SAND2021-2166C

Model Parallelism with Spatial Decomposition of Volumetric Data for Deep Learning

Gary Saavedra
Eric C. Cyr
Sandia National Laboratories

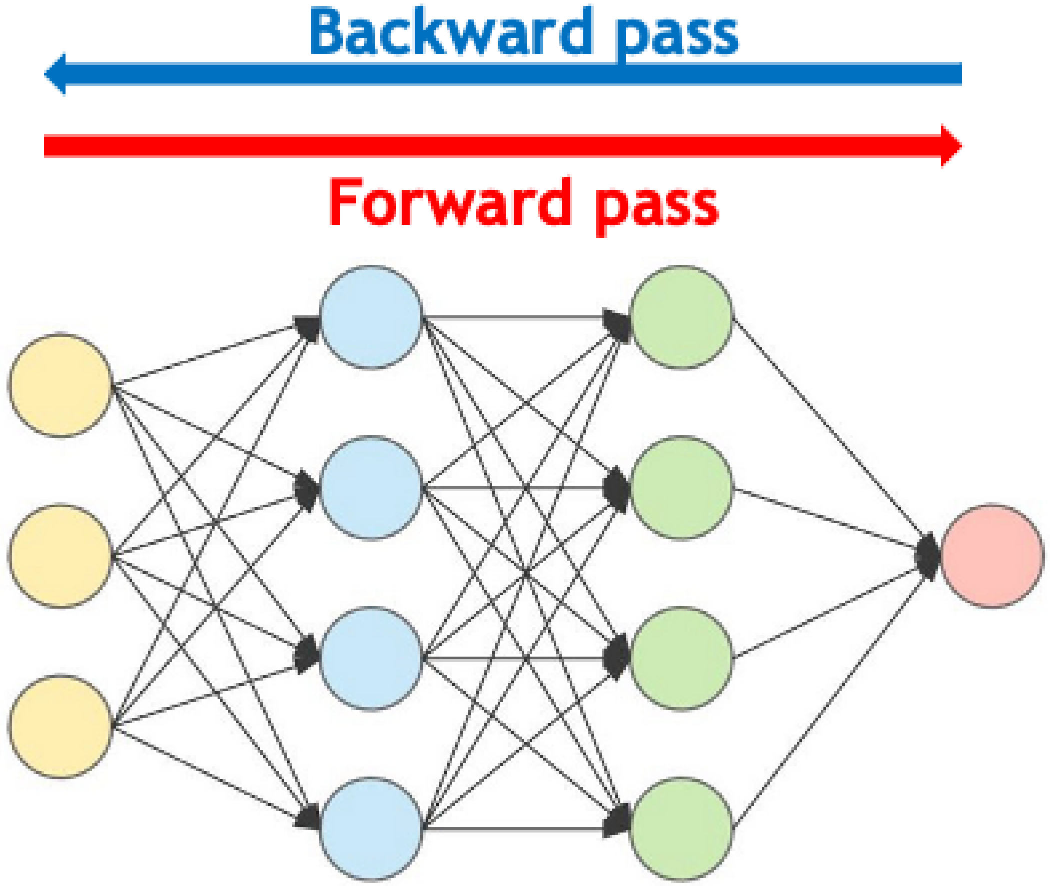
Jacob Schroder
University of New Mexico

Russell Hewett
Virginia Tech

SAND No:

Deep Learning and Time Parallelism

- More layers can improve performance
- Several groups have shown very deep networks offer improvement
  - GoogLeNet has 22 layers[1]
  - Huang et al show improvement on Cifar-10 data using up to 1200 layers[2]
  - Recurrent neural networks
- Very deep networks may be ignored due to training limitations.



Feedforward $$\mathbf{h}_{t+1} = f(\mathbf{h}_t, \theta_t)$$

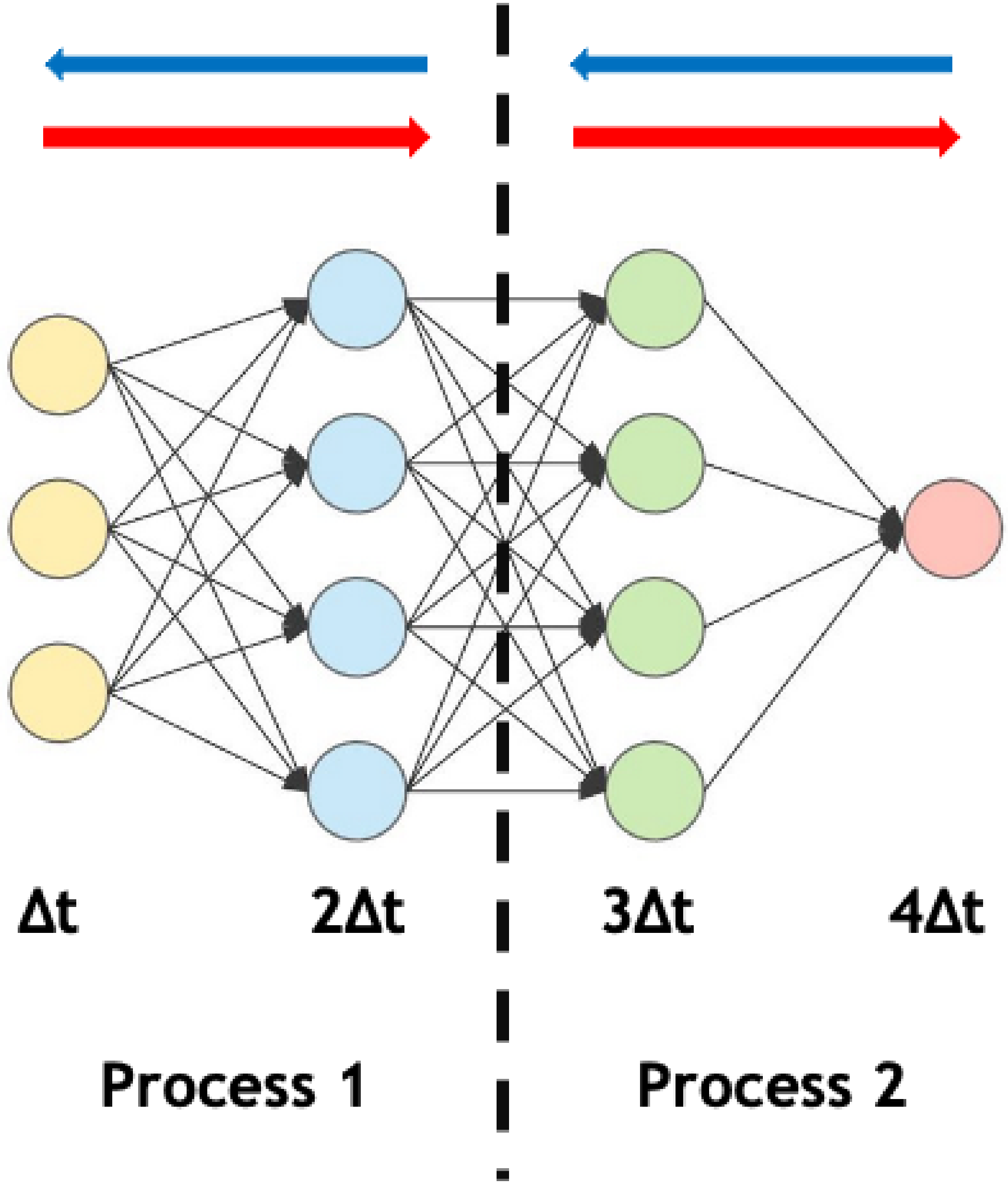
ResNet $$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

