Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 403 (2023) 115757
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Isogeometric analysis-based physics-informed graph neural network for studying traffic jam in neurons

Angran Li[a], Yongjie Jessica Zhang[a,b,*]

[a] Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, 15213, PA, USA
[b] Department of Biomedical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, 15213, PA, USA

Received 7 July 2022; received in revised form 27 October 2022; accepted 29 October 2022
Available online 18 November 2022

Dataset link: https://github.com/CMU-CBML/NeuronTransportPGNN

Abstract

The motor-driven intracellular transport plays a crucial role in supporting a neuron cell's survival and function, with motor proteins and microtubule (MT) structures collaborating to promptly deliver the essential materials to the right location in neuron. The disruption of transport may lead to the onset of various neurodegenerative diseases. Therefore, it is essential to study how neurons regulate the material transport process and have a better understanding of the traffic jam formation. In our earlier work, we developed a PDE-constrained optimization model and an isogeometric analysis (IGA) solver to simulate traffic jams induced by MT reduction and MT swirl. Here, we develop a novel IGA-based physics-informed graph neural network (PGNN) to quickly predict normal and abnormal transport phenomena such as traffic jam in different neuron geometries. In particular, the proposed method learns from the IGA simulation of the intracellular transport process and provides accurate material concentration prediction of normal transport and MT-induced traffic jam. The IGA-based PGNN model contains simulators to handle local prediction of both normal and two MT-induced traffic jams in pipes, as well as another simulator to predict normal transport in bifurcations. Bézier extraction is adopted to incorporate the geometry information into the simulators to accurately compute the physics informed loss function with PDE residuals. Moreover, a GNN assembly model is adopted to tackle different neuron morphologies by assembling local prediction into the entire geometry. In summary, the well-trained model effectively predicts the distribution of transport velocity and material concentration during traffic jam and normal transport with an average error less than 10% compared to IGA simulations. The effectiveness of the proposed model is demonstrated within several complex neuron geometries.

Keywords: Neuron transport phenomena; Traffic jam; Microtubule swirl; Physics-informed neural network; Isogeometric analysis

1. Introduction

The neuron is a basic component of the nervous system with a single long axon and broadly branched dendrites extended from its soma. Its unique morphology is specialized for transmitting information throughout the body in both electrical and chemical forms. To maintain its normal function and healthy growth, a neuron relies heavily on

* Corresponding author at: Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, 15213, PA, USA.
E-mail address: jessicaz@andrew.cmu.edu (Y.J. Zhang).

https://doi.org/10.1016/j.cma.2022.115757

the long-range active transport of various organelles and proteins along its cytoskeleton [1–3]. When intracellular transport breaks down, the aggregation of materials and the excessive swelling may occur and contribute to neuron degeneration. Such defects in axonal transport have been observed in various neurodegenerative diseases such as Huntington's, Parkinson's, amyotrophic lateral sclerosis (ALS), and Alzheimer's disease [4–8]. Therefore, it is critical to investigate the underlying transport mechanism to have a better understanding of their pathologies.

It is well known that the motor proteins and microtubule (MT) structures interplay to ensure the spatiotemporal demands for on-time delivery of materials in neuron. MT is one of the major components of cytoskeleton that serves as a "cellular highway" for efficient intracellular transport, and the motor proteins are responsible for moving the materials along MTs. The transport direction is determined by the polarity of MT and the type of motor proteins, with kinesins and dyneins moving towards the MT plus-ends and minus-ends, respectively [9–11]. The molecular basis of the intracellular transport has resulted in a series of mathematical models to explain the regulation of transport activity in neurons. For instance, the partial differential equations (PDEs) based models were developed for fast and slow axonal transport by considering the reversible binding of organelles, kinesins, and microtubules [12,13]. The governing equations of these models are usually of reaction-hyperbolic type, which are further generalized to account for multiple moving materials [14] and their diffusion [15,16]. Moreover, these generalized reaction–diffusion-transport models were revamped to study different transport defects such as the segregation of MTs and neurofilaments in neurological diseases [17], the formation of traffic jam [18–20], the onset and progression of Alzheimer's disease [21]. One major drawback of these models is that the complex neuron morphologies are usually neglected and the PDEs are solved in simple one-dimensional (1D) or 2D domain. Recent advances in computational modeling techniques introduce new capabilities for accurately solving PDE-governed physics system within complex domains. For instance, isogeometric analysis (IGA) [22] was developed based on the standard finite element method (FEM), aiming to bridge the gap between computer-aided design (CAD) and analysis. The basic idea of IGA is to adopt the same smooth spline basis functions in both geometry representation and analysis, which leads to superior accuracy and computational performance compared to FEM. IGA has proved its success in many applications including shell analysis [23–26], cardiovascular modeling [27–32], neuroscience simulation [33,34], as well as industrial application [35,36]. With the advances in IGA, we studied the effect of neuron morphologies on the transport process by performing IGA simulation of a 3D motor-assisted transport model in different neuron geometries [37]. Built upon this framework, we proposed a PDE-constrained optimization (PDE-CO) model by taking into account the potential transport regulation mechanisms and simulated two MT-induced traffic jams in neurons [38,39]. These simulations reveal that the distribution of MTs and various neuron morphologies have major influence on regulating the material transport to deliver the material to the correct location. In addition, the distribution of material and transport velocity obtained from IGA modeling can help identify possible transport defects. However, the high computational cost of IGA simulation brings challenges to provide a real-time transport analysis for clinical research.

The recent emerging deep learning (DL) technique has shown great potentials in handling high dimensionality and become a popular surrogate modeling approach for finding PDE solutions. In particular, physics-informed neural networks (PINNs) [40] have attracted increasing attention for being an efficient and flexible paradigm to solve both forward and inverse PDEs. The general idea of PINNs is to approximate the PDE solutions with a deep neural network and its loss function contains PDE residuals as physical constraints. This unique loss formulation enables physics-informed training based on the information from governing equations and sparse observation data. For instance, in fluid applications, PINNs have been used for fast surrogate modeling of idealized vascular flow problems in a forward parametric setting without relying on simulation data [41]. PINNs have also been adopted in an inverse modeling setting to infer unobservable information given observable data in cardiovascular problems [42,43]. Moreover, the fully connected-PINNs (FC-PINNs) was applied and effectively solved Navier–Stokes equations of laminar and turbulent flow [44], as well as the high-speed flow problems [45]. Gao et al. [46] proposed a geometry-adaptive physics-informed CNN to embed a pre-computed coordinate mapping into the classic CNN structure and solve parametric PDEs on irregular domains with unstructured grids. In [47], we developed a graph neural network (GNN)-based surrogate model to learn from motor-assisted transport simulation data and achieved accurate prediction of the material transport process within different neuron geometries.

In this study, we develop a novel IGA-based physics-informed graph neural network (PGNN) that effectively predicts complex normal and abnormal material transport phenomena such as MT-induced traffic jams. The governing equations of a PDE-constrained optimization (PDE-CO) transport model is embedded into the PGNN

simulator to provide a priori knowledge of the transport phenomena. The well trained PGNN model can provide a fast and accurate prediction of the material concentration and velocity distribution under different transport conditions, leading to an efficient DL framework for studying the abnormal traffic jams in a local region of complex neuron trees. The framework was applied to several complex neuron geometries achieving an average error less than 10% compared to IGA simulations. Our model also successfully predicts material accumulation and complex patterns of velocity such as vortex and reversing streamline, which provide a key understanding into the formation of complex traffic jam. In summary, there are four main contributions in this paper.

- Four PGNN simulators are specifically designed to predict complex transport phenomena in a local simple geometry, including two traffic jams caused by MT reduction and MT swirls in pipes, as well as normal transport in pipes and bifurcations;
- The simulator loss function contains the residual of the governing equations, which are evaluated using the Bézier extraction information of spline basis from IGA. The geometry information is thus preserved during model training, which results in a more accurate computation of PDE residuals in the loss function;
- To tackle the extensive neuron geometries, a GNN assembly model is adopted by combining local prediction into the entire neuron geometry with a penalty term to maintain consistent interface between two adjacent simulators; and
- The proposed IGA-based PGNN method can be extended to a generic framework to learn and predict other PDE-CO models in any geometry.

The rest of the paper is organized as follows. In Section 2, we provide a review of the material transport model employed in this work. In Section 3, we discuss in detail the IGA-based PGNN method used for obtaining the prediction of material concentration and transport velocity given different transport conditions. In Section 4, we demonstrate the effectiveness of the proposed method for predicting normal transport and traffic jam phenomena in several 2D and 3D neuron geometries. Conclusions and future work are given in Section 5.

2. Review of the PDE-CO material transport model

The material transport phenomena can be effectively simulated with a macroscopic molecular-motor-assisted transport model [37]. Built upon this transport model, we developed a PDE-CO model to account for the potential transport regulation from neurons during material transport process and predict the formation of a traffic jam in abnormal neurons [38,39]. We assume that there are two major transport regulation mechanisms: (1) the neuron actively controls the transport velocity to achieve an ideal velocity field; and (2) the neuron actively detects material distribution to avoid high concentration gradient within a local region. Based on these assumptions, the proposed optimization problem is described as

$$\underset{n_\pm, \boldsymbol{v}_\pm, f_\pm}{\arg\min}\ \mathcal{J}(n_\pm, \boldsymbol{v}_\pm, f_\pm) = \frac{1}{2}\int_0^T \int_\Omega (\boldsymbol{v}_\pm - V_\pm)^2 d\Omega dt + \frac{\alpha}{2}\int_0^T \int_\Omega \|\nabla n_\pm\|^2 d\Omega dt + \frac{\beta}{2}\int_0^T \int_\Omega f_\pm^2 d\Omega dt \tag{1a}$$

subject to

$$\frac{\partial n_0}{\partial t} = -(k_+ + k_-)n_0 + k'_+ l_+ n_+ + k'_- l_- n_- \quad \text{in } \Omega \times (0, T], \tag{1b}$$

$$\frac{\partial (l_\pm n_\pm)}{\partial t} + \boldsymbol{v}_\pm \cdot \nabla (l_\pm n_\pm) = D_\pm \nabla^2 (l_\pm n_\pm) + k_\pm n_0 - k'_\pm (l_\pm n_\pm) \quad \text{in } \Omega \times (0, T], \tag{1c}$$

$$\frac{\partial \boldsymbol{v}_\pm}{\partial t} + \boldsymbol{v}_\pm \cdot \nabla \boldsymbol{v}_\pm = -\nabla n_\pm + \nabla \cdot (\mu \nabla \boldsymbol{v}_\pm) + f_\pm \quad \text{in } \Omega \times (0, T], \tag{1d}$$

$$n_0 = n_i, \quad n_+ = \lambda_i n_i \quad \text{in } \partial\Omega_{inlet} \times (0, T], \tag{1e}$$

$$n_0 = n_o, \quad n_- = \lambda_o n_o \quad \text{in } \partial\Omega_{outlet} \times (0, T], \tag{1f}$$

where the open set $\Omega \subset \mathbb{R}^d$ represents the d-dimensional internal space of the neuron; $V_\pm$ is the ideal velocity field that the neuron wants to achieve; n_0, n_+ and n_- are the spatial concentrations of free diffusive, anterograde, and retrograde materials, respectively; $D_\pm$ is the diffusion coefficient of anterograde and retrograde materials; $\boldsymbol{v}_+$ and $\boldsymbol{v}_-$ are velocities of anterograde and retrograde materials, respectively; $k_\pm$ and $k'_\pm$ are rates of MT attachment and detachment of anterograde and retrograde materials, respectively; $l_\pm$ represents the density of MTs used for motor-assisted transport; $f_\pm$ represents the control forces that mediate the material transport; and μ is viscosity of

