

UNIVERSITY OF BIRMINGHAM

QUANTIFYING THE UNCERTAINTY AND GENERALISABILITY OF MACHINE LEARNING-BASED VIRTUAL STAINING FOR HIGH-THROUGHPUT SCREENING

by

SAMUEL MARK TONKS

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Krull Lab

School of Computer Science

University of Birmingham

Jan 2025

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

© Copyright by SAMUEL MARK TONKS, 2025

All Rights Reserved

Abstract

High-throughput screening (HTS) [1] with fluorescence microscopy [2] enables different biological structures within cells to be simultaneously revealed, at scale, so that drug compound effects can be quantified. Fluorescence microscopy, however, is expensive, time-consuming [3] and can negatively impact cell samples [4]. While label-free microscopy [5, 6], is cost-effective and widely available, the biological information is not as easily accessible compared to when using fluorescence microscopy because of a lack of intrinsic optical contrast between cellular components. Virtual staining via image-to-image translation [4, 7–12] – a machine learning (ML) technique – aims to translate unstained label-free microscopy images into multiple fluorescent microscopy images to obtain the benefits of fluorescence staining without the associated costs.

Despite impressive qualitative results [4, 7–10], little is known about the feasibility of virtual staining with HTS due to the lack of quantitative, biologically meaningful evaluation metrics. HTS data is continuously being generated across different laboratory settings and new compound perturbations [1]. Despite this, existing virtual staining works [4, 7–12] have been trained and tested on images sampled from the same setting-specific training data distribution. What they have not done is evaluate the generalisability of virtual staining to never-before-seen data, sampled from a different laboratory or cell type. In light of this, we can not know whether virtual staining models trained on setting specific data can generalise to other experimental settings putting the scalability of virtual staining with HTS in doubt.

Virtual staining is an ill-posed problem; given access to a bright-field image, it is very challenging to define a single virtual staining solution. This uncertainty is expressed

through the posterior distributions of possible virtual staining solutions and instance biological features given a bright-field image. Currently, very little understanding exists of how well virtual staining models can describe these posterior distributions. Furthermore, the challenge of comparing different predicted instance posterior distributions to a single target sample value from the true posterior distribution remains under-explored.

In the first part of this thesis, we introduce the motivations and core questions this work will address. We formally explore the use of HTS with conventional fluorescence microscopy and different virtual staining approaches. We then provide a background of the mathematical basis of some of the core approaches to virtual staining explored within this work and how they are connected. We provide a comprehensive review of the literature around different adaptations of the core approaches to virtual staining, approaches to evaluating virtual staining and generalisation within virtual staining.

In Chapter 4 we benchmark different virtual staining approaches and present the results of a newly developed evaluation pipeline that provides biologically meaningful results.

In Chapter 5 we further utilise this pipeline and provide an extensive analysis of the generalisation performance of virtual staining models for three common generalisation tasks within HTS.

In Chapter 6 we explore the predicted instance posterior distributions of two generative modelling approaches and present a principled method for directly predicting these distributions from the bright-field image. We systematically compare the predicted instance posterior distributions of both generative modelling approaches using three newly proposed evaluation criteria and compare our findings to population metrics.

Acknowledgments

I would like to thank the many people who helped through discussions, comments, debugging of code while conducting experiments, writing of various papers and this thesis. A special thank you to EPSRC for funding this research, the Alan Turing Institute and to my academic supervisors, Alexander Krull, Iain Styles and Ales Leonardis. I would like to thank GSK for sponsoring my PhD. In particular, I would like to thank my industrial supervisors, Minh Doan and Steve Hood, as well as colleagues whom I worked closely with at GSK, Cuong Nguyen, Chih-Yang Hsu, Ryan Musso, Ceridwen Hopely and Steve Titus. Finally, I would like to thank my partner, family and friends who have supported me throughout this journey.

Contents

Nomenclature **ix**

1 Introduction **1**

1.1 Conventional fluorescence staining with HTS 6

1.2 Virtual staining with HTS 6

1.2.1 Regression-based 7

1.2.2 Posterior-based 8

1.3 Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels? 9

1.4 Can virtual staining models for HTS generalise? 10

1.5 Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell? 12

1.6 GSK HTS Dataset 13

2 Background **15**

2.1 Conditional Variational Autoencoders (cVAEs) 16

2.2 Conditional Denoising Diffusion Probabilistic Model (cDDPM) 21

2.3 Conditional Generative Adversarial Networks (cGANs) 28

2.4 Practical implications of approaches 30

2.5 Conclusion 31

3 Literature Review **33**

3.1 Approaches to image-to-image translation 34

3.1.1 Pix2pix GANs . . . 37
3.1.2 Diffusion models . . . 39
3.1.3 Latent diffusion models . . . 41
3.1.4 Conclusion . . . 42
3.2 Evaluation of image-to-image translation approaches . . . 44
3.3 Generalization of image-to-image translation approaches . . . 49

4 Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels? 51
4.1 cGANs outperform regression-based approaches . . . 52
4.2 Virtually stained nuclei and cytoplasm images often consistently and faithfully reproduce the information found in fluorescence microscopy, virtually stained images of DNA-damage are usually less accurate . . . 58
4.2.1 Level 1 - Visual Inspection & Pixel/Patch-level Performance . . . 60
4.2.2 Level 2 - Biological Feature Representation . . . 62
4.2.3 Level 3 - RZ' Factor . . . 65
4.2.4 Discussion . . . 66

5 Can virtual staining for high-throughput screening generalise? 69
5.1 Training on non-toxic vs toxic samples improves virtual staining performance in particular for ovarian cell lines . . . 75
5.2 Generalising to new cell types is complex . . . 79
5.3 Generalisation to new cell types and phenotypes shows good generalisation performance . . . 83
5.4 Discussion . . . 84

6 Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell? 87
6.1 Bright2Bio-Net (B2B-Net) - a direct method for predicting the posterior distribution of instance-wise biological features . . . 92

6.2 How can we evaluate a predicted posterior with only a single ground truth sample from the true posterior distribution? 95
6.2.1 Experiments & Results 98
6.2.2 Can the log-likelihood, standard score & percentile rank computed over the predicted posterior distributions reveal new insights compared to population metrics? 100
6.3 Discussion 110

7 Conclusion **113**
7.0.1 Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels? . . 114
7.0.2 Can virtual staining models for HTS generalise? 115
7.0.3 Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell? 116
7.1 Future work 117
7.1.1 Further development and evaluation of Bright2Bio-Net 117
7.1.2 Further evaluation of proposed measurements in Chapter 6 section 6.2 using a simulated posterior distribution 117

References **119**

Appendix A: Additional data for Chapter 4 **133**
Evaluation of Cellpose masks for use in virtual staining evaluation pipeline . . 134

Appendix B: Additional data for Chapter 5 **139**

Appendix C: Additional data for Chapter 6 **147**

Nomenclature

Roman Symbols

$\hat{\mathbf{y}}_i$	Virtual stain prediction for bright-field $\mathbf{x}_i$
$\hat{\mathbf{y}}_i^l$	Virtual stain sample from $P_\theta(y_{i,j} \mid \mathbf{x}_{i,j})$
$\hat{\mathcal{Y}}_{i,j}$	Predicted feature value for bright-field $\mathbf{x}_i$ instance or nucleus j
$\mathbf{x}_i$	Bright-field image
$\mathbf{x}_{i,j}$	Bright-field nucleus or instance in image i
$\mathbf{y}_i$	Fluorescence image
$\mathbf{y}_{i,j}$	Fluorescence image containing only pixels for nucleus j with all other pixels masked
$\mathcal{Y}_{i,j}$	Nucleus feature value from nuclei fluorescence stain
$\tilde{x}_{i,j}$	Simulated bright-field nucleus or instance in image i
D_θ^x	Discriminator network for domain X
D_θ^y	Discriminator network for domain Y
g_θ	Regression-based function
G_θ^x	Generator network for domain X
G_θ^y	Generator network for domain Y

$P(\mathbf{y}_{i,j} \mid \mathbf{x}_{i,j})$ True posterior distribution for nucleus j in image i given bright-field image i nucleus j

$P(y_{i,j} \mid \mathbf{x}_{i,j})$ True biological feature posterior distribution for nucleus j in image i given bright-field image i nucleus j

$P_\theta(\mathbf{y}_{i,j} \mid \mathbf{x}_{i,j})$ Predicted virtual staining posterior distribution for nucleus j in image i given bright-field image i nucleus j

$P_\theta(y_{i,j} \mid \mathbf{x}_{i,j})$ Predicted biological feature posterior distribution for nucleus j in image i given bright-field image i nucleus j

S_θ^x Style Encoder network for domain X

S_θ^y Style Encoder network for domain Y

X Bright-field domain for image-to-image translation

Y Fluorescence domain for image-to-image translation

z_c Vector quantized codebook entry

L Total samples from posterior distribution

Subscripts

c Codebook entry

i Image

j Nucleus

k Specific mixture of Gaussian model component

Superscripts

l Sample

x Domain X

y Domain Y

Other Symbols

$\mathcal{N}$ The Gaussian distribution

$\mathcal{U}$ The Uniform distribution

Acronyms

AIC Akaike information criterion

B2B-Net Bright2Bio-Net

BIC Bayes information criterion

cDDPM Conditional denoising diffusion probabilistic model

CDF Cumulative distribution function

cGAN Conditional generative adversarial network

CGMs Conditional Generative Models

CNN Convolutional neural network

cVAE Conditional Variational Autoencoder

Cy5 Cyanine

DAPI 6-diamidino-2-phenylindole

DDIM Denoising diffusion implicit sampling

DMSO Dimethylsulfoxide

ELBO Evidence lower bound

FID Frechét inception distance

FITC Fluorescein

GAN Generative adversarial network

GSK Glaxosmithkline

HCI High content imaging

HTS High-throughput screening

I2I Image-to-image translation

IS Inception score

KLD Kullback-Leibler divergence

ML Machine learning

MSE Mean squared error

N-MAE Normalised mean absolute error

NLL Negative average log-likelihood

PDF Probability density function

pix2pix Pix2pix GAN

pix2pixHD Pix2pixHD GAN

PSNR Peak signal-to-noise ratio

RZ' RZ prime

SSIM Structural similarity index measure

VAE Variational Autoencoder

Wdist Wasserstein Distance

Chapter 1

Introduction

High-throughput screening (HTS) plays a crucial role in drug discovery by enabling the simultaneous testing of a large number of candidate drug compounds to assess their effects on cell cultures [1]. In recent years, high-content screening has leveraged advances in automation and imaging technology, enabling rich data capture of cellular phenotypes induced by compound perturbations at a high-throughput scale.

A widely used technique in HTS is multi-channel fluorescence microscopy [2]. Using different biomolecule-specific staining dyes enables biological structures within cells, such as the nuclei, cytoplasm, or DNA-damage, to be simultaneously imaged using fluorescent dyes with different optical spectra, with each dye captured in a separate image channel. As shown within a traditional HTS pipeline in Figure 1.1a the fluorescence channels can in turn be used to quantify specific biological features of individual cells, such as morphological, textural and intensity properties that enable the comparison of drug effects across a large number of compound perturbations. These biological features are the primary tool by which drug effect is assessed. Depending on the biomarker, fluorescence staining of a single plate can take around 30 minutes to complete [13].

Conventional fluorescence microscopy, however, has its limitations. The staining protocol typically requires the cells to be fixed and permeabilised - a process in which cells are preserved in their biological state, effectively frozen in time - thus limiting the application of this technique to single time-point studies. Furthermore, once the fluorescence staining has been completed, the cells can no longer be used. Multiple

time point studies are often required. In practice, this results in additional sets of HTS plates, one for each time point, to be fixed and stained ultimately leading to the imaging of different plates at different time points and a bottleneck in the process caused by fluorescence staining as shown in Figure 1.1a.

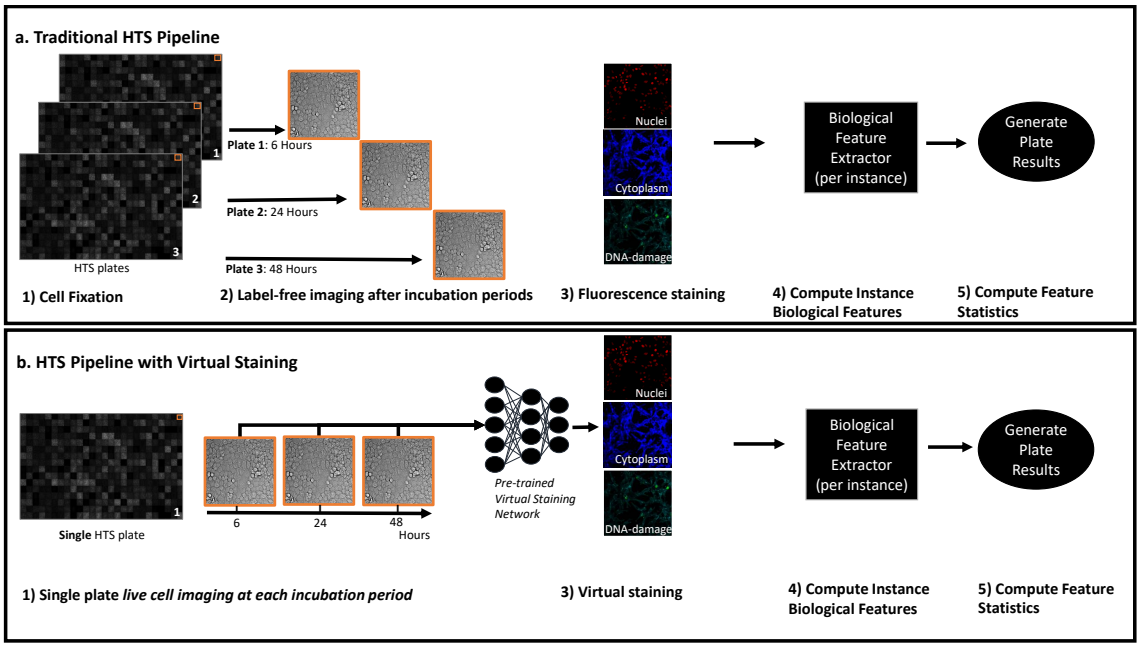


Figure 1.1: **Traditional high-throughput screening (HTS) pipeline vs. HTS pipeline with virtual staining.** A traditional HTS pipeline starts with cell fixation, where cells are treated with chemical agents to preserve their structure and cellular components. Results are extracted after N incubation periods, in this case, 3 at 6, 24, and 48 hours. After applying fluorescence staining and the downstream extraction of instance biological features, such as morphological, textural and intensity properties, the plate can no longer be used. This leads to the requirement for multiple plates at different time points. In contrast, HTS with virtual staining requires only a *single* plate, which can be imaged label-free at 6, 24, and 48 hours because the pre-trained virtual staining network takes only the bright-field as input and generates the virtual fluorescence channels. This single plate can then image *live cells* at multiple incubation periods. To virtually stain an entire plate via a pre-trained virtual staining network can take minutes, while fluorescence staining can take hours and, in some cases, days.

This can produce what is known as plate effects [3] in which the plates are exposed to different systematic biases such as those produced by incubation effects or defects in manufacturing.

Additionally, the fixation and staining process is expensive, requires specialist machinery, and can introduce significant phototoxicity and photobleaching that can perturb samples [4]. Furthermore, the number of fluorescence stains is inherently limited by

spectrum saturation.

Following these challenges, significant interest has been put into label-free microscopy, wherein images of cells are captured without fluorescence staining [4]. Label-free microscopy [5, 6], is cost-effective and scalable, but unfortunately, the biological information is not as easily accessible compared to fluorescence staining [14] because of the lack of contrast between the background and the biological structures of interest. This limits its use in extracting biological features and determining drug effects on cell cultures.

We hypothesise whether it is possible to extract the same biological information from label-free microscopy images as from fluorescence microscopy images. We believe there are certain forms of biological information (found in specific fluorescence channels) that can be predicted from the bright-field and others that can't. By determining this, we can remove the fluorescence image channels that can be extracted from the bright-field domain. This would free up channel space for new multiplex imaging combinations. For those fluorescence channels where the biological information can be extracted from the bright-field we can use a single plate to produce multi-time point studies and perform live-cell imaging as shown in Figure 1.1b.

An approach to designing such a method is to explore image-to-image translation (I2I) [7, 15], a machine learning (ML) technique – to translate label-free bright-field microscopy images into multiple fluorescent microscopy channels, each corresponding to a structure of biological interest. We hypothesise, such a method, could generate virtually stained fluorescence channels from low-cost and more easily accessible bright-field images. In contrast to a traditional HTS pipeline, a HTS pipeline with virtual staining via I2I, as shown in Figure 1.1, reduces the number of HTS plates from the number of incubation periods (3) to 1 because, by substituting virtual staining for fluorescence staining, the process does not introduce any fluorescence into the samples as the virtual staining model requires the label-free image as input which means we can perform multi-time point studies. This removes the requirement of multiple plates and the negative plate effects of fluorescence microscopy as previously mentioned, whilst also enabling live-cell imaging [16]. Furthermore, once trained, the virtual staining model can generate virtual

staining outputs for a single plate in seconds compared to fluorescence microscopy, which takes 30 minutes. As previously mentioned, it's these biological features that are the primary tool by which drug effect is assessed. It is therefore essential that these features are accurately reproduced in any kind of HTS with virtual staining pipelines such as the one shown in Figure 1.1b.

Virtual staining using I2I has been widely explored for multiple applications [4, 9–12] but not HTS. Despite the continued development of virtual staining, several obstacles must to be overcome before considering the routine application of virtual staining within HTS for drug discovery. In the following, we will identify three essential problems that will be investigated in this thesis.

Firstly, virtual staining produces impressive qualitative results [4, 9–12] however, it is critical to evaluate these results quantitatively. Early approaches [4, 10] to virtual staining evaluation have focused on individual pixel/patch-based measurements such as; Pearson correlation coefficient, the peak signal-to-noise ratio (PSNR) [17] and the structural similarity index measure (SSIM) [18]. These metrics are not explicitly designed to capture the full range of biological variation. Furthermore, there is a general lack of literature that focuses on how to quantitatively evaluate virtual staining models within the aims and constraints of HTS. As a consequence, we can not know the extent to which an HTS pipeline with virtual staining will produce biological results that correspond to those obtained when using a traditional HTS pipeline. Therefore, it's critical to ask **Can virtual staining accurately represent biological information within individual cells found in the fluorescence channels?**

Secondly, more recent virtual staining works [5, 11, 12, 19] have typically been evaluated on images sampled from the *same distribution* of images the model was trained on. Yet, in reality, HTS imaging data is highly diverse, generated across different settings that include imaging systems, magnifications, experiments, cell types, and phenotypes. The naive approach of training multiple setting-specific models allows us to understand the performance of virtual staining in a controlled data setting. However, training virtual staining models on specific subsets of HTS data in industrial applications is not scalable

because to generate the data to train the virtual staining model you would need to first run the traditional HTS pipeline with fluorescence staining, which removes the need for virtual staining. If virtual staining is to be considered as a viable imaging alternative to fluorescence microscopy within HTS pipelines then virtual staining models must be able to accurately represent the biological information found within individual cells of other settings. In other words, it is important to ask the question, **can virtual staining for HTS generalise?**

Finally, virtual staining is an ill-posed problem, meaning that for a given bright-field image we can not know what the corresponding fluorescence image looks like, without observing it directly. In other words, there is a posterior distribution of virtual staining solutions given a bright-field image. This also suggests that there is a posterior distribution of instance biological feature values that could be obtained given a label-free bright-field image. Looking at the instances within the bright-field images as shown in Figure 1.1, how much can we truly know about the biological feature posterior distributions? It is therefore important to ask **can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell?**

The purpose of this thesis is to address the three questions outlined above while using data and experimental guidance provided by GSK. Within Chapter 4, we benchmark different approaches to virtual staining and propose a framework for evaluating virtual staining for HTS that goes beyond pixel/patch-based measurements and quantifies the existence of meaningful biological features. In Chapter 5, we further improve this evaluation pipeline and conduct a large-scale benchmarking exercise to explore whether trained virtual staining models can generalise to unseen data from different data distributions commonly found within HTS. The experimental design of Chapter 5 was very much driven by GSK. Finally, in Chapter 6, we explore the posterior distributions of instance biological features and compare them to the population distributions over the entire test set. We present the design of a new direct approach to predicting the instance biological feature posterior distributions and present three novel quantitative approaches to evaluating these distributions with access to only a single ground truth sample.

In the remainder of this Chapter, before outlining our approach to addressing these questions in sections 1.3, 1.4, 1.5 we formally introduce the conventional use of fluorescence staining with HTS (1.1) and different approaches to virtual staining via image-image translation for HTS.

1.1 Conventional fluorescence staining with HTS

In conventional HTS, large numbers of paired label-free bright-field and fluorescence images $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$ are captured for different compounds to detect their effects on cell cultures. Here, $\mathbf{x}_i$ and $\mathbf{y}_i$ correspond to the bright-field and a single fluorescence channel image, extracted from the multi-channel fluorescence image. In practice, multiple corresponding single fluorescence channel images are created, with each channel capturing a specific biological structure of interest. In practice, each of these images contains many cells j also referred to as instances. Traditionally, as shown in Figure 6.1, images are processed using an instance segmentation pipeline separating individual cells and derive masked images $\mathbf{y}_{i,j}$ containing only pixel information for the respective cell j.

A feature extraction algorithm [20, 21], which can be thought of as a function $f(\mathbf{y}_{i,j})$ is then applied to each masked image to compute feature values

$$\mathcal{Y}_{i,j} = f(\mathbf{y}_{i,j}) \tag{1.1}$$

for the respective cell/instance. These features can be structural (e.g. the size of the nucleus) or textural (e.g. the number of spots within the nucleus). Finally, the extracted biological features can be aggregated and statistically analysed to measure possible drug effects.

1.2 Virtual staining with HTS

This section first introduces how the virtual staining task is approached when viewed as a regression-based problem and highlights the limitations of this approach. We then

introduce a posterior-based approach to the problem and the use of conditional generative models. We explain how these methods overcome the limits of regression-based approaches and how they are used within this thesis.

We can approach the task of virtual staining via I2I with access to a finite set of independent and identically distributed (i.i.d) training sample pairs $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$ from the bright-field domain X and fluorescence domain Y conventionally generated as part of a traditional HTS pipeline outlined in the previous section 1.1. Virtual staining can be understood as a mapping from the input vector space X to a target vector space Y where these spaces represent different representations of images of biological samples.

1.2.1 Regression-based

Early virtual staining works [4, 10] formulated the task as a regression problem. Regression-based approaches implement a function $g_\theta(\mathbf{x}) = \hat{\mathbf{y}} \approx \mathbf{y}$ with parameters θ via a neural network such as a convolutional neural network (CNN), that maps bright-field images to fluorescence images, $g_\theta : X \to Y$.

To train the network, often a quadratic loss function

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{y}_i - g_\theta(\mathbf{x}_i) \right\|_2^2, \tag{1.2}$$

corresponding to the mean squared error over all pixels, averaged over the training samples N is minimised. The obvious solution to Equation 1.2 is that $\mathbf{y}_i = g_\theta(\mathbf{x}_i)$. However, virtual staining is an ill-posed problem. No such function can exist because different $\mathbf{y}_i$ could potentially occur in the training data together with the same $\mathbf{x}_i$.

In fact, there is a posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ of possible fluorescence images y_i that could occur with a bright-field image x_i. As a consequence, training a network using Equation 1.2 will in the limit lead to a compromise solution $g_\theta(\mathbf{x}_i) = \mathbb{E}_{P(\mathbf{y}_i|\mathbf{x}_i)}[\mathbf{y}_i]$ [22]. This is demonstrated in Chapter 4, where we benchmark a regression-based approach [4] against three probabilistic conditional generative models [7, 23, 24].

You could use these virtual staining predictions to extract biological feature values as

outlined in section 1.1 via Equation 1.1. However, as each prediction is the mean prediction of all possible virtual staining solutions, not genuine samples, the predictions will very often be blurry due to the uncertainty in the mean prediction. Computing biological features from blurry results would produce inaccurate biological feature results. Figure 1.2 compares these two approaches and shows the increase blurriness in the regression-based prediction compared to samples from a posterior-based approach.

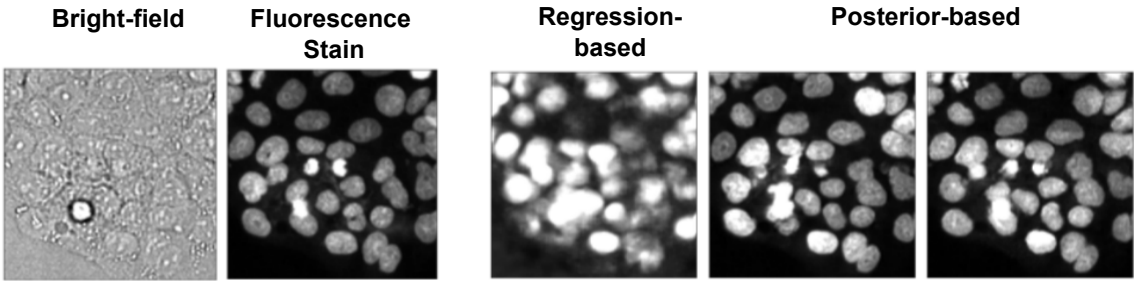


Figure 1.2: **Regression-based vs. Posterior Based (cGANs) Virtual Staining**. Bright-field, DAPI fluorescence image, a single regression-based prediction and two posterior-based method samples. We can observe the blurriness in the regression-based prediction caused by the uncertainty of the mean prediction. In comparison, the posterior-based samples appear much sharper.

1.2.2 Posterior-based

Instead of learning a function $g_\theta(\mathbf{x}_i)$, we are trying to approximate the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ using a conditional generative model[1]. There are many ways to implement conditional generative models. Three foundational methods include Variational Autoencoders (VAEs) [25, 26], Diffusion-based methods [27–29] and Conditional Generative Adversarial Networks (cGANs) [7] which we discuss in more detail in Chapter 2.

More formally, a conditional generative model takes the bright-field image $\mathbf{x}_i$ as input and allows for the drawing of L samples, $\hat{\mathbf{y}}_i^l$ indexed by l, from $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ the approximated posterior distribution.

For each sample $\hat{\mathbf{y}}_i^l$ we can then compute the biological feature value

$$\hat{\boldsymbol{\mathcal{y}}}_{i,j}^l = f(\hat{\mathbf{y}}_{i,j}^l) \tag{1.3}$$

[1]Note that some approaches like GANs or diffusion models do not explicitly compute the probability density but only learn to sample from it.

for each cell j. This is the same as in Equation 1.1 except we are no longer computing the biological feature values on mean predictions but on samples from an approximated posterior distribution.

When we extract multiple (in this case L) samples $\hat{\mathbf{y}}_i^l \sim P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ this provides us with the approximated posterior distribution $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ over the biological feature for each cell. One can also view the true biological feature value $\mathcal{Y}_{i,j}$ for individual cell j as being a sample from the true biological feature posterior distribution $P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$.

One challenge that remains is with access to only a single sample $\mathcal{Y}_{i,j} \sim P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ how can we evaluate the approximated posterior distribution $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$? In Chapter 6 we explore the per-cell biological feature posterior distributions of different conditional generative models and how to evaluate them given only $\mathcal{Y}_{i,j}$.

Alternatively, we can draw a single image sample $\hat{\mathbf{y}}_i$ from $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ and compare it to $\mathbf{y}_i$ and compute the biological feature value $\hat{\mathcal{Y}}_{i,j} = (\hat{\mathbf{y}}_{i,j})$ of each cell and compare it to $\mathcal{Y}_{i,j} = f(\mathbf{y}_{i,j})$ which we explore in Chapters 4 and 5.

1.3 Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels?

Traditional quantitative image quality metrics have proven [4, 10] insufficient for comprehensively capturing the nuanced capabilities of virtual staining. Early approaches to the evaluation of virtual staining [4, 10], and other conditional generative modelling domains like natural imagery [7, 24], relied primarily on pixel/patch-based measurements such as; Pearson correlation coefficient and PSNR [17]. However, these metrics have considerable shortcomings.

Firstly, they treat each pixel independently, meaning the measurements can not quantify any dependencies among neighbouring pixels that could be connected in some form. Secondly, the Pearson correlation coefficient fails to penalise substantial differences in

pixel intensity scales, potentially suggesting a strong correlation where meaningful visual similarity is lacking. Thirdly, these metrics can be positively skewed when averaging over an entire image by uniform background performance, rendering them unreliable indicators of virtual staining effectiveness. Finally, they do not directly quantify the relative effectiveness of virtual staining as a replacement for fluorescence staining within an HTS pipeline.

Developing robust methods to assess virtual staining performance remains a critical challenge within HTS. In Chapter 4 we propose a framework for evaluating virtual staining for HTS composed of three levels: pixel/patch-based, biological feature-based, and drug effect. Using this framework, we systematically explore if virtual staining model predictions are comparable in pixel space. Are the biological features of virtual staining predictions comparable to those from the corresponding fluorescence image? Are the drug effect results from virtual staining the same compared to when using fluorescence microscopy?

For a concise summary of our findings for this question, refer to section 7.0.1

1.4 Can virtual staining models for HTS generalise?

Typically [5, 11, 12, 19] virtual staining models are trained and tested on images sampled from the *same distribution*. In industrial applications, as part of ongoing scientific research, HTS data is continuously generated across a broad range of experimental settings and compound perturbations. Training virtual staining models on setting-specific HTS data for a particular experiment is simply not scalable. However, very little is known about how virtual staining models perform on images from different HTS data distributions compared to the HTS data distribution the model was trained on. The differences between HTS training and HTS test data distributions we refer to as distribution shifts. Example distribution shifts include unseen compounds, cell types or phenotypes. To address this gap we ask whether we can use the diversity of HTS data outlined in section 1.6 to train virtual staining models that can generalise to different distribution shifts.

In Chapter 5 we systematically investigate virtual staining performance across three

generalisation tasks: generalising to new cell types, generalising to new phenotypes and generalising to both new phenotypes and new cell types. We further develop the evaluation pipeline developed in Chapter 4 by extracting biological feature for individual instances (cells) instead of an average result across all instances per image. We introduce semantic segmentation performance measurements which further disentangle instance performance based on precision and recall. We empirically identify data generation practices to produce highly generalisable models for the three explored generalisation tasks and reveal insights into virtual staining performance using the revised evaluation pipeline.

For a concise summary of our findings for this question, refer to section 7.0.2

1.5 Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell?

Despite improvements in the evaluation of instance-level performance and the insights identified in Chapters 4 and 5, virtual staining has been evaluated using a *single* virtual staining prediction, sampled from the learned posterior distribution. As already alluded to in section 1 and as shown in the variability of virtual staining predictions in Chapter 4, virtual staining is an ill-posed problem; there can not be one virtual staining solution given a single bright-field image. Indirectly, this also means there can't be only one instance biological feature value given the bright-field image. To evaluate only a single sample disregards any quantification of uncertainty in virtual staining predictions within our evaluation framework. For high-impact applications of virtual staining, such as the work explored in this thesis, this uncertainty needs to be understood and quantified for different generative modelling approaches.

One could also ask whether it is possible to use an ML method to directly predict the posterior distribution of instance biological features from the bright-field image without the need for virtual staining? Such a method could directly produce uncertainty measurements over instance biological features instead of approximating these values via sampling and the use of an evaluation pipeline. Whether predicting the posterior distribution of instance biological features directly or indirectly, evaluating different predicted posterior distributions, for instance, biological features with access to only a single sample from the true posterior, remains an unexplored challenge.

In Chapter 6 we explore the instance biological feature posterior distributions of the previously explored cGAN approach in Chapter 4 and 5 as well as Conditional Denoising Diffusion Probabilistic Model (cDDPM), an adaptation of DDPM for the virtual staining. We present early principled works into the design of a novel direct method, Bright2Bio-Net, that learns to directly predict instance-wise biological feature distributions from bright-field images. We present three generative model evaluation metrics which given access to

only a single ground truth posterior sample can evaluate a predicted posterior distribution. These include the average log-likelihood, the Wasserstein distance between the standard scores and $N(0,1)$, and the Wasserstein distance between percentile ranks and $U(0,100)$. We provide a systematic analysis of the performance of the proposed measurements compared to population measurements used in Chapters 4 and 5. Our findings reveal new insights about the uncertainty for instance biological feature distributions obtained via virtual staining, and reveal new findings that are missed when using traditional population measurements.

For a concise summary of our findings for this question, refer to section 7.0.3

1.6 GSK HTS Dataset

All experiments within this thesis are based on a pool of 1,631,232 individual high-resolution 1080x1080 images generated as part of a GSK HTS assay screened for toxicity with one of three cell types: lung, ovarian and breast, represented by six different cell lines. The data comprises 118 16x24 well plates with each plate containing 384 wells comprised of a combination of dimethyl sulfoxide (DMSO) (negative control) as it has a relatively low order of systemic toxicity [30], 10 GSK candidate compounds (anonymised to C1, C2...C10) with 6 levels of toxicity from low to high and known apoptosis (programmed cell death) inducing compounds etoposide [31], and staurosporine [32] with high orders of toxicity (positive controls). All plates of one cell type have a fixed layout of compounds and controls across the wells. Every well consists of 9 fields of view, each containing a bright-field and three co-registered fluorescent stains; fluorescein (FITC) for cytoplasm, 6-diamidino-2-phenylindole (DAPI) for nuclei detection and Cyanine (Cy5) for DNA-damage. Figure 1.3 shows the compound and control layout for a lung cell plate, as well as the described hierarchy from plate to individual fluorescence image channels

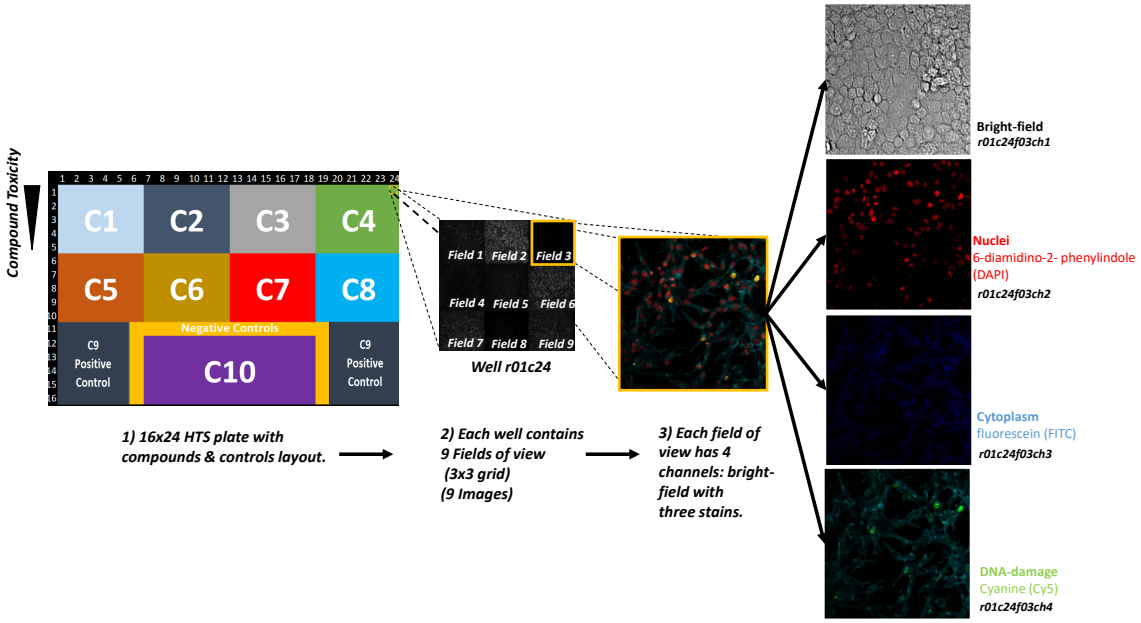


Figure 1.3: **GSK High-Throughput Screening Plate for Toxicity**. 16x24 well plate containing 384 wells with a combination of negative control wells; dimethyl sulfoxide (DMSO), 10 GSK candidate compounds with 6 levels of toxicity from low to high and positive control wells with known apoptosis (programmed cell death) inducing compounds, etoposide and staurosporine with high orders of toxicity. Each well is composed of 9 fields of view each containing a bright-field and three co-registered fluorescent stains; fluorescein (FITC) for cytoplasm, 6-diamidino-2-phenylindole (DAPI) for nuclei detection and Cyanine (Cy5) for DNA-damage.

Chapter 2

Background

In this Chapter, we will discuss some of the core generative modelling approaches to image-to-image translation. As mentioned in Chapter 1, image-to-image translation can be approached using regression-based methods or using conditional generative models. Conditional generative models are a specific type of generative model in which the goal is to approximate the conditional probability distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ instead of the unconditional probability distribution $P(\mathbf{y}_i)$. $P(\mathbf{y}_i|\mathbf{x}_i)$ can then be sampled from to produce $\hat{\mathbf{y}}_i$ given $\mathbf{x}_i$ as input. Image-to-image translation is best framed as a conditional generative modelling task because, unlike regression-based approaches which learn a 1-to-1 mapping from $\mathbf{x}_i$ to $\mathbf{y}_i$, these methods learn to model complex multimodal distributions of plausible outputs from the same input and in doing so capture uncertainty and produce realistic and sharper samples compared to regression-based approaches.

Latent variable models form a core component of modelling complex multimodal distributions by starting with a simple distribution (eg. Gaussian) and then transforming it using a latent variable. The observed data can be thought of as being generated from a simpler, lower-dimensional latent space. By learning the relationship between the latent space and the observed data, conditional generative models capture the underlying structure of the data and can be used to generate samples. For a more detailed explanation of latent variable models, we refer the reader to Doersch [33].

Conditional generative models can be classified into two broad categories: *explicit* likelihood-based and *implicit* [9]. Explicit conditional generative models, such as Condi-

tional Variational Autoencoders (cVAE) [25, 26, 34] and conditional diffusion probabilistic models [28] learn the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ by directly optimizing the evidence lower bound of the likelihood function. In contrast, implicit conditional generative models, such as Conditional Generative Adversarial Networks (cGANs), learn the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ but do not explicitly optimise its likelihood function. Instead of maximising the likelihood function or its lower bound, cGANs approximate the posterior distribution by simultaneously training a generative model G_θ^y that produces samples from $P(\mathbf{y}_i|\mathbf{x}_i)$ and a discriminative model D_θ. G_θ^y captures the data distribution $P(\mathbf{y}_i|\mathbf{x}_i)$, and D_θ^y estimates the probability that a sample came from the training data rather than G_θ^y [35].

We now review the core mathematical basis behind these three approaches and show how they are optimised. In Chapters 4 and 5 cGANs are used for all experiments while in Chapter 6 we compare the posterior distributions of cGANs and cDDPM. Despite not using cVAEs in any Chapter, we include the mathematical basis here as both cVAEs and cDDPM are explicit likelihood-based models and understanding how cVAEs are optimised can aid in understanding how cDDPM is optimised.

As introduced in Chapter 1, we begin with access to a finite set of i.i.d training sample image pairs $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$ from the bright-field domain and fluorescence domain conventionally generated as part of a traditional HTS pipeline. $\mathbf{x}_1 \cdots \mathbf{x}_N$ are drawn from $P(\mathbf{x})$ and $\mathbf{y}_1 \cdots \mathbf{y}_N$ are drawn from the unknown posterior distributions $P(\mathbf{y}_i|\mathbf{x}_i)$. We are interested in learning a parametric approximation to $P(\mathbf{y}_i|\mathbf{x}_i)$ via a neural network that enables sampling $\mathbf{y}_i$ given a single $\mathbf{x}_i$.

2.1 Conditional Variational Autoencoders (cVAEs)

This section draws heavily on the treatment by Doersch [33] on the motivations for VAEs to be very insightful. We borrow the structure and core VAE equations within the explanation but adapt them to the conditional VAE setting, required for image-to-image translation.

A VAE [25] consists of three components: an encoder, the latent space and a decoder. The encoder takes a fluorescence image $\mathbf{y}_i$ and maps it to the latent space. The decoder then samples from the latent space to produce the distribution $P(\mathbf{y}_i|\mathbf{z})$, which outputs

$\hat{\mathbf{y}}_i \approx \mathbf{y}_i$. Similar to the noise sample used as input to the generator in GANs [35], the VAE decoder takes as input a noisy sample from a simple distribution such as a standard Gaussian and learns to reshape the sample into an approximation of a more complex target distribution $P(\mathbf{y}_i|\mathbf{z}) \approx P(\mathbf{y}_i)$. To train the decoder, we require $P(\mathbf{z}|\mathbf{y}_i)$ which can not be computed analytically. Instead, we use an encoder network with parameters θ which learns to construct the latent space distribution $Q_\theta(\mathbf{z}|\mathbf{y}_i)$ such that it approximates the intractable distribution $P(\mathbf{z}|\mathbf{y}_i)$ helping to improve the reconstruction $\hat{\mathbf{y}}_i$. Once trained, the encoder is no longer required. We can sample from the latent space and use the decoder to generate samples from the approximation of the target distribution.

The cVAE [34] (as shown in Figure 2.1) also consists of an encoder, a latent space and a decoder. In addition to the fluorescence image $\mathbf{y}_i$ used within VAEs, cVAEs condition both the encoder and the decoder on the bright-field image $\mathbf{x}_i$ whereby the decoder wants to produce a distribution over the fluorescence images conditional on the bright-field images.

For each $\mathbf{x}_i$ and $\mathbf{y}_i$ pair within the bright-field domain X, and fluorescence domain Y the encoder implements a function via a neural network with parameters θ that approximates the distribution $P(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ via $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ taking both $\mathbf{x}_i$ and $\mathbf{y}_i$ as input and producing two values; $\mu_\theta(\mathbf{y}_i, \mathbf{x}_i)$ and $\sigma^2_\theta(\mathbf{y}_i, \mathbf{x}_i)$ per dimension of the latent space.

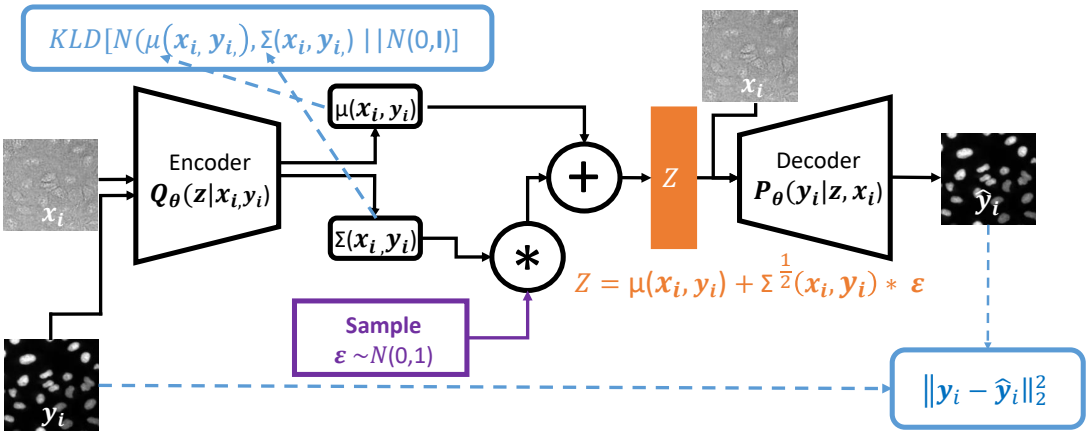


Figure 2.1: **Conditional VAE (cVAE) architecture illustrating the reparametrisation trick**. For a given bright-field image $\mathbf{x}_i$ and target fluorescence image $\mathbf{y}_i$ the deterministic encoder Q_θ takes both $\mathbf{x}_i$ and $\mathbf{y}_i$ as input. The encoder network (CNN) Q_θ outputs parameters $\mu(\mathbf{x}_i, \mathbf{y}_i)$ and $\Sigma(\mathbf{x}_i, \mathbf{y}_i)$ of the approximate posterior $q_\theta(z|\mathbf{x}_i, \mathbf{y}_i)$. The latent variable Z is sampled using the reparametrisation trick $Z = \mu(\mathbf{x}_i, \mathbf{y}_i) + \Sigma^{(1/2)}(\mathbf{x}_i, \mathbf{y}_i) \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. The decoder (CNN) $P_\theta(\mathbf{y}_i|z, \mathbf{x}_i)$ reconstructs the input $\mathbf{Y}_i$ from the sampled latent variable Z. The model is trained by minimising both the reconstruction loss $\left\|\mathbf{y}_i - \hat{\mathbf{y}}_i\right\|_2^2$ (which is equivalent to maximising the log-likelihood) and the KL divergence $KLD[N(\mu(\mathbf{x}_i), \Sigma^{1/2}(\mathbf{x}_i))||\mathcal{N}(0, 1)]$ between the approximate posterior and the prior. At inference time, we no longer require the encoder, only the decoder, which conditions on the $\mathbf{x}_i$ and a random noise sample ϵ to produce $\hat{\mathbf{y}}_i$.

The latent space is constructed according to the PDF $P(\mathbf{z})$, which is a Gaussian distribution $\mathcal{N}(0, \mathcal{I})$ per dimension of the high-dimensional latent space Z. The encoder outputs $\mu_\theta(\mathbf{y}_i, \mathbf{x}_i)$ and $\sigma_\theta^2(\mathbf{y}_i, \mathbf{x}_i)$ are used to transform a random Gaussian sample ϵ from $P(\mathbf{z})$ into a latent variable $\mathbf{z}$ relevant to $\mathbf{y}_i$ and $\mathbf{x}_i$.

This $\mathbf{z}$ is then used by the decoder to produce the distribution $P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)$ which $= \mathcal{N}(\mu_\theta(\mathbf{x}_i, \mathbf{z}), \sigma_\theta^2(\mathbf{x}, \mathbf{z}) \cdot \mathcal{I})$ whereby the μ_θ and σ_θ^2 are a function of the bright-field image $\mathbf{x}_i$ and the latent variable $\mathbf{z}$. We provide an overview of this process in Figure 2.1. The distribution $P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)$ parameterised by the decoder can then be computed, and is continuous in θ which enables back-propagation.

