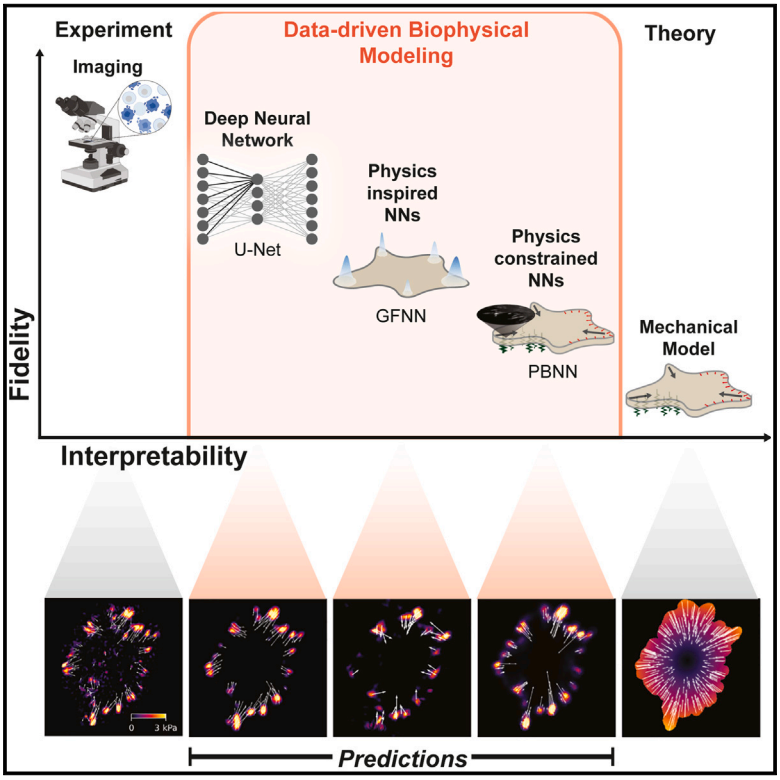
Cell

Machine learning interpretable models of cell mechanics from protein images

Graphical abstract



Authors

Matthew S. Schmitt, Jonathan Colen, Stefano Sala, ..., Margaret L. Gardel, Patrick W. Oakes, Vincenzo Vitelli

Correspondence

gardel@uchicago.edu (M.L.G.), poakes@luc.edu (P.W.O.), vitelli@uchicago.edu (V.V.)

In brief

Image-based deep learning approaches can predict the mechanical behavior of cells from experimental images of a single focal adhesion protein. These generalizable models of cellular forces can be used to advance understanding and control of cell adhesion.

Highlights

- Neural networks predict traction forces from a single FA protein field
- Networks generalize to unseen cells, cell types, and biochemical perturbations
- Both physics-constrained and agnostic methods learn interpretable rules to predict forces
- Traction forces are encoded across two length scales relating to FAs and cell shape

Schmitt et al., 2024, Cell *187*, 481–494
January 18, 2024
https://doi.org/10.1016/j.cell.2023.11.041

CellPress

Cell

CellPress

Theory

Machine learning interpretable models of cell mechanics from protein images

Matthew S. Schmitt,[1,2,3,5] Jonathan Colen,[1,2,3,5] Stefano Sala,[4] John Devany,[1,2] Shailaja Seetharaman,[1,2] Alexia Caillier,[4] Margaret L. Gardel,[1,2,*] Patrick W. Oakes,[4,*] and Vincenzo Vitelli[1,2,3,6,*]

[1]James Franck Institute, University of Chicago, Chicago, IL 60637, USA
[2]Department of Physics, University of Chicago, Chicago, IL 60637, USA
[3]Kadanoff Center for Theoretical Physics, University of Chicago, Chicago, IL 60637, USA
[4]Department of Cell & Molecular Physiology, Stritch School of Medicine, Loyola University Chicago, Maywood, IL 60153, USA
[5]These authors contributed equally
[6]Lead contact
*Correspondence: gardel@uchicago.edu (M.L.G.), poakes@luc.edu (P.W.O.), vitelli@uchicago.edu (V.V.)
https://doi.org/10.1016/j.cell.2023.11.041

SUMMARY

Cellular form and function emerge from complex mechanochemical systems within the cytoplasm. Currently, no systematic strategy exists to infer large-scale physical properties of a cell from its molecular components. This is an obstacle to understanding processes such as cell adhesion and migration. Here, we develop a data-driven modeling pipeline to learn the mechanical behavior of adherent cells. We first train neural networks to predict cellular forces from images of cytoskeletal proteins. Strikingly, experimental images of a single focal adhesion (FA) protein, such as zyxin, are sufficient to predict forces and can generalize to unseen biological regimes. Using this observation, we develop two approaches—one constrained by physics and the other agnostic—to construct data-driven continuum models of cellular forces. Both reveal how cellular forces are encoded by two distinct length scales. Beyond adherent cell mechanics, our work serves as a case study for integrating neural networks into predictive models for cell biology.

INTRODUCTION

The structure and dynamics of living cells are controlled by the physical properties of the cytoskeleton.[1,2] The cytoskeleton itself, however, is the product of complex biochemical circuits that regulate its dynamics and spatial organization.[3,4] The central challenge faced when studying the physical biology of the cell is to untangle this interplay between physics and biochemistry. Current modeling approaches lean heavily on intuition built upon centuries of work on classical continuum mechanics, where symmetries and conservation laws dictate both the variables that arise in such models, as well as the equations they obey.[5,6] Cells, however, are decidedly non-classical, relying instead on distributed enzymatic activity and non-equilibrium mechanochemical processes across a hierarchy of scales.[7,8] For example, forces in cells arise not only to restore local deformations, but also as a result of continuous remodeling regulated by biochemical signaling networks. This intertwined action of physics and biochemistry complicates coarse-graining and system parameterization of cell dynamics in terms of a few simply understood collective variables.[9,10]

Machine learning (ML) has the potential to overcome this challenge by augmenting existing physical models with biochemical information and even discovering new ones directly from the statistics of data.[11–14] These tools have proven very successful in structural biology for predicting protein structures directly from gene sequences.[15,16] Here, we illustrate the power of ML approaches in a classic cellular biology problem: how cytoskeletal proteins govern the mechanics of cells. Cells generate contractile forces, which are critical regulators of cell shape, adhesion, motility, and mechanotransduction.[17,18] Forces generated in the actin cytoskeleton are transmitted via transmembrane focal adhesions (FAs) to the extracellular matrix[19–21] where they can be measured directly with techniques like traction force microscopy (TFM).[22–24] TFM measurements coupled with live-cell imaging of fluorescently tagged cytoskeletal proteins have helped develop a number of biophysical models of cellular force generation and mechanosensing.[25–32] While providing insight into various local microscopic mechanisms, these models do not capture the broad heterogeneity of structures and behaviors in cells. As a result, they cannot fully account for how non-local and cell-scale properties such as cell morphology and FA structure and location affect, and even dominate, local forces.

In this work, we demonstrate how to harness the flexibility of neural networks to both improve existing models of cellular forces as well as discover new ones. We begin by training deep neural networks to predict forces directly from images of fluorescent cytoskeletal proteins, and in the process, we discover that a single FA protein, such as zyxin or paxillin, is sufficient to predict traction stresses. The ability to make accurate

predictions with only a single protein distribution does not imply that other proteins are biochemically redundant for force generation. Rather, it suggests the minimum amount of information, and hence minimal complexity of models, needed to predict the magnitude and orientation of cellular forces. These predictions are robust, as we find that the networks can generalize to previously unseen experimental and biological perturbations. To understand this generalizability, we probe the neural network to identify features that inform its predictions and further guide the formulation of two complementary mathematical models. First, we introduce a physics-constrained ML approach that augments existing mechanical cellular models[31–34]: it learns directly from data how a single, measured protein distribution sets the physically meaningful parameters of an effective linear-elastic model. Second, we cast away our mechanical hypotheses and demonstrate a purely data-driven pipeline that constructs relevant fields and distills effective equations, which predict cellular traction stresses. Despite incorporating varying degrees of model complexity and prior knowledge, all our approaches consistently reveal that models for force generation are characterized by the interaction of both local and non-local features. Our findings illustrate how FA proteins encode information of local forces at adhesion sites, as well as whole-cell contractility through their distribution in the cell, and demonstrate a suite of complementary approaches to build novel models of living systems.

RESULTS

Neural networks accurately predict traction forces from images of a single protein

To assess whether neural networks could make mechanical predictions from biochemical fields, we created a library by pairing fluorescence microscopy images of the FA protein zyxin in fibroblasts[35] (Figure 1A) with their corresponding traction forces as directly measured by TFM ($\vec{F}_{exp}$; Figures 1B and 1C). In total, our library contained images obtained from 31 separate time lapses of cells expressing zyxin and their associated traction force fields (see STAR Methods for details). In each of these cells, traction forces primarily localized along the cell boundary at FAs, as marked by zyxin accumulation, and pointed inward toward the cell body (Figures 1C and 1D). For our neural network, we chose a U-Net architectural backbone that learns large-scale features via successive strided convolutions, while skip connections between layers propagate fine-grained information and preserve local structure that may be lost during coarse-graining[36] (Figure 1A). While U-Nets have proven successful at solving the TFM inverse problem,[37,38] here, we push them to link biochemistry and mechanics. We augment this backbone with ConvNext blocks to improve accuracy and training efficiency[39] (Figure S1; see STAR Methods and supplemental information for architecture details). We trained the U-Net to directly predict traction forces, using the library of paired zyxin images as inputs ($\vec{F}_{NN}$; Figures 1C and 1D). The library was split into training and test sets containing 16 and 15 cells, respectively (Figure 1G). Although the U-Net was taught using only the training set, it learned to generalize and was able to accurately predict traction forces in cells from the test set, which it had never seen before (Figures 1C and 1D). The network predictions of traction forces agreed generally with experimental measurements in both location and magnitude (Figure 1C), with some smoothing occurring at the micron scale (Figure 1D).

To evaluate the U-Net accuracy, we compared the experimentally measured traction force directions (α_{exp}) and magnitudes ($|F_{exp}|$) with those predicted by the U-Net for all the cells in the test set (α_{NN}, $|F_{NN}|$). Figures 1E and 1F show the conditional distributions $p(\alpha_{exp}|\alpha_{NN})$ and $p(|F_{exp}||F_{NN}|)$ (see STAR Methods for additional details) along with the averages (solid line) and standard deviation (dotted line). The neural network achieves near-optimal accuracy for force angles as well as magnitudes up to ~4 kPa, which represents ~99.9% of pixels in the test dataset (Figures 1E, 1F, and S3). To evaluate the neural network's sensitivity to the test and train data used, we generated 22 random partitions of our 31-cell library into 16-cell training sets and 15-cell test sets. We trained a separate U-Net on each partition and evaluated the mean-squared error (MSE) of the force predictions (Figures 1H, and S3). The network performance varies weakly, depending on the cells present in the test and train set, with the MSE fluctuating by less than $\pm 5\%$ across test cells. As a comparison, the MSE varies by $\pm 2\%$ between cells measured on different days (Figure S3), denoted by D_i in Figure 1H. The network's sensitivity to training and testing data is thus similar in magnitude to systematic variations, which arise from differences in experimental preparation, rather than from fundamental differences between cells. Together, these results demonstrate that from a readily achievable amount of experimental data, a U-Net can robustly learn to make accurate predictions of traction forces from fluorescent images of a single FA protein, such as zyxin.

Zyxin-trained networks outperform other cytoskeletal proteins

In addition to identifying FAs, zyxin also reveals information about actin stress fiber organization and general cell geometry.[40] To determine which of these features was driving the U-Net performance, we tested the efficacy of other cytoskeletal proteins involved in force transmission: actin and myosin, the filaments and motors that make up the contractile network; paxillin, another FA protein; mitochondria, an organelle unconnected to the contractile machinery as a negative control; and binary masks of the cell morphology. For these experiments, we simultaneously expressed zyxin with the other proteins of interest. Except for mitochondria-trained networks, all networks learned to predict forces with some degree of accuracy, capturing the general localization and magnitude of traction stresses (Figure 2A). The probability distribution of angular error $\Delta\alpha = \alpha_{NN} - \alpha_{exp}$ peaked around zero for all proteins, differing only in the width of the distribution about the true value (Figure 2B). This distribution width was similar for networks trained on each protein, except for mitochondria that showed a high angular variance (Figure 2C). When comparing force magnitude predictions, we observed larger differences among the proteins, with the FA proteins zyxin and paxillin outperforming all others (Figures 2D and S4). Surprisingly, training networks on combined inputs of zyxin and these proteins did not improve performance, and they performed as well as a U-Net trained on zyxin alone (Figures 2E and S4). Similarly, combining paxillin and actin also

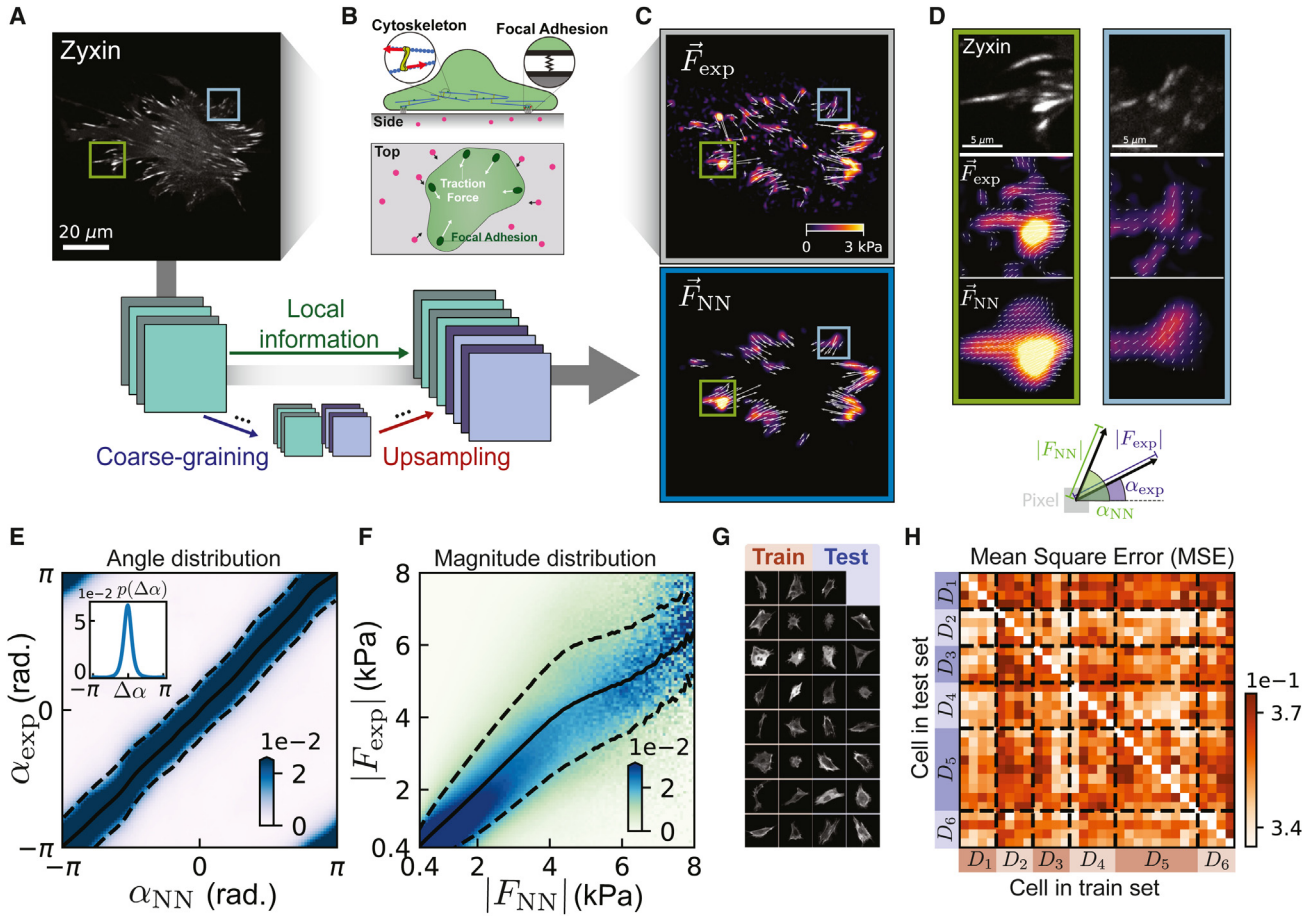


Figure 1. Neural networks accurately predict forces from images of a single protein

(A) Fluorescent protein intensities (e.g., EGFP-zyxin) are measured in cells spread on 2D polyacrylamide gels coated with fibronectin.

(B) Adherent cells generate forces via the contractile activity of the cytoskeleton. These traction forces are transmitted to the substrate through focal adhesions (FAs). By measuring the displacement of fluorescent beads embedded in the substrate (red circles), the traction forces can be reconstructed using traction force microscopy (TFM; see STAR Methods).

(C) (Top) Forces ($\vec{F}_{exp}$) recovered from experimental measurements of substrate deformations via TFM. (Bottom) U-Nets predict traction forces ($\vec{F}_{NN}$) from images of protein intensity. In both plots, the magnitude of the traction force is indicated by the color and the direction by the overlaid arrow.

(D) Zoomed-in view of colored boxes in (C).

(E and F) At each pixel, we measure $(\alpha_{exp}, \alpha_{NN}, |F_{exp}|, |F_{NN}|)$, which we bin to calculate the conditional angular distribution $p(\alpha_{exp}|\alpha_{NN})$ (E) and the conditional magnitude distribution $p(|F_{exp}||F_{NN}|)$ (F). An optimal predictor lies exactly along the diagonal. Solid lines denote the average of the distribution, while dashed lines mark one standard deviation. The angular distribution is strongly peaked along this diagonal (with additional peaks appearing due to periodicity), while the magnitude distribution remains on the diagonal up to $|F_{exp}| \approx 4$ kPa, which corresponds to 99.9% of pixels. Inset of (E) shows the probability distribution of angular error $\Delta\alpha = \alpha_{NN} - \alpha_{exp}$.

(G) Partition of 31-cell dataset into 16-cell training set and 15-cell test set. Every cell shown in this paper is in the test set and was not seen during training.

(H) Model mean-squared error (MSE) for 22 random train/test partitions. Dashed lines denote days on which cells were imaged. Pixel color p_{ij} is the average MSE of all models that use cell *i* for training and cell *j* for testing.

did not perform as well as zyxin alone (Figure 2E). Although the cellular forces themselves are generated by many interacting proteins, a single FA protein is sufficient to serve as a proxy for this microscopic complexity and contains enough information to determine the coarse-grained mechanical behavior. These results demonstrate that neural networks can be used to sort through potentially relevant proteins and identify a minimal subset that contains all the necessary information about the cell to predict forces. We proceeded using our highest-performing neural network, which was trained using zyxin alone.

Zyxin-trained networks generalize to new cell types and biological perturbations

While it is generally assumed that the underlying mechanics of contraction are universal,[18] we sought to explicitly test this by evaluating our U-Net (which was trained on images of fibroblasts) on images of other adherent cell types. Specifically, we imaged zyxin in individual human osteosarcoma epithelial cells (U2OS; Figure 3A) and paxillin in colonies of canine epithelial cells (MDCK; Figures 3D and S5). Without any retraining, the zyxin-trained U-Net generally predicted accurate traction force

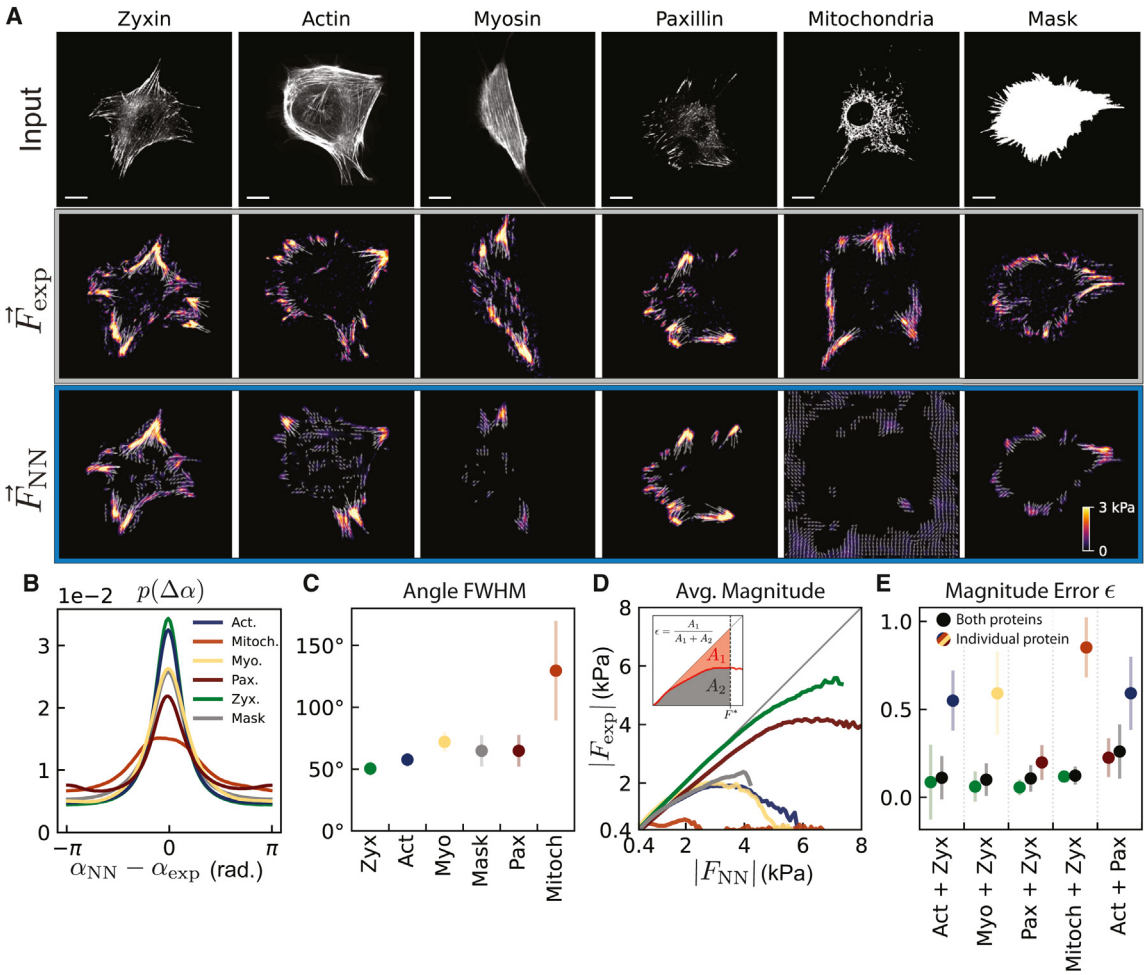


Figure 2. Zyxin-trained networks outperform other cytoskeletal proteins

(A) The predictive power of different cellular proteins inputs are compared by training neural networks on each protein individually. The comparison includes cytoskeletal proteins associated with force generation (actin, myosin), FA proteins (paxillin, zyxin), as well as a protein thought to be unrelated to force generation (mitochondria) and the binary cell mask. Scale bars are 20 μm.

