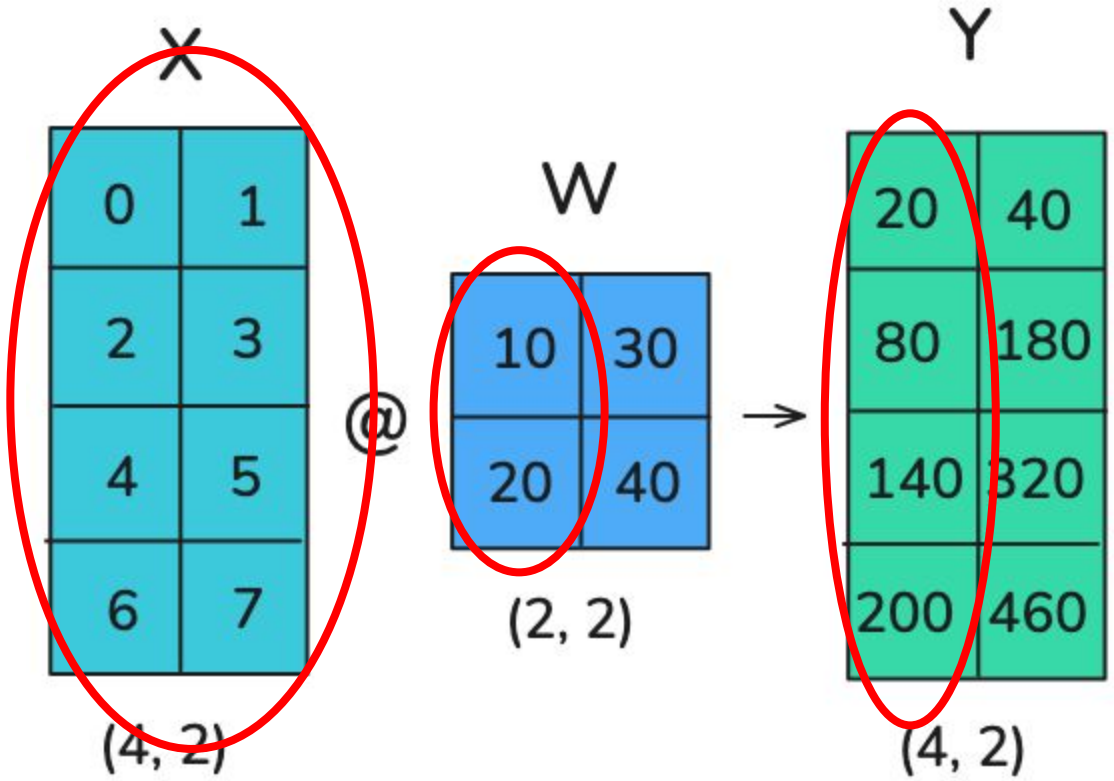

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

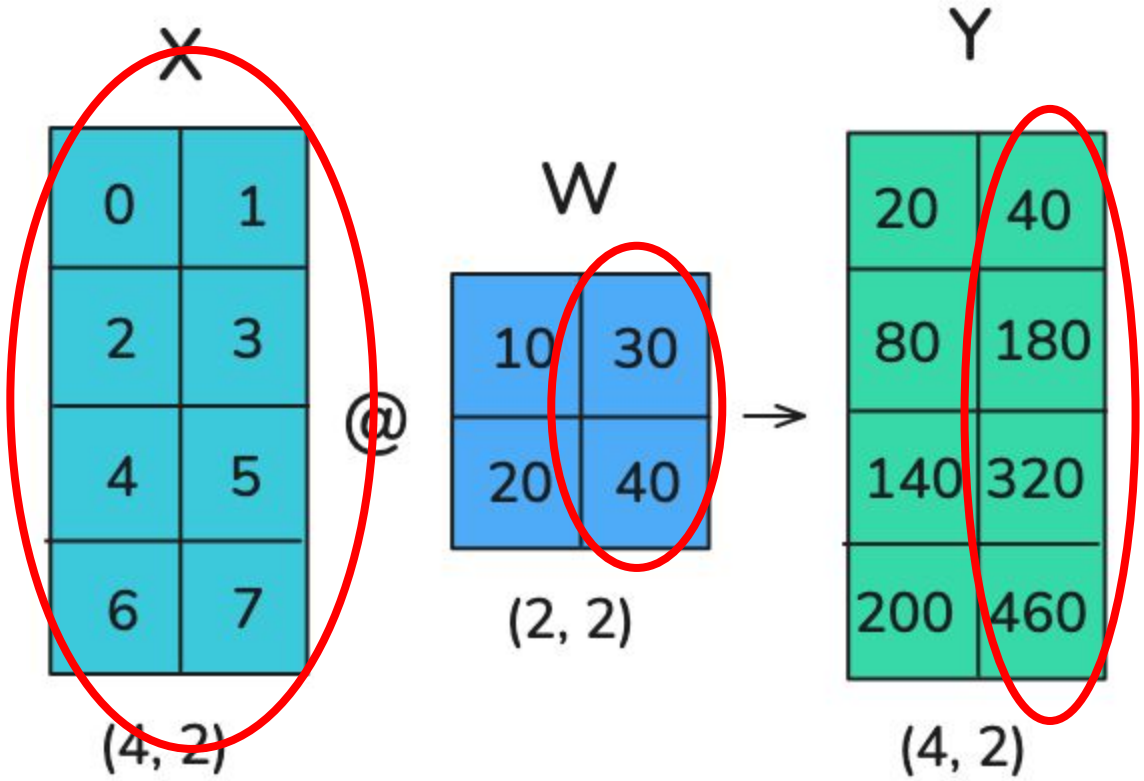

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

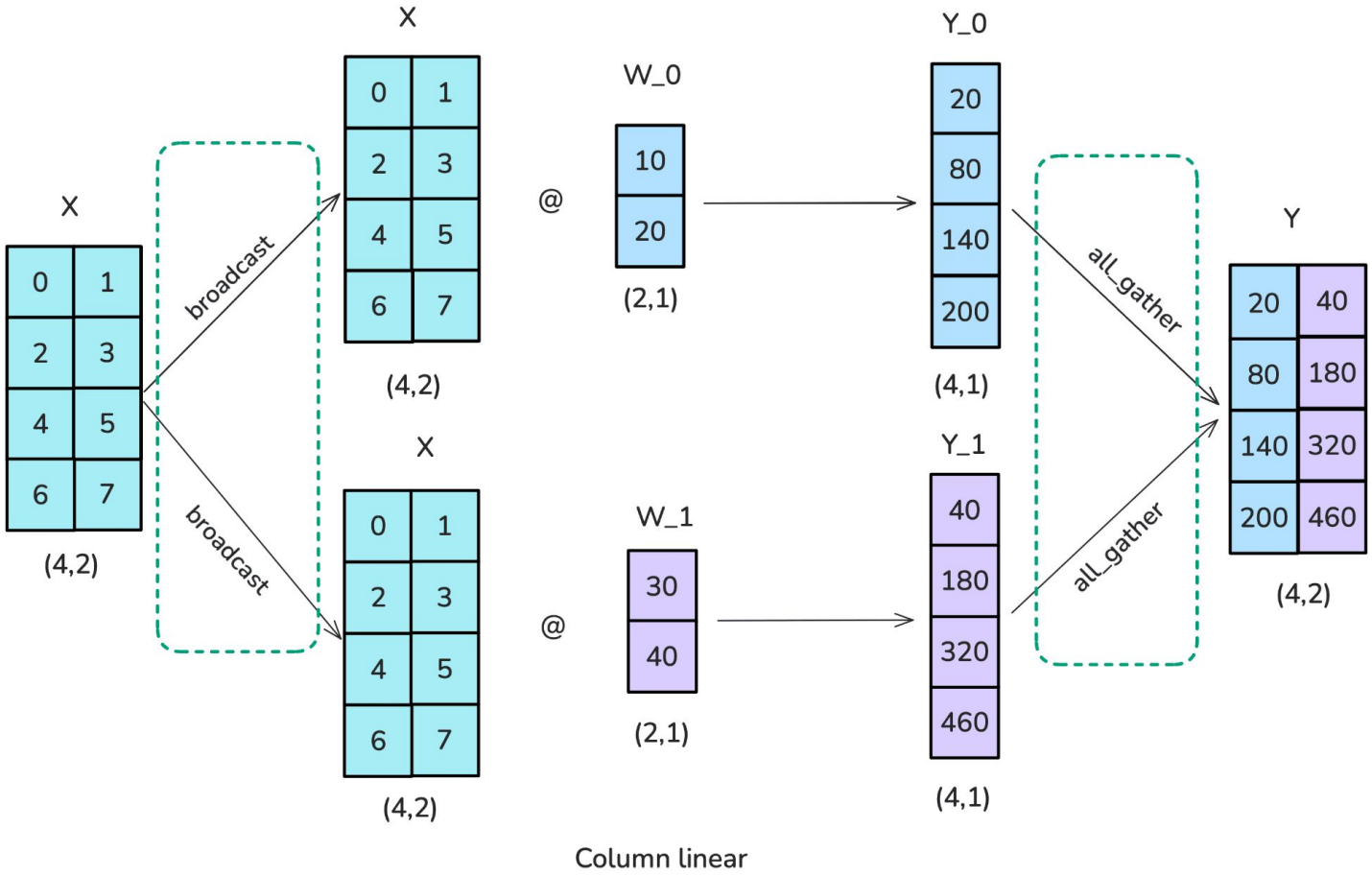

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)


