PaReprop: Fast Parallelized Reversible Backpropagation

Tyler Zhu* Karttikeya Mangalam*

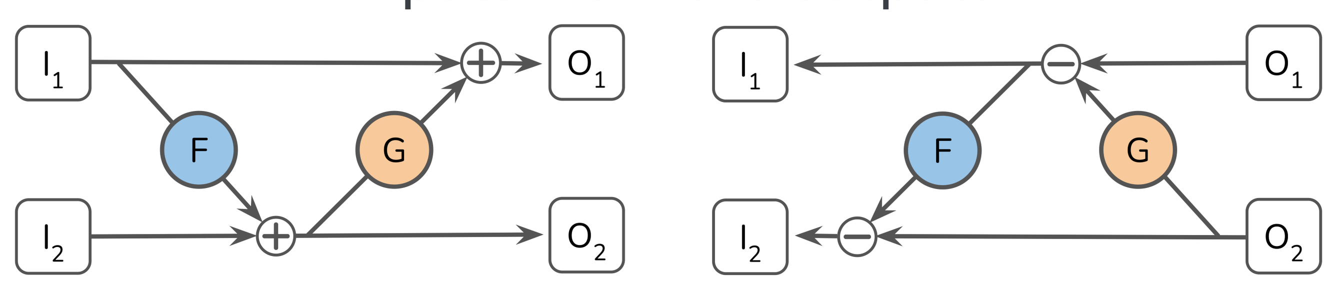
UC Berkeley

BAIR BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

I. Background: The Reversible Transformation

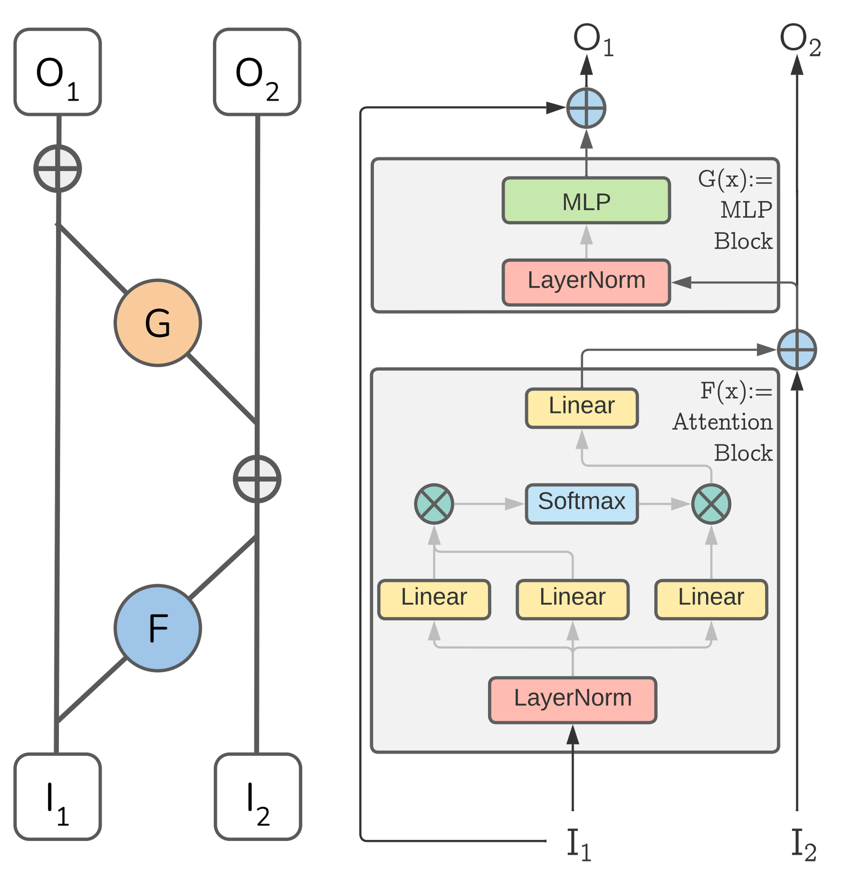
The Reversible Transformation analytically calculates the inputs from the outputs



$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \xrightarrow[T]{} \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} I_1 + G(I_2 + F(I_1)) \\ I_2 + F(I_1) \end{bmatrix} = \mathbf{O}$$

