

Towards Foundation Models for Brain Microscopy Image Analysis



May 4, 2025

Jason Ken Adhinarta
Advised by Prof. Donglai Wei

Boston College
Department of Computer Science
Honors Thesis

Abstract

Neuroscience imaging data presents unique challenges due to complex geometric structures and limited data availability. Through three projects, we show how leveraging geometric priors, foundation models, and curated benchmarks are promising strategies for developing computer vision techniques for neuroscience. First, using the Frenet-Serret Frame-based Decomposition, we study how invariances to curvilinear transformations induce data-efficient learning on point cloud part segmentation tasks. To verify our findings, we create CurviSeg, a synthetic dataset of 3D curvilinear structures, we curate DenSpineEM, comprised of 70 dendrites sourced from public electron microscopy datasets, and we evaluate the generalizability of our method on the IntraA intracranial aneurysm segmentation dataset. Secondly, we propose TriSAM, adapting the Segment Anything Model for 3D segmentation of blood vessels using tri-plane seed tracking. We demonstrate the effectiveness of our model on electron microscopy datasets sourced from mouse, macaque, and human cortical samples. Finally, we establish WormID-Bench, a benchmark for whole-brain neural activity extraction from *C. elegans*. We assess how recent detection, identification, and tracking models perform, with the eventual goal of promoting progress in reverse engineering the nervous system of the nematode. We hope that this body of work advances understanding in neuroscience through the lens of machine learning methodology.

Acknowledgements

Much credit is due

to Prof. Donglai Wei, who challenged me to ask the questions that *really matter*,
to Shixuan Gu, who made it his mission to get me to *a certain place*,
to labmates Adam Gohain, Micaela Roth, Michael Lin, who pulled me through *C. elegans*,
and not least to family and friends that made this endeavor possible.

To think that all this started with a pink Indonesian bill in St. Mary's South before the
invention of Schiller cubicles; *how inscrutable his ways!*

Contents

1	Frenet-Serret Frame-based Decomposition for Part Segmentation of 3D Curvilinear Structures	5
1.1	Introduction	6
1.2	Related Works	8
1.2.1	Applications in Medical Imaging	8
1.2.2	Methodological Foundations	8
1.3	Frenet-Serret Frame-based Decomposition	9
1.3.1	Method Overview	9
1.3.2	Formulation of Frenet-Serret Frame-based Decomposition	11
1.3.3	Properties of the Decomposition	12
1.3.4	Implementation	13
1.4	Datasets	14
1.4.1	CurviSeg Dataset	14
1.4.2	DenSpineEM Dataset	15
1.4.3	Intra Dataset	16
1.5	Experiments and Results	16
1.5.1	Property Validation with CurviSeg Dataset	16
1.5.2	Benchmark on Dendritic Spine Segmentation	18
1.5.3	Additional Evaluation on Intracranial Aneurysm Segmentation	20
1.6	Conclusion	20
2	TriSAM: Tri-Plane SAM for zero-shot cortical blood vessel segmentation in VEM images	21
2.1	Introduction	21
2.2	Related Work	23
2.3	BvEM Dataset	24
2.3.1	Dataset Description	24
2.3.2	Dataset Annotation	24

2.3.3	Dataset Statistics	25
2.4	Method	25
2.4.1	Overview	25
2.4.2	TriSAM Framework	26
2.4.3	Implementation Details	29
2.5	Experiments	29
2.5.1	Experimental Settings	29
2.5.2	Benchmark Results	29
2.5.3	Ablation Studies	31
2.5.4	Failure Cases	34
2.6	Conclusion	34
3	WormID-Bench: A Benchmark for Whole-Brain Activity Extraction in <i>C. elegans</i>	36
3.1	Introduction	36
3.2	Related Work	37
3.3	Dataset	38
3.4	Benchmark	39
3.4.1	Task 1: 3D Neuron Detection in 3D NeuroPAL Volumes	39
3.4.2	Task 2: 3D Neuron Identification in 3D NeuroPAL Volumes	39
3.4.3	Task 3: 3D Neuron Tracking in 3D Calcium Videos	40
3.5	Results	40
3.5.1	Task 1: 3D Neuron Detection in 3D NeuroPAL Volumes	41
3.5.2	Task 2: 3D Neuron Identification in 3D NeuroPAL Volumes	41
3.5.3	Task 3: 3D Neuron Tracking in 3D Calcium Videos	42
3.6	Limitation and Discussion	43

Chapter 1

Frenet-Serret Frame-based Decomposition for Part Segmentation of 3D Curvilinear Structures

*This section draws extensively from previous work done in collaboration with the Harvard Visual Computing Group and Lichtman Lab: Leslie Gu et al. “Frenet-Serret Frame-based Decomposition for Part Segmentation of 3D Curvilinear Structures”. In: *arXiv preprint arXiv:2404.14435* (2024).*

Accurately segmenting 3D curvilinear structures in medical imaging remains challenging due to their complex geometry and the scarcity of diverse, large-scale datasets for algorithm development and evaluation. In this paper, we use dendritic spine segmentation as a case study and address these challenges by introducing a novel Frenet–Serret Frame-based Decomposition, which decomposes 3D curvilinear structures into a globally smooth continuous curve that captures the overall shape, and a cylindrical primitive that encodes local geometric properties. This approach leverages Frenet–Serret Frames and arc length parameterization to preserve essential geometric features while reducing representational complexity, facilitating data-efficient learning, improved segmentation accuracy, and generalization on 3D curvilinear structures. To rigorously evaluate our method, we introduce two datasets: *CurviSeg*, a synthetic dataset for 3D curvilinear structure segmentation that validates our method’s key properties, and *DenSpineEM*, a benchmark for dendritic spine segmentation, which comprises 4,476 manually annotated spines from 70 dendrites across three public electron microscopy datasets, covering multiple brain regions and species. Our experiments on *DenSpineEM* demonstrate exceptional cross-region and cross-species generalization: models trained on the mouse somatosensory cortex subset achieve 91.9% Dice, maintaining strong performance in zero-shot segmentation on both mouse visual cortex (94.1% Dice) and human frontal lobe (81.8% Dice) subsets. Moreover, we test the generalizability of our method on the *IntraA* dataset, where it achieves 77.08% Dice (5.29% higher than prior arts) on intracranial aneurysm segmentation. These findings demonstrate the potential of our approach for accurately analyzing complex curvilinear structures across diverse medical imaging fields. Our dataset, code, and models are available at <https://github.com/VCG/FFD4DenSpineEM> to support future research.

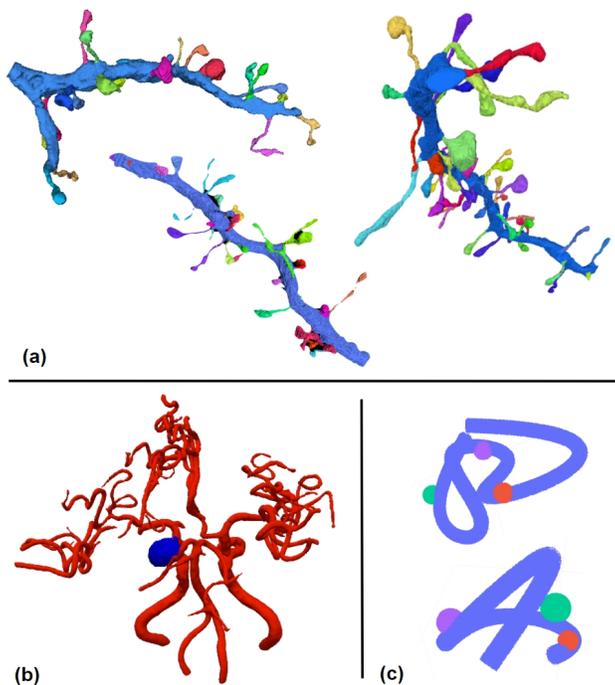


Figure 1.1: **Part Segmentation for 3D Curvilinear Structures.** Curvilinear structures from (a) DenSpineEM: Main experimental dataset on dendritic spine segmentation. (b) IntraA: Intracranial aneurysm segmentation dataset for testing cross-domain generalizability. (c) CurviSeg: Synthetic dataset for theoretical validation. Colors indicate segmentation labels.

1.1 Introduction

Deep learning-enabled 3D biomedical imaging has driven advancements in both scientific research (*e.g.*, connectomics [2, 3], protein structure prediction [4, 5]) and as a crucial tool in medical care (*e.g.*, bone lesion analysis [6, 7, 8], aneurysm detection [9]). While semantic segmentation algorithms, such as nn-UNet [10], have achieved strong results in various tasks, the segmentation of 3D curvilinear structures remains challenging due to their intricate geometry, varying thickness, and complex branching patterns [11]. These structures, characterized by their elongated, often branching nature following curved paths in three-dimensional space, are ubiquitous in biological and medical imaging, playing crucial roles in various systems from neuronal networks to vascular systems [12].

In this paper, we focus on dendritic spine segmentation as a representative task for 3D curvilinear structure analysis. Dendritic spines, small protrusions on neuronal dendrites, are crucial for synaptic transmission, and their morphology and density provide vital information about neuronal connectivity, making accurate segmentation essential for neuroscience research [12]. However, segmentation is challenging due to spines’ high density along dendrites, complex geometry, variable sizes and shapes, and intricate branching patterns [11]. The lack of benchmark datasets has led to reliance on simple heuristics without human-annotated comparisons, limiting the reliability of current methods.

Recent advances, such as deep learning-based workflows [13], joint classification and segmentation methods for 2-photon microscopy images [14], and interactive tools like 3dSpAn [15],

have improved performance. However, these approaches often require large training datasets or manual refinement and struggle to generalize across different imaging conditions and spine morphologies. This underscores the need for more data-efficient methods capable of handling the complexity of 3D curvilinear structures.

To address these challenges, we propose the *Frenet–Serret Frame-based Decomposition (FFD)*, which decomposes 3D curvilinear geometries into two components: a globally smooth C^2 continuous curve that captures the overall shape, and a cylindrical primitive that encodes local geometric properties. This approach leverages Frenet–Serret Frames and arc length parameterization to preserve essential geometric features while reducing representational complexity. The resultant cylindrical representation facilitates data-efficient learning, improved segmentation accuracy, and generalization on 3D curvilinear structures.

To validate the effectiveness of our approach, we introduce *CurviSeg*, a synthetic dataset for segmentation tasks of 3D curvilinear structures, which serves as a theoretical validation to verify the key properties of our method. Additionally, we present *DenSpineEM*, a benchmark dataset for dendritic spine segmentation, consisting of 4,476 manually annotated dendritic spines from 70 dendrites across three 3D electron microscopy (EM) image stacks (mouse somatosensory cortex, mouse visual cortex, and human frontal lobe). Using our decomposition, models trained on the large subset from the mouse somatosensory cortex achieve high segmentation performance (91.9% Dice) and demonstrate strong zero-shot generalization on both the mouse visual cortex (94.1% Dice) and human frontal lobe (81.8% Dice) subsets. Moreover, we demonstrate the generalizability of our method on the *Intra* dataset for intracranial aneurysm segmentation, where it achieves 77.08% DSC, outperforming the state-of-the-art by 5.29%, highlighting its effectiveness beyond dendritic spine segmentation to other medical imaging tasks.

Our contributions include:

- We propose the Frenet–Serret Frame-based Decomposition, decomposing 3D curvilinear geometries into a smooth C^2 curve and cylindrical primitive for efficient learning and robust segmentation.
- We develop *DenSpineEM*, a comprehensive benchmark for 3D dendritic spine segmentation, containing 4,476 manually annotated spines from 70 dendrites across three EM datasets, covering various brain regions and species.
- We introduce *CurviSeg*, a synthetic dataset for 3D curvilinear structure segmentation, used to validate our method and as a resource for other analyses.
- Our method achieves high segmentation accuracy with cross-species and cross-region generalization on dendritic spine segmentation, and surpasses state-of-the-art methods on intracranial aneurysm segmentation.

1.2 Related Works

1.2.1 Applications in Medical Imaging

3D Curvilinear Structure Analysis. In the medical domain, curvilinear structures are prevalent and critical, with applications spanning blood vessel segmentation [16], neuronal tracing [17], and airway tree extraction [18]. These structures, characterized by their tubular or filament-like shape, present unique challenges due to their complex geometry and intricate branching patterns. Traditional methods rely on hand-crafted features, such as the Hessian-based Frangi vesselness filter [19] and multi-scale line filter [20], which enhance tubular structures but often struggle with complex geometries and varying scales.

Recent advancements leverage machine learning techniques to improve robustness and accuracy. Sironi et al. [21] introduced a multi-scale regression approach for centerline detection, while deep learning methods (e.g., nnU-Net [10] and DeepVesselNet [22]) have shown superior performance in vessel segmentation tasks. Despite these advances, challenges persist in the medical domain, including high variability in structure appearance, resolution limitations, and the scarcity of large-scale annotated datasets [23]. Our work builds upon these foundations, using dendritic spine segmentation as a compelling example to address these challenges through our novel Frenet frame-based transformation.

Dendritic Spine Segmentation. Dendrites, with their curvy and elongated structure, serve as an excellent example for curvilinear structure analysis. Their protrusions, known as dendritic spines, play a crucial role in neuronal connectivity and plasticity [24]. The segmentation of these spines presents unique challenges across different imaging modalities. In light microscopy, where spines appear as tiny blobs due to limited resolution, research has focused on spine location detection [25], semi-automatic segmentation [26], and morphological analysis [27]. High-resolution electron microscopy (EM) has enabled more precise spine analysis, leading to two main approaches: morphological operations with watershed propagation [28], and skeletonization with radius-based classification [29]. However, these methods often rely on hand-tuned hyperparameters and require all voxels as input, limiting their effectiveness for large-scale data analysis. The field of dendritic spine segmentation faces two significant challenges: the lack of comprehensive benchmark datasets for rigorous evaluation, and the need for effective methods that can handle complex spine geometry in large-scale datasets. To address these challenges, we introduce both a large-scale 3D dendritic spine segmentation benchmark and a novel Frenet frame-based transformation method, potentially advancing curvilinear structure analysis in neuroscience and beyond.

1.2.2 Methodological Foundations

Preliminaries on Frenet-Serret Frame. To understand the geometric properties of curvilinear structures, we turn to the fundamental concept of the Frenet-Serret frame in differential geometry. In three-dimensional Euclidean space \mathbb{R}^3 , the Frenet-Serret frame (TNB frame) of a differentiable curve at a point is a triplet of three mutually orthogonal unit vectors (i.e., tangent, normal, and binormal) [30]. Specifically, let $\mathbf{r}(s)$ be a curve in Euclidean space

parameterized by arc length \mathbf{s} , then the Frenet-Serret frame can be defined by:

$$\mathbf{T} := \frac{d\mathbf{r}}{ds}, \quad \mathbf{N} := \frac{d\mathbf{T}}{ds} / \left\| \frac{d\mathbf{T}}{ds} \right\|, \quad \mathbf{B} := \mathbf{T} \times \mathbf{N}, \quad (1.1)$$

which satisfies the Frenet-Serret formulas:

$$\frac{d\mathbf{T}}{ds} = \kappa\mathbf{N}, \quad \frac{d\mathbf{N}}{ds} = -\kappa\mathbf{T} + \tau\mathbf{B}, \quad \frac{d\mathbf{B}}{ds} = -\tau\mathbf{N}, \quad (1.2)$$

where $\kappa(s)$ is curvature and $\tau(s)$ is torsion, measuring how sharply the curve bends and how much the curve twists out of a plane.

Originally formulated for physics applications [31], Frenet-Serret Frame has subsequently been adopted across diverse domains. In robotics and autonomous driving, it facilitates the optimization of trajectory planning [32]. The computer graphics community utilizes it for generating swept surface models [33], rendering streamline visualizations [34], and computing tool paths in CAD/CAM systems [35]. More recently, Frenet frame has been instrumental in characterizing protein structures in bioinformatics [36], underscoring their adaptability across varying scales and scientific disciplines. Our work extends this concept to the (bio)medical domain, specifically for the analysis and segmentation of dendritic spines, where we employ it to map these 3D curvilinear structures onto a standardized cylindrical coordinate system while preserving crucial geometric properties.

Computational Approaches for 3D Medical Imaging. 3D shapes in biomedical imaging, typically derived from CT (Computational Tomography) and EM (Electron Microscopy) scans, are often represented as voxels on discrete grids. Prior works [37, 38] predominantly use voxel representations, extending 2D approaches to 3D (*e.g.*, 3D UNet [39]) or employing sophisticated 3D operators [40]. However, voxel-based methods face challenges with high memory requirements and limited spatial resolution. Alternatively, point cloud representations offer a lightweight and flexible approach for 3D shape analysis [41]. They excel in extracting semantic information [42] and provide higher computational efficiency for large-scale objects. Given these advantages, our work primarily utilizes point cloud representations for analyzing 3D curvilinear structures.

1.3 Frenet–Serret Frame-based Decomposition

1.3.1 Method Overview

Intuition. Our intuition is based on the observation that curvilinear structures in biological systems often exhibit tree-like morphologies, with complexity arising from two main aspects:

- Global structure: The overall shape and orientation of the main structure, such as the elongation and curvature of a dendrite trunk or blood vessels.
- Local geometry: Smaller, often critical elements attached to or variations along the main structure, such as dendritic spines or vascular bifurcations.

For segmentation tasks, the global structure adds unnecessary complexity, expanding the learning space and increasing data requirements. Our approach separates these components

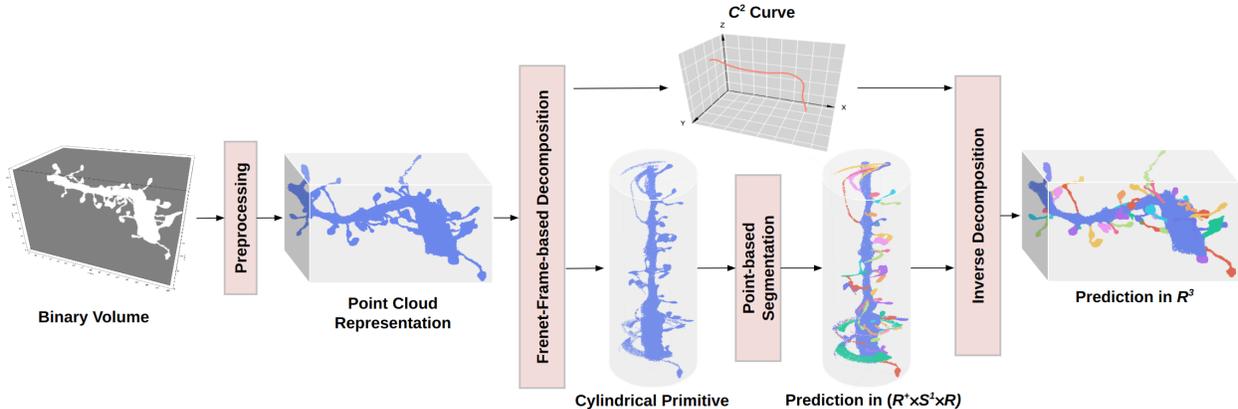


Figure 1.2: **Exemplary Pipeline of Dendritic Spine Segmentation using Frenet-Serret Frame-based Decomposition.** The pipeline consists of three main steps: 1) **Decomposition:** Converting dense volumes to point clouds in \mathbb{R}^3 , then decomposing into a C^2 curve and a cylindrical primitive ($\mathbb{R}^+ \times S^1 \times \mathbb{R}$). 2) **Segmentation:** Performing point-based segmentation on the cylindrical primitive, leveraging the simplified geometry for improved accuracy and efficiency. 3) **Inverse Decomposition:** Reconstructing the segmented structure back to \mathbb{R}^3 by combining the cylindrical primitive with the C^2 curve.

by decomposing the structure into standardized representations. Such decomposition enables efficient learning through standardized cylindrical representations that preserve intrinsic shape information while reducing global variations.

