DTU Compute
Department of Applied Mathematics and Computer Science

Generative modelling for data augmentation

Generative modelling for data augmentation of in vitro serous fluid ultrasound images to accelerate the validation of a non-invasive white blood cells counting medical device

Master Thesis

Generative modelling for data augmentation

Generative modelling for data augmentation of in-vitro serous fluid ultrasound images to accelerate the validation of a non-invasive white blood cells counting medical device

Master Thesis
July, 2023

By
Miguel Alejandro Rodríguez Álvarez

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science, Richard Petersens Plads, 324, 2800 Kgs. Lyngby, Denmark Denmark
www.compute.dtu.dk
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

This Master thesis was prepared at the Department of Applied Mathematics and Computer Science of the Technical University of Denmark (DTU) and in collaboration with Newborn Solutions in fulfillment of the requirements for acquiring a degree in Master of Science in Business Analytics, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of machine learning and deep learning.

Miguel Alejandro Rodríguez Álvarez - s212780

.....

Signature

.....

Date

Abstract

The limited amount of data in the medical domain and the necessity of training samples for increased performance of deep learning models is a recurrent challenge. Newborn Solutions seeks to accelerate the validation of their non-invasive white blood cells counting medical device Neosonics with synthetic in vitro images acquisition, being their next move the digitization of the image generation process.

This work presents the design, implementation and evaluation of a continuous scalar conditional Generative Adversarial Network to perform a GAN-based data augmentation for an image regression model on in vitro meningitis and in vitro peritoneal dialysis ultrasound images. The proposed GAN architecture and training scheme is capable of dealing with the images specific resolution and continuous scalar conditioning. General GAN design guidelines were followed, but novel design choices were also introduced: 1) using a combination of varying kernel sizes along the transposed convolutional layers in the generator to achieve the desired image resolution and 2) projecting both the input noise and condition values to a disentangled intermediate latent space.

The experiments and evaluation showed that the continuous scalar conditional Generative Adversarial Network is capable of generating good visual quality and diverse images containing features with high resemblance to real ones. However, the image regression task proved to be challenging and the image regressor performance could not be improved consistently by means of a GAN-based data augmentation approach.

Acknowledgements

Miguel Alejandro Rodríguez Álvarez, MSc Business Analytics, DTU
Author.

Aasa Feragen, Professor, [DTU Compute]
Academic supervisor

Beatrice Jobst, Senior Data Scientist, Newborn Solutions
Company's supervisor

Contents

Preface ii
Abstract iii
Acknowledgements iv

1 Introduction **1**
1.1 Newborn Solutions 1
1.2 Objective 2

2 Related work **3**

3 Theoretical background **5**
3.1 Generative Modelling 5
3.2 Generative Adversarial Networks 5

4 Methodology **9**
4.1 Dataset 9
4.2 GAN models 14
4.3 Regression model 26
4.4 Evaluation 28
4.5 Results 29

5 Discussion **55**

6 Future steps **57**

1 Introduction

Medical image aided diagnosis and treatment based on machine learning, specially deep learning models, has developed rapidly and is expected to continue growing given the promising results in terms of improved diagnosis accuracy and clinical workflow efficiency. Despite the interest, it remains challenging to develop medical imaging models, because high-quality labeled data is often scarce due to imaging cost, the time-consuming effort needed to annotate medical images and patient privacy concerns.

Data augmentation is a technique commonly used to increase the sample size of the training set by applying affine transformations such as flipping, rotation and scaling. To a certain extent, it alleviates the problem of limited sample size of medical images, but the diversity of the dataset is only slightly improved since the image content of the new samples is too similar to the original images.

In response, recent studies are exploring the application of generative modelling for machine learning based data augmentation. Generative models are capable of learning from available images and generate new samples that combine features. ”Among them, Generative Adversarial Network (GAN) has attracted the attention of researchers due to its remarkable image generation performance and has been widely used in the field of medical image augmentation” (Chen et al., 2022).

1.1 Newborn Solutions

Newborn Solutions is a medical device startup located in Barcelona that will enable non-invasive screening, diagnosis and monitoring of infections in serous body fluids, with an initial focus on infant meningitis. The company has developed the Neosonics, an ultrasound-based device coupling materials and artificial intelligence algorithms, to capture ultrasound images from the cerebrospinal fluid (CSF) directly below the infant fontanelle and make an image-based count of white blood cells (WBC) present in the fluid (a well-established diagnostic marker for meningitis infections), enabling a non-invasive prediction of meningitis infections.

Meningitis is an inflammation of the tissues that cover the brain and spinal cord, sometimes affecting the brain itself. In case of symptoms, it is necessary to conduct a blood test to check for a bacterial infection and obtain some spinal fluid by performing a lumbar puncture (LP). This procedure involves inserting a special needle into the child’s lower back to draw out spinal fluid from the bottom of the sac surrounding the spinal cord (American Academy of Pediatrics et al., 2019). Lumbar puncture is generally a safe procedure, but it must be performed by a doctor or specialist nurse, it takes approximately 30 minutes and the collected samples need to be sent to laboratory for testing and white blood cells counting, with the results available after 2 hours. A count of more than 30 cells$/mm^3$ is enough to consider the child is positive. Meningitis progresses quickly, so it must be detected early and treated aggressively. The Neosonics would offer an alternative to lumbar puncture, not requiring a specialist and with counting results available immediately. With early diagnosis and proper treatment, the chances of a good recovery increase significantly.

Future applications of the technology include uveitis, chorioamnionitis, septic arthritis and peritonitis. Peritonitis, an infection of the peritoneum with a 5-15% mortality rate, occurs in 25% of patients on home peritoneal dialysis. Currently, when a patient presents symptoms of infection or has observed turbidity in the drainage bag, a sample of effluent

fluid is obtained for white blood cell count and culture. If the count exceeds 100 cells$/mm^3$, antibiotic treatment is initiated pending culture results. However, the slow onset of these symptoms and the unreliability of the subjective assessment of the effluent appearance can lead to delays and decreases the efficacy of antibiotic treatment. In this context, the Neosonics would enable an image-based count of white blood cells from images acquired directly from the dialysis bag, alerting in case of an elevated count so that antibiotic treatment could be administered as soon as possible.

The Neosonics incorporates several deep learning models to process the acquired images and enable the image-based white blood cells count. The images used to train the predictive models for the meningitis application come from several hospitals in Spain and the African countries of Mozambique and Morocco (in vivo images). However, these sources are not providing enough diverse data and they have recently started creating new synthetic images in their laboratory (in vitro images) for both the meningitis and peritoneal dialysis applications. While valuable, physical modelling is tedious, time-consuming and challenging. For that reason, their next move is to further extend data augmentation via digitization of the image generation process.

1.2 Objective

The Master Thesis project aims to apply generative modelling for data augmentation of in vitro serous fluid ultrasound images to accelerate the validation of the non-invasive white blood cells counting medical device of Newborn Solutions. The specific contribution of this work is the design, implementation and evaluation of a continuous scalar conditional Generative Adversarial Network to perform a GAN-based data augmentation for an image regression model on in vitro meningitis and in vitro peritoneal dialysis ultrasound images.

Currently, the white blood cells counting is implemented as an image regression model to predict concentration levels from serous fluids ultrasound images. Training with a larger number of images from a more diverse set of concentrations is expected to improve the generalization capacity and the overall performance of the regression model. Therefore, the goal of the conditional GAN model is to generate images from a wide range of concentrations, even new ones not available in the original training data.

Initially, the generative and regression models will be trained and validated on in vitro meningitis images and, as a next step, on in vitro peritoneal dialysis data. If the GAN-based data augmentation is successful on the in vitro datasets, it could be extended to in vivo data where the on-demand acquisition of new and varied samples is not possible, thus accelerating the clinical trial recruitment and validation of the device.

2 Related work

The limited amount of data in the medical domain and the necessity of training samples for increased performance of deep learning models has resulted in extensive research on the application of different deep generative models for data augmentation in the medical domain.

Variational Autoencoders (VAEs) were one of the first deep generative models, but their application in data augmentation is limited. VAEs present advantages such as learning stability, low computational cost and output diversity. Pesteie et al. (2019) demonstrate the effectiveness of the approach on two independent clinical datasets consisting of ultrasound images of the spine and magnetic resonance images of the brain. However, the problem with VAEs is their tendency to often produce blurry and hazy output images (Chen et al., 2022), specially in high resolutions, with hierarchical VAE's trying to overcome this issue (Vahdat and Kautz, 2021).

Recently, a new type of deep generative models called Diffusion Models (DMs) has emerged and promises remarkable results with a great ability to generate realistic and diverse outputs (Kebaili et al., 2023). Although DMs are still in their infancy and are not yet well established in the medical field, Müller-Franzes et al. (2022) affirm in their comparison study that Deep Diffusion Models (DDMs) are a superior alternative to GANs for image synthesis in the medical domain. DMs are a promising alternative to previous generative models, but they present the drawbacks of high computational cost, slow generation and necessity of a large amount of data for effective training. This last requirement is sometimes difficult to meet when dealing with medical images and could discourage their application.

Generative Adversarial Networks are still the most widely used deep generative models, with contrasted ability to generate realistic images. There are numerous studies on the application of GAN-based models for data augmentation of medical images. In their study, Frid-Adar et al. (2018) state that generated medical images can be used for synthetic data augmentation, improving the performance of Convolutional Neural Networks (CNNs) for medical image classification even on a limited dataset of computed tomography (CT) images of 182 liver lesions. "Nevertheless, GANs also have their drawbacks, such as learning instability, difficulty in converging and suffering from mode collapse" (Kebaili et al., 2023).

3 Theoretical background

3.1 Generative Modelling

Generative modeling aims to ”learn a representation of an intractable probability distribution X defined over $\mathbb{R}^n$, where n typically is large, and the distribution is complicated” (Ruthotto and Haber, 2021), for example, a set of images.

The goal of generative modeling is to obtain a generator $G : \mathbb{R}^q \rightarrow \mathbb{R}^n$ that maps samples from a tractable distribution Z, usually a univariate Gaussian, supported in $\mathbb{R}^q$ referred to as the latent space to points in an intractable probability distribution X defined over $\mathbb{R}^n$ using a finite number of independent and identically distributed samples from X as training data. (Ruthotto and Haber, 2021) It is assumed that for each sample $\mathbf{x} \sim X$ there is at least one point $\mathbf{z} \sim Z$ such that $G(\mathbf{z}) \approx \mathbf{x}$.

The vector **z** that results in a given vector **x** is generally unknown. Additionally, it is important to note that the latent space dimension q will generally be lower than the dimension of the data space n. ”Having a generator that can map points from the simple distribution Z to the intractable distribution X allows to generate samples from the complicated space X, which is desired in many applications” (Ruthotto and Haber, 2021).

In recent years, feed-forward deep neural networks have been adopted to model the generator G for being generic function approximators and their ability to work in high dimensions effectively, resulting in what is known as Deep Generative Modelling (DGM) (Ruthotto and Haber, 2021). Designing an effective architecture is critical to model the generative process and solve the learning problem, but it is still a complicated task due to the lack of theoretical guidelines.

”DGM training is an ill-posed problem since uniquely identifying a probability distribution from a finite number of samples is impossible” (Ruthotto and Haber, 2021). Hence, the performance of the DGM will depend not only the design of the network, but on the choice of training objective, regularization and training algorithms.

3.2 Generative Adversarial Networks

Generative Adversarial Networks are one of the most popular and successful modeling frameworks within deep generative modeling. The name comes from its adversarial training process where the generative model G tries to generate samples from a given data distribution while the discriminator model D tries to differentiate real data samples from those generated by G.

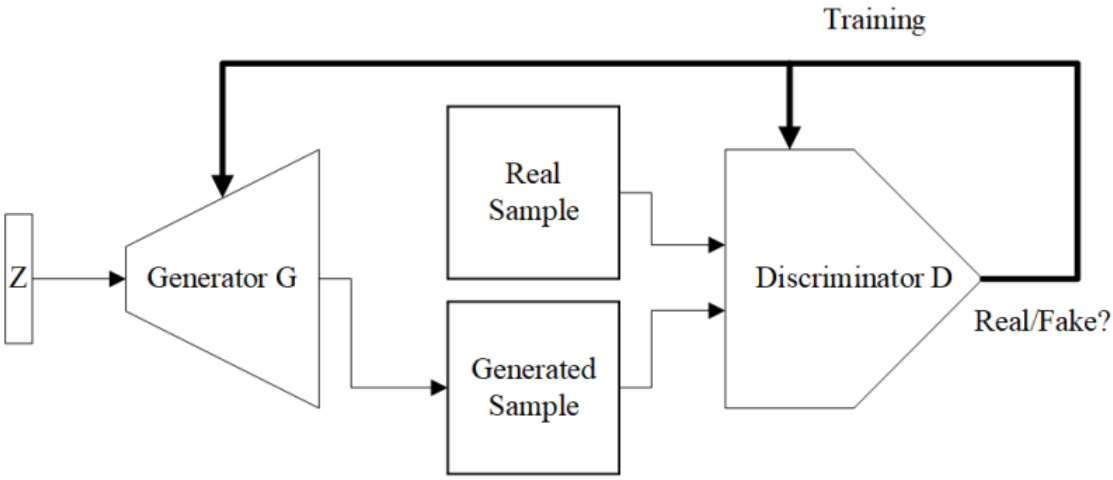

Figure 3.1: Generative Adversarial Network training scheme (Ongun and Temizel, 2018)

As formally described by Goodfellow et al. (2014), the generator's distribution p_g over data **x** is learned by defining a prior on input noise variables $p_z(\mathbf{z})$ and a mapping to data space as $G(\mathbf{z}; \theta_g)$, where G is a differentiable function represented by a feed-forward deep neural network with parameters θ_g. A second feed-forward deep neural network $D(\mathbf{x}; \theta_d)$ is defined, which outputs a single scalar representing the probability that **x** came from the data rather than p_g. The discriminator D is trained to maximize $\log D(\mathbf{x})$, therefore differentiating training data samples and samples from G, which is simultaneously trained to minimize $\log(1 - D(G(\mathbf{z})))$. In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$V(G, D) = \min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(x)}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[1 - \log D(G(\mathbf{z}))] \tag{3.1}$$

In practice, the minimax game is implemented using an iterative and numerical approach. Optimizing D to completion in the inner loop of training is computationally prohibitive and, on finite datasets, would result in overfitting. Instead, the training alternates between k steps of optimizing D and one step of optimizing G. This results in D being maintained near its optimal solution as long as G changes slowly enough.

Another practical consideration is that equation 3.1 might not provide sufficient gradient for proper learning of G. In early steps, D can reject the poorly generated samples from G with high confidence for being too distant from the training data distribution. This leads to the saturation of $log(1 - D(G(\mathbf{z})))$. Instead, training G to maximize $\log D(G(\mathbf{z}))$ results in much stronger gradients in early steps.

3.2.1 GAN evaluation

The major downside of latent variable models like GANs is that there is no certain and objective method of evaluating the performance, seen as the generative capabilities. Qualitative evaluation by visual examination of samples by human is still needed most of the time, although it can be time-consuming, subjective and cannot capture distributional characteristics (Saxena and Cao, 2023).

Several metrics have been proposed for the automatic evaluation of image generative models, trying to demonstrate a preference for models that generate realistic and

varied images and correlation with visual quality, being the most widely used metrics the Inception Score and the Fréchet Inception Distance.

The Inception Score (IS) proved to correlate well with human evaluation and scoring of the realism of generated images from the CIFAR-10 dataset (Salimans et al., 2016). The metric is computed from statistics of the output of the Inception V3 Network pre-trained on ImageNet, thus the name, when applied to generated images.

$$IS(G) = \exp\big(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y))\big) \tag{3.2}$$

where $\mathbf{x} \sim p_g$ indicates that $\mathbf{x}$ is an image sampled from p_g and $D_{KL}(p(y|\mathbf{x}) \parallel p(y))$ is the KL-divergence between the conditional class distribution $p(y|\mathbf{x})$ and the marginal class distribution $p(y) = \int_{\mathbf{x}} p(y|\mathbf{x})p_g(\mathbf{x})$. For evaluation, the distributions are approximated through sampling.

As explained by Barratt and Sharma (2018), the IS aims to codify two desirable qualities of a generative model:

1. The generated images should contain clear elements, being sharp rather than blurry, making the Inception Network highly confident there is a single element in the image and resulting in low entropy $p(y|\mathbf{x})$.
2. The generator should output a high diversity of images corresponding to different classes in ImageNet, resulting in high entropy $p(y)$

When both of these traits are satisfied by the generator, a large KL-divergence between the distributions $p(y)$ and $p(y|\mathbf{x})$ is expected, resulting in a desired high Inception Score of the generator. It must be noted that for a domain with K classes, the score is within the range $[1, K]$.

The Fréchet Inception Distance (FID) is the current standard metric for assessing the quality of generative models. Unlike the earlier Inception Score, which evaluates only the distribution of generated images, the FID compares the distribution of generated images with the distribution of real images (Heusel et al., 2018). For two sets of real images R and generated samples G, the FID computes the Fréchet distance between the real data distribution and generated data distribution using a given feature extractor f, typically the 2048-dimensional activation vector from the *pool3* layer of an Inception V3 Network pre-trained on ImageNet.

$$FID(R, G) = d^2(f(R), f(G)) = \left\|\mu^R - \mu^G\right\|^2 + Tr(\Sigma^R + \Sigma^G - 2(\Sigma^R\Sigma^G)^{\frac{1}{2}}) \tag{3.3}$$

where μ^R, Σ^R and μ^G, Σ^G are the means and covariance matrices of the distributions $f(R)$ and $f(G)$, respectively. For evaluation, the mean vectors and covariance matrices are approximated through sampling.

Since both IS and FID rely on applying the Inception V3 Network pre-trained on ImageNet, generative models trained on much different datasets might foster misleading results (Barratt and Sharma, 2018). However, the feature space mapping used in FID is believed to be more robust and meaningful than the probability-based approach from IS (Xu et al., 2018).

4 Methodology

4.1 Dataset

The image data used in the study comes from ultrasound video recordings captured in the laboratory using a Neosonics (the ultrasound device developed by Newborn Solutions) and a specific acquisition setup depending on the two different applications considered: meningitis and peritoneal dialysis.

4.1.1 Acquisition

In vitro meningitis

For the in vitro meningitis acquisition, fontanel-mimicking phantoms and fake hair were used to replicate the extraction conditions of infant meningitis patients. In this setup, the cerebrospinal fluid (CSF) is replaced by water and white blood cells by polystyrene particles (pp) with 5µm diameter. The in vitro meningitis dataset is composed of images recorded from two experiments under different conditions in October and December 2022. The experiment in October recorded images from two separate setups: one with a phantom of 4.4dB attenuation with hair and a phantom of 9.8dB attenuation without hair, in both the Neosonics was configured with 42dB gain and a voltage of 100V. The recorded concentrations (pp/µL) were 0, 3, 5, 7, 9, 15, 20, 30, 50 and 100. However, there was a problem in the recording of concentration 30 and it was discarded. Additionally, one drawback from in vitro acquisition is the formation of bubbles in water, which resemble to particles in the ultrasound images, this was not an issue except for concentration 0 images that had to be filtered out.

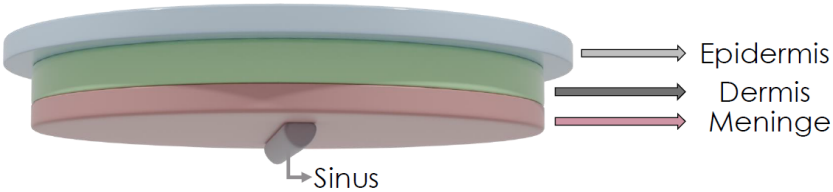

Figure 4.1: Fontanel-mimicking phantom design

The experiment in December recorded images from a phantom of 10.2dB attenuation without hair, but using 42dB and 48dB gain in the Neosonics with a voltage of 100V. The recorded concentrations were 20, 50, 200, 400, 600, 800, 1000 and 1500. However, there was a problem in the recording of concentration 1000 and it was discarded. Additionally, the concentration 1500 was later considered to be excessive and not relevant for the application, so it was excluded from the study.

The Data Department at Newborn Solutions concluded that the image level differences between the experiments were not significant and combining both could result in useful variability.

