ADVANCED INDIVIDUAL COURSE IN COMPUTATIONAL BIOLOGY
DD2402, MASTERS IN MACHINE LEARNING
STOCKHOLM, SWEDEN 2022

Application of Generative Adversarial Networks in Biological Image Synthesis

Course Project Report

Athanasios Charisoudis

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Authors

Athanasios Charisoudis <thacha@kth.se>
MSc. in Machine Learning / Research Engineer
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Erik Fransén
Dept. of Computational Science and Technology
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Supervisor

Erik Fransén
Dept. of Computational Science and Technology
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Abstract

Generative Modelling, a branch of Machine Learning that focuses on generating realistic-looking samples, has traditionally constituted the upper bound of what Machine and Deep Learning models can achieve. This regime has completely changed in recent years, especially after 2014, when I. Goodfellow presented a generative model comprising two competing neural networks: Generative Adversarial Network (GAN) [8]. Subsequently, a plethora of models based on GAN have been proposed with impressive results.

Concurrently, more and more research is devoted to developing techniques for demystifying biological functions at a cellular level. Among its purposes is creating artificial intelligence systems that provide insights into how different proteins operate and the way their co-location is connected with higher-level functions as captured in spectroscopic techniques. In an endeavor to apply modern machine learning techniques to synthesize cells imaged with fluorescent microscopy, in this project we employ Generative Adversarial Networks.

In particular, we use the network architectures presented by Osokin et al. [19] to generate realistic images of yeast fission cells imaged by fluorescent microscopy, aspiring that by being able to do so, the networks must capture the correlations present in the localization of the various proteins of interest. In addition, these models are able to generate multichannel images and therefore circumvent one major limitation of fluorescent microscopy.

Keywords

GAN, Fluorescent Microscopy, DCGAN, Wasserstein Loss, Gradient Penalty, FID, IS, Classifier 2-Sample Test, Yeast Fission, Cell Image Synthesis, GFP, RFP

Acknowledgements

At this point, I would really like to express my thankfulness to professor Erik Fransén for giving me the opportunity to apply Deep Learning techniques in the Computational Biology context and get acquainted with both fluorescent microscopy and the specially designed models for this domain. In addition, I am grateful that KTH offers the opportunity of attending individual courses wherein the interested ones can delve a bit deeper into their beloved domains.

Acronyms

BCE	Binary Cross Entropy
C2ST	Classifier 2-Sample Test
DCGAN	Deep Convolutional GAN
EMD	Earth-Mover's Distance
FID	Frechet Inception Distance
FM	Fluorescent Microscopy
FP	Fluorescent Protein
GAN	Generative Adversarial Network
GFP	Green Fluorescent Protein
IS	Inception Score
JSD	Jensen-Shannon Divergence
RFP	Red Fluorescent Protein
PPL	Perceptual Path Length
VAE	Variational Autoencoder
WGAN	Wasserstein GAN
WGAN-GP	Wasserstein GAN + Gradient Penalty

Contents

Chapter 1

Introduction

As briefly described above, this project will be based mostly on the work of Osokin et al. entitled ”GANs for Biological Image Synthesis” presented in [19], and partially on the work of Dodgsod et al. entitled as ”Reconstructing regulatory pathways by systematically mapping protein localization interdependency networks” and presented in [3]. In particular, the aspiration here is to re-implement the work of Osokin et al. to the largest extent possible, as will be explained in more detail in the following section. A second aspect of the implementation, involves understanding and using the LIN dataset published as part of the work of Dodgsod et al. (that also constituted the training dataset for the Osokin et al.) for the purposes of the aforementioned tasks.

In the remainder of this chapter a brief background of the scientific areas involved is laid, followed by the problem statement and the relevant research questions.

1.1 Background

This project applies generative modelling techniques in the domain of Computational Biology. In particular, GAN models are trained on images of Fission Yeast cells imaged with Fluorescent Microscopy. Fluorescence, initially described by Sir G. G. Stokes in 1852, has seen a surge of popularity during the past couple of decades. It has formed the basis of Fluorescent Microscopy which has enabled geneticists to probe *biological events in living cells with unprecedented resolution* [19]. As a consequence, larger and larger amount of cell image data have become available, leading to the emerging

field of bioimage informatics [17] in order to analyze those with the prominent aid of computer vision methods.

High-resolution cell images are among the most sought-after in that field, since their availability enables the creation of techniques to quantitatively analyze them and/or gain useful insights of the underlying biological mechanisms. When compared to natural images, cell images have much simpler geometric structure, but the co-location of the different molecules may signify something important about a biological function, the presence of a disturbance or disease, among others. This highlights the importance of modeling such images; Fluorescent Microscopy images are used for our purposes with the fluorescent ”tags” being placed on certain proteins of interest.

1.2 Problem Statement and Purpose

The fact that the localization of the various depicted molecules in a cell image is strongly correlated with higher-order biological functions, places a key challenge when trying to synthesize such images. In this project, we try creating GAN models that when trained on Fluorescent Microscopy cell images, are able to generate new ones hardly distinguishable from the original. The models should, therefore, be able to capture all these underlying correlation factors in order for the synthetic samples to be relevant for biological applications.

One aspect for the research problem at hand, is thus the creation of synthetic cell images wherein the placement of tagged proteins closely resembles real-world such localization and biological functions; specially-designed GAN architectures are used for this. Another aspect of equal importance, is overcoming a main limitation of Fluorescent Microscopy, that of limited number of channels that can be simultaneously depicted from the cell under the lens. Both of these are thoroughly discussed in subsequent sections.

1.3 Goals

Tackling both of the research questions presented in the previous paragraph forms the purpose and goals of this project. The provided data contain 2-channel images (i.e. images with two proteins tagged with different colors) wherein the red channel depicts

the localization of a certain protein located in areas of active cell growth and the green depicting either one of a pool of polarity proteins.

Since the red channel depicts the same protein in all images, we aspire that our trained models will be able to generate images with multiple green channels, based on the their association with the common information found in the red channel. Synthetic such images, would enable simultaneous visualization of the co-location of multiple green-labeled proteins as if they had been imaged together; stepping closer to artificially bypassing the limited-number-of-channels impediment of fluorescent microscopy. A second, minor goal, is to visualize the dynamic evolution of the proteins localization as cells grow proceed in their life stages, as can be seen by the red-labelled protein.

1.4 Outline

The next chapter is devoted to the presentation of the theoritical concepts necessary for the interested reader in order to comprehend the techniques and/or their significance. In the third chapter, we provide details about the datasets used and the developed models. Finally, the fourth chapter contains training curves, final synthesized images and evaluation metrics, leaving the conclusions and future extensions for an extra small chapter at the end.

Chapter 2

Theoretical Background

We begin this chapter by providing a short yet concise introduction to GANs as well the particular type of such models that form the basis of our experiments. In addition the used evaluation metrics are described as well as some major ”tricks” used to make training more stable. Then, a short introduction to Fluorescent Microscopy (FM) follows, leaving some key points about Fission Yeast cells for the last section of this chapter.

2.1 Generative Adversarial Networks

Principal element in the structure of GANs that sets them apart from other (Deep) Generative Models, is the existence and concurrent training of two networks: the Generator who tries to generate samples as similar as possible to the ones in the training set, and the Discriminator which is trained to distinguish real from generated samples. At every training step, Discriminator receives both real samples from the training set and ones synthesized by the Generator and is trained to correctly classify each of them in a binary classification setting. At the same step, the Generator receives random (most commonly white) noise and transforms it through a series of decoding operations to an image which is then fed to the Discriminator. As can be seen in Figure 2.1.1, the classification loss at the output of the latter is used to train both networks using Gradient Descent. Notably, at each of the two sub-stages of the training step, one network remains ”frozen”.

The Discriminator tries to minimize the classification objective for both the real and

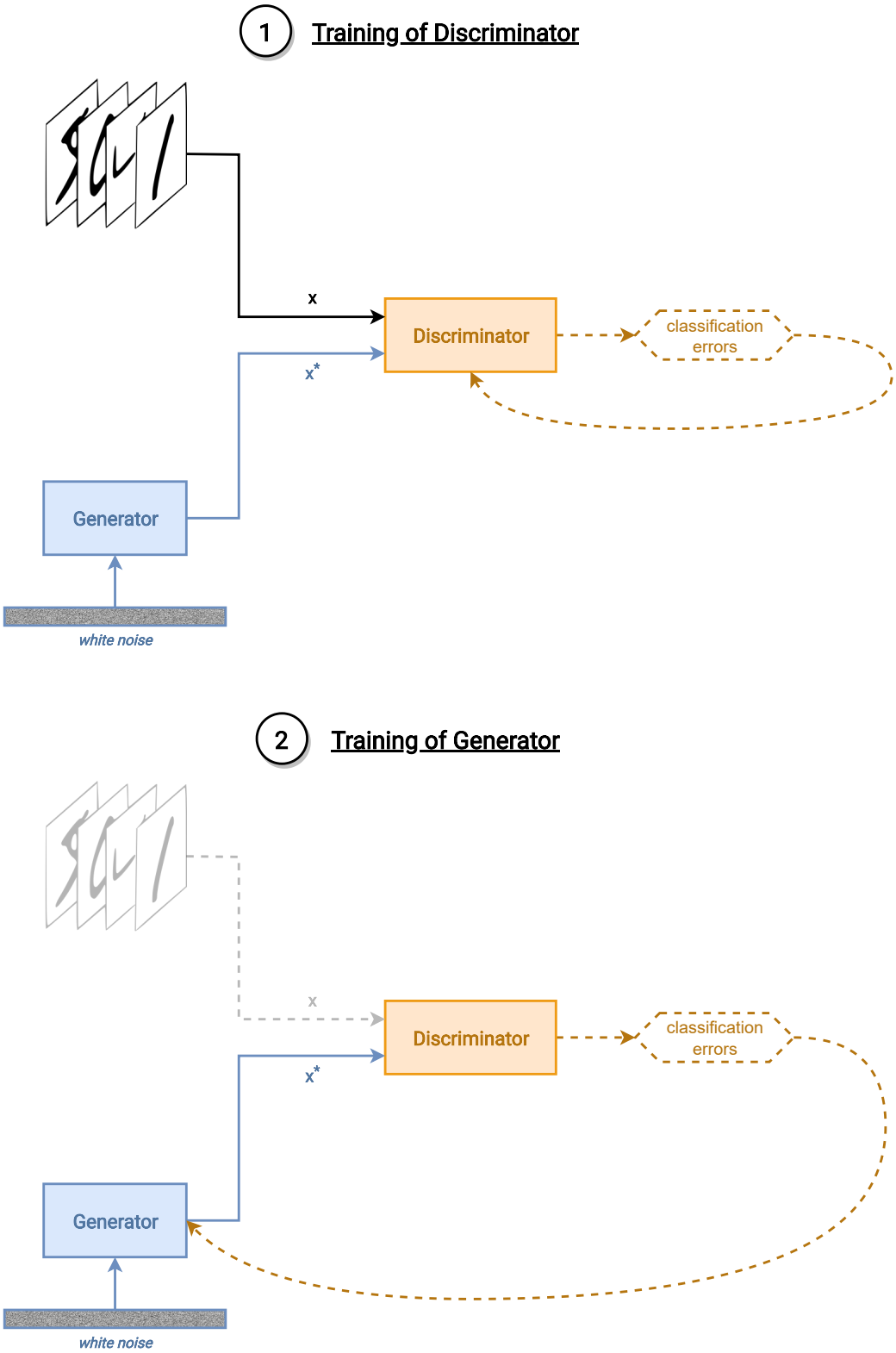


Figure 2.1.1: Visualization of GAN training step.
Source: Reconstruction from *"GANs in action: Deep learning with generative adversarial networks"* [13]