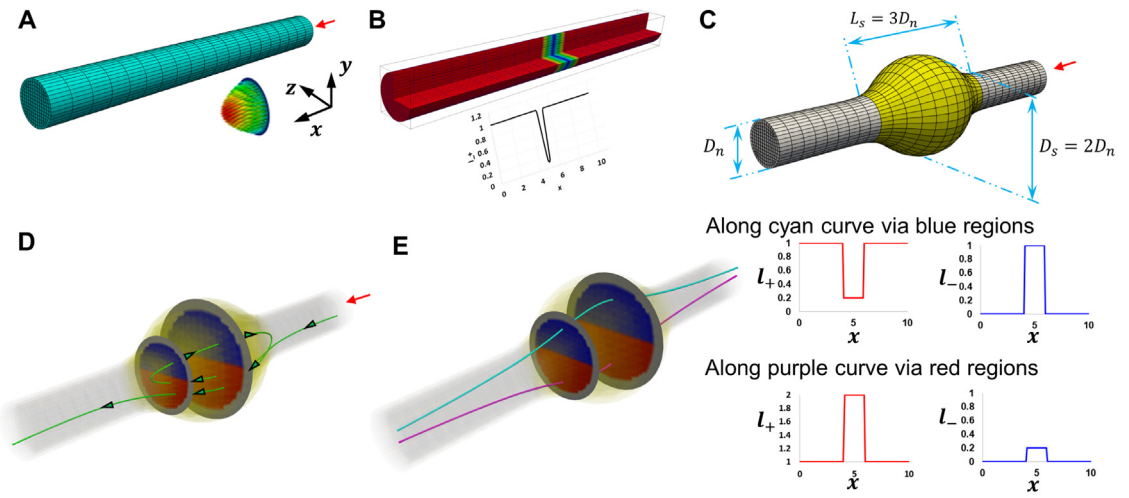

Fig. 1. Simulation settings for different transport conditions. (A) The control mesh for a single pipe geometry and the inlet velocity profile. Red arrow points to the inlet of material. (B) The spatial distribution of l_+ to represent the reduction of MTs in a local region. The curve plot describes the distribution of l_+ along x-axis. (C) The control mesh for a single pipe with swelling geometry. The normal diameter of the pipe is D_n. The yellow swelling region has a length of $L_s = 3D_n$ and its maximum diameter $D_s = 2D_n$. Red arrow points to the inlet of material. (D) An example of MT swirls in the swollen region. The green arrows show the transport path along swirly MTs. (E) The distribution of l_+ and l_- along the blue and red regions, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

traffic flow. Eqs. (1e) and (1f) impose stable concentrations of materials at the boundary of inlet and outlet ends. We have λ_i, λ_o represent the degree of loading at inlet and outlet ends, respectively [48]; and n_i, n_o represent the boundary values of n_0 at inlet and outlet ends, respectively.

Note that the transport regulation is achieved by minimizing the objective function in Eq. (1a). The first term models the velocity control mechanism by minimizing the mismatch between the ideal and practical velocity, whereas the second term models the traffic jam detection mechanism by penalizing the concentration gradient $\nabla n_\pm$ within the entire neuron cell. The third term is a regularization that measures the control forces applied by neuron to mediate the transport. In this work, we employ the PDE-CO model to simulate the normal transport and two types of MT-induced traffic jams by altering the spatial distribution of $l_\pm$. We use a single pipe geometry (Fig. 1A) to describe the simulation settings for each transport condition as below.

- *Normal transport.* In this study, we assume the material yields a unidirectional anterograde transport and thus the n_-, l_-, $\boldsymbol{v}_-$, k_-, k'_- terms are ignored in Eqs. (1b)–(1f). The ideal velocity field V_+ is defined by the solution of the steady-state Navier–Stokes equation in the neuron geometry. In particular, a parabolic velocity profile (Fig. 1A) is imposed as the inlet boundary condition and no-slip boundary condition is imposed on the wall. The maximum magnitude of the inlet velocity is set to be 1 μm/s [48]. The variational multiscale method [49] is employed to solve the Navier–Stokes equation and more implementation details about the solver in neuron can be found in [37]. The other parameters to define the normal transport is selected based on our previous studies [38,39] and summarized in Table 1.
- *Traffic jam caused by MT reduction.* The reduction of MTs from axons and dendrites contributes to the abnormal material aggregation in neurodegenerative disease [50,51]. Based on the setting for normal transport, we introduce the reduction of MTs in a local region of a single pipe geometry by setting l_+ as

$$l_+(x) = 1 - 0.9 * exp[-400(x - 0.5L)^4], \tag{2}$$

  where L is the total length of the single pipe. As shown in Fig. 1B, a sudden decrease of l_+ can be observed in the middle of the pipe. The definition of l_+ can be generalized along the skeleton of neuron to handle more complex 3D geometries.
- *Traffic jam caused by MT swirls.* The polar reorientation of MTs caused by overexpression of the tau protein can lead to MT swirls which further induce the traffic jam of organelle transport and swollen neuron

Table 1
The default parameters utilized in simulation.

Fixed parameter	Default value [Unit]	Parameter	Default value [Unit]
$D_{\pm}$	1.0 [$\mu m^3/s$]	$l_{\pm}$	1.0
$k_{\pm}$	1.0 [s^{-1}]	λ_i	2.0
$k'_{\pm}$	0.1 [s^{-1}]	λ_o	2.0
μ	0.1 [μm^{-1} s^{-1}]	α	1.0
n_i	1.0 [mol/μm^3]	β	1.0
n_o	0.0 [mol/μm^3]		

geometry [52]. To simulate this phenomena, we first introduce the swelling geometry in the middle of pipe geometry, as shown in Fig. 1C. Let the normal diameter of the pipe be D_n, the length L_s and the maximum diameter D_s of the yellow swelling region are set as $L_s = 3D_n$ and $D_s = 2D_n$, respectively. In addition, the MT distribution of $l_{\pm} = l_{\pm}(x, y, z)$ should be non-uniform on the cross-section corresponding to the direction and location of MT swirls. In particular, we define three regions in each cross-section (Fig. 1D): (1) red region that has more "+" direction MTs and larger l_+ value; (2) blue region that contains "–" direction MTs due to MT swirl; and (3) gray region with normal MT distribution. In Fig. 1D, the green MT goes through the red region in "+" direction twice while go through the blue region in "–" direction once and we set $l_+ = 2.0$ in the red region and $l_- = 1.0$ in the blue region. The definition of the $l_{\pm}$ on the cross-section of the swelling region is

$$\begin{cases} l_+ = 2.0, \quad l_- = 0.2 & \text{in red region,} \\ l_+ = 0.2, \quad l_- = 1.0 & \text{in blue region,} \\ l_+ = 1.0, \quad l_- = 0.0 & \text{otherwise.} \end{cases} \tag{3}$$

In this study, we assume the swirly MTs always follow the path to go through the lower swelling geometry twice, as shown in Fig. 1D. The distribution of l_+ and l_- along the blue and red regions from the inlet to outlet of the single pipe are shown in Fig. 1E. More detailed discussion on the definition of $l_{\pm}$ for other complex MT swirls can be found in [38,39].

Herein, we adopt cubic B-splines as the basis functions for geometric modeling and IGA in neurons. Given a neuron geometry with its skeleton information stored in a SWC file, we apply the skeleton-based sweeping method [27] to generate quadrilateral and hexahedral control meshes, on which surface [53] and volumetric B-splines [26] are constructed to represent the 2D and 3D neuron geometries, respectively. Regarding the implementation of IGA solver, the "all-at-once" method [54,55] is employed to formulate and solve the optimization model (Eq. (1a)) with PDE constraints (Eqs. (1b)–(1f)) simultaneously. We adopt the "discretize-then-optimize" scheme to obtain the discrete Lagrangian of the optimization problem as

$$\mathcal{K}_h = \mathcal{J}_h + \lambda^T \mathcal{C}_h = 0, \tag{4}$$

where $\mathcal{J}_h$ is the discretized objective function; $\mathcal{C}_h$ is the discretized PDE constraints; λ^T is the Lagrange multiplier and is also referred to as the "adjoint variable". By taking derivatives of the discrete Lagrangian with respect to state, control, and adjoint variables and setting the resulting expressions to zero, we obtain the first-order conditions, or Karush–Kuhn–Tucker (KKT) conditions. The resulting KKT system is then solved using IGA and more implementation details can be found in [38,39]. To support the following deep learning study, we use the IGA solver to create training data.

3. IGA-based PGNN model

In this section, we demonstrate the IGA-based PGNN model for predicting complex abnormal transport phenomena within different neuron morphologies. We first briefly review the definition of cubic B-splines and its Bézier extraction that contains the geometry information from IGA. We then introduce the PGNN simulators for the local prediction of normal and abnormal transport within pipes and bifurcations, and the GNN assembly model for the global prediction by combining the local predictions into the entire neuron tree geometry. We discuss in detail the loss function for each model and explain how the Bézier extraction information from IGA is incorporated into the computation of physics-based loss function for training.

3.1. A brief review of B-spline and Bézier extraction

Bézier extraction has been widely employed to implement the spline basis functions used in IGA [56,57]. The key idea of Bézier extraction is to represent the spline basis functions as element-wise Bernstein polynomials, which can be easily incorporated into an existing FEM framework. In this study, we adopt cubic B-splines as the basis functions for IGA and here we give a brief review of Bézier extraction for cubic B-splines. A univariate B-spline basis function is defined on a knot vector. The knot vector is a set of non-decreasing real numbers $\Xi = \{u_1, u_2, \ldots, u_{m+q+1}\}$, where q is the degree and m is the number of B-spline basis functions. The q-degree B-spline basis functions $\boldsymbol{N}_q(u) = \{N_{i,q}\}_{i=1}^{m}$ are defined recursively starting with piecewise constants

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \le u \le u_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

after which the remaining higher-order B-spline basis functions ($q > 0$) are constructed recursively using the Cox-de Boor recursion formula [58]

$$N_{i,q}^{k}(u) = \frac{u - u_i}{u_{i+q} - u_i} N_{i,q-1}(u) + \frac{u_{i+q+1} - u}{u_{i+q+1} - u_{i+1}} N_{i+1,q-1}(u). \tag{6}$$

Combined with a set of control points $\boldsymbol{P} = \{\boldsymbol{P}_i\}_{i=1}^{m}$, a B-spline curve of degree q can be defined as

$$S(u) = \sum_{i=1}^{m} \boldsymbol{P}_i N_{i,q} = \boldsymbol{P}^T \boldsymbol{N}_q(u). \tag{7}$$

B-spline basis functions support knot insertion without changing the geometric or parametric properties of the B-spline curve. Assuming a new knot $\overline{u} \in [u_k, u_{k+1})$ is inserted into the existing knot vector Ξ and we have the new knot vector $\overline{\Xi} = \{u_1, u_2, \ldots, u_k, \overline{u}, u_{k+1}, \ldots, u_{m+q+1}\}$. The $\overline{m} = m + 1$ new control points can be obtained by

$$\overline{\boldsymbol{P}}_i = \begin{cases} \boldsymbol{P}_1, & i = 1, \\ \gamma_i \boldsymbol{P}_i + (1 - \gamma_i)\boldsymbol{P}_{i-1}, & 1 < i < \overline{m}, \\ \boldsymbol{P}_m, & i = \overline{m}, \end{cases} \tag{8}$$

where

$$\gamma_i = \begin{cases} 1, & 1 \le i \le k - q, \\ \dfrac{\overline{u} - u_i}{u_{i+q} - u_i}, & k - q + 1 \le i \le k, \\ 0, & i \ge k + 1. \end{cases} \tag{9}$$

Letting $\overline{\boldsymbol{P}}^1 = \boldsymbol{P}$, Eq. (8) can be revised in a matrix form as

$$\overline{\boldsymbol{P}}^{j+1} = (\boldsymbol{C}^j)^T \overline{\boldsymbol{P}}^j, \tag{10}$$

where $\boldsymbol{C}^j$ is the Bézier extraction operator for the jth inserted knot. Note that the continuity of the basis is reduced by one with every repetition of an existing knot value, while the continuity of the B-spline curve is preserved with Eqs. (8)&(9). To obtain the Bézier element of cubic B-splines $\boldsymbol{N}_3(u)$, Bézier decomposition is adopted by repeating all interior knots of Ξ until they have a multiplicity equal to 4, and the whole Bézier extraction operator becomes

$$\boldsymbol{C}^T = (\boldsymbol{C}^r)^T (\boldsymbol{C}^{r-1})^T \ldots (\boldsymbol{C}^1)^T, \tag{11}$$

and we get

$$\boldsymbol{P}^b = \boldsymbol{C}^T \boldsymbol{P}, \tag{12}$$

where $\boldsymbol{P}^b$ represents the control points after Bézier extraction. Since the knot insertion does not change the geometry, we obtain the following relationship between the cubic B-spline basis functions and Bernstein polynomial

$$\boldsymbol{N}_3(u) = \boldsymbol{C}\boldsymbol{B}_3(u), \tag{13}$$

where $\boldsymbol{B}_3(u)$ is the cubic Bernstein basis functions after Bézier extraction.

