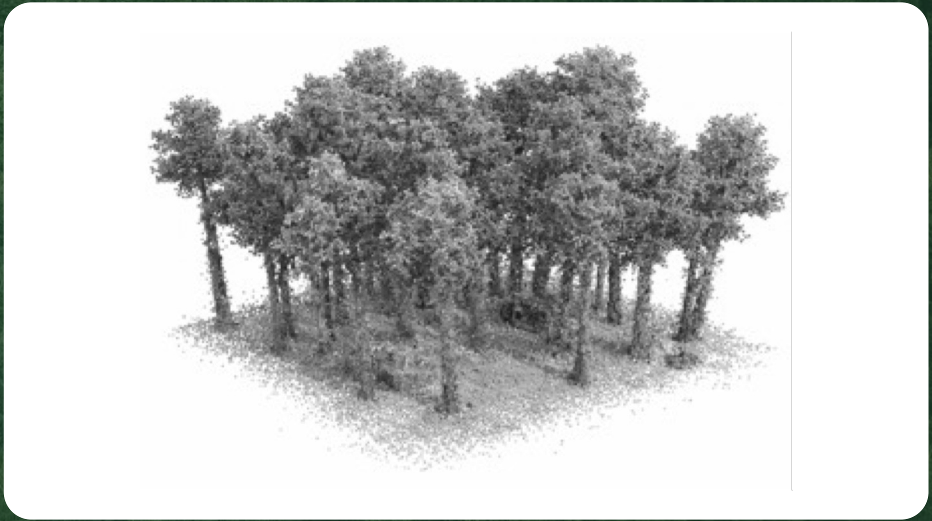
Freiburg ·2026 September 17

Seeing every tree in the forest

Structure-aware and promptable tree segmentation
in 3D forest LiDAR point clouds

Trung Thanh Nguyen
Nagoya University · RIKEN, Japan



ABOUT ME

Trung Thanh Nguyen

M.Sc. graduation
Nagoya University, Japan (2024)

EDUCATION

2022 Bachelor
Hanoi Univ. of Science and Technology, Vietnam

2024 Master
Nagoya University, Japan

2027 PhD (Expected)
Nagoya University

Nagoya University, Nagoya, Japan

PhD Candidate, Graduate School of Informatics, Department of Intelligent Systems

RIKEN, Kyoto, Japan

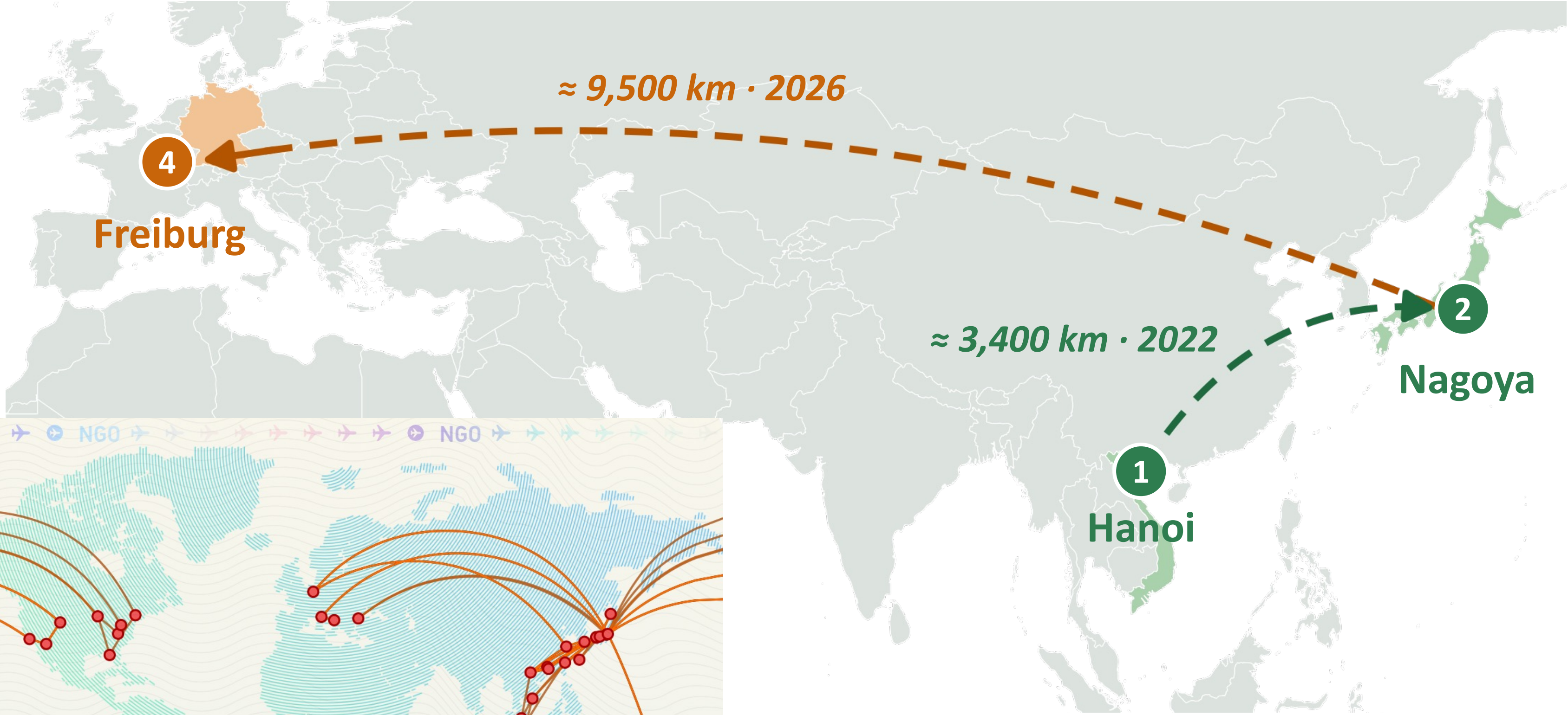
- Student Trainee
- Guardian Robot Project
- Multimodal Data Recognition Research Team

University of Freiburg, Germany

- Rising Star Fellow
- Cluster of Excellence Future Forests
- Host PI: **Prof. Teja Kattenborn**

