Manuscript submitted to **Biophysical** *Journal*
Article

Enhancing Robustness, Precision and Speed of Traction Force Microscopy with Machine Learning

Felix S. Kratz[1], Lars Möllerherm[1], and Jan Kierfeld[1,*]

[1]TU Dortmund University, Germany, Department of Physics
[*]Correspondence: Jan.Kierfeld@tu-dortmund.de

ABSTRACT Traction patterns of adherent cells provide important information on their interaction with the environment, cell migration or tissue patterns and morphogenesis. Traction Force Microscopy is a method aimed at revealing these traction patterns for adherent cells on engineered substrates with known constitutive elastic properties from deformation information obtained from substrate images. Conventionally, the substrate deformation information is processed by numerical algorithms of varying complexity to give the corresponding traction field via solution of an ill-posed inverse elastic problem. We explore the capabilities of a deep convolutional neural network as a computationally more efficient and robust approach to solve this inversion problem. We develop a general purpose training process based on collections of circular force patches as synthetic training data, which can be subjected to different noise levels for additional robustness. The performance and the robustness of our approach against noise is systematically characterized for synthetic data, artificial cell models and real cell images, which are subjected to different noise levels. A comparison to state-of-the-art Bayesian Fourier transform traction cytometry reveals the precision, robustness, and speed improvements achieved by our approach, leading to an acceleration of Traction Force Microscopy methods in practical applications.

SIGNIFICANCE Traction force microscopy is an important biophysical technique to gain quantitative information about forces exerted by adherent cells. It relies on solving an inverse problem to obtain cellular traction forces from image-based displacement information. We present a deep convolutional neural network as a computationally more efficient and robust approach to solve this ill-posed inversion problem. We characterize the performance and the robustness of our approach against noise systematically for synthetic data, artificial cell models and real cell images, which are subjected to different noise levels and compare performance and robustness to state-of-the-art Bayesian Fourier transform traction cytometry. We demonstrate that machine learning can enhance robustness, precision and speed in traction force microscopy.

INTRODUCTION

Many cellular processes are intrinsically connected to mechanical interactions of the cell with its surroundings. Mechanical surface forces control the shape of single cells or groups of cells in tissue patterns and morphogenesis (1). Forces alter cell behavior via mechanotransduction (2) and affect cell migration and adhesion. Gaining access to the forces (or tractions, i.e., forces per area) exerted by the cell during critical processes like migration or proliferation can give insight into their inner workings and aid the development of novel medication and treatment, e.g. by identifying changes of cellular forces in diseased states. Altered cell behavior is present for diseases (3) like atherosclerosis (4), deafness (5), or tumor metastasis (6).

*T*raction *F*orce *M*icroscopy (TFM) is a modern method to measure tractions exerted by an adherent cell by deducing them from the cell-induced deformations of an engineered external substrate of known elastic properties (7–9). Beyond adherent cells it has applications to a broader range of biological and physical systems where interfacial forces are of interest (10). TFM thus constitutes a classic inverse problem in elasticity, where tractions or forces are calculated from displacement for given material properties. This inverse problem turns out to be ill-posed, i.e., noise or slight changes in displacement input data induce large deviations in traction output data because of singular components of the elastic Green's tensor. This technical problem has been addressed by different regularization schemes that have been developed over the last two decades (11–15). Recent studies show that *M*achine *L*earning (ML) can be an elegant alternative to numerical schemes when the inverse problem to a bounded problem is ill-posed in the context of elasticity or rheology. This has been explored for other rheological inverse problem classes such as in pendant drop tensiometry (16). ML-aided traction force determination can thus provide an elegant way to improve TFM as a method, as recent studies have already begun to show (17, 18). A systematic investigation of ML-aided TFM with respect to an optimal general purpose training set that allows the machine to predict tractions accurately across many experimental situations as well as a systematic investigation of robustness with respect to noise, which is present in any experimental realization, are still lacking.

The first implementation of TFM was achieved by Harris and coworkers in the early 1980s, where thin silicone films are wrinkled by compressive surface stresses, inflicted by the traction field of the cell (19). Due to the inherent non-linearity of wrinkling and the connected difficulties solving the inverse elastic problem, this method has been superseded by linear elastic hydrogel marker based TFM introduced by Dembo et al. (20). Due to the simplicity of the hydrogel marker based approach, it is the most commonly used and most evolved method. Alternative techniques and extensions include micro-needle deformations (21), force microscopy with molecular tension probes (22), and 3D techniques (23). Wrinkling based TFM has recently been re-explored with generative adversarial neural networks with promising results (18).

In this paper, we focus on the hydrogel marker based technique and train a deep *C*onvolutional *N*eural *N*etwork (CNN), which has the capabilities to solve the inverse elastic problem reliably, giving fast and robust access to the traction pattern exerted by the cell onto a substrate. Specifically, we do this by numerically solving the elastic forward problem, where we prescribe generic traction fields and solve the governing elastic equations to generate an associated displacement field. The "artificial" displacement field generated this way is used as a training input for our NN, while we use the prescribed traction field as the labels for our training set. This way the network learns the mapping between displacement and traction fields and is able to generate traction fields for displacement fields never seen before, while still respecting the relevant governing elastic equations. We use traction force distributions generated from collections of circular force patches as training data, which seems a natural general choice to allow the NN to predict generic force distributions in cell adhesion but should also cover other future applications. We show that the proper, "physics-informed" choice of training data and inclusion of artificial noise is a similarly important step in the ML solution of the inverse problem as the proper choice of regularization in conventional TFM techniques, in order to achieve the best compromise between accuracy and robustness.

MATERIALS AND METHODS

Hydrogel Marker Based TFM

The hydrogel marker approach to TFM can be described as follows. First, a cross-linked gel substrate, often Polydimethylsiloxane (PDSM) or Polyacrylamide substrates (PAA) (24), is cultivated. The cross-linked gel can be classified as an elastic substrate with long linkage lifetimes compared to the imaging process (25).

Second, the substrate is coated with proteins prevalent in the extracellular matrix like collagen type I, gelatin, laminin, or fibronectin, allowing the cell to adhere to the substrate. Fluorescent marker beads embedded in the gel substrate aid the determination of cell-induced substrate deformations. The reference and stressed positions of the marker beads can be determined via various microscopy techniques, ranging from confocal to optical microscopy (19).

Third, to infer the displacement field from the marker bead positions, a particle tracking velocimetry (PTV) algorithm, a particle image velocimetry (PIV) algorithm or a CNN particle tracker (26) is used, which calculates the discrete displacement field.

The information about the displacement field, combined with the predetermined constitutive properties of the hydrogel substrate gives access to the traction field of the cell via the inverse solution of the elastic deformation problem. For homogeneous, isotropic, and linear elastic solids the displacement field $\vec{u}$ satisfies the equations of equilibrium in the bulk (27)

$$(1-2\nu)\Delta\vec{u}+\vec{\nabla}(\vec{\nabla}\cdot\vec{u})=0\,, \tag{1}$$

while the force balance at the surface is modified to account for external tractions $\vec{t}$ (forces per area applied to the surface)

$$\sigma\vec{n}=\vec{t}\,, \tag{2}$$

where $\vec{n}$ is the surface normal vector and σ the stress tensor.

The TFM gel substrate can be considered sufficiently thick to be modelled as an elastic half-space ($z>0$), bounded by the x-y-plane, at which traction forces $\vec{t}=\vec{t}(x,y)$ are applied. The displacements are a solution of the boundary problem (1), (2), which is given by the spatial convolution of the external traction field $\vec{t}(x,y)$ with the Green's tensor $\mathbf{G}$ over the boundary of the surface S (27):

$$\vec{u}(x,y,z)=\iint_S \mathbf{G}(x-x',y-y',z)\vec{t}(x',y')\mathrm{d}x'\mathrm{d}y'\,. \tag{3}$$

Up to this point both tractions and displacement are three-dimensional vectors. The full three-dimensional Green's tensor $\mathbf{G}(x-x', y-y', z)$ is given in the Appendix.

In traction force microscopy, it can be assumed that adherent cells exert *in-plane* surface tractions ($t_z = 0$), and we are interested in *in-plane* displacement $\vec{u} = (u_x, u_y)$ only because out-of-plane z-displacements are hard to quantify by microscopy. Moreover, out-of-plane displacements are small for incompressible materials (see Appendix). These assumptions make the problem (3) quasi-two-dimensional in the plane $z = 0$, such that the Green's tensor is given by the 2x2 matrix (20)

$$\mathbf{G}(x,y) = \frac{1+\nu}{\pi E r^3} \begin{pmatrix} (1-\nu)r^2 + \nu x^2 & \nu xy \\ \nu xy & (1-\nu)r^2 + \nu y^2 \end{pmatrix}. \tag{4}$$

It solves the elastic boundary problem for in-plane tractions and displacements if the tractions vanish at infinity.

TFM is essentially a technique to provide a numerical solution for the inverse elastic problem posed by asking to recover the traction field $\vec{t} = (t_x, t_y)$ from (3) via a deconvolution of the right hand side surface integral. This can be done in real space (11, 12, 20) or in Fourier space (28).

Employing the convolution theorem for the Fourier-transform $\mathscr{F}\mathscr{T}$ of a convolutional integral, the deconvolution problem encountered in (3) can equivalently be stated as performing two Fourier-transforms and one inverse Fourier-transform

$$\mathscr{F}\mathscr{T}\left(\vec{u}\right)(u,v) = \mathscr{F}\mathscr{T}\left(\mathbf{G}\right)(u,v)\ \mathscr{F}\mathscr{T}\left(\vec{t}\right)(u,v) \tag{5}$$

$$\vec{t}(x,y) = \mathscr{F}\mathscr{T}^{-1}\left\{\left(\mathscr{F}\mathscr{T}\left(\mathbf{G}\right)\right)^{-1}\mathscr{F}\mathscr{T}\left(\vec{u}\right)\right\}(x,y), \tag{6}$$

which is named *F*ourier-*T*ransform-*T*raction- *F*orce-*C*ytometry (FTTC) (28).

