THE UNIVERSITY OF CHICAGO

LEARNING PHYSICAL MODELS OF BIOLOGICAL MATERIALS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF PHYSICS

BY
JONATHAN COLEN

CHICAGO, ILLINOIS
DECEMBER 2023

to my grandfather

If you're presenting something new and you're not terrified that you drew the wrong conclusion, then you have a major personality flaw

-Margaret Gardel

All this happened, more or less

-*Slaughterhouse-Five*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Vincenzo Vitelli, without whom this would not have been possible. Vincenzo has been an amazing role model and provided invaluable guidance and scientific insight, as well as overall warmth, kindness, and understanding. I am especially grateful for the freedom he gave me to explore and make mistakes that has been essential to my growth as a scientist. His boundless enthusiasm to sit down alongside me and do the time-consuming pushes to work past scientific roadblocks, prepare and practice presentations, and write and rewrite papers ad infinitum, is nothing short of extraordinary. Thank you.

I am grateful to my thesis committee, Margaret Gardel, Arvind Murugan, and David Miller, for their helpful comments. Margaret has also been a supportive mentor and collaborator, and I am so fortunate to have been able to work with such a brilliant scientist and exceptional role model during graduate school. I would also like to thank Patrick Oakes and Sebastian Streichan for the guidance and insights they have provided in various projects.

Thank you to the Vitelli group, especially for the many entertaining lunch breaks. Colin, Tali, Michel, Danny, Yael, David, Rituparno, Luca, Hamed, Bryan, and Maciej have been amazing co-workers, mentors, and friends. I am also grateful to my collaborators from other groups, including Steven, Stefano, Nikolas, Fridtjof, Matt, Noah, Rui, and others. Special thanks go to Matthew Schmitt and Ming Han for the countless hours they have spent debugging code, writing and revising manuscripts, and patiently explaining subtle (or in many cases obvious) details for the nth time. They have taught me so much over the years and working alongside them has been an absolute joy.

Beyond research, I thank my friends for their support and much-needed distractions from physics over the years. I am particularly grateful to Anora, Sharba, Jan, Joe, Syrian, Dan, Gabrielle, and Adam from Chicago, Andrew, Erika, Sooraj, Philip, and my fellow explorers of Radiator Springs from back home, and Mike, Lily, Julie, Colby, Sam, and Alex from

Atlanta. Special thanks to David for always being willing to visit or chat anytime, from anywhere, or about anything. Meg and Xander have been excellent role models and even better friends throughout grad school. Meg, thank you so much for everything, especially for encouraging me to eat produce occasionally and teaching me that scientists can and should express their fears. Xander, the warmth of your heart vastly exceeds even the length of your arms and the Canadian-ness of your passport. I am so happy you invited us to your table at Jimmy's five years ago.

I am grateful to my parents and sister for their endless encouragement and support throughout grad school. I would not have been able to do this without them. I would also like to thank my parents-in-law for welcoming me with open arms.

Thank you to Curie and Emmy for helpful discussions. Chloe, I love you. I cannot wait to begin the next chapter of our lives together.

ABSTRACT

Describing biological phenomena using physics is a difficult problem. Biological systems such as cells and tissues exhibit emergent behavior which is driven by genes, proteins, and the interplay and feedback loops between them. To capture biological behavior in a physical theory, one has to sort through this complexity, often by hand, and determine how to account for these sub-cellular interactions. This physical theory must then be connected to experimental reality, posing an even greater challenge. To reconcile the laws of physics with the complexity of nature, one must sift through large experimental datasets in order to find the critical details which enable connections between these two pictures.

This thesis presents machine learning as a tool to streamline this process. Our method of data-driven biophysical modeling combines physical theory, biological insight, and machine learning to characterize and understand diverse phenomena. Using experimental case studies on protein dynamics, cell mechanics, and fruit fly embryo development, we show how this approach can not only predict the future of complex systems, but also help uncover interpretable rules governing their behavior.

CHAPTER 1

INTRODUCTION

A central goal of physics is to understand the rules that govern what happens around us. To do this, physicists have built general frameworks, often relying on symmetries and conservation laws, to model behavior ranging from the very large to the very small. Classical continuum mechanics is one such framework which — despite ignoring significant complexity at molecular, atomic, and subatomic scales — has proven successful at describing solids, fluids, and many materials in between. Cells are decidedly non-classical. The mechanics of cells are driven by actions and feedback loops involving many interacting proteins and protein structures. To describe cell behavior within the framework of classical mechanics, one must connect the biochemical objects which make things happen (i.e. proteins) to the mechanics one aims to capture (i.e. cell motion). Determining these couplings for a given experimental system, either by coarse-graining a detailed microscopic model or by adding terms phenomenologically, is a difficult problem due to the complexity of not just the biochemistry and mechanics, but also the experimental data itself.

Machine learning (ML) has the potential to overcome these challenges by building complex models directly from experimental data. Indeed, ML has achieved superhuman performance on a variety of problems — chess, Go, racing simulators, and protein structure prediction, to name a few — by searching for and recognizing patterns within complex datasets. While such models are predictive, they are not explanatory. A deep neural network is a black-box and it is difficult to determine *what* it has learned or *how* it makes a prediction. Thus, despite the success of deep learning in biology, an open question remains: **How can we use ML as a tool for building interpretable physical models of biological systems?**

1.1 Thesis overview

The thesis is organized as follows. First, we provide overviews of key concepts used throughout this work. **Chapter 2** offers an introduction to **active matter**, a framework which extends classical mechanics to describe materials which exist out of equilibrium. We outline common active matter models and their connections to biological systems. **Chapter 3** introduces **deep learning** by explaining the building blocks of neural networks, describing several common network architectures, and discussing considerations for choosing which neural network to use for a given task. This material was adapted from a series of supplemental "primers" on deep learning written alongside each of our papers.

The following chapters demonstrate **data-driven biophysical modeling** through a series of case studies spanning biological scales. **Chapter 4** examines active nematics, a material composed of biofilaments and molecular motor proteins, and shows how neural networks can connect experiments to hydrodynamic theory as well as forecast complex chaotic behavior in a model-free manner. **Chapter 5** focuses on the mechanics of single cells. We use machine learning tools to identify relevant proteins for predicting mechanical behavior. We introduce two data-driven pipelines — one constrained by physics and one more agnostic — which build complementary interpretable models that predict cellular forces from protein distributions. **Chapter 6** examines the multi-cellular dynamics of developing *Drosophila* embryos. By applying our machine learning toolkit to experimental movies of embryo development, we learn a set of equations which describe how protein dynamics and genetic patterns govern morphogenesis, or how organisms develop their shapes.

CHAPTER 2

ACTIVE MATTER

This chapter is an introduction to **active** materials, whose components perform work at the microscopic level and create large-scale motion. A classic example is a collection of self-propelled particles – these could be microscopic bacteria, small battery-powered robots, or people walking in a crowd. In each of these cases, the agents convert chemical energy (ATP, a battery, or a hamburger) into mechanical work (flagellar motion, spinning wheels, or moving feet). This conversion process, called **mechanotransduction**, injects energy at the agents' scale and moves the system out of equilibrium. Many biological systems are active, ranging from epithelial tissues to bird flocks to the cellular cytoskeleton, which is composed of biofilaments driven by molecular motors. This chapter will present some common active matter behaviors, such as flocking, through the lens of continuum theory. Each case will demonstrate how to extend classical continuum mechanics models of **passive** materials to the **active** regime where detailed balance is broken at the microscopic scale.

Describing active matter using continuum equations, as one would for a passive fluid or solid, requires careful re-examination of the symmetries and conservation laws present (or absent) in the system. For example, active materials violate conservation of energy. The components create mechanical energy at the microscale which powers self-sustained active flows and stresses. We will show how to account for this energy injection and examine its consequences. We begin with an introduction to classical hydrodynamics, a framework for writing down coarse-grained equations constrained by symmetries and conservation laws. Next, we examine self-propelled systems which exhibit a flocking transition. We then introduce active nematics and place a special focus on the behavior of topological defects. We will briefly discuss how activity influences elasticity, demonstrating its relevance to solid mechanics as well as fluid mechanics. Finally, we cover active gel theory and apply it to describe the mechanics of the cell cytoskeleton.

2.1 Hydrodynamics

The Navier-Stokes equations can describe water flowing through a pipe without knowing anything about the individual H_2O molecules that comprise the flow. These molecules move around and collide with each other, but the typical length ($\sim 10^{-10}$ m) and time ($\sim 10^{-10}$ s) scales of their interactions are much smaller than those of the pipe flow we are interested in. **Hydrodynamics** is a theoretical framework for describing such systems at long distances and long timescales relative to their microscopic components. Despite its origins, hydrodynamics captures more than just water. Hydrodynamic modeling has been applied to systems whose particles range from atoms, molecules, cells, or even birds [34, 119, 154]. The goal of this framework is to predict the behavior of materials in a universal manner, independent of microscopic details such as the individual positions of their constituents.

A core assumption of hydrodynamic models is that systems achieve **local thermalization**. That is, a local region, whose size is much larger than the typical particle size or inter-particle separation, will reach thermal equilibrium over a timescale which is much larger than the typical collision (or interaction) time between particles. The system can be described by thermodynamic variables which are uniform within each local region. Hydrodynamics seeks to describe the total system by modeling the dynamics of each hydrodynamic degree of freedom within each box. The number of these degrees of freedom typically corresponds to the number of conserved quantities in a system. The **charge** of a conserved quantity is a suitable hydrodynamic variable. For example, momentum is conserved in classical fluids, and so momentum density is a **relevant variable** [34].

After identifying the degrees of freedom, we can determine their dynamics using **conservation laws** constrained by **symmetry**. As an example, consider a system with one conserved quantity: the particle number N. The corresponding conserved charge is the particle density $\rho(\mathbf{x})$, and so this is our lone hydrodynamic variable. The dynamics is described

by a conservation law [34]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \tag{2.1}$$

Here $\mathbf{J}$ is a current which we will define via a constitutive relation. A typical approach is to perform a **derivative expansion** and write $J_i(\mathbf{x}) = C_i\,\rho + D_{ij}\,\partial_j \rho + E_{ijk}\,\partial^2_{jk}\rho + O(\partial^3)$. When ρ varies slowly in space[1], we can truncate the expansion at leading order. At equilibrium[2], the density will be uniform and there should be no current. Thus, we can set $C_i = 0$. By assuming **rotational invariance**, we can further constrain $D_{ij} \sim \delta_{ij}$. This allows us to write a the hydrodynamic equation for our system

$$\frac{\partial \rho}{\partial t} - \nabla \cdot (D\,\nabla \rho) = 0 \tag{2.2}$$

This is a **diffusion** equation and the relation $\mathbf{J} = -D\nabla\rho$ is known as Fick's Law. Taking the diffusion coefficient D to be constant, we recover Fourier's heat equation.

The above example was a derivation for a system with one conserved quantity. Using a similar approach, one can derive other familiar hydrodynamic equations. For example, by considering a system whose mass, momentum, and energy are conserved, one can derive the well-known Navier-Stokes equations of fluid mechanics.[3]

2.2 Flocking

Flocking is a paradigmatic example of collective behavior in active matter. Active constituents, such as flying birds, organize into large complex patterns whose size is much greater than an individual bird. We will examine flocking first from the discrete particle/bird perspective before constructing a continuum mechanical theory following [193].

1. i.e. ρ has long-wavelength behavior

2. The infinite-wavelength limit

3. For a detailed derivation, see Chaikin and Lubensky Chapter 8.4 [34].

2.2.1 Discrete theory: Vicsek model

The Vicsek model was the first attempt to describe the physics of flocking [202]. This model takes a classic physicist's view and models birds as a set of spins. This is a similar approach to modeling a ferromagnet. Just as a ferromagnet can spontaneously magnetise, so too can the birds spontaneously align, or flock (Figure 2.1a). In the Vicsek model, the birds have a position $\mathbf{x}$ and a direction θ. Their dynamics follow the update rule

$$\mathbf{r}_i(t+dt) = \mathbf{r}_i(t) + \mathbf{r}_i\, dt \tag{2.3}$$

$$\theta_i(t+dt) = \frac{1}{N_R} \sum_{\Delta_{ij}<R} \theta_j(t) + \eta_i(t) \tag{2.4}$$

The first equation updates each bird's position, assuming it moves with a constant speed v in its direction θ – the birds are self-propelled. The second equation updates each bird's direction by taking an average over its neighbors within a distance R. The term η_i is Gaussian noise with $\langle \eta_i(t) \rangle = 0$ and $\langle \eta_i(t)\eta_j(t') \rangle = \Delta \delta_{ij} \delta(t-t')$.

The update rule (2.4) is relaxational dynamics. At each time step, the value at each point is replaced by an average over its neighbors. On a regular 2D lattice with spacing a, (2.4) is the discrete form of a noisy diffusion equation. It is identical to solving the following equation using a finite-difference method.

$$\partial_t \theta = D\nabla^2\theta + \eta \quad \left(D = \frac{a^2}{4}\right) \tag{2.5}$$

At low densities or high noise, the bird orientations are either disordered or correlated only within small regions. However, at high densities and low noise, the birds align and move along a single chosen direction (Figure 2.1b). This is known as the **flocking transition** [202].

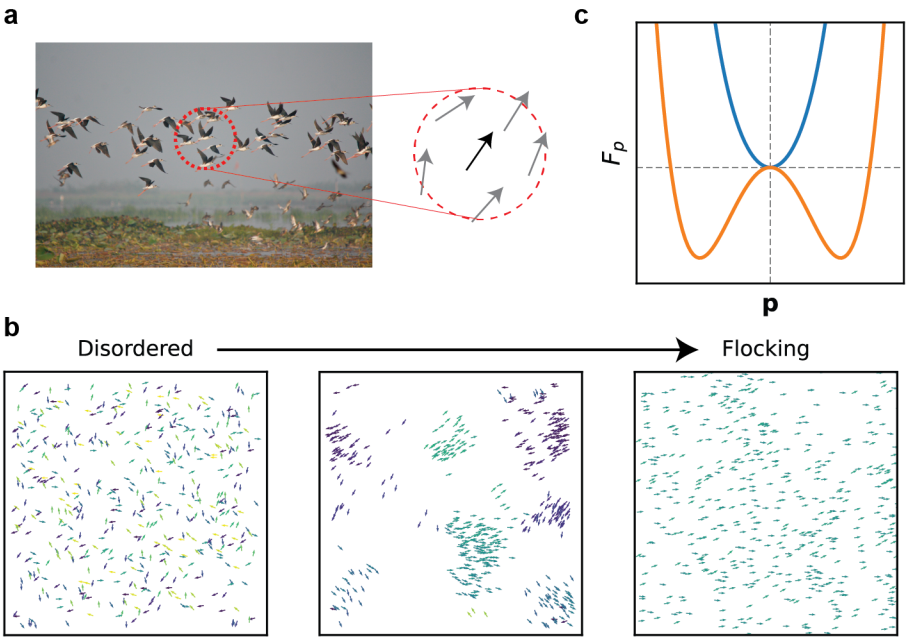

Figure 2.1: Modeling the flocking transition. (a) The Vicsek model models birds as spins which tend to locally align. (b) Spins exhibit a transition to uniform alignment along a chosen direction. (c) Toner-Tu Free energy F_p for $\alpha > 0$ (blue) and $\alpha < 0$ orange.

2.2.2 Continuum theory: Toner-Tu hydrodynamics

We can examine this flocking transition more closely using Toner-Tu theory, a continuum model which describes the collective behavior of the discrete birds defined by Equations. 2.3-2.4.

To build hydrodynamic theories, we start with symmetries and conservation laws. We assume no birds are born[4] or die, and so the number of birds is conserved. Because the birds are self-propelled, neither energy nor momentum are conserved. Thus, the hydrodynamic theory will have an equation for one conserved charge: the number density ρ. We also need an equation for the birds' motion direction $\mathbf{p}$ (also called polarization), which is a variable related to broken rotational invariance. We will discuss the significance of this later. To obtain our hydrodynamic variables ρ, $\mathbf{p}$, we must **coarse-grain** the variables in the Vicsek

4. The growth and development of flying organisms are covered in Chapter 6

model.

$$\rho(\mathbf{r},t) = \sum_n \delta(\mathbf{r} - \mathbf{r}_n(t)) \tag{2.6}$$

$$\mathbf{p}(\mathbf{r},t) = \frac{1}{\rho(\mathbf{r},t)} \sum_n \mathbf{p}_n(\mathbf{r},t)\delta(\mathbf{r} - \mathbf{r}_n(t)) \tag{2.7}$$

Here $\mathbf{p}_n$ is a vector for each birds' direction and points along its orientation angle θ_n. Although $\mathbf{p}$ relates to the birds' orientation, it is *not* required to be a unit vector. It is a local average of the birds' direction, and one can view its magnitude as a measurement of how aligned the birds are at point $\mathbf{r}$.

Following [119], it is convenient to write the dynamical equations for ρ and $\mathbf{p}$ in a way that separates equilibrium and non-equilibrium physics. The former can be written as arising from a free-energy functional F_p, leading to a compact form for the hydrodynamic equations.

$$\partial_t \rho + v_0 \nabla \cdot (\rho \mathbf{p}) = -\nabla \cdot \left(-\frac{1}{\gamma_\rho} \nabla \frac{\delta F_p}{\delta \rho} + \mathbf{f}_\rho \right) \tag{2.8}$$

$$\partial_t \mathbf{p} + \lambda_1 (\mathbf{p} \cdot \nabla)\mathbf{p} = -\frac{1}{\gamma} \frac{\delta F_p}{\delta \mathbf{p}} + \mathbf{f}_\mathbf{p} \tag{2.9}$$

Here, v_0 is the birds' self-propulsion speed, γ, γ_ρ are kinetic coefficients, and $\mathbf{f}_{\rho,\mathbf{p}}$ are noise terms. The term $\lambda_1(\mathbf{p} \cdot \nabla)\mathbf{p}$ resembles the nonlinear advective term in the Navier-Stokes equations. Here however, momentum is not conserved, and so there is no Galilean invariance to enforce $\lambda_1 = v_0$. Instead, λ_1 is a phenomenological coefficient related to the birds' properties, or more formally, the microscopic components of the system.

The free energy F_p can be defined phenomenologically. A common approach is to write all terms allowed by symmetry to leading order in gradients, in the spirit of Landau theory.[5]

$$F_p = \int d^2x \; \alpha |\mathbf{p}|^2 + \beta |\mathbf{p}|^4 + \frac{K}{2} |\nabla \mathbf{p}|^2 + \dots \tag{2.10}$$

5. This truncates additional, more complicated terms. The full expression is in [119]

We first consider F_p when $\mathbf{p}$ is uniform. Figure 2.1c, shows (2.10) for different signs of α. F_p is parabolic when $\alpha > 0$ and the free energy is minimized when $\mathbf{p} = 0$ and the net "magnetization" or orientation of the birds is $M \equiv \arg(\mathbf{p}) = 0$. This **disordered phase** has no net magnetization and the birds do not flock. When $\alpha < 0$, F_p is a wine-bottle potential and the disordered phase is unstable. The lowest-energy configuration occurs at any polarization $\mathbf{p}$ such that $p = \sqrt{\alpha/\beta}$. Thus, the birds will align with a finite polarization and achieve a net magnetization. This is the **ordered phase**, and the change from disordered to ordered phases is the **flocking transition**. There are infinitely many equivalent directions that the birds could flock along. The collective alignment of the flock along one chosen direction is an example of **spontaneous symmetry breaking**. Combining Equations 2.9-2.10 leads to the common form for the Toner-Tu equations[6] [193]

$$\partial_t \mathbf{p} + \lambda_1(\mathbf{p}\cdot\nabla)\mathbf{p} = -[\tilde{\alpha} + \tilde{\beta}|\mathbf{p}|^2]\mathbf{p} + \tilde{K}\nabla^2\mathbf{p} - \tilde{\nu}\nabla\frac{\rho}{\rho_0} + \frac{\tilde{\lambda}}{2}\nabla|\mathbf{p}|^2 - \tilde{\lambda}\mathbf{p}(\nabla\cdot\mathbf{p}) + \mathbf{f_p} \quad (2.11)$$

The first term on the right-hand side controls the flocking transition, while the second term is analogous to a viscosity term in the Navier-Stokes equations. The next two terms can be interpreted as gradients of an effective pressure – note however that this pressure is not a thermodynamic variable, but instead is defined using the density and polarization field. The fifth term does not have an analog in fluid mechanics, but is allowed because symmetry does not explicitly forbid it [193]. This is a common theme in deriving continuum equations for active matter, which typically resemble a corresponding passive system with additional terms allowed by lifting symmetry constraints [33, 119].

6. Here we have reinstated the terms omitted in Eq. 2.10

2.3 Active nematics

The Vicsek model and Toner-Tu theory describe **polar** active fluids – each bird has a head and a tail and flies in the direction of its head. We now turn to a system of **apolar** components which are head-tail symmetric. These **nematic** liquid crystals can be composed, for example, of granular rods or microscopic filaments [13, 41]. By adding molecular motors to this system, which attach to and pull nearby filaments, we obtain an **active nematic** [46, 119, 154]. Such materials are driven out of equilibrium by the energy injected by these motors at the microscopic scale. We will see that a hydrodynamic theory of active nematics which has just one leading-order term allowed by activity exhibits rich behavior, such as proliferating topological defects which are spontaneously self-propelled.

2.3.1 Continuum theory: active nemato-hydrodynamics

To construct a theory for active nematics, we will first consider the hydrodynamics of a passive nematic and then add further effects allowed by activity. We will assume constant density and conserved momentum in the passive case, which leads to an incompressible Navier-Stokes equation for the velocity $\mathbf{v}$. We require an additional equation for the nematic director $\mathbf{n}$, which is the coarse-grained filament orientation and is analogous to the polarization field in Toner-Tu theory. Note that the director has head-tail symmetry, so $\mathbf{n} = -\mathbf{n}$. The equations governing $\mathbf{n}$, $\mathbf{v}$ in a passive nematic are [13, 46, 119].[7]

$$\rho_0 \frac{D\mathbf{v}}{Dt} = -\nabla P + \eta \nabla^2 \mathbf{v} + \nabla \cdot \boldsymbol{\sigma} \tag{2.12}$$

$$\frac{D\mathbf{n}}{Dt} = -\frac{1}{\gamma}\left[\frac{\delta F}{\delta \mathbf{n}} - \left(\frac{\delta F}{\delta \mathbf{n}} \cdot \mathbf{n}\right)\mathbf{n}\right] \tag{2.13}$$

7. To hide complexity, we use D/Dt as a generalized time derivative. This includes the nonlinear advective term as well as an additional co-rotation term for $\mathbf{n}$. We also assume incompressibility $\nabla \cdot \mathbf{v} = 0$

The stress $\boldsymbol{\sigma}$ in (2.12) in a passive nematic is an elastic stress which couples flow dynamics to gradients in the nematic director. It is related to the molecular field, which is the derivative of the Frank free energy F [13, 41]

$$F = \frac{K}{2} \int \left[(\nabla \cdot \mathbf{n})^2 + (\mathbf{n} \cdot (\nabla \times \mathbf{n}))^2 + (\mathbf{n} \cdot (\nabla \times \mathbf{n}))^2 \right] d\mathbf{r} \tag{2.14}$$

Here the coefficient ρ_0 is the density and η, γ are the fluid and rotational viscosity, respectively. The Frank free energy is minimized when $\mathbf{n}$ is uniform and the filaments align. The terms in the integral are the three deformation modes allowed by symmetry: splay, twist, and bend. The coefficient K is an elastic modulus which determines how much the material resists deformation. Equation 2.14 yields a right-hand-side in (2.13) of $-\frac{K}{\gamma}\nabla^2\mathbf{n}$. Thus, a passive nematic's director undergoes relaxational dynamics.

To extend these equations to describe an active nematic, we need to introduce terms to describe the active forces from molecular motors. These forces inject energy and move the system out of equilibrium, and are therefore distinct from the elastic stresses which minimize the Frank free energy. The leading order contribution of the motors is to add an active stress to (2.12) $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\text{elastic}} + \boldsymbol{\sigma}^{\text{active}}$.

$$\sigma_{ij}^{\text{active}} = \alpha \; n_i n_j \tag{2.15}$$

We can understand this term through a symmetry argument that bypasses the algebraic complexities of the nemato-hydrodynamic equations [154]. The stress $\boldsymbol{\sigma}$ is a symmetric tensor, and so the leading-order active stress is the simplest symmetric term arising from the motor forces. Because the motors pull filaments along each other, their force will be directed parallel to $\mathbf{n}$, leading to the active stress (2.15). The coefficient α is the activity parameter which sets the strength of motor-induced stresses. This can be related to the density of motors, their pulling strength, their movement speed, or other microscopic effects. Despite

the apparent simplicity of (2.15), this single active term describes much of the complex dynamics of active nematics.

2.3.2 Nonequilibrium behavior: active nematic "turbulence"

Active nematic flow is governed by two competing stresses: active and elastic. The elastic stress works to align the filaments while the active stress tries to deform them. The interplay between the two sets the amount of activity-induced disorder in the material. We can use dimensional arguments to determine the winner of this competition. The strength of elastic stresses is set by K, the elastic modulus, which has units $[K] = \text{energy} \cdot \text{length}^{-1} = \text{force}$. The active stresses are set by α, the activity parameter, which has units $[\alpha] = [\sigma^a] = \text{force} \cdot \text{length}^{-2}$. The typical forces for a system of size L are thus $F^{\text{elastic}} \sim K$ and $F^{\text{active}} \sim \alpha L^2$. Equating these two gives a critical activity

$$\alpha_c \sim K/L^2 \tag{2.16}$$

When $\alpha > \alpha_c$, activity wins the competition and the system enters a chaotic state known as **active turbulence**.[8] When $\alpha < \alpha_c$, elastic stresses win and the system remains aligned in a **quiescent** state. We can rearrange the above equation to obtain an active length

$$\ell_\alpha \sim \sqrt{K/\alpha} \tag{2.17}$$

Simulations and experiments have found that characteristic lengths in active nematics are proportional to ℓ_α, including the director and velocity correlation lengths [82, 189, 190].

8. Active turbulence is a distinct phenomenon from classical fluid turbulence. In high Reynolds number turbulence, energy cascades from long to short length scales. The opposite happens in active media. Energy is injected at the microscopic level and cascades upwards to long length scales. This novel energy cascade is of much interest, see [4, 5]

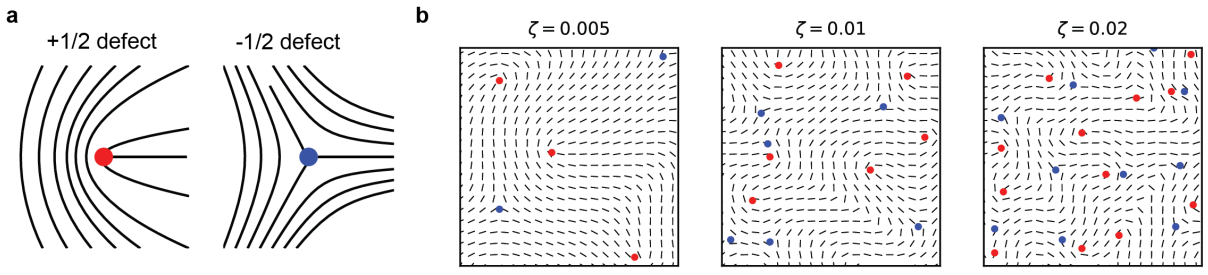

Figure 2.2: Active nematics. (a) Topological defects appear in opposite-charged pairs. (b) Increasing activity results in a greater defect density.

2.3.3 Active self-propulsion of topological defects

Nematic liquid crystals can have topological defects, or points where the nematic order breaks down and cannot be restored via a smooth deformation of the director field. A hallmark of active turbulence is the spontaneous formation of opposite-charged topological defect pairs with a defect density that scales like $n_d \sim 1/\ell_\alpha^2$ (Figure 2.2). Here, we will show that activity causes the defects to behave differently depending on their charge. $+1/2$ defects self-propel along their symmetry axis while $-1/2$ defects are more stationary.

Before diving into equations, we can try to understand the charge-dependent motility through symmetry arguments. $+1/2$ defects are polar – they are comet-shaped with a head and a tail (Figure 2.2). This creates imbalanced flows and a net propulsion. $-1/2$ defects have no polarity. The activity-induced flows are balanced and there is no self-propulsion. To make this symmetry argument more convincing, we will explicitly derive the motor-induced flows around a topological defects, following [68]. First, we make the (experimentally-justified) assumption that defect dynamics are overdamped, obeying the equation of motion

$$\zeta\left(\frac{d\mathbf{r}}{dt} - \mathbf{v}\right) = \mathbf{F} \tag{2.18}$$

Here $\mathbf{r}$ is the defect position, $\mathbf{v}$ is the local flow field, and $\mathbf{F}$ is the net force due to interactions

with other defects or external perturbations. We assume that the defects are far apart and that the flow relaxes much faster than the director ($\eta \ll \gamma$). This lets us ignore $\mathbf{F}$ and compute $\mathbf{v}$ using a Stokes equation.

$$\eta \nabla^2 \mathbf{v} - \nabla p + \nabla \cdot \sigma = 0, \qquad \nabla \cdot \mathbf{v} = 0 \tag{2.19}$$

The Stokes equation is linear and its solution can be decomposed $\mathbf{v} = \mathbf{v}^0 + \mathbf{v}^a + \mathbf{v}^e$, where $\mathbf{v}^0$ is the homogeneous solution ($\sigma = 0$) while the latter two terms are the flows from active and elastic stresses respectively. We compute them by convolving the active and elastic forces with a Green's function

$$v_i^\alpha(\mathbf{r}) = \int d^2 r' \, G_{ij}(\mathbf{r} - \mathbf{r}') \, f_j^\alpha(\mathbf{r}') \tag{2.20}$$

Here $\mathbf{f}^\alpha = \nabla \cdot \boldsymbol{\sigma}^\alpha$ is the force per unit area due to the respective stress term $\boldsymbol{\sigma}^\alpha$. The Green's function for the Stokes equation is the Oseen tensor, given in 2D by

$$G_{ij}(\mathbf{r}) = \frac{1}{4\pi\eta} \left[\left(\log \frac{\mathcal{L}}{r} - 1 \right) \delta_{ij} + \frac{r_i r_j}{r^2} \right] \tag{2.21}$$

$\mathcal{L}$ is a length scale dependent on the boundary condition. For a defect with charge k we define the director $\mathbf{n} = (\cos k\phi, \sin k\phi)$ and assume uniform concentration around the defect core. The active force $\mathbf{f}^a = \nabla \cdot \sigma^{\text{active}}$ is

$$\mathbf{f}^a = \nabla \cdot \sigma^a = \frac{\alpha}{2r} \begin{cases} \hat{\mathbf{x}} & k = +1/2 \\ -\cos 2\phi \hat{\mathbf{x}} + \sin 2\phi \hat{\mathbf{y}} & k = -1/2 \end{cases} \tag{2.22}$$

We can integrate the Stokes equation to obtain the active backflow [68]

$$\begin{aligned}
\mathbf{v}_+^a(r,\phi) &= \frac{\alpha}{12\eta}\left\{[3(R-r)+r\cos 2\phi]\,\hat{\mathbf{x}} + r\sin 2\phi\hat{\mathbf{y}}\right\} \\
\mathbf{v}_-^a(r,\phi) &= \frac{\alpha r}{12\eta R}\left\{\left[\left(\frac{3}{4}r-R\right)\cos 2\phi - \frac{R}{5}\cos 4\phi\right]\hat{\mathbf{x}} + \right. \\
&\quad \left. + \left[\left(\frac{3}{4}r-R\right)\sin 2\phi + \frac{R}{5}\sin 4\phi\right]\hat{\mathbf{y}}\right\}
\end{aligned} \tag{2.23}$$

The flow at the defect core is $\mathbf{v}_0 = \mathbf{v}(0,\phi)$. For $\pm 1/2$ defects, the propulsion velocity is

$$\mathbf{v}_0 = \begin{cases} \frac{\alpha R}{4\eta}\hat{\mathbf{x}} & k = +1/2 \\ 0 & k = -1/2 \end{cases} \tag{2.24}$$

Active stresses do not propel $-1/2$ defects. Any motion they exhibit is caused solely by elastic stresses coming from defect- or boundary-induced director distortions in the medium which are also present in the passive case. On the other hand, active stresses do propel $+1/2$ defects along their symmetry axes. The sign of α determines whether they move towards their head or their tail.[9]

2.4 Active solids

Until now, this chapter has focused on active fluids. We now turn to solids composed of active agents, following [165]. Just as we did for active nematics, we will re-evaluate classical continuum theory and lift certain constraints, such as energy conservation. This modifies the stress-strain relation and allows new elastic moduli to appear. These odd elastic moduli lead to novel phenomena including auxetic behavior and self-sustained energy cycles [59, 165].

9. $\alpha > 0$ for contractile systems and $\alpha < 0$ for extensile systems

2.4.1 Continuum theory: Active or Odd elasticity

Traditional elasticity assumes energy is conserved and therefore one derives forces by taking the gradient of a potential. Active or **odd elasticity** does not enforce energy conservation, as active forces inject energy and are typically non-conservative. Consider a general constitutive relation, which relates stresses and strains.

$$\sigma_{ij} = K_{ijkl}\, u_{kl} \tag{2.25}$$

Here K_{ijkl} is the elastic modulus tensor, σ_{ij} is the stress, and u_{kl} is the strain tensor $u_{kl} = (\nabla_k u_l + \nabla_l u_k)/2$. Conservative forces can be written as the derivative of a strain energy f, i.e. $\sigma_{ij} = \frac{\delta f}{\delta u_{ij}}$. If all forces are conservative and come from a general quadratic strain energy $f \sim C_{ijkl}\, u_{ij}\, u_{kl}$, we find $\sigma_{ij} = \frac{1}{2}\left(C_{ijkl} + C_{klij}\right) u_{kl}$ and therefore

$$K_{ijkl} = K_{klij} \tag{2.26}$$

Odd elasticity does not assume that all forces are conservative. Non-conservative forces, which cannot be represented as derivatives of an energy, are also allowed. We decompose the elastic modulus tensor $K_{ijkl} = K^e_{ijkl} + K^o_{ijkl}$ where

$$K^e_{ijkl} = K^e_{klij} \qquad K^o_{ijkl} = -K^o_{klij} \tag{2.27}$$

K^e_{ijkl} contains even moduli arising from conservative forces. It is symmetric under the interchange $ij \leftrightarrow kl$. K^o_{ijkl} holds odd elastic moduli which can arise due to non-conservative forces. These are anti-symmetric, or odd, under $ij \leftrightarrow kl$. Active elastic solids exhibit behavior which comes from this new antisymmetric component of the elastic modulus tensor.

In the special case of an isotropic 2D solid, the modulus tensor K_{ijkl} has 16 independent components while the strain u_{kl} has 4 components. u_{kl} is a 2×2 matrix, but we can represent

it as a column vector using.a basis of Pauli matrices. The 4 Pauli matrices are given by

$$\boldsymbol{\tau}^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \boldsymbol{\tau}^1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \boldsymbol{\tau}^2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \boldsymbol{\tau}^3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.28}$$

These matrices are *orthogonal* and obey the identity $\boldsymbol{\tau}^\alpha \cdot \boldsymbol{\tau}^\beta = 2\delta^{\alpha\beta}$. They correspond to 4 deformation modes: dilation, rotation, and two shears.

We can write any matrix u_{ij} as a column vector U^α such that $u_{ij} = U^\alpha \tau^\alpha_{ij}$. We can similarly write stress as a column vector $\sigma_{ij} = \sigma^\gamma \tau^\gamma_{ij}$. Here, the Pauli matrices correspond to stress modes of pressure, torques, and two shear stresses. In this basis, the constitutive relation is more compact

$$\sigma^\alpha = K^{\alpha\beta} U^\beta \tag{2.29}$$

$$K^{\alpha\beta} = \frac{1}{2}\, \tau^\alpha_{ij}\, K_{ijkl}\, \tau^\beta_{kl} \tag{2.30}$$

The elastic modulus tensor is a matrix in this basis, given by[10]

$$\begin{pmatrix} \\ \\ \\ \\ \end{pmatrix} = \begin{pmatrix} B & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ 0 & 0 & \mu & K^{\mathrm{o}} \\ 0 & 0 & -K^{\mathrm{o}} & \mu \end{pmatrix} \begin{pmatrix} \\ \\ \\ \\ \end{pmatrix} \tag{2.32}$$

Isotropy requires that many terms are zero. Solid body rotations create no stress, so $K^{\alpha 1} = 0$. The off-diagonal quadrants are zero as shear does not couple to rotation or dilation. B and μ are the familiar bulk and shear moduli from passive elasticity. The

10. Adopting the geometric notation from [165]

$$\sigma^\alpha = \begin{pmatrix} \\ \\ \\ \\ \end{pmatrix} \qquad u^\beta = \begin{pmatrix} \\ \\ \\ \\ \end{pmatrix} \tag{2.31}$$

remaining terms A and K^o are allowed when the constraint $K_{ijkl} = K_{klij} \to K^{\alpha\beta} = K^{\beta\alpha}$ is lifted. A couples dilation to torques while K^o rotates the shear stress response relative to the applied strain. These are the new odd elastic moduli.

2.4.2 Active engines with odd elasticity

When energy is conserved, the work done by elastic forces is a state function. It depends only on the initial and final configuration, i.e. strains, of the material. This is no longer true in an active solid, where the work depends on the path in strain space. With odd elasticity, work can be extracted from, or lost to, the system via **quasistatic strain cycles**. To see this, take $A = 0$ and consider the work done over some cyclic path in strain space

$$W = \oint \sigma_{ij}\, du_{ij} = \oint \sigma^\beta\, du^\beta = \iint dA\, \epsilon^{\alpha\beta} \frac{\partial \sigma^\beta}{\partial u^\alpha} \tag{2.33}$$

Only the odd components $K^{\alpha\beta} = K^o \epsilon^{\alpha\beta}$ will remain leading to

$$W = \iint dA\, \epsilon^{\alpha\beta}\epsilon^{\alpha\beta} K^o = 2K^o \times \text{Area} \tag{2.34}$$

Thus, an odd-elastic engine extracts work proportional to the odd elastic modulus and the area of the cycle in strain space. We can realize this in a microscopic model with odd elastic bonds whose spring force is $\mathbf{f} = -(k\hat{\boldsymbol{r}} + k^o\hat{\boldsymbol{\phi}})\delta r$. Consider the following cycle

1. Rotation: $\mathbf{r}_0 \to \mathbf{r}_0 - \delta\phi\, \hat{\boldsymbol{\phi}}$
2. Extension: $\mathbf{r}_0 - \delta\phi\, \hat{\boldsymbol{\phi}} \to \mathbf{r}_0 - \delta\phi\, \hat{\boldsymbol{\phi}} + \delta r\, \hat{\mathbf{r}}$
3. Rotation: $\mathbf{r}_0 - \delta\phi\, \hat{\boldsymbol{\phi}} + \delta r\, \hat{\mathbf{r}} \to \mathbf{r} + \delta r\, \hat{\boldsymbol{r}}$
4. Relaxation: $\mathbf{r} + \delta r\, \hat{\boldsymbol{r}} \to \mathbf{r}_0$

This is indeed a cycle as the spring returns to its original state. Step 1 performs no work as the spring is not extended and the net work by steps 2 and 4 is zero as the radial force is conservative. However, the work done in step 3 is $\mathbf{f} \cdot \delta\mathbf{r} = k^o \delta r \cdot \delta\phi(r_0 + \delta r)$. For infinitesimal

displacements this is approximately $k^o r_0 \, \delta r \delta\phi = k^o A$. The net work done over the cycle and is proportional to both k^o and the area enclosed by path. Ref. [165] shows this non-zero work enables self-sustained waves powered by odd-elastic energy cycles. This is not limited just to overdamped media containing abnormal springs – as we will see at the end of this chapter, microscopic activity can power deformation in biological systems as well.

2.5 Active gels

To conclude this chapter, we will show an example of how active matter theory can predict biological behavior. By modeling a cell membrane as a thin active gel, one can describe **cytokinesis**, or cell division. An **active gel** is a material consisting of a network of cross-linked filaments and molecular motors [152]. Beyond motor-induced deformation, active gel physics considers a second phenomenon that drives the material out of equilibrium. The filaments are polymers made of building blocks called monomers. The gel contains both polymers and free monomers not bound to a filament. The free monomers can attach to a filament, or polymerize, while the bound monomers can detach from a filament, or depolymerize. This spontaneous **polymerization** and **depolymerization** is a second energy transduction process which can create mechanical work from chemical energy.

In this section, we will first construct a theory for active gels following [152]. The two driving factors, motor-induced activity and (de-)polymerization result in equations with new active terms that break detailed balance. Next, we will apply this theory following [197], which demonstrates how active gel models predict cytokinesis.

2.5.1 Continuum theory: Active gel hydrodynamics

To construct a hydrodynamic theory, we start with conservation laws. In an active gel, momentum and number of monomers/motors are all conserved. We therefore require hydrodynamic equations for the dynamics of each corresponding conserved charge. Because

the gel is composed of filaments which can have a direction, we include an additional broken symmetry variable as with Toner-Tu and active-nematic hydrodynamics. Depending on whether there is polar or nematic order, this will be the polarization vector $\mathbf{p}$ or nematic director Q_{ij}, respectively.[11] The hydrodynamic equations come from a low-order gradient expansion restricted only to terms allowed by symmetry. The following sections will outline the equations for each collective variable and emphasize the new terms arising from activity.

Because an active gel has filaments and molecular motors, you might expect its hydrodynamics to resemble active nematics. You would be correct. The free monomers which can attach and disengage from the filaments will add a new wrinkle to the equations and make them more complicated, but capable of predicting biological behavior like cytokinesis.

Momentum

The momentum equation includes a stress σ_{ij} which for a passive system is

$$\sigma^p_{ij} = \eta \left(\nabla_i v_j + \nabla_j v_i - \frac{2}{3}(\nabla \cdot \mathbf{v})\, \delta_{ij} \right) + \bar{\eta}\, (\nabla \cdot \mathbf{v})\, \delta_{ij} + \sigma_{bs} \tag{2.35}$$

The first few terms are the classical linear stress tensor from Navier-Stokes, while σ_{bs} is a new component which depends on the density and orientation order parameter. Without time-reversal symmetry[12], the constitutive relation can have a new active stress term $\sigma^a_{ij} = \zeta Q_{ij} + \bar{\zeta}\delta_{ij}$ which is similar to that in active nematics. The coefficients ζ and $\bar{\zeta}$ are related to the motor and filament densities. They also scale with the difference $\Delta\mu$ between the chemical potential of the fuel (ATP) and that of its reaction products. Thus $\Delta\mu$ controls the activity in the gel and can drive the system out of equilibrium.

Crosslinkers bind filaments together which introduces a new elastic regime at intermediate

11. The nematic tensor $Q_{ij} = q(n_i n_j - \delta_{ij}/d)$ is a director representation that accounts for head-tail symmetry. q is the local ordering strength. At a defect, the director is undefined and $q = 0$.

12. No energy conservation

timescales. To account for this, we modify the constitutive relation with the Maxwell time τ_M – this is the timescale the crosslinkers remain bound.

$$\left(1+\tau_M\frac{D}{Dt}\right)(\sigma_{ij}-\sigma^a_{ij}-\sigma^{bs}_{ij})=\eta\left(\nabla_i v_j+\nabla_j v_i-\frac{2}{3}(\nabla\cdot\mathbf{v})\,\delta_{ij}\right)+\bar{\eta}\,(\nabla\cdot\mathbf{v})\,\delta_{ij} \quad (2.36)$$

When $\tau < \tau_M$, the derivative D/Dt dominates the left hand side. The material behaves like an elastic solid with shear modulus η/τ_M. The elastic regime breaks down at long time scales where the D/Dt term becomes negligible.

Monomer number

The monomer density ρ can be separated into polymerized (in filaments) monomers ρ_f and unbound monomers ρ_m. The total mass $\rho = \rho_f + \rho_m$ is conserved but the two sub-populations can fluctuate following

$$\partial_t\rho_f+\nabla\cdot\mathbf{J}^f_k=k_p\,\rho_m-k_d\,\rho_f \quad (2.37)$$

$$\partial_t\rho_m+\nabla\cdot\mathbf{J}^m_k=-k_p\,\rho_m+k_d\,\rho_f \quad (2.38)$$

The polymerization and depolymerization rates k_p, k_d do not obey detailed balance beause they represent different chemical processes. The monomer fluxes $\mathbf{J}^f$, $\mathbf{J}^m$ obey the constitutive relations

$$\mathbf{J}^{(f/m)}=\mathbf{J}^{(f/m)p}+\epsilon^{(f/m)}(\nabla\cdot\mathbf{n})\,\mathbf{n}+\epsilon'^{(f/m)}(\mathbf{n}\cdot\nabla)\,\mathbf{n} \quad (2.39)$$

Here (f/m) means the same equation form applies to both found and unbound monomers. We will apply this convention whenever equations are identical up to a choice of sub/superscript. The first right hand side term is the passive flux and is identical to that obtained in passive convection or diffusion problems. The remaining terms are due to activity. Because

there is no time-reversal symmetry, the constitutive relations can have terms with the symmetry of a vector – for example, the two active terms which come from filament distortions (splay/bend). For systems with polar order, the polarization $\mathbf{p}$ also has vector symmetry, allowing terms like $\lambda^{f/m}\,\mathbf{p}$.

The active terms allow monomers to flow spontaneously due to filament distortion. For polar active gels, local alignment can also create a monomer flux. The number of molecular motors is also conserved and they too can be separated into bound and unbound motors. Following the above procedure, one can find equations for molecular motor dynamics which have similar active terms.