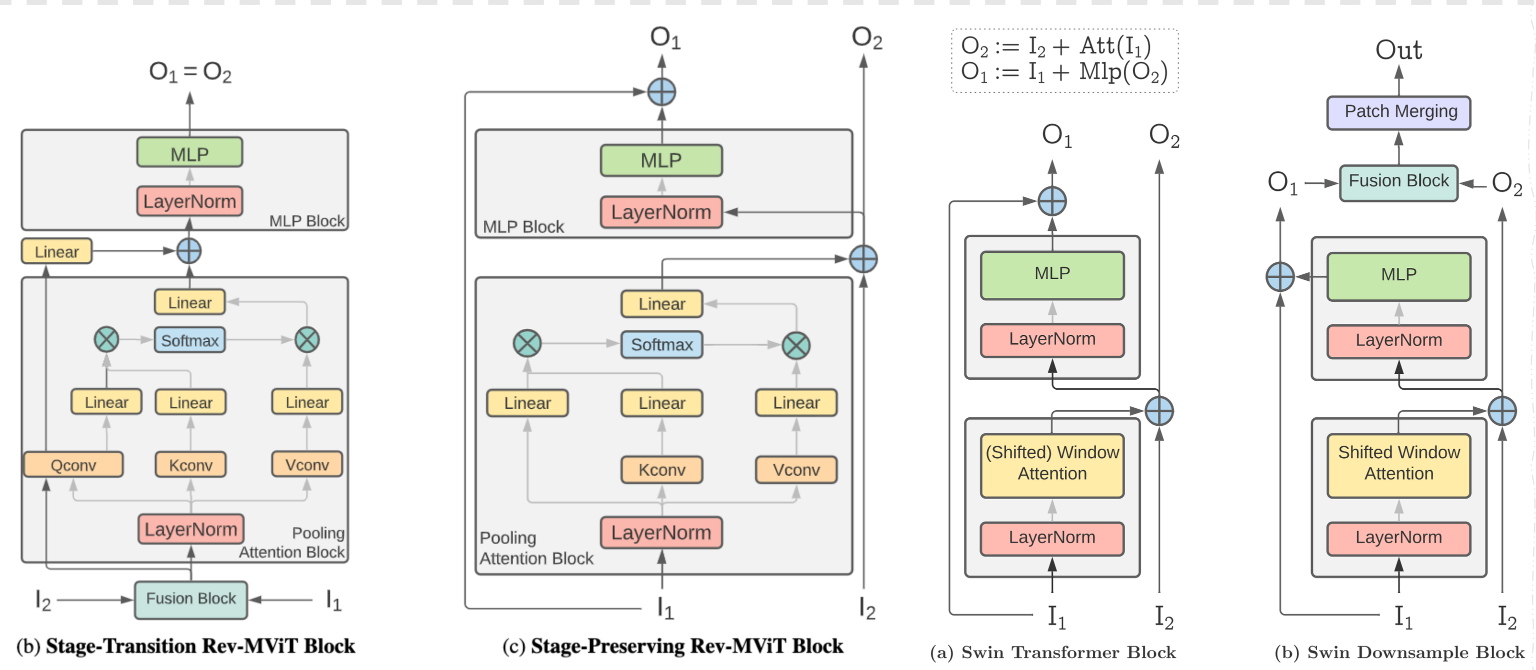
This allows intermediate activation recomputation in backward pass freeing the memory used for activation caching in forward.

II. Application: Reversible Vision Transformers



Rev-ViT [1] sets **F(x)** to Attention Block and **G(x)** to MLP Block of ViTs

- No performance drop
- Increases training throughput by up to 2.3 times!
- Same params, FLOPs, yet 15.5x smaller per image memory

