

Supplementary Information

AI-Driven Scan-to-Discovery in 20 Minutes: Automated Segmentation of Synchrotron Micro-CT at Scale

Xiaoya Chong¹⁺, David Abramov¹⁺, Pradyumna Elavarthi¹,
Elizabeth G. Clark¹, Petrus Zwart¹, Wiebke Koepp¹, Dilworth Y. Parkinson¹,
Dylan McReynolds¹, Hassina Bilheux⁷, Daniel Allan⁴, Hanfei Yan⁴, Johanna Nelson Weker⁵,
Mathew Joseph Cherukara⁶, Apurva Mehta⁵, Nicholas Schwarz⁶, Andrew McElrone^{2,3},
Tanny Chavez^{1*}, Alexander Hexemer^{1*}

¹Lawrence Berkeley National Laboratory, Berkeley, CA, United States

²USDA Agricultural Research Service, United States

³University of California Davis, Davis, CA, United States

⁴Brookhaven National Laboratory, Upton, NY, United States

⁵SLAC National Accelerator Laboratory, Menlo Park, CA, United States

⁶Argonne National Laboratory, Lemont, IL, United States

⁷Oak Ridge National Laboratory, Oak Ridge, TN, United States

⁺Equal contribution

^{*}Corresponding authors: tanchavez@lbl.gov, ahexemer@lbl.gov

June 21, 2026

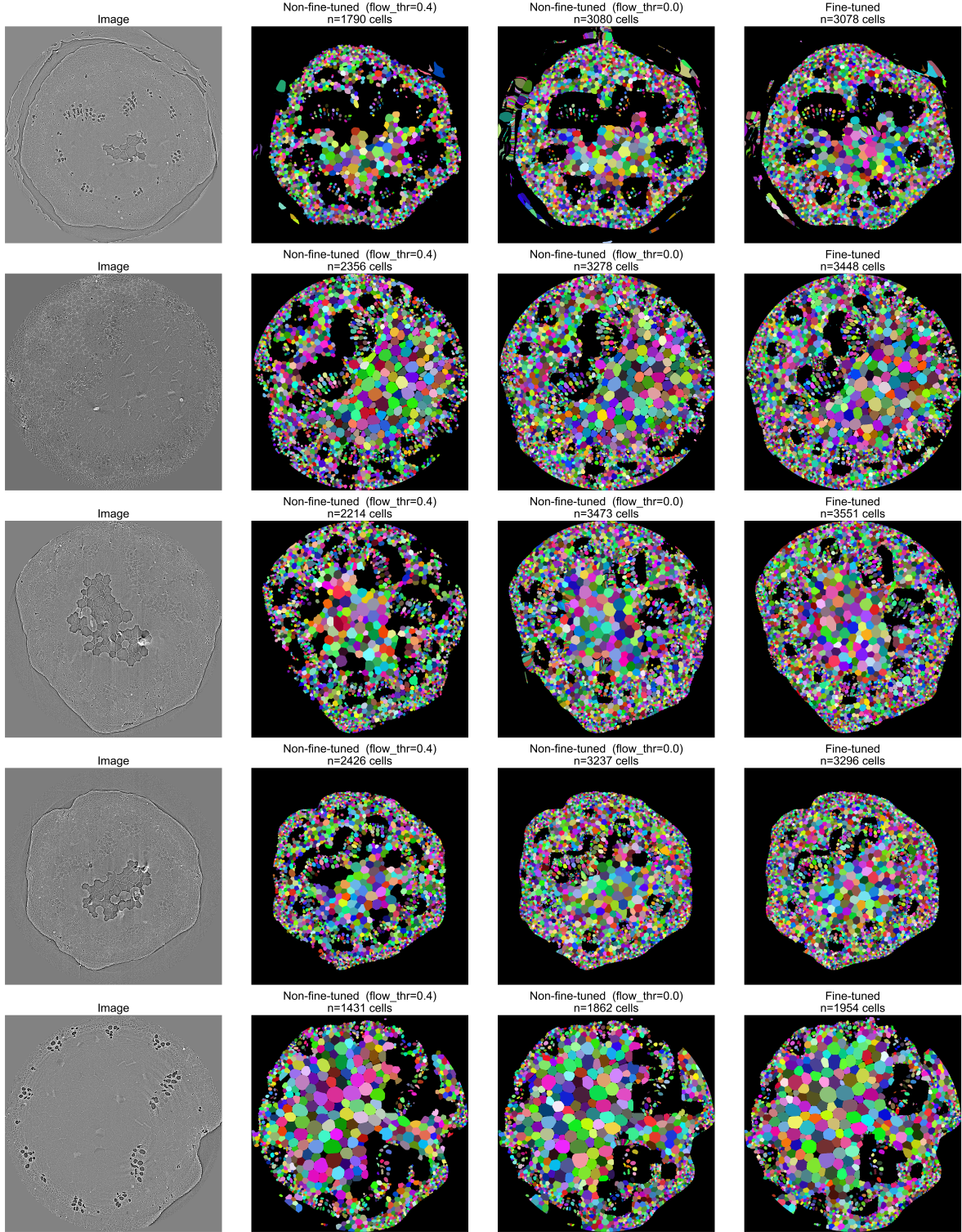


Figure 1: Comparison between non-fine-tuned and fine-tuned Cellpose-SAM [1] models on the test set. The flow threshold controls the maximum allowed flow error for each mask; setting it to 0 disables this constraint and enables detection of as many cells as possible. For the non-fine-tuned model, we use the default recommended flow threshold of 0.4 (second column). In the third column, we set the flow threshold to 0. The last column shows results from the fine-tuned model, which performs better than both previous settings: it detects more cells while removing several artifacts.

