Imperial College London

Imperial College London

Department of Computing

RL-based Adaptive Weighting for Composite Loss Functions in GANs for Compressed Sensing MRI Reconstruction

Author:
Joshua Cheng

Supervisor(s):
Pancham Shukla
Chintan Modi

Submitted in partial fulfillment of the requirements for the MSc degree in Computing of Imperial College London

19th February 2024

Abstract

Magnetic Resonance Imaging (MRI) is a widely-used imaging modality in the study of tissue structure and function. However the long scanning duration required for image acquisition limits its usage in more applications, as patients must remain still throughout to avoid movement artefacts from appearing. Amongst techniques which undersample the data to reduce imaging duration, Compressed Sensing (CS) holds the most potential, however it also suffers from a long reconstruction processing time. Attempts to apply deep learning to solve this issue have found great success; in particular, the use of Generative Adversarial Networks (GANs) has allowed for significant performance increases with reduced reconstruction time. Typically, training of GANs incorporates multiple loss functions which tends to lead to increased performance, although it is not well understood which loss components provide the most benefit.

To provide greater feedback on the use of different loss components, this study proposes an Adaptive Weighting GAN (AW-GAN) which utilises deep reinforcement learning to adjust the weighting applied to various loss components throughout training. A novel state and action space was designed for application of reinforcement learning to generator training. Evaluation against the performance of DAGAN shows a 19% reduction in NMSE, 5% increase in PSNR and 0.0187 increase in SSIM. The benefits and limitations of design decisions have been evaluated with areas for future work identified.

Acknowledgements

A special thank you to both Dr Pancham Shukla and Dr Chintan Modi for their invaluable support and advice throughout the entirety of this project, without which this would not have been possible.

Contents

List of Figures

1 Introduction

The motivations behind this study are explored, before the aims and scope of this project are outlined in detail. A brief explanation of the remaining structure of the report is provided at the end.

1.1 Motivations

Magnetic Resonance Imaging (MRI) has been established as one of the most effective forms of medical imaging available. The non-invasive technique has the flexibility to be applied to imaging various tissue types through the use of contrast agents, which affect how MR signals interact with different mediums. This provides appropriate contrast in standard imaging whilst also allowing for examination of tissue and organ functionality in the body [1]. Despite this versatility, more frequent usage of MRI has been limited by the economic and physical costs on the hospital and patient. An hour of usage could cost £350 - £500 [1] whilst also placing severe demands on the patient; they must remain still during long scanning periods to reduce movement artefacts, with some procedures requiring their breath to be held even if they have existing breathing conditions. Scanning duration remains the limiting factor in greater adoption of the imaging procedure in more applications.

Many avenues have been explored to reduce the procedure length. MRI records data into k-space (frequency domain), which is known to be sparse. Many image compression techniques rely on the sparsity of an image with sparsifying transforms to compress the data. Compressive Sensing (CS) suggests that if data is sparse, then it should be possible to undersample the data and achieve satisfactory reconstruction of the full dataset afterwards [2]. Using incoherent sampling of a sparse image and an iterative non-linear optimisation process, a signal can be undersampled below the Nyquist-Shannon criterion and be reconstructed without aliasing artefacts. Successful results have been recorded [3] however all traditional approaches have faced the same limitation; the reconstruction step requires significant processing time and remains impractical for clinical usage.

With the rise of machine learning, various attempts have been explored to utilise deep

learning to speed up the reconstruction phase. By offloading processing time to the training procedure and using prior knowledge of image features, faster and more accurate reconstruction can be achieved. Networks can learn the mapping between a reduced dataset and its expected original image, acting as the non-linear reconstruction step of the traditional CS method. Neural networks have the added benefit of memorising useful image features in the data during training, much like a person could learn the features which make up a particular object, allowing for increased performance. Novel research has identified Generative Adversarial Networks (GANs) as having a natural aptitude for solving this problem, given its nature of generating images from some latent code [4]. Whilst there have been attempts at using GANs for this purpose [5], the field of GANs remains relatively new.

The loss functions chosen for GANs remains an area of rapid development as researchers seek the optimal combinations of losses used for training [6]. The authors of DAGAN [7] noted that a combination of image, frequency (spectral) and perceptual similarity losses provided the greatest performance when training the generator, with weights provided only to scale their values to a similar magnitude. Weightings for losses tend to be empirically chosen, with little understanding of why certain values are optimal; ideally, the weightings would become part of the automatic training of the model. Minimal research has been performed on reinforcement learning (RL) based hyperparameter optimisation, with none exploring reinforcement learning with adaptive weighting for loss functions. In theory, adaptive weightings should allow for the model to be "pushed" out of local optimums through differing training pressures.

1.2 Project Aims

This project built upon the work of DAGAN and consisted of three main objectives, which have been listed below along with a more detailed explanation of the reasoning behind them. DAGAN's performance acted as a benchmark and allowed for development to be focused on the adaptive weighting rather than development of the GAN's architecture itself.

1. **Determine if there is an exploitable relationship between image, spectral and perceptual losses during the training process**.

   The authors of DAGAN found the greatest performance when all three of the afore-

mentioned losses were utilised for training; their assumption was that each loss component contains information that the others do not, thus a combination is required for the best results. They noted that the model was robust to various weights, with values mainly chosen to bring the losses to similar magnitudes for training [7]. There was no indication of the importance of each loss nor whether there was any correlation between them, both of which would be informative on whether reinforcement learning was suitable for the application.

2. **Design and implement a deep reinforcement learning model to provide adaptive weighting throughout the generator training.**

   The reinforcement model was built on top of the codebase for DAGAN to allow for a fair comparison of results. Most RL applications have been developed for game environments and so a novel environment would need to be designed for the model to interact with. Since the only applications of RL-based hyperparameter optimisation have used Q-learning [8, 9], the effectiveness of policy gradients in this area will be explored.

3. **Evaluate the performance effects of adaptive weighting and the proposed design**

   Direct comparison to DAGAN results would easily be possible given that the same base code was used for development. The same comparison metrics used by DAGAN will be considered when evaluating the performance impact of adaptive weighting. Given the novelty of the design, the strengths and limitations of this approach will be analysed and areas for future improvement identified.

1.3 Report Structure

A brief background on this project is provided, starting with technical details covering all the understanding needed for this study. Relevant works will be covered across all key areas in a broad literature review. Some preliminary work will be covered, exploring some of the work of DAGAN before an in-depth explanation of the design process behind AW-GAN. The results of various experimental configurations has been recorded before analysis of these results; insight has been given on the model's actions and reasoning. The study concludes with a summary of the work completed along with areas for future work.

2 Background

Given the variety of topics covered in this study, a technical background of the important areas has been provided to aid understanding in later sections. A literature review was conducted on all relevant fields to summarise the state of research and where this project sits within it.

2.1 Technical Background

Technical details of MRI, CS, GANs and RL have been provided where the information is relevant. The key mathematical points are highlighted to provide the understanding required for this study.

2.1.1 Magnetic Resonance Imaging

MRI uses the interaction between various magnetic fields and radiofrequency (RF) pulses to record measurements [10]. A permanently active superconducting magnet generates a magnet field whilst three additional electromagnets are controlled to generate magnetic fields in orthogonal directions. The resultant reading from an RF pulse in this configuration will uniquely identify a point in 3D space. This information is recorded directly into the frequency domain, known as k-space, which can be inverse Fourier transformed to the image domain. Whilst the three direction electromagnets allow for any 3D point to be acquired, the most common scanning method involves using one of the electromagnets to specify a 2D slice for imaging; the other electromagnets allow for scanning across the 2D plane and are referred to as the phase-encoding and frequency directions. This form of imaging is known as 2DFT Cartesian imaging and remains the most commonly used technique as image slices can be interpolated between to form a 3D model.

Measurements across k-space are recorded row by row, with rows being along the frequency direction and so perpendicular to the phase encoding direction (fig. 2.1). With one RF pulse, a whole line of k-space can be acquired by varying the frequency electromagnet. The repetition time TR is the time taken to measure a single line of k-space, which is the time between one RF pulse and the next. Given that TR remains the same if some

points along a line are missing, undersampling techniques omit entire lines rather than individual points.

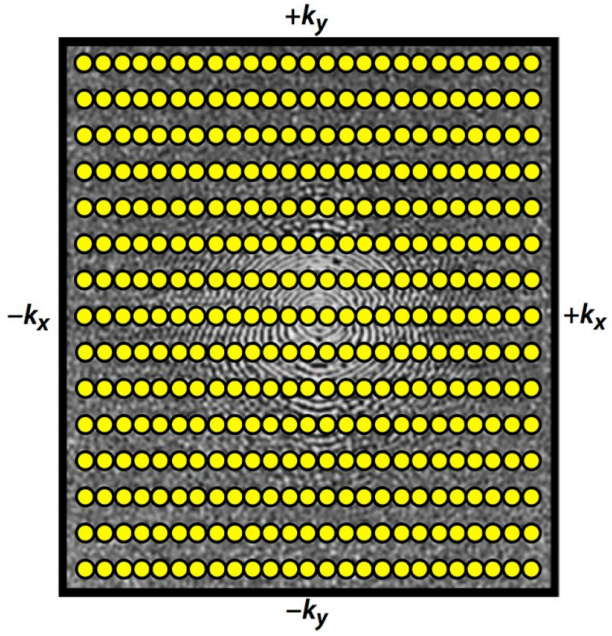


Figure 2.1: K-space measurements showing phase encoding (y) and frequency (x) directions [11]

2.1.2 Compressed Sensing

CS allows for significant undersampling of data and full recovery without aliasing artefacts despite violating the Shannon-Nyquist criterion. MRI is well suited for CS given the sparseness of MR images [3], thus allowing for many sparsifying transforms to be used. There are three key requirements to CS: a sparse domain, incoherent sampling and nonlinear reconstruction. A sparse dataset refers to one where the majority of the values are zero-valued. If data is sparse, then theoretically the zero-value measurements are not needed to capture the information of the dataset. CS follows this theory, requiring the signal to be sparse so that undersampling loses as little valuable data as possible. Since MR images are naturally sparse, they are easily compressed using sparsifying transforms such as the Direct Cosine Transform (DCT) [3].

The key to reconstruction without aliasing artefacts is in an incoherent sampling scheme. A conceptual understanding of this importance is shown in fig. 2.2. A uniform sampling scheme on a signal leaves aliasing artefacts which are impossible to differentiate from the

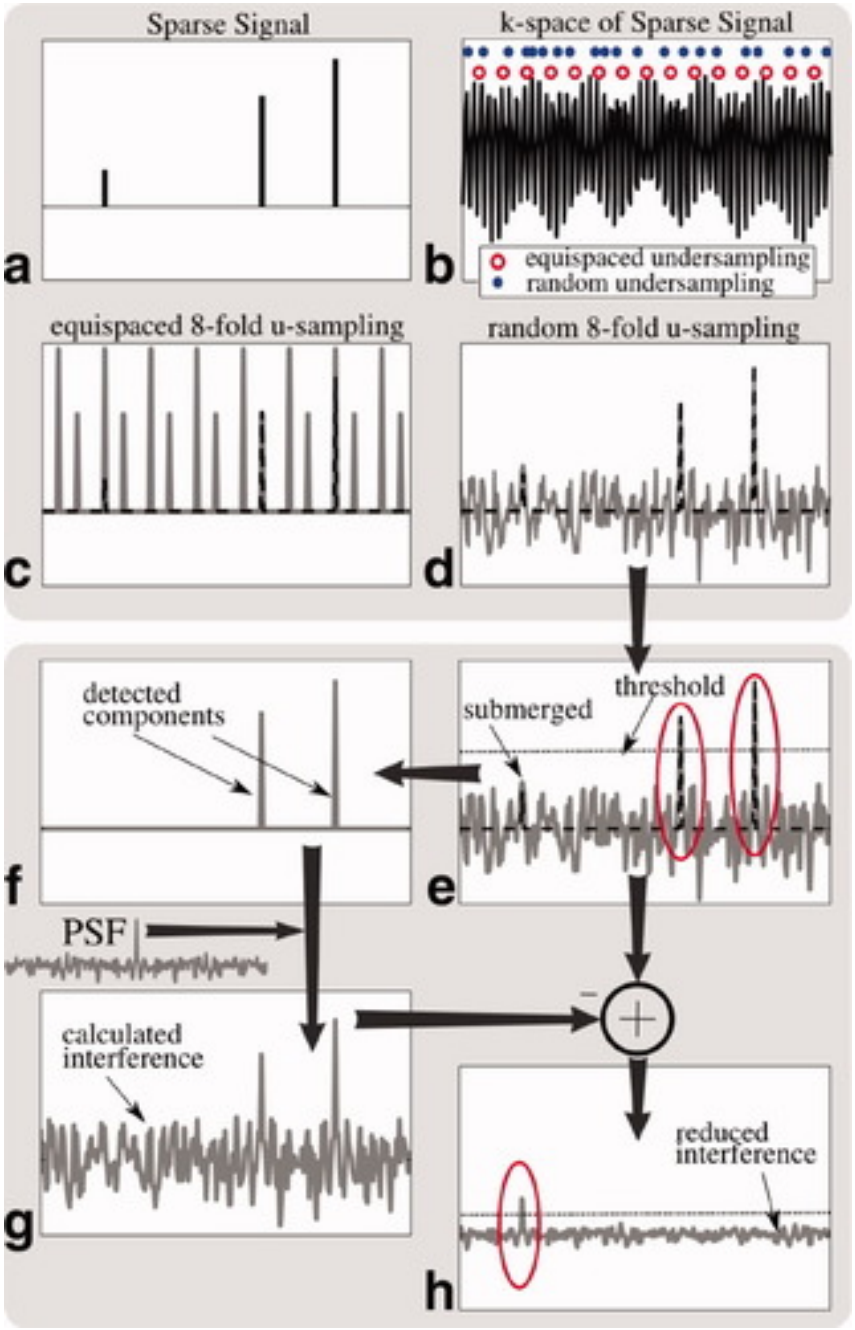


Figure 2.2: Conceptual visualisation of compressed sensing [3]

true signal; random sampling reveals the signal with superimposed noise. By identifying components of the true signal, calculating the expected interference from these and removing it from the measured data, weaker components of the underlying signal can be extracted. With MRI, sampling schemes are limited by the capabilities of the scanner and should be designed to reduce scanning length. As mentioned previously, benefits only arise from omitting entire lines of k-space, and given that k-space data is primarily at the origin, undersampling should be designed to keep the majority of this central data whilst being possible to implement on an actual scanner. The undersampling amount is inversely proportional to the acceleration factor of the technique (0.5 or 50% undersampling corresponds to an acceleration factor of 2).

