PNAS
www.pnas.org

# Supplementary Information for

## Artificial intelligence velocimetry and microaneurysm-on-a-chip for three-dimensional analysis of blood flow in physiology and disease

**Shengze Cai, He Li, Fuyin Zheng, Fang Kong, Ming Dao, George Em Karniadakis and Subra Suresh**

**Corresponding authors: Ming Dao, George Em Karniadakis, Subra Suresh**
**E-mail: mingdao@mit.edu, george_karniadakis@brown.edu, ssuresh@ntu.edu.sg**

**This PDF file includes:**

Supplementary text
Figs. S1 to S5 (not allowed for Brief Reports)
Tables S1 to S4 (not allowed for Brief Reports) SI
References

**Other supplementary materials for this manuscript include the following:**

Movies S1 and S2

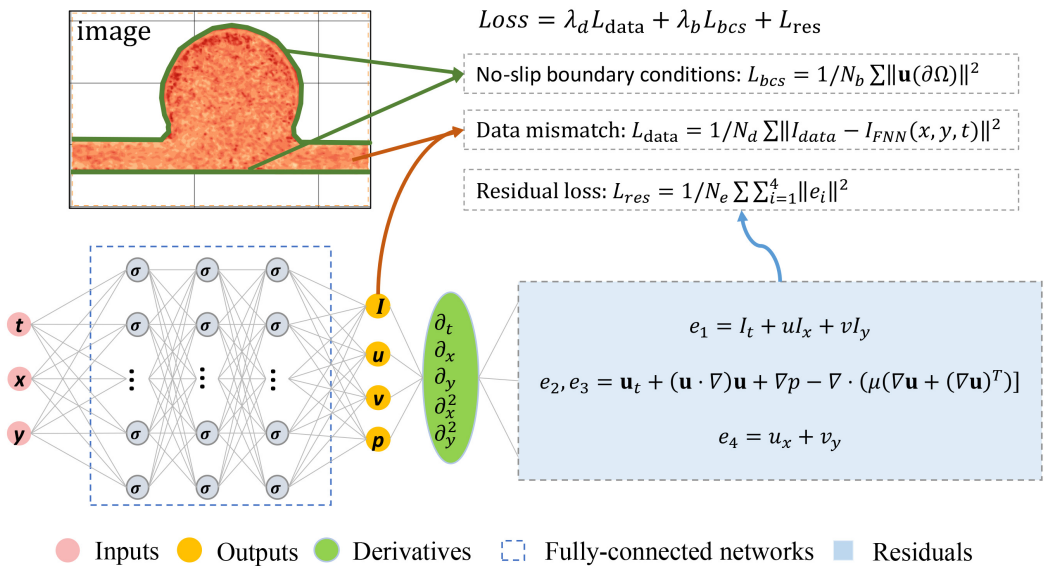


**Fig. S1.** Schematic diagram of 2D AIV model. A fully-connected network is used to approximate the key output parameters, namely $(I, u, v, p)$, by taking the space and time coordinates as inputs $(x, y, t)$. The governing equations for this problem are encoded in a residual network, where the derivatives are computed via automatic differentiation (AD) in the TensorFlow code (Google, Mountain View, CA, USA). Moreover, the no-slip boundary conditions are introduced on the channel walls (denoted by green lines at the top-left image), namely $\mathbf{u}(\partial\Omega) = 0$. The activation function for each neuron is $\sigma(\cdot) = \sin(\cdot)$. The loss function for training is composed of three terms: data mismatch, boundary conditions and residuals of conservation laws.

**Table S1. Number of training data points used for training of 2D AIV. $N_d$ represents the number of training data from microfluidic images. $N_e$ denotes the number of training points for computing the residues of the governing equations. $N_b$ designates the number of training points on the channel wall.**

| MA | MA#3 | MA#4 | MA#6 |
|---|---|---|---|
| $N_d$ | 7410 | 10353 | 14983 |
| $N_b$ | 510 | 522 | 587 |
| $N_e$ | 7410 | 10353 | 14983 |

## Supporting Information Text

### 1. Implementation details of AIV

**2D AIV model.** The details of the AIV model employed to infer the 2D velocity and pressure fields from 2D images is elaborated in Fig. S1. The inputs (purple circles) to the model are the time and 2D spatial coordinates $(t, x, y)$ and the outputs of the model (orange circles) are spatio-temporal scalar field $I$ which represents the image intensity as well as the 2D velocity and pressure field $(u, v, p)$. The encoded equations in the AIV model (in the blue box) include the image advection equation, the 2D Navier-Stokes equation and continuity equation.