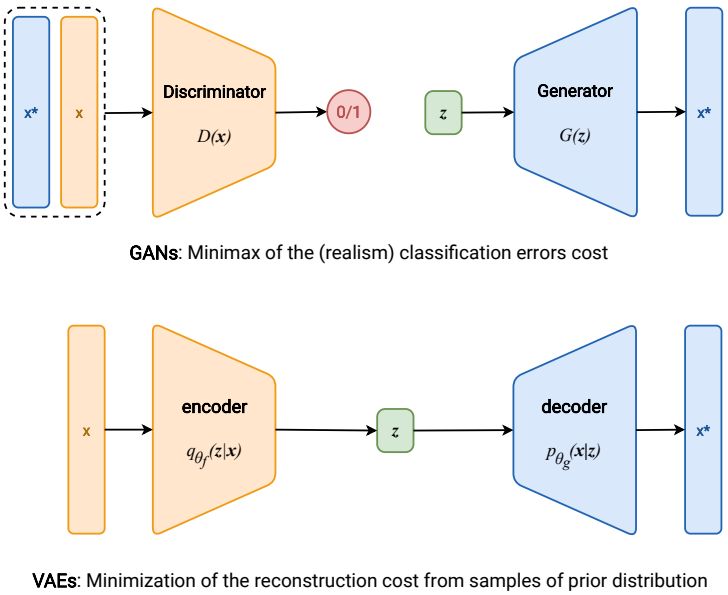


Figure 2.1.2: Comparison of the structure and training objectives between GANs and VAEs.
Source: Reconstruction from "*Flow-based Deep Generative Models*", Lilian Weng, 2018 [27]

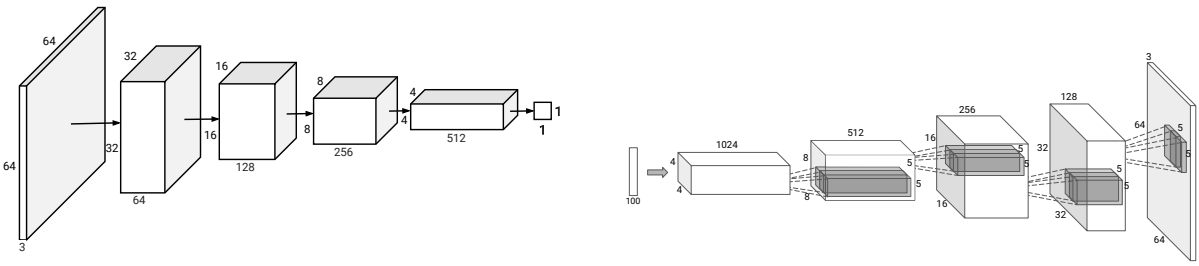
generated samples, whereas the Generator tries to maximize that objective for the samples it produces; GAN training is therefore not reliant on log-likelihood like training criteria but can be seen a zero-sum minimax game between the two models, as can be seen in the comparison with Variational Autoencoder (VAE) [12] in Figure 2.1.2.

2.1.1 Convolutional GANs

The initial GAN architecture as was presented by I. Goodfellow et al. in [8] comprised fully-connected layers and was applied in the simpler MNIST digits [14] dataset. To boost performance in more complex image datasets, Radford et al. introduced the Deep Convolutional GAN (DCGAN) [21] which replaced fully connected layers with convolutional (or transposed convolutional) ones in both the Discriminator and the Generator. We use modified versions of this network depicted in Figure 2.1.3 in this project.

2.1.2 GAN Training Objective

In contrast with other Deep Generative Models, the training objective of each of the two networks comprising the GAN does not solely rely on its parameters but also on the ones of the other network. Let θ_G be the trainable parameters of the Generator and



(a) DCGAN Discriminator
Source: "*Semantic Image Inpainting with Perceptual and Contextual Losses*", Yeh et al. 2016 [28]

(b) DCGAN Generator
Source: "*Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*", Radford et al. [21]

Figure 2.1.3: DCGAN Components for an image dataset containing 64×64 images.

$\boldsymbol{\theta_D}$ those of the Discriminator, while J_G and J_D their individual cost functions (i.e. the ones that each tries to minimize). In a zero sum, there is an equilibrium, where none "player" can improve their score, known as *Nash Equilibrium* [18]; this happens in GAN training when the samples produced by the Generator are indistinguishable from the ones in the training set. If the Binary Cross Entropy (BCE) (with the labels for real samples being 1 and for fake 0), then for the Discriminator the following objective has to be minimized:

$$\begin{aligned} J_D\left(\boldsymbol{\theta_D}, \boldsymbol{\theta_G}\right) &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} log\left(h\left(x^{(i)};\boldsymbol{\theta}\right)\right) + \left(1-y^{(i)}\right) log\left(1-h\left(x^{(i)};\boldsymbol{\theta}\right)\right)\right] \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[log\left(D\left(x^{(i)};\boldsymbol{\theta_D}\right)\right) + log\left(1-D\left(G\left(\boldsymbol{z}^{(i)};\boldsymbol{\theta_G}\right);\boldsymbol{\theta_D}\right)\right)\right] \\ &= -\frac{1}{m}\sum_{i=1}^{m} log\left(D\left(x^{(i)};\boldsymbol{\theta_D}\right)\right) - \frac{1}{m}\sum_{i=1}^{m} log\left(1-D\left(G\left(\boldsymbol{z}^{(i)};\boldsymbol{\theta_G}\right);\boldsymbol{\theta_D}\right)\right) \\ &\approx -\mathbb{E}_{x\sim p_{data}} log\left[D\left(x\right)\right] - \mathbb{E}_{z\sim p_{prior}} log\left[1-D\left(G\left(z\right)\right)\right] \end{aligned} \tag{2.1}$$

where $D(x)$ is Discriminator's output, $G(z)$ the one of the Generator given a random noise vector z, p_{data} the distribution from which training data are drawn (for images this will be of very high dimensionality) and p_{prior} the distribution of the random noise. Since, the Discriminator outputs probability, $D(x) \in [0,1]$, in order for 1 to be minimized it must learn to assign high probability in the real samples and low on Generator's ones.

The Generator in turn, tries to maximize the second term of Discriminator's objective

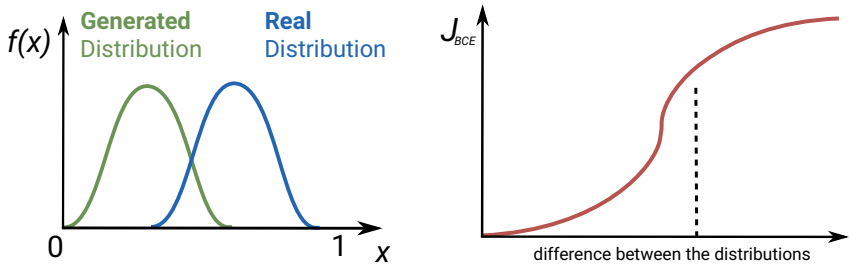


Figure 2.1.4: Visualization of saturation issue with BCE.
Source: Reconstruction from Generative Adversarial Networks Specialization, Zhou et al., DeepLearning.AI, 2021 [29]

1 (which is the only it can affect). Therefor, Generator's objective would be:

$$\begin{aligned} J_G\left(\boldsymbol{\theta_G}, \boldsymbol{\theta_D}\right) &= \frac{1}{m}\sum_{i=1}^{m}\left[\left(1-y^{(i)}\right) log\left(1-h\left(x^{(i)};\boldsymbol{\theta}\right)\right)\right] \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[log\left(1-D\left(G\left(\boldsymbol{z}^{(i)};\boldsymbol{\theta_G}\right);\boldsymbol{\theta_D}\right)\right)\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} log\left(1-D\left(G\left(\boldsymbol{z}^{(i)};\boldsymbol{\theta_G}\right);\boldsymbol{\theta_D}\right)\right) \\ &\approx \mathbb{E}_{z\sim p_{prior}}\, log\left[1-D\left(G\left(z\right)\right)\right] \end{aligned} \tag{2.2}$$

without the minus since it tries to minimize the above.[1]. As was showed in [8], this setup for GAN training has the extra property that the cost function is asymptotically consistent with minimizing the Jensen-Shannon Divergence (JSD) between the data distribution and the one "learned" by the Generator (i.e. from where the fake samples are drawn).

2.1.3 WGAN: Towards Non-Saturating Objective

One major drawback of using the BCE criterion, is the saturation induced by the use of logarithms. As can be seen in Figure 2.1.4 when the two distributions are significantly different (e.g. at the beginning of the training), BCE tends to be more saturating, leading to training instabilities; mainly vanishing gradients and mode collapse, both of which destabilize training or even prohibit it.

To circumvent that issue, various training criteria have been proposed. The most effective one is considered by the literature to be the Earth-Mover's Distance (EMD)

[1]Since the first term of is only dependent on the training set, it is common in literature to denote $J_G\left(\boldsymbol{\theta_G}, \boldsymbol{\theta_C}\right) = -J_D\left(\boldsymbol{\theta_C}, \boldsymbol{\theta_G}\right)$ which justifies the parallelism with a zero-sum game.

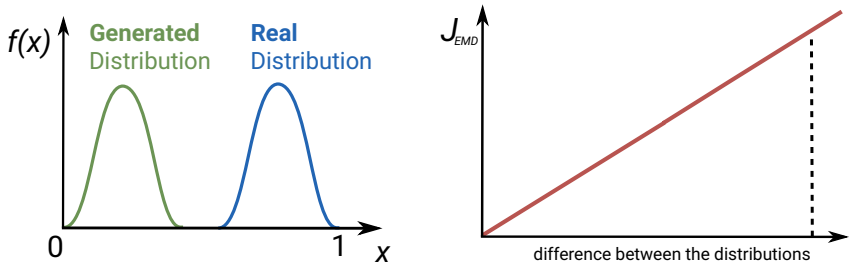


Figure 2.1.5: Visualization of the EMD between two distributions and for one suitable function. **Source:** Reconstruction from Generative Adversarial Networks Specialization, Zhou et al., DeepLearning.AI, 2021 [29]

or Wasserstein-1 Distance, which measures the work required to make the generated distribution "equal" to the data's one. Mathematically, EMD can be defined as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_D, \mathbb{P}_G)} \mathbb{E}_{(x,y)\sim\gamma}\left[\, \|x - y\| \,\right], \tag{2.3}$$

where $\Pi(\mathbb{P}_D, \mathbb{P}_G)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_D$ and $\mathbb{P}_G$. As explained by Arjovsky et al. [1], using the Kantorovich-Rubinstein duality to get rid of the infinum in 2.3, we end up with:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)] \tag{2.4}$$

where the supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$; Figure 2.1.5 visualizes the distance for one such function.