Common iterative techniques used for numerical deconvolution can become unstable when subjected to noisy data, which is why conventional approaches to the ill-posed inverse elastic problem rely on regularization techniques (e.g. Tikhonov(L2)- or Lasso(L1)-regularization) coupled with iterative minimization schemes (11–14, 29). This applies both to real space and Fourier space methods. Regularization improves stability while accuracy might suffer. The optimal choice of regularization parameters is important but subjective. In *B*ayesian *F*ourier *T*ransform *T*raction *C*ytometry (BFTTC) the regularization parameters need not be picked manually and heuristically, but they are inferred from probability theory, making an easy to use and objective FTTC method (15, 30).

The shortcomings of most conventional approaches are systematic under-predictions and edge smoothing of the constructed traction field, caused by the regularization (31), as well as elevated computational effort, inflicted by the computationally demanding iterative deconvolution techniques and transformations at play.

A recent trend in many fields, including the natural sciences, has shown the capabilities of ML-based approaches in such ill-posed and ill-conditioned scenarios (32, 33), often outperforming complex algorithms by orders of magnitude in computing time and precision, and thus allowing for new and more accessible workflows with reduced computational overhead. ML based approaches to traction force microscopy (17) and wrinkle force microscopy (18) have recently been discussed and find that deep CNNs can perform the deconvolution of (3), by learning the mapping from strain-space to surface traction-space in training. The existing NN approaches show a promising proof-of-concept, which we want to extend further in the present article. While regularization is particularly important in conventional TFM approaches for accuracy and stability, accuracy and robustness to noise of deep CNNs crucially depend on the choice of training data.

Machine Learning the Inverse Problem

If we want to teach a machine to solve an inverse problem for us, counter-intuitively, we do not need to know how to solve the inverse problem itself. We only need to know how to solve the corresponding forward problem i.e., we only need to know how to precisely formulate the learning task for the network and provide sample data that characterizes the problem well enough. In training, the machine detects correlations in the data and uses it to solve for arbitrary non-linear mappings from input space to output space. This is one of the groundbreaking traits deep learning offers and which, in combination with hardware acceleration, allows to solve problems not feasible or traceable before (16, 32, 33).

Thus, we are interested in formulating the learning task in the most precise way. A first step to this is a solid understanding of the forward problem and all the involved quantities including the proper, "physics-informed" choice of training data. Since we generate our training data numerically, the second step is a robust numerical framework, which outputs physically correct data. The third step is to find a sensible data representation, which contains all the relevant information. The final step is to find a network architecture with enough capacity for the problem, to train it and to evaluate its performance and robustness. We will address these points in the following and finally compare our network with state-of-the-art approaches.

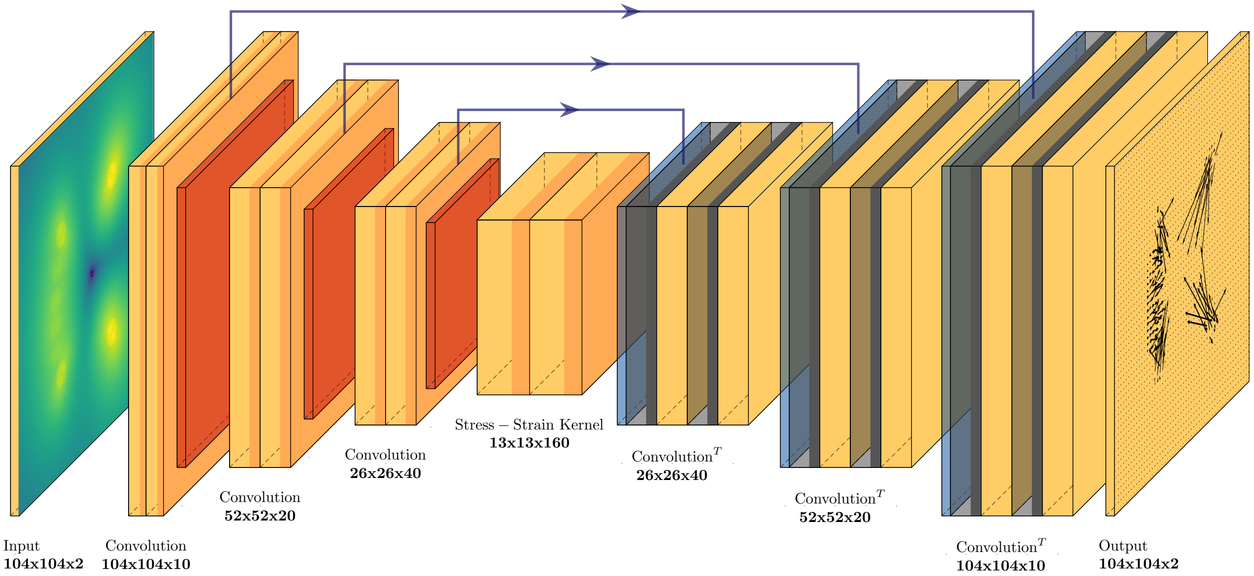

Figure 1: The network we employ is a *Unet* Convolutional Neural Network with a discretized displacement field as an input and a discretized traction field as an output. The mapping from input to output is learned in training by adapting the parameters of the convolutional and transposed convolutional layers of the network. The central bottleneck of the network can be interpreted as a representation of the stress-strain kernel of the elasticity employed here. Eventually the network will be able to reconstruct the traction field for displacement fields never seen before. We do not enforce a strict bottleneck, rather we allow for skip connections from the encoding process to the decoding process. The skip connections thus offer a way for the network to manipulate the decoding process with selected information gathered during encoding, increasing the capacity of the network.

Understanding the forward problem for traction patches

The forward problem we are trying to solve involves cell tractions on elastic substrates. Thus, essential to the performance of our NN is the accurate interpretation of cell-characteristic deformations of the substrate. Cells generated forces exerted onto the substrate via focal adhesion complexes with sizes in the μm-range and tractions in the range $\mathrm{nN}\,\mu\mathrm{m}^{-2} = \mathrm{kPa}$ (34). Forces are generated by tensing acto-myosin stress-fibers that attach to the focal adhesions and, therefore, have a well-defined direction over a focal adhesion complex. Therefore, typical cellular traction patterns consist of localized *patches*, which can comprise single or several focal adhesion complexes and are anchored to the substrate at positions $\vec{r}_i = (x_i, y_i)^{\mathrm{T}}$. Within these patches tractions have a well-defined in-plane angle γ_i with the x-axis resulting in a traction pattern

$$\vec{t}_i(\vec{r}) = t_i(|\vec{r} - \vec{r}_i|) \begin{pmatrix} \cos\gamma_i \\ \sin\gamma_i \end{pmatrix}. \tag{7}$$

These tractions are applied to circular patches of variable radius R_i at the anchored nodes (see Fig. 5), such that

$$t_i(|\vec{r} - \vec{r}_i|) = \begin{cases} t_{0,i}, & \text{for } |\vec{r} - \vec{r}_i| < R_i \\ 0, & \text{else.} \end{cases} \tag{8}$$

We model typical traction patterns as a linear superposition of traction patches (7) localized at different anchoring points. Within linear elasticity, the resulting displacement pattern is also a linear superposition of all the displacement patterns $\vec{u}_i$ caused by all traction patches i.

For a single traction patch, we solve the forward elastic problem by exploiting the convolution theorem

$$\mathscr{F}\mathscr{T}\{\vec{u}_i\} = \mathscr{F}\mathscr{T}\{G\}\mathscr{F}\mathscr{T}\{\vec{t}_i\} \tag{9}$$

where the Fourier transform of the Green's kernel in polar coordinates ρ and ϕ is known (28)

$$\begin{aligned} \tilde{G}(\rho,\phi) &\equiv \mathscr{F}\mathscr{T}\{G\}(\rho,\phi) \\ &= \frac{2(1+\nu)}{E\rho} \begin{pmatrix} (1-\nu) + \nu\sin^2(\phi) & -\nu\sin(\phi)\cos(\phi) \\ -\nu\sin(\phi)\cos(\phi) & (1-\nu) + \nu\cos^2(\phi) \end{pmatrix}, \end{aligned} \tag{10}$$

and the Fourier transform of the traction spot is given by the Hankel transform

$$\begin{aligned}\mathscr{F}\mathscr{T}\{t_i\}(\rho) &= 2\pi\int_0^\infty \mathrm{d}r\, t_i(r) r J_0(\rho r)\\ &= \frac{2\pi t_{0,i}}{\rho^2}\int_0^{\rho R_i}\mathrm{d}x\, x J_0(x)\\ &= \frac{2\pi t_{0,i} R_i}{\rho} J_1(\rho R_i)\end{aligned} \tag{11}$$

where J_n are the Bessel functions of the first kind.

