UC San Diego
UC San Diego Previously Published Works

Title

Modelling membrane curvature generation using mechanics and machine learning.

Permalink

https://escholarship.org/uc/item/0bd7n2ts

Journal

Journal of the Royal Society Interface, 19(194)

Authors

Malingen, S
Rangamani, Padmini

Publication Date

2022-09-01

DOI

10.1098/rsif.2022.0448

Peer reviewed

Author for correspondence:
P. Rangamani
e-mail: prangamani@ucsd.edu

Modelling membrane curvature generation using mechanics and machine learning

S. A. Malingen and P. Rangamani

Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, USA

PR, 0000-0001-5953-4347

The deformation of cellular membranes regulates trafficking processes, such as exocytosis and endocytosis. Classically, the Helfrich continuum model is used to characterize the forces and mechanical parameters that cells tune to accomplish membrane shape changes. While this classical model effectively captures curvature generation, one of the core challenges in using it to approximate a biological process is selecting a set of mechanical parameters (including bending modulus and membrane tension) from a large set of reasonable values. We used the Helfrich model to generate a large synthetic dataset from a random sampling of realistic mechanical parameters and used this dataset to train machine-learning models. These models produced promising results, accurately classifying model behaviour and predicting membrane shape from mechanical parameters. We also note emerging methods in machine learning that can leverage the physical insight of the Helfrich model to improve performance and draw greater insight into how cells control membrane shape change.

1. Introduction

Membranes compartmentalize cells while allowing controlled interactions across their interfaces. One of the membrane's core functions is facilitating communication across compartments, which can occur by uptake and release (endocytosis and exocytosis, respectively, when occurring across the external cell membrane). In addition to shaping a cell's microenvironment, these mechanisms act as homeostatic regulators [1,2]. Many elegant models of membrane deformation have been developed (as reviewed in [3]), and, in particular, Helfrich's continuum model based on thin shell elastic theory constrained by minimizing bending energy [4] is widely applicable. Using the Helfrich model, the equilibrium shape of the membrane can be predicted from a set of mechanical parameters (such as the bending rigidity of the membrane, its tension and many more). Modellers choose mechanical parameters to approximate biological/molecular mechanisms and to match experimentally measured values. Ultimately, this model can be used to show how cells may tune their mechanics to achieve experimentally observed shapes. Furthermore, the Helfrich model can also be used in reverse to determine the forces needed to maintain an experimentally observed membrane shape at equilibrium [5,6].

One of the primary challenges in each of these applications is prescribing the mechanical parameters of the membrane. Some parameters (such as tension, bending rigidity, pressure and size) have been measured experimentally, so modellers can reference from a rich body of the literature to determine reasonable ranges of values. Other parameters, like curvature and the functions used to describe how these values vary over the surface of the membrane, are heuristics that describe an amalgamation of factors. Ultimately, understanding the behaviour of the Helfrich model over large parameter spaces is an ongoing challenge,

leading to large amounts of labour in hand-tuning mechanical parameters to obtain convergent, biologically relevant results. Previously, sensitivity analysis has been used to determine how the prescription of the curvature function within the Helfrich model drives membrane energy and shape [7], which was a particularly vital advancement since preferred membrane curvature is not experimentally accessible. However, to the authors' knowledge the behaviour of the Helfrich model over large parameter spaces has not been documented. To fill this gap, we present several machine-learning models that enable systematic exploration of mechanical parameter space to understand membrane shape changes. In particular, we have used the formation of microparticles (MPs) as a case study for how different biophysical parameters drive shape change.

1.1. Microparticle formation is a mechanochemical process

MPs are a type of exocytic vesicle bounded by (and budding from) the plasma membrane in a process reminiscent of blebbing. They provide a mechanism for long-range communication between cells and tissues [8] that is vital in health, facilitating functions like blood clotting, but are deleterious in multiple diseases [9,10]. For instance, in cancer, they act as drivers of niche establishment [11–14]. Increased MP production also occurs in SCUBA divers during decompression [15,16]. In contrast to these examples of MPs in illness, physical exercise can modulate MP production as a signalling component in beneficial vascular adaptations that enhance blood flow to muscles [17]. Ultimately these are a few examples of how MPs enable long-range communication within an organism, using both their internal contents and the membrane itself as signalling platforms [9,18]. Finally, MP release can be a controlled process; in addition to their functional roles in health and disease, the regulation of their formation is an area of active investigation.

The formation of MPs is an inherently mechanochemical process [10]. We use spatially varying parameters in our continuum model to approximate the core biological mechanisms observed in MP formation. One of MP formation's hallmarks is the exposure of a negatively charged aminophospholipid, phosphatidylserine (PS), on the external membrane leaflet [9]. At high concentrations, PS induces membrane curvature [19–21]. We use a preferred curvature term to account for this process. In addition to PS exposure, curvature can be induced by myriad mechanisms [22], including from lateral heterogeneity in membrane composition [11,23], steric interactions between proteins adjacent to the membrane [24] (notably, cargo can be localized to the site of MP formation [11]), and even the glycocalyx can induce a preferred membrane curvature [25]. A high cytosolic calcium level in stimulated cells triggers PS exposure [9,26–28] and can activate calpain, which drives cortical cleavage and cytoskeletal remodelling [29]. Loss of cytoskeletal integrity can cause a decrease in membrane rigidity [28], and, along with cleaving the cytoskeleton, calpain can disrupt linkages between the membrane and the cortex [30]. The loss of integration with the cortex could allow the membrane to be pushed outwards due to internal hydrostatic force, similar to blebbing. Accordingly, we include spatially varying bending rigidity and outward pressure as tunable parameters in our biophysical model. Finally, all of these parameters coalesce to produce MPs that span a range

of sizes, and so we vary the patch area over which we adjust these parameters. While these biological events are well documented, it is unclear how they come together mechanically to accomplish MP formation.

We are informed by a rich body of membrane modelling literature [3], and our work builds from a previous model of clathrin-mediated endocytosis [31], a process that has been extensively studied from both an experimental and theoretical lens [6]. To adapt our model to MP formation, we are inspired by models of blebbing, where the cell membrane also balloons outward [32–34]. From this model, we derive energetically favourable membrane shapes for randomly sampled parameter combinations from a large mechanical parameter space. Subsequently, we used these data to train machine-learning models to predict membrane shape from mechanical parameters. Using this approach, we develop a predictive toolbox for mapping a wide range of mechanical parameters accessible in cells to shape outcomes figure 1.

2. Methods

2.1. Synthetic data generation

Ultimately the biophysical model we have used stems from minimizing the energetic cost of deforming a thin two-dimensional membrane (which is in mechanical equilibrium) embedded in three-dimensional space. The Helfrich energy functional describes the energetic cost of bending the membrane [4]. As was done in [31], we have used the Helfrich energy functional modified to allow spatially varying preferred curvature as a constitutive equation for the membrane

$$W = \kappa(\theta^\alpha)[H - C(\theta^\alpha)]^2 + \kappa_G(\theta^\alpha)K,$$

where κ is the bending rigidity of the membrane, which can vary spatially over the coordinate system, denoted here as θ^α, where $\alpha \in \{1, 2\}$; $H = \frac{1}{2}(k_1 + k_2)$ is the mean curvature of the membrane with k_1 and k_2 denoting the principal curvatures of the membrane; and C describes the spontaneous curvature of the membrane, which can vary spatially like the bending rigidity. Similarly, κ_G is the Gaussian bending rigidity, while $K = k_1 k_2$ is the Gaussian curvature. While elegant ansatz-based models of budding membranes have proven their utility [35,36] and may provide a faster, simpler avenue for predicting morphology, we chose, in this work, to retain the complexity of the Helfrich model to account for features like variable patch area, heterogeneous curvature and applied pressure. In the future, as more biological mechanisms are determined, the choice of modelling method can be revisited to ensure that the models encompass the vital biological components while reducing the complexity of computational methods.

We have adapted the model for membrane bending constrained by the Helfrich energy as derived by Hassinger *et al.* in [31]. This is valid since for both inward and outward budding the signs of the two principal curvatures match, which means that the Gaussian curvature, K, is equivalent for inward and outward budding. And since the signs of the principal curvatures are the same, the absolute magnitude of the mean curvature does not change, although its sign changes. Hence, in our model, there is no difference in the energetics of inward versus outward budding. The mathematical underpinnings of this model, and more generally, of thin fluid elastic membrane modelling, are explored in [37]. In brief, we assume that the membrane is in mechanical equilibrium, so the divergence of the stress vector field of the membrane summed with the pressure applied normal to the membrane is balanced by externally applied forces. This assumption also means that we neglect dynamics.
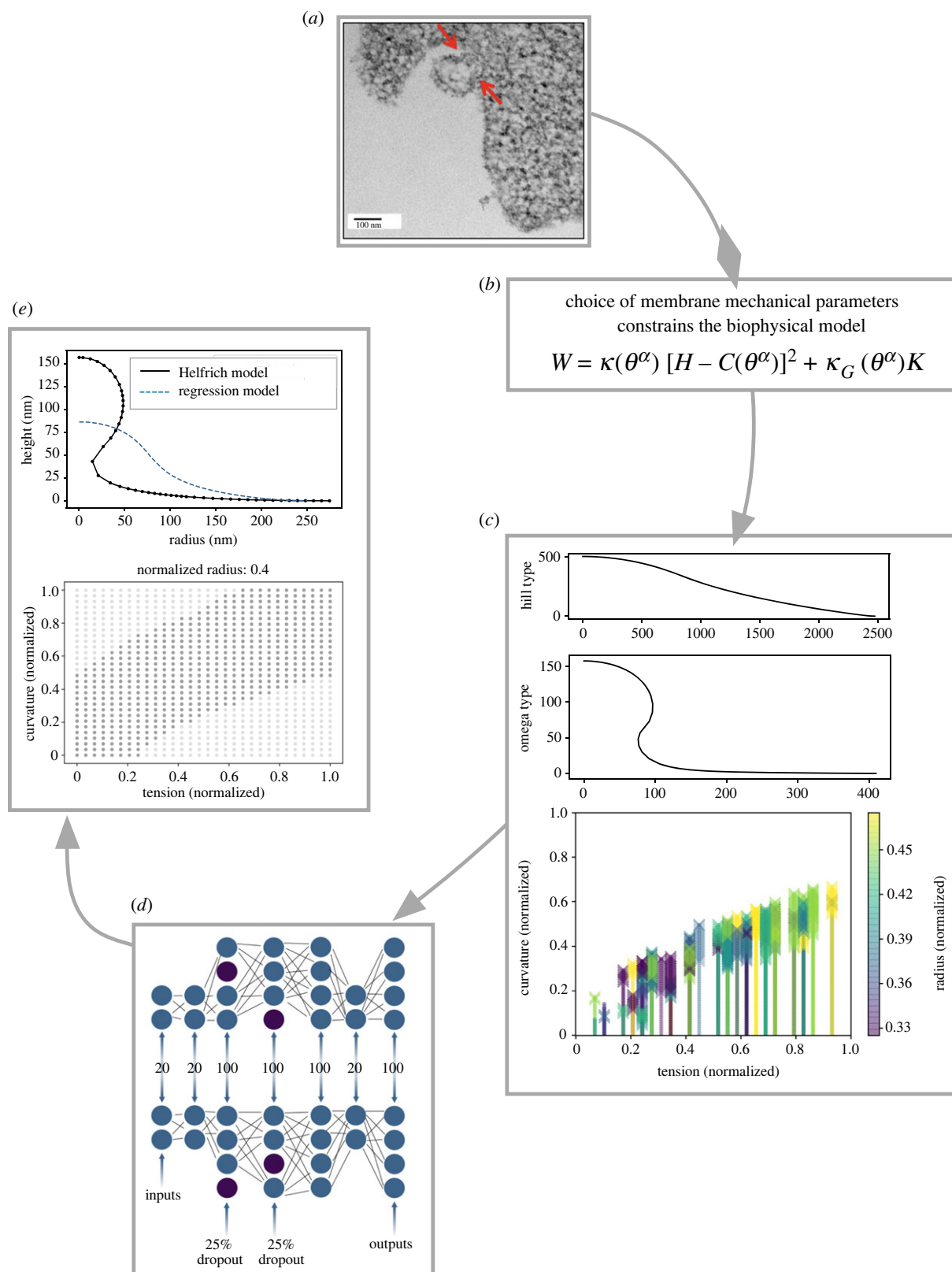
Figure 1. In this study, we used the classical Helfrich model to generate synthetic data that we used to train a machine-learning model to predict how the membrane forms MPs, given a set of mechanical parameters. (*a*) Membrane shape changes have been documented with microscopy techniques, for instance during microparticle formation (image from [73], figure reproduced with permission). (*b*) Biophysical models, like the Helfrich continuum model, can be paired with best-guess mechanical parameters in order to approximate experimentally observed shapes. (*c*) We used the Helfrich model to create a large synthetic dataset of membrane shapes. Here, we show a curve that was classified as a hill and a curve classified as an omega shape (in omega-classed shapes a neck forms which could dispose the membrane to bud). Classification was based on if the angle of a tangent to the curve exceeded 90°. We also show a scatterplot of classification results for a parameter combination where omega-classed buds are marked by an 'x' and hill-classed curves are marked with a small dot. Due to the dense sampling of curvature needed to obtain numerically convergent results solving the Helfrich model, the small dots overlap and appear almost as lines. (*d*) Using this synthetic dataset, we trained machine-learning models to classify model behaviour over large parameter spaces, and to predict shapes from mechanical parameters. In (*e*), we show samples of machine-learning results for classification and regression.