In vitro peritoneal dialysis

The in vitro peritoneal dialysis acquisition was done in May 2023 using a new version of the Neosonics. Images were recorded directly from the Baxter R5C4476 collector

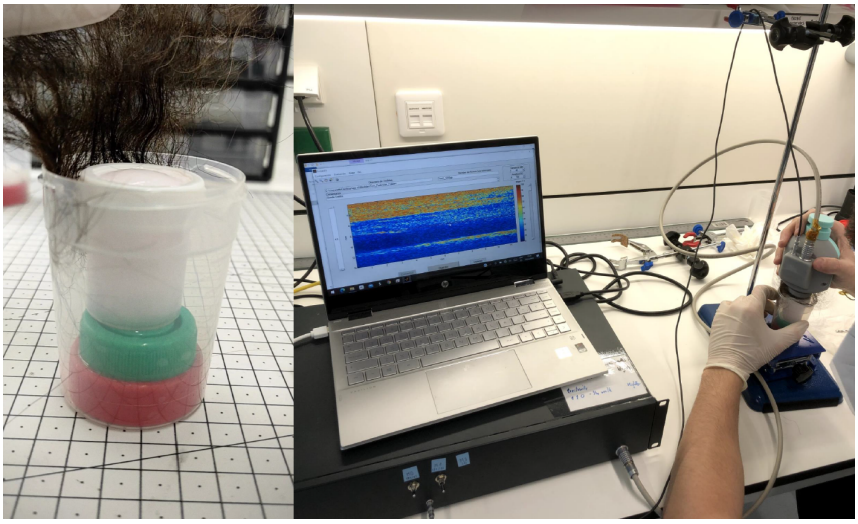

Figure 4.2: Setup for in vitro meningitis acquisition

plastic bag as in the intended real life application. However, the dialysis solution (typically containing a sugar called dextrose or a compound called icodextrin and minerals) was replaced by water and white blood cells by polystyrene particles with 5µm diameter. In the case of plastic bags, the formation of air bubbles is more frequent, while still an issue for the application, it will also occur during in vivo acquistion so not even concentration 0 images were filtered out. The Neosonics configuration was 42dB gain and 100V voltage. The recorded concentrations (pp/µL) were 0, 12, 35, 60, 80, 105, 113, 160, 218, 450, 580 and 1000.

Figure 4.3: Setup for in vitro peritoneal dialysis acquisition

4.1.2 Preprocessing

The preprocessing of images is the specified from the Data Department at Newborn Solutions. The Neosonics ultrasound one-channel raw signals are processed using a Butterworth bandpass filter with lowcut and highcut frequencies of 10×10^6 and 30×10^6, respectively, and a Hilbert transform.

Then, raw images are cropped to only contain the focal region of the image. Since images present some artifacts in their x-axis limits, the focal region is further cut out: 100 pixels in the left and 74 pixels in the right. The final image size is 200×556.

Particle traces are found in intensity values around 3000. Therefore, the dynamic

range is adjusted to enhance their visualization. The upper saturation threshold is set to 5500 and the lower threshold to 1000. With this narrower dynamic range, most of the intensity changes are due to the particles traces. A min-max normalization is later applied to set the images in the range $[0, 255]$.

Afterwards, a Sobel filter with a kernel size of 5 is applied on the x-direction. The visualization of particles traces is enhanced and allows a better differentiation from the liquid background.

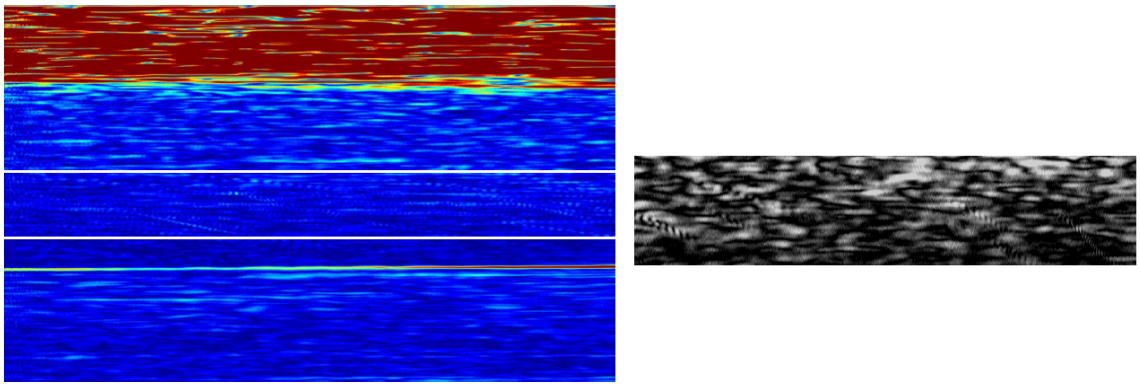

Figure 4.4: Raw image with focal region between white lines (left) and cropped focal region after Butterworth bandpass, Hilbert transform and dynamic range adjustment (right)

As a last step, images are scaled to the range of the Tanh activation function $[-1, 1]$ that will be used by the generative and regression models.

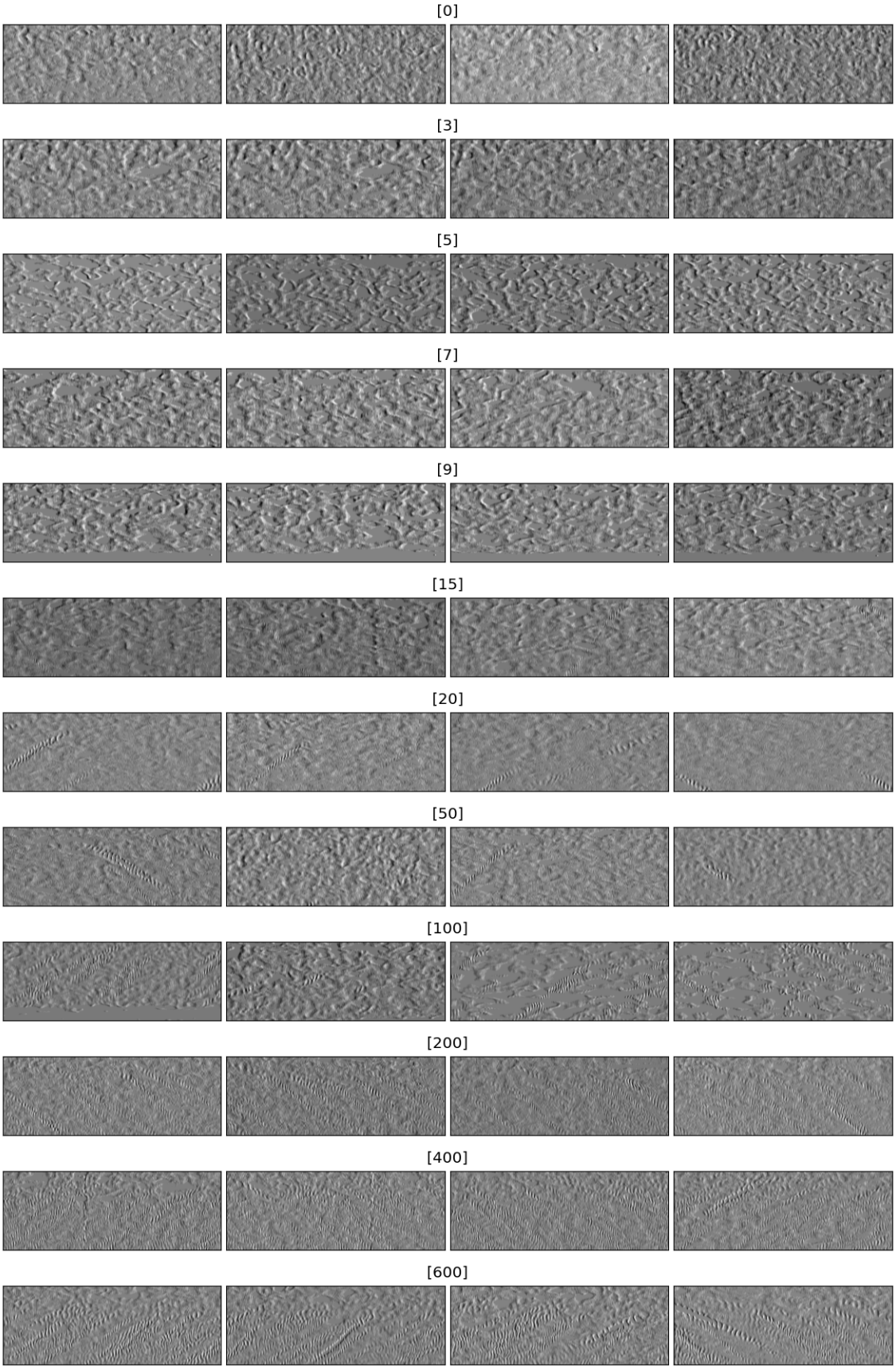

Figure 4.5: Preprocessed in vitro meningitis images

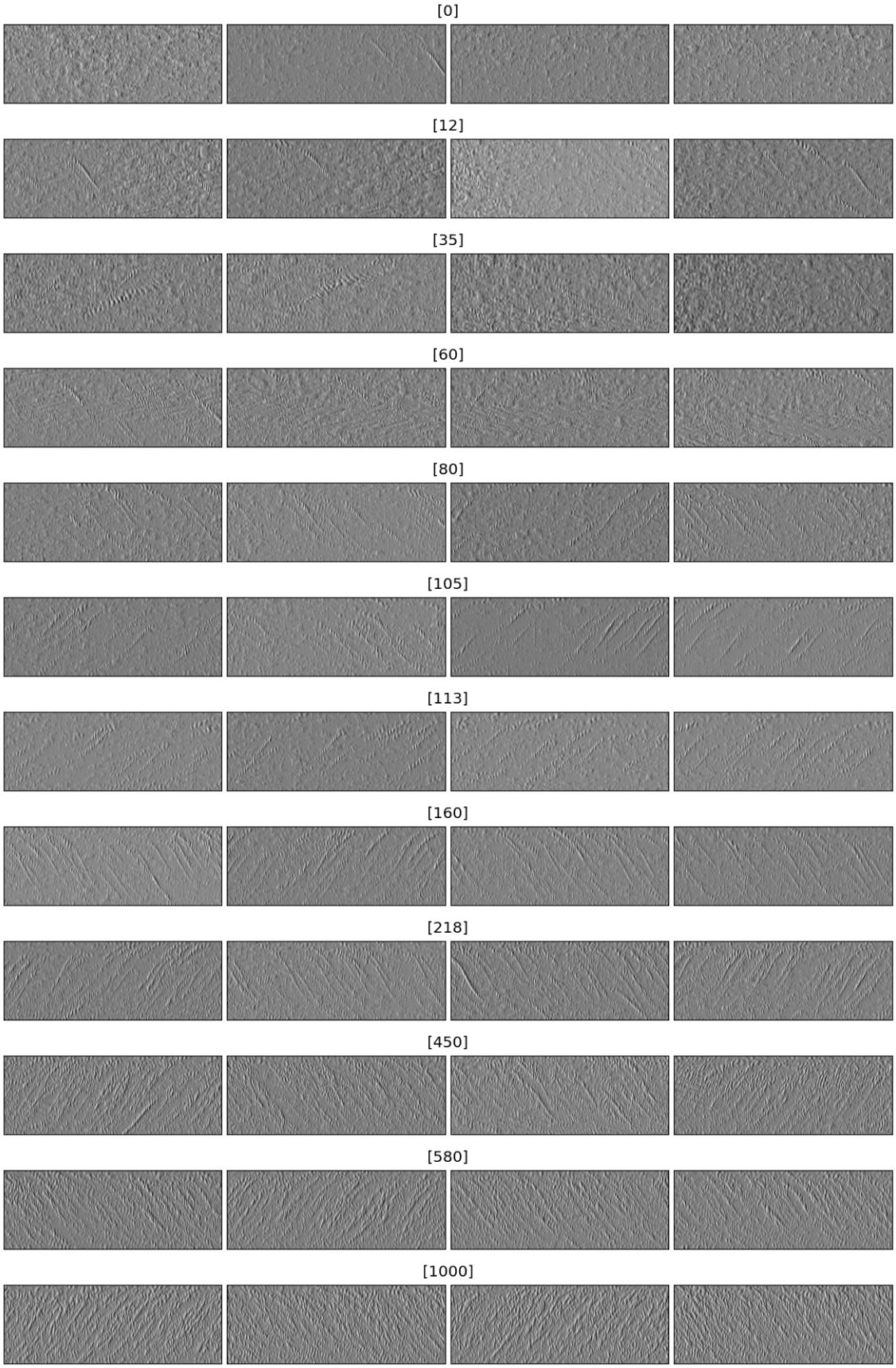

Figure 4.6: Preprocessed in vitro peritoneal dialysis images

4.1.3 Split

The datasets are divided in the usual train, validation and test splits. However, given the purpose of this study and how the data was acquired, there are some particularities in the splitting process.

On the one hand, the concentrations in training and validation are different from the concentrations in testing. The purpose of the generative model is helping the regressor robustness and generalization capability by providing training samples from a wider set of concentrations than the currently available. In order to properly evaluate the results and assess the impact of adding artificially generated images in the training process, it is necessary to have real ground truth images of unseen concentrations to test both the generative and regression models. The training and validation splits would contain the minimum and maximum concentration values and every other of the intermediate concentrations, keeping the rest for the test set.

On the other hand, it was empirically proven that including images from the same video file in training and validation led to overfitting due to visual properties other than particle traces. Therefore, the splitting between training and validation was restricted by the files. This led to a larger validation set than desired since for each training concentration, at least a whole file had to be used for validation.

In vitro meningitis

In the case of meningitis, the chosen training and validation concentrations were 0, 5, 9, 20, 100, 400 and 800, while the concentrations 3, 7, 15, 50, 200 and 600 were reserved for testing. Ideally, training and validation splits would have contained the same concentrations, but concentrations 5 and 9 came from a single recording file each one. Since it was decided that images from the same file should not be used in different splits, the concentration 9 images were used for training and concentration 5 images for validation. Details of the in vitro meningitis dataset and the resulting split on table 4.1.

In vitro peritoneal dialysis

The split for peritoneal dialysis was better structured and divided. For each concentration, 3 video files were recorded. Therefore, 2 files were used for the training and 1 for validation from the chosen training and validation concentrations 0, 35, 80, 113, 218, 580 and 1000. The test set contained all images from concentrations 12, 60, 105, 160 and 450. Details of the in vitro peritoneal dialysis dataset and the resulting split on table 4.2.

4.2 GAN models

”In recent times, GAN has achieved outstanding performance in producing natural images. However, there exist major challenges in training of GANs, i.e., mode collapse, non-convergence and instability, due to inappropriate design of network architecture, use of objective function and selection of optimization algorithm” (Saxena and Cao, 2023).

Besides the inherent difficulties of training GANs, this study had the additional challenges of dealing with non-squared images and continuous scalar conditioning of the models.

After reviewing most relevant GAN papers, none of them addressed the problem of dealing with non-square images. Extending these GANs architectures to work with rectangular images would only have required some minor changes if they had had even dimensions and integer aspect ratio. However, the in vitro image data resolution is 200×556.

Split	Video file	Concentration	Nº Images
Train	20221026_093136	0	81
	20221107_153315	9	77
	20221206_162705	20	61
	20221026_161648	100	97
	20221026_162308	100	89
	20221025_160131	100	83
	20221205_170110	400	87
	20221205_104703	800	97
Validation	20221026_093710	0	62
	20221108_170208	5	68
	20221206_162500	20	50
	20221025_160841	100	79
	20221205_165916	400	75
	20221205_104504	800	96
Test	20221108_161701	3	95
	20221107_160631	7	88
	20221109_103620	15	66
	20221109_104149	15	32
	20221109_104300	15	21
	20221206_103758	50	61
	20221206_103543	50	60
	20221102_101112	50	58
	20221102_102547	50	36
	20221205_162604	200	98
	20221205_162802	200	97
	20221206_170157	600	100
	20221206_170344	600	85

Table 4.1: In vitro meningitis dataset content and split

Similarly, no literature was found to address GAN conditioning on a range of continuous scalar values.

It must be noted that most GAN model design and training experimentation was performed on the in vitro meningitis dataset, which was the only available at the time.

4.2.1 Architecture

After extensive literature research and model exploration, a GAN architecture capable of dealing with the images specific resolution and continuous scalar conditioning is proposed. It follows some general GAN design guidelines, but incorporates novel elements too.

The architecture design is mainly based on the Deep Convolutional Generative Adversarial Networks (DCGANs) proposed by Radford et al. (2016). The core of their approach was adopting and modifying recently demonstrated changes to Convolutional Neural Networks (CNNs) architectures.

Firstly, deterministic spatial pooling functions (such as maxpooling) are replaced with strided convolutions, allowing the generator and discriminator to learn their own spatial upsampling and downsampling, respectively.

Split	Video file	Concentration	Nº Images
Train	20230426_134807	0	98
	20230426_134632	0	97
	20230426_132718	35	89
	20230426_132546	35	87
	20230426_135501	80	98
	20230426_135628	80	98
	20230426_142201	113	98
	20230426_141902	113	92
	20230426_160453	218	97
	20230426_160619	218	97
	20230426_162757	580	97
	20230426_162630	580	95
	20230426_163314	1000	97
	20230426_163604	1000	97
Validation	20230426_134510	0	96
	20230426_132844	35	86
	20230426_135332	80	92
	20230426_142034	113	91
	20230426_160316	218	96
	20230426_162458	580	93
	20230426_163439	1000	94
Test	20230426_133440	12	97
	20230426_133608	12	97
	20230426_133733	12	97
	20230426_131749	60	100
	20230426_132107	60	97
	20230426_131940	60	95
	20230426_140643	105	81
	20230426_140201	105	75
	20230426_140343	105	73
	20230426_155653	160	97
	20230426_155526	160	96
	20230426_155820	160	96
	20230426_161234	450	97
	20230426_161403	450	96
	20230426_161559	450	96

Table 4.2: In vitro peritoneal dialysis dataset content and split

Secondly, implementing fully convolutional neural networks for both the generator and discriminator. The first layer of the generator, which takes a uniform noise distribution Z as input, is the only fully connected layer, but the result is reshaped to be used as the start of the convolution stack. For the discriminator, the flattened output of the last convolutional layer is directly used as output.

In third place, using Batch Normalization (Ioffe and Szegedy, 2015), which normalizes the input to each layer to have zero mean and unit variance. This proved robustness to weights initialization and helped gradient flow in deep models, which is critical to get generators to begin learning. However, batchnorm is not applied to the generator output

layer and the discriminator input layer, since it resulted in sample oscillation and model instability (Radford et al., 2016).

Lastly, relative to the activation functions, the ReLU activation is recommended in the generator with the exception of the output layer which uses the Tanh function. Radford et al. (2016) also found the Leaky ReLU to work well within the discriminator, especially for higher resolution modeling.

Generator

The generator is designed to receive a 128 dimensional uniform distribution Z, commonly referred to as input noise, which is projected to a small spatial extent convolutional representation with several feature maps. Then, a series of 8 carefully designed convolutional layers convert the initial high level representation to the desired one channel 200×556 pixels image. Each of the first 7 convolutional layers consists of a fractionally-strided convolution, also known as transposed convolution, of varying kernel size as follows 3×5, 2×4, 2×3, 2×4, 2×3, 4×4 and 4×4, stride of 2, padding of 1 and no bias, batch normalization and leaky ReLU activation with 0.2 slope, while the last layer is a convolution of 3×3 kernel size, stride of 1 and padding of 1 followed by the Tanh activation.

The initial projection of the 128 dimensional uniform distribution Z proved crucial in the performance of the GAN. Simple projections lead to poor and unsatisfying results. Inspired by the disentanglement studies of the latent space from StyleGAN (Karras et al., 2019), the input is mapped to an intermediate latent space W using a 3-layer MLP with leaky ReLU activation with 0.2 slope except for the output. A disentangled latent space can be understood as consisting of linear subspaces, each of which controls one factor of variation. Training is expected to yield a less entangled intermediate latent space W, making it easier to generate realistic and varied images. Instead of sampling according to a fixed distribution, sampling comes from the learned continuous mapping $\mathbf{z} \rightarrow W$, where the factors of variation are expected to become more linear (Karras et al., 2019).

The key design choice of the generator comes from how to achieve the desired output images dimensional size of 200×556. After experimenting with different approaches, the best performing generator used a combination of varying kernel sizes along its transposed convolutional layers to gradually shape the dimensional size. This approach presented some problems on its own, but it proved to be the best solution.

The use of kernel sizes not divisible by the stride cause uneven overlap of the kernel, which makes some output values to receive signal from a larger number of inputs. This leads to checkerboard pattern artifacts in the generated images. ”Neural nets typically use multiple layers of deconvolution when creating images, iteratively building a larger image out of a series of lower resolution descriptions. While it is possible for these stacked transposed convolutions to cancel out artifacts, they often compound, creating artifacts on a variety of scales” (Odena et al., 2016).

The simpler solution to checkerboard artifacts would be using a kernel size divisible by the stride, avoiding the overlap issue. However, this was not an option given the images dimensional size. The solution suggested by Odena et al. (2016) is to separate out upsampling to a higher resolution, e.g., using nearest-neighbor interpolation or bilinear interpolation, from the convolution to compute features, thus replacing the transposed convolution with a resize-convolution operation. Meanwhile, Kinoshita and Kiya (2020) state that checkerboard artifacts caused by a transposed convolutional layers can be perfectly prevented by using their proposed fixed smooth convolutional layer after the transposed

convolution operations.

Generative modelling literature is biased towards the existing datasets of natural images, e.g., Imagenet, CIFAR-10 or CelebA. Therefore, although both the resize-convolution and fixed smooth convolutional layers methods seemed to work on these datasets, when applied in this study they failed to capture the patterns and details of the in vitro ultrasound images of interest.

In the end, only the transposed convolutions proved able to learn and generate the features and details present in the in vitro ultrasound images.

Finally, as described above and differently from DCGAN, Leaky ReLU activations are used within the generator too in an attempt to help the gradients flow easier through the architecture. Early experiments showed the GAN was suffering from the vanishing gradient problem: the generator was not receiving gradient values from the discriminator, preventing the learning process.

