Deep convolutional neural networks allow analysis of cell motility during stem cell differentiation and neoplastic transformation

Jacob C Kimmel, Andrew S Brack, and Wallace F Marshall,

***Abstract*—Cells in culture display diverse motility behaviors. In multiple contexts, cell motility reflects broader cell function, providing motivation to discriminate between different motility behaviors. Current methods to do so rely upon manual feature engineering. However, the types of features necessary to distinguish between motility behaviors can vary greatly depending on the biological context, and it is not always clear which features may be most predictive in each setting for distinguishing particular cell types or disease states. Convolutional neural networks (CNNs) are a class of machine learning models ideally suited to the analysis of spatial data, allowing for relevant spatial features to be learned as parameters of a model. Given that motility data is inherently spatial, CNNs are a promising approach to learn relevant features for motility analysis from data, rather than requiring a domain expert to engineer features by hand. Here, we apply CNNs to classify different motility behaviors by representing motility as a 3D space with markers denoting a cell's location at each time point. 3D CNNs provide accurate classification of several simulated motility behaviors, the motility behaviors of multiple cell types, and characteristic motility behaviors of commitment states in myogenic cells. Autoencoders were trained effectively to learn representations of these 3D motility spaces in an unsupervised manner. We show that this approach can achieve reliable detection of differentiation state for muscle stem cells and better-than-chance detection of neoplastic transformation in a cancer cell model. The variety of cell type differences we can detect suggests that the algorithm is generally applicable to novel cell types. While we have applied these methods to the analysis of cell motility, our scheme for representing motion spatially for analysis by CNNs is generalizable to other motion classification problems.**

***Index Terms*—convolutional neural network, cell motility, cell classification.**

I. Introduction

CELL motility is a diverse cellular behavior involving a complex regulatory network and dynamic reorganization of the cell's geometry [1], [2]. These motility behaviors can provide a useful window for inference of a cell's broader functionality. Neoplastic transformation has long been appreciated to alter cell motility behaviors, increasing the migration rate of various models in culture and serving as a mechanism for metastasis [3], [4], [5], [6], [7]. The motility behaviors of cancer cells in culture can even be predictive of broader tumor progression [8].

Likewise, the migration of progenitor cells is critical in early development and tissue regeneration [9]. Skeletal muscle stem cells (MuSCs) provide an accessible platform to study stem cell motility phenotypes *in vitro* by timelapse imaging. During embryonic development, MuSC precursors must migrate from early stage developmental structures (somites) to their adult location along the edge of muscle fibers in the trunk and limbs [10], [11]. In the adult, motility continues to play a critical role, as MuSCs migrate along muscle fibers *in vivo* to sites of injury to initiate tissue repair [12], [13]. Motility behaviors are heterogeneous between MuSCs and change during stem cell activation [14]. Heterogenous fitness for regeneration within the MuSC pool is well appreciated [15], and analysis of heterogeneous motility behaviors may provide an additional lens through which to decompose different MuSC phenotypes.

Given the biological importance of motility phenotypes, classification of cells based on motility behaviors has useful applications in research and diagnostics. Similarly, exploration of heterogeneity within the motility behaviors of a cell population may provide biological insights. However, it is often difficult to determine *a priori* which features of motility behavior will be predictive of a phenotype of interest, or allow for discrimination of heterogeneous behavior. Additionally, different phenotype classification tasks and cell populations may require distinct feature sets to extract valuable biological information. A method to algorithmically determine relevant features of cell motility for a given classification or discrimination task is therefore advantageous.

A. Related Work

To date, a number of tools have been proposed that rely upon a set of handcrafted features to quantify cell motility behaviors, providing some remarkable results [16], [17], [18], [19], [20]. Neural progenitor cells were discriminated by motility behavior alone [20], and genes that effect motility have been identified solely from timelapse imaging data [16]. These dramatic results demonstrate the potential insights that may be gathered from more extensive analysis of cell motility. However, these methods rely upon *a priori* generation of a feature set, and have thus far focused largely on features of speed and directional persistence. It is possible that more complex features may allow for improved discrimination of cell motility phenotypes, but it is difficult to predict what these features may be in each context.

J. C. Kimmel and W. F. Marshall are with the Department of Biochemistry and Biophysics, University of California San Francisco, California, CA, 94158, USA e-mail: Wallace.Marshall@ucsf.edu

A. S. Brack is with the Department of Orthapedic Surgery, University of California San Francisco, California, CA, 94158 USA e-mail: Andrew.Brack@ucsf.edu

©This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Convolutional neural networks provide an approach to learn relevant features from data, rather than handcrafting features *a priori*. In the field of computer vision, convolutional neural networks (CNNs) have recently made rapid advancements, demonstrating state-of-the-art performance on a variety of image classification tasks [21], [22], [23], [24]. CNNs utilize a set of parametrized kernels to extract image features, allowing distinct feature kernels to be learned for a given classification task [25]. In this way, CNNs are able to learn a "representation" of the problem's feature space. Feature space representations may also be learned in an unsupervised manner by training CNN autoencoder architecture to encode and decode [26], [27]. This approach may be useful for learning relevant motility features where an explicit classification task is not present.