As the authors of WGAN [1] suggest, to enforce the Discriminator to be 1-Lipschitz continuous (and thus avoid optimizing over the set of functions), one has to clip its weights to a constant interval. Alternatively, as Gulrajani et al. suggested in [9], we can encourage the Discriminator to be 1-Lipschitz continuous by penalizing the offset of its gradient's norm from 1 at random input points; a regularizing term is used for this. This choice of the objective function is conventionally called Wasserstein GAN + Gradient Penalty (WGAN-GP) and forms the preferred form of GAN training method throughout this project. Mathematically, WGAN-GP loss could be written as:

$$W_D\left(\boldsymbol{\theta_D}, \boldsymbol{\theta_G}\right) = -\,\mathbb{E}_{x\sim p_{data}}\, D\left(x\right) + \mathbb{E}_{z\sim p_{prior}}\, D\left(G\left(z\right)\right) + GP \tag{2.5}$$

for the discriminator (where GP is defined in [9]), and as:

$$W_G\left(\boldsymbol{\theta_D}, \boldsymbol{\theta_G}\right) = -E_{z\sim p_{prior}} D\left(G\left(z\right)\right) + GP \tag{2.6}$$

for the Generator (both networks try to minimize their ”own” losses).

2.2 Fluorescent Microscopy

One of the main limitations of optical cellular microscopy, is that many molecules appear the same color and thus are difficult to distinguish and study independently. At the middle of past century scientists discovered a protein (in some jellyfish cells) that glows when exposed to ultraviolet light. This was later identified as the Green Fluorescent Protein (GFP) [2, 26] which when undergo light of specific spectrum (called the absorption spectrum) it fluoresce, emitting green light at its emission spectrum with some μs delay. Various such proteins have been discovered since then, with a popular one being the Red Fluorescent Protein (RFP) and its peers [23].

Geneticists, can exploit this fluorescence phenomenon of some proteins and attach them to the proteins of interest via genetic engineer; a process called ”*tagging*”. Tagging different proteins with different Fluorescent Protein (FP) tags, one can simultaneously observe the co-location of the former in the cell. Higher color values (i.e. higher values at the corresponding channel in FM) correspond to higher concentrations of the tagged proteins at these locations. A model trained to generate such images, should be able to capture the correlations between the concentrations between different proteins and the spatial properties of the cell. Central limitation in FM is the inability to capture and visualize more than 3-4 channels due to overlaps in the absorption spectra [19].

2.3 The LIN Dataset

The dataset considered in this project, the LIN published by Dodgson et al. [3], contains fission yeast cell images and was used to study polarity networks (i.e. interdependency networks of proteins that correspond to cellular polarity factors). Fission yeast (*Schizosaccharomyces pombe*) is a popular model unicellular eukaryote for the study of the cell cycle. Its cells are rod-shaped and grow lengthwise (from 7 to $14\mu m$) maintaining a constant width of $4\mu m$[2]. Then they form a *cytokinetic ring in the middle, which is responsible for cleaving the mother cells into two daughters* [20].

[2]As was described by Gómez and Forsburg in [7], fission yeast cells initially grow at the pre-existing end only, until they reach a certain length when they switch to bipolar growing.

Each of the 180K images contained in LIN, is a stack of two FM channels centered on a cell: the red wherein the localization of `BGS4` (uniprot *BGS4_SCHPO*) is depicted, and the green where either one of 41 proteins that correspond to different polarity factors are depicted[3]. It's interesting to note that `BGS4` is responsible for cellular wall remodeling and thus localizes in the areas of active cell growth, while at the same time the size of the fission yeast cell is strongly related to the cell life stage. Therefore, tracking `BGS4` in the red channel one can conclude about the cell's "age"; this will appear particularly useful later when we try to model dynamical evolution of protein localizations through the cell's aging.

In this project, we follow the approach of Osokin et al. and restrain our focus to 6 out of the 41 polarity factors imaged (independently) in LIN dataset. These were: `Alp14`, `Arp3`, `Cki2`, `Mkh1`, `Sid2` and `Tea1`. Each of these proteins (will also be called "classes" interchangeably in what follows) control biological function "cellular polarity" but in slightly different ways. The first 3 training samples from each class are given in Figure 2.3.1. The images are 48×80 pixels and totalling at approximately 27K for these 6 training classes.

[3]A pixel side corresponds to $100nm$ in real size, and the green channel is taken $300nm$ away in the z-axis from the red. See [3] for more technical details.

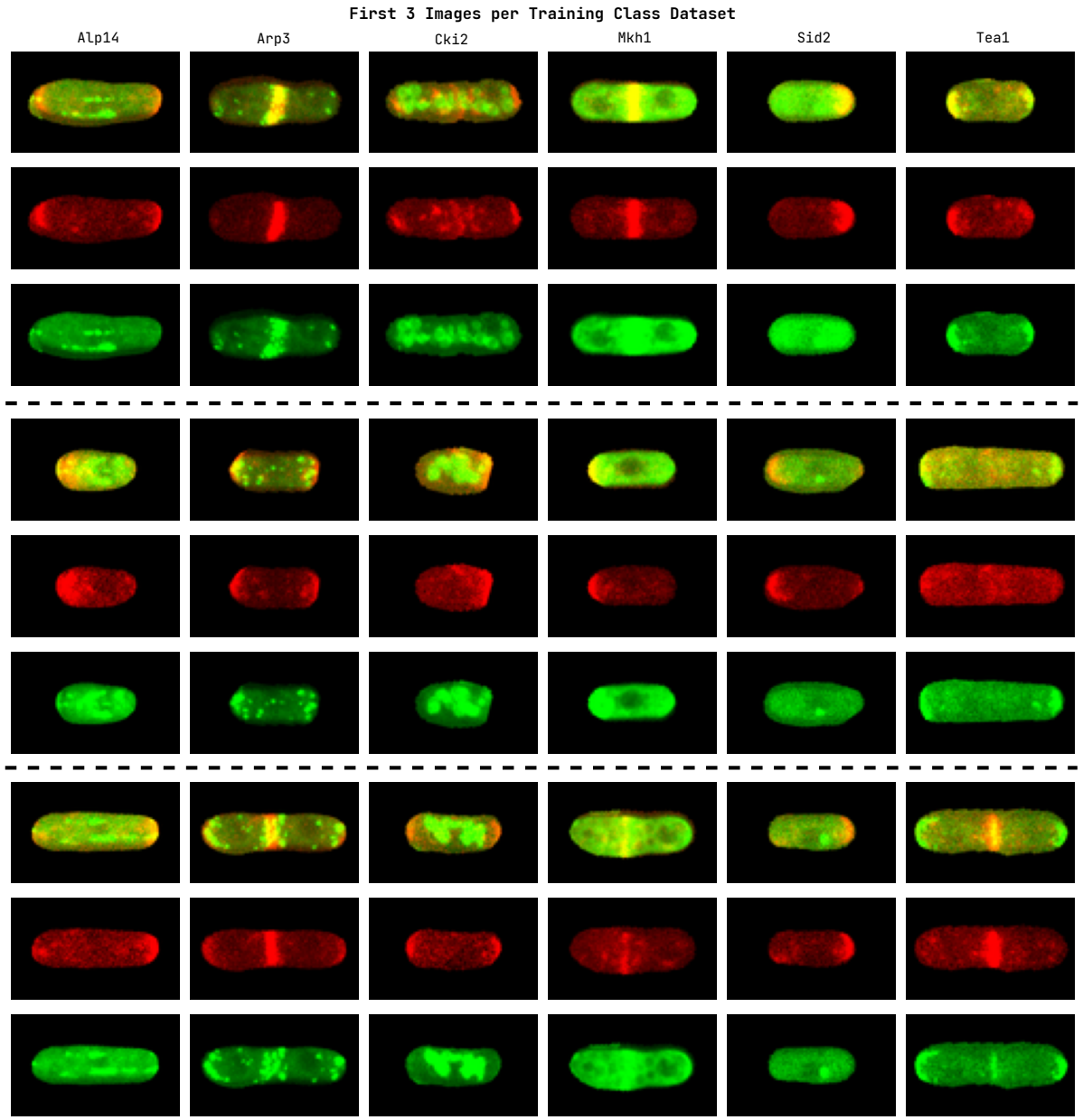


Figure 2.3.1: Visualization of images contained in LIN dataset: Each column is one of the 6 classes / green-tagged proteins. Each triplet of rows is one training sample, depicted as a whole, the red-tagged protein and the green-tagged protein portions that comprise it, respectively.

Chapter 3

Methodology

In our re-implementation of Osokin et al. [19], we focused on understanding and implementing the GAN models to synthesize 2-channel (as the training set) and multi-channel images. In what follows, a short enumeration of the developed models is given, followed by salient technical information of each model and ending with the metrics used to evaluate them.

3.1 GAN Models

The GAN models that were developed were the following:

- **One-class, non-separable**: similar to a naive form of DCGAN architecture, the Generator is trained using 2-channel images, the red-tagged `Bgs4` and one green-tagged protein or "class" at a time. In total, 6 such models were created (one for each of the 6 classes of cellular polarity).
- **One-class, separable**: as above, but now the generator follows the separable architecture as shown in Figure 3.1.1 and explained below.
- **Multi-Channel, separable**: where instead of using 2-channel images, the Nearest-Neighbor dataset was used, containing 7-channel images as can be seen in Figure 3.1.2. In addition separable architecture was employed, meaning that the 6 green channels are generated conditioned on the red; the proposed architecture features 1 generator block that will output 6 green channels and 1 that will output the red-channel (with the causality directions as mentioned in [19]).

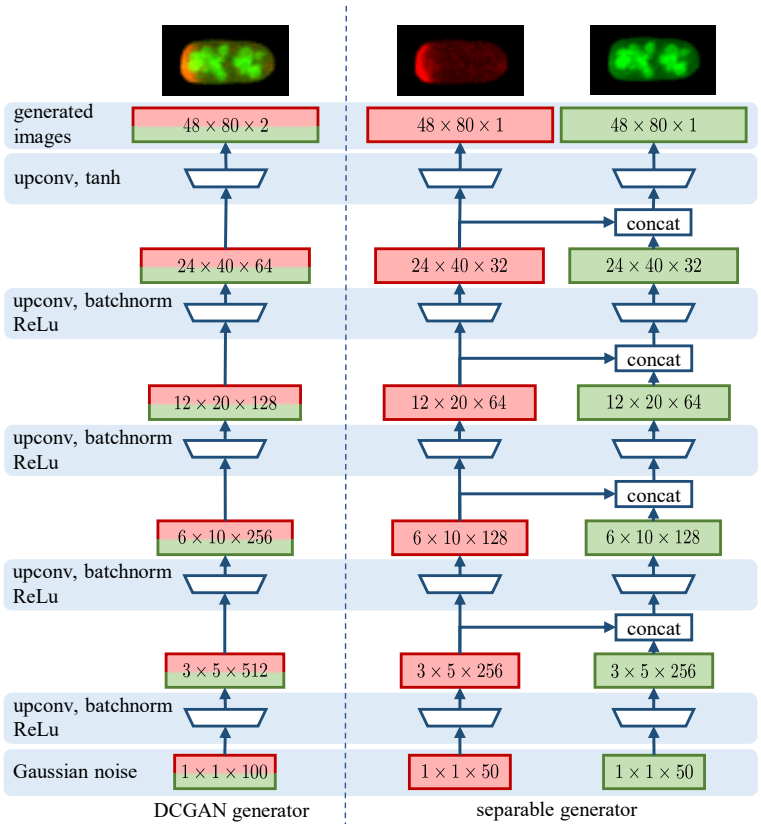


Figure 3.1.1: Non-separable vs. Separable Generator architectures.
Source: *"GANs for Biological Image Synthesis"*, Osokin et al., 2017 [19]

- **Star-Shaped, separable**: where the Generator again outputs (6+1)-channel images but now each green channel is independent of the others, and conditioned only on the red channel (i.e. there are one-way connections from the red "tower" to c green ones). This model is trained on the original 2-channel images and was found to be the most performant, both qualitatively and in terms of evaluation metrics.