Nonlinear reconstruction is required to find the sparsest image which matches the measured data. If there is an image m and some sparsifying transform Φ, minimising the L1 norm of this promotes greater sparsity in the image (minimising the L2 norm promotes many small but non-zero coefficients). Given y as the measurements of the undersampled data, an undersampled Fourier transform $\mathcal{F}_u$ of m should be as close to y as possible. As eq. (2.1) has multiple solutions, the goal of CS is to find the sparsest solution. Traditional optimisation techniques along with newer deep learning ones exist to solve this step (see section 2.2.2).

$$\begin{aligned} &minimise\ ||\Psi m||_1 \\ &\quad s.t.\ ||\mathcal{F}_u m - y||_2 < \epsilon \end{aligned} \tag{2.1}$$

2.1.3 Generative Adversarial Networks

The basic GAN (fig. 2.3) consists of two neural networks, a discriminator D and a generator G. The generator can be thought of as a function, which takes some random input (or latent code) z and generates an image $G(z)$ from it. Meanwhile, the discriminator is another function which takes some image x and outputs the probability of it being real or generated $D(x)$. Both networks start out untrained and learn synchronously, allowing both networks to improve each other with the goal of having the generator being capable of producing images indistinguishable from real ones. Networks must start untrained to provide suitable training gradients; a highly trained discriminator will have high certainty on decisions and thus cannot provide appropriate training direction despite the generator attempting to improve. By keeping the discriminator at a similar level of performance throughout, the generator is always given a suitable gradient to train upon. Variations

on this have been noted in section 2.2.3.

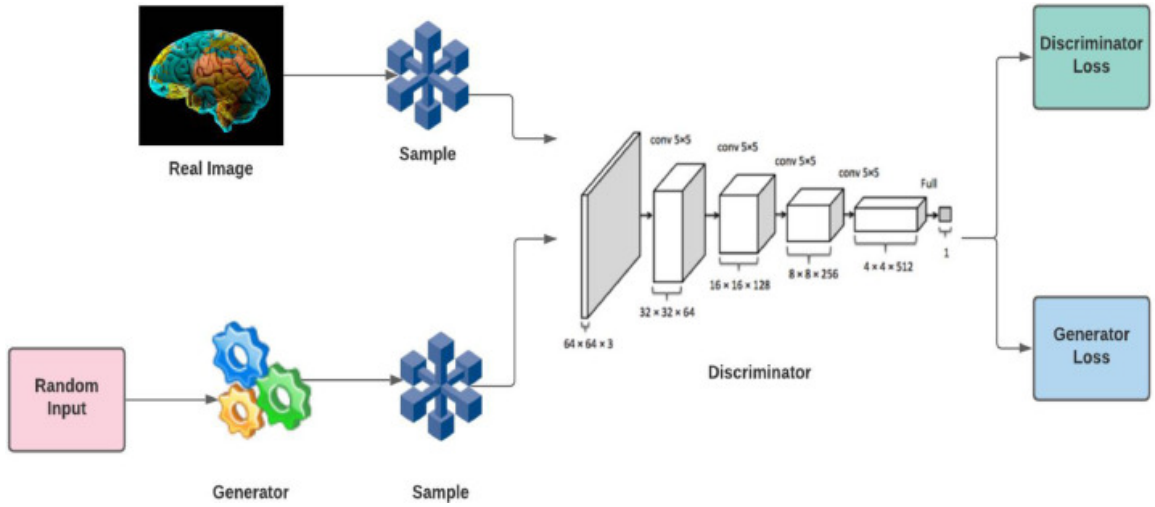


Figure 2.3: Diagram of a vanilla GAN [12]

The training of the generator and discriminator form a minimax problem. If p_{data} is the distribution of the real data and p_z is the distribution of the latent code, the discriminator aims to maximise the probability of correctly assigning labels such that x is from p_{data} and $G(z)$ is from p_z. Similarly, the generator aims to maximise the probability of $G(z)$ being identified as p_{data}, which can be thought of as minimising $G(z)$ being assigned p_z. This is written mathematically in eq. (2.2).

$$\min_{\Theta_G} \max_{\Theta_D} \mathcal{L}(\Theta_G, \Theta_D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_{\Theta_D}(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_{\Theta_D}(G_{\Theta_G}(z)))] \quad (2.2)$$

A variant of GANs known as a conditional GAN is particularly well suited to the application of CS reconstruction. Given some condition (the undersampled image), the GAN learns the mapping to the fully sampled image through training. The long reconstruction time of traditional CS is moved to the offline training period, thus active reconstruction time can be severely reduced.

2.1.4 Reinforcement Learning

Reinforcement learning is the third category of machine learning, alongside supervised and unsupervised learning. A network aims to learn optimal actions to perform in a given situ-

ation through repeated trials and rewards based on performance, with the aim to explore the actions possible in given situations and determine which provide the greatest reward. RL problems are typically modelled as a Markov Decision Process (MDP) (fig. 2.4), where an agent observes the state s of its environment and can opt to make some action a which results in a new state s' [13]. This move results in some reward $r(s, a, s')$ which is correlated with performing this action given the state observed. Through repetition of this process, the agent should be able to learn the optimal policy π to determine actions and thus maximise its reward.

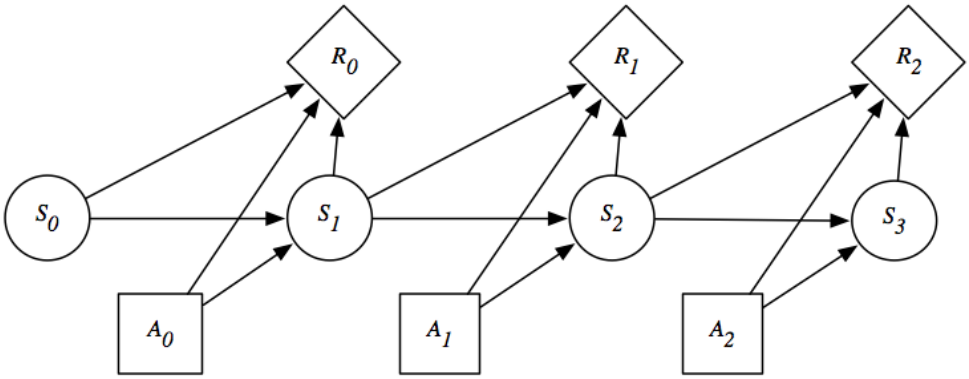


Figure 2.4: Flowchart of state, action and reward [14]

Training algorithms within RL can be divided into three categories: value functions, policy search methods and hybrid methods. Value-based methods are built upon an estimation of the value of being in a particular state, which can also be viewed as the potential reward. A quality function $Q(s, a)$ describes the expected reward given some initial action a in state s following pi (eq. (2.3) [13]). The value function consists of the immediate reward from this action r summed with the maximum of the value function out of all possible actions in the next state s'. Through training, these values will converge to more accurate ratings and allow the model to optimise its action choice, with the neural net acting as the map from state to Q value.

$$Q(s, a) = r(s, a) + \gamma \max_{a} Q(s', a) \tag{2.3}$$

Policy search methods do not consider a value function and instead seek to directly learn the optimal action policy. The neural net maps the policy, with parameters being updated during training to maximise return based on the policy [15]. Model outputs can be represented as a probabilities of taking different actions; a discrete action space would give

individual probabilities whilst a continuous action space would give a mean and standard deviation for each action. Gradient methods are typically employed for parameter updating, with an estimator of the gradient used for this situation known as the REINFORCE rule [16].

Actor critic methods are a hybrid of value functions and policy search techniques. The actor learns the policy using feedback from the critic, which contains the value function. Compared to plain policy search, actor critic methods result in less variance and instead more bias due to the response from the critic. The actor can be thought as of as improvement to the information being used to train standard policy search methods.

2.2 Related Work

A brief overview of the developments in improving MRI speed is given. Then, the need for CS is covered along with the problems traditional approaches have faced. An overview of GANs and their key developments has been covered before reviewing some of the research in the reinforcement learning area. The review concludes with an exploration of hyperparameter optimisation techniques and the identified gap in research.

2.2.1 Developments in MRI

MRI has remained as one of the most versatile imaging modalities available to clinicians to date, being the best imaging option for a vast number of applications. Through a combination of altering pulse sequence timings and the use of contrast agents applied to a patient, MR images provide superior soft-tissue contrast to all other imaging techniques [17]. This benefit is also seen when attempting to study organ and tissue functions, a process which can be performed non-invasively due to the nature of MRI [1]. With Jaspan, Fleysher and Lipton noting CT scans as the closest in non-invasive performance to MRI [17], the lack of ionising radiation from the latter makes it a far safer choice for patients.

However, a key issue limiting greater usage of MRI lies in the long scanning duration. Hollingsworth reported a cost of £350 - £500 per hour for MRI machine usage [1] and whilst other studies have not reported actual values, they have all noted the expensiveness of the procedure as a significant obstacle [3, 18, 19]. Similar to the exposure time for a camera, patients must remain still during the scanning procedure to prevent the ap-

pearance of movement artefacts. Whilst an attempt to reduce artefacts through holding their breath might have some effect for abdominal imaging [20], cardiac imaging presents a more difficult issue [21]. For patients with breathing difficulties, this becomes a much greater obstacle.

With the main difficulty being the acquisition time, research has been directed towards reducing this duration. The literature can be divided into two directions: fully sampled and undersampled techniques. The former primarily utilises multiple RF pulses per TR to acquire multiple lines of k-space in the same time. Mansfield's echo planar imaging [22] and Hennig et al.'s rapid acquisition with relaxing enhancement [23] both fall under this category; techniques in this category which have been adopted into clinical usage have been recorded by McRobbie et al. [10]. Whilst additional research exists in this area, advances are severely limited by the physical capabilities of existing clinical scanners; measuring the full k-space is limited by a physiological restriction from the patients [18].

Undersampling techniques are not restricted by such a limit, as they attempt to recover the full definition image from a partial measurement; Partial Fourier Imaging (PFI) and Parallel Imaging (PI) are the main methods in this category which have been adopted for clinical use. PFI exploits the complex conjugate symmetry present in k-space [24] to only require half the data to be recorded, although practical usage requires 60% for a robust reconstruction [1]. PI utilises multiple RF coils to acquire greater information at locations near to the coils, supplementing the missing data from undersampling [25]. Modern clinical scanners have incorporated PI although Hollingsworth's claims that PFI is also used does not agree with Jaspan [1, 17]. Both methods have been reported to suffer from a reduced Signal-to-Noise-Ratio (SNR), along with hard limits to their acceleration factors [20].

2.2.2 Studies on CS

In recent years, a new undersampling method was proposed known as Compressive Sensing (CS) [2]. Given a sparse image, if incoherently sampled, the image can be reconstructed with a nonlinear algorithm using significantly less data without aliasing artefacts. MRI meets both requirements of CS and as such, has been identified by many researchers as the next step for reducing scanning duration [3]. As CS is composed of three requirements, developments in the area can be categorised into those fields.

The sparseness of an image has been reported to influence the upper bound on the acceleration factor. Whilst theoretical undersampling limits have been derived [26], a practical limit of two to five times the number of sparse coefficients has been reported in multiple studies [1, 27]. With this in mind, research has delved into the evaluation and development of sparsifying transforms which would maximise sparsity. Within the CS-MRI area, wavelet transforms and Discrete Cosine Transforms (DCTs) have both seen significant usage although studies tend to report no performance difference from using one over the other; only Lustig, Donoho and Pauly found a slight increase in performance when using wavelet transforms [3]. The primary research direction here revolves around the use of dictionary learning [28]; a basis for a sparse transform is developed specifically for a type of signal to ensure the sparsest representation possible. The performance benefits noted by studies exploring dictionary learning in this field make it a viable approach [29, 30] however it will not be the focus of this study.