Segmentation Pipeline with FFD. We use dendritic spine segmentation as an exemplar to demonstrate the application of Frenet-Serret Frame-based Decomposition (FFD) for segmenting 3D curvilinear structures. As illustrated in Fig. 1.2, our pipeline consists of three main stages:

- *Decomposition:* Initially, binary EM volumes are converted to point clouds in \mathbb{R}^3 . We then perform skeletonization with topological pruning to extract the backbone (dendrite trunk) skeleton, parametrizing it as a C^2 continuous curve. Along this curve, we calculate Frenet-Serret Frames and reconstruct surrounding point clouds in a cylindrical coordinate system (Fig. 1.3). This forms a cylindrical primitive in $(\mathbb{R}^+ \times S^1 \times \mathbb{R})$, preserving essential local geometries.
- *Segmentation:* With its reduced learning space, the cylindrical primitive undergoes data-efficient segmentation, as well as enabling improved generalization across diverse samples.
- *Inverse Decomposition:* Finally, we transform the segmented cylindrical primitive and C^2 curve back to the original \mathbb{R}^3 space, completing the process.

This approach significantly boosts segmentation accuracy and generalization performance on dendritic spine segmentation task, as demonstrated in our experiments (Sec.1.5.2). In the following subsections, we provide the mathematical formulation of the decomposition (Sec.1.3.2), prove its properties (bijectivity and rotation-invariance, Sec.1.3.3), and detail the

implementation of the pipeline, including skeletonization and discrete Frenet-Serret Frame calculation (Sec.1.3.4).

1.3.2 Formulation of Frenet-Serret Frame-based Decomposition

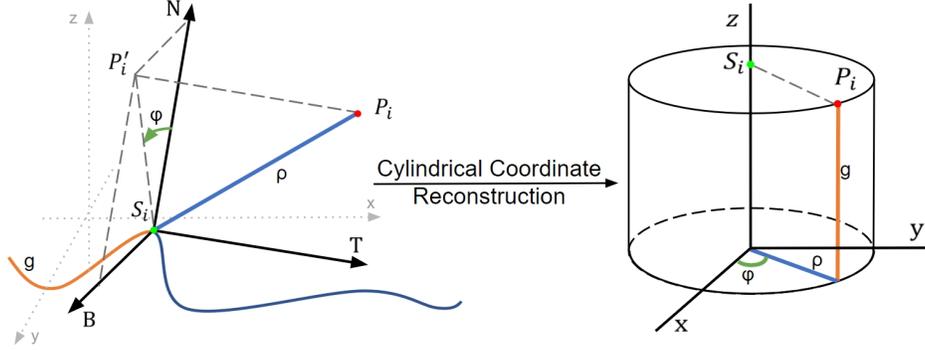


Figure 1.3: **Frenet-Serret Frame-based Transformation.** A key component of FFD is the transformation that maps point clouds in \mathbb{R}^3 to cylindrical coordinates (ρ, ϕ, g) in $(\mathbb{R}^+ \times S^1 \times \mathbb{R})$. It utilizes the Frenet-Serret Frame (T, N, B) of the curve at S_i , the nearest point to P_i . **Left:** The point P_i and curve S in a Cartesian coordinate system. ρ : distance between S_i and P_i , ϕ : angle between normal vector N and projection of $\overrightarrow{S_i P_i}$ onto NB -plane, g : curve arc length to S_i . **Right:** The reconstruction of P_i in a cylindrical coordinate system.

Denote $P = \{(x_i, y_i, z_i) \mid i = 1, \dots, n\} \subset \mathbb{R}^3$ as a point cloud, C as the space of C^2 curves in \mathbb{R}^3 that form the backbone skeleton of P . We formulate the decomposition:

$$\mathcal{D} : P \rightarrow C \times (\mathbb{R}^+ \times S^1 \times \mathbb{R})^n, \quad (1.3)$$

as a composition of two mappings: $\mathcal{D} = \mathcal{S} \circ \mathcal{F}$, where:

- $\mathcal{S} : P \rightarrow C$ is a skeletonization function that maps the point cloud to a C^2 continuous curve $S : [0, L] \rightarrow \mathbb{R}^3$, parameterized by arc length $s \in [0, L]$.
- $\mathcal{F} : C \times P \rightarrow C \times (\mathbb{R}^+ \times S^1 \times \mathbb{R})^n$ is a Frenet-Serret Frame-based transformation that reconstruct the point cloud in cylindrical coordinates, defined as:

$$\mathcal{F}(S, P) = (S, \{(\rho_i, \phi_i, g_i) \mid i = 1, \dots, n\}), \quad (1.4)$$

where $\{(\rho_i, \phi_i, g_i) \mid i = 1, \dots, n\}$ is the reconstructed point cloud in a cylindrical coordinate system.

Specifically, as depicted in Fig. 1.3, for each point P_i , we determine its closest point on the curve, $S_i = S(s_i)$, where $s_i = \min_{s \in [0, L]} \|P_i - S(s)\|$. Due to the continuity of S , the closest point is unique for almost all P_i ¹. The transformation is then defined as:

$$\rho_i = \|P_i - S_i\|, \quad g_i = \int_0^{s_i} \left\| \frac{dS(s)}{ds} \right\| ds = s_i,$$

¹For a continuous curve, almost every point in \mathbb{R}^3 has a unique closest point on the curve. The set of points with multiple equally closest points (*i.e.*, cut locus) is of measure zero and does not affect the overall transformation.

$$\phi_i = \arctan 2(\mathbf{v}_i \cdot \mathbf{b}_{s_i}, \mathbf{v}_i \cdot \mathbf{n}_{s_i}),$$

where \mathbf{v}_i represents the projection of the vector $\overrightarrow{S_i P_i}$ (denoted as \mathbf{u}_i) onto the normal-binormal plane, which can be calculated by $\mathbf{v}_i = A_i A_i^T \mathbf{u}_i$, where $A_i = [\mathbf{n}_{s_i}, \mathbf{b}_{s_i}]$ is a column orthogonal matrix.

1.3.3 Properties of the Decomposition

Properties. The Frenet–Serret Frame-based Decomposition possesses two key properties: 1) *Bijectivity*: The decomposition is invertible, allowing the cylindrical primitive and backbone curve to be transformed back to the original space without information loss. 2) *Rotation Invariance*: The decomposition is invariant to rotations of the input data, as the cylindrical primitive is constructed in a standardized coordinate system aligned with the backbone curve.

Benefits. These properties confer the following benefits: 1) Bijectivity enables segmentation to be performed in the simplified cylindrical space while preserving the ability to map results back to the original space accurately. 2) Rotation invariance eliminates the need for rotation augmentation and ensures consistent feature representation regardless of the input orientation.

Proof. To prove the properties of the decomposition \mathcal{D} , it suffices to prove the corresponding properties of \mathcal{F} . Given that $\mathcal{S} : P \rightarrow C$ is a fixed mapping for a given point cloud, the properties of $\mathcal{D} = \mathcal{F} \circ \mathcal{S}$ are fundamentally determined by $\mathcal{F} : C \times P \rightarrow C \times (\mathbb{R}^+ \times S^1 \times \mathbb{R})^n$. Therefore, we focus the proof on the Frenet-Serret Frame-based transformation \mathcal{F} . For notational convenience, we use $\mathcal{F}(P)$ to represent $\mathcal{F}(S, P)$ in our proofs, as S is fixed for a given input.

1) *Bijectivity*. To prove the transformation is bijective, we need to verify that it's both injective and surjective.

- **Injectivity:** Assume $P_{t_1}, P_{t_2} \in P$, with $\mathcal{F}(P_{t_1}) = \mathcal{F}(P_{t_2}) = (\rho, \phi, g)$. Let $S_t \in \mathbb{R}^3$ be their closest point on the skeleton S . If $P_{t_1} \neq P_{t_2}$, then $\overrightarrow{P_{t_1} P_{t_2}} = \delta \mathbf{t}$, $\delta \neq 0$, where \mathbf{t} is the tangent at S_t . As S is C^2 continuous, $\exists \epsilon > 0$ sufficiently small and $S'_t \in S$ such that $\overrightarrow{S'_t S_t} = \epsilon \mathbf{t}$ and $\|\overrightarrow{S'_t P_{t_1}}\|^2 = \|\overrightarrow{S_t P_{t_1}}\|^2 - \epsilon^2 + o(\epsilon^2)$. Hence $d(P_{t_1}, S'_t) < d(P_{t_1}, S_t)$, contradicting that S_t is the closest point to P_{t_1} on S . Hence, $\forall P_{t_1}, P_{t_2} \in P$ such that $\mathcal{F}(P_{t_1}) = \mathcal{F}(P_{t_2})$, we have $P_{t_1} = P_{t_2}$, *i.e.*, the transformation is injective.
- **Surjectivity:** As S is C^2 continuous, $\forall s_1, s_2 \in [0, L]$ ($s_1 \neq s_2$), we have $\|S_{s_1} - S_{s_2}\| > 0$. Hence, $\forall Y_t = (\rho_t, \phi_t, g_t) \in \mathbb{R}^+ \times S^1 \times \mathbb{R}$, $S_{s_t} \in S$ can be uniquely determined by $g_t = \int_0^{s_t} \|\frac{dS(s)}{ds}\| ds$. Denote the Frenet-Serret Frame at S_{s_t} as $(\mathbf{t}_{s_t}, \mathbf{n}_{s_t}, \mathbf{b}_{s_t})$. $\exists \delta \in (0, \rho_t)$, we have $P_t \in \mathbb{R}^3$ as $\overrightarrow{S_{s_t} P_t} = \delta(\sin \phi_t \mathbf{b}_{s_t} + \cos \phi_t \mathbf{n}_{s_t}) + \sqrt{\rho_t^2 - \delta^2} \mathbf{t}_{s_t}$, such that $\mathcal{F}(P_t) = Y_t$. Hence, $\forall Y_t \in \mathbb{R}^+ \times S^1 \times \mathbb{R}$, $\exists P_t \in \mathbb{R}^3$ such that $\mathcal{F}(P_t) = Y_t$, *i.e.*, the transformation is surjective.

2) *Rotation Invariance*. We prove the rotation invariance of \mathcal{F} by showing $\mathcal{F}(R(P_t)) = \mathcal{F}(P_t)$ for any $P_t \in \mathbb{R}^3$ and $R \in SO(3)$.

Let S_{s_t} be the closest point on S to P_t , $(\mathbf{t}(s_t), \mathbf{n}(s_t), \mathbf{b}(s_t))$ the Frenet-Serret Frame of S at S_{s_t} , $\mathbf{u}_t = \overrightarrow{S_{s_t} P_t}$, $A_t = [\mathbf{n}(s_t), \mathbf{b}(s_t)]$, and $\mathbf{v}_t = A_t A_t^T \mathbf{u}_t$. Under rotation R , the Frenet-Serret

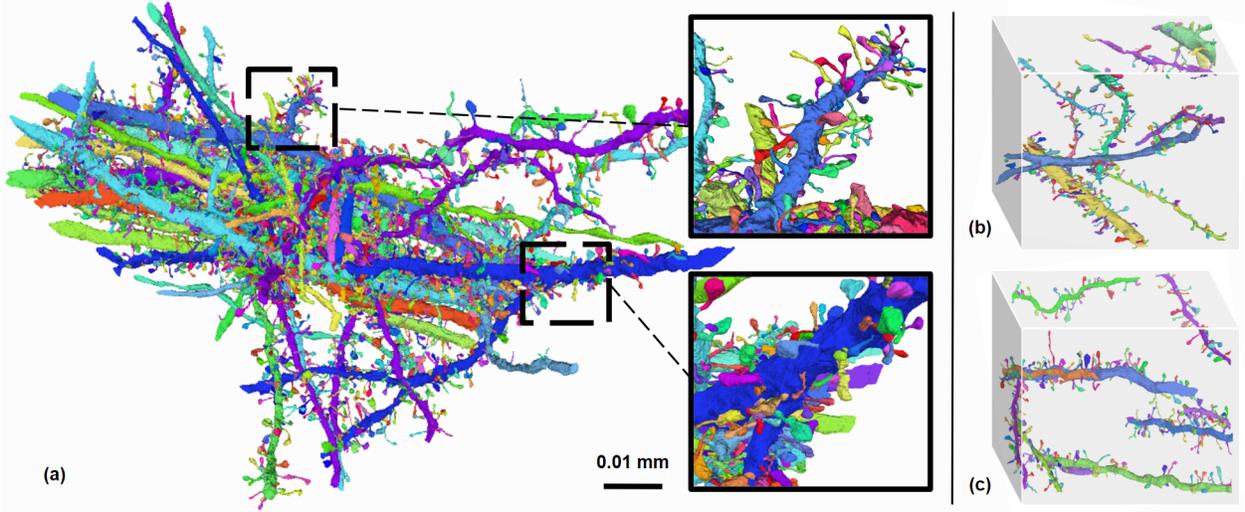


Figure 1.4: **DenSpineEM Dataset.** DenSpineEM contains 3 subsets: (a) M50: 50 mouse somatosensory cortex dendrites, (b) H10: 10 human visual cortex dendrites, (c) M10: 10 mouse visual cortex dendrites.

Frame rotates accordingly: $(\mathbf{t}'(s_t), \mathbf{n}'(s_t), \mathbf{b}'(s_t)) = (R\mathbf{t}(s_t), R\mathbf{n}(s_t), R\mathbf{b}(s_t))$, and $\mathbf{u}'_t = R(\mathbf{u}_t)$, $A'_t = R(A_t)$. Hence, $\mathbf{v}'_t = R(\mathbf{v}_t)$. Crucially:

$$\begin{aligned}
 (a) \quad \rho'_t &= \|R(P_t) - R(S_{s_t})\| = \|P_t - S_{s_t}\| = \rho_t \\
 (b) \quad g'_t &= g_t \\
 (c) \quad \phi'_t &= \arctan 2(R(\mathbf{v}_t) \cdot R(\mathbf{b}(s_t)), R(\mathbf{v}_t) \cdot R(\mathbf{n}(s_t))) \\
 &= \arctan 2(\mathbf{v}_t \cdot \mathbf{b}(s_t), \mathbf{v}_t \cdot \mathbf{n}(s_t)) = \phi_t
 \end{aligned}$$

(a), (b), and (c) hold because rotation preserves distances, arclength, and dot products, respectively. Thus, $\mathcal{F}(R(P_t)) = (\rho'_t, \phi'_t, g'_t) = (\rho_t, \phi_t, g_t) = \mathcal{F}(P_t)$, ensuring consistent transformation regardless of orientation.

1.3.4 Implementation

Backbone Skeletonization. We first apply the *Tree-structure Extraction Algorithm for Accurate and Robust Skeletons* (TEASAR) [43] to extract the initial skeleton from the input structure. TEASAR begins with a raster scan to locate an arbitrary foreground point, identifying its furthest point as the root. It then implements Euclidean distance transform to define a penalty field [44], guiding the skeleton through the target's center. Dijkstra's algorithm is applied to find the path from the root to the most geodesically distant point, forming a skeleton branch. Visited regions are marked by expanding a circumscribing cube around the path vertices. This process repeats until all points are traversed. Finally, the resultant skeleton is smoothed and upsampled via linear interpolation for density assurance. Although TEASAR effectively extracts initial skeletons, the intricate branching patterns of curvilinear biological structures can introduce unnecessary complexity into subsequent analyses. To simplify the topology, we prune minor branches from the extracted skeleton, preserving the main structure. We then traverse the simplified skeleton to identify individual

branches, cropping each as separate inputs while allowing overlap between adjacent branches to maintain continuity. Each cropped branch undergoes our transformation separately, facilitating focused processing of each branch within the overall structural context. Our pipeline is flexible for processing both volumetric and point cloud input; we use TEASAR in this study as it can be applied to both modalities with minor adjustments. Alternatively, we refer L1-medial skeletonization [45] as a robust approach for relatively small-scale point cloud data.

Discrete Frenet-Serret Frame Computation. We compute Frenet-Serret Frames along the curve to characterize local geometry, addressing both curved and straight segments. For curved segments, we apply standard Frenet-Serret formulas as defined in Eq. 1.1. To enhance numerical stability, we employ a curvature threshold $\epsilon = 1e - 8$, identifying near-straight segments where Frenet-Serret Frames become ill-defined. Our piecewise interpolation scheme handles straight segments effectively. Between curved parts, we linearly interpolate the normal vector, while at curve extremities, we propagate the normal from the nearest curved segment. For globally straight curves, we construct a single normal vector perpendicular to the tangent using the first non-collinear point and apply it consistently along the entire curve. To ensure frame orthonormality and further improve numerical stability, we apply Gram-Schmidt orthogonalization. Our Frenet-Serret Frame computation method is provided as a Python package², facilitating seamless integration into various geometric analysis and computational applications.

1.4 Datasets

1.4.1 CurviSeg Dataset

We introduce the CurviSeg dataset and make use of it for the first experiments in this paper. CurviSeg is defined as a synthetic dataset of 3D curvilinear structures with additional spherical objects for point cloud segmentation tasks. The curvilinear structures were generated using cubic B-spline interpolation of n randomly generated control points, where $n \sim \mathcal{U}\{5, 10\}$. The control points $\mathbf{p}_i \in \mathbb{R}^3$ were generated as:

$$\mathbf{p}_i = s \cdot \mathbf{r}_i, \quad i = 1, \dots, n \quad (1.5)$$

where $\mathbf{r}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ are random vectors sampled from a standard 3D normal distribution, and $s \sim \mathcal{U}(1, 3)$ is a uniform random scaling factor. The B-spline curve $\mathbf{C}(t)$ was then defined as:

$$\mathbf{C}(t) = \sum_{i=0}^{n-1} N_{i,3}(t) \mathbf{p}_i, \quad t \in [0, 1] \quad (1.6)$$

where $N_{i,3}(t)$ are cubic B-spline basis functions. This curve was evaluated at 500 equidistant points $\{t_j\}_{j=1}^{500}$ to form the skeleton. Points were distributed along this skeleton using a cylindrical coordinate system. For each skeleton point $\mathbf{C}(t_j)$, we generated a set of points $\mathbf{x}_{j,k}$ as:

$$\mathbf{x}_{j,k} = \mathbf{C}(t_j) + r \cos(\theta) \mathbf{n}_j + r \sin(\theta) \mathbf{b}_j \quad (1.7)$$

²<https://pypi.org/project/discrete-frenet-solver>

Table 1.1: **Overview of DenSpineEM Dataset.** We build upon 3 EM volumes with instance segmentation and annotate spine segmentation for 90 dendrites.

Name	Tissue	Size (μm^3)	#Dendrites	#Spines
DenSpine-M50	Mouse Somatosensory Cortex [47]	$50 \times 50 \times 30$	50	3,827
DenSpine-M10	Mouse Visual Cortex [48]	$30 \times 30 \times 30$	10	335
DenSpine-H10	Human Frontal Lobe [48]	$30 \times 30 \times 30$	10	319

where $r \sim \mathcal{U}(0, r_s)$, $r_s \sim \mathcal{U}(0.3, 0.7)$ is the slice radius, $\theta \sim \mathcal{U}(0, 2\pi)$, and \mathbf{n}_j and \mathbf{b}_j are the normal and binormal vectors at $\mathbf{C}(t_j)$, respectively. We added $m \sim \mathcal{U}\{1, 2, 3\}$ spherical objects to each structure. Each sphere, centered at \mathbf{c}_l , was placed tangent to a random point $\mathbf{x}_{j,k}$ on the main structure:

$$\mathbf{c}_l = \mathbf{x}_{j,k} + (r_s + r_b) \frac{\mathbf{x}_{j,k} - \mathbf{C}(t_j)}{\|\mathbf{x}_{j,k} - \mathbf{C}(t_j)\|} \quad (1.8)$$

where $r_b = kr_s$, $k \sim \mathcal{U}(1, 2)$. Points within each sphere were generated as:

$$\mathbf{y}_l = \mathbf{c}_l + r_b \cdot \mathbf{u}, \quad \mathbf{u} \sim \mathcal{U}(\mathbb{S}^2) \quad (1.9)$$

where \mathbb{S}^2 is the unit 2-sphere. The point density was kept consistent between the main structure and the spheres, calculated based on the total volume and target point count. Each point was labeled as either part of the main structure (0) or a sphere (1), forming a binary segmentation problem.

