Supplementary informations for: Inferring biophysical properties of membrane during endocytosis using machine learning

I. BOUNDARY CONDITIONS AND COORDINATES CONSTRAINT

The two variational equations eqr and eqz are fourth-order ordinary differential equations. The solutions to these two equations are not unique because there is a freedom to choose the scaled factor. To see this, note that $[r(u), z(u)]$ and $[r(u^2), z(u^2)]$ give the same membrane shape when u is varied from 0 to 1. In fact, u^2 in the bracket can be replaced with any monotonous function $g(u)$ that maps the interval $[0, 1]$ to itself. To avoid the non-uniqueness problem, we add a coordinates constraint

$$a' = 0 \quad \text{for} \quad u \in [0, 1]. \tag{1}$$

With this constraint, the parameter u is essentially the rescaled arclength and the value of a is the total arclength of the membrane profile.

As for the boundary conditions, we have a total of 8 equations:

$$\begin{cases} r(0) &= 0 \\ r(1) &= R_b \\ r''(0) &= 0 \\ r''(1) &= 0 \\ z(0) &= z_0 \\ z(1) &= 0 \\ z'(0) &= 0 \\ z'(1) &= 0 \end{cases}. \tag{2}$$

The first line is due to axisymmetry of the membrane shape. Note that for the purpose of computation in a computer, $r(0) = 0$ introduces a numerical singularity as it appears in the denominator of the curvature expression H of $2H = -\frac{1}{a}\left(\frac{b}{a^2} + \frac{z'}{r}\right)$,. In practice, we choose $r(0) = 0.001$, which is a very small number such that it would not influence the accuracy of the solution at the rest of the points. The second line sets the base radius R_b, which would be learned by the neuron network in the inverse problem. The third line is also due to axisymmetry of the membrane shape. The fifth and the sixth lines set the membrane heights. In the inverse problem, we set z_0 to be the same as the experimental profile. The seventh and eighth lines set the angle at the tip and at the base to be zero. As for the fourth line, it arises from the coordinate constraint (1) expressed at the boundary

$$a'(1) \propto r'(1)r''(1) + z'(1)z''(1) = r'(1)r''(1) = 0. \tag{3}$$

The fact that $r'(1)$ is not equal to zero leads to $r''(1) = 0$.

II. LOSS FUNCTION OF THE PINN

A. Forward problem

In order to solve ordinary differential equations with PINN, we need to specify the loss function of the neuron network. The forward problem is equivalent to solve the variational equations eqr and eqz with the coordinates constraint (1) and the boundary conditions (2), provided the values of all the parameters are given. The loss function of the forward problem L_for therefore can be constructed as the sum of three terms

$$L_\text{for} = L_\text{eqs} + L_\text{con} + L_\text{bc}, \tag{4}$$

with L_eqs from the two variational equations, L_con from the coordinates constraint, and L_bc from the boundary conditions. Their explicit expressions are given below:

$$L_\text{eqs} = \frac{1}{N+1}\sum_{i=1}^{N-1} eqr(i)^2 + \frac{1}{N+1}\sum_{i=1}^{N-1} eqz(i)^2, \tag{5}$$

$$L_\text{con} = \frac{1}{N+1}\sum_{i=1}^{N-1} a'(i)^2, \tag{6}$$

$$\begin{aligned} L_\text{bc} &= [r(0) - 0.001]^2 + [r(1) - R_b]^2 + r''(0)^2 + r''(1)^2 \\ &+ [z(0) - z_0]^2 + z'(0)^2 + z'(1)^2 + z(0)^2, \end{aligned} \tag{7}$$

where $f(i)$ represents the value of the function f evaluated at $u = \frac{i}{N}$. When the expression involves derivatives, we use automatic differentiation function provided by tensorflow. We find that $N = 16$ is able to solve the equations to a good accuracy in a reasonable time.