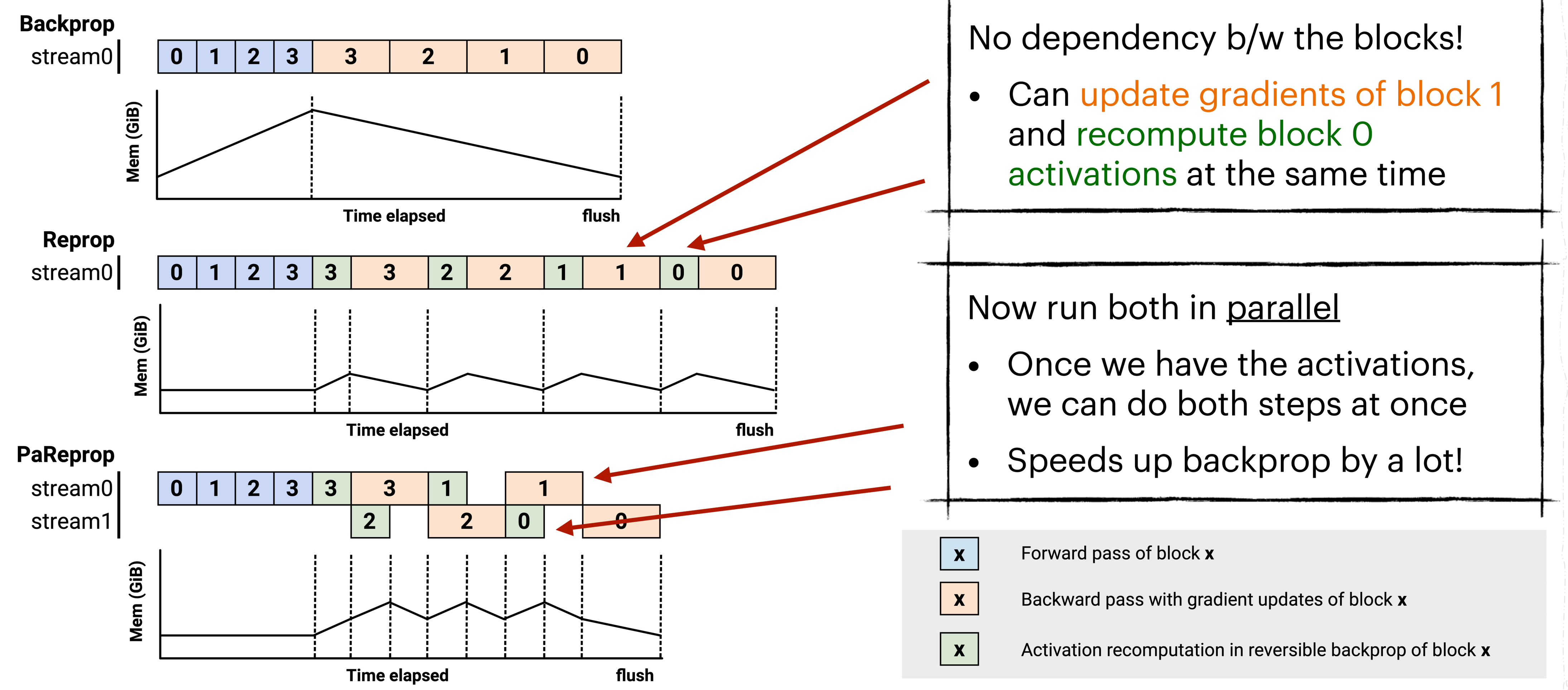


Reversible MViT Reversible Swin

Can also adapt hierarchical architectures like MViT (left) and Swin Transformer (right) w/ stage-transition blocks.

- Performance matches original; reversible transformers are a general, memory-efficient method for training.

III. Improving the Backpropagation



No dependency b/w the blocks!

