Medical Image Analysis 91 (2024) 102991

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Generative modeling of living cells with SO(3)-equivariant implicit neural representations

David Wiesner [a], Julian Suk [b], Sven Dummer [b], Tereza Nečasová [a], Vladimír Ulman [c], David Svoboda [a], Jelmer M. Wolterink [b,*]

[a] Centre for Biomedical Image Analysis, Masaryk University, Brno, Czech Republic
[b] Department of Applied Mathematics & Technical Medical Centre, University of Twente, Enschede, The Netherlands
[c] IT4Innovations, VSB – Technical University of Ostrava, Ostrava, Czech Republic

ARTICLE INFO

Dataset link: https://cbia.fi.muni.cz/research/simulations/implicit_shapes, https://github.com/MIAGroupUT/IMPLICIT-CELL-SURFACES

MSC:
41A05
41A10
65D05
65D17

Keywords:
Cell shape modeling
Neural network
Implicit neural representation
Generative model

ABSTRACT

Data-driven cell tracking and segmentation methods in biomedical imaging require diverse and information-rich training data. In cases where the number of training samples is limited, synthetic computer-generated data sets can be used to improve these methods. This requires the synthesis of cell shapes as well as corresponding microscopy images using generative models. To synthesize realistic living cell shapes, the shape representation used by the generative model should be able to accurately represent fine details and changes in topology, which are common in cells. These requirements are not met by 3D voxel masks, which are restricted in resolution, and polygon meshes, which do not easily model processes like cell growth and mitosis. In this work, we propose to represent living cell shapes as level sets of signed distance functions (SDFs) which are estimated by neural networks. We optimize a fully-connected neural network to provide an implicit representation of the SDF value at any point in a 3D+time domain, conditioned on a learned latent code that is disentangled from the rotation of the cell shape. We demonstrate the effectiveness of this approach on cells that exhibit rapid deformations (*Platynereis dumerilii*), cells that grow and divide (*C. elegans*), and cells that have growing and branching filopodial protrusions (A549 human lung carcinoma cells). A quantitative evaluation using shape features and Dice similarity coefficients of real and synthetic cell shapes shows that our model can generate topologically plausible complex cell shapes in 3D+time with high similarity to real living cell shapes. Finally, we show how microscopy images of living cells that correspond to our generated cell shapes can be synthesized using an image-to-image model.

1. Introduction

Accurate and reliable segmentation of biomedical optical microscopy images is a challenging task (Meijering, 2012, 2020), which is extremely time-consuming and tedious when performed manually, especially on 3D and 3D+time data (Coutu and Schroeder, 2013; Webb et al., 2003). There is a clear need for automatic segmentation methods, and deep learning methods have shown promising results (Stringer et al., 2021). However, these methods require large sets of information-rich and varied training data (Kozubek, 2016), consisting of pairs of microscopy images and their target segmentation masks.

There have been several efforts to collect large community data sets of images and masks, such as the Broad Bioimage Benchmark Collection[1] or BioImage.IO.[2] In case images are available but manual annotation is infeasible, masks generated by automatic algorithms are sometimes provided as a silver standard (Ulman et al., 2016; Burgos and Svoboda, 2022). For example, this approach was taken for a variety of optical microscopy scenarios in the Cell Tracking Challenge,[3] where silver-standard corpora were used for training new segmentation methods (Arbelle and Raviv, 2019; Löffler and Mikut, 2022).

Alternatively, if neither images nor masks are available, both can be synthesized. Previously, fully synthetic annotated images were obtained with carefully hand-crafted models based on measurements in real images (Lehmussola et al., 2007; Svoboda et al., 2009; Rajaram et al., 2012; Malm et al., 2015; Kovacheva et al., 2016; Murphy, 2016; Stegmaier et al., 2016; Svoboda and Ulman, 2017; Sorokin et al., 2018). Recently, deep learning methods, and in particular those

* Corresponding author.
E-mail addresses: wiesner@fi.muni.cz (D. Wiesner), j.m.wolterink@utwente.nl (J.M. Wolterink).

[1] https://bbbc.broadinstitute.org
[2] https://bioimage.io/#/?type=dataset
[3] http://celltrackingchallenge.net

https://doi.org/10.1016/j.media.2023.102991
Received 1 April 2023; Received in revised form 20 August 2023; Accepted 2 October 2023
Available online 5 October 2023
1361-8415/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

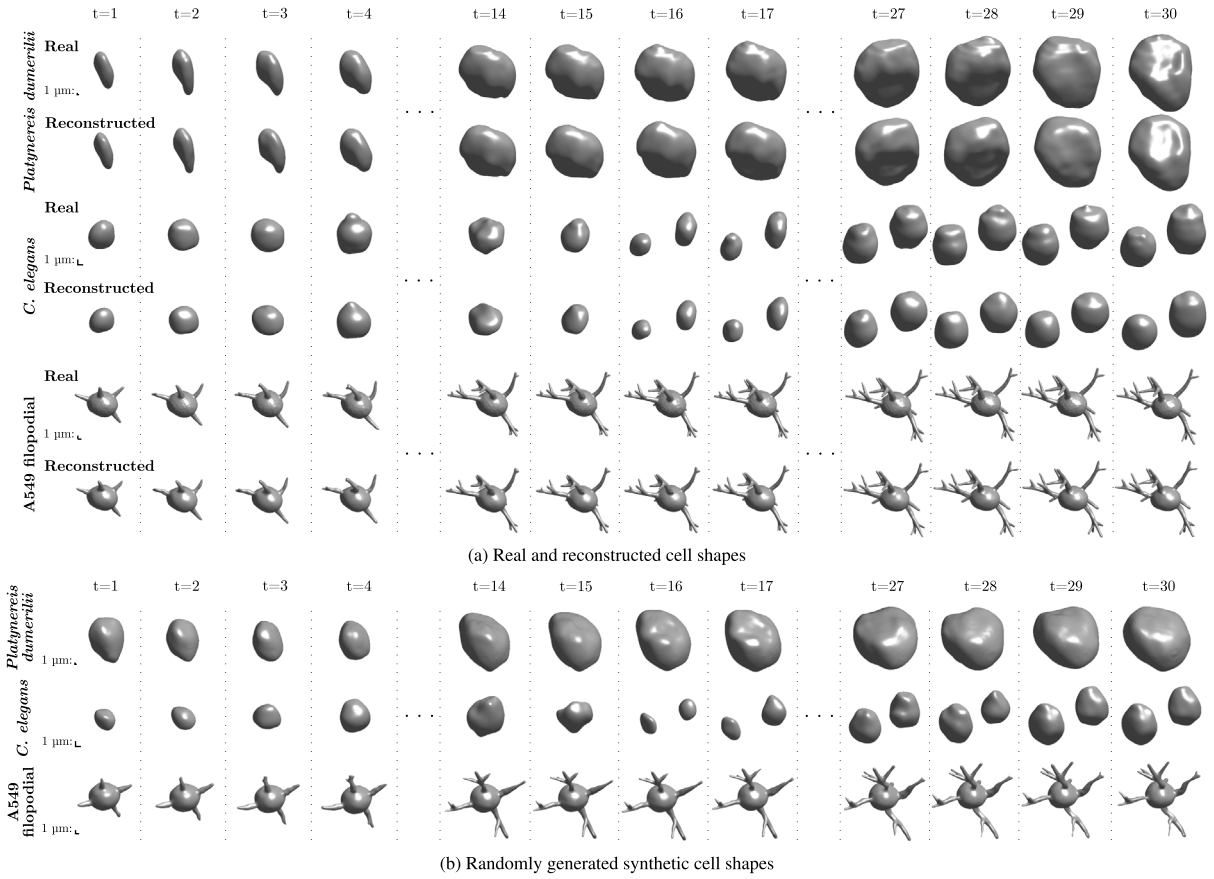


Fig. 1. Visual comparison of real, reconstructed (a), and randomly generated synthetic cell shapes (b) in 3D+time. The proposed method is able to synthesize living cell shapes that accurately mimic processes such as cell growth in *Platynereis dumerilii* cell (top), cell division in *C. elegans* cells (middle), and growth and branching of filopodial protrusions in A549 lung cancer cells (bottom).

using generative adversarial networks (GANs), have shown promising results for augmentation of existing training sets (Osokin et al., 2017; Goldsborough et al., 2017; Böhland et al., 2019; Bailo et al., 2019; Baniukiewicz et al., 2019; Kozlovský et al., 2021) or synthesis of completely new data sets (Fu et al., 2018; Dunn et al., 2019). The apparent strength of these deep learning-based approaches lies in their ability to learn information autonomously, usually the appearance of cells, and to create corresponding faithful new content.

The synthesis of new data pairs typically has two stages. The first stage consists of synthesizing a completely new shape (in 2D or 3D) or shape sequence (when time is included). In the second stage, these shapes are used to synthesize textured cell images that are visually similar to real microscopy images. GANs excel at this second stage owed in part to image-to-image translation methods such as *pix2pix* (Isola et al., 2017), which have found widespread application in microscopy image synthesis. However, the first stage requires fundamentally different models and, most importantly, shape representations that can deal with the variation in cell shapes. As exemplified in Fig. 1, cells exhibit a range of visually observable phenomena including global cell body changes such as cell growth, cell division, deformation in cell tissue, or projections of leading edge during cell motility, as well as localized body changes, such as blebbing (i.e., randomly growing and retracting blobs on the cell surface) or growing filopodia (i.e., highly motile thin protrusions budding off the main cell body).

In this work, we synthesize living cell shapes in 3D+time, which we do from user-given examples such as segmentation masks of real cells. We propose to model the cell surface as the zero level-set of a continuous signed distance function (SDF) in 3D space and time. Following the DeepSDF model proposed by Park et al. (2019), we represent this SDF function in an implicit neural representation (INR). Such an INR consists of a multilayer perceptron (MLP) that we jointly optimize with a latent space using a large set of 3D cell shape time-lapse sequences. Once trained, owing to the continuous implicit representation, the optimized model can be used to synthesize completely new cell shape sequences at any given spatial or temporal resolution. In contrast to voxel-based or mesh-based cell shape representations with discretized sampling rates, our approach alleviates the limitations on spatial and temporal resolution and allows us to model and represent cell shapes at an arbitrary level of detail. By parametrizing the cell surface implicitly through differentiable and, therefore, trainable neural networks, the complexity of the resulting model is independent of the spatial and temporal resolution and instead scales with the complexity of the cell surface. Hence, in contrast to voxel-based or mesh-based representations, the resulting network has fixed memory requirements.

We have previously proposed to represent living cell shapes using INRs (Wiesner et al., 2022). This work extends our previous work in several ways. First, we have adapted our model to be independent of spatial rotation of the cell. That is, we disentangle latent code and cell rotation so that two shapes that are rotated versions of each other will always be represented by the same latent code. This is a logical requirement in cell shape synthesis, where there is no canonical orientation, which allows us to learn more descriptive latent codes. Second, we have extended our data set with additional cell types so that we now optimize our model on three widely different cell shapes: *Platynereis dumerilii* embryo cells, *C. elegans* embryo cells, and A549 human lung carcinoma cells. Third, we have substantially extended our quantitative evaluation, and included an ablation study in which we evaluate the effect of rotation equivariance on the compactness of the learned latent space. We show how with these additions, we can generatively model different types of living cells with SO(3)-equivariant

INRs and use the synthesized shapes as input to a GAN model to synthesize pairs of time-lapse images and segmentation masks.

2. Related work

We summarize existing explicit cell shape representation and synthesis methods and provide a brief introduction to related work in implicit neural representations. Moreover, we provide an introduction to equivariance and invariance in geometric deep learning models.

2.1. Explicit representations

Cell shape masks organized on pixel or voxel grids are the predominant standard for cell shape representation. The shape and resolution of the masks are chosen so that they match those of the original microscopy image, allowing easy correspondence matching and overlaying of the original image. While this has advantages, the memory requirements of such a representation grow quadratically (in 2D) or cubically (in 3D) with resolution. Hence, to represent fine details, such as a cell body with thin protrusions, memory requirements explode. This is also the case when the shape needs to be evolving (Svoboda and Ulman, 2017), in which case the denser grid makes it less likely for the shape boundary pixels to fall off the grid and to lose boundary localization precision. Voxel grids have been used in Cellular Potts Models (Merks and Glazier, 2005) (CPMs), which simulate time-resolved 3D cell shapes within a cell population (Merks and Glazier, 2005; Swat et al., 2012; Starrußet al., 2014; Svoboda and Nečasová, 2020). Deep learning models using GANs for cell shape synthesis also rely on grids (Wiesner et al., 2019a; Fu et al., 2018; Baniukiewicz et al., 2019). These models are able to synthesize static cell shapes in 2D, pseudo-3D, and 3D. In the pseudo-3D approach, individual 2D slices are synthesized and subsequently composed into a 3D volume. One significant drawback of grid representations is that there are no topological guarantees, which might result in disconnected components.

An alternative way to represent cell shapes is to represent them as a combination of (overlapping) spheres (Dufour et al., 2005). Several physics-oriented systems developed to simulate and study cell populations have opted to use this representation of the shape (Van Liedekerke et al., 2015; Ghaffarizadeh et al., 2018). Other lattice-free models use ellipsoids, in particular to represent cell nuclei (Böhland et al., 2019; Dunn et al., 2019; Han et al., 2019). A drawback of such representations is their limited ability to represent cells with protrusions, blebs, and other fine details. Polygonal meshes provide an alternative choice when detailed representation of 2D manifolds in 3D is desired. The approach is very well established for static shapes, but living cell shapes such as we consider in this paper are challenging to model with triangular meshes (Li and Kim, 2016; Sorokin et al., 2018), and often lead to intersecting faces and other mesh artifacts (Li and Kim, 2016). Alternatively, cells can be represented using spherical harmonics (Ducroz et al., 2012).

2.2. Implicit neural representations

Implicit neural representations (INRs), also called *coordinate networks* or *neural fields*, have recently become a popular choice to represent signals in space and – optionally – time (Xie et al., 2022). INRs are based on the idea that a multilayer perceptron with coordinates as input can universally represent functions on a domain. However, in practice, multilayer perceptrons suffer from spectral bias, which means that they have difficulties representing high-frequency signals. Efforts to overcome this bias have focused on positional encoding of input coordinates (Mildenhall et al., 2020; Tancik et al., 2020) or the use of alternative activation functions (Sitzmann et al., 2020). INRs can be used to represent any function in any space, which has led to a range of applications in medical imaging. For example, INRs can represent sinograms for CT reconstruction (Sun et al., 2021), MRI images obtained from k-space measurements (Shen et al., 2022), deformation fields in image registration (Wolterink et al., 2022), or outputs in image-to-image synthesis (Chen et al., 2023). Of particular interest to the current work are INR representations of manifolds in space, typically via the implicit representation of an SDF. Gropp et al. (2020) showed how based on a limited number of training points, an INR can represent a continuous manifold in space. Park et al. (2019) demonstrated how an INR can be coupled with a latent space by conditioning the multilayer perceptron on a latent code, a feature that is critical to the current work. Recently, Erkoç et al. (2023) proposed an approach for unconditional generative modeling using INRs. First, a set of MLPs is optimized to implicitly represent given data samples. Subsequently, a diffusion model is trained on the optimized weights of the MLPs to model the underlying distribution. The resulting diffusion model is then used to synthesize new MLP weights that represent an INR to generate new data samples in 3D and 3D+time.