3.1.1 Modelling Causal Dependencies

Central to the Generators design is the modeling of the causal dependencies of the green-tagged proteins to the red-tagged one, i.e. of the green channels to the red. This was done by modifying the transposed convolutional layers of the DCGAN's Generator [21], by splitting their filters and providing one-way connections, leading to the *separable* model which is depicted in Figure 3.1.1. It is noteworthy that the Discriminator network receives 2-channel (or (c+1)-channel ones in general) regardless if the generator is separable or not.

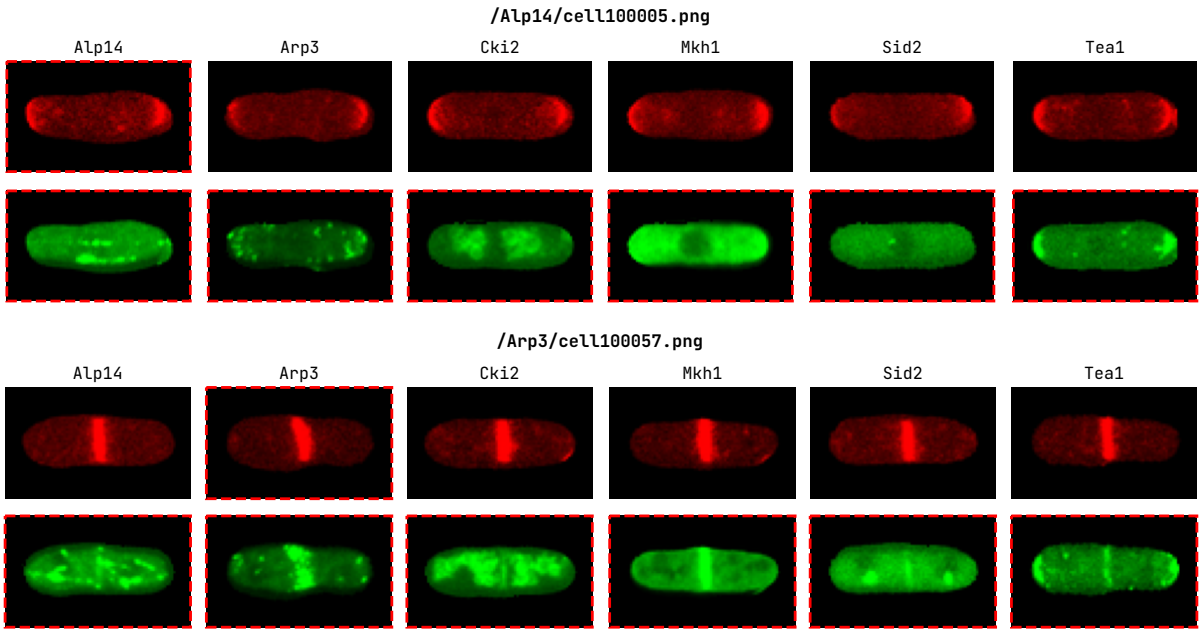


Figure 3.1.2: Two examples of multi-channel images present in our Nearest-Neighbors dataset. In the first and third rows the nearest neighbors (in every other class) in the red channel are given for the marked red channels. Below each of these rows, are the corresponding green channels of the original 2-channel images. Finally, the 7-channel images are composed by concatenating the marked channels.

3.1.2 Generating Multi-Channel Images

To generate multi-channel images, two different approaches were tried. Firstly, multi-channel images were mined from the original dataset using the 1st nearest neighbor (pixel-space distance) in the red channel for every other class than the image of interest, and stacking its green channel the image's nearest neighbors green channels. This nearest-neighbors dataset, samples of which are given in Figure 3.1.2, was used to train DCGAN variants that output $c + 1$ channels where c is the number of green channels. Secondly, a "*Star-Shaped*" Generator was used, that can generate multi-channel images but trained directly from the 2-channel images present in LIN dataset. It is notable, that when generating multi-channel images c Discriminators are used, each trained on real images of one training class and keeping only the red+1 green channel from the generated ones.

3.1.3 Evaluation Metrics

In order to evaluate the fidelity of the synthesized images a number of metrics was employed. The reader should note that these metrics solely -to a larger or lesser extent- approximate human judgement; sometimes employing humans for the final evaluation

is inevitable. The used evaluation metrics for the synthesized images were:

- **Inception Score (IS)** [25]: we feed every generated image through a trained InceptionNet [24], and record its classification output y given each input x. Then, the score is defined as:

$$\begin{aligned} IS &= exp\left(\mathbb{E}_{x\sim p_{model}}\left[KL\left(p\left(y|x\right)||p\left(y\right)\right)\right]\right) \\ &= exp\left(\mathbb{E}_{x\sim p_{model}}\left[p\left(y|x\right) * log\left(\frac{p(y|x)}{p(y)}\right)\right]\right) \end{aligned} \tag{3.1}$$

  Analytically, IS can take values in the range $[0, \infty)$, though practically due to the categorical distribution the values in can take are bound below from 1.0 (high-entropy output, low-fidelity samples) and above by number of classes (then the samples will have both high fidelity and high diversity). For the purposes of this project we used an InceptionNet v3 trained on ImageNET and therefore the number of classes is 1000.

- **Frechet Inception Distance (FID)** [10]: in this metric we do not use the output of trained classifier but the values from the last pooling layer (i.e. we use the classifier as an extractor of ImageNET embeddings). It uses the real images as well, and was found to be much more compliant with human judgement. Gaussian distributions are fitted in the real and fake embeddings and then FID is computed as the Fréchet (or dog-walking) distance (as was described by Fréchet in [5]) between the two distributions:

$$\left.\begin{aligned} X &\sim \mathcal{N}\left(\boldsymbol{\mu_X}, \Sigma_X\right) \\ Y &\sim \mathcal{N}\left(\boldsymbol{\mu_Y}, \Sigma_Y\right) \end{aligned}\right\} \Longrightarrow d(X,Y) = \left\|\boldsymbol{\mu_X} - \boldsymbol{\mu_Y}\right\|^2 + Tr\left(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X\Sigma_Y}\right) \tag{3.2}$$

- **Precision, Recall, F-Score for GANs** [22]: this is considered as the most accurate (w.r.t. human opinion) metric for comparing synthetic images with real ones both in terms of fidelity and of diversity. After retrieving the image embeddings, it approximates their manifolds using intra-manifold distances and a couple of binary rules. Then the Precision and Recall metrics are defined using the amount of overlap between the two approximated manifolds over the one of

the generated or the real respectively. F-Score is defined as:

$$\text{F}_1\text{-Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

- **Classifier 2-Sample Test (C2ST)** [15]: this metric is the only used by Osokin et al. to evaluate their models. In the GAN evaluation setting, it encompasses re-training a Discriminator network on portion of the test set of images (keeping the Generator frozen) and then using its mean classification score on the held-out portion of the test set as the value for C2ST. In our context, the Discriminator was identical in design to the one used in the GAN model under training. It noteworthy, that this is a pretty expensive computationally metric (e.g. when training on multi-channel images we have to train 6 different Discriminators to compute the C2ST); we defer its computation for the final model checkpoints only. In addition, in this project we only use BCE objective to train the Discriminators of C2ST, whereas Osokin et al. also tried with WGAN and WGAN-GP ones. Lower C2ST values usually correspond to sharper images closer to the real ones.

Finally, as in Osokin et al. [19], we also use a surrogate technique to evaluate and compare our models: we try reconstructing images of a held-out test set and computing the resulting pixel-space distance between the image and its reconstruction.

3.2 Experiments

The first and most basic set of experiments is training and synthesizing images from the 4 model variants presented above. This was not trivial, since we chose not to consult the code accompanying [19] but instead try to design and train models based on the written specifications.

Then, we tried evaluating and comparing our models based on IS, FID, F-Score and C2ST. For fairer comparison, we used the implementation of C2ST given in Osokin et al. [19]. We provide our results in Tables similar to the Tables 1-3 of [19].

Finally, we provide interpolation plots to depict the dynamical evolution of generated protein localizations using spherical interpolation in the latent space. For these last set of experiments we used our trained Star-Shaped model only to scale down

computational requirements.

Chapter 4

Results

The results are given in the order the corresponding experiments were listed. We try to constantly compare our results with the referenced ones in Osokin et al. [19].

4.1 Synthesizing Images from Trained Models

4.1.1 One-class Non-separable GAN

The first model that was trained was a 2-channel DCGAN to produce images resembling the ones of yeast cells tagged with `Bgs4` (red channel) + `Apl14` (green channel). So for this type of GAN, only one class (2-channel) images were used to train the model. Here we show the trained model on images of `Bgs4` (red) + `Alp14` (green).

The training was really unstable especially if the choice of criterion was a saturating one, such BCE between the Discriminator network's prediction and the target values (1 for real, 0 for generated images). We ended up using Wasserstein loss and Gradient Penalty (WGAN-GP) to make things work. In Figure 4.1.1 we provide synthesized versus real images for the one-class non-separable GAN, while next we list the final evaluation metric values for the (trained) generator.

The final evaluation metrics on the training set are given below. To compute those 3262 samples (i.e. all training samples of `Alp14`) were used and a pre-trained classifier to extract ImageNet embeddings.

- **FID=8.035** (lower is better)

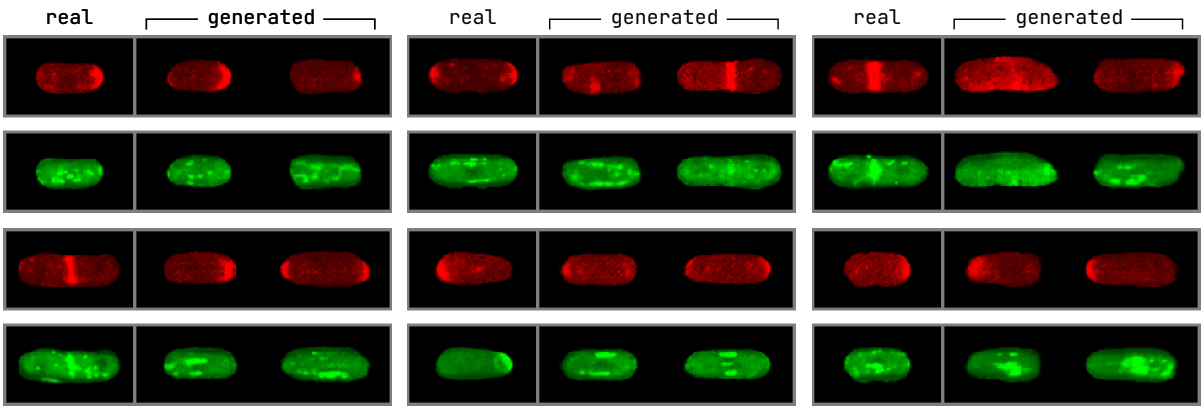


Figure 4.1.1: Real vs. Generated samples from our trained 1-class GAN after 3200 epochs. The odd rows present the red channels (`Bgs4`) while the even ones present their corresponding green channels (`Alp14`).