We are now in a position to train the neuron network to minimize the loss function L_for. Based on our experiences, we find that applying a so-called Hard Constraints Method [1, 2] has a higher prediction accuracy than directly minimizing the loss function L_for (Eq. 4). The idea of the Hard Constraints Method is to introduce a transformation $r(u) \to \widetilde{r}(u)$, and $z(u) \to \widetilde{z}(u)$ such that the transformed $\widetilde{r}(u)$ and $\widetilde{z}(u)$ automatically satisfy the boundary conditions. Specifically, note that there are 4 boundary conditions for each neural network output $r(u)$ and $z(u)$, we introduce the following transformations

$$\widetilde{r}(u) = r(u) + a_1 + b_1 u + c_1 u^2 + d_1 u^3 \tag{8}$$

and

$$\widetilde{z}(u) = z(u) + a_2 + b_2 u + c_2 u^2 + d_2 u^3, \tag{9}$$

and the coefficients in the polynomials can be solved by requiring $\widetilde{r}(u)$ and $\widetilde{z}(u)$ to fullfil the boundary conditions.

After the transformation, we can remove the boundary condition term $L_{\rm bc}$ in the loss function $L_{\rm for}$ to get a simpler loss function

$$L_{\rm for} = \frac{1}{N+1}\sum_{i=1}^{N-1}\widetilde{eqr}(i)^2 + \frac{1}{N+1}\sum_{i=1}^{N-1}\widetilde{eqz}(i)^2 + \frac{1}{N+1}\sum_{i=1}^{N-1}\widetilde{a}'(i)^2, \quad (10)$$

where $\widetilde{eqr}$, $\widetilde{eqz}$ and $\widetilde{a}'$ become equations with $\widetilde{r}$, $\widetilde{z}$ and their derivatives.

B. Inverse problem

Given an experimental profile which is a number of discrete points $[r_i, z_i]$ that depicts the membrane shape, we first perform a symmetrization procedure to obtain a function $[r_{\rm exp}(u), z_{\rm exp}(u)]$ that has the mirror symmetry, which is explained in Supplementary Information III. The inverse problem aims to find model parameters such that the transformed ML output $[\tilde{r}(u), \tilde{z}(u)]$ not only satisfies the variational equations, but also matches the symmetrized experimental profile . The loss function of the inverse problem therefore needs to incorporate $L_{\rm data}$ which measures the difference between the ML output and the symmetrized experimental profile. Specifically, we do interpolation of $R(Z_j)$ at equally spaced $Z_j = z_0 j/M, j = 0, 1, ..., M$ for both the ML output $[\tilde{r}(u), \tilde{z}(u)]$ and the symmetrized experimental profile $[r_{\rm exp}(u), z_{\rm exp}(u)]$. The loss function $L_{\rm data}$ then takes the squared distance between the two interpolated datasets, i.e.,

$$L_{\rm data} = \sum_{j=0}^{M}\left[\tilde{R}(Z_j) - R_{\rm exp}(Z_j)\right]^2, \quad (11)$$

in which $\tilde{R}(Z_j)$ and $R_{\rm exp}(Z_j)$ represent the interpolation results of the ML output and the symmetrized experimental profile at the same point Z_j, respectively.

III. SYMMETRIZATION ALGORITHM

In order to learn the model parameters, we need to compare the ML output with the experimental data, which is structured in 2D coordinates (x_i, y_i) that gives a single 2D cross-section of the 3D membrane. One example of the 2D cross-section curves is shown in Fig. S1. As the cell membrane is a 3D structure and only a 2D cross-sectional electron microscope image cannot restore entire asymmetric membrane shape, a symmetrization procedure is applied on the experimental profile to be compared with the ML output.