2.3. Equivariant and invariant shape learning

Geometric deep learning is a learning paradigm in which neural networks are constructed under consideration of symmetry, i.e., by specifying groups of transformations to which the networks are supposed to be equivariant or invariant. A neural network is called equivariant if transforming its input results in the same transformation applied to its output and invariant if transforming the input has no effect on its output. For instance, when we use a neural network to classify images of cats and dogs, the neural network should be invariant to the rotations of the animals in the picture. Furthermore, when performing segmentation using a neural network, the neural network should output a rotated segmentation when the input image is rotated. In other words, the neural network should be equivariant to rotations. Finally, invariance and equivariance play a role in shape representation. For example, shapes do not intrinsically change when an object is rotated. Consequently, if we have two rotated instances of an otherwise identical shape, we want them to share the same representation that is independent of rotations. Such symmetry can be induced in neural networks in different ways, e.g., by imposing and solving linear constraints on the trainable parameters (Finzi et al., 2021). In the context of conditional shape encoding and decoding, Atzmon et al. (2022) constructed equivariant and invariant layers via so-called frame averaging. Deng et al. (2021) enforce rotational symmetry between latent code and shape by casting latent codes to Euclidean vectors for which rotation is well-defined. In this work, instead of coupling the latent code with shape orientation, we aim to explicitly de-couple it, achieving task-specific (approximate) independence of rotation.

3. Method

This work describes a method to model the surface of living cells in space and time. We represent the time-evolving cell surface implicitly as the zero level-set of a time-dependent continuous SDF that is parametrized by a neural network. We condition the neural network on learned latent code that describes the dynamics of the cell in space and time. By sampling new codes from the learned latent distribution and using these as input to the model, we can synthesize new and unseen time-evolving shapes. Furthermore, we disentangle the latent code and the rotation of a shape. This facilitates learning a more compact latent space as all rotations of a particular shape are represented by a single latent code, with rotation explicitly defined by a separate parameter. This results in a rotation equivariant implicit representation of time-evolving cell shapes. Fig. 2 shows a diagram of the network.

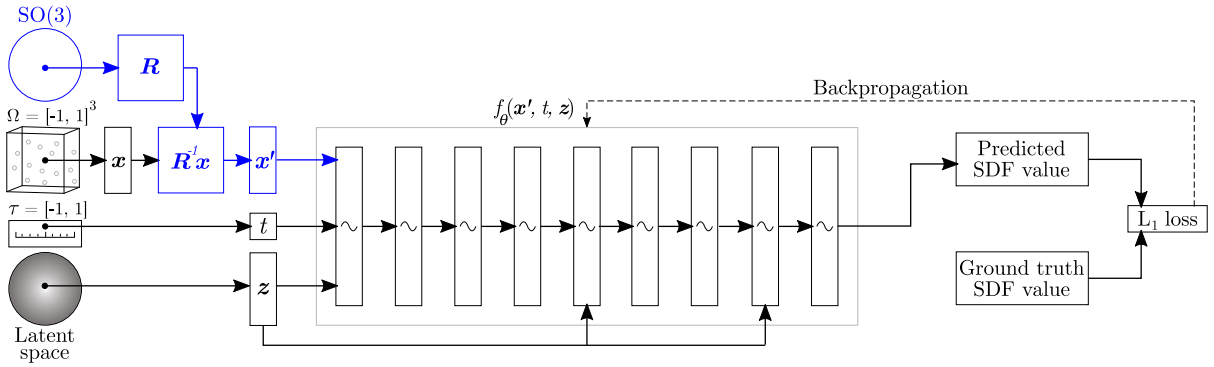


Fig. 2. Conceptual diagram of the proposed network and its SO(3)-equivariant extension (in blue). The neural network f_θ is given a latent code $\boldsymbol{z}$ sampled from a multivariate normal distribution, coordinates $\boldsymbol{x} = (x, y, z)$ from a spatial domain Ω, and a temporal coordinate t from temporal domain τ. Moreover, the latent code $\boldsymbol{z}$ is given to the network not only at the input but also at its fifth and eight layer. The network is optimized to output the SDF values at given points, whereas the latent codes are jointly optimized to match a given normal distribution. The trained network is able to output SDF values based on a given latent code at any coordinate in the space–time domain. When given new latent codes from the latent space, the trained network is able to infer new spatio-temporal SDFs and thus produce new time-evolving shapes. The equivariant extension modifies the sampling procedure of the spatial coordinates and introduces a rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$. Rotating the spatial coordinates $\boldsymbol{x}$ by a rotation matrix $\boldsymbol{R}$ results in a new rotated spatial coordinates $\boldsymbol{x}'$. The rotation matrix $\boldsymbol{R}$ is optimized along with the network weights θ and the latent codes $\boldsymbol{z}$. During inference, the network reconstructs an SDF of a time-evolving cell shape according to a randomly sampled latent code $\boldsymbol{z}$ and given spatial coordinates $\boldsymbol{x}$, temporal coordinate t, and rotation matrix $\boldsymbol{R}$.

3.1. Signed distance function

We propose to represent the evolution of a cell surface as the zero-level set of its time-evolving SDF. The SDF is a continuous function that, for any point in space, gives the signed Euclidean distance to the nearest point on the cell surface. By convention, its sign is positive for points outside the shape and negative for points inside the shape. Here, we also consider that the SDF of a living cell at a particular point in space evolves over time. More precisely, let $\Omega = [-1, 1]^3$ be a spatial domain, $\tau = [-1, 1]$ a temporal domain, and $\mathcal{M}_t$ be a 2D manifold embedded in Ω at time $t \in \tau$. For any point $\boldsymbol{x} = (x, y, z) \in \Omega$, the $\mathrm{SDF}_{\mathcal{M}_t} : \Omega \to \mathbb{R}$ is defined as

$$\mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x}) = \begin{cases} \min_{\boldsymbol{u} \in \mathcal{M}_t} \|\boldsymbol{x} - \boldsymbol{u}\|_2, & \boldsymbol{x} \text{ outside } \mathcal{M}_t \\ 0, & \boldsymbol{x} \text{ belonging to } \mathcal{M}_t \\ -\min_{\boldsymbol{u} \in \mathcal{M}_t} \|\boldsymbol{x} - \boldsymbol{u}\|_2, & \boldsymbol{x} \text{ inside } \mathcal{M}_t \end{cases} \tag{1}$$

The zero-level set, and thus the surface of the cell at time t, is represented by all points where $\mathrm{SDF}_{\mathcal{M}_t}(\cdot) = 0$.

3.2. Implicit neural representations

Recent works have shown that the function $\mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x})$ can be approximated using a multi-layer perceptron (MLP) f_θ with trainable parameters θ (Sitzmann et al., 2020; Park et al., 2019). Such an MLP, called an INR, takes a coordinate vector $\boldsymbol{x}$ as input and provides an approximation of $\mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x})$ as output. We here propose to also condition the MLP on a time parameter $t \in \tau$ to provide an approximation of the time-evolving SDF of $\mathcal{M}_t$ for arbitrary $t \in \tau$. Hence, we regress the SDF values at a position at a certain time. In addition, the MLP can be conditioned on a latent space vector $\boldsymbol{z}$ drawn from a multivariate Gaussian distribution with a spherical covariance $\sigma^2 I$, where I is the identity matrix. This latent code can be thought of as a low-dimensional encoding of a shape. By conditioning the MLP on a latent code, we are able to optimize a single model for a distribution of time-evolving shapes.

Combining all these terms results in an MLP $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$ that approximates the time-evolving SDF of the manifold $\mathcal{M}_t$ for arbitrary $t \in \tau$, given latent space vector $\boldsymbol{z}$. Now, we describe how we optimize such a model for cell shape sequences $\mathcal{M}_t$ using an *auto-decoder* strategy (Park et al., 2019).

3.3. Network optimization

We optimize the auto-decoder given a training set consisting of N cell shape sequences $\{\mathcal{M}^i\}_{i=1}^N$. For each cell shape sequence, reference values of its time-evolving SDF are known at a finite and discrete set of points in $\Omega \times \tau$. An important aspect of the auto-decoder model is that not only the parameters θ of the MLP are optimized during training, but also the latent code $\boldsymbol{z}_i$ for each training sequence $\mathcal{M}^i$. The loss function therefore consists of two components. The first component is the reconstruction loss that computes the L_1 distance between reference SDF values and their approximation by the MLP, i.e.,

$$\mathcal{L}_{rec}(f_\theta(\boldsymbol{x}, t, \boldsymbol{z}), \mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x})) = \|f_\theta(\boldsymbol{x}, t, \boldsymbol{z}) - \mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x})\|_1, \tag{2}$$

The second component is given by

$$\mathcal{L}_{code}(\boldsymbol{z}) = \frac{1}{\sigma^2} \|\boldsymbol{z}\|_2^2. \tag{3}$$

This term, with regularization constant $\frac{1}{\sigma^2}$, ensures that a compact latent space is learned and improves the speed of convergence (Park et al., 2019). The parameter σ^2 in the regularization term $\mathcal{L}_{code}$ corresponds to the variance of the Gaussian distribution used for sampling the latent codes. During training of the auto-decoder, we have access to a training set of N cell shape sequences and thus the full loss function is

$$\mathcal{L}(\theta, \{\boldsymbol{z}_j\}_{j=1}^N) = \mathbb{E}_{(\boldsymbol{x}, t)} \left(\sum_{i=1}^N \mathcal{L}_{rec}(f_\theta(\boldsymbol{x}, t, \boldsymbol{z}_i), \mathrm{SDF}_{\mathcal{M}_t^i}(\boldsymbol{x})) + \mathcal{L}_{code}(\boldsymbol{z}_i) \right), \tag{4}$$

where a latent code $\boldsymbol{z}_i$ is optimized for each shape $\{\mathcal{M}^i\}_{i=1}^N$. After optimization, we obtain an MLP f_θ that is able to approximate the SDF of a shape $\mathcal{M}_t^i$, given latent vector $\boldsymbol{z}_i$, spatial coordinates $\boldsymbol{x}$, and time coordinate t, i.e.,

$$f_\theta(\boldsymbol{x}, t, \boldsymbol{z}_i) \approx \mathrm{SDF}_{\mathcal{M}_t^i}(\boldsymbol{x}). \tag{5}$$

3.4. Network architecture

We represent the function $f_\theta(\boldsymbol{x}, t, \boldsymbol{z}_i)$ by an MLP, which can have any finite depth and width. Here, in all experiments, the MLP has 9 hidden layers, each containing 128 units. In the hidden layers, we use a *sine* periodic activation function, which was shown to be able to better represent finely detailed surfaces of complex shapes compared to the commonly used rectified linear unit (ReLU) (Sitzmann et al., 2020; Mildenhall et al., 2020; Wiesner et al., 2022). The weights of

layers using *sine* activations are initialized with a parameter ω_0, which controls the angular frequency of the sine functions. This parameter directly affects the range of frequencies that the model is able to represent, where low values encourage low frequencies and smooth surfaces, and high values favor high frequencies and finely detailed surfaces. As proposed in Sitzmann et al. (2020), we initialize the weight matrices of shape $(c_{\text{out}} \times c_{\text{in}})$ in hidden layers by drawing from a uniform distribution

$$\mathcal{U}\left(-\sqrt{\frac{6}{\omega_0^2\, c_{\text{in}}}}, \sqrt{\frac{6}{\omega_0^2\, c_{\text{in}}}}\right) \tag{6}$$

while we draw the weights in the input layer from

$$\mathcal{U}\left(-\frac{1}{c_{\text{in}}}, \frac{1}{c_{\text{in}}}\right). \tag{7}$$

We inject latent code vectors $\boldsymbol{z}_i$ in the first, fifth, and eighth layer of the network to improve reconstruction accuracy (Park et al., 2019). Moreover, we concatenate the coordinates $\boldsymbol{x}$ and t and inject them into all hidden layers. Preliminary experiments found this to be a requirement for convergence on long spatio-temporal sequences.

3.5. Rotation equivariance

Thus far, we have described a model that can be optimized to jointly learn reconstructions and learn a latent space of cell shape sequences. However, during training, we assign each shape a latent code $\boldsymbol{z}$ and optimize the latent space only for compactness. Therefore, it might happen that a simple rotation of the same cell is described by $f_\theta(\boldsymbol{x}, t, \tilde{\boldsymbol{z}})$ where $\tilde{\boldsymbol{z}} \neq \boldsymbol{z}$ or that there is no latent code representing the rotated shape time series at all. In other words, even though a rotated shape time series is, in essence, the same shape time series, the latent code $\boldsymbol{z}$ does not represent the identity of the shape time series, and the model has to learn the rotated shape time series as well. To make sure that the rotated shape time series are also included in the model and to let $\boldsymbol{z}$ represent the identity of a shape time series, we propose the following **equivariant** model.

Let $\boldsymbol{R} \in \mathrm{SO}(3)$ be a rotation matrix, where $\mathrm{SO}(3)$ is the 3D rotation group representing all rotations about the origin in 3D Euclidean space. Every rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is orthogonal, i.e., its transpose is equal to its inverse that corresponds to a rotation in the opposite direction. Now, assume we rotate a shape described by the zero level set of $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$ via the rotation matrix $\boldsymbol{R}$. The resulting shape is described by $\boldsymbol{R}\{\boldsymbol{x} \mid f_\theta(\boldsymbol{x}, t, \boldsymbol{z}) = 0\} = \{\boldsymbol{R}\boldsymbol{x} \mid f_\theta(\boldsymbol{x}, t, \boldsymbol{z}) = 0\}$ (where we let $\boldsymbol{R}$ act on the whole set by abuse of notation). By substituting $\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}$, we then see that the rotated shape is described by $\{\boldsymbol{x}' \mid f_\theta(\boldsymbol{R}^T\boldsymbol{x}', t, \boldsymbol{z}) = 0\}$, the zero level set of the function $f_\theta(\boldsymbol{R}^T\cdot, t, \boldsymbol{z})$. Hence, to represent rotated shape time series, we only need to apply a (transposed) rotation matrix to $\boldsymbol{x}$ in $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$.

As seen in Fig. 2, we achieve (approximate) equivariance by adding a rotation matrix $\boldsymbol{R}$ to our model that transforms our SDF $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$ to the SDF corresponding to the rotated shape time series $f_\theta(\boldsymbol{R}^T\boldsymbol{x}, t, \boldsymbol{z})$. We optimize this model similarly to the *non*-equivariant model, with some important differences. First, we assign a rotation matrix $\boldsymbol{R}$ to each training time series and optimize this rotation matrix during training. We parametrize $\boldsymbol{R}$ using rotation angles α_i, β_i, γ_i around axes x, y, and z, respectively. Using $\boldsymbol{R}_{\varphi_i}$ as the rotation matrix induced by the rotation angles $\varphi_i = (\alpha_i, \beta_i, \gamma_i)$, the loss function we obtain is:

$$\mathcal{L}(\theta, \{\boldsymbol{z}_j\}_{j=1}^N, \{(\alpha_j, \beta_j, \gamma_j)\}_{j=1}^N) = \mathbb{E}_{(\boldsymbol{x},t)}\left(\sum_{i=1}^N \mathcal{L}_{rec}(f_\theta(\boldsymbol{R}_{\varphi_i}^T\boldsymbol{x}, t, \boldsymbol{z}_i), \mathrm{SDF}_{\mathcal{M}_t^i}(\boldsymbol{x})) + \mathcal{L}_{code}(\boldsymbol{z}_i)\right). \tag{8}$$

We optimize angles α_i, β_i, γ_i along with latent codes $\boldsymbol{z}_i$ and network weights θ. The angles $\alpha_i, \beta_i, \gamma_i$ are initialized from $\mathcal{N}(0, \frac{\pi^2}{64})$ as suggested in Bepler et al. (2019). Note that we do not assume a canonical orientation for cell shapes, and angles can be different for each cell sequence.

