Toward Large-Scale Image Segmentation On Summit

Sudip K. Seal, Seung-Hwan Lim, Dali Wang,
Jacob Hinkle, Dalton Lunga, Aristeidis Tsaris

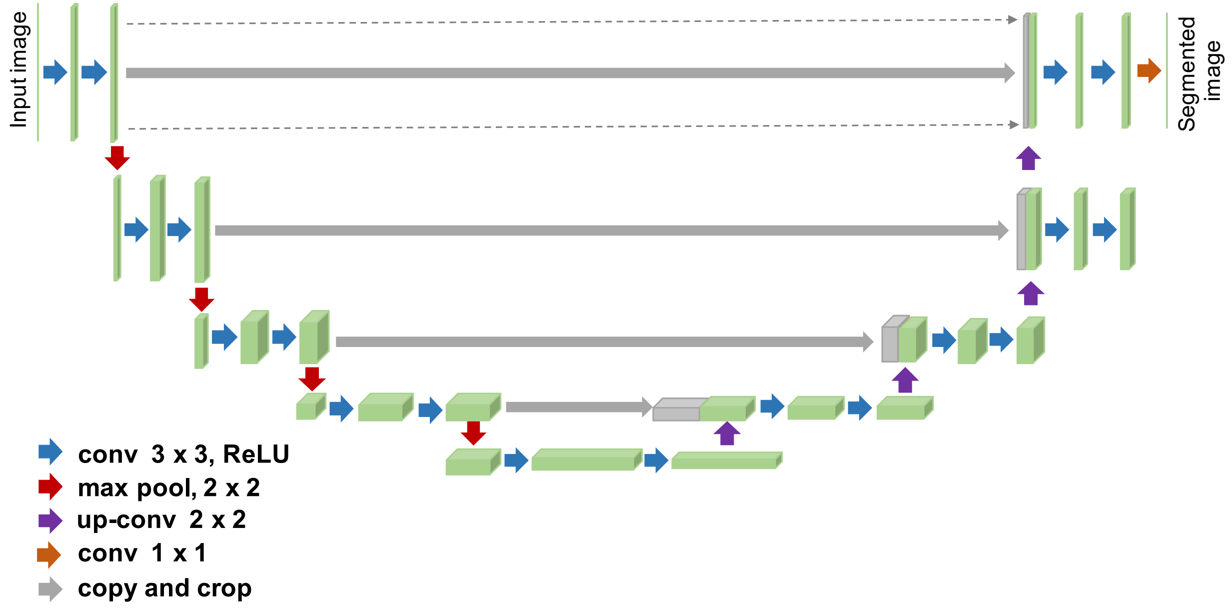
Oak Ridge National Laboratory, USA

August 19, 2020
International Conference on Parallel Processing
Alberta, Canada

Introduction
Semantic Segmentation of Images

- Given an image with $N \times N$ pixels and a set of k distinct classes, label each of the N^2 pixels with one of the k distinct classes.

- For example, given a 256×256 image of a car, road, buildings and people, a semantic segmentation of the image classifies each of the $256 \times 256 = 2^{16}$ pixels into one of $k = 4$ classes {car, road, building, people}.



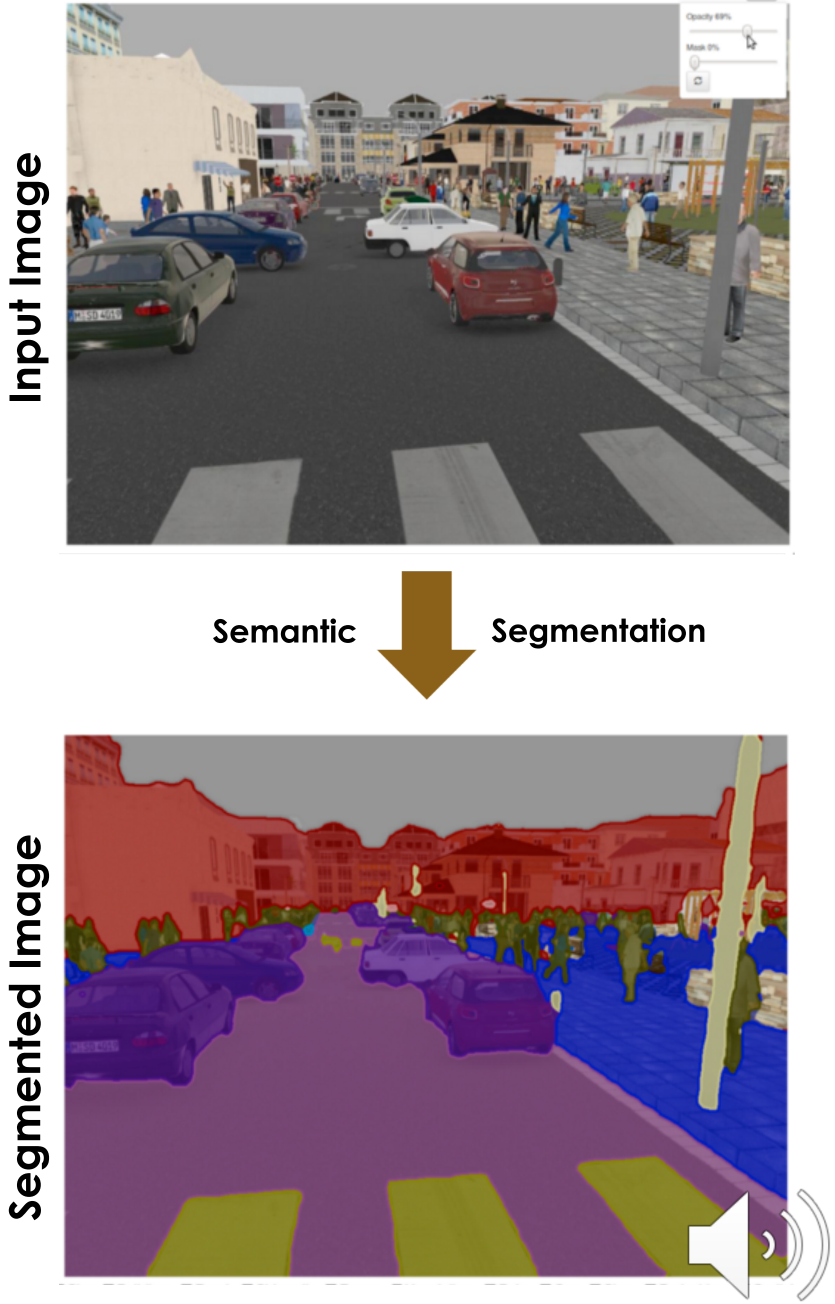
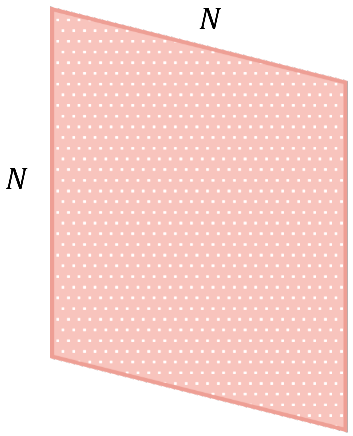