X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

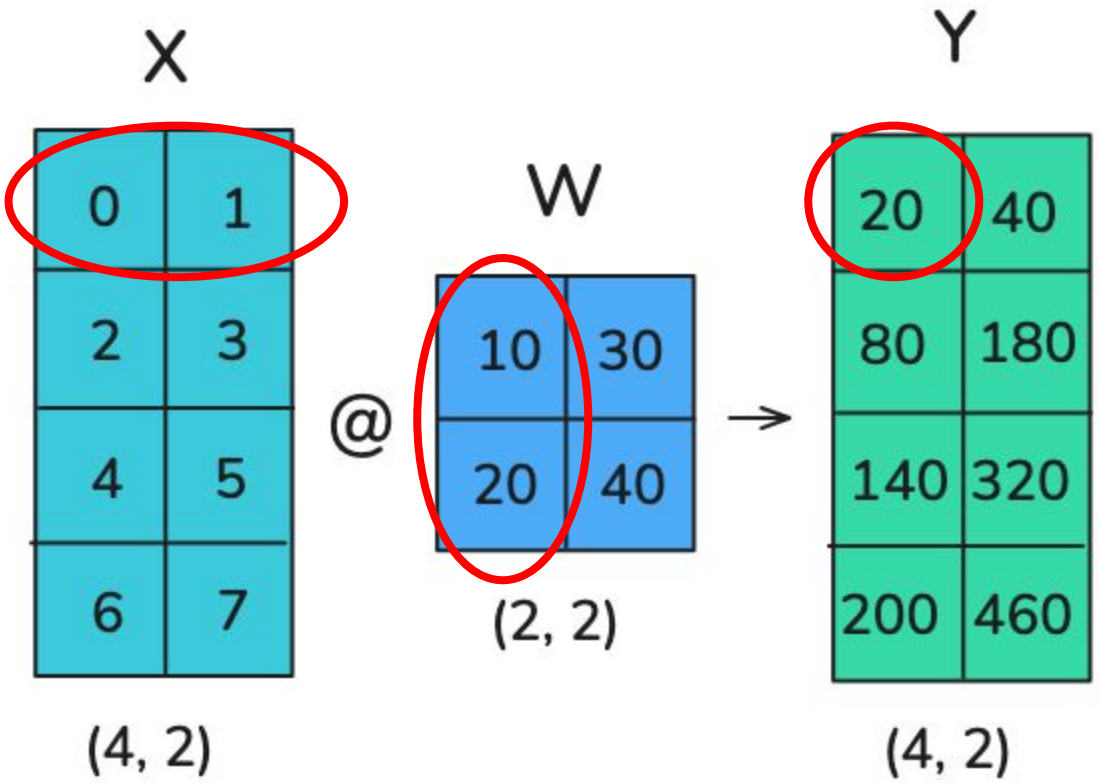

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

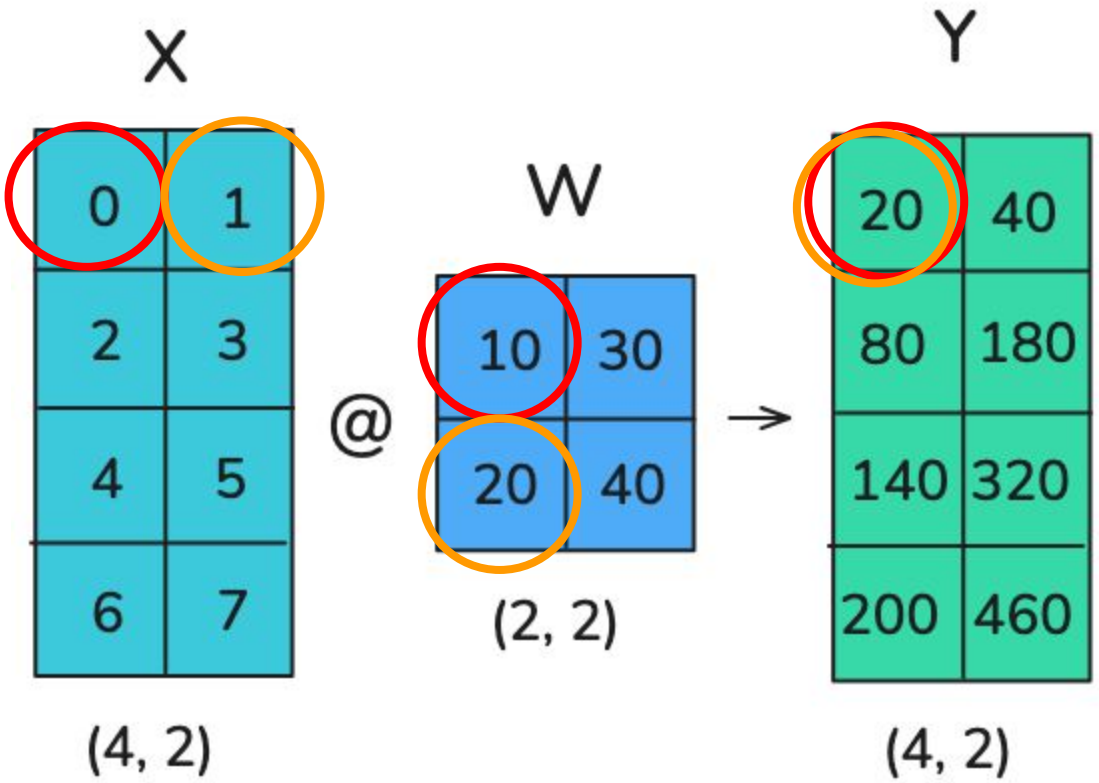

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

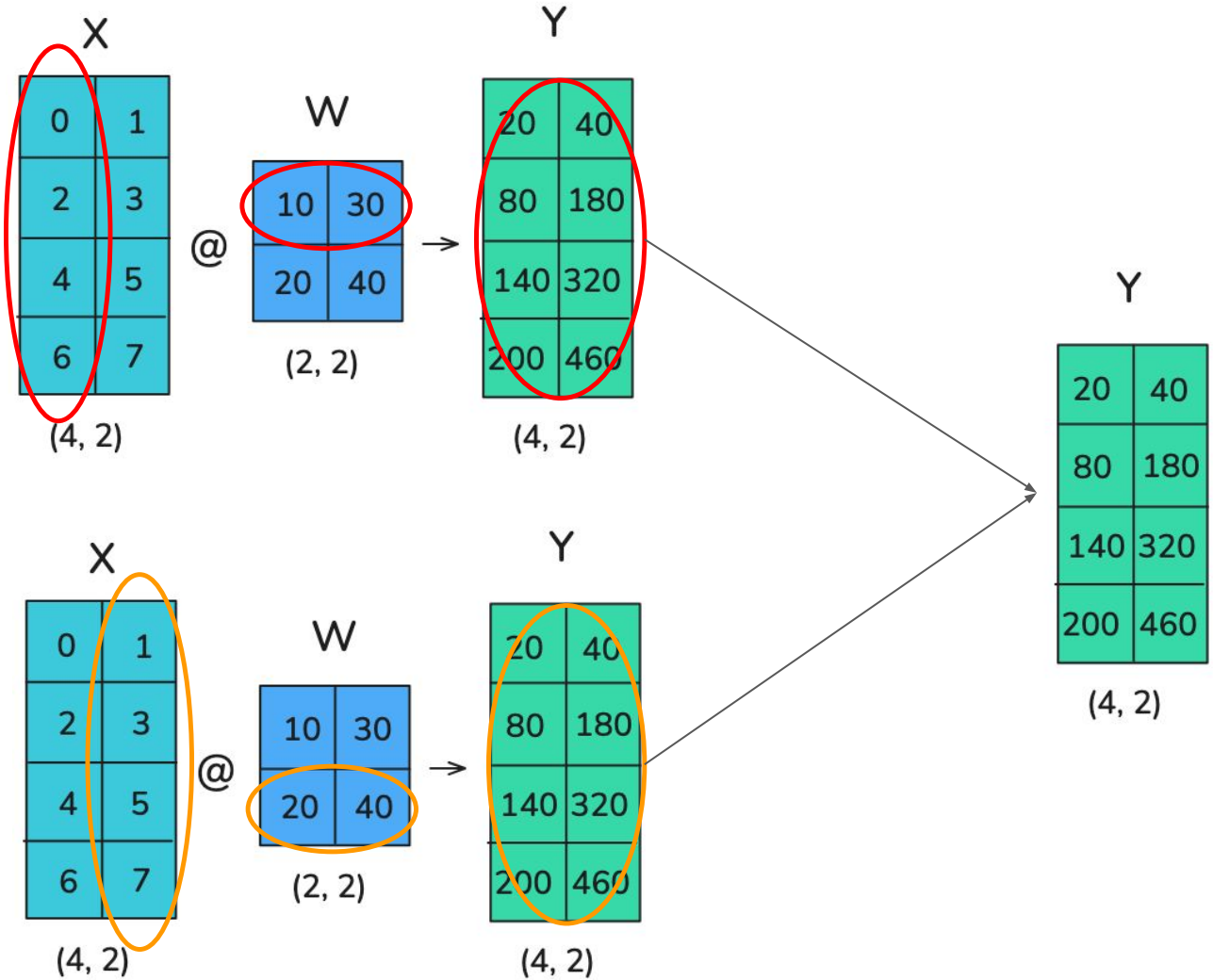

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)