CurviSeg comprises 2500 samples in total, where each sample contains 4096 points. The dataset is split into 80% training, 10% validation, and 10% testing sets.

1.4.2 DenSpineEM Dataset

We curate a large-scale 3D dendritic spine segmentation benchmark, *DenSpineEM*, with saturated manual annotation of three EM image volumes (Fig. 1.4). In total, DenSpineEM contains **4,520** spine instances from 69 fully segmented dendrites (Tab. 1.1). In comparison, existing dendrite spine segmentation datasets are either constructed by heuristic spine extraction methods [28, 29] or lack of thorough annotation [46].

Dataset Construction. We leverage three public EM image volumes with dense dendrite segmentation to construct the DenSpineEM dataset: one $50 \times 50 \times 50 \mu m^3$ volume from the mouse somatosensory cortex [47], two $30 \times 30 \times 30 \mu m^3$ volumes from the mouse visual cortex and the human frontal lobe respectively [48] (Tab. 1.1). We refer readers to the reference for dataset details.

DenSpine-M50. We first curate DenSpine-M50 from [47] as our main dataset due its existing segmented dendrites (100+) and spines (4,000+) which are analyzed in [46]. However, on most dendrites, the spine segmentation is not thorough, making it hard to train models for practical use due to the unknown false negative errors. We pick 50 largest dendrites from the existing annotation and manually proofread all spine instance segmentation. In the end, we obtain 3,827 spine instances.

DenSpine-{M10, H10}. To evaluate the generalization performance of the model trained on DenSpine-M50 across regions and species, we build two additional datasets from AxonEM image volumes [48]: DenSpine-M10 from another brain region in the mouse and DenSpine-H10

Table 1.2: **FFD Validation on CurviSeg.** We evaluate the segmentation performance, data efficiency, rotation invariance, and computation speed of three models with and without FFD.

Method	Segmentation Performance (DSC %)			Computation Speed	
	Full Data	25% Data	Test-time Rot.	Train (s/epoch)	Inf. (ms/sample)
PointCNN[51]	92.40	84.92	91.32	210.00	119.59
w. FFD	95.42	94.77	95.60	215.40	124.78
PointNet++[52]	87.99	56.91	85.87	75.81	32.34
w. FFD	95.17	94.33	95.18	82.70	38.05
DGCNN[53]	88.95	84.32	86.41	114.11	58.87
w. FFD	95.76	95.63	95.76	122.33	63.06

from the human. Although the AxonEM dataset only provides proofread axon segmentation, we are thankful to receive saturated segmentation results for both volumes from the authors. For each of the two volumes, we first pick 10 dendrites with various dendrite types and branch thicknesses and proofread their segmentation results. Then, we go through these dendrites and annotate the spine instance segmentation.

Annotation Protocol. For a high-quality ground truth annotation, we segment spines manually with the VAST software [49] to avoid introducing bias from automatic methods. To detect errors, we use the neuroglancer software [50] to generate and visualize 3D meshes of the segmentation of dendrites and spines. Four neuroscience experts were recruited to proofread and double-confirm the annotation results for spine instance segmentation.

1.4.3 IntrA Dataset

For additional evaluation of our method on intracranial aneurysm segmentation, we utilize the entire artery subset of the IntrA dataset [54], rather than the more commonly used segment subset [9]. This subset consists of 103 3D TOF-MRA images containing 114 aneurysms. The data is provided as surface models in Wavefront OBJ files, derived from original volumetric images ($512 \times 512 \times 300$, 0.496 mm slice thickness). By using full artery models, we present a more challenging and realistic scenario for aneurysm segmentation. The dataset excludes aneurysms smaller than 3.00 mm, with sizes ranging from 3.48 to 18.66 mm (Mean: 7.49 mm, SD: 2.72 mm). Most aneurysms are saccular, with one fusiform aneurysm included.

1.5 Experiments and Results

1.5.1 Property Validation with CurviSeg Dataset

We validate FFD on the CurviSeg toyset with three point-based models, using a batch size of 8 on a single A100 GPU, and assess segmentation performance with Dice.

Segmentation Performance. As shown in Tab. 1.2, FFD consistently improved segmentation performance across all models, with 3.01%~7.18% increase in DSC.

Data Efficiency. We compared models trained on varying data scales, from 25 to 2000 samples. As shown in Fig. 1.5, models with FFD maintain high, stable performance across all

Figure 1.5: **Data Efficiency Plot of FFD.** We compare models trained on varying scales of data from CurviSeg dataset.

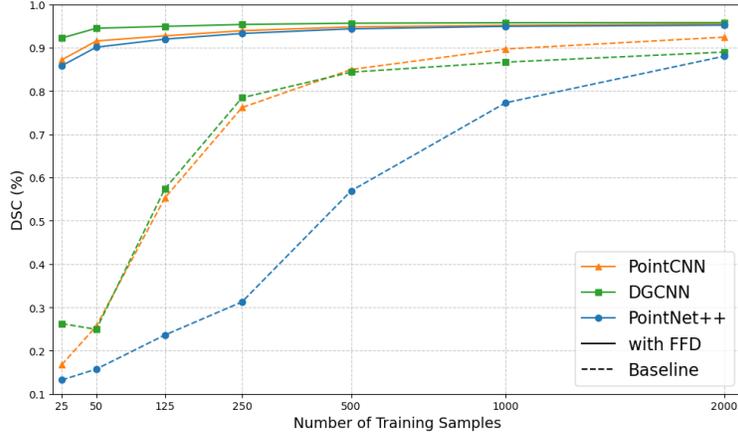


Table 1.3: **Segmentation Results on DenSpineEM.** CI 95 indicates 95% confidence interval. The results are calculated by the mean value of each fold.

Method	Subset	IoU (%)		CI 95 (%)		DSC (%)		CI 95 (%)		Spine Accuracy (%)
		Spine	Trunk	Spine	Trunk	Spine	Trunk	Spine	Trunk	
PointNet++[52]	M50	71.1	94.5	65.3 ~ 76.8	93.5 ~ 95.6	77.9	97.1	71.8 ~ 84.0	96.4 ~ 97.9	84.6 ± 6.44
	M10	73.1	88.1	71.1 ~ 75.2	86.7 ~ 89.4	81.4	93.5	79.6 ~ 83.1	92.7 ~ 94.3	77.2 ± 4.38
	H10	64.4	88.6	61.4 ~ 67.4	87.8 ~ 89.4	74.6	93.8	72.0 ~ 77.2	93.4 ~ 94.3	70.4 ± 5.86
PointNet++ w. FFD	M50	81.1	97.2	77.2 ~ 85.0	96.3 ~ 98.0	86.5	98.5	83.3 ~ 89.7	98.0 ~ 99.1	88.8 ± 6.14
	M10	81.2	94.4	79.5 ~ 83.0	93.5 ~ 95.2	87.4	97.1	86.4 ~ 88.4	96.6 ~ 97.5	85.6 ± 2.78
	H10	74.8	94.5	72.5 ~ 77.2	93.8 ~ 95.1	82.2	97.1	80.4 ~ 84.1	96.7 ~ 97.5	76.8 ± 6.18
RandLA-Net[55]	M50	18.4	57.1	13.1 ~ 23.8	43.4 ~ 68.8	24.1	71.7	18.0 ~ 30.2	61.8 ~ 82.0	43.0 ± 18.60
	M10	21.6	47.1	17.2 ~ 26.0	39.3 ~ 54.8	29.4	63.1	24.4 ~ 34.4	56.5 ~ 72.8	48.9 ± 14.17
	H10	22.4	54.8	15.7 ~ 29.1	42.5 ~ 67.1	30.1	70.0	21.4 ~ 38.8	59.0 ~ 81.0	49.1 ± 18.99
RandLA-Net w. FFD	M50	40.1	84.4	20.4 ~ 59.8	80.2 ~ 88.7	46.9	91.3	25.1 ~ 68.7	89.0 ~ 93.6	48.7 ± 22.00
	M10	40.0	77.0	24.9 ~ 55.2	69.0 ~ 85.0	48.4	86.7	31.5 ~ 65.2	81.7 ~ 91.7	37.1 ± 15.61
	H10	37.3	78.7	23.6 ~ 51.0	73.2 ~ 84.2	45.3	87.7	29.7 ~ 60.9	84.4 ~ 91.0	42.1 ± 20.70
PointTransformer[56]	M50	82.8	97.6	77.5 ~ 88.1	96.9 ~ 98.3	88.5	98.8	85.5 ~ 91.5	98.4 ~ 99.1	95.7 ± 1.01
	M10	80.8	92.1	77.9 ~ 83.7	91.4 ~ 92.7	88.8	95.7	87.6 ~ 89.9	95.4 ~ 96.1	91.7 ± 1.56
	H10	70.7	90.3	69.0 ~ 72.4	89.9 ~ 90.7	81.3	94.5	80.1 ~ 82.4	94.3 ~ 94.7	81.9 ± 1.58
PointTransformer w. FFD	M50	87.6	98.8	77.4 ~ 97.8	98.4 ~ 99.3	91.9	99.4	83.4 ~ 96.4	99.2 ~ 99.6	95.7 ± 1.56
	M10	89.1	96.0	88.1 ~ 90.1	95.4 ~ 96.6	94.1	98.0	93.3 ~ 94.9	97.6 ~ 98.3	95.8 ± 0.57
	H10	71.2	92.1	70.0 ~ 72.5	91.7 ~ 92.6	81.8	95.6	80.7 ~ 82.9	95.3 ~ 95.9	83.9 ± 2.04

data regimes, while baseline models experience sharp performance declines as data reduces. Notably, models with FFD trained on just 25% of the data (500 samples) perform similarly to those trained on the full dataset.

Rotation Invariance. We began by applying random SE(3) augmentation during test time. As shown in Tab. 1.2, with FFD, segmentation performance remained unchanged under rotations while non-FFD models experienced slight drops of 1.08%~2.54%. We further conducted a numerical analysis with 1000 SE(3)-augmented samples, comparing the representations $\mathcal{F}(P)$ and $\mathcal{F}(P_R)$. The average point-wise L2 distance was $\epsilon = (6.28 \times 10^{-26} \pm 9.13 \times 10^{-25})$, with a maximum distance of 1.85×10^{-23} , confirming the rotation invariance.

Computational Efficiency. The application of FFD introduced a marginal increase in computational cost. For the training set of 2000 samples, FFD resulted in approximately 5.40s ~ 8.22s increase in training time per epoch and 4.19ms ~ 5.71ms increase in inference time per sample, but this trade-off was minor compared to the notable improvements in

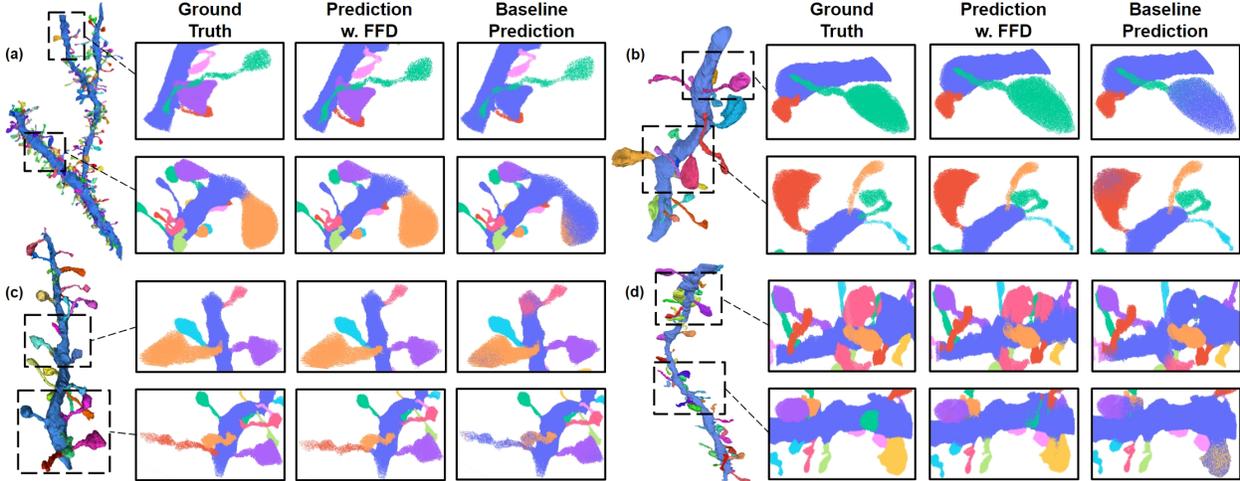


Figure 1.6: **Segmentation Results of PointTransformer on DenSpineEM.** (a~b) M50 subset; (c) M10 subset; (d) H10 subset.

segmentation performance.

Bijectivity. To empirically verify bijectivity, we randomly selected 1000 samples from CurviSeg and applied FFD, $\mathcal{D} : P \rightarrow C \times (\mathbb{R}^+ \times S^1 \times \mathbb{R})^n$, followed by its inverse, $\mathcal{D}^{-1} : C \times (\mathbb{R}^+ \times S^1 \times \mathbb{R})^n \rightarrow P'$. The average point-wise L2 distance between P and P' was $\epsilon = (8.98 \pm 7.21) \times 10^{-31}$, with a maximum error of 1.02×10^{-29} . These results confirm FFD’s bijectivity within numerical precision limits, demonstrating consistently low reconstruction errors across all samples.

1.5.2 Benchmark on Dendritic Spine Segmentation

Experiment Setup. We employ 5-fold cross-validation to train models on the M50 subset, with the M10 and H10 subsets used as test sets to evaluate cross-region and cross-species generalization, respectively. Given the extreme density of input dendrite volumes—ranging from 5.59×10^6 to 3.51×10^8 voxels, with an average of 4.82×10^7 and the sparse spine volume (0.077% to 6.99% of the dendrite), voxel-based models such as nnUNet struggle with the imbalance and requires prohibitively high memory. To address the density issue, we crop dendrites along trunk skeletons and convert them into point clouds as individual samples (Sec. 1.3.4). During training, we randomly sample 30,000 points as input; during inference, we perform repeated sampling to ensure full point cloud coverage.

Model Choice. We choose 30,000 as the sampling scale as it’s sufficient to preserve spine geometry and shapes, whereas fewer points risk losing critical information. Although 30,000 points do not constitute a large-scale point cloud, models like DGCNN, PointConv, and PointCNN encounter OOM issues on 4 NVIDIA A10 GPUs. Consequently, we selected PointNet++, PointTransformer, and RandLA-Net as baselines for their efficiency with large-scale point clouds.

Evaluation Metrics. Due to the significant foreground-background imbalance, the task is defined as binary segmentation, separating the trunk from the spine. Each spine initially

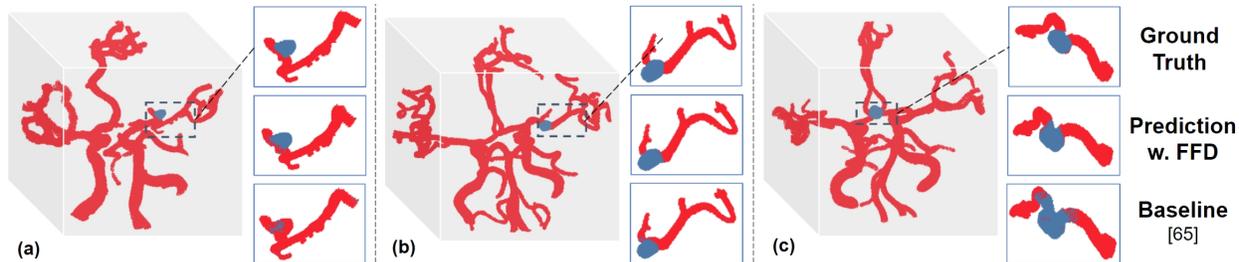


Figure 1.7: **Additional Evaluation on IntraA.** Visualization results on 3 cases from IntraA dataset are showed.

receives a unique label during dataset development; however, for experiments, segmentation is binarized to mitigate the imbalance. While these binary results can be further refined into multi-class labels via connected component grouping or clustering (e.g., DBScan), we evaluate model performance using only binary segmentation results to avoid post-processing bias. Specifically, we assess segmentation performance using DSC and IoU for both trunk and spine, with 95% confidence intervals for each metric. Spine prediction accuracy is also reported, with an individual spine considered correctly predicted if its Recall exceeds 0.7. All experiments are conducted on 4 NVIDIA A10 GPUs with PyTorch, and detailed settings along with metric tables for each fold are provided in the GitHub repository.

Results and analysis. We evaluate the segmentation performance on all three DenSpineEM subsets using models trained on the DenSpineEM-M50 subset.

Quantitative Analysis. We quantitatively evaluate the segmentation performance on the DenSpineEM dataset, as summarized in Table 1.3. Models with FFD consistently outperform baselines across all subsets. Specifically, PointTransformer w. FFD achieves the highest spine IoU and DSC on M50 with 87.6% and 91.9%, respectively, and maintains robust performance on M10 with a spine IoU of 89.1% and DSC of 94.1%. Even on the challenging H10 dataset, it attains a spine IoU of 71.2% and DSC of 81.8%, outperforming the baseline. Moreover, models with FFD exhibit high spine accuracy; for example, PointTransformer w. FFD achieves 95.7% on M50 and 95.8% on M10. The integration of FFD not only improves mean performance but also enhances consistency, as indicated by narrower 95% confidence intervals. For instance, PointNet++ w. FFD increases spine IoU from 71.1% to 81.1% on M50 over the baseline. Overall, adding FFD effectively enhances the models’ ability to segment spines accurately, improving both accuracy and generalization.

Qualitative Analysis. For qualitative analysis, we use predictions from the best-performing model, PointTransformer. We visualize two cases from the M50 dataset and one case each from M10 and H10 to evaluate generalization, as shown in Fig. 1.6. Models with FFD consistently outperform the baseline. On the M50 subset, the baseline predictions contain numerous false negatives, especially on large spines mistaken for trunks ((a), (b)-top), leading to missed spines after clustering. FFD implicitly adds a trunk skeleton prior, alleviating this issue and enhancing model robustness. In generalization tests, the model with FFD maintains robust performance on the M10 subset, while the baseline produces more false positives ((c)-top). For the H10 subset, where dendrites are longer with denser spines, both models’ performance degrades. The FFD model includes a few false positives on large spines

((d)-top) and false negatives on small spines ((d)-bottom), whereas the baseline heavily misses many spines with excessive false negatives.

1.5.3 Additional Evaluation on Intracranial Aneurysm Segmentation

Experiment Settings. We evaluated our method on the IntrA dataset using 5-fold cross-validation on the 103 TOF-MRA samples of the entire artery. The preprocessing pipeline involved voxelizing the surface model using the fast winding number method [57], skeletonizing the artery volume with TEASAR [43], pruning skeleton branches (node degree < 2 or edge length < 20 mm), and cropping the artery into vessel segments. We then applied our Frenet-Frame-based transformation and followed the two-step baseline method (detection-segmentation) [54]. For fair comparison with the baseline, we converted voxelized segmentation results back to surface point clouds before computing the Dice Similarity Coefficient (DSC) to measure segmentation accuracy.

Result Analysis. Our method achieved a DSC of 77.08% ($\pm 18.75\%$), surpassing the previous state-of-the-art performance of 71.79% ($\pm 29.91\%$) [54], which demonstrates both improved accuracy and significantly reduced variability in segmentation results. Fig. 1.7 demonstrates the qualitative superiority of applying to FFD over the baseline. In all three cases, our method more accurately delineates aneurysm boundaries (blue regions) within complex arterial structures.