Discriminator

The discriminator is designed to receive one channel 200×556 pixels images as input and generate a scalar output per image. This is achieved with a stack of 7 convolutional layers consisting of a convolution with 4×4 size kernel, stride of 2 and padding of 1, Layer normalization (except for the first layer) and leaky ReLU activation with 0.2 slope. A final convolution with a 3×8 kernel size matching the input size and no bias is applied to output the desired scalar value.

Experimentation proved that the generator had to use a varying kernel architecture, but it proved the opposite in the case of the discriminator. GAN models in literature tend to be symmetrical in terms of upscaling and downscaling rates in the generator and discriminator. However, since CNNs can generate checkerboard artifacts in both forward-propagation of upsampling layers and backpropagation of convolutional layers (Kinoshita and Kiya, 2020), implementing the same varying kernels as the generator in the discriminator resulted in a pronounced checkerboard effect. In an attempt to minimize the artifacts, a kernel size divisible by the stride was used in the discriminator layers except for the output. In fact, this architecture completely eliminated the checkerboard artifacts from the generated images.

Another design difference with the generator is the normalization method. Kurach et al. (2019) observed that adding Batch normalization to the discriminator hurts the performance and recommended using Spectral normalization (Miyato et al., 2018) or Layer normalization (Kurach et al., 2019). In the experiments, Layer normalization offered a better performance.

4.2.2 Loss functions

How to capture the difference between two distributions in GAN loss functions is an area of active research and several methods have been proposed for improving the quality of gradient, handling the issue of non-convergence and providing additional supervision to generator and discriminator during training. The three most common and widely used GAN loss function were explored: minimax, least squares and Wasserstein loss.

GAN

The minimax was introduced as the default loss function along the original GAN paper. The output of discriminator D is the probability of an image coming from the real data distribution. D is trained to maximize log $D(\mathbf{x})$ while G is trained to minimize log$(1 - D(G(\mathbf{z})))$.

In practice, G is trained to maximize $\log D(G(\mathbf{z}))$, which provides stronger gradients in the early steps of training. The GAN objective function can be defined as follows:

$$\max_D V_{GAN}(D) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[1 - \log D(G(\mathbf{z}))]$$
$$\max_G V_{GAN}(G) = \mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[\log D(G(\mathbf{z}))]$$

In terms of implementation, the discriminator can be seen as a binary classifier where real images are assigned the positive class 1 and generated images the negative class 0. Therefore, the binary cross entropy loss function is used. For the discriminator loss, the target for real images is the positive class and the target for fake images is the negative class, while for the generator the loss is computed using the discriminator output on fake images with the positive class as target.

It must be noted that the discriminator does not apply a sigmoid activation to its output. Instead, the sigmoid is applied afterwards combined with the crossentropy loss, taking advantage of the log-sum-exp trick for numerical stability.

LSGAN

Mao et al. (2017) argue that the original sigmoid cross entropy loss leads to the problem of vanishing gradients when updating the generator using the fake samples that are on the correct side of the decision boundary (0.5), but still far from the real data. Those samples will cause almost no signal error for the generator to update and try to pull them closer to the real data. Based on this observation, Mao et al. (2017) introduce Least Squares Generative Adversarial Networks (LSGANs), which adopt the least squares loss function for the output f the discriminator. ”The idea is simple yet powerful: the least squares loss function is able to move the fake samples toward the decision boundary, because the least squares loss function penalizes samples that lie in a long way on the correct side of the decision boundary” (Mao et al., 2017). The LSGAN objective function can be defined as follows:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[(D(\mathbf{x}) - b)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[(D(G(\mathbf{z})) - a)^2]$$
$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[(D(G(\mathbf{z})) - c)^2]$$

where a and b are the labels for fake data and real data, respectively, and c denotes the value that G wants D to believe for fake data.

One approach to determine the values of a, b and c is to satisfy the conditions of $b-c = 1$ and $b - a = 2$, for example, by setting $a = -1$, $b = 1$ and $c = 0$, such that the minimization problem yields minimizing the Pearson χ^2 divergence between $p_{data} + p_g$ and $2p_g$ (Mao et al., 2017). Another valid approach is to make G generate samples as real as possible by setting $c = b$. In practice, Mao et al. (2017) observed similar performance in their experiments.

In terms of implementation, the second approach was used for being more intuitive and similar to the original GAN loss. The label for fake data was set to $a = 0$ and the label for real data and the value that G wants D to believe for fake data were set to $b = c = 1$.

WGAN

Arjovsky et al. (2017) argue that the divergences which GANs typically minimize are potentially not continuous with respect to the generator’s parameters, leading to training difficulties. Instead, it is proposed to use the *Earth-Mover* or Wasserstein-1 distance

$W(q,p)$, "which is informally defined as the minimum cost of transporting mass in order to transform the distribution q into the distribution p (where the cost is mass times transport distance). Under mild assumptions, $W(q,p)$ is continuous everywhere and differentiable almost everywhere" (Gulrajani et al., 2017). The WGAN objective function can be defined as follows:

$$\max_D V_{WGAN}(D) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))]$$
$$\max_G V_{WGAN}(G) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z})]$$

where discriminator D is restricted to 1-Lipschitz and it is called a *critic* since it is not trained to classify, but rather to output a scalar score to samples.

The WGAN value function results in a critic whose gradients make the optimization and training of the generator easier and more stable. Empirically, Arjovsky et al. (2017) observed that the WGAN value function correlates with samples quality.

Originally, the weights of the critic were clipped to lie within a compact space $[-c, c]$ to enforce the Lipschitz constraint on the critic. However, Gulrajani et al. (2017) found that weight clipping in WGAN biases the critic towards much simpler functions and leads to optimization difficulties, which can be mitigated to some extent with Batch normalization in the critic and careful tuning of the clipping threshold c, but deep WGAN critics would still often fail to converge as a result of either vanishing or exploding gradients (Gulrajani et al., 2017).

Gulrajani et al. (2017) propose an alternative way to enforce the Lipschitz constraint. A differentiable function is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere, so it is possible to directly constraint the gradient norm of the critic's output with respect to its input. In practice, a soft version of the constraint is enforced by adding a penalty of the gradient norm for random samples $\hat{\mathbf{x}} \sim p_{\hat{x}}(\hat{\mathbf{x}})$. The WGAN-GP objective function can be defined as follows:

$$\max_D V_{WGAN-GP}(D) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))] - \lambda \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{x}}(\hat{\mathbf{x}})}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]$$
$$\max_G V_{WGAN-GP}(G) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z})]$$

$p_{\hat{x}}$ is implicitly defined by sampling uniformly along straight lines between pairs from the real data distribution p_{data} and the generator distribution p_g. Enforcing the unit gradient norm constraint everywhere is intractable, but enforcing it along these straight lines seems sufficient and experimentally results in good performance (Gulrajani et al., 2017). The penalty coefficient λ is used to weight the gradient penalty term. Gulrajani et al. (2017) found $\lambda = 10$ to work well across a variety of architectures and datasets.

Finally, Batch normalization changes the discriminator problem from mapping a single input to a single output to mapping from an entire batch of inputs to a batch of outputs. The gradient penalty objective is not valid in this setting, since it is intended to penalize the norm of the critic's gradient with respect to each input independently, and not the entire batch (Gulrajani et al., 2017). Therefore, the WGAN-GP loss function requires to avoid Batch normalization in the discriminator and Layer normalization is recommended, which is already included in the architectural design of the discriminator in this study.

4.2.3 Conditioning

Mirza and Osindero (2014) introduced Conditional Generative Adversarial Networks (CGAN), an extension of GANs where the model is conditioned on some auxiliary information **y**, such as class labels, text or images, introduced as an additional input to both the

generator and discriminator. Usually the condition **y** requires some kind of transformation to be concatenated to the input.

In this study, the conditioning of the GAN models is a primary goal. The challenge is exploring the possibility of conditioning and generating images in a range of continuous scalar values representing the concentration of particles (pp/µL). Once again, multiple conditioning methods were explored and chosen based on experimentation done over the in vitro meningitis dataset. Even a StarGAN approach was implemented, which was able to perform image-to-image translations for multiple domains using only a single model. However, the complexity of the training and some initial results that did not imprve previous models discouraged the continuity of its exploration.

Similarly to the input noise, the given condition is passed through a 3-layer MLP with leaky ReLU activation with 0.2 slope except for the output to transform and map it to the desired dimensionality. For the conditioning of the generator, there were two possibilities: noise and channel concatenation.

On the first approach, the 3-layer MLP maps the condition to the same dimensionality of the input noise and they are concatenated to form a single input for the generator.

On the second approach, the 3-layer MLP maps the condition to the same dimensionality of the input noise projection, a small spatial extent convolutional representation with several feature maps. Then, concatenation occurs channel wise.

The noise concatenation approach was unsuccessful in providing an effective conditioning to the GAN model. It was the channel concatenation approach that managed to condition the generator effectively, clearly impacting the generation of images.

Although the discriminator does not directly contribute to image generation, it should also be conditioned during training as originally described by Mirza and Osindero (2014). The input to the discriminator is a 200$\times$556 image, so the condition must also be mapped to this dimensionality and concatenated channel wise. Using a 3-layer MLP with leaky ReLU activation with 0.2 slope except for the output to map to such high dimensionality is costly in terms of network parameters, but it resulted to be the only effective approach. Repeating the condition up to the desired dimensionality was an idea to reduce the number of parameter, but it did not work neither in the generator nor in the discriminator, proving that a learned mapping is crucial.

Odena et al. (2017) proposed a variant of the GAN architecture known as Auxiliary Classifier GAN (AC-GAN), which can be seen as a different conditioning method. In this case, the generator G is still conditioned but not the discriminator D, which instead is designed to output both a value relative to the image source (real or generated) and a value concerning its label via an auxiliary head of the discriminator D_{aux}. The conditioning learning is introduced as an additional loss term LC valued as the log-likelihood of the correct class y.

$$LC = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[\log P(Y = y|D_{aux}(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log P(Y = y|D_{aux}(G(\mathbf{z}, y)))]$$

Therefore, the discriminator D is trained to maximize $V_{GAN}(D) - LC$ while G is trained to maximize $V_{GAN}(G) - LC$. According to Odena et al. (2017), this modification to the standard GAN formulation produces excellent results and appears to stabilize training.

The Auxiliary Classifier conditioning approach had to be adapted to the conditioning in this study: a range of continuous scalar values representing concentrations.

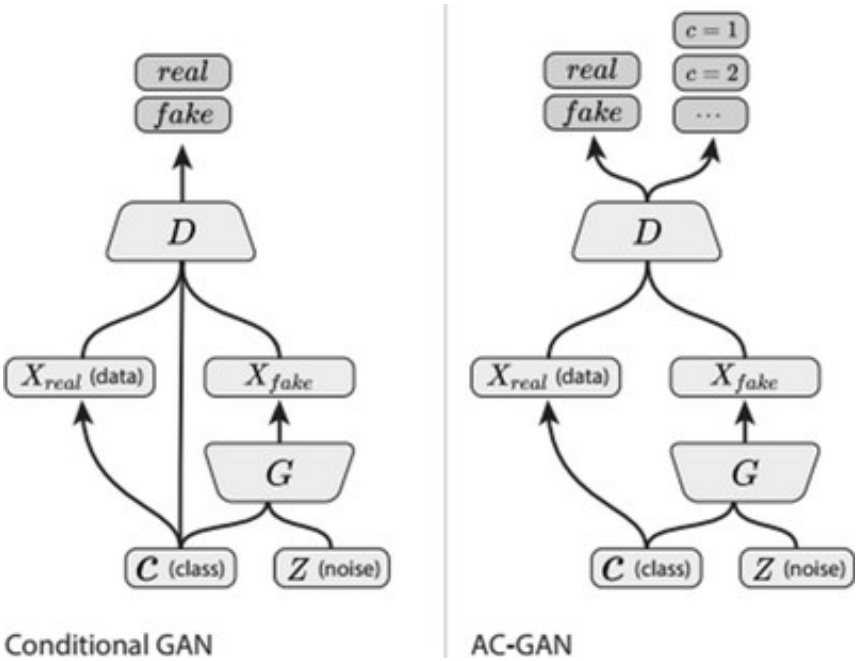

Figure 4.7: Conditional GAN (C-GAN) and Auxiliary Classifier GAN (AC-GAN) architectures, where x denotes the real image, c the class label, z the noise vector, G the Generator and D the Discriminator (Mino and Spanakis, 2018)

Firstly, the discriminator auxiliary head is implemented as a separate convolution with a 3×8 kernel size to match the input size and no bias, just like the applied in the original discriminator head. The output is a scalar value corresponding to the conditioned concentration.

Then, the log-likelihood is replaced by the Mean Absolute Error (MAE) as the loss used for the conditioning of the auxiliary head. As discussed later, MAE is chosen over Mean Square Error (MSE) for its robustness to outliers. The loss term LC is redefined for the condition y as:

$$LC(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(x)}[|y - D_{aux}(\mathbf{x}))|] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[|y - D_{aux}(G(\mathbf{z}, y))|]$$
$$LC(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\text{z}}(\mathbf{z})}[|y - D_{aux}(G(\mathbf{z}, y))|]$$

It must be noted that the Auxiliary Classifier approach was originally conceived to be used along the minimax GAN loss, but its extension to other losses is rather intuitive and has proven successful in other researches.

As a final remark, min max normalization was applied to the given conditioning concentrations for the stability of the 3-layer MLP and the GAN training in general.

4.2.4 Training

The training of the GAN models is described in the following algorithms 1, 2, 3 for each of the losses.

Relative to the algorithms, it is important to note that real and generated images are sampled in separated minibatches as suggested by (Radford et al., 2016), who observed that Batch Normalization works better this way, which may not be intuitive.

Algorithm 1 General training scheme for Conditional and Auxiliary Classifier Generative Adversarial Networks

Input: Training dataset with pairs of data **x** and labels **y**, generator G with parameters θ_g, discriminator D with parameters θ_d, the number of discriminator iterations per generator iteration k and batch size b

Output: Generator G with parameters θ_g

for number of training iterations **do**

for k steps **do**

Sample minibatch of b samples of real data images **x** and labels **y**

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the discriminator by descending its stochastic gradient:

if C-GAN **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\left[\log D(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) + \log\Big(1 - D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})\Big)\right]$$

else if AC-GAN **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\Bigg[\log D(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) + \log\Big(1 - D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})\Big) + \frac{1}{2}\left[\left|\mathbf{y}^{(i)} - D(\mathbf{x}^{(i)})\right| + \left|\mathbf{y}^{(i)} - D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)})\right|\right]\Bigg]$$

end if

end for

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the generator by descending its stochastic gradient:

if C-GAN **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\log D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})$$

else if AC-GAN **then**

$$\nabla\theta_g \frac{1}{b}\sum_{i=1}^{b}\log D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)}) + \left|\mathbf{y}^{(i)} - D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})\right|$$

end if

end for

The gradient-based updates can use any standard gradient-based learning rule.

Algorithm 2 General training scheme for Conditional and Auxiliary Classifier Least Squares Generative Adversarial Networks

Input: Training dataset with pairs of data **x** and labels **y**, generator G with parameters θ_g, discriminator D with parameters θ_d, the number of discriminator iterations per generator iteration k, batch size b, the label for fake data a, the label for real data b and the value that G wants D to believe for fake data c.

Output: Generator G with parameters θ_g

for number of training iterations **do**

for k steps **do**

Sample minibatch of b samples of real data images **x** and labels **y**

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the discriminator by descending its stochastic gradient:

if C-LSGAN **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\frac{1}{2}\left[(D(\mathbf{x}^{(i)},\mathbf{y}^{(i)})-b)^2+(D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})-a)^2\right]$$

else if AC-LSGAN **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\frac{1}{2}\Bigg[(D(\mathbf{x}^{(i)})-b)^2+(D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}))-a)^2+ \left[\left|\mathbf{y}^{(i)}-D_{aux}(\mathbf{x}^{(i)})\right|+\left|\mathbf{y}^{(i)}-D_{aux}(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)})\right|\right]\Bigg]$$

end if

end for

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the generator by descending its stochastic gradient:

if C-LSGAN **then**

$$\nabla\theta_g \frac{1}{b}\sum_{i=1}^{b}\frac{1}{2}D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}),\mathbf{y}^{(i)})-c)^2$$

else if AC-LSGAN **then**

$$\nabla\theta_g \frac{1}{b}\sum_{i=1}^{b}\frac{1}{2}\left[(D(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}))-c)^2+\left|\mathbf{y}^{(i)}-D_{aux}(G(\mathbf{z}^{(i)},\mathbf{y}^{(i)}))\right|\right]$$

end if

end for

The gradient-based updates can use any standard gradient-based learning rule.

Algorithm 3 General training scheme for Conditional and Auxiliary Classifier Wasserstein Generative Adversarial Networks with Gradient Penalty

Input: Training dataset with pairs of data **x** and labels **y**, generator G with parameters θ_g, discriminator D with parameters θ_d, the number of critic iterations per generator iteration k, batch size b and gradient penalty coefficient λ

Output: Generator G with parameters θ_g

for number of training iterations **do**

for k steps **do**

Sample minibatch of b samples of real data images **x** and labels **y**

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Sample minibatch of b linearly interpolated samples $\hat{\mathbf{x}}$ from the real images **x** and generated images $\tilde{\mathbf{x}} = G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)})$ such that $\hat{\mathbf{x}} = \epsilon\mathbf{x} + (1-\epsilon)\tilde{\mathbf{x}}$, where $\epsilon \sim U[0,1]$.

Update the discriminator by descending its stochastic gradient:

if C-WGAN-GP **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\left[D(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - D(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)}), \mathbf{y}^{(i)})) + \lambda(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2\right]$$

else if AC-WGAN-GP **then**

$$\nabla\theta_d \frac{1}{b}\sum_{i=1}^{b}\Bigg[D(\mathbf{x}^{(i)}) - D(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)})) + \lambda(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2 + \frac{1}{2}\left[\left|\mathbf{y}^{(i)} - D_{aux}(\mathbf{x}^{(i)})\right| + \left|\mathbf{y}^{(i)} - D_{aux}(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)}))\right|\right]\Bigg]$$

end if

end for

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the generator by descending its stochastic gradient:

if C-WGAN-GP **then**

$$\nabla\theta_g \frac{1}{b}\sum_{i=1}^{b} D(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)}), \mathbf{y}^{(i)}))$$

else if AC-WGAN-GP **then**

$$\nabla\theta_g \frac{1}{b}\sum_{i=1}^{b}\left[D(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)})) + \frac{1}{2}\left|\mathbf{y}^{(i)} - D_{aux}(G(\mathbf{z}^{(i)}, \mathbf{y}^{(i)}))\right|\right]$$

end if

end for

The gradient-based updates can use any standard gradient-based learning rule.

For training, models weights were initialized from a zero-centered Normal distribution with standard deviation 0.02 (Radford et al., 2016). All models were trained for 3000 epochs with batch size of 64 using the Adam optimizer (Kingma and Ba, 2017) with learning rate of 5×10^{-5}. However, for the GAN and LSGAN models the β_1 for Adam optimizer is set to 0.5, since Radford et al. (2016) reported the default value of 0.9 resulted in training oscillation and instability, and the β_2 to the default of 0.999. For the WGAN-GP model, the recommended settings $\beta_1 = 0$ and $\beta_2 = 0.9$ by Gulrajani et al. (2017) are used. Additionally, the training dataset was doubled by applying an horizontal flip on all images.

Experimentation and training was GPU-accelerated using the Nvidia Tesla A100 Ampere of 40GB and Nvidia Tesla V100 of 32GB available thanks to the High Performance Computing (HPC) services offered by DTU Computing Center (DCC). The Nvidia GeForce RTX 3090 of 24GB from Newborn Solutions was available too.

4.3 Regression model

The goal of Newborn Solutions for both the meningitis and peritoneal dialysis applications is estimating the concentration of white blood cells in serous fluids from ultrasound images. Technically, concentrations take continuous values, resulting in an image regression task. As part of this study, a regression model R was implemented as a feed-forward deep Convolutional Neural Network (CNN) with parameters θ_r.

First, the regression model was trained with real image data from the in vitro meningitis and peritoneal dialysis dataset, establishing baseline regression models for each application. Then, training the regressor with GAN-based data augmentation is expected to increase the generalization and the performance, specially in unseen concentrations.

4.3.1 Architecture

The regressor had to be designed to receive one channel 200×556 pixels images as input and generate a scalar output per image. At first, similarly to the discriminator in GAN models, this was achieved with a stack of 7 convolutional layers consisting of a convolution with 3×3 size kernel, stride of 1 and padding of 1, Batch normalization (except for the first layer), leaky ReLU activation with 0.2 slope and a 2×2 max pooling. A final convolution with a 3×8 kernel size matching the input size and no bias is applied to output the desired scalar value. The differences with the discriminator in GAN models is that downsampling is done via max pooling instead of using a stride of 2 and the use of Batch normalization.

However, initial experiments revealed this fully convolutional architecture was overfitting to the training data and not generalizing properly to the validation set. As a regularization method, dropout (Srivastava et al., 2014) was implemented. The term refers to randomly dropping out units in a neural network, which means to temporarily remove them along their incoming and outgoing connections.

However, dropout was originally intended to be applied in fully connected layers and the benefits of using dropout in convolutional layers could be limited (Tompson et al., 2015). Therefore, the previous last convolution is extended with Batch normalization and leaky ReLU activation with 0.2 slope. Then, the convolutional output is flattened and passes through a 2-layer MLP with dropout and leakyReLU activation with 0.2 slope on the hidden layer. The dropout probability is set at 0.5, which proved optimal for a wide range of networks and tasks according to Srivastava et al. (2014).

