SC15-009: Recent Advances in Physics-Informed Deep Learning

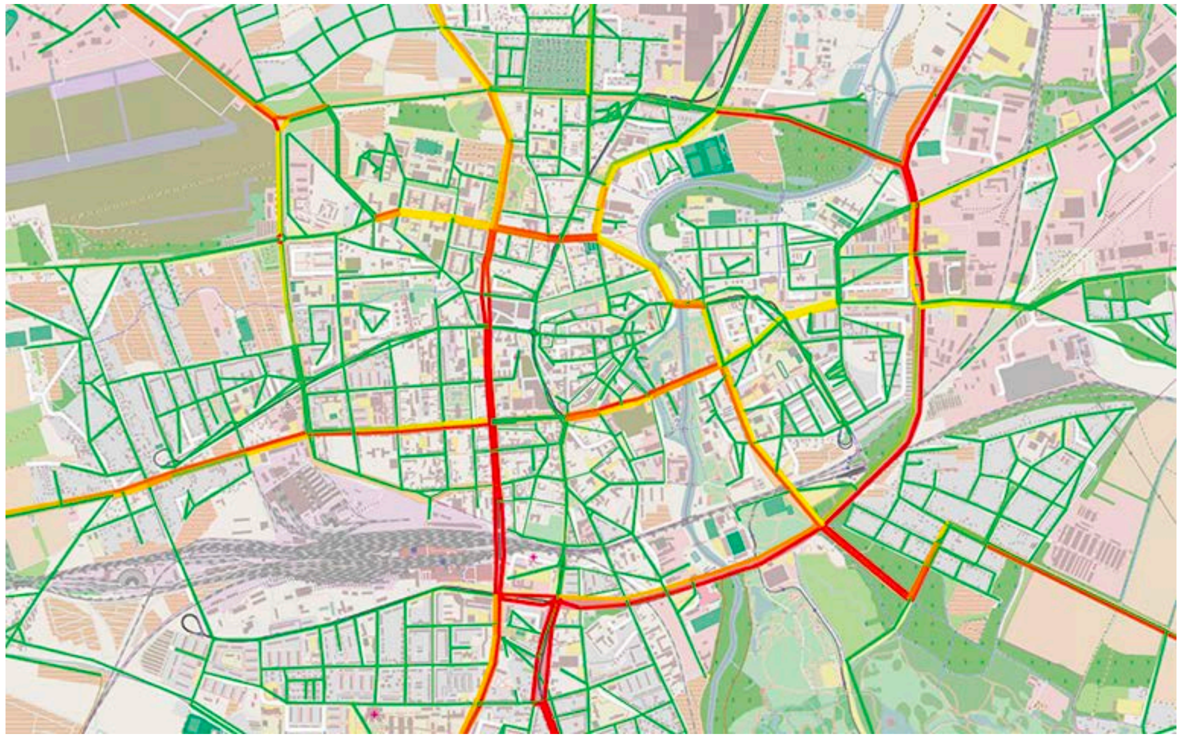
Solving PDEs on graphs with physics-informed neural networks

Paris Perdikaris
Department of Mechanical Engineering
University of Pennsylvania
email: pgp@seas.upenn.edu

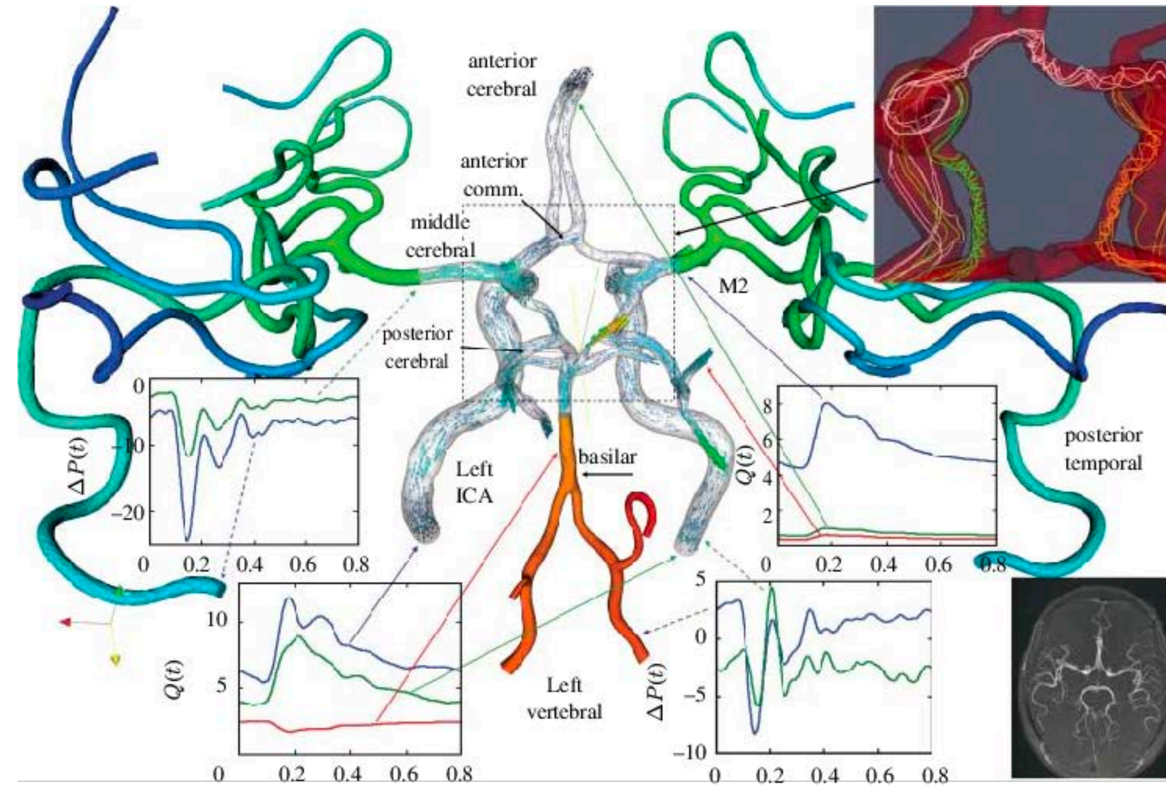
USNCCM15
Austin, TX
July 28, 2019

Penn
UNIVERSITY *of* PENNSYLVANIA

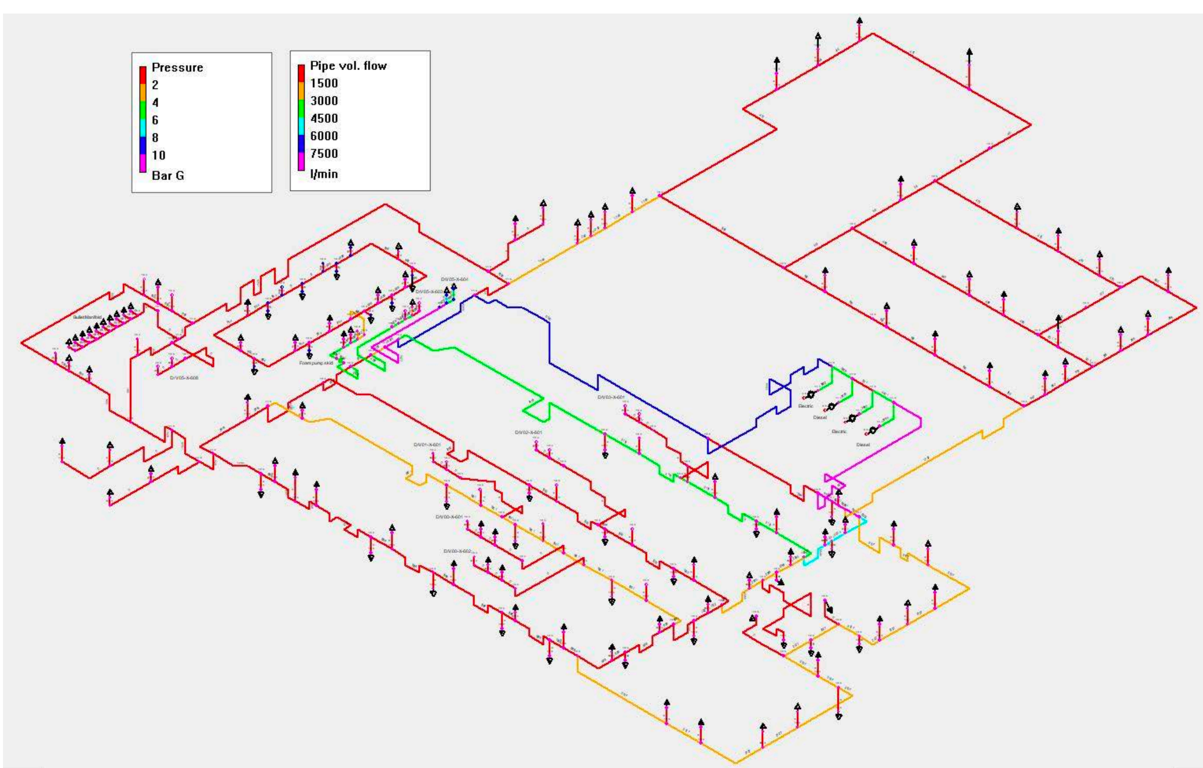
PDEs on graphs?