ABOUT ME

From Hanoi, to Nagoya, to Freiburg



Nagoya University ↔ RIKEN GRP
≈ 120 km

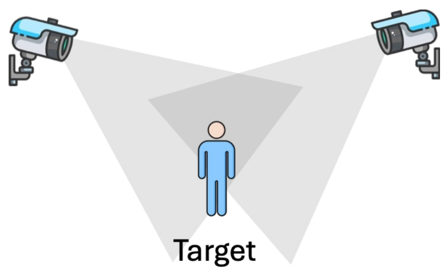

Hanoi, Vietnam


Nagoya, Japan


Kyoto, Japan

ABOUT ME

My research at a glance



PhD · NAGOYA & RIKEN

Human Actions from Multiple Sensors

#Keyword: Action Recognition, Multi-view, Multi-modal, Open-vocabulary

UNIVERSITY of FREIBURG

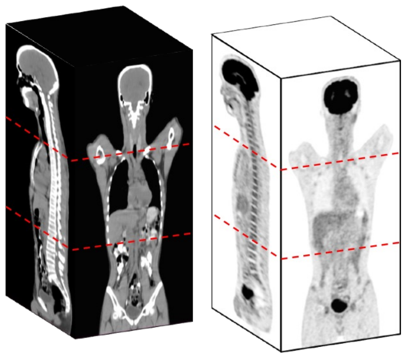
3D Forest Point Clouds
3dtrees.earth

#Keyword: 3D Trees, 3D Point Clouds, Instance/Semantic Segmentation

UCLA, USA

4D Point Cloud Representations

#Keyword: 4D Point Cloud, Representation Learning



AI4LIFE · HUST, VIETNAM

Medical Imaging and VLMs

#Keyword: Diffusion Model, Report Generation, CT/PET Imaging

BACKGROUND

Forests are 3D, and ecology needs them tree by tree

31%

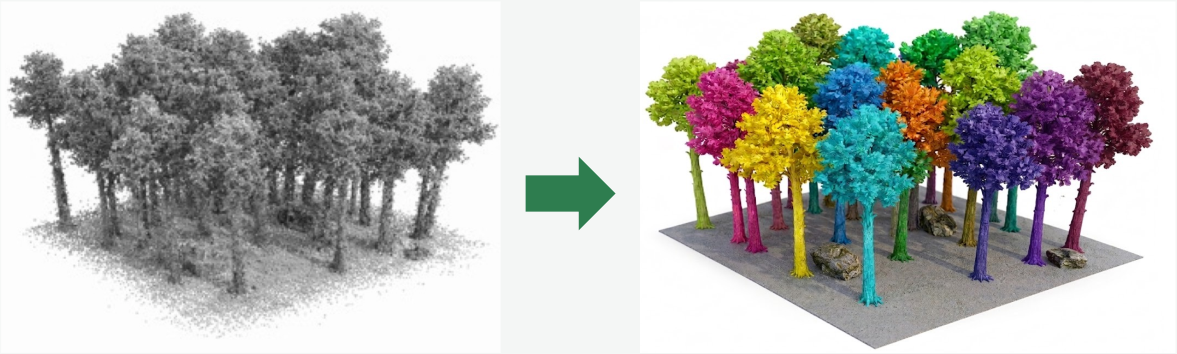
of Earth's land surface is covered by forest

Biomass and carbon
Stem and crown volume of each tree

Crown architecture
Canopy layers between neighbours

Health and biodiversity
Species, foliage vs. wood, forest condition

Joint Semantic + Instance Segmentation



Terrestrial,
Drone LiDAR Point Cloud

One label Per-tree,
+ Ground / Wood / Leaf

Every downstream trait starts from an accurate per-tree mask

BACKGROUND

Forest LiDAR is much harder than indoor 3D scenes

Massive Scale

Hundreds of millions of points per hectare at high acquisition density

Irregular Density

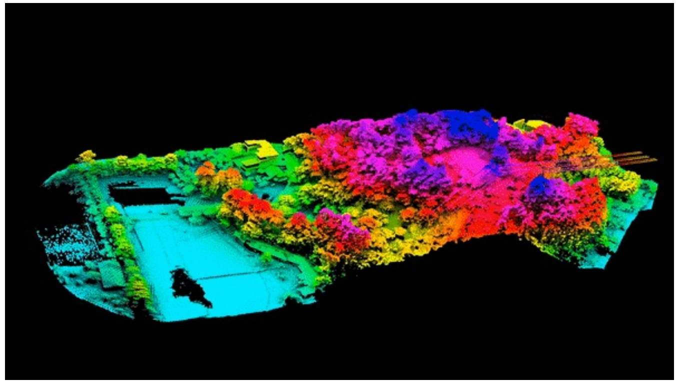
Occlusion and sensor geometry leave the upper canopy and far stems sparse

Entangled Canopies

Overlapping crowns make neighbouring trees hard to separate

Class Imbalance

Leaf points dominate;
Ground and Wood are much rarer

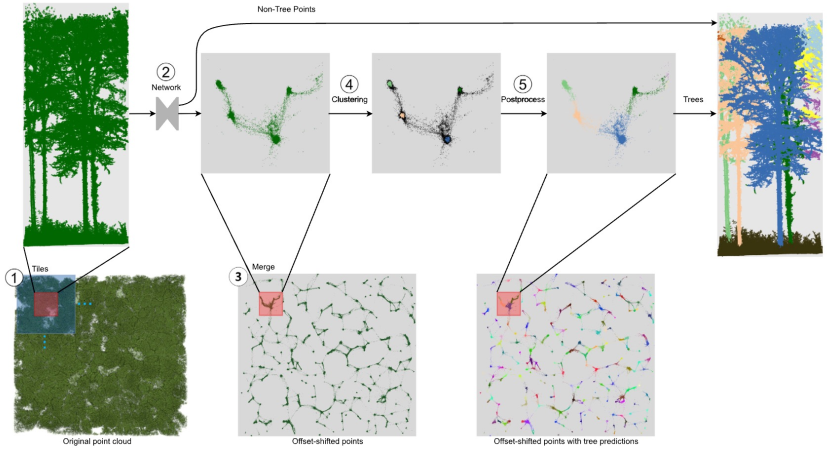


Existing methods and what is still missing

Clustering-based

ForAINet · TreeLearn · SegmentAnyTree

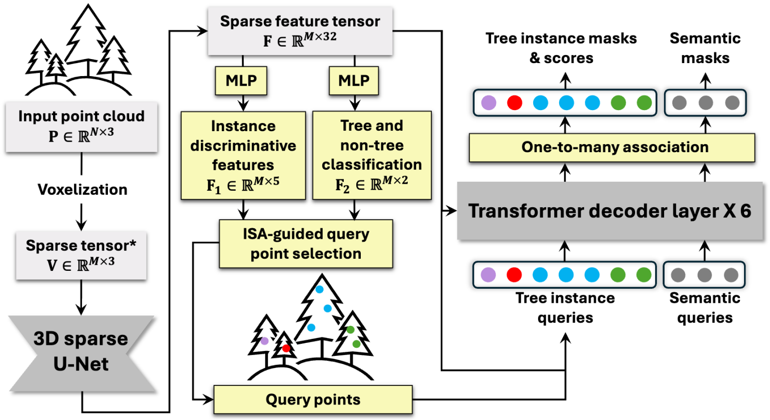
- Sparse-conv U-Net + mean-shift / DBSCAN grouping
- Hand-crafted grouping breaks in dense canopies



Query-based

OneFormer3D · ForestFormer3D

- Transformer-based: Global attention + mask queries
- State-of-the-art accuracy



* Represents non-empty voxels as a sparse feature tensor, where M is the number of voxels containing actual points.

Three Gaps

1. **Inference Cost**
   Quadratic attention: slow and memory-hungry on large scenes
2. **No Forest Priors**
   Generic queries ignore canopy geometry, miss weak trees
3. **No User Control**
   All-or-nothing output: cannot pick or fix one tree

- **Gaps 1–2 → ForestMamba**
- **Gap 3 → SelectAnyTree**

OUTLINE

Two steps: faster automatic, then interactive

PART 1

ForestMamba

BMVC 2026

Can forest structure make automatic segmentation both more accurate and cheaper?