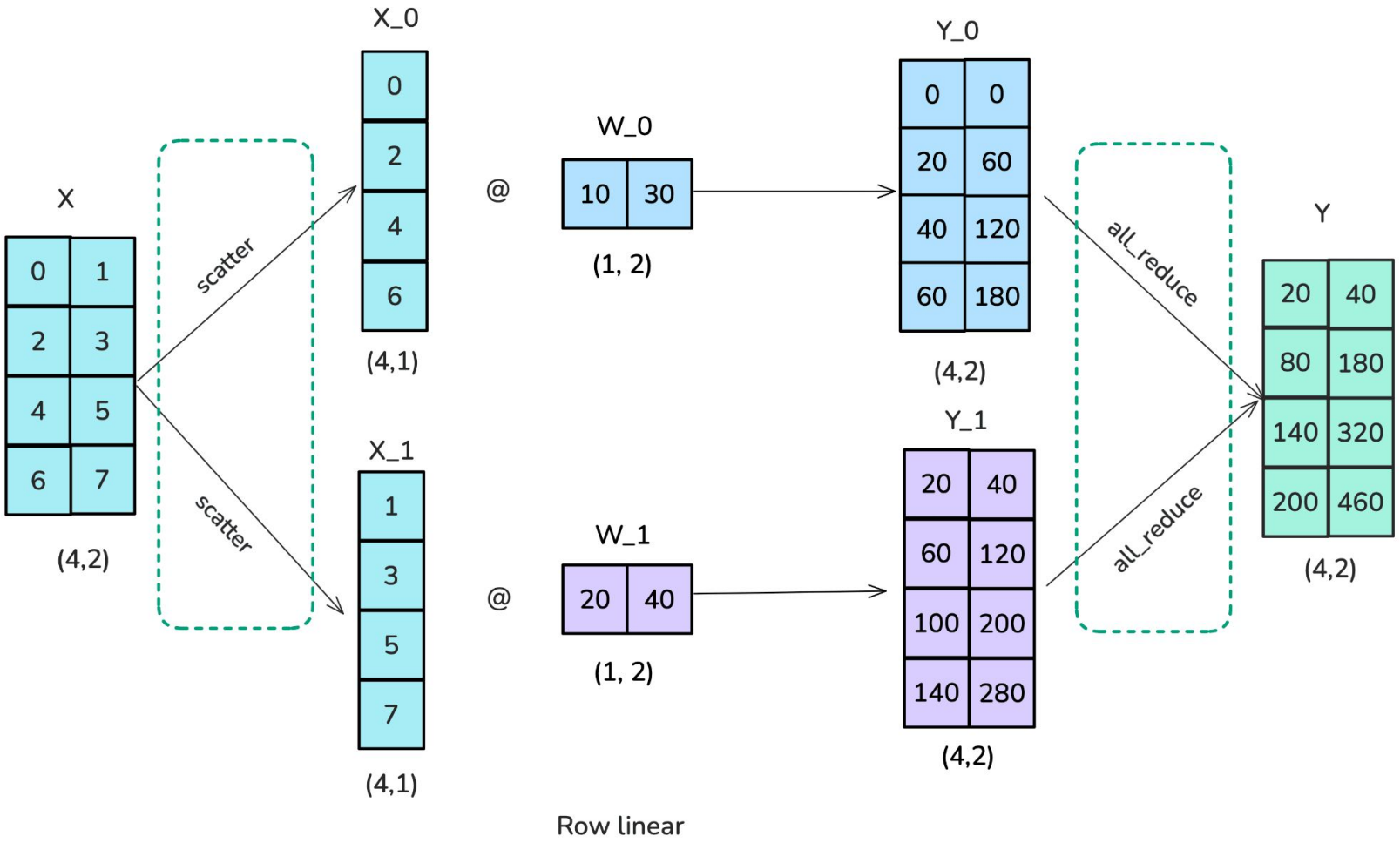


Column linear

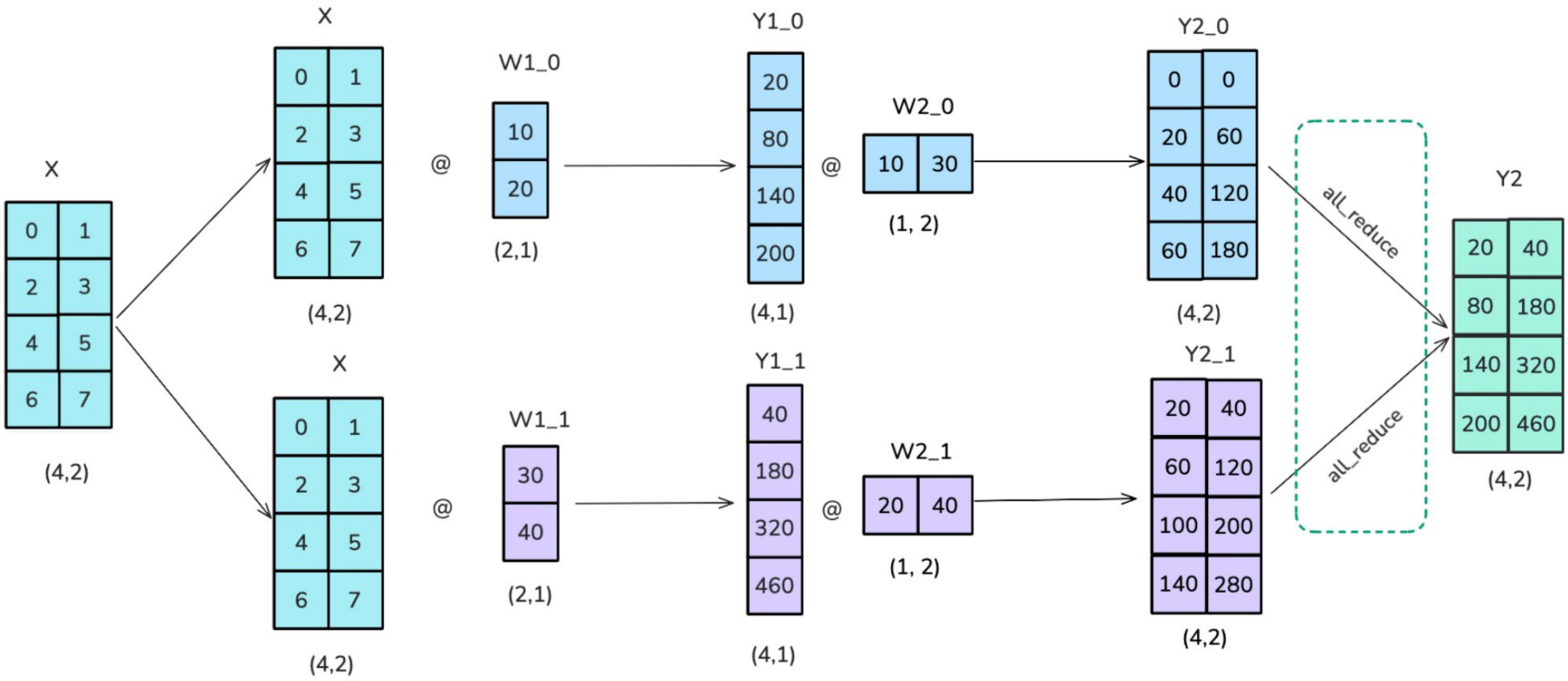

X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)


X
0
1
2
3
4
5
6
7
(4, 2)
@
W
10
30
20
40
(2, 2)
Y
20
40
80
180
140
320
200
460
(4, 2)


X
0 1
2 3
4 5
6 7
(4, 2)
@
W
10 30
20 40
(2, 2)
Y
20 40
80 180
140 320
200 460
(4, 2)
X
0 1
2 3
4 5
6 7
(4, 2)
@
W
10 30
20 40