The procedure is conducted in 3 steps: (1) pick up the point (x_0, y_0) which has the largest y_i and divide the points into two groups $(x^{\rm L}_{\rm j}, y^{\rm L}_{\rm j})$ and $(x^{\rm R}_{\rm k}, y^{\rm R}_{\rm k})$ with the left group $x^{\rm L}_{\rm j} < x_0$ and $x^{\rm R}_{\rm k} > x_0$, as illustrated in Fig. S1a. (2) Calculate the arclength s^L or s^R from (x_0, y_0) along the left and right curve respectively, with each small segment having a length of $\Delta S^{\rm L}_{\rm j} = \sqrt{(x^{\rm L}_{\rm j} - x^{\rm L}_{\rm j+1})^2 - (y^{\rm L}_{\rm j} - y^{\rm L}_{\rm j+1})^2}$ and $\Delta S^{\rm R}_{\rm k} = \sqrt{(x^{\rm R}_{\rm k} - x^{\rm R}_{\rm k+1})^2 - (y^{\rm R}_{\rm k} - y^{\rm R}_{\rm k+1})^2}$, and normalize the arclength to the same interval $u^{\rm L}_{\rm j} = s^{\rm L}_{\rm j}/S^{\rm L}$ and $u^{\rm R}_{\rm k} = s^{\rm R}_{\rm k}/S^{\rm R}$ with $S^{\rm L}$ or $S^{\rm R}$ denoting the total arclength of the left and right curve respectively. (3) Interpolate the function that maps the re-scaled arclength $u^{\rm L}_{\rm j}$ and $u^{\rm R}_{\rm k}$ to point $(x^{\rm L}_{\rm j}, y^{\rm L}_{\rm j})$ and $(x^{\rm R}_{\rm k}, y^{\rm R}_{\rm k})$, and then average $(x^{\rm L}(u), y^{\rm L}(u))$ and $(x^{\rm R}(u), y^{\rm R}(u))$ with the same u to get the symmetrized profile $(r(u), z(u))$ with $r(u) = \frac{1}{2}[x^{\rm L}(u) + x^{\rm R}(u)] - x_0$ and $z(u) = \frac{1}{2}[y^{\rm L}(u) + y^{\rm R}(u)]$, as illustrated in Fig. S1b.

IV. FINITE DIFFERENCE METHOD

In order to verify whether the ML output $[r(u), z(u)]$ satisfies the variational equations eqr and eqz governed by the Helfrich theory, we adopt another equivalent form of the variational equations by introducing a new variable $\psi(u)$, which is the tangent angle spanned between the tangential direction and the horizontal direction [3]. We solve the shape equations with the MATLAB function bvp4c, which is a solver designed for boundary value problem (BVP) of ordinary differential equations. The method is based on the finite difference method that implements the three-stage Lobatto IIIa formula [4]. The tangential angle $\psi(u)$ satisfies the following geometric relations:

$$r'(u) = a\cos\psi(u), \quad (12)$$

and

$$z'(u) = -a\sin\psi(u), \quad (13)$$

where a is assumed to be a constant as explained in Supplementary Information I.

To calculate the force f mentioned in the main text, we introduce another term in the free energy

$$E_f = -f\,[z(0) - z(1)] \quad (14)$$

to account for the work done by a point force f to pull the membrane from 0 to z_0. The force f can be considered as a Lagrangian multiplier to impose the membrane height $z(0) - z(1)$. The total free energy of membrane can be rewritten as

$$L = 2\pi\int_0^1 \mathcal{L}\left[r, r', z, z', \psi, \psi', a, \alpha, \beta; \kappa, \sigma, p, f\right] du, \quad (15)$$

