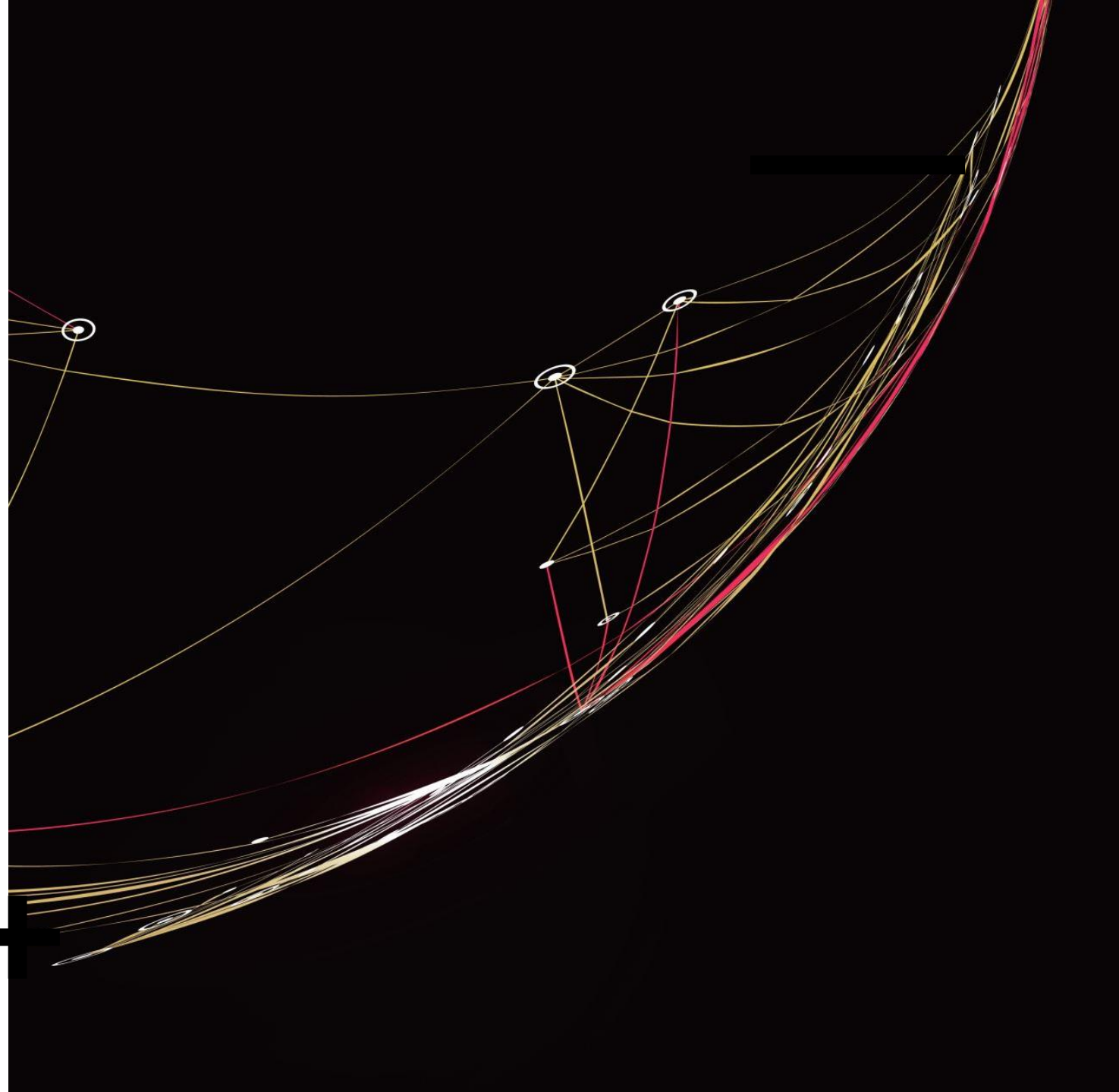
Untied Ulysses: Memory Efficient Context Parallelism via Headwise Chunking