- **F1-Score=0.913** (precision=0.890, recall=0.937) (higher is better)
- **IS=1.834** (higher is better)
- **PPL=1.3e-13** (lower is better)

The same metrics plus C2ST are given below for the test (hold-out) set (for the C2ST 10 runs were performed on different portions of the test-set, see Osokin et al. [19]):

- **FID=8.035** (lower is better)
- **F1-Score=0.913** (precision=0.890, recall=0.937) (higher is better)
- **IS=1.834** (higher is better)
- **PPL=1.3e-13** (lower is better)
- **C2ST=-0.531 $\pm$ 0.32** (lower is better, averaged over 10 folds)

4.1.2 One-class Separable GAN

Given the instabilities during training a single-class model we moved on to train all 6 GAN models (one for each class or polarity factor in the green channel) simultaneously, *averaging the individual losses before each backward pass*. This played a key role in making the training more stable and at the same time we had trained 6 models in less than 6×the time of training single class models. Therefore for this next section we present the results of training 6 independent one-class (2 channel) GAN models using the WGAN-GP loss. Aside from leaking the information (via averaging the loss) we

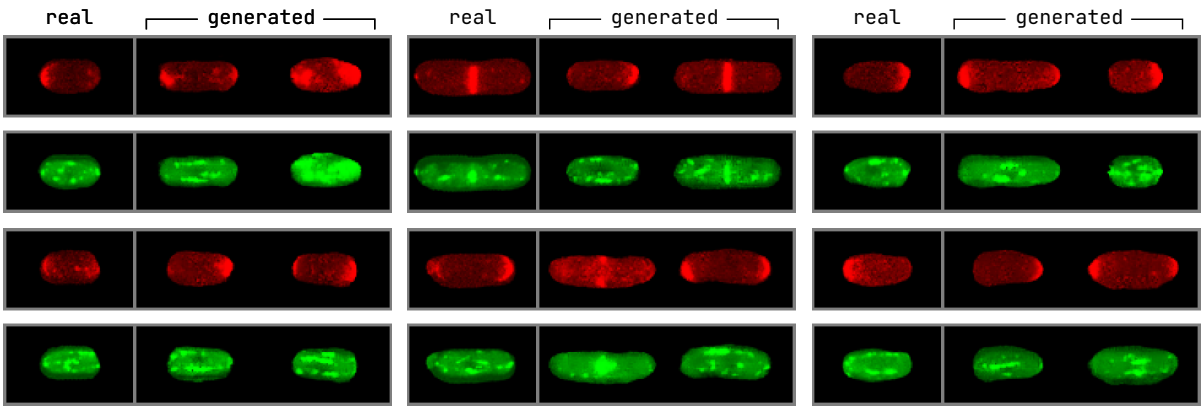


Figure 4.1.2: Real vs. Generated samples from the first of the six trained 1-class GAN models with Separable Generator architecture (after 543 epochs). The odd rows present the red channels (`Bgs4`) while the even ones present their corresponding green channels (`Alp14`).

also used the separable model architecture as depicted on Figure 3.1.1, for each model. Each model was fed with 2-channel images of its own class.

The final (mean over the 6 classes) evaluation metrics on the training set are:

- **FID=9.179** (lower is better)
- **F1-Score=0.910** (precision=0.889, recall=0.933) (higher is better)
- **IS=1.728** (higher is better)
- **Perceptual Path Length (PPL)=2.3e-2** (lower is better)

The above metrics and especially FID and F1 indicate that (on average) all models have learned to generate cell images the majority of which are not distinguishable from samples of the corresponding true data distributions, even from the 543th epoch. Same generated images are given in the figures that follow. In addition PPL values seem higher than the non-separable GAN since training with WGAN loss leads to higher diversity and sharper images and thus the PPL tends to be higher [11].

Comparing Figure 4.1.2 of the Separable Generator with the corresponding Figure 4.1.1 of the Non-Separable one, it seems that **the separable architecture produces sharper results with richer variations at an earlier stage than its non-separable counterpart**, a conclusion that goes along similar findings presented in the paper. Below, we also provide a figure containing real and generated images from all trained classes/models.

The same metrics plus C2ST are given below for the test (hold-out) set (for the C2ST 10

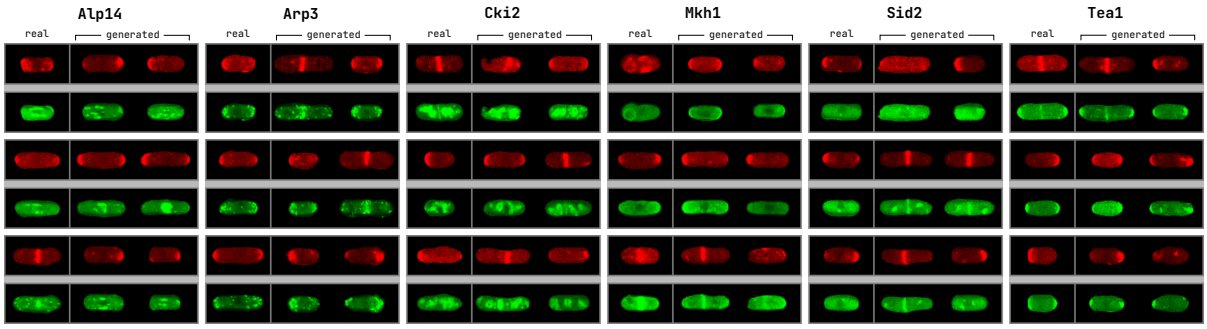


Figure 4.1.3: Real vs. Generated samples from all six 2-channel GAN models with Separable Generator architectures (after 543 epochs). The odd rows present the red channels (`Bgs4`) while the even ones present their corresponding green channels (the depicted protein/class is given at the top).

runs were performed on different portions of the test-set, see Osokin et al. [19]):

- **FID=9.41** (lower is better)
- **F1-Score=0.904** (precision=0.871, recall=0.923) (higher is better)
- **IS=1.718** (higher is better)
- **PPL=8.9e-3** (lower is better)
- **C2ST=1.931 ± 0.27** (lower is better, averaged over 10 folds; for `Alp14` class)

4.1.3 Multi-class Separable GAN using NN Images

The first architecture that produces actual multichannel images (not 2-channels only) is the Multichannel Separable GAN model. This is similar to the one-class GAN model but it is trained to output (1+6)-channel images with 1 red channel and 6 green ones, each corresponding to images of the corresponding class. The training set was obtained using nearest neighbors on the red channel, as explained above. WGAN-GP loss was used for training.

For the evaluation purposes, the red channel was duplicated 5 times (forming six 2-channel pairs given a 7-channel output) and each channel pair was compared against real images from the green channel's class. Having such a trained generator we can produce multiple green channels given the red one, thus overcoming known limitations of florescence microscopy. The final (mean) evaluation metrics on the training set are:

- **FID=4.341** (lower is better)

- **F1-Score=0.928** (precision=0.929, recall=0.928) (higher is better)
- **IS=1.701** (higher is better)
- **PPL=6.6e-3** (lower is better)

The same metrics plus C2ST are given below for the test (hold-out) set (for the C2ST 10 runs were performed on different portions of the test-set, see Osokin et al. [19]):

- **FID=4.448** (lower is better)
- **F1-Score=0.904** (precision=0.899, recall=0.908) (higher is better)
- **IS=1.599** (higher is better)
- **PPL=8.1e-3** (lower is better)
- **C2ST=3.104 $\pm$ 0.60** (lower is better, averaged over 10 folds; for `Alp14` class)

The above metrics and especially FID and F1 indicate that (on average) all models have learned to generate cell images the majority of which are not easily distinguishable from samples of the corresponding true data distributions, even after only 198 epochs. In addition, it seems that the multichannel GAN formulation results in significant improvements in the final evaluation metrics. Some generated images are given in Figure 4.1.4. C2ST exhibits higher values than the one-channel Separable GAN, probably due to the fact that it is at an earlier stage of training.

4.1.4 Star-Shaped Multichannel GAN

The second architecture that produces multichannel images (not 2-channels only) is the Star-Shaped (Separable) GAN model. The green channels are independently generated conditioned on the red one. WGAN-GP loss was used for training.

For the evaluation purposes, same logic as in Multichannel GAN was used. Having such a trained generator we can produce multiple green channels given the red one, thus overcoming known limitations of florescence microscopy. The final (mean) evaluation metrics on the training set are:

- **FID=3.441** (lower is better)
- **F1-Score=0.931** (precision=0.930, recall=0.931) (higher is better)
- **IS=1.871** (higher is better)

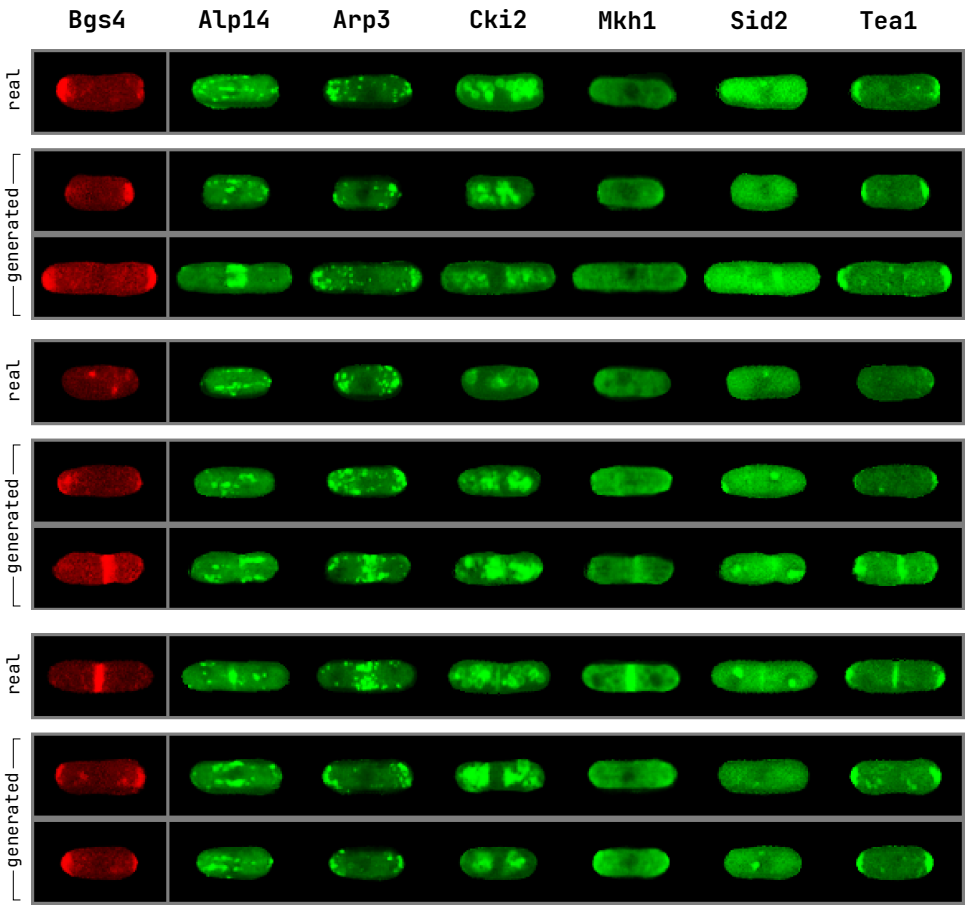


Figure 4.1.4: Real vs. Generated samples from the trained (1+6)-channel GAN model with Separable Generator architecture (after 198 epochs). The rows denoted as "real" present the red channel and its own plus five closest corresponding green channels extracted via NNs search; the rest rows contain one generation per each.