(2, 2)
Y
20 40
80 180
140 320
200 460
(4, 2)
Y
20 40
80 180
140 320
200 460
(4, 2)

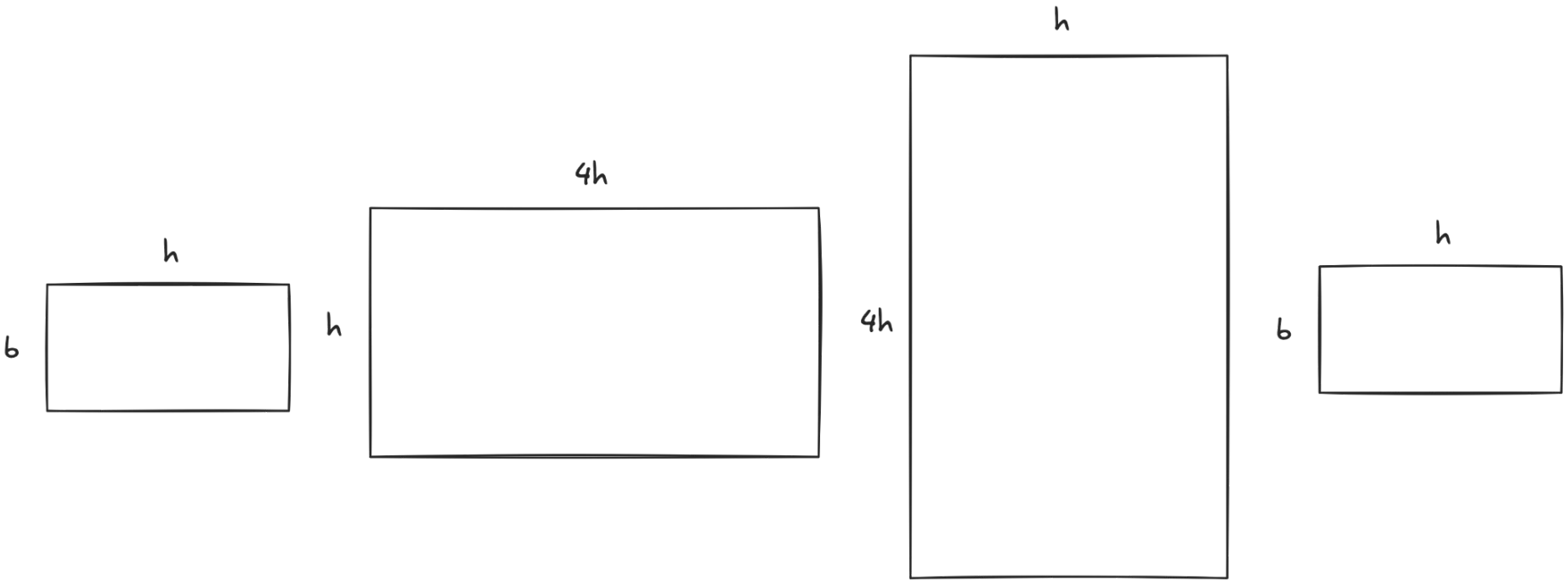

X
0 1
2 3
4 5
6 7
(4,2)
scatter
scatter
X_0
0
2
4
6
(4,1)
X_1
1
3
5
7
(4,1)
@
@
W_0
10 30
(1, 2)
W_1
20 40
(1, 2)
Y_0
0 0
20 60
40 120
60 180
(4,2)
Y_1
20 40
60 120
100 200
140 280
(4,2)
all_reduce
all_reduce
Y
20 40
80 180
140 320
200 460
(4,2)