ICML 2026

Ravi Ghadia, Maksim Abraham,
Sergei Vorobyov, Max Ryabinin

Together AI

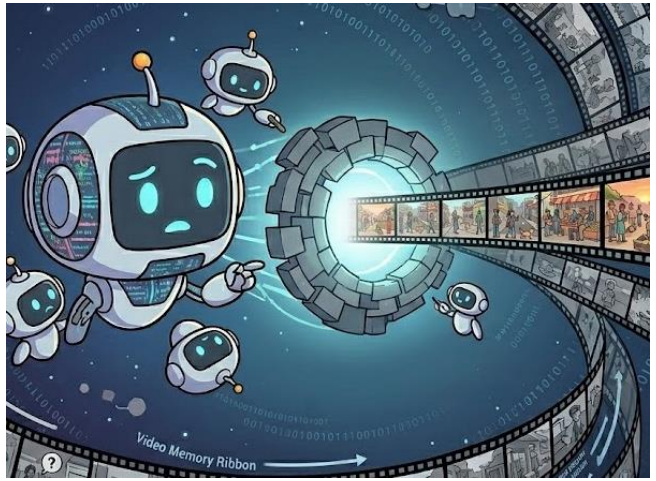


Agenda

- Long Context Training
- Context Parallelism
- Memory Efficiency
- UPipe
- Results
- Conclusion