Polarization

For gels with nematic order, the equations for Q_{ij} are identical to those for active nematics.[13] For polar gels, the dynamic equations instead become [152]

$$\frac{D\mathbf{p}}{Dt} = -\frac{1}{\gamma}\frac{\delta F}{\delta \mathbf{p}} + \bar{\nu}\,(\nabla\cdot\mathbf{v})\,\mathbf{p} + \nu\,\mathbf{p}\cdot\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T - \frac{2}{3}(\nabla\cdot\mathbf{v})\,\mathbf{I}\right) + \lambda_0\,\mathbf{p} + \lambda_1\,(\nabla\cdot\mathbf{p})\,\mathbf{p} + \lambda_2\,(\mathbf{p}\cdot\nabla)\,\mathbf{p} \quad (2.40)$$

The first term minimizes a free energy F which depends only on gradients of $\mathbf{p}$. The next two couple $\mathbf{p}$ to the flow field. Here γ is a rotational viscosity and ν, $\bar{\nu}$ are flow-alignment parameters. The $\lambda_0\,\mathbf{p}$ term is also present in passive gels. One can absorb it into the free energy as a Lagrange multiplier that enforces $\mathbf{p}\cdot\mathbf{p} = 1$.

The final two terms are unique to active polar gels. The λ_1 term creates active dynamics around splay distortions, whlie the λ_2 term produces dynamics around bend distortions. Both of these terms also appear in Toner-Tu theory [193], although we neglected them for simplicity in the above discussion. There, as is also the case here, they are allowed because

13. Earlier, we wrote these equations in terms of $\mathbf{n}$. A formulation using Q_{ij} is in Chapter 4.

symmetries and conservation laws do not explicitly prohibit them.

Summary

Active gel theory has much in common with the other active systems in this chapter. The polarization dynamics are reminiscent of Toner-Tu theory, while the momentum equation has the same active stress appearing in active nematics. Both active nematics and active gels are driven by molecular motors pulling on filaments.

On the other hand, the monomers binding and unbinding creates new terms that we have not seen before. The total monomer number is conserved, But the bound and free monomers are two distinct sub-groups whose interactions do not obey detailed balance. Beyond their chemical behavior, they are also driven by both the flow and orientation field.

2.5.2 Cytokinesis

The cell cytoskeleton is an active gel made up of crosslinked actin filaments and myosin molecular motors. Because of this, active gel theory can describe aspects of cell behavior and motion. Here, we follow [197] to demonstrate how active hydrodynamics can model **cytokinesis**, or cell division.

During cytokinesis, the cell membrane contracts over time until the cell separates into two. The membrane is attached to an actomyosin cortex and membrane dynamics are driven by the competition between cortical tension, which causes contraction, and cytoplasmic pressure, which prevents that contraction. To describe the former effect, we will model the actomyosin cortex using active gel theory. The latter effect comes from the cell interior, and we will assume it is a uniform hydrostatic pressure.

To model the cell membrane, we parameterize it with a curve in the $(\mathbf{e}_z, \mathbf{e}_r)$ plane. The full membrane is assumed axisymmetric about the $\mathbf{e}_z$ axis. Along this curve, we define a local coordinate system $(\mathbf{s}, \mathbf{n}, \mathbf{e}_\phi)$ where $\mathbf{s}, \mathbf{n}$ are unit vectors tangent and normal to the membrane

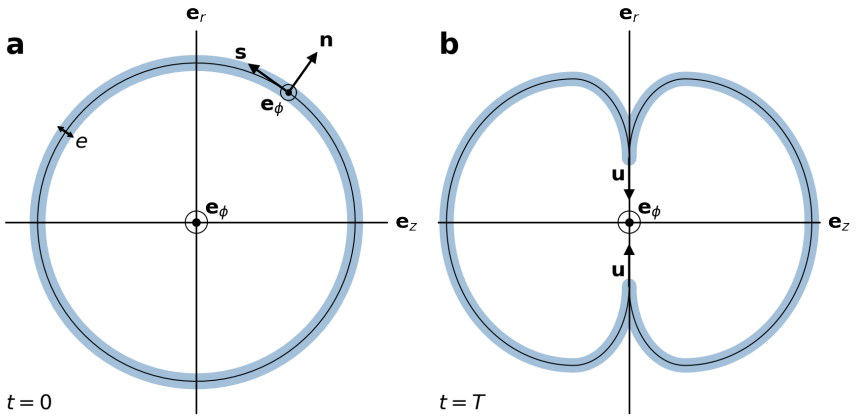

Figure 2.3: Application of active gel theory to cell division. (**a**) Setup of active gel model of cytokinesis described in 2.5.2. (**b**) Qualitative sketch of dynamics from solving (2.44)-(2.47). Original plot with numerical results is available at [197] Figure 1.

mid-line, respectively, and $\mathbf{e}_\phi$ is a unit vector in the azimuthal direction. These global and local coordinate systems are shown in Figure 2.3a.

The membrane can contract in the axial and azimuthal directions. The corresponding contractile tensions $N_s,\ N_\phi$ are given by integrating stress across the membrane and keeping to leading order in the small membrane width e

$$N_{s/\phi} = \int_{-e/2}^{e/2} d\xi\, \sigma_{ss/\phi\phi}(\xi) \approx e\, \sigma_{ss/\phi\phi} \tag{2.41}$$

The stress σ has passive and active components. With the strain rate $u_{ij} = (\nabla_i u_j + \nabla_j u_i)/2$, the passive stress is $\sigma^p_{ij} = 2\eta\, u_{ij} - p\, \delta_{ij}$ as in a classical Stokes flow.

Incompressibility requires $\nabla \cdot \mathbf{u} = 0$ and thus $u_{ss} + u_{\phi\phi} + u_{nn} = 0$. At the membrane surface $\xi = \pm e/2$, the normal stress σ_{nn} must be zero[14]. Because of this, the pressure p is $p \approx 2\eta\, u_{nn} = -2\eta(u_{ss} + u_{\phi\phi})$. The axial and azimuthal strain rates u_{ss}, $u_{\phi\phi}$ are, to leading

14. We assume it is constant in the transverse direction following the lubrication approximation

order, equal to their average values over the membrane width

$$d_{s/\phi} = \frac{1}{e}\int_{-e/2}^{e/2} d\xi\, u_{ss/\phi\phi}(\xi) \approx u_{ss/\phi\phi}(0) \tag{2.42}$$

Plugging these results into the expression for the passive membrane tensions $N_{s/\phi}$ gives

$$N_s^p = 2\eta e\,(2d_s + d_\phi) \qquad N_\phi^p = 2\eta e\,(d_s + 2d_\phi) \tag{2.43}$$

As before, the active stress is $\sigma_{ij}^a = \zeta\,\Delta\mu\,Q_{ij}$. Assuming that actin is parallel to the membrane requires $Q_{nn} = 0$. Further assuming that its in-plane polarization is isotropic gives $Q_{ss} = Q_{\phi\phi} = 1/2$ and the active membrane tensions are $N_s^a = N_\phi^a = \frac{e}{2}\zeta\,\Delta\mu$. Putting everything together, the total membrane tensions are

$$N_s = \frac{e}{2}\zeta\,\Delta\mu + 2\eta e\,(2d_s + d_\phi) \qquad N_\phi = \frac{e}{2}\zeta\,\Delta\mu + 2\eta e\,(d_s + 2d_\phi) \tag{2.44}$$

The activity coefficient ζ is related to the myosin motors. Experimental imaging of the protein RhoA, which regulates myosin activity, indcates that the motors are most active at the cell mid-line. We approximate this using a Gaussian profile centered at $s = 0$

$$\zeta(s,t) = \zeta_0 + I(t)\exp\left(-\frac{1}{2}\left[\frac{s}{w}\right]^2\right)\ \delta\zeta \tag{2.45}$$

Actin filament (de-) polymerization also contributes to membrane dynamics. Polymerization happens near the membrane surface where there are more free monomers. Depolymerization happens throughout the membrane. These effects control the dynamics of actin monomers a via

$$\frac{d\,(ae)}{dt} = -k_d\,a\,e + v_p a \tag{2.46}$$

Here k_d is a depolymerization rate and v_p is a polymerization velocity. The full membrane is parameterized as an axisymmetric curve with thickness e determined by the (de-) polymerization equation. The overall curve shape is determined by the competition between cell's cytoplasmic pressure P and the membrane tensions

$$\mathbf{div}(N_s, N_\phi) + P\,\mathbf{n} = 0 \tag{2.47}$$

The divergence operator depends on the membrane shape at a given time. Discretizing (2.44)-(2.47) yields a system of equations for the velocities which can be solved at each time step in order to determine the dynamics of the membrane shape. These equations are implemented numerically in [197] and evaluated for an initially spherical shape. Their results, approximately sketched in Figure 2.3, demonstrate that the active gel membrane model qualitatively reproduces the complex dynamics of cell division.

2.6 Summary

In active matter, microscopic components inject energy and induce emergent collective motion at the macroscale. This behavior appears in many biological contexts, including cell dynamics, bacterial swarms, flocking birds, and schooling fish. To model active materials using continuum theory, it is useful to lift constraints from a corresponding passive material. In the examples from this chapter, energy is no longer conserved. Lifting this constraint allows new terms in constitutive relations or hydrodynamic equations. In active gels, monomers are free to bind and unbind in a way that breaks detailed balance, which further modifies dynamical equations for density.

Active continuum theories are powerful, but they are also quite complicated. The equations are frequently coupled nonlinear partial differential equations. They are parameterized by many coefficients which are proxies for the microscopic components of the material.

However, it is difficult map between known microscopic properties, such as myosin motor concentration, and their corresponding hydrodynamic coefficients.

It is even less clear how to reliably apply active matter theory to biological systems. In our final example, we used active gel theory to build a simplified two-dimensional model that qualitatively described cell division. To achieve quantitative comparisons to real experimental data, we may have to extend these already complicated theories even further. Cells are controlled by many proteins which assemble and interact to produce large-scale mechanics. While a detailed microscopic model may account for all of these proteins, an effective continuum description might be able to get by using only a few. How to identify this **relevant biochemical information** and incorporate it into continuum theory via **mechano-chemical couplings** remains an open question.

CHAPTER 3

DEEP LEARNING

Machine learning (ML) refers to algorithms which learn from and adapt to data. A ML model trains on a dataset and learns behavior by updating its of parameters, also known as weights. By observing data, a model can optimize its weights to capture underlying patterns and then be applied to make predictions on unseen situations. In this chapter, we provide an introduction to **deep learning**, a machine learning framework centered around deep neural networks. Deep neural networks are a type of ML model with strong expressive power and a noted ability to generalize beyond their training. Unlike traditional approaches to statistical modeling, for instance linear regression, that are more or less curve fitting, neural networks have the capability to distill complex underlying physics and achieve robust predictions on high-dimensional data. We begin with some ML preliminaries and then provide an overview of neural networks, their building blocks, and several common network architectures. We conclude by summarizing considerations for selecting neural network models for biophysical problems. These insights were "human-learned" during the projects presented in this thesis.

3.1 Machine learning frameworks

Machine learning can be separated into three frameworks: supervised, unsupervised, and reinforcement learning. In **supervised learning**, a model examines a dataset containing labeled input-target pairs and learns to match each input to its target. A common example of supervised learning is linear regression, which attempts to find a slope α and intercept β such that $y_i = \alpha\, x_i + \beta$. Here, $\mathbf{x}$ is the vector of input data and $\mathbf{y}$ is the target data. In a linear regression problem, one can exactly solve for the weights α and β. More complex models, such as deep neural networks, are nonlinear and their parameters are typically found via gradient descent.

In **unsupervised learning**, the input data do not have specified targets. Instead, the network aims to optimize an objective which is carefully chosen so that the output identifies hidden patterns or a useful representation of the data. Unsupervised methods are particularly useful for simplifying complex data through dimensionality reduction, as well as modeling the probability distribution that a dataset is sampled from. One common unsupervised method is principal component analysis, which is a linear method of dimensionality reduction that finds the directions of maximal variance in the data.

Reinforcement learning is focused on the behavior of agents acting in an environment. The major goal is for intelligent agents to learn how to act in order to maximize their reward. A common application for reinforcement learning is games. In chess, for example, the players are agents, the board and pieces are the environment, and the goal is to learn a strategy which maximizes the chance of winning. By learning within this setup, AlphaZero achieved superhuman chess performance within 24 hours of training [174]. While reinforcement learning is an active and vibrant area of research, we do not focus on it in this dissertation.

3.2 Neural network building blocks

Neural networks are a nonlinear computing model inspired by the structure of the human brain which have proven effective at learning behavior from large complex datasets [108, 122]. They consist of several layers of interconnected neurons. The network computes an output by processing input data through each layer, additionally applying nonlinear activation functions which enhance expressivity. Each layer's input is combined with the trainable weights on each neuron connection to produce an output which in turn becomes the input for the next layer. This follows the equation

$$x_j^{\ell+1} = f(w_{ij}^{\ell} x_i^{\ell} + b_j^{\ell}) \tag{3.1}$$

Here ℓ is the layer index, w denotes the weights between neurons, b denotes a bias parameter to be trained on each neuron, and f is some fixed nonlinear function such as a sigmoid or tanh which enables neural networks to have strong expressive power. Here, we use the Einstein summation convention on the indices i and j, which represent the indices of neurons in each layer. During training, the weights and biases are adjusted to minimize the difference between the predicted output and the targeted output.

In a **fully-connected or dense neural network (DNN)** (Fig. 3.1A), each pair of neurons in adjacent layers are connected with a unique weight. When the number of neurons in a given layer is large, the number of weights to train can grow rapidly and become intractable. This can occur, for example, when the input layer is a spatially-extended field like the nematic director field. Architectural choices such as the use of convolutional or recurrent neural networks that employ shared weights can largely reduce the model complexity and make training tractable in such cases. The methods of weight sharing used by these architectures naturally account for symmetries present in many physical problems.

A **convolutional neural network (CNN)** is an architectural choice originally designed for image processing. They are commonly used to extract features in pictures that can contain millions of pixels. Nevertheless, a CNN is generally applicable to any input data that can be represented on a regular grid (of arbitrary dimension), such as the physical fields considered later in this dissertation. It takes advantage of translational invariance by employing patches of weights, known as filters, which are spatially scanned through the values on the grid (Fig. 3.1B). In this process, the same set of neural weights are shared by each input component, greatly reducing the number of weights to train. Sharing weights in this way also enables a CNN to be trained on a small-sized system and applied later to arbitrarily large systems. This not only avoids the issue of retraining the model for any different input sizes, which is typically required for dense neural network, but also allows us to conveniently generate large amounts of training data using small systems. While a single

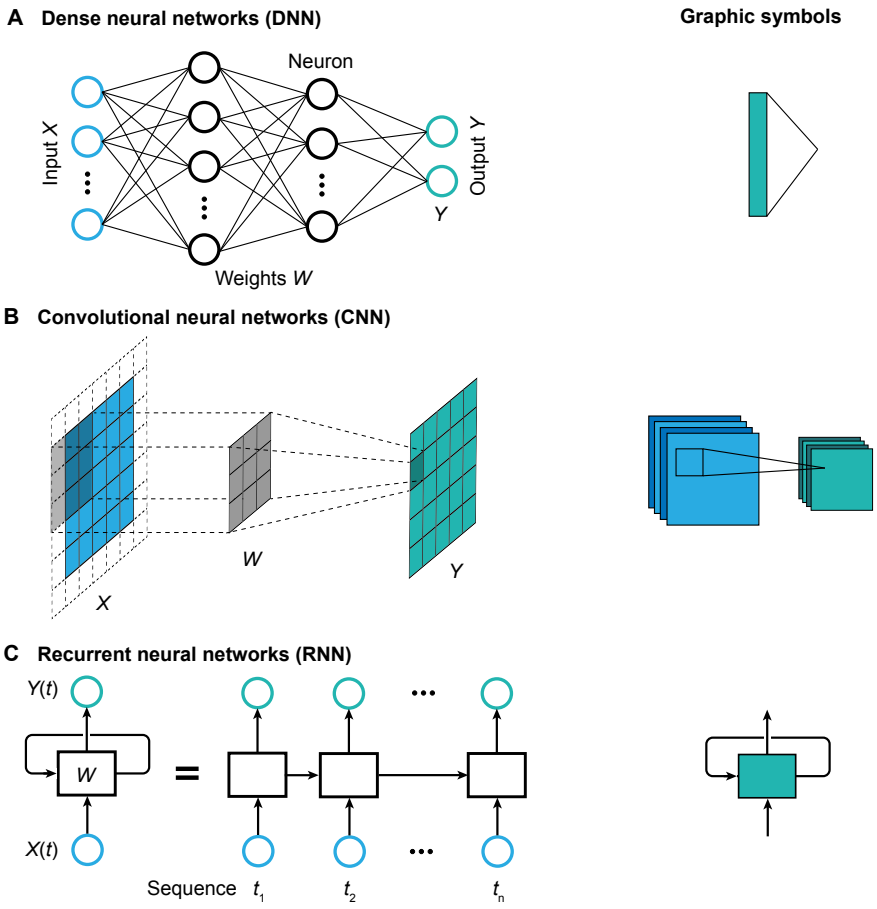

Figure 3.1: Overview of common neural network building blocks. (A) Dense neural networks (DNN). When an input vector X (blue) passes through the network, it is processed by multiple layers of neurons (black) to generate the final predictive output Y. Neurons (circle) at any two adjacent layers are fully connected. Each connection line contains a weight W, which will be optimized to capture the relation $Y(X)$ during the training process. (B) Convolutional neural networks (CNN). CNNs are commonly used for image processing. They employ neural patches (grey) to scan through the input. This not only takes the spatial connectivity of the input data into account but also largely reduces the model complexity by using shared weights. (C) Recurrent neural networks (RNN). RNNs are often applied to sequential data, i.e., $X(t)$ and $Y(t)$ with $t = t_1, t_2, \ldots, t_n$. Unlike a DNN that passes input directly to output, RNN also contains a recurrent link (loopy arrow), which can be expanded as another neural network passing information from the initial time t_1 all the way to the current time t_n. This allows us to capture memory effects along the sequence. These three common neural networks provide the building blocks for more advanced neural network architectures. On the right-hand side of each panel, we provided the corresponding graphic symbols that we used in the maintext.

convolutional layer can only collect local information up to the size of each filter, a neural network can obtain information from larger regions by using a stack of convolutional layers connected in sequence. By doing so, each subsequent layer of filters gets a progressively larger field of view (or receptive field) on the input.

A **recurrent neural network (RNN)** is an architectural choice designed to process sequential data. It exploits time invariance by scanning through an input sequence with the same neural cell (Fig. 3.1C). Unlike the convolutional filters that are only connected with adjacent layers, a recurrent neural cell is also connected with itself through a so-called recurrent connection (Fig. 3.1C). This allows a RNN to preserve useful information collected from the past time points and capture memory effects that could exist in a system. The prediction made at any point in time is based on both the current input and any information it has kept from the past. Because the same set of weights is used at each point in time, a RNN can be used to study complex dynamical problems even with long-term memory effects without requiring an excessive number of weights.

3.3 Neural network architectures

The **architecture** of a neural network refers to the number of layers, the amount of neurons in each layer, and the ways the neurons themselves are connected. Diverse architectures have been proposed and proven successful for various tasks across computer vision and natural language processing. Several of these have been adapted to data analysis problems in both biology and physics, ranging from biomedical image segmentation to rare event identification in high-energy physics. In addition to architecture, neural networks can be chosen to model different features of the data for different purposes. For example, given pairs of inputs x and outputs y, one may try to approximate a function $f(x) \approx y$ with a **deterministic** model, $p(y|x)$ with a **discriminative** model, or $p(x, y)$ with a **generative** model [133]. We focus on deep learning for building biophysical models from image data. In the following, we will

review important concepts related to neural network architectures. We will then discuss considerations for selecting architectures for biophysical modeling.

3.3.1 Convolutional neural networks

A **convolutional neural network (CNN)** is an architecture type originally designed for use in computer vision. Instead of learning unique weights for each pixel in an image, a CNN exploits translational invariance and learns a series of **filters** which are applied equivalently to all pixels to produce **feature maps**. This weight sharing greatly reduces the number of trainable parameters in each layer, allowing CNNs to be built deeper (more layers) and wider (more features). Adding more CNN layers means the NN can built more complex features which correlate larger regions of the image. Below, we review several common types of CNN.

An **autoencoder** is a tool used for compression and feature extraction. It contains two modules: an encoder and a decoder. The encoder uses CNN layers and downsampling operations to build a compressed, or **latent**, representation of the input. The decoder attempts to reconstruct the input from the latent representation. During training, the autoencoder learns to build a high-level latent features essential for reconstructing the image. In physics, autoencoder compression can be thought of as coarse-graining a system, and is used for dimensionality reduction or identifying useful degrees of freedom. They are also useful for distilling essential information from noisy data.

CNN weight sharing enables deeper networks, but deep CNNs can be more difficult to train and perform worse than their shallow counterparts. A **ResNet** alleviates this problem by using residual, or skip, connections [80]. These pass inputs across layers and combine them with the layer outputs. This improves the training performance of deep CNNs, enabling networks with hundreds of layers.

A **UNet** is a CNN tool that combines autoencoder and ResNet designs and has proven remarkably successful across diverse domains. It includes an encoder which compresses the

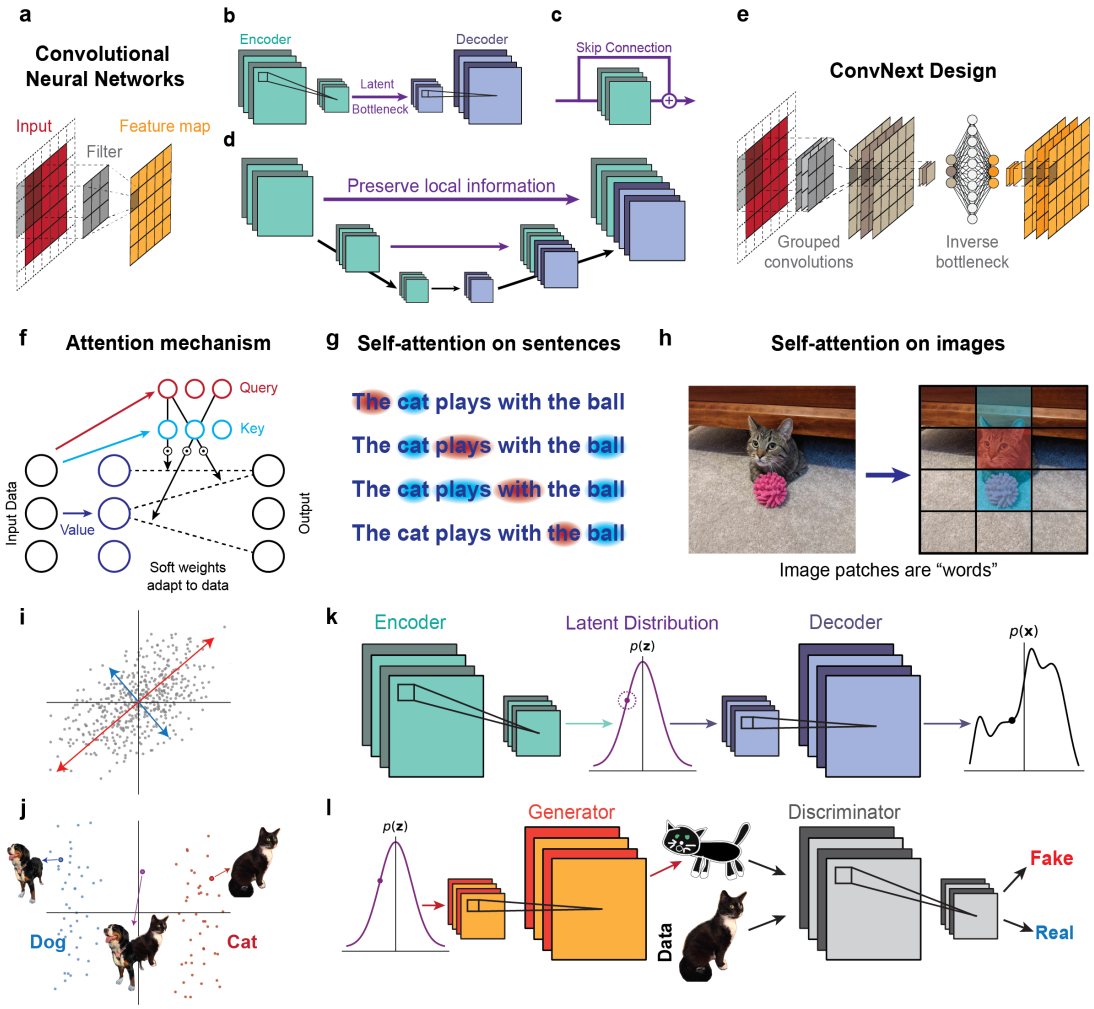

Figure 3.2: Overview of common neural network architectures. (**a**) CNN learns filters applied over an image. (**b**) Autoencoder architecture. (**c**) Skip connection in ResNet. (**d**) UNet architecture combines autoencoder and residual connections. (**e**) ConvNext modifies CNN to improve performance. (**f**) Transformers use self-attention mechanism computes soft weights (**g**) Schematic of self-attention on sentences. Red represents current word (query) and blue highlights the context (aligned keys) for each word. (**h**) Vision transformers apply self-attention after separating images into patches. Red highlights current patch (query) and blue highlights regions interacting with that (aligned keys) (**i**) PCA can model Gaussian-distributed data using learned principal axes. (**j**) Sampling non-gaussian data is more complicated. Generating a sample from the center of a bi-modal distrubtion may produce nonsense, represented here as a cat-dog. (**k**) Variational autoencoders model complex data distributions by learning a nonlinear mapping to a well-conditioned latent space. (**l**) Generative adversarial networks learn to generate realistic data by training a generator to fool a discriminator network.

input as well as skip connections which send information from the encoder to the decoder. This provides several advantages useful for biological and physical applications. The encoder can still perform compression and noise reduction, but in the process it may also remove important small-scale features. The skip connections mitigate this problem, allowing the UNet to preserve relevant fine-grained information and local structure. This is particularly useful for learning deterministic rules in data such as protein distributions, as the UNet can learn causal relationships about protein structures at different scales.

3.3.2 Attention-based networks

In traditional neural networks, the neuron weights are fixed during evaluation. Regardless of the input or the features extracted from the input, each layer always performs the same operation on its data. **Attention** is a mechanism that allows a neural network to act differently depending on context by allowing the network weights to change during evaluation. During training, an attention-based network learns how to determine context and the correct behavior for each context [24]. Explicitly, an attention mechanism encodes data into (**key**, **value**) pairs. It also computes a **query**, which describes the context to predict in. The alignment between a query and key determines the connection weight for the corresponding value. This alignment might be different depending on the input data, and thus the neuron connections can change during evaluation.

A **transformer** is a type of neural network which uses attention mechanisms for sequence prediction. It was originally developed for natural language processing applications [199]. This provides a more concrete interpretation to the attention mechanism. Consider translating a sentence – to process each word, a **query** might contain information about word itself while the **keys** might encode the other words in the sentence. If a key aligns closely to the query, then it provides important context for that word in the sentence – its **value** then plays a bigger role in choosing how to translate that word. Attention mechanisms are not

restricted to language models. In biology, one could view amino acids in proteins as analogs of words in sentences. Indeed, AlphaFold used attention-based building blocks in order to predict protein structures from amino acid sequences [90].

Transformers are effective at processing sequences, but not all data are sequences. **Vision transformers** adapt attention mechanisms to image analysis [101]. An image is made up of pixels just as a sentence is made up of words. However, images contain thousands to millions of pixels and computing attention between each pair of pixels (millions to billions) is simply not tractable. A vision transformer instead separates an image into patches – in the sentence analogy, the pixels (letters) make up patches (words) which make up the image (sentence). The fact that these images are not sequential is not an issue. A transformer learns the correct context for each word by examining the whole sentence, rather than the sequence of words leading up to it. Similarly, a vision transformer learns the correct context for each patch by examining the full image and learning the proper query-key alignment between patches. Vision transformers and their offshoots have achieved state-of-the-art performance at many computer vision tasks [115].

3.3.3 ConvNext: Improved CNN with transformer design principles

The performance of vision transformers comes with a price: these models often require more parameters, higher training costs, and larger datasets in comparison to CNNs. CNNs are simpler to implement and their core operation, the convolution which slides a filter over the image, is quite efficient. In the **ConvNext** paper [116], the authors reevaluated CNNs and carefully considered their fundamental design principles in the context of vision transformers. To do this, they applied iterative upgrades to a ResNet architecture, ranging from layer- (new activations, larger filters) to module- (grouped convolution, inverse-bottleneck design) to network-level changes (patch separation, new training techniques). The result was a family of models which were as accurate as transformers, but remained pure CNNs. ConvNext

designs can be incorporated into existing CNN architectures, such as those discussed in the previous sections. Later, we will discuss the advantages this provides from a biophysical modeling perspective.

3.3.4 Discriminative and generative models

Neural networks are often trained on data pairs (x, y), where x is an input, and y may be an output or label for the item x. In most real-world settings, there is no purely deterministic relationship between x and y, but instead a distribution of possible y's for every x. This distribution can either arise naturally due to stochasticity in the data, or artificially due to human error in generating labels y.

Generally, **discriminative** ML models aim to learn the conditional distribution $p(y|x)$, or a function thereof. In common regression tasks, one assumes a deterministic function $f(x) \approx y$. However, if the relationship is not truly deterministic one instead learns the best deterministic approximation, which is given by $f(x) = \mathbb{E}(y|x) = \sum_y y\, p(y|x)$. In addition to learning deterministic functions, certain classes of probabilistic neural networks (among others, **mixture density models** [16]) explicitly attempt to learn a probability distribution $p(y|x)$. This can be useful if noise is expected to be large relative to the deterministic part, or if the data is drawn from a highly non-Gaussian distribution.

A **generative model** aims to explicitly model the data distribution so that it can draw new samples that resemble the real data [20]. A simple yet illustrative example is principal component analysis (PCA), a common tool for dimensional reduction [122]. PCA aims to represent each data point using a linear combination of principal components, which are the "axes" along which the data is distributed. One can generate new samples using PCA by changing the set of component coefficients into a combination that has never been seen before. However, generating samples in this way is only valid for Gaussian data. For more complicated distributions, choosing a particular combination of component coefficients may

lead to unrealistic samples as they lie far outside the data's distribution. The central issue is that PCA does not provide insight into what choices of coefficients lead to realistic samples, and is the reason why it is not a true generative model. Proper generative models not only create new samples, but generally also provide a measure of probable such a sample is to occur. Below, we review several common neural network architectures that realize deep generative models.

We presented autoencoders as a tool for data compression and dimensional reduction, they make poor generative models. In an autoencoder, the relationship between the latent variables and the data has no constraints. Two similar inputs might have dramatically different latent variables, and two nearby latent variables might decode to completely different outputs. A **variational autoencoder (VAE)** adapts autoencoders for generative modeling by making the latent variables well-behaved. To do this, it optimizes two objectives: a reconstruction term common to a standard autoencoder, and a **distribution term** that enforces the latent distribution matches a simple form (usually a normal distribution) [97, 98]. While this second term (a Kullback-Leibler divergence) is computationally expensive, a VAE exploits a **reparameterization trick** to train efficiently. In short, this trick replaces latent variables with samples from their learned distributions, and this randomness enables efficient estimation of the second objective. Once trained, a VAE can draw samples from this well-behaved latent space and decode them into new data points that resemble the training dataset.

A perfect generative model produces outputs indistinguishable from real data. This means it must not only be able to recreate its training data, but should produce believable new samples. This is difficult. Neural networks require mathematical objectives – the second objective here is simply the question "real or fake?" A **generative adversarial network (GAN)** solves this issue by turning training into a game between a **generator** network and a **discriminator** network [71]. The discriminator learns to distinguish between real data

and the generator's samples. The generator learns to fool the discriminator. Both networks are trained simultaneously and thus push each other to improve. The discriminator can learn complex criteria that are hard to represent mathematically, and the generator can learn to satisfy these criteria and produce strikingly realistic images.

While GANs are a powerful tool for generating realistic data, their training setup of an adversarial game makes them difficult to train. VAEs are simpler to implement, but ultimately learns a limited approximation of the distribution using a neural network. Other types of generative models can learn more exact representations of the data distribution. **Normalizing flows** learn a set of invertible transformations which map samples from a gaussian distribution to the data samples [99, 158]. While these models allow for exact representation and sampling from the data distribution, their constraints (difficulty training large mixtures, restriction to invertible transformations) can limit their effectiveness at generating data from more complex distributions [20]. **Diffusion models** take a different approach which is inspired by nonequilibrium thermodynamics. They first destroy samples by iteratively adding noise, and then learn a model for the reverse process which reconstructs data from the noise [83, 176]. Diffusion models have led to state-of-the-art tools for image synthesis [208]. However, their generation process can be computationally-expensive due to the many denoising operations as well as the requirement that the latent space have the same dimensionality as the generated data [20]. From a physical perspective, they do not perform coarse-graining.

3.4 Summary

We have presented a small sample of the many successful neural network architectures. With all of these available choices, it can be difficult to determine which is best for a given application. In many cases, there is no "right" answer. Here, we outline general considerations for choosing network architectures for biophysics applications.

Data and training costs: The commonly-used datasets in computer vision and natural language processing – the home territory of many models presented above – are large, well-curated, and standardized. On the other hand, biophysical datasets may come from a single laboratory and are typically smaller and noisier. While transformers and their variants achieve state-of-the-art performance on language and image analysis tasks, they often require large datasets, many parameters, and long training times to do so [116]. CNNs on the other hand are straightforward to implement and train, which may make them a better choice for analyzing biological images. With the development of ConvNext, which allows pure CNNs to compete with vision transformers, choosing CNNs no longer comes at the cost of dramatically lower accuracy.

Interpretability: A goal of scientific research is to understand *why* systems behave as they do. Deep neural networks are great at predicting behavior, but their black-box nature often makes it difficult to determine what they have learned. Networks designed for unsupervised learning are useful in this regard as they produce representations of the data that may be used for further interpretation. Autoencoders learn to compress data and examining their compressed representations can shed light on the crucial ingredients of a system. Generative models learn a latent space from the data, and by sampling new data from this space, one can learn about the different "axes" that describe a system. We note that some modern generative models, such as GANs, can be difficult to train reliably [20]. VAEs strike a reasonable balance between ease of implementation/training and interpretability.

Physical intuition: In data-driven biophysical modeling, we aim to use machine learning to uncover the physical rules governing biological behavior. To choose a network architecture for biophysics, it is useful to consider the types of rules one aims to learn. CNNs exploit translational invariance, a common feature of physical models. CNN filters can implement local physics operations. For example, a Laplace filter which identifies local maxima is a finite-difference implementation of a diffusive term (i.e. $D\nabla^2\rho$) present in many hydro-

dynamic models [122]. Networks with a bottleneck structure (autoencoders and VAEs) can be viewed as coarse-graining the system, and their compressed latent representations can be interpreted as a system's relevant collective variables [31, 100]. Such coarse-graining may be impossible when behavior is driven by processes across scales, as is often the case in biology. Here, it may be necessary to use architectures which lack a bottleneck, such as flat CNNs, or ones which preserve information at small and intermediate scales such as UNets.

Modeling frameworks such as classical hydrodynamics and continuum mechanics have deterministic rules, and are thus well-suited to architectures such as CNNs or autoencoders that similarly make deterministic predictions. However, biological systems may exhibit fluctuations or stochasticity due to unobserved processes at smaller scales. Networks which use a variational approach such as VAE or other generative models may be better suited to learning this behavior. How to map models such as VAE or diffusion models to frameworks such as fluctuating hydrodynamics remains an open question.

CHAPTER 4

LEARNING ACTIVE-NEMATIC HYDRODYNAMICS

Hydrodynamic theories effectively describe many-body systems out of equilibrium in terms of a few macroscopic parameters. However, such parameters are difficult to determine from microscopic information. Seldom is this challenge more apparent than in active matter, where the hydrodynamic parameters are in fact fields that encode the distribution of energy-injecting microscopic components. Here, we use active nematics to demonstrate that neural networks can map out the spatio-temporal variation of multiple hydrodynamic parameters and forecast the chaotic dynamics of these systems. We analyze biofilament/molecular-motor experiments with microtubule/kinesin and actin/myosin complexes as computer vision problems. Our algorithms can determine how activity and elastic moduli change as a function of space and time, as well as ATP or motor concentration. The only input needed is the orientation of the biofilaments and not the coupled velocity field which is harder to access in experiments. We can also forecast the evolution of these chaotic many-body systems solely from image-sequences of their past using a combination of autoencoders and recurrent neural networks with residual architecture. In realistic experimental set-ups for which the initial conditions are not perfectly known, our physics-inspired machine learning algorithms can surpass deterministic simulations. Our study paves the way for artificial-intelligence characterization and control of coupled chaotic fields in diverse physical and biological systems, even in the absence of knowledge of the underlying dynamics.[1]

1. This chapter contains material previously published in [38], titled **Machine learning active-nematic hydrodynamics**. Re-use is permitted according to the copyright agreement used by *PNAS*. Text contains references to supplementary figures available online.

4.1 Introduction

Machine learning holds great promise as a tool capable of transforming quantitative modelling in the physical sciences [31, 122]. It takes data generated from simulations or collected from experiments and uses powerful nonlinear fitting functions to find the characteristic features behind the data. It can be used as either a continuous regression tool to extract physical principles or a discrete classifier to identify states of matter. Among the wide variety of machine learning techniques, neural networks [108, 166] have attracted much attention due to their strong predictive power and ability to establish complex models from common building blocks (see SI Appendix for a primer). Notable developments within condensed matter physics have led to machine learning algorithms capable of recognizing structural signatures of the glass transition [8, 168] or distinguishing phases of matter [32, 198]. They have also unveiled intriguing connections between deep learning and renormalization group methods [100, 123]. However, the use of machine learning as a tool for the experimental characterization and discovery of material properties is still in its infancy [40, 153]. Active nematics [46, 49, 53, 79, 95, 105, 119, 163, 188] provide an ideal material platform for machine-learning methods. While sufficiently well-characterized to be a reliable benchmark, their chaotic dynamics are hard to predict and rich in unexplored phenomena of relevance to both material science and biology [17, 46, 48, 50, 69, 121, 128, 164, 207, 216].

Unlike simple fluids, nematic liquid crystals are orientationally ordered media described by a director field $\mathbf{n}(\mathbf{r})$, which tracks the average orientation of their microscopic constituents (e.g., biofilaments or elongated molecules), in addition to the local velocity field $\mathbf{v}(\mathbf{r})$ [46, 119]. Representative images of the director field are shown in Fig. 4.1*A* and *B*. In equilibrium nematics, the filaments tend to align and gradients of $\mathbf{n}(\mathbf{r})$ are penalized by the Frank free-energy, which in three dimensions reads

$$F = \frac{K}{2} \int \left[(\nabla \cdot \mathbf{n})^2 + (\mathbf{n} \cdot (\nabla \times \mathbf{n}))^2 + (\mathbf{n} \times (\nabla \times \mathbf{n}))^2 \right] \mathrm{d}\mathbf{r} \tag{4.1}$$

where, for simplicity, all the elastic constants are set equal to K.

The introduction of microscopic energy sources into these orientationally ordered fluids generates out-of-equilibrium systems called active nematics [46, 119]. A common example is provided by cytoskeleton filaments, with molecular motors that generate active forces promoting inter-filament sliding. The resulting active stress, σ^a_{ij}, can be macroscopically described by

$$\sigma^a_{ij} = \alpha\, n_i n_j, \tag{4.2}$$

where $\alpha(\mathbf{r}, t)$ is an *a priori* unknown activity field related to the concentration of molecular motors or other energy sources, which is typically approximated as constant [1, 119].

4.2 Active-nematic hydrodynamics

The dynamical equations for active nematics involve a coupling between the director field $\mathbf{n}(\mathbf{r}, t)$ and the velocity field $\mathbf{v}(\mathbf{r}, t)$ [1, 46, 73, 119]. In particular, the director field evolves according to the equation

$$(\partial_t + \mathbf{v}\cdot\nabla)n_i = \lambda_{ijk}\partial_j v_k - \frac{1}{\gamma}\left[\frac{\delta F}{\delta n_i} - \left(\frac{\delta F}{\delta n_j}n_j\right)n_i\right], \tag{4.3}$$

where γ is the rotational viscosity. The tensor λ_{ijk} is given by

$$\lambda_{ijk} \equiv \frac{\lambda+1}{2}n_j\delta_{ik} + \frac{\lambda-1}{2}n_k\delta_{ij} - \lambda n_i n_j n_k \tag{4.4}$$

where λ is a dimensionless flow-alignment parameter and δ_{ij} is the Kronecker delta.

The velocity field evolves according to the equation

$$\rho_0(\partial_t + \mathbf{v}\cdot\nabla)v_k = -\partial_k P + \eta\nabla^2 v_k + \partial_j\left(\lambda_{ijk}\frac{\delta F}{\delta n_i}\right) + \partial_j\sigma^a_{jk} \tag{4.5}$$

where P is the hydrostatic pressure, η is the isotropic fluid viscosity, and σ^a is the active stress defined in (4.2). The final three terms on the right-hand side of (4.5) correspond to the viscous, elastic, and active stresses, respectively. The coupled equations of motion Eqs.(4.3–4.5) can also be cast in terms of a symmetric and traceless order parameter $Q_{ij} = S(n_i n_j - \delta_{ij}/3)$ with a scalar field S that quantifies the orientational-order strength (see Methods).

Determining how hydrodynamic parameters, such as α and K, vary in space and time as well as a function of ATP or motor concentrations is challenging [88, 105, 110, 111]. Even direct measurements [7, 22, 64, 180] that rely on controlled flow experiments are difficult to devise if the underlying flows are chaotic. This is precisely what happens in active nematics, where the energy injected by molecular motors at the microscale cascades to macroscopic scales, leading to chaotic flows mediated by the proliferation of topological defects [1, 119]. The mechanisms behind this process, loosely called active nematic turbulence, are not fully understood [1, 5, 17, 66, 67, 163, 170, 171, 188, 204]. Nonetheless, it is clear that most active nematic responses depend on the competition between active stresses, that promote director or velocity gradients, and viscoelastic stresses that resist them. As a consequence of this interplay, experimental measurements often access only non-trivial *combinations* of hydrodynamic parameters, e.g. α/K or ratios of elastic constants [105, 110]. Furthermore, the derivation of these parameters from realistic microscopic models is often prohibitively difficult [22]. This prompts us to seek approaches that bypass coarse-graining and extract hydrodynamic parameters like $\alpha(\mathbf{r}, t)$ directly from experiments.

In this paper, we first design a neural network to extract hydrodynamic parameters directly from experimental or simulated movies. This requires prior knowledge of the underlying physics. For systems whose governing equations are unknown, we introduce another neural network capable of forecasting the time evolution of a nonequilibrium system solely based upon its past. Detailed workflows for training and testing these neural networks are

provided in SI Appendix Fig. S2.

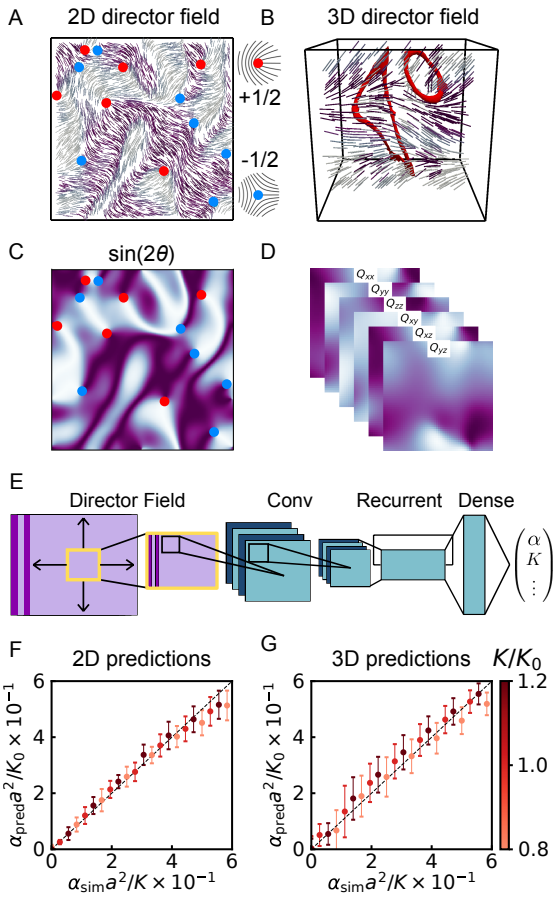

Figure 4.1: **Machine learned hydrodynamic parameters in Lattice-Boltzmann simulations.** (*A, B*) Nematic director fields in two (*A*) and three (*B*) dimensions. $^{+1}/_{2}$ and $^{-1}/_{2}$ defects in 2D are marked as red and blue dots, respectively. Disclination loops are indicated in red. (*C, D*) Continuous representations of the director field used by the network. In 2D, the network can use $\sin 2\theta$ where θ is the angle of the director field. In 3D, the network uses the tensor $Q_{ij} = n_i n_j - 1/3$. Color indicates the magnitude of these continuous representations. (*E*) Schematic of neural network architecture. The full input images are divided into patches, which are then fed into a set of convolutional filters, a LSTM recurrent layer, and a fully connected dense layer. The outputs are averaged into a final estimate for hydrodynamic parameters. (*F, G*) Predictive accuracy of rescaled dimensionless activity in simulation data in 2D and 3D at different values of K. Networks were trained at $K = K_0$ (see Methods).

4.3 Extracting hydrodynamic parameters as fields

To make progress, we recast the task of estimating the spatio-temporal variations of multiple hydrodynamic parameters as a computer-vision problem that can be effectively addressed by artificial intelligence. We begin by generating a library of director fields (Fig. 4.1*A* and *B*) for a wide range of activity in two and three dimensions using Lattice-Boltzmann simulations based on the continuum equations [105, 120] (see Methods). Using the simulated library, we train neural networks on continuous representations of $\mathbf{n}$ that account for the nematic symmetry $\mathbf{n} = -\mathbf{n}$. For instance, in two dimensions we use $\sin(2\theta)$ with θ denoting the tilt angle of $\mathbf{n}$, see Fig. 4.1*C* and *D*. The neural network architecture, shown schematically in Fig. 4.1*E*, contains (i) a single convolutional layer used for image processing, (ii) a recurrent layer that captures the system dynamics, and (iii) a dense layer that identifies the hydrodynamic parameters (see SI Appendix primer on neural networks and SI Appendix Fig. S1). We train these neural networks on simulation data for which we can obtain the exact values of the prescribed hydrodynamic parameters corresponding to each director field configuration. This allows us to conduct supervised learning, which would not be possible using experimental data alone. Once trained, these neural networks can be used to obtain hydrodynamic parameters in simulations as well as experiments (SI Appendix Fig. S2*A*).