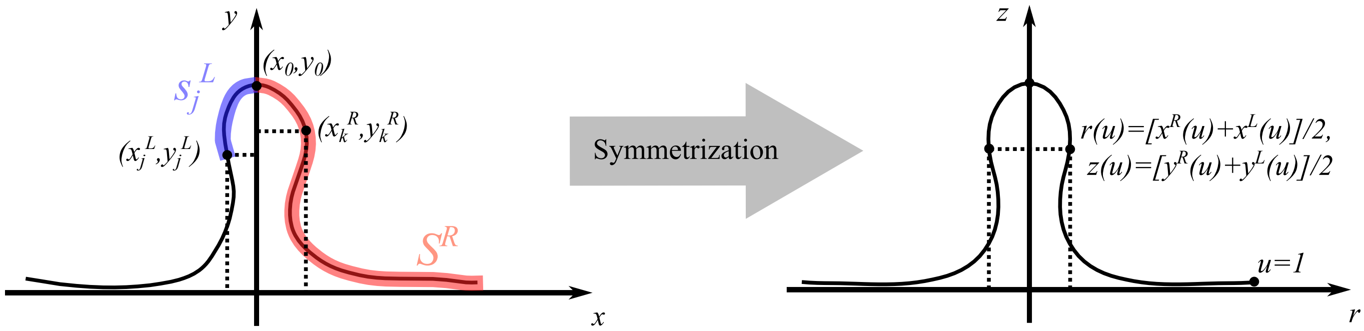

FIG. S1. Symmetrization of the experimental profile. (a) The experimental profile curve is divided into a left part $(x_{\rm j}^{\rm L}, y_{\rm j}^{\rm L})$ and a right part $(x_{\rm k}^{\rm R}, y_{\rm k}^{\rm R})$ from the highest point (x_0, y_0). Each part has its arclength $s_{\rm j}^{\rm L}$ or $s_{\rm k}^{\rm R}$, its total arclength S^L or S^R, and its re-scaled arclength $u_{\rm j}^{\rm L} = s_{\rm j}^{\rm L}/S^{\rm L}$ and $u_{\rm k}^{\rm R} = s_{\rm k}^{\rm R}/S^{\rm R}$. (b) Symmetrized experimental profile by taking the average of the left part and the right part at the same re-scaled arclength.

in which

$$\mathcal{L} = \frac{1}{2}\kappa\left(\frac{\sin\psi}{r} + \frac{\psi'}{a}\right)^2 ra + \sigma ra + \frac{p}{2}r^2 a\sin\psi - \frac{f}{2\pi}a\sin\psi + \alpha(r' - a\cos\psi) + \beta\left(z' + a\sin\psi\right). \quad (16)$$

Here $\alpha(u)$ and $\beta(u)$ are two Lagrangian multipliers to impose the geometric relations (12) and (13). By performing variations to the variables ψ, r and z, respectively, we obtain 3 corresponding equations

$$\psi'' = \frac{a^2 pr\cos\psi}{2\kappa} - \frac{a^2 f\cos\psi}{2r\kappa\pi} + \frac{a^2\alpha\sin\psi}{r\kappa} + \frac{a^2\cos\psi\sin\psi}{r^2} + \frac{a^2\beta\cos\psi}{r\kappa} - \frac{a\psi'\cos\psi}{r}, \quad (17)$$

$$\alpha' = a\sigma + apr\sin\psi - \frac{a\kappa\sin^2\psi}{2r^2} + \frac{\kappa(\psi')^2}{2a}, \quad (18)$$

$$\beta' = 0. \quad (19)$$

By performing variations to the variable a, we obtain a conserved quantity

$$H(u) = \frac{1}{2}pr^2\sin\psi - f\sin\psi - \alpha\cos\psi + \beta\sin\psi + \frac{1}{2}\kappa r\left[\left(\frac{\sin\psi}{r}\right)^2 - \left(\frac{\psi'}{a}\right)^2\right] + \sigma r = 0. \quad (20)$$

Eqs. (12), (13), and (17), (18), (19) constitute a system of ordinary differential equations, in which only Eq. (17) is second order, and all the others are first order. The system is equivalent to a total number of 6 first order ordinary differential equations. In addition, we have 2 unknown parameters, one being the total arclength a, the other being the force f. To complete the equations, we need a total number of 8 boundary conditions. They are listed below:

$$\begin{cases} \psi(0) &= 0 \\ r(0) &= 0 \\ z(0) &= z_0 \\ \beta(0) &= 0 \\ H(0) &= 0 \\ \psi(1) &= 0 \\ z(1) &= 0 \\ r(1) &= R_{\rm b} \end{cases}. \quad (21)$$

In summary we have constructed a well-defined BVP that is made of Eqs. (12), (13), and (17), (18), (19), two unknown parameters a and f, as well as 8 boundary conditions.