We also assume that the bilayer is incompressible, so a Lagrange multiplier prescribing that the bilayer's density is constant can be imposed. These two observations lead to reconstituting the energy of the membrane in terms of the Helfrich energy summed with the Lagrange multiplier. Finally, the equations of motion can be reframed to yield the force balance normal to the membrane

$$p + 2\lambda H = \Delta[\kappa(H - C)] + 2H\Delta\kappa_G - (\kappa_G)_{;\alpha\beta}b^{\alpha\beta}$$
$$+ 2\kappa(H - C)(2H^2 - \kappa) - 2\kappa H(H - C)^2, \qquad (2.1)$$

and the tangential force balance within the membrane

$$\lambda_{,\alpha} = \frac{\delta\kappa}{\delta\theta^\alpha}(H - C)^2 + 2\kappa(H - C)\frac{\delta C}{\delta\theta^\alpha} - \frac{\delta\kappa_G}{\delta\theta^\alpha}K. \qquad (2.2)$$

To simplify finding solutions to these force balances, we constrain ourselves to axisymmetric solutions where s describes the position along an arc, and θ is used here to describe the rotation of the arc about the z-axis (instead of describing the coordinate system as it was used previously), and the r-axis describes the radial distance from the z-axis. The angle between the tangent (a_s) to the arc and the base plane is denoted ψ. As demonstrated by Hassinger et al. [31], a system of six ordinary differential equations (ODEs) can be derived from these constraints. To solve this system, we used the same six boundary conditions, namely at $s = 0^+$

$$R(0^+) = 0, \quad L(0^+) = 0, \quad \psi(0^+) = 0 \qquad (2.3)$$

and at $s = S$

$$Z(S) = 0, \quad \psi(S) = 0, \quad \lambda(S) = \lambda_0. \qquad (2.4)$$

In these clamp boundary conditions, the angle ψ is fixed at 0 at both the beginning and the end of the simulation domain. The variable λ is interpreted as tension [38,39], and so while tension can vary along s, the tension at the outer edge of the domain is fixed at λ_0. We used the area parametrization of the system.

2.2. Mechanical parameter choice
2.2.1. Spatial heterogeneity
The composition of the membrane varies laterally, and we propose that variation in the mechanical features of the membrane are a core component of MP formation. Hence, we define a patch of the membrane at the centre of the simulation domain where the mechanical parameters are altered from their values over the rest of the membrane. Since cells can produce MPs spanning a broad range of sizes, we explored a range of patch sizes. We aimed to model the formation of MPs ranging from 25 to 500 nm in radius (r), so we set the patch area to be $4\pi r^2$, while the larger simulation domain was chosen as $4\pi(3r)^2$.

2.2.2. Tension
The apparent tension at the cell surface that constrains shape change is the sum of the in-plane tension in the plasma membrane with the adhesion of the membrane to the cortex [40]. Proteins linking the membrane to the cortex impede the flow of lipids into a tether creating viscous resistance that can be measured with dynamic tether pulling, characterizing the contribution of the cytoskeleton to apparent membrane tension [41]. Meanwhile, in-plane membrane tension can be characterized by isolating the membrane from the cortex, along with several other methods [42]. Since MP formation requires detaching from the cortex, we are most concerned with these values.

By measuring the tension of the membrane in blebs, the contribution of cytoskeletal adhesion can be removed, resulting in values near 0.003 pN nm^{-1} in rabbit renal proximal tubule epithelial cells and about 0.012 pN nm^{-1} in human melanoma

cells [43]. By contrast, the tension in membrane associated with the cortex can reach about 0.022 and 0.044 pN nm^{-1} in these cell types, respectively [43]. These values are similar to the approximately 0.003 pN nm^{-1} apparent membrane tension measured in neuronal growth cones [40], and the approximately 10 times larger tensions that have been measured in neutrophils [44,45].

While we have chosen a relatively small range of tension values, experimental measurements of membrane tension vary broadly with reports ranging from 0.003 to 0.276 pN nm^{-1}, while the in-plane membrane tension has been reported to take values from 0.003 to 0.15 pN nm^{-1} [42]. We have chosen tension to fall within 0.003–0.15 pN nm^{-1} since it encompasses both the reported values for in-plane membrane tension, along with the apparent tension for several cell types, but we neglect the upper range of apparent tension values that have been reported. It is noteworthy that tension varies across cell type, state and stretching regimes [42,46]; and its role in global and local regulation and the rate of its propagation through the membrane are an active area of discovery [41,42].

2.2.3. Bending rigidity
The bending rigidity of the cell membrane is often derived from experimentally measured values in giant unilamellar vesicles (GUVs) [47], or red blood cells [48], using techniques such as fluctuation analysis (e.g. [49]), or micropipette aspiration (e.g. [50]). By using a reduced system, the influence of individual constituents on membrane mechanical properties can be identified. For instance, the inclusion of sterols can increase the bending rigidity of some membranes [51], but not all [47]. While these techniques have provided an extensive literature base to estimate the rigidity of a lipid bilayer (usually between 10 and 150 $k_B T$ [48,52]), the bending rigidity of the cell membrane in vivo may be heterogenous and take a range of values. Bending rigidity can be influenced by a host of factors, ranging from charge [47], to extensions from the membrane like the glycocalyx, protein interactions with the membrane [53], or interaction with the cortex [32], and probably many other complexities. In light of these challenges, Steinkühler et al. [52] have measured the bending rigidity of the membrane in giant plasma membrane vesicles (GPMVs), effectively maintaining a biological membrane composition while isolating the mechanical properties of a cell membrane from cytoskeletal interactions. Probably, the bending rigidities they measure are similar to that of MPs, falling approximately in the range of 10–35 $k_B T$ (which at a temperature of 298 K corresponds to approximately 40–145 pN nm). The bending rigidity they measure in isolated membrane-derived vesicles is similar to that measured in expanding blebs [32,52], but when blebs begin retraction the rigidity of the membrane increases to about 215 pN nm, which is attributed to the formation of actin under the lifted membrane [32]. Further, Charras et al. [32] found that treatment with wheat germ agglutinin could increase the membrane rigidity to approximately 360 pN nm. To simplify our model, we set a fixed bending rigidity of the membrane surrounding the site of MP formation at 320 pN nm. Given the breadth of measured values for membrane rigidity, we allow the rigidity of the region from which the MP forms to fall within the range of 40–400 pN nm^{-1}. In our model set-up, we use a multiplier to denote the difference between base membrane rigidity and the rigidity of the patch where a MP is expected to form. Based on the range of bending rigidities, we found reported in the experimental literature we chose this multiplier to be in the range 0.125–1.25.

2.2.4. Pressure
To model hydrostatic pressure, we have applied an outward normal force ranging from zero to 0.0003 pN nm^{-2} to the membrane on the patch region. These values were based on the values

reported in a number of studies which were reviewed in [54], although we chose to explore only the lower range of pressure values they report. As is also reviewed in [54], hydraulic pressure is a critical driver of cell blebbing, which bears similarity to MP formation.

2.2.5. Curvature

In order to obtain convergent results from the continuum model, we initialized a flat membrane with zero mean curvature. Then we looped over small, increasing steps in curvature where the initial guess fed to the model was the curvature at step $i-1$ and the model was set to find an energetically favourable solution for the curvature at step i. For many parameter sets, the model did not converge after a given value of curvature. So while curvature values could reach $0.07\,\mathrm{nm}^{-1}$ (which corresponds to a radius of about 14 nm), this was rare. Instead, the model was allowed to slowly increase curvature in step sizes of $0.0007\,\mathrm{nm}^{-1}$ until it no longer converged.

2.3. Data generation

In order to obtain a convergent result from the system of ODEs, the solver needs an initial guess. Therefore, we start with a curvature of zero over the initial patch and a flat membrane as an initial guess and iterate over gradually increasing values of curvature, as described in the preceding methods section dedicated to curvature. Since the Helfrich model produces snapthrough instabilities for some parameter regimes [31], we used the final solution that produced a convergent result for increasing values of curvature as an initial guess for a loop where we iterated over decreasing values of curvature. We included both the ascending and descending results without distinction in the training set. Often, we could not find a convergent solution, especially in the descending direction. While on a case-by-case basis factors like the solver, mesh density or initial guess could be adjusted to search for a solution, this is not practical for a large parameter space. Hence, the absence of a convergent solution in our dataset does not preclude the existence of one. We used *scipy.integrate.solve_bvp* to solve the system of ODEs with the default Runge–Kutta method of order 5(4), a tolerance of 1×10^{-2} and the maximum number of nodes set to 1000 times the number of points on the mesh (2000). We found this solver to be amenable to our problem since it is relatively robust and fast, allowing us to rapidly generate a large dataset. However, other methods may yield convergent solutions where we could not find them using this solver (for instance, see [55–57]) The solver performed better when the initial mesh (between 0 and 1) was squared so that the density of points near the centre (which corresponds to the centre of the patch) was larger. The mesh was then resized to the total patch area.

2.4. Machine-learning model development

In this work, we aimed to use machine learning to classify membrane shape with the goal of creating phase maps that delineate regions of parameter space where omega-shaped buds are probable. We also created a regression model to predict equilibrium membrane shape explicitly from input parameters.

2.4.1. Classification models

We used the criteria that $\psi > 90°$ to label omega-shaped buds. In all cases, the model was tasked to predict if the membrane shape was an omega-shaped bud or not (usually a hill-shaped or flat bud). The features provided to the model were the physical parameters used in the continuum model of membrane shape. These include: mean curvature, the tension at the membrane's edge, the size of an initial patch or membrane, the bending rigidity of the patch relative to the rest of the membrane, and an outward

normal forced applied to the membrane patch which models pressure on a region of the membrane severed from the cortex. Since these parameters alone yield a five-dimensional feature space, we simplified this problem to two four-dimensional parameter spaces: in the results section, we first address how curvature, tension, patch size and pressure impact the formation of omega-shaped buds, then we address how curvature, tension, patch size and a difference in bending rigidity over the patch region can give rise to omega-shaped buds.

While there are many machine-learning methods that can be used for this type of classification problem [58], we restricted ourselves to xGBoost models and neural net models, creating one of each for each training set. The hyperparameters of the models were tuned by hand, and the model's performance at this stage was determined on a withheld dataset composed of 3741 points for the dataset varying pressure, and 2201 points for the dataset varying the difference in bending rigidity of the MP patch from that of the rest of the membrane.