We first apply this scheme to estimate a single parameter: the rescaled dimensionless activity $\alpha/K \times a^2$, where a denotes the pixel or voxel size for the director field image. Comparison of the machine learning predictions for the activity with the known values of α reveals good agreement for both two-dimensional (2D) and three-dimensional (3D) active nematics (Fig. 4.1*F* and *G*). Although these networks are trained on data generated at a single value of $K = K_0$, their accuracy persists for samples where K differs from K_0. We also stress that our machine learning model is robust and its predictive performance does not depend on the choice of K_0 (SI Appendix Fig. S3).

Hydrodynamic theories suggest how α can be estimated from the characteristic length

$\ell_d \propto \sqrt{K/\alpha}$ obtained by balancing the right hand sides of Eqs. (1) and (2) [82]. In 2D, ℓ_d can be interpreted (and experimentally extracted) as the average spacing between point disclinations, topological defects with index $^{+1}/_{2}$ and $^{-1}/_{2}$ shown in Fig. 4.1*A* as red and blue dots, respectively [67, 119]. This procedure cannot be carried out in the low-activity regime, where disclinations are not always present in the field of view. Furthermore, it does not extend to 3D samples where the dominant excitations are charge-neutral disclination loops [49] (Fig. 4.1*B*) whose activity-dependence is unclear. Extracting and measuring topological defects with a neural network would require multiple convolutional layers to achieve a sufficiently large receptive field, i.e. the total field of view for the last neural layer. Our networks, which contain just a single convolutional layer, simply exploit local spatial fluctuations of the director field.

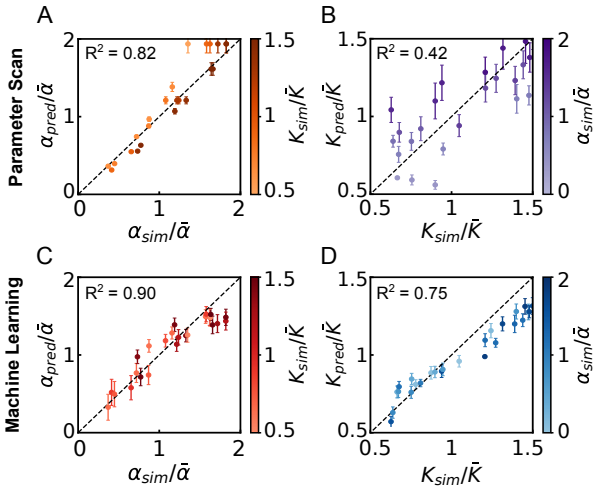

Figure 4.2: **Comparison of multiparameter estimation using neural networks and a high-throughput parameter scan.** (*A, B*) Simultaneous estimation of α and K using a high-throughput parameter scan. (*C, D*) Multiparameter estimation using our neural network. The network estimator outperforms the parameter scan approach for the predictions of both α and K. Here $\bar{\alpha}$ and $\bar{K}$ are the mean values of α and K from the training dataset. We quantify the performance for each parameter using the R^2 of the linear fit between the predictions and the ground truth (dashed line).

While capable of achieving high accuracy even in 3D or at low α, neural networks designed for single-parameter estimation still predict only combinations of parameters such

as $\ell_d \propto \sqrt{K/\alpha}$. To decouple α from K, one would need to resort to additional measurements [111] or *ad hoc* assumptions like the independence between K and α which is not always experimentally valid [110]. However, our neural networks are also capable of extracting multiple parameters without the need to devise a set of experiments that disentangle the parameters' interdependence. In Fig. 4.2, we demonstrate the performance of a machine learning model that uses the same architecture described above but is trained to predict α and K simultaneously.

Using a simulated dataset, we evaluate the performance of our machine learning model by comparing it to high-throughput scans of parameter space inspired by [111]. While the exact approach implemented in [111] uses all evolving fields, our machine learning model does not rely on knowledge of the velocity field to which the director is coupled. Because complete information about the system state is inaccessible in many experimental systems, we compare our method against parameter scans which use only the nematic director field. These scans aim to find the set (α, K) best fitting the observed nematic correlation length and correlation time (see SI Appendix Text and SI Appendix Fig. S4). For each parameter, we quantify the model performance using the R^2 value of the linear fit between the predicted parameter and the ground truth. We find that the machine learning model results in a higher R^2 than the parameter-scan approach (Fig. 4.2), indicating a more accurate prediction of the ground truth for each parameter.

So far we have tested our machine learning algorithms on numerical data. We now turn to experiments and begin by applying our multiparameter estimation trained on simulations to data obtained from microtubule-kinesin experiments [42, 163] (see Methods). It is known that in this system the rescaled activity α/K increases with ATP concentration [88, 110]. Here, we use machine learning to measure α and K independently, probing first how they vary with the ATP concentration, c. Inspection of Fig. 4.3*A* shows that the spatio-temporally averaged activity, α, predicted by our machine learning algorithms

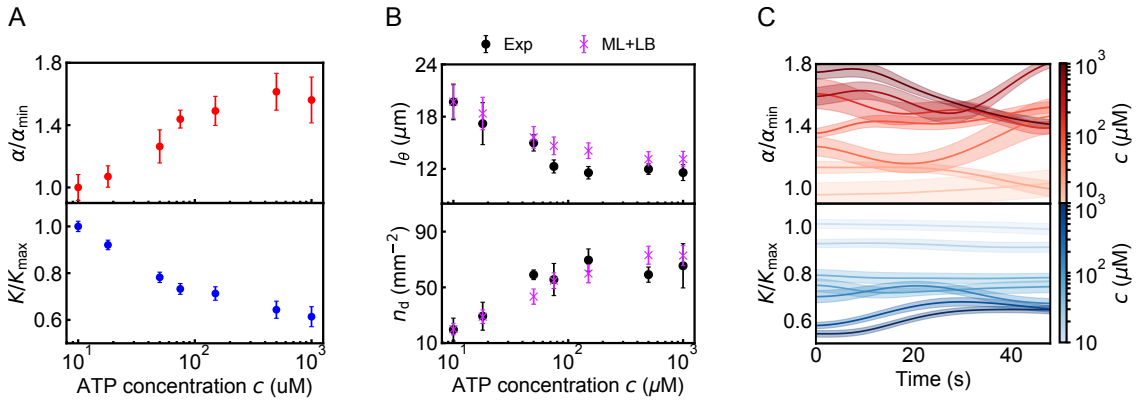

Figure 4.3: **Multi-parameter estimation and dynamics in microtubule-kinesin experiments.** (A) Dependence of spatio-temporally averaged activity and elastic modulus on ATP concentration. Here, $\alpha_{\min}$, $K_{\max}$ are the time-averaged predicted activity and elastic modulus at the lowest level of ATP concentration $c_{\min} = 10$ μM. (B) Comparison of director field correlation length l_θ and defect spacing $n_{\rm d}$ in experiments (Exp) and ML-informed lattice Boltzmann simulations (ML+LB). (C) Simultaneous prediction of activity and elastic modulus over time at different levels of ATP concentration. The shaded regions represent the standard-error of spatio-temporal fluctuations in the machine learning predictions. ATP concentration c is indicated by the color bar.

increases with c while the elastic modulus, K, decreases.[2] Similar results obtained from experiments on 3D microtubule-kinesin systems [49] and 2D actin-myosin systems [105, 216] are shown in SI Appendix Figs. S5–S6.

Before expanding on the capabilities of our parameter-estimation networks, we highlight their salient features. Our algorithms do not perform curve-fitting by (i) identifying observables for which the underlying theory is solvable and (ii) parameterizing them in terms of the sought-after coefficients. Instead, neural networks are trained on data obtained using whatever conditions are experimentally available without choosing in advance which reduced data representations to use (e.g. correlation functions or collective variables). Training neural networks differs from building lookup tables (or other *discrete* representations of the data), which are impractical for fields like $\mathbf{n}(\mathbf{r}, t)$ when the number of possible pixel (voxel)

2. This drop in K indicates the material resists deformation less at higher ATP concentrations and motor activities. The effects of molecular motors, which are responsible for crosslinking the nematic biofilaments, on material properties of active liquid crystals has been proposed theoretically [2] and recently investigated in experiments [156].

configurations vastly exceeds the number of available data points. When applied to previously unseen data, neural networks can still produce accurate predictions because they learn the *smooth* high-dimensional manifolds that map all possible realizations of the fields onto the corresponding hydrodynamic parameters.

Equipped with the ability to determine multiple hydrodynamic parameters from experimental data, we now proceed to put the hydrodynamic theory itself to the test. Here, we compare microtubule-kinesin experiments at different ATP concentrations to the evolution of Lattice-Boltzmann simulations calibrated using parameters machine-learned from the same experiments. As the chaotic nature of active nematics makes exact director field comparisons unreliable over long times, we instead resort to properties of the dynamical steady state. Using the spatial correlation function $C_s(\mathbf{r})$ for the director field (see Methods), we define the correlation length ℓ_θ such that $C_s(\ell_\theta) = 1/2$. We find that both the average correlation length and the defect density n_d calculated from machine learning informed Lattice Boltzmann simulations match experiments for a wide range of ATP concentrations (Fig. 4.3*B*). This procedure allows us to provide a partial experimental validation of the hydrodynamic theory with constant parameters. Note, however, that these hydrodynamic parameters are often taken as constant because measuring them locally is difficult.

Our machine learning methods provide a rare glimpse into the spatio-temporal variations of these hydrodynamic parameters. For example, we can extract from the microtubule-kinesin experiments the activity field, $\alpha(\mathbf{r}, t)$, whose coarse-grained dynamics stems from heterogeneities in the motor distribution. We start by plotting in Fig. 4.3*C* examples of time series for the sample-averaged $\alpha(t)$ and $K(t)$. The mean values are plotted as solid lines and their uncertainties are marked with shaded regions over a wide range of ATP concentrations (denoted by color bars). We compare the parameters extracted from experiments with those from simulations, where constant parameters are prescribed so that the fluctuations in the extracted parameters arise solely from the uncertainty of the machine learning predictions.

We find that the fluctuations of the machine-learned α are markedly more pronounced in experiments than in simulations, while the variation of the predicted K is comparable in both cases (SI Appendix Fig. S7). Our analysis also shows that the time variations in α and K are significantly larger than their spatial fluctuations. The machine-learned evidence discussed above suggests that a non-linear fluctuating hydrodynamic theory may better explain our experimental observations. Heuristically, the strong disruption of fiber alignment at large activity can trigger motor detachment-reattachment events causing the time modulation of α inferred by our algorithms.

Our machine learning models can be also applied to situations in which activity is engineered to *deliberately* vary in both time and space. We test this activity control scenario first in Lattice-Boltzmann simulations, where we prescribe spatio-temporal patterns of $\alpha(\mathbf{r}, t)$ [216]. Remarkably, neural networks trained on the data with constant activity can still accurately estimate a time-varying activity coefficient as shown in Fig. 4.4A and B where linear and sinusoidal activity profiles are probed. Since small director-field patches are sufficient to generate reliable predictions, we can generate a spatial activity map of $\alpha(\mathbf{r}, t)$ by applying our neural networks locally to each patch composing an image. By doing this, we are able to discern prescribed spatial activity patterns in Lattice-Boltzmann simulations, as demonstrated in Fig. 4.4C where activity is non-zero only in the central square.

We further test the capability of our machine learning models in extracting parameter fields using actin-myosin experiments [105]. In this system, one can alter the speed of some specialized molecular motors via selective exposure to light [132]. This phenomenon, informally called gear-shifting (see Methods), allows for precise *spatio–temporal* control of active stresses [216]. Inspection of Fig. 4.4D and SI Appendix Movie S1 reveals that our machine learning models can successfully identify the marked increase in activity that occurs as light is turned on (indicated by the dashed line in Fig. 4.4D). Furthermore, our approach can identify the activity changes that occur in selectively illuminated *spatial* domains in these

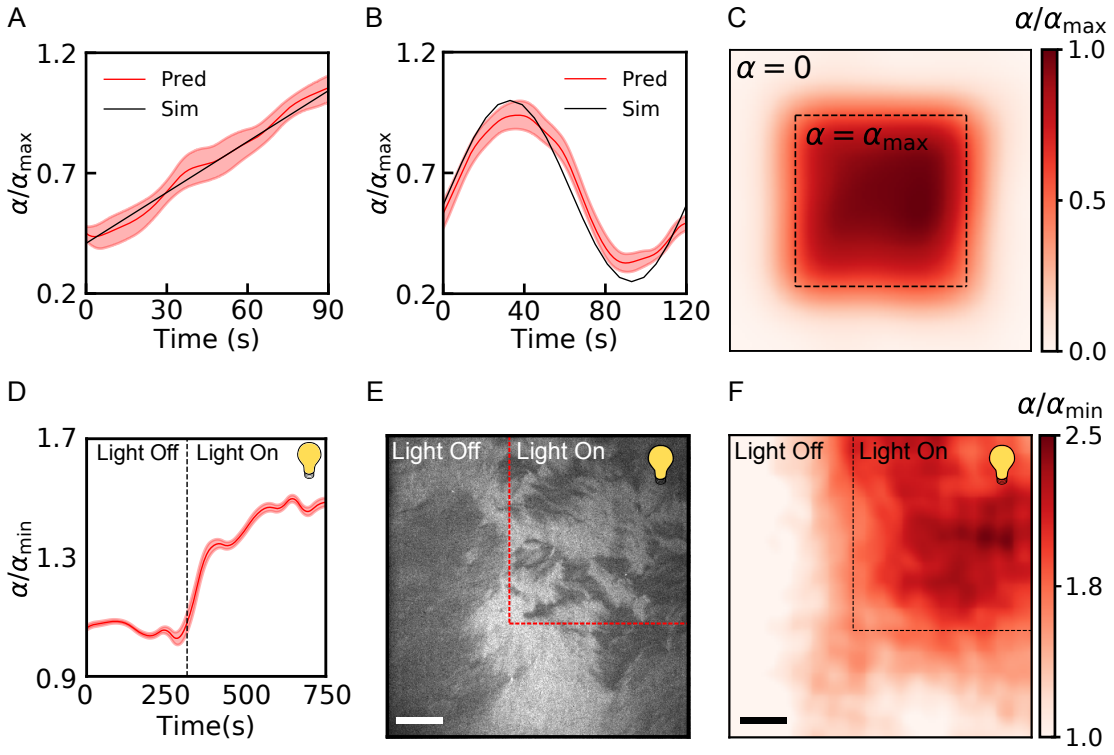

Figure 4.4: **Machine-learned activity field, $\alpha(\mathbf{r}, t)$, in simulations and actin-myosin experiments.** (A, B) ML-predicted activity on lattice Boltzmann simulations with spatially uniform activity prescribed to vary linearly (A) and sinusoidally (B) in time. (C) ML-predicted activity on simulations where the central square (dashed line) is activated. (D) ML-predicted activity vs. time on actin-myosin experiments where myosin motors are controlled through light-activated gearshifting. The dashed line indicates when light is switched on. (E, F) Direct image (E) and ML-predicted spatial activity profile (F) of a selectively illuminated actin nematic with light-activated gearshifting motors. For E, F the experimental data is the dataset reported in Fig. 1 of Zhang *et al.* [216]. Data for D are from the current study, following the approach used in [216]. Scale bars, 20 μm.

systems, see Fig. 4.4E, F and SI Appendix Movie S1. We can also extract the elastic modulus field $K(\mathbf{r}, t)$ in these experiments, but we find that K does not change significantly when light is applied to the gearshifting motors (see SI Appendix Fig. S8). The performance of our machine learning model in identifying spatio-temporally varying activity demonstrates its potential for (i) the control of engineered active materials and (ii) the inference of biochemical processes that take place at the microscopic level (such as the different experimental configurations summarized in SI Appendix Table S1).

4.4 Forecasting time evolution

We now ask: can neural networks forecast the evolution of chaotic many-body systems solely from image-sequences of their past? A time-honored approach to quantitative modelling relies on writing down equations and then solving them, analytically or via simulations, to make predictions. In what follows, we use the word *machine-learning model* to denote a very different approach [144]. Instead of solving the equations, we train neural networks on existing data and then ask them to forecast the future behavior of the chaotic system. A feature that distinguishes our forecasting neural networks from the ones used for parameter estimation, is that the former can be trained directly on experimental data while the latter rely on an underlying model. Our physics-inspired machine learning approach to forecasting the dynamics of active nematics consists in iterating the following two steps. First, we perform next-frame predictions using a neural network that does not know anything about the physics of the system. Second, we reduce any noise generated in the previous step by applying to each frame a physics-inspired sharpening algorithm. This sharpening filter harnesses the known propensity of the fibers to align (i.e., it minimizes the elastic energy in (4.1)) while being agnostic about the active forces driving the non-equilibrium dynamics (see Methods).

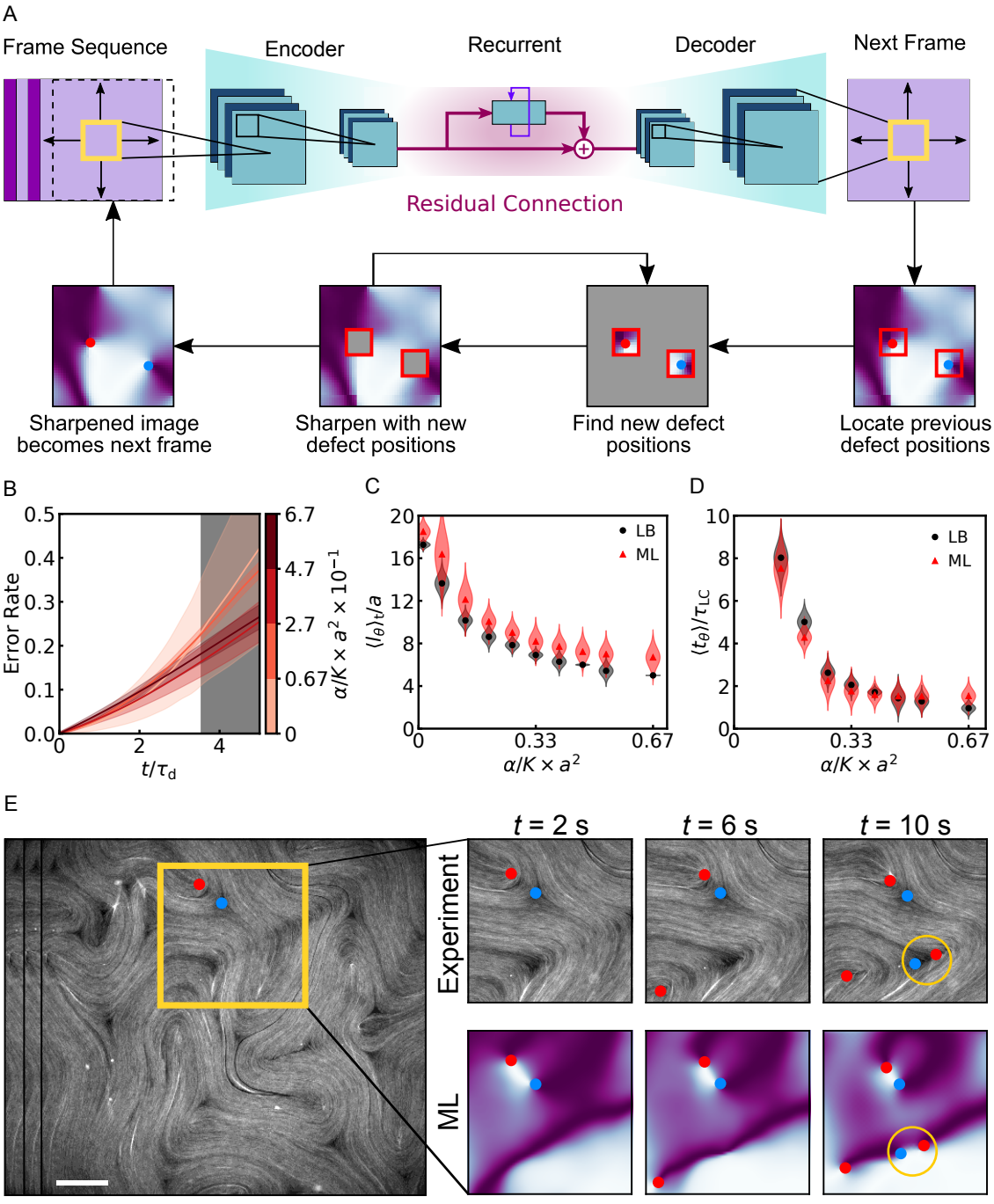

Figure 4.5: **Neural networks as surrogate models of time evolution.**

Figure 4.5: (cont.) (*A*) Schematic for forecasting active-nematic evolution. An encoder compresses images into a sequence of feature vectors. A residual recurrent block predicts the next feature vector, which is translated by a decoder into an output image. The output is sharpened using relaxational dynamics to update defect positions, which are then used to update the director field. (*B*) Pixel-wise error rate $1-\langle|\mathbf{n}_{\mathrm{ML}}\cdot\mathbf{n}_{\mathrm{LB}}|\rangle$ of the ML model versus time for different activity ranges. The gray region is beyond the simulation Lyapunov time (see Methods), which is measured in units of $\tau_{\mathrm{d}}=\eta/\alpha$, the characteristic defect lifetime. (*C-D*) Comparison of average correlation lengths (*C*) and times (*D*) in ML and LB simulations. τ_{LC} is a viscoelastic timescale $\tau_{\mathrm{LC}}=\gamma a^2/K$ [68]. (*E*) A defect nucleation event observed in experiment and predicted by the ML model trained on microtubule-kinesin experimental data. ML predicts $\sin(2\theta)$, where θ is the director field angle. $^{+1}/_{2}$ and $^{-1}/_{2}$ defects are marked as red and blue dots, respectively. Scale bar 100 μm.

Our forecasting neural networks are a modification of the autoencoder architecture, a popular tool in the computer vision community (see SI Appendix primer). A traditional autoencoder learns to compress an image to a feature vector which is then used to reconstruct the image. In our network, we insert a recurrent layer in between the encoder and the decoder to learn the system dynamics (see Methods). Crucial for ensuring high performance is the following algorithmic trick: a residual architecture [80] is used in the recurrent layer to capture the *difference* between frames, rather than the images themselves. Denoting the output of the recurrent layer as $\mathrm{RNN}(X_0, X_1, \cdots, X_t)$, the predicted next frame can be written as:

$$X_{t+1} = X_t + \mathrm{RNN}(X_0, X_1, \cdots, X_t) \tag{4.6}$$

Such a residual recurrent network resembles the discrete form of a general differential equation:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = f(X_0, X_1, \cdots, X_t). \tag{4.7}$$

As illustrated in Fig. 4.5*A*, the network encodes a time series of director-field images into a sequence of feature vectors. Next, it uses them to predict the future state of the system, and finally decodes this state back into a director-field image (Fig. 4.5*A*). For large systems,

the director field is divided into small overlapping domains. Machine learning predictions are made within each domain and then stitched into a final prediction of the next director field configuration. Although stitching could introduce artificial defects and image blurriness in the overlapping area between adjacent domains, these errors can be automatically corrected by the sharpening step (see Methods). Once trained on either simulated or experimental image sequences, the machine learning model can then forecast the defect dynamics in active nematics (SI Appendix Fig. S2B).

We first examine the performance of our forecasting neural networks on simulation data. Given a particular sequence of nematic configurations, our algorithm can reliably learn the spatio-temporal evolution of the director field including singular events such as defect annihilation and nucleation (see SI Appendix Movie S2 or selected examples in SI Appendix Fig. S9).

To systematically evaluate the accuracy of our machine learning predictions, we first compare the time-evolved director fields generated by machine learning and Lattice-Boltzmann simulations pixel by pixel. Such pixel-wise comparison is only meaningful within the Lyapunov time, the characteristic timescale after which a non-linear dynamical system becomes chaotic. Inspection of Fig. 4.5B shows that the pixel-wise error rate of the predicted director field $1 - \langle|\mathbf{n}_{ML} \cdot \mathbf{n}_{LB}|\rangle$ remains small within the Lyapunov time (see Methods). The Lyapunov time is the exponential diverging time for the evolution of a small difference in $\mathbf{n}(\mathbf{r}, t)$ introduced in the initial condition, which we find is equal to $t_\lambda \sim 3.6\tau_\mathrm{d}$ (τ_d denotes the average defect lifetime). Beyond the Lyapunov time (shaded region in Fig. 4.5B), even the Lattice-Boltzmann simulations are unreliable at the pixel-wise level due to numerical precision.

Our forecasting neural network predicts the evolution of the director field without knowing the velocity field $\mathbf{v}(\mathbf{r}, t)$ or ordering magnitude S. We compare the error rates of our algorithm to those of a Lattice Boltzmann simulation, where similarly no initial velocity

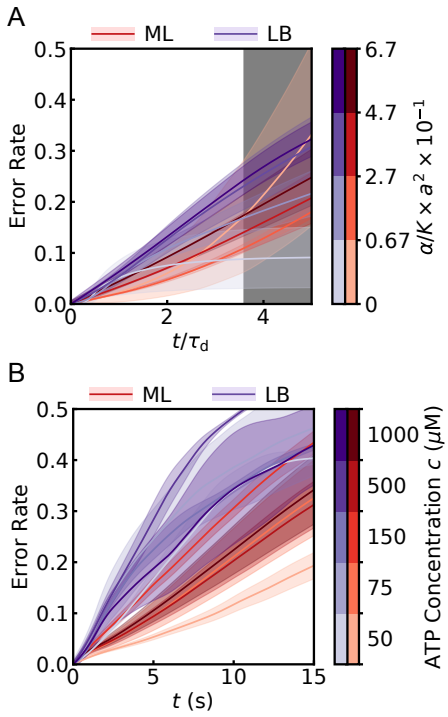

Figure 4.6: **Performance comparison between physics-inspired machine learning model and Lattice-Boltzmann simulations with parameters extracted using the multi-parameter estimation network.** To make a fair comparison as well as mimic the true experimental constraints, both approaches only take director fields as input, with no prior knowledge of the velocity field. We quantify their performance by measuring the pixel-wise error rate $(1 - \langle |\mathbf{n}_{\text{prediction}} \cdot \mathbf{n}_{\text{ground truth}}| \rangle$. (A) Error rates for predicting the simulated nemato-hydrodynamics at different levels of activity. Red curves show results for the machine learning model, while purple curves show results for velocity-uninformed Lattice-Boltzmann predictions. The gray area shows regions beyond the Lyapunov time for the Lattice Boltzmann simulations (see Methods). (B) Error rates for predicting the evolution of microtubule-kinesin experiments at different ATP concentrations. Here we emphasize that unlike Lattice Boltzmann simulations, the machine learning model does not implement any physical theory. Nevertheless, its performance matches or exceeds Lattice Boltzmann simulations.

or ordering magnitude information is provided (Fig. 4.6A). Our machine learning model achieves lower error rates up to the Lyapunov time, except at the lowest levels of activity when the system is nearly passive.

In order to evaluate the predictive accuracy of our machine learning methods for even longer times, we turn to properties of the dynamical steady state such as the director field correlation length ℓ_θ defined before and the correlation time t_θ defined by setting the time correlation function $C_t(t_\theta) = 1/2$ (see Methods). Previous numerical studies have shown that the quantity ℓ_θ is proportional to $\sqrt{K/\alpha}$ [82]. When comparing the predictions of our machine learning model against Lattice-Boltzmann simulations, we find that machine learning correctly captures the activity dependence of the characteristic length ℓ_θ, see Fig. 4.5C (corresponding results for the mean defect spacing ℓ_d are shown in SI Appendix Fig. S10). We stress that while ℓ_θ at steady state is plotted in both Fig. 4.3B and Fig. 4.5C, the former is generated from Lattice-Boltzmann simulations with machine-learned parameters whereas the latter is generated solely using our time-evolution neural network. Furthermore, our networks also reproduce the same activity dependence for t_θ as the Lattice-Boltzmann simulations (Fig. 4.5D), suggesting that they have learned to reproduce the correct dynamics expected at each level of activity.

After training a neural network *exclusively* on experimental data, we can successfully forecast the time evolution of the nematic director field including singular events such as topological defect nucleations or annihilations, see Fig. 4.5E for an example. Here we choose microtubule-kinesin experiments as a paradigmatic example, as they often exhibit clear defect dynamics. In SI Appendix Movie S3 we show the corresponding experimental video next to the one generated using machine learning. Inspection of these movies show no discernible differences in the defect dynamics between experiments and machine learning predictions. Similar agreement is obtained when our time-evolution neural networks are trained on Lattice-Boltzmann simulation data (SI Appendix Movie S3).

As an alternative approach, one can also forecast the evolution of active nematics by first extracting the parameters (α, K) using our parameter-estimation network and then plugging them into Lattice–Boltzmann simulations to model the corresponding hydrodynamics. We stress that this approach requires prior knowledge of the underlying physics. Notably, we find that our time-evolution network outperforms the physics-informed simulations in the context of practical applications to experimental data, where complete knowledge of the initial state is inaccessible (Fig. 4.6B). To evaluate the long-term validity of our predictions beyond specific realizations, we systematically check (as we did in Fig. 4.3B and Fig. 4.5C) that the steady-state values of ℓ_θ and n_d extracted from the machine-learned nematic director are in good agreement with experiments over a wide range of ATP concentrations (SI Appendix Fig. S11).

The successful performance of our time-evolution networks relies on the combination of autoencoders and recurrent networks that adopt a residual architecture. The convolutional autoencoders compress input images into feature vectors, which represent the underlying physics using fewer variables. This dimensional reduction not only diminishes the burden of the machine-learning model, but it also enables training on limited experimental datasets without overfitting. We stress that our recurrent networks predict the correct long-term dynamics at steady state only when supplemented with a residual architecture, see SI Appendix Figure S12. More generally, we expect the use of a residual architecture capturing the difference between frames to be crucial for all dynamical systems that are naturally governed by differential equations (see Eqs. (4.6, 4.7)).

4.5 Conclusion

The machine learning framework proposed here can estimate hydrodynamic parameters using only movies of the director field without requiring knowledge of the velocity field, even if the two are coupled. Because the framework primarily exploits local spatial fluctuations

of the nematic director, one can map out the spatio-temporal variation of hydrodynamic parameters and promote them from constants to fields. This provides a generic way to quantitatively connect experimental observations and theories.

When trained only on experimental data, neural networks are shown to be capable of forecasting the future, without theoretical knowledge of the underlying dynamics. This scenario is particularly intriguing for experimental systems that, unlike active nematics, lack a quantitative description of their coarse grained dynamics. In addition, the simplicity of our time-evolution neural network system makes it suitable for implementation of artificial-intelligence informed control of such systems. For example, induced spatio-temporal variations of active parameters combined with machine learning techniques could enable efficient control of complex flows and pattern formation in synthetic [69, 73, 139, 147, 207, 216, 219] and biological systems [50, 61, 121, 128, 164]. Beyond active and soft matter, our neural-network models could be employed in other contexts where coupled chaotic fields naturally occur, such as turbulent flows or magneto-hydrodynamics [25, 51, 113].

CHAPTER 5

LEARNING ADHERENT CELL MECHANICS

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 protein, such as zyxin, are sufficient to predict forces and generalize to unseen biological regimes. Using this observation, we develop two approaches – one constrained by physics, 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. [1]

5.1 Introduction

The structure and dynamics of living cells are controlled by the physical properties of the cytoskeleton [19, 146]. The cytoskeleton itself, however, is the product of complex biochemical circuits which regulate its dynamics and spatial organization [58, 186]. 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 which arise in such models as well as the equations they obey [149].

1. This chapter is adapted from work in press at *Cell* titled **Machine learning interpretable models of cell mechanics from protein images**, which has been released as a preprint at ArXiv:2303.00176. Text contains references to supplementary figures which are available online.

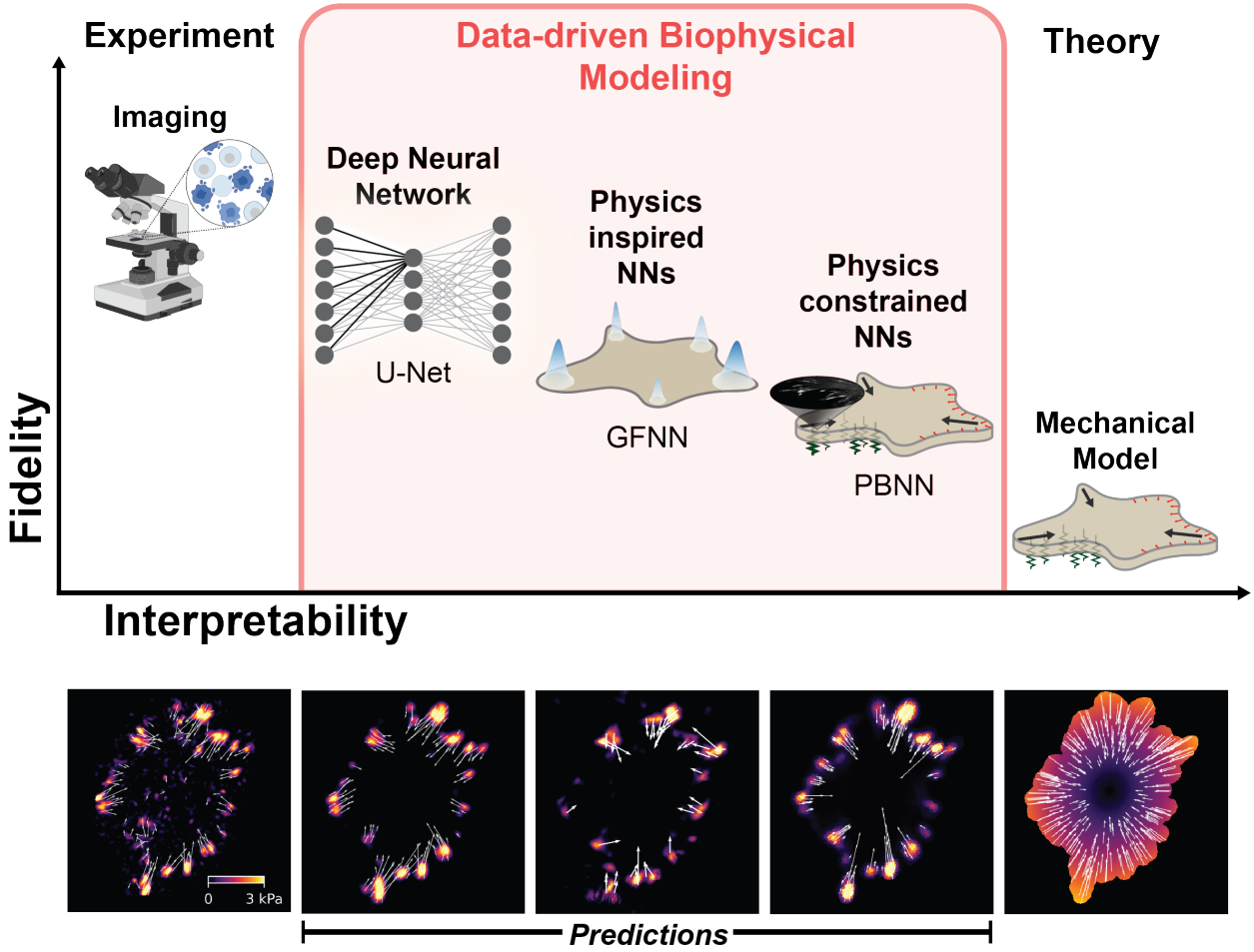

Figure 5.1: **Learning models of cell mechanics from protein images**. Data-driven biophysical modeling techniques occupy a spectrum of model complexity and interpretability. Deep neural networks are not limited to making black-box predictions, but can serve to test physical hypotheses and build models of various degrees of refinement.

Cells, however, are decidedly non-classical, relying instead on distributed enzymatic activity and non-equilibrium mechanochemical processes across a hierarchy of scales [9, 118]. 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 [152, 159].

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 [31, 37, 175, 214]. These tools have proven very successful in structural biology for predicting protein structures directly from gene sequences [90, 112]. 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 [87, 130]. Forces generated in the actin cytoskeleton are transmitted via transmembrane focal adhesions (FAs) to the extracellular matrix [29, 106, 169] where they can be measured directly with techniques like Traction Force Microscopy (TFM) [85, 162]. 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 [30, 78, 124, 135, 136, 138, 177, 203]. 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.

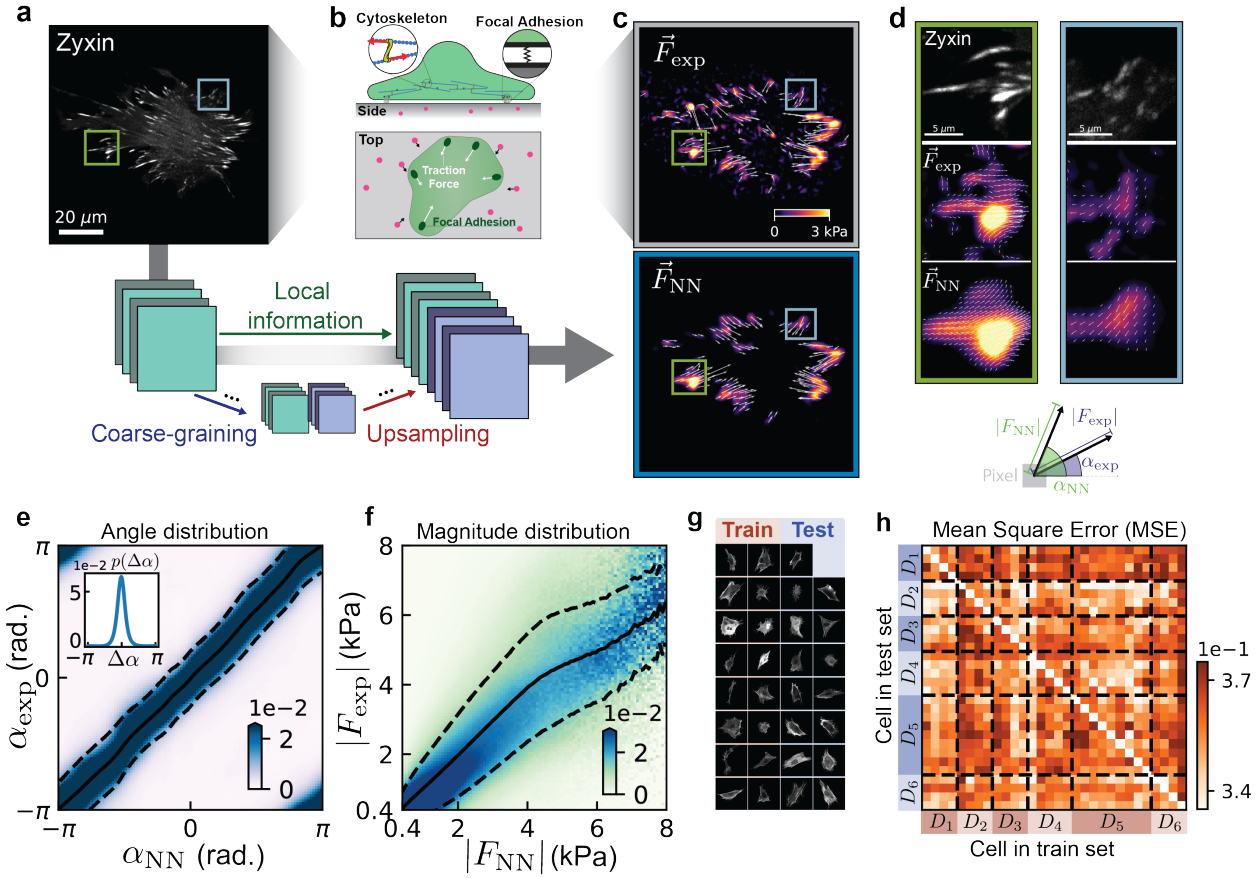

Figure 5.2: **Neural networks predict forces from images of a single protein.**

Figure 5.2: (cont.) (**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 Methods). (**c**) (Top) Forces ($\vec{F}_{\text{exp}}$) recovered from experimental measurements of substrate deformations via TFM. (Bottom) U-Nets predict traction forces ($\vec{F}_{\text{NN}}$) 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**) At each pixel, we measure $(\alpha_{\text{exp}}, \alpha_{\text{NN}}, |F_{\text{exp}}|, |F_{\text{NN}}|)$, which we bin to calculate the conditional angular distribution $p(\alpha_{\text{exp}}|\alpha_{\text{NN}})$ (**e**) and (**f**) the conditional magnitude distribution $p(|F_{\text{exp}}||F_{\text{NN}}|)$. 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 $|F_{\text{exp}}| \approx 4$ kPa, which corresponds to 99.9% of pixels. Inset of (**e**) shows the probability distribution of angular error $\Delta\alpha = \alpha_{\text{NN}} - \alpha_{\text{exp}}$. (**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-square error 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 which use cell i for training and cell j for testing.

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 discover that a single focal adhesion 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 the networks can generalize to previously unseen experimental and biological perturbations. To understand this generalizability, we probe the neural network to identify features which inform its predictions and further guide the formulation of two complementary mathematical models. First, we introduce a physics-constrained ML approach which augments existing mechanical cellular

models [52, 124, 136, 179]: 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 which constructs relevant fields and distills effective equations which predict cellular traction stresses. Despite incorporating varying degrees of model complexity and prior knowledge, all of 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.

5.2 Neural networks 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 ([84]; Fig. 5.2a) with their corresponding traction forces as directly measured by TFM ($\vec{F}_{\text{exp}}$; Fig. 5.2b-c). In total our library contained images obtained from 31 separate time-lapses of cells expressing zyxin and their associated traction force fields (see 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 inwards towards the cell body (Fig. 5.2c,d). For our neural network, we chose a U-Net architectural backbone which 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 ([161]; Fig. 5.2a). While U-Nets have proven successful at solving the TFM inverse problem [102, 206], here we push them to link biochemistry and mechanics. We augment this backbone with ConvNext blocks to improve accuracy and

training efficiency [116] (see Methods and SI for architecture details). We trained the U-Net to directly predict traction forces using the library of paired zyxin images as inputs ($\vec{F}_{\mathrm{NN}}$; Fig. 5.2c,d). The library was split into training and test sets containing 16 and 15 cells, respectively (Fig. 5.2g). 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 (Fig. 5.2c,d). The network predictions of traction forces agreed generally with experimental measurements in both location and magnitude (Fig. 5.2c), with some smoothing occurring at the micron scale (Fig. 5.2d).

To evaluate the U-Net accuracy, we compared the experimentally measured traction force directions (α_{exp}) and magnitudes ($|F_{\mathrm{exp}}|$) to those predicted by the U-Net for all the cells in the test set (α_{NN}, $|F_{\mathrm{NN}}|$). Fig. 5.2e,f shows the conditional distributions $p(\alpha_{\mathrm{exp}}|\alpha_{\mathrm{NN}})$ and $p(|F_{\mathrm{exp}}||F_{\mathrm{NN}}|)$ (see 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 $\sim$4 kPa, which represents approximately 99.9% of pixels in the test dataset (Fig. 5.2e,f; SI Fig. 6). 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 square error (MSE) of the force predictions (Fig. 5.2h; SI Fig. 7). 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 (SI Fig. 7), denoted by D_i in Fig. 5.2h. 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 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 focal adhesion protein, such

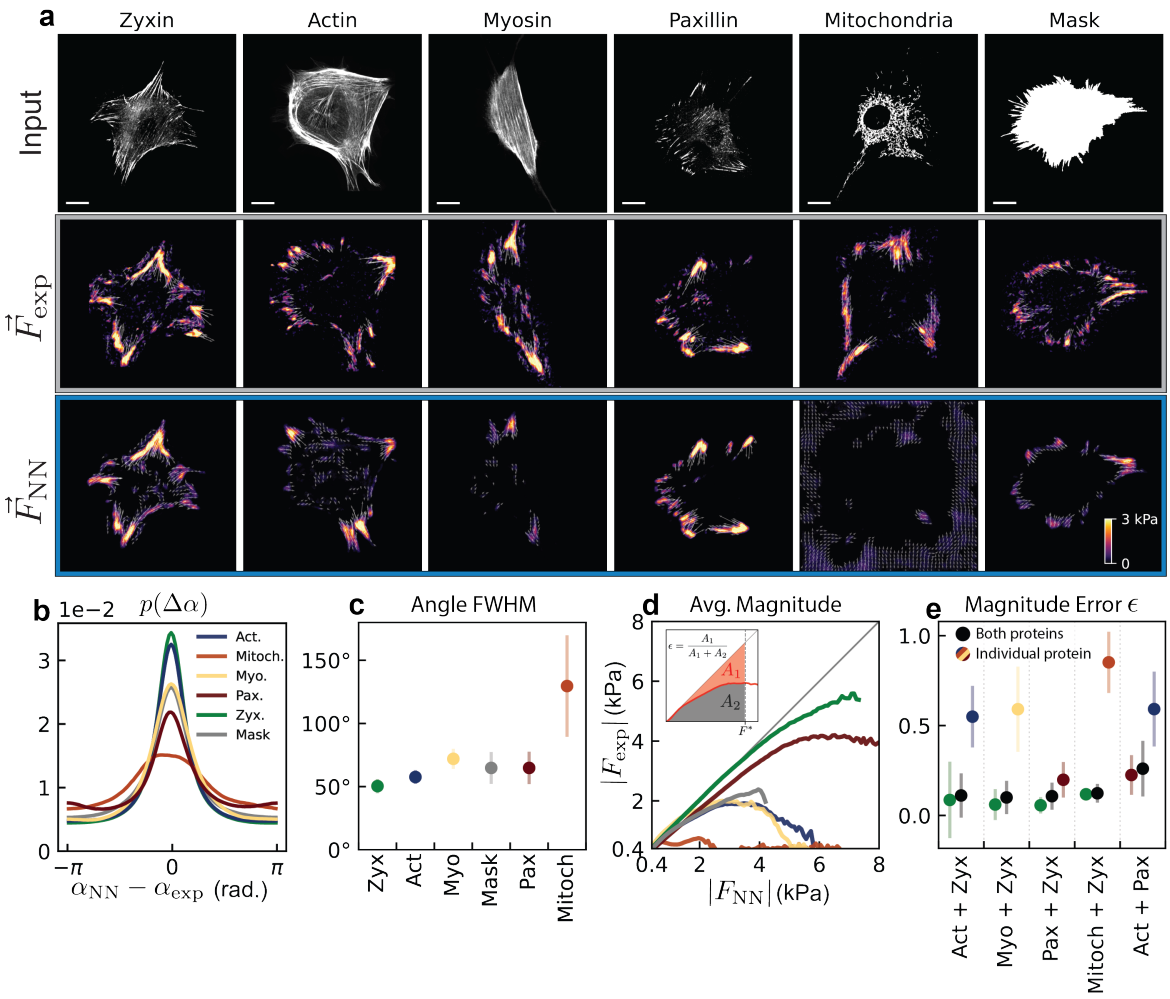

Figure 5.3: **Zyxin-trained networks outperform other cytoskeletal proteins.** (**a**) The predictive power of different cellular proteins inputs are compared by training neural networks on each protein individually. The comparison includes cytoskeletal proteins associated with force generation (actin, myosin), FA proteins (paxillin, zyxin), as well as a protein thought to be unrelated to force generation (mitochondria) and the binary cell mask. (**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, with the exception of the mitochondria network which 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 Methods and SI Fig. 10 for details).

as zyxin.

