Bioinformatics, 2018, 1–9
doi: 10.1093/bioinformatics/bty923
Advance Access Publication Date: 6 November 2018
Original Paper

Bioimage informatics

Computational modeling of cellular structures using conditional deep generative networks

Hao Yuan[1], Lei Cai[1], Zhengyang Wang[2], Xia Hu[2], Shaoting Zhang[3] and Shuiwang Ji[2,*]

[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA, [2]Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA and [3]Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

*To whom correspondence should be addressed.

Associate Editor: Robert Murphy

Received on May 25, 2018; revised on September 14, 2018; editorial decision on October 14, 2018; accepted on November 5, 2018

Abstract

Motivation: Cellular function is closely related to the localizations of its sub-structures. It is, however, challenging to experimentally label all sub-cellular structures simultaneously in the same cell. This raises the need of building a computational model to learn the relationships among these sub-cellular structures and use reference structures to infer the localizations of other structures.

Results: We formulate such a task as a conditional image generation problem and propose to use conditional generative adversarial networks for tackling it. We employ an encoder–decoder network as the generator and propose to use skip connections between the encoder and decoder to provide spatial information to the decoder. To incorporate the conditional information in a variety of different ways, we develop three different types of skip connections, known as the self-gated connection, encoder-gated connection and label-gated connection. The proposed skip connections are built based on the conditional information using gating mechanisms. By learning a gating function, the network is able to control what information should be passed through the skip connections from the encoder to the decoder. Since the gate parameters are also learned automatically, we expect that only useful spatial information is transmitted to the decoder to help image generation. We perform both qualitative and quantitative evaluations to assess the effectiveness of our proposed approaches. Experimental results show that our cGAN-based approaches have the ability to generate the desired sub-cellular structures correctly. Our results also demonstrate that the proposed approaches outperform the existing approach based on adversarial auto-encoders, and the new skip connections lead to improved performance. In addition, the localizations of generated sub-cellular structures by our approaches are consistent with observations in biological experiments.

Availability and implementation: The source code and more results are available at https://github.com/divelab/cgan/.

Contact: sji@tamu.edu

1 Introduction

Sub-cellular structures are the structures localized within a cell, and typical examples are cell membrane, nucleus, cytoskeleton, chloroplast and different proteins. It is of great importance to understand the localizations of sub-cellular structures in a cell, since they determine the functions of cells. For example, proteins are the key components in the cell and carry out most of the cell functions (Lodish *et al.*, 1995). However, even the same protein may lead to different

cell functions when localized at different locations (Faust and Montenarh, 2000). Hence, it is especially important to study the localizations of different proteins in different cells. A popular approach is to build cell models, such as the location proteomics (Murphy, 2005), to learn the relationships between cell states, cell functions and sub-cellular structure locations. It is, however, difficult to experimentally distinguish and label all sub-cellular structures simultaneously in the same cell. It is hence important to build computational models to capture the localizations of different sub-cellular structures. Then such computational models can predict the locations of unlabeled sub-cellular structures and improve the predictions of cell models. Specifically, in this work, our models try to learn the relationships among cell membrane, nucleus and different types of proteins. If we let each channel of an image contains the information about one of the sub-cellular structures in the cell, the relationships between different structures can be represented by the relationships between image channels. In this way, we formulate the problem as an image generation task which generates the specific channel we need, given the other channels of an image.

Recently, several studies investigated the synthesis of sub-cellular structures (Ulman *et al.*, 2016). One straightforward way is to merge multiple neighboring point-like signals to obtain complicated structures (Baddeley *et al.*, 2010; Ulman *et al.*, 2016; Wu *et al.*, 2010). Another way is to employ image feature based approaches to model different cell components (Boland and Murphy, 2001; Carpenter *et al.*, 2006; Rajaram *et al.*, 2012). In addition, a few seminal studies have proposed generative models for such tasks and obtained promising results (Peng and Murphy, 2011; Zhao and Murphy, 2007). They built different parametric sub-models to learn different cell components, such as nuclear shape, cell shape, protein size and shape and organelle distributions. Their models also learn to capture relationships among different cell components. Such parametric approaches can model different types of sub-cellular structures and are expected to generate images following the same underlying distribution as the training set images. However, different sub-cellular structures are highly correlated in cells. The relationships among them are complicated and non-linear. Most of these approaches only employ traditional learning techniques, which may not fully capture highly complex relationships as compared with deep learning methods.

With the widespread use of computational methods, several studies have shown that generative models are particularly useful for such tasks (Peng and Murphy, 2011; Zhao and Murphy, 2007). By learning relationships among different sub-cellular structures, we can experimentally label a few structures in each cell and computationally predict the remaining ones. Recent work (Johnson *et al.*, 2017b) proposed an image generative model based on adversarial auto-encoders (AAEs) (Makhzani *et al.*, 2015). The AAE networks are derived from Variational Auto-encoders (VAEs) (Kingma and Welling, 2013) and follow an encoder–decoder structure. Instead of employing the Kullback–Leibler divergence as in VAEs, AAEs apply discriminators to encourage the latent variables to fit a normal distribution. As far as we know, this is the only deep learning method dealing with this sub-cellular structure generation problem. The method builds two different adversarial auto-encoder networks to learn the relationships between different sub-cellular structures. However, VAE-based methods tend to generate blurry images, especially for high resolution and intricate datasets (Dosovitskiy and Brox, 2016; Zhao *et al.*, 2017). In addition, the existing approach generates images from the vectorial latent representations of the input and hence tends to generate images with large variations during the testing phase. Furthermore, the spatial information is largely lost due to the down-sampling operations and up-sampling operations in its model. Then the generated structures may not localize accurately, which is demonstrated in our experiments.

