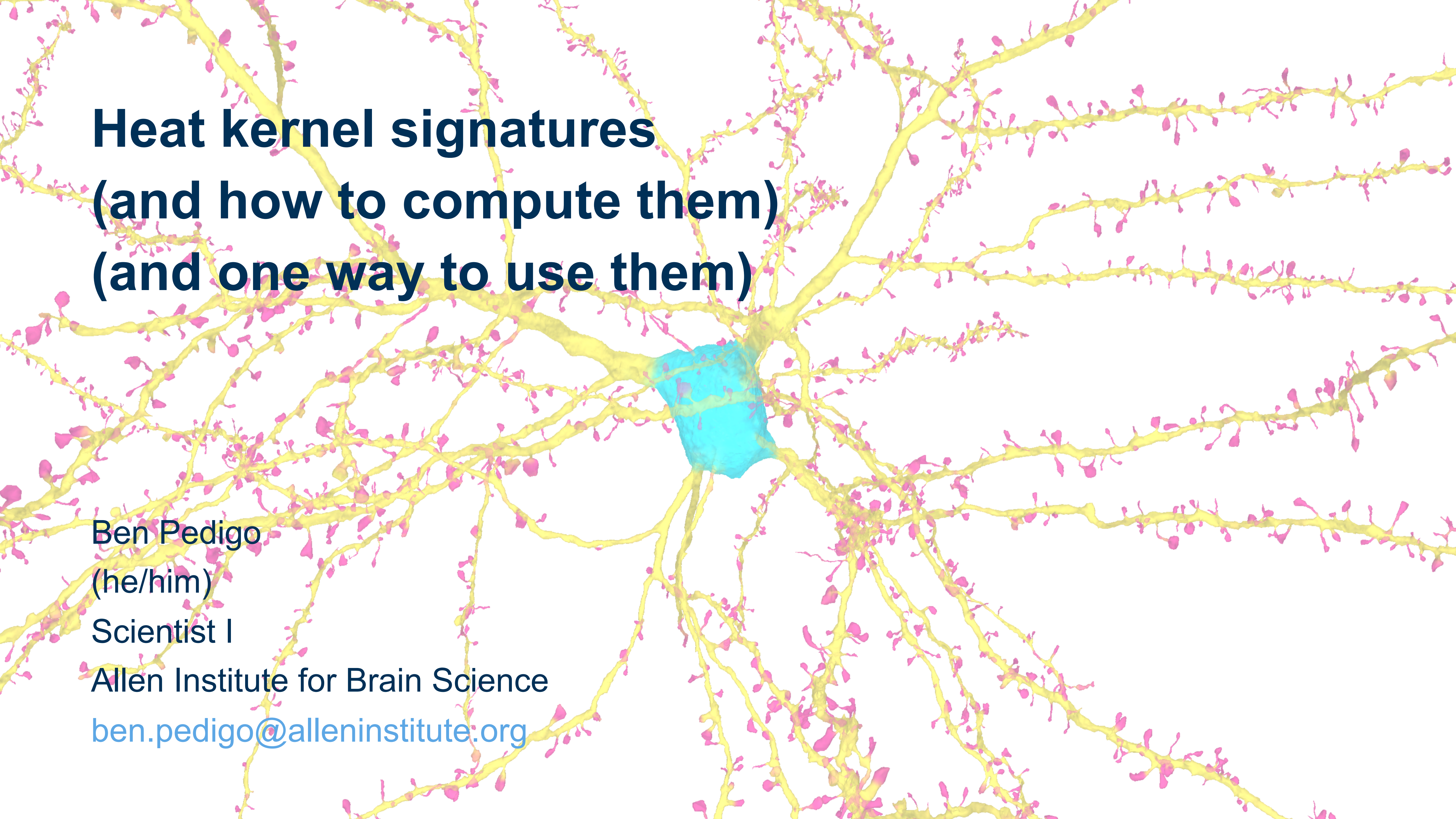


Heat kernel signatures (and how to compute them) (and one way to use them)

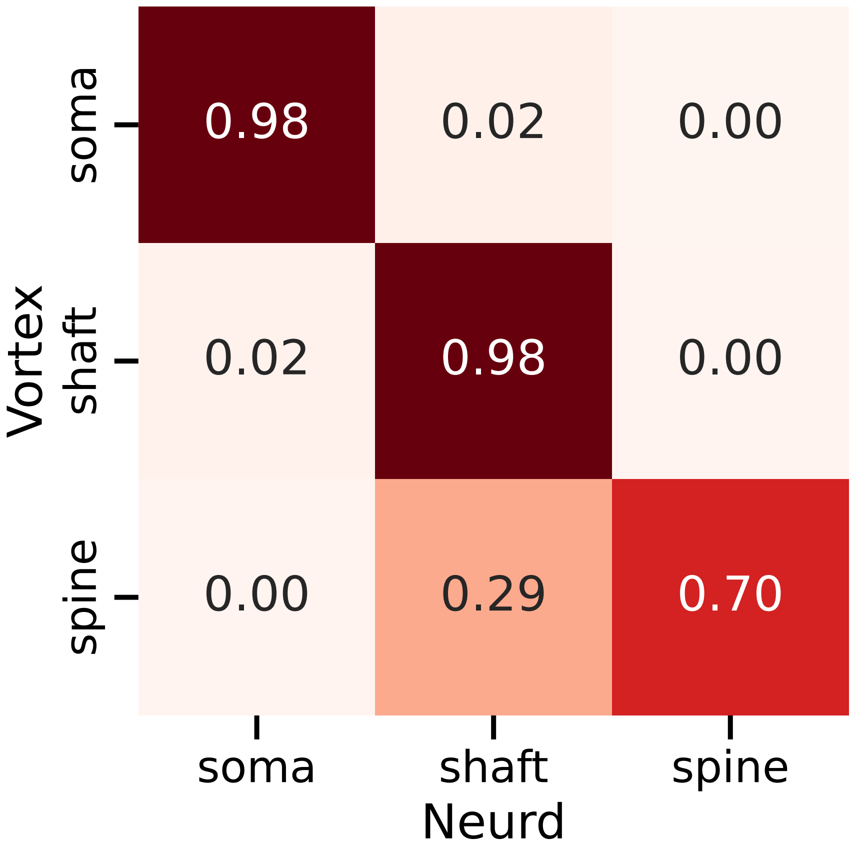
Ben Pedigo
(he/him)
Scientist I
Allen Institute for Brain Science
ben.pedigo@alleninstitute.org

Outline

- **Motivation**
- Intuition for heat kernel signatures
- Computing heat kernel signatures
- Application to spine prediction
- Extensions

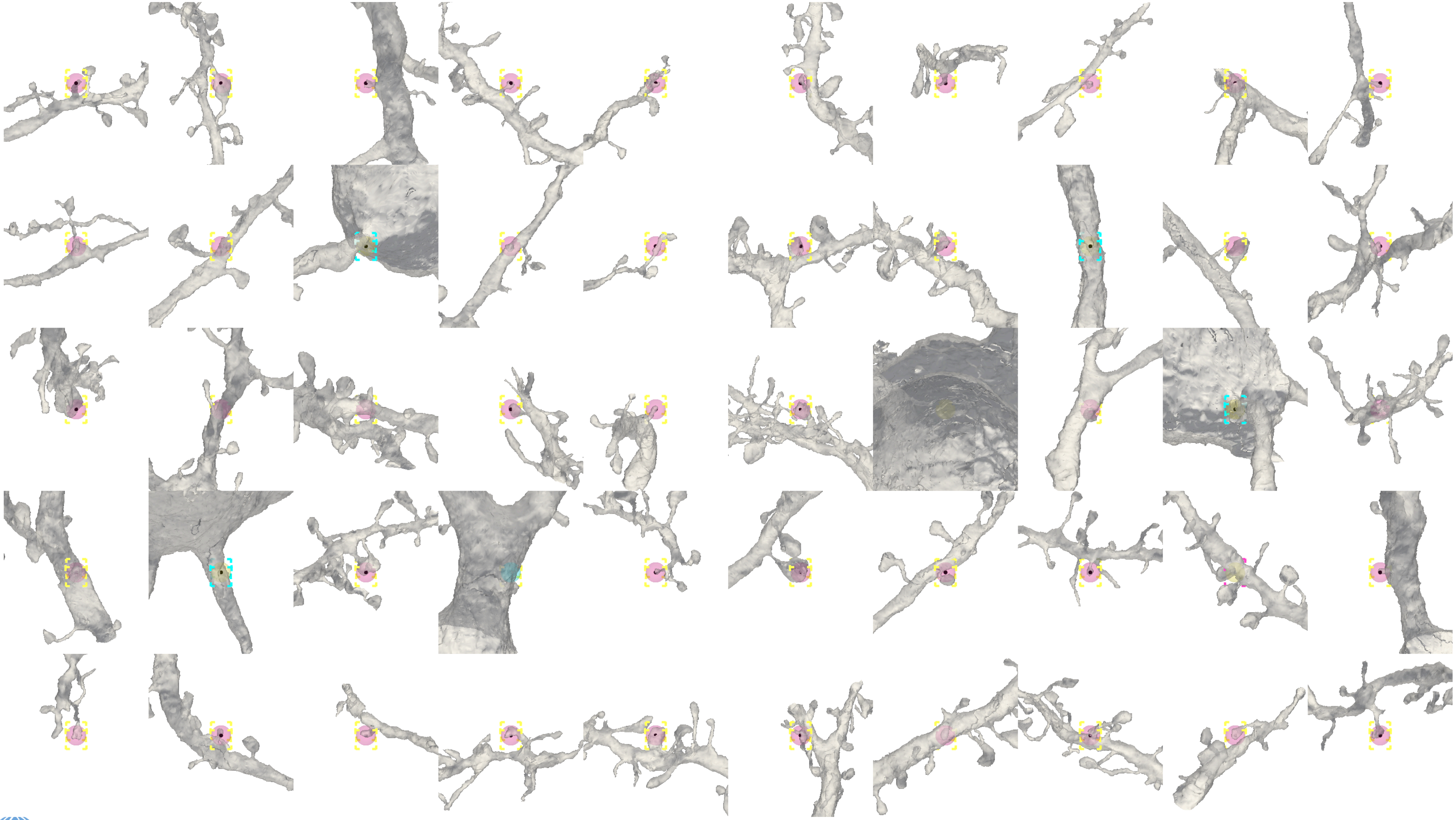
NEURD classifies many spines as shaft

Bethanny Danskin, Erika Neace, Rachael Swanstrom

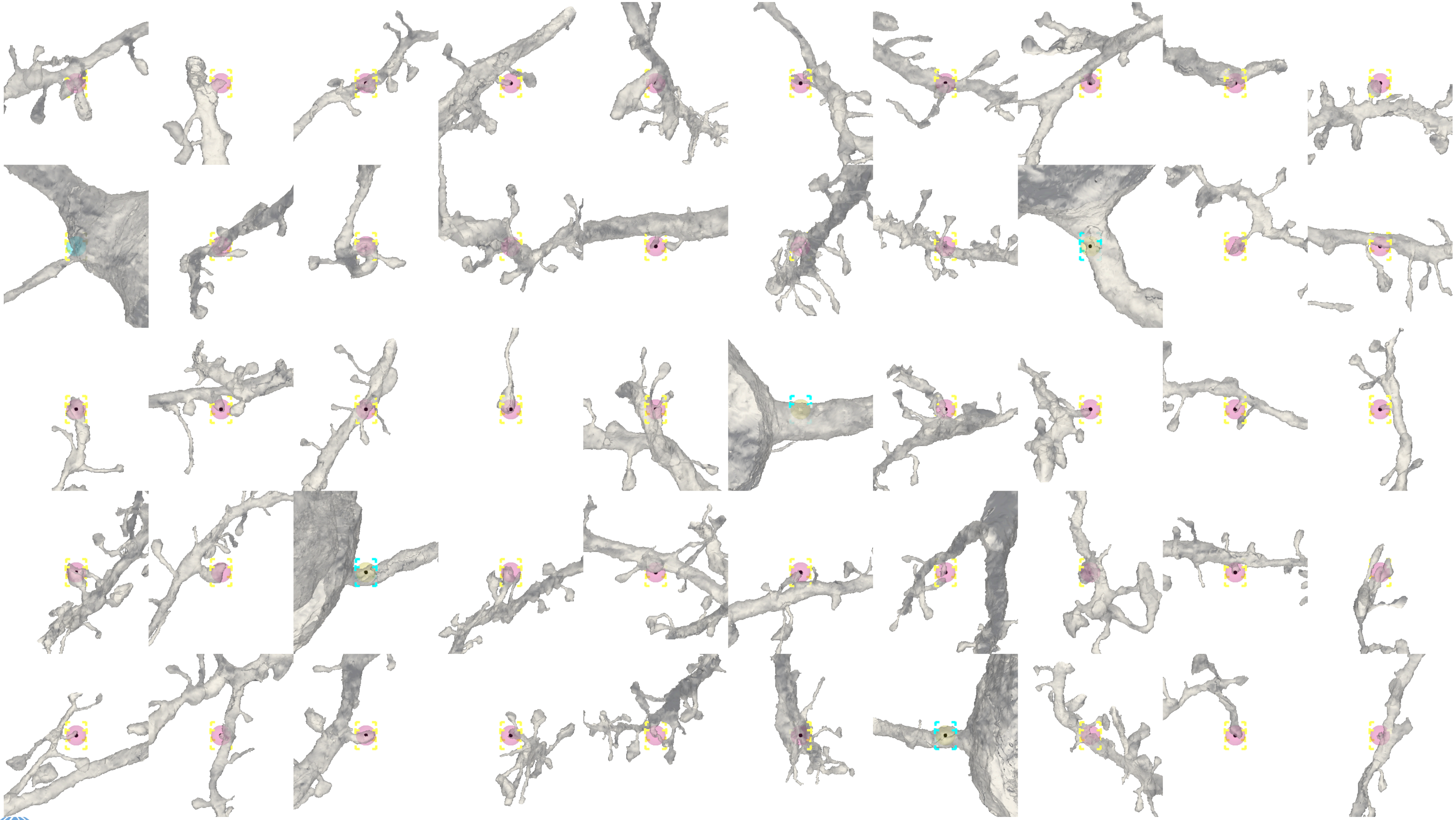


Coverage: 66% of VORTEX compartment labels are in the NEURD table

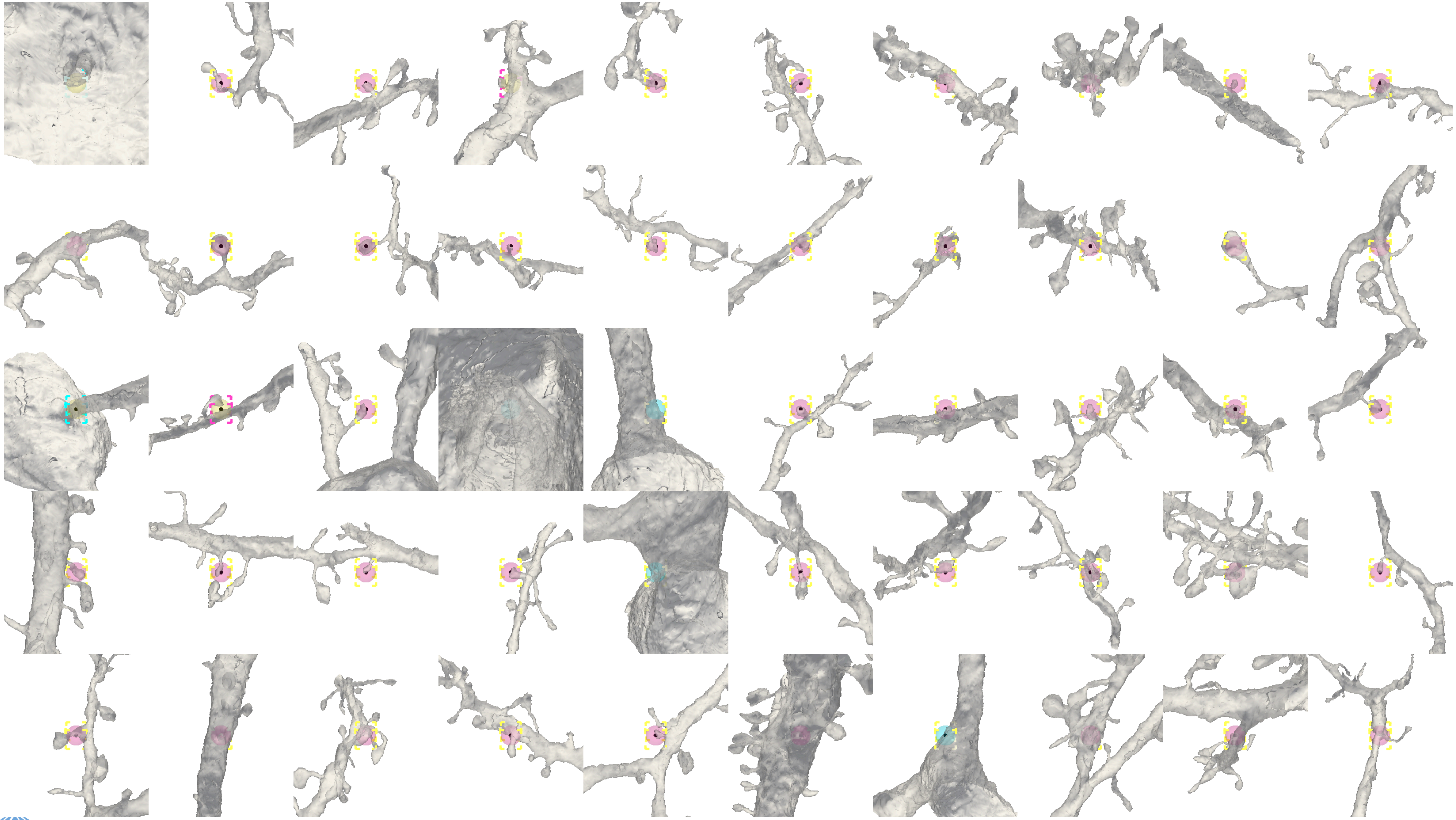
soma shaft spine vortex model



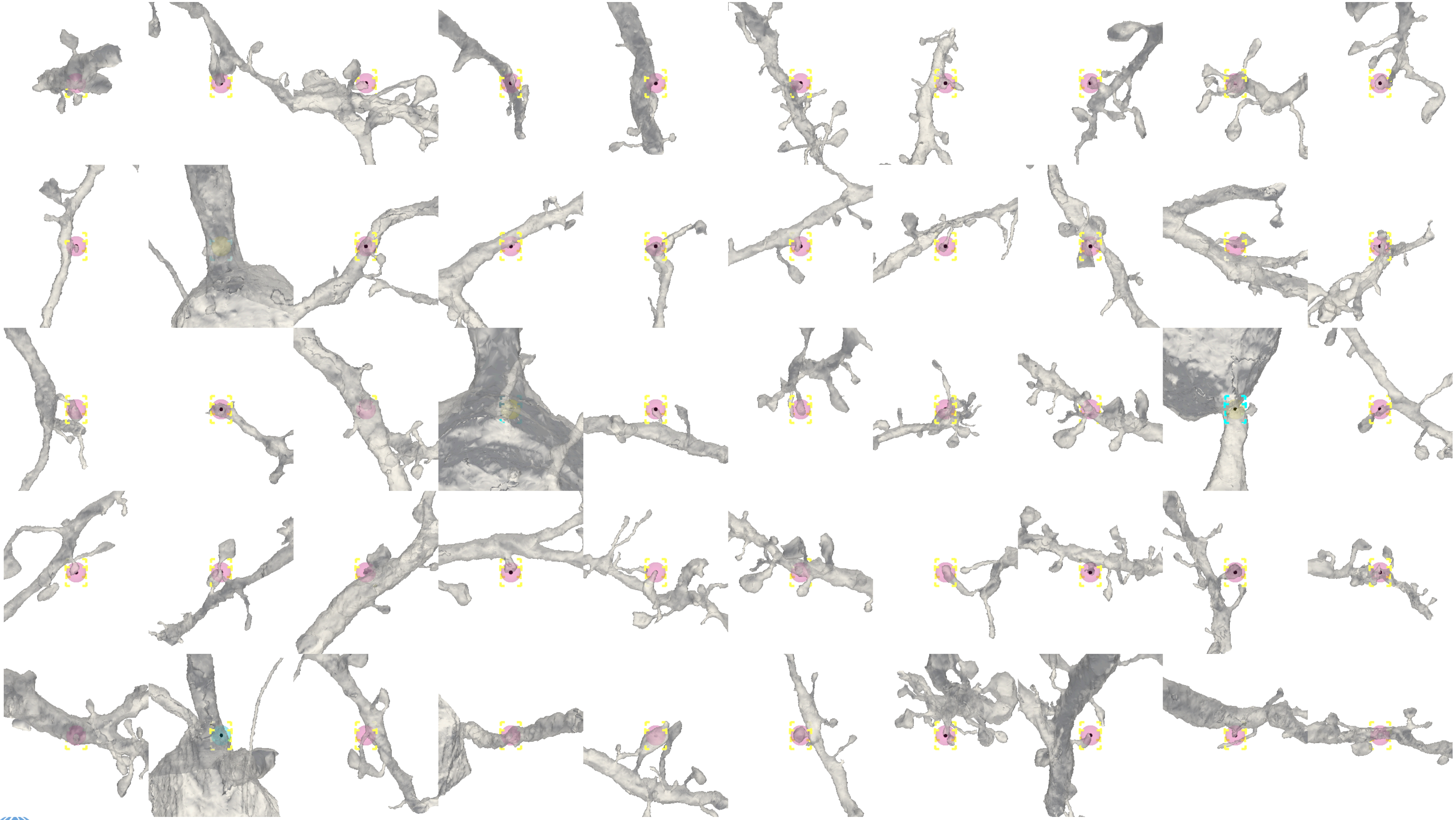
soma shaft spine vortex model



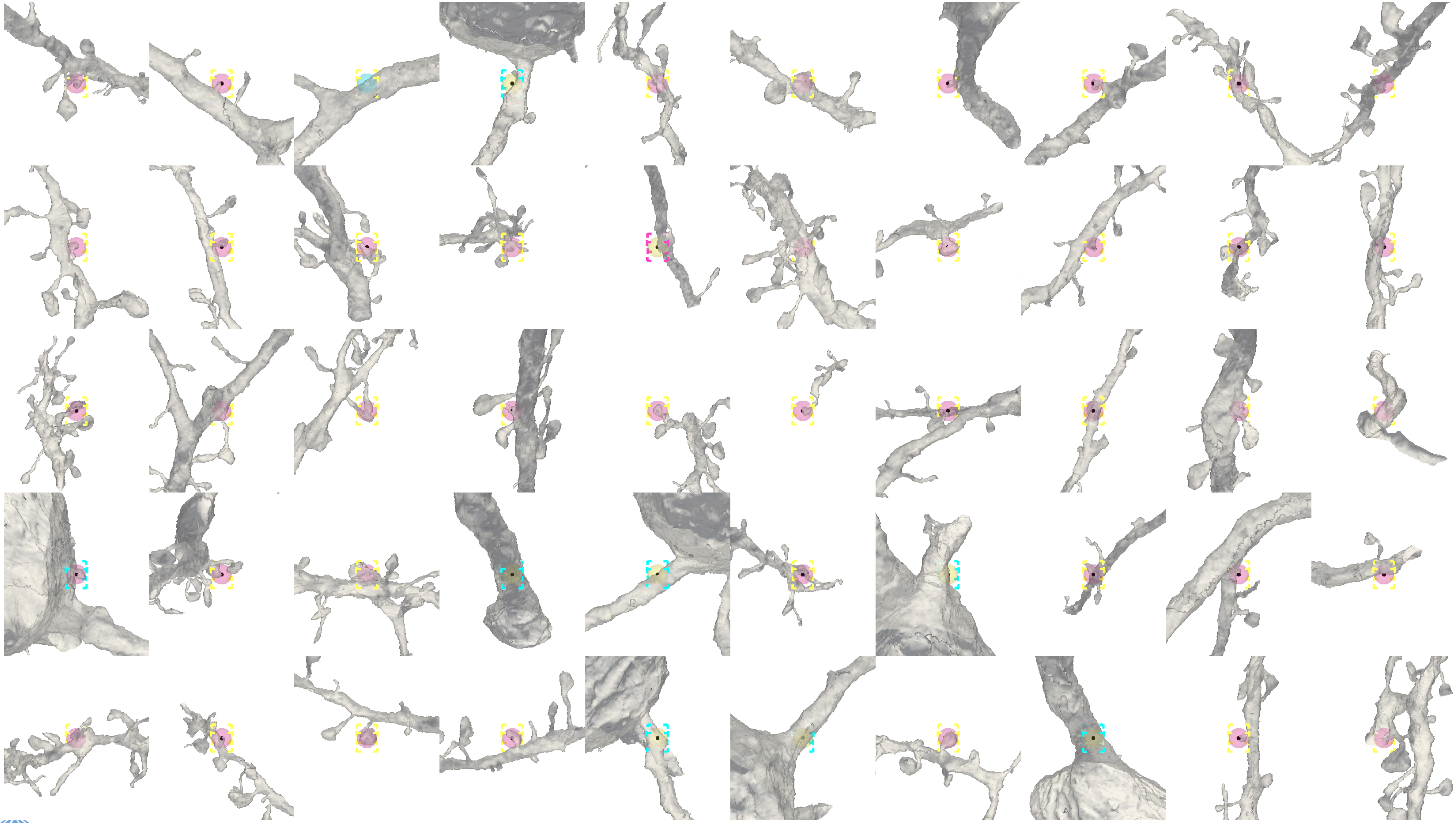
soma shaft spine vortex model



soma shaft spine vortex model



soma shaft spine vortex model



Morphological feature learning

Resolution:

Segmentation/imagery > Mesh > Skeleton

Speed:

Skeleton > Mesh > Segmentation/imagery

How to people do learning on meshes?

