

# Not all AI is GenAI; not all GenAI is LLMs

## Vision AI for Science and Engineering Applications

Mohamed Wahib

RIKEN Center for Computational Science, Japan



16<sup>th</sup> June 2025

# Work of Many Collaborators

## **RIKEN-CCS**

Satoshi Matsuoka  
Kento Sato  
Jens Domke  
Jun Igarashi  
Aleksandr Drozd  
Emil Vatai  
Lingqi Zhang  
Peng Chen  
Sara Moukir

## **Science Tokyo**

Toshio Endo  
Rio Yokota  
Riichiro Hira  
Takayuki Nishiyo  
Chen Zhuang  
Du Wu  
Tengfei Wang  
Ivan Ivanov

## **AIST**

Liu Xin , Truong Thao Nguyen

## **Hokkaido U.**

Masaharu Munetomo, Enzhi Zhang

## **Spring-8**

Kentaro Uesugi, Takaki Hatsui

## **Kyoto U.**

Seo Akira, Yosuke Higo

## **KTH**

Artur Podobas

## **DDN**

Emmanuel Jeannot

## **BSC**

Jesus Labarta, Marc Clasca, Mario Acosta, Kai Keller

## / **Nvidia/Koç U.**

Didem Unat, Ilyas Turimbetov

## **Télécom SudParis**

François Trahay

## **ANL**

A. Dubey, I. Cloet, X. Wu, J. O'Neal, Klaus

## **UC Berkeley**

Saathvik Selvan, Connor Chen

## **VirginiaTech**

Johann Rudi, Wuchun Feng

## **ORNL**

Xiao Wang, Issac Lyngaas

## **Emory U.**

Katherine Keegan

## **Intel**

Balazs Gerofi

## **HKBU**

Ameilie Zhou, Yuxin Wang

## **NUIST**

Xue Yu

## **A\*STAR**

Luo Tao



# Why at all is AI Relevant to Science/Engineering

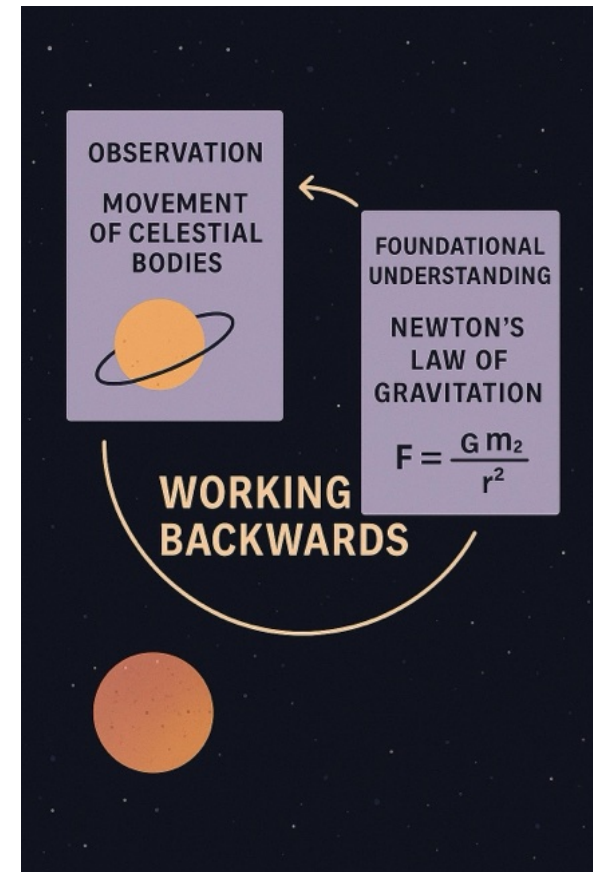
Astronomers observed **regular, predictable motion** of celestial bodies



*What governs this motion?*



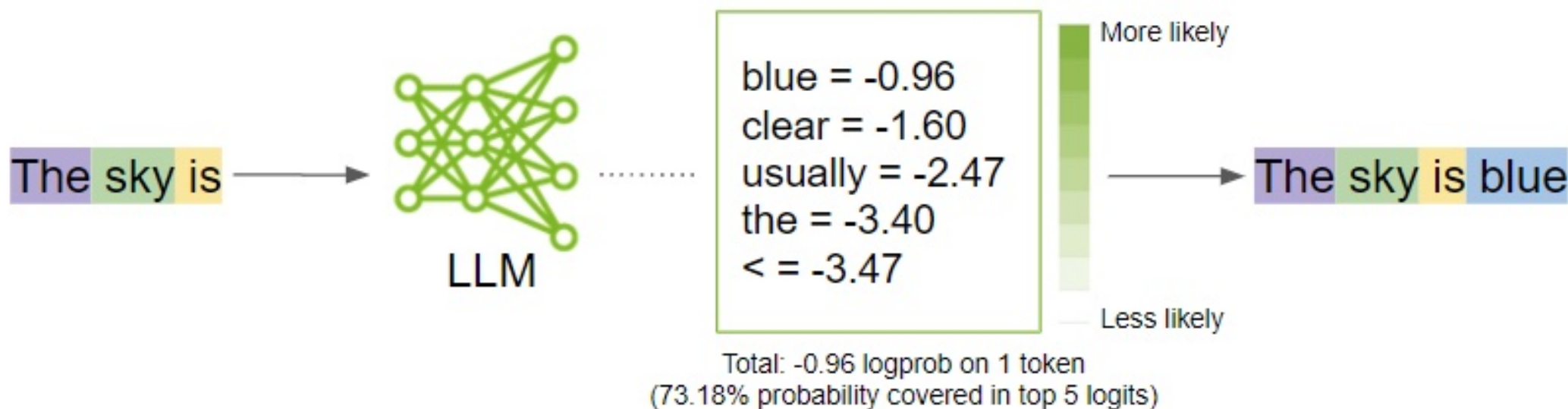
Newton worked backward from these observations and proposed: A **universal force** acts between all masses



Science is reverse engineering nature → (mostly) by observing patterns

# AI (Machine Learning) is Good at Finding Patterns

When you use ChatGPT, you are asking the model to **predict from patterns** it learned

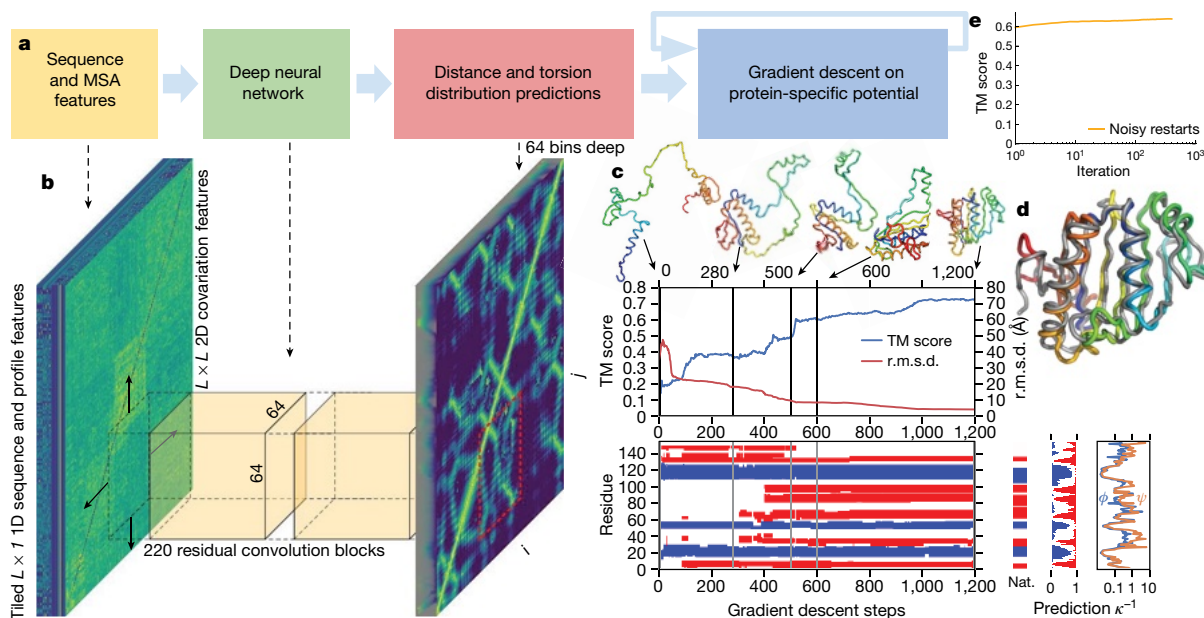


AI in Science: use AI pattern recognition capability to understand nature

# Breakthroughs in AI-based Science: Finding Patterns

## nature

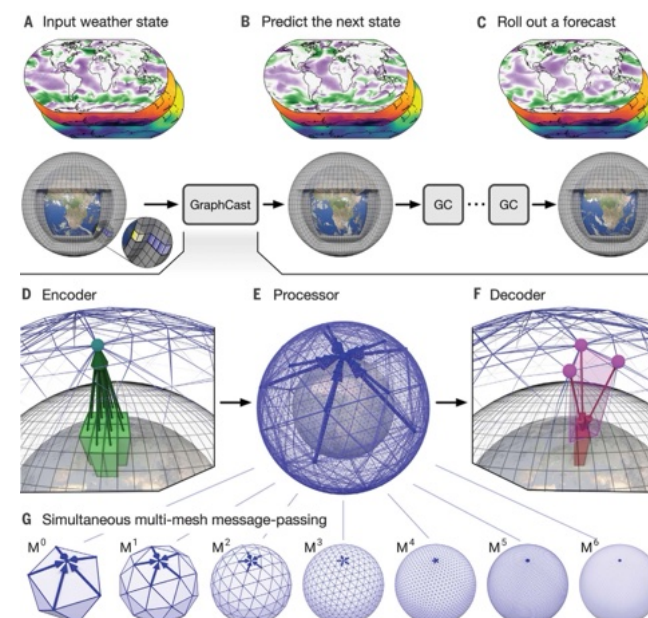
Improved protein structure **prediction** using potentials from deep learning



AlphaFold

## Science

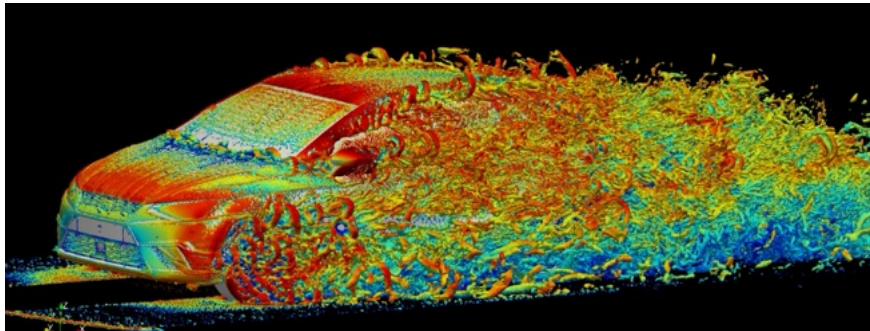
Learning skillful medium-range global weather **fore-casting**



GraphCast

# AI for Spatial and Spatio-temporal Data

# Complex Spatial-Temporal Data Is Common for Science & Engineering



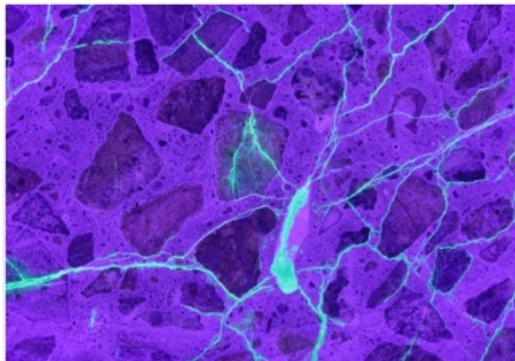
Fluid dynamics



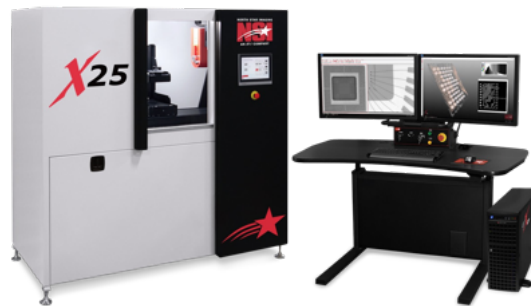
Climate/Weather Prediction



Satellite Remote Sensing



X-ray/electron microscopy



Industrial inspection



Synchrotron beamline



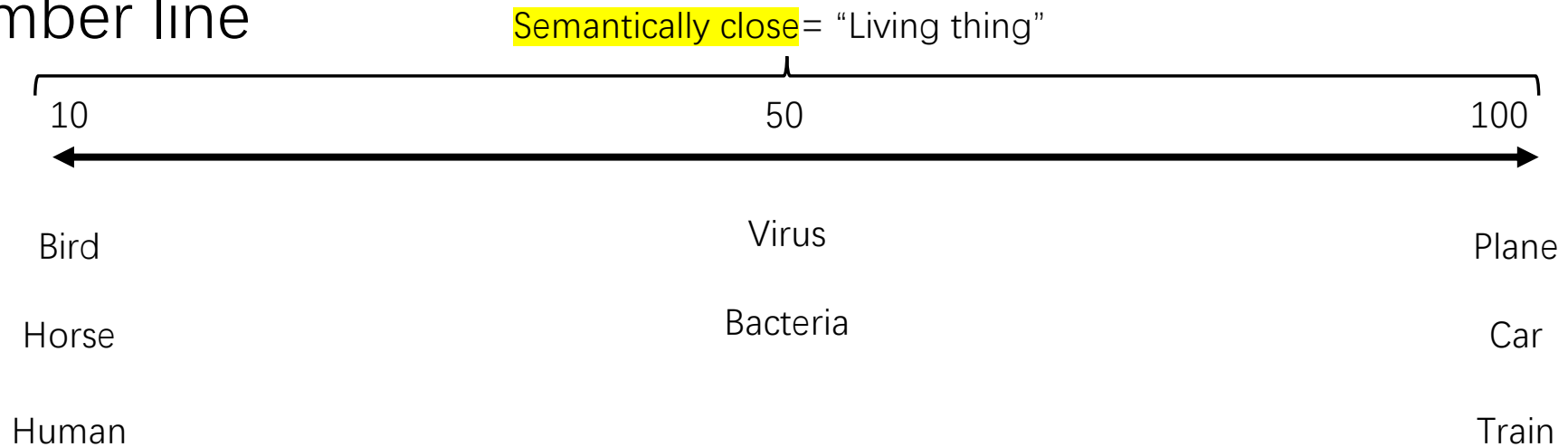
# Multi-dimensional Images in Science & Engineering

| Type                           | Resolution                                            | Tokens/Sample<br>Patch = 162/163 | Dataset Sizes | Dimensions<br>(Channels)                    |
|--------------------------------|-------------------------------------------------------|----------------------------------|---------------|---------------------------------------------|
| Weather\Climate Simulations    | 100s <sup>3</sup><br>10s channels<br>(ERA5 dataset)   | ~ 300K                           | ~10 PB        | 3 Spatial +<br>1 Temporal +<br>(N Channels) |
| Satellite Images               | 1000s <sup>3</sup><br>10s channels                    | ~5M                              | ~ 10s TB      | 2 Spatial +<br>1 Temporal +<br>(N Channels) |
| Microscopic<br>(Ex: Pathology) | 100K <sup>2</sup>                                     | ~100Ks<br>(4x4 patch)            | ~ 10s TBs     | 2 Spatial +<br>(1 or N Channels)            |
| Video                          | 100s <sup>2</sup><br>~ Hours (24 f/s)<br>(YouTube-8m) | ~1M                              | ~1 PB         | 2 Spatial +<br>1 Temporal +<br>(N Channels) |
| X-Ray CT<br>(Ex: SP- $\mu$ CT) | ~8-12K <sup>3</sup><br>>163 new beam                  | ~1B                              | ~100s TB      | 3 Spatial +<br>(1 Channel)                  |
| MRI<br>(Ex: dMRI)              | ~4K <sup>3</sup><br>(sub 5-micron)                    | ~ 30M                            | ~ 10s TB      | 3 Spatial +<br>(N Channels)                 |

# What is a Vision Transformer?

# Self-attention TL;DR: Text and Embeddings

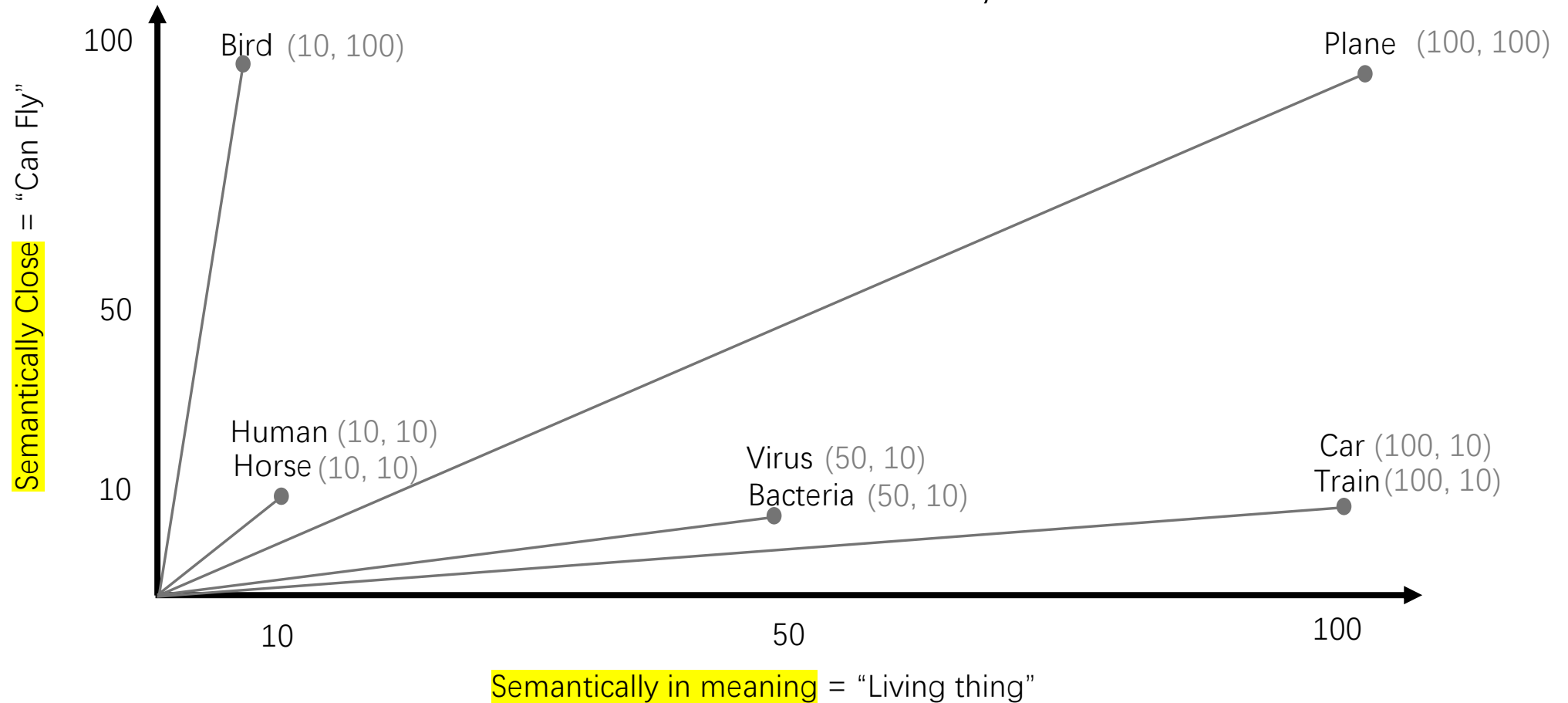
- Modeling language:
  - Find how words relate to and affect the meaning of other words
- Use a single number to represent each word, words **semantically close in meaning** are close to each other on the number line



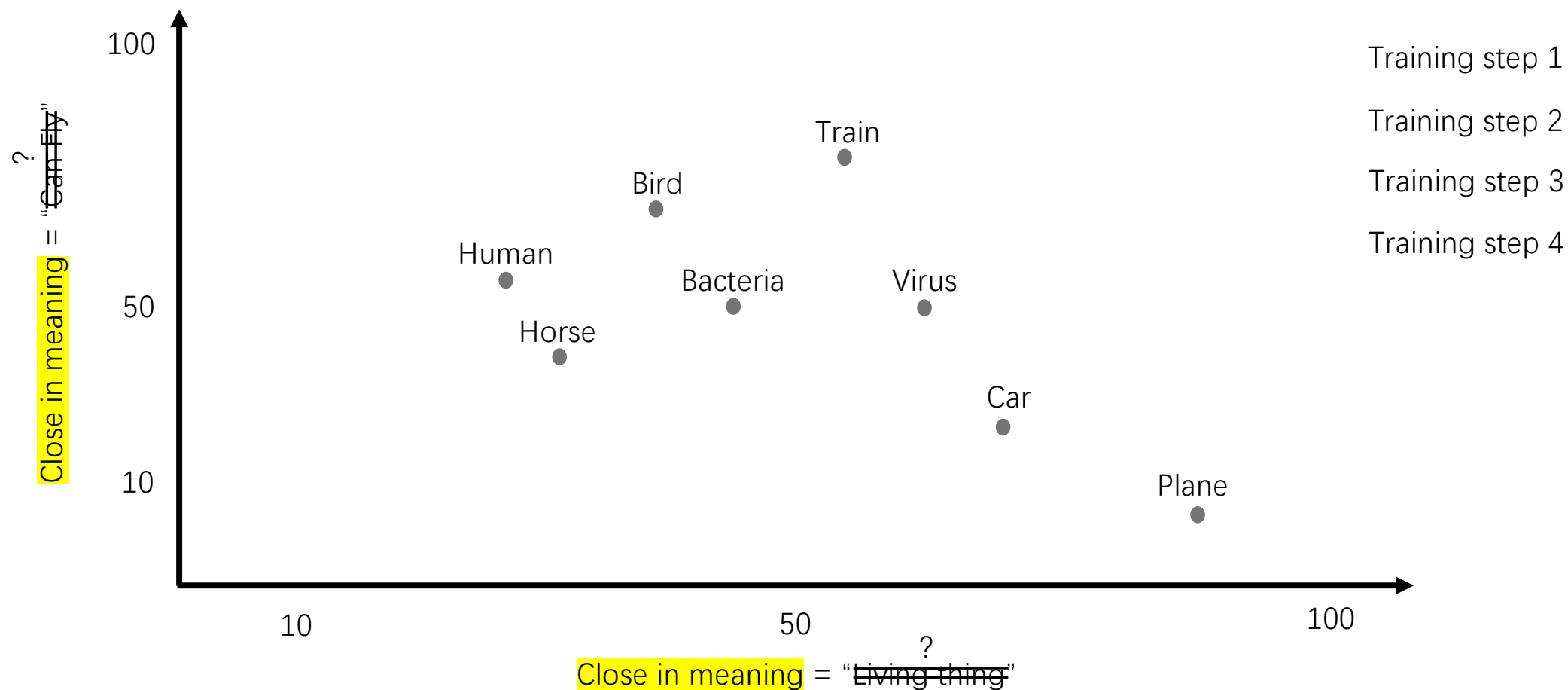


# Text and Embeddings

- But words can be close in one attribute, and not others

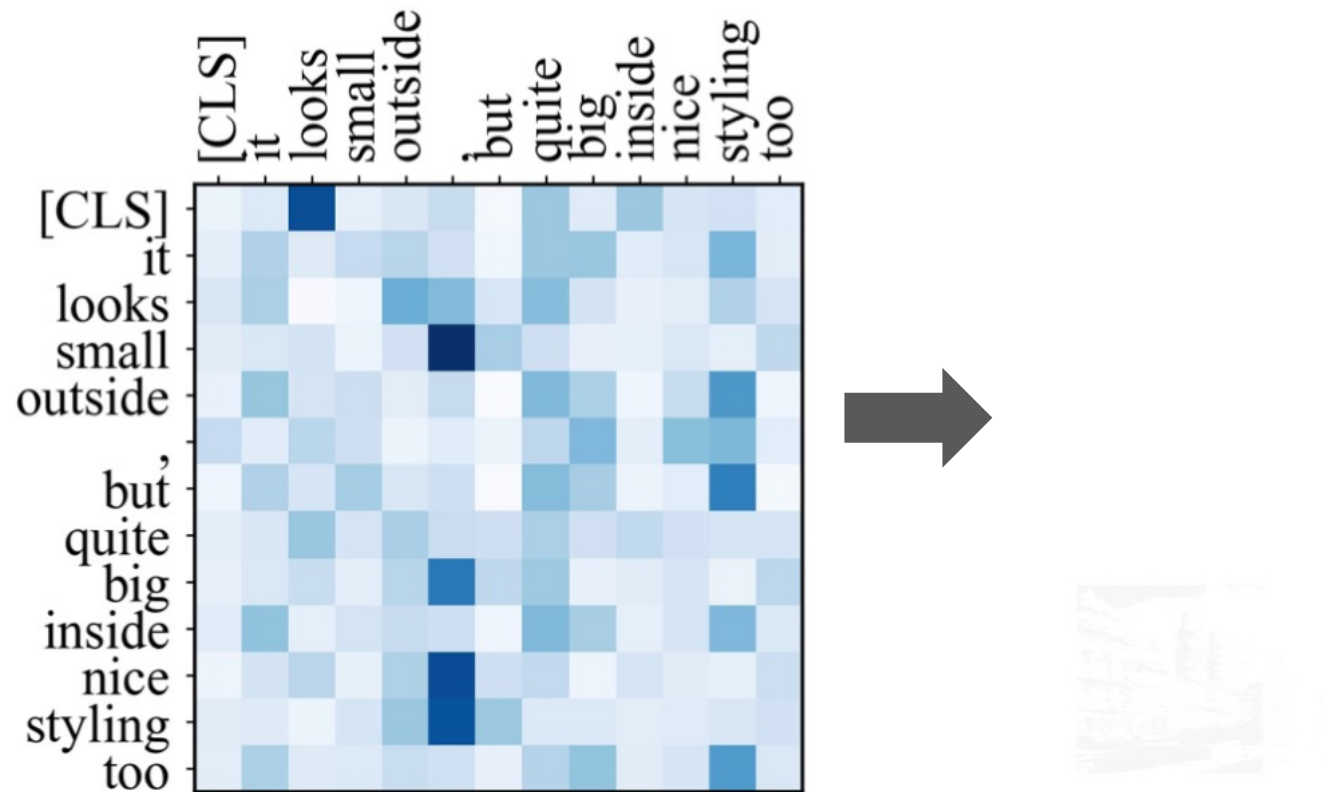


LLMs: a discovery about the regularity of the semantics of human communication, rather than a discovery about the intelligence of neural networks