5.3 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 [210]. To determine which of these features was driving the U-Net performance, we tested the efficacy of other cytoskeletal proteins involved in force transmission, including: actin and myosin, the filaments and motors which make up the contractile network; paxillin, another focal adhesion 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. With the exception of mitochondria-trained networks, all networks learned to predict forces to some degree of accuracy, capturing the general localization and magnitude of traction stresses (Fig. 5.3a). The probability distribution of angular error $\Delta\alpha = \alpha_{\mathrm{NN}} - \alpha_{\mathrm{exp}}$ peaked around zero for all proteins, differing only in the width of the distribution about the true value (Fig. 5.3b). This distribution width was similar for networks trained on each protein, with the exception of mitochondria which showed a high angular variance (Fig. 5.3c). When comparing force magnitude predictions, we observed larger differences amongst the proteins, with the FA proteins zyxin and paxillin outperforming all others (Fig. 5.3d; SI Figs. 8,9). 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 (Fig. 5.3e; SI Fig. 8). Similarly, combining paxillin and actin also did not perform as well as zyxin alone (Fig. 5.3e). Although the cellular forces themselves are generated by many interacting proteins, a single focal adhesion 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 which

contains all the necessary information about the cell to predict forces. We proceed using our highest-performing neural network, which was trained using zyxin alone.

5.4 Zyxin-trained networks generalize across cell types and perturbations

While it is generally assumed that the underlying mechanics of contraction are universal [130], 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; Fig. 5.4a) and paxillin in colonies of canine epithelial cells (MDCK; Fig. 5.4d; SI Fig. 11). Without any retraining, the zyxin-trained U-Net generally predicted accurate traction force directions and magnitudes for both new cell types (Fig. 5.4b-c,e-f), that were comparable to differences between training on different cytoskeletal proteins (SI Fig. 12). This was true despite the MDCK data being taken on a softer substrate (2.8 kPa vs. 16 kPa shear modulus) using a different microscope. While these changes in experimental setup can induce small errors (see SI Figs. 13-15), 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 minutes at a basal contractile state before adding 5 μM of the Rho Kinase (ROCK) inhibitor Y-27632 for 45 minutes, and then washing out the drug and imaging for a final 45 minutes (Fig. 5.4g-h). 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 [137, 183], while the washout reversed each of these trends. Despite having never seen these drug perturbations, the network still predicted the overall changes in global traction forces (Fig. 5.4g) and the local changes at FAs (Fig. 5.4h) during both the drug treatment and the subsequent recovery following wash-out. Together, these results

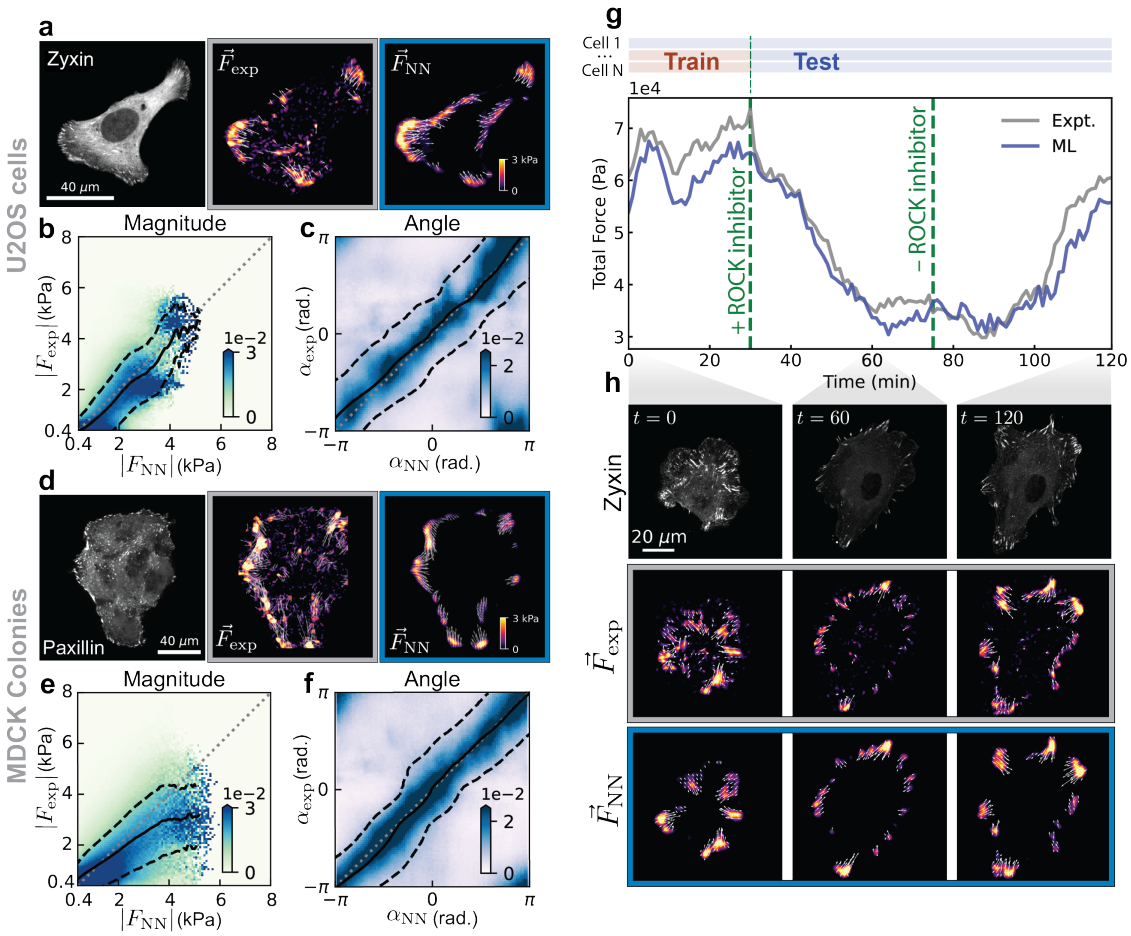

Figure 5.4: **Zyxin-trained networks generalize to new cell types and biological perturbations.** The fibroblast-trained network of Fig. 5.2 is evaluated on (**a-c**) individual U2OS cells expressing zyxin, and (**d-f**) colonies of MDCK cells expressing paxillin. Pixel-wise distributions of angle and magnitude predictions for U2OS cells (**b-c**) and MDCK cells (**e-f**), as in Fig. 5.2. (**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.

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.

5.5 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 (Fig. 5.2a). To probe how the network interprets these features, we trained U-Nets on random image crops of sizes ranging from 10 μm up to 130 μm in our input data (Fig. 5.5a). Even when trained on only a small fraction of the cell, these networks learned models which were accurate on average for both force magnitude (Fig. 5.5b-c) and direction (Fig. 5.5d-e). In both of these measurements, improvements in the prediction accuracy was negligible as the input size increased beyond $\sim$25 μm (Fig. 5.5c,e). 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 [124, 136, 151, 192, 195, 203] and FA distribution [60, 77, 114, 191] 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 [129]. 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 (Fig. 5.5f). While the network did not systematically respond to increases in cell edge curvature (Fig. 5.5h), we did find that force production increased with total cell size (Fig. 5.5i). This result is consistent with previous work showing that force generation scales with cell area [136], and further demonstrates that the network is sensitive to large-scale features of cell geometry.

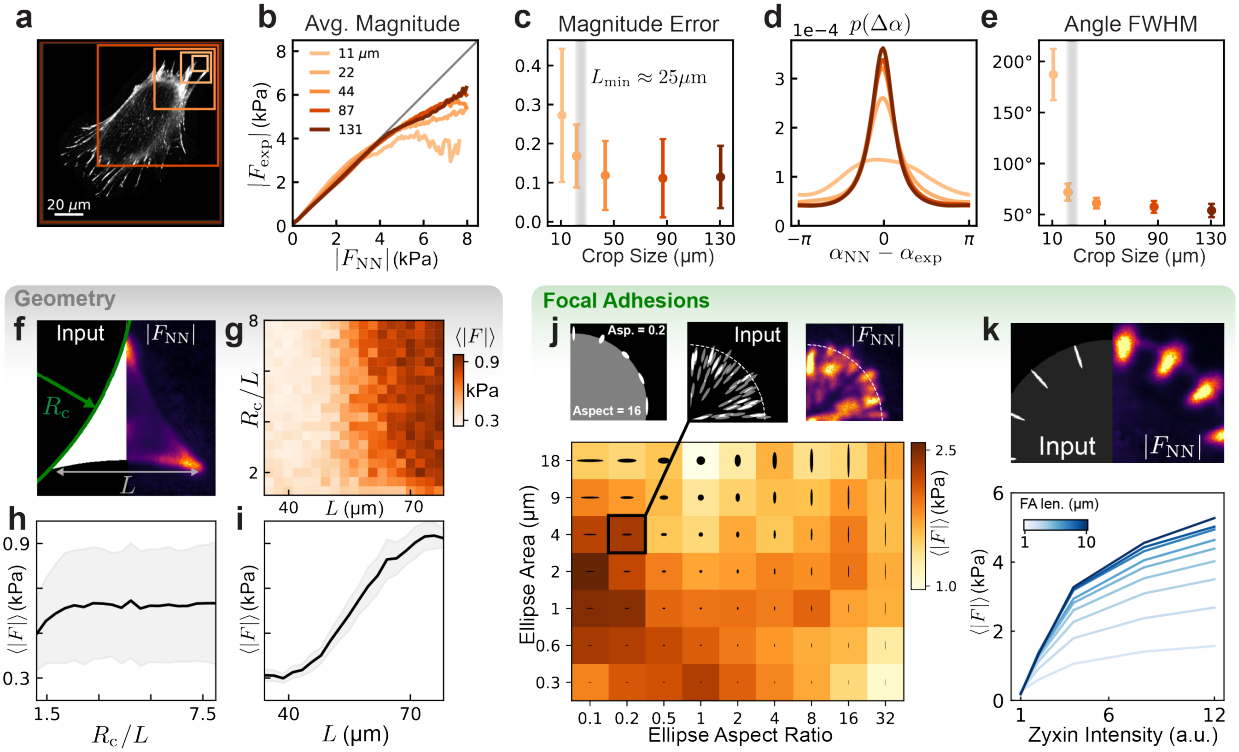

Figure 5.5: **Neural networks identify features of cell adhesion and morphology** (**a**) Networks are trained with varying crop sizes, ranging from 64 pixels ($\approx 10\mu$m) to 768 pixels ($\approx 130\mu$m). All networks are trained on the same data and have the same architecture as the U-Net of Figs. 5.2 and 5.4. (**b**) Average force magnitude (defined in Fig. 5.2) for varying crop sizes. (**c**) Magnitude error as a function of crop size, using the same metric defined in Fig. 5.3d. (**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**) Full width half maxima of the distributions in (**d**) as a function of crop size. FWHM reduces dramatically at a crop size of $\approx 25\,\mu$m, 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 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 probe the role of FA-like features, we created synthetic cells composed of elliptical "FAs" of varying area and aspect ratio which were distributed randomly throughout a circular cell (Fig. 5.5j; SI Fig. 16-17). The aspect ratio was defined with respect 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 $\sim 2\ \mu m^2$ and aspect ratio of ~ 0.1 (i.e. those pointed radially), consistent with experimental descriptions of FAs [62, 91, 151, 182]. We further investigated the role of FA intensity by creating circular cells with uniformly distributed ellipses of fixed intensity and length along the edge (Fig. 5.5k). Upon increasing the intensity of the ellipses, we found a nonlinear 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 [138]. 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 which allow it to accurately generalize predictions of force generation across cell types and biomechanical states.

5.6 The physical bottleneck: 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 typically lack the ability to make predictions under wide ranges of cell shapes and distributions of localized FAs [30, 52, 78, 124, 135, 136, 138, 177, 179, 203]. Here, we demonstrate how to incorporate zyxin into continuum mechanical models using neural networks, thereby learning

relationships between proteins and physical parameters which enhance the generalizability of physical models.

We resort to an existing model that views the cell as an effective two-dimensional active elastic gel adhered to a substrate [52, 124, 136]. 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 (Fig. 5.6a). 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 SI for details). Here, we extend this model by considering a spatially-varying adhesion field $Y(x)$, to account for the inhomogeneous distribution of focal adhesion sites in the cell [179]. 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

$$\vec{F}(x) = Y[\zeta](x)\vec{u}(x). \tag{5.1}$$

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 Figs. 1-4 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 (Fig. 5.6c; SI Fig. 18). We train the PBNN to predict parameters which minimize the mean squared error between predicted forces and the experimentally-measured forces. In each iterative training step, the adjoint method [194] 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 (Fig. 5.6c,d). The predicted force angles (Fig. 5.6e) and magnitudes (Fig. 5.6f), however, are less accurate on average across the dataset than those predicted by the unconstrained zyxin-trained U-Net of Figs. 1-4. 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 (Fig. 5.6e,f), 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 (Fig. 5.6g,h). 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 (Fig. 5.6d,h). 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 (Fig. 5.6i). 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 $\bar{Y}$ with the sum of zyxin values above a given threshold (Fig. 5.6h, SI Fig. 19). Upon increasing the threshold,

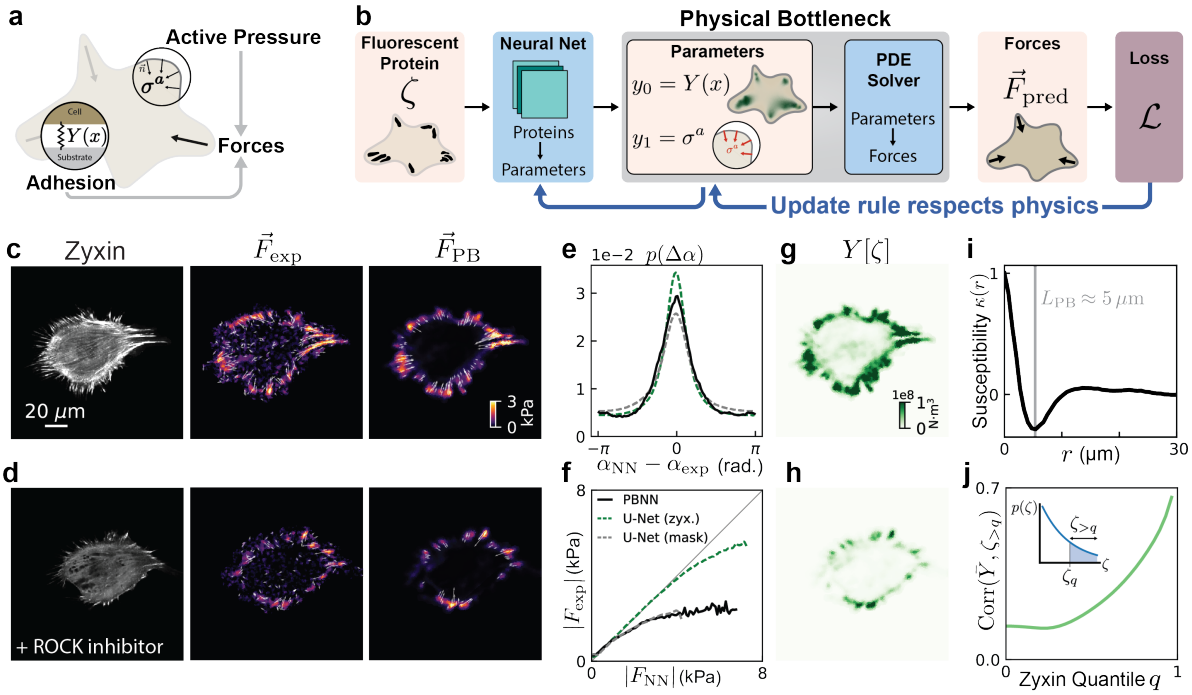

Figure 5.6: **The physical bottleneck: learning adhesion enhances an effective elastic model.** (**a**) We model the cell as an elastic medium which is subject to a uniform active contractility σ^a and is pinned to the substrate with a spatially varying adhesion field $Y(x)$. (**b**) A neural network 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 Fig. 5.3. 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, $\kappa(x_i, x_j) = \dfrac{\partial Y(x_i)}{\partial \zeta(x_j)}$ is sharply peaked within a radius of a few microns. (**j**) We correlate the average adhesion value $\bar{Y}$ with the amount of zyxin above a threshold ζ_q, where q denotes the qth quantile of the zyxin distribution (inset). $\bar{Y}$ correlates strongly only with the highest values of zyxin.

$\bar{Y}$ 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 due to the fact that a cell, unlike a passive lattice of masses and springs, continuously undergoes cytoskeletal remodelling 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 focal adhesions 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.

5.7 Green's function neural networks reveal 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 (Fig. 5.7a). Specifically, we assume that the force can be written as a function of machine-learned fields derived

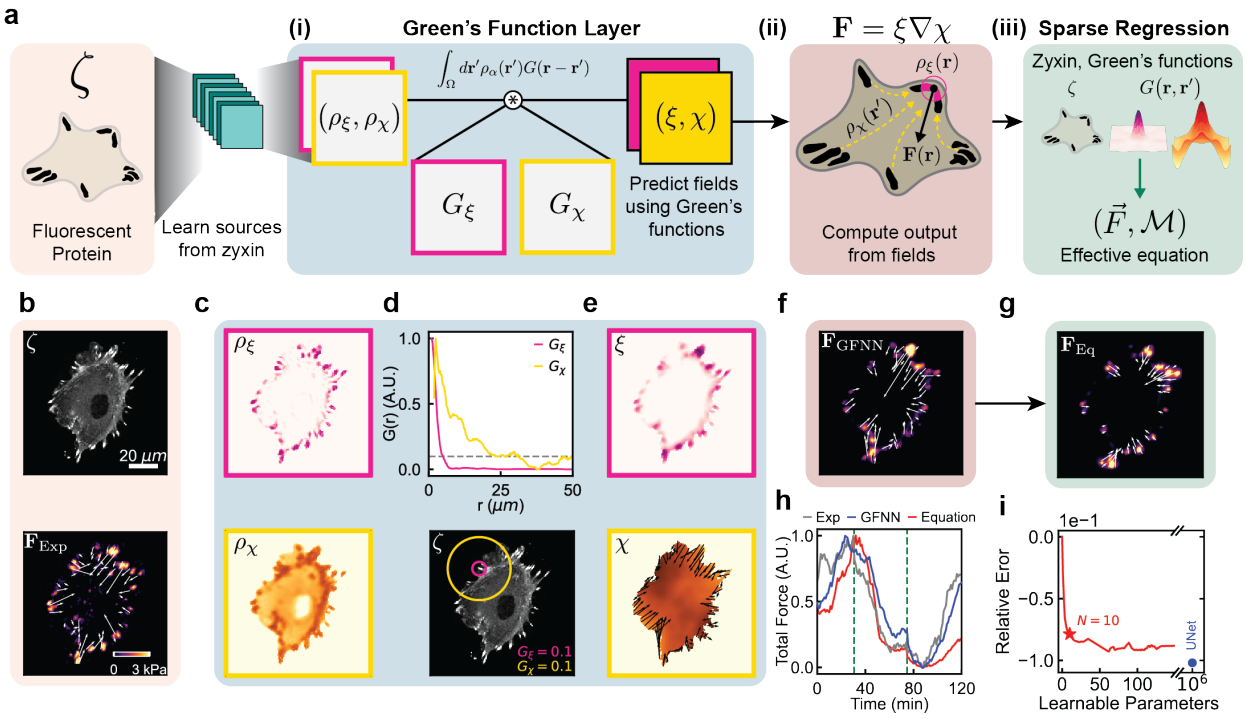

Figure 5.7: **Green's function neural networks: physics-agnostic model-building reveals length scales and effective equations.** (**a**) Green's-function neural networks (GFNNs) first extract the fields and long-range interactions needed to predict forces. Next, sparse regression builds effective equations fitting the machine-learned model. (**b**) The GFNN predicts traction forces from the zyxin intensity field (*top*), which we compare to 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 SI) based on the GFNN which predict the force field. (**h**) Time course of predicted forces during a ROCK inhibitor experiment. We compare the experimental forces (grey) to 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 to a baseline model $\mathbf{F} = 0$ with no learnable parameters. Star denotes the average performance of a 10-term equation.

from zyxin (yellow and pink boxes in Fig. 5.7a(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 (Fig. 5.7a(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 (Fig. 5.7b), 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 Fig. 5.7c-e) from the zyxin images which it uses to predict the traction stresses (see Methods and SI). 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 (Fig. 5.7f). Specifically, the GFNN learned two fields ξ, χ in terms of which predictions of the traction forces can be made

$$\vec{F} = \xi(x)\vec{\nabla}\chi(x). \tag{5.2}$$

Such a representation is reminiscent of Coulomb electrostatics, with ξ and χ analogous to the charge and electric potential, respectively (see SI Fig. 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 Eq. (5.2) resembles in form the physics-informed Eq. (5.1). The "charge" ξ in Eq. (5.2) identifies local peaks in zyxin intensity that are similar

to focal adhesions (Fig. 5.7c,e top). The Green's function for ξ, G_ξ, decays over a very short length scale $\sim 5\mu$m (Fig. 5.7d), suggesting that it is determined by local information at the adhesion site (Fig. 5.7a(ii),d). The "potential" χ is less localized and its Green's function, G_χ, accumulates zyxin information from a larger area of the cell (Fig. 5.7a(ii),c-e 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 focal adhesion "charge" and a cellular "potential".

To simplify this model further, we used sparse regression to build effective equations which approximate the traction forces (Fig. 5.7a(iii)). A qualitatively accurate analytical formula (Fig. 5.7g) can be obtained using only a handful of terms inspired by the GFNN (see SI for full equation). This formula is dramatically compressed compared to 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 (Fig. 5.7h,i). 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 (Fig. 5.7i). Our proposed pipeline demonstrates how to extract effective equations which map protein distributions to traction forces without knowing the explicit underlying relations. Although no physical input was used to derive them, the structure of Eq. 5.2 and the learned equation (see SI) are strikingly similar to Eq. 5.1. In particular, ξ and Y are both fields which accumulate zyxin information within focal adhesions, while $\vec{\nabla}\chi$ and $\vec{u}$ are vector fields which propagate information throughout the cell.

5.8 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

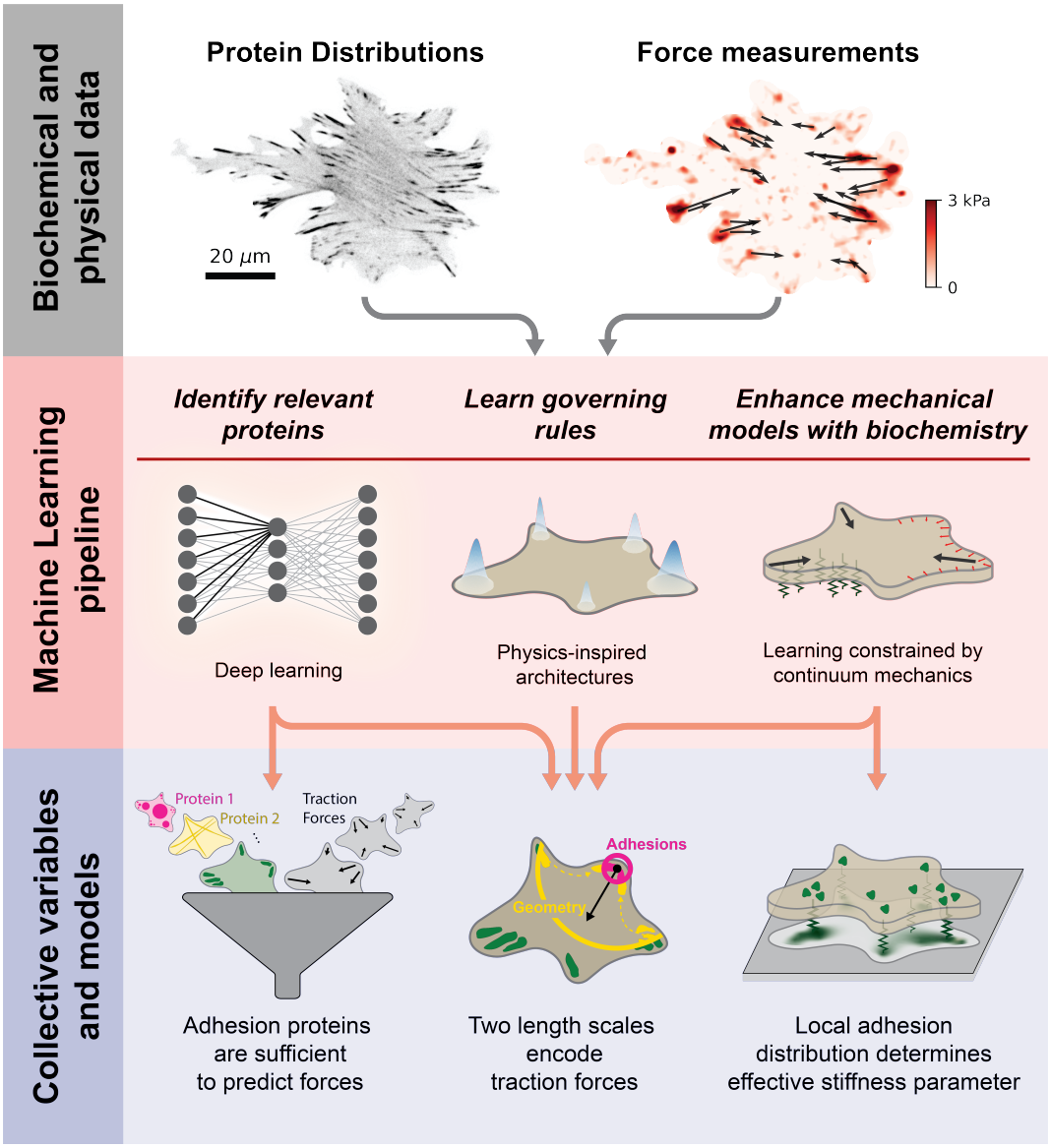

Figure 5.8: **Data-driven biophysical modeling**
We present a suite of machine-learning approaches to identify and interpret links between biological information contained in images of protein distributions and traction force measurements in adherent cells. Deep neural networks require no prior knowledge 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.

a single focal adhesion 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 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 (Fig. 5.8). 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 focal adhesions and describes the relationship between force magnitude and local zyxin intensity (Fig. 5.5j-k; Fig. 5.6g-j; Fig. 5.7c-e, 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 the PBNN it is accounted for in the PDE's boundary conditions (Fig. 5.5c,e; Fig. 5.6a-b; Fig. 5.7c-e,bottom). Moreover, in the PBNN and GFNN the fields

corresponding to long length scales $(\chi, \vec{u})$ and short length scales (ξ, Y) are coupled in a strikingly similar way (Eqs. 5.1-5.2). Neural networks are a complex black box and care must be taken when analyzing their behavior. By using multiple methods which 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're coupled. This information is present in the distribution of multiple focal adhesion proteins, which might explain a certain degree of interchangeability between zyxin and paxillin 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 focal adhesions [74, 106] 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 [70, 72, 89, 185]. 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 which 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 [6, 27, 28, 148, 150], epithelial tissue dynamics [39, 45, 164] and

morphogenesis [3, 121, 160, 181]. They could also be coupled to recent large quantitative datasets describing organelle positioning and interactions [201] to glean additional insights. These approaches represent a step forward towards harnessing the versatility of machine learning to tackle the complexity of living systems.

5.8.1 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 (Fig. 5.4 and SI Figs. 11-13). 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 (SI Figs. 14-15), which suggests that elastic substrates cannot be discarded. Beyond data quality, our analysis is limited by the use of two-dimensional image data. Cells are not two-dimensional 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.

CHAPTER 6

LEARNING THE DYNAMICS OF MORPHOGENESIS

Morphogenesis is the process whereby the body of an organism develops its target shape. The morphogen BMP is known to play a conserved role across bilaterians in determining the dorsoventral (DV) axis. Yet, how BMP governs the spatio-temporal dynamics of cytoskeletal proteins that drive morphogenetic flow remains an open question. Here, we use machine learning to mine a morphodynamic atlas of Drosophila development, and construct a mathematical model capable of predicting the coupled dynamics of myosin, E-cadherin, and morphogenetic flow. Mutant analysis reveals that it is BMP that sets the initial condition of this dynamical system according to the following signaling cascade. BMP establishes DV pair-rule-gene patterns that in turn set-up an E-cadherin gradient which creates a counter running myosin gradient through mechanochemical feedbacks. Using neural tube organoids, we unveil how BMP, and the signalling cascade it triggers, prime the conserved dynamics of neuroectoderm morphogenesis from fly to humans.[1][2]

6.1 Introduction

Morphogenesis is the process by which the shape of an organism emerges from the coordinated behavior of groups of cells. Turing's pioneering work traced morphogenesis to the presence of "chemical substances called morphogens, reacting together and diffusing through a tissue" [196]. Molecular biology has since identified these morphogens as molecules whose concentration modulates the expressions of various genes [65]. This in turn determines a

1. This chapter is adapted from a forthcoming manuscript: **How BMP governs the dynamics of neuroectoderm morphogenesis across species**, by Matthew Lefebvre*, Jonathan Colen*, Nikolas Claussen*, Fridtjof Brauns, Marion Raich, Noah Mitchell, Michel Fruchart, Vincenzo Vitelli, and Sebastian Streichan.

2. Text contains references to Boxes, Methods, and Supplementary Material. Boxes and selected Supplementary Material are included at the end of the chapter, while Methods are available in Appendix A. Additional Supplementary Material is available online.

cell's fate – namely, its future identity or that of its daughter cells. During early development, morphogen concentrations set up a spatial coordinate system and body axes on which the embryo organizes future tissues and organs [172]. However, it is force-generating proteins – not morphogens – that actually move each group of cells into the right place. Active mechanics describes how biology harnesses such physical processes [6, 119, 152]. A central challenge in developmental biology is to unveil general mechanisms by which morphogens control the active mechanics of shape-creating proteins. A paradigmatic example is the role of bone morphogenetic protein (BMP) signaling in establishing the neuroectoderm, the first step in forming the central nervous system [14, 15]. This role is evolutionarily conserved across organisms with bilateral symmetry, likely because it traces back to a common ancestor. Beyond BMP, many force-generating proteins have a conserved role across species, yet the exact process by which BMP controls these proteins is not known. This raises the questions: *how does BMP control the active mechanics that execute morphogenesis* and *is such a mechanism conserved across species?*

To understand the dynamics of morphogenesis, one needs to find causal relationships between biochemical and biophysical observables, such as protein distributions and cellular flow. Recent progress in light-sheet microscopy has yielded unprecedented glimpses into tissue and protein kinematics across the embryo [104]. However, these imaging advances come at a price: they produce very large datasets with little a priori knowledge of what minimal information (typically encoded in proteins) should be considered [157]. Machine learning (ML) is a powerful tool to work with such complex datasets [108]. It allows one to identify reduced sets of variables and use them to perform statistical inference or predict the future of a system [31, 122]. But high-performance ML models do not come with a Rosetta stone. To go beyond their black-box limitations and interpret them, we need to supplement ML techniques with physical and biological insights [93]. To unveil *how BMP controls the dynamics of morphogenesis across species*, we develop a pipeline that combines *in*

toto light sheet microscopy, interpretable machine learning, and theories of active mechanics. We then apply this integrated approach to perform a comparative study of neuroectoderm morphogenesis in *Drosophila* and in a stem cell-based human neural tube model.

In *Drosophila*, an initial major morphogenetic movement, part of gastrulation, is the elongation of the neuroectoderm along the anterior-posterior (AP) axis, known as germ-band extension (GBE) [86] (Fig. 6.1a-b). It involves transcription factors patterned in periodic stripes along the AP axis (stripes in Fig. 6.1c), that are encoded by homeobox domain containing genes known as pair-rule-genes (PRGs). PRG patterns have been proposed to organize GBE by regulating Toll-like receptors [141, 213]. However, the mechanism that would lead from Toll-like receptors to tissue flow along the axis of elongation remains unclear (see Box 1). Quantitative mechanical analysis using live-imaging of *Drosophila* has linked tissue flow to a contractility gradient along the dorso-ventral (DV) axis, the axis orthogonal to the AP direction [181]. A patterned force-generating protein, non-muscle myosin-II (myosin), causes cell interfaces on the ventral side to contract more than those on the dorsal side, which in turn drives tissue flows in the neuroectoderm (Fig. 6.1b). In addition to cell-level AP-polarization of cells, this suggests that morphogenetic flow also relies on a large-scale myosin imbalance along the DV axis. Recent work has found these myosin gradients are not only due to spatially-varying gene expression, but emerge from mechanochemical feedback loops [21, 75, 109] (Fig. 6.1d).

In humans, the neuroectoderm also undergoes major morphogenetic remodelling: an initial folding process creates the neural tube, the precursor of the brain [94, 205]. The mechanics of folding requires both basal motility of the surface ectoderm and apical contractility of the neural ectoderm [94]. However, how *Drosophila* PRGs and their human counterparts regulate force-generating proteins to perform these movements remains unknown [109, 141, 142]. Here, we propose and test the following hypothesis: the dynamics of force-generating proteins in both the human and the fly are regulated by mechanochemical

feedback loops involving the proteins themselves as well as transcription factors, under the overall control of BMP.

This paper is organized as follows. We first develop a pipeline for analyzing *Drosophila* embryos that combines light-sheet microscopy, active matter theories, and machine-learning. Using this pipeline, we identify a set of biophysical quantities predictive of their own future — tissue flow, myosin, and E-cadherin — and construct a set of equations describing their coupled dynamics. Next, we present a mutant analysis that reveals how BMP sets the initial condition of this dynamical system through a multi-step signaling cascade. First, BMP establishes DV-graded patterns that set-up a global E-cadherin gradient. Second, mechanochemical feedback convert this E-cadherin gradient into a myosin gradient in the opposite, or counter-running, direction. To probe the generality of this mechanism, we turn to experiments on human stem cell-based neural tube organoids that unveil how BMP, and the signaling cascade it triggers, prime the conserved dynamics of neuroectoderm morphogenesis from flies to humans.

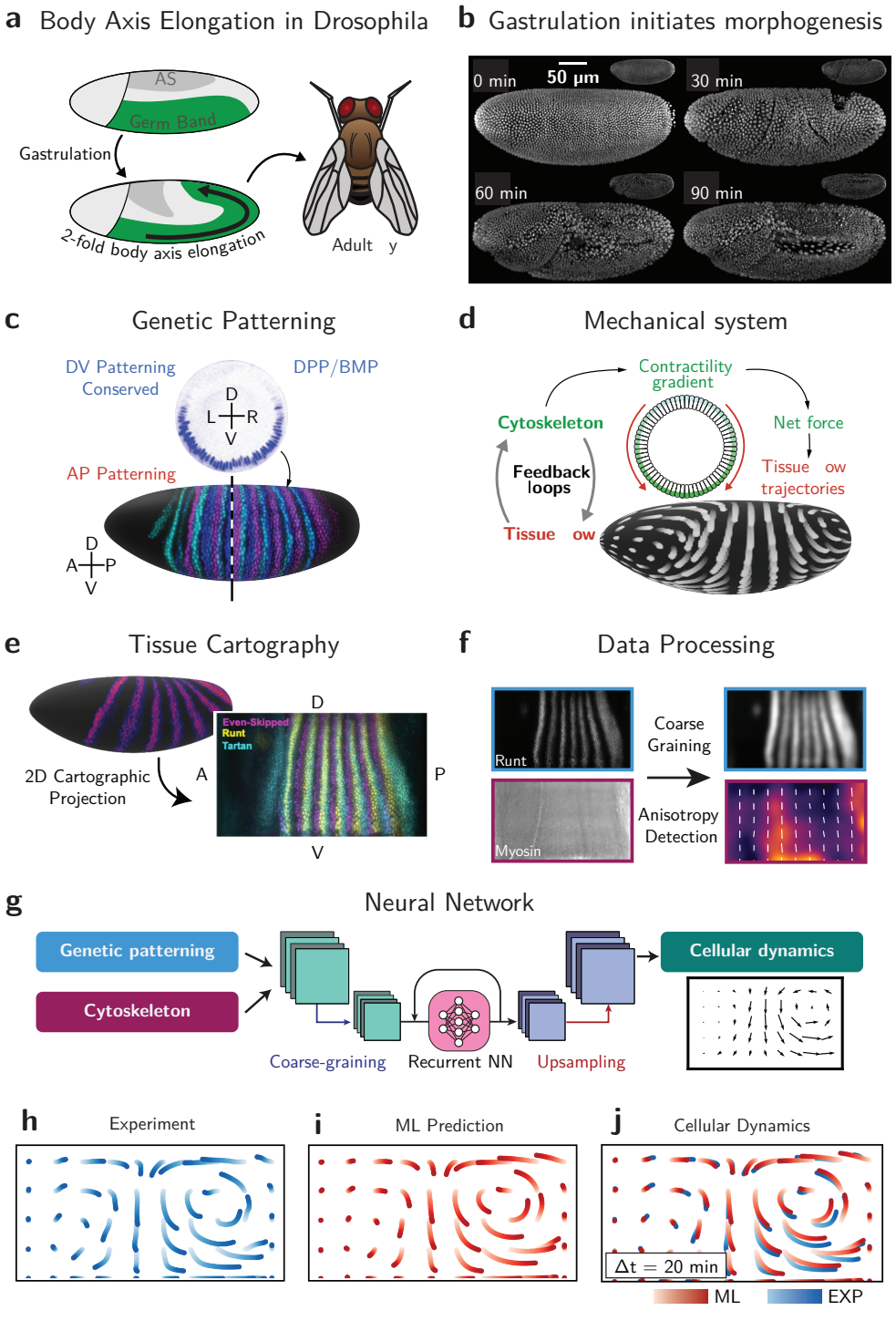

Figure 6.1: **Drosophila GBE as a system for understanding the interplay between genetic patterning and cytoskeletal force generation using machine learning.**

Figure 6.1: (cont.) (**a**) During Drosophila GBE, the body axis undergoes a 2-fold elongation over the course of 90 minutes. (**b**) Snapshots of nuclei from *in toto* live recording of *Drosophila* gastrulation, comprising invagination of the ventral furrow and GBE. (**c**) (Top, cartoon) The conserved morphogen DPP (the *Drosophila* homolog of BMP) patterns cell fates along the DV axis. (Bottom) The AP axis is patterned by striped expression of PRGs, three of which are shown (Even-skipped: blue, Runt: cyan, Paired: magenta, virtual overlay created from 3 different embryos). We find that PRGs are also graded along the DV axis. This patterning hierarchy shapes cytoskeletal recruitment and drives tissue flow. (**d**) A gradient of myosin recruitment along the DV axis generates forces that drive tissue flow. Mechanical feedback loops respond to tissue deformation, sculpting local patterns of cytoskeletal recruitment. (**e**) Tissue cartography projects the curved 3D embryo surfaces onto a 2D plane. (**f**) Additional image processing includes spatial downsampling/smoothing and optional anisotropy detection via a radon transform algorithm. (**g**) A NN forecasts tissue dynamics from the measured biological initial conditions. The network is a residual autoencoder which maps the inputs to a latent vector, predicts dynamics using a recurrent layer, and then translates the latent vector sequence into flow fields. (**h**) Trajectories of test points in the flow fields measured in experiment. (**i**) Trajectories in flow fields predicted by the NN. (**j**) Overlay of (**h**) and (**i**) showing agreement between over 20 minutes of forecasted behavior.

6.2 Neural networks learn tissue dynamics from myosin distribution

To understand the rules that control *Drosophila* GBE, we aim to isolate a small set comprising the most relevant proteins, and construct biophysical equations to describe their pattern and dynamics, generating testable hypotheses. To do so, we leverage a morphodynamic atlas of *Drosophila* gastrulation which contains movies of *in-toto* protein expression for various patterning genes and force-generating cytoskeletal components. We [125] (Fig. 6.1c-d) mine the data using machine learning [31, 108, 122]. We first train deep neural networks (NNs) to learn tissue dynamics using different sets of biophysical quantities contained in the atlas (Fig. 6.1d-e). As detailed in the SI (Fig. 6.9), this allows us to identify a restricted set of biophysical quantities that are predictive of both future tissue flow patterns and patterns of junctional myosin recruitment patterns . Our results corroborate the crucial role of myosin in driving tissue flow during GBE, in agreement with previous work relating the instanta-

neous myosin distribution and tissue flow [181]. In short, myosin creates forces that put the cells in motion. When given an initial condition of myosin measured prior to ventral furrow (VF) formation, a trained NN can forecast tissue flow for 20 minutes and maintain excellent agreement with experiments throughout the initial period of GBE (Fig. 6.1f-h; this entire process takes approx. 45 min).

6.3 Minimal active matter models require control fields and mechanical feedback

To gain further insight into tissue dynamics during GBE, we performed a principal component analysis (PCA) of dynamic time series data representing the entire timecourse of GBE. This technique organizes data into features called "principal components" (PCs), sorted by their importance in the available data. As shown in Fig. 6.2, a small number of PCs can account for the main features of GBE morphodynamics. Over the course of GBE, a gradient of junctional myosin recruitment develops along the dorso-ventral axis (Figs. 6.2c'-f'). Prior to the initiation of GBE ($t = 0$ min post ventral furrow initiation), there is very little motion of the germband. Over time, tissue flow increases, and the flow field evolves into a quasi-stationary state composed of four vortices located at antero-lateral and postero-lateral positions on either side of the embryo (t from 5 to 20 mins), see Figs. 6.2c-f. Crucially, patterns of myosin recruitment and the spatiotemporal positions at which tissue flow vorticies develop are both robust across multiple embryos. This behavior holds even in embryos which lack a VF (ex. *twist* and *snail* mutants) and therefore do not undergo mechanical symmetry breaking [75].

In the SI (Fig. 6.10-6.11), we attempt to reproduce these qualitative features of myosin and flow dynamics using minimal active matter models. Under reasonable biological assumptions, a dynamical model including only myosin and flow cannot reproduce the long-lived GBE-like state observed in experiments (Fig. 6.2c). Because myosin and velocity are nearly

uniform at $t = 0$, such a model can only create a myosin gradient (and the GBE flow it produces) via a pattern-forming instability. We rule this out because it would lead to patterns randomly shifted along the DV axis, in contradiction with experimental observations. In contrast, a simple model where a control field modulates myosin concentration can produce a quasi-stationary state. This control field induces a myosin gradient along the DV axis which in turn causes a GBE-like flow with four vortices. This quasi-stationary state is prolonged using more complex models including advection and junction rotation, by including a mechanical feedback coupling strain-rate to myosin recruitment as suggested by previous works [57, 75, 109, 181, 211] (see Fig. 6.12-6.13).

