No Compromise – Using Unified Memory for High Resolution Medical Image AI

Joe Yeh, M.D., CEO

Outline

- Dimension problem with medical image AI
- Ways to overcome dimension problems
- Using unified memory for CNN training
  - Challenges
  - Improved methods
- Results of medical image AI using high resolution images

aetherAI

Dimension problem with Medical Image AI

- How much can a Tesla V100 (32Gb) take in ?
- For ResNet-101, batch size=32, it can take in images of 512*512*3
- For ResNet-101, batch size=1, it can take in image of 3880*3880*3
- For 3D ResNet-101, batch size=32, it can take in images of 92*92*42*1
- For 3D ResNet-101, batch size=1, it can take in image of 577*577*42*1

aetherAI

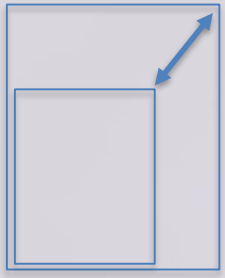
Typical Resolution of Medical Image

- Chest radiograph : 4000*5000 uint16
- Computed tomography : 512*512*50 uint16
- Low-dose lung CT: 512*512*500 uint16
- Digital Whole Slide Image : 100,000*50,000*3 uint8

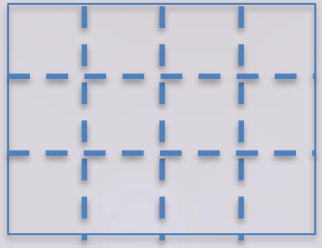
aetherAI

Current approaches to deal with size problems with medical image AI

- Resizing



- Patch-based methods

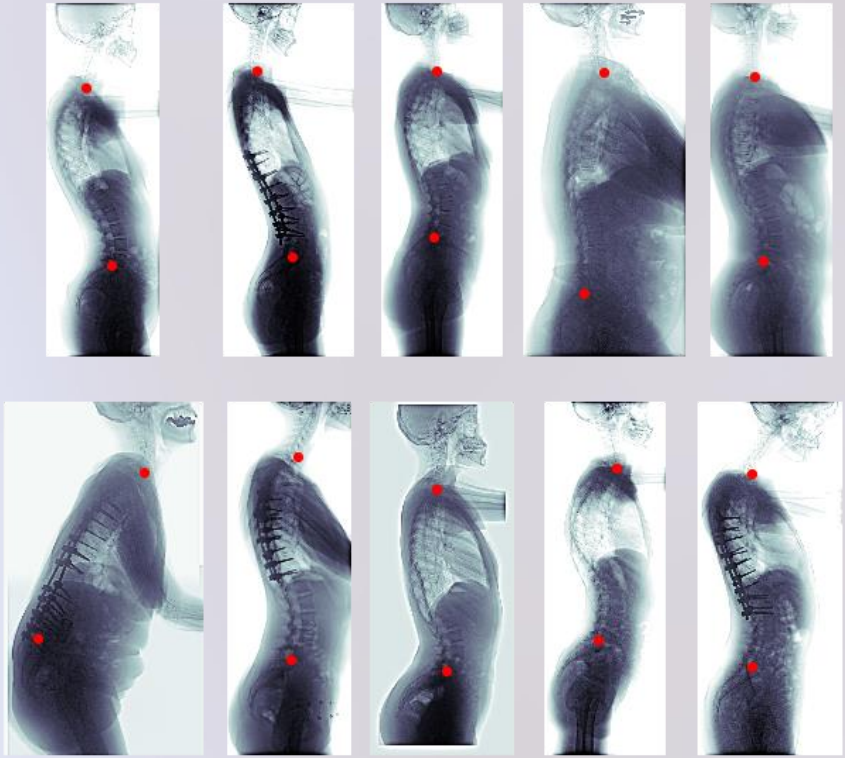


aetherAI

Does input size really matter ?

aetherAI

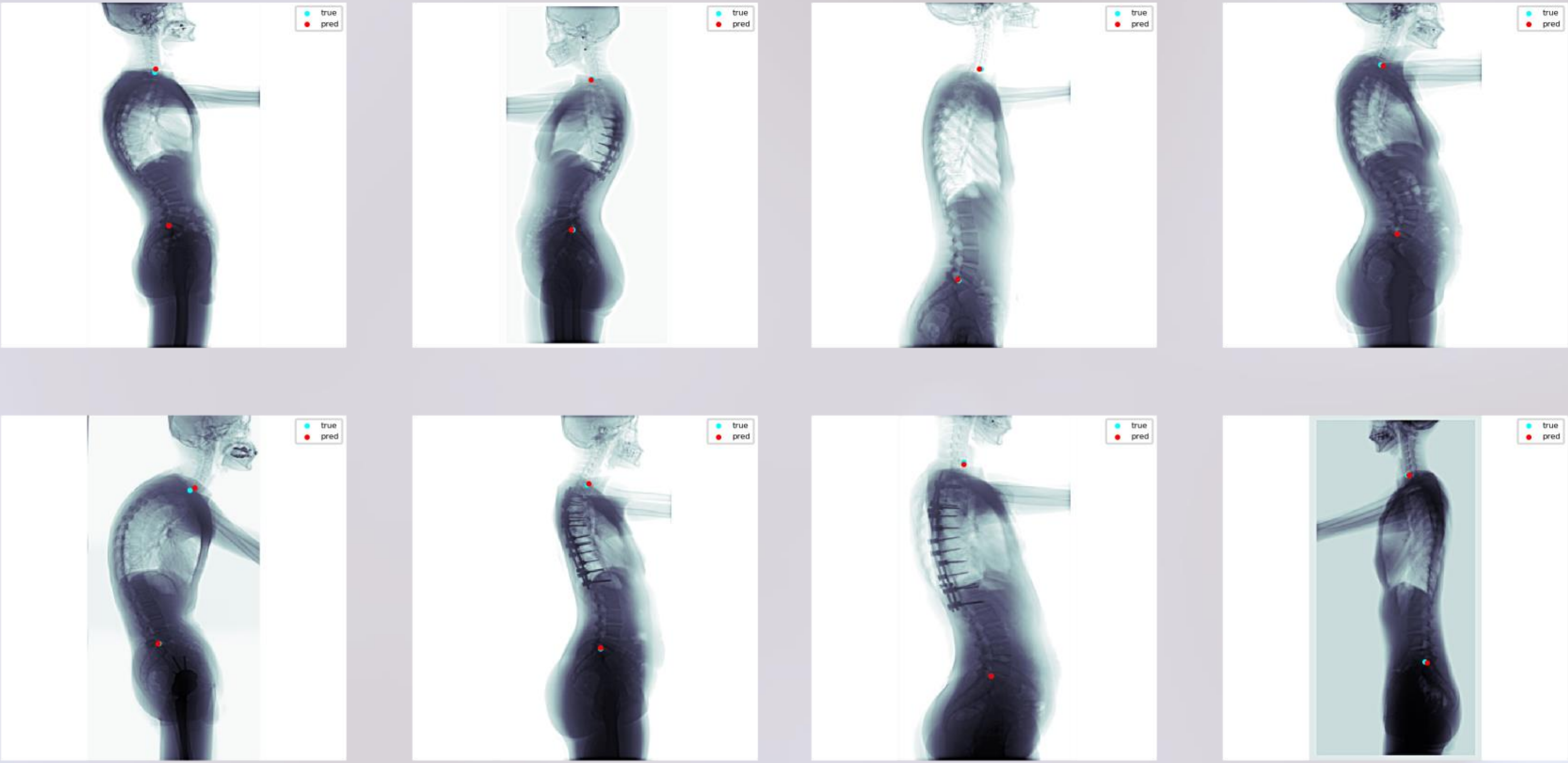
Automatic Analysis of Standing Lateral Radiograph



