GAMer 2: A System for 3D Mesh Processing of Cellular Electron Micrographs

Christopher T. Lee*, Justin G. Laughlin*, Nils Angliviel de La Beaumelle, Rommie E. Amaro, J. Andrew McCammon, Ravi Ramamoorthi, Michael J. Holst, and Padmini Rangamani

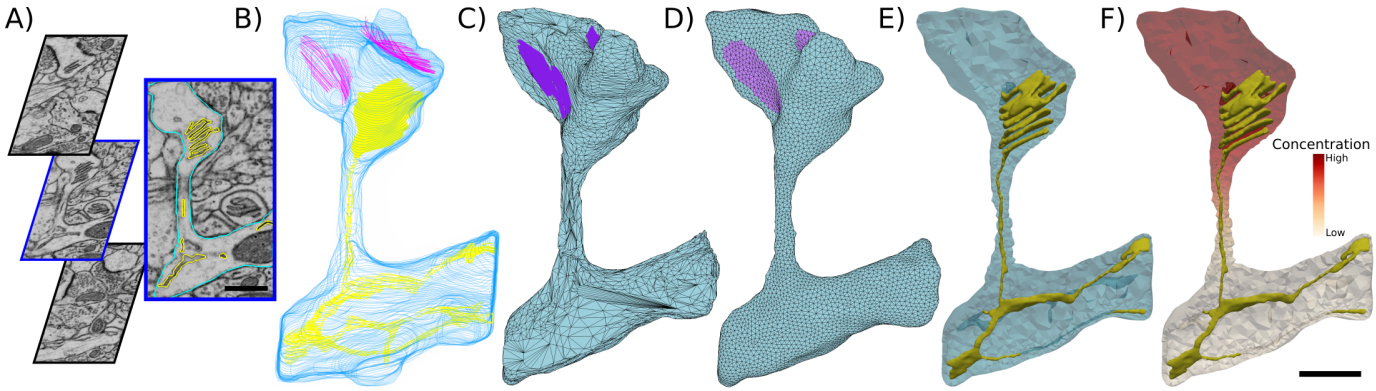

Fig. 1. Pipeline from electron microscopy data to a reaction-diffusion finite element simulation on a well-conditioned unstructured tetrahedral mesh. A) Contours of segmented data overlaid on raw slices of electron microscopy data, B) Stacked contours from all slices of segmented data, C) Primitive initial 3D mesh reconstructed by existing `IMOD` software, D) Surface mesh after processing with our system; note the significantly higher quality of the mesh. E) Unstructured tetrahedral mesh suitable for finite element simulation obtained with `TetGen` software, F) Reaction-diffusion model simulated using `FEniCS` software. Scale bar: 500 nm.

***Abstract*—*Objective:* Recent advances in electron microscopy have, for the first time, enabled imaging of single cells in 3D at a nanometer length scale resolution. An uncharted frontier for *in silico* biology is the ability to simulate cellular processes using these observed geometries. However, this will require a system for going from EM images to 3D volume meshes which can be used in finite element simulations. *Methods:* In this paper, we develop an end-to-end pipeline for this task by adapting and extending computer graphics mesh processing and smoothing algorithms. Our workflow makes use of our recently rewritten mesh processing software, `GAMer 2`, which implements several mesh conditioning algorithms and serves as a platform to connect different pipeline steps. *Results:* We apply this pipeline to a series of electron micrographs of dendrite morphology explored at three different length scales and show that the resultant meshes are suitable for finite element simulations. *Conclusion:* Our pipeline, which consists of free and open-source community driven tools, is a step towards routine physical simulations of biological processes in realistic geometries. *Significance:* We posit that a new frontier at the intersection of computational technologies and single cell biology is now open. Innovations in algorithms to reconstruct and simulate cellular length scale phenomena based on emerging structural data will enable realistic physical models and advance discovery.**

***Index Terms*—Mesh Generation, Finite Elements Simulation, Volume Electron Microscopy, Cell Modeling, Reaction-Diffusion, Dendritic Spines**

List of Acronyms

BPAP	Back Propagating Action Potential
EM	Electron Microscopy
EPSP	Excitatory Postsynaptic Potential
ER	Endoplasmic Reticulum
FEA	Finite Element Analysis
FIB-SEM	Focused-ion Beam Milling Scanning Electron Microscopy
LST	Local Structure Tensor
NMDAR	*N*-methyl-D-aspartate Receptor
PM	Plasma Membrane
PSD	Postsynaptic Density
SBF-SEM	Serial Block-Face Scanning Electron Microscopy

I. Introduction

UNDERSTANDING structure-function relationships at the cellular length scales (nm to μm) is one of the central goals of modern cell biology. While structural determination techniques are routine for very small and large scales such as molecular and tissue, high-resolution images of mesoscale subcellular scenes were historically elusive. This was primarily

* Contributed equally to this work.

Manuscript received Jan 2, 2019; CTL, REA, JAM, and MJH are supported in part by the National Institutes of Health under grant number P41-GM103426. CTL, and JAM are also supported by the NIH under RO1-GM31749. CTL also acknowledges support from the NIH Molecular Biophysics Training Grant T32-GM008326. MJH was supported in part by the National Science Foundation under awards DMS-CM1620366 and DMS-FRG1262982. PR was supported by AFOSR FA9550-18-1-0051 and JGL was supported by a fellowship from the UCSD Center for Transscale Structural Biology and Biophysics/Virtual Molecular Cell Consortium.

CTL, REA, and JAM: Department of Chemistry and Biochemistry; JGL, NALB, and PR: Department of Mechanical and Aerospace Engineering; RR: Department of Computer Science; MJH: Department of Mathematics; University of California, San Diego, La Jolla, CA, 92093 USA (correspondence e-mail: prangamani@ucsd.edu).

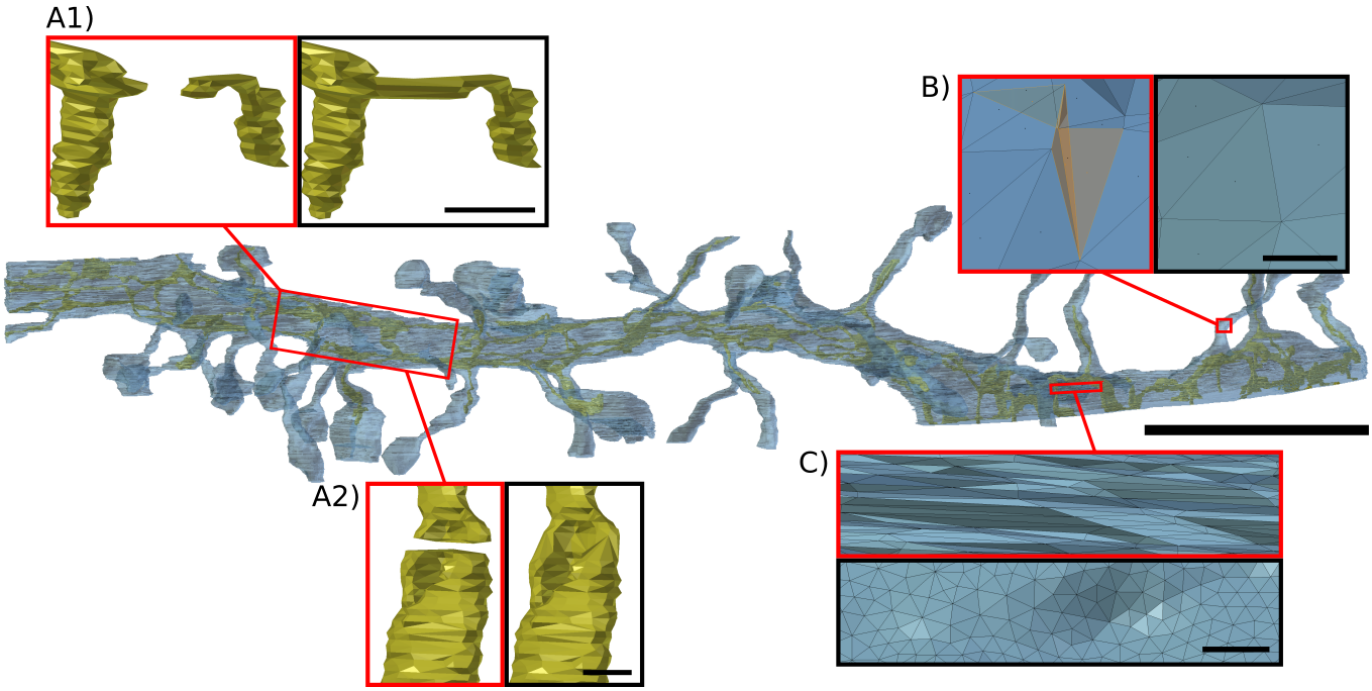

Fig. 2. Initial surface mesh model of a dendritic segment with subcellular organelles imaged by FIB-SEM, courtesy of Wu et al. [1], contains many mesh artifacts and is not compatible with physical modeling. The blue surface represents the plasma membrane (PM), while yellow represents the membrane of the endoplasmic reticulum (ER). (A1/A2) Left, due to the fine ultrastructure of the process, portions of the ER are not resolved by the microscope and become erroneously disconnected as separate meshes. Right, biological realism is manually restored by bridging the two ER segments. Scale bars: A1) 100 nm, A2) 20 nm. (B) Left, intersecting faces prevent generation of a tetrahedral mesh. Right, `GAMer` smoothing algorithms automatically untangle knots of intersecting faces. Scale bar: 20 nm. (C) Top, high aspect ratio triangles. Bottom, well conditioned surface mesh. Scale bar: 100 nm. Scale bar for full geometry: 4 μm.

due to the diffraction limits of visible light and the limitations of X-ray and Electron Microscopy (EM) hardware. Over the past decade, technological improvements such as improved electron direct detectors have enabled the practical applications of techniques such as volume electron microscopy. Advances in microscopy techniques in recent years have opened windows into cells, giving us insight into cellular organization with unprecedented detail. Volumetric EM enables the capture of 3D ultrastructural datasets (i.e., images where fine structures such as membranes of cells and their internal organelles are resolved, as shown in (Fig. 1A)). Using these geometries as the basis of simulations provides an opportunity for *in silico* animation of various cellular processes and the generation of experimentally testable hypotheses. However, to the best of our knowledge, there is no current free and open-source system for going from EM images to 3D meshes, which can be used in finite element simulations.