# Transformer: Generalizing to Any Sequence of Tokens

- Input is a **sequence of tokens**
- Tokens can be anything (ex: words, flattened patches of images\*)



\*Dosovitskiy et al. in [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale](#)

# ViT and Inductive Bias

- Universal law of approximation



Any continuous function  $f$

$$f : R^N \rightarrow R^M$$

Can be realized by a fully connected network with hidden layer(s)

- Bound on approximation error



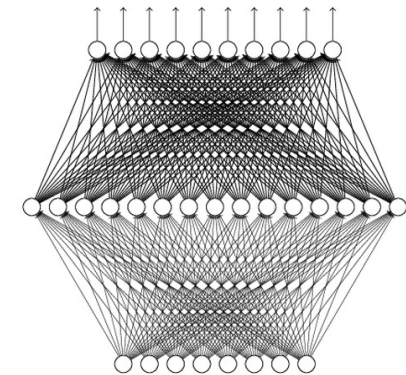
- Generalization error



- Optimizer error



- FC Networks not enough (in Practice)

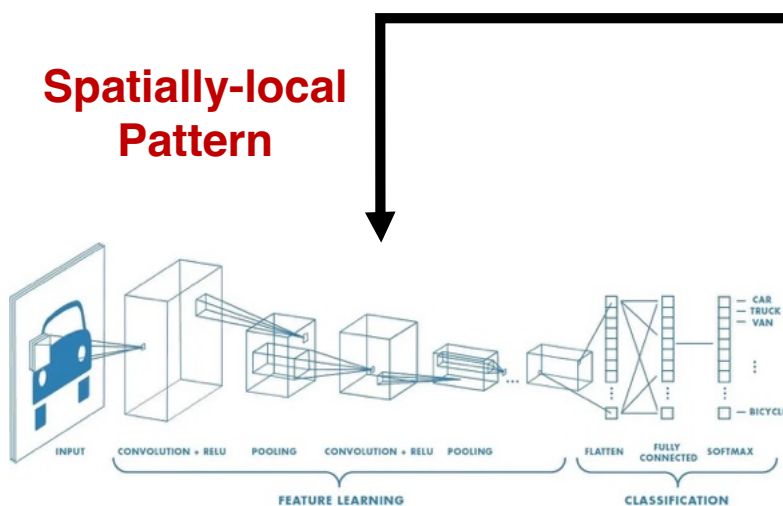


Reference for the reason:

<http://neuralnetworksanddeeplearning.com/chap4.html>

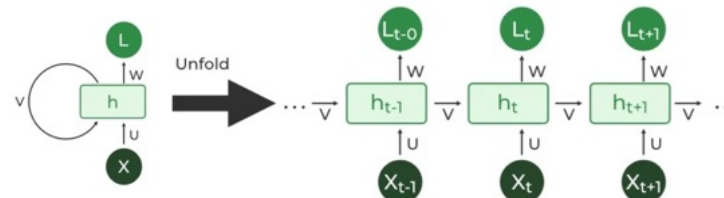
# ViT and Inductive Bias

- Inductive bias:
  - Assumptions a learning algorithm uses to predict on unseen data
- Neural Networks → sophisticated pattern recognition
  - Add network elements to “work better” with the pattern in the data



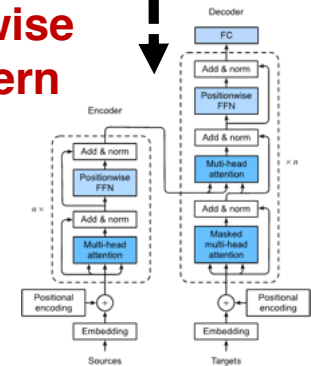
**Convolution Neural Network**

**Temporal Pattern**



**Recurrent Neural Network**

**Pairwise Pattern**



**Transformers  
(w/ self-attention)**

# ViT and Inductive Bias

- From the Inductive bias POV
  - ViT might not seem to make much sense
  - Receptive field is entire image
    - Every piece of the image attends to all other pieces of the image
- **Positives**
  - Model learns EVERYTHING → can ingest massive datasets
- **Negatives**
  - Model learns EVERYTHING → expensive; finds irrelevant patterns

# ViT Issues vs. Text Transformer Issues

- Transformer consume sequence of tokens
  - Fitting to the nature of text
- Text tokens: atomic semantically distinct, rich in information,
  - Visual tokens: geometrically related and sparse in semantics
- **Loss of spatial hierarchy information becomes more pronounced**
  - When working with high-resolution or high-dimension images



# ViT Challenges

- ❶ Long sequence (when ingesting all image elements at high res)
- ❷ Shifting bottleneck (elements outside the Transformer encoder)
- ❸ Tokenization (managing temporal dimension)
- ❹ Different parallelism in training (vs. text Transformer)
- ❺ Multi-modality

# Two Main Principles on Solving Real-world Problems

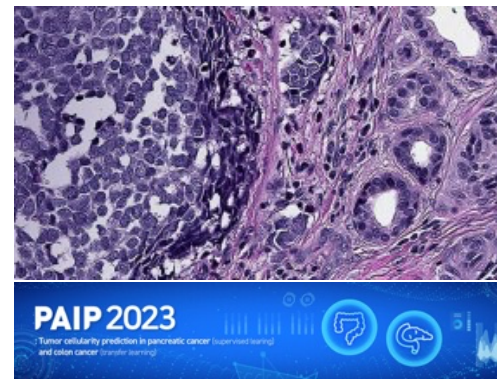
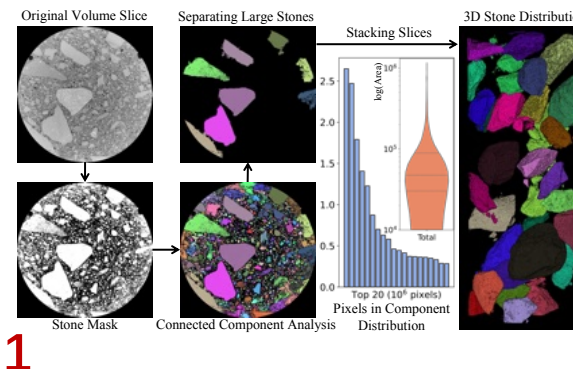
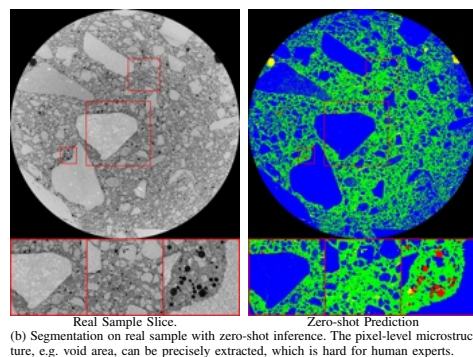
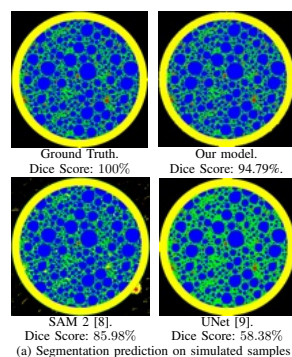
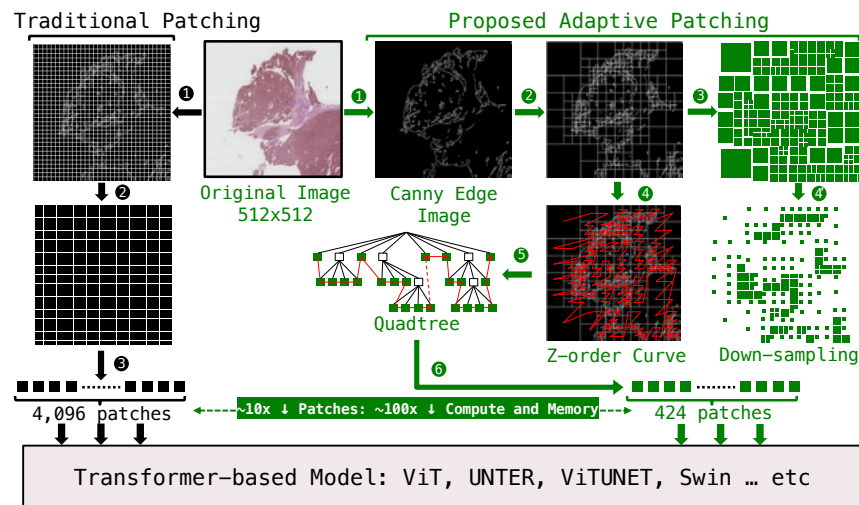
- ① Start from Scientific Inquiry; Work Back to Solution
- ② AI is a toolbox; always pick the right tool



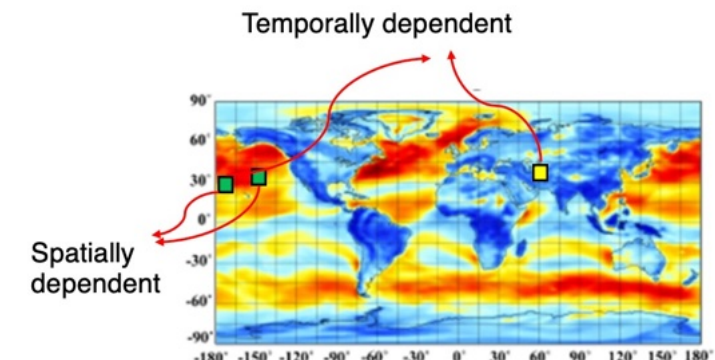
# Vision Transformers Use in Real-world Problems

Vision Transformers used in production, Examples:

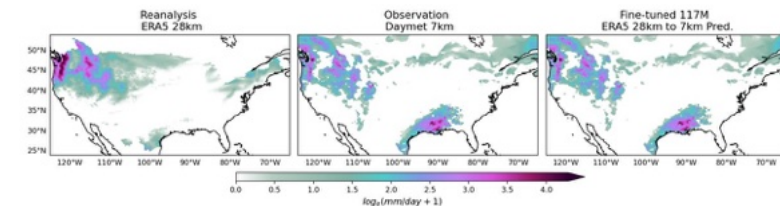
a) microscopic pathology, b) X-ray CT road samples, c) weather prediction



3



2



1



2



3

6

# ViT Challenges

- ① Long sequence (when ingesting all image elements at high res)
- ② Shifting bottleneck (elements outside the Transformer encoder)
- ③ Tokenization (managing temporal dimension)
- ④ Different parallelism in training (vs. text Transformer)
- ⑤ Multi-modality

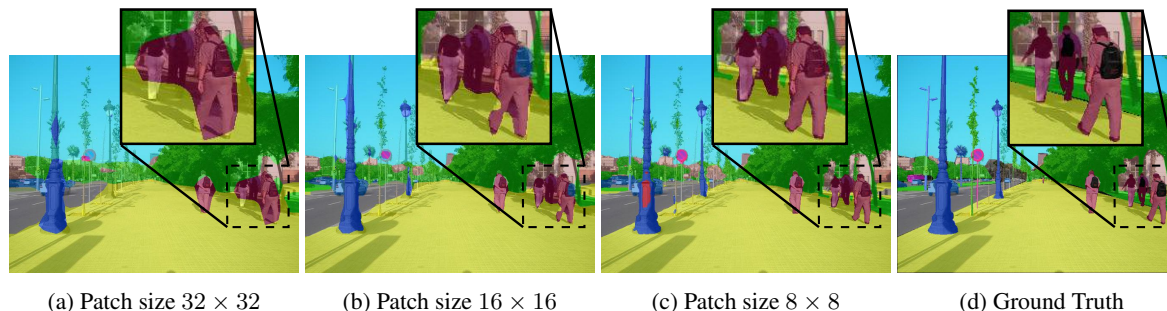
# LLMs for 3D Segmentation

## ➤ Using LLMs for Vision (ex: Vision Transformers)

- Because of self-attention, the receptive field is the entire image!
- Split image to patches (ex: 16x16)
- Feed patches to LLMs

## ➤ Segmentation

- Larger patches → model learns global meaningful segmentation; produces poor boundaries
- Smaller patches are qualitatively better
- 4x4 patches for 4K<sup>3</sup> 3D image = 1,000,000,000 tokens/image



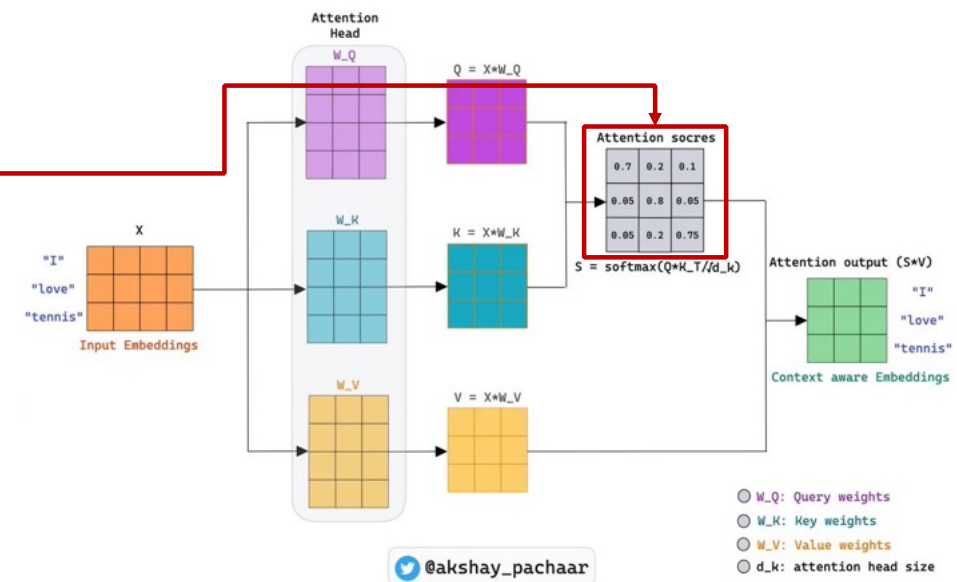
### Impact of Model Patch Size on the Segmentation Maps\*

\* Strudel et al. Segmenter: Transformer for Semantic Segmentation, ICCV'21 (<https://arxiv.org/pdf/2105.05633.pdf>)

# Longer Sequence: a Challenge

- The longer the sequence, the more the **context** that can be extracted
  - Ex: feeding an LLM entire books, library of papers, RAG, or **segmentation**
  - GPT-4-turbo → 128,000 tokens – GPT4-32k → 32,768 tokens (1 Token =  $\frac{3}{4}$  Word)

- Compute and memory cost  $\propto \text{sequence}^2$



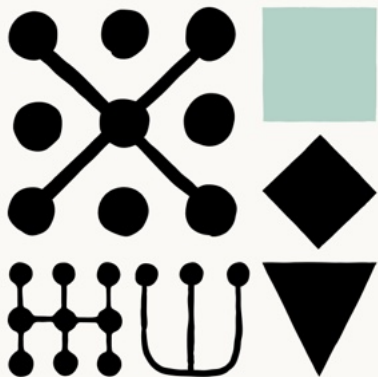
# Methods for Longer Sequence

ANTHROPIC

[Claude](#) [API](#) [Solutions](#) [Research](#) [Commitments](#) [Learn](#) [News](#)

[Try Claude](#)

Engineering at Anthropic



## How we built our multi-agent research system

Published Jun 13, 2025

Our Research feature uses multiple Claude agents to explore complex topics more effectively. We share the engineering challenges and the lessons we learned from building this system.

*“Multi-agent architectures effectively scale token usage for tasks that exceed the limits of single agents.”*



# Compatibility of Longer Sequence Approaches in Training

|                           | Alternative for Attention | Hierarchal Training | Attention Approximation | Cache Blocking | Reduce Tokens | Sequence Parallelism |
|---------------------------|---------------------------|---------------------|-------------------------|----------------|---------------|----------------------|
| Alternative for Attention |                           | ?                   | ✗                       | ?              | ✓             | ?                    |
| Hierarchal Training       | ?                         |                     | ✓                       | ✓              | ✓             | ✓                    |
| Attention Approximation   | ✗                         | ✓                   |                         | ?              | ✓             | ✗                    |
| Cache Blocking            | ?                         | ✓                   | ?                       |                | ✓             | ✓                    |
| Reduce Tokens             | ✓                         | ✓                   | ✓                       | ✓              |               | ✓                    |
| Sequence Parallelism      | ?                         | ✓                   | ✗                       | ✓              | ✓             |                      |



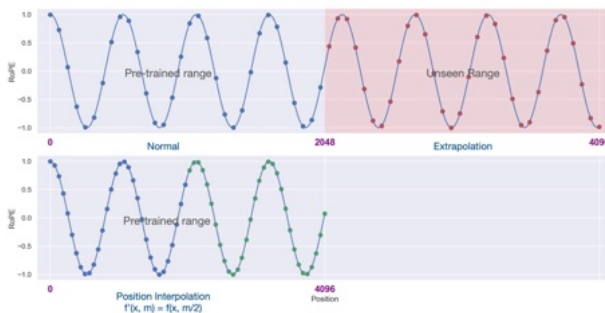
# Methods for Longer Sequence

Alternative for Attention  
 Hierarchical Training  
 Attention Approximation  
 Cache Blocking  
 Sequence Parallelism  
 Reduce Amount of Tokens

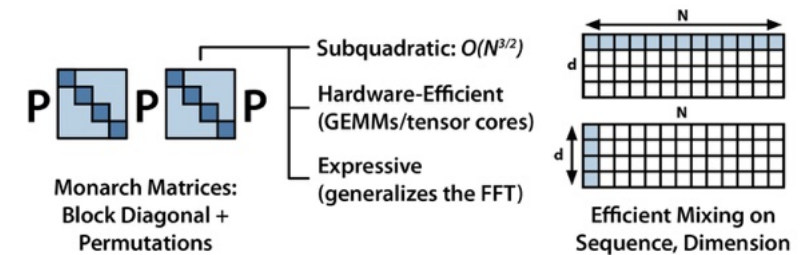
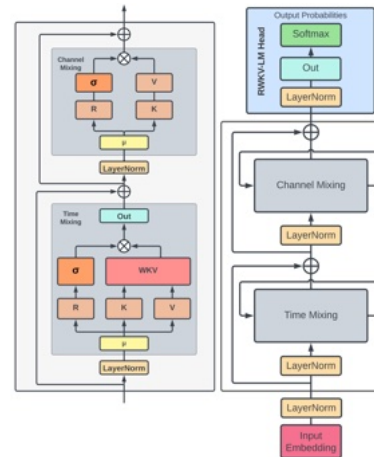
## ➤ Alternative Mechanism for Attention

**Monarch:** use convolution to compute Attention  
 (FFT for convolution)

**Positional Interpolation:** downscale vs. extrapolating  
 (Extend window size)



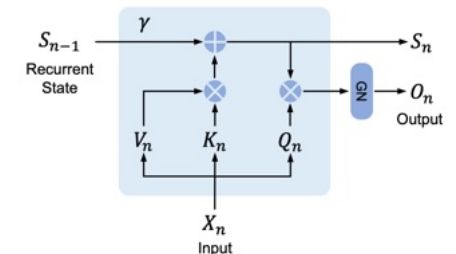
**RWKV:** Transformer in training;  
 RNN in inference  
 (Linear attention)



**RetNet:** a retention mechanism for attention modeling  
 (parallel recurrency)

$$O = \text{GN}((QK^T \odot D)V)$$

The diagram shows the RetNet equation with inputs  $Q$ ,  $K$ , and  $V$  from input  $X$ , and a recurrent state  $S_{n-1}$  from the previous step.



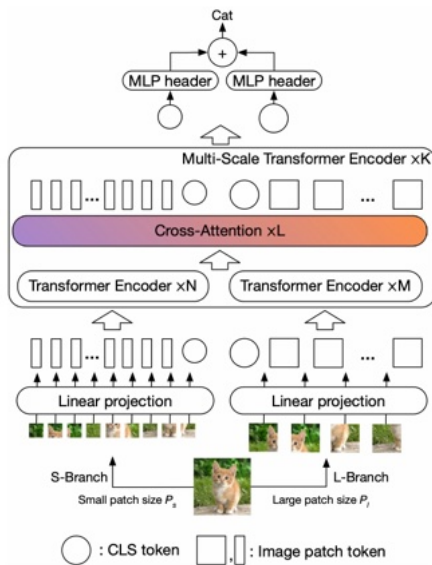
# Methods for Longer Sequence

Alternative for Attention  
Hierarchical Training  
Attention Approximation  
Cache Blocking  
Sequence Parallelism  
Reduce Amount of Tokens

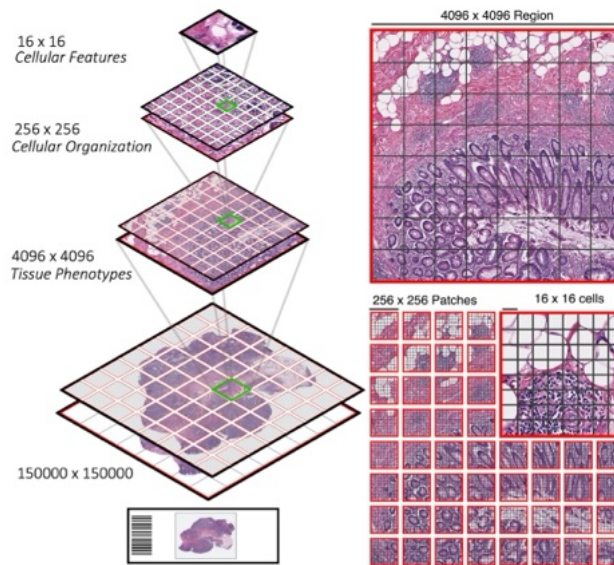
## ➤ Hierarchical Training

- Train multiple transformers at different levels of abstraction
- The transformer at the lowest abstraction level trains on the shortest sequence segments.
- The transformer at the next higher level uses the previous level outputs as additional input to train on longer segments.