While CNNs are most commonly applied to tasks involving analysis of shape in two-dimensional images at a single time-point, convolution is a natural analytical tool for any input information with spatial dimensions. CNNs have been successfully applied to a diverse set of non-imaging domains, including natural language processing [28], bird song segmentation [29], and EEG recordings [30]. Perhaps most clearly mirroring our challenge of motion classification, CNNs have performed well in the classification of hand-gestures from video recordings [31], [32]. These successful implementations have simply extended CNNs to consider three-dimensional images as inputs, where one axis is time. This approach has also allowed for successful classification of videos using CNNs [33], [34]. We wanted to determine if CNNs could similarly be applied to the problem of cell motility phenotype classification. Cell motility is inherently 3D spatial data, where one dimension is time. If the spatial nature of cell motility data is represented explicitly as a 3D image, in the same manner used for gesture classification, CNNs may allow for motility phenotype classification without *a priori* definition of handcrafted features.

Here, we present *Lanternfish*, a tool to represent motility paths as 3D images, classify different motility behaviors, and learn motility features in an unsupervised fashion using CNNs. *Lanternfish* represents cell motility using a set of positional markers in a 3D volume, with the depth axis representing time. We demonstrate that standard CNN architectures are sufficient to accurately distinguish experimentally observed cell motility phenotypes represented in this way. Additionally, we show that autoencoder architectures can be trained successfully on these 3D motility representations for use as unsupervised feature extractors.

II. Spatial Representations of Motility Paths

Motion data in a two-dimensional plane is inherently three dimensional, with two dimensions in physical space (x and y) and a single time dimension t. Each of these dimensions has relevant spatial meaning, and spatial relationships are required to fully represent the motion of an object. This spatial nature makes motion an ideal candidate for the application of convolutional neural networks, which specialize in learning representations of spatial data.

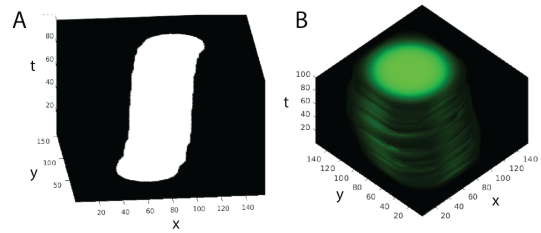

Fig. 1. Representative motility traces using different markers of location. (A) Disk structuring element markers, (B) Gaussian distribution markers.

In order to apply CNNs to the analysis of motion, motion must be represented as a static 3D image. Cell motility is typically recorded as the position of the cell centroid at each time point. To represent this time series of positions as a 3D image, we first produce a simple 3D representation of an (x, y) path by placing a 1 pixel (px) binary marker on the location of the object at each time point t in a single slice of a cube with dimensions (X, Y, T), leaving all other values at 0, where $x \in X$, $y \in Y$, and $t \in T$. Viewed one plane at a time along the t dimension, this cube is simply a video of the 2D path representing the object's location with a 1 px marker. However, this trivial representation presents a very sparse feature space, and intuitively may not allow for efficient learning of convolutional kernels.

In expectation of this sparsity problem, we produced tools to build representations that mark an object's location in each (X, Y, T) plane with a binary disk of arbitrary size or Gaussian distribution of arbitrary variance, instead of a single 1 px point. Gaussian distributions are scaled [0, 1] for each σ value. The resulting representation resembles a "stack of dinner plates" (Fig. 1). These representations contain information about the objects location at more (x, y) coordinates within a plane than the 1 px representations, so we reasoned that they may aide learning of 3D convolutional kernels.

Further information can be encoded by setting the amplitude of the disk or distribution in each t plane based on some real valued measurement. For instance, instantaneous speed or object size could be encoded as amplitudes. Compression of motion paths may be necessary due to GPU memory constraints. For all experiments performed here, motion paths were compressed 4- or 6-fold in the (x, y) dimensions by simple integer division of (X, Y) coordinates.

III. CNN Architecture

A. Classification architecture

For classification of different types of motion, we apply a standard CNN architecture utilizing 3D convolutional and max pooling layers, diagrammed in (Fig. 2). 3D Convolutional layers convolve the 3D motion cube inputs with a set of parametrized kernels, passing the convolutional outputs to the layers above. The max pooling layers perform a $\max$ operation for voxels in an 3D-window, reducing the input size, and returns the resulting output to the layer above. This architecture is similar to well known 2D classification architectures [21], [35]. All convolutional layers are paired

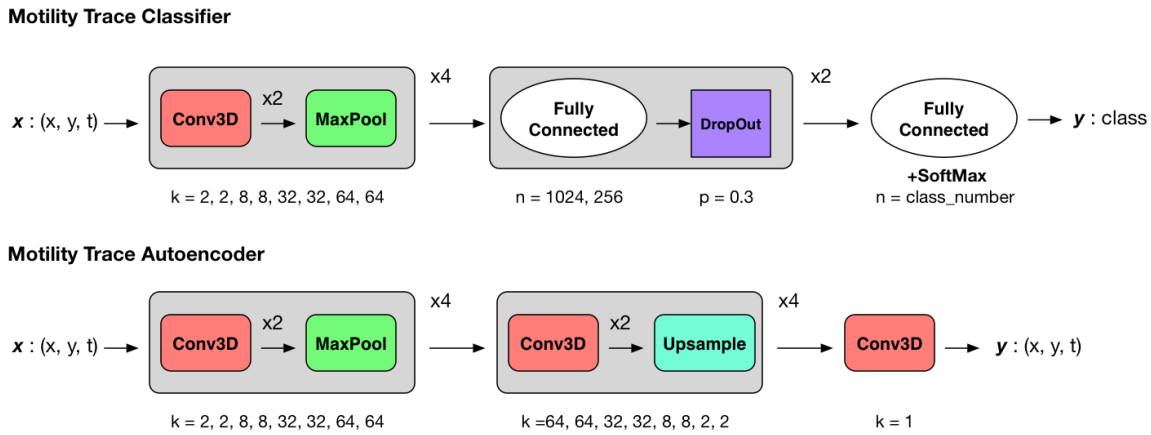

Fig. 2. Motility analysis CNN classification and autoencoder architectures. All convolutional layers are three dimensional and paired with a rectified linear unit activation. Pooling and upsampling layers operate with kernels of size (2, 2, 2) and stride of 2. Zero padding is performed as needed in the autoencoder model to match input size. Here, k is the number of kernels learned by each convolutional layer, n is the number of units in a fully connected layer, and p is the proportion of neurons dropped in a Dropout layer.

