Addressing Class Imbalance in Neuropathology through Generative Adversarial Networks

Author: Ciaran Coleman[1]

MSc Data Science and Machine Learning

Primary Supervisor: Neil Oxtoby
Co-supervisors: Tammaryn Lashley & Andre Altmann

Revision date: 18 July 2022

Acknowledgements

I would like to thank my supervisor Dr Neil Oxtoby, for the support and advice he has given me throughout this project. To my co-supervisors Tammaryn Lashley and Andre Altmann, I extend my gratitude for their assistance whenever called on. In particular, Tammaryn for providing me with all the necessary background and contribution as the evaluator in the Visual Turing Test. Finally, I would like to say thank you to Moucheng Xu for our discussion that provided useful insight.

Abstract

Digitisation of Gigapixel histopathological whole slide images has led to widespread adoption of state-of-the-art computer vision techniques for efficient processing and analysis. In neuropathology, validation of recent work highlights the promise of integrating deep learning techniques for classification of Alzheimer's Disease and other neurodegenerative diseases. Nevertheless, the distribution of Aβ morphologies in these datasets is highly skewed, with cored plaques and cerebral amyloid angiopathy making up a small proportion of total instances. This is likely to be detrimental to deep learning classifiers that typically expect balanced data; we therefore propose to use generative modelling to oversample minority classes. Our model, PlaqueGAN, generates high quality synthetic samples across a wide range of modes that are indistinguishable to an expert neuropathologist. For oversampling imbalanced data, we propose a simple method based on the class-confidence score of a pre-trained classifier; this filters out PlaqueGAN samples that are low quality, or exhibit manifold intrusion. When training a downstream classifier on oversampled datasets, we find that PlaqueGAN with image selection produces the highest macro-averaged area under the precision recall curves (AUPRC). This exciting result hints that PlaqueGAN might be a viable solution for oversampling immunohistochemically stained Aβ datasets.

Contents

1 Introduction **2**
- 1.1 Motivations . . . 2
- 1.2 Thesis Aims . . . 3
- 1.3 Outline of Thesis . . . 3
  - 1.3.1 Code Availability . . . 3

2 Background **4**
- 2.1 The Class Imbalance Problem . . . 4
- 2.2 Classical Data-level Approaches . . . 5
  - 2.2.1 Random Oversampling . . . 5
  - 2.2.2 Random Undersampling . . . 5
  - 2.2.3 Oversampling with Augmentation . . . 6
  - 2.2.4 Synthetic Minority Oversampling Technique . . . 6
- 2.3 Generative Oversampling . . . 7
- 2.4 A Brief Introduction to Generative Adversarial Networks . . . 7
  - 2.4.1 GANs in Medical Imaging . . . 8
  - 2.4.2 Common GAN Failure Modes . . . 9
  - 2.4.3 Balancing GAN . . . 10
  - 2.4.4 Promising Advances in GANs for Imbalanced Data . . . 10

3 Methods **12**
- 3.1 Base Model: FastGAN . . . 12
- 3.2 PlaqueGAN . . . 13
  - 3.2.1 Measures to Improve Gradient Flow . . . 13
  - 3.2.2 Measures to Improve Training Stability . . . 13
  - 3.2.3 Measures to Increase Diversity . . . 16
- 3.3 Measures to Improve Discriminator Representation . . . 17
  - 3.3.1 Mixed Precision Training . . . 18
- 3.4 Self-Attention PlaqueGAN (SA-PlaqueGAN) . . . 19
- 3.5 Synthetic Image Selection . . . 20

4 Experiments and Results **22**
- 4.1 Datasets . . . 22
- 4.2 GAN Evaluation Metrics . . . 24

4.2.1 Our Approach to Evaluation . . . 24
4.2.2 Quantitative Metrics . . . 25
4.2.3 Qualitative Metrics . . . 29
4.3 Oversampling Evaluation Metrics . . . 31
4.4 Computing Environment . . . 31
4.5 PlaqueGAN Experiments . . . 31
4.5.1 Training Procedure . . . 31
4.5.2 Experiment on Proposed Improvements . . . 32
4.5.3 Experiments on All Datasets . . . 33
4.5.4 Qualitative Evaluation . . . 35
4.5.5 Experiments with Self-Attention (SA-PlaqueGAN) . . . 37
4.6 Oversampling Experiments . . . 38
4.6.1 Classifier Architecture . . . 39
4.6.2 Training Procedure . . . 40
4.6.3 Oversampling Requirements . . . 40
4.6.4 Baseline Oversampling Methods . . . 41
4.6.5 PlaqueGAN-synthesised Oversampling . . . 43
4.6.6 Final Results . . . 45

5 Further Discussion **47**
5.1 PlaqueGAN Captures a Diverse Set of Modes with High Fidelity . . . 47
5.2 Does Improving Gradient Flow in PlaqueGAN Actually Help? . . . 48
5.3 Labels from Single Experts May Prove Problematic . . . 50
5.4 Attention May Not Always Be What You Need . . . 50

6 Conclusions and Future Work **52**

Chapter 1

Introduction

1.1 Motivations

In the field of histopathology, digitisation of whole slide images (WSIs) – microscope slides of tissue that may be Gigapixels in size – has led to a digital revolution; with vast amounts of high resolution data available, this opens up possibilities to apply computer vision techniques to automate laboratory pipelines, establish visual standards and improve analysis throughput, freeing up pathologists' workload [17].

In neuropathology – specifically for Alzheimer's Disease – pathological diagnosis through examination of WSIs of brain tissue is the 'gold' standard [78]. The process of reaching a pathological diagnosis requires semi-quantitative scoring using guidelines set by the Consortium to Establish a Registry for Alzheimer's Disease (CERAD) [55]. According to these guidelines, the density of neuritic (cored) Amyloid-β plaques is key in the diagnosis of Alzheimer's Disease.

Recent works by Tang et al. [74] and Wong et al. [81] look at applying deep learning techniques to WSIs of immunohistochemically stained regions of the brain to help classify Alzheimer's Disease. Results from [74] show correlation between the scoring by a Convolutional Neural Network (CNN) and semi-quantitative scoring by an expert neuropathologist, alongside good precision and recall metrics. Vizcarra et al. [78] validate the methodology of [74] by extending it to data from a different brain bank; they too find similar correlation between semi-quantitative scoring and CNN scoring. The successful validation of this methodology offers promise in integrating deep learning in the neuropathology pipeline.

Although these works can reduce the neuropathologists' time spent scoring WSIs, one particular issue is highlighted in both [74, 81]: the Aβ morphologies of most interest are comparatively scarce with diffuse plaques dominating. Only 1-2% of candidate samples identified through traditional computer vision techniques are classed as cored plaques or Cerebral Amyloid Angiopathy (CAA). The datasets that are used are therefore highly skewed and this can be detrimental to the performance of the CNN. It would also be prohibitively expensive and time-consuming to try and increase minority class instances through further annotation by expert neuropathologists.

1.2 Thesis Aims

We seek to address the issue of imbalanced datasets for Aβ morphologies in immunohistochemically stained WSIs of brain tissue. As part of our methodology, we wish to explore the use of generative modelling – namely Generative Adversarial Networks – in synthesising minority instances of Aβ morphologies such as cored plaques and CAA to balance the dataset. In doing so, we may also reduce the time required to label quality datasets. To the best of our knowledge, this presents a first attempt at generative modelling the image distribution of Aβ morphologies.

We aim to answer the following questions:

1. *Can we find a GAN architecture that is able to synthesise high quality, novel instances of Aβ morphologies that are indistinguishable from real samples?*
2. *Is the GAN able to sufficiently cover the modes of the real data distribution, and does it avoid the issue of memorisation?*
3. *Even if the GAN can generally produce high quality samples, are there any instances of poor quality? If so, can we eliminate this using a selection procedure?*
4. *Can we improve the robustness of downstream classifiers for Alzheimer's Disease with datasets balanced by GAN-oversampling minority data?*

1.3 Outline of Thesis

The remainder of this thesis is structured as follows: In chapter 2 we provide background on techniques – classical and generative – that can be used to tackle class imbalance. In chapter 3 we formalise our model, PlaqueGAN, and detail the motivation behind its architecture and training. We also introduce our approach to synthetic selection. We present details of our experiments to evaluate PlaqueGAN as well as assess its viability for oversampling minority data in chapter 4. We provide some discussion in this chapter, but delve deeper into analysing the results in chapter 5. Finally, in chapter 6 we summarise the main findings and provide suggestions for promising future directions.

1.3.1 Code Availability

The code used in the project is available at `https://github.com/ciaran-coleman/PlaqueGAN`

Chapter 2

Background

We begin with a description of the class imbalance problem and its relation to histological image datasets and the wider field of medical imaging. Next, we summarise classical *data-level* techniques often employed to reduce its impacts and highlight the potential shortcomings of such techniques. We provide an overview of generative modelling, in particular Generative Adversarial Networks, which have exploded in popularity recently. We comment on their suitability to the problem, citing cases of medical imaging GANs that have been successful for data augmentation, and describe common failure modes that may be detrimental to downstream performance. Finally, we review a selection of recent research that may (directly or indirectly) assist in balancing datasets.

2.1 The Class Imbalance Problem

A dataset is said to exhibit class imbalance when the *imbalance ratio* between majority and minority classes is above moderate, sometimes ranging in the significant or extreme. Well-known examples of imbalance are the binary classification problems of detecting spam e-mail or fraudulent credit card transactions, where positive occurrences are much less frequent than negative occurences.

In the domain of deep learning, and more widely machine learning, much research has surrounded the difficulties of learning from imbalanced datasets. This is because popular learning algorithms and architectures often expect balanced datasets [27]; such datasets make benchmarking and comparing approaches on tasks such as classification accuracy fairer and simpler as it removes imbalanced data as a potential confounder. The downside is that algorithms and architectures that become widespread from their state-of-the-art (SoTA) status are often designed, refined and tested on such carefully curated, balanced datasets. When given an imbalanced dataset to learn from, performance often deteriorates [68] because learners place too much emphasis on the majority class, resulting in increased mis-classification rates for minority classes [33].

In the wild, it is common for data to be generated in an imbalanced manner. This is particularly the case with medical datasets, where the task is often to discriminate between normal/abnormal or healthy/unhealthy cases. *Normal* is normally assigned the negative class and *abnormal* the postive class. Intrinsically, abnormal samples occur far less frequently but are commonly of greater interest, making them more important to predict accurately; especially where false negatives carry

much more serious consequences than false positives.

This applies to histological datasets, where healthy tissue is much more prevalent. These datasets are also often multi-labelled in nature. For multi-labelled datasets, the problem is exacerbated as there are now two sources of imbalance: firstly, for any particular class, the number of negative instances may greatly outweigh positive instances. Secondly, the overall number of positive instances may be unevenly distributed among the classes. This reiterates the need to mitigate the imbalance problem.

2.2 Classical Data-level Approaches

Methods for tackling class imbalance can be grouped under two umbrella terms: *data-level* approaches and *algorithm-level* approaches [27]. The former involves manipulation of the data distribution, typically by means of adding new instances to the minority classes, subtracting instances from the majority class, or a combination of the two. The oversampling approach forms the focus of the thesis.

Algorithm-level approaches, on the other hand, target the training process directly. This includes popular methods that adjust the loss function, such as Focal Loss [48], Asymmetric Loss [5] and Class-balanced Loss [15], which share a commonality in that they perform per-sample re-weighting of the loss, often down-weighting the majority class and up-weighting minority classes. Although we do not explore these methods in this thesis, they should not be dismissed as they have demonstrated strong empirical performance on imbalanced datasets and are potentially complementary to data-level approaches.

2.2.1 Random Oversampling

In random oversampling, the number of minority instances is increased through random replication of existing minority instances. This is favoured for its simplicity and efficiency, but can often result in overfitting to those samples that are replicated [27]. The degree of overfitting may also be sensitive to the imbalance ratio of the dataset; a higher ratio meaning that minority instances are replicated more times. Random oversampling may also lack effectiveness due to there being no new information added to the dataset.

2.2.2 Random Undersampling

Opposite to random oversampling, random undersampling balances the dataset by randomly deleting instances belonging to the majority class. Equally simple and efficient, this approach is generally less favoured in comparison to oversampling as information is destroyed. Intuitively, the removal of majority samples that are in some sense 'close' to minority samples – i.e. they present more difficult cases that are informative to classifier learning – could reshape the decision boundaries and degrade performance.

Algorithm 1. SMOTE $(\mathcal{S}, r, k)$

Input: $\mathcal{S}$: Seed samples, samples of the minority class $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, m$
Input: r: Imbalance percentage
Input: k: Number of nearest neighbors
for $i = 1, 2, ..., m$ **do**
  Compute distances $\|\mathbf{x}_i - \mathbf{x}_j\|_2, \forall i \neq j$
  Find the k nearest neighbors asociated to the k minimum distances
  Compute the number of synthetic samples to be generated from $\mathbf{x}_i$, $n = \text{round}(r/100)$
  for $z = 1, 2, ..., n$ **do**
    Select a random integer ε between 1 and k
    Draw a random vector from a uniform multivarite distribution $\boldsymbol{\lambda} \sim \mathcal{U}_d(0, 1)$
    Compute the synthetic sample $\mathbf{s}_i^z = \boldsymbol{\lambda} \circ (\mathbf{x}_i - \mathbf{x}_\varepsilon) + \mathbf{x}_i$ where $\circ$ is the Hadamard product between vectors
  end for
end for
return The set of $n \times m$ synthetic samples $\{s_i^z\}$, $i = 1, 2, \ldots, m_-$, $z = 1, 2, \ldots, n$

Figure 2.1: Algorithm for Synthetic Minority Oversampling Technique (SMOTE). From [11]

2.2.3 Oversampling with Augmentation

Typically, data augmentation is applied in real-time during training of deep networks. This is primarily to improve generalisation performance through inflating the size of the training data, such that the network never sees the same instance twice during training. In the context of images, augmentations often include minor alterations that are label-preserving such as rotations, flipping, translation and colour jitter. More destructive augmentations such as cutout may also be applied.

Here, we consider augmentations as an offline step, improving upon random oversampling through altering the instances selected for replication before adding them to the dataset. As these instances are no longer duplicates, classifiers are less likely to overfit and new information may actually be gleaned from the augmentations. Nevertheless, care must be taken with the severity of augmentations used; in medical imaging, there is a risk that certain augmentations may alter the semantic content of the images [68], or completely remove certain labels in the case of multi-label learning (e.g. with cutout).

2.2.4 Synthetic Minority Oversampling Technique

Chawla et al. [11] introduced their seminal work, Synthetic Minority Oversampling Technique (SMOTE), that generated new information through *synthesis* of minority class samples. To generate new minority samples, SMOTE first randomly selects a minority class instance and one of its k-nearest neighbours. The element-wise distances between the two samples are then calculated and scaled by random numbers drawn from a uniform distribution. Finally, these scaled distances are added to the original instance to create the new instance. For further details, the algorithm for SMOTE from [22] is shown in Figure 2.1.

Since its inception, numerous variants of the SMOTE algorithm have been proposed to improve

upon the original implementation. In particular, SMOTE does not consider majority class samples. This can often result in instances of manifold intrusion where separation between the classes is not clear [18]. Borderline-SMOTE [26] is one of many variants that incorporates information from majority class samples, guiding the synthesis of minority samples to lie near the borders between classes; the idea being that these will provide higher quality information for the classifier.

Another downside to SMOTE is that it is generally only suitable on lower-dimensional vector or tabular data [68]; operating in the feature space was its original intention, thus it is not well-adapted to being applied directly to images in the image space. Part of this is due to the curse of dimensionality, where assigning nearest neighbours based on Euclidean distances in high-dimensional spaces essentially become meaningless as all samples look equally distant. Additionally, linear interpolation of RGB pixel intensities can often lead to unrealistic images.

Nevertheless, SMOTE has seen some success when applied directly to images in histopathology datasets [65], improving downstream classification performance. This success is unexpected, but could be explained by the lower complexity and variation seen in histopathological images compared to natural images; meaning linear interpolation between images are less likely to stray far outside of the manifold.

2.3 Generative Oversampling

All data-level methods described so far make no attempt to learn the underlying distribution of the data. A better approach to synthesising new minority class samples would be to learn this underlying, true data distribution and then sample from it. Deep generative models are an active area of machine learning research that seeks to do just this.

Due to the complexity of high-dimensional image data, this challenging task has been approached in numerous ways through deep neural networks [24, 68]. Two of the more popular methods which can be optimised via standard gradient descent are Variational Autoencoders (VAEs) [39] and Generative Adversarial Networks (GANs) [23]. Significant advances in recent years to partially rectify teething problems has seen the balance shift in favour of GANs. This is because VAEs often adopt a mean squared error (MSE) reconstruction loss, which tends to produce more blurry images by assigning equal importance across all pixels, and is thus unable to model high-frequency features in images [68].

2.4 A Brief Introduction to Generative Adversarial Networks

Introduced by Goodfellow et al. [23] in 2014, GANs have quickly risen in the ranks to become the de facto for image synthesis tasks. The motivation behind GANs is that they provide a means to obtain an implicit representation (P_g) of the data generating distribution (P_r)[14]. We wish for the distribution P_g to approximate P_r as closely as possible.

To achieve this, the GAN framework involves two networks – the *generator*, G, and the *discriminator*, D, often parameterised as deep neural networks with respective parameters $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$. Originally, both generator and discriminator were multi-layer perceptrons, but this was soon

usurped by convolutional backbones [64] due to their established adaptability to computer vision tasks.

These networks constitute the 'players' which are in adversarial competition in a two-player game. In this game, G samples a vector $\mathbf{z}$ from the latent space $\boldsymbol{\mathcal{Z}}$ and maps this to a sample in the desired output domain, for example the image space: $G : \mathbf{z} \rightarrow \hat{\mathbf{x}}$ where $\hat{\mathbf{x}}$ is in the space of real RGB images (continuing our example). The discriminator D alternates between receiving samples of real images and generated images, classifying them as either real (drawn from P_r) or generated (drawn from P_g). The output of D is in a sense a probability $p = D(\mathbf{x}, \boldsymbol{\theta}_D)$ that $\mathbf{x}$ is a real sample. This discriminator task reduces to binary classification, thus binary cross-entropy loss is suitable:

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{x}\sim P_r}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim \boldsymbol{\mathcal{Z}}}[\log(1 - D(G(\mathbf{z}))] \tag{2.1}$$

The discriminator's training objective is to **maximise** $-\mathcal{L}_D$ while the goal of the generator is the opposite, i.e. to **minimise** $-\mathcal{L}_D$. Intuitively, the generator tries to produce images that fool the discriminator into classifying as real. This boils down to a value function $V(G, D)$ of a zero-sum game:

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\mathbf{x}\sim P_r}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim \boldsymbol{\mathcal{Z}}}[\log(1 - D(G(\mathbf{z}))]] \tag{2.2}$$

Both networks are trained through alternating stochastic gradient descent. Theoretically at convergence, generated samples should be no different to real samples; meaning the discriminator outputs probability $p = 0.5$ everywhere. This point is referred to as the Nash Equilibrium - where neither network can improve further in their own objective. We see later in this chapter that often such convergence is optimistic.

2.4.1 GANs in Medical Imaging

As there is no known research on generative modelling for neuropathological Aβ datasets similar to ours, it was unknown whether the method would prove successful without baselines to compare against. Instead, we look to cases where GANs have been applied successfully to histopathological data as a proxy for potential success.

Xue et al. [84, 83] apply a conditional GAN model, HistoGAN, to augment histopathological data for the purpose of improving accuracy of precancer diagnosis. They combine HistoGAN with a selective synthetic sampling procedure to remove low quality and uninformative samples, and demonstrate an overall improvement in classification accuracy of 6.7% on a dataset of cervical histopathology and 2.8% on a metastatic cancer dataset. For Levine et al. [46] image synthesis, via a progressively growing GAN architecture, is applied to ovarian cancer histopathology images. They find their GAN capable of generating images no different to real samples when reviewed by an expert pathologist. As in [83], they too find GAN-synthesised images to be useful for dataset augmentation, leading to improved area under the curve (AUC) when a downstream CNN is trained on the augmented dataset.