Note that the control mesh of neuron is unstructured with extraordinary points (EPs) on it, which should be carefully handled during spline construction. Herein, the EPs on the quadrilateral control mesh are handled using the D-patch framework [59] to introduce degenerated surface patches with their partial derivatives vanish around EPs. In each face around EPs, four face-based spline control points are assigned and computed with the vertex-based spline control points of this face. The Bézier control points are defined with vertex and face-based control points to obtain the Bézier extraction operator. The Bézier extraction operator is then modified using the split-then-smoothen [60] approach to achieve C^1 continuity around EPs. On the hexahedral control mesh, blending functions are defined with Bernstein-Bézier splines to naturally handle EPs [53]. For each interior element of a hexahedral control mesh, its 8 control points are first used to define the body Bézier control points. Then, the face, edge and corner Bézier control points are obtained by averaging the nearest body points. The Bézier control points in the boundary elements are defined using a similar approach except the boundary Bézier points are defined using the boundary quadrilateral mesh. In this way, we define the blending function and the Bézier extraction operator in the entire mesh. The detailed implementation on handling EPs in the surface and volumetric spline construction can be found in [25,53], respectively. All these methods directly support Bézier extraction of the spline representation, which are convenient to use and motivate us to utilize the Bézier information for the loss function computation of PDE residuals in the PGNN model.

3.2. PGNN model overview

The aim of our IGA-based PGNN model is to learn from the material transport simulation data and predict the abnormal transport phenomena within different neuron morphologies. To address the geometry diversity of neurons, we use the graph representation of the neuron tree and build the IGA-based PGNN model using the graph network (GN) framework [61]. A neuron tree geometry can be decomposed into two basic structures: pipe and bifurcation (yellow square/triangle/pentagon and red circle in Fig. 2, respectively). Each structure can be treated as one node in the graph and the nodes can be connected following the skeleton of the neuron tree to constitute the graph. Here, the decomposition is performed automatically to ensure the decomposed structures have similar length and the length is within a range of (0, 10]. The decomposition also uses an input parameter to control the average length of the decomposed structure so that a suitable decomposition for different geometries can be obtained. Based on the graph representation of the neuron tree, two separate PGNN simulators are built for the pipe (G_S^p) and the bifurcation (G_S^b), respectively. Given input features x_{global}^k at the kth time step that includes node locations, simulation parameters and initial nodal concentration, the simulators can output the local nodal concentration result at the $(k+1)$th time step x_{local}^{k+1} in the pipe and the bifurcation, respectively. To obtain a consistent global concentration result, a GNN assembly model (G_A) is adopted to learn the interaction between different structures so that given intermediate value x_{local}, the model can assemble different structures and output the final prediction x_{global}^k on the graph. In summary, our IGA-based PGNN model consists of two major components:

- The IGA-based PGNN simulators for local prediction in four different cases: (1) normal transport in pipe, (2) traffic jam in pipe caused by reduced MTs, (3) traffic jam in pipe caused by MT swirls, and (4) normal transport in bifurcation; and
- The GNN assembly model to handle three types of assembly: (1) pipe-pipe assembly, (2) pipe-bifurcation assembly, and (3) bifurcation-bifurcation assembly, for global prediction in the entire neurite network.

3.3. IGA-based PGNN simulators

The simulators are designed using a recurrent "GN blocks + MLP Decoder" architecture as shown in Fig. 2. The input of simulators contains the control mesh of a pipe or bifurcation, as well as a feature vector $h^0 = (\mathbf{x}, \mathbf{p}, \mathbf{z}^{t_k})$, where $\mathbf{x}$ contains the coordinates of each node on the control mesh; $\mathbf{p}$ denotes the parameter setting ($D_\pm$, $k_\pm$, $l_\pm$, and etc.) in Eq. (1); $\mathbf{z}^{t_k} = (n_0, n_\pm, v_\pm)^{t_k}$ represents the predicted concentration and velocity values at t^k. In this study, since we only modify $l_\pm$ to define different transport conditions, $\mathbf{p}$ has a dimension of 2 and the dimension of h^0 is 11 for 2D geometry and 14 for 3D geometry, respectively. The input feature vectors are encoded by a series of GN blocks to learn the hidden layer representations that encode both local graph structure and features of nodes. Given an input graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, the GN block takes its connectivity information $(\mathbf{V}, \mathbf{E})$ and the nodal

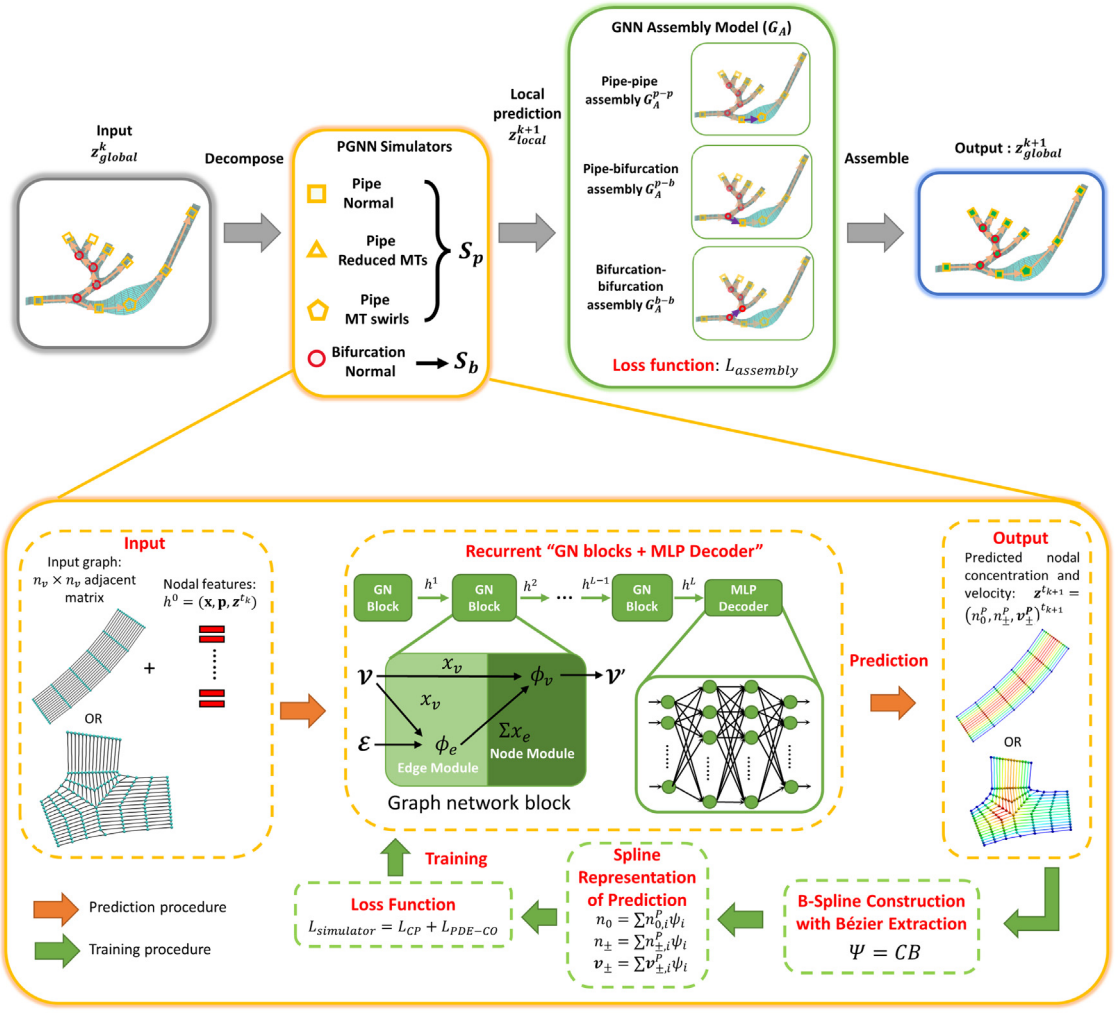

Fig. 2. An overview of the IGA-PGNN model. The input neuron geometry is first decomposed into pipes and bifurcations to create the graph representation of the neuron tree. Next, the input features z^k_{global} of each pipe or bifurcation are processed by the corresponding PGNN simulators (S_p or S_b) to generate intermediate concentration result z^{k+1}_{local}. The PGNN simulators adopt a recurrent "GN blocks + MLP Decoder" architecture and utilize the B-spline information in Bézier extraction format for computing physics-based loss function in training. Four types of simulators are trained to handle different geometries and transport conditions. Then, the GNN assembly model (G_A) takes z^{k+1}_{local} as input and computes the interaction between adjacent simulators to predict global concentration result z^{k+1}_{global}.

features $\boldsymbol{x}_v$ from $\mathbf{V}$ as input, and outputs a graph with the updated nodal features $\mathbf{V}'$. $\mathbf{V}'$ serves as the hidden layer representation and it has the same size as the input nodal feature vector. In our PGNN simulators, the GN block includes two modules ϕ_e, ϕ_v to update the edge and node features, respectively. The edge module ϕ_e first computes edge attributes $\boldsymbol{x}_e$ on each edge $e_{ij} \in \mathbf{E}$ using the nodal feature $\boldsymbol{x}_{v_i}$ and $\boldsymbol{x}_{v_j}$ of its two end nodes. For each node $v_i \in \mathbf{V}$, the edge attributes of all the edges connected to v_i are aggregated as $\boldsymbol{x}_{e,v_i} = \sum_j \boldsymbol{x}_{e_{ij}}$. Then, the node module ϕ_v obtains the updated nodal feature $\boldsymbol{x}'_{v_i}$ with $\boldsymbol{x}_{\boldsymbol{v}_i}$ and $\boldsymbol{x}_{e,v_i}$. Both ϕ_v and ϕ_e are a multilayer perceptron (MLP) consisting of two hidden layers (with ReLU activations), each layer with the size of 32. The L-step GN blocks follow Algorithm A1 to encode all the nodal features and generate a series of updated latent graph with the hidden layer representations $h_1, h_2, \ldots, h_L$ ($L = 3$ in this study). Then, a MLP is used to decode the hidden layer representation h_L and output predicted velocity and concentration values $\boldsymbol{z}^{t_{k+1}}$ at the next time step t_{k+1}. The MLP has three hidden layers (with ReLU activations), followed by a non-activated output layer, each layer with the size of 32. The PGNN simulators aim to provide accurate local prediction of velocity and material

distribution in pipes and bifurcations Our previous studies [38,39] have revealed that both geometry and transport condition have significant influence on the patterns of velocity and material distribution. To ensure the simulators are competent for different scenarios, we train separate simulators which are specialized for normal and abnormal transport prediction, respectively. The abnormal simulators are only adopted in the region with traffic jam, whereas the normal simulators handle the prediction in the remaining regions. Since most neurodegenerative diseases are associated with the disruption of axonal transport, we assume the two types of traffic jams described in Section 2 only occur in the pipe geometry. Combining with the normal simulators in different geometries, we have different PGNN simulators trained to predict the following four cases: (1) normal transport in pipe; (2) traffic jam in pipe caused by reduced MTs; (3) traffic jam in pipe caused by MT swirls; and (4) normal transport in bifurcation. All the simulators share the same recurrent "GN blocks + MLP Decoder" architecture but are trained with pipe and bifurcation datasets, respectively.