Image credit: https://mc.ai/how-to-do-semantic-segmentation-using-deep-learning/

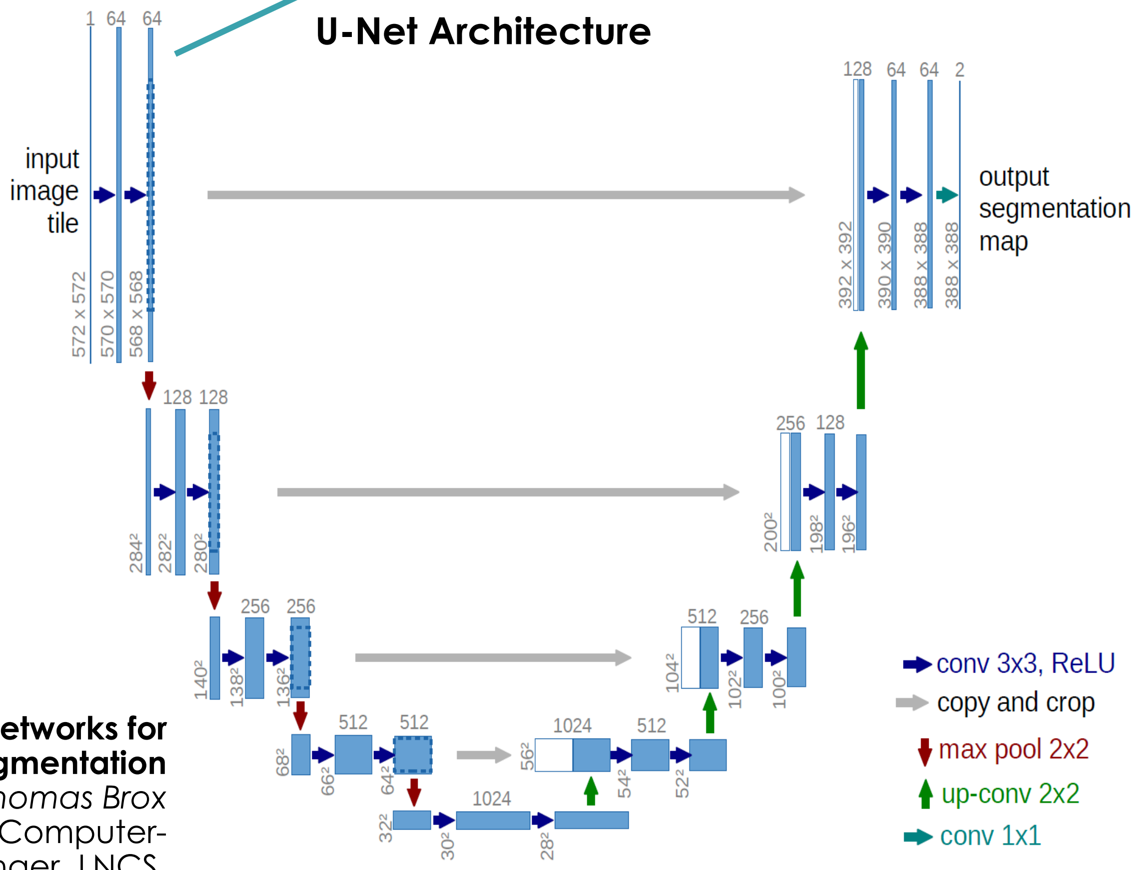
The U-Net Model

$$\epsilon = (3 \cdot 2^{L-2} - 1)dn_c$$

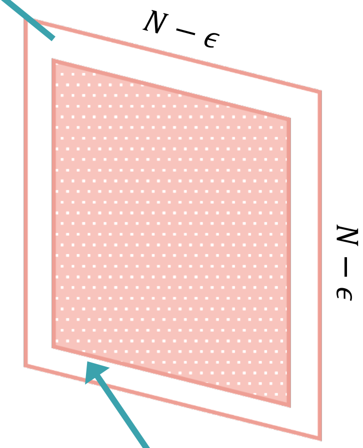
Input Image

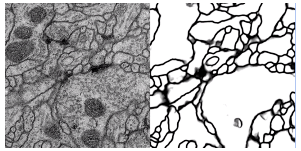


U-Net Architecture



Output Image



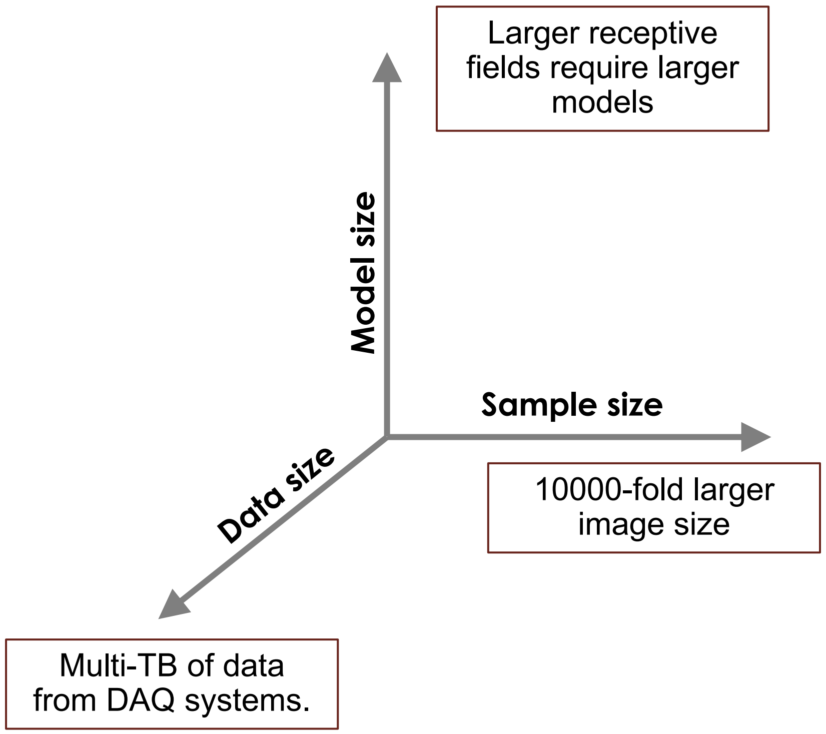


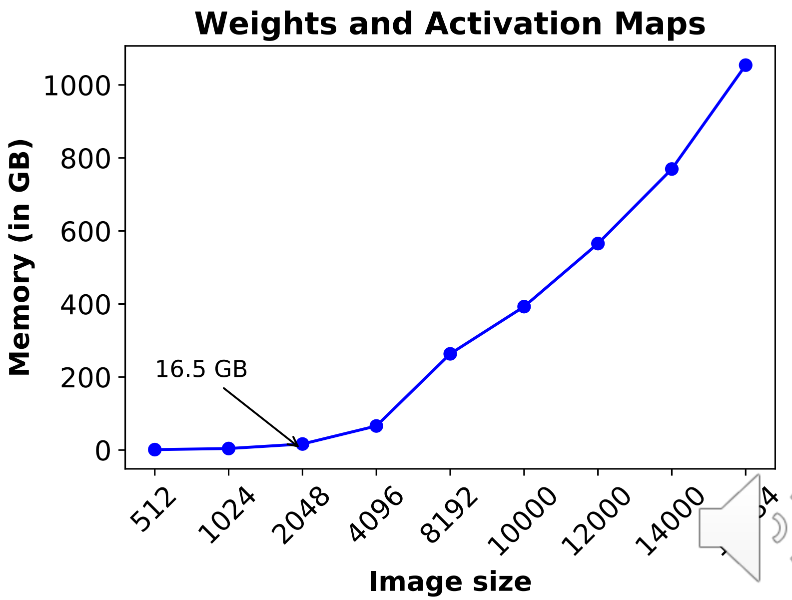
U-Net: Convolutional Networks for Biomedical Image Segmentation
Olaf Ronneberger, Philipp Fischer, Thomas Brox
Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015.

- Refer to this as the $\epsilon -$ region (halo).
- Halo width (ϵ) is a function of U-Net architecture (depth, channel width, filter sizes, etc.).
- Halo width (ϵ) determines the receptive field of the model.
- Larger the receptive field, wider the length-scales of identifiable objects.

Why Is It A Summit-scale Problem?

- Satellite images collected at high-resolutions (30-50 cm) yield very large 10,000 x 10,000 images.