It is known that GAN-based approaches usually generate more photorealistic images and the generated images tend to have smaller variations (Isola *et al.*, 2016; Kim *et al.*, 2017; Yi *et al.*, 2017; Zhu *et al.*, 2017). Hence, we propose approaches based on conditional GANs for this task. The conditional information consists of two parts; namely labeled reference structures (cell membrane and nucleus) and the protein type. We design the generator of our model as an encoder–decoder network (Badrinarayanan *et al.*, 2015; Long *et al.*, 2015) with skip connections (He *et al.*, 2016a, b) to share spatial information between the encoder and decoder. Furthermore, in such a conditional generation task, we believe that the conditional information should be incorporated to determine what should be shared between the encoder and decoder. Hence, we develop three different types of connections known as self-gated connection, encoder-gated connection and label-gated connection, respectively. All of the proposed skip connections are built based on the conditional information and employing the gating mechanisms (Dauphin *et al.*, 2016). There are two main contributions in our work; these are, we propose to adaptively apply conditional GANs for such tasks to generate more photorealistic images with smaller variations and we propose three different types of skip connections to incorporate the conditional information to control the information flow in the networks.

In order to show the effectiveness of our proposed approaches, we conduct several experiments and evaluate the results both qualitatively and quantitatively. Qualitative results show that our approaches are visually better than the existing method. The shape and location of the structures generated by our approach match the ground truths in the test set and are consistent with experimental observations in biological sciences. Furthermore, we evaluate different approaches quantitatively using the Parzen window log-likelihood estimation (Breuleux *et al.*, 2011). Results indicate that our methods outperform the existing method and the newly proposed connections can improve the performance of models.

2 Background and related work

In this section, we present a brief introduction of generative adversarial networks (GANs) in Subsection 2.1. Then we describe existing work on conditional generative adversarial networks in Subsection 2.2. We also discuss an existing approach for the cellular structure generation problem.

2.1 Generative adversarial networks

Estimating high-dimensional distributions is a challenging task as it requires a tremendous amount of training samples. Recent studies have shown the success of GANs in learning high-dimensional distributions implicitly, and they have been used to generate photorealistic images (Denton *et al.*, 2015; Radford *et al.*, 2015). GANs do not produce an explicit density function from the training data, but can be used to generate samples from the learned distributions. They consist of two distinct networks; namely, a generator G and a discriminator D. The generator G learns to capture the data distribution, and the discriminator D learns to discriminate samples generated from the generator G and those from the training data. These two networks are trained iteratively. Specifically, D is trained to distinguish real samples X from generated samples $\hat{X}$, while G is trained to fool the discriminator by trying to generate samples that

are similar to the real ones. The training process can be interpreted as a two-player minimax game.

Given a noise vector z sampled from a prior distribution $p_z(z)$, G generates a sample $\hat{X}$ using the learned mapping from $p_z(z)$ to the true data distribution. Meanwhile, D takes a sample as input and outputs a single scalar representing the probability that this sample belongs to the true data distribution. Mathematically, the objective function of GANs can be expressed as

$$\min_G \max_D \mathbb{E}_{X\sim p_{\text{data}}(X)}[\log D(X)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))], \quad (1)$$

where $p_{\text{data}}(X)$ represents the true data distribution.

2.2 Conditional generative adversarial networks

Although GANs are capable of learning high-dimensional distributions, they cannot be directly applied to many image-related applications, since they do not take conditional information into consideration. To this end, conditional GANs (cGANs) have been proposed by incorporating conditional information in image generation. In cGANs, many different types of conditional information can be used, including discrete class information (Mirza and Osindero, 2014), text information (Reed *et al.*, 2016; Zhang *et al.*, 2016) and image information (Denton *et al.*, 2016; Isola *et al.*, 2016; Zhu *et al.*, 2017).

Different from GANs, both the generator G and discriminator D in cGANs are coupled with the conditional information y. Given y and the sampled noise z, G generates an image $\hat{X}$. For the discriminator D, the input is a pair consisting of an image X and the conditional information y. Then D estimates the probability of such a pair being real. Note that in cGANs, a real pair not only requires that both the image and the conditional information are from the true data distribution, but also requires them to be consistent with each other. Formally, the objective function of cGANs can be written as

$$\min_G \max_D \mathbb{E}_{X,y\sim p_{\text{data}}(X,y)}[\log D(X,y)] + \mathbb{E}_{y\sim p_{\text{data}}(y),z\sim p_z(z)}[\log(1 - D(G(y,z),y))], \quad (2)$$

where $p_{\text{data}}(y)$ denotes the distribution of the conditional information.

It is worth noting that we can feed the paired input into cGANs in many ways. For example, if y is a one-hot vector representing class information, y and z can be concatenated and fed into G. In this case, the input pair of D contains an image and a vector. One can replicate the vector y multiple times spatially and perform concatenation with the image in the depth dimension before feeding it into D (Radford *et al.*, 2015). When y represents text information, it can be converted to a high-dimensional vector using embedding and then compressed to a lower dimension using a fully-connected layer. Then it can be used in a similar way as discussed above. If the conditional information y is an image, the input of D is a pair of images. It is straightforward to concatenate them and feed it into D. For the generator, one can apply a trainable network to extract vectorized representations of the input image and concatenate them to the noise z (Isola *et al.*, 2016).

2.3 Cellular structure modeling

In Johnson *et al.* (2017b), a deep learning model for cellular structure modeling is proposed. This method uses a conditional generative model based on adversarial auto-encoders (AAE). Note that AAE networks are derived from VAE networks, which also follow an encoder–decoder structure. Instead of employing the Kullback–Leibler divergence as in VAE, AAE applies a discriminator to encourage the latent variable to follow a normal distribution.

The model in Johnson *et al.* (2017b) consists of two different AAE networks. The first one takes information of cell membrane and nucleus as input and learns their shapes. It produces encoded latent representations of inputs and encourages the latent variables to follow a normal distribution. In addition, it generates a reconstructed image which is expected to be close to the input. The second AAE network learns the relationships between sub-cellular structures dependent on the encoded latent representations. The latent variables consists of three parts; namely encoded representations of the cell and nucleus, encoded representations of sub-cellular structure and the type information. It encourages the encoded representations of cell and nucleus to be similar to the latent encodings of the first network and the type information to be close to the vectorized representations of ground truth type. In addition, it encourages the encoded representations of sub-cellular structure to follow the normal distribution. After training, the decoder of the second AAE network is employed to generate the desired sub-cellular structure based on cell membrane, nucleus and the type of sub-cellular structure. The input consists of three parts: the latent encodings of cell membrane and nucleus from the first AAE network, the randomly sampled representations of sub-cellular structure and the vectorized representations of type information. In this model, discriminators are used to make the reconstructed images similar to the input images.