1. Vertical-priority sparse Mamba encoder
2. Treetop-seeded queries
3. Linear-time Mamba query decoder

PART 2

SelectAnyTree

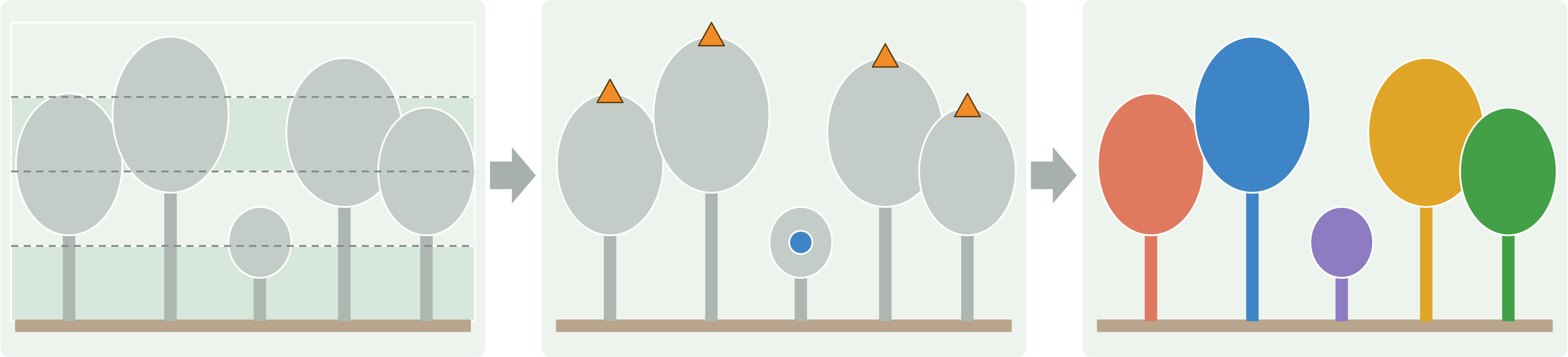
Under Review

Can a user select, and correct, any single tree with a few clicks?

1. Encode the scene once, reuse for every click
2. Click-to-query prompt encoder
3. Treetop-seeded as a free first click

ForestMamba: the idea

In: LiDAR points (x, y, z) **Out:** every tree, plus ground / wood / leaf



1 Read the Forest

Layer by layer, from the ground up

2 Mark the Treetops

One marker per tree, plus extra ones for small trees

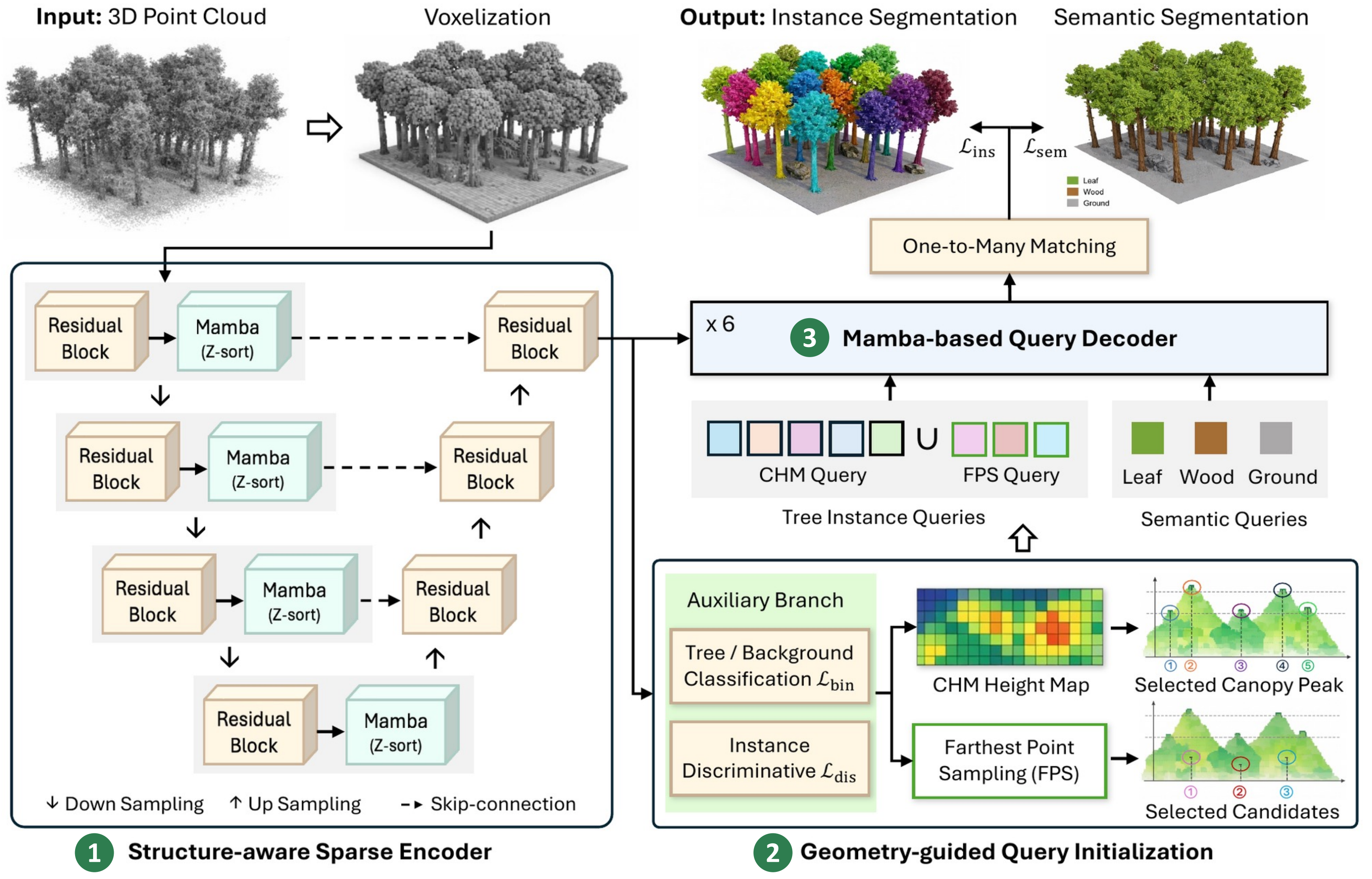
3 Grow Each Tree

Each marker collects the points of its tree

In Plain Words

ForestMamba uses two things every forester knows: Trees grow upward, and every tree has a top

ForestMamba



1 Sparse Encoder

Sparse residual blocks + Mamba; voxels ordered ground-to-canopy

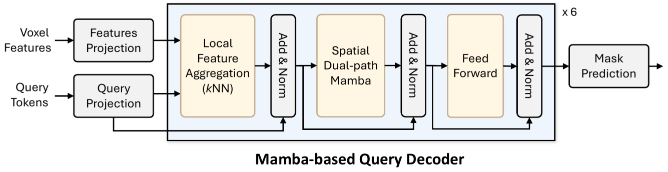
2 Geometry-guided Query Initialization

Auxiliary branch (tree / background) → CHM peaks + FPS queries

3 Mamba-based Query Decoder

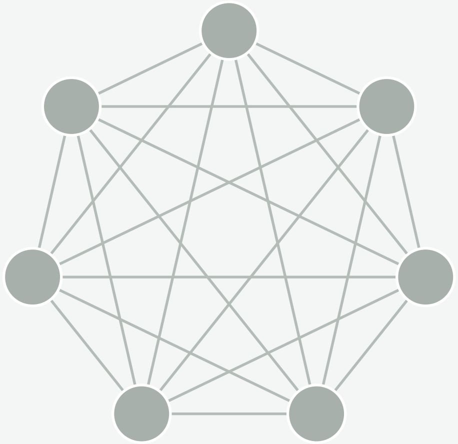
× 6 layers; one-to-many matching; outputs trees + ground / wood / leaf

INSIDE 3: ONE DECODER LAYER



Why Mamba?