More formally, cVAEs learn to maximise the conditional log posterior distribution of each $\mathbf{y}_i$ under the generative process according to

$$\max \log P(\mathbf{y}_i|\mathbf{x}_i) = \log \int P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i) P(\mathbf{z}) dz. \tag{2.1}$$

whereby the encoder produces a distribution $Q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i)$, over the latent space given a pair of $\mathbf{x}_i$ and $\mathbf{y}_i$. We can view this as the expectation of $P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)P(\mathbf{z})$ under $Q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i)$

$$\log P(\mathbf{y}_i|\mathbf{x}_i) = \log \mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)} \left[\frac{P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)P(\mathbf{z})}{Q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i)} \right]. \tag{2.2}$$

As shown in [22] via the Jensen's inequality the $\log \mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\cdots]$ is always greater than $\mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)} \log[\cdots]$ therefore by moving the log inside of the expectation, we can get a lower bound on $\log P(\mathbf{y}_i|\mathbf{x}_i)$. Doing so and expanding the terms from Equation 2.2 gives:

$$\log P(\mathbf{y}_i|\mathbf{x}_i) \geq \mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)] - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i)||P(\mathbf{z})] \tag{2.3}$$

which represents the variational evidence lower bound "ELBO", which is the loss function used for training VAEs [25, 34].

We will now explore this in more detail, utilising the structure outlined in [33] whilst adding more detail and missing steps where we feel appropriate. We begin by first exploring how the target distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ relates to $\mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}P(\mathbf{y}_i|\mathbf{z}_i, \mathbf{x}_i)$. We remove the conditioning of $\mathbf{x}_i$ and replace $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ with $Q_\theta(\mathbf{z})$ for clarity purposes.

We begin with the Kullback-Leibler divergence (KLD) between $Q_\theta(\mathbf{z})$ and $P(\mathbf{z}|\mathbf{y}_i)$:

$$\mathrm{KLD}[Q_\theta(\mathbf{z})||P(\mathbf{z}|\mathbf{y}_i)] = \mathbb{E}_{z\sim Q(\mathbf{z})}[\log Q_\theta(\mathbf{z}) - \log P(\mathbf{z}|\mathbf{y}_i)]. \tag{2.4}$$

Using Bayes rule we can obtain both $P(\mathbf{y}_i)$ and $P(\mathbf{y}_i|\mathbf{z})$ from Equation 2.4 as:

$$\mathrm{KLD}[Q_\theta(\mathbf{z})||P(\mathbf{z}|\mathbf{y}_i)] = \mathbb{E}_{z\sim -Q_\theta(\mathbf{z})} \left[\log Q_\theta(\mathbf{z}) - \frac{\log P(\mathbf{y}_i|\mathbf{z})P(\mathbf{z})}{\log P(\mathbf{y}_i)} \right]. \tag{2.5}$$

Using the quotient and product logarithms rules, we can rewrite Equation 2.5 as:

$$\mathrm{KLD}[Q_\theta(\mathbf{z})||P(\mathbf{z}|\mathbf{y}_i)] = \mathbb{E}_{z\sim Q_\theta(\mathbf{z})}[\log Q_\theta(\mathbf{z}) - \log P(\mathbf{y}_i|\mathbf{z}) - P(\mathbf{z}) + \log P(\mathbf{y}_i)]. \tag{2.6}$$

Following [33], for Equation 2.6 we can then take $\log P(\mathbf{y}_i)$ out of the expectation because it does not depend on $\mathbf{z}$, negating both sides, rearranging and contracting part of $\mathbb{E}_{Q_\theta(\mathbf{z})}$

into another KLD term produces:

$$-\log P(\mathbf{y}_i) - \mathrm{KLD}[Q_\theta(\mathbf{z})||P(\mathbf{z}|\mathbf{y}_i)] = \mathbb{E}_{Q_\theta(\mathbf{z})}[\log P(\mathbf{y}_i|\mathbf{z})] - \mathrm{KLD}[Q_\theta(\mathbf{z})||P(\mathbf{z})]. \quad (2.7)$$

Since we want to predict $P(\mathbf{y}_i)$, we want to construct the distribution Q such that it makes the $\mathrm{KLD}[Q(\mathbf{z})||P(\mathbf{z})]$ small [33].

We now reintroduce the $\mathbf{x}_i$ and the original $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ which leads to the right-hand-side of the equation showing the "ELBO" as in Equation 2.3:

$$\begin{aligned} &-\log P_\theta(\mathbf{y}_i|\mathbf{x}_i) - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)] \\ &= \underbrace{\mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\log P_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x}_i)] - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P(\mathbf{z})]}_{\text{(ELBO)}}. \end{aligned} \quad (2.8)$$

Whilst the true $P(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ cannot be computed analytically, with an encoder $P_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ with sufficient capacity it should in principle equal $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ which then leaves $-\log P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ to be minimized. On the right-hand side term, the ELBO (evidence lower bound) is written as:

$$\min -\log P_\theta(\mathbf{y}_i|\mathbf{x}_i) \geq \mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)] - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P(\mathbf{z})], \quad (2.9)$$

which is a lower bound on the $-\log P_\theta(\mathbf{y}_i|\mathbf{x}_i)$. The $\mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P(\mathbf{z})]$ regularizes $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ to be close to $P(\mathbf{z})$. Both distributions are multivariate Gaussian distributions and can therefore be computed in closed form.

The $\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)$ is the reconstruction term which encourages the model to reconstruct $\mathbf{y}_i$ well from $\mathbf{z}$, as outlined in [33] we can take one sample of $\mathbf{z}$ and treat $\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)$ as an approximation of $\mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i)]$ as we are already performing stochastic gradient descent over $\mathbf{y}_i$ and $\mathbf{x}_i$ sampled from X and Y and compute the gradient of:

$$\log P_\theta(\mathbf{y}_i|\mathbf{z}, \mathbf{x}_i) - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P(\mathbf{z})]. \quad (2.10)$$

However, we need to compute the gradient through a layer of the model that samples $\mathbf{z}$ from

$Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$, which in its current form has no gradient. To overcome this, VAEs [25, 34] use the "reparametrization trick" [25] (as shown in Figure 2.1) which moves the sampling to an input layer and then given $\mu(\mathbf{x}_i, \mathbf{y}_i)$ and $\Sigma(\mathbf{x}_i, \mathbf{y}_i)$ — the mean and covariance of $Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)$ — we can sample from $\mathcal{N}(\mu((\mathbf{x}_i, \mathbf{y}_i), \Sigma(\mathbf{x}_i, \mathbf{y}_i)$ by first sampling $e \sim \mathcal{N}(0, I)$, then computing $z = \mu(\mathbf{x}_i, \mathbf{y}_i) + \Sigma^{1/2}(\mathbf{x}_i, \mathbf{y}_i) \cdot e$. With the addition of the expectation with respect to the training data, the full equation on which the gradient is computed is:

$$\begin{aligned}&\mathbb{E}_{(\mathbf{x}_i,\mathbf{y}_i)\sim(X,Y)} - \log P_\theta(\mathbf{y}_i|\mathbf{x}_i) \\ &\geq \mathbb{E}_{(\mathbf{x}_i,\mathbf{y}_i)\sim(X,Y)} \left[\mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y}_i,\mathbf{x}_i)}[\log P_\theta(\mathbf{y}_i|\mathbf{x}_i, \mathbf{z} = \mu((\mathbf{x}_i, \mathbf{y}_i), \Sigma(\mathbf{x}_i, \mathbf{y}_i) * \epsilon)] - \mathrm{KLD}[Q_\theta(\mathbf{z}|\mathbf{y}_i, \mathbf{x}_i)||P(\mathbf{z})]\right].\end{aligned} \tag{2.11}$$

Figure 2.1 shows the cVAE generative process and how the reparametrization trick is implemented.

2.2 Conditional Denoising Diffusion Probabilistic Model (cDDPM)

The conditional denoising diffusion probabilistic model (cDDPM) is an adaptation of the denoising diffusion probabilistic model (DDPM) [28, 29] for conditional image generation. At a high level the model is composed of two processes: the forward and the reverse process.

The forward process (Q) starts with a fluorescence image and progressively adds Gaussian noise until only Gaussian noise remains.

The reverse process starts with the Gaussian noise and aims to undo the noise added in the forward process, conditional on the bright-field input to predict the conditional log-likelihood of the fluorescence image given a bright-field image. This distribution P is parameterised by a neural network with parameters θ, with a μ and σ at each pixel. In DDPM [28], the σ is known so the neural network need only to predict μ as shown in Figure 2.2.

In practice, as we will motivate in the following section, the model is trained by minimising the quadratic loss between the known noise value ϵ and the predicted noise of

the network ϵ_θ for each pixel.

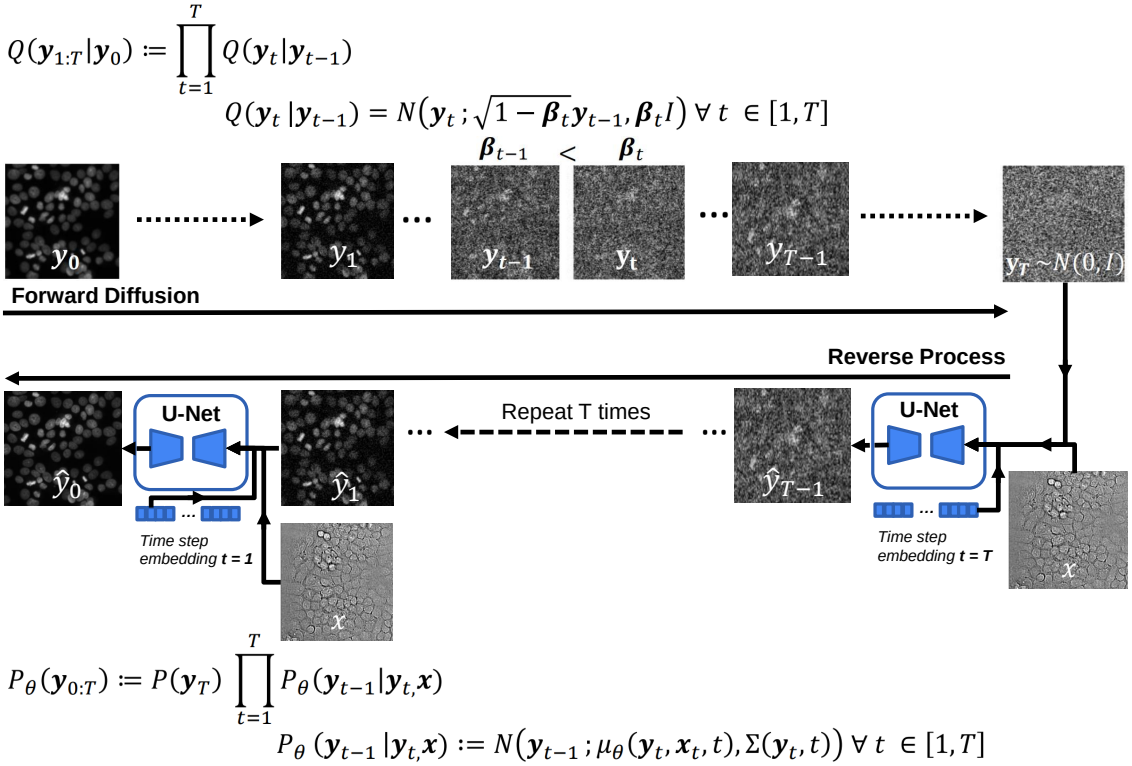


Figure 2.2: **Conditional Denoising Diffusion Probabilistic Model - Forward and Reverse processes.** The forward diffusion process Q (left to right) gradually adds increasing levels of Gaussian noise to the target fluorescence image $\mathbf{y}_0$ according to a noise schedule $\beta_1, \ldots, \beta_T$ until $\mathbf{y}_T \sim \mathcal{N}(0, \mathcal{I})$. This process in the simplest case has no learnable parameters and is defined as the product of the conditional probabilities $Q(\mathbf{y}_t|\mathbf{y}_{t-1})$. For a given t, $\mathbf{y}_t$ is sampled from a normal distribution with a mean of the previous time step and new variance. The reverse inference process P (right to left) iteratively denoises the target image conditioned on a source image $\mathbf{x}$ and the time step embedding t to produce a denoised image at $t-1$. This process is repeated until it reaches a denoised prediction for the original input image $\hat{\mathbf{y}}_0$

Following the works of [28, 29, 36, 37] we now explain how the forward and reverse process of cDDPMs works in more detail. To avoid confusion with time indexes we omit the index i for both bright-field $\mathbf{x}_i$ and fluorescence image $\mathbf{y}_i$, referring to each as $\mathbf{x}$ and $\mathbf{y}$.

The *forward process*, progressively adds increasing levels of Gaussian noise to a fluorescence image $\mathbf{y}$ (which we refer to as $\mathbf{y}_0$ because it has zero added Gaussian noise) over a sequence of time steps T according to a noise schedule $\beta_1, ..., \beta_T$ until we have random noise mimicking a sample from a standard Gaussian distribution. More formally,

as in [28] the forward process is defined as:

$$Q(\mathbf{y}_{1:T}|\mathbf{y}_0) := \prod_{t=1}^{T} Q(\mathbf{y}_t|\mathbf{y}_{t-1}), \tag{2.12}$$

where $Q(\mathbf{y}_{1:T}|\mathbf{y}_0)$ is the product of the conditional probabilities $Q(\mathbf{y}_t|\mathbf{y}_{t-1})$ from $t = 1 : T$. As in [28] to transition from $\mathbf{y}_{t-1}$ to $\mathbf{y}_t$ the following distribution is defined:

$$Q(\mathbf{y}_t|\mathbf{y}_{t-1}) := \mathcal{N}(\mathbf{y}_t|\sqrt{1-\beta_t}\mathbf{y}_{t-1}, \sqrt{\beta_t}\mathbf{I}), \tag{2.13}$$

whereby $\mathbf{y}_t$ can be viewed as some scaled version of $\mathbf{y}_{t-1}$ + additional noise.

At time point T, we want to end up with $y_T \sim \mathcal{N}(0, \mathbf{I})$, meaning the contribution of $\mathbf{y}_{t-1}$ and as a consequence $\mathbf{y}_0$ to y_T should be close to 0 while the contribution of the noise should be close to 1. With a sufficient number of time steps T, the scaling applied in Equation 2.13 can be viewed as the smooth diffusion of a sample $\mathbf{y}_0$ from a complex distribution to a simple Gaussian distribution. That is, as β_t increases, the weight of $\mathbf{y}_{t-1}$ decreases while the weight of the noise increases. Because each time point t is only dependent on the previous time point, as outlined in [28], the forward process is a stochastic Markov chain.

Importantly, one can characterize the distribution of any $\mathbf{y}_t$ given the clean image $\mathbf{y}_0$ which we refer to as the posterior distribution of $Q(\mathbf{y}_t|\mathbf{y}_0)$ by marginalizing out the intermediate steps [36] as:

$$\begin{aligned} \mathbf{y}_t &= \sqrt{1-\beta_t}\mathbf{y}_{t-1} + \sqrt{\beta_t}\mathcal{N}(0, \mathbf{I}), \quad \alpha_t = 1 - \beta_t \\ \mathbf{y}_t &= \sqrt{\alpha_t}\mathbf{y}_{t-1} + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \end{aligned} \tag{2.14}$$

Where ϵ is a noise sample from a standard Gaussian distribution. In Equation 2.14 where

$\mathbf{y}_t$ is expressed in terms of $\mathbf{y}_{t-1}$ we can also express $\mathbf{y}_t$ in terms of $\mathbf{y}_{t-1}$ and $\mathbf{y}_{t-2}$ as:

$$\begin{aligned} \mathbf{y}_t &= \sqrt{\alpha_t}\underbrace{\mathbf{y}_{t-1}}_{(1)} + \sqrt{1-\alpha_t}\epsilon_t. \\ \mathbf{y}_t &= \sqrt{\alpha_t}(\underbrace{\sqrt{\alpha_{t-1}}\mathbf{y}_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1}}_{(2)}) + \sqrt{1-\alpha_t}\epsilon_t \end{aligned} \tag{2.15}$$

where we replace (1) with (2). Then by separating the $\mathbf{y}_{t-2}$ term we have:

$$\mathbf{y}_t = \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}\mathbf{y}_{t-2} + \underbrace{\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-1}}_{(1)} + \underbrace{\sqrt{1-\alpha_t}\epsilon_t}_{(2)}. \tag{2.16}$$

Terms (1) and (2) are two independent Gaussian noises with variance $\alpha_t - \alpha_t\alpha_{t-1}$ in (1) and $1-\alpha_t$ in (2). These can be rewritten as one Gaussian:

$$\begin{aligned} \mathbf{y}_t &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{y}_{t-2} + \sqrt{1-\alpha_t+\alpha_t-\alpha_t\alpha_{t-1}}\epsilon \\ \mathbf{y}_t &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{y}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon. \end{aligned} \tag{2.17}$$

We can then repeat this process by substituting $\mathbf{y}_{t-3}$ for $\mathbf{y}_{t-2}$ until T steps where we are left with:

$$\mathbf{y}_t = \sqrt{\underbrace{\alpha_t\alpha_{t-1}\alpha_{t-2}\ldots\alpha_2\alpha_1}_{(1)}}\mathbf{y}_0 + \sqrt{1-\underbrace{\alpha_t\alpha_{t-1}\alpha_{t-2}\ldots\alpha_2\alpha_1}_{(2)}}\epsilon \tag{2.18}$$

where (1) = (2) and $\prod_{i=1}^{t}\alpha_i = \bar{\alpha}_t$ we can then rewrite Equation 2.18 as:

$$\mathbf{y}_t = \sqrt{\bar{\alpha}_t}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2.19}$$

and access any $\mathbf{y}_t$ given a clean image $\mathbf{y}_0$ and $\bar{\alpha}_t$ which can be computed through $\mathbf{y}_0$ and a sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

This posterior distribution $Q(\mathbf{y}_t|\mathbf{y}_0)$ is important when parameterising the reverse process and formulating a variational lower bound on the log-likelihood [28], which we discuss in the following section.

The *reverse process* tries to undo the forward process, enabling the generation of images from noise. The reverse process is also a stochastic Markov chain that can be

viewed as a sequence of T denoising steps, with Gaussian transitions that have the same form as the forward process outlined in Equation 2.13. In addition, the reverse process also conditions on the bright-field image $\mathbf{x}$. Formally, the reverse process is defined as:

$$P(\mathbf{y}_{0:T}) := P(\mathbf{y}_T) \prod_{t=1}^{T} P(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x}). \tag{2.20}$$

Sampling from Equation 2.20 is intractable because when computing the distribution $P(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x})$, $\mathbf{y}_t$ is derived from the forward process 2.12 which is derived from the input image $\mathbf{y}_0$ which the reverse process doesn't have access to. However, we can use a neural network with parameters θ to learn an approximate parametrised form of the distribution on which we can sample. This distribution can be written as

$$P_\theta(\mathbf{y}_{0:T}) := P(\mathbf{y}_T) \prod_{t=1}^{T} P_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x}). \tag{2.21}$$

To transition from $\mathbf{y}_t$to $\mathbf{y}_{t-1}$ the following distribution is defined:

$$P_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x}) := \mathcal{N}(\mathbf{y}_t; \mu_\theta(\mathbf{x}, \mathbf{y}_t, t), \Sigma(\mathbf{y}_t, t). \tag{2.22}$$

As outlined in the forward process, in the simplest of cases, the noise schedule is known, so the values of μ are what we are interested in learning. The goal of approximating an intractable distribution via a neural network is similar to the cVAE case (discussed in Section 2.1), where we want to generate images as close to the data distribution as possible by explicitly optimising the conditional log-likelihood over the observed data. This can be written for cVAEs and cDDPM as follows:

$$\underbrace{\log P(\mathbf{y}|\mathbf{x}) = \int P(\mathbf{y}|z, \mathbf{x}) dz}_{\text{cVAE}} \equiv \underbrace{\log P(\mathbf{y}_0|\mathbf{x}) = \log \int P(\mathbf{y}_{0:T}|\mathbf{x}) dy_{1:T}}_{\text{cDDPM - Reverse Process}}. \tag{2.23}$$

Instead of conditioning on a single latent variable z for cVAEs, for the reverse diffusion process, we move through a sequence of latent variables $\mathbf{z} = \mathbf{y}_{1:T}$ which corresponds to all noisy versions of $\mathbf{y}$, where $Q_\theta(\mathbf{z}|\mathbf{y}, \mathbf{x})$ in the cVAE case is $\equiv Q(\mathbf{y}_t|\mathbf{y}_{t-1})$ but Q is fixed

with no learnable parameters. As stated in Section 2.1 Equation 2.2 for cVAEs we can compute

$$\log P(\mathbf{y}_0|\mathbf{x}) = \log \mathbb{E}_{Q_\theta(\mathbf{z}|\mathbf{y},\mathbf{x})}\left[\frac{P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})}{Q_\theta(\mathbf{z}|\mathbf{x},\mathbf{y})}\right], \tag{2.24}$$

the corresponding equation for cDDPM can be viewed as:

$$\log P(\mathbf{y}_0|\mathbf{x}) = \log \mathbb{E}_{Q(\mathbf{y}_{1:T}|\mathbf{y}_0,\mathbf{x})}\left[\frac{P_\theta(\mathbf{y}_{0:T}|\mathbf{x})}{Q(\mathbf{y}_{1:T}|\mathbf{y}_0)}\right], \tag{2.25}$$

where the denominator is the forward process and the numerator is the reverse process. Given that the forward and reverse processes are Markov chains we can view the term within the expectation of Equation 2.25 as the product of the conditional probabilities:

$$\log P(\mathbf{y}_0|\mathbf{x}) = \log \mathbb{E}_{Q(\mathbf{y}_{1:T}|\mathbf{y}_0)}\left[\frac{P(\mathbf{y}_T)\prod_{t=1}^{T} P_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t,\mathbf{x})}{Q(\mathbf{y}_1|\mathbf{y}_0)\prod_{t=2}^{T} Q(\mathbf{y}_t|\mathbf{y}_{t-1})}\right]. \tag{2.26}$$

Notice, the term t = 1 is pulled out from the product in the denominator, which starts at t = 2. The authors [28] do this to avoid numerical issues when applying Bayes theorem to $Q(\mathbf{y}_1|\mathbf{y}_0)$. Instead, they apply Bayes theorem to the contents of the product from t = 2 to obtain:

$$Q(\mathbf{y}_1|\mathbf{y}_0)\prod_{t=2}^{T} Q(\mathbf{y}_t|\mathbf{y}_{t-1}) = Q(\mathbf{y}_1|\mathbf{y}_0)\prod_{t=2}^{T}\frac{Q(\mathbf{y}_{t-1}|\mathbf{y}_t,\mathbf{y}_0)Q(\mathbf{y}_t|\mathbf{y}_0)}{Q(\mathbf{y}_{t-1}|\mathbf{y}_0)}. \tag{2.27}$$

After expanding the product in Equation 2.27 and removing all cancelled-out terms, we have:

$$Q(\mathbf{y}_1|\mathbf{y}_0)\prod_{t=2}^{T} Q(\mathbf{y}_t|\mathbf{y}_{t-1}) = Q(\mathbf{y}_T|\mathbf{y}_0)\prod_{t=2}^{T} Q(\mathbf{y}_{t-1}|\mathbf{y}_t,\mathbf{y}_0). \tag{2.28}$$

Substituting the right-hand side back into the denominator in Equation 2.26 and separating out the t = 1 within the product from the numerator, we have:

$$\log P(\mathbf{y}_0|\mathbf{x}) = \log \mathbb{E}_{Q(\mathbf{y}_{1:T}|\mathbf{y}_0)}\left[\frac{P(\mathbf{y}_T)P_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{x})\prod_{t=2}^{T} P_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t,\mathbf{x})}{Q(\mathbf{y}_T|\mathbf{y}_0)\prod_{t=2}^{T} Q(\mathbf{y}_{t-1}|\mathbf{y}_t,\mathbf{y}_0)}\right]. \tag{2.29}$$

As shown in the cVAE case via Equation 2.3, by moving the log inside of the expec-

tation, we can get a lower bound on $\log P(\mathbf{y}_0|\mathbf{x})$:

$$\log P(\mathbf{y}_0|\mathbf{x}) \geq \mathbb{E}_{Q(\mathbf{y}_{1:T}|\mathbf{y}_0)} \left[\underbrace{\log \frac{P(\mathbf{y}_T)}{Q(\mathbf{y}_T|\mathbf{y}_0)}}_{(1)} + \underbrace{\log P_\theta(\mathbf{y}_0|\mathbf{y}_1, \mathbf{x})}_{(2)} + \underbrace{\sum_{t=2}^{T} \log \frac{P_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x})}{Q(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{y}_0)}}_{(3)} \right], \tag{2.30}$$

which is very similar to the lower bound of the cVAE in Equation 2.9.

Term (1) is similar to the KLD term over the $P(z)$ in cVAE, but as Q is fixed with no learnable parameters this is ignored during training. Term (2) is the reconstruction of $\mathbf{y}_0$ given the last reverse step $\mathbf{y}_1$ and bright-fied image $\mathbf{x}$ which is the same as $P(\mathbf{y}_i|\mathbf{z}, \mathbf{x})$ in Equation 2.9 for cVAEs. Finally, term (3), is the sum of the KL divergences between the intermediate Gaussians produced by the reverse and forward processes, which are now of the same form. Note that Q has access to $\mathbf{y}_0$, which makes the denoising process far easier, and so P_θ given access to $\mathbf{x}$ is trying to mimic Q. As shown in [28] for the unconditional setting, for the conditional setting, term (3) can be written as:

$$\begin{aligned} &\mathbb{E}_{Q(\mathbf{y}_{1:T}|\mathbf{y}_0)} \mathrm{KLD}[\mathcal{N}(\mathbf{y}_{t-1}; \mu_\theta(\mathbf{y}_t|\mathbf{x}), \Sigma_Q(t)) || \mathcal{N}(\mathbf{y}_{t-1}; \mu_Q(\mathbf{y}_t), \Sigma_Q(t))] \\ &= \frac{1}{2\sigma_Q^2} \frac{\beta_t^2}{(1-\bar{\alpha}_t)(\alpha_t)} ||(\epsilon_\theta - \epsilon)_2^2||. \end{aligned} \tag{2.31}$$

Term (2) in Equation 2.30 is also modelled under Equation 2.31 by approximating the noise ϵ that was added at $\mathbf{y}_0$ to get to $\mathbf{y}_1$ as ϵ_θ. In practice, in the unconditional setting, authors [28] ignore the scaling factor and therefore, in the conditional setting, the loss that is optimised is:

$$\mathcal{L}_{\text{simple}}(\theta) := \mathbb{E}_{\mathbf{x},t,\mathbf{y}_0,\epsilon} \left[||\epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}_{(1)}, \mathbf{x}, t) - \epsilon||^2 \right], \tag{2.32}$$

where term (1) is $\mathbf{y}_t$ which along with the bright-field image $\mathbf{x}_i$ and time point t is used as input to the neural network parametierised by θ to predict noise ϵ.

2.3 Conditional Generative Adversarial Networks (cGANs)

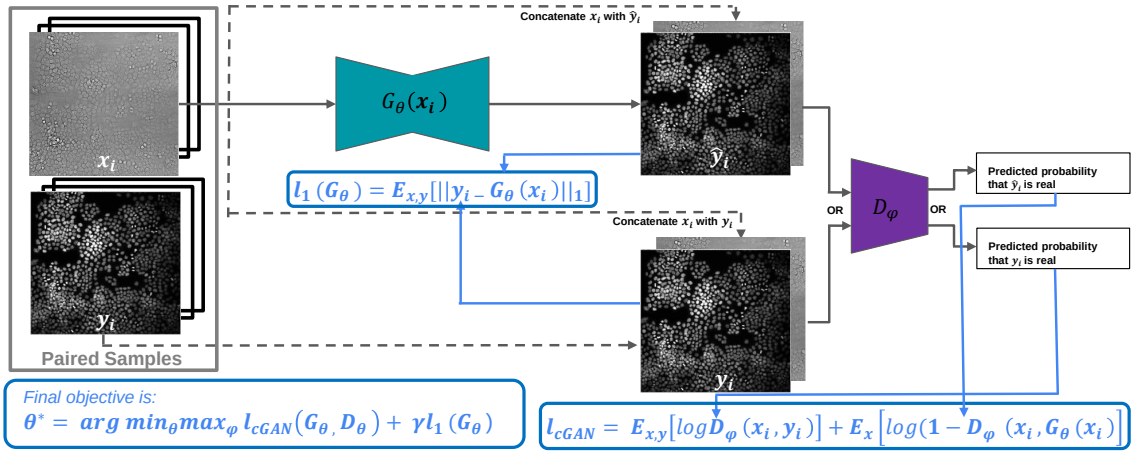


Figure 2.3: **Conditional Generative Adversarial Network (cGAN) architecture and loss function**. The generator G_θ (typically a U-net with residual connections and dropout for stochasticity) takes the bright-field $\mathbf{x}_i$ as input and generates a virtual staining prediction $\hat{\mathbf{y}}_i$. The discriminator D_φ (also typically a U-net with residual connections) then receives two inputs, either $\hat{\mathbf{y}}_i$ or $\mathbf{y}_i$ concatenated with the conditional input $\mathbf{x}_i$. The combined input is passed through D_φ to produce a probability value indicating whether the input was the true fluorescence stain or a virtual staining prediction. We show both scenarios in the figure. The objective of the $\mathcal{L}_{cGAN}$ is shown where G_θ tries to minimise this objective against an adversarial D_φ that tries to maximise it. To improve performance an $\mathcal{L}_1$ reconstruction loss term is introduced between the virtual staining prediction and the target fluorescence stain. The contribution of the $\mathcal{L}_1$ is determined via the parameter γ

Conditional Generative Adversarial Networks (cGANs) [7, 24, 35] are implicit generative models. Unlike cVAE and cDDPM, which are conditioned on both $\mathbf{x}_i$ and Gaussian noise z cGAN implementations [7] showed that the model learnt to ignore the noise. Noise was only introduced into the generator during training via dropout and via MC-Dropout [38] applied during inference, limiting the variability of the model predictions [7]. Additionally, cGANs do not explicitly optimise the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ or its lower bound but instead learn to sample from its PDF.

cGANs do this by simultaneously training a generative model G_θ^Y and a discriminative model D_θ^y. G_θ^y captures the data distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ via $\hat{P}(\mathbf{y}_i|\mathbf{x}_i)$, and D_θ^y estimates the probability that a sample came from $P(\mathbf{y}_i|\mathbf{x}_i)$ rather than G_θ^y [35].

D_θ^y is trained to maximise the log-likelihood of correctly distinguishing real samples from the generator outputs. G_θ^y is trained to minimise D_θ^y's ability to distinguish real from

fake samples. Both D_θ^y and G_θ^y are trained following a min-max objective function;

$$\mathcal{L}_{cGAN} = \min_{G_\theta^y} \max_{D_\theta^y} V(D_\theta^y, G_\theta^y) = \mathbb{E}_{x,y}[\log D_\theta^y(\mathbf{x}_i, \mathbf{y}_i)] + \mathbb{E}_{\mathbf{x}_i}[\log(1 - D_\theta^y(\mathbf{x}_i, G_\theta^y(\mathbf{x}_i))], \tag{2.33}$$

where G_θ^y tries to minimise this objective against an adversarial D_θ^y that tries to maximise it taking the concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$. D_θ^y aims to classify the concatenation of the source image $\mathbf{x}_i$ and its corresponding target image $\mathbf{y}_i$ as real, $D_\theta^y(\mathbf{x}_i, \mathbf{y}_i) = 1$, while classifying $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ as fake, $D_\theta^y(\hat{\mathbf{y}}_i, \mathbf{y}_i, \mathbf{x}_i) = 0$.

Nevertheless, cGANs that rely solely on Equation 2.33 do not produce consistent results. More specifically, the virtual staining sample may not share a similar global structure as the desired target image [39]. To counteract this issue, a pixel reconstruction loss, such as the $\mathcal{L}_1$ loss, is incorporated [7, 40]. This is achieved by calculating the mean absolute error (MAE) between the target and predicted images:

$$\mathcal{L}_1 = \mathbb{E}_{x,y,z}\left[||\mathbf{y}_i - G_\theta^y(\mathbf{x}_i)||_1\right] \tag{2.34}$$

such that the final training objective is given by:

$$\min_{G^y} \max_{D^y} \mathcal{L}_{cGAN} + \gamma \mathcal{L}_1, \tag{2.35}$$

with $\gamma > 0$ as a weighting hyperparameter.

2.4 Practical implications of approaches

From a practical standpoint, each generative modelling approach presents a unique trade-off between speed, quality, and robustness—factors that significantly impact their deployability in real-world virtual staining workflows. Table 2.1 summarises cVAEs, cGANs and cDDPM across 6 practical measurement criteria.

Measurement	cVAEs	cGANs	cDDPM
Inference Speed	**Fast**	**Fast**	Slow
Pixel Quality	Medium	**High**	**High**
Perceptual Quality	Medium	**High**	**High**
Population Distribution Quality	Med-High	**High**	**High**
Training Stability	**High**	Low	**High**
Condition Sensitivity	Medium	Low	**High**

Table 2.1: **Comparison of the practical implications of cVAEs, cGANs and cDDPM.** Across the majority of metrics, cDDPM demonstrates the highest performance but it is limited by its inference speed. cGANs appear very suitable for practical scenarios where speed and quality are most important, but caution must be taken around training instabilities and condition sensitivity. Finally, cVAEs offer fast inference and high training stability, making them appealing to requirements that require real-time, diverse outputs.

cVAEs, offer fast inference and high training stability, making them appealing for resource-constrained environments or applications that require real-time outputs. However, their moderate pixel and perceptual quality may limit their use in high-fidelity diagnostic scenarios, where visual accuracy is paramount. Their medium condition sensitivity suggests moderate responsiveness to input conditions (e.g. bright-field, cell type and phenotype), which could necessitate further tuning to maintain reliability across diverse biological settings.

In contrast, cGANs excel in both pixel and perceptual fidelity, making them well-suited for applications where visual realism is critical, such as pathology workflows involving expert human interpretation. However, their low training stability can pose significant engineering challenges, often requiring extensive hyperparameter tuning, early stopping and regularisation to avoid mode collapse or convergence issues. Additionally, they are prone to hallucinations that, if not correctly quantified, can lead to inaccurate, highly realistic false positives. Furthermore, cGANs low condition sensitivity, specifically to the bright-field, may hinder their ability to produce virtually stained outputs that, on a per-

cell level, contain accurate biological feature values specific found in the corresponding fluorescence-stained cell.

Finally, cDDPMs offer state-of-the-art quality across all output dimensions, including population-level distribution matching, but their slow inference speed makes them less practical for high-throughput scenarios. Their high condition sensitivity, however, enables precise control over generated outputs, which can be highly advantageous in settings where reproducibility and specificity are critical. Whilst they produce the same pixel and perceptual quality as cGANs they are e

2.5 Conclusion

In conclusion, while each of the three modelling paradigms brings distinct strengths, none offers a fully satisfactory solution for virtual staining when all practical dimensions are considered simultaneously. This reveals a clear gap in current methodologies: existing models either compromise on inference speed or robustness, or require trade-offs in biological precision and controllability. Addressing this gap is critical for advancing virtual staining from proof-of-concept demonstrations toward reliable, real-world diagnostic and screening tools. Future work should thus focus on hybrid architectures, architectural optimisations, or knowledge-distillation-based methods that retain the quality and controllability of diffusion models while achieving the speed and stability of cVAEs or cGANs. Moreover, there remains an unmet need for deeper understanding of how these models generalise to common HTS data distribution shifts, such as, cell types, phenotypes, and technical settings—a prerequisite for widespread clinical adoption. This raises a natural question: if all three models can generate virtually stained bright-field images that are ultimately used to quantify biological features, could we instead train a model to predict those biological features directly from the original bright-field images?

Within Chapter 3, we provide a comprehensive review of a broad range of extensions to the three core algorithms (cVAEs, cDDPM, cGANs) explored in this section. Then in Chapters 4 and 5, we explore the use of cGANs for several virtual staining tasks within HTS. We then introduce cDDPM and compare its performance against a cGAN in

Chapter 6 and introduce Bright2Bio-Net, which is a novel method that learns to predict these biological features directly from the bright-field image.

Chapter 3

Literature Review

As outlined in Chapters 1 and 2, image-to-image translation can be approached as a regression problem or by using conditional generative models. Section 3.1 of this Chapter begins by reviewing how approaches to image-to-image translation have evolved from early regression-based models [4, 10] to the more recent generative modelling approaches [7, 8, 11, 19, 24, 27, 28, 39, 41–48]. While maintaining the focus on approaches to image-to-image translation, we also include new architectures and improvements that were first developed for unconditional generative models and later adapted to conditional generative models.

Despite the considerable improvements in approaches to image-to-image translation, the development of evaluation protocols for determining these methods' performance remains far less explored. As these methods have evolved, so have the axioms on which we need to evaluate their performance. Early regression-based approaches were predominantly evaluated in the pixel space with access to some ground truth target, where we want to compare it pixel-by-pixel to our prediction. As techniques have evolved into probabilistic methods, we are now measuring distributions as well as sample quality. Furthermore, as I2I is applied to specific applications, such as virtual staining for HTS, metrics that provide meaningful insights to answering specific application-related questions are also of vital importance. In Section 3.2 of this Chapter, we aim to critically review evaluation protocols for generative models and discuss their trade-offs.

The combination of improved I2I methods and advancements in developing accu-

rate quantitative measures that reflect the nuances between different generative modelling approaches are in part responsible for the increased interest in these techniques within industrial applications. However, a key consideration for large-scale applications is scalability. Devising strategies for training generalisable models is an under-explored area of research within I2I and the wider machine learning (ML) community. In Section 3.3 we aim to critically review current works on generalizable I2I models, including virtual staining models as well as generalizable models within ML more broadly.

3.1 Approaches to image-to-image translation

We begin as previously outlined in Chapter 1 with access to a finite set of i.i.d training sample pairs $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$ from a source and a target domain such as the bright-field and the fluorescence domains. We are interested in learning a parametric approximation to $P(\mathbf{y}_i|\mathbf{x}_i)$ that enables for sampling $\mathbf{y}_i$ given a single $\mathbf{x}_i$. The task of learning this parametric approximation we term image-to-image translation (I2I).

One of the simplest approaches to I2I is to view the task as a regression problem where the aim is to learn a function $g_\theta(\mathbf{x}) = \hat{\mathbf{y}} \approx \mathbf{y}$ with parameters θ via a neural network, that maps the training $(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_N, \mathbf{y}_N)$ pairs, $g_\theta : X \rightarrow Y$.

A number of approaches have been explored for virtual staining, but none have specifically focused on high-throughput screening. Early virtual staining works [4, 10] in applications such as transmitted light and fluorescence microscopy minimised a quadratic loss function via a U-Net [49] and predictions were shown to produce pixel values correlated with those found in the real fluorescence images [4]. However, predictions contained blurry results [4, 10] because virtual staining is an ill-posed problem. No such function exists, given the training data, because each label-free input image could have many corresponding fluorescence images. Regressing to the mean possible virtual staining prediction is a compromise across the different virtual staining predictions. The true unknown set of possible virtual staining solutions can be viewed as the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ of potential fluorescence images $\mathbf{y}_i$ that could occur given the label-free input image $\mathbf{x}_i$.

Conditional generative models (CGMs) [7, 8, 11, 19, 41] learn to approximate this

posterior distribution. These methods have outperformed regression-based approaches for I2I in virtual staining and other domain areas, which have led to dramatic performance improvements on natural image datasets [50, 51] and other bioimaging applications [9]. Unlike regression-based approaches to virtual staining which learn a deterministic direct mapping from the input to output, conditional generative models learn to approximate the posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ which can be used to sample virtual staining solutions $\hat{\mathbf{y}}_i$.

Many of the design improvements to CGMs for image-to-image translation, such as vector quantization [42] or spectral normalization [52] that have led to performance improvements, were first explored for unconditional generative models. In light of this, in the following section, where this is the case, we explore the development of the method in the unconditional setting first and then review how these methods were adapted to the conditional setting.

One of the early generative modelling approaches was *Auto-regressive models* (ARMs) [53–55] which directly model the log-likelihood of the training data by progressively predicting each pixel based on previous pixels. ARMs were shown to have strong distribution coverage [54]. However, the sequential generation of pixels makes for slow image generation, limiting the scalability of ARMs. *Normalizing flows* [56–58] and *VAEs* [25, 34, 59] (discussed in detail in Chapter 2) improved sampling speed but early approaches were unable to produce the quality of samples from generative adversarial networks (GANs).

In light of this, *Vector Quantised Variational Auto Encoders* (VQ-VAE) [42] present several changes to VAEs to reduce blurriness and improve sample quality. First, a discrete latent encoding is introduced via a code book instead of the continuous latent representation used in VAEs (as discussed in Chapter 2). In short, the encoder E_θ takes as input $\mathbf{x}_i$ and produces a vector quantized representation $z = E_\theta(\mathbf{x}_i)$ to which each pixel within $\mathbf{x}_i$ is mapped to 1 of C code book embedding vectors by taking $\mathcal{L}_2$ norm between z at each pixel and the nearest codebook embedding vector $\mathbf{z}_c$ producing the $Q(\mathbf{z} = \mathbf{c}|\mathbf{x}_i)$. The quantised representation $Q(\mathbf{z}|\mathbf{x}_i)$ is then used as input to the decoder, which outputs the reconstruction of $\mathbf{y}_i$. During inference, the decoder only needs to look up the nearest codebook entry, which can be faster than sampling from the continuous latent distribution

of VAEs [42].

Instead of using the VAE KL divergence term (as outlined in Chapter 2 Equation 2.8) between $P(z)$ and $Q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i)$ VQ-VAEs use an additional auto-regressive model; PixelCNN [53] to learn the appropriate distribution $P(z)$ of each codebook entry using ancestral sampling [22]. When using VQ-VAEs the bias introduced by VAEs of enforcing the latent space to be similar to the standard Gaussian distribution is removed because the additional PixelCNN can learn the appropriate distribution tailored to the input data. VQ-VAEs do not suffer from the issue of "posterior collapse" in VAEs, i.e., the latents are ignored as the decoder is powerful enough to model the input data perfectly [42]. This is analogous to the challenges with GANs [7] whereby the generator can ignore the conditional noisy input [7]. However, some of the samples produced by VQ-VAE are still blurry [42] and not as realistic compared to GANs [7]. *VQ-VAE2* [43] was later introduced to address these challenges. VQ-VAE2 is very similar to VQ-VAE but instead uses a hierarchical-VAE [60] approach, generating multi-scale vector quantised codes, each with learnt prior distributions. This approach enables modelling local and global information separately as well as each PixelCNN [53] learning the appropriate distribution $P(z)$ tailored to the correlations that exist at different scales [43]. The authors [43] qualitatively show reduced blurriness in image samples and quantitative performance improvements compared to VQ-VAE [42].

VQ-VAEs were later adapted for image-to-image translation. Named *VQ-I2I* [44], both $\mathbf{x}_i$ and $\mathbf{y}_i$ pairs are encoded using a shared encoder that produces separate quantised code books for $\mathbf{x}_i$ and $\mathbf{y}_i$. These are then used as input into separate decoder networks for $\mathbf{x}_i$ and $\mathbf{y}_i$ to produce both translated outputs. In addition, VQ-I2I [44] also has two discriminator networks (D^x_θ, D^y_θ) for $\mathbf{x}_i$ and $\mathbf{y}_i$, two style encoders (S^x_θ, S^y_θ), one for each domain, used to extract style codes $S^y_\theta(\mathbf{y}_i)$ and $S^x_\theta(\mathbf{x}_i)$ from the input images. The same style encoders are then used to extract the style codes from the predicted $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{y}}_i$ as $S^x_\theta(\hat{\mathbf{x}}_i)$, $S^y_\theta(\hat{\mathbf{y}}_i)$. During training, the $\mathcal{L}_1$ norm between the styles obtained from the real input data compared to the decoder outputs is computed. Similarly to VQ-VAE, the codebook loss, which is $\mathcal{L}_2$ norm of the encoder output and the nearest codebook embedding vector z_c is computed [44].

In contrast to VQ-VAE, the codebook loss is encouraged to learn the domain-invariant

content information while the style vectors learn the style information. Qualitative results showed sharper predictions and fewer repeating patterns compared to the benchmark supervised methods [7, 61]. However, quantitative measurements including Frechét inception distance (FID) [62] and the natural image quality evaluator (NIQE) [63] showed that performance was still not as good *pix2pix GAN* [7].

3.1.1 Pix2pix GANs

Pix2pix GAN is an implicit generative model and was the first conditional generative adversarial network (cGAN) (discussed in Chapter 2) used for image-to-image translation. It explored I2I of 8-bit low-resolution natural images with results showing improved realism and less blurriness [35] compared to previous regression-based approaches [4, 10] and VQ-I2I [44]. However, despite these improvements, pix2pix GAN suffered from mode collapse [7, 35] whereby the model failed to capture the full target posterior distribution. While producing sharper images, pix2pix GAN was prone to hallucinating realistic spatial structure even where it did not exist in the input image [7]. Pix2pix GAN also suffered from unstable training [7].

cVAE-GAN [64] sought to combine the quality of pix2pix GAN samples with the distribution capturing properties of VAEs. By using a cVAE as the generator, cVAE-GAN produced smaller distances between its predicted population and the target population distributions compared to pix2pix GAN whilst retaining high fidelity samples [64]. However, cVAE-GAN [64] had a very complex architecture and loss functions, which meant the balance between the generator and discriminator components was difficult to maintain, leading to training instability and potential mode collapse [65]. Another challenge with pix2pix GAN and cVAE-GAN was that performance would decline at high resolution [24, 64]. One reason, highlighted by [24] was that the $\mathcal{L}_1$ reconstruction loss failed to capture the perceptual judgement humans exhibit. For example, if two identical images are shifted by one or two pixels, the reconstruction loss would be high when in fact, as humans, we would still recognise the similarities [66].