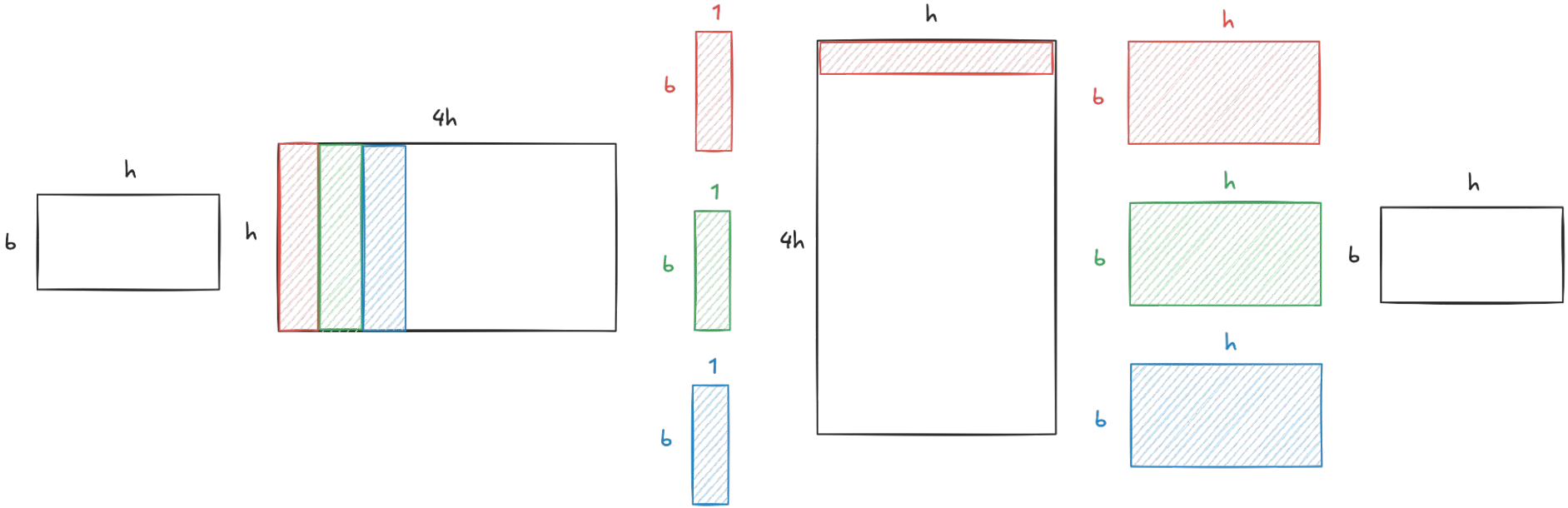
Row linear

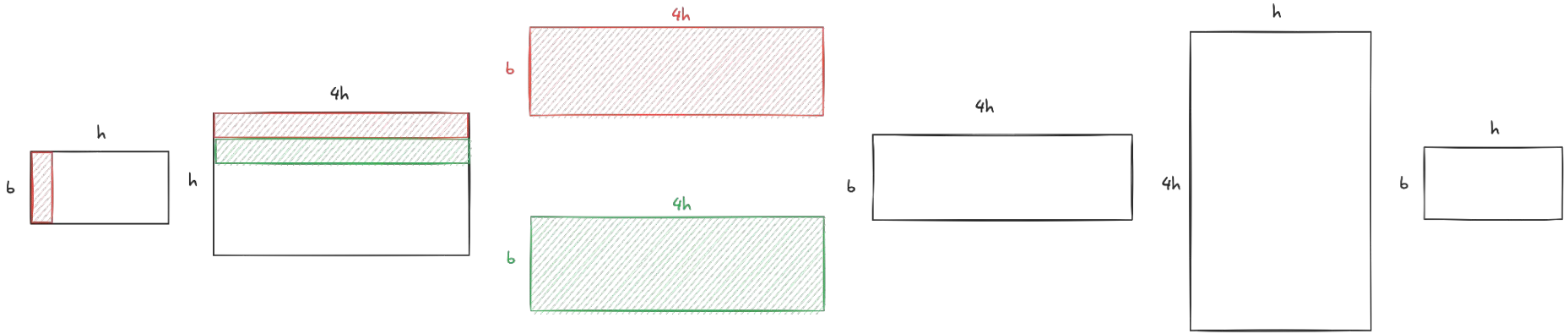


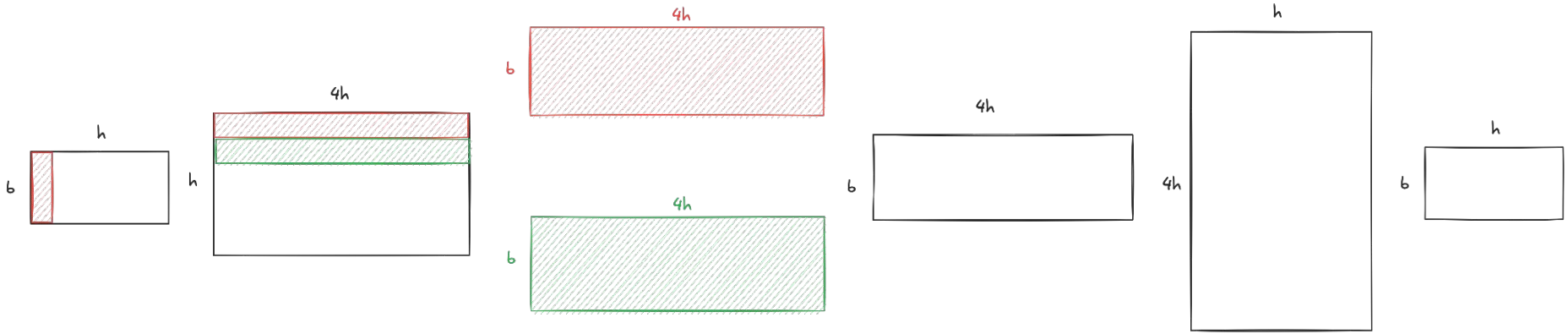
Tensor parallelism with column linear + row Linear

why combine 2 operations?