Long Context Training

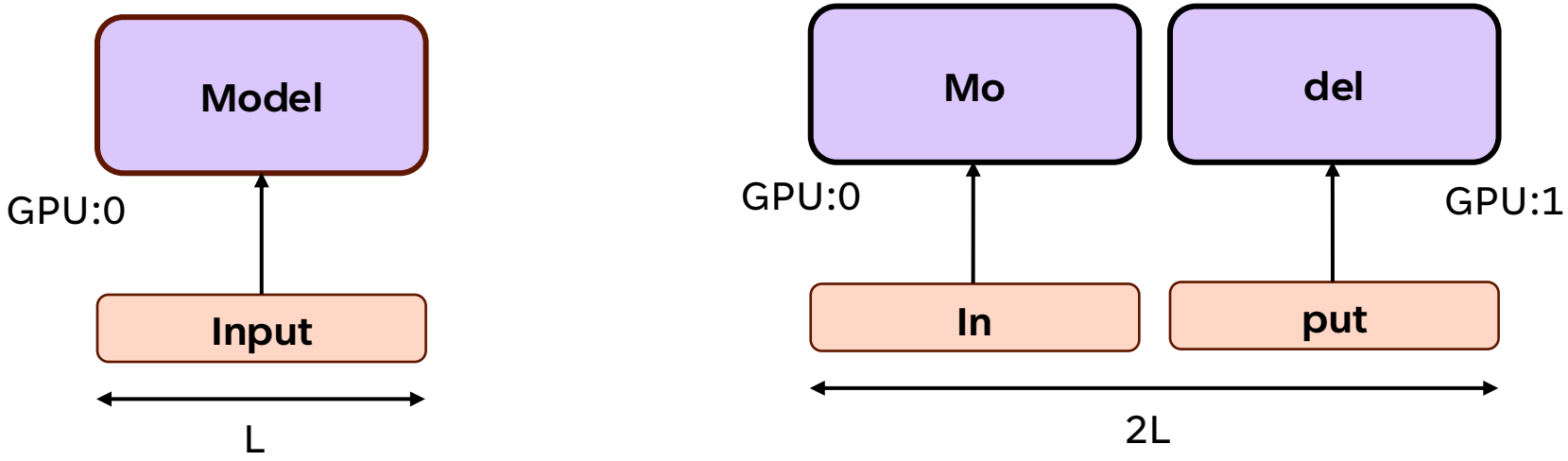
Code Assistants



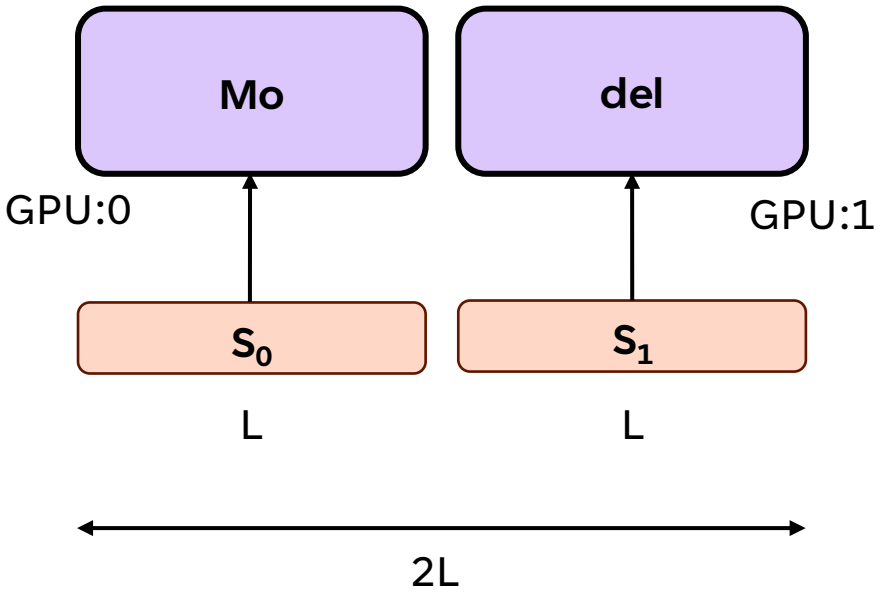
Video Processing

Document Summarization

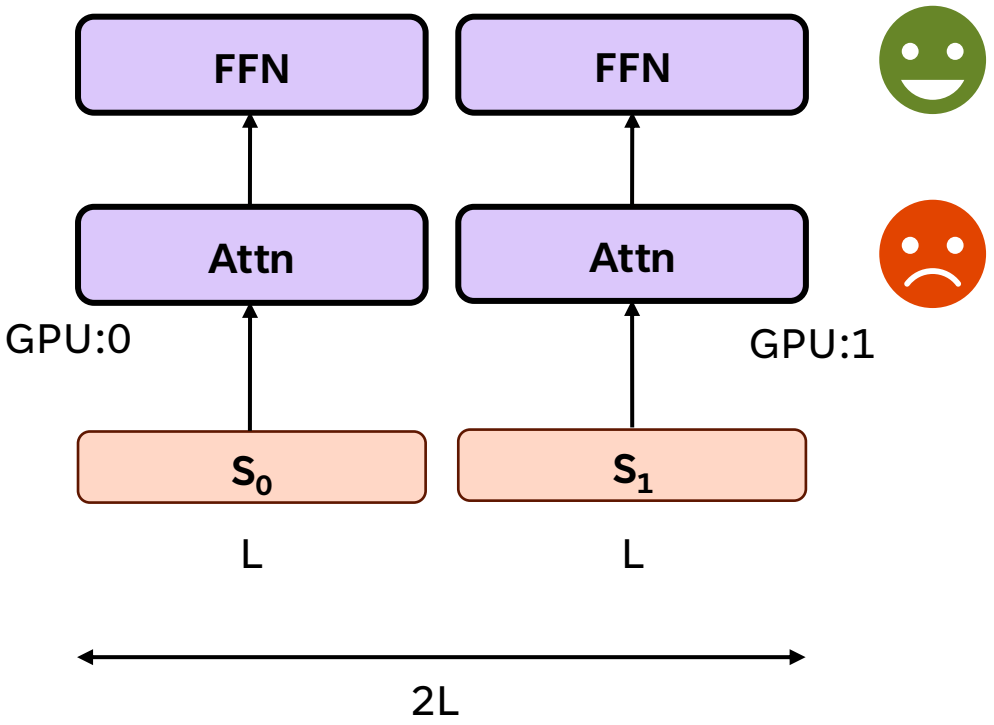
Context Parallelism



Context Parallelism