In addition to these models, we also found that a k-nearest neighbours (kNN) model using the closest neighbour ($k = 1$) had a decent baseline performance. For the pressure dataset, it correctly classified 2972 hill-shaped deformations and 73 omega-shaped deformations while misclassifying 42 hill-shaped deformations as omega-shaped deformations and 60 omega-shaped deformations as hill-shaped deformations. For the dataset where we varied the bending rigidity of the central patch, the model correctly classified 2511 hill-shaped deformations and 42 omega-shaped deformations while misclassifying 28 hill-shaped deformations as omega-shaped deformations and 58 omega-shaped deformations as hill-shaped deformations. Although this model performed decently, the notion of a nearest neighbour in our mechanical parameter space is contrived since parameters are in different units. Additionally, one of the parameters that can be adjusted in a kNN is the value of k, which represents the number of nearest neighbours voting towards the predicted points value. Since our dataset is imbalanced, we chose only to use $k = 1$.

2.4.2. Methods for imbalanced classification problem

Both of the datasets are imbalanced, meaning that there are many more non-omega-shaped (hill-shaped) membrane profiles than omega-shaped ones. The dataset investigating the impact of pressure has a total of 24 499 samples with only 1513 (about 6.2%) classified as omega-shaped buds. The dataset investigating bending rigidity has even fewer omega-shaped buds, with a total of 18 304 data points and only 453 (about 2.5%) representing omega-shaped curves. Since imbalanced classification is an important problem (for instance, in medical diagnostics), many approaches have been created to optimize classifier performance. Broadly, these can be viewed as adjusting the ratio of the classes in the training set (for instance, synthetic minority oversampling technique and iterations thereof [59–62]) and re-weighting the minority class within the model framework itself. We used two of the simplest approaches: in the xGBoost model, we used a built-in hyperparameter (scale_pos_weight) to give preference to the minority class [63], while in the neural net model we re-sampled the training set to have a ratio of 7 : 1 majority to minority data points.

2.4.3. Neural net

Our neural net model was heavily based on a tutorial from the tensor flow core for classification on imbalanced data [64]. In brief, we used a sequential model with five dense layers and one dropout layer. Dropout is a useful tool for preventing model over-fitting and enhancing a model's ability to generalize for an unseen test set [65]. We used the rectified linear activation function in all layers except for the final output layer, where we

used a sigmoid activation function for binary classification. The model was constructed with four hidden layers and a dropout layer, in addition to a single output layer: 10/100/100/50% dropout/10/1. The model architecture and weights are available in the GitHub repository: https://github.com/RangamaniLabUCSD/Modeling_membrane_curvature_generation_using_mechanics_and_machine_learning. We used a batch size of 32.

We used a callback function (see [66] for a tutorial on callback functions) for early stopping with a patience of 200 epochs, and a maximum of 4000 epochs was allowed during model training. Additionally, we used another callback function to save the best model developed during model fitting, also with a patience of 200 epochs. In both cases, these callbacks were set to monitor the area under the precision-recall curves (AUPRC) (which is explained in the following section).

2.4.4. xGBoost

xGBoost can be a highly effective machine-learning method for classification problems [67]. Selecting hyperparameters to prescribe the model requires balancing a model that can both account for the complexity of the data and mitigate over-fitting. In table 1 we have recorded the hand-tuned hyperparameters that we used to define the model. While we chose to hand-tune hyperparameters in this instance, the performance of an xGBoost model can be increased through hyperparameter optimization, which is an active area of research (for example, see publications such as [68–70]).

2.4.5. Measuring classification model performance

Confusion matrices are one way to assess a classification model's performance, allowing the reader to determine the number of times a model correctly predicts each class, and the number of times it misclassified. In a binary classification problem, each of the boxes represents either a true negative, false negative, true positive or false positive. However, our classification models do not return strictly binary results; instead they return a probability (between 0 and 1) of an instance belonging to the 0 or 1 class. The confusion matrices presented for each model in this paper represent the model's prediction when a threshold of 0.5 was chosen for determining class membership. However, this threshold can be increased or decreased, resulting in infinitely many possible confusion matrices for a single trained ML model. To measure performance across all possible thresholds, the numbers of a confusion matrix can be embedded in a single metric, and the measure reported along a curve. Then the model's performance over all thresholds can be quantified as the area under the curve. The metric for model performance needs to meet the needs of the classification problem.

Since there is a large imbalance in the number of omega to non-omega membrane shapes in our dataset, several commonly used metrics of classification performance (namely accuracy and the receiver operating characteristic) are inappropriate since they count the number of correctly identified majority class instances. To better quantify the model's capacity to correctly identify the minority class we use the precision-recall curve (PRC). Precision is the proportion of times that the model correctly predicted the minority class (the true positives divided by the total number of positive predictions). The recall (or sensitivity) of the model, by contrast, is the proportion of actual positive cases that the model correctly identifies (the number of true positives divided by the number of false negatives plus true positives). Because neither precision nor recall take into account the number of majority cases that the model correctly identifies, the PRC highlights the model's ability to discern the minority class without being influenced by the model's ability to accurately predict the majority class. For a helpful tutorial on these ideas see [71]. Precision and recall values, then, are individual values attributed to a model with a given threshold,

Table 1. The hyperparameters prescribing the xGBoost model were chosen manually.

xGBoost hyperparameters	
objective	**binary: logistic**
learning_rate	0.01
max_depth	100
min_child_weight	10
gamma	1
subsample	0.2
colsample_bytree	0.9
seed	23
n_estimators	700
scale_pos_weight	$\frac{no.\ non-omegas}{no.\ of\ omegas}$ in train set

so the PRC is parametrized by threshold values in the interval [0,1] for a given model. Model performance can then be assessed from the AUPRC.

2.5. Neural net for regression

Phase-space visualization compresses the shape of the membrane to a binary indication of behaviour for a given set of mechanical parameters. While this reduction is useful, we also sought to create a regression model that used mechanical parameters as the input features to predict the diverse shapes the membrane may adopt. The training dataset for this problem was generated in the same step as the regression problem. In addition to whether or not the membrane adopted an omega shape, we recorded the x and y locations of 50 points interpolated along the curve (downsampling from an initial mesh of 2000 points). The neural network was tasked to then predict from a point in mechanical feature space the x and y locations that define the curve. The dense network was composed of nine layers: two dropout layers, six dense layers and a dense output layer of 100 nodes, with a structure of 20/20/100/25% dropout/100/25% dropout/100/20/100. We used a rectified linear unit activation function, Adam optimizer with a small learning rate of 0.001, batch size of 64 and mean absolute error (MAE) loss function. Similar to the classification problem, we used callbacks for early stopping and a checkpoint to record the model that produced the lowest MAE across all training epochs, both with a patience of 200 epochs. In total, the model was limited to a maximum of 4000 epochs.

2.5.1. Measuring regression model performance

The regression model was tasked to minimize the MAE between individual 'x' and 'y' coordinates along an objective curve. In reporting model performance, we again used the MAE and supplemented it with a scaled error metric, since large membrane patches with relatively similar shapes can receive a higher MAE than smaller membrane patches with very different shapes. The scaled error was computed by first dividing both the prediction and the actual membrane curve by the maxima of the x and y locations, respectively, of the validation curve. By handling it in this way, differences in the height and diameter of the membrane between the prediction and actual data are still apparent, but are scaled relative to true curve's dimensions. Then the minimum Euclidean distance between each point of the predicted curve was computed across all of the points documented along the true curve. Ultimately, this value is still greater than or equal to the

distance between the predicted point and the target curve. This second metric is a truer estimate of the model's capacity to correctly predict membrane shape regardless of size. Using a metric like this as the loss function for the model itself, however, is not practical, since we seek to predict a relatively even distribution of points along the entire curve, while this particular error metric could be minimized by placing all points near the outer clamp boundary condition where the 'y' coordinate approaches zero and the 'x' coordinate could be inferred from initial patch radius.

We inquired if there is a correlation between error and the distance between the test (or validation) point in mechanical parameter space and the nearest training point. Since these parameters are measured in different units and cover unrelated numerical ranges, measuring 'distance' in mechanical parameter space is necessarily contrived. We normalized the range of each mechanical parameter to fall within [0,1] and, since we used a four-dimensional parameter space, used the L1 (Manhattan) norm to compute the distance [72].

3. Results and discussion

Cells deform their membranes in a host of processes, and here, we used MP formation as a case study. In modelling MP formation, we apply the elegant Helfrich continuum model to derive energetically favourable membrane shapes for a given set of mechanical parameters. Shape transformations depend on a host of mechanical properties of the membrane, such as the bending rigidity of the bilayer, its tension, and more; these also constrain biophysical models of membrane shape change. While some of these parameters are experimentally accessible, they can vary spatially and temporally within a cell, making model parametrization non-trivial. Further, the curvature parameter, in particular, is used in continuum models to approximate the effect of an amalgamation of forces (protein-induced spontaneous curvature, steric repulsions and charge interactions, for instance) that drive the membrane to bend. Therefore, the preferred curvature of a membrane is a heuristic that cannot be measured, even while model outcomes depend intimately on its prescription [7]. Historically, biophysicists have achieved relevant model results by manually tuning the mechanical parameters based on ranges from experimental work and using techniques like grid search to inform phase behaviour. However, for high-dimensional parameter spaces, these techniques are very time-consuming and can limit the interpretability of results. Therefore, we sought to use machine learning to represent model behaviour over two randomly sampled, four-dimensional mechanical parameter spaces. We present the results of this work into two categories: classification, where we used xGBoost and a neural network to create phase spaces, and regression, where we used a neural network to predict membrane shape from mechanical parameters.

3.1. Classification models predict phase behaviour

We used the Helfrich continuum model to predict curves representing membrane shape deformations for randomly sampled points in mechanical parameter space. A curve was classified as an omega-shaped bud if the angle ψ (figure 2) exceeded 90°, and as a hill-shaped deformation otherwise (in figure 1c, we show examples of curves generated from the Helfrich model where the first was classified as a hill shape and the second as an omega shape). In figures 3
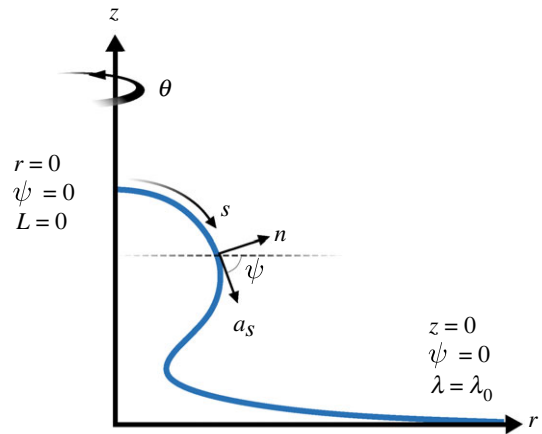

Figure 2. Schematic illustrating the key variables used to prescribe the model, and the boundary conditions that constrain it.

and 4, we show the results of the Helfrich continuum model where parameter combinations yielding omega-shaped buds were marked with an 'x', while parameter combinations that yielded a hill-shaped curve are marked with dots. The first three rows of these figures show slices of parameter space for small, medium and large ranges of patch area (which corresponds to MP size).

In figure 3, where pressure was varied, we show that for small and medium patch areas there is a clear trend that as tension increases an increased preferred curvature is needed to drive the formation of omega-shaped buds. By contrast, for large patches, only a few parameter combinations resulted in omega-shaped buds, and these usually required relatively high pressures. The final panel of figure 3 shows a visual separation in the regions of parameter space where omega-shaped buds are more likely to occur from regions where they are unlikely to occur. However, the relationship of four mechanical parameters is not readily visualized.

The results shown in figure 4, where the bending rigidity of the patch relative to the rest of the membrane was varied (and applied pressure was set to zero), do not show as clear a pattern as those in figure 3 where the pressure applied to the patch area was varied. Generally, omega-shaped buds did not occur when the bending rigidity of the patch was much smaller than that of the rest of the membrane. Occasionally, for the largest patch sizes an omega-shaped bud could be formed; these tended to occur when the bending rigidity of the patch was larger. Additionally, the same general trend of increasing curvature necessitated with increasing membrane tension was seen, but there was not a clear visual separation of regions of parameter space where MPs may form from where they do not.