Quiros et al. [63], provide a use for GANs other than for augmenting histopathology datasets. PathologyGAN combines a state-of-the-art conditional GAN [10] with elements from a state-of-the-art unconditional GAN [35]. Namely, the introduction of a mapping network and style mixing [36]

result in PathologyGAN having a disentangled latent space that captures important tissue features in colorectal cancer and breast cancer datasets. Through latent space exploration techniques such as applying vector arithmetic, and dimensionality reduction techniques such as Uniform Manifold Approximation and Projection (UMAP) [52], they demonstrate the interpretability of the latent space. As in [46], Visual Turing Tests with expert pathologists reveal no clear difference between generated and real images.

These examples provide some degree of indication that GANs would find similar success in neuropathology datasets.

2.4.2 Common GAN Failure Modes

In the formative years, successful training of GANs was seen to be a challenge in itself. This has been accredited to the optimisation problem representing the difficult task of finding a saddle point (i.e. the Nash equilibrium) [67]. Well-known modes in which a GAN could fail are:

Non-convergence

Training two separate neural networks leads to a highly non-convex optimisation problem. The dynamics of alternating stochastic gradient descent between generator and discriminator networks can result in *cycling*, whereby the network parameters become trapped in an 'orbit', failing to converge to their optimal values [3]. This manifests as poor image quality which fails to improve throughout training. Samples of low quality are likely to lie outside of the manifold of real data (Figure 2.2c).

Mode Collapse

The mode collapse issue is a common source of training instability in GANs, hypothesised to be the result of undesirable local equilibria and empirically found to be accompanied by the discriminator exhibiting sharp gradients [41]. Mode collapse is typically characterised by a generator producing a severely restricted subset of modes of the real data distribution; multiple latent vectors $\mathbf{z}$ essentially map to the same output point $G(\mathbf{z})$. This is illustrated in Figure 2.2e.

Mode Dropping

A less severe form of the mode collapse problem, with mode dropping the generator is only able to adequately learn a subset of the real distribution (Figure 2.2d). Less dense regions of the real distribution, or 'harder' examples, are often underrepresented without an explicit objective to guide the generator to cover all of them. This is an extremely common occurence in GANs, and can be difficult to detect when there exist many subtle modes that are difficult to visually separate.

Overfitting

Overfitting is notoriously hard to spot in GANs due to a lack of reliable quantitative metrics that can adequately detect it, and visual checks will automatically favour a memory GAN [8, 9]. With overfitting, generated samples lie close to real samples on the manifold (Figure 2.2b).

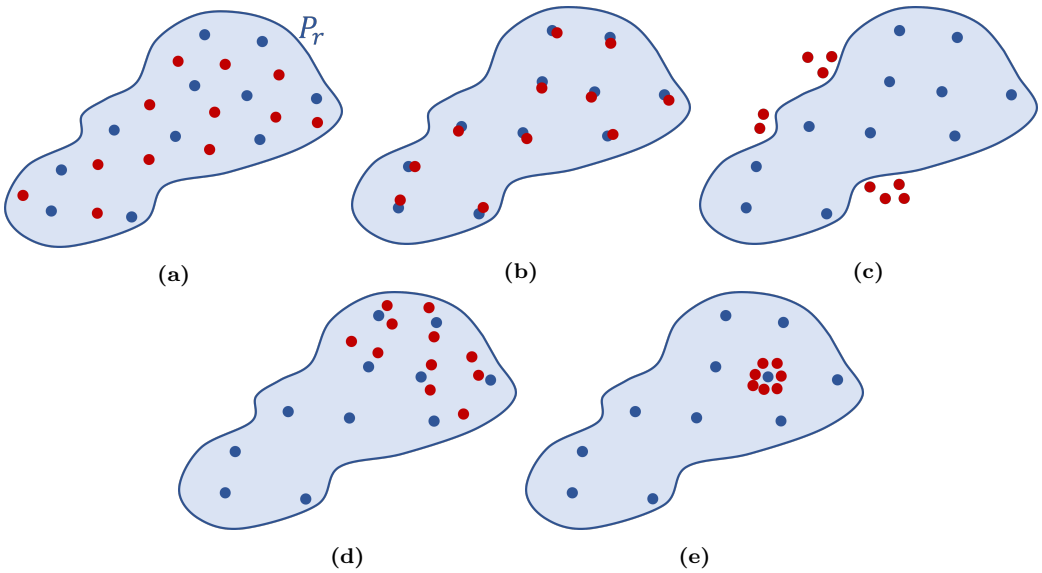


Figure 2.2: Visualisation of common failure modes in GANs. (a) Ideal distribution of generated samples (red) compared to real samples (blue). Generated samples fill the gaps of the real distribution, denoted P_r. (b) Overfitting, where generated samples are very similar to real samples (c) Unrealistic generated samples that lie outside of the real distribution (d) Mode dropping. The generator is only able to learn a portion of the real distribution (e) Mode collapse. A severe form of mode dropping, where the generator can only produce samples around a very restricted subset of the real distribution.

2.4.3 Balancing GAN

Although GANs offer much potential to solve class imbalances in data, there are surprisingly few papers that focus directly on architectural and training improvements in the imbalanced setting. This is likely due to imbalanced datasets having insufficient minority samples for GANs to train on.

The Balancing GAN (BAGAN) introduced by Mariani et al. [51] is an example of GAN research that attempts to tackle the issue directly. They integrate an autoencoder, first trained using all training data, and use it to initialise the weights of the generator and discriminator. At the beginning of training the generator inherits from the decoder half of the autoencoder, thus giving it prior knowledge of what images corresponding to the latent vector drawn $\mathbf{z}$ should look like. This acts to stabilise training. Additionally, they enforce balancing of classes within each minibatch to upweight the importance of minority classes during training.

Nevertheless, BAGAN still shows stability issues when trained on medical image datasets [30] where intra-class variation can be greater than inter-class variation, making it difficult to tell classes apart. This calls into question the suitability of such an architecture for our histological datasets, where Aβ morphologies share these issues.

2.4.4 Promising Advances in GANs for Imbalanced Data

The issues of GAN training highlighted in §2.4.2 are exacerbated when training with limited data and limited batch sizes (due to hardware restrictions) [49]. There exist many different methods of addressing these issues and some will be described in chapter 3. However, it is outside the scope

of this thesis to survey all of them. Instead, we look only at a couple of recent techniques which could assist training GANs for the task of balancing data.

Differential Augmentation

Both Zhao et al. [91] and Karras et al. [38] independently discovered augmentation techniques that enable GANs to be trained with limited data. Much like how data augmentation for training CNNs mitigate overfitting, Differentiable Augmentation [91] and Adaptive Discriminator Augmentation (ADA) [38] act to prevent the discriminator overfitting in limited data scenarios. In doing so, constructive feedback from the discriminator continues which prevents divergence of training.

Both schemes address the limitation that augmenting real training data leads to GANs inheriting the augmentations; destructive augmentations like cutout are therefore catastrophic to the quality of generated images when these augmentations leak. Instead, differentiable augmentations are applied to both real and generated images, allowing the gradients to backpropagate through them without altering the target distribution.

Self-supervised Discriminators

Chen et al. [12] hypothesise that discriminators in a purely adversarial setup are prone to *catastrophic forgetting*, where the learned features important to the adversarial task can vary greatly throughout training. They introduce the idea of auxiliary tasks (other than classification) with its own loss; the notion being that tasks separate to the main adversarial task can regularise the discriminator by inducing it to learn meaningful features of the images. This can enhance training stability, and quality of generated samples [31].

A Model that Combines Both

Liu et al. [49] combine both of these ideas in their architecture, FastGAN, to address the stability issues encountered not just in small datasets, but with limited hardware as well (where batch sizes must remain small).

By employing DiffAugment and an autoencoding auxiliary task for the discriminator, their model yields high fidelity results on a number of low-shot datasets – with as few as 100 samples – whilst exhibiting fast convergence. They demonstrate that FastGAN surpasses the state of the art StyleGAN2-ADA (another GAN conceived to train on limited datasets) when trained on a single GPU. This work provides promise that GANs might be able to learn from severely imbalanced datasets where minority instances are only in the hundreds. We discuss the elements of FastGAN in greater detail in §3.1.

Chapter 3

Methods

In this chapter, we detail our strategy for mitigating the class imbalance problem in histological datasets of Aβ morphologies. We begin by selecting a baseline GAN architecture from existing literature and explain why it is suitable to our requirements. We then propose a series of considered modifications to both architecture and training procedure to increase diversity and training stability, arriving at PlaqueGAN: an unconditional GAN for synthesising new histological instances of Aβ morphologies. Finally, we discuss our approach to synthetic image selection and how it can increase the likelihood of PlaqueGAN succeeding in oversampling data.

3.1 Base Model: FastGAN

We select the 'FastGAN' architecture introduced by Liu et al. [49] as an appropriate backbone on which to build our model, PlaqueGAN. As mentioned in §2.4.4, this model has shown promising results in the few-shot setting, with datasets containing fewer samples than considered in this thesis. Importantly, the architecture also meets our requirements of generating high fidelity images with limited time and computational resources.

The success of FastGAN is attributed to two techniques that target improved training stability. The first of these is the novel Skip-Layer Excitation (SLE) modules implemented in the generator. The SLE module performs channel-wise recalibration between feature maps that are far apart in the generation process. This offers the advantage of shortcut gradient flows, strengthening feedback during training to avoid the vanishing gradient problem. It also means that informative features learned at lower levels of the generator can more efficiently transfer to the generation process at higher levels.

The second – and perhaps more important – technique introduced is the self-supervised autoencoding task of the discriminator. Here, the backbone of the discriminator is viewed as an encoder and intermediate feature maps represent different levels of image encoding. The auxiliary task of reconstructing real images from intermediate feature maps induces the discriminator to learn useful representations which are less prone to fluctuate wildly throughout training. Compared to the auxiliary rotation task in [12], the autoencoding task is more appropriate to our datasets. This is because Aβ morphologies are largely rotationally symmetric, and human perception of the morphologies is unaffected by the rotation. Trying to predict whether an image of an Aβ mor-

phology has been rotated would be an impossible task for the discriminator, unlike with natural images where there is a notion of what the 'correct' way up is.

3.2 PlaqueGAN

We now propose a number of changes to the FastGAN model that focus on improving diversity of generated images, in addition to reinforcing training stability. With these changes implemented, we call our model PlaqueGAN after its purpose of generating realistic examples of Aβ plaque morphologies.

Diagrams of the generator G and (two) discriminators D_1, D_2 of PlaqueGAN are provided in Figures 3.1 and 3.2 respectively. In these diagrams, changes made to the FastGAN model are outlined in green for clarity.

3.2.1 Measures to Improve Gradient Flow

To improve upon FastGAN's convergence speed, we propose a couple of changes to strengthen the gradient flows in both generator and discriminator. We consider the use of SLE modules within the main discriminator, as its depth and complexity is similar to that of the generator. We also increase the number of SLE modules for a 256×256 image from 2 to 3 in both generator and main discriminator. The SLE modules are lightweight [49] so this should not incur much memory or speed penalty.

We also choose to drop the Tanh activations from both generator and decoder outputs of the discriminator. Tanh activations were originally used to enforce the output space of generated images to be in the same range as real images ($[-1, 1]$), but recent state-of-the-art GANs such as StyleGAN and StyleGAN2 [35, 36] omit this. Its importance on image quality is therefore unclear, but it does present yet another layer gradients need to backpropagate through. Omitting should theoretically lead to stronger gradient signals.

3.2.2 Measures to Improve Training Stability

Inspired by the Multi-scale Gradients GAN (MSG-GAN) of Karnewar et al. [34], we look to increase the regularisation of PlaqueGAN by applying constraints to the generator architecture. Rather than output images solely at 256×256 pixel resolution, we synthesise a second image at a resolution of 128×128 through a (1×1) convolution of feature maps in the generator's penultimate layer. This regularisation means that both penultimate and final layer feature maps can be readily projected into the image space, which acts to constrain their expressiveness.

Unlike MSG-GAN, where the lower resolution image would be incorporated through concatenation to the relevant layer of a *single* discriminator, we instead couple the lower resolution image to a *second* discriminator. We refer to the main discriminator as D_1 and second discriminator as D_2 and choose for D_2 to have a simplified structure (architecture shown in Figure 3.2b) in an effort to balance allocation of memory. We reformulate the overall adversarial loss as an ensemble

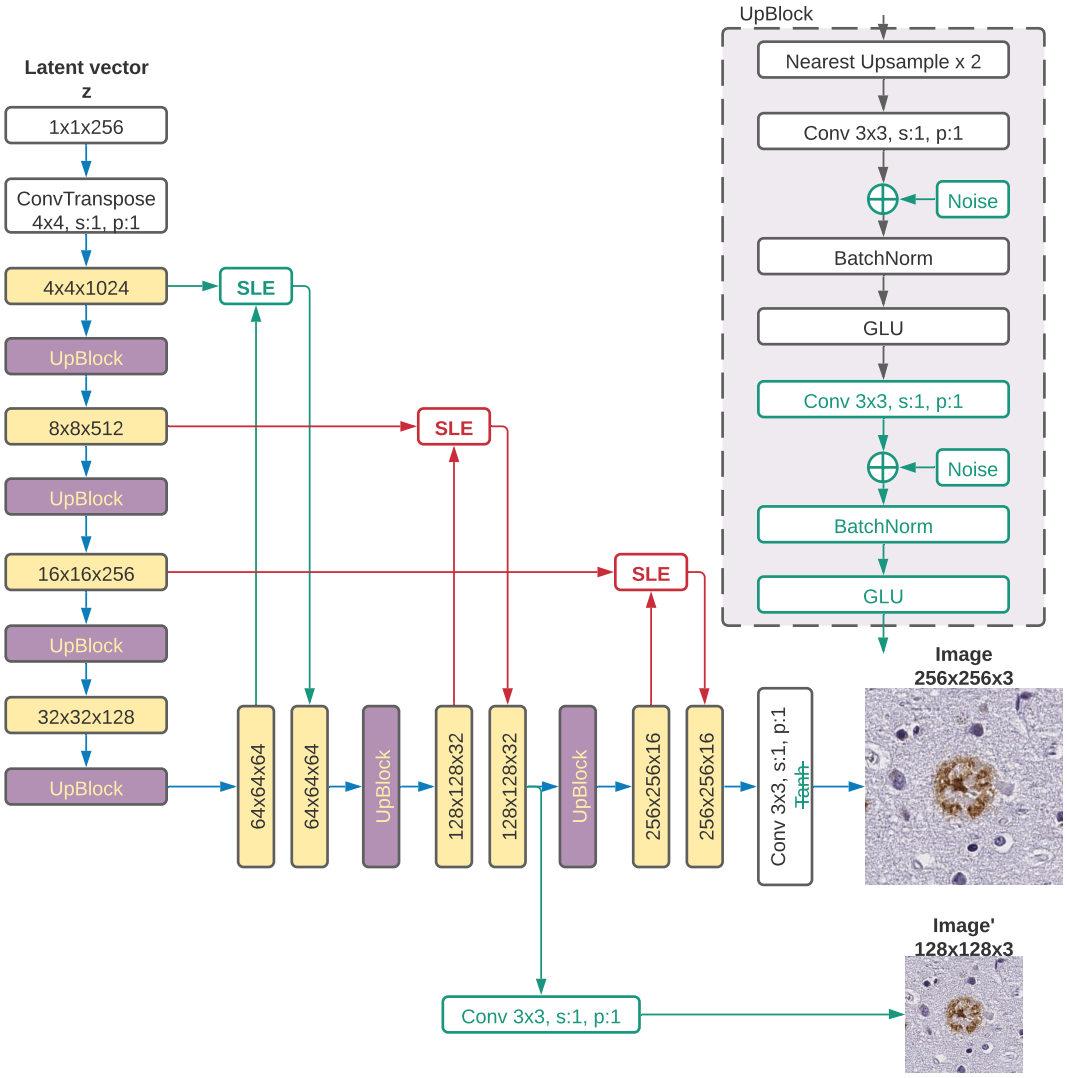
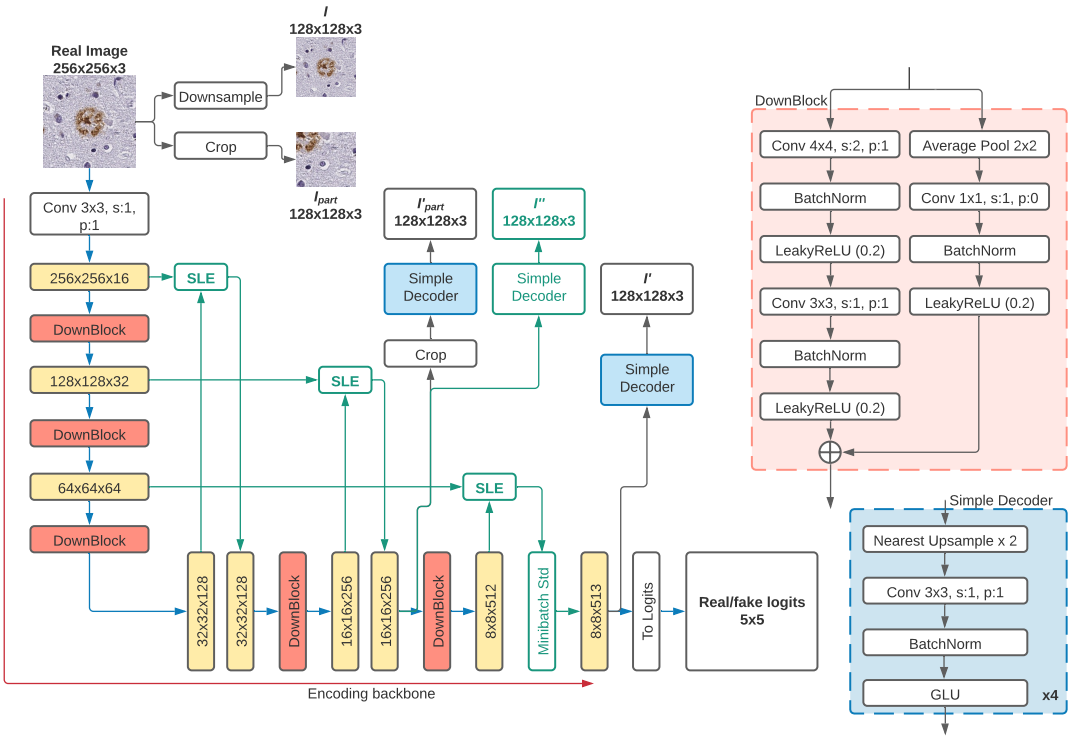
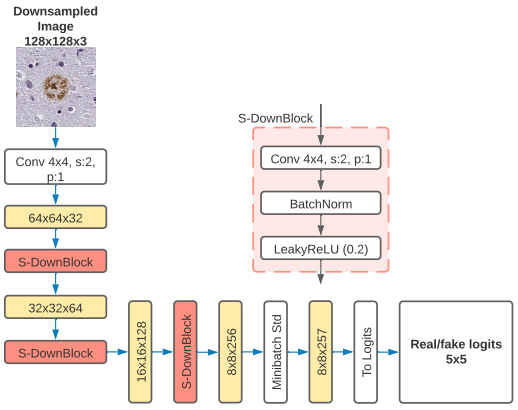


Figure 3.1: Architecture of the generator, G of PlaqueGAN. All modules highlighted in green indicate the architectural changes to FastGAN. These include increased capacity, stochastic variation, an additional SLE module as well as a second image output at 128×128 pixel resolution.