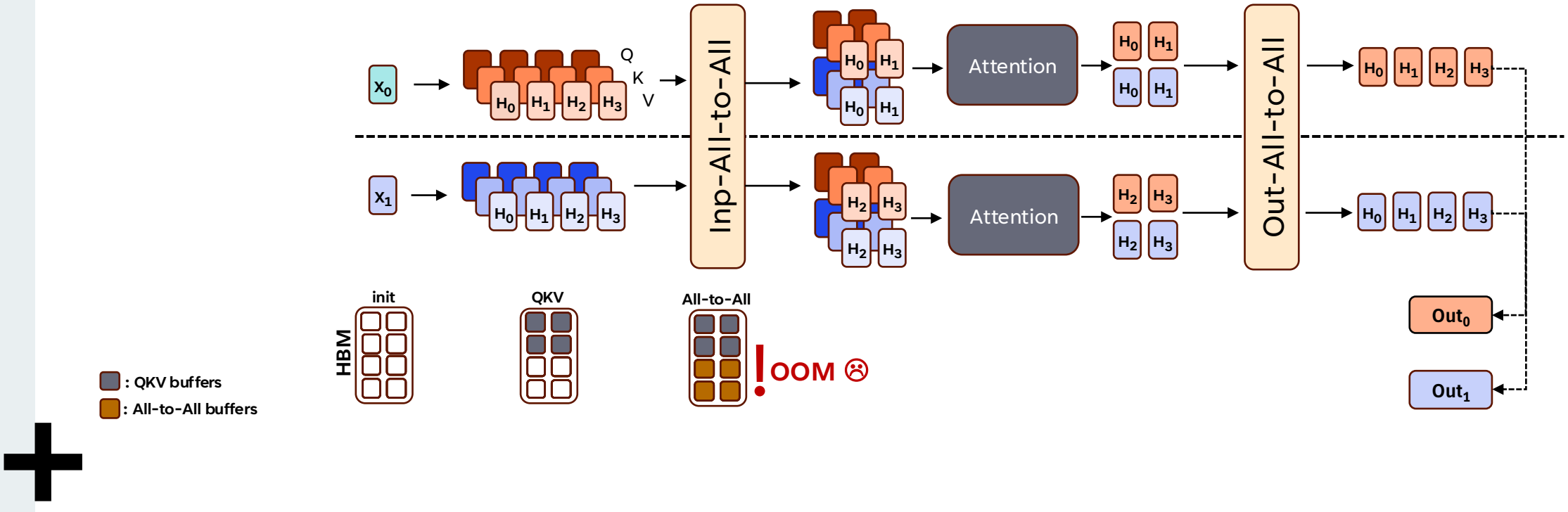
Context Parallelism



DeepSpeed Ulysses

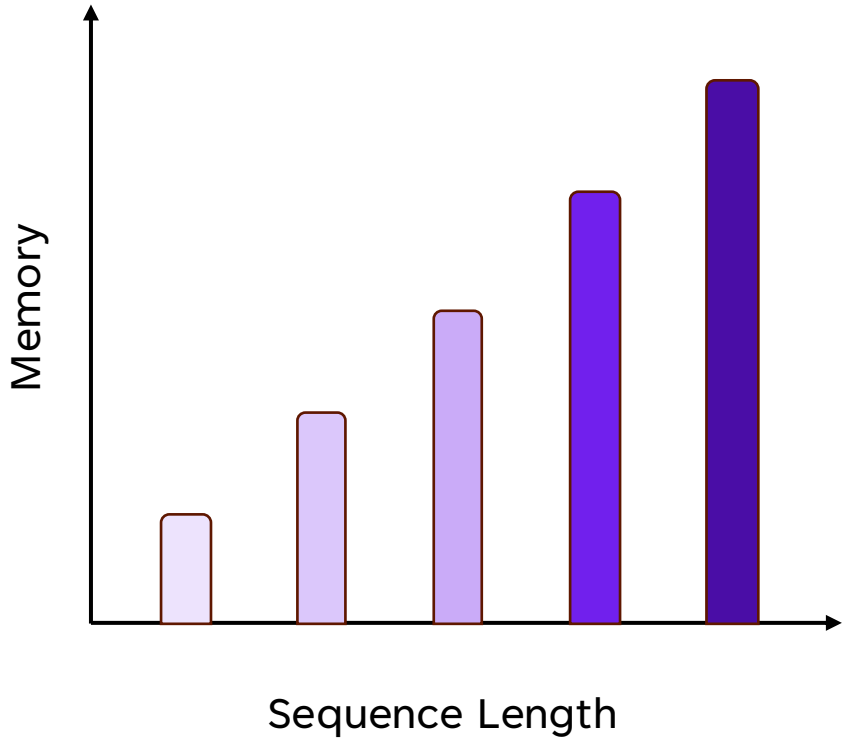
- Attention is independent across attention heads
- Rearrange tensors to **switch** sharding:
  - From sequence length dimension
  - To head dimension