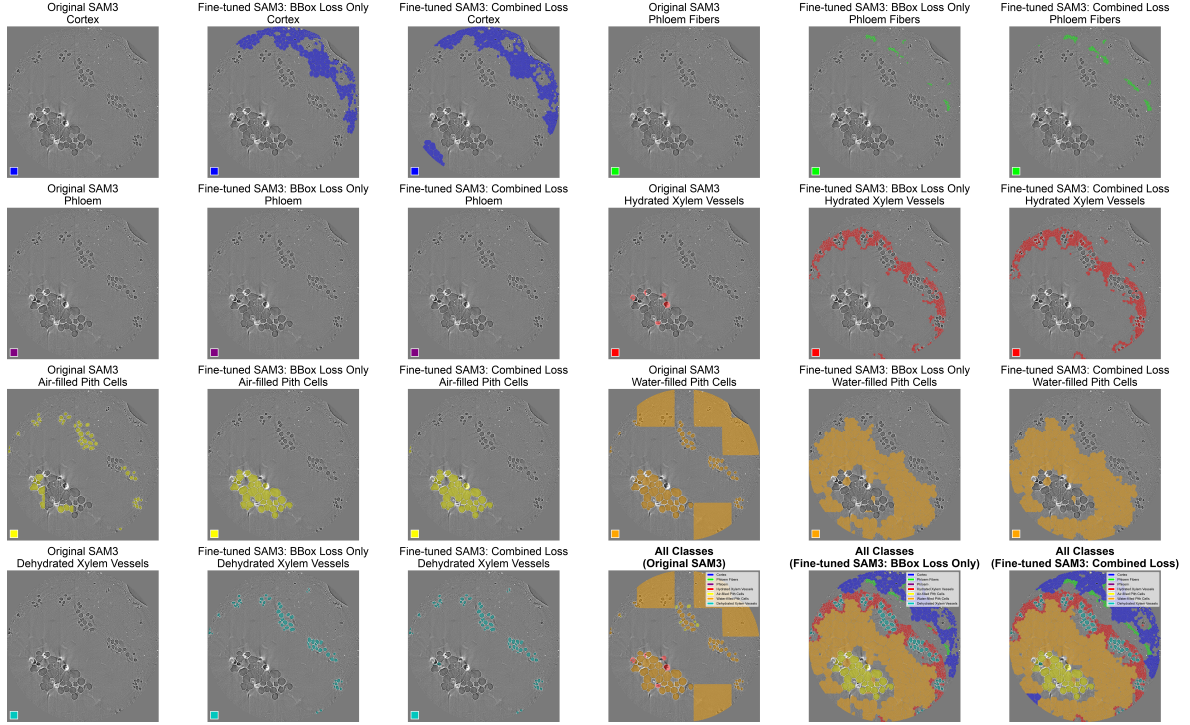


Figure 2: Comparison between non-fine-tuned and fine-tuned SAM3 [2] models on the test set. The non-fine-tuned model fails on this task. The model fine-tuned with a combined loss generally predicts more cells than the model trained using only bounding-box loss. All models struggle to distinguish phloem fibers from hydrated xylem vessels, because these structures appear visually similar. To address this issue, we incorporate DINOv3-Seg (DINOv3 [3] with a mask module) to provide global context and filter out incorrect predictions. In practice, detecting more cells (including some artifacts) is preferable to missing cells, since false positives can be filtered later using the DINOv3-Seg semantic mask. Therefore, the combined loss provides a better trade-off for this task.