- Goal : To teach neural network to recognize the center of C7 spine and superior posterior corner of the Sacrum (for calculating SVA)
- Dataset : ~1500 annotated radiographs
- 80% data for training, 10% for validation, 10% for testing

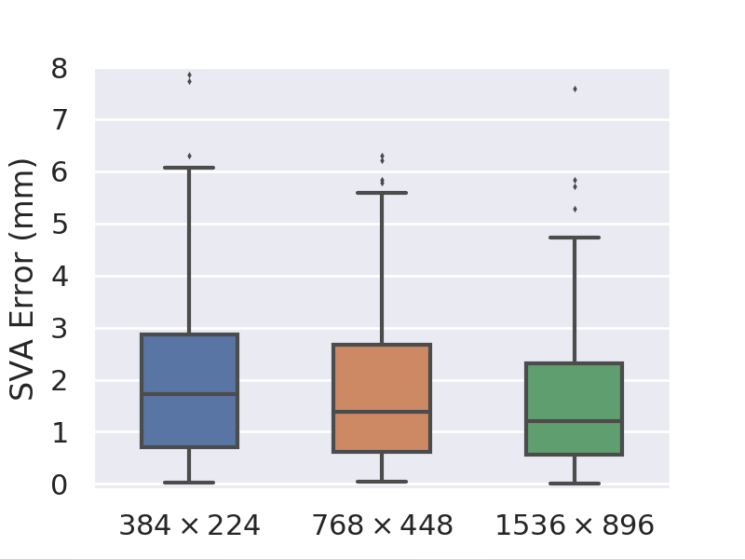
aetherAI

Prediction on Test Images



Results of Using Different Image Resolution

- Model: ResUNet35
- Performance metric : mean absolute error (in mm)
- Training batch size : 8 (2 per GPU, 4 GPUs total)



	384 × 224	768 × 448	1536 × 896
Memory consumption	8Gb	14Gb	>32 Gb

aetherAI

Ways to increase maximum input size

- Explicit device placement
- vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design
- TFLMS: Large Model Support in TensorFlow by Graph Rewriting
- CUDA Unified Memory

aetherAI

Explicit Device Placement

- How : Manual allocation of memory and compute
- Pros : Easy to implement in codes
- Cons : Data placed on system memory can only be processed by CPU
- To maximize performance, a rule of thumb is to place most frequently-used allocations on GPU memory to leverage data reuse.
- However, in DNN training, almost all allocations are accessed equally twice (forward and backward passes) in a batch.

Dynamic Swapping

- How : Dynamically swapping data between system and GPU memory in runtime.
- To maximize the performance, data should be swapped to GPU memory on every compute.
- Swapping mechanism is suitable for DNN training.
  - Access pattern is predetermined. Easy to schedule swapping.
- Implementations:
  - vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design [MICRO'16]
  - TFLMS: Large Model Support in TensorFlow by Graph Rewriting

vDNN

- Proposed swapping strategies for DNN to reduce memory requirement.
- Swapping the entire layer as its basic unit.
- The implementation is not released.

IBM Large Model Support in Tensorflow (LMS)

- How : Analysis and rewriting of computation graph.
- More general than vDNN since the network is no longer composed of layers
- The implementation is provided in IBM PowerAI package.
- Since GPU cores cannot directly access system memory, all data required by an operation should be in GPU memory. Once its size is too large to fit in, out-of-memory error occurs.

CUDA Unified Memory

- Unified Memory (UM) makes system memory accessible for GPU.
- Out-of-memory error due to limited GPU memory is eliminated since data can be placed anywhere.
- Because of low bandwidth of system memory access, data should better be placed in GPU memory.
- CUDA UM provides driver-defined swapping strategy like LRU, and APIs to hint data prefetch and placement.
    - In our experiments, training DNN on unified memory is slow. Default swapping mechanism may not be optimal.

aetherAI

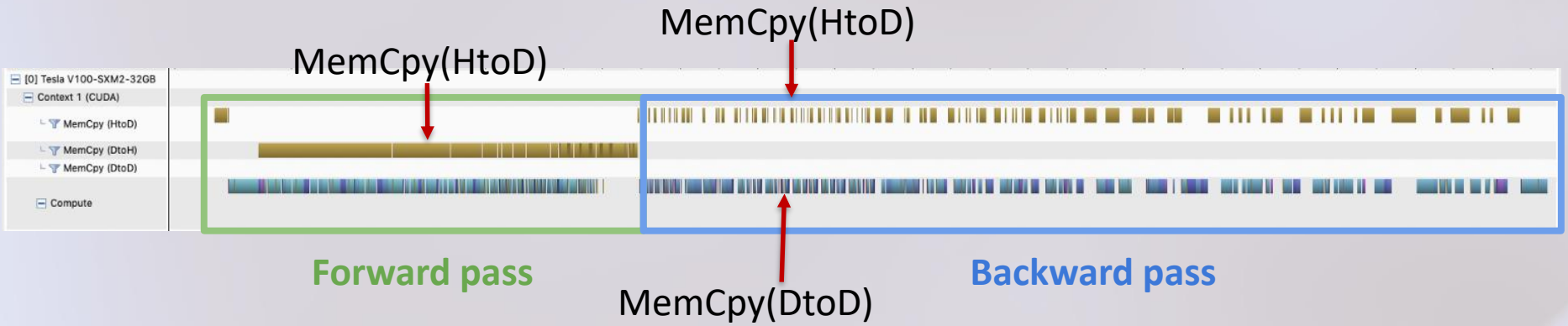
Comparisons

	Explicit Device Placement	Large Model Support	Unified Memory
Maximal model size	Limited by system memory	Limited by GPU memory	Limited by system memory
Performance	Extremely slow when CPU processes most ops	Great	Slow, Needs tuning
Programmability	Needs efforts	Great	Great

aetherAI

Observing the swapping strategies (LMS)

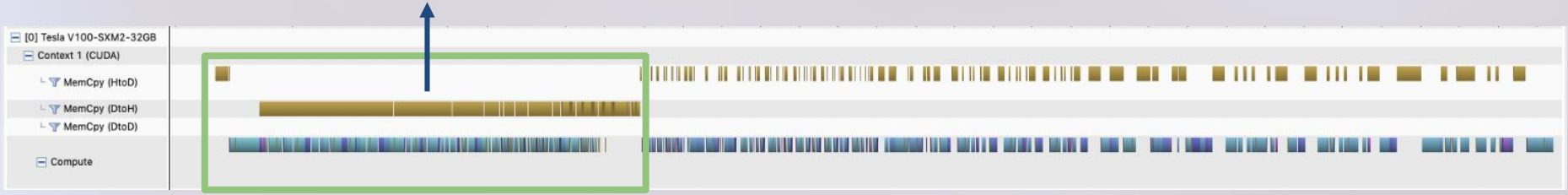
- Resnet-50 v1, batch size: 1, image size: 6000*6000(RGB)
- Visualized by NVIDIA Visual Profiler



aetherAI

Observing the swapping strategies (LMS)

In forward pass, layer outputs should be kept for back propagation but not immediately used. LMS swaps these data to system memory to spare more space.