(a) D_1



(b) D_2

Figure 3.2: Structure of the primary and secondary discriminators of PlaqueGAN. (a) Primary discriminator D_1 performs the auxiliary reconstruction task. All modules highlighted in green indicate the architectural changes to FastGAN. (b) Second discriminator D_2 has a much simpler structure in comparison, without SLE modules and auxiliary tasks.

average of the individual losses of each discriminator:

$$\mathcal{L}_{\text{adv}} = \frac{1}{2}(\mathcal{L}_{\text{adv},D_1} + \mathcal{L}_{\text{adv},D_2}), \tag{3.1}$$

where:

$$\mathcal{L}_{\text{adv},D_i} = -\mathbb{E}_{\mathbf{x}_i \sim P_r}\left[\min(0, -1 + D_i(\mathbf{x}_i))\right] - \mathbb{E}_{\hat{\mathbf{x}}_i \sim G(\mathbf{z})}\left[\min(0, -1 - D_i(\hat{\mathbf{x}}_i))\right], \;\; i \in \{1, 2\} \tag{3.2}$$

is the adversarial hinge loss introduced by Lim et al. for GeometricGAN [47]. $\mathbf{x}_1$, $\mathbf{x}_2$ refer to the full-size real image and downsampled real image respectively, while $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$ are the full-size (256×256) and lower resolution (128×128) images produced by the generator (Figure 3.1).

The overall loss for the discriminator and generator are therefore:

$$\mathcal{L}_D = \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{recon}} \tag{3.3}$$

$$\mathcal{L}_G = \frac{1}{2}(-\mathbb{E}_{\mathbf{z} \sim \mathcal{Z}}[D_1(G(\mathbf{z})[1])] - \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}}[D_2(G(\mathbf{z})[2])]), \tag{3.4}$$

where $G(\mathbf{z})[1] = \hat{\mathbf{x}}_1$, $G(\mathbf{z})[2] = \hat{\mathbf{x}}_2$ are the full-image and intermediate image output by the generator respectively. In [20], the benefit of multiple discriminators are thought to be due to:

1. Better odds of the generator receiving constructive feedback. This may be of particular importance at the start of training, where a single discriminator is likely to overpower the generator; adversarial loss falls to zero and the generator no longer receives any useful feedback. By having more than 1 discriminator, the likelihood of the individual adversarial losses all dropping to zero simultaneously is lowered.

2. Functioning as an ensemble can help reduce the variance of the discriminator feedback, which will in turn guide the generator more directly towards improvement.

Although we only add one discriminator in this thesis, we believe that this still provides a positive effect on training stability.

3.2.3 Measures to Increase Diversity

Minibatch Standard Deviation

A common limitation of discriminator architectures is the lack of an explicit objective to guide generators to produce diverse images. Instead, they typically review images in isolation and are thus unable to identify whether a batch of generated images all look similar.

Salimans et al. [67] introduced the general concept of Minibatch Discrimination to allay this deficiency. They describe Minibatch Discrimination as any technique that uses information about the entire minibatch to enhance the discriminator's ability to distinguish between real and fake samples. They suggest one such formulation, but this involves learning a large 3D projection tensor and introduces significant computational overhead. Karras et al. [37] construct a lightweight, simpler approach on the intuition that the minibatch information need not be particularly complicated. They determine that the mean standard deviation across the minibatch, for every feature map and spatial location of the (typically) penultimate layer of the discriminator, is sufficient to

improve diversity. The activations to the final layer of the discriminator are then concatenated by a single constant feature map, whose entries are the mean standard deviation, replicated across all spatial dimensions.

As this approach is quick to compute and requires no additional learnable parameters, we include the Minibatch Standard Deviation layer in both main and secondary discriminators of PlaqueGAN.

Stochastic Variation

Introduced by Karras et al. [35] for StyleGAN, the idea behind stochastic variation is to offer the generator a path of least resistance when learning how to generate minor perturbations to localised features. This frees up some of its capacity to focus on learning more important, high-level semantics of the training data.

Stochastic variation is achieved through injection of per-pixel random noise, drawn i.i.d. from a standard Gaussian, to all feature maps following a convolution. It is extremely lightweight as there is a single learnable weight per noise injection module, which linearly scales the amplitude of noise added to the feature maps. For PlaqueGAN, we follow this implementation by injecting per-pixel noise after every convolution.

Increased Generator Capacity

The FastGAN architecture uses a single convolutional layer at each resolution of the generator to keep it lightweight and speed up training. However, if the generator capacity is overly restricted, it may only learn a portion of the real distribution well. This can lead to mode dropping and reduced diversity of generated images.

As our datasets only require generation of 256×256 pixel images, unlike FastGAN which was intended to generate images up to a resolution of 1024×1024, we are able to transfer the excess capacity into having two convolutional layers at every resolution of the PlaqueGAN generator. This is more in line with SoTA models [35, 36, 10].

3.3 Measures to Improve Discriminator Representation

Auto-encoding Reconstruction Loss

In the original FastGAN paper, the classical mean-squared (ℓ_2) error is chosen to model the reconstruction loss of the main discriminator's auxiliary task. Following the notation in [49]:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{f}\sim D_{1,\text{enc}}(\mathbf{x}),\mathbf{x}\sim I_{\text{real}}} \left[\| \mathcal{G}(\mathbf{f}) - \mathcal{T}(\mathbf{x}) \| \right] \tag{3.5}$$

In this formulation, $\mathcal{T}$ is a transformation (cropping or resizing) on $\mathbf{x}$, a real image sampled from the training data I_{real}. The function $\mathcal{G}$ contains both the transformation (if cropping) and decoding of an intermediate feature map $\mathbf{f}$, extracted by pushing the image $\mathbf{x}$ through the encoding backbone of the main discriminator $D_{1,enc}$.

A popular metric for image reconstruction, the MSE found widespread adoption primarily for its desirable properties of convexity and differentiability. Nevertheless, exclusive use of this

loss function often leads to less desirable results, producing blurry images that may also contain artifacts [90]. Reconstruction quality metrics derived from MSE such as Peak Signal-to-Noise Ratio (PSNR) are also shown to correlate poorly with human judgment when various distortions such as blurring, noise and contrast changes are introduced [56].

We therefore propose the Learned Perceptual Image Patch Similarity (LPIPS) [89] metric for the reconstruction error, hypothesising that it will enhance the representations learned by the auxiliary task of the main discriminator:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{f}\sim D_{1,\text{enc}}(\mathbf{x}),\mathbf{x}\sim I_{\text{real}}}\left[d_{\text{LPIPS}}\left(\mathcal{G}(\mathbf{f}),\mathcal{T}(\mathbf{x})\right)\right], \tag{3.6}$$

where $d_{\text{LPIPS}}(\mathbf{x},\mathbf{x}_0)$ is the LPIPS distance between two image patches $\mathbf{x}$ and $\mathbf{x}_0$, computed by taking the ℓ_2 distance between them in the feature space of a pre-trained CNN. For further details of how the distance is calculated, please refer to Equation (1) in §3 of [89]. The authors find that ImageNet pre-trained network architectures such as VGG [72] and AlexNet [42] have internal features that correspond well with human judgement; but show that further calibration of the network by addition of a small number of parameters, combined with training specifically for the task of predicting image quality using a perceptual similarity dataset, leads to further improvement in correlation. This is the *lin* configuration, of which we use the VGG variant of (VGG-lin) to compute the LPIPS distance throughout this thesis.

A Third Decoder

We investigate the effect of a third decoder for the auxiliary task, operating at the same scale of the texture decoder (16×16) and outputting an image $I^{''} \in \mathbb{R}^{128\times128\times3}$. However, unlike the texture decoder which aims to reconstruct a cropped section of the original image, this decoder seeks to reconstruct the full (but downsampled) image; much like the decoder operating at 8×8. We believe that this can enforce better continuity between feature representations at 16×16 and 8×8 as the features at 16×16 must now learn both textural and global structure rather than texture alone.

3.3.1 Mixed Precision Training

Although not expected to directly improve PlaqueGAN performance, mixed precision training [54] enables more complex architectures to run on moderate hardware by reducing the memory requirements (up to $2\times$ depending on the architecture) through use of half-precision floating point numbers in place of single-precision numbers during training. The authors introduce gradient scaling and maintaining a 'master copy' of weights in single-precision to overcome the limitations of representing small numbers – and loss of precision – when using half-precision. They show that doing so sees no loss in performance compared to single-precision training across a wide range of models and tasks, GANs included.

For PlaqueGAN, mixed precision training enables the insertion of self-attention modules to the generator and discriminator architectures at layers with higher spatial dimension features, as well as across multiple layers. This is all achievable on a single GPU with only 6GB of VRAM.

3.4 Self-Attention PlaqueGAN (SA-PlaqueGAN)

Since first being introduced for neural machine translation in the domain of Natural Language Processing (NLP) [2], attention mechanisms have become ubiquitous across multiple fields of machine learning. Self-attention in particular, which captures the response at a position within a sequence relative to every other position of the same sequence [77], has shown promise in computer vision tasks.

In GANs, self-attention was introduced by Zhang et al. [88] with their SA-GAN architecture. Since then, self-attention has become increasingly widespread in state-of-the-art models such as BigGAN [10] and numerous medical imaging GANs [63, 83, 1]. The theory behind the utility of self-attention in GANs is that they assist in modelling long-range dependencies in the spatial dimension of features. The majority of GAN backbones are convolutional networks that typically use small, e.g. (3×3), kernels which operate highly locally; requiring deeper networks to gain larger receptive fields. This can be especially problematic and inefficient for moderate to high resolution images. Through combining (rather than replacing) convolutional layers with self-attention, the GAN generator and discriminator are able to exploit high correlation in both local and distant spatial locations.

The self-attention module in SA-GAN is based on the non-local module of Wang et al. [79], which is itself an adaptation of dot-product attention introduced by Vaswami et al. [77] to computer vision. As a high-level overview, self-attention operates by:

1. Projecting the input feature maps onto three latent spaces – termed the Query, $Q \in \mathbb{R}^{(H\times W)\times d_k}$, Key $K \in \mathbb{R}^{(H\times W)\times d_k}$ and Value $V \in \mathbb{R}^{(H\times W)\times d_v}$ – through (1×1) convolutions. d_k and d_v are the dimensionality of the key and value, whilst H and W refer to the height and width of the input feature map.

2. Computing the pairwise similarities exhaustively between spatial locations through a matrix multiplication $S = QK^\top$ followed by softmax normalisation $\rho()$ on rows to form the attention maps.

3. These attention maps are then used to aggregate the values of V through a second matrix multiplication $D = \rho(S)V$,

4. before a (1×1) convolution projects the self-attention feature maps (O) to be of the same dimensionality as the input.

5. Finally, the self-attention feature map is scaled by a learnable parameter γ before summing with the input in a residual fashion: $Y = \gamma O + X$, where X is the input.

However, there is a significant drawback of this formulation due to high computational and memory complexity from the matrix multiplication $S = QK^\top$. As $S \in \mathbb{R}^{n\times n}$, where $n = (H \times W)$, the memory complexity scales quadratically with the input size $\mathcal{O}(n^2)$. This limits how many layers self-attention can be inserted into as well as which layers it can be used on, as layers with higher feature map spatial dimensions would require too much memory.

Recently, Shen et al. [71] provided an efficient reformulation of the dot-product attention, exploiting the associative property of matrix multiplication. They switch the order of matrix

multiplication from $(QK^\top)V \longrightarrow Q(K^\top V)$. They show that this is equivalent when scaling normalisation is used whilst approximately equivalent in the case of softmax normalisation (which we use), with negligible performance decrease. This comes at a substantial reduction in memory complexity to $\mathcal{O}(d_k \times d_v)$.

The efficient attention module allows us to integrate self-attention with PlaqueGAN at multiple layers of both generator and discriminator, using modest hardware with only 6GB of VRAM. We adapt the official implementation[1], with a single attention head and dimensionalities $d_k = d_v = 64$. As suggested in [88], we use the learnable scaling factor γ and initialise it to zero at the start of training. The notion behind this is to not hinder learning of convolution weights early on in training by gradually learning the more complex task of modelling long-range dependencies.

3.5 Synthetic Image Selection

All of our methods so far have focussed on improvements to the architecture and training of PlaqueGAN. However, there is a crucial element still missing that may ultimately determine whether PlaqueGAN (or SA-PlaqueGAN) is viable as an oversampling technique for imbalanced datasets: image selection. In this thesis, we refer to image selection as any process that alters the distribution of images generated by a GAN, either manually or automatically, by choosing specific instances to keep and discarding the remainder.

Particularly because the datasets involved are small, and intra-class variance can be greater than inter-class variance in the case of cored and diffuse plaques, it is prudent to expect that not all images generated by PlaqueGAN will be informative for oversampling minority Aβ morphologies. Some samples may be of low quality or with generator artifacts, or exhibit manifold intrusion, whereby the generated image more closely resembles a different class. On this basis, we reason that image selection can help prevent the use of these instances in the oversampling process and lead to better results on our downstream classification task.

Multiple works by Xue et al. [84, 83, 86] investigate sample selection in an effort to improve performance on downstream tasks involving histopathological datasets. Feature-based filtering is applied in [84], where the ℓ_2 distance between visual features – extracted by a pre-trained classifier – of synthetic images are compared to the class centroids of features for real images. Images that lie far away from their relevant class centroid are rejected. In [83], they expand on the feature-based filtering with a prior step of entropy-based filtering, rejecting low-entropy samples that cannot be confidently classified by the same pre-trained classifier. At each stage, the lowest scoring half of samples are rejected; meaning they must generate $4\times$ the number of images to arrive at the intended number of selected images. Finally, reinforcement learning (RL) for sample selection is employed in [86]. A transformer-based *controller* uses features extracted by the pre-trained classifier to output a set of actions to select or reject each generated image; a separate classifier is then trained in the loop using the augmented dataset and validation accuracy used as the reward to update the controller's selection policy. They find all methods improve upon augmentation without selection, with the RL method resulting in best downstream performance.

Bhattarai et al. [6] approach the problem similarly with a pre-trained classifier they call the *data sampler*. We refer to the data sampler as a *selection classifier* in this thesis. One of their

[1] https://github.com/cmsflash/efficient-attention

approaches sub-samples the generated data based on conditional class probability (class confidence score). The class confidence score for a generated sample $\hat{\mathbf{x}}$ is computed as $P(y_t|\hat{\mathbf{x}}, \theta_{sc})$ where θ_{sc} are the parameters of the selection classifier network and y_t is the intended target label. They subsequently rank the synthetic samples from highest to lowest class confidence and select the top-k to add to the original dataset.

We base our approach off of this class confidence method because of its intuitiveness and simplicity. However, we propose to use confidence thresholding as opposed to selecting the top-K results – which can be sensitive to the amount of data sampled. Let $\mathbf{y}_t = \{y_t^{(c)}\}_{c=1}^{3}$ be a vector of target Aβ morphologies whose entries are 1 for intended morphologies otherwise 0. We set a confidence threshold f_{thresh} to *reject* samples:

Reject if for any $c \in \{1, 2, 3\}$: $\begin{cases} P(y_t^{(c)}|\hat{\mathbf{x}}, \theta_{sc}) < f_{\text{thresh}} & \text{if } y_t^{(c)} = 1 \\ P(y_t^{(c)}|\hat{\mathbf{x}}, \theta_{sc}) \geq (1 - f_{\text{thresh}}) & \text{if } y_t^{(c)} = 0 \end{cases}$

All other samples are accepted.

Chapter 4

Experiments and Results

We now focus on answering the questions laid out in §1.2 through a series of experiments. We begin this chapter by describing the plaque datasets and summarising the evaluation metrics used for both GAN and oversampling experiments. For GAN evaluation, we justify our choice to not report certain popular metrics.

For the experiments, we first conduct an ablation study on PlaqueGAN using just cored plaques to examine how each contribution alters performance relative to a suitable baseline. Once a 'best' configuration is found, we apply PlaqueGAN to other minority plaque morphologies and quantitatively and qualitatively assess the synthesised plaques. We also explore the effects of adding self attention layers to PlaqueGAN.

Finally, PlaqueGAN oversampled datasets are compared to other oversampling methods in an offline setting when training a downstream classifier. A coarse hyperparameter search is carried out to determine a suitable strategy for selecting which synthesised images are added to the dataset.

4.1 Datasets

For both GAN experiments and augmentation experiments, we elected to use the dataset procured by Tang et al. [74], which was made openly available[1]. We provide an overview of the procurement process but for full details, please refer to [74] as well as their supplementary

[1] `https://zenodo.org/record/1470797`

Table 4.1: Summary of the dataset distribution by coarse-grained labels. *Negatives here refers to cases where the neuropathologist labels as negative or when there is insufficient presence of morphology within the tile.

Dataset	Total	Cored	Diffuse	CAA	Negative*
Training	61,370	2,141	48,123	2,227	9,761
Validation	8,630	381	7,487	126	732
Test I	10,873	98	10,480	6	330
Test II	6,229	458	5,604	82	355

Table 4.2: Summary of the dataset distribution by fine-grained labels. *Negatives here refers to cases where the neuropathologist labels as negative or when there is insufficient presence of morphology within the tile.

Dataset	Cored	Diffuse	CAA	Cored - diffuse	CAA-diffuse	Cored - CAA	Cored - diffuse - CAA	Negative*
Training	1,624	47,249	1,855	509	364	7	1	9,761
Validation	301	7,391	110	80	16	0	0	732
Test I	57	10,439	6	41	0	0	0	330
Test II	198	5,334	72	509	260	10	0	355

materials. The dataset contains a total of 63 de-identified Whole Slide Images (WSIs) from the archives of the University of California's Alzheimer's Disease Center Brain Bank. The WSIs are specifically taken from the superior and middle temporal gyrus (MTG) of the brain and have been stained with a 4G8 Aβ antibody. The case cohort span the full range of ABC score [32] for the classification of AD neuropathologic change, from *Not AD* to *High*. This includes the full range of CERAD scoring for neuritic plaques from 0 (*None*) to 3 (*frequent*).

43 of the raw WSIs were chosen to provide 256×256 pixel tiles of center-cropped, individual Aβ plaque candidates at $20\times$ magnification following stain normalization. A single neuropathologist provided labels for plaque candidates identified by bounding boxes, and overall tile labels automatically calculated by aggregating the amount of morphology present in each. This accounts for the presence of multiple Aβ morphologies in proximity. Training, validation and hold-out sets are created by splitting by WSI source 29–4–10 respectively, such that no two sets share tiles from the same source. Splitting by source allows reduced bias when assessing generalisation performance.

A summary of the datasets in terms of coarse-grained and fine-grained labels are provided in Table 4.1 and Table 4.2 respectively. It should be noted that for the training and validation sets, an intermediate CNN was trained to prioritise candidate tiles more likely to contain minority plaques. Even though this assistance increased the likelihood of the neuropathologist seeing minority plaques, they were still only identified in 22% of tiles [74]. This highlights the issue of finding and labelling the significantly rarer cored and CAA plaques.

One issue with the Tang et al. data is the insufficient number of CAA plaques within the test dataset (Test Set I in Tables 4.1 and 4.2). Consequently, results reported in [74] for final classifier performance could not reliably include metrics for CAA, and was instead implied through correlation between CNN scoring and CERAD scoring of a second hold-out set of 20 WSIs. We instead supplement Test Set I with Test Set II, procured for a recent paper by Wong et al. [81] and also made publicly available[2]. The dataset contains 43 MTG WSIs split over two phases; this time from three institutions, each using a different staining antibody. We use only WSIs from phase-one and UC Davis (11 total) to maintain consistency with the primary dataset, ensuring a common 4G8 staining antibody throughout and processed in the same way as [74] to yield individual 256×256 pixel tiles. A difference in this secondary hold-out set is that each tile has labels from five separate expert neuropathologists. Wong et al. find that models created using a consensus-of-two labelling – i.e. a label is considered positive in a tile if at least two of five neuropathologists are in agreement – produced the best AUPRC and AUROC compared to other consensus-of-n strategies and individual-expert models. On this basis, we select this consensus-

[2] `https://osf.io/xh2jd/`

of-two labelling as the ground truth for Test Set II, but keep it separate from Test Set I due to the difference in labelling. All WSIs in both datasets were scanned using an Aperio AT2 at 20× magnification.

