PaReprop: Fast Parallelized Reversible Backpropagation

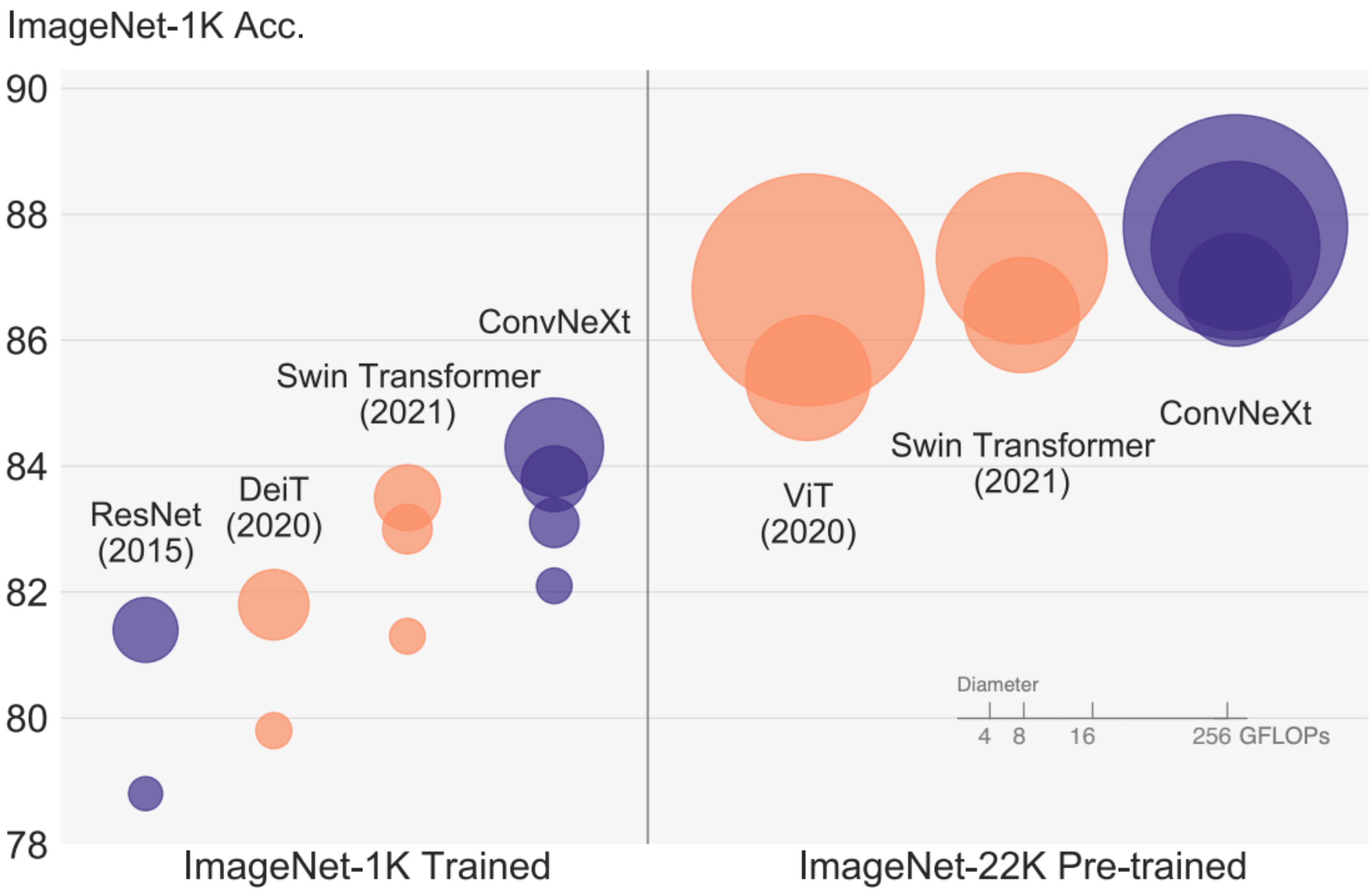
Transformers for Vision Workshop @ CVPR 2023, Spotlight