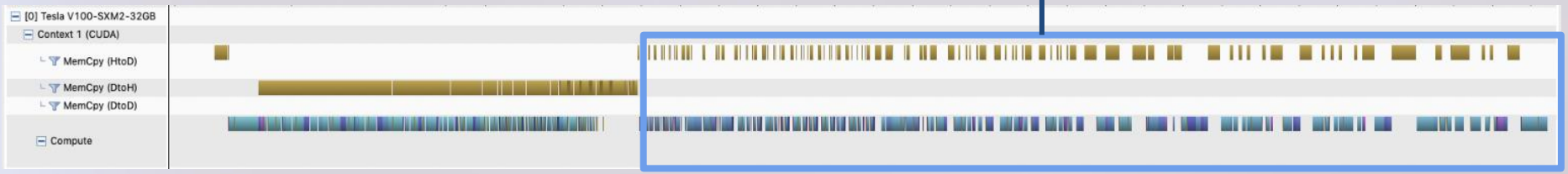


Forward pass

aetherAI

Observing the swapping strategies (LMS)

In backward pass, layer outputs in system memory are swapped in to GPU memory for computation.

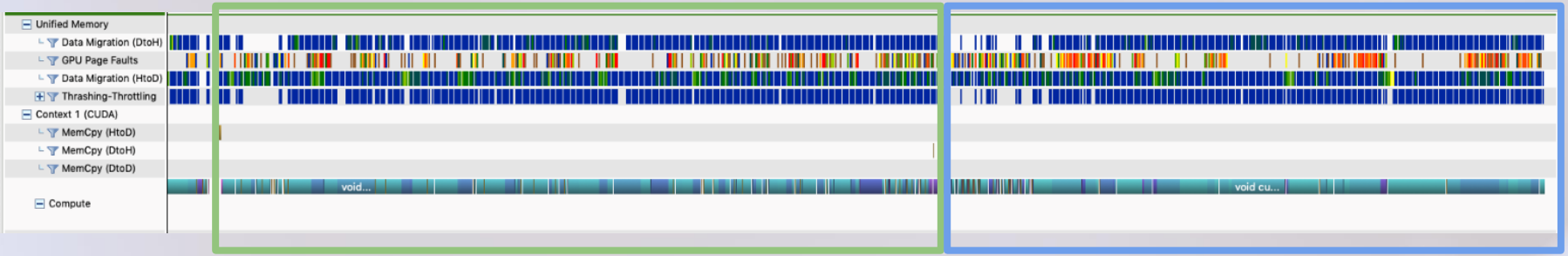


Backward pass

aetherAI

Observing the swapping strategies (Unified Memory)

- Swapping in and out everywhere during training.
- Data recently accessed are moved to GPU memory, and in the meanwhile other least-recently-used pages are kicked out to free space.



Forward pass

Backward pass

aetherAI

Way to improve throughput of Unified Memory

- Group execution
- Eager outward (device to host) swapping
- Prefetch

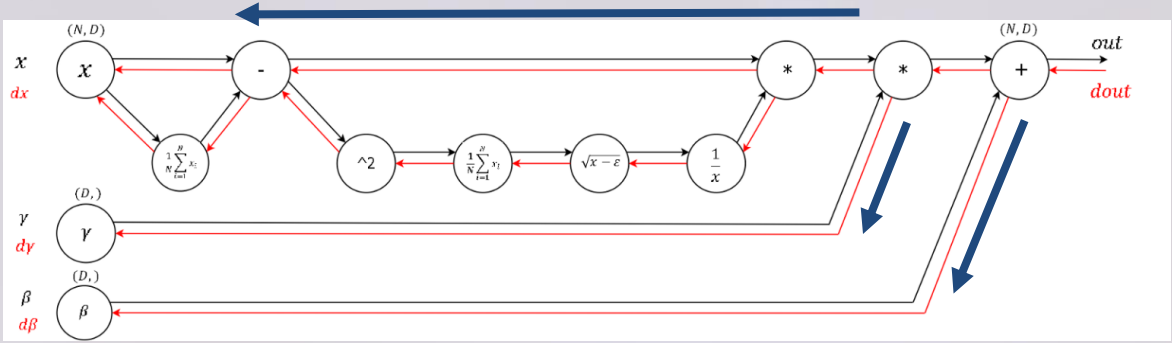
aetherAI

Group Execution on Backprop

- Motivation:

  Typical backpropagation processes the network in parallel. Although the mechanism increases throughput ordinarily, it requires more memory space (working set). The large working set aggravates thrashing when there is insufficient GPU memory.

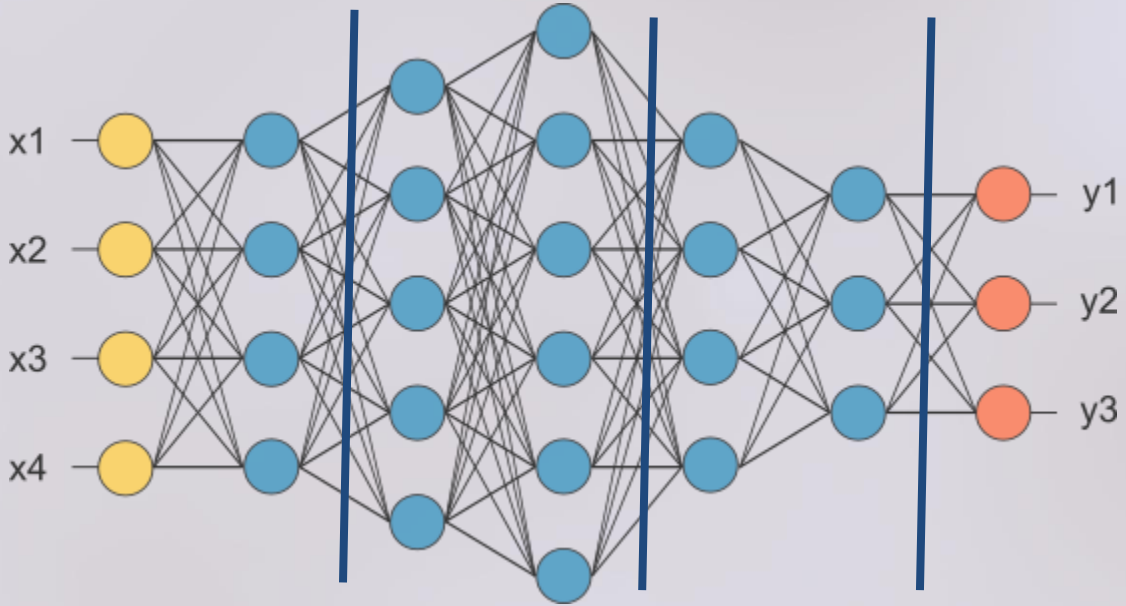
- Design Philosophy: Reduce parallelism



aetherAI

Layer Grouping

- Perform backward pass group by group to reduce parallelism.



aetherAI

Auto Layer Grouping

- Group granularity needs tuning to balance parallelism and working set size.
- Auto layer grouping algorithm:
  1. Working set size of each layer is derived by examining the tensor graph.
  2. Set a maximal working set size per group, say 8GB.
  3. Union several layers into a group if working set size not exceeds.