The Fourier transformed displacement field $\mathscr{F}\mathscr{T}\{\vec{u}\}$ is now accessible and can be converted back to the displacement field by performing the inverse Fourier transform

$$\vec{u}_i(x,y) = \mathscr{F}\mathscr{T}^{-1}\{\mathscr{F}\mathscr{T}\{\vec{u}_i\}(\rho,\phi)\}. \tag{12}$$

The inverse Fourier transform can be performed analytically in polar coordinates centered around the corresponding anchored node with a scaled radial component $\tilde{r}_i \equiv |\vec{r}-\vec{r}_i|/R_i$ and an angle θ with the x-axis,

$$\begin{aligned} u_x^i = \frac{R_i t_{0,i}(1+\nu)}{\pi E}&\left[\left((1-\nu)N_1(\tilde{r}_i)+\nu N_2(\tilde{r}_i,\theta)\right)\cos(\gamma_i)\right.\\ &\left.-\nu N_3(\tilde{r}_i,\theta)\sin(\gamma_i)\right]\end{aligned} \tag{13}$$

$$\begin{aligned} u_y^i = \frac{R_i t_{0,i}(1+\nu)}{\pi E}&\left[\left((1-\nu)N_1(\tilde{r}_i)+\nu N_4(\tilde{r}_i,\theta)\right)\sin(\gamma_i)\right.\\ &\left.-\nu N_3(\tilde{r}_i,\theta)\cos(\gamma_i)\right]\end{aligned} \tag{14}$$

where $N_{1,2,3,4}(\tilde{r}_i,\theta)$ are specific functions, that describe the geometric dependence of the displacement field, and are obtained by explicitly solving the occurring inverse Fourier transforms in the Appendix (see eqs. (22), (23), (24), (25)). Strictly speaking, this analytical solution of the forward elastic problem for a single traction patch anchored at R_i is valid on an infinite substrate. We will neglect finite size effects in the following, and use this analytical solution also on finite substrates. The solution for many traction patches anchored at different points is obtained by linear superposition.

Numerically solving the forward problem

We consider a square substrate of size $L\times L$, in which displacements are analyzed (the total substrate size can be larger). Typical sizes are in the range $L\sim 10-100\mu\mathrm{m}$. We use the size L to non-dimensionalize all length scales: $\bar{u}\equiv u/L$, $\bar{r}\equiv r/L$ and $\bar{R}_i\equiv R_i/L$, such that the substrate in which displacements are observed has unit size always. Typical focal adhesion patch sizes R_i are in the range of several $\mu\mathrm{m}$ (15, 34); in dimensionless units, we take $\bar{R}_i\sim 0.05$ as typical value. The above dimensionless coordinate $\tilde{r}$ remains unchanged by non-dimensionalization. Furthermore, we use the elastic constant E as a scale for the tractions: $\bar{t}\equiv t/E$. Typical hydrogel substrate elastic moduli of $E\sim 10\mathrm{kPa}$ (15) and tractions in the range up to $5\mathrm{kPa}$ (34) imply typical dimensionless tractions up to $\bar{t}\sim 0.5$.

We create an $N\times N$-grid, on which we discretize the solution of eqs. (13) and (14) for a supplied traction patch $\bar{t}_{0,i}$ with direction γ_i and use superposition of the individual patch solutions for all anchored nodes, such that we get the full displacement field for a number of n circular traction patches of variable radius $R_{1,\ldots,n}$.

Architecture of the deep convolutional neural network

We choose to employ a *Unet* structure (35) consisting of an *input* encoder, which extracts and compresses the relevant information from the high dimensional *input* displacement field into a lower dimensional representation. From the lower dimensional and compressed displacement information we inflate the dimensionality again with an encoder, such that we finally receive the representation of a traction field in the *output* of the network, as shown in Fig. 1. The motivation for this choice is the conceptual similarity of image procession tasks such as segmentation, which involve local classification of an image, to the assignment of local "traction labels" to each grid point of the "displacement image".

Characteristic for *Unet* networks are the skip connections that transfer precise location information, which would otherwise be lost throughout the encoding process, from stages of the encoding process to counterparts in the decoding process.

Our network is a fully convolutional NN, where the encoding part is a stack of convolutional blocks, and max-pooling layers, while the symmetric decoding part consists of transposed convolutional layers, skip connections and convolutional

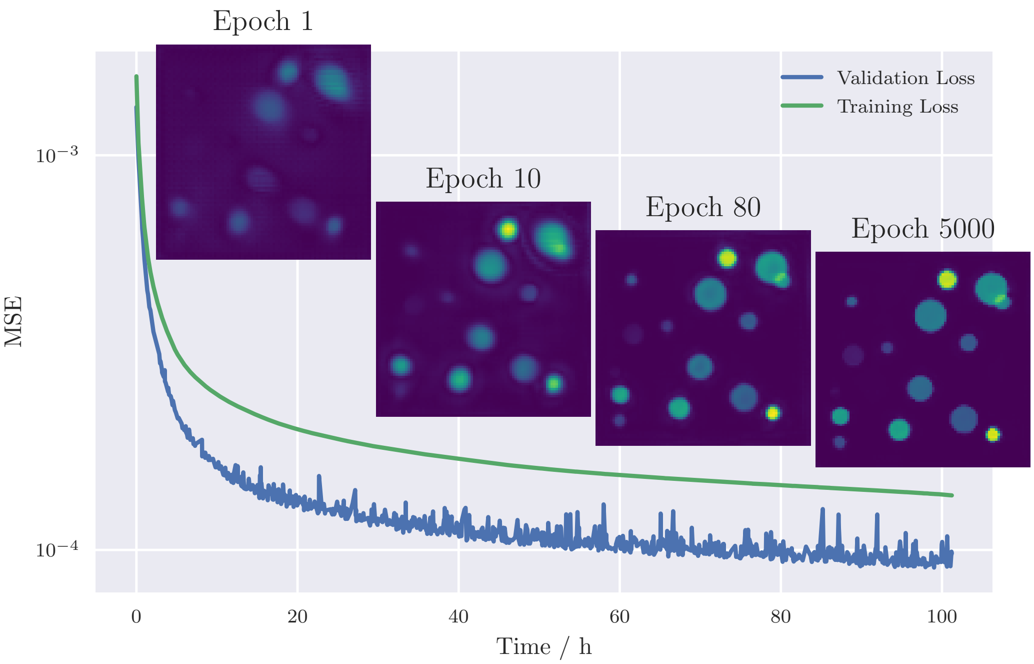

Figure 2: The time evolution of training and validation MSE during training of the neural network. Already after the first epoch a rough traction reconstruction is achieved. After 10 Epochs the reconstruction gains significantly in terms of visual sharpness, which is further increased in the following epochs. We stop the training process after 5000 epochs, because longer training yields diminishing returns.

blocks. Each convolutional block consist of two convolutional layers, with one *Dropout* layer and *LeakyReLU* activation functions, which introduce non-linearity. Specially, the first layer possesses a sigmoid activation function and the last layer uses a linear activation function, ensuring that the output maps to the domain of a traction field. The encoding part uses size 3 kernels and size 1 strides, while the decoding part uses size 4 kernels and size 2 strides to avoid checkerboard effects that would otherwise negatively impact performance.

The entire network structure is implemented using the Keras Python API (36).

Training data sampling and the training process

To train our NN we choose a 104×104-grid ($N = 104$) which holds the discrete representation of the dimensionless displacement and traction fields. The training data is generated by numerically solving the explicit forward problem in dimensionless form as outlined in the section Numerically solving the forward problem.

The traction distribution $\vec{\bar{t}}(x,y)$ is generated by a random number (uniformly sampled in $[10,50]$) of traction spots $\bar{t}_i(x,y)$, where the radius $\bar{R}_i$ is drawn uniformly in the range $[0.02,0.08]$ with a random center point $\{x_i,y_i\} \in [\bar{R}_i + 0.05, 1-\bar{R}_i - 0.05]^2$. The traction magnitude $\bar{t}_{0,i}$ is uniformly distributed in the range $[0,0.5]$ and the polar angle γ_i is uniformly distributed in $[0,2\pi]$.

While traction values and patch sizes are typical for adherent cells, our training data is more general in the sense that other important characteristics of cellular force patters, such as the occurrence of force dipoles at the end of stress fibers, are *not* contained in our training data. This makes our approach more general compared to Ref. (17), where training was performed on traction patters typical for migrating cells. In combination with non-dimensionalization, this will allow us to easily adapt the training process to other applications of TFM in interfacial physics (10) in future applications. Below, we will demonstrate the ability of the CNN to specialize from our general patch-based training set to artificial and real cell data. As convenient model to generate realistic cell traction data artificially we use the contractile network model of Ref. (37).

We expect a NN trained with noisy data to also perform better when confronted with noisy data. To test this hypothesis, we add different levels of background noise to the displacement field $\bar{u}$ in our training data. In order to evaluate the effects on robustness we train two types of NN:

- A network $\mathrm{NN}_{\mathrm{low}}$ is trained with a low level of background noise: To each dimensionless training displacement field value $\vec{\bar{u}}$ a Gaussian noise with a variance $\bar{\sigma}$ that is 0.5% of the average variance of the dimensionless displacement field over all training samples,

$$\bar{\sigma} = 0.005\langle \sigma_{\bar{u}} \rangle, \tag{15}$$

where $\langle ... \rangle$ is an average over all training samples.

- A network $\mathrm{NN_{high}}$ is trained with a high level of background noise that is 5% of the average variance of the dimensionless displacement field over all training samples,

$$\bar{\sigma} = 0.05\langle \sigma_{\bar{u}} \rangle. \tag{16}$$

As a loss or performance metric we use the mean-square error (MSE) between the output guess for tractions and the corresponding labels of the input traction data. We train the network with the MSE

$$\mathrm{MSE} = \frac{1}{M} \sum_{m=1}^{M} \sum_{(x,y)\in \mathrm{grid}} |\vec{t}_{\mathrm{in,m}}(x,y) - \vec{t}_{\mathrm{out,m}}(x,y)|^2 \tag{17}$$

averaged over M training batches as an objective function. We train in batches of $M = 132$ by backpropagation using the Adagrad algorithm.

During training, we evaluate the loss MSE (17) for the training data and a validation MSE for unknown displacement data of the same type. The validation and training errors in Fig. 2 show constant learning and generalization of the model without over-fitting. In total, the training for 5000 epochs took $\sim 100\,\mathrm{h}$ on an *NVIDIA QUADRO RTX 6000* GPU, with the main learning advancements occurring in the first $5\,\mathrm{h}$. Each epoch consists of 25000 randomly chosen traction patch distributions.

