
Generative Modeling with Conditional Autoencoders: Building an Integrated Cell

Gregory R. Johnson

Rory M. Donovan-Maiye

Mary M. Maleckar

Allen Institute for Cell Science, 615 Westlake Ave N, Seattle, WA 98109

GREGJ@ALLENINSTITUTE.ORG

RORYDM@ALLENINSTITUTE.ORG

MOLLYM@ALLENINSTITUTE.ORG

Abstract

We present a conditional generative model to learn variation in cell and nuclear morphology and the location of subcellular structures from microscopy images. Our model generalizes to a wide range of subcellular localization and allows for a probabilistic interpretation of cell and nuclear morphology and structure localization from fluorescence images. We demonstrate the effectiveness of our approach by producing photo-realistic cell images using our generative model. The conditional nature of the model provides the ability to predict the localization of unobserved structures given cell and nuclear morphology.

1. Introduction

A central biological principle is that cellular organization is strongly related to function. Location proteomics (Murphy, 2005) addresses this by aiming to determine cell state – i.e. subcellular organization – by elucidating the localization of *all* structures and how they change through the cell cycle, and in response to perturbations, e.g., mutation. However, determining cellular organization is challenged by the multitude of different molecular complexes and organelles that comprise living cells and drive their behaviors (Kim et al., 2014). Currently, the experimental state-of-the-art for live cell imaging is limited to the simultaneous visualization of only a limited number of tagged (2-6 tagged) molecules. Modeling approaches can address this limitation by integrating subcellular structure data from diverse imaging experiments. Due to the number and diversity of subcellular structures, it is necessary to build models that generalize well with respect to both representation and interpretation.

Image feature-based methods have previously been em-

ployed to describe and model cell organization (Boland & Murphy, 2001; Carpenter et al., 2006; Rajaram et al., 2012). While useful for discriminative tasks, these approaches do not explicitly model the relationships between subcellular components, limiting the application to integration of all of these structures.

Generative models are useful in this context. They capture variation in a population and encode it as a probability distribution, accounting for the relationships among structures. Fundamental work has previously demonstrated the utility of expressing subcellular structure patterns as a generative model, which can then be used as a building block for models of cell behavior, i.e. (Murphy, 2005; Donovan et al., 2016).

Ongoing efforts to construct generative models of cell organization are primarily associated with the CellOrganizer project (Zhao & Murphy, 2007; Peng & Murphy, 2011). That work implements a “cytometric” approach to modeling that considers the number of objects, lengths, sizes, etc. from segmented images and/or inverse procedural modeling, which can be particularly useful for both analyzing image content and approaching integrated cell organization. These methods support parametric modeling of many subcellular structure types and, as such, generalize well when low amounts of appropriate imaging data are available. However, these models may depend on preprocessing methods, such as segmentation, or other object identification tasks for which a ground truth is not available. Additionally, there may exist subcellular structures for which a parametric model does not exist or may not be appropriate e.g., structures that vary widely in localization (diffuse proteins), or reorganize dramatically during e.g. mitosis or during a stimulated state (such as microtubules).

Thus, the presence of key structures for which current methods are not well suited motivates the need for a new approach that generalizes well to a wide range of structure localization.

Recent advances in adversarial networks (Goodfellow et al.,

2014) are relevant to our problem. They have the ability to learn distributions over images, generate photo-realistic exemplars, and learn sophisticated conditional relationships; see e.g. Generative Adversarial Networks (Goodfellow et al., 2014), Variational Autoencoders/GAN (Larsen et al., 2015), Adversarial Autoencoders (Makhzani et al., 2015).

Leveraging these recent advances, we present a non-parametric model of cell shape and nuclear shape and location, and relate it to the variation of other subcellular components. The model is trained on data sets of 300–750 fluorescence microscopy images; it accounts for the spatial relationships among these components, their fluorescent intensities, and generalizes well to a variety of localization patterns. Using these relationships, the model allows us to predict the outcome of unobserved experiments, as well as encode complex image distributions into a low dimensional probabilistic representation. This latent space serves as a compact coordinate system to explore variation.

In the following sections, we present the model, a discussion of the training and conditional modeling, and initial results which demonstrate its utility. We then briefly discuss the results in context, current limitations of the work and future extensions.

2. Model Description

Our generative model serves several distinct but complementary purposes. At its core, it is a probabilistic model of cell and nuclear shape (specifically, of cell shape and nuclear shape and *location*) wedged to a probability distribution of structure localization (e.g. the localization of a certain protein) conditional on cell and nuclear shape. This model, *in toto*, can be used both as a classifier for images of localization pattern where the protein is unknown, and as a tool with which one can predict the localization of unobserved structures *de novo*.

The main components of our model are two autoencoders; one which encodes the variation in cell and nuclear shape, and another which learns the relationship between subcellular structures dependent on this encoding.

Notation

The images input and output by the model are multi-channel (see figure 2). Each image x consists of both reference channels r and a structure channel s . Here, the cell and nuclear channels together serve as reference channels, and the structure channel varies, taking on one of the following structure types: α -actinin (actin bundles), α -tubulin (microtubules), β -actin (actin filaments), desmoplakin (desmosomes), fibrillarin (nucleolus), lamin B1 (nuclear membrane), myosin IIB (actomyosin bundles), Sec61 β (endoplasmic reticulum),

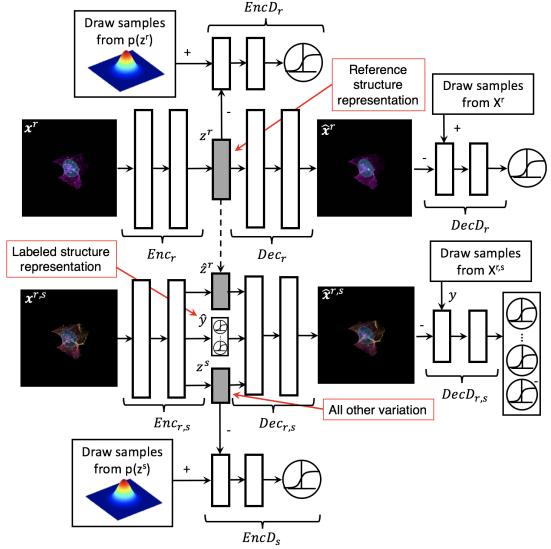


Figure 1: The presented model. The top half of the diagram outlines the reference structure model; the bottom half shows conditional model. The parallel white boxes indicate a nonlinear function. The model is a probabilistic model of cell and nuclear shape (specifically, of cell shape and nuclear shape and *location*) wedged to a probability distribution of structure localization (e.g. the localization of a certain protein) conditional on cell and nuclear shape. This model can be used both as a classifier for images of localization pattern where the protein is unknown, and as a tool for prediction of the localization of unobserved structures *de novo*. The main components are two autoencoders: one encoding the variation in cell and nuclear shape, and another which learns the relationship between subcellular structures dependent on this encoding. See Notation and Model description for details. Figure adapted from (Makhzani et al., 2015)

TOM20 (mitochondria), and ZO1 (tight junctions). We denote which content is being used by the use of superscripts; $\mathbf{x}^{r,s}$ indicates all channels are being used, whereas \mathbf{x}^s indicates only the structure channel is being used, and \mathbf{x}^r only the reference channels. We use y to denotes an index-valued categorical variable indicating which structure type is labeled in \mathbf{x}^s . For example, $y = 1$ might correspond to the α -actinin channel being active, $y = 2$ to the α -tubulin channel, etc. While y is a scalar integer, we also use \mathbf{y} , a one-hot vector representation of y , with a one in the y th element of \mathbf{y} and zeros elsewhere.

2.1. Model of cell and nuclear variation

We model cell shape and nuclear shape using an autoencoder to construct a latent-space representation of these reference channels. The model (figure 1, upper half) attempts to map images of reference channels to a multivariate normal distribution of moderate dimension – here we use a sixteen dimensional distribution. The choice of a normal distribution as the prior for the latent space is in many respects one of convenience, and of small consequence to the model. The nonlinear mappings learned by the encoder and decoder are coupled to both the shape and dimensionality of the latent space distribution; the mapping and the distribution only function in tandem – see e.g. figure 4 in (Makhzani et al., 2015).