The approach in Johnson *et al.* (2017b) is based on AAE. It is known that GAN-based approaches usually outperform VAE-based methods in many image generation tasks. This is because VAE-based approaches tend to produce blurry images on complex datasets (Dosovitskiy and Brox, 2016; Zhao *et al.*, 2017). Since AAEs are derived from VAEs, they also inherit the blurry image generation problem. In addition, in their model, only the decoder of the second AAE network is employed to generate the desired sub-cellular structure based on three latent vectors. It tends to generate images with large variations because there are fewer constraints in these vectors. Furthermore, the spatial information is largely lost in its information flow while this information is important to determine the shape and location of the desired sub-cellular structures. Without explicit information shared between encoders and decoders, the generated structures may not localize correctly. Hence, in this work, we propose to employ conditional GANs for such tasks and design our networks to overcome these limitations.

3 Conditional generative models for cellular structure modeling

In this section, we describe the cellular structure generation problem in Subsection 3.1. Then the general framework of our model is presented in Subsection 3.2. After that, we introduce our design of generators and three types of proposed models in Subsection 3.3. Finally, the architecture of discriminator networks is discussed in Subsection 3.4.

3.1 The cellular structure generation problem

Understanding cellular organization and sub-cellular structure localization is of significant importance, since they are highly related to cell functions. Due to the diversity of different molecular complexes, it is challenging to experimentally label all structures in the same cell simultaneously and determine the cellular organization. Hence, it is important to apply a computational approach to learn the underlying relationship and representations of those structures. We formulate this problem as an image generation task in which we use images channels to represent different sub-cellular structures.

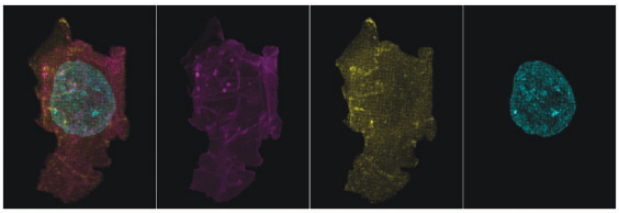

Fig. 1. Illustration of channels in a cellular image containing the alpha actinin protein. The leftmost image contains all channels, and the following ones represent cell membrane (magenta), alpha actinin structure (yellow) and nucleus (cyan), respectively. The images have been colored and cropped for visualization purpose

Formally, let X denotes a cellular image with three channels. The first channel contains information of cell membrane; the second channel contains the distribution of one sub-cellular structure, such as one type of proteins and the third channel is the nucleus, as shown in Figure 1. The cell membrane channel and the nucleus channel together serve as the reference channels, denoted as X^r, as they are available in all images. The sub-cellular structure channel serves as the structure channel, denoted as X^s (Johnson *et al.*, 2017b). Then each cellular image X consists of the reference channels X^r and the structure channel X^s; that is, $X = X^{r,s} = [X^r, X^s]$. We use y to represent the type of the sub-cellular structure in X^s. Given the reference channels X^r of a cellular image and any desired sub-cellular structure type y, we aim at generating the corresponding structure channel $X^{\hat{s}}$, which is expected to be similar to the true X^s. Some examples are presented in Figure 2. By using the same X^r and different y as inputs, we can obtain the localizations of different sub-cellular structures in the same cell.

3.2 Problem formulation

The problem mentioned above can be considered as a conditional generation task, where the conditional information consists of the reference channels X^r and the type information y. We propose to employ conditional GANs for such tasks. Specifically, we build an encoder–decoder network with skip connections as the generator to incorporate the conditional information. It follows the general structure of the 'U-Net' network (Ronneberger *et al.*, 2015), but is coupled with different ways of connections instead of applying the original skip connections. The generator of our model consists of two parts; namely an encoder and a decoder. It incorporates the conditional information X^r and y with the sampled noise z and outputs the generated structure channel, denoted as $X^{\hat{s}}$. The input to our discriminator is a tuple, which contains three parts: the reference channels X^r, the type information y and the structure channel X^s (or $X^{\hat{s}}$). It is noteworthy that we integrate the structure channel with the reference channels and they together become one image $X^{r,s}$ (or $X^{r,\hat{s}}$). Finally, the discriminator estimates the probability of such a tuple being real. The structure of our model is shown in Figure 3. The generator tries to generate structure channels similar to the real ones and fool the discriminator. The discriminator is trained to distinguish if the structure channels come from the true distribution. Mathematically, the objective function of cGANs can be expressed as

$$\begin{aligned}&\min_G \max_D \mathbb{E}_{X^{r,s},y\sim p_{\text{data}}(X^{r,s},y)}[\log D(X^{r,s},y)]\\&+\mathbb{E}_{X^r,y\sim p_{\text{data}}(X^r,y),z\sim p_z(z)}[\log(1-D([X^r,G(X^r,y,z)],y))],\end{aligned} \tag{3}$$

where $[X^r, G(X^r, y, z)] = [X^r, X^{\hat{s}}] = X^{r,\hat{s}}$, and the discriminator D tries to maximize this objective while the generator G tries to minimize it.