V. ROBUSTNESS TEST AGAINST NOISE

In order to test the performance of our method in the presence of data noise, we use the FD method to solve the membrane shape equations for a particular set of parameters $\kappa = 20\ k_BT$, $p = 1$ kPa, $\sigma = 0.01$ pN/nm, $R_b = 50$ nm, $z_0 = 100$ nm to get the membrane profile $[r(u), z(u)]$. We then interpolate the profile at a number of discrete points $\mathbf{X}_i = [r(u_i), z(u_i)]$ and add noise to the data $\mathbf{Y}_i = \mathbf{X}_i + [\epsilon_i^1, \epsilon_i^2]$ to generate an artificial experimental profile $\mathbf{Y}_i$, where ϵ_i^1 and ϵ_i^2 are Gaussian white noise with a zero average and a standard deviation of σ_ϵ. The learning procedure is then used to extract the model parameters from these generated profiles and the results are shown in Fig. S3. If the standard deviation of the Gaussian white noise is smaller than 1 nm, our method gives a both accurate and precise estimation of the parameters. However, when the standard deviation is greater than 1.5 nm, though the mean value over 10 repeated learnings remains close to the ground truth, the standard deviations become very large.

[1] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, Physics-informed neural networks with hard constraints for inverse design, SIAM Journal on Scientific Computing **43**, B1105 (2021).

[2] C. Leake and D. Mortari, Deep theory of functional connections: A new method for estimating the solutions of partial differential equations, Machine learning and knowledge extraction **2**, 37 (2020).

[3] F. Jülicher and U. Seifert, Shape equations for axisymmetric vesicles: a clarification, Physical Review E **49**, 4728 (1994).

[4] J. Kierzenka and L. F. Shampine, A bvp solver based on residual control and the maltab pse, ACM Transactions on Mathematical Software (TOMS) **27**, 299 (2001).

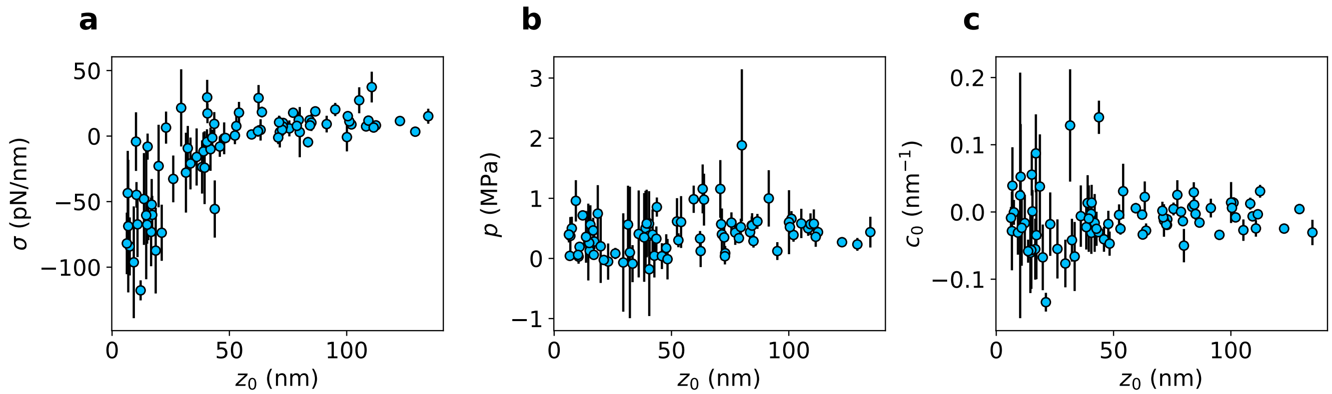

FIG. S2. Statistical analysis of ML-learned parameters with c_0. Model parameters as a function of the membrane height, with the membrane tension σ in (a), the osmotic pressure p in (b), and the spontaneous curvature c_0 in (c).