***Training data.*** The training data sets for AIV include (1) the data points extracted from the microfluidic images $\{\mathbf{x}_i, t_i, I_i\}_{i=1}^{N_d \times N_t}$ for computing $\mathcal{L}_{data}$; (2) the residual points $\{\mathbf{x}_i, t_i\}_{i=1}^{N_e \times N_t}$ used in the governing equations for computing $\mathcal{L}_{res}$; and (3) the points selected on the boundaries of the channel wall $\{\mathbf{x}_i, t_i\}_{i=1}^{N_b \times N_t}$ for computing $\mathcal{L}_{bcs}$, where $N_t$ denotes the number of image frames. The number of data points $N_d$ and the number of points on the wall boundary $N_b$ are selected based on the number of pixels in the microfluidic images, while $N_e$ is selected to be $N_e = N_d$ for the prediction of 2D velocity and pressure fields. The values of $N_d$, $N_b$ and $N_e$ used for the three MAOAC channels with different geometries are listed in Table S1. 100 sequential images (i.e., $N_t = 100$) are extracted from the blood flow videos for each of these three MAOAC channels to train AIV.

***Hyperparameters.*** As illustrated in Fig. S1, the AIV model is composed of a fully-connected network containing 10 hidden layers with 80 neurons per layer. For each layer, the input-output relationship can be expressed as: $Y = \sigma(WX + b)$, where $W$ and $b$ are the trainable weights and biases, respectively. The activation function for each neuron is $\sigma(\cdot) = \sin(\cdot)$. All the weights and biases are randomly initialized by the Xavier scheme (1). The training of the network is performed using *Adam* optimizer (2). In order to calculate the residuals ($e_1$,$e_2$,$e_3$ and $e_4$ in Fig. S1) in $\mathcal{L}_{res}$, the partial differential operators in the governing equations are computed using automatic differentiation (AD) where the derivatives in the governing equations are

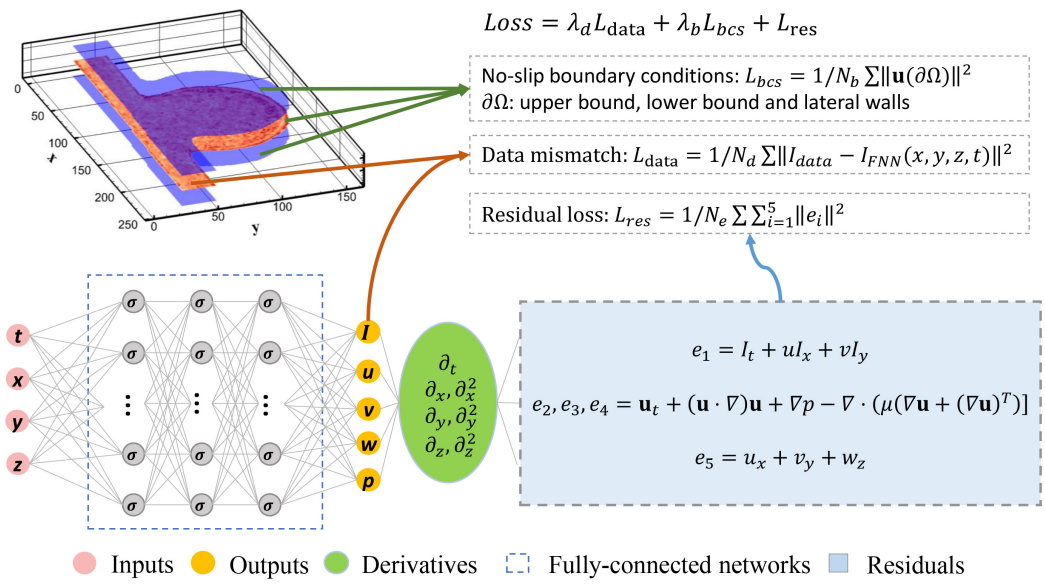


**Fig. S2.** Schematic diagram of 3D AIV model. A fully-connected network is used to approximate the solutions, namely $(I, u, v, w, p)$, by taking the space and time coordinates as inputs $(x, y, z, t)$. The governing equations include the transport equation of image intensity, the three-component momentum equations and the continuity equation.

approximated by the derivatives of the output with respect to the input of the neural networks. $\lambda_d$ and $\lambda_b$ are the weighing coefficients to account for the relative contributions of the three terms in the total loss function. We note that for AIV, a relatively large $\lambda_d$ and $\lambda_b$ can accelerate optimization because the image data can be quickly regressed, but it may also result in overfitting. More details of how to select the appropriate weighing coefficients in the loss function for training PINNs can be found in (3, 4). In this work, the weighting coefficient for $L_{data}$ is selected to be $\lambda_d = 10$ through a trial-and-run process. As we impose no-slip boundary conditions on the channel wall, we implement a stronger penalization ($\lambda_b = 100$) on loss resulting from the boundary conditions ($L_{bcs}$) to constrain the magnitude of the velocity on the channel wall to be approximately zero.