(B) While all networks trained in (A) predict accurate force directions on average, the distribution of errors varies depending on protein.

(C) We quantify the angular error by the full-width half-maximum (FWHM) value of the distributions in (B). The networks performed similarly, except for the mitochondria network that showed a much larger FWHM. Error bars denote standard deviations of the error across different cells.

(D) NNs trained on focal adhesion proteins, in particular zyxin, predict force magnitudes more accurately than those trained on other inputs. Inset shows calculation of magnitude error, which measures the cumulative distance from the diagonal up to $F^* = 6$ (black dashed line).

(E) Zyxin outperforms all other proteins in predicting force magnitudes, and training on zyxin plus other proteins does not improve performance. Error bars denote standard deviations as in (C). One cell in the actin dataset was an outlier and was excluded (see STAR Methods and Figure S4 for details).

directions and magnitudes for both new cell types (Figures 3B, 3C, 3E, and 3F), which were comparable to differences between training on different cytoskeletal proteins (Figure S6). This was true despite the MDCK data being taken on a softer substrate (2.8 vs. 16 kPa shear modulus) using a different microscope. While these changes in experimental setup can induce small errors (see Figures S7, S8, and S9), the ability of the network to generalize to different cell types, adhesion proteins, and cell clusters suggests that it has learned some underlying general law governing traction force generation.

To probe this idea further, we next challenged our U-Net model to make predictions in response to a biochemical perturbation. We imaged cells for 30 min at a basal contractile state before adding 5 μM of the Rho-kinase (ROCK) inhibitor Y-27632 for 45 min and then washing out the drug and imaging for a final 45 min (Figures 3G and 3H). Adding Y-27632 resulted in a drop in traction forces, an increase in overall cell area, and a reduction in the size of FAs, as expected,[41,42] while the wash-out reversed each of these trends. Despite having never seen these drug perturbations, the network still predicted the overall changes in global traction forces (Figure 3G) and the local changes at FAs (Figure 3H) during both the drug treatment and the subsequent recovery following wash-out. Together, these results indicate that the distribution of zyxin alone is a faithful proxy for the mechanical state of a cell and is sufficient to predict traction forces under a wide variety of conditions.

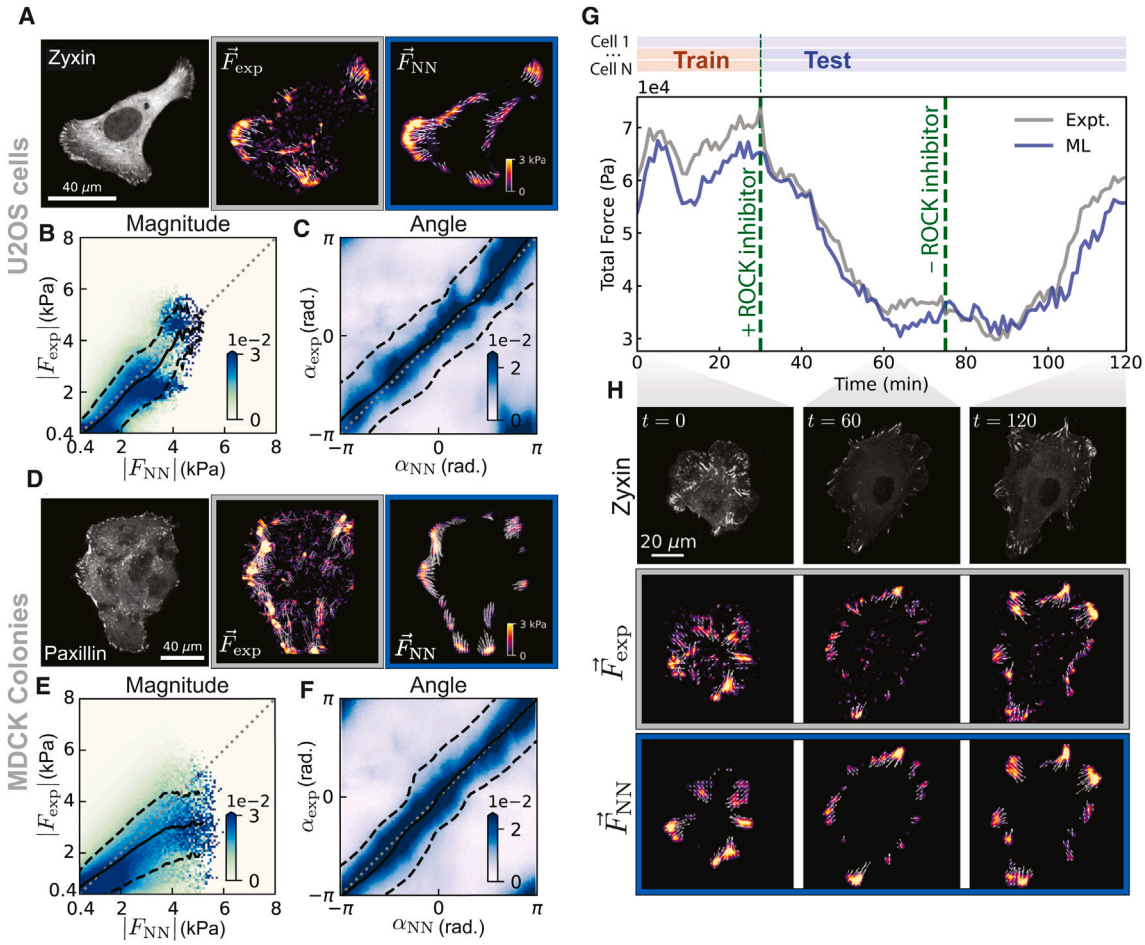


Figure 3. Zyxin-trained networks generalize to new cell types and biological perturbations

(A–G) The fibroblast-trained network of Figure 1 is evaluated on (A–C) individual U2OS cells expressing zyxin, and (D–F) colonies of MDCK cells expressing paxillin. Pixel-wise distributions of angle and magnitude predictions for U2OS cells (B and C) and MDCK cells (E and F), as in Figure 1. (G) The same network, which was trained only on fibroblasts in their basal contractile state, is evaluated on fibroblasts perturbed with the ROCK inhibitor Y-27632. The wash-in at t = 30 min impairs cytoskeletal contractility resulting in lower total force, which recovers after the drug is washed out at t = 75 min.

(H) Three snapshots from the time series in (G) demonstrate the NN's ability to capture redistribution of forces seen during the perturbation.

Neural networks identify features of cell adhesion and morphology

Motivated by the success of the U-Net at predicting traction forces, we next sought to identify features of the zyxin distributions that are relevant for making those predictions. Zyxin encodes both micron-scale structures, such as FAs ($\sim$ 1 μm), as well as cell-scale structures like stress fibers $\sim$ 10–100 μm (Figure 1A). To probe how the network interprets these features, we trained U-Nets on random image crops of sizes ranging from 10 up to 130 μm in our input data (Figure 4A). Even when trained on only a small fraction of the cell, these networks learned models that were accurate on average for both force magnitude (Figures 4B and 4C) and direction (Figures 4D and 4E). In both of these measurements, improvements in the prediction accuracy was negligible as the input size increased beyond ~25 μm (Figures 4C and 4E). This indicates that the U-Net does not need to know the whole-cell geometry and that it can make accurate predictions by considering a smaller neighborhood around any given point.

Previous work has suggested that both cell morphology[30–32,43–45] and FA distribution[46–49] can impact force generation. To understand how the U-Nets interpreted these features, we generated synthetic "cells" to systematically vary these features and examine the trained models' response.[50] To probe the role of cell morphology, we evaluated the mask-trained U-Net on cells that were triangular in shape with a width L and whose edges were arcs with radius of curvature R_c (Figure 4F). While the network did not systematically respond to increases in cell edge curvature (Figure 4H), we did find that force production increased with total cell size (Figure 4I). This result is consistent with previous work showing that force generation scales with cell area,[32] and it further demonstrates that the network is sensitive to large-scale features of cell geometry.

To probe the role of FA-like features, we created synthetic cells composed of elliptical "FAs" of varying area and aspect ratio that were distributed randomly throughout a circular cell (Figures 4J and S10, and S11). The aspect ratio was defined with respect

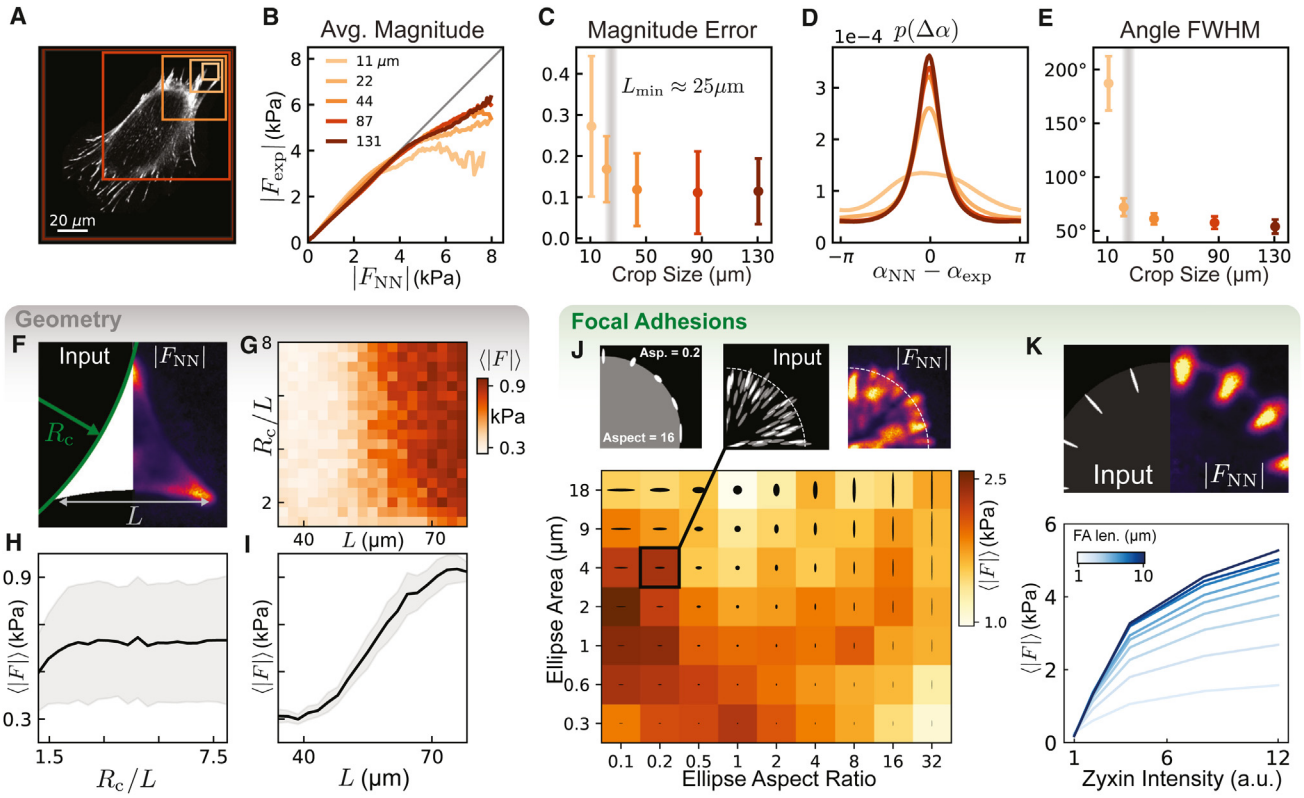


Figure 4. Neural networks identify features of cell adhesion and morphology

(A) Networks are trained with varying crop sizes, ranging from 64 pixels (≈ 10 μm) to 768 pixels (≈ 130 μ m). All networks are trained on the same data and have the same architecture as the U-Net of Figures 1 and 3.

(B) Average force magnitude (defined in Figure 1) for varying crop sizes.

(C) Magnitude error as a function of crop size, using the same metric defined in Figure 2D.

(D) Distribution of angular errors $\Delta\alpha = \alpha_{NN} - \alpha_{exp}$ for each crop size. Larger crops cause the distribution to peak sharply about $\Delta\alpha = 0$.

(E) FWHM of the distributions in (D) as a function of crop size. FWHM reduces dramatically at a crop size of ≈25 μm, beyond which it plateaus.

(F) Synthetic cells of size L consist of three points connected by circular arcs with radius R_c.

(G) Dependence of average force predicted by mask-trained U-Net on radius of curvature relative to the size of the synthetic cell, and cell size.

(H) Averaging along the x axis of (G) shows that average predicted force is independent of relative radius of curvature.

(I) Averaging along the y axis of (G) shows that average predicted force increases as a function of cell size. Shaded region in both (H) and (I) denotes one standard deviation.

(J) (Top) Synthetic cells composed of ellipses of varying aspect ratio (defined relative to radial direction) and area, which are randomly distributed in a circular boundary. A section of one such cell is shown along with the force magnitudes predicted by the zyxin-trained U-Net. (Bottom) Average predicted force magnitudes vary with aspect ratio and area.

(K) Additional synthetic cells are generated of evenly spaced, radially oriented focal adhesions with varying length and intensity. (Top) A section of one such cell is shown along with the force magnitudes predicted by the zyxin-trained U-Net. (Bottom) Average predicted force varies with zyxin intensity.

to the radial direction, allowing us to simultaneously probe the response of the network to both orientation and size of the FA-like structures. We found that the zyxin-trained U-Net predicted the highest forces for ellipses of area ∼2 μm^2 and aspect ratio of ∼ 0.1 (i.e., those pointed radially), consistent with experimental descriptions of FAs.[45,51–53] We further investigated the role of FA intensity by creating circular cells with uniformly distributed ellipses of fixed intensity and length along the edge (Figure 4K). Upon increasing the intensity of the ellipses, we found a non-linear response where the magnitude of the predicted traction forces rose sharply at first and continued to grow at a slower rate at higher intensities, consistent with previous reports of zyxin intensity increasing with applied force.[28] This retrospective analysis revealed how the neural network transforms many different specific features of the zyxin signal into cellular force predictions. Instead of memorizing complex, uninterpretable correlations in the training data, the U-Net identified biological features that allow it to accurately generalize predictions of force generation across cell types and biomechanical states.

Physical bottleneck neural networks: Learning adhesion enhances an effective elastic model

While the U-Net-learned rules for predicting forces from zyxin generalize far beyond the domain on which it was trained, it is not transparent how the network uses features of the input data to make predictions. In comparison, previous models inspired by classical continuum theory rely on simple hypotheses allowing for maximum interpretability. However, they

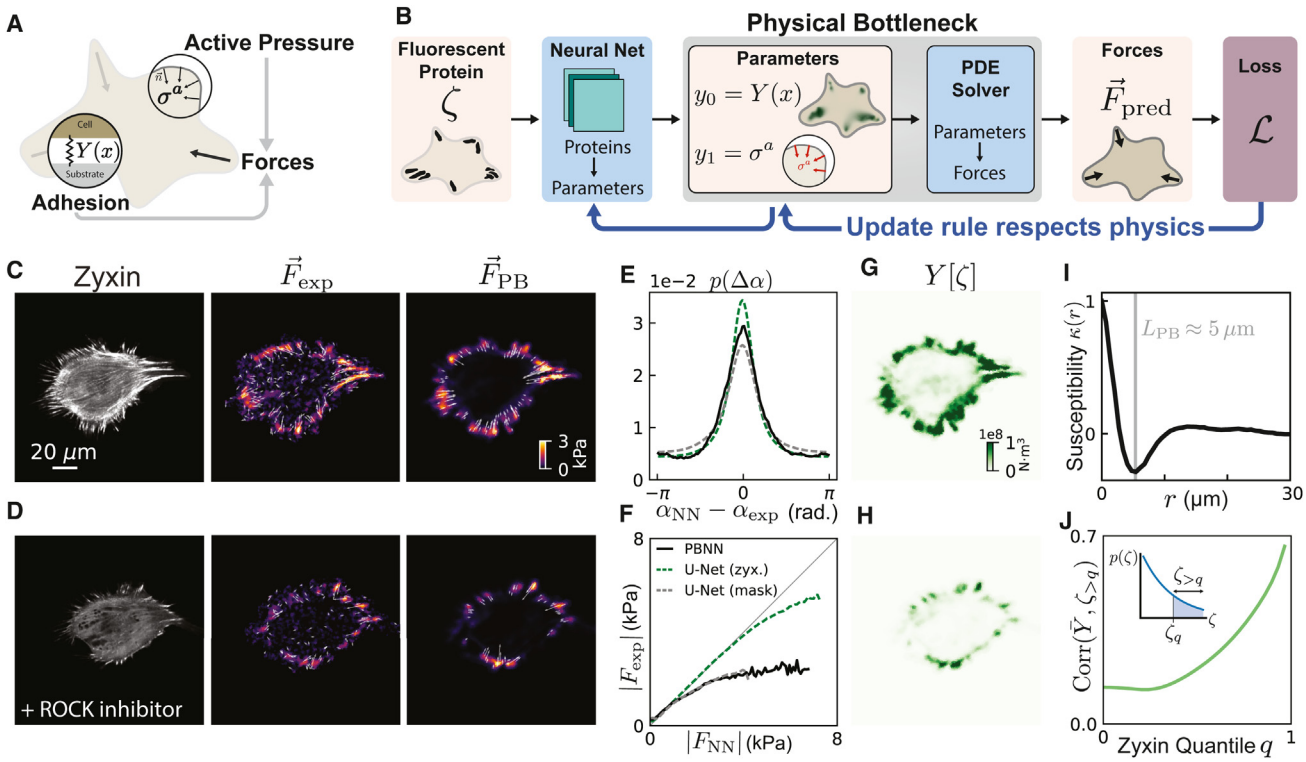


Figure 5. The physical bottleneck: Learning adhesion enhances an effective elastic model

(A) We model the cell as an elastic medium that is subject to a uniform active contractility σ^a and is pinned to the substrate with a spatially varying adhesion field $Y(x)$.

(B)–(F) (B) A neural network learns to map zyxin to model parameters, which a PDE solver uses to compute the forces. Network weights are optimized such that physical constraints are always satisfied. We refer to this strong enforcement of the PDE as a "physical bottleneck" neural network (PBNN). The elastic model with non-uniform adhesion captures forces in cells in both high (C) and low-contractility regimes (D). The PBNN predicts both forces directions (E) and magnitudes (F) similarly on average with the full mask-trained U-Net of Figure 2. The angle FWHM for the PBNN is 68° and for the mask U-Net 65°, while for the zyxin U-Net, it is 50°.

(G–J) The PBNN reveals how the learned adhesion field depends on zyxin. (G) The learned $Y(x)$ is highly heterogenous and captures the location of FAs. (H) The magnitude of the Y field decreases in response to the ROCK inhibition but remains localized to FAs. (I) The "susceptibility" of the PBNN, $\kappa(x_i, x_j) = \frac{\partial Y(x_i)}{\partial \zeta(x_j)}$ is sharply peaked within a radius of a few microns. (J) We correlate the average adhesion value $\overline{Y}$ with the amount of zyxin above a threshold ζ_q, where q denotes the qth quantile of the zyxin distribution (inset). $\overline{Y}$ correlates strongly only with the highest values of zyxin.

typically lack the ability to make predictions under wide ranges of cell shapes and distributions of localized FAs.[25–34] Here, we demonstrate how to incorporate zyxin into continuum mechanical models using neural networks, thereby learning relationships between proteins and physical parameters that enhance the generalizability of physical models.