Here, we introduce our recently redesigned software, `GAMer` (Geometry-preserving Adaptive MeshER), as a tool connecting and implementing methods for 3D meshing and reconstruction of cellular electron micrographs (Fig. 1). The input to this pipeline are stacks of high-resolution images (Fig. 1A) from which contours of different organelles and cellular components that are separated by membranes are segmented and identified (Fig. 1B). These steps are often done in experimental and imaging labs. Superposition of these 2D contours based on the known z-separation gives us the first 3D contour data set (Fig. 1B). Traced contours are then constructed into a primitive 3D surface mesh using `IMOD` (Fig. 1C) [2]. The surface meshes of the external and internal membranes are then conditioned to improve element quality via `GAMer`; faces are also marked using a `GAMer` `Blender` interface to allow for different boundary conditions during simulation (Fig. 1D) [3]. `TetGen` takes these high-quality surface meshes as inputs and generates an unstructured tetrahedral mesh (Fig. 1E) [4]. This mesh is now ready for finite element based physical simulation of different cellular processes (Fig. 1F).

II. Image acquisition and segmentation

We briefly summarize sample preparation in this section for completeness. Sample preparation begins with either cell culture or harvesting of biological tissues of interest. Subsequent preparation steps can vary depending upon the particular volume EM imaging modality used but primarily include sample dehydration, fixation/staining, embedding, and imaging through the different cross-sections [5]–[8].

Once the images are captured, in order to construct an initial mesh model from the data, the boundaries of features must first be identified. Much of this relies on the expertise of biologists for recognition of organelles and membrane domains in cells. During the segmentation process, the algorithm or researcher must carefully separate boundary signal from noise. Various schemes ranging from manual tracing, thresholding and edge-detection, to deep-learning based approaches have been employed to perform image segmentation [8].

A. Meshing challenges

A variety of challenges for meshing and subsequent physical simulations can arise at each step. Even with perfect experimental execution, and despite the enhanced surface contrast from heavy metal stains, the membranes of cells and their internal organelles are often poorly behaved and contain sharp and otherwise irregular geometries that are difficult to segment. In more serious cases, thinly sliced samples can tear or become contaminated during handling. Methodological errors are also possible. For example, Serial Block-Face Scanning Electron Microscopy (SBF-SEM) datasets in optimum conditions may have 3 nm lateral (x,y) resolution but 25 nm axial (z) resolution, limited by the slicing capability of the ultramicrotome [5]. Anisotropic resolution in tandem with variable slice thickness can cause loss of axial detail.

There are many EM software which post-process image stacks to correct for these and other artifacts not mentioned. Most of our datasets have been manually segmented and corrected in software such as `IMOD` [2]. `IMOD` has the capacity to perform a contour-tiling operation to generate a preliminary surface mesh suitable for basic 3D visualization. These meshes, however, are not directly suitable for physical simulations due to various mesh artifacts. Some of these include intersecting faces, non-manifold features, and high aspect ratio faces, as shown in Fig. 2. To resolve these problems, we perform surface mesh conditioning steps described below.

III. Methods

A. Mesh Processing

Our main contribution in this work is a surface mesh processing library `GAMer` 2, which features algorithms as described by Yu *et al.* [9], [10]. We have recently rewritten `GAMer` in C++ using the `CASC` data structure [11] as the underlying mesh representation. The code is licensed under LGPL v2.1 and can be obtained from GitHub (https://github.com/ctlee/gamer) [12]. `GAMer` can be used as a stand-alone library, or alternatively the algorithms can be accessed through a `GAMer Blender` add-on. `Blender` not only provides a customizable mesh visualization environment, but also tools such as sculpt mode, which allows users to flexibly manipulate the geometry [3]. We briefly review the concepts behind the mesh processing algorithms from Yu et al. [9], [10].

1) Local Structure Tensor: The mesh processing operations in `GAMer` are designed to preserve the local geometry; algorithms seek to respect the geometric ground truth observed in the micrographs. We use a Local Structure Tensor (LST) to account for the local geometry [13]–[15]. The LST is defined as follows,

$$T(\mathbf{v}) = \sum_{i=1}^{N_r} \mathbf{n_i} \otimes \mathbf{n_i} = \sum_{i=1}^{N_r} \begin{pmatrix} n_i^x n_i^x & n_i^x n_i^y & n_i^x n_i^z \\ n_i^y n_i^x & n_i^y n_i^y & n_i^y n_i^z \\ n_i^z n_i^x & n_i^z n_i^y & n_i^z n_i^z \end{pmatrix}, \quad (1)$$

where $\mathbf{v}$ is the vertex of interest, N_r is the number of neighbors in the r-ring neighborhood, and $n_i^{x,y,z}$ form the normal of the ith neighbor vertex. Vertex normals are defined as the weighted average of incident face normals. Performing the eigendecomposition of the LST, we obtain information on the principal orientations of normals in the local neighborhood [16]. The magnitude of the eigenvalue corresponds to the amount of curvature along the direction of the corresponding eigenvector. Inspecting the magnitude of the eigenvalues gives several geometric cases:

- Planes: $\lambda_1 \gg \lambda_2 \approx \lambda_3 \approx 0$
- Ridges and valleys: $\lambda_1 \approx \lambda_2 \gg \lambda_3 \approx 0$
- Spheres and saddles: $\lambda_1 \approx \lambda_2 \approx \lambda_3 > 0$

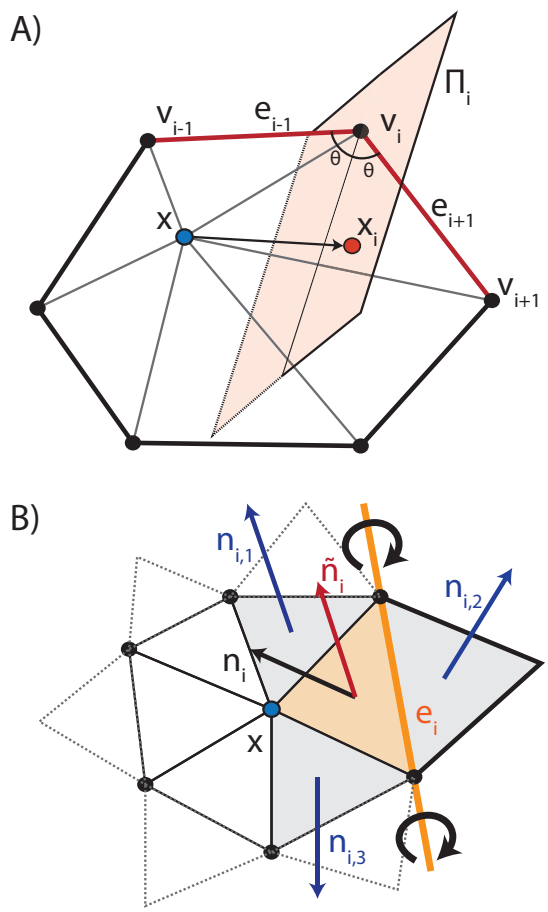

Fig. 3. Graphical description of the A) angle-based surface mesh conditioning, and B) anisotropic normal smoothing algorithms which are previously described by Yu et al. [9], [10] and implemented in `GAMer`.

2) Feature preserving mesh smoothing: Finite elements simulations are sensitive to the quality of the mesh. Poor quality meshes can lead to unbounded error, numerical instability, long times to solution, and non-convergence. Generally, triangulations with high aspect ratios produce larger errors compared with equilateral elements [17].

To improve the conditioning of the surface meshes derived from microscopy images, we use an angle-weighted Laplacian smoothing approach, as shown in Fig. 3A. This scheme is an extension to meshes embedded in 3D of the angle weighted smoothing scheme described by Zhou and Shimada for meshes embedded in 2D [18]. In essence, this algorithm applies local torsion springs to the 1-ring neighborhood of a vertex of interest to balance the angles.