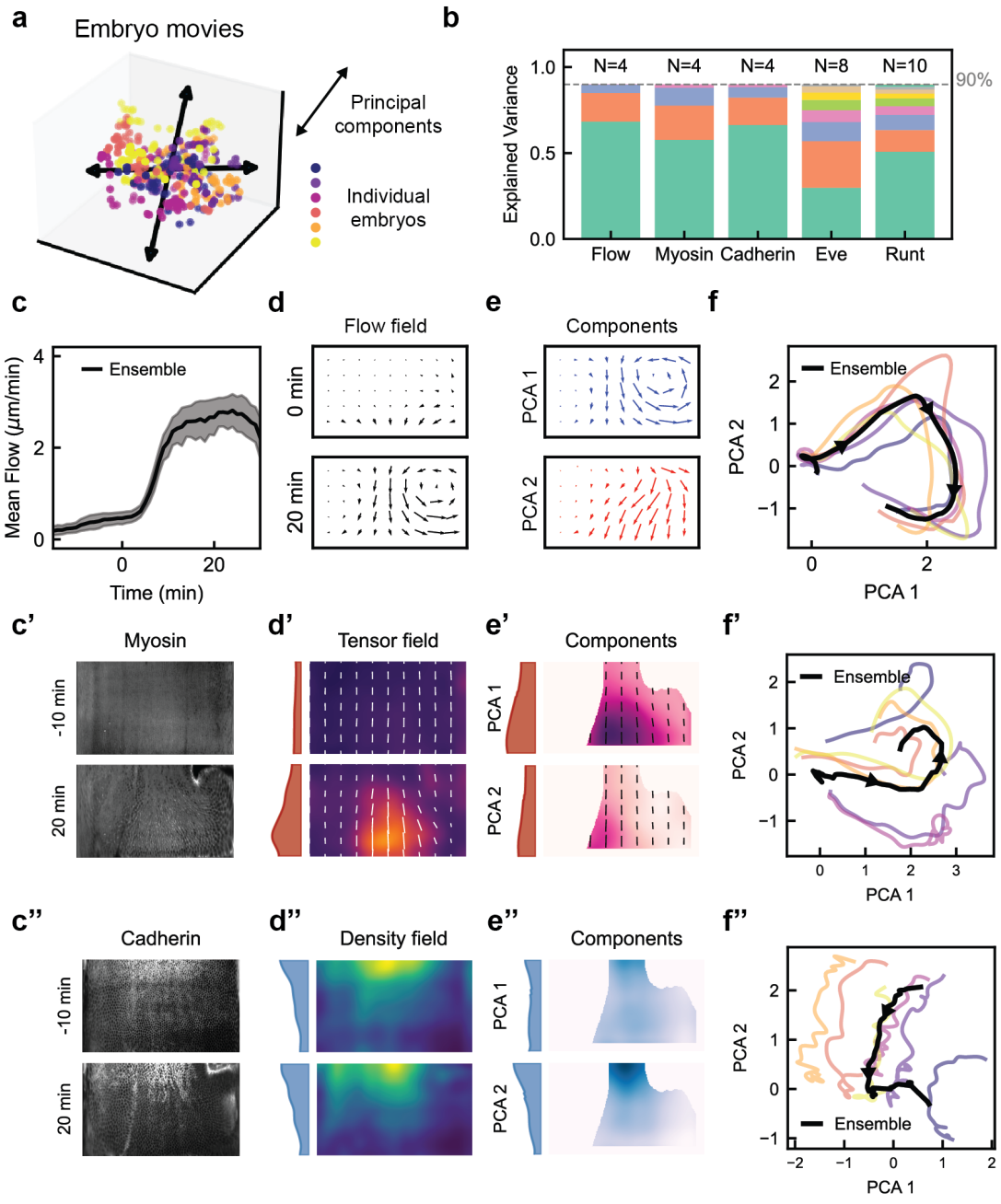

Figure 6.2: **Dimensional reduction describes patterns on the embryo surface.**

Figure 6.2: (cont.) (**a**) Principal component analysis (PCA) represents high-dimensional data using linear combinations of learned components. Each frame from the morphodynamic atlas [125] is an individual sample with data dimension equal to the number of pixels. (**b**) PCA for different fields in the morphodynamic atlas. N is the number of components needed to capture 90% of data variation. Flow and the cytoskeletal proteins myosin and E-cadherin require few PCs ($N = 4$) while PRGs require more ($N \geq 8$). (**c**) Average flow magnitude over time, with VF at $t = 0$ min. (**d**) Representative snapshots of the flow field for VF and GBE. (**e**) Primary PCs for the flow field. The first captures GBE flow while the second is a DV component which produces VF flow. (**f**) Projection of flow movies onto primary PCs, demonstrating smooth and reproducible dynamics. (**c**'-**f**') PCA for myosin showing that the primary behavior is a developing DV gradient. Here, we mask out the cephalic furrow and posterior midgut regions which become prominent at later times. (**c**"-**f**") PCA for E-cadherin, whose PCs also show a DV gradient opposite myosin. The similarity between the PCA of myosin and E-cadherin fields suggests their evolution may be coupled.

6.4 E-cadherin may intermediate between transcription factors and force-generating proteins

We now aim to identify the control field that establishes the myosin recruitment gradient. Mutant phenotypes of the PRGs Even-Skipped and Runt exhibit abnormal myosin patterns and abrogated tissue flow during GBE, demonstrating their importance for myosin recruitment [57, 86, 109, 213]. However, we find that both Even-Skipped and Runt require nearly twice as many PCs to encode as myosin or flow (Fig. 6.2b). This reflects the greater spatio-temporal complexity of PRG expression profiles which have clear AP oriented stripes (Fig.6.4b,c). In addition, these complex PRG expression patterns are continuously advected by the flow [109]. These observations are consistent with the idea that while PRGs are important for myosin recruitment and ultimately for tissue flow, downstream cytoskeletal effectors may be necessary to organize myosin recruitment and tissue flow.

To this end, we considered E-cadherin, a hallmark component of the sub-apical adhesion cortex. This complex and dynamic network of signaling molecules and cytoskeletal proteins also recruits myosin motors. Because of this, we expect that features which affect myosin recruitment may be reflected in the expression profile of E-cadherin (or other integral adhe-

sion complex proteins with similar expression profiles, see SI online). Indeed, it has recently been shown that high levels of E-cadherin can inhibit junctional recruitment of myosin [184].

E-cadherin exhibits less complex spatio-temporal behavior than PRGs, requiring the same number of PCs as myosin and tissue flow (Fig. 6.2b). E-cadherin is expressed in a dorso-ventral gradient before significant apical myosin recruitment or the onset of tissue flow in the germband (Fig. 6.2c-c"). The E-cadherin profile maintains higher expression at the dorsal pole throughout GBE (Fig. 6.2c"). When myosin recruitment initiates during VF invagination, the direction of the emerging gradient is opposite that of E-cadherin (Fig. 6.2d'-d"). In both cases, expression levels vary smoothly along the DV axis. We note that in other contexts, opposing patterning gradients are known to shape gene expression profiles [10, 11, 47, 103, 127, 200, 212].

These observations suggest that the E-cadherin profile may reflect features of the sub-apical adhesion complex which play a role in organizing myosin recruitment. Indeed, while mutant phenotypes demonstrate PRG expression is necessary for tissue flow and myosin recruitment [57, 86, 109], our results suggest that E-cadherin or other components of the sub-apical adhesion cytoskeleton (such as Bazooka/Par3) may play a more direct role in regulating myosin expression during GBE.

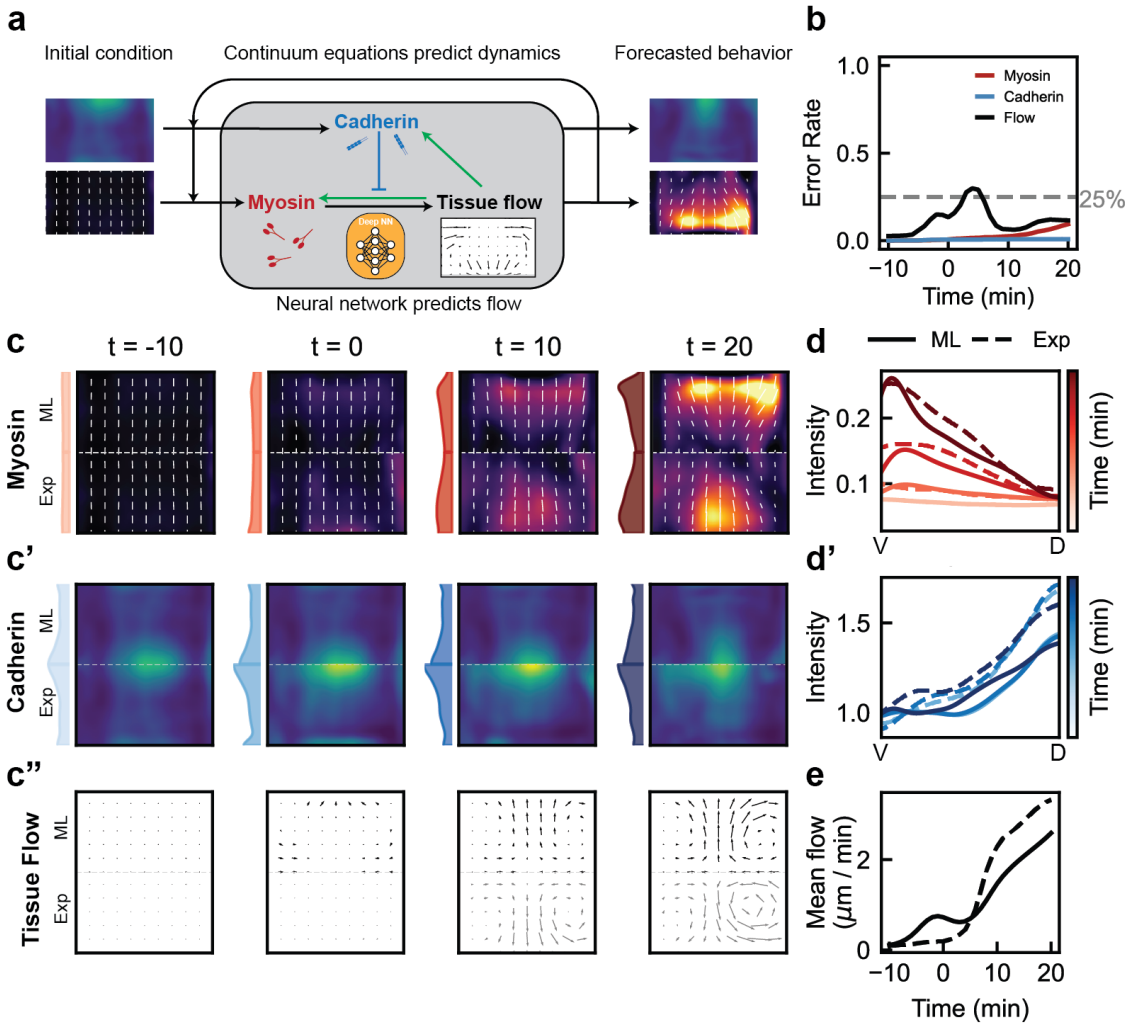

Figure 6.3: **Predicting dynamics of cytoskeletal proteins and flow.**

Figure 6.3: (cont.) (**a**) Schematic of the machine-learned dynamical system. At each time step, a NN (Fig. 6.1e-h) translates myosin to the current tissue flow. Eqs. 6.1-6.2 predict the time derivatives of the myosin and E-cadherin fields, which are integrated to forecast embryo behavior from initial conditions. (**b**) Error rate of predictions over 30 minutes, beginning 10 minutes before VF. The error for each field remains within ≈25% over the course of the movie. (**c**) Snapshots of myosin evolution at 10-minute increments. The top half of each image shows the model prediction (ML) while the bottom half shows the experimentally-observed behavior (EXP). Additional rows show E-cadherin (**c**') and tissue flow (**c**") at the same timepoints. (**d**) Average intensity of predicted myosin (solid line) and experiment (dashed line) along a DV cut through the center of the embryo, showing that a gradient develops from uniform initial conditions (**d**') Similar DV cuts for the E-cadherin field, showing an initial DV gradient that is maintained throughout the trajectory. (**e**) Average flow magnitude predicted by the model and measured in experiment (dashed line). The model predicts the onset of GBE flow arising from the developing DV contractility gradient.

6.5 Machine learning yields interpretable dynamics of neuroectoderm morphogenesis

Our current tentative picture is as follows: myosin puts cells into motion leading to tissue flow during GBE and our neural networks identify myosin as a necessary ingredient for forecasting these tissue dynamics. In minimal continuum equations, an additional DV-graded field is needed to organize myosin recruitment. Our PCA results suggest that E-cadherin is a possible candidate for this second field. To assemble these ingredients – myosin, tissue flow, and E-cadherin – into a mathematical model that describes experimental data, we use a ML technique known as SINDy [26]. Our goal is to learn biophysical equations for the coupled dynamics of myosin and E-cadherin that are compatible with the biological constraints delineated in previous sections and the SI (Fig. 6.10-6.13). Our procedure and the equations it identifies are detailed in 6.9.2.

These dynamical equations must be supplemented with a mathematical relation that explains how flow is instantaneously generated by myosin. At a qualitative level, this relation can be captured by a Stokes equation [181] (Eq. 6.3 in SI) which models the tissue as a viscous fluid. However, to reach quantitative agreement with experiments, we must account

for external forces due to a biological phenomenon not captured by a Stokes equation (VF invagination). To do so, we instead use a NN to instantaneously map between myosin and flow. This NN alongside evolution equations for myosin and E-cadherin forms our complete dynamical model (Fig. 6.3a and 6.9.2). Our hybrid model predicts the development of a dorso-ventral myosin gradient opposite to the E-cadherin gradient (Fig. 6.3c-d') and the transition to a vortical flow pattern (Fig. 6.3c"-e). We accurately forecast the developmental trajectory for up to 30 minutes, starting 10 minutes before initiation of the VF, and ending during GBE (Fig. 6.3a-b).

6.5.1 Tissue flow requires cytoskeletal gradients downstream of BMP/DPP signalling.

Our model predicts that convergent extension tissue flow during GBE requires a DV oriented contractility gradient, which we hypothesize is controlled by a gradient in cytoskeletal organization, as measured by E-cadherin. We can test this hypothesis directly using Drosophila DV patterning mutants in which all cells along the DV axis adopt the same fate. We characterize tissue flow, and cytoskeletal expression profiles in both lateralized ($Toll^{RM9}$) and dorsalized (spz^{4}) embryos, in which cells uniformly adopt lateral, neural ectodermal fates or dorsal ectoderm fates respectively. Myosin recruitment is uniformly low in dorsalized embryos, and E-cadherin expression is uniformly high (Fig. 6.4a'-c',e-f)). In contrast, myosin recruitment is uniformly high in lateralized embryos whereas E-cadherin expression is low (Fig. 6.4a"-c",e-f)). In both cases, graded cytoskeletal expression is eliminated (compare to Fig. 6.4a-c). Importantly, tissue flow is nearly eliminated in both DV patterning mutant classes (Fig. 6.4g). These results demonstrate that junctional myosin recruitment is not sufficient to drive convergent extension flow during GBE. Indeed, lateralized embryos have high levels of myosin recruitment uniformly. Rather, the generation of a contractility gradient is necessary for flow.

Building off of the observation that DV patterning mutants have serious phenotypes during GBE, we next asked which component of the DV patterning system mediates this effect. We investigated the effect of Decapentaplegic (DPP, *Drosophila* homolog of BMP) signaling during GBE. DPP is a key morphogen responsible for differentiation of cell fate along the Drosophila DV body axis. Hypomorphic reduction of DPP signaling flattens the DV gradient of both E-cadherin expression (Fig. 6.4b"') and myosin recruitment (Fig. 6.4c"'). Flow is also abrogated in dpp mutants (Fig. 6.4g). These results suggest that the regulatory effect of DV patterning during GBE –sculpting the expression of the contractile cytoskeleton – requires DPP/BMP signaling.

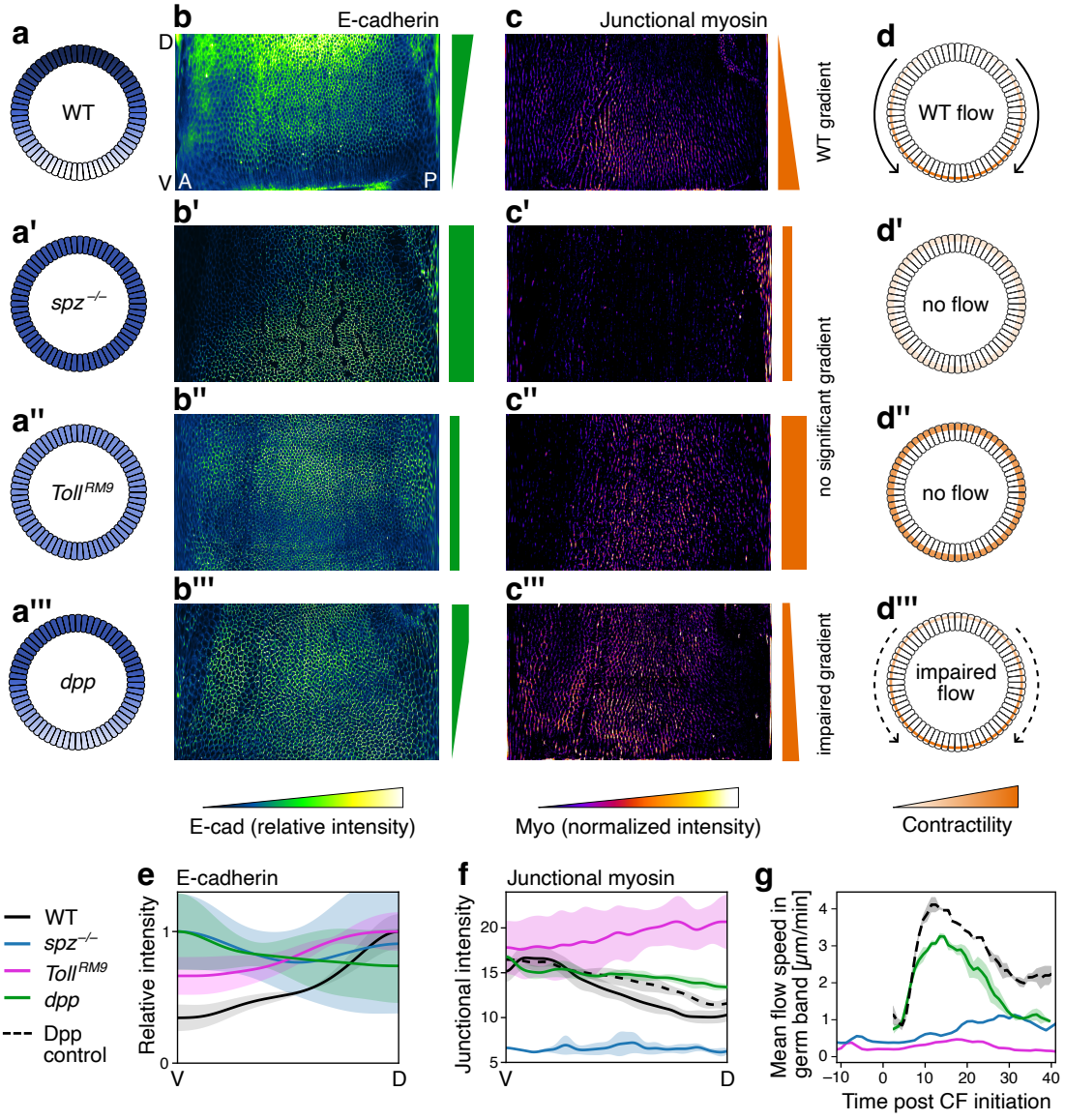

Figure 6.4: **DV patterning mutants abolish cytoskeletal gradients and tissue flow.**

Figure 6.4: (cont.) (**a**–**a‴**) Cartoon of DV patterning "clock" of the *Drosophila* gastrula. Shading indicates different genetically patterned cell fates. (a") In $Toll^{RM9}$ and (a') spz^4 embryos, cells around the entire DV circumference uniformly adopt a ventro-lateral or dorso-lateral fate respectively. (a"') In *dpp* mutants (*Drosophila* BMP), dorsal-most fates are eliminated, and lateral fates shift towards the dorsal pole. (**b**–**b‴**) E-cadherin at $t = 0$ minutes post CF initiation in four genetic conditions. Bars indicate the presence or absence of an expression gradient. E-cadherin in (**b"'**) was measured in dpp^4 , $snail^{IIG05}$ double mutants. (**c**–**c‴**) Cytosolic-normalized myosin at $t = 0$ minutes in four genetic conditions. Myosin in (**c"'**) was measured in $Dpp^{hin46/+}$ mutants. (**d**–**d‴**) Gradients of myosin contractility create forces that drive tissue flow. DV-patterning mutants abolish ($Toll^{RM9}$, spz^4) or impair ($Dpp^{hin46/+}$) tissue flow (quantified below). (**e**) Quantification of E-cadherin gradient along the DV axis in DV mutants ($N = 3$ embryos for each genotype). E-cadherin in (**e**) was measured in dpp^4 , $snail^{IIG05}$ double mutants. (**f**) Quantification of junctional myosin levels in $Toll^{RM9}$,spz^4, and WT. Myosin is high in $Toll^{RM9}$, low in spz^4, and graded in WT. $Dpp^{hin46/+}$ shows an intermediate phenotype with impaired grading. Data in (**e-f**) based on $N = 3$ embryos for each genotype. (**g**) Quantification of tissue flow demonstrates that flow is abolished in both $Toll^{RM9}$ and spz^4 mutants. These embryos were imaged at 22^o C. $Dpp^{hin46/+}$ mutants impair but do not abolish tissue flow. Both $Dpp^{hin46/+}$ and $Dpp^{+/+}$ control embryos were imaged at 27^o C which increases the speed of gastrulation over embryos imaged at lower temperatures [125].

6.6 BMP signaling may pattern E-cadherin via homeobox genes

How does DPP/BMP regulate cytoskeletal recruitment and drive tissue flow? Previous work has demonstrated that the PRGs – effectors of AP patterning information – are necessary for cytoskeletal recruitment and tissue flow during GBE [86, 141, 213]. Intriguingly, we find that the PRGs exhibit DV oriented gradients of expression that fall roughly into three classes (Fig. 6.5a-b'): (i) Runt, Paired, and Sloppy-paired expression is low dorsally and high ventro-laterally (Fig. 6.5a-b and SI online for variation along a single Runt stripe), (ii) Hairy expression is high dorso-laterally, low ventro-laterally, and low at the dorsal pole (iii) Even-skipped and Fushi-Tarazu expression is uniform along the DV axis (Fig. 6.5b-b'). DV-modulation is also observed in two transcriptional targets of the PRGs that are known to affect cytoskeletal recruitment and GBE tissue flow downstream of PRGs – Toll-6 (a toll-like transmembrane receptor, TLR) and Tartan [140, 141] (Fig. 6.5b).

In lateralized (Fig. 6.5d,f)and dorsalized (Fig. 6.5f) embryos ($Toll^{RM9}$), graded Runt expression is abolished, although striped expression along the AP axis is unaffected. In hypomorphic DPP mutants, the slope of the DV Runt gradient is abrogated (Fig. 6.5e,f). Together, these observations are consistent with the abolition of graded cytoskeletal recruitment and tissue flow in DV patterning mutants we observed above, and suggest that PRGs might be effectors of DV regulation during GBE: DPP modulates PRG expression, which in turn modulates the cytoskeleton, creating the DV gradients required for tissue flow.

We now test this hypothesis more directly by analyzing PRG mutants and investigating if the DV oriented expression gradients, and not only the overall levels or the anisotropy, of cytoskeletal components are affected. Even-skipped eve^{R13} mutants exhibit a shallower E-cadherin gradient than control embryos prior to axis elongation (Fig. 6.5g,g' and SI). Previous work established that these mutants have lower myosin levels overall [141, 213]. Note that while Even-Skipped expression is not strongly DV-graded (Fig. 6.5b), its downstream effectors, such as Toll-6 [141] are.

Next, we over-expressed (OE) Even-Skipped under UAS control, driven by ubiquitous maternal Gal4 drivers (*67,15≫UAS-eve* see materials and methods for details). In response, we observe higher levels of anisotropic myosin extending towards the dorsal pole and a weaker global myosin gradient (Fig. 6.5h-h"). Consistent with our expectations, tissue flow in both eve^{R13} and *67,15≫UAS-eve* mutants is weaker than in control embryos (Fig 5i). Note that we observe high levels of anisotropic myosin recruitment in *eve-OE* mutants, even though Even-Skipped AP stripes become less pronounced in the presence of ubiquitous Even-Skipped expression, in contrast to the prediction of the *heterotypic juxtaposition model* (See Box 1) [213].

Together, these observations support the hypothesis that DPP/BMP acts via the AP-patterned PRGs to establish a DV-graded profile of E-cadherin expression before tissue flow begins in the germband. This gradient in cell adhesion then shapes the extent of myosin

recruitment along the DV axis. The resulting contractility gradient drives tissue flow. A visual summary of this proposed mechanism is show in Fig. 6.5c. Based on our model above, previous work [75, 109], and the results we obtained from over-expression of Even-Skipped, we propose that PRGs do not recruit myosin directly [213], but indirectly by modulating mechanical feedback loops. For instance, the optogenetic experiments of Ref. [75] showed that feedback responding to cell stretching is weaker dorsally than ventrolaterally, potentially due to different PRG expression levels.

Finally, we do not exclude the possibility that the DV patterning could influence the cytoskeleton via a PRG-independent pathway. However, patterns of cytoskeletal recruitment are dramatically altered in PRG mutants [141, 213]. In agreement with these observations, our findings: (i) that PRGs are graded along the DV axis, (ii) and that cytoskeletal gradients are weaker in PRG mutants — imply that at least part of the cytoskeletal DV patterning is mediated via the PRGs.

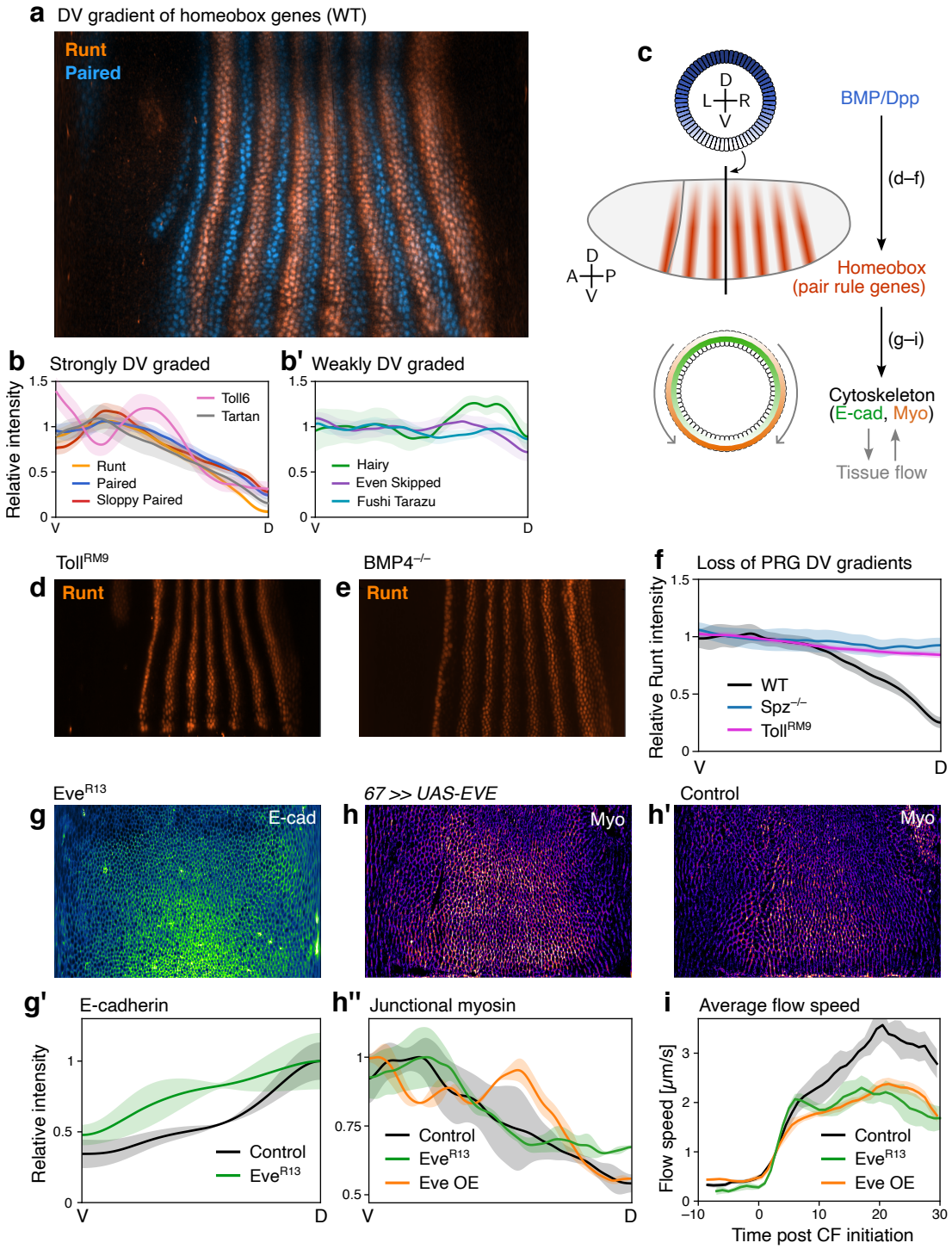

Figure 6.5: **DV gradients of cytoskeletal protein expression are driven by DV gradients of AP patterning genes.**

Figure 6.5: (cont.) (**a**) *Drosophila* PRGs are patterned along the DV axis. Runt and Paired are striped along the AP axis and graded along the DV axis, with less expression dorsally. (**b-b'**) Quantification of the DV gradients of 6 PRGs. Shaded areas indicate standard deviations across samples. (**b**) Includes two transmembrane receptors - Toll-6 and Tartan - that mediate the influence of PRGs on the junctional cytoskeleton [140, 141]. (**c**) Cartoon of proposed mechanism: DPP/BMP acts through homeobox genes to pattern the cytoskeleton. (**d-f**) Mutating the DV patterning system (**d**: $Toll^{RM9}$; **e**: $Dpp^{hin46/+}$) abolished the DV gradient of PRGs. (f) Quantifications of a representative PRG, Runt. (**g-g'**) A null allele of Even-skipped (eve^{R13}) reduces but does not abolish the DV-gradient of E-cadherin. ($N = 3$ embryos). (**h-h'**) Ubiquitous over expression of Even-Skipped (*67,15≫UAS-eve*) drives ectopic myosin expression. (**h"**) Myosin gradients are reduced in *67,15≫UAS-eve* ($N = 2$ embryos for each curve). (**i**) Tissue flow is slower but not abolished in both eve^{R13} and *67,15≫UAS-eve* ($N = 3$ embryos per genotype). All live embryos imaged at 22^o C.

6.7 Comparing *Drosophila* to a human stem-cell based model

BMP plays a conserved role in determining cell fates in the neural ectoderm [14] across species. In Drosophila, the neural ectoderm accounts for much of the tissue which undergoes convergence and extension during GBE. We hypothesize above that BMP signaling results in a graded DV pattern of homeobox genes (PRGs). Graded PRG expression along the DV axis then modulates mechanical feedback loops which control cytoskeletal proteins and ultimately drive tissue flow (Fig. 6.5c). We were interested in understanding the extent to which analogous mechanism could be at play in other systems where BMP signaling is necessary.

For this purpose, we turned to a human stem cell-based model of neural tube morphogenesis [94]. In this model, stem cell cultures confined on 2D micropatterns form 3D lumina upon matrigel addition. Lumina are then exposed to neural induction media supplemented with TGF-β inhibitor (SB 431542) and BMP4 which reproducibly drives fold formation in a region of neural ectoderm surrounded by surface ectoderm [94]. Fate differentiation and tissue flow depend on the generation of a radial BMP4 expression gradient, with levels of BMP4 activity at structure edges (Fig. 6.6a). As represented in Fig. 6.6a-b', folding of the neural ectoderm occurs 24 hours after BMP4 exposure and leads to tube closure by 120

hours after BMP4 exposure [23, 94]. Myosin inhibition prevents this neural tube closure [94], suggesting that a myosin-driven contractility gradient is essential for tissue morphogenesis.

Focusing on the first 24 hours post BMP exposure (hpb), we analyze the spatial distributions of several proteins. First, we find that human orthologs of *Drosophila* PRGs (which belong to the class of homeobox transcription factors) exhibit graded patterns at 24 hbp: the gene Zic2 is more strongly expressed in the center of the neural plate (the future neural epithelium) than at the periphery (the future surface ectoderm), while Hes1 is more strongly expressed at the periphery than in the center (Fig. 6.6f-f″). Zic2 is known to affect neural tube closure in mice[54, 131] via a pathway involving BMP signaling [209].

Second, we find that E-cadherin and myosin exhibit counter-running gradients visible by 24 hpb (Fig. 6.6b-d): E-cadherin concentration is higher at the periphery of the culture, while myosin concentration is higher in the center. As shown in Fig. 6.6e, e', the pattern of E-cadherin at 24 hpb is similar to a readout of BMP signaling at 20 hbp.

These results suggest that both force-generating proteins and PRG orthologs are modulated along the BMP4 signaling gradient. To test this further, we eliminated BMP4 signaling by treating our neural tube cultures with 200 nM LDN 193189 concomitant to neural induction. The resulting structures completely lacked surface ectodermal differentiation as all cells differentiated into neural ectoderm. Additionally, no folding occurred (Fig. 6.6g). No gradients were observed in either E-cadherin(Fig. 6.6h,h') or PRG ortholog (Fig. 6.6i,i') expression profiles.

We note that there are two overt differences between the *in vitro* neural tube model and Drosophila GBE. (1) The geometries of the two structures are distinct [21, 75, 109] and (2) while myosin is recruited to the adherens-junctional belt in both systems (Fig. 6.6d), it is polarized in the *Drosophila* germ band and unpolarized in the synthetic neural tube. However, in both systems, there is a key axis – defined by BMP patterning – along which a contractility gradient needs to be established. Independent of the particular kinetic mechanism

(i.e. polarized vs unpolarized myosin), large-scale gradients of contractility along an axis are necessary for tissue flow. Therefore, while the specific parameters informing cytoskeletal recruitment and tissue flow will undoubtedly vary in different developmental contexts, general trends, such as the capacity of BMP to drive the generation of contractility gradients through cytoskeletal recruitment, may be conserved.

Together, these results demonstrate that core components of the BMP signaling cascade observed in *Drosophila* body axis elongation also inform ex-vivo human neural tube closure, suggesting that important trends in BMP signaling transduction are conserved across species in radically different geometric contexts.

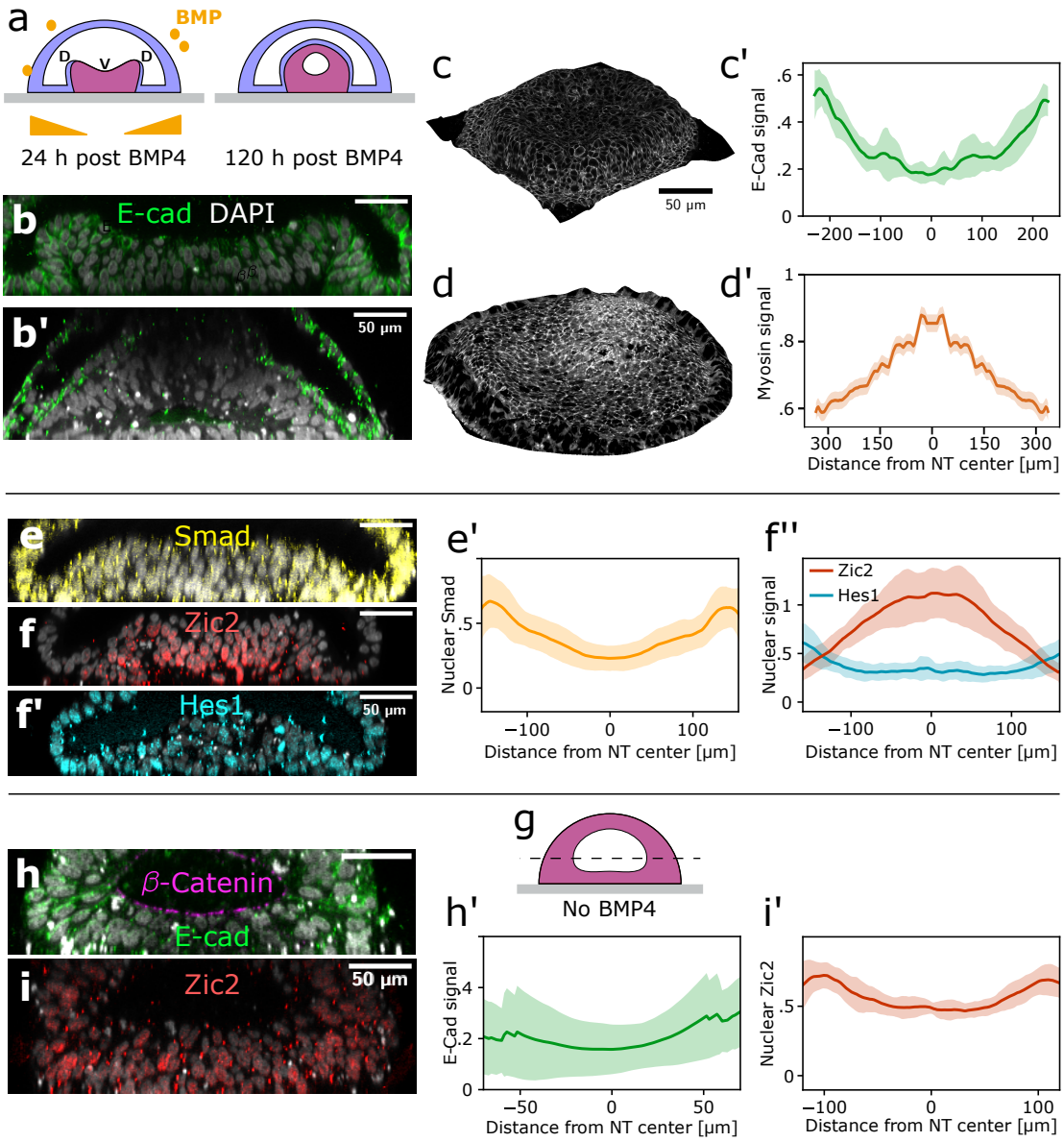

Figure 6.6: **DV patterning in human neural tube organoids (NTO).**

Figure 6.6: (cont.) (**a**) 3d organoids recapitulate folding morphogenesis of human neural tubes. Neural tube folding occurs within a primary lumen (blue dome). Blue tissue: surface ectoderm, red tissue: neural ectoderm. Cartoon shows neural tube at 24 and 120 hours post addition of BMP4 (hpb). BMP4 forms a radial gradient, with cells at the outer edge of the structure exposed to the highest levels of BMP4. Folding is complete by 120 hpb. (**b-b'**) Cross sections of NTO at 24 (**b**) and 120 (**b'**) hpb. (**c**) Apical surface of NTO at 24 hpb, showing apical E-cadherin. E-cadherin is graded, with low levels in the future neural ectoderm, which is not exposed to BMP4 signaling. (**c'**) Quantification of radial gradient in apical E-cadherin expression based on 20μm-sections around the apical surface ($N = 6$ NTOs). This and all quantifications below use a shaded area representing standard deviation and fluorescent intensity of the images normalized so that $0 = 5^{\text{th}}$ percentile, $1 = 95^{\text{th}}$ percentile. (**d**) Apical surface of NTO at 24 hpb, showing apical myosin. Myosin is graded opposite E-cadherin. (**d'**) Quantification of radial gradient in apical myosin expression based on 20μm-sections around the apical surface ($N = 2$ NTOs). (**e-e'**) NTO stained for the BMP4 target Smad1/5/9 as a readout for BMP4 signaling at 20 hpb. (**e'**) Quantification of the radial gradient in nuclear SMAD1/5/9 expression at 24 hpb ($N = 6$ NTOs). This quantification, as well as (**f"**, **i'**) are based on average nuclear signal across z-slices in NTO (surface ectoderm dome was excluded from analysis). (**f-f"**) PRG homologs Zic2 (**f**) and Hes1 (**f'**) display a radial gradient at 24 hpb (**f"**, $N = 6$ NTOs). (**g**) When BMP4 signaling is inhibited, NTO morphogenesis fails and the entire structure differentiates into neural ectoderm without folding. (**h-h'**) NTO treated with BMP4 inhibitor LDN ($200n$M) at time point equivalent to (**b**). E-cadherin levels on the apical surface are low. Quantification shows low E-cadherin levels and lack of DV gradient (**h'**). (**i-i'**) Addition of $200n$M LDN abolishes Zic2 gradient ($N = 6$ NTOs). Time point equivalent to 24 hpb.

6.8 Discussion

In this work, we have proposed a mechanism for neuroectoderm morphogenesis and argued that it is conserved across species. Figure 6.7 summarizes our mechanism and compares morphogenetic movements in the neural epithelium of *Drosophila* (left) and in a human neural tube model (right). In both cases, BMP signaling gradients are relayed into a global E-cadherin pattern via homeobox domain containing genes (PRGs or their mammalian homologs). E-cadherin in turn sets up a global pattern of force-generating cytoskeletal proteins, such as myosin, by modulating mechanical feedback loops involving tissue-scale stress or externally-imposed strains [75, 109]. This results in a coordinated morphogenetic flow during axis elongation in *Drosophila*. Whether human neural tube closure relies on similar

mechanical feedback loops for governing protein dynamics remains an exciting question [131]. A key feature of our mechanism is the tight integration between dorso-ventral (DV) and anterior-posterior (AP) patterning systems. This interplay between the DV and AP axes might be conserved beyond bilaterians. In *Nematostella* (sea anemones), for example, BMP signaling regulates a distinct set of homeobox genes [63] also associated with the mechanical processes sculpting the body [81].

The consensus *heterotypic juxtaposition model* of *Drosophila* neuroectoderm extension holds that direct genetic regulation locally modulates cell adhesion and anisotropic forces at cell boundaries. Our model presents an alternative picture in which the critical ingredient for shape evolution is the dorso-ventral symmetry-breaking of mechano-chemical feedback loops. Our results also support global modulation of cell adhesion, realized by the dorso-ventral E-cadherin gradient created by BMP signaling. Indeed, our results are in agreement with classical work [56] showing that the direction of *Drosophila* neuroectoderm extension can be reversed by inverting the BMP gradient via RNA injection. The recent discovery of Toll-like transmembrane receptors (TLRs) in a broad class of invertebrate systems [12] could indicate a molecular mechanism for this large-scale adherens junctions modulation. Indeed, we find a global DV modulation patterns in TLRs (Fig 5b). Under the control of PRGs, TLRs are isotropically localized to the membrane [107, 187] and strongly impact neural epithelium elongation in *Drosophila*. Yet how precisely TLRs regulate the cytoskeleton remains incompletely understood [142].

By applying machine learning techniques to live-imaged *Drosophila* development data, we found an autonomous dynamical system which quantitatively and causally reproduces the coupled dynamics of tissue flow and cytoskeletal proteins (myosin and E-cadherin). Decoupled from interpretative models, NNs are unable to distinguish correlation from causation. In order to obtain a causal model, we had to complement the machine learning methods with insights from physics and biology, and to put the predictions of the model to test using addi-

tional experiments. A further simplification allowed us to obtain a fully analytical minimal model that captures most qualitative features of neuroectoderm morphogenesis, but not in a quantitative way. However, our coarse-grained continuum models have several limitations: they do not describe the VF and show discrepancies with experimental strain-rate feedback measurements (Box 2 and SI). Resolving such effects may require more sophisticated models, such as graph NNs to process cell-level data.

We conclude by asking: why should the mechanisms by which morphogenesis unfolds be conserved? In biology, usual sources of universality are the conservation of genes by natural selection and convergent evolution. Morphogenesis has the peculiarity (shared, for instance, with locomotion) of strongly mixing biological and mechanical constraints – organisms have to abide to the laws of mechanics to get their shape. These constraints come together in the cytoskeleton, where genetics codes for conserved proteins that in turn execute the mechanical processes shaping tissues. For this reason, we hope that our approach may facilitate the systematic identification of the universal physical mechanisms (conserved across species) that biology uses to seed the shape of organisms.

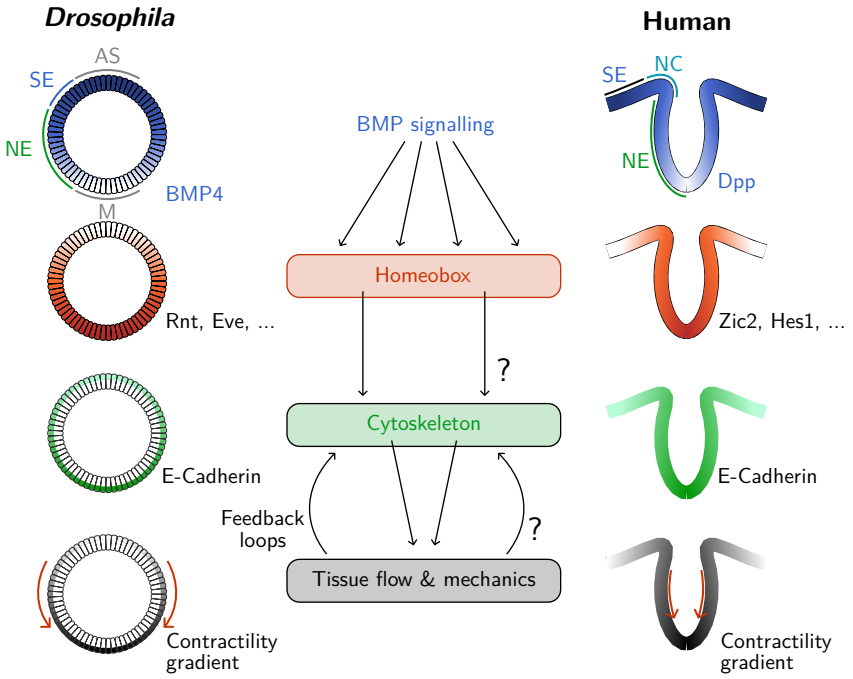

Figure 6.7: DV patterning from the fly to the human neural tube. Hypothesized morphogenetic parallels between *Drosophila* gastrulation and human neural tubes. SE: surface ectoderm, NE: neural ectoderm, M: mesoderm, AS: amnioserosa. Dpp/BMP signaling establishes a gradient in Homeobox transcription factors (PRGs in *Drosophila*), which lead to large scale cytoskeletal gradients, notably of E-cadherin and adherens-junctional proteins. These gradients lead to tissue flow, which can feed back to the cytoskeleton. Open questions in the neural tube organoids are the causal role of homeobox gradients, and the presence or absence of mechanical feedback loops.

6.9 Boxes and Supplementary Information

The main text of this manuscript refers to Boxes, Methods, and Supplementary Information. In this section, I include the SI sections related to the interpretation of neural network results, minimal active matter models of the embryo, and interpretation of the machine-learned equations used in Fig. 6.3 and in Box 2. Methods are included in the Appendix to this thesis and will be available online.

6.9.1 Box 1. Juxtaposition model of Drosophila germ-band extension

This box summarizes the current view of how genetic patterning controls GBE. Tissue flow during GBE can be quantitatively predicted based on measured myosin anisotropy [181]. Therefore, understanding GBE requires a understanding of how the myosin is controlled, in particular its anisotropy and its DV gradient.