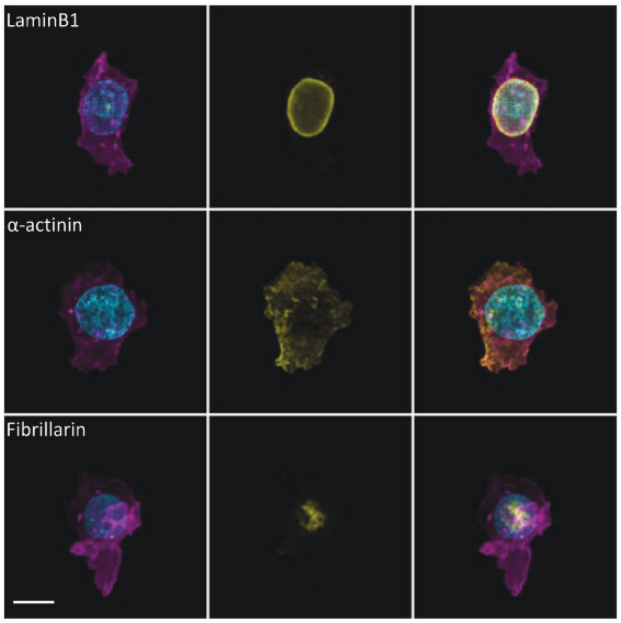

Fig. 2. Examples of the cellular structure generation problem. The first column is the input X^r and shows the type of protein we try to generate. The second column represents the generated protein $X^{\hat{s}}$. The last column is the image integrating X^r and $X^{\hat{s}}$ together. The scale bar at the bottom left represents 10 μm, and the examples in this figure share the same scale ratio

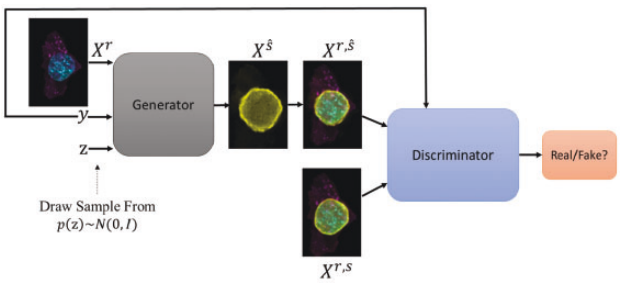

Fig. 3. The general structure of our proposed model

Recently, a few studies have shown that it is beneficial to train GANs with another auxiliary loss, such as the $L1$ or $L2$ distance loss (Isola *et al.*, 2016; Ledig *et al.*, 2016; Pathak *et al.*, 2016). In our model, we use the $L2$ loss to help the training of the generator. This means that the generator not only tries to fool the discriminator, but also generates images that are close to the ground truths in an $L2$ sense. Formally, the objective function associate with $L2$ loss can be written as

$$\min_G \mathbb{E}_{X^r,X^s,y\sim p_{\text{data}}(X^r,X^s,y),z\sim p_z(z)}[||X^s - G(X^r,y,z)||_2]. \tag{4}$$

Note that the generator tries to minimize this objective while the discriminator is not related. We combine the cGANs objective and the $L2$ objective to train the model. The generator and discriminator are trained iteratively; that is, we train the discriminator one step to maximize the cGANs objective and then train the generator one step to minimize both the cGANs objective and $L2$ objective. These steps are repeated.

3.3 The proposed generator networks

Different from traditional conditional image generation tasks, the conditional information of sub-cellular structure generation tasks

consists of two parts; those are, the reference channels X^r and the type information y. In our work, X^r is represented as an image, and y is a one-hot vector. We need to combine these two parts together with the sampled noise z to generate the structure channel. As discussed in Subsection 2.2, to combine the information of an image and a vector, we can map the image to a vector representation through a trainable network and concatenate it to the vector. Hence, we employ an encoder–decoder network for our generator.

The encoder consists of several convolutional layers (LeCun *et al.*, 1998) with stride equal to two and a final fully-connected layer. It takes the reference channels X^r as inputs and extracts a vectorized representation of X^r, denoted as z^r. Then z^r is combined with the one-hot vector y and a sampled noise vector z via concatenation. After that, the combined information is fed into the decoder network, which contains a fully-connected layer followed by several de-convolutional layers of stride two (Gao *et al.*, 2017). Finally, the decoder outputs the generated structure channel, denoted as $X^{\hat{s}}$. Note that the number of de-convolutional layers in the decoder should be equal to the number of convolutional layers in the encoder.

In addition, the localizations of sub-cellular structures in a cell are highly related to the shape and location of its membrane and nucleus. For example, the protein LaminB1 always surrounds the DNA, which means it always localizes to the inner boundaries of nucleus. Such spatial information is useful for the sub-cellular structure generation, but it cannot be perfectly conveyed because of the down-sampling and up-sampling operations in our encoder–decoder generator. Adding skip connections between the encoder and decoder is shown to be beneficial in many tasks where global spatial information is of great importance, such as in image segmentation task (Ronneberger *et al.*, 2015). Hence, we use skip connections in our model and the structure of our generator network is shown in Figure 4.

In traditional 'U-Net', the skip connections use concatenation, which means the information in the encoder is simply copied and concatenated to the information of decoder. However, in such a conditional generation task, we believe it is beneficial to only share useful information between the encoder and decoder, and what is useful depends on the conditional information. For example, the protein LaminB1 always surrounds the DNA so that the shape and location of DNA are more important to the generation of LaminB1. In another case, the protein Alpha-actinin always localizes to the inner cell membranes and hence the localization of membrane is more useful for the generation of Alpha-actinin. Therefore, it is desirable to incorporate the conditional information, especially the label information y, when building skip connections between the encoder and decoder.

We propose to apply gating mechanisms to build the skip connections. The skip connections can be considered as paths through which information flows from the encoder to the decoder. The gating mechanisms can control such information flow and have shown their benefits in many tasks (Chung *et al.*, 2014; Dauphin *et al.*, 2016). By learning a gating function, the networks are allowed to determine what information should be passed through the skip connections between the encoder and decoder. In this work, we propose several different gated connections to guide the flow of conditional information.

Self-Gated Connections: First of all, instead of simply copying the information of encoder and concatenate it to the decoder, we propose to take the conditional information X^r into account when building the skip connections. The gating function learns to propagate only a fraction of information from the encoder to decoder. As shown in Figure 5, X_1 is the information of one layer in the encoder that contains the low-level features of input X^r, while X_2 refers to information of the corresponding layer in the decoder. Mathematically, the self-gated connection can be represented as

$$g = \sigma(c(X_1)), \quad o = [X_1 \otimes g, X_2], \tag{5}$$

where $c(\cdot)$ represents convolution, $\sigma(\cdot)$ denotes sigmoid function, $\otimes$ refers to element-wise multiplication, $[\cdot,\cdot]$ denotes concatenation and o represents the output.