The primary architecture of the model is that of an autoencoder, which itself consists of two networks: an encoder Enc_r that maps an image \mathbf{x} to a latent space representation \mathbf{z} via a learned deterministic function $q(\mathbf{z}^r | \mathbf{x}^r)$, and a decoder Dec_r to reconstruct samples from the latent space representation using a similarly learned function $g(\hat{\mathbf{x}}^r | \mathbf{z}^r)$.

We use the following notation for these mappings:

$$\mathbf{z}^r = q(\mathbf{z}^r | \mathbf{x}^r) = \text{Enc}_r(\mathbf{x}^r) \quad (1)$$

$$\hat{\mathbf{x}}^r = g(\hat{\mathbf{x}}^r | \mathbf{z}^r) = \text{Dec}_r(\mathbf{z}^r) \quad (2)$$

where an input image \mathbf{x} is distinguished from a reconstructed image $\hat{\mathbf{x}}$ by the hat over the vector.

2.1.1. ENCODER AND DECODER

The autoencoder minimizes the pixel-wise binary cross-entropy loss of the input and reconstructed input using binary cross entropy,

$$\mathcal{L}_{x^r} = H(\hat{\mathbf{x}}^r, \mathbf{x}^r) \quad (3)$$

where

$$H(\hat{\mathbf{u}}, \mathbf{u}) = -\frac{1}{n} \sum_p u_p \log \hat{u}_p + (1 - u_p) \log (1 - \hat{u}_p) \quad (4)$$

and the sum is over all the pixels p in all the channels in the images \mathbf{u} . We use this function for all images regardless of content (i.e. we use it for \mathbf{x}^r and $\mathbf{x}^{r,s}$)

2.1.2. ENCODING DISCRIMINATOR

In addition to minimizing the above loss function, the autoencoder’s latent space – the output of Enc_r – is regularized by the use of a discriminator EncD_r , the encoding discriminator. This discriminator EncD_r attempts to distinguish between latent space embeddings that are mapped from the input data, and latent space embeddings that are generative drawn from the desired prior latent space distribution (which here is a sixteen dimensional multivariate normal). In attempting to fool the discriminator, the autoencoder is forced to learn a latent space distribution $q(\mathbf{z}^r)$ that is similar in form to the prior distribution $p(\mathbf{z}^r)$ (Makhzani et al., 2015).

The encoding discriminator EncD_r is trained on samples from both the embedding space $\mathbf{z} \sim q(\mathbf{z}^r)$ and from the desired prior $\tilde{\mathbf{z}} \sim p(\mathbf{z}^r)$. We refer to \mathbf{z} as observed samples, and $\tilde{\mathbf{z}}$ as generated samples, and use the subscripts obs and gen to indicate these labels. Trained on these samples, EncD_r outputs a continuous estimate of the source distribution, $\hat{v}^{\text{EncD}_r} \in (0, 1)$.

The objective function for the encoding discriminator is thus to minimize the binary-cross entropy between the true labels v and the estimated labels \hat{v} for generated and observed images:

$$\mathcal{L}_{\text{EncD}_r} = H(\hat{v}_{\text{gen}}^{z^r}, v_{\text{gen}}^{z^r}) + H(\hat{v}_{\text{obs}}^{z^r}, v_{\text{obs}}^{z^r}) \quad (5)$$

2.1.3. DECODING DISCRIMINATOR

The final component of the autoencoder for cell and nuclear shape is an additional adversarial network DecD_r , the decoding discriminator, which operates on the output of the decoder to ensure that the decoded images are representative of the data distribution, similar to that of (Larsen et al., 2015). We train DecD_r on images from the data distribution, $\mathbf{x}_{\text{obs}}^r \sim \mathbf{X}^r$, which we refer to as observed images, and on decoded draws from the latent space, $\mathbf{x}_{\text{gen}}^r \sim \text{Dec}_r(\mathbf{z}^r)$, which we refer to as generated images. The loss function for the decoding discriminator is then:

$$\mathcal{L}_{\text{DecD}_r} = H(\hat{v}_{\text{gen}}^{x^r}, v_{\text{gen}}^{x^r}) + H(\hat{v}_{\text{obs}}^{x^r}, v_{\text{obs}}^{x^r}) \quad (6)$$

2.2. Conditional model of structure localization

Given a trained model of cell and nuclear shape variation from the above network component, we then train a conditional model of structure localization localization upon the learned cell and nuclear shape model. This model (figure 1, lower half) consists of several parts, similar to those above: the core is a tandem encoder $\text{Enc}_{r,s}$ and decoder $\text{Dec}_{r,s}$ that encode and decode images to and from a low dimensional latent space; in addition, a discriminative decoder EncD_s regularizes the latent space, and a discriminative decoder $\text{DecD}_{r,s}$ ensures that the decoded images are similar to the input distribution.

2.2.1. CONDITIONAL ENCODER

The encoder $\text{Enc}_{r,s}$ is given images containing both the reference structure and structures of protein localization, $\mathbf{x}_{r,s}$ and produces three outputs:

$$\hat{\mathbf{z}}^r, \hat{\mathbf{y}}, \mathbf{z}^s = \text{Enc}_{r,s}(\mathbf{x}^{r,s}) = q(\hat{\mathbf{z}}^r, \hat{\mathbf{y}}, \mathbf{z}^s | \mathbf{x}^{r,s}) \quad (7)$$

Here $\hat{\mathbf{z}}^r$ is the reconstructed cell and nuclear shape latent-space representation learned in Section 2.1, $\hat{\mathbf{y}}$ is an estimate of which structure channel was learned, and \mathbf{z}^s is a latent variable that encodes all remaining variation in image content not due to cell/nuclear shape and structure channel. Therefore \mathbf{z}^s is learned dependent on the latent space embeddings of the reference structure, \mathbf{z}^r .

The loss function for the reconstruction of the latent space embedding of the cell and nuclear shape is the mean squared error between the embedding \mathbf{z}^r learned from the cell and nuclear shape autoencoder and the estimate $\hat{\mathbf{z}}^r$ of that embedding produced by the conditional portion of the model:

$$\mathcal{L}_{\hat{\mathbf{z}}^r} = \text{MSE}(\mathbf{z}^r, \hat{\mathbf{z}}^r) = \frac{1}{n} \|\mathbf{z}^r - \hat{\mathbf{z}}^r\|^2 \quad (8)$$

The output $\hat{\mathbf{y}}$ in equation 7 is a probability distribution over structure channels, giving an estimate of the class label for the structure. In our notation, y is an integer value representing the true structure channel, and takes an integer value $1 \dots K$, while \mathbf{y} is the one-hot encoding of that label, a vector of length K equal to 1 at the y th position and 0 otherwise. Similarly, $\hat{\mathbf{y}}$ is a vector of length K whose k th element represents the probability of assigning the label $y = k$.

We use the softmax function to assign these probabilities. In general, the softmax function is given by

$$\text{LogSoftMax}(\mathbf{u}, i) = \log \left(\frac{e^{\mathbf{u}_i}}{\sum_j e^{\mathbf{u}_j}} \right) \quad (9)$$

the loss function for $\hat{\mathbf{y}}$ is then

$$\mathcal{L}_y = -\text{LogSoftMax}(\hat{\mathbf{y}}, y) \quad (10)$$

The final output of the conditional encoder \mathbf{z}^s can be interpreted as a variable that encodes the variation in the localization of the labeled structure independent of cell and nuclear shape.