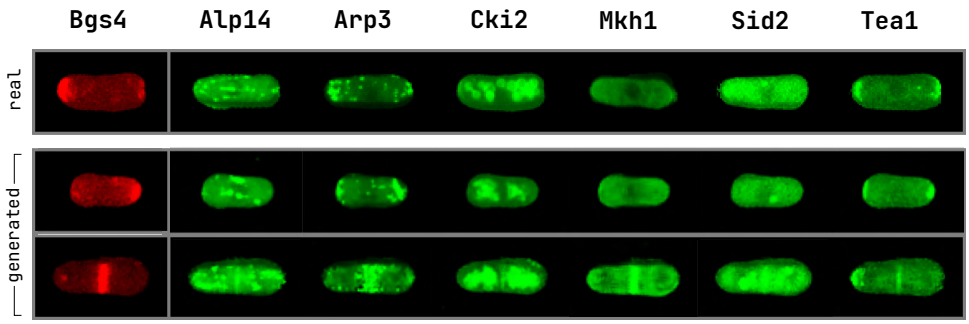


Figure 4.1.5: Real vs. Generated samples from the trained (1+6)-channel GAN model with Star-Shaped Generator architecture (after 1700 epochs). The rows denoted as "real" present the red channel and its own plus five closest corresponding green channels extracted via NNs search; the rest rows contain one generation per each.

- **PPL=9.1e-5** (lower is better)

The same metrics plus C2ST are given below for the test (hold-out) set (for the C2ST 10 runs were performed on different portions of the test-set, see Osokin et al. [19]):

- **FID=3.441** (lower is better)
- **F1-Score=0.931** (precision=0.930, recall=0.931) (higher is better)
- **IS=1.871** (higher is better)
- **PPL=9.1e-5** (lower is better)
- **C2ST=0.792 ± 0.33** (lower is better, averaged over 10 folds; for `Alp14` class)

Similar comments as in the previous multichannel model can be made here. However, this model was trained for 1700 epochs and thus its metrics are more robust and trustworthy. Some generated images are given in Figure 4.1.5. C2ST exhibits very similar values with the one-channel Separable GAN, which are also the best we got.

Comparing Figure 4.1.5 of the Separable Generator with the corresponding Figure 4.1.3 of independent generations we can observe two main differences:

1. **Higher Fidelity**: the fluorescent images generated by the Star-Shaped GAN present better realism and seem to have captured from that early stage of training the co-location of proteins in the cell life-cycle (as determined by the generated red channel).
2. **Ability to Visualize 6 GFPs and 1 RFP**: the generated channels are no longer independently produced but are explicitly conditioned on the way the red

channel is generated. This can serve our higher aim, which is to overcome a fundamental fluorescent microscopy limitation, that of using more tag channels simultaneously. In addition, via this way we essentially depict 6 GFPs all conditioned on the localization of the same red channel (tagged by a RFP), something almost impossible to obtain experimentally.

4.1.5 Visualizing the Cell Life Cycle

In the Figure 4.1.6, below, we also provide an interpolation on the latent space through the direction on cell ”age”. As was said earlier, `Bgs4` localizes in areas of active cell growth, while the size of the cell strongly relates to its age. As was mentioned in [19] and studied by Martin et al. in [**polarization_vs_stage**], **the changes in localization of the other proteins through cell life cycle, is well studied**; this experiment could therefore be used as a means of validating our synthesizer's intrinsic ”knowledge”.

To generate multichannel images at different we relied on the causal modeling of the green-tagged proteins to `Bgs4`, captured by our separable generators and especially on the Star-Shaped one. This ensures that the *output of the green channel will remain consistent to the red* [19] one as we interpolate between our generated images based on the red channel. The evolution of localization was done by spherically interpolating between latents that correspond to generations at different cellular size as perceived by the red channels.

Two noteworthy observations can be made from Figure 4.1.6:

1. `Arp3` is seen to gradually change its localization from the tips of the cell towards the middle as the cell progresses to the mitosis stage; this has also been found in relevant literature by Martin et al. in [16].

2. `Sid2` tends to localize around the actyomiosin ring and increase its concentration as the cell progresses towards its division, while it can also be seen concentrating around the two created nuclei during the mitosis. Both of these observations have been experimentally confirmed e.g. by Feoktistova et al. in [4].

Both of these correspond to previous experimental findings and enable qualitative validation of the successful training of the Star-Shaped model.

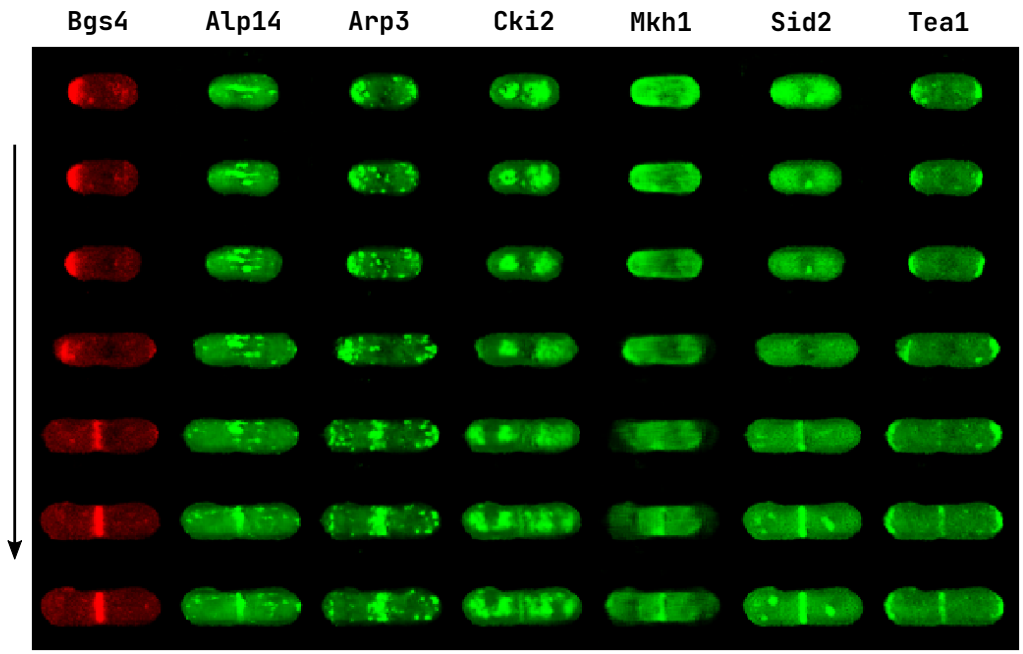


Figure 4.1.6: Evolution of proteins localization during different stages of the cell life cycle. The images were generated from a Star-Shaped WGAN-GP model after 1700 epochs of training.

4.2 Models Comparison

As the last part of our experiments we quantitatively compare the trained models. A point of attention is that due to limited computational resources, the models were trained for the same amount of epochs. The first, One-Class Non-Separable GAN was trained for 3200 epochs, while its Separable counterpart with WGAN-GP loss was trained for 543. The Multichannel Separable WGAN-GP model was trained just for 198 epochs since its training was much heavier (especially in terms of RAM needed), while the Star-Shaped Separable WGAN-GP model was trained for 1700 epochs. In the Table 4.2.1 that follows we have gathered the **final C2ST values** for each trained model; these values can therefore not be directly comparable.

As can be seen the values follow the same trends over classes and models. In particular:

- `Alp14` and `Tea1` seem the ”easiest” classes to synthesize samples for. All the models tend to exhibit lower C2ST values on these.
- **One-class models** (separable or not) tend to produce significantly lower C2ST scores and thus **higher-fidelity images of their class**. This was also found in [19].

	one-class non-separable	one-class separable	multi-channel separable	star-shaped
# epochs	3200 (3200)	543 (3200)	198 (3200)	1700 (3200)
Alp14	0.53 ± 0.3 GAN (0.6 ± 0.3)	1.93 ± 0.27 (1.2 ± 0.2)	3.10 ± 0.6 (2.3 ± 0.5)	0.79 ± 0.33 (0.6 ± 0.3)
Arp3	-	3.01 ± 0.58 (2.4 ± 0.4)	5.97 ± 0.49 (4.2 ± 0.4)	3.01 ± 0.55 (2.1 ± 0.5)
Cki2	-	2.12 ± 0.44 (1.0 ± 0.3)	4.98 ± 0.74 (3.6 ± 0.5)	1.88 ± 0.21 (1.2 ± 0.3)
Mkh1	-	1.30 ± 0.53 (0.5 ± 0.4)	8.18 ± 0.68 (6.6 ± 0.5)	2.99 ± 0.71 (2.4 ± 0.6)
Sid2	-	1.98 ± 0.67 (1.0 ± 0.5)	5.33 ± 0.60 (3.2 ± 0.6)	1.84 ± 0.68 (1.1 ± 0.6)
Tea1	-	1.65 ± 0.49 (0.8 ± 0.5)	5.00 ± 0.66 (2.8 ± 0.5)	1.94 ± 0.73 (1.1 ± 0.4)

Table 4.2.1: Results of C2ST (with the WGAN-GP objective in C2ST's Discriminator) comparing the trained models after their last training epoch. All the models except from the 1st one, were trained with WGAN-GP. The values in parentheses correspond to the reported ones by Osokin et al. in [19], while the dashed to not trained models/classes.

- From the Multichannel models, Star-Shaped Generators perform the best. This was also found in [19]. In addition, their values are close to the published ones even when almost half-trained.

In order to better analyze the above results and compare them more fairly, we re-evaluate all models on images of Alp14 class and at after their 198th epoch of training. We record the values for each of 10 runs needed to calculate the (mean) C2ST and perform the Friedman non-parametric statistical test from [6] in order to decide on the existence of an overall "best" model. The variable of interest is C2ST value (continuous) and the common subjects are the images of the Alp14 class, while the different Generators are the one under comparison. The Friedman test tests the null hypothesis that repeated measurements of the same individuals have the same distribution; in our case it is used to check if the different models trained on the same images produce consistent generations as those are measured via C2ST.

The mean and standard deviation of the C2ST scores for the models evaluated after epoch 198 are given in the Table 4.2.2.

Using the underlying scores from each run and each model test the Null Hypothesis

	one-class non-separable	one-class separable	multi-channel separable	star-shaped
# epochs	198	198	198	198
`Alp14`	0.95 ± 0.35 GAN	2.19 ± 0.37	3.10 ± 0.60	2.41 ± 0.44

Table 4.2.2: Results of C2ST (with the WGAN-GP objective in C2ST's Discriminator) comparing the trained models after their 198th training epoch. All the models except from the 1st one, were trained with WGAN-GP on images of `Alp14` class.

"*All generators perform equally*", we computed the Friedman Test's Chi-Square and P-Value:

- **Chi-Square**: 20.64
- **P-Value**: 0.001

Both of these results, especially the p-value which is much less than the typical significance level of 5% or 2%, indicate that **we should reject the null hypothesis**. This confirms our initially-formed belief that **one-class models are potentially more able to produce images of higher fidelity and achieve lower C2ST scores**. Under the goals of the present work, though, it is more preferable to have well-performing Multichannel (multi-class) trained models in order to use them to circumvent the aforementioned limitations of Fluorescent Microscopy.