- Most computer vision workloads deal with images of $O(10^2 \times 10^2)$ resolution (for example, ImageNet).
- This work targets ultra-wide extent images with $O(10^4 \times 10^4)$ resolution ⇒ 10,000-fold **larger data samples**!
- At present, requires many days to train a single model (even on special-purpose DL platforms like DGX boxes).
- Hyperparameter tuning of these models take much longer.
- Need accurate scalable high-speed training framework.
- **Large U-Net models** are needed to resolve multi-scale objects (buildings, solar panels, land cover details).
- Advanced DAQ systems generate vast amounts of high-resolution images ⇒ **large data volume**.



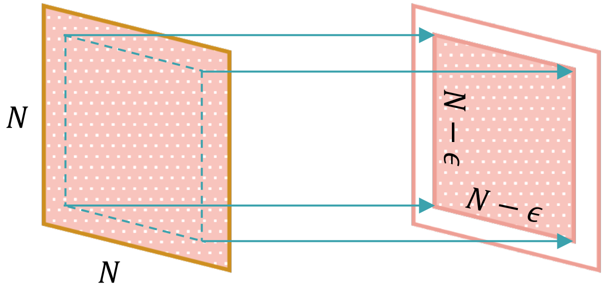


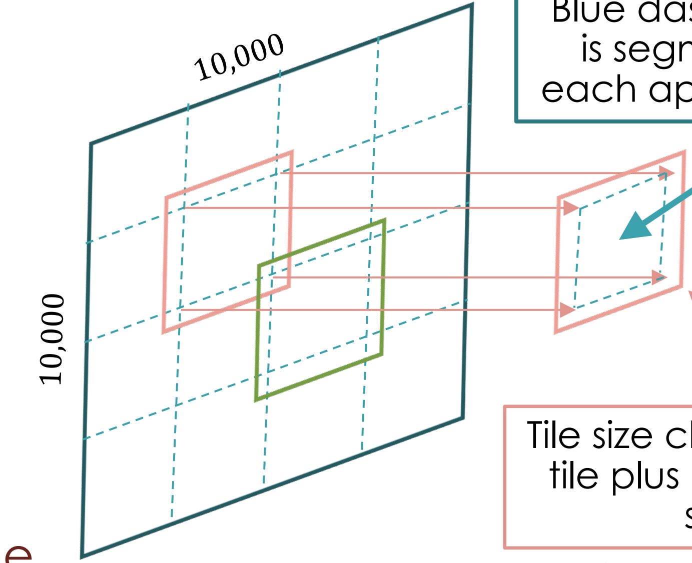
OAK RIDGE National Laboratory

Sample Parallelism - Taming Large Image Size

Leveraging Summit's Vast GPU Farm

- Given a $N\times N$ image, U-Net segments a $(N-\epsilon)\times(N-\epsilon)$ inset square.