2.2.2. ENCODING DISCRIMINATOR

The latent variable \mathbf{z}^s is similarly regularized by an adversary EncD_s that enforces the distribution of this latent variable be similar to a chosen prior $p(\mathbf{z}^s)$. The loss function for the adversary takes the same form as equation 5:

$$\mathcal{L}_{\text{EncD}_r} = H(\hat{v}_{\text{gen}}^{z^s}, v_{\text{gen}}^{z^s}) + H(\hat{v}_{\text{obs}}^{z^s}, v_{\text{obs}}^{z^s}) \quad (11)$$

2.2.3. CONDITIONAL DECODER

The conditional decoder $\text{Dec}_{r,s}$ outputs the image reconstruction given the latent space embedding $\hat{\mathbf{z}}^r$, the class estimator $\hat{\mathbf{y}}$, and the structure channel variation \mathbf{z}^s :

$$\hat{\mathbf{x}}^{r,s} = \text{Dec}_{r,s}(\hat{\mathbf{z}}^r, \hat{\mathbf{y}}, \mathbf{z}^s) = g(\mathbf{x}^r | \hat{\mathbf{z}}^r, \hat{\mathbf{y}}, \mathbf{z}^s) \quad (12)$$

The loss function for image reconstruction takes the same form as equation 3, the binary cross entropy between the input and reconstructed image:

$$\mathcal{L}_{\mathbf{x}^{r,s}} = H(\hat{\mathbf{x}}^{r,s}, \mathbf{x}^{r,s}). \quad (13)$$

2.2.4. DECODING DISCRIMINATOR

As in the cell and nuclear shape model, attached to the decoder $\text{Dec}_{r,s}$ is an adversary $\text{DecD}_{r,s}$ intended to enforce that the reconstructed images are similar in distribution to the input images. The output of this discriminator is a vector $\hat{\mathbf{y}}^{\text{DecD}_{r,s}}$ that has $|y| + 1 = K + 1$ output labels, which take a value in $[1, \dots, K, \text{gen}]$. That is, $\hat{\mathbf{y}}^{\text{DecD}_{r,s}}$ has one slot for real images of each particular labeled structure channel, and one additional slot for reconstructed (aka, generated) images of all channels. The loss function is therefore

$$\mathcal{L}_{\text{DecD}_{r,s}} = -\text{LogSoftMax}(\hat{\mathbf{y}}^{\text{DecD}_{r,s}}, y) \quad (14)$$

2.3. Training procedure

The training procedure occurs in two phases. We first train the model of cell and nuclear shape variation, components Enc^r , Dec^r , EncD^r , DecD^r , to convergence (algorithm 1). We then train the conditional model, components $\text{Enc}^{r,s}$, $\text{Dec}^{r,s}$, EncD^s , $\text{DecD}^{r,s}$ (algorithm 2).

In training the model, we adopt three strategies from (Larsen et al., 2015): we limit error signals to relevant networks by propagating the gradient update from any DecD through only Dec , we update decoders with respect Adversarial discrimination of generated and reconstructed images, and we weight the gradient update from the discriminators with the scalars γ_{Enc} and γ_{Dec} . The parameters are therefore updated as follows:

$$\theta_{\text{Enc}_r} \xleftarrow{+} \nabla_{\theta_{\text{Enc}_r}} (\mathcal{L}_{x_r} + \gamma_{\text{Enc}} \mathcal{L}_{\text{EncD}_s}) \quad (15)$$

$$\theta_{\text{Dec}_r} \xleftarrow{+} \nabla_{\theta_{\text{Dec}_r}} (\mathcal{L}_{x_r} + \gamma_{\text{Dec}} \mathcal{L}_{\text{DecD}_s}) \quad (16)$$

$$\theta_{\text{Enc}_{r,s}} \xleftarrow{+} \nabla_{\theta_{\text{Enc}_{r,s}}} (\mathcal{L}_{x_{r,s}} + \mathcal{L}_{\hat{z}_r} + \mathcal{L}_y + \gamma_{\text{Enc}} \mathcal{L}_{\text{EncD}_s}) \quad (17)$$

$$\theta_{\text{Dec}_{r,s}} \xleftarrow{+} \nabla_{\theta_{\text{Dec}_{r,s}}} (\mathcal{L}_{x_{r,s}} + \gamma_{\text{Dec}} \mathcal{L}_{\text{DecD}_{r,s}}) \quad (18)$$

2.4. Integrative Modelling

Beyond encoding and decoding images, we are able to leverage the conditional model of structure localization given cell

Algorithm 1 Training procedure reference structure model

```

 $\theta_{\text{Enc}_r}, \theta_{\text{Dec}_r}, \theta_{\text{EncD}_r}, \theta_{\text{DecD}_r} \leftarrow$  initialize network parameters
repeat
     $X^r \leftarrow$  random mini-batch from reference set
     $Z^r \leftarrow \text{Enc}_s(X^r)$ 
     $\hat{X}^r \leftarrow \text{Dec}_r(\hat{Z}^r)$ 
     $\hat{V}_{\text{gen}}^{\text{EncD}_r} \leftarrow \text{EncD}_r(\tilde{Z}^r)$ 
     $\hat{V}_{\text{obs}}^{\text{EncD}_r} \leftarrow \text{EncD}_r(Z^r)$ 
     $\hat{V}_{\text{obs}}^{\text{DecD}_r} \leftarrow \text{DecD}_r(X^r)$ 
     $\hat{V}_{\text{gen}}^{\text{DecD}_r} \leftarrow \text{DecD}_r(\text{Dec}(\tilde{Z}^r))$ 
     $\mathcal{L}_{\text{DecD}_r} \leftarrow H(\hat{V}_{\text{obs}}^{\text{DecD}_r}, V_{\text{obs}})$ 
        +  $H(\hat{V}_{\text{gen}}^{\text{DecD}_r}, V_{\text{gen}})$ 
     $\theta_{\text{DecD}_r} \leftarrow \nabla_{\theta_{\text{DecD}_r}} \mathcal{L}_{\text{DecD}_r}$ 
     $\mathcal{L}_{\text{EncD}_r} \leftarrow H(\hat{V}_{\text{gen}}^{\text{EncD}_r}, V_{\text{gen}})$ 
        +  $H(\hat{V}_{\text{obs}}^{\text{EncD}_r}, V_{\text{obs}})$ 
     $\theta_{\text{EncD}_r} \leftarrow \nabla_{\theta_{\text{EncD}_r}} \mathcal{L}_{\text{EncD}_r}$ 
     $\mathcal{L}_{\hat{X}^r} \leftarrow H(\hat{X}^r, X^r)$ 
     $\mathcal{L}_{\text{EncD}_r} \leftarrow H(\hat{V}_{\text{obs}}^{\text{EncD}_r}, V_{\text{gen}})$ 
     $\mathcal{L}_{\text{DecD}_r} \leftarrow H(\hat{V}_{\text{gen}}^{\text{DecD}_r}, V_{\text{obs}}) + H(\text{DecD}_r(\hat{X}^r), V_{\text{obs}})$ 
     $\theta_{\text{Enc}_r} \leftarrow \nabla_{\theta_{\text{Enc}_r}} \mathcal{L}_{\hat{X}^r} + \gamma_{\text{Enc}} \mathcal{L}_{\text{EncD}_r}$ 
     $\theta_{\text{Dec}_r} \leftarrow \nabla_{\theta_{\text{Dec}_r}} \mathcal{L}_{\hat{X}^r} + \gamma_{\text{Dec}} \mathcal{L}_{\text{DecD}_r}$ 
until convergence

```

and nuclear shape as a tool to predict the localization of unobserved structures, $p(x^s|x^r, y)$. In particular, we use the maximum likelihood structure localization given the cell and nuclear channels. The procedure for predicting this localization is shown in algorithm 3.