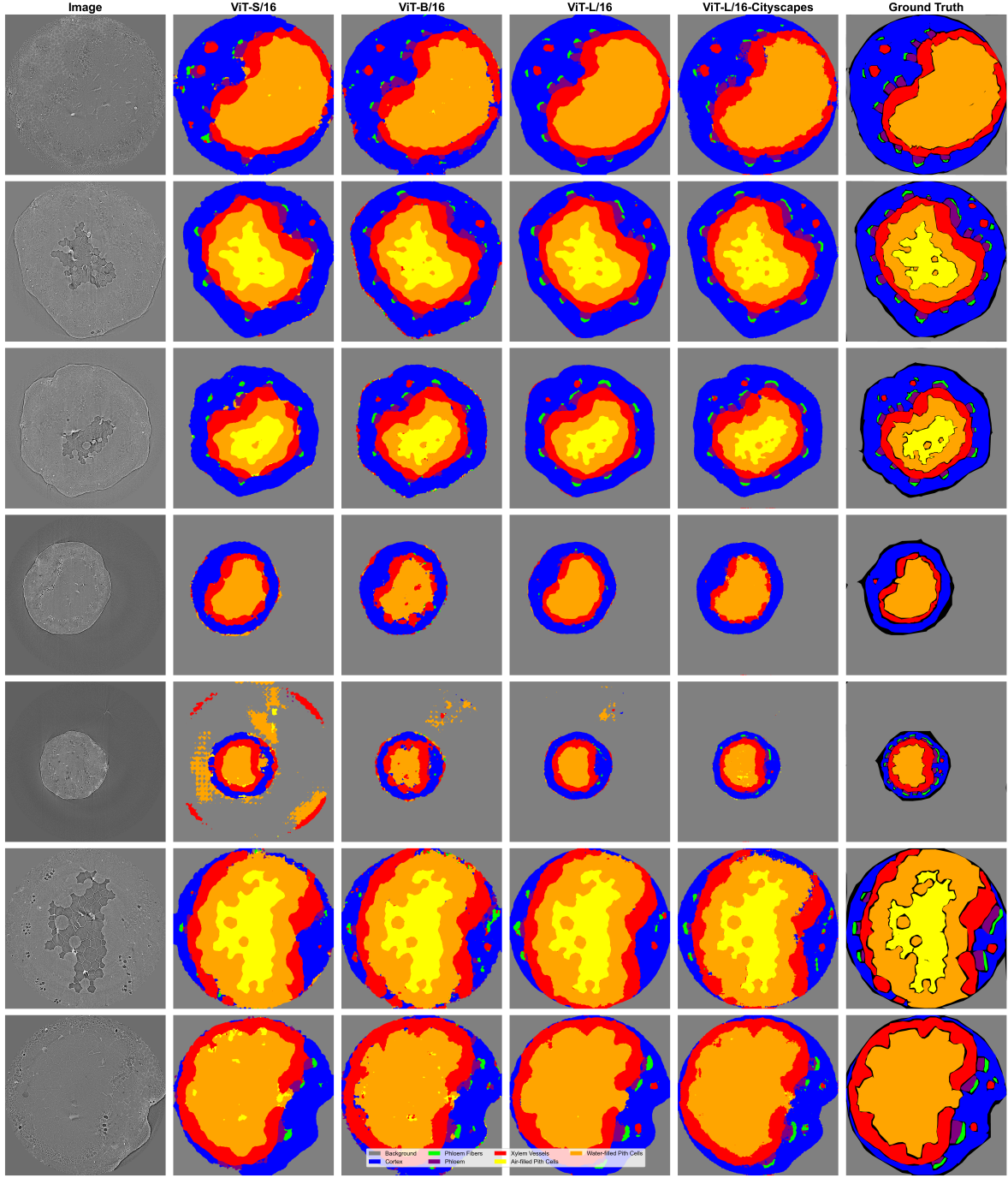


Figure 3: Comparison of different fine-tuned DINOv3-Seg backbones on the test set. This semantic segmentation task contains six classes, where hydrated xylem vessels and dehydrated xylem vessels are merged into a single class. The evaluated backbones include ViT-S/16, ViT-B/16, and ViT-L/16. The ViT-L/16-Cityscapes model is first fine-tuned on the Cityscapes [4] dataset and then further fine-tuned on our dataset.