***Training strategy.*** The training of AIV are performed using a two-step optimization. In the first step, we train the network with 5,000 epochs with initial learning rate $10^{-3}$ and we observe that the total loss function has reached a plateau. One epoch in the training means that all the selected training points ($N_d + N_b + N_c$ in Table S1) have been used to train the neural network once. In this step, $L_{data}$ are predominately minimized through the optimization process such that the AIV model can attain a good regression to the microfluidic image data. Next, we exclude the data term ($L_{data}$) as well as the image advection equation (residual $e_1$) in the loss function, and train the neural network for another 50000 epochs with a smaller learning rate ($10^{-4}$) to fine-tune the velocity and pressure fields based on the residuals of Navier–Stokes (NS) equations.

**3D AIV model.** The 3D AIV model, as shown in Fig. S2, is composed of a neural network containing 10 layers with 100 neurons per layer. The training points for the residual loss ($L_{res}$) are selected uniformly in the 3D computational domain as illustrated in Fig.7(A) in the maintext. We select the number of training points for computing the residues of the governing equations, $N_e = 2N_d$, where $N_d$ is the number of 2D image pixels. The number of training points on the channel wall $N_b$ is selected to be 50000 for the 3D case and they are uniformly distributed on the upper bound, lower bound and lateral channel walls. We train the 3D AIV following the same strategy as introduced in the 2D AIV model.

## 2. Methods for comparisons

**Variational optical flow.** Variational optical flow is one of the most popular techniques in the computer vision community for estimation of the movement of objects in a sequence of images (5–7). The estimation of flow field is performed by minimizing an objective function which is expressed as

$$\min_{\mathbf{u}} J = \int_{\Omega} \left[ (\partial_t I + \nabla I \cdot \mathbf{u})^2 + \alpha \parallel \nabla \mathbf{u} \parallel^2 \right] d\Omega. \quad [1]$$

The first term in the function considers the data extracted from the sequential images under the assumption that the targeted objects have the fixed brightness in the sequential images. The second term is a regularization term for smoothing the velocity field. The optical flow-based methods have been developed for fluid flow motion estimation, especially for PIV (8–10). In this work, we implement the multi-resolution algorithm (10) to solve Eq. 1. The algorithm and the corresponding code can be found at: https://github.com/shengzesnail/coarse_to_fine_HS_PIV. The weighting coefficient $\alpha$ is 100 and the result is obtained by taking the average of 100 instances.

**Deep-PIV.** Deep-PIV, designed for particle image velocimetry, is a deep convolutional neural network with an encoder-decoder architecture (11, 12). The original Deep-PIV model was trained using thousands of particle images and the estimation of the flow field relies on the correlation of the particle positions at two consecutive images (11). In the current work, we employ the pretrained Deep-PIV model in (11) to predict the velocity field in MAOAC channels without further training because the motion of blood cells in the blood flow is analogous to particle motion. The algorithm and the corresponding code can be found at: https://github.com/shengzesnail/PIV-LiteFlowNet-en. The result of Deep-PIV is obtained by taking the average of 100 instances.

## 3. Modeling the transport of RBCs and platelets in the microchannel using dissipative particle dynamics

We perform dissipative particle dynamics (DPD) simulations to model the transport of red blood cells (RBCs) and platelets in channel MA#3 (BNR = 3.6), as shown in Fig. 6 in the main paper. The DPD method is a mesoscopic particle-based simulation technique, where each DPD particle represents a cluster of molecules and it interacts with other DPD particles through soft pairwise forces (13, 14). DPD method is capable of capturing the hydrodynamic behavior of fluids at the mesoscale, and it has been successfully employed to study complex fluids, such as blood (15, 16). The equation of motion for each DPD particle $i$ is governed by the sum of pair interactions $\mathbf{f}_i$ with the surrounding particles $j$. The time evolution of velocity ($\mathbf{v}_i$) and position ($\mathbf{r}_i$) of a particle $i$ with mass $m_i$ is determined by Newton's second law of motion:

$$\mathrm{d}\mathbf{r}_i = \mathbf{v}_i\,\mathrm{d}t\,; \quad \mathrm{d}\mathbf{v}_i = \mathbf{f}_i/m_i\,\mathrm{d}t. \tag{2}$$

where $dt$ is the simulation timestep size. These two equations of motion of DPD particles are solved using a velocity-Verlet algorithm. In the classical DPD method (13, 14), the total force $\mathbf{f}_i$ exerted on particle $i$ by particle $j$ is composed of a conservative force ($\mathbf{F}^C_{ij}$), a dissipative force ($\mathbf{F}^D_{ij}$), and a random force ($\mathbf{F}^R_{ij}$) given by