The importance of a random sampling scheme, unlike the uniform one used in PFI and PI, has been covered in section 2.1.2 however additional problems arise when practical usage is considered. The scanning sequence of a clinical scanner is limited by the smoothness of the k-space trajectory [3] and whether it actually results in a reduced scanning period. A large amount of research has been conducted on variable density sampling; using prior knowledge of the k-space data and how the majority of low frequency data is concentrated at the origin, sampling schemes can increase the density of readings at this location whilst reducing it elsewhere to undersample the data without losing vital information. Success has been found with this method with Cartesian, radial and spiral imaging trajectories [31, 32]. Other methods, such as Monte-Carlo based sampling schemes [3], have also been proposed although less research exists which verify the performance improvements they noted. Developments in this area require in-depth knowledge of the capabilities of a MRI scanner and so this field has not been prioritised by other researchers. To replicate viable undersampling techniques, studies have mostly used Poisson and Gaussian (1D and 2D) masks which provide the incoherence needed for CS [7].

By far, the greatest limitation of CS has been noted to be the reconstruction step. Many studies have explored various iterative algorithms for optimisation of reconstruction, with varying degrees of success [3]. The iterative nature of the problem led to the application of a variety of algorithms in attempts to reduce the reconstruction time and improve performance [33, 27, 34]; despite the wide variety of research in this area, reconstruction times with traditional CS remained over 30 minutes [17]. Significant improvements arose

when deep learning was introduced to the problem; reconstruction speeds were reduced to milliseconds with comparable accuracy, exceeding the performance of using dictionary learning [35]. The majority of research utilises convolutional neural nets for their suitability to image generation problems, given their ability to learn significant features and structures [36].

2.2.3 A Brief Review of GANs

GANs were first introduced in 2014 [4], utilising a pair of competing neural nets to encourage greater performance. With the rapid growth of this field as more applications are found, a vast amount of literature exists with various approaches taken to categorise it all; a compilation of many of the GANs made until the end of 2018 was recorded by Hindupur [37]. A brief overview of the history of GAN development will be provided before focusing on the main limitations and its applications to MRI. More detailed reviews are referenced for the reader's interest. [38, 39].

Following the development of the vanilla GAN, conditional GANs were introduced to allow for generation of an image based on additional information rather than random noise [40], a feature used extensively in CS-MRI [5]. Typical usage passes the conditional information to both the generator and discriminator [41] although success has been found when passing only to the generator [7]; the context of the application is important in this decision. Semi-supervised [42] and auxiliary classifier GANs [43] both aid in supervised classification problems, which the literature indicates is another key direction of research. BiGANs and VAEGANs implement an additional encoder network which allows for reversing the mapping which takes some latent code to an image [39, 44]. Diagrams of these different GANs are shown in fig. 2.5.

Three key issues have been identified in the literature regarding GAN development: mode collapse, vanishing gradient and training convergence. Vanishing gradient is a known problem in general gradient descent techniques, where the gradient of the loss becomes too small to provide any training direction. Mode collapse occurs when the generator produces similar or identical samples regardless of noise, thus failing to learn different modes [38]. Use of the Wasserstein distance has been reported to yield improvements in various studies [45], becoming a staple in GAN development. Further work was performed by Gulrajani et al., who noted reduced discriminator capacity from this method and so incorporated an additional penalisation term to the gradients [46]; verification of these

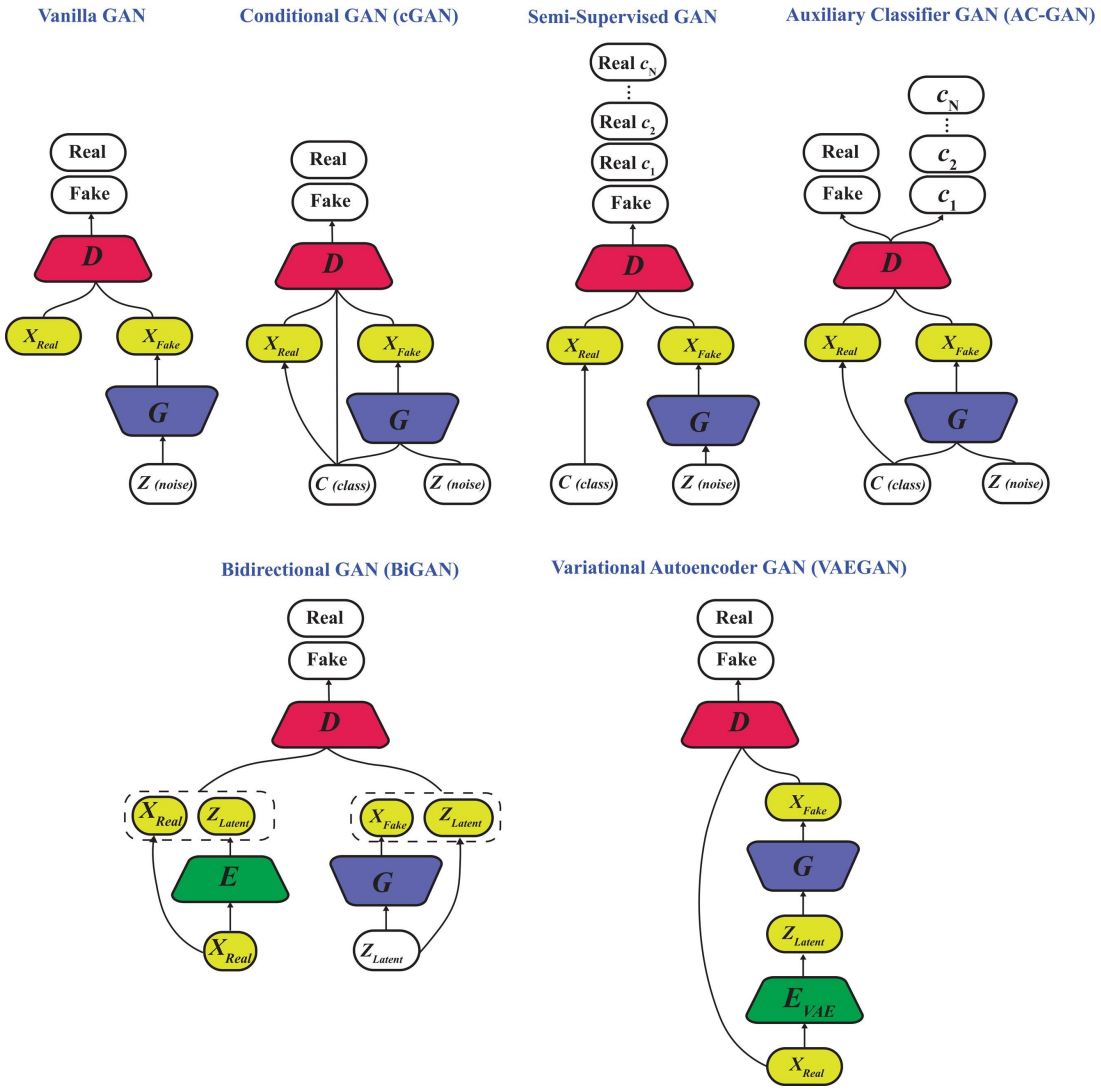


Figure 2.5: Structural diagram for different types of GANs

improvements has not been found.

Alternative distance metrics are a common area of research to improve training convergence and provide suitable gradients for training. Arora et al. proposed a neural net distance which improved convergence however suffered at providing useful gradients [47]. Research has also explored the use of additional loss metrics to provide an improved gradient for better performance (table 2.1). An optimal combination of these loss functions has not been identified in the literature, although work around CS-MRIs tends to use the image, spectral and perceptual losses along with the adversarial loss [7, 5].

Table 2.1: Different loss metrics used in generator training [39]

Loss	**Description**
Adversarial	Loss defined by the discriminator
Image	Typically L2 loss in the pixel domain
Spectral	Typically L2 loss in the frequency domain
Perceptual	Loss given by a visual similarity network (VGG-16)
Total Variation	Reduces noise, used in super resolution
Saliency	Measures difference in saliency maps
Edge	Measures difference in edges

Architectural design of GANs has also been an area of consideration within research. Deep Convolutional GANs (DCGANs) [48] found more success with both accuracy and stability, utilising convolutional layers; these have generally been accepted as a superior architecture to dense layers for image data, with its ability to identify image features through its layers [49, 50]. Springenberg et al. explored the use of convolutional layers to replace the typical max-pooling layers used in convolutional nets [50] however there is no indication that other researchers have attempted to replicate this. U-Nets [51] were used by Yang et al. for the development of DAGAN [7], a GAN designed for CS-MRI reconstruction, noting great performance increases as a result. The use of skip connections lends itself to the application of CS-MRI well, with the goal being to add missing information rather than recreate the entire dataset. Additional research in this area reports a similar approach [52], suggesting the incorporation of U-Nets to provide significant improvement to the standard convolutional net.

2.2.4 Advances in RL

With the large interest in reinforcement learning in recent years, the field of research has expanded greatly. Salvador classified development into three different areas: flat,

hierarchical and metal reinforcement learning techniques [53]. For this review, flat RL techniques will be covered which have a relatively straightforward action space for consideration; these have been sub-divided further into value function based methods, policy search methods and actor-critic ones [13]. Developments within each area will be covered to provide a brief overview of progression in research.

Value function methods originated from Watkin's and Dayan's Q-learning method [54], which described the techniques used to assign values to actions and states to allow an agent to make optimal decisions. The combination of deep learning with this technique gave Deep Q Networks (DQNs), pioneered by Deepmind where superhuman performance was recorded across multiple Atari games [55]. The addition of experience replay to the training procedure allowed for faster learning in situations where states would often be repeated. Double DQNs were then introduced in the same year [56] to improve upon the accuracy of value estimations, leading to greater performance in several of the Atari games featured in Deepmind's study. Dueling DQNs were also introduced with similar performance benefits [57], also using a second network but instead modelling the advantage and value functions. Many more variants have been introduced since but the above three represent the most significant innovations in the area.

Vanilla policy gradient, also known as the reinforce method, was found to be the simplest policy search method introduced by Connell and Mahadevan [58]. Through using the REINFORCE rule to provide an estimate for the gradient, the model learnt an optimal policy to dictate actions using gradient descent methods for convergence. A development upon the basic policy gradient algorithm known as Trust Region Policy Optimisation (TRPO) [59] has been noted by various studies for providing significant improvements when optimising more complex networks. The same authors provided further studies which revolved around the development of a simpler network with more generality in the form of Proximal Policy Optimisation (PPO) [60]. These methods represent the most commonly used variants, although novel research in the area has produced many more since.

Known as actor critic methods, the final category of flat RL techniques utilises a combination of both value and policy techniques, typically utilising two networks to do so. First introduced in 1999 by Konda and Tsitsiklis [61], significant developments were not made until the introduction of more novel techniques in the surrounding areas. The use of asynchronous training methods found significant improvements in both training performance and training time required [62]. Further performance benefits were found through

incorporation of a double DQN variant within an actor critic framework [63]. Whilst developments in this area tend to derive from improvements in its composites, research solely on the improvement of the actor critic procedure exists [64].

2.2.5 Hyperparameter Optimisation

The increasing complexity of neural networks has lead to a greater number of hyperparameters which require tuning to optimise performance. Most literature requiring hyperparameter optimisation has utilised some form of grid search, or full factorial design [65]. Through exploration of a finite and discrete set of hyperparameter combinations, an optimal location can be identified, although this method would not scale when faced with multiple parameters with a variety of options [66]. Bergstra and Bengio proposed an improvement on standard grid search with the random grid search [67], allowing for greater flexibility in the search algorithm (fig. 2.6). Exploration can be focused on more important hyperparameters with greater significance to the performance results. Additional optimisation could be applied to random search to improve convergence to near optimal values [68], although it was noted that Bayesian optimisation techniques were still faster for convergence.