1) Szegedy et al. Going Deeper with Convolutions. 2015.
2) Huang et al. Deep Networks with Stochastic Depth. 2016.
3) kdnuggets.com

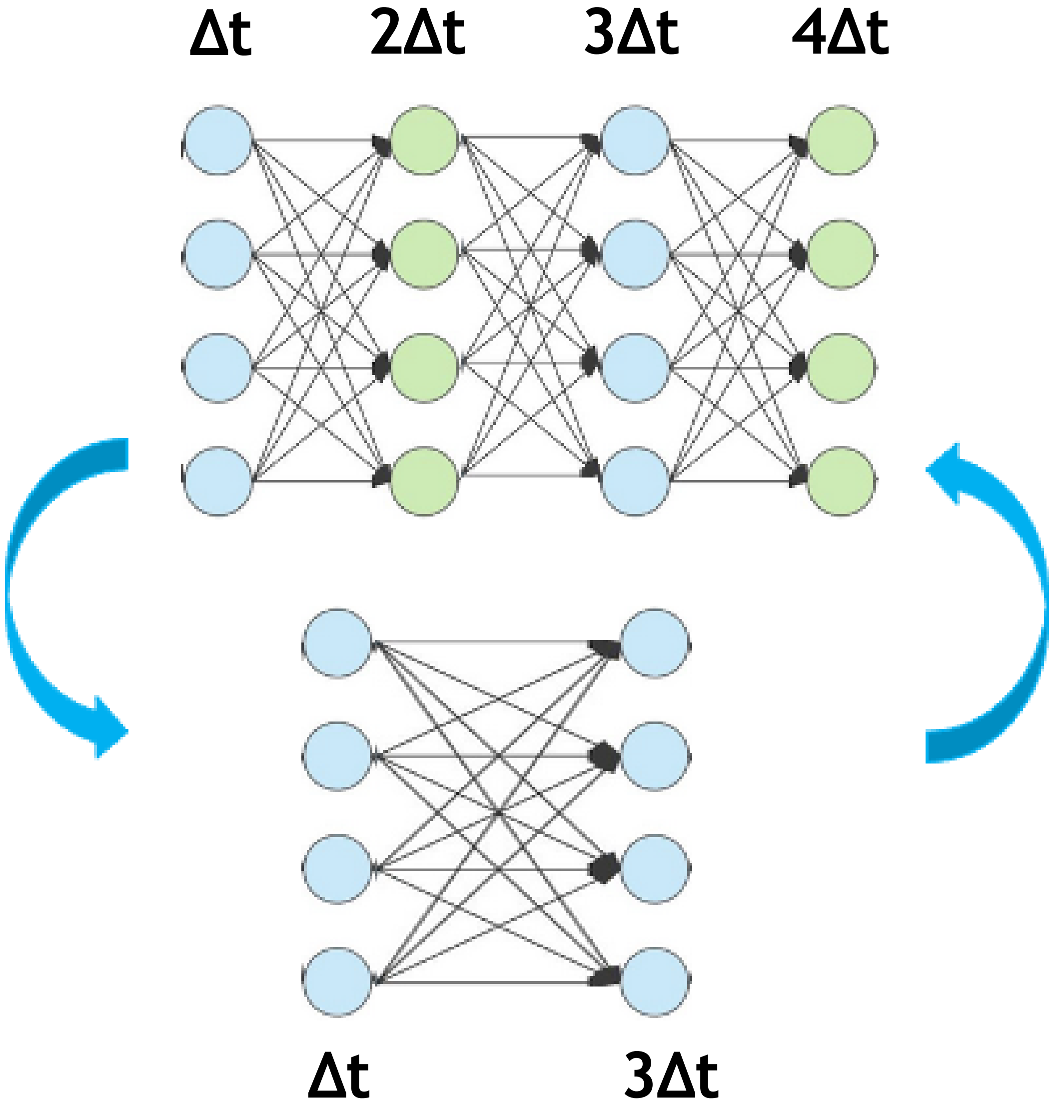
Deep Learning and Time Parallelism

- Separate groups of layers between processors

- Neural networks are not naturally parallelized in time



Time decomposition using multigrid

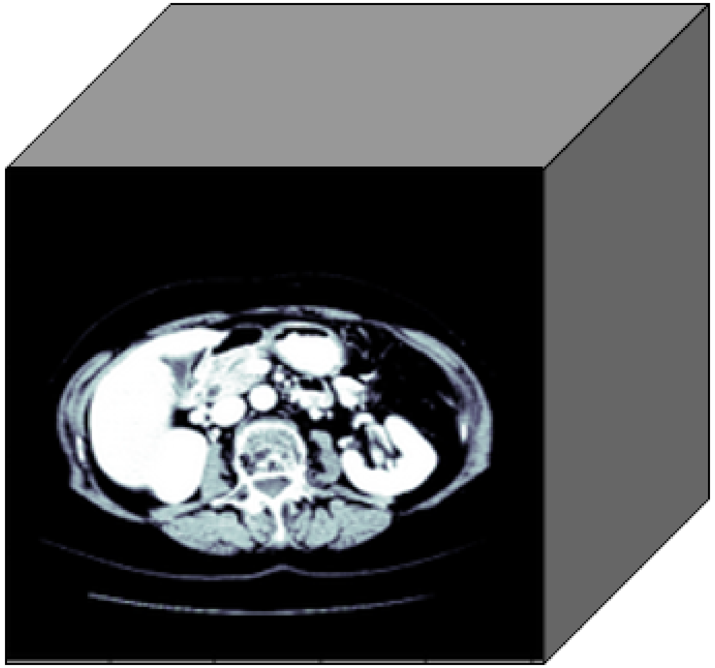


Coarsening: reduce number of layers