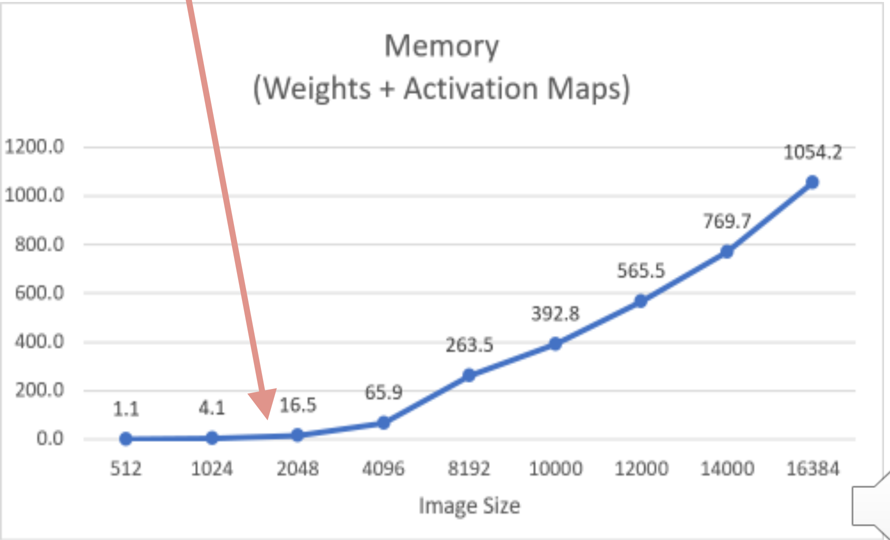




Blue dashed square is segmented for each appended tile.

Tile size chosen such that appended tile plus model parameters fit on a single Summit GPU.

- Partition each $N\times N = 10000\times10000$ image sample into non-overlapping tiles.
- Append an extra halo region of width ϵ along each side of each tile.
- Assign each appended tile to a Summit GPU. Use standard U-Net to segment appended tile.
- Each GPU segments an area equal to that of the original non-overlapping tile.



OAK RIDGE National Laboratory

Performance of Sample-Parallel U-Net Training

- Optimal tiling for each 10000×10000 sample image was found to be 8×8.

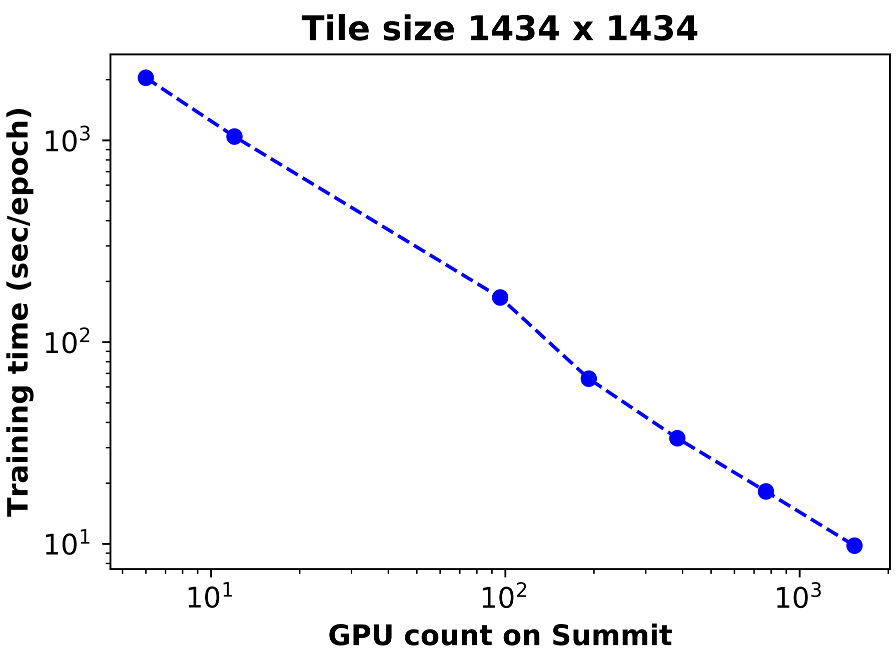
- Each 1250×1250 tile was appended with a halo of width $\epsilon = 92$ and assigned to a single Summit GPU.
  - 10 – 11 Summit nodes to train each 10000×10000 image sample.

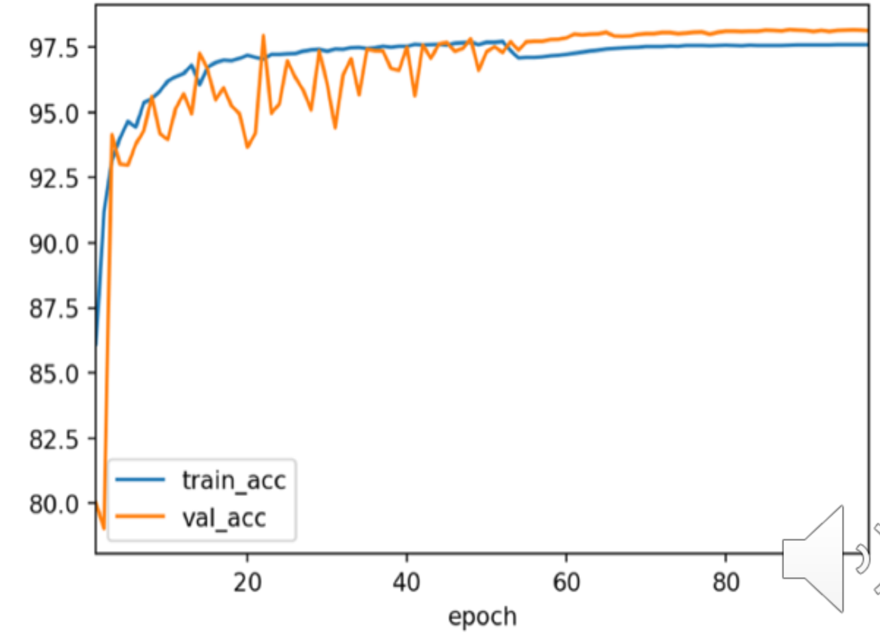
- A U-Net model was trained on a data set of 100 $10000 \times 10000 \times 4$ satellite images, collected at 30-50 cm resolution.