These results gave us a visualization for how phase maps should look. Launching from this point, we built two machine-learning models to predict whether the membrane would adopt an omega shape from the mechanical parameters constraining the Helfrich model. In essence, the models were tasked to interpolate how the membrane would behave based on nearby training points. Both the xGBoost model (figure 5) and neural network model (figure 6) performed well for the dataset where pressure was varied, with the AUPRC being 0.84 and 0.93, respectively. For the dataset varying bending rigidity, the neural network performed well, with an AUPRC of 0.82 (figure 7)
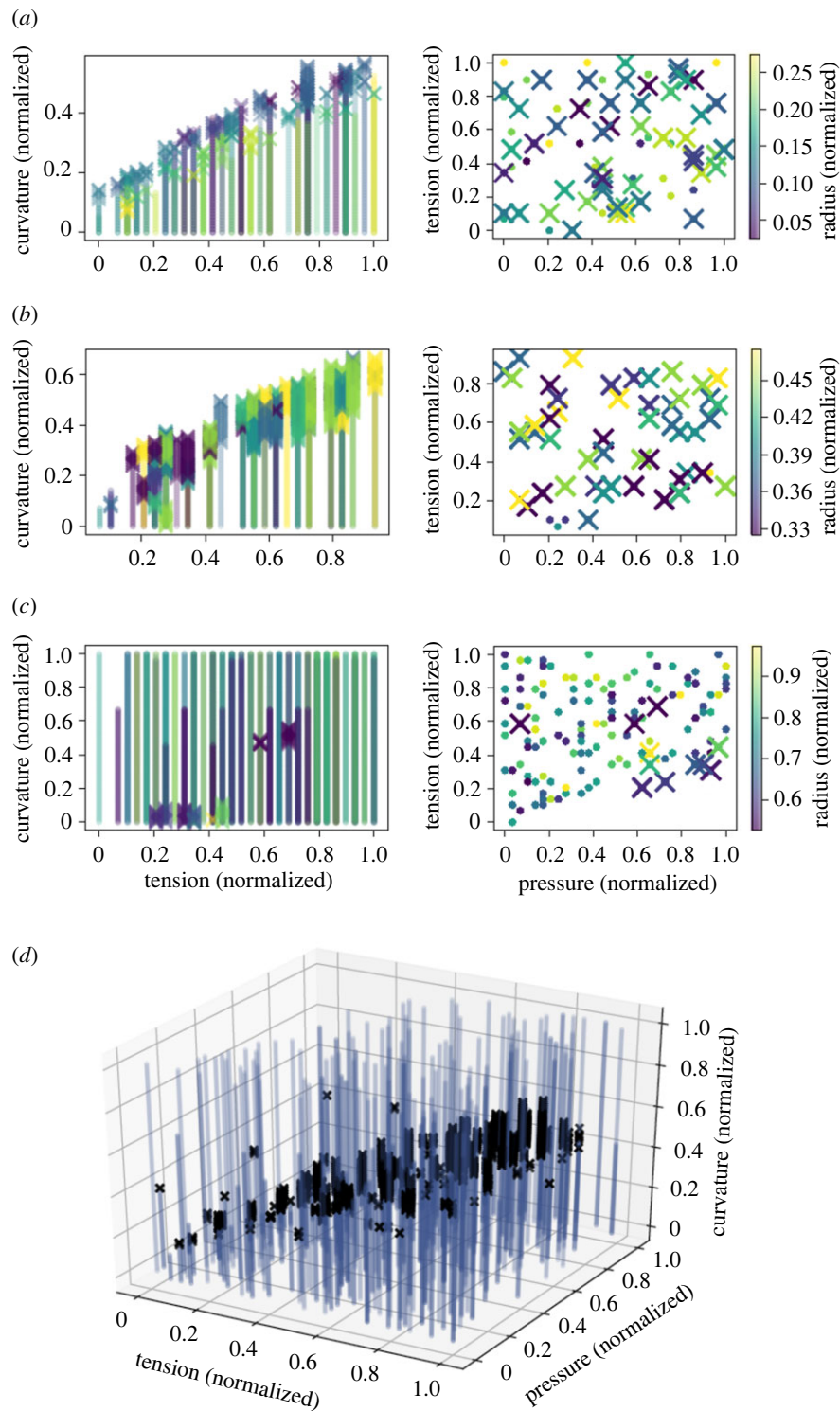
Figure 3. Results of the continuum model show regions of the four-dimensional mechanical parameter space (pressure, tension, curvature and patch size) where omega-shaped buds formed. (*a*) Parameter combinations yielding an omega-shaped bud are marked with an 'x' while those yielding a hill shape are marked with a dot. Here, we show data for patches of normalized radius smaller than 0.3 with the colour map corresponding to the size of the patch. For small patches, as tension is increased, preferred curvature also needs to increase in order for omega-shaped buds to form. (*b*) Similar to small patches, for medium-sized patches with normalized radius between 0.3 and 0.5, as tension increases, preferred curvature must also increase to yield an omega-shaped bud. (*c*) For the largest patches (normalized radius between 0.5 and 1), omega-shaped buds are rare, usually only forming for relatively large pressure forces. (*d*) A three-dimensional plot of parameter combinations yielding omega-type buds (black 'x's) and combinations yielding hill-type deformations (blue dots) shows a band of parameter space where omega-type buds are likely to form. Here, the data for all patch sizes are overlayed.

while the xGBoost model had a poorer performance with an AUPRC of 0.39 (figure 8). We chose to measure model performance using the PRC due to the large class imbalance (with omega-shaped buds in the minority class). These models were notably different in two regards. The first is that in the region of parameter space where we were not able to obtain convergent results from the Helfrich model, the machine-learning models were forced to extrapolate. Interestingly, the neural network guessed in these uncharted regions that the membrane would not adopt an omega shape; meanwhile, the xGBoost model guessed that the membrane would adopt an omega shape. Given the limitations of our
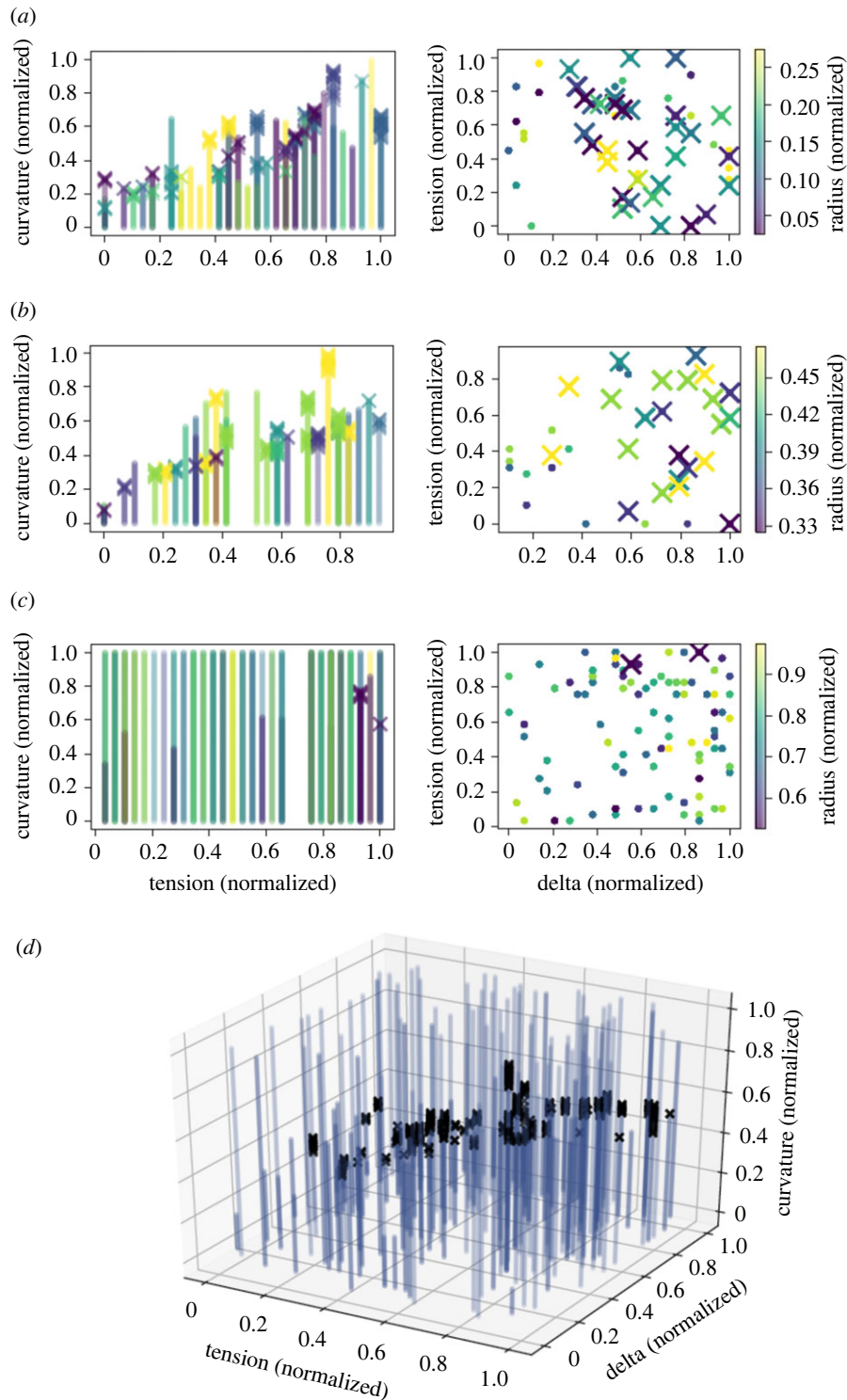
Figure 4. Change in the bending rigidity of the patch region from that of the rest of the membrane shows that omega-shaped buds are less likely to form in regions of parameter space where bending rigidity is lost. (*a*) Parameter combinations yielding an omega-shaped bud are marked with an 'x'. The colour map corresponds to the size of the patch. For the smallest patches (normalized radius less than 0.3), omega-shaped buds were fairly common except where delta was small (regions where the patch was less rigid than the surrounding membrane). (*b*) A similar trend to small patch sizes was observed for medium-sized patches (normalized radius between 0.3 and 0.5). (*c*) For the largest patch sizes (normalized radius between 0.5 and 1), very few omega-shaped buds were formed. (*d*) A three-dimensional plot of parameter combinations yielding omega-shaped buds (marked with black 'x's) and parameter combinations yielding hill-type deformations (marked with blue dots) shows that most parameter combinations did not yield omega-typed buds. In this panel, all patch sizes are overlayed.