with a rectified linear unit (ReLU) activation ($\max(0, x)$) [36], utilize unit strides $s = 1$, and convolve with $(3, 3, 3)$ kernels. Convolutional layers pad input images by 1 px by reflecting edge values to avoid reduction of input size by convolution. Max pooling layers operate with $(2, 2, t)$ kernels and stride $s = 2$, where $t \in \{0, 2\}$ at different points in the network. This allows for less pooling in the time dimension, which may be smaller than the x and y dimensions, as in our case.

Fully connected layers are the same as in a traditional neural network, in which each perceptron unit considers input from all units in the previous layer, and outputs to all units in the next layer [37]. Dropout layers eliminate a random proportion p of fully connected units from a fully connected layer during each forward pass, reducing reliance upon individual units and preventing overfitting [38]. Two fully connected layers with dropout ($p = 0.3$, where p is the proportion of neurons dropped per epoch) and ReLU activations are utilized at the bottom of the network. Final class outputs are returned by a fully connected layer with a number of neurons equal to the number of classes and a *softmax* activation. Notably, we find that stacking multiple convolutional layers is necessary for effective training, possibly due to the increased receptive field size of deeper layers in stack.

Classification networks were trained using stochastic gradient descent with momentum ($\mu = 0.5$). Categorical crossentropy was used as a loss function. We find that training is sensitive to the learning rate ϵ, and thus utilize a low initial learning rate $\epsilon_0 = 0.005$ with a rapid decay function

$$\epsilon_{i+1} = \epsilon_i d^{i+1}$$

where $i \in [0, N]$ is the training epoch, ϵ_0 is the initial learning rate, and d is a decay coefficient, set to $d = 0.8$ for our experiments. We find that the *Adadelta* algorithm [39] performs poorly for training of our classification networks (*data not shown*).

B. Autoencoder architecture

We utilize a similar autoencoder architecture, employing stacked 3D convolutional and max pooling layers at the bottom of the network to encode the input, followed by stacked 3D convolutions and upsampling layers to decode the input (Fig. 2). As in the classification architecture, all convolutional layers are paired with a ReLU activation. Autoencoder networks were trained with the *Adadelta* optimization algorithm [39], utilizing crossentropy or mean-squared error as the loss function for binary and Gaussian representations respectively. This architecture resembles others in the literature [25], [26].

IV. Experimental Results

A. CNNs accurately classify simulated motility behaviors

To determine if CNNs could discriminate between different types of motion under ideal conditions, we first trained classification networks on simulated data from 3 distinct biologically relevant models of motion, namely random walking, Levy flights, and fractional Brownian motion. Random walking is motion with normally distributed random step sizes and directionality. Random walking is observed in freely diffusing biomolecules [40]. Levy flights similarly display random directionality, but step sizes are instead chosen from a long-tailed Levy distribution. Levy flights are observed in multiple biological systems and optimize path finding [41], [42], [43], [44]. Fractal Brownian motion models a random walk with long term dependence, similarly relevant as a representation of regulated motion in biology [45], [46]. By starting with simulated data we can optimize parameters using large sample sizes that would be difficult to obtain with living cells.

Random walks, Levy flights, and fractal Brownian motion were simulated for classification, each with a mean displacement of 5 (x, y) units per time step. Simulations were carried out for 101 time steps and restricted to a $(2048, 2048)$ pixel plane, representing the field-of-view that might be expected using a standard 4 megapixel microscopy camera. Compressed tracks traveling more than 156 px from the origin in any