- The training time per epoch was shown to be ~**12 seconds** using **1200 Summit GPUs** compared to ~**1,740 seconds** on a **DGX-1**.

- Initial testing revealed no appreciable loss of training/validation accuracy using the new parallel framework.

+100X Faster U-Net Training



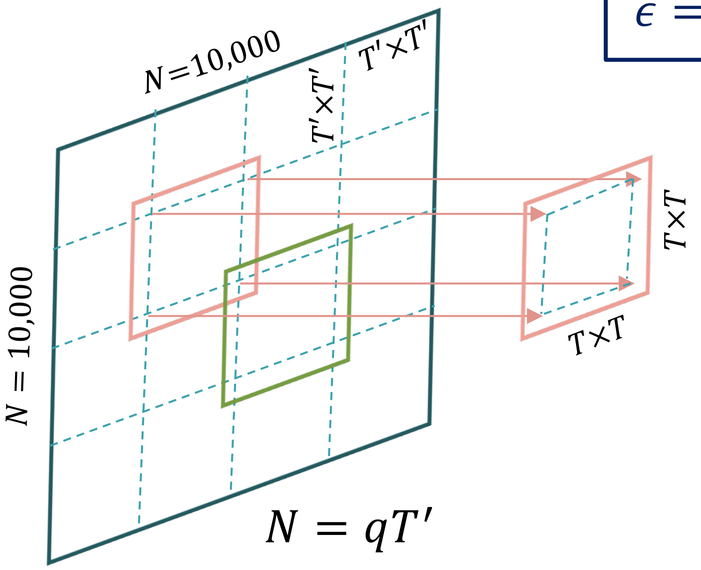


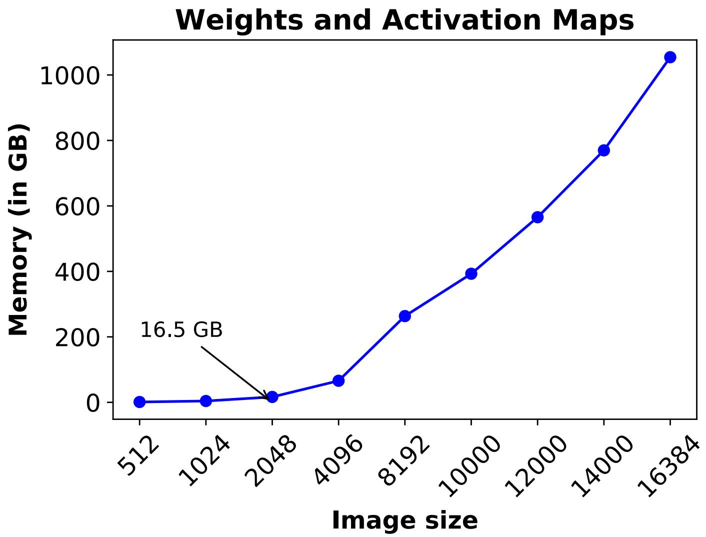
OAK RIDGE National Laboratory

Limitations of Sample Parallelism

- $K \rightarrow Filter\ size$
- $S \rightarrow Stride\ length$
- $P \rightarrow Padding\ lize$
- $n_c \rightarrow No.of\ convs\ per\ level$
- $L \rightarrow no.of\ UNet\ levels$
- $N{\times}N = q^2(T'{\times}T')$