Outline

- Motivation
- **Intuition for heat kernel signatures**
- Computing heat kernel signatures
- Application to spine prediction
- Extensions

Eurographics Symposium on Geometry Processing 2009
Marc Alexa and Michael Kazhdan
(Guest Editors)

Volume 28 (2009), Number 5

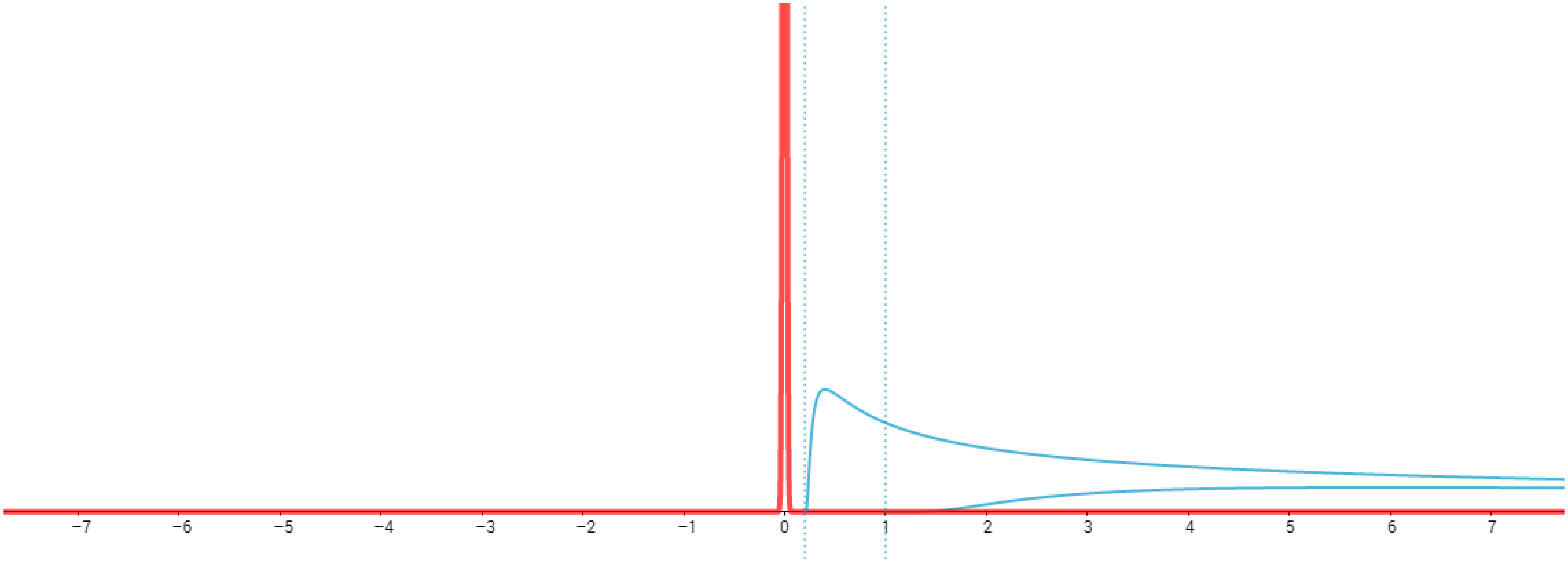
A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion

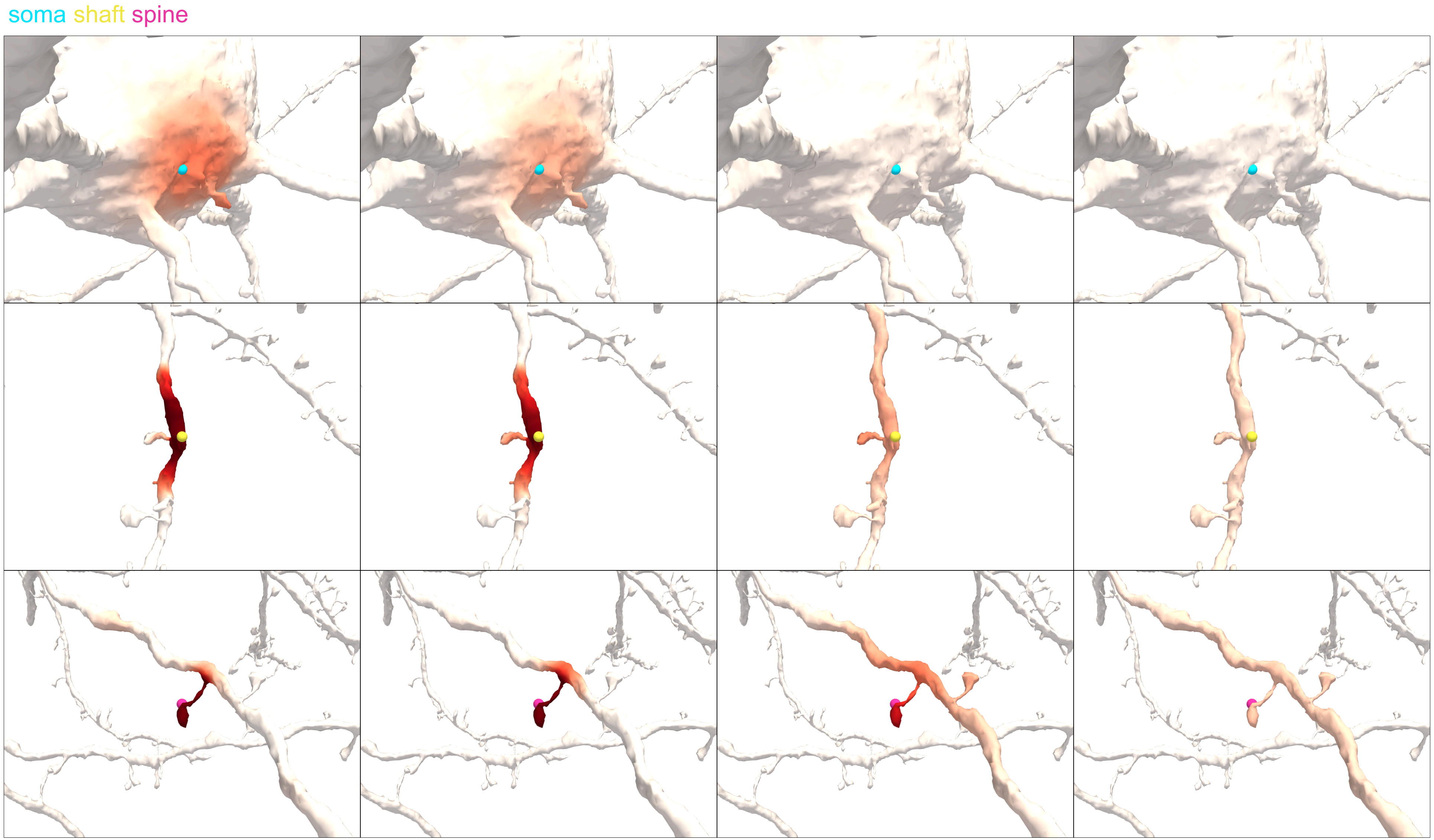
Jian Sun   Maks Ovsjanikov   Leonidas Guibas

Stanford University

Heat diffusion

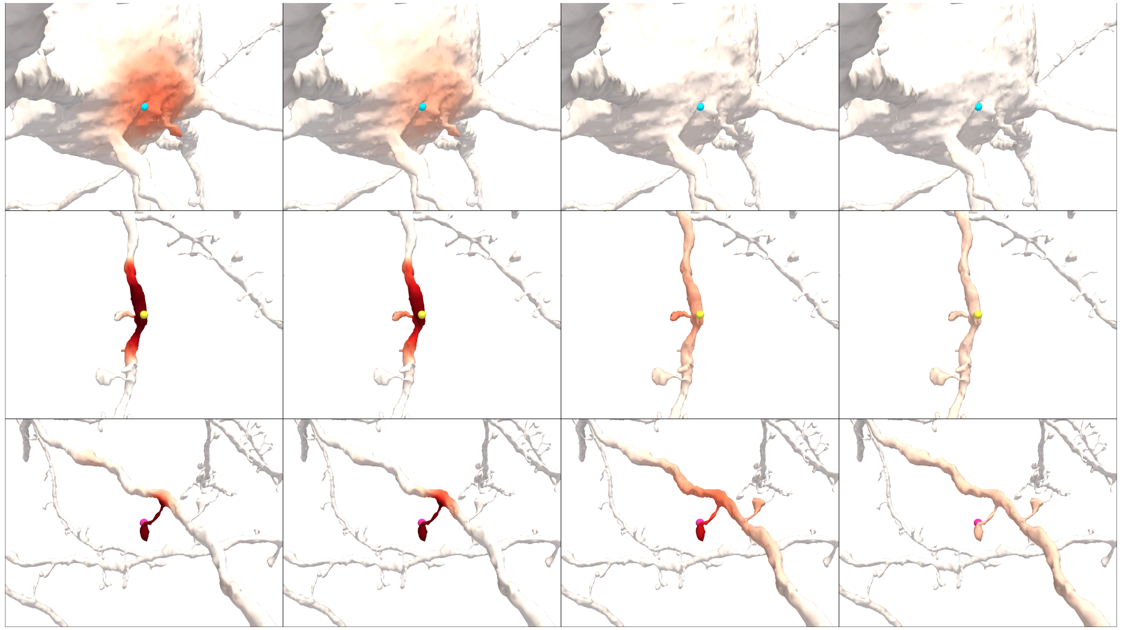
Imagine placing a unit of heat at a point on a surface, watching how that heat diffuses

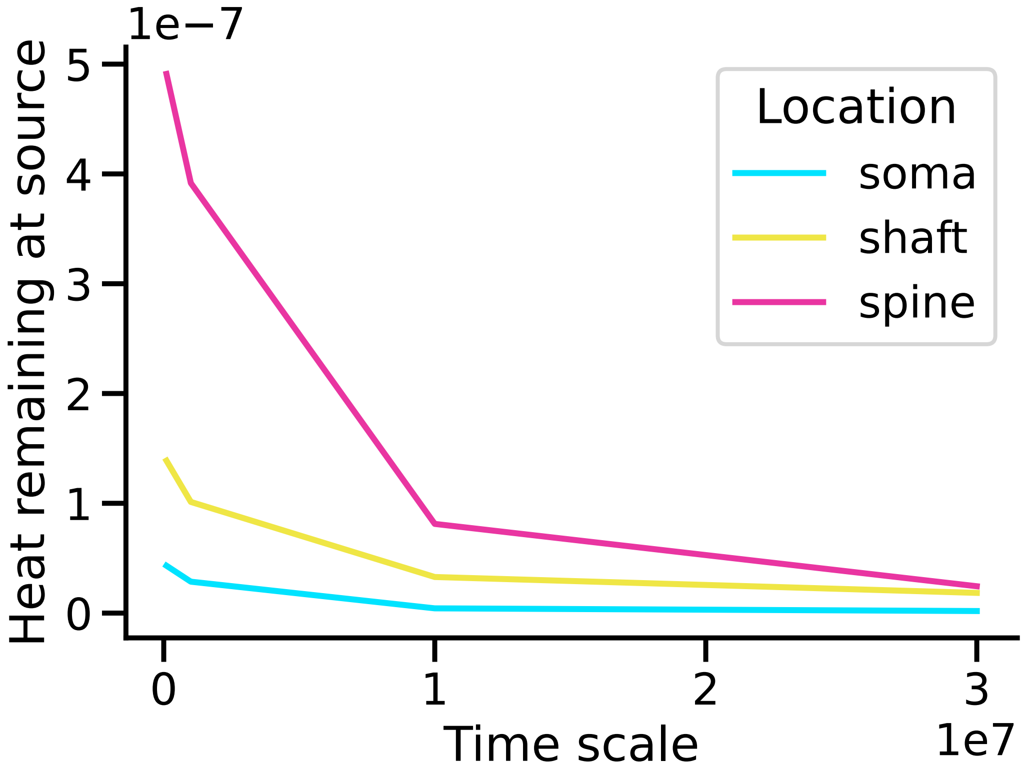




Increasing time →

Tracking heat diffusion





Defining the heat kernel signature (HKS)

$k_t(x, y)$: the amount of heat that diffuses from point x to point y after time t.

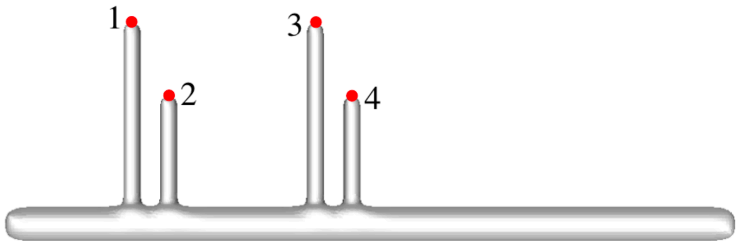
Consider $k_t(x, x)$: how much heat is left at x after some amount of time t.

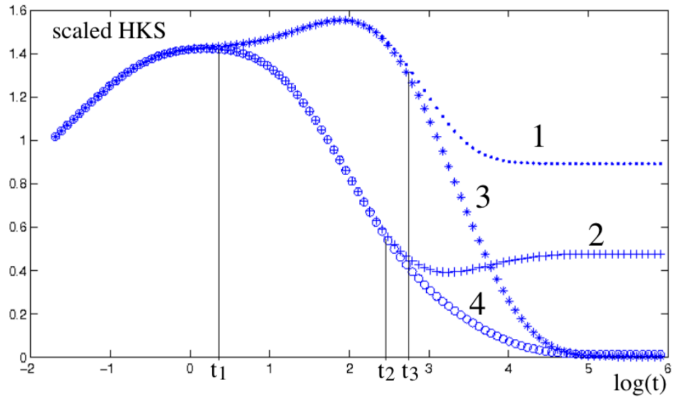
For timescales $T = \{t_1, t_2, \ldots t_d\}$, the HKS for a point on the mesh x is

$$HKS(x) = [k_{t_1}(x, x), k_{t_2}(x, x), \ldots, k_{t_d}(x, x)]$$

Often scale these: $\frac{k_{t_1}(x,x)}{\sum_i k_{t_1}(i,i)}$

Intuition for HKS matching





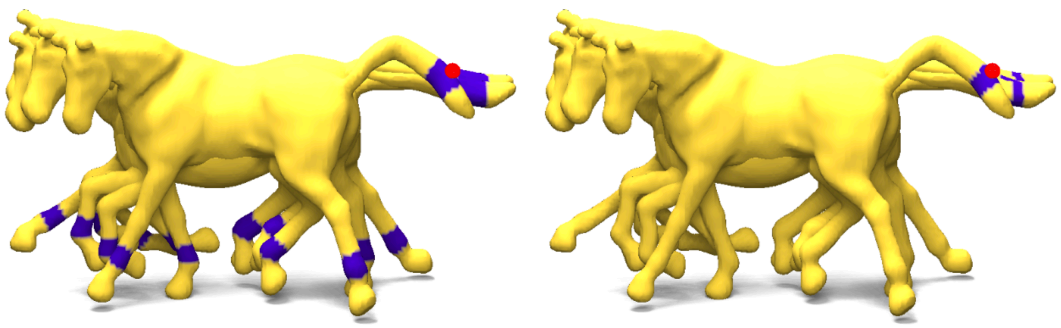
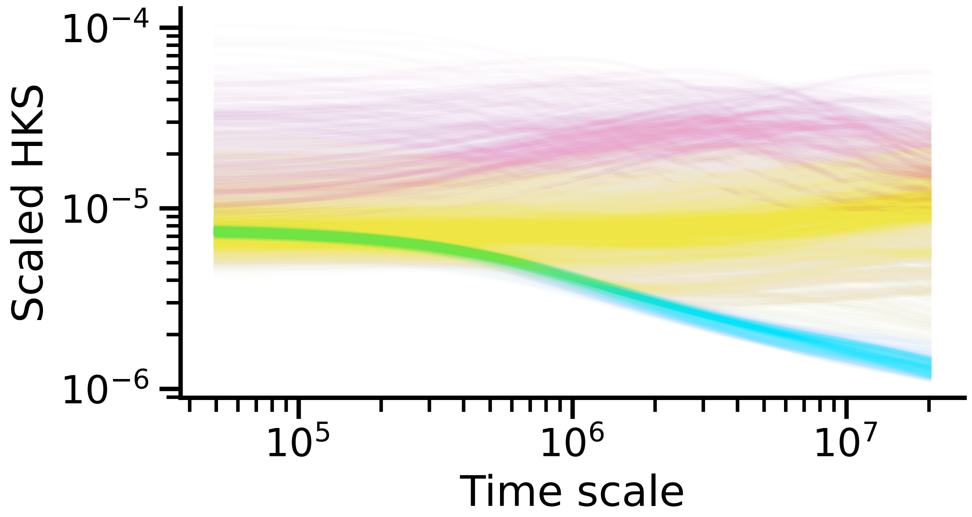
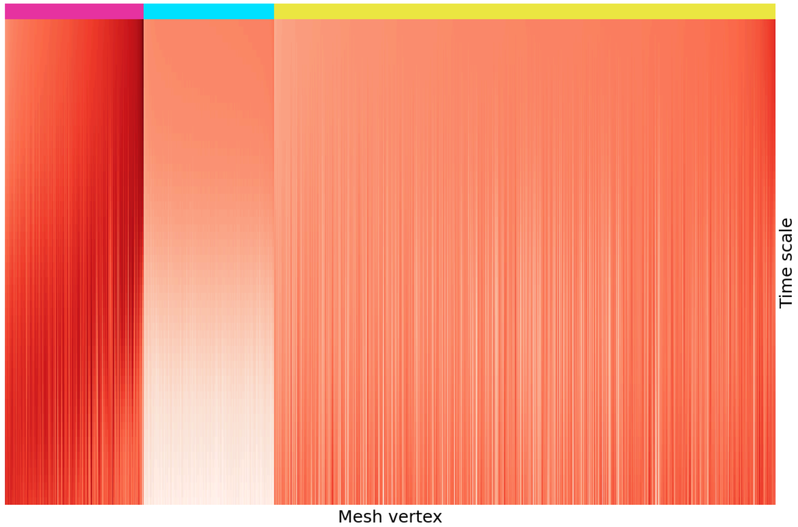


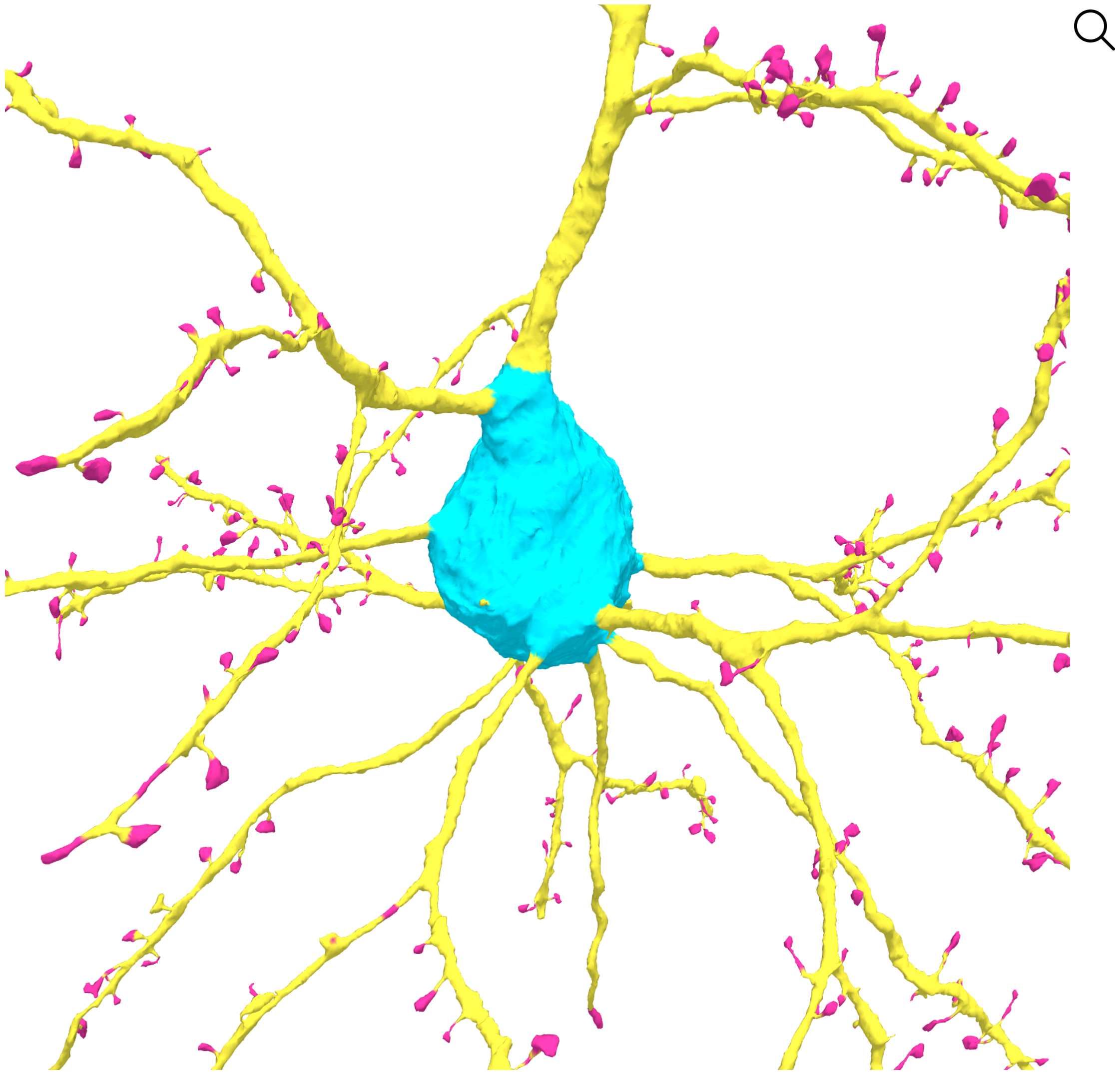
Figure 8: *Four different poses of a horse. left: matching based on half of t's; right: matching based on all t's*

> ...all four points have isometric neighborhoods at small scales, their HKS's are the same for small t's ($< t_1$).
>
> ...Point 1 and point 3 have isometric neighborhoods at middle scales and thus their HKS's coincide even for middle t's ($[t_1, t_3]$)...

Clustering on heat kernel signatures







Outline

- Motivation
- Intuition for heat kernel signatures
- **Computing heat kernel signatures**
- Application to spine prediction
- Extensions

Heat diffusion

Evolution of heat u over time t is governed by the heat equation:

$$\frac{\partial u}{\partial t} = \Delta u$$

where Δ is the Laplacian (2nd derivative) operator.

Heat transferred from point x to y at time t is given by the heat kernel $k_t(x, y)$:

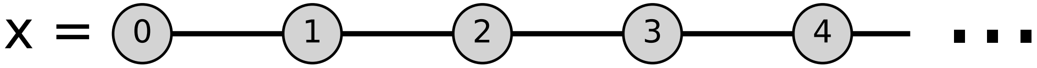
$$k_t(x, y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$

where λ_i and ϕ_i are the eigenvalues and eigenvectors of the Laplacian operator.