1.6 Conclusion

In this study, we proposed the Frenet–Serret Frame-based Decomposition as an effective solution for accurately segmenting complex 3D curvilinear structures in (bio)medical imaging. By decomposing these structures into globally smooth curves and cylindrical primitives, we achieve reduced representational complexity and enhanced data-efficient learning. Our method demonstrates exceptional cross-region and cross-species generalization on the DenSpineEM dataset, which we developed as a comprehensive benchmark for dendritic spine segmentation, achieving high Dice scores in zero-shot segmentation tasks. Additionally, the significant performance improvement on the IntrA dataset underscores its versatility across different medical imaging applications. These results highlight the potential of our approach to advance the analysis of intricate curvilinear structures.

Chapter 2

TriSAM: Tri-Plane SAM for zero-shot cortical blood vessel segmentation in VEM images

This section draws extensively from previous work done in collaboration with the Harvard Visual Computing Group and Lichtman Lab: Jia Wan et al. “TriSAM: Tri-Plane SAM for zero-shot cortical blood vessel segmentation in VEM images”. In: arXiv preprint arXiv:2401.13961 (2024).

While imaging techniques at macro and mesoscales have garnered substantial attention and resources, microscale Volume Electron Microscopy (vEM) imaging, capable of revealing intricate vascular details, has lacked the necessary benchmarking infrastructure. In this paper, we address a significant gap in this field of neuroimaging by introducing the first-in-class public benchmark, **BvEM**, designed specifically for cortical blood vessel segmentation in vEM images. Our BvEM benchmark is based on vEM image volumes from three mammals: adult mouse, macaque, and human. We standardized the resolution, addressed imaging variations, and meticulously annotated blood vessels through semi-automatic, manual, and quality control processes, ensuring high-quality 3D segmentation. Furthermore, we developed a zero-shot cortical blood vessel segmentation method named TriSAM, which leverages the powerful segmentation model SAM for 3D segmentation. To extend SAM from 2D to 3D volume segmentation, TriSAM employs a multi-seed tracking framework, leveraging the reliability of certain image planes for tracking while using others to identify potential turning points. This approach effectively achieves long-term 3D blood vessel segmentation without model training or fine-tuning. Experimental results show that TriSAM achieved superior performances on the BvEM benchmark across three species. Our dataset, code, and model are available online at <https://jia-wan.github.io/bvem>.

2.1 Introduction

With around 2% of body weight, our brain receives around 20% of blood supply. Most of the energy and nutrients are consumed by the neurons, and neuron function is sensitive to the blood supply [61, 62]. The blood supply can even be adjusted following the consumption of brain

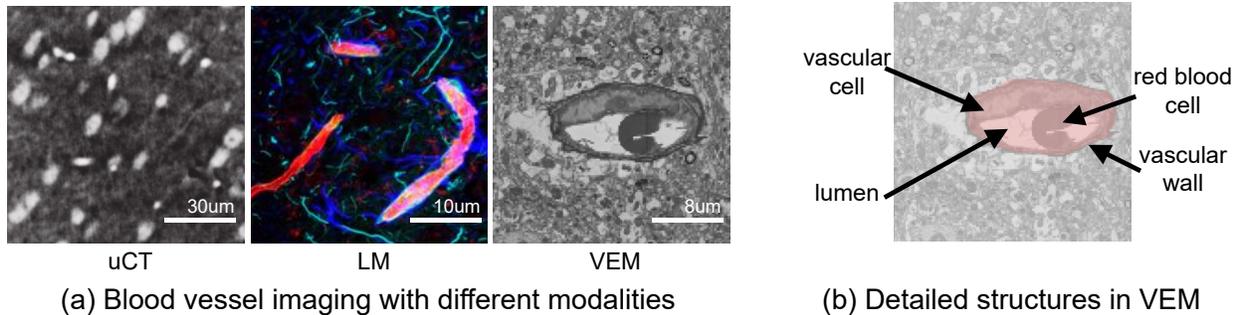


Figure 2.1: Imaging modalities for blood vessel analysis. (a) Both microtomography (μ CT) [59] and light microscopy (LM) [60] can only capture blood vessels in the cortex at the sub-micron resolution without ultrastructure details. (b) Volume electron microscopy (VEM) can show unbiased details of the vasculature including all types of cells at a higher resolution.

regions, called neurovascular coupling [63]. Alterations of blood vessel structures are observed in many brain diseases, *e.g.*, Alzheimer’s and vascular dementia [64]. Thus, blood vessels in the brain have been extensively investigated with various imaging modalities at different resolutions (Figure 2.1a). Compared to the macro-level imaging (*e.g.*, CT [59], MRI [65]) and mesoscale-level imaging (*e.g.*, light microscopy [60]), volume electron microscopy (VEM) [66] can further reveal the detailed ultrastructure including all vascular cells (Figure 2.1b) for in-depth analysis. However, no large-scale annotated VEM dataset exists to develop and evaluate automated 3D blood vessel segmentation methods.

Traditionally, the imaging methods at the macro and mesoscale are widely used and have produced a large amount of data, and a variety of image segmentation algorithms, public datasets, and evaluation methods have been developed [67, 68]. At the microscale level, the sample size of VEM is normally limited and most image analyses focus on neuron reconstruction, and blood vessels are largely ignored. Recently, owing to the rapid improvement of imaging technology, the sample size of VEM is significantly increased covering all the layers of the cerebral cortex of mouse [69] and human brain [70], as well as the whole brain of fly [71]. Moreover, imaging the whole mouse brain using VEM technology is under planning [72].

Thus, we first curate the **BvEM dataset**, the largest-to-date public benchmark dataset for cortical blood vessel segmentation in VEM images to foster segmentation method development. The raw image volumes are from recent publications, which are the largest for each of the three mammals: mouse, macaque, and human acquired at different VEM facilities. We downsampled the volumes to a consistent resolution and performed extensive blood vessel annotation, including manual proofreading, semi-automatic segmentation error correction, and quality control, involving multiple rounds of scrutiny by neuroscience experts to ensure accuracy and completeness. However, on the BvEM dataset, existing 3D blood vessel segmentation methods suffer from two major challenges: the diversity of the image appearance due to variations in the imaging pipeline and the complexity of the 3D blood vessel morphology. Conventional blood vessel segmentation heavily depends on a substantial volume of manually annotated data, a resource that is notably scarce in the existing literature.

To address these challenges, we propose a zero-shot 3D segmentation method, **TriSAM**,

leveraging the recent 2D segmentation foundation model, Segment Anything Model (SAM) [73], to handle the appearance diversity. To re-purpose SAM for 3D segmentation, we developed a multi-seed segment tracking framework based upon the video object segmentation method with SAM [74]. As it is easier to track 2D blood vessel segments along the blood flow direction, TriSAM selects the best 2D plane for SAM-based tracking and further introduces a recursive seed sampling strategy that performs tri-plane selection at the potential turning points for efficiency. The proposed method is similar to Flood-Filling Networks (FFN) [75] by extending the mask in 3D space. Instead of training a 3D neural network, we utilize SAM for prediction since it generalizes better. Second, FFN densely extends to 3D while the proposed method extends along the blood vessels. Finally, FFN uses only one seed while we are using growing seeds for long-term tracking. Experimental results demonstrate that the proposed TriSAM method significantly outperforms the prior state-of-the-art methods on the proposed BvEM benchmark across all three species.

2.2 Related Work

Blood vessel segmentation methods. Most existing VEM image segmentation algorithms were developed for neurons [76] and synapses [77, 78, 79]. The traditional VEM dataset size is too small to study blood vessel architecture. Recently, with rapid technology improvement, VEM sample size reached the cubic millimeter scale covering all the layers of cerebral cortex [70, 69], providing rich details of vasculature structure at scale. However, due to the lack of efficient and effective automatic methods, the blood vessels were segmented manually in the human cortex dataset [70]. On another front, automatic blood vessel segmentation methods have been developed for other image modalities, such as light microscopy and MRI, at a lower resolution. We refer to [67, 68] for detailed reviews on filter-based methods [80, 81] and deep learning methods [82, 83]. Due to the burden of annotation, efforts have been made to decrease the need for annotations [84]. In this paper, we propose the first VEM cortical blood vessel benchmark to foster novel methods.

Segment anything-based methods. As a foundation model for image segmentation, the recently proposed Segment Anything Model [73] has garnered significant attention and has been extended to a variety of domains [85, 86, 87, 88] including object tracking [85, 74], image inpainting [86], image matting [89], super-resolution [90], 3D point cloud [91], and image editing [92]. Despite SAM’s remarkable generalization capabilities, it still encounters some challenges in practical applications. One of these challenges is the huge computation costs due to the heavyweight image encoder. FastSAM [93] adopted a conventional CNN detector with an instance segmentation branch for the segment anything task with real-time speed. MobileSAM [94] proposed decoupled distillation to obtain a small image encoder, which achieved approximately five times faster speed compared to FastSAM while also being seven times smaller in size. Therefore the MobileSAM is employed in our proposed method. Although SAM demonstrates impressive generalization capabilities, it faces challenges in specialized domains like medical [95, 96] or biological images [97], especially in handling 3D data. [88] assessed the SAM model’s zero-shot segmentation performance in the context of digital pathology and showed scenarios where SAM encounters difficulties. [98] extensively evaluates the SAM for medical image segmentation across 19 diverse datasets, highlighting

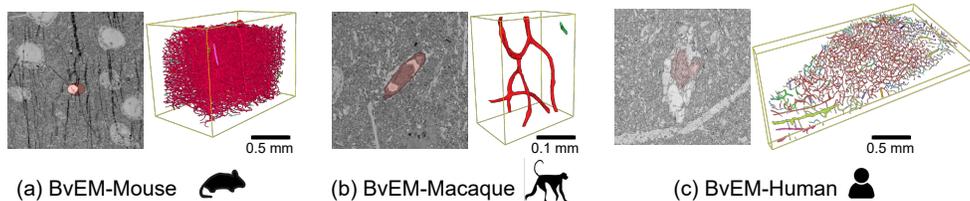


Figure 2.2: The proposed BvEM dataset. We proofread the blood vessel instance segmentation (displayed in different colors) in the three largest publicly available VEM volumes: (a) mouse [69], (b) macaque [101], and (c) human [70] samples acquired at different VEM labs.

SAM’s performance variability. To address the domain gap between natural and medical images, SAM-Adapter [95], SAM-Med2D [99], and Medical SAM Adapter [100] introduced Adapter modules and trained the Adapter with medical images. They attained good performance on various medical image segmentation tasks. MedSAM [87] adapted SAM with more than one million medical image-mask pairs and attained accurate segmentation results. MicroSAM [97] also presented a segment anything model for microscopy by fine-tuning SAM with microscopy data. Unlike these approaches that require model fine-tuning for adaptation, our method lifts the blood vessel segmentation capabilities of SAM from 2D images to 3D volumes without any model fine-tuning.

2.3 BvEM Dataset

2.3.1 Dataset Description

Tissue samples. We built the BvEM dataset upon the largest publicly available VEM image volumes for three mammal species: visual cortex from an adult mouse [69], superior temporal gyrus from an adult macaque [101], and temporal lobe from an adult human [70]. Each volume was acquired with different protocols at different facilities and we refer to respective papers for more details. As shown in Fig. 2.2, the imaging quality and the appearance of blood vessels vary drastically across these three volumes, showcasing the cutting-edge large-scale VEM imaging pipelines.

Image volumes. We processed the original image volumes into a standardized form that is suitable for benchmarking. We first downsampled all three VEM image volumes to a near-isotropic resolution ($\sim 200\text{-}300$ nm) along each dimension, which is a good balance between rich image details for biological analysis and the nature of capillary diameter. Then, we trimmed the image boundary of the BvEM-Mouse and BvEM-Human volumes, where the blood vessels are hard to annotate due to the missing image content.

2.3.2 Dataset Annotation

Initial annotation. The paper on the BvEM-human volume provides a manual proofread blood vessel segmentation, where many segments are incomplete or disconnected due to missing annotations. The papers for BvEM-mouse and BvEM-macaque volumes only provide

Table 2.1: Dataset information. Despite the difference in the scale and the geometry of the image volume, the largest blood vessel instance (rendered in red) is significantly bigger than the rest combined. The last column shows the max/total length of the blood vessel instances.

Sample	Resolution (nm)	Size (voxel)	Length: (mm)
Mouse (TEM)	320× 256×256	2495×3571×2495	1.6/1.7
Macaque (SBEM)	240×176×176	450×1271×995	713.3/714.5
Human (MultiSEM)	264×256×256	661×7752×13500	107.2/126.7

dense 3D instance segmentation and we manually selected the blood vessel segments. Due to the large scale of the BvEM-Mouse and BvEM-Human volumes, we only annotated every 4 slices, where the z-dimension resolution is around $1\mu\text{m}$.

Automatic error detection. For the false split errors, which are the majority source of error, we computed the skeleton of the segmentation and detected all the skeleton endpoints that do not touch the volume boundary as candidates. For the false merge errors, we computed the intersection-over-union (IoU) of 2D segments on neighboring slices and detected areas of small IoU value as candidates.

Manual proofreading. We used the VAST lite software [49] to accelerate the manual proofreading process by using the provided dense segmentation results as templates. Instead of manually delineating segments manually, proofreaders can coarsely draw or fill the segment mask that is snapped to the detailed boundary in the template. Assisted with the 3D blood vessel visualizations, two neuroscience experts proofread each volume in multiple rounds until no disagreement.

2.3.3 Dataset Statistics

As shown in Tab. 2.1, the BvEM-Macaque volume has around 0.5G voxels, and the BvEM-Mouse and BvEM-Human volumes are around 80 and 121 times bigger, respectively. From the blood vessel instance segmentation annotation, we automatically extracted skeleton centerlines [102] and computed the length for each blood vessel instance. Due to the hyper-connectivity nature of cortical blood vessels, the length of the largest instance is around 99%, 95%, and 85% for each volume. The histogram of the blood vessel radius in the proposed dataset is shown in Figure 2.3. The mouse and the human dataset have similar radius distributions peaked around capillaries, while the macaque volume has no capillaries.

2.4 Method

2.4.1 Overview

We aim to leverage the 2D generalist Segment Anything Model (SAM) to build a zero-shot 3D VEM blood vessel segmentation method, which can be widely applied to various image volumes. Given an input seed position, existing SAM-based tracking methods work well when the blood vessel travels only in one axis. However, in reality, blood vessels not only turn in alternating directions but also bifurcate, leading to drastic shape and scale changes.

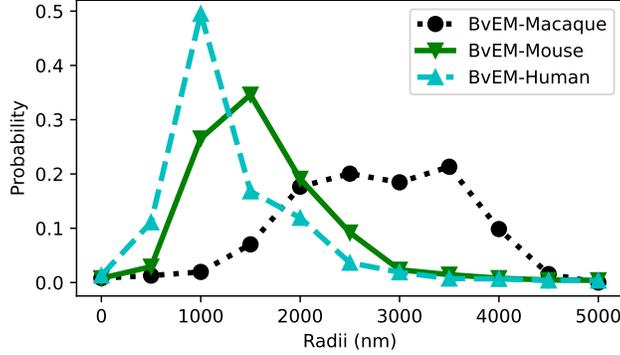


Figure 2.3: Dataset statistics. Empirical distribution of blood vessel radius.

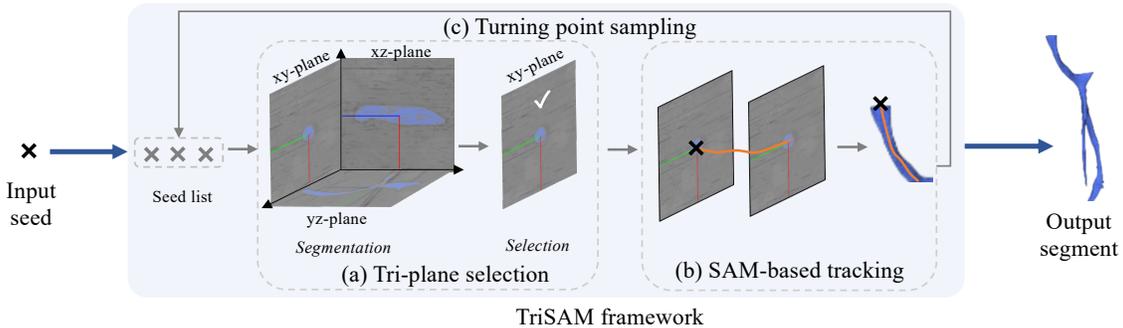


Figure 2.4: TriSAM framework. (a) Tri-plane selection is first proposed to select the best plane for tracking. (b) SAM-based tracking leverages SAM to perform short-term tracking given a seed location and a tracking axis. (c) Recursive seed sampling exploits potential turning points for long-term tracking.

Instead, we represent the whole connected blood vessel segment that contains the input seed as a graph, $G = (V, E)$, where each node $v \in V$ is a tubular branch segment that does not bifurcate. Thus, the segmentation problem can be solved with the **graph traversal** algorithm. After **visiting one node** (*i.e.*, running SAM-based tracking methods given the seed), we sample new seed positions to **find neighboring nodes** to segment recursively.

Thus, the proposed TriSAM framework has the following three modules: (1) *Tri-plane selection module* to pick the best 2D plane to segment the object (Fig. 2.4a); (2) *SAM-based tracking module* to predict 2D masks along the selected axis (Fig. 2.4b); (3) *Turning point sampling module* to propose potential turning points as new seeds based on existing segmentation results (Fig. 2.4c) to recursively grow all branches of the blood vessel. Note that the second module can be any existing SAM-based tracking method.

2.4.2 TriSAM Framework

We first explain the three modules in the TriSAM framework (Fig. 2.4) and then provide the algorithmic description (Alg. 1). In practice, we sample many initial seeds, run the TriSAM framework for each seed, and fuse the segmentation results.

Tri-plane selection module. Unlike videos, 3D VEM image volumes I can be segmented

on 2D planes along different axes, $p \in \{x, y, z\}$, based on the object morphology. However, tracking along some axis p can be harder than others due to the irregular shape of the 2D cross-section of the 3D blood vessels on the corresponding 2D plane I_p . We later verify such a phenomenon empirically as shown in Table 2.3 in which tracking along the y axis is more effective than tracking along the x axis for a given seed position.

This module aims to select the best axis p^* for tracking, given the SAM segmentation results, $\text{SAM}(I_p, s)$, on each tri-plane I_p centered at the seed position s as the approximation of the blood vessel cross-sections. In theory, the ideal axis for tracking is along the blood flow direction, where the corresponding 2D image plane is tangent to the blood flow direction with the smallest cross-section segment area (Fig. 2.4a, xy-plane). In addition, SAM outputs the probability, P_{SAM} , for the segment result, reflecting the confidence in the naturalness of its shape. Thus, to combine the two cues above, we pick the axis with the smallest segment size on the 2D plane and a probability of at least the threshold τ (PLANE-SELECT in Alg. 1).

$$p^* = \arg \min_p \text{Area}(\text{SAM}(I_p, s)), \text{ s.t. } P_{\text{SAM}}(I_p, s) > \tau. \quad (2.1)$$

To avoid an infinite recursion, traditional graph traversal algorithms check if a node is marked visited. In our case, a tubular branch segment may grow into different blood vessel segments along different tracking axes, *e.g.*, bifurcation regions. Thus, we mark the combination of the seed and the tri-plane tracking axis (s, p^*) visited if seed s falls in the segment that is predicted through tracking along p^* axis (VISITED in Alg. 1). Note that $p^* = \phi$ if no plane is selected and $\text{VISITED}((s, p^*))$ is True. To implement the ‘‘VISITED’’ function, we store both the segmentation result and the p^* , which is omitted in Alg. 1 for simplicity.

SAM-based tracking module. Given the seed position s , the selected tracking axis p^* and SAM’s initial 2D segment mask, we need to produce a 3D segment by tracking the 2D mask with SAM in both directions (*e.g.*, $x+$ and $x-$). Note that naively propagating the mask center or bounding box as the SAM prompt for the next image slice leads to poor segmentation results, as SAM may output segments with inconsistent sizes or shrinking sizes along the axis respectively. Many existing sophisticated SAM-based tracking methods can be directly used.