Given a vertex $\mathbf{x}$ with the set of 1-ring neighbors $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$, where N is the number of neighbors, ordered

such that $\mathbf{v}_i$ is connected to $\mathbf{v}_{i-1}$ and $\mathbf{v}_{i+1}$ by edges. The 1-ring is connected such that $\mathbf{v}_{N+1} := \mathbf{v}_1$ and $\mathbf{v}_{-1} := \mathbf{v}_N$. Traversing the 1-ring neighbors, we define edge vectors $\mathbf{e}_{i-1} := \overrightarrow{\mathbf{v}_i\mathbf{v}_{i-1}}$ and $\mathbf{e}_{i+1} := \overrightarrow{\mathbf{v}_i\mathbf{v}_{i+1}}$. This algorithm seeks to move $\mathbf{x}$ to lie on the perpendicularly bisecting plane Π_i of $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$. For each vertex in the 1-ring neighbors, we compute the perpendicular projection, $\mathbf{x_i}$, of $\mathbf{x}$ onto Π_i. Since small surface mesh angles are more sensitive to change in $\mathbf{x}$ position than large angles, we prioritize their maximization. We define a weighting factor, $\alpha_i = \frac{\mathbf{e}_{i-1}\cdot\mathbf{e}_{i+1}}{|\mathbf{e}_{i-1}|\cdot|\mathbf{e}_{i+1}|}$, which inversely corresponds with $\angle(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$. The average of the projections weighted by α_i gives a new position of $\mathbf{x}$ as follows,

$$\bar{\mathbf{x}} = \frac{1}{\sum_{i=1}^{N}(\alpha_i + 1)} \sum_{i=1}^{N} (\alpha_i + 1)\mathbf{x}_i. \tag{2}$$

There are many smoothing algorithms in the literature; the angle-weighted Laplacian smoothing algorithm described here can outperform other popular smoothing strategies such as those described in [19] and [20] which are primarily focused on optimizing the smoothness of surface normals for computer graphics applications and not mesh angles. Our goal is not to provide an elaborate comparison against existing algorithms in this manuscript but to demonstrate the utility of our pipeline for biological images. These images can produce meshes that often contain hundreds of thousands to millions of faces and can cause global optimization based algorithms to fail. Therefore, our approach is a local operation, making it particularly suitable for cellular images.

Conceptually the fidelity of the local geometry can be maintained by restricting vertex movement along directions of low curvature. This constraint is achieved by anisotropically dampening vertex diffusion using information contained in the LST. Although the weighted vertex smoothing scheme, as described, will reasonably preserve geometric structure, the structure preservation can be further improved by using the LST. Computing the eigendecomposition of the LST, we obtain eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$, which correspond to principal orientations of local normals. We project $\bar{\mathbf{x}} - \mathbf{x}$ onto the eigenvector basis and scale each component by the inverse of the corresponding eigenvalue,

$$\hat{\mathbf{x}} = \mathbf{x} + \sum_{k=1}^{3} \frac{1}{1+\lambda_k}[(\bar{\mathbf{x}} - \mathbf{x}) \cdot \mathbf{E}_k]\mathbf{E}_k. \tag{3}$$

This has the effect of dampening movement along directions of high curvature i.e., where λ is large. In this fashion, our algorithm not only improves triangle aspect ratios, but does so while preserving local geometric features. We note that our actual implementation iterates between rounds of vertex smoothing and conventional angle based edge flipping to achieve the desired smoothing effect. Edge flips are common in mesh processing, and provide a mechanism for both improving angles and reducing the valency of vertices. A comparison of the angle-weighted smoothing algorithm with and without LST correction is shown in Fig. 4.

3) Feature preserving mesh decimation: The number of degrees of freedom in the mesh influences the computational burden of subsequent physical simulations. One strategy to reduce the number of degrees of freedom is to perform mesh decimation or simplification.

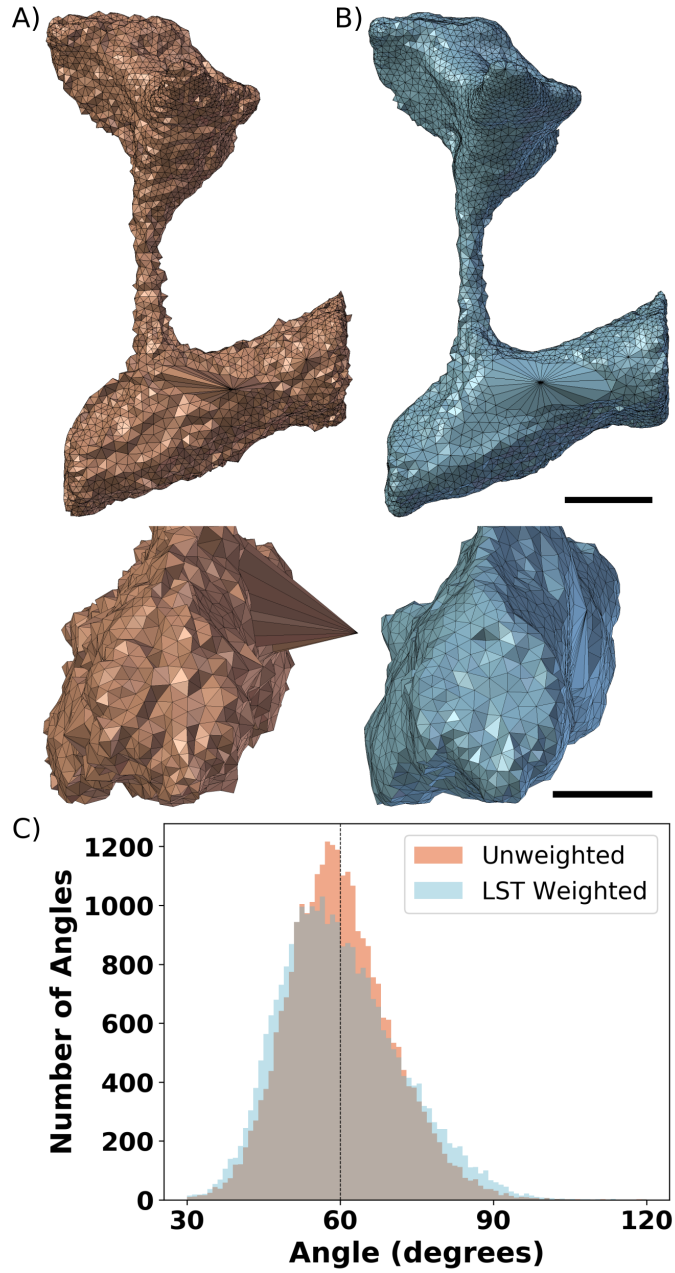

Fig. 4. Comparison of 50 iterations of angle weighted smoothing algorithm A) without and B) with Local Structure Tensor (LST) based correction. The LST helps to preserve the geometric structure albeit with slight degradation to the mesh angles. LST is a simple metric to capture local geometric information which can be used to constrain conditioning operations. C) Mesh angles are improved in both the LST weighted and unweighted meshes. Scale bar top row: 500 nm, bottom row: 250 nm.

There are many strategies for decimation including topology preserving Euler operators and other algorithms such as vertex clustering which do not guarantee topological invariance. It is typically desirable to preserve the mesh topology for physical simulation based applications. Conventional Euler operations for mesh decimation include vertex removal, edge collapse, and half-edge collapse. As noted earlier, finite elements sim-

ulations are sensitive to angles of the mesh. Edge and half-edge collapses can sometimes lead to vertices with high or low valency and therefore poor angles. To avoid this problem we employ the vertex removal algorithm. First, vertices to be decimated are selected based upon some criteria, discussed below. We then remove the vertex and re-triangulate the resulting hole. This is achieved using a recursive triangulation approach, which heuristically balances the edge valency. Given the boundary loop, we first connect vertices with the fewest incident edges. This produces two resulting holes that we then fill recursively using the same approach. When a hole contains only three boundary vertices, they are connected to make a face. We note that while this triangulation scheme balances vertex valency, it may degrade mesh quality. We solve this by running the geometry preserving smoothing algorithm on the local region.

We employ two criteria for selecting vertices to remove. First, to selectively decimate vertices in low or high curvature regions, again information from the LST can be used. By comparing the magnitudes of the eigenvalues of the LST we can select for regions with different geometries. For example, to decimate vertices in flat regions of the mesh, given eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, vertices can be selected by checking if the local region satisfies,

$$\frac{\lambda_2}{\lambda_1} < R_1, \tag{4}$$

where R_1 is a user specified flatness threshold (smaller is flatter). In a similar fashion, vertices in curved regions can also be selected. However, decimation of curved regions is typically avoided due to the potential for losing geometric information.

Instead, to simplify dense areas of the mesh, we employ an edge length based selection criterion,

$$\frac{\max_{i=1}^{N_1} d(\mathbf{x}, \mathbf{v}_i)}{\overline{D}} < R_2, \tag{5}$$

where N_1 is the number of vertices in the 1-ring neighborhood of vertex $\mathbf{x}$, $d(\cdot,\cdot)$ is the distance between vertices $\mathbf{x}$ and $\mathbf{v}_i$, $\overline{D}$ is the mean edge length of the mesh, and R_2 is a user specified threshold. This criterion allows us to control the sparseness of the mesh. We note that the aforementioned criteria are what is currently implemented in `GAMer`, however the vertex removal decimation scheme can be employed with any other selection criteria.