We adhere to using only the training dataset for training the GAN, thus avoiding a potential situation where GAN memorisation leaks examples from validation and/or test examples into training, inflating the GAN-augmented performance.

4.2 GAN Evaluation Metrics

Reliable evaluation of GAN-synthesised data is essential to guide model architecture and training strategy such that they benefit intended downstream tasks. The rapid progress, and therefore adoption, of GANs in recent years has necessitated further research into evaluation measures. Although a handful have become more widely adopted than others, the field is still in its infancy and is unlikely to produce a scenario where one-metric-fits-all [8]. Rather than rely on a single metric, we follow the proposal by Theis et al. [75] to select a few that are tailored towards our application.

At a high level, evaluation of GANs target two aspects of sample generation: visual fidelity (in the case of image data) and diversity. The former assesses the quality (realism) of generated samples S_g, whilst the latter assesses whether they cover all modes of the real distribution.

4.2.1 Our Approach to Evaluation

Where generated images are intended solely for a well-defined downstream task, for which there is already a specific criterion to benchmark success, this criterion may perhaps be the most important evaluation measure of the GAN generator [24]. It indicates how well the generator can be deployed in practice, i.e. its data augmentation utility [8]. As an example, in [45] classification accuracy is the task-specific criterion; the premise being that synthetic samples are likely to be both high quality and diverse if a classifier trained on a mixture of real and generated data achieves higher accuracy than one trained only on real data. In our particular case of balancing data for training, we instead seek improvement in the Area Under Precision Recall Curve (AUPRC) for the multilabel classification setting (§4.3). However, such indirect methods may be uninformative for diagnosing failure modes of the generator and the image generation process. It also has a high computational and time cost, making it unsuitable for tracking model development under resource constraints. We therefore turn to more efficient measures during model development, selecting those likely to serve as good proxies for downstream performance. We use the desiderata for GAN-generated neuropathology images discussed in [ref. introduction/ background section] to help inform the most suitable metrics.

Both *Quantitative* and *Qualitative* measures are selected as each have their advantages and disadvantages. Human evaluation is still of importance because many quantitative metrics rely on some variant of an ImageNet [16] pre-trained classifier to extract features from images to compare distributions. Using these classifiers out-of-the-box on non-natural images – including those in the medical domain – is yet to be thoroughly validated [59, 87], and may carry a risk of not correlating well with human judgment. Furthermore, some quantitative metrics are sensitive to sample size,

Table 4.3: Averaged baselines for quantitative evaluation metrics considered in this paper calculated on real samples. For all metrics, 30× repeat calculations were done by dividing real samples from the training set randomly into two equal, disjoint sets before calculation. *For cored-diffuse and CAA-diffuse morphologies, small dataset sizes may mean PRDC baselines are less reliable. Metrics considered have baselines consistent across Aβ morphologies unlike FID (shown for comparison).

Morphology	Precision	Recall	Density	Coverage	1NN Acc. (real)	1NN Acc. (gen)	KID	FID_{2048}	FID_{768}
Cored	0.850	0.853	1.002	0.969	0.500	0.502	0.000	18.0	0.039
CAA	0.893	0.896	0.992	0.968	0.504	0.499	0.000	33.2	0.074
Cored-diffuse*	0.909	0.900	1.015	0.970	0.495	0.502	0.000	35.8	0.092
CAA-diffuse*	0.882	0.883	1.010	0.973	0.489	0.507	0.000	67.6	0.192

and may even be unreliable when applied in a low-data regime. On the other hand, qualitative metrics are largely restricted to assessing only fidelity, disregarding diversity in all but the most extreme cases where the data either has few modes or the generator suffers from severe mode collapse. When selecting quantitative metrics, we establishing desiderata for the GAN generated neuropathology images can help inform which are most suitable.

A comprehensive review of a wide range of GAN evaluation metrics can be found in [8, 9].

4.2.2 Quantitative Metrics

Unlike other popular generative models such as VAEs, GANs lack an objective function that translates well to a quantitative measure of performance. For the original GAN in [23], estimates of the average log-likelihood are done via Gaussian Parzen windows [61]. However, the authors acknowledge the high variance and poor performance of the estimate in high dimensional spaces. Theis et al. [75] confirm the unsuitability of Parzen windows, demonstrating that the likelihood estimates produced can be far from a model's true likelihood and lead to inconsistent ranking when compared to other estimates. The suitability of average log-likelihood is more generally called into question due to independence from visual fidelity. Such issues meant that practitioners often found qualitative, visual assessments to be more reliable in the earlier days of GAN research.

More recently, numerous sample-based GAN evaluation measures have emerged that correlate better with human perception. We focus on a family of measures that share a common processing pipeline, shown in [ref. fig]. In all the metrics that follow, features for real samples S_r and generated samples S_g are first extracted, typically using a pre-trained CNN classifier. The importance of operating in the convolutional feature space as opposed to the pixel-space is empirically justified in [82], the reasoning being that small transformations can lead to wild changes in pixel-space distances even if they do not alter the perceptual meaning of the images. Choice of the feature extractor itself does not appear as important [82], but it is sensible to use those originally proposed for consistency when comparing results.

Special mention should be given to the Inception Score (IS) [67] as the metric that popularised use of an ImageNet pre-trained classifier. Unlike measures we consider, IS operates in the softmax probability space of the eponymous Inception V3 [73] network rather than the convolutional space. IS is the (exponentiated) Kullback-Liebler Divergence (KL) between the conditional and marginal

class distributions over generated data:

$$IS(P_g) = \exp\left(\mathbb{E}_{\mathbf{x}\sim P_g}[KL(p(y|\mathbf{x}) \parallel p(y))]\right), \tag{4.1}$$

where $\mathbf{x}$ is an image sampled from the distribution of the generator P_g, $KL(P \parallel Q)$ is the KL Divergence between two probability distributions P and Q, $p(y|\mathbf{x})$ is the class-conditional distribution for the image, and $p(y) \approx \frac{1}{N}\sum_{n=1}^{N} p(y|\mathbf{x}_n = G(\mathbf{z}_n))$ is the marginal class distribution. IS rewards images that are easily classifiable – $p(y|\mathbf{x})$ is low entropy – and equal representation of all classes across a sample of images – $p(y)$ is high entropy.

Limitations of IS such as a lack of comparison between the distribution of generated samples and real samples, sensitivity to image resolution [58] and sensitivity of the softmax space to datasets other than ImageNet [4, 82] have seen it fall out of favour with researchers, replaced by some of the measures below.

Fréchet Inception Distance

A second notable mention is the Fréchet Inception Distance (FID), introduced by Heusel et al. [29], which improves on IS by comparing generated samples to real samples. Furthermore, it uses the final pooling layer of the Inception V3 network to extract vision features in the convolutional space instead of the softmax space, making it more robust to different datasets[3]. With ϕ denoting the function mapping images to the Inception features, extracted features from real images $\phi(\mathbf{x}_r)$ and generated images $\phi(\mathbf{x}_g)$ are modelled as continuous multivariate Gaussians with respective empirical means $\boldsymbol{\mu}_r, \boldsymbol{\mu}_g$ and empirical covariances $\boldsymbol{\Sigma}_r, \boldsymbol{\Sigma}_g$. The Fréchet distance [21] between two multivariate Gaussian distributions is then given by [19]:

$$FID(P_r, P_g) = |\boldsymbol{\mu}_r - \boldsymbol{\mu}_g|^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{\frac{1}{2}}\right) \tag{4.2}$$

The authors show empirically that FID responds correctly to image disturbances that affect human perception of quality, and in a separate study [50] it is demonstrated that FID can detect intra-class mode dropping; something that IS cannot recognise as it only detects when specific *classes* are dropped rather than *modes*.

Nevertheless, FID is not perfect. FID is known to be biased as a function of the number of samples N used as well as the architecture of the generator [13], sensitive to the image processing library and compression used [60], and is limited in expressive power by assuming distributions to be multivariate Gaussians [7]. Due to the bias, a large number of samples (typically $N = 10,000$ or $N = 50,000$) are required for calculation. This is often not possible for smaller datasets such as ours, where we are further limited as the number of real samples for the minority Aβ plaques are fewer than the feature dimension, $\phi(\mathbf{x}) \in \mathbb{R}^{2048}$. This means the covariance matrix is rank deficient and taking the square root in Equation (4.2) can result in complex numbers or NaNs. A proposed solution might be to extract spatial features, for example just before the Inception network's auxiliary classifier, and apply global spatial average pooling to reduce them to a dimension lower than 2048. In the case of global average pooled features before the auxiliary classifier the dimension is 768. However, correlation with human judgment has yet to be confirmed

[3]This is only relative to IS. FID is likely still sensitive to non-ImageNet datasets

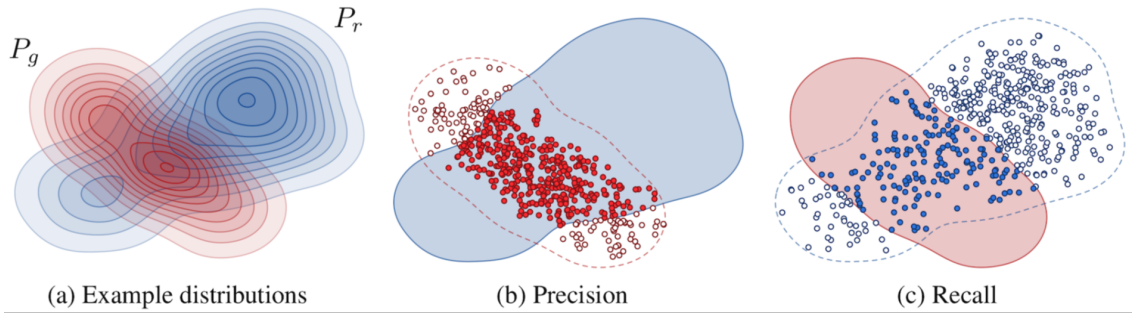


Figure 4.1: Illustration of Precision and Recall from [43]. (a) Example illustrative distributions of real data P_r and generated data P_g. (b) Precision measures the proportion of generated images that lie inside the manifold of real data. (c) Recall measures the proportion of real images that lie inside the manifold of generated data.

for these alternative features. An interesting observation we find for both FID_{2048} and FID_{768} in 4.3 when establishing baselines based on real data, is that there is no consistent target across the different Aβ morphologies. For these reasons, we do not report FID in our experiments.

Kernel Inception Distance

Bińkowski et al. [7] propose a metric similar to FID, but with several advantages. The Kernel Inception Distance (KID) also uses Inception V3 features from the final pooling layer, but measures the distance between two samples not by assuming they are from a parametric distribution like FID, but as the square of the Maximum Mean Discrepancy (MMD) [25] between the features. This can be calculated efficiently using the kernel trick. For two samples $X = \{\mathbf{x}_i\}_{i=1}^m$, $Y = \{\mathbf{y}_j\}_{j=1}^n$ drawn i.i.d. from distributions P, Q, the squared MMD can be estimated by [7]:

$$MMD^2(X,Y) = \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{mn} \sum_{i}^{m} \sum_{j}^{n} k(\mathbf{x}_i, \mathbf{y}_j) \quad (4.3)$$

For computation of KID, it is common to take $m = n = N$. The authors propose a cubic polynomial kernel $k(x, y) = (\frac{1}{d} x^\top y + 1)^3$, where $d = 2048$ is the dimension of the Inception V3 features, allowing the comparison of skewness in addition to mean and variance. This gives KID a richer representation to discriminate the distributions than FID. They also demonstrate that KID is unbiased as a function of the number of samples N, which is beneficial to our studies where we wish to compare how the GAN performs for each Aβ morphology (which have different dataset sizes). We adapt the official implementation code[4] and calculate KID with N equal to the size of the training dataset.

Precision and Recall, Density and Coverage

Both FID and KID are single-value metrics which can obfuscate the two objectives of high fidelity and diversity, making it difficult to diagnose or compare model performance fairly. Lucic et al. [50] recognise the similarity between these objectives and the commonly used *Precision* and *Recall*

[4] `https://github.com/mbinkowski/MMD-GAN`

metrics for discriminative models, and build the insight that *Precision* relates to the quality or fidelity of generated samples while *Recall* assesses how much of the real distribution is covered by the learned distribution. These concepts are illustrated in Figure 4.1. Sajjadi et al. [66] approach Precision and Recall through K-means clustering of samples embedded in the Inception V3 feature space, reducing to a one-dimensional problem where histograms of real and fake image cluster assignments can be compared. However, drawbacks to this method are that its reliance on the k-means algorithm makes it sensitive to initialisation and it outputs a continuous set of metrics, making it difficult to decide which to use [57]. The work of Sajjadi et al. is simplified through Improve Precision and Recall in [43], whereby the manifolds of the real and learned probability distributions are estimated by forming hyperspheres around samples whose radii are determined by the distance to their k-nearest neighbours.

Naeem et al. [57] argue that Improved Precision and Recall still has room for improvement. Predominantly, they highlight that the construction of the hyperspheres in [43] does not take into account the relative density of samples in proximity. As a result, it is sensitive to outlier data, building hyperspheres with large radii that overestimate the true manifold; this leads to an inflation of Recall and undesirable behaviour whereby Precision can be increased by generating samples around real outliers. They introduce *Density* and *Coverage* metrics to tackle these issues, where Density rewards the generation of samples in dense regions of the real manifold. Coverage is designed to alleviate the issue of constructing a manifold of generated samples, where there tend to be more outliers. Instead, the manifold is constructed only on real samples and the proportion of real samples whose hypersphere contains at least one generated sample is calculated.

For both improved precision and recall and density and coverage, we use $k = 5$ nearest neighbours as suggested in [57] and use the official implementation[5] of [57] to calculate both. We use equal sample size for both real and generated data, and set it equal to the size of the training dataset.

Leave One Out 1-Nearest Neighbour Classification Accuracy

Thus far, metrics considered all suffer from one flaw: they can be deceived by a memory GAN that replicates the training dataset (i.e. perfect overfitting). Xu et al. [82] support the use of 1-nearest neighbour classifier in GAN evaluation. The general procedure is to first obtain sample sets of equal size from real and generated samples. Next, a 1-NN classifier is trained on the two sample sets, where real samples are assigned positive labels and generated samples negative labels. The leave-one-out accuracy of the classifier is then calculated, for all samples as well as separately for real and generated samples.

The intuition behind the 1-NN accuracy is that, in an ideal scenario, real and generated images will have nearest neighbours that are equally likely to come from either class; the LOO accuracy should therefore be around 50%. In the case of severe overfitting, the nearest neighbours of real samples will be generated and vice versa; thus the accuracy will fall below 50%, with 'perfect' overfitting yielding an accuracy of 0%. Conversely, an accuracy above 50% signifies that the distributions are different as they exhibit some degree of separability.

Finer-grained diagnosis is possible with the 1-NN Accuracies through comparison of accuracies

[5] https://github.com/clovaai/generative-evaluation-prdc

for real and generated images. Xu et al. note that in the case of mode dropping, nearest neighbours of both real and generated images are typically generated images: this results in a 1-NN Accuracy (real) that is significantly lower than 1-NN Accuracy (gen.). We use the official implementation[6] of [82].

Baseline metrics

We calculate baselines for each of the intended quantitative metrics to check that they are sensible for our dataset, as well as to provide indicative targets to aim for and compare against. We calculate the baselines as an average of 30× repeat calculations; for each repeat, we divide the real samples from the training set randomly into two disjoint sets of equal size and compare these two sets. The results in Table 4.3 show that all metrics considered give baselines that can be reasonably expected. For example, KID is extremely small, and the 1-NN accuracies are around their optimal value of 0.5. For comparison, we also calculate FID using the toolbox from Seitzer [69] both on the original $2048-d$ features as well as the reduced $768-d$ features of the Inception Network. We see that FID varies significantly between datasets which makes comparison in model performance between datasets more difficult.

4.2.3 Qualitative Metrics

Visual Turing Test

Uncertainty over the reliability of automated metrics for evaluating generative models in medical imaging has meant that visual Turing tests are still highly relevant [59, 76]. The basic format of such tests is to show human evaluators a number of randomly mixed real and generated images and ask that they score them as real or fake. These results are then typically aggregated into scores of real/fake recognition rate (% correctly identified) or deception rate (% incorrectly identified). Unlike simple datasets of natural images that can be scored by most human evaluators, medical images require experts in the field to carry out the evaluation; often resulting in tests being more expensive and less able to capture intra- and inter-rater variation due to resource limitations.

Visual Turing tests often lack a standardised framework, each researcher introducing idiosyncrasies which can make comparisons across papers difficult. Zhou et al. [92] recognised this shortfall and established two variants of what they call Human Eye Perceptual Evaluation (HYPE) tests, $\text{HYPE}_{\text{time}}$ and HYPE_{∞}. In $\text{HYPE}_{\text{time}}$, the minimum time needed for humans to classify images as real or fake is determined by showing evaluators images under adaptive time constraints. A longer $\text{HYPE}_{\text{time}}$ implies a generative model that can produce more realistic outputs. HYPE_{∞} is a simplified and more cost-effective method that removes the time constraint and instead measures the deception rate for real and fake images. Participants are shown an equal number of real and fake images (50 of each is proposed) and a score of 50% indicates that evaluators are unable to discriminate between real and fake images any better than chance. The recommendation is for scores to be aggregated over 30 evaluators as an appropriate tradeoff between the precision and cost of obtaining the results.

We adopt HYPE_{∞} for our visual Turing test. However, we are unable to meet the recommendation for 30 participants with access to only a single expert neuropathologist. We propose two

[6] `https://github.com/xuqiantong/GAN-Metrics`

changes to the framework to reduce the impact of these limitations:

1. We conduct 2× repeats of the test. For each test, a disjoint set of real and fake images are shown to the expert neuropathologist such that no two tests contain the same images.

2. We increase the number of real and fake images shown to the evaluator from 50 to 75. The first 50 images comprised of 25 each real and fake are then discarded before using the remaining 50 each to calculate the HYPE_∞ score. Our reasoning for this is that neuropathologists may not be as accustomed to viewing single 256×256 pixel tiles of individual plaques, and a *bedding-in* phase might be helpful to familiarise them with the format.

As only 2× repeats are conducted, we report the range of the results instead of means and standard deviations. All images shown to the evaluator are generated by randomly sampling from the latent space without any further curation. We use the free online annotation tool CVAT[7] [70], whereby images are uploaded to a web server for the neuropathologist to annotate.

Nearest Neighbours

An intuitive method for detecting memorisation, and one used often throughout the brief history of GANs, is to compare generated samples with their corresponding nearest neighbours in the real training set. For images, early use of this technique calculated nearest neighbours in the pixel space. For the same reasons discussed in §4.2.2, distances in the pixel space are unsuitable and it is now more common to measure distances in the convolutional feature space of pre-trained ImageNet classifiers. LPIPS (§3.3) has also been used as a distance measure between images for nearest neighbours recently [49, 91]. We see later in §4.5.2 that using LPIPS for reconstruction loss provides significant improvement in the speed of GAN training and quality of generated images. We therefore consider that LPIPs might capture perceptually meaningful differences between neuropathology images, making it suitable for finding training set images that have been memorised or overfit to. We display the top 4 nearest neighbours beside each generated image to further our chances of spotting overfitting.

Latent Space Interpolation

A second method to visually detect overfitting or memorisation of the generator is to review a series of generated images interpolated gradually between two arbitrary generated images. This is done by first sampling two points $\mathbf{z}_1,\ \mathbf{z}_2 \sim \boldsymbol{\mathcal{Z}} \in \mathbb{R}^d$ from the latent space of the generator and then using either linear or spherical linear interpolation to generate intermediate images from latent vectors connecting the two points. The intuition is that a well-generalised generator will exhibit *smoothness* or *continuity* as it transitions between points in the latent space. On the other hand, a generator that overfits or simply memorises training data will result in sharp discontinuities as many latent points map to few images. As the prior for the generator is often a high dimensional Gaussian, [80] advocates use of spherical linear interpolation to ensure that interpolated latent vectors are more faithful to the prior. We find agreement with this line of thought during our experiments and therefore apply spherical linear interpolation always.