3.6. Proving equivariance

So far, we have not discussed the rationale behind calling this new model equivariant. To see this, we note that the latent code and rotation matrix $\boldsymbol{R}$ belonging to a new shape time series $\mathcal{M}_t$ are found by minimizing the following loss function over $\varphi = (\alpha, \beta, \gamma)$ and $\boldsymbol{z}$:

$$\mathbb{E}_{(\boldsymbol{x},t)}\left(\mathcal{L}_{rec}(f_\theta(\boldsymbol{R}_\varphi^T\boldsymbol{x}, t, \boldsymbol{z}), \mathrm{SDF}_{\mathcal{M}_t}(\boldsymbol{x})) + \mathcal{L}_{code}(\boldsymbol{z})\right). \tag{9}$$

or, more precisely, its estimate using M SDF tuples $\{(\boldsymbol{x}_i, t_i, s_i)\}_{i=1}^M$ of $\mathcal{M}_t$ with $s_i = \mathrm{SDF}_{\mathcal{M}_{t_i}}(\boldsymbol{x}_i)$:

$$\frac{1}{M}\sum_{i=1}^M\left(\mathcal{L}_{rec}(f_\theta(\boldsymbol{R}_\varphi^T\boldsymbol{x}_i, t_i, \boldsymbol{z}), s_i) + \mathcal{L}_{code}(\boldsymbol{z})\right) \tag{10}$$

When minimizing the latter loss function over $(\boldsymbol{R}_\varphi, \boldsymbol{z})$, we have the following equivariance property:

Theorem 1. *Let the reconstruction of a shape time series $\mathcal{M}_t$ be obtained by minimizing Eq. (10) over $(\boldsymbol{R}_\varphi, \boldsymbol{z})$. Denote by $\mathcal{M}_t^{\tilde{\boldsymbol{R}}}$ the shape time series $\mathcal{M}_t$ rotated by applying the rotation matrix $\tilde{\boldsymbol{R}}$. Assume $(\boldsymbol{R}, \boldsymbol{z})$ is a solution to the optimization problem for $\mathcal{M}_t$ and let the data points for $\mathcal{M}_t^{\tilde{\boldsymbol{R}}}$ be given by*

$$\{(y_i, t_i, s_i)\}_{i=1}^M = \{(\tilde{\boldsymbol{R}}x_i, t_i, s_i)\}_{i=1}^M.$$

Then, $(\tilde{\boldsymbol{R}}\boldsymbol{R}, \boldsymbol{z})$ is a solution to the optimization problem for $\mathcal{M}_t^{\tilde{\boldsymbol{R}}}$.

A proof of the above theorem and a proof of an infinite sample version of the theorem can be found in the supplementary material. Theorem 1 tells us that when rotating the time series of a shape and the spatial part of the used data $\{(\boldsymbol{x}_i, t_i, s_i)\}_{i=1}^M$ by a rotation matrix $\tilde{\boldsymbol{R}}$, a good reconstruction of the rotated time series can be found by changing the obtained rotation matrix in an equivariant way while leaving the latent code $\boldsymbol{z}$ invariant. In this way, in our equivariant model, the latent code $\boldsymbol{z}$ truly represents the identity of the shape time series while $\boldsymbol{R}$ captures the rotational part.

4. Data

To demonstrate the ability of our proposed method to model different phenomena occurring during the cell cycle, we selected three diverse 3D time-lapse biomedical data sets. First, *Platynereis dumerilii* cells exhibit rapid non-rigid cell shape deformations over time. Second, *C. elegans* cells were selected to demonstrate growth of the embryo and clear divisions of mother cells into their daughters. Third, A549 carcinoma cells feature growing and branching protrusions on a blebbing cell main body. Each cell type thus exhibits distinct shape features that we want to model using the proposed method. The 3D+time data sets were acquired in fluorescence microscopy, and the microscopy images were complemented with full segmentation masks. The segmentation masks for *Platynereis dumerilii* and *C. elegans* cells were produced using automatic image analysis algorithms. The A549 carcinoma cells are computer generated with the segmentation masks produced jointly with the synthetic microscopy images. Here, we describe each data set in more detail and discuss the data preparation procedure used to produce suitable SDF data for training the models.

4.1. Platynereis dumerilii embryo cells

Platynereis dumerilii is a sea worm that lives in tropical coastal areas and reaches a length from 2 to 4 cm when fully grown. It is commonly used in biological evolution studies as a model organism. The fluorescently-stained nuclei of a developing *Platynereis dumerilii* embryo were acquired in its early stage of development. The acquisition was done using a SiMView light sheet microscope with double illumination and double detection objectives (Tomer et al., 2012). Specifically, using

the illumination objectives Olympus XLFLUOR 4×/340/0.28, and the detection objectives Nikon CFI75 LWD 16×/0.8 W. The acquisition was done with a time step of 90 s for 300 time points, making the overall experiment time 7 h and 30 min. The spatial resolution of the images is 700×660×113 voxels, with a voxel size of 0.406×0.406×2.031 μm in the x, y, and z axis, respectively.

4.2. C. elegans embryo cells

Caenorhabditis elegans is a transparent worm living in temperate soil environments and reaching approximately 1 mm in length. Its molecular and evolutionary biology was extensively studied (Brenner, 1974) and, to this day, it is a widely used model organism in biological studies. The nuclei of *C. elegans* embryo cell population were fluorescently stained in its early stage of development (Murray et al., 2008) and acquired using a Zeiss LSM 510 Meta confocal laser scanning microscope with Plan-Apochromat 63×/1.4 (oil) objective lens. The images were acquired with a 60 s time step over 250 time points with the overall experiment time being 4 h and 10 min. The spatial resolution of the acquisition is 708×512×35 voxels with voxel size of 0.09×0.09×1.0 μm. This data set is freely available from the website of the Cell Tracking Challenge (Maška et al., 2023).

4.3. A549 human lung carcinoma cells

The A549 lung carcinoma cell line was cultivated from samples of cancerous human lung tissue. It is commonly used as a model in cancer studies and in the testing and development of drug therapies. This is a synthetic data set of simulated GFP-actin-stained A549 lung cancer cells embedded in a Matrigel matrix (Sorokin et al., 2018). Both the membrane of a cell and its growing and branching filopodial protrusions are fluorescently stained. The acquisition process is simulated using a virtual Zeiss Axiovert 200M inverted fluorescence microscope with a Yokogawa CSU-10 confocal unit and Zeiss 40×/1.30 (oil) objective. The time step is 20 s with 30 time points resulting in the overall experiment duration of 10 min. The spatial resolution is 300×300×300 voxels with a voxel size of 0.125×0.125×0.125 μm. This data set was generated using the CytoPacq web-interface (Wiesner et al., 2019b). We simulated 33 time-lapse sequences, where each sequence captures one growing A549 filopodial cell.

4.4. Data preprocessing

To prepare suitable training data sets, we processed the segmentation masks of the *Platynereis dumerilii* cells, the *C. elegans* cells, and the A549 filopodial cells. We partitioned these masks into voxel volumes, where each voxel volume contains a single cell shape. Moreover, for each cell, we prepared these voxel volumes for the first 30 time points from its inception. Cell shapes were centered at each time point according to their centroid and aligned according to their principal axes. For each cell type, we prepared 33 time-lapse sequences with 30 time points. Note that for the *C. elegans* cells, the prepared voxel volumes contain two daughter cells in the second half of the time-lapse sequence due to the mitosis of a selected mother cell. In this case, each daughter cell was aligned separately.

We subsequently precomputed SDFs for each time point in these time-lapse sequences, i.e., for each sequence we computed 30 three-dimensional SDFs. Each time point was then represented by 256×256×256 discrete SDF point samples. One time-evolving cell is thus represented by 30×256×256×256 samples, defining 30 time points of its 3D shape. These point samples constitute the training data sets, $\mathcal{D}^{SDF}_{Plat}$ of *Platynereis dumerilii* cells, $\mathcal{D}^{SDF}_{Cele}$ of *C. elegans* cells, and $\mathcal{D}^{SDF}_{Filo}$ of A549 filopodial cells. 3D renders of time-evolving shapes from the training sets are shown in Fig. 1. All data preparation and visualization algorithms were implemented in Matlab R2021a.

5. Experiments and results

We present a series of experiments with the proposed auto-decoder and evaluate the results both quantitatively and qualitatively. Specifically, we investigate the reconstruction of cell sequences, synthesis of new shapes with randomly sampled latent codes, temporal interpolation between consecutive time points, and compactness of the learned latent spaces. Finally, we demonstrate how synthetic shapes can be used as input to an image-to-image model that synthesizes corresponding microscopy images.

Training parameters We trained separate models on $\mathcal{D}^{SDF}_{Plat}$, $\mathcal{D}^{SDF}_{Cele}$, and $\mathcal{D}^{SDF}_{Filo}$. All models were trained for 2000 epochs with a batch size of 5. More precisely, we select a time-evolving cell 5 times with repetition and subsequently select an individual time point from each selected time-evolving cell. The weights were initialized with ω_0 set to 30, following the initialization scheme proposed in Sitzmann et al. (2020), and optimized using the Adam optimizer (Kingma and Ba, 2014) with a learning rate 10^{-4}, which was reduced every 350 epochs by a factor of 0.5. The latent codes were initialized randomly from $\mathcal{N}(0, 0.01^2)$, as suggested in Park et al. (2019), and their dimensionality was set to 64. The given hyperparameters were used in all experiments unless stated otherwise. The networks and the respective training and inference procedures were implemented in Python using PyTorch (Paszke et al., 2019) and PyTorch3D (Ravi et al., 2020).

SDF sampling We randomly and non-uniformly sampled 1,000,000 SDF point samples per time point at each training epoch. The sampling procedure is illustrated in Fig. 3. To determine the sampling parameters, we analyzed the training data sets. As stated in Section 4.4, the training data sets consist of 256^3 discrete SDF point samples per time point. The number of point samples (mean ± standard deviation) representing the cell surface and its interior are 260,903 ± 186,188 in $\mathcal{D}^{SDF}_{Plat}$, 515,533 ± 260,459 in $\mathcal{D}^{SDF}_{Cele}$ (considering both daughter cells), and 294,265 ± 41,406 in $\mathcal{D}^{SDF}_{Filo}$. The values were obtained by counting the points x where $\mathrm{SDF}(x) \leq 0$ at each time point over all sequences in the given data set. As the cell shape occupies only a fraction of the considered 3D space, we sampled from the training data sets non-uniformly, as suggested in Park et al. (2019). Specifically, 70% of a training batch is composed of points with distance to the cell surface less or equal to 0.6 μm (including the points with negative distance representing the cell interior) to ensure that the cell boundary and thus the zero level-set is well represented. Conversely, the remaining points with a distance greater than 0.6 μm form the remaining 30% of a batch, as they represent only the empty space around the cell. The non-uniform sampling allowed us to reduce the overall number of point samples and thus decrease the time and memory needed to train the models. Additionally, it also made the training optimization procedure focus more on the important points, as the L_1 distance in the reconstruction loss $\mathcal{L}_{rec}$ is computed against these point samples.

Inference parameters Because the trained model is continuous, it can be used to generate point clouds, meshes, or voxel volumes (Park et al., 2019). In this work, we use binary volumes of size 256×256×256 voxels for quantitative evaluation and meshes obtained using marching cubes (Lorensen and Cline, 1987) for visualization. Each voxel volume and mesh represents a shape of a time-evolving cell at a given time point. With a trained model, generating an SDF of a time-evolving shape with 30×256×256×256 point samples took approximately 15 s on an NVIDIA A100 and required 3 GB of GPU memory. The stated memory consumption represents a forward pass through the MLP. It is composed of the network parameters and corresponding computational graph, along with the network inputs (i.e., a latent code and a grid of spatial and temporal coordinates) and the network outputs representing the inferred SDF values at given spatio-temporal coordinates. The proposed MLP has 168.5k parameters and is thus an order of magnitude smaller than convolutional generative models such as DCGAN (Radford et al., 2015) with 3.5M parameters. For inference, we divide the coordinate grid of 30×256×256×256 points into subsets of 1×32×32×32

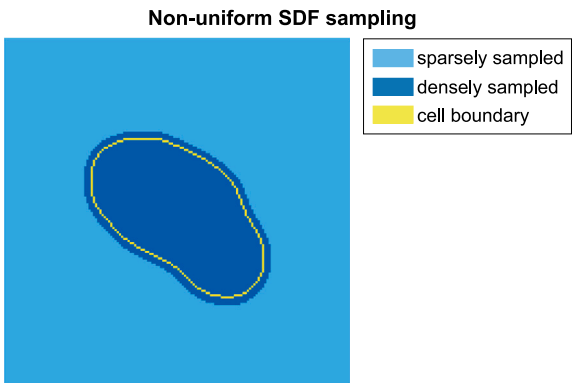


Fig. 3. The figure illustrates the non-uniform SDF sampling process used during training. As the cell shape occupies only a fraction of the space, non-uniform sampling allows to reduce the number of point samples by focusing on the points that contain the most important information.

points. These subsets are given to the MLP, and the network outputs are subsequently concatenated to form the resulting SDF of a time-evolving shape. The size of the grid subset can be set depending on the available GPU memory and has no impact on the quality of resulting SDF as points are processed individually.

Evaluation parameters For the reconstruction of cell shapes, we compute the Dice similarity coefficients (DSC). To compare distributions of cells, we compute descriptive statistics (i.e., mean and interquartile range) of shape descriptors, namely surface [μm^2], volume [μm^3], and sphericity (Costa and Cesar Jr., 2000), where sphericity ranges from 0 to 1, with 1 representing an ideal sphere. Please note that we accounted for the cell division present in $\mathcal{D}^{SDF}_{Cele}$ by splitting each respective voxel volume sequence into two sequences in order to evaluate the shape of each daughter cell separately. Each evaluated sequence of voxel volumes thus represents a single time-evolving cell shape.

5.1. Reconstruction and synthesis of time-evolving shapes

In this experiment, we evaluated the ability of the models to reconstruct a cell shape sequence given its learned latent code $\boldsymbol{z}$ and to synthesize a new cell shape sequence given randomly sampled latent code. A separate model was optimized on $\mathcal{D}^{SDF}_{Plat}$, $\mathcal{D}^{SDF}_{Cele}$, and $\mathcal{D}^{SDF}_{Filo}$ to learn an implicit representation of *Platynereis dumerilii* cells, *C. elegans* cells, and A549 filopodial cells, respectively.

Figs. 4 and 5 show descriptive statistics for cell surface, volume, and sphericity over all time-lapse sequences between real, reconstructed, and randomly generated synthetic shapes produced by the equivariant model and the *non*-equivariant model, respectively. The plots for reconstructed *Platynereis dumerilii* shapes show an excellent overlap of the descriptive statistics. The synthetic shapes fit within the interquartile range of the real data set while keeping similar mean values and generally exhibit lower variability compared to the real ones while retaining the mean volume and surface as the cells grow over time.