datasets, it is not reasonable to expect either model's predictions to capture actual membrane shape. Further, the absence of data in these regions hints at the possibility that energy-minimizing shapes for these parameter combinations may not exist or different computational schemes may be needed to solve these equations [7,56,57]. A second more

subtle difference between the predictions of the models is the edges in the phase maps which do not precisely align. It is possible that either model could perform better in these regions under different training conditions such as increasing the sampling of parameter space, particularly around phase transitions.
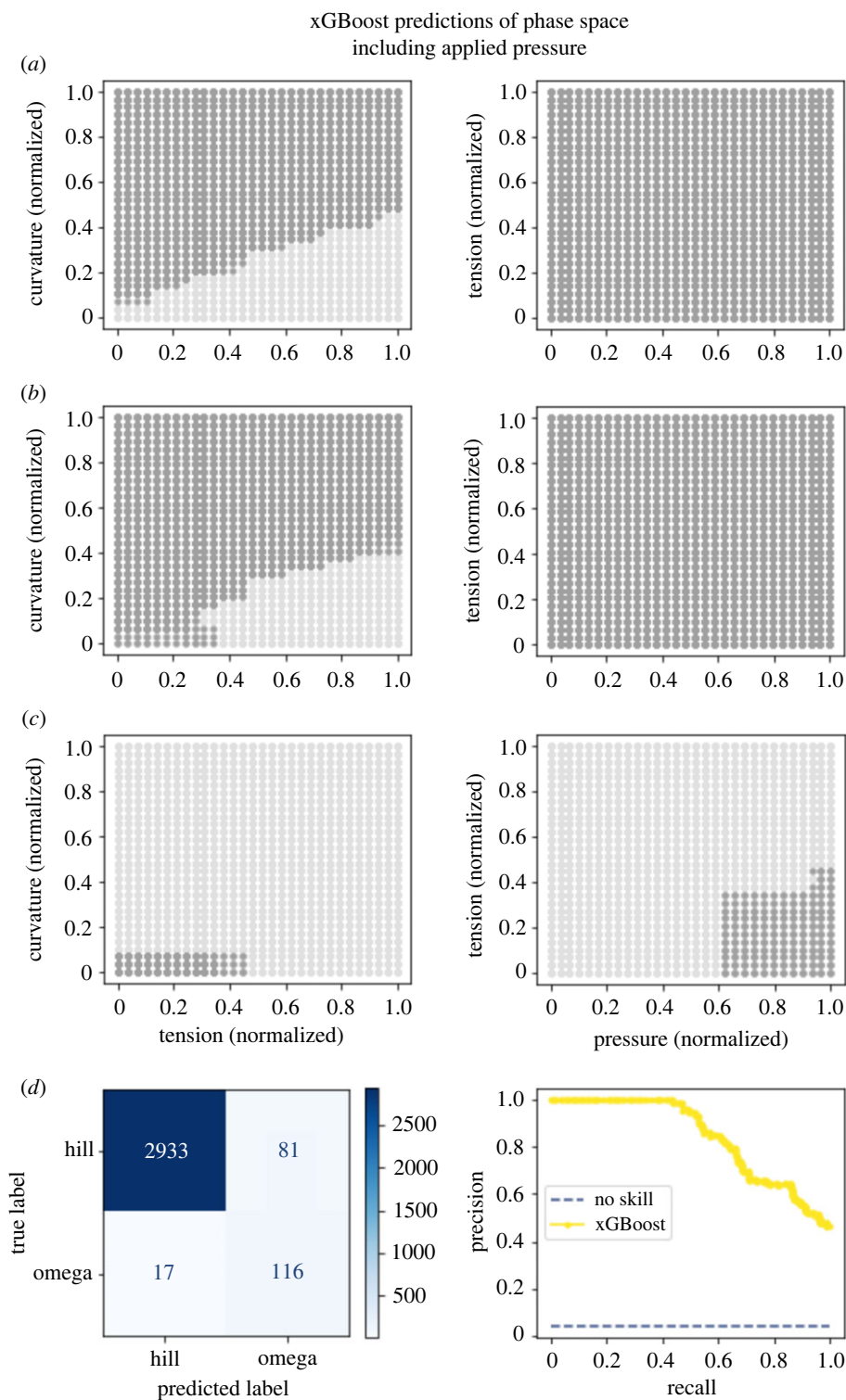
Figure 5. An xGBoost model captures a phase map of mechanical parameter space. (*a*) Model predictions for patches with a normalized radius of 0.15 where dark grey dots represent an omega class prediction and light grey represent a hill class prediction. (*b*) Model predictions for patches with a normalized radius of 0.4. (*c*) Model predictions for patches with a normalized radius of 0.7. (*d*) The confusion matrix for this model demonstrates that the model has a relatively high false positive rate. The model overall had high performance with an AUPRC of 0.84.

These classification models may enable researchers to more rapidly tune mechanical parameters by providing a computationally light-weight classification. Additionally, they may allow us to bridge the gap between experimental observations of membrane deformation through electron microscopy and computational modelling of membrane mechanics.

3.2. A regression model predicts membrane shape

Compressing the membrane shape to a binary categorization of omega- or non-omega-shaped buds allowed us to create phase diagrams that succinctly convey trends. However, while this compression provides an excellent overview of how a thin elastic membrane behaves, it limits our ability to understand the diversity of shapes the membrane may adopt. Further, obtaining a convergent result from the Helfrich model for a given set of parameters can be quite time consuming, requiring a user to tune the mesh size and step sizes. Therefore, we sought to train a regression model to explicitly predict membrane shape for a given set of mechanical parameters. One of the challenges of this problem, in
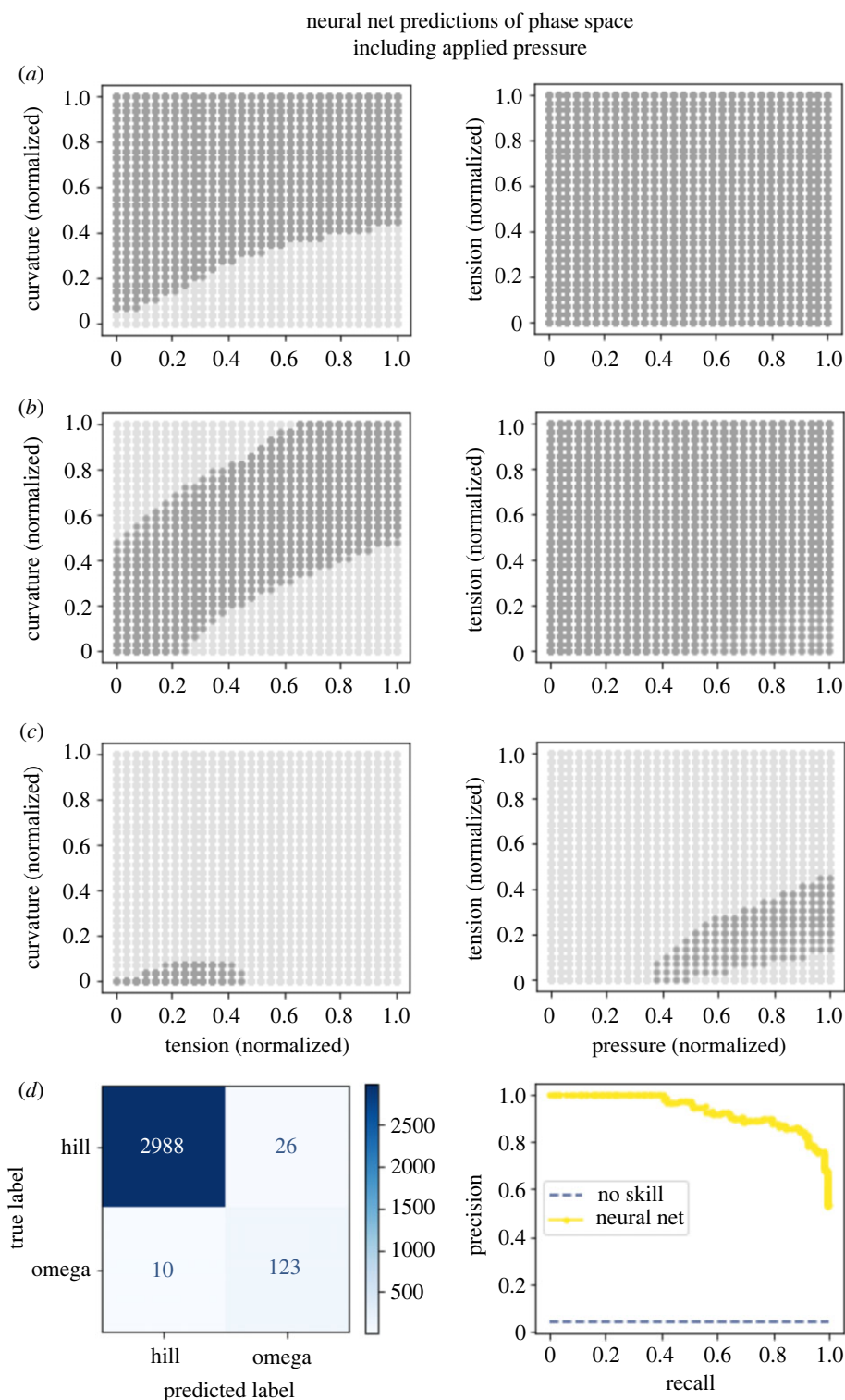
Figure 6. A neural network yields an accurate phase map of mechanical parameter space. The model recapitulates the general trends observed visually in figure 3. (a) Model predictions for patches with a normalized radius of 0.15. (b) Model predictions for patches with a normalized radius of 0.4. (c) Model predictions for patches with a normalized radius of 0.7. (d) The confusion matrix for this model with a threshold of 0.5 shows generally good performance and emphasizes the class imbalance of the dataset. The precision-recall plot for this model demonstrates strong performance, with an area under the curve of 0.93.

particular, is that conceptually we are not seeking to find the location of individual points but instead to trace a curve. In contrast to this conceptual goal, the regression framework we used is tasked to predict 50 'x' and 'y' locations along the curve, minimizing the distance between the predicted 'x' location and the actual 'x' location, and likewise for the 'y' location. While a custom loss function minimizing the distance to the curve itself could be defined, that would probably result in a model that accurately predicts the most stable portion of the membrane at the outermost edge while neglecting points along the region of the curve that varies the most (the bud region).

Embedded in the goal of predicting a two-dimensional curve is the need for a suitable metric of the similarity of the two curves to measure error. We used the MAE (averaged over all of the 'x' and 'y' coordinates) to create a loss function and to measure the model's performance. As we previously noted, this is an overestimate since the predicted point could be closer to the curve than it is to the target point. Additionally, we used the L1 norm, which yields an error
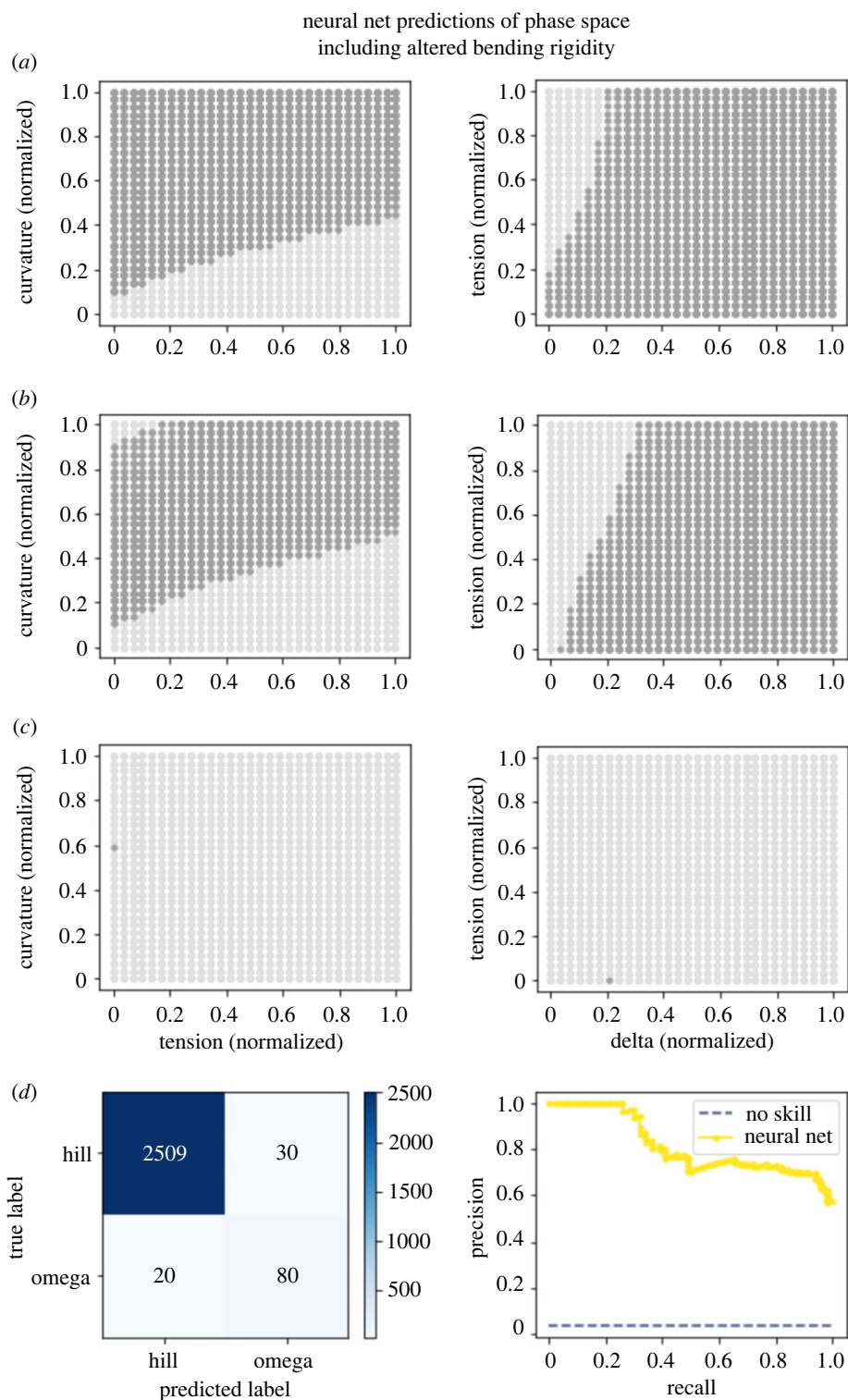
Figure 7. A neural network model is able to create a phase map of mechanical parameter space. (*a*) Model predictions for patches with a normalized radius of 0.15 where dark grey dots represent an omega class prediction and light grey represent a hill class prediction. (*b*) Model predictions for patches with a normalized radius of 0.4. (*c*) Model predictions for normalized radius of 0.7. (*d*) The confusion matrix for this model demonstrates that, while overall the model was able to correctly classify the small number of positive results, it had a small bias to identify false positives. The AUPRC was still relatively high at 0.82.

estimate greater than (or equal to) the L2 norm. Since the MAE takes size into account, it is possible that large MPs with a close shape match could have a greater error than small MPs where the shapes have a large discrepancy. Therefore, we supplemented our measurements of model performance with a second metric that first normalizes size based on the maximum 'x' and 'y' values of the data from the Helfrich model and then uses the L2 norm to calculate the distance to the target point. In designing the ML model, we maintained variation in the size of the membrane patch

rather than normalizing and tasking the model to predict shape alone since the mechanics of forming large MPs are different than small ones, as demonstrated in figures 3 and 4.