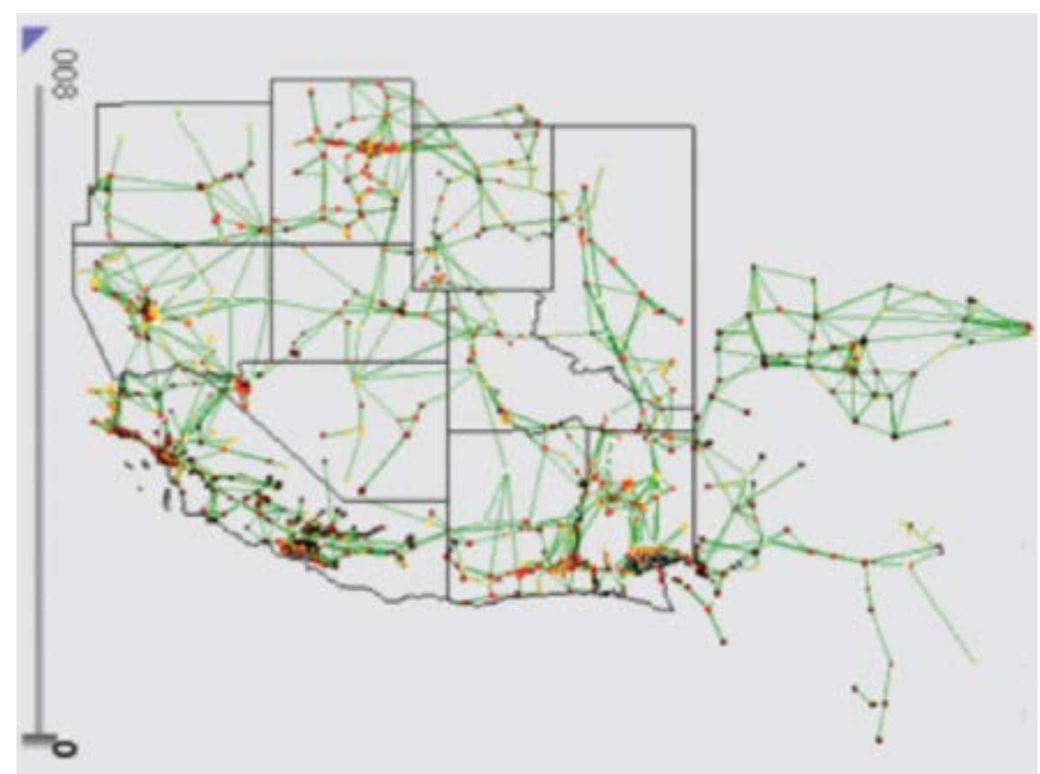
Traffic flow



Blood flow

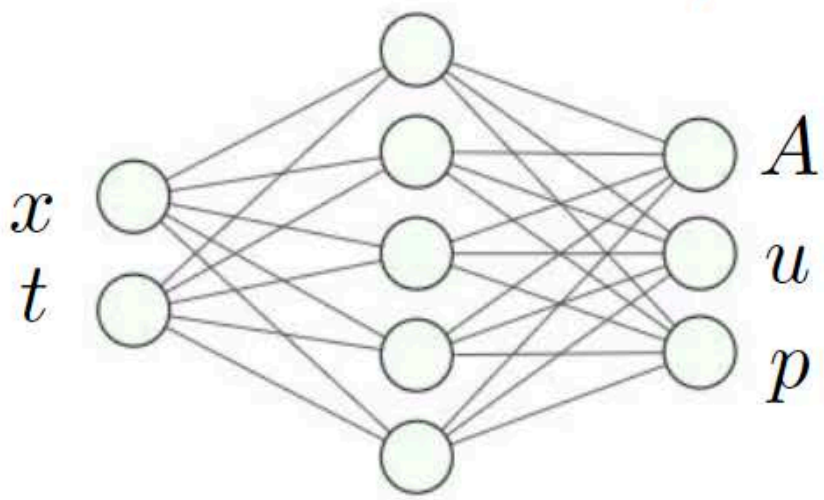


Cooling systems



Power grids

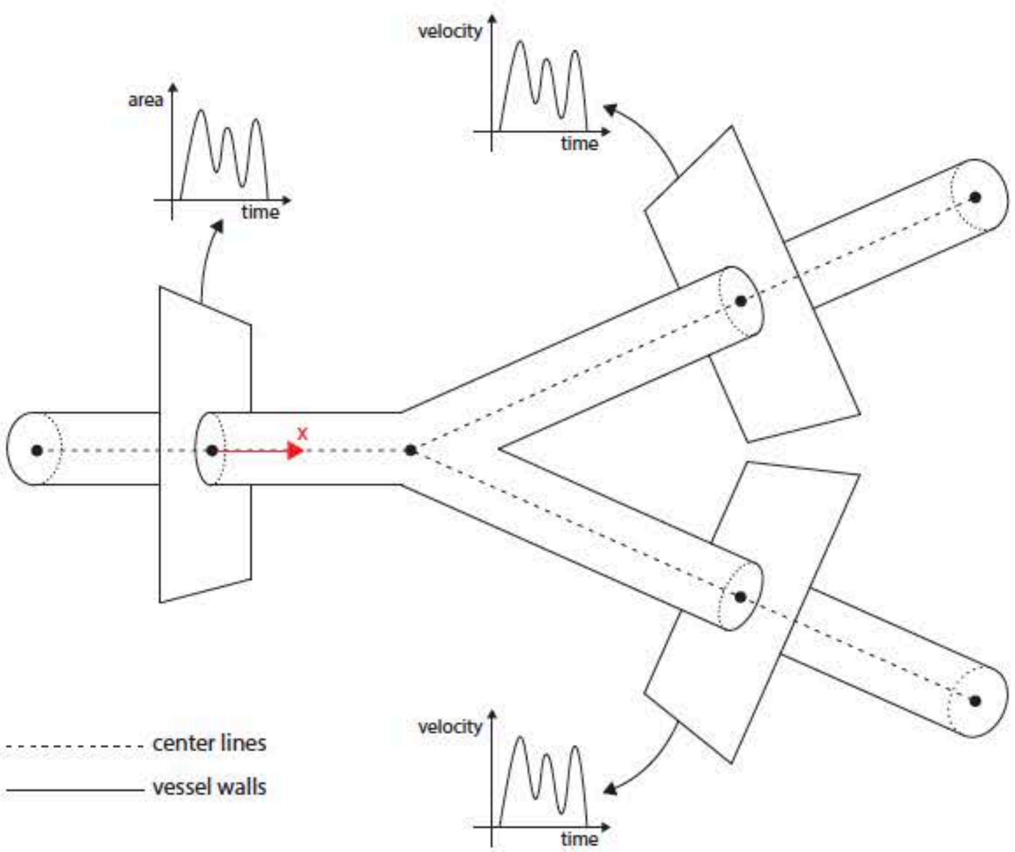
Physics-Informed neural networks



$$\mathcal{L} = \sum_{j=1}^{D_M} \mathcal{L}^j_{\text{measurement}} + \sum_{j=1}^{D} \mathcal{L}^j_{\text{residual}} + \sum_{k=1}^{D_I} \mathcal{L}^k_{\text{interface}}$$

Data:

Velocity & wall displacement



**no data for the pressure*

Physics:

1D pulsatile flow in compliant arteries

$$\frac{\partial A}{\partial t} + \frac{\partial Au}{\partial x} = 0,$$
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0,$$
$$p = p_{ext} + \beta(\sqrt{A} - \sqrt{A_0}).$$