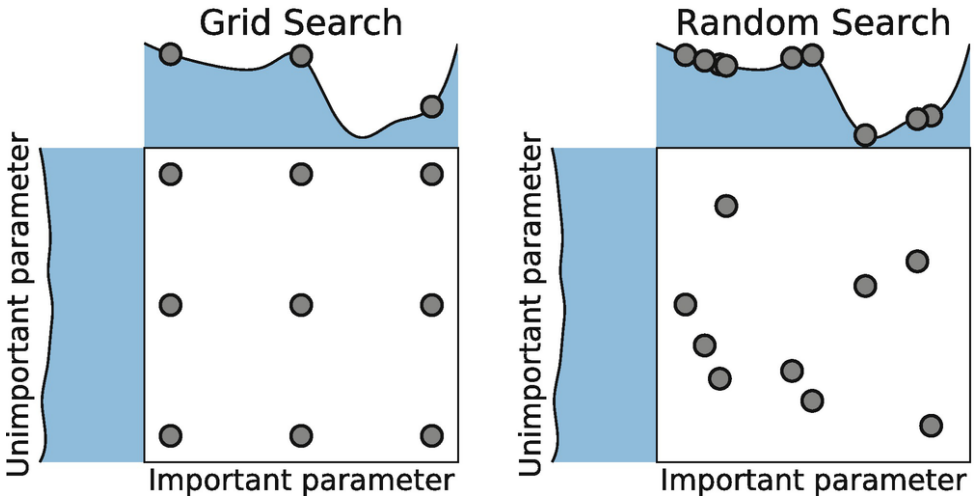


Figure 2.6: Comparison of grid search to random search [67]

Bayesian optimisation techniques have experienced great success in literature [69], providing an automatic procedure based on Bayes Theorem to optimise parameters without expert knowledge required. Success has been reported in multiple areas, including image classification [66] and speech recognition [70]. Development of Bayesian optimisation

techniques has explored the incorporation of greater knowledge of the training process, removing the blackbox surrounding the procedure. Swersky's Freeze-Thaw algorithm incorporated knowledge of learning curves to predict if running trials had the potential to exceed the best recorded performance, with early termination otherwise to reduce overall runtime of the optimisation procedure [71]. Attempts have also been made to automate the selection of training algorithms and preprocessing techniques along with hyperparameters [72], thus reducing the need for manual design.

Evolutionary algorithms have been presented as another avenue of research regarding optimisation techniques. Through the use of a mutating and reproducing population with random hyperparameters, the best performing ones could be identified for further mutation and potential improvement. Whilst there have been comparisons between different population-based approaches [73], the most well-known approach is noted by many to be the Covariance Matrix Adaption Evolutionary Strategy (CMA-ES) [74]. Compared to Bayesian techniques, the exploration rate of CMA was found to completely surpass it [75]; the naturally parallel nature of population algorithms allows for significantly more configurations to be explored given hardware capable of running trials in parallel.

Additional work has been found on the use of reinforcement learning in this area [8, 9], where Q-learning has been applied to guide the model's parameters to optimal values with great success. Both studies found significant performance improvements although details of the implementation were not found. With regards to active adaption of hyperparameters such as weightings for loss functions, only one study was found which utilised a mathematical balancing of weights [76]. No efforts have been made to introduce adaptive weighting using reinforcement learning.

3 Preliminary Work

A brief background on DAGAN's content loss is provided before detailed analysis of the individual losses and their correlation with each other. Importantly, perceptual loss is evaluated for usefulness as a reward metric.

3.1 DAGAN and Content Loss

Since this study has been built upon the work of DAGAN [7], a brief overview of the architectural design and the loss functions used has been provided. Convolutional layers were used for the discriminator's classification task as is standard for computer vision problems. The generator utilised a U-Net style architecture with skip connections along with batch normalisation and leaky ReLU activation functions, with a hyperbolic tangent activation function for the final layer. DAGAN could be thought of as a variant upon the standard conditional GAN; the undersampled image was passed as conditional input u_x however the reconstructed image was taken as $\hat{x}_u = G(x_u) + x_u$ rather than just the generator output. The generator acted as a refinement network which was trained to fill in the missing information rather than recreate the entire image, thus improving robustness against vanishing and exploding gradient issues. One of their key novelties was in the multi-part content loss function used for generator training, shown in fig. 3.1.

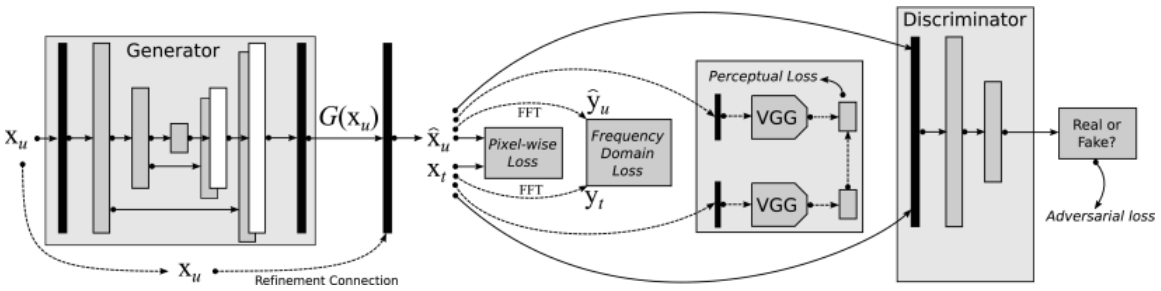


Figure 3.1: Architectural design of DAGAN [7]

The equations used in DAGAN's novel content loss for generator training have been listed below [7]; x_t is the ground truth image with $\hat{x}_u$ being the reconstruction, y denotes the frequency domain version of x and f_{vgg} is the VGG network developed by Oxford for

image recognition [77]. Normalised Mean Square Error (NMSE) of the image in both the pixel domain (eq. (3.1) or g_mse) and the frequency domain (eq. (3.2), or spectral loss g_fft) are used, along with a perceptual similarity neural net developed by the Visual Geometry Group (VGG) at Oxford University [77] (eq. (3.3) or g_vgg). Along with the adversarial loss (eq. (3.4)), the content loss combined all losses to provide a total loss eq. (3.5). Weights (α, β, γ) are assumed to have been empirically chosen, with the main focus being to bring the losses to similar scales. These weights remained static whilst a learning rate decay of 0.5 every 5 epochs helped the model to converge to an optimum point.

$$\min_{\Theta_G} \mathcal{L}_{iMSE}(\Theta_G) = \frac{1}{2}||x_t - \hat{x}_u||_2^2 \tag{3.1}$$

$$\min_{\Theta_G} \mathcal{L}_{fMSE}(\Theta_G) = \frac{1}{2}||y_t - \hat{y}_u||_2^2 \tag{3.2}$$

$$\min_{\Theta_G} \mathcal{L}_{VGG}(\Theta_G) = \frac{1}{2}||f_{vgg}(x_t) - f_{vgg}(\hat{x}_u)||_2^2 \tag{3.3}$$

$$\min_{\Theta_G} \mathcal{L}_{GEN}(\Theta_G) = -\log(D_{\Theta_D}(G_{\Theta_G}(x_u))) \tag{3.4}$$

$$\mathcal{L}_{TOTAL} = \alpha\mathcal{L}_{iMSE} + \beta\mathcal{L}_{fMSE} + \gamma\mathcal{L}_{VGG} + \mathcal{L}_{GEN} \tag{3.5}$$

The NMSE in the image domain, Peak Signal-to-Noise-Ratio in dB (PSNR) and Structural Similary Index (SSIM) were used for evaluation of the model's performance; all three are standard metrics used for evaluation of computer vision performance. Their refinement net results were verified by this study and showed that incorporation of additional loss components resulted in greater performance. The authors suggested that each component contained information missing from others which would be useful in training, although the degree of importance of each was unknown.

3.2 Observations of Image and Spectral Loss

In order to determine if adjusting the importance of losses during training could provide performance benefits, the relationship between them was investigated. Given the small margin for improvement at low small undersampling ratios, analysis and experimentation was focused at 10x undersampling (10% mask) for greater insight into the usefulness of improvements. Training and testing was performed on 20% of the dataset used in DAGAN (see section 5.1). Figure 3.2 shows a detailed view of how the image and spectral losses varied during training. There was a clear correlation between the two losses, with both following the same general trend. Interestingly, their gradients differed at certain points, suggesting that some generated images were closer with regards to the image than the frequency domain, or vice-versa. This reinforced the assumption that each loss term contained information that was missing from the other.

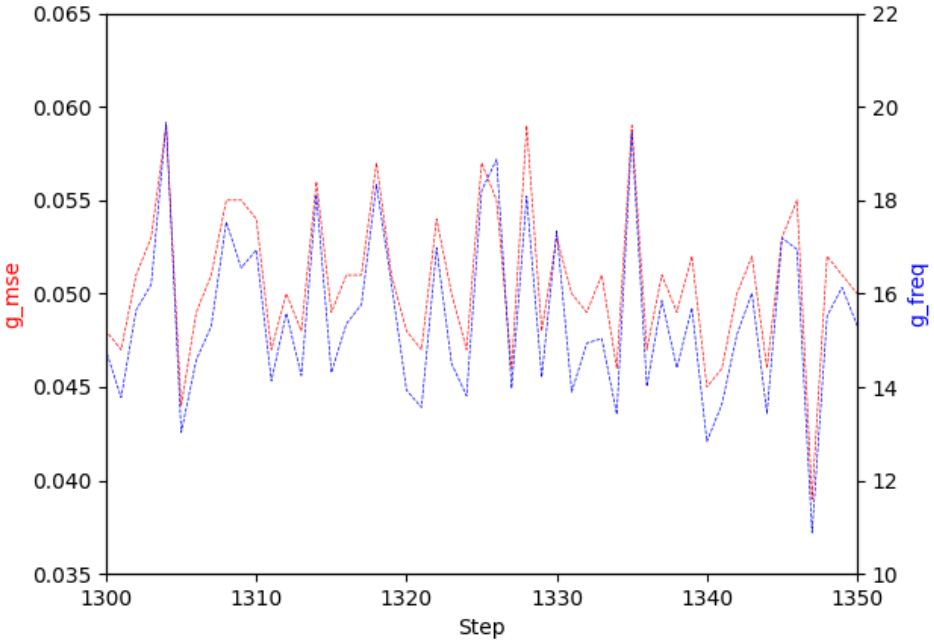


Figure 3.2: Variation of image and spectral loss during training with 10% mask

Comparison of the two loss functions is shown in fig. 3.3. A strong positive correlation was easily visible, linear up until $g\_mse = 0.07$ before the signal became noisier. Early stages of training would have generated images with little relevance to the input, thus leading to image and spectral losses with less correlation. Measures of correlation were

calculated using Pearson's and Spearman's. Pearson's (eq. (3.7)) uses the covariance (Cov) along with the standard deviation of x (σ_x) and y (σ_y) to determine the strength of a linear relationship between the variables. Spearman's (eq. (3.6)) orders each set of data and assigns a rank from 1 to n, with d_i indicating the difference between points of data at the same rank. The resulting coefficient gives the strength of a monotonic relationship, which is particularly useful for non-linear data. Values can range from +1 (strong positive correlation) to -1 (strong negative correlation) with 0 indicating no correlation. Whilst it was expected that both losses would experience a strong positive correlation, the imperfectness of this correlation would reveal the degree of information which differed between loss components.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{3.6}$$

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X, Y)}{\sigma_x \sigma_y} \tag{3.7}$$

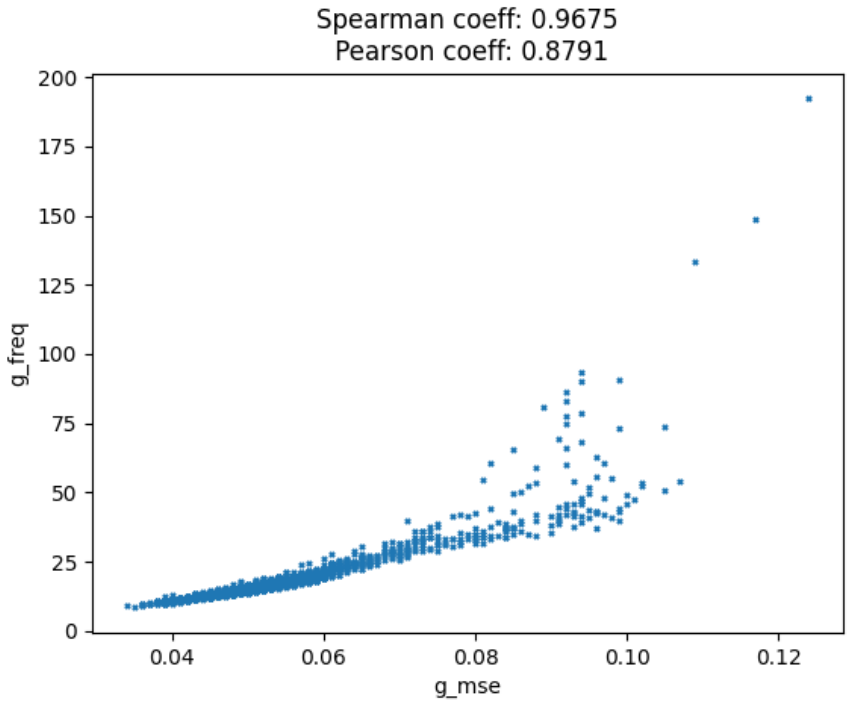


Figure 3.3: Comparison of image against spectral loss with Pearson and Spearman correlation coefficients

Calculating correlations using Spearman's rank correlation coefficient and Pearson's correlation coefficient gave 0.9675 and 0.8791 respectively. Whilst the data contained a strong linear portion, the non-linearity of the second half of the diagram suggested a monotonic relationship which would be better identified with Spearman's. This theory was shown to be correct, given by the greater Spearman's coefficient, however Pearson's identified a strong linear component regardless. The strong correlation given by Spearman's implied that for each image generated, if there was reduced similarity in the image domain, the spectral domain also differed by the same. There was some information which experienced improved image but worsened spectral loss and vice-versa, a sign that the inclusion of both losses was beneficial.