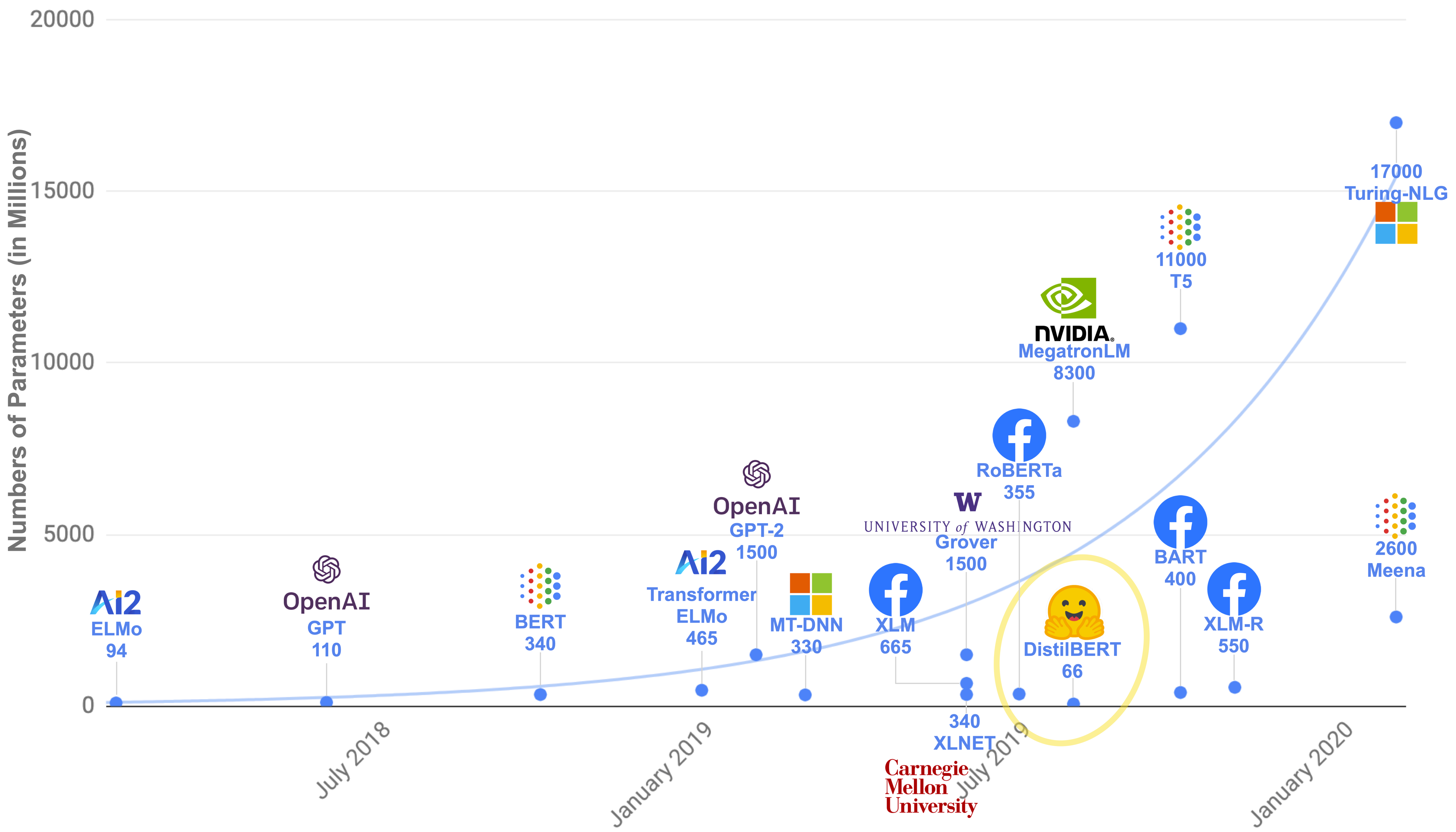
 

Tyler Zhu*, Karttikeya Mangalam*. UC Berkeley
06/18/2023

Motivation

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH





As models scale up, can we make more general, memory-efficient architectures?

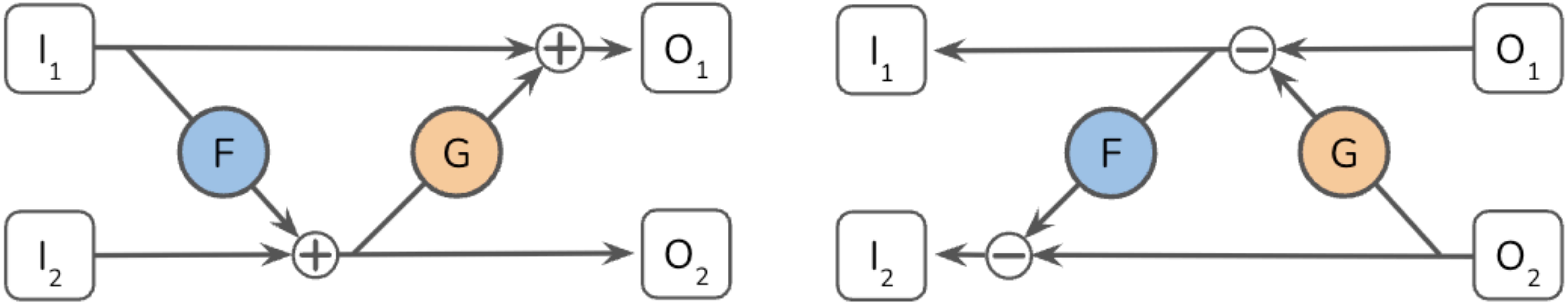
A ConvNet for the 2020s. Liu et. al, CVPR 2022.

https://huggingface.co/learn/nlp-course/chapter1/4

Reversible Transformations

- Key property: perfectly reconstruct inputs from outputs
- Functions F, G, need **not** be analytically invertible

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \underset{T_1}{\longmapsto} \begin{bmatrix} I_2 + F(I_1) \\ I_1 \end{bmatrix} = \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} := \mathbf{O} \quad \mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \underset{T_2}{\longmapsto} \begin{bmatrix} I_1 + G(I_2) \\ I_2 \end{bmatrix} = \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} := \mathbf{O}$$

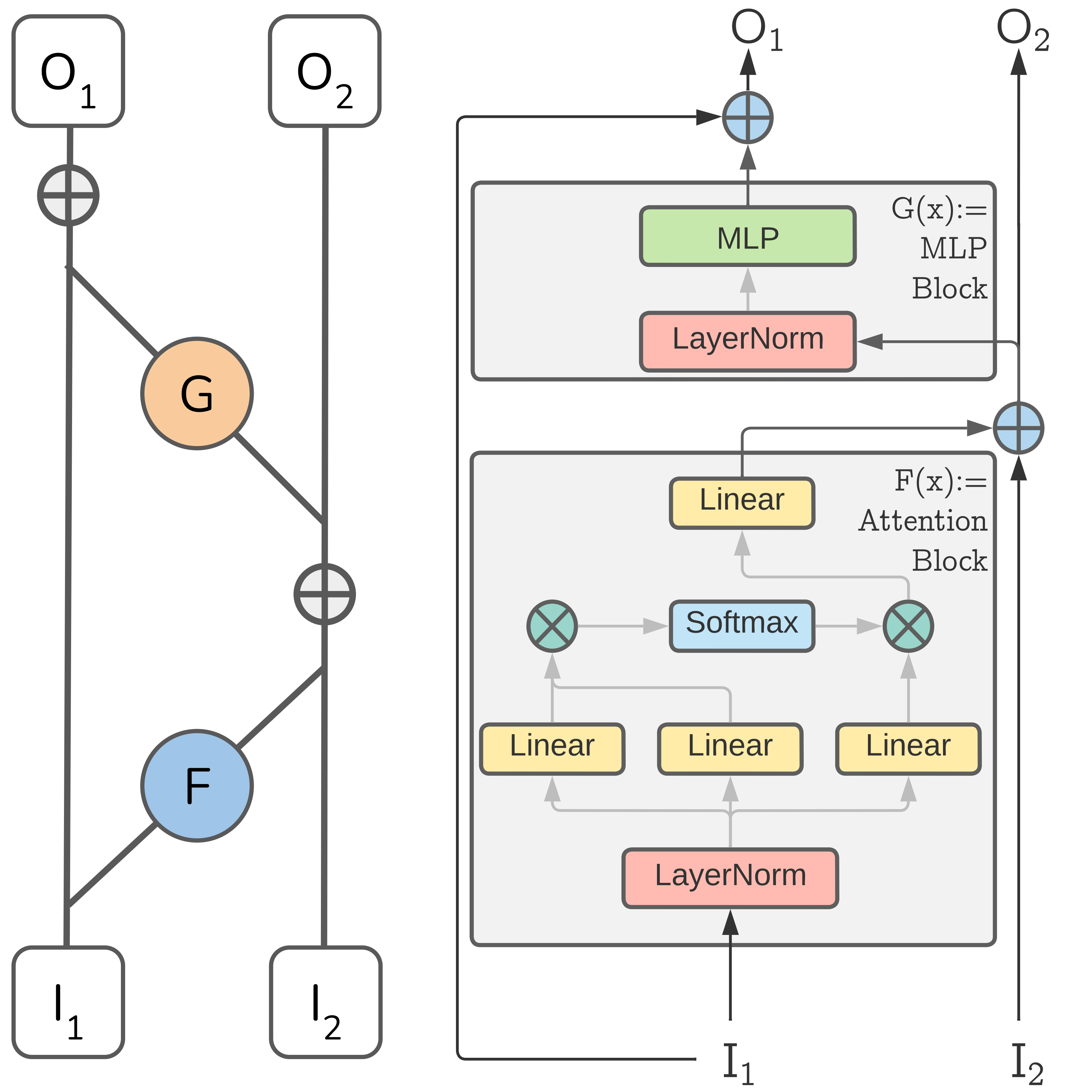


Nice: Non-linear independent components estimation. Dinh et. al, ICLR Workshop 2015.