First, X_1 passes through a convolutional layer with a sigmoid activation function to obtain a weight matrix. This weight matrix has the same spatial dimensions as those of X_1, and the value of each element is between 0 and 1. We perform element-wise multiplication between X_1 and the weight matrix, and then concatenate it with X_2 to obtain the output. In this way, only a fraction of X_1 is shared with X_2, and the weight matrix determines what to be shared. We term it self-gated connection because the weight matrix is calculated from X_1 and is used to multiply by X_1 itself.

Encoder-Gated Connections: We also propose another type of gated connection, termed encoder-gated connection. Instead of using gating functions to control the information flow from the encoder to the decoder, we propose to use the information in encoder to guide what information should be propagated through the decoder layers. The operations of the proposed encoder-gated connection are illustrated in Figure 6, where X_1 and X_2 have been defined above. Formally, the encoder-gated connection can be represented as

$$g = \sigma(c(X_1)), \quad o = X_2 \otimes g. \tag{6}$$

The way to compute the weight matrix is similar to the case of self-gated connection in that X_1 is fed to a convolutional layer with sigmoid activation function and produces a weight matrix. Note that the weight matrix has the same spatial dimensions as those of

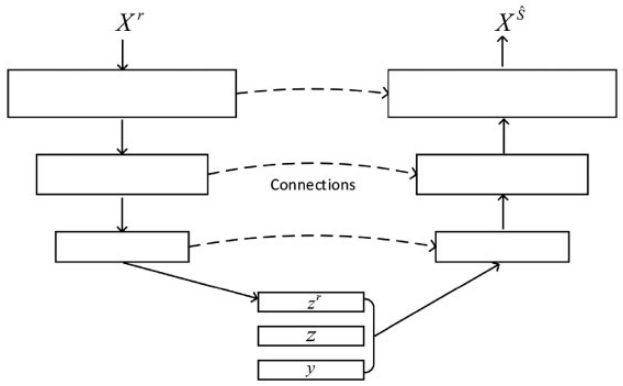

Fig. 4. The structure of the generator in our model. The operation among z^r, z and y is concatenation

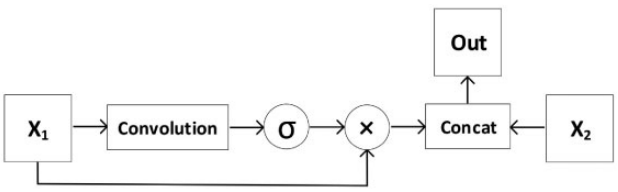

Fig. 5. Illustration of the self-gated connection, where σ denotes the sigmoid activation function and 'Concat' denotes concatenation

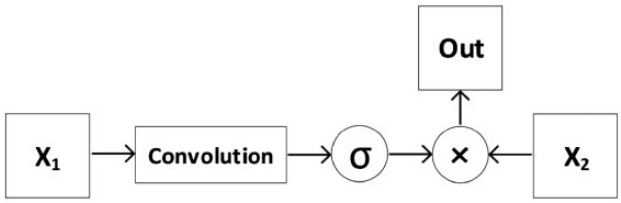

Fig. 6. Illustration of the encoder-gated connection, where σ denotes the sigmoid activation function

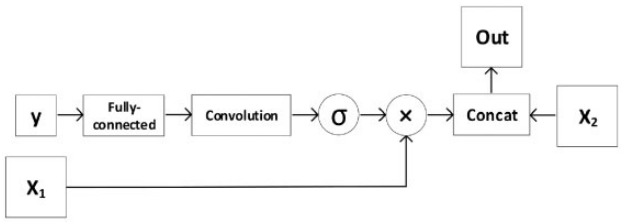

Fig. 7. Illustration of the label-gated connection, where σ denotes the sigmoid activation function and Concat denotes simple concatenation

X_2, not X_1. Instead of multiplying the weight matrix with X_1 itself, we perform element-wise multiplication between the weight matrix and X_2. The weight matrix can select useful information in X_2 to be preserved as output. Since the output of current layer is also the input of the next layer, the weight matrix controls the information flow between different decoder layers. The main difference between self-gated connection and encoder-gated connection is that the former controls information sharing between the encoder and decoder while the latter determines the information flow between decoder layers.

Label-Gated Connections: Both of the above connections are built based on the conditional information X^r. As can be seen from the examples mentioned above, the type information y also contributes to the localizations of generated structures. Hence, we propose another type of gated connection, termed label-gated connection, to incorporate the type conditional information between the encoder and decoder. Different from the self-gated connections and encoder-gated connections, the weight matrix here is generated from the type information y. The operations of label-gated connections are shown in Figure 7. The type information y first passes through a fully-connected layer with no activation function, which maps the one-hot vector y to a high dimensional space. Then it is fed into a convolutional layer with a sigmoid function to obtain a weight matrix, whose spatial dimensions are the same as those of X_1. After that, X_1 is multiplied by the weight matrix, and we concatenate the result with X_2 to produce the output. In this way, both the conditional type information y and reference channels X^r are incorporated to control the information flow from the encoder to the decoder. The mathematic formulation of label-gated connection can be expressed as

$$g = \sigma(c(f(y))), \quad o = [X_1 \otimes g, X_2], \tag{7}$$

where $f(\cdot)$ denotes the fully-connected layer.

3.4 Discriminator networks

The discriminator of our model consists of several convolutional layers and fully-connected layers. It takes a tuple as input, which consists of the reference channels X^r, the structure channel X^s (or $X^{\hat{s}}$) and the type information y. As shown in Figure 3, the reference channels and structure channel can be integrated together as one image ($X^{r,s}$ or $X^{r,\hat{s}}$). Then the input becomes a pair consisting of an image and a one-hot vector. In order to combine these two parts together, we choose to replicate y multiple times spatially and perform a depth concatenation with the image before feeding it into the discriminator. It is noteworthy that such operations will be performed twice, in the different layers of the discriminator. Finally, the discriminator outputs a single value, which estimates the probability of the input tuple being real. Note that the input tuple is real if all components are from true data distribution and they are consistent with each other. In addition, we add noise to the input of discriminator, since it is shown to be useful to improve the stability of GAN-based models (Arjovsky and Bottou, 2017).