We resort to an existing model that views the cell as an effective two-dimensional (2D) active elastic gel adhered to a substrate.[31–33] The main attraction of this minimal model is its simplicity: it represents the complex processes governing cell adhesion and contractility in terms of only two parameters, a uniform adhesion strength Y and a global active stress σ^a (Figure 5A). The forces are calculated as $\vec{F}(x) = Y\vec{u}(x)$, where $\vec{u}(x)$ is the displacement field found by minimizing the system's free energy (see supplemental information for details). Here, we extend this model by considering a spatially varying adhesion field $Y(x)$ to account for the inhomogeneous distribution of FA sites in the cell.[34] Inspired by the success of the U-Net, we connect both physical parameters to chemical quantities by making them zyxin-dependent, $Y[\zeta](x)$ and $\sigma^a[\zeta]$, with $\zeta(x)$ denoting the experimentally determined zyxin distribution, so that forces are now given by $\vec{F}(x) = Y[\zeta](x)\vec{u}(x)$.

While classical methods exist to estimate model parameters from experimental force data, they do not account for the additional constraint that the parameters are functions of zyxin. To overcome this limitation, we introduce a "physical bottleneck" neural network (PBNN) architecture. The U-Net of Figures 1, 2, 3, and 4 calculates forces by processing hundreds of features calculated in the latent layers of the network. In contrast, our physical bottleneck computes only two features from which forces are calculated in a deterministic and well-understood way. Concretely, the PBNN calculates $Y[\zeta](x)$ and $\sigma^a[\zeta]$ with a neural network and feeds them as parameters into a PDE solver to calculate traction forces (Figures 5C and S2). We train the PBNN to predict parameters that minimize the MSE between predicted forces and the experimentally measured forces. In each iterative training step, the adjoint method[54] is used to calculate updates to the physical model parameters, which are

then passed to the neural network using backpropagation. This two-step process ensures that updates to the neural network obey the stringent constraints of the physical model.

The PBNN accurately predicts forces and generalizes to cells perturbed by the ROCK inhibitor Y-27632 (Figures 5C and 5D). The predicted force angles (Figure 5E) and magnitudes (Figure 5F), however, are less accurate on average across the dataset than those predicted by the unconstrained zyxin-trained U-Net of Figures 1, 2, 3, and 4. This behavior is expected due to the additional constraints imposed on the PBNN. The PBNN nevertheless makes predictions on par with the mask-trained U-Net (Figures 5E and 5F), which indicates that the two parameters learned at the physical bottleneck contain at least as much relevant information for force prediction than anything an unconstrained deep U-Net could infer from the cell morphology alone. Moreover, the U-Net processes its latent features with a nearly arbitrarily complex function, while the PBNN processes the Y field and σ^a into forces via a simple differential equation.

The introduction of a zyxin-dependent adhesion field $Y[\zeta](x)$ was sufficient to make the physical model competitive with fully deep U-Nets. The notation $Y[\zeta](x)$ is used to indicate that Y is a functional of the zyxin field ζ, which varies in space. We occasionally omit the dependence on ζ for brevity. We found that the learned field is strongly heterogeneous and localizes to FA sites (Figures 5G and 5H). Furthermore, the intensity of $Y(x)$ decreases in response to the ROCK inhibitor Y-27632 and mirrors the reorganization and reduction in number of FAs (Figures 5D and 5H). However, it is not immediately clear how the PBNN calculated $Y(x)$ from the spatial distribution of zyxin $\zeta(x)$. To characterize how the adhesion at a point $\mathbf{x}_i$ depends on zyxin at a point $\mathbf{x}_j$, we defined the susceptibility, or linear response, of the network as $\kappa_{\mathbf{x}_i,\mathbf{x}_j} = \frac{\partial Y(\mathbf{x}_i)}{\partial \zeta(\mathbf{x}_j)}$. The susceptibility curve exhibits a rapid decay with a minimum at $\approx 5\ \mu$m (Figure 5I). Its shape resembles a Laplace filter commonly used in peak-finding algorithms, indicating that $Y(x)$ is associated with maxima in the zyxin signal. We further probed the dependence of Y on zyxin by correlating the average adhesion in each image $\overline{Y}$ with the sum of zyxin values above a given threshold (Figures 5H and S12). Upon increasing the threshold, $\overline{Y}$ becomes significantly more correlated with zyxin, suggesting that the magnitude of the adhesion field is set primarily by the highest zyxin values. Together, these results indicate that the adhesion field is encoding high-value peaks of zyxin intensity, which correspond to FAs.

The parameters learned by the PBNN are subject to the assumptions of the model used to constrain them. The elastic model makes predictions about displacements within the cell, which are not directly accessible experimentally using TFM, nor is it clear what undeformed reference frame these displacements should be measured from. This is owing to the fact that a cell, unlike a passive lattice of masses and springs, continuously undergoes cytoskeletal remodeling, even if no external deformations are applied. Nevertheless, the PBNN is still a powerful tool to test our hypothesized model, and it informs us of the minimal necessary ingredients required to predict traction stresses. We showed that cell shape (encoded as boundary conditions), a global contractile "set-point" σ^a, and a field $Y(x)$ encoding FAs were sufficient to make predictions. Furthermore, we find that a linear partial differential equation describing an intermediate displacement field is an adequate mathematical model to describe the observed behavior.

Green's function neural networks: Physics-agnostic model-building reveals length scales and effective equations

The success of the PBNN relies on generating plausible hypothesized models, hence the insights it produces are biased by the specific model prescribed. We now investigate whether we can relax these constraints to gain insights even in the absence of strong mechanical hypotheses. To do this, we turn to a physics-inspired approach to identify machine-learned rules that are agnostic to specific underlying physical models. This method again trades the complexity of our deep U-Net for fewer, more interpretable operations (Figure 6A). Specifically, we assume that the force can be written as a function of machine-learned fields derived from zyxin (yellow and pink boxes in Figure 6A(i)). While these fields are analogous to the PBNN's displacement $\vec{u}(x)$ and adhesion $Y(x)$ fields, we do not demand that these quantities obey linear elasticity or any other particular continuum theory. We only require that their non-local machine-learned relationships with zyxin density are represented by Green's functions. The Green's function method is a general tool to calculate a system's response to localized perturbations. For example, the Green's function of classical electrostatics is the $1/r$ potential that determines the effect of a charge located at a distance r away. With the aid of our machine-learned Green's functions, we will similarly seek to determine how the local traction force depends on zyxin density throughout the cell (Figure 6A(ii)). In contrast to the physical bottleneck, this is a question for which we do not have the luxury of a readily available formula.

Using the same input zyxin images (Figure 6B), we train a Green's function neural network (GFNN) to characterize spatial interactions between our input zyxin images and their respective traction maps. The GFNN learns a series of sources and fields (drawn in yellow and pink in Figures 6C–6E) from the zyxin images that it uses to predict the traction stresses (see STAR Methods and supplemental information). While in principle a GFNN can learn any number of fields, we found that a minimally complex model could achieve accurate predictions using only two (Figure 6F). Specifically, the GFNN learned two fields, ξ, χ, in terms of which predictions of the traction forces can be made as $\vec{F} = \xi(x)\vec{\nabla}\chi(x)$. Such a representation is reminiscent of Coulomb electrostatics, with ξ and χ analogous to the charge and electric potential, respectively (see Methods S1 Figure 4 and discussion for a demonstration of the GFNN method on 2D Coulomb electrostatics data). Going back to our mechanical model, note that this machine-learned formula resembles in form the physics-informed model $\vec{F} = Y\vec{u}$. The "charge" ξ identifies local peaks in zyxin intensity that are similar to FAs (Figures 6C and 6E, top). The Green's function for ξ, G_ξ, decays over a very short length scale $\sim 5\ \mu$m (Figure 6D), suggesting that it is determined by local information at the adhesion site (Figures 6A(ii) and 6D). The "potential" χ is less localized and its Green's function, G_χ, accumulates zyxin information from a

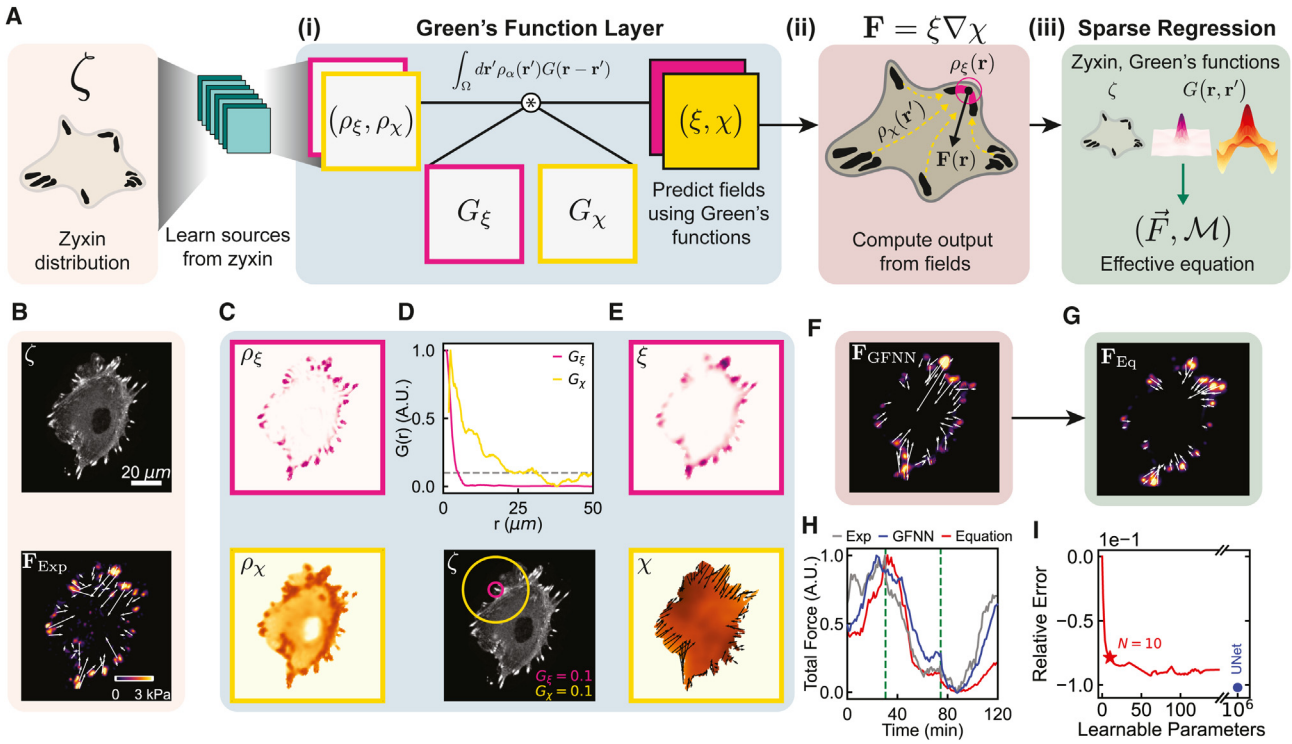


Figure 6. Green's function neural networks: Physics-agnostic model-building reveals length scales and effective equations

(A) Green's-function neural networks (GFNNs) first extract the fields and long-range interactions needed to predict forces. Next, sparse regression builds effective equations fitting the machine-learned model.

(B) The GFNN predicts traction forces from the zyxin intensity field (top), which we compare with the forces measured in experiment (bottom).

(C–E) The GFNN learns sources ρ_ξ, ρ_χ (C) from local zyxin information. These sources are integrated with machine-learned Green's functions (D) to produce the fields ξ, χ (E). The Green's functions G_ξ, G_χ decay over different length scales representing regions over which protein information accumulates (D, top). G_ξ decays over roughly a focal adhesion size, while G_χ decays more slowly across the cell.

(F) The predicted force field from the GFNN agrees well with the ground truth (B, bottom).

(G) Using sparse regression, we learn a formula (see supplemental information) based on the GFNN, which predicts the force field.

(H) Time course of predicted forces during a ROCK inhibitor experiment. We compare the experimental forces (gray) with those predicted by GFNN (blue) and the effective equation (red). The dashed lines indicate the drug wash-in and wash-out times.

(I) Sparse regression yields equations of varying complexity. We plot the improvement in mean-squared error of sparse-regressed models as a function of their complexity, compared with a baseline model $\mathbf{F} = 0$ with no learnable parameters. Star denotes the average performance of a 10-term equation.

larger area of the cell (Figures 6A(ii) and 6C–6E, bottom). This longer decay length suggests that the "potential" can infer aspects of the cell morphology from the zyxin distribution. Thus, our GFNN model predicts traction forces from interactions between a FA "charge" and a cellular "potential."

To simplify this model further, we used sparse regression to build effective equations that approximate the traction forces (Figure 6A(iii)). A qualitatively accurate analytical formula (Figure 6G) can be obtained using only a handful of terms inspired by the GFNN (see supplemental information for full equation). This formula is dramatically compressed, compared with the full U-Net, which contains 10^5 times more parameters. Nevertheless, it can capture 77% of the U-Net predictions and also generalizes to the biochemical perturbations induced by our ROCK inhibition experiments (Figures 6H and 6I). This illustrates how the U-Net, a complex black box, can be distilled into a similarly accurate formula consisting of two non-local interactions and parameterized by only a handful of terms (Figure 6I). Our proposed pipeline demonstrates how to extract effective equations that map protein distributions to traction forces without knowing the explicit underlying relations. Although no physical input was used to derive them, the structure $\vec{F} = \xi\vec{\nabla}\chi$ and the learned equation (see supplemental information) are strikingly similar to the PBNN model $\vec{F} = Y\vec{u}$. In particular, ξ and Y are both fields that accumulate zyxin information within FAs, while $\vec{\nabla}\chi$ and $\vec{u}$ are vector fields that propagate information throughout the cell.

DISCUSSION

Here, we established that deep neural networks can predict the contractile mechanics of cells directly from images of protein distributions. Our results demonstrate that images of a single FA protein, such as zyxin, contain sufficient information to accurately predict traction forces. We showed that a network trained on images of one cell type collected from one microscope can generalize across a range of cell types, experimental setups, and biomechanical regimes. These results illustrate the utility of machine learning for extracting robust predictions from

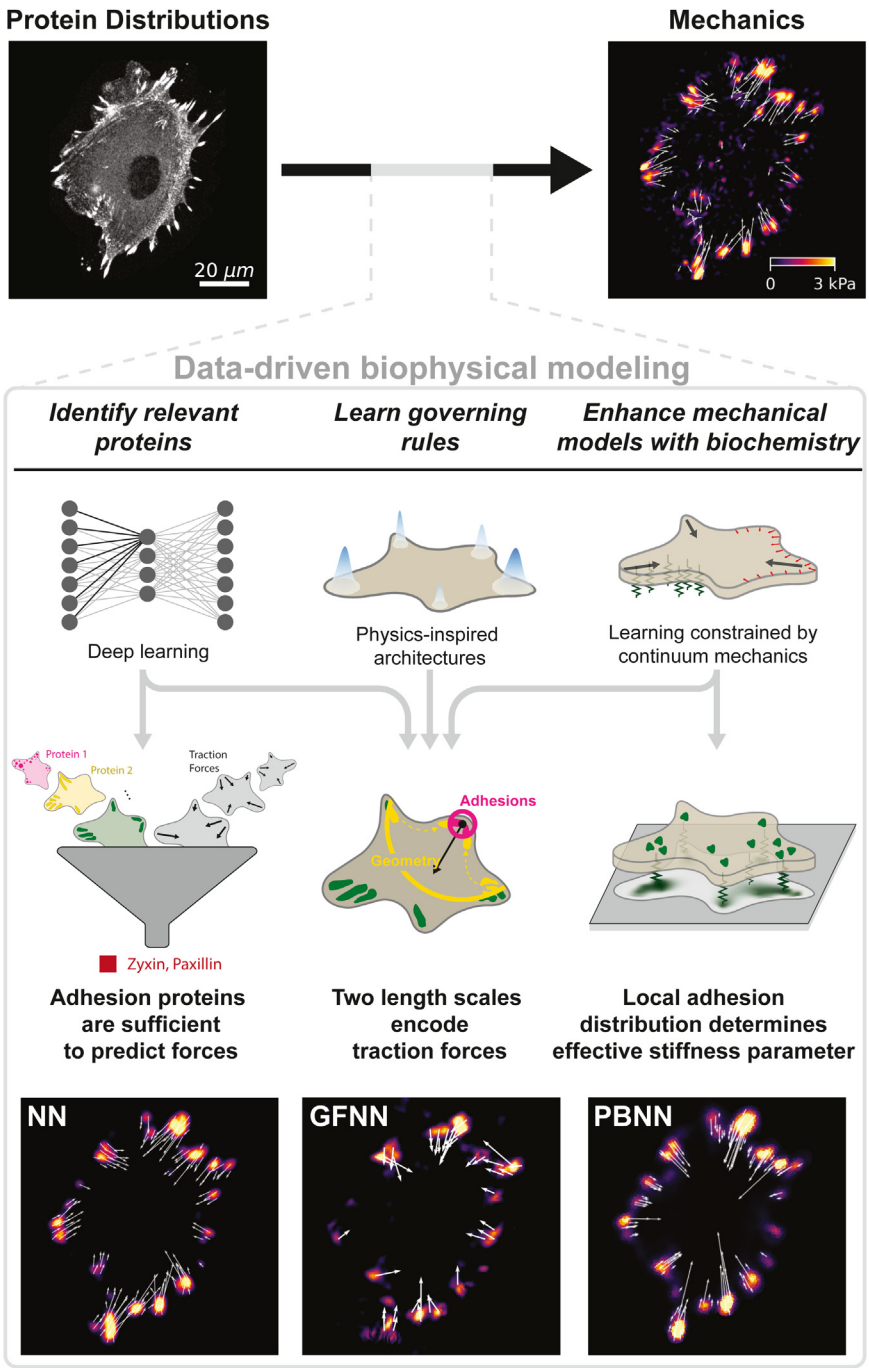


Figure 7. Data-driven biophysical modeling

We present a suite of machine learning approaches to identify and interpret links between biological information contained in images of protein distributions and traction force measurements in adherent cells.

Deep neural networks require no prior knowledge but can identify relevant components for achieving robust generalizable predictions. Here, they found that a single adhesion protein is sufficient to predict traction forces. Physics-inspired models use structural constraints to learn governing rules but remain agnostic to any specific theory. Physics-constrained learning enhances existing models by linking biochemical information to physical parameters, such as a zyxin-dependent adhesion field in a linear elastic model. All three methods accurately predicted traction stresses and revealed a consistent theme of forces encoded by adhesion protein information over two length scales—one associated with the adhesions themselves and another related to cell morphology.

heterogeneous biological data. Such methods can be implemented with a readily achievable volume of experimental image data. This makes them particularly well suited to predict mechanical behavior in situations where proteins can be easily imaged, but physical measurements are difficult.

We introduced three data-driven approaches for biophysical modeling, which incorporate machine learning at various points in the model building process to reveal new insights (Figure 7). While deep neural networks are not directly interpretable, we demonstrated the utility of synthetic data for identifying relevant components in processes with many interacting proteins. Next, we introduced a novel PBNN to test and enhance existing models, which revealed the protein-dependence of effective physical parameters. Finally, our GFNN approach traded some of the complexity of deep U-Nets for interpretable operations, uncovering long-range interactions and even an analytical formula that describes the system behavior. These methods represent an alternative approach to hypothesis testing and formulation in the framework of data-driven biophysical modeling

All three approaches, despite being subject to dramatically different constraints and assumptions, revealed two important length scales. One length scale of a few microns is consistent with the size of individual FAs and describes the relationship between force magnitude and local zyxin intensity (Figures 4J, 4K, 5G–5J, and 6C–6E, top). Predicting force directions, however, requires information encoded over a larger length scale. In the GFNN and U-Net, this scale of tens of microns is associated with aspects of cell morphology, while in the PBNN, it is accounted for in the PDE's boundary conditions (Figures 4C, 4E, 5A, 5B, and 6C–6E, bottom). Moreover, in the PBNN and GFNN, the fields corresponding to long length scales (χ, $\vec{u}$) and short length scales (ξ, Y) are coupled in a strikingly similar way ($\vec{F} = Y\,\vec{u}$ and $\vec{F} = \xi\vec{\nabla}\chi$). Neural networks are a complex black box and care must be taken when analyzing their

behavior. By using multiple methods that produce consistent results, we become more confident that the rule learned through our data-driven framework is generalizable and independent of the method we use to derive it.

From these rules we can also hypothesize why single FA proteins are sufficient to make accurate predictions. The shorter length scale identified by these models seems directly encoded by the FA itself, while the longer length scale relating to cell geometry can be inferred by integrating over many FAs. Additional information could potentially be gleaned from the geometry and orientation of the FAs, which are determined by the stress fibers to which they are coupled. This information is present in the distribution of multiple FA proteins, which might explain a certain degree of interchangeability between zyxin and paxillin as revealed in our analysis of MDCK cells. As to why zyxin appears to slightly outperform paxillin, we speculate this could be related to its force-sensitive recruitment to actin and FAs,[21,55] but further research will be required.