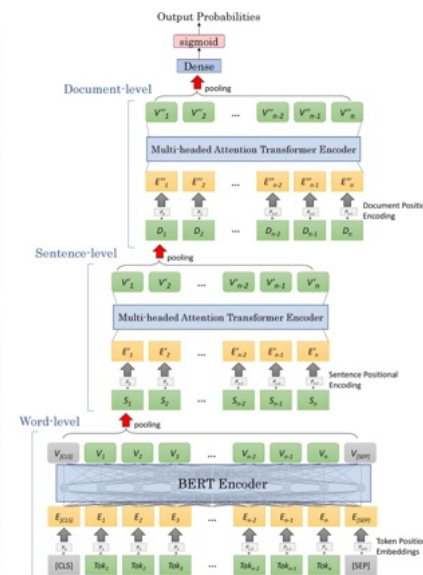
### CrossViT



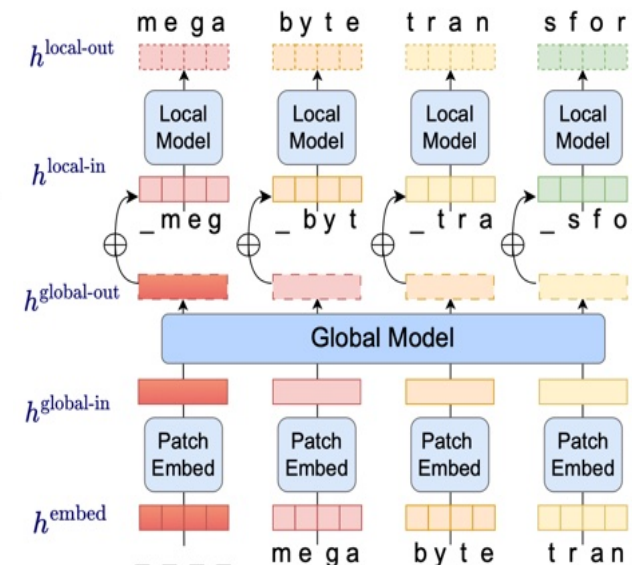
### Hierarchical ViT



### Three-level Transformer



### MEGABYTE



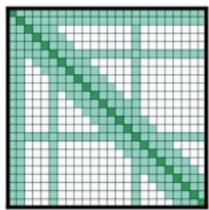
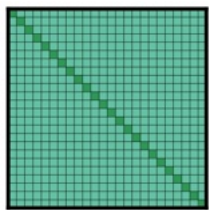
# Methods for Longer Sequence

Alternative for Attention  
Hierarchical Training  
Attention Approximation  
Cache Blocking  
Sequence Parallelism  
Reduce Amount of Tokens

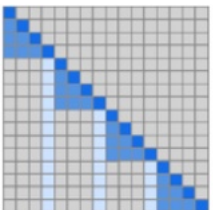
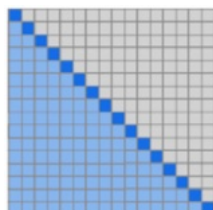
## ➤ Attention Approximation

- Approximate self-attention operation through sparse sampling, lox-rank approx., infrequent update ...
- Note: “sparse” when talking about LLMs now mean **sparse sequence**, not sparse model

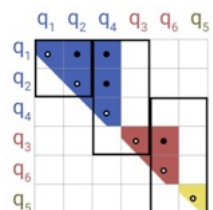
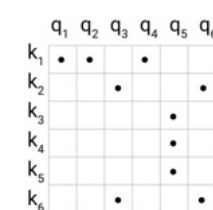
### Longformer



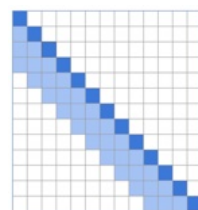
### Sparse Transformer (Hash attention)



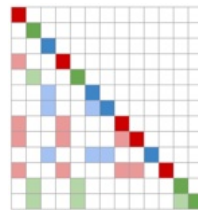
### Reformer



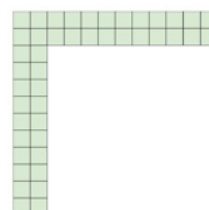
### Routing Transformer



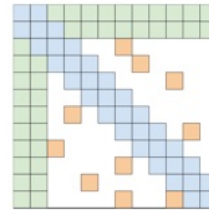
Local attention



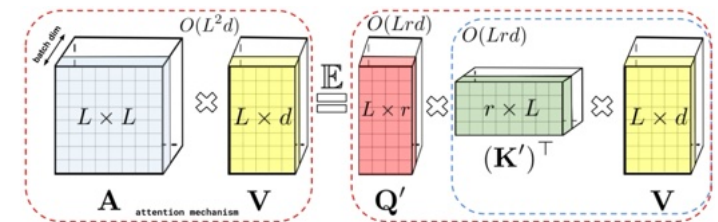
### Big Bird



Global attention



### Performers



### Linear Transformers

$$A_l(x) = V' = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V \quad \Rightarrow \quad V'_i = \frac{\sum_{j=1}^N \text{sim}(Q_i, K_j) V_j}{\sum_{j=1}^N \text{sim}(Q_i, K_j)}.$$

### In-frequent Update

$$s_i = \text{dot}(q, k_i), \quad s'_i = e^{s_i}, \quad \text{attention}(q, k, v) = \frac{\sum_i v_i s'_i}{\sum_j s'_j}.$$

### LazyFormer

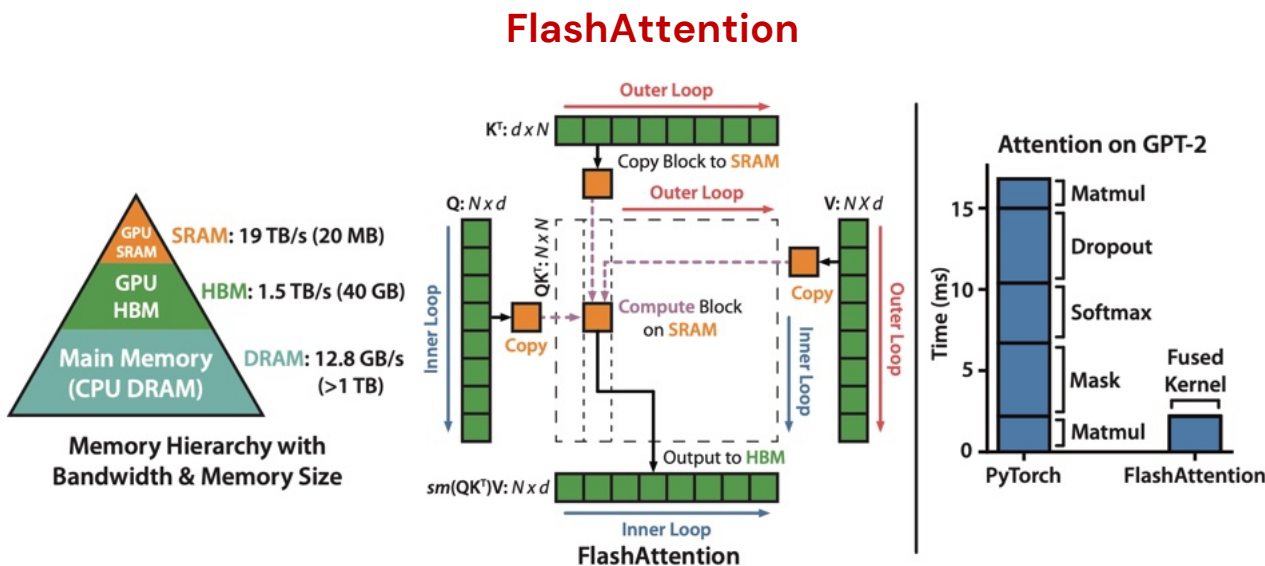
**Remove dropout in self-attention.**  
**Wider Layers.**

# Methods for Longer Sequence

Alternative for Attention  
Hierarchical Training  
Attention Approximation  
Cache Blocking  
Sequence Parallelism  
Reduce Amount of Tokens

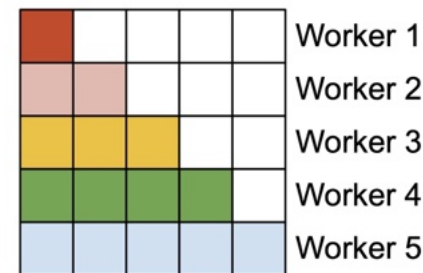
## ➤ Cache Blocking

- No approximation
- Can support longer sequences by blocking the attention matrix in scratchpad memory
- Aggregate amount of work stays the same → **support longer sequence, but not faster**



## FlashAttention2

### Forward pass



### Backward pass



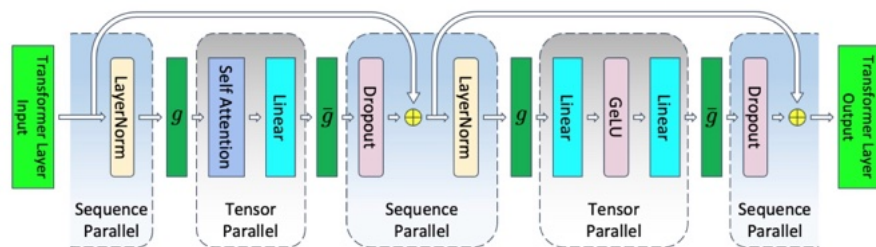


# Methods for Longer Sequence

Alternative for Attention  
Hierarchical Training  
Attention Approximation  
Cache Blocking  
Sequence Parallelism  
Reduce Amount of Tokens

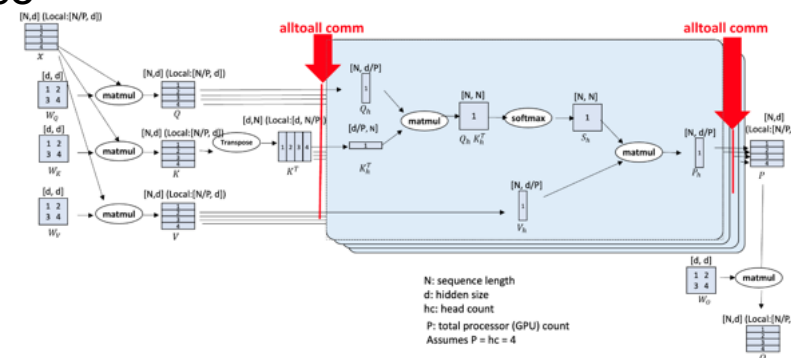
- Sequence/context Parallelism
  - Distributing long sequences among GPUs as short contiguous segments
  - Communication overhead due to token inter-dependence

## Nvidia Sequence Parallelism



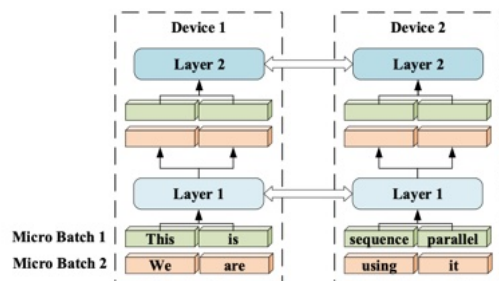
\*Not really sequence parallelism; parallelizes Dropout and LayerNorm

## MS DeepSpeed-Ulysses



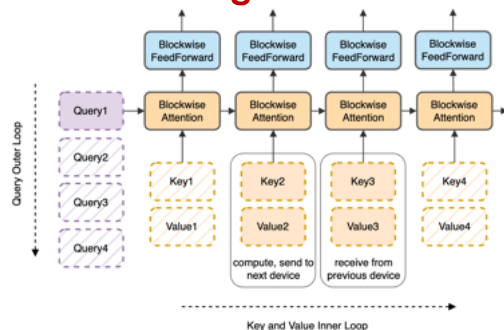
\*Not really sequence parallelism; full attention is still at each worker

## Long Sequence

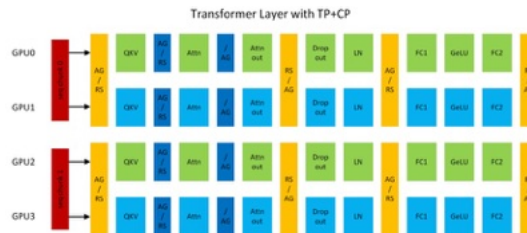


\*No significant speedup

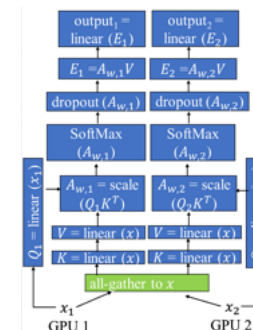
## Ring Attention



## Nvidia Context Parallel



## Fully Distributed Sequence



# Methods for Longer Sequence

Alternative for Attention  
Hierarchical Training  
Attention Approximation  
Cache Blocking  
Sequence Parallelism  
Reduce Amount of Tokens

## ➤ Reduce Amount of Tokens

- Current practice: divide input to tokens, feed all tokens to the model
- Feed the model less tokens: BEFORE tokens are ingested or DURING passing through the model

### (Learned) Token Pruning

|                |                                                                        |             |
|----------------|------------------------------------------------------------------------|-------------|
| Layer 1        | This is the best restaurant, and I will be returning for another meal. | 15 tokens ▼ |
| Layer 4        | This is the best restaurant, and I will be returning for another meal. | 11 tokens ▼ |
| Layer 8        | This is the best restaurant, and I will be returning for another meal. | 4 tokens ▼  |
| Layer 12       | This is the best restaurant, and I will be returning for another meal. | 2 tokens ▼  |
| Classification | Positive Sentiment                                                     |             |

### Adaptive Patching (ViT)

Patch the images in a "smarter" way

# Tumor Cellularity Prediction in Pancreatic Cancer and Colon Cancer

➤ Very high resolution (up to 100,000 x 100,000 pixels)

➤ Used in pathology

➤ Ex: PAIP dataset

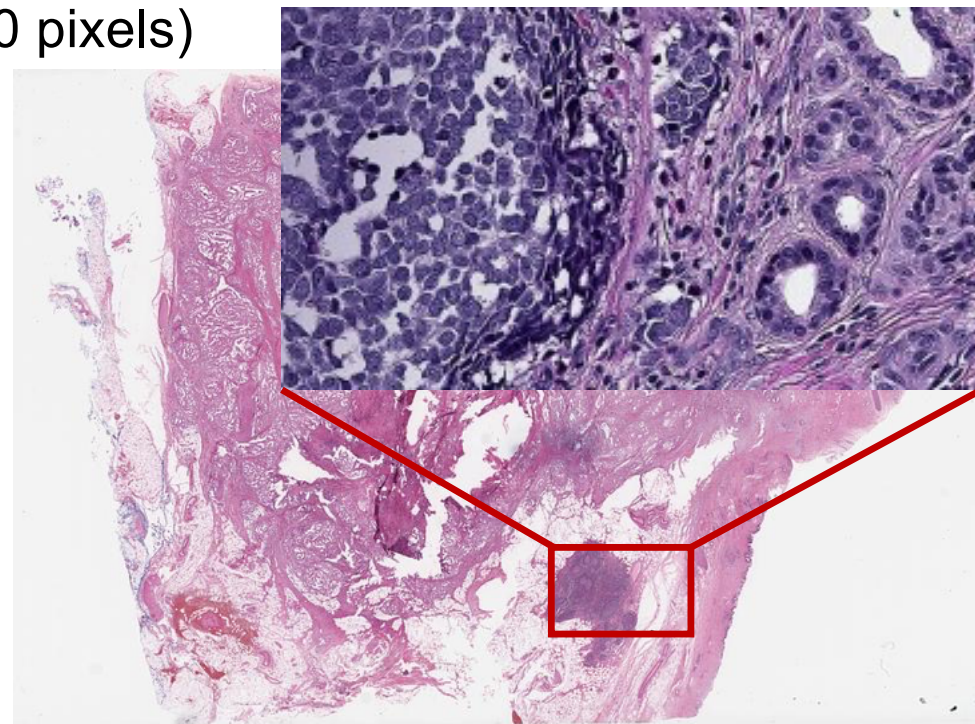
➤ Pancreas

➤ **Diagnostic:** Perineural Invasion

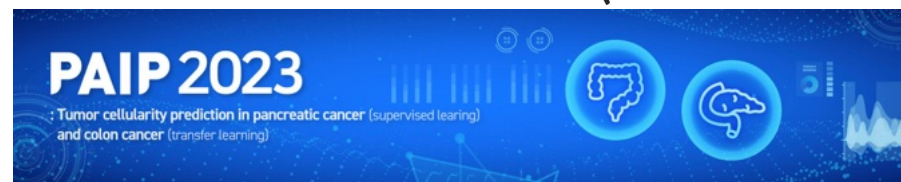
➤ Segmentation with Vision Transformer (ViT)

➤ Challenge:

**PAIP 2023: Tumor cellularity prediction in pancreatic cancer and colon cancer (transfer learning)**

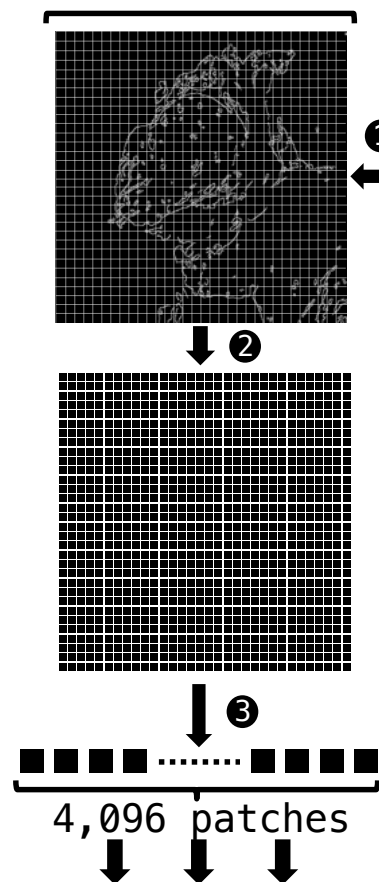


\* E. Zhang et al. Adaptive Patching for High-resolution Image Segmentation with Transformers, SC'24