**velocity, wall displacement and pressure are all correlated through the physics*

Interfaces:

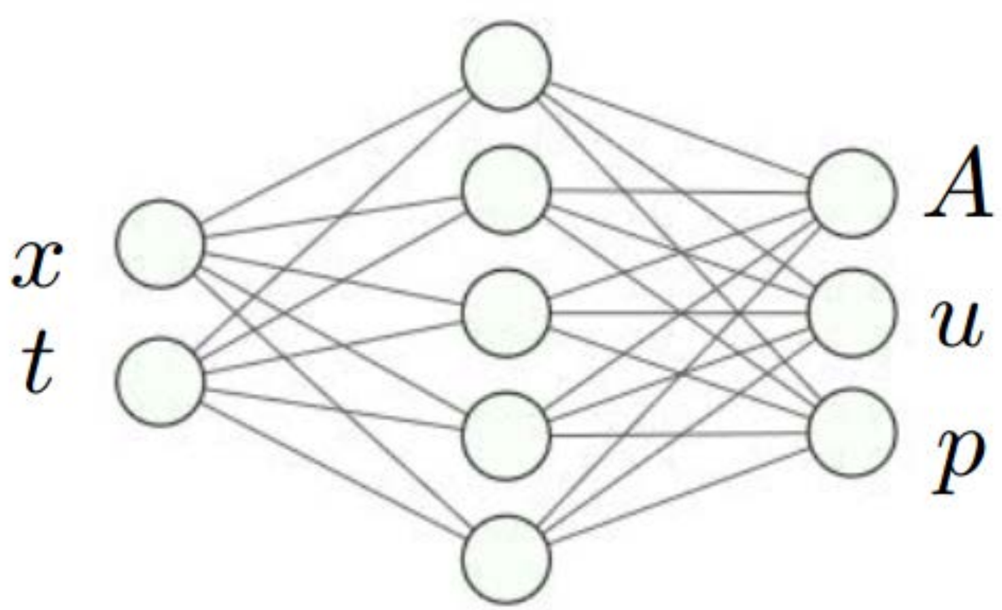
Conservation of mass and total pressure

$$A_1 u_1 = A_2 u_2 + A_3 u_3,$$
$$p_1 + \frac{1}{2}\rho u_1^2 = p_2 + \frac{1}{2}\rho u_2^2,$$
$$p_1 + \frac{1}{2}\rho u_1^2 = p_3 + \frac{1}{2}\rho u_3^2.$$

**conservation principles allow us to propagate information across the arterial network*

Neural Network architecture and optimization objective

We employ a fully-connected feed forward architecture:



or

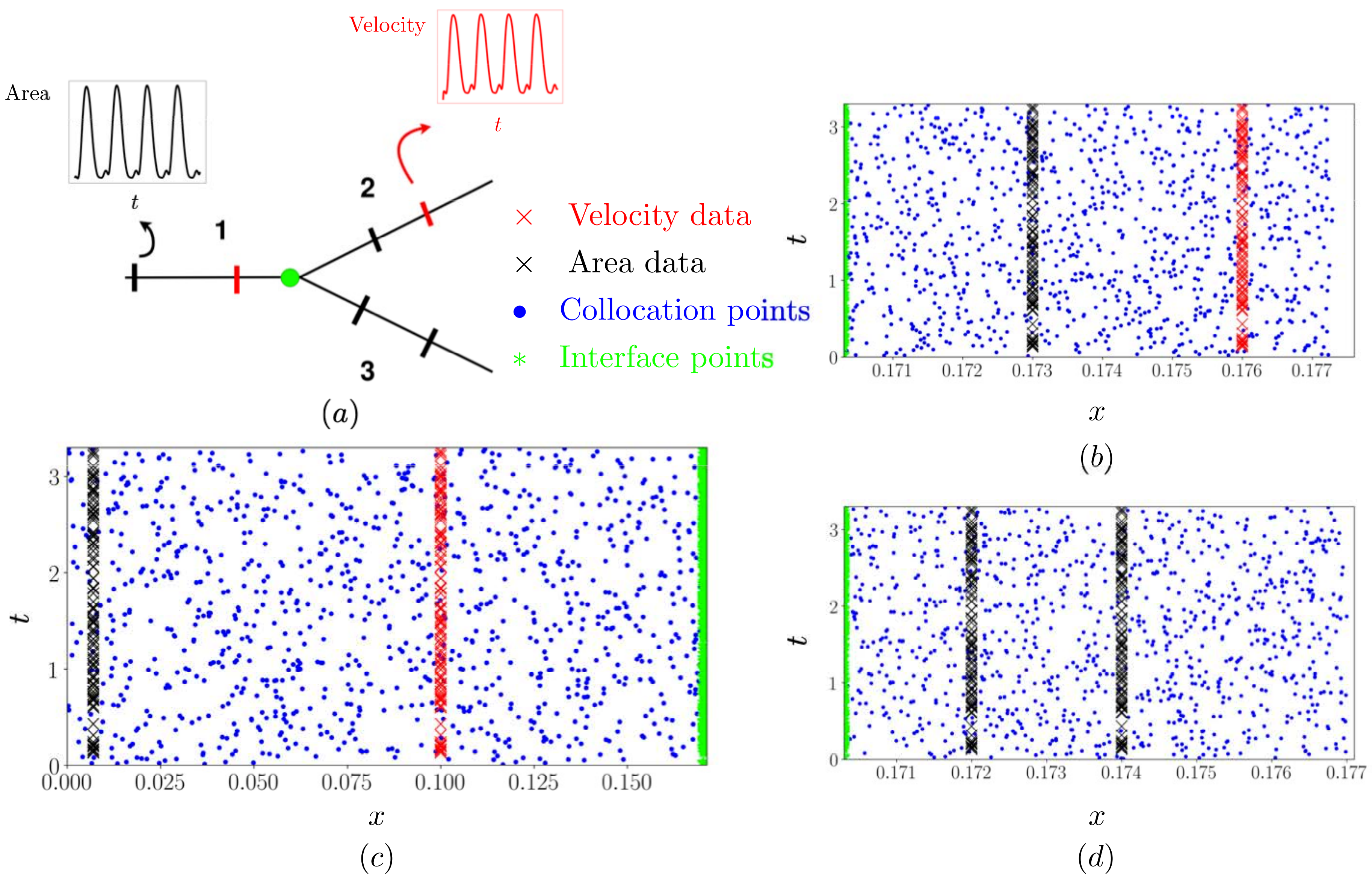
$$[x, t] \xrightarrow{f_{\theta^j}} [A^j(x,t), u^j(x,t), p^j(x,t)]$$

We employ a loss function of the form:

$$\mathcal{L} = \sum_{j=1}^{D_M} \mathcal{L}^j_{\text{measurement}} + \sum_{j=1}^{D} \mathcal{L}^j_{\text{residual}} + \sum_{k=1}^{D_I} \mathcal{L}^k_{\text{interface}}, \tag{8}$$

where $\mathcal{L}^j_{\text{measurement}}$, $\mathcal{L}^j_{\text{residual}}$ and $\mathcal{L}^k_{\text{interface}}$ correspond to the loss at **measurement** points, the **residual** loss of the PDEs and the loss at **interface** points, respectively.

Arterial network geometry and measurements



- **Velocity** and **wall displacement** measurements.
- **No pressure data are provided.**

Loss function: Measurements

This part of the loss function has the form (take vessel $\#j$ as example):

$$
\begin{aligned}
\mathcal{L}^j_{\text{measurement}} =& \frac{1}{N^j_u} \sum_{i=1}^{N^j_u} (u^j(x_i, t_i) - u^j(x_i, t_i; \theta^j))^2 + \\
&+ \frac{1}{N^j_A} \sum_{i=1}^{N^j_A} (A^j(x_i, t_i) - A^j(x_i, t_i; \theta^j))^2, \qquad j = 1, \dots, D_M,
\end{aligned} \tag{9}
$$

where N^j_A, N^j_u represent the number of measurements for A and u in vessel $\#j$ respectively. Also, D_M denote total number of vessels in which we have measurements. In the above equation $u^j(x_i, t_i; \theta^j)$ and $A^j(x_i, t_i; \theta^j)$ represent the outputs given the parameters o f the neural network for vessel $\#j$.