4 Experimental studies

4.1 Dataset and experimental setup

We use the 2D cellular image dataset released by Allen Institute for Cell Science. The data are obtained from a 3D confocal microscopy dataset by maximum intensity projection (Johnson *et al.*, 2017b). There are 6077 cellular images in total and each image contains channels representing the cell membrane, nucleus and a labeled sub-cellular structure (protein). There are 10 different types of sub-cellular structures in this dataset, including α-actinin, α-tubulin, β-actin, desmoplakin, fibrillarin, lamin B1, myosin IIB, Sec61β, TOM20 and ZO1. We randomly split the dataset into training set (5000 images) and testing set (1077 images). Each image is scaled to 256×256 pixels by bilinear interpolation, and the resolution is 0.317 μm/pixel (Johnson *et al.*, 2017b).

The encoder part of our generator consists of six convolutional layers followed by a fully-connected layer. The stride is set to 2, and the kernel size is set to 4×4 in convolutional layers. The numbers of output channels are doubled in each layer, starting from 64. For all layers in the encoder, batch normalization (Ioffe and Szegedy, 2015) is applied and parametric rectified linear unit (PReLU) (He *et al.*, 2015) is employed as activation functions. The dimension of latent variables z^r is set to 16. We set the dimension of the sampled noise z to 16 as well and sample it from the normal distribution $N(0, I)$. There are 10 types of sub-cellular structures in total, so the dimension of y is 10.

In the decoder of our model, there is a fully-connected layer followed by six deconvolutional layers with a stride equal to 2. The kernel size of de-convolutional layers is set to 4×4. The number of output channels for each de-convolutional layer is the same as its corresponding layer in the encoder. We choose the PReLU as the activation function and apply batch normalization for all layers. In addition, different skip connections between the encoder and decoder are also applied in different models.

The discriminator consists of four convolutional layers followed by two fully-connected layers. The stride and kernel size are the same as above. The output channels of convolutional layers are doubled in each layer, starting from 32. We choose to apply leaky rectified linear unit (LReLU) as the activation function for convolutional layers and the sigmoid function for fully-connected layers. As mentioned in Section 3.4, the one-hot vector y is replicated spatially and concatenated with the image. Such operations are performed twice; that is, on the input of the first and the third convolutional layers. In addition, the noise we add to the input of discriminator is sampled from a normal distribution with mean equal to 0 and SD equal to 0.01.

We implement our methods using TensorFlow and conduct our experiments on one Tesla K80 GPU. The learning rate is $2 \times e^{-4}$ and the batch size is 10. We follow the standard procedure to train cGANs by performing one gradient descent step on discriminator first, then one step on generator and repeat. We apply the Adam optimizer (Kingma and Ba, 2014) with momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4.2 Qualitative results

We conduct experiments to compare the performance of different gated connections in the conditional GAN models; namely the self-gated connection, encoder-gated connection, label-gated connection and the original skip connection. Note that, we term the original skip connection in 'U-Net' as copy connection to avoid confusion. In addition, we compare our GAN-based approaches with the AAE-based approach mentioned in Section 2.3. To the best of our knowledge, this is the only existing deep learning method dealing with such structure generation problem. We use the source code released by the original authors to produce their results.

Given the reference channels X^r and the sub-cellular structure type information y of an image from the test set, we compare the generated structure channels of different models. The results are shown in Figure 8, where all samples are randomly selected. We conduct experiments for all 10 types of sub-cellular structures, and each row corresponds to one type. In each row, the leftmost column is the reference channels X^r of a cell and the second one is the observed sub-cellular structure (ground truth), which shows the true shape, location and density distribution. The following columns represent the generated sub-cellular structures of different models. We can observe that the localizations of sub-cellular structures generated by cGANs models can match the ground truths precisely, which means cGANs models can learn the underlying relationships between X^r, y and X^s well. However, the shape and location of images generated by AAE-based approaches are different from the ground truths. Furthermore, the existing AAE-based approach tends to generate blurrier images than ours. On the other hand, the sub-cellular structures generated by cGANs approaches also match the properties observed in biological experiments. For example, the localization of protein α-tubulin is consistent with the shape and position of the membrane; protein fibrillarin always localizes within the nucleus, and protein LaminB1 always localizes to the inner boundaries of the nucleus.

For the proteins Desmoplakin and ZO1, our cGANs models can learn the shape and location correctly but not for the density distribution. This is because these two types of proteins have much fewer training examples than the others. The AAE-based approach learns better density distributions for these two types of proteins, but the localizations do not match the ground truths. Overall, our cGAN-based approaches outperform the AAE-based method in term of visual comparison. It is difficult to visually compare the results of different cGANs models since the generated structures are very similar in term of localization and only vary slightly for the density distribution. Hence, we present quantitative evaluation in the next subsection. More experimental results are released online.

4.3 Quantitative results

We perform quantitative analysis using the Parzen window log-likelihood estimation (Breuleux *et al.*, 2011). The underlying idea of Parzen window log-likelihood estimation is to estimate the probability of the test set data under the probability distribution of generated samples. It fits a Gaussian Parzen window to the generated samples and estimates the log-likelihood (Goodfellow *et al.*, 2014). This approach is widely used in many generative models where the exact likelihood is not tractable (Goodfellow *et al.*, 2014; Makhzani *et al.*, 2015).

Fig. 8. Qualitative comparison between different approaches. Different rows show the results for different types of sub-cellular structure. In each row, the leftmost image is the input X^r and the second one is the ground truth. The following ones are generated structures using different approaches in the following order: cGANs with self-gated connection, cGANs with encoder-gated connection, cGANs with label-gated connection, cGANs with copy connection and the existing AAE-based approach. The scale bar at the bottom left represents 10 μm, and the examples in this figure share the same scale ratio

Table 1. Parzen window log-likelihood estimates on the whole test dataset

Model	Log-Likelihood
cGANs with self-gated connection	88 700 ± 42
cGANs with encoder-gated connection	88 721 ± 42
cGANs with label-gated connection	**88 791 ± 40**
cGANs with copy connection	88 568 ± 39
AAE-based approach	87 689 ± 55

The bold characters indicate the best evaluation scores.