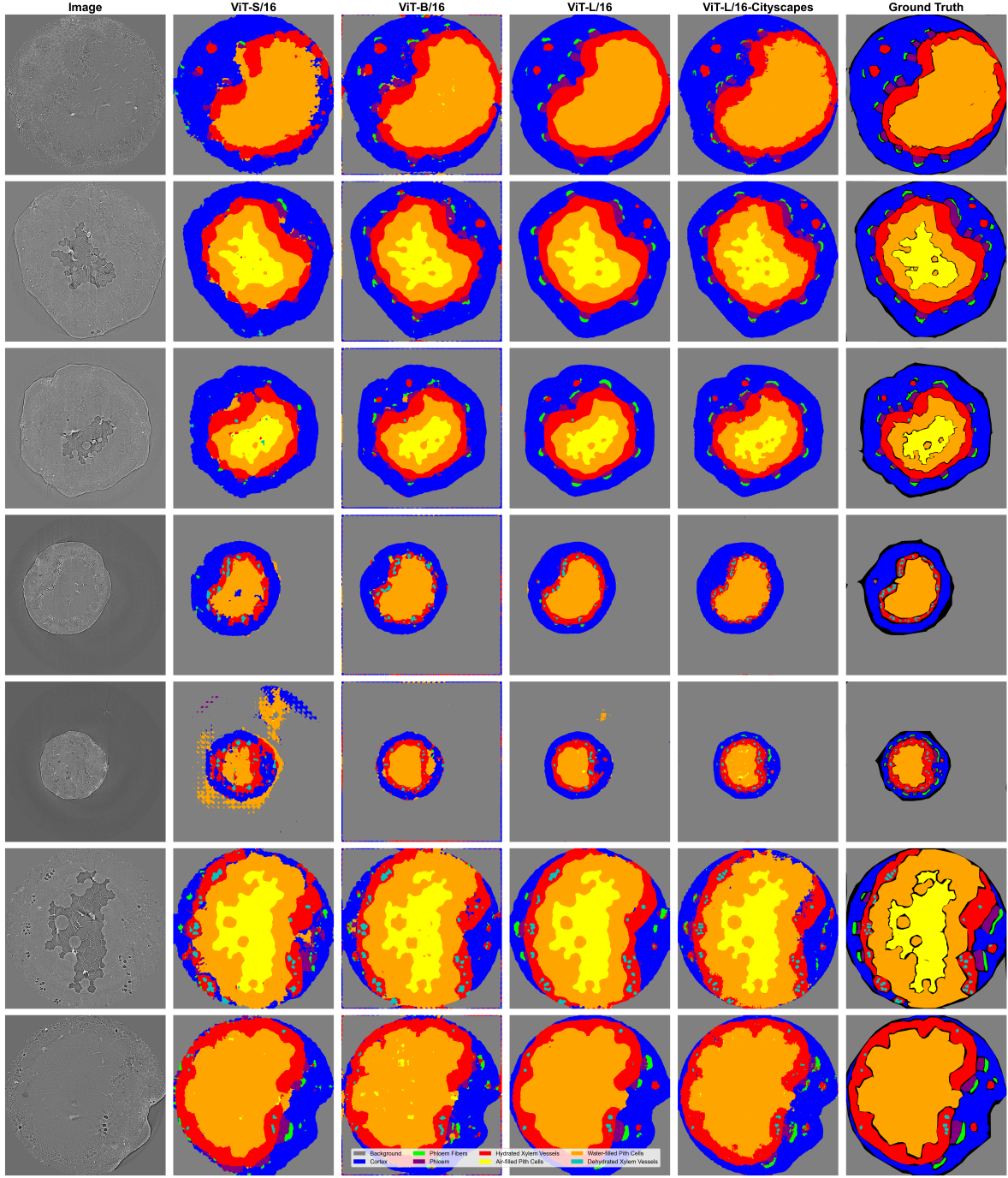


Figure 4: Comparison of different DINOv3-Seg backbones fine-tuned on our dataset and evaluated on the test set. This semantic segmentation task contains seven classes, where hydrated and dehydrated xylem vessels are predicted as separate classes. Some images may not contain dehydrated vessels (e.g., the first three rows). The evaluated backbones include ViT-S/16, ViT-B/16, and ViT-L/16. The ViT-L/16-Cityscapes model is first fine-tuned on the Cityscapes dataset and then further fine-tuned on our dataset.

Table 1: Comparison results for semantic segmentation on petiole micro-CT datasets with different numbers of classes. Mean Intersection over Union (mIoU) is reported (higher is better).

#cls	Model	mIoU
6	Mask2Former [5]	0.0491
	EoMT [6]	0.0000
	Cellpose-SAM	–
	SAM3	0.0719
	DINOv3-Seg (fine-tuned on our data)	0.8085
	DINOv3-Seg (fine-tuned on Cityscapes + our data)	0.8189