3. Results

3.1. Data Set

For the experiments presented here, we use a collection of 2D segmented cell images generated from a maximum intensity projection of a 3D confocal microscopy data set from human induced pluripotent stem cells gene edited to express mEGFP on proteins that localize to specific structures, e.g. α -actinin (actin bundles), α -tubulin (microtubules), β -actin (actin filaments), desmoplakin (desmosomes), fibrillarin (nucleolus), lamin B1 (nuclear membrane), myosin IIB (actomyosin bundles), Sec61 β (endoplasmic reticulum), TOM20 (mitochondria), and ZO1 (tight junctions). Details of the source image collection are available via the Allen Cell Explorer at <http://allencell.org>. Briefly, each image consists of channels corresponding to the nuclear signal, cell membrane signal, and a labeled sub-cellular structure of interest (see figure 2). Individual cells were segmented, and each channel was processed by subtracting the

Algorithm 2 Training procedure for conditional relationship model

```

 $\theta_{\text{Enc}_{r,s}}, \theta_{\text{Dec}_{r,s}}, \theta_{\text{EncD}_s}, \theta_{\text{DecD}_{r,s}} \leftarrow$  initialize network parameters
repeat
     $X^{r,s}, Y, Z^r \leftarrow$  random mini-batch
        from reference and structure set
     $\hat{Z}^r, \hat{Y}, Z^s \leftarrow \text{Enc}_{r,s}(X^{r,s})$ 
     $\hat{X}^{r,s} \leftarrow \text{Dec}_s(\hat{Z}^r, \hat{Y}, Z^s)$ 
     $\hat{V}_{\text{gen}}^{\text{EncD}_s} \leftarrow \text{EncD}_s(\tilde{Z}^s)$ 
     $\hat{V}_{\text{obs}}^{\text{EncD}_s} \leftarrow \text{EncD}_s(Z^s)$ 
     $\hat{Y}_{\text{obs}} \leftarrow \text{DecD}_{r,s}(X^{r,s})$ 
     $\hat{Y}_{\text{gen}} \leftarrow \text{DecD}_{r,s}(\text{Dec}(\hat{Z}^r, \hat{Y}, \tilde{Z}^s))$ 
     $\mathcal{L}_{\text{EncD}_s} \leftarrow H(\hat{V}_{\text{gen}}^{\text{EncD}_s}, V_{\text{gen}}) + H(\hat{V}_{\text{obs}}^{\text{EncD}_s}, V_{\text{obs}})$ 
     $\theta_{\text{EncD}_s} \leftarrow \nabla_{\theta_{\text{EncD}_s}} \mathcal{L}_{\text{EncD}_s}$ 
     $\mathcal{L}_{\text{DecD}_{r,s}} \leftarrow -\text{LogSoftMax}(\hat{Y}_{\text{obs}}, Y)$ 
        -  $\text{LogSoftMax}(\hat{Y}_{\text{gen}}, Y_{\text{gen}})$ 
     $\theta_{\text{DecD}_{r,s}} \leftarrow \nabla_{\theta_{\text{DecD}_{r,s}}} \mathcal{L}_{\text{DecD}_{r,s}}$ 
     $\mathcal{L}_{\hat{X}^{r,s}} \leftarrow H(\hat{X}^{r,s}, X^{r,s})$ 
     $\mathcal{L}_Y \leftarrow -\text{LogSoftMax}(\hat{Y}, Y)$ 
     $\mathcal{L}_{\hat{Z}^r} \leftarrow \text{MSE}(\hat{Z}^r, Z^r)$ 
     $\mathcal{L}_{\text{EncD}_s} \leftarrow H(\hat{V}_{\text{obs}}^{\text{EncD}_s}, V_{\text{gen}})$ 
     $\mathcal{L}_{\text{DecD}_{r,s}} \leftarrow -\text{LogSoftMax}(\hat{Y}_{\text{gen}}, Y)$ 
        -  $\text{LogSoftMax}(\text{DecD}_{r,s}(\hat{X}^{r,s}), Y)$ 
     $\theta_{\text{Enc}_{r,s}} \leftarrow \nabla_{\theta_{\text{Enc}_{r,s}}} \mathcal{L}_{\hat{X}^{r,s}} + \mathcal{L}_Y + \mathcal{L}_{\hat{Z}^r} + \gamma_{\text{Enc}} \mathcal{L}_{\text{EncD}_s}$ 
     $\theta_{\text{Dec}_{r,s}} \leftarrow \nabla_{\theta_{\text{Dec}_{r,s}}} \mathcal{L}_{\hat{X}^{r,s}} + \gamma_{\text{Dec}} \mathcal{L}_{\text{DecD}_{r,s}}$ 
until convergence

```

Algorithm 3 Structure integration procedure

```

trained  $\text{Enc}_r$  and  $\text{Dec}_{r,s}$ 
 $x^r \leftarrow$  reference structure image
 $z^r \leftarrow \text{Enc}_r(x^r)$ 
for each structure in structures do
     $y \leftarrow$  structure
     $z^s \leftarrow \text{argmax}_{z_s} p(z_s)$ 
     $\hat{x}_{r,s} \leftarrow \text{Dec}_{r,s}(z^r, y, z^s)$ 
    append  $\hat{x}_s$  to  $x_{\text{out}}$ 
end for

```

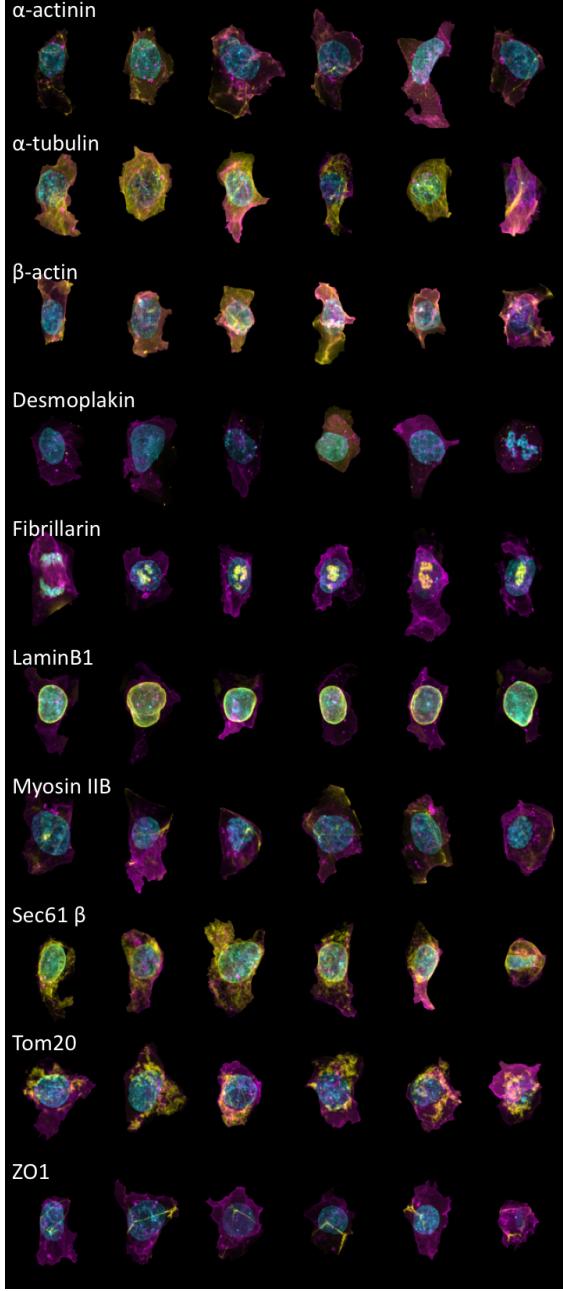


Figure 2: Example images for each of the 10 labeled structures of focus in this paper. Rows correspond to observed microscopy images, used as inputs to the model, for six arbitrary cells, each with a particular fluorescently labeled structure as named, shown in yellow. The reference structures, the cell membrane and nucleus (DNA), are shown in magenta and cyan, respectively. Images have been cropped for visualization purposes. See figure S6a for isolated observed structure channel only.

most populous pixel intensity, zeroing-out negative-valued pixels, rescaling image intensity between 0 and 1, and max-projecting the 3D image along the height-dimension. The cells were aligned by the major axis of the cell shape, and centered according to the center of mass of the segmented nuclear region, and flipped according to image skew. Each of the 6077 cell images were rescaled to $0.317 \mu\text{m}/\text{px}$, and padded to 256×256 pixels. The model took approximately 16 hours to train on one Pascal Titan X GPU.