4.3.2 Loss functions

There exist a variety of different loss functions for regression tasks, and which one to choose can depend on a number of factors including the regression model, derivatives calculation and time efficiency or the presence of outliers in data. The 2 most commonly used losses were considered: Mean Square Error (MSE) and Mean Absolute Error (MAE).

The Mean Square Error (MSE) is the mean of squared differences between the target **y** and predicted $\tilde{\mathbf{y}}$ values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}^{(i)} - \tilde{\mathbf{y}}^{(i)})^2$$

The Mean Absolute Error (MAE) is the mean of absolute differences between the target **y** and predicted $\tilde{\mathbf{y}}$ values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\mathbf{y}^{(i)} - \tilde{\mathbf{y}}^{(i)}\right|$$

The MSE loss tends to weight outliers more due to the squaring of errors. Both the in vitro meningitis and peritoneal dialysis datasets are prone to outliers, specially in lower concentrations, where individual frames might not be that representative of the real concentration due to a lack or excess of particle traces in that particular image. Therefore, MAE is chosen over Mean Square Error (MSE) for its robustness to outliers.

4.3.3 Training

The training of regression models is described in algorithm 4.

Algorithm 4 General training scheme for Regression model

Input: Training dataset with pairs of data **x** and labels **y**, regressor R with parameters θ_r and batch size b

Output: Regressor R with parameters θ_r

for number of training iterations **do**

Sample minibatch of b samples of images **x** and labels **y**

Sample minibatch of b noise samples **z** from univariate Gaussian noise prior $p_z(\mathbf{z})$

Update the regressor by descending its stochastic gradient:

$$\nabla\theta_r \frac{1}{b}\sum_{i=1}^{b}\left[\left|R(\mathbf{x}) - \mathbf{y}^{(i)}\right|\right]$$

end for

The gradient-based updates can use any standard gradient-based learning rule.

For training, models weights were initialized from a zero-centered Normal distribution with standard deviation 0.02 (Radford et al., 2016). All models were trained for 300 epochs with batch size of 64 using the Adam optimizer with learning rate of 5×10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Additionally, the training and validation datasets were doubled by applying an horizontal flip on all images.

In order to make a fair comparison of the different regression models, the number of images per concentration was fixed to 1000 and 500 for training and validation, respectively. In the case of the baseline regression model, missing images were randomly re-sampled, while for the GAN augmented regressors they were generated on the fly.

Experimentation and training was GPU-accelerated using the Nvidia Tesla A100 Ampere of 40GB and Nvidia Tesla V100 of 32GB available thanks to the High Performance Computing (HPC) services offered by DTU Computing Center (DCC). The Nvidia GeForce RTX 3090 of 24GB from Newborn Solutions was available too.

4.3.4 Evaluation

The evaluation of regression models is based on error metrics over the corresponding in vitro meningitis or peritoneal dialysis images from the test set.

The error metrics used are the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

The Root Mean Square Error (RMSE) is the squared root of the mean of squared differences between the target $\mathbf{y}$ and predicted $\tilde{\mathbf{y}}$ values.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}^{(i)} - \tilde{\mathbf{y}}^{(i)})^2}$$

The Mean Absolute Percentage Error (MAPE) is the mean of the absolute percentage errors between the target $\mathbf{y}$ and predicted $\tilde{\mathbf{y}}$ values.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\mathbf{y}^{(i)} - \tilde{\mathbf{y}}^{(i)}}{\mathbf{y}^{(i)}}\right|$$

Error metrics are reported for the general regression model performance but also for each concentration separately in order to assess possible differences in performance between lower and higher concentrations.

As a final remark, the model is evaluated using an inference setting where negative values are mapped to 0, since negative concentrations are not feasible.

4.4 Evaluation

The evaluation of this study consists in both the evaluation of GAN models performance in terms of visual quality and conditional generative capacities but also on the impact of GAN data augmentation in the performance of the regression model. However, it must be noted that GAN data augmentation is only performed with GAN models that proved good enough generative capacities.

As previously discussed, the Fréchet Inception Distance (FID) is the current standard metric for assessing the quality of generative models. However, the evaluation of conditional image generation should assess both image quality and conditional consistency. There is research regarding quantitative evaluation of conditional generative models, and the two metrics Conditional Inception Score (CIS) and Conditional Fréchet Inception Distance (CFID) presented by Benny et al. (2021) were implemented as a form of evaluation in this study, but the results and conclusions that could be drawn were as uninformative as the regular FID.

The approach to evaluate both the image quality and conditioning capacity of the generative models was to compute matrices of FID scores between subsets of images from each concentration. Concretely, three different matrices are used for an overall evaluation.

The first FID matrix is computed between subsets of real images to establish a reference of how real images from same and different concentrations compare among them.

The two subsets of real images are sampled randomly and independently in a 1:1 ratio from each train, validation and test set.

The second FID matrix is the one that compares real images with the generated images for all concentrations. The real images include all the images from the validation and test sets, the training set the GAN model was trained on is not included. This FID matrix is useful to evaluate the quality of the generated images by comparing same concentrations FID scores with the reference matrix. Additionally, the conditioning of the GAN can be assessed by comparing different concentration FID scores: the greater difference in concentration the higher the FID score should be.

Lastly, the FID matrix among subsets of generated images from all concentrations is used to evaluate the impact of conditioning. Once again, with proper conditioning, differences in concentration should reflect on higher FID scores.

Once a GAN model proved good enough generative capabilities, the final evaluation consisted in training a regression model using both real and generated images from an extended training and validation set of concentrations and assessing the performance on the test set. The new training and validation sets were defined to contain the original available concentrations and the intermediate concentration between consecutive original concentrations. Therefore, for the in vitro meningitis dataset these concentrations were 0, 2.5, 5, 7, 9, 14.5, 20, 60, 100, 250, 400, 600 and 800, while for the in vitro peritoneal dialysis dataset the concentrations were 0, 12.5, 35, 57.5, 80, 96.5, 113, 165.5, 218, 399, 580, 790 and 1000.

This final evaluation is directly related to the ultimate goal of this study and Newborn Solutions: improving the performance of regression models via GAN-based data augmentation.

4.5 Results

4.5.1 GAN models

The training of GAN models was challenging and required numerous experiments and iterations to first achieve converging models and then improving the performance. The following is an outline of the the main experiments, findings and final results for each of the models and datasets. Onwards, WGAN-GP models are referred simply as WGAN for conciseness.

In vitro meningitis

Firstly, the FID matrix of in vitro meningitis images was computed to establish a reference.

	Real [0]	Real [3]	Real [5]	Real [7]	Real [9]	Real [15]	Real [20]	Real [50]	Real [100]	Real [200]	Real [400]	Real [600]	Real [800]
Real [0]	5.8	64.9	177.1	134.5	207.7	84.5	115.6	55.9	91.7	106.8	125.7	93.3	132.8
Real [3]	71.7	15.4	151.8	65.5	141.0	36.3	78.8	63.3	98.2	122.3	140.0	144.4	171.1
Real [5]	196.8	174.3	23.1	115.1	73.3	148.0	255.3	242.7	150.8	310.3	309.1	249.9	292.5
Real [7]	141.1	74.1	119.1	21.3	92.9	81.0	134.8	132.8	126.6	190.5	197.2	202.3	215.3
Real [9]	206.3	152.0	76.4	89.6	19.5	133.9	215.7	217.3	146.1	275.2	276.2	251.9	275.6
Real [15]	79.9	35.6	143.7	75.1	137.6	19.6	57.2	51.1	74.8	114.2	133.3	131.4	161.8
Real [20]	108.6	74.8	226.3	123.3	203.0	58.6	13.2	40.9	91.6	68.8	88.5	120.1	129.4
Real [50]	58.4	58.8	224.5	129.4	216.8	59.3	40.1	11.5	86.8	55.9	81.6	93.5	114.4
Real [100]	87.5	93.1	128.6	116.7	144.4	70.9	102.6	85.9	8.3	135.9	141.4	98.5	134.7
Real [200]	101.0	114.0	292.7	184.2	277.2	118.0	62.8	51.4	129.8	8.8	23.1	59.5	55.4
Real [400]	122.6	131.8	293.9	187.4	275.6	134.0	77.8	76.9	136.8	23.4	7.6	52.5	33.4
Real [600]	86.3	129.6	224.9	183.3	244.5	120.7	111.8	82.3	89.7	57.4	47.7	9.4	28.4
Real [800]	131.3	165.3	280.1	206.1	278.4	160.6	123.0	108.7	130.7	52.8	26.7	26.4	7.2

Figure 4.8: Reference FID matrix on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The reference FID matrix is coherent in general: lower FID scores within the same concentrations (in the diagonal) and higher FID scores the larger the concentration difference (the further from the diagonal). This is an indicator of the expected differences of images from different concentrations. However, the trend presents some fluctuations, which could mean there are several sources of variation and not only the number of particles traces.

C-GAN/AC-GAN

The Conditional and Auxiliary Classifier GAN models using the minimax loss function on the in vitro meningitis dataset were the first implemented and they posed all the common challenges of GAN training: vanishing gradients, mode collapse, non-convergence and instability.

In order to fix the instability and vanishing gradients issue, Arjovsky and Bottou (2017) suggested adding continuous noise to the inputs of the discriminator, therefore smoothing the distribution of the probability mass. A similar approach to reduce the vulnerability of neural networks to adversarial examples is label smoothing, which consists in replacing the 0 and 1 fake and real targets for the discriminator with smoothed values, like 0.1 or 0.9 (Salimans et al., 2016). These solutions intend to discourage the discriminator from being overconfident and relying on a limited set of features to determine if an image is real or generated.

Unfortunately, none of these techniques and no amount of hyperparameter tuning were enough to achieve convergence and avoid mode collapse: one of the main failure modes of GANs where the generator collapses to a parameter setting where it always emits a unique or reduced set of outputs. Therefore, the generated visual features are very limited.

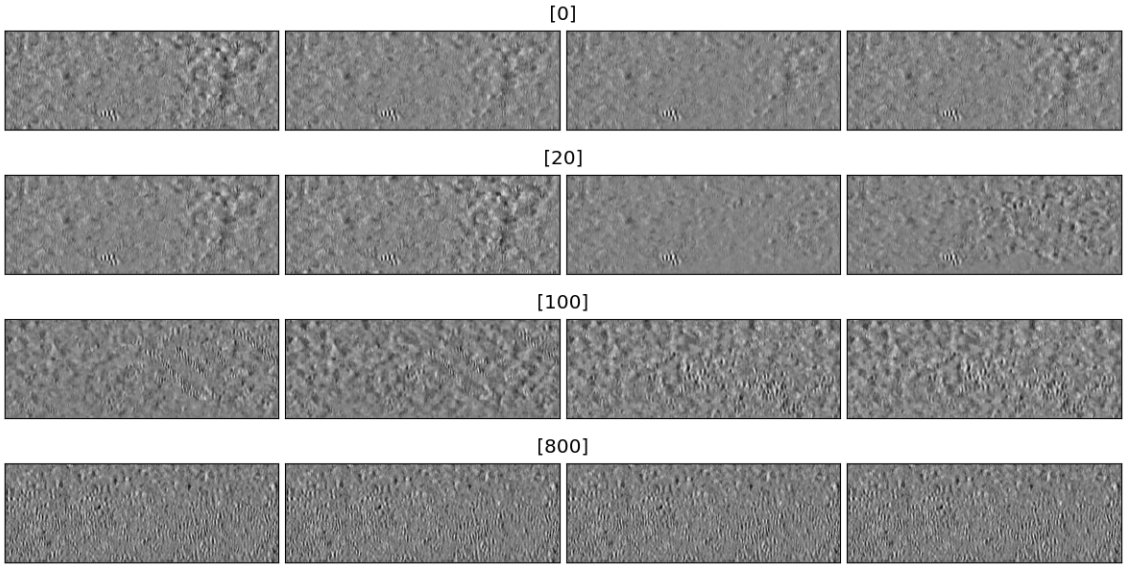

Figure 4.9: C-GAN generated images of in vitro meningitis

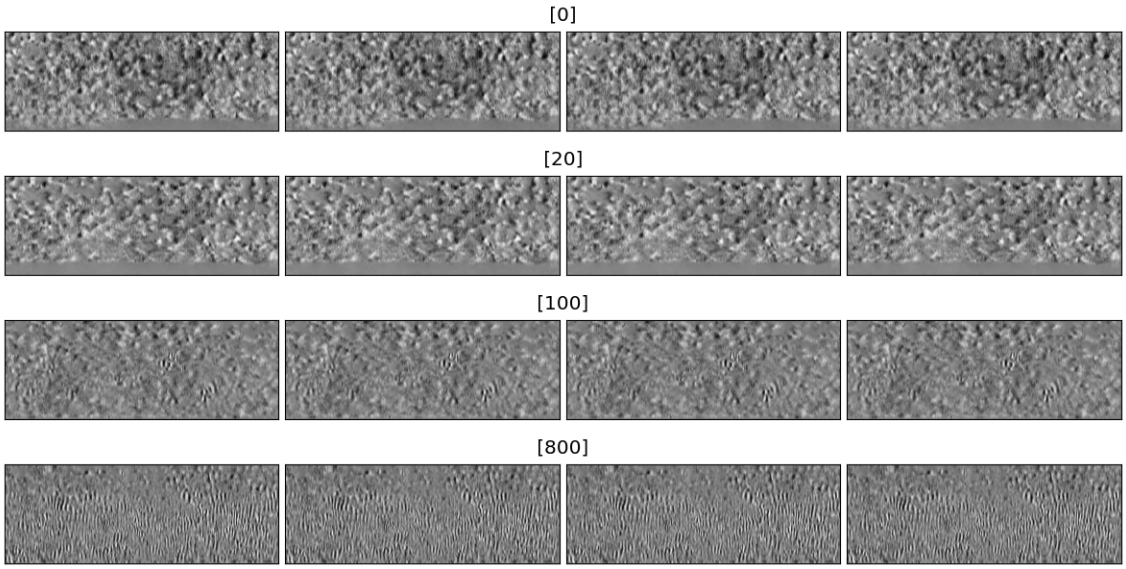

Figure 4.10: AC-GAN generated images of in vitro meningitis

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	159.0	165.6	166.5	167.5	173.4	180.9	256.3	144.7	210.0	331.4	261.6	241.9
Real [3]	141.4	147.6	147.9	148.4	152.6	157.3	220.9	138.0	176.4	309.8	260.3	240.5
Real [5]	312.0	318.4	320.9	321.0	330.5	340.9	416.9	340.6	422.6	510.4	443.7	421.9
Real [7]	197.9	202.8	203.6	203.7	207.9	211.2	268.5	222.1	254.6	358.0	316.7	294.8
Real [15]	147.8	153.7	154.6	155.6	161.1	166.9	248.2	168.7	206.5	338.0	285.4	266.7
Real [20]	210.9	217.5	219.4	219.7	227.5	236.5	307.0	216.1	294.7	411.1	344.6	323.0
Real [50]	144.8	150.3	151.3	152.2	156.7	162.6	245.6	149.3	174.9	304.2	248.3	233.5
Real [100]	140.0	145.2	146.4	147.4	152.4	157.1	242.4	170.0	189.3	308.8	261.5	243.8
Real [200]	158.6	162.8	163.1	165.4	168.6	171.0	258.3	174.8	153.2	255.8	206.4	199.9
Real [400]	166.1	170.1	170.4	172.7	175.5	177.3	267.8	187.3	156.0	251.4	204.2	197.5
Real [600]	195.2	199.7	201.1	202.7	208.6	215.6	307.5	209.5	236.2	330.1	263.9	251.8
Real [800]	207.7	211.0	212.0	214.1	218.0	221.7	310.7	232.5	228.6	300.1	244.0	235.3

Figure 4.11: Real-Fake FID matrix of C-GAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	249.8	237.3	235.0	225.2	207.6	203.2	382.4	193.0	220.8	251.6	226.9	264.9
Real [3]	216.1	205.4	203.6	195.3	185.2	186.1	346.4	187.7	182.8	244.3	228.1	266.4
Real [5]	333.8	320.8	306.3	280.1	248.3	227.9	477.6	296.7	420.4	442.9	416.9	440.2
Real [7]	228.1	216.6	209.8	196.2	181.5	185.5	371.6	247.1	257.8	300.5	283.4	314.8
Real [15]	244.6	232.2	228.9	217.9	203.8	201.6	374.1	196.5	207.2	259.5	242.6	285.4
Real [20]	277.4	263.4	255.7	237.8	211.0	196.3	406.0	217.0	288.2	335.9	312.6	347.7
Real [50]	263.4	251.2	252.5	247.1	240.0	242.9	389.8	212.2	179.6	212.3	192.1	239.2
Real [100]	238.0	225.3	223.3	214.8	205.0	204.4	360.3	196.8	189.4	220.7	205.6	247.9
Real [200]	280.5	272.5	277.1	275.6	278.8	291.0	398.4	240.5	162.0	141.1	122.1	168.7
Real [400]	279.2	272.2	276.5	274.5	277.2	289.6	396.7	250.0	160.3	121.0	103.1	146.6
Real [600]	305.4	295.3	295.2	286.5	273.1	272.8	422.7	244.0	231.7	199.1	175.3	215.5
Real [800]	307.5	298.9	300.5	294.4	284.0	290.8	422.0	272.1	225.4	160.5	137.7	173.7

Figure 4.12: Real-Fake FID matrix of AC-GAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The poor generative capabilities are reflected in the real-fake FID matrices, which present high scores even within same concentrations and no trend of increasing FID scores from the diagonal for both the C-GAN and AC-GAN models.

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	0.0	0.6	1.7	1.4	2.3	5.5	9.0	118.3	104.5	121.4	216.0	165.8	144.9
Fake [3]	0.7	0.9	1.0	1.2	2.7	5.5	8.5	118.1	102.4	124.9	217.5	165.5	144.4
Fake [5]	1.2	1.5	1.5	0.6	1.9	4.1	7.6	119.8	105.8	124.5	216.6	165.8	144.6
Fake [7]	1.2	0.7	1.1	0.2	0.2	1.9	4.8	114.3	106.8	122.2	211.4	162.9	142.0
Fake [9]	2.1	1.3	1.4	0.5	0.0	1.2	3.5	110.8	105.4	118.0	208.4	161.4	140.7
Fake [15]	5.4	3.8	3.2	2.5	1.4	0.4	0.8	102.3	105.5	119.4	203.2	158.0	137.5
Fake [20]	8.9	7.1	6.3	5.3	3.9	1.6	0.1	95.2	104.0	117.1	196.4	153.5	133.4
Fake [50]	123.1	120.9	118.1	119.9	116.6	111.6	99.4	0.3	123.7	137.4	167.1	166.9	155.6
Fake [100]	106.4	107.1	101.8	105.6	107.6	110.5	104.3	118.8	0.3	100.9	222.7	168.0	149.9
Fake [200]	125.7	125.6	123.4	123.4	120.9	120.3	115.7	134.5	101.3	0.1	102.9	88.8	84.6
Fake [400]	216.8	215.0	214.2	212.5	208.6	203.2	195.6	166.9	221.4	103.3	0.0	34.3	49.3
Fake [600]	166.1	164.8	164.1	163.2	161.5	158.2	152.8	165.3	167.9	89.4	34.7	0.0	10.7
Fake [800]	145.8	144.6	144.0	143.2	141.7	138.4	133.8	153.4	149.6	85.4	49.2	10.8	0.1

Figure 4.13: Fake-Fake FID matrix of C-GAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	0.2	7.3	17.5	32.3	46.9	91.2	136.1	125.4	145.8	189.2	255.8	254.3	265.7
Fake [3]	6.8	0.3	4.8	16.5	29.5	72.0	113.5	128.7	136.8	190.1	257.9	253.6	265.8
Fake [5]	16.6	4.3	0.4	5.6	15.7	56.1	95.4	142.9	139.7	203.2	271.5	264.7	275.0
Fake [7]	32.0	16.2	5.5	0.3	4.3	39.6	72.8	165.9	141.9	220.0	285.4	275.0	283.0
Fake [9]	46.5	29.0	15.4	4.2	0.2	29.0	59.4	176.5	143.0	232.7	291.2	279.0	285.0
Fake [15]	91.3	72.0	56.5	40.5	29.6	0.4	18.5	225.4	155.5	266.1	305.2	290.2	291.0
Fake [20]	134.9	112.2	94.9	73.0	59.4	18.9	0.1	279.3	167.2	297.9	337.2	317.5	319.3
Fake [50]	127.4	130.6	145.8	166.4	177.6	224.7	280.4	0.1	156.7	236.7	317.3	320.3	327.5
Fake [100]	145.6	136.6	140.4	141.4	142.5	154.5	166.2	156.9	0.0	183.2	262.3	248.8	270.5
Fake [200]	189.7	190.2	205.1	219.7	231.9	265.4	297.2	236.1	182.9	0.1	95.4	102.1	135.6
Fake [400]	255.8	257.4	273.0	285.3	290.4	305.0	336.7	316.6	261.9	95.3	0.0	16.0	32.9
Fake [600]	254.2	253.0	265.9	274.9	278.3	290.3	317.2	319.5	248.9	101.7	16.1	0.0	21.4
Fake [800]	265.6	265.3	276.2	283.0	284.4	291.0	318.9	327.0	270.7	135.5	33.1	21.3	0.0

Figure 4.14: Fake-Fake FID matrix of AC-GAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The fake-fake FID matrices with such low FID scores within the same concentration reveal the mode collapse GAN models are suffering. As a positive remark, given the FID scores with different concentrations both conditioning methods seem to be working: the mode collapse is different for each concentration.