Reversible Vision Transformers (RevViT)

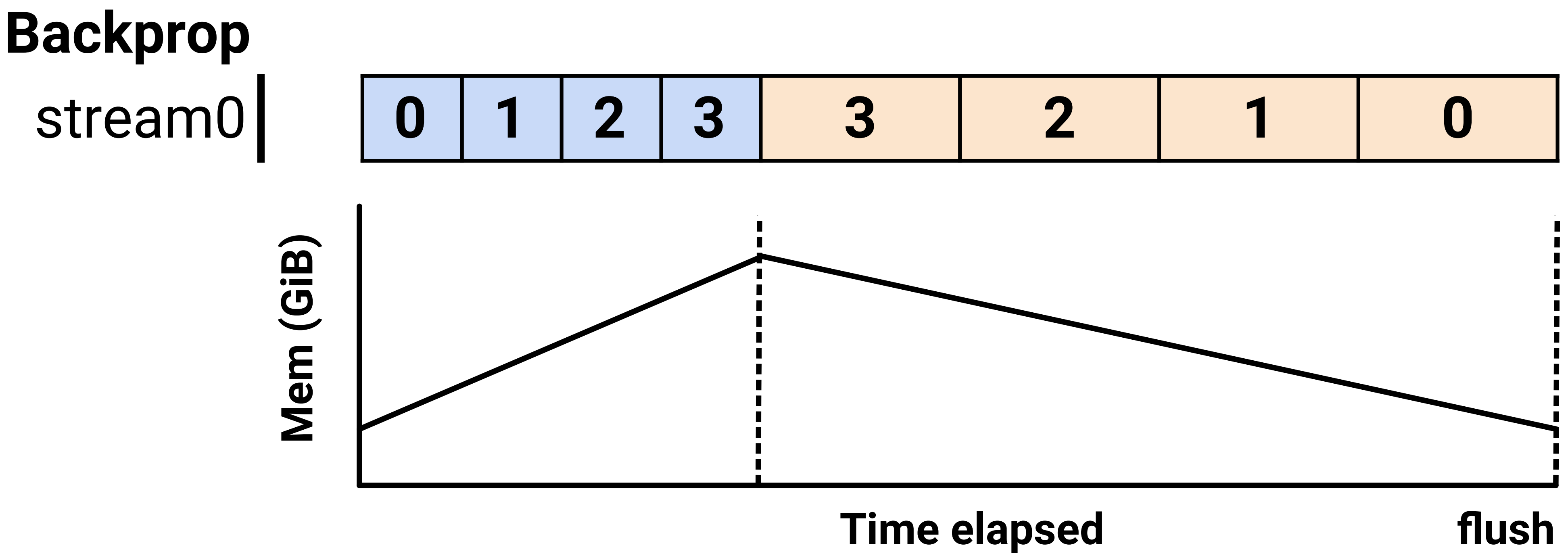
- Extend reversible transformations to transformers
- Set F(x) = Attention Block, G(x) = MLP Block
  - Ignore activation caching in forward pass
  - Recover them in the backward pass
- Achieves equal perf, uses less memory

model	Acc	**Memory** (MB/img)	**Maxiumum Batch Size**	GFLOPs	Param (M)
ViT-S [59]	79.9	66.5	207	4.6	22
Rev-ViT-S	79.9	**8.8** ↓7.6×	**1232** ↑6.0×	4.6	22
ViT-B [59]	81.8	129.7	95	17.6	87
Rev-ViT-B	81.8	**17.0** ↓7.6×	**602** ↑6.3×	17.6	87



Reversible Vision Transformers. Mangalam et. al, CVPR 2022.