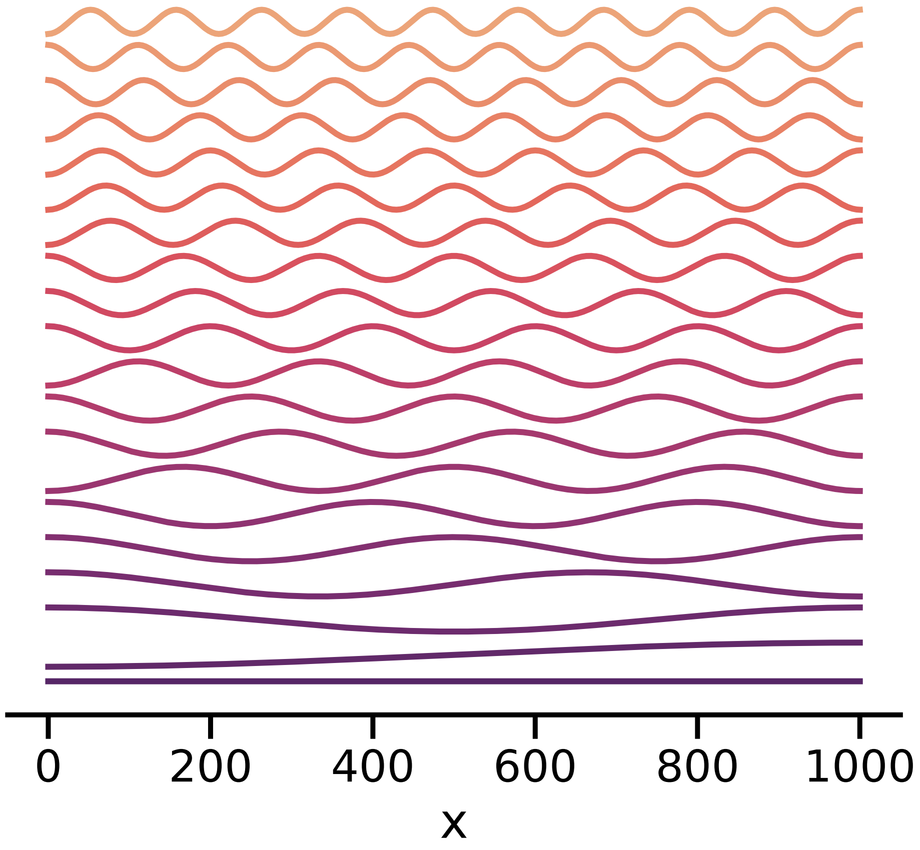
We just need these eigenvectors/eigenvalues to describe heat

Heat on a 1D grid

For a 1D grid,

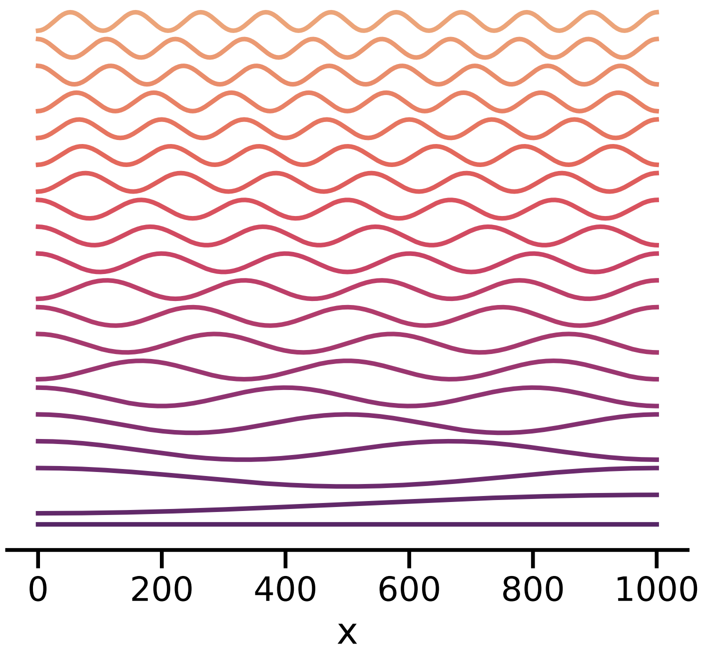


the eigenvectors of the Laplacian are the Fourier series:

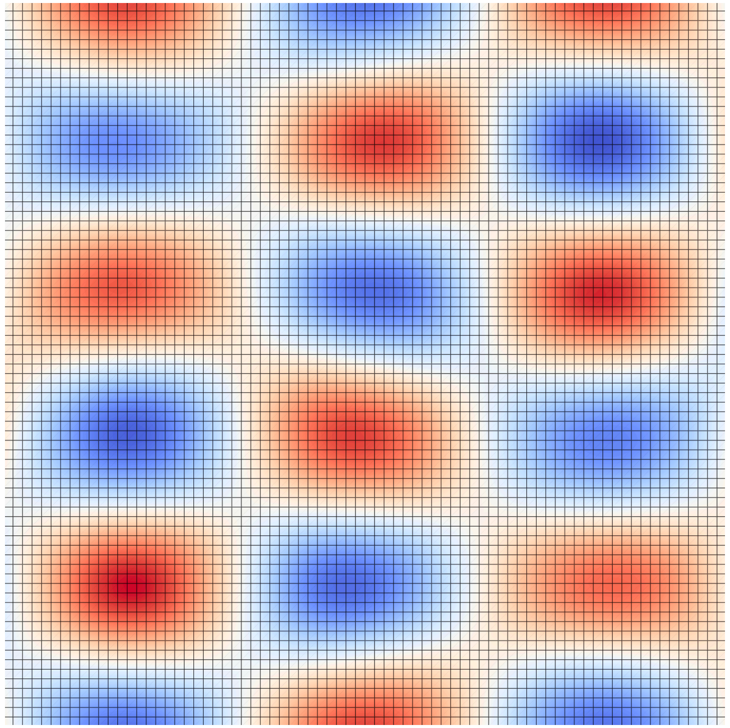


Laplacian eigenvectors

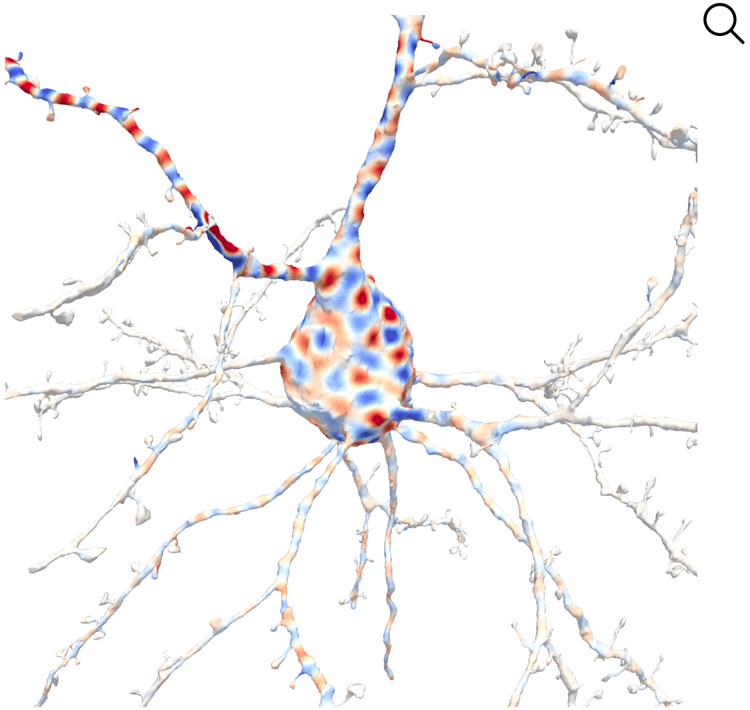
1D grid



2D grid



Mesh



Computing the eigendecomposition

- ✅ Very sparse problem: power iteration methods/ARPACK are efficient
- ✅ Can truncate the eigendecomposition to get an approximate solution
- ❌ Need $O(\text{Thousands})$ of eigenvectors to get resolution down to the scale of spines, mesh has $O(\text{Millions})$ of points
  - Was taking $\sim$ 1-3 Hours to compute eigendecomposition on a single neuron

Improvement #1: band-by-band algorithm

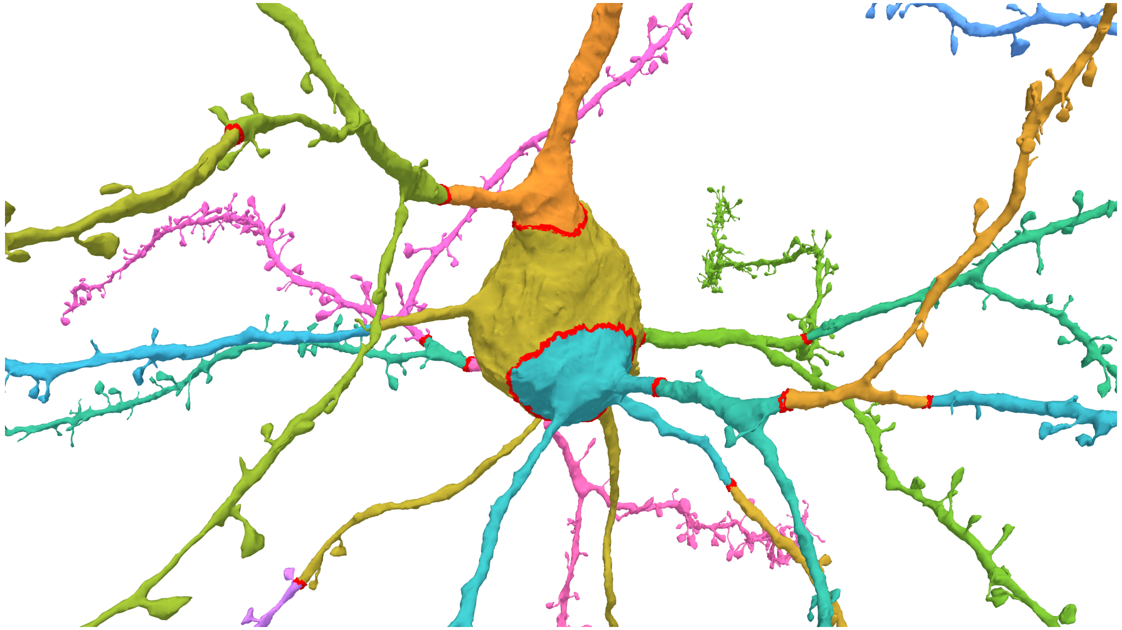
Band-by-band algorithm of Vallet and Levy (2008):

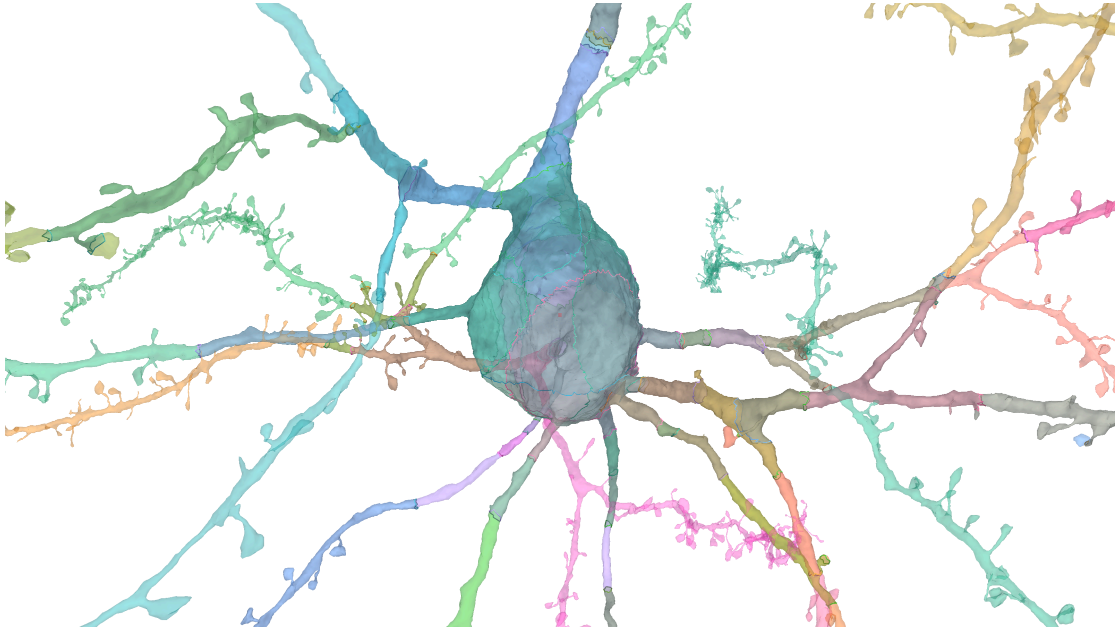
- Use the "shift-invert" trick, do $\tilde{L} = L - \lambda_S I$ for some λ_S
    - Converts the problem to one where power iteration methods are efficient for that range of eigenvalues
- Compute eigenpairs (ARPACK)
- Compute contribution of each eigenpair to HKS, throw away
    - Memory efficient
- Compute a new λ_S, repeat until reach desired eigenvalue

Vallet and Levy, *Eurographics* (2008)

Improvement #2: chunking

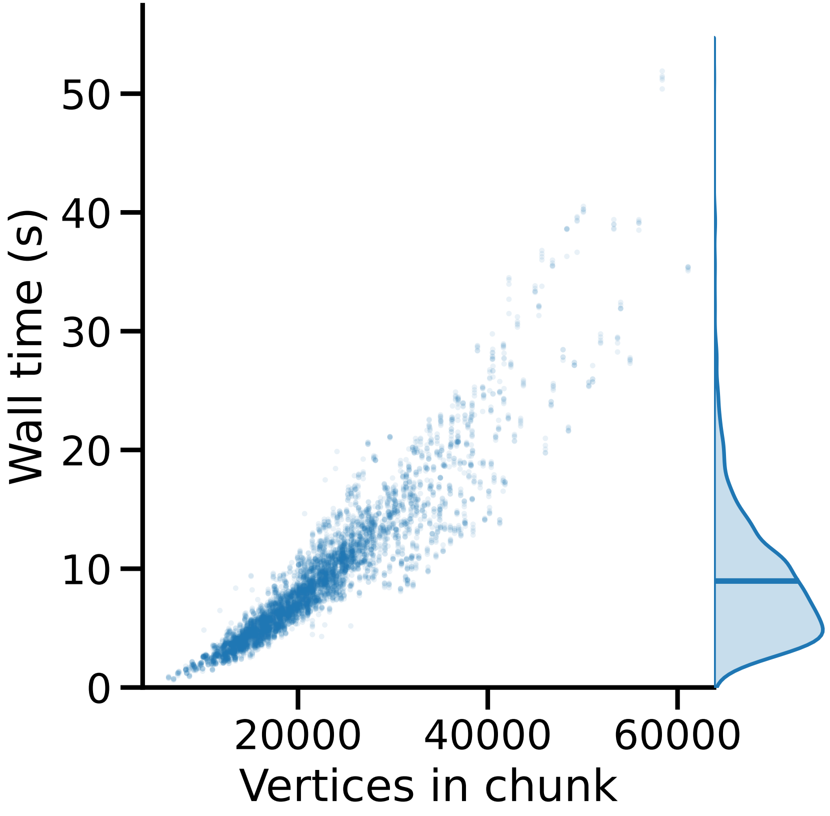
- Intuition: don't need low frequency information to distinguish local features
- Can break the mesh into pieces, compute the eigendecomposition on each chunk
- Use overlapping mesh chunking to minimize edge effects at borders



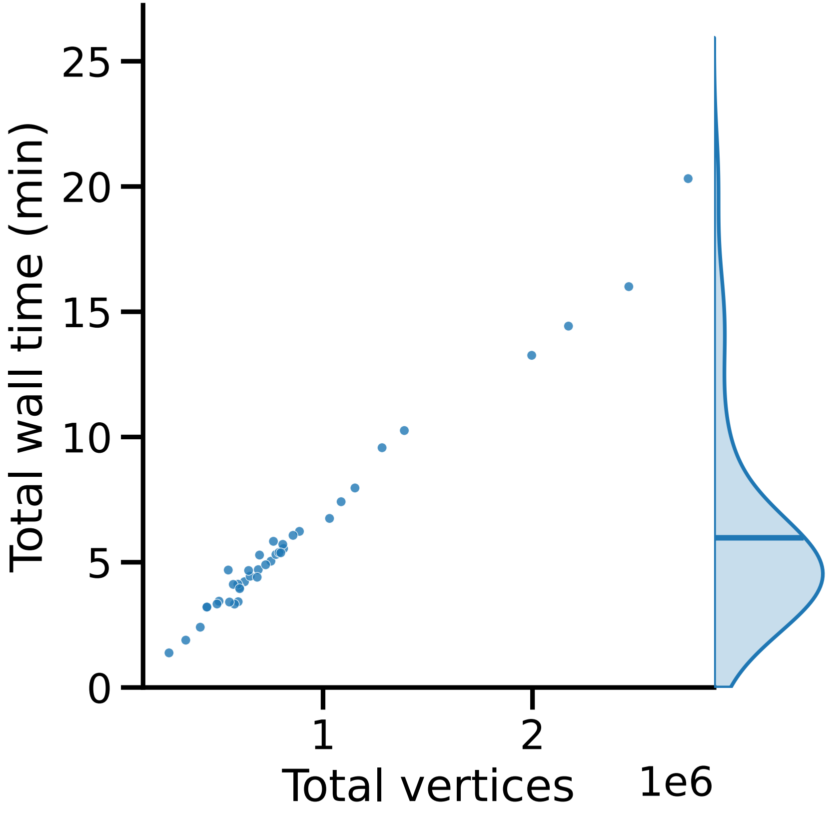


Timing

Per chunk



Whole neuron



$*$ Doesn't include mesh simplification/subdivision, adds $\sim 1 - 3$ minutes per neuron

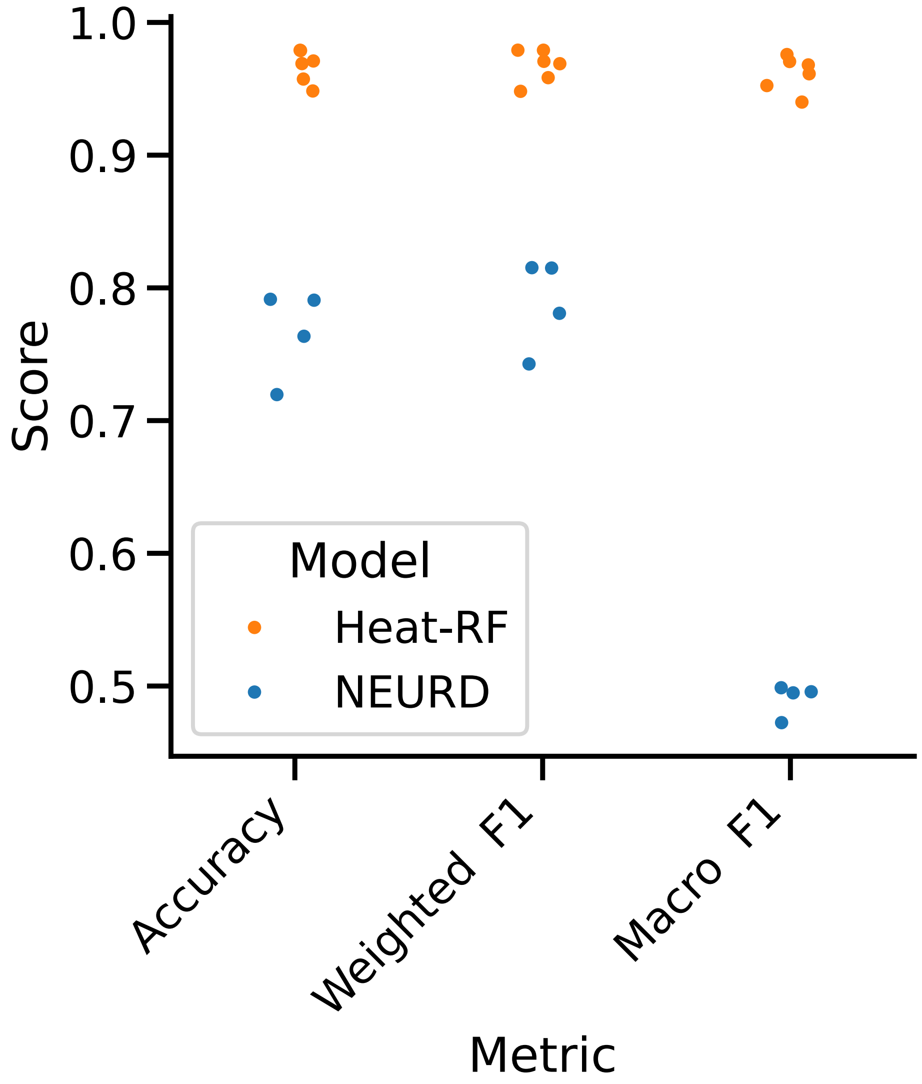
Outline

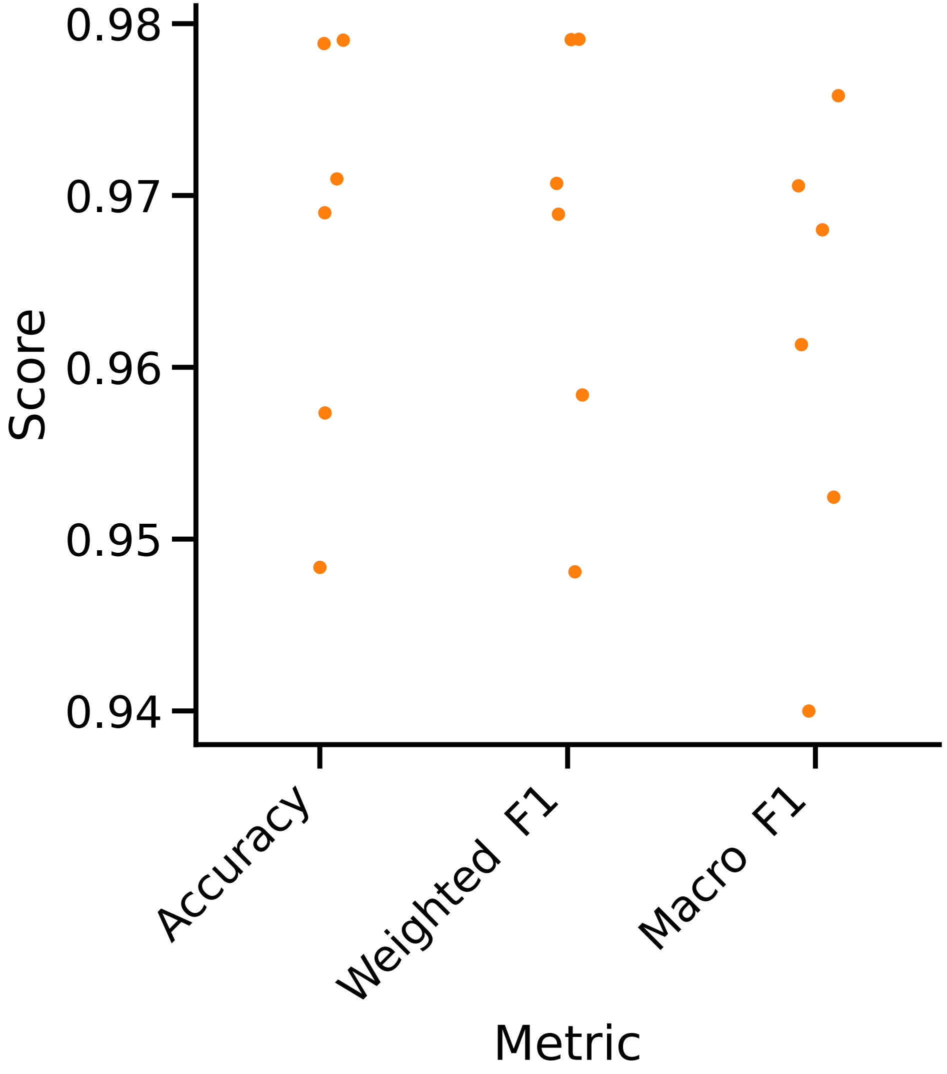
- Motivation
- Intuition for heat kernel signatures
- Computing heat kernel signatures
- **Application to spine prediction**
- Extensions