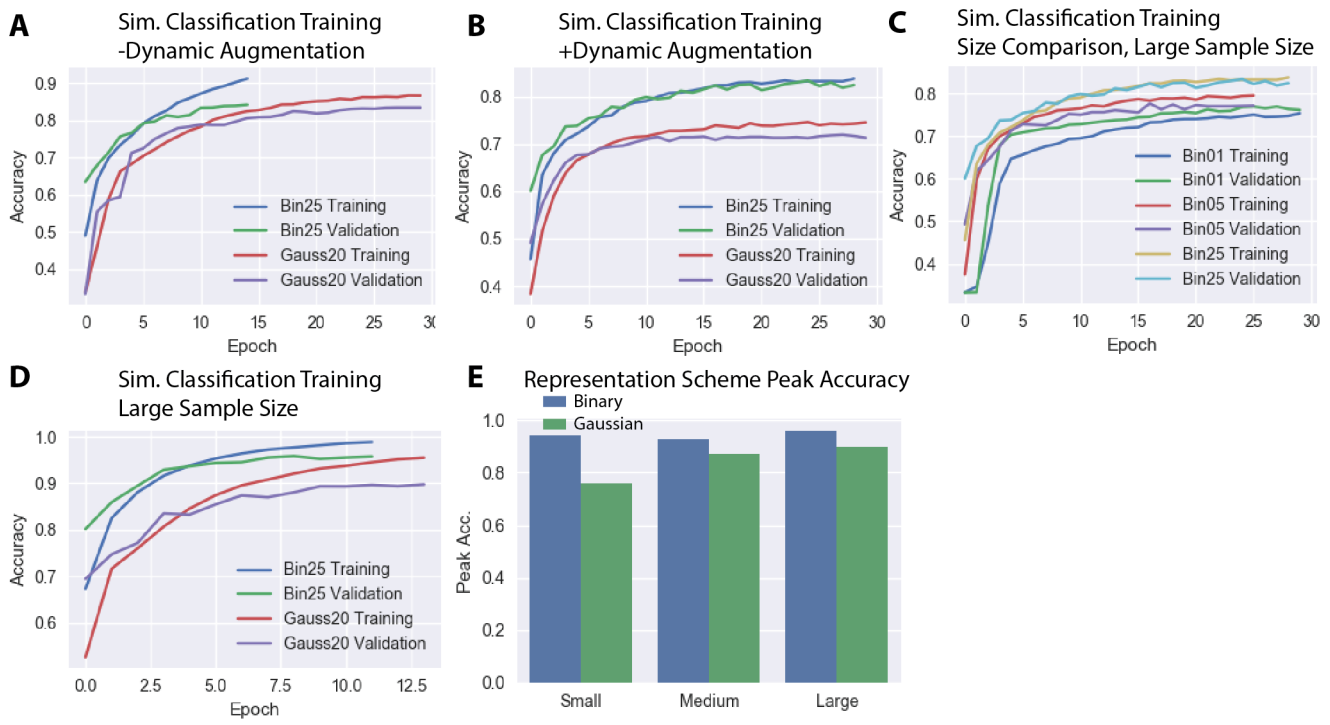

Fig. 3. CNNs can discriminate between different types of motion represented as 3D images. Training plots of three class discrimination between simulated random walks, Levy flights, and fractal Brownian motion using (A) large binary disk markers and (B) large Gaussian markers. (C) Summary of classification accuracies for different representations.

direction were removed from analysis, to prevent dilution of the representation space by a few outlier tracks. Remaining tracks were represented in $(156, 156, 101)$ vx cubes. 4-fold compression and cropping were performed to meet GPU memory constraints. nVidia GTX 1080 (Pascal) GPUs were used for all experiments.

Experiments with 3,000 samples per class were performed using binary disks of three diameters $d \in \{1, 5, 25\}$ and broad Gaussian distributions of different variance $\sigma \in \{3, 10, 20\}$ as place markers. Training was ceased in all models after the validation loss failed to improve for 3 consecutive epochs. The largest binary disk representations achieved ˜84% validation accuracy after 30 epochs, and the largest Gaussian representations of the same data yielded ˜81% validation after 30 epochs of training (Fig. 3A). Both binary and Gaussian representations appear to overfit in later epochs, as evidenced by the divergence of the training performance from validation performance (Fig. 3A). We find that dynamic data augmentation in the form of horizontal and vertical flipping (i.e. rotating about the t axis) reduces this overfitting in both representation types (Fig. 3B). In both representation schemes, training was accelerated by increases in the marker size (Fig. 3C, binary representations shown). While the sparse 1 px binary representation eventually achieved comparable accuracy to the largest binary representation ($d = 25$), training was much slower in initial epochs (Fig. 3C). This suggests that feature learning is accelerated by increases in the size of representation markers.

Subsequent experiments were performed with 13,500 samples per class, again using both binary disk and Gaussian distribution representations. The largest binary disk representations reach ˜96% validation accuracy, while Gaussian distributions reached ˜92% accuracy (Fig. 3D). At all scales, binary representations achieved a higher peak validation accuracy than Gaussian distribution representations (Fig. 3E). These results suggest that 3D CNNs are sufficient to distinguish different classes of motion represented as 3D images, and that multiple representation schemes can be effective. Large binary representation schemes appear to be the most effective representation scheme we tested. Therefore, we utilize large binary representations for all further experiments with live cell data.

B. CNNs accurately discriminate cell types by motility behavior

After validating that CNNs were sufficient to distinguish simulated classes of motion, we applied the same classification networks to distinguish different types of cell motility. Cell motility was tracked in three different cell types by timelapse imaging for 10 hours, followed by segmentation and tracking by standard methods. Mouse embryonic fibroblasts (MEFs) are commonly used for *in vitro* cell culture assays, and neoplastic transformation of these cells has been demonstrated to alter their motility behaviors. We tracked both wild-type and neoplastic (*c-Myc* overexpression, *HRas-V12*) MEFs to compare their motility behaviors. Muscle stem cells (MuSCs) are the obligate stem cell of the skeletal muscle, and their motility is known to be effected by their activation state. Activated MuSCs commit to become myoblasts, a transit