This module aims to provide a simple yet effective approach by generating better SAM prompts for the next image slice. Empirically, we find prompting SAM with both the enlarged bounding box and the center of the segment generates better segment tracking results (SAM-TRACK in Alg. 1).

$$\text{prompt} = \{(x, y), (x, y, \gamma w, \gamma h)\}, \quad (2.2)$$

where $(x, y, \gamma w, \gamma h)$ is the bounding box of the segmentation and γ is the scaling factor. To ensure the quality of the predicted SAM mask, the proposed tracking module terminates when the SAM probability for the predicted mask is lower than the threshold τ .

Turning point sampling module. As discussed above, SAM-based tracking module can not handle blood vessel segments that change directions or bifurcate, where the 2D segment along the original direction changes drastically. Thus, when the SAM-based tracking module terminates in the middle of the volume, we need to find **turning points** as new seeds to track segments along other directions. Naively, we can densely sample points from the segmentation result and run the PLANE-SELECT module to find points that prefer other directions. However,

Algorithm 1 TriSAM Framework

Require: 3D image volume I and initial seed s_0 , threshold τ

```
Initialize the prediction:  $\mathbf{P} = \phi$ 
Initialize the seed list:  $\mathbf{S} = \{s_0\}$ 
while  $\mathbf{S} \neq \phi$  do
  Take  $s$  from  $\mathbf{S}$ 
   $p^* = \text{PLANE-SELECT}(I, s)$ 
  if not  $\text{VISITED}(s, p^*)$  then
     $seg = \text{SAM-TRACK}(I, s, p^*)$ 
    if  $seg \neq \phi$  then
       $seeds = \text{TURNINGPOINT-SAMPLE}(seg, p^*)$ 
       $\mathbf{P} = \mathbf{P} \cup seg$ 
       $\mathbf{S} = \mathbf{S} \cup seeds$ 
    end if
  end if
end while
return  $\mathbf{P}$ 
```

such an approach can be inefficient as most points may prefer the original tracking direction, and ineffective due to false turning points caused by SAM errors.

Instead, we design this module to sample turning points around the last point prompt position from the SAM-based tracking module. We first predict 2D SAM segmentation seg_p along the other two directions ($p \in \{x, y, z\} \setminus \{p^*\}$) at the last point prompt position, which has a high probability of capturing the tangent cross-section of the swerving blood vessel. Then, we sample K points from each of the 2D segmentation seg_p with the Farthest Point Sampling (FPS) method as turning points added to the seed list.

$$seeds = \bigcup_{p \in \{x, y, z\} \setminus \{p^*\}} \text{FPS}(seg_p, K), \quad (2.3)$$

where FPS selects well-spaced points from a dataset by starting with a random point and iteratively adding the farthest point from the current selection. This method ensures good spatial coverage and uniformity.

TriSAM Framework Algorithm. Given the initial seed s_0 , the proposed TriSAM framework applies the graph traversal algorithm to segment the whole connected blood vessel segment (Alg. 1). As either breadth-first-search or depth-first-search works, we use a generic “list” data structure (\mathbf{S}) to store the seeds to visit. At each step, we take out a seed $s \in \mathbf{S}$, predict the tubular branch with the `PLANE-SELECT` and `SAM-TRACK` modules (*i.e.*, visit a node), and then find its neighboring branches to segment with the `TURNINGPOINT-SAMPLE` module (*i.e.*, add neighbors as new seeds). The TriSAM framework runs recursively until the seed list $\mathbf{S} = \phi$.

2.4.3 Implementation Details

Image Pre-processing. The large-scale datasets have many image artifacts and missing slices, which degrade the SAM-based tracking results. We utilize the temporal smoothing method along the z-axis to deflicker the images.

Initial seed generation. Initial seeds can be effectively generated with global color thresholding since the pixels of blood vessels are brighter than the background. To improve efficiency, we only keep the center of each connected component as the final seeds.

SAM details. Unless stated otherwise, we use MobileSAM [94] instead of the standard SAM [73] in all conducted experiments to improve the inference speed. To clean up SAM results, holes are filled and small connected components are removed through binary morphological operations.

Hyperparameter selection. As a zero-shot approach, TriSAM sets the hyperparameters to reasonable values instead of using ground truth data. In later ablation studies, we verify the robustness of hyperparameters within the reasonable range. For the global color thresholding for seed generation, η is set to 98 percentile. For SAM-based tracking, the probability threshold τ is set to 0.8.

Inference. For BvEM-Macaque, we directly predict the whole volume. For others, $k \times 1024 \times 1024$ subvolumes are used since 1024×1024 is the default resolution for SAM where $k = 661$ for BvEM-Human and $k = 818$ for BvEM-Mouse. The subvolume results are later fused to form the final prediction. All experiments are conducted on an NVIDIA-A100 GPU.

2.5 Experiments

2.5.1 Experimental Settings

Evaluation Metrics. We use the Precision, Recall, and Accuracy metric defined in [103], where $Accuracy = \frac{TP}{TP+FP+FN}$.

$$Precision = \frac{TP}{TP + FP}, \quad (2.4)$$

$$Recall = \frac{TP}{TP + FN}, \quad (2.5)$$

$$Accuracy = \frac{TP}{TP + FP + FN}, \quad (2.6)$$

where TP , FP , and FN are instance-level true positive, false positive, and false negative respectively. We use instance-level metrics since it is more sensitive to split errors. In particular, the Hungarian algorithm is used to match ground-truth instances and predicted instances with negative Accuracy as the cost matrix. We used the whole dataset for evaluation and computed the score on the largest instance segment of each volume due to its dominating size.

2.5.2 Benchmark Results

Methods in Comparison. We compare the proposed TriSAM with both zero-shot baselines and supervised methods. The compared zero-shot methods are global color thresholding

Table 2.2: Benchmark results on the proposed BvEM dataset. We evaluate the initial blood vessel annotation to show the significant amount of our proofreading effort.

Method	Setting	BvEM-Mouse			BvEM-Macaque			BvEM-Human		
		Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc
Initial annotation	N/A	93.74	36.62	35.74	1.65	23.17	1.57	100.00	25.68	25.68
Color thresholding	N/A	86.45	37.32	35.26	95.14	21.65	21.42	41.77	1.92	1.87
MAESTER [104]	unsupervised	2.16	18.95	0.94	22.08	40.30	16.64	0.29	5.03	0.27
3D UNet [69]	supervised	13.45	0.91	0.67	16.56	86.95	16.16	68.63	2.46	2.43
mUNET [105]	supervised	3.54	49.55	6.59	24.34	29.57	23.74	3.44	23.20	5.86
SAM+IoU tracking[73]	zero-shot	63.59	0.27	0.27	74.39	1.89	1.88	18.19	23.58	12.92
TriSAM (ours)	zero-shot	84.12	66.75	59.28	78.41	74.97	62.14	31.35	25.57	16.39

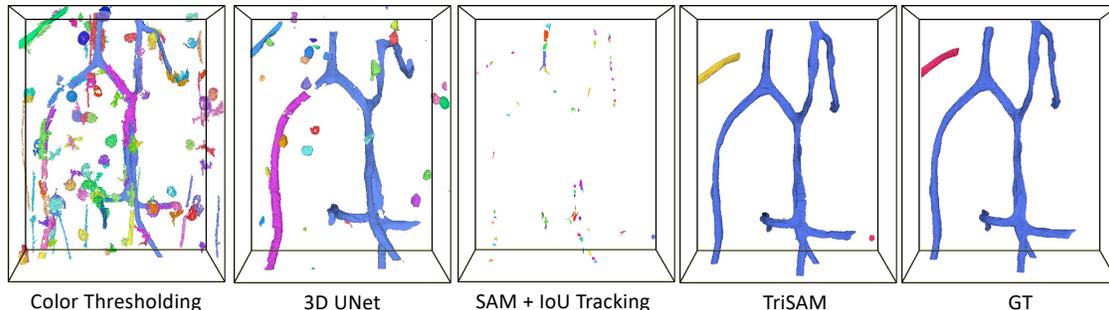


Figure 2.5: Qualitative results of instance segmentation on the BvEM-Macaque dataset. Different colors represent different instances of blood vessels. The methods of color thresholding and 3D UNet often produce false positives, meaning they mistakenly identify parts of the image as blood vessels when they are not. On the other hand, SAM+IoU tracking tends to miss many blood vessels altogether. Among all the methods tested, TriSAM segmentation performs the best and is the most accurate in identifying blood vessels.

and SAM+IoU tracking. For color thresholding, we first perform (3D) Gaussian blurring with $\sigma = 1$ on 3D chunks ($10 \times 512 \times 512$). Then we label all voxels that are 3 standard derivations above mean as positive. Finally, connected components with less than 1000 voxels are filtered out. For SAM+IoU Tracking, we segment all objects in each z-slice of the dataset using automatic mask generation. Then we track each blood vessel using the first labeled slice as seeds. Our simple tracking algorithm finds the mask in the next slice with maximum IoU with the current slice. If the max IoU is above a threshold, we assign this mask to the current object and continue tracking. We also tried SAM+IoU tracking with microSAM [106] weights that have been finetuned on EM images. This model however does not work well on our dataset. We expect this is because microSAM has been finetuned on high-resolution EM images and does not generalize to our low-resolution dataset. We further compare TriSAM with the supervised method 3D U-Net [69]. We use the implementation from [107]. For supervised methods, we cropped subvolumes at the center of each volume composing approximately 10% in size. These annotated subvolumes were divided into a 1-1 train-val split.

Results Analysis. The results are shown in Tab. 2.2. First, both the Color Thresholding and SAM+IoU Tracking methods exhibit significant performance variability across three volumes, highlighting the diversity of our dataset and the sensitivity of these methods to different species. Furthermore, both of these unsupervised methods demonstrate relatively poor performance, underscoring the challenges of the zero-shot setting in the BvEM dataset. Additionally,

Table 2.3: Ablation studies on different plane selection strategies. The proposed tri-plane approach achieves the best overall accuracy with comparable speed.

Method	Pre. (%)	Rec. (%)	Acc. (%)	Speed (sec)
Single-plane (z)	75.28	48.48	41.82	324
Single-plane (y)	79.37	58.47	50.75	307
Single-plane (x)	69.73	12.14	11.53	345
Single-plane (fusion)	71.78	74.15	57.41	976
Tri-plane	78.41	74.97	62.14	335

the 3D UNet, as a supervised learning approach, also yields subpar results, indicating poor generalization of models trained with limited data. Finally, TriSAM significantly outperforms other methods as it not only accurately segments the boundary but also tracks the blood vessels in the long term.

Qualitative Results. The final instance segmentation results on BvEM-Macaque are shown in Fig. 2.5. Color thresholding segments bright pixels, inadvertently capturing nuclei cells while overlooking darker pixels corresponding to blood vessels. Training the 3D UNet model with limited data results in confusion with background elements. IoU tracking fails to capture a significant portion of the blood vessel, revealing its ineffectiveness in tracking. TriSAM prediction emerges as the most accurate method, affirming its effectiveness. We further visualize the TriSAM’s performance on complex vessel structures in both the mouse and human datasets in Fig. 2.6. These additional examples demonstrate our method’s ability to handle challenging cases, including tortuous vessels and U-turns. While performance may vary depending on vessel complexity, our tri-plane selection strategy effectively adapts to changes in direction by dynamically choosing optimal viewing planes at potential turning points.

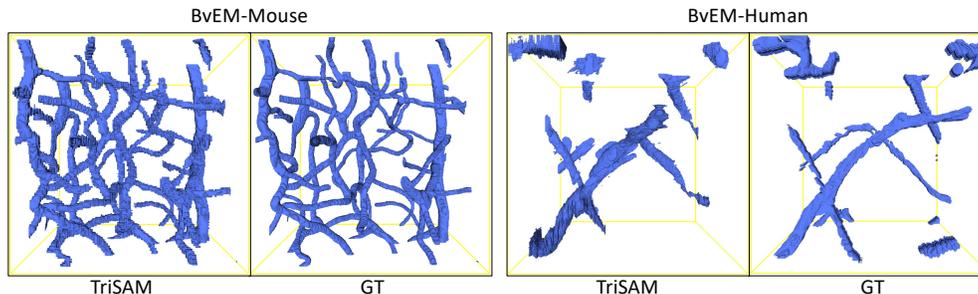


Figure 2.6: 3D visualizations on complex vessel structures from mouse and human datasets.

2.5.3 Ablation Studies

We conducted a comprehensive series of ablation studies exclusively using the BvEM-Macaque dataset due to the computation constraints.

Oracle analysis. We conduct an oracle analysis to showcase the advantage of the proposed TriSAM framework for its capability to deal with complicated vascular geometry. For the SAM-based tracking module in TriSAM, we plug in an oracle 2D segmentation method which returns the connected component of the ground truth mask containing the prompt. As shown

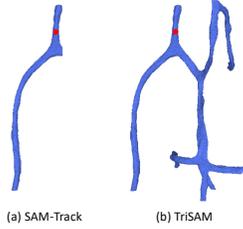


Figure 2.7: Results with oracle 2D segmentation.

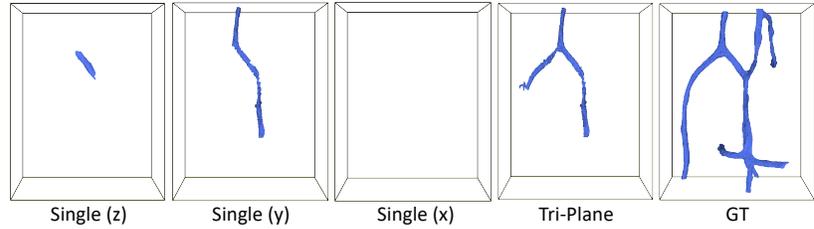


Figure 2.8: Results for different plane selection strategies without recursive seed sampling.

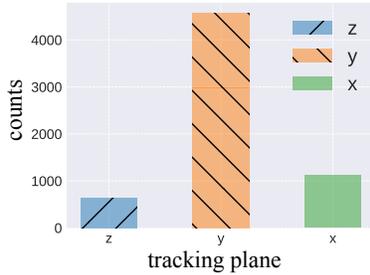


Figure 2.9: Histogram of the tri-plane selection.

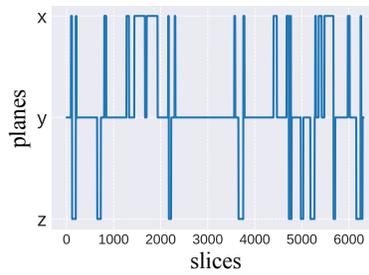


Figure 2.10: Dynamics of the tri-plane selection.

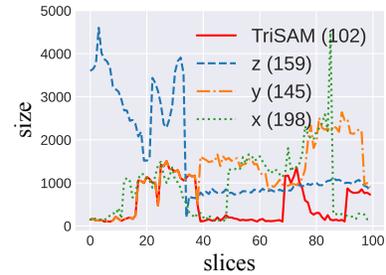


Figure 2.11: Dynamics of segment size.

in Fig. 2.7b, the TriSAM framework can perfectly segment the whole vasculature with the oracle 2D segmentation. In comparison (Fig. 2.7a), existing SAM-based video object tracking methods can only track along one axis, missing bifurcated branches.

Tri-plane selection module. We first compare our method with the single-plane methods to evaluate the effectiveness of the tri-plane. For the single-plane method, we choose one plane as the main plane and only track along the chosen plane. The results on the BvEM-Macaque volume are shown in Tab. 2.3. We have observed significant differences in performance among the three single-plane methods, with accuracy ranging from 11.53% to 50.75%, depending on the chosen tracking plane, which indicates the importance of the chosen tracking plane. This variability can be attributed to the tubular nature of blood vessel extensions within biological organisms, resulting in the generation of intricate mask shapes in certain planes, while simpler mask shapes are produced in others. Then we fuse the results of three Single-Plane methods and attain a higher accuracy of 57.41%, which demonstrates that the segmentation results from different planes exhibit a high degree of complementarity with each other. Instead, We see that tri-plane selection exploits the blood vessel 3D structures by tracking along a suitable plane and attains the highest accuracy of 62.14%.

We visualize the segmentation results with one initial seed on BvEM-Macaque in Fig. 2.8. The origin point is the seed location. We see that the performance is sensitive to the selection of the tracking plane. If the plane is not well-selected, the segmentation result can be empty as the example of tracking along the x-axis shows. The best result is tracking along the y-axis which is still worse than the proposed method since it considers potential turning points to leverage the 3D structure. The best result is tracking along the y-axis. However, it still falls short of the proposed method’s performance, as the latter takes into account potential turning points to exploit the 3D structure and perform long-term tracking.

Table 2.4: Ablation study results on the choice of SAM models. The MobileSAM model achieves better performance with faster inference speed and smaller model size.

Backbone	Pre (%)	Rec (%)	Acc (%)	Speed	Parameters
SAM [73]	77.65	66.03	55.48	1535s	615M
MobileSAM [94]	78.41	74.97	62.14	335s	9.66M

Table 2.5: Ablation studies on different seed sampling strategies. The proposed recursive seed sampling approach achieves the best overall accuracy with comparable speed.

Strategy	Pre. (%)	Rec. (%)	Acc. (%)	Speed (sec)
Naive	79.37	58.47	50.75	307
Dense sampling	86.94	24.84	23.95	2001(↑ 552%)
Recursive sampling	78.41	74.97	62.14	335 (↑ 9%)

SAM-based tracking module. We examine the effect of different SAM variants, *i.e.*, MobileSAM, on the TriSAM performance. As shown in Tab. 2.4, the inference time of MobileSAM is 22% of the original SAM which confirms that MobileSAM significantly improves the inference speed. Moreover, the performance of MobileSAM is even better than the original SAM, possibly because the distilled small model is less prone to overfitting to the original natural image domain.

Turning point sampling module. To perform long-term tracking and fully leverage the 3D blood vessel structure, we introduced recursive seed sampling by considering potential turning points. To validate its effectiveness, we report the results on the BvEM-Macaque volume in Tab. 2.5 where the runtime comparison for segmentation prediction on the entire BvEM-Macaque data is also included. Compared to the baseline without recursive seed sampling “naive” and dense seed sampling “dense”, the proposed method achieves the best performance with less than 10% running time increase. One naive baseline is to remove the recursive seed sampling component and not consider any potential turning points. This strategy is simple and fast but it fails to exploit the 3D shape prior leading to poor performance. Another strategy is to select the best plane for every tracking step/slice, which densely performs SAM segmentation on each step across three planes. Therefore, the performance is limited. The proposed method achieves the best performance with less than 10% running time increase on the BvEM-Macaque dataset. Unfortunately, it significantly increases the computation cost. Compared to our method, DenseSAM’s running time is 5.97 times longer since it needs to segment 3 planes for every tracking step. However, we were surprised to observe that the performance of the Dense Redirection strategy was even worse. This could be attributed to the frequent axis changes potentially leading to the omission of certain parts of the blood vessel and causing splitting errors.

Tri-plane dynamics. We delve deeper into the plane dynamics in Fig. 2.9 and 2.10. The majority of the selected plane predominantly tracks along the y-axis. This observation aligns with the experimental results presented in Tab.2.3, where it is evident that a single plane tracking with the y-axis outperforms the z-axis and x-axis. This is because the blood vessel flows mainly along the y-axis in the test volume. In Fig. 2.11, we explore the size dynamics using various methods, where the mean derivation is shown in parentheses. The size variation observed with our proposed method is relatively smaller compared to tracking along the

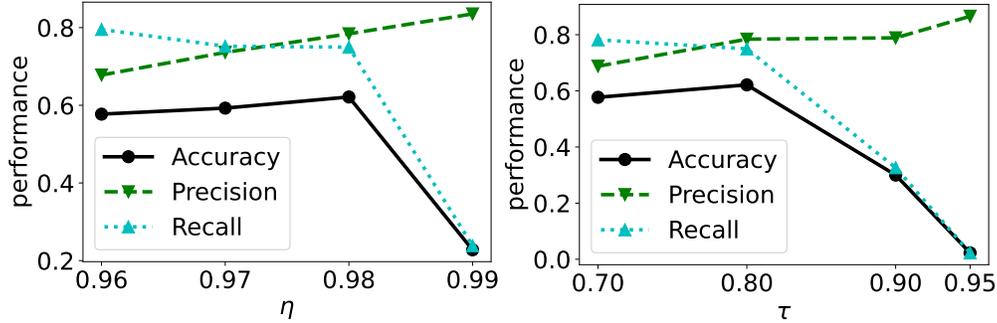


Figure 2.12: Ablation study results on hyperparameters η and τ .