Transformer: Everyone talks to everyone



Like a meeting where every person talks with every other person

2× bigger plot
→ **4× more work**

Used by the previous best models: accurate, but slow and memory-hungry on large plots

Mamba: Pass a note down the line

The note is passed along →

Like reading a book once while taking notes

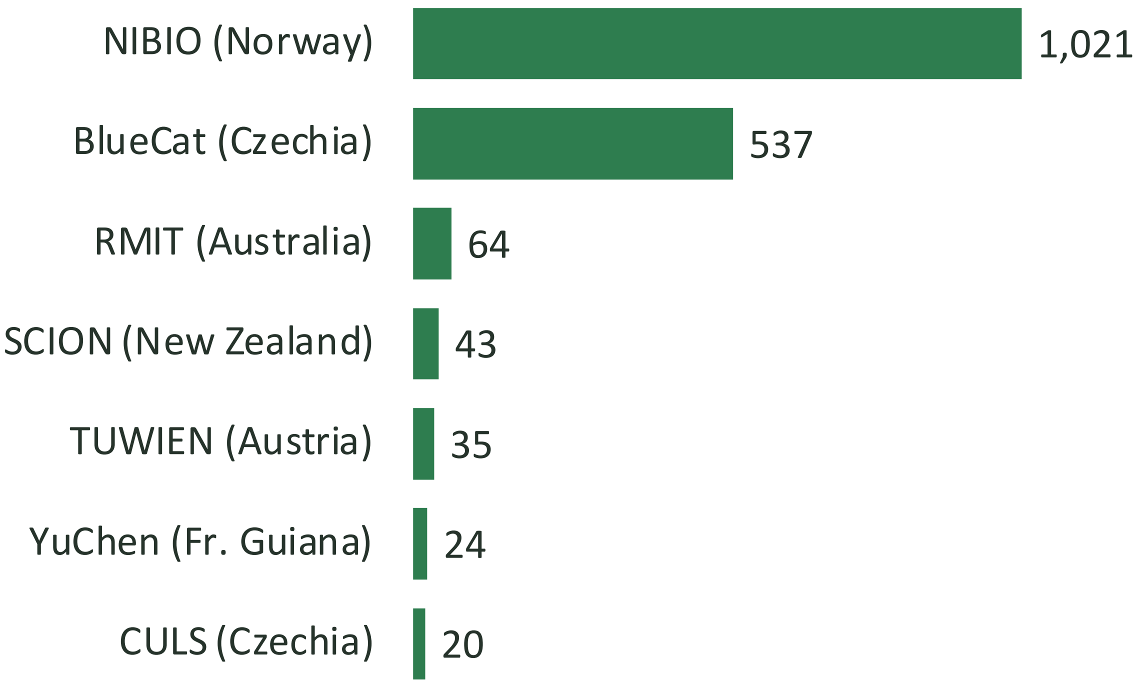
2× bigger plot
→ **2× more work**

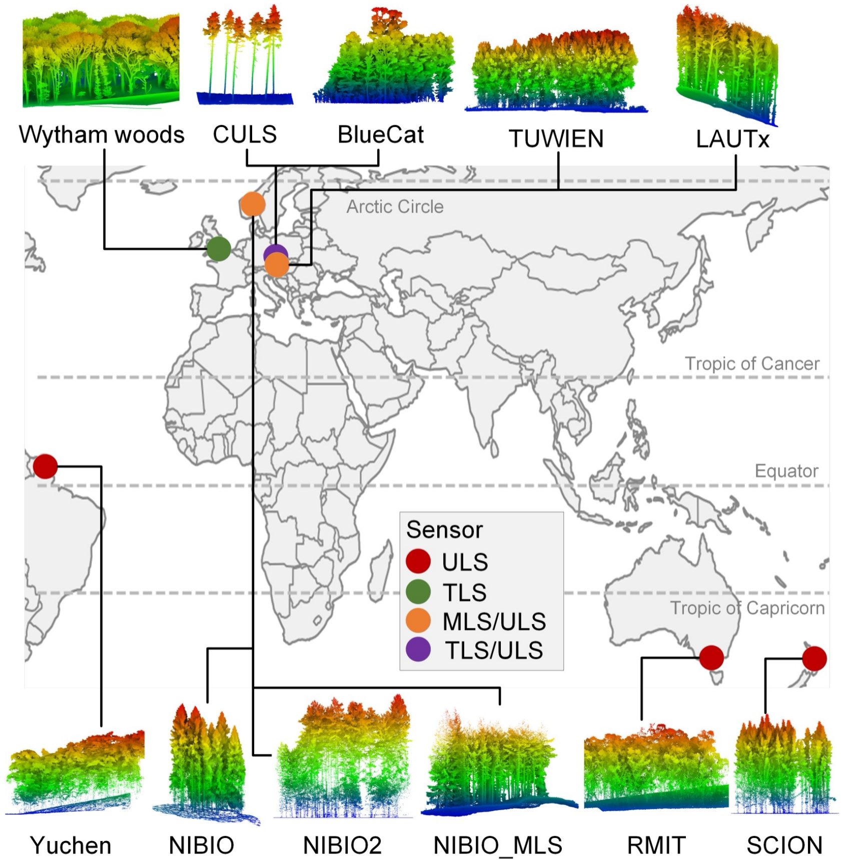
One condition: the reading order must make sense. For trees, we read from the ground up

In plain words: Mamba keeps the model fast even for very large forest plots

How we test it: seven forests, four continents

Test trees per plot region (FOR-instanceV2)





In plain words: Seven very different forests – four continents, from boreal to tropical
Evaluation metrics: Precision, Recall, F1, Coverage, and mIoU