DeepSpeed Ulysses



Memory Efficiency

– Activation memory increases linearly with sequence length



Memory Efficiency

– Activation memory increases linearly with sequence length

– Memory demand becomes prohibitive for long context training

	Stage	Inputs	Intermediate tensors			Outputs	Total
			Type	Memory	Ratio		
❶	Embedding	$4 \cdot S$ (int32)	–	–	–	$2 \cdot S \cdot d_{model}$	$2 \cdot S \cdot d_{model}$
❷	Attention	$2 \cdot S \cdot d_{model}$	QKV all-to-all	$6 \cdot S \cdot H \cdot d_{head}$ $6 \cdot S \cdot H \cdot d_{head}$	$H = d_{model}/d_{head}$	$2 \cdot S \cdot d_{model}$	$16 \cdot S \cdot d_{model}$
❸	Feed-forward	$2 \cdot S \cdot d_{model}$	Intermediate	$8 \cdot S \cdot d_{ff}$	$d_{ff} \approx 2.67 \cdot d_{model}$	$2 \cdot S \cdot d_{model}$	$25 \cdot S \cdot d_{model}$
❹	Cross-entropy	$2 \cdot S \cdot d_{model}$	Logits + LogSoftmax	$8 \cdot S \cdot V$	$V \approx 30 \cdot d_{model}$	Loss	$240 \cdot S \cdot d_{model}$

Memory Efficiency

Arctic Long Sequence Training (ALST)

- Tiling: Chunking along sequence length
  - TiledMLP (FFN), LigerKernel (CELoss)
- DeepSpeed Ulysses for Attention
  - Memory remains unaffected

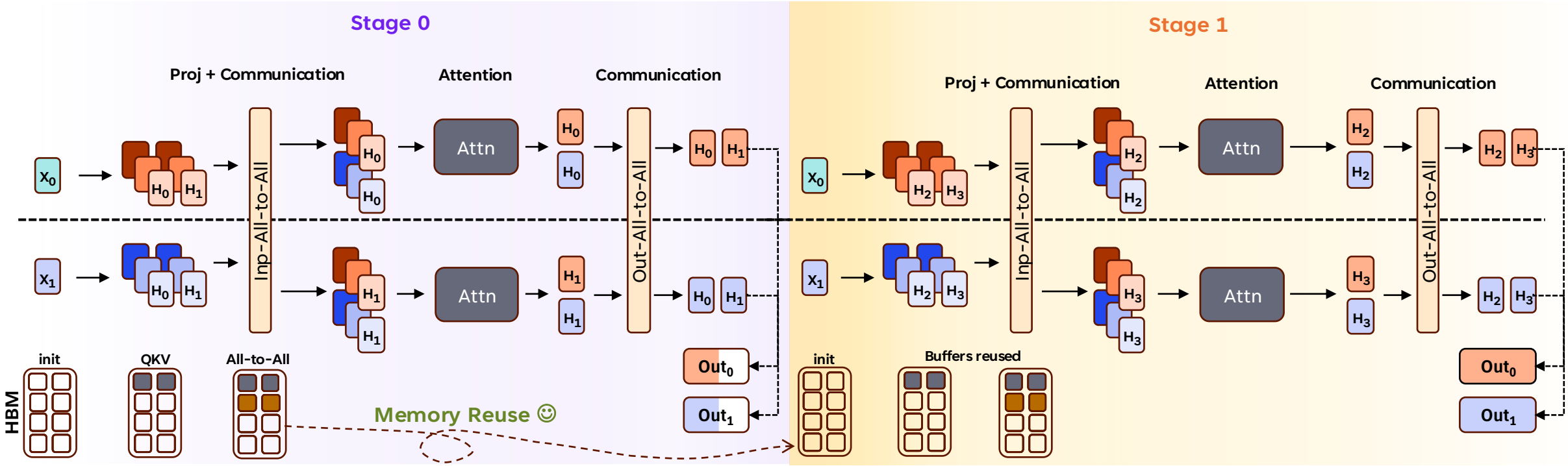
Fully Pipelined Distributed Transformer (FPDT)

- Tiling similar to ALST (FFN / CELoss)
- Chunks attention computation, uses online softmax
- CPU Offloading: Poor throughput