Finally, the approaches presented here are applicable beyond simple models of cellular contractility. Interpretable machine learning methods can lead to an improved understanding of the rules and equations governing spatiotemporal behavior in diverse biological systems.[56–59] They may be used to test and enhance existing models, as well as learn entirely new ones, in areas where first-principles approaches to biophysics fail. We only consider prediction of forces from proteins, but an autonomous dynamic model will need to be closed by a relation that predicts how protein distributions evolve in time. Our work suggests that it may suffice to consider only the dynamics of an effective adhesion field, rather than accounting for the precise details of cytoskeletal rearrangement. The methods introduced here could aid in developing mechano-chemical descriptions of diverse systems such as migrating cells,[60–64] epithelial tissue dynamics,[65–67] and morphogenesis.[68–71] They could also be coupled to recent large quantitative datasets describing organelle positioning and interactions[72] to glean additional insights. These approaches represent a step forward toward harnessing the versatility of machine learning to tackle the complexity of living systems.

Limitations of the study

Our data-driven biophysical modeling pipeline inherently relies on the data itself. Despite the demonstrated ability of our networks to generalize to unseen data, their predictions depend on the data in subtle ways. Variations in data quality caused by different microscopes, the choice of imaging fluorophore, substrate stiffness, or even how individual cells express proteins can affect the accuracy of U-Net predictions (Figures 3 and S5-S7). We can account for some of these effects via normalization, but to further improve this generalizability, it may be useful to assemble a wide-ranging dataset using an ensemble of experimental conditions. In particular, generalizing to substrates of different stiffness is complicated by the role of the TFM regularization parameter (Figures S8 and S9), which suggests that elastic substrates cannot be discarded. Beyond data quality, our analysis is limited by the use of 2D image data. Cells are not 2D objects, and so we would not expect our method to generalize to structured 3D environments where out-of-plane mechanical interactions become important. In this paper, we also restricted our analysis to time-independent models. However, we observe that cells move significantly throughout each movie. Future work may find that additional biochemical information is needed to capture the cells' full dynamic behavior.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
  - Mammalian expression vectors
  - Cell culture and transfection
  - Live cell imaging
  - Traction force microscopy experiments
  - Gel stiffness measurements
- METHOD DETAILS
  - Data processing
  - Training data
  - U-Net architecture
  - Synthetic cells
  - Effective elastic model
  - Physical bottleneck
  - Green's function neural networks
  - Sparse regression
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Outlier determination
  - Optimal predictors and histogram plots

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cell.2023.11.041.

ACKNOWLEDGMENTS

The authors thank M. Fruchart, C. Scheibner, E. Efrati, and M. Han for helpful discussions and suggestions. M.S.S. was supported by the National Science Foundation under grant no. 2022023 and DMR-2011864. J.C. acknowledges support from the National Science Foundation under grant DMR-2118415. S. Seetharaman is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Futures Program, and by the American Heart Association (AHA; Grant no. 915248). M.L.G. and V.V. acknowledge partial support from the UChicago Materials Research Science and Engineering Center (NSF DMR-2011864). M.L.G. acknowledges support from the National Institutes of Health (NIH) through awards R01-GM143792 and R01-GM104032. P.W.O. acknowledges support in part by a National Science Foundation CAREER Award #2000554 and National Institutes of Health (NIH) National Institute of Allergy and Infectious Disease(NIAID) award P01-AI102851 and National Institute of General Medical Sciences (NIGMS) award R01-GM148644.V.V. acknowledges support from the Army Research Office under grant W911NF-22-2-0109 and W911NF-23-1-0212 and the Theory in Biology program of the Chan

Zuckerberg Initiative. This research was supported from the National Science Foundation through the Center for Living Systems (grant no. 2317138). This work was completed in part with resources provided by the University of Chicago's Research Computing Center.

AUTHOR CONTRIBUTIONS

P.W.O., M.L.G., and V.V. conceived and supervised the project. S. Sala, J.D., S. Seetharaman, and P.W.O. designed and performed cell-imaging experiments. S. Sala, A.C., and P.W.O. designed and performed substrate stiffness measurement experiments. S. Sala and P.W.O. processed the data with code written by P.W.O. M.S.S. and J.C. developed the neural networks and analyzed results. M.L.G., P.W.O., and V.V. secured funding. M.S.S., J.C., M.L.G., P.W.O., and V.V. wrote the manuscript with input from all authors.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: March 21, 2023
Revised: September 20, 2023
Accepted: November 29, 2023
Published: January 8, 2024

REFERENCES

1. Pegoraro, A.F., Janmey, P., and Weitz, D.A. (2017). Mechanical properties of the cytoskeleton and cells. Cold Spring Harb. Perspect. Biol. *9*, a022038.
2. Blanchoin, L., Boujemaa-Paterski, R., Sykes, C., and Plastino, J. (2014). Actin dynamics, architecture, and mechanics in cell motility. Physiol. Rev. *94*, 235–263.
3. Fletcher, D.A., and Mullins, R.D. (2010). Cell mechanics and the cytoskeleton. Nature *463*, 485–492.
4. Svitkina, T. (2018). The actin cytoskeleton and actin-Based motility. Cold Spring Harb. Perspect. Biol. *10*, a018267.
5. Phillips, R.B., Kondev, J., and Theriot, J. (2009). Physical Biology of the Cell (Garland Science).
6. Van Saarloos, W., Vitelli, V., and Zeravcic, Z. (2023). Soft Matter: Concepts, Phenomena and Applications (Princeton University Press).
7. MacKintosh, F.C., and Schmidt, C.F. (2010). Active cellular materials. Curr. Opin. Cell Biol. *22*, 29–35.
8. Battle, C., Broedersz, C.P., Fakhri, N., Geyer, V.F., Howard, J., Schmidt, C.F., and MacKintosh, F.C. (2016). Broken detailed balance at mesoscopic scales in active biological systems. Science *352*, 604–607.
9. Prost, J., Jülicher, F., and Joanny, J.-F. (2015). Active gel physics. Nat. Phys. *11*, 111–117.
10. Romani, P., Valcarcel-Jimenez, L., Frezza, C., and Dupont, S. (2021). Crosstalk between mechanotransduction and metabolism. Nat. Rev. Mol. Cell Biol. *22*, 22–38.
11. Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L. (2019). Machine learning and the physical sciences. Rev. Mod. Phys. *91*, 45002.
12. Cichos, F., Gustavsson, K., Mehlig, B., and Volpe, G. (2020). Machine learning for active matter. Nat. Mach. Intell. *2*, 94–103.
13. Zaritsky, A., Jamieson, A.R., Welf, E.S., Nevarez, A., Cillay, J., Eskiocak, U., Cantarel, B.L., and Danuser, G. (2021). Interpretable deep learning uncovers cellular properties in label-free live cell images that are predictive of highly metastatic melanoma. Cell Syst. *12*, 733–747.e6.
14. Soelistyo, C.J., Vallardi, G., Charras, G., and Lowe, A.R. (2022). Learning biophysical determinants of cell fate with deep neural networks. Nat. Mach. Intell. *4*, 636–644.
15. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. Nature *596*, 583–589.
16. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. Science *379*, 1123–1130.
17. Iskratsch, T., Wolfenson, H., and Sheetz, M.P. (2014). Appreciating force and shape—the rise of mechanotransduction in cell biology. Nat. Rev. Mol. Cell Biol. *15*, 825–833.
18. Murrell, M., Oakes, P.W., Lenz, M., and Gardel, M.L. (2015). Forcing cells into shape: the mechanics of actomyosin contractility. Nat. Rev. Mol. Cell Biol. *16*, 486–498.
19. Schwarz, U.S., and Gardel, M.L. (2012). United we stand: integrating the actin cytoskeleton and cell-matrix adhesions in cellular mechanotransduction. J. Cell Sci. *125*, 3051–3060.
20. Burridge, K., and Guilluy, C. (2016). Focal adhesions, stress fibers and mechanical tension. Exp. Cell Res. *343*, 14–20.
21. Kuo, J.C. (2013). Mechanotransduction at focal adhesions: integrating cytoskeletal mechanics in migrating cells. J. Cell. Mol. Med. *17*, 704–712.
22. Sabass, B., Gardel, M.L., Waterman, C.M., and Schwarz, U.S. (2008). High resolution traction force microscopy based on experimental and computational advances. Biophys. J. *94*, 207–220.
23. Huang, Y., Schell, C., Huber, T.B., Şimşek, A.N., Hersch, N., Merkel, R., Gompper, G., and Sabass, B. (2019). Traction force microscopy with optimized regularization and automated bayesian parameter selection for comparing cells. Sci. Rep. *9*, 539.
24. Yunfei, H., Christoph, S., Tobias, B.H., Ahmet Nihat, Ş., Nils, H., Rudolf, M., Gerhard, G., and Benedikt, S. (2019). Traction force microscopy with optimized regularization and automated bayesian parameter selection for comparing cells. Scientific Reports *9*, 539.
25. Soiné, J.R.D., Brand, C.A., Stricker, J., Oakes, P.W., Gardel, M.L., and Schwarz, U.S. (2015). Model-based traction force microscopy reveals differential tension in cellular actin bundles. PLOS Comput. Biol. *11*, e1004076.
26. Cao, X., Lin, Y., Driscoll, T.P., Franco-Barraza, J., Cukierman, E., Mauck, R.L., and Shenoy, V.B. (2015). A chemomechanical model of matrix and nuclear rigidity regulation of focal adhesion size. Biophys. J. *109*, 1807–1817.
27. Notbohm, J., Banerjee, S., Utuje, K.J.C., Gweon, B., Jang, H., Park, Y., Shin, J., Butler, J.P., Fredberg, J.J., and Marchetti, M.C. (2016). Cellular contraction and polarization drive collective cellular motion. Biophys. J. *110*, 2729–2738.
28. Oakes, P.W., Wagner, E., Brand, C.A., Probst, D., Linke, M., Schwarz, U.S., Glotzer, M., and Gardel, M.L. (2017). Optogenetic control of RhoA reveals zyxin-mediated elasticity of stress fibres. Nat. Commun. *8*, 15817.
29. Hanke, J., Probst, D., Zemel, A., Schwarz, U.S., and Köster, S. (2018). Dynamics of force generation by spreading platelets. Soft Matter *14*, 6571–6581.
30. Vignaud, T., Copos, C., Leterrier, C., Toro-Nahuelpan, M., Tseng, Q., Mahamid, J., Blanchoin, L., Mogilner, A., Théry, M., and Kurzawa, L. (2021). Stress fibres are embedded in a contractile cortical network. Nat. Mater. *20*, 410–420.
31. Mertz, A.F., Banerjee, S., Che, Y., German, G.K., Xu, Y., Hyland, C., Marchetti, M.C., Horsley, V., and Dufresne, E.R. (2012). Scaling of traction forces with the size of cohesive cell colonies. Phys. Rev. Lett. *108*, 198101.
32. Oakes, P.W., Banerjee, S., Marchetti, M.C., and Gardel, M.L. (2014). Geometry regulates traction stresses in adherent cells. Biophys. J. *107*, 825–833.
33. Edwards, C.M., and Schwarz, U.S. (2011). Force localization in contracting cell layers. Phys. Rev. Lett. *107*, 128101.

34. Solowiej-Wedderburn, J., and Dunlop, C.M. (2022). Sticking around: cell adhesion patterning for energy minimization and substrate mechanosensing. Biophys. J. *121*, 1777–1786.

35. Hoffman, L.M., Jensen, C.C., Kloeker, S., Wang, C.L., Yoshigi, M., and Beckerle, M.C. (2006). Genetic ablation of zyxin causes Mena/VASP mislocalization, increased motility, and deficits in actin remodeling. J. Cell Biol. *172*, 771–782.

36. Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. In Med. Image Comput. Comput. Assisted Interv. MICCAI, N. Navab, J. Hornegger, W.M. Wells, and A.F. Frangi, eds. (Springer International Publishing), pp. 234–241.

37. Wang, Y.L., and Lin, Y.C. (2021). Traction force microscopy by deep learning. Biophys. J. *120*, 3079–3090.

38. Kratz, F.S., Möllerherm, L., and Kierfeld, J. (2022). Enhancing robustness, precision and speed of traction force microscopy with machine learning. Biophys. J. *122*, 3489–3505.

39. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A ConvNet for the 2020s. arXiv.. https://doi.org/10.48550/arXiv.2201.03545.

40. Yoshigi, M., Hoffman, L.M., Jensen, C.C., Yost, H.J., and Beckerle, M.C. (2005). Mechanical force mobilizes zyxin from focal adhesions to actin filaments and regulates cytoskeletal reinforcement. J. Cell Biol. *171*, 209–215.

41. Stricker, J., Beckham, Y., Davidson, M.W., and Gardel, M.L. (2013). Myosin II-Mediated focal adhesion maturation is tension insensitive. PLOS One *8*, e70652.

42. Oakes, P.W., Bidone, T.C., Beckham, Y., Skeeters, A.V., Ramirez-San Juan, G.R.R.-S., Winter, S.P., Voth, G.A., and Gardel, M.L. (2018). Lamellipodium is a myosin-independent mechanosensor. Proc. Natl. Acad. Sci. USA *115*, 2646–2651.

43. Théry, M., Pépin, A., Dressaire, E., Chen, Y., and Bornens, M. (2006). Cell distribution of stress fibres in response to the geometry of the adhesive environment. Cell Motil. Cytoskeleton *63*, 341–355.

44. Tseng, Q., Wang, I., Duchemin-Pelletier, E., Azioune, A., Carpi, N., Gao, J., Filhol, O., Piel, M., Théry, M., and Balland, M. (2011). A new micropatterning method of soft substrates reveals that different tumorigenic signals can promote or reduce cell contraction levels. Lab Chip *11*, 2231–2240.

45. Prager-Khoutorsky, M., Lichtenstein, A., Krishnan, R., Rajendran, K., Mayo, A., Kam, Z., Geiger, B., and Bershadsky, A.D. (2011). Fibroblast polarization is a matrix-rigidity-dependent process controlled by focal adhesion mechanosensing. Nat. Cell Biol. *13*, 1457–1465.

46. Gardel, M.L., Sabass, B., Ji, L., Danuser, G., Schwarz, U.S., and Waterman, C.M. (2008). Traction stress in focal adhesions correlates biphasically with actin retrograde flow speed. J. Cell Biol. *183*, 999–1005.

47. Han, S.J., Bielawski, K.S., Ting, L.H., Rodriguez, M.L., and Sniadecki, N.J. (2012). Decoupling substrate stiffness, spread area, and micropost density: a close spatial relationship between traction forces and focal adhesions. Biophys. J. *103*, 640–648.

48. Thievessen, I., Thompson, P.M., Berlemont, S., Plevock, K.M., Plotnikov, S.V., Zemljic-Harpf, A., Ross, R.S., Davidson, M.W., Danuser, G., Campbell, S.L., et al. (2013). Vinculin-actin interaction couples actin retrograde flow to focal adhesions, but is dispensable for focal adhesion growth. J. Cell Biol. *202*, 163–177.

49. Liu, Y., Medda, R., Liu, Z., Galior, K., Yehl, K., Spatz, J.P., Cavalcanti-Adam, E.A., and Salaita, K. (2014). Nanoparticle tension probes patterned at the nanoscale: impact of integrin clustering on force transmission. Nano Lett. *14*, 5539–5546.

50. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. Proc. Natl. Acad. Sci. USA *116*, 22071–22080.

51. Geiger, B., Bershadsky, A., Pankov, R., and Yamada, K.M. (2001). Transmembrane crosstalk between the extracellular matrix–cytoskeleton crosstalk. Nat. Rev. Mol. Cell Biol. *2*, 793–805.

52. Kanchanawong, P., Shtengel, G., Pasapera, A.M., Ramko, E.B., Davidson, M.W., Hess, H.F., and Waterman, C.M. (2010). Nanoscale architecture of integrin-based cell adhesions. Nature *468*, 580–584.

53. Stricker, J., Aratyn-Schaus, Y., Oakes, P.W., and Gardel, M.L. (2011). Spatiotemporal constraints on the force-dependent growth of focal adhesions. Biophys. J. *100*, 2883–2893.

54. Tröltzsch, F. (2000). Optimal control of partial differential equations: theory, methods and applications. In Graduate Studies in Mathematics, *112* (American Mathematical Society).

55. Guo, W.H., and Wang, Y.L. (2007). Retrograde fluxes of focal adhesion proteins in response to cell migration and mechanical signals. Mol. Biol. Cell *18*, 4519–4527.

56. Gowda, K., Ping, D., Mani, M., and Kuehn, S. (2022). Genomic structure predicts metabolite dynamics in microbial communities (3). Cell *185*, 530–546.e25.

57. Supekar, R., Song, B., Hastewell, A., Choi, G.P.T., Mietke, A., and Dunkel, J. (2023). Learning hydrodynamic equations for active matter from particle simulations and experiments. Proc. Natl. Acad. Sci. USA *120*, e2206994120.

58. Golden, M., Grigoriev, R.O., Nambisan, J., and Fernandez-Nieves, A. (2023). Physically informed data-driven modeling of active nematics. Sci. Adv. *9*, eabq6120.

59. Joshi, C., Ray, S., Lemma, L.M., Varghese, M., Sharp, G., Dogic, Z., Baskaran, A., and Hagan, M.F. (2022). Data-driven discovery of active nematic hydrodynamics. Phys. Rev. Lett. *129*, 258001.

60. Pertz, O., Hodgson, L., Klemke, R.L., and Hahn, K.M. (2006). Spatiotemporal dynamics of rhoa activity in migrating cells. Nature *440*, 1069–1072.

61. Ponti, A., Machacek, M., Gupton, S.L., Waterman-Storer, C.M., and Danuser, G. (2004). Two distinct actin networks drive the protrusion of migrating cells. Science *305*, 1782–1786.

62. Alert, R., and Trepat, X. (2020). Physical models of collective cell migration. Annu. Rev. Condens. Matter Phys. *11*, 77–101.

63. Brückner, D.B., Arlt, N., Fink, A., Ronceray, P., Rädler, J.O., and Broedersz, C.P. (2021). Learning the dynamics of cell–cell interactions in confined cell migration. Proc. Natl. Acad. Sci. USA *118*, e2016602118.

64. Brückner, D.B., Schmitt, M., Fink, A., Ladurner, G., Flommersfeld, J., Arlt, N., Hannezo, E., Rädler, J.O., and Broedersz, C.P. (2022). Geometry adaptation of protrusion and polarity dynamics in confined cell migration. Phys. Rev. X *12*, 31041.

65. Devany, J., Sussman, D.M., Yamamoto, T., Manning, M.L., and Gardel, M.L. (2021). Cell cycle-dependent active stress drives epithelia remodeling. Proc. Natl. Acad. Sci. USA *118*, e1917853118.

66. Copenhagen, K., Alert, R., Wingreen, N.S., and Shaevitz, J.W. (2021). Topological defects promote layer formation in Myxococcus xanthus colonies. Nat. Phys. *17*, 211–215.

67. Saw, T.B., Doostmohammadi, A., Nier, V., Kocgozlu, L., Thampi, S., Toyama, Y., Marcq, P., Lim, C.T., Yeomans, J.M., and Ladoux, B. (2017). Topological defects in epithelia govern cell death and extrusion. Nature *544*, 212–216.

68. Maroudas-Sacks, Y., Garion, L., Shani-Zerbib, L., Livshits, A., Braun, E., and Keren, K. (2021). Topological defects in the nematic order of actin fibres as organization centers of hydra morphogenesis. Nat. Phys. *17*, 251–259.

69. Streichan, S.J., Lefebvre, M.F., Noll, N., Wieschaus, E.F., and Shraiman, B.I. (2018). Global morphogenetic flow is accurately predicted by the spatial distribution of myosin motors. Elife *7*.

70. Romeo, N., Hastewell, A., Mietke, A., and Dunkel, J. (2021). Learning developmental mode dynamics from single-cell trajectories. eLife *10*, e68679.

71. Alba, V., Carthew, J.E., Carthew, R.W., and Mani, M. (2021). Global constraints within the developmental program of the Drosophila wing. eLife *10*, e66750.

72. Viana, M.P., Chen, J., Knijnenburg, T.A., Vasan, R., Yan, C., Arakaki, J.E., Bailey, M., Berry, B., Borensztejn, A., Brown, E.M., et al. (2023). Integrated intracellular organization and its variations in human iPS cells. Nature *613*, 345–354.

73. Hoffman, L.M., Nix, D.A., Benson, B., Boot-Hanford, R., Gustafsson, E., Jamora, C., Menzies, A.S., Goh, K.L., Jensen, C.C., Gertler, F.B., et al. (2003). Targeted disruption of the murine zyxin gene. Mol. Cell. Biol. *23*, 70–79.

74. Borghi, N., Lowndes, M., Maruthamuthu, V., Gardel, M.L., and Nelson, W.J. (2010). Regulation of cell motile behavior by crosstalk between cadherin- and integrin-mediated adhesions. Proc. Natl. Acad. Sci. USA *107*, 13324–13329.

75. Smith, M.A., Blankman, E., Gardel, M.L., Luettjohann, L., Waterman, C.M., and Beckerle, M.C. (2010). A zyxin-mediated mechanism for actin stress fiber maintenance and repair. Dev. Cell *19*, 365–376.

76. Van Rossum, G., and Drake, F.L. (2009). Python 3 Reference Manual, CreateSpace.

77. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In Adv. Neural Inf. Process. Syst. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, eds. (Curran Associates, Inc.), 8024–8035.