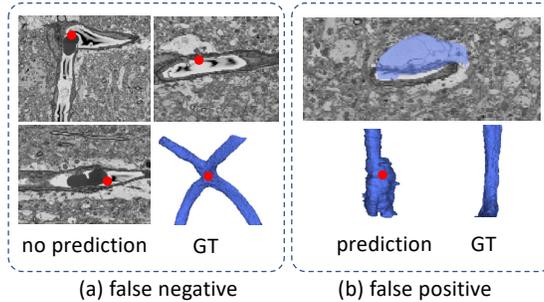


Figure 2.13: Failure cases of the SAM-based tracking module.

y-axis and significantly less than when tracking along the z-axis and x-axis.

Hyperparameters. We compute the accuracy of TriSAM with varying hyperparameters for seed generation (η) and SAM-based tracking module (τ). As shown in Fig. 2.12, TriSAM’s performance is robust within the reasonable range of hyperparameters.

2.5.4 Failure Cases

The main source of error in the proposed TriSAM framework is the SAM-based tracking module, where SAM can not produce accurate 2D mask given the prompt. For the false negative case (Fig. 2.13a), SAM predicts an empty mask when the shape and appearance of the blood vessels are complex (e.g., conjunction point) for all three axes. For the false positive case (Fig. 2.13b), SAM falsely segment the neighboring region with the blood vessel due to the similar image appearance.

2.6 Conclusion

In this paper, we contribute the largest-to-date public benchmark, the BvEM dataset, for cortical blood vessel segmentation in 3D VEM images. We also developed a zero-shot blood vessel segmentation method, TriSAM, based on the powerful SAM model, offering an efficient and accurate approach for segmenting blood vessels in VEM images. With tri-plane selection, SAM-based tracking, and recursive seed sampling, our TriSAM effectively exploits the 3D blood vessel structure and attains superior performance compared with existing zero-shot

and supervised technologies on BvEM across three species, marking a critical step towards unlocking the mysteries of neurovascular coupling and its implications for brain health and pathology. With the availability of the BvEM dataset and the TriSAM method, researchers are now equipped with valuable tools to drive breakthroughs in VEM-based cortical blood vessel segmentation and further our understanding of the brain's intricate vascular network. By addressing a significant gap in the field of neuroimaging, we have laid the foundation for advancing the understanding of cerebral vasculature at the microscale and its intricate relationship with neural function.

Chapter 3

WormID-Bench: A Benchmark for Whole-Brain Activity Extraction in *C. elegans*

This section draws extensively from previous work done in collaboration with the NYU Neuroinformatics Lab: Jason Adhinarta et al. “WormID-Benchmark: Extracting Whole-Brain Neural Dynamics of C. elegans At the Neuron Resolution”. In: bioRxiv (2025), pp. 2025–01.

The nematode *C. elegans* is a premier model organism for studying neural circuit function due to its fully mapped connectome and genetically identifiable neurons. Recent advances in 3D light microscopy and fluorescent protein tagging have enabled whole-brain imaging at single-neuron resolution. However, extracting meaningful neural dynamics from these high-resolution recordings requires addressing three fundamental challenges: (i) accurate detection of individual neurons in fluorescence images, (ii) precise identification of neuron classes based on anatomical and colorimetric cues, and (iii) robust tracking of neurons over time in calcium imaging videos. To systematically evaluate these challenges, we introduce WormID-Bench, a large-scale, multi-laboratory dataset comprising 118 worms from five distinct research groups, along with standardized evaluation metrics for detection, identification, and tracking. Our benchmark reveals that existing computational approaches show substantial room for improvement in sensitivity, specificity, and generalization across diverse experimental conditions. By providing an open and reproducible benchmarking framework¹, WormID-Bench aims to accelerate the development of high-throughput and scalable computational tools for whole-brain neural dynamics extraction in *C. elegans*, setting the stage for broader advancements in functional connectomics.

3.1 Introduction

The ability to robustly resolve whole-brain activity at single-neuron resolution remains a major challenge in neuroscience. *Caenorhabditis elegans* (*C. elegans*) serves as a powerful model system due to its fully mapped connectome and well-characterized neuronal

¹<https://github.com/focolab/WormND>

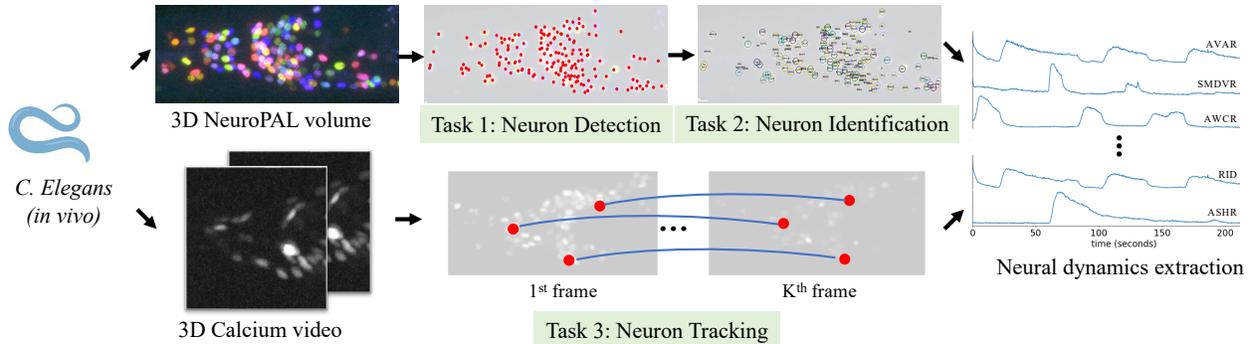


Figure 3.1: **WormID-Bench**. We benchmark each step of the computational pipeline to extract neural dynamics of *in vivo* *C. elegans* from microscopy images.

classes [109, 110]. Recent advances in 3D light-sheet microscopy and fluorescent genetic labeling (NeuroPAL) [111] enable the tracking of neuronal activity across the entire nervous system. However, a key bottleneck is the development of reliable computational methods to identify the activity for each neuron, which involves neuron detection and identification from multichannel NeuroPAL images, and neuron tracking from calcium images across diverse experimental conditions.

Automating these tasks has seen limited success; the dense distribution of similar-colored neurons and non cellular structure makes detection and identification challenging. And neural dynamics can make accurate tracking difficult, as the brightness and appearance of neurons can vary widely over time based on their firing activity [112]. Worse still, the imaging data is diverse in image format and appearance, biological variability, and batch effects across different labs and experimental setups [113, 114, 115, 116, 117, 118]. Recently WormID [119] curated several publicly available datasets with complete or semi-complete ground truth annotations that span multiple imaging setups and laboratories. However, there remains a lack of clearly defined tasks, standardized evaluation metrics, and benchmarks for state-of-the-art computer vision methods in worm neural dynamics extraction.

To address this, we introduce WormID-Benc to evaluate whole-brain neural dynamics extraction in *C. elegans* in Fig. 3.1 . Our benchmark provides (1) a structured evaluation protocol for neuron detection, identification, and tracking; (2) reproducible metrics and ranking procedures to ensure fair and unbiased model evaluation; (3) comprehensive benchmark results on the WormID dataset.

3.2 Related Work

We limit our review to light microscopy which is standard for *in vivo* live imaging.

3D Cell Detection Benchmarks. 3D cell detection is always the first step for any neuron-level imaging-related task. Previously Alwes *et al.* [120] collected and annotated volumes of 3D images of histone fluorescent protein expression in Parhyale. Another valuable resource is jGCaMP8 transgenic mice dataset, which applies two-photon imaging to record neurons in mouse visual cortex [121]. However, there are few large-scale datasets that cover neuronal volumes in *C. elegans* [122] with comprehensive annotations for cell detection, identification,

Table 3.1: Summary of WormID-Bench datasets and their parameters. #DET, #ID, and #TRAJ represent the average number of annotated neuron for detection, identification, and trajectory across samples respectively.

Dataset	resolution ($\mu\text{m}/\text{px}$)	3D NeuroPAL			3D Calcium Video		
		#samples	#DET	#ID	#samples	#frames	#TRAJ
NP [111]	$0.21 \times 0.21 \times 0.75$	10	193	190	×	×	×
HL [124]	$0.33 \times 0.33 \times 1.0$	9	119	64	×	×	×
SF [125]	$0.54 \times 0.54 \times 0.54$	38	70	70	×	×	×
EY [116]	$0.27 \times 0.27 \times 1.5$	21	177	175	21	961	177
KK [126]	$0.32 \times 0.32 \times 1.5$	9	154	154	9	1646	155
SK1 [119]	$0.16 \times 0.16 \times 1^1$	21	111	48	4	1500	108
SK2 [119]	$0.32 \times 0.32 \times 0.75^1$	10	173	49	2	3000	131

and cell tracking. Unlike previous datasets, *C. elegans* microscopy images tend to have lower resolution and have a more non-cellular structure, making the segmentation task challenging.

3D Neuron Identification Benchmarks. Neuron identification from fluorescent microscopy images is notoriously difficult. While NeuroPAL method [111] deterministically colors every neuron a stereotyped fluorescent barcode in *C. elegans*, which is identical across all stages of development for all individuals. The color code and relatively fixed position between neurons greatly simplify the identification problem by reducing the number of potential labels for a neuron.

3D Cell Tracking Benchmarks. The popular Cell Tracking Challenge [123] consists of ten 3D time-lapse microscopy datasets with various non-neuron cells. It provides dense segmentation annotations at each time frame, along with tracking annotations that indicate correspondences between frames and cell-splitting events. For neural dynamics extraction, annotating cell center points is sufficient, while minimizing significant manual labeling labor. To our knowledge, the 36 worms video dataset we are using is the only dataset with annotations for neuron subtype identities, their tracked positions, and their neural activity.

3.3 Dataset

For this benchmark, we use the WormID corpus of NeuroPAL and calcium imaging datasets curated in Sprague *et al.* [119], consisting of seven datasets from five different labs comprised of 118 total worms (Tab. 3.1). Further details can be found at WormID.org and in Sprague *et al.* [119].

Data Selection. For the detection and identification (ID) tasks, we omit 14 worms that contain significant non-linear deformities or color artifacts since many of the models we test assume that inputs are roughly aligned along the principal axes of the worm body and rely on color information. This leaves 104 remaining datasets that we use to train and benchmark each of the detection and ID approaches. For the tracking task, we omit 38 worm videos in SF [125] which does not provide enough information to extract ground truth annotations.

Data Split. Most previous approaches to the tasks outlined here typically only train and test on a small subset of data, usually all collected by one lab, limiting the generalizability of

¹Videos in SK1 and SK2 have varying z-resolutions and we refer to [119] for details.

these approaches. We split the datasets into five equal groups with balanced representation from each dataset. For each task, we perform 5-fold cross-validation and report results across multiple metrics. For the ID tasks, the whole NP dataset, which is the standard reference dataset for NeuroPAL imaging, is used for training all groups but is not used in evaluation. Thus, we report an average of cross-validated metrics across 104 worms for the detection task, 94 worms for the ID task, and 36 worms for the tracking task.

3.4 Benchmark

3.4.1 Task 1: 3D Neuron Detection in 3D NeuroPAL Volumes

Objective. The first step is to detect neurons in the NeuroPAL volumes while distinguishing them from non-neural cells and structures of the nematode. Due to the lack of visible cell boundaries and the low-resolution of the images, each neuron was annotated with the center point instead of the mask. This is particularly challenging due to the presence of background artifacts, variability in fluorescence intensity, and differences in imaging setups across laboratories.

Evaluation Metrics. We use evaluation metrics from the OCELOT Cell Detection Challenge [127], primarily the mean F1-score, along with precision and recall. Since our neuron annotations are point-based, we introduce a distance threshold to assess detection accuracy. A detection is considered correct if it falls within distance d_{th} of a ground truth neuron; otherwise it is a false positive. If multiple ground truth neurons are within d_{th} , the nearest one is matched. To account for varying proximity requirements, we evaluate at two distance thresholds: $d_{\text{th}} \in \{3\mu\text{m}, 6\mu\text{m}\}$, roughly 1-2 times the nucleus diameter, which ensures spatial accuracy and alignment with calcium imaging for cell tracking.

Baseline Models. Neuron detection methods can be categorized into point-based and mask-based approaches. Point-based methods predict a heatmap of neuron center locations, with peak detection algorithms used to extract precise coordinates, such as nn-UNet adopted for heatmap regression [105]. Mask-based methods leverage pre-trained generalist models, such as CellPose [128] and microSAM [129], to segment neuron regions. The centroid of each masked object is extracted as its cell coordinate.

3.4.2 Task 2: 3D Neuron Identification in 3D NeuroPAL Volumes

Objective. The 302-neuron nervous system of *C. elegans* is eutelic, meaning each neuron has a consistent identity across individuals. The computational task involves the 302-way classification for each detected neuron center in a 3D NeuroPAL image volume. Despite its stereotyped nervous system, neuron identification is challenging due to natural variability in cell positions for different individuals, body distortion during movement, color overlap for several neurons, and imaging inconsistencies between different labs. Moreover, the input detection results can have missed neurons or false positives, and some neurons remain difficult to distinguish even for experts.

Evaluation Metrics. We adopted the standard classification accuracy to measure the proportion of correctly labeled neuron centers in test images. The accuracy is reported for

both top-1 and top-5 ranked assignments. Based on the number of neurons labeled, datasets are categorized into low and high label counts, as fewer labels typically yield higher accuracy since the easiest neurons are identified first. In contrast, fully labeled datasets tend to have lower accuracy due to the ambiguity of harder-to-identify neurons.

Baseline Models. Neuron identification methods fall into two main categories: alignment-based and classification-based approaches. Alignment-based methods perform non-rigid point cloud registration to align predicted neuron positions with a reference template. Techniques such as Coherent Point Drift (CPD) [130] are commonly used, followed by neuron label assignment using the Hungarian algorithm [131] or a learned statistical atlas [115]. Classification-based methods train machine learning models to directly predict neuron identities. Examples include a transformer-based approach [132] and a graph-based conditional random fields model [124]. For learning-based methods, we evaluate both pre-trained and fine-tuned models to assess their generalization across datasets.

3.4.3 Task 3: 3D Neuron Tracking in 3D Calcium Videos

Objective. The task involves tracking neurons across video frames and estimating activity from image intensity. Neuron tracking is challenging due to biological noise, deformations, missing neurons, and variability in data collection across labs. Previous methods have used pose registration to align neurons under deformation of the worm body. Additionally, inactive neurons may be undetectable in the calcium images while remaining visible under fluorescent channels invariant to excitation.

Evaluation metrics. We adopted the evaluation metrics from the Cell Tracking Challenge [123], specifically Detection Accuracy (DET) and Tracking Accuracy (TRA). TRA is based on the Acyclic Oriented Graph Matching measure [133], which computes the edit-distance between graphs, robustly handling cases where tracks are split or swapped. As in Task 1, setting a distance threshold is crucial to assessing detection and tracking accuracy. Therefore, we apply two distance thresholds, $d_{th} \in \{3\mu m, 6\mu m\}$.

Baseline methods. Neuron tracking methods can be classified into matching-based and propagation-based approaches. Matching-based methods detect neuron center points in individual video frames and establish correspondences across frames. Matches are determined using integer linear programming (ILP) [134] or directly predicted by a feed-forward neural network. Propagation-based methods estimate optical flow to compute dense voxel-level correspondences between neighboring frames. Neuron center points are then propagated from the previous frame to the current frame by following these correspondences [135].

3.5 Results

For each task, we report 5-fold cross-validation results with the standard deviation to alleviate the bias from the data split.

Table 3.2: Neuron detection results (Task 1). We report metrics using two different distance thresholds for for both pre-trained and re-trained ([†]) models.

Method	$d_{th}=3\mu m$			$d_{th}=6\mu m$		
	Precision	Recall	F1	Precision	Recall	F1
CellPose [128]	0.45±0.02	0.58±0.01	0.47±0.01	0.50±0.03	0.66±0.01	0.53±0.02
Micro-SAM [129]	0.41±0.01	0.43±0.02	0.39±0.01	0.50±0.01	0.53±0.03	0.47±0.01
nn-UNet [†] [105]	0.75±0.03	0.70±0.02	0.71±0.02	0.80±0.03	0.75±0.02	0.76±0.01

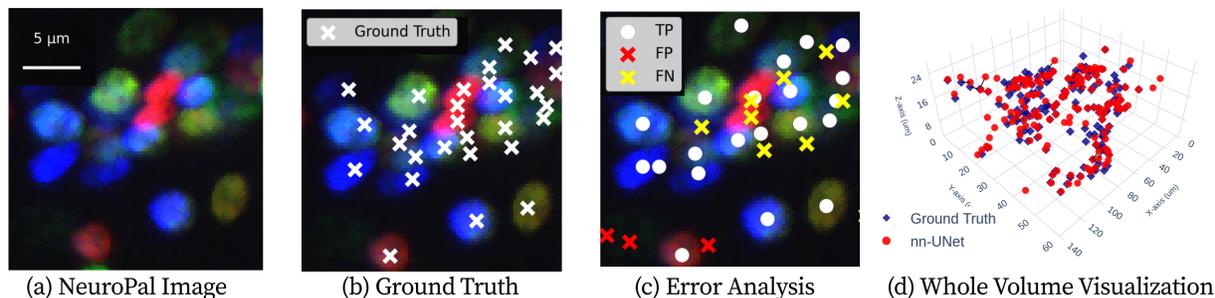


Figure 3.2: Qualitative neuron detection results (Task 1). (a) A zoomed-in NeuroPAL image, (b) ground truth detection points overlaid, (c) nn-UNet detection results showing true positives, false positives, and false negatives, and (d) whole volume visualization of ground truth and nn-UNet prediction in 3D.

3.5.1 Task 1: 3D Neuron Detection in 3D NeuroPAL Volumes

Quantitative results. We evaluated pre-trained and retrained detection models, establishing a baseline for comparison in Tab. 3.2. CellPose used two-channel images to match its pretraining setup, achieving better out-of-the-box performance than pretrained Micro-SAM. nn-UNet outperformed all models, highlighting the effectiveness of fine tuning point-based method for neuron detection.

Qualitative results. As shown in Fig. 3.2, most errors occur in regions with densely packed neurons of similar colors. The false negative errors in yellow are often due to low intensity or contrast of ground truth neurons blending into background noise. The false positive errors in red are often caused by non-neuronal cells or background noise in the image that are misclassified into neurons.

CellPose and Micro-SAM are not natively designed for 3D volumetric NeuroPAL images with color, which presents challenges in directly adapting these methods to such data. Despite this, the pre-trained models produce reasonable results, achieved through the post-hoc stitching of 2D segmentation maps.

3.5.2 Task 2: 3D Neuron Identification in 3D NeuroPAL Volumes

Quantitative results. As shown in Tab. 3.3, all methods have higher accuracy with a low number of labels compared to a high number of labels. Since identification is a classification task, performance improves when there are more candidate choices, which explains the higher accuracy in Top-5 compared to Top-1. The traditional methods (CPD, Statistical

Table 3.3: Neuron identification results (Task 2). Mean top-1 and top-5 accuracies are reported for both pre-trained and re-trained ([†]) models.