Stemming from this, *Pix2pixHD GAN* [24] was developed. The method introduced a

perceptual loss function [66] that corresponded to the $\mathcal{L}_1$ loss between feature representations obtained for the virtual staining predictions and feature representations obtained for the target images. During training, these representations were extracted from different layers of a pre-trained VGG network, and the $\mathcal{L}_1$ between the virtual staining representation and target representation averaged over all layers was minimised. The authors also used 2 discriminators instead of the single discriminator used in pix2pix GAN. Each discriminator was identical in design, except one performed its task on down-sampled versions of the training data. Maintaining the same receptive field of both discriminators and down-sampling the training data used by one mimicked using a larger receptive field for one discriminator, enabling both fine-grained detail and more global structure to be captured [24]. Predictions from both scales were then aggregated to improve performance at high resolution. This approach resulted in the removal of large incorrect recurring patterns observed when only a single discriminator was used with the original image resolution [24]. Additionally, the authors introduced a multi-resolution generator, composed of two component networks, a global and a local generator network. This approach was used to aggregate global and local information for the image synthesis task [24] and was shown to lead to improved quantitative performance compared to pix2pix GAN [24].

Multi-resolution pipelines are a well-established practice in computer vision [67] with early works suggesting two scales are often enough [68]. However, *MedGAN* [39], which explored I2I for PET-CT translation, introduced CasNet. CasNet [39] is a cascaded generator architecture that concatenates 6 pix2pix generators into a single generator network. Here, the output of the first generator is passed as the input to the second generator until the 6th generator produces the final prediction. This approach is similar to the stacked U-nets adopted by [69] and the progressive GAN framework [70, 71]. Results showed that this progressive refinement of predictions led to considerable improvements in structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR) compared to pix2pix [39].

Another line of improvements to pix2pix GANs focused on constraining the discriminator through gradient penalties [72–74] and spectral normalization [52]. Spectral

normalisation enforces a bound on the weights of each layer of the discriminator by normalising the weights by the largest singular weight matrix value (spectral norm). This helps the weights not grow too large during training, aiming to reduce mode collapse and improve training stability with little additional computational effort [45]. Following this, other works introduced self-attention blocks from [75] to produce *Self Attention GAN* (SA-GAN) [45], which used spectral normalisation with self-attention in both the discriminator and generator to help improve the modelling of global structure. Other works [46] explored methods for scaling up cGANs training to benefit from larger batches and model capacity. Results on ImageNet, a dataset containing 1.2 million RGB images from 1000 classes showed that increasing the batch size by a factor of 8 improved the state-of-the-art (SOTA) Inception Score (IS) [76] by 46% and increasing the models capacity via increasing the number of channels in each layer by 50% further improved the IS by 21% [46]. However, this increased scale came at the cost of high GPU resource requirements, environmental impact and training instability with models often requiring early stopping before training would collapse [46].

Until 2021, GANs [46] held the state of the art for image generation when measured for sample quality via commonly used metrics such as FID [62] and IS [76]. However, as previously discussed GANs are prone to mode collapse, and it has been shown in [43] that GANs capture less diversity than VQ-VAE [42], VQ-VAE2 [43] and other likelihood-based methods [28, 47]. Since the log-likelihood that explicit likelihood-based models assign to *all* training examples is minimised, likelihood-based models, in principle, cover *all* modes of the data, and do not suffer from the problems of mode collapse [35] seen in GAN samples. This presents a trade-off between distribution coverage and sample quality. As a result, much work has been done to achieve GAN-like sample quality from likelihood-based models, in particular through diffusion models.

3.1.2 Diffusion models

Diffusion models are a class of likelihood models that, when compared to GANs have been shown to produce images of high quality [27, 28] whilst also obtaining increased

distribution coverage and training stability. We discuss the core mathematical principles of diffusion models, in particular, the *conditional denoising diffusion probabilistic model* (cDDPM) in Chapter 2. The original *DDPM* [28] was also adapted to the conditional setting for the I2I task of image super-resolution, in which the aim is to learn to translate low-resolution images to a high-resolution version of the same image. *Super-resolution via repeated refinement (SR3)* [36] was shown to produce samples that achieved a higher fool rate (percentage of times a subject picks a model output over the true image) than cGAN and regression-based samples. *Palette* [8] was later introduced, which explored several image-to-image translation tasks (colourisation, inpainting, uncropping, and JPEG restoration), outperforming cGAN and regression baselines, producing diverse and high-fidelity outputs. Both Palette and SR3 required a large number of denoising steps during sampling, which meant far slower inference speed compared to cGAN sampling. *Syn-diff* [77] explored MRI-CT translation and introduced an additional discriminator network and an adversarial loss term used in the denoising process. This was the same form as [7] but the discriminator would distinguish between the noise at a time point t from the forward compared to the predicted noise model of the generator at time point t. The key difference is that the size of the step used in the reverse process was much larger with Syndiff, and therefore the total steps during inference were reduced, improving sampling speed whilst maintaining strong PSNR and SSIM in its predictions relative to cGAN baselines [77].

Improvements to DDPM were later explored by [47] as *improved denoising diffusion probabilistic model* (IDDPM) this included a faster sampling procedure, which reduced the total steps from 2k to 250 without compromising on FID and the need for an additional discriminator or adversarial loss term. However, the FID was still higher than the benchmark SOTA cGAN. In addition, despite claims in [28] that parameterising the variance schedule had no positive effect on performance, authors [47] showed it improved the likelihood results, making them closer to VQ-VAE2 [43].

Following this, authors [48] later published a set of diffusion model architecture improvements and a scheme for balancing sample quality with distribution coverage that led to diffusion models, for the first time, outperforming SOTA GANs [46] on image

generation tasks using ImageNet. In principle, the model architecture improvements were similar to the scaling up of cGANs in [46] and other improvements [45, 52] applied to pix2pix GAN [7], such as increasing the depth of the models, increasing the number of self-attention heads and using self-attention at different resolutions. The scheme in [48] was centred on using a classifier, like a discriminator in cGANs to improve the denoising process. The classifier-guided diffusion model [48] predicted the previous noisy image given the current noisy image and the class of the original image.

In practice, a classifier was pre-trained with class labels and noisy images from the same noise distribution as the diffusion model [48]. This pre-trained classifier was used during the reverse process to predict the class label given the noisy input. For a given time step, the generated image and the known class label are fed into the pre-trained classifier and the gradient of the log probability of the classifier is obtained and used to guide the prediction of the noise input in the previous timestep. Despite these improvements, classifier guidance [48] was restricted to labelled datasets, it introduced complexities into the training of the model and was viewed as improving performance because it resembled techniques used by GANs rather than purely using diffusion [78].

Classifier-free guidance [78] was later introduced which instead of using a pre-trained classifier, or mimicking techniques used by GANs, trained a single neural network which mixes estimates of noise ϵ from an unconditional denoising diffusion model and noise ϵ from a class conditional denoising diffusion model. Sampling is performed using a linear combination of the conditional and unconditional score estimates [78]. The authors' results show that boosting IS and FID metrics can be accomplished without the integration of pre-trained classifier model weights in a non-adversarial way [78]. However, due to sampling from two diffusion models (unconditional and conditional), the sampling speed was considerably slower than the single classifier-guided diffusion model.

3.1.3 Latent diffusion models

Latent diffusion models(LDMs) [79] were later introduced for performing diffusion with high-resolution images. LDMs perform diffusion within a compressed latent space rather

than the image space used in [28, 47, 48, 78]. Much like an Autoencoder, an additional encoder network is used to map the input into the latent space and then a decoder network to map back to the image space. This made the generation of high-resolution and high-fidelity samples more computationally efficient than previous [78] as the diffusion process is happening in a much lower-dimensional space [79]. Despite these improvements, LDMs have a more complex architecture, which can make the training process more challenging and require more careful optimisation compared to simpler generative models [79]. Termed foundational diffusion models for image synthesis by [80], this approach was later improved further by replacing all convolution neural networks with purely transformer-based [81] architectures [82].

3.1.4 Conclusion

Image-to-image translation (I2I) involves learning the conditional distribution $P(\mathbf{y}_i|\mathbf{x}_i)$, such as mapping bright-field microscopy to fluorescence images as explored within this thesis. We summarise the model architectures, loss functions, inference speeds and training requirements explore within this section in Table 3.1.

Model	Generator Architecture	Loss Terms	Total Params	Optimizer	Training Time (512x512)	Training Compute	Inference Speed (512x512) [s]
F-Net [4]	U-Net with 5 layers	L1 loss	~58M	Adam with LR 0.001	~1-2 days	Single GPU (V100) or greater	~0.06 s per image
cVAE [34]	U-Net with 5 layers + MLP	Reconstruction loss (L1/L2) + Kullback-Liebler divergence (KLD)	~62M	Adam with LR 0.0002	~2-3 days	Single GPU (V100) or greater	~0.06 s per image
VQ-I2I [44]	Res-Net 9 blocks + VQ	Reconstruction loss + VQ codebook loss	~13–15M	Adam with LR 0.0002	~1-2 days	Single GPU (V100) or greater	~10.00 s per image
Pix2Pix [7]	U-Net with 5 layers with instance normalisation	adversarial loss + L1 loss	~17M	Adam with LR 0.0002 and linear decay policy	~1-2 days	Single GPU (V100) or greater	~0.07 s per image
Pix2PixHD [24]	2 (local+global) U-Nets with 5 layers residual connections	adversarial loss + perceptual loss	~62–77M	Adam with LR 0.0002 and linear decay policy	~2-4 days	Multi GPU (V100) or greater	~0.08 s per image
cVAE-GAN [64]	U-Nets with 5 blocks	Reconstruction loss + KLD + adversarial loss	~50-80M	Adam with LR 0.0002	~1-2 days	Single GPU (V100) or greater	~ 0.07 s per image
MedGAN [39]	CasNet (U-Net ×3)	L1 loss + adversarial loss + perceptual loss	~45M	Adam with LR 0.0002	~3-5 days	Multi GPU (V100) or greater	~0.1–1 s per image
cDDPM	U-Net [49] (time and input conditioned)	Denoising score matching loss (MSE)	~120–150M	Adamax with LR 0.002	~5-10 days	Multi GPU (V100) or greater	~ 40-42 s per image
LDM [79]	U-Net [49](time and input conditioned) + Autoencoder	Denoising loss (in latent space) + Perceptual loss	~ 140–200M	Adam with LR 0.0002	~7-12 days	Multi GPU (RTX 6000) or greater	~40-45 s per image
SR3 [83]	U-Net + self-attention	Denoising score matching loss	~ 300M	Adam with LR 0.0002	~7-12 days	Multi GPU (RTX 6000) or greater	~45-55 s per image
Palette [8]	U-Net + self-attention	Denoising score matching loss	~ 552M	Adam with LR 0.0002	~7-10 days	Single TPU	~0.8 s per image
Syndiff [77]	U-Net + Autoencoder	Denoising score matching loss + adversarial loss	~ 140–200M	Adam with LR 0.0002	~5-10 days	Multi GPU (RTX 6000) or greater	~40-45 s per image

Table 3.1: **Comparison of image-to-image translation models core training, architecture and inference speeds** Summary of model architectures, loss functions, training times, hardware requirements, parameter sizes, and inference speeds for 512×512 resolution inputs. Details obtained where possible from original papers and GitHub repositories.

Early approaches [4, 10] used regression-based U-Nets with pixel-wise losses, but often produced blurry outputs due to the ill-posed nature of the task. Conditional generative models (CGMs) were introduced to model the full posterior distribution rather than just

its mean. Likelihood-based models like auto-regressive methods [53–55], VAEs [25, 34, 59], and normalising flows [56–58] offered theoretical grounding but initially lacked image quality or efficient sampling. Discrete latent variable models like VQ-VAE [42] and VQ-I2I [44] improved training stability and sharpness, though they still lagged behind GAN-based methods on standard quality metrics.

GAN-based models, starting with pix2pix [7], greatly improved image sharpness without compromising on inference speed, but faced issues like mode collapse and hallucinated features. Later enhancements, including pix2pixHD [24], MedGAN [39], and SA-GAN [45] focused on architectural scaling, perceptual losses [66], and attention mechanisms [81] to increase realism at higher resolution, although often at the cost of stability, training time and compute resources. More recently, diffusion models [28, 84] such as DDPM [28], SR3 [36] and Palette [8] have emerged as strong alternatives, offering better distribution coverage and training stability, albeit with slower inference. Hybrid approaches like Syndiff [36] attempt to blend the strengths of diffusion and adversarial training. Latent diffusion models (LDMs) [79] were then introduced to overcome the challenges of using diffusion with high-resolution images by performing diffusion in a latent space, reducing computational requirements without compromising on sample fidelity.

In this thesis, motivated in part by a collaboration with GSK, we focus on three representative classes of generative models: a regression-based model, a conditional GAN (cGAN), and a conditional denoising diffusion probabilistic model (cDDPM). These models are selected to balance simplicity, output realism, and the capacity to capture uncertainty, with the overarching goals of achieving high perceptual and pixel-level quality in virtual staining, while also maintaining tractable inference time.

Beginning with the regression-based model as a baseline enables a straightforward evaluation of how well a direct pixel-wise mapping from bright-field to fluorescence stained images performs. This model helps assess the learnability of the task, given the GSK dataset, without the added complexity of generative components. Then exploring a cGAN, to encourage more realistic and sharper outputs, which is particularly important in capturing high-frequency features present in fluorescence staining. Finally, I investigate a

cDDPM, a class of diffusion-based models that excels at modelling complex posteriors and producing high-fidelity results, albeit at a higher computational cost and slower inference speeds.

Rather than starting with more complex models such as diffusion models, I adopt a bottom-up strategy, beginning with simpler, interpretable approaches and progressively increasing model complexity. This is especially relevant given the variability in the quality of bright-field inputs; a better input signal often reduces the need for model complexity. This incremental approach allows me to systematically benchmark how each model class performs across criteria such as reconstruction accuracy, perceptual quality, and inference speed.

In parallel to these I2I models, I introduce an alternative learning task: bypassing virtual staining altogether by directly predicting biological features directly from bright-field images. For this, I train a U-Net using both mean squared error (MSE) and negative log-likelihood (NLL) losses to learn a mapping from images to feature space. This approach enables the extraction of biologically meaningful information without the intermediate step of generating stained images, offering a potentially faster and more direct route for downstream analysis.

3.2 Evaluation of image-to-image translation approaches

As outlined in the previous section, generative modelling approaches such as diffusion models, GANs, and variational auto-encoders have made considerable performance leaps compared to regression-based methods. Despite these achievements, the development of quantitative evaluation measurements remains less explored. Choosing the right evaluation metric based on the chosen method and application remains a challenging task.

Early regression-based approaches [4, 10] to virtual staining evaluated the intensity of virtual staining predictions using pixel/patch-based measurements between the intensity of the target fluorescence and the virtual staining prediction. These standard image quality measurements included; the *Pearson correlation coefficient* [85], the *peak signal-to-noise ratio* (PSNR) [17] and the *structural similarity index measure* (SSIM) [18]. When

quantifying performance using the Pearson correlation coefficient, the metric does not punish the scale of the predicted pixel intensity value relative to the target intensity value. This means that the Pearson coefficient could suggest a strong correlation when in fact the pixel values are very different, which within virtual staining is not a desired property. Additionally, when using these intensity measurements the uniform background pixels can positively skew the results due to high performance when translating background to background.

Our early work [86] found despite reasonably high performance for SSIM and PSNR, these scores do not give a good indication of whether or not the virtually stained channel could be used instead of the real fluorescence channel. Recent works [12] devised evaluation pipelines that provided biologically meaningful measurements of generative model performance. Such measurements were used to further explore the suitability of generative models for virtual staining in specific bioimaging applications.

More broadly, one very common measurement considered the standard evaluation approach [87] for quantifying generative modelling performance is the *average log-likelihood*, which measures the likelihood of the true data under the generated data distribution. More formally the average log-likelihood can be written as

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \log P_\theta(\mathbf{y}_i). \tag{3.1}$$

As previously highlighted, the log-likelihood for many generative models is intractable which makes it a challenging evaluation metric. For some models, evaluating generated samples is more feasible and, in principle, should provide information about the log-likelihood.

The *Parzen window* approach [22] to density estimation works by taking a finite set of samples generated by a model and then using those as the centroids of a mixture of Gaussian model [9]. The log-likelihood of the unseen examples is then computed under the mixture of Gaussian model. Even with a large number of samples, the Parzen window estimates can fall short in approximating a model's true log-likelihood when the data dimensionality is high [9]. Comparing log-likelihoods has received some criticism

because one can achieve poor log-likelihood, but high image quality such as in the case of a mixture of Gaussian model where the standard deviations are very small producing high-quality samples but a poor log-likelihood. Conversely, it is also possible to have a high log-likelihood and poor image quality as shown in [88]. These findings show that in some cases, the log-likelihood and image sample quality are moderately unrelated.

Often, generative models are formulated as density estimators whereby each image is treated as a continuous random variable. Whether using natural images stored as 8-bit integers or biological images typically stored as 16-bit integers, modelling these discrete data distributions using densities can impact log-likelihood results [89]. If the right amount of uniform noise is added, the log-likelihood of the continuous model on the now de-quantised data is closely related to the log-likelihood of a discrete model on the discrete data [89–91].

Another method for evaluating the quality of generative models is the *Inception Score* (IS) [92], which uses the Inception V3 [59] pre-trained on ImageNet [51], a dataset containing 1.2 million RGB images from 1000 classes to calculate:

$$\text{IS} = \exp \mathbb{E}_{\hat{\mathbf{y}}_i \sim P_G} \left[\text{KL}(P_\theta(c|\hat{\mathbf{y}}_i) || P_\theta(c) \right], \tag{3.2}$$

where $\hat{\mathbf{y}}_i \sim P_G$ is a generated image, $P_\theta(c|\hat{\mathbf{y}}_i)$ is the conditional class distribution computed via the inception network, and $P_\theta(c) = \int P_\theta(c|\hat{\mathbf{y}}_i) P_G(\hat{\mathbf{y}}_i) dc$ is the marginal class distribution.

As highlighted by [93] the two key qualities this metric seeks to capture are: (i) the generative model should produce a diverse range of images across all classes in ImageNet, meaning that $P_\theta(c)$ should be uniform, and (ii) the generated images should feature distinct objects, implying that the predicted probabilities $P_\theta(c|\hat{\mathbf{y}}_i)$ should resemble a one-hot vector with low entropy. When both of these qualities are met, the KL divergence between $P(c)$ and $P_\theta(c|\hat{\mathbf{y}}_i)$ is maximised. Consequently, a higher Inception Score (IS) indicates better performance [93].

When evaluating generative models using ImageNet, IS was shown to have a reasonable correlation with the quality and diversity of generated images [92]. IS was also shown to

correlate well with human scoring of the realism of generated images from the CIFAR-10 dataset [92]. However, the metric presents many limitations. Firstly, it is specific to the natural image domain and RGB images, making it unusable in many other image-generative model applications, such as biomedical imaging. More specifically, the Inception model classifies images into the 1,000 ImageNet classes none of which are relevant to biomedical tasks. Second, it requires class labels, which are costly to acquire. Thirdly, as shown in [94] the IS can be influenced by the weights from different training runs or implementations. For example, the mean Inception Score is 3.5% higher for ImageNet validation images, and 11.5% higher for CIFAR validation images, depending on whether a Keras or Torch implementation of the Inception Network is used, both of which have almost identical classification accuracy. Finally, it fails to capture when a learnt distribution has collapsed into a smaller number of modes, a common challenge faced with cGANs.

The *Frechét Inception Distance* [95] embeds data into the feature space given by a specific layer of the inception network (viewed as a multivariate Gaussian). The generated samples and real data are embedded separately, and the mean and covariance are estimated. The Frechét distance between these two Gaussians (which is the same as the Wasserstein distance) is then used to quantify the quality of generated samples

$$\mathrm{FID}(r, g) = ||\mu_r - \mu_g||_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r + \Sigma_g)^1/2\right), \tag{3.3}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the means and covariances of a mixture of Gaussian model for the real and generated samples.

In contrast to the IS, the FID presents a number of improvements. First, it does not require labelled class data because it compares the intermediate feature representations obtained from ground truth and predicted images, not just class probabilities as in IS. Second, IS can miss certain image artefacts if they do not impact the classification result while FID detects more subtle distribution differences. Finally, FID is sensitive to hallucinated, but realistic modes as well as to mode collapse, as shown in [96].

The FID has been adapted for specific tasks: Class-aware FID [97], Frechét ChemNet distance for molecules imagery [62] or the Frechét video distance for video [98].

More recently the FID has been adapted to the conditional image generation setting, the Conditional Frechét Inception Distance (cFID) [99], despite the common use of the FID for evaluating image-to-image translation which is a conditional image generation task, the measurement itself was not originally designed for the conditional setting. A good cFID score signals that the model succeeded in its two tasks: generating a diverse set of samples that fully capture the underlying output distribution; and generating realistic outputs that correspond to the correct input [99].

Both the IS, FID and the outlined variants share a crucial weakness: since they yield a one-dimensional score, they are unable to distinguish between different failure cases [100]. One solution proposed by [100] is to disentangle the divergence of the predicted distribution $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ and target distribution $P(\mathbf{y}_i|\mathbf{x}_i)$into two components: *precision and recall* which was later improved by [101].

For a given image sample the corresponding activation vector after the second fully connected layer is extracted from a pre-trained VGG-16 classifier [102]. For each feature set, a manifold is approximated by calculating pairwise Euclidean distances between all feature vectors in the set and, for each feature vector, forming a hypersphere with radius equal to the distance to its kth nearest neighbour [101]. Here precision is defined as how much of $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ manifold lies within $P(\mathbf{y}_i|\mathbf{x}_i)$ manifold while recall should measure how much of $P(\mathbf{y}_i|\mathbf{x}_i)$ manifold lies within $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ manifold. A low FID may indicate high precision (realistic images), high recall (a large amount of variation), or anything in between [101]. As observed in [101], recall was very low for certain classes implying much variation had been missed, but FID was nonetheless good for both. There are limitations evidenced in [103] such as the arbitrary picking of k, the number of samples as well as the measurements not being robust to outliers. The framework and algorithmic clarity were later improved by [103] who modelled notions of *Density and Coverage*, still using nearest neighbour-based comparison, but still lacked a formal proof for the measurements relation to population level properties [104].

All of the explored metrics metrics provides a way to assess the overall quality of a

population of generated images, in some cases it yields only a binary result for an individual sample [101, 103, 104] for the majority only a result for the entire population [22, 62, 92, 95, 97–101, 103, 104] and therefore is not suitable for ranking images and at a more granular level the instances within images by their quality.

3.3 Generalization of image-to-image translation approaches

As shown in Section 3.1 virtual staining using I2I has been widely explored using approaches based on a regression-loss [4, 10], conditional generative adversarial network (cGANs) [11, 70] and diffusion-based approaches [11, 12]. While these works have shown the significant potential of virtual staining, the ability of virtual staining models to generalise to images containing variations not present in the training data remains under-explored. In practice, HTS imaging data is highly diverse, being generated across different imaging systems, experiments, cell types, and phenotypes. This is known as the generalisation gap [105] and has been identified as a key reason for the lack of reusable virtual staining models, limiting its potential impact within large-scale applications. These challenges are analogous to those in DNA sequence modelling [106], where models are typically trained on specific cell types and fail to generalise to new cell types. In offline reinforcement learning [107–109] where datasets predominantly focus on solving the task in the same environment, limiting the evaluation of generalisation to unseen environments.

Across these methods, when tested on samples similar to those in the training sets, virtual staining predictions have been shown to consistently and reliably produce high-quality images at the pixel-level [4, 10, 11, 70] that preserve the majority of biological information found in fluorescence images [12, 86]. In order to determine the usability of virtual staining at scale, these systems need to be able to generalise under various data distribution shifts

Despite improvements in virtual staining, very few models [110, 111] are known to have been integrated into routine clinical workflows in part due to the need for regulatory approval. A recent review [105] of 161 peer-reviewed computational pathology articles identified a core reason being the generalisation gap; models failing to maintain perfor-

mance for unseen data with a shifted distribution. Nevertheless, in real-world applications, encountering unseen data is very common. For example in high-throughput screenings (HTS) [1] a single imaging machine, among many in a lab, generates imaging data for large numbers of compounds at different concentration levels tested on cell cultures of different cell types (lung or breast) composed of different cell lines (H1299, HCC827).

Early virtual staining works [4] showed models trained on images of hiPSC cells did not perform as well when generalising to HEK-293, cardiomyocytes and HT-1080 cells. Although gross image features were visually comparable, morphological detail improved when the model was trained on data of the same cell type. Meanwhile, other virtual staining works [10] showed that when trained separately on images containing cortical and motor neuron cells, generalising to a single well of a breast cell line (MDE-MD-231), sourced from a new laboratory and new transmitted-light technology, led to an increase in pixel-level performance. Similarly, authors in [12] trained virtual staining models utilising a diverse set of 290 cell types and phenotypes from the JUMP Cell Painting Dataset [112], also testing on a single hold-out plate, providing limited exploration of performance under domain shift. The question of whether virtual staining models can generalise at scale remains underexplored.

A trivial solution to generalisation would be to train virtual staining models using samples from each domain shift, but in practice, training this number of models is computationally very expensive and not scalable. Alternatively, domain adaptation methods for I2I such as DAI2I [113] and 0ST [114] have been shown to improve out-of-distribution performance on natural image datasets. None of the aforementioned approaches to bridging the generalisation gap have focused on whether certain image domains, when used as training sets, are better able to learn domain-invariant features compared to others.

Chapter 4

Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels?

Aspects of the Chapter 4 were published in 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI) [86] *with collaborations from Chih-Yang Hsu, Steve Hood, Ryan Musso, Ceridwen Hopely, Steve Titus and Minh Doan from GSK and supervisory support from my supervisors Alexander Krull, and Iain Styles from the University of Birmingham. GSK collaborators provided the dataset itself, industrial knowledge about the dataset, aspects of code used for running the experiments in GSK high-performance computing environment and the computational resources required to run my experiments. Alex and Iain provided guidance on the different image-to-image translation approaches as well as feedback for experimental design, aided in analysing results, and gave feedback on my written paper submission. I conducted all experiments, reviewed and analysed all results, and was the lead author on all paper contents.*

This Chapter is composed of two parts. In part 1, we begin by benchmarking one regression-based and three conditional generative modelling approaches to virtual nuclei staining. Each approach is then evaluated using pixel-level and nuclei-detection measurements. For the top 2 performing methods, we then explore how this performance improves as more training data is introduced. In part 2, in addition to the virtual nuclei staining task, we also explore the virtual cytoplasm and virtual DNA-damage tasks. We utilise the best-performing virtual staining approach identified in part 1. We present our evaluation pipeline and explore the extent to which virtual staining can accurately represent the biological information within individual cells found in different fluorescence channels.

4.1 cGANs outperform regression-based approaches

We begin by benchmarking four approaches to virtual staining; F-Net a regression-based method [4], pix2pix GAN [7], pix2PixHD GAN [24] two supervised generative models and CycleGAN [23] an unsupervised generative model. At the time of conducting these experiments, F-Net [4] had previously been used for virtual staining applications, pix2pix was the first GAN implementation for image-to-image translation with improved realism compared to regression-based approaches and pix2pixHD was shown to demonstrate performance gains at high-resolution compared to pix2pix. CycleGAN, was implemented to explore performance given unpaired training data. However, CycleGAN was unable to generate virtual staining predictions that were visually similar to the ground truth so we only report the average scores without any visualisations for this method.

Dataset & Training: A subset of the already mentioned dataset in 1.6 was used. This included 7,000 bright-field and nuclei stain image pairs randomly sampled across 50 plates and 6 lung cell lines. The data was split 70% for training, 10% for validation and 20% for test. Hyperparameters such as the number of layers, total parameters, optimizer and learning rates can be found in Table 3.1

Evaluation: Each method was evaluated using image quality metrics SSIM [18] (with a patch size of 11x11) and PSNR [17] and for nuclei segmentation using Jaccard index [115] and F1 score [116]. Nuclei masks were obtained for the fluorescence stain and

the virtual nuclei stains using Cellpose's [117] pre-trained nuclei segmentation model. All stains were pre-processed with a gamma correction of 0.3 before the Cellpose mask was generated. This was identified to positively improve the masks generated by Cellpose during a process of evaluating the quality of the Cellpose [117] masks for the fluorescence stains. After the gamma correction, the Cellpose masks were verified and had an average acceptance rate of 97%. Please see Appendix X for the full analysis that led to this outcome.

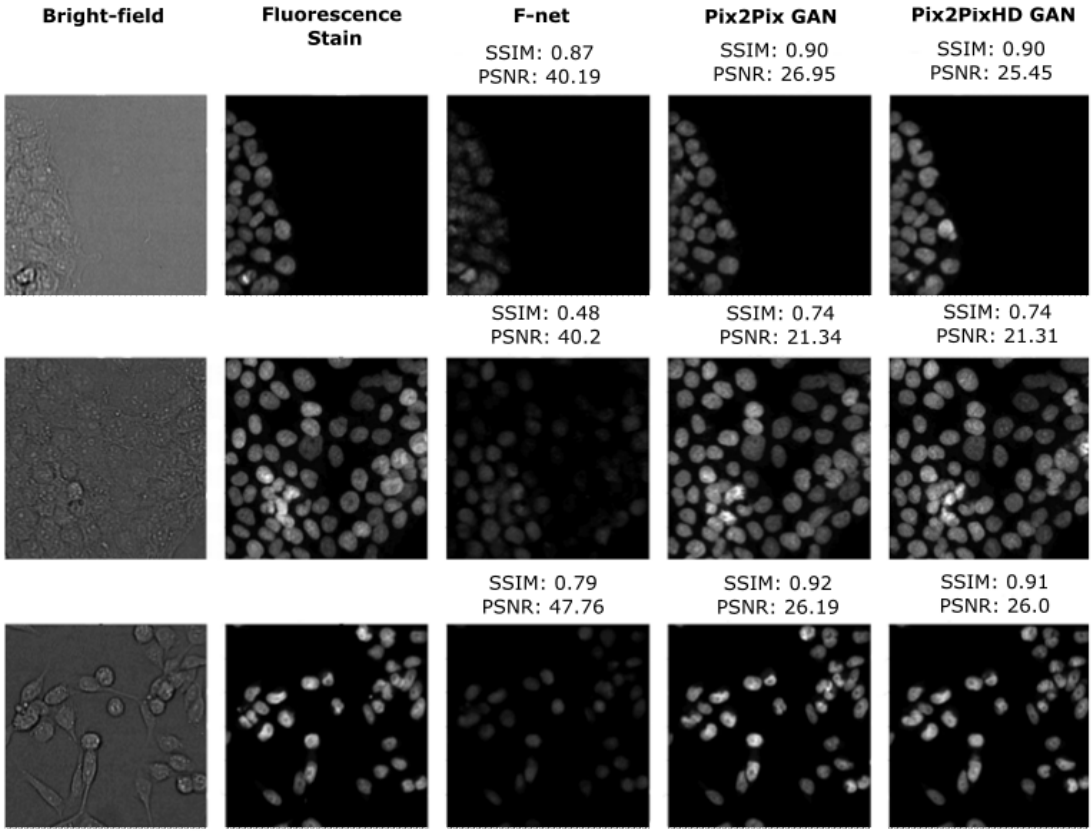


Figure 4.1: **Ground truth fluorescence vs. virtual nuclei stain for F-net, Pix2pix and Pix2PixHD.** SSIM and PSNR for the 256x256 region of interest (ROI) are reported. The deterministic F-net prediction, despite having consistently high PSNR, shows clear differences to the fluorescence stain, including missing nuclei, and the predicted nuclei appear blurry. Meanwhile, samples from both GAN-based methods appear sharp, and the general shape of the nuclei are reproduced well.

Upon visual inspection of Figure 4.1 we observe for pix2pix and pix2pixHD, the virtual stains appear to localise a large proportion of nuclei but exhibit differences in cellular morphology and intensity values compared to the fluorescence stain target. In

contrast, the virtual stains produced by F-Net appear blurry and lack the sharpness and diversity of intracellular structure observed from pix2pix and pix2pixHD, supporting previous findings [4]. The observations for pix2pix and pix2pixHD are verified by the SSIM and PSNR shown for each of the three virtual stains. However, when looking at the PSNR results for F-net one can observe very high scores despite visually clear signs that information has been lost. One might interpret this as incorrect or misaligned but the high PSNR is due to the regression loss being the mean-squared error (MSE) as outlined in Chapter 1 which forms a component of PSNR. The model has learned to minimize the MSE but the averaging of pixels is causing the lack of clear nuclei boundaries.

Figure 4.2 shows the nuclei segmentation performance of all three methods with Jaccard Index and F1 score results. Despite the blurry predictions of F-Net, Cellpose is able to predict a small subset of nuclei. However, in comparison to pix2pix and pix2pixHD, significantly fewer nuclei are translated demonstrated by the high amount of false negatives shown in red exhibited in the nuclei mask errors. Meanwhile, pix2pix and pix2pixHD achieved similar results with a large proportion of nuclei being accurately predicted. The error plots of Figure 4.2 show that for pix2pix and pix2pixHD, a small proportion of nuclei are completely missed. Looking at the bright-field input it is clear that this is caused by an artifact in the bright-field image which obscures the nuclei. These artefacts are not uncommon in high-throughput screening and as shown can make it very challenging to predict the virtual staining. It can also be seen in blue false positive nuclei shapes for both pix2pix and pix2pixHD where despite the edges looking sharp and realistic they are different from the target fluorescence stain.

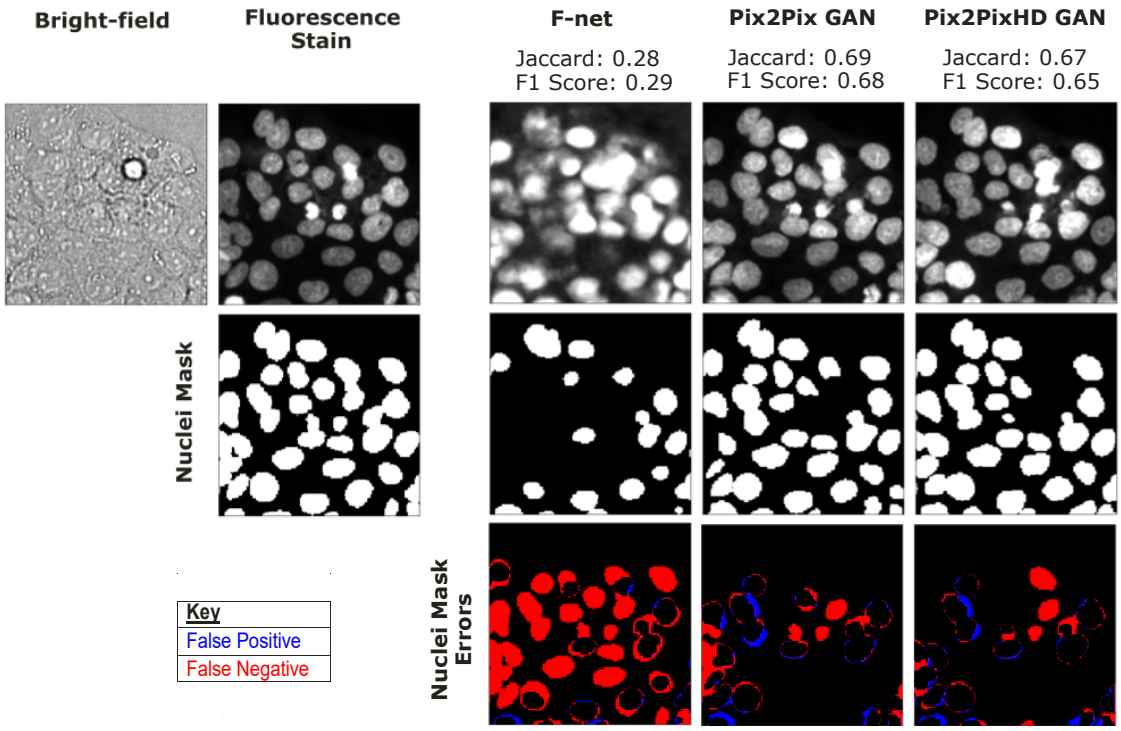


Figure 4.2: **Ground truth mask vs. virtual nuclei mask for F-net, Pix2pix and Pix2pixHD.** Nuclei masks obtained from Cellpose [117] pre-trained nuclei segmentation model for the ground truth fluorescence and each virtual nuclei stain. The systematic review of Cellpose pretrained nuclei model can be found in Appendix A. Jaccard index and F1 score for the 256x256 region of interest are reported along with segmentation errors. An intersection over union (IoU) threshold of 0.7 was used to determine a true positive, false positive (blue), and false negative (red) predicted pixels are highlighted. The mask obtained from the F-net prediction shows the highest level of false negatives, with multiple nuclei missing in part due to the blurry nature of the virtual stain. Meanwhile, both GAN-based methods show similar decreased levels of false negatives compared to F-net but in contrast show hallucinations of areas of predicted nuclei not present in the fluorescence stain.

Table 4.1 shows the average performance for all methods across the 1,400 test set images. The highest SSIM was 0.94 ± 0.05 equally achieved by pix2pix and pix2pixHD. For all other metrics pix2pix achieved the highest performance with a PSNR of 34.6 ± 8.8, F1 score of 0.73 ± 0.18, Jaccard index of 0.72 ± 0.11 F-Net scored on average the lowest for all metrics out of the three supervised methods with an SSIM of 0.7 ± 0.01, PSNR of 30.0 ± 5.8, F1 score of 0.18 ± 0.15, Jaccard index of 0.37 ± 0.19. The full list of results can be seen in Table 4.1. Consistently, when evaluating virtual staining using pixel/patch-based and instance-based metrics, we observed the superiority of GAN-based methods compared to regression-based approaches. Both pix2pix and pix2pixHD produced virtual staining

Method	SSIM	PSNR	Jaccard Index	F1 Score
F-Net	0.50 ± 0.01	30 ± 5.8	0.37 ± 0.19	0.18 ± 0.15
Pix2pix GAN	**0.94 ± 0.05**	**34.8 ± 3.8**	**0.72 ± 0.18**	**0.73 ± 0.18**
Pix2pixHD GAN	**0.94 ± 0.05**	34.5 ± 4	0.71 ± 0.12	0.70 ± 0.18
CycleGAN	0.18 ± 0.06	14.5 ± 2.0	0.28 ± 0.1	0.27 ± 0.21

Table 4.1: **Average scores on 1,400 test set images for Fnet, pix2pix GAN, Pix2pixHD GAN and CycleGAN.** Across all metrics we observe Pix2pix GAN achieves the highest performance while Pix2pixHD GAN achieves equal SSIM and very similar performance across all other metrics. F-net (regression-based) shows considerably lower performance across all metrics except for PSNR, where the performance is lower but still good due to the overlap between the mean squared error (MSE) term in PSNR and the MSE regression loss. CycleGAN shows poor performance across all metrics due to the lack of supervision.

predictions with greater pixel-wise sharpness and realism compared to the regression-based approach which led to higher SSIM and PSNR results. When evaluating virtual staining predictions at the instance level both GAN-based predictions enabled Cellpose to consistently localize the large majority of nuclei which when compared to the masks obtained from the real fluorescence image led to higher average F1 scores compared to the regression-based approach. An additional observation for the GAN-based methods is that despite at the pixel level, certain nuclei appearing sharp and realistic when the Jaccard Index and F1 score were computed hallucinations were observed in Figure 4.2. This suggests further analysis is required around the quantification of uncertainty of the GAN-based method predictions.

In order, to determine if virtual staining is a suitable alternative imaging method for high-throughput screening, We also explored the performance of GAN-based methods given access to limited dataset sizes. As most ML applications in the sciences require dedicated experimental data generation, we wanted to show what the performance expectation could be given access to a range of paired sample sets. Furthermore, we wanted to explore whether pix2pix and pix2pixHD GAN performance would increase as we scaled the training set size from 3,600 up to 15,000 paired samples. To explore this we randomly sampled 9 training sets; containing 70, 140, 280, 560, 1120, 2240, 4,900, 10,500 and 15,000 bright-field and nuclei stain pairs from lung cell lines. Both pix2pix GAN and pix2pixHD GAN were then trained independently on all 9 training sets and evaluated on

the previously used 1,400 test set images.

Figure 4.3 shows the SSIM, PSNR, Jaccard index, and F1 score for pix2pix and pix2pixHD on the virtual nuclei staining with each of the 9 training set sizes. Pix2pix outperforms pix2pixHD with training set sizes less than 4,900 across all metrics but as the training set size increases to 10,500 plus pix2pixHD outperforms pix2pix across all metrics while pix2pix performance begins to plateau. We observed significant performance gains from training a model with training sets in the 100s compared to training sets in the 1000s as the training set increases beyond 10,00 the performance gains become much smaller.

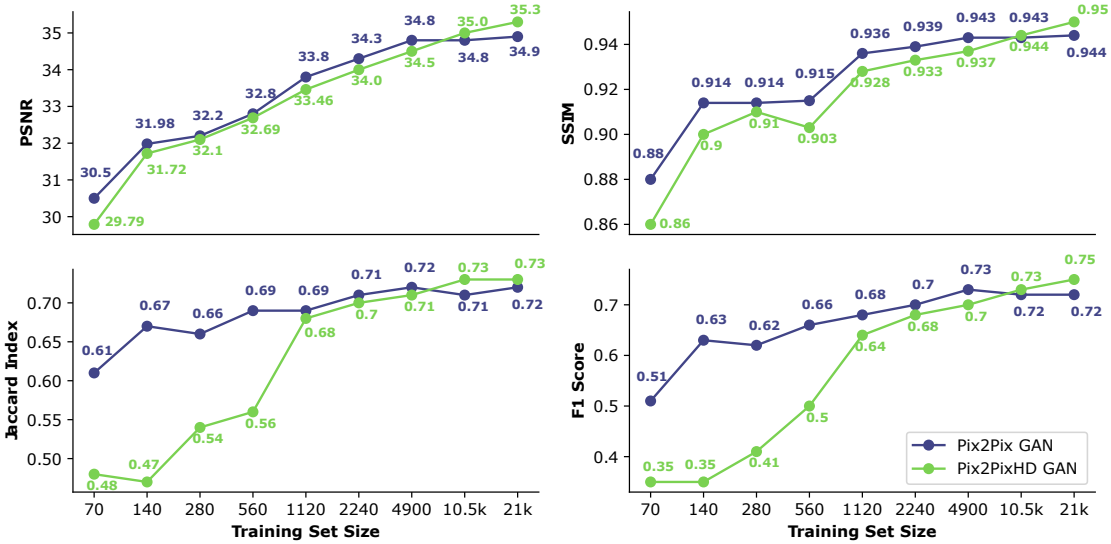


Figure 4.3: **Average SSIM, PSNR, Jaccard Index and F1 Score results when training Pix2pix and Pix2pixHD on different dataset sizes and evaluating on the same test set of 1,400 images.** Pix2pix consistently outperforms Pix2pixHD with training set sizes of less than 4,900 across all metrics. As the training set size increases to 10,500 and 21,000 Pix2pixHD begins to outperform Pix2pix across all metrics and shows an upward pattern of performance improvement, while Pix2pix performance begins to plateau.

4.2 Virtually stained nuclei and cytoplasm images often consistently and faithfully reproduce the information found in fluorescence microscopy, virtually stained images of DNA-damage are usually less accurate

Here, we will for the first time, systematically investigate which biological features are consistently reproduced accurately in virtual staining via pix2pixHD and which are likely to be lost, opening the door to potential wide-spread future applications in drug discovery.

To achieve robust quantitative results, we leverage the GSK HTS dataset containing more than 1.6 million individual high-resolution images from six lung, six ovarian, and six breast cell lines consisting of paired bright-field and fluorescent channels showing cytoplasm, nuclei and DNA-damage.

Figure 4.4 shows the first version of our evaluation pipeline. Level 1, considers the established pixel/patch-level metrics (SSIM [18], PSNR [17]) and examines the variability in virtual staining predictions. Level 3, uses the software Columbus [21] to compute 50 morphological, textural and intensity features that are frequently used in HTS. The results are compared between the real fluorescence channels and those obtained for different virtual staining. Finally, in level 3, a common HTS assay quality metric, robust Z prime (RZ') factor [118] is computed on the feature values obtained from real and virtually stained images.

Dataset: Our experiments are based on the same pool of 1,631,232 individual images generated as part of a GSK HTS assay. The data comprises 118 16x24 well plates each of which contains negative and positive controls along with 10 GSK compounds screened for toxicity with one of three cell types; lung, ovarian and breast. Every well consists of 9 fields of view each containing a bright-field and three co-registered fluorescent stains; fluorescein (FITC) for cytoplasm, 6-diamidino-2-phenylindole (DAPI) for nuclei detection and Cyanine (Cy5) for DNA-damage. Each cell type was represented by six different cell lines.

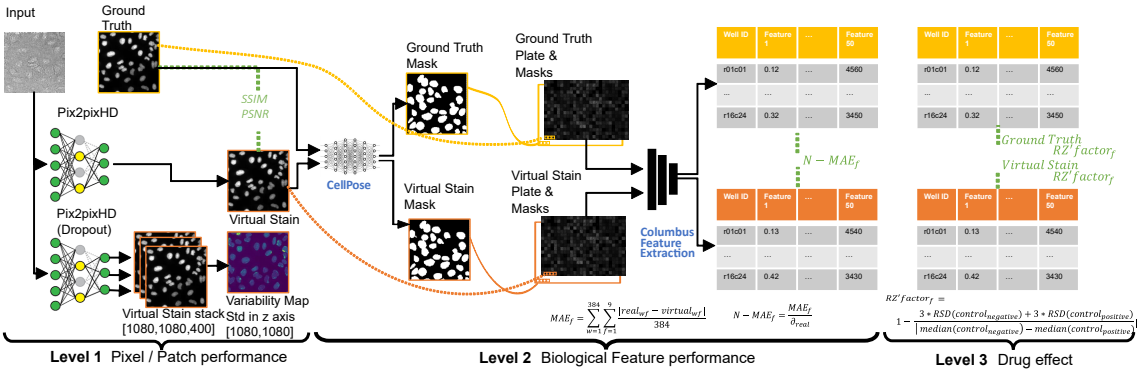


Figure 4.4: **Virtual staining evaluation pipeline Version 1.**

Level 1. Pixel / Patch Performance: SSIM and PSNR are calculated between the virtual staining and fluorescent stains. One way to determine the level of confidence a GAN has in its prediction is to sample multiple solutions via MC-dropout [119] during inference. We use this technique to produce 400 solutions on which the pixel-wise standard deviation was calculated to visualise prediction variability, i.e., lack of confidence.