Loss function: Residuals

This collocation loss function takes the following form (take vessel $\#j$ as example):

$$\mathcal{L}^j_{\text{residual}} = \frac{1}{N_r^j}\sum_{i=1}^{N_r^j}(r_A^j(x_i,t_i;\theta^j))^2 + \frac{1}{N_r^j}\sum_{i=1}^{N_r^j}(r_u^j(x_i,t_i;\theta^j))^2 + \tag{10}$$

$$+ \frac{1}{N_r^j}\sum_{i=1}^{N_r^j}(r_p^j(x_i,t_i;\theta^j))^2, \qquad j = 1,\ldots,D, \tag{11}$$

where N_r^j represent the number of collocation points in vessel $\#j$. Also, D denote the total number of vessels in our arterial network. The terms $r_A^j(x_i,t_i;\theta^j)$, $r_u^j(x_i,t_i;\theta^j)$ and $r_p^j(x_i,t_i;\theta^j)$ represent the residual of area, velocity and pressure respectively.

Loss function: Interfaces

In general, the interface loss has the form (take bifurcation point $\#k$ as example):

$$
\begin{aligned}
\mathcal{L}^k_{\text{interface}} =& \frac{1}{N_b^k}\sum_{i=1}^{N_b^k}(A_1^k(x_k,t_i;\theta_1^k)u_1^k(x_k,t_i;\theta_1^k) - A_2^k(x_k,t_i;\theta_2^k)u_2^k(x_k,t_i;\theta_2^k)- \\
&-A_3^k(x_k,t_i;\theta_3^k)u_3^k(x_k,t_i;\theta_3^k))^2+ \\
&+\frac{1}{N_b^k}\sum_{i=1}^{N_b^k}(p_1^k(x_k,t_i;\theta_1^k) + \frac{1}{2}u_1^k(x_k,t_i;\theta_1^k)^2 - p_2^k(x_k,t_i;\theta_2^k)- \\
&-\frac{1}{2}u_2^k(x_k,t_i;\theta_2^k)^2)^2+ \\
&+\frac{1}{N_b^k}\sum_{i=1}^{N_b^k}(p_1^k(x_k,t_i;\theta_1^k) + \frac{1}{2}u_1^k(x_k,t_i;\theta_1^k)^2 - p_3^k(x_k,t_i;\theta_3^k)- \\
&-\frac{1}{2}u_3^k(x_k,t_i;\theta_3^k)^2)^2, \qquad k = 1,\dots,D_I,
\end{aligned}
\tag{12}
$$

where the indices 1, 2, 3 in $\mathcal{L}_b^k$ denote the father and daughter vessels, respectively, at each bifurcation. Also, D_I denotes the total number of bifurcation points in our arterial network. N_b^k represent the number of collocation points on the interface boundaries.

Normalization and non-dimensionalization

In equation (1), the order of magnitude of the different physical quantities, velocity, cross-sectional area and pressure, have a significant relative difference, e.g., $P \sim 10^6\ Pa$, $A \sim 10^{-5}\ m^2$ and $u \sim 10\ m/s$, which casts great difficulty on the training of the neural network [1]. For this purpose we choose:

$$L = \sqrt{\frac{1}{N}\sum_{j=1}^{D} = (A_0^j)}, \qquad U = 10,$$

where $j = 1, ..., D$. We define the quantities:

$$\hat{u} = \frac{u}{U}, \quad \hat{A} = \frac{A}{A^o}, \quad \hat{p} = \frac{p}{p_0}, \quad x_* = \frac{x}{L}, \quad t_* = \frac{t}{T}, \tag{13}$$

where $p_0 = \rho U^2$, $T = \dfrac{L}{U}$ and $A^o = L^2$.

[1] Glorot, X. and Bengio, Y., 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

Normalization and non-dimensionalization

It is shown that normalizing the input to have zero mean and unit variance makes the training of the neural network more efficient as it prevents gradient saturation and provides stable updates[2]. In this step, we normalize x_* and t_* to have zero mean and unit variance, in the vessel $\#j$, and get:

$$\hat{x}^j = \frac{x_*^j - \mu_{x_*}^j}{\sigma_{x_*}^j}, \quad \hat{t} = \frac{t_* - \mu_{t^*}}{\sigma_{t_*}}, \tag{14}$$

where $\mu_{x_*}^j$, $\sigma_{x_*}^j$ and $\mu_{t_*}^j$, $\sigma_{t_*}^j$ the mean and standard deviation of the spatial and temporal variables of $\#j$th vessel, respectively.

[2] Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Normalization and non-dimensionalization

Using the variables stated above, we derive the updated system of equations (take vessel $\#j$ as example):

$$\begin{aligned}
&\frac{1}{\sigma_{t_*}}\frac{\partial \hat{A}^j}{\partial \hat{t}} + \frac{1}{\sigma^j_{x_*}}\hat{A}^j\frac{\partial \hat{u}^j}{\partial \hat{x}^j} + \frac{1}{\sigma^j_{x_*}}\hat{u}^j\frac{\partial \hat{A}^j}{\partial \hat{x}^j} = 0,\\
&\frac{1}{\sigma_{t_*}}\frac{\partial \hat{u}^j}{\partial \hat{t}} + \frac{1}{\sigma^j_{x_*}}\hat{u}^j\frac{\partial \hat{u}^j}{\partial \hat{x}^j} + \frac{1}{\sigma^j_{x_*}}\frac{\partial \hat{p}^j}{\partial \hat{x}^j} = 0,\\
&\hat{p}^j = \frac{1}{p_0}(p_{ext} + \beta(\sqrt{\hat{A}^j A^o} - \sqrt{A_0})), \qquad j = 1, \ldots, D.
\end{aligned} \tag{15}$$

This is the non-dimensionalized and normalized form of the governing equations that is employed as a physical constraint in the residual part of the loss function $\mathcal{L}^j_{\text{residual}}$.

Normalization and non-dimensionalization

In order to be consistent with the derivation above, we have to follow the same procedure for every condition we impose to the model. In this notion we derive the non-dimensional continuity conditions, by inserting the non-dimensionalizing quantities into the conservation laws. By doing so, we get:

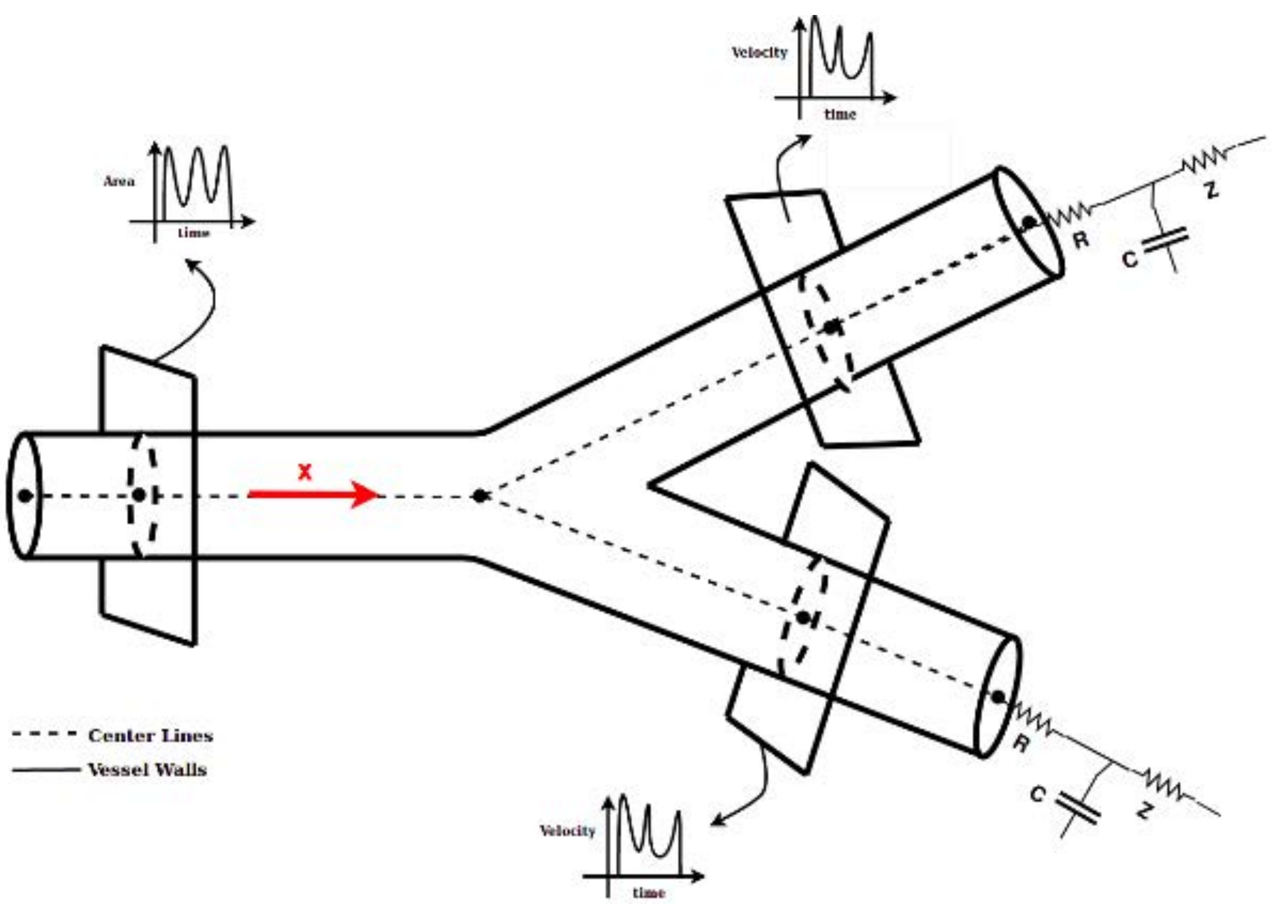
$$\hat{A}_1 \hat{u}_1 = \hat{A}_2 \hat{u}_2 + \hat{A}_3 \hat{u}_3, \tag{16}$$