[7] https://cvat.org/

4.3 Oversampling Evaluation Metrics

The hold-out test datasets we use exhibit the same skew as the training datasets. Thus, the use of common evaluation measures such as classification accuracy are potentially misleading for imbalanced data.

Instead, we focus on the area under the precision recall curve (AUPRC), which provides a more meaningful comparison between methods for imbalanced data; particularly as we are more concerned with correctly classifying the positive cases of minority classes.

4.4 Computing Environment

One goal of the project is to establish whether viable deep generative models can be trained on a modest computing environment in reasonable time. This may be important for laboratories that do not have the resources or access to high-performance deep learning hardware. For all architectures explored, we train on a Lenovo Legion Y540 laptop with:

- a 12 core Intel Core i7-8750H CPU @ 2.20Ghz
- a single NVIDIA GeForce GTX-1060 GPU with 6GB GDDR5 VRAM and 1280 CUDA Cores
- 16GB of RAM
- 64-bit Windows 10 Operating System

All models for both GAN experiments and Augmentation experiments are implemented in PyTorch [62].

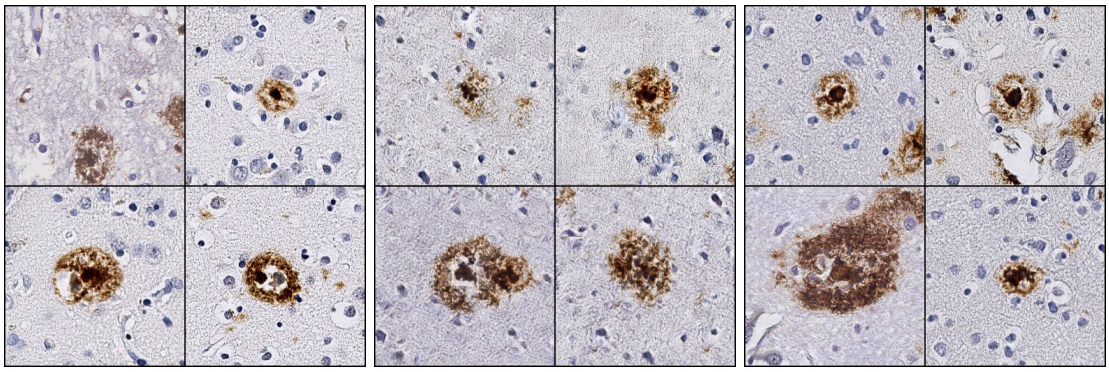
4.5 PlaqueGAN Experiments

In this section, we detail our experiments on PlaqueGAN and report any key results. The main purposes of these experiments is to confirm that the model and training procedures proposed in §3.2 lead to improvements in both image quality and diversity over the baseline architecture; check whether performance is consistent across training datasets of various sizes; and validate that generated images of plaques display the correct morphological hallmarks through qualitative assessment.

We exclude experiments involving tiles that include both cored and CAA morphologies (7 samples available), and all three morphologies (only 1 sample available) due to insufficient size of these training datasets.

4.5.1 Training Procedure

We follow the training procedure for FastGAN [49] in the interest of time, and because the authors have demonstrated that these achieve convergence across multiple datasets with multiple resolution images. Training uses the ADAM optimiser [40] with momentum coefficients $\beta_1 = 0.5,\ \beta_2 = 0.999$ and $\epsilon = 10^{-8}$, and a learning rate $\alpha = 0.0002$ for both generator and discriminator. For all experiments, weights of the generator and discriminator are updated via a standard alternating



(a) Real **(b)** MSE reconstruction loss **(c)** LPIPS reconstruction loss

Figure 4.2: Effect of autoencoding reconstruction loss on generated core plaque quality after 50k training iterations. (a) Real examples (b) Generated with the baseline using MSE reconstruction loss for auxiliary discriminator task (c) When using LPIPS reconstruction loss for auxiliary discriminator task. LPIPS reconstruction loss results in faithful depictions of the background tissue which MSE reconstruction loss is unable to achieve.

gradient strategy. This also includes tracking the exponential moving average (EMA) of generator weights [85] with discount factor $\gamma = 0.999$. Standard augmentations of horizontal and vertical flipping are applied randomly to all images, whilst Differentiable Augmentation (DiffAugment) [91] is used to transform both real and fake images before they are shown to the discriminator. We use the default settings of DiffAugment[8] used in FastGAN training with a policy including random cutout, translation and colour jitter.

The number of training iterations varies by experiment: due to time constraints, we use 50k iterations – equivalent to 400k real and fake images seen by discriminator – for the first experiment concerning cumulative contributions of model and training improvements. This is far fewer than typical numbers seen by a discriminator (e.g. $\geq$ 10M images in [37, 36]), so convergence is not guaranteed. Therefore we double the number of iterations to 100k to assess whether the generator continues to improve at the end of the experiment; based on this result, we run the second experiment with the finalised PlaqueGAN model and training settings separately on the four minority plaque datasets. For the final PlaqueGAN settings, training time is approximately 12hrs for 50k iterations or 24hrs for 100k iterations using the computing environment described in §4.4.

4.5.2 Experiment on Proposed Improvements

We build up PlaqueGAN from the baseline [49] incrementally with our proposed alterations and additions to demonstrate their utility. This provides appropriate granularity for understanding the improvements without having to exhaustively try all combinations of modifications. For all of the configurations, we use only the cored plaque dataset. We do so because this is one of two datasets with a moderate number of samples, and also because cored plaques are arguably most important to deep learning pipelines relating to classification of Alzheimer's Disease.

The progression of PlaqueGAN is tracked using the quantitative evaluation metrics from §4.2.2 and recorded in Table 4.4. Dissecting these results, we make the following observations:

[8] Originally from `https://github.com/mit-han-lab/data-efficient-gans`

- **LPIPS is superior to MSE for reconstruction loss.** With this single change to the baseline, metrics improve considerably across the board. This appears most in improvements in proxies for fidelity, with config B holding the highest Precision and Density scores. Visual comparison of cored plaques produced by both config A and B after 50k iterations in Figure 4.2 demonstrates noticeable improvement in image quality when using LPIPS as the auto-encoding reconstruction loss for the discriminator auxiliary task. This is clearest when comparing the background tissue, where glial cells and neurons are much more similar to real images when using LPIPS.

- **Improving fidelity is easier than diversity**. This is a well-known issue for GANs as common failure modes described in §2.4.2 can produce high fidelity results with poor diversity. Although config B has the highest Precision and Density, low Recall and a high 1-NN Accuracy (gen.) implies that most generated images are grouped closely around a few modes of the real distribution. Low diversity is unlikely to prove successful when augmenting datasets for the downstream task.

- **Stochastic noise and Minibatch Standard Deviation work as intended.** Both of these changes were introduced explicitly to encourage the generator to synthesise diverse images that cover more modes of the real distribution. When each of these changes are added, there is noticeable gain in Recall and 1-NN Accuracy (gen.) metrics.

- **The baseline generator capacity was impeding performance.** Increasing from one to two convolution layers per generator block with config E led to substantial improvement in diversity metrics such as Recall and 1-NN Accuracies (particularly for generated samples). This suggests that the baseline generator model was too restrictive to learn the wider distribution of real images. Although this comes at both a memory and time penality, it is justified by the performance improvement.

It should be noted that all of these configurations fit entirely on the 6GB GPU memory without requiring AMP.

4.5.3 Experiments on All Datasets

We train the remaining CAA, cored-diffuse and CAA-diffuse datasets with PlaqueGAN config H, evaluating the same set of quantitative metrics. We switch from reporting the average of the final three saved model iterations to selecting a 'best' iteration, as these model weights are carried forward to generate images for qualitative assessment as well as the data oversampling experiments in §4.6. With a number of metrics to consider, we simplify the iteration selection process with a heuristic we call GAN Training Progress Index (GTPI) that roughly tracks the progress of GAN training:

$$GTPI = (1 - |2Acc_{1NN(\mathbf{x}_r)} - 1|) \times (1 - |2Acc_{1NN(\mathbf{x}_g)} - 1|) \times \exp(-10KID), \tag{4.4}$$

Table 4.4: Summary of results showing model evolution when adding modifications incrementally to the baseline architecture and training procedure. Experiments were carried out on cored plaques only and the discriminator saw 800K images during training except for configuration H. For all metrics, we report the average of the last 3 saved iterations of the model to reduce the influence of fluctuations during training. Best results for each metric are highlighted in **bold**, excluding configuration H. (↑) indicates higher score is better, (↓) indicates lower score is better.

Configuration	Precision (↑)	Recall (↑)	Density (↑)	Coverage (↑)	1NN Acc. (real)	1NN Acc. (gen.)	KID (↓)
A) Baseline [49]	0.831	0.015	0.795	0.280	0.903	0.999	0.0662
B) + LPIPS loss	**0.940**	0.275	**1.632**	**0.892**	**0.558**	0.928	0.0207
C) + Stochastic noise	0.928	0.351	1.610	0.885	0.576	0.894	0.0157
D) + Improved Gradient Flow	0.902	0.301	1.266	0.843	0.622	0.924	0.0262
E) + Dual Discriminator	0.895	0.403	1.227	0.864	0.611	0.854	0.0219
F) + Increased G Capacity	0.887	0.566	1.095	**0.892**	0.599	0.783	0.0211
G) + Minibatch Stdev	0.832	**0.662**	0.918	0.867	0.613	**0.726**	**0.0112**
H) + Train 2× longer	0.890	0.628	1.204	0.937	0.548	0.708	0.0070

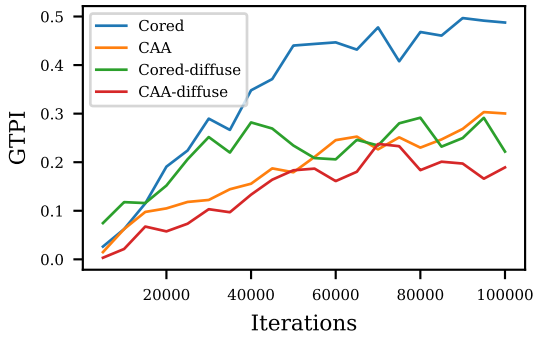


Figure 4.3: Evolution of the GAN Training Progress Indicator over 100k iterations of training for each of the Aβ morphologies.

where the two expressions in brackets transform the 1-NN accuracy for real and generated images respectively, such that a bracketed value of 1 is achieved when the metric is at an ideal 0.5 and 0 if the learned distribution completely overfits ($Acc_{1NN} = 0$) or is completely separable ($Acc_{1NN} = 1$). The GTPI is bounded between 0 and 1, where a value of 1 is the ideal target when $KID = 0$ and both 1-NN accuracies are 0.5. We choose to use these three metrics as we found that 1-NN accuracies are closely related to the Recall and Coverage metrics, and KID incorporates some notion of quality. We explored other combinations of metrics that also include Precision and Density, but found these exhibited more noise.

The training curves for GTPI are plotted in Figure 4.3 with an accompanying breakdown of individual metrics at the iteration with highest GTPI displayed in Table 4.5. The GTPI plot shows that PlaqueGAN training improves mostly monotonically for cored and CAA datasets in comparison to cored-diffuse and CAA-datasets, which peak earlier in training. Cored PlaqueGAN has a GTPI up to twice that of the other plaques; the varied difficulty in training the datasets may be linked to the dataset size, or perhaps because aspects of the model architecture or training procedure are more suited to some morphologies. The steady upward trajectory of GTPI for CAA is indicative that training has not yet converged, and extending could be beneficial.

Table 4.5: Main results showing final metrics for PlaqueGAN when trained separately on the four minority plaque datasets. In all cases, the discriminator saw 800k real and fake images during training, the equivalent of 100k iterations with a batch size of 8. We report metrics at the point in model training that indicates best performance according to the heuristic in Equation (4.4).

Morphology	Iteration	Precision (↑)	Recall (↑)	Density (↑)	Coverage (↑)	1NN Acc. (real)	1NN Acc. (gen.)	KID (↓)
Cored	90,000	0.890	0.637	1.214	0.940	0.531	0.714	0.0075
CAA	100,000	0.836	0.651	0.991	0.895	0.674	0.750	0.0081
Cored-diffuse	80,000	0.876	0.729	1.013	0.906	0.739	0.674	0.0156
CAA-diffuse	70,000	0.868	0.659	1.058	0.942	0.731	0.703	0.0296

Table 4.6: Results of the HYPE_∞ visual Turing test of PlaqueGAN for the four minority Aβ morphologies.

Aβ Morphology	$\textbf{HYPE}_\infty$(%)	Fakes Error (%)	Reals Error (%)
Cored	52.0 – 54.0	22.7 – 34.7	69.3 – 85.3
CAA	42.7 – 50.0	30.7 – 49.3	42.7 – 50.7
Cored-diffuse	41.3 – 49.3	34.7 – 45.3	37.3 – 64.0
CAA-diffuse	48.0 – 50.7	22.7 – 32.0	64.0 – 78.7

4.5.4 Qualitative Evaluation

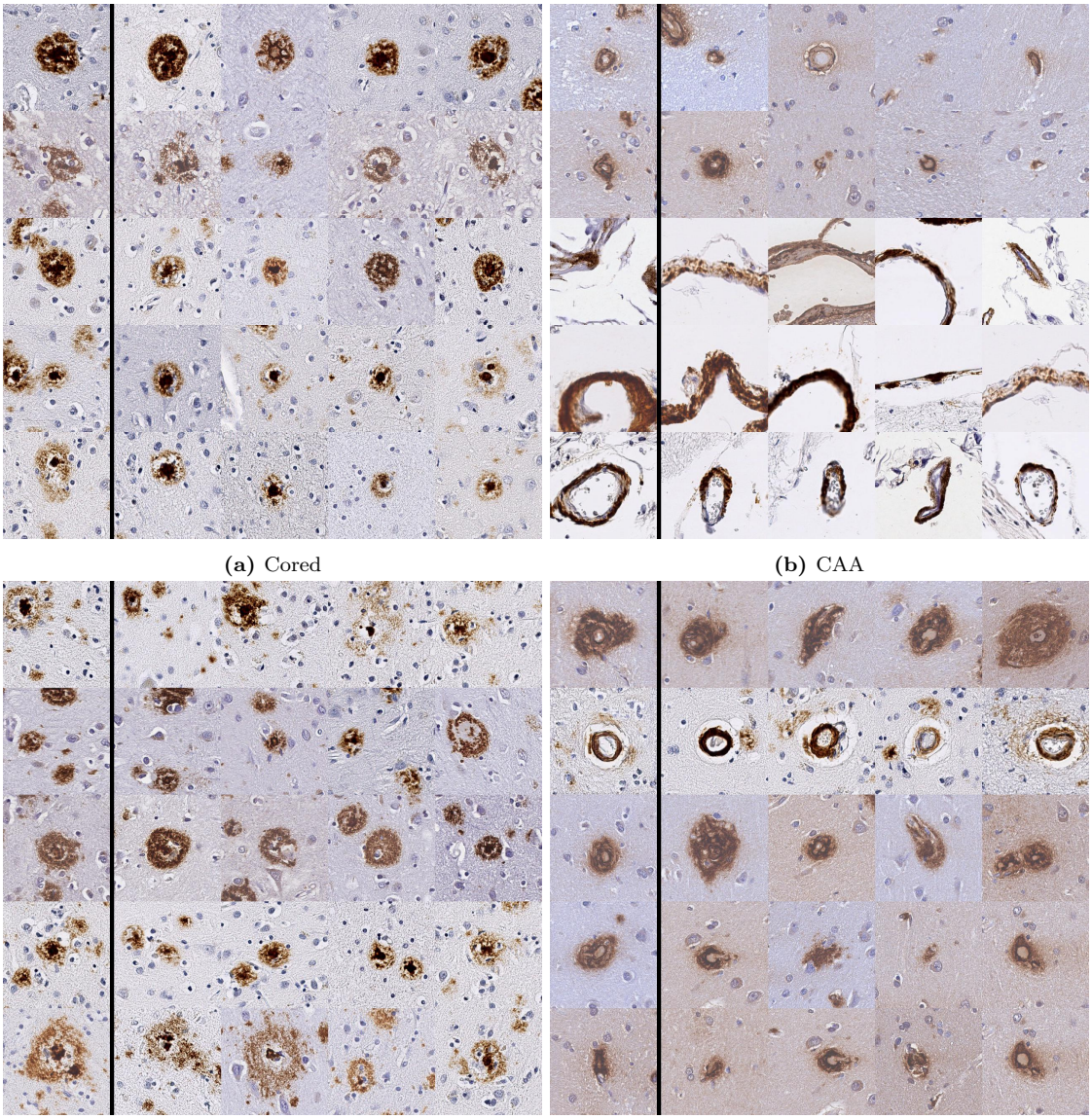
We supplement the quantitative results of §4.5.3 through visual inspection of images produced by PlaqueGAN for the individual plaques.

Visual Turing Test

The results of the HYPE_∞-style Turing tests on PlaqueGAN are summarised in Table 4.6. Scores for the four minority plaque combinations are all above 40%, signifying that the neuropathologist found it difficult to distinguish between samples generated by PlaqueGAN and real samples. PlaqueGAN is thus able to synthesise high fidelity images. When the score is broken down by the fake and real deception rates, a pattern emerges that suggests the neuropathologist tended towards labelling images as fake, regardless of whether the image is real or fake. This is particularly the case for tiles with only cored plaques and tiles where both CAA and diffuse plaques are present. Potential reasons for this are discussed further in §5.3.

Nearest Neighbours

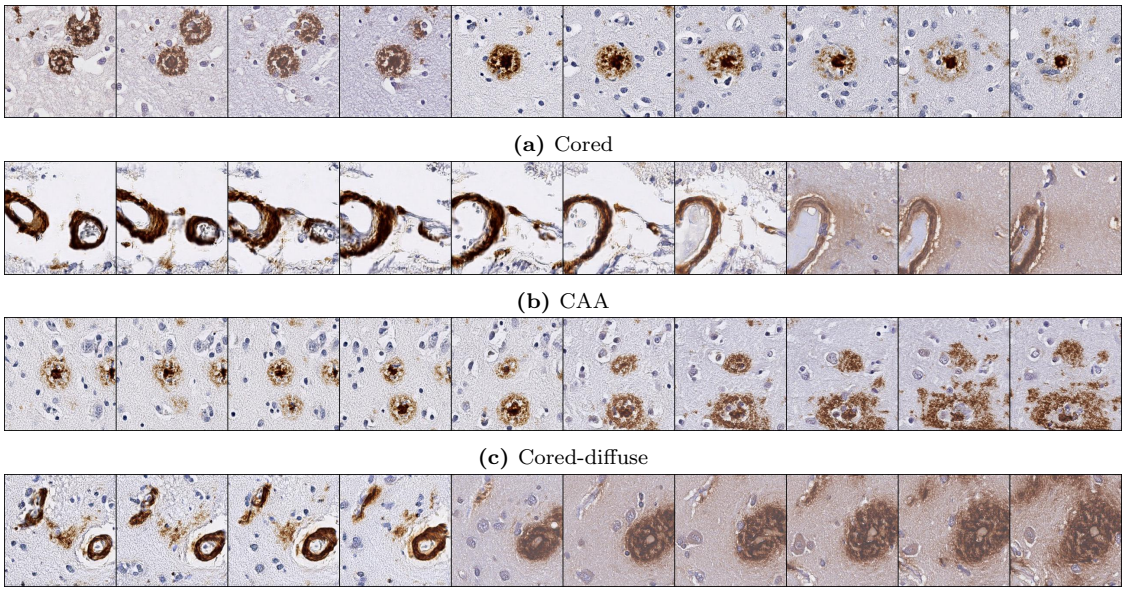
Closest neighbours, measured by LPIPS, between generated and real plaque images from training data in the feature space of a pre-trained VGG16 are displayed in Figure 4.4. The pairwise comparison between the images suggest that PlaqueGAN is able to learn the general morphological features of real images for the Aβ, gray matter tissue, as well as the colour characteristics. We see that the generated images are sufficiently different from their closest neighbours, giving added confidence that the GAN is not memorising the training data.