BFTTC algorithm

In order to evaluate the performance of our CNN on in comparison to conventional TFM methods, we employ the Bayesian Fourier Transform Traction Cytometry (BFTTC) algorithm as a standard to compare with. The algorithm is described in Refs. (15, 30) and has been made publicly available by the authors at https://github.com/CellMicroMechanics/Easy-to-use_TFM_package.

RESULTS

We analyze the performance of our ML approach on a set of error metrics and in comparison with the BFTTC algorithm (15, 30) as a state-of-the-art conventional TFM method. Importantly, we want to discriminate between background noise and signal while also evaluating magnitude and angle reconstruction precision to infer whether our network generalizes to data never seen before. This will be done first for artificial displacement data for random circular traction patches, which is generated in the same way as the training data and contains an additional varying level of noise. Subsequently, we can evaluate the performance of the CNN on artificial cell data using the same error metrics and, finally, we apply the CNN to real cell data.

Evaluation metrics and application to artificial patch-based data

We employ six evaluation metrics, four of which have already been introduced by Huang and coworkers (15) or earlier, to better distinguish between noise and signal, as well as between bias and resolution. Precise definitions of all evaluation metrics are given in the Appendix. In Fig. 3 we evaluate these metrics for varying noise levels $\tilde{\sigma}$ on our artificial data patch-based data.

The noise applied to the displacement field data is randomly chosen from a Gaussian distribution centered around zero, with standard deviation σ. Let the dimensionless displacement field standard deviation be $\mathrm{std}\bar{u}$, then we define our noise levels $\tilde{\sigma} = \sigma/\mathrm{std}\bar{u}$, such that $\tilde{\sigma}$ is the relative noise applied to the displacement field. In the following considerations we vary $\tilde{\sigma}$ between $1\,\%$ and $10\,\%$.

First, we discuss the Deviation of Traction Magnitude at Adhesions (DTMA) (13, 15) which evaluates the precision of traction magnitude reconstruction at the focal adhesion points (see eq. (26) in the Appendix), thus the DTMA is zero for a perfect reconstruction, negative for an underestimation and positive for an overestimation of traction magnitudes. In Fig. 3 we see similar DTMA scores for all approaches in the limit $\tilde{\sigma} \to 0$, with a systematic under-prediction of tractions. For increasing noise floors both the BFTTC and the low noise network ($\mathrm{NN_{low}}$) depart from this common score and start to overestimate tractions. While DTMA score for the BFTTC method eventually saturates the low noise network DTMA score linearly rises with the noise floor $\tilde{\sigma}$. The high noise network ($\mathrm{NN_{high}}$) retains a comparatively constant DTMA score and always under-predicts the traction magnitude.

Second, we evaluate the Deviation of Traction Magnitude in the Background (DTMB) (15), which quantifies how accurate the traction magnitude reconstruction works in the background (see eq. (27) in the Appendix), thus, if there is no prediction of

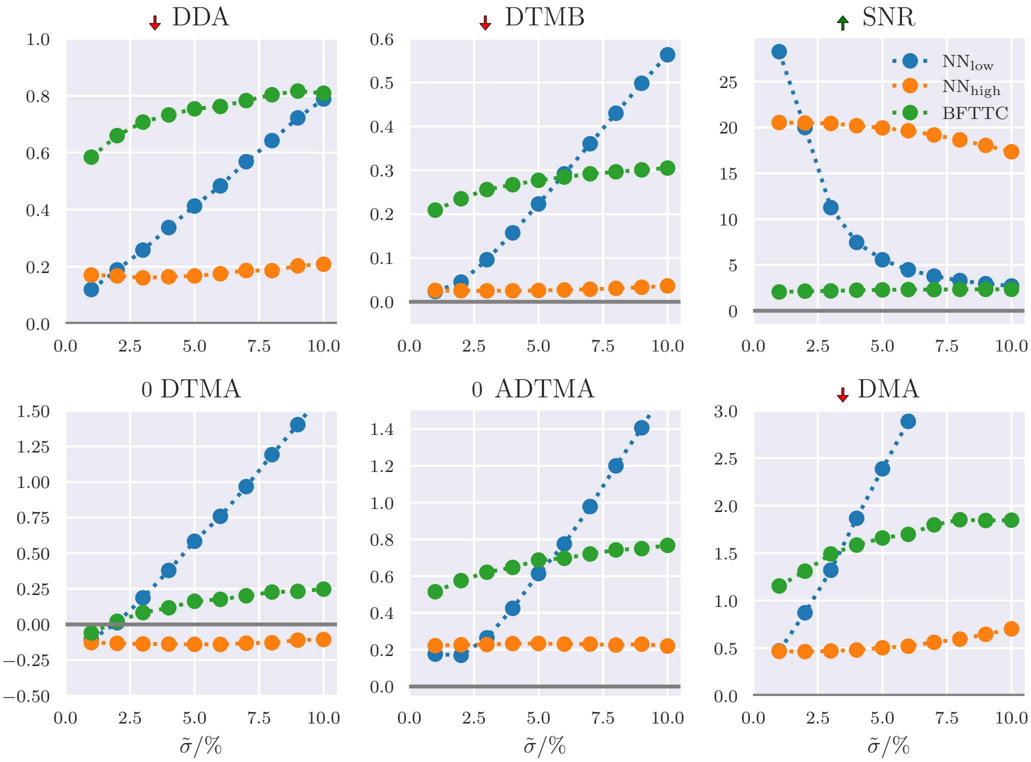

Figure 3: The comparison of our networks $\mathrm{NN}_{\mathrm{low}}$ (trained with low noise background) and $\mathrm{NN}_{\mathrm{high}}$ (trained with high noise background) with a state of the art conventional BFTTC approach shows the precision across the six evaluation metrics (see text) for varying noise levels $\tilde{\sigma}$ on artificial patch-based data. This test is performed with an ensemble of traction spots randomly chosen in count, size, magnitude and orientation, testing our networks performance on data similar to the training data. The arrows next to the metric name indicate whether higher or lower is better; "0" indicates that the metric has a sign and a value of zero is optimal. Both of our networks outperform the BFTTC method in low noise scenarios $\tilde{\sigma} < 5\,\%$ for most metrics. With the high noise network $\mathrm{NN}_{\mathrm{high}}$ we trade low noise fidelity for elevated noise handling capabilities.

an underground noise floor, not associated with any focal adhesion point, the DTMB score will be zero. Both of our NNs have a much lower DTMB score in than the BFTTC method in the limit $\tilde{\sigma} \to 0$, which should manifest in a much less noisy traction force reconstruction. While the low noise network again departs from that score linearly the high noise network again stays comparatively constant.

Third, the Signal to Noise Ratio (SNR) (15) also gives an insight into the noise floor of predictions (see eq. (28) in the Appendix), it is high for a precise distinction between background noise and actual focal adhesion induced deformation and goes to zero for an increasingly noisy reconstruction. Both of our networks have a consistently higher SNR than the BFTTC method, undermining the assumption that the networks will yield a less noisy reconstruction overall. The low noise network SNR decays quickly with increasing noise levels, while the high noise network is more resilient against the increases in noise floor.

Fourth, the Deviation of the Maximum traction at Adhesions (DMA) (15) gives a more detailed insight into the consistency for high amplitude tractions within a focal adhesion point (see eq. (29) in the Appendix). A perfect reconstruction would yield a DMA score of zero, while under-predictions give negative scores and over-predictions positive scores. We can again observe that both of our networks give similar scores for low noise scenarios, which are both lower than the score of the BFTTC method. The low noise network again departs linearly from this common value, while the high noise network DMA score stays at a consistent level.

Fifth, we introduce the Absolute Deviation of Traction Magnitude at Adhesions (ADTMA) that is similar to the DTMA, but evaluates the absolute deviations, capturing the actual reliability of reconstructions more precisely than DTMA, since alternating under- and over-predictions do not cancel out in this score. We can again see the same qualitative behavior as in most of the other metrics: Both networks are more precise in the low noise region and the low noise network gets linearly

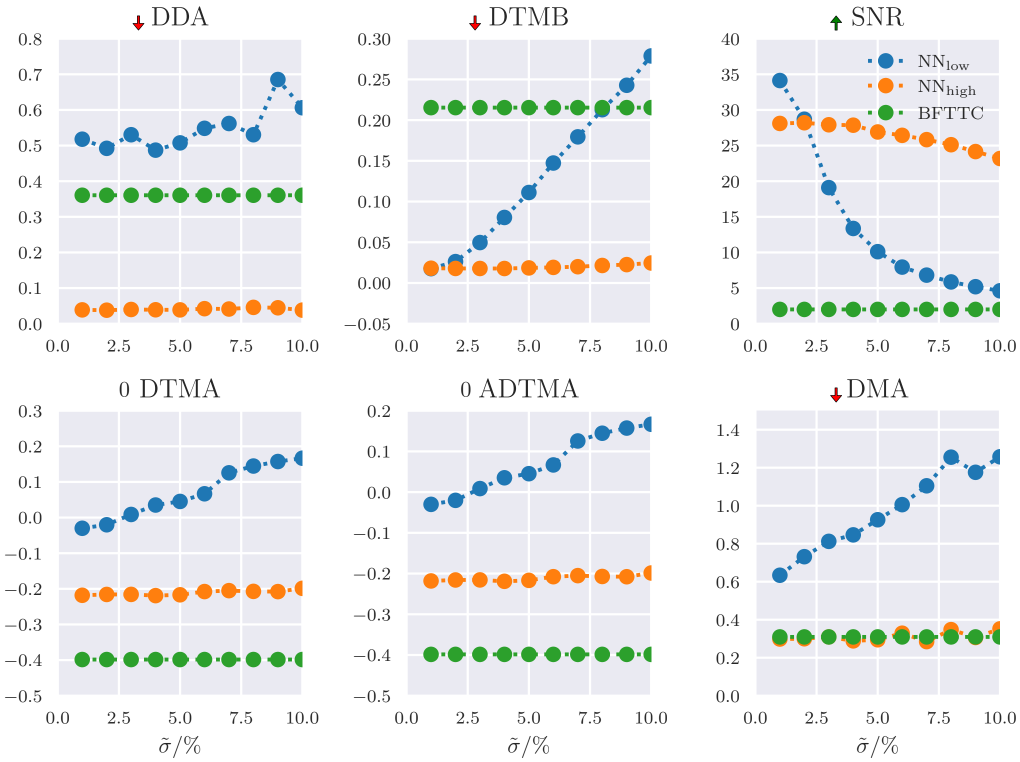

Figure 4: We compare our networks NN$_{\text{low}}$ and NN$_{\text{high}}$ with the BFTTC method for data generated from an artificial cell model using the same six evaluation metrics for varying noise levels $\tilde{\sigma}$ as in Fig. 3. Since the method of generating the data is now different from the training process, we expect to see the generalization potential of our networks more clearly than in Fig. 3. The low noise network still performs well on this data, however, especially in the DDA score we see a worse performance compared to Fig. 3. Additionally, we see an amplified underestimation of the traction magnitudes for the high noise network and the BFTTC method. With respect to noise, the high noise network seems to give the best compromise between precision and regularization of the output traction fields.

worse, while the high noise network is robust against increases in background noise.