We first perform this quantitative evaluation on the whole test dataset, regardless of the type information. The evaluation results are reported in Table 1. Among the five approaches, cGANs with label-gated connection have the best quantitative results. The three cGAN methods with our proposed skip connections share very similar results, and their results are all better than the one with copy connection. Furthermore, all cGANs approaches perform better than

the AAE-based approach in term of Parzen window log-likelihood estimation.

We also perform quantitative evaluations for different sub-cellular structure types, as shown in Table 2. We can observe from the results that the cGANs with self-gated connection have the best performance for six types of proteins. In addition, cGANs with encoder-gated connection outperform other methods for proteins β-actin and TOM20. The AAE-based approach performs better for Desmoplakin and Fibrillarin. Generally speaking, the results of cGANs approaches are better than the existing AAE-based approach. Together with the qualitative results, we can conclude that our proposed cGANs approaches perform better than the existing approach both qualitatively and quantitatively. In addition, the three proposed skip connections are shown to be useful for such conditional generation task.

4.4 Integrating structures

As mentioned in Subsection 3.1, the challenging task in biological experiments is to experimentally label all structures simultaneously. We propose to apply a computational approach to build a model for such a task so that it can generate the localizations of different structures. In order to show the effectiveness of our methods, we conduct experiments using the same input X^r but different input y. It indicates how different sub-cellular structures localize in the same cell and how different sub-cellular structures are related to each other. We report the results obtained from cGANs with label-gated connection in Figure 9. In this experiment, we randomly select three images from test set and use their reference channels X^r as input. In the results, we show the generated structures for proteins α-actinin, Fibrillarin, LaminB1 and Tom20. The localizations of generated sub-cellular structures are consistent with the properties observed in biological experiments. It also matches our observations in Subsection 4.2. In this way, we can obtain the localization of any needed sub-cellular structure in a cell based on other structures and the learned relationships.

5 Conclusions and discussions

The localizations of sub-cellular structures in a cell are important because knowing such localizations is helpful to determine the functions of the cell. However, it is difficult to experimentally observe all structures of the same cell. We formulate such problem as a conditional image generation problem and apply conditional GANs to learn the relationships among different sub-cellular structures. We design the generators of our model as an encoder–decoder network with skip connections to incorporate the conditional information. Furthermore, instead of applying the original copy connection in 'U-Net', we propose to incorporate the conditional information to build the skip connections. Three different types of skip connections are proposed, including self-gated connection, encoder-gated connection and label-gated connection.

We conduct experiments to compare the performance of our proposed models with an existing AAE-based approach. Qualitative results show that the structures generated by our approaches match the ground truths precisely, in both shape and location. In addition, the localizations of different sub-cellular structures are consistent with biological observations. The results of cGANs methods are visually better than those of the existing approach. Furthermore, quantitative evaluations are performed using the Parzen window log-likelihood estimation. It is shown that our approaches outperform the AAE-based approach and the proposed skip connections can improve the performance. Overall, our proposed approaches have demonstrated the ability to learn the underlying relationships between different structures.

It is noteworthy that our model can be extended to learn relationships among different proteins. For examples, the reference channels may contain the information of two types of proteins, and the structure channel refers to another type of protein. In this way, the models can learn the relationships among different types of proteins, without knowing the information of nucleus and cell membrane. In addition, the AAE-based approach was extended to deal with 3D images recently (Johnson *et al.*, 2017a). Our model can also be extended to handle 3D data by designing 3D networks. For convolutional and de-convolutional layers, we can employ 3D convolutional and deconvolutional layers. The 3D version of

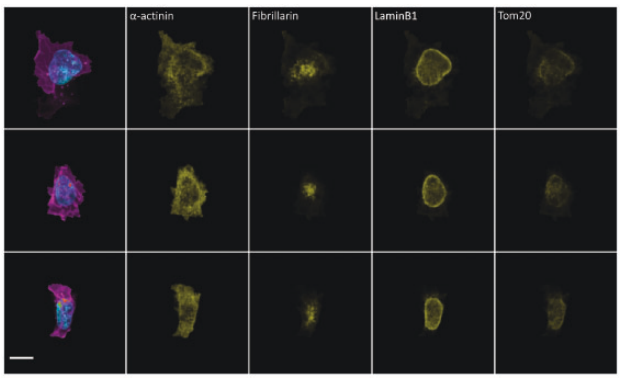

Fig. 9. Examples showing localizations of different proteins in the same cell. Results of different cells are shown in different rows. In each row, the leftmost image is the input X^r, and the following ones are different types of generated sub-cellular structures. The scale bar at the bottom left represents 10 μm, and the examples in this figure share the same scale ratio

Table 2. Parzen window log-likelihood estimates for different types of proteins

Structure type	Self-gated	Encoder-gated	Label-gated	Copy connection	AAE-based
α−actinin	**86 701 ± 191**	86 498 ± 208	86 565 ± 204	86 696 ± 193	85 761 ± 220
α−tubulin	**87 467 ± 108**	87 466 ± 108	87 096 ± 95	87 103 ± 99	86 578 ± 98
β−actin	86 985 ± 108	**87 024 ± 111**	87 020 ± 105	86 854 ± 101	86 598 ± 114
Desmoplakin	85 633 ± 221	85 964 ± 255	85 737 ± 272	86 249 ± 231	**86 440 ± 233**
Fibrillarin	85 041 ± 334	85 067 ± 319	85 133 ± 325	85 369 ± 308	**86 004 ± 280**
LaminB1	**86 460 ± 182**	86 408 ± 186	86 423 ± 187	86 337 ± 184	85 820 ± 187
Myosin IIB	**85 883 ± 236**	85 256 ± 288	85 743 ± 257	85 676 ± 264	84 854 ± 283
Sec61 β	**86 483 ± 104**	86 401 ± 110	86 283 ± 114	86 301 ± 112	85 515 ± 118
Tom20	86 832 ± 162	**86 939 ± 160**	86 754 ± 156	86 923 ± 149	85 905 ± 174
ZO1	**85 857 ± 316**	84 905 ± 315	85 062 ± 304	85 370 ± 292	85 068 ± 297

The bold characters indicate the best evaluation scores.

fully-connected layers contains a large number of parameters and may lead to memory issues. We can replace the fully-connected layers by 3D convolutional layers to avoid this issue.