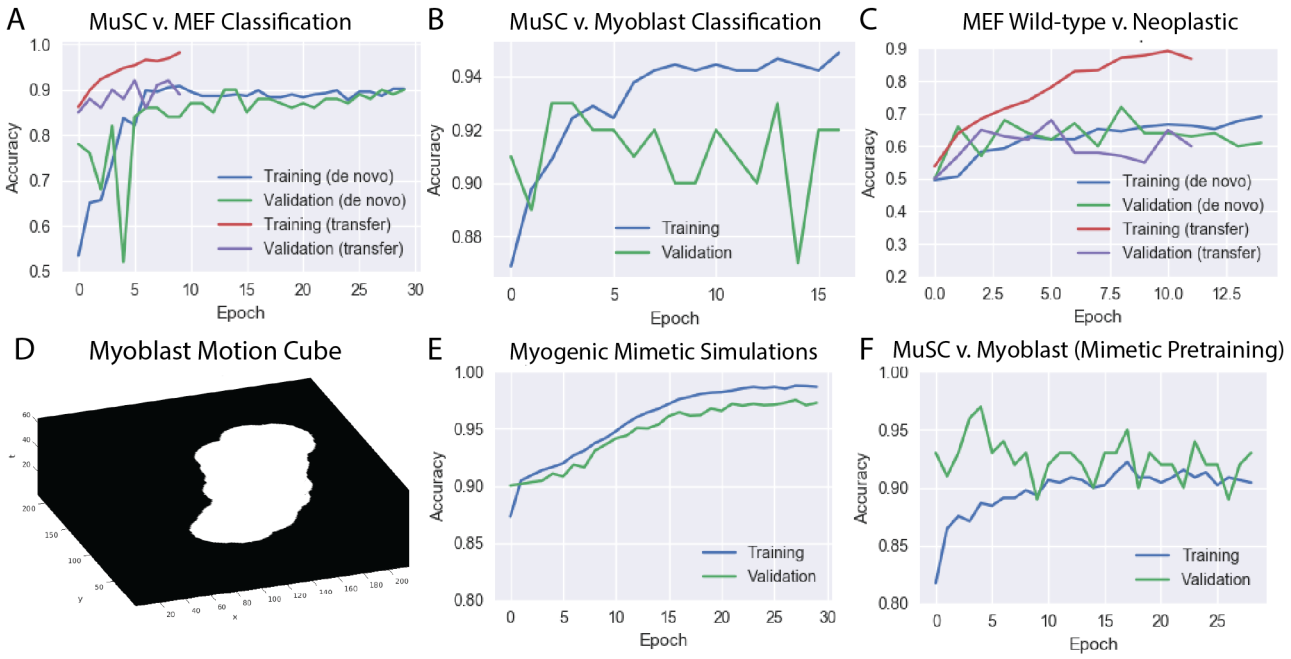

Fig. 4. CNNs can discriminate between different cell types and cell states based on motility. (A) Training accuracy of classification between MEFs and MuSCs with and without transfer learning. (B) Training performance of classification between MuSCs and myoblasts during training with transfer learning from classification of simulated motion models. (C) Training performance of classification between wild-type and neoplastic MEFs. (D) Representative myoblast motion cube representation. (E) Training performance of classification between mimetic simulations of MuSCs and myoblasts. (F) Training performance of classification between real cell data of MuSCs and myoblasts using cell mimetic simulations as pre-training.

amplifying myogenic progenitor cell. We tracked both MuSCs and myoblasts to compare motility between these states of myogenic commitment (see Methods for culture details).

As a proof-of-principle, we trained a classification CNN to discriminate between MEF and MuSC motility, represented using large binary disks ($diameter = 25$ px) in 3D space as described above. The classification network was trained for 30 epochs on MuSC motility traces and MEF motility traces ($n = 165$ per class). $n = 50$ motility traces from each class were used for validation.

The network required roughly 10 epochs with this small data set to begin decreasing the training loss, but proceeded to train effectively thereafter. Validation accuracy reached a peak at 90% (Fig. 4A). These results indicate that even with a very small data set such as this, CNNs can be effectively trained to discriminate different types of cell motility *de novo* (Fig. 4A).

We next wanted to investigate if transfer learning with weights transferred from classifiers trained on simulated data could improve performance, as has been demonstrated broadly with CNNs in 2D image analysis tasks [47]. The same experiment was repeated transferring weights from the trained simulated motion classifier network referenced above to initialize the model. This network trained much more rapidly, as might be expected. The network reached 85% validation accuracy in a single epoch on this small data set, and peaked at 92% validation accuracy (Fig. 4A). These results indicate that training on simulated motion data is an effective pre-training initialization for classification of real cell motility phenotypes.

C. CNNs provide discriminative power between cell activation states

To determine if CNNs can distinguish between more nuanced differences in cell state, classifiers were trained to discriminate between MuSCs and myoblasts ($n = 225$ per class) to determine if CNNs could be used to identify different states of myogenic commitment. A representative example of a myoblast motion cube representation is shown in (Fig. 4D). Dynamic data augmentation was utilized due to the small available sample size. Motion cubes were horizontally and vertically flipped to increase training set diversity without perturbing the representation of motility.

Transfer learning was utilized as above, taking advantage of weights learned from simulated data classification. Classification reached a peak of 93% validation accuracy discriminating MuSCs and myoblasts (Fig. 4C), indicating that CNN classification of motility alone is sufficient to discriminate states of myogenic commitment.

Similarly, classifiers were trained to discriminate between wild-type and neoplastic MEFs ($n = 146$ per class) with transfer learning from the simulated motion classifier. Classification failed to achieve validation accuracy >70% (Fig. 4C). The more nuanced phenotypic difference between wild-type and neoplastic MEFs may be an inherently more challenging classification problem. The small available sample size likely compounds this difficulty and exacerbates the classifier's poor performance.

D. Cell mimetic simulations allow for effective pre-training from small sample sizes

Given the success of pre-training by classification of simulated models of motion, we next attempted to generate simulated data that more accurately reflected real cell motility to enhance pre-training efficacy. For a set of real cell motility data, we measure the displacements and turning behavior of each cell. Displacements are measured simply as the Euclidean distance between each set of sequential timepoints. The turning direction at a point t_i is determined as the angle between the vectors that connect points t_{i-1} to t_i and t_i to t_{i+1}.