$$\hat{p}_1 + \frac{1}{2}(\hat{u}_1)^2 = \hat{p}_2 + \frac{1}{2}(\hat{u}_2)^2, \tag{17}$$

$$\hat{p}_1 + \frac{1}{2}(\hat{u}_1)^2 = \hat{p}_3 + \frac{1}{2}(\hat{u}_3)^2. \tag{18}$$

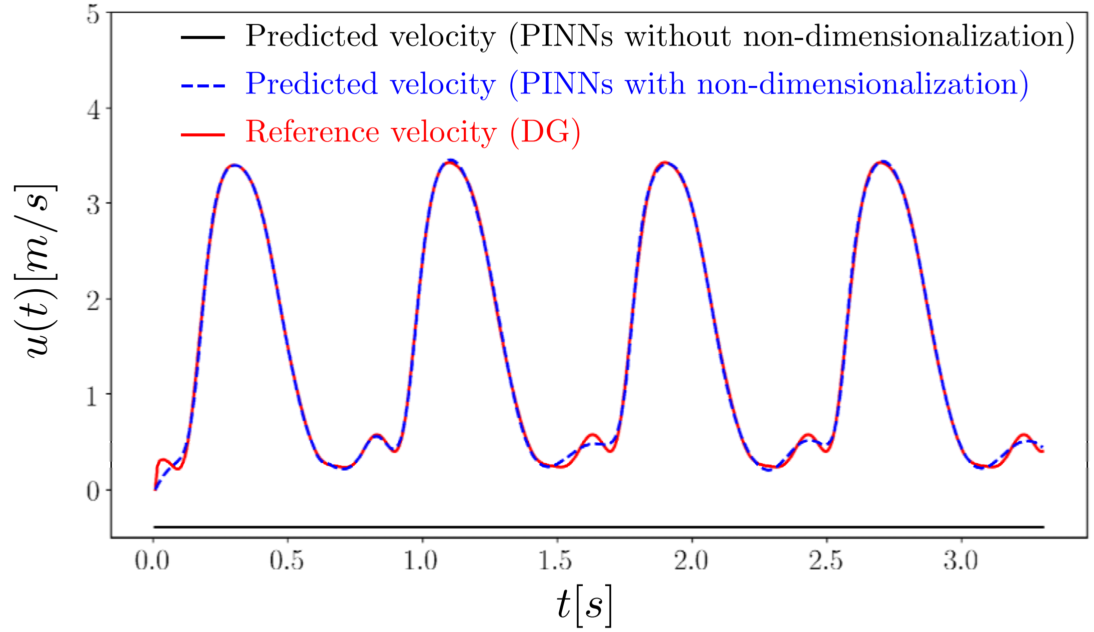
This form of the continuity equation is employed as a physical constraint in the interface part of the loss function $\mathcal{L}^j_{\text{interfaces}}$.

Flow through a prototype carotid bifurcation

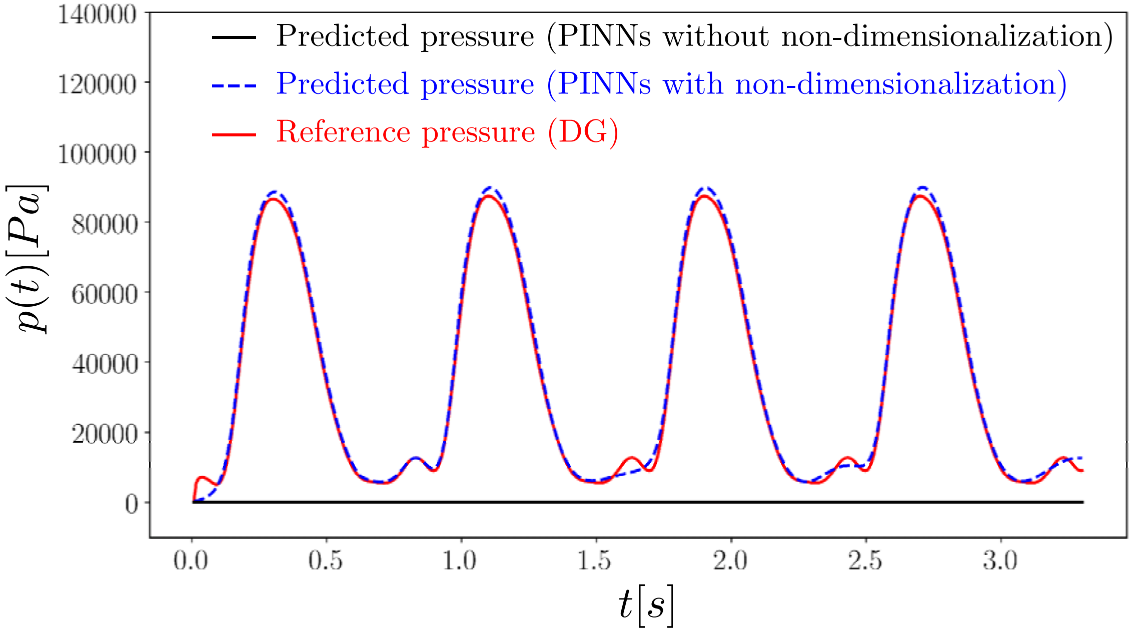


We assume we have obtained data for **cross-sectional area** and **velocity** at the inlets and outlets of a prototype carotid bifurcation. Our goal is given these measurements, to predict the values of the pressure at the middle points of the domain. We present the results for the case of **dimensionalized**, **non-dimensionalized** physics informed neural networks and the **reference solution** acquired by discontinuous Galerkin. asd

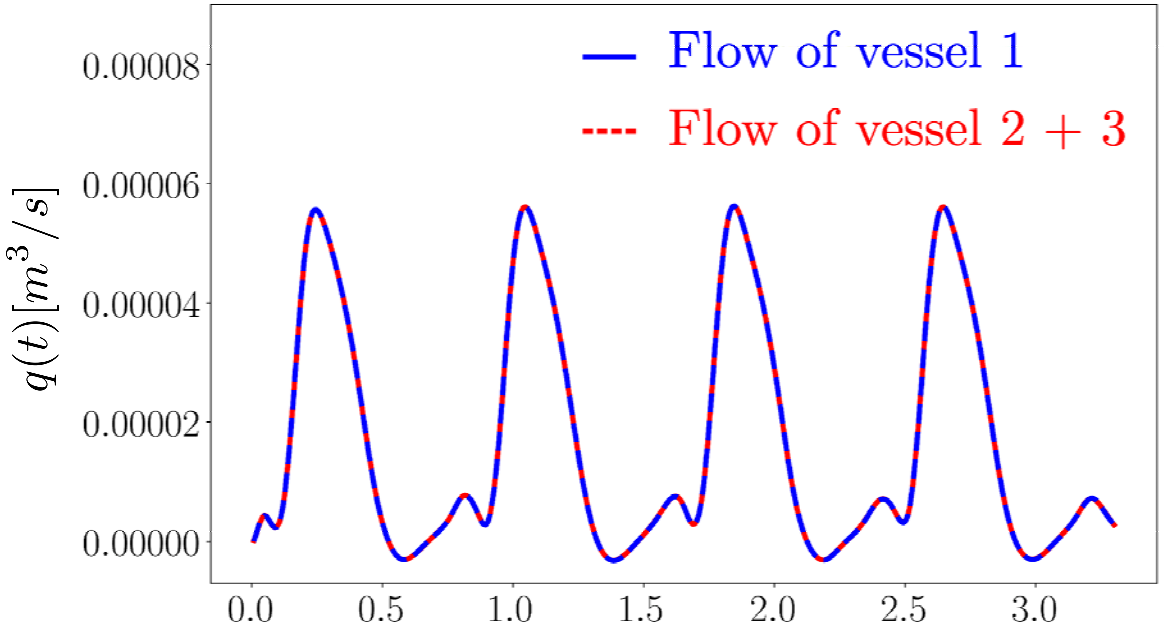
Flow through a prototype carotid bifurcation