Results of Group Execution on Backprop

	LMS	Vanilla UM	Grouping(B)	Grouping(E)
256	161 ± 7	243 ± 1	215 ± 2	214 ± 2 .
512	46.0 ± 1.1	65.6 ± 0.2	64.2 ± 0.2	63.1 ± 0.4
768	21.1 ± 0.4	14.2 ± 6.9	15.3 ± 4.3	16.7 ± 5.1
1024	about 8	2.01 ± .28	2.02 ± .09	2.39 ± .12

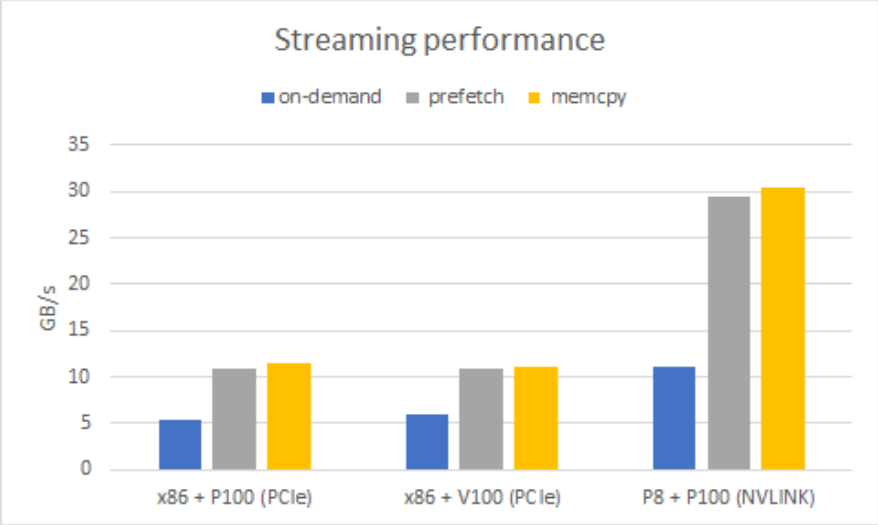
Grouping(B): Slicing groups by blocks.

Grouping(E): Slicing groups by equalizing working set to 2048 MB.

aetherAI

Why Data Prefetch?

- On-demand data migration caused by page fault is not as efficient as explicit memory copy and prefetch.

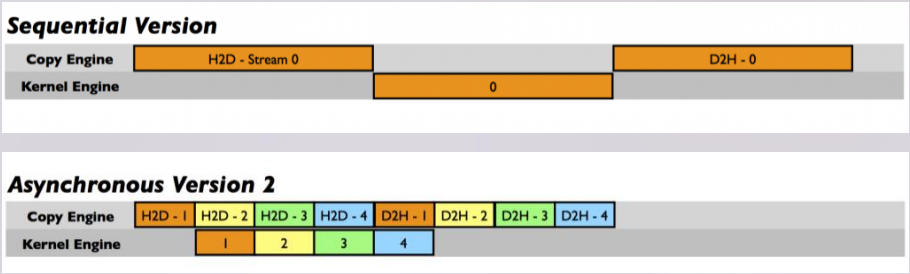


aetherAI

Source : https://devblogs.nvidia.com/maximizing-unified-memory-performance-cuda/

Why Data Prefetch? (cont.)

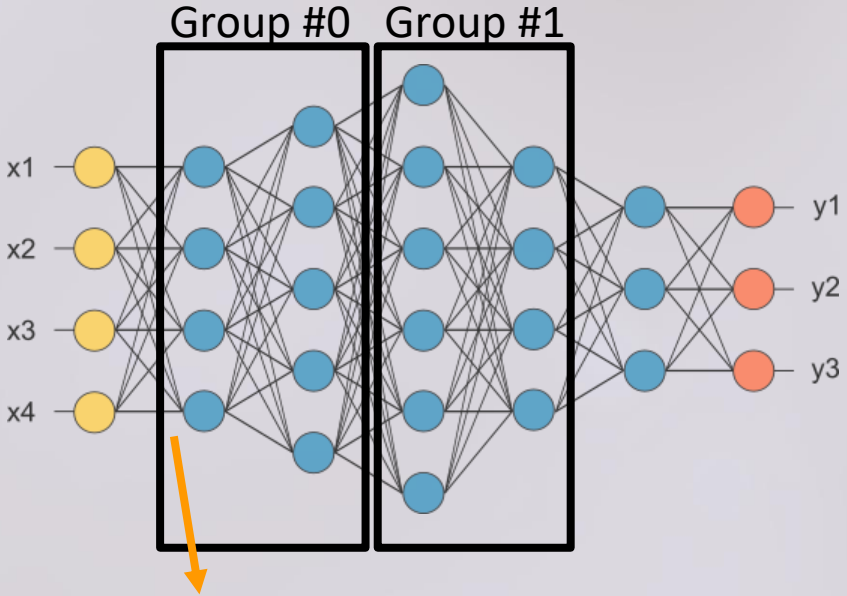
- Prefetch leverages data transfer overlap.



Source : https://devblogs.nvidia.com/how-overlap-data-transfers-cuda-cc/

Data Prefetch

- Use cuMemPrefetchAsync API.

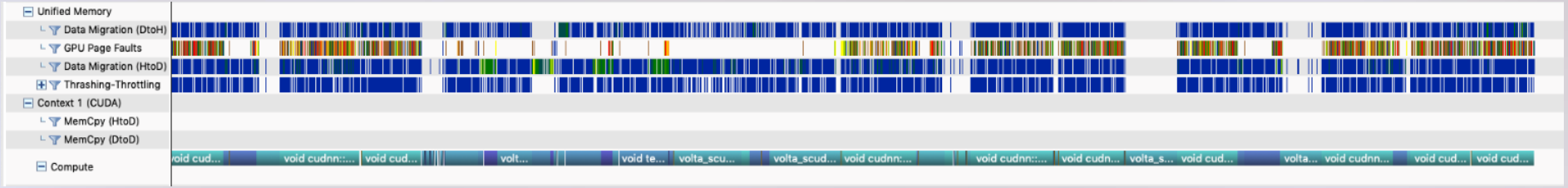


Start prefetching data requiared by Group #1.

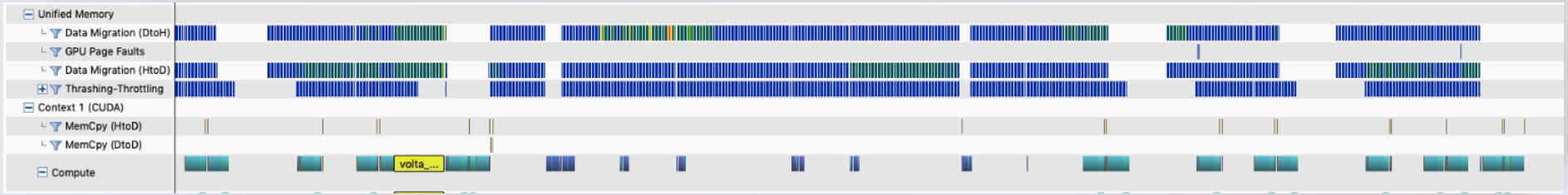
aetherAI

Visualization

Before:



After:

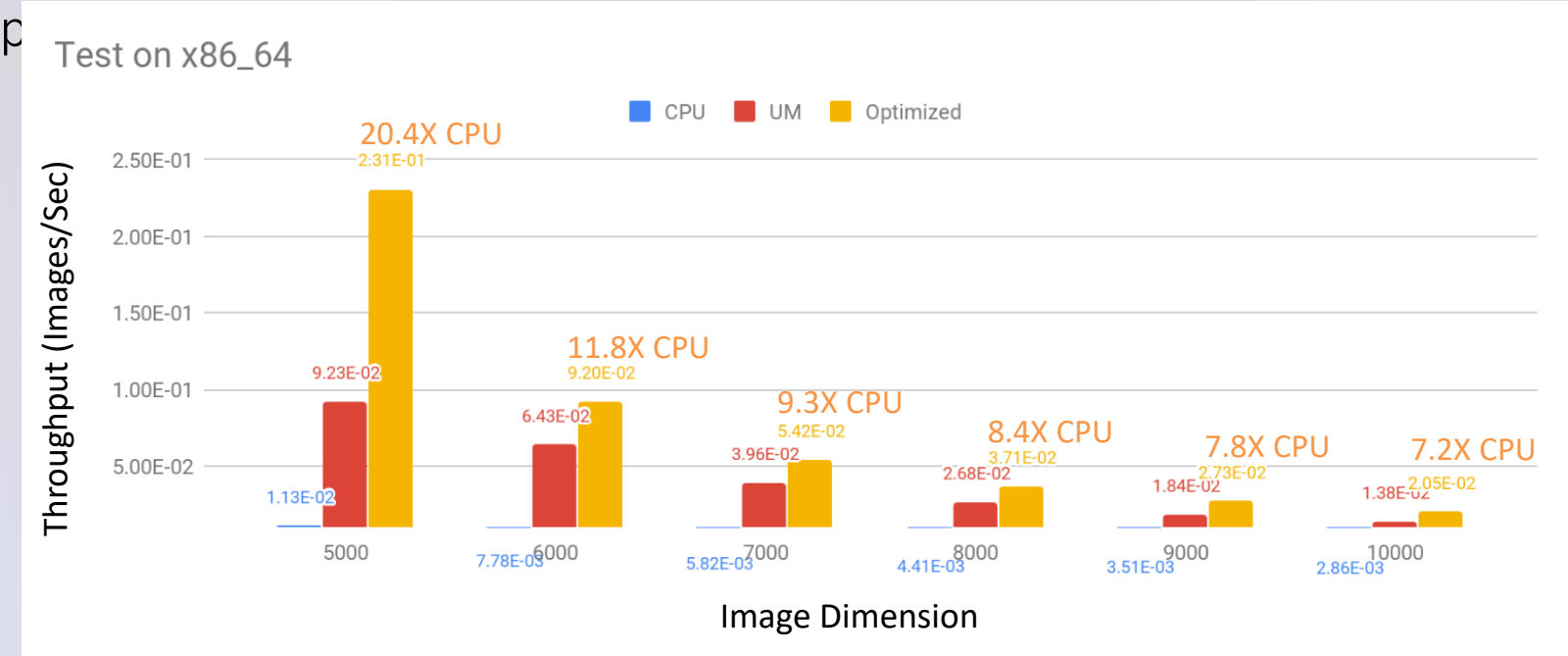


Almost all page faults are eliminated!

aetherAI

Results on TAIWANIA 2

Resnet-50 v1 with batch size 1. Our method achieves 1.4~2.5x sp



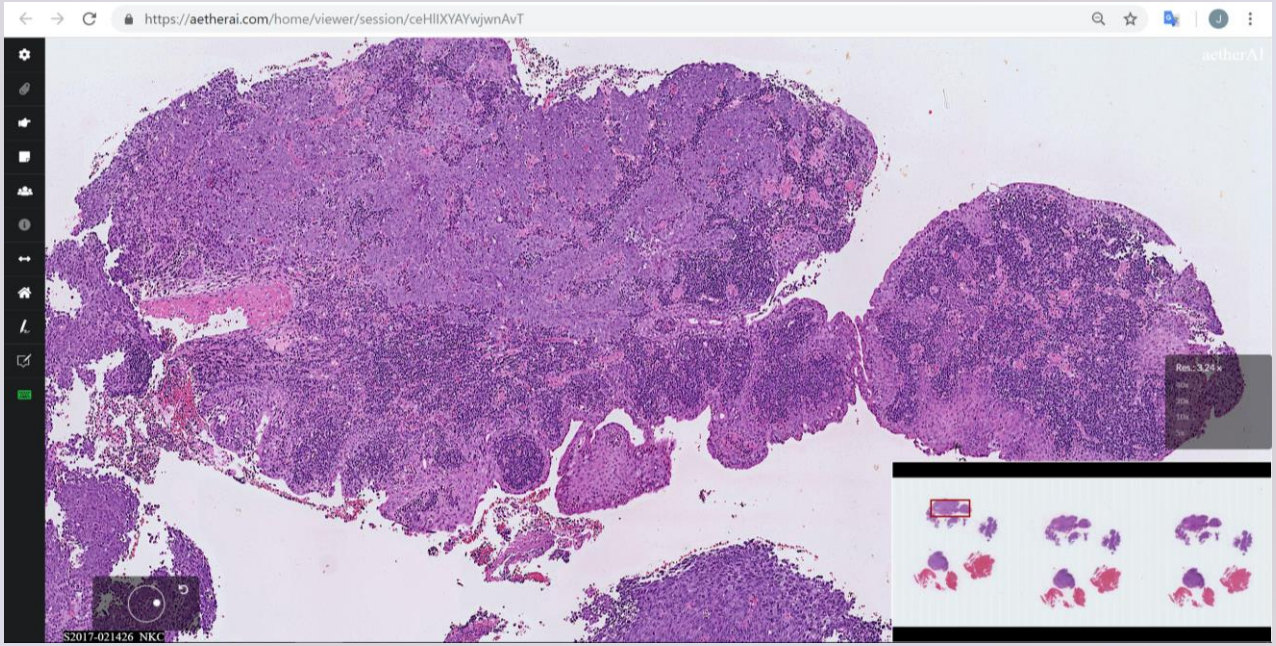
Results of Using Unified Memory for High-Res Medical Image AI

- Digital pathology
  - Cancer screening model
- Radiology
  - Bone radiograph keypoint detection

aetherAI

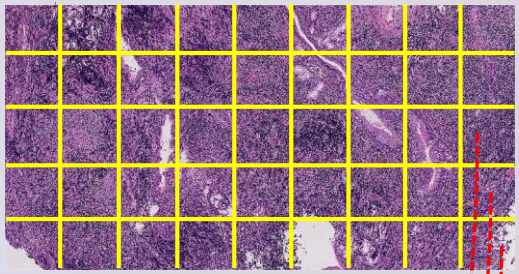
Digital Whole Slide Image (WSI)

- Generated by slide scanner
- Resolution can be up to 200,000 * 100,000 pixels (20 Billion)