3.3 Interactions with Perceptual Loss

The relationship between perceptual loss (from the VGG network) and image and spectral losses were also investigated; a strong correlation would suggest the use of perceptual loss as the basis of a reward function for a reinforcement network as a suitable move. Figure 3.4 shows the relationship between all three losses. A strong correlation was clearly present between all three components, even within initial stages of the training where losses were more extreme. Given the large differences in scale between the losses, adaptive weighting could provide benefits in the form of automatically tuning weights to allow the generator to learn from all components effectively. For this to work, a strong correlation was needed between the perceptual loss and other components, especially considering that the perceptual loss was just the output of a neural network and did not follow a similar formulation to NMSE of image and spectral domains.

A closer analysis of image against perceptual loss is provided in fig. 3.5, along with Pearson's and Spearman's coefficients. From both the coefficients and a visual inspection, the losses indicated a strong linear correlation between image and perceptual losses. Spectral and perceptual loss (fig. 3.6) also showed a strong linear correlation which deteriorated above a spectral loss of 50. Correlations were weaker than that of the image-spectral relationship, primarily because perceptual loss would be calculated with a significantly more complex equation given by the neural net.

From the analysis above, it was evident that sufficient correlation existed to justify the use of perceptual losses as a reward element. In theory, the adversarial loss from the discriminator could also have been used as it would in theory provide a similar training

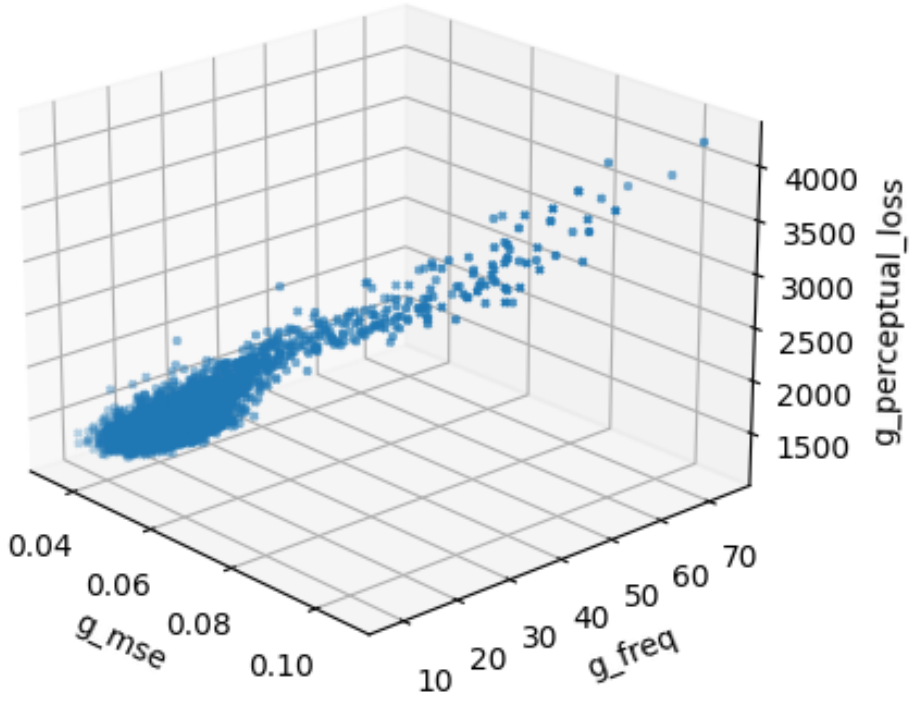


Figure 3.4: 3D scatter plot of image, spectral and perceptual losses

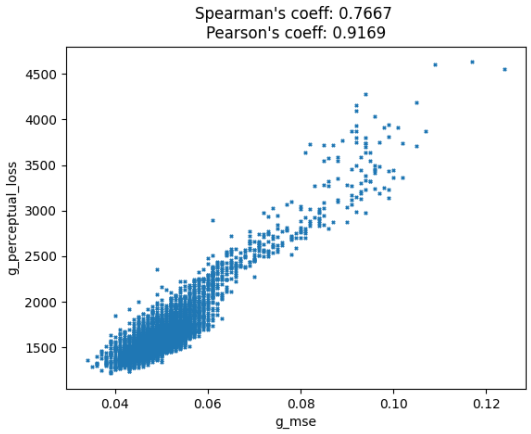


Figure 3.5: Image against perceptual loss

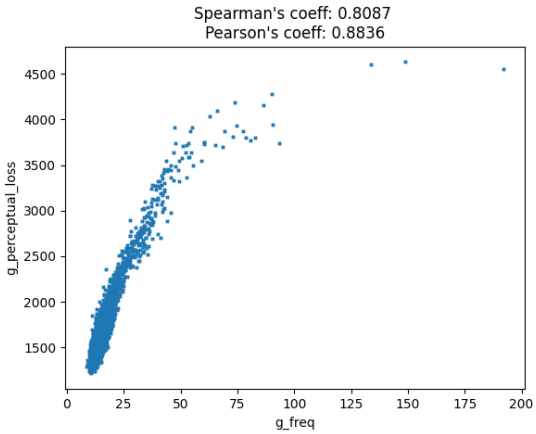


Figure 3.6: Spectral against perceptual loss

pressure. If the weighting for the perceptual loss (γ) was also being optimised, this would have been the only solution as the reward function must be separated from the losses so that the network does not arbitrarily lower all the weights to a minimal value to get the "best" performance; in this case, either option was viable. The VGG network provided greater insight into the visual similarity of the image to the target rather than just a probability of it being a real image, thus it was assumed that the reward function would be more accurate using this over adversarial loss.

4 Adaptive Weighting Design

Design of a reinforcement learning model required multiple considerations. One of the factors with greatest influence on performance was what state was visible to the network, along with how it could interact with it. The training algorithm used along with the reward function would also affect the performance significantly. After these areas are covered, details of how the model integrated with DAGAN along with details of its implementation are given.

4.1 Defining a State

As covered in section 2.1.4, a RL problem requires an environment for an agent to act within. Given that a significant proportion of reinforcement learning development has been centered around the gaming environment, a large part of the difficulty with this particular application was in designing a suitable environment to present to the agent (in this case, the model). Open source libraries in Python exist with pre-built environments for popular games designed to interact with Python's RL libraries, however none of these would be useful for this specific application [78]; an environment for neural network training was not available given the novelty of the application. Other game-based applications used a screenshot of the current game screen passed through a convolutional net to provide a suitable state for learning, however training a GAN did not provide any suitable imagery to usage (an image from the output would not provide any use for learning). Instead, the factors important to training had to be identified before being structured into a custom state.

This problem was approached considering the mindset of a person attempting to manually tune the weights. Since the RL network emulates a person learning, the factors important in making an informed decision needed to be given to the neural net as well. Considering the overall outcome was to provide the weights to be used for the next training iterations, the primary factors affecting that should be the reported losses (image, spectral and perceptual) from the previous iterations. The number of previous iterations k would be a hyperparameter which required experimentation to determine an appropriate value. Given that the model was choosing weights, the previous weights that were in effect would

provide additional information, especially when coupled with the losses that resulted from it. This situation bore resemblance to typical forecasting problems using deep learning and with those conditions, common practice was to also pass in the gradient of the data; this provided a sense of the general direction in which data was heading. In addition, given that the actions were later chosen to increment or decrement from the current value (section 4.2) rather than to give an entirely new value, the relative perception of data made more sense. Equations (4.1), (4.2), (4.3), (4.4) and (4.5) represent the stored components of state which are appended together in eq. (4.6) for a singular state corresponding to an action. n states are then considered during each parameter update (eq. (4.7)).

$$\boldsymbol{\alpha}_s = [\alpha_1, \cdots, \alpha_k] \tag{4.1}$$

$$\boldsymbol{\beta}_s = [\beta_1, \cdots, \beta_k] \tag{4.2}$$

$$\boldsymbol{i}_s = [i_1, \cdots, i_k] \tag{4.3}$$

$$\boldsymbol{f}_s = [f_1, \cdots, f_k] \tag{4.4}$$

$$\boldsymbol{p}_s = [p_1, \cdots, p_k] \tag{4.5}$$

$$\boldsymbol{state}_k = [\boldsymbol{\alpha}_s, \boldsymbol{\beta}_s, \boldsymbol{i}_s, \boldsymbol{f}_s, \boldsymbol{p}_s] \tag{4.6}$$

$$\boldsymbol{state}_{full} = \begin{bmatrix} state_k \\ state_{k+1} \\ \vdots \\ state_{k+n} \end{bmatrix} \tag{4.7}$$

Since the scales of the losses and weightings would vary on entirely different scales, a form of normalisation needed to be used. The most commonly used method was Z-Score normalisation which involved subtracting the mean and dividing by some factor of the

standard deviation to bring all values to an average of zero. Two issues were present with application of this method to the adaptive weighting problem. Firstly, the losses of the neural network were expected to steadily decrease over time from extremely large values at the beginning of training to much smaller values afterwards. The majority of readings would be situated at the lower end of the range however standardisation would result in extremely small values due to the large standard deviation. Both the mean and standard deviation would not be useful reflections upon the dataset and between various experimental configurations or even different applications of adaptive weighting entirely, the two statistics would be vastly different, requiring recalculation each time. Instead, the logarithmic function was used to bring all values to a similar scale. Whilst imperfect as it did not cover the intricacies of the relations in the dataset, all values would have the same magnitude. The exact equation is given in eq. (4.8) where the normalised version of x is given by $\tilde{x}$.

$$\tilde{x} = \begin{cases} \log(|x| + 1), & \text{if } x \geq 0 \\ -\log(|x| + 1), & \text{otherwise} \end{cases} \tag{4.8}$$

4.2 Discretising Actions

The original goal of this study was to implement adaptive weighting for all three of the extended loss components. However, given that each weight could either increase, decrease, or stay the same at each update, this lead to a total of 27 unique combinations of actions. The use of a discrete action space was desired so that analysis of the model's responses could be related to a finite set of actions. With more actions to choose from, even an action with a higher probability than all others would be unlikely to be picked when considering the sheer number of additional options to choose from. With this in mind, only two weights were considered which would lower the action space to 9 distinct moves. This was further reduced to 5 moves, as shown in table 4.1, which would still allow for all distinct actions as a combination of two moves.

The quantity by which to change each weight also required experimentation. Initially, changes were made by an absolute value of 1, with a small exploration range available in order to examine the behaviour of the network. This approach would not be viable for the use case of adaptive weighting as the chosen step size would need to be appropriate for all weightings being adapted, regardless of the scale. DAGAN's authors had used

Table 4.1: The discretised action space, with + as increase, = as stay the same, - as decrease for each weight

Number	α	β
0	+	=
1	=	+
2	=	=
3	-	=
4	=	-

weightings of 15 and 0.1 [7] which already presented problems when attempting to explore possible weightings with sufficient resolution. Furthermore, the established exploration range should be sufficiently big to allow the model to perform an exhaustive search and reach optimal values; the step size would need to be adaptive for this to work. As the specifics of the adaption required were unknown, a percentage-based approach was taken, where the change would be based on a fixed percentage of the previous weight. A similar balance regarding step size was still required; a small percentage allowed for greater resolution during searching due to the smaller steps however the resultant change in weighting may not provide significant effects for the model to learn from.

4.3 Training Method and Rewards

The choice of training method used was intentionally chosen with knowledge of it likely being sub-optimal. Section 2.1.4 noted that whilst research in this area has only utilised Q-learning, policy-based methods have been found to have greater success in situations where the value function was difficult to learn. The simplest policy method is the Vanilla Policy Gradient (VPG) method, also known as REINFORCE. By using this method, performance should benefit from only needing to identify a policy rather than the value function and the effectiveness of this proposal would be more easily discerned. VPG has known weaknesses, although the known solutions may not be applicable for this situation; experience replay is typically used to allow for training from examples earlier on in the procedure however the relevance of such actions may be limited given the rapidly changing state. Instead, this study has acted as a baseline from which further research can be performed to identify relevant developments in policy gradient methods.

The reward signal has been kept simple; the gradient of the perceptual loss is taken over the k instances and averaged, with an overall decrease in loss leading to a reward of +1, an

increase being -1 and 0 otherwise for that action. The model was set to make an action every k iterations, with AW-net performing parameter updates after a set number of episodes (which was also investigated for optimal values). Training occurred concurrently with the remainder of the GAN, without any specific synchronisation of events such as updating parameters at the same time.