$$\epsilon = (3 \cdot 2^{L-2} - 1)dn_c \qquad d = \frac{N(S-1)+K-2P}{S} - 1$$

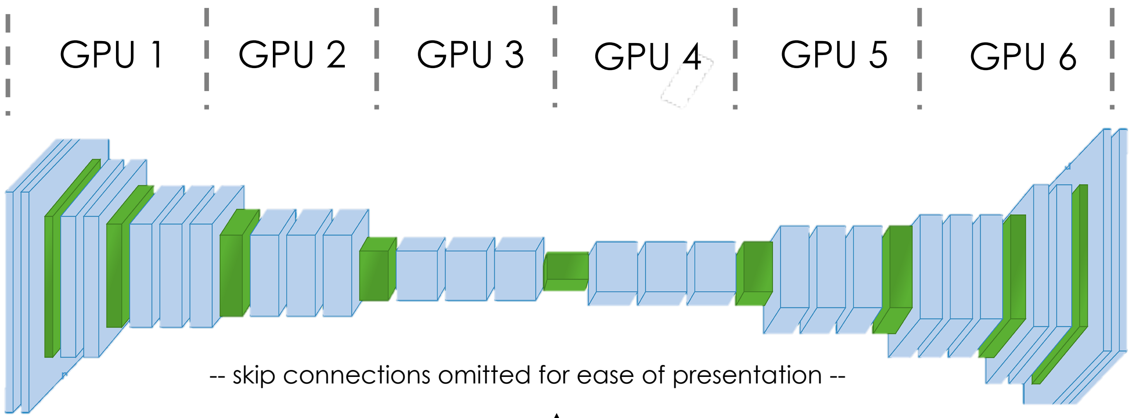


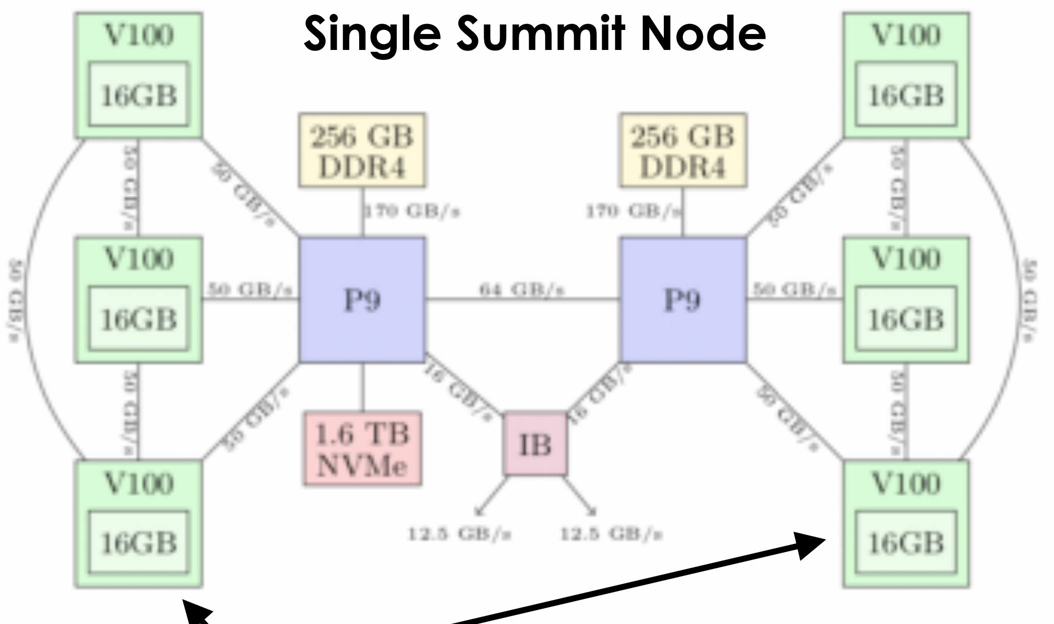


- An image of size $N\times N$ is partitioned into a $q\times q$ array of $T'\times T'$ tiles.
- $E \sim \dfrac{Total\ volume\ of\ computations\ per\ tile}{Total\ volume\ of\ useful\ computations\ per\ tile} = O\left(\dfrac{T^2}{T'^2}\right) \sim O\left(1 + q\dfrac{4\epsilon}{N}\right)$
- Ideally, $E = 1$.
- Decreasing q (increasing tile sizes) increases the memory requirement and quickly overtakes memory available per GPU.
- Decreasing ϵ decreases the receptive field of the model.
- On the other hand, the goal is to decrease q and increase ϵ.
- Decrease $q \Rightarrow$ increasing tile size T' and decreasing ϵ steers away from target receptive fields.
- **To satisfy both, larger U-Net models than can fit on a GPU needed.**
- **Need model-parallel execution.**

Model-Parallelism - Taming Large Model Size

Node-level Pipeline-Parallel Execution





- Number of consecutive layers mapped to a GPU is called **partition**.
- Number of layers in each partition is called **balance**.
- Subdivide each mini-batch of tiles into smaller **micro-batches** that are assigned to each partition.
- Micro-batches per partition ≡ $mbpp$

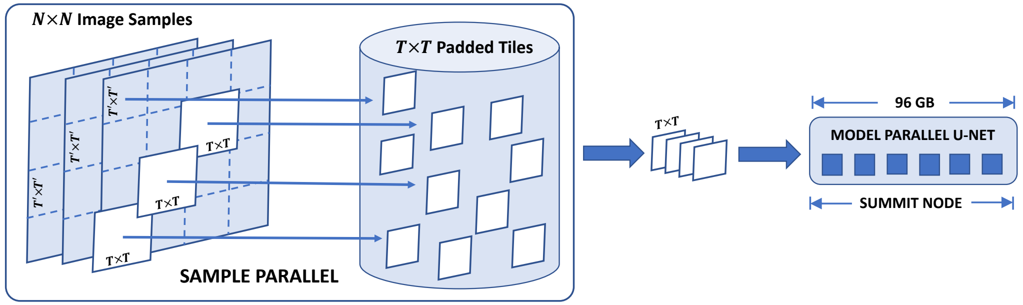
Memory needed/GPU = size(micro-batch) + size(partition)

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
D_3				F_{30}	F_{31}	F_{32}	F_{33}	B_{33}	B_{32}	B_{31}	B_{30}				Update	
D_2			F_{20}	F_{21}	F_{22}	F_{23}			B_{23}	B_{22}	B_{21}	B_{20}			Update	
D_1		F_{10}	F_{11}	F_{12}	F_{13}					B_{13}	B_{12}	B_{11}	B_{10}		Update	
D_0	F_{00}	F_{01}	F_{02}	F_{03}							B_{03}	B_{02}	B_{01}	B_{00}	Update	

TorchGPipe: PyTorch implementation of Gpipe* Framework

* Huang, Yanping, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le and Zhifeng Chen. "GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism." *NeurIPS* (2019).

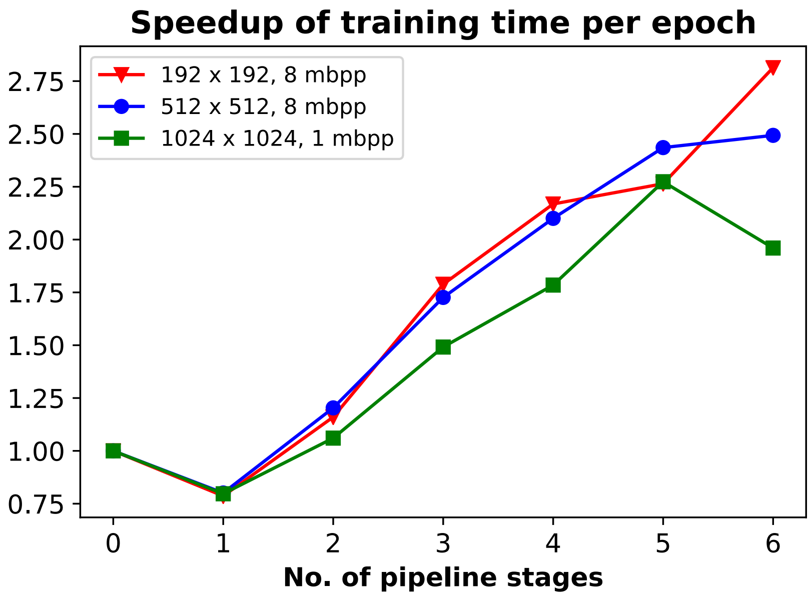
Model Parallel Experiments
Single Node Execution