C-LSGAN/AC-LSGAN

The Least Squares GAN models suffered from the same problems of non-convergence and mode collapse. Again, hyperparameter tuning and the noise and label smoothing methods were of no help. The C-LSGAN generated images presented a slightly wider set of features from real images, being an improvement from the previous GAN models. Conversely, the AC-LSGAN model generated completely noisy images.

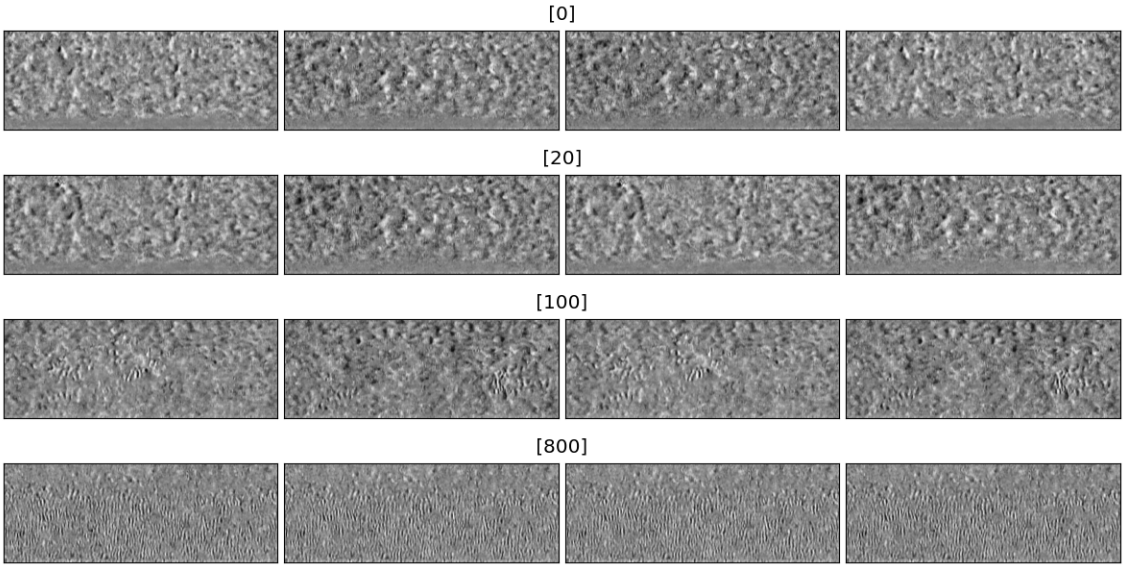

Figure 4.15: C-LSGAN generated images of in vitro meningitis

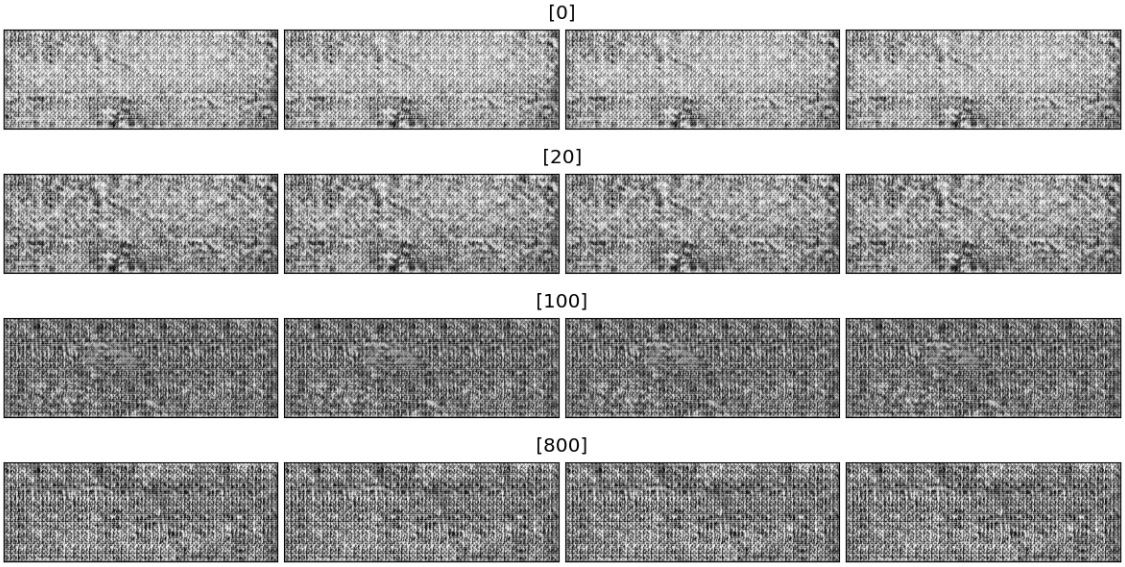

Figure 4.16: AC-LSGAN generated images of in vitro meningitis

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	339.3	341.4	345.1	340.6	336.4	353.7	353.1	303.5	294.9	329.9	339.8	325.8
Real [3]	284.0	286.3	290.2	285.2	283.9	298.1	292.5	247.1	233.3	292.4	308.7	295.8
Real [5]	436.4	437.0	442.0	436.8	435.0	450.9	466.9	408.6	423.2	478.0	489.5	471.3
Real [7]	308.2	309.8	313.7	309.1	309.0	322.7	323.3	281.1	268.5	316.2	331.0	317.9
Real [15]	310.2	312.1	316.5	311.4	309.1	321.2	324.2	274.0	260.5	311.1	331.5	323.3
Real [20]	358.0	359.2	364.0	358.4	354.8	370.3	374.4	319.5	330.2	398.8	412.3	395.1
Real [50]	335.2	337.6	340.6	335.8	331.1	344.2	340.3	295.9	271.1	288.1	306.1	299.0
Real [100]	305.2	307.4	311.1	306.5	305.3	318.0	320.3	265.7	242.9	273.6	298.4	293.6
Real [200]	359.1	362.2	364.3	360.5	358.1	369.5	361.1	319.0	267.7	226.3	244.8	248.5
Real [400]	360.8	364.3	366.1	362.7	360.4	371.7	365.2	321.8	263.6	208.7	224.2	229.8
Real [600]	401.0	404.3	406.8	403.1	398.0	410.7	412.5	361.3	327.6	308.2	323.0	320.5
Real [800]	402.3	405.5	407.2	404.3	400.7	413.0	410.3	366.9	316.9	260.5	273.6	276.5

Figure 4.17: Real-Fake FID matrix of C-LSGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	277.5	283.3	287.0	291.7	320.5	348.9	356.1	326.7	348.1	354.6	347.6	340.8
Real [3]	253.8	257.9	260.0	262.8	281.1	305.0	317.2	302.0	324.5	329.9	319.1	312.3
Real [5]	399.0	401.0	402.6	405.8	428.2	444.3	456.2	417.5	429.2	426.0	420.9	419.4
Real [7]	261.1	262.8	263.3	264.8	277.1	293.6	307.6	299.0	318.4	319.9	308.2	302.5
Real [15]	266.2	270.3	272.2	274.7	289.9	311.3	324.8	311.7	329.8	333.2	322.0	315.2
Real [20]	322.8	327.5	330.6	334.9	363.2	389.5	396.2	369.4	390.7	392.2	384.2	379.7
Real [50]	243.6	249.2	252.2	255.4	274.9	302.1	309.0	294.6	316.5	324.1	314.6	306.1
Real [100]	238.3	241.8	243.2	245.0	257.8	276.8	287.7	273.6	289.7	294.2	286.2	281.1
Real [200]	220.8	226.6	228.9	230.9	244.0	269.9	275.6	257.2	275.8	288.0	279.6	271.4
Real [400]	211.3	216.8	218.7	220.3	230.9	255.4	262.6	243.8	261.1	274.1	265.6	257.4
Real [600]	273.3	278.9	282.0	285.5	307.6	333.4	338.1	308.4	327.9	339.3	332.1	326.5
Real [800]	236.1	241.6	244.2	246.9	264.7	289.2	293.3	267.1	285.4	298.0	291.1	285.1

Figure 4.18: Real-Fake FID matrix of AC-LSGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The real-fake FID matrices present high scores even within same concentrations and no trend of increasing FID scores from the diagonal for both the C-LSGAN and AC-LSGAN.

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	0.2	2.3	5.1	6.7	9.1	15.3	24.9	53.0	81.8	114.5	259.2	282.9	254.3
Fake [3]	2.0	1.6	2.1	3.6	5.6	12.7	22.9	52.8	80.0	115.9	260.2	284.5	255.3
Fake [5]	4.5	3.0	1.1	1.5	3.7	10.8	21.7	51.8	80.1	117.7	264.6	289.2	259.5
Fake [7]	6.4	3.8	2.2	0.6	1.5	8.3	19.8	48.6	77.3	114.5	262.8	287.5	258.0
Fake [9]	8.2	5.3	3.6	1.3	0.7	6.2	17.7	46.4	75.8	115.2	264.7	290.0	260.0
Fake [15]	14.9	11.8	10.4	8.7	6.0	1.4	9.4	44.2	69.1	112.8	265.5	291.4	260.4
Fake [20]	24.5	21.8	21.1	20.3	17.8	9.8	0.4	43.7	68.8	114.2	269.5	296.3	265.9
Fake [50]	52.3	49.2	50.9	49.2	47.6	43.7	43.8	0.1	56.8	86.7	254.5	280.0	248.6
Fake [100]	79.1	76.8	78.1	76.7	74.3	70.0	68.4	55.3	0.1	71.3	233.6	262.0	234.0
Fake [200]	113.2	112.1	117.2	115.9	113.8	112.4	113.9	86.0	71.0	0.3	133.4	160.7	139.9
Fake [400]	261.1	261.6	267.8	268.4	264.6	269.6	271.8	256.4	237.6	133.1	0.2	26.3	36.5
Fake [600]	282.4	283.3	289.5	290.5	286.3	293.5	295.8	278.9	263.7	157.5	26.2	0.0	18.3
Fake [800]	253.9	254.6	259.7	261.5	256.8	264.2	265.8	248.5	236.3	138.2	37.7	19.0	0.1

Figure 4.19: Fake-Fake FID matrix of C-LSGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	0.0	1.1	2.8	5.4	8.8	30.7	61.8	74.1	102.9	128.7	121.8	109.4	100.0
Fake [3]	1.1	0.1	0.6	2.2	4.9	24.9	54.1	70.3	98.1	123.3	116.8	103.8	94.9
Fake [5]	2.9	0.6	0.1	0.7	2.5	19.9	47.4	65.3	93.9	118.8	112.8	99.3	90.7
Fake [7]	5.2	2.1	0.7	0.1	0.8	15.2	41.3	60.4	89.7	113.9	108.4	94.5	86.3
Fake [9]	8.7	4.8	2.5	0.7	0.1	10.4	34.4	55.6	86.7	109.6	104.7	90.1	82.1
Fake [15]	30.6	24.8	20.1	15.0	10.2	0.1	11.0	43.4	82.4	99.6	97.6	79.2	71.5
Fake [20]	62.0	54.1	47.9	41.0	34.3	10.9	0.1	49.7	91.0	101.7	101.8	80.1	74.7
Fake [50]	74.0	70.1	65.5	60.3	55.3	43.5	49.6	0.2	66.5	97.1	93.9	80.0	79.6
Fake [100]	103.0	98.1	93.6	89.7	86.7	82.1	90.5	66.2	0.0	13.2	16.5	21.2	29.1
Fake [200]	128.6	123.2	118.3	113.6	109.4	99.2	101.1	96.4	13.1	0.0	6.7	16.4	25.2
Fake [400]	121.9	116.9	112.4	108.3	104.6	97.4	101.2	93.5	16.5	6.7	0.0	7.9	16.7
Fake [600]	109.5	103.8	98.9	94.4	89.9	78.9	79.7	79.6	21.2	16.5	7.9	0.0	4.3
Fake [800]	100.2	95.1	90.5	86.3	82.1	71.5	74.4	79.3	29.2	25.3	16.7	4.3	0.0

Figure 4.20: Fake-Fake FID matrix of AC-LSGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The low FID scores within the same concentration in the fake-fake FID matrices reveal the mode collapse both the C-LSGAN and AC-LSGAN are suffering. Again, the FID scores with different concentrations suggest the mode collapse is different for each concentration meaning the conditioning methods are effective.

C-WGAN/AC-WGAN

The Wasserstein loss finally resulted in a stable and convergent C-WGAN and AC-WGAN models with no sign of mode collapse. Even without hyperparameter tuning, the WGAN models were able to generate good visual quality and diverse images containing features with high resemblance to real ones.

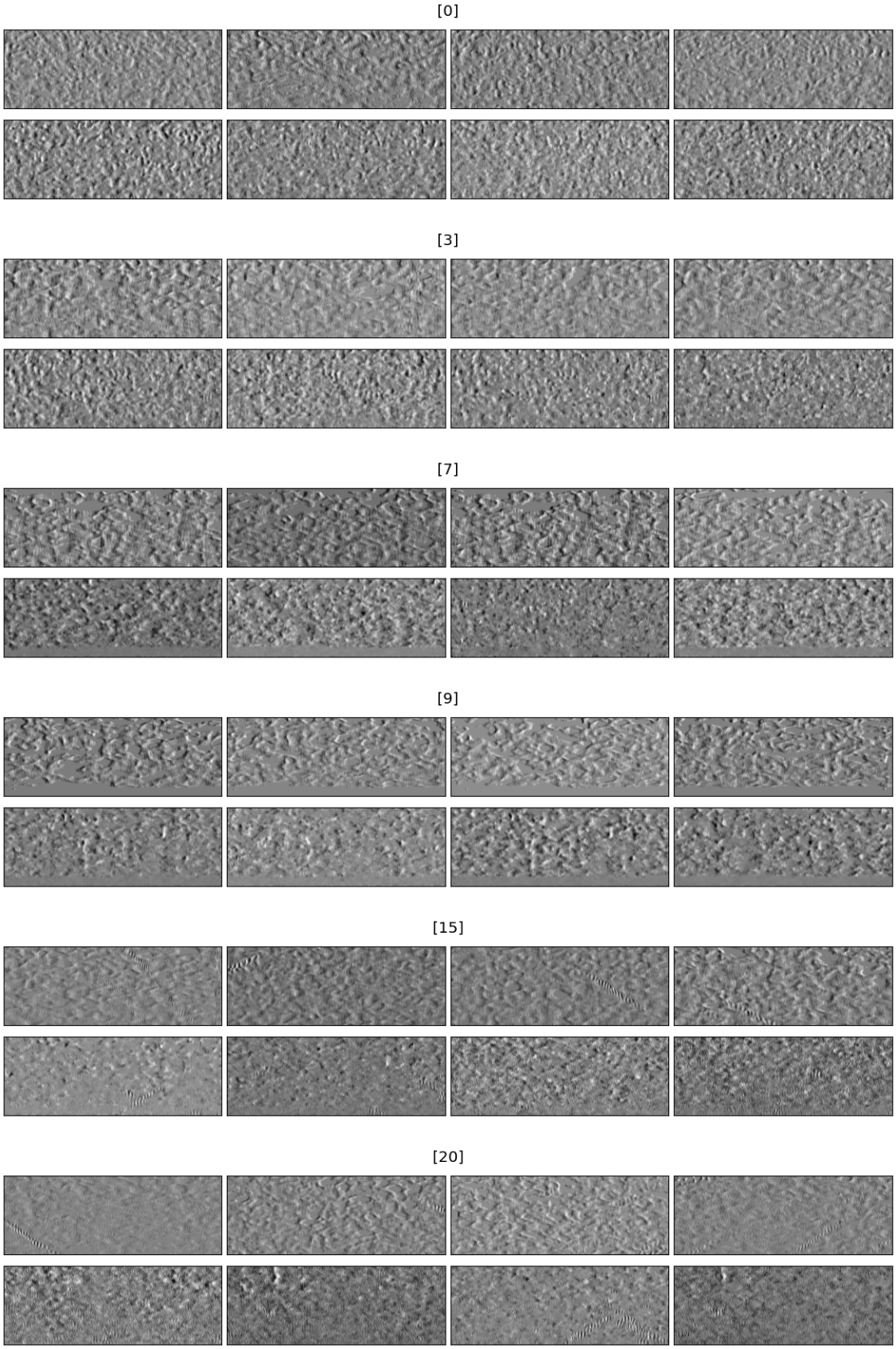

Figure 4.21: Real (top) and C-WGAN generated (bottom) images of in vitro meningitis

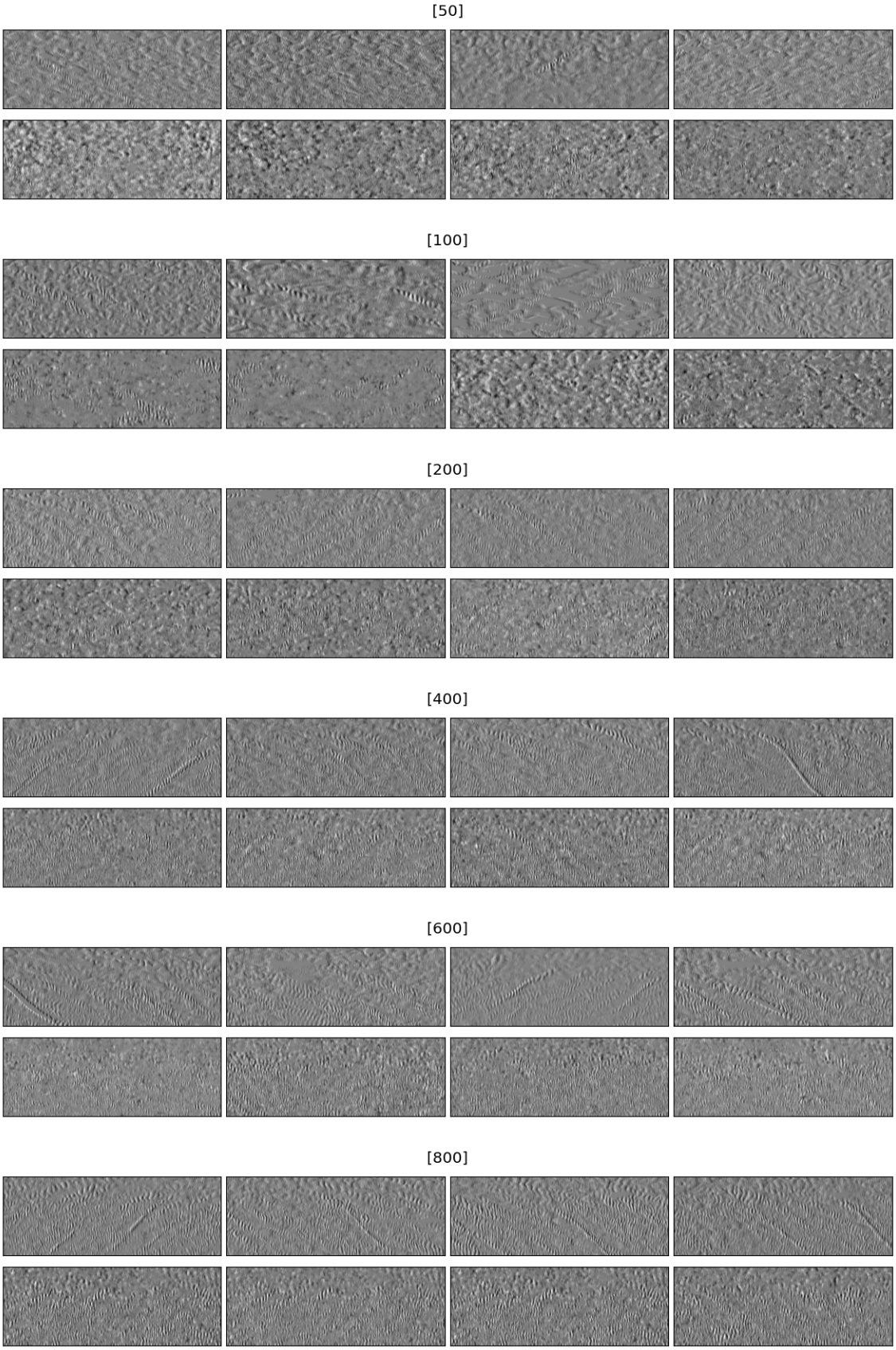

Figure 4.21: Real (top) and C-WGAN generated (bottom) images of in vitro meningitis