4.4 Implementation

The flow diagram of DAGAN was provided in fig. 3.1 and the proposed modifications which allow for adaptive weighting are shown in fig. 4.1 and fig. 4.2. An additional model has been included (AW-net) to modify the weightings of DAGAN during training; the new network took in an input matching the state described previously, with experiments using 1 and 2 hidden layers and ReLU activation functions. Losses and weights are passed to AW after the logarithmic scaling mentioned in section 4.1. Equation (3.5) effectively became eq. (4.9), where $\mathcal{A}_x$ represents the x output from the adaptive weighting network.

$$\mathcal{L}_{TOTAL} = \mathcal{A}_\alpha \mathcal{L}_{iMSE} + \mathcal{A}_\beta \mathcal{L}_{fMSE} + \gamma \mathcal{L}_{VGG} + \mathcal{L}_{GEN} \tag{4.9}$$

TensorLayer was used for creating the VPG network, which is a high-level Tensorflow wrapper. The code for DAGAN was already written with this thus it was easiest to follow suit and keep the same codebase rather than redesign in another library entirely. As this study was built upon the work provided in DAGAN, the code used for this study has been integrated into the existing files. Specifically, the VPG network has been appended to the end of "model.py" and "train_mod.py" contains a modified version of the training process to include training of the reinforcement model. Experimental settings have been included in "config.py" with additional code written to parse and visualise the losses.

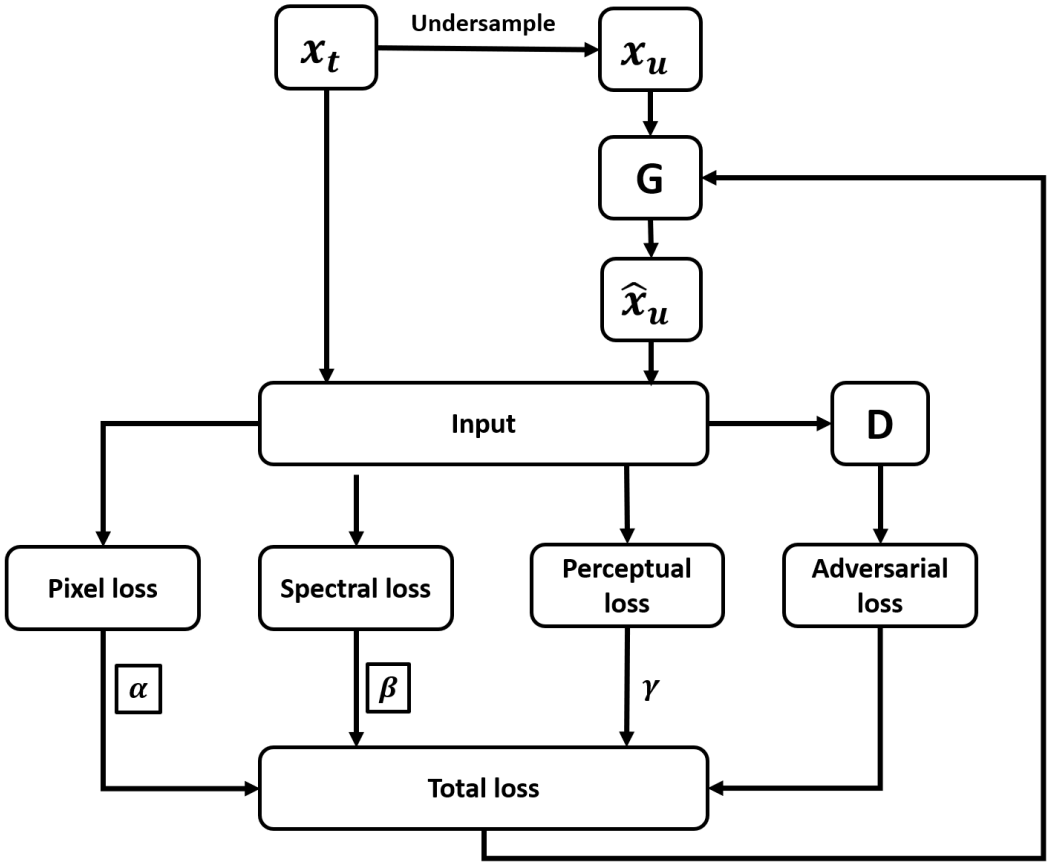


Figure 4.1: Diagram of modifications to DAGAN architecture

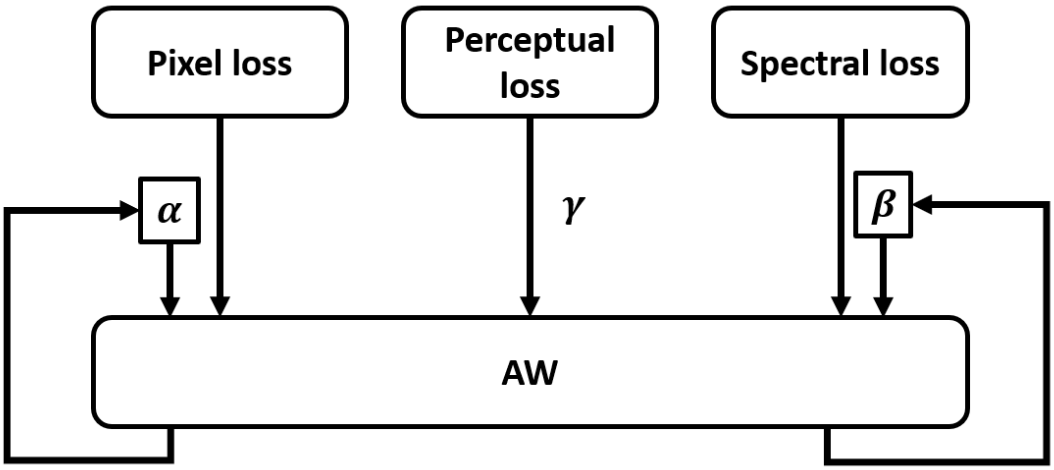


Figure 4.2: Diagram of AW-GAN architecture

5 Results

The results of the different experiment configurations are outlined below. Performance of the models was recorded along with the change in weightings and probability of actions over time.

5.1 Metric Comparison

Table 5.1 shows NMSE, SSIM and PSNR scores for different experiment configurations, with the first three trials representing the original DAGAN architecture on the full dataset, a zero-filled reconstruction on the reduced dataset and DAGAN trained on the reduced dataset. All trials which had improved performance compared to DAGAN trained with the same dataset were highlighted in bold. Metrics explained in section 3.1 have been used for performance evaluation. The majority of the training factors matched those of DAGAN; some reinforcement network parameters were varied between experiments to quantify their effects on performance:

1. **Dataset**: Whilst DAGAN was trained on the MICCAI 2013 grand challenge dataset, using 100 T_1-weighted MRI datasets for training (70% training and 30% validation) and tested on 50 datasets, the time required for training each model was over 24 hours. Performance attained from training with the large dataset was also high and left little room for further improvement. Considering this, all training for this study was conducted with 20% of the datasets (20 and 10 datasets respectively for training and testing). Training times were reduced to an average of 6 hours and there was a minor drop in performance. This allowed for improved testing of whether adaptive weighting was producing significant benefits which could make up for a reduced dataset.

2. **Episodes**: The number of episodes used for each update of the reinforcement network parameters was also varied between trials. Each episode was associated with a specific action and was used by the network to correlate between actions and resulting performance. Once a suitable number was determined, further experimentation used the same number of episodes.

3. **State size** k: The state size was also varied to determine optimal conditions. The last 20, 30 and 40 time-steps were considered before choosing an optimal value for further experimentation. A final configuration (*) removed all absolute values from its state and only considered the gradients of each parameter.

4. **Learning rate decay** d: DAGAN had originally chosen $d = 0.5$ for their experimentation. Initial trials used the same value however given the nature of adaptive weighting, the decay rate was reduced to 0.9 for further investigation.

5. **Step size** s: Actions were limited to adjusting one parameter by a specified step size. This was either an absolute value of 1 or a percentage increment/decrement by either 50% or 10%.

6. **Weight ranges** w_{min}, w_{max}: Weights were allowed to vary between a specified range. Initial experiments were limited to $[1, 20]$ before varying the minimum to 0.01 and the maximum to 50 and then 100.

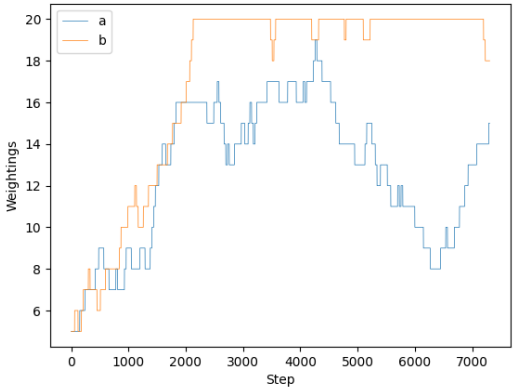
All models were trained with the refinement method developed by DAGAN, using a 10% Gaussian 1D mask. For all trials with adaptive weighting (and also DAGAN trained on the small dataset), α and β were initialised at 5 to evaluate performance from the same starting weight.

Table 5.1: NMSE, SSIM and PSNR metrics for different experiment configurations, with the first three being benchmarks

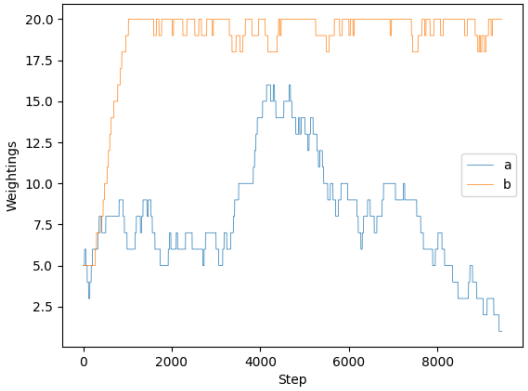
Dataset	Episodes	k	d	s	w_{min}	w_{max}	NMSE	SSIM	PSNR
Full	N/A	N/A	0.5	1	1	20	0.1644	0.9633	35.3650
Small	N/A: Zero-filled reconstruction						0.3350	0.7680	28.4711
Small	N/A	N/A	0.5	1	1	20	0.1794	0.9446	33.8818
Small	10	20	0.5	1	1	20	0.1745	0.9477	34.1218
Small	10	30	0.5	1	1	20	0.1800	0.9443	33.8540
Small	5	40	0.5	1	1	20	0.1733	0.9486	34.1783
Small	**5**	**30**	**0.5**	**1**	**1**	**20**	**0.1683**	**0.9516**	**34.4380**
Small	**5**	**30**	**0.9**	**1**	**1**	**20**	**0.1474**	**0.9611**	**35.5783**
Small	**5**	**30**	**0.9**	**50%**	**1**	**50**	**0.1593**	**0.9551**	**34.8907**
Small	5	30	0.9	50%	0.01	100	0.2024	0.9396	32.8200
Small	**5**	**30**	**0.9**	**10%**	**0.01**	**100**	**0.1613**	**0.9545**	**34.8060**
Small	**5**	**30**	**0.9**	**10%**	**0.01**	**50**	**0.1554**	**0.9584**	**35.1216**
Small	**5**	**30***	**0.9**	**10%**	**0.01**	**50**	**0.1452**	**0.9633**	**35.7129**

5.2 Variation of Weights

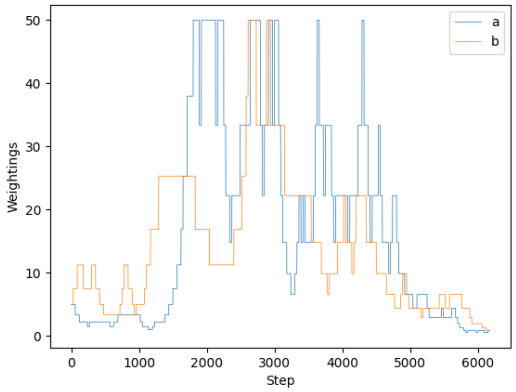
Figure 5.1 shows the variation of α and β during the training process for the top performing models. All of the models below were trained on the reduced dataset with 5 episodes per training and $k = 30$. The learning rate decay, step size and weight ranges are outlined in the captions. Likewise, fig. 5.2 shows the variation of probabilities for the respective models during training.