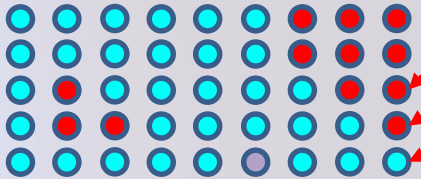


aetherAI

Two-Level AI Model for Cancer Detection on Whole Slide Image



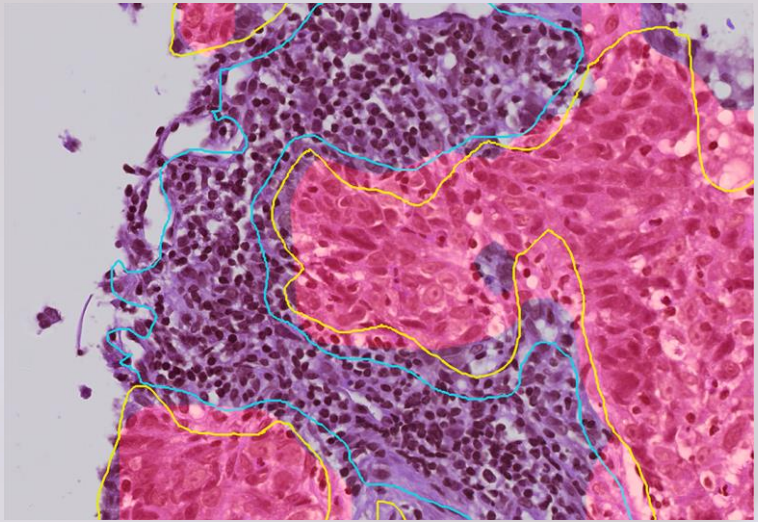
Divide WSI into patches



Patch-level model (>10M Patches)
Background, Benign, Cancer
Classification accuracy : 98%

Slide-level model
260 Training, 100 Testing
Classification Accuracy : 97%

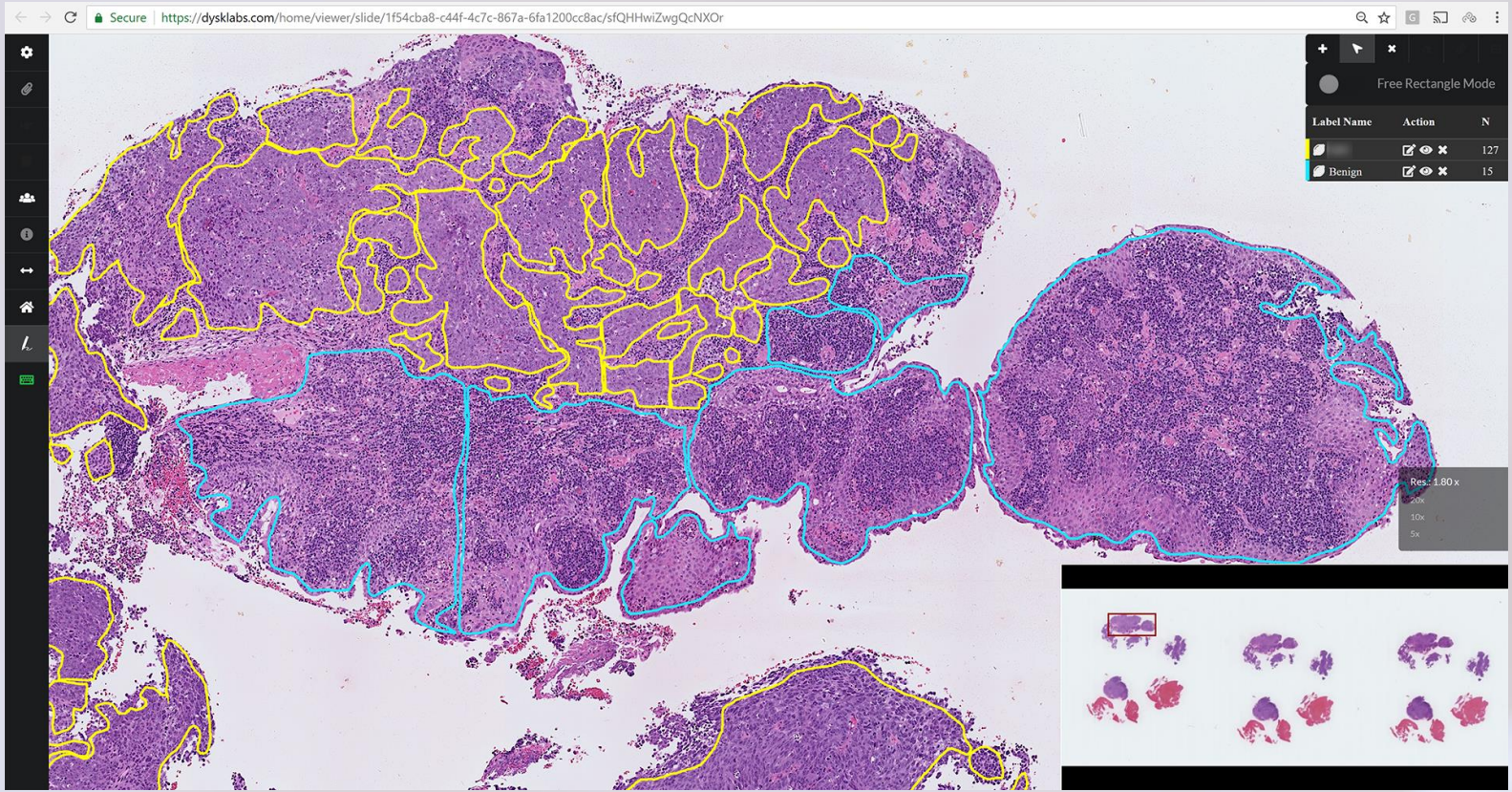
Benign or NPC ?



Ground Truth : Cancer, Normal Tissue
Shadowed area : Cancer predicted by AI

aetherAI

Annotation for Digital Pathology AI



Using images of entire specimen to train CNN a.k.a. the no-fuss approach

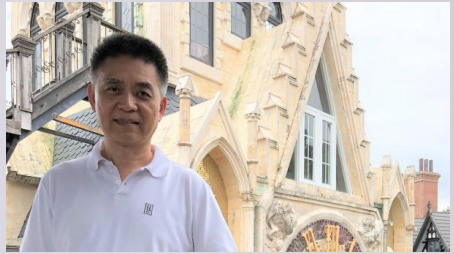
- Input size: 10000 x 10000 x 3 (RGB)
- Model : ResNet-50
- Training set : 780 images (357 NPC, 423 Benign)
- Validation set size: 68 images (32 NPC, 36 Benign)
- Hardware : HGX-1 nodes on Taiwania 2 Supercomputer, 8 Tesla V100(32gb) and 768 Gb system memory per node
- With batch size = 1, 360 Gb system memory is used for training through Unified Memory
- Each update takes **2.5 minutes**.

aetherAI

National Center for High-Performance Computing（NCHC）Taiwan

Director General
Shepherd Shi

Deputy Director General
His Ching Lin



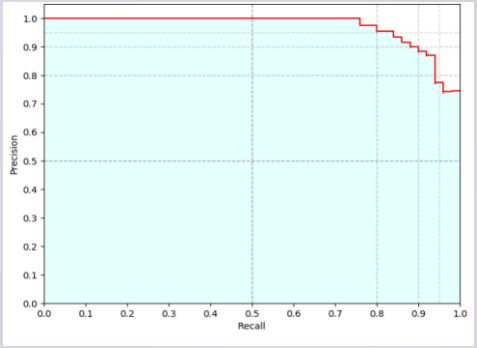
Deputy Director General
Sam Chu

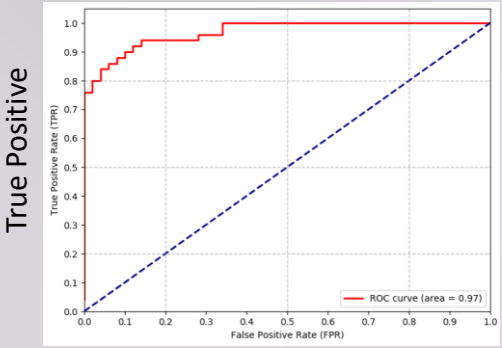
Slide-Level Prediction Testset Performance