Quantitative results

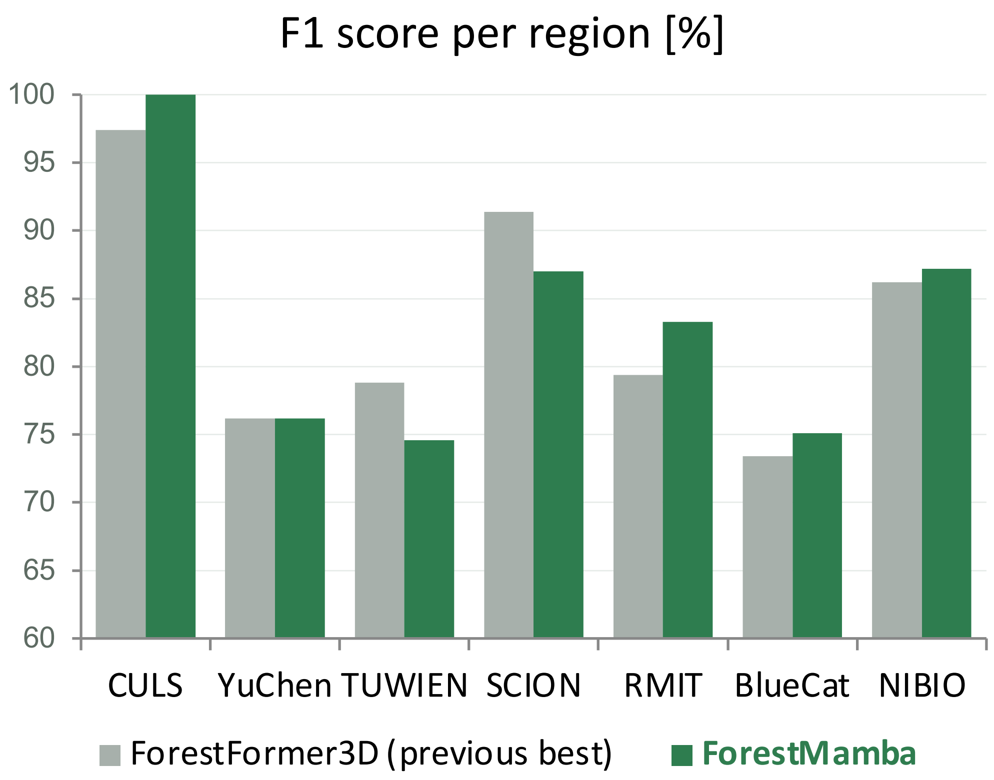
Method	P	R	F1	Cov.	mIoU
TreeLearn	82.0	36.6	50.6	52.2	–
ForAINet	86.7	65.8	72.8	63.9	82.0
OneFormer3D	82.2	67.3	74.0	63.8	80.8
ForestFormer3D	91.3	74.9	82.3	70.5	82.1
ForestMamba	**91.4**	**76.6**	**83.4**	**71.9**	**82.3**

P = Precision; R = Recall; Cov. = Coverage

+1.7
Recall (R): more real trees found

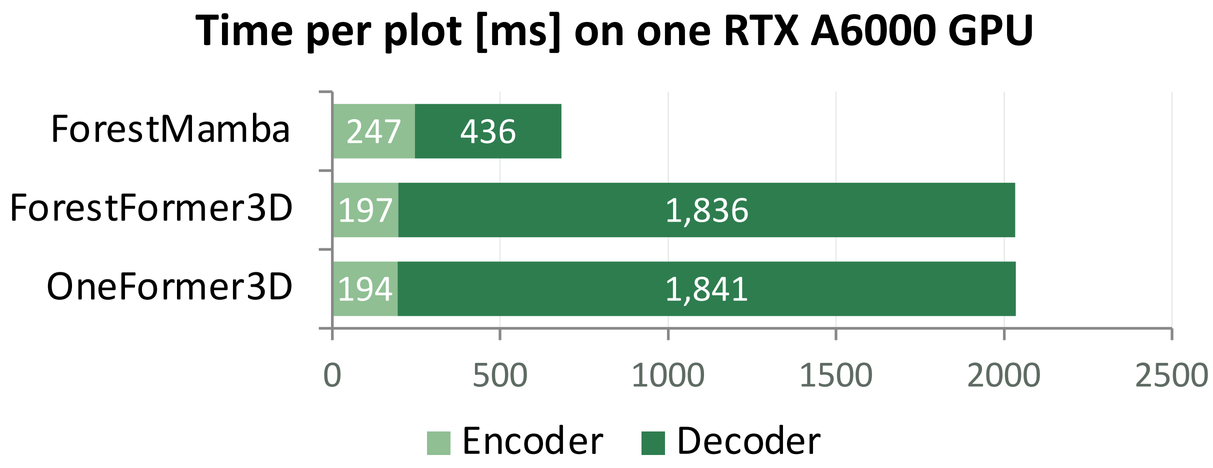
+0.1
Precision (P): no extra false trees

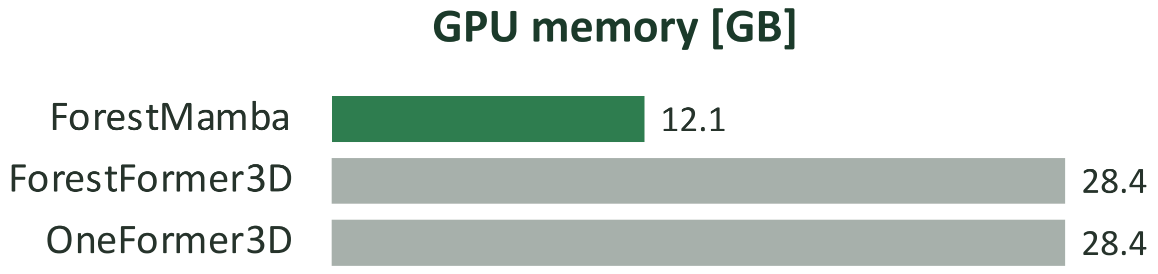
+1.1
F1 vs. best previous model



In plain words: ForestMamba finds more of the real trees without adding false ones
Gains are largest in big, dense plots; two small plots (TUWIEN, SCION) are slightly worse