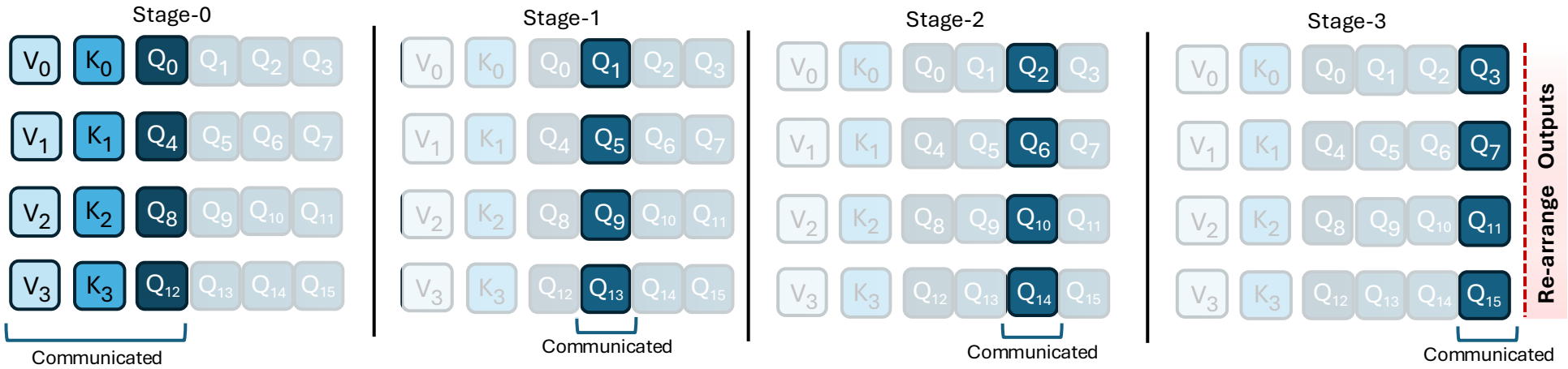
UPipe

- Long context: Attention on a subset of heads saturate the GPU
- Serialize attention computation across heads
  - Reuse memory buffers over subsequent stages

UPipe


Stage 0
Stage 1
Proj + Communication
Attention
Communication
Inp-All-to-All
Out-All-to-All
Attn
X_0
X_1
H_0
H_1
H_2
H_3
Out_0
Out_1
HBM
init
QKV
All-to-All
Memory Reuse ☺
Buffers reused