Cells are decomposed into a set of k clusters by k-means clustering on a set of parameters measured from these displacement and turn angle distributions. The number of clusters $k = 5$ was chosen empirically to capture the diversity of the cell phenotypes while still leaving non-trivial numbers of cells in each cluster. For each cluster, a bounded Johnson distribution is fit to the aggregate distribution of displacements and the aggregate distribution of turn angles. Simulated samples are generated by randomly sampling displacement magnitudes and turn angles from the fitted Johnson distributions for T time steps. To represent a population of cells, the proportion of simulations generated from each cluster is proportional to the cluster's prevalence in the original cell data. This approach may be conceptually likened to the bag-of-words model [48], in which k-means clustering is used to decompose features into a representative "vocabulary." By sampling from each of k clusters proportionally, we aim to capture and simulate heterogeneous phenotypes within a cell population, rather than simply reproducing a single averaged phenotype that may not be representative of any true cell phenotype.

We generated "cell mimetic" simulations for MuSCs and myoblasts by the above method, with $n = 15,000$ simulated samples for each of the two activation states. A classification network was pretrained by classifying between the two simulated data sets, reaching 97% validation accuracy (Fig. 4E). The weights from this pretrained network were used to initialize a classifier trained on real MuSC and myoblast motility traces ($n = 225$ per class), as performed above. This classifier reached 97% peak validation accuracy, as compared to 94% validation accuracy utilizing a simulated motion model classification for pretraining (Fig. 4F). These results indicate that classification of simulated data that mimics a real cell data set is a more effective pretraining regimen than classification of generic simulated motion models.

E. Autoencoders allow unsupervised learning of representations in motion feature space

Results up to this point indicate that supervised classification of different cell motility phenotypes using convolutional neural networks is effective, and that standard dynamic data augmentation and transfer learning techniques perform well in this paradigm. However, in the analysis of motility data, supervised classification data is not always available. For instance, to explore the heterogeneity of phenotypes in a given population, there is no obvious method to generate supervised classification data that may be used to learn relevant feature kernels by optimization of a standard classification loss function. This would also be an issue in the identification of heterogeneous motility behaviors in patient biopsy samples, in which the distinguishing features are not known *a priori*. Training CNNs as autoencoders in an unsupervised fashion has been used in other contexts to learn relevant feature kernels where no obvious classification problem is present [26], [27]. We next attempted to train autoencoders on our 3D representations of cell motility to learn relevant feature kernels in the absence of a supervised classification problem.

A standard autoencoder architecture utilizing stacked convolutions, followed by upsampling and stacked convolutional layers was trained on 13,500 samples of each class for three types of simulated motion (random walk, Levy flight, fractal Brownian motion). This architecture represents a mirroring of the bottom convolutional layers of our classification architecture about an encoded representation (Fig. 2). Binary crossentropy was used as a loss function when training on binary disk representations, while mean squared error was used to as a loss function for Gaussian distribution representations. All autoencoders successfully reduced loss over several training epochs (Fig 5A). When visually inspected, autoencoder outputs appear to accurately reflect input motility representations (Fig 5B, C).

To determine if autoencoders trained on 3D motility representations could be employed as feature extractors, we utilized the output of the autoencoder's central layer (the encoded representation) as features. In our initial autoencoder architecture (Fig. 2), this encoded representation was still fairly high-dimensional ($d = 1024$), hindering distinction of simulated data from these features (Supp. Fig. 3). We next trained a second "bottle-necked" autoencoder architecture with two fully-connected layers ($n = 256$) paired with dropout ($p = 0.3$) at the center.

Features from the center of this "bottle-necked" autoencoder appear useful for class separation when the feature space is visualized (Fig. 6D). A Random Forest classifier ($n = 100$) trained to distinguish simulated motility phenotypes using exclusively the "bottle-necked" features achieved ˜80% accuracy (5 fold cross-validated). This result indicates that the "bottle-necked" features contain some discriminative power, albeit less than a CNN trained to classify in a supervised manner.

Autoencoders are often used for unsupervised pre-training prior to a classification problem. When classifiers were trained to discriminate between the three simulated classes of motion, transfer learning from the top layers of an autoencoder moderately slows training, rather than expediting it (Fig 6E). This may indicate that the feature kernels necessary to accurately separate motion classes in this context are distinct from those that allow for the most efficient autoencoding.

V. Conclusion

Convolutional neural networks allow for representative learning, or learning of features relevant for the description of a feature space. These experiments show that CNNs may be applied to the analysis of motion data by representing motion as a three-dimensional space. We demonstrate that