Level 2. Biological Feature Performance: Nuclei segmentation masks are generated from the real and virtual nuclei channels using Cellpose [117]. To obtain high-quality segmentations, we apply a gamma correction of 0.3 before feeding the images to Cellpose. The value of 0.3 was validated by manually checking a subset of random nuclei and cytoplasm fluorescence images and the Cellpose segmentation masks. The ground truth and virtually stained plates with instance masks are used as input to Columbus [21] feature extraction pipeline to generate scores per image for 50 biological features. The MAE between fluorescence feature scores and each virtual staining stage are computed and then normalised by the standard deviation found in the ground truth feature scores to give a normalise- MAE (N-MAE).

Level 3. Drug Effect: Ground truth and virtually stained feature scores are used to generate feature-wise RZ' factors [118]. This method compares for each feature the distribution in negative and positive control sections of the plate, producing a value in the range of $(-\infty \leq RZ' \leq 1)$. A value close to 1 indicates the feature distributions are very dissimilar, and this feature provides a potentially strong measurement for cell damage. We compute the change in RZ' factor for each feature between the values obtained from the fluorescence stain and each virtual stain.

Training: For each cell type, three models were independently trained to translate from bright-field to fluorescent stain. For each cell type, 30,000 bright-field and fluorescence stain image pairs were sampled and split 70% for train, 20% for validation and 10% for testing. In addition, for each cell type, a single plate, excluded from any training or validation sets, was used as a test set for levels 2 and 3 of the analysis. This work used the pix2pixHD [24] architecture adapted for our image resolution via rescaling. The same hyper-parameters as [24] were used, after tuning we found 3 discriminators instead of the default 2 and a batch size of 16 achieved the best performance on the validation set. Each model was trained for a maximum of 100 epochs using early stopping.

4.2.1 Level 1 - Visual Inspection & Pixel/Patch-level Performance

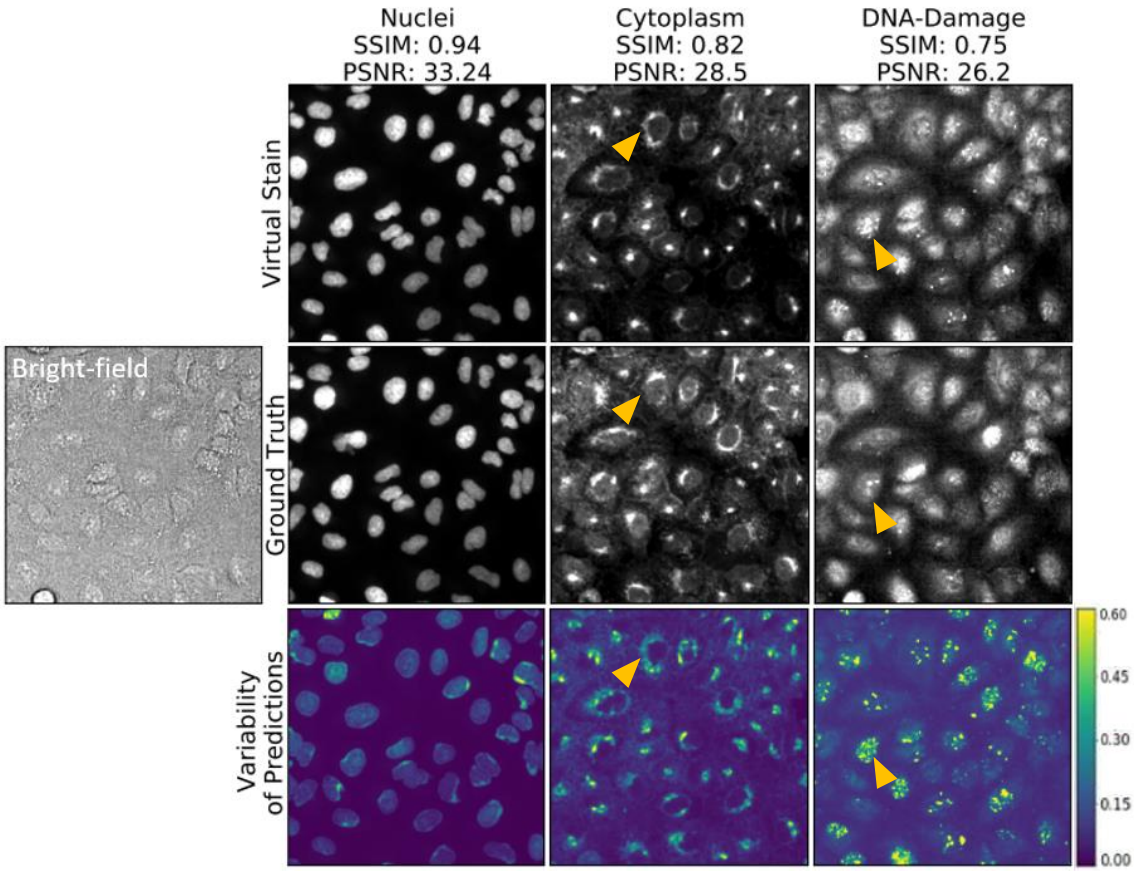


Figure 4.5: **Ground truth fluorescence vs. virtual stains for a high cell density region of interest (ROI) in a lung cell line.** SSIM and PSNR for the ROI as well as prediction variability maps (VMAP), computed as the pixel-wise standard deviation of multiple predictions generated from the same bright-field input using dropout [120]. Each VMAP is normalised by the standard deviation of the ground truth fluorescence image. The general shape of nuclei and cells are reproduced well. Despite virtual staining appearing plausible, certain biological features, DNA-damage spots and cytoplasm intensity show clear differences and subsequent high prediction variability. Examples are shown at yellow arrowheads.

Upon visual inspection (Figure 4.5), the general shape of the nuclei and cytoplasm are represented well in the virtual stains. However, aspects such as the position of DNA-damage spots or the intensity of the cytoplasm are visually plausible but incorrect. This is because the required information is simply not contained in the bright-field channel. One way to determine the level of confidence a cGAN has in its prediction is to sample multiple solutions via dropout [119] during inference. This technique was used to produce 400 solutions on which the pixel-wise standard deviation was calculated to visualize prediction variability, i.e., lack of confidence. Figure 4.5 shows the highest variability in the DNA-

Damage spots and areas of high intensity in the cytoplasm channel suggesting high levels of uncertainty.

Our first step to quantitatively evaluate our virtual staining is to use pixel/patch-level metrics. In contrast to previous virtual staining work [4, 10] that evaluated its predictions using the Pearson correlation between the real and virtual pixel intensities, two well-established image quality metrics were used to compare our virtually stained images to the real fluorescence stains.

PSNR [17] was selected as it is sensitive to shift and scale of pixel intensities and SSIM [18] because it is invariant in that respect and it. Across all cell types, virtual nuclei stain and virtual cytoplasm achieved very similar average SSIM scores of 0.96 ± 0.02 and 0.95 ± 0.02 while virtual DNA-damage scored 0.87 ± 0.10. Similarly, for PSNR, virtual nuclei and cytoplasm achieved an average of 35.2 ± 1.6 and 35.1 ± 2.8, respectively, while virtual DNA damage achieved 34.9 ± 3.1. Compared to the results from image restoration [121] these numbers indicate good quality. However, microscopy images contain large areas of uniform background leading to overstated pixel-level quality. Both metrics dramatically drop when considering a smaller region of interest with less background (Figure 4.5).

These results show that virtual nuclei and virtual cytoplasm are reproduced more accurately compared to the virtual DNA-damage. Despite appearing visually plausible and achieving relatively high performance for SSIM and PSNR, it can be seen that certain biological structures in this channel are reproduced with low confidence and differ from the real fluorescence channels.

4.2.2 Level 2 - Biological Feature Representation

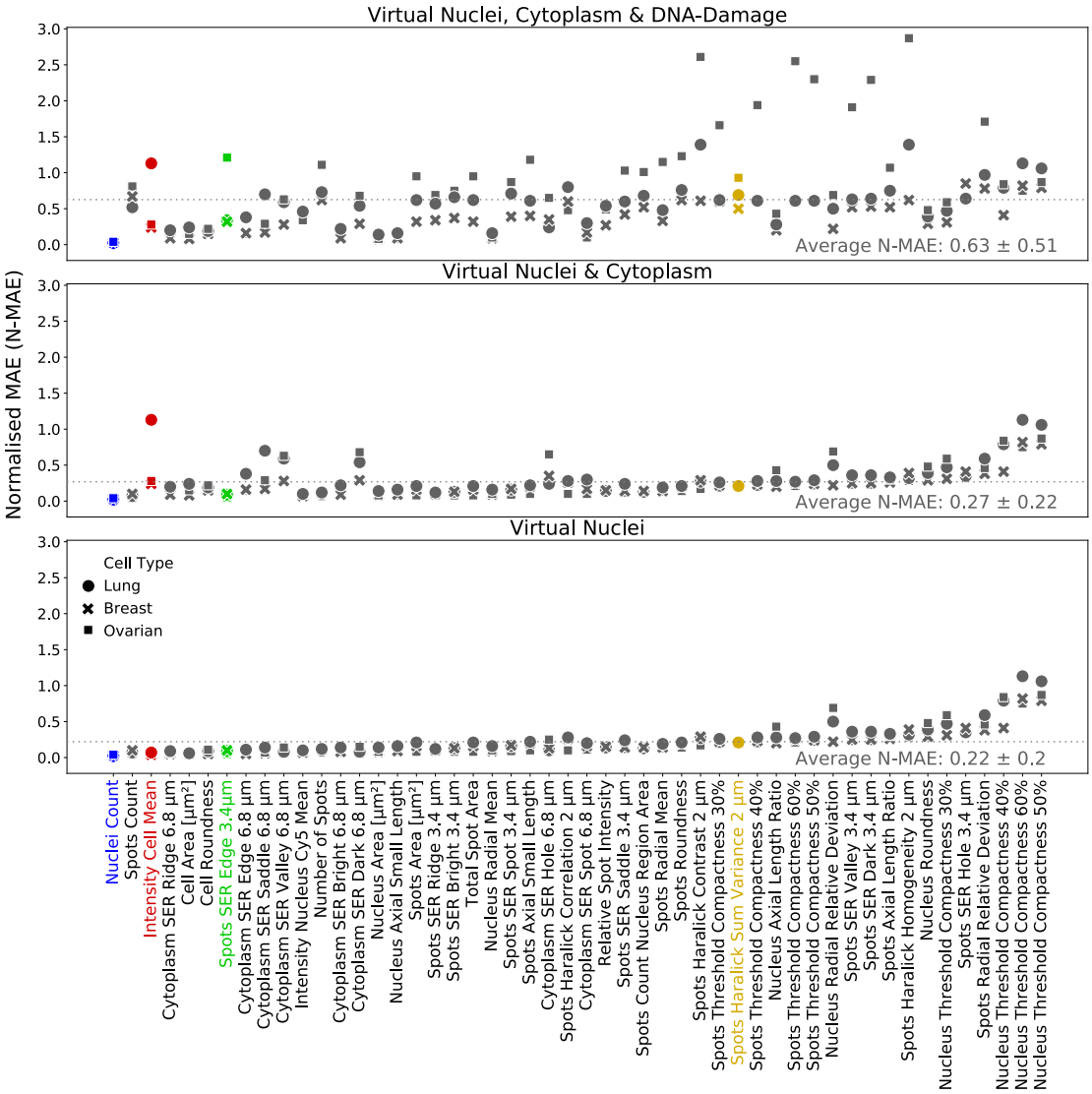


Figure 4.6: **Normalised MAE (N-MAE) between fluorescence feature scores and each virtual staining stage for 50 biological features across all three cell types.** The nuclei channel feature distributions appear very similar, the cytoplasm feature distributions are moderately different when the virtual cytoplasm is introduced, meanwhile the two remaining DNA-damage spot features show considerable differences when the virtual DNA-damage channel is used.

In HTS, the toxicity of compounds is often evaluated by computing quantitative morphological, textual and intensity features for each cell and comparing their statistics to positive and negative controls. For I2I to be useful for HTS it must faithfully translate biological features of interest. As shown in Figure 4.4 Columbus [21] was utilized for the feature extraction pipeline, designed for the original HTS assay to see if these feature statistics are reproduced in our virtually stained images. The Columbus pipeline takes a

complete HTS plate as input. Here, a held-out plate of each cell type that was not used for training or validation was used. The feature extraction begins with the segmentation of the nuclei, which then enables the segmentation of the cytoplasm and DNA-damage spots. Based on these segmentation outputs the scores for 50 features are then computed from their respective channel, see Figure 4.6. The feature score is computed per well by first quantifying the score across all cells per field of view and then averaging across all nine fields of view for all features.

To evaluate our virtual staining, we first generated feature scores from the real fluorescence channels and then introduced the virtual staining in 3 stages; stage 1 involved virtual nuclei only, stage 2 virtual nuclei and virtual cytoplasm and in stage 3 all virtually stained channels were used. For each cell type, feature and stage, the mean absolute error (MAE) between the real fluorescence and virtual staining feature scores was computed over all wells. To enable the comparison and interpretation of MAE across different feature ranges the resulting MAE for each feature was normalised (N-MAE) by the standard deviation of the real fluorescence feature score.

Stage 1 with virtual nuclei only, achieved an average N-MAE score of 0.22 ± 0.2. Remarkably, a large proportion of features have produced stable N-MAE across the different cell types. As highlighted in Figure 4.6 the "Nuclei Count" feature from the nuclei channel, consistently scores the lowest N-MAE across all cell types, verifying the qualitative observations of the virtual nuclei channel in Figure 4.5. The largest errors are observed for the nuclei channel features associated with "Nucleus Threshold Compactness %", which measures nuclei shape at different intensity thresholds. This finding aligns with the uncertainty in shape shown for the nuclei channel in Figure 4.5.

In stage 2, as the virtual cytoplasm is introduced the average N-MAE rises to 0.27 ± 0.22. The increase in error is predominantly driven by features that rely on cytoplasm intensity or cell shape. Figure 4.6 shows an intensity-based feature, "Intensity Cell Mean" which, is not reproduced well for all cell types and in particular for the lung. The problem of incorrect cell intensity was also visible in Figure 4.5. Finally, in stage 3, when all virtually stained channels are used a dramatic increase in error for many features extracted from

the DNA-damage channel can be observed, resulting in the average N-MAE increasing to 0.63 ± 0.51. In Figure 4.6, two examples of this are highlighted; "Spots Haralick Sum Variance 2µm" and "Spots SER Edge 3.4µm" which are both measurements for the texture of DNA-damage spots.

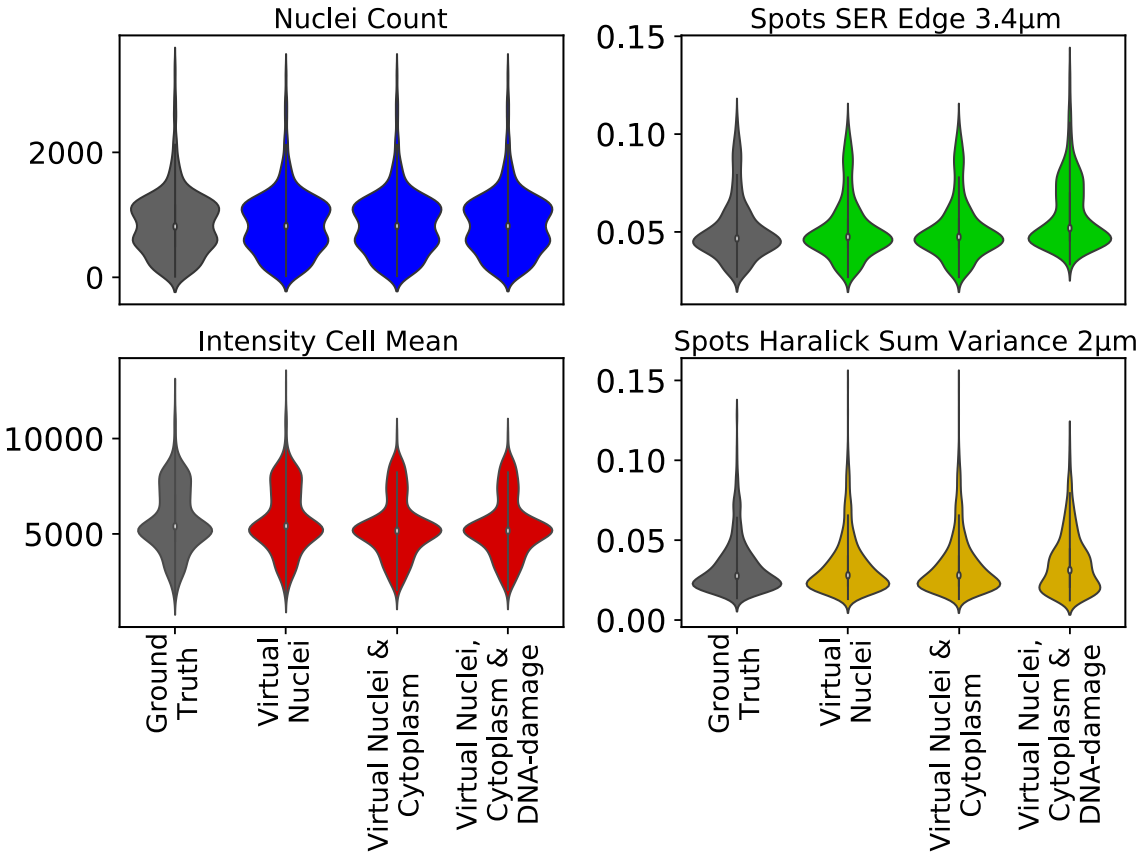


Figure 4.7: **Distribution of feature scores highlighted for fluorescence and virtual staining stages.** On average, the virtual nuclei and virtual cytoplasm achieved lower N-MAE compared to virtual DNA-damage. One nuclei, one cytoplasm and two DNA-damage spot features are highlighted.

For each of the features highlighted in Figure 4.6, it can be seen in Figure 4.7 the distribution of values at each virtual staining stage compared to the distribution of the real fluorescence values. For the nuclei channel feature, the virtual nuclei distribution is almost indistinguishable from the real fluorescence distribution and remains unchanged throughout. The distributions of feature scores for the cytoplasm and DNA-damage features are visually impacted when the corresponding virtual channel is introduced suggesting that any measurements based on them are no longer reliable.

4.2.3 Level 3 - RZ' Factor

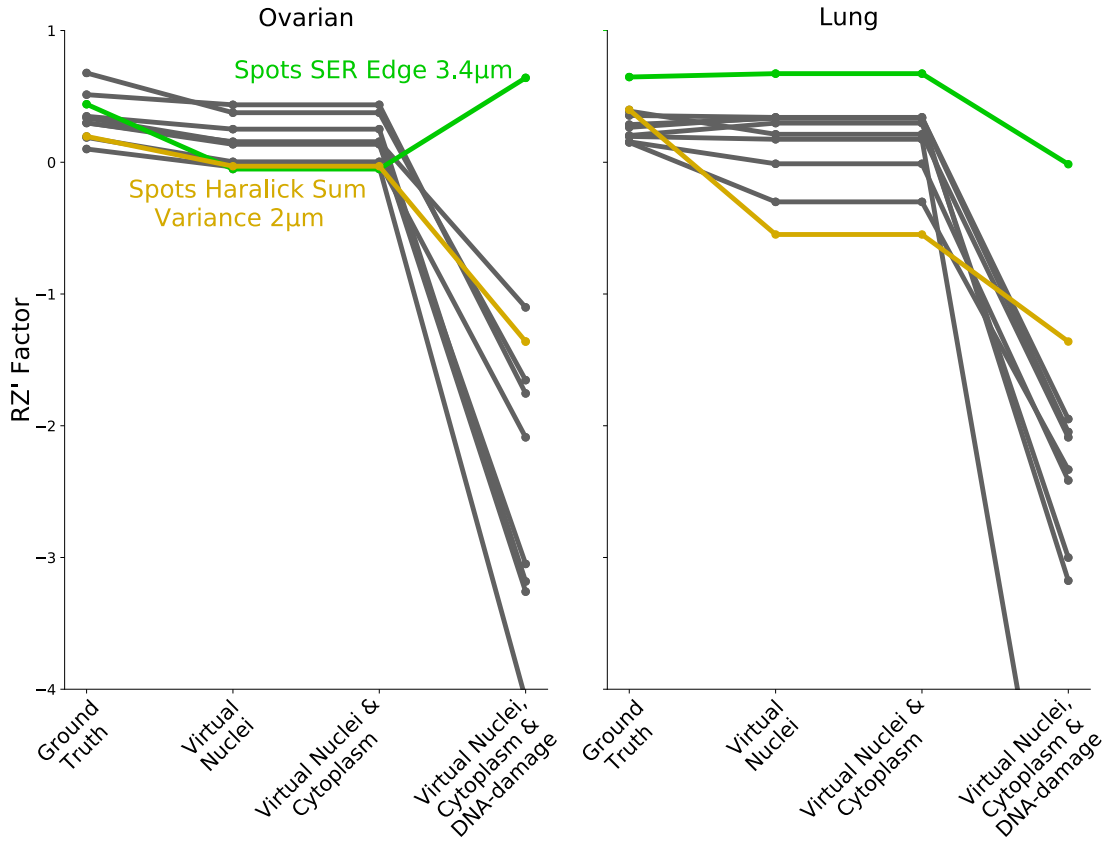


Figure 4.8: **Highest 10 RZ' factor obtained with fluorescence and the change in RZ' factor with the progressive virtually staining stages for lung and ovarian plates.** For both ovarian and lung, we begin on the left-hand side with the highest RZ' factor scores obtained from the ground truth images. We then plot as a line for each virtual staining stage, the change in RZ' score. In the perfect scenario, there would be no change and we would observe a straight line. However, for virtual nuclei and cytoplasm, we observe small changes in RZ' factor. But, as the virtual DNA-damage is introduced the RZ' factor for the majority of features plummets, indicating that these features are no longer usable for HTS.

In HTS assay screening pipelines for cell damage, after biological features of each plate have been extracted, the plate quality is checked using the RZ' factor [118].The method compares the feature distributions in negative and positive control sections of the plate. The RZ' factor is a value in the range of $(-\infty \leq RZ' \leq 1)$ that is computed for each feature. A value close to 1 indicates the feature distributions are very dissimilar and that this feature provides a potentially strong measurement for cell damage. Often an HTS study will focus on the features with the highest RZ' factors. A value close to 0 or negative suggests that the feature distributions are highly similar and therefore not suitable

for measuring cell damage.

The RZ' factors is computer and the ten features with the highest scores from the real fluorescence images are selected to compare with the RZ' factors from all stages of virtual staining. For the breast plate analysed, all RZ' factors were negative, indicating that they are not suitable for HTS and as such they were excluded from further analysis.

For the lung and ovarian plates the highest 10 RZ' factors all rely on the DNA-damage channel. Figure 4.8 shows remarkably small changes occur for many features when the virtual nuclei and virtual cytoplasm channels are used. However, the introduction of virtual DNA-damage leads to a substantial reduction in the majority of RZ' factors, indicating that they are no longer usable in the toxicity study. We again highlight "Spots Haralick Sum Variance 2µm" in Figure 4.8 as an example, where the drop in RZ' factor aligns with the increases in N-MAE shown in Figure 4.6 and the distorted distribution shown in Figure 4.7. The same pattern can be seen for the "Spots SER Edge 3.4µm" feature in the lung cell type. However, unexpectedly, an increased RZ' factor in the ovarian cell line as the virtual DNA-damage is introduced can be observed. One might naively interpret this as an improvement introduced by virtual staining. However, considering the increased error for ovarian and the deformed distribution in Figure 4.7 We believe that this simply reflects inaccuracies in the translation that are systematically different in the positive and negative controls.

4.2.4 Discussion

Here, we have investigated I2I as a technique to extract information from the bright-field usually requiring fluorescence staining in HTS. Remarkably, across all three levels of our analysis, both virtual nuclei and virtual cytoplasm have consistently performed well; achieving high pixel/patch-level quality, producing similar scores for the majority of features found in the real fluorescence channels and leading to relatively small changes in RZ' factor outcomes compared to the fluorescence. As such, we believe that the use of virtual nuclei and virtual cytoplasm stains as a non-invasive imaging method could become a viable alternative to nuclei and cytoplasm fluorescence staining. These findings suggest

a potential reduction in imaging time for the two labels, as high-throughput screening with virtual staining eliminates the need for multiple plates across incubation periods, requiring only a single plate, and generates stained samples within minutes rather than days. The removal of these stains from fluorescence staining could also enable new multiplex imaging combinations. Using only a single plate with no fluorescence staining enables the use of live-cell imaging and removes plate effects, improving the consistency and physiological relevance of experimental results. In addition, these findings suggest that virtual nuclei and cytoplasm staining could be used as a low-cost pre-requisite stage before a large-scale HTS study is conducted to give a preliminary indication of biological feature representation and drug effect outcomes.

In contrast, the virtual DNA-damage channel has consistently shown relatively higher errors across all levels of analysis and cell types. In particular, the DNA-damage spots showed high variability and considerable losses of information for DNA-damage features such as "Spots SER Edge 3.4µm" and large changes in RZ' factor compared to the other two virtual channels. We believe it is not possible to predict the precise spot location because this information is not present in the bright-field image. Despite this, features such as "Spot Count" still achieved a low N-MAE for certain cell types as they are invariant to the precise spot location. One limitation of this component of our analysis is its inability to evaluate N-MAE per instance instead of averaging over all instances per well which would help improve the accuracy with which we evaluate virtual staining.

We saw that the use of the virtual DNA-damage channel could lead to an improved RZ' factor for individual features even when the error of the feature values had increased. This is unexpected because an increased RZ' factor indicates a potentially improved ability to distinguish between negative and positive controls which should be negatively correlated with increased errors of feature values. Ultimately, the positive correlation arises from incorrect I2I and its applicability should be subject to further scrutiny.

Chapter 5

Can virtual staining for high-throughput screening generalise?

Aspects of Chapter 5 were published to Arxiv [122] *and the Bioimaging Computing Workshop at ECCV 2024 with collaborations from Cuong Nguyen, Steve Hood, Ryan Musso, Ceridwen Hopely, Steve Titus and Minh Doan from GSK and supervisory support from my supervisors Alexander Krull, and Iain Styles from the University of Birmingham. GSK collaborators provided the dataset itself, industrial knowledge about the dataset, aspects of code used for running the experiments in GSK high-performance computing environment, and the computational resources required to run my experiments. Alex and Iain provided guidance on the experimental design, aided in analysing results, and gave feedback on my written paper submission. I conducted all experiments, reviewed and analysed all results, and was the lead author on all paper contents.*

The findings in Chapter 4 revealed that the use of virtual nuclei and virtual cytoplasm stains as a non-invasive imaging method could become a viable alternative to nuclei and cytoplasm fluorescence staining. While these findings and other recent works [11, 12, 19, 86, 123] have shown promising results they have all been *limited* to evaluating virtual staining performance on images sampled from the same distribution as the images the model was trained on. Within HTS, in order for virtual staining to be considered a viable image alternative to fluorescence microscopy virtual staining models must be able

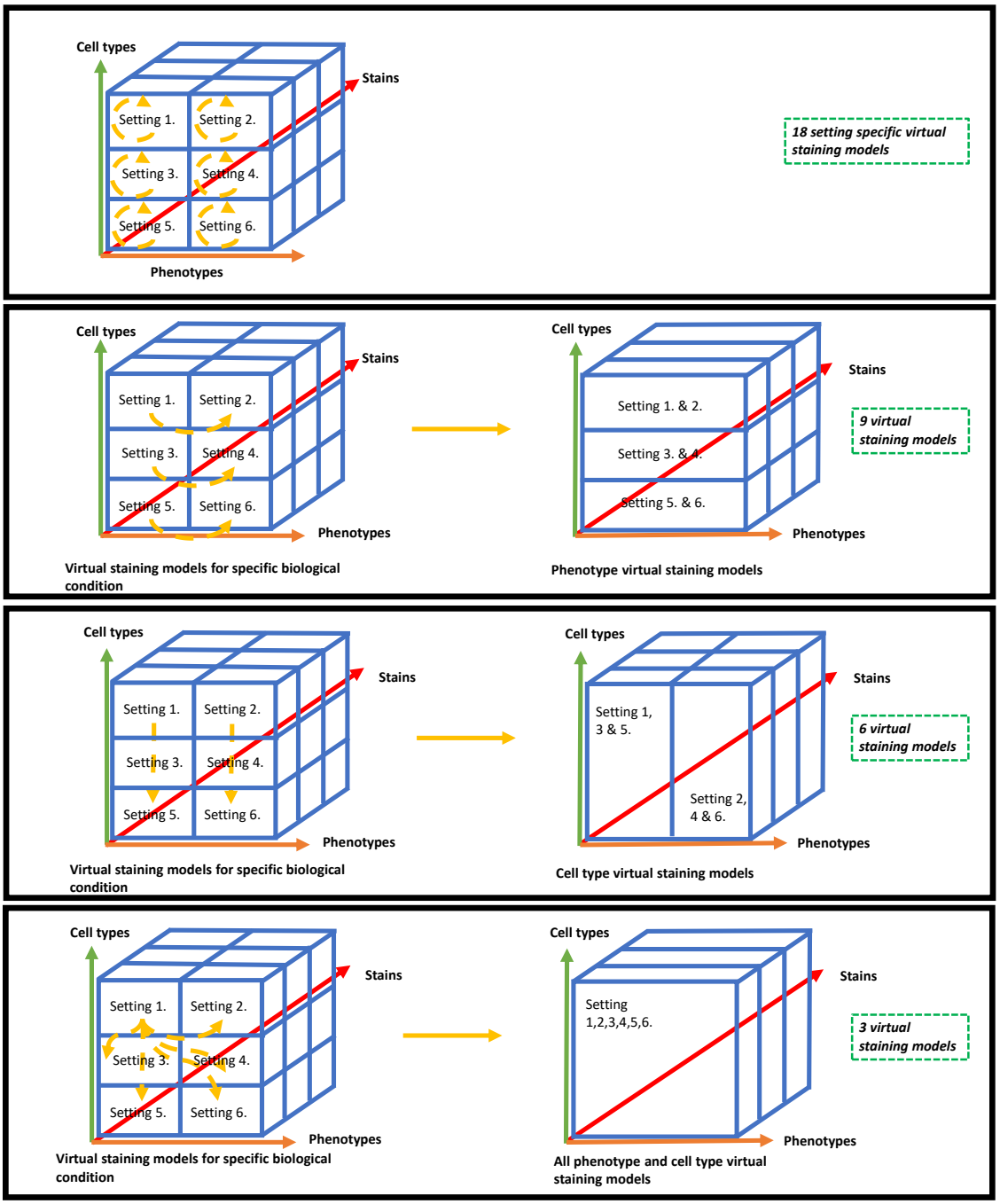


Figure 5.1: **Visualisation of total virtual staining models required to train and maintain when using only 2 dimensions of possible biological settings (cell type and phenotype) with 3 cell types and 2 phenotypes across 3 staining tasks.** Training and maintaining setting specific virtual staining models is not scalable. Exploring generalisation and whether subsets of HTS can lead to more generalisable models can drastically reduce the number of virtual staining models required as shown in rows 2-4. However, little is known about whether subsets of HTS data can lead to more generalisable models.

to generalise to unseen data from different distributions from that of its training data. Without generalisable virtual staining models the question of whether virtual staining is suitable for HTS remains in doubt.

Within this Chapter, we provide the first large-scale, data-centric approach to determine the extent to which virtual staining can generalise under three common data distribution shifts within HTS.

The results of this analysis serve a three-fold purpose. First, as most ML applications in the sciences require dedicated experimental data generation, this Chapter would like to guide the best way to generate data that would produce maximally generalisable models. Second, by systematically exploring if certain biological or technical setting specific training sets lead to more generaliable models we can drastically reduce the number of virtual staining models required to train and maintain (as shown in Figure 5.1). Finally, as scientists need to extract biological insights from virtual stains, a framework is required to quantify the domain of applicability of virtual staining.

In this Chapter, we explore the generalisability of virtual staining models under three common

- **Task 1**: Generalising to new phenotypes
- **Task 2**: Generalising to new cell types
- **Task 3**: Generalising to new phenotypes & cell types

We first discuss the general points about the dataset, training, inference and evaluation procedures and then the three generalisation tasks.

Dataset: To investigate these three tasks, 772,416 HTS images, consisting of bright-field and 3 co-registered widely used fluorescence stains; fluorescein (FITC) for cytoplasm, 6-diamidino-2-phenylindole (DAPI) for nuclei detection and Cyanine (Cy5) for DNA-damage with 3 cell types (ovarian, lung, breast) and 2 phenotypes that correspond to the conditions under which the cells have been cultured (toxic, non-toxic). The wells that contain DMSO (negative control) and the three lowest levels of concentration for 10 GSK prospective compounds are defined as being non-toxic, and the three highest concentrations of each compound as well as etoposide and staurosporine (positive controls) are toxic. Example bright-field and fluorescence stain images of non-toxic and toxic sample conditions for each cell type are shown in Figure 5.2.

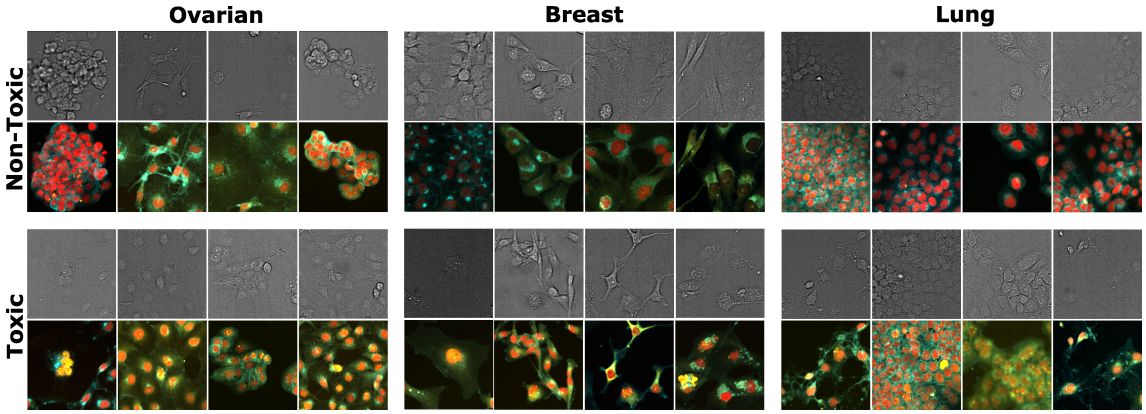


Figure 5.2: **GSK HTS dataset comprised of three different cell types; ovarian, breast, and lung and two phenotypes; non-toxic and toxic**. 2x3 image grid with each grid point comprised of 4 randomly selected bright-field and fluorescence stain image pairs (shown as a composite image) for the cell type and phenotype combination. The composite image shows the nuclei stain (DAPI) in red, the cytoplasm stain (FITC) in cyan, and the DNA-damage stain (Cy5) in yellow. Within the dataset, we observe variability within the toxic and non-toxic samples of each cell type, as well as anatomical differences between the different cell types. We also observe higher levels of cell state heterogeneity in the non-toxic samples compared to more homogeneity in the toxic samples. In this work, we explore the generalisation performance across all three virtual staining tasks for three common HTS data distribution shifts; generalising to an unseen phenotype, generalising to unseen cell types, and generalising to an unseen phenotype and cell types combined.

For each cell type and stain, 27,000 bright-field and fluorescence image pairs were sampled from toxic and non-toxic labeled wells separately, with 21,000 image pairs used for training and 6,000 image pairs used for validation. Random samples for each cell type and labelled wells can be seen in Figure 5.2. In addition, for each cell type, three plates, excluded from any training or validation sets, were used to sample toxic and non-toxic test sets. The test sets for both lung and ovarian toxic were 5,562 and for non-toxic were 4,806, while breast toxic was 3,510 and non-toxic was 6,856.

Training & Inference: Using the non-toxic and toxic samples for each cell type, three models were independently trained with different random initialisations of weights to translate from bright-field to each fluorescence stain. In the following, all performance metrics are computed as the mean performance over all three seeds.

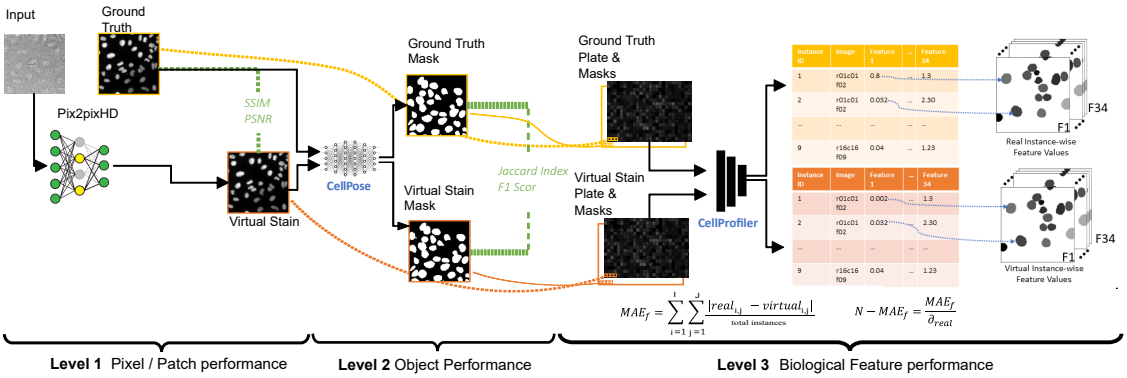


Figure 5.3: **Virtual staining evaluation pipeline Version 2 (Version 1 in Figure 4.4).**
Level 1. Pixel / Patch Performance: We evaluate performance using two established pixel patch-level metrics (SSIM [18], PSNR [17]). No change from version 1.
Level 2. Object Performance: I evaluate the instance segmentation quality that can be obtained from virtually stained nuclei (DAPI) and cytoplasm (FITC) channels. I compare the masks generated from the fluorescence and virtual staining using common segmentation metrics; Jaccard Index [124] for instance-wise pixel area quality and F1 score [116] for object detection. To compute the F1 score, we use a Jaccard Index [124] threshold of 0.7 in alignment with previous nuclei detection works [125].
Level 3. Biological Feature Performance:we utilise Cellprofiler [20], which given an input image and a corresponding segmentation mask (which delineates individual cells or structures), performs instance-level analysis for a collection of 209 morphological, intensity and textural, among other features frequently used in HTS. The segmentation masks are validated to have 98% acceptance rate as shown in Appendix A Using a random forest-based feature selection method similar to [126] we extract the 20 most informative features identified on each of the fluorescence test sets and compare the results between the fluorescence channels and those obtained from virtual staining. The mean absolute error (MAE) between the fluorescence and virtual staining feature scores is computed over all instances, building on previous works [86]. To enable the comparison and interpretation of MAE across different feature ranges, the resulting MAE for each feature is normalised (N-MAE) by the standard deviation of the fluorescence feature score. Finally, across all three levels of evaluation, we report the difference between the mean score obtained for each of the trained models and the baseline model trained on images sampled from the same distribution as the test set. This enables us to compare the change in performance across the different cell type and phenotype combinations relative to each task's virtual staining baseline.

This resulted in a total of 54 models being trained; one model for each of the 2 phenotypes × 3 cell types × 3 channels × 3 initialisations. This work used the pix2pixHD [24] architecture adapted for our image resolution via rescaling [86].

The same hyperparameters as Chapter 4 experiments were used. Each model was trained for a maximum of 200 epochs using early stopping. A total of 243 individual inference runs were completed each corresponding to the different train and test combinations. For task 1, 27 models were trained (× 3 cell types × 3 stains × 3 initialisations) for each

model. For each model, inference was performed on the non-toxic and toxic test sets, producing 54 inference runs in total. For task 2, 27 models were trained and inference was run on the two different cell type test sets, producing 54 runs in total. Finally, for task 3, 9 models were trained for each cell type and phenotype (3 stains $\times$ 3 initialisations), for the toxic phenotype models we ran inference on all three toxic cell type test sets, and for the non-toxic, only the two toxic test sets with different cell types. This led to 45 inference runs per cell type, totalling 135, plus the 108 from tasks 1 and 2, giving a total of 243 runs. Each was evaluated at three levels: pixel/patch-based, instance-based and biological-feature-based. The complete table of results for each inference run across all three levels of evaluation can be found in Appendix B.

Evaluation: In level 1, performance is evaluated using two established pixel/patch-level metrics (SSIM [18] with a patch size of 11x11 and PSNR [17]).

In level 2 we evaluate the instance segmentation quality that can be obtained from virtually stained nuclei (DAPI) and cytoplasm (FITC) channels. Cellpose [117] is used to generate instance masks for the fluorescence and virtual staining channels. To obtain high-quality segmentations, we apply a gamma correction of 0.3 before feeding the images to Cellpose. The value of 0.3 was validated by manually checking a subset of random nuclei and cytoplasm fluorescence images and the Cellpose segmentation masks. The masks generated from the fluorescence and virtual staining are then compared using common segmentation metrics; Jaccard index [124] for instance-wise pixel area quality and F1 score [116] for object detection. To compute the F1 score, an intersection-over-union threshold of 0.7 is used in alignment with previous nuclei detection works [125].

In level 3, We utilise a Cellprofiler [20] pipeline that takes the intensity image and Cellpose masks as input to compute instance-wise scores for a collection of 209 morphological, intensity and textural, among other features frequently used in HTS. The types of features are similar to those used in [86], but by extracting these from Cellprofiler [20] we can compute values for each instance instead of a single mean value over all instances in an image Using a random forest-based feature selection method similar to previous works [126] we extract the twenty most informative features identified on each of the

fluorescence test sets and compare the results between the fluorescence channels and those obtained from virtual staining. The mean absolute error (MAE) between the fluorescence and virtual staining feature scores is computed over each instance, building on previous works [86]. To enable the comparison and interpretation of MAE across different feature ranges, the resulting MAE for each feature is normalised (N-MAE) by the standard deviation of the fluorescence feature score. Finally, across all three levels of evaluation, the difference between the mean score obtained for each of the trained models and the baseline model trained on images sampled from the same distribution as the test set is reported. This enables us to compare the change in performance across the different cell types and phenotype combinations relative to each task's virtual staining baseline. This can be seen in Figure 5.3.

5.1 Training on non-toxic vs toxic samples improves virtual staining performance in particular for ovarian cell lines

We begin by exploring, for each cell type, how virtual staining models trained on images containing samples of one phenotype, non-toxic, perform on images containing samples of a different phenotype, toxic. The difference in performance across the three levels of evaluation between the virtual staining models trained on non-toxic samples and the virtual staining models trained on toxic samples of the same cell type is reported.

Notably, across all three levels of evaluation, all three cell types, and all three virtual staining tasks shown in Figure 5.4, we find training on non-toxic samples can lead to improved results for several metrics compared to training on toxic samples of the same cell type. In particular, for all virtual staining tasks we see consistently improved performance in PSNR, Jaccard Index, and F1 score across all cell types as well as a majority showing improved performances in SSIM and N-MAE. However, in test sets, such as lung toxic for the virtual cytoplasm consistently we observe improved performance in levels 1 and 2 but

worsening results in level 3.

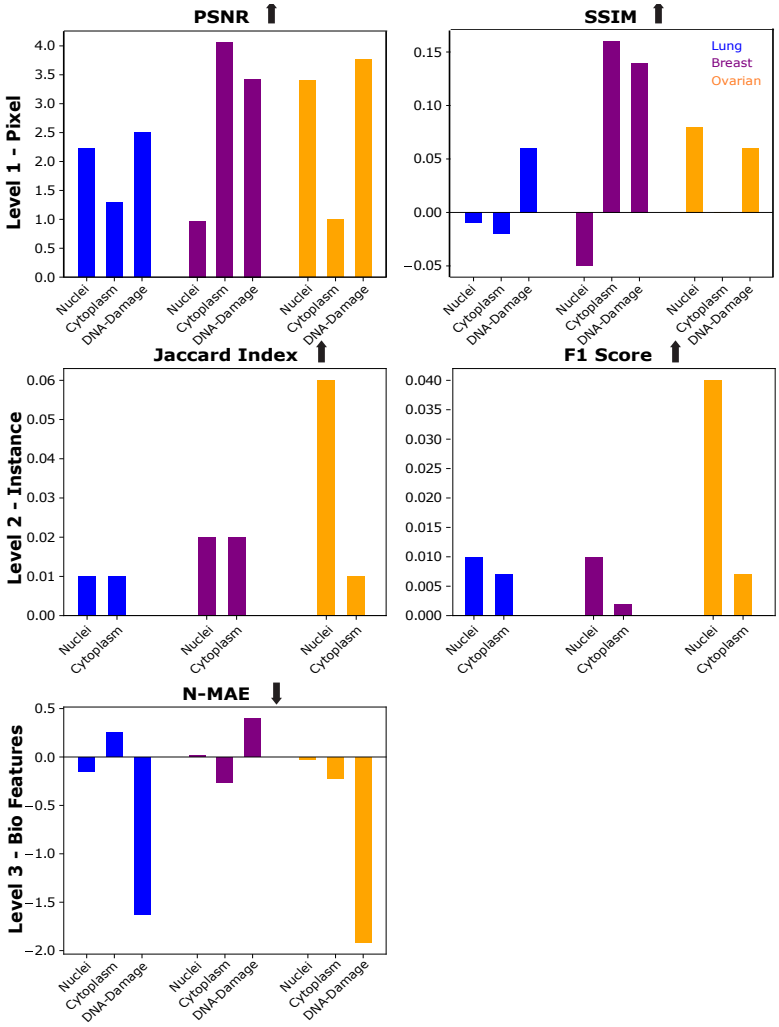


Figure 5.4: **Generalisation performance of virtual staining models to an unseen phenotype across three levels of evaluation**. Each chart represents the results for that metric. Within the chart, all virtual stain channels are shown separately and grouped by cell type. Each bar shows the average difference between the virtual stain models trained on non-toxic and the baseline virtual stain models trained on toxic samples. For all three cell types and virtual stain tasks, the PSNR, Jaccard Index and F1 Score results reveal improved performance from training on non-toxic samples compared to training on toxic samples. Consistently across all metrics, training on ovarian non-toxic leads to improved performance when generalising to images of ovarian toxic cells.