4) Feature preserving anisotropic normal-based smoothing: To remove additional bumpiness from the mesh, we use a normal-based smoothing approach [21], [22], as shown in Fig. 3B. The goal is to produce smoothly varying normals across the mesh without compromising mesh angle quality. Given a vertex $\mathbf{x}$ of interest, for each incident face i, with normal $\mathbf{n}_i$ we rotate $\mathbf{x}$ around a rotation axis defined by opposing edge e_i such that $\mathbf{n}_i$ aligns with the mean normal of neighboring faces $\bar{\mathbf{n}}_i = \sum_{j=1}^{3} \mathbf{n}_{ij}/3$. We denote the new position which aligns $\mathbf{n}_i$ and $\bar{\mathbf{n}}_i$ as $R(\mathbf{x}; e_i, \theta_i)$. Summing up the rotations and weighting by incident face area, a_i, we get an updated position,

$$\bar{\mathbf{x}} = \frac{1}{\sum_{i=1}^{N_1} a_i} \sum_{i=1}^{N_1} a_i R(\mathbf{x}; e_i, \theta_i). \tag{6}$$

This is an isotropic scheme which is independent of the local geometric features; meaning that many iterations of this algorithm may weaken sharp features.

Instead, we use an anisotropic scheme [22], [23] to compute the mean neighbor normals,

$$\bar{\mathbf{n}}_i = \frac{1}{\sum_{j=1}^{3} e^{K(\mathbf{n}_i \cdot \mathbf{n}_{ij})}} \sum_{j=1}^{3} e^{K(\mathbf{n}_i \cdot \mathbf{n}_{ij})} \mathbf{n}_{ij}, \tag{7}$$

where K is a user defined positive parameter which scales the extent of anisotropy. Under this scheme, the weighting function decreases as a function of the angle between $\mathbf{n}_i$ and $\mathbf{n}_{ij}$ resulting in the preservation of sharp features.

5) Boundary marking and tetrahedralization: To support the definition of boundary conditions on the mesh, it is conventional to assign boundary marks or identifiers which correspond to different boundary definitions in the physical simulation. In simplified and idealized geometries it is possible to define functions to assign boundary values. However, in subcellular scenes where the geometry may be tortuous and local receptor clusters can be arbitrarily distributed on the manifold, boundary definition is a non-trivial challenge. The `GAMer` `Blender` add-on supports the facile user-based definition of boundary markers on the surface [3]. Users can utilize any of the face selection methods which `Blender` provides to select boundaries to mark. Boolean operations and other geometric strategies provided natively in `Blender` can also be used for selection. Boundary markers are associated with a unique material property which helps visually delineate marked assignments. After boundaries are marked, stacks of surface meshes corresponding to different domains can be grouped and passed through `GAMer` into `TetGen` for tetrahedralization [4].

B. Mesh Generation Pipeline

Electron micrographs and segmentations from Wu *et al.* [1] were graciously shared by de Camilli and coworkers (Fig. 1A, B). We generated preliminary meshes of the geometry using the `imod2obj` utility included with `IMOD` [2] (Fig. 1C). The quality of the mesh was improved using algorithms described in Section III-A and implemented in `GAMer` 2.0.1 [12] (Fig. 1D). Some features, such as disconnections of the Endoplasmic Reticulum (ER), were manually reconnected using `Blender` mesh sculpting features. Additional discussion of the mesh artifacts and the curation process is described in the example applications. Boundaries were marked and the conditioned surface mesh was tetrahedralized using `TetGen` [4] (Fig. 1E). Simulations of a reaction-diffusion system were run for a full dendritic branch geometry using `FEniCS` [24] (Fig. 1F).

IV. Application of the Pipeline to Different Systems

We demonstrate the application of our pipeline to dendritic spine reconstructions of different sizes. All images of dendritic spines are from neurons taken from mouse cerebral cortex or nucleus accumbens from a recent publication using Focused-ion Beam Milling Scanning Electron Microscopy (FIB-SEM)

TABLE I
VERTEX AND ELEMENT COUNTS FOR MESHED GEOMETRIES

		Surface Mesh		Volume Mesh*	
		# Vertices	# Triangles	# Vertices	# Tetrahedra
Single Spine	Initial PM	4,695	9,330	–	–
	Conditioned PM	6,924	13,844	13,734	62,557
	Initial ER	6,546	19,654	–	–
	Conditioned ER	36,294	72,620	53,134	211,018
Two Spines	Initial PM	160,733	320,976	–	–
	Conditioned PM	18,027	36,050	28,989	122,082
Dendritic Segment	Initial PM	207,448	410,896	–	–
	Conditioned PM	126,336	252,668	194,848	798,626

*Non-manifold and other mesh artifacts prevent the tetrahedralization of initial meshes

[1]. In addition to their important role in synaptic and structural plasticity, these cellular structures demonstrate highly tortuous morphologies, high surface-to-volume ratios, and a geometric intricacy that serves as a good test-bed for our pipeline.

We have generated Finite Element Analysis (FEA) compatible meshes of several geometries of increasing length scale: the ER of the single spine geometry which requires nanometer precision (Fig. 5E, F), the Plasma Membrane (PM) of the single spine which has a length scale of a couple of microns (Fig. 5D), the two spine geometry, a few microns (Fig. 6), and the dendrite with about 40 spines, with a length scale in the tens of microns (Fig. 7) [1]. We describe some of the challenges associated with the mesh conditioning process for each system. We demonstrate the utility of this pipeline by showing a simulated time-series of calcium influx due to *N*-methyl-D-aspartate Receptor (NMDAR) by a Back Propagating Action Potential (BPAP) and Excitatory Postsynaptic Potential (EPSP) in the full dendritic segment geometry (Figs. 8 and 9). We also show a simulation of a generic reaction-diffusion equation in these geometries, and evaluate convergence with respect to `GAMer` mesh conditioning algorithms (Fig. 10).

A. Meshing segments of increasing length scales

We have constructed a mesh of a single spine geometry based on EM images taken of the mouse cerebral cortex, shown in Fig. 5. This geometry contains two membranes: the PM, Fig. 5A, and that of the ER, Fig. 5B. Due to differences in the morphology of the PM versus the ER, each component presents with it different meshing challenges. The PM contains several large holes on the surface which correspond to the top and bottom of the image stack. As the dendrite meanders throughout the tissue block, the experimental choice of sample cutting planes may result in truncation of the image of the structure. We have remedied this truncation by triangulating the holes and smoothing out the local curvature.

The ER mesh contained many more initial artifacts. This is because the detailed and variable nature of the ER membrane can be poorly resolved by the imaging method. Nixon-Abel and coworkers found using superresolution fluorescence microscopy that ER tubules have a diameter of 50 to 100 nm and sheet-like structures at the cellular periphery can be much finer [25]. Some ER morphology cannot even be resolved by the powers of EM [1].

For example, if the ER undergoes large conformational changes between z-slices then tubules may appear disconnected. Alternatively, the boundaries of the ER membrane may have poor contrast and can sometimes be missed during segmentation. In these cases, we manually curate the mesh to reconnect the broken ER segments, Fig. 5B. Fig. 2A1 and A2 illustrate this process in more detail. This spine also contains a specialized form of ER termed the spine apparatus, Fig. 5C, F, which consists of seven folded cisternae. This highly organized structure bears geometrical similarities to a parking garage structure and helicoidal geometries [26]–[29]. The geometric detail of the spine apparatus is preserved by the conditioning process in our pipeline.

In Fig. 5, we show the distribution of the angles of the surface mesh before and after conditioning. One metric of a well-conditioned mesh is that all the surface triangles are nearly equilateral [17]. Prior to conditioning, the angle distribution is spread out and contains many large and small angles. After processing using `GAMer`, the angles of the mesh are improved, as indicated by the peaked distribution around 60 degrees. Fig. 2C shows a closeup of high aspect ratio triangles before and after processing with `GAMer`.

It is worth noting that even though this mesh of PM of the single spine is the simplest geometry we examine, there are many intersecting faces in the initial mesh. This is a common artifact which appears in most initial meshes but is easily and automatically resolved by `GAMer`'s vertex smoothing algorithm as shown in Fig. 2B.