About 3× faster, with less than half the memory





2.0 s → 0.7 s
time per plot

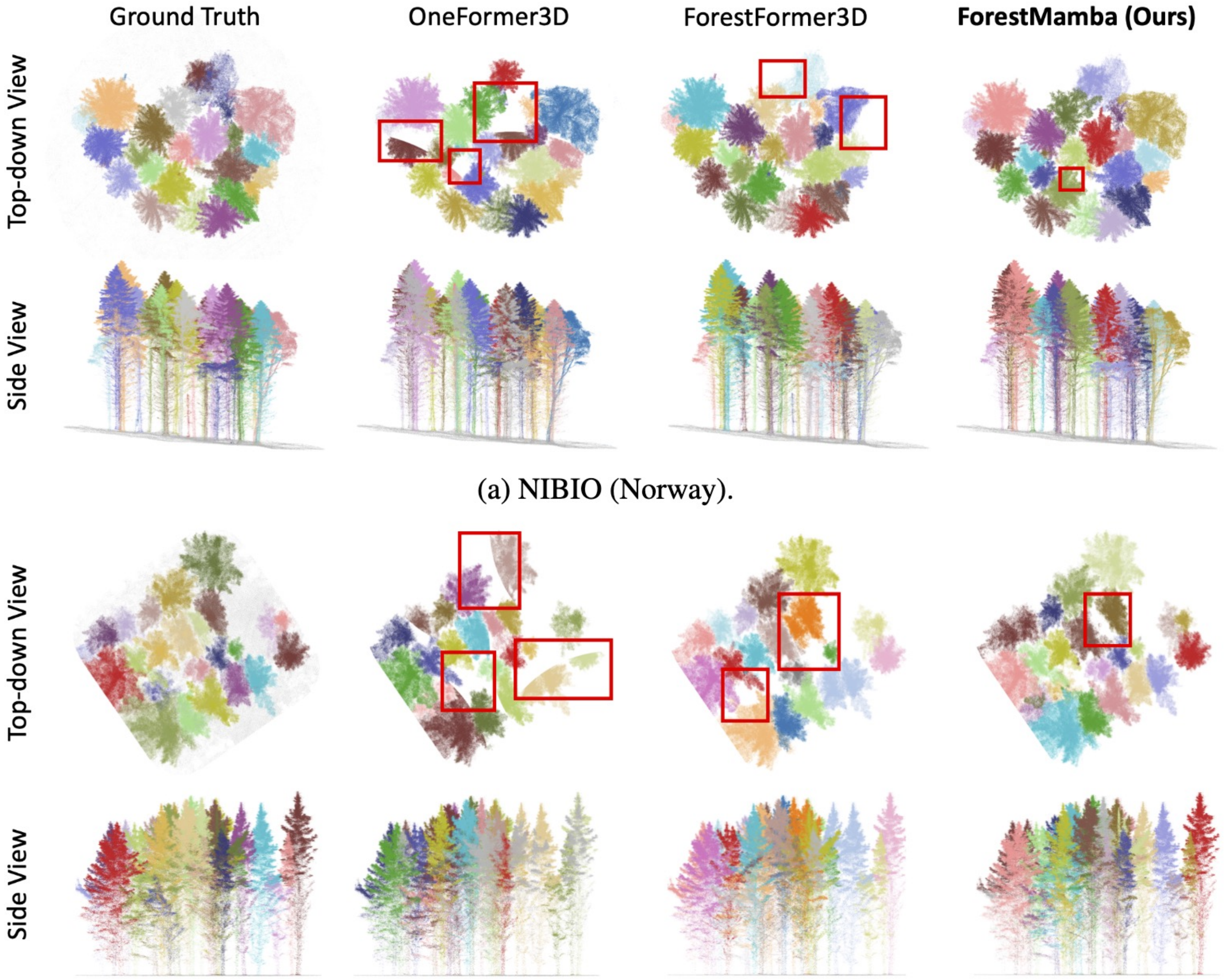
28 → 12 GB
GPU memory

4.2×
faster tree outlining

same size
17.3M vs 17.5M parameters

In plain words: On the same GPU, a plot that took about 2 seconds now takes 0.7 seconds, and it needs 12 GB instead of 28 GB of GPU memory

Qualitative results



(a) NIBIO (Norway).

(b) SCION (New Zealand).

What to look for

- **Red boxes:** missing or fragmented trees
- **Previous models:** several errors per scene in dense crowns and partly hidden trees
- **ForestMamba:** one small error per scene, recovers missing trees in top-down view
- **Side view:** stems and crowns are segmented as one coherent instance

PART 2 · SELECTANYTREE

Automatic is not enough for annotation

A fixed set of masks

Even at 83 F1, a user cannot pick one tree or repair a wrong one; errors mean manual editing or retraining

Labels are the bottleneck

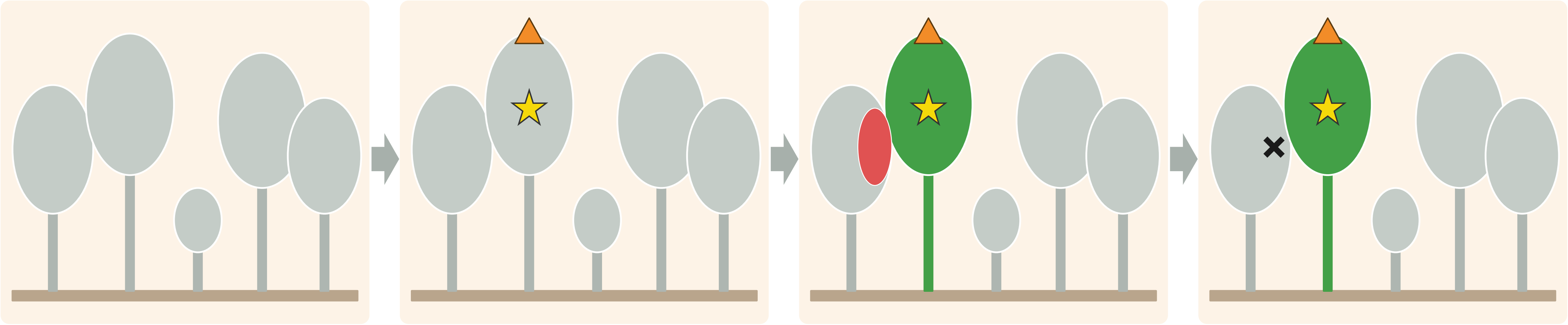
A hectare has millions of points and hundreds of crowns; per-tree annotation is slow expert work

Promptable segmentation

3D versions exist, but for indoor objects, shapes, and parts, not forests

Can a user select, and correct, any tree with a few clicks?

SelectAnyTree: the idea



1 Load the plot
read once,
kept in memory

2 Click a tree
its treetop (▲); optional

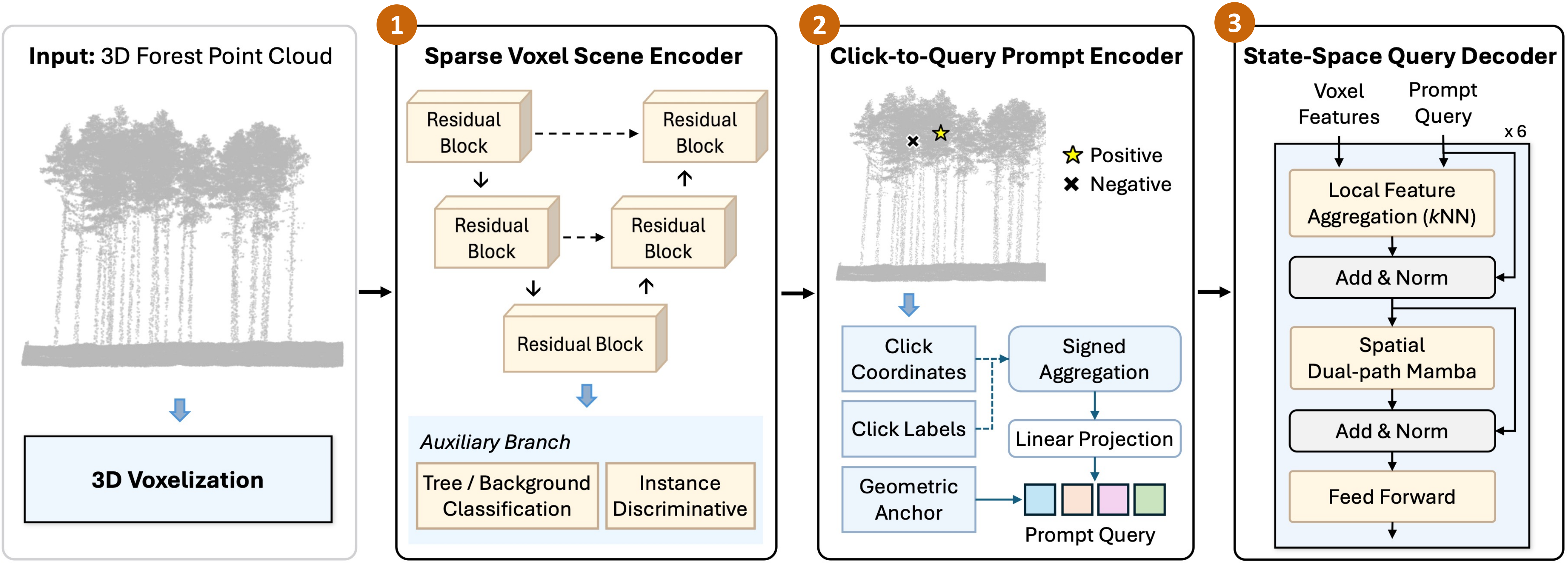
3 See the tree
here: a bit too much (red)

4 Fix with a click
× = “not this part”

★ click = “this tree” × click = “not this part” each click gives an answer in a fraction of a second

In plain words: It works like the “magic wand” in photo editing, but for single trees in a 3D forest