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

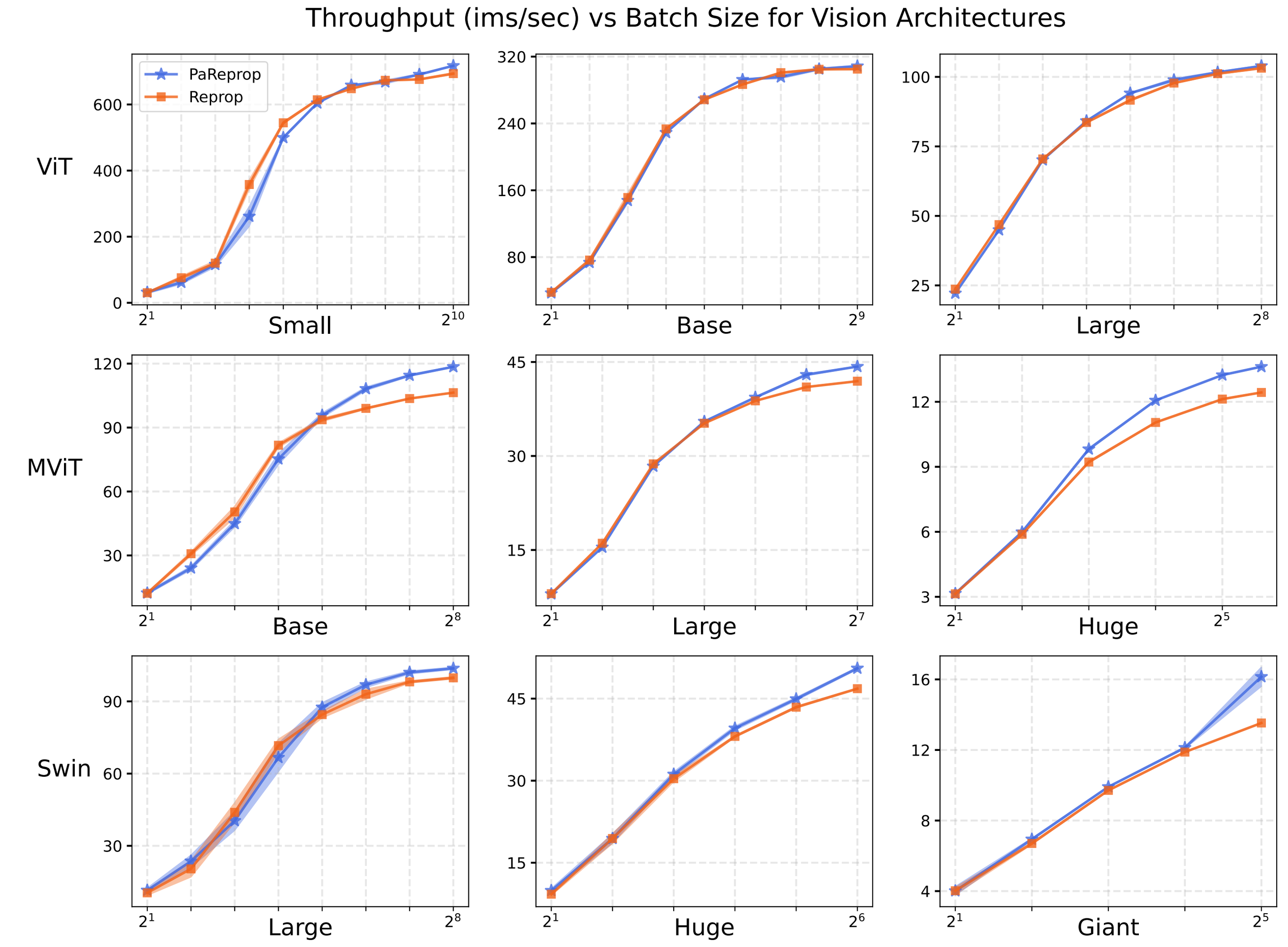
Now run both in parallel

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

Reversible backpropagation (Reprop) is done sequentially; not necessary! Using PyTorch CUDA streams, we can run both together at once and in theory be as fast as normal backprop (~25% max theoretical speedup possible).

IV. Better Throughput Across Vision & Language Transformers

Throughput (ims/sec) vs Batch Size for Vision Architectures



PaReprop boosts throughput up to 20% (almost at 25%)

Best with low-memory or huge, mixed models

Selected Previous Work

[1] Mangalam, K., Fan, H., Li, Y., Wu, C., Xiong, B., Feichtenhofer, C., & Malik, J. “Reversible vision transformers." *CVPR* 2022. *(+ Figure Credits)*