78. Schneider, C.A., Rasband, W.S., and Eliceiri, K.W. (2012). NIH Image to ImageJ: 25 years of image analysis. Nat. Methods *9*, 671–675.

79. The MathWorks Inc. (2022). MATLAB Version: 9.13.0 (r2022b).

80. Butler, J.P., Tolić-Nørrelykke, I.M., Fabry, B., and Fredberg, J.J. (2002). Traction fields, moments, and strain energy that cells exert on their surroundings. Am. J. Physiol. Cell Physiol. *282*, C595–C605.

81. Hoffman, L.M., Jensen, C.C., Chaturvedi, A., Yoshigi, M., and Beckerle, M.C. (2012). Stretch-induced actin remodeling requires targeting of zyxin to stress fibers and recruitment of actin regulators. Mol. Biol. Cell *23*, 1846–1859.

82. Sala, S., and Oakes, P.W. (2021). Stress fiber strain recognition by the LIM protein testin is cryptic and mediated by RhoA. Mol. Biol. Cell *32*, 1758–1771.

83. Lee, D., Rahman, M.M., Zhou, Y., and Ryu, S. (2015). Three-dimensional confocal microscopy indentation method for hydrogel elasticity measurement. Langmuir *31*, 9684–9693.

84. Dimitriadis, E.K., Horkay, F., Maresca, J., Kachar, B., and Chadwick, R.S. (2002). Determination of elastic moduli of thin layers of soft material using the atomic force microscope. Biophys. J. *82*, 2798–2810.

85. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: transformers for image recognition at scale. arXiv.. https://doi.org/10.48550/arXiv.2010.11929.

86. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: hierarchical vision transformer using shifted windows. arXiv.. https://doi.org/10.48550/arXiv.2103.14030.

87. Loshchilov, I., and Hutter, F. (2019). Decoupled weight decay regularization. In International Conference on Learning Representations, *2019*.

88. Mitusch, S.K., Funke, S.W., and Dokken, J.S. (2019). dolfin-adjoint 2018.1: automated adjoints for fenics and firedrake. J. Open Source Software *4*, 1292.

89. Kingma, D.P., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv.. https://doi.org/10.48550/arXiv.1412.6980.

90. Brunton, S.L., Proctor, J.L., and Kutz, J.N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. USA *113*, 3932–3937.

91. Kaptanoglu, A.A., de Silva, B.M., Fasel, U., Kaheman, K., Goldschmidt, A.J., Callaham, J., Delahunt, C.B., Nicolaou, Z.G., Champion, K., Loiseau, J.-C., et al. (2022). Pysindy: A comprehensive python package for robust sparse system identification. J. Open Source Software *7*, 3994.

92. de Silva, B., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J., and Brunton, S. (2020). Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data. J. Open Source Software *5*, 2104.

93. Guthrie, W.F. (2020). NIST/SEMATECH e-Handbook of Statistical Methods (NIST Handbook 151). note: Section 1.3.5.17.

94. Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comp. Phys. *378*, 686–707.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
Dulbecco's modified eagle medium 1	Corning	MT10013CV
Leibovitz's L-15 medium without phenol red	Gibco	21083-027
Fetal Bovine Serum	Corning	MT35-010-CV
Antibiotic-antimycotic Solution	Corning	MT30004CI
L-glutamine	Fisher Scientific	MT25005CIRF
ROCK inhibitor Y-27632	Cayman Chemical Company	10005583
3-aminopropyl-trimethyoxysilane	Acros Organics	313255000
70% Glutaraldehyde	Electron Microscopy Sciences	16360
40% acrylamide solution	Biorad	1610140
2% bis-acrylamide solution	Biorad	1610142
0.04 m Dark Red fluorescent microspheres	Invitrogen	F8789
Sulfo-Sanpah	Pierce Scientific	22589
Rat tail collagen I	Corning	354236
Human plasma fibronectin	Millipore	FC010
Deposited data		
TFM data	This paper	https://uchicago.box.com/s/663yzjrxh41antctsyu2i872artqrmtd
Original code	This paper	https://doi.org/10.5281/zenodo.10438518
Experimental models: cell lines		
Mouse embryonic fibroblasts	Hoffman et al.[73]	N/A
Mouse embryonic fibroblasts zyxin KO	Hoffman et al.[73]	N/A
Mouse embryonic fibroblasts zyxin KO + EGFP-zyxin	Hoffman et al.[73]	N/A
Human Osteosarcoma cells (U2OS)	ATCC	HTB-96
EGFP-Paxillin-MDCK (G Type II cells)	Borghi et al.[74]	N/A
Oligonucleotides		
Zyxin FW primer: 5'- CCGCTCGAGCTATGGCGGCCC -3'	This paper	N/A
Zyxin RV primer: 5'- CGGGATCCCTACGTCTGGGCTCT -3'	This paper	N/A
Recombinant DNA		
Plasmid: mApple-actin	Addgene	RRID:Addgene_54862
Plasmid: mApple-paxillin	Addgene	RRID:Addgene_54935
Plasmid: mApple-myosin light chain	Addgene	RRID:Addgene_54920
Plasmid: mito-mGarnet	Addgene	RRID:Addgene_104309
Plasmid: eGFP-Paxillin	Addgene	RRID;Addgene_15233
Plasmid: eGFP-zyxin	Smith et al.[75]	N/A
Plasmid: mApple-zyxin	This paper	N/A
Software and algorithms		
Python version 3.7.10	Van Rossum and Drake[76]	python.org
PyTorch version 1.10.0	Paszke et al.[77]	pytorch.org
ImageJ	Schneider et al.[78]	imagej.org
Matlab	The MathWorks Inc.[79]	mathworks.com/products/matlab.html

(*Continued on next page*)

Continued		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
TFM analysis	Butler et al.,[80] Sabass et al.,[22] Huang et al.[23]	https://github.com/OakesLab/TFM
U-Net code	This paper	https://github.com/schmittms/cell_force_prediction
Physical bottleneck code	This paper	https://github.com/schmittms/physical_bottleneck
GFNN code	This paper	https://github.com/jcolen/cell_force_gfnn
Other		
Neon Electroporation System	Thermo Fisher Scientific	MPK5000
3.5 mm Stainless Steel Ball Bearing	Uxcell	ux0292
1.5 mm Stainless Steel Ball Bearing	Uxcell	ux0274
1/8 in Tungsten Carbide Ball Bearing	Uxcell	ux0891

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact: Vincenzo Vitelli (vitelli@uchicago.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All TFM data, both raw (.tif) and processed (.npy) files, used to train the models in this work have been deposited at https://uchicago.box.com/s/663yzjrxh41antctsyu2i872artqrmtd and are publicly available as of the date of publication. DOIs are listed in the key resources table.
- Code to perform TFM calculations can be found at https://github.com/OakesLab/TFM
- A maintained and up-to-date version of the code for the U-Net models, as well as a walk-through of the code is available at github.com/schmittms/cell_force_prediction/. Maintained code for the physical bottleneck can be found at github.com/schmittms/physical_bottleneck/. Maintained code for the Green's function neural networks can be found at github.com/jcolen/cell_force_gfnn/. Code on Github is not associated with a permanent identifier and may change in the future. Original code has therefore also been deposited at Zenodo with permanent identifier https://doi.org/10.5281/zenodo.10438518.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

Mammalian expression vectors

mApple-actin (Addgene plasmid #54862), mApple-paxillin (Addgene plasmid #54935), mApple-myosin light chain (Addgene plasmid #54920) and mito-mGarnet (Addgene plasmid #104309) vectors were a kind gift from Michael Davidson. EGFP-Paxillin (Addgene plasmid #15233) was a kind gift from Rick Horwitz. For expression as mApple-fusion protein, cDNA encoding zyxin was amplified (Forward primer: 5'- CCGCTCGAGCTATGGCGGCCC -3', Reverse primer: 5'- CGGGATCCCTACGTCTGGGCTCT -3') from the eGFP-zyxin vector (kind gift from the Waterman lab) and cloned into the mApple-C1 vector (Addgene plasmid #54631) using the XhoI and BamHI restriction sites.

Cell culture and transfection

Mouse embryonic fibroblasts (MEFs) stably expressing EGFP-zyxin[73,81] were a kind gift of Mary Beckerle's laboratory (University of Utah, Salt Lake City, UT). Human Osteosarcoma (U2OS) cells were purchased from ATCC (Manassas, VA). MEFs and U2OS cells were cultured in DMEM (MT10013CV, Corning) supplemented with 10% fetal bovine serum (MT35-010-CV, Corning) and 1% antibiotic–antimycotic solution (MT30004CI, Corning) at 37°C and 5% CO2. MDCK cells were cultured in DMEM supplemented

with 2mM L-glutamine and 10% fetal bovine serum, also at 37°C and 5% CO2. At 24 h before each experiment, cells were transfected with 5 g total DNA using a Neon electroporation system (ThermoFisher Scientific) and plated on polyacrylamide gels for traction force microscopy analysis.

Live cell imaging

MEFs and U2OS were imaged in Leibovitz's L-15 medium without phenol red (21083-027, Gibco), 10% fetal bovine serum (MT35-010-CV, Corning), and 1% antibiotic, antimycotic solution (MT30004CI, Corning) at 37°C on a Marianas Imaging System (Intelligent Imaging Innovations) consisting of an Axio Observer 7 inverted microscope (Zeiss) attached to a W1 Confocal Spinning Disk (Yokogawa) with Mesa field flattening (Intelligent Imaging Innovations), a motorized X,Y stage (ASI), and a Prime 95B sCMOS (Photometrics) camera. Illumination was provided by a TTL triggered multifiber laser launch (Intelligent Imaging Innovations) consisting of 405, 488, 561, and 637 nm lasers, using a 63X, 1.4 NA Plan-Apochromat objective (Zeiss). Temperature and humidity were maintained using a Bold Line full enclosure incubator (Oko Labs). The microscope was controlled using Slidebook 6 Software (Intelligent Imaging Innovations). All imaging was performed as single confocal slices. FA proteins and the gel were imaged at the same focal plane. Actin, myosin, and mitochondria were imaged at a slightly higher focal plane to achieve optimal focus of the structures being imaged. Cells were imaged for 2 h at 1 min intervals, with typically 5-6 cells being imaged per experiment. When used, a 2X concentration of 5 μM of the ROCK inhibitor Y27632 (10005583, Cayman Chemical Company) in imaging media was added after 30 min. After another 45 min (i.e. 75 min in total), the drug containing media was replaced with fresh imaging media.

Summary table of the datasets considered in this work. We label each dataset by the day on which it was taken. In the case of MDCK, "number of cells" indicates the number of cell clusters. Cells were imaged at a frequency of 1 min $^{-1}$, so the time duration in minutes corresponds to the number of frames in each time series.

Day	Cell type	Proteins	Number of cells	Time duration (min)	Notes
1	MEF	zyxin	4	180	
2	MEF	zyxin	4	240	
3	MEF	zyxin, actin	4	120	Y-27632 from T = 30–75
4	MEF	zyxin, actin	5	120	Y-27632 from T = 30–75
5	MEF	zyxin, paxillin	4	120	Y-27632 from T = 30–75
6	MEF	zyxin, myosin	10	120	Y-27632 from T = 30–75
7	MEF	zyxin, paxillin	7	120	Y-27632 from T = 30–75
8	MEF	zyxin, mitoch.	7	120	Y-27632 from T = 30–75
9	MEF	actin, paxillin	10	120	Y-27632 from T = 30–75
10	U2OS	zyxin	5	120	Y-27632 from T = 30–75
11	U2OS	zyxin	12	120	Y-27632 from T = 30–75
12	MDCK	paxillin	17 (clusters)	42	

EGFP-Paxillin-expressing MDCK (G Type II cells[74]) were imaged using a Nikon Ti-E Spinning Disk Confocal microscope with a 40x, 1.15 NA WI objective. Images were acquired at 5-min intervals for 4 h using 488 and 642 lasers, and standard filter sets (Em 525/50, Em 700/75) (Chroma Technology, Bellows Falls, VT). Samples were mounted on the microscope in a humidified stage top incubator maintained at 37C and 5% CO2. Images were acquired using the Andor Zyla 4.2 CMOS camera (Andor Technology, Belfast, UK).

Traction force microscopy experiments

Traction force microscopy was performed as described previously.[22,82] Coverslips were prepared by incubating with a 2% solution of (3-aminopropyl)trimethyoxysilane (313255000, Acros Organics) diluted in isopropanol. Coverslips were washed with DI water 5 times for 10 min and cured overnight at 37°C. Coverslips were incubated with 1% glutaraldehyde (16360, Electron Microscopy Sciences) in ddH20 for 30 min at room temperature and washed 3 times for 10 min in distilled water, air dried and stored at room temperature. Polyacrylamide gels (shear modulus for MEFs and U2OS cells: 16 kPa—final concentrations of 12% acrylamide (1610140, Bio-Rad) and 0.15% bis-acrylamide (1610142, Bio-Rad), and 10 kPa-final concentrations of 7.5% acrylamide and 0.2% bis-acrylamide; shear modulus for MDCK cells: 2.8 kPa-final concentrations of 7.5% acrylamide and 0.1% bis-acrylamide) were embedded with 0.04-μm fluorescent microspheres (F8789, Invitrogen) and polymerized on activated glass coverslips for 30 min - 1 h at room temperature. After polymerization, gels were rehydrated for 45 min, treated with cross-linker Sulfo-Sanpah (22589, Pierce Scientific) and photo-activated for 5 min. Polyacrylamide gels were then washed 3 times with PBS and coupled to matrix proteins, rat tail collagen I (for MDCK cells, overnight at 4°C; Corning) or human plasma fibronectin (for MEFs and U2OS cells, 1 h at room temperature; FC010,

Millipore). Following matrix protein cross-linking, cells were plated on the gels and allowed to adhere overnight. Cells were imaged the following day. Immediately after imaging, cells were removed from the gel using 0.05% SDS and a reference image of the fluorescent beads in the unstrained gel was taken.

Analysis of traction forces was performed using code written in Python according to previously described approaches.[22,29,82] Code is available at https://github.com/OakesLab/TFM. Prior to processing, images were flat-field corrected and aligned to the reference bead image with the cell detached. Other acquired channels were shifted using the same alignment measurements from the bead channel. Displacements in the beads were calculated using an optical flow algorithm in OpenCV (Open Source Computer Vision Library, https://github/itseez/opencv) with a window size of 8 pixels. Traction stresses were calculated using the Fourier Transform Traction Cytometry (FTTC) approach[22,23,80] as previously described, with a regularization parameter of 4.1×10^{-4} for the 16 kPa data set, 2.1×10^{-3} for the 10 kPa data set, and 1.7×10^{-3} for the 2.8 kPa data set.

Gel stiffness measurements

Gels were fabricated as described above, with the only difference that a spacer was used during polymerization to create a thicker gel of ≈ 300-350 m in height. Gel stiffness was measured be measuring the deformation caused by a stainless steel ball bearing 3.5 *mm* in diameter, as previously described.[83] Briefly, the gel height was measured by taking the difference between the bottom and top of the gel. A confocal z-stack with a step size of 1.25 μm was then taken through the top of the gel, and the deformation was determined by finding the center of the indentation and fitting a circle with radius equivalent to the bearing. This depth measurement was repeated in two orthogonal directions and averaged. The gel Young's modulus was then determined using a modified Hertz model[83,84] to account for the gel being thin gels bonded to a surface. At least two measurements were taken per gel, and experiments were repeated in triplicate. Gel stiffness value represent the shear modulus.

METHOD DETAILS

Data processing

Fluorescent images are normalized to have similar values across all cells, for all different proteins considered. For each cell, we calculate the mean value of the fluorescent signal f within the cell mask, $\mu_{\text{in}}^{\text{cell}} = \langle\langle f(x,t)\rangle_{x\in\text{mask}}\rangle_t$, and the average value of the signal outside the mask $\mu_{\text{out}}^{\text{cell}} = \langle\langle f(x,t)\rangle_{x\notin\text{mask}}\rangle_t$. The signal is then normalized as $f_{\text{norm}}(x,t) = (f(x,t) - \mu_{\text{out}}^{\text{cell}})/(\mu_{\text{in}}^{\text{cell}} - \mu_{\text{out}}^{\text{cell}})$ and any negative values (corresponding to values below the noise value of empty space) are set to 0. This ensures that f_{norm} has a mean value of approximately 1. Cell masks are binary and are generated by thresholding the zyxin channel in each image and filling any holes which appear.

Due to variations in substrate preparation, forces measured by cell depend slightly on the experimental round they belonged to. In our case this corresponds to the day on which they were measured (cf. Figure 1). We therefore normalize the forces of each cell by the average within their dataset, $\mu_F^{\text{day}} = \langle\langle|F(x,t)|\rangle_{x,t}\rangle_{\text{cell}\in\text{day}}$, so that $\vec{F}_{\text{norm}}^{\text{cell}} = \vec{F}^{\text{cell}}/\mu_F^{\text{day}}$ for each cell in day. Normalized fluorescent signals and forces are used everywhere in this work.

Training data

The training and testing data used for all networks in this work is shown below.

Overview table of the training and testing data used in this work. "D" stands for "day", corresponding to datasets in the previous table. For the protein experiments (U-Net [2]), separate networks were, for each protein input, trained on all but one cell that was reserved for testing. For example, the dataset in row 5 ("D3-4") contains 8 cells. We therefore train 8 identical U-Nets on the zyxin channel from 7 cells, withholding a different cell for testing each time.

Network	Figures	Trained on	Evaluated on
U-Net [1]	1, 4	(16 cells) D1-6, zyxin	D1: {2}, D2: {1, 2}, D3:{2,3,5}
		(first 30 frames of each cell)	D4: {4,5}, D5: {2,4}, D6: {1, 3,4,5,6}
	3A–3C	see above	U2OS
	3D–3F	see above	MDCK
	3G and 3H	see above	D4: {4}
U-Net [2]	2	(8 cells) D3-4, zyxin OR mask OR actin OR zyxin and actin	(1 cell; each cell chosen once)
U-Net [2]	2	(10 cells) D5, 7, zyxin OR mask OR paxillin OR zyxin and paxillin	(1 cell; each cell chosen once)
U-Net [2]	2	(9 cells) D6, zyxin OR mask OR myosin OR zyxin and myosin	(1 cell; each cell chosen once)
U-Net [2]	2	(6 cells) D8, zyxin OR mitoch. OR myosin OR zyxin and mitoch.	(1 cell; each cell chosen once)
U-Net [2]	2	(10 cells) D9, paxillin OR actin OR paxillin and actin	(1 cell; each cell chosen once)
PBNN	5	D1: {3,5}, D2: {2,4}, D3: {1,3}, D4: {1,2,4}, D6: {1}, zyxin	D1: {2,4}, D2: {3,5}, D3: {2,5}, D4: {3,5}, D6: {2,3,4,5},
GFNN	6	D1 {1, 2} D2 {3}	D3 {1}

The dataset used to train the U-Net of Figures 1, 3, and 4 consists of 31 cells from days 1-6. Of these, 8 were not subjected to any perturbation (Days 1 and 2). For training we randomly select 16 cells from the full set of 31. To ensure that only images of cells in their basal contractile are present, we include only the first 30 minutes of each time series. Even for cells not subject to the perturbation, we only take the first 30 minutes to ensure that each cell is represented equally in the dataset. In total, this amounts to 480 training frames. For evaluation, full time series are used. The test statistics shown in Figures 1, 3, and 4 are computed from the cells shown in the table above.

For the U-Nets trained on individual proteins shown in Figure 2 we use 5 datasets. The first is composed of cells from Days 3 and 4, where actin and zyxin were measured simultaneously; the second from Days 5 and 7, with zyxin and paxillin measured; the third from Day 6 with zyxin and myosin measured; the fourth from Day 8 with zyxin and mitochondria measured; the fifth from Day 9 with paxillin and actin measured. For each combination of protein inputs, we train a network on data from all but one cell. We repeat this, with separate networks, using each cell in the dataset as the hold-out cell one time. The training data consists of full time series (which includes Y-27632 perturbations, if present). We cap the total size of the training dataset to 600 frames, which is roughly the size of the smallest dataset (Day 6, zyxin + mitochondria), for a fair comparison.

U-Net architecture

Neural networks are implemented in Python using the Pytorch library. Code for network implementation, training, and evaluation is available online at github.com/schmittms/cell_force_prediction.

Predicting force distributions from protein fluorescent images is an image analysis problem and many neural network variants have been proposed for such tasks. In the supplemental information, we provide a primer on common network types and relevant considerations for choosing among them. We opted for a convolutional neural network (CNN) due to their straightforward implementation and training procedure. A CNN also exploits the spatial structure of the data to limit the number of trainable parameters. While attention-based networks such as vision transformers have also proven successful at image analysis tasks,[85,86] they come with higher computational and training costs and require very large datasets. To achieve competitive performance with visual attention networks while maintaining efficiency, we instead adapted our CNNs with ConvNext design.[39] We found success with a U-Net architecture which combines aggressive coarse-graining with skip connections that preserve fine-grained features and crucially can learn to generalize well from limited data.[36]

Channel structure for the U-Net used in Figures 1, 3, and 4. We set C=4, while L varies depending on input image size. Strided convolutions in the encoder layers have a stride of 4. The U-Nets in Figure 2 are the same, but do not have encoder block 2, skip block 3, or decoder block 2. They also have only 3 ConvNext blocks everywhere instead of 4.