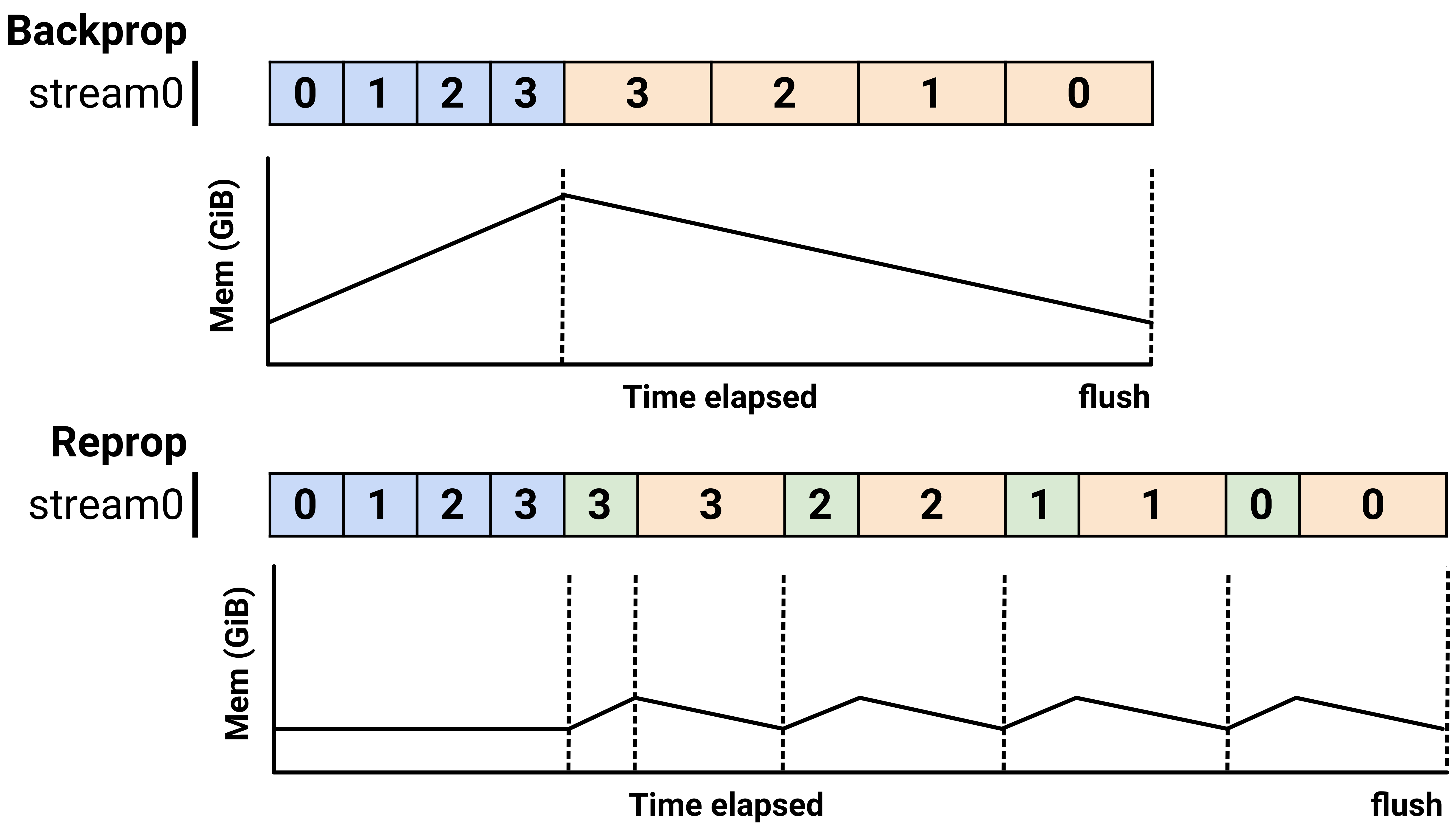
The backprop in detail



x	Forward pass of block **x**
x	Backward pass with gradient updates of block **x**

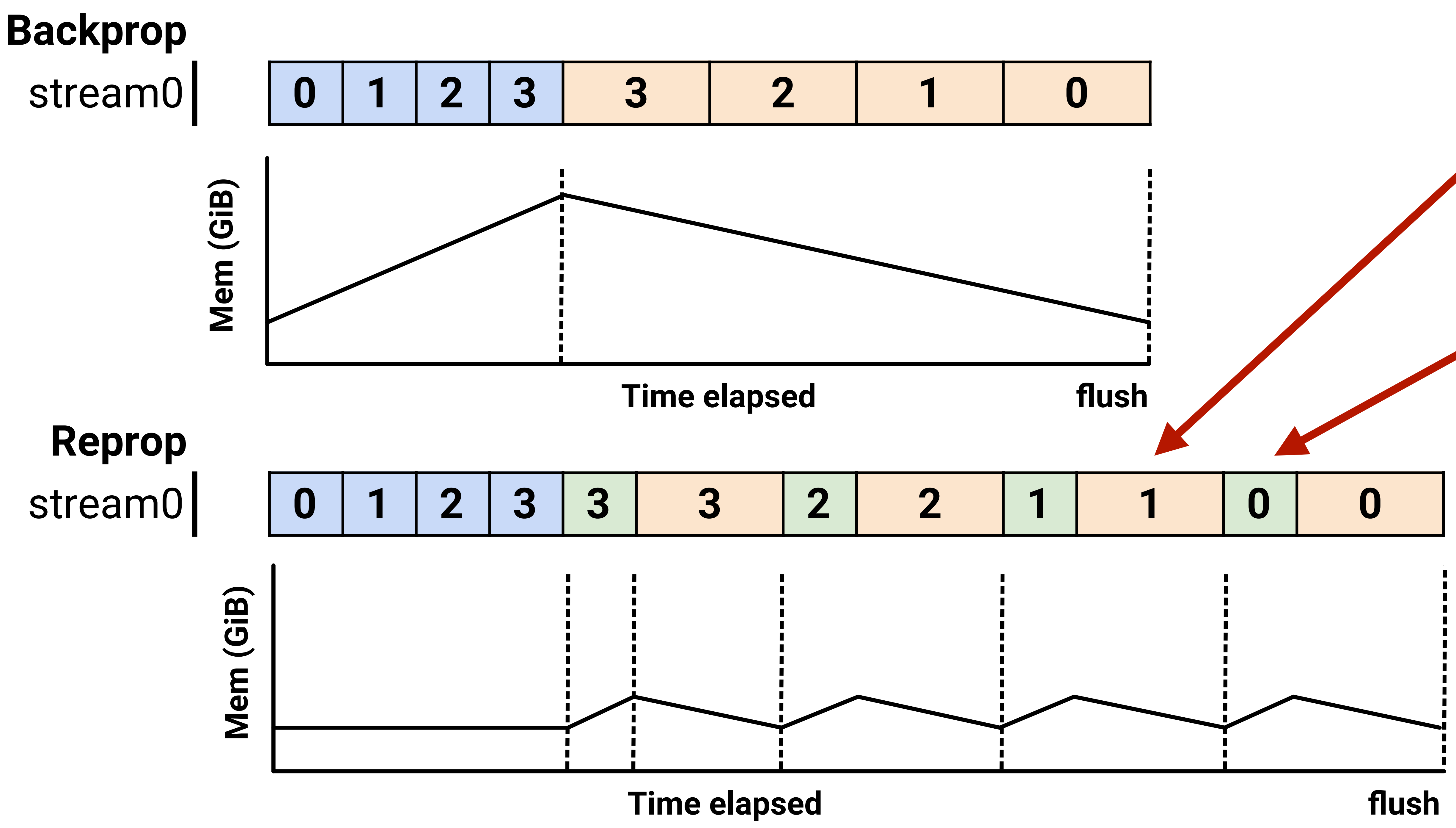
The backprop in detail

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH



- x — Forward pass of block **x**
- x — Backward pass with gradient updates of block **x**
- x — Activation recomputation in reversible backprop of block **x**