(a) Cored **(b)** CAA

(c) Cored-diffuse **(d)** CAA-diffuse

Figure 4.4: Nearest neighbours in VGG16 feature space according to LPIPS distance (a) Cored (b) CAA (c) Cored-diffuse (d) CAA-diffuse plaques. For each row in the grids, the leftmost image is the generated plaque, with the remaining columns showing the four closest images in the training set (rightmost corresponding to furthest from generated sample). Best viewed digitally.



(a) Cored

(b) CAA

(c) Cored-diffuse

(d) CAA-diffuse

Figure 4.5: Example images generated through spherical linear interpolation between two latent vectors $\mathbf{z}_1$, $\mathbf{z}_2$. (a) Cored (b) CAA (c) Cored-diffuse (d) CAA-diffuse. The first image in each row is generated from $\mathbf{z}_1$ and the last image from $\mathbf{z}_2$. Best viewed digitally.

Latent Space Interpolation

In Figure 4.5, we present example traversals between two points in the PlaqueGAN latent space via spherical linear interpolation. These examples demonstrate the smoothness of transitions, providing further qualitative evidence that PlaqueGAN has not mode collapsed or memorised the training data. It is interesting to see the gradual change to the Aβ morphology, addition/ subtraction of plaques, and gradual change to tissue (particularly for Figure 4.5b). Stain colouration changes also appear to be adequately handled (Figures 4.5c and 4.5d), though this perhaps occurs more abruptly. Mapping out the latent space visually gives early indication that PlaqueGAN's latent space has at least some degree of disentanglement.

4.5.5 Experiments with Self-Attention (SA-PlaqueGAN)

We investigate whether PlaqueGAN can be extended further by introducing self-attention to one or more layers in the generator and discriminator. We use config G from Table 4.4 as the baseline for these experiments, i.e. the final PlaqueGAN model but with training reverted to 50k iterations in the interest of time. All experiments are carried out on the cored plaque training set only, in line with experiments in §4.5.2.

We experiment with efficient self-attention modules inserted at different resolution blocks of the generator, from 16×16 to the highest resolution of 256×256; and mirror this for the main discriminator, D_1. With the aid of AMP training, we extend self-attention modules to multiple resolution blocks from 32×32 – 128×128; as well as to the secondary discriminator D_2 at 128×128 and 16×16 spatial resolution of features.

We track the quantitative metrics for self-attention in individual layers in rows A) to F) in

Table 4.7: Summary of results when modifying PlaqueGAN configuration G with self-attention. In configurations B to G, attention is applied in both the generator G and the main discriminator D_1 at the same resolutions. In configuration H, attention is also further added the secondary discriminator D_2 at the 128×128 and 16×16 resolution blocks. Best results for each metric are highlighted in **bold**.

Attention Layers	Precision (↑)	Recall (↑)	Density (↑)	Coverage (↑)	1NN Acc. (real)	1NN Acc. (gen.)	KID (↓)
A) PlaqueGAN-G (None)	0.832	**0.662**	0.918	0.867	0.613	0.726	0.0112
B) 16×16	0.869	0.602	1.083	0.912	0.592	0.752	**0.0101**
C) 32×32	0.866	0.615	1.065	**0.915**	**0.581**	0.744	0.0105
D) 64×64	0.882	0.579	1.129	0.910	0.582	0.762	0.0121
E) 128×128	0.871	0.602	1.118	0.903	0.583	0.750	0.0137
F) 256×256	0.843	0.618	0.910	0.875	0.634	0.740	0.0141
G) 32×32 – 128×128	0.827	0.656	0.890	0.884	0.628	**0.720**	0.0126
H) + Attention in D_2	**0.913**	0.519	**1.334**	**0.915**	0.537	0.792	0.0134

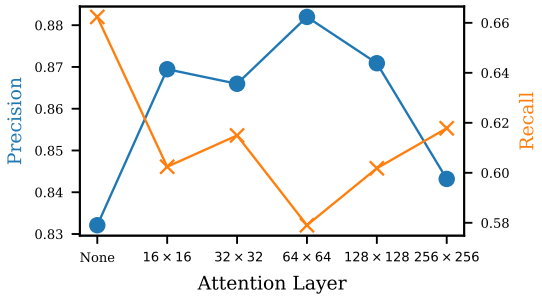


Figure 4.6: Precision and recall metrics for SA-PlaqueGAN when self-attention is inserted at different layers of the generator and discriminator. All models trained on the cored plaque dataset only.

Table 4.7. For all cases of single self-attention layers, these results suggest that SA-PlaqueGAN produces higher fidelity images – evidenced by the increase in Precision and Density – but does so at the expense of diversity, with Recall lower than PlaqueGAN in configs B through F (Figure 4.6). For the single-value metric of KID, improvements upon the baseline are seen when attention is applied earlier in the generator.

Interestingly, applying self-attention to multiple blocks of the generator and main discriminator (config G) results in an overall decrease in performance when compared to the baseline. However, adding self-attention to the secondary discriminator shifts the balance between fidelity and diversity to an extreme, with highest Precision and Density scores with the sacrifice of lowest Recall and 1-NN Accuracy (gen.) scores. We provide additional commentary on the implications of these results in §5.4.

4.6 Oversampling Experiments

We turn our attention towards answering the main question of the thesis:

> *Can GAN-synthesised oversampling of minority classes improve downstream classification of neuropathologies?*

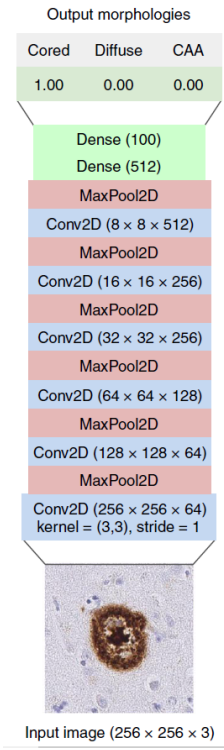


Figure 4.7: Classifier architecture trained for all augmentation experiments. This is also the architecture of the classifier used for image selection. Taken from [74].

To answer this, we focus on the offline setting where minority classes are oversampled as a pre-processing step. Oversampled data are therefore permanent additions to the training data and all training samples are seen during each epoch of training. This is in contrast to real-time augmentation, where transformations are applied at random as the training data is loaded in from memory. Real-time augmentation can vastly increase the pool of training data depending on the method of oversampling – every epoch could result in an entirely different set of training data – and its use is generally advocated to improve generalisation; however, it can be prohibitively slow for methods such as SMOTE and GANs. The alternative of simulating it through pre-processing would require too much hard disk space as this scales linearly with the number of training epochs.

4.6.1 Classifier Architecture

For fair comparison of oversampling methods, we employ a constant classifier architecture throughout. We use the six-layer CNN classifier introduced in [74] to maintain consistency between papers. The classifier architecture is shown in Figure 4.7, taking a 256×256 pixel tile as input and returning a probability for each of the three Aβ morphologies being present. There are six rounds of layers alternating between zero-padded convolutions with 3×3 kernels and spatial max-pooling with a 2×2 window and stride of 2, halving the spatial dimension after each convolutional layer. The learned features are then flattened and passed through two densely-connected layers before output logits are transformed into probabilities with a sigmoid function. All other layers use rectified linear units as the non-linear activation function.

Table 4.8: Oversampling requirements for fine-grained labels of minority classes. The final training dataset size is the sum of the number of original samples and additional required samples.

Morphology	Imbalance Ratio	#Original	#Additional Required
Cored	$r_{\text{cored}} = 22$	1,624	35,728
CAA	$r_{\text{CAA}} = 21$	1,855	38,955
Cored-diffuse	$r_{\text{cored}} = 22$	509	11,198
CAA-diffuse	$r_{\text{CAA}} = 21$	364	7,644
Cored-CAA	$r_{\text{cored}} + r_{\text{CAA}} = 43$	7	301
Cored-diffuse-CAA	$r_{\text{cored}} + r_{\text{CAA}} = 43$	1	43

4.6.2 Training Procedure

We mostly follow the same training procedure in [74], using backpropagation with the ADAM optimizer and PyTorch's MultiLabelSoftMarginLoss (without manual class rescaling weights). Without any weight rescaling, this is the same as standard Binary Cross-entropy (BCE) Loss for each of the classes. Default PyTorch momentum coefficients $\beta_1 = 0.9,\ \beta_2 = 0.99$ are used, alongside a learning rate $\alpha = 8 \times 10^{-4}$ and weight decay $\gamma_{wd} = 8 \times 10^{-3}$. The learning rate is reduced by a multiplicative factor $\gamma_{lr} = 0.4$ every 15 epochs. During training only, spatial dropout is applied to entire feature maps following convolutional layers with probability 0.2, whilst a dropout probability of 0.5 is applied to the two densely connected layers.

Differences in our training are: no real-time data augmentation; early stopping based on validation loss with a patience of 10 epochs; and use of AMP to enable us to match the batch size of 64 used in [74]. We run each experiment three times to account for sensitivity to initialisation and other sources of stochasticity, where each run is tied to a unique random seed.

4.6.3 Oversampling Requirements

In Tang et al. [74], the oversampling requirements are based on the imbalance ratio between majority and minority classes according to their coarse labels. To determine the number of additional minority class samples, the ratios of diffuse to cored plaques, $r_{\text{cored}} = \frac{N_{\text{diffuse}}}{N_{\text{cored}}}$, and diffuse to CAA, $r_{\text{CAA}} = \frac{N_{\text{diffuse}}}{N_{\text{CAA}}}$, are first calculated and rounded down. The number of samples required are then calculated by multiplying the actual number of training samples by this imbalance ratio. For fine-grained labels with more than one minority plaque, the total imbalance ratio is the sum of individual ratios for each minority plaque present in the tile. There is no oversampling of negative samples. The overall requirements are summarised in Table 4.8.

A comparison of the training dataset size by coarse-grained label before and after applying oversampling is shown in Table 4.9. We see that this balancing strategy only roughly balances the dataset. We acknowledge there is still a degree of imbalance with fewer positive cases than negative cases for all Aβ morphologies, as well as the distribution of positive cases between Aβ morphologies (diffuse is still the majority in this regard). Nevertheless, for the purposes of consistency with the existing papers, we follow this strategy.

Table 4.9: Dataset distribution by coarse-grained labels before and after oversampling. Percentages do not necessarily add up to 100 due to the multi-label setting. *Negatives here refers to cases where the neuropathologist labels as negative or when there is insufficient presence of morphology within the tile.

Training Dataset	Total	Cored	Diffuse	CAA	Negative*
Before Oversampling	61,370	2,141 (3.5%)	48,123 (78.4%)	2,227 (3.6%)	9,761 (15.9%)
After Oversampling	155,239	49,411 (31.8%)	67,008 (43.2%)	49,170 (31.7%)	9,761 (6.3%)

Table 4.10: Aβ morphologies which can or cannot be oversampled using the techniques in this thesis. Please see relevant sections describing the implementations for further details.

	Aβ Morphology	Cored	CAA	Cored-diffuse	CAA-diffuse	Cored-CAA	Cored-diffuse-CAA
Method	Simple	✓	✓	✓	✓	✓	✓
	Standard Augmentation	✓	✓	✓	✓	✓	✓
	SMOTE [11]	✓	✓	✓	✓	✓	×
	AugmentAndMix [28]	✓	✓	✓	✓	✓	✓
	PlaqueGAN	✓	✓	✓	✓	×	×

4.6.4 Baseline Oversampling Methods

We establish a number of baselines, both from classical and more recent techniques, to compare GAN-synthesised oversampling with and detail our implementations below. Not every oversampling technique is able to meet the numbers required in Table 4.8; we indicate which fine-labelled morphologies can and cannot be oversampled in Table 4.10. If a particular morphology cannot be oversampled using the technique, we revert to simple oversampling to maintain the dataset statistics.

Simple Oversampling

This is the oversampling method used in [74, 81]. Images from each morphology are simply replicated according to their overall imbalance ratio in Table 4.8. For example, any image with both cored and CAA plaques are replicated 43 times and added to the dataset.

Oversampling with Standard Augmentations

For this technique, minority class images from the original dataset are first replicated according to the Simple Oversampling strategy, before a set of standard computer vision augmentations are applied to the images. This is akin to online image augmentation, only it is done just once as a pre-processing step. We apply augmentations that should be label-preserving: random horizontal and vertical flips; random resized cropping (crop size between 0.8 and 1.0 of the original image size) with random aspect ratio between $\frac{3}{4}$ and $\frac{4}{3}$; random 90 deg rotations; and random colour jitter equivalent to that used in [74].

SMOTE

For image data, SMOTE is more commonly used on vectors of extracted features. Directly applying it to images in the RGB space is often discouraged as the linear combinations of RGB pixel intensities can often lead to unrealistic images.

To use the CNN classifier described in Section 4.6.1, SMOTE must be performed in the image space. This baseline is of interest as there is some research suggesting direct application of SMOTE to synthesise new histopathological images can improve downstream classification [65]. This success is unexpected, but could be explained by the lower complexity and variation seen in histopathological images compared to natural images; meaning linear combinations of images are less likely to be outside of the manifold.

To perform SMOTE-based oversampling, we use the 'Vanilla' implementation of SMOTE in the Imbalanced-learn Python toolbox [44]. As the SMOTE algorithm only accepts vectorised inputs, we first unravel the RGB images ($\mathbb{R}^{256\times256\times3} \rightarrow \mathbb{R}^{196,608}$). With such a high-dimensional space, the SMOTE algorithm can be slow when finding nearest neighbours on larger image datasets. Also, attempting to generate tens of thousands of synthetic images at once is not possible due to memory constraints. Instead, we apply SMOTE using random minibatches of data, growing our pool of synthetic data over multiple iterations. Minibatches of training data are sampled randomly without replacement each iteration, and the minibatch size is determined as a fraction of the dataset size: 0.3 for cored and CAA datasets, 1 (i.e. all data used) for cored-diffuse, CAA-diffuse and cored-CAA datasets. Proceeding with minibatches means that nearest neighbours are not exact for the cored and CAA datasets, but the minibatch size ($\approx$ 500 samples) should be sufficient to find reasonable approximations. SMOTE requires at least two samples from the same class, thus we are unable to apply the technique to the case where all three morphologies are present.

AugmentAndMix Oversampling

AugMix [28] has proven to be effective at tackling situations where the data distributions of training and testing sets do not match; making models less prone to memorising noise in the training distribution and therefore more able to generalise well when various corruptions and perturbations shift the distributions apart. Its original form combines a novel online augmentation method with a consistency loss. However, the results of the ablation study in the paper demonstrates that the augmentation strategy is the main factor driving improvement. This augmentation strategy they coin *AugmentAndMix*. We are therefore interested in assessing the potential of AugmentAndMix to oversample minority data.

Similar to oversampling with standard augmentations, this technique begins with the replication of minority samples using Simple Oversampling. We then perform augmentation on each replicated image via AugmentAndMix using the authors' official implementation[9]. We use all available simple augmentations with a stochastic depth of augmentation chain (ranging between 1 and 3 chained augmentations), an augmentation width of 3, and an augmentation severity of 3 (on a scale of 1 to 10). Standard augmentations are randomly applied to images prior to AugmentAndMix to futher increase the diversity.

[9] `https://github.com/google-research/augmix`

4.6.5 PlaqueGAN-synthesised Oversampling

The small training dataset sizes for cored-CAA and cored-diffuse-CAA morphologies means we are unable to train PlaqueGAN to synthesise these tiles; instead, we use simple oversampling to achieve the numbers required. This is a small fraction of the overall dataset, so we assume that doing so will not confound our experimental results and any conclusions drawn from them.

Image Selection by Class Confidence

We begin by conducting a hyperparameter search on the image selection process described in Section 3.5 to find a suitable confidence threshold against which the selection classifier will reject GAN-synthesised images.

To guide us in selecting suitable values for the search, we first visualise the distribution of class confidence for the real training images and unfiltered GAN-synthesised images when passed through the pre-trained classifier of [74]. These are shown in Figure 4.8 as the yellow and green histograms respectively. Green borders indicate where high confidence is expected whilst red borders indicate where confidence should be low. We observe that in all four cases, PlaqueGAN generates a small proportion of images with low confidence (< 0.2) when they should be high confidence. For the cored only dataset, the classifier also assigns high probability to the diffuse class in a small proportion of images. This confirms the intuition for a strategy where a confidence threshold f_{thresh} is used to reject generated images below the threshold for intended classes, but also to reject generated images with confidence above $1 - f_{\text{thresh}}$ for unintended classes.

We also show the histograms for the case $f_{\text{thresh}} = 0.2$ in purple on Figure 4.8, and see that filtering generally leads to an increased probability of images that classify with high confidence. For example in Figure 4.8a, we see that filtering leads to a higher proportion of generated images with cored plaque confidence > 0.9 compared to real images, but does not improve the proportion of images generated at intermediate confidences, e.g. between 0.8 and 0.9.

As a further check, we visualise samples of images rejected based on $f_{\text{thresh}} = 0.2$ in Figure 4.9. We note that cored rejects often have little Aβ morphology present, or otherwise generate samples that are more akin to diffuse plaques with the characteristic amyloid core missing (sometimes synthesised incorrectly with a glial cell or neuron in place of it). For CAA rejects, we again see a number of cases where there is no distinct $A\beta$ morphology. Where there is morphology present, it tends to deviate from the common halo-like structure surrounding the blood vessels. For both cored-diffuse and CAA-diffuse rejects, there only appears to be one of the two expected morphologies present. This instills some confidence that the image selection process is working as intended.

Based on these observations, we consider rejection confidence thresholds $f_{\text{thresh}} = [0, 0.2, 0.5, 0.7]$ for the hyperparameter search. For each threshold, we train three separate classifiers following the training procedure covered in §4.6.2 and record the average AUPRC on the validation set in Table 4.11, reserving evaluation on the two hold-out test sets for the final experiment.