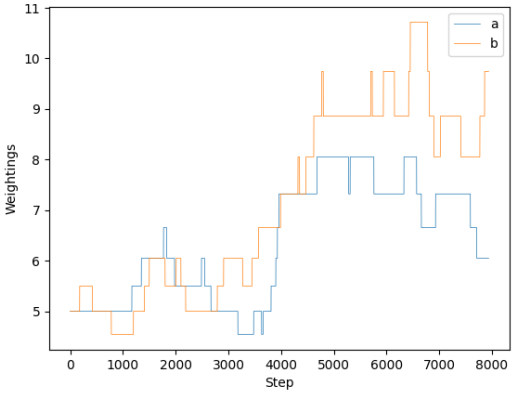
(a) $d = 0.5$, step $= 1$, $1 < w < 20$



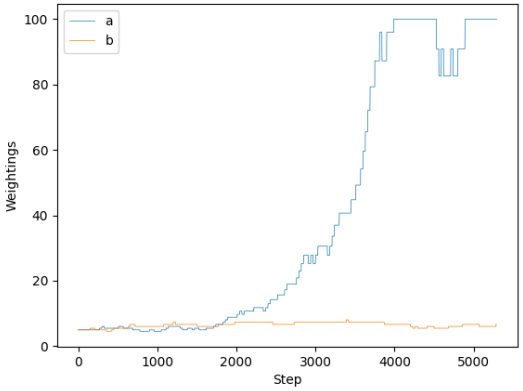
(b) $d = 0.9$, step $= 1$, $1 < w < 20$



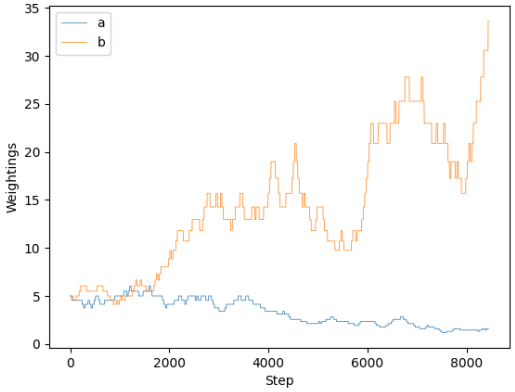
(c) $d = 0.9$, step $= 50\%$, $1 < w < 50$



(d) $d = 0.9$, step $= 10\%$, $1 < w < 50$

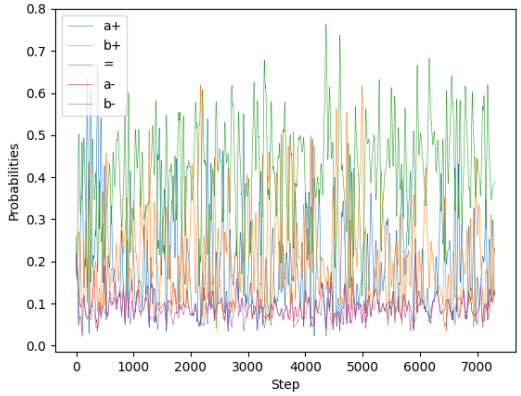


(e) $d = 0.9$, step $= 10\%$, $0.01 < w < 100$

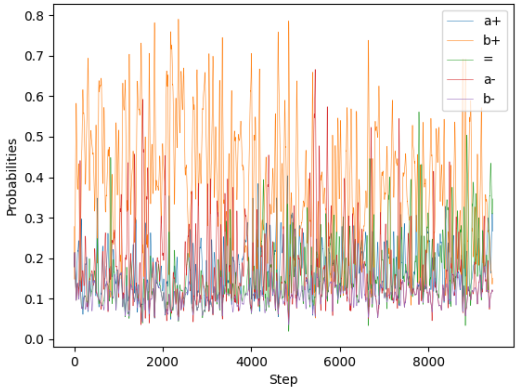


(f) $d = 0.9$, step $= 50\%$, $0.01 < w < 50$, reduced state

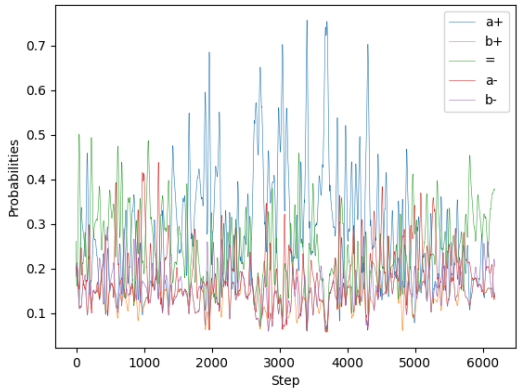
Figure 5.1: Variation of weights during training for different models



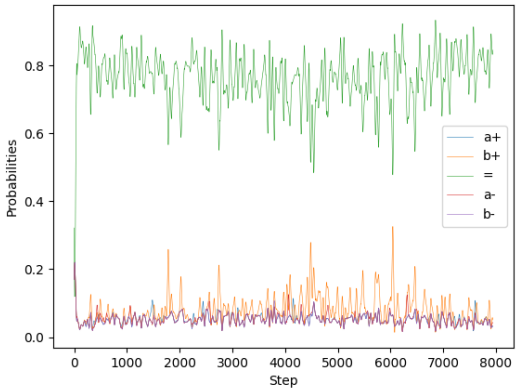
(a) $d = 0.5$, step $= 1$, $1 < w < 20$



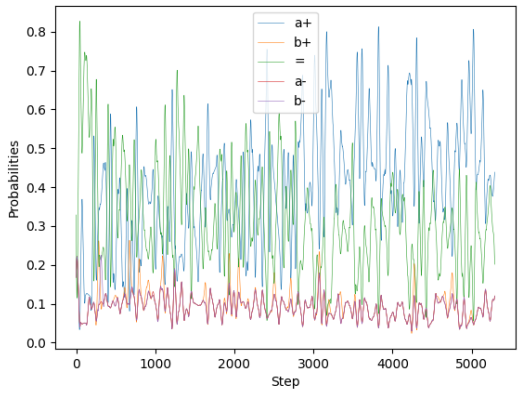
(b) $d = 0.9$, step $= 1$, $1 < w < 20$



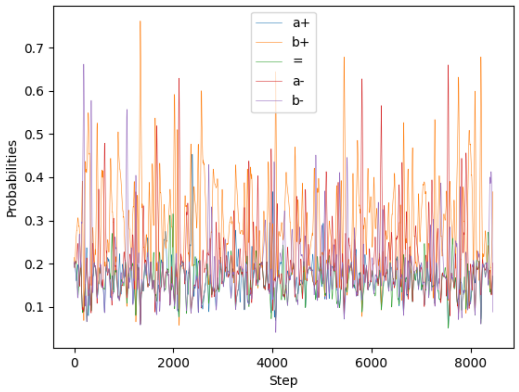
(c) $d = 0.9$, step $= 50\%$, $1 < w < 50$



(d) $d = 0.9$, step $= 10\%$, $1 < w < 50$



(e) $d = 0.9$, step $= 10\%$, $0.01 < w < 100$



(f) $d = 0.9$, step $= 50\%$, $0.01 < w < 50$, reduced state

Figure 5.2: Variation of probabilities during training for different models

6 Evaluation

With various aspects of this project requiring the development of novel designs, the suitability of these design choices has been evaluated. Analyse of the results has revealed the strengths and weaknesses of the adaptive weighting model, along with proposed solutions to some of the key issues present. A brief discussion on the architectural design of the model itself has been included.

6.1 Performance Evaluation

Adaptive weighting has managed to exceed the performance of traditional DAGAN when non-optimal initial weights are specified. Whilst DAGAN was then limited by its capacity to learn based on these weights, AW-GAN was able to adjust the weights to more useful values through reinforcement learning. The best performing model was found to be using the reduced state (using only gradients) with 5 episodes, $k = 30$, $d = 0.9$, $s = 10\%$, and $w = [0.01, 50]$. This surpassed the equivalently trained DAGAN on all metrics by 0.0342 on NMSE, 0.0187 on the SSIM and 1.8311 on PSNR. Performance exceeded that of DAGAN trained with the full dataset and optimal weights, with 0.0192 improvement on NMSE and 0.3479 on PSNR. The majority of the percentage-based step size models found improved performance compared to DAGAN, with more improvement found through parameter optimisation.

Initial trials had failed to beat the original DAGAN's performance; their defining quality was the 50% learning rate decay that had not been changed. An explanation for this reduced performance is given in section 6.2. It should be noted that this performance was only evaluated with the reduced dataset and it is unknown whether this improvement would carry over to a model trained with significantly more data. Despite this, there was clear evidence that adaptive weighting could provide performance benefits to neural network training with multi-part loss functions. If the impact of each loss component was not well understood, this approach would allow for the significance of each component to be learnt and fed back to network training.

With the current research in GANs, the optimal loss functions have been chosen by intu-

ition combined with experimentation, with increases in performance difficult to explain. Adaptive weighting could provide feedback to the generator on which losses should provide more information at different points in the training, thus optimising loss components. A more responsive adaptation system with greater action space could allow for significantly more losses to be considered, which could make this a general purpose add-on for GAN training.

6.2 Weightings and Probabilities

The weights suggested by the network do not correlate with those chosen by DAGAN's authors, with the majority of cases leading to β being greater than α. Prioritisation of image or spectral loss was dependent on the current training state rather than an optimal value existing. There was also no clear correlation between the two weightings; in figs. 5.1a, 5.1b, 5.1e and 5.1f, the weights varied almost independently of each other whilst the other two configurations showed both weights following similar behaviour throughout the training process. The results would fluctuate heavily from the random selection of actions and so multiple repeats or a more robust training algorithm would be required to verify if optimal weights were being chosen. However, given the improved performance, there was reasonable evidence to suggest that adaptive weightings provided greater benefit than hard-coded weightings used only for scaling losses.

Typically, neural networks would incorporate a learning rate decay which would reduce the step size over time to allow the network to fine-tune its performance. It was assumed that the reinforcement model would learn similar behaviour here, opting to reduce the weightings used for both components to lower the overall loss, mirroring the effect of the decaying learning rate. This was only present in fig. 5.1c and so its cause cannot be confirmed. A lack of convergence and overall noisy and random patterns suggested that more thought must be given to the design of this reinforcement model to achieve robust results.

As shown in fig. 5.2, the probability of the different actions experienced strong variation throughout the training process with dissimilar results between configurations. This could be partly explained by the natural improvement of the generator and how it coincided with the reinforcement training. As the DAGAN study noted, the model produced reasonable performance with similar weights, demonstrating some degree of robustness to the weightings used. Since the generator loss would naturally decrease regardless of what

the weights are adjusted, the initial actions chosen by the reinforcement model receive a strong reward and remained more likely to be chosen for subsequent rounds. This created a snowball effect which lead to one action being chosen repeatedly for supposedly improved performance, which will be referred to as the *early reinforcement* problem. Evidence of this can be found in the weights variation; many of the weights in any configuration experienced a strong trend towards either the minimum or maximum, which also tended to be the same as the initial direction. Probability graphs showed that if an action was classified as unlikely at the beginning, it would remain at a similar level for the rest of the process.

A potential solution would be to refrain from starting adaptive weighting until losses had dropped to a consistent value. Losses should then reflect the actions taken rather than just the overall trend of the network. It is possible that the network would be stuck in a local optimum and unable to explore a large enough range of weights to escape this and so the optimal time to begin adapting weights must be explored. Alternatively, the number of episodes used per training could be increased to allow for more actions (and their rewards) considered when updating parameters. Initial trials varying this had not accounted for the need to reduce d and so the optimal value could not be concluded from these results.

6.3 State Considerations

The state size k and number of episodes needed to be balanced to provide useful training information. Feedback on implementing a particular weighting should be present in the state, which presented a lower bound to the state. On the other hand, too long of a state would be inefficient as the information would be reflecting the general trend of the training losses rather than the impact of the current weighting. Furthermore, the frequency of updates correlated with k and so a balance had to be found. A later consideration led to the removal of any absolute values from the state and only kept gradients. Given the large variation in magnitude of absolute values both within a training and between different configurations, the usefulness of passing this data through to a neural network is extremely limited. As the final configuration demonstrated, removal of this data did not impede the model's ability to learn and instead improved performance as there was significantly less data to consider. More inspiration could be taken from forecasting networks when considering the state used, such as the inclusion of a second order differential term.

For skewed data such as the loss components, with a small but significant number of extreme outliers, logarithms have been noted to be an appropriate form of scaling. However, the effect this would have on a model's capability to learn was unknown and so the choice of scaling may have negatively impacted performance and presented an intrinsic limitation to the system. If the late start suggestion from section 6.2 was incorporated, loss components would have a much smaller range with which standardisation could work.

The number of episodes used for training had a similar effect; more episodes allowed for more actions and rewards to be considered during updates. However this would also lead to reduced frequency of updates to the action probabilities which might be more suitable for converging to a singular optimal weight rather than active adaption throughout the training process. An episode size of 10 failed to provide enough opportunities for probability updates and led to decreased performance so 5 was used for later trials. Considering that there were also 5 actions available, it was unknown why such a small number of episodes would lead to improved performance; the model would only have 5 decisions to evaluate of which each decision had 5 possibilities. More experimentation is needed with the optimal state, decay and step size identified.

6.4 Action Space

From table 5.1, the optimal step size was 10% of the current value, with a smaller range typically leading to better performance. The percentage-based step size allowed for efficient exploration of weight values regardless of scale whilst the smaller range prevented the model from overextending the search for optimal values. Even with the largest range of $[0.01, 100]$, the model still performed better than DAGAN; despite the weights converging to the upper and lower bounds fig. 5.1e, the probability graph fig. 5.2e showed a slight trend for alpha's increment possibility increasing over time and the equalising probability decreasing, suggesting some learned behaviour over time rather than just initial reinforcement.