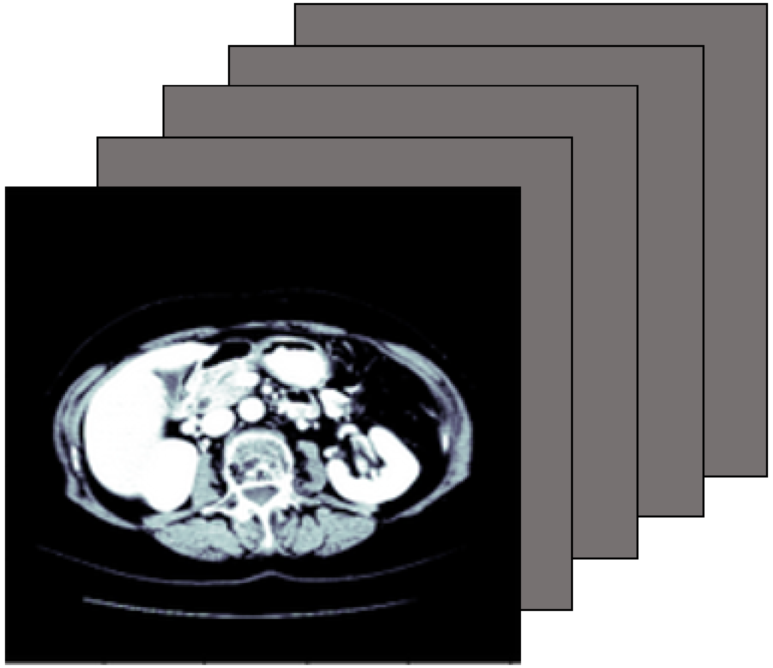
Refinement: use coarse output to inform

Multigrid improves time decomposition approximations

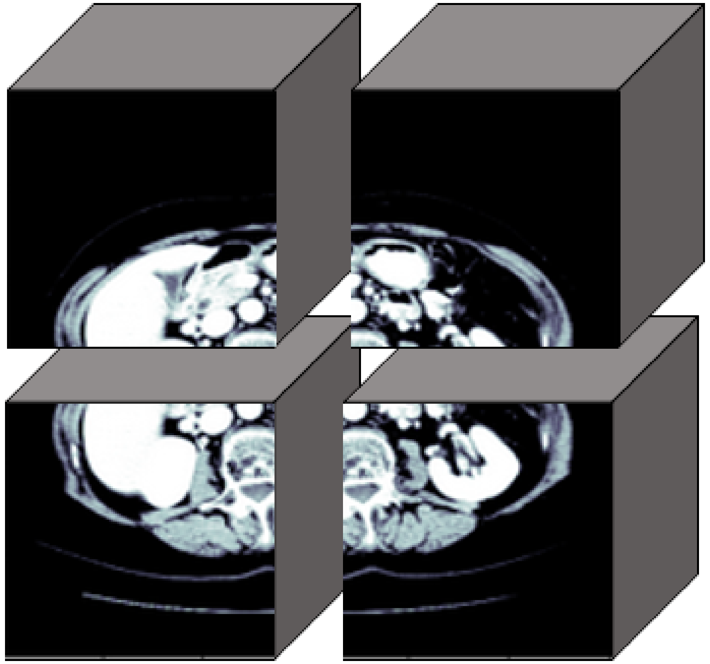
Deep Learning and Spatial Decomposition



Original



2D slices

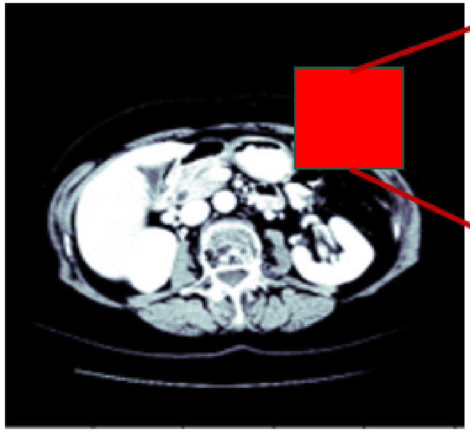


3D decomposition

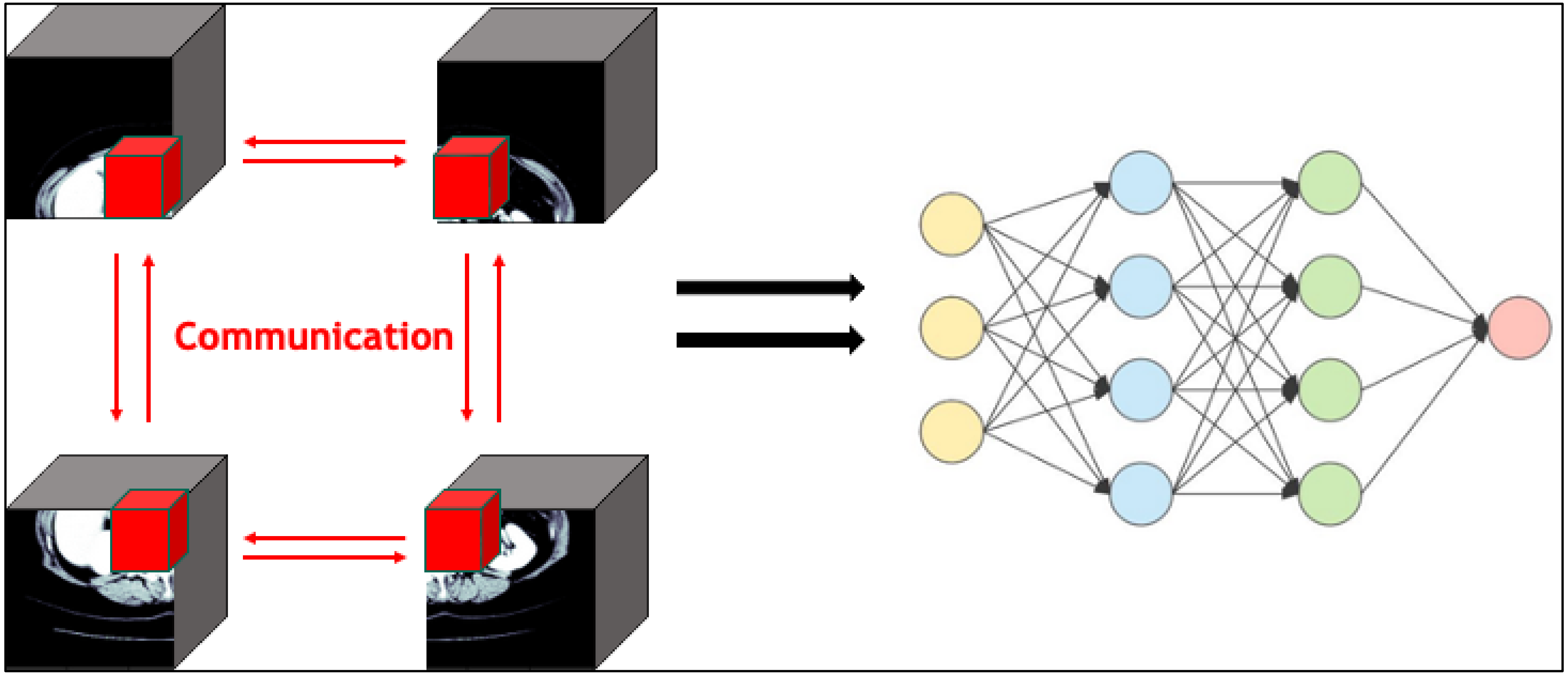
Medical image segmentation sees improvements when using 3D spatial decomposition.