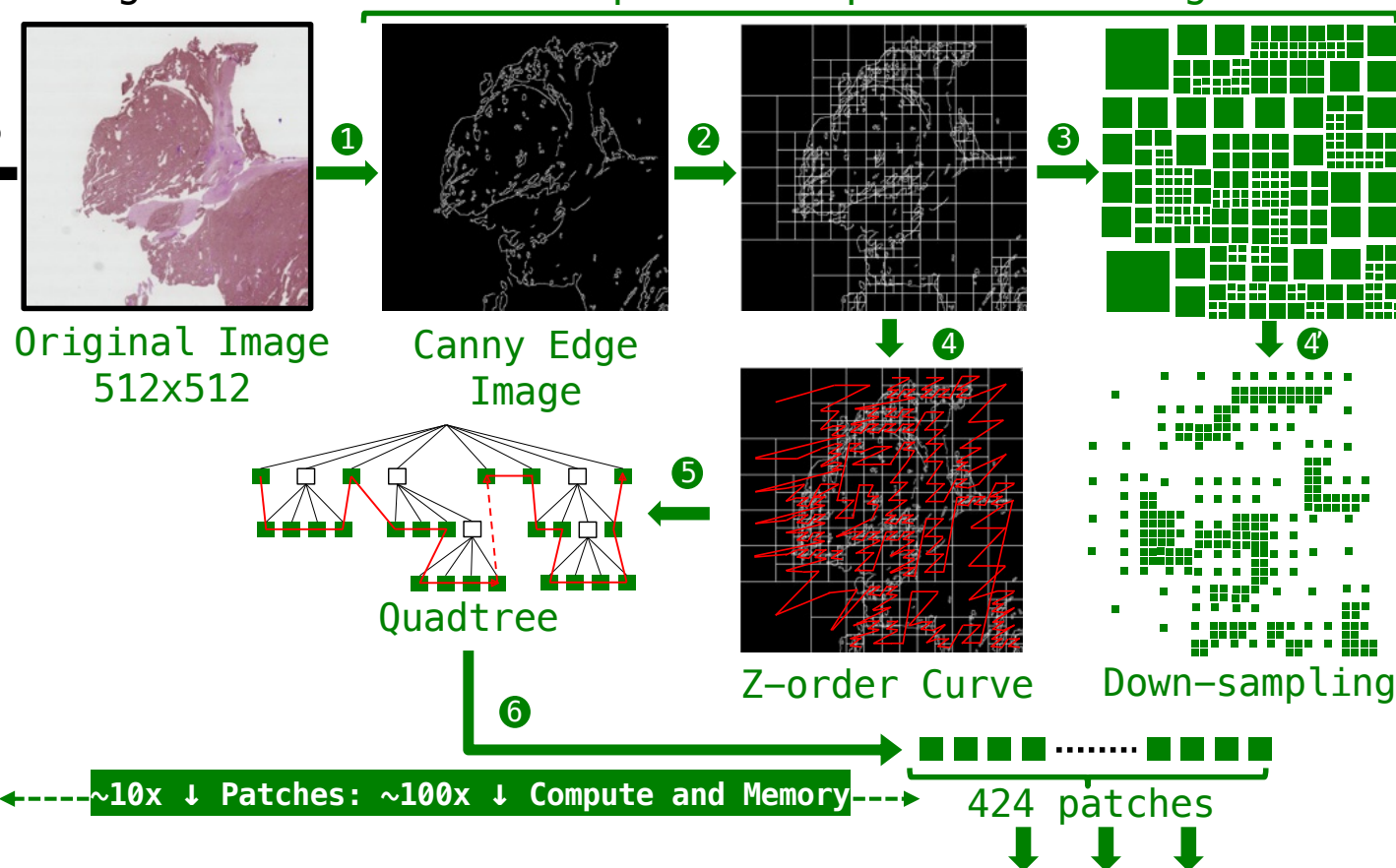


# Adaptive Patching: Ingest Only the Data that Matters

Traditional Patching

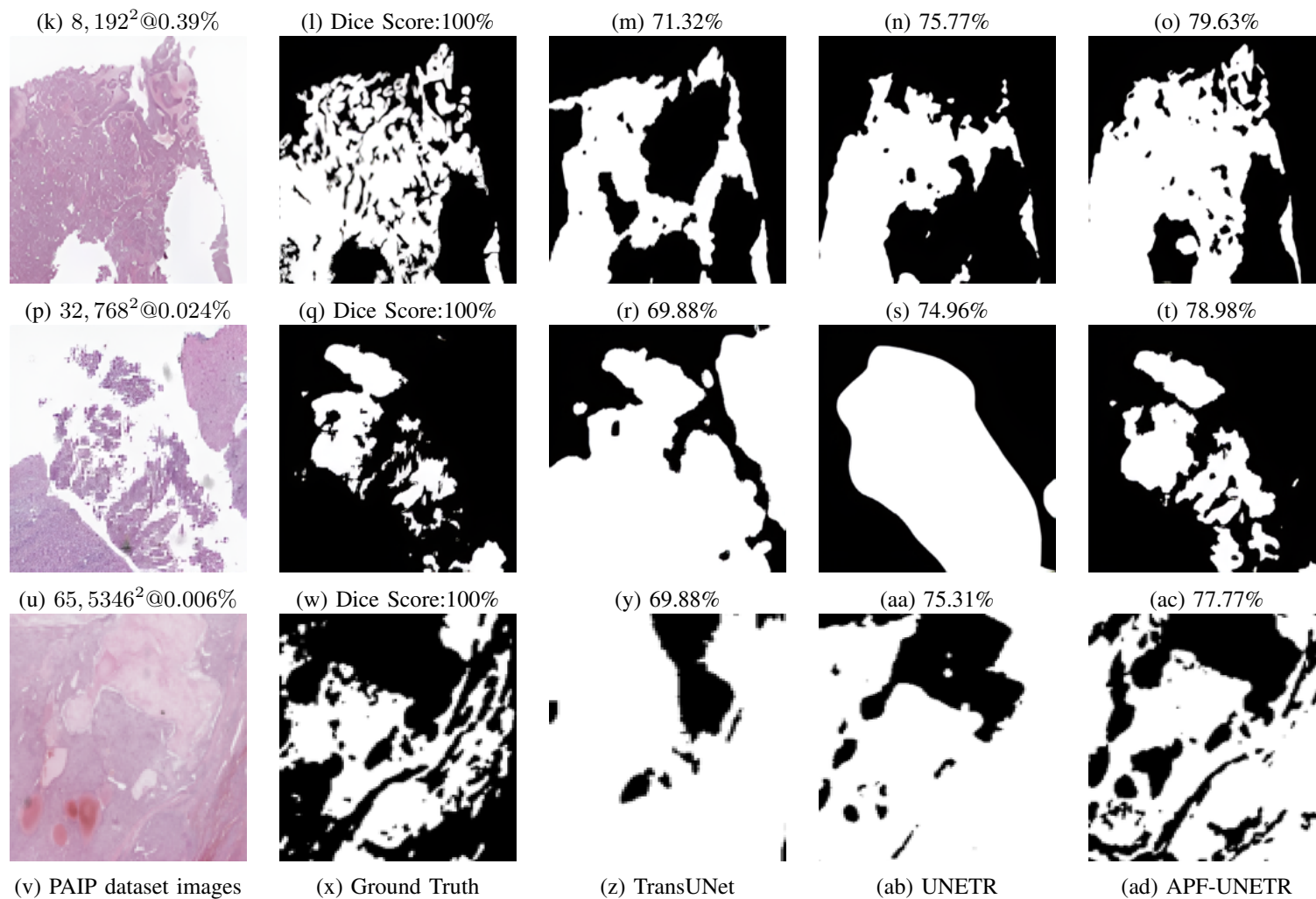


Proposed Adaptive Patching



Transformer-based Model: ViT, UNTER, ViTUNET, Swin ... etc





# Results: Speedup

| Resolution                   | Model-Patch                | Sec/Image         | Sequence Length | Quadtree Depth | Dice Score (%) | Speedup (Sec/Image) | Speedup (Time to Convergence) |
|------------------------------|----------------------------|-------------------|-----------------|----------------|----------------|---------------------|-------------------------------|
| 512 × 512<br>1 GPU           | <b>APF-4</b><br>UNETR-4    | 0.06495<br>0.4863 | 1,024<br>16,384 | 7<br>-         | 77.88<br>77.31 | 7.48×               | 12.71×                        |
| 1,024 × 1,024<br>8 GPUs      | <b>APF-8</b><br>UNETR-8    | 0.14284<br>1.0863 | 1,024<br>16,384 | 7<br>-         | 75.63<br>75.72 | 7.6×                | 12.92×                        |
| 4,096 × 4,096<br>128 GPUs    | <b>APF-16</b><br>UNETR-32  | 0.32231<br>1.8613 | 2,116<br>16,384 | 8<br>-         | 75.74<br>75.77 | 5.77×               | 9.8×                          |
| 8,192 × 8,192<br>256 GPUs    | <b>APF-16</b><br>UNETR-64  | 1.1613<br>2.6618  | 2,116<br>16,384 | 9<br>-         | 76.13<br>75.27 | 2.29×               | 3.89×                         |
| 16,384 × 16,384<br>512 GPUs  | <b>APF-32</b><br>UNETR-128 | 1.7613<br>5.1179  | 1,024<br>16,384 | 9<br>-         | 75.92<br>75.89 | 2.9×                | 4.93×                         |
| 32,768 × 32,768<br>1024 GPUs | <b>APF-32</b><br>UNETR-256 | 2.1567<br>8.1896  | 2,116<br>16,384 | 10<br>-        | 75.32<br>74.96 | 3.79×               | 6.44×                         |
| 65,536 × 65,536<br>2048 GPUs | <b>APF-32</b><br>UNETR-512 | 5.733<br>13.218   | 4,096<br>16,384 | 11<br>-        | 75.82<br>75.31 | 2.3×                | 3.91×                         |

# Results: Quality

- Since we can reduce the sequence length
- We could increase the patch size, get better results (for the same compute budget)

| Resolution      | Model               | Patch | GPUs | Sec/Image/GPU | Depth | Sequence Length | Dice Score   | Dice Improvement |
|-----------------|---------------------|-------|------|---------------|-------|-----------------|--------------|------------------|
| 8,192 × 8,192   | <b>APF (+UNTER)</b> | 2     | 256  | 2.3314        | 12    | 10,609          | <b>79.56</b> | 5.70%            |
|                 |                     | 4     | 256  | 2.1314        | 11    | 8,464           | 78.31        |                  |
|                 |                     | 8     | 128  | 1.7867        | 10    | 4,096           | 77.61        |                  |
|                 |                     | 16    | 64   | 1.1613        | 9     | 2,116           | 76.13        |                  |
|                 | UNETR               | 64    | 256  | 2.6618        | -     | 16,384          | 75.27        |                  |
|                 | TransUNet           | -     | 256  | 2.3678        | -     | -               | 70.89        |                  |
|                 | U-Net               | -     | 32   | 1.2858        | -     | -               | 63.21        |                  |
| 16,384 × 16,384 | <b>APF (+UNTER)</b> | 2     | 512  | 4.8792        | 13    | 16,384          | <b>80.62</b> | 6.23%            |
|                 |                     | 4     | 256  | 3.1231        | 12    | 8,464           | 79.31        |                  |
|                 |                     | 8     | 256  | 1.8574        | 11    | 4,096           | 78.84        |                  |
|                 |                     | 16    | 128  | 1.6421        | 10    | 2,116           | 77.43        |                  |
|                 | UNETR               | 128   | 512  | 5.1179        | -     | 16,384          | 75.89        |                  |
|                 | TransUNet           | -     | 512  | 6.1296        | -     | -               | 70.46        |                  |
|                 | U-Net               | -     | 256  | 2.7825        | -     | -               | 62.97        |                  |
| 32,768 × 32,768 | <b>APF (+UNTER)</b> | 4     | 1024 | 7.8916        | 13    | 16,384          | <b>78.98</b> | 5.36%            |
|                 |                     | 8     | 512  | 6.1792        | 12    | 8,464           | 78.31        |                  |
|                 |                     | 16    | 512  | 4.1685        | 11    | 4,096           | 77.61        |                  |
|                 |                     | 32    | 256  | 2.1567        | 10    | 2,116           | 76.13        |                  |
|                 | UNETR               | 256   | 1024 | 8.1896        | -     | 16,384          | 74.96        |                  |
|                 | TransUNet           | -     | 1024 | 10.001        | -     | -               | 69.88        |                  |
|                 | U-Net               | -     | 512  | 4.2714        | -     | -               | 61.38        |                  |
| 65,536 × 65,536 | <b>APF (+UNTER)</b> | 8     | 2048 | 12.697        | 13    | 16,384          | <b>77.77</b> | 3.27%            |
|                 |                     | 16    | 1024 | 8.793         | 12    | 8,464           | 76.11        |                  |
|                 |                     | 32    | 512  | 5.733         | 11    | 4,096           | 75.41        |                  |
|                 |                     | 64    | 256  | 3.961         | 10    | 2,116           | 75.13        |                  |
|                 | UNETR               | 512   | 2048 | 13.218        | -     | 16,384          | 75.31        |                  |
|                 | TransUNet           | -     | 2048 | 14.3516       | -     | -               | 67.67        |                  |
|                 | U-Net               | -     | 1024 | 5.961         | -     | -               | 59.69        |                  |

# AI for Cone-beam X-ray Computed Tomography

- X-ray CT widely used
- Current Generation X-ray CT rely on cone-beam scanners
- Higher quality; high-resolution real-time distributed reconstruction is intractable
- Use ViT to do geometry correction to make the real-time reconstruction tractable
  - $4K^3$  in 16 seconds and  $8K^3$  in a few minutes (on 1,024 GPUs)

Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer



Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer

Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer



Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer

Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer



Scans from Southampton University for Bio research



high-resolution Bumblebee generated on ABCI supercomputer

Scans from Southampton University for Bio research





Scans from Southampton University for Bio research

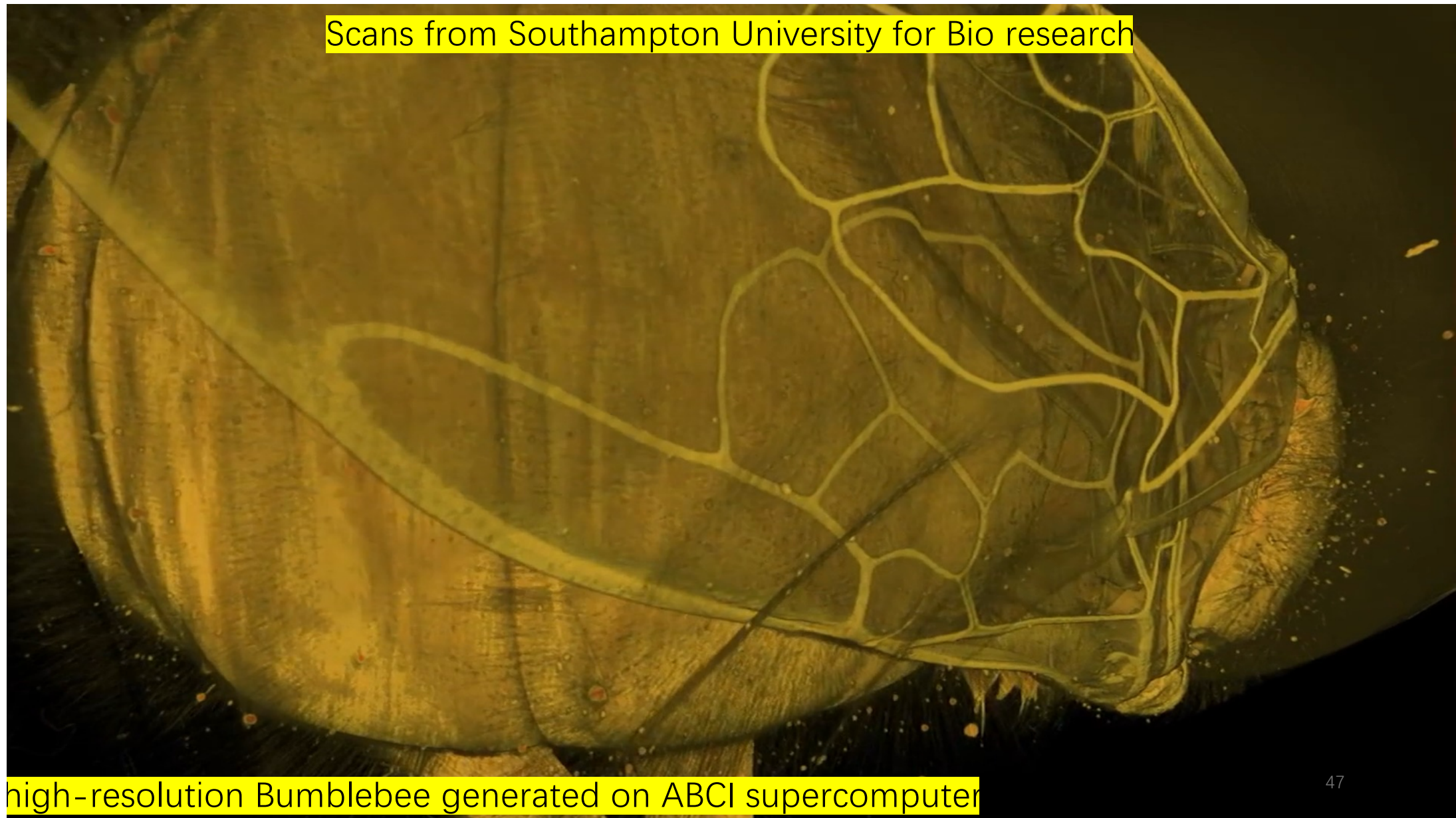


Scans from Southampton University for Bio research



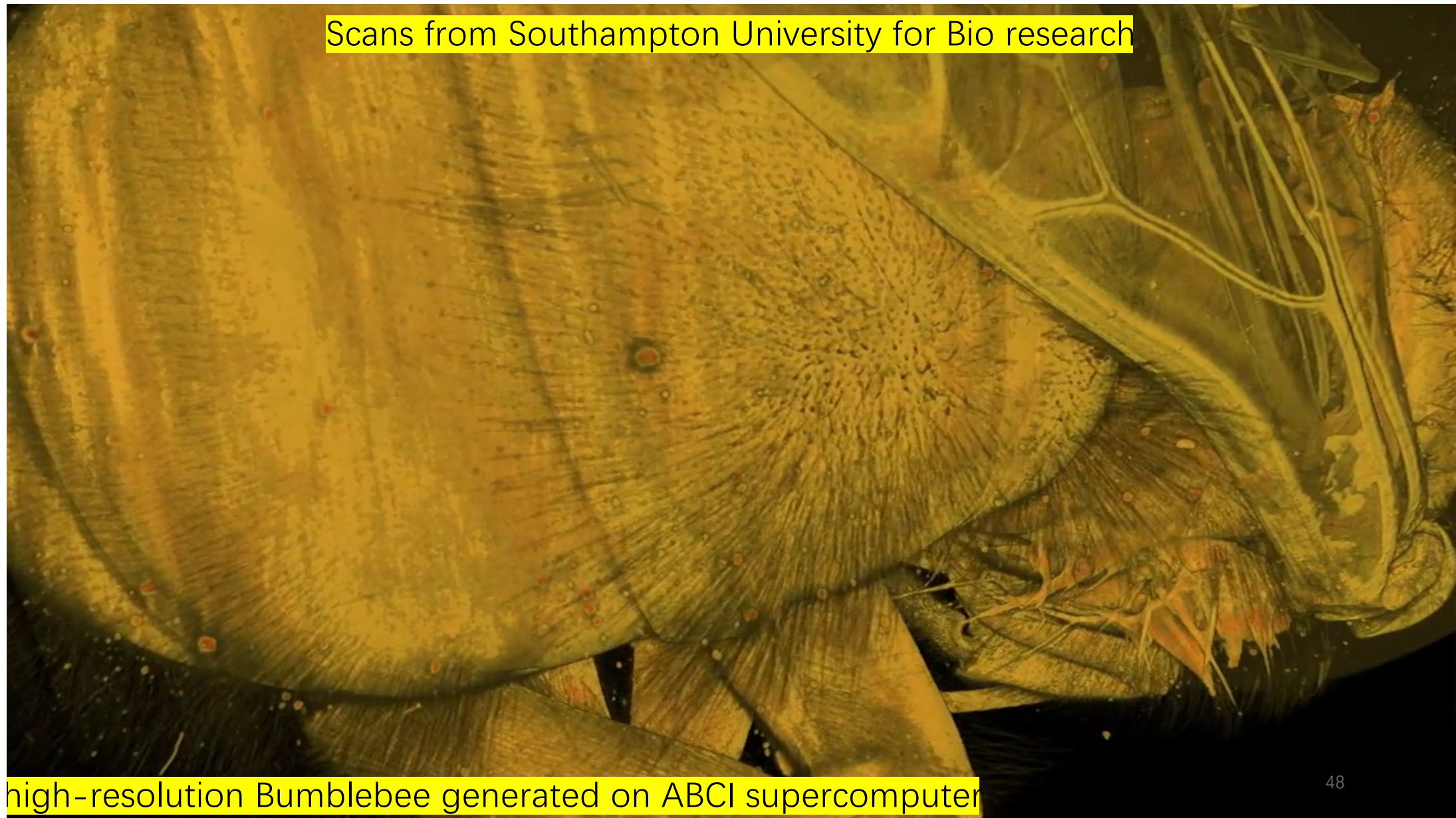


Scans from Southampton University for Bio research





Scans from Southampton University for Bio research



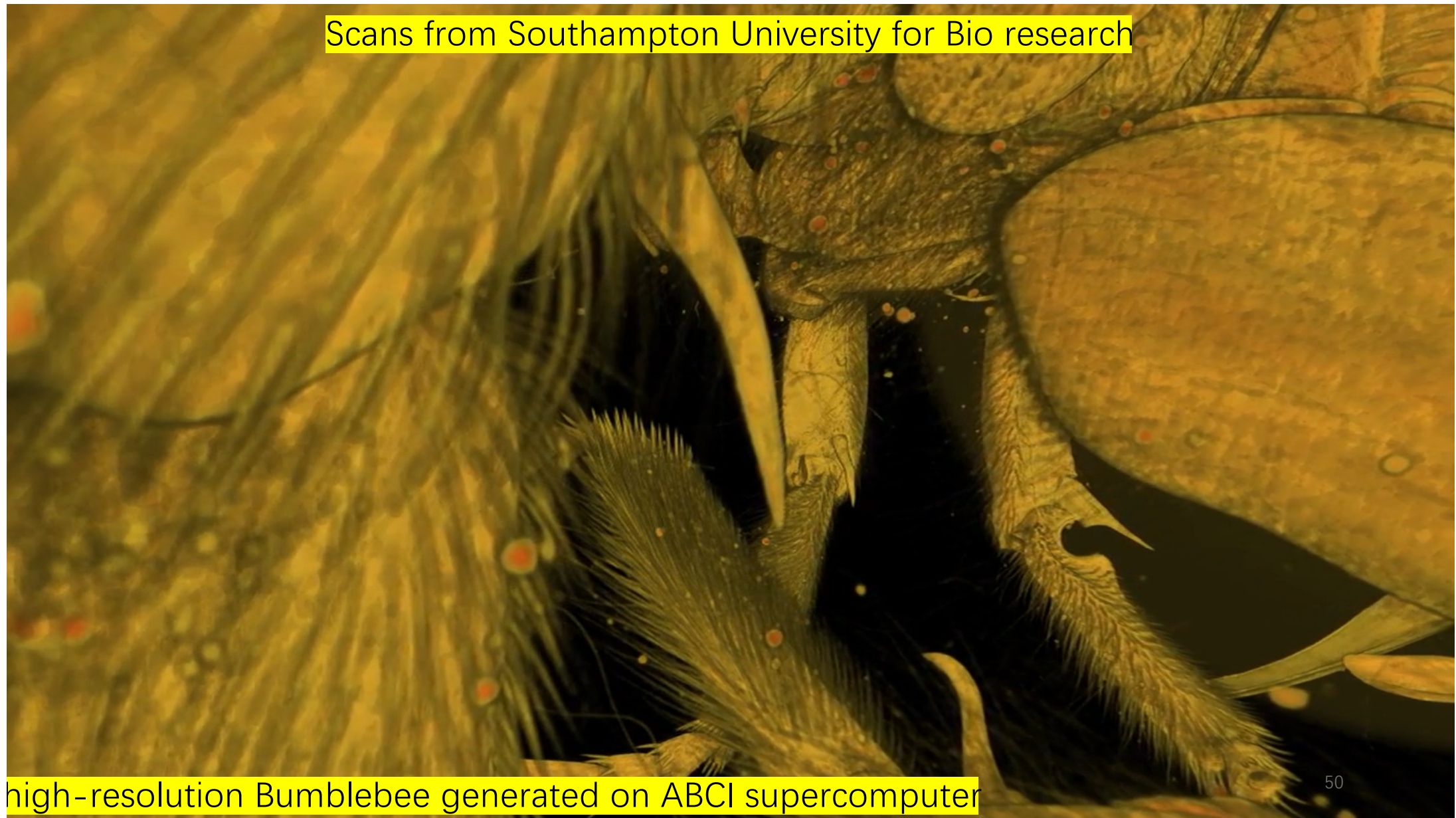


Scans from Southampton University for Bio research





Scans from Southampton University for Bio research



# ViT Challenges

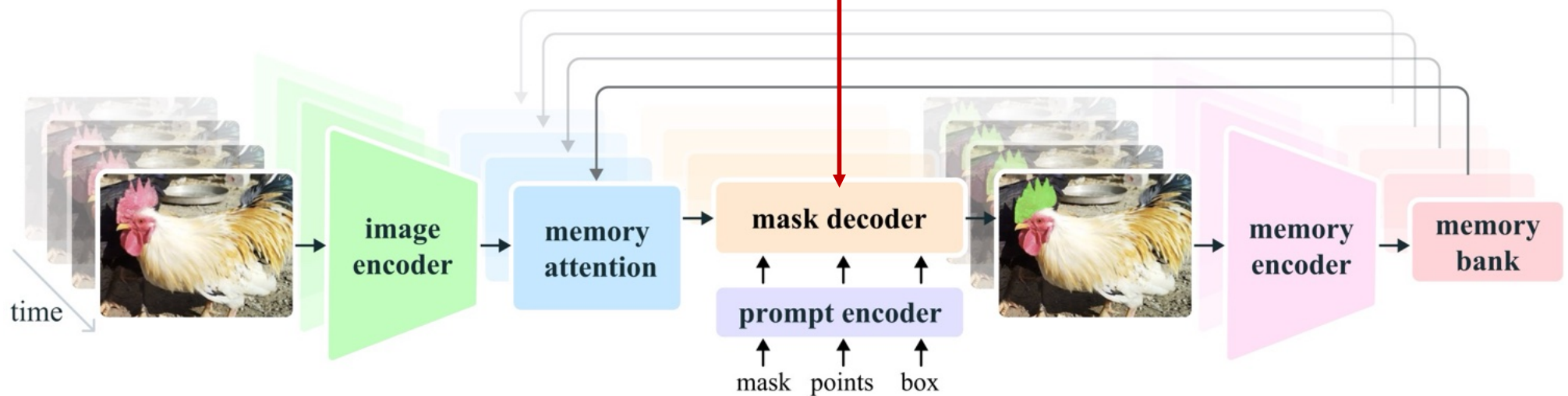
- ❶ Long sequence (when ingesting all image elements at high res)
- ❷ Shifting bottleneck (elements outside the Transformer encoder)
- ❸ Tokenization (managing temporal dimension)
- ❹ Different parallelism in training (vs. text Transformer)
- ❺ Multi-modality



# SoTA Models Overwhelmed by High-resolution

## ➤ In Segmentation

- We must map encoded features back to full-resolution pixel predictions
- **CNN mask decoder** (too much memory required)