7	Mask2Former	0.0588
	EoMT	0.0000
	Cellpose-SAM	–
	SAM3	0.0646
	DINOv3-Seg (fine-tuned on our data)	0.7739
	DINOv3-Seg (fine-tuned on Cityscapes + our data)	0.8033

Table 2: Wall-clock runtime (seconds) for each pipeline stage as a function of the number of NERSC Perlmutter GPU nodes. Mean and standard deviation are reported across n replicates per configuration; ‘–’ indicates a single sample (no std available).

Stage	#nodes	Mean (s)	Std (s)	Min (s)	n
Reconstruction	1	109.5	2.6	106	4
	2	62.8	4.9	60	4
	4	34.5	0.6	34	4
	8	21.3	0.5	21	4
	16	19.5	3.0	18	4
DINOv3-Seg	1	274.5	116.5	184	4
	2	220.5	84.4	175	4
	4	212.6	85.5	149	7
	8	187.5	12.8	176	4
SAM3	1	3530.0	–	3530	1
	2	2173.0	457.1	1880	4
	8	672.8	127.1	586	4
	16	533.8	203.1	420	4
	24	418.0	51.3	385	4
	32	357.3	53.8	277	4
	40	403.5	7.1	396	4
	46	413.5	3.3	410	4
Combine Segmentations	1	88.3	10.5	82	4
	4	82.8	1.3	81	4
	8	82.8	1.0	82	4
	16	84.5	1.3	83	4