Historical Context: PRG mutants have strong GBE phenotypes[86]. The orientation of PRG stripes and anisotropic myosin (which drives GBE) are superficially similar, especially at the beginning of GBE[109, 213]. These two observations have led to the hypothesis that anisotropic myosin is recruited preferentially to interfaces between cells that express different levels of PRGs. Several members of the Toll-family of transmembrane receptors (TLR) which are also expressed in stripes along the DV axis have been recognized as possible effectors of PRG mediated myosin recruitment. The TLRs are transcriptional targets of the PRGs, and TLR mutants phenocopy PRG mutants during GBE [141]. Additive effects on tissue flow are seen when multiple TLRs are disrupted [140, 141].

It remains unclear how the TLRs – which are isotropically distributed at cell edges – facilitate anisotropic myosin recruitment. The *heterotypic juxtaposition model* predicts that an instructive anisotropic cue comes from neighboring cells sitting on either side of a DV oriented TLR stripe recognizing and responding to heterotypic TLR expression at their shared interface. However, *heterotypic juxtaposition models* predict that knocking out all TLRs,

and knocking out all but one should have the same phenotype. This is inconsistent with the observation mentioned above that TLR mutations are additive [142]. Additionally, while the striped pattern of PRGs rotates over time due to tissue flow, the orientation of myosin anisotropy is only weakly deflected[109]. As a result, the orientations of PRG stripes and myosin uncouple quickly once tissue flow in the germband begins. Furthermore, individual cell-cell junctions change their myosin levels as they rotate into and out of alignment with the DV axis of the embryo [55], while the *heterotypic juxtaposition model* predicts that junctions should have fixed 'identities' during tissue flow.

An Alternative View: Instead of directly controlling myosin recruitment, gene expression patterns may modulate the strength of mechanical feedback loops. Mechanical feedback is know to influence myosin recruitment [57], and recent work using optogenetics has shown that junctional deformation leads to myosin recruitment, irrespective of junction orientation[75]. In this work, we present a framework that can reproduce the spatiotemporal profile of myosin, E-cadherin, and tissue flow as a result of mechanical feedback. Myosin anisotropy is produced from mechanical anisotropy, in two ways. Mechanical stress is anisotropic, due to cylindrical shape of the embryo (discussed in Ref. [109]). The axes of stress are illustrated as dashed lines in the schematic embryo above. Further, cell edges are anisotropically stretched (as measured by the strain rate) by the invagination of the VF. Both cues are parallel to the DV axis. Next, we explain the myosin DV gradient by the spatial modulation of the strength of mechanical feedback, with high activity of tension- and deformation-based feedback loops ventro-laterally, and low activity dorsally. In the schematic, this is illustrated by different shades of green. Indeed, optogenetic measurements show that strain-rate leads to weaker mysoin recruitment dorsally than ventrally[75], and analysis of cell geometry indicates tension-feedback is absent dorsally[21].

We hypothesize that modulation of mechanical feedback is carried out by one or more proteins that (1) are bound to adherens junction complexes (where myosin recruitment

occurs), (2) generate persistent patterns based on sufficiently slow turnover rates, (3) are dorso-ventrally graded. While E-cadherin fulfills these requirements, here it is treated as an agnostic representative of a broader class of DV graded cytoskeletal components including Bazooka/Par3 (SI online) and β-Catenin [125].

Our model reconciles several discordant observations. The direction of myosin anisotropy rotates little since tension-feedback aligns it with the static axis of geometric stress. New junctions that rotate into alignment with the direction of stress will recruit myosin. The onset of GBE coincides with VFI since VF strain recruits myosin. We predict that the phenotypes of PRGs and TLRs arise because these proteins play a role in mechanical feedback. Finally, we can cast our model into a set of equations that reproduce the onset of GBE, for E-cadherin, myosin, and tissue flow.

6.9.2 Box 2. Learning closed-loop hydrodynamics of embryo development

After considering minimal models, we use the SINDy method to learn rules from the data itself. In brief, this method constructs dynamic equations from a large *library* of possible terms, with the aim of fitting the data with as few terms as possible (see SI for details). Eq. 6.1 is a passive advection equation for E-cadherin. The cartoons below Eq. 6.2 are visual interpretations of each term. The LHS is constrained to the co-rotational derivative for the given velocity field. The first RHS term captures the propensity for myosin motors to detach from junctions. The second is tension recruitment due to local myosin-driven active stress and the third is an embryo-scale static stress. Finally, there is a mechanical feedback which recruits myosin to strained junctions.

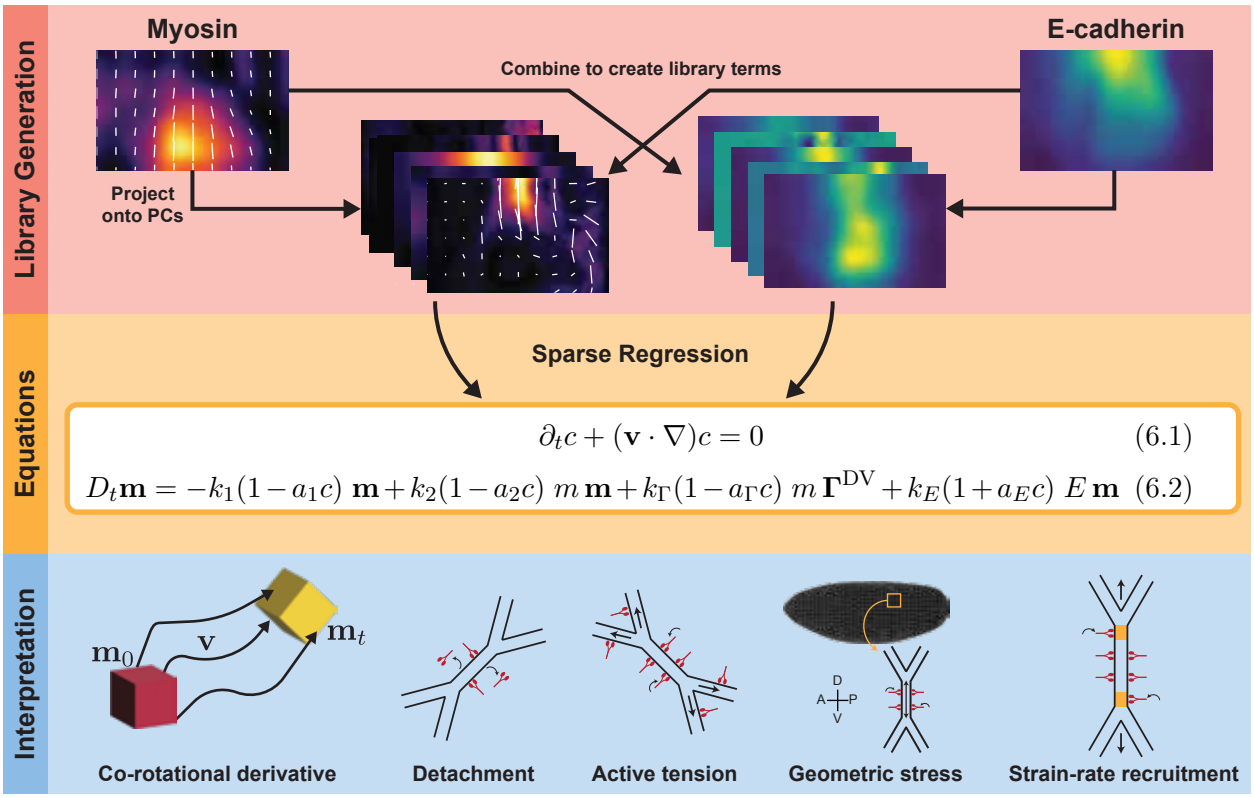

Figure 6.8: **Learning dynamical equations for cytoskeletal proteins**

Each mechanism is modulated by the cadherin pattern c. The $(1-c)$ prefactors establish a myosin DV-gradient opposite cadherin. The strain-rate recruitment term increases myosin concentration as a tissue is stretched. We note that the coarse-grained field $\mathbf{E}$ cannot distinguish contributions from two distinct biological processes, cell rearrangement and cell stretching [18]. Biologically, only the latter should recruit myosin. The learned strain rate term has a $(1+c)$ prefactor, which we interpret as distinguishing the strains from these two sources. Rearrangement dominates laterally while deformation primarily occurs dorsally, correlating with the cadherin pattern. Thus, the strain rate term in (6.2) plays a different role than the strain rate feedback measured in [75].

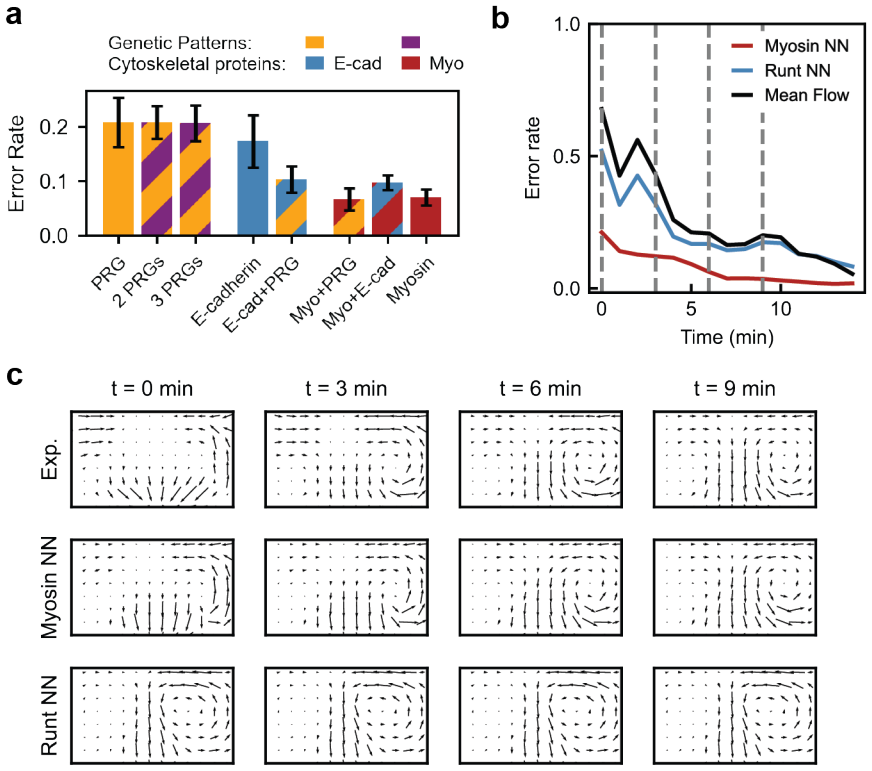

Figure 6.9: **Myosin is an optimal predictor of tissue flow.** (**a**) Average test error rate for NNs trained on PRGS, cytoskeletal proteins, and combinations. Networks trained using myosin as input outperformed all others. (**b**) Error-rate for single predicted trajectories for networks trained on myosin (red) and a PRG (runt, blue). Error rates were calculated against the ensemble-averaged flow trajectory. We also plot the error-rate for a "mean-field" prediction (black), which is given by the time- and embryo-averaged flow fields over the entire dataset. (**c**) Snapshots of experimental and predicted flow fields at three-minute intervals marked by grey lines in (**b**). While the myosin-trained network predicts a transition from a VF flow pattern to a vortical GBE configuration, the PRG-trained network predicts a constant flow field over time.

6.9.3 Identifying predictive proteins using neural networks

Using a morphodynamic atlas [125], we trained NNs to predict tissue flow from PRGs and cytoskeletal proteins. Myosin yielded the best predictions (6% error on unseen embryos), while E-cadherin (16%) and the live-imaged PRGs Runt (23%) or Even-Skipped (20%) performed worse. To train on multiple patterns, we supplemented each movie with an ensemble average of a second time-aligned field. Training on more than one PRG (including the static-imaged Fushi-Tarazu, Paired, and Sloppy-Paired) yielded no improvement, while supplementing E-cadherin with myosin or a PRG did improve results. Including other fields alongside myosin produced no improvement beyond the accuracy achieved by myosin alone (see Fig. 6.9a).

To contextualize the error rates reported in Fig. 6.9a, we used myosin-trained and PRG-trained NNs to predict 15-minute trajectories from initial conditions coinciding with VF onset ($t = 0$). In Fig. 6.9b-c, we plot the error rates over time and include sample snapshots of the predicted flow from each trajectory. The myosin-trained NN predicts a transition from a VF to a vortical GBE flow configuration. The PRG-trained NN gives a nearly-constant field which only slightly outperforms a constant "mean-field" prediction of the time- and ensemble-averaged flow field. Both achieve low error rates after the onset of GBE where the flow is quasi-static but have higher error rates during VF and GBE onset when the flow changes rapidly. Indeed, the network which predicts using myosin obtains a lower error rate over the entire trajectory (7.3%) than either the PRG (21%) or "mean-field" (26%) predictions.

Biological perspective

While myosin may be all a NN needs to generate flow, the same is not true for the embryo. Myosin binds to actin and produces stresses which propagate through the tissue across cell junctions. The NN results do not imply that other cytoskeletal proteins are unnecessary for axis elongation. Rather, the NN learned to use myosin as a proxy for this complex

microscopic machinery, and can infer the necessary information to predict tissue-scale flow from myosin alone.

Mechanical perspective

The NNs reaffirm the pivotal role of myosin in driving flow during axis elongation but they are not constrained by physics. The features they extract from myosin can be more complex than what might enter a hydrodynamic model via a local gradient expansion. Indeed, a recent study found that distilling a NN which predicted cell mechanics from proteins into a continuum mechanical model required adding new information – displacements and cell geometry – which the NN never saw [167]. Similarly, a hydrodynamic model for *Drosophila* might require more than just myosin.

6.9.4 Minimal active matter modeling

Myosin alone cannot reproduce GBE dynamics

Using minimal continuum models, we aim to reproduce (i) a long-lived vortex flow pattern and (ii) a long-lived myosin DV gradient (see Fig. 6.2c'-f'). As a starting point, we consider the following equations.

$$\mu\nabla^2\mathbf{v} - \nabla P + \alpha\nabla\cdot\mathbf{m} = 0 \tag{6.3}$$

$$D_t\mathbf{m} = A\ \mathbf{m} + B\ \mathbf{m}\,\mathrm{Tr}(m) + D\ \nabla^2\mathbf{m} \tag{6.4}$$

Eq. 6.3 is an incompressible ($\nabla\cdot\mathbf{v} = 0$) Stokes flow driven by myosin active stress $\nabla\cdot\mathbf{m}$ [181]. To account for myosin anisotropy on cell-cell junctions, we use the local orientation angle $\theta(\mathbf{r})$ to write a director $\mathbf{n} = (\cos\theta, \sin\theta)$ and analogue to a nematic order parameter $\mathbf{m} = m\,\mathbf{n}\mathbf{n}^T$ (m is the local myosin intensity). Eq. 6.4 describes myosin dynamics in terms of $\mathbf{m}$ where the left hand side is the co-rotational derivative $D_t\mathbf{m} = \partial_t\mathbf{m} + (\mathbf{v}\cdot\nabla)\mathbf{m} + (\mathbf{\Omega}\cdot\mathbf{m} - \mathbf{m}\cdot\mathbf{\Omega})$. The

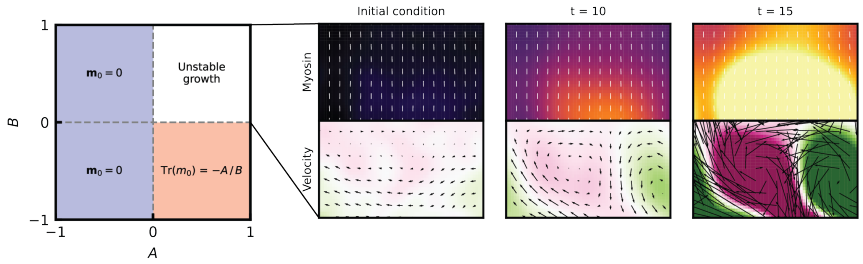

Figure 6.10: **Myosin alone cannot reproduce GBE dynamics.** *(Left)* Phase diagram for the system of equations Eq. 6.3-6.4. *(Right)* Prediction of pattern-forming in a myosin-only dynamical system enabled by a hydrodynamic instability.

right hand side contains minimal terms modeling myosin feedback (albeit somewhat non-biological, as myosin is bound in cells and cannot directly diffuse). This system of equations has two spatially-uniform fixed points $\mathbf{m}(\mathbf{r}) = \mathbf{m}_0$, with $\mathrm{Tr}(m_0) = 0, -A/B$ (with constant myosin, $\mathbf{v} \to 0$ and $D_t \to \partial_t$), and the latter exists only when $AB < 0$. Unless $A, B > 0$, the system will evolve to a constant stable fixed point and produce no flow. Otherwise, the system is unstable to long-wavelength perturbations with wavenumber $q^2 < A/D$. In this case, the model produces a plausible myosin and flow pattern, but due to the instability it will grow without bound. In our simulation, the myosin anisotropy direction is inherited from a polarized, nearly homogeneous, experimental initial condition.

Hydrodynamics with a control field

We now add a stationary spatially-patterned field $c(\mathbf{r})$ which cross-regulates the myosin.

$$\dot{m} + (\mathbf{v} \cdot \nabla)m = A\,[1 - k\,c(\mathbf{r})]\,m + Bm^2 \tag{6.5}$$

While myosin includes concentration and orientation dynamics, we focus on the former in Eq. 6.5. A non-uniform myosin pattern m can be maintained by $c(\mathbf{r}) = (A\,m + B\,m^2 - \mathbf{v} \cdot \nabla m)/k\,m$. This stationary configuration replaces the non-trivial fixed point (red) in the above phase diagram. As an example, we numerically computed a pattern $c^\star$ which would

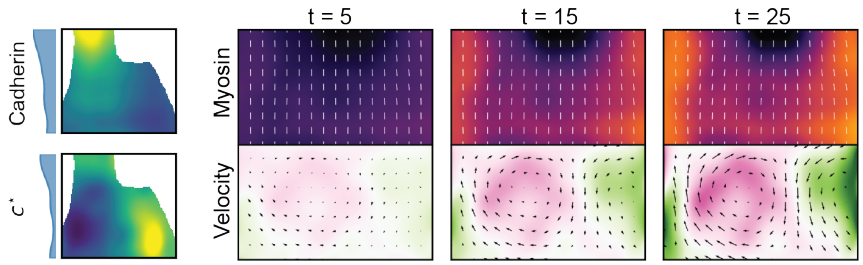

Figure 6.11: **Hydrodynamics with a control field.** *(Left)* Comparison of an experimental E-cadherin field with a field $c^\star$ obtained by setting the RHS of Eq. 6.5 to zero. Both fields are DV-graded with higher levels dorsally. *(Right)* Integration of Eq. 6.5 with the experimental E-cadherin field produces a myosin pattern which is longer lived than one generated via a hydrodynamic instability (Fig. 6.10).

maintain the GBE configuration at $t = 20$ minutes post-VF and found it is DV-graded similar to E-cadherin. We integrated the system using the *experimental* E-cadherin field, and observed it evolve to a patterned myosin field.

Orientational dynamics —myosin rotation by vorticity— prevents this pattern from being a true fixed point. Over longer timescales, the pattern disappears after GBE flow destroys the orientational order of myosin. In the subsequent sections, we show mechanical feedback slows this effect, enabling a longer-lived GBE state.

Myosin anisotropy modifies hydrodynamics

After focusing on the myosin concentration in previous sections, we now consider orientation dynamics.

$$\dot{\mathbf{m}} + (\mathbf{v}\cdot\nabla)\mathbf{m} + (\boldsymbol{\Omega}\cdot\mathbf{m} - \mathbf{m}\cdot\boldsymbol{\Omega}) = A\left[1 - kc(\mathbf{r})\right]\mathbf{m} + B\,\mathbf{m}\mathrm{Tr}(m) \tag{6.6}$$

Eq. 6.6 has a co-rotation term which describes how myosin orientation deflects due to gradients in flow. Note that we have dropped the unbiological diffusion term $D\nabla^2\mathbf{m}$. We separate concentration and orientation dynamics using the stream function ψ ($\mathbf{v} = \nabla\times\psi$)

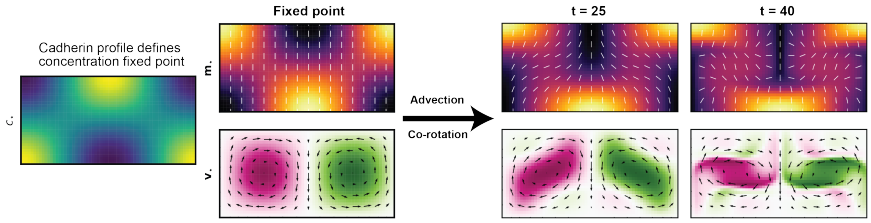

Figure 6.12: **Orientational dynamics destroy fixed point over time.** *(Left)* For a simplified control field profile $c^\star$, the system evolves to a quasi-fixed point $m^\star$ which qualitatively reproduces GBE-like flow. *(Right)* The nonlinear advection and co-rotation terms in Eq. 6.7 create sharp gradients in myosin orientation, eventually destroying the GBE flow pattern.

and orientation matrix $\mathbf{D}_\theta = \mathbf{n}\mathbf{n}^T$.

$$\underbrace{\Big[\dot{m} - \nabla\psi \times \nabla m\Big]}_{\text{Concentration dynamics}} \mathbf{D}_\theta + m \underbrace{\Big[\dot{\theta} - \nabla\psi \times \nabla\theta + \frac{1}{2}\nabla^2\psi\Big]}_{\text{Orientation dynamics}} \nabla_\theta \mathbf{D}_\theta = \Big[A(1-kc)m + Bm^2\Big]\mathbf{D}_\theta \tag{6.7}$$

This decomposition shows that the RHS creates a stationary concentration pattern via a fixed source $c(\mathbf{r})$, but the orientation will always vary, as argued in Ref. [109]. Simulations support this result. We integrate from an initial condition with a simple control field profile $c^\star = c_0 + c_1 \cos\left(\frac{2\pi x}{L_x} + \frac{2\pi y}{L_y}\right)$ and myosin $m = m_0,\ \theta = \pi/2$. The myosin concentration approaches the fixed point $m^\star$ defined by the control field such that the RHS in Eqs. 6.6-6.7 $= 0$ but this GBE-like configuration is short-lived. Vorticity creates sharp gradients in the myosin orientation, producing unrealistic and rapidly-diverging flow patterns.

Mechanical feedback slows destruction of orientational order

The flow caused by myosin rotates junctional orientations and ultimately destabilizes the GBE configuration in Eq. 6.6. However, experiments have shown that tissue flow is nearly static during GBE [125], suggesting that additional hydrodynamic terms may be necessary to agree with biology. Recent work has demonstrated that mechanical feedback can recruit myosin to cell-cell junctions [75]. Including a strain-coupling term proportional to $\{\mathbf{m}, \mathbf{E}\} =$

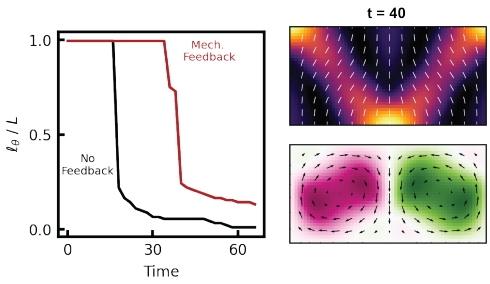

Figure 6.13: **Mechanical feedback term preserves orientational order.** *(Left)* Comparison of correlation lengths over time in simulations with (red) and without (black) mechanical feedback terms. The mechanical feedback preserves orientational ordering for nearly twice as long. *(Right)* Plot of simulated myosin and flow fields at $t = 40$ (A.U.) in a model with the mechanical feedback term $\{\mathbf{m}, \mathbf{E}\}$. Compared to Fig. 6.12, the modified model better preserves the vortical flow structure associated with germband-extension.

$\mathbf{m}\cdot\mathbf{E}+\mathbf{E}\cdot\mathbf{m}$ appears to slow the rate at which θ changes and preserves the GBE flow pattern nearly twice as long. To understand this, consider dynamics about $m = m^\star,\ \theta = \pi/2$. Here, the new term becomes $\{\mathbf{m}, \mathbf{E}\} = -m\mathbf{D}_\theta \nabla^2_{xy}\psi$, and the dynamics are given by Eq. 6.8.

$$\left[\dot{m} - \nabla\psi \times \nabla m + k_E\, m\nabla^2_{xy}\psi\right]\mathbf{D}_\theta + m\left[\dot{\theta} + \frac{1}{2}\nabla^2\psi\right]\nabla_\theta \mathbf{D}_\theta = 0 \tag{6.8}$$

At first, it is surprising that $\{\mathbf{m}, \mathbf{E}\}$ preserves orientational order, as it only suppresses $\dot{m}$ rather than $\dot{\theta}$. Consider instead $\ddot{\theta} = -\frac{1}{2}\nabla^2\dot{\psi}$, the rate at which $\dot{\theta}$ grows. Differentiating the Stokes equation yields

$$\nabla^4\dot{\psi} = -2\nabla^2\ddot{\theta} = \frac{\alpha}{\mu}\left[\nabla^2_{xy}\dot{m} - (\nabla^2_x - \nabla^2_y)m\dot{\theta}\right] \tag{6.9}$$

In this Poisson equation for $\ddot{\theta}$, mechanical feedback suppresses the source term containing $\dot{m}$. Rather than eliminate $\dot{\theta}$, mechanical feedback adjusts the concentration (and flow) dynamics to slow $\dot{\theta}$'s growth. This prevents *accelerating* destruction of orientational order (see plot of correlation length ℓ_θ over time) and allows the system to maintain GBE flow longer. In the subsequent section, we interpret the term $\{\mathbf{E}, \mathbf{m}\}$ in more detail.

6.9.5 Interpretation of of machine-learned continuum equations

Here we connect the learned continuum equations to previous models of myosin dynamics during GBE. The right hand sides of the SINDy equations Eqs.(6.1)-(6.2), are the usual convective derivatives, because myosin $\mathbf{m}$ and E-cadherin c are localized to cells that are transported by tissue flow velocity $\mathbf{v}$. As explained in the preceding section, the myosin pattern is strongly anisotropic, which is why we describe it using a nematic tensor, while E-cadherin, appears localized to cell-cell interfaces in an isotropic manner and can be described by a scalar density.

We note that the SINDy equations combine several terms, and are non-linear, which makes a complete analysis, in particular of the instabilities, challenging. Distinguishing strain-rate and stress (tension) effects, for example, is difficult as they are typically correlated. Moreover, SINDy solves an underdetermined regression problem with potentially many equivalent solutions, meaning interpreting individual terms in (6.2) must be done with caution. Our equation is not intended to model other, distinct morphogenetic events that occur simultaneously with axis elongation, such as ventral furrow and posterior midgut invagination. We do not attempt a complete analysis here and instead comment on the interpretation of the different terms in (6.2) and their connection to the prior literature.

E-cadherin dynamics

The left-hand side of the learned E-cadherin equation (6.1) is 0. This means E-cadherin is conserved. Indeed, biologically, the turnover of E-cadherin, which is a trans-membrane molecule, is very slow [36], and the E-cadherin pattern can be considered as conserved over a timescale of 20 minutes.

Myosin dynamics: turnover

We now turn to the myosin dynamics in equation (6.2). The term $-k_1\mathbf{m}$ corresponds to constant detachment of myosin from junctions. Such detachment is indeed measured experimentally using FRAP [109].

Myosin dynamics: tension feedback

The learned equations (6.1)-(6.2) contain additional RHS terms that model tension feedback, or recruitment of myosin on edges under higher tension. This mechanism was previously analyzed in Ref. [109], which considered only the local orientation $\theta \in [0, \pi]$ of the myosin nematic tensor $\mathbf{m}$, and not its magnitude. It was shown that the behavior of $\theta(t, \mathbf{r})$ can be understood as a combination of advection (i.e. tissue rotation), detachment, and recruitment by a static source. These mechanisms lead to following equation for the full myosin tensor:

$$\partial_t \mathbf{m} + (v_k \nabla_k)\mathbf{m} + [\mathbf{\Omega}, \mathbf{m}] = -k_1 \mathbf{m} + k \cdot (\mathbf{m} + \mathbf{\Gamma}^{\mathrm{DV}}) \tag{6.10}$$

of which Eqs. (6.1)-(6.2) are an extension. The static source $\mathbf{\Gamma}^{\mathrm{DV}}$ was hypothesized to be the surface stress that balances the internal turgor pressure in the embryo. Because the embryo is elongated, this stress is anisotropic, and because the embryo geometry does not change during tissue flow, it is static.

Ref. [21] analyzed GBE on a single-cell level and found that the dynamics of active tension T on single edges (computed using tension inference), was well described by excitable tension recruitment:

$$\partial_t T = kT^n, \quad n > 1 \tag{6.11}$$

Tension feedback has previously been found experimentally [57, 211]. The mesoscale tension

recruitment term $k_2\, m\, \mathbf{m}$ can be viewed as a coarse-graining of Eq. 6.11.

Myosin dynamics: Strain rate feedback

Ref. [75] has shown that junctions recruit myosin in response to deformation strain rate. The paper analyzes single junctions of length ℓ and myosin level m. Strain rate recruitment is modelled by :

$$\frac{\dot{m}}{m} = \alpha(\mathbf{r})\frac{\dot{\ell}}{\ell} \tag{6.12}$$

This type of feedback was first postulated theoretically as a way for cells to ensure convergence to mechanical equilibrium while maintaining tissue plasticity [134]. The feedback coefficient α does not depend on the junction orientation, but it does depend on the position along the DV axis. It was found experimentally that α was low dorsally and high ventrally. To interpret this strain-rate feedback in the context of Eqs. 6.1-6.2, we first coarse-grain Eq. 6.12 to determine how it might appear in a continuum model. Next, we examine a discrepancy between the feedback coefficient in Eq. 6.2 and the patterned mechanical feedback reported in [75].

Coarse-graining single-cell level strain rate feedback

We can connect the continuum equation terms involving the strain rate with this edge-level model. The anticommutator $\{E, m\}_{ij} = (E_{ik}m_{kj} + m_{ik}E_{kj})/2$ can be seen as an (inexact) coarse graining of Eq. 6.12. To see this, define the single-edge myosin tensor m^e_{ij}, computed from the junction orientation $\mathbf{n}$ and the myosin level m of a junction e: $m^e_{ij} = mn_in_j$. In presence of local strain rate E, the junction will become stretched depending on its relative

orientation to the strain axes:

$$\frac{\dot{\ell}}{\ell} = \alpha n_i E_{ij} n_j = \alpha \frac{m^e_{ij} E_{ji}}{m} \tag{6.13}$$

Second, if the orientation $\mathbf{n}$ is not perfectly aligned with the eigenvectors of E, the junction orientation will also be affected.

$$\partial_t \mathbf{n}_i = E_{ij} n_j - (n_j E_{jk} n_k) n_i \tag{6.14}$$

The second term here ensures that $\mathbf{n}$ remains a unit vector. Because of its non-linear nature, this term leads to subtleties when coarse graining. Ignoring these issues, however, we can combine the two preceding equations into tensorial form:

$$\partial_t m^e_{ij} = \alpha \{E, m^e\}_{ij} \tag{6.15}$$

Another way to derive Eq. 6.15 is to model the local angular distribution $m(\theta)$ of myosin intensity as a function of edge orientation. This distribution gets shifted by the strain rate E, and the resulting equations can be expanded in a Fourier basis in θ and repackaged into a tensor. However, similar subtleties make this coarse graining inexact.

Eq. 6.15 is the term that is considered in Eq. 6.8, and the strain rate term $\mathrm{Tr}[E]\mathbf{m}$ in Eq. 6.1 has a similar shape. Note that a commutator $\{E, m\}$ can also occur as part of the convective derivative. However, because the per-edge levels of myosin m transform as a density (myosin is diluted if an edge is stretched), such a convective term would have the opposite sign as strain rate feedback, which is a form of "anti-dilution".

Discrepancy of SINDy model with experiment due to different types of tissue strain

While a strain-rate term does appear in the SINDy equations, its dorso-ventral modulation behaves unexpectedly. Based on the experimental results explained above, we were expecting high levels of strain-rate ventro-laterally, and low levels above. However, the E-cadherin-dependent strain-rate coefficient in Eq. 6.2 is $(1 + a_E\, c)$ with $a_E > 0$, so that the opposite is true: in the SINDy model, the strain rate feedback is strongest dorsally.Furthermore, in the SINDy model, strain rate feedback does not significantly contribute to the establishment of the DV myosin gradient, which is opposite to the results of Ref. [75]. Instead, our learned equations generate a DV myosin gradient through excitable tension dynamics which are controlled by the E-cadherin field. To understand the discrepancy between these two pictures, we need to revisit how strain can appear in a tissue.

In the absence of cell division and death, a tissue can either deform by stretching individual cells, and hence junctions, or by cell rearrangement without changing cells shapes [18]. Ref. [75] established a feedback only in response to cell stretching. Cell rearrangement will not induce a similar feedback, because during cell rearrangements individual junctions are not stretched.

Immediately before *Drosophila* GBE, mesodermal cells along the ventral pole invaginate, forming the VF. This causes cells in the ventral lateral domain of the germband to stretch, driving a temporary tissue flow towards the ventral pole. Subsequently, flow is dominated by cell rearrangements [21]. Our coarse-grained flow field, which is computed from image data using PIV and further processed via smoothing and PCA, omits the cell-scale information needed to distinguish between these two modes of driving tissue flow — cell stretching and cell rearrangement.

This lack of distinction between rearrangement and cell shape provides a possible explanation for the discrepancy between [75] and our machine-learned strain rate term. The $(1 + a_E\, c)$

E-cadherin dependence learned by SINDy reduces feedback in the lateral regions, where strain-rate during GBE primarily comes from rearrangement [21]. Strain-rate feedback in the lateral region would kill the myosin field and stop flow, because there, strain rate E and myosin orientation are opposite. It strengthens feedback in the dorsal tissue, which deforms via both rearrangement and deformation [21]. The role of our machine-learned feedback, which appears to prioritize deformation and ignore rearrangement, is to preserve myosin at the dorsal pole at late times. We note that this also suppresses steep myosin gradients which would cause the tissue flow to continue to grow. Such an effect is consistent with our earlier analysis which uses minimal hydrodynamic models to show that mechanical feedback can help stabilize flow patterns during GBE.

CHAPTER 7

CONCLUSIONS AND OUTLOOK

In this thesis, we developed a set of tools for studying active and biological materials which combine machine learning, traditional physics, and biological insight. When applied to experimental case studies across biological scales, our methods unveiled predictive and interpretable models. These results present machine learning as a promising pathway for studying complex systems which evade traditional approaches to physical modeling. While our methods are data-driven, they are not automatic. In each case study, domain knowledge played a crucial role in driving our analysis. We refined and applied our tools while working in close collaboration with experimental groups. This was necessary in order to better understand the assumptions — implicit and explicit, physical and biological — inherent to our models.

Machine learning methods are capable of achieving excellent predictions of complex systems. To move beyond prediction and extract insight, it is important to use these powerful tools in direct pursuit of physical questions. In this thesis, we used machine learning to identify relevant variables for biophysical models, examine how experimental perturbations appear in active continuum theories, and test proposed mechanisms governing biological behavior. Identifying these questions enabled us to tailor our models and develop interpretable architectures. The power of deep learning tools makes it tempting to immediately train models on new datasets, and these initial results are often exciting. However, it can be difficult to push further without determining a suitable (and actionable) scientific problem.

To push data-driven physical modeling beyond the presented case studies, we need to expand our tools as well as our understanding of how they interact with the intricacies of each problem. Below, I outline some possible directions for future investigation.

- Geometry plays a crucial role in many biological systems including the cells and flies presented in this thesis. Our neural networks could treat the experimental data as images, but our subsequent physical analysis required us to add geometric terms. Adapt-

ing our methods to incorporate system geometry from the outset, with tools such as MeshCNN, could help streamline *post hoc* physical analysis.

- In our case studies, we were fortunate to collaborate with groups that collected large well-curated datasets, but this is not always the case. Because it is often costly to obtain high-quality data, it is crucial to develop methods which can learn from minimal examples. Introducing physical constraints is one path forward, while unsupervised learning can negate the time-consuming process of labeling large datasets. Beyond this, insights from few-shot learning may help adapt our physical modeling methods to difficult scenarios with limited data.

- Systems such as crowds or epithelial tissues blur the line between discrete and continuum. In such cases, tools from geometric deep learning such as graph neural networks could help shed light on the interagent interactions, while our existing methods could simultaneously identify a continuum description. Identifying correspondences between such complementary analyses could produce a more unified and informative model than either could achieve by itself. How to coarse-grain a “microscopic" graph neural network to a “continuum" CNN remains an open question.

APPENDIX A

METHODS

This appendix contains methods for the papers presented in this thesis. I have omitted experimental methods for brevity. Interested readers can find complete experimental details and supplementary information at the references provided at the start of each chapter.

A.1 Active-nematic hydrodynamics

Active nematohydrodynamics and lattice Boltzmann simulations

Simulation data for training and testing was generated using a hybrid Lattice Boltzmann method which has been used in prior studies of different types of active nematics [105, 178, 216, 218]. The symmetric and traceless tensorial order parameter of the nematic is defined as

$$\mathbf{Q} = S(\mathbf{nn} - \mathbf{I}/3) \tag{A.1}$$

with S being the scalar order parameter, $\mathbf{n}$ being the unit vector describing the local nematic orientation, and $\mathbf{I}$ being an identity tensor. The following governing equation of the nematic microstructure, namely Beris-Edwards equation (A.2) reads

$$(\partial_t + \mathbf{u} \cdot \nabla)\mathbf{Q} - \mathbf{S}(\mathbf{W}, \mathbf{Q}) = \Gamma \mathbf{H} \tag{A.2}$$

where $\mathbf{u}$ is the velocity vector, $\mathbf{W}$ is the velocity gradient $\nabla \mathbf{u}$, and Γ is related to the rotational viscosity γ_1 via $\Gamma = 2S_0^2/\gamma_1$ with S_0 the equilibrium scalar order parameter[44]. Here, the generalized advection term $\mathbf{S}(\mathbf{W}, \mathbf{Q})$ is defined as

$$\begin{aligned}\mathbf{S}(\mathbf{W},\mathbf{Q}) = &(\xi\mathbf{A}+\mathbf{\Omega})(\mathbf{Q}+\mathbf{I}/3)\\ &+(\mathbf{Q}+\mathbf{I}/3)(\xi\mathbf{A}-\mathbf{\Omega})\\ &-2\xi(\mathbf{Q}+\mathbf{I}/3))\,\mathrm{Tr}(\mathbf{Q}\mathbf{A})\end{aligned} \tag{A.3}$$

with $\mathbf{A} = (\mathbf{W}+\mathbf{W}^T)/2$ being the strain rate tensor, $\mathbf{\Omega} = (\mathbf{W}-\mathbf{W}^T)/2$ being the vorticity, and ξ being a flow-alignment parameter setting the Leslie angle. The molecular field $\mathbf{H}$ is a symmetric, traceless projection of the functional derivative of the free energy of the nematic. Its index form reads

$$H_{ij} = \frac{1}{2}\left(\frac{\delta F}{\delta Q_{ij}}+\frac{\delta F}{\delta Q_{ji}}\right) - \frac{\delta_{ij}}{3}\,\mathrm{Tr}\left(\frac{\delta F}{\delta Q_{kl}}\right) \tag{A.4}$$

in which the free energy functional is $F = \int_V f dV$. Its density f takes the following form: [155]

$$\begin{aligned}f = &\frac{A_0}{2}\left(1-\frac{U}{3}\right)Q_{ij}Q_{ij} - \frac{A_0 U}{3}Q_{ij}Q_{jk}Q_{ki}\\ &+\frac{A_0 U}{4}(Q_{ij}Q_{ij})^2 + \frac{1}{2}L\partial_k Q_{ij}\partial_k Q_{ij}\end{aligned} \tag{A.5}$$

where A_0, U are material constants and L is related to the Frank elastic constant under the one-constant-approximation. Eq.A.2 is solved using a finite difference method.

The hydrodynamic flow is governed by a momentum equation:

$$\begin{aligned}&\rho(\partial_t + u_j\partial_j)u_i = \partial_j\Pi_{ij}\\ &+\eta\partial_j\left[\partial_i u_j + \partial_j u_i + (1-3\partial_\rho P_0)\partial_\gamma u_\gamma \delta_{ij}\right]\end{aligned} \tag{A.6}$$

where ρ is density, η is the isotropic viscosity, and $P_0 = \rho T - f$ is the hydrostatic pressure with T being the temperature. The additional stress has two contributions, $\Pi_{ij} = \Pi^p_{ij} + \Pi^a_{ij}$,

where the first term is passive in its nature accounting for the anisotropy, and is defined as

$$\begin{aligned}\Pi^p_{ij} = &- P_0\delta_{ij} - \xi H_{ik}\left(Q_{kj} + \frac{1}{3}\delta_{kj}\right) \\ &- \xi\left(Q_{ik} + \frac{1}{3}\delta_{ik}\right)H_{kj} \\ &+ 2\xi\left(Q_{ij} + \frac{1}{3}\delta_{ij}\right)Q_{kl}H_{kl} \\ &- \partial_j Q_{kl}\frac{\delta F}{\delta\partial_i Q_{kl}} + Q_{ik}H_{kj} - H_{ik}Q_{kj}\end{aligned} \tag{A.7}$$

The active stress that drives the system out-of-equilibrium reads[1, 120]

$$\Pi^a_{ij} = -\alpha Q_{ij} \tag{A.8}$$

in which $\alpha > 0$ describes an extensile active nematic, as is the case for the experimental systems discussed in this manuscript. Eq.A.6 is solved simultaneously via a lattice Boltzmann method over a D3Q15 grid [43]. Additional details on this method can be found in [217].

Typical simulation parameters were $\Gamma = 0.13$, $\eta = 0.33$, $A = 0.1$, and $U = 3.5$, leading to $S_0 \sim 0.62$. For Figs. 4.1, 4.5, simulation were trained on $K = 0.075$, $\alpha \in [0, 0.05]$. The range of K for testing in Fig. 4.1F and G was $K \in [0.06, 0.09]$. For the multiparameter estimator used in Figs. 4.2, 4.3, 4.4, SI Appendix Fig. S6-S8, $K \in [0.06, 0.20]$ and $\alpha \in [0, 0.05]$. For the experimental prediction in 3D (SI Appendix Fig. S5), models were trained on $K = 0.1$, $\alpha \in [0, 0.09]$ as initial predictions indicated that the range of α first used for training was insufficient. For all the simulations prescribed with constant activity, the director field was recorded after the system reached a dynamical steady state.

Machine learning models

Neural networks are implemented in Python using the Pytorch library. Code for data preparation, network implementation, training, and evaluation is available online at `https:`

//github.com/jcolen/ML_ActiveNematics The machine learning training and testing workflows are summarized in SI Appendix Fig. S2.

Parameter estimation models

Parameter estimation networks contain between 1-2 convolutional layers with hyperbolic tangent activation functions, each of which is followed by a max pooling layer and a dropout layer with dropout probability of 0.15. The convolutional layers are further connected with a single recurrent layer implemented with a long short-term memory cell. Lastly, a dense layer with linear activation function is added to output the predicted parameters. An example architecture is shown in Fig. 4.1*E*. To make predictions on large director field images, the network randomly selects patches and ensemble averages the results into a final prediction. For networks using a recurrent layer, the model accepts a sequence of director field frames, rather than a single frame.

Three parameter estimation network architectures are used in this paper. The first, used to predict activity in two-dimensional nematics (Fig. 4.1*F*, SI Appendix Fig. S3), has a single convolutional layer with 32 filters of size 3×3, a single 2×2 max pooling layer, a recurrent layer implemented using a long short-term memory (LSTM) with hidden size 32, and a fully-connected layer with 32 neurons. This model accepts input sequences of 32×32 pixel image patches and was trained on a dataset of 6,000 director field frames separated by 10 simulation time steps, at 12 different levels of activity. The second model, used to predict activity in three-dimensional nematics (Fig. 4.1 *G*, SI Appendix Fig. S3 and S5) has a similar architecture, but with $5 \times 5 \times 5$ convolutional filters and no recurrent layer. This model accepts image volumes of size $32 \times 32 \times 32$ and was trained on a dataset of 6,000 director field configurations, separated by 100 time steps, at 12 levels of activity. The third model is used for simultaneous prediction of activity and elastic modulus in two-dimensional nematics (Fig. 4.2, 4.3, Fig. 4.4 and SI Appendix Fig. S6-S8). This network has the same structure

as the other two-dimensional model, but outputs two values and was trained on a dataset of 15,000 image frames, generated with 30 different combinations of activity and elastic modulus. The accuracy of this multi-parameter estimator is summarized in Fig. 4.2 *C-D*.

The networks used on two-dimensional active nematics were trained on continuous representations of the two-dimensional director field such as $\sin 2\theta$ or $\cos 2\theta$. The choice of continuous representation had no effect on the predictive accuracy of the model. The networks used on three-dimensional active nematics were trained on the traceless tensor $Q_{ij} = n_i n_j - 1/3$, which is the nematic order parameter without ordering magnitude coefficient.

Networks were trained for 100 epochs on director field configurations generated using Lattice-Boltzmann simulations. Each frame of training data was a 200×200 director field image with periodic boundary conditions. These datasets were augmented by applying random rotations, flips, and shifts during the training procedure. During each epoch, each input frame was randomly cropped to the predictive network input size. During training, we used an 80-20 training-validation split on the input dataset.

Forecasting model achitecture

The neural network for predicting time evolution, depicted in Fig. 4.5 *A*, is comprised of three parts: an encoder, recurrent layers, and a decoder. The encoder uses a sequence of convolutional layers to downsample input images into feature vectors. The decoder accepts feature vectors and uses convolutional layers to upsample those feature vectors back into images. A traditional autoencoder is comprised of these two layers only, and is an effective method of reducing data dimensionality. In our model, we insert the recurrent layers in between the encoder and decoder, so that dynamics can be computed on the encoded feature vectors. A benefit of this approach is that the dimensional reduction achieved by the encoder allows for smaller recurrent layers, reducing network complexity and improving performance.