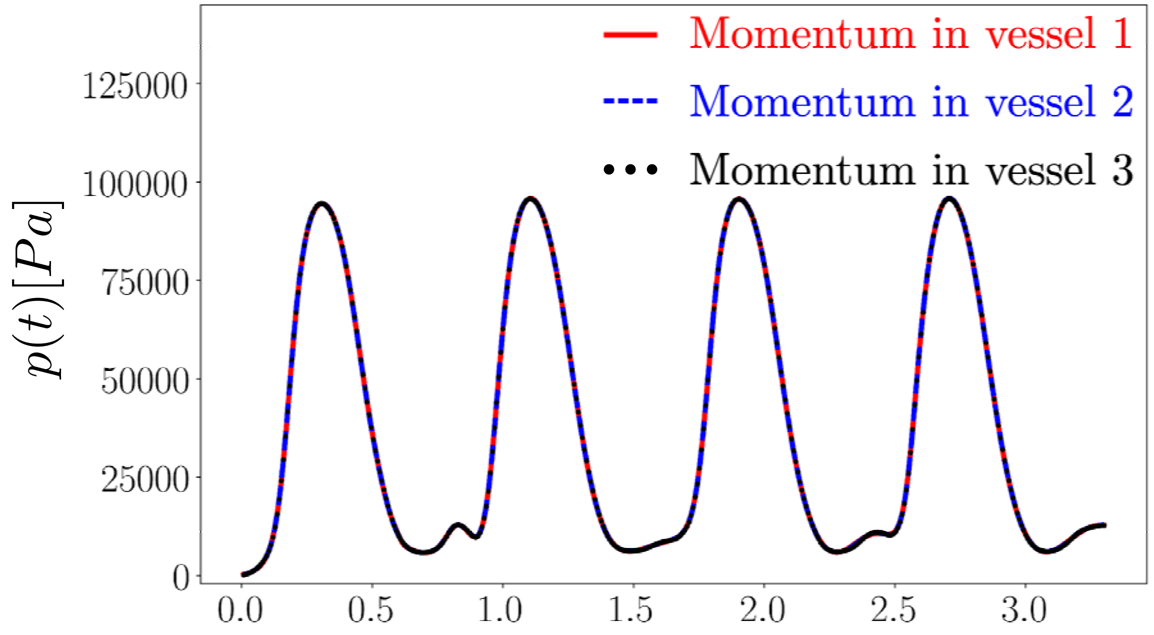


(a)

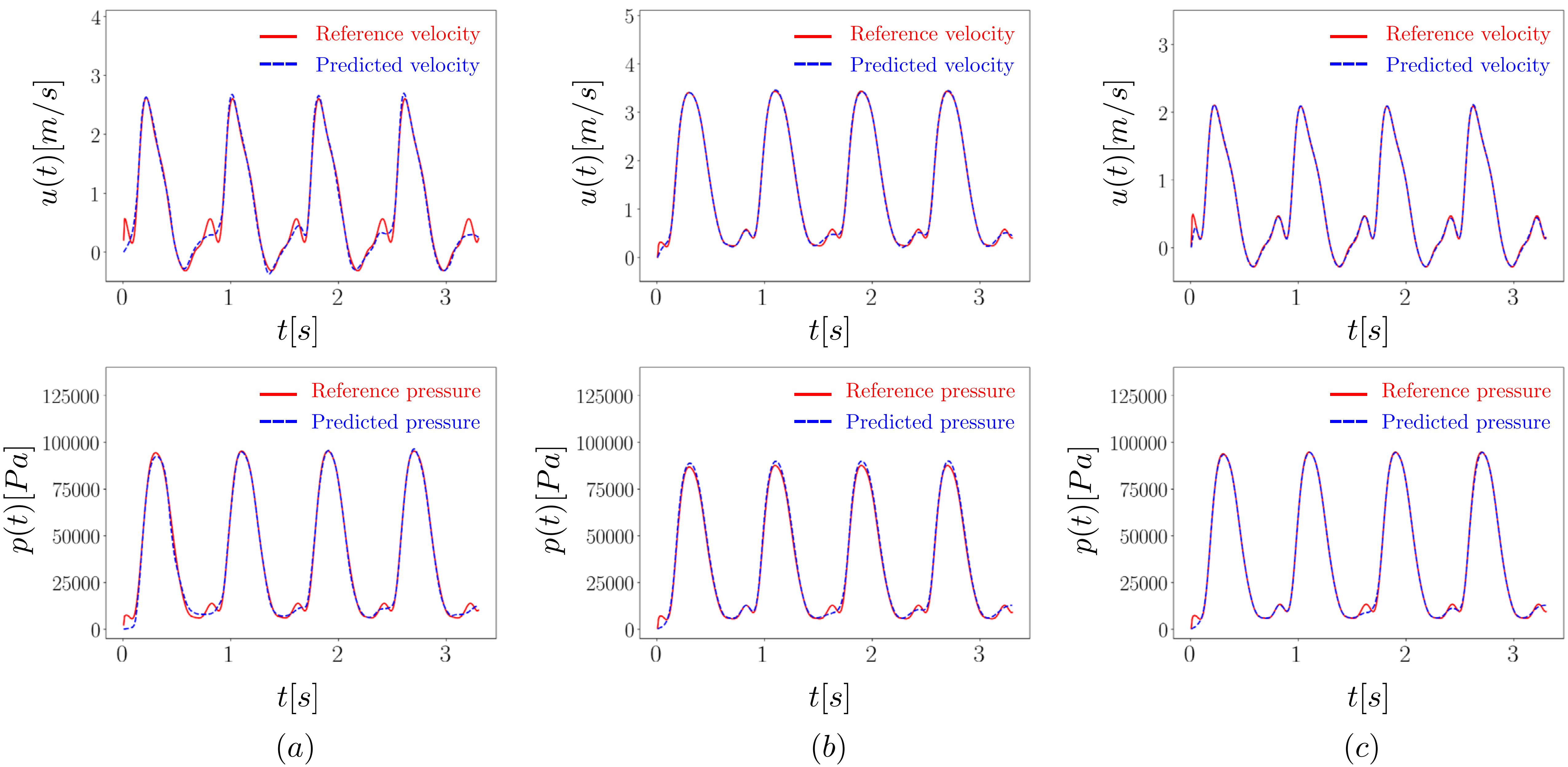


(b)