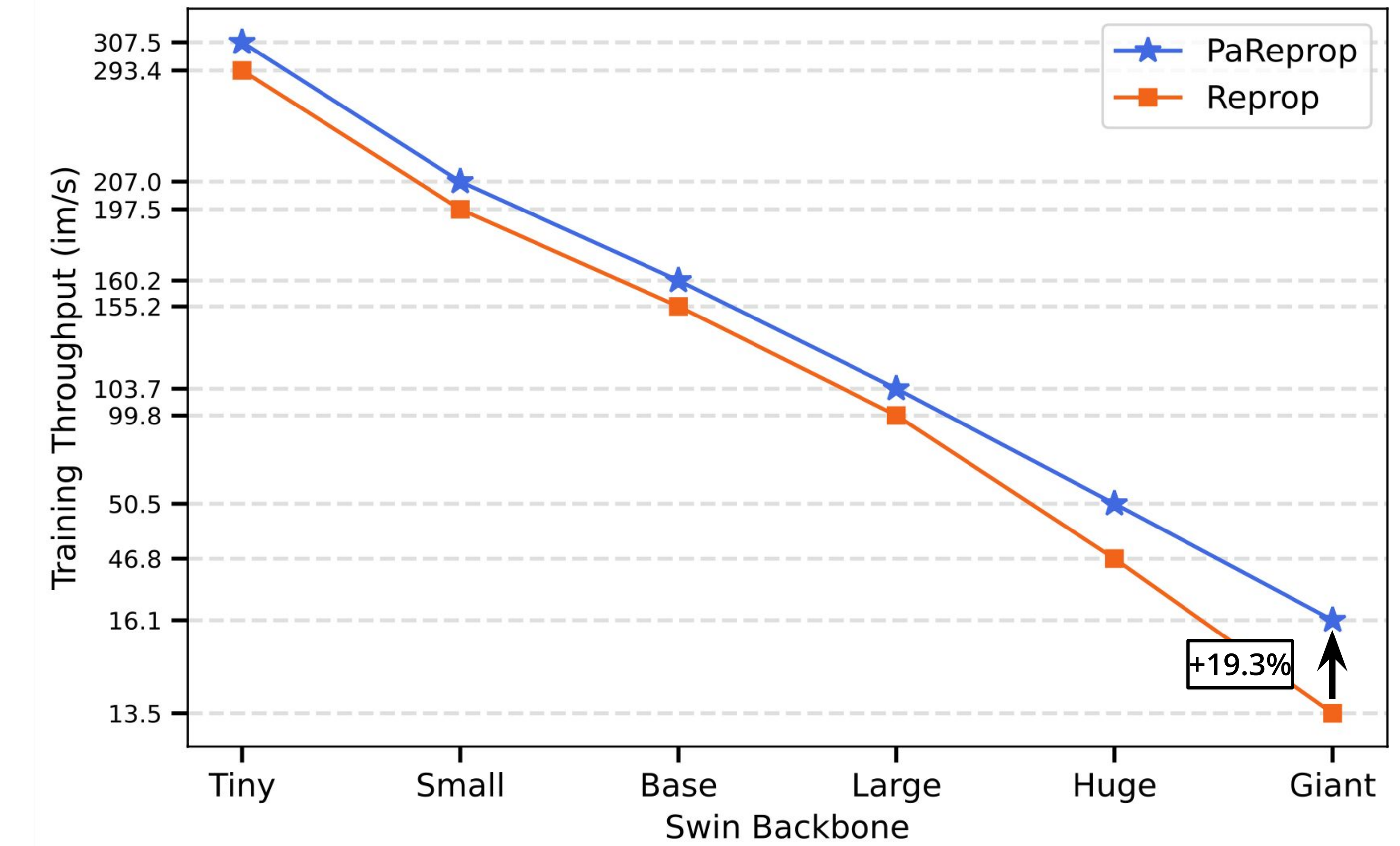
[2] Gomez, Aidan N., et al. "The reversible residual network: Backpropagation without storing activations." *NIPS 2017*

[3] Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., & Feichtenhofer, C. (2021). Multiscale vision transformers. *CVPR* 2022

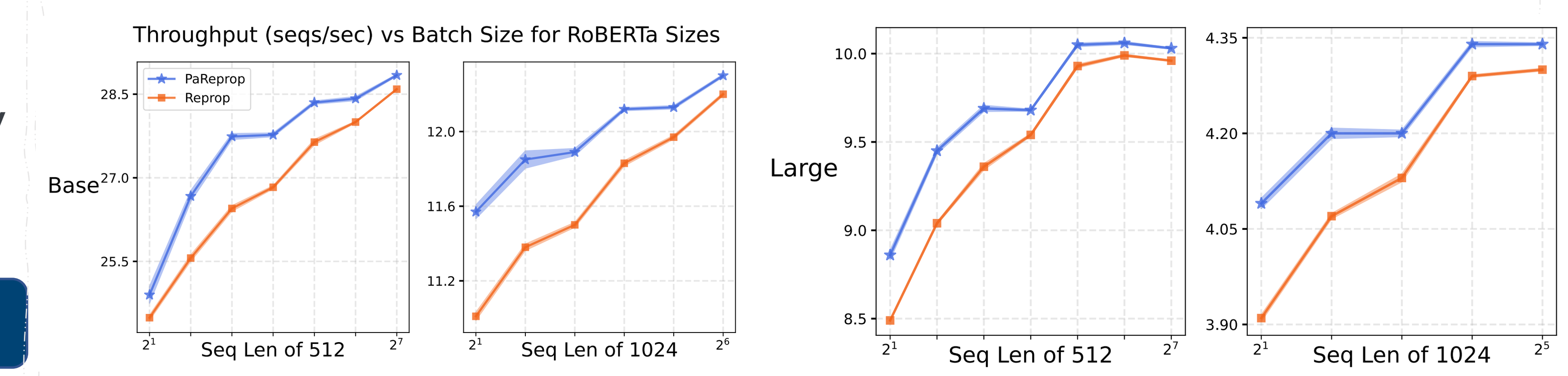
[4] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *ICLR 2020.*