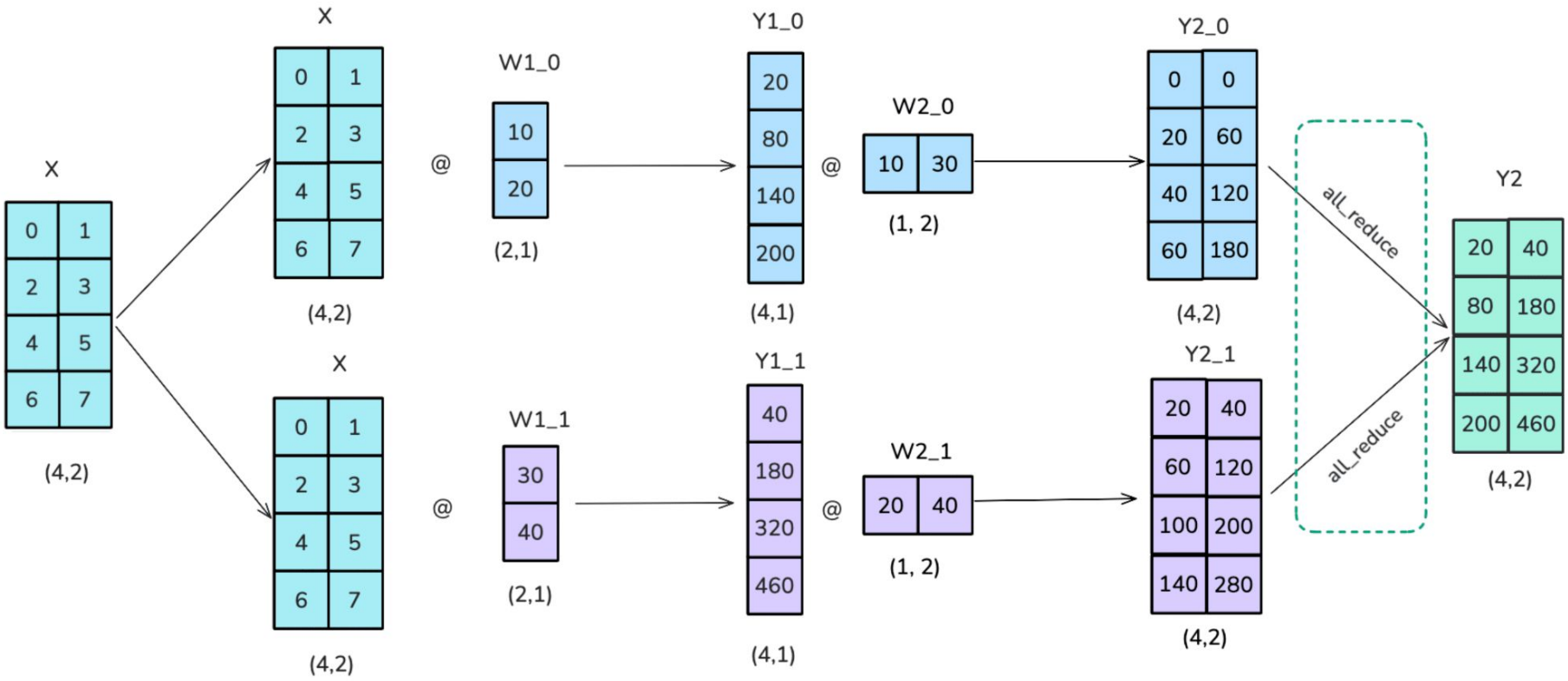

h
b
4h
h
h
4h
h
b


h
b
4h
h
1
b
1
b
1
b
h
4h
h
b
h
b
h
b
h
b

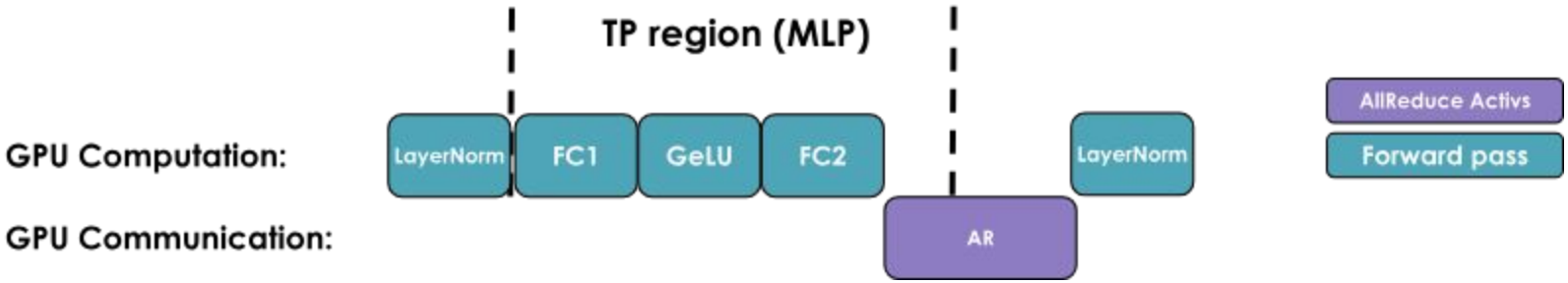

h
b
4h
h
4h
b
4h
b
4h
b
h
4h
h
b

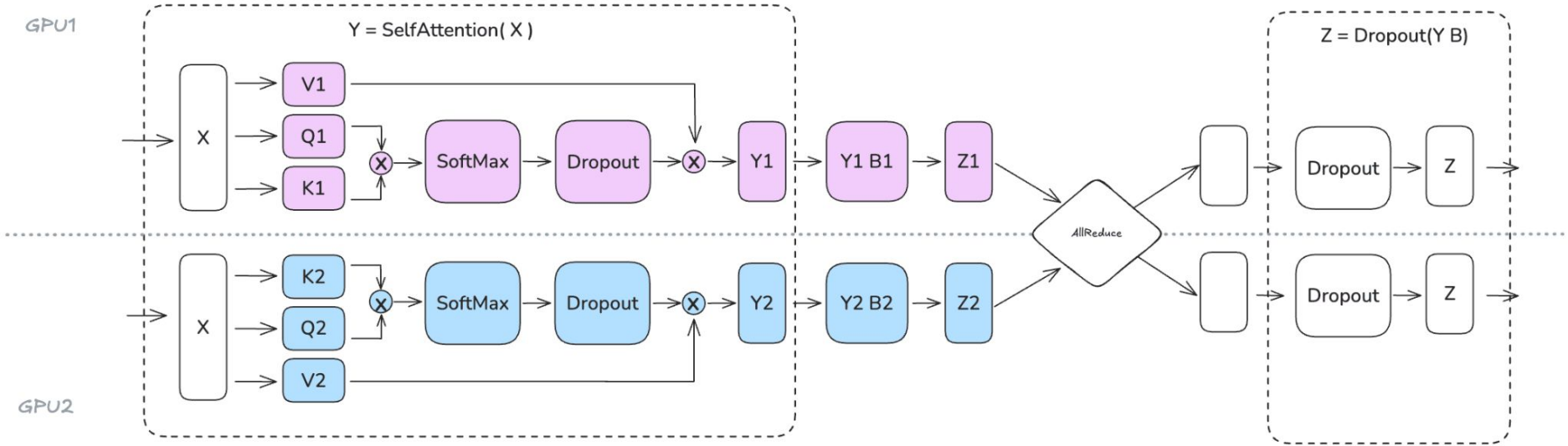


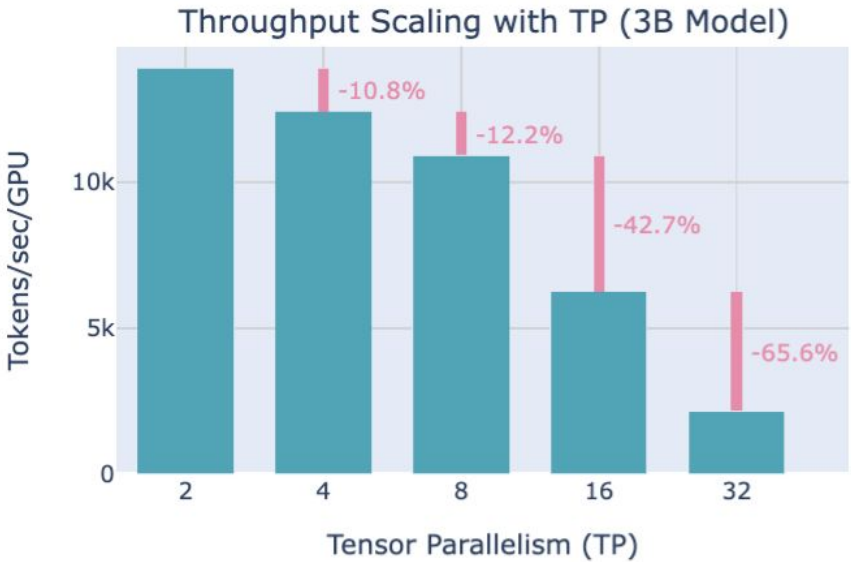
b * h << b * 4h + all_reduce

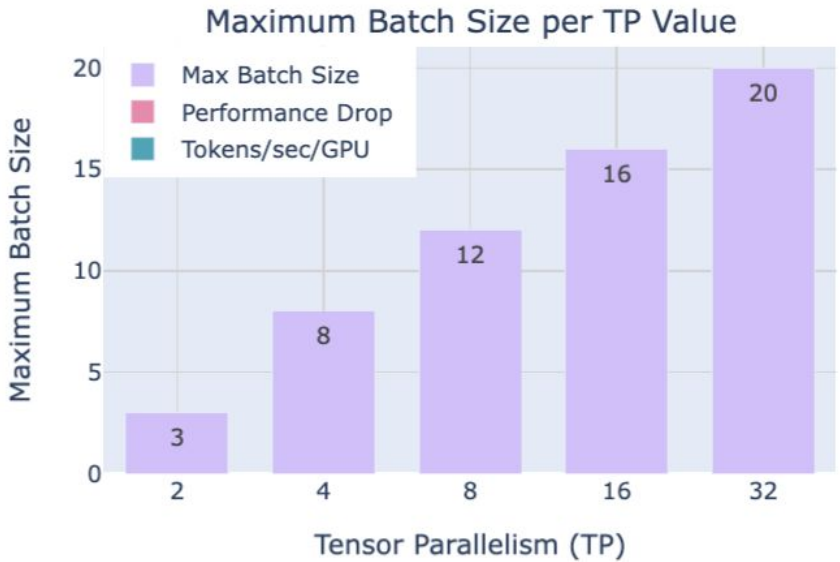


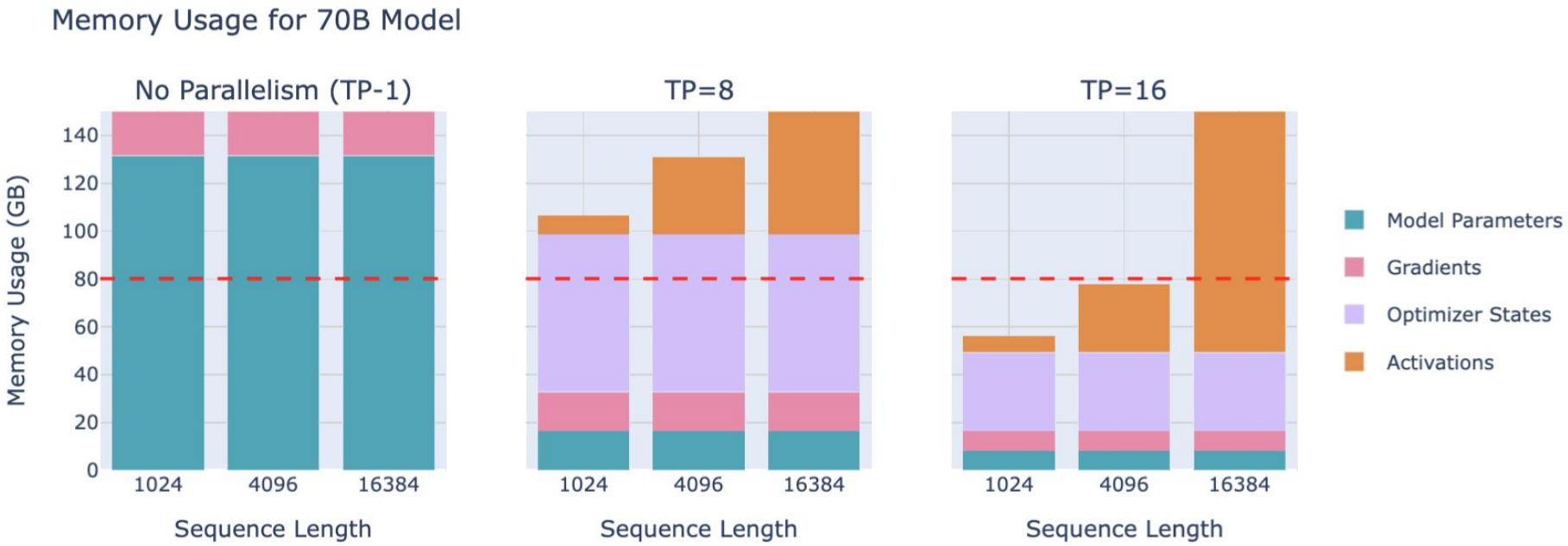