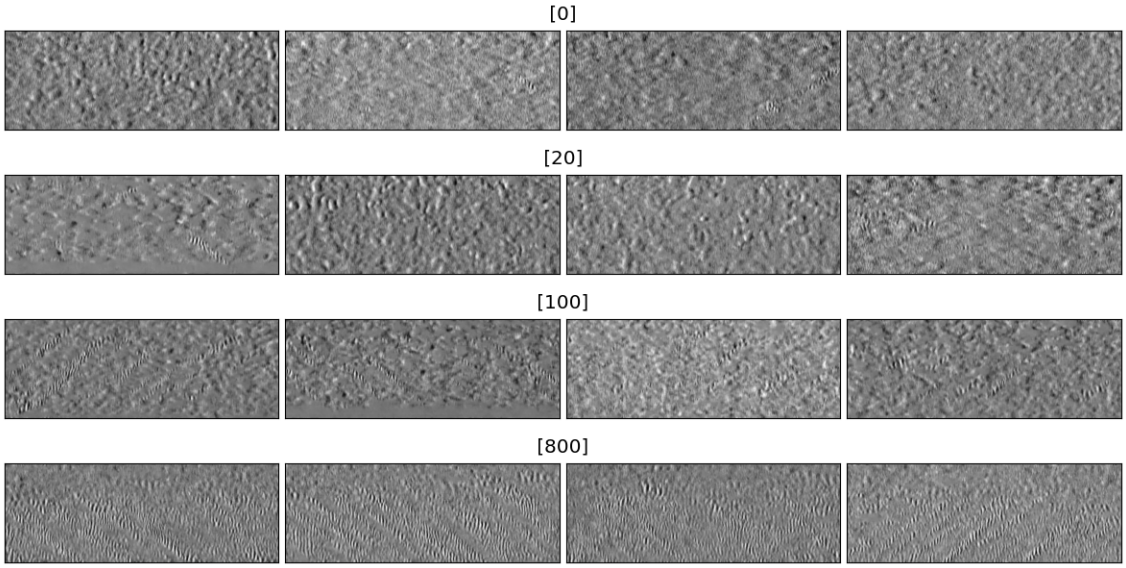

Figure 4.22: AC-WGAN generated images of in vitro meningitis

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	33.5	33.0	40.5	63.1	90.0	97.2	74.1	54.2	108.7	152.6	150.8	143.6
Real [3]	73.1	66.4	68.5	80.4	81.9	82.3	73.3	76.4	108.3	151.8	160.1	156.7
Real [5]	231.5	212.5	198.4	181.3	218.4	266.0	245.6	223.9	332.4	367.6	354.6	347.6
Real [7]	150.2	139.2	134.0	128.1	126.1	140.8	139.7	146.5	186.3	216.0	218.4	216.7
Real [15]	94.7	86.3	85.2	91.0	79.1	82.6	71.9	74.3	118.9	165.3	170.4	167.8
Real [20]	113.6	101.5	96.0	97.0	124.3	154.7	130.1	107.9	201.7	248.7	243.8	237.1
Real [50]	66.1	66.8	75.0	95.3	80.5	65.7	52.1	50.2	63.9	106.1	109.3	107.6
Real [100]	106.9	100.6	99.6	104.3	82.9	72.3	63.2	75.0	102.4	137.6	140.0	138.7
Real [200]	106.2	113.5	125.7	152.0	122.5	84.5	75.3	81.6	31.5	44.0	41.6	42.9
Real [400]	128.9	137.0	148.0	172.9	145.3	103.9	96.0	102.7	45.5	42.4	38.0	40.1
Real [600]	110.2	115.9	125.1	147.6	148.8	139.0	104.1	82.7	105.3	121.0	102.3	97.3
Real [800]	148.3	156.5	166.4	187.4	169.5	145.7	125.1	117.6	97.3	90.7	71.3	70.1

Figure 4.23: Real-Fake FID matrix of C-WGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Real [0]	97.6	94.5	96.1	89.4	92.0	94.1	109.7	88.0	110.9	200.9	173.3	155.9
Real [3]	82.1	81.5	83.6	83.2	89.6	92.9	115.3	96.1	109.3	188.3	178.2	164.9
Real [5]	249.7	241.2	245.0	237.6	230.7	230.8	222.0	227.1	319.5	407.5	371.8	355.3
Real [7]	138.5	136.6	139.6	140.5	143.8	148.7	164.9	159.2	191.0	250.6	240.8	229.7
Real [15]	95.6	92.8	94.1	93.4	97.7	100.6	115.3	92.3	112.6	200.4	187.1	174.1
Real [20]	147.1	140.9	142.8	137.0	133.3	133.7	136.8	124.7	186.7	291.3	261.7	244.7
Real [50]	89.6	87.9	87.8	85.2	93.6	97.5	117.3	84.7	69.1	148.4	131.4	119.0
Real [100]	97.5	96.1	97.0	94.8	99.9	100.4	114.8	92.2	100.3	171.3	158.4	146.0
Real [200]	135.5	135.6	135.6	132.7	142.1	145.7	163.6	122.1	56.2	88.2	68.5	59.4
Real [400]	153.2	153.7	153.7	151.0	160.3	162.6	179.4	140.0	70.5	85.4	65.3	55.7
Real [600]	168.4	162.4	162.3	152.9	151.2	150.7	148.5	117.0	106.9	171.5	124.8	107.4
Real [800]	188.8	185.9	185.5	178.5	179.6	180.5	183.4	151.5	115.0	141.7	99.6	86.1

Figure 4.24: Real-Fake FID matrix of AC-WGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The real-fake FID matrices show higher but similar scores to the reference matrix, proving good visual quality of the generated images, and, in general, increasing FID scores from the diagonal, indicating proper conditioning. However, in some cases the lowest FID score is not with the same concentration but with a different one, usually close. This would be expected for test concentrations never seen during training that the model is trying to infer, but it also occurs with training concentrations.

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	3.3	7.4	19.9	54.6	78.7	91.0	102.4	70.9	47.7	104.8	147.6	145.3	134.9
Fake [3]	7.3	5.2	11.4	38.3	60.3	80.9	100.8	68.8	44.9	112.1	157.1	154.2	143.8
Fake [5]	20.4	11.3	8.3	21.5	39.5	69.1	100.9	70.2	47.2	124.8	171.5	168.3	158.3
Fake [7]	49.8	33.1	19.2	9.0	16.6	53.5	100.9	76.4	58.5	144.3	190.3	186.7	178.3
Fake [9]	76.0	56.8	38.5	15.4	9.8	43.0	100.8	80.1	66.5	158.4	204.2	200.2	193.2
Fake [15]	91.9	79.5	69.9	57.5	46.8	9.2	35.5	35.0	48.0	107.4	152.7	154.9	153.8
Fake [20]	101.6	95.9	96.4	101.9	99.7	37.2	6.5	20.5	55.0	55.9	94.0	103.3	106.5
Fake [50]	72.8	69.6	70.7	82.4	83.0	37.7	23.1	4.7	21.5	60.0	105.2	107.6	106.2
Fake [100]	45.8	44.6	46.4	63.4	69.2	48.5	57.0	20.6	2.8	76.4	128.1	125.2	119.1
Fake [200]	109.5	113.7	127.2	153.5	165.8	116.9	56.7	63.3	84.3	3.4	18.5	28.2	32.6
Fake [400]	148.9	154.9	169.6	194.8	207.5	158.1	91.3	104.7	131.0	20.9	3.1	8.0	14.0
Fake [600]	145.3	151.7	165.8	190.1	202.2	157.7	99.8	105.9	126.4	29.8	7.2	2.8	5.1
Fake [800]	134.8	141.7	156.2	182.1	195.8	157.1	104.3	105.3	121.0	34.6	13.3	4.9	2.5

Figure 4.25: Fake-Fake FID matrix of C-WGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [3]	Fake [5]	Fake [7]	Fake [9]	Fake [15]	Fake [20]	Fake [50]	Fake [100]	Fake [200]	Fake [400]	Fake [600]	Fake [800]
Fake [0]	7.4	11.4	16.7	14.2	12.8	22.0	27.3	64.0	64.7	84.7	154.0	151.0	141.8
Fake [3]	12.1	13.7	17.0	14.7	14.0	20.6	25.6	60.6	63.9	88.2	159.8	153.0	143.8
Fake [5]	13.4	14.2	16.5	13.8	13.3	19.4	23.9	56.5	59.2	84.1	158.1	151.3	141.1
Fake [7]	13.6	14.7	18.2	15.4	14.0	19.6	23.8	58.2	63.3	89.5	160.7	155.1	146.4
Fake [9]	12.7	12.9	15.4	12.6	9.9	13.2	14.6	43.6	49.2	86.9	166.1	154.8	144.2
Fake [15]	21.7	19.7	20.2	17.5	13.4	12.3	11.7	34.2	43.0	90.8	176.6	161.0	149.7
Fake [20]	28.8	26.3	26.9	23.0	17.0	12.1	7.1	21.5	35.0	98.1	189.1	169.7	157.7
Fake [50]	63.0	56.9	54.8	51.4	44.2	32.9	22.4	8.4	18.0	104.5	212.3	183.5	169.7
Fake [100]	66.7	63.2	63.3	58.4	52.5	45.5	36.7	20.1	3.3	68.9	181.4	153.7	139.9
Fake [200]	82.6	88.2	91.8	81.4	83.0	91.8	93.8	106.5	66.5	5.7	62.5	56.5	50.4
Fake [400]	151.4	162.6	168.2	157.1	162.2	177.8	185.3	213.0	179.2	58.9	3.8	19.8	26.6
Fake [600]	148.0	155.1	157.7	148.2	151.5	162.2	166.7	185.2	152.9	54.4	19.2	4.7	6.6
Fake [800]	140.5	147.0	149.0	139.5	142.4	151.9	155.2	172.1	139.8	49.2	26.6	6.5	2.8

Figure 4.26: Fake-Fake FID matrix of AC-WGAN on in vitro meningitis dataset. The lowest FID score for each concentration row is highlighted in blue

The fake-fake FID matrices show both conditioning methods are working properly: the lowest scores are in the diagonal and increasing away from it. In some cases, this trend is not constant resulting in lower FID scores when there is a higher concentration difference. Additionally, the not too low FID scores within same concentrations are a sign of diversity in the generation and no mode collapse.

In vitro peritoneal dialysis

Firstly, the FID matrix of in vitro peritoneal dialysis images was computed to establish a reference.

	Real [0]	Real [12]	Real [35]	Real [60]	Real [80]	Real [105]	Real [113]	Real [160]	Real [218]	Real [450]	Real [580]	Real [1000]
Real [0]	4.2	47.9	27.1	61.7	96.7	115.3	130.6	103.8	83.3	84.0	104.1	97.9
Real [12]	40.0	6.0	15.8	16.6	27.0	43.6	54.2	50.4	42.4	61.3	65.2	74.7
Real [35]	25.2	15.5	5.1	23.5	48.2	66.1	78.8	61.8	45.8	57.5	68.4	71.2
Real [60]	54.0	16.1	22.4	6.0	26.1	42.8	50.9	45.1	35.4	56.0	65.3	78.3
Real [80]	86.6	25.9	48.9	24.9	3.1	10.5	13.4	25.6	29.6	57.7	50.2	73.4
Real [105]	105.5	42.2	66.9	40.8	10.1	5.7	10.4	26.7	35.6	66.0	55.2	80.6
Real [113]	122.2	53.7	80.7	48.6	12.8	10.2	2.9	39.0	48.8	81.5	70.3	96.3
Real [160]	99.3	50.7	66.4	46.0	27.6	29.3	41.1	5.3	11.0	30.2	22.8	47.2
Real [218]	79.7	42.6	49.7	36.2	29.8	36.1	48.5	9.4	3.7	18.9	19.8	38.5
Real [450]	77.6	57.3	55.4	54.0	57.0	66.3	80.4	29.0	16.0	5.4	16.0	21.6
Real [580]	99.0	61.2	69.3	65.3	50.8	57.6	70.9	22.8	19.8	18.4	3.0	11.8
Real [1000]	95.0	71.3	71.9	78.9	74.3	82.7	96.5	46.2	36.2	23.6	11.7	2.9

Figure 4.27: Reference FID matrix on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The reference FID matrix presents the lowest FID scores within the same concentrations and there is a trend of increasing FID scores the higher the concentration difference, except in some cases. This could mean the presence of other sources of variation and not only the number of traces.

C-GAN/AC-GAN

The Conditional and Auxiliary Classifier GAN models using the minimax loss function on the in vitro peritoneal dialysis dataset presented similar problems as on the in vitro

meningitis dataset. Once again, mode collapse could not be avoided, but the generated images were not completely noisy and presented certain patterns and features similar to real data. The C-GAN model is able to capture and generate a wider set of features compared to the AC-GAN model.

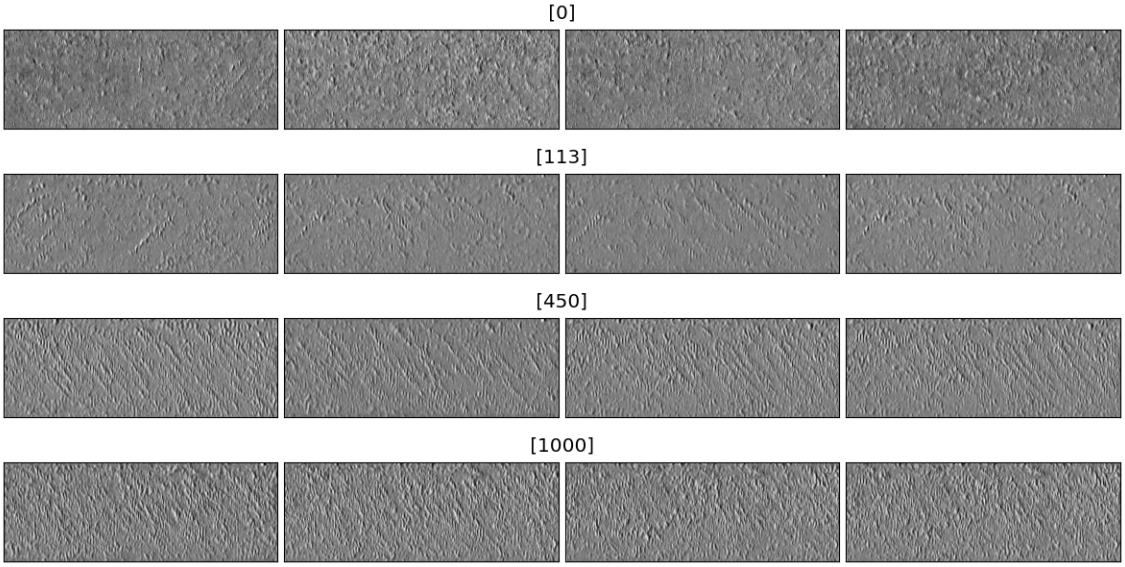

Figure 4.28: C-GAN generated images of in vitro peritoneal dialysis

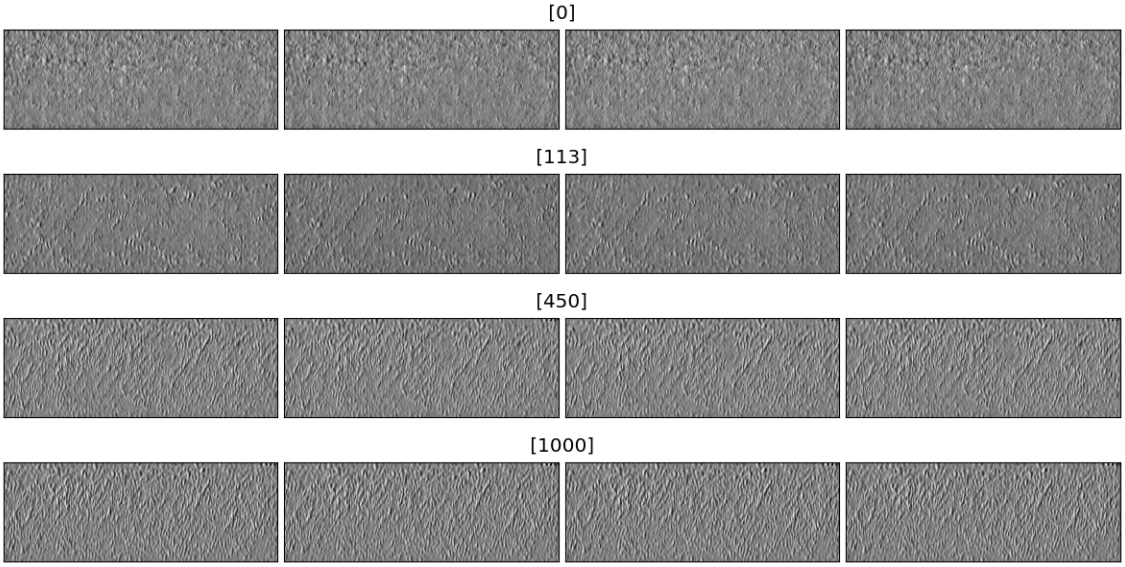

Figure 4.29: AC-GAN generated images of in vitro peritoneal dialysis

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	149.4	155.1	187.5	210.3	214.7	214.1	212.1	185.8	160.8	119.6	114.8	97.4
Real [12]	153.5	134.8	120.1	127.8	140.1	136.9	136.3	115.0	114.4	115.0	112.2	119.9
Real [35]	147.0	131.1	120.8	127.8	136.2	134.7	134.0	115.3	107.9	105.8	101.6	109.1
Real [60]	166.2	148.9	131.8	134.4	145.8	142.9	142.8	123.7	123.0	130.1	124.4	133.7
Real [80]	177.6	154.9	128.1	126.1	135.2	131.4	130.8	115.3	111.7	130.4	124.6	141.2
Real [105]	185.9	158.0	122.0	112.3	119.0	113.1	112.9	103.2	108.4	142.5	137.4	161.1
Real [113]	198.5	168.3	126.5	115.3	121.5	115.5	115.7	107.8	122.2	162.9	157.8	182.1
Real [160]	187.0	165.5	140.8	131.5	132.9	129.7	128.8	116.7	90.2	114.0	104.4	126.9
Real [218]	178.6	159.6	138.9	133.8	137.8	134.6	133.5	117.1	92.4	107.1	96.7	113.7
Real [450]	177.5	162.8	151.4	150.7	157.3	154.4	153.3	135.2	100.6	94.6	82.7	93.4
Real [580]	183.7	164.2	143.5	141.3	149.6	145.3	143.5	124.1	89.6	91.1	79.0	94.8
Real [1000]	172.6	155.9	141.2	145.8	157.8	152.9	150.8	126.2	95.0	81.8	68.5	75.5

Figure 4.30: Real-Fake FID matrix of C-GAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	133.2	79.7	55.7	61.4	105.3	119.0	115.4	85.3	120.2	89.7	116.1	110.4
Real [12]	290.6	215.1	137.6	87.0	64.8	82.5	85.9	116.0	115.9	114.3	138.2	127.1
Real [35]	267.0	193.5	119.4	74.2	54.5	67.9	68.5	94.5	100.6	102.0	129.2	118.3
Real [60]	299.5	222.9	149.4	95.0	62.7	81.0	83.6	116.7	123.5	126.3	152.0	140.2
Real [80]	331.0	255.8	170.9	107.1	58.1	77.4	81.5	130.5	120.8	132.8	153.8	142.5
Real [105]	373.9	301.2	205.5	137.9	72.7	92.7	98.2	169.2	144.9	160.0	174.1	167.1
Real [113]	396.9	321.4	223.2	154.7	80.3	101.3	105.8	187.6	164.2	179.9	194.3	189.2
Real [160]	337.9	275.1	185.8	117.7	60.8	85.2	95.2	136.1	106.5	122.0	136.9	127.4
Real [218]	314.2	250.5	165.6	102.5	54.7	77.5	87.0	120.6	96.4	104.2	121.4	109.3
Real [450]	286.2	230.8	149.1	88.6	61.4	82.4	93.2	101.7	80.2	77.2	86.7	79.9
Real [580]	319.0	263.1	171.5	102.6	61.6	87.4	102.6	123.2	90.3	85.7	96.9	88.5
Real [1000]	298.2	248.9	159.2	95.5	74.4	100.1	117.3	122.9	90.7	68.5	77.4	71.9

Figure 4.31: Real-Fake FID matrix of AC-GAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The generative capabilities are still far from good, which reflect in a real-fake FID matrix with high scores, even within same concentrations, and no trend of increasing FID scores from the diagonal. The AC-GAN model presents, in general, lower FID scores in spite of suffering from a more severe mode collapse.