Finally, we introduce a measure to more precisely quantify the orientation resolution via the Deviation of Traction Direction at Adhesions (DDA)

$$\mathrm{DDA} = \langle p(\gamma^{\mathrm{pre}}, \gamma^{\mathrm{tru}})^2 \rangle \tag{18}$$

between predicted and true traction angles γ (see eq. (31) in the Appendix for a more precise definition of the average); $p(\alpha, \beta)$ measures the *periodic* distance between two angles α and β. A small DDA indicates precise traction direction reconstruction. For both of our networks the direction reconstruction is more precise than the BFTTC method across the range of tested noise levels.

Across all six measures we observe the following trends for network NN$_{\text{low}}$ trained with a low level of background noise as compared to network NN$_{\text{high}}$ trained with a high level of background noise: NN$_{\text{low}}$ perform superior to the BFTTC-standard *and* NN$_{\text{high}}$ for low-noise data, because they are trained on low noise data. Their performance deteriorates, however, for higher noise levels $\tilde{\sigma}$, where their performance drops below NN$_{\text{high}}$ but also below the BFTTC-standard. NN$_{\text{high}}$ finds a better compromise between robustness and accuracy such that it outperforms the BFTTC-standard across *all* noise levels.

Tractions of artificial cells

In order to test the ability of the CNN to specialize from our general patch-based training set to realistic cell data, we first test the model on artificial cell data. The advantage of artificial cell data is that the *"true"* tractions are precisely known. A convenient model to generate realistic cell traction data artificially is the contractile network model (37). In this model, the stress fibers are active cable links with specific nodes anchored to the substrate at positions $\vec{r}_i = (x_i, y_i)^{\mathrm{T}}$. To construct a typical

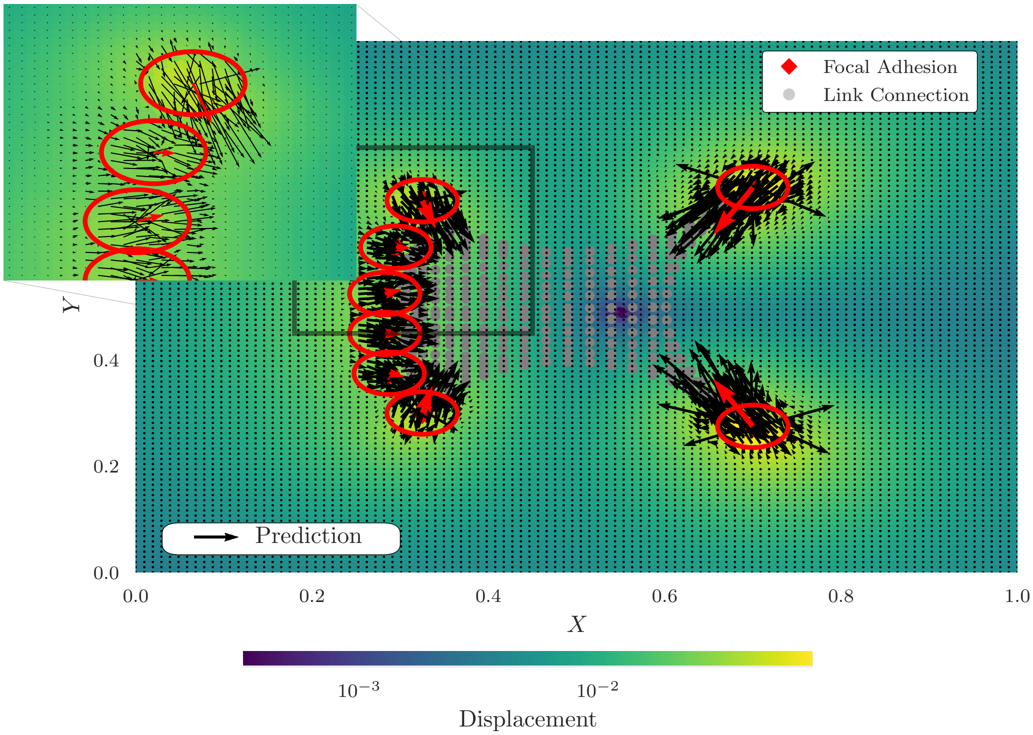

Figure 5: A model cell with circular focal adhesion points. The model cell perturbs the elastic substrate it is resting on by generating tractions (red arrows) at the focal adhesion spots (red circles), resulting in the color coded displacement field; tractions (red arrows) are generated based on a contractile network model (see text). Read arrows are the "true" average tractions generated by the cell model over the red circles, while the black arrows indicate the local tractions that the $\mathrm{NN}_{\mathrm{low}}$ network predicts at the discrete grid spots.

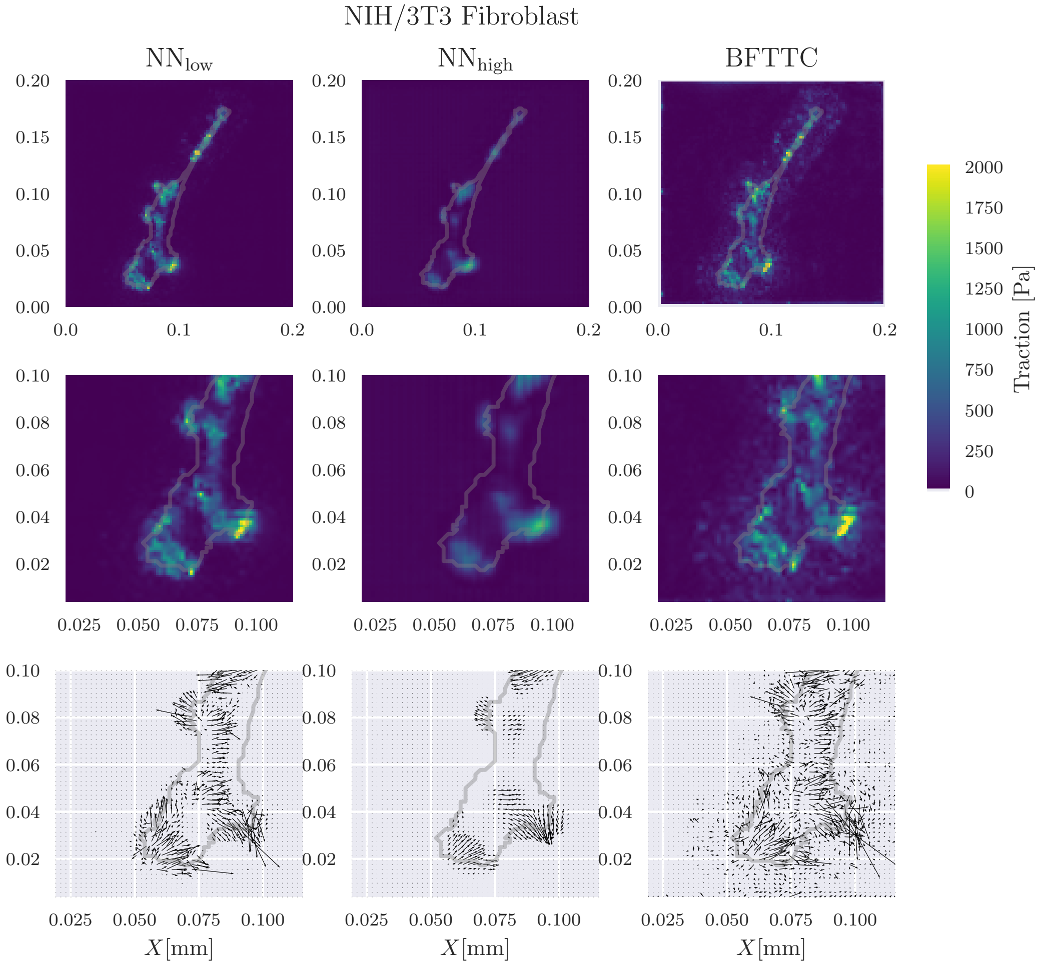

Figure 6: Comparison of traction reconstruction for a real Fibroblast between our networks NN$_{\text{low}}$ and NN$_{\text{high}}$ and the BFTTC method. Although we do not have the "correct" traction field at hand for a quantitative evaluation of precision across the methods, we see compatible results across the board, while both networks have a significantly reduced noise floor.

cell shape with lamellipodium and tail focal adhesions the tractions $\vec{t}_i(\vec{r})$ generated at each anchor point are generated by minimizing the total energy of the cable network. These tractions are then applied to circular patches of radius $R = 0.04$ under an angle γ_i given by the stress fiber orientation at the anchored nodes (see Fig. 5).