\* <https://arxiv.org/pdf/2408.00714>

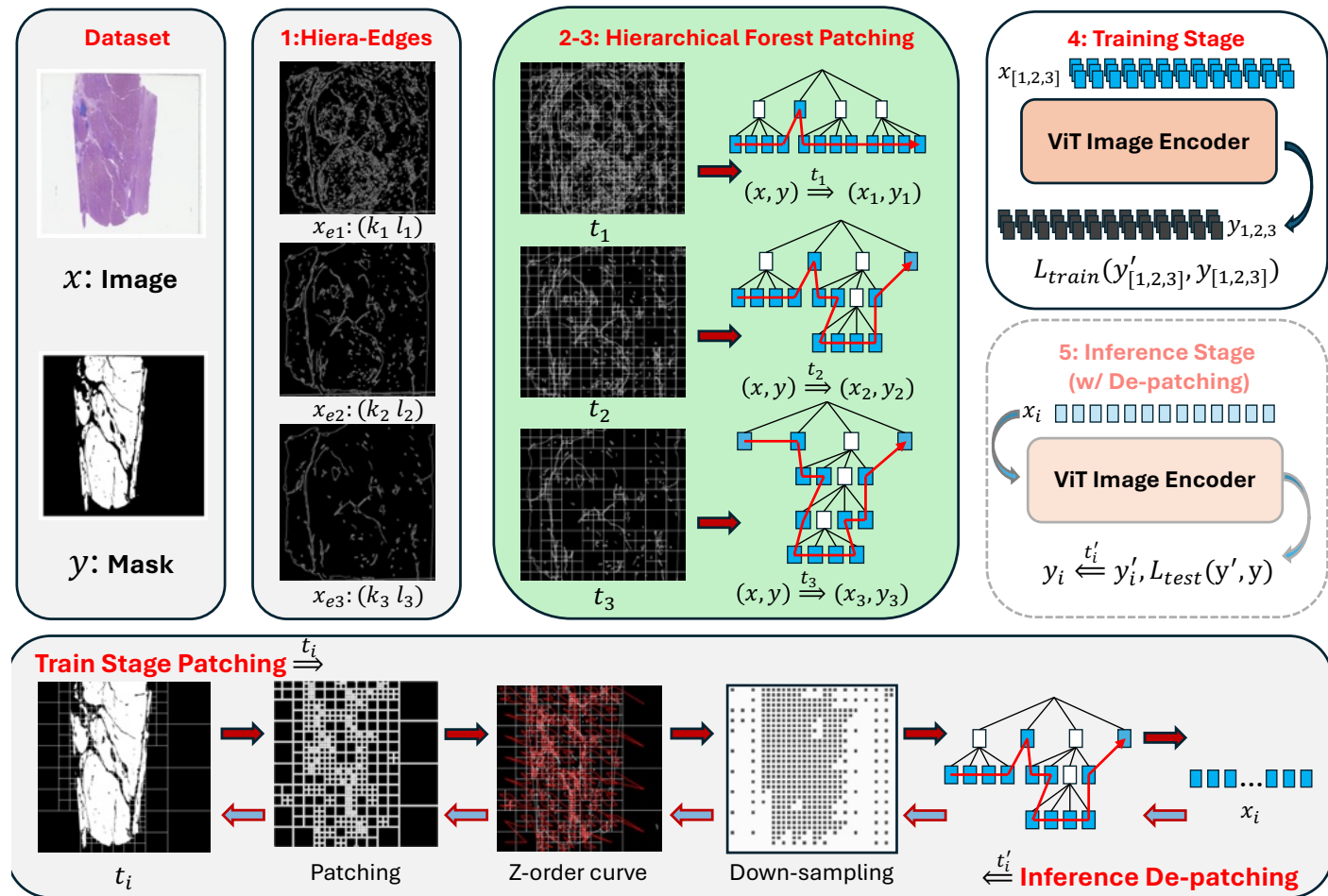
Meta's SAM 2 Model\*



# Symmetrical Hierarchical Forest with Pretrained ViT Encoder for High-Resolution Medical Segmentation

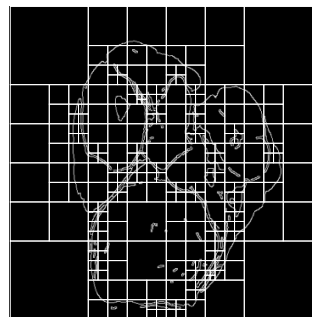
## Remove Decoder

Apply a reverse depatching scheme to the output embeddings of the transformer encoder, eliminating the need for convolution-based decoders

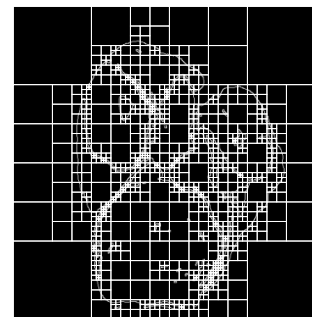


# Symmetrical Hierarchical Forest with Pretrained ViT Encoder for High-Resolution Medical Segmentation

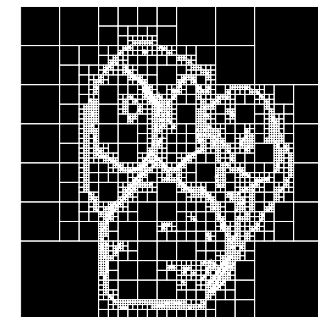
| Resolution      | Model          | Patch | GPUs | Sec/Image/GPU | Depth | Sequence Length | Dice Score   | Dice Improvement |
|-----------------|----------------|-------|------|---------------|-------|-----------------|--------------|------------------|
| 512 × 512       | <b>SAP+SAM</b> | 8     | 1    | 0.0438        | 6     | 512             | <b>75.31</b> | <b>+0.14</b>     |
|                 | SAM            | 16    | 1    | 0.1983        | -     | 1,024           | 61.39        |                  |
|                 | AP+UNTER       | 8     | 1    | 0.0581        | 6     | 576             | 75.17        |                  |
|                 | UNETR          | 16    | 1    | 0.1477        | -     | 1,024           | 74.88        |                  |
|                 | TransUNet      | -     | 1    | 0.1783        | -     | -               | 73.32        |                  |
|                 | UNet           | -     | 1    | 0.0438        | -     | -               | 70.32        |                  |
| 1,024 × 1,024   | <b>SAP+SAM</b> | 2     | 1    | 0.0991        | 9     | 1,024           | <b>78.67</b> | <b>+0.25</b>     |
|                 | SAM            | 8     | 8    | 0.3826        | -     | 16,384          | 66.56        |                  |
|                 | AP+UNTER       | 2     | 8    | 0.2314        | 9     | 1,024           | 78.42        |                  |
|                 | UNETR          | 8     | 32   | 1.0863        | -     | 16,384          | 75.72        |                  |
|                 | TransUNet      | -     | 8    | 1.3247        | -     | -               | 72.38        |                  |
|                 | UNet           | -     | 1    | 0.0981        | -     | -               | 68.92        |                  |
| 4,096 × 4,096   | <b>SAP+SAM</b> | 2     | 8    | 0.3766        | 11    | 4,096           | <b>78.67</b> | <b>+0.14</b>     |
|                 | SAM            | 32    | 64   | 1.6183        | -     | 16,384          | 71.05        |                  |
|                 | AP+UNTER       | 2     | 128  | 0.6938        | 11    | 4,096           | <b>79.63</b> |                  |
|                 | UNETR          | 32    | 128  | 1.8613        | -     | 16,384          | 75.77        |                  |
|                 | TransUNet      | -     | 128  | 2.1637        | -     | -               | 71.32        |                  |
|                 | UNet           | -     | 16   | 0.3712        | -     | -               | 64.11        |                  |
| 8,192 × 8,192   | <b>SAP+SAM</b> | 2     | 16   | 1.5327        | 12    | 8192            | <b>79.68</b> | <b>+0.12</b>     |
|                 | SAM            | 64    | 128  | 2.5168        | -     | 16,384          | 67.31        |                  |
|                 | AP+UNTER       | 2     | 256  | 2.3314        | 12    | 10,609          | 79.56        |                  |
|                 | UNETR          | 64    | 256  | 2.6618        | -     | 16,384          | 75.27        |                  |
|                 | TransUNet      | -     | 256  | 2.3678        | -     | -               | 70.89        |                  |
|                 | UNet           | -     | 32   | 1.2858        | -     | -               | 63.21        |                  |
| 16,384 × 16,384 | <b>SAP+SAM</b> | 2     | 32   | 3.2741        | 13    | 16,384          | <b>80.98</b> | <b>+0.36</b>     |
|                 | SAM            | 128   | 256  | 5.6714        | -     | 16,384          | 67.63        |                  |
|                 | AP+UNTER       | 2     | 512  | 4.8792        | 13    | 16,384          | 80.62        |                  |
|                 | UNETR          | 128   | 512  | 5.1179        | -     | 16,384          | 75.89        |                  |
|                 | TransUNet      | -     | 512  | 6.1296        | -     | -               | 70.46        |                  |
|                 | UNet           | -     | 256  | 2.7825        | -     | -               | 62.97        |                  |
| 32,768 × 32,768 | <b>SAP+SAM</b> | 4     | 64   | 3.4631        | 13    | 16,384          | <b>81.43</b> | <b>+2.45</b>     |
|                 | SAM            | 256   | 512  | 9.1213        | -     | 16,384          | 62.34        |                  |
|                 | AP+UNTER       | 4     | 1024 | 7.8916        | 13    | 16,384          | 78.98        |                  |
|                 | UNETR          | 256   | 1024 | 8.1896        | -     | 16,384          | 74.96        |                  |
|                 | TransUNet      | -     | 1024 | 10.001        | -     | -               | 69.88        |                  |
|                 | UNet           | -     | 512  | 4.2714        | -     | -               | 61.38        |                  |
| 65,536 × 65,536 | <b>SAP+SAM</b> | 8     | 256  | 3.6112        | 13    | 16,384          | <b>82.96</b> | <b>+5.19</b>     |
|                 | SAM            | 1024  | 1024 | 12.983        | -     | 16,384          | 61.68        |                  |
|                 | AP+UNTER       | 8     | 2048 | 12.697        | 13    | 16,384          | 77.77        |                  |
|                 | UNETR          | 1024  | 2048 | 13.218        | -     | 16,384          | 75.31        |                  |
|                 | TransUNet      | -     | 2048 | 14.352        | -     | -               | 67.67        |                  |
|                 | UNet           | -     | 1024 | 5.961         | -     | -               | 59.69        |                  |



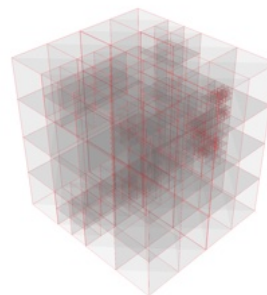
(b) Sequence Length=256



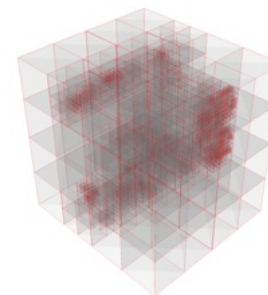
(c) Sequence Length=1024



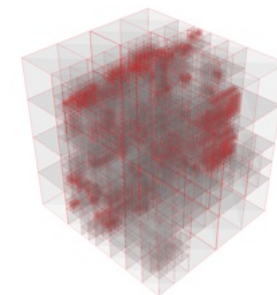
(d) Sequence Length=4096



(f) Sequence Length=260



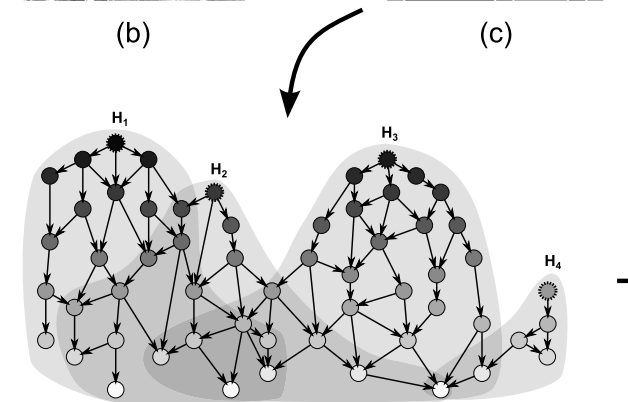
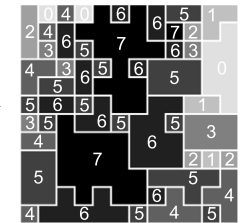
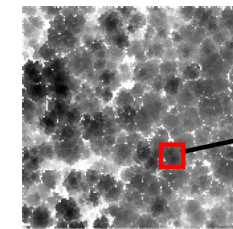
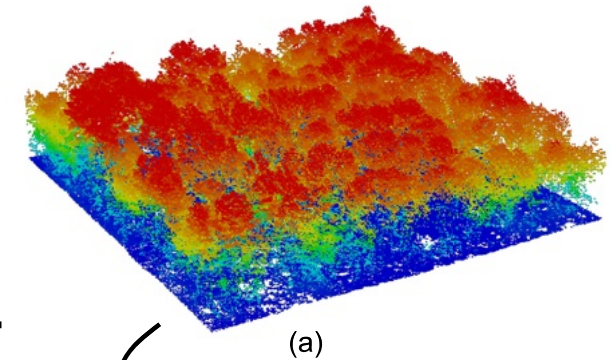
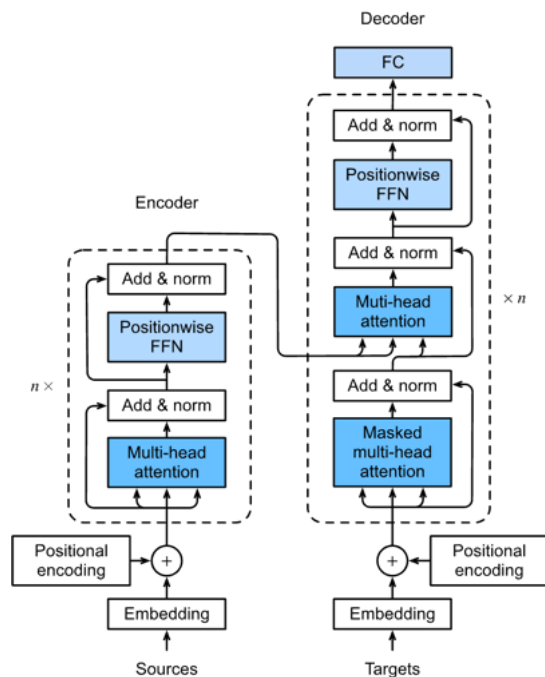
(g) Sequence Length=1030



(h) Sequence Length=4096

# Can ViT Learn Where to Look?

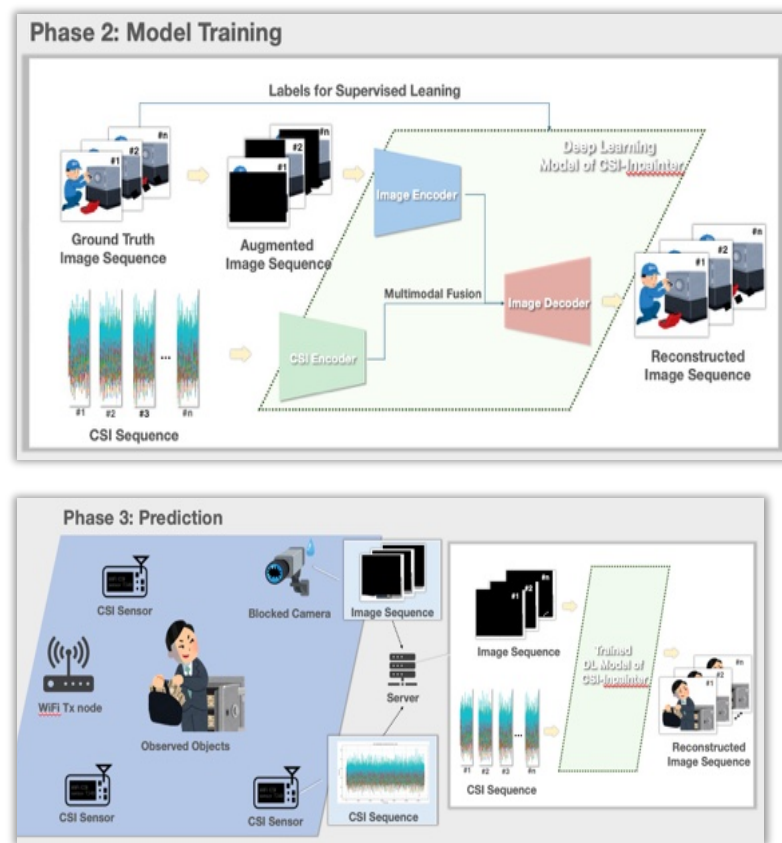
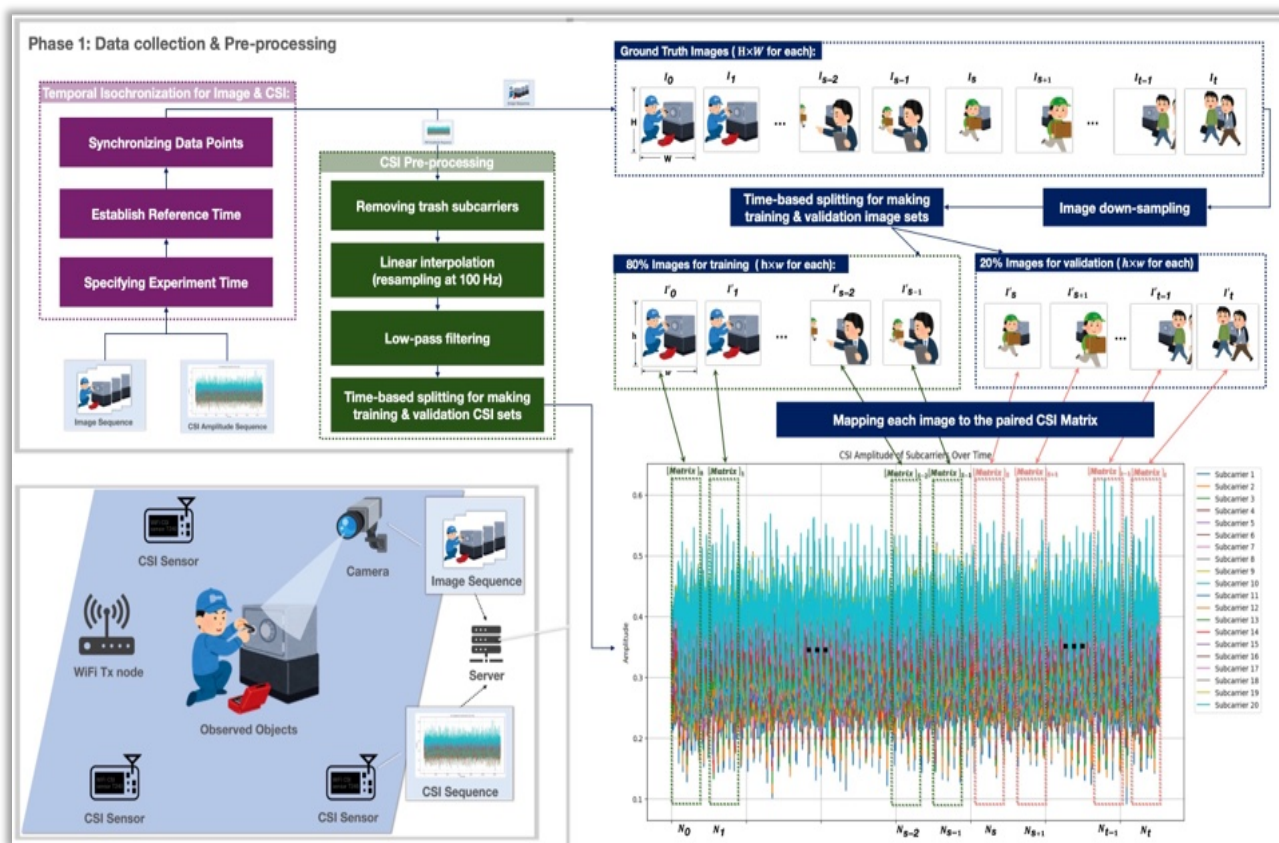
- HUMANS tell the model where to look
- Can the model learn where to look?
  - by being fed the spatial hierarchy





# Visual Scene Recovery from Wi-Fi CSI

## ➤ CSI-Inpainter: CSI-guided obstacle removal



# Visual Scene Recovery from Wi-Fi CSI

RF-Based Multimodal Obstacle Removal

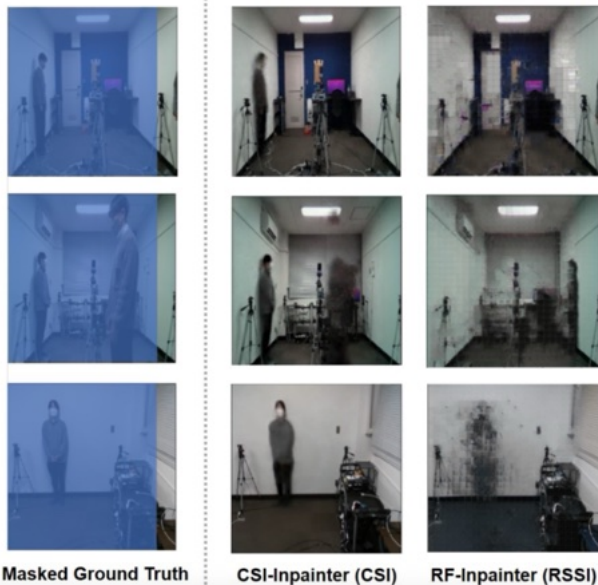
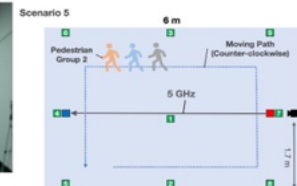
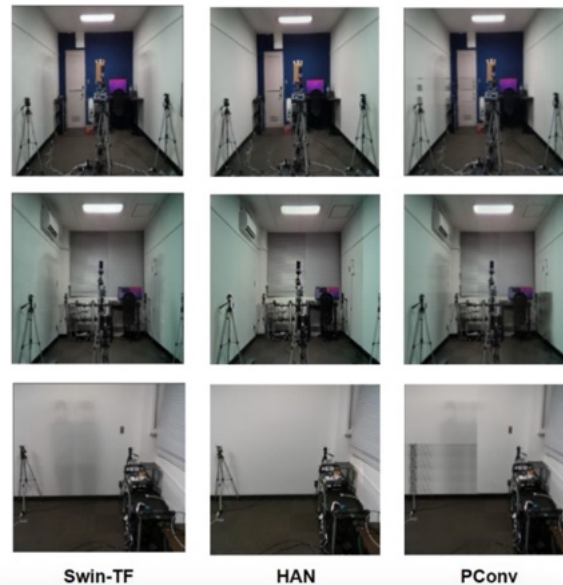


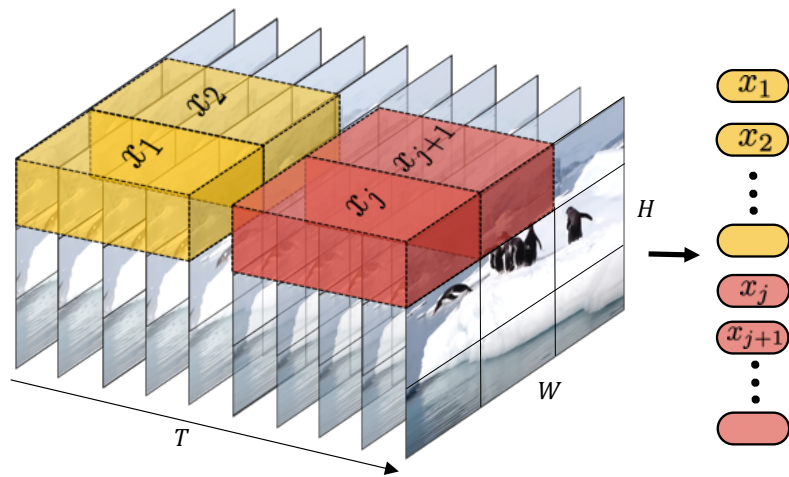
Image-only Obstacle Removal



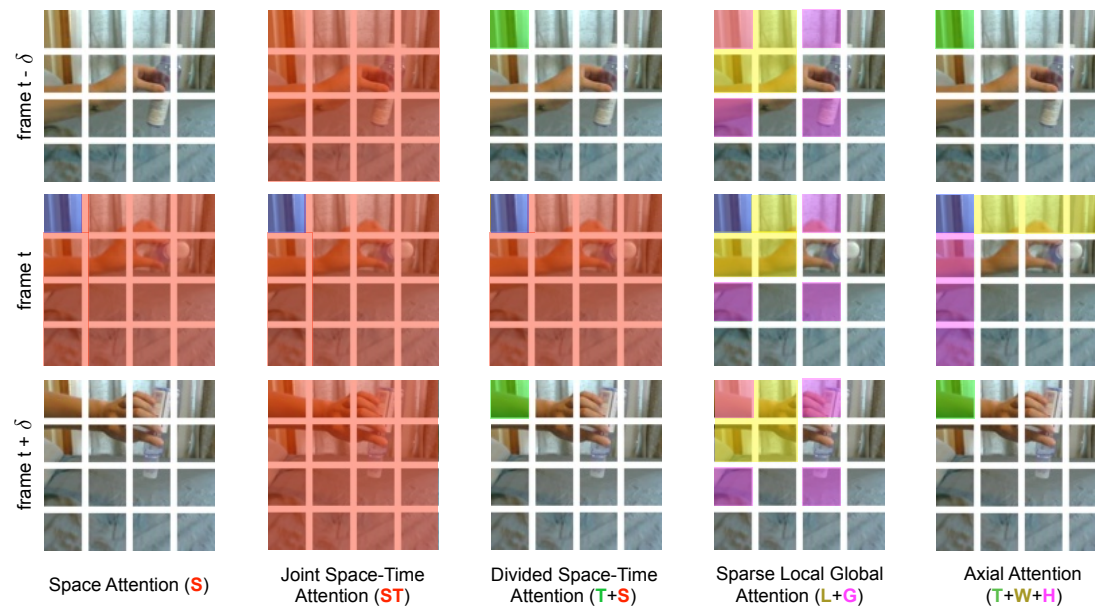
# ViT Challenges

- ① Long sequence (when ingesting all image elements at high res)
- ② Shifting bottleneck (elements outside the Transformer encoder)
- ③ Tokenization (managing temporal dimension)
- ④ Different parallelism in training (vs. text Transformer)
- ⑤ Multi-modality