$$\mathbf{F}^C_{ij} = a_{ij}(1-\frac{r_{ij}}{r_c})\hat{\mathbf{r}}_{ij} \quad for \quad r_{ij} \le r_c; \quad 0 \quad for \quad r_{ij} > r_c, \tag{3}$$

$$\mathbf{F}^D_{ij} = \gamma\omega_d(r_{ij})(\hat{\mathbf{r}}_{ij}\cdot\hat{\mathbf{v}}_{ij})\hat{\mathbf{r}}_{ij}, \tag{4}$$

$$\mathbf{F}^R_{ij} = \sigma\omega_r(r_{ij})\frac{\zeta_{ij}}{\sqrt{dt}}\hat{\mathbf{r}}_{ij}, \tag{5}$$

where $r_c$ is a cut-off distance beyond which the interaction force is considered to be zero. $a_{ij}$, $\gamma$, $\sigma$ are the conservative, dissipative and random coefficients, respectively. The DPD parameters used in Eqs. (3)-(5) for all types of DPD particles are given in Table.S2. $r_{ij}$ is the distance between particle $i$ and $j$. $\hat{\mathbf{r}}_{ij}$ is a unit vector. $\hat{\mathbf{v}}_{ij}$ is the difference between the $\mathbf{v}_i$ and $\mathbf{v}_j$. $\zeta_{ij}$ is a Gaussian random number with zero mean and unit variance. The parameters $\gamma$ and $\sigma$ and the weight functions are coupled through the fluctuation-dissipation theorem and are related by $\omega_d = \omega_r^2$ and $\sigma^2 = 2\gamma k_B T$, where $k_B$ is the Boltzmann constant and $T$ is the temperature of the system. The weight function $\omega_r(r_{ij}) = (1-r_{ij}/r_c)^k$ with $k = 1$ in the standard DPD method (13, 14), whereas other values of $k$ have been used to increase the fluid viscosity (15, 17, 18). More detailed description of DPD method can be found in (13, 19).

In addition to blood plasma modeled by collections of free DPD particles, the membrane of blood cells suspended in the plasma, including RBCs and platelets, is constructed by a 2D triangulated network with $N_v$ vertices (DPD particles). The vertices are connected by $N_s$ elastic bonds to impose proper membrane mechanics. These DPD representations of RBCs and platelets were validated and widely used in the previous studies for both healthy and diseased cells (18, 20–22). The free energy $V_{cell}$ of a single cell is given by

$$V_{cell} = V_s + V_b + V_{a+v}. \tag{6}$$

The elastic energy $V_s$ representing the elastic interactions of the cell membrane is defined by

$$V_s = \sum_{j\in 1\ldots N_s}\left[\frac{k_{\mathrm{B}}Tl_m(3x_j^2-2x_j^3)}{4p(1-x_j)} + \frac{k_p}{l_j}\right], \tag{7}$$

where $p$ is the persistence length, $k_p$ is the spring constant, $k_{\mathrm{B}}T$ is the energy unit, $l_j$ is the length of the spring $j$, $l_m$ is the maximum spring extension, and $x_j = l_j/l_m$. $p$ and $k_p$ are computed by balancing the forces at equilibrium and from their relation to the macroscopic shear modulus, $\mu_s$:

$$\mu_s = \frac{\sqrt{3}k_{\mathrm{B}}T}{4pl_mx_0}\left(\frac{x_0}{2(1-x_0)^3} - \frac{1}{4(1-x_0)^2} + \frac{1}{4}\right) + \frac{3\sqrt{3}k_p}{4l_0^3}, \tag{8}$$

where $l_0$ is the equilibrium spring length and $x_0 = l_0/l_m$. The bending resistance $V_b$ of the cell membrane is modeled by

$$V_b = \sum_{j\in 1\ldots N_s} k_b\left[1-cos(\theta_j-\theta_0)\right], \tag{9}$$

where $k_b$ is the bending constant, and it is related to the macroscopic bending rigidity $k_c$ where $k_b = 2k_c/\sqrt{3}$, $\theta_j$ is the instantaneous angle between two adjacent triangles having the common edge $j$, and $\theta_0$ is the spontaneous angle. In addition,

**Table S2. DPD parameters used in simulations.** $r_c$ **is a cut-off radius,** $a_{ij}$ **is the conservative coefficient,** $\gamma$ **is the dissipative coefficient, and** $k$ **is the weight function exponent. In all simulations, we set the particle mass** $m = 1$**, and the thermal energy** $k_BT = 0.10$ **in DPD units. Note that S = solvent (representing plasma), R = RBC and P = platelet.**

| type | $r_c$ | $a_{ij}$ | $\gamma$ | $k$ |
|---|---|---|---|---|
| S-S | 1.58 | 5.0 | 20.0 | 0.20 |
| S-R | 1.5 | 0.0 | 45.0 | 0.20 |
| S-P | 1.5 | 0.0 | 10.0 | 0.20 |
| R-R | 1.0 | 10.0 | 10.0 | 0.20 |
| R-P | 1.0 | 10.0 | 10.0 | 0.20 |
| P-P | 1.0 | 10.0 | 10.0 | 0.20 |