In addition to Fig. 3, we want to evaluate the aforementioned metrics (DDA, ADTMA, DTMB, DMA, DTMA, SNR) on a displacement field generated by an artificial cell as shown in Fig. 5. Again, we add Gaussian noise to the displacement field with varying noise levels $\tilde{\sigma}$ and evaluate the behavior of our networks NN$_{\text{low}}$, NN$_{\text{high}}$, and the BFTTC method in Fig. 4.

We see qualitatively similar results to Fig. 3 in the SNR and DTMB metrics, while the absolute performance in those metrics is better (higher SNR and lower DTMB scores) for the artificial cell data. This is likely due to the lower number of traction patches in total and the equal radii of all traction patches. In the DDA score we can see that our low noise network is not able to reconstruct the direction of tractions robustly. The high noise network as well as the BFTTC method are able to reconstruct the directions more reliably.

The artificial cell data show a clear tendency towards a traction magnitude underestimation (DTMA, ADTMA) for both our high noise network and the BFTTC method. Since we are generating tractions in strongly bounded range due to the cable network, the artifacts of the strong regularization in the BFTTC method and the high noise network begin to show more clearly than in Fig. 3.

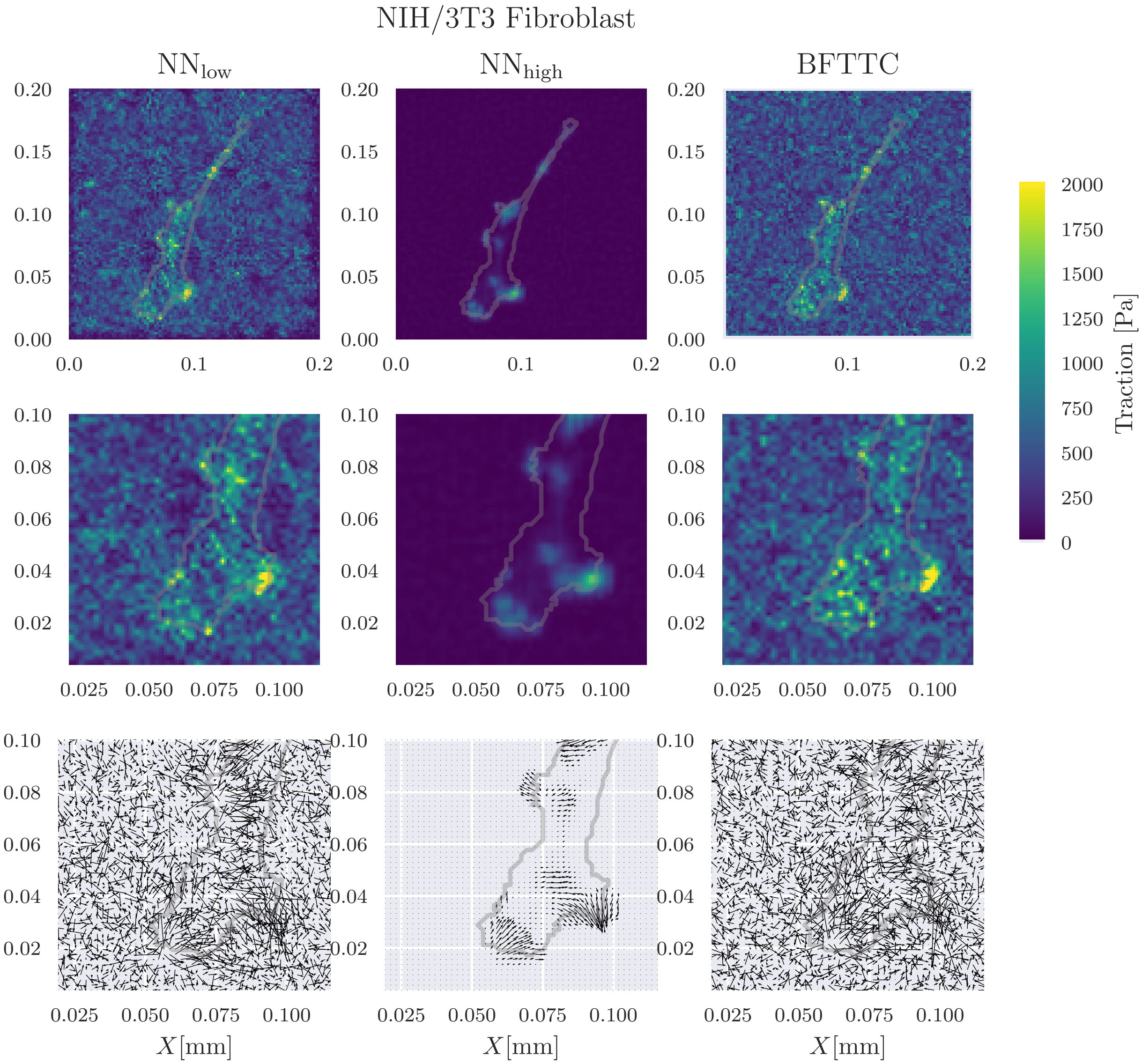

Figure 7: Adding noise to the Fibroblast displacement field shows strong noise robustness of our network $\mathrm{NN_{high}}$, which has been trained for high noise scenarios. The low noise network $\mathrm{NN_{low}}$ and the BFTTC method again yield qualitatively similar results but exhibit strong noise artifacts in the reconstructed displacement field.

Tractions of real cells

Finally, we want to test our ML approach on real cell images. Of course, we do not have access to the *"true"* traction field for those images, however, we can qualitatively compare the results obtained from our ML approach with those obtained by the well tested BFTTC method. The example cell results shown in Fig. 6 are for a NIH/3T3 (National Institutes of Health 3T3 cultivated) Fibroblast on a substrate with length $L = 200.2\,\mu\mathrm{m}$ and elastic modulus $E = 10670\,\mathrm{Pa}$ (the data was made available in Ref. (17)). It is apparent that the network trained with low noise ($\mathrm{NN}_{\mathrm{low}}$) reconstructs a traction field, which is similar to that of the BFTTC method, while the noise in the vicinity of the cell is significantly reduced. The network trained with a high noise floor gives a more regular traction pattern and cuts off lower amplitude tractions.

The reason for these results is that the network trained with low noise exhibits a SNR superior both to BFTTC and the network trained with high noise levels if noise in the experimental data is low (see Figs. 3 and 4); for artificial cell data also the DTMA and ADTMA of the network trained with low noise is superior for low experimental noise levels (see Fig. 4). A low noise in experimental data seems to be realized here. We can thus infer that the tractions reconstructed by the high noise network systematically under-predict the real tractions for this particular data.

As expected from our prior analysis, the resistance to additional noise is much better for the network which saw high noise levels during training and the low noise network is visually less noisy than the BFTTC method, while retaining the capability to reconstruct small tractions. The robustness of the high noise network is highlighted in Fig. 7, where the cell data is superposed with significant background noise of $\tilde{\sigma} = 10\,\%$. Both our low noise network and the BFTTC method produce a noisy traction field in this case, while the high noise network still displays a similar traction pattern in both of these cases.

Overall, the result from the low noise network and the BFTTC method is more detailed and fragmented, while the high noise network reconstruction is more regular and fits the circular focal adhesion point model more consistently and stays invariant for a wide range of background noise levels. Apparently, the noise the network sees in training directly controls the regularization of the reconstruction and it would be possible to create intermediate networks with higher sensitivity, but lower noise invariance.

These two networks could thus be employed for different experimental scenarios: When the experimental error is small, and the displacement field can be accurately reconstructed the low noise network produces superior results to the BFTTC method, while performing an order of magnitude faster on regular hardware. If the experimental error is high, or the displacement field reconstruction is imprecise the high noise network provides a robust way of extracting the traction field, while conventional methods are plagued by high background noise in this case.

DISCUSSION

We present a ML approach to TFM via a deep convolutional NN trained on a general set of artificial displacement-traction data derived from the analytic solution of the elastic forward problem for random ensembles of circular traction patches. We find a significant enhancement of the robustness of the NN if the training data is subjected to an appropriate level of additional noise. This provides a computationally efficient way to accelerate traction force microscopy as a method and improves both on accuracy and noise resilience of conventional approaches, while reducing the computational complexity, and thus execution time by an order of magnitude compared to state-of-the-art conventional approaches. It is apparent from our analysis that ML approaches have the potential to shift the paradigm in solving inverse problems away from conventional iterative methods, towards educated regressors which are trained on a well understood and numerically simple to solve forward problem.

In conventional TFM approaches the inverse elastic problem is ill-posed and the suitable choice of regularization in the inversion procedure is crucial and has been a topic of active research over the last twenty years. ML approaches circumvent the need for explicit regularization and provide an implicit regularization by a proper choice of the network architecture, i.e., convolutional NNs for TFM, and after proper training. Our work shows that the suitable choice of "physics-informed" training data and, moreover, the suitable choice of noise on the training data governs the applicability of the NN and the compromise between accuracy and robustness in ML approaches, somewhat analogous to the role of the regularization procedure in conventional TFM approaches.

We employ a sufficiently general patch-based training set and show that this allows the CNN to successfully specialize to artificial cell data and real cell data. Moreover, training with an additional background noise that is 5% of the average variance of the dimensionless displacement field (the NN_{high} network (16)), gives a robustness against noise in the NN performance that is superior to state-of-the-art conventional TFM approaches without significantly compromising accuracy. We can systematically back these claims by characterizing both the NN performance and the performance of state-of-the-art conventional TFM (the BFTTC method) via six error metrics both for the patch-based training set (Fig. 3) and the artificial cell data set (Fig. 4), which are two data sets where we can compare the prediction to the true traction labels.