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	0.0	11.4	55.6	87.8	87.0	89.8	87.7	78.4	109.7	120.6	123.5	111.8
Fake [12]	10.5	0.2	30.2	62.0	66.6	67.0	65.7	56.0	94.4	115.6	117.1	110.4
Fake [35]	53.7	31.2	0.4	26.1	47.4	43.5	43.2	33.2	89.6	126.8	126.5	131.0
Fake [60]	86.8	64.5	25.6	0.5	26.5	25.3	27.0	25.2	84.5	137.5	135.9	152.7
Fake [80]	86.1	68.8	46.0	24.1	0.4	7.9	10.7	31.9	74.3	134.6	133.7	152.1
Fake [105]	86.8	68.7	44.0	26.1	7.4	0.8	1.6	27.4	73.9	134.4	133.9	150.5
Fake [113]	86.7	67.9	41.9	25.0	10.9	1.5	0.6	23.9	72.1	132.8	133.1	149.5
Fake [160]	76.4	57.6	31.7	25.2	32.5	26.0	24.2	0.2	56.0	102.6	103.4	119.2
Fake [218]	106.5	93.5	88.6	86.4	75.8	76.4	72.4	57.2	0.4	44.9	48.1	71.7
Fake [450]	117.7	113.1	123.8	136.5	132.6	134.3	129.0	101.0	45.0	0.1	16.3	30.5
Fake [580]	120.8	114.6	124.1	135.6	133.3	135.0	130.7	102.6	48.8	16.0	0.3	23.2
Fake [1000]	110.8	109.6	131.5	155.1	154.1	154.3	149.8	121.5	75.2	30.0	23.5	0.2

Figure 4.32: Fake-Fake FID matrix of C-GAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	0.0	27.5	69.5	142.9	243.0	250.3	247.1	164.8	220.8	183.5	204.9	211.1
Fake [12]	27.3	0.0	39.7	104.5	186.3	189.6	182.3	124.0	186.1	154.5	180.7	180.1
Fake [35]	69.2	40.0	0.0	45.6	107.8	111.8	108.6	73.3	113.7	77.8	97.5	102.2
Fake [60]	142.6	105.2	45.8	0.1	56.9	71.7	77.1	36.8	61.6	48.7	64.6	73.7
Fake [80]	242.6	186.9	108.1	56.5	0.0	17.9	27.5	76.3	82.9	90.8	111.3	107.1
Fake [105]	250.3	190.7	112.3	71.5	17.8	0.1	6.6	76.0	85.0	101.0	123.9	113.6
Fake [113]	246.3	182.6	108.9	76.8	27.6	6.6	0.0	79.7	97.8	114.1	137.8	128.1
Fake [160]	163.8	123.9	73.2	36.7	76.5	75.9	80.0	0.1	40.7	64.3	96.0	96.5
Fake [218]	220.5	186.7	113.8	61.8	83.0	85.3	97.9	40.4	0.0	52.8	71.1	62.4
Fake [450]	183.4	155.3	77.9	48.8	90.7	101.1	113.8	64.2	52.8	0.0	19.5	25.6
Fake [580]	203.9	180.9	97.1	64.9	111.6	124.4	137.9	95.5	71.3	19.4	0.0	19.0
Fake [1000]	210.5	180.5	102.1	74.1	107.2	114.0	128.0	96.2	62.6	25.6	18.9	0.0

Figure 4.33: Fake-Fake FID matrix of AC-GAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The fake-fake FID matrices reveal the mode collapse with such low FID scores within the same concentration. As a positive remark, the conditioning seems to be working given the FID scores with different concentrations: the mode collapse is different for each concentration. Additionally, it can be observed how the C-GAN is able to generate more varied samples given the higher FID scores in the diagonal compared to the AC-GAN, whose fake-fake FID matrix presents extremely low FID scores.

C-LSGAN/AC-LSGAN

The Least Squares GAN models continued to suffer from mode collapse. However, while the C-LSGAN was able to represent a limited set of realistic visual features, the AC-LSGAN could only generate noisy images.

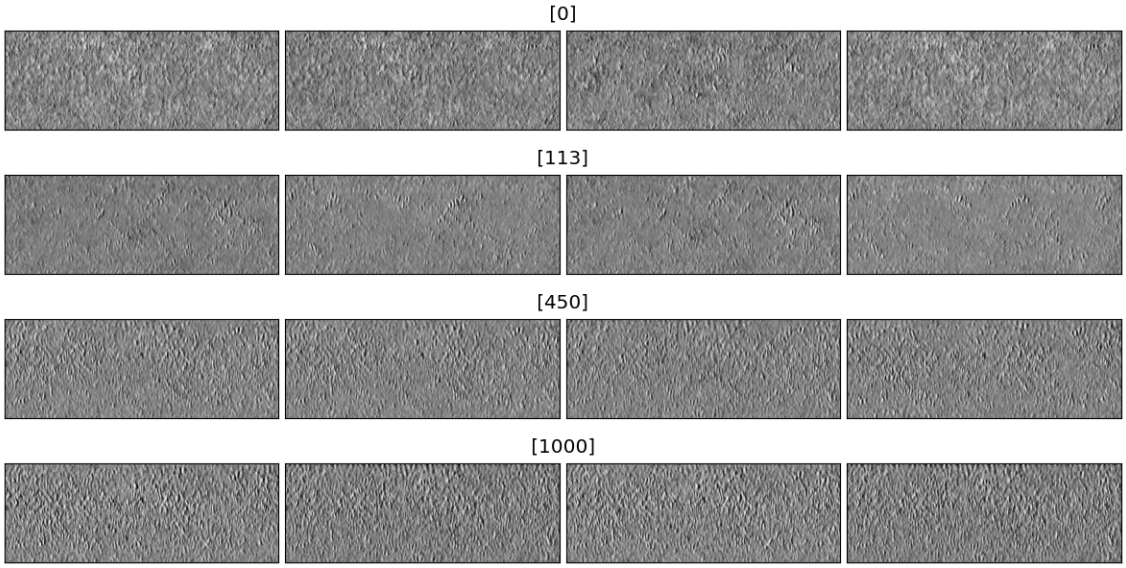

Figure 4.34: C-LSGAN generated images of in vitro peritoneal dialysis

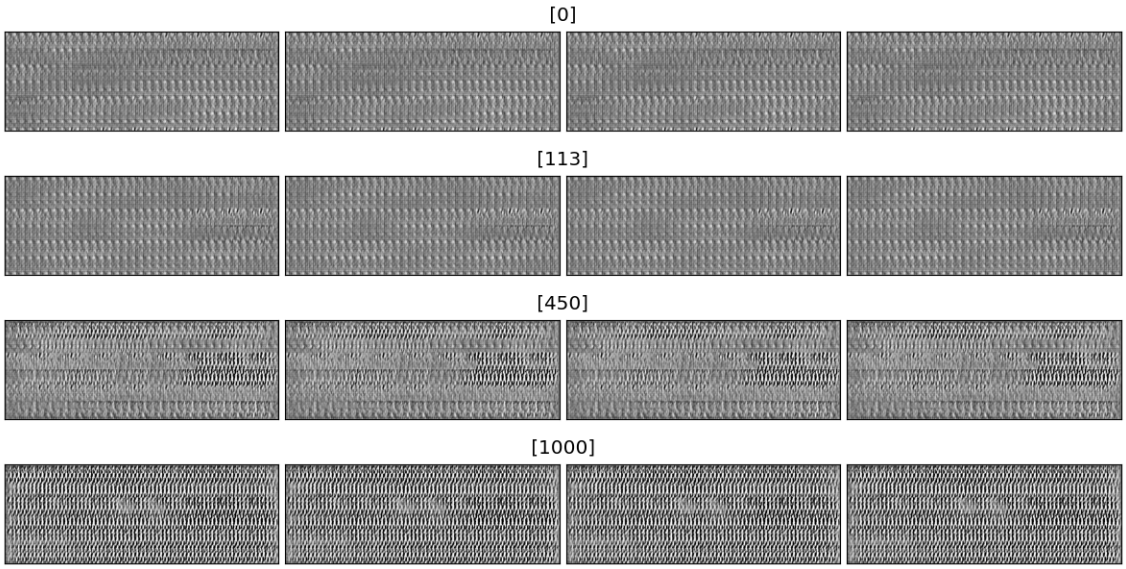

Figure 4.35: AC-LSGAN generated images of in vitro peritoneal dialysis

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	139.2	131.4	140.7	127.1	148.9	148.1	148.1	122.2	134.6	117.9	119.9	127.8
Real [12]	158.9	141.8	109.7	116.1	93.9	93.5	95.9	114.5	108.9	173.5	177.3	222.1
Real [35]	150.8	135.1	103.5	107.5	87.5	87.8	90.1	105.3	97.7	154.1	157.7	202.2
Real [60]	177.5	160.5	122.8	130.1	102.1	103.3	106.3	128.0	116.2	183.4	183.7	228.9
Real [80]	185.5	167.5	121.9	133.8	95.0	96.3	99.5	130.0	112.3	198.6	202.0	250.6
Real [105]	200.2	180.5	129.7	145.2	96.8	98.3	102.2	138.3	124.0	222.9	231.0	283.7
Real [113]	216.2	196.4	144.0	161.9	107.9	110.0	114.0	154.8	141.4	245.9	255.2	309.8
Real [160]	192.6	173.8	123.3	136.2	98.2	99.7	102.8	129.4	99.3	190.3	193.5	237.5
Real [218]	185.1	166.5	120.3	130.8	98.7	99.4	101.7	123.8	95.1	174.9	175.5	216.9
Real [450]	177.7	160.1	119.5	126.4	109.3	109.7	111.1	119.1	93.1	152.0	152.0	183.8
Real [580]	183.3	163.7	119.4	128.6	105.3	104.1	105.0	121.8	94.3	169.5	170.4	206.9
Real [1000]	169.6	150.5	116.1	122.6	113.3	110.7	110.8	116.3	95.0	156.4	157.9	182.7

Figure 4.36: Real-Fake FID matrix of C-LSGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	273.5	292.2	304.1	310.3	309.6	307.2	312.3	337.5	354.5	294.9	312.3	287.8
Real [12]	343.3	358.5	366.3	369.7	371.3	367.6	372.5	388.0	398.4	326.9	328.2	325.5
Real [35]	328.6	344.4	352.4	356.1	356.9	353.0	357.4	371.9	383.7	314.0	317.6	309.4
Real [60]	349.5	365.3	373.0	378.4	379.3	374.2	378.4	391.3	399.7	324.9	326.4	324.8
Real [80]	363.7	377.3	383.5	387.7	389.4	384.4	388.9	400.9	409.3	335.7	334.3	338.9
Real [105]	381.4	393.0	398.3	401.3	404.3	400.6	405.9	421.9	430.6	358.0	352.8	363.3
Real [113]	396.1	407.3	412.1	414.7	417.6	413.6	418.9	434.6	444.1	371.5	365.1	377.4
Real [160]	364.0	376.0	382.4	388.4	390.9	386.3	391.2	403.5	409.8	335.8	332.5	338.4
Real [218]	349.1	362.1	369.0	375.7	377.7	372.6	377.0	388.5	394.7	320.7	319.5	320.2
Real [450]	321.6	334.9	342.7	350.3	351.8	346.5	350.4	360.3	366.8	291.9	295.4	295.9
Real [580]	349.9	362.8	370.3	376.7	379.0	373.6	377.9	387.2	390.8	315.8	312.9	320.6
Real [1000]	342.9	356.0	364.2	370.0	372.2	366.1	370.2	378.6	381.8	307.5	303.6	313.2

Figure 4.37: Real-Fake FID matrix of AC-LSGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The C-LSGAN real-fake FID matrix presents high scores even within same concentrations and no trend of increasing FID scores from the diagonal. The same occurs with the AC-LSGAN FID matrix, but with much higher FID scores.

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	0.2	10.4	43.4	40.5	75.1	74.2	71.7	65.8	98.0	110.4	137.0	149.9
Fake [12]	10.8	0.3	33.6	32.1	63.8	62.6	60.3	54.5	85.1	105.7	131.8	148.8
Fake [35]	43.5	34.8	0.1	19.7	32.8	30.8	30.6	37.7	53.6	110.8	132.5	165.5
Fake [60]	39.1	31.4	20.3	0.1	36.7	37.3	36.0	29.7	56.0	91.7	114.0	144.5
Fake [80]	75.1	64.8	34.9	35.7	0.0	6.9	9.4	41.9	54.5	128.0	147.8	192.0
Fake [105]	72.5	62.1	32.7	35.8	6.9	0.8	1.9	39.8	52.0	124.4	144.5	187.6
Fake [113]	70.4	60.2	31.7	34.5	9.5	1.8	0.2	36.3	49.1	117.9	137.7	180.1
Fake [160]	63.5	52.7	39.1	28.4	42.6	41.7	38.0	0.6	43.5	77.6	98.0	129.2
Fake [218]	97.8	85.2	55.2	55.9	54.7	52.6	50.3	42.9	0.1	76.2	86.8	126.5
Fake [450]	116.8	110.4	117.2	97.8	132.8	129.7	124.0	84.3	81.0	0.1	28.0	59.4
Fake [580]	131.4	124.9	129.0	109.9	143.1	140.9	135.3	96.1	84.8	26.3	0.3	51.9
Fake [1000]	153.1	149.9	168.3	147.8	193.6	190.5	184.0	133.7	128.3	59.3	52.4	0.3

Figure 4.38: Fake-Fake FID matrix of C-LSGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	0.0	5.1	15.3	25.5	31.9	50.6	64.1	155.2	197.8	179.1	225.2	213.1
Fake [12]	5.1	0.0	5.2	14.2	21.1	38.9	51.3	139.6	184.4	173.2	223.4	217.8
Fake [35]	15.3	5.1	0.0	5.7	12.4	27.8	38.9	123.3	169.4	166.7	220.4	218.8
Fake [60]	25.5	14.0	5.6	0.0	3.8	17.7	28.5	112.3	161.2	163.6	217.5	217.2
Fake [80]	31.9	21.0	12.5	3.8	0.0	8.3	17.4	95.0	141.6	148.5	205.3	207.2
Fake [105]	50.6	38.8	27.7	17.7	8.3	0.0	2.3	59.0	99.8	115.0	170.9	182.5
Fake [113]	63.9	51.0	38.7	28.5	17.4	2.3	0.0	43.3	81.8	101.9	159.0	176.4
Fake [160]	155.2	139.6	123.3	112.6	95.1	59.0	43.2	0.0	17.9	53.7	114.3	183.5
Fake [218]	197.6	184.2	169.2	161.5	141.6	99.5	81.6	17.9	0.0	40.8	94.9	190.5
Fake [450]	179.2	173.3	166.7	164.0	148.7	115.0	101.9	53.8	40.8	0.0	30.0	105.5
Fake [580]	225.2	223.5	220.4	217.9	205.5	170.9	159.0	114.3	95.0	30.1	0.0	78.8
Fake [1000]	213.0	217.7	218.8	217.4	207.3	182.5	176.4	183.3	190.6	105.5	78.8	0.0

Figure 4.39: Fake-Fake FID matrix of AC-LSGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The fake-fake FID matrices confirm the existence of mode collapse for both the C-LSGAN and AC-LSGAN models, with very low FID scores within the same concentrations. At the same time, it shows a proper conditioning capable of generating distinct images for different concentration values.

C-WGAN/AC-WGAN

Once again, the Wasserstein GAN models avoided the mode collapse issue and achieved the best quality images in terms of visual features and diversity. There are no visual differences between the C-WGAN and AC-WGAN models generated images.

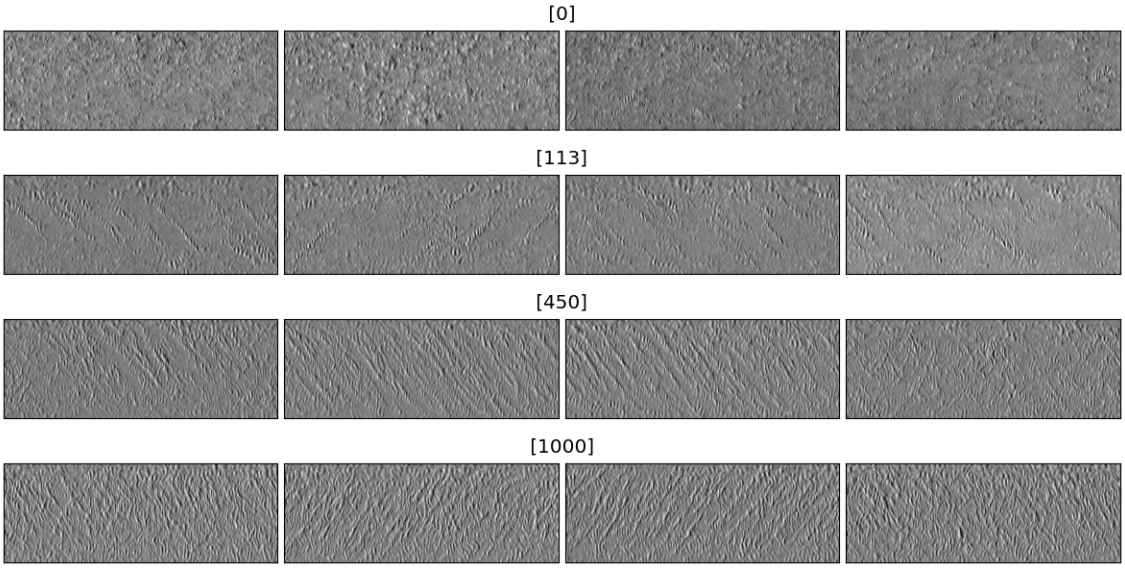

Figure 4.40: C-WGAN generated images of in vitro peritoneal dialysis

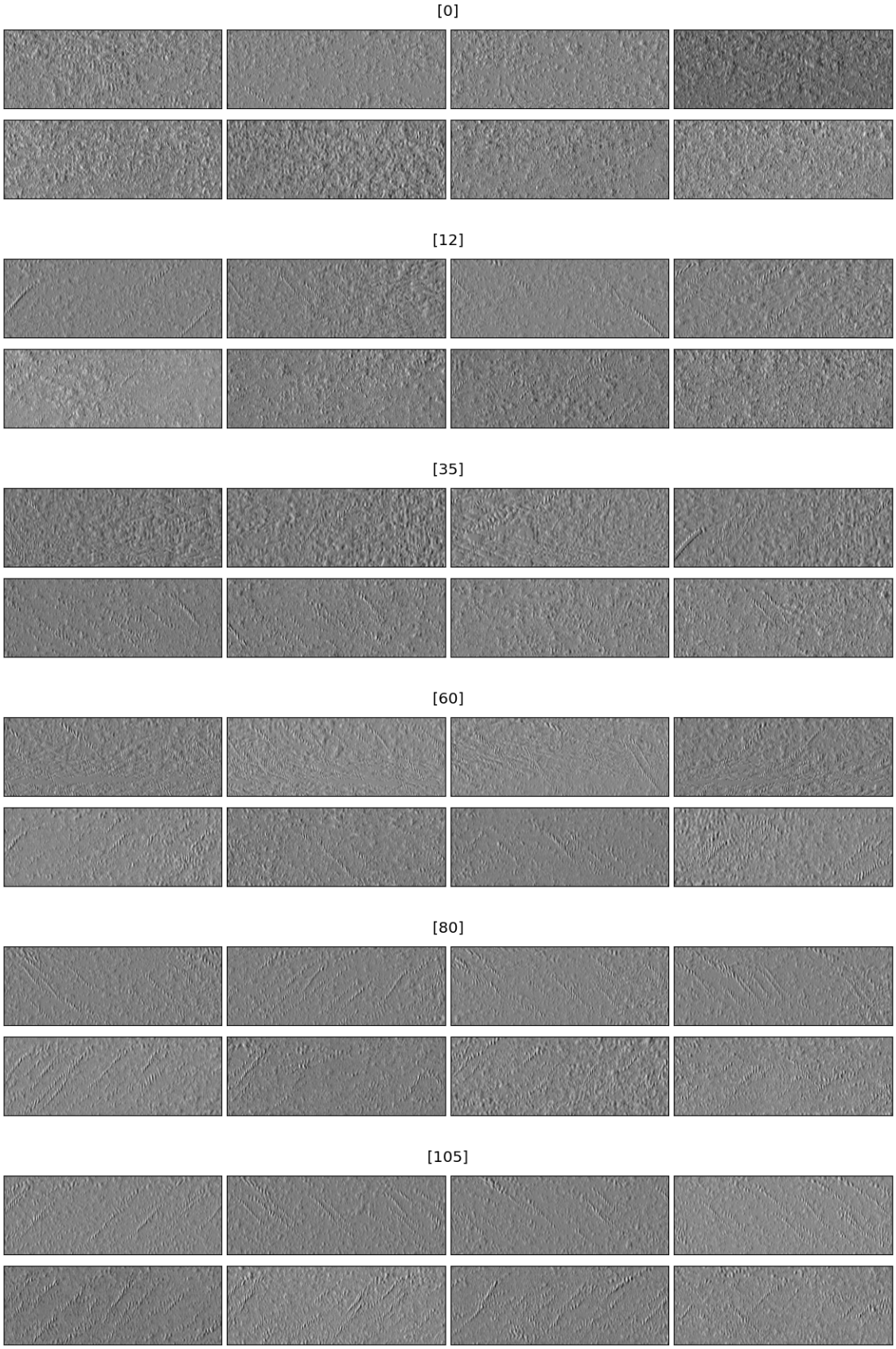

Figure 4.41: Real (top) and AC-WGAN generated (bottom) images of in vitro peritoneal dialysis