**Table S3. Cell membrane parameters for RBC and platelet models.** $N_v$ **is the number of DPD particles on the membrane,** $l_m$ **is the maximum bond extension,** $l_0$ **is the equilibrium bond length,** $k_b$ **is the bending constant,** $\mu_s$ **is the shear modulus,** $A_0^{\mathbf{tot}}$ **and** $V_0^{\mathbf{tot}}$ **are the specified cell area and volume, respectively,** $k_d + k_a$ **is the combined area constraint coefficient, and** $k_v$ **is the volume constraint coefficient.**

| cell | $N_v$ | $l_m/l_0$ | $k_b$ | $\mu_s$ | $A_0^{\text{tot}}$ $(V_0^{\text{tot}})$ | $k_d + k_a$ $(k_v)$ |
|---|---|---|---|---|---|---|
| RBC | 500 | 1.8 | 6.025 | 100.0 | 132.87 (92.45) | 5000 (5000) |
| PLT | 48 | 1.8 | 100.0 | $10^4$ | 19.63 (6.02) | 5000 ($10^4$) |

the area and volume constraints $V_{a+v}$ are imposed to mimic the area-preserving lipid bilayer and the incompressible interior fluid. The corresponding energy is given by

$$V_{a+v} = \sum_{j\in 1\dots N_t} \frac{k_d(A_j - A_0)^2}{2A_0} + \frac{k_a(A_{\text{cell}} - A_0^{\text{tot}})^2}{2A_0^{\text{tot}}} + \frac{k_v(V_{\text{cell}} - V_0^{\text{tot}})^2}{2V_0^{\text{tot}}}, \quad [10]$$

where $N_t$ is the number of triangles in the membrane network, $A_0$ is the equilibrium value of a triangle area, and $k_d$, $k_a$ and $k_v$ are the local area, global area and volume constraint coefficients, respectively. The terms $A_0^{\text{tot}}$ and $V_0^{\text{tot}}$ are targeted cell area and volume. The cell membrane parameters used in Eqs (7)-(10) for all blood cell models are given in Tables S2 and S3, respectively.

In order to prevent cell overlap, we employ a Morse potential between membrane particles from different blood cells and it is expressed as

$$V_M(r) = D_e[e^{2\beta(r_0-r)} - 2e^{\beta(r_0-r)}], \quad [11]$$

where $r$ and $r_0$ are the distance and equilibrium distance between two membrane particles, respectively. $D_e$ is the well-depth of the potential, and $\beta$ characterizes the interaction range. By properly setting the parameters, we can ensure sufficiently strong repulsive forces between cell membrane particles and prevent their overlap. We present the Morse potential parameters used for cell-cell interactions in Table S4. Note that the cutoff radius $r_{c,morse} = 1$ for all the Morse interactions is given.

To drive blood flow in the microchannel, we apply a constant body force to each DPD particle and we tune this body force such that the velocity of the solvent particles around the centerline of the post-aneurysm channel, as shown in Fig.6(C) in the maintext, is close to the maximum velocity of 1.65 mm/s in the channel inferred by AIV.

In order to convert the DPD units to the physical units, we define the length scale of the simulation as $[L] = 1 \times 10^{-6}m$. Then, the time scale of the system can be evaluated by

$$[t] = [L]\frac{\eta^P}{\eta^M}\frac{\mu_s^M}{\mu_s^P} \quad [12]$$

where $\mu_s$ is the RBC membrane shear modulus, $\eta$ is the plasma viscosity, and superscripts M and P denote the model (DPD) and physical units, respectively. Using membrane shear modulus of healthy RBCs $\mu_s^p = 4.73 \times 10^{-6}N/m$ and plasma viscosity $\eta^P = 1.2 \times 10^3 Pas$, the length scale is computed to be $[t] = 2.27 \times 10^{-4}s$.

## 4. 3D results of AIV for microchannel MA#6

The AIV prediction and the CFD simulation results of shear stress for MA#6 (BNR = 5) along two lines on two different planes in the channel depth direction are shown in Fig. S3.