Layer	Size in	Size out	Details
Prepended block	$1 \times L \times L$	$C \times L \times L$	Conv2d, 4× ConvNext blocks
Skip block 0	$C \times L \times L$	$C \times L \times L$	4× ConvNext blocks
Encoder block 0	$C \times L \times L$	$2C \times \frac{L}{4} \times \frac{L}{4}$	4× ConvNext, BN, Strided Conv2d, GELU
Skip block 1	$2C \times \frac{L}{4} \times \frac{L}{4}$	$2C \times \frac{L}{4} \times \frac{L}{4}$	(cf. skip 0)
Encoder block 1	$2C \times \frac{L}{4} \times \frac{L}{4}$	$4C \times \frac{L}{16} \times \frac{L}{16}$	(cf. encoder 0)
Skip block 2	$4C \times \frac{L}{16} \times \frac{L}{16}$	$4C \times \frac{L}{16} \times \frac{L}{16}$	(cf. skip 0)
Encoder block 2	$4C \times \frac{L}{16} \times \frac{L}{16}$	$8C \times \frac{L}{64} \times \frac{L}{64}$	(cf. encoder 0)
Skip block 3	$8C \times \frac{L}{64} \times \frac{L}{64}$	$8C \times \frac{L}{64} \times \frac{L}{64}$	(cf. skip 0)
Decoder block 2	$2C \times \frac{L}{64} \times \frac{L}{64}$	$C \times \frac{L}{16} \times \frac{L}{16}$	Upsample, Concat, 4× ConvNext, Conv2d
Decoder block 1	$6C \times \frac{L}{16} \times \frac{L}{16}$	$2C \times \frac{L}{4} \times \frac{L}{4}$	(cf. decoder 2)
Decoder block 0	$3C \times \frac{L}{4} \times \frac{L}{4}$	$C \times L \times L$	(cf. decoder 2)
Appended block	$C \times L \times L$	$2 \times L \times L$	4× ConvNext blocks, Conv2d

The channel structure of the U-Net is shown above. Most layers are composed of blocks with a ConvNext structure.[39] Briefly, they consist of a layer-wise convolution, batch normalization, an inverse-bottleneck depth-wise convolution, activation function, and finally a depth-wise convolution. Our ConvNext blocks have a layer-wise kernel size of 7 and increase channels in the inverse

bottleneck by a factor of 4. For all other convolutions, we use a kernel size of 3. Dropout is used with a dropout probability of 10%. A detailed illustration of the architecture is shown in Figure S2. The number of encoding (coarse-graining) layers was set by the minimal image size we processed in the paper. With a minimal image size of 64 × 64 and 4 × 4 downsampling convolutions, this restricted us to 3 encoding layers. The hyperparameters of the ConvNext blocks, including kernel size, inverse bottleneck width, and normalization layers, are chosen according to the optimal values found in.[39]

The U-Net is trained with the Adam optimizer with weight decay ("AdamW",[87]) with an initial learning rate of 0.001. The learning rate is scheduled to decay exponentially with rate 0.99. We use a batch size of 8.

For the U-Net used in Figures 1, 3, and 4, training data consists of 480 randomly sampled frames from time-lapse series of 16 cells (of 31 cells total). For the U-Nets used in Figure 2, training data consists of 600 randomly sampled frames with a variable number of cells for training (see "overview table of the training and testing data used in this work"). Each data sample contains an input image (either zyxin, another protein, the mask, or a two-channel zyxin + protein image) paired with the corresponding traction force map measured via TFM. Traction force maps have two channels, which we represent as magnitudes and angles rather than x and y components. In all cases, the network is trained for 300 epochs (passes through the entire training data set). As a loss function, we take the MSE for the magnitude component, and a 2π-periodic MSE for the angles.

Synthetic cells

We consider three variants of synthetic cell for the experiments shown in Figure 4. The first variant captures large-scale features of cell geometry. We generate cells of triangles whose sides are given by circular arcs. The cell shape is parameterized by L, the distance between the corners of the triangle, and R_c, the radius of curvature of the circular arcs. Forces measured in Figures 4G– 4I correspond to the average force across the area of the cell. These synthetic cells were fed as input to a U-Net trained on cell geometry.

The second class of synthetic cells models the distribution of focal adhesion-like objects in the cell. The intensity structure of these adhesions was chosen to match those of experimental adhesions, see Figure S10 for details. In each cell, ellipses of a given aspect ratio and area were randomly distributed (uniformly with a density of 60%) in a circle of fixed radius of 200 pixels ($\approx 34\mu$ m). Each cell is parameterized by the corresponding area and aspect ratio of the ellipses. Each ellipse had an intensity of 1, and they were allowed to overlap. Hence, the input image contained a range of (integer) intensities. Ellipse aspect ratio was defined relative to the radial direction, so probing aspect ratio in effect probed focal adhesion orientation. We evaluate the predicted force by calculating the average force on regions where a focal adhesion is present.

The role of zyxin intensity was probed by creating cells consisting of equidistant elliptical adhesions on a circular cell "background". These synthetic cells are parameterized by the intensity of the background B, the radius of the cell R, the angular density of focal adhesions D ($D = 1$ corresponds to no angular space between neighboring adhesions), and the length L and intensity I of focal adhesion ellipses. The intensity of the background models zyxin intensity in the cell away from focal adhesions. The zyxin intensity at focal adhesions typically has values in the range 4-12 (a.u.), while the background has values in the range 0-1. In Figure 4K, we show the change in intensity for $B = 0.8$ and $D = 0.5$; results do not strongly depend on B and D. To model the intensity profile of FAs seen in experiment, at the edges of the FA ellipses intensity increases linearly over 2 pixels until the specified FA intensity is reached. We evaluate the predicted force by calculating the average force on regions where a focal adhesion is present.

In the supplemental information, we show an additional variant of "synthetic cell" used to probe length-scale dependence of the neural network. These fake cells were generated by binarizing the zyxin images via thresholding, and applying dilations to achieve binary regions of different sizes. While this procedure does not afford fine control over the size of the resulting regions, it preserves some aspect of the distribution of FAs in the cell and results in sufficient regions that trends can be extracted. The results suggest that information is integrated over a length scale of $\sim 5\mu$ m, consistent with our other findings in this work. See Figure S11 for details.

Note all values of zyxin here and in the rest of the paper are given in units after normalization described in "Data processing" above.

Effective elastic model

We consider a model of the cell as an effective two-dimensional linear elastic medium. While originally introduced to model cells on micropillar arrays,[33] it has been extended to describe cells uniformly adhered to 2D substrates.[32] The free energy of the cell is

$$U = \frac{h}{2}\int dA\left(\sigma_{ij}^{\mathrm{el}} + \sigma^a\delta_{ij}\right)u_{ij} + \frac{1}{2}\int dA\, Y(x)u_iu_i \tag{1}$$

where $u_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ and $\sigma_{ij}^{\mathrm{el}}$ is the elastic stress tensor. h is the height of the cell, which is assumed to be small. As described in the main text, $Y(x)$ models a adhesion or pinning force which penalizes deformations, while σ^a serves as an active pressure term. Minimization of the elastic free energy leads to force balance equations for $\vec{u}(x)$:

$$h\partial_j\sigma_{ij}^{\mathrm{el}} = Y(x)u_i \quad (\text{in bulk}) \tag{2}$$

$$\sigma_{ij}^{\mathrm{el}}n_j = -\sigma^a n_i \quad (\text{on boundary}). \tag{3}$$

In addition to these conditions, we also require that σ^{el} and $\vec{u}$ are related via the constitutive relation

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\frac{\nu}{1-2\nu}\delta_{ij}u_{kk} + u_{ij}\right), \quad (4)$$

where E and ν are the effective Young's modulus and Poisson ratio, respectively, of the cell. Combining the force balance equations with the constitutive relation gives a PDE which determines $\vec{u}$.

Physical bottleneck

The physical bottleneck consists of a neural network step joined with a PDE-solver step. The neural network is implemented in the PyTorch library, and the PDE-solver is implemented with the dolfin-adjoint library.[88] At each step during training, we first predict a field $Y(x)$ and scalars σ^a and ν (the Poisson ratio, which we find to be nearly constant $\nu \approx -1$) using a neural network with zyxin as input. The convolutional neural network used to calculate Y is a shallow U-Net with structure shown in the table below. The network used to calculate the scalars consists of one convolutional layer which aggressively coarse-grains the image by a factor of 16, followed by fully-connected layers.

Channel structure for the physical bottleneck neural network. The top section describes the network used to predict the field $Y(x)$. Here, $C = 32$, all Conv2d layers have a kernel size of 5, and ConvNext blocks have kernel size of 15 and inverse bottleneck factor of 4. GELU is used as the activation function throughout. The bottom section describes the fully connected network used to predict the constants σ^a and ν. We use $N = 32$. In this network, every layer is followed by a ReLU activation.

Layer	Size in	Size out	Details
Prepended block	$1 \times L \times L$	$C \times L \times L$	Conv2d
Skip block	$C \times L \times L$	$C \times L \times L$	10 ConvNext blocks
Encoder block	$C \times L \times L$	$2C \times \frac{L}{4} \times \frac{L}{4}$	BN, Strided Conv2d, ReLU
Skip block 1	$2C \times \frac{L}{4} \times \frac{L}{4}$	$2C \times \frac{L}{4} \times \frac{L}{4}$	10 ConvNext blocks
Decoder block	$3C \times \frac{L}{4} \times \frac{L}{4}$	$1 \times L \times L$	Upsample, Concat, Conv2d
Strided Conv2d	$1 \times L \times L$	$16 \times \frac{L}{16} \times \frac{L}{16}$	Followed by flattening
FC	$\cdot L^2/16$	N	
FC	N	N	Layer repeated 10 times
FC	N	2	

The parameters output by the neural networks are mapped to a mesh (for spatially-varying parameters) after which they are fed as inputs to a PDE solver. To solve both forward PDE problems and derive adjoints (described in the following), we use the dolfin-adjoint library.[88] The PDE solver calculates a displacement field $\vec{u}(x)$ satisfying the PDE imposed by the physical model. Forces are calculated as $Y(x)\vec{u}(x)$ and compared to the experimentally measured values to give the loss $\mathcal{L}$, which is simply the mean-squared error. Gradients $\partial\mathcal{L}/\partial Y(x)$ etc. are computed using the adjoint method.

We briefly introduce the adjoint method,[54] a widely-used technique to optimize PDE parameters in control or data-assimilation tasks. We consider a PDE which acts on a field $u(x)$ and has parameters $p(x)$. One wants to optimize a function of the PDE's solution $J(u)$. This can be cast as a constrained optimization problem where one wants to minimize the Lagrangian

$$\mathcal{L}(u, v, p) = J(u) + \langle v, \mathcal{D}u \rangle.$$

Here $\mathcal{D}$ denotes the PDE we wish to optimize (which depends on $p(x)$) and $v(x)$, introduced as a Lagrange multiplier to enforce that u satisfies $\mathcal{D}u = 0$, is called the adjoint state. The angled brackets denote an inner product on the function space in which u and v live. Gradients of the Lagrangian $\partial\mathcal{L}/\partial p$ are given in terms of v, which is itself found by solving the adjoint PDE $\mathcal{D}^* p = f(u)$. The adjoint PDE is determined from the Euler-Lagrange equation $\partial\mathcal{L}/\partial u = 0$.

In practice, the adjoint equations are solved using automatic differentiation. We use dolfin-adjoint to calculate $\partial\mathcal{L}/\partial Y(x)$, $\partial\mathcal{L}/\partial\sigma^a$ and $\partial\mathcal{L}/\partial\nu$. These gradients are passed directly to PyTorch's autograd library to update the neural networks which predict $Y(x)$, σ^a and ν.

Green's function neural networks

Green's Function Neural Networks (GFNN) were implemented using the Pytorch Library. To predict traction forces with a GFNN, we used the Clebsch decomposition $\vec{F}_{NN} = \nabla\varphi + \xi\nabla\chi$, which is possible for any vector field. We hypothesized that each Clebsch variable was the solution to a linear partial differential equation (PDE) whose source was a function of the local zyxin density.

$$\mathcal{D}_\varphi \varphi = \rho_\varphi[\zeta]; \mathcal{D}_\xi \xi = \rho_\xi[\zeta]; \mathcal{D}_\chi \chi = \rho_\chi[\zeta] \quad (5)$$

To predict traction forces subject to this hypothesis, we trained a GFNN to compute each Clebsch variable. Under ((5)), each term is the integral of a source and a Green's function.

$$\varphi(\vec{x}) = \int d^2\vec{r}\, G_\varphi(\vec{x} - \vec{r})\rho_\varphi(\vec{r}); \xi(\vec{x}) = \int d^2\vec{r}\, G_\xi(\vec{x} - \vec{r})\rho_\xi(\mathbf{r}); \chi(\vec{x}) = \int d^2\vec{r}\, G_\chi(\vec{x} - \vec{r})\rho_\chi(\mathbf{r}) \quad (6)$$

For the network presented in Figure 6, we trained on three cells imaged under normal conditions and evaluated on an unseen cell to which the ROCK inhibition had been applied. The inputs were the zyxin density and the target outputs were traction force predictions from U-Net discussed in Figures 1, 2, 3, and 4. We center-cropped each input-output pair to a box of size 1024 × 1024 pixels, downscaled by a factor of 4, and applied a Fourier cutoff with $k_{max} = 50$. The GFNN used the zyxin field as input and learned to predict forces through the Clebsch decomposition and its corresponding Green's functions. It predicted the sources ρ_α using a shallow convolutional neural network and represented the Fourier-transformed Green's function as a three-channel 256×256 complex float tensor ($N = 256$ matches the downscaled images size in pixels). The complete network structure is shown in the table below.

GFNN architecture for cell force prediction. The network includes convolutional blocks inspired by the ConvNext architecture. Grouped convolutions accumulate local information within each channel, while 1×1 convolutions with the inverse-bottleneck structure enable the network to learn complex local functions at each pixel while maintaining a minimal receptive field.

Module	Layer	Channels	Details
Block 1	Conv2d	1→64	k3 × 3, groups = 64
	Conv2d	64→256	k1 × 1
	Sine		Activation function
	Conv2d	256→64	k1 × 1
Block 2	Conv2d	64→64	k3 × 3, groups = 64
	Conv2d	64→256	k1 × 1
	Sine		Activation function
	Conv2d	256→64	k1 × 1
Sources	Conv2d	64→3	k1 × 1
Integration	FFT2		$\{\varphi, \xi, \chi\}(\mathbf{q}) = G_i(\mathbf{q}) \cdot f_i(\mathbf{q})$
	Green's functions		
	IFFT2		
Output	Clebsch		$\mathbf{F} = \nabla\varphi + \xi\nabla\chi$

We trained the network for 200 epochs with batch size 8, learning rate $\lambda = 10^{-2}$ on the Green's functions and $\lambda = 10^{-4}$ on all other parameters. We used the Adam optimizer[89] and scheduled the learning rate to decrease by a factor of 10 whenever the loss function failed to improve for 10 epochs. The GFNN learned to minimize the following loss function with $\beta = 0.1$

$$\mathcal{L} = \sum \left(\vec{F} - \vec{F}_{GFNN}\right)^2 + \beta \sum |G^\alpha(\vec{q})|^2 \quad (7)$$

After training, we found that the φ field contributed minimally to the predictions. In Figure S3, we demonstrate that the $\nabla\varphi$ term in the Clebsch representation accounts for 1.1% of the overall traction force field and is not necessary for the GFNN to generalize to experimental perturbations. Because of this, we omitted it from our analysis in Figure 6.

To demonstrate the performance of GFNN as well as the Clebsch decomposition approach, in Figure S4 we train a GFNN to predict forces in a 2D Coulomb electrostatic system. When trained on synthetic data, the network learns to perfectly predict forces and learns Green's functions which agree well with the ground truth, i.e. the Coulomb force law.

Sparse regression

We performed sparse regression[90] in Python using the PySINDy library.[91,92] Our candidate library was informed by the Green's function neural network results. We assumed that the sources ρ_α were expressible as linear combinations of local zyxin gradients and approximated G_α using a set of radially-decaying functions. We used a set of local scalar derivatives $\rho_i \in \{\zeta, \nabla^2\zeta, (\nabla\zeta)^2, \zeta^2, \zeta\nabla^2\zeta, \zeta(\nabla\zeta)^2\}$ and chose the following candidate functions for the Green's functions.

$$G_i(r) \in \{r^{-1}, \log(r), r, e^{-\ell_\xi}, e^{-\ell_\chi}\} \quad (8)$$

The last two terms are exponentially decaying functions whose length scales ℓ_α were fit to the machine-learned G_ξ, G_χ in Figure 6D. We included the remaining three terms as slowly-decaying functions which might capture the long-range behavior of G_χ. From the set of sources ρ_i and Green's functions G_i, we constructed a library such that $\vec{F}$ could be represented as a linear combination of the following terms.

$$\vec{F}(\vec{x}) = \underbrace{\sum_{ij}\left[w^{\varphi}_{ij}\nabla\int d\vec{r}\, G_i(|\vec{x}-\vec{r}|)\rho_j(\vec{r})\right]}_{\nabla\varphi} + \underbrace{\sum_{ijk\ell}\left[w^{\xi\chi}_{ijk\ell}\left(\int d\vec{r}\, G_i(|\vec{x}-\vec{r}|)\rho_j(\vec{r})\right)\nabla\left(\int d\vec{r}\, G_k(|\vec{x}-\vec{r}|)\rho_\ell(\vec{r})\right)\right]}_{\xi\nabla\chi} \tag{9}$$

The weight grouping for $\xi\nabla\chi$ ((9)) is necessary as sparse regression is framed as a *linear* optimization problem. To obtain the weights $\vec{w}$, we used an elastic net objective.

$$\vec{w} = \text{argmin}\left[\langle\left(\vec{F}-\vec{F}(\vec{w})\right)^2\rangle + \alpha\|\vec{w}\|_1 + \frac{1}{2}\alpha\|\vec{w}\|^2\right] \tag{10}$$

Here, α is a parameter which sets the level of solution complexity, which we set to $\alpha = 0.75$. For training, we used the first 16 minutes of the cell movie (Day 3 cell 1, see first table), all of which were before the ROCK inhibitor was applies at $t = 30$ min. We performed sparse regression on 50,000 randomly-selected pixels in the 16 training frames (approximately 8% of the data in the training frames). This yielded an effective equation with 10 terms (see supplemental information). Figure 6G shows a sample prediction at $t = 40$ min, 10 minutes after the ROCK inhibitor was applied. Figure 6H summarizes the predictions from this equation over the entire movie.

A different choice of α yields equations of different complexity. For Figure 6I, we fit formulas using 17 values of α in the range $[10^{-4}, 10^{1}]$. We performed this procedure for each cell in the dataset and recorded the number of terms in the resulting formula and the mean-squared error (MSE) with experiment. As a baseline, we also recorded the MSE of the U-Net with experiment. To quantify how adding terms to the formula improves predictions, we defined the Relative Error metric in Figure 6I as $MSE_{\text{SINDy}}(\alpha) - MSE_0$, where MSE_0 is the error of a model with zero learnable parameters $\vec{F} = 0$, representing the $\alpha\to\infty$ limit. Thus, a model which uses more learnable parameters to achieve higher accuracy will have a negative relative error. To contextualize the performance of the sparse regression models, we compared the relative error of the learned equations to that of the U-Net and found that on average, a 10-term equation achieved 77% of the U-Net relative error.

QUANTIFICATION AND STATISTICAL ANALYSIS

Outlier determination

One cell in the actin dataset was an outlier, and was excluded from the calculation of the mean in Figure 2C. We tested for outliers using the Iglewicz-Hoaglin outlier test[93] and illustrate these results in Figure S7.

Optimal predictors and histogram plots

Probability distributions shown in angle and magnitude plots (for example, Figures 1E and 1F) are calculated by binning all pixels of all frames in the test set to calculate the number of joint occurrences of $\left|\vec{F}_{\text{exp}}\right|$ and $\left|\vec{F}_{\text{NN}}\right|$. The histogram is normalized to yield a probability and divided by the marginal distribution to calculate conditional probabilities. The "average" curves in Figures 1E, 1F, 2B, 2D, 3B, 3C, 3E, 3F, 4B, 4D, 5E, and5F are given by $C(F_N) = \mathbb{E}_{F_E}[F_E|F_N]$ (or analogously for angles).

In this work, we evaluate predictions by relying on conditional distributions $p(|F_{\text{exp}}||F_{\text{NN}}|)$. (In the following we consider only force magnitudes and write $|F| = F$, and we abbreviate $F_{\text{NN}} \equiv F_N$ and $F_{\text{exp}} \equiv F_E$ for brevity). This choice is motivated by the fact that, in the presence of noise, the conditional average $C(F_N) = \mathbb{E}_{F_E}[F_E|F_N]$ will satisfy $C(F_N) = F_N$ for a theoretically optimal predictor and will thus lie along the diagonal in the $F_E - F_N$ plane. On the other hand, $C(F_E) = \mathbb{E}_{F_N}[F_N|F_E]$ will generally not lie along this diagonal.