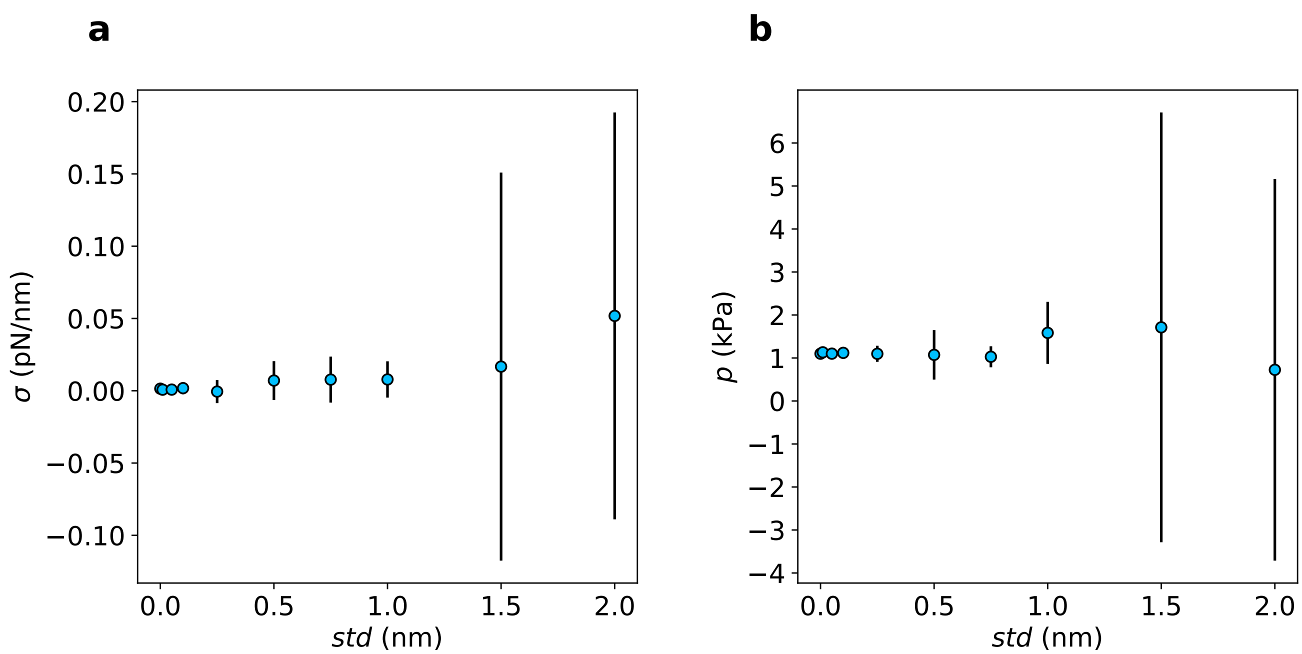

FIG. S3. Robustness test of the learning method against noise. (a) The learned membrane tension σ as a function of the noise strength (std). The blue dots represent the mean value over 10 repeated learnings and the error bars indicate the standard deviation. (b) The learned osmotic pressure p as a function of the noise strength. The ground truth parameters are $\kappa = 20k_BT$, $p = 1\text{kPa}$, $\sigma = 0.01\text{pN/nm}$, $R_b = 50$ nm, $z_0 = 100$ nm.