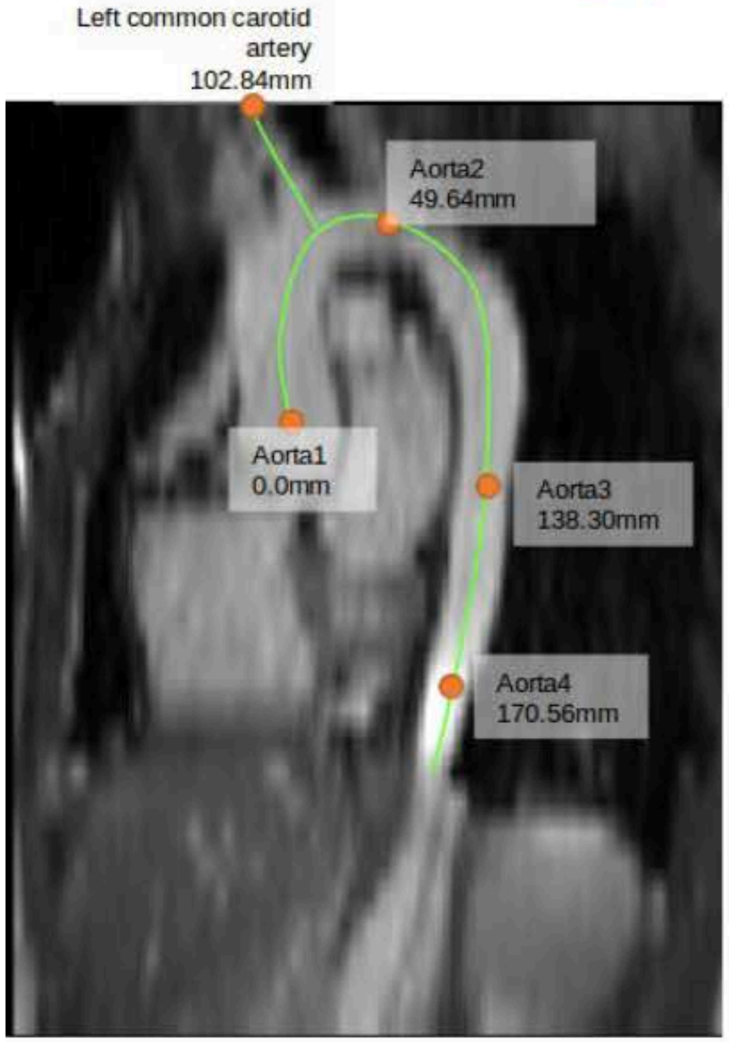




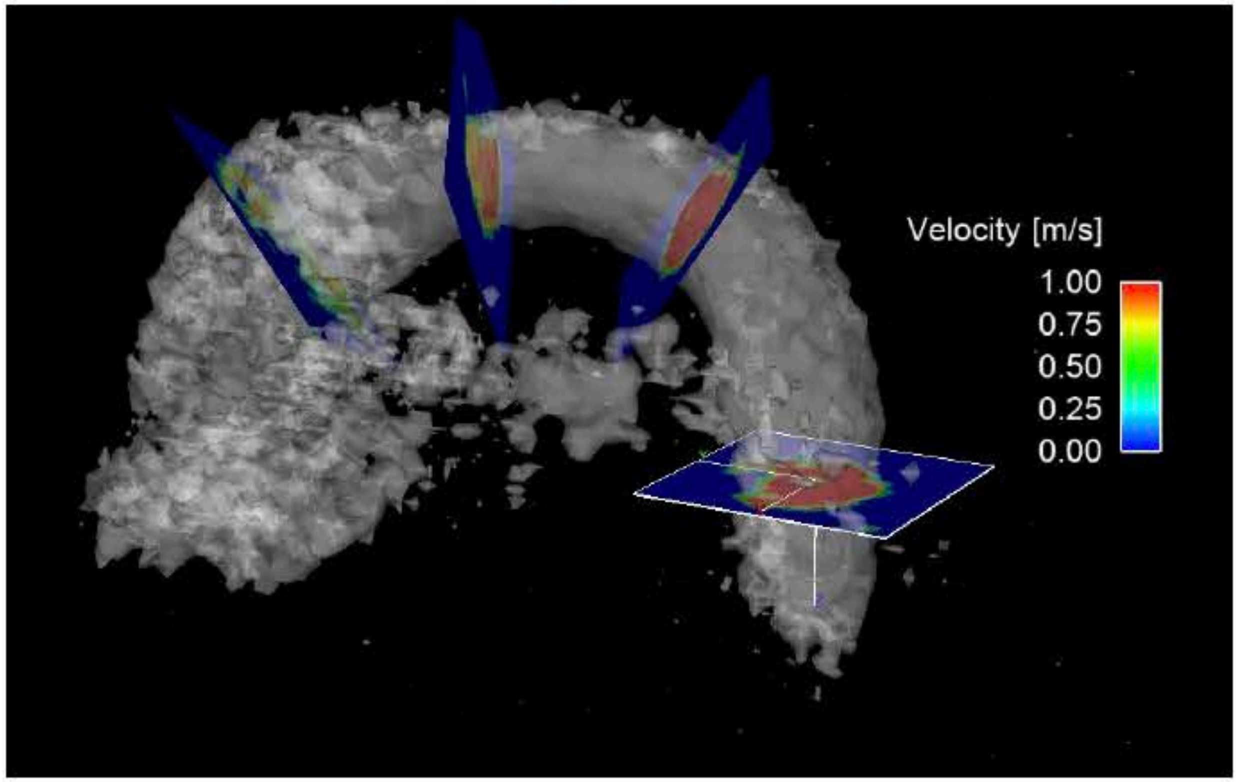
Flow through a prototype Y-shaped bifurcation



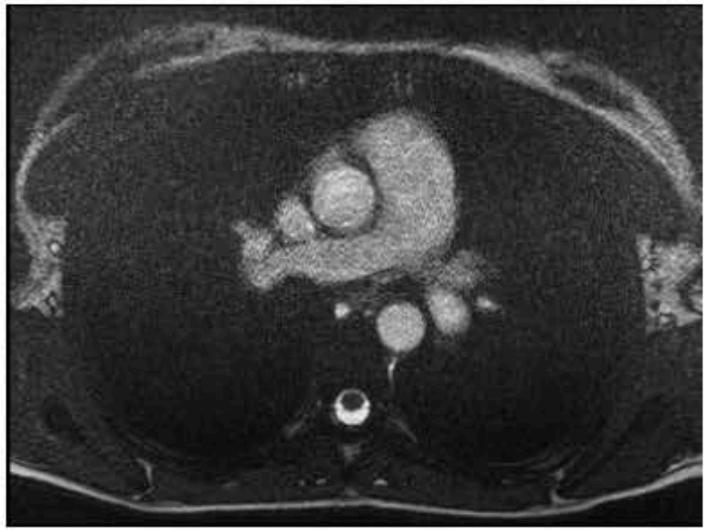
Predicting pulse wave propagation from 4D flow MRI data