We believe this is in part due to the heterogeneity of cell expression that occurs in non-toxic healthy cells, providing an increase in the diversity of cell states to train generalisable virtual staining models on. In contrast, the toxic cells are induced into specific homogeneous cell states, leading to potentially less diversity in training data, leading to less generalisable models. Some evidence of this can be seen in Figure 5.2, where despite being non-toxic, small amounts of naturally occurring DNA-damage signal

in all non-toxic cell types images are shown.

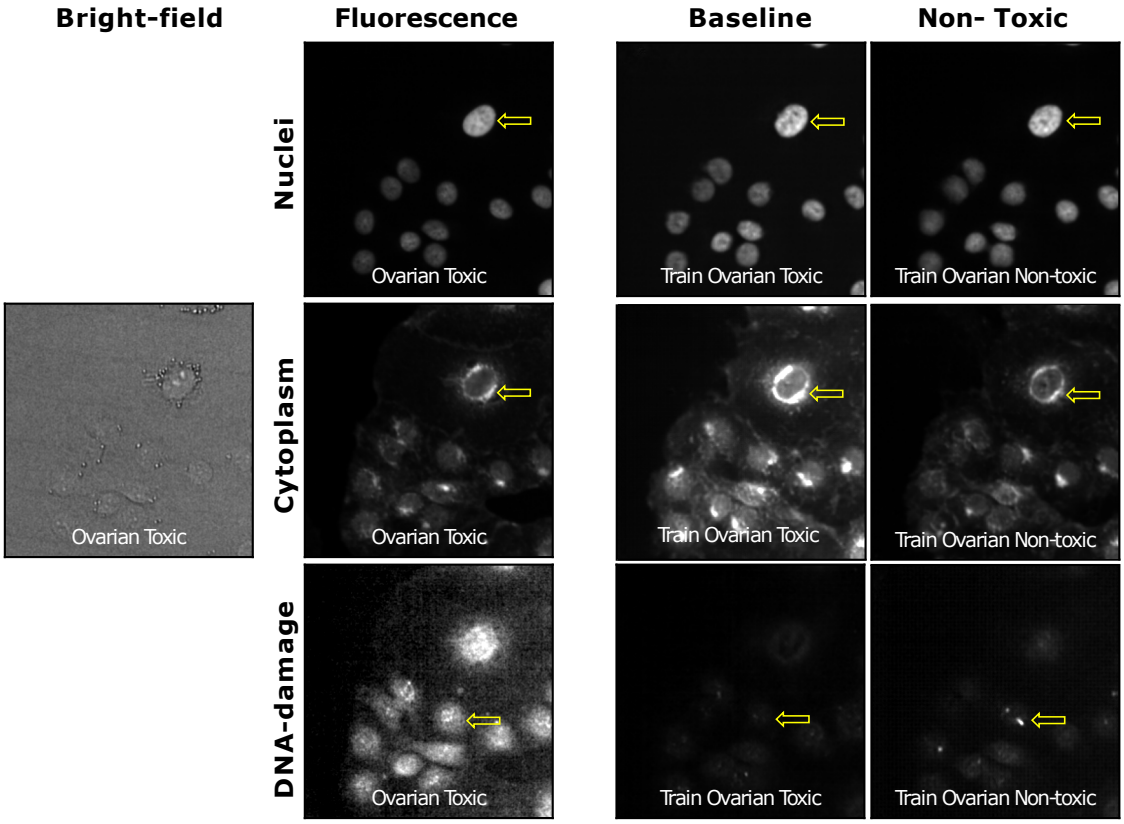


Figure 5.5: **Qualitative results for the task of generalising to an unseen phenotype; ovarian toxic from ovarian non-toxic**. Randomly selected bright-field and paired fluorescence for nuclei, cytoplasm and DNA-damage stains alongside the virtual staining predictions from each of the virtual stain models trained on ovarian toxic samples and the virtual stain models trained on ovarian non-toxic samples. We observe that the general shape of nuclei and cells are reproduced well relative to the baseline and fluorescence stain. The DNA-damage spots are considerably different from the fluorescence stain for both the baseline and the model trained on non-toxic samples. Examples for all three virtual stain tasks are shown by yellow arrows.

Across all measurements and virtual staining channels, when generalising to ovarian toxic samples, training on ovarian non-toxic samples leads to improved performance compared to training on ovarian toxic samples. Upon visual inspection of Figure 5.5 for the virtual nuclei and virtual cytoplasm stains, both the baseline and non-toxic models have produced predictions that replicate the general shape and intensity of a large number of cells seen in the fluorescence.

The prediction of the virtual nuclei trained on non-toxic is almost indistinguishable from the prediction of the virtual nuclei trained on toxic. However, upon closer inspection, as shown in yellow for certain nuclei the intensity profile and shape of the non-toxic nuclei

more closely resembles that found in the fluorescence. Similar findings are highlighted for the cytoplasm intensity profile of individual cells. In contrast, the virtual DNA-damage prediction for both trained models is very similar to each other, with the model trained on non-toxic showing a small number of DNA-damage spots, but they both display considerable losses of information compared to the fluorescence stain.

Figure 5.6 shows the N-MAE obtained for the 20 most informative biological features

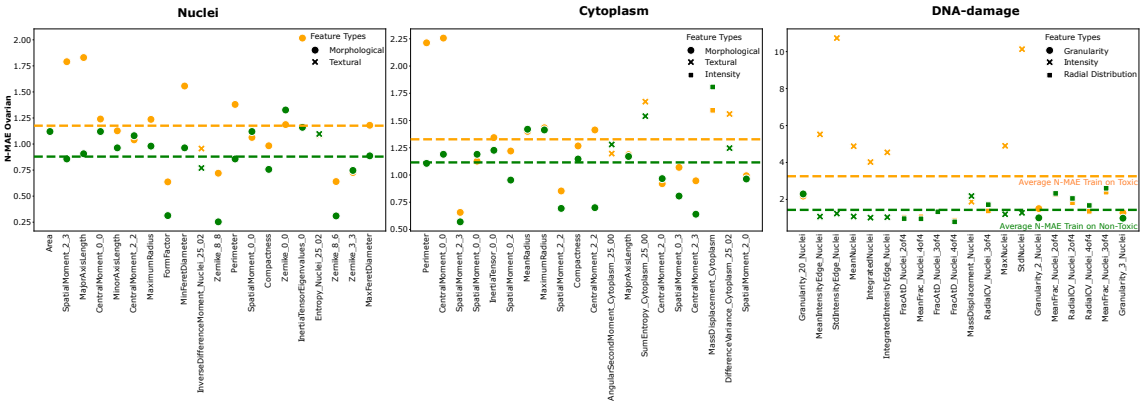


Figure 5.6: **N-MAE values of the virtual stain models trained on ovarian non-toxic and ovarian toxic samples for the 20 CellProfiler features identified on the fluorescence stain.** A N-MAE of 0.1 is equivalent to 10% of one standard deviation identified in the fluorescence stain. The green line shows the average N-MAE for the virtual stain models trained on ovarian non-toxic samples and the yellow line shows the average N-MAE for the virtual stain models trained on ovarian toxic samples. Across a diverse set of feature training on ovarian non-toxic leads to a biological feature representation that more closely aligns with that found in the fluorescence stains compared to training on ovarian toxic.

for the virtual staining models trained on ovarian toxic and ovarian non-toxic samples. Across all virtual staining tasks, a reduction in the average N-MAE when training on images of ovarian non-toxic is observed. Consistently, it can be observed that this result is not driven by a small number of outliers, but by reductions in N-MAE across a diverse set of biological features. These findings support the previously shown results that virtual nuclei and cytoplasm models trained on non-toxic samples of specific cell types, such as ovarian, when generalising to toxic samples of the same cell type, can learn meaningful and diverse biological feature representations that more closely align with the fluorescence stains.

5.2 Generalising to new cell types is complex

Having explored the generalisation to an unseen phenotype, in this section, we focus on the second task of generating virtually stained samples of bright-field images of a different cell type to the images the virtual staining models were trained on.

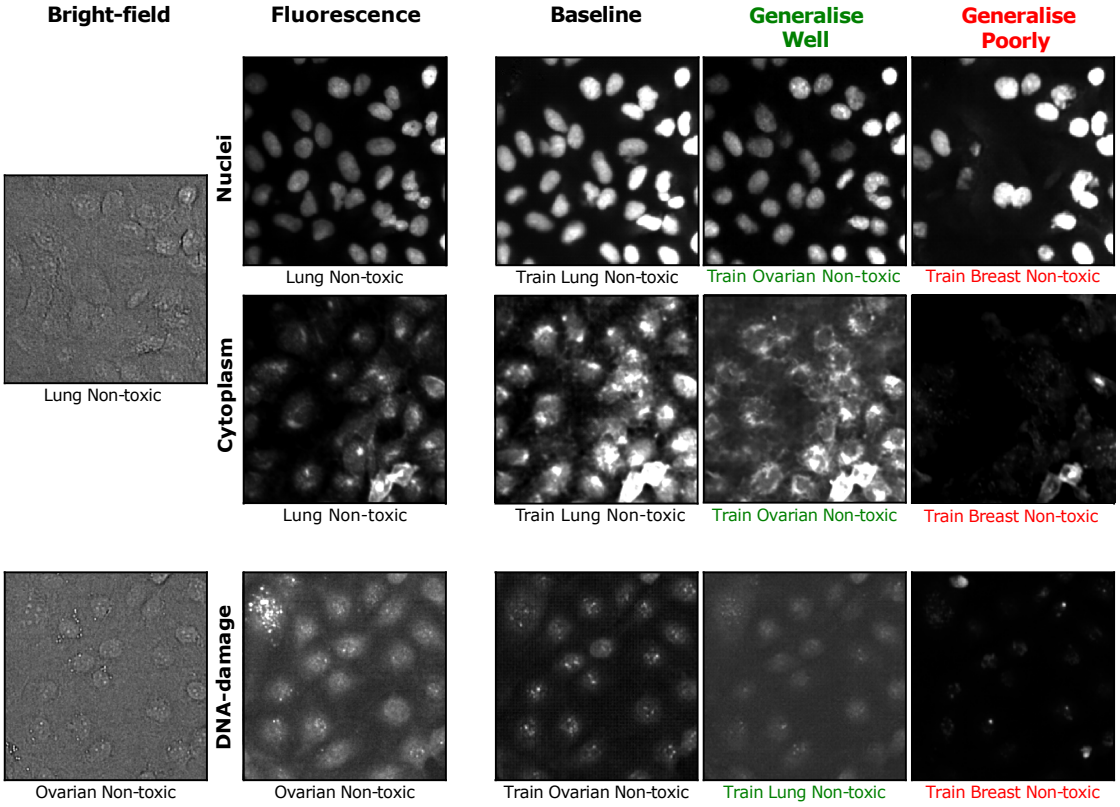


Figure 5.7: **Qualitative results for the task of generalising to unseen cell types.** Randomly selected bright-field and fluorescence for nuclei, cytoplasm and DNA-Damage stains with the virtual stain from the baseline model and the virtual stain models that generalise well and generalise poorly. Both the virtual nuclei and virtual cytoplasm trained on images of ovarian non-toxic cells can reproduce the general shape of the lung cells and, in some cases, the intensity profile well relative to the baseline model trained on images of lung cells. Meanwhile, the models trained on images of breast cells show a considerable number of nuclei and cytoplasm missing, as well as incorrect morphology. For the virtual DNA-damage although the model trained on lung does well relative to the model trained on breast, there are still very clear differences in intensity and DNA-damage spot locations between all three virtual stain predictions and the fluorescence stain.

Upon visual inspection of Figure 5.7, it shows that for the virtual nuclei and virtual cytoplasm task, the model trained on ovarian cells can reproduce the general shape of the lung nuclei and cytoplasm compared to the baseline model trained on images of lung cells. Although gross image features can be visually seen to be comparable, performance for

morphological detail improved when the model was trained on data of the same cell type, as also seen in previous works [4, 10]. The model trained on the breast images is unable to reproduce several nuclei and cytoplasm that can be seen in the fluorescence stain. For the virtual nuclei staining task, across the entire lung non-toxic test set, we can see very similar patterns in Figure 5.8. When training on ovarian non-toxic images compared to training on breast non-toxic, we observe considerably higher image scores. Here, we can observe that this performance is consistent across the entire test set and not driven by a small number of outlines.

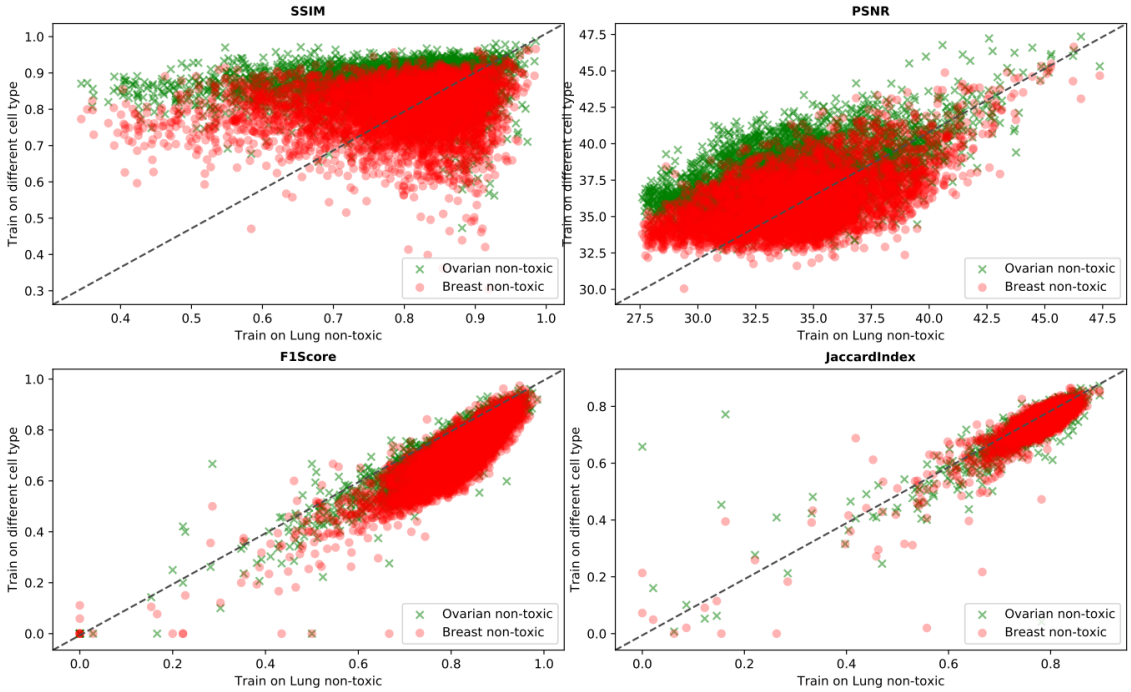


Figure 5.8: **Scatterplots of SSIM, PSNR, F1 Score and Jaccard Index for generalisation to lung of the virtual ovarian and breast nuclei models**. Along the x-axis, we have the values obtained from the model trained on lung (baseline) and on the y-axis the corresponding values obtained from ovarian (green), breast (red) and lung (shown as diagonal).

What is also interesting to note is that for a number of images, the scores for **both** the models trained on breast and the models trained on ovarian are higher than the baseline model trained on lung non-toxic. This considerably reduces when we evaluate performance at the instance level, providing further evidence of the importance of evaluating virtual staining beyond pixel/patch-level measurements. Similar performance can be seen for the virtual cytoplasm task in Appendix B Figure B.6.

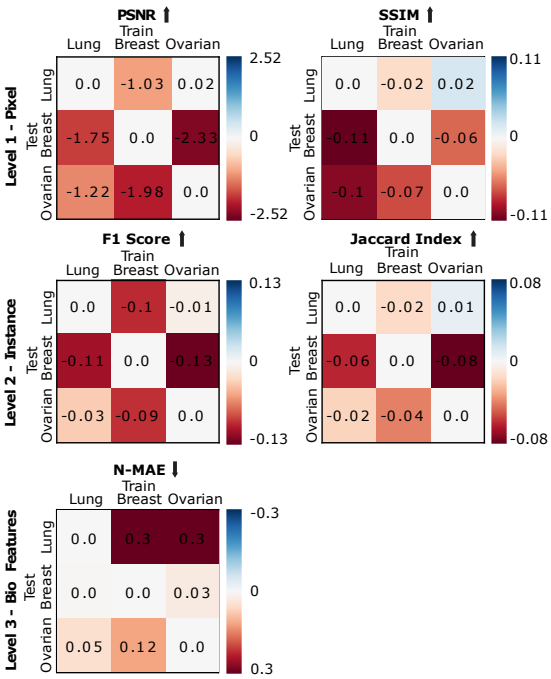
Figure 5.9a shows the different train and test cell type combinations explored for this

generalisation task and the mean scores for the virtual nuclei and cytoplasm models for each level of evaluation. In general, some high-level patterns of consistently good and bad generalisation performance can be seen across the different train and test combinations for levels 1 and 2 of our evaluation pipeline.

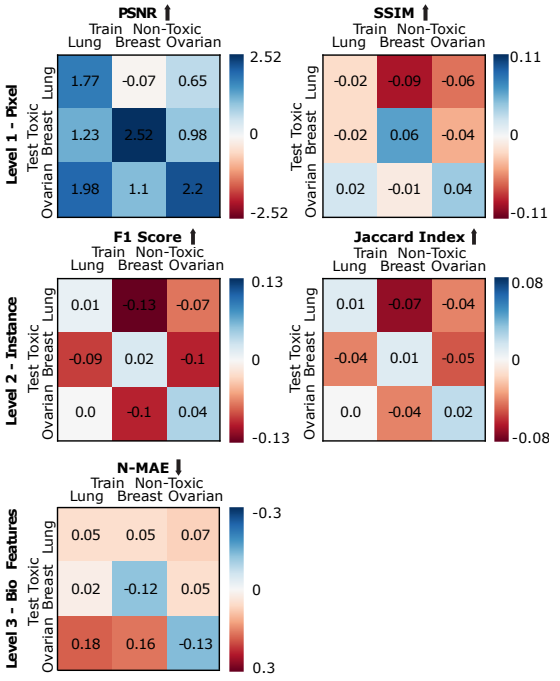
For all metrics in levels 1 and 2, the virtual nuclei and virtual cytoplasm models trained on images of ovarian cells consistently outperform the corresponding virtual staining models trained on images of breast cells when testing on images of lung cells (shown in row 1) and, in some cases leads to a very small positive effect compared to the baseline model. However, when evaluating the biological features in level 3, there is a substantial increase in N-MAE, highlighting the importance of evaluating virtual staining in the biological feature space as opposed to using only common pixel/patch-level and instance-level metrics. Interestingly, despite virtual models trained on images of ovarian cells, they performed well on images of lung cells. Figure 5.9a shows this is not true for the majority of metrics for generalising to images of breast cells, providing further evidence that the generalisation to multiple different cell types is complex and that generalisation performance to one cell type does not necessarily mean the same generalisation performance to a different cell type.

In contrast to the virtual staining models trained on images of ovarian cells, for all three levels of evaluation, the virtual nuclei and cytoplasm models trained on images of breast cells produce scores that show the largest difference in performance compared to the baselines when testing on images of ovarian and lung cells.

We also note that generalising to images of breast cells from models trained on images of ovarian or lung cells leads to the lowest generalisation performance. This suggests that images of breast cells are not as good as lung or ovarian cells for training virtual staining models that generalise well to unseen cell types, and they are also difficult to generalise to. Finally, for the virtual nuclei and virtual cytoplasm results, there is a lack of symmetry in performance across the different cell types for each of the measurements. The large majority of results do not show that performance when training on one cell type and testing on another is very similar if the train and test cell types are interchanged. For



(a) **Task 2. Generalisation to unseen cell types. .** The values along the diagonal represent the baseline and are therefore 0. Models trained on breast cell images consistently generalise poorly to other cell types and breast cell images appear hard to generalise to. Levels 1 and 2 show ovarian can generalise well to lung for pixel and instance-level metrics, but not when evaluating the biological features. Good generalisation to one cell type does not imply the same for other cell types.



(b) **Task 3. Generalisation to unseen cell types and phenotypes.** For each metric, the layout of the train/test and the conditional formatting scales are the same as Figure 5.9a. The values along the diagonal represent the results from generalising to an unseen phenotype and are kept in for completion. We observe similar patterns of performance when training and testing on images of breast cells to the generalisation to unseen cell types as well as considerable improvements in PSNR and N-MAE for several train and test combinations.

Figure 5.9: **Generalisation performance of virtual nuclei and virtual cytoplasm models.** Both (a) and (b) have 5 heatmaps with each showing the average results for each metric within the 3 levels of evaluation. Each heatmap shows the cell type train and test set combinations. The metric values are the differences between the models and the baseline model. For each metric, conditional formatting is centred at 0 and scaled to the worst performance across tasks 2 and 3 (to allow for comparison) and it's negative, with red indicating bad generalisation performance changes and blue positive.

the virtual DNA-damage models (not shown in Figure 5.9a), similar to virtual nuclei and cytoplasm, training on images of breast cells and testing on images of ovarian and lung cells consistently leads to large differences across all metrics. Meanwhile, training on images of lung cells when testing on images of ovarian and breast cells leads to small positive effects across PSNR and SSIM, as well as a very small increase in N-MAE. Examining the individual test set image scores for each cell type generalisation task across all three virtual staining models, for all metrics, we find that the differences are consistent

across multiple images and not driven by a small number of outliers.

5.3 Generalisation to new cell types and phenotypes shows good generalisation performance

In the final section, the task combines the two previous distribution shifts and explores how virtual staining models trained on images of non-toxic cells of one cell type generalise to images of toxic cells of another cell type.

We report the difference in performance across the three levels of evaluation between the virtual staining models trained on non-toxic samples and the virtual staining models trained on toxic samples of the same cell type.

Generally good generalisation performance In a similar approach to Figure 5.9a, Figure 5.9b shows the different non-toxic training and toxic test cell type image set combinations explored for this generalisation task and the corresponding mean scores for the virtual nuclei and virtual cytoplasm models for each of the three levels of evaluation.

Similar differences between the performance results within levels 1 and 2 against level 3 are found in task 1. When training on non-toxic lung cells and testing on toxic ovarian improvements in pixel/patch-wise and instance-wise metrics relative to the baseline, increases in N-MAE when evaluating the biological feature representation can be observed.

In general, there are considerable improvements across all metrics compared to those shown in Figure 5.9a which supports our findings from the phenotype generalisation task. In particular, the increase of positive PSNR results and the reduction of the highest N-MAE from 0.3 to 0.18. However, in some cases, such as when testing on ovarian toxic where an increase in N-MAE was shown. For levels 1 and 2, similar generalisation difficulties are observed when training and testing on images of breast cells.

5.4 Discussion

Here, we have investigated the generalisation performance of virtual nuclei, virtual cytoplasm and virtual DNA-damage models for three common HTS generalisation tasks. Firstly, generalising to an unseen phenotype. Secondly, generalising to unseen cell types. Finally, generalising to an unseen phenotype and cell types combined. Performance has been evaluated using metrics at the pixel/patch level, instance level and biological feature level.

For the first generalisation task, for both virtual nuclei and virtual cytoplasm, training on non-toxic samples leads to not only good generalisation but improved performance relative to training on toxic samples across all levels of evaluation. In particular, when testing on toxic ovarian samples training on non-toxic ovarian samples produced predictions that achieved higher pixel/patch-level quality, more accurate instance-wise translation and scores for the majority of biological features that more closely represented the score found in the fluorescence channels.

Previous work on the generation of The JUMP Cell Painting Dataset [83] suggests it is necessary to have training sets with a high diversity of phenotypes to effectively train machine learning models. However, these results demonstrate that for this specific HTS dataset and the explored virtual staining tasks, training on non-toxic samples alone can lead to not only good generalisation but also performance improvements when measured using a variety of evaluation metrics. As such, we believe that for virtual nuclei and virtual cytoplasm staining of toxic samples training on widely available non-toxic samples is a viable alternative to training on toxic samples. These findings could lead to a reduction in the number of cell- and phenotype-specific models needed to efficiently utilise virtual staining for diverse HTS datasets.

For the second generalisation task, we find generalising to unseen cell types is complex. We identify for pixel/patch-wise and instance-based metrics, training on ovarian and testing on lung led to good generalisation performance for the virtual nuclei and virtual cytoplasm. However, when evaluating at the biological feature level, we observed relatively high differences in N-MAE values. These results reveal that determining the correct evaluation

metric for a chosen virtual staining application is important to effectively evaluate whether certain cell-type-specific training sets can better generalise compared to other cell types. Additionally, we find that good generalisation to one unseen cell type does not necessarily mean the same can be expected for another cell type. In contrast, we observe that training on images of breast cells consistently leads to bad generalisation performance across all levels of evaluation. We also found that models trained on other cell types were poor at generalising to images of breast cells. Further analysis into the three non-toxic cell type data sets revealed, after a manual inspection of a random subset of 5,000 images from each cell type, that breast cells are distributed more sparsely than the other cell types (Figure 5.2). The average number of cells in the inspected breast images was less than half of that in ovarian and a third of that in lung. This corresponds to the anatomical function of breast cells, which require more fat in their surroundings [127], reducing the number of cells in any image as shown in Figure 5.2.

On the other hand, endocrine signaling [128] of ovarian cells requires that they be closely packed together, and the vital role of lung cells to exchange gas, in principle, requires more cells. We believe that these structural differences are responsible for the poor generalisation performance of images of breast cells. An additional possible explanation could be that the lower density of breast cells corresponds to a smaller volume of training data for the virtual staining models.

For the final generalisation task, we see the same challenges when training on and generalising to images of breast cells for the virtual nuclei and virtual cytoplasm. However, in general, across all levels of evaluation and the explored celltype and phenotype combinations, we find that training on non-toxic is preferable even when additionally generalising to toxic samples of an unseen cell type (See Figure 5.9b compared to Figure 5.9a). We saw particular improvements in PSNR and N-MAE and relatively similar performance for SSIM, F1 Score and Jaccard Index.

Despite what appears to be good virtual DNA-damage generalisation performance across certain evaluation metrics for all three generalisation tasks, when examining the absolute values, the results are poor compared to virtual nuclei and virtual cytoplasm,

affirming the findings from previous work [86]. This becomes clear qualitatively in Figure 5.5 and Figure 5.7 and quantitatively in Figure 5.6, where the N-MAE values are noticeably larger than in the other two channels.

Future work should explore the generalisation performance to a broader selection of unseen cell types and investigate further the issues with producing accurate virtual DNA-damage stains.

Chapter 6

Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell?

Chapters 4 and 5 explore how to evaluate virtual staining within HTS pipelines, with all experiments evaluating the performance of a *single* virtual staining prediction given the corresponding bright-field image. However, as outlined in Chapter 1, virtual staining is an ill-posed problem; given an image such as a bright-field with inherent noise and lack of contrast between the background and the structures of interest, there does not exist a single virtual staining solution. In fact, it is impossible to know exactly what the corresponding fluorescence image looks like based purely on the bright field image, i.e. there is a posterior distribution $P(\mathbf{y}_i|\mathbf{x}_i)$ of possible fluorescence images and therefore a posterior distribution of virtual staining solutions $P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ that could correspond to the same bright-field.

When virtual staining is approached as a regression problem like F-net [4] previously explored in Chapter 4, the virtual staining predictions produce blurry results as shown in Figure 4.1, tending towards to the mean of possible fluorescence images according to

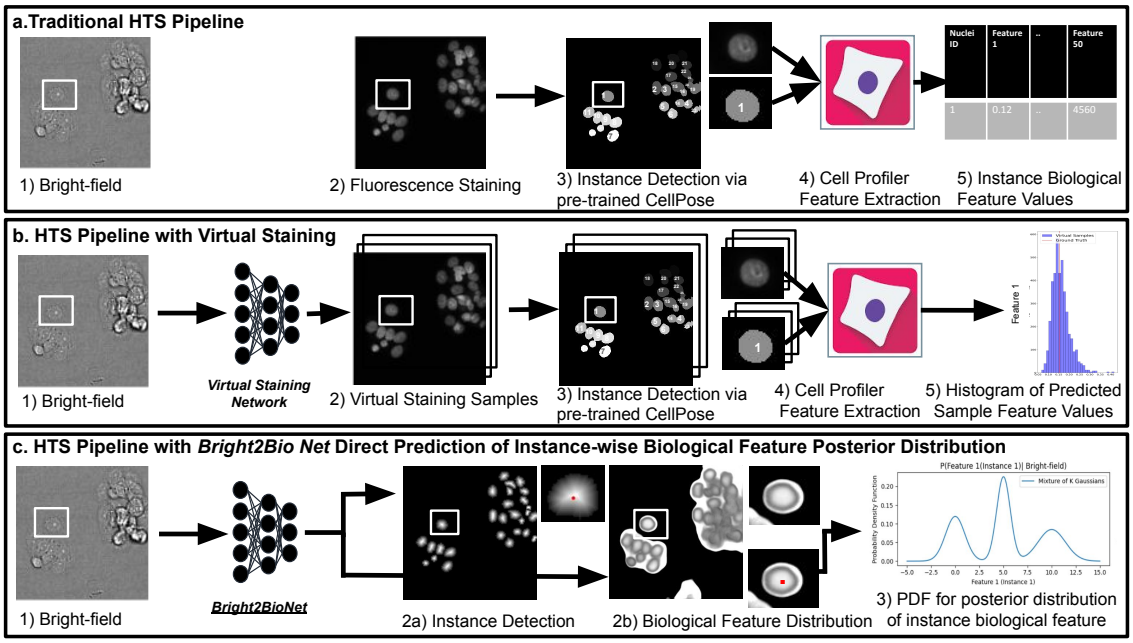


Figure 6.1: **Evolution of high-throughput screening pipelines. a. Traditional HTS pipeline:** Perform fluorescence staining on the bright-field image, generate instance masks, and use fluorescence images and instance masks to extract instance-wise biological features from CellProfiler. **b. HTS pipeline with virtual staining.** A virtual staining network takes a bright-field image as input and generates multiple virtual staining samples. Instance masks are obtained for samples using Cellpose. Samples and masks are used to generate instance-wise biological features from Cellprofiler, which then form an approximate posterior distribution and can be compared to the known ground truth biological feature value.**c. HTS pipeline with direct prediction of instance-wise biological feature posterior distribution.** Bright2Bio-Net takes the bright-field as input and performs two tasks; instance detection and direct prediction of pixel-wise biological feature posterior distribution.

$P(\mathbf{y}_i|\mathbf{x}_i)$.

Conditional generative models, such as those explored in Chapters 4 and 5 learn to approximate this distribution as $P_\theta(\mathbf{y}_i|\mathbf{x}_i) \approx P(\mathbf{y}_i|\mathbf{x}_i)$. This is why the virtual staining samples from GAN-based approaches appear sharper and more realistic than regression-based results.

By extracting multiple samples $\hat{\mathbf{y}}_i \sim P_\theta(\mathbf{y}_i|\mathbf{x}_i)$ and applying instance detection as shown in Figure 6.1b we obtain a set of instance samples $\hat{\mathbf{y}}_i$. Explicitly, within this chapter, an instance refers to a nucleus, we use both terms interchangeably.

For a given biological feature f we can compute the feature values $\hat{y}^l_{i,j} = f(\hat{\mathbf{y}}^l_{i,j})$ for each sample l and each instance j in image i.

We can use the samples $\hat{y}^l_{i,j}$ with a density estimator; $P_\theta(y_{i,j}|\mathbf{x}_{i,j}) \approx P(y_{i,j}|\mathbf{x}_{i,j})$ the posterior distribution of instance biological feature values.

It is then natural to ask if we could predict the posterior distributions of the biological features $P(y_{i,j}|\mathbf{x}_{i,j})$ directly from the unstained bright-field image without explicitly generating virtual fluorescence images? Such an approach would remove the reliance on external instance detection and feature extraction pipelines and enable direct quantification of the biological features of interest (like in [129]) from the bright-field alone. To investigate this problem, we designed Bright2Bio-Net (B2B-Net) (See Figure 6.1c), a convolutional neural network (CNN) architecture to directly predict the biological feature posterior distributions from unstained bright-field microscopy images. The network learns to perform two tasks; instance detection and a pixel-wise prediction of the biological feature posterior distributions, each parameterised as a mixture of Gaussian models. We introduce the newly proposed concept in section 6.1, but as this is early work we do not evaluate its performance. An important question is how can we quantify the quality of predicted posterior distributions $P_\theta(y_{i,j}|\mathbf{x}_{i,j})$, whether derived from samples or predicted directly from B2B-Net? In other words, how can we check if $P_\theta(y_{i,j}|\mathbf{x}_{i,j})$ closely approximates $P(y_{i,j}|\mathbf{x}_{i,j})$?

While we have direct access to the former PDF, we do not have direct access to the latter. Instead, we only have access to is a single sample $y_{i,j}$ from $P(y_{i,j}|\mathbf{x}_{i,j})$ but how can we use this sample to evaluate the predicted posterior distribution $P_\theta(y_{i,j}|\mathbf{x}_{i,j})$?

In Chapter 4 we avoid this problem by combining the samples $y_{i,j}$ as an approximation of $P(y)$, the true prior distribution over feature values, which we term the true *population distribution*. We then compare this population distribution with the predicted population distribution $P_\theta(y)$ obtained by combining all predicted samples $\hat{y}_{i,j}$ thus ignoring the correspondents between input $\mathbf{x}_{i,j}$ and feature values $y_{i,j}$.

Quantitatively, these two distributions can be compared using common metrics such as the Wasserstein distance (WDist) and the Kullback-Leibler divergence (KLD) computed between the predicted and target populations to quantify their difference. Related works on inverse problems in computer vision [36, 130] follow similar approaches.

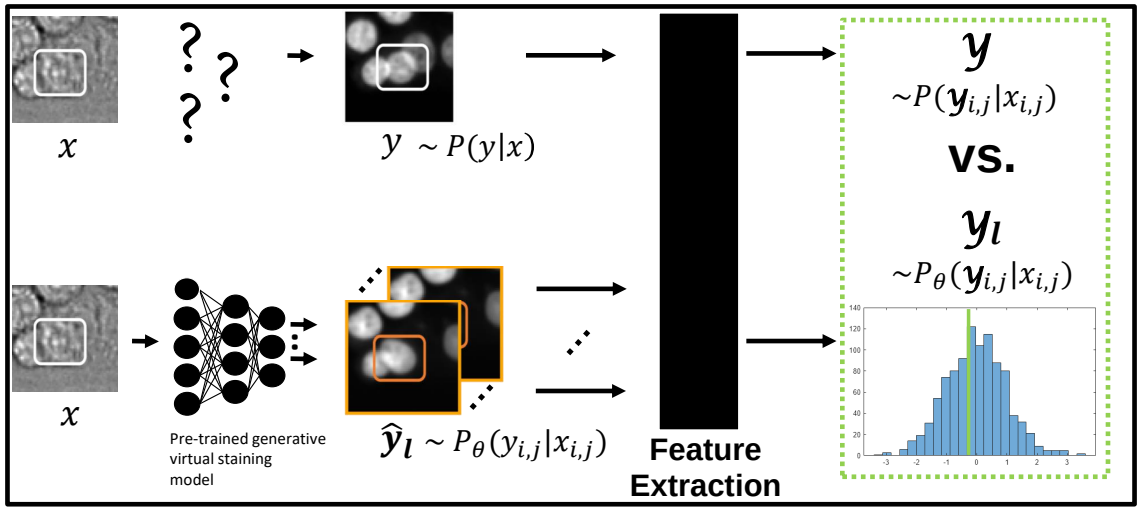


Figure 6.2: **How can we evaluate a predicted posterior distribution with access to only a single sample from the true posterior distribution?** Given the bright-feild alone, it is impossible to know what the true fluorescence stain is, there is a posterior distribution $P(y|x)$ of possible solutions. Typically, we have access to a single sample y from this distribution which we can pass through a feature extraction algorithm to generate a corresponding sample $\boldsymbol{y}$ for the instance wise biological feature posterior distribution $P(\boldsymbol{y}_{i,j}|\mathbf{x}_{i,j})$ for image i instance j. Probabilistic generative models such as cGANs or as we explore in this chapter diffusion models can be used to extract l samples $\hat{y}_l \sim P_\theta(y_{i,j}|\mathbf{x}_{i,j})$ which can then be passed into the feature extractor to produce a biological feature posterior distribution $P_\theta(\boldsymbol{y}_{i,j}|\mathbf{x}_{i,j})$ for that bright-field instance shown here as a histogram. One challenge remains: how can we evaluate a predicted posterior distribution with only a target single sample?

The total population can be viewed as a hierarchical structure, such as a pyramid, with each level representing levels of granularity or sub-population present in the training data, like cell types, and phenotypes, through to individual biological structures such as cell nuclei. We argue in this work that yes, we want the overall population distributions, i.e. the predicted and target pyramids to be similar, but it is important that in principle, they are similar because of similarities within the population at different levels of granularity. In particular, we focus on individual nucleus results and argue that accurate instance posterior distributions should result in at least as good or better population distributions; however, the reverse is not always true. Evaluating the populations as a whole could lead to assumptions about the similarity at more granular levels of the data. It also does not enable the quantification of feature combinations for new or very rare cellular events that can not be observed just by looking at the population distributions.

Currently, there does not exist a well-understood and commonly used metric that

enables the comparison of a distribution to a single sample value such as $P_\theta(y_{i,j}|\mathbf{x}_{i,j})$ compared to $y_{i,j}$ from $P(y_{i,j}|\mathbf{x}_{i,j})$. For this application, a good metric should capture the uncertainty in the predicted posterior distribution and quantify how close it is to the true posterior distribution.

In section 6.2, we present three new evaluation metrics to measure the quality of the predicted biological feature posterior distributions when only a single ground truth sample is available. Namely, the average log-likelihood, the Wasserstein distance (Wdist) between the standard scores and $\mathcal{N}(0,1)$ and Wdist between the percentile ranks and $U(0,100)$. We will for the first time, analyse the advantages of these new metrics compared to common population metrics, including the WDist and the KLD computed over the predicted and target biological feature populations. The predicted feature populations will be extracted from virtual staining samples obtained from two conditional generative models; Denoising Diffusion Probabilistic model (DDPM) [28], adapted for the conditional setting to form cDDPM (as discussed in Chapter 2) and the previously explored pix2pixHD [86].

In summary, the contributions are as follows:

- Early principled works into Bright2BioNet, a novel deep learning-based method for predicting biological feature posterior distributions directly from an unstained bright-field.
- Three new evaluation metrics for computing the similarity of an instance predicted posterior distribution to a single sample from the true posterior.
- A thorough analysis of these new evaluation metrics compared to common population measurements for two conditional generative models pre-trained for virtual staining.

6.1 Bright2Bio-Net (B2B-Net) - a direct method for predicting the posterior distribution of instance-wise biological features

In this section, we propose the design of B2B-Net –an ML technique– to directly translate label-free bright-field microscopy images into instance biological feature posterior distributions. Compared to previous virtual staining works, B2B-Net removes the reliance on external instance detection and feature extraction pipelines and enables direct quantification of the biological features of interest from the label-free bright-field microscopy images alone. B2B-Net, as shown in Figure 6.3, is a convolutional neural network (CNN)

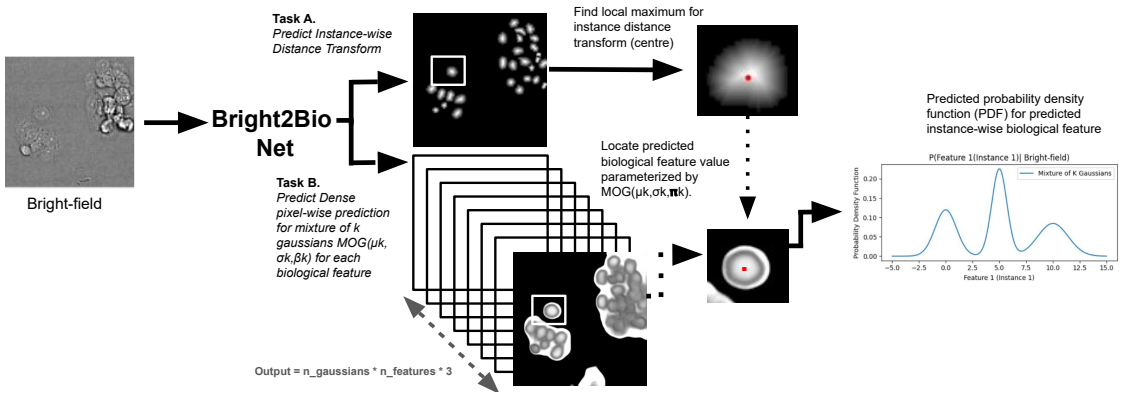


Figure 6.3: **Bright2Bio Net**. Takes a bright-field as input and learns two tasks. **Task A** - instance detection by minimising the mean squared error (MSE) between a predicted and target distance transform. **Task B** - a dense per-pixel prediction of 34 biological feature posterior distributions, each parameterised by a mixture of Gaussians and learnt by minimising the negative log-likelihood. For each instance, the local maximum of the predicted distance transform is used to determine the predicted feature distributions from task B.

architecture that learns to perform two tasks; instance detection and a dense pixel-wise prediction of the biological feature posterior distributions, parametrised as a Gaussian mixture model. The network takes as input a batch of bright-field images and produces an output for each task. The output for task A is a predicted distance transform of each nucleus concerning its boundary, and the output for task B is a pixel-wise prediction of individual biological feature posterior distributions parametrised as a mixture of Gaussian models. As shown in Figure 6.3, during inference, the predicted centre (local maximum)

of each instance corresponding to the nucleus that the pixel belongs to is identified from the predicted distance transform of task A. The pixel location is then used to determine the pixel for the predicted pixel-wise biological feature posterior distribution.

The architecture of B2B-Net is based on a U-Net [49] starting with 64 filters and the number of filters increases by 2^n to a depth of 4. Each encoder block produces two outputs; the traditional feature maps plus attention maps that are passed through the skip connections to be concatenated with the corresponding decoder block.

The final number of output channels is equal to (the number of Gaussian components $*$ the number of features $*$ 3 channels; one for each mixture of Gaussian parameter) + 1 for the distance transform. For example, 3 Gaussian components leads to 172$((3 * 18 * 3) + 1)$ output channels.

Among the different hyper-parameters of the network, one core design choice was the number of Gaussian components to use. Although this may appear to only directly impact task B in practice we found this to indirectly impact task A. By design, increasing the number of Gaussians makes the learning of task B more complex, which, without any additional regularisation led to the weights of the network being optimised more towards task B, negatively impacting task A. Introducing weights to both loss terms helped with balancing the two tasks. The value of this weight would vary depending on the number of Gaussians used for task B and would require further investigation.

Task A. Instance detection early investigation into suitable loss functions explored the binary cross-entropy loss with binary instance masks. However, during training, we observed that when nuclei were dividing or touching the network would produce predicted nuclei masks that had merged incorrectly. Additionally, for this particular setting, the goal is not to produce a semantic segmentation mask over all nuclei pixels but to locate nuclei centres so that the x and y coordinates can be used to localise the predicted pixel-wise biological feature posterior distributions obtained in task B. In light of this, for task A, the network uses the Mean-squared error loss,

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{6.1}$$

where y_i is the target distance transform and $\hat{y}_i$ is the predicted distance transform. The distance transform pixel values represent the distance from that pixel to the nearest edge of an instance. The distance transform was applied recursively, one instance at a time, so that the central pixel(s) are set to 1 and there is a border of 2 pixels between neighbouring nuclei. At inference time, the local instance maxima in x and y are obtained from the predicted distance transform and used to localise the predictions from task B.

The performance of task A can be evaluated using accuracy, precision, recall and the F1 score. We determine a true positive to be a single predicted local maximum in the instance distance transform that lies within the border of the known target distance transform. False positives are predicted local maxima outside of any instance border or any duplicates (ie an instance border that has already been identified as a true positive). False negatives are labelled as any target instances where no local maximum was predicted.

Task B - pixel-wise prediction of biological feature posterior distributions B2B-Net learns to predict a pixel-wise estimation of the predicted posterior distribution, parametrised as a mixture of Gaussian models for 18 biological features. Given a mixture of Gaussian models with K components, for an input bright-field image x and predicted parameters: w_k weight of the k-th mixing coefficient, μ_k the mean of the k-th Gaussian component and σ_k^2 the variance of the k-th, the predicted probability density function of the mixture is:

$$P(\hat{\mathbf{y}}|w, \mu, \sigma) = \sum_{k=1}^{K} w_k \cdot \mathcal{N}(\mathrm{x}; \mu_k, \sigma_k^2),$$

where $\mathcal{N}(\mathrm{x}; \mu_k, \sigma_k^2)$ is the Gaussian probability density function for component k:

$$\mathcal{N}(\mathrm{x}; \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp -\frac{(\mathrm{x} - \mu_k)^2}{2\sigma_k^2}.$$

The network is trained via the

$$\text{Negative Log-likelihood (NLL)} = -\log \sum_{k=1}^{K} w_k \cdot \mathcal{N}(\mathbf{x}; \mu_k, \sigma_k^2).$$

6.2 How can we evaluate a predicted posterior with only a single ground truth sample from the true posterior distribution?