Spine prediction

- Used dense spine labels for 6 neurons
    - *Bethanny Danskin, Erika Neace, Rachael Swanstrom*
- Trained on HKS features from the mesh point closest to synapse center point
- Used a simple random forest, didn't do much tuning or exploration here
- Didn't try to do anything with the axon, so that gets labeled arbitrarily

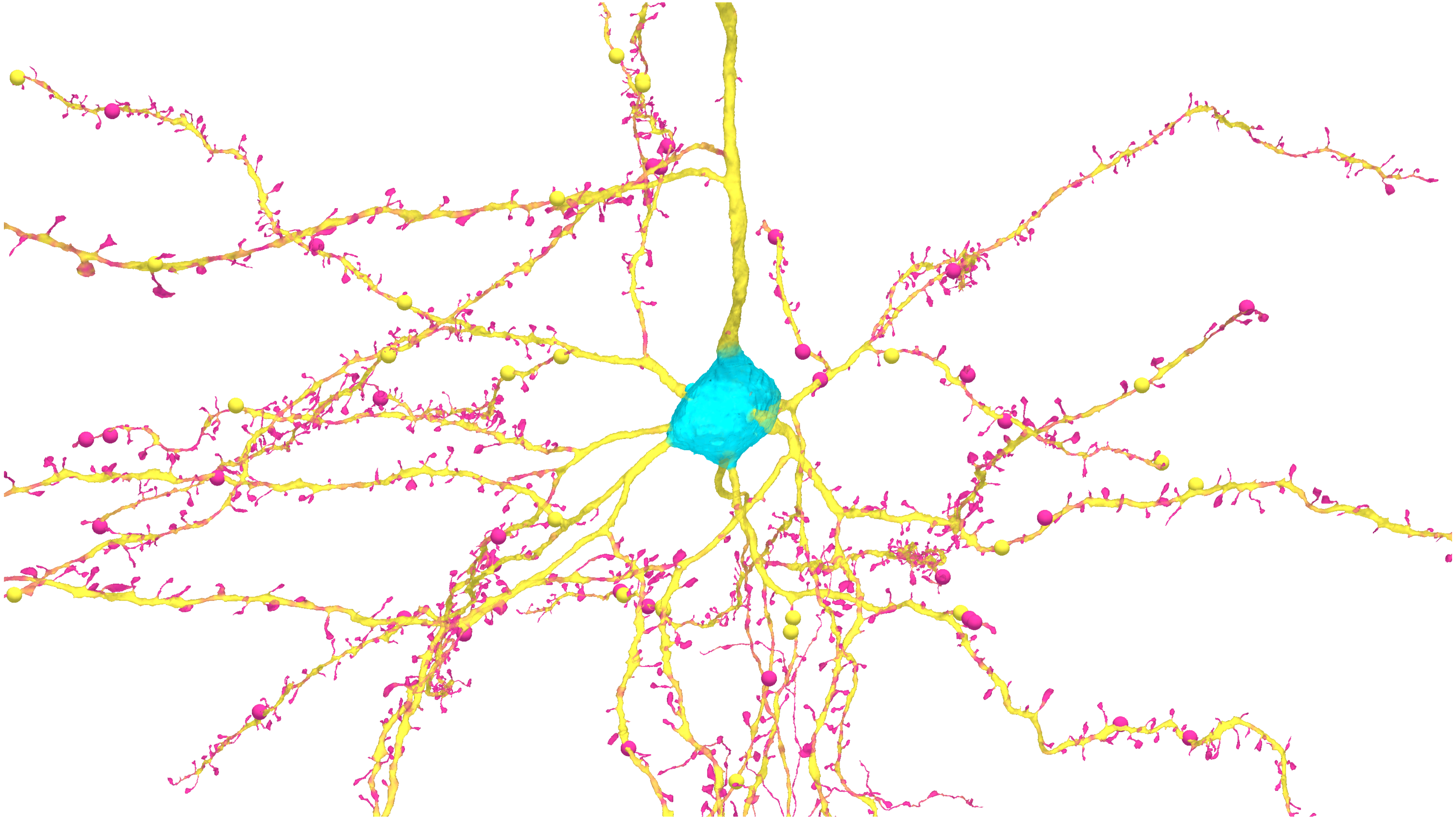
Random forest, leave-one-neuron-out testing



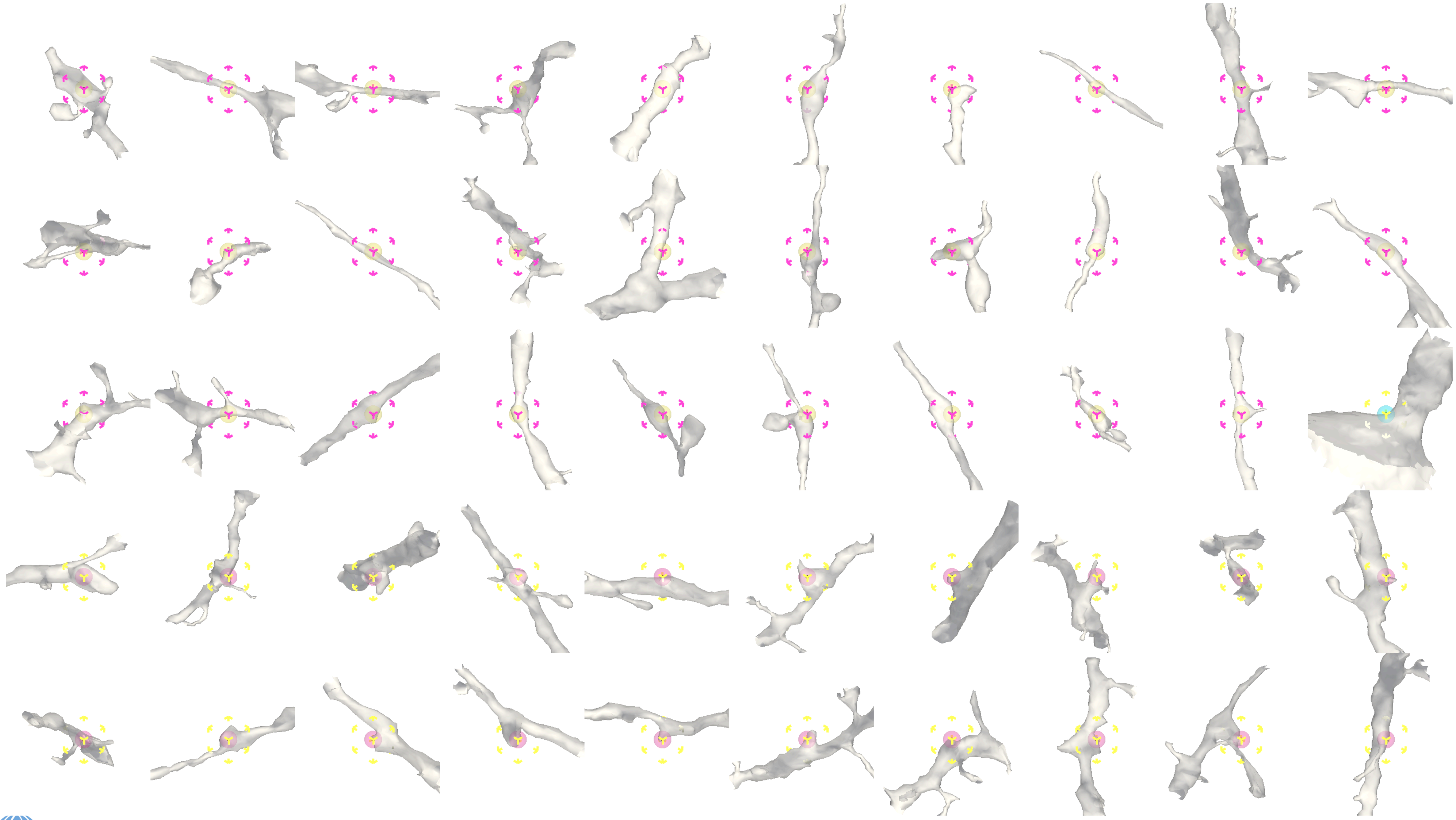


Worst neuron

soma shaft spine vortex model

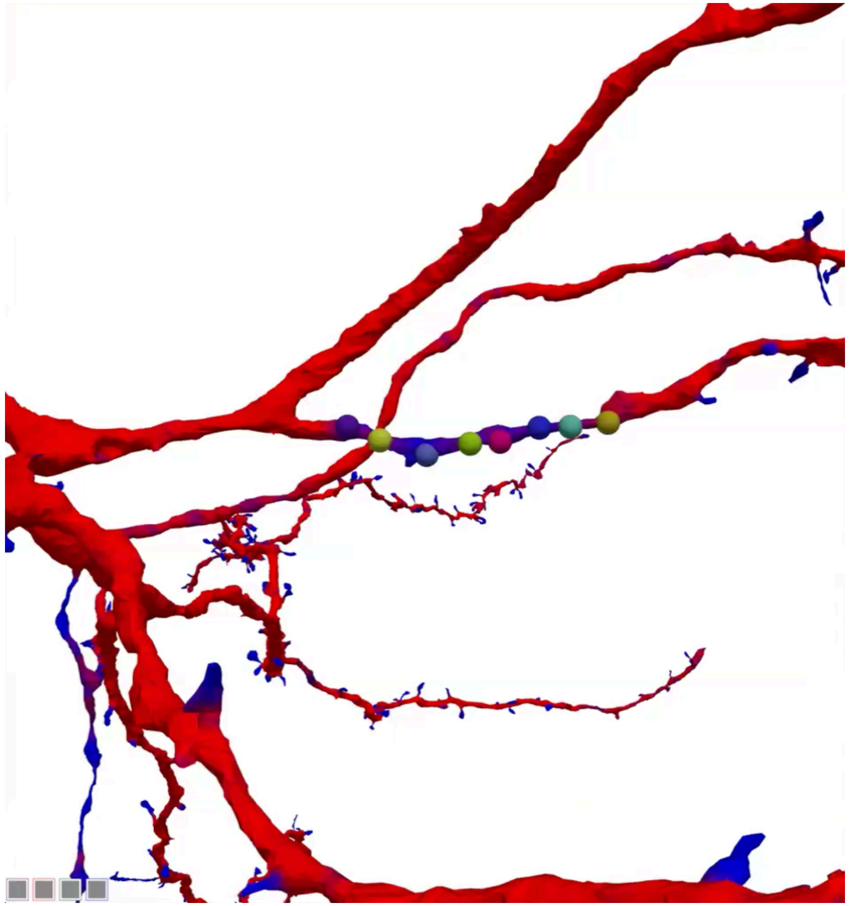


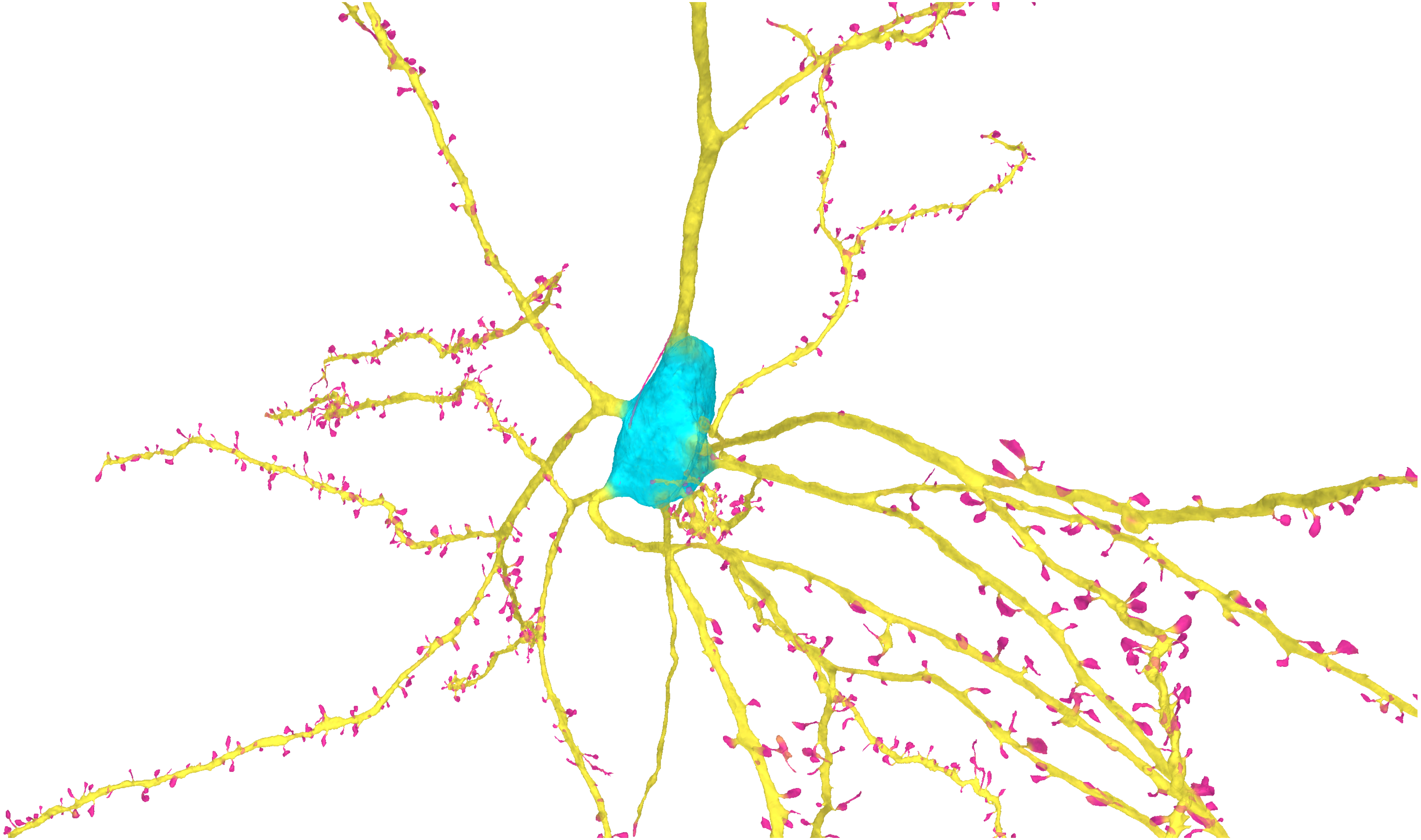
soma shaft spine vortex model

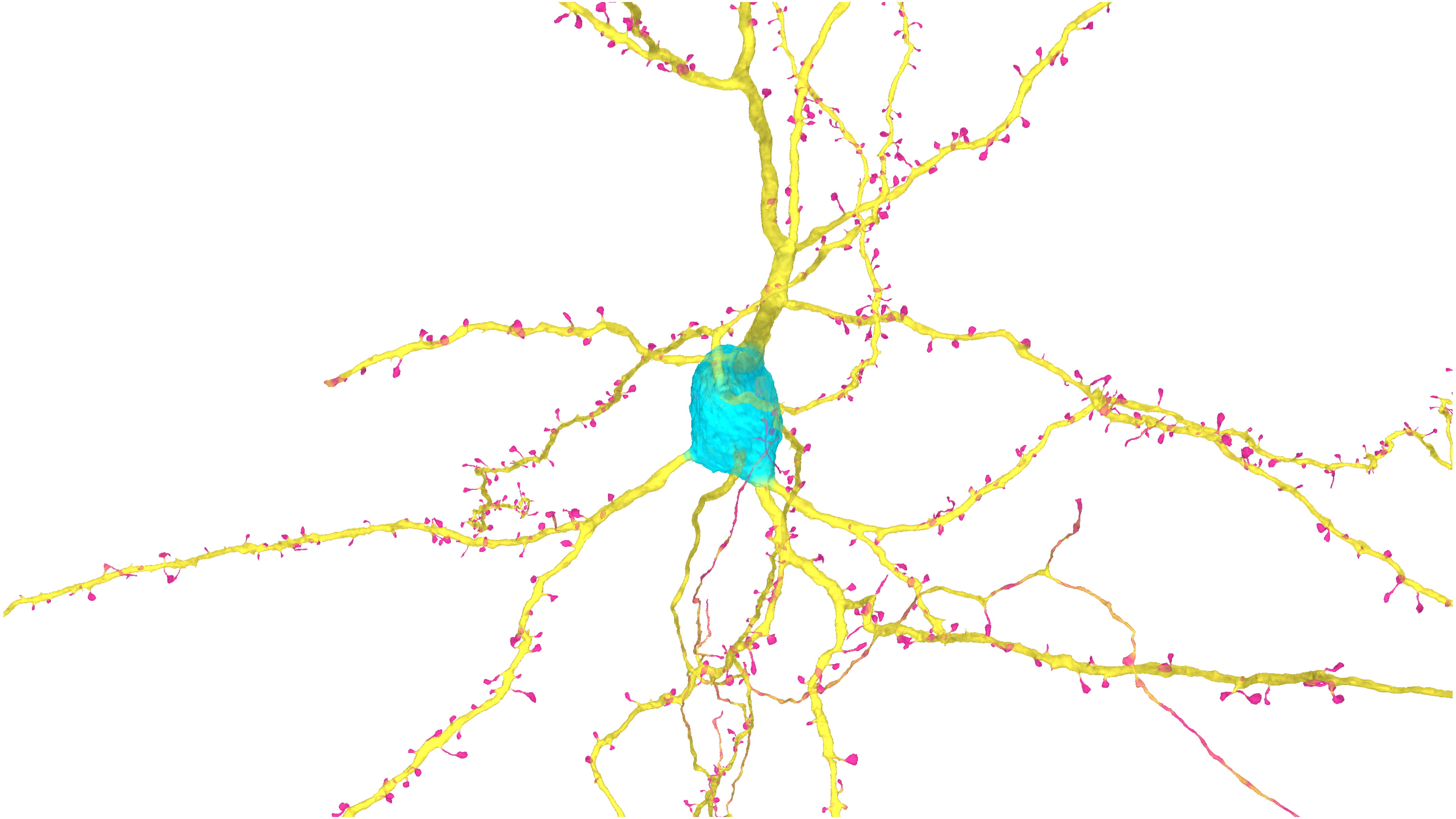


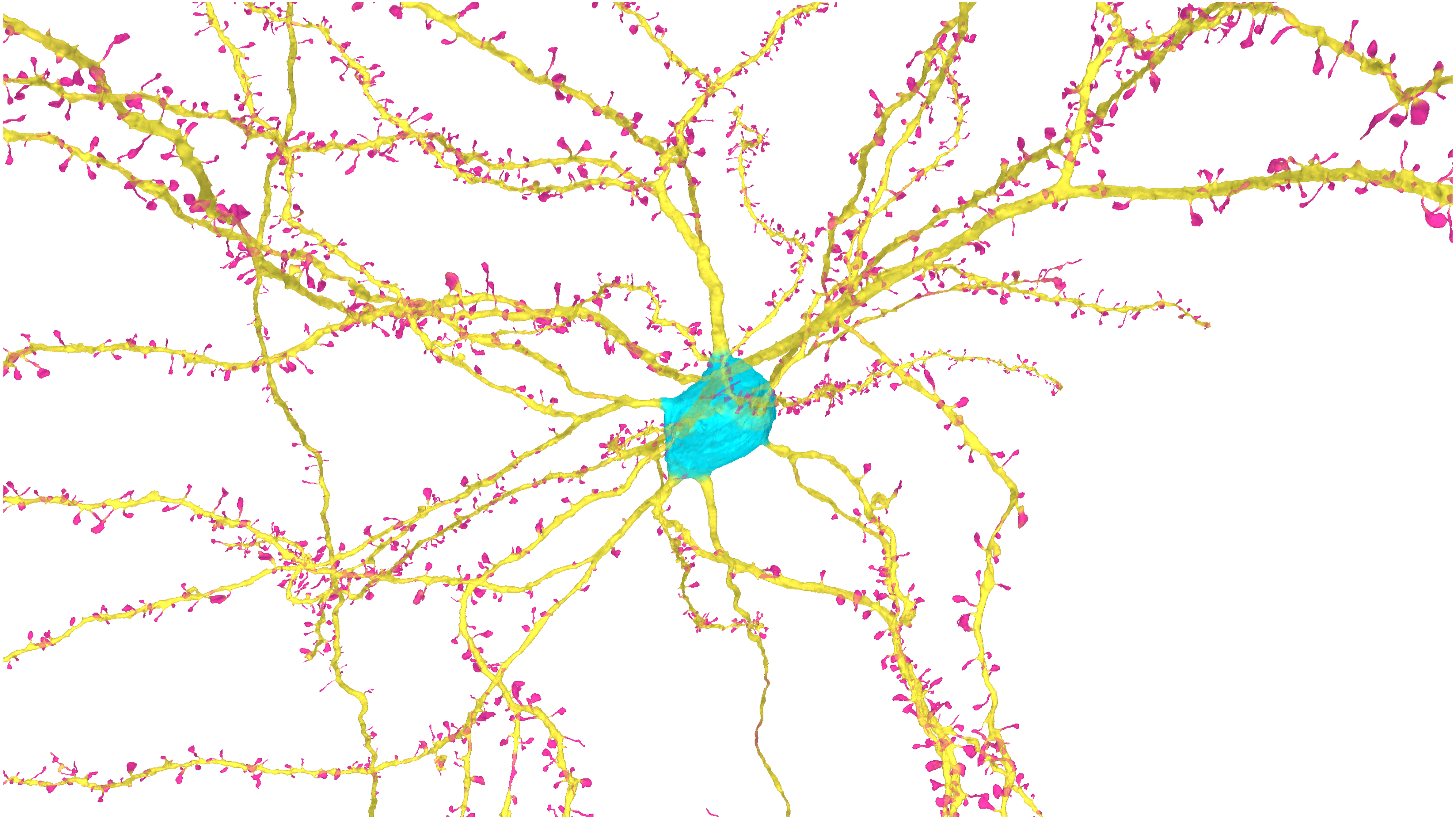
Pseudo-active learning

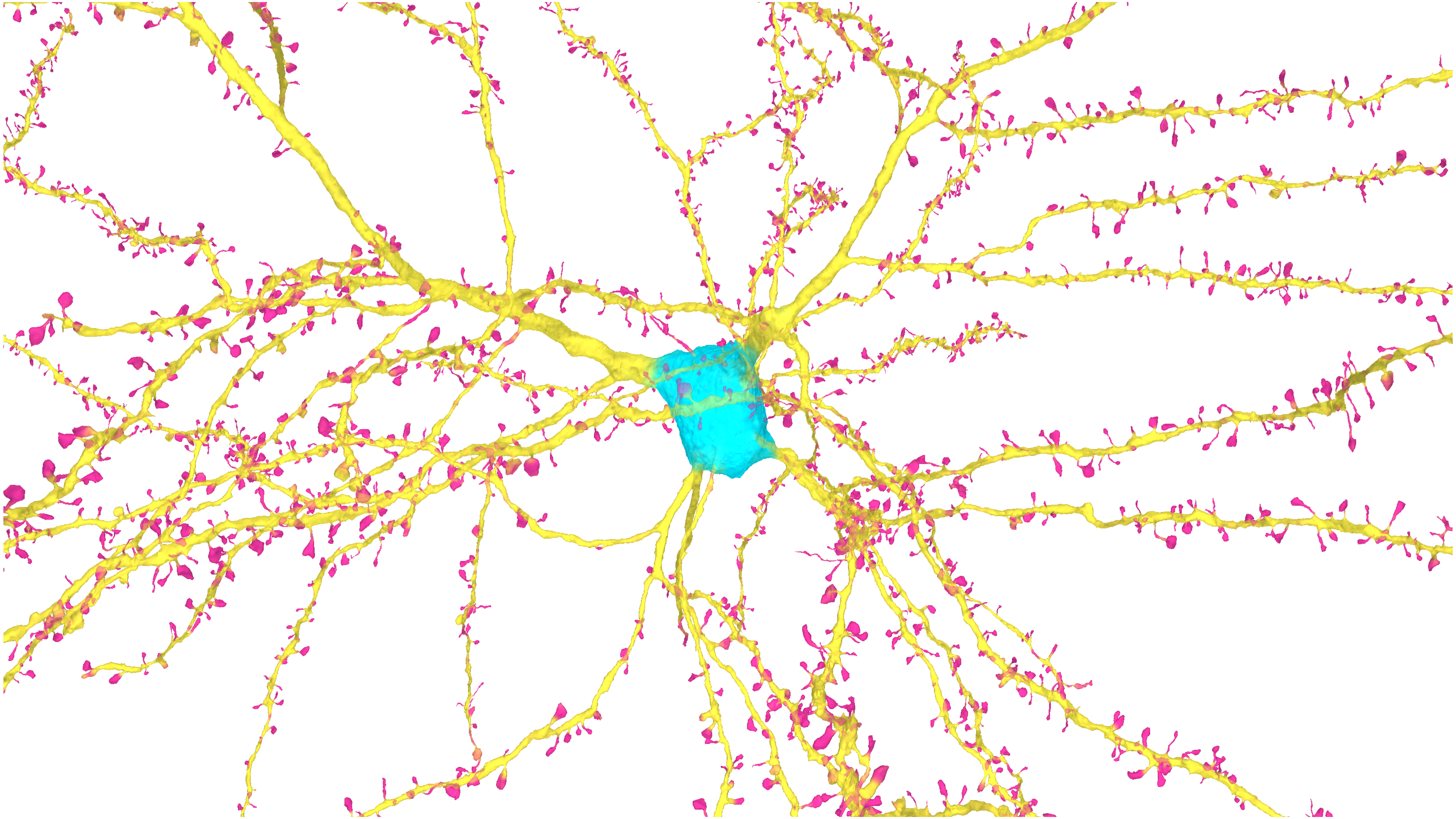
- Used model from these 6 neurons to predict on 20 more neurons
  - These varied more in dendrite thickness and spine density
- Plotted the posterior on the mesh, hand-labeled points I thought looked bad
- Took ~ 2 clicky hours
- Retrain
- Applied model to another 20 more neurons (not the ones trained on)