For real cell data, we find that a NN trained with low noise (0.5%) gives the best performance if the experimental data is of high quality with low noise levels (see Fig. 6). For noisy experimental data, on the other hand, the NN trained with high levels of noise (5%) clearly performs best (see Fig. 7). This suggests that it might be beneficial to employ both types of NNs depending on the experimental scenario.

We make all NNs discussed in this work freely available for further use in TFM. We use a 104 × 104-grid for the displacement data. Experimental data will typically provide even higher resolution but can be easily adapted by interpolation to the 104 × 104-grid. By using non-dimensionalized units, the NNs made available with this work are widely applicable across different problems and can also be easily further adapted, for example, to problems where typical tractions are not limited to the range $t < 0.5E$ that we have assumed here by repeating the training process. All necessary routines to re-train a NN to new traction levels, new characteristic patch sizes, or other noise levels are made freely available with this work at https://gitlab.tu-dortmund.de/cmt/kierfeld/mltfm.

AUTHOR CONTRIBUTIONS

F.S.K., L.M. and J.K. developed the theoretical and computational model. F.S.K. and L.M. performed all computations, construction and training of neural networks, and data analysis. F.S.K. and J.K. wrote the original draft. F.S.K., L.M. and J.K. reviewed and edited the manuscript.

DECLARATION OF INTEREST

The authors declare no competing interests.

ACKNOWLEDGMENTS

F.S.K. acknowledges support by the German Academic Scholarship Foundation.

REDUCTION OF THE ELASTIC GREEN'S TENSOR TO TWO DIMENSIONS

The full three-dimensional Green's tensor of the boundary problem (1), (2) on the half-space $z > 0$ is given by (11, 27):

$$\mathbf{G}(x,y,z) = \frac{1+\nu}{2\pi E}\begin{pmatrix} \frac{2(1-\nu)r+z}{r(r+z)} + \frac{(2r(\nu r+z)+z^2)x^2}{r^3(r+z)^2} & \frac{(2r(\nu r+z)+z^2)xy}{r^3(r+z)^2} & \frac{xz}{r^3} - \frac{(1-2\nu)x}{r(r+z)} \\ \frac{(2r(\nu r+z)+z^2)xy^2}{r^3(r+z)^2} & \frac{2(1-\nu)r+z}{r(r+z)} + \frac{(2r(\nu r+z)+z^2)y^2}{r^3(r+z)^2} & \frac{yz}{r^3} - \frac{(1-2\nu)y}{r(r+z)} \\ \frac{xz}{r^3} + \frac{(1-2\nu)x}{r(r+z)} & \frac{yz}{r^3} + \frac{(1-2\nu)y}{r(r+z)} & \frac{2(1-\nu)}{r} + \frac{z^2}{r^3} \end{pmatrix}, \tag{19}$$

The tensor is not symmetric, in general. On the surface $z = 0$, this becomes

$$\mathbf{G}(x,y,0) = \frac{1+\nu}{\pi E r^3}\begin{pmatrix} (1-\nu)r^2+\nu x^2 & \nu xy & -\frac{1}{2}(1-2\nu)xr \\ \nu xy & (1-\nu)r^2+\nu y^2 & -\frac{1}{2}(1-2\nu)yr \\ \frac{1}{2}(1-2\nu)xr & \frac{1}{2}(1-2\nu)yr & (1-\nu)r^2 \end{pmatrix}. \tag{20}$$

For incompressible polymer materials with $\nu \approx 1/2$, the coupling between in-plane tractions and out-of-plane displacements is small.

DETAILS OF THE SOLUTION OF THE FORWARD PROBLEM FOR TRACTION PATCHES

For the solution of the forward problem for a tractions patch of size R_i, we obtained the functions $N_j(\tilde{r}_i,\theta)$ in (13) and (14). They are defined as

$$N_j(\tilde{r}_i,\theta) \equiv \int_0^{2\pi} d\phi \int_0^{\infty} d\rho \frac{J_1(\rho R_i)}{\rho} f_j(\phi) e^{-i\rho R_i \tilde{r}_i \cos(\phi-\theta)} \tag{21}$$

with

$$\begin{aligned} f_1(\phi) &\equiv 1 \\ f_2(\phi) &\equiv \sin^2\phi \\ f_3(\phi) &\equiv \sin\phi\cos\phi \\ f_4(\phi) &\equiv \cos^2\phi. \end{aligned}$$

By definition, $N_2 + N_4 = N_1$.

The remaining integrals in the functions $N_j(\tilde{r}_i, \theta)$ can be performed analytically:

$$\begin{aligned} N_1(\tilde{r}_i) &= 2\pi \int_0^\infty dx \frac{J_1(x)J_0(x\tilde{r}_i)}{x} \\ &= \begin{cases} 4\mathrm{E}\left(\tilde{r}_i^2\right) & \tilde{r}_i < 1 \\ 4\left[\tilde{r}_i \mathrm{E}\left(\tilde{r}_i^{-2}\right) - (\tilde{r}_i - \tilde{r}_i^{-1})\mathrm{K}\left(\tilde{r}_i^{-2}\right)\right] & \tilde{r}_i > 1 \end{cases} \end{aligned} \tag{22}$$

with the complete elliptic integrals $E(m) = \int_0^{\pi/2} (1 - m\sin^2\theta)^{1/2} d\theta$ and $K(m) = \int_0^{\pi/2} (1 - m\sin^2\theta)^{-1/2} d\theta$;

$$\begin{aligned} N_2(\tilde{r}_i, \theta) &= 2\pi \int_0^\infty dx \left(\frac{J_1(x)J_1(x\tilde{r}_i)}{x^2 \tilde{r}_i} - \sin^2\theta \frac{J_1(x)J_2(x\tilde{r}_i)}{x} \right) \\ &= \begin{cases} 4\sin^2\theta \mathrm{E}\left(\tilde{r}_i^2\right) + \frac{4}{3\tilde{r}_i^2}\cos(2\theta)\left[(1+\tilde{r}_i^2)\mathrm{E}\left(\tilde{r}_i^2\right) + (1-\tilde{r}_i^2)\mathrm{K}\left(\tilde{r}_i^2\right)\right] & \tilde{r}_i < 1 \\ \frac{2}{3\tilde{r}_i}\left[\left(3\tilde{r}_i^2 + (2-\tilde{r}_i^2)\cos(2\theta)\right)\mathrm{E}\left(\tilde{r}_i^{-2}\right) + (1-\tilde{r}_i^2)(3-\cos(2\theta))\mathrm{K}\left(\tilde{r}_i^{-2}\right)\right] & \tilde{r}_i > 1 \end{cases}; \end{aligned} \tag{23}$$

$$\begin{aligned} N_3(\tilde{r}_i, \theta) &= -2\pi \sin\theta\cos\theta \int_0^\infty dx \frac{J_1(x)J_2(x\tilde{r}_i)}{x} \\ &= \begin{cases} \frac{2}{3\tilde{r}_i^2}\sin(2\theta)\left[\left(\tilde{r}_i^2 - 2\right)\mathrm{E}\left(\tilde{r}_i^2\right) + 2(1-\tilde{r}_i^2)\mathrm{K}\left(\tilde{r}_i^2\right)\right] & \tilde{r}_i < 1 \\ \frac{2}{3\tilde{r}_i}\sin(2\theta)\left[\left(\tilde{r}_i^2 - 2\right)\mathrm{E}\left(\tilde{r}_i^{-2}\right) + (1-\tilde{r}_i^2)\mathrm{K}\left(\tilde{r}_i^{-2}\right)\right] & \tilde{r}_i > 1 \end{cases}; \end{aligned} \tag{24}$$

$$N_4(\tilde{r}_i, \theta) = N_1(\tilde{r}_i) - N_2(\tilde{r}_i, \theta). \tag{25}$$

DEFINITIONS OF EVALUATION METRICS

We employ six evaluation metrics (see Figs. 3 and 4). Their definition is based on a comparison of traction predictions $\vec{t}^{\text{pre,s}}(x,y)$ in sample s compared to "true" tractions $\vec{t}^{\text{tru,s}}(x,y)$, which are known for the artificial data for random circular traction patches. We evaluate all six metrics by averaging over $S = 100$ samples; the sample average is denoted by $\langle ... \rangle$. All traction vectors $\vec{t}_{i,v}$ in patch i ($i = 1, ..., n$) are indexed by v. All tractions vectors $\vec{t}_{b,w}$ outside patches are considered as belonging to the background b and indexed by w. For completeness we give the precise definitions of all six evaluation metrics:

1. Deviation of Traction Magnitude at Adhesions (DTMA) (15):

$$\mathrm{DTMA} = \left\langle \mathrm{mean}_i \frac{\mathrm{mean}_v\left(|\vec{t}_{i,v}^{\text{pre,s}}| - |\vec{t}_{i,v}^{\text{tru,s}}|\right)}{\mathrm{mean}_v\left(|\vec{t}_{i,v}^{\text{tru,s}}|\right)} \right\rangle. \tag{26}$$

Note that $\mathrm{mean}_v\left(|\vec{t}_{i,v}^{\text{tru,s}}|\right) = |\vec{t}_{i,v}^{\text{tru,s}}|$ because artificial traction data is piecewise constant in traction patches.

2. Deviation of Traction Magnitude in the Background (DTMB) (15):

$$\mathrm{DTMB} = \left\langle \frac{\mathrm{mean}_w\left(|\vec{t}_{b,w}^{\text{pre,s}}| - |\vec{t}_{b,w}^{\text{tru,s}}|\right)}{\mathrm{mean}_i\mathrm{mean}_v\left(|\vec{t}_{i,v}^{\text{tru,s}}|\right)} \right\rangle. \tag{27}$$

Note that $|\vec{t}_{b,w}^{\text{tru,s}}| = 0$ because artificial traction data exactly vanishes outside patches.