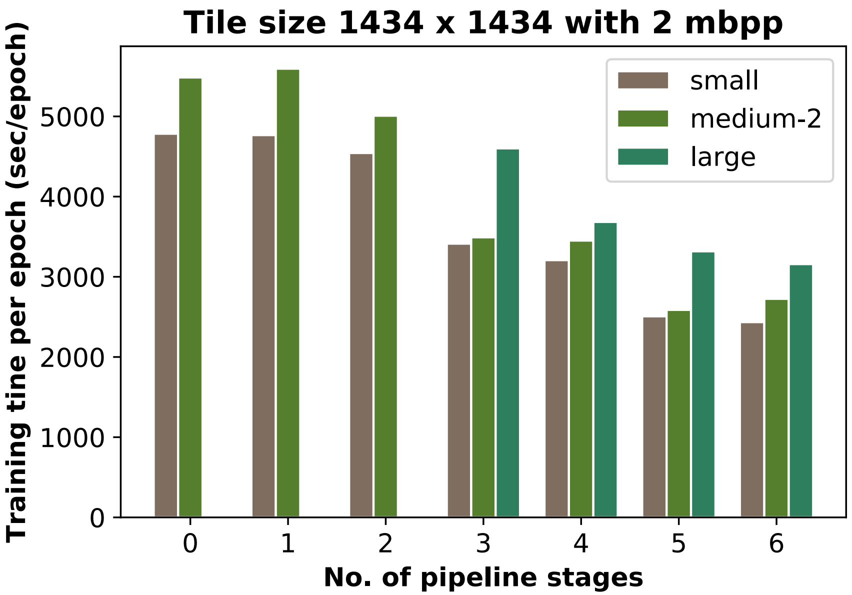
Benchmark U-Net Models

Model	No. of Levels	Conv. Layers Per Level	No. of Trainable Parameters	ϵ
Small (Standard)	5	2	72,301,856	92
Medium-1	5	5	232,687,904	230
Medium-2	6	2	289,357,088	188
Large	7	2	1,157,578,016	380

- $10\times$ larger number of trainable parameters.
- $4\times$ fold larger receptive field.



$2.8\times$ **(192)**, $2.5\times$ **(512)** and $2\times$ **(1024)** speedup using 6 pipeline stages.

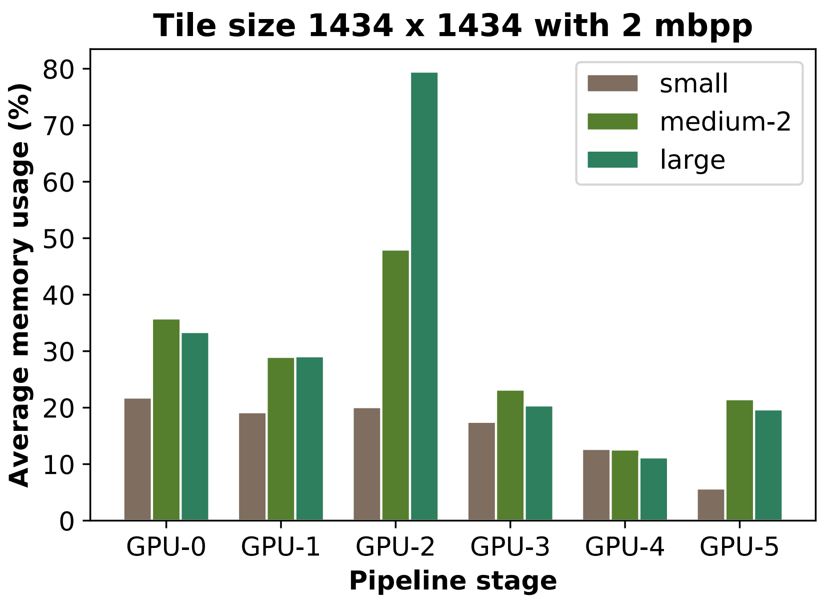


Speedup **doubles** (small: **1.97**; medium-2: **2.01**) as no. of pipeline stages increases from 1 to 6.