The models reported in this paper accept director field images processed into the 2-

channel input $(\sin(2\theta), \cos(2\theta))$, where θ is the local orientation angle of the director field. The encoder contains two convolutional layers of stride-2 with 4 and 6 output channels, respectively. The decoder architecture mirrors that of the encoder, accepting a 6-channel feature vector and using two stride-2 transposed convolutional layers with 4 and 2 output channels, respectively. All convolutional layers use 4×4 kernels and are followed by batch normalization, which improves training performance, as well as hyperbolic tangent activation. The recurrent portion is a two-layer long short-term memory (LSTM) unit implemented as a residual network, or resnet, with a shortcut that directly connects input and output of the entire LSTM cell. Given a sequence of feature vectors, the resnet computes a small residual to be added to the input, rather than computing a full output feature vector from scratch. For input sequences with small time separations, the residual vector is sparse, which helps improve training performance and predictive accuracy, see SI Appendix Fig. S12 *A* and *B*. In particular, the residual architecture allowed the model to more effectively learn the activity-dependent temporal dynamics present in this system, see SI Appendix Fig. S12 *C*.

These models were trained using a two-step training procedure. First, the encoder, resnet, and decoder were trained together for 100 epochs. Next, the weights in the resnet were frozen and the encoder and decoder were trained together for 50 epochs. Training data was generated either using Lattice-Boltzmann simulations or directly from experiments. The Lattice-Boltzmann training data consisted of 200×200 director field images with periodic boundary conditions, separated by 6, 10, and 25 simulation time steps. Each simulation dataset contained 6,000 director field configurations at 12 levels of activity and was augmented during training using random flips, shifts, and crops. As before, we used an 80-20 training/validation split. Different models were trained on each dataset, with input image sizes of 48×48, 64×64, and 120×120. In the main text, we report results from the best performing of these models, which were trained on data with a frame separation of 10 time steps and use 48×48 input image size. All predictions are made using input sequences of 7

frames.

The experimental data consisted of 1,500 director field configurations extracted from microtubule-kinesin experiments at 5 different ATP concentrations (see Experimental Methods). We did not train on experiments with ATP concentrations of 10 μM and 18 μM as the time between snapshots was 5$\times$ longer than for the other ATP concentrations. Here, we also used an 80-20 training validation split and augmented data using random flips and crops. The results reported in this paper are for a model with an input size of 48×48.

Stitching forecasted predictions

While the models were trained to predict the evolution of director field patches, the error rates and characteristic length and time scales reported in Fig 4.5 are computed for full images in the testing dataset. To obtain the predicted configuration of the full director field, the model stitches together predictions made on overlapping subdomains of the image. Here, each pixel will appear in the prediction for multiple subdomains. The final prediction for each pixel is given by the weighted average of predictions from each subdomain. For a pixel located at $\mathbf{r} = (x, y)$, the weight given to the predicted value from a subdomain centered at $\mathbf{r_0} = (x_0, y_0)$ is the Gaussian weight with $\sigma = R$, the radius of the subdomain. Thus, more credence is given to domains in which the pixel is farther from the boundary. For all results reported in this paper, predictions were stitched together from 48×48 ($R = 24\sqrt{2}$) domains which overlapped by 8 pixels.

Sharpening algorithm for forecasting

To reduce any noise or artificial defects that arise from stitching together neural network predictions, we use a physically-motivated sharpening procedure. This algorithm only exploits the fact that the system is composed of fibers which tend to align while knowing nothing about the active forces present in the system.

The sharpening procedure minimizes the elastic free energy of a system composed of such fibers. Following [34, 41], we write the elastic free energy density as

$$\begin{aligned} f_d = &\frac{1}{2}K_1(\nabla \cdot \mathbf{n})^2 + \frac{1}{2}K_2(\mathbf{n} \cdot \nabla \times \mathbf{n})^2 \\ &+ \frac{1}{2}K_3(\mathbf{n} \times (\nabla \times \mathbf{n}))^2 \end{aligned} \tag{A.9}$$

Assuming a two dimensional system parameterized by an angle θ as $\mathbf{n} = (\cos\theta, \sin\theta)$, this becomes:

$$\begin{aligned} f_d = &\frac{1}{2}K_1(\sin\theta\partial_x\theta - \cos\theta\partial_y\theta)^2 \\ &+ \frac{1}{2}K_3(\cos\theta\partial_x\theta + \sin\theta\partial_y\theta)^2 \end{aligned} \tag{A.10}$$

In the one elastic constant approximation $K_1 = K_3 = K$, this reduces to

$$f_d = \frac{1}{2}K\left[(\partial_x\theta)^2 + (\partial_y\theta)^2\right] \tag{A.11}$$

The elastic free energy is minimized by setting $\frac{\delta f_d}{\delta\theta} = 0$, leading to the Laplace equation:

$$\nabla^2\theta = 0 \tag{A.12}$$

Thus, the elastic free energy minimization can be accomplished by applying relaxational dynamics to the director field. We implement this using a standard finite-differences approach, slightly modified to account for the nematic symmetry $\mathbf{n} = -\mathbf{n}$.

We first apply relaxational dynamics in a small box surrounding the topological defect positions from the previous director field frame. Because the winding number around the boundary of this box is fixed and nonzero, this sharpens the director field around each defect without risking removing the defect. Next, the director field is fixed inside the box and relaxational dynamics are applied in the defect-free region. This procedure is applied iteratively to sharpen the raw predicted image.

This procedure will work if the defect has not moved outside of the box between image frames. Assuming a timestep of τ, box size of R, and characteristic defect velocity v_d, this condition is satisfied if $v_d\tau < R$. We can approximate v_d using the relation provided by [68] for an isolated $+1/2$ defect, $v_d \approx \alpha l_d/\eta$. Here, we insert $l_d \approx \sqrt{K/\alpha}$, the mean defect spacing, as the radius of the defect-free region surrounding the $+1/2$ defect. Thus, the defect will remain in the box if $\sqrt{\alpha K}\tau/\eta < R$. The simulation data used in Fig. 4.5 had $K = 0.075$, $\eta = 0.33$, $\alpha_{\max} = 0.05$, and $\tau = 10$, leading to $R > 2$. The data reported in the main text was generated using a 5×5 box, corresponding to $R \in [2.5, 3.5]$. We chose the smallest possible value for R above this threshold, as it prevented the immediate annihilation of recently-nucleated defect pairs, which would otherwise be close enough to be enclosed by the same box. This would result in a net zero winding around the boundary, leading to their removal by the sharpening procedure.

Applications to experiment

Before being fed into parameter-prediction models, the actin-myosin images are adjusted in ImageJ to remove outliers using a median filter. The fixed orientation of the dye along the actin filaments and the polarization of the laser yields a polarized image whose intensity is proportional to $\cos^2\theta$, where θ is the director field orientation. From this we extract the continuous director representation $\cos 2\theta$ and downsample. For actin-myosin, we downsample by a factor of 6 to an effective pixel width of $a = 1$ μm, a convention that has been used in the past when comparing this Lattice-Boltzmann code with actin-myosin nematics [105, 215].

The microtubule-kinesin orientation fields were extracted from fluorescence images using the ImageJ plugin OrientationJ, which determines the structure tensor from intensity gradients. We use the extracted director field to obtain continuous director representations and then downsample them. Microtubule-kinesin data was downsampled to an effective pixel width of $a = 2.6$ μm, as the length scale of spatial variations in the raw data is larger. As a

machine learning model will be less reliable when applied to data significantly different from its training set, we aimed to downsample data such that the correlation length fell within the range of our simulation training dataset.

Determination of Lyapunov time

Active nematics are a nonlinear system characterized by a positive Lyapunov exponent. As a result, direct comparison of time-evolved director field configurations is not necessarily valid for long timescales. Pixel-wise accuracy should not be expected beyond the Lyapunov time, particularly as our predictive model lacked complete information about the system. While we report pixel-wise accuracy in the main text (Fig. 4.5B), knowledge of the chaotic dynamics of these systems is important to contextualize these results.

The grayed out region in Fig. 4.5B is bounded by the Lyapunov time as determined from Lattice-Boltzmann simulations. To find this timescale, we ran Lattice-Boltzmann simulations at different levels of activity and saved an intermediate state of the system. We then perturbed this state and continued the simulation. At each level of activity, we ran 10 trials from 10 separate intermediate states. Each simulation time-evolved at 200x200 grid, from which 100 points were randomly selected and tracked over time. By comparing these randomly selected pixels as a function of time, we extracted the Lyapunov exponent which was inverted to obtain the Lyapunov time. This quantity was dependent on activity, with more active systems exhibiting a shorter Lyapunov time. However, when we rescaled by the characteristic defect lifetime $\tau_d = \eta/\alpha$, we found that all values coalesced to approximately $\tau = 3.6\tau_{\mathrm{d}}$.

As infinitesimal pixel-wise changes would be eliminated by relaxational dynamics, we used a more global method of perturbing the intermediate state. First, we computed the singular value decomposition of the order parameter tensor field $Q_{ij}(\mathbf{r})$. We then fractionally changed 10 elements of the singular matrix by random amounts between -10% and +10%

and used the new matrix to reconstruct the perturbed order parameter field. This method of globally varying the intermediate state yielded non-vanishing pixel-wise deviations that showed exponentially growing behavior.

Characteristic length and time scales

Direct comparisons of the machine learning predicted director field to Lattice-Boltzmann simulations are unreliable beyond the Lyapunov time. To evaluate the validity of our predictions over longer time scales, we compare instead characteristic length and time scales of the machine learning predicted dynamical steady state. For a given order parameter configuration $Q_{ij}(\mathbf{r}, t)$, with i and j running over x, y, we define the spatial correlation function $C_s(\mathbf{r}, t)$ as

$$C_s(\mathbf{r}, t) = \frac{\langle Q_{ij}(\mathbf{r}, t) Q_{ij}(0, t) \rangle}{\langle Q_{ij}(0, t) Q_{ij}(0, t) \rangle} \tag{A.13}$$

and the time correlation function $C_t(\mathbf{r}, t)$

$$C_t(\mathbf{r}, t) = \frac{\langle Q_{ij}(\mathbf{r}, t) Q_{ij}(\mathbf{r}, 0) \rangle}{\langle Q_{ij}(\mathbf{r}, 0) Q_{ij}(\mathbf{r}, 0) \rangle} \tag{A.14}$$

where indices i and j are contracted following the Einstein summation convention. Using Eqs. (A.13) and (A.14), we define the director field correlation length ℓ_θ such that $C_s(\ell_\theta, t) = 1/2$ and the correlation time t_θ such that $C_t(r, t_\theta) = 1/2$.

In Fig. 4.5*C* and *D*, we compare the average values of ℓ_θ, t_θ as found in machine learning predicted director field frames to those of Lattice-Boltzmann simulations. Here, we iterate the predictive model to predict large (200×200) image frames over a long time ($t = 30\ \tau_{\mathrm{LC}}$) and compute the time-averaged correlation length and spatially-averaged correlation time. Correlation lengths and correlation times are computed using only the machine learning-generated image frames. In SI Appendix Fig. S10, we report the time-averaged mean-defect

spacing, defined as $\ell_{\mathrm{d}} = 1/\sqrt{n_{\mathrm{d}}}$, where n_{d} is the defect density.

Comparisons for multiparameter estimation

In Fig. 4.2, we compare the performance of our neural networks for multiparameter estimation to an approach adapted from [111]. The original method used correlation functions calculated from both the director and velocity fields to simultaneously fix multiple parameters in bacterial active nematics. We modified this approach to remove its dependence on the velocity field, which is not available in our active nematics experiments. To determine multiple parameters, we scan the parameter space and find the pair (α, K) that yields the best match for the observed nematic correlation length and correlation time. As the correlation length has been shown to depend on the ratio K/α [82] and the correlation time appears to depend on α but not K, these two quantities are sufficient to fix α and K.

Comparisons for Time Evolution

In Fig. 4.6, we compare the performance of our time evolution neural networks to Lattice-Boltzmann simulations. Our neural networks do not receive any information about the velocity field $\mathbf{v}(\mathbf{r}, t)$ or ordering strength S which are not directly observable in our active nematics experiments. For a direct comparison of the ability to predict the evolution of coupled chaotic fields when one is not easily measurable, we provide these Lattice Boltzmann simulations no initial information about the velocity field or ordering strength. In Fig. 4.6*A*, the machine learning predictions and the unknown-velocity Lattice Boltzmann simulations are compared to a baseline Lattice Boltzmann simulation with full information about all fields. In Fig. 4.6*B*, the machine learning predictions and unknown-velocity Lattice-Boltzmann simulations are compared to the observed director field evolution in microtubule-kinesin experiments. Here, the Lattice-Boltzmann simulations use parameters machine-learned from the same experiments (see Fig. 4.3*A*).

A.2 Adherent cell mechanics

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	

Table A.1: Summary 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.

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. Fig. 5.2). 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 $\vec{F}^{\text{cell}}_{\text{norm}} = \vec{F}^{\text{cell}}/\mu_F^{\text{day}}$ 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 in Table A.2, which we elaborate on here.

The dataset used to train the U-Net of Figs. 5.2,5.4,5.5 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 Figs. 5.2,5.4,5.5 are computed from the cells shown in Table A.2.

For the U-Nets trained on individual proteins shown in Fig. 5.3 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

Network	Figures	Trained on	Evaluated on
U-Net[1]	1, 4	(16 cells) D1-6, zyxin (first 30 frames of each cell)	D1: {2}, D2: {1,2}, D3:{2,3,5}, D4: {4,5} D5: {2,4}, D6: {1,3,4,5,6}
	3a-c	"	U2OS
	3d-f	"	MDCK
	3g-h	"	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}

Table A.2: Overview of the training and testing data used in this work. "D" stands for "Day", corresponding to datasets in Table A.1. For the protein experiments (U-Net[2]), separate networks were, for each protein input, trained on all but one cell which 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.

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 SI, 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 [101, 115], 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 [116]. 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 [161].

The channel structure of the U-Net is shown in Table. A.3. Most layers are composed of blocks with a ConvNext structure [116]. 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 SI Fig. 2. 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 [116].

The U-Net is trained with the Adam optimizer with weight decay ("AdamW", [117]) 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.

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

Table A.3: Channel structure for the U-Net used in Figs. 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 Fig. 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.

For the U-Net used in Figs. 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 Figs. 2, training data consists of 600 randomly sampled frames with a variable number of cells for training (see Table A.2). 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 Fig. 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 Fig. 4g,h,i 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 SI Fig. 16 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 Fig. 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 Supplementary 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 SI Fig. 17 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 [52], it has been extended to describe cells uniformly adhered to 2D substrates [136]. The free energy of the cell is

$$U = \frac{h}{2}\int dA\,(\sigma_{ij}^{\mathrm{el}} + \sigma^a \delta_{ij}) u_{ij} + \frac{1}{2}\int dA\, Y(x) u_i u_i \tag{A.15}$$

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 $\vec{u}(x)$:

$$h\partial_j \sigma_{ij}^{\mathrm{el}} = Y(x) u_i \quad \text{(in bulk)} \tag{A.16}$$

$$\sigma_{ij}^{\mathrm{el}} n_j = -\sigma^a n_i \quad \text{(on boundary)}. \tag{A.17}$$

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	

Table A.4: 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.

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

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\frac{\nu}{1-2\nu}\delta_{ij}u_{kk} + u_{ij}\right), \tag{A.18}$$

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 [126]. 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 Table A.4. 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 (see Table A.4).

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 [126]. 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 [194], 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 $\vec{F}_{\mathrm{NN}} = \nabla\phi + \xi\nabla\chi$, 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.

$$\mathcal{D}_\phi\phi = \rho_\phi\left[\zeta\right]; \quad \mathcal{D}_\xi\xi = \rho_\xi\left[\zeta\right]; \quad \mathcal{D}_\chi\chi = \rho_\chi\left[\zeta\right] \tag{A.19}$$

To predict traction forces subject to this hypothesis, we trained a GFNN to compute each Clebsch variable. Under (A.19), each term is the integral of a source and a Green's function.

$$\phi(\vec{x}) = \int d^2\vec{r}\, G_\phi(\vec{x}-\vec{r})\rho_\phi(\vec{r}) \tag{A.20}$$

$$\xi(\vec{x}) = \int d^2\vec{r}\, G_\xi(\vec{x}-\vec{r})\rho_\xi(\mathbf{r}) \tag{A.21}$$

$$\chi(\vec{x}) = \int d^2\vec{r}\, G_\chi(\vec{x}-\vec{r})\rho_\chi(\mathbf{r}) \tag{A.22}$$

For the network presented in Fig. 5.7, 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 Fig. 5.2-5.5. 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 Table A.5. 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 [96] 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 $\beta = 0.1$

$$\mathcal{L} = \sum (\vec{F} - \vec{F}_{GFNN})^2 + \beta \sum |G^{\alpha}(\vec{q})|^2 \tag{A.23}$$

After training, we found that the ϕ field contributed minimally to the predictions. In SI Fig. 3, we demonstrate that the $\nabla\phi$ 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 Fig. 5.7.

To demonstrate the performance of GFNN as well as the Clebsch decomposition approach, in SI Fig. 4 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 [26] in Python using the PySINDy library [92, 173]. 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

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

Table A.5: 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.

functions for the Green's functions.

$$G_i(r) \in \left\{ r^{-1}, \log(r), r, e^{-\ell_\xi}, e^{-\ell_\chi} \right\} \tag{A.24}$$

The last two terms are exponentially decaying functions whose length scales ℓ_α were fit to the machine-learned G_ξ, G_χ in Fig. 5.7d. 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.

$$\vec{F}(\vec{x}) = \underbrace{\sum_{ij}\left[w_{ij}^{\phi}\nabla\int d\vec{r}\,G_i\left(|\vec{x}-\vec{r}|\right)\rho_j\left(\vec{r}\right)\right]}_{\nabla\phi} + \underbrace{\sum_{ijk\ell}\left[w_{ijk\ell}^{\xi\chi}\left(\int d\vec{r}\,G_i\left(|\vec{x}-\vec{r}|\right)\rho_j\left(\vec{r}\right)\right)\nabla\left(\int d\vec{r}\,G_k\left(|\vec{x}-\vec{r}|\right)\rho_\ell\left(\vec{r}\right)\right)\right]}_{\xi\nabla\chi} \quad \text{(A.25)}$$

The weight grouping for $\xi\nabla\chi$ (A.25) is necessary as sparse regression is framed as a *linear* optimization problem. To obtain the weights $\vec{w}$, we used an elastic net objective.

$$\vec{w} = \text{argmin}\left[\langle\left(\vec{F}-\vec{F}(\vec{w})\right)^2\rangle + \alpha||\vec{w}||_1 + \frac{1}{2}\alpha||\vec{w}||^2\right] \quad \text{(A.26)}$$

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 Table A.1), 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 SI). Fig. 5.7g shows a sample prediction at $t = 40$ min, 10 minutes after the ROCK inhibitor was applied. Fig. 5.7h summarizes the predictions from this equation over the entire movie.

A different choice of α yields equations of different complexity. For Fig. 5.7i, 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 Fig.5.7i 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. For the model reported in Fig. 5.7, the learned equation was

$$\begin{aligned}\mathbf{F} = -0.7\big((\nabla\zeta)^2 \star e^{-r/\ell_1}\big)\vec{\nabla}\big(\zeta \star r\big) - 0.2\big(\nabla^2\zeta \star e^{-r/\ell_1}\big)\vec{\nabla}\big((\nabla\zeta)^2 \star \log|r|\big) + \\ 2.0\big(\zeta\nabla^2\zeta \star e^{-r/\ell_1}\big)\vec{\nabla}\Big[\quad\Big(\zeta - 0.4\zeta^2 - 0.1(\nabla\zeta)^2\Big) \star \log|r| - \\ 2.4\Big(\zeta - 0.5\zeta^2 + 0.6(\nabla\zeta)^2 - 0.1\zeta(\nabla^2\zeta) + 0.5\zeta\nabla^2\zeta\Big) \star r\Big] \quad \text{(A.27)}\end{aligned}$$

Here $\star$ denotes a convolution operation $a \star b = \int d\vec{r} a(|\vec{x} - \vec{r}|)b(\vec{r})$.

Outlier determination

One cell in the actin dataset was an outlier, and was excluded from the calculation of the mean in Fig. 5.3c. We tested for outliers using the Iglewicz-Hoaglin outlier test [76] and illustrate these results in SI Fig. 13.

Optimal predictors and histogram plots

Probability distributions shown in angle and magnitude plots (for example, Fig. 1e,f) are calculated by binning all pixels of all frames in the test set to calculate the number of joint occurrences of $|\vec{F}_{\text{exp}}|$ and $|\vec{F}_{\text{NN}}|$. The histogram is normalized to yield a probability and divided by the marginal distribution to calculate conditional probabilities. The "average" curves in Figs. 1e, 1f, 2b, 2d, 3b, 3c, 3e, 3f, 4b, 4d, 5e, 5f 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

$F_{\mathrm{NN}} \equiv F_N$ and $F_{\mathrm{exp}} \equiv F_E$ 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

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

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

$$F_N^*(X) = \arg\min_{F_N} L_X(F_N).$$

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 SI, 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.

A.3 Dynamics of morphogenesis

A.3.1 Data handling: image processing and ensemble time alignment

Image processing

To obtain coarse-grained data for training the deep neural networks and sparse dynamical equations, we used the morphodynamic atlas of *Drosophila* development [125]. This contains time series' of fluorescent images of a variety of biomarkers including cytoskeletal proteins and pair rule gene expression patterns. To convert the raw data into image suitable for our machine learning algorithms, we performed spatial downsampling smoothing which averaged over the microscopic, cell-level variations within the data. For the pair rule gene images, we performed a gaussian smoothing with $\sigma = 6\mu$m. For e-cadherin, which appears at cell junctions and therefore is more spatially inhomogeneous, we smoothed over a larger window of $\sigma = 15\mu$m. All fields were downsampled by a factor of 8.5, leaving 236×200 images with spatial dimension $452 \times 533\mu$m.

In order to characterize junctional myosin, we first performed cytosolic normalization which defines a myosin concentration relative to the average level in the cytoplasm [75, 109]. The cytoplasm intensity I_c is computed via a top-hat transform of the raw intensity I over a single-cell region. The cytosolic-normalized signal is then

$$I_n = \frac{I - I_c}{I_c} \tag{A.28}$$

To characterize junctional anisotropy, we used the radon transform technique introduced in [181]. The radon transform method integrates the signal intensity within small windows

along lines of varying orientation. This maps bright cell edges to peaks in a Radon plane whose coordinates define a position and orientation for the edge. We use the maximum in the Radon plane to define the local orientation angle θ and director $\mathbf{n} = (\cos\theta,\ \sin\theta)$. As these orientation angles are defined modulo 180°, we define myosin anisotropy using a nematic tensor

$$\mathbf{m} = m\,\mathbf{n}\mathbf{n}^T = m \begin{pmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{pmatrix} \tag{A.29}$$

As with the e-cadherin data, we applied gaussian smoothing to the myosin tensor over a window $\sigma = 15\mu$m and downsampled by a factor of 8.5. This produced a four-channel 236×200 image with spatial dimension $452 \times 533\mu$m, where each channel represented a component of the myosin tensor $\mathbf{m}$.

Ensemble time alignment

To compare behavior across different embryos, we used time-alignment procedures reported previously for the *Drosophila* atlas [125]. Briefly, the pair rule gene datasets were co-imaged with Runt and aligned by comparing the position and geometry of the runt stripe pattern at each time point. To align the live-imaged datasets of cytoskeletal proteins, we computed the spatially-averaged flow magnitude $\langle v \rangle$ and selected a constant offset for each embryo that minimized the difference between $\langle v \rangle(t)$ and that computed for a reference embryo. SI Fig. REF shows the aligned time series' for each embryo imaged for myosin and e-cadherin, demonstrating the reproducibility across embryos. The ensemble average across all live-imaged embryos is plotted in Fig. 6.2c.

A.3.2 Deep Neural Networks

Deep Neural Network Architecture

To forecast from initial conditions, we use an encoder-decoder architecture with a residual recurrent block at the latent bottleneck [38]. The network learns a set of latent variables from the inputs via a compressing convolutional encoder and computes the dynamics of those variables using a LSTM cell. Next the network transforms the latent variable trajectory into a spatially-resolved flow field via an upsampling convolutional decoder. At the latent bottleneck, we use the VAE reparameterization trick [97] to enforce a smooth latent space. This means that for each latent parameter, the encoder computes a mean and variance of a latent probability distribution. During training, the latent vector is sampled from this distribution rather than computed deterministically from the inputs. All neural networks were implemented using Pytorch [143].

We use ConvNext blocks [116] in our neural network architecture. A standard convolutional layer learns $k \times k$ kernels for each input and output channel, meaning a $N \to N$ convolutional layer has k^2N^2 tunable weights. In contrast, a ConvNext block learns spatial kernels that act only on each input channel via a grouped convolution, and then mixes each channel using 1×1 kernels. In the second step, it uses an inverse bottleneck structure with expansion factor B, which is equivalent to applying a fully-connected neural network with hidden size $B \times N$ at each point in space. A ConvNext block uses $k^2N + 2BN^2$ weights. In our network, which uses $k = 7$ and $B = 4$, each block requires approximately 5 times fewer weights than an equivalent standard convolutional layer.

Deep Neural Network Training

We train for 100 epochs using the Adam optimizer with a learning rate of $\lambda = 10^{-4}$. We use a learning rate scheduler which decreases the learning rate by a factor of 10 if the validation

Module	Layer	Data Size	Channels	Details
Encoder	Conv2d	236×200	$C \to 32$	k4×4, stride 4
	2×ConvNext	59×50	$32 \to 32$	
	Conv2d		$32 \to 64$	k2 × 2, stride 2
	2×ConvNext	29×25	$64 \to 64$	
	Conv2d		$64 \to 128$	k2 × 2, stride 2
	2×ConvNext	14×12	$128 \to 128$	
	Conv2d		$128 \to 256$	k2 × 2, stride 2
	2×ConvNext	7×6	$256 \to 256$	
Latent	Linear		$256 \times 7 \times 6 \to 2 \times 32$	outputs (μ, σ^2)
	VAE trick			$\mathbf{z} = \mathcal{N}(\mu, \sigma^2)$
Evolver	LSTM		$32 \to 32$	$\mathbf{z}_{t+1} = \mathbf{z} + LSTM(\mathbf{z})$ 2 layers, $h = 64$
Decoder	ConvTranspose2d	7×6	$32 \to 256$	k2 × 2, stride 2
	2×ConvNext	14×12	$256 \to 256$	
	ConvTranspose2d		$256 \to 128$	k2 × 2, stride 2
	2×ConvNext	29×25	$128 \to 128$	
	ConvTranspose2d		$128 \to 64$	k2 × 2, stride 2
	2×ConvNext	59×50	$64 \to 64$	
	ConvTranspose2d		$64 \to 32$	k4 × 4, stride 4
	2×ConvNext	236×200	$32 \to 32$	
	Conv2d		$32 \to 2$	k1 × 1

Table A.6: Forecasting NN architecture.

Layer	Details	Channels	Kernel	Groups
Conv2d	Mix space	$N \to N$	7x7	N
LayerNorm				
Conv2d	Mix channels	$N \to 4N$	1x1	1
GELU	Nonlinear activation			
Conv2d	Inverse bottleneck	$4N \to N$	1x1	1
Dropout	p=0.1			

Table A.7: ConvNext block architecture

loss has not improved for 10 epochs. Each training sample is a randomly selected trajectory of length $2 + \mathcal{N}(0, 3)$ minutes. The inputs are a set of biological markers and the outputs are the flow field $\mathbf{v}$. Each embryo was imaged for only one biological marker, so we use ensemble averages in order to train networks to predict from multiple inputs. Here, we generate ensembles by aggregating all marked embryos with a timepoint within 1 minute of the live-imaged initial condition, and averaging over 2 randomly selected embryos in that ensemble. During training, the network learns to minimize the following loss function

$$\mathcal{L} = \sum_t \mathrm{Res}\big[\mathbf{v}_{NN}(t), \mathbf{v}_{Exp}(t)\big] + \beta D_{KL}\big[\mathcal{N}(\mu, \sigma^2)||\mathcal{N}(0, 1)\big] \tag{A.30}$$

$$\mathrm{Res}(\mathbf{u}, \mathbf{v}) = \frac{\langle \mathbf{u}^2\rangle \mathbf{v}^2 + \mathbf{u}^2\langle \mathbf{v}^2\rangle - 2\mathbf{u}\cdot\mathbf{v}\sqrt{\langle \mathbf{u}^2\rangle\langle \mathbf{v}^2\rangle}}{2\langle \mathbf{u}^2\rangle\langle \mathbf{v}^2\rangle} \tag{A.31}$$

This joint loss function contains two terms. The first is a reconstruction loss based on the residual metric defined in [181]. The second term is a Kullback-Leibler divergence between the latent variables and a normal distribution. The coefficient β, which we set to 10^{-4} during training, determines the relative importance during training between the KL divergence term and the reconstruction residual.

A.3.3 Principal component analysis

To perform the decomposition into sparse componets, we constructed a pipeline using the scikit-learn Python library [145]. The pipeline accepted the processed image data and iteratively fit and transformed using the following steps

1. StandardShaper - a custom class to transform input data to size $[N_{samples}, C, H, W]$, where H, W are the image dimensions, $C = 4$ for tensor data, $C = 2$ for flow fields, and $C = 1$ for scalar fields.

2. LeftRightSymmetrize - a custom class which symmetrizes the image along the lateral direction, to prevent any left-right asymmetry from dominating the principal components

3. Masker - a custom class which applies a specified crop and mask to the data. In all cases we cropped 10 pixels (22 μm) from the boundary in each direction to eliminate distortion at anterior/posterior poles, as well as behavior at the ventral furrow. At later time points we masked out regions by the cephallic furrow and posterior midgut.

4. ModifiedStandardScaler - Unlike traditional standardization which centers the data by subtracting the mean, we subtract the ensemble average of all timepoints before ventral furrow invagination. Note that by not centering the data to have zero mean before decomposing, this pipeline is not *technically* PCA.

5. TruncatedSVD - Singular Value Decomposition truncated to $N = 16$ components.

A.3.4 SINDy fitting procedure and library construction

Library construction

For each field in the library ($\mathbf{m}$, $\mathbf{v}$, c), we first projected onto the first N PCA components, where $N =$ the number of components that explain 95% of variance. From these smoothed projections of the raw data, we computed the following libraries of scalar and tensor terms:

$$\mathcal{L}_c = \left\{1,\ c,\ \mathrm{Tr}(\mathbf{m}),\ v^2, \nabla \cdot \mathbf{v}\right\} \tag{A.32}$$

$$\mathcal{L}_m = \left\{\mathbf{m},\ \mathbf{m}\mathrm{Tr}(\mathbf{m}),\ \mathbf{\Omega},\ \mathbf{E},\ \mathbf{\Gamma}^{\mathrm{DV}}\right\} \tag{A.33}$$

Here $\mathbf{\Omega} = (\nabla\mathbf{v} - \nabla\mathbf{v}^T)/2$ is the vorticity tensor, and $\mathbf{E} = (\nabla\mathbf{v} + \nabla\mathbf{v}^T)/2$ is the strain rate tensor. We also included a static DV-aligned source $\mathbf{\Gamma}^{\mathrm{DV}}$ in the tensor library. We used these base sets to generate composite libraries using the following operations

1. Advective terms: $(\mathbf{v}\cdot\nabla)c$, $(\mathbf{v}\cdot\nabla)\mathbf{m}$
2. Symmetric couplings of tensors in $\mathcal{L}_m$. Note that this creates the co-rotation term $[\mathbf{\Omega},\mathbf{m}]$
3. Nonlinear couplings of scalars in $\mathcal{L}_c$
4. Outer product of tensor library and $\{1, c\}$ to include cadherin-modulated terms
5. Removal of any terms higher than first order in derivatives

We used these operations to generate a candidate library and stored it in an H5F filesystem for faster access during repeated training operations. We also stored the projections of each field onto the principal components. Finally, we included the time derivatives of the projected fields, computed using smoothed finite differences with a window size of 7 minutes, for use in the SINDy fitting procedure [26].

In the atlas, myosin and cadherin were measured on different embryos. To compute terms which couple $\mathbf{m}$ and c, we leveraged time-aligned ensemble averages enabled by the *Drosophila* atlas. Specifically, to compute a term like $c\,\mathbf{m}(t=t_0)$ for an embryo live-imaged for myosin, we used $c = \langle c(|t-t_0|<1)\rangle$. A similar replacement was used for terms like $c\,\mathrm{Tr}(\mathbf{m})$ for cadherin-imaged embryos.

SINDy fitting

To fit equations to the data using the above libraries, we collected each term and standardized them so that they had zero mean and unit variance over the entire dataset. We used the SINDy procedure and the Sequentially-Thresholded Least Squares (STLSQ) selection algorithm [26]. Rather than fit equations to the bare time derivative, we instead fit to the material derivative by adding the relevant terms to the left hand side before optimization. That is, we fit a myosin equation to the target $\dot{\mathbf{m}} + (\mathbf{v}\cdot\nabla)\mathbf{m} + [\mathbf{\Omega},\mathbf{m}]$, with the latter two

terms removed from the candidate library. Similarly, we fit a cadherin equation to the target $\dot{c} + (\mathbf{v} \cdot \nabla)c$ and removed the latter term from the candidate library.

STLSQ iteratively performs a regularized least squares minimization procedure. At each iteration, it removes any library terms whose standardized coefficients are below a thresholding parameter τ, which we set to $\tau = 0.01$. This process is repeated until the result converges. For the least squares fitting, we used L2 (ridge) regularization with a ridge parameter $\alpha = 10$.

The SINDy procedure is not deterministic, especially when the data is noisy and the library is large or contains several nearly equivalent terms. To account for this, we repeated the fitting procedure and kept terms that were present in an ensemble average over these runs. The terms present in (6.2) were found via an ensemble average over 10 trials, each trained on 10% of the dataset.

When performing an ensemble average of trials on a large library, the magnitudes of each coefficient may be suppressed by the presence of subleading terms that appeared stochastically in different models within the overall ensemble. For this reason, we repeated the fitting procedure using a library restricted to only the terms present in (6.2). We used the resulting equation, listed below, to integrate the model and generate the results in Fig. 6.3.

$$\partial_t \mathbf{m} + (\mathbf{v} \cdot \nabla)\mathbf{m} + [\Omega, \mathbf{m}] = -0.06(1 - 0.9\,c)\ \mathbf{m} + 0.56(1 - 0.7\,c)\ m\,\mathbf{m} + 0.49(1 + 0.6\,c)\ E\,\mathbf{m} + 0.05(1 - 0.8\,c)\ m\,\mathbf{\Gamma}^{\mathrm{DV}} \quad \text{(A.34)}$$

Hybrid integration of cytoskeletal and tissue dynamics

To integrate the cytoskeletal dynamics, we used a hybrid model which forecasts myosin and cadherin using (6.1)-(6.2) and predicts instantaneous tissue flow using a neural network (Fig. 6.3a). At each time step, we computed the field $\mathbf{v}$ from the current myosin field $\mathbf{m}$ using the neural network trained and reported in Fig. 6.1e-h. In order to translate the instantaneous flow rather than forecast, we omitted the recurrent LSTM layer (Evolver

module in Table A.6) and only applied the Encoder, Latent, and Decoder modules to the $\mathbf{m}$ field. Next, we computed the derivatives $\dot{\mathbf{m}}$, $\dot{c}$ according to the machine-learned dynamical equations. To eliminate artifacts, we performed two postprocessing steps. First, we zeroed each time derivative at the anterior and posterior poles (15 pixels or 33 μm from each pole) in order to eliminate artifacts due to distortion that may be amplified further when taking spatial gradients, following [181]. Second, we applied a gaussian filter ($\sigma = 3$ pixels $= 6\mu$m) to each field in order to smooth over upsampling artifacts introduced by the neural network. We performed a fourth order Runge-Kutta integration using the torchdiffeq library [35] for 30 minutes, beginning 10 minutes prior to VF and ending approximately 14 minutes after GBE onset. Following [181], we compare predictions using the normalized error metric (A.31).

While past work has proposed that a Stokes equation can determine tissue flow from the global myosin distribution [181], we have instead used a neural network to predict instantaneous flow. The reason for this is that [181] required adding a ventral cut to the 3D embryo mesh to model cell internalization during VF. This spontaneous change in geometry is a distinct morphogenetic event that is beyond the ability of our hydrodynamic models to predict. Instead, we used a NN which has learned to predict flow observed in experiment from myosin directly. It does not need to know about the system's changing geometry, but instead infers its effects from features of the myosin pattern. Using a NN kept the dynamical system closed and autonomous such that it could be integrated from initial conditions.

A.3.5 Analysis of experimental data for Figures 4-6

In this section, we describe the analysis of the experimental data shown in main text Figs. 4-6. We used slightly different image processing protocols than for the preceding Figures 1-3, where the main aim is to pre-process data for machine learning models.

Analysis of *Drosophila* e-cadherin and mysoin levels in DV and AP mutants

Pullback images were created from 3d lightsheet data using tissue cartography as described in Materials & Methods. To time-align movies from DV mutants, we used the formation of the cephalic furrow and the invagination of the posterior midgut, all of which are preserved in DV patterning mutants. For AP mutants (Fig. 5), we used PIV to time align. Since the overall level of flow is very low, the uncertainty due to this manual alignment was not a serious concern. Data from fixed embryos (e-cadherin) was aligned by comparing the morpholgy, in particular the characteristic folds, with the live movies.

We pre-processed the myosin and e-cadherin images as follows. Myosin was analysed using the cytosolic normalization filter described above. E-cadherin has very low levels in the cytosol, so the cytosolic normalization cannot be used. Instead, we normalized the contrast of all images by setting the 5^{th} and 95^{th} percentiles to 0 and 1, respectively. Note that therefore, comparisons of absolute levels of e-cadherin are not possible, in contrast to myosin. To measure myosin and e-cadherin gradients, we averaged the signal over the AP axis in a rectangular window that spanned the distance between the cephalic furrow and the beginning of the region that invaginates to form the posterior midgut. Tissue flow was analysed using PIV as described above.

We also analyzed the AP-variation of myosin and e-cadherin. As examined in Ref. [109], there are no systematic AP stripes in the myosin pattern. In SI Fig.

Analysis of *Drosophila* AP genetic patterning

To analyze the gradients of homeobox genes, we normalized the images by setting the 5^{th} and 95^{th} percentiles to 0 and 1, respectively. We then we averaged the signal over the AP axis in a curved window that spanned the distance between the 2^{nd} and 7^{th} Runt stripes. We adjusted the signal by the cell density, which is higher on the dorsal side of the embryo. We applied the same procedure to the TLRs Tartan and Toll6 shown in Fig. 5.

Analysis of neural tube organoids

All images of neural tube organoids (NTOs) were taken on a Leica SP6 confocal microscope as z-stacks. For this system, time alignment is trivial since the onset of the morphogenetic process is controlled by the addition of BMP4. In all quantifications we exclude the "dome", i.e. the single-cell layer surrounding the outer lumen, making the cutoff where the cells do not adhere to the glass substrate anymore. We performed three different kinds of image analyses on these datasets.

1. For myosin and e-cadherin as shown in Fig. 6, we used surface extraction and tissue cartography to extract the apical surface of the NTO. For myosin, we used these surfaces to quantify the radial gradient. In the case of myosin, we found fluorescent granules in the NTO lumen, close to the apical surface. We removed these granules computationally where possible, and excluded regions where the granules were too close to the apical surface to be separated from it from our quantifications.

2. For e-cadherin, we used a slightly different method to quantify the radial gradient which was more convenient for a large number of samples. We created radial slices, as shown in Fig. 6b, rotating the sample by 45 degree increments to obtain multiple perspectives, and then extracted the apical surface within such a 2d slice. We then quantified the e-cadherin levels along the extracted surface.

3. For dataset showing nuclear-localized markers (Hes1, Zic2, Smad 1/5/9), we used the DAPI channel to create a nuclear mask, excluding all non-nuclear signal. We then created radial slices as before, and computed the average signal, normalized by nuclei area, along the z-axis (vertical axis in Fig. 6).

For all three analyses, we standardized the images by setting the 5^{th} and 95^{th} percentiles to 0 and 1, respectively.

REFERENCES

[1] R. Aditi Simha and Sriram Ramaswamy. Hydrodynamic fluctuations and instabilities in ordered suspensions of self-propelled particles. *Physical review letters*, 89(5):058101, 2002.

[2] Aphrodite Ahmadi, M. C. Marchetti, and T. B. Liverpool. Hydrodynamics of isotropic and liquid crystalline active polymer solutions. *Physical Review E*, 74(6):061913, December 2006. Publisher: American Physical Society.

[3] Vasyl Alba, James E Carthew, Richard W Carthew, and Madhav Mani. Global constraints within the developmental program of the Drosophila wing. *eLife*, 10:e66750, June 2021. Publisher: eLife Sciences Publications, Ltd.

[4] Ricard Alert, Jaume Casademunt, and Jean-François Joanny. Active Turbulence. *Annual Review of Condensed Matter Physics*, 13(1):143–170, 2022. _eprint: https://doi.org/10.1146/annurev-conmatphys-082321-035957.

[5] Ricard Alert, Jean-François Joanny, and Jaume Casademunt. Universal scaling of active nematic turbulence. *Nature Physics*, 16:682–688, 2020.

[6] Ricard Alert and Xavier Trepat. Physical Models of Collective Cell Migration. *Annual Review of Condensed Matter Physics*, 11(1):77–101, 2020. _eprint: https://doi.org/10.1146/annurev-conmatphys-031218-013516.

[7] Nicolas Bain and Denis Bartolo. Dynamic response and hydrodynamics of polarized crowds. *Science*, 363(6422):46–49, 2019. Publisher: American Association for the Advancement of Science.

[8] Victor Bapst, Thomas Keck, A Grabska-Barwińska, Craig Donner, Ekin Dogus Cubuk, SS Schoenholz, Annette Obika, AWR Nelson, Trevor Back, Demis Hassabis, and Pushmeet Kohli. Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, 16(4):448–454, 2020. Publisher: Nature Publishing Group.

[9] Christopher Battle, Chase P Broedersz, Nikta Fakhri, Veikko F Geyer, Jonathon Howard, Christoph F Schmidt, and Fred C MacKintosh. Broken detailed balance at mesoscopic scales in active biological systems. *Science*, 352(6285):604–607, April 2016.

[10] Danny Ben-Zvi and Naama Barkai. Scaling of morphogen gradients by an expansion-repression integral feedback control. *Proceedings of the National Academy of Sciences*, 107(15):6924–6929, March 2010. Publisher: Proceedings of the National Academy of Sciences.

[11] Danny Ben-Zvi, Ben-Zion Shilo, and Naama Barkai. Scaling of morphogen gradients. *Current Opinion in Genetics and Development*, 21(6):704–710, December 2011. Publisher: Elsevier BV.

[12] Matthew A. Benton, Matthias Pechmann, Nadine Frey, Dominik Stappert, Kai H. Conrads, Yen-Ta Chen, Evangelia Stamataki, Anastasios Pavlopoulos, and Siegfried Roth. Toll Genes Have an Ancestral Role in Axis Elongation. *Current Biology*, 26(12):1609–1615, 2016.

[13] Antony N Beris and Brian J Edwards. *Thermodynamics of flowing systems: with internal microstructure*. Oxford University Press on Demand, 1994.

[14] Ethan Bier. Evolution of Development: Diversified Dorsoventral Patterning. *Current Biology*, 21(15):R591–R594, 2011.

[15] Ethan Bier and Edward M. De Robertis. BMP gradients: A paradigm for morphogen-mediated developmental patterning. *Science*, 348(6242):aaa5838, June 2015. Publisher: American Association for the Advancement of Science.

[16] Christopher Bishop. Mixture Density Networks, February 1994.

[17] C. Blanch-Mercader, V. Yashunsky, S. Garcia, G. Duclos, L. Giomi, and P. Silberzan. Turbulent Dynamics of Epithelial Cell Cultures. *Phys. Rev. Lett.*, 120(20):208101, May 2018. Publisher: American Physical Society.

[18] Guy B Blanchard, Alexandre J Kabla, Nora L Schultz, Lucy C Butler, Benedicte Sanson, Nicole Gorfinkiel, L Mahadevan, and Richard J Adams. Tissue Tectonics: Morphogenetic Strain Rates, Cell Shape Change and Intercalation. *Nature Methods*, 6(6):458–464, 2009.

[19] Laurent Blanchoin, Rajaa Boujemaa-Paterski, Cecile Sykes, and Julie Plastino. Actin dynamics, architecture, and mechanics in cell motility. *Physiol. Rev.*, 94(1):235–263, January 2014.

[20] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7327–7347, November 2022. arXiv:2103.04922 [cs, stat].

[21] Fridtjof Brauns, Nikolas H. Claussen, Eric F. Wieschaus, and Boris I. Shraiman. Epithelial flow by controlled transformation of internal force-balance geometry. *bioRxiv*, 2023. Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2023/05/31/2023.05.30.542935.full.pdf.

[22] Antoine Bricard, Jean-Baptiste Caussin, Nicolas Desreumaux, Olivier Dauchot, and Denis Bartolo. Emergence of macroscopic directed motion in populations of motile colloids. *Nature*, 503(7474):95–98, 2013. Publisher: Nature Publishing Group.

[23] George Britton, Idse Heemskerk, Rachel Hodge, Amina A. Qutub, and Aryeh Warmflash. A Novel Self-Organizing Embryonic Stem Cell System Reveals Signaling Logic Underlying the Patterning of Human Ectoderm. *Development*, page dev.179093, 2019.

[24] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, May 2021. arXiv:2104.13478 [cs, stat].

[25] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020.

[26] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113.

[27] David B. Brückner, Nicolas Arlt, Alexandra Fink, Pierre Ronceray, Joachim O. Rädler, and Chase P. Broedersz. Learning the dynamics of cell–cell interactions in confined cell migration. *Proceedings of the National Academy of Sciences*, 118(7):e2016602118, February 2021. Publisher: Proceedings of the National Academy of Sciences.