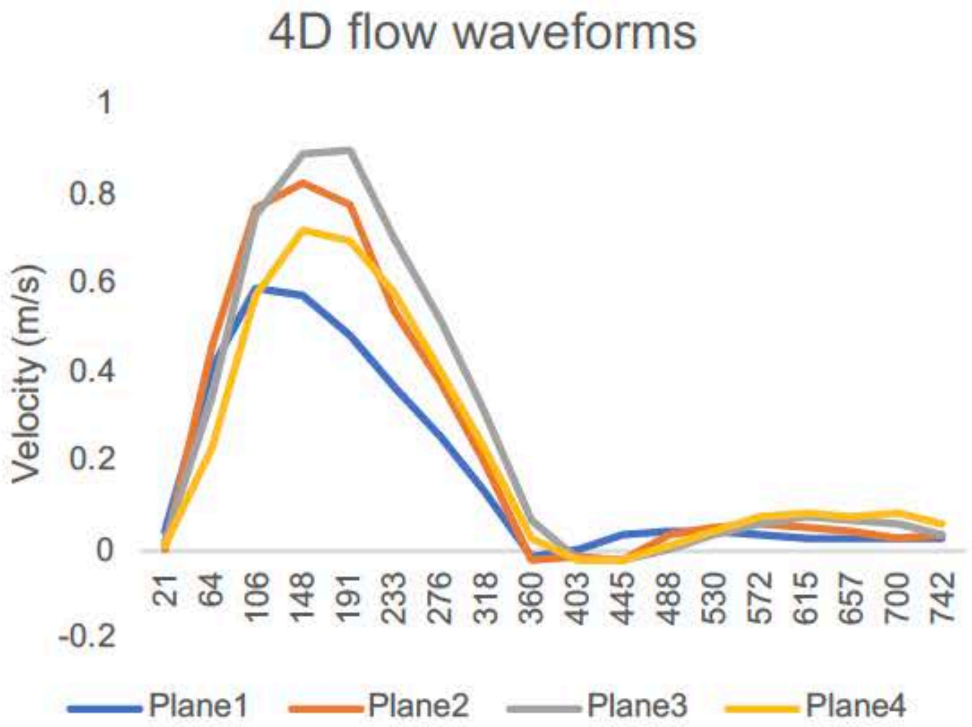


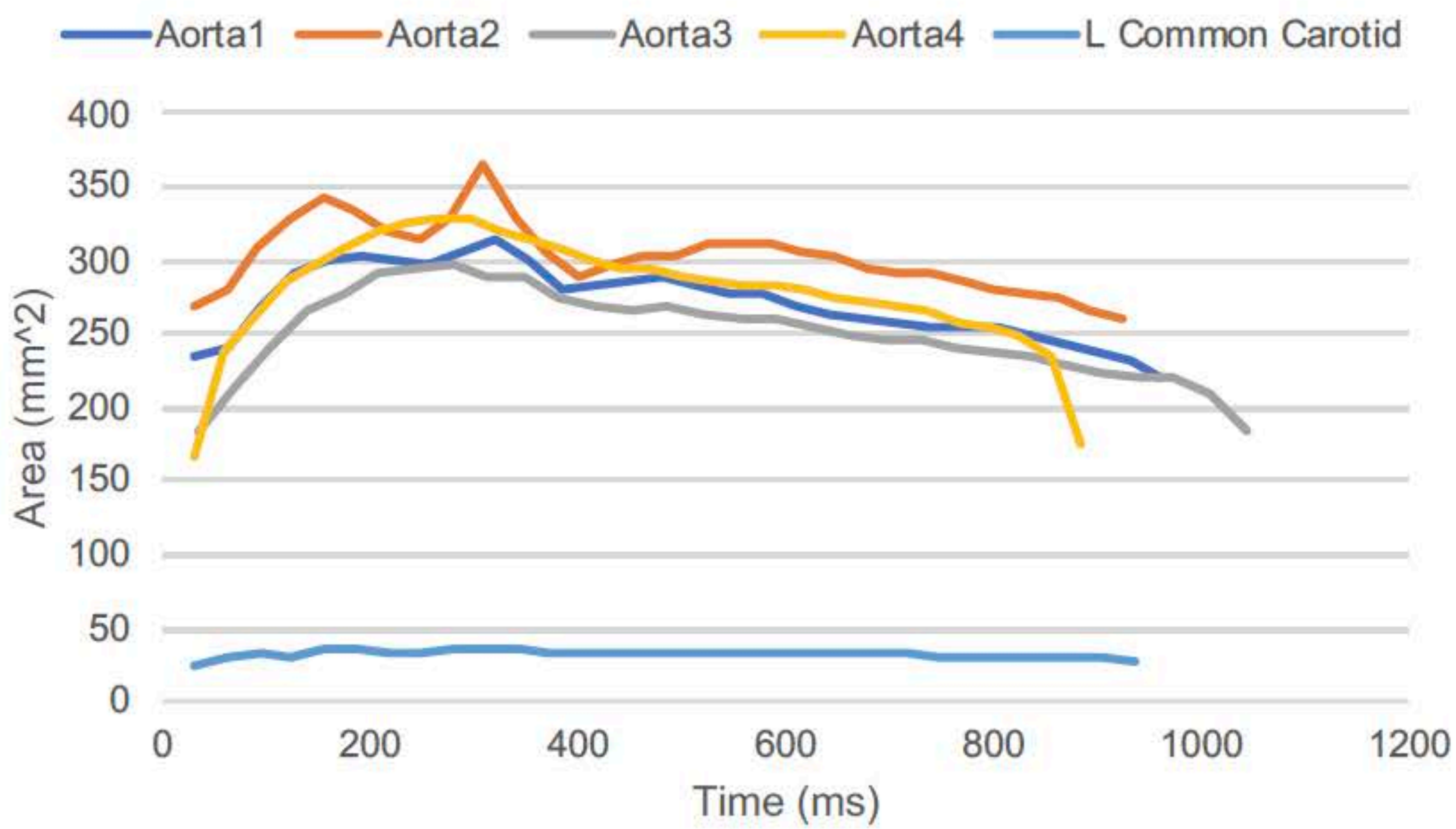
4D flow MRI



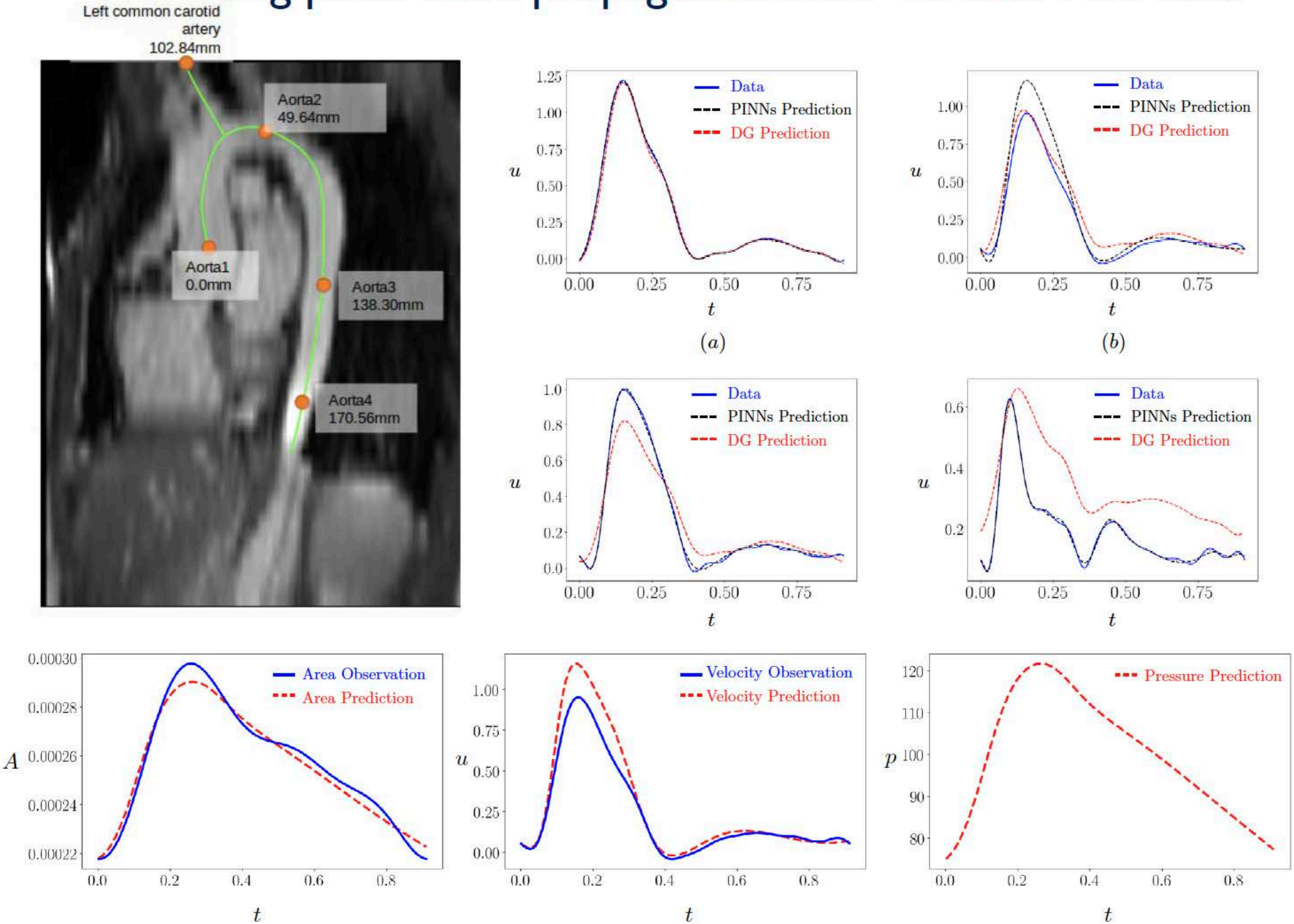
2D Cine & Phase contrast images







Predicting pulse wave propagation from 4D flow MRI data



Acknowledgements:

George Kissas
(Penn)

Yibo Yang
(Penn)

Eileen Hwuang
(Penn)

Walter Witschey
(Penn)

John Detre
(Penn)

Hands-on tutorial:

https://github.com/PredictiveIntelligenceLab/USNCCM15-Short-Course-Recent-Advances-in-Physics-Informed-Deep-Learning/blob/master/notebooks/Graph%20PINNs.ipynb

Yang, Y., & Perdikaris, P. (2018). Physics-informed deep generative models. Neural Information Processing Systems, Workshop on Bayesian Deep Learning.

Yang, Y., & Perdikaris, P. (2019). Adversarial uncertainty quantification in physics-informed neural networks. Journal of Computational Physics.

Code: https://github.com/PredictiveIntelligenceLab/UQPINNs