The backprop in detail

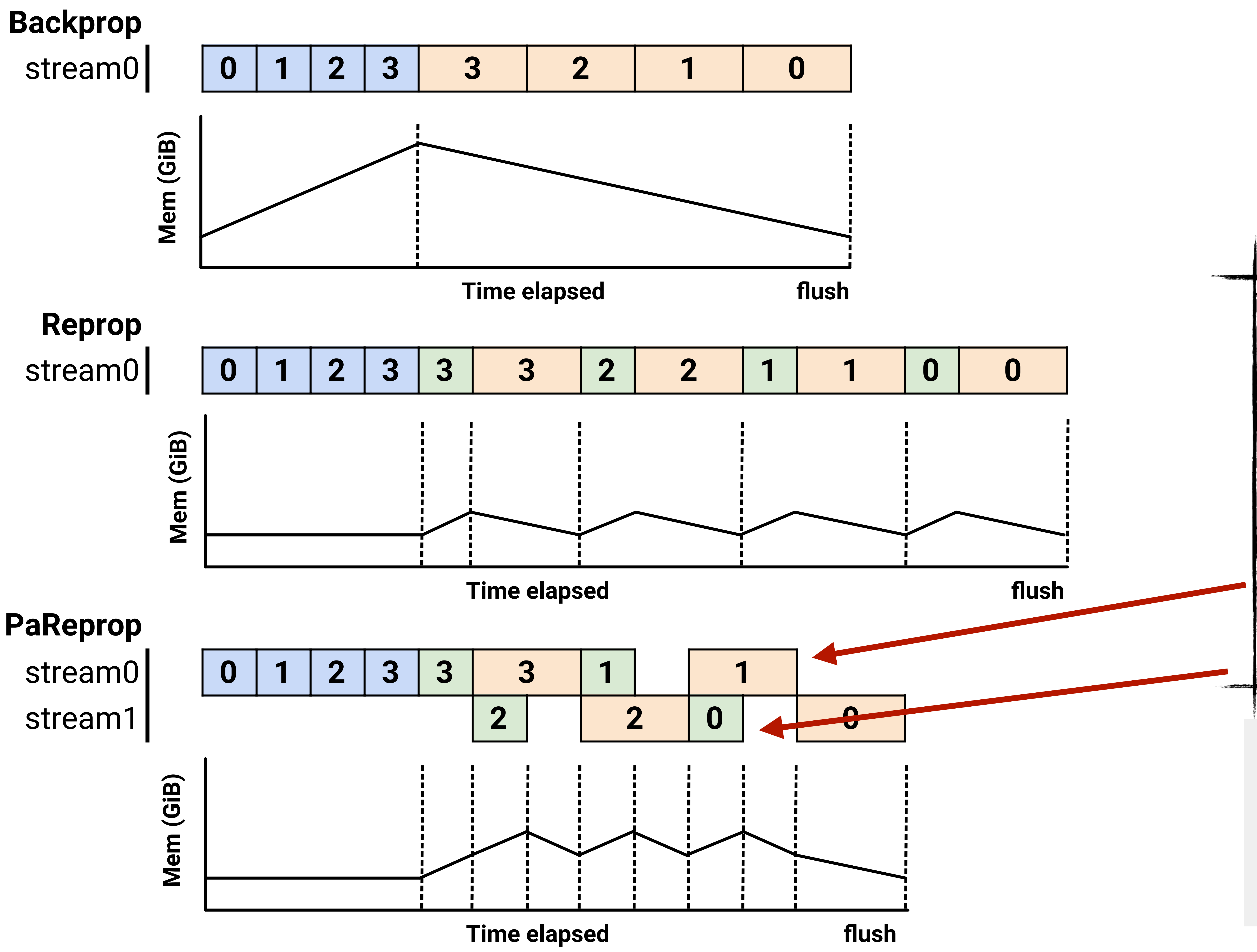


No dependency b/w the blocks!

- Can update gradients of block 1 and recompute block 0 activations at the same time

x	Forward pass of block **x**
x	Backward pass with gradient updates of block **x**
x	Activation recomputation in reversible backprop of block **x**