High Resolution Medical Image Analysis with Spatial Partitioning. Hou et al. 2019

How spatial decomposition works



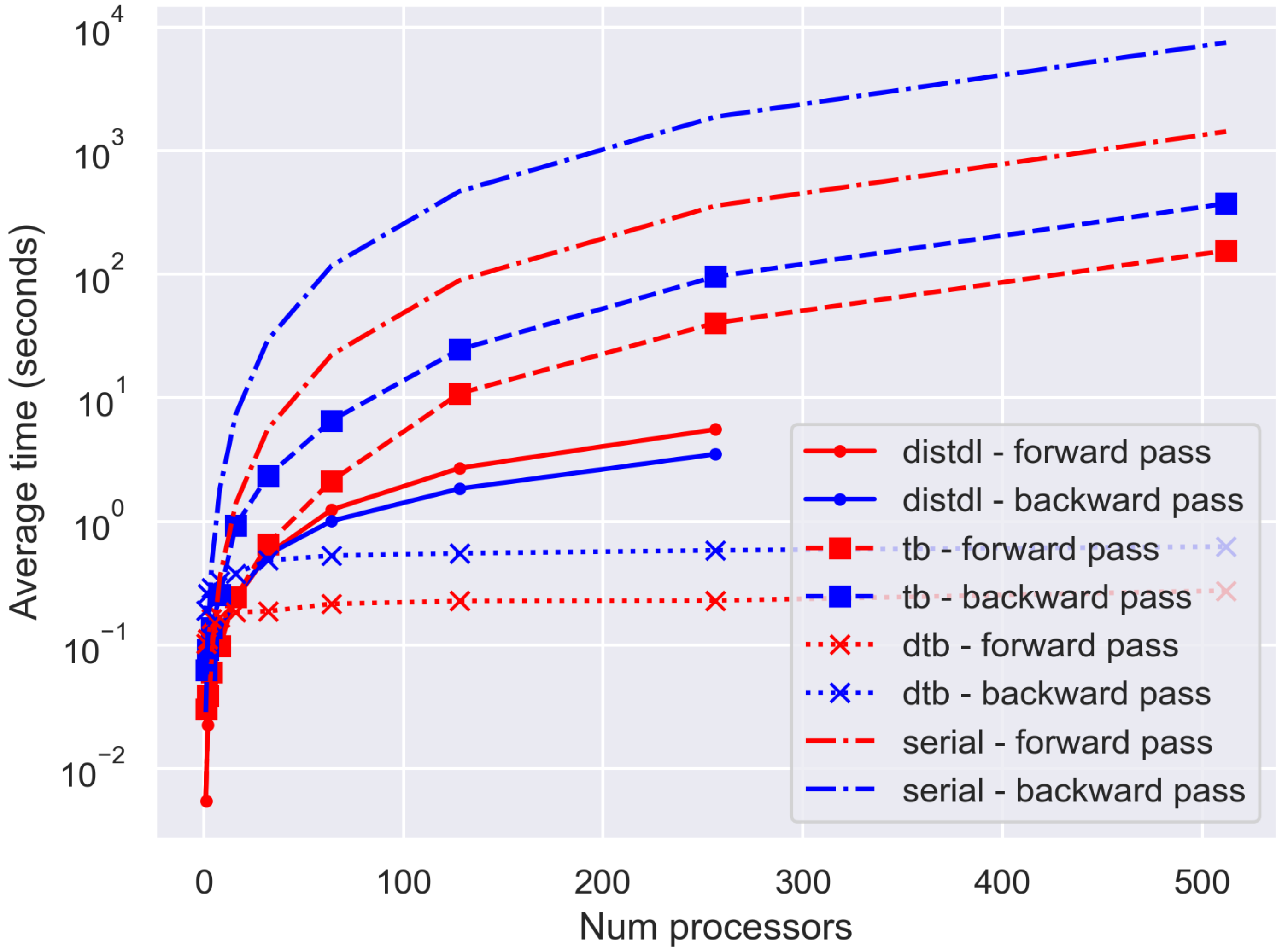
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	1	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$



Approach – combine spatial and time parallelism

- Combine two PyTorch frameworks: **DistDL** and **TorchBRAID**.
- The combo framework is currently called **DTB.**
- Compare performance of space-time parallelism with lone space or time parallelism
- Architecture
  - Processor - 2.1 GHz Intel Broadwell E5-2695 v4 : 2 sockets : 18 cores
  - RAM per node – 128 GB
  - 1.8 pFlops