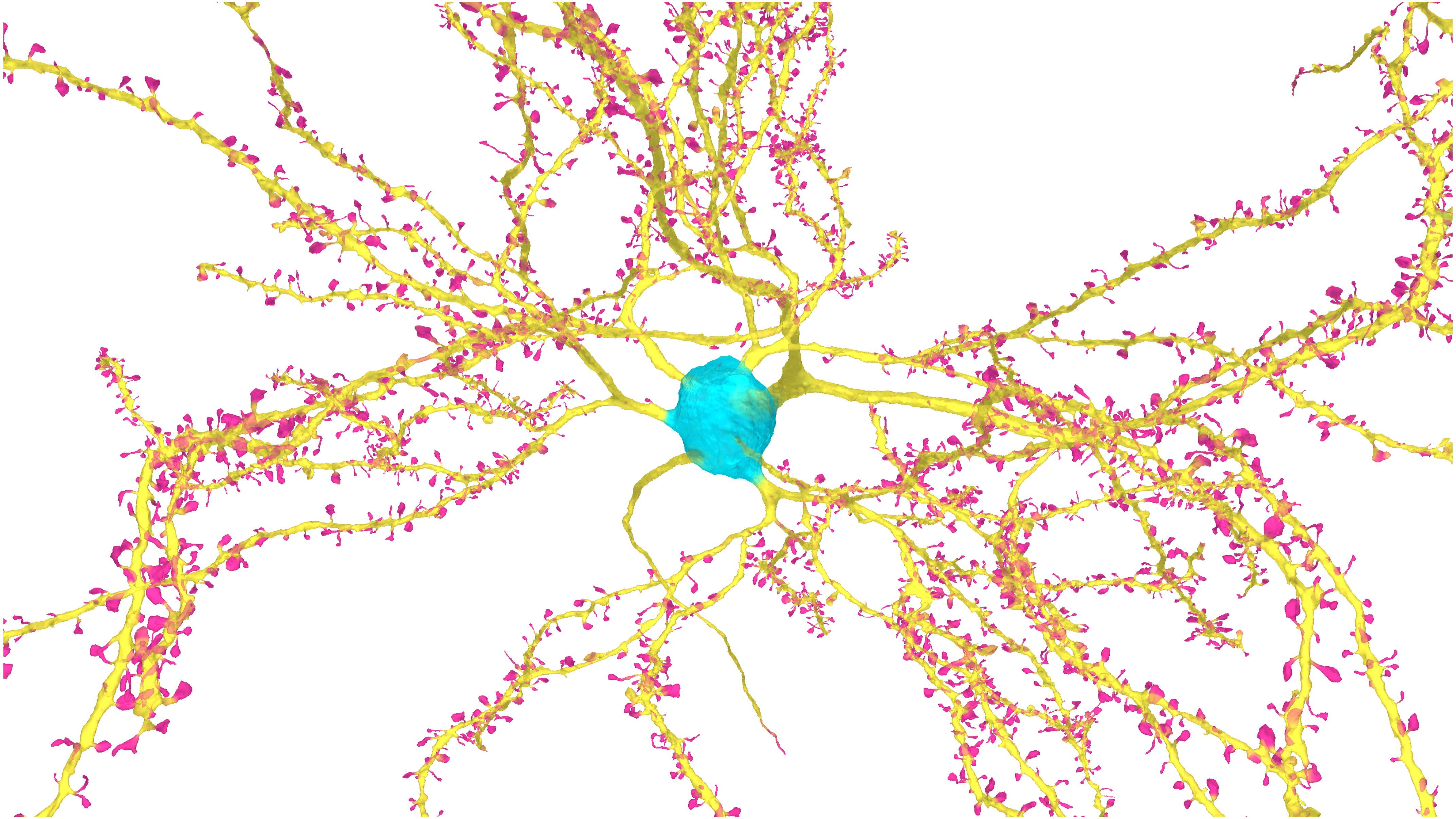


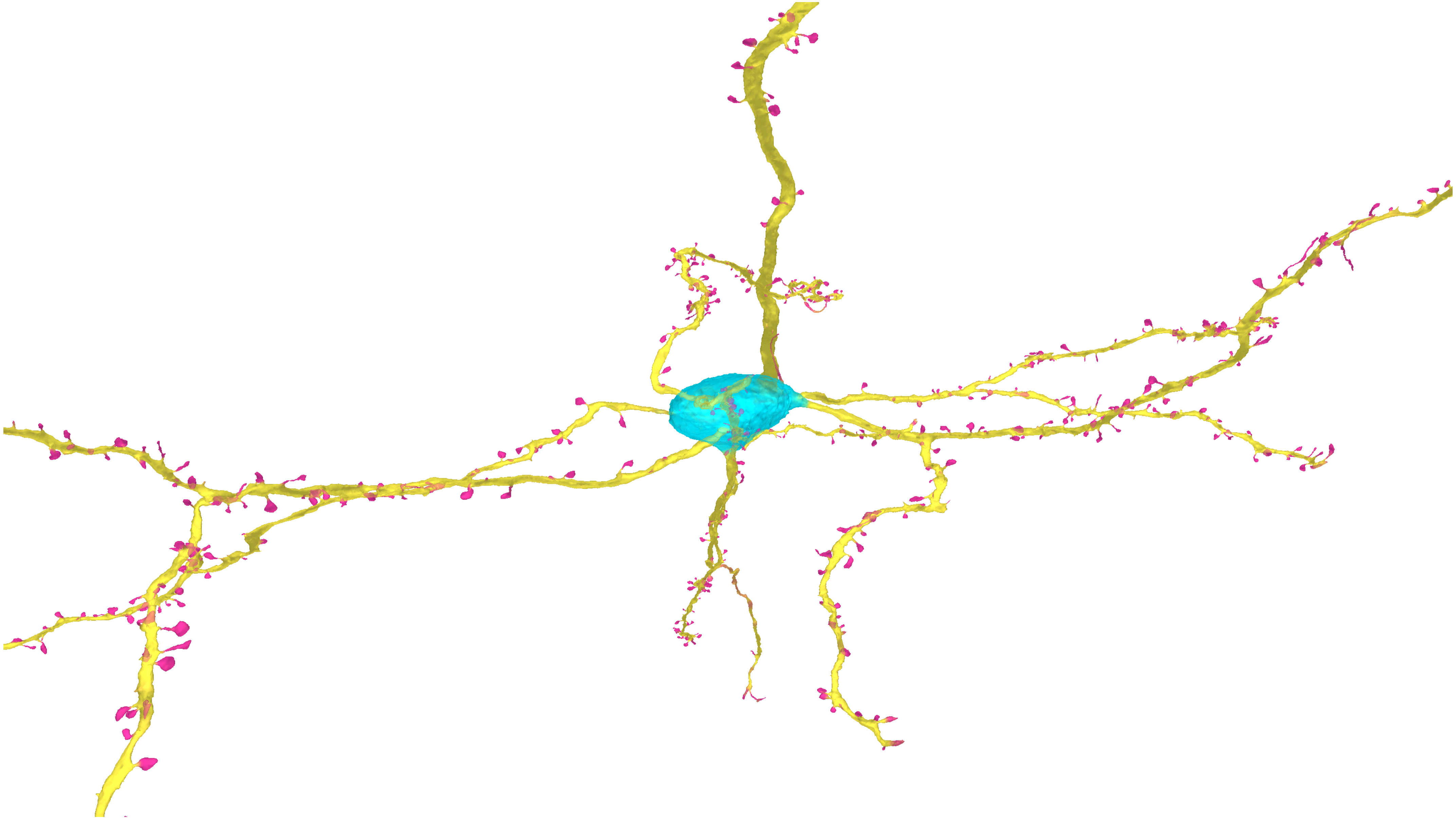


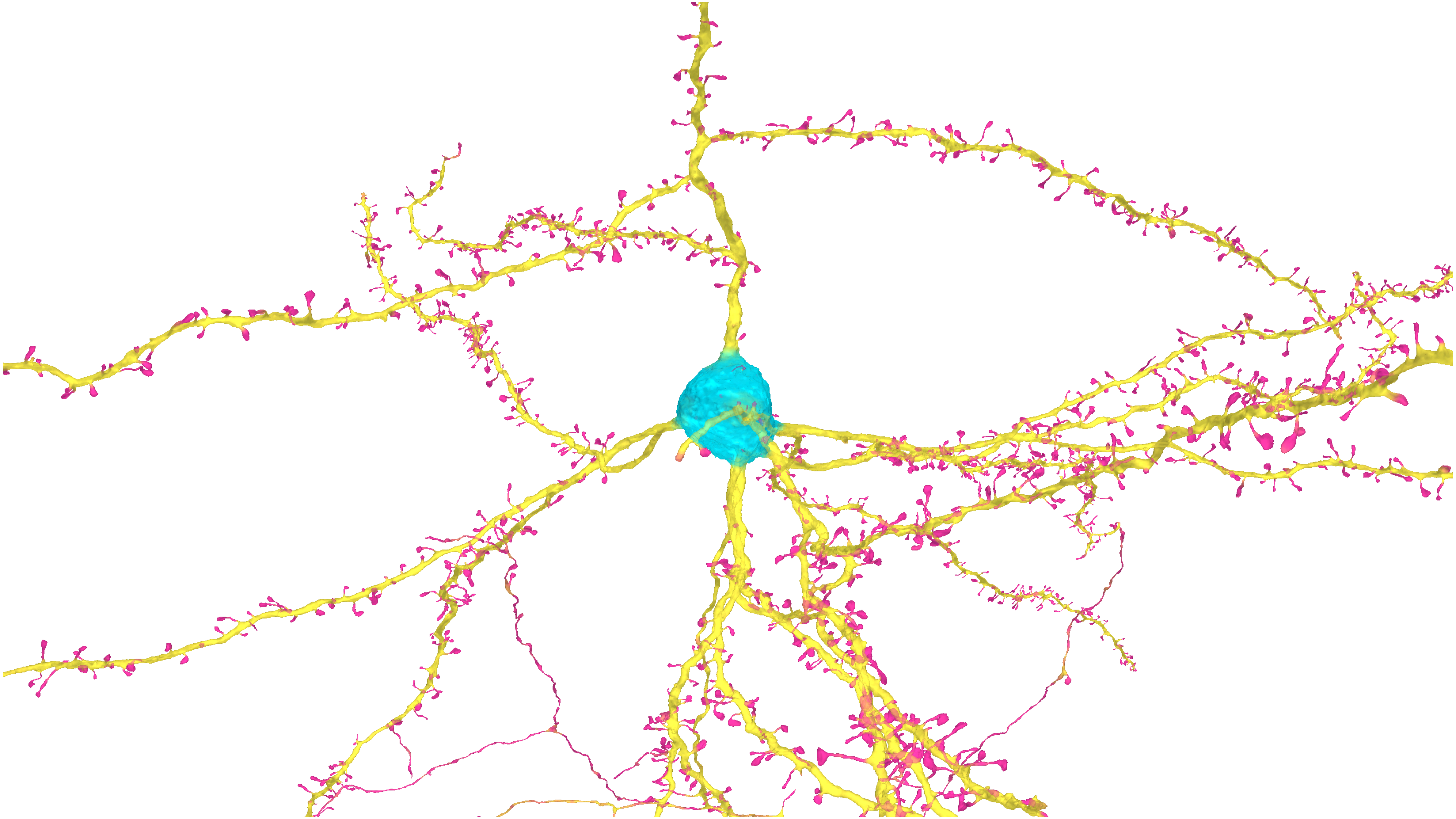


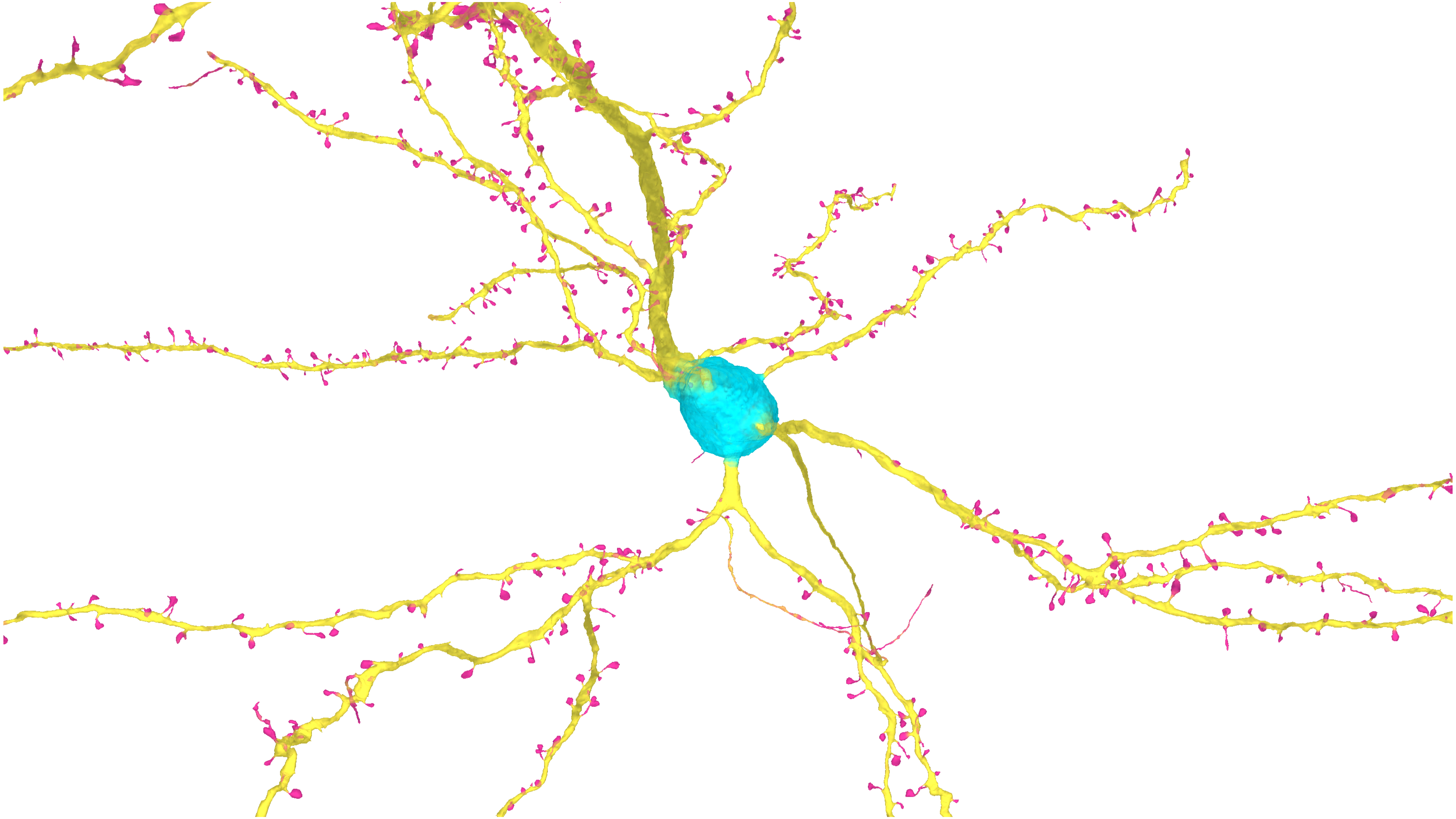


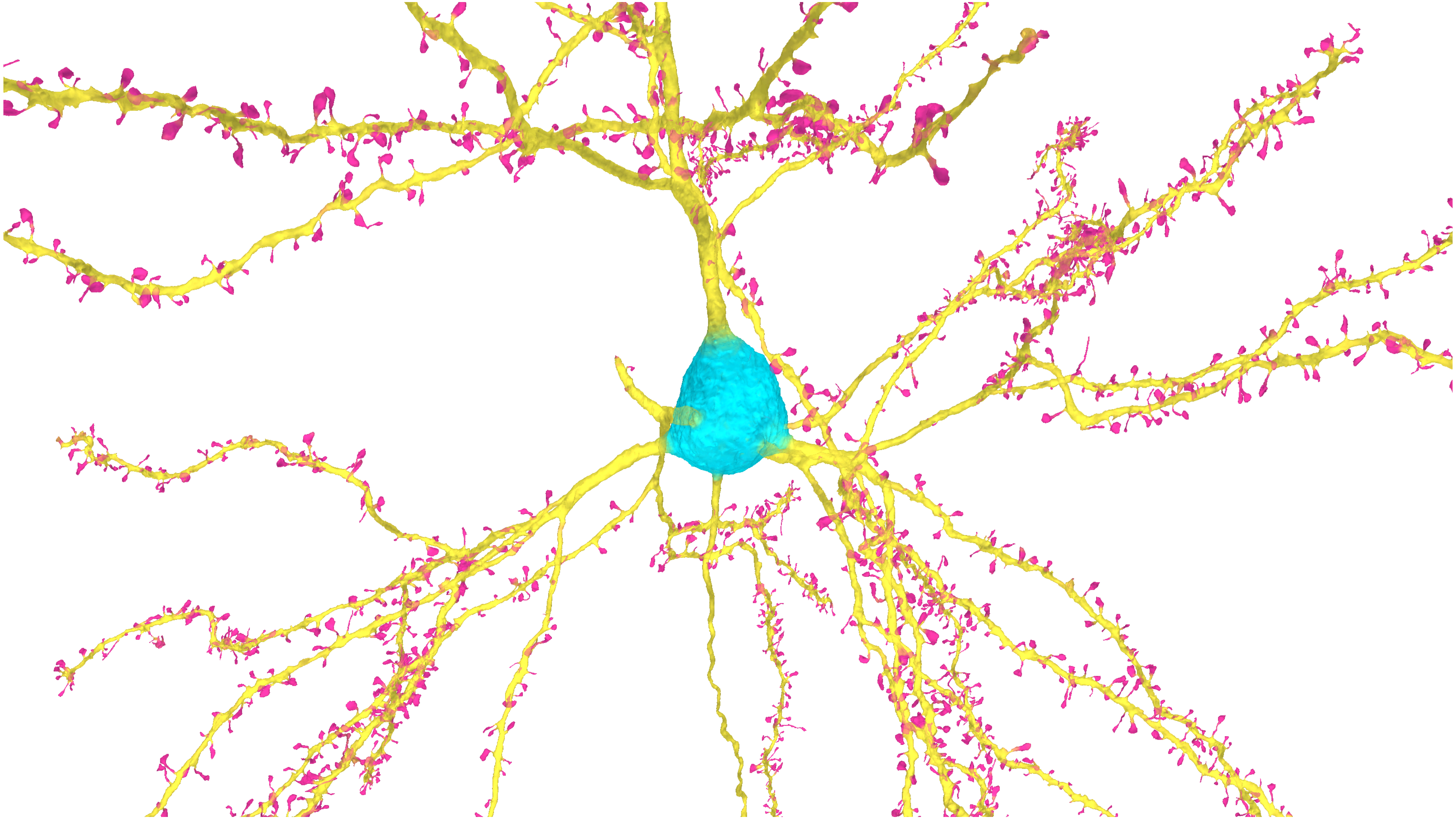


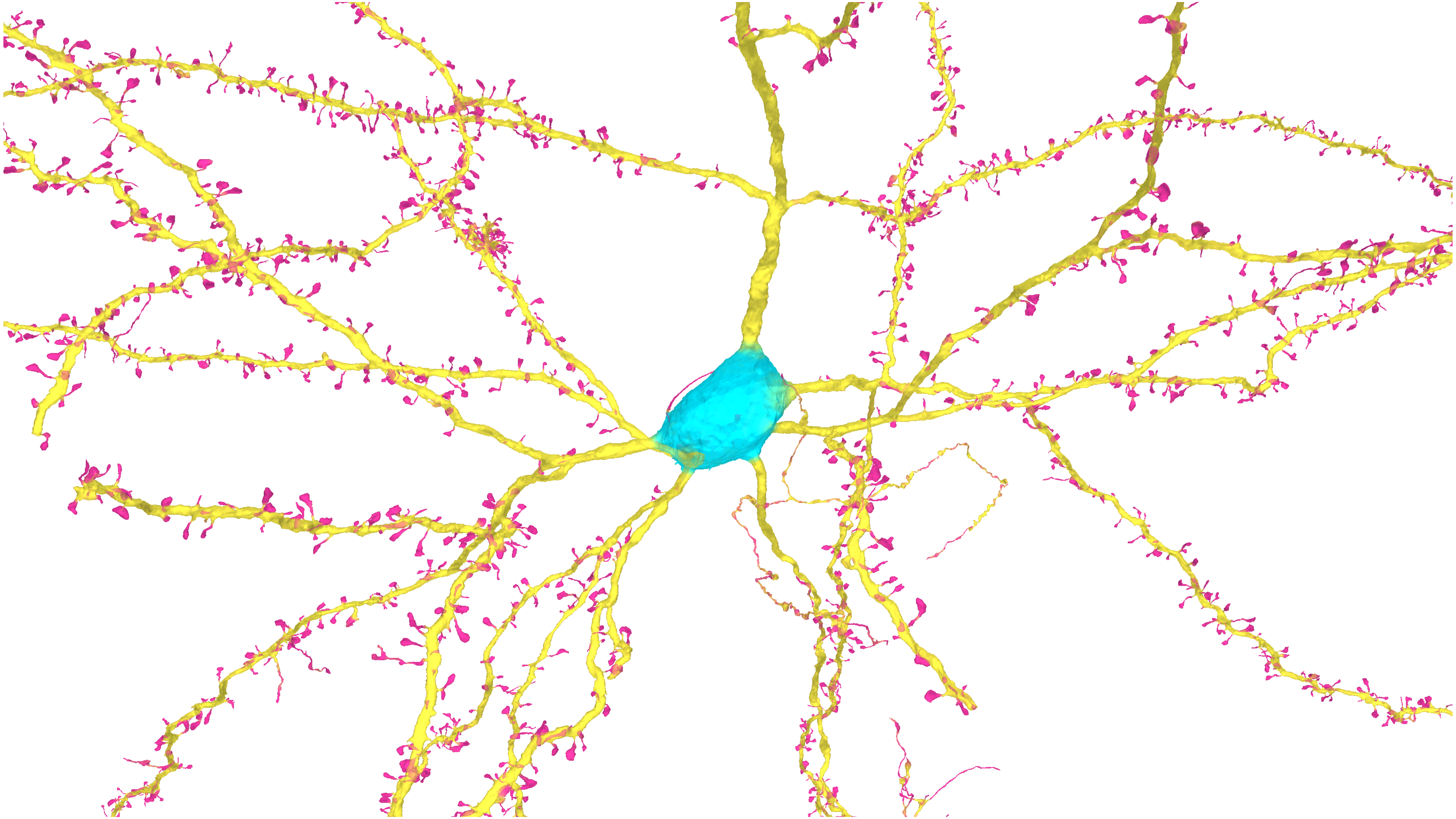


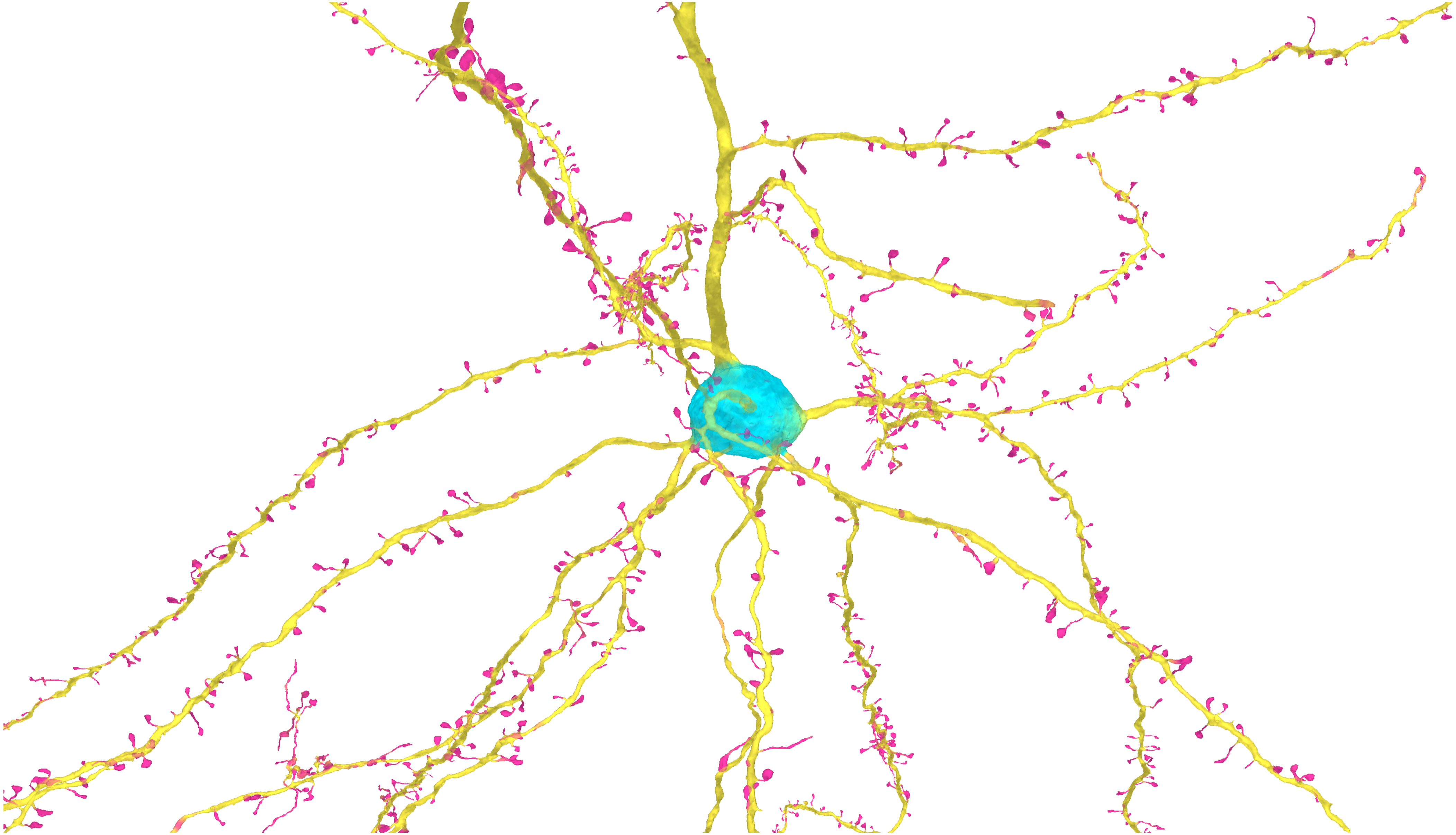


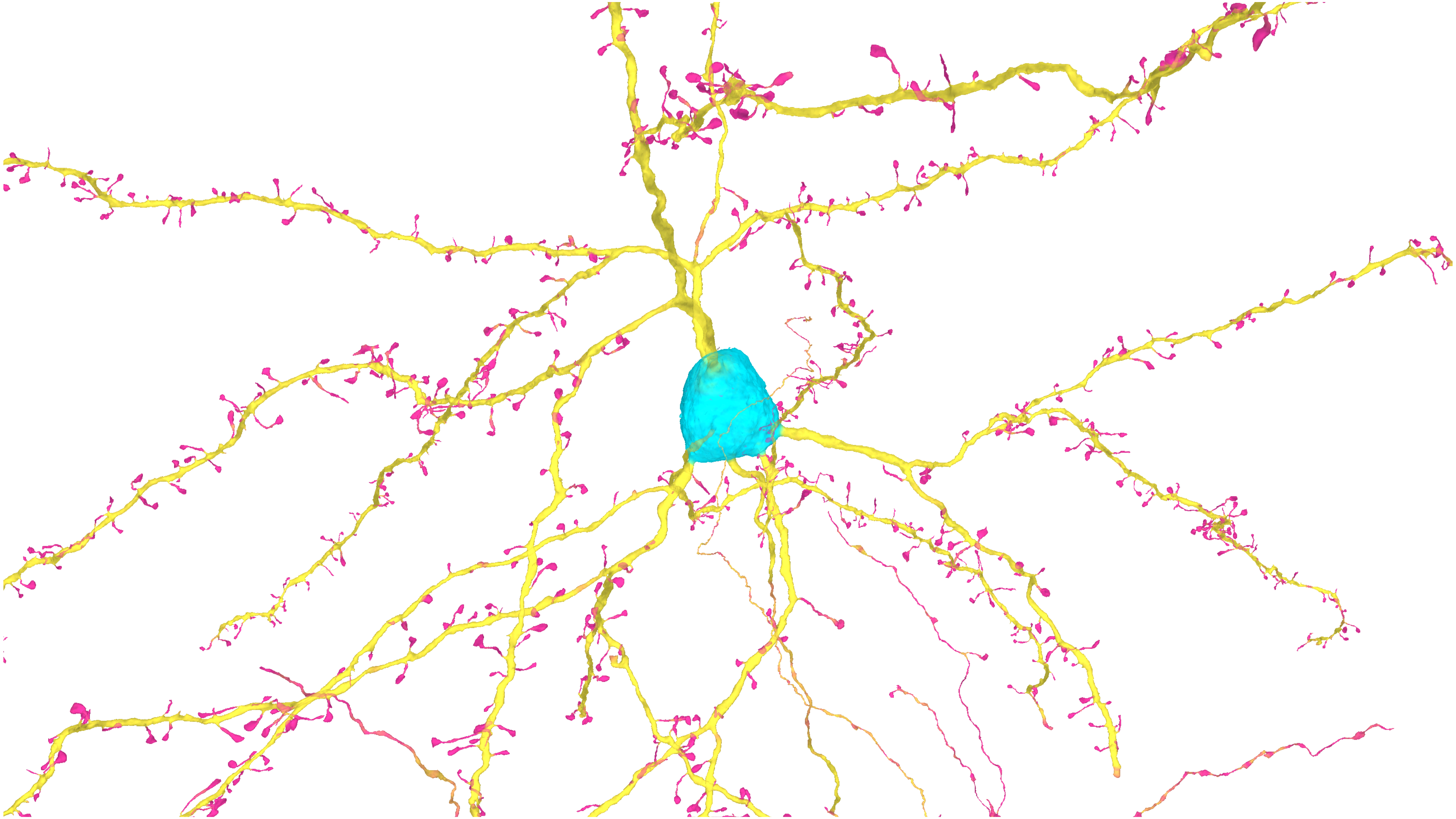


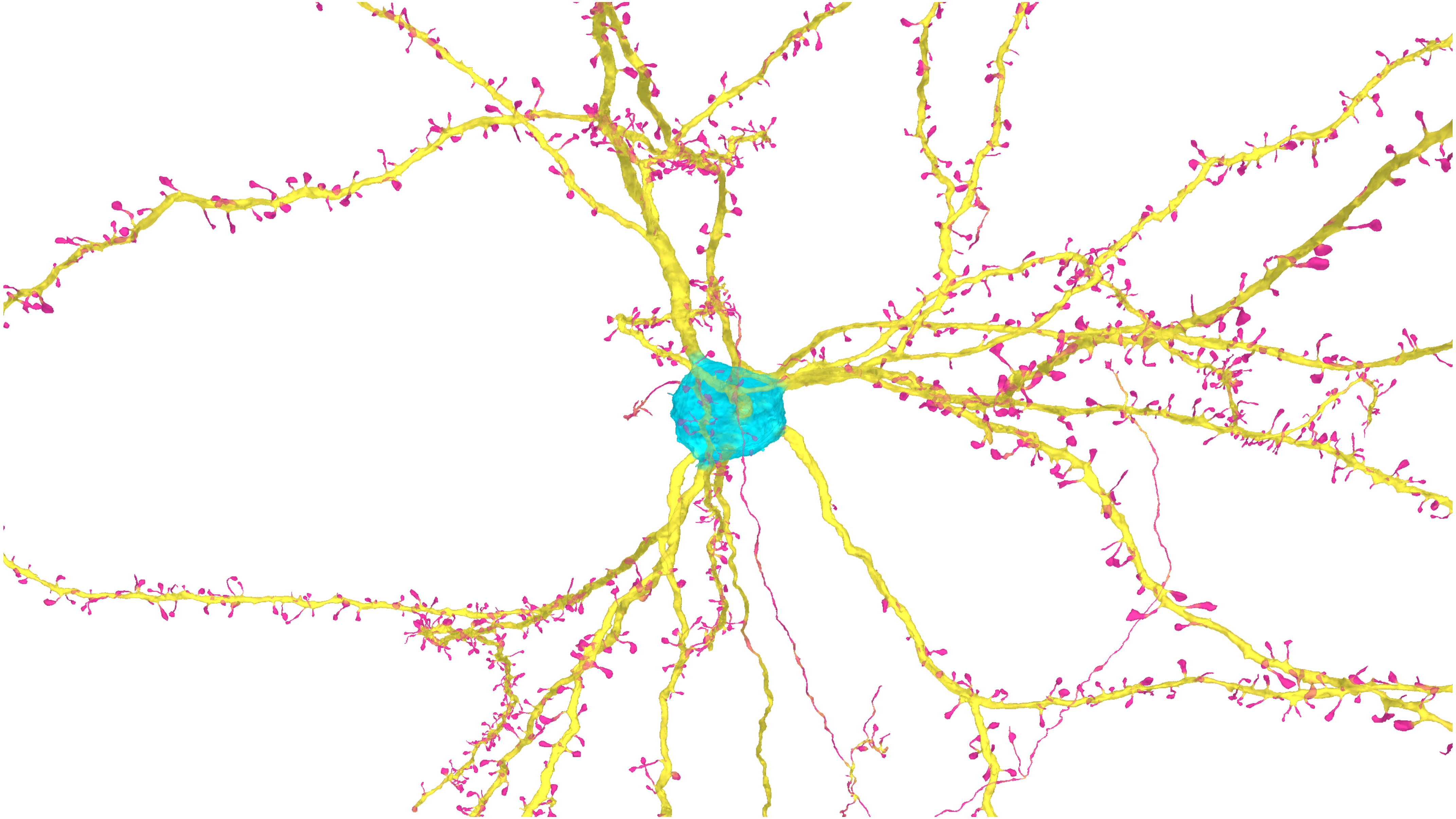


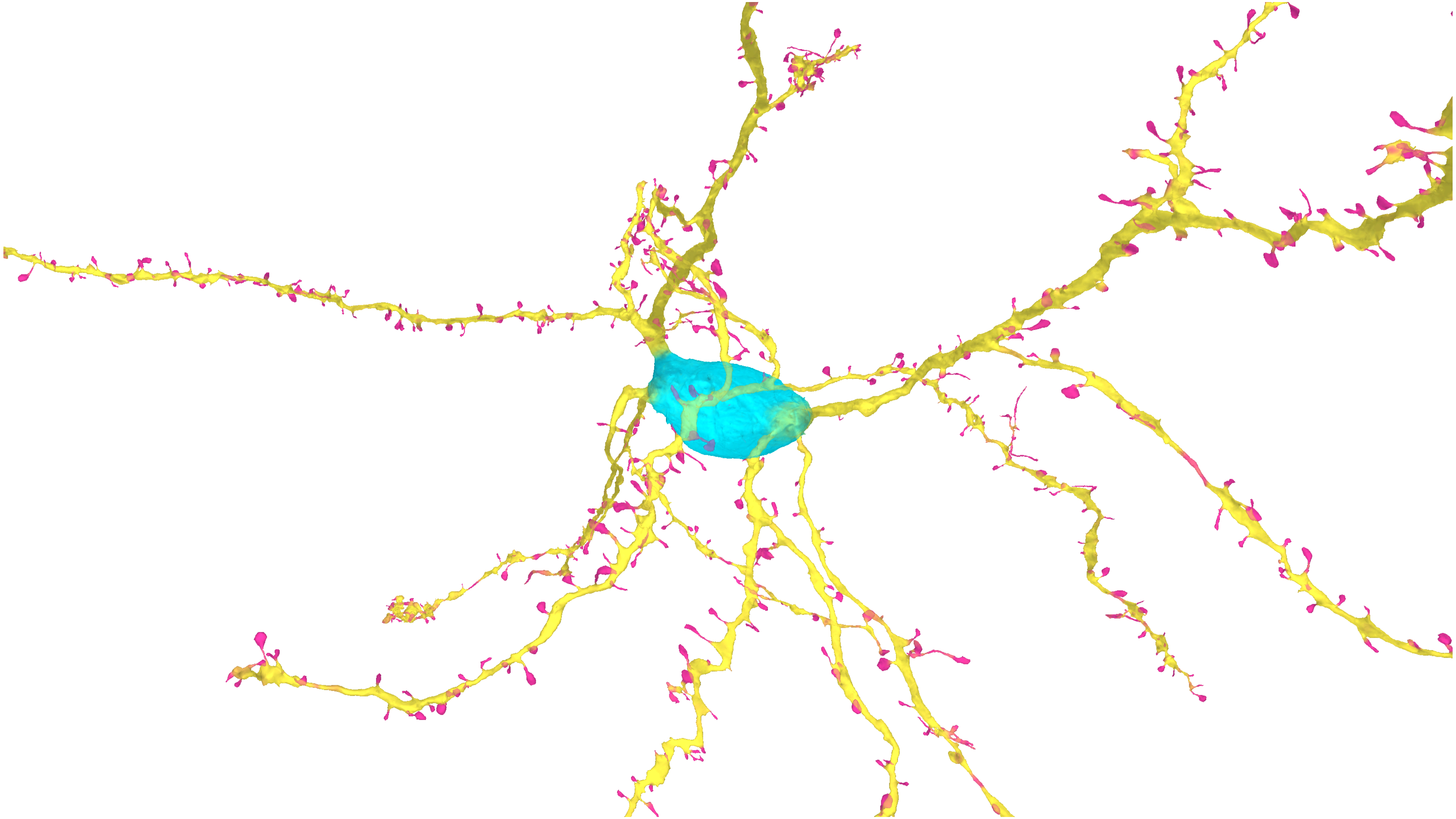


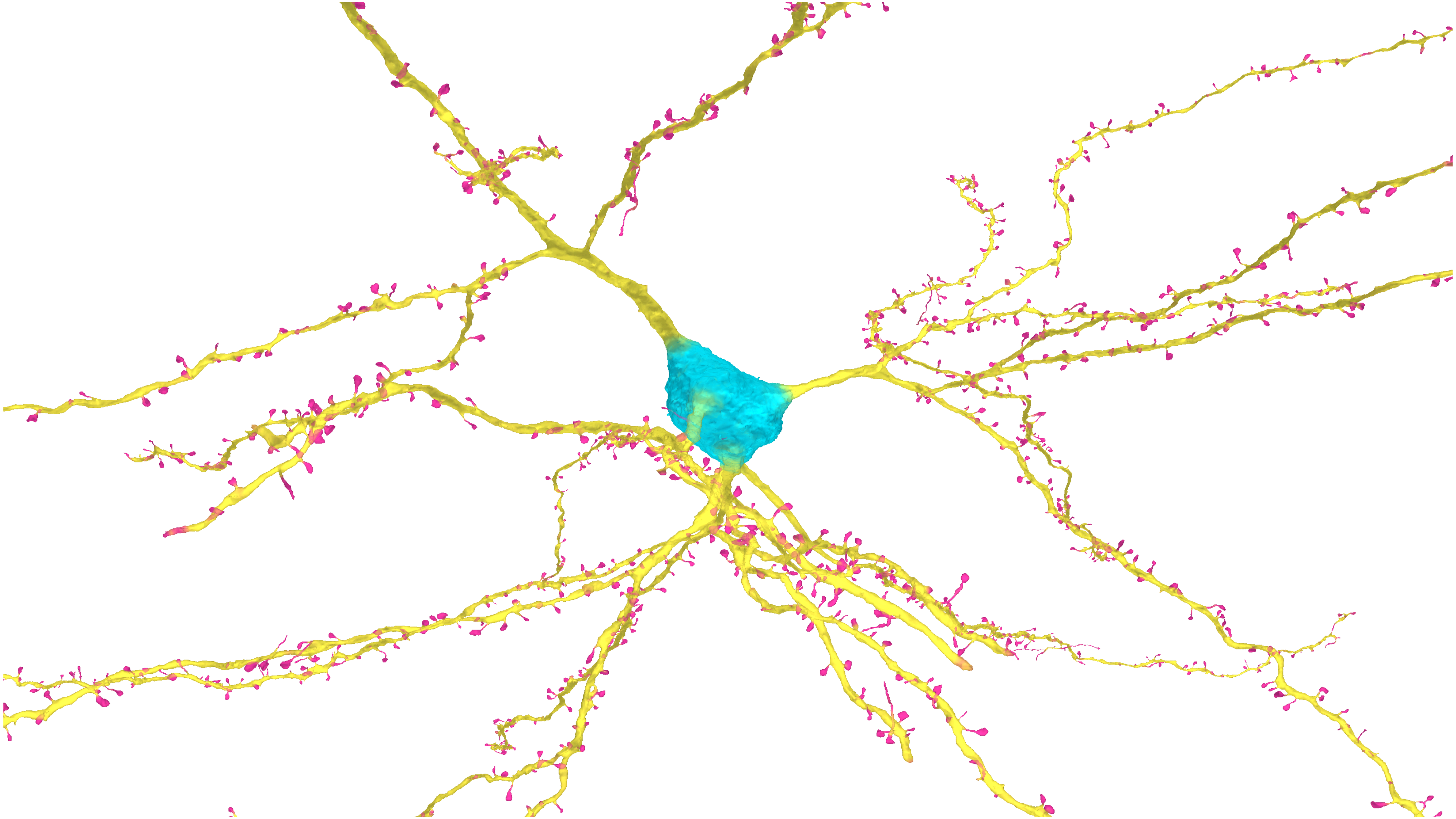


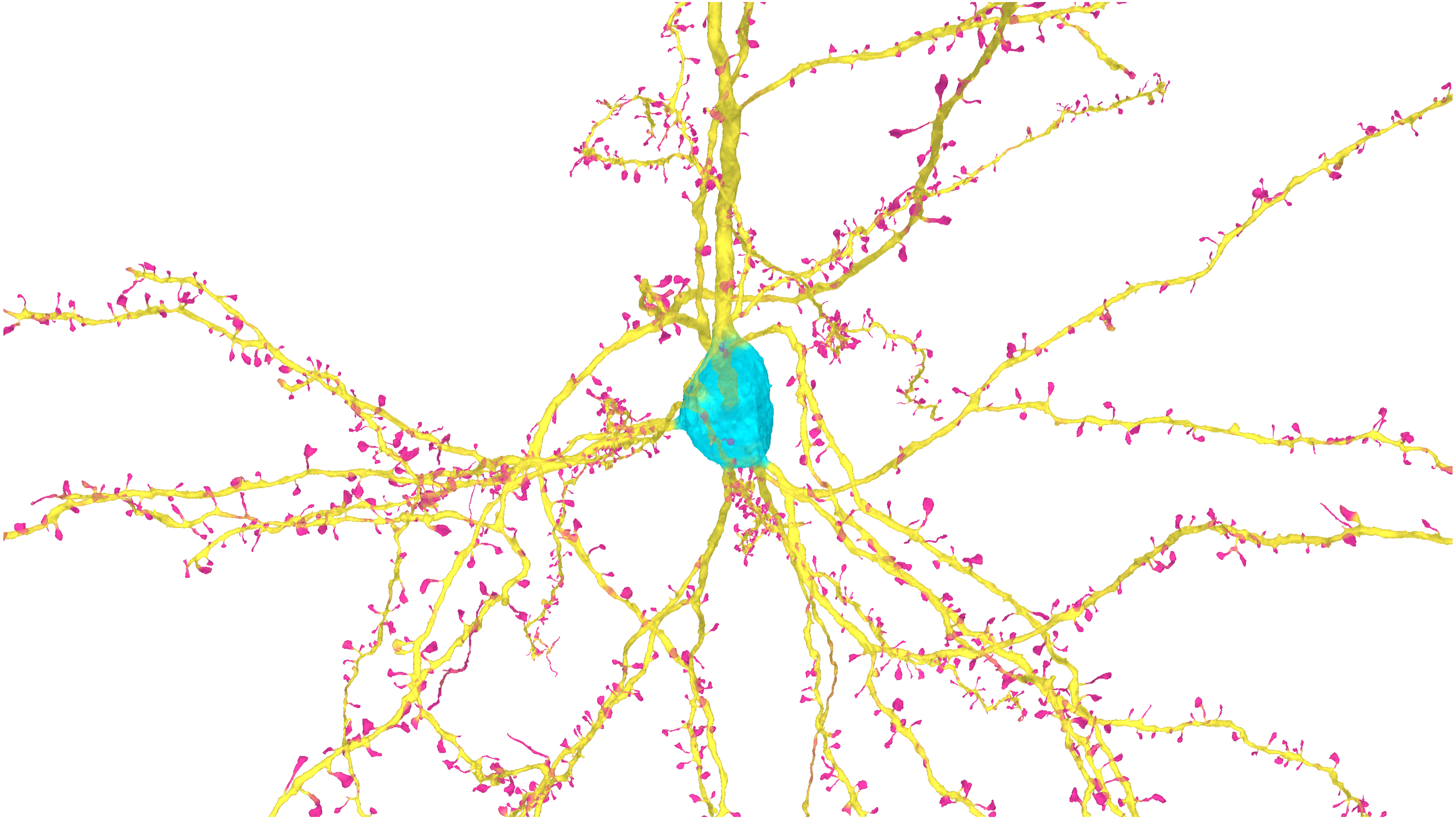




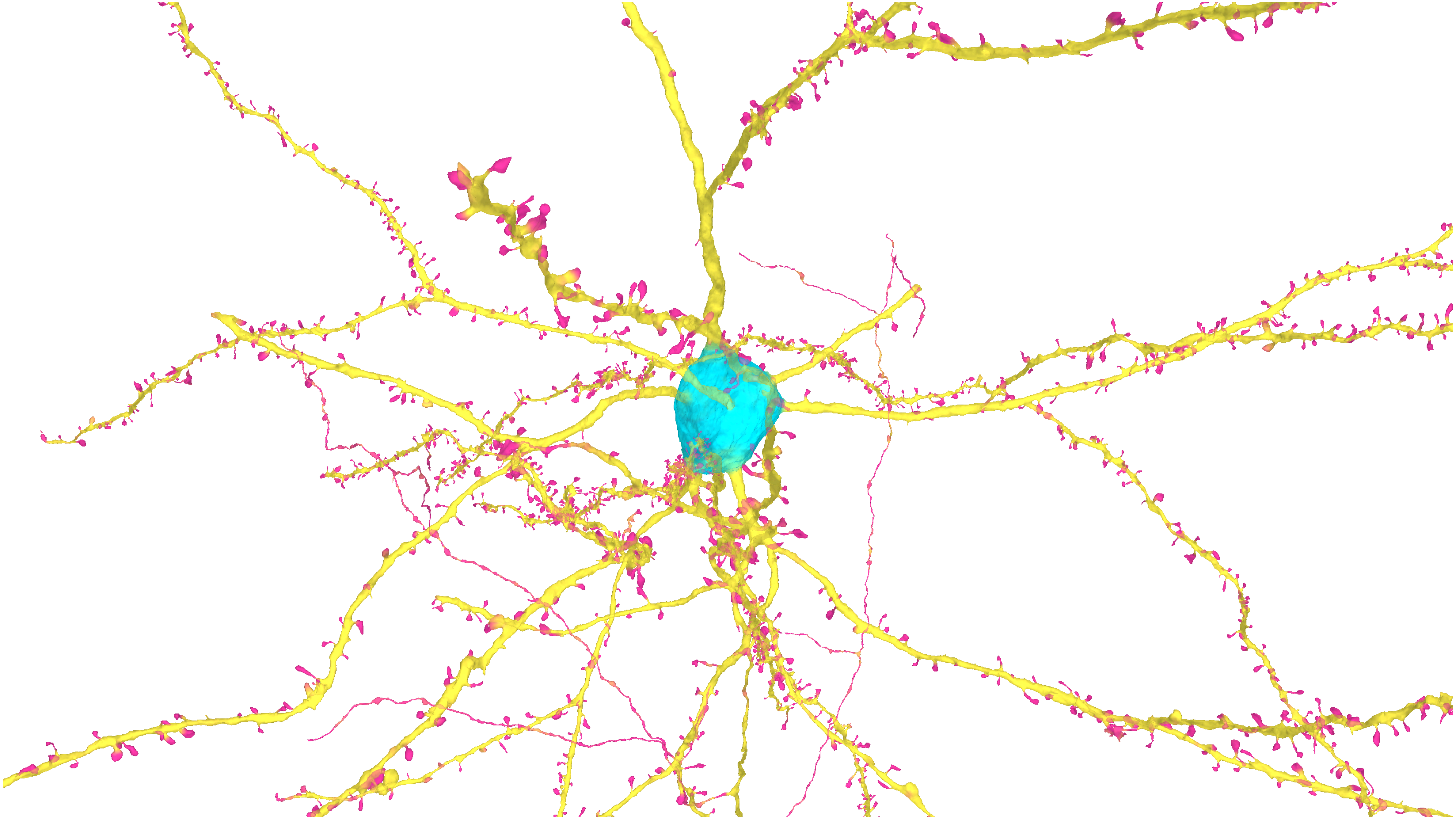


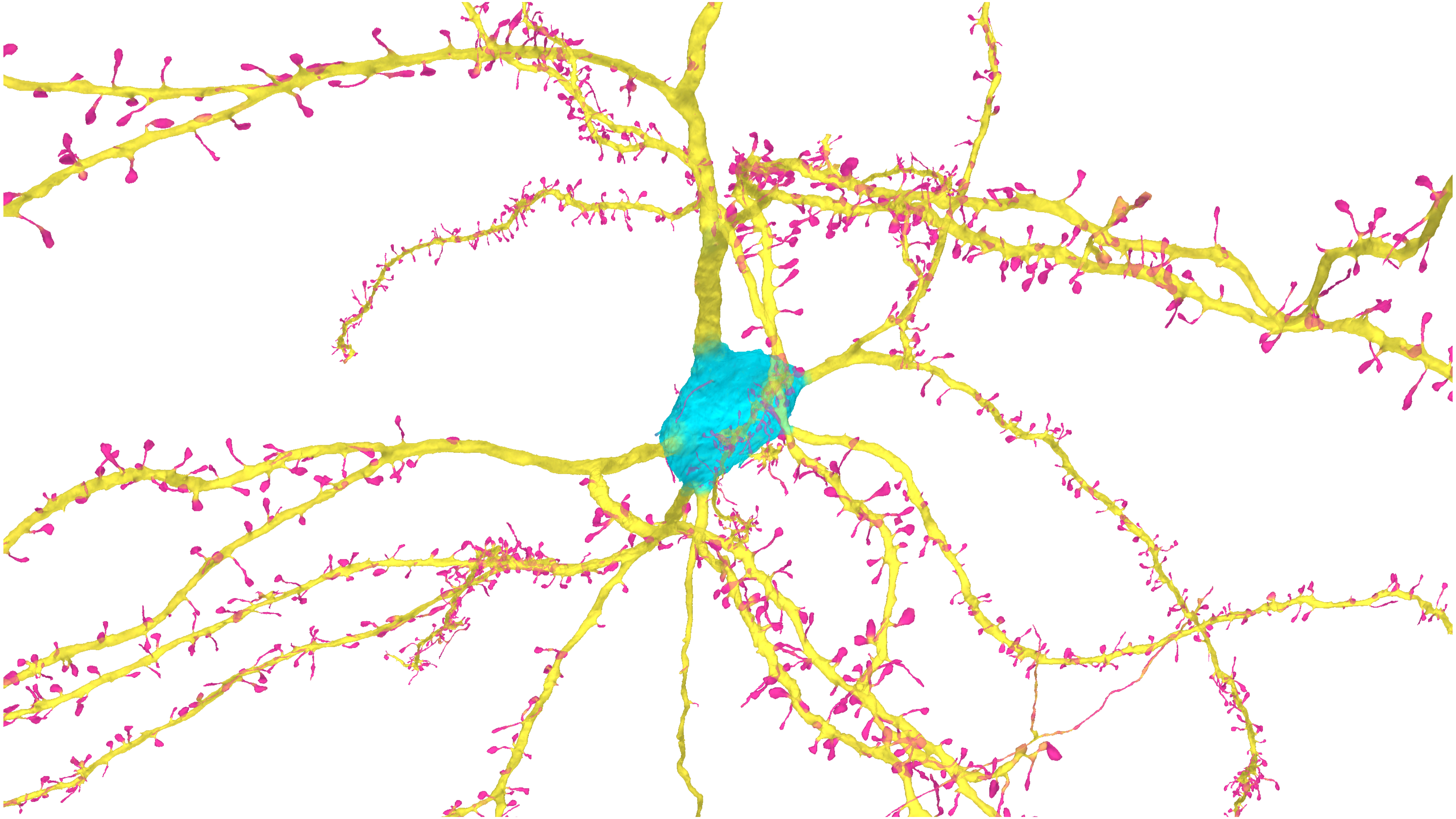


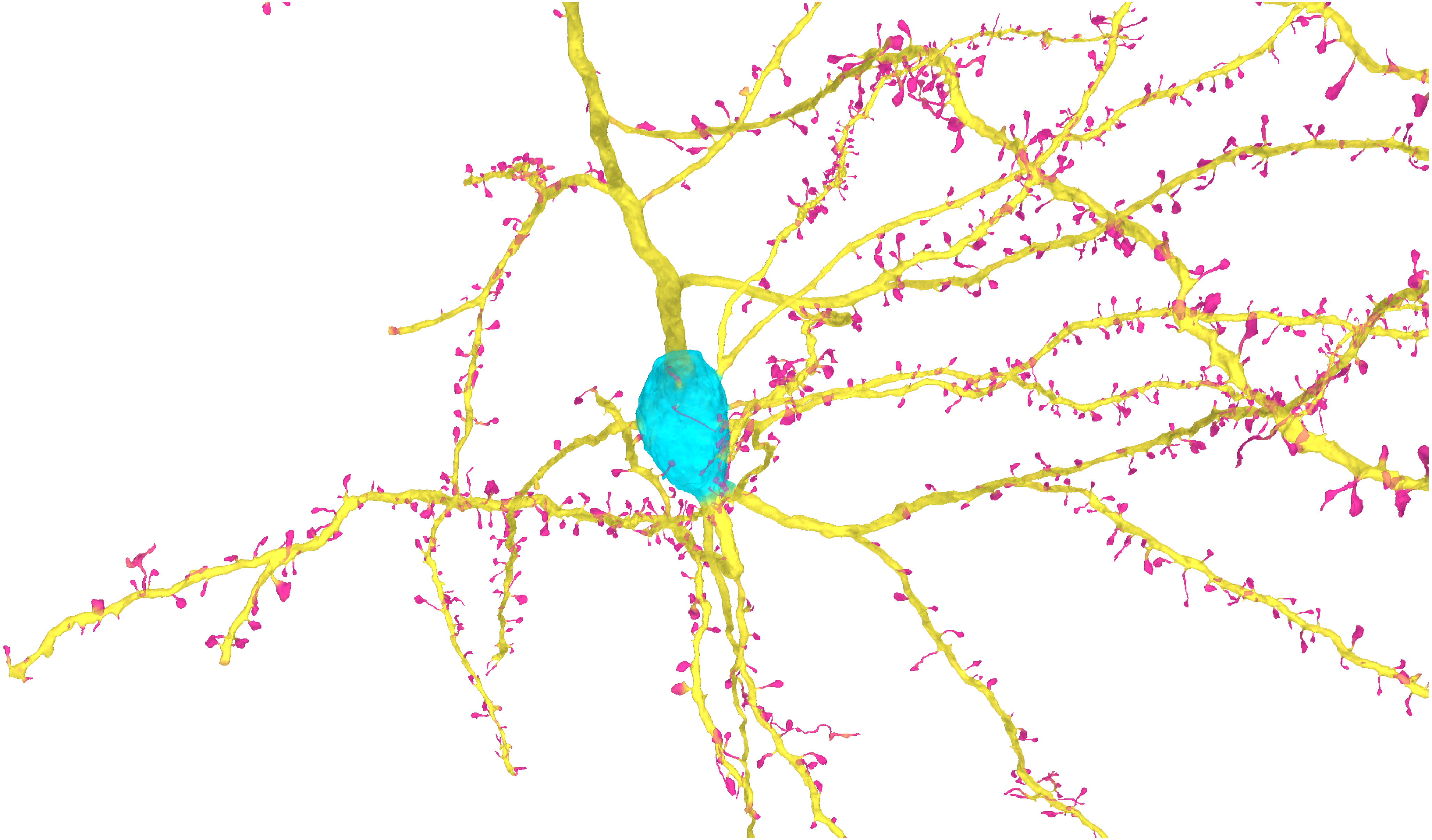


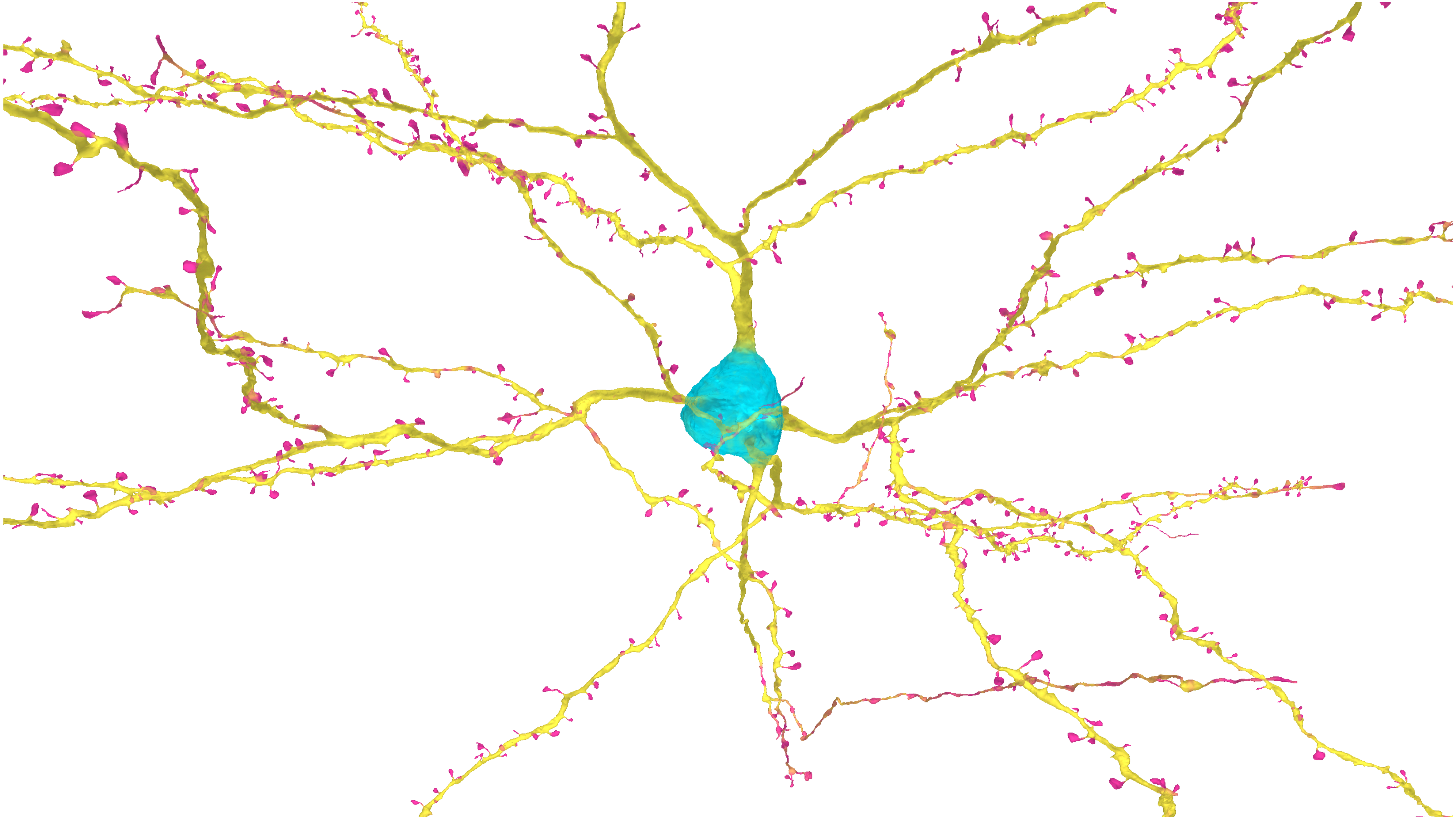


ALLEN INSTITUTE for BRAIN SCIENCE

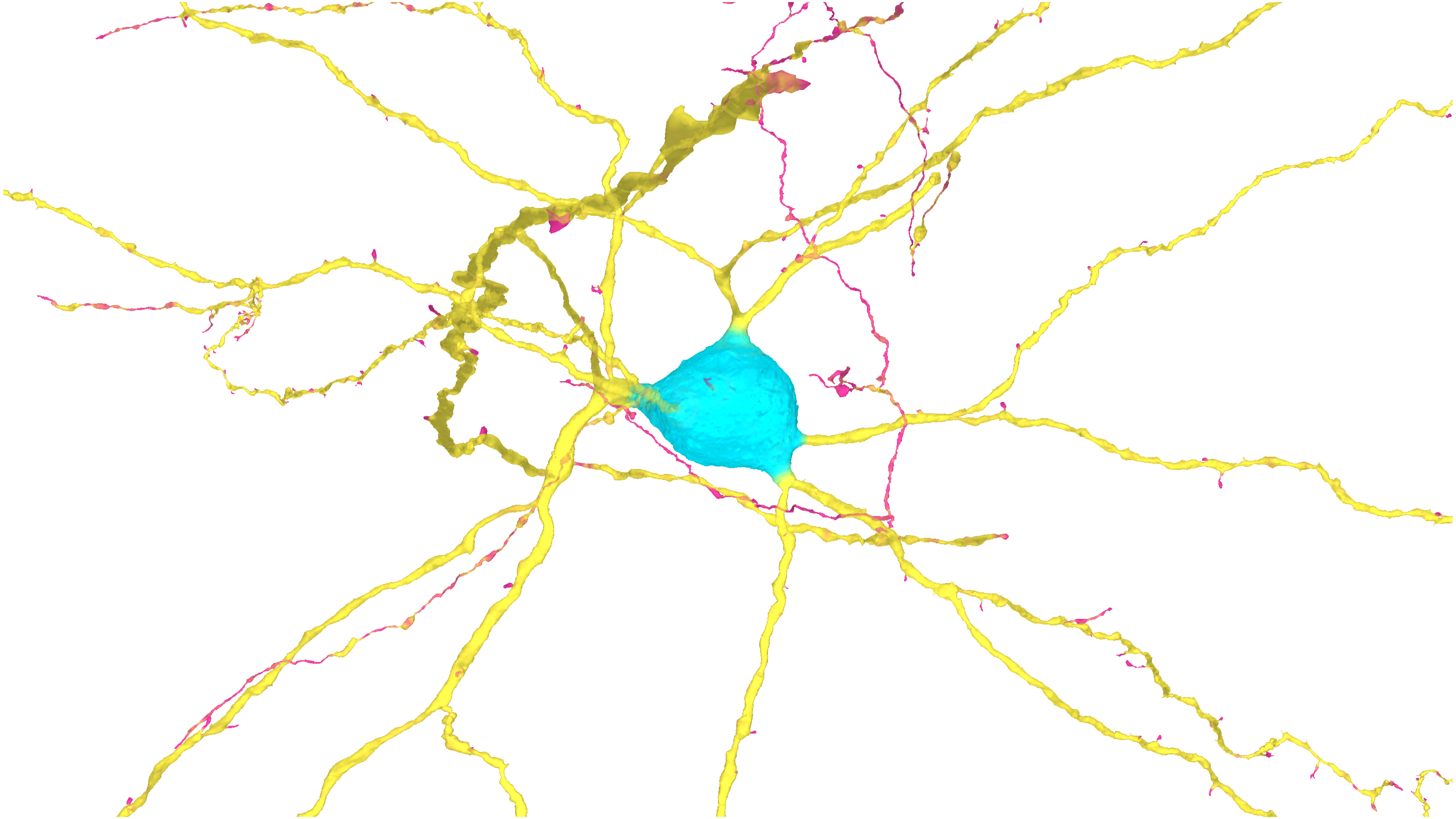


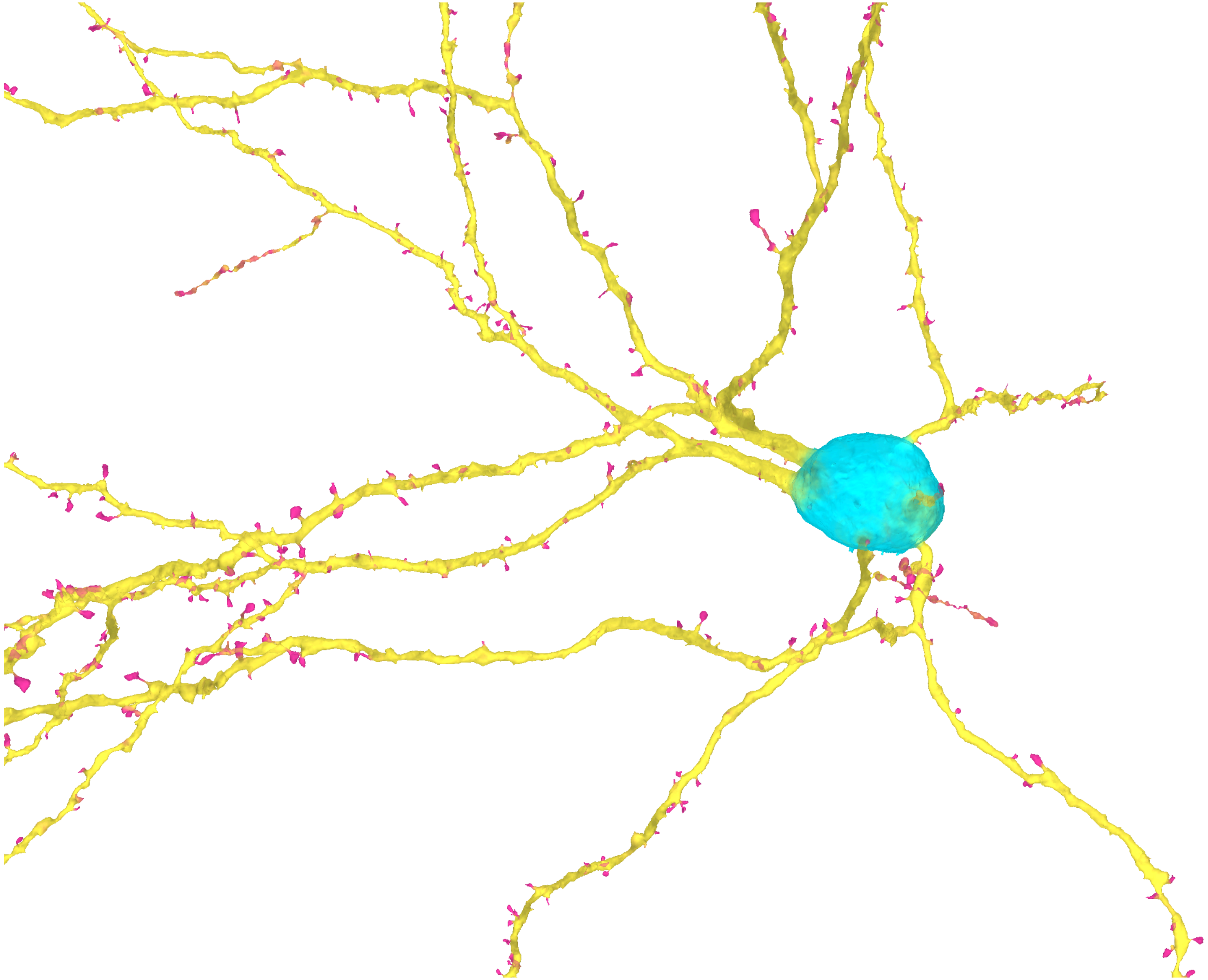


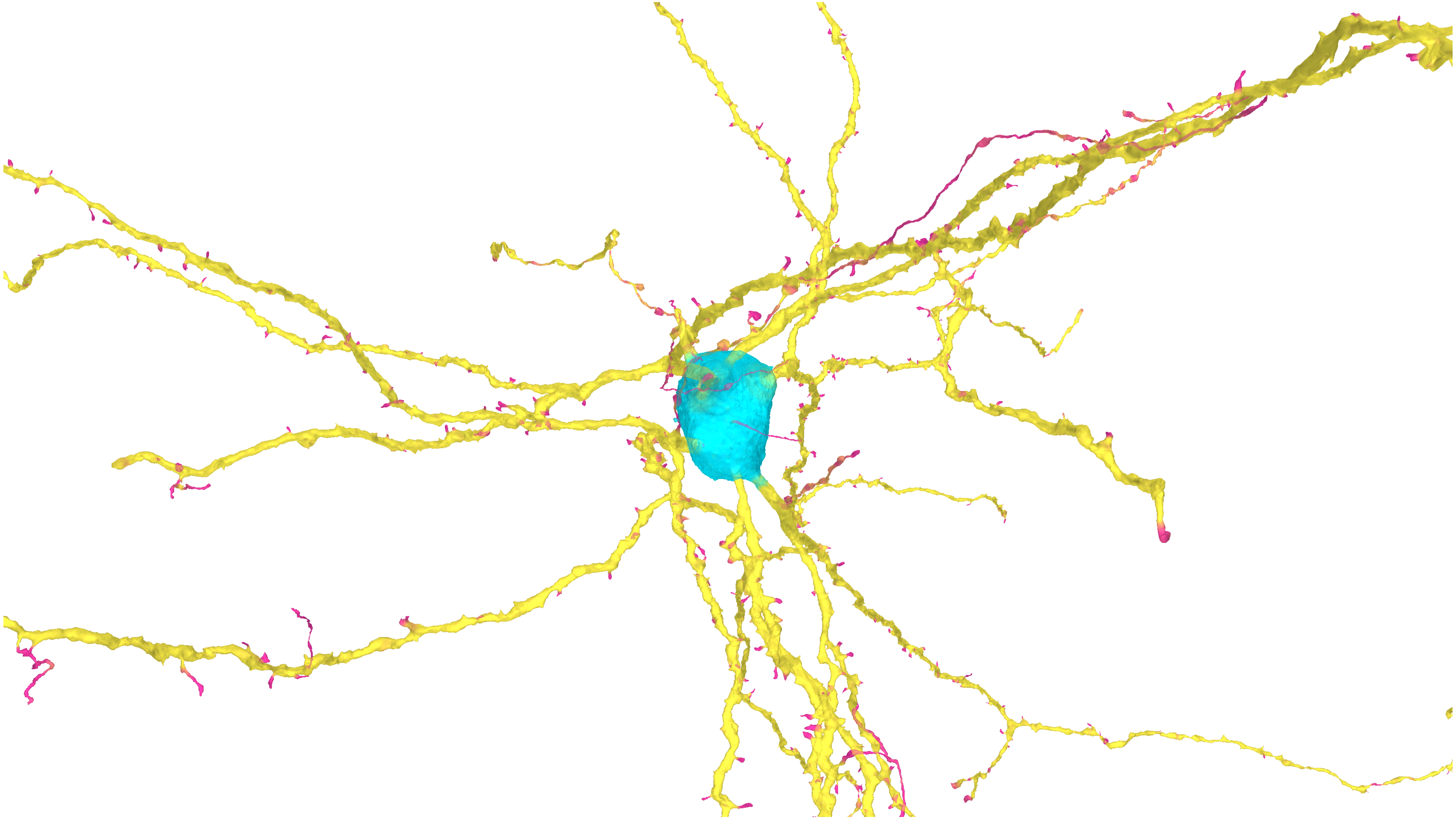


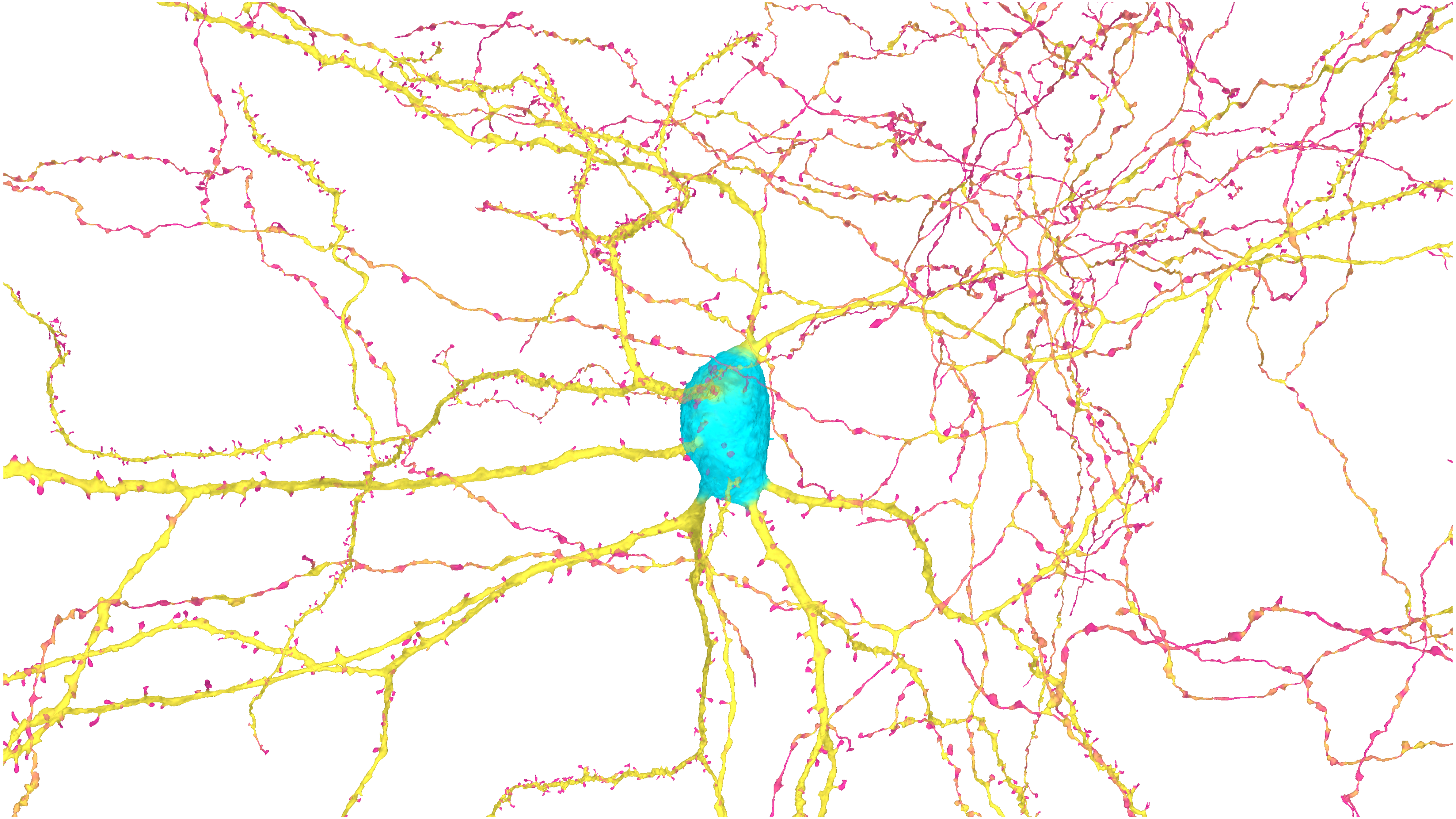


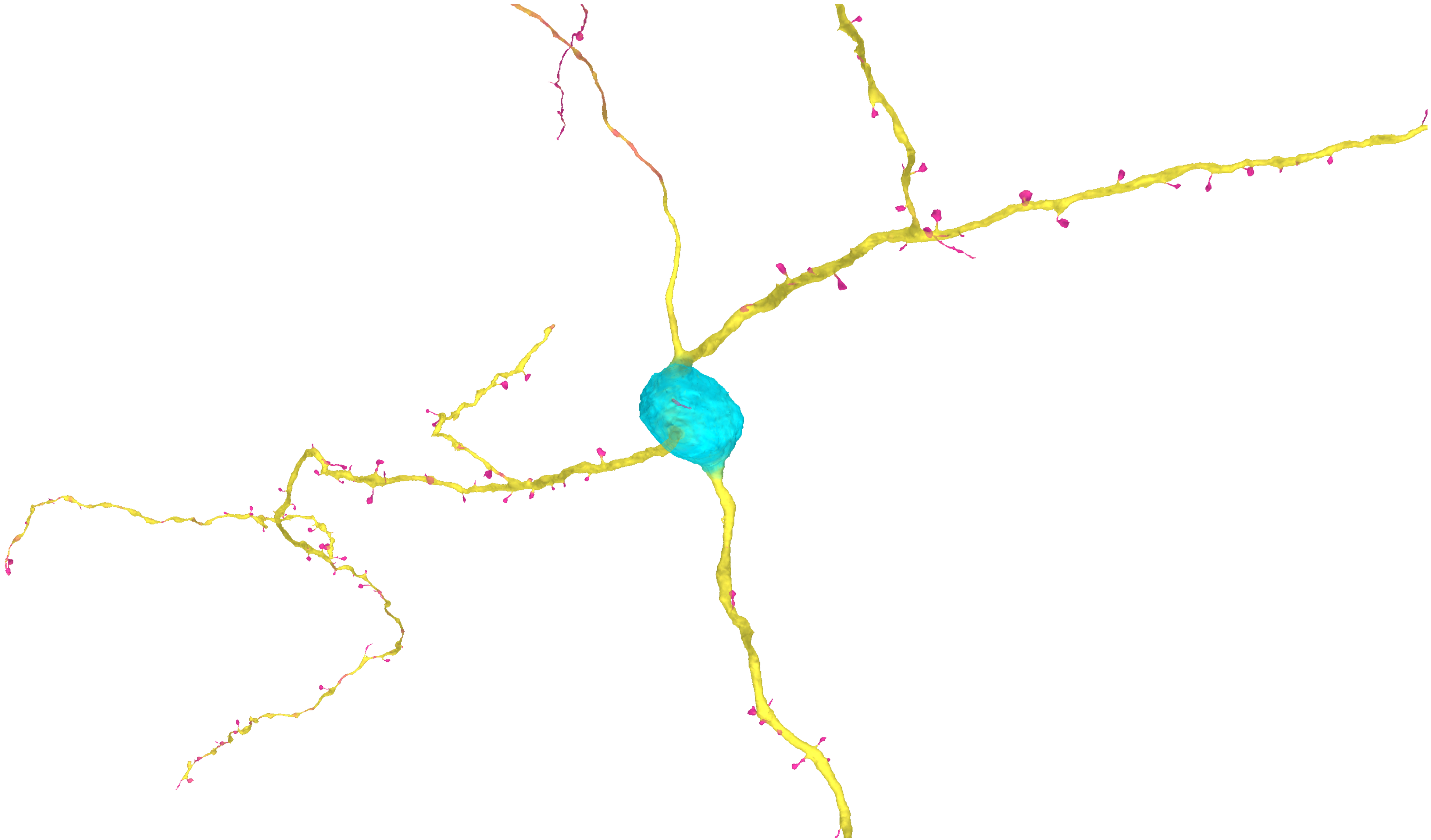
Inhibitory neurons



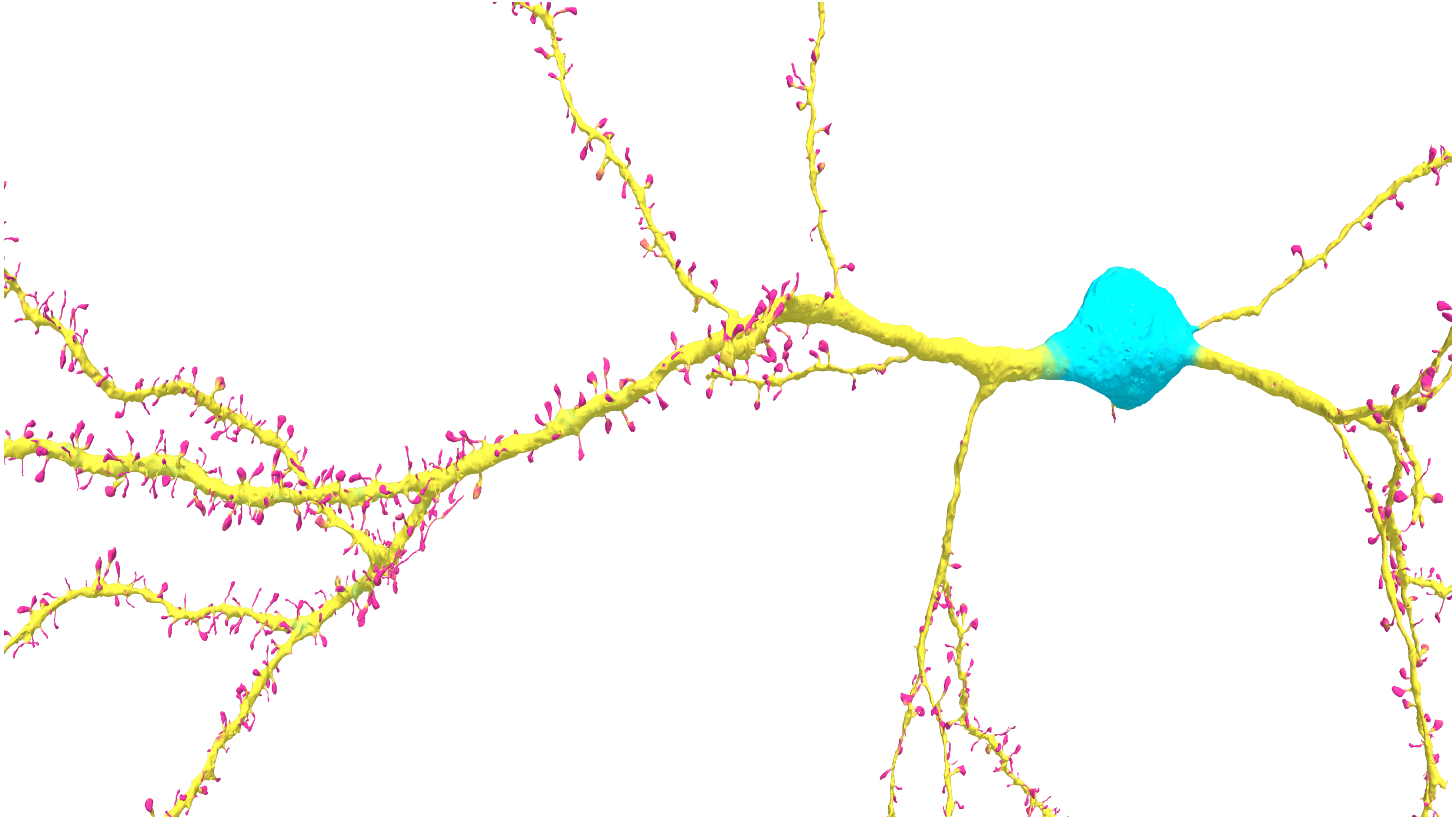








Zero-shot prediction on a H01 neuron



ALLEN INSTITUTE for BRAIN SCIENCE

Outline

- Motivation
- Intuition for heat kernel signatures
- Computing heat kernel signatures