**Table S4. Morse potential parameters for cell-cell interactions.** $D_e$ **is the well depth of the potential,** $r_0$ **is the zero force distance, and** $\beta$ **characterizes the interaction range. Note that R = RBC and P = platelet**

| type | $D_e$ | $\beta$ | $r_0$ |
|---|---|---|---|
| R-R | 5.0 | 2.0 | 0.95 |
| R-P | 10.0 | 2.0 | 1.0 |
| P-P | 10.0 | 2.0 | 1.0 |

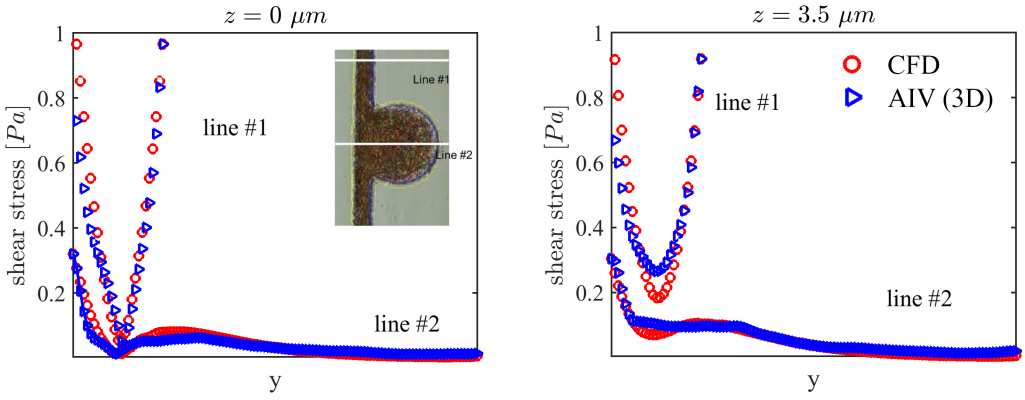


**Fig. S3.** Comparison between the AIV prediction and the CFD simulation of wall shear stress for MA#6 (BNR = 5) along two lines on two different planes in the channel depth direction. Left: z = 0 $\mu m$, right: z = 3.5 $\mu m$. Red circles: CFD results, blue triangles: AIV results.

## 5. 3D results of AIV for microchannel MA#3

We also employ 3D AIV model to predict the velocity field, pressure field and wall shear stress for MAOAC channel with BNR = 2 and the corresponding 3D results are illustrated in Fig. S4 and Fig. S5.

## 6. List of supplementary movies

**Movie S1.** A bright-field micro-aneurysm-on-a-chip experimental video segment, from which the velocity and pressure fields of blood flow were inferred by the AIV model. The original images were recorded with a frame rate of 500 fps, while the video here has been slowed down to 12.5 fps.

**Movie S2.** A fluorescence-stained micro-aneurysm-on-a-chip experimental video segment, from which we performed platelet-tracking to obtain the local velocities of the flow. The original images were recorded with a frame rate of 60 fps, while the video here has been slowed down to 12.5 fps.

## References


1. X Glorot, Y Bengio, Understanding the difficulty of training deep feedforward neural networks in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. pp. 249–256 (2010).
2. DP Kingma, J Ba, Adam: A method for stochastic optimization. (30 Jan 2017).
3. X Jin, S Cai, H Li, GE Karniadakis, NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **426**, 109951 (2021).
4. S Wang, Y Teng, P Perdikaris, Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv preprint arXiv:2001.04536 (13 January 2020).
5. B Horn, B Schunck, Determining optical flow. *Artificial Intelligence* **17**, 185–203 (1981).
6. G Aubert, R Deriche, P Kornprobst, Computing optical flow via variational techniques. *SIAM J. on Appl. Math.* **60**, 156–182 (1999).
7. A Bruhn, J Weickert, C Feddern, T Kohlberger, C Schnorr, Variational optical flow computation in real time. *IEEE Transactions on Image Process.* **14**, 608–615 (2005).
8. T Corpetti, D Heitz, G Arroyo, E Mémin, A Santa-Cruz, Fluid experimental flow estimation based on an optical-flow scheme. *Exp. Fluids* **40**, 80–97 (2006).
9. T Liu, L Shen, Fluid flow and optical flow. *J. Fluid Mech.* **614**, 253–291 (2008).
10. D Heitz, E Mémin, C Schnörr, Variational fluid flow measurements from image sequences: synopsis and perspectives. *Exp. Fluids* **48**, 369–393 (2010).