Improved AUPRC performance is seen for all thresholds of image filtering, with a threshold $f_{\text{thresh}} = 0.5$ having the best score for cored plaques, whilst a threshold $f_{\text{thresh}=0.7}$ has the best

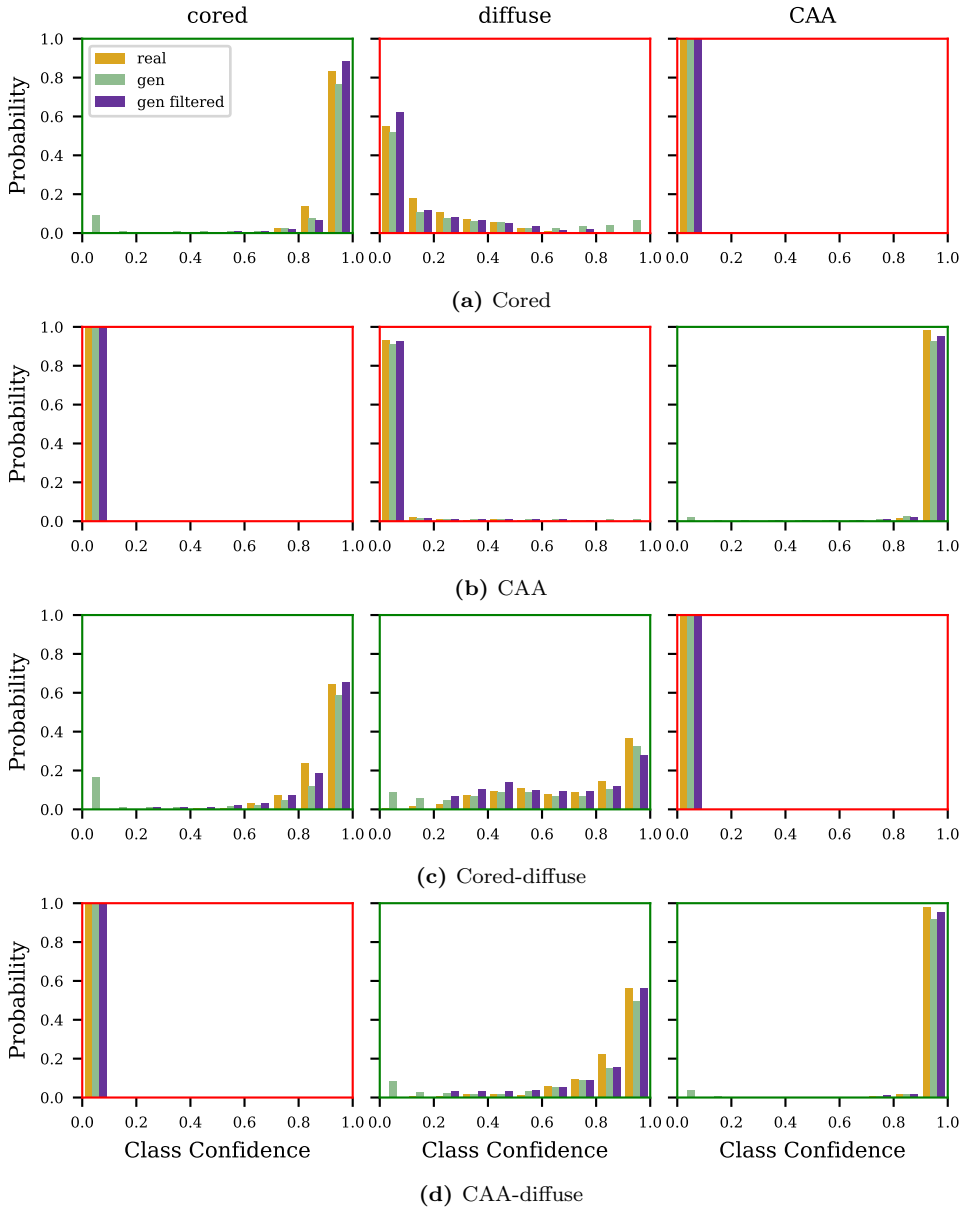
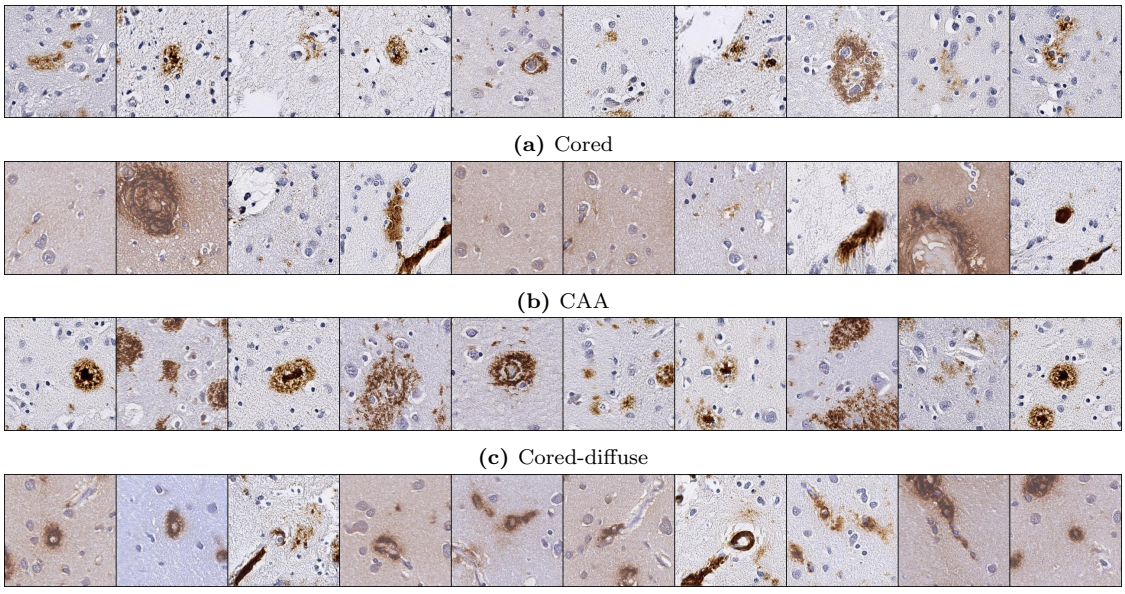


Figure 4.8: Discretised probability distributions of class confidence for real, generated and generated samples with image selection ($f_{\text{thresh}} = 0.2$) for the four minority Aβ morphologies PlaqueGAN is trained on. Green borders indicate where class confidence should be high for a morphology, whilst red borders indicate where class confidence should be low.

Table 4.11: AUPRC results of GAN synthetic oversampling on the validation dataset with different confidence thresholds for image selection. Reported AUPRC are averaged over three runs.

	Validation AUPRC			
Confidence Threshold	Cored	Diffuse	CAA	Ave.
0 (No selection)	0.644	0.990	0.639	0.758
0.2	0.680	0.990	0.678	0.783
0.5	**0.716**	0.990	0.687	**0.798**
0.7	0.700	0.990	**0.692**	0.794



(a) Cored

(b) CAA

(c) Cored-diffuse

(d) CAA-diffuse

Figure 4.9: Uncurated examples of images rejected when the class confidence threshold is set to 0.2. (a) Cored rejects lack the characteristic central, stained amyloid core. The generator instead places glial cells or neurons at the core of the plaque, or otherwise generates plaques devoid of any sort of core which is more like a diffuse plaque. (b) CAA rejects may also lack clear Aβ morphology, or the classic halo or ring-like structure. (c) Cored-diffuse rejects tend to produce only one of either morphology instead of both. (d) CAA-diffuse rejects similarly fail to produce both intended morphologies. Best viewed digitally.

score for CAA. There is not a lot to separate these two thresholds, so we settle on $f_{\text{thresh}} = 0.5$ as strong performance on cored plaques is most important for classifiers whose purpose is to distinguish between AD and non-AD cases. This threshold also has the highest macro-averaged AUPRC score on the validation set.

4.6.6 Final Results

For all baseline oversampling methods, as well as PlaqueGAN oversampled with $f_{\text{thresh}} = [0, 0.5]$, we train the classifier from scratch according to §4.6.2. We also include the case without any oversampling as another baseline.

Average AUPRC results on both hold-out datasets are shown in Table 4.12. These results indicate:

- **Image selection improves downstream performance for PlaqueGAN oversampling.** We report higher AUPRC for cored plaques in both test sets, as well as for CAA in Test Set II when using image selection with PlaqueGAN. This is likely attributed by the filtering out of generated samples whose morphology did not match the intended label.

- **PlaqueGAN significantly outperforms baselines for CAA. Less so on cored plaques.** On Test Set II, the performance gain on CAA is true even without image selection. For cored

Table 4.12: Main oversampling results on two separate test sets. Reported AUPRC are averaged over three runs. For Test Set I, AUPRC for CAA, and therefore macro-averaged AUPRC, are omitted due to insufficient samples. Best macro-averaged and individual results on each test set are highlighted in **bold**.

	Test Set I AUPRC		Test Set II AUPRC			
Oversampling Method	Cored	Diffuse	Cored	Diffuse	CAA	Ave.
None	0.565	0.997	**0.678**	0.987	0.388	0.685
Simple	0.555	0.997	0.579	0.985	0.477	0.680
Standard Augmentation	**0.662**	**0.998**	0.575	0.989	0.261	0.608
SMOTE	0.504	0.997	0.483	0.982	0.402	0.623
AugmentAndMix	0.619	**0.998**	0.659	**0.990**	0.522	0.724
PlaqueGAN (no selection)	0.528	0.997	0.481	0.982	0.594	0.686
PlaqueGAN (with selection)	0.569	0.997	0.559	0.983	**0.632**	**0.725**

plaques, the improvement is less clear as PlaqueGAN only marginally improves upon no oversampling and simple oversampling for Test Set I, whilst it outperforms only SMOTE on Test Set II (although it scores close to Simple and Standard Augmentation oversampling). The reasons for inconsistent performance between cored and CAA morphologies is unclear at this time. If treating all morphologies equally, PlaqueGAN edges AugmentAndMix with the highest macro-averaged AUPRC on Test Set II (0.725 versus 0.724), providing some evidence that PlaqueGAN can ameliorate the class imbalance issue in neuropathology.

- **Oversampling with standard augmentations is only effective on one test set.** On Test Set I, this strategy for oversampling yields the highest AUPRC score for cored plaques by a considerable margin. On Test Set II, cored AUPRC is middle of the pack, but CAA AUPRC is the worst performing. We hypothesise the reasoning for inconsistency between test sets in §5.3.

- **Applying SMOTE direct to Aβ histopathology images is unsuccessful.** Contrary to the findings of [65] who found an improvement in classification performance on breast cancer histopathology datasets, results here suggest that SMOTE is detrimental to the downstream performance. This aligns with the intuition that linear combinations between pixels of two images is likely to produce unrealistic samples outside of the training data manifold.

- **AugmentAndMix offers consistent performance across both hold-out sets.** On both Test Set I and Test Set II, this strategy has the second best macro-averaged AUPRC and does not show severe performance imbalance between cored and CAA pathologies.

Chapter 5

Further Discussion

In this chapter, we present some deeper analysis and thought behind the experimental results of chapter 4.

We start by visualising, through dimensionality reduction, how PlaqueGAN evolves during training to cover the distribution of real images. We also defend our choice to keep improved gradient flow in PlaqueGAN's architecture, when metrics suggested it was detrimental to performance. An important discussion is had on the role of single expert labellers as a potential confounder to findings before we end the chapter reasoning why self attention failed to improve PlaqueGAN.

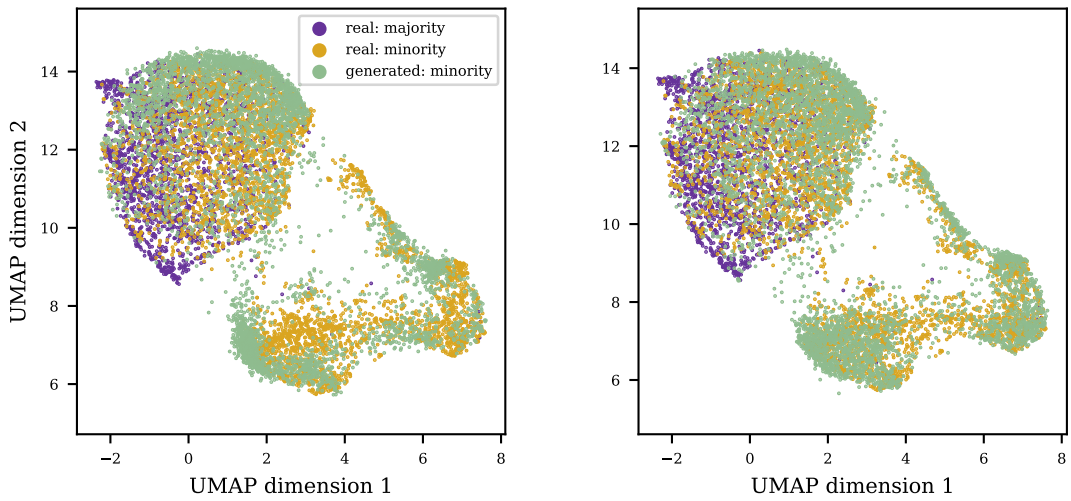
5.1 PlaqueGAN Captures a Diverse Set of Modes with High Fidelity

Both quantitative and qualitative assessments in §4.5 demonstrate PlaqueGAN's effectiveness in learning the distribution of real neuropathology slides. 1-NN Accuracy scores are above 0.5, while inspection of LPIPS nearest neighbours and the smoothness of the generator latent space build evidence that PlaqueGAN has not simply memorised the training data; instead, it is able to imagine novel and diverse samples of various combinations of minority Aβ morphologies.

To understand PlaqueGAN further, we can apply dimensionality reduction techniques to visually compare the distributions of real and generated images. Specifically, we use Uniform Manifold Approximation and Projection (UMAP) [52, 53] from the official implementation[1]. Default parameters of a 2-dimensional embedding, ℓ_2 distance, a minimum distance of 0.1 and 15 nearest neighbours are used for the algorithm. We first fit UMAP on the training data, using all minority classes and a random subset of $2,000$ samples of majority diffuse plaques. We then project any generated images of minority classes using the learned embeddings.

In Figure 5.1, we compare how the distribution of generated images evolves during the training of PlaqueGAN. Early on in training (Figure 5.1a), high density clustering of generated samples can be seen in a small number of regions and indicate that generator diversity is limited. Furthermore, there are a number of generated samples (around the middle) that appear to be outside of the

[1] `https://github.com/lmcinnes/umap`



(a) Early training **(b)** Final

Figure 5.1: UMAP embeddings of a subset of majority real training plaques, all minority real training plaques, and an equal number of minority generated plaques (without selection). (a) Early in training (5k iterations), generated images clump in high density to a small number of locations. (b) By the end of training, the generator covers a wider area of the real image distribution, whilst producing fewer samples outside of the manifold.

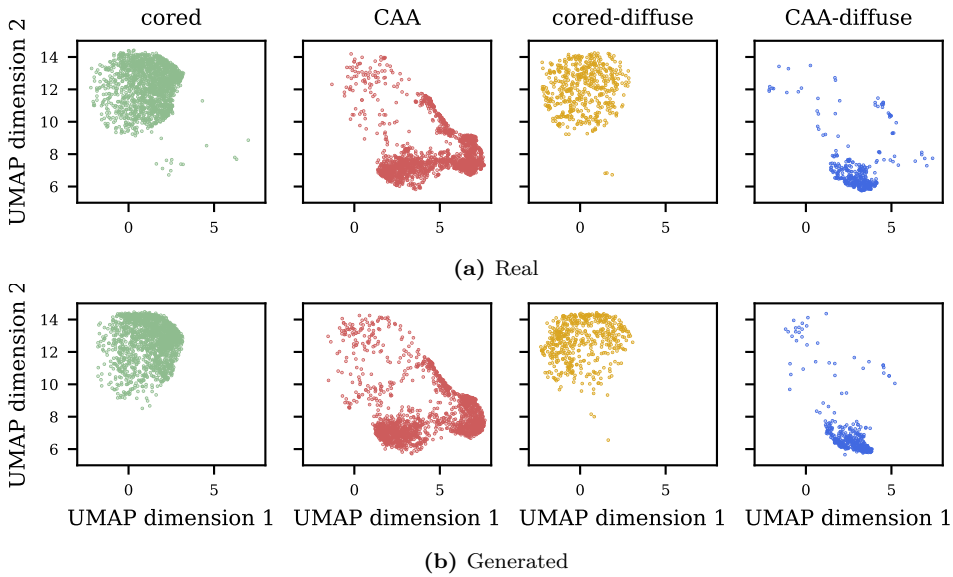
real manifold; an indication of low fidelity. By the end of training (Figure 5.1b), the PlaqueGAN generator matches the distribution more closely, with fewer samples outside of the real manifold. Sample diversity and fidelity are improved over the course of training.

The distribution of the majority diffuse plaque morphology is seen to have significant overlap with minority morphologies. This is unsurprising, considering that cored-diffuse samples will share similar features due to majority class presence, whilst cored plaques are also morphologically similar to diffuse plaques. It may also be the case that 2D embeddings are too constrained to adequately separate these classes.

We further plot the distribution of real and generated samples by minority plaque type in Figure 5.2. We see that for CAA, PlaqueGAN may be less successful in modelling the low density region – potentially generating a number of images outside of the real distribution. With cored-diffuse images, we see a slight difference in the shape of the distribution along the bottom. It appears that PlaqueGAN has not been able to fully cover the distribution of the real images, and might explain why the 1-NN Accuracy (real) result in Table 4.5 for cored-diffuse plaques is highest.

5.2 Does Improving Gradient Flow in PlaqueGAN Actually Help?

When incrementally applying changes to the baseline FastGAN in §4.5.2, we found that adding improvements to the gradient flow of the discriminator and generator (config D) to config C worsened performance across all metrics. This was rectified by subsequently adding a second



(a) Real

(b) Generated

Figure 5.2: UMAP representations of (a) Real and (b) Generated images from PlaqueGAN, split by minority plaque label.

Table 5.1: Further ablation of PlaqueGAN to understand the role of improved gradient flow and the dual discriminator.

Configuration	Precision (↑)	Recall (↑)	Density (↑)	Coverage (↑)	1NN Acc. (real)	1NN Acc. (gen.)	KID (↓)
PlaqueGAN config C	**0.928**	0.351	**1.610**	0.885	**0.576**	0.894	0.0157
D) + Improved Gradient Flow	0.902	0.301	1.266	0.843	0.622	0.924	0.0262
D′) Dual Discriminator	0.918	0.357	1.452	**0.887**	0.596	0.890	**0.0144**
config E (D + D′)	0.895	**0.403**	1.227	0.864	0.611	**0.854**	0.0219

discriminator D_2, but it is unclear whether overall performance is hampered by the gradient flow changes. We therefore ablate the model further, beginning from config C of PlaqueGAN, to view effects of the improved gradient flow and second discriminator in isolation.

Quantitative metrics from these ablations are summarised in Table 5.1. We see that adding the second discriminator leads to minor improvements across diversity metrics compared to config C. Using the second discriminator in tandem with improved gradient flow (config D′), we see a greater improvement in Recall and 1-NN Accuracy (gen.), signifying a possible interaction between these two techniques. We hypothesise that too strong a gradient flow early on in training could amplify noise in the feedback from the discriminator, hindering generator improvement. Combining a second discriminator might act to smooth out the noise through averaging, allowing the generator to better see the 'true' amplified signal. Also, as diversity is comparatively more difficult to improve than fidelity, improved gradient flow in the final PlaqueGAN model is warranted.

5.3 Labels from Single Experts May Prove Problematic

Visual Turing test results in Section 4.5.4 are interesting as higher real error rates indicate the expert neuropathologist tended to label images as fake. Using the first 50 images as a bedding-in process may be insufficient for the evaluator to be accustomed to viewing restricted tiles of single (or at most a few) plaques without surrounding context. The interface of CVAT may hinder this further as automatic enlarging of the images to fill the screen could negatively impact the perception of quality.

A second, plausible explanation for high real error rates may be down to idiosyncrasies among expert neuropathologists. In the consensus study of Wong et al. [81], it was found that cored plaques exhibited the least agreement among neuropathologists. Only 11% of all cored-labelled images had complete agreement from all 5 neuropathologists in comparison to 33% for CAA-labelled images and 65% for diffuse-labelled images. In fact, almost half of all cored-labelled images had only one annotator label it as positive. This degree of disagreement could realistically create the scenario where real images annotated by one expert are dismissed as fake by another.

Inconsistency between oversampling results on the two hold-out datasets in §4.6 may also be due to one set being labelled by a single expert whilst the other is a consensus-of-two neuropathologists. We hypothesise that results on Test Set I may not be a good proxy for generalisation performance as it is labelled by the same single expert as the training and validation datasets. As a result, models trained with the training dataset may be able to learn the labelling biases specific to that neuropathologist. Hyperparameters set on the validation set such f_{thresh} may also be misguided. This leaking of information could lead to inflated scores on the test set for models that inherit these biases. Perhaps oversampling with standard augmentation inadvertently reinforces these biases, resulting in far better performance on the single expert-labelled dataset than the consensus-labelled dataset. If this were true, it might signal that models trained with AugmentAndMix and PlaqueGAN oversampling strategies are more capable of learning general features of the underlying data distribution, even in the presence of the label bias.

These speculations invite future research to study methods that mitigate annotator bias and verify their applicability on Aβ histopathology datasets. Through training PlaqueGAN on a dataset with less annotator bias, this may strengthen the viability of PlaqueGAN as an oversampling tool to resolve the class imbalance issue.