PaReprop: Parallelized Reversible Backprop

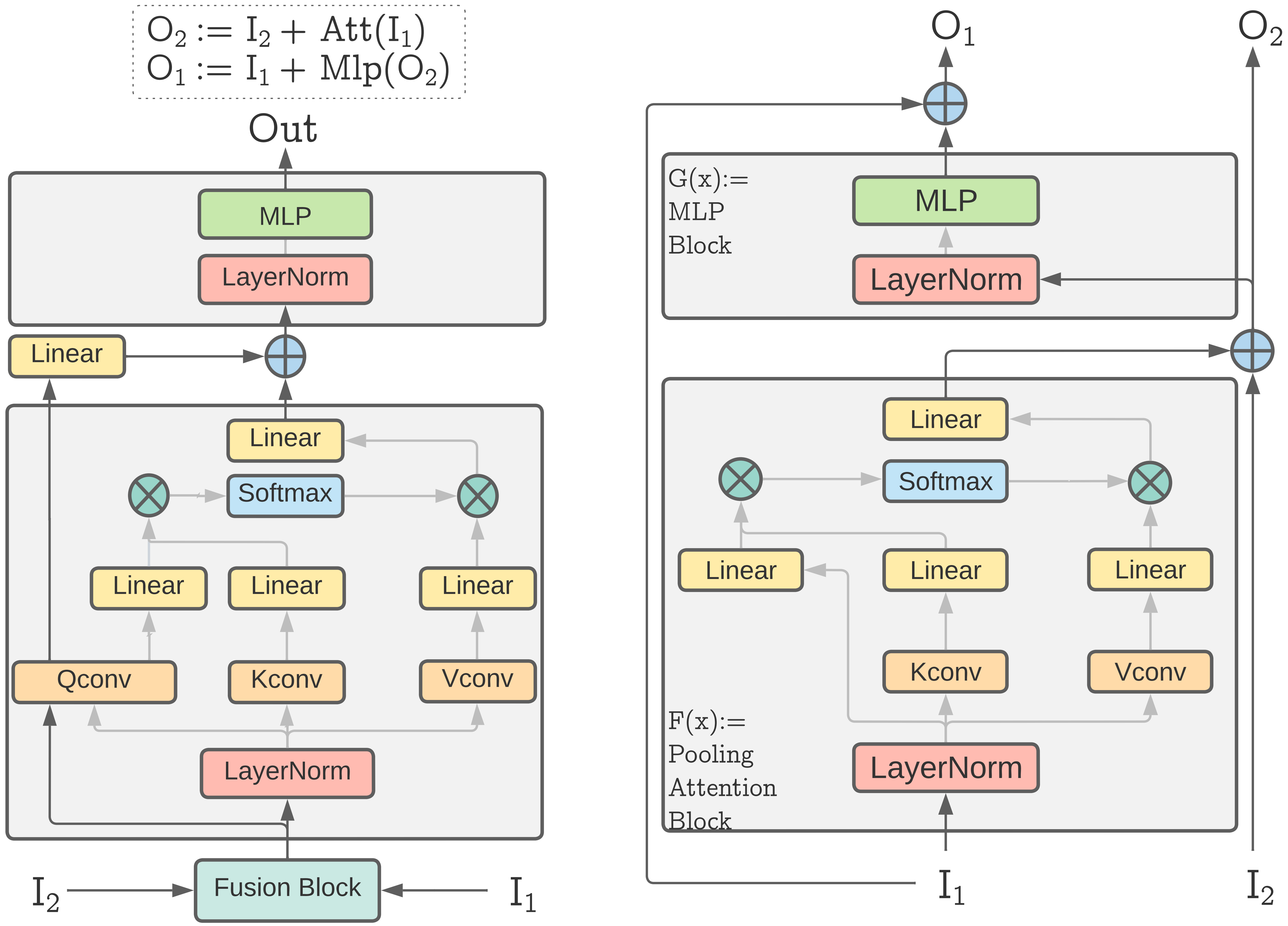


Now run both in parallel

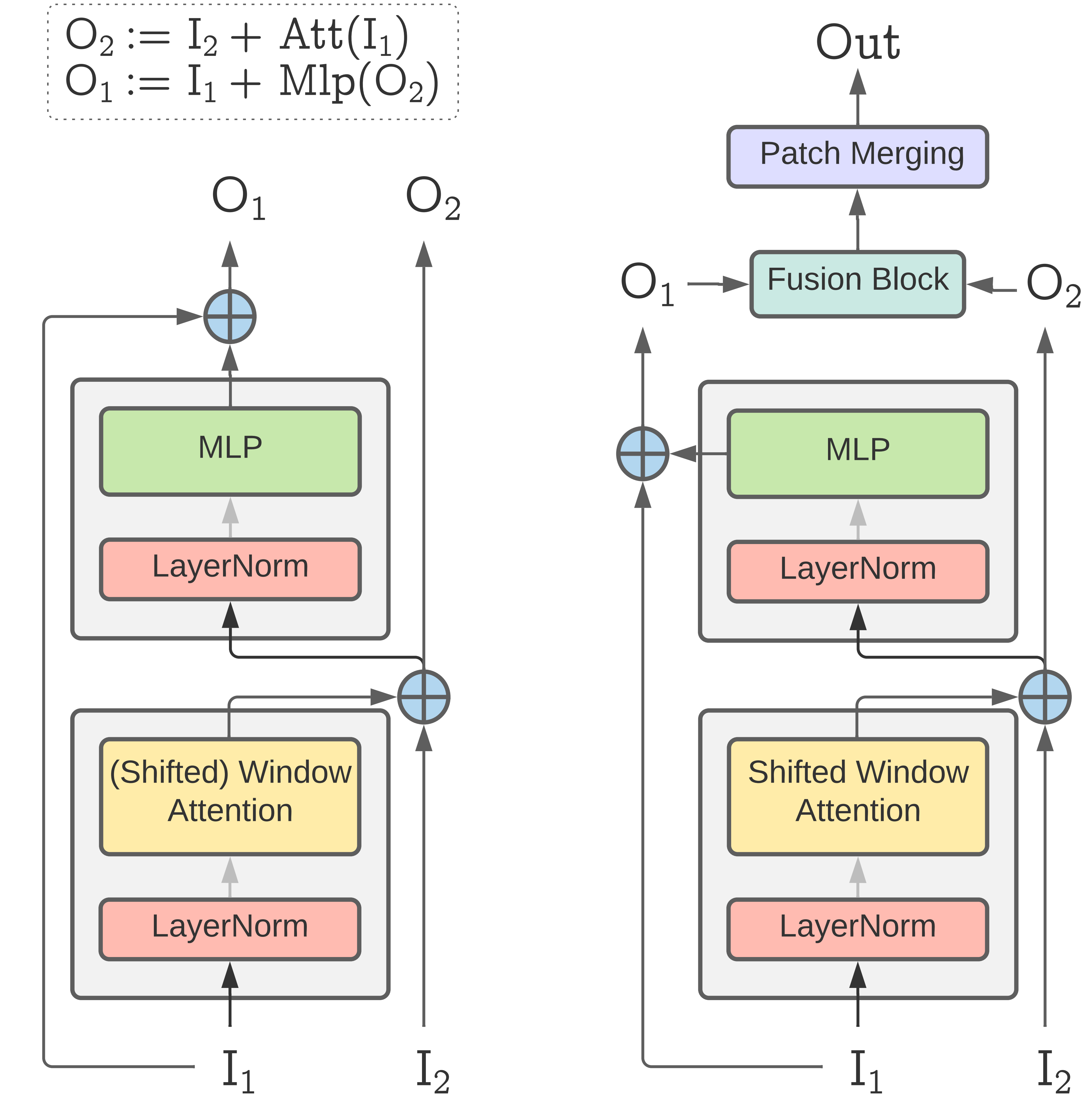
- Once we have the activations, we can do both steps at once
- Speeds up backprop significantly!

x	Forward pass of block **x**
x	Backward pass with gradient updates of block **x**
x	Activation recomputation in reversible backprop of block **x**