SelectAnyTree: from clicks to one tree mask



Task **Point cloud P, clicks C → Tree mask**

① Load the plot once → ② The click become prompt **(Main of SelectAnyTree)** → ③ See the tree

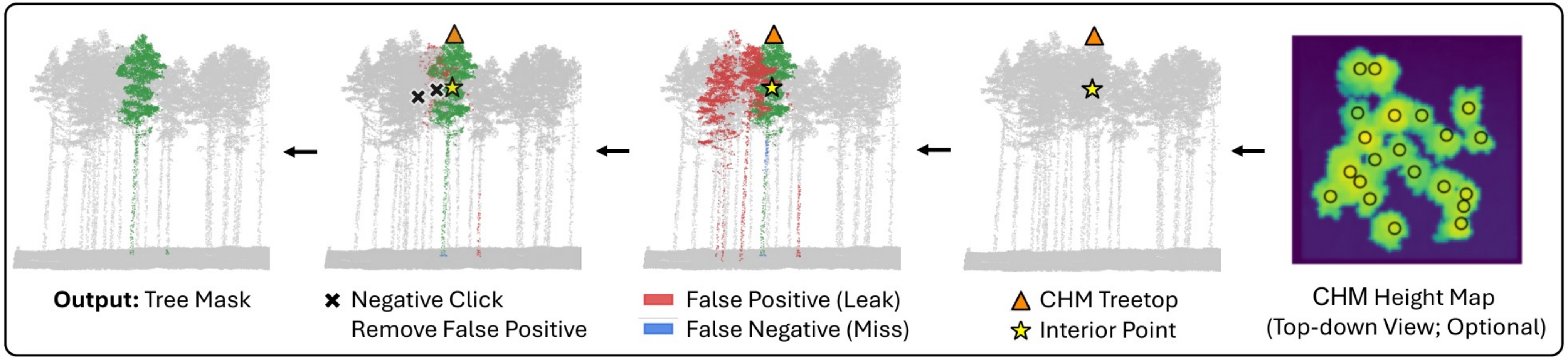
How we test it: a program plays the annotator

1 Click the tree centre
Plus the treetop (Optional)

2 Look at the answer
Compare it with the real tree

3 Click the biggest mistake
★ for a missed part, × for a wrong part; up to 5 clicks

A EXAMPLE (READ FROM RIGHT TO LEFT)



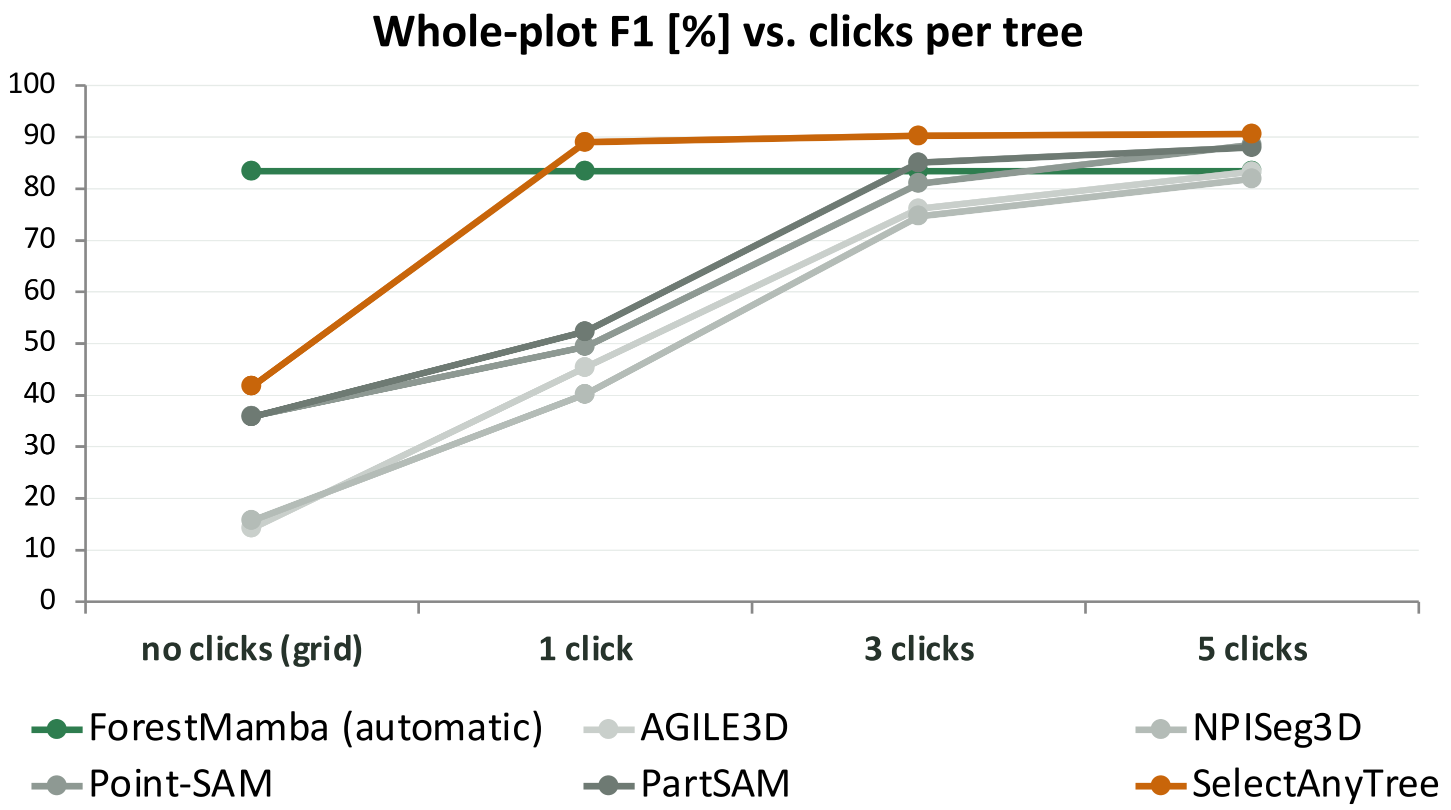
In plain words: The program always clicks exactly where the model is most wrong
Real people are less precise, so these are best-case numbers

One click → about 80% overlap with the real tree

Method	In-distribution						Cross-dataset Generalization						CHM helps?
	IoU @ Clicks ↑			NoC @ IoU ↓			IoU @ Clicks ↑			NoC @ IoU ↓			
	1	3	5	0.7	0.8	0.9	1	3	5	0.7	0.8	0.9	
AGILE3D [36]	43.8	64.6	71.9	4.6	5.9	7.8	40.8	62.0	68.7	5.2	6.4	8.3	✗
+ w/ CHM	43.7	59.3	66.7	5.3	6.5	8.3	40.6	56.0	62.5	5.7	7.0	8.8	
NPISeg3D [14]	36.9	61.0	68.5	5.0	6.3	8.1	38.3	59.3	65.1	5.3	6.9	8.8	✓
+ w/ CHM	46.4	63.9	68.3	4.8	6.2	8.2	42.6	61.7	69.3	5.1	6.5	8.5	
Point-SAM [38]	53.4	74.8	80.6	3.4	4.4	6.1	54.5	75.0	81.2	3.3	4.5	6.5	✓
+ w/ CHM	55.2	75.1	81.6	3.4	4.4	6.0	55.7	76.2	81.5	3.2	4.5	6.5	
PartSAM [39]	48.8	71.2	78.3	3.7	4.8	6.6	31.4	53.9	63.9	5.9	7.1	8.5	✓
+ w/ CHM	53.1	74.0	80.8	3.6	4.7	6.8	44.8	66.1	75.5	4.3	5.8	7.8	
SelectAnyTree (Prop.)	77.9	81.3	81.1	3.0	3.9	5.5	76.5	80.3	81.6	2.7	3.7	6.0	✓
+ w/ CHM	**79.9**	**81.9**	**82.4**	**2.9**	**3.8**	**5.4**	**77.7**	**81.5**	**82.2**	**2.6**	**3.4**	**5.7**	