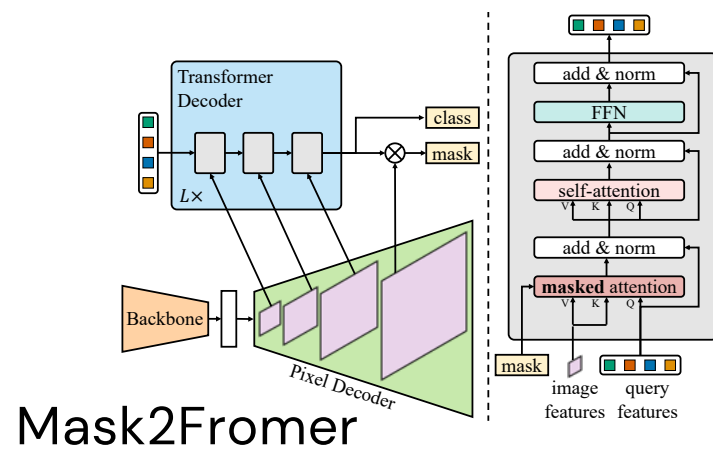
# Tokenizing Spatio-temporal Data is Tricky



ViViT



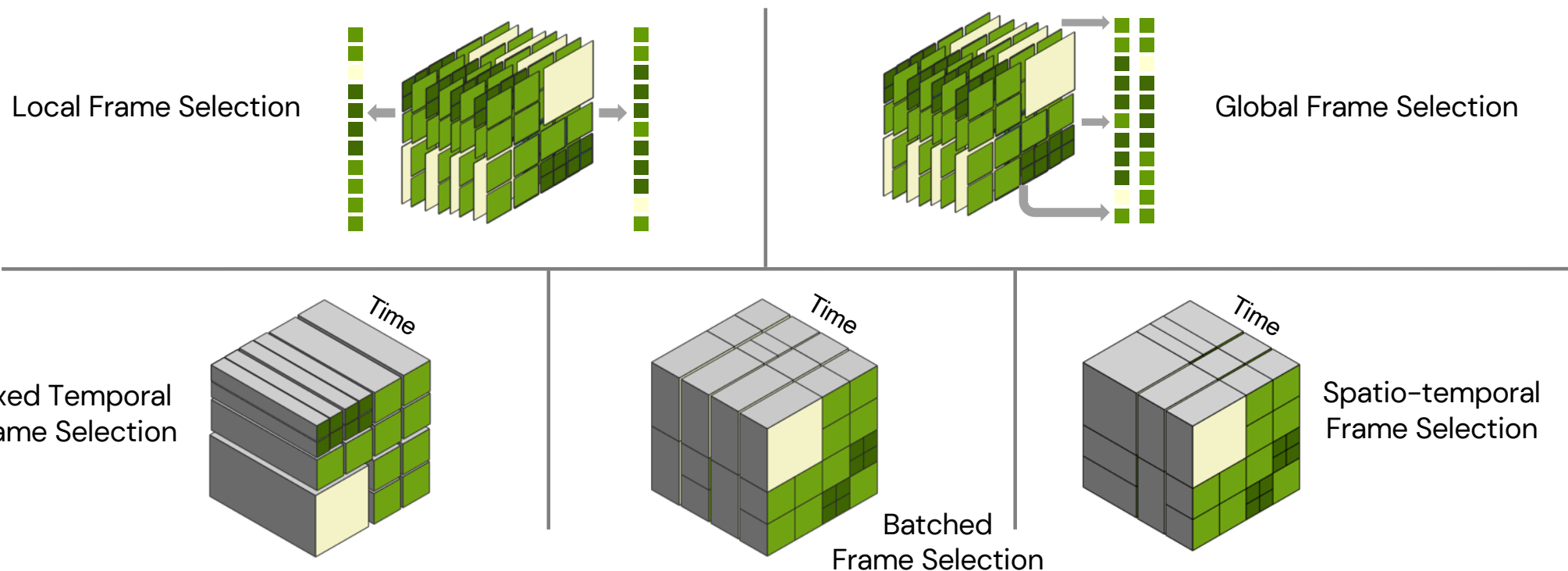
TimesFormer





# How to Tokenize Spatio-temporal Data

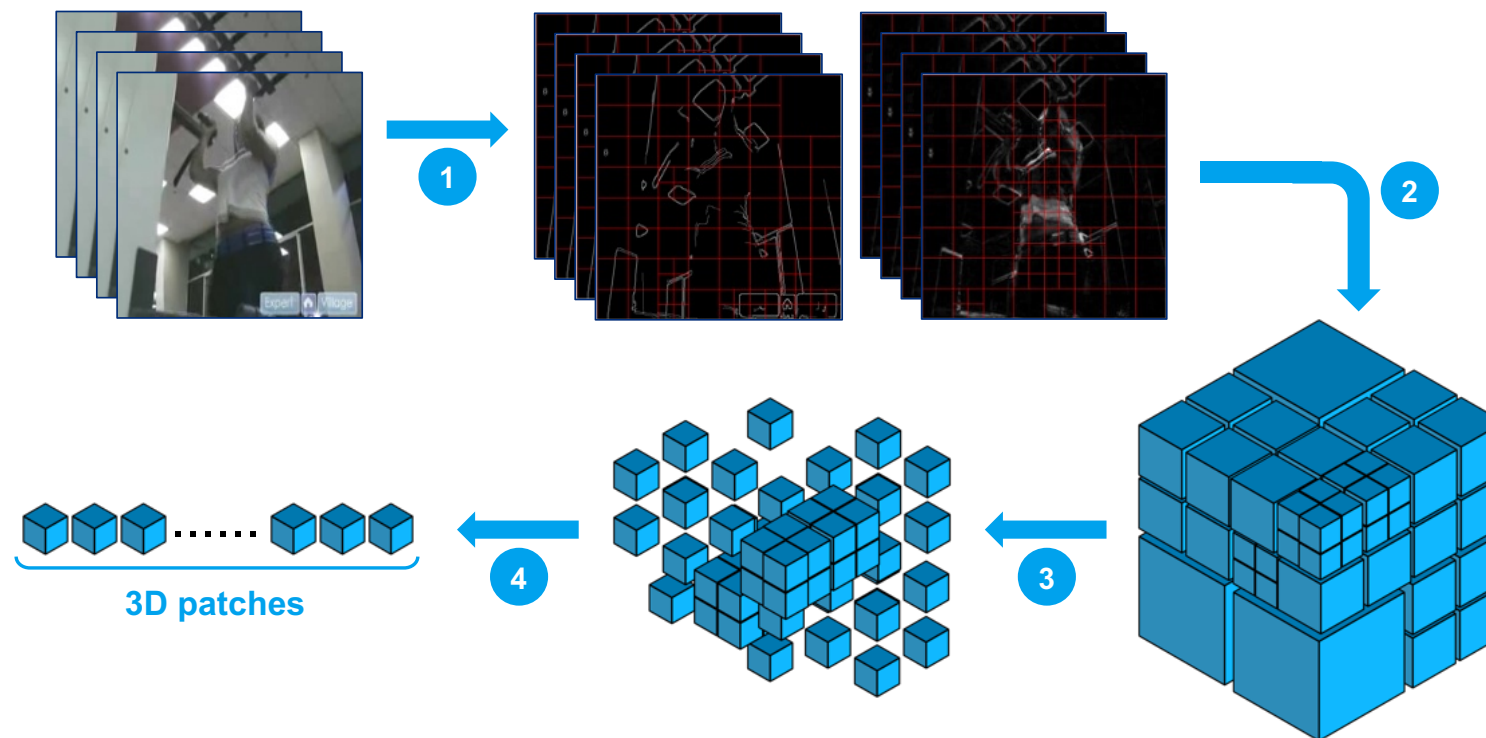
➤ Different tokenization schemes aligned with adaptive pathcing



We observe how Local and Global Frame Selection operate on a frame-by-frame basis to form quadrees, while the other three schemes work across three dimensions. Among these three, Batch and Temporal-Spatial Frame Selection divide the time dimension, with the former using even divisions and the latter using uneven divisions.

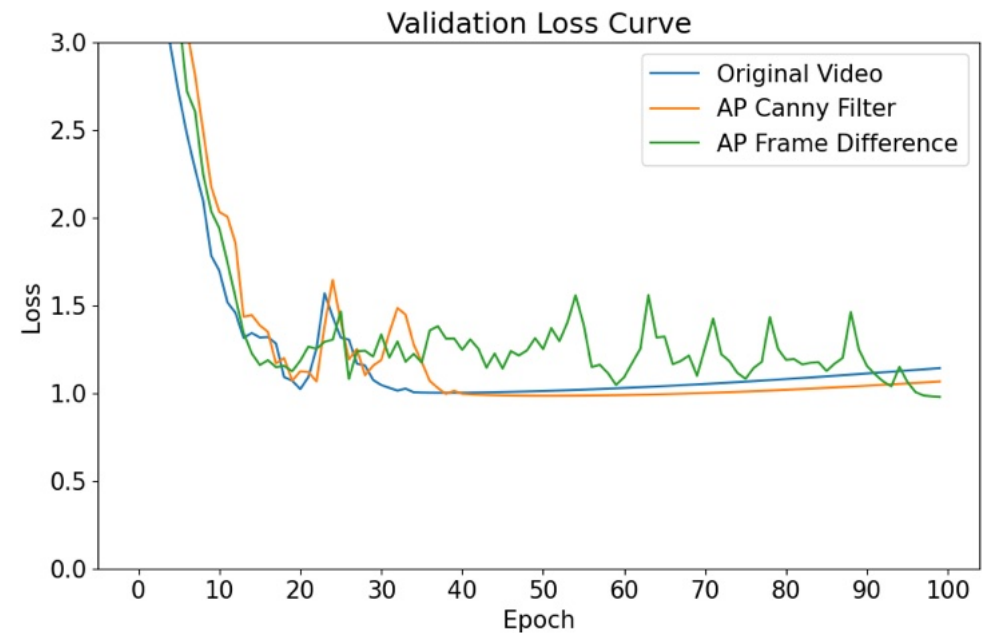
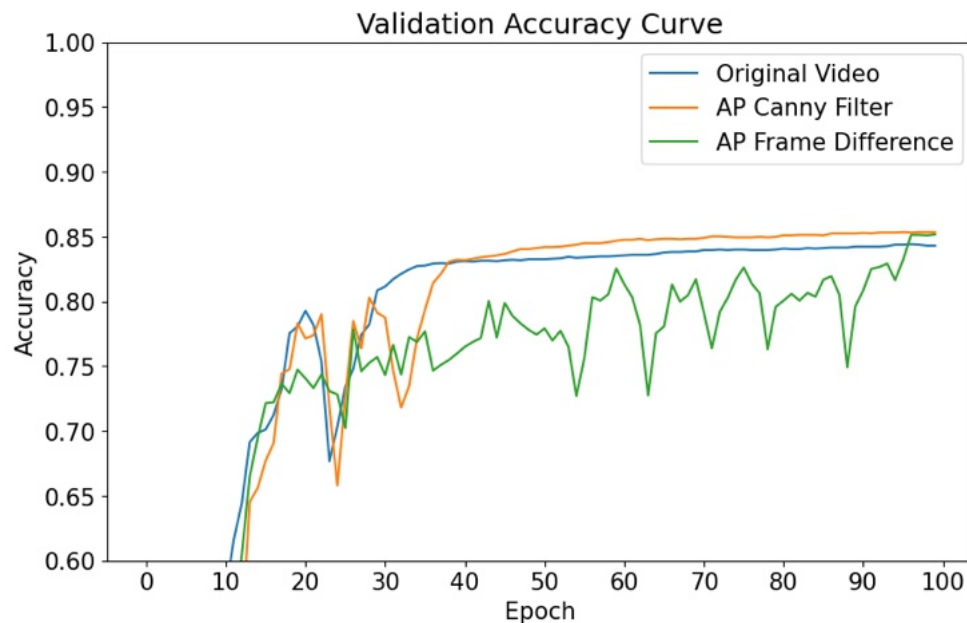
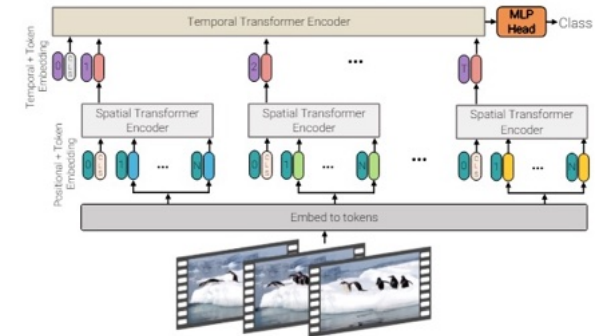
# Adaptive Patching for Videos

- For videos we adaptively patch temporally
  - merging spatial and temporal adaptiveness?
    - customize the adaptive scheme based on the nature of the input dataset? AP as a means for compression, followed by a scheme to arrange patches to recreate the input video?



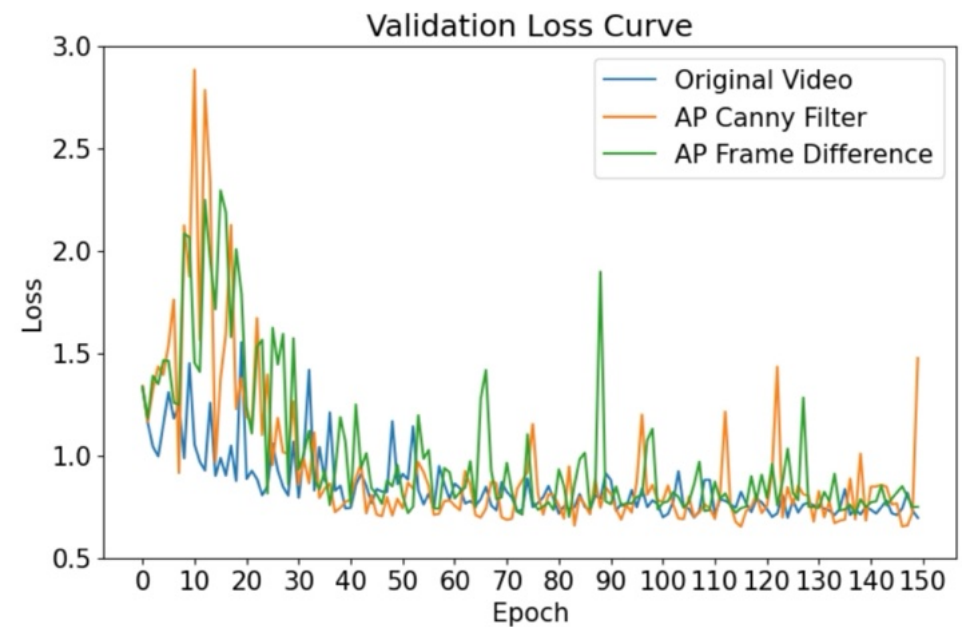
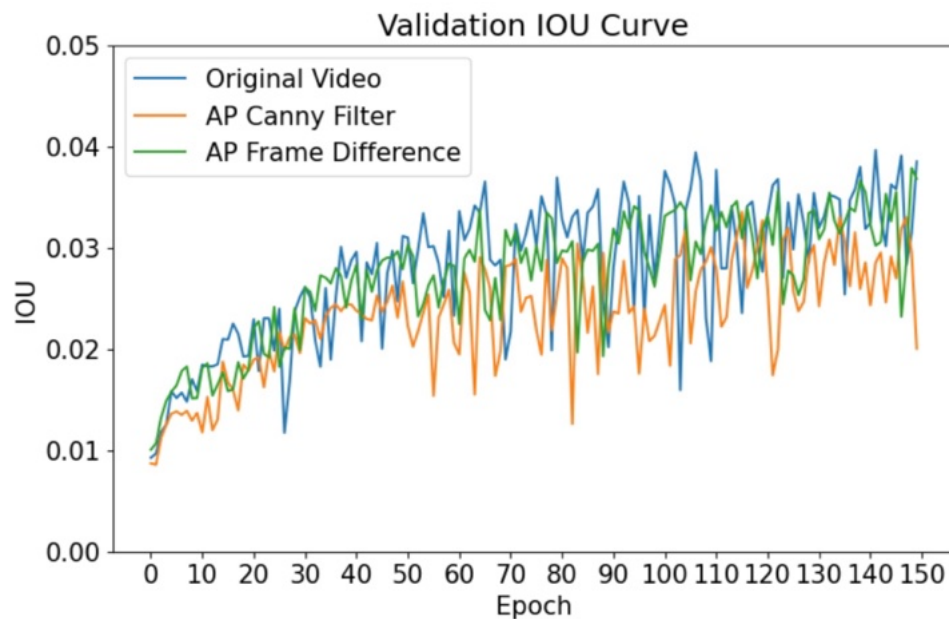
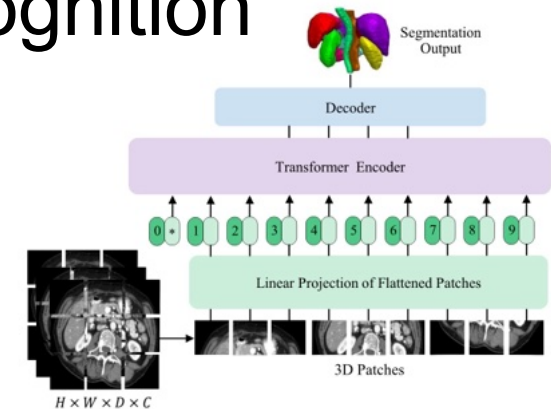
# Application in Video Action Recognition

- We use the **Video Vision Transformer (ViViT)** model for this task, containing attention in the spatial and temporal dimensions
- AP able to achieve comparable metrics with **up to 4x memory reduction** while maintaining the same number of patches



# Application in Video Action Recognition

- We use the **UNet Transformers (UNETR)** model for this task, combining a transformer encoder with a convolutional decoder
- AP able to achieve comparable metrics with **up to 8x memory reduction** while maintaining the same number of patches



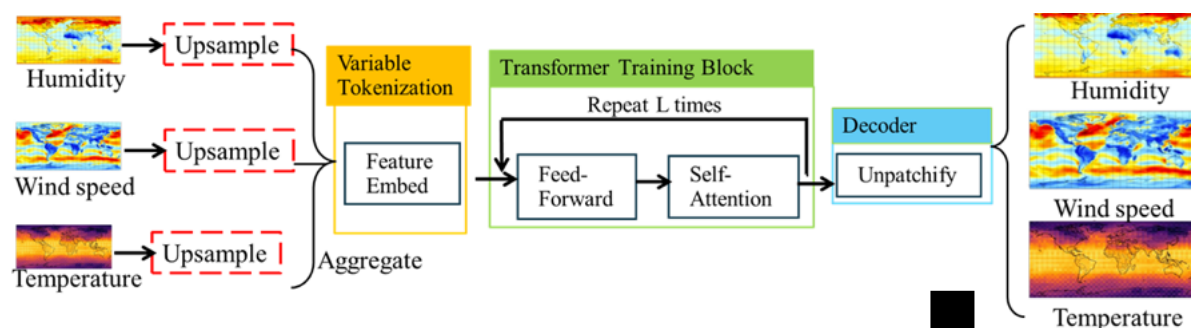


# ViT Challenges

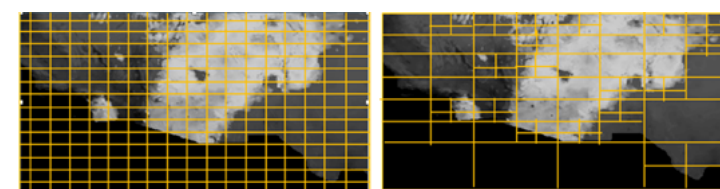
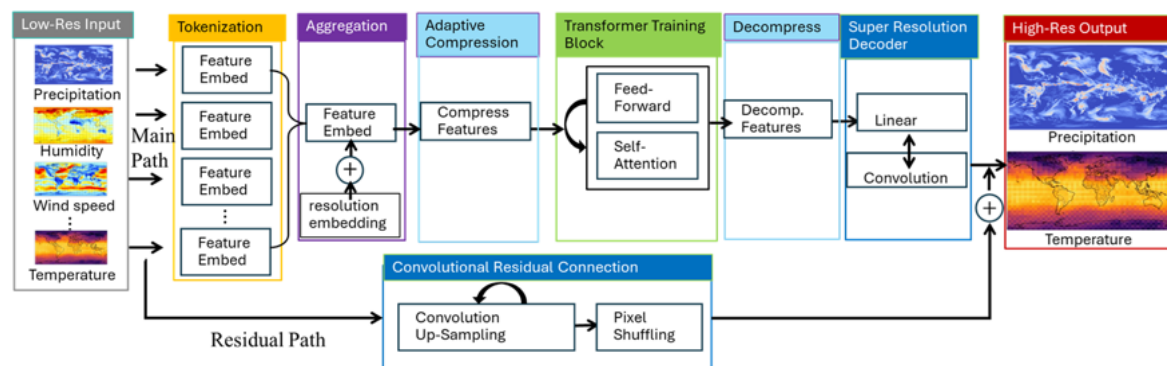
- ❶ Long sequence (when ingesting all image elements at high res)
- ❷ Shifting bottleneck (elements outside the Transformer encoder)
- ❸ Tokenization (managing temporal dimension)
- ❹ Different parallelism in training (vs. text Transformer)
- ❺ Multi-modality



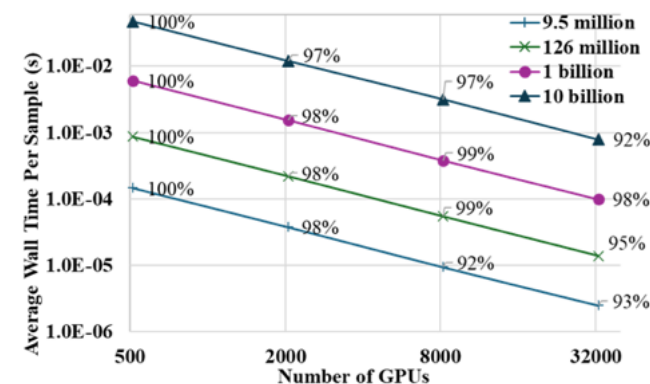
# ORBIT-2: Scaling Exascale Vision Foundation Models for Weather and Climate Downscaling



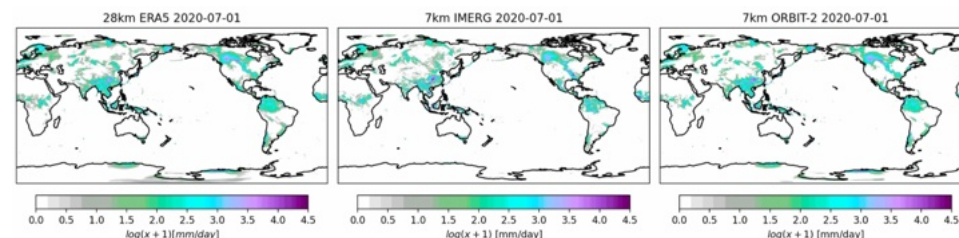
- Operating directly on compressed inputs
- Condition prediction for reduced uncertainty