[28] David B. Brückner, Matthew Schmitt, Alexandra Fink, Georg Ladurner, Johannes Flommersfeld, Nicolas Arlt, Edouard Hannezo, Joachim O. Rädler, and Chase P. Broedersz. Geometry Adaptation of Protrusion and Polarity Dynamics in Confined Cell Migration. *Phys. Rev. X*, 12(3):031041, September 2022. Publisher: American Physical Society.

[29] Keith Burridge and Christophe Guilluy. Focal Adhesions, Stress Fibers and Mechanical Tension. *Exp. Cell Res.*, October 2015.

[30] Xuan Cao, Yuan Lin, Tristian P Driscoll, Janusz Franco-Barraza, Edna Cukierman, Robert L Mauck, and Vivek B Shenoy. A Chemomechanical Model of Matrix and Nuclear Rigidity Regulation of Focal Adhesion Size. *Biophys. J.*, 109(9):1807–1817, November 2015.

[31] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, December 2019. arXiv:1903.10563 [astro-ph, physics:cond-mat, physics:hep-th, physics:physics, physics:quant-ph].

[32] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.

[33] Michael E. Cates. Active Field Theories, April 2019. arXiv:1904.01330 [cond-mat].

[34] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, 1995. Publication Title: Principles of Condensed Matter Physics.

[35] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural Ordinary Differential Equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman,

N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[36] Yi-Jiun Chen, Juan Huang, Lynn Huang, Erin Austin, and Yang Hong. Phosphorylation Potential of Drosophila E-Cadherin Intracellular Domain Is Essential for Development and Regulating Adherens Junction Biosynthetic Dynamics. *Development*, page dev.141598, 2017.

[37] Frank Cichos, Kristian Gustavsson, Bernhard Mehlig, and Giovanni Volpe. Machine learning for active matter. *Nature Machine Intelligence*, 2(2):94–103, February 2020. ISBN: 2522-5839.

[38] Jonathan Colen, Ming Han, Rui Zhang, Steven A. Redford, Linnea M. Lemma, Link Morgan, Paul V. Ruijgrok, Raymond Adkins, Zev Bryant, Zvonimir Dogic, Margaret L. Gardel, Juan J. de Pablo, and Vincenzo Vitelli. Machine Learning Active-Nematic Hydrodynamics. *Proceedings of the National Academy of Sciences*, 118(10):e2016708118, 2021.

[39] Katherine Copenhagen, Ricard Alert, Ned S. Wingreen, and Joshua W. Shaevitz. Topological defects promote layer formation in Myxococcus xanthus colonies. *Nature Physics*, 17:211–215, 2021.

[40] E. D. Cubuk, R. J. S. Ivancic, S. S. Schoenholz, D. J. Strickland, A. Basu, Z. S. Davidson, J. Fontaine, J. L. Hor, Y.-R. Huang, Y. Jiang, N. C. Keim, K. D. Koshigan, J. A. Lefever, T. Liu, X.-G. Ma, D. J. Magagnosc, E. Morrow, C. P. Ortiz, J. M. Rieser, A. Shavit, T. Still, Y. Xu, Y. Zhang, K. N. Nordstrom, P. E. Arratia, R. W. Carpick, D. J. Durian, Z. Fakhraai, D. J. Jerolmack, Daeyeon Lee, Ju Li, R. Riggleman, K. T. Turner, A. G. Yodh, D. S. Gianola, and Andrea J. Liu. Structure-property relationships from universal signatures of plasticity in disordered solids. *Science*, 358(6366):1033–1037, 2017. Publisher: American Association for the Advancement of Science.

[41] P. G. de Gennes and J. Prost. *The Physics of Liquid Crystals*. Oxford University Press, 1995. ISSN: 0031-9228.

[42] Stephen J DeCamp, Gabriel S Redner, Aparna Baskaran, Michael F Hagan, and Zvonimir Dogic. Orientational order of motile defects in active nematics. *Nature Materials*, 14(11):1110–1115, 2015. Publisher: Nature Publishing Group.

[43] C. Denniston, D. Marenduzzo, E. Orlandini, and J. M. Yeomans. Lattice Boltzmann algorithm for three-dimensional liquid-crystal hydrodynamics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 362(1821):1745–1754, 2004.

[44] Colin Denniston, Enzo Orlandini, and J. M. Yeomans. Lattice Boltzmann simulations of liquid crystal hydrodynamics. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 63(5):1–10, 2001.

[45] John Devany, Daniel M. Sussman, Takaki Yamamoto, M. Lisa Manning, and Margaret L. Gardel. Cell cycle-dependent active stress drives epithelia remodeling. *Proceedings of the National Academy of Sciences*, 118(10):e1917853118, 2021. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1917853118.

[46] Amin Doostmohammadi, Jordi Ignés-Mullol, Julia M. Yeomans, and Francesc Sagués. Active nematics. *Nature Communications*, 9(1), 2018.

[47] Julien O. Dubuis, Gašper Tkačik, Eric F. Wieschaus, Thomas Gregor, and William Bialek. Positional information, in bits. *Proceedings of the National Academy of Sciences*, 110(41):16301–16308, October 2013. Publisher: Proceedings of the National Academy of Sciences.

[48] G. Duclos, C. Blanch-Mercader, V. Yashunsky, G. Salbreux, J. F. Joanny, J. Prost, and P. Silberzan. Spontaneous shear flow in confined cellular nematics. *Nature Physics*, 14(7):728–732, 2018. Publisher: Springer US.

[49] Guillaume Duclos, Raymond Adkins, Debarghya Banerjee, Matthew S.E. Peterson, Minu Varghese, Itamar Kolvin, Arvind Baskaran, Robert A. Pelcovits, Thomas R. Powers, Aparna Baskaran, Federico Toschi, Michael F. Hagan, Sebastian J. Streichan, Vincenzo Vitelli, Daniel A. Beller, and Zvonimir Dogic. Topological structure and dynamics of three-dimensional active nematics. *Science*, 367(6482):1120–1124, 2020.

[50] Guillaume Duclos, Christoph Erlenkämper, Jean-François Joanny, and Pascal Silberzan. Topological defects in confined populations of spindle-shaped cells. *Nature Physics*, 13(1):58–62, 2017. Publisher: Nature Publishing Group.

[51] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence Modeling in the Age of Data. *Annual Review of Fluid Mechanics*, 51(1):357–377, 2019.

[52] Carina M. Edwards and Ulrich S. Schwarz. Force Localization in Contracting Cell Layers. *Phys. Rev. Lett.*, 107(12):128101, September 2011. Publisher: American Physical Society.

[53] Perry W Ellis, Daniel JG Pearce, Ya-Wen Chang, Guillermo Goldsztein, Luca Giomi, and Alberto Fernandez-Nieves. Curvature-induced defect unbinding and dynamics in active nematic toroids. *Nature Physics*, 14(1):85–90, 2018. Publisher: Nature Publishing Group.

[54] Sarah Escuin, Saba Rose Raza-Knight, Dawn Savery, Carles Gaston-Massuet, Gabriel L. Galea, Nicholas D. E. Greene, and Andrew J. Copp. Dual Mechanism Underlying Failure of Neural Tube Closure in the \emphZic2 Mutant Mouse. *Disease Models & Mechanisms*, 16(3):dmm049858, 2023.

[55] Dene L. Farrell, Ori Weitz, Marcelo O. Magnasco, and Jennifer A. Zallen. SEGGA: A Toolset for Rapid Automated Analysis of Epithelial Cell Polarity and Dynamics. *Development*, 144(9):1725–1734, 2017.

[56] Edwin L. Ferguson and Kathryn V. Anderson. Decapentaplegic Acts as a Morphogen to Organize Dorsal-Ventral Pattern in the Drosophila Embryo. *Cell*, 71(3):451–461, 1992.

[57] Rodrigo Fernandez-Gonzalez, Sérgio de Matos Simoes, Jens-Christian Röper, Suzanne Eaton, and Jennifer A Zallen. Myosin II Dynamics Are Regulated by Tension in Intercalating Cells. *Developmental Cell*, 17(5):736–743, 2009.

[58] Daniel A Fletcher and R Dyche Mullins. Cell mechanics and the cytoskeleton. *Nature*, 463(7280):485–492, January 2010.

[59] Michel Fruchart, Colin Scheibner, and Vincenzo Vitelli. Odd Viscosity and Odd Elasticity. *Annual Review of Condensed Matter Physics*, 14(1):471–510, 2023. _eprint: https://doi.org/10.1146/annurev-conmatphys-040821-125506.

[60] Margaret L Gardel, Benedikt Sabass, Lin Ji, Gaudenz Danuser, Ulrich S Schwarz, and Clare M Waterman. Traction stress in focal adhesions correlates biphasically with actin retrograde flow speed. *J. Cell Biol.*, 183(6):999–1005, December 2008.

[61] Jacques Gautrais, Francesco Ginelli, Richard Fournier, Stéphane Blanco, Marc Soria, Hugues Chaté, and Guy Theraulaz. Deciphering Interactions in Moving Animal Groups. *PLoS Computational Biology*, 8(9), 2012.

[62] B. Geiger, A. Bershadsky, R. Pankov, and K. M. Yamada. Transmembrane crosstalk between the extracellular matrix–cytoskeleton crosstalk. *Nature Reviews. Molecular Cell Biology*, 2(11):793–805, November 2001.

[63] Grigory Genikhovich, Patrick Fried, M. Mandela Prünster, Johannes B. Schinko, Anna F. Gilles, David Fredman, Karin Meier, Dagmar Iber, and Ulrich Technau. Axis Patterning by BMPs: Cnidarian Network Reveals Evolutionary Constraints. *Cell Reports*, 10(10):1646–1654, 2015.

[64] Delphine Geyer, Alexandre Morin, and Denis Bartolo. Sounds and hydrodynamics of polar active fluids. *Nature Materials*, 17(9):789–793, 2018. Publisher: Springer US.

[65] Darren Gilmour, Martina Rembold, and Maria Leptin. From morphogen to morphogenesis and back. *Nature*, 541(7637):311–320, January 2017. Publisher: Springer Science and Business Media LLC.

[66] Luca Giomi. Geometry and topology of Turbulence in active nematics. *Physical Review X*, 5(3):1–11, 2015.

[67] Luca Giomi, Mark J. Bowick, Xu Ma, and M. Cristina Marchetti. Defect annihilation and proliferation in active Nematics. *Physical Review Letters*, 110(22):1–5, 2013.

[68] Luca Giomi, Mark J. Bowick, Prashant Mishra, Rastko Sknepnek, and M. Cristina Marchetti. Defect dynamics in active nematics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2029), 2014.

[69] Luca Giomi, Žiga Kos, Miha Ravnik, and Anupam Sengupta. Cross-talk between topological defects in different fields revealed by nematic microfluidics. *Proceedings of the National Academy of Sciences*, 114(29):E5771–E5777, 2017. Publisher: National Acad Sciences.

[70] Matthew Golden, Roman O. Grigoriev, Jyothishraj Nambisan, and Alberto Fernandez-Nieves. Physically informed data-driven modeling of active nematics. *Science Advances*, 9(27):eabq6120, July 2023. Publisher: American Association for the Advancement of Science.

[71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[72] Karna Gowda, Derek Ping, Madhav Mani, and Seppe Kuehn. Genomic structure predicts metabolite dynamics in microbial communities. *Cell*, 185(3):530–546.e25, February 2022.

[73] Richard Green, John Toner, and Vincenzo Vitelli. Geometry of thresholdless active flow in nematic microfluidics. *Phys. Rev. Fluids*, 2(10):104201, October 2017. Publisher: American Physical Society.

[74] Wei-hui Guo and Yu-li Wang. Retrograde Fluxes of Focal Adhesion Proteins in Response to Cell Migration and Mechanical Signals. *Molecular Biology of the Cell*, 18(11):4519–4527, November 2007. Publisher: American Society for Cell Biology (mboc).

[75] Hannah J. Gustafson, Nikolas Claussen, Stefano De Renzis, and Sebastian J. Streichan. Patterned Mechanical Feedback Establishes a Global Myosin Gradient. *Nature Communications*, 13(1):7050, 2022.

[76] William F. Guthrie. NIST/SEMATECH e-Handbook of Statistical Methods (NIST Handbook 151), 2020.

[77] Sangyoon J Han, Kevin S Bielawski, Lucas H Ting, Marita L Rodriguez, and Nathan J Sniadecki. Decoupling substrate stiffness, spread area, and micropost density: a close spatial relationship between traction forces and focal adhesions. *Biophys. J.*, 103(4):640–648, August 2012.

[78] Jana Hanke, Dimitri Probst, Assaf Zemel, Ulrich S Schwarz, and Sarah Köster. Dynamics of force generation by spreading platelets. *Soft Matter*, 14(31):6571–6581, August 2018.

[79] Jérôme Hardoüin, Rian Hughes, Amin Doostmohammadi, Justine Laurent, Teresa Lopez-Leon, Julia M. Yeomans, Jordi Ignés-Mullol, and Francesc Sagués. Reconfigurable flows and defect landscape of confined active nematics. *Communications Physics*, 2(1):1–9, 2019.

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[81] Shuonan He, Florencia Del Viso, Cheng-Yi Chen, Aissam Ikmi, Amanda E. Kroesen, and Matthew C. Gibson. An Axial Hox Code Controls Tissue Segmentation and Body Patterning in Nematostella Vectensis. *Science*, 361(6409):1377–1380, 2018.

[82] Ewan J. Hemingway, Prashant Mishra, M. Cristina Marchetti, and Suzanne M. Fielding. Correlation lengths in hydrodynamic models of active nematics. *Soft Matter*, 12(38):7943–7952, 2016. Publisher: Royal Society of Chemistry.

[83] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[84] Laura M Hoffman, Christopher C Jensen, Susanne Kloeker, C-L Albert Wang, Masaaki Yoshigi, and Mary C Beckerle. Genetic ablation of zyxin causes Mena/VASP mislocalization, increased motility, and deficits in actin remodeling. *J. Cell Biol.*, 172(5):771–782, February 2006.

[85] Yunfei Huang, Christoph Schell, Tobias B Huber, Ahmet Nihat Amp X0015e Imamp X0015f Ek, Nils Hersch, Rudolf Merkel, Gerhard Gompper, and Benedikt Sabass. Traction force microscopy with optimized regularization and automated Bayesian parameter selection for comparing cells. *Sci. Rep.*, 9(1):1–16, January 2019.

[86] K.D. Irvine and E. Wieschaus. Cell Intercalation during Drosophila Germband Extension and Its Regulation by Pair-Rule Segmentation Genes. *Development*, 120(4):827–841, 1994.

[87] Thomas Iskratsch, Haguy Wolfenson, and Michael P Sheetz. Appreciating force and shape—the rise of mechanotransduction in cell biology. *Nat. Rev. Mol. Cell Biol.*, 15(12):825–833, December 2014.

[88] Abhijeet Joshi, Elias Putzig, Aparna Baskaran, and Michael F. Hagan. The interplay between activity and filament flexibility determines the emergent properties of active nematics. *Soft Matter*, 15(1):94–101, 2019. Publisher: The Royal Society of Chemistry.

[89] Chaitanya Joshi, Sattvic Ray, Linnea M. Lemma, Minu Varghese, Graham Sharp, Zvonimir Dogic, Aparna Baskaran, and Michael F. Hagan. Data-Driven Discovery of Active Nematic Hydrodynamics. *Physical Review Letters*, 129(25):258001, December 2022. Publisher: American Physical Society.

[90] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas

Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, July 2021.

[91] Pakorn Kanchanawong, Gleb Shtengel, Ana M Pasapera, Ericka B Ramko, Michael W Davidson, Harald F Hess, and Clare M Waterman. Nanoscale architecture of integrin-based cell adhesions. *Nature*, 468(7323):580–584, November 2010.

[92] Alan A. Kaptanoglu, Brian M. de Silva, Urban Fasel, Kadierdan Kaheman, Andy J. Goldschmidt, Jared Callaham, Charles B. Delahunt, Zachary G. Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, J. Nathan Kutz, and Steven L. Brunton. PySINDy: A comprehensive Python package for robust sparse system identification. *Journal of Open Source Software*, 7(69):3994, 2022. Publisher: The Open Journal.

[93] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. Number: 6 Publisher: Nature Publishing Group.

[94] Eyal Karzbrun, Aimal H. Khankhel, Heitor C. Megale, Stella M. K. Glasauer, Yofiel Wyle, George Britton, Aryeh Warmflash, Kenneth S. Kosik, Eric D. Siggia, Boris I. Shraiman, and Sebastian J. Streichan. Human Neural Tube Morphogenesis in Vitro by Geometric Constraints. *Nature*, 599(7884):268–272, 2021.

[95] Felix C. Keber, Etienne Loiseau, Tim Sanchez, Stephen J. DeCamp, Luca Giomi, Mark J. Bowick, M. Cristina Marchetti, Zvonimir Dogic, and Andreas R. Bausch. Topology and dynamics of active nematic vesicles. *Science*, 345(6201):1135–1139, 2014.

[96] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

[97] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2013 International Conference on Learning Representations*, December 2013.

[98] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. arXiv:1906.02691 [cs, stat].

[99] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, November 2021. arXiv:1908.09257 [cs, stat].

[100] Maciej Koch-Janusz and Zohar Ringel. Mutual information, neural networks and the renormalization group. *Nature Physics*, 14(6):578–582, June 2018. Number: 6 Publisher: Nature Publishing Group.

[101] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *2021 International Conference on Learning Representations*, 2021.

[102] Felix S. Kratz, Lars Möllerherm, and Jan Kierfeld. Enhancing Robustness, Precision and Speed of Traction Force Microscopy with Machine Learning, September 2022. Pages: 2022.09.02.506331 Section: New Results.

[103] Dmitry Krotov, Julien O. Dubuis, Thomas Gregor, and William Bialek. Morphogenesis at criticality. *Proceedings of the National Academy of Sciences*, 111(10):3683–3688, February 2014. Publisher: Proceedings of the National Academy of Sciences.

[104] Uros Krzic, Stefan Gunther, Timothy E Saunders, Sebastian J Streichan, and Lars Hufnagel. Multiview Light-Sheet Microscope for Rapid in Toto Imaging. *Nature Methods*, 9(7):730–733, 2012.

[105] Nitin Kumar, Rui Zhang, Juan J. De Pablo, and Margaret L. Gardel. Tunable structure and dynamics of active liquid crystals. *Science Advances*, 4(10):1–13, 2018.

[106] Jean-Cheng Kuo. Mechanotransduction at focal adhesions: integrating cytoskeletal mechanics in migrating cells. *Journal of Cellular and Molecular Medicine*, 17(6):704–712, June 2013.

[107] Jules Lavalou, Qiyan Mao, Stefan Harmansa, Stephen Kerridge, Annemarie C. Lellouch, Jean-Marc Philippe, Stephane Audebert, Luc Camoin, and Thomas Lecuit. Formation of Polarized Contractile Interfaces by Self-Organized Toll-8/Cirl GPCR Asymmetry. *Developmental Cell*, 56(11):1574–1588.e7, 2021.

[108] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[109] Matthew F Lefebvre, Nikolas H Claussen, Noah P Mitchell, Hannah J Gustafson, and Sebastian J Streichan. Geometric control of myosin II orientation during axis elongation. *eLife*, 12:e78787, January 2023. Publisher: eLife Sciences Publications, Ltd.

[110] L.M. Lemma, S.J. DeCamp, Z. You, L. Giomi, and Z. Dogic. Statistical properties of autonomous flows in 2D active nematics. *Soft Matter*, 15:3264–3272, 2019.

[111] He Li, Xia-qing Shi, Mingji Huang, Xiao Chen, Minfeng Xiao, Chenli Liu, Hugues Chaté, and H. P. Zhang. Data-driven quantitative modeling of bacterial active nematics. *Proceedings of the National Academy of Sciences*, 116(3):777–785, 2019. Publisher: National Academy of Sciences.

[112] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022. Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2022/10/31/2022.07.20.500902.full.pdf.

[113] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016. Publisher: Cambridge University Press.

[114] Yang Liu, Rebecca Medda, Zheng Liu, Kornelia Galior, Kevin Yehl, Joachim P Spatz, Elisabetta Ada Cavalcanti-Adam, and Khalid Salaita. Nanoparticle tension probes patterned at the nanoscale: impact of integrin clustering on force transmission. *Nano Lett.*, 14(10):5539–5546, October 2014.

[115] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Montreal, QC, Canada, October 2021. IEEE.

[116] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, June 2022. ISSN: 2575-7075.

[117] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2017.

[118] F MacKintosh and C Schmidt. Active cellular materials. *Curr. Opin. Cell Biol.*, 22:29–35, August 2010.

[119] M. C. Marchetti, J. F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, Madan Rao, and R. Aditi Simha. Hydrodynamics of soft active matter. *Reviews of Modern Physics*, 85(3):1143–1189, 2013.

[120] D. Marenduzzo, E. Orlandini, and J. M. Yeomans. Hydrodynamics and rheology of active liquid crystals: A numerical investigation. *Physical Review Letters*, 98(11):3–6, 2007.

[121] Yonit Maroudas-Sacks, Liora Garion, Lital Shani-Zerbib, Anton Livshits, Erez Braun, and Kinneret Keren. Topological defects in the nematic order of actin fibers as organization centers of Hydra morphogenesis. *Nature Physics*, 17:251–259, 2021.

[122] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. *Physics Reports*, 810:1–124, May 2019. arXiv:1803.08823 [cond-mat, physics:physics, stat].

[123] Pankaj Mehta and David J Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

[124] Aaron F Mertz, Shiladitya Banerjee, Yonglu Che, Guy K German, Ye Xu, Callen Hyland, M Cristina Marchetti, Valerie Horsley, and Eric R Dufresne. Scaling of traction forces with the size of cohesive cell colonies. *Phys. Rev. Lett.*, 108(19):198101, May 2012.

[125] Noah P Mitchell, Matthew F Lefebvre, Vishank Jain-Sharma, Nikolas Claussen, Marion K Raich, Hannah J Gustafson, Andreas R Bausch, and Sebastian J Streichan. Morphodynamic Atlas for Drosophila Development, 2022.

[126] Sebastian K. Mitusch, Simon W. Funke, and Jørgen S. Dokken. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software*, 4(38):1292, 2019. Publisher: The Open Journal.

[127] Yoshihiro Morishita and Yoh Iwasa. Accuracy of positional information provided by multiple morphogen gradients with correlated noise. *Physical Review E*, 79(6):061905, June 2009. Publisher: American Physical Society (APS).

[128] Romain Mueller, Julia M Yeomans, and Amin Doostmohammadi. Emergence of active nematic behavior in monolayers of isotropic cells. *Physical review letters*, 122(4):048004, 2019. Publisher: APS.

[129] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1900654116.

[130] Michael Murrell, Patrick W Oakes, Martin Lenz, and Margaret L Gardel. Forcing cells into shape: the mechanics of actomyosin contractility. *Nat. Rev. Mol. Cell Biol.*, 16(8):486–498, August 2015.

[131] Takeharu Nagai, Jun Aruga, Osamu Minowa, Takashi Sugimoto, Yoshiki Ohno, Tetsuo Noda, and Katsuhiko Mikoshiba. Zic2 Regulates the Kinetics of Neurulation. *Proceedings of the National Academy of Sciences*, 97(4):1618–1623, 2000.

[132] Muneaki Nakamura, Lu Chen, Stuart C. Howes, Tony D. Schindler, Eva Nogales, and Zev Bryant. Remote control of myosin and kinesin motors using light-activated gearshifting. *Nature Nanotechnology*, 9:693–697, 2014.

[133] Andrew Y Ng and Michael I Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems*, 14, 2001.

[134] Nicholas Noll, Madhav Mani, Idse Heemskerk, Sebastian J. Streichan, and Boris I. Shraiman. Active Tension Network Model Suggests an Exotic Mechanical State Realized in Epithelial Tissues. *Nature Physics*, 13(12):1221–1226, 2017. Publisher: Nature Publishing Group.

[135] Jacob Notbohm, Shiladitya Banerjee, Kazage J C Utuje, Bomi Gweon, Hwanseok Jang, Yongdoo Park, Jennifer Shin, James P Butler, Jeffrey J Fredberg, and M Cristina Marchetti. Cellular Contraction and Polarization Drive Collective Cellular Motion. *Biophys. J.*, 110(12):2729–2738, June 2016.

[136] Patrick W. Oakes, Shiladitya Banerjee, M. Christina Marchetti, and Margaret L. Gardel. Geometry regulates traction stresses in adherent cells. *Biophysical Journal*, 107(4):825–833, 2014.

[137] Patrick W Oakes, Tamara C Bidone, Yvonne Beckham, Austin V Skeeters, Guillermina R Ramirez-San Juan, Stephen P Winter, Gregory A Voth, and Margaret L Gardel. Lamellipodium is a myosin-independent mechanosensor. *Proc. Natl. Acad. Sci. U. S. A.*, 115(11):2646–2651, March 2018.

[138] Patrick W Oakes, Elizabeth Wagner, Christoph A Brand, Dimitri Probst, Marco Linke, Ulrich S Schwarz, Michael Glotzer, and Margaret L Gardel. Optogenetic control of RhoA reveals zyxin-mediated elasticity of stress fibres. *Nat. Commun.*, 8:15817, June 2017.

[139] Jeremie Palacci, Stefano Sacanna, Asher Preska Steinberg, David J. Pine, and Paul M. Chaikin. Living Crystals of Light-Activated Colloidal Surfers. *Science*, 339(6122):936–940, 2013. Publisher: American Association for the Advancement of Science.

[140] Adam C. Paré, Pooja Naik, Jay Shi, Zachary Mirman, Karl H Palmquist, and Jennifer A Zallen. An LRR Receptor-Teneurin System Directs Planar Polarity at Compartment Boundaries. *Developmental Cell*, 51(2):208–221, 2019.

[141] Adam C. Paré, Athea Vichas, Christopher T Fincher, Zachary Mirman, Dene L Farrell, Avantika Mainieri, and Jennifer A Zallen. A Positional Toll Receptor Code Directs Convergent Extension in Drosophila. *Nature*, 515(7528):523–527, 2014.

[142] Adam C. Paré and Jennifer A. Zallen. Cellular, Molecular, and Biophysical Control of Epithelial Cell Intercalation. In *Current Topics in Developmental Biology*, volume 136, pages 167–193. Elsevier, 2020.

[143] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. Issue: arXiv:1912.01703 _eprint: 1912.01703.

[144] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Phys. Rev. Lett.*, 120(2):024102, January 2018. Publisher: American Physical Society.

[145] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-Learn: Machine Learning in Python, 2018. Issue: arXiv:1201.0490 _eprint: 1201.0490.

[146] Adrian F Pegoraro, Paul Janmey, and David A Weitz. Mechanical Properties of the Cytoskeleton and Cells. *Cold Spring Harb. Perspect. Biol.*, 9(11), November 2017.

[147] Chenhui Peng, Taras Turiv, Yubing Guo, Qi Huo Wei, and Oleg D. Lavrentovich. Command of active matter by topological defects and patterns. *Science*, 354(6314):882–885, 2016.

[148] Olivier Pertz, Louis Hodgson, Richard L Klemke, and Klaus M Hahn. Spatiotemporal dynamics of RhoA activity in migrating cells. *Nature*, 440(7087):1069–1072, April 2006.

[149] Robert Brooks Phillips, Jane Kondev, and Julie Theriot. *Physical Biology of the Cell*. Garland Science, 2009.

[150] A. Ponti, M. Machacek, S. L. Gupton, C. M. Waterman-Storer, and G. Danuser. Two Distinct Actin Networks Drive the Protrusion of Migrating Cells. *Science*, 305(5691):1782–1786, 2004. _eprint: https://www.science.org/doi/pdf/10.1126/science.1100533.

[151] Masha Prager-Khoutorsky, Alexandra Lichtenstein, Ramaswamy Krishnan, Kavitha Rajendran, Avi Mayo, Zvi Kam, Benjamin Geiger, and Alexander D Bershadsky. Fibroblast polarization is a matrix-rigidity-dependent process controlled by focal adhesion mechanosensing. *Nature Cell Biology*, 13(12):1457–1465, December 2011.

[152] Jacques Prost, Frank Jülicher, and Jean-François Joanny. Active gel physics. *Nat. Phys.*, 11(2):111–117, February 2015. Publisher: Springer Science and Business Media LLC.

[153] Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao, and Taritree Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560(7716):41–48, 2018. Publisher: Nature Publishing Group.

[154] Sriram Ramaswamy. The Mechanics and Statistics of Active Matter. *Annual Review of Condensed Matter Physics*, 1(1):323–345, 2010.

[155] Miha Ravnik and Slobodan Žumer. Landau–de Gennes modelling of nematic liquid crystal colloids. *Liquid Crystals*, 36(10-11):1201–1214, 2009. Publisher: Taylor & Francis.

[156] Steven A Redford, Jonathan Colen, Jordan L Shivers, Sasha Zemsky, Mehdi Molaei, Carlos Floyd, Paul V Ruijgrok, Vincenzo Vitelli, Zev Bryant, Aaron R Dinner, and others. Motor crosslinking augments elasticity in active nematics. *arXiv preprint arXiv:2308.16831*, 2023.

[157] Emmanuel G Reynaud, Jan Peychl, Jan Huisken, and Pavel Tomancak. Guide to Light-Sheet Microscopy for Adventurous Biologists. *Nature Methods*, 12(1):30–34, 2015.

[158] Danilo Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1530–1538. PMLR, June 2015. ISSN: 1938-7228.

[159] Patrizia Romani, Lorea Valcarcel-Jimenez, Christian Frezza, and Sirio Dupont. Crosstalk between mechanotransduction and metabolism. *Nat. Rev. Mol. Cell Biol.*, 22(1):22–38, January 2021.

[160] Nicolas Romeo, Alasdair Hastewell, Alexander Mietke, and Jörn Dunkel. Learning developmental mode dynamics from single-cell trajectories. *eLife*, 10:e68679, December 2021. Publisher: eLife Sciences Publications, Ltd.

[161] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[162] Benedikt Sabass, Margaret L Gardel, Clare M Waterman, and Ulrich S Schwarz. High resolution traction force microscopy based on experimental and computational advances. *Biophys. J.*, 94(1):207–220, December 2008.

[163] Tim Sanchez, Daniel T.N. Chen, Stephen J. Decamp, Michael Heymann, and Zvonimir Dogic. Spontaneous motion in hierarchically assembled active matter. *Nature*, 491(7424):431–434, 2012. Publisher: Nature Publishing Group.

[164] Thuan Beng Saw, Amin Doostmohammadi, Vincent Nier, Leyla Kocgozlu, Sumesh Thampi, Yusuke Toyama, Philippe Marcq, Chwee Teck Lim, Julia M Yeomans, and Benoit Ladoux. Topological defects in epithelia govern cell death and extrusion. *Nature*, 544(7649):212–216, 2017. Publisher: Nature Publishing Group.

[165] Colin R Scheibner, Anton Souslov, Debarghya Banerjee, P. Surowka, William T.M. Irvine, and Vincenzo Vitelli. Odd elasticity. *Nature Physics*, 16:475–480, 2020.

[166] Jürgen Schmidhuber. Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. Publisher: Elsevier Ltd.

[167] Matthew S. Schmitt, Jonathan Colen, Stefano Sala, John Devany, Shailaja Seetharaman, Margaret L. Gardel, Patrick W. Oakes, and Vincenzo Vitelli. Zyxin is all you need: machine learning adherent cell mechanics, February 2023. arXiv:2303.00176 [cond-mat, physics:physics].

[168] Samuel Schoenholz, Eikin Cubuk, Daniel Sussman, Efthimios Kaxiras, and Andrea Liu. A structural approach to relaxation in glassy liquids. *Nature Physics*, 12(5):469–471, 2016. Publisher: Nature Publishing Group.

[169] Ulrich S Schwarz and Margaret L Gardel. United we stand: integrating the actin cytoskeleton and cell-matrix adhesions in cellular mechanotransduction. *J. Cell Sci.*, 125(Pt 13):3051–3060, July 2012.

[170] Suraj Shankar and M. Cristina Marchetti. Hydrodynamics of Active Defects: From Order to Chaos to Defect Ordering. *Phys. Rev. X*, 9(4):041047, December 2019. Publisher: American Physical Society.

[171] Suraj Shankar, Sriram Ramaswamy, M. Cristina Marchetti, and Mark J. Bowick. Defect Unbinding in Active Nematics. *Phys. Rev. Lett.*, 121(10):108002, September 2018. Publisher: American Physical Society.

[172] Guojun Sheng, Alfonso Martinez Arias, and Ann Sutherland. The primitive streak and cellular principles of building an amniote body through gastrulation. *Science*, 374(6572), December 2021. Publisher: American Association for the Advancement of Science (AAAS).

[173] Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5(49):2104, 2020. Publisher: The Open Journal.

[174] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018. Publisher: American Association for the Advancement of Science.

[175] Christopher J Soelistyo, Giulia Vallardi, Guillaume Charras, and Alan R Lowe. Learning biophysical determinants of cell fate with deep neural networks. *Nature Machine Intelligence*, 4(7):636–644, June 2022. Publisher: Nature Publishing Group.

[176] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR, June 2015. ISSN: 1938-7228.

[177] Jérôme R D Soiné, Christoph A Brand, Jonathan Stricker, Patrick W Oakes, Margaret L Gardel, and Ulrich S Schwarz. Model-based Traction Force Microscopy Reveals Differential Tension in Cellular Actin Bundles. *PLoS Comput. Biol.*, 11(3):e1004076, March 2015.

[178] Andrey Sokolov, Ali Mozaffari, Rui Zhang, Juan J De Pablo, and Alexey Snezhko. Emergence of radial tree of bend stripes in active nematics. *Physical Review X*, 9(3):031014, 2019. Publisher: APS.

[179] Josephine Solowiej-Wedderburn and Carina M. Dunlop. Sticking around: Cell adhesion patterning for energy minimization and substrate mechanosensing. *Biophysical Journal*, 121(9):1777–1786, May 2022. Publisher: Elsevier.

[180] Vishal Soni, Ephraim S. Bililign, Sofia Magkiriadou, Stefano Sacanna, Denis Bartolo, Michael J. Shelley, and William T.M. Irvine. The odd free surface flows of a colloidal chiral fluid. *Nature Physics*, 15:1188–1194, 2019.

[181] Sebastian J Streichan, Matthew F Lefebvre, Nicholas Noll, Eric F Wieschaus, and Boris I Shraiman. Global Morphogenetic Flow Is Accurately Predicted by the Spatial Distribution of Myosin Motors. *eLife*, 7:e27454, 2018.

[182] Jonathan Stricker, Yvonne Aratyn-Schaus, Patrick W Oakes, and Margaret L Gardel. Spatiotemporal constraints on the force-dependent growth of focal adhesions. *Biophys. J.*, 100(12):2883–2893, June 2011.

[183] Jonathan Stricker, Yvonne Beckham, Michael W Davidson, and Margaret L Gardel. Myosin II-Mediated Focal Adhesion Maturation Is Tension Insensitive. *PLoS One*, 8(7):e70652, 2013.

[184] Angughali Sumi, Peran Hayes, Arturo D'Angelo, Julien Colombelli, Guillaume Salbreux, Kai Dierkes, and Jérôme Solon. Adherens Junction Length during Tissue Contraction Is Controlled by the Mechanosensitive Activity of Actomyosin and Junctional Recycling. *Developmental Cell*, 47(4):453–463.e3, 2018.

[185] Rohit Supekar, Boya Song, Alasdair Hastewell, Gary P. T. Choi, Alexander Mietke, and Jörn Dunkel. Learning hydrodynamic equations for active matter from particle simulations and experiments. *Proceedings of the National Academy of Sciences*, 120(7):e2206994120, February 2023. Publisher: Proceedings of the National Academy of Sciences.

[186] Tatyana Svitkina. The Actin Cytoskeleton and Actin-Based Motility. *Cold Spring Harb. Perspect. Biol.*, 10(1):a018267, January 2018.

[187] Masako Tamada, Jay Shi, Kia S. Bourdot, Sara Supriyatno, Karl H. Palmquist, Omar L. Gutierrez-Ruiz, and Jennifer A. Zallen. Toll Receptors Remodel Epithelia by Directing Planar-Polarized Src and PI3K Activity. *Developmental Cell*, 56(11):1589–1602.e9, 2021.

[188] Amanda J Tan, Eric Roberts, Spencer A Smith, Ulyses Alvarado Olvera, Jorge Arteaga, Sam Fortini, Kevin A Mitchell, and Linda S Hirst. Topological chaos in active nematics. *Nature Physics*, 15(10):1033–1039, 2019. Publisher: Nature Publishing Group.

[189] Sumesh P. Thampi, Ramin Golestanian, and Julia M. Yeomans. Velocity correlations in an active nematic. *Physical Review Letters*, 111(11):2–6, 2013.

[190] Sumesh P. Thampi, Ramin Golestanian, and Julia M. Yeomans. Vorticity, defects and correlations in active turbulence. *Philosophical Transactions of the Royal Society A*, 372(2029), 2014.

[191] Ingo Thievessen, Peter M Thompson, Sylvain Berlemont, Karen M Plevock, Sergey V Plotnikov, Alice Zemljic-Harpf, Robert S Ross, Michael W Davidson, Gaudenz Danuser, Sharon L Campbell, and Clare M Waterman. Vinculin-actin interaction couples actin retrograde flow to focal adhesions, but is dispensable for focal adhesion growth. *J. Cell Biol.*, 202(1):163–177, July 2013.

[192] Manuel Théry, Anne Pépin, Emilie Dressaire, Yong Chen, and Michel Bornens. Cell distribution of stress fibres in response to the geometry of the adhesive environment. *Cell Motility*, 63(6):341–355, 2006. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cm.20126.

[193] J. Toner, Y. Tu, and S. Ramaswamy. Hydrodynamics and phases of flocks. *Annals of Physics*, 318:170–244, 2005.

[194] Fredi Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, 2000.

[195] Qingzong Tseng, Irene Wang, Eve Duchemin-Pelletier, Ammar Azioune, Nicolas Carpi, Jie Gao, Odile Filhol, Matthieu Piel, Manuel Théry, and Martial Balland. A new micropatterning method of soft substrates reveals that different tumorigenic signals can promote or reduce cell contraction levels. *Lab Chip*, 11(13):2231–2240, July 2011.

[196] Turing, Alan. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.

[197] H. Turlier, B. Audoly, J. Prost, and J.-F. Joanny. Furrow Constriction in Animal Cell Cytokinesis. *Biophys. J.*, 106(1):114–123, January 2014. Publisher: Elsevier BV.

[198] Evert P.L. Van Nieuwenburg, Ye Hua Liu, and Sebastian D. Huber. Learning phase transitions by confusion. *Nature Physics*, 13(5):435–439, 2017. _eprint: 1610.02048.

[199] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[200] Roman Vetter and Dagmar Iber. Precision of morphogen gradients in neural tube development. *Nature Communications*, 13(1), March 2022. Publisher: Springer Science and Business Media LLC.

[201] Matheus P Viana, Jianxu Chen, Theo A Knijnenburg, Ritvik Vasan, Calysta Yan, Joy E Arakaki, Matte Bailey, Ben Berry, Antoine Borensztejn, Eva M Brown, Sara Carlson, Julie A Cass, Basudev Chaudhuri, Kimberly R Cordes Metzler, Mackenzie E Coston, Zach J Crabtree, Steve Davidson, Colette M DeLizo, Shailja Dhaka, Stephanie Q Dinh, Thao P Do, Justin Domingus, Rory M Donovan-Maiye, Alexandra J Ferrante, Tyler J Foster, Christopher L Frick, Griffin Fujioka, Margaret A Fuqua, Jamie L Gehring, Kaytlyn A Gerbin, Tanya Grancharova, Benjamin W Gregor, Lisa J Harrylock, Amanda Haupt, Melissa C Hendershott, Caroline Hookway, Alan R Horwitz, H Christopher Hughes, Eric J Isaac, Gregory R Johnson, Brian Kim, Andrew N Leonard, Winnie W Leung, Jordan J Lucas, Susan A Ludmann, Blair M Lyons, Haseeb Malik, Ryan McGregor, Gabe E Medrash, Sean L Meharry, Kevin Mitcham, Irina A Mueller, Timothy L Murphy-Stevens, Aditya Nath, Angelique M Nelson, Sandra A Oluoch, Luana Paleologu, T Alexander Popiel, Megan M Riel-Mehan, Brock Roberts, Lisa M Schaefbauer, Magdalena Schwarzl, Jamie Sherman, Sylvain Slaton, M Filip Sluzewski, Jacqueline E Smith, Youngmee Sul, Madison J Swain-Bowden, W Joyce Tang, Derek J Thirstrup, Daniel M Toloudis, Andrew P Tucker, Veronica Valencia, Winfried Wiegraebe, Thushara Wijeratna, Ruian Yang, Rebecca J Zaunbrecher, Ramon Lorenzo D Labitigan, Adrian L Sanborn, Graham T Johnson, Ruwanthi N Gunawardane, Nathalie Gaudreault, Julie A Theriot, and Susanne M Rafelski. Integrated intracellular organization and its variations in human iPS cells. *Nature*, 613(7943):345–354, January 2023.

[202] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel Type of Phase Transition in a System of Self-Driven Particles. *Physical Review Letters*, 75(6):1226–1229, August 1995. Publisher: American Physical Society.

[203] Timothée Vignaud, Calina Copos, Christophe Leterrier, Mauricio Toro-Nahuelpan, Qingzong Tseng, Julia Mahamid, Laurent Blanchoin, Alex Mogilner, Manuel Théry, and Laetitia Kurzawa. Stress fibres are embedded in a contractile cortical network. *Nat. Mater.*, 20(3):410–420, March 2021.

[204] R. Voituriez, J. F. Joanny, and J. Prost. Spontaneous flow transition in active polar gels. *Europhysics Letters*, 70(3):404–410, 2005.

[205] John B. Wallingford. Neural Tube Closure and Neural Tube Defects: Studies in Animal Models Reveal Known Knowns and Known Unknowns. *American Journal of Medical Genetics Part C: Seminars in Medical Genetics*, 135C(1):59–68, 2005.

[206] Yu-li Wang and Yun-Chu Lin. Traction force microscopy by deep learning. *Biophysical Journal*, 120(15):3079–3090, August 2021.

[207] Kun-Ta Wu, Jean Bernard Hishamunda, Daniel TN Chen, Stephen J DeCamp, Ya-Wen Chang, Alberto Fernández-Nieves, Seth Fraden, and Zvonimir Dogic. Transition from turbulent to coherent flows in confined three-dimensional active fluids. *Science*, 355(6331):eaal1979, 2017. Publisher: American Association for the Advancement of Science.

[208] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models: A Comprehensive Survey of Methods and Applications, March 2023. arXiv:2209.00796 [cs].

[209] Patricia Ybot-Gonzalez, Carles Gaston-Massuet, Gemma Girdler, John Klingensmith, Ruth Arkell, Nicholas D. E. Greene, and Andrew J. Copp. Neural Plate Morphogenesis during Mouse Neurulation Is Regulated by Antagonism of Bmp Signalling. *Development*, 134(17):3203–3211, 2007.

[210] Masaaki Yoshigi, Laura M Hoffman, Christopher C Jensen, H Joseph Yost, and Mary C Beckerle. Mechanical force mobilizes zyxin from focal adhesions to actin filaments and regulates cytoskeletal reinforcement. *J. Cell Biol.*, 171(2):209–215, October 2005.

[211] Jessica C Yu and Rodrigo Fernandez-Gonzalez. Local Mechanical Forces Promote Polarized Junctional Assembly and Axis Elongation in Drosophila. *eLife*, 5:1–15, 2016.

[212] Marcin Zagorski, Yoji Tabata, Nathalie Brandenberg, Matthias P. Lutolf, Gašper Tkačik, Tobias Bollenbach, James Briscoe, and Anna Kicheva. Decoding of position in the developing neural tube from antiparallel morphogen gradients. *Science*, 356(6345):1379–1383, June 2017. Publisher: American Association for the Advancement of Science (AAAS).

[213] Jennifer A Zallen and Eric F Wieschaus. Patterned Gene Expression Directs Bipolar Planar Polarity in Drosophila. *Developmental Cell*, 6:343–355, 2004.

[214] Assaf Zaritsky, Andrew R Jamieson, Erik S Welf, Andres Nevarez, Justin Cillay, Ugur Eskiocak, Brandi L Cantarel, and Gaudenz Danuser. Interpretable deep learning uncovers cellular properties in label-free live cell images that are predictive of highly metastatic melanoma. *Cell Syst*, 12(7):733–747.e6, July 2021.

[215] Rui Zhang, Nitin Kumar, Jennifer L. Ross, Margaret L. Gardel, and Juan J. De Pablo. Interplay of structure, elasticity, and dynamics in actin-based nematic materials. *Proceedings of the National Academy of Sciences of the United States of America*, 115(2):E124–E133, 2017.

[216] Rui Zhang, Steven A Redford, Paul V Ruijgrok, Nitin Kumar, Ali Mozaffari, Sasha Zemsky, Aaron R Dinner, Vincenzo Vitelli, Zev Bryant, Margaret L Gardel, and Juan J De Pablo. Structuring Stress for Active Materials Control. *arXiv preprint arXiv:1912.01630*, 2019.

[217] Rui Zhang, Tyler Roberts, Igor S. Aranson, and Juan J. De Pablo. Lattice Boltzmann simulation of asymmetric flow in nematic liquid crystals with finite anchoring. *Journal of Chemical Physics*, 144(8), 2016.

[218] Rui Zhang, Ye Zhou, Mohammad Rahimi, and Juan J De Pablo. Dynamic structure of active nematic shells. *Nature communications*, 7(1):1–9, 2016. Publisher: Nature Publishing Group.

[219] Simon Čopar, Žiga Kos, Tadej Emeršič, and Uroš Tkalec. Microfluidic control over topological states in channel-confined nematic flows. *Nature communications*, 11(1):1–10, 2020. Publisher: Nature Publishing Group.