Chapter 5

Conclusion - Future Work

All in all, in the context of this very interesting project the ability of Generative Models to capture the correlations between different proteins was studied. And while GANs, a special class of such models, have been widely applied towards generating natural images, the results and conclusions drawn suggest that they could successfully be applied to Biological Image Synthesis.

Modelling the causal dependencies between the green and red-tagged proteins in FM images of fission yeast cells, our models were able to synthesize new ones exhibiting realism but also being able to depict many such channels simultaneously. In addition, we presented an experiment where the temporal evolution of those localizations were visualized as the cell artificially progressed though its life-cycle.

Future extensions of such a work, include training multi-channel models on more polarity factors and using human experts in order to gain more trustworthy insights on the generation abilities of our trained models. In addition, cell-image classifiers could be trained from scratch on relevant datasets, in order to make the existing GAN evaluation metrics more robust.

Bibliography

[1] Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. “Wasserstein GAN”. In: *arXiv e-prints*, arXiv:1701.07875 (Jan. 2017), arXiv:1701.07875. arXiv: 1701.07875 [stat.ML].

[2] Chalfie, M, Tu, Y, Euskirchen, G, Ward, W W, and Prasher, D C. “Green fluorescent protein as a marker for gene expression”. en. In: *Science* 263.5148 (Feb. 1994), pp. 802–805.

[3] Dodgson, James, Chessel, Anatole, Vaggi, Federico, Giordan, Marco, Yamamoto, Miki, Arai, Kunio, Madrid, Marisa, Geymonat, Marco, Abenza, Juan Francisco, Cansado, José, Sato, Masamitsu, Csikasz-Nagy, Attila, and Carazo Salas, Rafael Edgardo. “Reconstructing regulatory pathways by systematically mapping protein localization interdependency networks”. In: *bioRxiv* (2017). DOI: 10.1101/116749. eprint: https://www.biorxiv.org/content/early/2017/03/14/116749.full.pdf. URL: https://www.biorxiv.org/content/early/2017/03/14/116749.

[4] Feoktistova, Anna, Morrell-Falvey, Jennifer, Chen, Jun-Song, Singh, N Sadananda, Balasubramanian, Mohan K, and Gould, Kathleen L. “The fission yeast septation initiation network (SIN) kinase, Sid2, is required for SIN asymmetry and regulates the SIN scaffold, Cdc11”. en. In: *Mol. Biol. Cell* 23.9 (May 2012), pp. 1636–1645.

[5] Fréchet, M. Maurice. “Sur quelques points du calcul fonctionnel”. In: *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22.1 (Dec. 1906), pp. 1–72. ISSN: 0009-725X. DOI: 10.1007/BF03018603. URL: https://doi.org/10.1007/BF03018603.

[6] Friedman, Milton. “A Correction: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance”. In: *Journal of the American*

Statistical Association 34.205 (1939), pp. 109–109. ISSN: 01621459. URL: http://www.jstor.org/stable/2279169 (visited on 06/27/2022).

[7] Gómez, Eliana and Forsburg, Susan. “Analysis of the Fission Yeast Schizosaccharomyces pombe Cell Cycle”. In: *Methods in molecular biology (Clifton, N.J.)* 241 (Feb. 2004), pp. 93–111. DOI: 10.1385/1-59259-646-0:93.

[8] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. “Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1406.2661 (June 2014), arXiv:1406.2661. arXiv: 1406.2661 [stat.ML].

[9] Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. “Improved Training of Wasserstein GANs”. In: *arXiv e-prints*, arXiv:1704.00028 (Mar. 2017), arXiv:1704.00028. arXiv: 1704.00028 [cs.LG].

[10] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *arXiv e-prints*, arXiv:1706.08500 (June 2017), arXiv:1706.08500. arXiv: 1706.08500 [cs.LG].

[11] Karras, Tero, Laine, Samuli, Aittala, Miika, Hellsten, Janne, Lehtinen, Jaakko, and Aila, Timo. “Analyzing and Improving the Image Quality of StyleGAN”. In: *arXiv e-prints*, arXiv:1912.04958 (Dec. 2019), arXiv:1912.04958. arXiv: 1912.04958 [cs.CV].

[12] Kingma, Diederik P and Welling, Max. “Auto-Encoding Variational Bayes”. In: *arXiv e-prints*, arXiv:1312.6114 (Dec. 2013), arXiv:1312.6114. arXiv: 1312.6114 [stat.ML].

[13] Langr, Jakub and Bok, Vladimir. *GANs in action: deep learning with generative adversarial networks*. Manning Publications, 2019.

[14] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[15] Lopez-Paz, David and Oquab, Maxime. “Revisiting Classifier Two-Sample Tests”. In: *arXiv e-prints*, arXiv:1610.06545 (Oct. 2016), arXiv:1610.06545. arXiv: 1610.06545 [stat.ML].

[16] Martin, Sophie G and Arkowitz, Robert A. “Cell polarization in budding and fission yeasts”. en. In: *FEMS Microbiol. Rev.* 38.2 (Mar. 2014), pp. 228–253.

[17] Meijering, Erik, Carpenter, Anne E., Peng, Hanchuan, Hamprecht, Fred A., and Olivo-Marin, Jean-Christophe. “Imagining the future of bioimage analysis”. In: *Nature Biotechnology* 34.12 (Dec. 2016), pp. 1250–1255. ISSN: 1546-1696. DOI: 10.1038/nbt.3722. URL: https://doi.org/10.1038/nbt.3722.

[18] Nash, John F. “Equilibrium points in n-person games”. In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49. DOI: 10.1073/pnas.36.1.48. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.36.1.48. URL: https://www.pnas.org/doi/abs/10.1073/pnas.36.1.48.

[19] Osokin, Anton, Chessel, Anatole, Carazo Salas, Rafael E., and Vaggi, Federico. “GANs for Biological Image Synthesis”. In: *arXiv e-prints*, arXiv:1708.04692 (Aug. 2017), arXiv:1708.04692. arXiv: 1708.04692 [cs.CV].

[20] Pollard, Thomas D and Wu, Jian-Qiu. “Understanding cytokinesis: lessons from fission yeast”. en. In: *Nat. Rev. Mol. Cell Biol.* 11.2 (Feb. 2010), pp. 149–155.

[21] Radford, Alec, Metz, Luke, and Chintala, Soumith. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1511.06434 (Nov. 2015), arXiv:1511.06434. arXiv: 1511.06434 [cs.LG].

[22] Sajjadi, Mehdi S. M., Bachem, Olivier, Lucic, Mario, Bousquet, Olivier, and Gelly, Sylvain. “Assessing Generative Models via Precision and Recall”. In: *arXiv e-prints*, arXiv:1806.00035 (May 2018), arXiv:1806.00035. arXiv: 1806.00035 [stat.ML].

[23] Shaner, Nathan C., Campbell, Robert E., Steinbach, Paul A., Giepmans, Ben N. G., Palmer, Amy E., and Tsien, Roger Y. “Improved monomeric red, orange and yellow fluorescent proteins derived from Discosoma sp. red fluorescent protein”. In: *Nature Biotechnology* 22.12 (Dec. 2004), pp. 1567–1572. ISSN: 1546-1696. DOI: 10.1038/nbt1037. URL: https://doi.org/10.1038/nbt1037.

[24] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. “Going Deeper with Convolutions”. In: *arXiv e-prints*, arXiv:1409.4842 (Sept. 2014), arXiv:1409.4842. arXiv: 1409.4842 [cs.CV].

[25] Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jonathon, and Wojna, Zbigniew. "Rethinking the Inception Architecture for Computer Vision". In: *arXiv e-prints*, arXiv:1512.00567 (Dec. 2015), arXiv:1512.00567. arXiv: 1512.00567 [cs.CV].

[26] Tsien, Roger Y. "THE GREEN FLUORESCENT PROTEIN". In: *Annual Review of Biochemistry* 67.1 (1998). PMID: 9759496, pp. 509–544. DOI: 10.1146/annurev.biochem.67.1.509. eprint: https://doi.org/10.1146/annurev.biochem.67.1.509. URL: https://doi.org/10.1146/annurev.biochem.67.1.509.

[27] Weng, Lilian. "Flow-based Deep Generative Models". In: *lilianweng.github.io/lil-log* (2018). URL: http://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html.

[28] Yeh, Raymond A., Chen, Chen, Yian Lim, Teck, Schwing, Alexander G., Hasegawa-Johnson, Mark, and Do, Minh N. "Semantic Image Inpainting with Deep Generative Models". In: *arXiv e-prints*, arXiv:1607.07539 (July 2016), arXiv:1607.07539. arXiv: 1607.07539 [cs.CV].

[29] Zhou, Sharon, Zelikman, Eric, and Zhou, Eda. *Generative Adversarial Networks (GANs) Specialization*. Feb. 2021. URL: https://www.deeplearning.ai/program/generative-adversarial-networks-gans-specialization/.

Appendix - Contents

Appendix A

First Appendix

A.1 Training of One-class Non-separable GAN

The training was really unstable especially if the choice of criterion was a saturating one, such Binary Cross-Entropy between the Discriminator network's prediction and the target values (1 for real, 0 for generated images). We ended up using Wasserstein loss and Gradient Penalty to make things work (did not manage to find how the authors managed to stabilize training with BCE loss). Below, the training curves are given during the 3200 training epochs.

As can be seen, the training was really unstable at the beginning but eventually the loss did its trick resulting in a smooth convergence for the last 2.5K epochs.

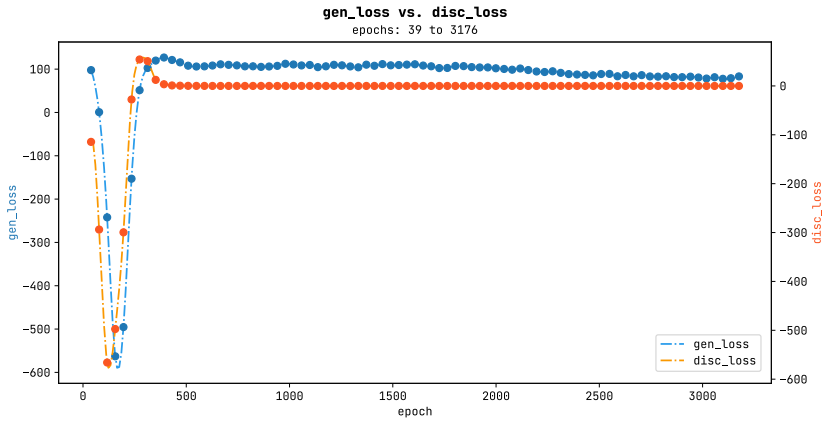


Figure A.1.1: Training curves for One-class Non-separable GAN model trained on images of `Bgs4` (red) and `Alp14` (green) yeast cell proteins. The loss used was Wasserstein loss + Gradient Penalty to encourage Lipschitz-1 continuity on the Discriminator.