Method	Low #ID labels (easy)		High #ID labels (hard)		Overall	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CPD [130]	0.39±0.15	0.65±0.14	0.39±0.09	0.67±0.10	0.39±0.14	0.65±0.13
Stats. Atlas [115]	0.40±0.12	0.64±0.12	0.41±0.08	0.72±0.05	0.41±0.11	0.66±0.11
Stats. Atlas [†] [115]	0.66±0.14	0.89±0.07	0.54±0.06	0.79±0.06	0.62±0.13	0.86±0.08
fDNC [132]	0.52±0.04	0.73±0.04	0.29±0.07	0.50±0.09	0.45±0.03	0.66±0.03
fDNC [†] [132]	0.60±0.14	0.79±0.11	0.38±0.16	0.61±0.20	0.53±0.18	0.74±0.17
CRF ID [124]	0.59±0.20	0.84±0.19	0.47±0.07	0.73±0.05	0.55±0.18	0.81±0.17
CRF ID [†] [124]	0.81±0.09	0.95±0.05	0.57±0.06	0.78±0.04	0.74 ±0.14	0.89±0.09

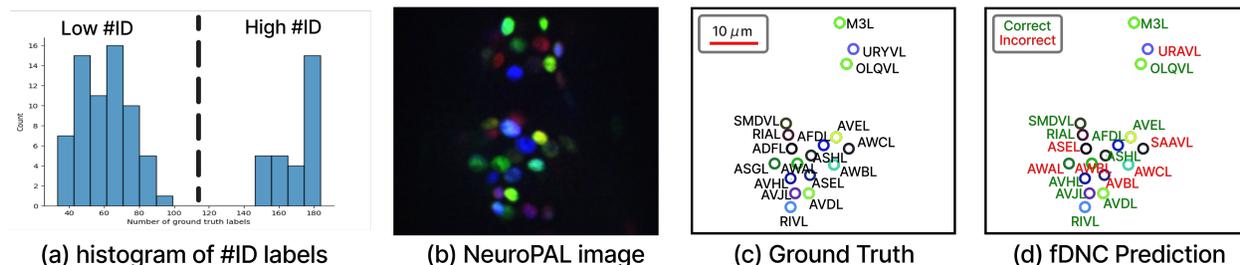


Figure 3.3: Qualitative neuron identification results (Task 2). (a) We divide the samples into low and high number of ID labels for fine-grained analysis. We show (b) one image from the original 3D NeuroPAL volume, (c) the ground truth label by human experts, and (d) the prediction result by fDNC. The wrong prediction (e.g. ASGL) comes from a nearby neuron with a similar color (AWAL).

Atlas) have relatively lower accuracy than the deep learning methods (CRF ID, fDNC) when they are not finetuned on all datasets. However, there is a significant improvement in their performance after retraining on the full datasets, especially for the CRF ID, which has the best performance among all methods.

Qualitative results. As shown in Fig. 3.3, a common source of errors is misclassification due to nearby neurons with similar colors, particularly in methods that rely solely on positional information. The CPD method, which lacks additional information, serves as a baseline for comparison. In contrast, the other three methods improve accuracy by incorporating color and prior probability distributions, enabling more reliable neuron identification. The fDNC method applies a transformer network for matching but is sensitive to worm body orientation during pre-processing, likely contributing to its lower accuracy. The Statistical Atlas method uses a statistical model, while the CRF ID method utilizes a graphical model, leveraging structured prior knowledge for improved performance. The high accuracy of both methods after re-training demonstrates the importance of training on diverse datasets.

3.5.3 Task 3: 3D Neuron Tracking in 3D Calcium Videos

Quantitative results. As shown in Table 3.4, Ultrack consistently outperforms 3DeeCell-Tracker in both detection (DET) and tracking (TRA) accuracy across distance thresholds.

Table 3.4: Neuron tracking results (Task 3). We report metrics using two different distance thresholds using 5-fold cross-validation.

Method	$d_{th}=3\mu m$		$d_{th}=6\mu m$	
	DET	TRA	DET	TRA
3DeeCellTracker [126]	0.38±0.06	0.36±0.06	0.42±0.08	0.40±0.07
Ultrack [134]	0.70±0.04	0.66±0.05	0.81±0.05	0.78±0.05

Table 3.5: Tracking results with sampled oracle detections with [126] at $d_{th}=3\mu m$.

Oracle %	TRA
100	0.99
80	0.74
60	0.52

Although both methods use the same StarDist detections, Ultrack benefits from additional ultrametric contours maps. To assess the impact of detection quality on tracking performance, we input ground truth detections at 100%, 80%, and 60% to 3DeeCellTracker. As expected, the DET metric matches the sampling percentage. As shown in Table 3.5, 3DeeCellTracker significantly improves with better detections, highlighting the need for more accurate detection methods to enhance overall tracking performance.

3.6 Limitation and Discussion

WormID-Bench provides a structured framework for benchmarking whole-brain neural dynamics extraction in *C. elegans*, yet several limitations persist. Despite its diverse dataset, it may not fully encompass the range of experimental conditions and imaging setups used across different research groups, potentially introducing domain gaps that affect model generalization. Additionally, existing detection, identification, and tracking methods exhibit inconsistent performance across setups, highlighting the need for more robust, domain-adaptive approaches. Another critical challenge is the cascading error propagation across tasks—errors in neuron detection can lead to misidentifications, which subsequently compromise tracking performance, amplifying inaccuracies at each stage. Future work should explore end-to-end learning pipelines that mitigate these dependencies through joint optimization strategies. Furthermore, while the benchmark promotes standardized evaluation, enhancements such as active learning for dataset expansion, the integration of spatial-temporal priors, and leveraging foundation models could further improve generalization across laboratories.

Bibliography

- [1] Leslie Gu, Jason K. Adhinarta, Mikhail Bessmeltsev, Jiancheng Yang, Yongjie Jessica Zhang, Wenjie Yin, Daniel Berger, Jeff Lichtman, Hanspeter Pfister, and Donglai Wei. “Frenet-Serret Frame-based Decomposition for Part Segmentation of 3D Curvilinear Structures”. In: *arXiv preprint arXiv:2404.14435* (2024).
- [2] Alex Fornito, Andrew Zalesky, and Michael Breakspear. “The connectomics of brain disorders”. In: *Nature Reviews Neuroscience* 16 (2015), pp. 159–172.
- [3] Jun Yao Wu, Su-Ji Cho, Katherine D. Descant, Peter H. Li, Alexander Shapson-Coe, Michał Januszewski, and et al. “Mapping of neuronal and glial primary cilia contactome and connectome in the human cerebral cortex”. In: *Neuron* 112 (2023), 41–55.e3.
- [4] John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, and et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (2021), pp. 583–589.
- [5] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, and et al. “Accurate structure prediction of biomolecular interactions with AlphaFold 3”. In: *Nature* 630 (2024), pp. 493–500.
- [6] Jiancheng Yang, Shixuan Gu, Donglai Wei, Hanspeter Pfister, and Bingbing Ni. “RibSeg Dataset and Strong Point Cloud Baselines for Rib Segmentation from CT Scans”. In: *MICCAI*. Springer. 2021, pp. 611–621.
- [7] Liang Jin, Shixuan Gu, Donglai Wei, Jason Ken Adhinarta, Kaiming Kuang, Yongjie Jessica Zhang, and et al. “RibSeg v2: A Large-scale Benchmark for Rib Labeling and Anatomical Centerline Extraction”. In: *IEEE Transactions on Medical Imaging* 43 (1 2024), pp. 570–581.
- [8] Jiancheng Yang, Rui Shi, Liang Jin, Xiaoyang Huang, Kaiming Kuang, D. Wei, and et al. “Deep Rib Fracture Instance Segmentation and Classification from CT on the RibFrac Challenge”. In: *ArXiv abs/2402.09372* (2024).
- [9] Xi Yang, Ding Xia, Taichi Kin, and Takeo Igarashi. “Intra: 3D Intracranial Aneurysm Dataset for Deep Learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 2653–2663.
- [10] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus Maier-Hein. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature Methods* 18 (2020), pp. 203–211.

- [11] Yutaro Kashiwagi, Takahito Higashi, Kazuki Obashi, Yuka Sato, Noboru H. Komiyama, Seth G. N. Grant, and et al. “Computational geometry analysis of dendritic spines by structured illumination microscopy”. In: *Nature Communications* 10 (2019).
- [12] Benjamin D. Boros, Kelsey M. Greathouse, Erik G. Gentry, Kendall Curtis, Elizabeth L. Birchall, Marla Gearing, and et al. “Dendritic spines provide cognitive resilience against Alzheimer’s disease”. In: *Annals of Neurology* 82 (2017).
- [13] Isabel Vidaurre-Gallart, Isabel Fernaud-Espinosa, Nicusor Cosmin-Toader, Lidia Talavera-Martínez, Miguel Martin-Abadal, Ruth Benavides-Piccione, and et al. “A Deep Learning-Based Workflow for Dendritic Spine Segmentation”. In: *Frontiers in Neuroanatomy* 16 (2022).
- [14] Ertunç Erdil, Ali Ozgur Argunsah, Tolga Tasdizen, Devrim Ünay, and Müjdat Çetin. “A joint classification and segmentation approach for dendritic spine segmentation in 2-photon microscopy images”. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)* (2015), pp. 797–800.
- [15] Subhadip Basu, Nirmal Das, Ewa Baczynska, Monika Bijata, André Zeug, Dariusz M Plewczynski, and et al. “3dSpAn: an interactive software for 3D segmentation and analysis of dendritic spines”. In: *bioRxiv* (2019).
- [16] David Lesage, Elsa D. Angelini, Isabelle Bloch, and Gareth Funka-Lea. “A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes”. In: *Medical image analysis* 13 6 (2009), pp. 819–45.
- [17] Ludovica Acciai, Paolo Soda, and Giulio Iannello. “Automated Neuron Tracing Methods: An Updated Account”. In: *Neuroinformatics* 14 (2016), pp. 353–367.
- [18] Pechin Lo, Bram van Ginneken, Joseph M. Reinhardt, Tarunashree Yavarna, Pim A. de Jong, Benjamin Irving, and et al. “Extraction of Airways From CT (EXACT’09)”. In: *IEEE Transactions on Medical Imaging* 31 (2012), pp. 2093–2107.
- [19] Alejandro F Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. “Multiscale Vessel Enhancement Filtering”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 1998.
- [20] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, and et al. “Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images”. In: *Medical image analysis* 2 2 (1998), pp. 143–68.
- [21] Amos Sironi, Engin Türetken, Vincent Lepetit, and Pascal V. Fua. “Multiscale Centerline Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (2016), pp. 1327–1341.
- [22] Giles Tetteh, Velizar Efremov, Nils Daniel Forkert, Matthias Schneider, Jan Stefan Kirschke, Bruno Weber, and et al. “DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes”. In: *Frontiers in Neuroscience* 14 (2018).

- [23] Geert J. S. Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, and et al. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.
- [24] Esther A Nimchinsky, Bernardo L. Sabatini, and Karel Svoboda. “Structure and function of dendritic spines.” In: *Annual review of physiology* 64 (2002), pp. 313–53.
- [25] Xuerong Xiao, Maja Djuricic, Assaf Hoogi, Richard W Sapp, Carla J Shatz, and Daniel L Rubin. “Automated dendritic spine detection using convolutional neural networks on maximum intensity projected microscopic volumes”. In: *Journal of neuroscience methods* 309 (2018), pp. 25–34.
- [26] Nirmal Das, Ewa Baczynska, Monika Bijata, Blazej Ruszczycycki, Andre Zeug, Dariusz Plewczynski, and et al. “3dSpAn: An interactive software for 3D segmentation and analysis of dendritic spines”. In: *Neuroinformatics* (2021), pp. 1–20.
- [27] Junyoung Choi, Sang-Eun Lee, Yein Lee, Eunji Cho, Sunghoe Chang, and Won-Ki Jeong. “DXplorer: A Unified Visualization Framework for Interactive Dendritic Spine Analysis using 3D Morphological Features”. In: *IEEE Transactions on Visualization and Computer Graphics* (2021).
- [28] Gregg Wildenberg, Hanyu Li, Griffin Badalamente, Thomas D Uram, Nicola J Ferrier, and Narayanan Kasthuri. “Large-scale dendritic spine extraction and analysis through petascale computing”. In: *bioRxiv* (2021).
- [29] Sven Dorkenwald, Nicholas L Turner, Thomas Macrina, Kisuk Lee, Ran Lu, Jingpeng Wu, and et al. “Binary and analog variation of synapses between cortical pyramidal neurons”. In: *BioRxiv* (2019).
- [30] H. T. H. Piaggio. “Differential Geometry of Curves and Surfaces”. In: *Nature* 169 (1952), pp. 560–560.
- [31] Richard Bishop. “There is More than One Way to Frame a Curve”. In: *American Mathematical Monthly* 82 (1975), pp. 246–251.
- [32] Jianyu Huang, Zuguang He, Yutaka Arakawa, and Billy Dawton. “Trajectory Planning in Frenet Frame via Multi-Objective Optimization”. In: *IEEE Access* 11 (2023), pp. 70764–70777.
- [33] Gökhan Köseoğlu and Mustafa Kemal Bilici. “Involutive sweeping surfaces with Frenet frame in Euclidean 3-space”. In: *Heliyon* 9 (2023).
- [34] Andrew J. Hanson and Hui Ma. “Quaternion Frame Approach to Streamline Visualization”. In: *IEEE Trans. Vis. Comput. Graph.* 1 (1995), pp. 164–174.
- [35] Helmut Pottmann and Michael G. Wagner. “Contributions to Motion Based Surface Design”. In: *Int. J. Shape Model.* 4 (1998), pp. 183–196.
- [36] Shuangwei Hu, Martin Lundgren, and Antti J. Niemi. “Discrete Frenet frame, inflection point solitons, and curve visualization with applications to folded proteins.” In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 83 6 Pt 1 (2011), p. 061908.
- [37] Hariharan Ravishankar, Rahul Venkataramani, Sheshadri Thiruvankadam, Prasad Sudhakar, and Vivek Vaidya. “Learning and incorporating shape models for semantic segmentation”. In: *MICCAI*. Springer. 2017, pp. 203–211.

- [38] Yan Wang, Xu Wei, Fengze Liu, Jieneng Chen, Yuyin Zhou, Wei Shen, and et al. “Deep distance transform for tubular structure segmentation in ct scans”. In: *CVPR*. 2020, pp. 3833–3842.
- [39] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. “3D U-Net: learning dense volumetric segmentation from sparse annotation”. In: *MICCAI*. Springer. 2016, pp. 424–432.
- [40] Jiancheng Yang, Xiaoyang Huang, Yi He, Jingwei Xu, Canqian Yang, Guozheng Xu, and et al. “Reinventing 2d convolutions for 3d images”. In: *IEEE Journal of Biomedical and Health Informatics* (2021).
- [41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *CVPR*. 2017, pp. 652–660.
- [42] Ngoc-Vuong Ho, Tan Nguyen, Gia-Han Diep, Ngan Le, and Binh-Son Hua. “Point-Unet: A Context-Aware Point-Based Neural Network for Volumetric Segmentation”. In: *MICCAI*. Springer. 2021, pp. 644–655.
- [43] M. Sato, I. Bitter, M.A. Bender, A.E. Kaufman, and M. Nakajima. “TEASAR: tree-structure extraction algorithm for accurate and robust skeletons”. In: *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*. IEEE. DOI: 10.1109/pccga.2000.883951.
- [44] Ingmar Bitter, Arie E. Kaufman, and Mie Sato. “Penalized-Distance Volumetric Skeleton Algorithm”. In: *IEEE Trans. Vis. Comput. Graph.* 7 (2001), pp. 195–206.
- [45] Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and et al. “L1-medial skeleton of point cloud”. In: *ACM Transactions on Graphics (TOG)* 32 (2013), pp. 1–8.
- [46] Netanel Ofer, Daniel R Berger, Narayanan Kasthuri, Jeff W Lichtman, and Rafael Yuste. “Ultrastructural analysis of dendritic spine necks reveals a continuum of spine morphologies”. In: *Developmental neurobiology* (2021).
- [47] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, and et al. “Saturated Reconstruction of a Volume of Neocortex”. In: *Cell* 162.3 (July 2015), pp. 648–661. DOI: 10.1016/j.cell.2015.06.054.
- [48] Donglai Wei, Kisuk Lee, Hanyu Li, Ran Lu, J Alexander Bae, Zequan Liu, and et al. “AxonEM Dataset: 3D Axon Instance Segmentation of Brain Cortical Regions”. In: *MICCAI*. 2021.
- [49] Daniel R Berger, H Sebastian Seung, and Jeff W Lichtman. “VAST (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3D image stacks”. In: *Frontiers in neural circuits* (2018).
- [50] *Neuroglancer*. URL: <https://github.com/google/neuroglancer>.
- [51] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. “PointCNN: Convolution On X-Transformed Points”. In: *Neural Information Processing Systems*. 2018. URL: <https://api.semanticscholar.org/CorpusID:53399839>.

- [52] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *NIPS* 30 (2017).
- [53] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Transactions on Graphics (TOG)* 38 (2018), pp. 1–12. URL: <https://api.semanticscholar.org/CorpusID:94822>.
- [54] Xi Yang, Ding Xia, Taichi Kin, and Takeo Igarashi. “A two-step surface-based 3D deep learning pipeline for segmentation of intracranial aneurysms”. In: *Computational Visual Media* 9 (2020), pp. 57–69.
- [55] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Agathoniaki Trigoni, and A. Markham. “RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 11105–11114. URL: <https://api.semanticscholar.org/CorpusID:208290898>.
- [56] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. “Point Transformer V3: Simpler, Faster, Stronger”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 4840–4851. URL: <https://api.semanticscholar.org/CorpusID:266335894>.
- [57] Gavin Barill, Neil G. Dickson, Ryan M. Schmidt, David I. W. Levin, and Alec Jacobson. “Fast winding numbers for soups and clouds”. In: *ACM Transactions on Graphics (TOG)* 37 (2018), pp. 1–12.
- [58] Jia Wan, Wanhua Li, Jason K. Adhinarta, Atmadeep Banerjee, Evelina Sjostedt, Jingpeng Wu, Jeff Lichtman, Hanspeter Pfister, and Donglai Wei. “TriSAM: Tri-Plane SAM for zero-shot cortical blood vessel segmentation in VEM images”. In: *arXiv preprint arXiv:2401.13961* (2024).
- [59] Eva L Dyer, William Gray Roncal, Judy A Prasad, Hugo L Fernandes, Doga Gürsoy, Vincent De Andrade, Kamel Fezzaa, Xianghui Xiao, Joshua T Vogelstein, Chris Jacobsen, et al. “Quantifying mesoscale neuroanatomy using X-ray microtomography”. In: *eneuro* 4.5 (2017).
- [60] Jeffrey J Lochhead, Erica I Williams, Elizabeth S Reddell, Emma Dorn, Patrick T Ronaldson, and Thomas P Davis. “High Resolution Multiplex Confocal Imaging of the Neurovascular Unit in Health and Experimental Ischemic Stroke”. In: *Cells* 12.4 (2023), p. 645.
- [61] Benjamin J Andreone, Baptiste Lacoste, and Chenghua Gu. “Neuronal and vascular interactions”. In: *Annual review of neuroscience* 38 (2015), pp. 25–46.
- [62] Achim Peters, U Schweiger, L Pellerin, C Hubold, KM Oltmanns, M Conrad, B Schultes, Jet Born, and HL Fehm. “The selfish brain: competition for energy resources”. In: *Neuroscience & Biobehavioral Reviews* 28.2 (2004), pp. 143–180.
- [63] Costantino Iadecola. “Neurovascular regulation in the normal brain and in Alzheimer’s disease”. In: *Nature Reviews Neuroscience* 5.5 (2004), pp. 347–360.