3.2. Model implementation

A summary of the model architectures is described in Section B. We based the architectures and their implementations on a combination of resources, primarily (Larsen et al., 2015; Makhzani et al., 2015; Radford et al., 2015), and Kai Arulkumaran’s Autoencoders package (Arulkumaran, 2017).

We found that adding white noise to the first layer of decoder adversaries, DecD^r and $\text{DecD}^{r,s}$, stabilizes the relationship between the adversary and the autoencoder and improves convergence as in (Sønderby et al., 2016) and (Salimans et al., 2016).

We choose a sixteen dimensional latent space for both Z^r and Z^s .

3.3. Training

To train the model, we used the Adam optimizer (Kingma & Ba, 2014) to perform gradient-descent, with a batch size of 32, learning rate of 0.0002 for all model components (Enc_r , Dec_r , EncD_r , DecD_r , $\text{Enc}_{r,s}$, $\text{Dec}_{r,s}$, EncD_s , $\text{DecD}_{r,s}$), with γ_{Enc} and γ_{Dec} values of 10^{-4} and 10^{-5} respectively. The dimensionality of the latent spaces Z^r and Z^s were set to 16, and the prior distribution for both is an isotropic gaussian.

We spit the data set into 95% training and 5% test (for more details see table S8), and trained the model of cell and nuclear shape for 150 epochs, and the conditional model for 220 epochs. The model was implemented in Torch7 (Collobert et al., 2011), and ran on an Nvidia Pascal TitanX. The model took approximately 16 hours to train. Further details of our implementation can be found in the software repository.

The training curves for the reference and conditional model are shown in figure S3.

3.4. Experiments

We performed a variety of “experiments” exploring the utility of our model architecture. While quantitative assessment is paramount, the nature of the data makes qualitative assessment indispensable as well, and we include experiments

of this type in addition to more traditional measures of performance.

3.4.1. IMAGE RECONSTRUCTION

A necessary but not sufficient condition for our model to be of use is that the images of cells reconstructed from their latent space representations bear some semblance to the native images. Examples of image reconstruction from the training and test set are shown in figure S1 for our reference structures and figure S2 for the structure localization model. As seen in the figures, the model is able to recapitulate the essential localization patterns in the cells, and produce accurate reconstructions in both the training and test data.

3.4.2. LATENT SPACE REPRESENTATION

We explored the generative capacity of our model by mapping out the variation in cell morphology due to traversal of the latent space. Since the latent spaces in our model are sixteen dimensional and isotropic, dimensionality reduction techniques are of little value, and we resorted to mapping 2D slices of the space.

To demonstrate this variation is smooth, we plot the first two dimensions of the latent space for cell and nuclear shape variation are shown in figure S4. The first two dimensions of the latent space for structure variation are shown in figure S5. In both figures, the orthogonal dimensions are set to their MLE value of zero.

3.4.3. IMAGE CLASSIFICATION

While classification is not our primary use-case, it is a worthwhile benchmark of a well-functioning multi-class generative model. To evaluate the performance of the class-label identification of $\text{Enc}^{r,s}$ we compared the results of the predicted labels and true labels on our hold out set. A summary of the results of our multinomial classification task is shown in table S9. As seen in the table, our model is able to accurately classify most structure, and has trouble only on the poorly sampled or underrepresented classes.

3.4.4. INTEGRATING CELL IMAGES

Conditional upon the cell and nuclear shape, we predict the most likely position of any particular structure via algorithm 3. Some examples of the maximum likelihood estimate of structure localization given cell and nuclear shapes is shown in figure 3.

4. Discussion

Building models that capture relationships between the morphology and organization of cell structures is a difficult problem. While previous research has focused on con-

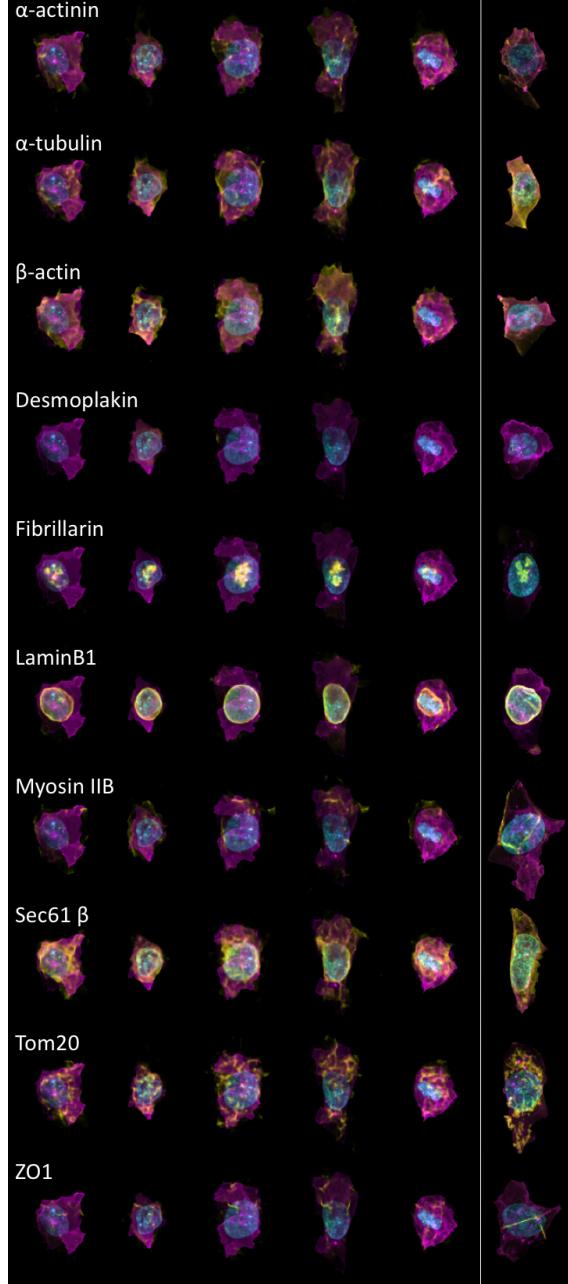


Figure 3: Most probable localization patterns predicted for selected cells for each structure (rows, top to bottom, structure as labeled, shown in yellow). The first 5 columns show the maximum likelihood of localization for each structure, given the cell and nuclear shape. The last column (far right) shows an experimentally observed cell with that labeled structure for comparison. As before, reference structures, cell membrane and nucleus (DNA), are in magenta and cyan, respectively. Images have been cropped for visualization purposes. Note for example how fibrillarin resides within the DNA, and lamin B1 surrounds the DNA. See figure S6b for structure channel only.

structing application-specific parametric approaches, due to the extreme variation in localization among different structures, these approaches may not be convenient to employ for all structures under all conditions. Here, we have presented a nonparametric conditional model of structure organization that generalizes well to a wide variety of localization patterns, encodes the variation in cell structure and organization, allows for a probabilistic interpretation of the image distribution, and generates high quality synthetic images.

Our model of cell and subcellular structure differs from previous generative models (Zhao & Murphy, 2007; Peng & Murphy, 2011; Johnson et al., 2015): we directly model the localization of fluorescent labels, rather than the detected objects and their boundaries. While object segmentation can be essential in certain contexts, and helpful in others, when these approaches are not necessary, it can be advantageous to omit these non-trivial intermediate steps. Our model does not constitute a “cytometric” approach (i.e. counting objects), but due to the fact that we are directly modeling the localization of signal, we drastically reduce the modeling time by minimizing the amount of segmentation and the task of evaluating this segmentation with respect to the “ground truth”.