(a) No adaptive compression (b) With adaptive compression



(b) Strong Scaling Efficiencies Across Models and GPUs

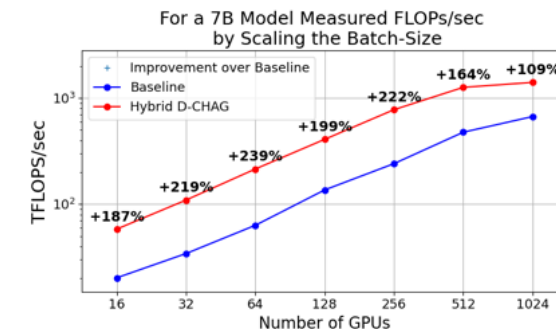
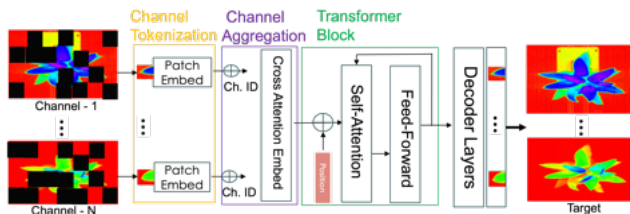
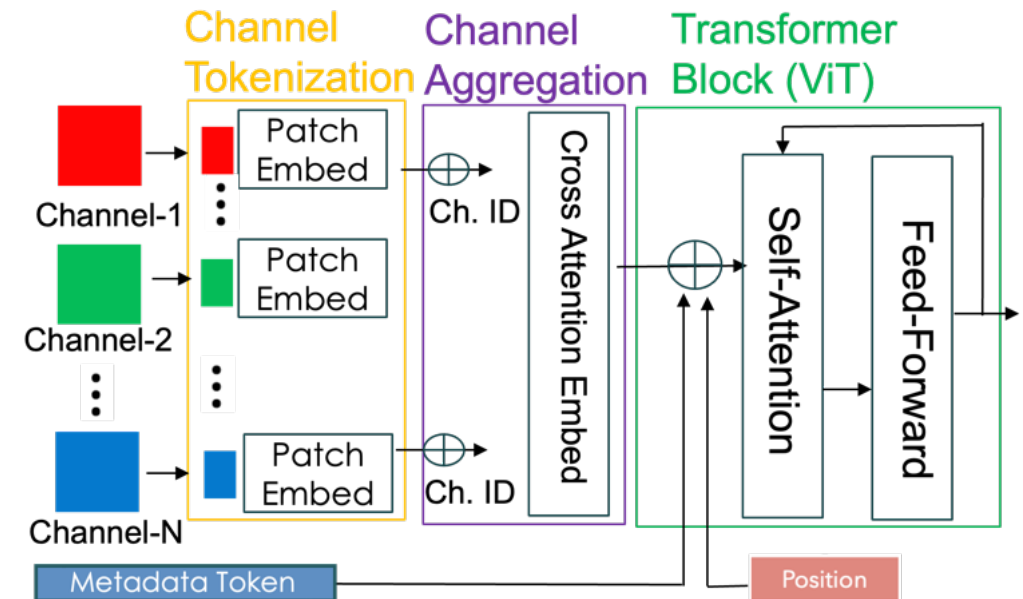


<https://arxiv.org/abs/2505.04802>

# Distributed Cross-Channel Hierarchical Aggregation for Foundation Models

## Distributed Tokenization

For high number of channels: distribute tokenization and implement a hierarchical strategy for channel aggregation.

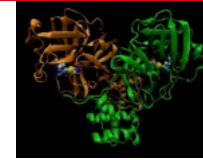
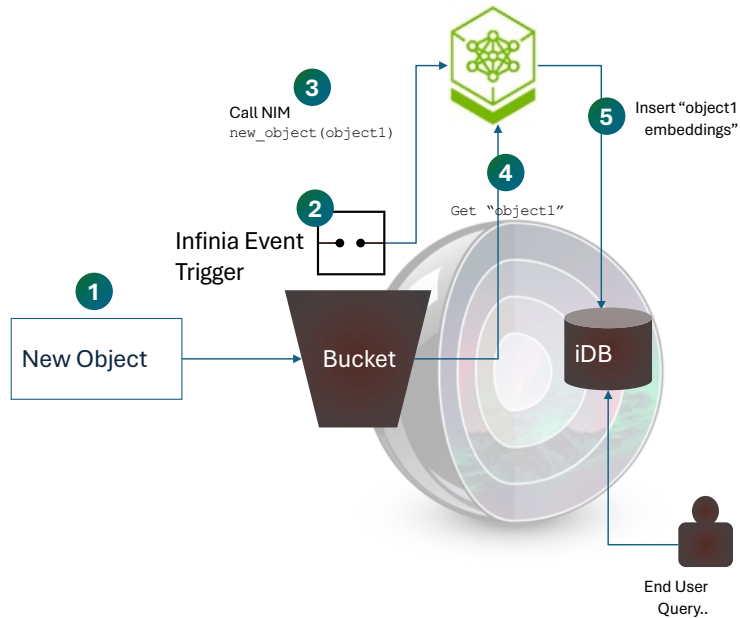


# ViT Challenges

- ① Long sequence (when ingesting all image elements at high res)
- ② Shifting bottleneck (elements outside the Transformer encoder)
- ③ Tokenization (managing temporal dimension)
- ④ Different parallelism in training (vs. text Transformer)
- ⑤ Multi-modality

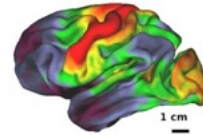


# Multi-agent Approach for Multi-modality



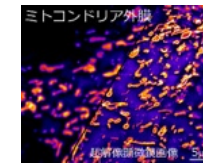
Proteins

Information Retrieval APIs (or Code)



Neural Activities

Information Retrieval APIs (or Code)



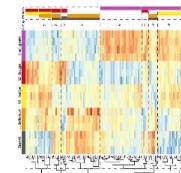
Images

Information Retrieval APIs (or Code)



Omics lipidome

Information Retrieval APIs (or Code)



Genome/Transcriptome

Information Retrieval APIs (or Code)



Other Phenotypes

Information Retrieval APIs (or Code)

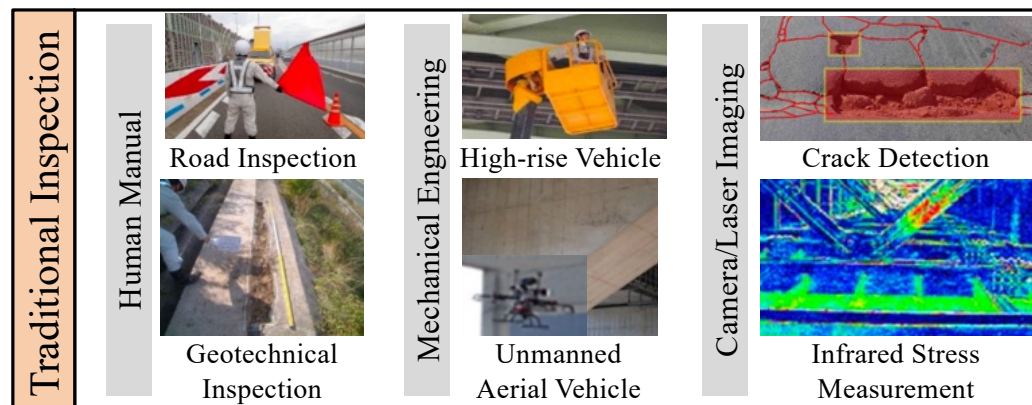
# Real-world Problem Covering Almost all the Challenges Presented so Far

# Paradigm Shift in Infrastructure Inspection Technology

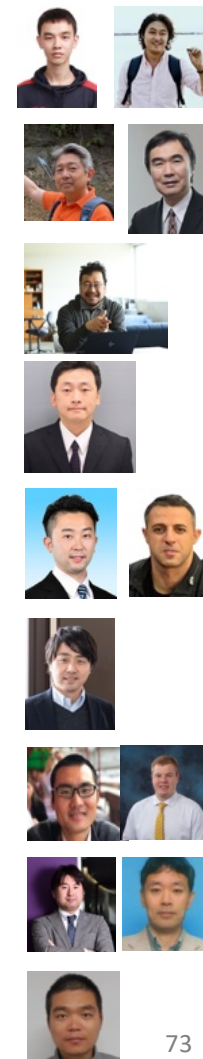
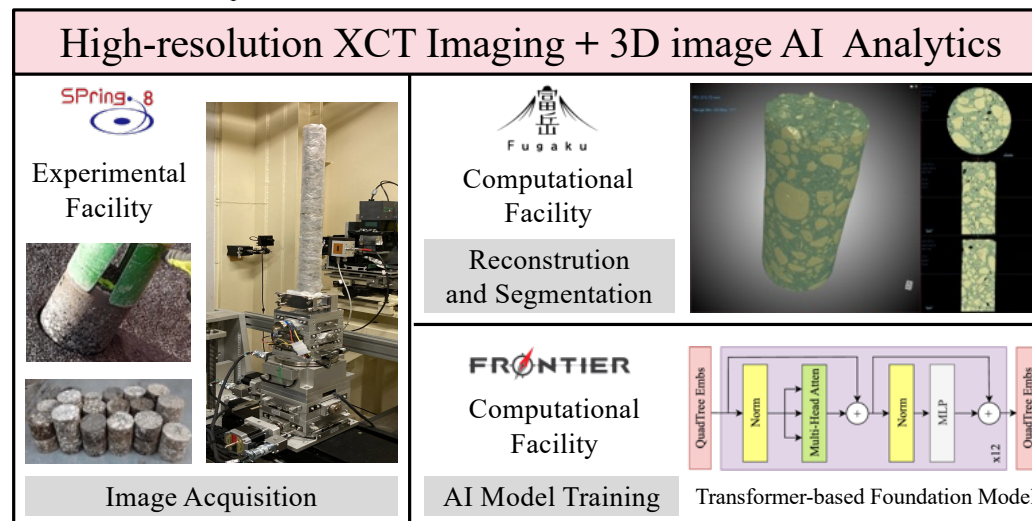


**High performance  
Imaging + AI +  
analytics**

- Replace traditional engineering
- Full fusion of imaging and AI

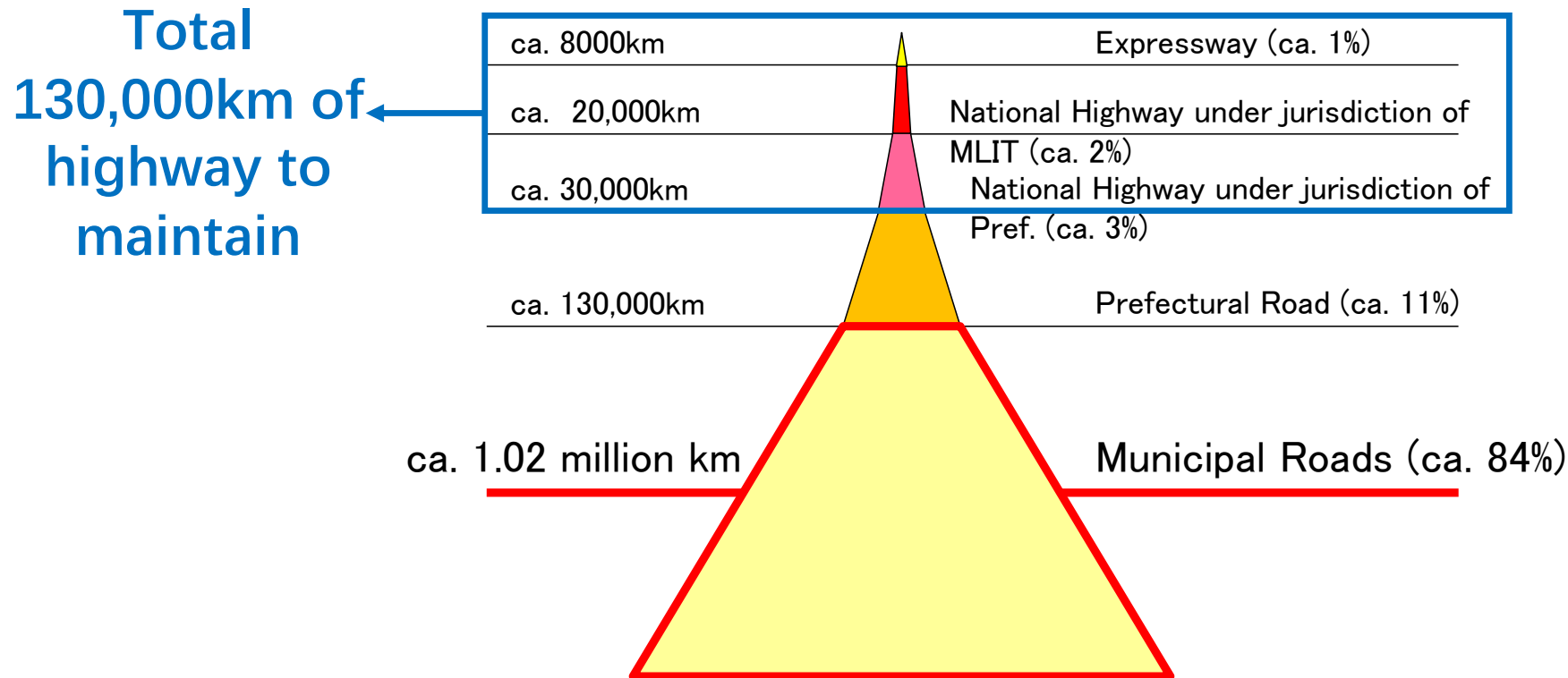


Infrastructure Inspection Paradigm Shift



# Total Road Length in Japan is ca. 1.21 Million KMs

## 【Road types and their percentages in Japan】





# Highway Network West Japan (Osaka, Kyoto, Kobe)



1号環状線 四ツ橋付近

1964年6月28日、土佐堀～湊町間が開通

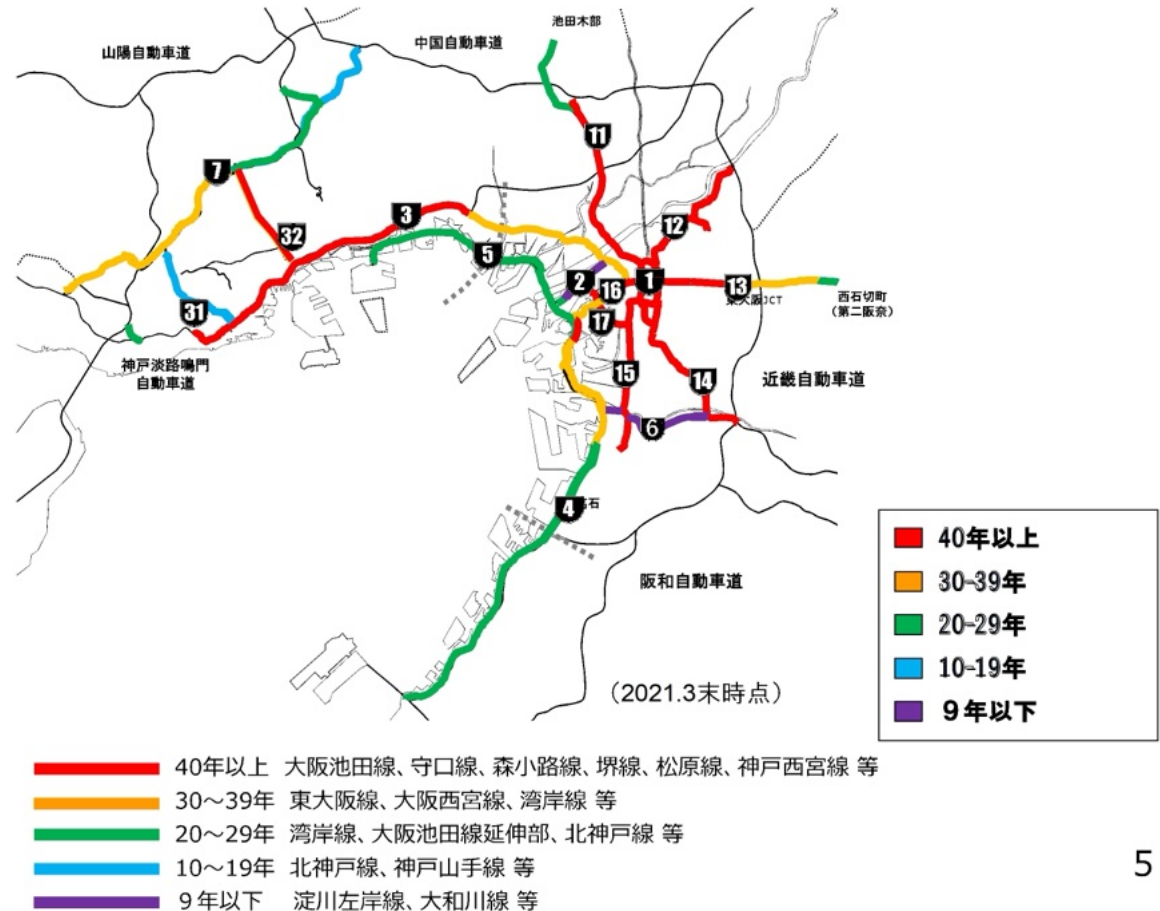


大和川第三トンネル

(6号大和川線：供用中)



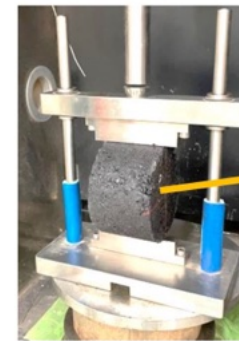
淀川左岸線 (2期) 建設中



\* Source: Sakai @Hanshin Highway

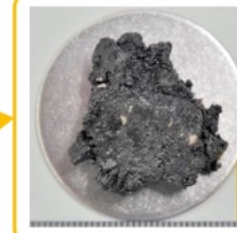
# How to Inspect Roads for Maintenance?

- Mechanical inspection
  - Time: 10s of years
  - Cost: \$ Billions

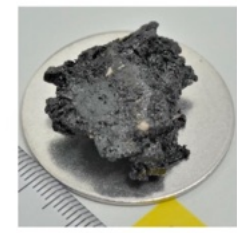


By EN12697-24 ANNEX-E

Broken parts



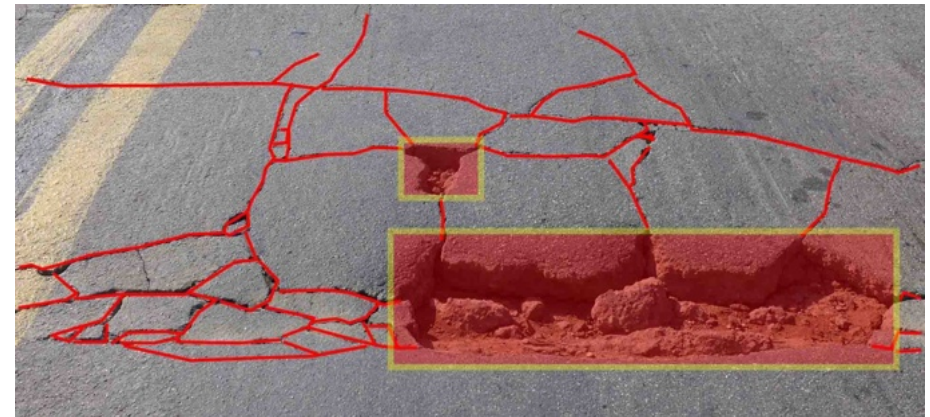
Visible light  
(above)



Visible light  
(right diagonal)

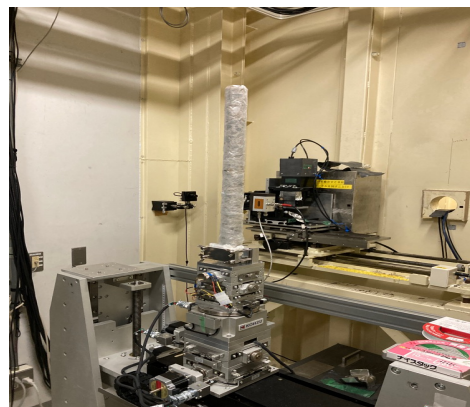
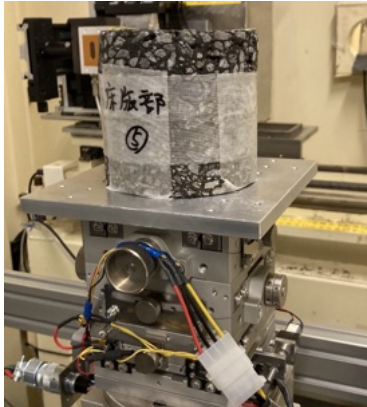
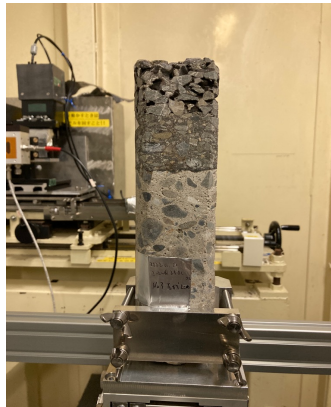
Fatigue test: Accelerated Crack Simulation

- Camera/laser Imaging technology
  - Good for fast screening of visible surface cracks, depressions etc
  - Not a reliable technology for understanding sub-surface conditions





# Mechanical Inspection



“Actual-scale test track” by Taisei-Rotech

# Vehicles Extracting 100s Samples per Day

- Core samples extraction machines mounted on vehicles



+





# Collecting Samples

- Extract cylindrical samples from core of asphalt layers



\* *Example: if one sample every 10KM, then  $130,000 / 10 = 13,000$  samples*

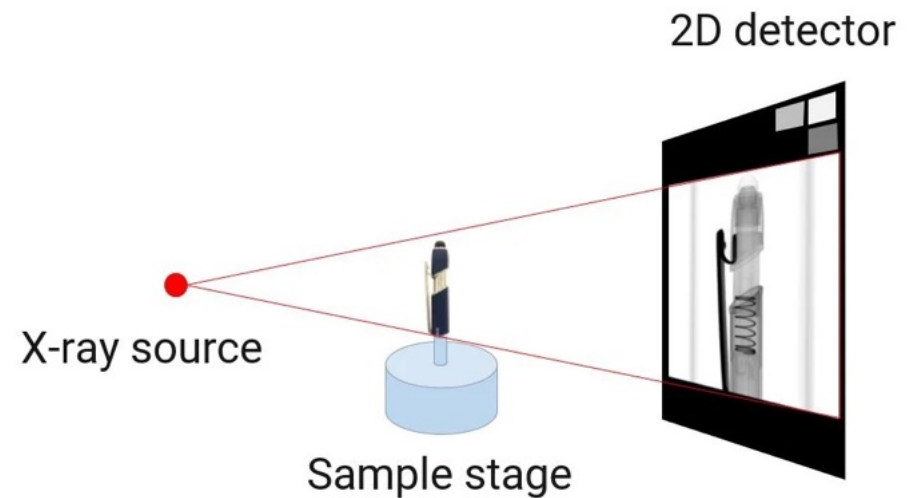
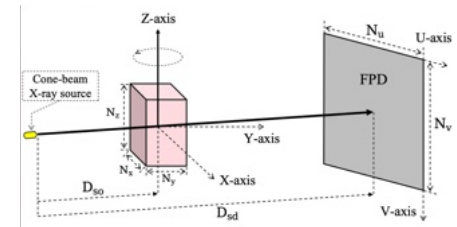