The loss function in the PGNN simulators is specifically designed to encourage the model to learn the physics underlying the simulation data. The detailed form of the PGNN simulator loss function is

$$\mathcal{L}_{simulator} = \mathcal{L}_{CP} + \mathcal{L}_{PDE-CO}. \tag{14}$$

The first term $\mathcal{L}_{CP}$ minimizes the difference between the prediction and IGA simulation results to encourage the neural network to match the observation on the control points. The control point loss function is defined as

$$\begin{aligned}\mathcal{L}_{CP} = &\frac{1}{N_{CP}}\sum_{i=1}^{N_{CP}}\left[(n_{0,i}^{P}-n_{0,i}^{G})^2+(n_{\pm,i}^{P}-n_{\pm,i}^{G})^2+\|\boldsymbol{v}_{\pm,i}^{P}-\boldsymbol{v}_{\pm,i}^{G}\|^2\right]\\ &+\frac{1}{N_{CP}}\sum_{i=1}^{N_{CP}}\left\|\frac{\boldsymbol{v}_{\pm,i}^{P}}{\eta+\|\boldsymbol{v}_{\pm,i}^{P}\|}-\frac{\boldsymbol{v}_{\pm,i}^{G}}{\eta+\|\boldsymbol{v}_{\pm,i}^{G}\|}\right\|^2,\end{aligned} \tag{15}$$

where N_{CP} is the number of control points, $\eta = 10^{-4}$ is a small constant to avoid singularities, and superscripts P and G denote the prediction and the ground truth value on the control mesh, respectively. The first term in $\mathcal{L}_{CP}$ encourages the neural network to match the magnitude of concentration and velocity. Motivated by the complex velocity streamline pattern observed in [39], we introduce the second term in $\mathcal{L}_{CP}$ to instruct the neural network to fit the direction of the velocity.

The second term $\mathcal{L}_{PDE-CO}$ is built based on the PDE-CO model such that the underlying transport model is utilized to inform the PGNN training. In particular, we use Gaussian quadrature to evaluate the weak form of the PDE residuals and the objective function $\mathcal{J}$ (Eq. (1a)). Let $\psi = \{\psi_i\}_{i=1}^{N_{CP}}$ be the spline basis represented using Bézier extraction and we have the predicted PDE-CO solution

$$n_0 = \sum_{i=1}^{N_{CP}} n_{0,i}^{P}\psi_i, \quad n_\pm = \sum_{i=1}^{N_{CP}} n_{\pm,i}^{P}\psi_i, \quad \boldsymbol{v}_\pm = \sum_{i=1}^{N_{CP}} \boldsymbol{v}_{\pm,i}^{P}\psi_i, \tag{16}$$

where the superscript P denotes the nodal prediction on the control points. Let ω be the weighting function, the weak formulation for Eqs. (1b)–(1d) can be written as

$$\begin{cases} r_{n_0} = \int_0^T\int_\Omega \omega(\dfrac{\partial n_0}{\partial t} + (k_+ + k_-)n_0 - k'_+ l_+ n_+ - k'_- l_- n_-)\, d\Omega dt, \\ r_{n_\pm} = \int_0^T\int_\Omega \omega\dfrac{\partial (l_\pm n_\pm)}{\partial t} + \omega \boldsymbol{v}_\pm \cdot \nabla(l_\pm n_\pm) + D_\pm \nabla\omega\cdot\nabla(l_\pm n_\pm) - \omega k_\pm n_0 + \omega k'_\pm (l_\pm n_\pm)\, d\Omega dt, \\ r_{\boldsymbol{v}_\pm} = \int_0^T\int_\Omega \omega\dfrac{\partial \boldsymbol{v}_\pm}{\partial t} + \omega\boldsymbol{v}_\pm\cdot\nabla\boldsymbol{v}_\pm + \omega\nabla n_\pm + \nabla\omega\cdot(\mu\nabla\boldsymbol{v}_\pm) - \omega f_\pm\, d\Omega dt. \end{cases} \tag{17}$$

By introducing $\{\xi_k, \alpha_k\}_{k=1}^{n_q}$ as the Gaussian quadrature points and weights for integral over the domain Ω, we impose the physical constraints by encouraging the objective function $\mathcal{J}$ (Eq. (1a)) and the weak form of PDE residuals (Eq. (17)) to be zero and the loss function takes the following form:

$$\mathcal{L}_{PDE-CO} = \sum^{k}\alpha_k\mathcal{J}^k + (\sum^{k}\alpha_k r_{n_0}^k)^2 + (\sum^{k}\alpha_k r_{n_\pm}^k)^2 + (\sum^{k}\alpha_k r_{\boldsymbol{v}_\pm}^k)^2, \tag{18}$$

where $\mathcal{J}^k$, $r_{n_0}^k$, $r_{n_\pm}^k$ and $r_{\boldsymbol{v}_\pm}^k$ are the objective function, the PDE residuals evaluated at the kth quadrature point ξ_k. In Eq. (14), we equally weight both loss terms since we believe both terms are equally important for training the simulator model to obtain accurate predictions.

In our earlier study [47], we implemented two GNN-based simulators that predict the material concentration during normal transport in pipe and bifurcation geometries. The PDE residuals of a motor-assisted transport model [37] was evaluated in the loss function for a physics-informed training. Specifically, the partial derivatives were directly computed on the discrete sample points using the finite difference method, which is less accurate compared to the evaluation with continuous spline basis functions in the current PGNN simulators. Due to the overhaul on the underlying transport model, the PGNN simulators also show a superior improvement by supporting traffic jam prediction in the pipe geometry.

3.4. GNN assembly model

The purpose of the GNN assembly model is to take the local prediction from simulators as input and provide the global concentration distribution on the entire geometry. As shown in Fig. 2, each neuron geometry can be represented as a graph with pipe or bifurcation structure as node and the connectivity relation as edge. There are three types of edges based on different simulator combinations on its two ends and thus the assembly model includes three components: (a) pipe and pipe G_A^{p-p}; (b) pipe and bifurcation G_A^{p-b}; and (c) bifurcation and bifurcation G_A^{b-b}. Note that the traffic jam only occurs in a local region and the assembly between normal transport simulators is more general on the entire geometry. Therefore, the assembly between traffic jam and normal transport simulators is not considered in this study. Once the local prediction on each node is obtained, the GNN assembly model loops each edge on the graph and utilizes the "message-passing" scheme to gather predicted results from its neighboring simulator nodes. Given the local predictions $\mathbf{z}_{local,s_1}$, $\mathbf{z}_{local,s_2}$ from two neighboring simulators s_1 and s_2, the GNN assembly model G_A generates the global prediction as $[\mathbf{z}_{global,s_1}, \mathbf{z}_{global,s_2}] = G_A(\mathbf{z}_{local,s_1}, \mathbf{z}_{local,s_2})$. All three components share the same MLP architecture with three hidden layers (with ReLU activations), followed by a non-activated output layer, each layer with the size of 32.

The loss function in the GNN assembly model is designed to ensure the prediction result is consistent on the interface between two adjacent simulators. Here, we impose the same boundary values on the interface between any two assembled simulators and the loss function is defined as

$$\begin{aligned}\mathcal{L}_{assembly} = &\frac{1}{N}\sum_{i=1}^{N}\left\{(n_{0,i}^{P} - n_{0,i}^{G})^2 + (n_{\pm,i}^{P} - n_{\pm,i}^{G})^2\right\} + \frac{1}{M}\sum_{j=1}^{M}\Big\{(n_{0,j}^{s_1,interface} - n_{0,j}^{s_2,interface})^2 \\ &+ (n_{\pm,j}^{s_1,interface} - n_{\pm,j}^{s_2,interface})^2\Big\} \\ &+ \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{v}_{\pm,i}^{P} - \boldsymbol{v}_{\pm,i}^{G}\|^2 + \frac{1}{M}\sum_{j=1}^{M}\|\boldsymbol{v}_{\pm,j}^{s_1,interface} - \boldsymbol{v}_{\pm,j}^{s_2,interface}\|^2,\end{aligned} \tag{19}$$

where N denotes the number of nodes on two assembled simulators, M denotes the number of nodes on the interface, superscript P and G denote the prediction and the ground truth values on the graph, respectively. Superscripts s_1 and s_2 denote the prediction values from the first and second simulators, respectively.

3.5. Model evaluation

To evaluate the accuracy of the predicted concentration and velocity distributions, we adopt the mean relative error (MRE) defined as

$$MRE(\boldsymbol{y}^P, \boldsymbol{y}^G) = \frac{\sqrt{\sum_{i=1}^{N}\frac{1}{N}(\boldsymbol{y}_i^P - \boldsymbol{y}_i^G)^2}}{\max\|\boldsymbol{y}^G\| - \min\|\boldsymbol{y}^G\|} \times 100\%, \tag{20}$$

where N denotes the number of nodes in the output, $\boldsymbol{y}_i^P$ and $\boldsymbol{y}_i^G$ denote the predicted and ground truth values of the ith node in a given mesh, respectively, $\max\|c^G\|$ and $\min\|c^G\|$ denote the maximum and minimum nodal values from the ground truth result, respectively.

Table 2
The MRE results of simulator and assembly model.

Dimension	Simulator				Assembly model
	Pipe (Normal)	Pipe (Reduced MTs)	Pipe (MT swirls)	Bifurcation (Normal)	
2D	5.72 ± 1.65%	7.80 ± 1.96%	8.62 ± 2.27%	6.60 ± 2.41%	7.35 ± 1.32%
3D	6.10 ± 2.75%	8.54 ± 2.48%	8.98 ± 3.26%	7.04 ± 1.87%	8.92 ± 0.85%

4. Numerical testings and discussion

In this section, we present the data generation and training results of the IGA-based PGNN models. We evaluate the performance of our proposed methods on the prediction of two different traffic jams induced by (1) reduced MTs; and (2) MT swirls. The velocity and material distribution predictions in different 2D and 3D neuron geometries are compared with their corresponding IGA results to demonstrate the effectiveness of our model. The IGA solver of the PDE-CO transport model is implemented using PETSc [62], and the IGA-based PGNN model is implemented using PyTorch [63] and the "PyTorch Geometric" library [64]. All computations are conducted on the XSEDE (Extreme Science and Engineering Discovery Environment) supercomputer Bridges at the Pittsburgh Supercomputer Center [65,66].

4.1. Data generation and training for the IGA-based PGNN model

To create the training dataset for different simulators, we reuse the two complex zebrafish neurons (NMO_66731 and NMO_66748 in the NeuroMorpho.Org database [67]) to extract pipe and bifurcation geometries as in [47]. As shown in Fig. 3, we directly extract the pipe and bifurcation structures from the control mesh of these two trees and obtain 100 geometries for each structure. To generate the training dataset for the pipe and bifurcation simulators in normal transport condition, we run the PDE-CO simulation in NMO_66731 and NMO_66748 with 100 different input boundary values n_i ranging from 0.1 to 5.0. All the other parameters adopt the default values in Table 1. The time step for each simulation is set to be 0.1 s. We simulate until the transport process is steady and then extract the simulation results of 100 different geometries for each simulator from these two neurons. Regarding the training dataset for the traffic jam simulators in pipe, we run the PDE-CO simulation within the extracted pipe geometries. In particular, the swelling geometry used in the MT swirls simulation is introduced in the middle of each pipe geometry using the approach described in Fig. 1C. For each pipe structure, we employ the $l_{\pm}$ distribution described in Section 2 to model the two MT-induced traffic jams with 100 different input boundary values n_i. Overall, we obtain 10,000 training samples for each of the four simulators. To generate training dataset for the GNN assembly model, we reuse the previous normal transport simulation results of the 100 different boundary conditions in two zebrafish neurons. Based on the graph representation of these two trees, we follow three basic assembly structures and extract 40 different geometries for each type of assembly, and thus we obtain 4000 training samples for each of the three assembly models.

For both simulator and assembly models, we perform 4-fold cross validation with 75% samples as the training dataset and we evaluate the performance of the model using the rest 25% samples as the test dataset. We use Eq. (20) to evaluate the MRE for each trained model and the results are summarized in Table 2. We find that the pipe simulators perform the best in the normal transport and becomes worse when traffic jam is introduced, which is reasonable since the velocity and concentration patterns during traffic jam (such as the vortex shape velocity streamline observed in [39]) is harder to be captured by the simulators. For normal transport prediction, the simulators have slightly better performance in pipe than bifurcation, which is similar to the results reported in [47]. In addition, we compare our results with [68] that utilizes physics-informed graph neural Galerkin networks (PI-GGN) for solving PDE problems in different geometries. The PI-GNN can achieve a relative error of 5% for solving linear elasticity problem in a 3D hollow cylinder, which is comparable to the performance of our PGNN simulators. In this study, we select 10% as the error tolerance because we need to deal with a variety of 3D geometries and more complex PDE-CO problems. Including more training data could help achieve higher accuracy.