- Application to spine prediction
- **Extensions**

HKS modifications

Volumetric HKS (w/ or w/o voxelization): Raviv et al. 2010; Rustamov et al. 2009; Rustamov 2011

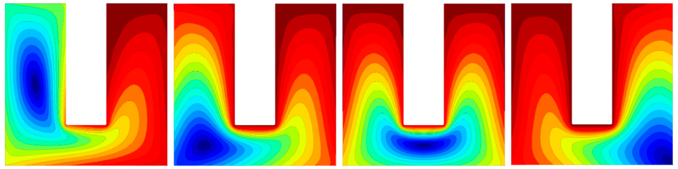


Figure 3: *Variation of the interior distance as the source point varies.*

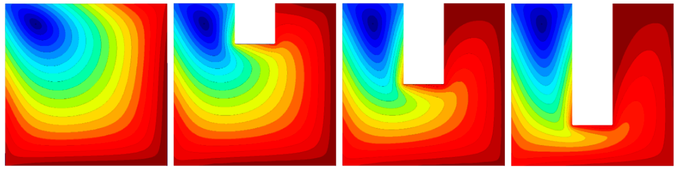


Figure 4: *The effect varying shape on the interior distance.*

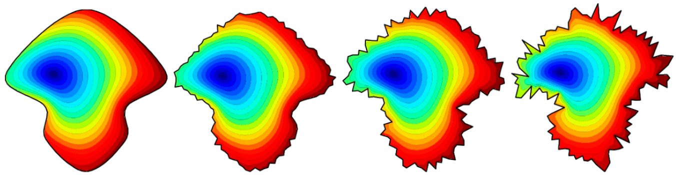
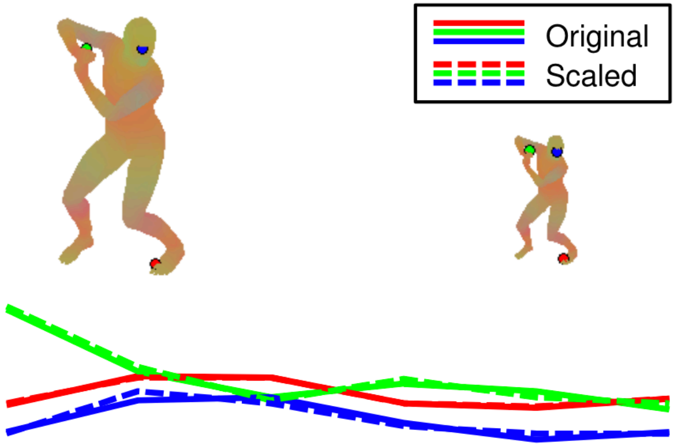


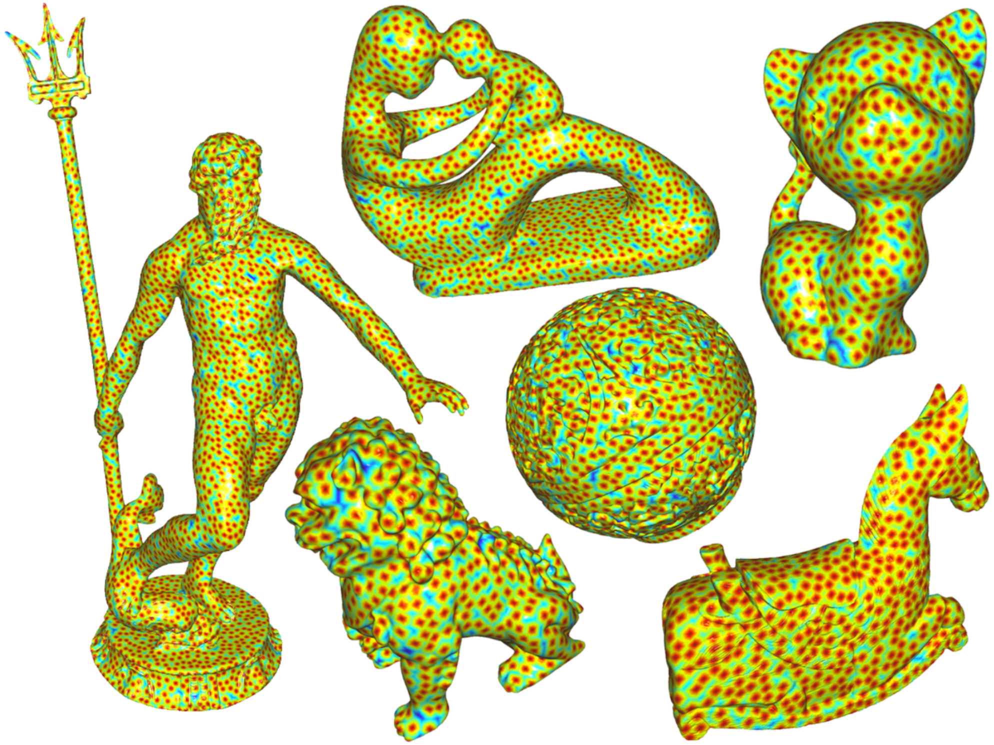
Figure 5: *The effect of adding increasing amounts of boundary noise on the interior distance.*

Scale-invariance: Bronstein et al. 2011

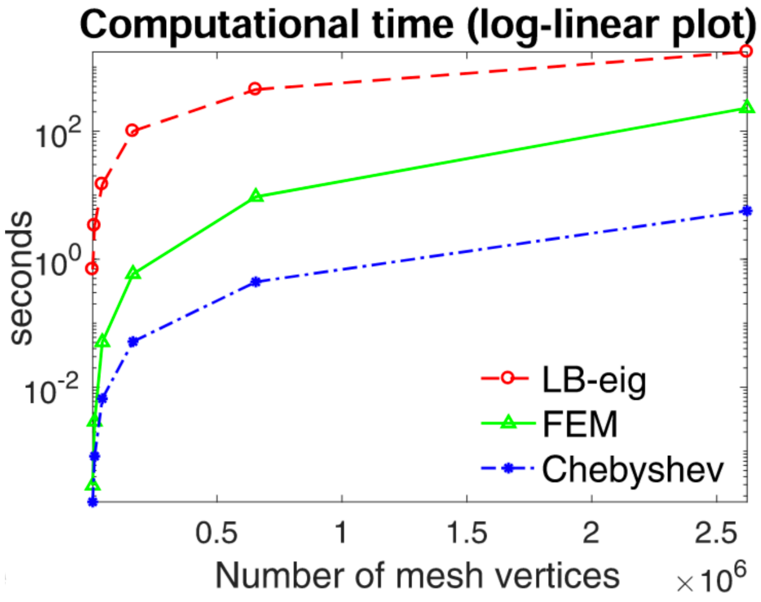


Computation

Projection-based methods: Nasikun et al. 2018; Nasikun et al. 2022; Magnet and Ovsjanikov 2023



Chebyshev polynomials: Hammond et al. 2009; Shuman et al. 2011; Huang et al. 2020

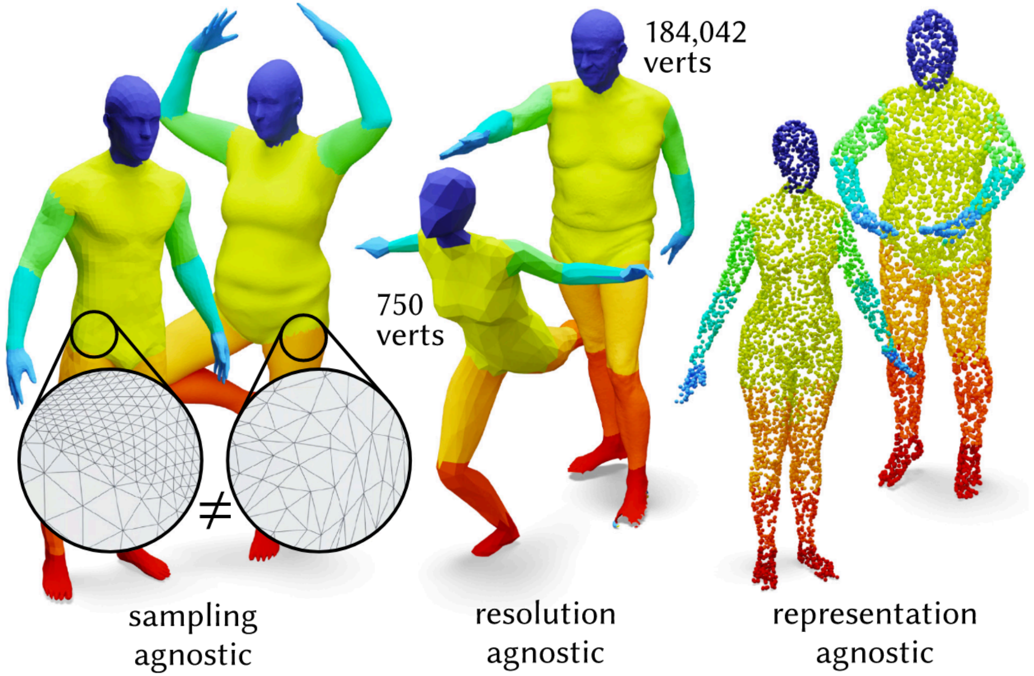


Learning

Learning more general functions of eigenvectors to discriminate classes: Litman & Bronstein 2014; Boscaini et al. 2015; Smirnov & Solomon 2021

$$\mathbf{p}(x) = \sum_{k \geq 1} \mathbf{f}(\nu_k)\phi_k^2(x),$$