In figure 9*b*, we show the curves in the test set for which the model yielded the highest normalized error and MAE error for both omega-shaped and hill-shaped curves. We also display in figure 9*c* examples of mean error predictions. These curves yield several instructive insights. First, the ML model predicts results that may be unphysical, crossing over into negative 'x' values. The loss function could be
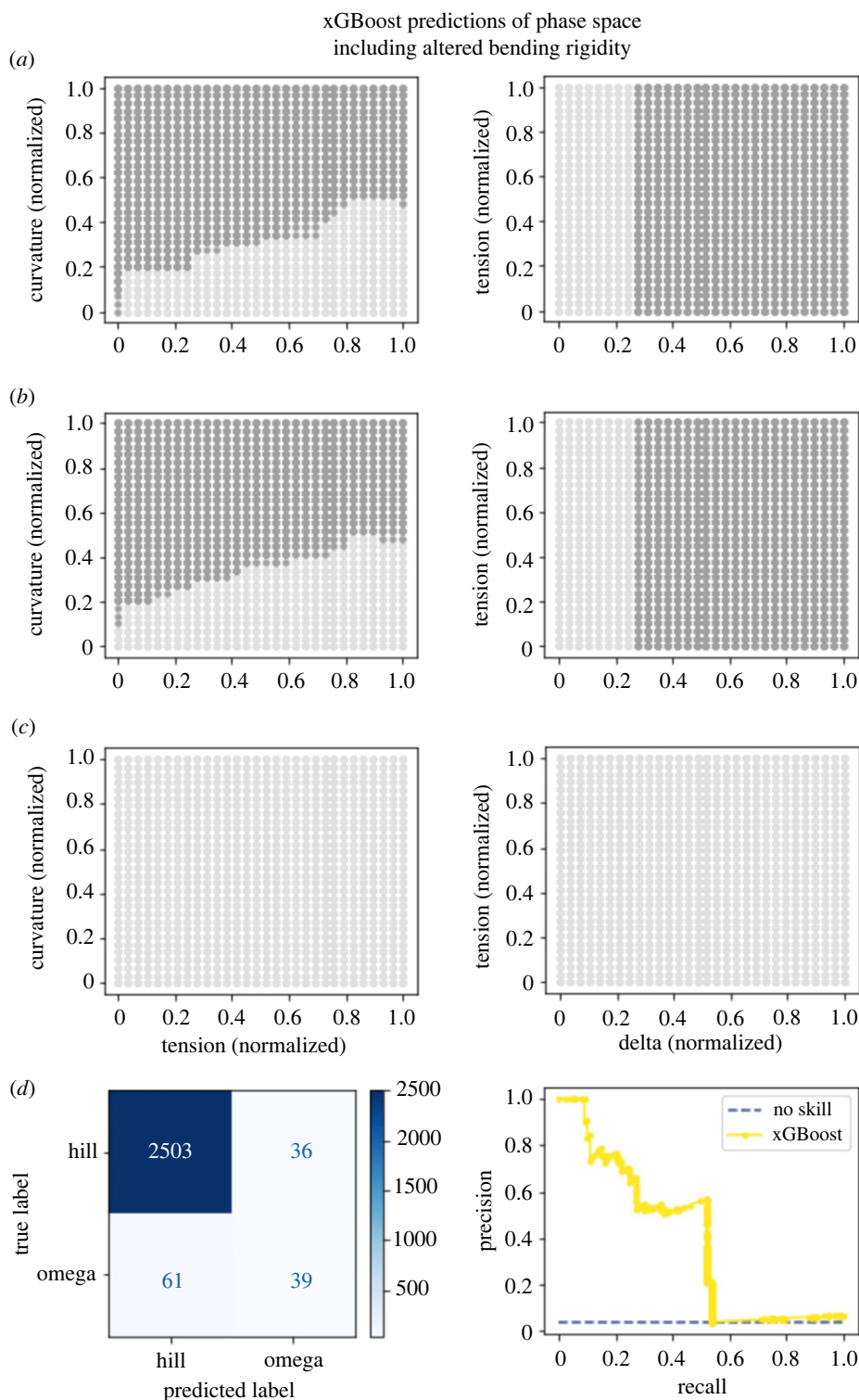
Figure 8. The xGBoost model makes a reasonable prediction of phase behaviour. (*a*) Model predictions for patches with a normalized radius of 0.15 where dark grey dots represent an omega class prediction and light grey represent a hill class prediction. (*b*) Model predictions for patches with a normalized radius of 0.4. (*c*) Model predictions for patches with a normalized radius of 0.7. (*d*) The confusion matrix for this model shows that the model had a false positive rate near the true positive rate, and a false negative rate larger than the true positive rate. Nonetheless, the model still had an AUPRC of 0.39.

further customized to penalize unphysical predictions like this. Next, we see that using a normalized error metric (blue) allowed a discrepancy in the prediction of a small deformation hill curve to be identified, but the loss function of the model does not penalize this highly since the absolute amplitude of the curve is relatively small. Finally, the highest MAE error omega prediction had an especially difficult task—the result of the Helfrich model for this parameter combination appears to be on the way towards a pearling morphology, which we observed very rarely. Probably,

there were very few (or no) curves in the training set that resembled this one, and so an ML model with no physical laws governing it could not predict that the membrane would adopt this shape. In figure 9*c*, we display the mean error ML predictions for both omega and hill-type curves measured with both MAE and normalized error. Hill-shaped curves were predicted with lower error than omega-shaped curves. Probably, this is due to the imbalance between hill-type and omega-type curves in the training set. These case studies are borne out across the test set, as shown by
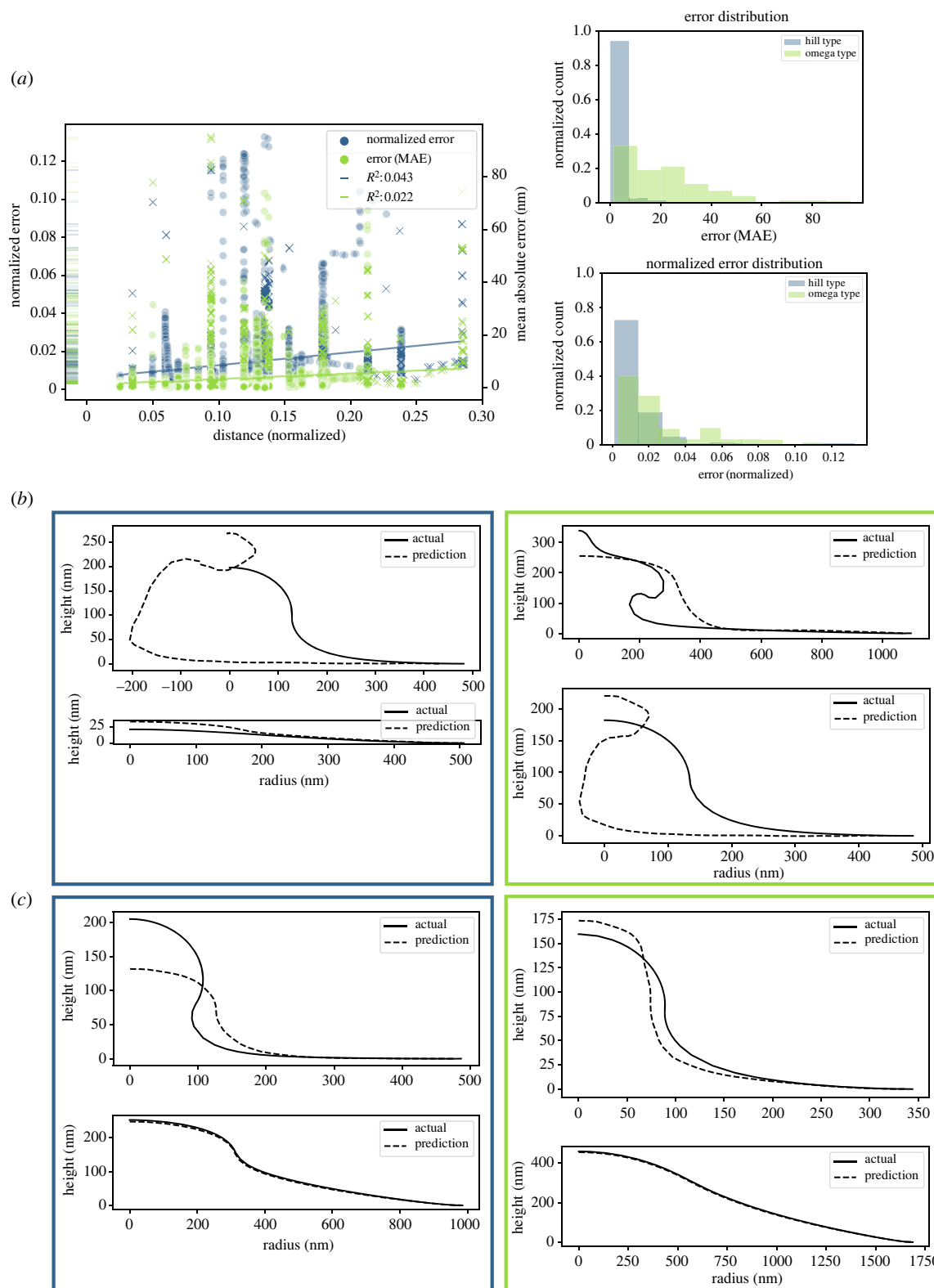
Figure 9. A neural net model tasked to perform regression to predict membrane shape performs decently well under two error metrics. (*a*). The error for a single curve is plotted against the normalized distance to the nearest point in mechanical parameter space that was included in the training set. Two error metrics were used—normalized error re-scaled the actual data between zero and one in both the *r* and *z* dimensions, and the predicted values were scaled based on the same values as the train data. This metric was used to assess shape change without factoring in scale. Also included is the mean absolute error. Regression on error versus distance shows only very weak correlations between the error of model's predictions and the distance to the nearest training point. The distribution of error is more spread out in MAE for omega-shaped buds, but not for normalized error. In (*b*), samples of curves that had the largest error predictions for normalized error (blue, left-hand column) and MAE (green, right-hand column) are shown. In (*c*), samples of curves that had mean error predictions for normalized error (blue, left-hand column) and MAE (green, right-hand column) are shown.