The graphs for *C. elegans* show that the descriptive statistics are similar, with a moderate shift toward lower values for the surface of reconstructed shapes. Conversely, we can observe that the sphericity is higher. Note the mitosis that the cells undergo at time point 15. We can observe from the plots that shapes get smaller and less round before growing again in the second part of the time-lapse sequence, and the reconstructed shapes reproduce this behavior very closely. The synthetic shapes similarly show a moderate shift toward the lower surface and volume while exhibiting higher sphericity. The models were able to accurately reproduce the cell division occurring in the middle of the time-lapse sequence. The visualization of the mitotic division can be seen in Fig. 1.

Table 1
p-values of the two-sample Kolmogorov–Smirnov test computed on the shape descriptors of the real and reconstructed shapes produced using the proposed equivariant model.

Cell type	*p*-value		
	Surface	Volume	Sphericity
Platynereis dumerilii	0.634	0.252	0.387
C. elegans	0.077	0.472	<0.001
A549 filopodial	0.064	0.230	<0.001

The reconstructed filopodial cells also exhibit excellent similarity. The reconstruction accurately retains the cell volume. We can observe moderately higher sphericity and lower surface area. This is due to the growing and branching filopodial protrusions that are thin and have sharp edges. The reconstructed filopodial cells exhibit slightly rounder edges and thus lose a moderate amount of their surface area. The synthetic shapes exhibit excellent overlap with respect to the cell volume and, similarly to the reconstruction, a moderately lower surface and higher sphericity.

The results show that the models are able to accurately reproduce cell growth of *Platynereis dumerilii*, mitosis of *C. elegans*, and growing and branching protrusions of A549 lung cancer cells. Randomly generated synthetic shapes exhibit reasonably similar shape features and high visual similarity to the real ones for all three cell lines, as shown by the visual comparison in Fig. 1.

Additionally, Table 1 lists the *p*-values of the two-sample Kolmogorov Smirnov (KS) test computed on the shape descriptors of the real and reconstructed shapes produced using the equivariant model. The KS test retained the null hypothesis ($p > 0.05$) that the shape descriptors are from the same distribution at 5% significance level for all tests except for the sphericity of the synthetic *C. elegans* and A549 human carcinoma cells, reflecting our observations.

Random sampling of latent codes To generate new synthetic cell shapes, we randomly sampled new latent codes within the optimized latent space. For *Platynereis dumerilii* and *C. elegans*, we randomly sampled 33 new latent codes $\boldsymbol{z}$ from $\mathcal{N}(0, 0.001)$. For filopodial cells, noise vectors sampled from $\mathcal{N}(0, 0.0005)$ were added to the optimized latent codes obtained after training.

The parameters of the normal distribution were obtained by analyzing the latent codes of the models trained on $\mathcal{D}^{SDF}_{Plat}$ $\mathcal{D}^{SDF}_{Cele}$, and $\mathcal{D}^{SDF}_{Filo}$ data set. For the first two data sets, the models yielded latent codes with standard deviations of 0.0004 and 0.0008, respectively. We rounded these numbers up to 0.001 to ensure good coverage of the learned latent spaces and used this value to define the normal distribution for sampling new codes. The model trained on $\mathcal{D}^{SDF}_{Filo}$ yielded latent codes with a standard deviation of 0.0007. In this case, randomly sampled latent codes resulted in cells with incorrectly placed protrusions, regardless of the chosen normal distribution. Each cell in the filopodial data set has distinctly growing protrusions, and the shared similarity is only the main body of the cells. By random sampling in the latent space, the model correctly inferred the common features, in this case, the main cell body. However, the model was not able to learn a sufficiently generalized spatial and temporal representation of the protrusions, which vary in placement, branching, and growth direction between all given cells. We experimentally determined that sampling new latent codes within the proximity of already learned latent codes yields correct shapes that exhibit reasonable variability.

5.2. Latent space compactness

In this experiment, we evaluated the impact of the SO(3)-equivariant extension on the compactness of the optimized latent space. In other words, we evaluated the ability of the equivariant model to learn a latent space that is rotation independent. The equivariant extension is depicted in blue in Fig. 2. We trained the *non*-equivariant model (Wiesner et al., 2022) and the proposed equivariant model on

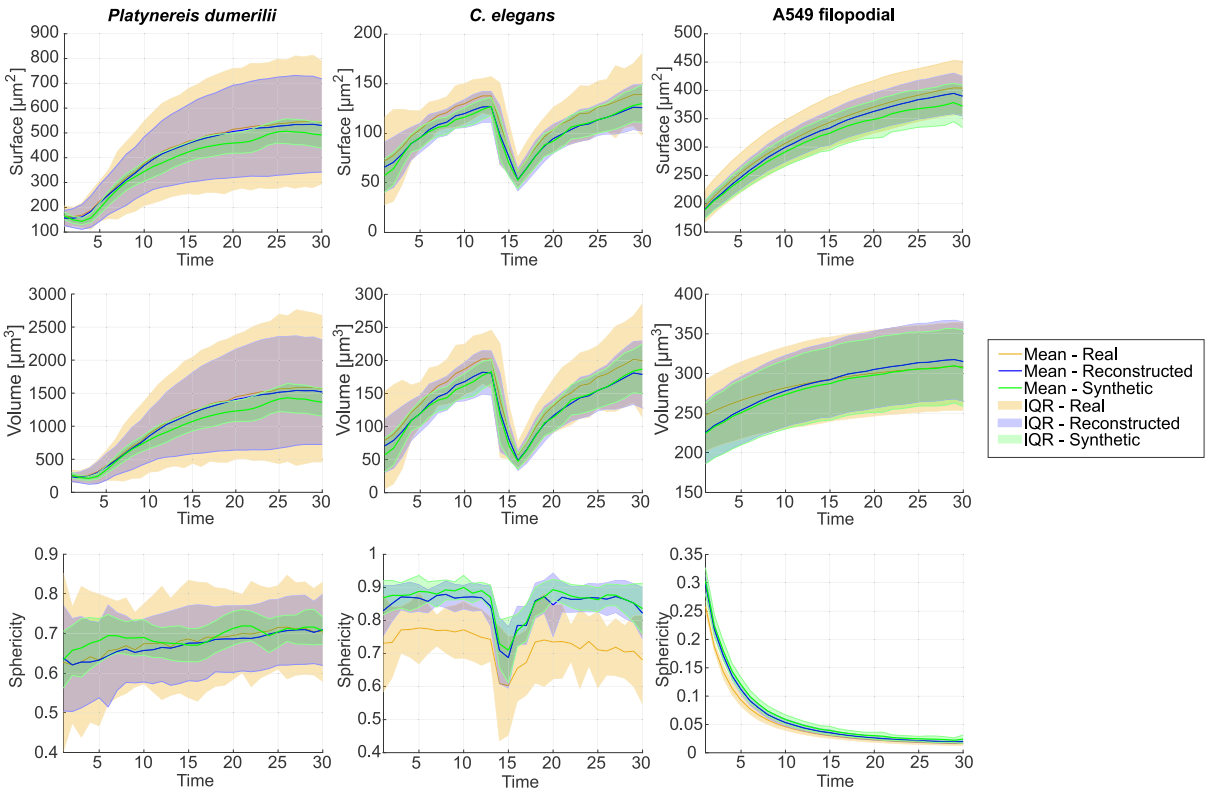


Fig. 4. Quantitative evaluation of reconstructed cell shapes produced using the proposed equivariant model. The plots show descriptive statistics of shape descriptors, i.e., surface [µm^2], volume [µm^3], and sphericity, at each time point in the time-lapse sequence for a given cell type (30 timepoints in total). From the left, the columns represent *Platynereis dumerilii* cells, *C. elegans* cells, and A549 filopodial cells. Real shapes from the training sets are denoted as "Real", shapes reconstructed by the model from the optimized latent codes are denoted as "Reconstructed", and shapes generated by random sampling are denoted as "Synthetic". Each plot shows the mean and interquartile range (IQR) of the respective shape descriptor, with the values computed at each given time point over all time-lapse sequences (33 in total). The closer are the respective colored plotted values together, the more similar the shapes are, where yellow stands for shapes from the real set, blue for the reconstructed shapes, and green for the synthetic shapes.

Table 2
Dice similarity coefficient (DSC) (mean ± standard deviation) of shapes reconstructed using the equivariant and the *non*-equivariant model with respect to latent code dimensionality. Results were computed over reconstructed shapes at time points 30, where the cells are fully grown, over 33 time-lapse sequences. The values show the similarity of the reconstructed cell shapes to the shapes from the training data set. An ideal identical shape would have DSC equal to 1. Cases in which a specific model yielded better results are highlighted in bold.

Latent dimensionality	4	8	16	32	64	128
DSC – equivariant model	**0.780 ± 0.036**	**0.867 ± 0.400**	**0.924 ± 0.320**	**0.954 ± 0.009**	0.958 ± 0.012	**0.959 ± 0.112**
DSC – *non*-equivariant model	0.684 ± 0.270	0.789 ± 0.031	0.872 ± 0.038	0.934 ± 0.027	**0.959 ± 0.015**	0.958 ± 0.015

a set of randomly rotated time-evolving cell shapes. For this experiment, we selected the $\mathcal{D}_{Plat}^{SDF}$ data set. To prepare the training data, we pre-computed 4 random rotations of 25 time-evolving cell shapes from $\mathcal{D}_{Plat}^{SDF}$ to obtain a training set with 100 time-evolving shapes. Specifically, for each rotation, we uniformly sampled angles α, β, and γ from $(-\pi, \pi]$, and rotated the cell shape around the axis x, y, and z, respectively. Please note that the same angles were used at all time points for a given time-evolving shape. The dimensionality of latent codes was set to 64 for both models, as in the other experiments. Furthermore, the number of point samples, sampled randomly from the training data set at each epoch, was decreased to 250,000, and the batch size was increased to 20 to allow the model to observe more shapes at the same time and better optimize their rotations.

After training both models, we evaluated the resulting latent spaces. Each model learned a mapping of 100 time-evolving shapes to 100 locations in the respective latent space. For visual comparison, we computed a low-dimensional representation of the latent spaces using t-SNE and plotted the resulting points in a scatter plot. Fig. 6(a) shows the latent space of the *non*-equivariant model, and Fig. 6(b) shows the latent space of the equivariant model. Rotated variants of each cell are represented by a unique marker. The plots show that the rotated variants of each cell are close together in the latent space of the equivariant model. Furthermore, we computed principal component analysis (PCA) of the latent codes learned by the models and then plotted the variability expressed by the individual principal components in Fig. 7(a). The equivariant model exhibits 99.9% variability in the first 20 principal components compared to 61.2% of the *non*-equivariant model. In other words, PCA shows that it may be possible to further reduce the dimensionality of the latent codes for the equivariant model. To evaluate this hypothesis, we trained both models on the same data set of rotated cells with latent code dimensionalities ranging from 4 to 128. Fig. 7(b) and Table 2 show that the equivariant model retained the reconstruction similarity with latent dimensionality set to higher or equal to 32 (DSC of 0.954 ± 0.009 with dimensionality 32), whereas the *non*-equivariant model required latent dimensionality of 64 or higher to achieve comparable results (DSC of 0.959 ± 0.015 with dimensionality 64, and 0.934 ± 0.027 with dimensionality 32). This experiment provides empirical evidence that the equivariant model is able to reconstruct shapes with higher similarity when the latent code dimensionality is reduced and supports the outcome of the PCA analysis.

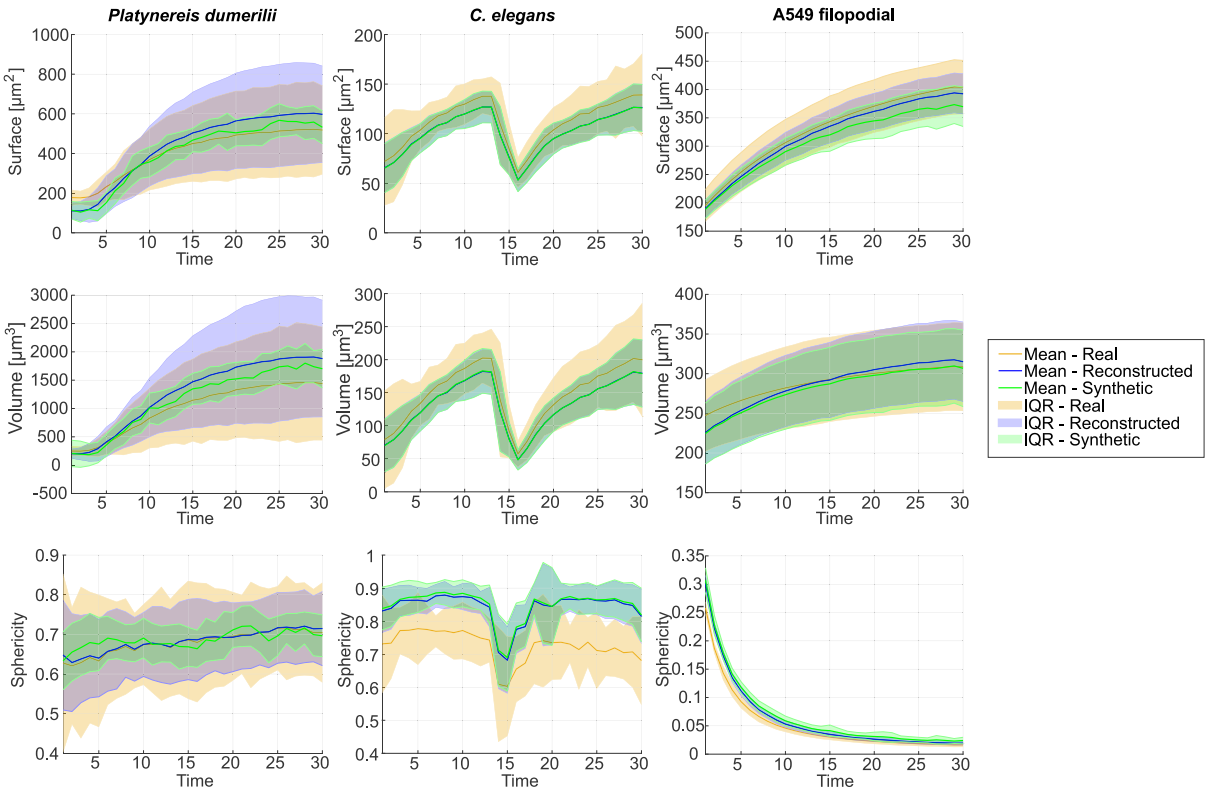
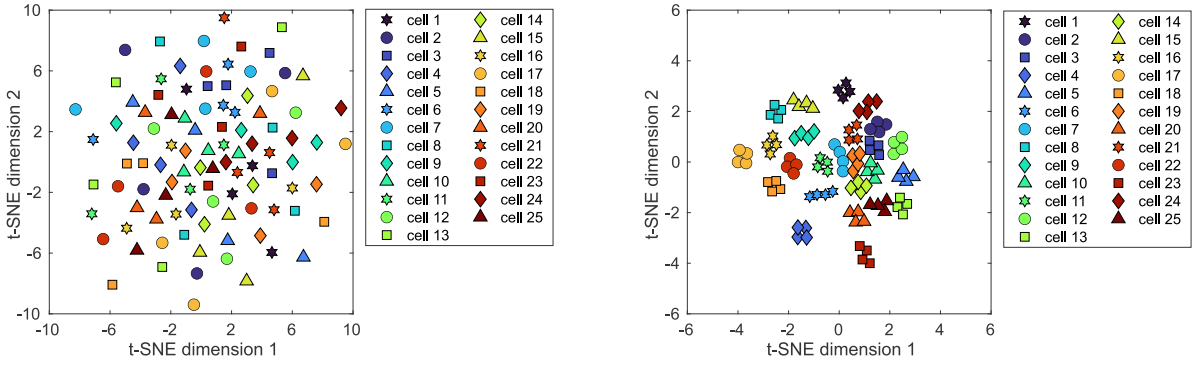


Fig. 5. Quantitative evaluation of reconstructed cell shapes produced using the *non*-equivariant model. The plots show descriptive statistics of shape descriptors, i.e., surface [µm^2], volume [µm^3], and sphericity, at each time point in the time-lapse sequence for a given cell type (30 timepoints in total). From the left, the columns represent *Platynereis dumerilii* cells, *C. elegans* cells, and A549 filopodial cells. Real shapes from the training sets are denoted as "Real", shapes reconstructed by the model from the optimized latent codes are denoted as "Reconstructed", and shapes generated by random sampling are denoted as "Synthetic". Each plot shows the mean and interquartile range (IQR) of the respective shape descriptor, with the values computed at each given time point over all time-lapse sequences (33 in total). The closer are the respective colored plotted values together, the more similar the shapes are, where yellow stands for shapes from the real set, blue for the reconstructed shapes, and green for the synthetic shapes.