Although the ER structure is significantly more complex, the angles of the mesh are also improved, albeit to a lesser extent than the PM. In scenarios such as this where the length scales of interest are closer to the acquisition resolution, it may be necessary to increase the number of triangles to accurately capture the fine details. Table I summarizes the number of vertices and triangles in the initial vs conditioned meshes as well as vertices and tetrahedra in the resultant volumetric meshes. To accurately capture the curvature of the PM mesh in Fig. 5D about 48% more triangles were needed compared to

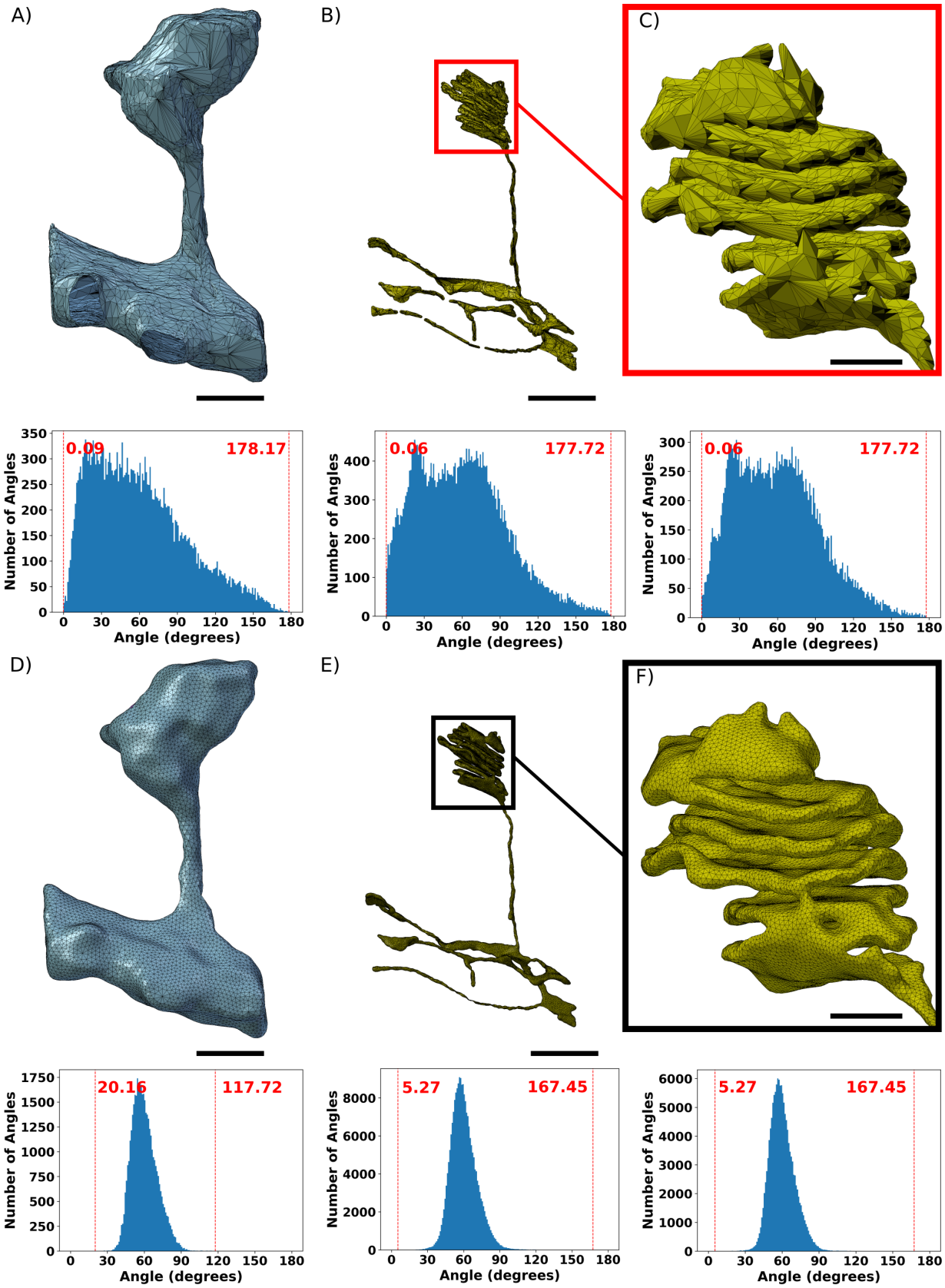

Fig. 5. Meshes of the single spine geometry pre-(A, B, C) and post-conditioning (D, E, F). Columns show (left) plasma membrane, (center) spine apparatus and endoplasmic reticulum, and (right) zoom in of spine apparatus. Note that there are two large holes in the preliminary plasma membrane mesh (A) corresponding to the start of the sample block-face. The distribution of angles for each mesh is shown. After processing in `GAMer`, the minimum and maximum angles of the mesh are improved and the angle distribution becomes peaked at 60 degrees. Scale bars: left, middle: 500 nm, right: 100 nm

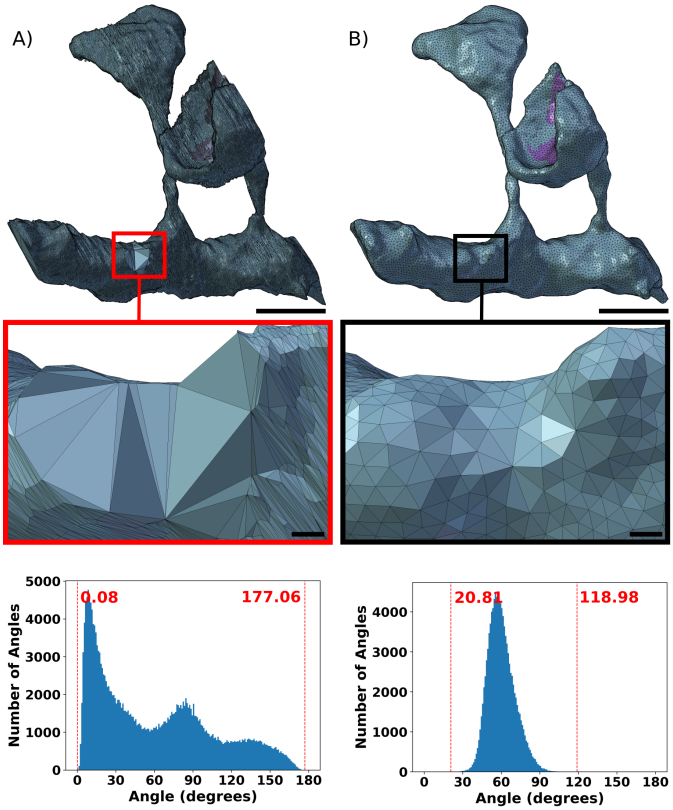

Fig. 6. Surface meshes of two neighboring dendritic spines. A) before conditioning and B) after. The angles of the mesh are improved by GAMer processing. Scale bars: top row: 1 μm, middle row: 50 nm.

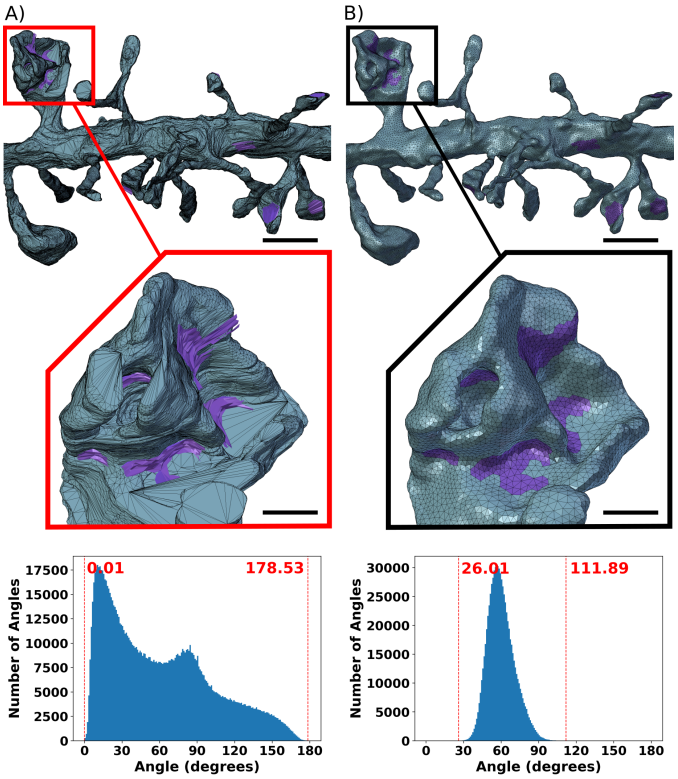

Fig. 7. Zoomed view of portion of dendrite segment with many spines. A) before conditioning and B) after. The angles of the mesh are improved by GAMer processing. Scale bars: top row: 1 μm, middle row: 250 nm.

the ER mesh in Fig. 5E which required 270% more triangles, both relative to the initial mesh.

The pipeline described here is also applicable for larger systems as we demonstrate with two spines and a full dendrite. The two spine geometry shown in Fig. 6 is a few microns in length. Based on the length scales we would expect a well conditioned mesh for this geometry to contain approximately double the number of triangles in the single spine mesh; however, the orientation of z-stacks in this mesh is different from that in the single spine geometry which led to an abnormally large number of triangles: 320,976 versus just 9,330 in the mesh of PM in the single spine. After GAMer conditioning algorithms were applied, the number of triangles was reduced to 36,050, a much more reasonable count. As demonstrated, our pipeline is robust and can handle cases where the initial mesh either generates too few or too many triangles as required for capturing geometric details.

At the tens of microns length scale, we constructed a mesh of a full dendritic segment. We show a zoomed in section of the mesh before and after conditioning in Fig. 7. As in the one and two spine cases, our system robustly handles artifacts such as poor quality triangles and intersecting faces; Fig. 7F shows that the distribution of the angles post conditioning are comparable to the one and two spine examples, showing that size does not alter the pipeline's capability to produce well-conditioned meshes. Fig. 7B shows an intricate spine head with many different regions of Postsynaptic Density (PSD) shown in purple; Fig. 7E shows the preserved geometry post-conditioning and the PSD marked with GAMer for use as boundary conditions.