5.4 Attention May Not Always Be What You Need

In the original paper for Self-Attention in GANs (SAGAN), and others that followed such as BigGAN, the benefit of adding nonlocal attention layers was generally confirmed quantitatively through tracking the single-value FID metric as well as visual examination of generated image fidelity. The impact on diversity, however, is less explored.

For our plaque datasets, the addition of self-attention to the middle layers of the PlaqueGAN generator and discriminator similarly indicate improvement in the single-value metric of KID. However, we find that this is accompanied by a decrease in diversity, as measured by Recall and 1-NN Accuracy (gen.) metrics. This trade-off could mean that self-attention is not beneficial to our intended task of oversampling minority plaques, where generating sufficiently different and

diverse plaque examples is of importance. As the utility of self-attention is linked to the need to model long-range dependencies across the features, we consider that this might not be as crucial to the Aβ plaque datasets. Features such as the central amyloid core closely surrounded by degenerating neurites in cored plaques are highly local, and could mean the narrow receptive field of the convolutional layers is sufficient.

We note that models such as SAGAN and BigGAN are class-conditional, trained to generate images from many classes. We postulate that self-attention might prove its worth more so in the presence of multiple classes; a class-conditional version of PlaqueGAN would be a research direction that could help answer this. We also recognise that self-attention may have aided generation for datasets whose tiles include more than one morphology and this is one of the drawbacks of running experiments primarily on the cored plaque dataset.

Chapter 6

Conclusions and Future Work

In this thesis, we concentrated on ameliorating class imbalance present in neuropathology datasets. These datasets have recently proven a promising addition to the neuropathologists' toolbox, having been used to train deep learning models that can assist in scoring WSIs for Alzheimer's Disease and other neuropathologic changes [74, 81]. To achieve this, we introduce PlaqueGAN: an unconditional generative adversarial network based on the FastGAN [49] architecture, that can synthesise new samples of minority classes with limited time and computational resources; limited training data; and without training stability issues that commonly plague GANs. At the time of writing, we believe this to be the first attempt at generative modelling of Aβ morphologies in immunohistochemically stained slides of human brain; previous attempts being mainly focussed on synthesising cancer tissue in other organs such as the liver and skin.

Through thorough quantitative and qualitative evaluation, we demonstrated PlaqueGAN's ability to generate a diverse set of high fidelity images across plaque morphologies without memorisation of training data. A Visual Turing Test with an expert neuropathologist shows deception rates $> 40\%$, confirming the difficulty of distinguishing real images from PlaqueGAN-synthesised images. Dimensionality reduction through UMAP provides further qualitative evidence that the generated distribution is similar to that of the training data, whilst ablation studies on PlaqueGAN confirm that the proposed changes to the FastGAN architecture and training are meaningful for improving image fidelity and diversity – both are intuitively important such that generated samples provide new information for downstream models to learn from.

For oversampling training data with PlaqueGAN, we proposed a simple technique for image selection based on class confidence scores, extracted by passing generated images through a pretrained *selection classifier*. We showed empirically that filtering prior to oversampling leads to better downstream classification performance on plaques. PlaqueGAN with image selection produced the highest macro-averaged AUPRC on a held-out test set labelled by a consensus-of-two expert neuropathologists; this in comparison to classical oversampling methods such as replication with standard augmentations and SMOTE (albeit applied unconventionally in the image space), as well as the more recent AugmentAndMix method. This provides initial evidence of PlaqueGAN's viability for alleviating class imbalance issues in image datasets containing Aβ morphologies processed from WSIs. This could greatly reduce the burden on neuropathologists, who would otherwise require hours spent tediously labelling tiles to obtain sufficiently large samples

of scarcer cored plaques and CAA. Instead, PlaqueGAN offers the potential to sample unique instances of minority morphologies infinitely, and to build datasets as large as memory allows. Alternatively, PlaqueGAN can be integrated as an online method if practitioners are willing to trade the storage memory requirements for training speed.

There are numerous opportunities to expand upon – and address some of the limitations of – this work. Firstly, we note that due to PlaqueGAN being unconditional, separate models must be trained and sampled from for each unique combination of plaque morphologies. This is inefficient not just because multiple versions of the model have to be trained and stored, but also because it throws away potentially useful information about correlations between the morphologies. It also prevents PlaqueGAN learning from single-shot or very few-shot training datasets. A class-conditional extension of PlaqueGAN could better learn these correlations and incorporate majority diffuse-only as well as negative training examples; this should reduce the likelihood of manifold intrusion or producing samples without any significant Aβ pathology, but might also allow it to synthesise combinations of morphologies not seen often during training. SA-PlaqueGAN might also prove its worth in a conditional setting where multiple Aβ combinations must be learned concurrently. The challenge here is to ensure balanced training of the GAN for each morphology.

In terms of diversity, a clear gap still exists between the learned distributions of PlaqueGAN's generator and the real distributions of images. This can be seen when comparing Recall and 1-NN Accuracy baselines in Table 4.3 to PlaqueGAN in Table 4.5. Adapting the concept of a style mixing generator from the StyleGAN architectures [35, 36] might aid in closing this gap. To extract the most out of style mixing, a mapping network $\mathcal{M}$ [35] should also be incorporated to transform the original latent space to an intermediate latent space $\mathcal{M} : \boldsymbol{\mathcal{Z}} \longrightarrow \boldsymbol{\mathcal{W}}$. The idea behind the mapping network is to provide a further disentangled latent space, where individual elements of a latent vector $\mathbf{w} \sim \boldsymbol{\mathcal{W}}$ control different features of the generated images. This enhanced control could allow practitioners to target generation of Aβ morphologies with specific characteristics of interest. The work of Quiros et al. [63] is an example of successful integration of a mapping network with style mixing to histopathological images of cancer tissue.

Finally, we support further research into image selection techniques for PlaqueGAN. Whilst our simple method helped filter out worst generated images, it only passively applies a single threshold for all combinations of plaque morphologies. We see from Figure 4.8 that probability distributions of class confidence are irregular. For example, the probability distribution of reals for presence of diffuse plaques differs between Cored-diffuse and CAA-diffuse images – the pre-trained classifier is overall more confident in diffuse plaque presence for CAA-diffuse cases. Adapting thresholds to the distribution of reals might therefore be more sensible, but could prove difficult to tune. Application of active methods such as reinforcement learning, where the decision-making criteria to include or exclude generated samples is instead learned, could lead to better oversampling that translates to improved downstream performance when training a classifier on PlaqueGAN-generated data.

Bibliography

[1] Ibrahim Saad Ali, Mamdouh Farouk Mohamed, and Yousef Bassyouni Mahdy. *Data Augmentation for Skin Lesion using Self-Attention based Progressive Generative Adversarial Network*. 2019. arXiv: 1910.11960 [eess.IV].

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[3] Samuel A. Barnett. *Convergence Problems with Generative Adversarial Networks (GANs)*. 2018. arXiv: 1806.11382 [cs.LG].

[4] Shane Barratt and Rishi Sharma. *A Note on the Inception Score*. 2018. arXiv: 1801.01973 [stat.ML].

[5] Emanuel Ben-Baruch et al. *Asymmetric Loss For Multi-Label Classification*. 2021. arXiv: 2009.14119 [cs.CV].

[6] Binod Bhattarai et al. "Sampling Strategies for GAN Synthetic Data". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2020). DOI: 10.1109/icassp40776.2020.9054677. URL: http://dx.doi.org/10.1109/ICASSP40776.2020.9054677.

[7] Mikołaj Bińkowski et al. *Demystifying MMD GANs*. 2021. arXiv: 1801.01401 [stat.ML].

[8] Ali Borji. "Pros and Cons of GAN Evaluation Measures". In: *CoRR* abs/1802.03446 (2018). arXiv: 1802.03446. URL: http://arxiv.org/abs/1802.03446.

[9] Ali Borji. "Pros and Cons of GAN Evaluation Measures: New Developments". In: *CoRR* abs/2103.09396 (2021). arXiv: 2103.09396. URL: https://arxiv.org/abs/2103.09396.

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. 2019. arXiv: 1809.11096 [cs.LG].

[11] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[12] Ting Chen et al. *Self-Supervised GANs via Auxiliary Rotation Loss*. 2019. arXiv: 1811.11212 [cs.LG].

[13] Min Jin Chong and David Forsyth. *Effectively Unbiased FID and Inception Score and where to find them*. 2020. arXiv: 1911.07023 [cs.CV].

[14] Antonia Creswell et al. "Generative Adversarial Networks: An Overview". In: *IEEE Signal Processing Magazine* 35.1 (Jan. 2018), pp. 53–65. ISSN: 1053-5888. DOI: 10.1109/msp.2017.2765202. URL: http://dx.doi.org/10.1109/MSP.2017.2765202.

[15] Yin Cui et al. *Class-Balanced Loss Based on Effective Number of Samples*. 2019. arXiv: 1901.05555 [cs.CV].

[16] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[17] Neofytos Dimitriou, Ognjen Arandjelović, and Peter D Caie. *Deep Learning for Whole Slide Image Analysis: An Overview*. 2019. arXiv: 1910.11097 [cs.CV].

[18] Georgios Douzas and Fernando Bacao. "Effective data generation for imbalanced learning using conditional generative adversarial networks". In: *Expert Systems with Applications* 91 (2018), pp. 464–471. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.09.030. URL: https://www.sciencedirect.com/science/article/pii/S0957417417306346.

[19] D.C Dowson and B.V Landau. "The Fréchet distance between multivariate normal distributions". In: *Journal of Multivariate Analysis* 12.3 (1982), pp. 450–455. ISSN: 0047-259X. DOI: https://doi.org/10.1016/0047-259X(82)90077-X. URL: https://www.sciencedirect.com/science/article/pii/0047259X8290077X.

[20] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. *Generative Multi-Adversarial Networks*. 2017. arXiv: 1611.01673 [cs.LG].

[21] Maurice Fréchet. "Sur la distance de deux lois de probabilité". In: *Comptes Rendus Hebdomadaires des Seances de L Academie des Sciences* 244.6 (1957), pp. 689–692.

[22] Andrés Felipe Giraldo-Forero et al. "Managing Imbalanced Data Sets in Multi-label Problems: A Case Study with the SMOTE Algorithm". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by José Ruiz-Shulcloper and Gabriella Sanniti di Baja. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 334–342.

[23] I. Goodfellow et al. "Generative Adversarial Nets". In: *NIPS*. 2014.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[25] Arthur Gretton et al. "A Kernel Two-Sample Test". In: *J. Mach. Learn. Res.* 13.null (Mar. 2012), pp. 723–773. ISSN: 1532-4435.

[26] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning". In: *Advances in Intelligent Computing*. Ed. by De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887. ISBN: 978-3-540-31902-3.

[27] Haibo He and Edwardo A. Garcia. "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. DOI: 10.1109/TKDE.2008.239.

[28] Dan Hendrycks et al. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*. 2020. arXiv: 1912.02781 [stat.ML].

[29] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500 (2017). arXiv: 1706.08500. URL: http://arxiv.org/abs/1706.08500.

[30] Gaofeng Huang and Amir Hossein Jafari. "Enhanced balancing GAN: minority-class image generation". In: *Neural Computing and Applications* (June 2021). ISSN: 1433-3058. DOI: 10.1007/s00521-021-06163-8. URL: http://dx.doi.org/10.1007/s00521-021-06163-8.

[31] Rui Huang et al. "FX-GAN: Self-Supervised GAN Learning via Feature Exchange". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 3183–3191. DOI: 10.1109/WACV45572.2020.9093525.

[32] Bradley T Hyman et al. "National Institute on Aging–Alzheimer's Association guidelines for the neuropathologic assessment of Alzheimer's disease". In: *Alzheimer's & dementia* 8.1 (2012), pp. 1–13.

[33] Justin M Johnson and Taghi M Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (2019), pp. 1–54.

[34] Animesh Karnewar and Oliver Wang. *MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks*. 2020. arXiv: 1903.06048 [cs.CV].

[35] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2019. arXiv: 1812.04948 [cs.NE].

[36] Tero Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*. 2020. arXiv: 1912.04958 [cs.CV].

[37] Tero Karras et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2018. arXiv: 1710.10196 [cs.NE].

[38] Tero Karras et al. *Training Generative Adversarial Networks with Limited Data*. 2020. arXiv: 2006.06676 [cs.CV].

[39] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[40] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[41] Naveen Kodali et al. *On Convergence and Stability of GANs*. 2017. arXiv: 1705.07215 [cs.AI].

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[43] Tuomas Kynkäänniemi et al. *Improved Precision and Recall Metric for Assessing Generative Models*. 2019. arXiv: 1904.06991 [stat.ML].

[44] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.

[45] Timothée Lesort, Jean-François Goudou, and David Filliat. "Training Discriminative Models to Evaluate Generative Ones". In: *CoRR* abs/1806.10840 (2018). arXiv: 1806.10840. URL: http://arxiv.org/abs/1806.10840.

[46] Adrian B Levine et al. "Synthesis of diagnostic quality cancer pathology images by generative adversarial networks". In: *The Journal of pathology* 252.2 (2020), pp. 178–188.

[47] Jae Hyun Lim and Jong Chul Ye. *Geometric GAN*. 2017. arXiv: 1705.02894 [stat.ML].

[48] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. 2018. arXiv: 1708.02002 [cs.CV].

[49] Bingchen Liu et al. *Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis*. 2021. arXiv: 2101.04775 [cs.CV].

[50] Mario Lucic et al. *Are GANs Created Equal? A Large-Scale Study*. 2018. arXiv: 1711.10337 [stat.ML].

[51] Giovanni Mariani et al. *BAGAN: Data Augmentation with Balancing GAN*. 2018. arXiv: 1803.09655 [cs.CV].

[52] L. McInnes, J. Healy, and J. Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *ArXiv e-prints* (Feb. 2018). arXiv: 1802.03426 [stat.ML].

[53] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29 (2018), p. 861.

[54] Paulius Micikevicius et al. *Mixed Precision Training*. 2018. arXiv: 1710.03740 [cs.AI].

[55] Suzanne S Mirra et al. "The Consortium to Establish a Registry for Alzheimer's Disease (CERAD): Part II. Standardization of the neuropathologic assessment of Alzheimer's disease". In: *Neurology* 41.4 (1991), pp. 479–479.

[56] Aamir Mustafa et al. *Training a Better Loss Function for Image Restoration*. 2021. arXiv: 2103.14616 [eess.IV].

[57] Muhammad Ferjad Naeem et al. *Reliable Fidelity and Diversity Metrics for Generative Models*. 2020. arXiv: 2002.09797 [cs.CV].

[58] Augustus Odena, Christopher Olah, and Jonathon Shlens. *Conditional Image Synthesis With Auxiliary Classifier GANs*. 2017. arXiv: 1610.09585 [stat.ML].

[59] Richard Osuala et al. *A Review of Generative Adversarial Networks in Cancer Imaging: New Applications, New Solutions*. 2021. arXiv: 2107.09543 [eess.IV].

[60] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. "On Buggy Resizing Libraries and Surprising Subtleties in FID Calculation". In: *ArXiv* abs/2104.11222 (2021).

[61] E. Parzen. "Mathematical Considerations in the Estimation of Spectra". In: *Technometrics* 3 (1961), pp. 167–190.

[62] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[63] Adalberto Claudio Quiros, Roderick Murray-Smith, and Ke Yuan. *PathologyGAN: Learning deep representations of cancer tissue*. 2021. arXiv: 1907.02644 [eess.IV].

[64] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[65] Md Shamim Reza and Jinwen Ma. "Imbalanced Histopathological Breast Cancer Image Classification with Convolutional Neural Network". In: *2018 14th IEEE International Conference on Signal Processing (ICSP)*. 2018, pp. 619–624. DOI: 10.1109/ICSP.2018.8652304.

[66] Mehdi S. M. Sajjadi et al. *Assessing Generative Models via Precision and Recall*. 2018. arXiv: 1806.00035 [stat.ML].

[67] Tim Salimans et al. "Improved Techniques for Training GANs". In: *CoRR* abs/1606.03498 (2016). arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498.

[68] Vignesh Sampath et al. "A survey on generative adversarial networks for imbalance problems in computer vision tasks". In: *Journal of big Data* 8.1 (2021), pp. 1–59.

[69] Maximilian Seitzer. *pytorch-fid: FID Score for PyTorch*. https://github.com/mseitzer/pytorch-fid. Version 0.1.1. Aug. 2020.

[70] Boris Sekachev et al. *opencv/cvat: v1.1.0*. Version v1.1.0. Aug. 2020. DOI: 10.5281/zenodo.4009388. URL: https://doi.org/10.5281/zenodo.4009388.

[71] Zhuoran Shen et al. *Efficient Attention: Attention with Linear Complexities*. 2020. arXiv: 1812.01243 [cs.CV].

[72] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[73] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[74] Ziqi Tang et al. "Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline". In: *bioRxiv* (2019). DOI: 10.1101/454793. eprint: https://www.biorxiv.org/content/early/2019/03/11/454793.full.pdf. URL: https://www.biorxiv.org/content/early/2019/03/11/454793.

[75] Lucas Theis, Aäron van den Oord, and Matthias Bethge. *A note on the evaluation of generative models*. 2016. arXiv: 1511.01844 [stat.ML].

[76] M. Tschuchnig, G. Oostingh, and M. Gadermayr. "Generative Adversarial Networks in Digital Pathology: A Survey on Trends and Future Potential". In: *Patterns* 1 (2020).

[77] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[78] Juan C Vizcarra et al. "Validation of machine learning models to detect amyloid pathologies across institutions". In: *Acta neuropathologica communications* 8.1 (2020), pp. 1–13.

[79] Xiaolong Wang et al. *Non-local Neural Networks*. 2018. arXiv: 1711.07971 [cs.CV].

[80] Tom White. *Sampling Generative Networks*. 2016. arXiv: 1609.04468 [cs.NE].

[81] Daniel R. Wong et al. "Deep learning from multiple experts improves identification of amyloid neuropathologies". In: *bioRxiv* (2021).

[82] Qiantong Xu et al. “An empirical study on evaluation metrics of generative adversarial networks”. In: *CoRR* abs/1806.07755 (2018). arXiv: 1806.07755. URL: http://arxiv.org/abs/1806.07755.

[83] Yuan Xue et al. “Selective synthetic augmentation with HistoGAN for improved histopathology image classification”. In: *Medical Image Analysis* 67 (2021), p. 101816. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2020.101816. URL: https://www.sciencedirect.com/science/article/pii/S1361841520301808.

[84] Yuan Xue et al. *Synthetic Augmentation and Feature-based Filtering for Improved Cervical Histopathology Image Classification*. 2019. arXiv: 1907.10655 [eess.IV].

[85] Yasin Yazıcı et al. *The Unusual Effectiveness of Averaging in GAN Training*. 2019. arXiv: 1806.04498 [stat.ML].

[86] Jiarong Ye et al. *Synthetic Sample Selection via Reinforcement Learning*. 2020. arXiv: 2008.11331 [cs.CV].

[87] Xin Yi, Ekta Walia, and Paul Babyn. “Generative adversarial network in medical imaging: A review”. In: *Medical Image Analysis* 58 (2019), p. 101552. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2019.101552. URL: https://www.sciencedirect.com/science/article/pii/S1361841518308430.

[88] Han Zhang et al. *Self-Attention Generative Adversarial Networks*. 2019. arXiv: 1805.08318 [stat.ML].

[89] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV].

[90] Hang Zhao et al. “Loss Functions for Image Restoration With Neural Networks”. In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.

[91] Shengyu Zhao et al. *Differentiable Augmentation for Data-Efficient GAN Training*. 2020. arXiv: 2006.10738 [cs.CV].

[92] Sharon Zhou et al. *HYPE: A Benchmark for Human eYe Perceptual Evaluation of Generative Models*. 2019. arXiv: 1904.01121 [cs.CV].