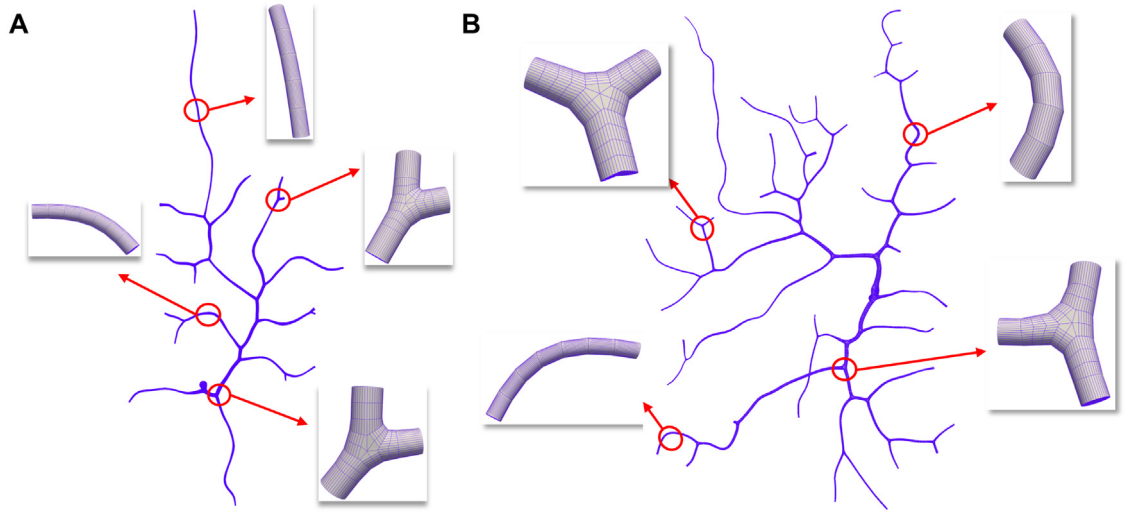

Fig. 3. The extraction of pipe and bifurcation structures from the control mesh of (A) NMO_66731 and (B) NMO_66748. The zoom-in pictures show the extracted geometries of pipe and bifurcation samples used for data generation.

4.2. Results for complex 2D neuron geometries

After the simulators and the assembly model in the PGNN framework are well trained, we use the model to predict concentration and velocity distributions during traffic jam within two 2D complex neuron geometries. The results are compared with the corresponding IGA results, as shown in Fig. 4. The predefined velocity field V_+ is computed by solving a steady-state Navier–Stokes equation in two neuron geometries and the other simulation parameter settings are set using default values (Table 1). To introduce the traffic jam caused by the reduction of MTs, we reduce the MT distribution of l_+ within the red dashed circle in Fig. 4A&E. The velocity field and material distribution results from the PGNN model and the nodal errors between PGNN and IGA results are shown in Fig. 4A&E and B&F, respectively. We find that the sudden decrease of velocity and increase of concentration in the traffic jam region are accurately captured by the PGNN model, though the prediction error around the traffic jam region is higher than the other regions. We evaluate the MRE for both velocity and concentration prediction in two geometries and all the MREs are below 10%. For each geometry, we also plot the material distribution from the inlet to each outlet and compare the distribution of IGA and PGNN results in Fig. 5A&C. The overall prediction results match well with IGA results with similar location of the traffic jam region and minor difference on the maximum concentration value. By observing the downstream branches that are affected by the traffic jam region (Curves 1–4 in Fig. 5A and Curves 3–8 in Fig. 5C), we find that less materials are transported to these outlets than the other outlets and more material accumulates before the traffic jam region.

We further study the performance of the PGNN model in predicting traffic jam caused by MT swirls. We introduce the swelling region in the red dashed region in Fig. 4C&G and modify the MT distribution for the counter-clockwise MT swirl as described in Section 2. The simulation parameter settings stay the same as in Table 1. The velocity field and material distribution results from the PGNN model and the nodal errors between PGNN and IGA results are shown in Fig. 4C&G and D&H, respectively. We observe in Fig. 4C&G that the accumulation of concentration is well captured with a relatively higher error compared to the other regions, and the MREs of velocity and concentration are less than 9%. In Fig. 4D&H, we find that the prediction error is larger at the high concentration region between the bifurcation and swelling geometries. The error may come from the GNN assembly model since the assembly between the traffic jam and normal transport simulators is infrequent and thus not considered in this study. We believe it is reasonable to ignore such assembly since the current GNN assembly model can still provide accurate global predictions with the MREs of every example in Figs. 6&7 below 10%. We also plot the material distribution from the inlet to each outlet and for both IGA and PGNN results in Fig. 5B&D. The predicted traffic jam location is close to IGA results while the PGNN model provides a smaller maximum concentration value than IGA. By comparing the curves in Fig. 5B, we find that the traffic jam not only causes the accumulation in the branch

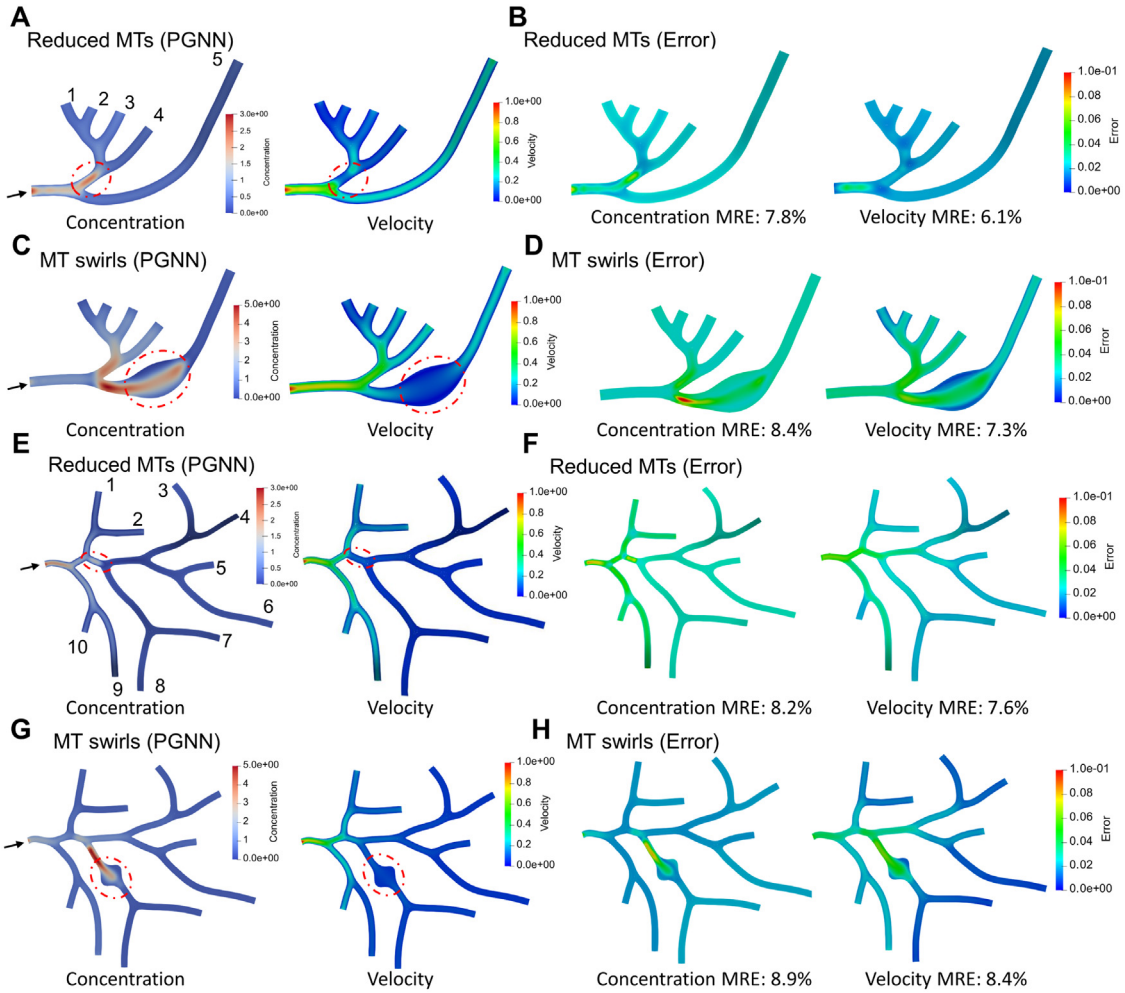

Fig. 4. The comparison between IGA simulation and PGNN prediction of the traffic jams in two 2D neuron trees extracted from (A–D) NMO_54504; and (E–H) NMO_54499. (A) The predicted concentration distribution (left) and velocity (right) of the traffic jam caused by reduced MTs. Black arrow points to the material inlet and the red dashed circle labels the traffic jam region. (B) The nodal errors of concentration and velocity between IGA and PGNN results. (C, D) Same layout as in (A, B) except the traffic jam is caused by MT swirls. (E–H) Same layout as in (A–D). The numbers in (A, E) label each outlet of the two neuron trees. Unit for color bars: Velocity: μm/s and Concentration: mol/μm^3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with MT defects (Curves 1–4) but also leads to the increase of concentration in its neighboring branches (Curve 5). A similar observation can be obtained by comparing Curves 7&8 with Curves 3–6 in Fig. 5D, which indicates that the tree structure of neuron helps to mitigate the material aggregation by digesting the excessive materials with other affected branches.

4.3. Results for complex 3D neuron geometries

As a follow up, we demonstrate the robustness of our method in handling complex neuron geometries in 3D. The well-trained PGNN model is adopted to predict two types of traffic jams within two 3D neuron trees extracted from NMO_54499 and the prediction results are compared with the corresponding IGA simulations, as shown in Figs. 6&7. Similar to Section 4.2, for each geometry, we introduce traffic jams in a local region (red dashed circle in Figs. 6&7(A, B, G, H)). The MT distribution $l_{\pm}$ are altered accordingly to model the reduced number of MTs