Precision-Recall

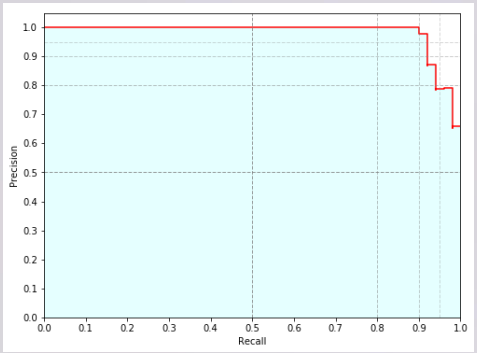
True vs False Positive

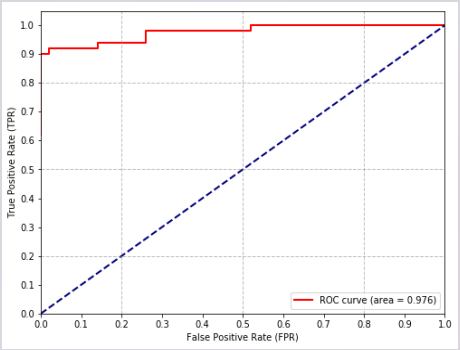
No-fuss model





Two-stage model





Precision

True Positive

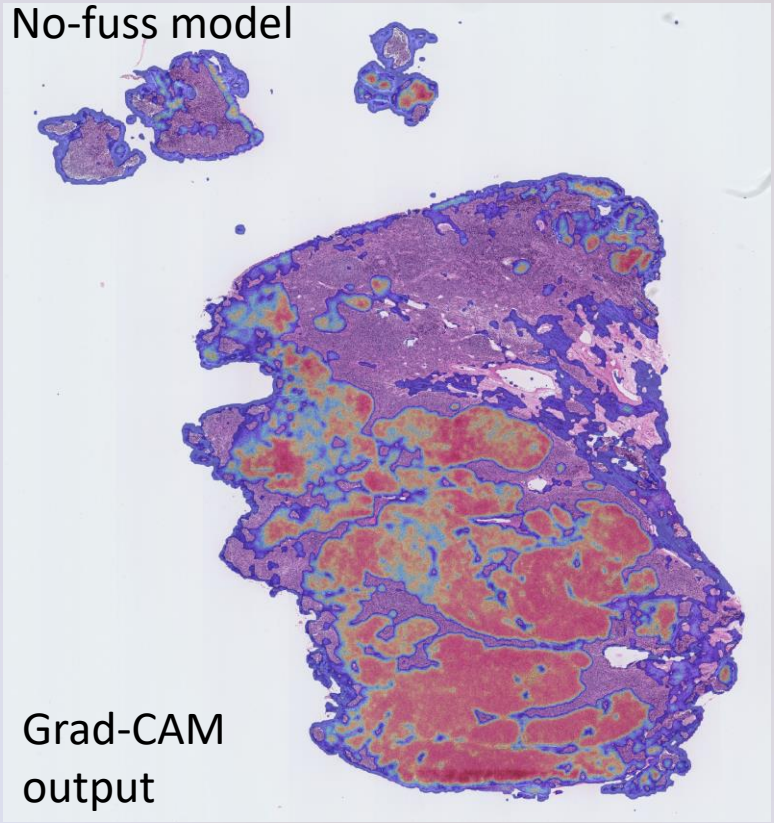
Recall

False Positive

aetherAI

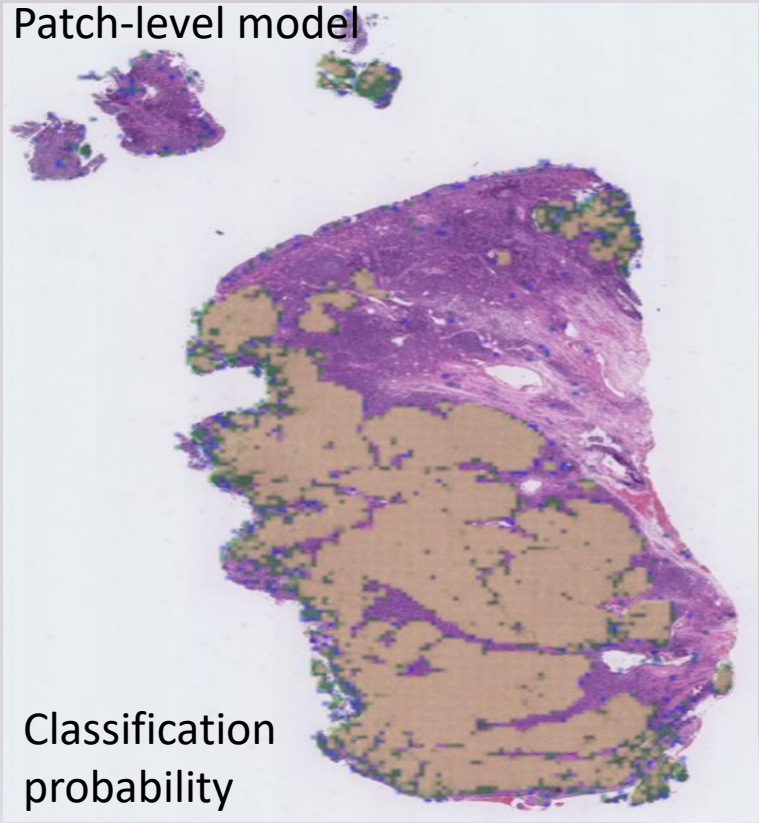
Comparison of the two approaches

No-fuss model



Grad-CAM output

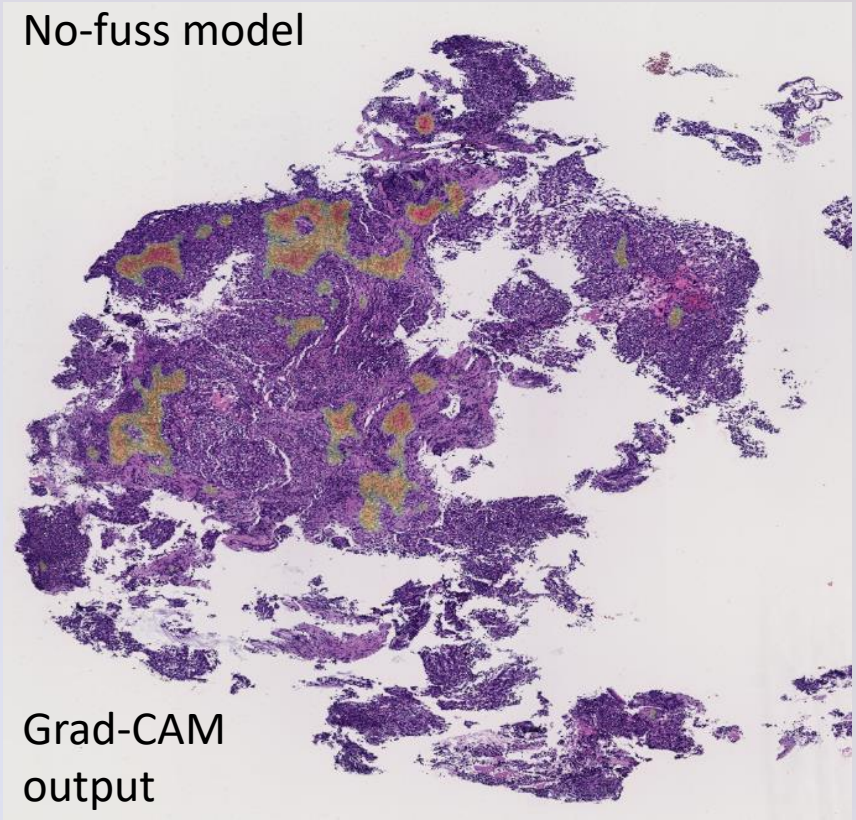
Patch-level model



Classification probability

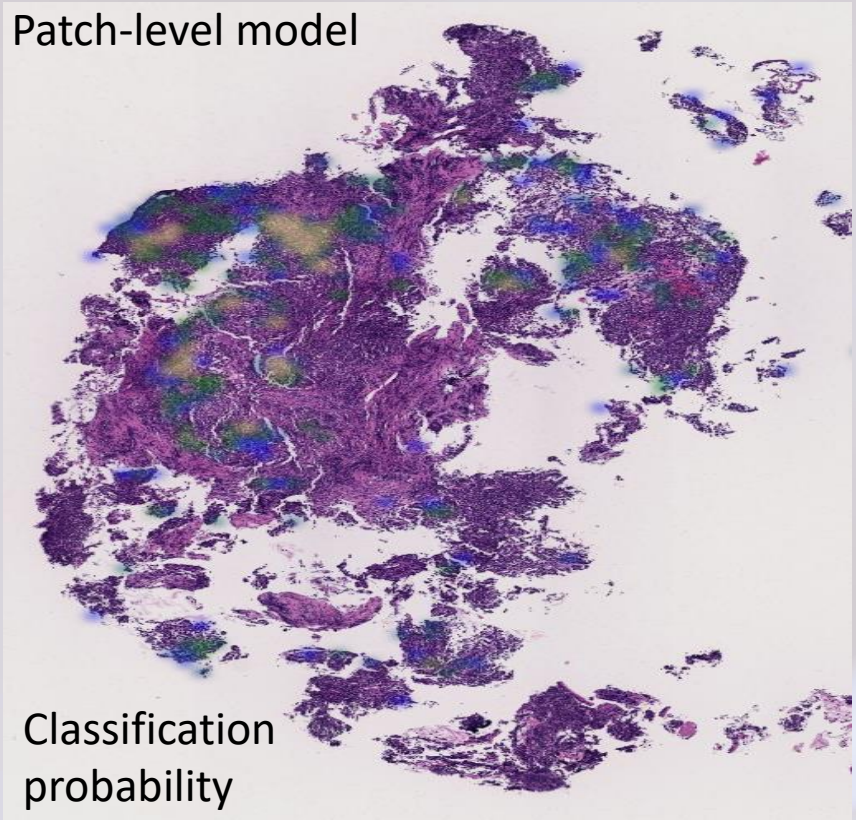
Comparison of the two approaches