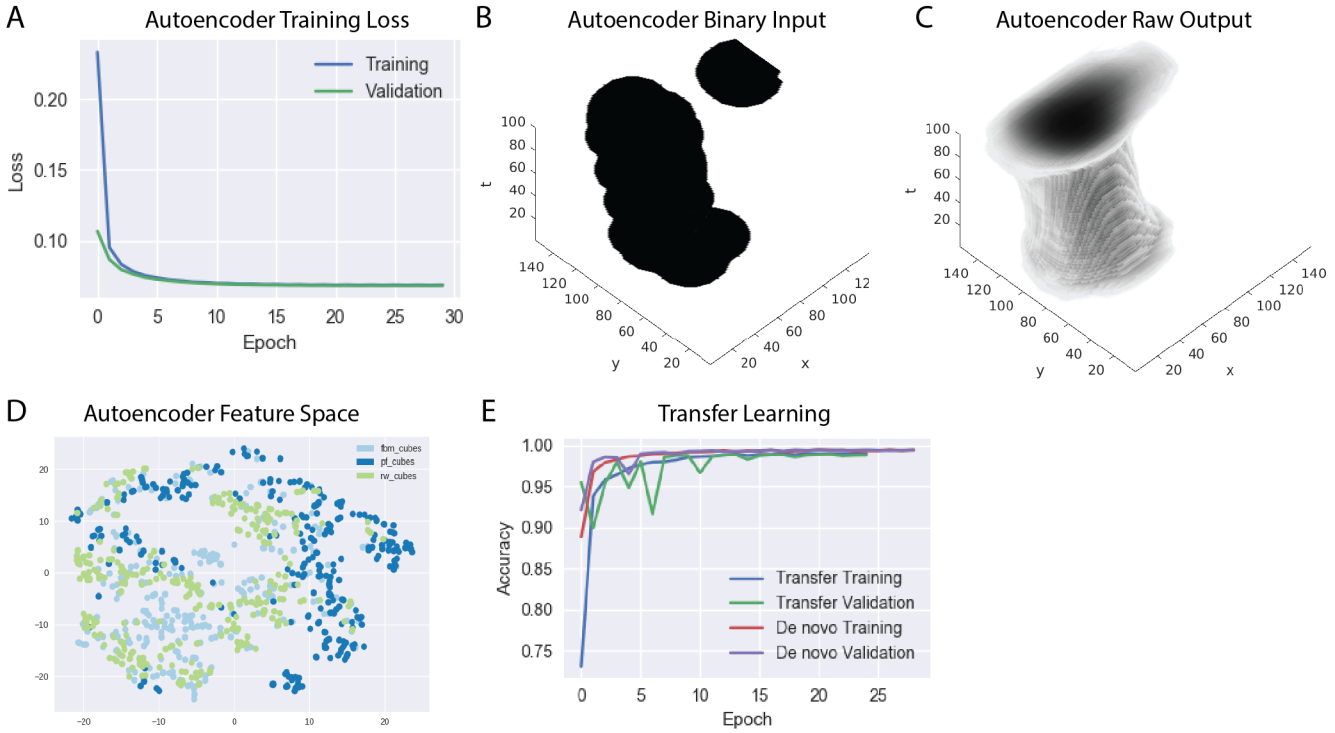

Fig. 5. CNN autoencoders can learn representations of motion feature space in an unsupervised manner. (A) Training plots for a motility trace autoencoder, trained on three simulated classes of motion, (B) sample input and (C) output images. (D) Visualization of autoencoder feature space with tSNE. (E) Training accuracy of classifiers with transfer learning and de novo.

CNNs are capable of discriminating between simulated models of motion and multiple types of cell motility. Additionally, we find that CNN autoencoders can be trained effectively on these 3D motion representations in an unsupervised fashion. In our cell data sets, we find that CNNs effectively discriminate between different cell types, and different states of myogenic progenitor activation. While we apply the methods described here to cell biology, there is no conceptual limitation that prevents application to other fields where discrimination based on motion recordings is desired. In the field of cell biology, analysis of motility with CNNs may allow for useful insights to be gathered in contexts where relevant features are non-obvious or laborious to construct.

Acknowledgment

This work was funded by NSF grant MCB-1515456 to W.F.M., NIH grants AR060868 and AR061002 to A.S.B., and a PhRMA Foundation fellowship to J.C.K. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1650113 to J.C.K. J.C.K and W.F.M. are members of the NSF Center for Cellular Construction, NSF Grant No. 1548297.

References

[1] T. J. Mitchison and L. P. Cramer, "Actin-based cell motility and cell locomotion." *Cell*, vol. 84, no. 3, pp. 371–379, Feb. 1996.

[2] D. A. Lauffenburger and A. F. Horwitz, "Cell migration: A physically integrated molecular process," *Cell*, vol. 84, no. 3, pp. 359–369, 1996.

[3] H. T. Enterline and D. R. Coman, "The ameboid motility of human and animal neoplastic cells." *Cancer*, vol. 3, no. 6, pp. 1033–1038, Nov. 1950.

[4] D. R. Coman, "Mechanism of the Invasiveness of Cancer," *Science*, vol. 105, no. 2727, pp. 347–348, Apr. 1947.

[5] E. Sahai, M. F. Olson, and C. J. Marshall, "Cross-talk between Ras and Rho signalling pathways in transformation favours proliferation and increased motility." *The EMBO Journal*, vol. 20, no. 4, pp. 755–766, Feb. 2001.

[6] C. B. Pollock, S. Shirasawa, T. Sasazuki, W. Kolch, and A. S. Dhillon, "Oncogenic K-RAS is required to maintain changes in cytoskeletal organization, adhesion, and motility in colon cancer cells," *Cancer Res*, vol. 65, no. 4, pp. 1244–1250, 2005.

[7] M. A. Nieto, "Epithelial plasticity: a common theme in embryonic and cancer cells." *Science (New York, N.Y.)*, vol. 342, no. 6159, pp. 1 234 850–1 234 850, Nov. 2013.

[8] C. L. Smith, O. Kilic, P. Schiapparelli, H. Guerrero-Cazares, D.-H. Kim, N. I. Sedora-Roman, S. Gupta, T. O'Donnell, K. L. Chaichana, F. J. Rodriguez, S. Abbadi, J. Park, A. Quiñones-Hinojosa, and A. Levchenko, "Migration Phenotype of Brain-Cancer Cells Predicts Patient Outcomes," *Cell Rep*, vol. 15, no. 12, pp. 2616–2624, Jun. 2016.

[9] D. J. Laird, U. H. von Andrian, and A. J. Wagers, "Stem Cell Trafficking in Tissue Development, Growth, and Disease," *Cell*, vol. 132, no. 4, pp. 612–630, Feb. 2008.