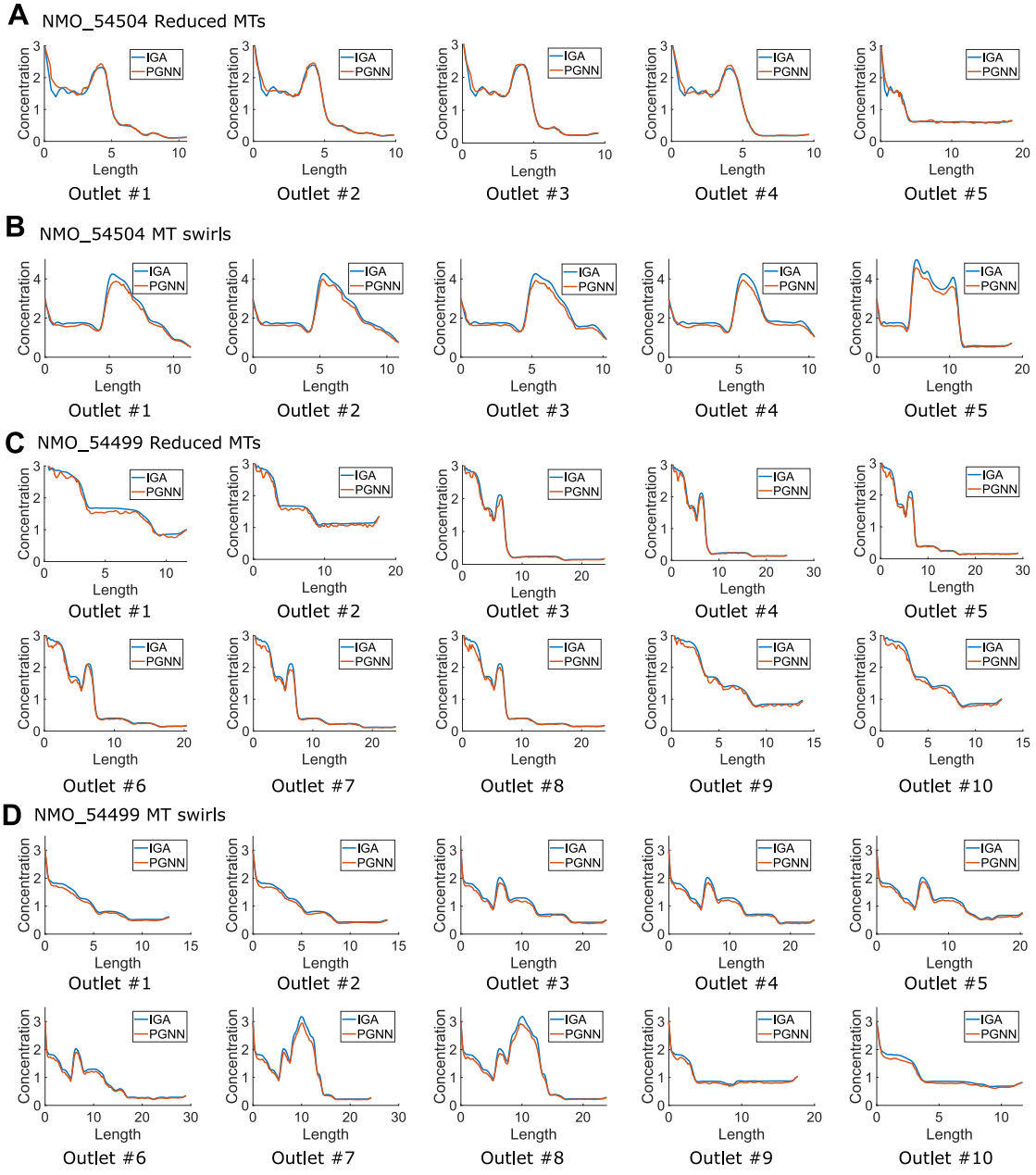

Fig. 5. The comparison of concentration curve on the centerline from the inlet to every outlet between IGA and PGNN results in two 2D neuron trees extracted from (A, B) NMO_54504; and (C, D) NMO_54499. The curves are obtained from results in (A) Fig. 4A&B; (B) Fig. 4C&D; (C) Fig. 4E&F; and (D) Fig. 4G&H.

and MT swirls, as discussed in Section 2. For traffic jam induced by MT swirls, we still increase the skeleton radius in the red dashed circle regions and assume the MT swirls have the reverse path on the top part of these swollen regions as in Fig. 1D. The remaining simulation parameters stay the same as in Table 1.

Fig. 6 shows the traffic jam prediction caused by reduced number of MTs (l_+) for transport in the traffic jam regions. Similar to the IGA results, our PGNN model captures the sudden decrease of velocity (Fig. 6B&H) and a higher material concentration (Figs. 6A&7G) can be observed in these regions. We also compare the velocity streamline in the local traffic jam regions (Fig. 6(C, F)&(I, L)). We observe that the predicted velocity magnitude

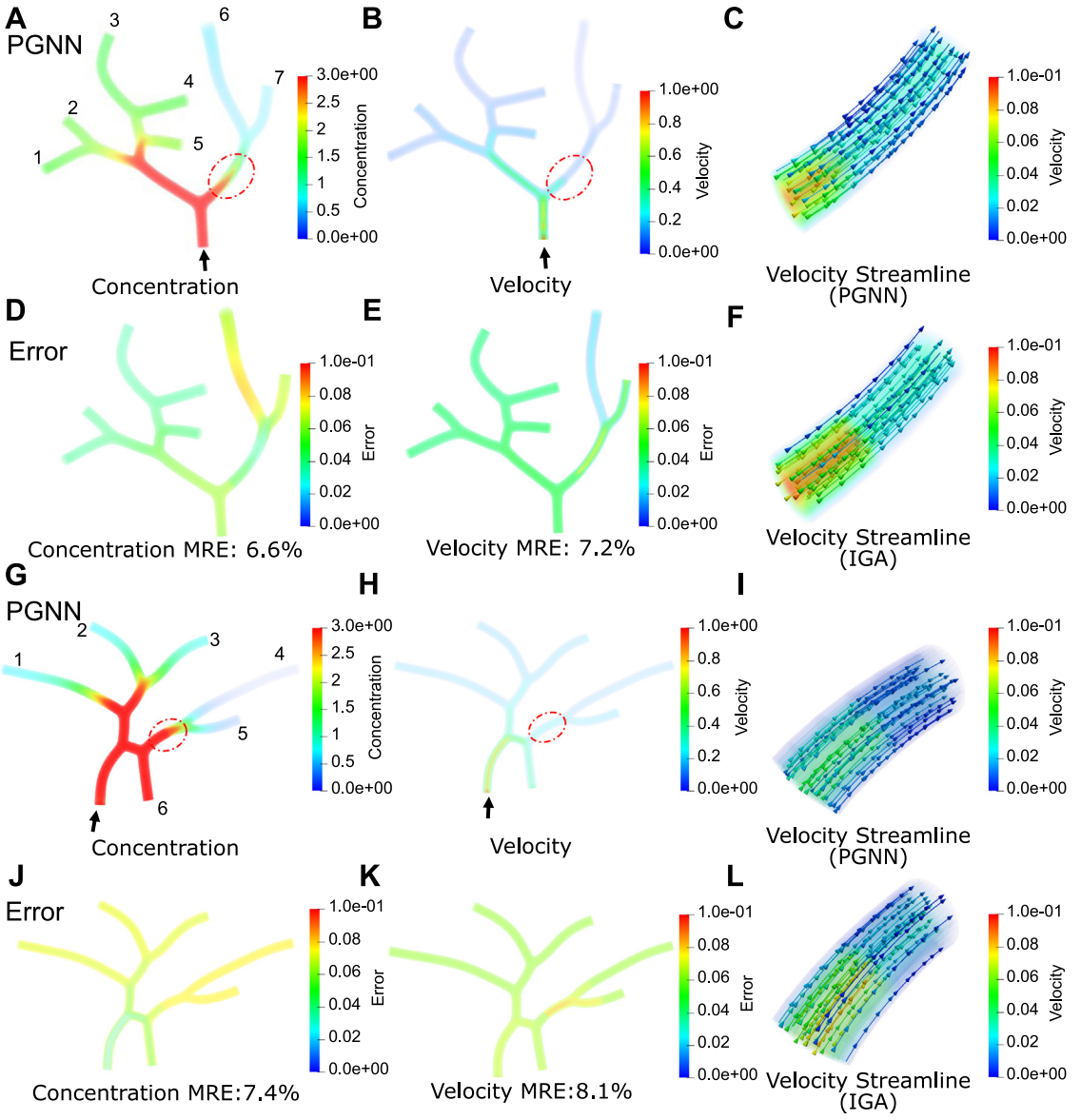

Fig. 6. The comparison between IGA simulation and PGNN prediction of the traffic jam induced by reduced MTs in two complex 3D neuron trees extracted from NMO_54499. (A–F) The comparison between IGA and PGNN results of the first neuron tree. (A, D) The predicted concentration and the nodal concentration errors. (B, E) The predicted velocity and the nodal velocity errors. (C, F) The velocity streamline in the red dashed circle of (A) from PGNN and IGA, respectively. (G–L) The comparison results for the second neuron tree. Same layout as in (A–F). Black arrow points to the inlet of the material transport. The numbers in (A, G) label each outlet of the two neuron trees. Unit for color bars: Velocity: μm/s and Concentration: mol/μm^3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is slightly larger compared to the IGA results while the velocity direction is accurately predicted. In Fig. 6(D, E)&(J, K), we plot the nodal errors of the concentration and velocity distribution between the prediction and ground truth. We find that the PGNN model yields slightly larger error in the velocity prediction around the traffic jam region (Fig. 6E&K), as well as in the concentration prediction downstream the traffic jam region in Fig. 6D. The possible reason is that the error from the traffic jam region is propagated to the downstream branches during the GNN assembly process. For each geometry, we also plot the material distribution from the inlet to each outlet and

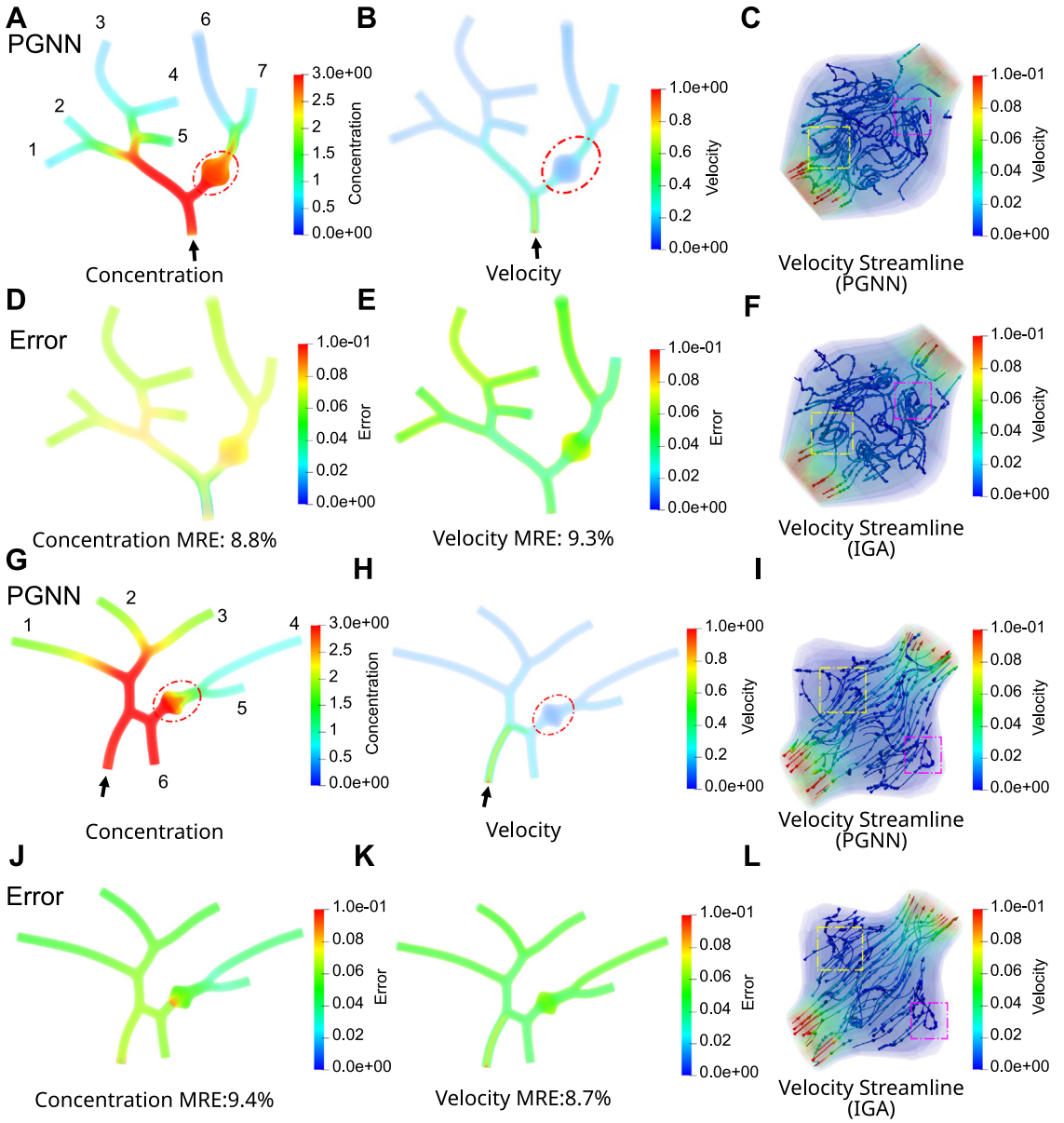

Fig. 7. The comparison between IGA simulation and PGNN prediction of the traffic jam induced by MT swirls in two 3D complex neuron trees extracted from NMO_54499. (A–F) The comparison between IGA and PGNN results of the first neuron tree. (A, D) The predicted concentration and the nodal concentration errors. (B, E) The predicted velocity and nodal velocity errors. (C, F) The velocity streamline in the red dashed circle of (A) from PGNN and IGA, respectively. Yellow and pink bounding boxes label the regions with similar vortex-shape streamlines. (G–L) The comparison results for the second neuron tree. Same layout as in (A–F) except that yellow and pink bounding boxes label the regions with similar reversing streamlines. Black arrow points to the inlet of the material transport. The numbers in (A, G) label each outlet of the two neuron trees. Unit for color bars: Velocity: μm/s and Concentration: mol/μm^3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

compare the distribution of IGA and PGNN results in Fig. 8A&C. By observing the branches that are affected by the traffic jam region (Curves 6&7 in Fig. 8A and Curves 4&5 in Fig. 8C), we find that more materials are trapped to the traffic jam region and thus materials reaching these outlets is significantly less than the other outlets.