Corresponding email: tyler.zhu@berkeley.edu (Scan QR code for Project Page)

V. Great for Hierarchical & Language Models



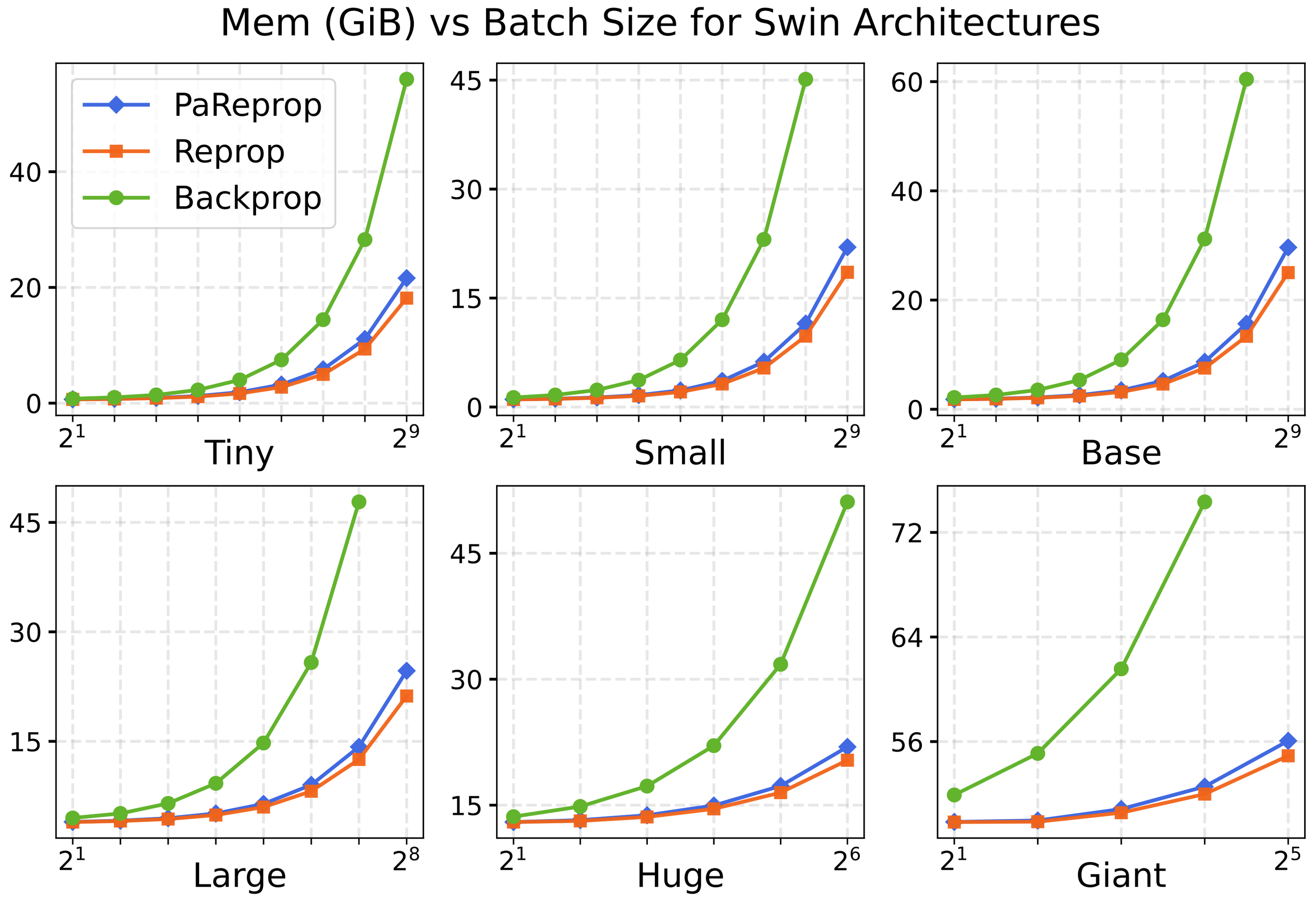
PaReprop does especially well with hierarchical models due to their non-homogenous composition of operations.



PaReprop also improves reversible backpropagation on language transformers (our proposed Rev-RoBERTa).

VI. Negligible Memory Cost vs. Savings

Mem (GiB) vs Batch Size for Swin Architectures



- Reprop & PaReprop are memory-efficient vs. Backprop
- Memory cost of PaReprop is small vs. the overall savings