We begin with access to a collection of bright-field images $(\mathbf{x}_1 \cdots \mathbf{x}_N)$ where each $\mathbf{x}_{i,j}$ corresponds to an instance j in the bright-field image i. We also have access to a single biological feature value for each instance sampled from the true posterior distribution $\mathcal{Y}_{i,j} \sim P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$. We can approximate $P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ as $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ either directly or via a conditional generative model as outlined in the previous section. However, how can we evaluate a distribution to a single known value? In the following section, we introduce three novel evaluation metrics for precisely this task; quantifying the similarity between $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ and $\mathcal{Y}_{i,j} \sim P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$.

Average Log-Likelihood

The first proposed metric is the

$$\text{Average Log-Likelihood} = \frac{1}{N}\sum_{j=1}^{N} \log P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j}), \tag{6.2}$$

where $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ is the density estimation of the predicted biological feature posterior distribution $P(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ for a given bright-field image i and instance j.

Motivated by this, we show that the average log-likelihood is a substitute for the average KLD over the dataset. We show this by first writing the formula for the average Kullback-Leibler divergence as

$$\frac{1}{N}\sum_{j=1}^{N} D_{\text{KL}}(P(\mathcal{Y}|\mathbf{x}_{i,j})||P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})) = \frac{1}{N}\sum_{j=1}^{N}\int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log(\frac{P(\mathcal{Y}|\mathbf{x}_{i,j})}{P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})})d\mathcal{Y}, \tag{6.3}$$

We can rewrite Equation 6.3 as the expectation over all instances j and feature values $\mathcal{Y}$

as

$$= \frac{1}{N}\sum_{j=1}^{N} \underbrace{\int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log P(\mathcal{Y}|\mathbf{x}_{i,j}) d\mathcal{Y}}_{(1)} - \underbrace{\int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j}) d\mathcal{Y}}_{(2)}. \tag{6.4}$$

We can observe that term (1) is negative entropy of $P(\mathcal{Y}|\mathbf{x}_{i,j})$ and it is not dependent on $P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})$ but instead remains *constant*.

$$= \underbrace{constant}_{(1)} - \frac{1}{N}\sum_{j=1}^{N} \underbrace{\int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j}) d\mathcal{Y}}_{(2)}. \tag{6.5}$$

Term (2) can be interpreted as the expected value of $\log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})$ with respect to the distribution $P(\mathcal{Y}|\mathbf{x}_{i,j})$, which can be approximated using samples from the true posterior $P(\mathcal{Y}|\mathbf{x}_{i,j})$. Here, we approximate it using the single sample we have available in the dataset $\mathcal{Y}_{i,j}$ as

$$= constant - \frac{1}{N}\sum_{j=1}^{N} \log P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j}) - \text{error}_{i,j}. \tag{6.6}$$

The $\text{error}_{i,j}$ for different instances j is defined as

$$\text{error}_{i,j} = \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j}) - \int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j}) d\mathcal{Y}. \tag{6.7}$$

Since the samples $\log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})$ are by definition centred around the expected value $\int P(\mathcal{Y}|\mathbf{x}_{i,j}) \cdot \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})$, the distributions of $\text{error}_{i,j}$ must be centred around 0. By rewriting Equation 6.6 as

$$\underbrace{constant}_{(1)} - \underbrace{\frac{1}{N}\sum_{j=1}^{N} \log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})}_{(2a)} + \underbrace{\frac{1}{N}\sum_{j=1}^{N} \text{error}_{i,j}}_{(2b)}, \tag{6.8}$$

we can see for large N, 2b within Equation 6.8 will go to 0 and we can indeed use

Equation 6.2 to approximate Equation 6.3 up to a constant shift

$$\frac{1}{N}\sum_{j=1}^{N}\log P_\theta(\mathcal{Y}|\mathbf{x}_{i,j}) \approx \text{constant} - \frac{1}{N}\sum_{j=1}^{N} D_{\text{KL}}\left(P(\mathcal{Y}|\mathbf{x}_{i,j})||P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})\right). \tag{6.9}$$

In practice, $P(\mathcal{Y}|\mathbf{x}_{i,j})$ and $P_\theta(\mathcal{Y}|\mathbf{x}_{i,j})$ can be any kernel density estimators, in our application we use a mixture of Gaussian models fitted to the true and predicted posterior samples.

Standard Scores

The first measurement is the WDist [131] between the distribution of standard scores and $\mathcal{N}(0,1)$. The

$$\text{Standard Score} = \frac{(\mathcal{Y}_{\text{i,j}} - \hat{\mu}_{i,j})}{\hat{\sigma}_{i,j}}, \tag{6.10}$$

where $\mathcal{Y}_{i,j}$ is the target feature value, for instance, j in fluorescence image i and $\hat{\mu}_{i,j}$ and $\hat{\sigma}_{i,j}$ are the mean and standard deviations of the predicted posterior distribution $P_\theta(\mathcal{Y}_{i,j}P|\mathbf{x}_j)$ for the same instance j in fluorescence image i computed for a given $\mathbf{x}_{i,j}$. The standard score quantifies how many standard deviations the sample target feature value is from the predicted posterior distribution. We compute this over all instances and plot the distribution of standard scores. If the predicted posterior closely resembles the real feature values, the distribution of standard scores should resemble a $\mathcal{N}(0,1)$, indicating the predicted posterior distributions are very similar to the target sample value. We measure this similarity using the WDist between the feature standard scores and the $\mathcal{N}(0,1)$.

Percentile Ranks

$$\text{Percentile Rank } = F(\mathcal{Y}_{i,j}) \times 100 \tag{6.11}$$

where $F(\mathcal{Y}_{i,j})$ is the cumulative distribution function(CDF) for $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ the predicted instance posterior distribution for a target feature value $\mathcal{Y}_{i,j}$ scaled by 100 to convert into percentages. The CDF can be approximated by sampling a large number of values from

the distribution $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$:

$$\text{Percentile Rank} = \frac{\text{Number of samples less than } y_j \text{ in } P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j}) \ (+1)}{\text{Number of samples}} \times 100, \quad (6.12)$$

$P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ The percentile rank quantifies the percentage of values in $P_\theta(\mathcal{Y}_{i,j}|\mathbf{x}_{i,j})$ that are less than the sample target feature value $\mathcal{Y}_{i,j}$. A percentile rank of 0 indicates that the minimum value in the predicted posterior distribution ($\hat{\mathcal{Y}}_j$) is lower than or equal to the true sample value ($\mathcal{Y}_{i,j}$). A percentile rank of 100 indicates that the minimum value in the predicted posterior distribution is higher than the true sample value. For each feature, across all instances, if the predicted posterior distribution closely resembles the real feature values, the distribution of percentile ranks should resemble a $\mathcal{U}(0, 100)$. We measure this similarity using the Wdist between the feature percentile ranks and the $\mathcal{U}(0, 100)$.

We analyse the Wdist from the standard score and percentile rank for each feature, comparing them to the Wdist computed over the entire population of biological feature values.

6.2.1 Experiments & Results

Dataset: The experiments are based on a pool of 76,707 individual images. 50,944 images were generated as part of the same GSK HTS dataset used in Chapters 4 and 5; the images are 1080x108 and composed of bright-field and 6-diamidino-2-phenylindole (DAPI) pairs for nuclei detection. The data is randomly sampled from 49 16x24 well plates containing a combination of 6 different ovarian cell lines. 25,472 images are nucleus instance masks generated from the DAPI training and validation sets via Cellpose [117]. To obtain high-quality masks, we apply the same gamma correction of 0.3 applied in previous Chapters before inputting the images into Cellpose [117]. We then use the same evaluation pipeline as in Chapter 5, Figure 5.3 with Cellprofiler [20] that takes the DAPI fluorescence images and masks as input to compute instance-wise scores for a collection of 18 biological features frequently used in HTS. The types of features include a range of intensity and radial distribution measurements. For Bright2Bio-Net, we utilise the same pipeline to

extract the biological feature values for the training and validation sets. We also compute the distance transform for the Cellpose instance masks.

The remaining 291 images are comprised of 97 bright-field, 97 DAPI fluorescence and 97 Cellpose masks used as a test set sampled from a single plate not used in any training of validation sets.

Training & Inference: This work uses the pix2pixHD [24] architecture adapted for previous virtual staining works [86] in Chapters 4 and 5 with the use of MC-Dropout [132] to enable sampling of virtual staining solutions. The network was trained on the full 1080x1080 image resolution with random affine data augmentation applied during data loading. The network was trained using the same hyperparameters outlined in Chapter 5 for a maximum of 200 epochs using early stopping. Previous works [48, 133–135] have shown the superiority of Diffusion-based models compared to GANs for a range of image-related tasks. Unlike diffusion, the distribution learnt by GANs has been shown to suffer from mode collapse [35], so evaluating two sampling methods with different characteristics should enable a deeper understanding of how our proposed evaluation metrics work.

De-noising Diffusion Probabilistic model (DDPM) [28] based on its interpretation ability and performance for image generative tasks. DDPM adapted to work for conditional image generation by performing the forward pass on the fluorescence image and then conditioning the decoder on the bright-field at each time point t during the reverse process. Conditional DDPM (cDDPM) was trained on random crops of 256x256, with random affine data augmentation applied. A learning rate of 0.00001 with an exponential moving average decay was used alongside an Adam [136] optimiser. cDDPM was based on the U-Net [49], starting with 64 filters and the number of filters increasing by 2^n to a depth of 4. Denoising Diffusion Implicit (DDIM) sampling [84] was used to improve inference speed,d reducing the number of sampling steps from 1,000 to 250. The network was trained for a maximum of 200 epochs with a batch size of 16 using early stopping. 1000 time steps were used, Both baselines were trained to translate from bright-field to 6-diamidino-2-phenylindole (DAPI) nuclei on the same image pairs.

During inference, for each test set bright-field image Bright2Bio-Net outputs predicted

biological feature posterior distributions, while both baseline virtual staining methods generated 1,000 virtual staining samples, to which nuclei instance masks and biological features were extracted to approximate the same biological feature posterior distributions. The samples were then processed as shown in Figure 6.1b to produce instance-wise predicted posterior distributions over 18 biological features (see Appendix C Table C.1 for the full breakdown of features).

In practice, to ensure an accurate approximation of the predicted posterior via sampling, only instances that occurred in all 1,000 samples for both baselines were evaluated. This ensures that when exploring the proposed evaluation metrics, the results are not skewed by insufficient sampling. The population distribution of a feature for each method is defined as the collection of all 1,000 samples across all instances.

6.2.2 Can the log-likelihood, standard score & percentile rank computed over the predicted posterior distributions reveal new insights compared to population metrics?

In this section we explore the performance across the predicted sample population for cDDPM and pix2pixHD using two population measurements; the Wdist between the target and predicted population distributions and the KLD between a mixture of Gaussian model fitted to the same population distributions.

The number of Gaussians was determined for each feature based on the ELBO of the Akaike Information Criterion (AIC) and the Bayes Information Criterion (BIC) obtained on the real feature population. The ELBO describes the point at which the gradient of the BIC and AIC scores with respect to an increasing number of components begins to level out. Typically, both AIC and BIC would behave similarly. However, when this was not the case because the BIC ($\log \text{likelihood} + \text{k} \log(N)$) takes into consideration the sample size N where AIC ($\log \text{likelihood} + 2\text{k}$) does not and given this was being computed on the entire population with a large sample size we take the number of Gaussians at the ELBO of BIC.

The population measurement results for cDDPM and pix2pixHD are then compared to

the newly proposed instance posterior distribution measurements discussed in Section 6.2; the average log-likelihood, the Wdist of the standard scores and $\mathcal{N}(0, 1)$, and the Wdist of the percentile ranks and $\mathcal{U}(0, 100)$.

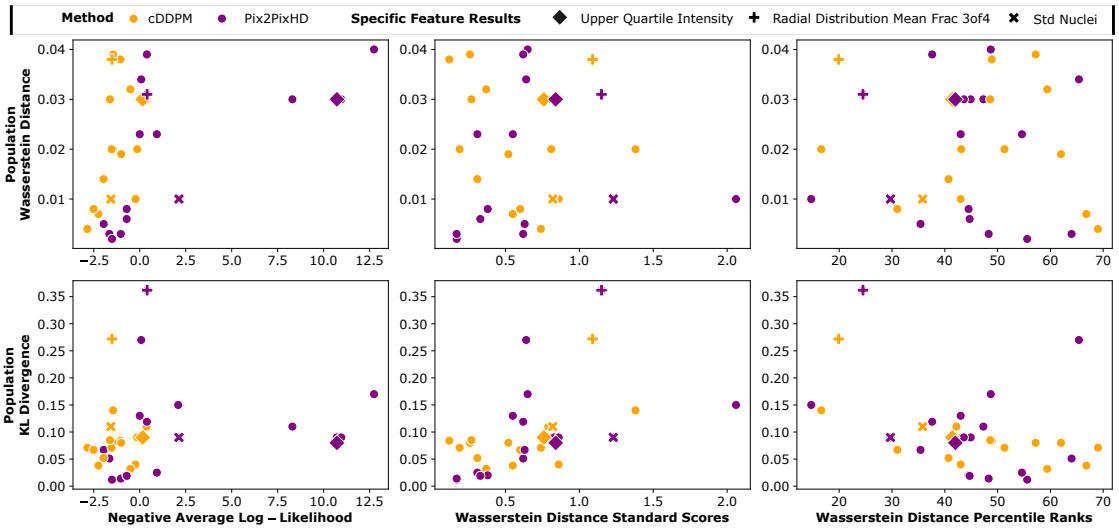


Figure 6.4: **Scatter plots of cDDPM and Pix2pixHD performance across all biological features for both population measurements and the three proposed measurement combinations. We highlight three example features and show the performance of both methods.** For the negative average log-likelihood (NLL), there is a shift in clustering of results in the x-axis that can not be seen in the y-axis. This suggests that for a number of biological features, there are things that can not be observed when evaluating at the population level compared to when using the NLL to evaluate the posterior distributions at the instance level. In particular, this can be seen in the "Upper Quartile Intensity" and the "Std Intensity" features. A similar but less extreme pattern can be observed for the Wdist of the standard scores, where the performance gap between cDDPM and Pix2pixHD is considerably larger on the x-axis compared to the y-axis.

Figure 6.4 shows the relationship between the performance of cDDPM and pix2pixHD for both population measurements and the proposed average log-likelihood, Wdist of standard scores and Wdit of percentile ranks. For the proposed average log-likelihood, there is a clear right shift along the x-axis of pix2pixHD results compared to cDDPM that can not be seen in the y-axis. In particular, the highlighted "Upper Quartile Intensity", "Std Intensity" and "Radial Distribution Mean Frac 3of4" show very small differences in performance along the population measurements (y-axis) relative to the difference in average log-likelihood (x-axis). This suggests that for several features, when evaluating only at the population level cDDPM and pix2pixHD perform similarly but when evaluating at the instance level using the average log-likelihood, there appear to be performance

differences that would otherwise go unnoticed using population measurements. This pattern, although less obvious, can also be seen in the results for the WDist of standard scores, where the scale of the performance difference is larger in the Wdist of standard scores (x-axis) than in the population measurements (y-axis).

How can it be that the performance at the population level is different from the instance level and what can the proposed measurements tell us about why this is happening?

To explore these questions, we now deep-dive into the results for a subset of biological features. Figure 6.5 visualises the distribution of results for both population measurements and the proposed instance posterior distribution measurements as well as each measurement statistic.

For the previously mentioned "Upper Quartile Intensity" feature, when measuring the performance across the two population measurements, for both the Wdist and the KLD pix2pixHD performs better than cDDPM, producing a Wdist of 0.029 compared to 0.031 for cDDPM and a KLD of 0.08 compared to 0.09 for cDDPM. Visually, the population distribution of cDDPM appears to have more values on its right tail as well as less of a central peak compared to pix2pixHD, the μ of the population distribution for cDDPM is 1.18 ± 0.8 compared to 1.20 ± 1.07 for pix2pixHD and 1.21 ± 1.14 for the target distribution.

In contrast, when looking at the distribution of log-likelihoods, computed over the instance posterior distributions, pix2pixHD has a considerably longer left tail compared to cDDPM suggesting for many instances, the probability of the true sample value occurring within the predicted posterior is very low.

The highlighted average log-likelihood of cDDPM is considerably higher than pix2pixHD, despite cDDPM producing lower Wdist and KLD at the population level. For a range of additional intensity and radial distribution features within Figure 6.5, we observe a similar pattern of small performance differences at the population level despite several pix2pixHD instance posterior predictions producing low log-likelihood results (shown in the left tails).

These findings led to the following questions: Are there some biological feature values found in fluorescence microscopy that never occur in the pix2pixHD virtual staining

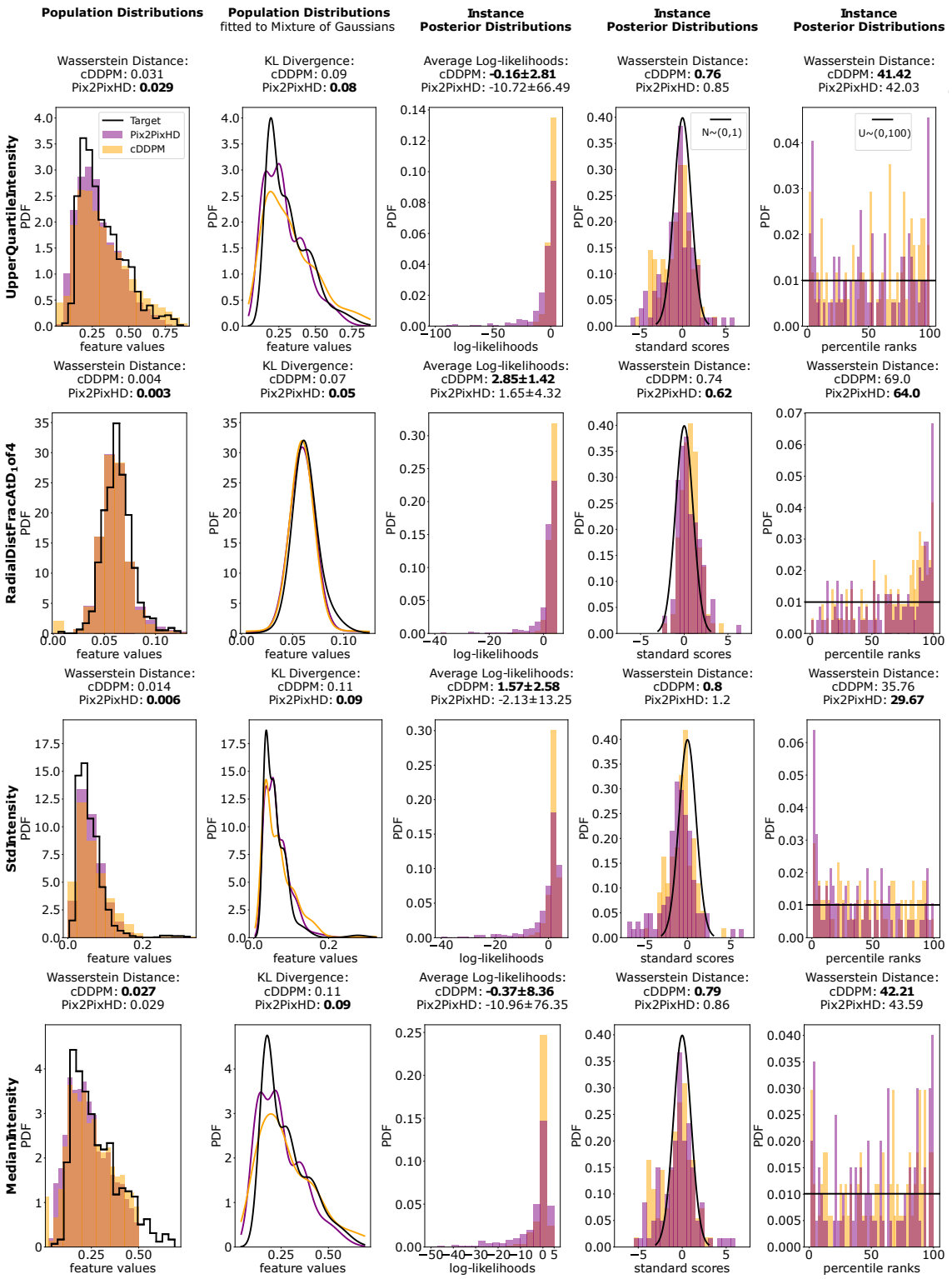


Figure 6.5: **Feature population distribution measurements vs. the proposed instance feature posterior distribution measurements.** Population distributions and statistics are shown in columns 1 and 2 and the proposed instance posterior distributions and statistics are shown in columns 3, 4 and 5. Log-likelihood left tails have been truncated for visualisation purposes. The population measurements show similar performance for both cDDPM and pix2pixHD. The Log-likelihood distributions show considerably longer left tails for pix2pixHD compared to cDDPM, showing at the population level cDDPM and pix2pixHD appear very similar, but at the instance level, this appears not to be the case. Similar but less extreme patterns can be seen in the standard score distributions and the spikes at 0 and 100 in the percentile rank distributions.

samples that do occur in cDDPM virtual staining samples? In other words, is pix2pixHD unable to produce specific biological features in its virtual staining samples that cDDPM can? Or are they just occurring for the wrong instances?

In the case where the biological feature values never occur for pix2pixHD, we would observe poor performance when using both the traditional population measurements as well as our newly proposed instance posterior criterion. However, this is not the case.

What these results provide evidence of is that while pix2pixHD appears to produce an approximately correct distribution of feature values at the population level, paradoxically we observe for several instance feature values this is not the case, suggesting that pix2pixHD is predicting the instance feature values but not for the right instances.

In principle, the average log-likelihood and the KLD should be the same up to a shift as derived in Equation 6.9. As an additional experiment for both cDDPM and pix2pixHD, we compute the log-likelihoods of the target value according to the Gaussian mixture model fitted to the predicted population distribution. Figure 6.6 explicitly compares the distribution of log-likelihoods computed over the population to the log-likelihood results computed over the instance posterior distributions. Here, we can directly compare both log-likelihood distributions for a subset of the biological features.

Looking at the population log-likelihoods, as already shown for the Wdist and KLD computed over the population distributions in columns 1 and 2 of Figure 6.5 pix2pixHD performs reasonably compared to cDDPM. We can observe features such as "Std Intensity" where cDDPM produces an average population log-likelihood of 2.1 ± 0.72 compared to pix2pixHD which produces 2.13 ± 1.22 or the "Median Intensity" feature where cDDPM produces an average population log-likelihood of 0.69 ± 0.5 compared to pix2pixHD which produces 0.71 ± 0.77. For the same features, when the log-likelihood is computed on the instance posterior distributions, we can observe in row 1 compared to row 2 of Figure 6.6 the longer left tail of the negative log-likelihoods for the instance posterior distributions compared to the concentrated population distributions. On average, we can observe that for pix2pixHD, the instance posterior log-likelihood results for the previously mentioned "Std Intensity" and "Median Intensity" are -2.13 ± 13.25 and -10.96 ± 76.35. It can also

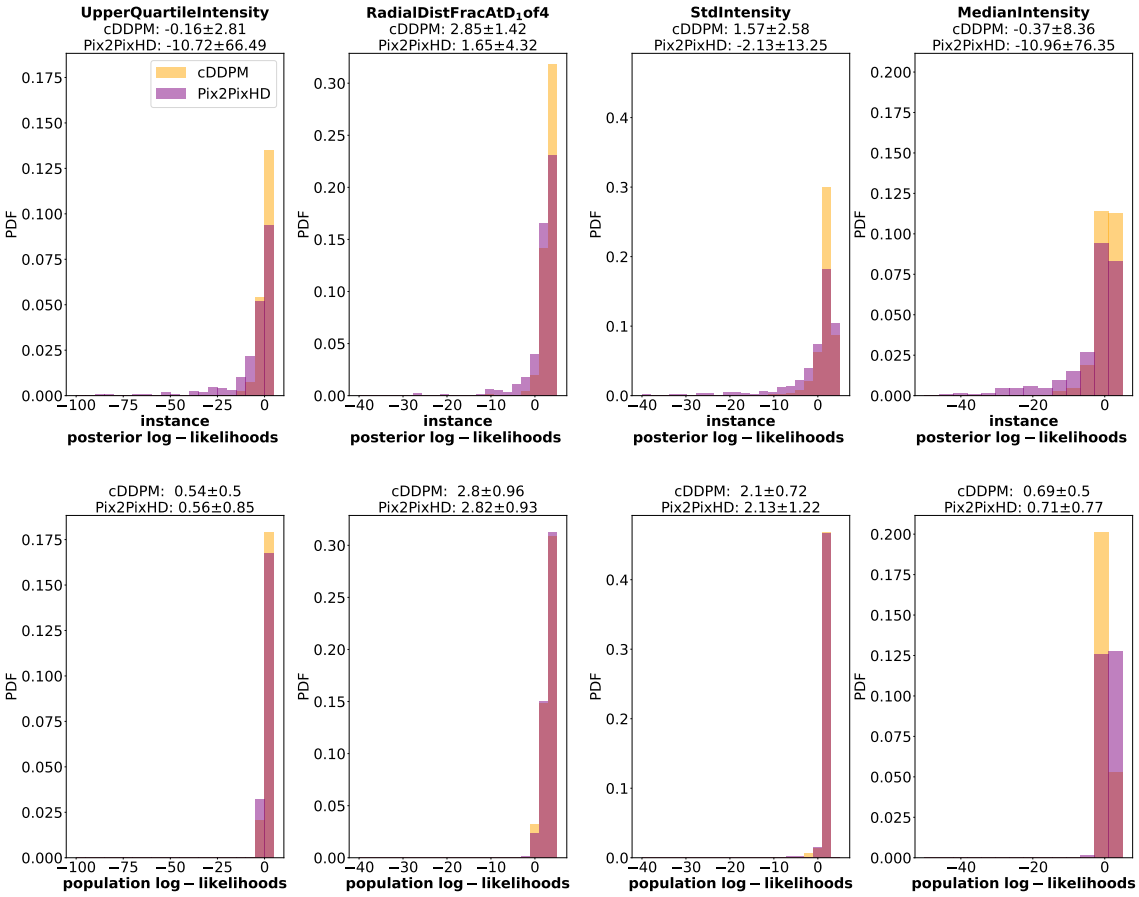


Figure 6.6: **Log-Likelihood of Predicted Posteriors vs Log Log-likelihood over Population**. For multiple biological features, we can observe that when we compute the log-likelihood over the population, we see reasonable performance for pix2pixHD, but when we compute the log-likelihood over each instance pix2pixHD in many cases is unable to produce an accurate prediction for the predicted posterior. cDDPM also shows a similar pattern but the scale of the difference is far greater for pix2pixHD.

be seen for cDDPM for the "Median Intensity" feature that despite having higher instance posterior log-likelihoods compared to pix2pixHD, when evaluating at the instance level we still observe considerably longer tails compared the population log-likelihoods showing for many instances the probability of the true sample value occurring within the predicted posterior of cDDPM is very low. By directly comparing the log-likelihoods at the instance level to those at the population level, we can explicitly observe that when evaluating generative models, while the output may be accurate at the population level, it may not be correct for individual instances, and this discrepancy often goes unnoticed when only considering population statistics. We discuss whether population-level statistics alone are enough when making decisions in section 6.3.

Looking at the proposed Wdist of standard scores and the standard score distributions in column 4 of Figure 6.5, although less pronounced than the log-likelihood distributions shown in column 3, the longer left distribution tails for pix2pixHD compared to cDDPM can also be seen. For example, the "Std Intensity" and the "Median Intensity" features show evidence that for many instances, the true feature value is considerably different (more than ± 5 σ) from the predicted posterior distribution of pix2pixHD. Yet, when we look at both the scale of the difference between the Wdist of standard scores and the average log-likelihood as well as the long tails of the two distributions for pix2pixHD we observe considerably larger differences for the log-likelihood. In principle, the instances that have produced the long left tail of log-likelihoods should correspond to the instances that are outside of the $\mathcal{N}(0, 1)$ in the standard score plots. Consistently, across all features, the average log-likelihood in Figure 6.5 column 3 is always higher for cDDPM, while this is not the case for the Wdist of standard scores as shown for the "Radial Distribution FracAtd 1of4" in Figure 6.5 column 4 and for 3 other features in Appendix C Table C.1. This suggests that the log-likelihood is more heavily penalising some of the target values compared to the standard score.

One thing to note is that this long tail can not be seen in the percentile rank distribution due to the hard cut-off at 0 and 100. When using the percentile rank, for target feature values that fall outside of the predicted posterior distribution it is not possible to know how far away the real sample value is from the predicted posterior beyond it being greater than the largest value or smaller than the smallest value. This leads to the peeks of values at 0 and 100 as can be observed in all the percentile rank distribution plots in the final column of Figure 6.5. The measurements' inability to quantify the full scale of performance could negatively impact its interpretation, as can be observed for the "Std Intensity" in Figure 6.5. Here, the Wdist of percentile ranks for pix2pixHD is lower than that of cDDPM which also aligns with the population statistics in columns 1 and 2 despite cDDPM having a lower Wdist of standard scores and a higher average log-likelihood. This is because the Wdist of percentile ranks is treating a large number of values with rank 0 (shown by the peak at 0) to be the same.

Having identified the long left tail of the log-likelihood distributions, how do the predicted posterior distributions of instances along these tails compare to the target feature value and qualitatively how do the virtual staining samples compare to the target fluorescence? Figure 6.7 highlights for the "Upper Quartile Intensity" and "Radial Distribution Mean Frac 3of4" features three instance posterior log-likelihood results within the log-likelihood distribution, the corresponding instance within the bright-field input, the corresponding instance predicted posterior distributions as well as the instance highlighted in the median sample for cDDPM and pix2pixHD and the target fluorescence image (additional feature results can be found in the Appendix C Figure C.1 and C.2)). For each row, the instance reviewed scores a progressively improved log-likelihood for pix2pixHD as shown by the dashed purple line compared to cDDPM shown by a dashed orange line.

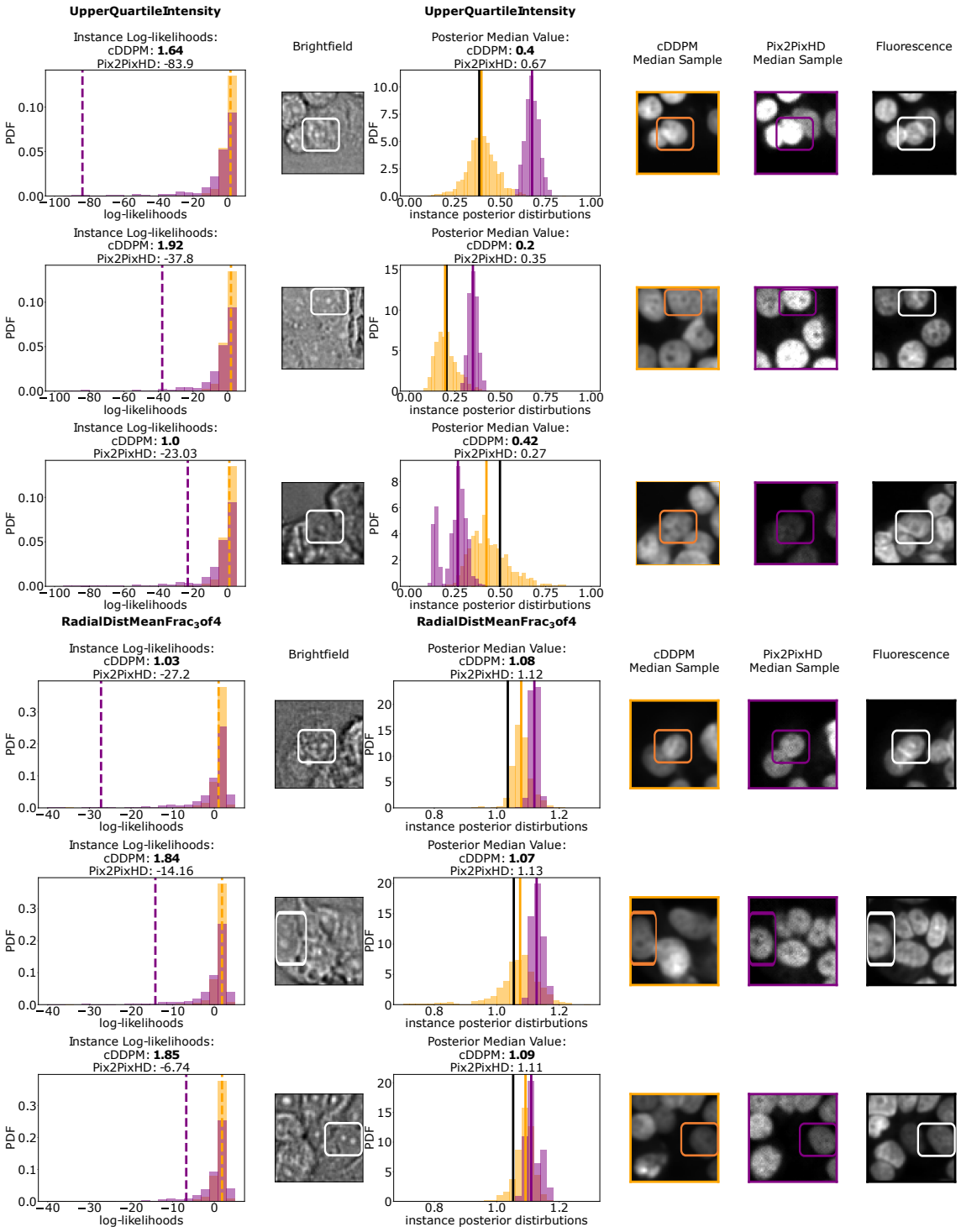


Figure 6.7: **Individual instance feature performance vs qualitative results**. For each row for both cDDPM and pix2pixHD, we highlight the instance result within the distribution of log-likelihoods, the bright-field input and corresponding instance, and the predicted instance posterior distributions. We also highlight the median values of both methods versus the target value and qualitatively show the median virtual staining instance sample versus the target fluorescence instance. Each feature contains three rows, each with improved instance log-likelihood results for pix2pixHD. The predicted posterior distributions of cDDPM are much wider than pix2pixHD and are always closer to the target value. We can observe for both feature types as the log-likelihood improves, the pix2pixHD predicted posterior distribution moves closer to the target value. Qualitatively, we can see that for both features pix2pixhd either over and under-saturates the instance intensity compared to cDDPM.

In general, across both features the posterior distributions of pix2pixHD progressively move closer to the target value as the log-likelihood result improves. The posterior distributions of pix2pixHD are much more narrow than cDDPM which aligns with known challenges with cGANs of mode collapse [35].

When looking at the predicted posterior distribution in Figure 6.7 column 3 of pix2pixHD, for the lowest log-likelihood results pix2pixHD completely misses the target value while cDDPM is far closer. The median value of cDDPM is also much closer to the target compared to pix2pixHD. Visually, for the "Upper Quartile Intensity" and the "Radial Distribution FractAtd 3of4" features we can observe the median sample of pix2pixHD is more saturated than cDDPM which appears to more closely resemble the target instance intensity.

Looking at the improved log-likelihood results in Figure 6.7 from row 1 to row 3 for each feature we can observe the predicted posterior distributions of pix2pixHD move towards the target value but across all examples, we still observe under or over-saturation of intensity values compared to cDDPM which produces an over smoothed intensity profile. In particular, for the "Radial Distribution FractAtd 3of4" it is the width of the distribution that leads to an improved log-likelihood score compared to pix2pixHD.

What these findings show is that pix2pixHD does produce the correct distribution of biological feature values at the population level, but when we focus on values specific to individual instances we observe for several different features and instances pix2pixHD is far less able to reproduce a predicted posterior distribution that fits well to the target feature value. This suggests that pix2pixHD is unable to know for which bright-field instance it should produce the biological feature values but in general knows it should occur with a specific distribution across the population. pix2pixHD often gets the per instance feature prediction wrong, but coincidentally across the entire instance population the value exists. When we evaluate only at the population level this can not be observed and can mislead us into believing that the biological feature values are very similar when in fact on an instance basis this is not the case.

6.3 Discussion

Here, we have presented the concept of Bright2Bio-Net a novel idea for predicting instance-wise biological feature posterior distributions given a label-free bright-field input. The early exploration of Bright2Bio-Net showcases ideas for an alternative approach to previous virtual staining works that directly predicts instance-wise posterior distributions of biological features without the need for virtual staining or external evaluation pipelines. Future work will look to evaluate the performance of this method compared to the explored virtual staining sampling methods.

Additionally, we have presented three new evaluation criteria; the average log-likelihood, the WDist of standard scores and $\mathcal{N}(0, 1)$ and the Wdist of percentile ranks with $\mathcal{U}(1, 100)$, all of which provide a quantitative similarity measurement between a predicted posterior distribution and a *single* true posterior sample. In general, we observe qualitatively and quantitatively that cDDPM performs better than pix2pixHD in generating accurate instance posterior distributions. When examining the proposed evaluation criterion a high-level Figure 6.4 showed when evaluating the two explored generative models; cDDPM and pix2pixHD the average log-likelihood revealed performance insights that would otherwise go unnoticed using population measurements. Examining the distribution of log-likelihoods showed long tails of the distribution for a broad range of intensity and radial distribution features for pix2pixHD compared to cDDPM showing that the probability of numerous true sample values occurring within the predicted posterior of pix2pixHD was very low. Using the same log-likelihood measurement on the population as explored on the instance posterior distributions directly showed the difference in performance outcomes depending on whether evaluating at the population or the instance level.

What these findings show is that paradoxically, when evaluating generative models the predictions may be accurate at the population level but are not always accurate for individual instances across a number of biological features. The question then becomes whether performance at the population level is enough? Is it the case for some situations that it is? Perhaps, but in principle, knowing that the performance is good at the population level because subsets of the population distribution, such as the individual instance

posterior distributions, are accurate, can only result in as good or even better population statistics. The ability to evaluate instance biological feature posterior distributions could improve our understanding of virtual staining performance in unseen populations of cells and aid in the identification of rare events that can not be observed just by looking at the population distributions.

Although less prominent, similar long tails shown in the log-likelihood distributions could be observed in the distribution of standard scores with a number of instances producing standard scores $\pm 5\,\sigma$ from the target value despite very similar population-level performance. Unfortunately, the same clear pattern could not be seen for the percentile rank distributions due to the hard cut-off at 0 and 100 which limited the ability of the measurement to quantify how far above or below the true sample was from the predicted posterior distribution.

Why is it that the log-likelihood appears to penalise the target values more than the standard score and percentile rank? In the extreme case, if the generative model ignores the input and instead of predicting a posterior distribution predicts the population distribution then of course the target value will sit within the predicted "posterior" distribution and the standard scores would fit a $\mathcal{N}(0,1)$ and percentile ranks would fit a $\mathcal{U}(0,100)$ producing Wdist scores of 0.

If we use the analogy of a simplified roulette table, where the population corresponds to the total number of possible bets which would be values from 0-36. Theoretically, predicting the population can be viewed as equally spreading your total bet across the 37 numbers, corresponding to a wider distribution with lower reward and lower punishment. In comparison, placing the same total bet on a subset of numbers could produce a higher reward but also a higher punishment. This spread can be viewed as the σ in the standard score shown in Equation 6.10 which normalises the bet with respect to the risk taken when placing the bet. Similarly, the percentile rank can only evaluate the bet within the spread of the bet itself (1,100). In contrast, for the average log-likelihood as formulated in Equation 6.9 no normalisation is specific to the bet of the predicted posterior distribution. Instead, the correct normalisation is the unknown *constant* term; the negative entropy

of the true posterior $P(\mathcal{Y}|\mathbf{x}_j)$ which is why it is more heavily penalising the predicted posterior distribution compared to the standard score and percentile rank measurements.

In addition, because the constant term is the same for all generative models (as derived in section 6.2) the metric still holds when comparing different generative models. However, because it is unknown, the metric is unable to, in absolute terms, quantify how close the predicted posterior distribution is to the target feature value. Future work could explore the use of the negative entropy of the predicted posterior distribution, but as previously mentioned, this would then restrict the measurement as derived in Equation 6.9 from being able to compare different generative models.

Future work could also explore rather than computing the Wdist of the standard scores and a $\mathcal{N}(0,1)$ instead of computing the KLD of the standard scores and a $\mathcal{N}(0,1)$. In comparison to the Wdist, the KLD would produce much higher distance scores, for the pix2pixHD instance values outside of the target $\mathcal{N}(0,1)$ shown for features such as the "Std Intensity" or the "Median Intensity" features in Figure 6.5 column 4 which could produce similar long tails as shown in the Figure 6.5 column 3.

One question that remains is how do the proposed measurements perform when we have access to the true posterior distribution rather than just a single sample? This would then make the unknown constant known, enabling the average log-likelihood to provide results in absolute terms. It would also mean that the standard score and percentile rank measurements could be reformulated so that the normalisation is computed over the σ of the true posterior distribution.

Future work could explore a simulated regime in which we will build a probabilistic bright-field simulator that generates bright-field images containing biological features on which we can use Bayes theorem to obtain an analytical solution to the simulated target posterior distribution. We can then compute either the KLD between the predicted and simulated true posterior distributions and compare the results to our proposed measurements to see if the results are similar. Furthermore, future work will extend the proposed measurements to other inverse problems such as image de-noising to determine the suitability of the measurements in evaluating generative models for other tasks.

Chapter 7

Conclusion

At the start of this research, high-throughput screening pipelines mainly relied on fluorescence staining to quantify cellular characteristics to compare drug effects across compound perturbations. These methods, while effective, were costly, time-consuming, and prone to plate effects when repeated across multiple time points. Meanwhile, image-to-image translation methods were gaining attention in the natural image domain for their visually striking results, but their potential for translating label-free bright-field images into virtual fluorescence channels, what we term virtual staining, remained largely unexplored.

This thesis represents a comprehensive exploration of the evaluation of different image-to-image translation methods used for virtual staining within high-throughput screening (HTS). We present a number of novel quantitative measurements for measuring the performance of virtual staining as well as a new method for directly predicting biological feature distributions from label-free bright-field images. We provide analysis with mathematical derivations to compare and contrast three commonly used architectures and loss functions within Chapter 2 and provide a cohesive review of the literature on the many variants in Chapter 3. In Chapter 1, we introduce three questions in sections 1.3, 1.4 and 1.5 that this thesis addresses in Chapters 4, 5 and 6 . Those questions are now reinstated along with summarised answers.

7.0.1 Can virtual staining accurately represent the biological information within individual cells found in the fluorescence channels?

Within Chapter 4 we benchmark a number of image-to-image translation methods for virtual staining and present a novel evaluation pipeline that evaluates the performance of virtual staining at three levels; pixel/patch level, biological feature level and drug effect.

We demonstrate that not only do virtual nuclei and virtual cytoplasm staining predictions have high pixel-level results and visually striking appearance, but the samples can also reproduce biologically meaningful information found in the fluorescence channels and lead to relatively small changes in drug effect outcomes compared to fluorescence channels. In contrast, virtually stained DNA-damage showed high prediction variability, unrealistic biological features and notable inaccuracies in drug effect.

The evaluation techniques and findings present important steps forward in quantifying the suitability of different virtual staining tasks as a viable imaging alternative to fluorescence staining within HTS, leading to a number of tangible benefits. Firstly, the use of virtual nuclei and cytoplasm models instead of fluorescence channels enables the quantification of dynamic live cell bright-field imaging over time. Secondly, it reduces the number of HTS plates required for nuclei and cytoplasm staining from the total number of incubation periods to a single plate, saving time and unwanted plate variability. Thirdly, substituting fluorescence nuclei and cytoplasm markers for virtual staining can free up fluorescence channel space, enabling new multi-plex imaging combinations. Finally, the time required to perform nuclei and cytoplasm staining of HTS plates reduces from ~7 days to minutes.

The techniques presented to evaluate virtual staining and the findings shown within this thesis have led to the adoption of virtual nuclei and cytoplasm models within GSK for HTS and have also been extended to other image modalities, such as tissue imaging.

7.0.2 Can virtual staining models for HTS generalise?

Having trained and tested virtual staining models on subsets of HTS data from a single biological setting in Chapter 4, in Chapter 5, we explored whether particular subsets of HTS data, when used as training sets, could lead to more generalisable virtual staining models. We explored three generalisation tasks: generalising to new phenotypes, generalising to new cell types and generalising to new cell types and phenotypes combined. After conducting a large-scale investigation, we identified several interesting findings.

Firstly, across all three levels of evaluation, cell types, and virtual nuclei and cytoplasm staining tasks, we found that when testing on toxic samples, training on non-toxic samples can not only generalise well, but also lead to improved results for several metrics compared to training on toxic samples of the same cell type. We believe these results are driven by the heterogeneity of cell expression that occurs in non-toxic healthy cells compared to the more homogeneous cell states in toxic samples, leading to less diversity in training data and subsequently worse generalisation performance.

Secondly, when generalising to new cell types, we found the task to be more complex, with different outcomes depending on the level of evaluation used, good generalisation to one cell type was not indicative of good generalisation to another cell type and consistently breast cells were challenging to generalise too and not good for training generalisble models. We believe that structural differences across cell types, such as the anatomical function of breast cells or the endocrine signalling [128] of ovarian cells, make for a much more challenging generalisation task.

Finally, for the final generalisation task, we observed similar challenges when training on and generalising to images of breast cells for the virtual nuclei and virtual cytoplasm. However, in general, across all levels of evaluation and the explored cell type and phenotype combinations, we find that training on non-toxic is preferable even when additionally generalising to toxic samples of an unseen cell type.

As non-toxic samples are typically more readily available and less expensive to acquire compared to specific toxic samples, this could lead to a reduction in the barriers to entry for HTS labs looking to apply virtual staining on specific toxic phenotype datasets.