Even considering these differences, our model is compatible with existing frameworks and will allow for mixed parametric and non-parametric localization relationships, where our model can be used for predicting localization of structures when an appropriate parametric representation may not exist.

Our model permits several straightforward extensions, including the obvious extension to modeling cells in three dimensions. Because of the flexibility of our latent-space representation, we can potentially encode information such as position in the cell cycle, or along a differentiation pathway. Given sufficient information, it would be possible to encode a representation of “structure space” to predict the localization of unobserved structures, or “perturbation space”, such as in (Paolini et al., 2006), and potentially couple this with active learning approaches (Naik et al., 2016) to build models that learn and encode the localization of diverse subcellular structures under different conditions.

Software and Data

The code for running the models used in this work is available at https://github.com/AllenCellModeling/torch_integrated_cell

The data used to train the model is available at <s3://aics.integrated.cell.arxiv.paper.data>.

Acknowledgements

We would like to thank Robert F. Murphy, Julie Theriot, Rick Horwitz, Graham Johnson, Forrest Collman, Sharmishtaa Seshamani and Fuhui Long for their helpful comments, suggestions, and support in the preparation of the manuscript.

Furthermore, we would like to thank all members of the Allen Institute for Cell Science team, who generated and characterized the gene-edited cell lines, developed image-based assays, and recorded the high replicate data sets suitable for modeling. We particularly thank Liya Ding for segmentation data. These contributions were absolutely critical for model development.

We would like to thank Paul G. Allen, founder of the Allen Institute for Cell Science, for his vision, encouragement and support.

Author Contributions

GRJ conceived, designed and implemented all experiments. GRJ, RMD, and MMM wrote the paper.

References

- Arulkumaran, Kai. Autoencoders, 2017. URL <https://github.com/Kaixhin/Autoencoders>.
- Boland, Michael V and Murphy, Robert F. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela cells. *Bioinformatics*, 17(12):1213–1223, 2001.
- Carpenter, Anne E, Jones, Thouis R, Lamprecht, Michael R, Clarke, Colin, Kang, In Han, Friman, Ola, Guertin, David A, Chang, Joo Han, Lindquist, Robert A, Moffat, Jason, Golland, Polina, and Sabatini, David M. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.
- Collobert, Ronan, Kavukcuoglu, Koray, and Farabet, ClÃrment. Torch7: A matlab-like environment for machine learning, 2011.
- Donovan, Rory M, Tapia, Jose-Juan, Sullivan, Devin P, Faeder, James R, Murphy, Robert F, Dittrich, Markus, and Zuckerman, Daniel M. Unbiased rare event sampling in spatial stochastic systems biology models using a weighted ensemble of trajectories. *PLoS computational biology*, 12(2):e1004611, 2016.
- Goodfellow, Ian J, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative Adversarial Networks. *arXiv.org*, June 2014.
- Johnson, G R, Buck, T E, Sullivan, D P, Rohde, G K, and Murphy, R F. Joint modeling of cell and nuclear shape variation. *Molecular Biology of the Cell*, 26(22):4046–4056, November 2015.
- Kim, Min-Sik, Pinto, Sneha M, Getnet, Derese, Nirujogi, Raja Sekhar, Manda, Srikanth S, Chaerkady, Raghothama, Madugundu, Anil K, Kelkar, Dhanashree S, Isserlin, Ruth, Jain, Shobhit, et al. A draft map of the human proteome. *Nature*, 509(7502): 575–581, 2014.
- Kingma, Diederik P and Ba, Jimmy. Adam: A Method for Stochastic Optimization. *arXiv.org*, December 2014.
- Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv.org*, December 2015.
- Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, Goodfellow, Ian, and Frey, Brendan. Adversarial Autoencoders. *arXiv.org*, November 2015.
- Murphy, R F. Location proteomics: a systems approach to subcellular location. *Biochemical Society transactions*, 33(Pt 3):535–538, June 2005.
- Naik, Armaghan W, Kangas, Joshua D, Sullivan, Devin P, and Murphy, Robert F. Active machine learning-driven experimentation to determine compound effects on protein patterns. *eLife*, 5:e10047, February 2016.
- Paolini, Gaia V, Shapland, Richard H B, van Hoorn, Willem P, Mason, Jonathan S, and Hopkins, Andrew L. Global mapping of pharmacological space. *Nature biotechnology*, 24(7):805–815, July 2006.
- Peng, Tao and Murphy, Robert F. Image-derived, three-dimensional generative models of cellular organization. *Cytometry Part A*, 79A(5):383–391, April 2011.
- Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv.org*, November 2015.
- Rajaram, Satwik, Pavie, Benjamin, Wu, Lani F, and Altschuler, Steven J. PhenoRipper: software for rapidly profiling microscopy images. *Nature Methods*, 9(7):635–637, June 2012.
- Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved Techniques for Training GANs. *arXiv.org*, June 2016.
- Sønderby, Casper Kaae, Caballero, Jose, Theis, Lucas, Shi, Wenzhe, and Huszár, Ferenc. Amortised MAP Inference for Image Super-resolution. *arXiv.org*, October 2016.
- Zhao, Ting and Murphy, Robert F. Automated learning of generative models for subcellular location: Building blocks for systems biology. *Cytometry Part A*, 71A(12): 978–990, 2007.

A. Supplementary Figures

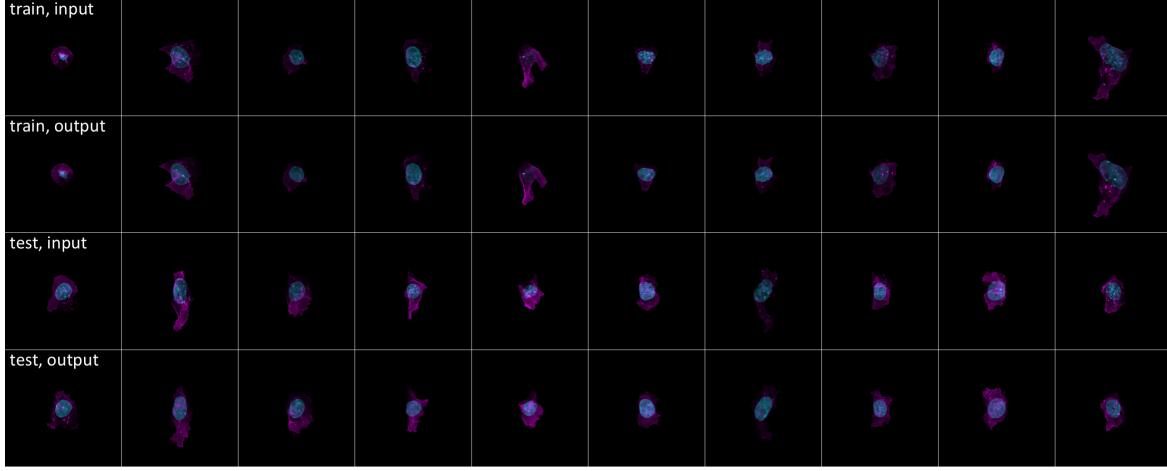


Figure S1: Image input (rows 1 and 3) and reconstruction (rows 2 and 4) from the reference model, showing training set (above two rows), and test set (bottom two rows).