As shown in Fig. 7, we then apply the same approach to predict the abnormal transport with MT swirls to obtain velocity field and concentration distribution results and compare with the IGA results in the same geometry.

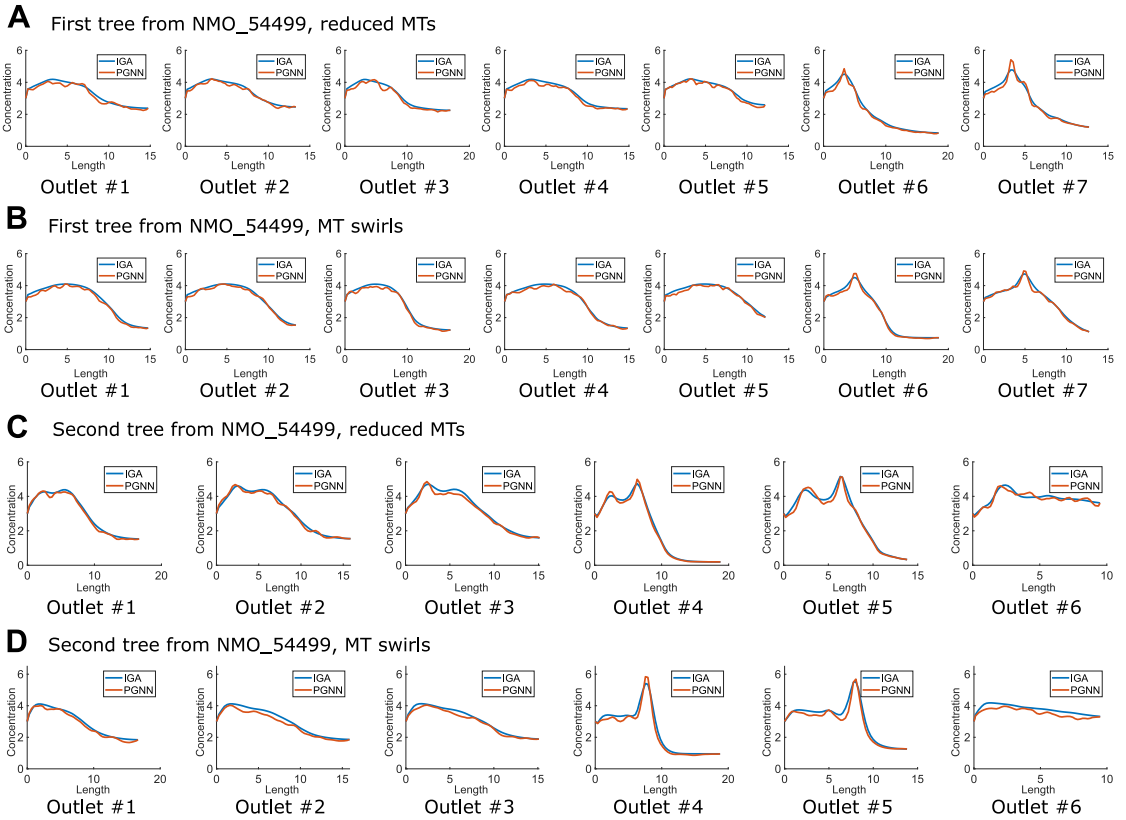

Fig. 8. The comparison of concentration curve on the centerline from the inlet to every outlet between IGA and PGNN results in two 3D neuron trees extracted from NMO_54504. The curves are obtained from results in (A) Fig. 6C&F; (B) Fig. 7C&F; (C) Fig. 6I&L; and (D) Fig. 7I&L.

We observe similar phenomenon that the velocity magnitude decreases and material accumulates in the swollen region. In other branches that are not downstream the swollen region, the material concentration also increases to mitigate the traffic jam in the swollen region. In particular, we observe large prediction error in the swelling region of Fig. 7J. The possible reason is that the geometry diversity within the training data for a pipe with MT swirls may not be sufficient to handle the significant geometry changes of various swelling cases, which leads to inaccurate predictions in the entire neuron trees. In Fig. 7(C, F)&(I, L), we also compare the velocity streamline within the swollen region. Though the streamline is not exactly the same, we find that the PGNN model predicts the vortex-shape (see yellow and pink bounding boxes in Fig. 7C&F) and reversing streamlines (see yellow and pink bounding boxes in Fig. 7I&L) as in the IGA results. Obviously, the non-uniform spatial MT distribution of MT swirls contributes to the complex transport flow. Due to these vortex and reversing streamlines, materials have to experience longer transport path or even be trapped within the swelling region and they form severe traffic jam. The successful prediction of these features proves the effectiveness of the proposed method in accurately predicting traffic jam phenomenon within neurons.

4.4. Computational cost

In Table 3, we summarize the computational cost of our PGNN model and compare with the IGA simulation. The cost of our framework is mainly dominated by the one-time offline training time of the PGNN model, with approximately 11 h and 23 h for 2D geometries and 3D geometries, respectively. The prediction time is negligible compared to the training time and is also much shorter than the IGA simulation time. To evaluate the computational cost improvement of the PGNN model, we define a speedup ratio as the ratio between the IGA computation time

Table 3
Computational cost of all tested complex neuron geometries.

Dimension	Model name	IGA simulation time (mins)	PGNN prediction time (mins)	Speedup ratio (IGA vs PGNN)	PGNN training time (mins)
2D	NMO_54504, reduced MTs (Fig. 4A&B)	16.4	0.6	27.3	653
	NMO_54504, MT swirls (Fig. 4C&D)	21.2	0.9	23.6	
	NMO_54499, reduced MTs (Fig. 4E&F)	29.7	1.1	27	
	NMO_54499, MT swirls (Fig. 4G&H)	41.3	1.7	24.1	
3D	NMO_54499, reduced MTs (Fig. 6A-F)	136.8	3.4	40.2	1402
	NMO_54499, MT swirls (Fig. 7A-F)	161.6	4.1	39.4	
	NMO_54499, reduced MTs (Fig. 6G-L)	230.4	5.2	44.3	
	NMO_54499, MT swirls (Fig. 7G-L)	281.1	6.1	46.1	

and the PGNN prediction time. We find that the PGNN model achieves up to 27 times faster compared to 2D IGA simulations and the speedup ratio further increases to 46 when performing prediction in 3D geometries, indicating that the PGNN model can serve as an efficient tool for predicting complex transport phenomena in neurons.

5. Conclusion and future work

In this paper, we develop an IGA-based PGNN model to predict and study the complicated traffic jam phenomena during neuron material transport process. Built upon our previous GNN-based deep learning model [47], the proposed method is adaptive to any neuron morphology by using a graph representation of the neuron. The basic idea is to treat any neuron morphology as a graph consisting of only pipe and bifurcation structures. Then, the PGNN model employs physics-informed simulators to provide high accurate prediction in these simple geometries and uses an assembly model to obtain the global results in the entire geometry. Compared to [47], the proposed PGNN framework mainly features a physics-informed loss function that is evaluated using the spline information in IGA. The cubic B-spline basis functions are represented with the Bézier extraction operator and integrated into the weak formulation of the PDE residuals. In this way, both input control mesh and spline information are reused in the PGNN model to provide essential geometry information for prediction.

The proposed method also supports traffic jam prediction thanks to the training dataset obtained from a PDE-CO transport model [38,39]. Given input boundary conditions, geometry information and transport conditions, the well trained PGNN simulators are able to provide high accuracy results (MRE $< 9\%$) in both 2D and 3D neuron geometries. Using our IGA-based PGNN model, we study abnormal transport processes in different geometries and discover several spatial patterns of the transport process. In particular, we examine the formation of traffic jams due to the reduction of MTs and MT swirls in the local region. The PGNN model effectively predicts the material aggregation and velocity reduction in the traffic jam region. Specifically, the concentration distribution is precisely inferred with minor error on the magnitude and location of the maximum concentration region in comparison with the IGA results (Figs. 5&8). We also observe that the local traffic jam decreases the material transported to the downstream outlets, and may even induces material accumulation in the neighboring region to mitigate the traffic. Moreover, the PGNN model also successfully captures the unique spatial patterns of transport velocities during traffic jam, such as vortex and reversing streamlines. These results further explain how the non-uniform MT distributions affect the transport velocity and hinder the smooth delivery of the material.

Our study has its limitations, which we will address in the future work. First, the current model is limited to only two types of traffic jams in the pipe geometry. It does not take into account traffic jam occurs in the bifurcation geometry. Nor does it take into account traffic jam caused by other defects during transport, such as the failure of molecular motors [69]. Second, the MT distribution $l_{\pm}$ has the same pattern assigned to different pipes for the data generation of each traffic jam, which limits the prediction of complex concentration and velocity patterns caused by different $l_{\pm}$ settings. Furthermore, the assembly model only considers the assembly between different geometries but neglects the transport status of the geometry. For instance, the assembly model is not specifically trained to handle the assembly between a traffic jam simulator and a normal simulator. To further improve the performance of our PGNN model, we will optimize the training dataset by including different types of traffic jams and simulator assembly, as well as increasing the variety of $l_{\pm}$ distributions to generalize its application to handle more complicated

transport conditions in neurons. Despite these limitations, the proposed IGA-based PGNN method manages to learn the underlying transport mechanisms from simulation data and infer the transport process under different transport conditions, making it a useful tool for determining the potential transport disruption for clinical research. The model can also be extended to a general deep learning framework for solving other PDE-CO models in complex geometry.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and data availability

The source code for our model and all input data are available for download from a public software repository located at https://github.com/CMU-CBML/NeuronTransportPGNN. All data generated during this study can be reconstructed by running the source code.

Acknowledgments

The authors acknowledge the support of NSF, USA grant CMMI-1953323, a PITA (Pennsylvania Infrastructure Technology Alliance), USA, and a PMFI (Pennsylvania Manufacturing Fellows Initiative), USA grant. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation, USA grant number ACI-1548562. Specifically, it used the Bridges-2 system, which is supported by NSF award number ACI-1928147, at the Pittsburgh Supercomputing Center (PSC).

Appendix. GN block algorithm

The detailed computation of our GN block is characterized by Algorithm A1.

Algorithm A1: GN Block

Input: Graph, $G = (V, E, \boldsymbol{x}_v)$
for *each edge* $e_{ij} \in E$ **do**
  Compute edge attributes: $\boldsymbol{x}_{e_{ij}} = \phi_e(\boldsymbol{x}_{v_i}, \boldsymbol{x}_{v_j})$;
end
for *each node* $v_i \in V$ **do**
  Aggregate edge attributes: $\boldsymbol{x}_{e,v_i} = \sum_j \boldsymbol{x}_{e_{ij}}$;
  Compute output: $\boldsymbol{x}'_{v_i} = \phi_v(\boldsymbol{x}_{v_i}, \boldsymbol{x}_{e,v_i})$;
end
Output: Graph, $G = (V, E, \boldsymbol{x}'_v)$

References

[1] N. Hirokawa, S. Niwa, Y. Tanaka, Molecular motors in neurons: transport mechanisms and roles in brain function, development, and disease, Neuron 68 (4) (2010) 610–638.
[2] S. Maday, A.E. Twelvetrees, A.J. Moughamian, E.L. Holzbaur, Axonal transport: cargo-specific mechanisms of motility and regulation, Neuron 84 (2) (2014) 292–309.
[3] C.I. Maeder, K. Shen, C.C. Hoogenraad, Axon and dendritic trafficking, Curr. Opin. Neurobiol. 27 (2014) 165–170.
[4] K.J. De Vos, A.J. Grierson, S. Ackerley, C.C. Miller, Role of axonal transport in neurodegenerative diseases, Nat. Rev. Neurosci. 31 (2008) 151–173.
[5] S. Gunawardena, L.S. Goldstein, Polyglutamine diseases and transport problems: deadly traffic jams on neuronal highways, Arch. Neurol. 62 (1) (2005) 46–51.
[6] S. Millecamps, J.-P. Julien, Axonal transport deficits and neurodegenerative diseases, Nat. Rev. Neurosci. 14 (3) (2013) 161–176.
[7] N.L. Kononenko, G.A. Claßen, M. Kuijpers, D. Puchkov, T. Maritzen, A. Tempes, A.R. Malik, A. Skalecka, S. Bera, J. Jaworski, et al., Retrograde transport of TrkB-containing autophagosomes via the adaptor AP-2 mediates neuronal complexity and prevents neurodegeneration, Nature Commun. 8 (1) (2017) 1–16.