B. Simulation of reaction-diffusion equations in dendrites

Using the mesh of the dendritic segment (Fig. 7) we simulated NMDAR activation due to a BPAP and EPSP along the entire dendrite in Fig. 8, see also Movie S1 available online. Because the goal of this simulation was not to show biological accuracy, but rather to demonstrate that our pipeline is capable of producing biophysically relevant FEA simulations, we use a simplified version of the model found in Bell et al. [30].

We model a BPAP and EPSP which stimulates NMDAR opening and calcium ion influx into the cell. The reaction-diffusion of u, corresponding to calcium ion concentration, is described as follows,

$$\frac{\partial u}{\partial t} = D\nabla^2 u - \frac{u}{\tau} \quad \text{in} \quad \Omega, \tag{8}$$

where D is the diffusion coefficient, ∇^2 is the Laplacian operator, τ is a characteristic decay time, and Ω is the volumetric domain. We define boundary conditions corresponding to the ionic flux through NMDA receptors, J_{NMDAR}, lining the post synaptic density, $\partial\Omega_{\text{psd}}$,

$$D(\mathbf{n} \cdot \nabla u) = J_{\text{NMDAR}}(t) \quad \text{on} \quad \partial\Omega_{\text{psd}}, \tag{9}$$

where $\mathbf{n}$ is the outwardly-oriented unit normal vector, and J_{NMDAR} is of the form,

$$J_{\text{NMDAR}} = G_{\text{NMDAR}}(t)B(V)(V(t) - V_{\text{rest}})\alpha. \tag{10}$$

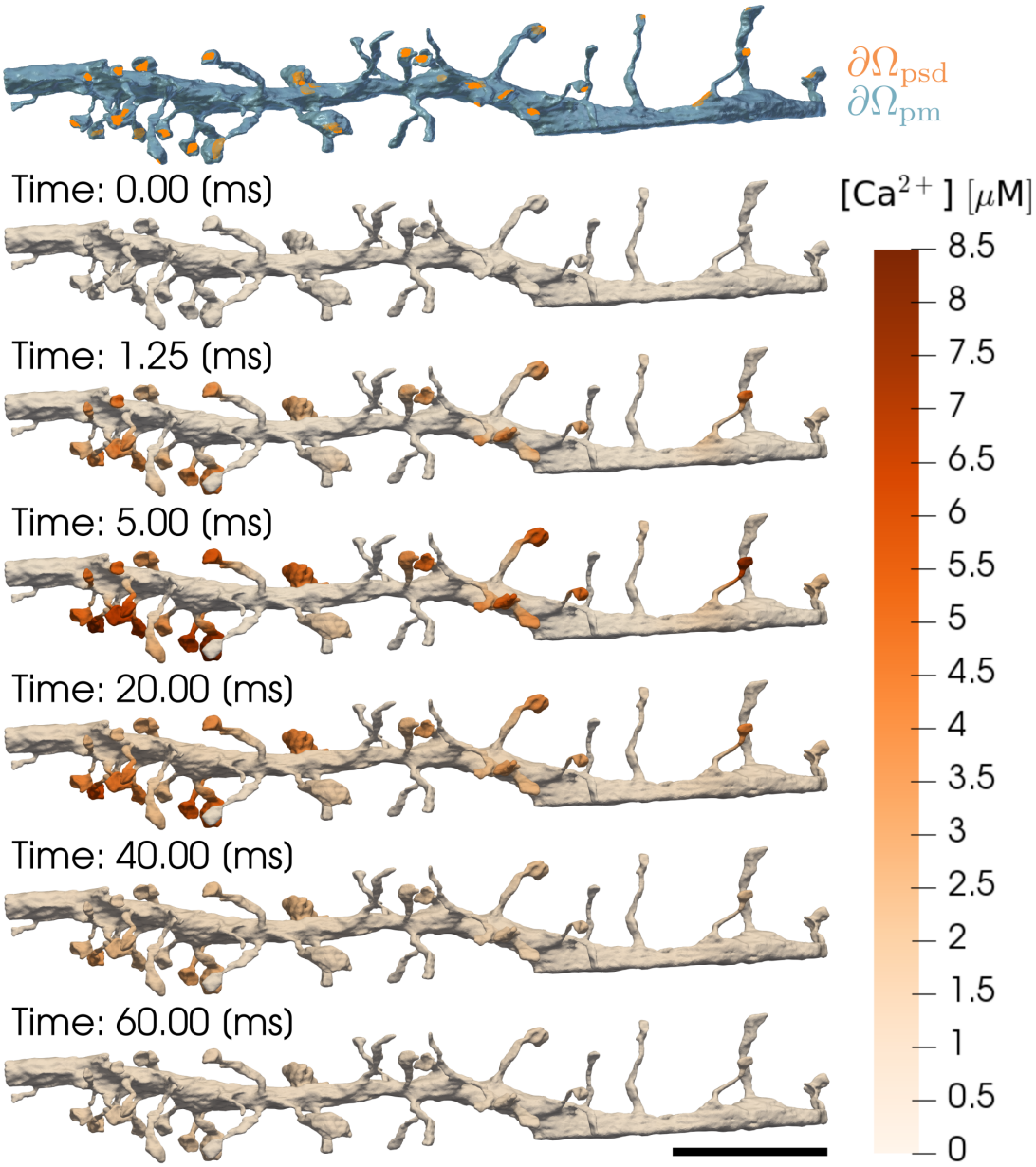

Fig. 8. Time series of calcium dynamics from NMDA receptor opening in response to a prescribed membrane voltage trace in a full dendritic segment. Boundaries demarcating the plasma membrane (PM) and post synaptic density (PSD) are shown in blue and orange respectively (top). Snapshots of calcium ion concentration throughout the domain are also shown for several time points. We apply a voltage corresponding to a back propagating action potential and excitatory postsynaptic potential. NMDA receptors localized at the PSD membrane, open in response to the voltage and calcium flows into the cell. Over time, the NMDA receptors close, and calcium is scavenged by calcium buffers. Scale bar: $4\,\mu$m.

$G_{\text{NMDAR}}(t)$ is a variable conductance which accounts for deactivation of the receptor, $B(V)$ is a term which accounts for Mg^{2+} inhibition, the voltage difference $V(t) - V_{\text{rest}}$ is prescribed to emulate a BPAP and EPSP, and α is a scaling term which groups factors such as probability of opening, receptor area density at the PSD, etc.

On the remainder of the plasma membrane which we denote as $\partial\Omega_{\text{pm}}$, we enforce no-flux boundary conditions,

$$D(\mathbf{n} \cdot \nabla u) = 0 \quad \text{on} \quad \partial\Omega_{\text{pm}}. \tag{11}$$

At time $t = 0$, we set the initial concentration of calcium ions to naught throughout the volume of the dendrite,

$$u(\mathbf{x}, t = 0) = 0 \quad \text{in} \quad \bar{\Omega}. \tag{12}$$

Where $\bar{\Omega}$ is the union of the volumetric and boundary domains,

$$\bar{\Omega} \equiv \Omega \cup \partial\Omega. \tag{13}$$

The surface of the geometry is composed of only post synaptic density and plasma membrane,

$$\partial\Omega \equiv \partial\Omega_{\text{psd}} \cup \partial\Omega_{\text{pm}}. \tag{14}$$

The model was solved using `FEniCS` [24] to run finite elements simulations. A backward Euler time discretization

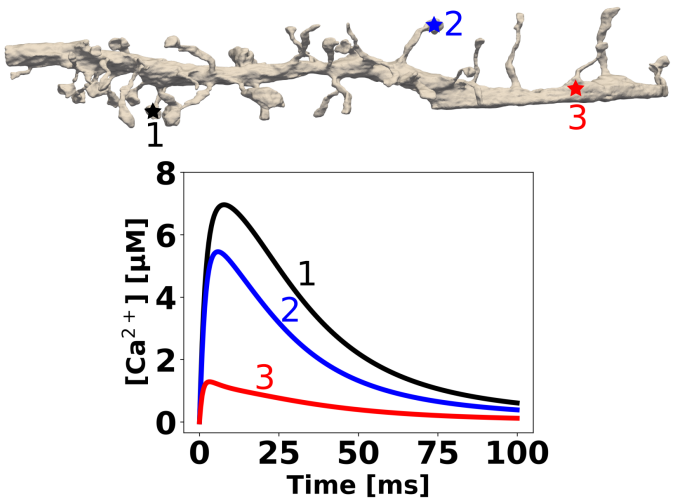

Fig. 9. Representative traces of Ca^{2+} concentration over time at three positions. Spine and PSD morphology affect the calcium ion dynamics.

scheme was used for its property of unconditional stability when solving linear problems; a timestep size of $417\,\mu$s was used. The resultant linear system of equations was solved using LU-decomposition.

The back propagating potential stimulates the entirety of the dendritic branch simultaneously, leading to the opening of NMDARs localized to the PSD. Calcium ions enter into the dendritic spine heads through the open NMDA receptors localized at the post synaptic density. We first observe that the calcium dynamics in spines are spine size, spine shape and PSD-dependent. We observe that the narrow spine necks act as a diffusion barrier to calcium, preventing diffusing calcium ions from entering the dendritic shaft, consistent with observations in the literature [31]–[34]. Fig. 9 shows several representative traces of calcium ion dynamics in different regions of the geometry.