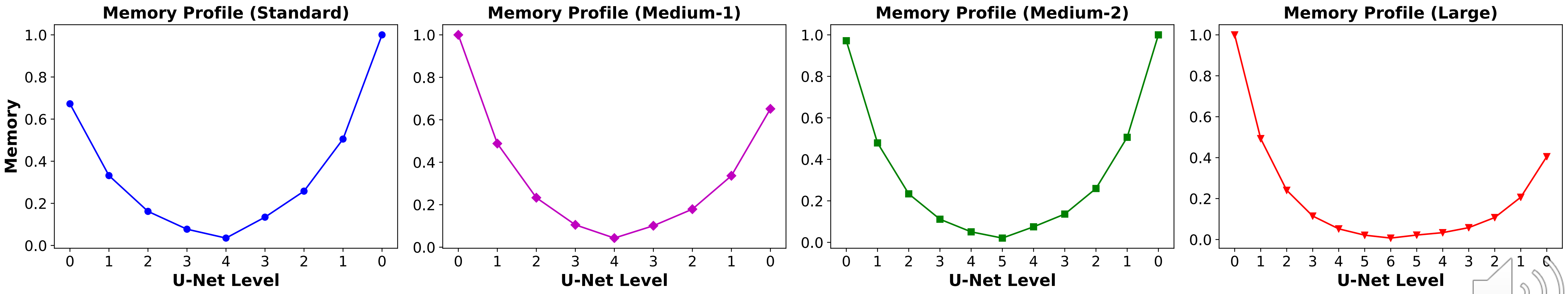
OAK RIDGE National Laboratory

Need for Performance Improvement

Single Node Execution

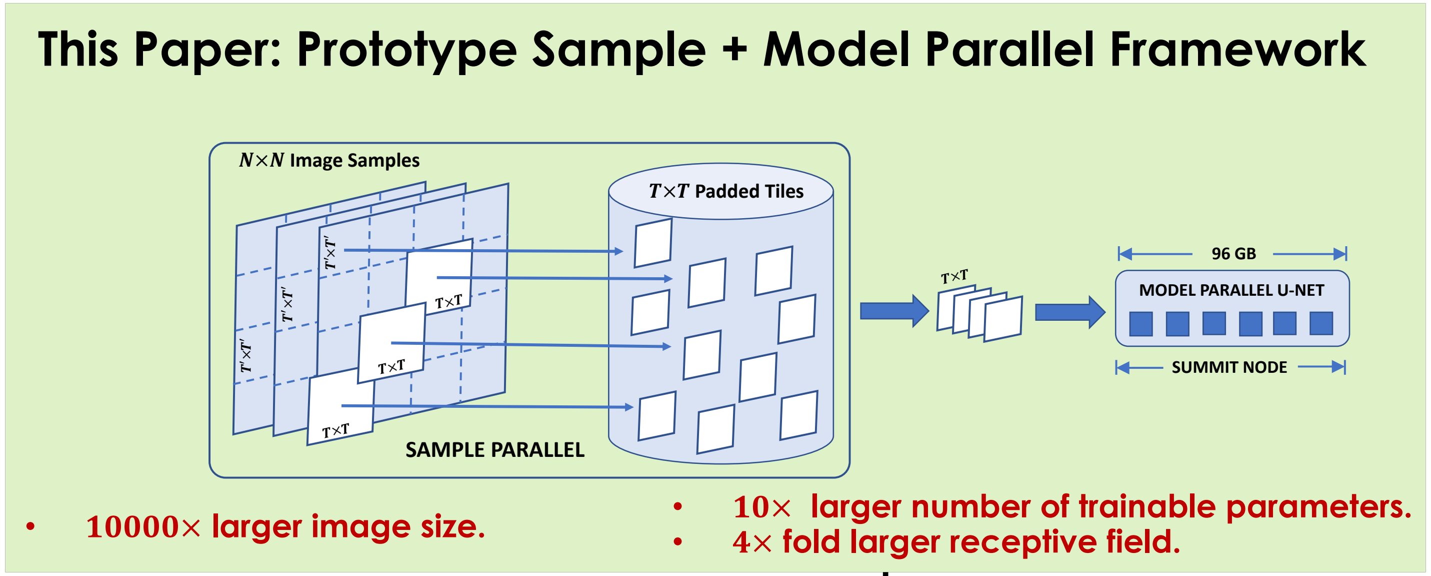


- Small, Medium-2 and Large Models:
  - **Layers:** 109, 129 and 149.
  - **Balances:** small {14, 24, 30, 22, 12, 7}; medium-2 {16, 26, 38, 26, 12, 11}; large {18, 30, 44, 30, 14, 13}.
- Need load balanced pipelined execution.
- Encoder memory: $E_\ell = O\left(I_\ell^2 + 2^\ell n_f \sum_{i=1}^{n_c} (I_\ell - i \cdot d)^2\right)$
- Decoder memory: $D_{\ell'} = O\left(2^{\ell'} n_f \left(2I_{\ell'}^2 + \sum_{i=1}^{n_c} (I_{\ell'} - i \cdot d)^2\right)\right)$
- Memory profile: $E_\ell + D_\ell$, vs. ℓ , $\ell' = L - \ell$

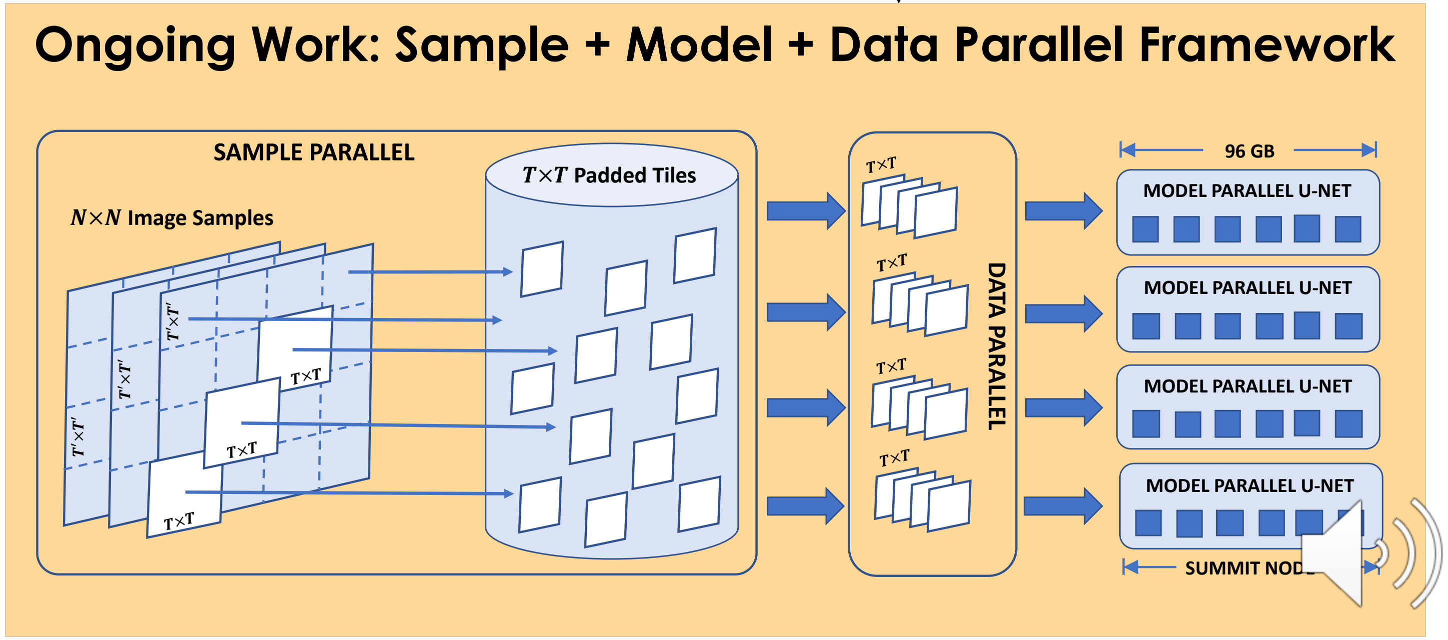


Wrapping Up

- Training image segmentation neural network models become extremely challenging when:
  - Image sizes are very large
  - Desired receptive fields are large
  - Volume of training data is large.
- Fast training/inference needed for geo-sensing applications –satellite imagery, disaster assessment, precision agriculture, etc.
- This work is a first step – can train 10× larger U-Net models with 4× larger receptive field on 10000× larger images.
- Ongoing efforts are underway to integrate load balancing heuristics and data-parallel execution to handle large volumes of training data efficiently.



Load Balance Heuristics → ← Data Parallelism



THANK YOU