[8] H. Zhang, C. Zhang, J. Vincent, D. Zala, C. Benstaali, M. Sainlos, D. Grillo-Bosch, S. Daburon, F. Coussen, Y. Cho, et al., Modulation of AMPA receptor surface diffusion restores hippocampal plasticity and memory in Huntington's disease models, Nature Commun. 9 (1) (2018) 1–16.
[9] S. Roy, Seeing the unseen: the hidden world of slow axonal transport, Neuroscientist 20 (1) (2014) 71–81.
[10] E. Chevalier-Larsen, E.L. Holzbaur, Axonal transport and neurodegenerative disease, Biochim. Biophys. Acta (BBA)-Mol. Basis Dis. 1762 (11–12) (2006) 1094–1108.
[11] S.R. May, A.M. Ashique, M. Karlen, B. Wang, Y. Shen, K. Zarbalis, J. Reiter, J. Ericson, A.S. Peterson, Loss of the retrograde motor for IFT disrupts localization of smo to cilia and prevents the expression of both activator and repressor functions of gli, Dev. Biol. 287 (2) (2005) 378–389.
[12] J.J. Blum, M.C. Reed, A model for fast axonal transport, Cell Motil. 5 (6) (1985) 507–527.
[13] J. Blum, M. Reed, A model for slow axonal transport and its application to neurofilamentous neuropathies, Cell Motil. Cytoskeleton 12 (1) (1989) 53–65.
[14] M.C. Reed, S. Venakides, J.J. Blum, Approximate traveling waves in linear reaction-hyperbolic equations, SIAM J. Appl. Math. 50 (1) (1990) 167–180.
[15] E.A. Brooks, Probabilistic methods for a linear reaction-hyperbolic system with constant coefficients, Ann. Appl. Probab. 9 (3) (1999) 719–731.
[16] L. Popovic, S.A. McKinley, M.C. Reed, A stochastic compartmental model for fast axonal transport, SIAM J. Appl. Math. 71 (4) (2011) 1531–1556.
[17] C. Xue, B. Shtylla, A. Brown, A stochastic multiscale model that explains the segregation of axonal microtubules and neurofilaments in neurological diseases, PLoS Comput. Biol. 11 (8) (2015) e1004406.
[18] A. Kuznetsov, A. Avramenko, A macroscopic model of traffic jams in axons, Math. Biosci. 218 (2) (2009) 142–152.
[19] A. Kuznetsov, Effect of the degree of polar mismatching on traffic jam formation in fast axonal transport, Comput. Methods Biomech. Biomed. Eng. 13 (6) (2010) 711–722.
[20] A. Kuznetsov, Effect of vesicle traps on traffic jam formation in fast axonal transport, Math. Biosci. 226 (2) (2010) 147–155.
[21] M. Bertsch, B. Franchi, N. Marcello, M.C. Tesi, A. Tosin, Alzheimer's disease: a mathematical model for onset and progression, Math. Med. Biol.: J. IMA 34 (2) (2017) 193–214.
[22] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195.
[23] D. Benson, Y. Bazilevs, M.-C. Hsu, T.J. Hughes, Isogeometric shell analysis: the Reissner–Mindlin shell, Comput. Methods Appl. Mech. Engrg. 199 (5–8) (2010) 276–289.
[24] H. Casquero, L. Liu, Y. Zhang, A. Reali, J. Kiendl, H. Gomez, Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells, Comput. Aided Des. 82 (2017) 140–153.
[25] H. Casquero, X. Wei, D. Toshniwal, A. Li, T.J. Hughes, J. Kiendl, Y.J. Zhang, Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112765.
[26] X. Wei, X. Li, K. Qian, T.J. Hughes, Y.J. Zhang, H. Casquero, Analysis-suitable unstructured T-splines: multiple extraordinary points per face, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114494.
[27] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, T.J. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Comput. Methods Appl. Mech. Engrg. 196 (29–30) (2007) 2943–2959.
[28] Y. Zhang, X. Liang, J. Ma, Y. Jing, M.J. Gonzales, C. Villongco, A. Krishnamurthy, L.R. Frank, V. Nigam, P. Stark, et al., An atlas-based geometry pipeline for cardiac Hermite model construction and diffusion tensor reorientation, Med. Image Anal. 16 (6) (2012) 1130–1141.
[29] Y. Zhang, Challenges and advances in image-based geometric modeling and mesh generation, in: Y.J. Zhang (Ed.), Image-Based Geometric Modeling and Mesh Generation, Springer, 2013, pp. 1–10.
[30] B. Urick, T.M. Sanders, S.S. Hossain, Y.J. Zhang, T.J. Hughes, Review of patient-specific vascular modeling: template-based isogeometric framework and the case for CAD, Arch. Comput. Methods Eng. 26 (2) (2019) 381–404.
[31] Y. Yu, Y.J. Zhang, K. Takizawa, T.E. Tezduyar, T. Sasaki, Anatomically realistic lumen motion representation in patient-specific space–time isogeometric flow analysis of coronary arteries with time-dependent medical-image data, Comput. Mech. 65 (2) (2020) 395–404.
[32] Y.J. Zhang, Geometric Modeling and Mesh Generation from Scanned Images, CRC Press, 2016.
[33] K. Qian, A. Pawar, A. Liao, C. Anitescu, V. Webster-Wood, A.W. Feinberg, T. Rabczuk, Y.J. Zhang, Modeling neuron growth using isogeometric collocation based phase field method, Sci. Rep. 12 (1) (2022) 1–10.
[34] A. Pawar, Y.J. Zhang, NeuronSeg_BACH: automated neuron segmentation using B-spline based active contour and hyperelastic regularization, Commun. Comput. Phys. 28 (3) (2020) 1219–1244.
[35] Y. Yu, X. Wei, A. Li, J.G. Liu, J. He, Y.J. Zhang, HexGen and Hex2Spline: polycube-based hexahedral mesh generation and spline modeling for isogeometric analysis applications in LS-DYNA, in: Springer INdAM Serie: Proceedings of INdAM Workshop "Geometric Challenges in Isogeometric Analysis", Springer, 2020.
[36] Y. Yu, J.G. Liu, Y.J. Zhang, HexDom: polycube-based hexahedral-dominant mesh generation, in: K.M. Rebén Sevilla (Ed.), The Edited Volume of Mesh Generation and Adaptation: Cutting-Edge Techniques for the 60th Birthday of Oubay Hassan. SEMA-SIMAI Springer Series, Springer, 2021.
[37] A. Li, X. Chai, G. Yang, Y.J. Zhang, An isogeometric analysis computational platform for material transport simulation in complex neurite networks, Mol. Cell. Biomech. 16 (2) (2019) 123–140.
[38] A. Li, Y.J. Zhang, Modeling material transport regulation and traffic jam in neurons using PDE-constrained optimization, Sci. Rep. 12 (1) (2022) 1–13.

[39] A. Li, Y.J. Zhang, Modeling intracellular transport and traffic jam in 3D neurons using PDE-constrained optimization, J. Mech. 38 (2022) 44–59.
[40] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[41] L. Sun, H. Gao, S. Pan, J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, Comput. Methods Appl. Mech. Engrg. 361 (2020) 112732.
[42] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, Science 367 (6481) (2020) 1026–1030.
[43] G. Kissas, Y. Yang, E. Hwuang, W.R. Witschey, J.A. Detre, P. Perdikaris, Machine learning in cardiovascular flows modeling: predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112623.
[44] X. Jin, S. Cai, H. Li, G.E. Karniadakis, NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations, J. Comput. Phys. 426 (2021) 109951.
[45] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112789.
[46] H. Gao, L. Sun, J.-X. Wang, PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, J. Comput. Phys. 428 (2021) 110079.
[47] A. Li, A.B. Farimani, Y.J. Zhang, Deep learning of material transport in complex neurite networks, Sci. Rep. 11 (1) (2021) 1–13.
[48] D. Smith, R. Simmons, Models of motor-assisted transport of intracellular particles, Biophys. J. 80 (1) (2001) 45–68.
[49] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (1–4) (2007) 173–201.
[50] H. Zempel, E.-M. Mandelkow, Linking amyloid-β and tau: amyloid-β induced synaptic dysfunction via local wreckage of the neuronal cytoskeleton, Neurodegener. Dis. 10 (1–4) (2012) 64–72.
[51] A.J. Matamoros, P.W. Baas, Microtubules in health and degenerative disease of the nervous system, Brain Res. Bull. 126 (2016) 217–225.
[52] O.A. Shemesh, H. Erez, I. Ginzburg, M.E. Spira, Tau-induced traffic jams reflect organelles accumulation at points of microtubule polar mismatching, Traffic 9 (4) (2008) 458–471.
[53] X. Wei, Y.J. Zhang, T.J. Hughes, Truncated hierarchical tricubic C^0 spline construction on unstructured hexahedral meshes for isogeometric analysis applications, Comput. Math. Appl. 74 (9) (2017) 2203–2220.
[54] M. Stoll, A. Wathen, All-at-once solution of time-dependent Stokes control, J. Comput. Phys. 232 (1) (2013) 498–515.
[55] F. Yılmaz, B. Karasözen, An all-at-once approach for the optimal control of the unsteady Burgers equation, J. Comput. Appl. Math. 259 (2014) 771–779.
[56] M.J. Borden, M.A. Scott, J.A. Evans, T.J. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, Internat. J. Numer. Methods Engrg. 87 (1–5) (2011) 15–47.
[57] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, T.J. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, Internat. J. Numer. Methods Engrg. 88 (2) (2011) 126–156.
[58] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 2012.
[59] U. Reif, A refineable space of smooth spline surfaces of arbitrary topological genus, J. Approx. Theory 90 (2) (1997) 174–199.
[60] D. Toshniwal, H. Speleers, T.J. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: geometric design and isogeometric analysis considerations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 411–458.
[61] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.
[62] S. Abhyankar, J. Brown, E.M. Constantinescu, D. Ghosh, B.F. Smith, H. Zhang, PETSc/TS: a modern scalable ODE/DAE solver library, 2018, arXiv:1806.01437.
[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.
[64] M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch geometric, 2019, arXiv preprint arXiv:1903.02428.
[65] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G.D. Peterson, et al., XSEDE: accelerating scientific discovery, Comput. Sci. Eng. 16 (5) (2014) 62–74.
[66] N.A. Nystrom, M.J. Levine, R.Z. Roskies, J.R. Scott, Bridges: a uniquely flexible HPC resource for new communities and data analytics, in: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled By Enhanced Cyberinfrastructure, 2015, pp. 1–8.
[67] G.A. Ascoli, D.E. Donohue, M. Halavi, NeuroMorpho.org: a central resource for neuronal morphologies, J. Neurosci. 27 (35) (2007) 9247–9251.
[68] H. Gao, M.J. Zahr, J.-X. Wang, Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems, Comput. Methods Appl. Mech. Engrg. 390 (2022) 114502.
[69] M. Martin, S.J. Iyadurai, A. Gassman, J.G. Gindhart Jr., T.S. Hays, W.M. Saxton, Cytoplasmic dynein, the dynactin complex, and kinesin are interdependent and essential for fast axonal transport, Mol. Biol. Cell 10 (11) (1999) 3717–3728.