Reversible MViT, Swin



Reversible MViT

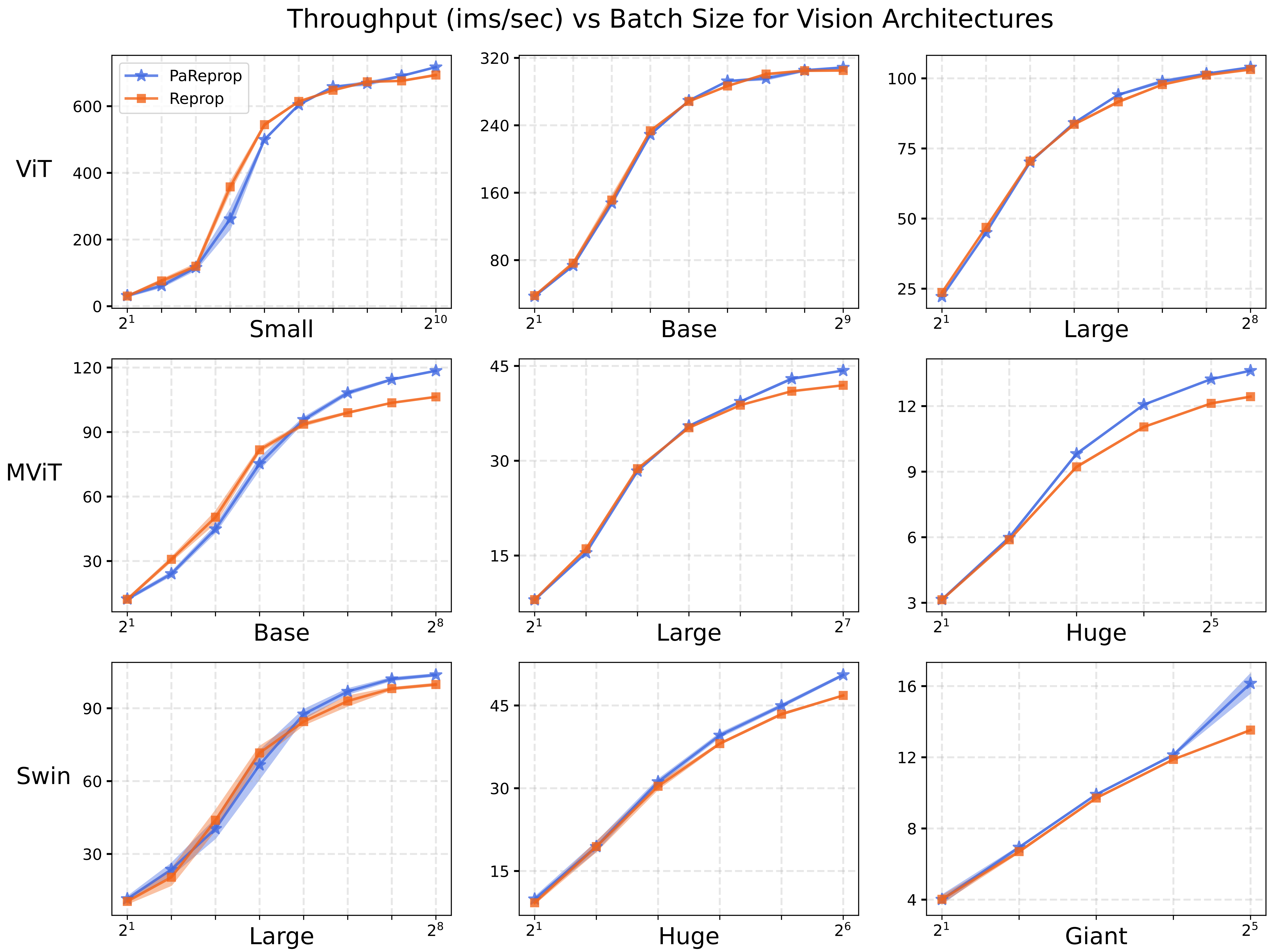


(a) Swin Transformer Block

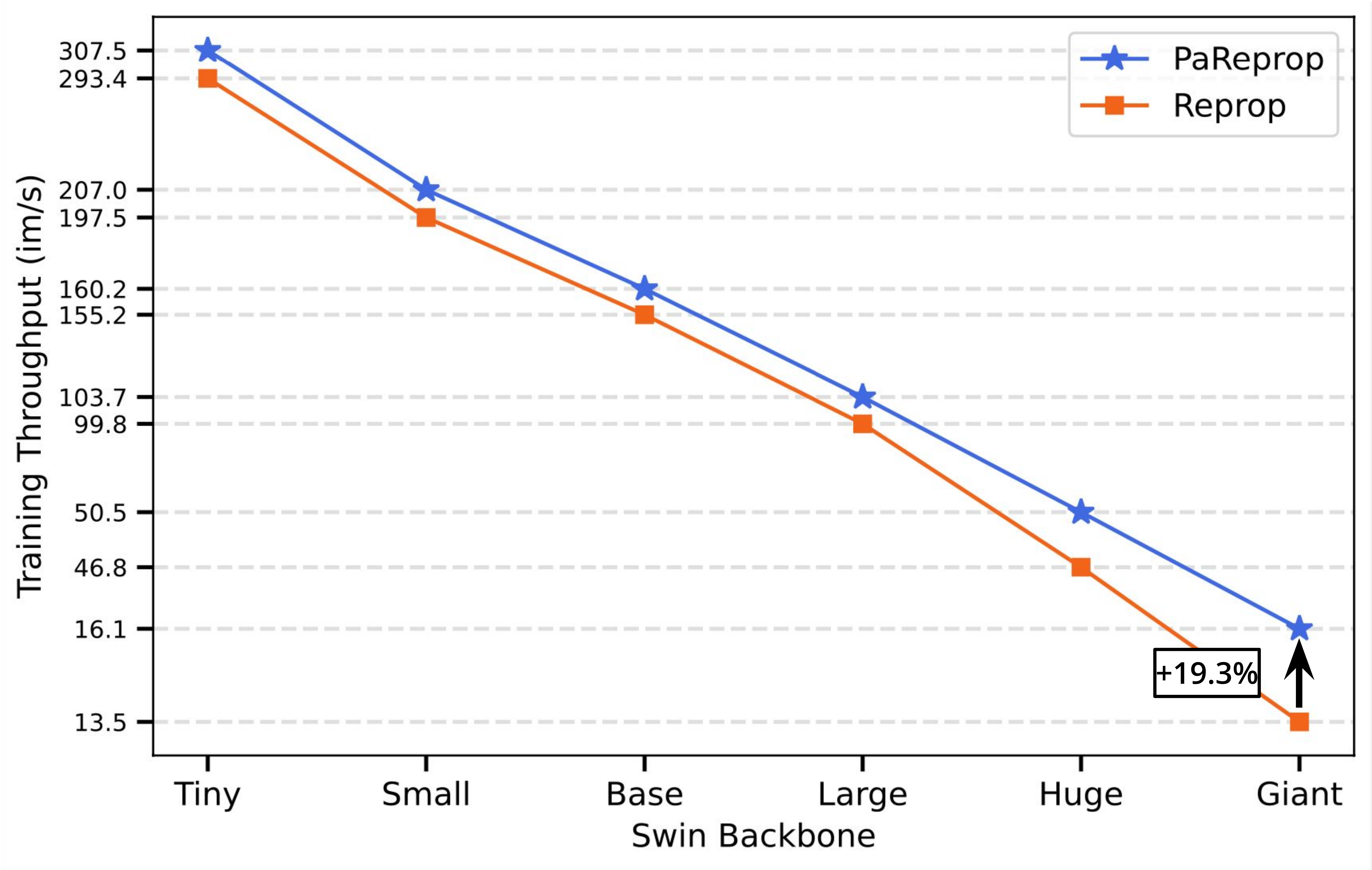
(b) Swin Downsample Block

Reversible Swin

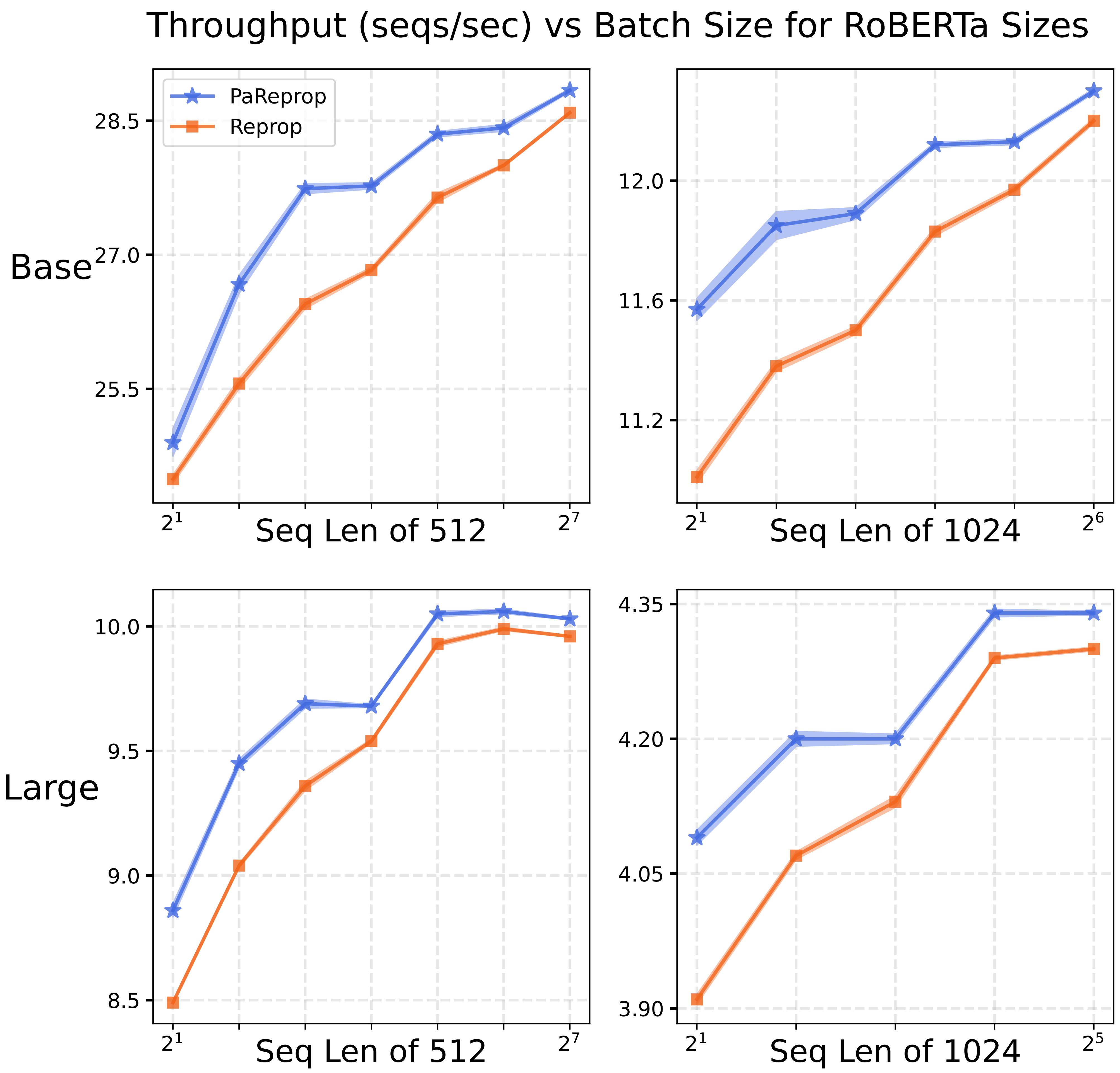
PaReprop improves throughput on vision archs



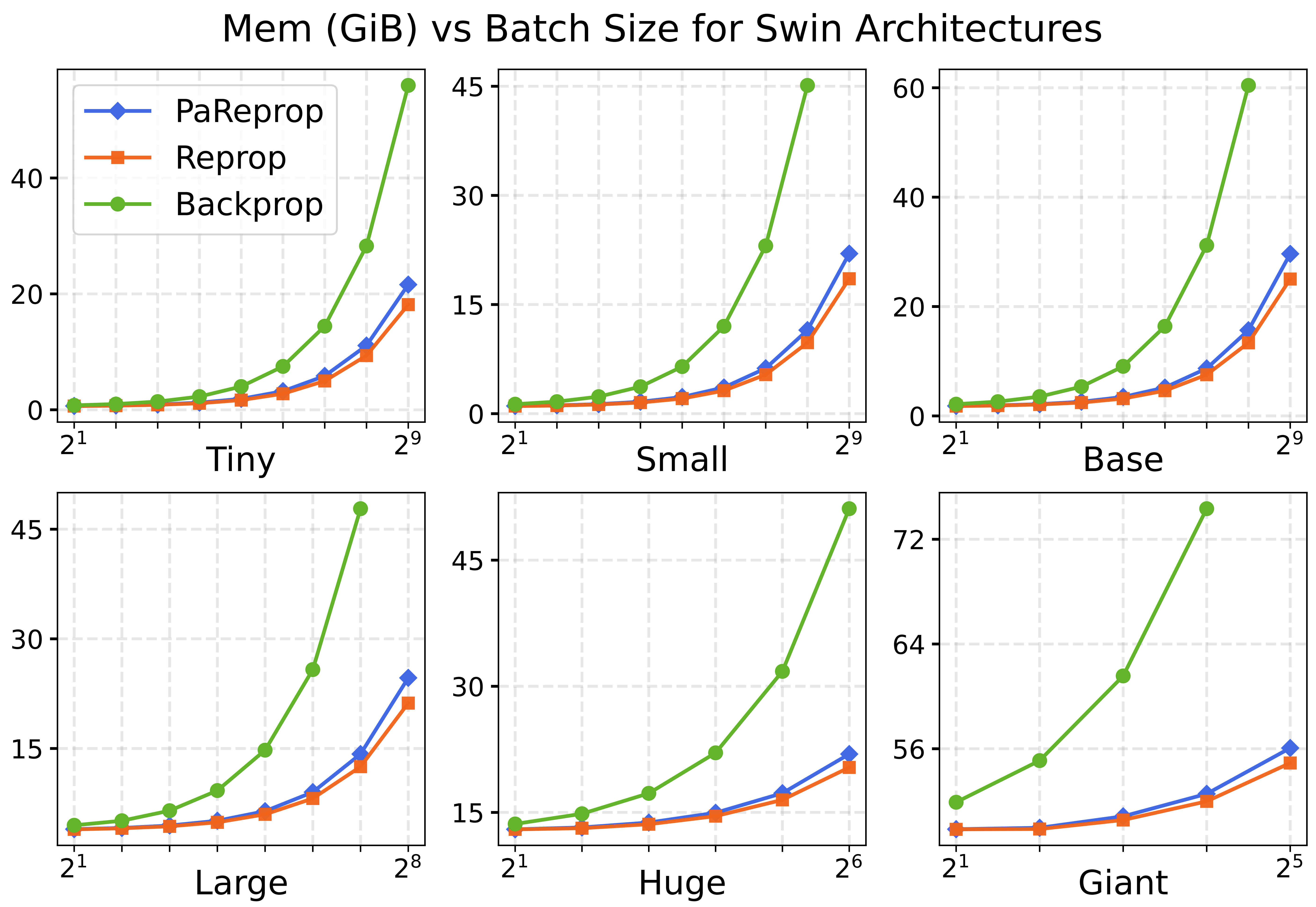
Large gains for Hierarchical Vision Transformers



Good for Language Models too!



Memory used for PaReprop is negligible



Conclusion

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

- Reversible architectures offer extremely memory-efficient training at the cost of some re-computations
- PaReprop essentially negates them by parallelizing the backward pass
- In practice, PaReprop performs on par or better than Reprop, even reaching up to 20% speedups in throughput (25% theoretical max)
  - Very good with mixed architectures (like hierarchical ViTs)
- Check out our poster! Website: www.tylerzhu.com/pareprop

Project Page