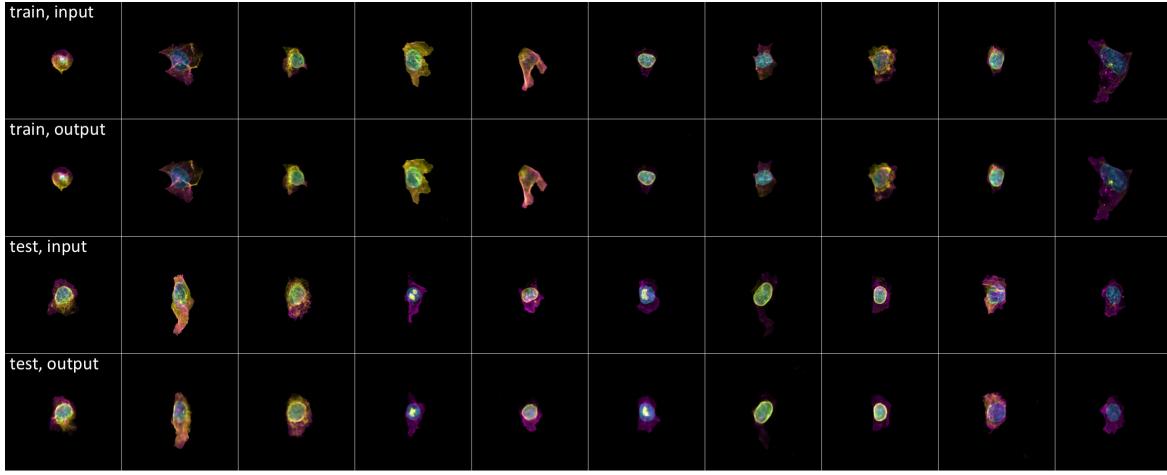


Figure S2: Image input (rows 1 and 3) and reconstruction (rows 2 and 4) from the structure model, showing training set (above two rows), and test set (bottom two rows).

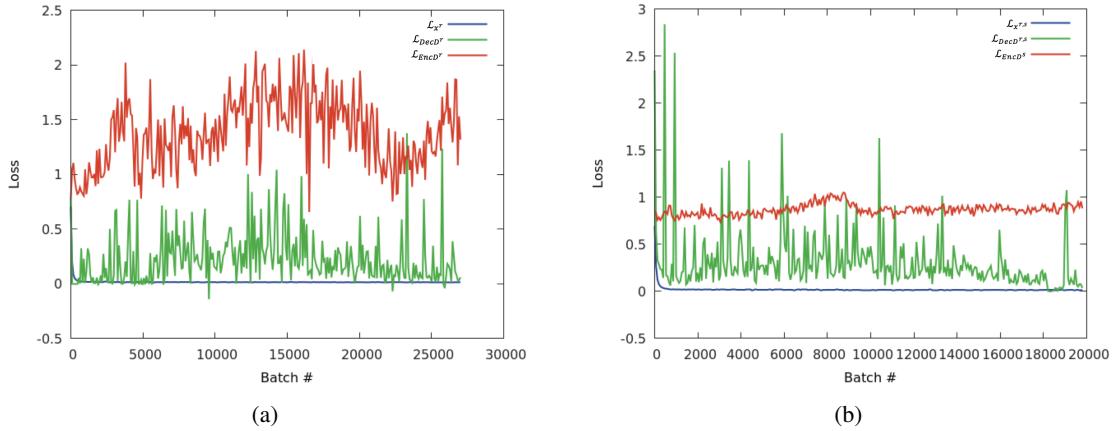


Figure S3: Training curves for the training of the reference model (a) and conditional model (b)

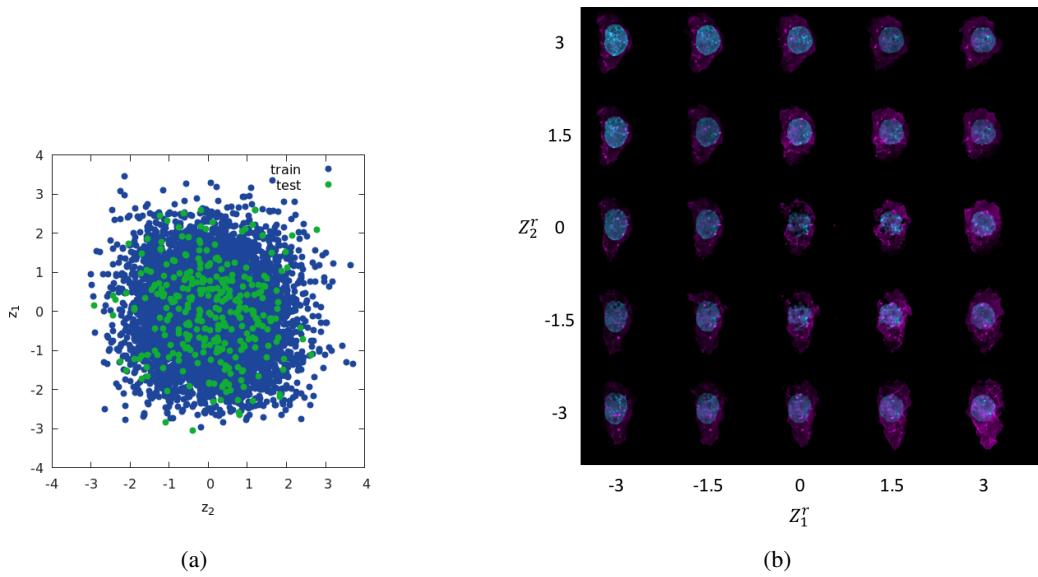


Figure S4: (a) shows the first two dimensions of the reference structure latent space Z^r . (b) shows the first two dimensions of the latent space sampled at -3, -1.5, 0, 1.5 and 3 standard deviations in Z_1^r (horizontal) and Z_2^r (vertical). Images have been cropped for visualization purposes.

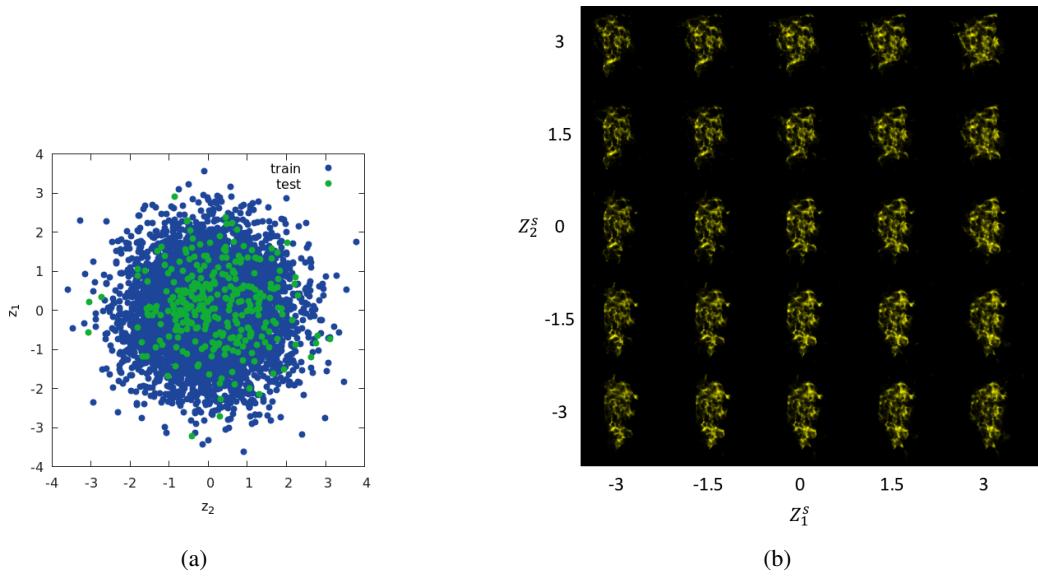


Figure S5: (a) shows the first two dimensions of the reference structure latent space Z^s . (b) shows the first two dimensions of the TOM20 latent space sampled at -3, -1.5, 0, 1.5 and 3 standard deviations in Z_1^s (horizontal) and Z_2^s (vertical). Images have been cropped for visualization purposes.