Furthermore, in the context of GSK it led to a 50% reduction in the number of setting-specific virtual staining models necessary to train and maintain, whilst also providing valuable feedback into the ongoing strategy of data generation for future virtual staining models.

7.0.3 Can virtual staining models correctly capture the distribution of possible fluorescence images and biological feature values for each cell?

Despite the scale of the generalisation experiments in Chapter 5 and the use of pix2pixHD, to the virtual staining tasks, the evaluation of virtual staining performance was based on quantifying how a single virtual staining sample performed when compared to the corresponding fluorescence image. However, virtual staining is an ill-posed problem, meaning there is a posterior distribution of possible virtual staining solutions that could have come from the bright-field.

In Chapter 6 we explored the performance of the predicted posterior distributions of biological features computed via multiple virtual staining samples for the already explored pix2pixHD with the use of MC-dropout [132] and a more recent sampling-based generative model, cDDPM [28]. However, one challenge remained, given any kernel density estimate, how can we evaluate a predicted posterior distribution with access to only a single sample from the target posterior distribution?

We presented three novel evaluation metrics; the average log-likelihood, the Wasserstein distance (Wdist) of the standard scores and $\mathcal{N}(0,1)$ and the Wdist of the percentile ranks and $\mathcal{U}(1,100)$. We showed that the average log-likelihood is the same as the KLD between the predicted and true posterior distributions up to a constant shift that is independent of the predicted posterior distribution. We then compared the performance of our proposed instance posterior measurements to commonly used population measurements. We identified for pix2pixHD GAN that, despite similar population distributions to the ground truth population (as shown in Chapter A 4.7 and in Chapter 6 Figure 6.6), for many cells and biological features the probability of numerous true sample values

occurring within the predicted cell posterior of Pix2pixHD was very low. This suggests that the model, in many cases, can reproduce the biological features, but does not know for which instance within the bright-field input it should occur. Paradoxically, this could not be observed when looking at the population distribution alone, only from our newly proposed measurements.

The findings from our proposed measurements, improve our understanding of virtual staining performance at an instance level and can be applied to any kernel density estimator. We also posed a further question of can we bypass virtual staining and design a method to directly predict the distribution of biological features for each cell from the bright-field? We presented early principled works into the design of a novel deep-learning based method: Bright2Bio-Net, which is able to perform this task.

7.1 Future work

7.1.1 Further development and evaluation of Bright2Bio-Net

Chapter 6 presents the early development of Brigh2Bio-Net, a novel architecture based on a convolutional neural network. Given a bright-field as input, the model directly predicts instance-wise posterior distributions for a range of biological features. Future work would benefit from developing this method further and evaluating its performance using the evaluation metrics explored within Chapter 6 compared to the indirect sampling methods explored within this thesis. If this approach yields promising results, it could be used in conjunction with or to replace virtual staining as a faster way to quantify biological information within cells.

7.1.2 Further evaluation of proposed measurements in Chapter 6 section 6.2 using a simulated posterior distribution

The three metrics proposed within chapter 6 section 6.2 used to evaluate cDDPM and pix2pixHD could be further validated by exploring a simulated regime. Currently, the

metrics are designed to quantify the quality of a predicted posterior distribution with access to only one sample from the true posterior distribution. By using simulation, we would be able to analytically compute the true posterior of a feature of interest and compare it to the predicted posterior distribution. This would also allow us to further check if the results of the proposed measurements correspond to the results obtained using the simulation and if the difference between he true KLD and the equation 6.5 corresponds to a constant shift.

To achieve this, we could define prior probability distributions $P(\mathcal{Y}_{i,j})$ over each biological feature. Since we're operating in a simulated environment, we know the analytical likelihood function $P(\tilde{x}_{i,j}|\mathcal{Y}_{i,j})$ of observing a bright-field sample given feature values sampled from the prior probabilities. This could include rendering the cell, applying filters to simulate optics as well as a statistical model of the noise. Combining both the prior and likelihood function using Bayes theorem we can transform $P(\tilde{x}_{i,j}|\mathcal{Y}_{i,j})$ and obtain a true biological feature posterior distribution for our simulation

$$P(\mathcal{Y}_{i,j}|\tilde{x}_{i,j}) = \frac{P(\tilde{x}_{i,j}|\mathcal{Y}_{i,j}) * P(\mathcal{Y}_{i,j})}{P(\tilde{x}_{i,j})} \tag{7.1}$$

References

1. Szymański, P., Markowicz, M. & Mikiciuk-Olasik, E. Adaptation of high-throughput screening in drug discovery—toxicological screening tests. *International journal of molecular sciences* **13,** 427–452 (2011).

2. Bradbury, S. & Evennett, P. J. *Contrast techniques in light microscopy* (Garland Science, 2020).

3. Ding, K.-F., Finlay, D., Yin, H., Hendricks, W. P., Sereduk, C., Kiefer, J., Sekulic, A., LoRusso, P. M., Vuori, K., Trent, J. M., *et al.* Analysis of variability in high throughput screening data: applications to melanoma cell lines and drug responses. *Oncotarget* **8,** 27786 (2017).

4. Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F. & Johnson, G. R. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature Methods* **15,** 917–920 (2018).

5. Harrison, P. J., Gupta, A., Rietdijk, J., Wieslander, H., Carreras-Puigvert, J., Georgiev, P., Wählby, C., Spjuth, O. & Sintorn, I.-M. Evaluating the utility of brightfield image data for mechanism of action prediction. *PLOS Computational Biology* **19,** e1011323 (2023).

6. Gupta, A., Harrison, P. J., Wieslander, H., Rietdijk, J., Puigvert, J. C., Georgiev, P., Wählby, C., Spjuth, O. & Sintorn, I.-M. Is brightfield all you need for mechanism of action prediction? *bioRxiv,* 2022–10 (2022).

7. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. *Image-to-image translation with conditional adversarial networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 1125–1134.

8. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D. & Norouzi, M. *Palette: Image-to-image diffusion models* in *ACM SIGGRAPH 2022 conference proceedings* (2022), 1–10.

9. Borji, A. Pros and cons of GAN evaluation measures. *Computer vision and image understanding* **179,** 41–65 (2019).

10. Christiansen, E. M., Yang, S. J., Ando, D. M., Javaherian, A., Skibinski, G., Lipnick, S., Mount, E., O'neil, A., Shah, K., Lee, A. K., *et al.* In silico labeling: predicting fluorescent labels in unlabeled images. *Cell* **173,** 792–803 (2018).

11. Cross-Zamirski, J. O., Mouchet, E., Williams, G., Schönlieb, C.-B., Turkki, R. & Wang, Y. Label-free prediction of cell painting from brightfield images. *Scientific reports* **12,** 10001 (2022).

12. Cross-Zamirski, J. O., Anand, P., Williams, G., Mouchet, E., Wang, Y. & Schönlieb, C.-B. Class-Guided Image-to-Image Diffusion: Cell Painting from Brightfield Images with Class Labels. *arXiv preprint arXiv:2303.08863* (2023).

13. High throughput and high content screening microscopy: A microscopy-based guideline. *European Pharmaceutical Review* **14,** 26–30 (2009).

14. Pirone, D., Lim, J., Merola, F., Miccio, L., Mugnano, M., Bianco, V., Cimmino, F., Visconte, F., Montella, A., Capasso, M., *et al.* Stain-free identification of cell nuclei using tomographic phase microscopy in flow cytometry. *Nature photonics* **16,** 851–859 (2022).

15. Yingxue, P., Jianxin, L., Tao, Q. & Zhibo, C. Image-to-image translation: Methods and applications. *arXiv preprint arXiv:2101.08629* (2021).

16. Stephens, D. J. & Allan, V. J. Light microscopy techniques for live cell imaging. *science* **300,** 82–86 (2003).

17. Faragallah, O. S., El-Hoseny, H., El-Shafai, W., Abd El-Rahman, W., El-Sayed, H. S., El-Rabaie, E.-S. M., Abd El-Samie, F. E. & Geweid, G. G. A comprehensive survey analysis for present solutions of medical image fusion and future directions. *IEEE Access* **9,** 11358–11371 (2020).

18. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13,** 600–612 (2004).

19. Imboden, S., Liu, X., Payne, M. C., Hsieh, C.-J. & Lin, N. Y. Trustworthy in silico cell labeling via ensemble-based image translation. *Biophysical Reports* **3** (2023).

20. Carpenter, A. E. e. a. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology* **7,** 1–11 (2006).

21. Elmer, P. *Columbus Image Data Storage and Analysis System* https://www.perkinelmer.com/uk/product/image-data-storage-and-analysis-system-columbus. Accessed: 10/10/2022.

22. Bishop, C. M. & Nasrabadi, N. M. *Pattern recognition and machine learning* **4** (Springer, 2006).

23. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. *Unpaired image-to-image translation using cycle-consistent adversarial networks* in *Proceedings of the IEEE international conference on computer vision* (2017), 2223–2232.

24. Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J. & Catanzaro, B. *High-resolution image synthesis and semantic manipulation with conditional gans* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 8798–8807.

25. Kingma, D. P. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

26. Larsen, A. B. L., Sønderby, S. K., Larochelle, H. & Winther, O. *Autoencoding beyond pixels using a learned similarity metric* in *International conference on machine learning* (2016), 1558–1566.

27. Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. & Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

28. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33,** 6840–6851 (2020).

29. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. *Deep unsupervised learning using nonequilibrium thermodynamics* in *International conference on machine learning* (2015), 2256–2265.

30. Wilson, J. E., Brown, D. E. & Timmens, E. K. A Toxicologic Study of Dimethyl Sulfoxide. *Toxicology and Applied Pharmacology* **7,** 104–12 (Jan. 1965).

31. Kobayashi, K. & Ratain, M. J. Pharmacodynamics and long-term toxicity of etoposide. *Cancer Chemotherapy and Pharmacology* **34 Suppl,** S64–8 (1994).

32. Qiao, L., Koutsos, M., Tsai, L. L., Kozoni, V., Guzman, J., Shiff, S. J. & Rigas, B. Staurosporine inhibits the proliferation, alters the cell cycle distribution and induces apoptosis in HT-29 human colon adenocarcinoma cells. *Cancer letters* **107,** 83–9 (Oct. 1996).

33. Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

34. Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* **28** (2015).

35. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C. & Bengio, Y. *Generative Adversarial Nets* in *NIPS* (2014), 2672–2680.

36. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J. & Norouzi, M. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence* **45,** 4713–4726 (2022).

37. Das, A. *An Introduction to Diffusion Probabilistic Models* https://ayandas.me/blogs/2021-12-04-diffusion-prob-models.html.

38. Gal, Y. *et al.* Uncertainty in deep learning (2016).

39. Armanious, K., Jiang, C., Fischer, M., Küstner, T., Hepp, T., Nikolaou, K., Gatidis, S. & Yang, B. MedGAN: Medical image translation using GANs. *Computerized medical imaging and graphics* **79,** 101684 (2020).

40. Zhao, H., Li, H. & Cheng, L. Synthesizing filamentary structured images with GANs. *arXiv preprint arXiv:1706.02185* (2017).

41. Zhang, C., Tang, Y., Zhao, C., Sun, Q., Ye, Z. & Kurths, J. Multitask gans for semantic segmentation and depth completion with cycle consistency. *IEEE Transactions on Neural Networks and Learning Systems* **32,** 5404–5415 (2021).

42. Van Den Oord, A., Vinyals, O., *et al.* Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017).

43. Razavi, A., Van den Oord, A. & Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems* **32** (2019).

44. Chen, Y.-J., Cheng, S.-I., Chiu, W.-C., Tseng, H.-Y. & Lee, H.-Y. *Vector quantized image-to-image translation* in *European Conference on Computer Vision* (2022), 440–456.

45. Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. *Self-Attention Generative Adversarial Networks* in *Proceedings of the 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) **97** (PMLR, Sept. 2019), 7354–7363.

46. Brock, A. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096* (2018).

47. Nichol, A. Q. & Dhariwal, P. *Improved denoising diffusion probabilistic models* in *International conference on machine learning* (2021), 8162–8171.

48. Dhariwal, P. & Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* **34,** 8780–8794 (2021).

49. Ronneberger, O., Fischer, P. & Brox, T. *U-net: Convolutional networks for biomedical image segmentation* in *International Conference on Medical image computing and computer-assisted intervention* (2015), 234–241.

50. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. *The cityscapes dataset for semantic urban scene understanding* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 3213–3223.

51. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. *Imagenet: A large-scale hierarchical image database* in *2009 IEEE conference on computer vision and pattern recognition* (2009), 248–255.

52. Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

53. Van Den Oord, A., Kalchbrenner, N. & Kavukcuoglu, K. *Pixel recurrent neural networks* in *International conference on machine learning* (2016), 1747–1756.

54. Salimans, T., Karpathy, A., Chen, X. & Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517* (2017).

55. Child, R., Gray, S., Radford, A. & Sutskever, I. Generating long sequences with sparse transformers. arXiv 2019. *arXiv preprint arXiv:1904.10509* (1904).

56. Rezende, D. & Mohamed, S. *Variational inference with normalizing flows* in *International conference on machine learning* (2015), 1530–1538.

57. Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* **31** (2018).

58. Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016).

59. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. *Rethinking the inception architecture for computer vision* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 2818–2826.

60. Vahdat, A. & Kautz, J. NVAE: A deep hierarchical variational autoencoder. *Advances in neural information processing systems* **33,** 19667–19679 (2020).

61. Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O. & Shechtman, E. Toward multimodal image-to-image translation. *Advances in neural information processing systems* **30** (2017).

62. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S. & Klambauer, G. Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling* **58,** 1736–1741 (2018).

63. Mittal, A., Soundararajan, R. & Bovik, A. C. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters* **20,** 209–212 (2012).

64. Liu, M.-Y., Breuel, T. & Kautz, J. Unsupervised image-to-image translation networks. *Advances in neural information processing systems* **30** (2017).

65. Arjovsky, M., Chintala, S. & Bottou, L. *Wasserstein generative adversarial networks* in *International conference on machine learning* (2017), 214–223.

66. Johnson, J., Alahi, A. & Fei-Fei, L. *Perceptual losses for real-time style transfer and super-resolution* in *European conference on computer vision* (2016), 694–711.

67. Burt, P. J. & Adelson, E. H. in *Readings in computer vision* 671–679 (Elsevier, 1987).

68. Brown, M., Lowe, D. G., *et al. Recognising panoramas.* in *ICCV* **3** (2003), 1218.

69. Shah, S., Ghosh, P., Davis, L. & Goldstein, T. Stacked U-Nets: A no-frills approach to natural image segmentation. arXiv 2018. *arXiv preprint arXiv:1804.10343*.

70. Upadhyay, U., Chen, Y., Hepp, T., Gatidis, S. & Akata, Z. *Uncertainty-guided progressive GANs for medical image translation* in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24* (2021), 614–624.

71. Karras, T. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196* (2017).

72. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems* **30** (2017).

73. Kodali, N., Abernethy, J., Hays, J. & Kira, Z. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215* (2017).

74. Mescheder, L., Geiger, A. & Nowozin, S. *Which training methods for GANs do actually converge?* in *International conference on machine learning* (2018), 3481–3490.

75. Wang, X., Girshick, R., Gupta, A. & He, K. *Non-local neural networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 7794–7803.

76. Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. *Inception-v4, inception-resnet and the impact of residual connections on learning* in *Proceedings of the AAAI conference on artificial intelligence* **31** (2017).

77. Özbey, M., Dalmaz, O., Dar, S. U., Bedel, H. A., Özturk, Ş., Güngör, A. & Çukur, T. Unsupervised medical image translation with adversarial diffusion models. *IEEE Transactions on Medical Imaging* (2023).

78. Ho, J. & Salimans, T. Classifier-free diffusion guidance. *arXiv preprint 2207.12598* (2022).

79. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-resolution image synthesis with latent diffusion models* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), 10684–10695.

80. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J. & Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

81. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

82. Peebles, W. & Xie, S. *Scalable diffusion models with transformers* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 4195–4205.

83. Chandrasekaran, S. N., Ackerman, J., Alix, E., Ando, D. M., Arevalo, J., Bennion, M., Boisseau, N., Borowa, A., Boyd, J. D., Brino, L., *et al.* JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *BioRxiv,* 2023–03 (2023).

84. Song, J., Meng, C. & Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

85. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y. & Cohen, I. Pearson correlation coefficient. *Noise reduction in speech processing,* 1–4 (2009).

86. Tonks, S., Hsu, C., Hood, S., Musso, R., Hopely, C., Doan, M., Edwards, E., Krull, A. & Styles, I. *Evaluation of virtual staining for high-throughput screenings* in *20th IEEE International Symposium on Biomedical Imaging* (2023).

87. Tolstikhin, I. O., Gelly, S., Bousquet, O., Simon-Gabriel, C.-J. & Schölkopf, B. Adagan: Boosting generative models. *Advances in neural information processing systems* **30** (2017).

88. Van den Oord, A. & Dambre, J. *Locally-connected transformations for deep gmms* in *International Conference on Machine Learning (ICML): Deep learning Workshop* (2015), 1–8.

89. Theis, L., Oord, A. v. d. & Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).

90. Uria, B., Murray, I. & Larochelle, H. *A deep and tractable density estimator* in *International Conference on Machine Learning* (2014), 467–475.

91. Van den Oord, A. & Schrauwen, B. Factoring variations in natural images with deep gaussian mixture models. *Advances in neural information processing systems* **27** (2014).

92. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. Improved techniques for training gans. *Advances in neural information processing systems* **29** (2016).

93. Betzalel, E., Penso, C., Navon, A. & Fetaya, E. A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935* (2022).

94. Barratt, S. & Sharma, R. A note on the inception score. *preprint arXiv:1801.01973* (2018).

95. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017).

96. Lucic, M., Kurach, K., Michalski, M., Gelly, S. & Bousquet, O. Are gans created equal? a large-scale study. *Advances in neural information processing systems* **31** (2018).

97. Liu, S., Wei, Y., Lu, J. & Zhou, J. An improved evaluation framework for generative adversarial networks. *arXiv preprint arXiv:1803.07474* (2018).

98. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M. & Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717* (2018).

99. Soloveitchik, M., Diskin, T., Morin, E. & Wiesel, A. Conditional frechet inception distance. *arXiv preprint arXiv:2103.11521* (2021).

100. Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O. & Gelly, S. Assessing generative models via precision and recall. *Advances in neural information processing systems* **31** (2018).

101. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J. & Aila, T. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems* **32** (2019).

102. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

103. Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y. & Yoo, J. *Reliable fidelity and diversity metrics for generative models* in *International Conference on Machine Learning* (2020), 7176–7185.

104. Cheema, F. & Urner, R. *Precision recall cover: A method for assessing generative models* in *International Conference on Artificial Intelligence and Statistics* (2023), 6571–6594.

105. Wagner, S. J., Matek, C., Shetab Boushehri, S., Boxberg, M., Lamm, L., Sadafi, A., Waibel, D. J., Marr, C. & Peng, T. Make deep learning algorithms in computational pathology more reproducible and reusable. *Nature Medicine* **28,** 1744–1746 (2022).

106. Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P. & Kelley, D. R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods* **18,** 1196–1203 (2021).

107. Mediratta, I., You, Q., Jiang, M. & Raileanu, R. The Generalization Gap in Offline Reinforcement Learning. *arXiv preprint arXiv:2312.05742* (2023).

108. Cobbe, K., Klimov, O., Hesse, C., Kim, T. & Schulman, J. *Quantifying generalization in reinforcement learning* in *International conference on machine learning* (2019), 1282–1289.

109. Levine, S., Kumar, A., Tucker, G. & Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

110. Van der Laak, J., Litjens, G. & Ciompi, F. Deep learning in histopathology: the path to the clinic. *Nature medicine* **27,** 775–784 (2021).

111. Echle, A., Rindtorff, N. T., Brinker, T. J., Luedde, T., Pearson, A. T. & Kather, J. N. Deep learning in cancer pathology: a new generation of clinical biomarkers. *British journal of cancer* **124,** 686–696 (2021).

112. Chandrasekaran, S. N. e. a. Jump cell painting dataset: Morphological impact of 136,000 chemical and genetic perturbations. *BioRxiv* (Jan. 2023).

113. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R. & Kim, K. *Image to image translation for domain adaptation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 4500–4509.

114. Luo, Y., Liu, P., Guan, T., Yu, J. & Yang, Y. Adversarial style mining for one-shot unsupervised domain adaptation. *Advances in neural information processing systems* **33,** 20612–20623 (2020).

115. Real, R. & Vargas, J. M. The probabilistic basis of Jaccard's index of similarity. *Systematic biology* **45,** 380–385 (1996).

116. Sasaki, Y. e. a. The truth of the F-measure. *Teaching, Tutorial Materials, Version: 26th October* **1,** 1–5 (2007).

117. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods* **18,** 100–106 (2021).

118. Zhang, J.-H., Chung, T. D. & Oldenburg, K. R. A simple statistical parameter for use in evaluation and validation of high throughput screening assays. *Journal of biomolecular screening* **4,** 67–73 (1999).

119. Kendall, A. & Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems* **30** (2017).

120. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15,** 1929–1958 (2014).

121. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M. & Aila, T. Noise2Noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189* (2018).

122. Tonks, S., Nguyer, C., Hood, S., Musso, R., Hopely, C., Titus, S., Doan, M., Styles, I. & Krull, A. Can virtual staining for high-throughput screening generalize? *arXiv preprint arXiv:2407.06979* (2024).

123. Wieslander, H., Gupta, A., Bergman, E., Hallström, E. & Harrison, P. J. Learning to see colours: Biologically relevant virtual staining for adipocyte cell images. *Plos one* **16,** e0258546 (2021).

124. Jaccard, P. The distribution of the flora in the alpine zone. 1. *New Phytologist* **11,** 37–50 (1912).

125. Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., *et al.* Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. *Nature methods* **16,** 1247–1253 (2019).

126. Saabas, A. *Selecting Good Features - Part III: Random Forests* Data Science Central Blog. https://blog.datadive.net/selecting-good-features-part-iii-random-forests/.

127. Ellis, H. & Mahadevan, V. Anatomy and physiology of the breast. *Surgery (Oxford)* **31,** 11–14 (2013).

128. Cooper, G. M. The Cell: A Molecular Approach. *2nd edition. Sunderland (MA): Sinauer Associates; 2000.Signaling Molecules and Their Receptors.* (2000).

129. Rajaram, S., Pavie, B., Wu, L., *et al.* PhenoRipper: software for rapidly profiling microscopy images. *Nature Methods* **9,** 635–637 (2012).

130. Razghandi, M., Zhou, H., Erol-Kantarci, M. & Turgut, D. *Variational autoencoder generative adversarial network for synthetic data generation in smart home* in *ICC 2022-IEEE International Conference on Communications* (2022), 4781–4786.

131. Rüschendorf, L. The Wasserstein distance and approximation theorems. *Probability Theory and Related Fields* **70,** 117–129 (1985).

132. Gal, Y. & Ghahramani, Z. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning* in *international conference on machine learning* (2016), 1050–1059.

133. Mazé, F. & Ahmed, F. *Diffusion models beat gans on topology optimization* in *Proceedings of the AAAI conference on artificial intelligence* **37** (2023), 9108–9116.

134. Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P.-A. & Li, S. Z. A Survey on Generative Diffusion Models. *IEEE Transactions on Knowledge and Data Engineering* **36,** 2814–2830 (2024).

135. Stypułkowski, M., Vougioukas, K., He, S., Zięba, M., Petridis, S. & Pantic, M. *Diffused heads: Diffusion models beat gans on talking-face generation* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2024), 5091–5100.

136. Zhang, Z. *Improved adam optimizer for deep neural networks* in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)* (2018), 1–2.

137. Palanisamy, G., Ponnusamy, P. & Gopi, V. P. An improved luminosity and contrast enhancement framework for feature preservation in color fundus images. *Signal, Image and Video Processing* **13,** 719–726 (2019).

Appendix A

Appendix A: Additional data for Chapter 4

A.1 Evaluation of Cellpose masks for use in virtual staining evaluation pipeline

Cellpose [117] provides a generalist segmentation solution that accurately delineates cell bodies, membranes, and nuclei across diverse microscopy image types without retraining or parameter tweaking. Utilising the pre-trained nuclei and cytoplasm models within our evaluation pipeline (shown in Figure 4.4 & Figure 5.3), enabled the cell nuclei and cell cytoplasm of virtual staining predictions and fluorescence microscopy images to be detected and then evaluated. However, by using these models, we introduce additional models into our pipeline therefore, we felt it necessary to systematically evaluate Cellpose model performance to ensure that it did not skew our downstream results.

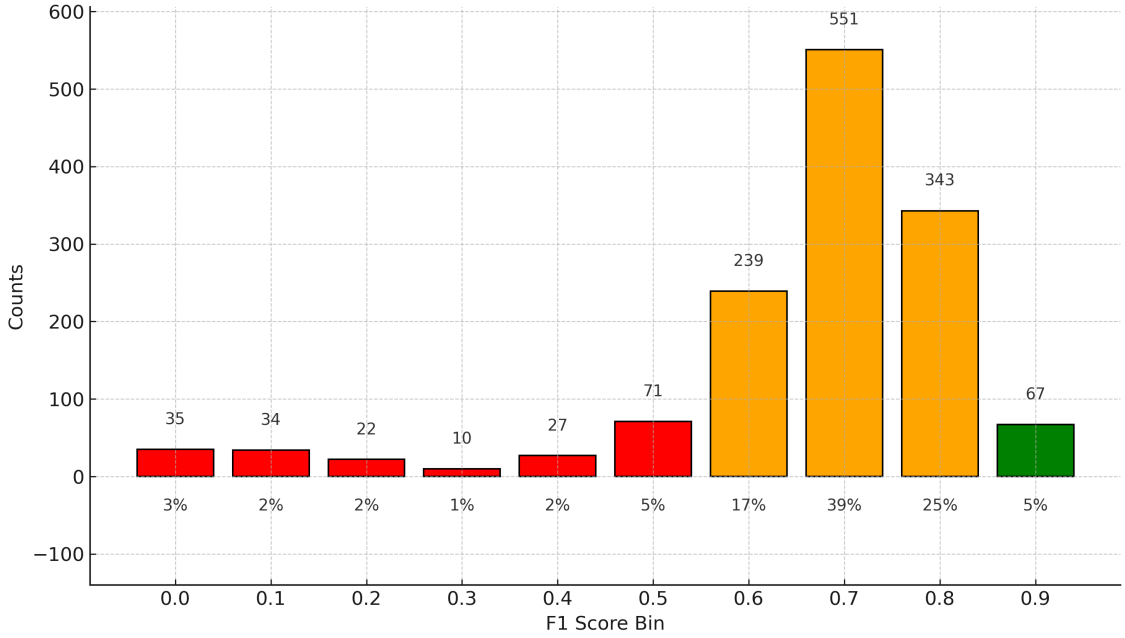


Figure A.1: **Histogram of F1 scores for virtual nuclei staining of images of lung cell lines.** We randomly sampled images from the bottom 15% of F1 scores shown in red. On several examples, we identified errors coming from the masks generated by Cellpose.

In Chapter 4 Table 4.1, we show the nuclei segmentation results for Pix2PixHD GAN trained on bright-field and DAPI image pairs from lung cell lines. The results show an average F1 score of 0.70 ± 0.18. Despite this being promising, we examine the distribution of F1 score results as a histogram in Figure A.1 to further understand what the types of error are that are causing the lowest scoring results. After randomly sampling images from the bottom 15% of F1 scores shown in red we identified errors not coming from the virtual

staining but from the masks generated by CellPose. In Figure A.2 (shown in yellow), we observe areas where Cellpose has not been able to detect the object in both cases, this is an error in Cellpose, but it doesn't adversely impact our virtual staining evaluation. However, as shown in blue, when the mask obtained from the ground truth image is missing cells that can clearly be seen in the ground truth fluorescence and have otherwise been detected from the mask obtained from the virtual stain this negatively impacted our results. When evaluated against the ground truth mask, these cells are classified as false positives when they are in fact true positives.

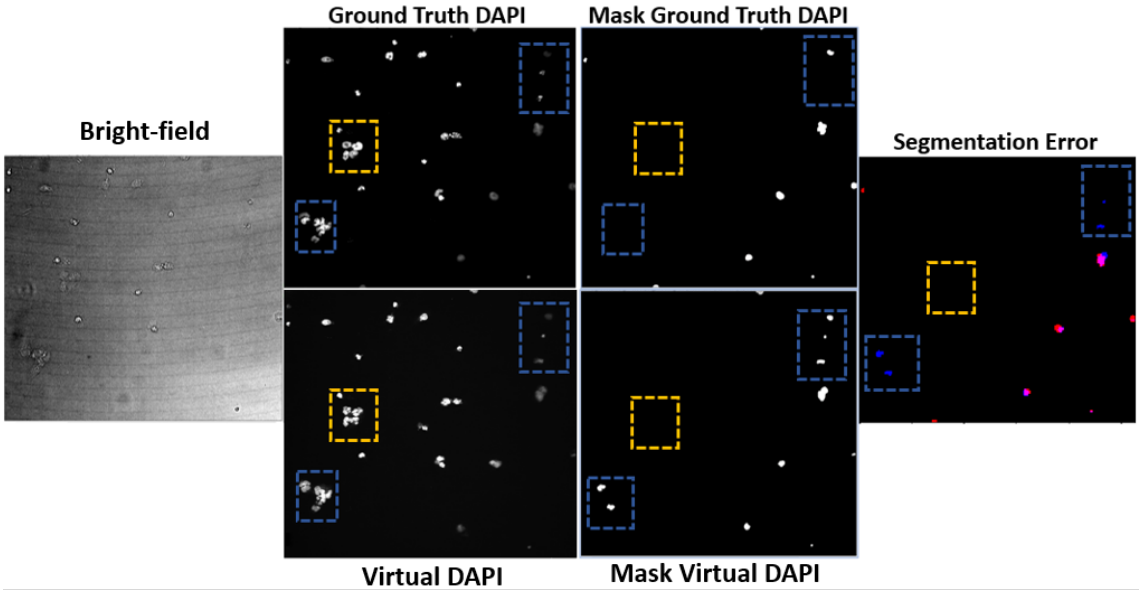


Figure A.2: **Qualitatively show two types of error coming from Cellpose masks.** Show a randomly sampled bright-field input image, the corresponding ground truth DAPI stain and the Cellpose mask obtained from the ground truth DAPI stain as well as the virtual DAPI stain and the corresponding Cellpose mask. The pre-trained nuclei Cellpose model was used for both inputs. We then show the segmentation error. We highlight two types of error, firstly, in yellow, where there exists a prediction in both ground truth and virtual DAPI stain, but no nuclei masks are predicted. Secondly, highlighted in blue, we show two examples where the mask obtained from the ground truth stain is missing nuclei that have otherwise been detected from the mask obtained from the virtual DAPI stain. When evaluated against the ground truth mask, these cells are classified as false positives (shown in blue) when they are true positives, which negatively skews the results.

Based on these findings we further evaluated the masks generated from Cellpose using ground truth DAPI fluorescence images. We uniformly sampled 1,152 DAPI and CellPose mask patches of size 70x70 across 3 plates (1 sample from each of the 384 wells) and manually labelled the mask and image pair as "Accept" or "Reject" based on whether

they correctly identified all nuclei in the DAPI image. We identified on average, CellPose segmentation masks were rejected for 11% ± 1% of cases. Upon examining the rejected

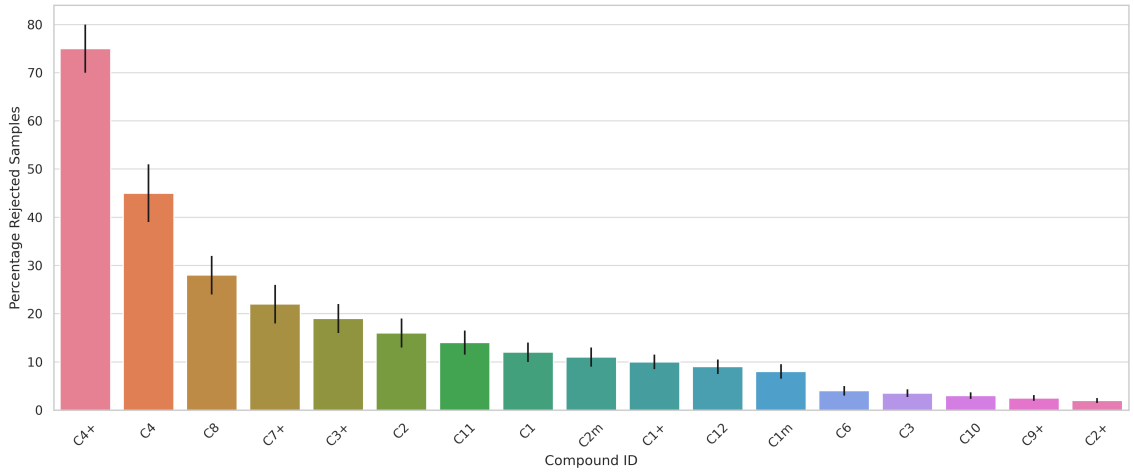


Figure A.3: **Percentage of rejected samples by compound.** After evaluating 1,152 uniformly sampled DAPI and Cellpose mask pairs across a fixed compound layout of 3 plates, we identified an 11% ± 1% rejection rate. Here we plot the rejection percentage per compound and show over 74% rejection for C4+ and 37% for C4. Removal of these compounds led to a 0.04 increase in average F1 score to 0.74. However, segmenting these compounds was critical for the experiment, so we proceeded to explore alternative solutions.

11% we identified that the proportion of rejected samples out of the total samples for each compound varied considerably (as shown in Figure A.3). We identified that 74% of samples from compound C4+ and 37% of samples from compound C4 were rejected. Removal of these examples led to a 0.04 improvement in F1 score to 0.74 but segmenting of these compounds was critical for the experiment, so an alternative solution was required to reduce the rejection rate.

We explored several possible preprocessing tasks that included transforming the Cellpose input from 16-bit to 8-bit to help bring the distribution closer to a Gaussian, but saw no improvement. We then explored the preprocessing step of gamma correction [137], which applies a nonlinear transformation to the pixel values where a value < 1 brightens dark regions and compresses highlights, reducing intensity variance. Figure A.4 qualitatively shows for randomly selected rejected C4+ samples the improvement in the Cellpose mask after performing gamma correction ($\gamma = 1/3$). Based on these findings, we conducted further analysis to observe how the use of gamma correction across all compounds would impact the rejection rating of 11%. Through the use of a GUI (as shown in Figure A.5),

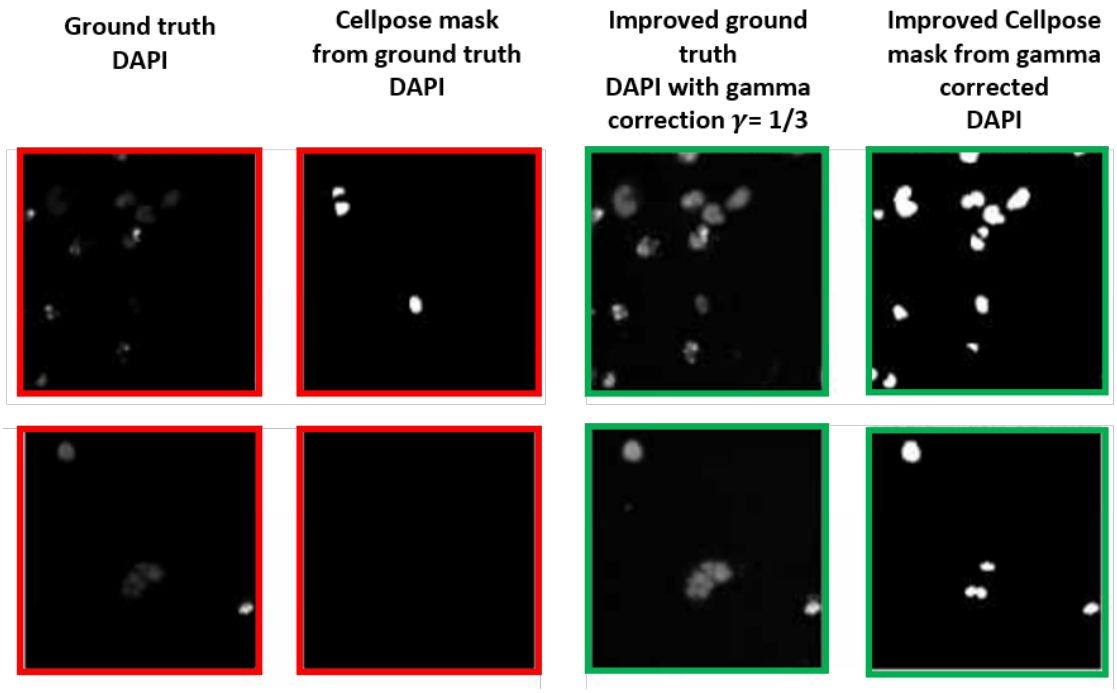


Figure A.4: **Qualitative improvement in Cellpose Mask generation when using a gamma correction of 1/3 as a preprocessing step for the DAPI input image.** We show two examples of the poorly performing C4+ compound, where by using a gamma correction of 1/3, we show considerable improvements in the generated Cellpose mask.

1,152 DAPI image patches were re-sampled, each DAPI image sample was displayed along with a gamma corrected DAPI image and a Cellpose mask. The mask displayed could be either the mask obtained from the original DAPI image or the gamma corrected image, with the likelihood of either being equal.

For the 1,152 image patches, 85 were rejected, and of those 85, only 10 rejections came from the gamma-corrected mask, producing on average a 2% rejection rate (10/576). For the 10 gamma rejections, 8/10 came from the C4 and C4+ compounds previously identified. The rejection rate for C4+ reduced from 74% to 20% and the rejection rate for C4 reduced from 37% to 11%.

In conclusion, by using a gamma correction of γ = 1/3 we were able to systematically reduce the rejection rate from 11% to 2% which meant that with 98% confidence in the Cellpose mask and the use of gamma correction we were able to improve the F1 score from 0.70 ± 0.18 to 0.81 ± 0.17. The gamma correction improved the intensity profile of cell nuclei in both the ground truth images and virtual staining, which reduced the misclassification of nuclei as shown as discussed in Figure A.2, ultimately leading to more

reliable and accurate object-level performance evaluation. These tests were replicated

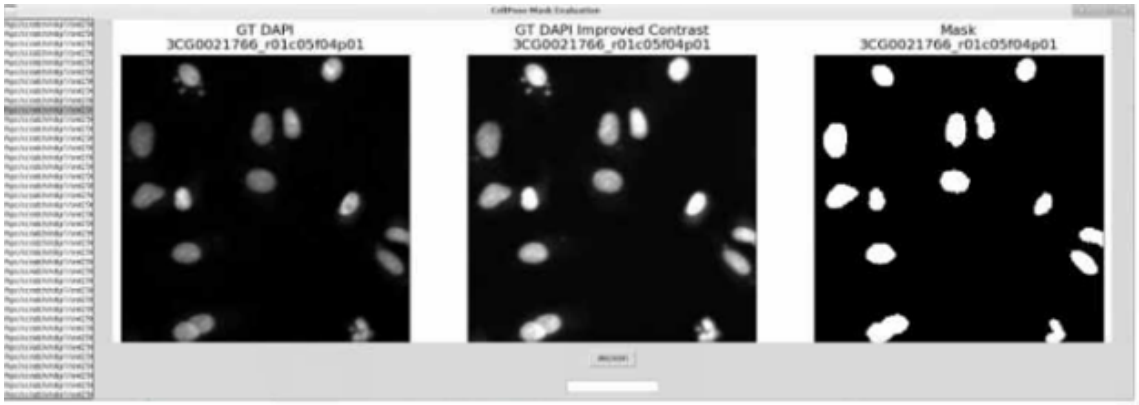


Figure A.5: **Evaluating the use of gamma correction as a preprocessing step for improving Cellpose mask generation.** For 1,000 images, a patch from the ground truth DAPI and the gamma corrected DAPI were displayed along with a Cellpose mask. The mask could have been generated from either dapi image (the likelihood of each being equal) but this was not known. The user was asked to "Accept" or "Reject" the mask based on either dapi image.

with breast and ovarian cell lines. For breast cell line images, an average of 15% of masks were rejected, and only 4% of those came from the gamma corrected mask, while for ovarian, 14% of masks observed were rejected, and only 3% of those rejected came from the gamma corrected mask. This conclusively demonstrates that with a gamma correction of $\gamma = 1/3$, we can be confident that the Cellpose outputs are representative and that our object-level metrics are calculated correctly.

Appendix B

Appendix B: Additional data for Chapter 5

ID	stain	seed	train cell type	train phenotype	test cell type	test phenotype	PSNR	SSIM	F1 score	Jaccard Index
1	fitc	42	breast	nontoxic	breast	nontoxic	35 ± 2.15	0.84 ± 0.04	0.29 ± 0.21	0.39 ± 0.26
2	fitc	123	breast	nontoxic	breast	nontoxic	35.1 ± 2.2	0.81 ± 0.05	0.32 ± 0.18	0.43 ± 0.27
3	fitc	42	breast	nontoxic	breast	toxic	35.6 ± 2.36	0.8 ± 0.10	0.44 ± 0.16	0.58 ± 0.11
4	fitc	123	breast	nontoxic	breast	toxic	35.1 ± 2.39	0.75 ± 0.12	0.43 ± 0.15	0.57 ± 0.11
5	fitc	72	breast	nontoxic	breast	toxic	35.8 ± 2.77	0.85 ± 0.05	0.46 ± 0.16	0.59 ± 0.12
6	fitc	72	breast	nontoxic	breast	nontoxic	30.4 ± 3.3	0.9 ± 0.06	0.41 ± 0.14	0.53 ± 0.13
7	fitc	72	breast	toxic	breast	toxic	31.5 ± 3.5	0.91 ± 0.05	0.42 ± 0.15	0.54 ± 0.13
8	fitc	42	breast	nontoxic	lung	nontoxic	33 ± 2.34	0.74 ± 0.08	0.17 ± 0.16	0.34 ± 0.25
9	fitc	123	breast	nontoxic	lung	nontoxic	34.2 ± 2.98	0.74 ± 0.12	0.24 ± 0.16	0.46 ± 0.18
10	fitc	42	breast	nontoxic	lung	toxic	35.6 ± 3.38	0.74 ± 0.12	0.31 ± 0.18	0.52 ± 0.17
11	fitc	123	breast	nontoxic	lung	toxic	34.4 ± 2.95	0.67 ± 0.14	0.33 ± 0.17	0.54 ± 0.16
12	fitc	123	breast	nontoxic	ovarian	nontoxic	35.1 ± 2.28	0.75 ± 0.08	0.18 ± 0.15	0.33 ± 0.25
13	fitc	42	breast	nontoxic	ovarian	nontoxic	37.2 ± 2.1	0.8 ± 0.07	0.34 ± 0.15	0.46 ± 0.21
14	fitc	72	lung	nontoxic	breast	toxic	33.8 ± 2.78	0.72 ± 0.10	0.34 ± 0.16	0.53 ± 0.12
15	fitc	72	lung	nontoxic	breast	nontoxic	26.9 ± 3.6	0.83 ± 0.09	0.3 ± 0.15	0.5 ± 0.15
16	fitc	72	lung	toxic	breast	toxic	27.5 ± 3.6	0.86 ± 0.06	0.31 ± 0.14	0.52 ± 0.13
17	fitc	42	breast	nontoxic	ovarian	toxic	36 ± 2.47	0.71 ± 0.13	0.27 ± 0.13	0.5 ± 0.12
18	fitc	123	breast	nontoxic	ovarian	toxic	34.9 ± 2.24	0.63 ± 0.16	0.28 ± 0.12	0.5 ± 0.11
19	fitc	42	lung	nontoxic	breast	nontoxic	34 ± 2.3	0.66 ± 0.12	0.25 ± 0.11	0.4 ± 0.21
20	fitc	123	lung	nontoxic	breast	nontoxic	33.1 ± 2.27	0.69 ± 0.10	0.2 ± 0.17	0.35 ± 0.24
21	fitc	123	lung	nontoxic	breast	toxic	33.3 ± 2.64	0.64 ± 0.14	0.31 ± 0.15	0.51 ± 0.11
22	fitc	42	lung	nontoxic	breast	toxic	32.8 ± 2.47	0.61 ± 0.15	0.34 ± 0.16	0.52 ± 0.12
23	fitc	72	ovarian	toxic	breast	toxic	27.4 ± 3.9	0.82 ± 0.08	0.27 ± 0.14	0.47 ± 0.15
24	fitc	42	lung	nontoxic	lung	nontoxic	34.2 ± 2.99	0.79 ± 0.10	0.27 ± 0.22	0.38 ± 0.27
25	fitc	123	lung	nontoxic	lung	nontoxic	33.1 ± 2.78	0.81 ± 0.12	0.38 ± 0.21	0.5 ± 0.3
26	fitc	72	ovarian	nontoxic	breast	toxic	34.2 ± 2.61	0.76 ± 0.08	0.37 ± 0.15	0.53 ± 0.12
27	fitc	72	ovarian	nontoxic	breast	nontoxic	26.6 ± 3.6	0.82 ± 0.08	0.25 ± 0.13	0.44 ± 0.14
28	fitc	42	lung	nontoxic	lung	toxic	36.4 ± 3.66	0.77 ± 0.14	0.55 ± 0.19	0.63 ± 0.15
29	fitc	123	lung	nontoxic	lung	toxic	35.8 ± 4.04	0.75 ± 0.17	0.49 ± 0.18	0.59 ± 0.14
30	fitc	42	lung	nontoxic	ovarian	nontoxic	35.6 ± 2.74	0.78 ± 0.09	0.27 ± 0.22	0.36 ± 0.28
31	fitc	123	lung	nontoxic	ovarian	nontoxic	34.2 ± 3.9	0.8 ± 0.08	0.31 ± 0.21	0.41 ± 0.18
32	fitc	42	lung	nontoxic	ovarian	toxic	35.6 ± 3.04	0.68 ± 0.17	0.44 ± 0.15	0.58 ± 0.11
33	fitc	123	lung	nontoxic	ovarian	toxic	35.9 ± 3.46	0.7 ± 0.18	0.37 ± 0.15	0.54 ± 0.11
34	fitc	123	ovarian	nontoxic	breast	nontoxic	31.2 ± 2.8	0.7 ± 0.10	0.25 ± 0.14	0.38 ± 0.26
35	fitc	42	ovarian	nontoxic	breast	nontoxic	33.4 ± 2.22	0.73 ± 0.09	0.23 ± 0.18	0.35 ± 0.24
36	fitc	72	breast	nontoxic	lung	toxic	35.9 ± 3.83	0.78 ± 0.10	0.32 ± 0.17	0.52 ± 0.16
37	fitc	72	breast	nontoxic	lung	nontoxic	31.7 ± 4.2	0.89 ± 0.06	0.35 ± 0.14	0.55 ± 0.1
38	fitc	72	breast	toxic	lung	toxic	31.5 ± 3.5	0.92 ± 0.4	0.41 ± 0.14	0.53 ± 0.13
39	fitc	123	ovarian	nontoxic	breast	toxic	31.4 ± 3.51	0.5 ± 0.19	0.32 ± 0.15	0.5 ± 0.12
40	fitc	42	ovarian	nontoxic	breast	toxic	33.9 ± 2.41	0.68 ± 0.11	0.31 ± 0.16	0.5 ± 0.13
41	fitc	123	ovarian	nontoxic	lung	nontoxic	33.3 ± 2.55	0.77 ± 0.09	0.23 ± 0.21	0.37 ± 0.27
42	fitc	42	ovarian	nontoxic	lung	nontoxic	34.2 ± 2.61	0.8 ± 0.12	0.4 ± 0.22	0.45 ± 0.25
43	fitc	42	ovarian	nontoxic	lung	toxic	36 ± 3.41	0.77 ± 0.11	0.38 ± 0.21	0.55 ± 0.18
44	fitc	123	ovarian	nontoxic	lung	toxic	33.3 ± 4.82	0.61 ± 0.22	0.41 ± 0.19	0.55 ± 0.15
45	fitc	123	ovarian	nontoxic	ovarian	nontoxic	36.6 ± 3	0.85 ± 0.08	0.34 ± 0.27	0.4 ± 0.31
46	fitc	42	ovarian	nontoxic	ovarian	nontoxic	38.2 ± 2.85	0.87 ± 0.09	0.41 ± 0.18	0.45 ± 0.28
47	fitc	72	lung	nontoxic	lung	toxic	37.4 ± 4.36	0.85 ± 0.09	0.49 ± 0.18	0.61 ± 0.14

Figure B.1: **Page 1. Individual results for 243 inference runs for Chapter 2. Evaluating how well does virtual staining generalises?** Each row shows the scores for pixel level, instance level and biological feature level.