the wide spread of the omega-shaped curves' error distribution compared with the hill-shaped curves (figure 9*a*). Finally, we asked if the error in the ML model's predictions

depended on the distance between the test point in mechanical parameter space to the nearest train point. Performing a regression showed that distance did not explain much of
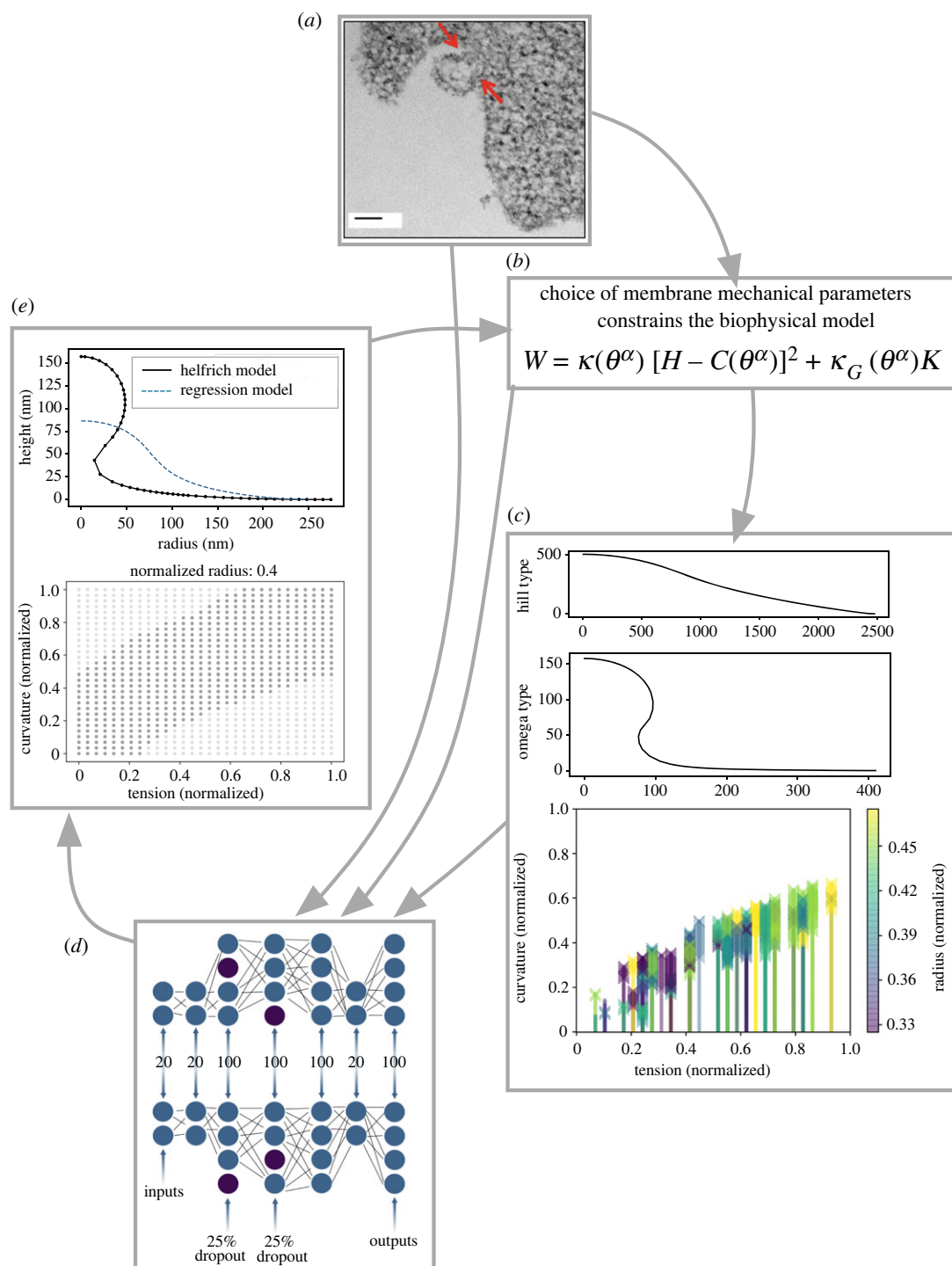
Figure 10. In the future, emerging techniques in ML could be combined with the Helfrich model to create a more robust and reasoned model, and potentially shed light on how observed membrane shapes relate to the mechanical characteristics of the membrane. (*a*) In this paper, we did not use experimentally documented curves (such as those that could be traced from data like that in [73]) to train our model, but we suggest that in the future they could be incorporated into a training dataset, in addition to inspiring the choice of biophysical modelling constraints. (*b*) The Helfrich continuum model can be used both to generate a synthetic dataset and could also be used to define the error of a machine-learning model's predictions. (*c*) Synthetic data can be used in training an ML model. (*d*) ML models that combine powerful physical insight like the Helfrich model with experimental data supplemented with synthetic data may yield more accurate predictions. (*e*) We hope for a future where machine-learning models may not only perform forward-predictive tasks, but may also use shapes to infer not only the forces needed to sustain them [5,6], but also the sets of mechanical parameters that may give rise to them.

the variation in error (MAE error had an R^2 of 0.043 and the normalized error had an R^2 of 0.022).

Although we must express the limitations of this regression framework, the model generally returned reasonable predictions and is a first step in this arena. Alternative modelling frameworks are being created that will probably lead to improved predictions. For instance, physics-informed neural nets (PINNs) use a custom loss function designed from known governing physics to inform if the model's predictions conform to governing physics and to adjust model weights in accordance with decreasing the model's loss [74], and are increasingly a tool applied to biological problems [75]. Another method merging machine learning with mathematical models is differentiable physics (for instance, see [76]). In the future, these avenues may create a rapid, robust model that links mechanical parameter space to shape changes of the membrane in

the forward direction. These methods could take better advantage of modern computing resources and the robust body of existing physical models and numerical methods. Additionally, the large volume of realistic geometries coming online from electron microscopy data may provide training and test datasets (for instance, the Cell Image Library (http://www.cellimageli brary.org/pages/about) and Janelia (https://www.janelia. org/)). Further, developments in physics-informed machine learning increasingly may allow inverse problems and parameter estimation tasks to be tackled. Eventually, these developments may allow the integration of small experimental datasets to infer sets of mechanical parameters that could give rise to observed shapes (figure 10).

4. Conclusion and future directions

In this paper, we developed two machine-learning models that demonstrate the feasibility of classifying membrane behaviour across mechanical parameter space and predicting membrane shape given mechanical parameters. The parameter spaces we explored are still relatively small (four dimensions). Going forward, if larger parameter spaces are to be explored, care must be taken due to the *Curse of Dimensionality* (for instance, see [72,77]), which means that as the number of dimensions grows, random sampling will preferentially explore the edges, rather than the core, of parameter space. A future avenue in need of exploration is to determine which parameters are the most important drivers of shape change in order to guide the selection of a reasonably sized space. Parametric sensitivity analysis is a tool that may help with narrowing down the subset of parameters. We also demonstrate optimistic performance of a

deep neural net trained to predict membrane shape from mechanical parameters. ML models like this one could eventually drastically reduce the time needed to visualize the results of the Helfrich model for a choice of parameters within the bounds described by the training dataset. This may benefit researchers since manual parameter tuning to match model results to biological phenomena can be very time-consuming. However, before this goal can be realized, lower error predictions would be desirable (particularly for shapes like omega buds that are only sparsely represented in parameter space). Methods such as PINNs and the merging of differentiable physics with neural nets may enable lower error predictions and harness the robust physical models available for membrane deformation. And, perhaps one of the most exciting open challenges in this field will be to use shapes drawn from experimental data to infer the sets of mechanical parameters that might give rise to observed forms.

References

1. Das Gupta A, Krawczynska N, Nelson ER. 2021 Extracellular vesicles—the next frontier in endocrinology. *Endocrinology* **162**, bqab133. (doi:10.1210/endocr/bqab133)

2. Freyssinet JM. 2003 Cellular microparticles: what are they bad or good for? *J. Thromb. Haemost.* **1**, 1655–1662. (doi:10.1046/j.1538-7836.2003.00309.x)

3. Alimohamadi H, Rangamani P. 2018 Modeling membrane curvature generation due to membrane–protein interactions. *Biomolecules* **8**, 120. (doi:10.3390/biom8040120)

4. Helfrich W. 1973 Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C* **28**, 693–703. (doi:10.1515/znc-1973-11-1209)

5. Alimohamadi H, Vasan R, Hassinger JE, Stachowiak JC, Rangamani P. 2018 The role of traction in membrane curvature generation. *Mol. Biol. Cell* **29**, 2024–2035. (doi:10.1091/mbc.E18-02-0087)

6. Lee HJ, Peterson EL, Phillips R, Klug WS, Wiggins PA. 2008 Membrane shape as a reporter for applied forces. *Proc. Natl Acad. Sci. USA* **105**, 19 253–19 257. (doi:10.1073/pnas.0806814105)

7. Rangamani P, Behzadan A, Holst M. 2021 Local sensitivity analysis of the 'membrane shape equation' derived from the Helfrich energy. *Math. Mech. Solids* **26**, 356–385. (doi:10.1177/1081286520953888)

8. Müller P, Schier AF. 2011 Extracellular movement of signaling molecules. *Dev. Cell* **21**, 145–158. (doi:10.1016/j.devcel.2011.06.001)

9. Hugel B, Martínez MC, Kunzelmann C, Freyssinet JM. 2005 Membrane microparticles: two sides of the coin. *Physiology* **20**, 22–27. (doi:10.1152/physiol.00029.2004)

10. Morel O, Morel N, Jesel L, Freyssinet JM, Toti F. 2011 Microparticles: a critical component in the nexus between inflammation, immunity, and thrombosis. In *Seminars in immunopathology*, vol. 33 (eds BO Roep, KT Coppieters, MG Herrath), pp. 469–486. New York, NY: Springer.

11. Van Niel G, Raposo G. 2018 Shedding light on the cell biology of extracellular vesicles. *Nat. Rev. Mol. Cell Biol.* **19**, 213. (doi:10.1038/nrm.2017.125)

12. Raposo G, Stoorvogel W. 2013 Extracellular vesicles: exosomes, microvesicles, and friends. *J. Cell Biol.* **200**, 373–383. (doi:10.1083/jcb.201211138)

13. Diamant M, Tushuizen ME, Sturk A, Nieuwland R. 2004 Cellular microparticles: new players in the field of vascular disease? *Eur. J. Clin. Invest.* **34**, 392–401. (doi:10.1111/j.1365-2362.2004.01355.x)

14. Herring J, McMichael M, Smith S. 2013 Microparticles in health and disease. *J. Vet. Intern. Med.* **27**, 1020–1033. (doi:10.1111/jvim.12128)

15. Thom SR et al. 2015 Association of microparticles and neutrophil activation with decompression sickness. *J. Appl. Physiol.* **119**, 427–434. (doi:10.1152/japplphysiol.00380.2015)

16. Thom SR, Milovanova TN, Bogush M, Bhopale VM, Yang M, Bushmann K, Pollock NW, Ljubkovic M, Denoble P, Dujic Z. 2012 Microparticle production, neutrophil activation, and intravascular bubbles following open-water SCUBA diving. *J. Appl. Physiol.* **112**, 1268–1278. (doi:10.1152/japplphysiol.01305.2011)

17. Di Credico A, Izzicupo P, Gaggi G, Di Baldassarre A, Ghinassi B. 2020 Effect of physical exercise on the release of microparticles with angiogenic potential. *Appl. Sci.* **10**, 4871. (doi:10.3390/app10144871)

18. Morel O, Jesel L, Freyssinet JM, Toti F. 2011 Cellular mechanisms underlying the formation of circulating microparticles. *Arterioscler. Thromb. Vasc. Biol.* **31**, 15–26. (doi:10.1161/ATVBAHA.109.200956)

19. Hirama T, Lu SM, Kay JG, Maekawa M, Kozlov MM, Grinstein S, Fairn GD. 2017 Membrane curvature induced by proximity of anionic phospholipids can initiate endocytosis. *Nat. Commun.* **8**, 1–14. (doi:10.1038/s41467-017-01554-9)