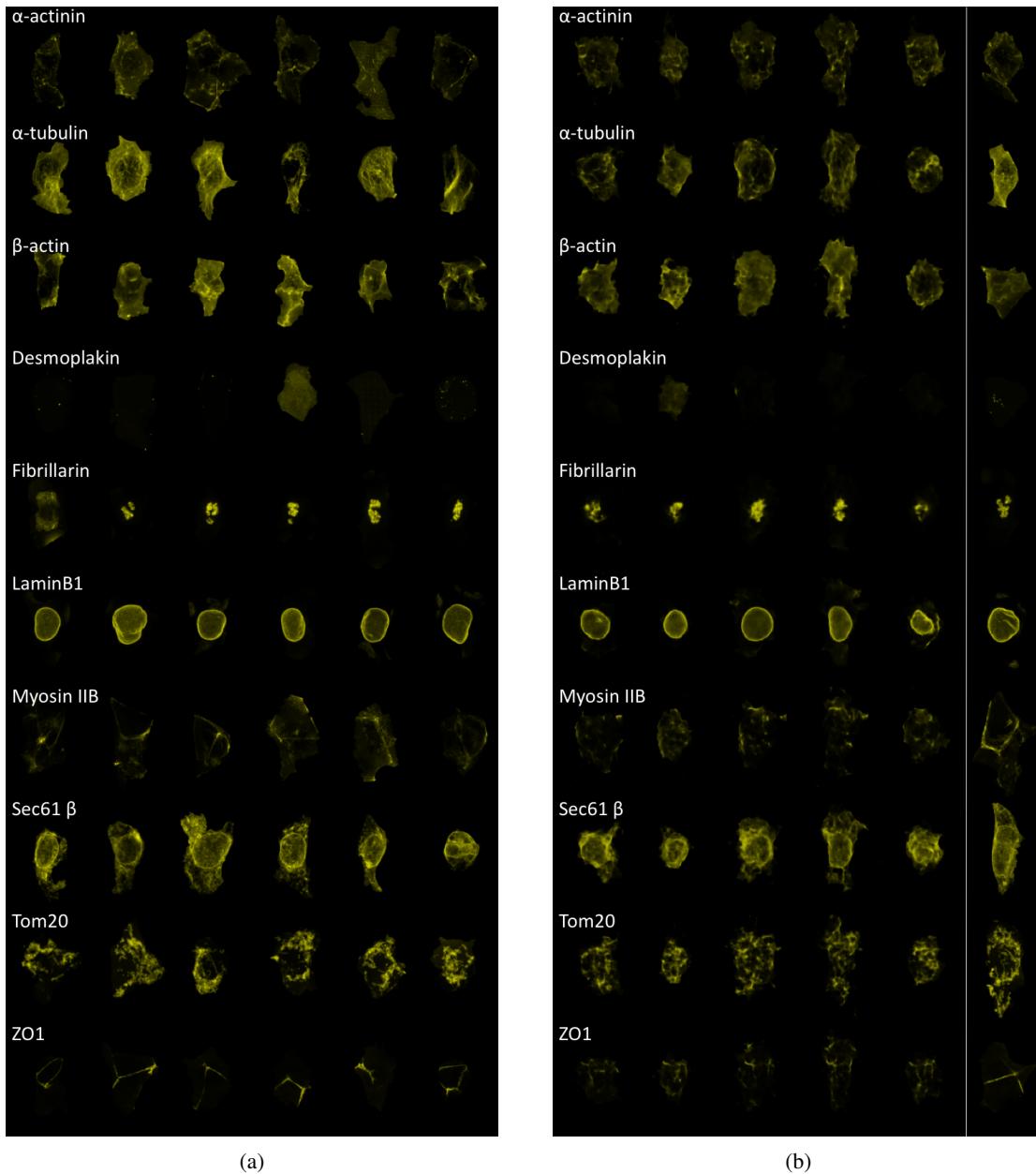


Figure S6: (a) Example structure channels for each of the 10 labeled structures in this paper and (b) predicted most probable localization patterns for selected cells from each labeled pattern. The first 5 columns show the maximum likelihood localization for the corresponding structures given the same cell and nuclear shape. The last column shows an observed cell with that labeled structure. Rows correspond to structure types. Images have been cropped for visualization purposes.

B. Model Architectures

4×4 64 CONV ↓	BNORM	PReLU
4×4 128 CONV ↓	BNORM	PReLU
4×4 256 CONV ↓	BNORM	PReLU
4×4 512 CONV ↓	BNORM	PReLU
4×4 1024 CONV ↓	BNORM	PReLU
4×4 1024 CONV ↓	BNORM	PReLU
$ Z^r $ FC	BNORM	PReLU

Table S1: Architecture of Enc^r

1024 FC	BNORM	PReLU
4×4 1024 CONV ↑	BNORM	PReLU
4×4 512 CONV ↑	BNORM	PReLU
4×4 256 CONV ↑	BNORM	PReLU
4×4 128 CONV ↑	BNORM	PReLU
4×4 64 CONV ↑	BNORM	PReLU
$4 \times 4 r $ CONV ↑	BNORM	SIGMOID

Table S2: Architecture of Dec^r

1024 FC		LEAKY RELU
1024 FC	BNORM	LEAKY RELU
512 FC	BNORM	LEAKY RELU
1 FC		SIGMOID

Table S3: Architecture of EncD^r and EncD^s

+ WHITE NOISE $\sigma = 0.05$		
4×4 64 CONV ↓	BNORM	LEAKYReLU
4×4 128 CONV ↓	BNORM	LEAKYReLU
4×4 256 CONV ↓	BNORM	LEAKYReLU
4×4 512 CONV ↓	BNORM	LEAKYReLU
4×4 512 CONV ↓	BNORM	LEAKYReLU
4×4 1 CONV ↓		SIGMOID

Table S4: Architecture of DecD^r

4×4 64 CONV ↓	BNORM	PReLU
4×4 128 CONV ↓	BNORM	PReLU
4×4 256 CONV ↓	BNORM	PReLU
4×4 512 CONV ↓	BNORM	PReLU
4×4 1024 CONV ↓	BNORM	PReLU
4×4 1024 CONV ↓	BNORM	PReLU
{K FC, $ Z^r $ FC, $ Z^s $ FC}	{BNORM, BNORM, BNORM}	{SOFTMAX, , }

Table S5: Architecture of $\text{Enc}^{r,s}$

1024 FC	BNORM	PReLU
4×4 1024 CONV ↑	BNORM	PReLU
4×4 512 CONV ↑	BNORM	PReLU
4×4 256 CONV ↑	BNORM	PReLU
4×4 128 CONV ↑	BNORM	PReLU
4×4 64 CONV ↑	BNORM	PReLU
$4 \times 4 r+s $ CONV ↑	BNORM	SIGMOID

Table S6: Architecture of $\text{Dec}^{r,s}$

+ WHITE NOISE $\sigma = 0.05$		
4×4 64 CONV ↓	BNORM	LEAKYReLU
4×4 128 CONV ↓	BNORM	LEAKYReLU
4×4 256 CONV ↓	BNORM	LEAKYReLU
4×4 512 CONV ↓	BNORM	LEAKYReLU
4×4 512 CONV ↓	BNORM	LEAKYReLU
4×4 K+1 CONV ↓		SIGMOID

Table S7: Architecture of $\text{DecD}^{r,s}$

C. Data

Labeled Structure	#total	#train	#test
α -actinin	493	462	31
α -tubulin	1043	1002	41
β -actin	542	513	29
Desmoplakin	229	219	10
Fibrillarin	988	953	35
Lamin B1	785	739	46
Myosin IIB	157	149	8
Sec61 β	835	784	51
TOM20	771	723	48
ZO1	234	229	5

Table S8: Labeled structures and their train/test split

α -actinin	22	2	5	1	0	0	0	0	0	1
α -tubulin	0	36	3	0	0	0	0	1	1	0
β -actin	3	7	19	0	0	0	0	0	0	0
Desmoplakin	1	0	1	7	0	0	0	0	0	1
Fibrillarin	0	0	0	0	35	0	0	0	0	0
Lamin B1	0	0	0	0	0	46	0	0	0	0
Myosin IIB	2	0	0	1	0	0	0	0	1	4
Sec61 β	0	1	0	0	0	0	0	50	0	0
TOM20	0	1	0	0	0	0	0	0	47	0
ZO1	1	0	0	1	0	0	2	0	0	1

Table S9: Labeled structure class prediction results on hold out