To see this, consider our dataset as a set of pairs $\{X^{(i)}, F^{(i)}\}$ indexed by i where F_i is force magnitude at some pixel, and X_i is the distribution of zyxin in a neighborhood of that pixel. The neighborhood is set by the receptive field of the neural network. Due to either biological or experimental noise, there is a joint (non-deterministic) distribution $p(X, F_E)$ from which our data is drawn. The loss function for the force predictions $F_N(X)$ can be written

$$L(F_N) = \mathbb{E}_X\mathbb{E}_{F_E}\left[(F_N - F_E)^2\middle|X\right] \equiv \mathbb{E}_X L_X(F_N).$$

The (Bayes) optimal predictor is one which optimizes, *for every X*,

$$F_N^*(X) = \underset{F_N}{\operatorname{argmin}} L_X(F_N).$$

In can be shown that L_X is minimized by $F_N^*(X) = \mathbb{E}_{F_E}[F_E|X]$. Note that for this to be valid for all X, our network must be sufficiently expressive, else our model would be constrained and we would not (necessarily) be able to satisfy this condition for all X independently. If we do indeed have sufficient (infinite) expressivity, this is the (Bayes) optimal predictor.

In the supplemental information, we show that with an optimal predictor, the conditional averages satisfy $C(F_N) = \mathbb{E}_{F_E}[F_E|F_N] = F_N$ and $C(F_E) = \mathbb{E}_{F_N}[F_N|F_E] = \mathbb{E}_{p(F'_E|F_E)}[F'_E]$. Here $p(F'_E|F_E)$ denotes the posterior predictive distribution $p(F'_E|F_E) = \int dX p(F'_E|X)p(X|F_E)$. We additionally show that even in the case of Gaussian random variables, the mean of the posterior predictive distribution $p(F'_E|F_E)$ is, for a given F_E, *smaller* than F_E. Thus even for an optimal predictor, the line defined by $C(F_E)$ lies below the diagonal. For this reason, we evaluate our predictions by considering $C(F_N)$ and its distance from the diagonal, which is 0 for an optimal predictor.

Supplemental figures

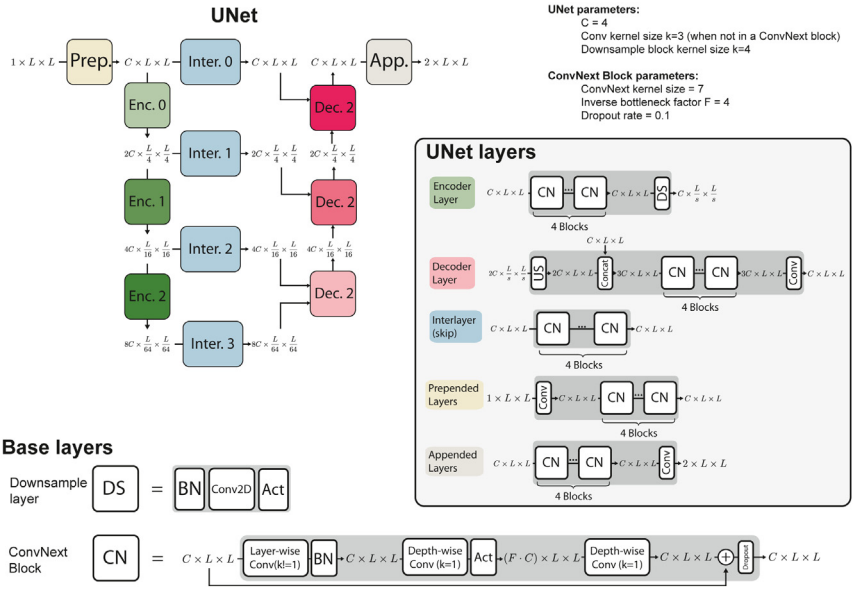


Figure S1. U-Net architecture, related to Figures 1, 2, 3, and 4

U-Net architecture augmented with ConvNext residual blocks. The notation (C, L, L) denotes the size of the image at each stage in the network, with C the number of channels and L the image size. Images are always square, and we must start with an image size which is multiple of 64 pixels. As an activation function, we use GELU (Gaussian error linear unit) throughout.

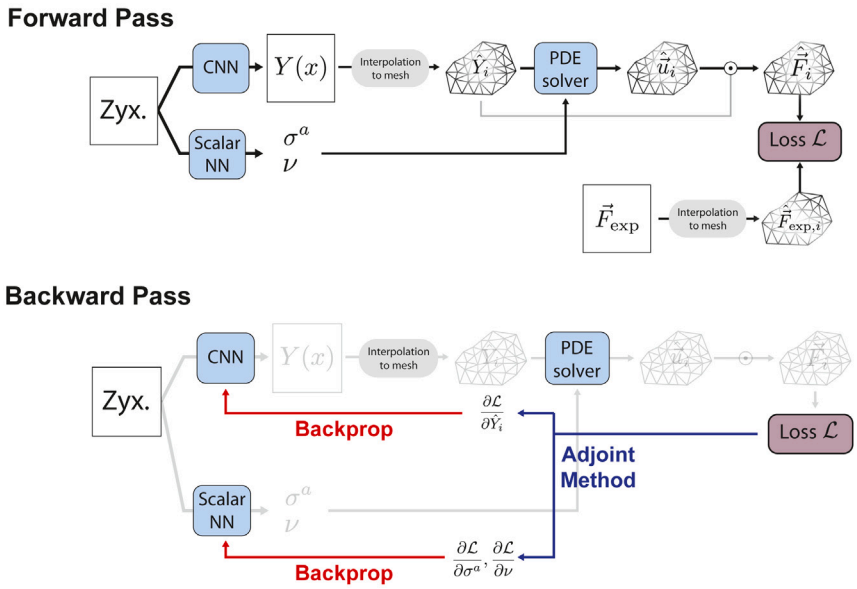


Figure S2. Physical Bottleneck Forward and Backward Passes, related to Figure 5

The physical bottleneck differs from other neural network-based parameter estimation techniques by enforcing a hard constraint, which stands in contrast to the soft constraints used by PINN-based approaches.[94] Hard constraints are enforced by training the network using gradients generated by the adjoint method described in the previous section. These gradients are then fed to PyTorch's automatic differentiation pipeline to pass them through the neural network.

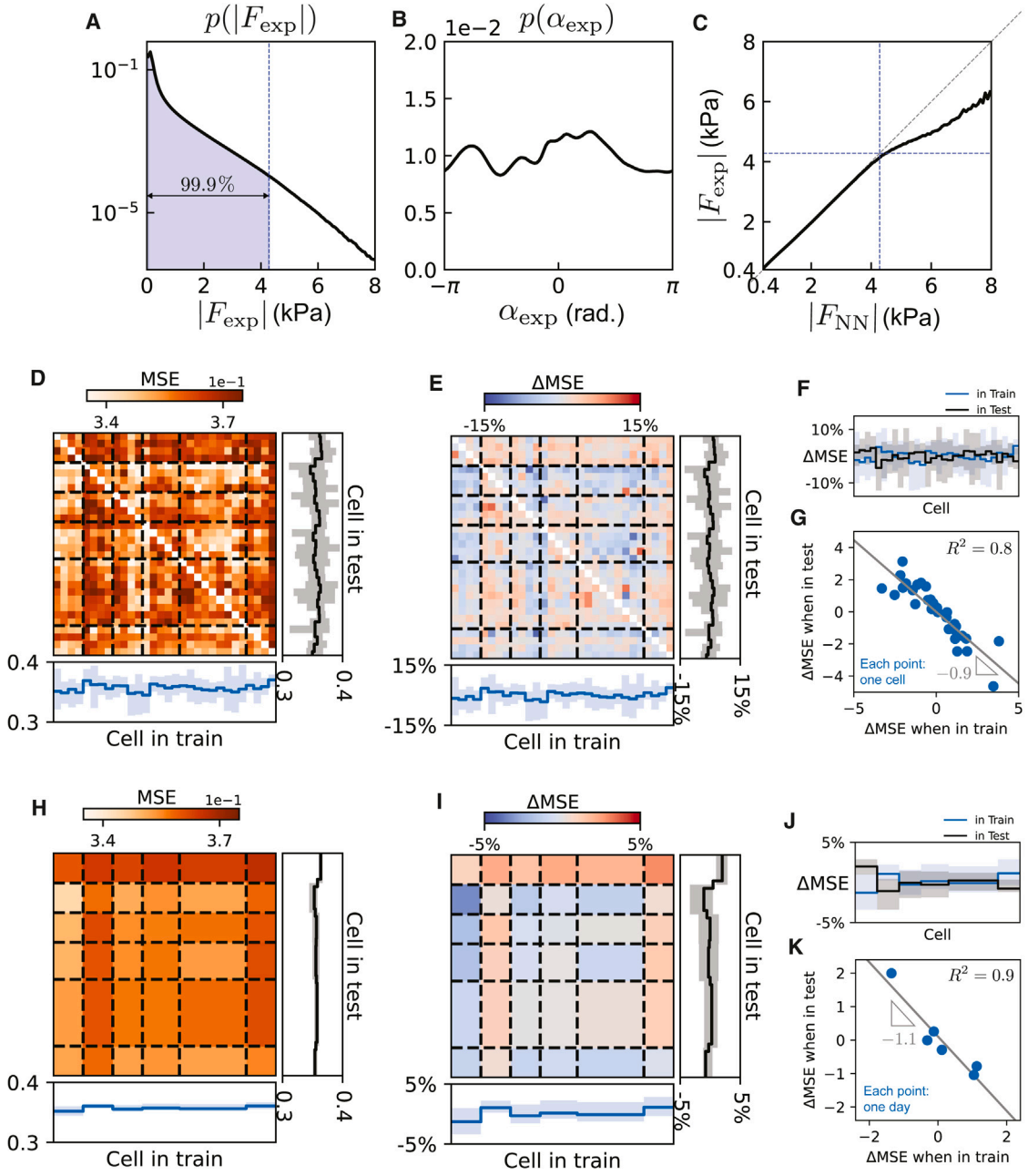


Figure S3. Distribution of force magnitudes and angles and MSE variations, related to Figure 1

(A) Histogram of distribution of force magnitudes in the test dataset for Figure 1. Vertical line denotes the 99.9th percentile of force magnitudes.

(B) Histogram of force directions in the test dataset.

(C) Average magnitude prediction (reproduction from Figure 1F in the main text) Vertical and horizontal dotted lines denote the 99.9th percentile of force magnitudes, as in (A).

(D) (Center) Reproduction of matrix in Figure 1H of the main text. (Right) for each cell in the test dataset, we calculate the mean-squared error (MSE) averaged over all possible cells in the train dataset (black solid line). In other words, we take the horizontal average over the "x" axis of the matrix. The shaded region denotes the maximum and minimum MSE for each test cell. (Bottom) We calculate the average performance across all *test* cells, for each cell in the training set, by averaging over the "y" axis of the matrix (center). Shaded region corresponds to maximum and mininum MSE for that particular training cell.

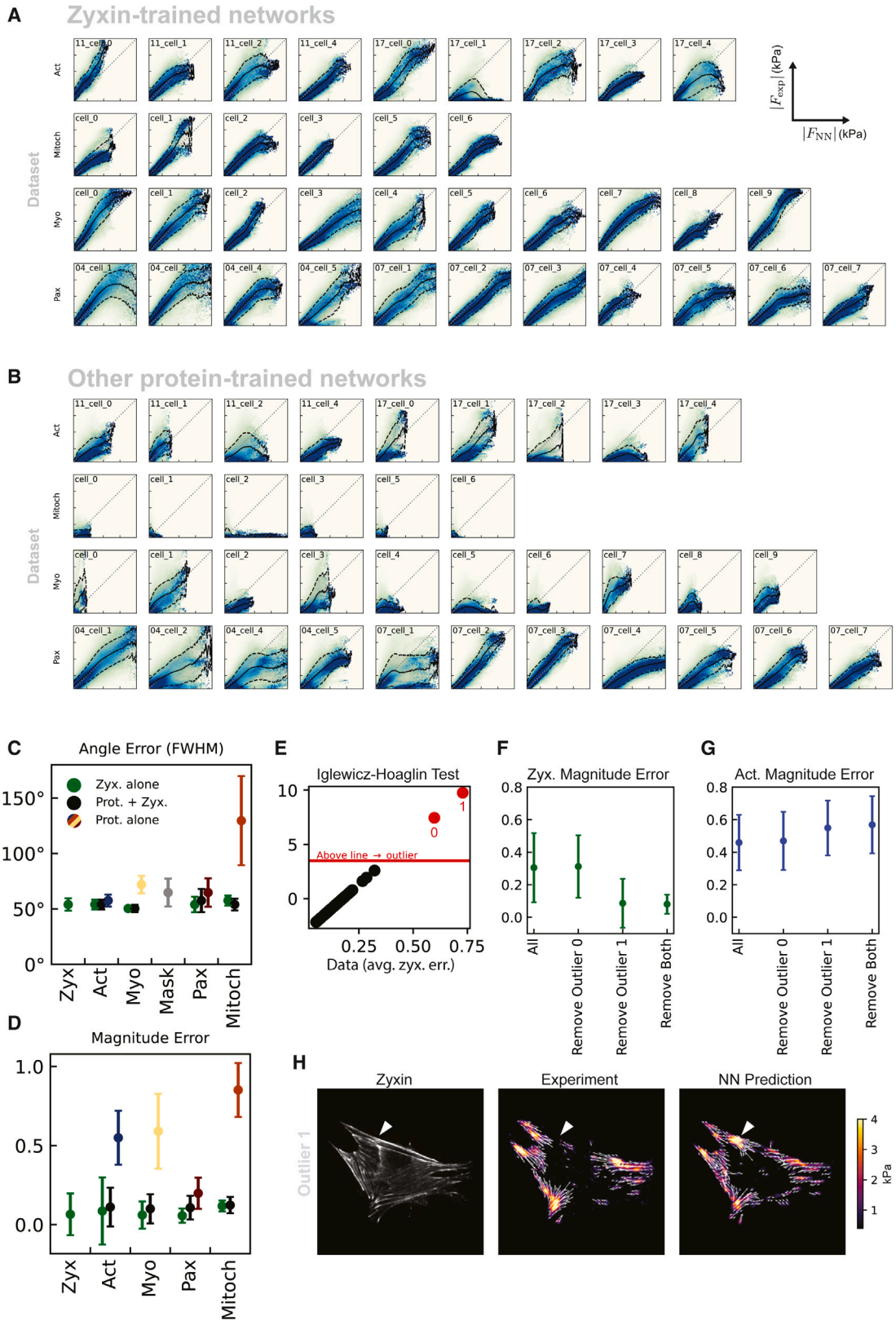
(E) Same as (D), but instead showing MSE relative to the average. For each entry in the matrix, $\Delta\ \text{MSE}_{ij} = \frac{\text{MSE}_{ij} - \langle \text{MSE} \rangle}{\langle \text{MSE} \rangle}$ where $\langle \text{MSE} \rangle$ is the total average MSE.

(F) We compare the average Δ MSE for each cell when it is in the test set (black) or the training set (blue). (These are the same curves as in (E) right and bottom.)

(G) Scatterplot of the average Δ MSE for each cell when it is in the test set versus the training set. For each cell, presence in the test or train set may result in a relative increase or decrease of MSE by roughly ±4%. Improvement when the cell is in the test versus train set is nearly perfectly anticorrelated, with a correlation

(legend continued on next page)

constant of -0.9. This indicates that the cells which are hard to predict (high test MSE) lead to better predictive models when they are included in the training set (lower average MSE when in train).
(H–K) Same matrices as in (D)–(G) but averaged over the experimental day on which the cells were imaged (delimited by dashed lines). Δ MSE varies by experimental day up to ±2% of the total average MSE.



(legend on next page)

Figure S4. Angle and magnitude errors for networks trained on different proteins, related to Figure 2

(A) and (B) Here we show distributions of forces for all models used to generate Figures 2B–2E. For each cell in a dataset, we train a model on all cells except the held-out cell. These distributions show the distribution of predicted magnitudes for the model evaluated on the held-out cell. (A) Models using zyxin as the input for the neural network. (B) Models trained using the other protein (actin, mitochondria, myosin, or paxillin) measured in each dataset.

(C) and (D) (C) shows the error in angles and (D) the error in magnitudes for the U-Nets trained on different proteins. Each protein is imaged jointly with zyxin, hence each protein has a comparable zyxin-trained network (green dots). Training on combinations of proteins (black) does not improve prediction. For each cell in the corresponding protein's dataset, we train a network and withhold that cell for testing. Error bars denote standard deviations of these networks on varying the test cell.

(E) Iglewicz-Hoaglin outlier test[93] values for the hold-one-out experiments on the actin dataset in Figures 2D and 2E of the main text.

(F) and (G) Change in magnitude error by removing one of the two identified outliers. Outlier 1 is responsible for a threefold increase in zyxin magnitude error, while changing the actin magnitude error only marginally.

(H) Predictions on the outlier 1 cell ("17_cell_1" in A and B). The exceptionally large error value is due to high forces predicted at one location where in experiment there are no forces (white arrow).

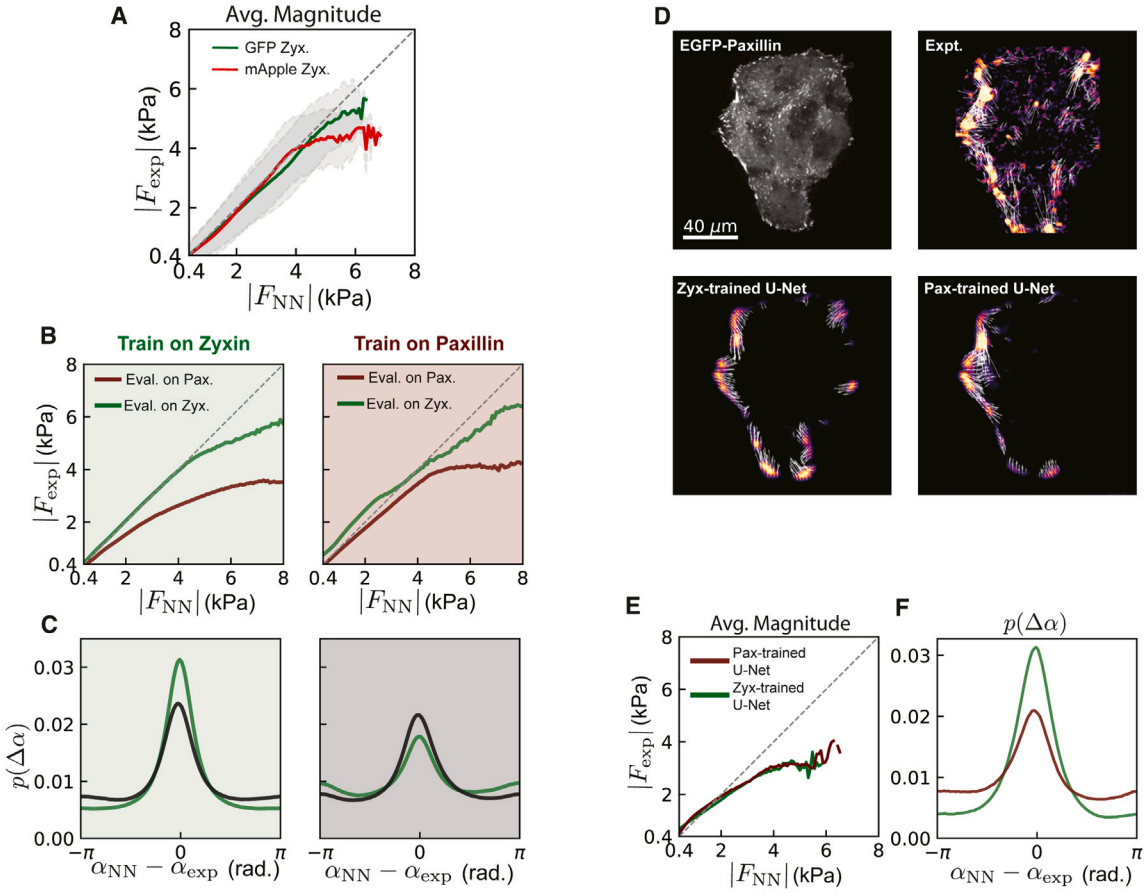


Figure S5. Comparison of zyxin- and paxillin-trained U-Nets, related to Figure 3

Zyxin and paxillin are both LIM-domain proteins which localize to focal adhesions. We find that these proteins are to some degree interchangeable, in the sense that a network trained on zyxin (paxillin) can still make accurate force predictions when evaluated on paxillin (zyxin). In Figures 2D and 2E of the main text, zyxin slightly outperforms paxillin in magnitude predictions.

(A) Part of this difference may be attributed to differences between fluorophores. Even when imaging zyxin only, mApple tends to slightly underperform GFP at high forces.

(B) and (C) In (B) and (C), we evaluate the predictions of zyxin- (left) and paxillin-trained (right) networks when made on both zyxin or paxillin images (green and red lines, respectively).

(D) MDCK colonies were imaged with EGFP paxillin. We predict forces on these colonies with our network trained on mApple-paxillin cells, as well as EGFP-zyxin cells (bottom row).

(E) and (F) Both networks predict force magnitudes similarly well, but the zyxin-trained network makes more accurate angle predictions (F).