ID	stain	seed	train cell type	train phenotype	test cell type	test phenotype	PSNR	SSIM	F1 score	Jaccard Index
62	fitc	42	breast	toxic	breast	toxic	27.6 ± 1.3	0.25 ± 0.12	0.41 ± 0.16	0.55 ± 0.11
63	fitc	123	breast	toxic	lung	toxic	34.3 ± 2.75	0.66 ± 0.15	0.32 ± 0.16	0.55 ± 0.14
64	fitc	42	breast	toxic	lung	toxic	27.6 ± 1.16	0.21 ± 0.14	0.29 ± 0.17	0.51 ± 0.16
65	fitc	123	breast	toxic	ovarian	toxic	34.7 ± 2.28	0.62 ± 0.17	0.27 ± 0.11	0.51 ± 0.1
66	fitc	42	breast	toxic	ovarian	toxic	27.6 ± 1.23	0.17 ± 0.12	0.23 ± 0.12	0.46 ± 0.13
67	fitc	123	lung	toxic	breast	toxic	32.6 ± 2.24	0.59 ± 0.14	0.28 ± 0.15	0.5 ± 0.13
68	fitc	42	lung	toxic	breast	toxic	32.7 ± 2.58	0.59 ± 0.16	0.31 ± 0.17	0.49 ± 0.13
69	fitc	72	ovarian	toxic	ovarian	toxic	35.2 ± 4	0.94 ± 0.05	0.51 ± 0.18	0.6 ± 0.15
70	fitc	123	lung	toxic	lung	toxic	35.5 ± 3.47	0.74 ± 0.15	0.46 ± 0.18	0.59 ± 0.14
71	fitc	42	lung	toxic	lung	toxic	36 ± 3.52	0.73 ± 0.15	0.46 ± 0.18	0.58 ± 0.14
72	fitc	72	ovarian	nontoxic	ovarian	toxic	37.7 ± 3.27	0.85 ± 0.08	0.51 ± 0.17	0.62 ± 0.11
73	fitc	72	ovarian	nontoxic	ovarian	nontoxic	32.8 ± 4.1	0.9 ± 0.08	0.48 ± 0.14	0.6 ± 0.12
74	fitc	123	lung	toxic	ovarian	toxic	35.3 ± 2.86	0.67 ± 0.17	0.36 ± 0.15	0.55 ± 0.11
75	fitc	42	lung	toxic	ovarian	toxic	35.7 ± 3.16	0.66 ± 0.18	0.36 ± 0.14	0.53 ± 0.11
76	fitc	123	ovarian	toxic	breast	toxic	32.8 ± 2.06	0.56 ± 0.13	0.25 ± 0.14	0.46 ± 0.13
77	fitc	42	ovarian	toxic	breast	toxic	32.7 ± 2.13	0.58 ± 0.16	0.32 ± 0.17	0.5 ± 0.14
78	fitc	123	ovarian	toxic	lung	toxic	35.4 ± 3.11	0.71 ± 0.13	0.35 ± 0.2	0.52 ± 0.17
79	fitc	42	ovarian	toxic	lung	toxic	35.2 ± 3.03	0.71 ± 0.13	0.37 ± 0.23	0.53 ± 0.19
80	fitc	123	ovarian	toxic	ovarian	toxic	35.8 ± 2.64	0.66 ± 0.15	0.42 ± 0.15	0.56 ± 0.12
81	fitc	42	ovarian	toxic	ovarian	toxic	35.5 ± 3.18	0.66 ± 0.16	0.47 ± 0.16	0.6 ± 0.12
82	dna	42	breast	nontoxic	breast	nontoxic	37.1 ± 2.1	0.85 ± 0.07	0.81 ± 0.11	0.78 ± 0.1
83	dna	123	breast	nontoxic	breast	nontoxic	37.5 ± 2.21	0.88 ± 0.05	0.84 ± 0.13	0.78 ± 0.1
84	dna	42	breast	nontoxic	breast	toxic	37.4 ± 1.99	0.81 ± 0.11	0.73 ± 0.19	0.72 ± 0.15
85	dna	123	breast	nontoxic	breast	toxic	37.8 ± 2.32	0.89 ± 0.05	0.67 ± 0.19	0.68 ± 0.14
86	dna	42	breast	nontoxic	lung	nontoxic	36 ± 2.04	0.8 ± 0.07	0.7 ± 0.11	0.74 ± 0.06
87	dna	123	breast	nontoxic	lung	nontoxic	36.3 ± 1.99	0.81 ± 0.05	0.71 ± 0.08	0.74 ± 0.06
88	dna	42	breast	nontoxic	lung	toxic	36.5 ± 2.58	0.67 ± 0.22	0.54 ± 0.28	0.6 ± 0.26
89	dna	123	breast	nontoxic	lung	toxic	38.7 ± 3.36	0.85 ± 0.08	0.47 ± 0.26	0.57 ± 0.24
90	dna	123	breast	nontoxic	ovarian	nontoxic	38.9 ± 1.8	0.82 ± 0.09	0.78 ± 0.11	0.73 ± 0.2
91	dna	42	breast	nontoxic	ovarian	toxic	38.6 ± 2.69	0.78 ± 0.19	0.68 ± 0.25	0.67 ± 0.21
92	dna	123	breast	nontoxic	ovarian	toxic	40.3 ± 2.05	0.9 ± 0.07	0.62 ± 0.24	0.64 ± 0.2
93	dna	72	breast	nontoxic	breast	toxic	38.1 ± 2.56	0.86 ± 0.07	0.75 ± 0.19	0.73 ± 0.14
94	dna	72	breast	nontoxic	breast	nontoxic	32.6 ± 3.3	0.96 ± 0.02	0.7 ± 0.22	0.72 ± 0.17
95	dna	72	breast	toxic	breast	toxic	34.2 ± 3.5	0.97 ± 0.02	0.64 ± 0.24	0.67 ± 0.18
96	dna	42	lung	nontoxic	breast	nontoxic	36.4 ± 2.6	0.82 ± 0.08	0.72 ± 0.14	0.72 ± 0.09
97	dna	123	lung	nontoxic	breast	nontoxic	37 ± 2.5	0.81 ± 0.07	0.68 ± 0.11	0.68 ± 0.1
98	dna	42	lung	nontoxic	breast	toxic	37.4 ± 2.61	0.84 ± 0.08	0.6 ± 0.19	0.64 ± 0.14
99	dna	123	lung	nontoxic	breast	toxic	37.2 ± 2.5	0.84 ± 0.08	0.55 ± 0.19	0.63 ± 0.15
100	dna	42	lung	nontoxic	lung	nontoxic	38.4 ± 2.17	0.92 ± 0.03	0.81 ± 0.09	0.78 ± 0.05
101	dna	123	lung	nontoxic	lung	nontoxic	39.1 ± 2.45	0.9 ± 0.05	0.78 ± 0.08	0.7 ± 0.09
102	dna	42	lung	nontoxic	lung	toxic	40.1 ± 2.96	0.86 ± 0.12	0.67 ± 0.29	0.67 ± 0.22
103	dna	123	lung	nontoxic	lung	toxic	40 ± 2.88	0.87 ± 0.11	0.63 ± 0.28	0.65 ± 0.22
104	dna	72	lung	nontoxic	breast	toxic	37.6 ± 2.91	0.84 ± 0.08	0.65 ± 0.19	0.68 ± 0.14
105	dna	72	lung	nontoxic	breast	nontoxic	29.8 ± 3.6	0.75 ± 0.07	0.57 ± 0.25	0.63 ± 0.18
106	dna	72	lung	toxic	breast	toxic	31.3 ± 3.7	0.96 ± 0.03	0.57 ± 0.25	0.62 ± 0.18
107	dna	42	lung	nontoxic	ovarian	nontoxic	39.8 ± 2.29	0.83 ± 0.1	0.85 ± 0.12	0.78 ± 0.09
108	dna	123	lung	nontoxic	ovarian	nontoxic	39.8 ± 2.4	0.83 ± 0.1	0.85 ± 0.12	0.78 ± 0.09

Figure B.2: **Page 2. Individual results for 243 inference runs for Chapter 2. Evaluating how well does virtual staining generalises?** Each row shows the scores for pixel level, instance level and biological feature level.

ID	stain	seed	train cell type	train phenotype	test cell type	test phenotype	PSNR	SSIM	F1 score	Jaccard Index
125	dna	42	ovarian	nontoxic	ovarian	toxic	40.8 ± 2.31	0.88 ± 0.1	0.72 ± 0.24	0.7 ± 0.2
126	dna	123	ovarian	nontoxic	ovarian	toxic	40.9 ± 2.1	0.9 ± 0.06	0.72 ± 0.24	0.68 ± 0.22
127	dna	72	lung	nontoxic	lung	toxic	41.3 ± 3.48	0.93 ± 0.04	0.67 ± 0.29	0.67 ± 0.23
128	dna	72	lung	nontoxic	lung	nontoxic	33.8 ± 4.9	0.76 ± 0.09	0.73 ± 0.25	0.71 ± 0.19
129	dna	72	lung	toxic	lung	toxic	35.3 ± 3.7	0.98 ± 0.1	0.6 ± 0.27	0.64 ± 0.19
130	dna	72	ovarian	toxic	lung	toxic	38.2 ± 3.1	0.98 ± 0.01	0.62 ± 0.03	0.68 ± 0.17
131	dna	72	ovarian	nontoxic	lung	toxic	40.2 ± 3.25	0.89 ± 0.07	0.61 ± 0.27	0.64 ± 0.22
132	dna	72	ovarian	nontoxic	lung	nontoxic	37.2 ± 4	0.98 ± 0.01	0.72 ± 0.25	0.72 ± 0.18
133	dna	123	breast	toxic	breast	toxic	37.6 ± 2.05	0.81 ± 0.09	0.72 ± 0.19	0.71 ± 0.14
134	dna	42	breast	toxic	breast	toxic	38.6 ± 2.74	0.92 ± 0.04	0.73 ± 0.19	0.71 ± 0.14
135	dna	72	breast	nontoxic	ovarian	toxic	40.7 ± 2.19	0.89 ± 0.09	0.68 ± 0.25	0.69 ± 0.2
136	dna	72	breast	nontoxic	ovarian	nontoxic	30.1 ± 4.5	0.94 ± 0.06	0.59 ± 0.23	0.68 ± 0.15
137	dna	72	breast	toxic	ovarian	toxic	34.7 ± 5.1	0.97 ± 0.02	0.5 ± 0.27	0.59 ± 0.23
138	dna	123	breast	toxic	lung	toxic	38.3 ± 2.79	0.77 ± 0.14	0.54 ± 0.27	0.61 ± 0.23
139	dna	42	breast	toxic	lung	toxic	39.6 ± 3.51	0.89 ± 0.07	0.54 ± 0.27	0.61 ± 0.24
140	dna	123	breast	toxic	ovarian	toxic	40.5 ± 2.2	0.88 ± 0.1	0.68 ± 0.24	0.68 ± 0.19
141	dna	42	breast	toxic	ovarian	toxic	41 ± 2.58	0.91 ± 0.07	0.68 ± 0.25	0.67 ± 0.2
142	dna	123	lung	toxic	breast	toxic	37.5 ± 2.54	0.85 ± 0.08	0.63 ± 0.2	0.66 ± 0.15
143	dna	42	lung	toxic	breast	toxic	37.6 ± 2.81	0.87 ± 0.07	0.62 ± 0.19	0.65 ± 0.14
144	dna	72	lung	nontoxic	ovarian	toxic	40.9 ± 2.84	0.82 ± 0.12	0.75 ± 0.24	0.71 ± 0.19
145	dna	72	lung	nontoxic	ovarian	nontoxic	33.8 ± 4.9	0.77 ± 0.09	0.75 ± 0.18	0.75 ± 0.11
146	dna	72	lung	toxic	ovarian	toxic	31.7 ± 4.8	0.97 ± 0.02	0.69 ± 0.27	0.67 ± 0.21
147	dna	123	lung	toxic	lung	toxic	39.5 ± 2.75	0.83 ± 0.16	0.66 ± 0.29	0.66 ± 0.22
148	dna	42	lung	toxic	lung	toxic	39.9 ± 3.15	0.87 ± 0.16	0.67 ± 0.28	0.67 ± 0.22
149	dna	123	lung	toxic	ovarian	toxic	41.1 ± 2.51	0.9 ± 0.11	0.74 ± 0.24	0.7 ± 0.18
150	dna	42	lung	toxic	ovarian	toxic	41 ± 2.6	0.89 ± 0.1	0.74 ± 0.24	0.69 ± 0.18
151	dna	123	ovarian	toxic	breast	toxic	37.1 ± 2.4	0.85 ± 0.06	0.51 ± 0.18	0.56 ± 0.15
152	dna	42	ovarian	toxic	breast	toxic	34.7 ± 1.91	0.61 ± 0.16	0.58 ± 0.19	0.64 ± 0.15
153	dna	123	ovarian	toxic	lung	toxic	39.1 ± 3.54	0.86 ± 0.09	0.5 ± 0.24	0.55 ± 0.21
154	dna	42	ovarian	toxic	lung	toxic	33.9 ± 2.52	0.47 ± 0.2	0.55 ± 0.27	0.62 ± 0.23
155	dna	72	ovarian	toxic	ovarian	toxic	38.2 ± 3.7	0.98 ± 0.02	0.61 ± 0.28	0.66 ± 0.21
156	dna	72	ovarian	nontoxic	ovarian	toxic	42.1 ± 2.35	0.94 ± 0.05	0.76 ± 0.24	0.73 ± 0.19
157	dna	72	ovarian	nontoxic	ovarian	nontoxic	34.3 ± 4.2	0.96 ± 0.04	0.71 ± 0.21	0.73 ± 0.14
158	dna	123	ovarian	toxic	ovarian	toxic	40.9 ± 2.4	0.92 ± 0.07	0.68 ± 0.23	0.64 ± 0.19
159	dna	42	ovarian	toxic	ovarian	toxic	34.5 ± 3.32	0.57 ± 0.2	0.72 ± 0.25	0.71 ± 0.21
160	dna	42	breast	nontoxic	ovarian	nontoxic	39.6 ± 1.9	0.9 ± 0.06	0.81 ± 0.13	0.77 ± 0.1
161	dna	42	ovarian	nontoxic	ovarian	nontoxic	38.1 ± 2.01	0.95 ± 0.03	0.79 ± 0.14	0.78 ± 0.11
162	dna	123	ovarian	nontoxic	ovarian	nontoxic	41.1 ± 2.08	0.94 ± 0.04	0.87 ± 0.12	0.8 ± 0.1
163	cy5	42	breast	nontoxic	breast	nontoxic	35.4 ± 5.26	0.65 ± 0.22	n/a	n/a
164	cy5	123	breast	nontoxic	breast	nontoxic	31.2 ± 5.1	0.61 ± 0.18	n/a	n/a
165	cy5	72	breast	nontoxic	breast	toxic	39.2 ± 4.46	0.78 ± 0.19	n/a	n/a
166	cy5	123	breast	nontoxic	breast	toxic	28.8 ± 2.87	0.37 ± 0.24	n/a	n/a
167	cy5	42	breast	nontoxic	breast	toxic	38.3 ± 4.27	0.81 ± 0.17	n/a	n/a
168	cy5	42	breast	nontoxic	lung	nontoxic	37.1 ± 6.97	0.6 ± 0.31	n/a	n/a
169	cy5	123	breast	nontoxic	lung	nontoxic	25.2 ± 5.87	0.5 ± 0.21	n/a	n/a
170	cy5	42	breast	nontoxic	lung	toxic	36.8 ± 5.65	0.66 ± 0.24	n/a	n/a
171	cy5	123	breast	nontoxic	lung	toxic	29.7 ± 3.31	0.41 ± 0.3	n/a	n/a

Figure B.3: **Page 3. Individual results for 243 inference runs for Chapter 2. Evaluating how well does virtual staining generalises?** Each row shows the scores for pixel level, instance level and biological feature level.

ID	stain	seed	train cell type	train phenotype	test cell type	test phenotype	PSNR	SSIM	F1 score	Jaccard Index	Average NMAE for Top 20 features
188	cy5	123	lung	nontoxic	ovarian	nontoxic	34 ± 6.46	0.58 ± 0.3	n/a	n/a	1.52
189	cy5	42	lung	nontoxic	ovarian	toxic	35.1 ± 6.21	0.59 ± 0.3	n/a	n/a	0.92
190	cy5	123	lung	nontoxic	ovarian	toxic	35.7 ± 6.12	0.61 ± 0.29	n/a	n/a	0.92
191	cy5	72	breast	nontoxic	lung	toxic	38.3 ± 6.73	0.64 ± 0.31	n/a	n/a	0.63
192	cy5	123	ovarian	nontoxic	breast	nontoxic	34.2 ± 4.35	0.62 ± 0.25	n/a	n/a	1.27
193	cy5	42	ovarian	nontoxic	breast	nontoxic	31.6 ± 4.2	0.59 ± 0.21	n/a	n/a	1.27
194	cy5	123	breast	toxic	lung	toxic	37 ± 5.92	0.66 ± 0.25	n/a	n/a	0.64
195	cy5	123	ovarian	nontoxic	breast	toxic	36.9 ± 5	0.71 ± 0.23	n/a	n/a	1.28
196	cy5	42	ovarian	nontoxic	breast	toxic	35.3 ± 5.08	0.66 ± 0.25	n/a	n/a	1.32
197	cy5	42	ovarian	nontoxic	lung	nontoxic	34 ± 4.9	0.63 ± 0.25	n/a	n/a	1.57
198	cy5	123	ovarian	nontoxic	lung	nontoxic	34.8 ± 5.68	0.63 ± 0.2	n/a	n/a	1.57
199	cy5	42	ovarian	nontoxic	lung	toxic	36.1 ± 6.47	0.64 ± 0.28	n/a	n/a	0.63
200	cy5	123	ovarian	nontoxic	lung	toxic	37.4 ± 6.92	0.66 ± 0.27	n/a	n/a	0.62
201	cy5	42	ovarian	nontoxic	ovarian	nontoxic	36.4 ± 6.69	0.65 ± 0.26	n/a	n/a	1.44
202	cy5	123	ovarian	nontoxic	ovarian	nontoxic	36.4 ± 5.4	0.66 ± 0.2	n/a	n/a	1.42
203	cy5	72	lung	nontoxic	lung	toxic	36.2 ± 7.25	0.65 ± 0.3	n/a	n/a	0.58
204	cy5	42	ovarian	nontoxic	ovarian	toxic	38.4 ± 6.67	0.7 ± 0.26	n/a	n/a	1.01
205	cy5	123	ovarian	nontoxic	ovarian	toxic	39.1 ± 7.04	0.71 ± 0.26	n/a	n/a	0.99
206	cy5	42	breast	toxic	lung	toxic	28.3 ± 3.06	0.14 ± 0.16	n/a	n/a	0.65
207	cy5	72	breast	toxic	ovarian	toxic	34.2 ± 6.6	0.72 ± 0.14	n/a	n/a	6.54
208	cy5	72	ovarian	nontoxic	lung	toxic	37.3 ± 6.64	0.65 ± 0.26	n/a	n/a	0.60
209	cy5	72	breast	nontoxic	ovarian	toxic	39.6 ± 6.66	0.66 ± 0.3	n/a	n/a	0.98
210	cy5	123	breast	toxic	ovarian	toxic	38.4 ± 5.96	0.7 ± 0.24	n/a	n/a	1.02
211	cy5	42	breast	toxic	ovarian	toxic	28 ± 2.66	0.12 ± 0.12	n/a	n/a	1.03
212	cy5	72	lung	toxic	breast	toxic	34.2 ± 5.1	0.96 ± 0.03	n/a	n/a	0.20
213	cy5	72	lung	toxic	lung	toxic	39.3 ± 5.4	0.98 ± 0.1	n/a	n/a	5.50
214	cy5	123	lung	toxic	breast	toxic	36.1 ± 5.59	0.66 ± 0.26	n/a	n/a	1.19
215	cy5	42	lung	toxic	breast	toxic	28.5 ± 2.23	0.29 ± 0.22	n/a	n/a	1.15
216	cy5	123	lung	toxic	lung	toxic	36.1 ± 7.12	0.65 ± 0.29	n/a	n/a	0.57
217	cy5	72	lung	nontoxic	ovarian	toxic	35.1 ± 6.89	0.57 ± 0.31	n/a	n/a	0.94
218	cy5	42	lung	toxic	lung	toxic	28.5 ± 2.41	0.29 ± 0.25	n/a	n/a	0.56
219	cy5	72	lung	toxic	ovarian	toxic	39.2 ± 5.6	0.97 ± 0.02	n/a	n/a	6.64
220	cy5	123	lung	toxic	ovarian	toxic	34.1 ± 6.66	0.53 ± 0.32	n/a	n/a	0.92
221	cy5	42	lung	toxic	ovarian	toxic	28.4 ± 2.51	0.26 ± 0.26	n/a	n/a	0.91
222	cy5	72	ovarian	toxic	lung	toxic	34.1 ± 6.3	0.68 ± 0.13	n/a	n/a	5.54
223	cy5	123	ovarian	toxic	lung	toxic	32.4 ± 6.18	0.49 ± 0.32	n/a	n/a	0.64
224	cy5	42	ovarian	toxic	lung	toxic	34.9 ± 6.26	0.6 ± 0.29	n/a	n/a	0.62
225	cy5	42	ovarian	toxic	ovarian	toxic	36.7 ± 6.66	0.65 ± 0.28	n/a	n/a	1.01
226	cy5	123	ovarian	toxic	ovarian	toxic	33.9 ± 6.9	0.53 ± 0.32	n/a	n/a	1.03
227	cy5	72	ovarian	toxic	ovarian	toxic	34.1 ± 6.4	0.72 ± 0.15	n/a	n/a	6.65
228	cy5	72	ovarian	nontoxic	ovarian	toxic	38.5 ± 6.89	0.68 ± 0.27	n/a	n/a	0.94
229	cy5	72	breast	nontoxic	breast	nontoxic	27.5 ± 6	0.57 ± 0.1	n/a	n/a	1.21
230	cy5	72	lung	nontoxic	breast	nontoxic	33 ± 5.1	0.95 ± 0.03	n/a	n/a	1.27
231	cy5	72	ovarian	nontoxic	breast	nontoxic	26.4 ± 6.1	0.52 ± 0.1	n/a	n/a	1.28
232	cy5	72	breast	nontoxic	lung	nontoxic	31.9 ± 7.3	0.67 ± 0.15	n/a	n/a	6.80
233	cy5	72	lung	nontoxic	lung	nontoxic	39.5 ± 5.3	0.98 ± 0.02	n/a	n/a	6.14
234	cy5	72	ovarian	nontoxic	lung	nontoxic	33.8 ± 6.9	0.7 ± 0.14	n/a	n/a	6.66
235	cy5	72	breast	nontoxic	ovarian	nontoxic	32.3 ± 7.6	0.67 ± 0.15	n/a	n/a	7.34
236	cy5	72	lung	nontoxic	ovarian	nontoxic	39.2 ± 6.2	0.98 ± 0.03	n/a	n/a	7.40
237	cy5	72	ovarian	nontoxic	ovarian	nontoxic	33.9 ± 7	0.68 ± 0.14	n/a	n/a	7.39
238	cy5	42	breast	toxic	breast	toxic	28.3 ± 2.59	0.1 ± 0.06	n/a	n/a	1.21
239	cy5	123	breast	toxic	breast	toxic	38.5 ± 4.6	0.81 ± 0.18	n/a	n/a	1.23
240	cy5	42	ovarian	toxic	breast	toxic	36.3 ± 4.87	0.69 ± 0.23	n/a	n/a	1.28
241	cy5	123	ovarian	toxic	breast	toxic	31.8 ± 5	0.49 ± 0.28	n/a	n/a	1.36
242	cy5	72	ovarian	toxic	breast	toxic	34.2 ± 6.6	0.72 ± 0.14	n/a	n/a	0.41
243	cy5	72	breast	toxic	breast	toxic	29.2 ± 6.1	0.62 ± 0.1	n/a	n/a	0.12

Figure B.4: **Page 4. Individual results for 243 inference runs for Chapter 2. Evaluating how well does virtual staining generalises?** Each row shows the scores for pixel level, instance level and biological feature level.

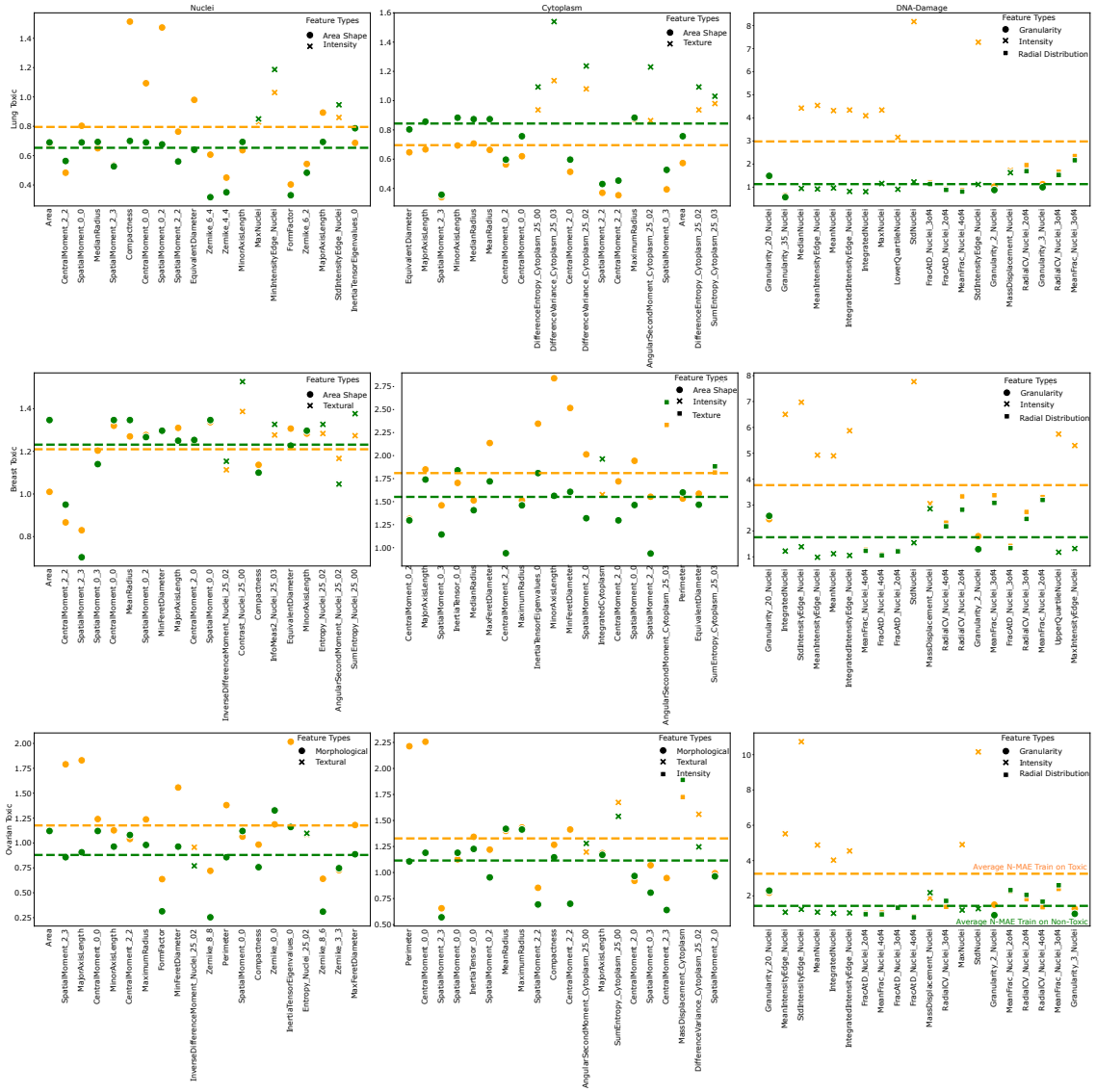


Figure B.5: **N-MAE values of the virtual stain models trained on non-toxic and toxic samples for lung, breast and ovarian samples. N-MAE is obtained for the 20 most informative biological features identified on the fluorescence stain.** A N-MAE of 0.1 is equivalent to 10% of the standard deviation identified in the fluorescence stain. The green line represents the average N-MAE for the virtual stain models trained on non-toxic samples, and the yellow line represents the average N-MAE for the virtual stain models trained on toxic samples. Across a diverse set of feature types training on non-toxic leads to a biological feature representation that more closely aligns with that found in the fluorescence stains compared to training on toxic.

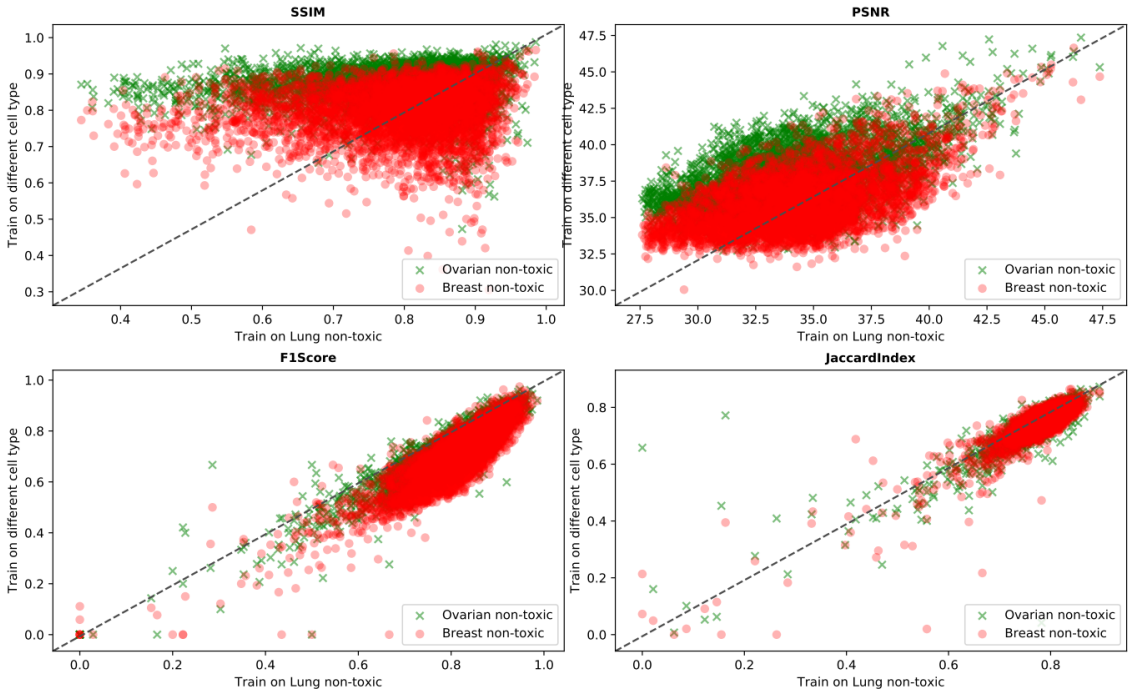


Figure B.6: **Scatterplots of SSIM, PSNR, F1 Score and Jaccard Index for generalisation to lung of the virtual ovarian and breast cytoplasm models**. Along the x-axis, we have the values obtained from the model trained on lung (baseline) and on the y-axis the corresponding values obtained from ovarian (green) breast (red) and lung (shown as diagonal).

Appendix C

Appendix C: Additional data for Chapter 6

Feature	Method	*Population Measurements*		*Proposed Metrics*		
		WassersteinDistance Population Values	**KL Divergence Population Values fitted to GMM**	**Average Log-likelihoods**	**Wasserstein Distance Standard Scores &~N (0,1)**	**Wasserstein Distance Percentile Ranks &~U(1,100)**
LowerQuartileIntensity	cDDPM	**0.02**	0.12	**0.17**	**0.81**	**44.1**
	Pix2PixHD	0.03	**0.11**	-8.30	0.82	47.3
MeanIntensityEdge	cDDPM	**0.01**	**0.04**	**0.23**	0.86	**43.0**
	Pix2PixHD	0.04	0.17	-12.74	**0.65**	48.7
MeanIntensity	cDDPM	**0.02**	**0.09**	**0.14**	**0.81**	**43.1**
	Pix2PixHD	0.03	**0.09**	-10.71	0.83	44.9
MedianIntensity	cDDPM	**0.03**	0.11	**-0.37**	**0.79**	**42.2**
	Pix2PixHD	**0.03**	**0.09**	-10.96	0.86	43.6
StdIntensityEdge	cDDPM	0.02	**0.14**	**1.45**	**1.38**	16.6
	Pix2PixHD	**0.01**	0.15	-2.09	2.06	**14.7**
StdIntensity	cDDPM	**0.01**	0.11	**1.57**	**0.82**	35.8
	Pix2PixHD	**0.01**	**0.09**	-2.13	1.23	**29.7**
UpperQuartileIntensity	cDDPM	**0.03**	0.09	**-0.16**	**0.76**	**41.4**
	Pix2PixHD	**0.03**	**0.08**	-10.72	0.84	42.0
RadialDistribution_FracAtD_1of4	cDDPM	0.004	0.071	**2.85**	0.74	69.0
	Pix2PixHD	**0.003**	**0.051**	1.65	**0.62**	**64.0**
RadialDistribution_FracAtD_2of4	cDDPM	0.007	0.038	**2.25**	0.55	66.8
	Pix2PixHD	**0.002**	**0.012**	1.51	**0.17**	**55.6**
RadialDistribution_FracAtD_3of4	cDDPM	0.008	**0.067**	**2.51**	**0.60**	**31.0**
	Pix2PixHD	**0.005**	**0.067**	1.96	0.63	**35.4**
RadialDistribution_FracAtD_4of4	cDDPM	0.014	0.052	**1.97**	**0.31**	40.7
	Pix2PixHD	**0.003**	**0.014**	1.03	0.17	48.3
RadialDistribution_MeanFrac_1of4	cDDPM	0.032	0.032	**0.51**	0.37	59.4
	Pix2PixHD	**0.023**	**0.025**	-0.93	**0.31**	**54.6**
RadialDistribution_MeanFrac_2of4	cDDPM	**0.038**	**0.084**	**1.05**	**0.12**	48.9
	Pix2PixHD	0.039	0.119	-0.39	0.62	**37.6**
RadialDistribution_MeanFrac_3of4	cDDPM	0.038	**0.272**	**1.51**	**1.09**	**19.9**
	Pix2PixHD	**0.031**	0.362	-0.40	1.15	24.5
RadialDistribution_MeanFrac_4of4	cDDPM	0.039	**0.080**	**1.45**	**0.26**	**57.2**
	Pix2PixHD	**0.034**	0.270	-0.08	0.64	65.4
RadialDistribution_RadialCV_1of4	cDDPM	**0.019**	**0.080**	**1.01**	**0.52**	62.0
	Pix2PixHD	0.023	0.130	0.002	0.55	**43.0**
RadialDistribution_RadialCV_2of4	cDDPM	0.020	0.071	**1.53**	**0.19**	51.3
	Pix2PixHD	**0.008**	**0.020**	0.71	0.38	**44.5**
RadialDistribution_RadialCV_3of4	cDDPM	0.030	0.085	**1.62**	**0.27**	48.6
	Pix2PixHD	**0.006**	**0.019**	0.71	0.33	**44.7**

Table C.1: **Traditional population measurements vs. the newly proposed metrics for measuring 18 biological features obtained via virtual staining samples from cDDPM and Pix2pixHD.** Traditional distance measurements are the Wasserstein distance across the entire sample population and the KL-divergence of the same sample population fitted to a Gaussian mixture model. The newly proposed measurements computed for each instance-wise biological feature posterior distribution (approximated with 1,000 virtual staining samples) with respect to the single known ground truth sample; the log-likelihood, the standard score and the percentile rank. Features such as "UpperQuantileIntensity, "RadialDistribution_Radial_MeanFrac_3of4" and "RadialDistribution_Radial_MeanFrac_4of4" show that our proposed metrics reveal conflicting outcomes compared to the population metrics due to observations at the instance level that can not be observed at the population level. For features such as "MeanIntensity" quantifying performance at the instance-level compared to the population level leads to the same overall outcome but the scale of the outcome is far clearer. The percentile rank is shown to conflict with the log-likelihood and standard score results for a number of features.

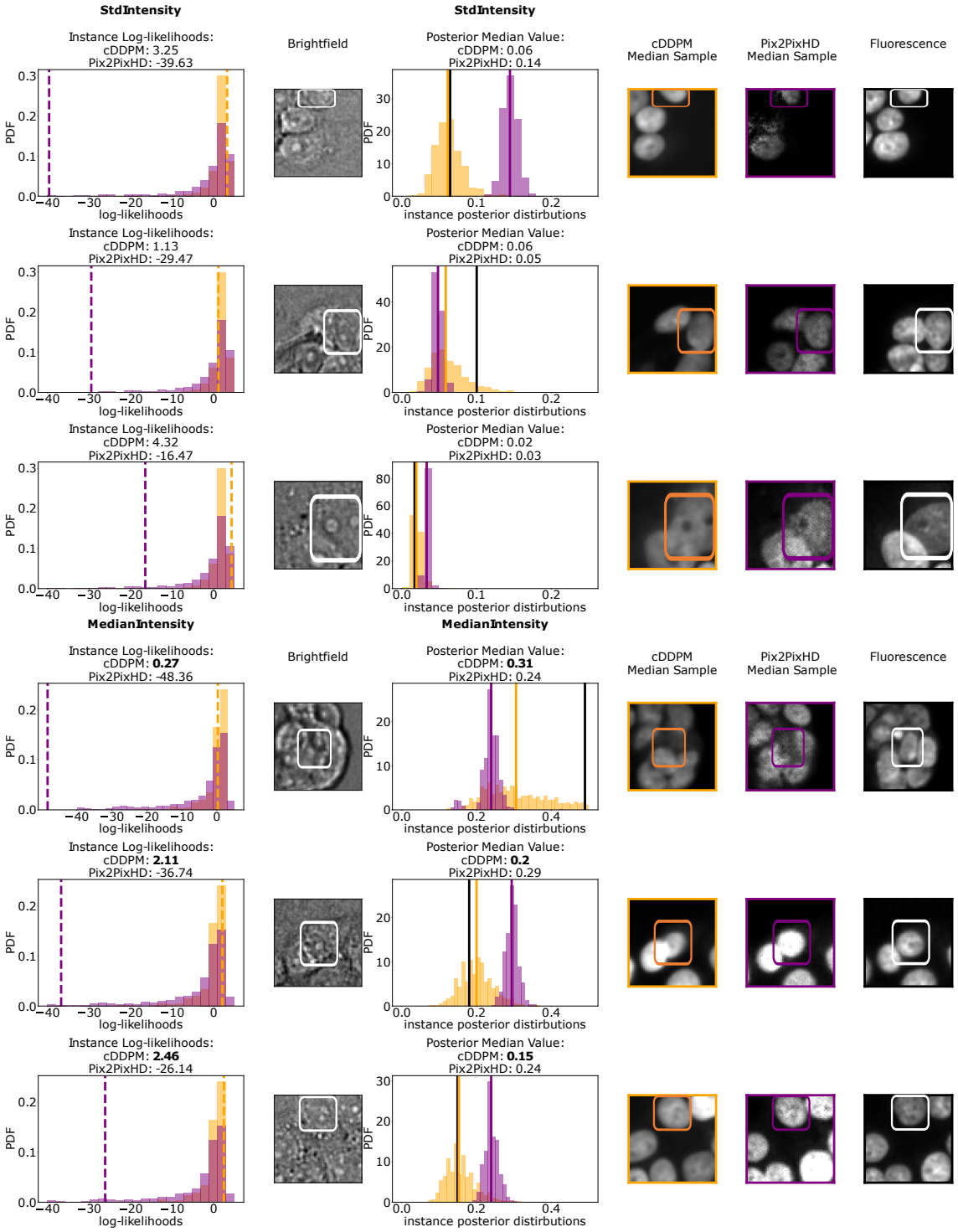


Figure C.1: See below caption

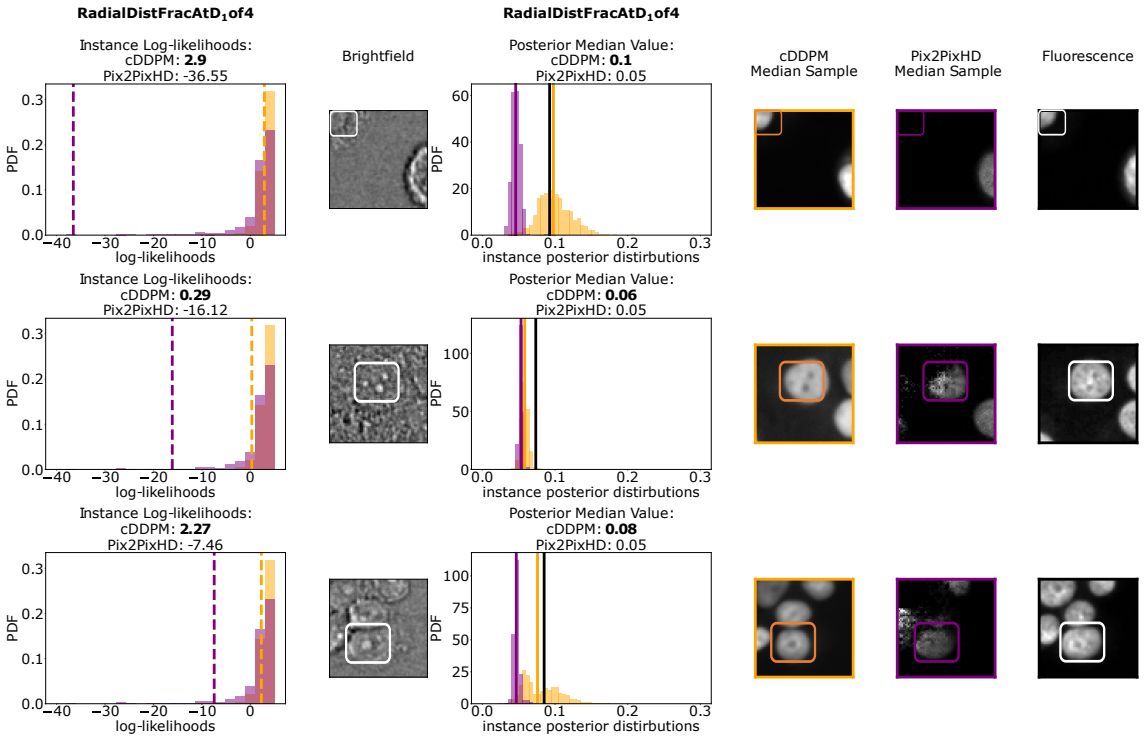


Figure C.2: **Individual instance feature performance vs qualitative results**. For each row for both cDDPM and pix2pixHD, we highlight the instance result within the distribution of log-likelihoods, the bright-field input and corresponding instance, the predicted instance posterior distributions. We also highlight the median values of both methods versus the target value and qualitatively show the median virtual staining instance sample versus the target fluorescence instance. Each feature contains three rows each with improving instance log-likelihood results for pix2pixHD. The predicted posterior distributions of cDDPM are much wider than pix2pixHD and are always much closer to the target value. We can observe for both feature types, as the log-likelihood improves, the pix2pixHD predicted posterior distribution moves closer to the target value. Qualitatively, we can see for both features pix2pixhd either over and under saturates the instance intensity compared to cDDPM.