UPipe: GQA Scheduling



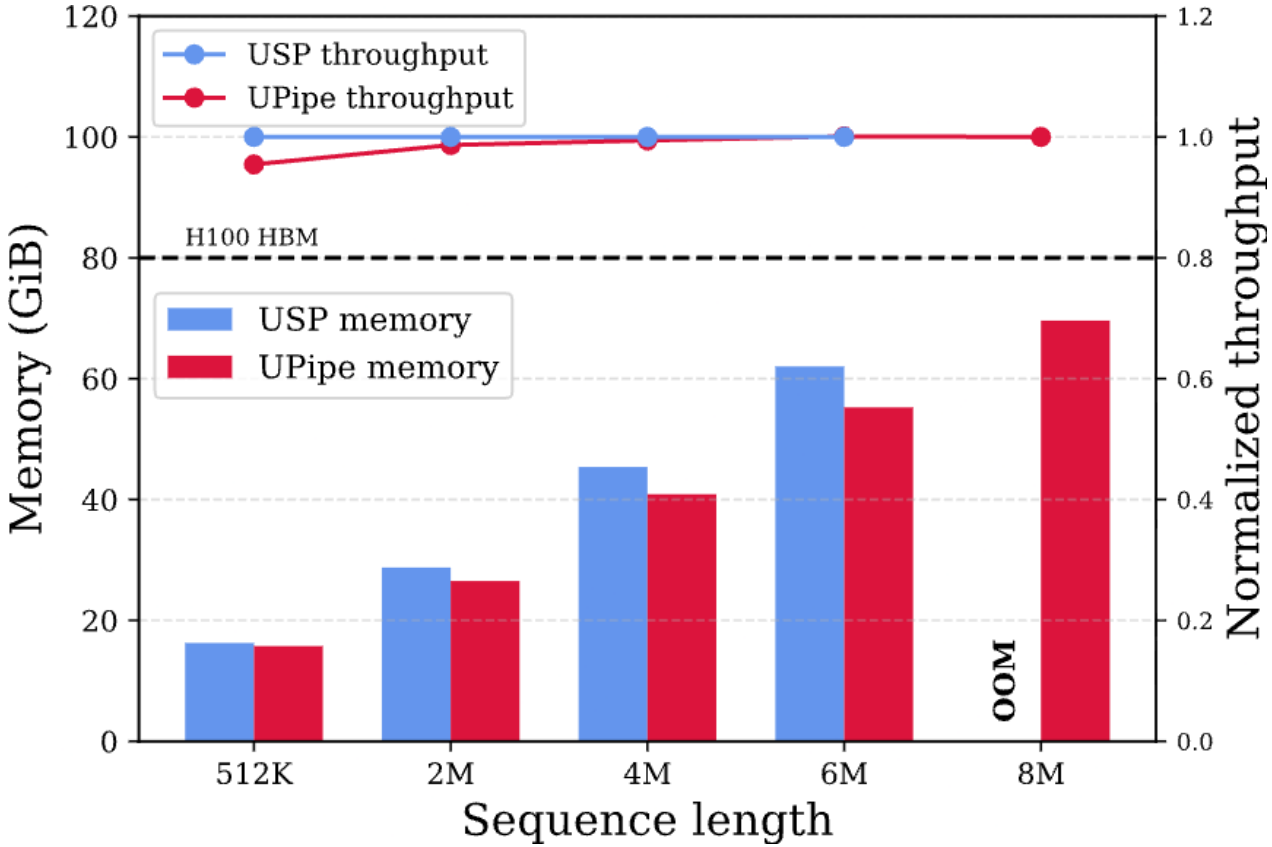
Results: Single Node (8xH100)

	Method	128K	256K	512K	1M	2M	3M	4M	5M
Llama 3-8B	Native PyTorch	1373.87	845.99	474.30	249.85	OOM	–	–	–
	Ring	2064.90	1387.67	841.05	458.51	237.99	159.96	OOM	–
	Ulysses	**2320.47**	**1503.80**	**878.63**	**475.33**	246.05	162.41	OOM	–
	FPDT	1171.68	884.75	621.20	382.42	219.53	153.48	119.76	–
	UPipe	2281.05	1487.29	867.17	472.53	**246.07**	**166.32**	**125.56**	**98.25**

Results: Multi Node (16xH100)

	Method	128K	256K	512K	1M	2M	3M	4M	5M
Qwen3-32B	Native PyTorch	127.03	112.20	91.39	OOM	–	–	–	–
	Ring	418.39	308.88	194.44	110.27	58.45	OOM	–	–
	Ulysses	**545.29**	**370.70**	**217.04**	**117.02**	**59.98**	OOM	–	–
	FPDT	286.40	217.85	151.91	95.88	55.41	38.86	27.66	–
	UPipe	483.29	339.56	204.46	113.26	**59.56**	**40.42**	**29.97**	OOM

Results: Multi Node (16xH100)



Training Llama 3-8B on two 8xH100 nodes

Reinvesting Memory Savings

Strided Activation Offloading

	Method	128K	256K	512K	1M
TPS	Ulysses	2417.79	**1516.51**	**880.27**	**476.45**
	UPipe	2377.98	1515.59	878.73	470.34
	UPipe-SAO	**2445.10**	1512.69	872.57	471.07
GiB	Ulysses	62.94	63.01	63.15	63.43
	UPipe	53.24	53.31	53.45	53.73
	UPipe-SAO	62.39	62.47	62.61	62.89

Reinvesting Memory Savings

Selective Activation Checkpointing

Method	Throughput	Memory
Ulysses	864.16	27.34
Ulysses+SAC	**1066.01**	27.85
UPipe	850.97	**25.24**
UPipe+SAC	1056.77	25.75

Conclusion

- Long context training necessitates context parallelism
- UPipe improves memory efficiency of context parallelism
- Reinvesting saved memory further helps improve training throughput
- Future work:
  - Improved all-to-all implementation for better memory reuse
  - SM-free collectives for better communication/computation overlap