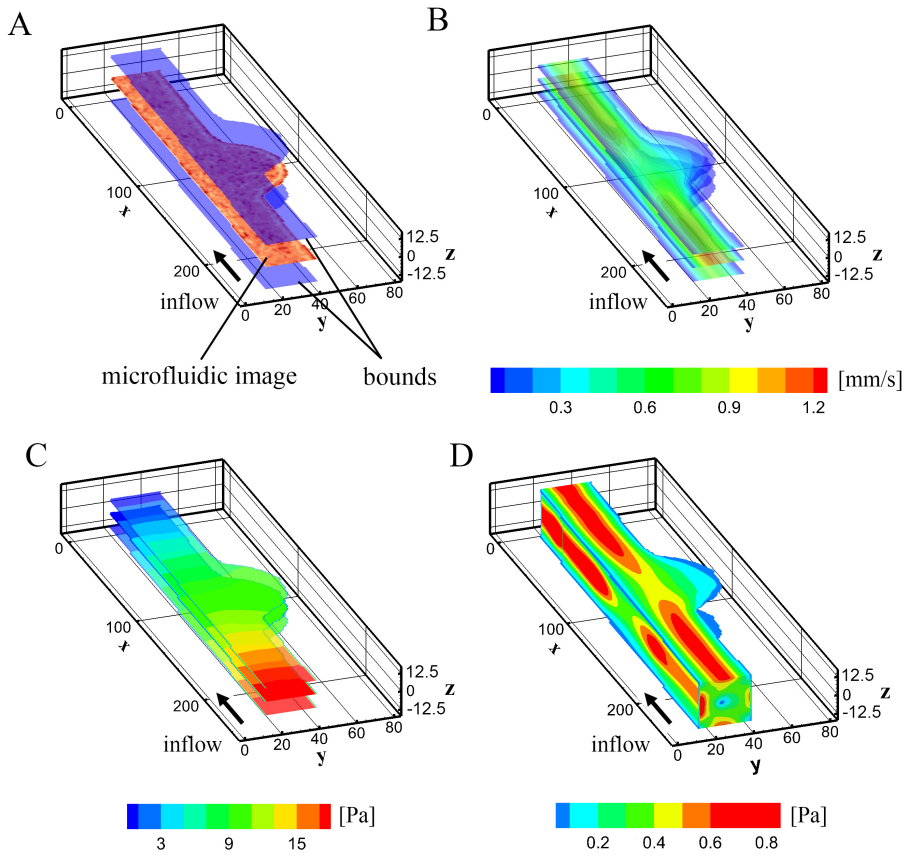


**Fig. S4.** 3D AIV predictions for MAOAC channel with BNR = 2. (A) A 3D computational domain is constructed by extruding the 2D domain along the depth direction ($z$) by 25 $\mu m$ ($z \in [-12.5, 12.5]$ $\mu m$), consistent with the depth of the MAOAC channels. We assume that the microfluidic images capture the motion of blood cells at the middle plane in the channel depth direction ($z = 0$). (B) Velocity magnitudes and (C) pressure fields at three different cross-sections ($z = 0, \pm 7.5$ $\mu m$) along the depth of the channel. (D) Shear stress on the channel wall. AIV results are averaged over 100 image frames.


11. S Cai, J Liang, Q Gao, C Xu, R Wei, Particle image velocimetry based on a deep learning motion estimator. *IEEE Transactions on Instrumentation Meas.* **69**, 3538–3554 (2020).
12. S Cai, et al., Deep-PIV: a new framework of PIV using deep learning techniques in *13th Int. Symp. on Particle Image Velocimetry, Munich, Germany.* (2019).
13. P Espanol, P Warren, Statistical mechanics of dissipative particle dynamics. *Eur. Lett.* **30**, 191 (1995).
14. RD Groot, PB Warren, Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The J. Chem. Phys.* **107**, 4423–4435 (1997).
15. DA Fedosov, W Pan, B Caswell, G Gompper, GE Karniadakis, Predicting human blood viscosity in silico. *Proc. Natl. Acad. Sci. U.S.A* **108**, 11772–11777 (2011).
16. T Ye, N Phan-Thien, CT Lim, Particle-based simulations of red blood cells - a review. *J. Biomech.* **49**, 2255–2266 (2016).
17. X Fan, N Phan-Thien, S Chen, X Wu, T Yong Ng, Simulating flow of dna suspension using dissipative particle dynamics. *Phys. Fluids* **18**, 063102 (2006).
18. DA Fedosov, B Caswell, GE Karniadakis, A multiscale red blood cell model with accurate mechanics, rheology, and dynamics. *Biophys. J.* **98**, 2215–2225 (2010).
19. RD Groot, PB Warren, Dissipative particle dynamics: bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.* **107**, 4423–4435 (1997).
20. IV Pivkin, GE Karniadakis, Accurate coarse-grained modeling of red blood cells. *Phys. Rev. Lett.* **101**, 118105 (2008).
21. A Yazdani, GE Karniadakis, Sub-cellular modeling of platelet transport in blood flow through microchannels with constriction. *Soft Matter* **12**, 4339–4351 (2016).
22. H Lei, GE Karniadakis, Probing vasoocclusion phenomena in sickle cell anemia via mesoscopic simulations. *Proc. Natl. Acad. Sci. U.S.A* **110**, 11326–11330 (2013).