3. Signal to Noise Ratio (SNR) (15):

$$\mathrm{SNR} = \left\langle \frac{\mathrm{mean}_i\mathrm{mean}_v\left(|\vec{t}_{i,v}^{\text{pre,s}}|\right)}{\mathrm{std}_w\left(\vec{t}_{b,w}^{\text{tru,s}}\right)} \right\rangle, \tag{28}$$

where std is the standard deviation.

4. Deviation of the Maximum traction at Adhesions (DMA) (15):

$$\mathrm{DMA} = \left\langle \mathrm{mean}_i \frac{\max_v \left(|\vec{t}^{\,\mathrm{pre,s}}_{i,v}| \right) - \max_v \left(|\vec{t}^{\,\mathrm{tru,s}}_{i,v}| \right)}{\mathrm{mean}_v \left(|\vec{t}^{\,\mathrm{tru,s}}_{i,v}| \right)} \right\rangle . \tag{29}$$

   Note that $\max_v \left(|\vec{t}^{\,\mathrm{tru,s}}_{i,v}| \right) = |\vec{t}^{\,\mathrm{tru,s}}_{i,v}|$ because artificial traction data is piecewise constant in traction patches.

5. Absolute Deviation of Traction Magnitude at Adhesions (ADTMA):

$$\mathrm{ADTMA} = \left\langle \mathrm{mean}_i \left| \frac{\mathrm{mean}_v \left(|\vec{t}^{\,\mathrm{pre,s}}_{i,v}| - |\vec{t}^{\,\mathrm{tru,s}}_{i,v}| \right)}{\mathrm{mean}_v \left(|\vec{t}^{\,\mathrm{tru,s}}_{i,v}| \right)} \right| \right\rangle . \tag{30}$$

6. Deviation of Traction Direction at Adhesions (DDA):

$$\mathrm{DDA} = \left\langle \mathrm{mean}_i \mathrm{mean}_v p^2 (\gamma^{\mathrm{pre,s}}_{\mathrm{i,v}}, \gamma^{\mathrm{tru,s}}_{\mathrm{i,v}}) \right\rangle , \tag{31}$$

   where $p(\alpha, \beta)$ measures the *periodic* distance between two angles α and β.

REFERENCES

1. Lecuit, T., and P. F. Lenne, 2007. Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis. *Nat. Rev. Mol. Cell Biol.* 8:633–644.

2. Vogel, V., and M. Sheetz, 2006. Local force and geometry sensing regulate cell functions. *Nat. Rev. Mol. Cell Biol.* 7:265–275.

3. Jaalouk, D. E., and J. Lammerding, 2009. Mechanotransduction gone awry. *Nat. Rev. Mol. Cell Biol.* 10:63–73.

4. Hahn, C., and M. A. Schwartz, 2009. Mechanotransduction in vascular physiology and atherogenesis. *Nat. Rev. Mol. Cell Biol.* 10:53–62.

5. Vollrath, M. A., K. Y. Kwan, and D. P. Corey, 2007. The micromachinery of mechanotransduction in hair cells. *Annu. Rev. Neurosci.* 30:339–365.

6. Yu, H., J. K. Mouw, and V. M. Weaver, 2011. Forcing form and function: biomechanical regulation of tumor evolution. *Trends Cell Biol.* 21:47–56.

7. Plotnikov, S. V., B. Sabass, U. S. Schwarz, and C. M. Waterman, 2014. High-Resolution Traction Force Microscopy. *In* J. C. Waters, and T. Wittman, editors, Quantitative Imaging in Cell Biology, Academic Press, volume 123 of *Methods in Cell Biology*, 367–394.

8. Schwarz, U. S., and J. R. Soiné, 2015. Traction force microscopy on soft elastic substrates: A guide to recent computational advances. *Biochim. Biophys. Acta - Mol. Cell Res.* 1853:3095–3104.

9. Lekka, M., K. Gnanachandran, A. Kubiak, T. Zieliński, and J. Zemła, 2021. Traction force microscopy – Measuring the forces exerted by cells. *Micron* 150:103138.

10. Style, R. W., R. Boltyanskiy, G. K. German, C. Hyland, C. W. MacMinn, A. F. Mertz, L. A. Wilen, Y. Xu, and E. R. Dufresne, 2014. Traction force microscopy in physics and biology. *Soft Matter* 10:4047–4055.

11. Dembo, M., and Y. L. Wang, 1999. Stresses at the cell-to-substrate interface during locomotion of fibroblasts. *Biophys. J.* 76:2307–2316.

12. Schwarz, U., N. Balaban, D. Riveline, A. Bershadsky, B. Geiger, and S. Safran, 2002. Calculation of Forces at Focal Adhesions from Elastic Substrate Data: The Effect of Localized Force and the Need for Regularization. *Biophys. J.* 83:1380–1394.

13. Sabass, B., M. L. Gardel, C. M. Waterman, and U. S. Schwarz, 2008. High resolution traction force microscopy based on experimental and computational advances. *Biophys. J.* 94:207–220.

14. Kulkarni, A. H., P. Ghosh, A. Seetharaman, P. Kondaiah, and N. Gundiah, 2018. Traction cytometry: regularization in the Fourier approach and comparisons with finite element method. *Soft Matter* 14:4687–4695.

15. Huang, Y., C. Schell, T. B. Huber, A. N. Şimşek, N. Hersch, R. Merkel, G. Gompper, and B. Sabass, 2019. Traction force microscopy with optimized regularization and automated Bayesian parameter selection for comparing cells. *Sci. Rep.* 9:539.

16. Kratz, F. S., and J. Kierfeld, 2020. Pendant drop tensiometry: A machine learning approach. *J. Chem. Phys.* 153:094102.

17. Wang, Y.-l., and Y.-C. Lin, 2021. Traction force microscopy by deep learning. *Biophys. J.* 120:3079–3090.

18. Li, H., D. Matsunaga, T. S. Matsui, H. Aosaki, G. Kinoshita, K. Inoue, A. Doostmohammadi, and S. Deguchi, 2022. Wrinkle force microscopy: a machine learning based approach to predict cell mechanics from images. *Commun. Biol.* 5:361.

19. Harris, A., P. Wild, and D. Stopak, 1980. Silicone rubber substrata: a new wrinkle in the study of cell locomotion. *Science* 208:177–179.

20. Dembo, M., T. Oliver, A. Ishihara, and K. Jacobson, 1996. Imaging the traction stresses exerted by locomoting cells with the elastic substratum method. *Biophys. J.* 70:2008–2022.

21. Tan, J. L., J. Tien, D. M. Pirone, D. S. Gray, K. Bhadriraju, and C. S. Chen, 2003. Cells lying on a bed of microneedles: An approach to isolate mechanical force. *Proc. Natl. Acad. Sci. U. S. A.* 100:1484–1489.

22. Brockman, J. M., A. T. Blanchard, V. Pui-Yan, W. D. Derricotte, Y. Zhang, M. E. Fay, W. A. Lam, F. A. Evangelista, A. L. Mattheyses, and K. Salaita, 2018. Mapping the 3D orientation of piconewton integrin traction forces. *Nat. Methods* 15:115–118.

23. Steinwachs, J., C. Metzner, K. Skodzek, N. Lang, I. Thievessen, C. Mark, S. Münster, K. E. Aifantis, and B. Fabry, 2016. Three-dimensional force microscopy of cells in biopolymer networks. *Nat. Methods* 13:171–176.

24. Beningo, K. A., and Y.-L. Wang, 2002. Flexible substrata for the detection of cellular traction forces. *Trends Cell Biol.* 12:79–84.

25. Schwarz, U. S., and S. A. Safran, 2013. Physics of adherent cells. *Rev. Mod. Phys.* 85:1327–1381.

26. Newby, J. M., A. M. Schaefer, P. T. Lee, M. G. Forest, and S. K. Lai, 2018. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2D and 3D. *Proc. Natl. Acad. Sci. U. S. A.* 115:9026–9031.

27. Landau, L.D. and Lifshitz, E.M. and Kosevitch, A.M. and Pitaevskiĭ, L.P., 1986. Theory of Elasticity. Course of theoretical physics. Pergamon Press, New York.

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

29. Han, S. J., Y. Oak, A. Groisman, and G. Danuser, 2015. Traction microscopy to identify force modulation in subresolution adhesions. *Nat. Methods* 12:653–656.

30. Huang, Y., G. Gompper, and B. Sabass, 2020. A Bayesian traction force microscopy method with automated denoising in a user-friendly software package. *Comput. Phys. Commun.* 256:107313.

31. Soiné, J. R. D., N. Hersch, G. Dreissen, N. Hampe, B. Hoffmann, R. Merkel, and U. S. Schwarz, 2016. Measuring cellular traction forces on non-planar substrates. *Interface Focus* 6:20160024.

32. Adler, J., and O. Öktem, 2017. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* 33:124007.

33. Xu, L., J. Ren, C. Liu, and J. Jia, 2014. Deep convolutional neural network for image deconvolution. *Advances in Neural Information Processing Systems* 2:1790–1798.

34. Balaban, N. Q., U. S. Schwarz, D. Riveline, P. Goichberg, G. Tzur, I. Sabanay, D. Mahalu, S. Safran, A. Bershadsky, L. Addadi, and B. Geiger, 2001. Force and focal adhesion assembly: a close relationship studied using elastic micropatterned substrates. *Nat. Cell Biol.* 3:466–472.

35. Ronneberger, O., P. Fischer, and T. Brox, 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. volume 9351, 234–241.

36. Chollet, F., et al., 2015. Keras. https://keras.io.

37. Guthardt Torres, P., I. B. Bischofs, and U. S. Schwarz, 2012. Contractile network models for adherent cells. *Phys. Rev. E* 85:011913.