Funding

This work was supported by National Science Foundation grants [DBI-1661289, IIS-1615035, IIS-1633359, IIS-1718840, ABI-1661280, CNS-1629913] and Defense Advanced Research Projects Agency grant [N66001-17-2-4031].

Conflict of Interest: none declared.

References

Arjovsky,M. and Bottou,L. (2017) Towards principled methods for training generative adversarial networks. *arXiv Preprint arXiv: 1701.04862*.

Baddeley,D. *et al*. (2010) Model based precision structural measurements on barely resolved objects. *J. Microscopy*, **237**, 70–78.

Badrinarayanan,V. *et al*. (2015) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *arXiv Preprint arXiv: 1511.00561*.

Boland,M.V. and Murphy,R.F. (2001) A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela cells. *Bioinformatics*, **17**, 1213–1223.

Breuleux,O. *et al*. (2011) Quickly generating representative samples from an rbm-derived process. *Neural Comput*., **23**, 2058–2073.

Carpenter,A.E. *et al*. (2006) Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol*., **7**, R100.

Chung,J. *et al*. (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv Preprint arXiv: 1412.3555*.

Dauphin,Y.N. *et al*. (2016) Language modeling with gated convolutional networks. *arXiv Preprint arXiv: 1612.08083*.

Denton,E. *et al*. (2016) Semi-supervised learning with context-conditional generative adversarial networks. *arXiv Preprint arXiv: 1611.06430*.

Denton,E.L. *et al*. (2015) Deep generative image models using a laplacian pyramid of adversarial networks. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Barcelona, pp. 1486–1494.

Dosovitskiy,A. and Brox,T. (2016) Generating images with perceptual similarity metrics based on deep networks. *In: Advances in Neural Information Processing Systems*. Curran Associates, Inc., Montreal, pp. 658–666.

Faust,M. and Montenarh,M. (2000) Subcellular localization of protein kinase ck2. *Cell Tissue Res*., **301**, 329–340.

Gao,H. *et al*. (2017) Pixel deconvolutional networks. *arXiv Preprint arXiv: 1705.06820*.

Goodfellow,I. *et al*. (2014) Generative adversarial nets. *In: Advances in Neural Information Processing Systems*. Curran Associates, Inc., Montreal, pp. 2672–2680.

He,K. *et al*. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, Santiago, pp. 1026–1034.

He,K. *et al*. (2016a) Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, pp. 770–778.

He,K. *et al*. (2016b) Identity mappings in deep residual networks. In: *European Conference on Computer Vision*. Springer, Amsterdam, The Netherlands, pp. 630–645.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*. JMLR.org, Lille, pp. 448–456.

Isola,P. *et al*. (2016) Image-to-image translation with conditional adversarial networks. *arXiv Preprint arXiv: 1611.07004*.

Johnson,G.R. *et al*. (2017a) Building a 3d integrated cell. *bioRxiv*.

Johnson,G.R. *et al*. (2017b) Generative modeling with conditional autoencoders: building an integrated cell. *arXiv Preprint arXiv: 1705.00092*.

Kim,T. *et al*. (2017) Learning to discover cross-domain relations with generative adversarial networks. *arXiv Preprint arXiv: 1703.05192*.

Kingma,D. and Ba,J. (2014) Adam: a method for stochastic optimization. *arXiv Preprint arXiv: 1412.6980*.

Kingma,D.P. and Welling,M. (2013) Auto-encoding variational bayes. *arXiv Preprint arXiv: 1312.6114*.

LeCun,Y. *et al*. (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.

Ledig,C. *et al*. (2016) Photo-realistic single image super-resolution using a generative adversarial network. *arXiv Preprint arXiv: 1609.04802*.

Lodish,H. *et al*. (1995) *Molecular Cell Biology*, Vol. **3**. Scientific American Books, New York.

Long,J. *et al*. (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Boston, MA, pp. 3431–3440.

Makhzani,A. *et al*. (2015) Adversarial autoencoders. *arXiv Preprint arXiv: 1511.05644*.

Mirza,M. and Osindero,S. (2014) Conditional generative adversarial nets. *arXiv Preprint arXiv: 1411.1784*.

Murphy,R.F. (2005) Location proteomics: a systems approach to subcellular location. *Biochem Soc Trans*., **33**, 535–538.

Pathak,D. *et al*. (2016) Context encoders: feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, pp. 2536–2544.

Peng,T. and Murphy,R.F. (2011) Image-derived, three-dimensional generative models of cellular organization. *Cytometry Part A*, **79**, 383–391.

Radford,A. *et al*. (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv Preprint arXiv: 1511.06434*.

Rajaram,S. *et al*. (2012) Phenoripper: software for rapidly profiling microscopy images. *Nat. Methods*, **9**, 635.

Reed,S. *et al*. (2016) Generative adversarial text to image synthesis. *arXiv Preprint arXiv: 1605.05396*.

Ronneberger,O. *et al*. (2015) U-net: convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, Munich, pp. 234–241.

Ulman,V. *et al*. (2016) Virtual cell imaging: a review on simulation methods employed in image cytometry. *Cytometry Part A*, **89**, 1057–1072.

Wu,Y. *et al*. (2010) Quantitative determination of spatial protein-protein correlations in fluorescence confocal microscopy. *Biophys. J*., **98**, 493–504.

Yi,Z. *et al*. (2017) Dualgan: unsupervised dual learning for image-to-image translation. *arXiv Preprint arXiv: 1704.02510*.

Zhang,H. *et al*. (2016) Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv Preprint arXiv: 1612.03242*.

Zhao,S. *et al*. (2017) Towards deeper understanding of variational autoencoding models. *arXiv Preprint arXiv: 1702.08658*.

Zhao,T. and Murphy,R.F. (2007) Automated learning of generative models for subcellular location: building blocks for systems biology. *Cytometry Part A*, **71**, 978–990.

Zhu,J.-Y. *et al*. (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv Preprint arXiv: 1703.10593*.