[10] M. Buckingham, L. Bajard, T. Chang, P. Daubas, J. Hadchouel, S. Meilhac, D. Montarras, D. Rocancourt, and F. Relaix, "The formation of skeletal muscle: from somite to limb." *Journal of Anatomy*, vol. 202, no. 1, pp. 59–68, Jan. 2003.

[11] S. M. Hughes and H. M. Blau, "Migration of myoblasts across basal lamina during skeletal muscle development." *Nature*, vol. 345, no. 6273, pp. 350–353, May 1990.

[12] C. F. Bentzinger, J. von Maltzahn, N. A. Dumont, D. A. Stark, Y. X. Wang, K. Nhan, J. Frenette, D. D. W. Cornelison, and M. A. Rudnicki, "Wnt7a stimulates myogenic stem cell motility and engraftment resulting in improved muscle strength." *J Cell Biol*, vol. 205, no. 1, pp. 97–111, Apr. 2014.

[13] L. A. S. Alfaro, S. A. Dick, A. L. Siegel, A. S. Anonuevo, K. M.

McNagny, L. A. Megeney, D. D. W. Cornelison, and F. M. V. Rossi, "CD34 Promotes Satellite Cell Motility and Entry into Proliferation to Facilitate Efficient Skeletal Muscle Regeneration," *Stem Cells*, vol. 29, no. 12, pp. 2030–2041, Nov. 2011.

[14] A. L. Siegel, K. Atchison, K. E. Fisher, G. E. Davis, and D. D. W. Cornelison, "3D Timelapse Analysis of Muscle Satellite Cell Motility," *Stem Cells*, vol. 27, no. 10, pp. 2527–2538, Oct. 2009.

[15] M. T. Tierney and A. Sacco, "Satellite Cell Heterogeneity in Skeletal Muscle Homeostasis," *Trends Cell Biol*, pp. 1–11, Mar. 2016.

[16] A. S. Sebag, S. Plancade, C. Raulet-Tomkiewicz, R. Barouki, J.-P. Vert, and T. Walter, "A generic methodological framework for studying single cell motility in high-throughput time-lapse data," *Bioinformatics*, vol. 31, no. 12, pp. 320–328, 2015.

[17] A. Sacan, H. Ferhatosmanoglu, and H. Coskun, "CellTrack: an open-source software for cell tracking and motility analysis." *Bioinformatics*, vol. 24, no. 14, pp. 1647–1649, Jul. 2008.

[18] L. Martens, G. Monsieur, C. Ampe, K. Gevaert, and J. Vandekerckhove, "BMC Bioinformatics," *BMC bioinformatics*, vol. 7, no. 1, pp. 289–6, 2006.

[19] M.-A. Bray and A. E. Carpenter, "CellProfiler Tracer: exploring and validating high-throughput, time-lapse microscopy image data." *BMC bioinformatics*, vol. 16, no. 1, p. 368, 2015.

[20] A. R. Cohen, F. L. A. F. Gomes, B. Roysam, and M. Cayouette, "Computational prediction of neural progenitor cell fates." *Nature methods*, vol. 7, no. 3, pp. 213–218, Mar. 2010.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.

[22] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular Systems Biology*, vol. 12, no. 7, p. 878, 2016.

[23] K. He, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Microsoft Research, Redmond, United States, Jan. 2016, pp. 770–778.

[24] D. A. Van Valen, D. A. Van Valen, T. Kudo, T. Kudo, K. M. Lane, K. M. Lane, D. N. Macklin, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, and M. W. Covert, "Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments," *PLoS computational biology*, vol. 12, no. 11, p. e1005177, Nov. 2016.

[25] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *An MIT Press book in preparation* . . . , 2015.

[26] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, Dec. 2013.

[27] Y. Bengio, P. Lamblin, and D. Popovici, "Greedy layer-wise training of deep networks," *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2007.

[28] A. Conneau and H. Schwenk, "Very deep convolutional networks for text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2016, pp. 1107–1116.

[29] L. Neal, F. Briggs, and R. Raich, "Time-frequency segmentation of bird song in noisy acoustic environments," *Acoustics*, pp. 2012–2015, 2011.

[30] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks," in *ICLR*, Nov. 2015.

[31] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand Gesture Recognition With 3D Convolutional Neural Networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–7.

[32] D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, and J.-M. Odobez, "Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition." *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.

[33] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, 2015.

[34] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732, 2014.

[35] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, Sep. 2014, pp. 730–734.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 807–814.

[37] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[38] N. Srivastava, G. E. Hinton, and A. Krizhevsky, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[39] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv*, Dec. 2012.

[40] H. C. Berg, *Random Walks in Biology*. Princeton University Press, 1993.

[41] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Levy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.

[42] G. M. Viswanathan, E. P. Raposo, and M. G. E. da Luz, "Lévy flights and superdiffusion in the context of biological encounters and random searches," *Physics of Life Reviews*, vol. 5, no. 3, pp. 133–150, Sep. 2008.

[43] M. A. Lomholt, K. Tal, R. Metzler, and K. Joseph, "Lévy strategies in intermittent search processes are advantageous," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 32, pp. 11 055–11 059, Aug. 2008.

[44] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches." *Nature*, vol. 401, no. 6756, pp. 911–914, Oct. 1999.

[45] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM review*, vol. 10, no. 4, pp. 422–437, 1968.

[46] B. B. Mandelbrot, "A Fast Fractional Gaussian Noise Generator," *Water Resources Research*, vol. 7, no. 3, pp. 543–553, Jun. 1971.

[47] J. Yosinski, J. Clune, and Y. Bengio, "How transferable are features in deep neural networks?" *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014.

[48] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval." *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 591–606, Apr. 2009.