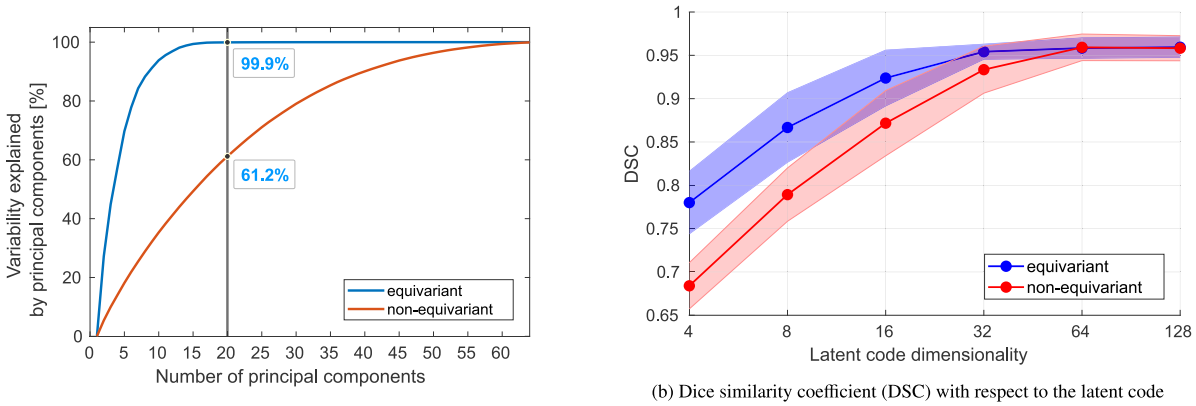


(a) *Non*-equivariant model (b) Equivariant model

Fig. 6. Latent space of the *non*-equivariant model (a) and latent space of the equivariant model (b). The figure shows a comparison of the latent spaces on a low dimensional representation computed using t-SNE. There are 25 time-evolving cell shapes that have each been randomly rotated four times.

5.3. Latent space exploration

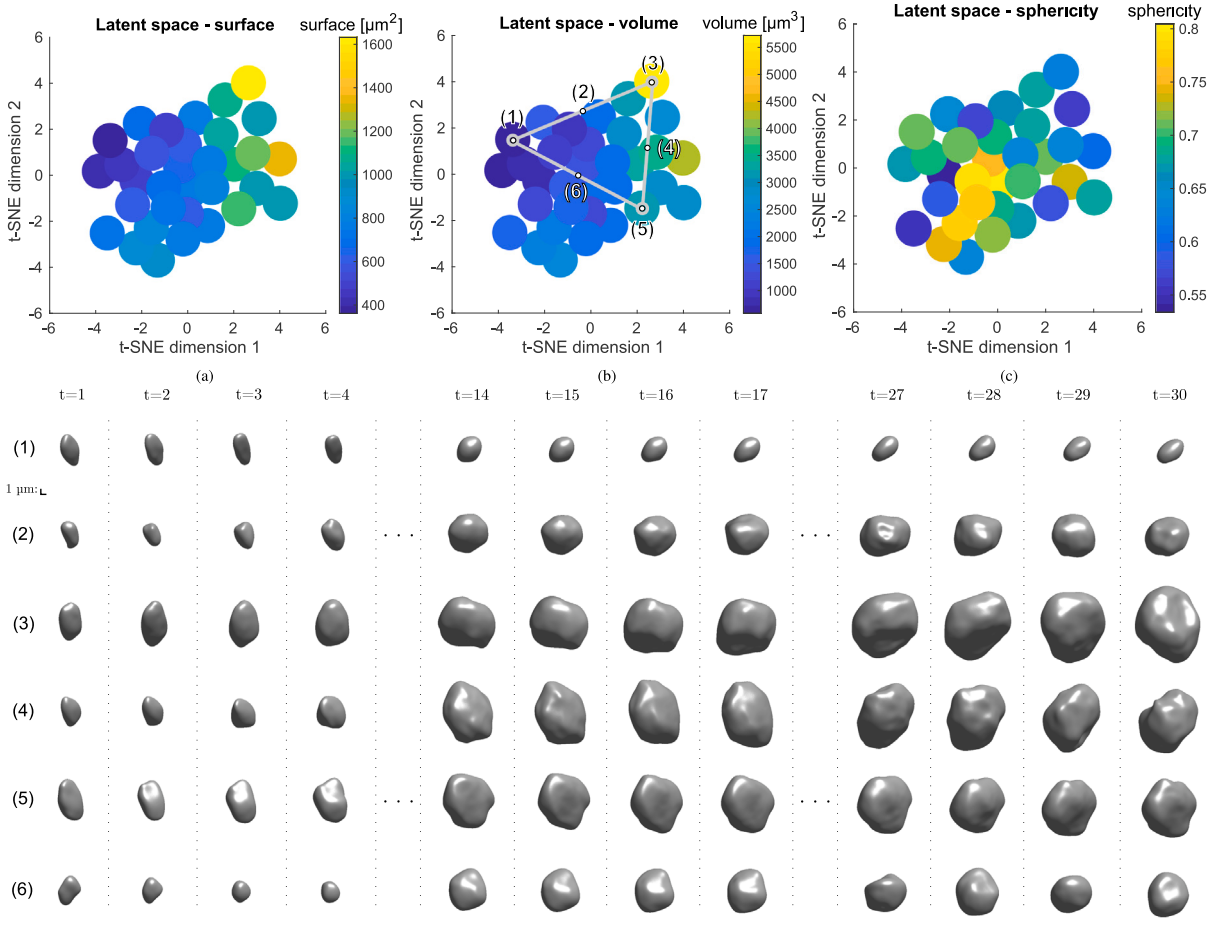
In this experiment, we took a closer look at the latent space of the proposed equivariant model in order to assess its properties and determine how different positions in the latent space, represented by the latent codes, affect the resulting cell shape. We evaluate the latent space of the proposed equivariant model trained on 33 time-evolving shapes in $\mathcal{D}^{SDF}_{PLat}$. The training procedure yielded 33 latent codes with dimensionality of 64, representing a low dimensional embedding of the time-evolving shapes in $\mathcal{D}^{SDF}_{PLat}$. The standard deviation of the resulting latent codes was 0.0008. To visualize the latent space and the positions of the latent codes, we computed a low dimensional representation using t-SNE. Figs. 8(a), 8(b), and 8(c) show scatter plots containing markers representing the positions of individual cells in the latent space with points colored according to the shape descriptors that were computed on the cells, specifically, surface, volume, and sphericity, respectively. As each point in the latent space represents a time-evolving cell at multiple time points, we colored the points according to the mean descriptor values computed over all time points. The plots show that different parts of the learned latent space correspond to cells with different surface, volume, and sphericity characteristics. In Fig. 8(d), we show how different positions in the latent space affect the mean volume of a cell. We sampled latent codes along a linear trajectory, forming a triangle. Here, (1), (3), and (5) correspond to shapes that the model has seen during optimization. (1) represents a cell with the lowest mean volume (586.5 µm^3), (3) a cell with the highest mean volume



(a) Principal component analysis

(b) Dice similarity coefficient (DSC) with respect to the latent code dimensionality

Fig. 7. Principal component analysis (PCA) of the latent codes learned by the equivariant and the *non*-equivariant model (a) and a plot of Dice similarity coefficient (DSC) of real shapes and shapes reconstructed using each model with respect to the latent code dimensionality (b). The PCA plot shows the amount of information present in the latent codes, whereas the DSC plot shows how the latent code dimensionality affects the reconstruction results of the models.



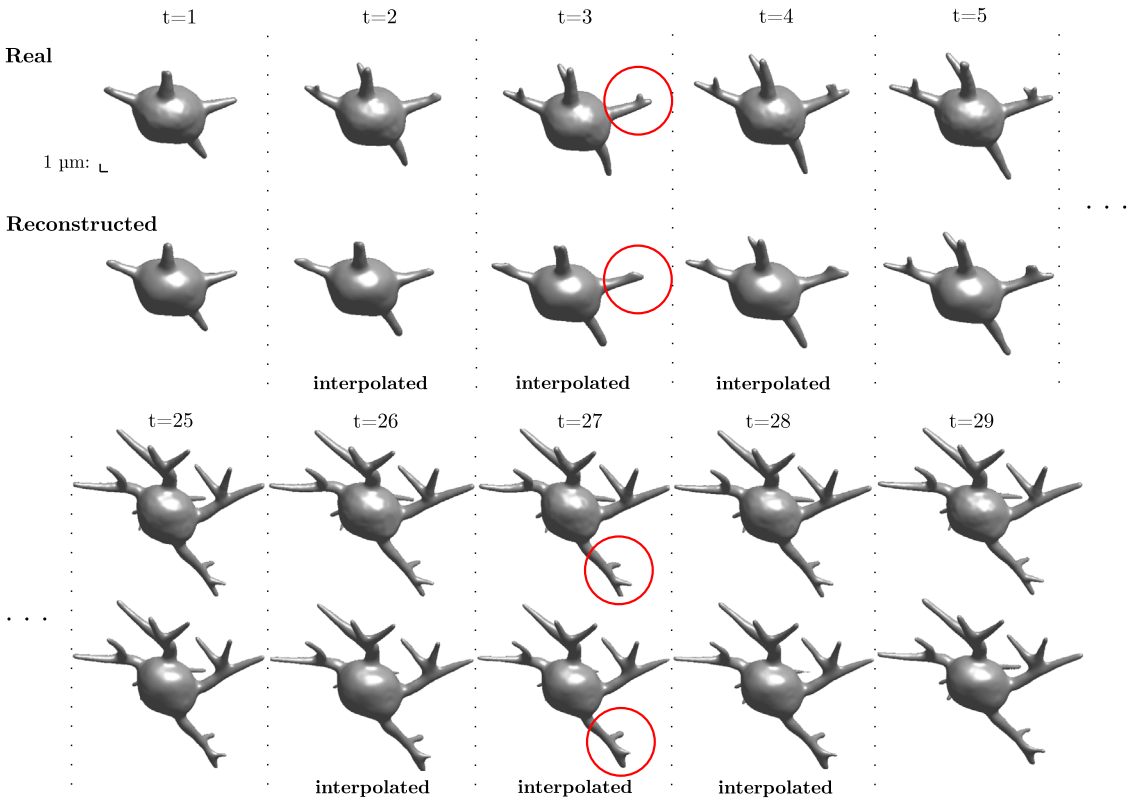
(d) Time-evolving shapes with mean cell volume determined by the latent code

Fig. 8. The figure shows a low dimensional representation of the latent space learned by the equivariant model trained on $\mathcal{D}_{Plat}^{SDF}$ and a visualization of the reconstructed cell shapes sampled from selected locations in the latent space. The low dimensional representation was computed using t-SNE, and the points representing the latent codes were colored according to the mean cell surface (a), volume (b), and sphericity (c). (d) shows a visual comparison of shapes with mean volume determined by the latent code. The latent codes were sampled along linear trajectories, where position (1) represents the lowest mean volume (586.5 μm^3), position (3) the highest mean volume (5719.9 μm^3), and position (5) the average volume (3324.0 μm^3).

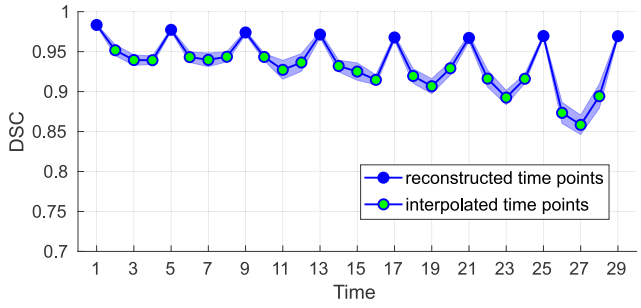
(5719.9 μm^3), and (5) a cell with an average volume (3324.0 μm^3). (2), (4), and (6) are sampled along a linear trajectory between the other ones. The visual comparison shows that the position in the latent space indeed determines the cell volume and that sampling latent codes along a linear trajectory results in a proper change in the cell volume.

5.4. Temporal interpolation

As the proposed shape representation is continuous, the model can be used to produce time-evolving shapes at arbitrary spatial and temporal resolution without the need for additional training. In this



(a) Visual comparison of real and reconstructed shapes



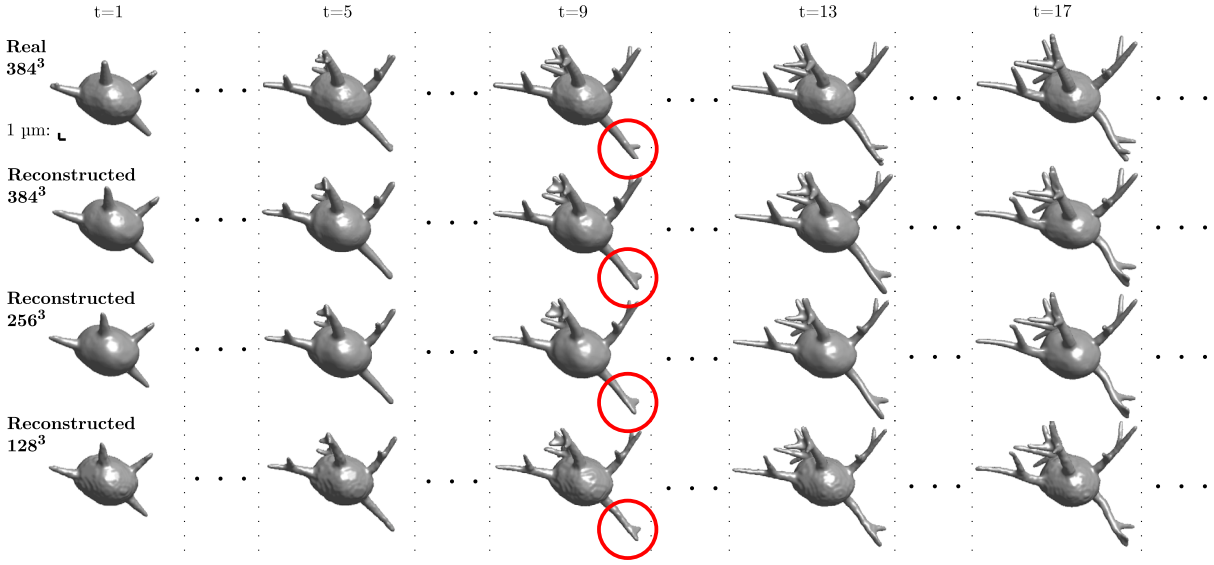
(b) Similarity of real and reconstructed shapes

Fig. 9. We trained the proposed equivariant model on 33 time-evolving shapes in the $\mathcal{D}_{Filo}^{SDF}$ data set, with each sequence limited to every fourth time point. Specifically, we trained on time points $1, 5, 9, \ldots, 29$, making a total of 8 time points per time series. (a) shows a visual comparison of real and reconstructed shapes on a single sequence, with the interpolated time points that the model has not seen explicitly marked as "interpolated". The differences are most apparent on the tips of the protrusions and are marked by a red circle. (b) shows Dice similarity coefficient (DSC) of real and reconstructed shapes computed over all sequences at each time point with the interpolated time points distinctly marked.