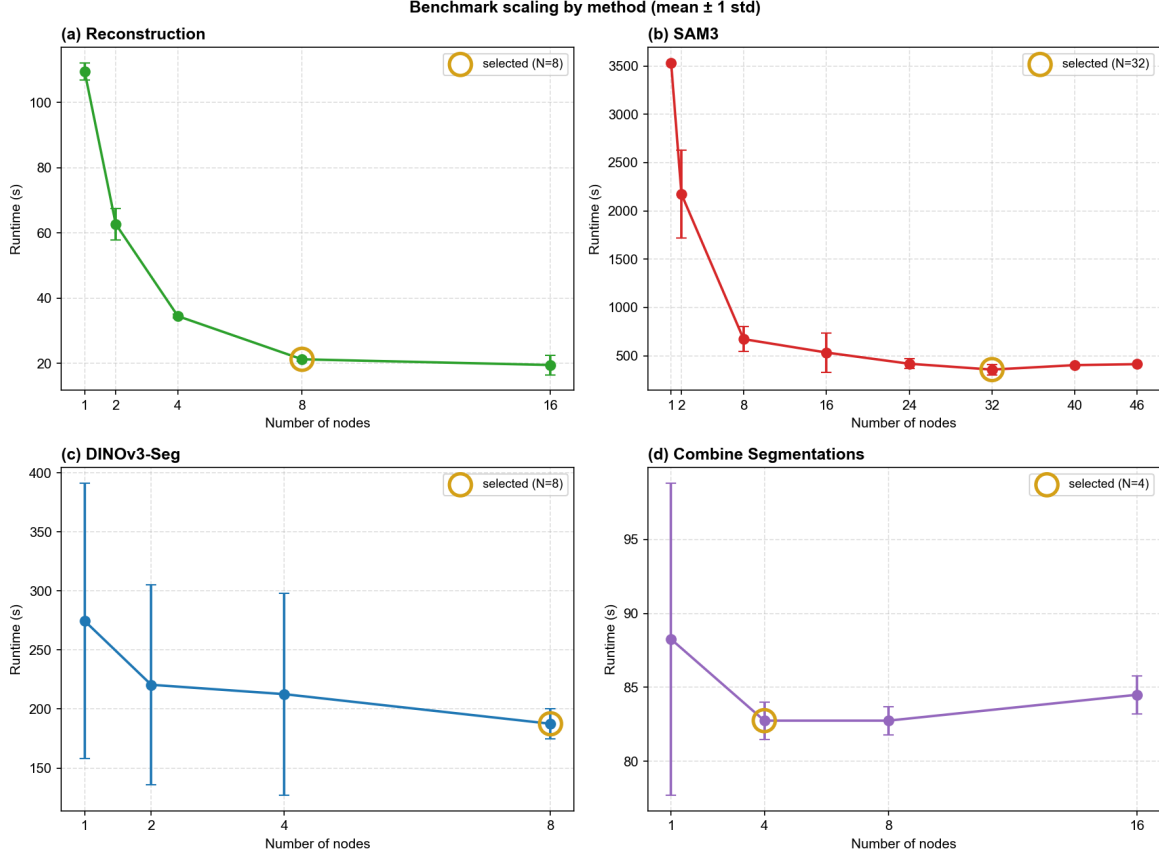


Figure 5: Strong-scaling behavior of pipeline stages on NERSC Perlmutter GPU nodes. Wall-clock runtime (mean $\pm$ 1 SD) versus the number of allocated GPU nodes for the four pipeline stages applied to a 9.36 GB micro-CT dataset (20260221_135217_petiole22.h5): (a) reconstruction, (b) SAM3 segmentation, (c) DINOv3-Seg, and (d) combination. Each configuration was run with $n=4$ replicates unless otherwise noted: SAM3 at 1 node was run once (single-replicate point shown without an error bar) due to its 1 hour wall time, and DINOv3-Seg at 4 nodes was run 7 times to better characterize its high run-to-run variance. Reconstruction scales well through 8 nodes with marginal further improvement at 16 nodes. SAM3 shows the largest absolute speedup between 1 and 8 nodes and continues to improve through 32 nodes, beyond which runtime plateaus and slightly increases. DINOv3-Seg scales through 8 nodes; because DINOv3-Seg and SAM3 run in parallel in the pipeline, DINOv3-Seg is not rate-limiting and was not tested at higher node counts. The combine stage shows no meaningful scaling; we adopted 4 nodes for downstream runs as it offered slightly lower mean runtime and lower variance than 1 node. For (a), (b), and (c), the reported times are extracted from SLURM job output and exclude Prefect overhead, job-submission latency, and queue wait time. For (d), this timing includes submission overhead.

Table 3: Wall-clock runtime per pipeline stage at NERSC Perlmutter and ALCF Polaris, pooled across 15 grapevine petiole datasets (file sizes 1.6–15 GB) with $n = 4$ replicates per dataset. Times reported as mean $\pm$ standard deviation in minutes. Each GPU node has 4 NVIDIA A100 GPUs. CPU nodes on ALCF Polaris have a single 64 core AMD EPYC “Milan” processor, and CPU nodes on NERSC Perlmutter have two 64 core AMD EPYC 7763 processors.