Using approximate diffusion as an operator for local aggregation: Sharp et al. 2020



Summary

- Introduced the application of heat kernel signatures to neuron morphology
  - Even without learning, capture some local structures of morphology
- Showed how to scale computation of HKS to scale/resolution of neuronal meshes
- Showed these features can be used to create accurate classifiers (at least for spines) with relatively little training data
- There is a rich literature extending these ideas with different computational and learning techniques

Acknowledgements

Network Anatomy
Clay Reid
Agnes Bodor
Adam Bleckert
JoAnn Buchanan
Casey M. Schneider-Mizell
Dan Bumbarger
Derrick Brittain
Forrest Collman
Steven Cook
Nuno da Costa
Bethanny Danskin
Cameron Devine
Sven Dorkenwald
Leila Elabbady
Emily Joyce
Dan Kapner
Sam Kinn
Cheryl Lea
Melissa Lerch
Xiaoyu Lu
Gayathri Mahalingam
Erika Neace
Ben Pedigo
Sharmi Seshamani
Jenna Schardt
Rachael Swanstrom
Marc Takeno
Russel Torres
Keith Wiley
Wenjing Yin
Chi Zhang

PM
Lynne Becker
Florence D'Orazi
Sarah Naylor
Shelby Suckow
David Vumbaco
Susan Sunkin

Morphology and 3D Reconstruction
Rachel Dalley
Clare Gamlin
Staci Sorensen
Grace Williams

Modeling & Simulation
Ani Nandi