Tensor parallelism with column linear + row Linear

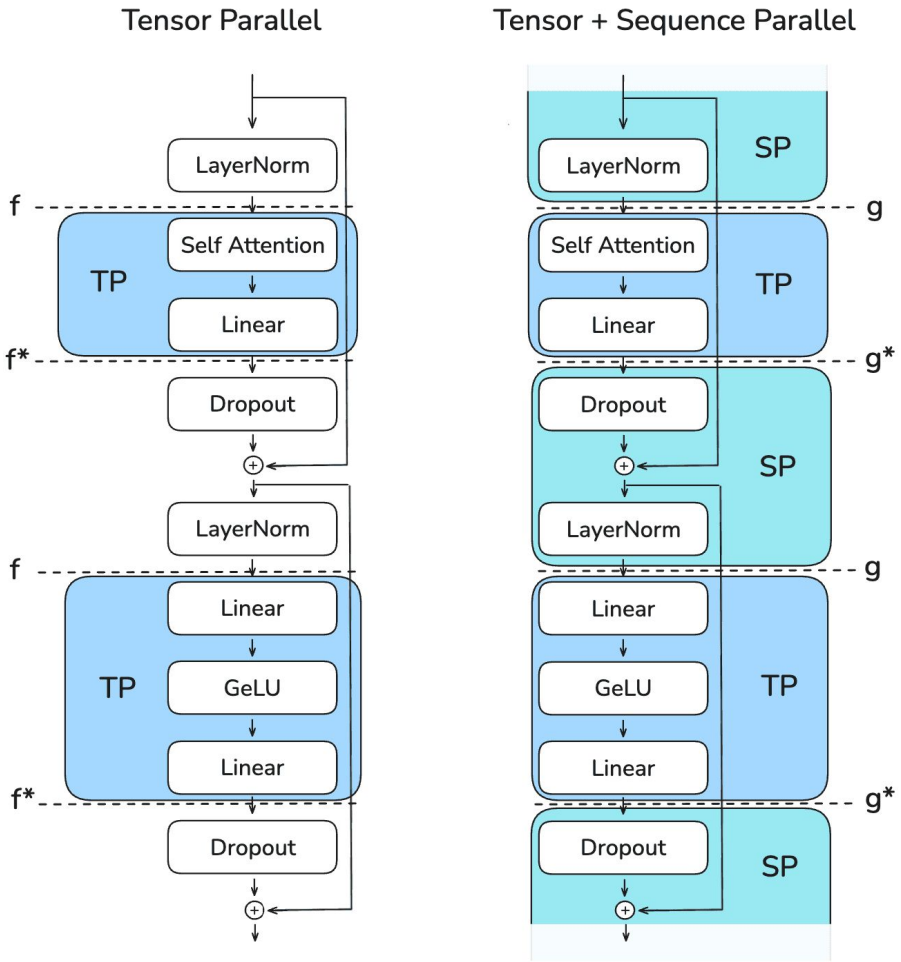

TP region (MLP)
GPU Computation:
GPU Communication:
LayerNorm
FC1
GeLU
FC2
AR
LayerNorm
AllReduce Activs
Forward pass

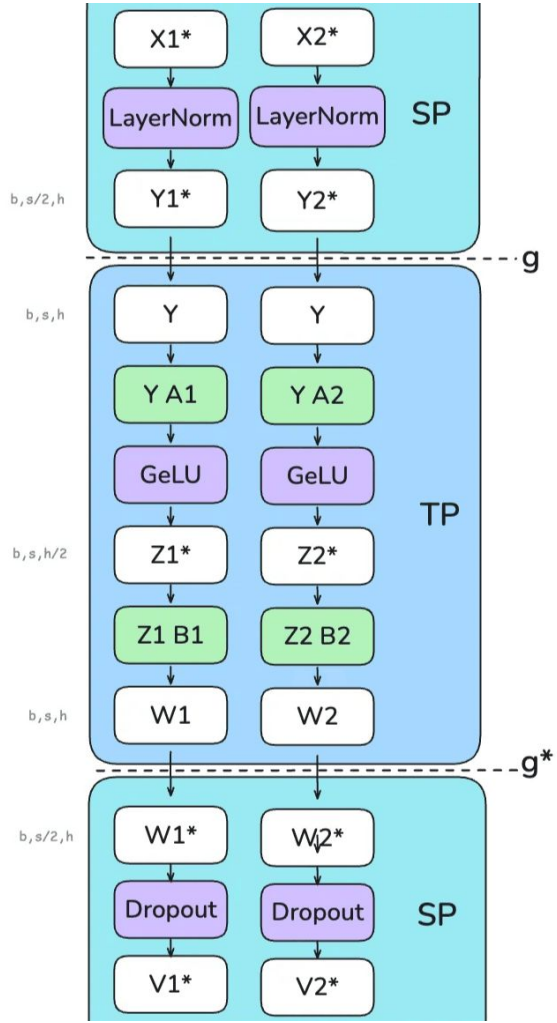

GPU1
Y = SelfAttention(X)
X
V1
Q1
K1
SoftMax
Dropout
Y1
Y1 B1
Z1
AllReduce
Z = Dropout(Y B)
Dropout
Z
GPU2
X
K2
Q2
V2
SoftMax
Dropout
Y2
Y2 B2
Z2
Dropout
Z