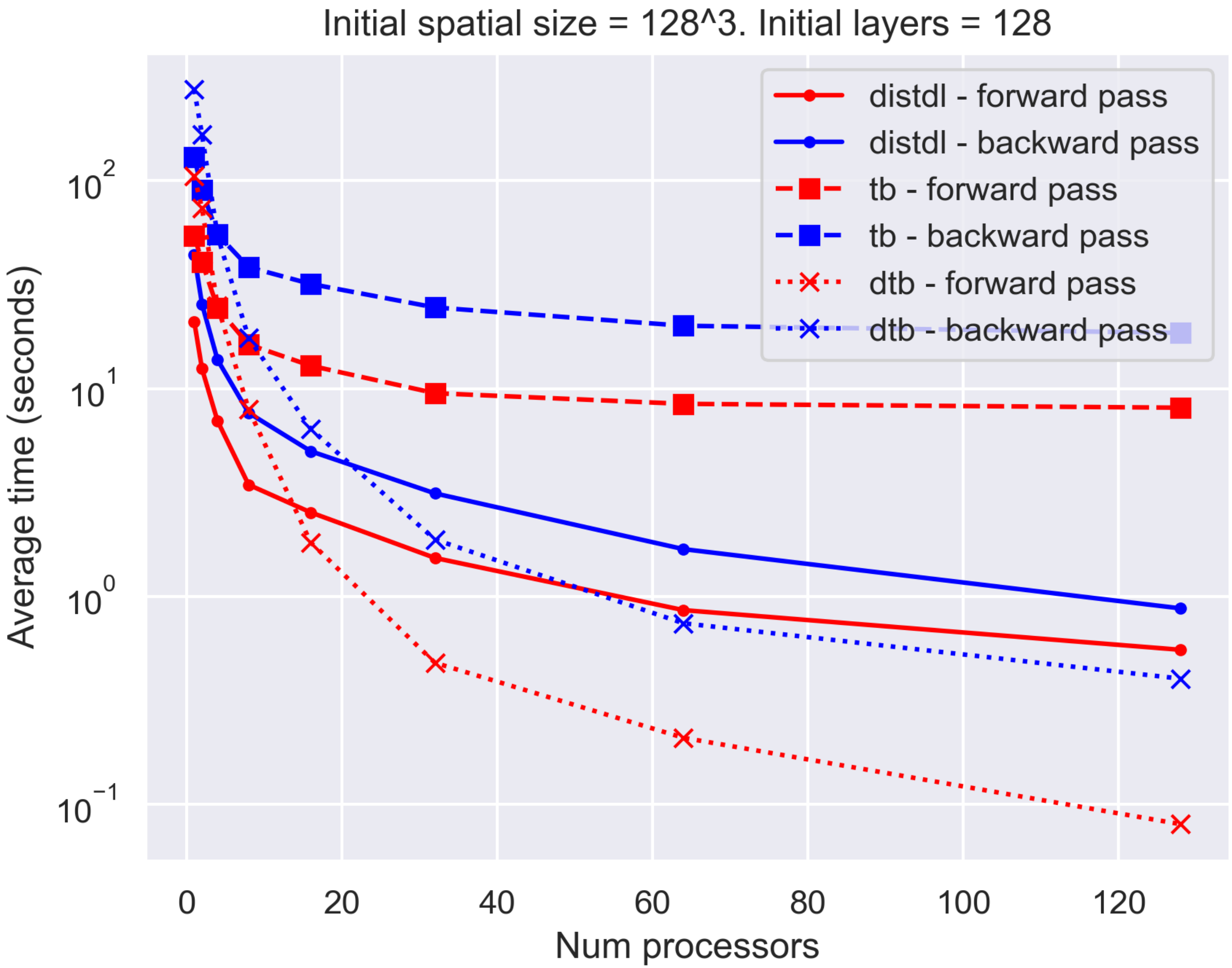
Results – Convolution layers



Problem Size per Process (Space, Layer)			
Procs	DistDL	TB	DTB
1	16^3, 8	16^3, 8	16^3, 8
2	16^3, 2x8	2x16^3, 8	16^3, 8
2^2	16^3, 2^2x8	2^2x16^3, 8	16^3, 8
2^3	16^3, 2^3x8	2^3x16^3, 8	16^3, 8
2^4	16^3, 2^4x8	2^4x16^3, 8	16^3, 8
…	…	…	…
2^N	16^3, 2^Nx8	2^Nx16^3, 8	16^3, 8