# Scan Samples with X-ray

- Scan (2D projections) at RIKEN Spring-8 Synchrotron

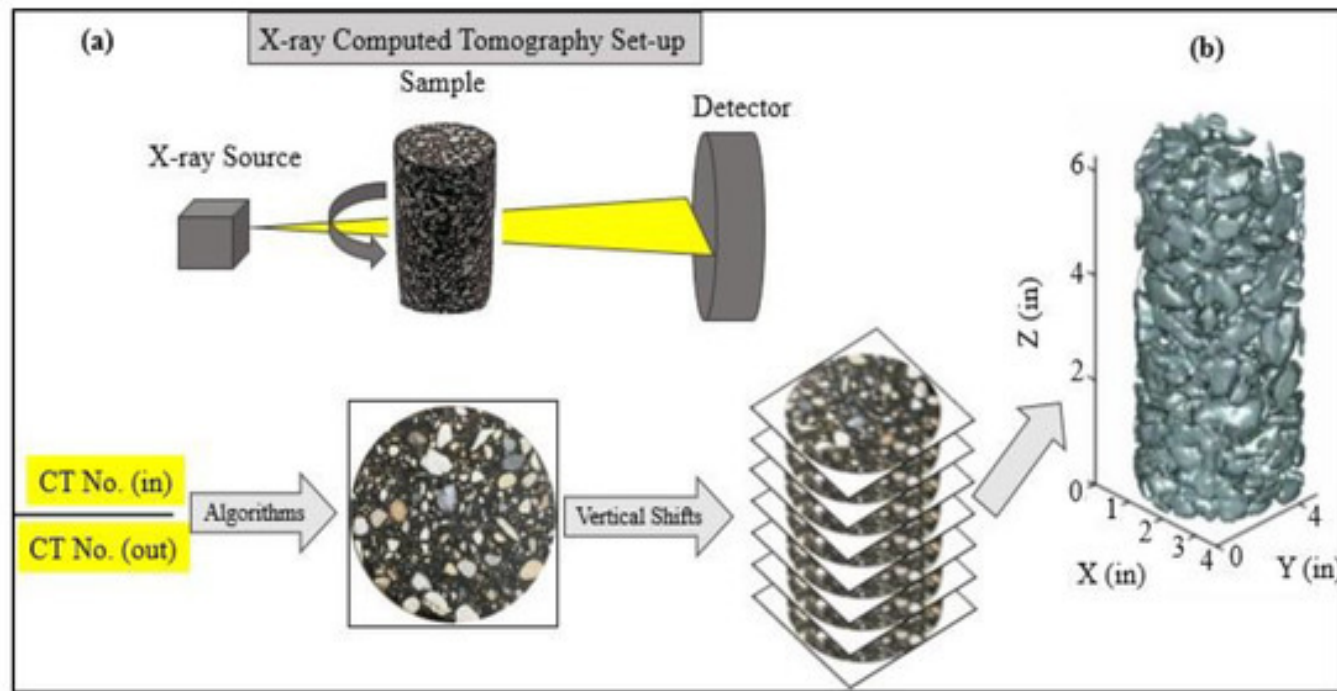


The synchrotron radiation facility RIKEN SPring-8

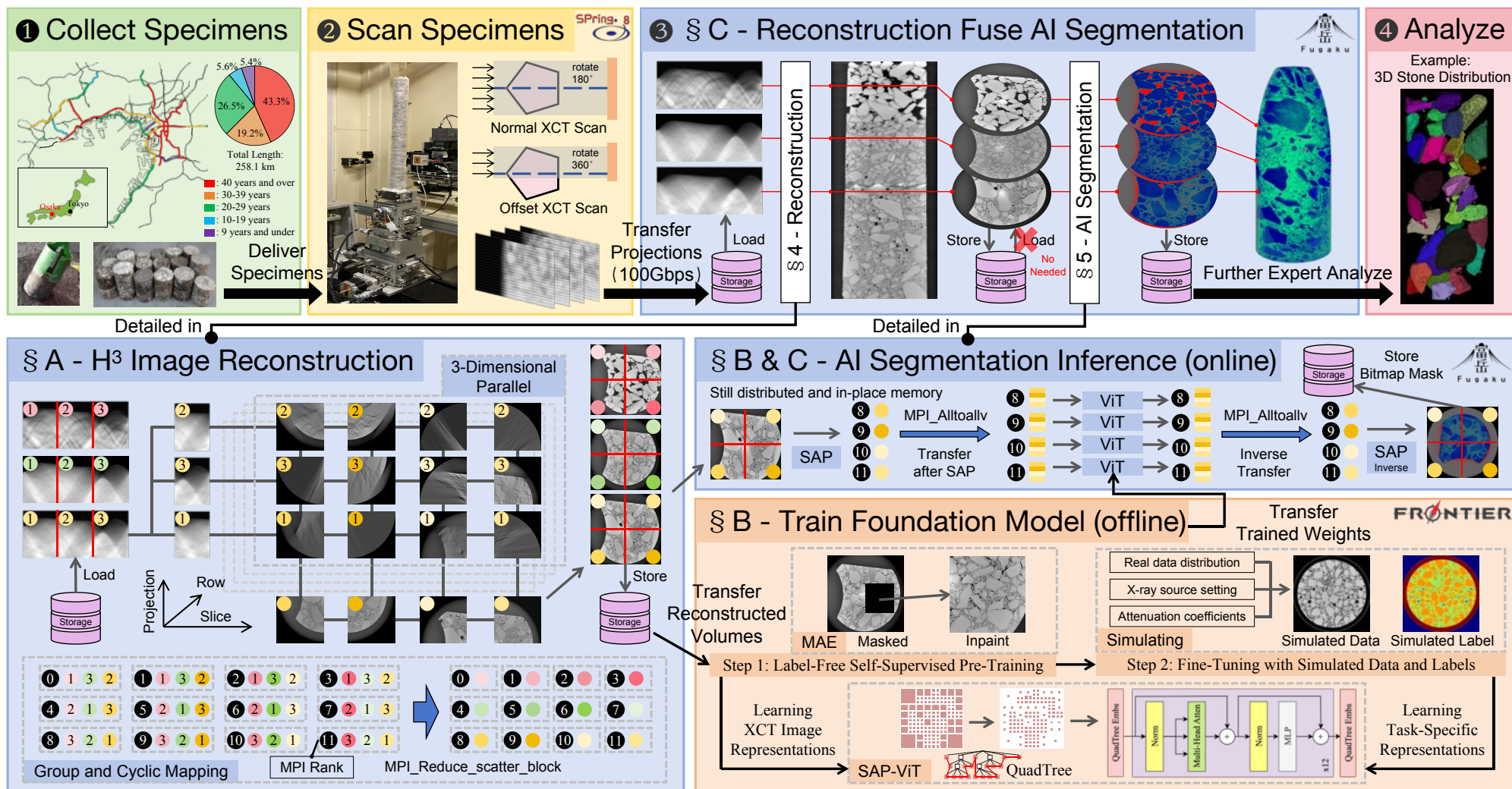


# High Resolution 3D Image Reconstruction

- High-performance high-resolution X-ray CT image reconstruction



# Paradigm Shift in Infrastructure Inspection Technology



# H<sup>3</sup>: High-throughput, High-perf., High-resolution CT

## H<sup>3</sup>

End-to-end pipeline to reconstruct  
10s of 16K resolution 3D images in  
one go (full-system scale)

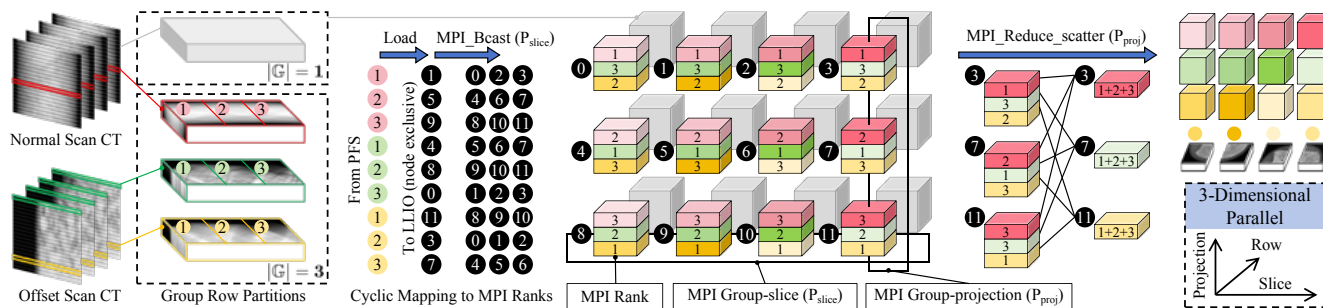
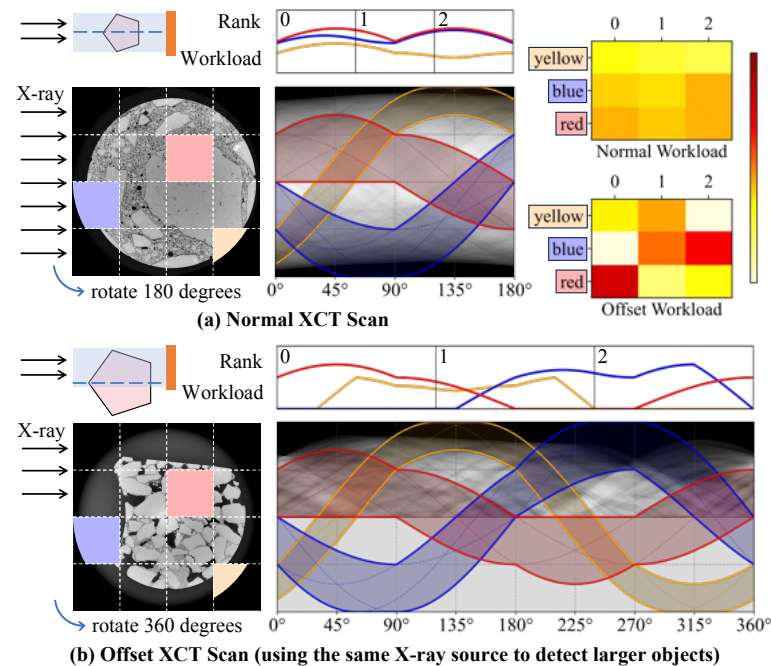
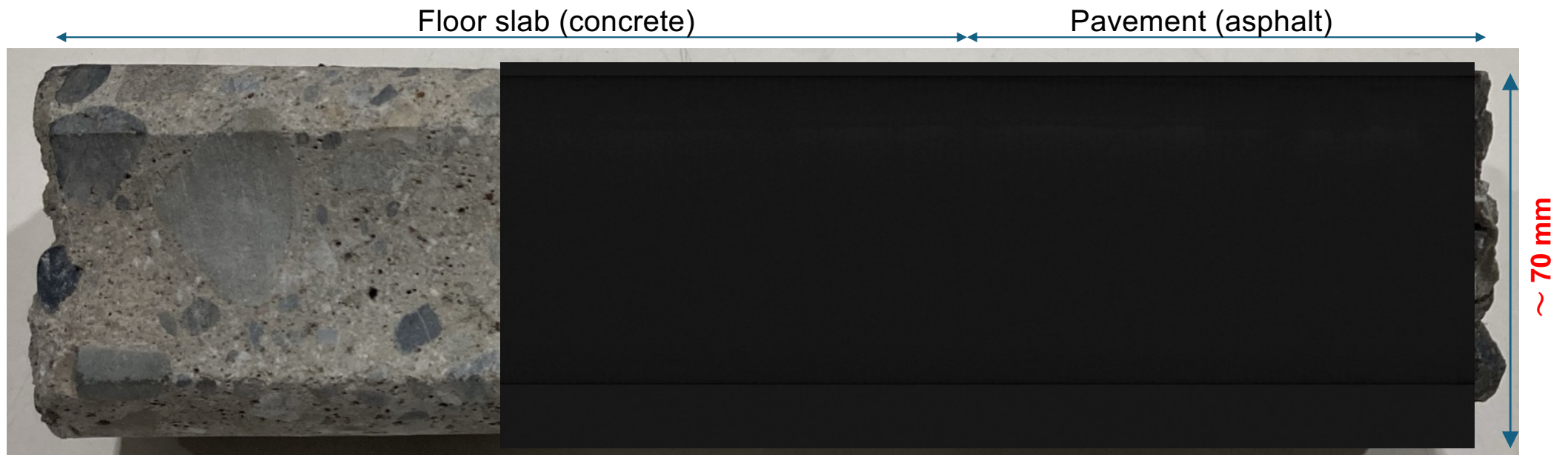


Fig. 5: Proposed H<sup>3</sup> imaging algorithm in ROVAI: Group row partitions and cyclic mapping to MPI ranks.



# End-to-end Image Reconstruction

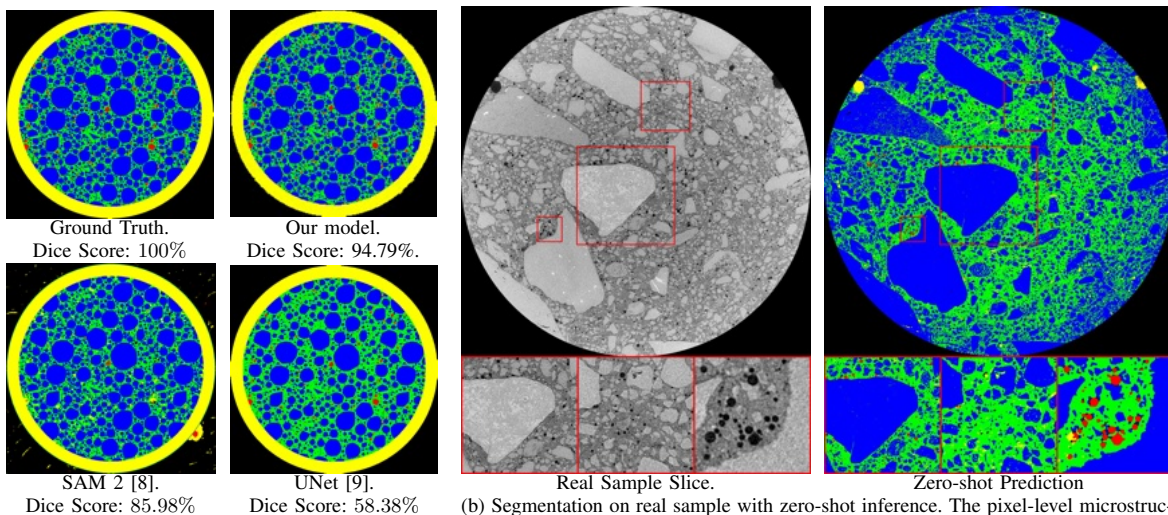
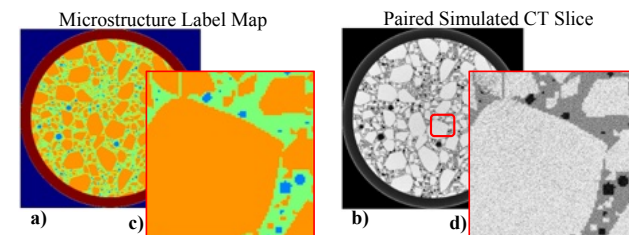
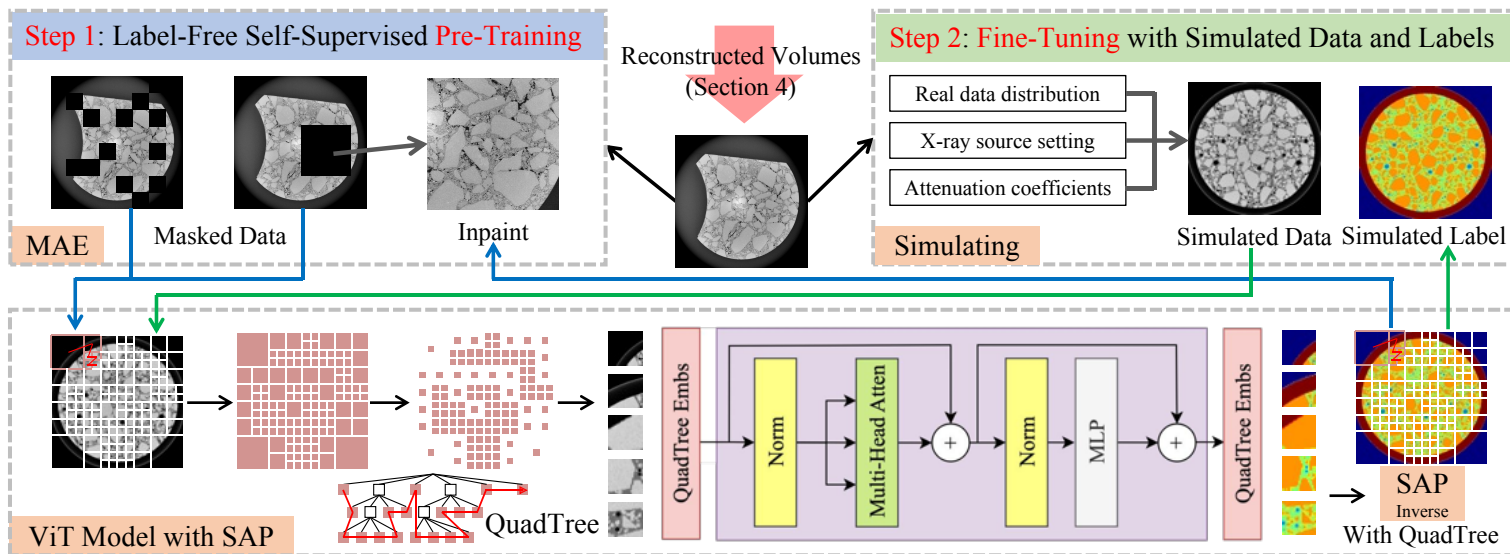


Cores were collected from road surfaces used for 20 to 30 years on the Hanshin Expressway.

High-resolution 4,192<sup>3</sup> Asphalt Core generated on Fugaku in 12 seconds (12,288 nodes: ~7% Fugaku)

\* Courtesy of Kentaro Uesugi (RIKEN Spring-8)

# Training a Foundation Model

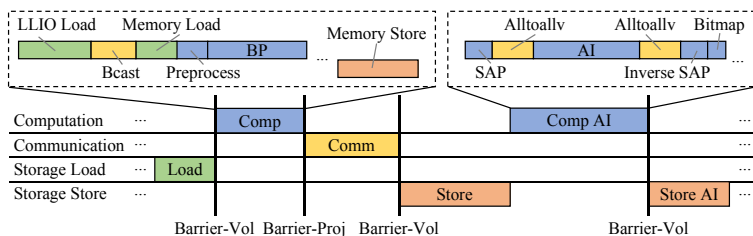


## Several Challenges

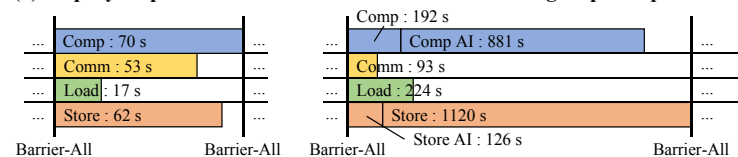
Sequence length, tokenization, shifting bottleneck, and **NO LABELED DATA**

# Results

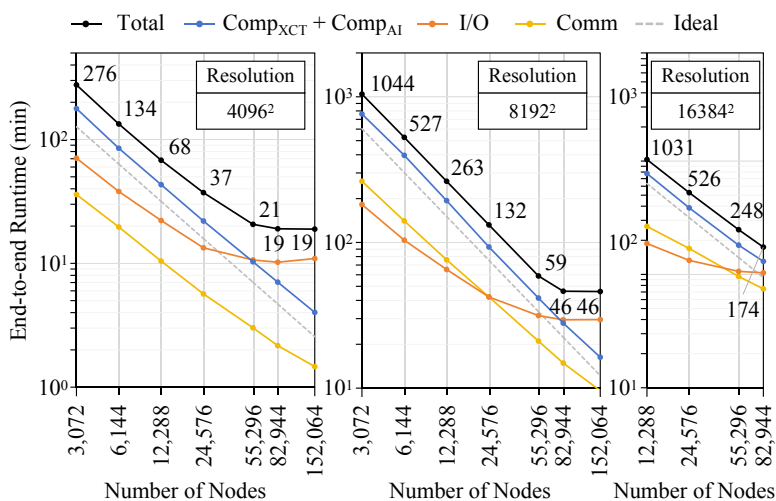
Performance
Analytics



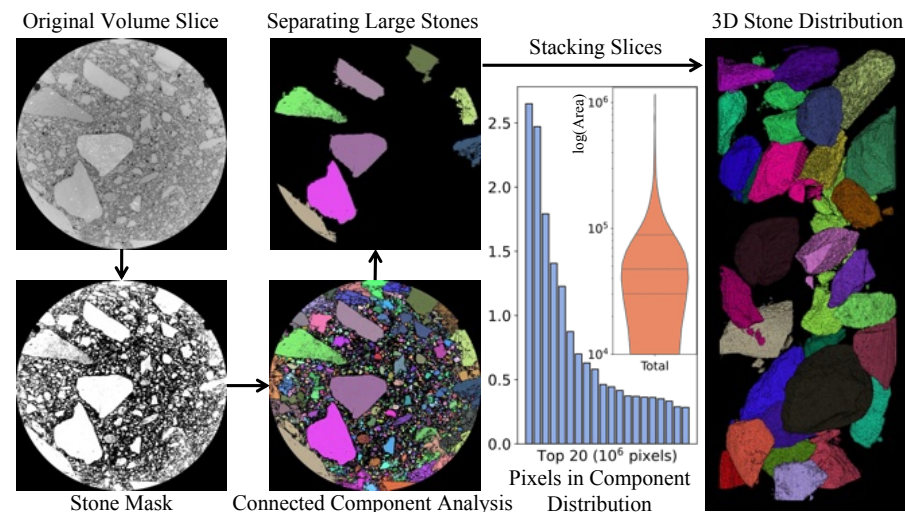
(a) Step-by-step workflow of the ROVAI framework for group row partitions.



(b) Pipeline for H3 imaging. (c) Pipeline for entire ROVAI framework.



(a) Strong scaling evaluated by full Fugaku; ROVAI reconstructing 46 specimens into various resolution.



| Road Inspection Analysis Items                 | T | XCT | XCT+AI |
|------------------------------------------------|---|-----|--------|
| Detect surface degradation                     | ✓ | ✓   | ✓      |
| Visualize sub-surface conditions               |   | ✓   | ✓      |
| Measure road shear strain angle                |   | ✓   | ✓      |
| Measure density of concrete and aggregate      |   | ✓   | ✓      |
| Measure the distribution of voids              |   | ✓   | ✓      |
| Measure the state of material around voids     |   | ✓   | ✓      |
| Measure volume ratio of aggregate, stone, etc. |   | ✓   | ✓      |

| Datset                                                             | Model            | Patch Size       | GPU (hours) | Epochs | Dice (%)     |
|--------------------------------------------------------------------|------------------|------------------|-------------|--------|--------------|
| 780 unique volumes<br>w/ simulated masks<br>(8,192×8,192×(50~120)) | U-Net [9]        | N/A              | 1,280       | 500    | 58.38        |
|                                                                    | Swin UNETR [10]  | 256 <sup>2</sup> | 5,120       | 1,000  | 63.74        |
|                                                                    | SAM 2[8]         | 128 <sup>2</sup> | 5,120       | 1,000  | 85.98        |
|                                                                    | <b>Our Model</b> | 2 <sup>2</sup>   | 5,120       | 1,000  | <b>94.79</b> |

# Workshop Questions

**Q1** : What missing interoperability layers (software, standard, or abstraction) would most accelerate convergence between traditional HPC linear algebra workflows and today's extreme-scale AI workloads.

**A1**: Coupling HPC application with AI is a challenge

**Q2** : Looking ahead to 2030, do you expect the principal bottleneck for extreme-scale AI to be data, algorithms, resilience or energy, and how does that prediction shape your research priorities today

**A2**: Money.

**Q3** : Given the different developments in architecture processors for AI and "computational science", do you think we'll see a convergence or divergence of roadmaps?

**A3**: HPC has, and will, adapt itself to the hardware designed for AI