Throughput Scaling with TP (3B Model)
Tokens/sec/GPU
10k
5k
0
-10.8%
-12.2%
-42.7%
-65.6%
2
4
8
16
32
Tensor Parallelism (TP)

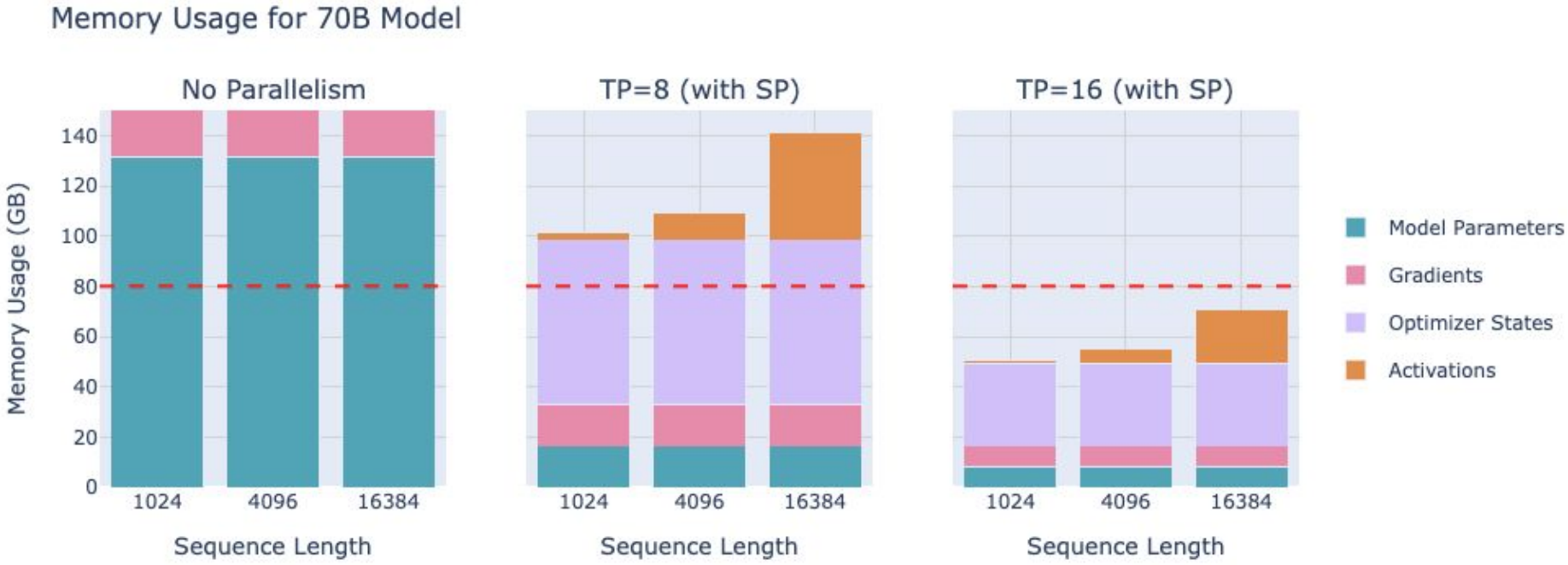

Maximum Batch Size per TP Value
Maximum Batch Size
Max Batch Size
Performance Drop
Tokens/sec/GPU
20
15
10
5
0
3
8
12
16
20
2
4
8
16
32
Tensor Parallelism (TP)

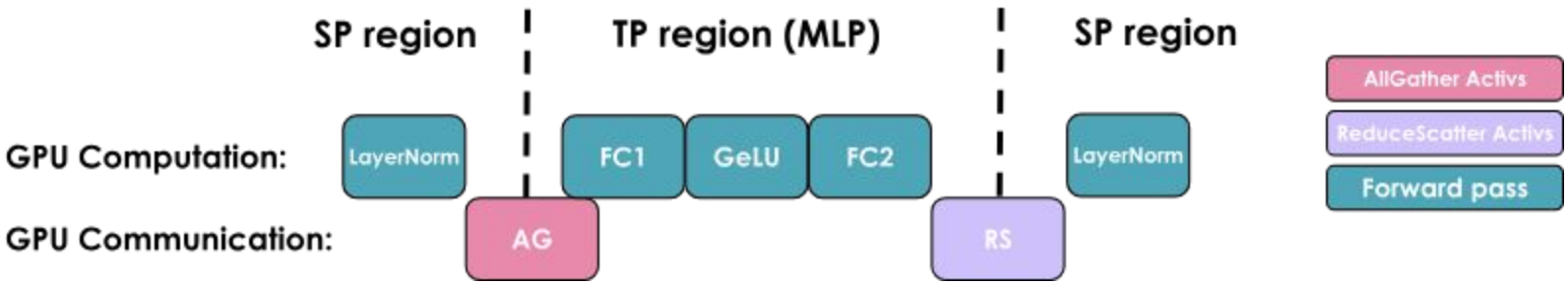

Memory Usage for 70B Model
No Parallelism (TP-1)
TP=8
TP=16
Memory Usage (GB)
0
20
40
60
80
100
120
140
1024
4096
16384
Sequence Length
Model Parameters
Gradients
Optimizer States
Activations


Tensor Parallel
Tensor + Sequence Parallel
LayerNorm
f
Self Attention
TP
Linear
f*
Dropout
LayerNorm
f
Linear
TP
GeLU
Linear
f*
Dropout
SP
g
TP
g*
SP
g
TP
g*
SP


X1*
X2*
LayerNorm
LayerNorm
SP
b,s/2,h
Y1*
Y2*
g
b,s,h
Y
Y
Y A1
Y A2
GeLU
GeLU
TP
b,s,h/2
Z1*
Z2*
Z1 B1
Z2 B2
b,s,h
W1
W2
g*
b,s/2,h
W1*
W2*
Dropout
Dropout
SP
V1*
V2*

Region	TP only	TP with SP
Enter TP (column-linear)	h: sharded (`weight_out` is sharded) s: full	h: sharded (`weight_out` is sharded) s: **all-gather** to full
TP region	h: sharded s: full	h: sharded s: full
Exit TP (row-linear)	h: full (`weight_out` is full + **all-reduce** for correctness) s: full	h: full (`weight_out` is full + **reduce-scatter** for correctness) s: **reduce-scatter** to sharded
SP region	h: full s: full	h: full s: sharded


Memory Usage for 70B Model
No Parallelism
TP=8 (with SP)
TP=16 (with SP)
Memory Usage (GB)
0
20
40
60
80
100
120
140
1024
4096
16384
Sequence Length
Model Parameters
Gradients
Optimizer States
Activations


SP region
TP region (MLP)
SP region
GPU Computation:
GPU Communication:
LayerNorm
FC1
GeLU
FC2
LayerNorm
AG
RS
AllGather Activs
ReduceScatter Activs
Forward pass

Still, there are two limits to TP+SP: if we scale the sequence length the activation memory will still blow up in the TP region, and if the model is too big to fit with TP=8 we will see a massive slowdown due to the inter-node connectivity.

see you next time