DTB exhibits weak scaling behavior while others slow down

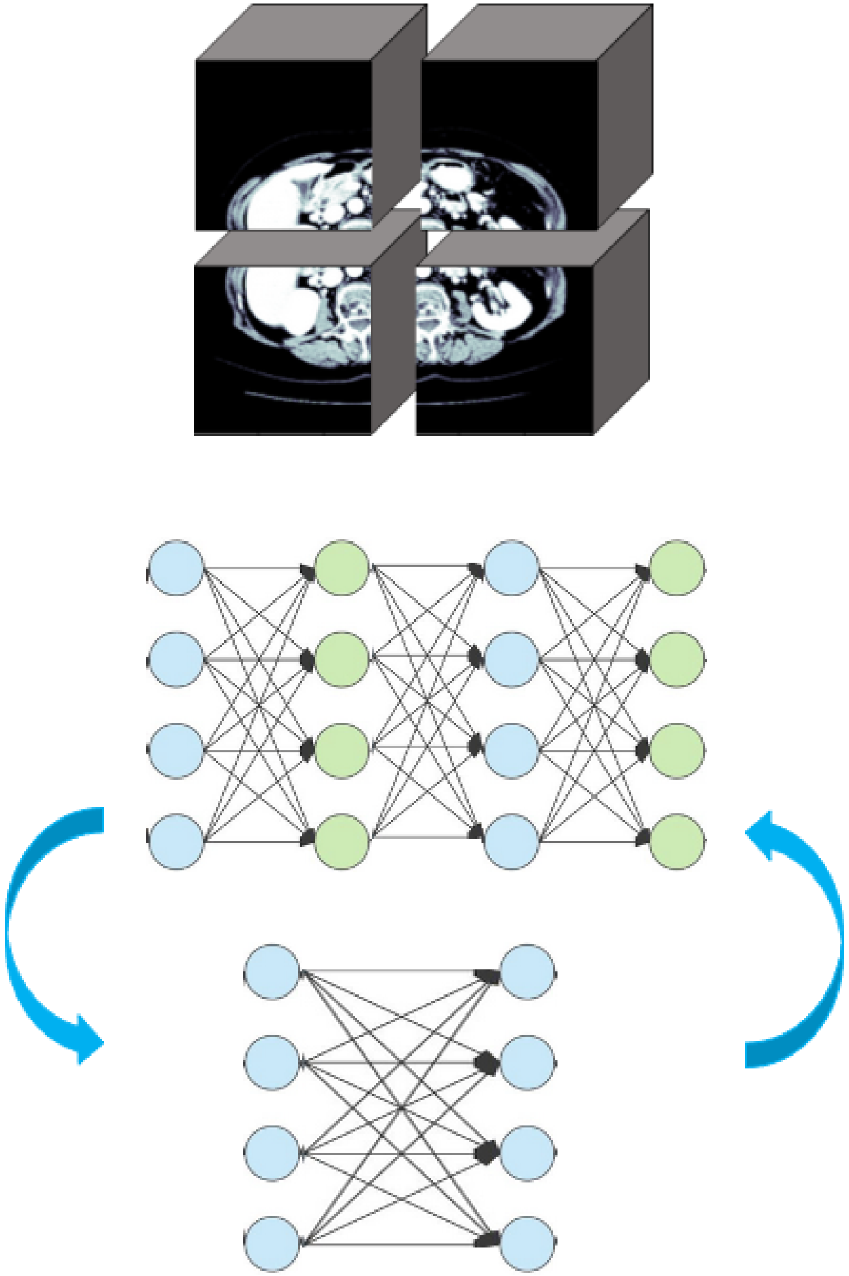
Results – Convolution layers



DTB is up to 2 orders of magnitude faster

Conclusion and Future Work

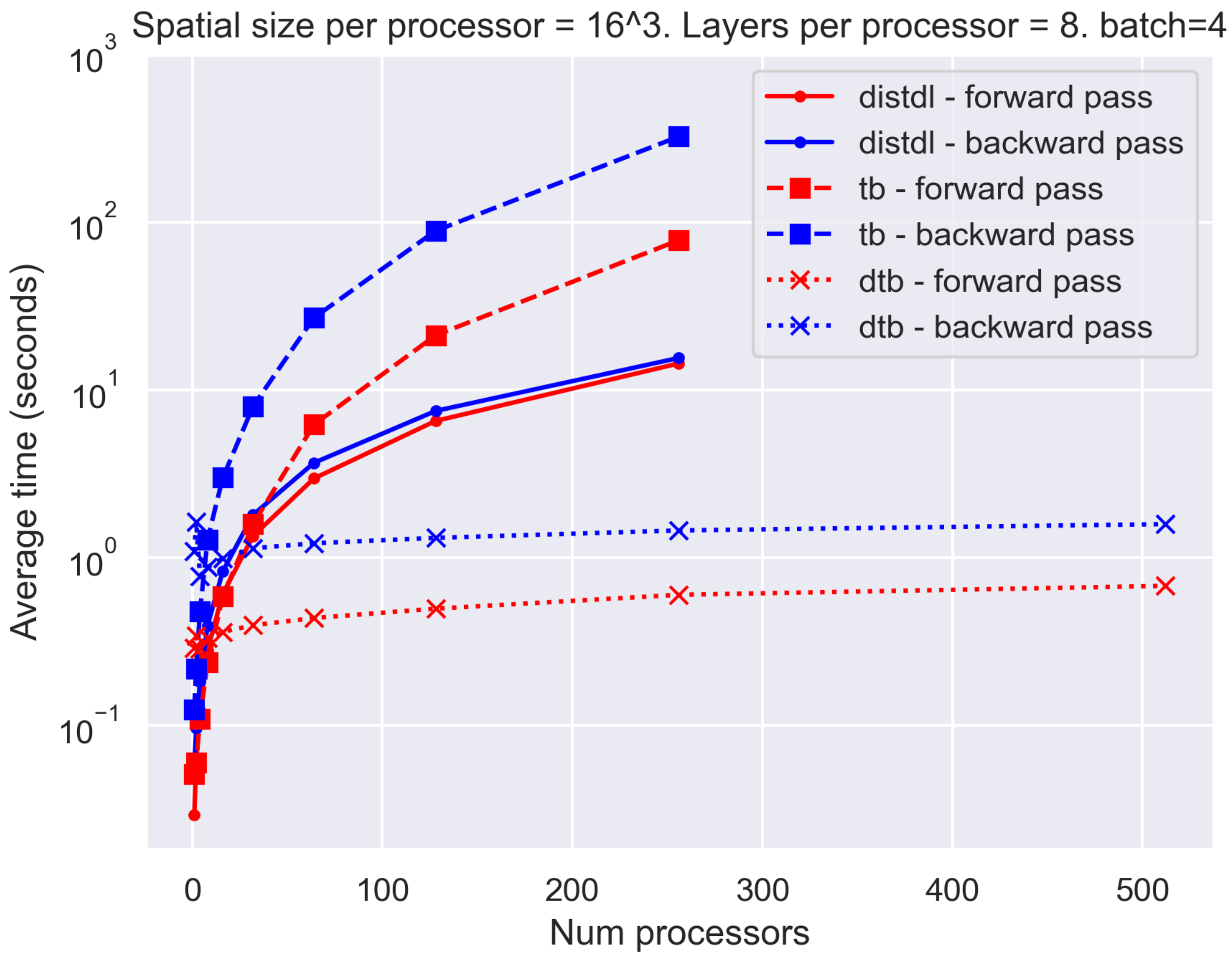
- **Important takeaways**
  - Spatial + time parallelism can offer significant speedups
  - The combination enables research into deeper networks for large image segmentation

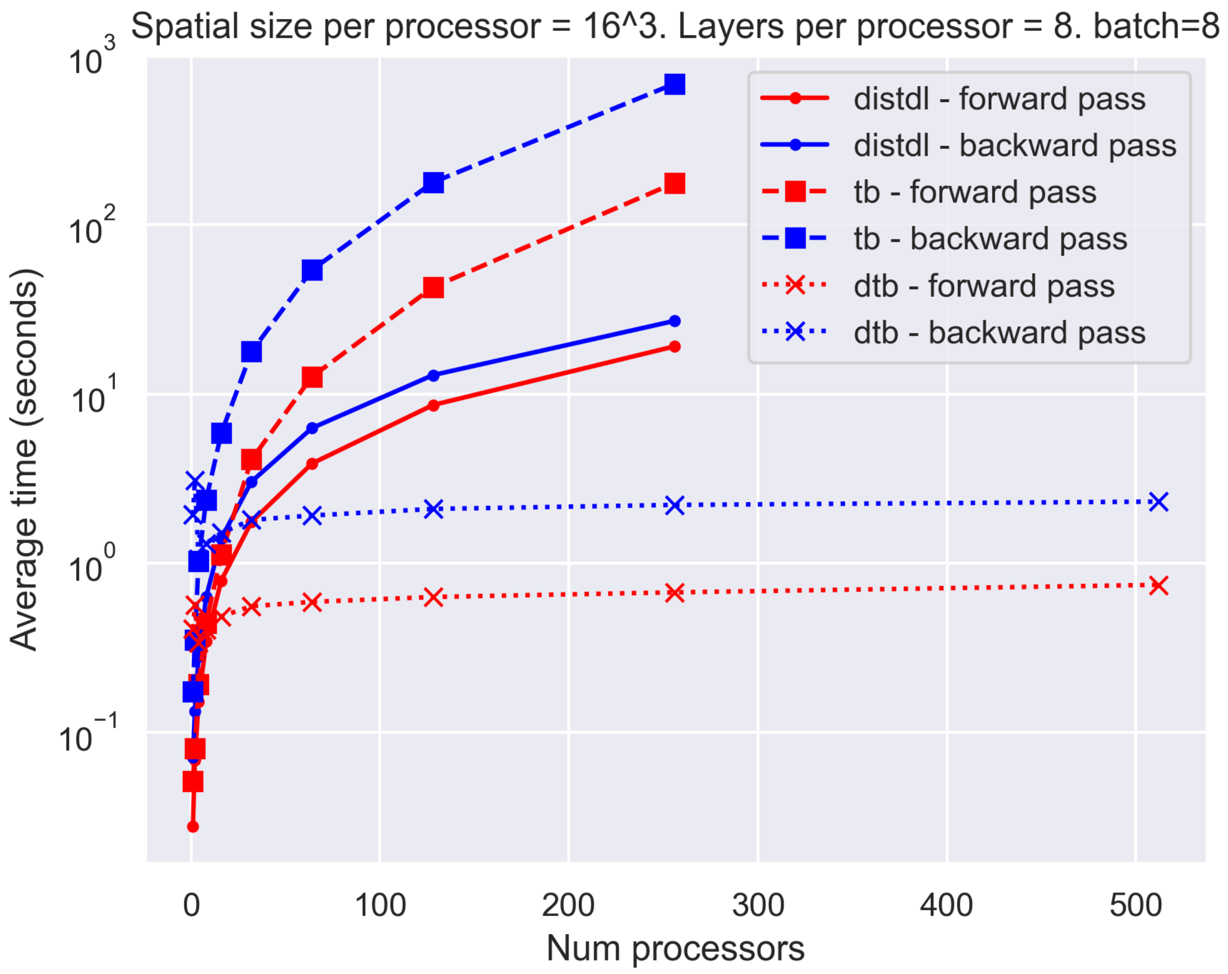


- **Future work**
  - Apply framework to real-world datasets
  - Can we integrate spatial multigrid for further speedups?