This example demonstrates that the meshes produced by `GAMer` through the workflow are directly compatible with finite element simulations and will allow for the generation of biophysically relevant hypotheses.

C. Mesh convergence analysis

A common error metric used in FEA is the L_2 norm of the difference between a solution computed on a coarser mesh (u') and a solution computed on a very fine mesh, which is taken to be the ground-truth (u), i.e.,

$$\varepsilon_{L_2} = \left(\int_{\Omega} (u' - u)^2 \, d\Omega \right)^{\frac{1}{2}}. \tag{15}$$

This is a standard procedure that is used to measure h-refinement convergence rates; however between iterations of `GAMer` algorithms the boundaries of the mesh are perturbed slightly. Attempting to use ε_{L_2} as an error metric is problematic as its integrand is undefined in regions where Ω', the domain of u', and Ω do not intersect.

Therefore, to illustrate the convergence of solutions as the mesh quality is improved using `GAMer`, we used an error metric based on the relative difference in total molecules,

$$\varepsilon_{\text{rel}} = \left| \frac{\int_{\Omega'} u' \, d\Omega' - \int_{\Omega} u \, d\Omega}{\int_{\Omega} u \, d\Omega} \right|. \tag{16}$$

Fig. 10 (A-D) shows intermediary steps of the `GAMer` refinement process. For each mesh, a given number of smoothing iterations was performed; any remaining artifacts that would prevent tetrahedralization such as intersecting faces were removed and the resultant holes were re-triangulated. The surface meshes were all tetrahedralized using `TetGen` with the same parameters. The first time step of a reaction-diffusion partial differential equation with a constant Neumann boundary condition was solved and the resulting solutions were compared using Eq. (16), where Fig. 10E was assumed to be the ground-truth. Fig. 10F shows that the relative error consistently decreased as a result of further mesh conditioning in `GAMer`. The mesh with no smoothing operations applied had almost five times more molecules than the ground-truth mesh after just one time step. This analysis highlights not only the importance of using a high quality mesh in FEA but also that `GAMer` can generate such high quality meshes.

V. Discussion

The relationship between cellular shape and function is being uncovered as systems, structural biology, and physical simulations converge. Beyond traditional compartmentalization, plasma membrane curvature and cellular ultrastructure have been shown to affect the diffusion and localization of molecular species in cells [35], [36]. For example, fluorescence experiments have shown that the dendritic spine necks act as a diffusion barrier to calcium ions, preventing ions from entering the dendritic shaft [31]. Complementary to this and other experiments, various physical models solving reaction-diffusion equations in idealized geometries have been developed to further interrogate the structure-function relationships [30], [35], [37]–[40].

An important next step will be to expand the spatial realism of these models to incorporate realistic geometries as informed by volume imaging modalities. Our tool `GAMer` serves as an important step towards filling the need for community driven tools to generate meshes from realistic biological scenes. We have demonstrated the utility of the mesh conditioning algorithms implemented in `GAMer` for a variety of systems across several length scales and upwards of hundreds of thousands of triangles. The volume meshes that result from our tools are high quality (Fig. 1E) and we show that they can be used directly in finite elements simulations of reaction-diffusion systems (Fig. 8).

Bundled with `GAMer` we include a `Blender` add-on which exposes our mesh conditioning algorithms to the `Blender` environment. `Blender` acts as a user interface that provides visual feedback on the effects of `GAMer` mesh conditioning operations. `Blender` also enables the painting of boundaries using its many mesh selection tools. Beyond the algorithms in `GAMer`, `Blender` also provides an environment for manual curation of mesh artifacts.

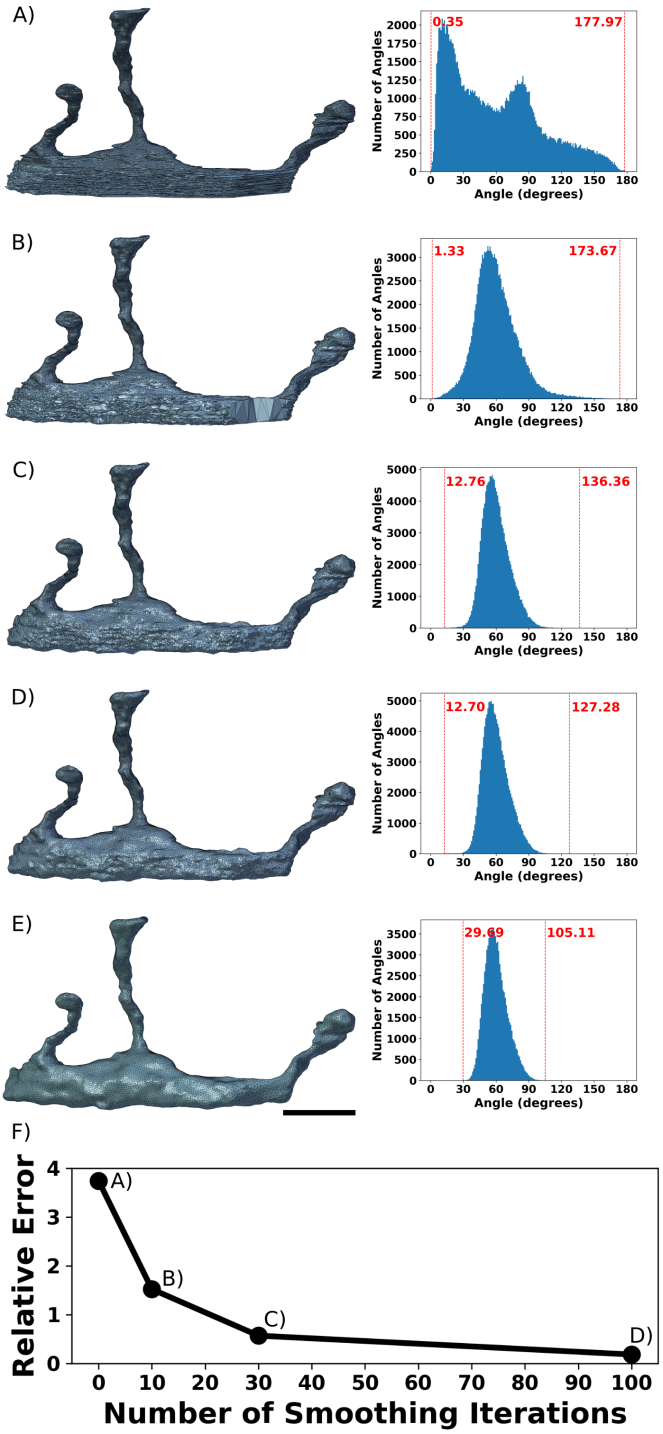

Fig. 10. (A-D) Mesh after 0, 10, 30, 100 smoothing operations (artifacts which prevented tetrahedralization, e.g. intersecting faces, were removed and the holes were re-meshed). E) shows the finalized mesh. F) Relative error was calculated using Eq. (16) using the final form of the mesh, (E), as the ground truth. Scale bar = 1 μm.

Current meshing methods are limited by the need for human biological insight. Experimental setups for volume electron microscopy are arduous and often messy. Microscopists take great care to optimize the experimental conditions, however small variatons can lead to sample contamination, tears, precipitation of stain, or other problems. Many of these issues will manifest as artifacts on the micrographs, which makes it challenging to evaluate the ground truth. Automated segmentation algorithms using computer vision and machine learning approaches can fail as a result of these artifacts, and biologists will default to the time-tested, reliable but error-prone mode of manually tracing boundaries.

This is a unique opportunity for biological mesh generation to differentiate from other meshing tools employed in other engineering disciplines. To account for the problems induced by biology and wet experiments along with physical simulations, we anticipate that, to realize an automated mesh generation pipeline will require the development of specialized algorithms which tightly couple information across the workflow. As additional annotated datasets become available, machine learning models can be trained to perform tasks which are currently manually executed such as reconnecting disconnected ER tubules.

To accelerate the goal of achieving routine cell modeling with realistic geometries, experimentalists can contribute by sharing segmented datasets from their work along with biological questions of interest. In exchange, modelers can generate testable predictions and measurements inaccessible to current experimental methods. Through this interdisciplinary exchange, any gaps in our current meshing workflows can be identified and patched.

VI. Conclusions

In this study, we have described a computational pipeline that uses contours from electron microscopy images as input to generate surface and volume meshes that are compatible with finite element simulations for reaction-diffusion processes. Using this pipeline, we have demonstrated the spatio-temporal dynamics of calcium influx in multiple spines along a dendrite. Future efforts will focus on the development of biologically relevant models and generation of experimentally testable hypotheses.

Acknowledgments

We would like to thank Prof. Pietro De Camilli and coworkers for sharing their datasets from Wu et al. [1], Dr. Tom Bartol, Dr. Matthias Haberl, Mr. Evan Campbell, Profs. Brenda Bloodgood and Mark Ellisman for many enlightening discussions during the course of this work. CTL especially thanks Dr. John B. Moody, and Mr. Mason V. Holst for discussions on GAMer 2 code development and SWIG wrapping. We thank Dr. Tom Bartol for additional help with using Blender and the design of Blender addons. We thank Ms. Miriam Bell, Ms. Kiersten Scott, Ms. Jennifer Fromm, and Dr. Donya Ohadi for critical comments and suggestions for improving this manuscript.