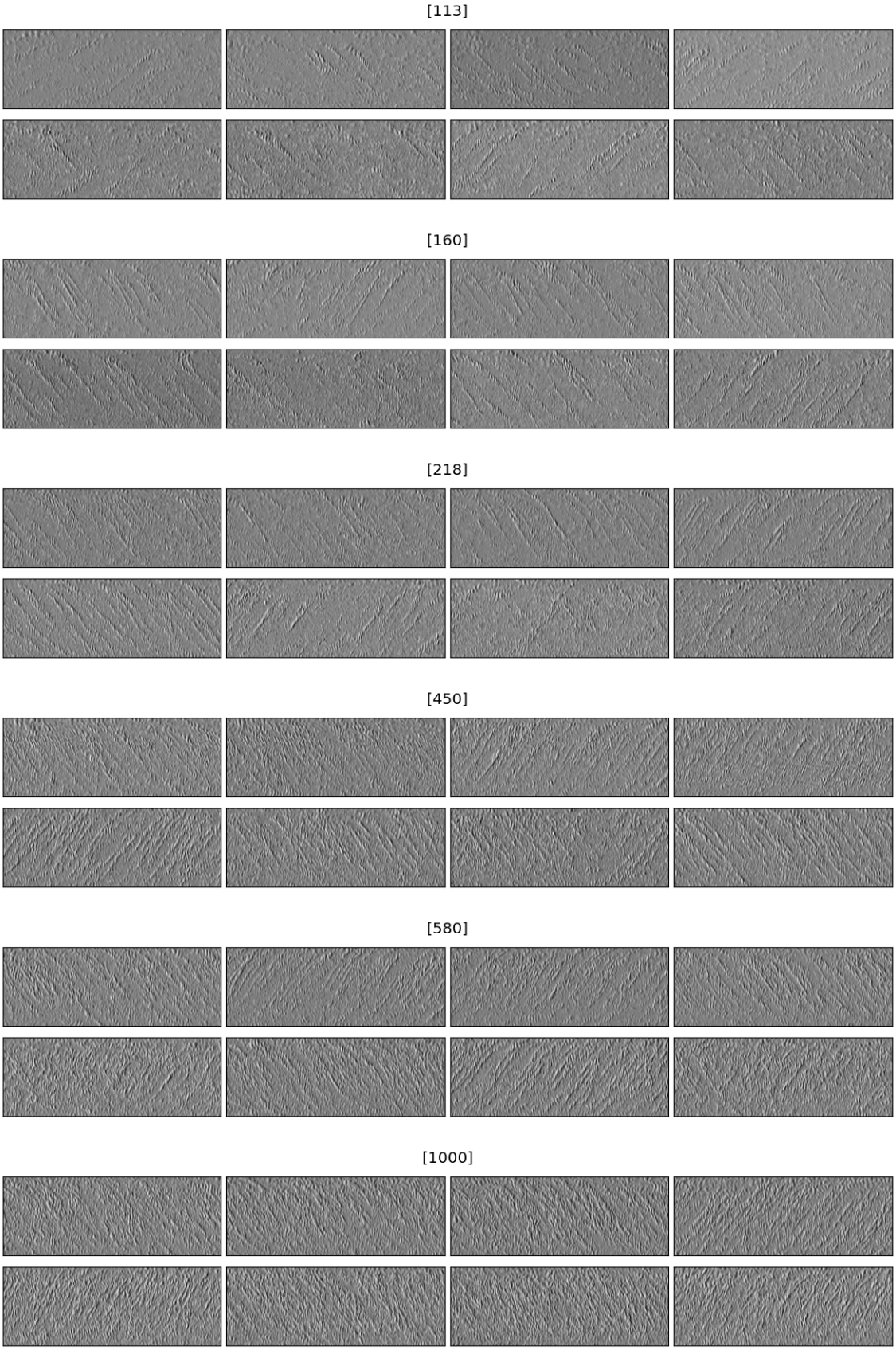

Figure 4.41: Real (top) and AC-WGAN generated (bottom) images of in vitro peritoneal dialysis

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	56.2	30.9	30.7	60.3	82.9	118.5	122.6	139.3	122.4	127.3	133.1	123.1
Real [12]	168.0	124.9	53.8	39.3	50.3	61.8	65.0	76.3	69.8	71.7	74.5	72.5
Real [35]	148.1	107.8	40.9	28.4	38.7	51.6	54.2	66.5	61.4	68.9	72.4	71.6
Real [60]	177.5	135.8	63.6	42.9	51.3	61.2	64.1	73.4	68.6	72.7	75.6	75.6
Real [80]	206.6	164.5	80.9	49.5	51.6	54.2	57.4	62.2	59.7	63.1	65.1	70.0
Real [105]	241.1	200.1	109.4	69.2	64.2	53.5	56.1	55.8	61.2	63.8	64.9	75.6
Real [113]	263.2	222.0	128.3	86.9	79.9	65.2	66.2	68.1	78.3	80.1	80.0	90.8
Real [160]	217.8	178.6	97.6	55.8	48.9	45.2	51.4	39.1	28.0	27.4	31.7	43.9
Real [218]	200.3	161.0	84.1	49.5	46.9	50.5	56.0	47.3	29.6	27.2	31.2	37.3
Real [450]	180.9	143.3	80.2	52.0	51.4	59.2	66.4	56.7	30.5	22.2	25.7	24.9
Real [580]	214.1	173.7	100.4	67.8	63.5	66.7	73.3	61.1	33.1	19.6	20.5	19.6
Real [1000]	203.4	164.2	102.5	80.4	79.0	86.8	93.3	83.7	49.0	27.8	26.7	16.9

Figure 4.42: Real-Fake FID matrix of C-WGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Real [0]	30.8	34.3	78.8	112.1	120.6	131.3	129.0	127.5	117.7	105.8	109.5	92.0
Real [12]	115.8	76.7	41.4	38.1	40.6	43.9	48.1	47.6	55.5	54.6	61.0	68.2
Real [35]	101.7	66.7	37.7	37.9	40.2	43.6	45.2	45.9	52.2	53.7	60.9	65.5
Real [60]	129.9	91.5	52.7	44.0	44.4	44.6	48.2	47.0	55.6	57.2	65.2	73.3
Real [80]	156.0	111.1	56.5	40.1	37.8	35.1	38.8	36.2	46.4	51.7	59.5	74.5
Real [105]	189.0	136.4	66.0	38.5	35.3	30.2	35.5	35.3	51.1	61.9	69.7	91.6
Real [113]	211.0	155.8	77.7	47.5	43.5	38.1	43.9	47.7	68.3	80.2	86.6	109.3
Real [160]	168.0	124.4	72.4	49.7	45.4	35.6	36.9	23.1	20.1	25.0	34.4	55.5
Real [218]	151.5	112.5	68.5	52.4	49.1	42.1	42.8	29.0	22.2	21.6	29.0	43.9
Real [450]	132.3	102.2	77.6	68.0	65.5	57.4	57.0	39.1	24.9	15.9	19.6	22.3
Real [580]	159.1	123.8	87.2	73.2	69.6	60.8	60.9	42.2	27.2	15.3	15.4	24.0
Real [1000]	145.4	116.9	94.4	88.1	86.5	80.9	81.2	61.8	41.5	20.0	14.5	13.2

Figure 4.43: Real-Fake FID matrix of AC-WGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The real-fake FID matrices show higher but similar distribution of scores to the reference matrix, proving good visual quality of the generated images, and, in general, increasing FID scores from the diagonal, indicating proper conditioning. However, in some cases the lowest FID score is not with the same concentration but with a different one, usually close. This would be expected for test concentrations never seen during training that the model is trying to infer, but it also occurs with training concentrations.

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	3.4	10.6	58.6	98.1	122.8	158.5	157.0	177.8	175.4	191.9	199.7	194.6
Fake [12]	10.7	4.4	32.1	68.4	92.5	127.3	127.4	145.9	140.0	152.1	159.1	152.8
Fake [35]	57.9	36.2	3.1	15.5	30.4	55.7	57.1	72.7	70.4	83.1	88.8	85.9
Fake [60]	97.0	73.0	16.0	4.7	8.2	23.9	26.4	35.5	37.3	54.9	60.4	65.5
Fake [80]	120.5	96.8	30.3	7.9	3.0	10.9	12.5	18.1	23.5	44.5	49.1	58.5
Fake [105]	155.5	133.0	56.5	24.1	10.9	5.4	5.0	9.3	26.4	50.3	53.4	67.9
Fake [113]	155.1	133.8	57.7	26.1	12.1	5.2	3.3	9.5	29.4	54.7	57.4	71.6
Fake [160]	177.5	154.9	76.4	37.2	19.1	10.3	10.5	5.3	18.0	42.7	46.5	64.5
Fake [218]	172.7	145.0	71.5	37.4	24.4	26.5	30.3	17.2	2.9	15.3	19.7	33.0
Fake [450]	188.7	156.4	83.4	55.1	45.7	49.3	54.4	40.3	14.5	3.5	4.6	13.1
Fake [580]	196.4	163.8	89.1	60.7	50.7	53.1	57.4	44.7	19.3	5.0	2.4	8.4
Fake [1000]	190.3	155.4	85.1	65.7	60.7	67.3	72.0	62.8	33.1	13.3	8.6	2.0

Figure 4.44: Fake-Fake FID matrix of C-WGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

	Fake [0]	Fake [12]	Fake [35]	Fake [60]	Fake [80]	Fake [105]	Fake [113]	Fake [160]	Fake [218]	Fake [450]	Fake [580]	Fake [1000]
Fake [0]	2.7	13.1	63.6	108.2	122.2	130.3	133.5	136.7	124.5	119.5	119.1	109.0
Fake [12]	9.8	4.2	31.2	65.8	78.1	86.4	90.1	96.7	91.3	90.6	92.0	88.6
Fake [35]	60.4	26.3	3.2	12.8	19.3	26.8	30.6	42.6	50.4	57.9	63.2	72.1
Fake [60]	107.3	61.3	12.7	4.2	4.7	8.8	11.2	23.2	38.9	52.5	60.3	76.5
Fake [80]	120.9	73.4	18.8	5.0	2.8	5.2	5.9	16.7	34.0	50.6	59.2	77.5
Fake [105]	129.4	82.3	27.2	10.0	5.9	4.9	4.4	11.3	27.6	46.2	55.7	75.4
Fake [113]	130.7	84.0	28.6	10.9	6.0	4.6	2.9	8.5	23.8	42.9	52.4	72.8
Fake [160]	134.0	90.6	40.0	21.9	16.1	11.4	7.8	4.2	11.6	29.6	40.1	60.7
Fake [218]	125.1	88.5	50.0	38.0	33.9	27.8	23.5	10.9	2.7	13.2	21.9	39.3
Fake [450]	117.0	85.9	58.2	53.3	52.2	47.4	45.0	30.0	13.1	3.4	5.1	15.2
Fake [580]	115.6	86.0	61.2	58.6	58.4	54.5	52.5	38.9	21.1	5.3	1.9	8.1
Fake [1000]	106.9	83.7	70.5	74.6	76.2	73.3	72.7	59.0	37.8	16.2	8.3	1.7

Figure 4.45: Fake-Fake FID matrix of AC-WGAN on in vitro peritoneal dialysis dataset. The lowest FID score for each concentration row is highlighted in blue

The fake-fake FID matrices show the conditioning is working properly: being the lowest in the diagonal and increasing away from the diagonal. The only case it does not occur is on the test concentration 105, presumably because it is too close to the training concentration 113. This shows the conditioning suffers with small differences in concentration. Additionally, the not too low FID scores within same concentrations are a sign of diversity in the generation and no mode collapse.

4.5.2 Regression models

The training of regression models was simpler than the previous GAN models and required minor hyperparameter tuning for training stability and convergence. The following is an outline of the experiments and final results for each dataset.

In vitro meningitis

Baseline regressor

Firstly, the baseline regressor was trained only using real images from the training and validation concentration set. Image regression is challenging and the general performance of the baseline regressor is not satisfactory and has room for improvement. The

per concentration metrics allow a more detailed overview of the weaknesses of the regression model, in this case, the lower concentrations, which are the most relevant for clinical purposes.

C-WGAN augmented regressor

The C-WGAN model achieved the best GAN evaluation metrics on the meningitis dataset both regarding visual quality and conditioning capacity. The C-WGAN augmented regressor was trained using real images and C-WGAN generated images from training concentrations and intermediate concentrations between consecutive original concentrations.

The C-WGAN augmented regressor improves the general performance of the baseline regressor, but it is still far from desired evaluation metrics. Furthermore, the per concentration metrics reveal that the improvement is on higher concentrations, with lower concentrations still having large errors.

Model	RMSE	MAE	MAPE (%)
Baseline regressor	73.47	52.98	114.00
C-WGAN augmented regressor	61.75	41.96	108.17

Table 4.3: General evaluation metrics of baseline and C-WGAN augmented regressors on in vitro meningitis test set

Concentration (pp/μL)	Baseline regressor			C-WGAN augmented regressor		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
3	4.96	4.76	158.58	12.36	11.59	386.23
7	38.42	37.09	529.89	26.10	24.79	354.12
15	24.34	18.33	122.18	16.70	11.78	78.53
50	30.39	28.97	57.93	27.92	26.14	52.29
200	129.50	117.93	58.97	78.18	62.04	31.02
600	79.53	67.03	11.17	102.76	82.38	13.73

Table 4.4: Evaluation metrics per concentration of baseline and C-WGAN augmented regressors on in vitro meningitis test set

In vitro peritoneal dialysis

Baseline regressor

Firstly, the baseline regressor was trained only using real images from the training and validation concentration set. The general performance of the regressor on peritoneal dialysis can still be improved. Once again, the per concentration metrics reveal the performance is quite good on higher concentrations but it decreases in lower concentrations.

AC-WGAN augmented regressor

The AC-WGAN model achieved the best GAN evaluation metrics on the in vitro peritoneal dialysis dataset both regarding visual quality and conditioning capacity. The AC-WGAN augmented regressor was trained using real images and AC-WGAN generated images from training concentrations and intermediate concentrations between consecutive original concentrations.

The AC-WGAN augmented regressor improves the general performance of the baseline regressor, but it is still far from desired evaluation metrics. Furthermore, the per concentration metrics reveal that the improvement is on the higher concentrations, with lower concentrations still having large errors.

Model	RMSE	MAE	MAPE (%)
Baseline regressor	40.73	30.30	54.89
AC-WGAN augmented regressor	36.12	27.68	58.48

Table 4.5: General evaluation metrics of baseline and AC-WGAN augmented regressors on in vitro peritoneal dialysis test set

Concentration (pp/μL)	Baseline regressor			AC-WGAN augmented regressor		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
12	23.63	21.87	182.25	26.47	24.92	207.69
60	33.83	21.29	35.49	29.87	16.48	27.47
105	20.13	17.35	16.52	23.33	20.82	19.83
160	31.66	29.51	18.44	32.88	29.84	18.65
450	70.27	58.96	13.10	56.17	45.06	10.01

Table 4.6: Evaluation metrics per concentration of baseline and AC-WGAN augmented regressors on in vitro meningitis test set

5 Discussion

The design and implementation of GAN models was challenging. Numerous methods and approaches were tried to avoid the problems of mode collapse, vanishing gradients and non-convergence. In the end, it was only the objective function that made a difference. The original minimax loss and the least square loss could not escape mode collapse in any of the experiments, while the Wasserstein loss completely avoided it without additional hyperparameter tuning.

Relative to the conditioning, both the Conditional and Auxiliary Conditional approaches proved similar performance in all GAN models, effectively conditioning the image generation. The C-WGAN model offered the best generation capabilities on the in vitro meninningitis data while the AC-WGAN model was better on the in vitro peritoneal dialysis data. Additionally, in the GAN models that captured realistic visual features, the conditioning worked as expected: correlating with the number of particles traces present in the image.

The main goal of the continuous scalar conditioning approach in this study was to generate images corresponding to concentrations not seen during training. According to the real-fake FID matrixes, generated images were not always the most similar to the corresponding real images for the given concentration, whether it was present during training or not. This could be seen as the conditioning underperforming, but it could be the opposite. Both visual inspection and the reference FID matrixes revealed that in vitro data presents other sources of variability apart from the concentration and corresponding number of traces. The GAN models did also capture some of this other visual variability per concentration, but it is likely that they generalized this variability and spread it out over all the range of concentrations. If this was the case, the GAN generated images conditioning could be higher correlated to the number of traces than the real images themselves. This hypothesis could be supported by the fact that in the fake-fake FID matrixes images from the same concentration are the most similar and the higher the concentration difference the higher the FID scores while this does not always occur in the reference FID matrixes between real images.

The performance of both GAN and regression models was better in higher concentration images, as shown by the real-fake FID matrixes and the regression evaluation metrics per concentration. This is explained by the fact that particles traces in higher concentrations overlap creating a pattern throughout the image that is easier to process and represent by convolutional layers than isolated particles traces that occur in lower concentrations. Additionally, lower concentrations present a great variability in the number of particles traces per image within the same concentration, resulting in an inherently noisy dataset.

The performance and evaluation metrics of models on the in vitro peritoneal dialysis dataset were better than on the in vitro meningitis dataset. The learning process from the earlier in vitro meningitis acquisition combined with the simplified acquisition from plastic bags compared to fontanel-phantoms, resulted in an improved acquisition methodology and conditions that reduced the visual variability from external sources. Additionally, the in vitro peritoneal dialysis dataset contained a larger number of training samples. In consequence, the in vitro peritoneal dialysis task was simpler as reflected on the performance of both the GAN and regression models.

Finally, regarding the ultimate goal of improving the generalization capacity and the overall performance of the regression model, results were not satisfactory. In the case of in vitro meningitis, there was a significant improvement of the C-WGAN augmented regressor over the baseline regressor on the general evaluation metrics. However, when inspecting the per concentration evaluation metrics it can be seen that it was mostly caused by fluctuations in the high errors of the lower concentrations. This means the regressor is still far from delivering a good enough performance. On in vitro peritoneal dialysis, there was a slight improvement in performance of the AC-WGAN augmented regressor over the baseline regressor on the general evaluation metrics. However, the per concentration evaluation metrics show that the improvement is not consistent, lowering errors in some concentrations but increasing them in others.

In conclusion, this study achieved the design and implementation of a continuous scalar conditional Generative Adversarial Network capable of generating good visual quality and diverse images containing features, concretely particles traces, with high resemblance to real ones. Additionally, the continuous scalar conditioning allowed to train the GAN model to generate images from intermediate concentrations never seen during training. However, the image regression task to predict concentrations proved to be challenging and the initial poor results of the baseline regressor could not be improved consistently by means of a GAN-based data augmentation approach.

6 Future steps

This study explored the implementation of continuous scalar conditional Generative Adversarial Networks for data augmentation in an image regression task. The results of the generative models met the expectations but not their application for data augmentation in regression.

The evaluation metrics of the GAN models revealed a better performance on higher concentration images than on lower concentration images. A different approach could benefit the generation of low concentration images, where particles traces are fewer and appear isolated. A mask-conditional GAN could be trained to generate images with the number, shape and location of particle traces given by a binary mask. This would allow a more controlled image generation process in lower concentrations that could improve the visual quality as well as the fidelity to the desired concentration value.

Regarding the regression task, the poor performance could be not only due to a lack of data, meaning data augmentation by itself would not solve the problem. Similarly to GAN models, there was a clear difference in performance between lower and higher concentrations that could be solved with a different approaches for each of them. For higher concentrations, the current image regression model could deliver good performance, but lower concentrations could require a truly image-based count approach implemented by means of an object detection or instance segmentation model.

Finally, GAN-based data augmentation could go beyond generating images to contain particles traces from given concentrations. Although in vitro images resemble in vivo images considerably, this mimicking could be further improved with generative modelling in a style transfer based approach from in vitro to in vivo images.

Bibliography

American Academy of Pediatrics, Altmann, T. R., and Hill, D. L. (2019). *Caring for Your Baby and Young Child: Birth to Age 5*. American Academy of Pediatrics.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan.

Barratt, S. and Sharma, R. (2018). A note on the inception score.

Benny, Y., Galanti, T., Benaim, S., and Wolf, L. (2021). Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731.

Chen, Y., Yang, X.-H., Wei, Z., Heidari, A. A., Zheng, N., Li, Z.-C., Chen, H., Hu, H., Zhou, Q., and Guan, Q. (2022). Generative adversarial networks in medical image augmentation: A review. *Computers in Biology and Medicine*, 144:105382.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of wasserstein gans.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2018). Gans trained by a two time-scale update rule converge to a local nash equilibrium.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks.

Kebaili, A., Lapuyade-Lahorgue, J., and Ruan, S. (2023). Deep learning approaches for data augmentation in medical imaging: A review. *Journal of Imaging*, 9(4).

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kinoshita, Y. and Kiya, H. (2020). Fixed smooth convolutional layer for avoiding checkerboard artifacts in cnns. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3712–3716.

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. (2019). A large-scale study on regularization and normalization in gans.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks.

Mino, A. and Spanakis, G. (2018). Logan: Generating logos with a generative adversarial neural network conditioned on color.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks.

Müller-Franzes, G., Niehues, J. M., Khader, F., Arasteh, S. T., Haarburger, C., Kuhl, C., Wang, T., Han, T., Nebelung, S., Kather, J. N., and Truhn, D. (2022). Diffusion probabilistic models beat gans on medical images.

Odena, A., Dumoulin, V., and Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill*.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans.

Ongun, C. and Temizel, A. (2018). Paired 3d model generation with conditional generative adversarial networks.

Pesteie, M., Abolmaesumi, P., and Rohling, R. N. (2019). Adaptive augmentation of medical data using independently conditional variational auto-encoders. *IEEE Transactions on Medical Imaging*, 38(12):2807–2820.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks.

Ruthotto, L. and Haber, E. (2021). An introduction to deep generative modeling. *CoRR*, abs/2103.05180.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans.

Saxena, D. and Cao, J. (2023). Generative adversarial networks (gans survey): Challenges, solutions, and future directions.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. (2015). Efficient object localization using convolutional networks.

Vahdat, A. and Kautz, J. (2021). Nvae: A deep hierarchical variational autoencoder.

Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., and Weinberger, K. (2018). An empirical study on evaluation metrics of generative adversarial networks.

Technical
University of
Denmark

Richard Petersens Plads, 324
2800 Kgs. Lyngby, Denmark
Tlf. 4525 1700

www.compute.dtu.dk