Stage	Facility	Resources	Type	Time (min)
Data Transfers	NERSC	ESNet/Globus	Transfer	0.3 ± 0.2 (n=16)
	ALCF	ESNet/Globus	Transfer	0.4 ± 0.3 (n=10)
Reconstruction	NERSC	16 CPU Nodes	Compute	0.3 ± 0.1 (n=60)
	ALCF	8 CPU Nodes	Compute	0.7 ± 0.4 (n=60)
SAM3	NERSC	32 GPU Nodes	Compute	7.2 ± 1.8 (n=60)
	ALCF	4 GPU Nodes	Compute	6.5 ± 4.6 (n=60)
DINOv3-Seg	NERSC	8 GPU Nodes	Compute	5.2 ± 1.6 (n=60)
	ALCF	4 GPU Nodes	Compute	2.2 ± 1.5 (n=60)
Combine	NERSC	4 CPU Nodes	Compute	1.5 ± 0.3 (n=60)
	ALCF	2 CPU Nodes	Compute	1.1 ± 0.7 (n=60)

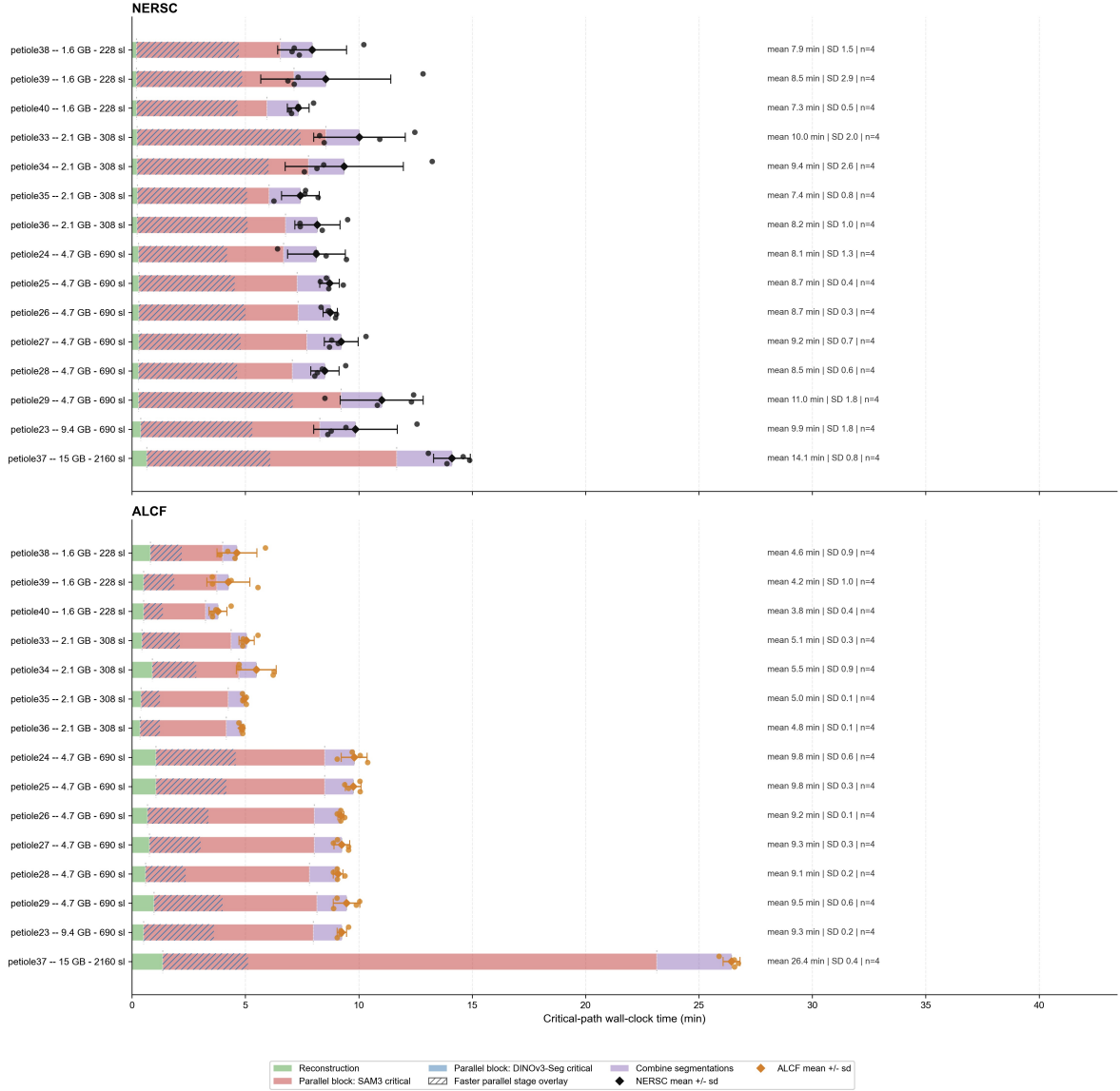


Figure 6: Per-dataset critical-path timing comparison for NERSC and ALCF. Stacked horizontal bars show the mean workflow composition for each grapevine petiole micro-CT dataset, grouped by facility. Each row corresponds to one dataset, annotated by scan identifier, raw file size, and number of reconstructed slices. The NERSC configuration used 16 CPU nodes for reconstruction, 32 GPU nodes for SAM3, 8 GPU nodes for DINOv3-Seg, and 4 CPU nodes for the final combine step. The ALCF configuration used 8 CPU nodes for reconstruction, 4 GPU nodes for SAM3, 4 GPU nodes for DINOv3-Seg, and 2 CPU nodes for the final combine step. Bars decompose the critical path into reconstruction, the slower of the two parallel segmentation stages, and final mask combination; the faster parallel segmentation branch is shown as a hatched overlay. Points and error bars indicate the mean critical-path wall-clock time ± 1 SD across replicate runs, with individual replicate values shown as small points. Right-side text reports the per-dataset mean, standard deviation, and number of replicates. The comparison illustrates that ALCF provides shorter runtimes for several smaller datasets under the tested configuration, whereas the larger NERSC allocation improves scaling for the largest dataset.