20. Kay JG, Fairn GD. 2019 Distribution, dynamics and functional roles of phosphatidylserine within the cell. *Cell Commun. Signal.* **17**, 1–8. (doi:10.1186/s12964-019-0438-z)

21. Tiberti ML, Antonny B, Gautier R. 2020 The transbilayer distribution of polyunsaturated phospholipids determines their facilitating effect on membrane deformation. *Soft Matter* **16**, 1722–1730. (doi:10.1039/C9SM02107H)

22. Jarsch IK, Daste F, Gallop JL. 2016 Membrane curvature in cell biology: an integration of molecular mechanisms. *J. Cell Biol.* **214**, 375–387. (doi:10.1083/jcb.201604003)

23. Del Conde I, Shrimpton CN, Thiagarajan P, López JA. 2005 Tissue-factor–bearing microvesicles arise from lipid rafts and fuse with activated platelets to initiate coagulation. *Blood* **106**, 1604–1611. (doi:10.1182/blood-2004-03-1095)

24. Stachowiak JC, Hayden CC, Sasaki DY. 2010 Steric confinement of proteins on lipid membranes can drive curvature and tubulation. *Proc. Natl Acad. Sci. USA* **107**, 7781–7786. (doi:10.1073/pnas.0913306107)

25. Shurer CR et al. 2019 Physical principles of membrane shape regulation by the glycocalyx. *Cell* **177**, 1757–1770. (doi:10.1016/j.cell.2019.04.017)

26. Bevers EM, Williamson PL. 2010 Phospholipid scramblase: an update. *FEBS Lett.* **584**, 2724–2730. (doi:10.1016/j.febslet.2010.03.020)

27. Rubin O, Canellini G, Delobel J, Lion N, Tissot JD. 2012 Red blood cell microparticles: clinical relevance. *Transfus. Med. Hemother.* **39**, 342–347. (doi:10.1159/000342228)

28. Pasquet JM, Dachary-Prigent J, Nurden AT. 1996 Calcium influx is a determining factor of calpain activation and microparticle formation in platelets. *Eur. J. Biochem.* **239**, 647–654. (doi:10.1111/j.1432-1033.1996.0647u.x)

29. Randriamboavonjy V, Fleming I. 2012 All cut up! The consequences of calpain activation on platelet function. *Vascul. Pharmacol.* **56**, 210–215. (doi:10.1016/j.vph.2012.02.009)

30. O'Connell DJ, Rozenvayn N, Flaumenhaft R. 2005 Phosphatidylinositol 4, 5-bisphosphate regulates activation-induced platelet microparticle formation. *Biochemistry* **44**, 6361–6370. (doi:10.1021/bi047344c)

31. Hassinger JE, Oster G, Drubin DG, Rangamani P. 2017 Design principles for robust vesiculation in clathrin-mediated endocytosis. *Proc. Natl Acad. Sci. USA* **114**, E1118–E1127. (doi:10.1073/pnas.1617705114)

32. Charras GT, Coughlin M, Mitchison TJ, Mahadevan L. 2008 Life and times of a cellular bleb. *Biophys. J.* **94**, 1836–1853. (doi:10.1529/biophysj.107.113605)

33. Tinevez JY, Schulze U, Salbreux G, Roensch J, Joanny JF, Paluch E. 2009 Role of cortical tension in bleb growth. *Proc. Natl Acad. Sci. USA* **106**, 18581–18586. (doi:10.1073/pnas.0903353106)

34. Fang C, Hui T, Wei X, Shao X, Lin Y. 2017 A combined experimental and theoretical investigation on cellular blebbing. *Sci. Rep.* **7**, 1–11.

35. Sens P, Turner MS. 2006 Budded membrane microdomains as tension regulators. *Phys. Rev. E* **73**, 031918. (doi:10.1103/PhysRevE.73.031918)

36. Lipowsky R. 1993 Domain-induced budding of fluid membranes. *Biophys. J.* **64**, 1133–1138. (doi:10.1016/S0006-3495(93)81479-6)

37. Steigmann D. 1999 Fluid films with curvature elasticity. *Arch. Ration. Mech. Anal.* **150**, 127–152. (doi:10.1007/s002050050183)

38. Rangamani P, Mandadap KK, Oster G. 2014 Protein-induced membrane curvature alters local membrane tension. *Biophys. J.* **107**, 751–762. (doi:10.1016/j.bpj.2014.06.010)

39. Rangamani P. 2022 The many faces of membrane tension: challenges across systems and scales. *Biochim. Biophys. Acta* **1864**, 183897. (doi:10.1016/j.bbamem.2022.183897)

40. Hochmuth F, Shao JY, Dai J, Sheetz MP. 1996 Deformation and flow of membrane into tethers extracted from neuronal growth cones. *Biophys. J.* **70**, 358–369. (doi:10.1016/S0006-3495(96)79577-2)

41. Sitarska E, Diz-Muñoz A. 2020 Pay attention to membrane tension: mechanobiology of the cell surface. *Curr. Opin Cell Biol.* **66**, 11–18. (doi:10.1016/j.ceb.2020.04.001)

42. Sens P, Plastino J. 2015 Membrane tension and cytoskeleton organization in cell motility. *J. Phys.: Condens. Matter* **27**, 273103.

43. Dai J, Sheetz MP. 1999 Membrane tether formation from blebbing cells. *Biophys. J.* **77**, 3363–3370. (doi:10.1016/S0006-3495(99)77168-7)

44. Evans E, Kukan B. 1984 Passive material behavior of granulocytes based on large deformation and recovery after deformation tests. *Blood* **64**, 1028–1035.

45. Needham D, Hochmuth R. 1992 A sensitive measure of surface stress in the resting neutrophil. *Biophys. J.* **61**, 1664–1670. (doi:10.1016/S0006-3495(92)81970-7)

46. Phillips R. 2018 Membranes by the numbers. In *Physics of biological membranes* (eds P Bassereau, P Sens), pp. 73–105. New York, NY: Springer.

47. Dimova R. 2014 Recent developments in the field of bending rigidity measurements on membranes. *Adv. Colloid Interface Sci.* **208**, 225–234. (doi:10.1016/j.cis.2014.03.003)

48. Monzel C, Sengupta K. 2016 Measuring shape fluctuations in biological membranes. *J. Phys. D: Appl. Phys.* **49**, 243002. (doi:10.1088/0022-3727/49/24/243002)

49. Faizi HA, Reeves CJ, Georgiev VN, Vlahovska PM, Dimova R. 2020 Fluctuation spectroscopy of giant unilamellar vesicles using confocal and phase contrast microscopy. *Soft Matter* **16**, 8996–9001. (doi:10.1039/D0SM00943A)

50. Evans E, Rawicz W. 1990 Entropy-driven tension and bending elasticity in condensed-fluid membranes. *Phys. Rev. Lett.* **64**, 2094. (doi:10.1103/PhysRevLett.64.2094)

51. Henriksen J, Rowat AC, Ipsen JH. 2004 Vesicle fluctuation analysis of the effects of sterols on membrane bending rigidity. *Eur. Biophys. J.* **33**, 732–741. (doi:10.1007/s00249-004-0420-5)

52. Steinkühler J, Sezgin E, Urbančič I, Eggeling C, Dimova R. 2019 Mechanical properties of plasma membrane vesicles correlate with lipid order, viscosity and cell density. *Commun. Biol.* **2**, 1–8. (doi:10.1038/s42003-019-0583-3)

53. Stachowiak JC, Brodsky FM, Miller EA. 2013 A cost-benefit analysis of the physical mechanisms of membrane curvature. *Nat. Cell Biol.* **15**, 1019–1027. (doi:10.1038/ncb2832)

54. Chengappa P, Sao K, Jones TM, Petrie RJ. 2018 Intracellular pressure: a driver of cell morphology and movement. *Int. Rev. Cell Mol. Biol.* **337**, 185–211.

55. Auddya D, Zhang X, Gulati R, Vasan R, Garikipati K, Rangamani P, Rudraraju S. 2021 Biomembranes undergo complex, non-axisymmetric deformations governed by Kirchhoff–Love kinematicsand revealed by a three-dimensional computational framework. *Proc. R. Soc. A* **477**, 20210246. (doi:10.1098/rspa.2021.0246)

56. Zhu C, Lee CT, Rangamani P. 2022 Mem3DG: modeling membrane mechanochemical dynamics in 3D using discrete differential geometry. *Biophys. J.* **121**, 71a. (doi:10.1016/j.bpj.2021.11.2371)

57. Vasan R, Rudraraju S, Akamatsu M, Garikipati K, Rangamani P. 2020 A mechanical model reveals that non-axisymmetric buckling lowers the energy barrier associated with membrane neck constriction. *Soft Matter* **16**, 784–797. (doi:10.1039/C9SM01494B)

58. Singh A, Thakur N, Sharma A. 2016 A review of supervised machine learning algorithms. In *2016 3rd Int. Conf. on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315. IEEE.

59. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. 2002 SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357. (doi:10.1613/jair.953)

60. Han H, Wang WY, Mao BH. 2005 Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Int. Conf. on Intelligent Computing*, pp. 878–887. New York, NY: Springer.

61. Nguyen HM, Cooper EW, Kamei K. 2011 Borderline over-sampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigms* **3**, 4–21. (doi:10.1504/IJKESDP.2011.039875)

62. Brownlee J. 2020 *SMOTE for imbalanced classification with Python*. See https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

63. Brownlee J. 2020 *How to configure XGBoost for imbalanced classification*. See https://www.tensorflow.org/tutorials/structured_data/imbalanced_data (accessed 2022).

64. Tensor Flow Core. 2022 *Classification on imbalanced data*. See https://www.tensorflow.org/tutorials/structured_data/imbalanced_data.

65. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014 Dropout: a simple way to

prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958.

66. Brownlee J. 2018 *Use early stopping to halt the training of neural networks at the right time.* See https://machinelearningmastery.com/how-to-stop-training-deep-neuralnetworks-at-the-right-time-using-early-stopping/ (accessed 2022).

67. Chen T, Guestrin C. 2016 Xgboost: A scalable tree boosting system. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Francisco, CA, 13–17 August*, pp. 785–794.

68. Putatunda S, Rama K. 2018 A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost. In *Proc. of the 2018 Int. Conf. on Signal Processing and Machine Learning, Shanghai, China, 28–30 November*, pp. 6–10. New York, NY: Association for Computing Machinery.

69. Kavzoglu T, Teke A. 2022 Advanced hyperparameter optimization for improved spatial prediction of

shallow landslides using extreme gradient boosting (XGBoost). *Bull. Eng. Geol. Environ.* **81**, 1–22. (doi:10.1007/s10064-022-02708-w)

70. Qin C, Zhang Y, Bao F, Zhang C, Liu P, Liu P. 2021 XGBoost optimized by adaptive particle swarm optimization for credit scoring. *Math. Probl. Eng.* **2021**.

71. Brownlee J. 2018 *How to Use ROC Curves and Precision-Recall Curves for Classification in Python*. See https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.

72. Aggarwal CC, Hinneburg A, Keim DA. 2001 On the surprising behavior of distance metrics in high dimensional space. In *Int. Conf. on Database Theory*, pp. 420–434. New York, NY: Springer.

73. Hafiane A, Genest J. 2017 ATP binding cassette A1 (ABCA1) mediates microparticle formation during high-density lipoprotein (HDL) biogenesis.

Atherosclerosis **257**, 90–99. (doi:10.1016/j.atherosclerosis.2017.01.013)

74. Raissi M, Perdikaris P, Karniadakis GE. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707. (doi:10.1016/j.jcp.2018.10.045)

75. Yazdani A, Lu L, Raissi M, Karniadakis GE. 2020 Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS Comput. Biol.* **16**, e1007575. (doi:10.1371/journal.pcbi.1007575)

76. Thuerey N, Holl P, Mueller M, Schnell P, Trost F, Um K. 2021 *Physics-based Deep Learning*. See https://physicsbaseddeeplearning (accessed 2021).

77. Verleysen M, François D. 2005 The curse of dimensionality in data mining and time series prediction. In *Int. Work—Conf. on Artificial Neural Networks*, pp. 758–770. New York, NY: Springer.