Tom Chartrand
Anatoly Buchin
Yina Wei
Soo Yeun Lee
Costas Anastassiou

Technology
Tim Fliss
Rob Young
And others

IT
Brian Youngstrom
Stuart Kendrick
Scott Harrison
Nathaniel Middleton
And others

MPE
Jay Borseth
Collin Farrell
And others

MindScope
Reza Abbasi-Asi
Anton Arkhipov
Michael Buice
Daniel Denman
Brian Hu
Josh Larkin
Stefan Mihalas
Daniel Millmann
Gabe Ocker
Naveen Ouellette
Kevin Takasaki
Saskia de Vries
Jun Zhuang

Alen Institute for Brain Sciences
Tanya Daigle
Shenqin Yao
Nikolas Jorstad
Trygve Bakken
Rebecca Hodge
Nathan Gouwens
Bosiljka Tasic
Ed Lein
Hongkui Zeng
And many others

Princeton
Sven Dorkenwald
Tommy Macrina
Sebastian Seung
Nick Turner
And team

Baylor
Jake Riemer
Andreas Tolias
And team

Harvard Medical School
Brett Graham
Wei-Chung Lee
And team

Janelia
Khaled Khairy
Stephan Saalfeld
Carolyn Ott
Jennifer Lippincott-Schwartz
And others

JHU
Jenna Glatzer
Dwight Bergles

APL
Brock Wester
And team

Neuro Surgery and Behavior
Lab Animal Services
Transgenic Colony Management
Finance
Legal

Computing Resources
BBP5 Supercomputing Resources
National Energy Research Computing Center
AI HPC
Google Cloud

Funding
IARPA - MICRONS
NSF - NeuroNex
NIH – BICCN