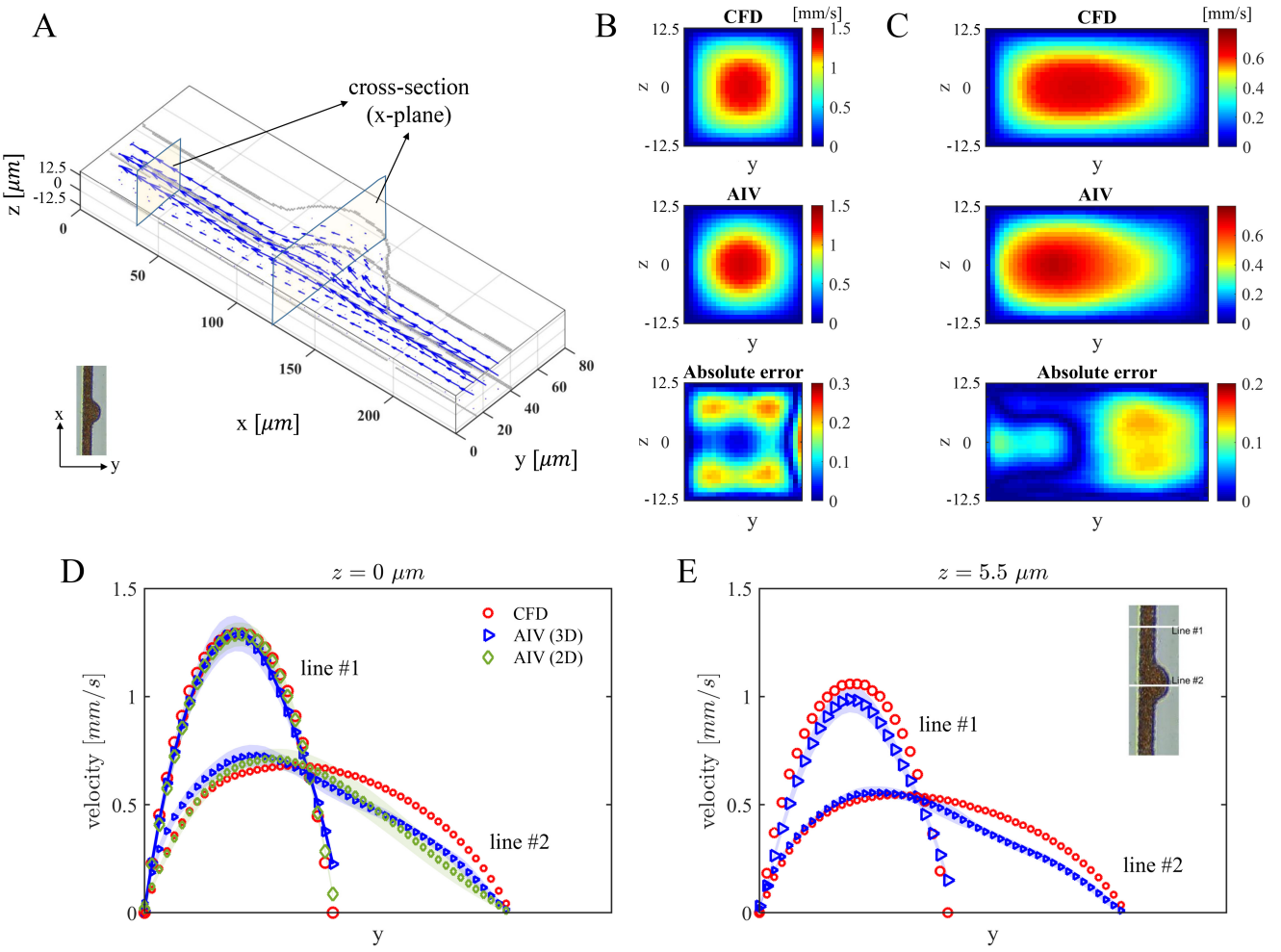


**Fig. S5.** Comparison of 3D AIV predictions with results of CFD simulations for MAOAC channel with BNR = 2. (A) 3D velocity vectors inferred from AIV. Two cross-sections along the x-axis are selected to make comparisons with CFD simulations. (B) Velocity magnitudes at the cross-section located at $x = 35$ $\mu$m in the microchannel. From top to bottom: CFD simulation, AIV model, and absolute error. The relative $L_2$-norm error at this cross-plane is $14.23\%$. (C) Velocity magnitudes at the cross-section located at $x = 130$ $\mu$m in the microchannel. From top to bottom: CFD simulation, AIV model, and absolute error. The relative $L_2$-norm error is $16.49\%$. (D)-(E) Velocity profiles along two cross lines at planes $z = 0$ $\mu$m and $z = 5.5$ $\mu$m. The symbols of AIV predictions signify the time-averaged values from 100 image frames and the shadows represent the standard deviations.