No-fuss model



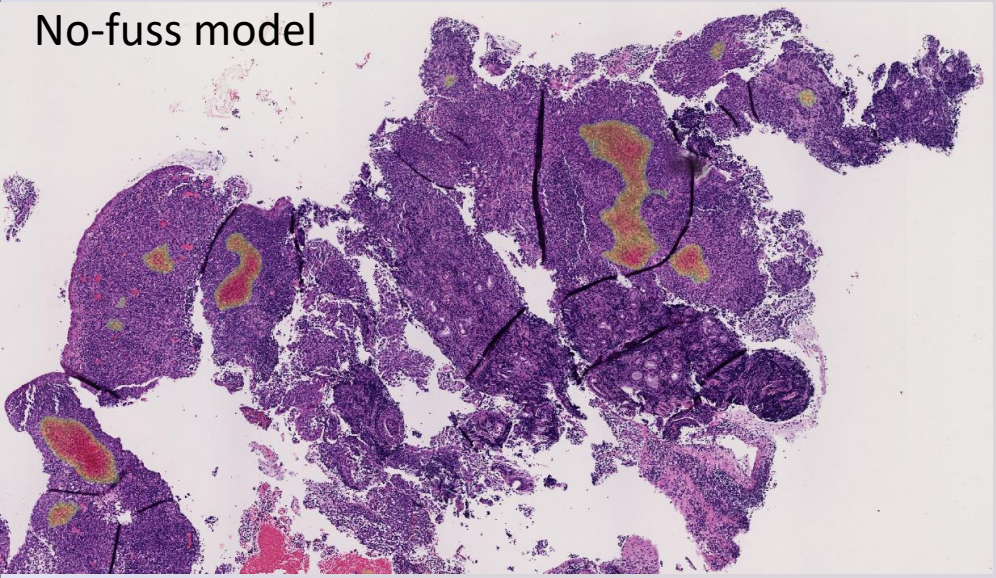
Grad-CAM output

Patch-level model

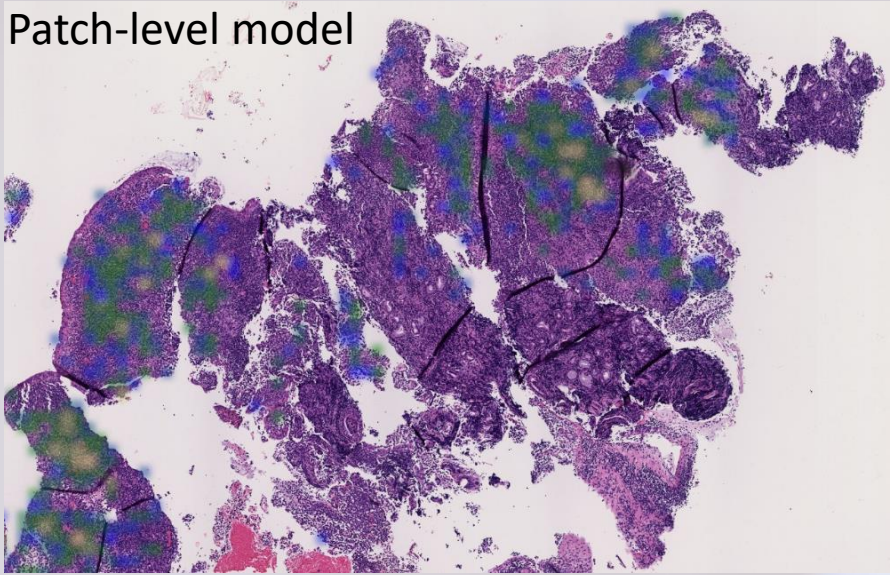


Classification probability

Comparison of the two approaches



Grad-CAM output



Classification probability

What's the Impact ?

- Improved throughput for digital pathology AI pipeline
  - Traditional : 6 months of annotation, 2 months of model training
  - Improved : ~~6 months of annotation~~, 2 months of model training

aetherAI

Embracing the Future of AI-Powered Pathology

aetherAI

info@aetherai.com