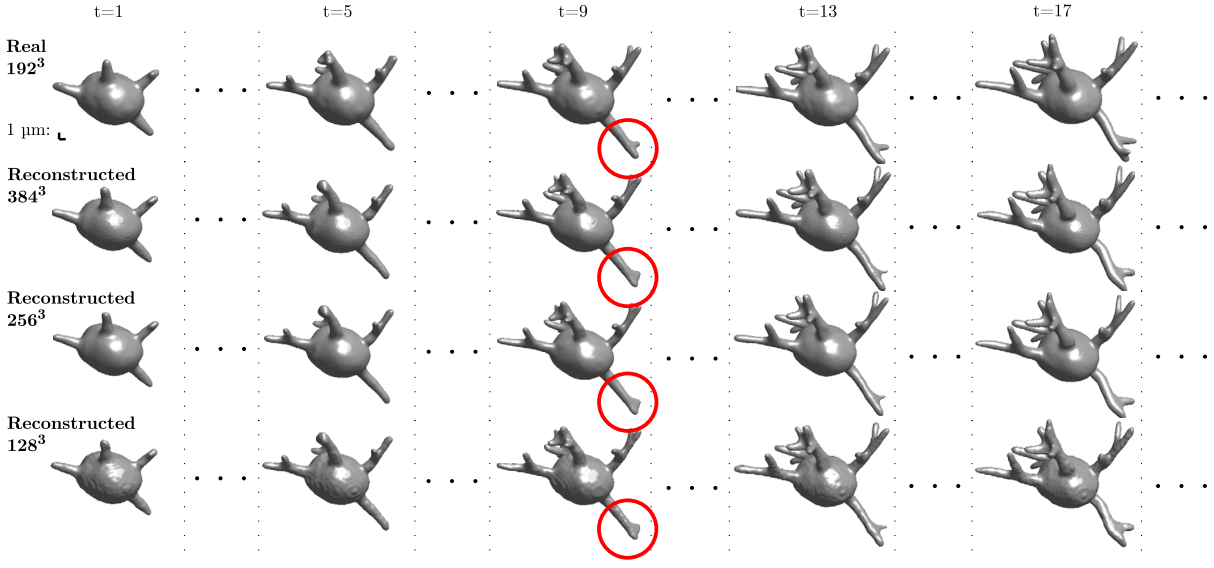
experiment, we evaluated the ability of the model to interpolate in time. We trained the proposed equivariant model on every fourth time point present in the $\mathcal{D}_{Filo}^{SDF}$ data set (i.e., $1, 5, 9, \ldots, 29$), each time-evolving cell was thus represented by 8 time points. In other words, the model has only "seen" 8 time points for each of the 33 cells present in the data set. We subsequently used the model to fill in missing time points and compared the result to the real shapes. As time coordinates used for training and inference are linearly spaced in the interval from -1.0 to 1.0, representing the first and last time point, respectively, we shortened all sequences from 30 to 29 frames by discarding the last frame to ensure that the time coordinates are aligned. Fig. 9(a) shows a visual comparison of real and interpolated cell shapes on a single sequence. As the visual differences are difficult to discern with the naked eye, we computed the similarity of the shapes using DSC, which is shown in Fig. 9(b). The plot shows the mean and standard deviation of DSC computed over all sequences at each time point. The DSC shows that the interpolation similarity decreases toward the end of the sequence, where the cell protrusions are fully grown, and the shape is most complex. The interpolation yielded the lowest DSC of 0.858 ± 0.012 at time point 27. The visual comparison shows that the model is able to interpolate between time points, and DSC corroborates that the produced shapes retain reasonable similarity to the real cells even when interpolating multiple consecutive time points.

5.5. Spatial interpolation and sampling grid

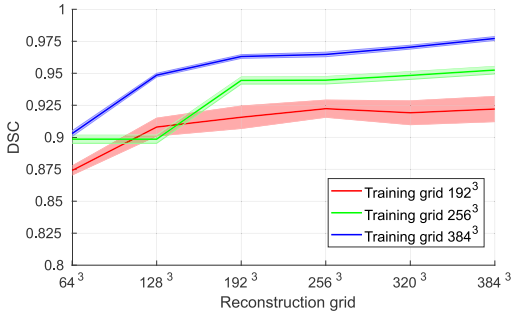
In this experiment, we investigated how the size of the sampling grid affects the shape reconstruction. We trained the proposed equivariant model with shape SDFs represented by three different grids: 192×192×192, 256×256×256, and 384×384×384. As the original microscopy images of A549 filopodial cells are isotropic and exhibit high spatial resolution of 300×300×300 voxels, we chose shapes of this cell line for this experiment. We already had the $\mathcal{D}_{Filo}^{SDF}$ data set consisting of 256×256×256 point samples per time point. To prepare suitable training data sets for the other two models, we computed



(a) Model trained on grid of 384×384×384 samples

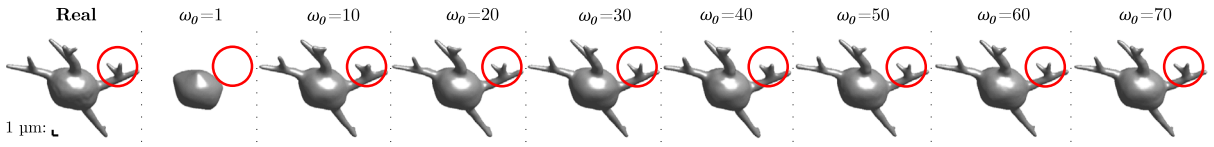


(b) Model trained on grid of 192×192×192 samples

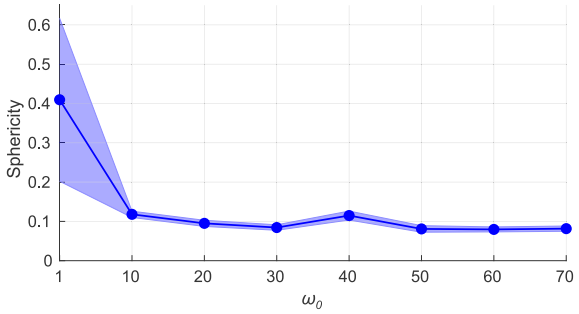


(c) Dice similarity coefficient with respect to the sampling grid

Fig. 10. Reconstructed shapes with respect to the training and reconstruction grids. The proposed equivariant model was trained on D^{SDF}_{Filo} with a grid of 192^3, 256^3, and 384^3. Subsequently, the model was used to reconstruct cell shapes with different grids. Specifically, 384^3, 320^3, 256^3, 192^3, 128^3, and 64^3. Visualization of the reconstruction results showing selected time points of a single time evolving cell is shown for the model trained with a grid 384^3 (a) and grid 192^3 (b). The differences are most prominent on the tips of the protrusions, which are marked by a red circle. The Dice similarity coefficient (DSC) of real and reconstructed shapes with respect to the training and reconstruction grid is shown in (c).



(a) Visual comparison of shapes reconstructed with different values of the ω_0 parameter



(b) Sphericity of shapes reconstructed with different values of the ω_0 parameter

Fig. 11. Comparison of reconstructed shapes with respect to the ω_0 parameter. We trained the proposed equivariant model on the D_{Filo}^{SDF} data set with ω_0 parameter ranging from 1 to 70. (a) shows a visual comparison of real and reconstructed shapes on a single cell at time point 30 where the protrusions are fully grown. The differences between real and reconstructed shapes are most apparent on the tips of the protrusions and are marked by a red circle. (b) shows the mean and standard deviation of sphericity computed at time point 30 over all reconstructed shapes with respect to the ω_0 parameter.

SDFs and created two additional variants of the data set, $D_{Filo}^{SDF_192}$ and $D_{Filo}^{SDF_384}$, with sampling grids of 192×192×192, and 384×384×384 points, respectively. We will refer to resulting models as `model_192`, `model_256`, and `model_384`, with respect to the grid used for training.

After training the models, we reconstructed the cell shapes using different grids, ranging from 64^3 to 384^3. As the shape representation with the proposed model is continuous, the model can be used to reconstruct shapes using an arbitrary grid, isotropic or anisotropic, in other words, to interpolate in space. Figs. 10(a) and 10(b) show a visual comparison of shapes reconstructed using `model_384` and `model_192`, respectively. As the visual differences may be difficult to spot with the naked eye, we also computed DSC for reconstructed cells, including all three models and grid sizes, in Fig. 10(c). DSC was computed over all sequences at time point 30, where the cell protrusions are fully grown. All Reconstructed cell shapes at the given time point were compared against real shapes with the grid of 384^3. Voxel volumes of shapes reconstructed with smaller grids were upsampled using the nearest neighbor algorithm to match the 384^3 grid for DSC computation.

The reconstruction with grid of 384^3 yields the best DSC on all models. The results show that it is possible to further increase the reconstruction similarity by training the model on SDFs with denser grids, as demonstrated by `model_384` trained on $D_{Filo}^{SDF_384}$. Using denser training grids does not affect memory consumption, which is determined only by the batch size and the number of points sampled at each epoch. However, the grid density affects the number of epochs that the models need to converge. Specifically, `model_192` required 1000 epochs, `model_256` 2000 epochs, and `model_384` needed 4000 epochs.

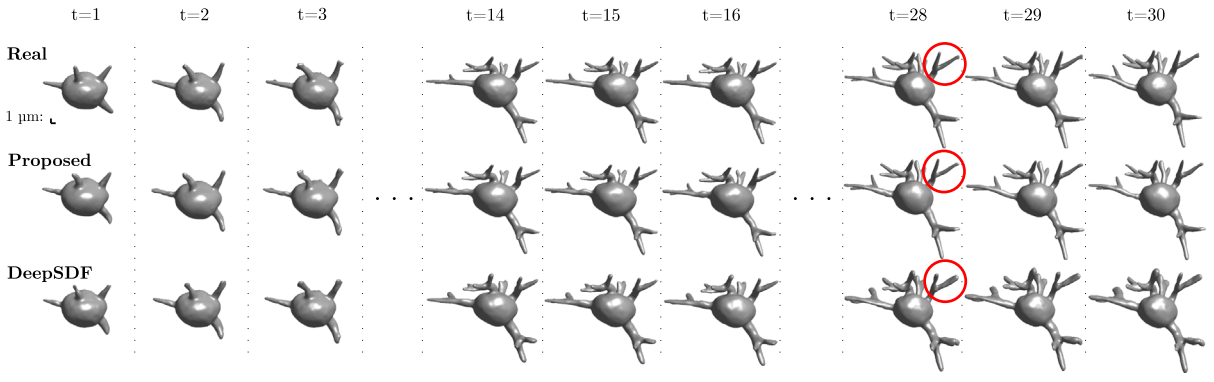
5.6. Spectral bias

In this experiment, we investigate how the ω_0 parameter affects the spectral bias of the model. The weights of layers using periodic activations are initialized using a parameter ω_0 representing the angular frequency of the periodic functions. This parameter was shown to affect the spectral bias of the neural network in Sitzmann et al. (2020), and its recommended value was 30, which is the value that we used in our other experiments. To investigate how the ω_0 parameter affects the shape reconstruction of the proposed equivariant model, we trained multiple models on the D_{Filo}^{SDF} data set with values of ω_0 ranging from 1 to 70. Fig. 11(a) shows a visual comparison of a real cell shape from the training data set and the shapes reconstructed with models trained with different ω_0 values. For visualization, we selected a single shape at time point 30 where the cell protrusions are fully grown, and the cell exhibits the lowest sphericity. The differences are most apparent on the tips of the protrusions, where the models trained with ω_0 lower than 30 produce shapes with a rounder and less defined structure. On the other hand, increasing ω_0 beyond 30 did not seem to yield any visually observable improvements over the default recommended value. Fig. 11(b) shows a plot of sphericity with respect to different ω_0 values computed over all reconstructed cells at time point 30. The results indicate that the ω_0 parameter indeed affects the spectral bias of the model and that increasing this parameter beyond the value recommended by Sitzmann et al. (2020) does not yield any measurable improvements in sharpness of the reconstructed shapes.

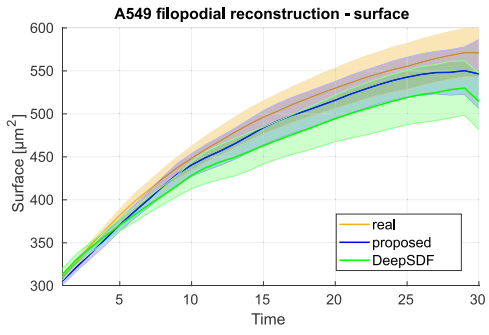
5.7. Comparison with DeepSDF

In this experiment, we compared the proposed equivariant model with DeepSDF (Park et al., 2019). DeepSDF differs from the *non*-equivariant and the equivariant models presented in this study in its architecture. Specifically, DeepSDF uses ReLU activation functions, and the MLP is composed of 8 layers. In comparison, our models have MLP with 9 layers and use *sine* activation functions, where the equivariant model is further extended to be rotation equivariant. Periodic activation functions (*sine*) proposed by Sitzmann et al. (2020) are expected to yield better results in comparison to ReLU activations. Specifically, they should allow the model to converge faster and to better fit high-frequency signals, such as shapes with sharp edges. In order to compare the methods, we used a DeepSDF extended to model shapes in 3D+time (Wiesner et al., 2022).

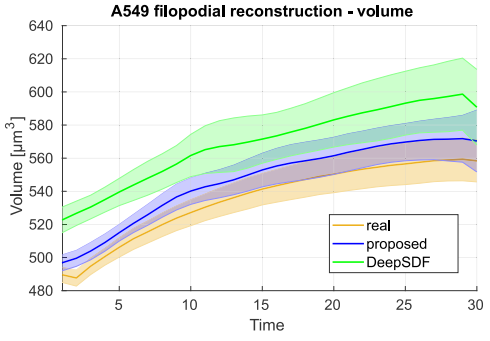
We experimented on the D_{Filo}^{SDF} data set, which consists of 33 complex cell shapes with sharp growing and branching protrusions. Furthermore, we extended the number of training epochs to 4000 to give both models enough time to converge and set the latent code dimensionality of DeepSDF to 256 (Park et al., 2019). We evaluated the results of both models at epoch 4000. A visual comparison of the real and resulting reconstructed cell shapes is shown in Fig. 12(a). The differences are most apparent on the cell protrusions toward the end of the sequence when the cell is fully grown. A visual inspection of the shapes shows that the reconstruction using DeepSDF yields thicker protrusions with rounder tips compared to the proposed model with periodic activations that was able to match the real shapes more closely.