- [64] Raj N Kalaria. “Vascular basis for brain degeneration: faltering controls and risk factors for dementia”. In: *Nutrition reviews* 68.suppl_2 (2010), S74–S87.
- [65] Donald M McDonald and Peter L Choyke. “Imaging of angiogenesis: from microscope to clinic”. In: *Nature medicine* 9.6 (2003), pp. 713–725.
- [66] Christopher J Peddie, Christel Genoud, Anna Kreshuk, Kimberly Meechan, Kristina D Micheva, Kedar Narayan, Constantin Pape, Robert G Parton, Nicole L Schieber, Yannick Schwab, et al. “Volume electron microscopy”. In: *Nature Reviews Methods Primers* 2.1 (2022), p. 51.
- [67] Mohammad Raihan Goni, Nur Intan Raihana Ruhaiyem, Muzaimi Mustapha, Anusha Achuthan, and Che Mohd Nasril Che Mohd Nassir. “Brain vessel segmentation using deep learning—a review”. In: *IEEE Access* (2022).
- [68] Sara Moccia, Elena De Momi, Sara El Hadji, and Leonardo S Mattos. “Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics”. In: *Computer methods and programs in biomedicine* 158 (2018), pp. 71–91.
- [69] The MICrONS Consortium, J. Alexander Bae, et al., and Chi Zhang. “Functional connectomics spanning multiple areas of mouse visual cortex”. In: *bioRxiv* (2023). DOI: 10.1101/2021.07.28.454025. eprint: <https://www.biorxiv.org/content/early/2023/04/19/2021.07.28.454025.full.pdf>. URL: <https://www.biorxiv.org/content/early/2023/04/19/2021.07.28.454025>.
- [70] Alexander Shapson-Coe, Michał Januszewski, Daniel R Berger, Art Pope, Yuelong Wu, Tim Blakely, and et al. “A connectomic study of a petascale fragment of human cerebral cortex”. In: *BioRxiv* (2021).
- [71] Sven Dorkenwald, et al., and the FlyWire Consortium. “Neuronal wiring diagram of an adult brain”. In: *bioRxiv* (2023). DOI: 10.1101/2023.06.27.546656. eprint: <https://www.biorxiv.org/content/early/2023/07/11/2023.06.27.546656.full.pdf>. URL: <https://www.biorxiv.org/content/early/2023/07/11/2023.06.27.546656>.
- [72] Larry F Abbott, Davi D Bock, Edward M Callaway, Winfried Denk, Catherine Dulac, Adrienne L Fairhall, and et al. “The mind of a mouse”. In: *Cell* 182.6 (2020), pp. 1372–1376.
- [73] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. “Segment anything”. In: *arXiv preprint arXiv:2304.02643* (2023).
- [74] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. “Track anything: Segment anything meets videos”. In: *arXiv preprint arXiv:2304.11968* (2023).
- [75] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. “High-precision automated reconstruction of neurons with flood-filling networks”. In: *Nature methods* 15.8 (2018), pp. 605–610.

- [76] Kisuk Lee, Nicholas Turner, Thomas Macrina, Jingpeng Wu, Ran Lu, and H Sebastian Seung. “Convolutional nets for reconstructing neural circuits from brain images acquired by serial section electron microscopy”. In: *Current Opinion in Neurobiology* 55 (2019). Machine Learning, Big Data, and Neuroscience, pp. 188–198. ISSN: 0959-4388. DOI: <https://doi.org/10.1016/j.conb.2019.04.001>.
- [77] Jingpeng Wu, Yicong Li, Nishika Gupta, Kazunori Shinomiya, Pat Gunn, Alexey Polilov, Hanspeter Pfister, Dmitri Chklovskii, and Donglai Wei. “An Out-of-Domain Synapse Detection Challenge for Microwasp Brain Connectomes”. In: *arXiv preprint arXiv:2302.00545* (2023).
- [78] Nicholas L Turner, Kisuk Lee, Ran Lu, Jingpeng Wu, Dodam Ih, and H Sebastian Seung. “Synaptic partner assignment using attentional voxel association networks”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1–5.
- [79] Julia Buhmann, Arlo Sheridan, Caroline Malin-Mayor, Philipp Schlegel, Stephan Gerhard, Tom Kazimiers, Renate Krause, Tri M Nguyen, Larissa Heinrich, Wei-Chung Allen Lee, et al. “Automatic detection of synaptic partners in a whole-brain Drosophila electron microscopy data set”. In: *Nature methods* 18.7 (2021), pp. 771–774.
- [80] Philbert S Tsai, John P Kaufhold, Pablo Blinder, Beth Friedman, Patrick J Drew, Harvey J Karten, Patrick D Lyden, and David Kleinfeld. “Correlations of neuronal and microvascular densities in murine cortex revealed by direct counting and colocalization of nuclei and vessels”. In: *Journal of Neuroscience* 29.46 (2009), pp. 14553–14570.
- [81] Xiang Ji, Tiago Ferreira, Beth Friedman, Rui Liu, Hannah Liechty, Erhan Bas, Jayaram Chandrashekar, and David Kleinfeld. “Brain microvasculature has a common topology with local differences in geometry that match metabolic load”. In: *Neuron* 109.7 (2021), pp. 1168–1187.
- [82] Giles Tetteh, Velizar Efremov, Nils D Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Björn H Menze. “Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes”. In: *Frontiers in Neuroscience* 14 (2020), p. 1285.
- [83] Michelle Livne, Jana Rieger, Orhun Utku Aydin, Abdel Aziz Taha, Ela Marie Akay, Tabea Kossen, Jan Sobesky, John D Kelleher, Kristian Hildebrand, Dietmar Frey, et al. “A U-Net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease”. In: *Frontiers in neuroscience* 13 (2019), p. 97.
- [84] Vien Ngoc Dang, Francesco Galati, Rosa Cortese, Giuseppe Di Giacomo, Viola Marconetto, Prateek Mathur, Karim Lekadir, Marco Lorenzi, Ferran Prados, and Maria A Zuluaga. “Vessel-CAPTCHA: an efficient learning framework for vessel annotation and segmentation”. In: *Medical Image Analysis* 75 (2022), p. 102263.
- [85] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. “Segment and track anything”. In: *arXiv preprint arXiv:2305.06558* (2023).

- [86] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. “Inpaint anything: Segment anything meets image inpainting”. In: *arXiv preprint arXiv:2304.06790* (2023).
- [87] Jun Ma and Bo Wang. “Segment anything in medical images”. In: *arXiv preprint arXiv:2304.12306* (2023).
- [88] Ruining Deng, Can Cui, Quan Liu, Tianyuan Yao, Lucas W Remedios, Shunxing Bao, Bennett A Landman, Lee E Wheless, Lori A Coburn, Keith T Wilson, et al. “Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging”. In: *arXiv preprint arXiv:2304.04155* (2023).
- [89] Jingfeng Yao, Xinggang Wang, Lang Ye, and Wenyu Liu. “Matte Anything: Interactive Natural Image Matting with Segment Anything Models”. In: *arXiv preprint arXiv:2306.04121* (2023).
- [90] Zhihe Lu, Zeyu Xiao, Jiawang Bai, Zhiwei Xiong, and Xinchao Wang. “Can SAM Boost Video Super-Resolution?” In: *arXiv preprint arXiv:2305.06524* (2023).
- [91] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. “Segment Any Point Cloud Sequences by Distilling Vision Foundation Models”. In: *arXiv preprint arXiv:2306.09347* (2023).
- [92] Shanghua Gao, Zhijie Lin, Xingyu Xie, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. “EditAnything: Empowering Unparalleled Flexibility in Image Editing and Generation”. In: *Proceedings of the 31st ACM International Conference on Multimedia, Demo track*. 2023.
- [93] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. “Fast Segment Anything”. In: *arXiv preprint arXiv:2306.12156* (2023).
- [94] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. “Faster Segment Anything: Towards Lightweight SAM for Mobile Applications”. In: *arXiv preprint arXiv:2306.14289* (2023).
- [95] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Shangzhan Zhang, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. “SAM Fails to Segment Anything?—SAM-Adapter: Adapting SAM in Underperformed Scenes: Camouflage, Shadow, and More”. In: *arXiv preprint arXiv:2304.09148* (2023).
- [96] Shizhan Gong, Yuan Zhong, Wena Ma, Jinpeng Li, Zhao Wang, Jingyang Zhang, Pheng-Ann Heng, and Qi Dou. “3DSAM-adapter: Holistic Adaptation of SAM from 2D to 3D for Promptable Medical Image Segmentation”. In: *arXiv preprint arXiv:2306.13465* (2023).
- [97] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. “Segment Anything for Microscopy”. In: *bioRxiv* (2023), pp. 2023–08.
- [98] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. “Segment anything model for medical image analysis: an experimental study”. In: *Medical Image Analysis* 89 (2023), p. 102918.

- [99] Junlong Cheng, Jin Ye, Zhongying Deng, Jianpin Chen, Tianbin Li, Haoyu Wang, Yanzhou Su, Ziyang Huang, Jilong Chen, Lei Jiang, et al. “SAM-Med2D”. In: *arXiv preprint arXiv:2308.16184* (2023).
- [100] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. “Medical sam adapter: Adapting segment anything model for medical image segmentation”. In: *arXiv preprint arXiv:2304.12620* (2023).
- [101] Sahil Loomba, Jakob Straehle, Vijayan Gangadharan, Natalie Heike, Abdelrahman Khalifa, Alessandro Motta, Niansheng Ju, Meike Sievers, Jens Gempt, Hanno S Meyer, et al. “Connectomic comparison of mouse and human cortex”. In: *Science* 377.6602 (2022), eabo0924.
- [102] W Silversmith, JA Bae, PH Li, and AM Wilson. “Kimimaro: Skeletonize densely labeled 3D image segmentations”. In: *Zenodo <https://doi.org/10.5281/zenodo.5539912>* (2021).
- [103] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. “Star-convex polyhedra for 3D object detection and segmentation in microscopy”. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 3666–3673.
- [104] Ronald Xie, Kuan Pang, Gary D Bader, and Bo Wang. “MAESTER: Masked Autoencoder Guided Segmentation at Pixel Resolution for Accurate, Self-Supervised Subcellular Structure Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3292–3301.
- [105] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211.
- [106] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas R. Dengel, Sheraz Ahmed, and Constantin Pape. “Segment Anything for Microscopy”. In: *bioRxiv* (2023). URL: <https://api.semanticscholar.org/CorpusID:261125076>.
- [107] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches Barro, Marion Louveaux, Christian Wenzl, Sören Strauss, David Wilson-Sánchez, Rena Lymbouridou, Susanne S Steigleder, Constantin Pape, Alberto Bailoni, Salva Duran-Nebreda, George W Bassel, Jan U Lohmann, Miltos Tsiantis, Fred A Hamprecht, Kay Schneitz, Alexis Maizel, and Anna Kreshuk. “Accurate and versatile 3D segmentation of plant tissues at cellular resolution”. In: *eLife* 9 (July 2020). Ed. by Christian S Hardtke, Dominique C Bergmann, Dominique C Bergmann, and Moritz Graeff, e57613. ISSN: 2050-084X. DOI: 10.7554/eLife.57613. URL: <https://doi.org/10.7554/eLife.57613>.
- [108] Jason Adhinarta, Jizheng Dong, Tianxiao He, Junxiang Huang, Daniel Sprague, Jia Wan, Hyun Jee Lee, Zikai Yu, Hang Lu, Eviatar Yemini, et al. “WormID-Benchmark: Extracting Whole-Brain Neural Dynamics of *C. elegans* At the Neuron Resolution”. In: *bioRxiv* (2025), pp. 2025–01.

- [109] John G White, Eileen Southgate, J Nichol Thomson, Sydney Brenner, et al. “The structure of the nervous system of the nematode *Caenorhabditis elegans*”. In: *Philos Trans R Soc Lond B Biol Sci* (1986).
- [110] Daniel Witvliet, Ben Mulcahy, James K Mitchell, Yaron Meirovitch, Daniel R Berger, Yuelong Wu, Yufang Liu, Wan Xian Koh, Rajeev Parvathala, Douglas Holmyard, et al. “Connectomes across development reveal principles of brain maturation”. In: *Nature* (2021).
- [111] Eviatar Yemini, Albert Lin, Amin Nejatbakhsh, Erdem Varol, Ruoxi Sun, Gonzalo E Mena, Aravinthan DT Samuel, Liam Paninski, Vivek Venkatachalam, and Oliver Hobert. “NeuroPAL: a multicolor atlas for whole-brain neuronal identification in *C. elegans*”. In: *Cell* 184.1 (2021), pp. 272–288.
- [112] Jeffrey P Nguyen, Frederick B Shipley, Ashley N Linder, George S Plummer, Mochi Liu, Sagar U Setru, Joshua W Shaevitz, and Andrew M Leifer. “Whole-brain calcium imaging with cellular resolution in freely behaving *Caenorhabditis elegans*”. In: *Proceedings of the National Academy of Sciences* 113.8 (2016), E1074–E1081.
- [113] Yu Toyoshima, Terumasa Tokunaga, Osamu Hirose, Manami Kanamori, Takayuki Teramoto, Moon Sun Jang, Sayuri Kuge, Takeshi Ishihara, Ryo Yoshida, and Yuichi Iino. “Accurate automatic detection of densely distributed cell nuclei in 3D space”. In: *PLoS computational biology* 12.6 (2016), e1004970.
- [114] Yu Toyoshima, Stephen Wu, Manami Kanamori, Hirofumi Sato, Moon Sun Jang, Suzu Oe, Yuko Murakami, Takayuki Teramoto, Chanhyun Park, Yuishi Iwasaki, et al. “Neuron ID dataset facilitates neuronal annotation for whole-brain activity imaging of *C. elegans*”. In: *BMC biology* 18 (2020), pp. 1–20.
- [115] Erdem Varol, Amin Nejatbakhsh, Ruoxi Sun, Gonzalo Mena, Eviatar Yemini, Oliver Hobert, and Liam Paninski. “Statistical Atlas of *C. elegans* neurons”. In: *MICCAI* (2020).
- [116] Amin Nejatbakhsh, Erdem Varol, Eviatar Yemini, Vivek Venkatachalam, Albert Lin, Aravinthan DT Samuel, and Liam Paninski. “Extracting neural signals from semi-immobilized animals with deformable non-negative matrix factorization”. In: *bioRxiv* (2020), pp. 2020–07.
- [117] Amin Nejatbakhsh, Erdem Varol, Eviatar Yemini, Oliver Hobert, and Liam Paninski. “Probabilistic joint segmentation and labeling of *c. elegans* neurons”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*. Springer. 2020, pp. 130–140.
- [118] Yuxiang Wu, Shang Wu, Xin Wang, Chengtian Lang, Quanshi Zhang, Quan Wen, and Tianqi Xu. “Rapid detection and recognition of whole brain activity in a freely behaving *Caenorhabditis elegans*”. In: *PLoS computational biology* 18.10 (2022), e1010594.

- [119] Daniel Yutaka Sprague, Kevin Rusch, Raymond L Dunn, Jackson Borchardt, Greg Bubnis, Grace Chiu, Chentao Wen, Ryoga Suzuki, Shivesh Chaudhary, Ben Dichter, et al. “Unifying community-wide whole-brain imaging datasets enables robust automated neuron identification and reveals determinants of neuron positioning in *C. elegans*”. In: *bioRxiv* (2024), pp. 2024–04.
- [120] Frederike Alwes, Camille Enjolras, and Michalis Averof. “Live imaging reveals the progenitors and cell dynamics of limb regeneration”. In: *Elife* 5 (2016), e19766.
- [121] Tim Wang, Marius Pachitariu, Caiying Guo, Albert K Lee, Jae Sung Lee, Yan Zhang, Marton Rozsa, Jeremy Hasseman, Ilya Kolb, Jim D. Cox, Crystal Lopez, Loren Looger, and Karel Svoboda. “jGCaMP8 transgenic mice”. In: (Nov. 2023). DOI: 10.25378/janelia.24565837.v3. URL: https://janelia.figshare.com/articles/dataset/Untitled_Item_b_jGCaMP8_transgenic_mice_b_/24565837.
- [122] Błażej Ruszczycki, Katarzyna Karolina Pels, Agnieszka Walczak, Katarzyna Zamłyńska, Michał Such, Andrzej Antoni Szczepankiewicz, Małgorzata Hanna Hall, Adriana Magalska, Marta Magnowska, Artur Wolny, et al. “Three-dimensional segmentation and reconstruction of neuronal nuclei in confocal microscopic images”. In: *Frontiers in Neuroanatomy* 13 (2019), p. 81.
- [123] Martin Maška, Vladimír Ulman, Pablo Delgado-Rodriguez, Estibaliz Gómez-de-Mariscal, Tereza Nečasová, Fidel A Guerrero Peña, Tsang Ing Ren, Elliot M Meyerowitz, Tim Scherr, Katharina Löffler, et al. “The Cell Tracking Challenge: 10 years of objective benchmarking”. In: *Nature Methods* 20.7 (2023), pp. 1010–1020.
- [124] Shivesh Chaudhary, Sol Ah Lee, Yueyi Li, Dhaval S Patel, and Hang Lu. “Graphical-model framework for automated annotation of cell identities in dense cellular images”. In: *Elife* 10 (2021), e60321.
- [125] Adam A Atanas, Jungsoo Kim, Ziyu Wang, Eric Bueno, McCoy Becker, Di Kang, Jungyeon Park, Talya S Kramer, Flossie K Wan, Saba Baskoylu, et al. “Brain-wide representations of behavior spanning multiple timescales and states in *C. elegans*”. In: *Cell* 186.19 (2023), pp. 4134–4151.
- [126] Chentao Wen, Takuya Miura, Venkatakaushik Voleti, Kazushi Yamaguchi, Motosuke Tsutsumi, Kei Yamamoto, Kohei Otomo, Yukako Fujie, Takayuki Teramoto, Takeshi Ishihara, et al. “3DeeCellTracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images”. In: *Elife* 10 (2021), e59187.
- [127] Jeongun Ryu, Aaron Valero Puche, JaeWoong Shin, Seonwook Park, Biagio Brattoli, Jinhee Lee, Wonkyung Jung, Soo Ick Cho, Kyunghyun Paeng, Chan-Young Ock, et al. “Ocelot: Overlapped cell on tissue dataset for histopathology”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 23902–23912.
- [128] Carsen Stringer and Marius Pachitariu. “Cellpose3: one-click image restoration for improved cellular segmentation”. In: *bioRxiv* (2024). DOI: 10.1101/2024.02.10.579780.

- [129] Anwai Archit, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Sagnik Gupta, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. “Segment Anything for Microscopy”. In: *bioRxiv* (2023). DOI: 10.1101/2023.08.21.554208.
- [130] Andriy Myronenko and Xubo Song. “Point Set Registration: Coherent Point Drift”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (Dec. 2010), pp. 2262–2275. ISSN: 0162-8828. DOI: 10.1109/tpami.2010.46.
- [131] H. W. Kuhn. “The Hungarian method for the assignment problem”. In: *Naval Research logistics Quarterly* 2 (1955).
- [132] Xinwei Yu, Matthew S Creamer, Francesco Randi, Anuj K Sharma, Scott W Linderman, and Andrew M Leifer. “Fast deep neural correspondence for tracking and identifying neurons in *C. elegans* using semi-synthetic training”. In: *eLife* 10 (July 2021). Ed. by Gordon J Berman, e66410. DOI: 10.7554/eLife.66410. URL: <https://doi.org/10.7554/eLife.66410>.
- [133] Pavel Matula, Martin Maška, Dmitry V Sorokin, Petr Matula, Carlos Ortiz-de-Solórzano, and Michal Kozubek. “Cell tracking accuracy measurement based on comparison of acyclic oriented graphs”. In: *PloS one* 10.12 (2015), e0144959.
- [134] Jordão Bragantini, Merlin Lange, and Loïc Royer. “Large-Scale Multi-Hypotheses Cell Tracking Using Ultrametric Contours Maps”. In: *arXiv preprint arXiv:2308.04526* (2023).
- [135] James Ryu, Amin Nejatbakhsh, Mahdi Torkashvand, Sahana Gangadharan, Maedeh Seyedolmohadesin, Jinmahn Kim, Liam Paninski, and Vivek Venkatachalam. “Versatile multiple object tracking in sparse 2D/3D videos via deformable image registration”. In: *PLOS Computational Biology* 20.5 (2024), e1012075.