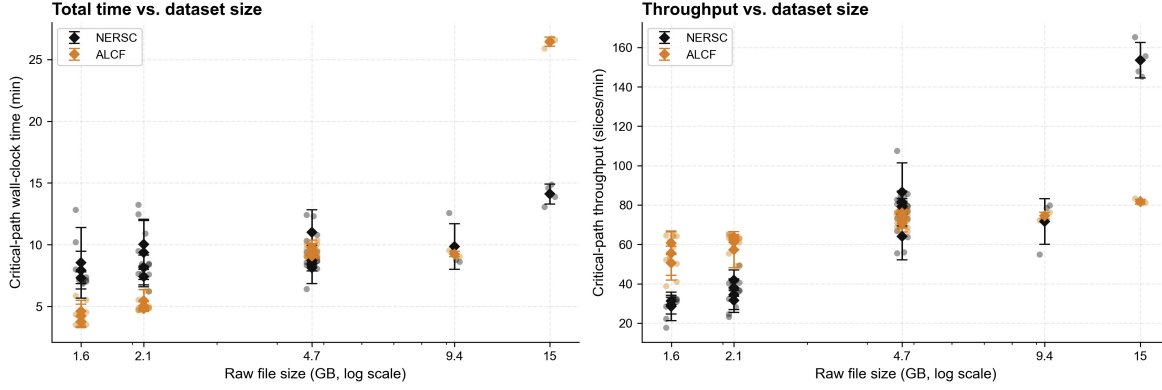


Figure 7: Critical-path runtime and throughput as a function of raw file size at NERSC and ALCF. Left: total critical-path wall-clock time for each grapevine petiole micro-CT dataset, grouped by raw file size. Right: corresponding critical-path throughput, computed as reconstructed slices processed per minute. Points show individual replicate runs, while diamonds and error bars indicate the mean ± 1 SD for each facility and file-size group. The NERSC configuration used 16 CPU nodes for reconstruction, 32 GPU nodes for SAM3, 8 GPU nodes for DINOv3-Seg, and 4 CPU nodes for the final combine step. The ALCF configuration used 8 CPU nodes for reconstruction, 4 GPU nodes for SAM3, 4 GPU nodes for DINOv3-Seg, and 2 CPU nodes for the final combine step. ALCF achieves shorter critical-path runtimes for several smaller datasets under the tested configuration, whereas NERSC shows higher throughput and better scaling for the largest dataset.

References

- [1] Marius Pachitariu, Michael Rariden, and Carsen Stringer. Cellpose-sam: Superhuman generalization for cellular segmentation. *bioRxiv*, 2025.
- [2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.
- [3] Oriane Simeoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michael Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.
- [4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [6] Tommie Keressies, Niccolo Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan De Geus. Your vit is secretly an image segmentation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25303–25313, 2025.