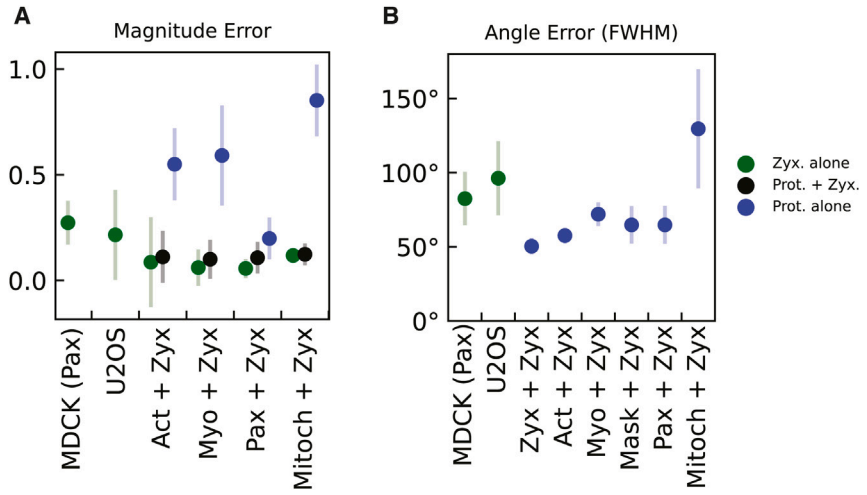


Figure S6. Error metrics for predictions on new cell types, related to Figure 3

(A) Magnitude errors (defined in Figures 2D and 2E of the main text) for predictions of fibroblast-trained U-Net when evaluated on MDCK cell clusters and U2OS cells. We show the results of networks trained on different proteins for reference (Figure 2E of the main text).

(B) Angle errors (full width half maxima) for predictions of fibroblast-trained U-Net when evaluated on MDCK cell clusters and U2OS cells.

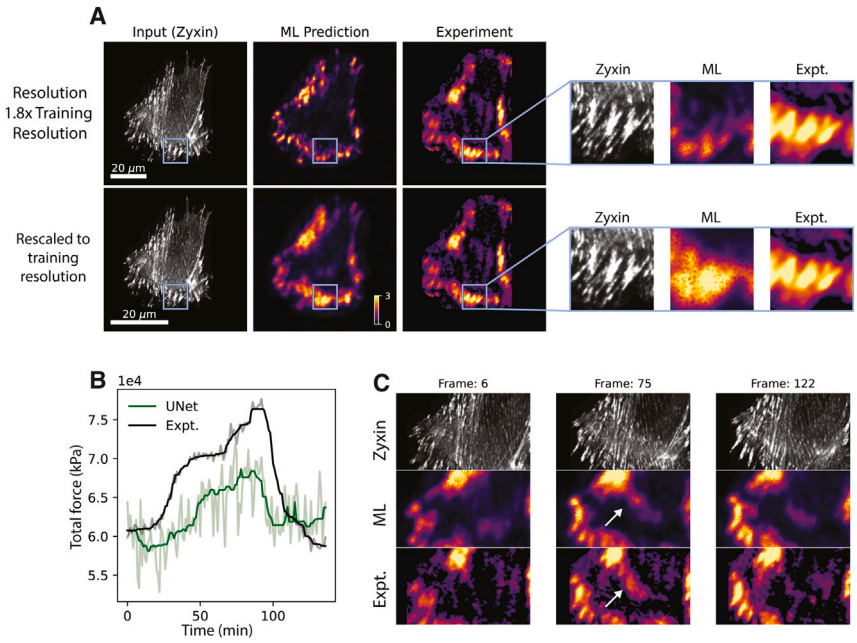


Figure S7. U-Net dependence on scale, and further generalizations, related to Figure 3

(A) We evaluated the network on cells imaged with a microscope of 1.8× the resolution on which the network was trained. (Top) As the input image is apparently larger (in terms of pixels) by a factor 1.8, predictions are less accurate and the network predicts extremely localized features (right). (Bottom) Predictive accuracy can be recovered by downsampling the image to the training resolution, after which the network predicts the correct large-scale distribution of forces.

(B) We optogenetically stimulate RhoA recruitment to a region of the cell,[28] which causes an increase in total force exerted by the cell. Solid curves are smoothed versions of the corresponding lighter curves.

(C) The optogenetic perturbation causes an accumulation of zyxin and the emergence of traction forces under the stimulated region (white arrow). Surprisingly, the U-Net accurately predicts these new forces within the cell despite seeing no such phenomena during training.

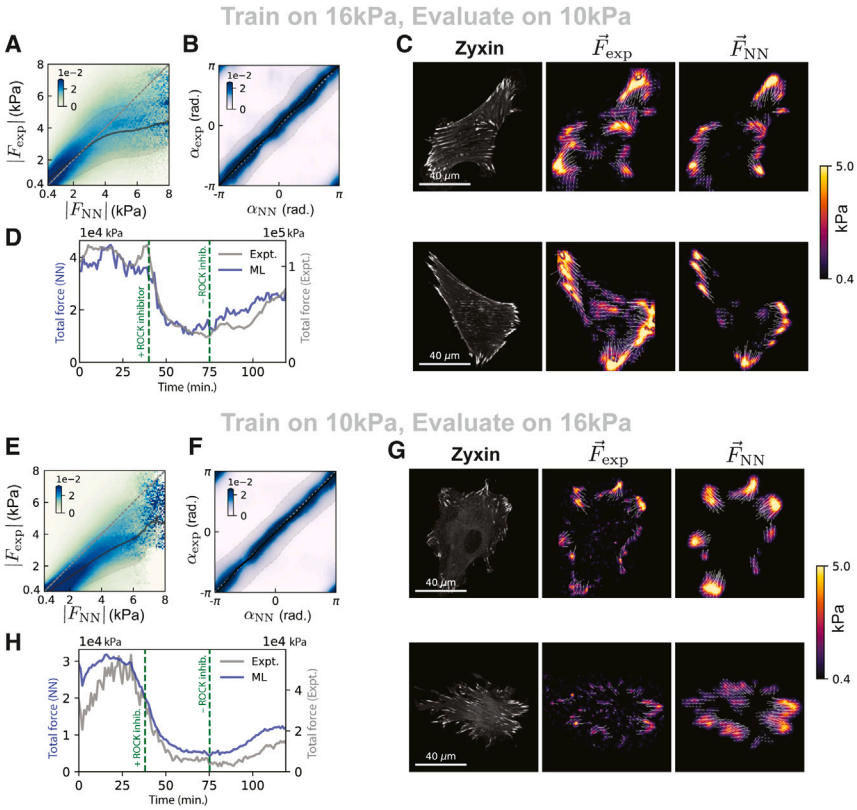


Figure S8. Evaluating a trained U-Net on cells adhered to substrates of different stiffness, related to Figure 3

We take a U-Net trained with cells adhered to a 16 kPa substrate (the same in Figures 1,3, and 4 in the main text) and evaluate the network's ability to generalize to cells adherent to 10 kPa substrate.

(A) Distribution of forces $p(|F_{exp}||F_{NN}|)$. Predictions are accurate on average (black line) up to 8kPa, albeit with slight underprediction at low forces (the average curve lies above the diagonal), and slight overprediction at high forces (average is below the diagonal).

(B) Distribution of angles $p(\alpha_{exp}|\alpha_{NN})$.

(C) Sample predictions for two different cells. Note that forces are more smeared out in the experimental data, likely due to the higher regularization parameter used for TFM with soft substrates. (D) The network captures the relative change in forces during ROCK perturbation experiments. The total force is slightly different, again likely due to the higher regularization parameter in experiment. (E) Force distribution for a network trained on 10 kPa cells, evaluated on 16 kPa. The average (black line) lay below the diagonal, indicating that the network is overpredicting magnitudes. This effect may be in part due to the effect of different regularization parameters, see Figure S4.

(F) Distribution of angles $p(\alpha_{exp}|\alpha_{NN})$.

(G) Sample predictions for one cell. While the locations of the predicted forces are mostly accurate, the predicted forces are much more spread out than in experiment. This is likely due to the larger regularization parameter in the data used to train the U-Net and may also be due to the smaller amount of data used to train the network.

(H) Despite these differences, the network correctly captures the ROCK inhibition experiment. Error bars in (A), (B), (E), and (F) correspond to one standard deviation in each direction.

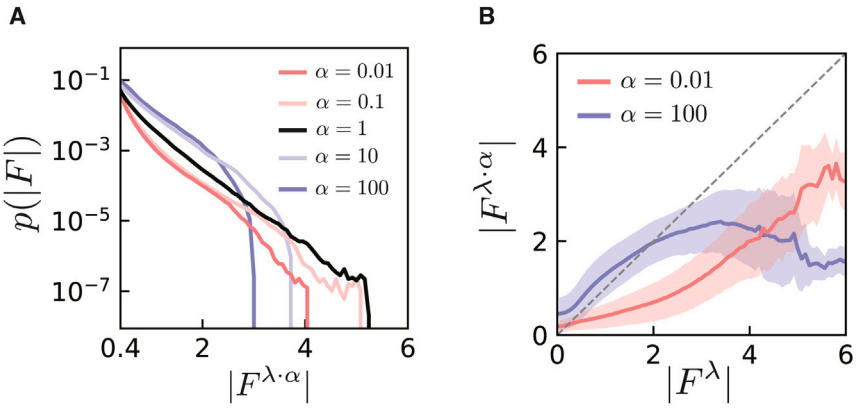


Figure S9. Impact of TFM regularization parameter, related to Figure 3

(A) TFM requires a choice of parameter λ that regularizes the inverse elastic Green's function which is used to compute forces from substrate displacements. The choice of regularization parameter impacts the forces found by TFM and determining the "correct" parameter is a matter of active research.[24] Here we show how the distribution of forces shifts due to varying regularization parameter $\lambda\cdot\alpha$. These are probability distributions for the *normalized* forces, using the normalization procedure as described in STAR Methods. Low-regularization forces are normalized by a larger amount, hence the entire distribution is scaled and shifted towards smaller forces: the red curves lie below the black curve. Thus, the effective forces in the low-regularization case are smaller.

(B) We directly compare the force maps generated with two different regularization parameters by generating the same histograms in Figure S4 where the "predicted" forces are those generated at a particular regularization λ and the "experimental" forces are generated at a different regularization $\lambda\cdot\ \alpha$. Even a "perfect" predictor appears to make errors by deviating from the diagonal. The behavior here is qualitatively similar to the observed curves in Figure S4. In particular, when evaluated on forces with a higher regularization parameter (blue; corresponding to Figure S4), small forces are slightly underpredicted while large forces are overpredicted. When evaluated on a lower regularization parameter (red; corresponding to Figure S4) forces appear to be overpredicted.

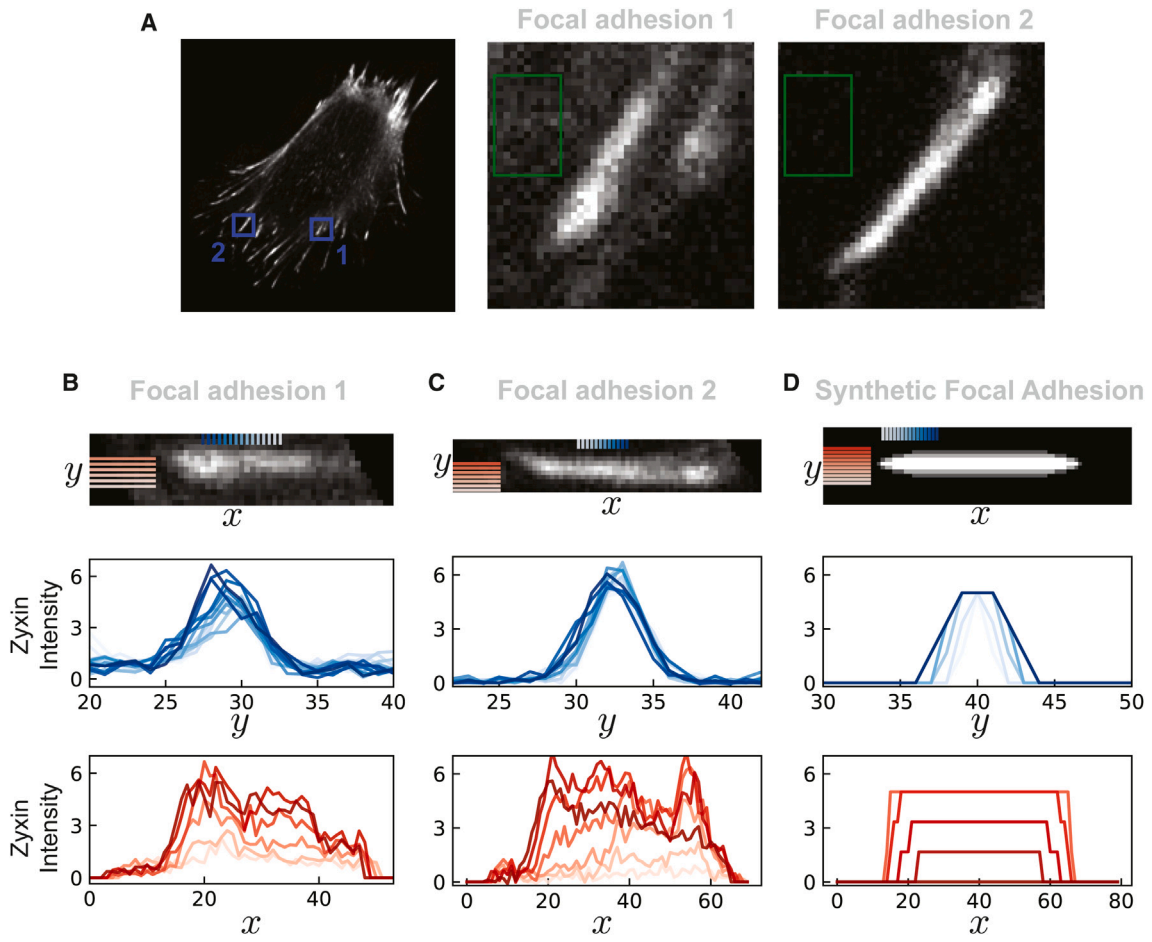


Figure S10. Synthetic cell construction, related to Figure 4

To create focal adhesions for our synthetic cells (Figure 4 of the main text), we attempt to mimic the structure of experimentally observed FAs.

(A) We select two FAs from a cell. The first is in a region of relatively high "background" zyxin intensity. The average intensity in the green box is 0.8, in contrast to the average intensity of 0.1 in the green box for FA 2.

(B) and (C) We align each FA with the x axis and examine its profile along the x and y directions shown in shades of red and blue, respectively.

(D) We approximate FAs as ellipses with a linearly decreasing intensity at the edges.

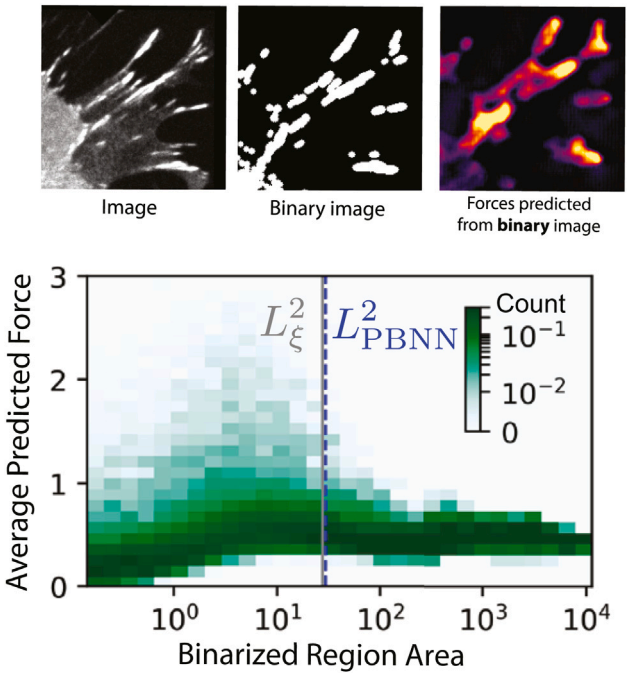


Figure S11. Additional synthetic cells: binarized cells, related to Figure 4

(A) We test the trained U-Net's response to more "natural" features by generating synthetic inputs to the neural network from actual cell images. To generate synthetic cells, we binarize the zyxin signal via thresholding, and then dilate the binary regions. This allows us to change the size of certain cell features, and by increasing the dilation size, certain features may merge to create yet larger structures. (B) We measure the average force F predicted for each connected binary region and show the distribution of $p(F|A)$ where A is region's area (in μm^2). The strongest force predictions occurs for structures of area 5 μm, consistent with the results of Figure 4J. We see that above $A \sim 50 \mu m^2$, the network no longer predicts large forces. This cut-off agrees with the short length scales learned by the physical bottleneck (L_{PBNN}), as well as the Green's function neural network L_ξ.

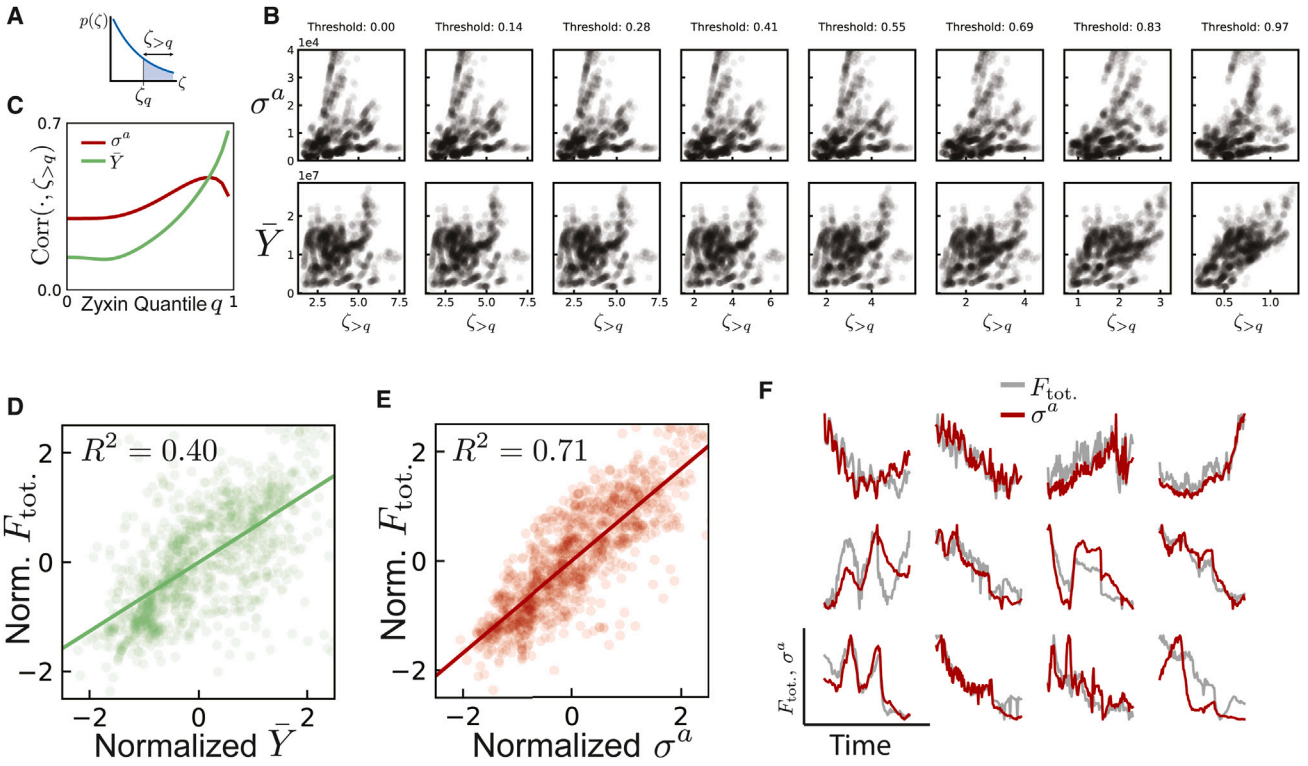


Figure S12. Supplementary analysis of physical bottleneck parameters, related to Figure 5

(A) For each image in the dataset, we calculate the total zyxin above a quantile q of the zyxin distribution, which we call ζ_q. The sum of zyxin above this quantile is $\zeta_{>q} = \sum_{\zeta > \zeta_q} \zeta$. A threshold (quantile) of 0.0 means we take the entire sum of zyxin in the image (far left), while a threshold (quantile) of 0.97 means we take the sum of the top 3% of zyxin values.

(B) For a given threshold value, we can generate scatter plots of σ^a or $\overline{Y}$ with ζ_q for each image. (C) For increasing quantile values, $\overline{Y}$ becomes increasingly well-correlated with total zyxin, suggesting that it depends only on the highest values of zyxin. In contrast, σ^a does not become increasingly correlated, suggesting that it is *not* solely a function of high zyxin values.

(D) We correlated normalized versions of average adhesion and total predicted force, $(\overline{Y} - \mu_{\overline{Y}})/\sigma_{\overline{Y}}$ where $\mu_{\overline{Y}}, \sigma_{\overline{Y}}$ corresponds to the mean and standard deviation of $\overline{Y}$ for all frames across the entire dataset (and analogously for the total force, F_{tot}).

(E) We find that σ^a is more closely correlated to F_{tot} than $\overline{Y}$. For clarity of presentation, outliers beyond 2 standard deviations are cropped in both (D) and (E).

(F) Illustration of the evolution of σ^a and the total predicted force as a function of time for each cell in the test set.