References

[1] Y. Wu, C. Whiteus, C. S. Xu, K. J. Hayworth, R. J. Weinberg, H. F. Hess, and P. De Camilli, "Contacts Between the Endoplasmic Reticulum and Other Membranes in Neurons," *Proc. Natl. Acad. Sci. USA*, vol. 114, no. 24, pp. E4859–E4867, jun 2017.

[2] J. R. Kremer, D. N. Mastronarde, and J. McIntosh, "Computer Visualization of Three-Dimensional Image Data Using IMOD," *J. Struct. Biol.*, vol. 116, no. 1, pp. 71–76, jan 1996.

[3] Blender Online Community, *Blender - a Free and Open-Source 3D Computer Graphics Softare Toolset*, Blender Foundation, Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[4] H. Si, "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," *ACM Trans. Math. Softw.*, vol. 41, no. 2, pp. 1–36, feb 2015.

[5] C. J. Peddie and L. M. Collinson, "Exploring the Third Dimension: Volume Electron Microscopy Comes of Age," *Micron*, vol. 61, pp. 9–19, jun 2014.

[6] B. Titze and C. Genoud, "Volume Scanning Electron Microscopy for Imaging Biological Ultrastructure," *Biol. Cell*, vol. 108, no. 11, pp. 307–323, nov 2016.

[7] S. Borrett and L. Hughes, "Reporting Methods for Processing and Analysis of Data from Serial Block Face Scanning Electron Microscopy," *J. Microsc.*, vol. 263, no. 1, pp. 3–9, jul 2016.

[8] W.-T. Tsai, A. Hassan, P. Sarkar, J. Correa, Z. Metlagel, D. M. Jorgens, and M. Auer, "from Voxels to Knowledge: A Practical Guide to the Segmentation of Complex Electron Microscopy 3D-Data," *J. Vis. Exp.*, no. 90, aug 2014.

[9] Z. Yu, M. J. Holst, Y. Cheng, and J. McCammon, "Feature-Preserving Adaptive Mesh Generation for Molecular Shape Modeling and Simulation," *J. Mol. Graph. Model.*, vol. 26, no. 8, pp. 1370–1380, jun 2008.

[10] Z. Yu, M. J. Holst, and J. Andrew McCammon, "High-Fidelity Geometric Modeling for Biomedical Applications," *Finite Elem. Anal. Des.*, vol. 44, no. 11, pp. 715–723, jul 2008.

[11] C. T. Lee, J. B. Moody, R. E. Amaro, J. A. McCammon, and M. Holst, "The implementation of the colored abstract simplicial complex and its application to mesh generation," *arXiv*, vol. abs/1807.01417, 2018. [Online]. Available: http://arxiv.org/abs/1807.01417

[12] C. T. Lee, J. B. Moody, and M. J. Holst, "GAMer 2.0 Software." [Online]. Available: https://github.com/ctlee/gamer

[13] H. Knutsson, "Representing Local Structure Using Tensors," *Comput. Vis. Lab. LINKOPING Univ.*, pp. 244–251, 1989.

[14] H. Haußecker and B. Jähne, "A Tensor Approach for Local Structure Analysis in Multi-Dimensional Images," in *3D Image Analysis and Synthesis*, G. Girod, H. Niemann, and H. P. Seidel, Eds., 1996, pp. 171–178.

[15] J.-J. Fernández and S. Li, "an Improved Algorithm for Anisotropic Nonlinear Diffusion for Denoising Cryo-Tomograms," *J. Struct. Biol.*, vol. 144, no. 1-2, pp. 152–161, oct 2003.

[16] J. Weickert, *Anisotropic Diffusion in Image Processing*. B. G. Teubner, 1998.

[17] J. R. Shewchuk, "What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures," *Proc. 11th Int. Meshing Roundtable*, vol. 94720, pp. 115–126, 2002.

[18] T. Zhou and K. Shimada, "An Angle-Based Approach to Two-Dimensional Mesh Smoothing," *Proc. 9th Int. Meshing Roundtable*, 2000.

[19] G. Taubin, "a Signal Processing Approach to Fair Surface Design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '95*, 1995.

[20] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '99*, 1999.

[21] C. Y. Chen and K. Y. Cheng, "A Sharpness Dependent Filter for Mesh Smoothing," *Comput. Aided Geom. Des.*, 2005.

[22] Z. Yu and C. Bajaj, "A Segmentation-Free Approach for Skeletonization of Gray-Scale Images Via Anisotropic Vector Diffusion," in *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. 415–420.

[23] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990.

[24] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The FEniCS Project Version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[25] J. Nixon-Abell, C. J. Obara, A. V. Weigel, D. Li, W. R. Legant, C. S. Xu, H. A. Pasolli, K. Harvey, H. F. Hess, E. Betzig, C. Blackstone, and J. Lippincott-Schwartz, "Increased Spatiotemporal Resolution Reveals Highly Dynamic Dense Tubular Matrices in the Peripheral ER," *Science (80-.).*, 2016.

[26] M. Terasaki, T. Shemesh, N. Kasthuri, R. W. Klemm, R. Schalek, K. J. Hayworth, A. R. Hand, M. Yankova, G. Huber, J. W. Lichtman, T. A. Rapoport, and M. M. Kozlov, "Stacked Endoplasmic Reticulum Sheets Are Connected by Helicoidal Membrane Motifs," *Cell*, vol. 154, no. 2, pp. 285–296, jul 2013.

[27] W. F. Marshall, "Differential Geometry Meets the Cell," *Cell*, vol. 154, no. 2, pp. 265–266, jul 2013.

[28] T. Shemesh, R. W. Klemm, F. B. Romano, S. Wang, J. Vaughan, X. Zhuang, H. Tukachinsky, M. M. Kozlov, and T. A. Rapoport, "A Model for the Generation and Interconversion of ER Morphologies," *Proc. Natl. Acad. Sci. USA*, vol. 111, no. 49, pp. E5243–E5251, dec 2014.

[29] Y. Shibata, T. Shemesh, W. A. Prinz, A. F. Palazzo, M. M. Kozlov, and T. A. Rapoport, "Mechanisms Determining the Morphology of the Peripheral ER," *Cell*, vol. 143, no. 5, pp. 774–788, nov 2010.

[30] M. Bell, T. Bartol, T. Sejnowski, and P. Rangamani, "Dendritic Spine Geometry and Spine Apparatus Organization Govern the Spatiotemporal Dynamics of Calcium," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/08/07/386367

[31] B. L. Bloodgood, "Neuronal Activity Regulates Diffusion Across the Neck of Dendritic Spines," *Science (80-.).*, vol. 310, no. 5749, pp. 866–869, nov 2005.

[32] B. L. Bloodgood and B. L. Sabatini, "Ca2+ Signaling in Dendritic Spines," *Curr. Opin. Neurobiol.*, vol. 17, no. 3, pp. 345–351, jun 2007.

[33] B. L. Bloodgood, A. J. Giessel, and B. L. Sabatini, "Biphasic Synaptic Ca Influx Arising from Compartmentalized Electrical Signals in Dendritic Spines," *PLoS Biol.*, vol. 7, no. 9, p. e1000190, sep 2009.

[34] B. L. Bloodgood, B. L. Sabatini, and A. Van Dongen, *NMDa Receptor-Mediated Calcium Transients in Dendritic Spines*. Boca Raton, FL, USA: CRC Press, 2009.

[35] P. Rangamani, A. Lipshtat, E. U. Azeloglu, R. C. Calizo, M. Hu, S. Ghassemi, J. Hone, S. Scarlata, S. R. Neves, and R. Iyengar, "Decoding Information in Cell Shape," *Cell*, vol. 154, no. 6, pp. 1356–1369, sep 2013.

[36] R. C. Calizo, A. Ron, M. Hu, S. Bhattacharya, W. G. M. Janssen, J. Hone, S. Scarlata, P. Rangamani, and R. Iyengar, "Cell Shape Regulates Subcellular Organelle Location to Control Short-term Ca2+ Signal Dynamics in VSMC," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/10/31/161950

[37] S. R. Neves, P. Tsokas, A. Sarkar, E. A. Grace, P. Rangamani, S. M. Taubenfeld, C. M. Alberini, J. C. Schaff, R. D. Blitzer, I. I. Moraru, and R. Iyengar, "Cell Shape and Negative Links in Regulatory Motifs Together Control Spatial Information Flow in Signaling Networks," *Cell*, 2008.

[38] A. Cugno, T. M. Bartol, T. J. Sejnowski, R. Iyengar, and P. Rangamani, "Geometric Principles of Second Messenger Dynamics in Dendritic Spines," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/10/20/444489

[39] D. Ohadi, D. L. Schmitt, B. Calabrese, S. Halpain, J. Zhang, and P. Rangamani, "Computational Modeling Reveals Frequency Modulation of Calcium-cAMP/PKA Pathway in Dendritic Spines," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/01/16/521740

[40] D. Ohadi and P. Rangamani, "Geometric Control of Frequency Modulation of cAMP Oscillations Due to Ca2+-bursts in Dendritic Spines," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/01/15/520643