In plain words: With one click, SelectAnyTree outlines about 80% of the tree correctly
Other methods reach about 55%

One click per tree beats the best automatic model



In plain words: Without a person, use the automatic model, **ForestMamba**
If someone can give each tree one click, **SelectAnyTree** gives a better result (89.0 F1)

Smallest and fastest promptable model

Method	Params [M] ↓	Time [ms] @ Clicks ↓			Peak GPU [GB] ↓
		1	3	5	
AGILE3D [36]	39.3	412 ± 241	1,828 ± 653	3,286 ± 1,043	**1.0**
NPISeg3D [14]	40.0	249 ± 131	1,307 ± 386	2,353 ± 664	3.1
Point-SAM [38]	311.0	1,612 ± 602	5,062 ± 2,035	8,587 ± 3,472	22.4
PartSAM [39]	225.0	1,549 ± 581	2,558 ± 986	3,562 ± 1,423	2.4
SelectAnyTree (Prop.)	**19.4**	**122 ± 98**	298 ± 196	571 ± 184	1.9
+ w/ CHM	**19.4**	164 ± 159	**280 ± 156**	**549 ± 126**	1.9

19.4M

Parameters
16× fewer than Point-SAM (311M)

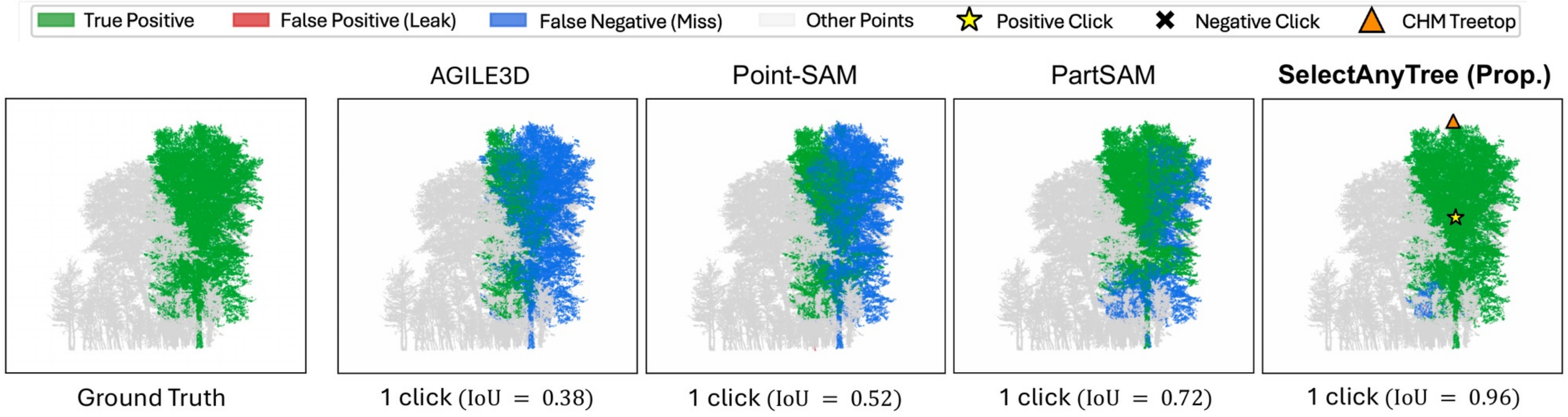
571 ms

For 5 clicks
4× faster than NPISeg3D, 15× faster than Point-SAM

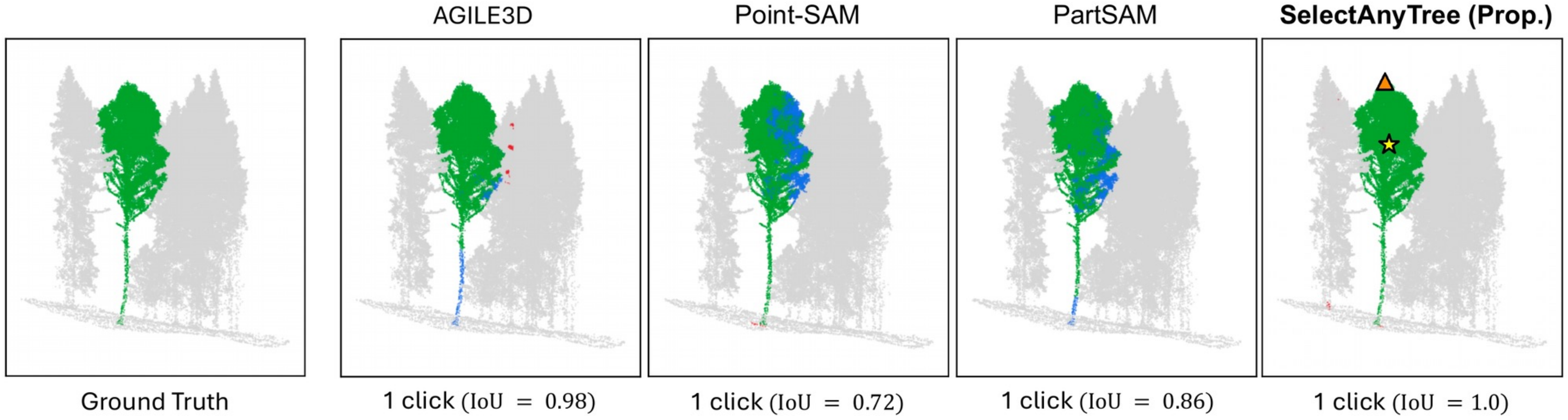
1.9 GB

Peak GPU memory
12× less than Point-SAM; AGILE3D uses 1.0 GB

Qualitative results



(a) BlueCat (Czech Republic) [34]: A dense temperate forest with strongly overlapping crowns.



- **BlueCat (dense):** baselines miss large parts (blue); SelectAnyTree reaches 0.96 IoU
- **NIBIO (conifers):** all methods are closer; only SelectAnyTree gets 1.0
- **Two cues:** the treetop (▲) anchors the apex, the click (★) grounds the stem

SUMMARY

Two methods that work together

	ForestMamba	SelectAnyTree
In one sentence	Finds every tree and its parts automatically	Let pick and fix any tree with a click
Forest knowledge used	Trees grow upward; treetops mark trees	Treetops mark trees (Optional)
Key result	**83.4 F1 · about 3× faster** **2.3× less memory**	**About 80% overlap from 1 click** **89.0 F1 with 1 click per tree**
Good for	Large automatic forest inventories	Making training labels, fixing errors
Status	**BMVC 2026 · on 3Dtrees.earth**	**Under Review (3DV)**

In plain words: ForestMamba for a fast, automatic first pass
SelectAnyTree to check and correct individual trees in a few clicks