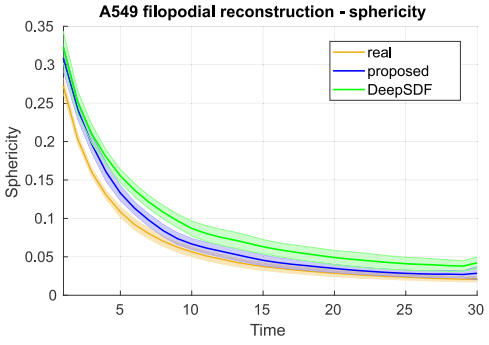
(a) Visual comparison of real and reconstructed shapes



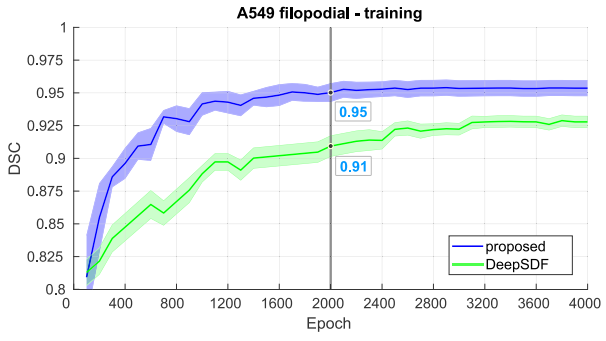
(b) Shape surface over time



(c) Shape volume over time



(d) Shape sphericity over time



(e) Dice similarity coefficient (DSC) of real and reconstructed shapes

Fig. 12. Comparison of the proposed equivariant model with DeepSDF. Both models were trained on the $\mathcal{D}_{Filo}^{SDF}$ data set for 4000 epochs. (a) shows a visual comparison of real and reconstructed shapes on a single sequence. The differences between the shapes are most apparent on the protrusions (red circle), where the shapes reconstructed using the proposed model are sharper and more faithful to the real ones. The plots (b), (c), (d) show mean and standard deviation of surface, volume, and sphericity, respectively, at each time point over all sequences. Where "real" represents real cells from the training data set, "proposed" represents the shapes reconstructed using the equivariant model, and "DeepSDF" represents the shapes reconstructed using DeepSDF. Additionally, we compare the convergence of the models using Dice similarity coefficient (DSC) (e). DSC of real and reconstructed shapes was computed every 100th epoch over all sequences at the last time point (30), where the cell protrusions are fully grown.

We show quantitative evaluation using shape descriptors, specifically, surface in Fig. 12(b), volume in Fig. 12(c), and sphericity in Fig. 12(d), and the training convergence in Fig. 12(e). The quantitative results support the visual inspection, with DeepSDF producing rounder shapes that exhibit higher volume due to the increased thickness of the protrusions. At time point 30 where the cell is fully grown (and exhibits the lowest sphericity), DeepSDF and the proposed model yielded sphericity 0.042 ± 0.008 and 0.029 ± 0.009, respectively, compared to real shapes with 0.021 ± 0.004. To show the training convergence, we computed DSC of real and reconstructed shapes every 100th epoch over all sequences at time point 30. The proposed model was able to converge toward a DSC of 0.950 ± 0.007 in 2000 epochs, whereas DeepSDF yielded DSC of 0.909 ± 0.008 at epoch 2000 and needed another 2000 epochs to converge toward DSC of 0.928 ± 0.004. The results show that the reconstruction similarity and the convergence speed of the proposed equivariant model are measurably improved in comparison to DeepSDF.

5.8. Cell classification based on a latent code

In this experiment, we investigated the application of the proposed equivariant model in a downstream classification task. We optimized a single model on all three data sets, specifically, $\mathcal{D}_{Plat}^{SDF}$, $\mathcal{D}_{Cele}^{SDF}$, and $\mathcal{D}_{Filo}^{SDF}$. Each data set consists of 33 time-evolving cell shapes of a given cell line, making a total of 99 cells. We visualized the latent space of the model using t-SNE in Fig. 13. The shape time series of each cell line form a distinct cluster in the latent space. The model distinguishes between different shape features of each cell line and puts similar shapes close together in the latent space. On data sets where the cell class labels are not known, the cells can be labeled according to their position in the latent space.

5.9. Conditional synthesis of textured cell images

To demonstrate the application of the method, we used synthetic time-evolving shapes (see Section 5.1) to generate synthetic data sets

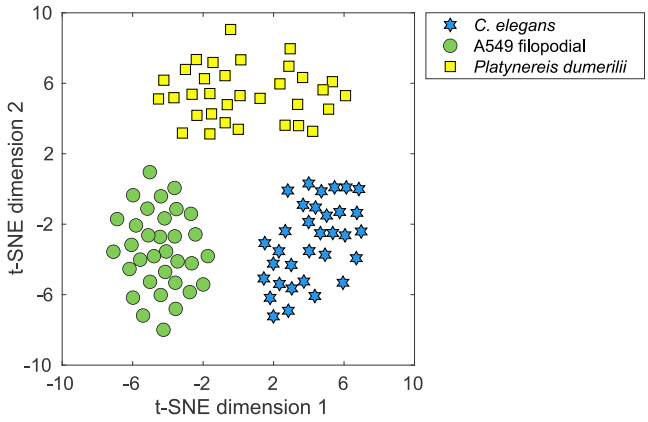


Fig. 13. Cell classification based on a latent code. The figure shows a low dimensional representation of the learned latent space computed using t-SNE. The model was trained on three cell lines, specifically, *Platynereis dumerilii* cells, *C. elegans* cells, and A549 filopodial cells. Each cell line is represented by 33 time-evolving cell shapes, making a total of 99 cells over all three cell lines. Time-evolving cell shapes of each line form separate, distinct clusters in the latent space.

with pairs of a cell shape mask and a corresponding microscopy image. A comparison of real and synthetic microscopy images is shown in Fig. 14. Such data sets are used as training data for segmentation networks or for testing and evaluation of image analysis methods. In the latter case, we refer to them as *benchmarking* data sets.

The segmentation masks were obtained using maximum intensity projections of the voxel volumes of SDFs produced using the proposed method. The textured cell images were generated using a conditional GAN, more specifically pix2pixHD (Wang et al., 2018), which was trained on microscopy images and corresponding cell masks of *Platynereis dumerilii* cells, *C. elegans* cells, and A549 human carcinoma cells. The resulting data sets are produced in 2D+time and contain 33 time-lapse sequences with 30 time points for each cell type.

6. Discussion

In this work, we have proposed a generative model for living cell shapes in 3D+time. We represent evolving cell shapes using the zero-level set of their signed distance function, which is implicitly represented in a fully-connected neural network. This implicit neural representation is fully continuous and thus allows the synthesis of highly detailed shapes in virtually unlimited spatial and temporal resolution. By disentangling shape from rotation, we obtain a compact latent code that allows reconstruction and synthesis of cells with diverse shape and growth characteristics. In a series of experiments, we have shown that this model can be used to accurately reconstruct, synthesize, and interpolate complex and changing cell shapes.

In the proposed model, shape and rotation are disentangled, so that the latent space describes only shape. This has several advantages. First, we showed that this results in learning a more compact latent space, as fewer latent space dimensions might be necessary to describe a data set. Second, by explicitly specifying the rotation, the model is capable of generating cell shapes at any desired angle in 3D space. Third, by allowing the model to focus on the optimization of a latent description of shape, it might be able to learn a latent space in which similar cell shapes are properly clustered. This might allow unsupervised learning based on cell shapes, clustering, and identification of categories of cells. As the latent code describes shape changes over time, it is likely to also capture differences in morphology between cells, and provide insights into cell development which – in turn – can be used for deriving accurate quantitative models, e.g., for embryogenesis. Such flexibility and level of detail would not have been possible with existing voxel-based methods (Svoboda and Ulman, 2017; Fu et al., 2018; Baniukiewicz et al., 2019), nor in our previous work employing implicit neural representations (Wiesner et al., 2022)

In our experiments, we have included three diverse data sets to demonstrate the versatility of our approach: *Platynereis dumerilii* embryo cells, *C. elegans* embryo cells, and A549 lung adenocarcinoma cancer cells. The *Platynereis dumerilii* and *C. elegans* embryo cells are commonly used in biological evolution studies as model organisms. The A549 lung adenocarcinoma cancer cells are used in cancer studies and are subject to active research because filopodia and their relationship to cell migration are of great importance to the testing and development of drug therapies and understanding of the formation of cancer metastases. Our experiments showed that the model is able to accurately represent diverse time-evolving cell shapes and the phenomena occurring during the cell cycle, such as growth and mitosis. In general, descriptive statistics of synthesized cells matched those of real cells. In particular, the surface and volume of real and synthetic cells followed the same distribution. However, we found that the sphericity of synthetic cells was generally higher than that of real cells, indicating that it might still be challenging to properly synthesize details on the surface. A possible mitigation strategy for this might be to allow adaptive shape sampling. One can use sparse sampling where the shape is very smooth, and dense sampling where the shape contains fine details, or simply deserves more attention and thus accuracy, like protrusions on a cell surface. Similarly, we can adaptively sample in time, where high temporal resolution has been shown to improve segmentation and tracking results on time-lapse data (Coca-Rodríguez and Lorenzo-Ginori, 2014).

By random sampling in the optimized latent space, we were able to generate new time-evolving cell shapes with visually plausible features. This random sampling produced consistent results when synthesizing new *Platynereis dumerilii* and *C. elegans* cells. However, in the case of the A549 filopodial cells, we observed that the fully-grown protrusions of the new synthetic shapes had a tendency to disconnect from the main cell body in the second half of the time-lapse sequence. To mitigate this phenomenon, we randomly sampled new latent codes close to known latent codes of our training data. Additional regularization terms could help with optimizing a latent space that would be more suitable for sampling complex and heterogeneous time-evolving shapes, such as the A549 filopodial cells. It is also possible that the complexity of this data set requires a more complex latent space distribution to sample new shapes. In future work, we will investigate how more structure and guarantees can be obtained for the latent space and explore Gaussian mixture models (Reynolds, 2009), which are able to generate more complex data distributions than a single isotropic Gaussian.

A limitation of the current study is that the A549 human lung carcinoma cells that we used to optimize our synthesis model were themselves synthesized (Sorokin et al., 2018) and thus might be slightly different from real living cells. We argue that this is only a minor limitation, as we use these data sets to demonstrate the efficacy of the model and show that our method can indeed synthesize cells that mimic the distribution of these reference cells. Moreover, we only model individual cells, while in reality, cells form populations. In future work, we wish to explore extensions to arrive at a 3D+time model capable of synthesizing living cell populations, describing not only the time-evolving shapes of cells but also cell trajectories and even cell interactions within the population. This would allow the synthesis of data sets that can be used for the development of instance segmentation or tracking models, which can distinguish between cells in a population.

Implicit neural representations are a versatile tool and, depending on the desired application, the inferred SDFs can be converted to mesh-based, voxel-based, or point cloud representations. To demonstrate one potential application of the proposed method, we used a GAN conditioned on the synthetic cell shapes to prepare data sets with pairs of textured cell images and reference annotation for all three cell types. This is a similar approach to those previously used for synthesis of new

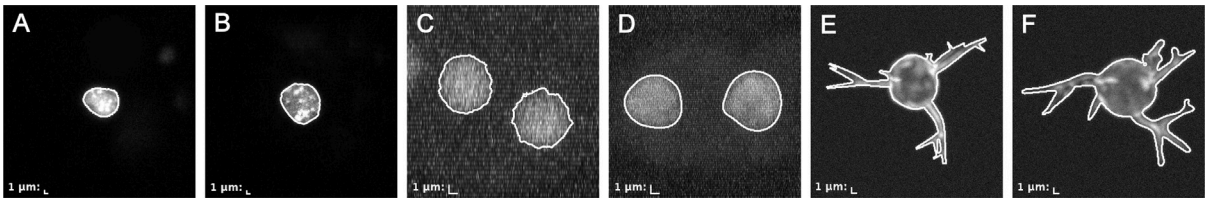


Fig. 14. Comparison of real (A, C, E) and synthetic (B, D, F) images of *Platynereis dumerilii* (A, B), *C. elegans* (C, D), and A549 filopodial cells (E, F). The images show one frame from the respective 2D time-lapse data sets. The segmentation masks are represented as white contours. The masks were obtained using the proposed method and the texture was produced using an image-to-image model.

data (Osokin et al., 2017; Goldsborough et al., 2017; Böhland et al., 2019; Bailo et al., 2019; Baniukiewicz et al., 2019; Kozlovský et al., 2021), and the acquired data sets could be valuable for training, evaluation, and benchmarking of image analysis algorithms. In this study, we demonstrated the proposed method on evolving cell shapes from optical microscopy, but in principle, this approach can be used for learning shapes and spatio-temporal dynamics of diverse organisms at both micro and macro scales, and even shapes visible in different modalities. For example, this method could be adapted to synthesize brain atrophy in patients with Alzheimer's disease or the progression of abdominal aortic aneurysms (Alblas et al., 2023). For unsupervised particle selection in single-particle electron cryo-microscopy workflows, we additionally plan to explore the now-empirically observed capability to recognize similar shapes.

7. Conclusion

We have presented a method that allows accurate spatio-temporal representation and synthesis of highly-detailed time-evolving shapes and structures in microscopy imaging. The method uses a neural network to implicitly represent time-evolving shapes and the occurring visual phenomena and deformations. This representation is fully continuous and equivariant with respect to shape rotations and allows for the synthesis of shapes in virtually unlimited spatial and temporal resolution at any given rotation.

In conclusion, conditional rotation equivariant implicit neural representations are a suitable representation for generative modeling of living cells.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Source code and data sets are publicly available online at https://cbia.fi.muni.cz/research/simulations/implicit_shapes and https://github.com/MIAGroupUT/IMPLICIT-CELL-SURFACES.

Acknowledgments

This work was partially funded by the 4TU Precision Medicine programme supported by High Tech for a Sustainable Future, a framework commissioned by the four Universities of Technology of the Netherlands. Jelmer M. Wolterink was supported by the NWO domain Applied and Engineering Sciences VENI grant (18192). We acknowledge the support by the Ministry of Education, Youth and Sports of the Czech Republic (MEYS CR) (Czech-BioImaging Projects LM2023050 and CZ.02.1.01/0.0/0.0/18_046/0016045). This project has received funding from the European High-Performance Computing Joint Undertaking (JU) and from BMBF/DLR under grant agreement No 955811. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, the Czech Republic, Germany, Ireland, Sweden and the United Kingdom.

The data set of *Platynereis dumerilii* embryo cells is courtesy of Mette Handberg-Thorsager and Manan Lalit, who both have kindly shared it with us.

The shape descriptors were computed and plotted using an online tool for quantitative evaluation, Compyda, available at https://cbia.fi.muni.cz/compyda. We thank its authors T. Nečasová and D. Múčka for kindly giving us early access to this tool and facilitating the evaluation of the proposed method.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.media.2023.102991.