Backup Slides

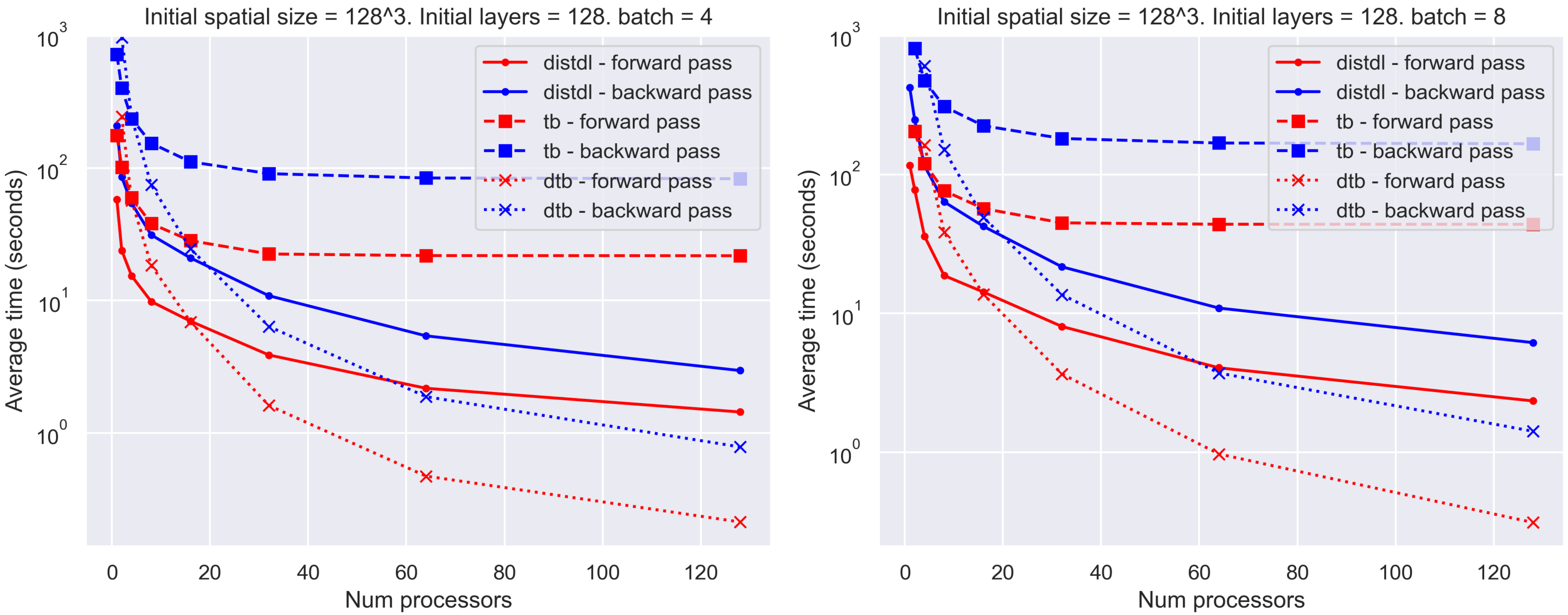
Results – Convolution followed by batch norm