A.1.1 GAN Evaluation Metrics Evolution

To evaluate the fidelity and diversity of images produced by a generative model (in an unpaired setting such as here), the most widely used metrics are FID and a modified version of F1-score. Below the evolution of these curves are presented:

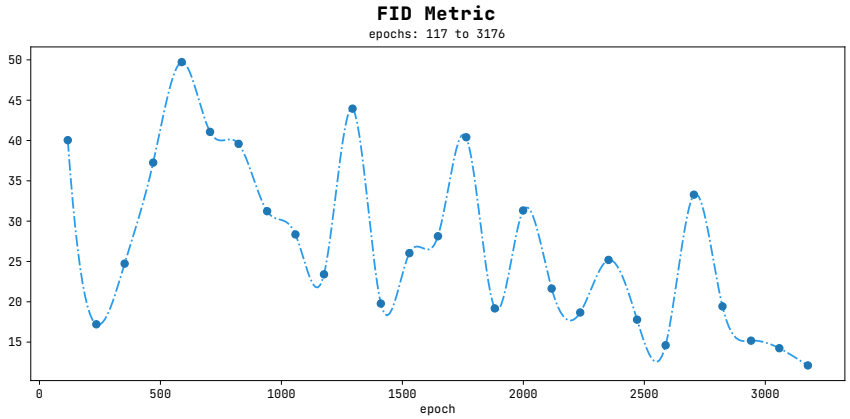


Figure A.1.2: Evolution of FID evaluation metric during training of One-class Non-Separable GAN model.

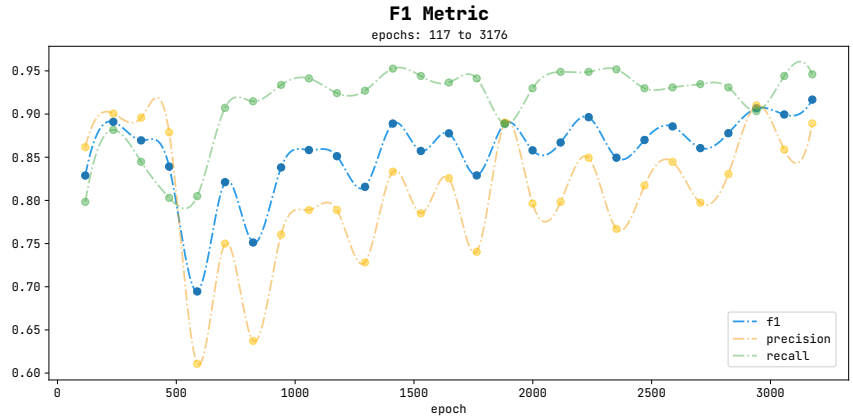


Figure A.1.3: Evolution of F1-Score evaluation metric during training of One-class Non-Separable GAN model.

A.2 Training of One-class Separable WGAN-GP

To keep the report uncluttered we skip providing training plots and just mention that the overall the training was much more stable than training each model individually, especially in the version where the criterion was the Wasserstein loss + Gradient Penalty.The models were trained for a total of 543 epochs (3200 epochs were used in [19]) due to computational resources limitations. However, as also mentioned in

the paper, the models converge well before the 1000th epoch and subsequent ones are only used to squeeze every last bit of performance out of the models.

A.2.1 GAN Evaluation Metrics Evolution

Below the evolution of these curves are presented:

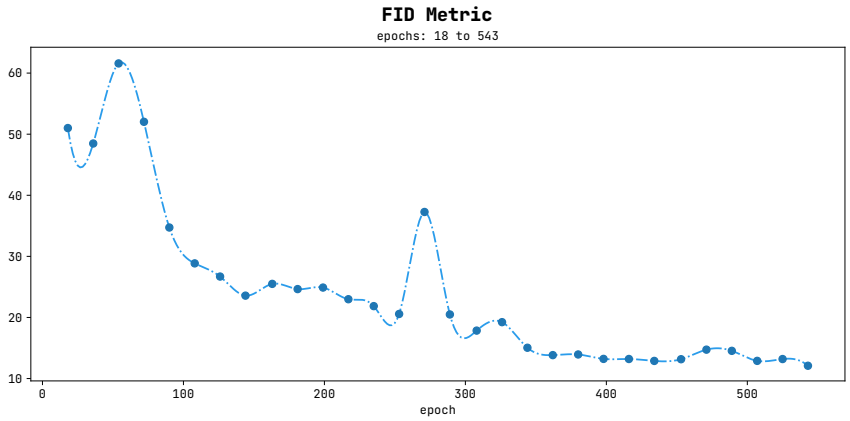


Figure A.2.1: Evolution of FID evaluation metric during training of One-class Separable WGAN-GP model.

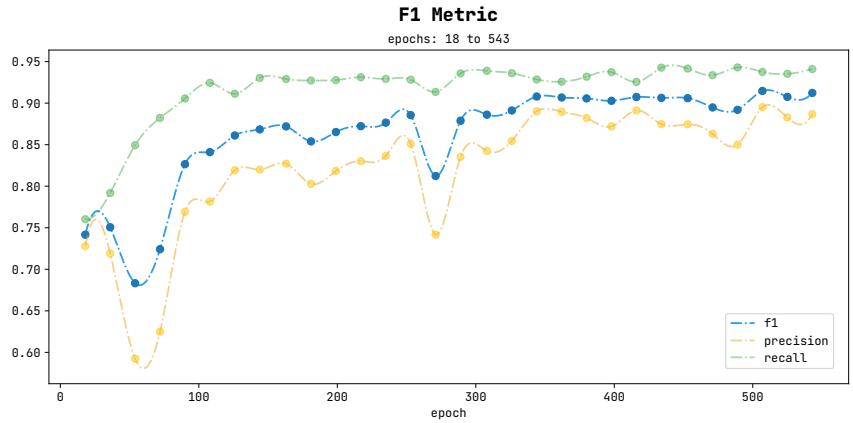


Figure A.2.2: Evolution of F1-Score evaluation metric during training of One-class Separable WGAN-GP model.

A.3 Training of Multichannel Separable WGAN-GP

To keep the report uncluttered I also skip providing here training plots and just mention that the overall the training was even more stable than the training six 2-channel GAN models and of course vastly more stable than training each individual non-separable models, especially in the version where the criterion was the Wasserstein loss + Gradient Penalty. This multichannel model were trained for

a total of 200 epochs (3200 epochs were used in [19]) due to computational resources limitations. However, as also mentioned in the paper, the models converge well before the 1000th epoch and subsequent ones are only used to squeeze every last bit of performance out of them.

A.3.1 GAN Evaluation Metrics Evolution

Below the evolution of these curves are presented:

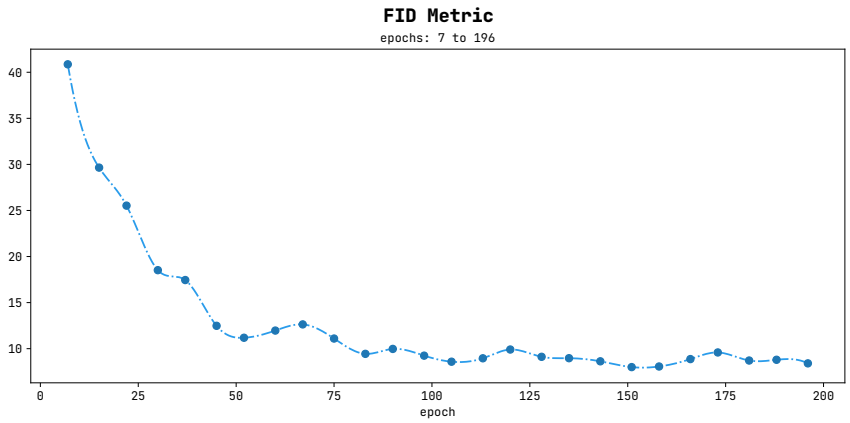


Figure A.3.1: Evolution of FID evaluation metric during training of Multichannel Separable WGAN-GP model.

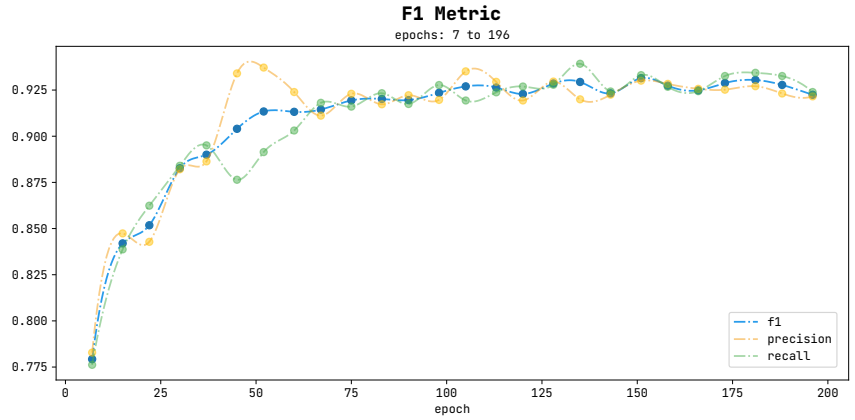


Figure A.3.2: Evolution of F1-Score evaluation metric during training of Multichannel Separable WGAN-GP model.

A.4 Training of Star-Shaped Separable WGAN-GP

The overall training behaviour was smoother and the models seem to converge faster and lower values of the evaluation metrics. The Star-Shaped model was trained for a

total of 1700 epochs (3200 epochs were used in [19]) due to computational resources limitations.

A.4.1 GAN Evaluation Metrics Evolution

Below the evolution of these curves are presented:

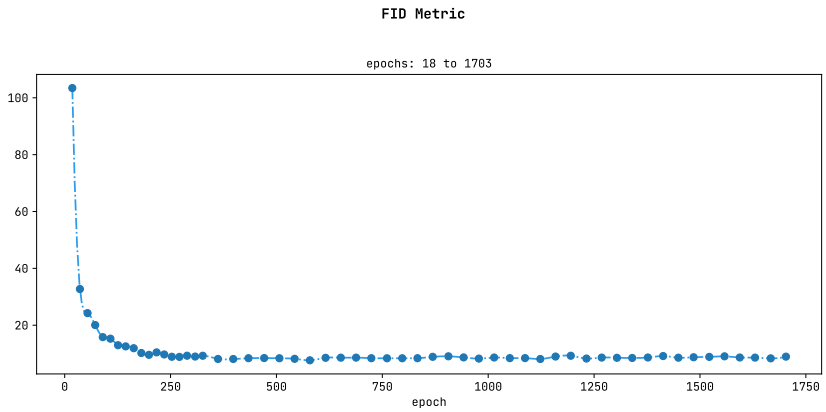


Figure A.4.1: Evolution of FID evaluation metric during training of Star-Shaped Separable WGAN-GP model.

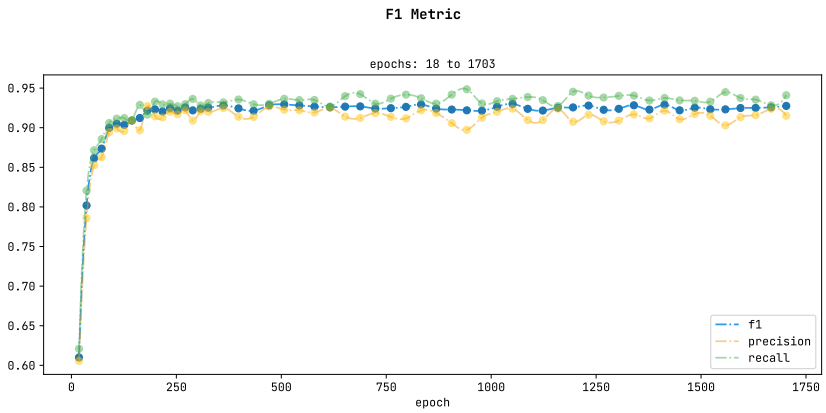


Figure A.4.2: Evolution of F1-Score evaluation metric during training of Star-Shaped Separable WGAN-GP model.