References

Alblas, D., Hofman, M., Brune, C., Yeung, K.K., Wolterink, J.M., 2023. Implicit neural representations for modeling of abdominal aortic aneurysm progression. Functional Imaging and Modeling of the Heart. ISBN: 978-3-031-35302-4, pp. 356–365,

Arbelle, A., Raviv, T.R., 2019. Microscopy cell segmentation via convolutional LSTM networks. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, pp. 1008–1012.

Atzmon, M., Nagano, K., Fidler, S., Khamis, S., Lipman, Y., 2022. Frame averaging for equivariant shape space learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 631–641.

Bailo, O., Ham, D., Shin, Y.M., 2019. Red blood cell image generation for data augmentation using conditional generative adversarial networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1039–1048.

Baniukiewicz, P., Lutton, E.J., Collier, S., Bretschneider, T., 2019. Generative adversarial networks for augmenting training data of microscopic cell images. Front. Comput. Sci. (ISSN: 2624-9898) 1.

Bepler, T., Zhong, E., Kelley, K., Brignole, E., Berger, B., 2019. Explicitly disentangling image content from translation and rotation with spatial-VAE. Adv. Neural Inf. Process. Syst. 32.

Böhland, M., Scherr, T., Bartschat, A., Mikut, R., Reischl, M., 2019. Influence of synthetic label image object properties on GAN supported segmentation pipelines. In: Proceedings 29th Workshop Computational Intelligence. pp. 289–305.

Brenner, S., 1974. The genetics of Caenorhabditis elegans. Genetics 77 (1), 71–94.

Burgos, N., Svoboda, D., 2022. Biomedical Image Synthesis and Simulation: Methods and Applications. Academic Press.

Chen, Y., Staring, M., Wolterink, J.M., Tao, Q., 2023. Local implicit neural representations for multi-sequence MRI translation. In: 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI). pp. 1–5. http://dx.doi.org/10.1109/ISBI53787.2023.10230409.

Coca-Rodríguez, A., Lorenzo-Ginori, J.V., 2014. Effects of interpolation on segmentation in cell imaging. Comput. Sist. 18 (1), 97–109.

Costa, L.d.F.D., Cesar Jr., R.M., 2000. Shape Analysis and Classification: Theory and Practice. CRC Press, Inc..

Coutu, D.L., Schroeder, T., 2013. Probing cellular processes by long-term live imaging–historic problems and current solutions. J. Cell Sci. 126 (17), 3805–3815.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., Guibas, L.J., 2021. Vector neurons: a general framework for SO(3)-equivariant networks. In: IEEE/CVF International Conference on Computer Vision (ICCV).

Ducroz, C., Olivo-Marin, J.-C., Dufour, A., 2012. Characterization of cell shape and deformation in 3D using spherical harmonics. In: 9th International Symposium on Biomedical Imaging (ISBI). IEEE, pp. 848–851.

Dufour, A., Shinin, V., Tajbakhsh, S., Guillén-Aghion, N., Olivo-Marin, J.-C., Zimmer, C., 2005. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. IEEE Trans. Image Process. 14 (9), 1396–1410.

Dunn, K.W., Fu, C., Ho, D.J., Lee, S., Han, S., Salama, P., Delp, E.J., 2019. DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data. Sci. Rep. 9 (1), 1–15.

Erkoç, Z., Ma, F., Shan, Q., Nieß ner, M., Dai, A., 2023. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. arXiv preprint arXiv:2303.17015.

Finzi, M., Welling, M., Wilson, A.G., 2021. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In: Proceedings of the 38 Th International Conference on Machine Learning.

Fu, C., Lee, S., Joon Ho, D., Han, S., Salama, P., Dunn, K.W., Delp, E.J., 2018. Three dimensional fluorescence microscopy image synthesis and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2221–2229.

Ghaffarizadeh, A., Heiland, R., Friedman, S.H., Mumenthaler, S.M., Macklin, P., 2018. PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems. PLoS Comput. Biol. 14 (2), e1005991.

Goldsborough, P., Pawlowski, N., Caicedo, J.C., Singh, S., Carpenter, A.E., 2017. CytoGAN: Generative modeling of cell images. bioRxiv.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y., 2020. Implicit geometric regularization for learning shapes. In: Proceedings of the 37th International Conference on Machine Learning. pp. 3789–3799.

Han, S., Lee, S., Fu, C., Salama, P., Dunn, K.W., Delp, E.J., 2019. Nuclei counting in microscopy images with three dimensional generative adversarial networks. In: Medical Imaging 2019: Image Processing. vol. 10949, International Society for Optics and Photonics. SPIE, pp. 753–763.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kovacheva, V.N., Snead, D., Rajpoot, N.M., 2016. A model of the spatial tumour heterogeneity in colorectal adenocarcinoma tissue. BMC Bioinf. 17 (1), 255.

Kozlovský, M., Wiesner, D., Svoboda, D., 2021. Transfer learning in optical microscopy. In: Svoboda, D., Burgos, N., Wolterink, J.M., Zhao, C. (Eds.), Simulation and Synthesis in Medical Imaging. Springer International Publishing, Cham, ISBN: 978-3-030-87592-3, pp. 77–86.

Kozubek, M., 2016. Challenges and benchmarks in bioimage analysis. Focus on Bio-Image Informatics, ch. 9 231–262, Springer,

Lehmussola, A., Ruusuvuori, P., Selinummi, J., Huttunen, H., Yli-Harja, O., 2007. Computational framework for simulating fluorescence microscope images with cell populations. IEEE Trans. Med. Imaging 26 (7), 1010–1016.

Li, Y., Kim, J., 2016. Three-dimensional simulations of the cell growth and cytokinesis using the immersed boundary method. Math. Biosci. 271, 118–127.

Löffler, K., Mikut, R., 2022. EmbedTrack—Simultaneous cell segmentation and tracking through learning offsets and clustering bandwidths. IEEE Access 10, 77147–77157.

Lorensen, W.E., Cline, H.E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. ACM SIGGRAPH Comput. Graph. 21 (4), 163–169.

Malm, P., Brun, A., Bengtsson, E., 2015. Simulation of bright-field microscopy images depicting pap-smear specimen. Cytometry A 87 (3), 212–226.

Maška, M., Ulman, V., Delgado-Rodriguez, P., Gómez-de Mariscal, E., Nečasová, T., Guerrero Peña, F.A., Ren, T.I., Meyerowitz, E.M., Scherr, T., Löffler, K., et al., 2023. The Cell Tracking Challenge: 10 years of objective benchmarking. Nature Methods 1–11.

Meijering, E., 2012. Cell segmentation: 50 years down the road [life sciences]. IEEE Signal Process. Mag. 29 (5), 140–145.

Meijering, E., 2020. A bird's-eye view of deep learning in bioimage analysis. Comput. Struct. Biotechnol. J. 18, 2312–2325.

Merks, R.M., Glazier, J.A., 2005. A cell-centered approach to developmental biology. Physica A 352 (1), 113–130.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. Springer, pp. 405–421.

Murphy, R.F., 2016. Building cell models and simulations from microscope images. Methods 96, 33–39.

Murray, J.I., Bao, Z., Boyle, T.J., Boeck, M.E., Mericle, B.L., Nicholas, T.J., Zhao, Z., Sandel, M.J., Waterston, R.H., 2008. Automated analysis of embryonic gene expression with cellular resolution in *C. elegans*. Nature Methods 5 (8), 703–709.

Osokin, A., Chessel, A., Carazo Salas, R.E., Vaggi, F., 2017. GANs for biological image synthesis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2233–2242.

Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019. DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., dÁlché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. vol. 32, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Rajaram, S., Pavie, B., Hac, N.E., Altschuler, S.J., Wu, L.F., 2012. SimuCell: a flexible framework for creating synthetic microscopy images. Nat. Methods 9 (7), 634.

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., Gkioxari, G., 2020. Accelerating 3D deep learning with PyTorch3D. arXiv preprint arXiv:2007.08501.

Reynolds, D.A., 2009. Gaussian mixture models. Ency. Biometrics 741 (659–663).

Shen, L., Pauly, J., Xing, L., 2022. NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. IEEE Trans. Neural Netw. Learn. Syst..

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G., 2020. Implicit neural representations with periodic activation functions. Adv. Neural Inf. Process. Syst. 33, 7462–7473.

Sorokin, D.V., Peterlík, I., Ulman, V., Svoboda, D., Nečasová, T., Morgaenko, K., Eiselleová, L., Tesařová, L., Maška, M., 2018. FiloGen: a model-based generator of synthetic 3-D time-lapse sequences of single motile cells with growing and branching filopodia. IEEE Trans. Med. Imaging 37 (12), 2630–2641.

Starruß, J., de Back, W., Brusch, L., Deutsch, A., 2014. Morpheus: a user-friendly modeling environment for multiscale and multicellular systems biology. Bioinformatics 30 (9), 1331–1332.

Stegmaier, J., Arz, J., Schott, B., Otte, J.C., Kobitski, A., Nienhaus, G.U., Strahle, U., Sanders, P., Mikut, R., 2016. Generating semi-synthetic validation benchmarks for embryomics. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). IEEE, pp. 684–688.

Stringer, C., Wang, T., Michaelos, M., Pachitariu, M., 2021. Cellpose: a generalist algorithm for cellular segmentation. Nat. Methods 18 (1), 100–106.

Sun, Y., Liu, J., Xie, M., Wohlberg, B.E., Kamilov, U.S., 2021. CoIL: Coordinate-based internal learning for imaging inverse problems. IEEE Trans. Comput. Imaging 7 (LA-UR-21-21526).

Svoboda, D., Kozubek, M., Stejskal, S., 2009. Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. Cytometry A 75 (6), 494–509.

Svoboda, D., Nečasová, T., 2020. Image-based simulations of tubular network formation. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, pp. 1608–1612.

Svoboda, D., Ulman, V., 2017. MitoGen: A framework for generating 3D synthetic time-lapse sequences of cell populations in fluorescence microscopy. IEEE Trans. Med. Imaging 36 (1), 310–321.

Swat, M.H., Thomas, G.L., Belmonte, J.M., Shirinifard, A., Hmeljak, D., Glazier, J.A., 2012. Multi-scale modeling of tissues using CompuCell3D. Methods Cell Biol. 110, 325–366.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. Adv. Neural Inf. Process. Syst. 33, 7537–7547.

Tomer, R., Khairy, K., Amat, F., Keller, P.J., 2012. Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. Nat. Methods 9 (7), 755–763.

Ulman, V., Svoboda, D., Nykter, M., Kozubek, M., Ruusuvuori, P., 2016. Virtual cell imaging: A review on simulation methods employed in image cytometry. Cytometry A 89 (12), 1057–1072.

Van Liedekerke, P., Palm, M., Jagiella, N., Drasdo, D., 2015. Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. Comput. Part. Mech. 2, 401–444.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8798–8807.

Webb, D., Hamilton, M.A., Harkin, G., Lawrence, S., Camper, A.K., Lewandowski, Z., 2003. Assessing technician effects when extracting quantities from microscope images. J. Microbiol. Methods 53 (1), 97–106.

Wiesner, D., Nečasová, T., Svoboda, D., 2019a. On generative modeling of cell shape using 3D GANs. In: International Conference on Image Analysis and Processing. Springer, pp. 672–682.

Wiesner, D., Suk, J., Dummer, S., Svoboda, D., Wolterink, J.M., 2022. Implicit neural representations for generative modeling of living cell shapes. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2022. Springer Nature Switzerland, Cham, ISBN: 978-3-031-16440-8, pp. 58–67.

Wiesner, D., Svoboda, D., Maška, M., Kozubek, M., 2019b. CytoPacq: A web-interface for simulating multi-dimensional cell imaging. Bioinformatics 35 (21), 4531–4533.

Wolterink, J.M., Zwienenberg, J.C., Brune, C., 2022. Implicit neural representations for deformable image registration. In: International Conference on Medical Imaging with Deep Learning. PMLR, pp. 1349–1359.

Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S., 2022. Neural fields in visual computing and beyond. In: Computer Graphics Forum. vol. 41, Wiley Online Library, pp. 641–676, (2).

David Wiesner is a Ph.D. candidate at the Centre for Biomedical Image Analysis at the Department of visual computing, Faculty of Informatics, Masaryk University, Brno, Czech Republic. He completed his M.Sc. in Computer Science in 2018 at the Faculty of Informatics, Masaryk University. He spent a three-month research stay at the University of Twente, Enschede, The Netherlands, where he worked on cell modeling using implicit neural representations. His current research areas include deep learning, generative modeling, synthesis of biomedical image data, and optimization strategies.

Julian Suk received his M.Sc. degree in Applied Mathematics in 2020 at the Technical University of Munich, Munich, Germany with a thesis on second-order optimization for deep neural networks. Currently he is a Ph.D. student at the Department of Applied Mathematics of the University of Twente, Enschede, The Netherlands. During his Ph.D., his research focusses on applying geometric deep learning to 3D medical data. His research interests lie in deep learning, mathematical optimization, group theory, partial differential equations, and dynamical systems and he likes to combine these fields in the paradigm of scientific machine learning.

Sven Dummer received his M.Sc. degree in Applied Mathematics in 2021 at the University of Twente, Enschede, The Netherlands. Currently, he is a Ph.D. Student at the Mathematics of Imaging and AI group at the Applied Mathematics department of the University of Twente. During his Ph.D., his research focusses on applying mathematical theory in combination with Deep Learning for obtaining more knowledge about microscopic plant cells. His research interests lie in Functional Analysis, Optimal Transport, Differential Geometry, and Group theory, and he likes to apply these in combination with state-of-the-art techniques like Deep Learning.

Tereza Nečasová is a data analyst of biomedical data. She focuses on statistics of biomedical image data and analysis of clinical data, including survival analysis. She graduated with a Master's degree in Computational biology at the Faculty of Science at Masaryk University, Brno, Czech Republic. Now she is a Ph.D. candidate at the Faculty of Informatics, Masaryk University, where she cooperates with the Centre for Biomedical Image Analysis. Her current research area is the validation of synthetic microscopy image data.

Vladimír Ulman is a senior researcher at IT4Innovations National Supercomputing Center at VSB – Technical University of Ostrava, Czech Republic since 2020. He is also with Masaryk University, Brno, Czech Republic where he obtained Ph.D. in computer science in 2011. He is attending to the processing, representing, and visualizing of large image data on several fronts. His interests are ranging from unsupervised parallel processing of the images to collaborative working and sharing of the large image data, and from creating synthetic annotated data for algorithms training to their benchmarking. He has experience with images mostly from confocal and light sheet microscopy.

David Svoboda is an associate professor at the Department of visual computing of the Faculty of Informatics, Masaryk University, Brno, Czech Republic. He completed his Ph.D. in computer science with a thesis on segmentation of volumetric histopathological images. He spent a half-year research visit at Manchester Metropolitan University, Manchester, UK, where he focused on the problems on edge detection using the statistics-based filtering. Since 2006, he has been with the Centre for Biomedical Image Analysis at Masaryk University. His current research fields include the manipulation of huge image data and the generation of synthetic microscopy image data.

Jelmer M. Wolterink is a tenure track assistant professor at the Department of Applied Mathematics of the University of Twente, Enschede, The Netherlands. He obtained his Ph.D. thesis from the University of Utrecht with a thesis on machine learning-based analysis of cardiovascular MR and CT images. Since 2020, he has been with the Mathematics of Imaging & AI group of the University of Twente, where his research interests include novel (geometric) deep learning methods for medical image analysis, with a focus on cardiovascular applications.