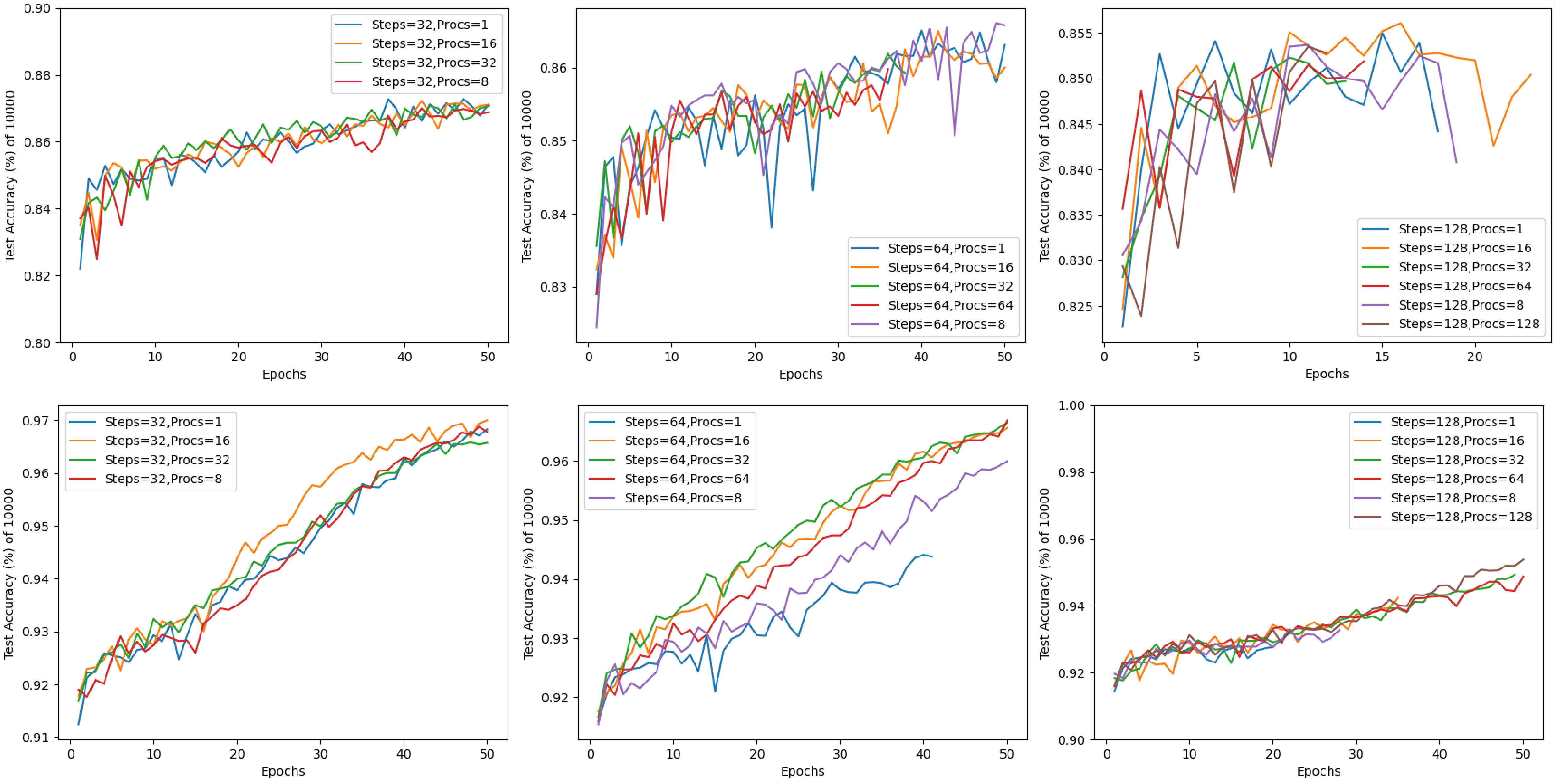


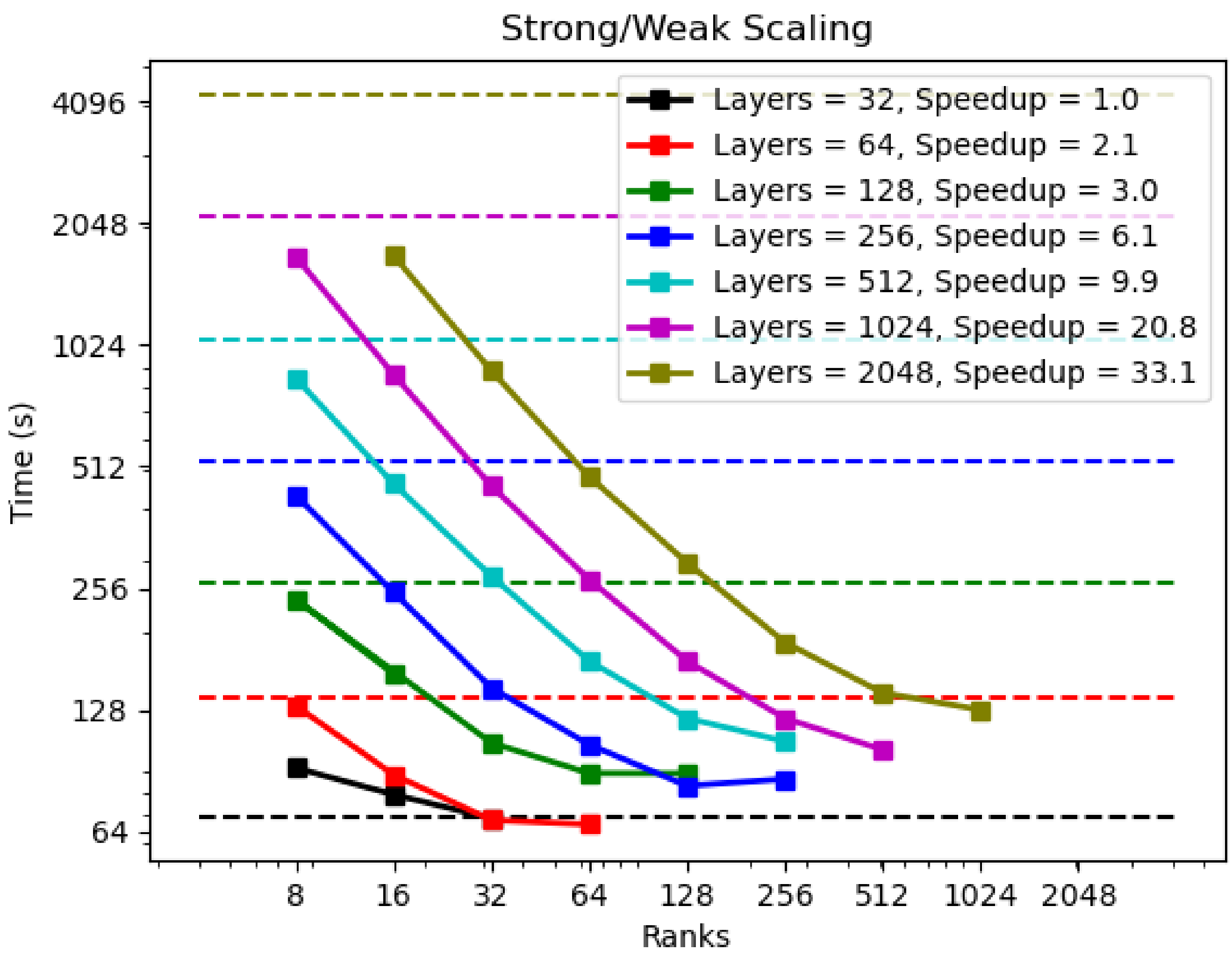
DTB exhibits weak scaling behavior while others slow down

Results – Convolution followed by batch norm



DTB is up to 2 orders of magnitude faster


Steps=32,Procs=1
Steps=32,Procs=16
Steps=32,Procs=32
Steps=32,Procs=8
Test Accuracy (%) of 10000
Epochs
Steps=64,Procs=1
Steps=64,Procs=16
Steps=64,Procs=32
Steps=64,Procs=64
Steps=64,Procs=8
Test Accuracy (%) of 10000
Epochs
Steps=128,Procs=1
Steps=128,Procs=16
Steps=128,Procs=32
Steps=128,Procs=64
Steps=128,Procs=8
Steps=128,Procs=128
Test Accuracy (%) of 10000
Epochs
Steps=32,Procs=1
Steps=32,Procs=16
Steps=32,Procs=32
Steps=32,Procs=8
Test Accuracy (%) of 10000
Epochs
Steps=64,Procs=1
Steps=64,Procs=16
Steps=64,Procs=32
Steps=64,Procs=64
Steps=64,Procs=8
Test Accuracy (%) of 10000
Epochs
Steps=128,Procs=1
Steps=128,Procs=16
Steps=128,Procs=32
Steps=128,Procs=64
Steps=128,Procs=8
Steps=128,Procs=128
Test Accuracy (%) of 10000
Epochs


Strong/Weak Scaling
Layers = 32, Speedup = 1.0
Layers = 64, Speedup = 2.1
Layers = 128, Speedup = 3.0
Layers = 256, Speedup = 6.1
Layers = 512, Speedup = 9.9
Layers = 1024, Speedup = 20.8
Layers = 2048, Speedup = 33.1
Time (s)
4096
2048
1024
512
256
128
64
8
16
32
64
128
256
512
1024
2048
Ranks