By reducing the learning rate decay, performance increased significantly across the majority of trials. As the generator learns by some learning rate multiplied by the multi-part loss, if the learning rate is decreased severely, none of the changes to individual loss components will have significant effect on overall performance. In theory, a learning rate decay should not be necessary as the network should be able to learn to automatically reduce weightings, however this could prove too complex of a pattern to learn in addition to the

existing adaption. Instead, work should be directed towards incorporating the perceptual loss weighting into the network, allowing for greater adaptive capability. One important issue would be what to use as the basis for the reward function, as there must not be a direct link between the overall loss and the reward function; the model could learn to reduce all weightings to zero in this situation for improved performance.

Currently, the action space is limited in terms of options. A continuous action space could be implemented to allow for greater flexibility in weight adjustment, along with capacity to handle more weights. Whilst other studies have shown large performance increases with q-learning, no other work has been performed on how to adapt VPG to this task. The designed state and actions are novel and can act as a starting point for research in this direction, with optimisation available both in terms of the presentation of the problem to the network, and the hyperparameters of the network itself.

6.5 Review of Design

With a relatively small input size, the AW net was kept simple, consisting of only 1-2 hidden layers depending on the experiment variation. Whilst this prevented the model from overfitting to states and actions, it would also have limited its capability to learn complex patterns. Furthermore, the AW net itself contained many hyperparameters which would require optimisation to maximise performance, which would create another issue similar to the one it aimed to solve. Additional research must be conducted into the robustness of this network to different hyperparameters, or whether optimal values exist which would be applicable for multiple applications. The use of more advanced policy search algorithms would prove useful here, given their reported robustness to hyperparameter settings.

6.6 Referring to Objectives

This project had three main objectives, listed in section 1.2. The relationship between image, spectral and perceptual losses has been quantified and justification for each loss providing information that the others do not has been given. Although there was not standardised method for achieving this object, the use of correlation coefficients should provide a quantitative measure of information gain/loss between different images. Where

one image has an improved pixel loss but increased spectral loss should indicate an imbalance of information within the two domains.

A deep reinforcement learning model has been created, with the design process detailed to understand the author's thought process behind it. A novel state and action space was designed for this purpose, given that existing studies tested a limited range of discrete value combinations. AW-GAN's approach has provided a foundation for future work within this area which should reduce the need for hyperparameter optimisation further.

This model has been experimented with and the performance effects have been evaluated in detail, with comments on the effectiveness of certain design choices. More in-depth testing would have been preferred if given more time to identify the limiting factors in this design and to determine if performance was improved on reduced undersampling ratios. Whilst a quantitative analysis has been performed using the given metrics, a qualitative analysis should be used for comparison of the actual MR images which have been produced. Given the significant noise still present in the reconstructed image at high undersampling ratios, the author has been unable to determine if significant improvement was present.

7 Conclusion

A brief summary of the work achieved is provided, from reviewing the motivations and research in this field to the actual implementation and performance of the design. Key areas for future work are summarised from the previous section before a brief discussion on the ethical and legal implications of this project.

7.1 Summary of Work

The need for faster MRI has been established, along with the traditional methods being used today. An understanding of CS and its place in fast MRI research has been explored, with strengths and weaknesses explained. Applications of deep learning in this area have been noted, along with novel areas of research that showed great potential. A technical background along with a brief literature review on all relevant areas has been provided, which should be sufficient to provide a foundation of knowledge for any new research in this area.

Analysis of the results from DAGAN have provided justification for a more in-depth examination of the loss components used in GANs. From this, the AW-GAN was developed with a novel state and action space design for application of RL to adaptive weighting for loss components. Performance improvements have been shown to be possible using VPG training, with improvements of 0.0342 on NMSE, 0.0187 on the SSIM, and 1.8311 dB on PSNR compared to the equivalently trained DAGAN.

Through experimentation with various configurations, the importance of different factors in the design of this network has been noted. Usefulness of various states has been shown along with attempts to improve the action space. The instability in the training process has been identified and associated with the early reinforcement problems discussed before. All issues which were identified have suggested explanations along with potential fixes to be applied in future work (discussed within chapter 6 but summarised in section 7.2).

The principles behind this study have established a new foundation of research within the fast MRI community. With adaptive weighting, multi-part loss functions can be given greater complexity with minimal hyperparameter tuning required. Reconstruction

accuracy of severely undersampled MRI images has been improved, allowing for even greater scanning speeds and thus greater adoption of the imaging modality in other uses. Usage of CT scans could be reduced if the adaptive weighting technique can be verified thoroughly in a larger experiment range.

7.2 Future Considerations

Multiple areas for improvement were identified along with potential solutions in chapter 6. For ease of reading, these have been summarised below along with additional comments.

One of the key problems identified with this design was the early reinforcement problem (section 6.2). The proposed solution was delayed activation of AW-net training to reduce the influence of the strong trend during initial training stages. Alternative solutions could be to identify the relationship of the general reduction in training losses over time and remove this when calculating the reward function; this would isolate the effects of adaptive weighting within the rewards and allow for an improved training environment. Either method would require careful tuning of hyperparameters to ensure that the model continued to gain useful information for training. In the former method, it might prove difficult for adaptive weighting to identify the necessary moves to shift the model out of a local optimum, whilst the latter could also remove the effects of adaptive weighting itself.

Later experiments with state components revealed that initial experiments contained state information which had worsened performance. The model had been unable to learn useful features from this data and ended up producing worse results. Considering this, further research must be performed to identify additional information which could provide relevant information for training. The author has suggested the inclusion of the second order differential of each component, taking inspiration from forecasting networks. Additional components such as skew and kurtosis may also prove useful to provide more information on the trend of the data, along with relative comparisons of image and spectral losses which may indicate the need for prioritisation of a particular component. The logarithmic scaling may not be the optimal solution to the varying magnitudes problem. Additional research may exist which can provide a more informative but scaled dataset.

The use of VPG was to identify which issues that the algorithm typically experienced were

still applicable to this situation. Whilst it has been shown that Q-learning can achieve hyperparameter optimisation, there have been no studies on using policy methods. Whilst this study has laid the foundation for application of policy methods to this area, further work would be needed to increase the robustness of the results. Experience replay could be tested as without the absolute values in state, there could be more similarities in the states and thus repeated learning would provide benefits. Other potential areas of improvement could be found in the use of trust regions and policy optimisation, as well as incorporating continuous action spaces.

7.3 Legal, Social, Ethical and Professional Considerations

The code repository of DAGAN is available for public use [7] and has been appropriately referenced throughout this study. The data used from the MICCAI 2013 grand challenge dataset is also publicly available and contained no identifying information. No additional data collection was required so there are no concerns regarding personal information collection and storage.

Whilst this study was aimed at improving the performance of GANs for CS-MRI, the principles explored can apply to GANs in general. With the increasing concern around deepfake technology, an improvement in the performance of GANs can lead to more realistic deepfakes being produced [79]. Their potential for the creation and proliferation of false information and reputation disruption is a growing concern in the modern world. Whilst creating GANs for deepfakes requires significant technical understanding, the development of simple applications for other users to use is not unlikely. It could be argued that this work makes the development of GANs even simpler, removing the need to manually tune weights for losses. However, at the current state of the design, improvements have not been verified with larger datasets and there are clear issues to becoming a general solution for GANs.

Bibliography

[1] Kieren Grant Hollingsworth. Reducing acquisition time in clinical mri by data undersampling and compressed sensing reconstruction. *Physics in Medicine and Biology*, 60:R297–R322, 11 2015. ISSN 0031-9155. doi: 10.1088/0031-9155/60/21/R297. URL `https://iopscience.iop.org/article/10.1088/0031-9155/60/21/R297`.

[2] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52: 1289–1306, 4 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.871582. URL `http://ieeexplore.ieee.org/document/1614066/`.

[3] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58: 1182–95, 12 2007. ISSN 0740-3194. doi: 10.1002/mrm.21391. URL `http://www.ncbi.nlm.nih.gov/pubmed/17969013`.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 6 2014. URL `http://arxiv.org/abs/1406.2661`. Contains proof that the minmax value function leads to the generator getting the data distribution.

[5] Guang Yang, Jun Lv, Yutong Chen, Jiahao Huang, and Jin Zhu. Generative adversarial networks (gan) powered fast magnetic resonance imaging – mini review, comparison and perspectives, 5 2021. URL `http://arxiv.org/abs/2105.01800`.

[6] Alaa Abu-Srhan, Mohammad A.M. Abushariah, and Omar S. Al-Kadi. The effect of loss function on conditional generative adversarial networks. *Journal of King Saud University - Computer and Information Sciences*, 3 2022. ISSN 13191578. doi: 10.1016/j.jksuci.2022.02.018. URL `https://linkinghub.elsevier.com/retrieve/pii/S1319157822000519`.

[7] Guang Yang, Simiao Yu, Hao Dong, Greg Slabaugh, Pier Luigi Dragotti, Xujiong Ye, Fangde Liu, Simon Arridge, Jennifer Keegan, Yike Guo, and David Firmin. Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE Transactions on Medical Imaging*, 37:1310–1321, 6 2018. ISSN

0278-0062. doi: 10.1109/TMI.2017.2785879. URL https://ieeexplore.ieee.org/document/8233175/.

[8] Hadi S. Jomaa, Josif Grabocka, and Lars Schmidt-Thieme. Hyp-rl : Hyperparameter optimization by reinforcement learning, 6 2019. URL http://arxiv.org/abs/1906.11527.

[9] Fatma M. Talaat and Samah A. Gamel. Rl based hyper-parameters optimization algorithm (roa) for convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, 1:3, 3 2022. ISSN 1868-5137. doi: 10.1007/s12652-022-03788-y. URL https://link.springer.com/10.1007/s12652-022-03788-y.

[10] Donald W. McRobbie, Elizabeth A. Moore, Martin J. Graves, and Martin R. Prince. *MRI from Picture to Proton*. Cambridge University Press, 9 2006. ISBN 9780521683845. doi: 10.1017/CBO9780511545405. URL https://www.cambridge.org/core/product/identifier/9780511545405/type/book.

[11] Allen D. Elster. Radial sampling, 5 2022. URL https://mriquestions.com/radial-sampling.html.

[12] Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1:100004, 4 2021. ISSN 26670968. doi: 10.1016/j.jjimei.2020.100004. URL https://linkinghub.elsevier.com/retrieve/pii/S2667096820300045.

[13] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine*, 34:26–38, 8 2017. ISSN 1053-5888. doi: 10.1109/MSP.2017.2743240. URL http://arxiv.org/abs/1708.05866http://dx.doi.org/10.1109/MSP.2017.2743240.

[14] D Poole and A Mackworth. Artificial intelligence - foundations of computational agents – 9.5 decision processes, 2010. URL https://artint.info/html/ArtInt_224.html.

[15] Jan Peters. Policy gradient methods. *Scholarpedia*, 5:3698, 2010. ISSN 1941-6016. doi: 10.4249/scholarpedia.3698. URL http://www.scholarpedia.org/article/Policy_gradient_methods.

[16] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 5 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL http://link.springer.com/10.1007/BF00992696.

[17] Oren N Jaspan, Roman Fleysher, and Michael L Lipton. Compressed sensing mri: a review of the clinical literature. *The British Journal of Radiology*, 88: 20150487, 12 2015. ISSN 0007-1285. doi: 10.1259/bjr.20150487. URL http://www.birpublications.org/doi/10.1259/bjr.20150487. PMID: 26402216.

[18] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, 25:72–82, 3 2008. ISSN 1053-5888. doi: 10.1109/MSP.2007.914728. URL http://ieeexplore.ieee.org/document/4472246/.

[19] J Fletcher, M D Clark, F A Sutton, R Wellings, and K Garas. The cost of mri: changes in costs 1989-1996. *The British Journal of Radiology*, 72:432–437, 5 1999. ISSN 0007-1285. doi: 10.1259/bjr.72.857.10505004. URL http://www.birpublications.org/doi/10.1259/bjr.72.857.10505004.

[20] Shreyas S. Vasanawala, Marcus T. Alley, Brian A. Hargreaves, Richard A. Barth, John M. Pauly, and Michael Lustig. Improved pediatric mr imaging with compressed sensing. *Radiology*, 256:607–616, 8 2010. ISSN 0033-8419. doi: 10.1148/radiol.10091218. URL http://pubs.rsna.org/doi/10.1148/radiol.10091218.

[21] Albert Hsiao, Michael Lustig, Marcus T. Alley, Mark Murphy, Frandics P. Chan, Robert J. Herfkens, and Shreyas S. Vasanawala. Rapid pediatric cardiac assessment of flow and ventricular volume with compressed sensing parallel imaging volumetric cine phase-contrast mri. *American Journal of Roentgenology*, 198:W250–W259, 3 2012. ISSN 0361-803X. doi: 10.2214/AJR.11.6969. URL https://www.ajronline.org/doi/10.2214/AJR.11.6969.

[22] P Mansfield. Multi-planar image formation using nmr spin echoes. *Journal of Physics C: Solid State Physics*, 10:L55–L58, 2 1977. ISSN 0022-3719. doi: 10.1088/0022-3719/10/3/004. URL https://iopscience.iop.org/article/10.1088/0022-3719/10/3/004.

[23] J. Hennig, A. Nauerth, and H. Friedburg. Rare imaging: A fast imaging method for clinical mr. *Magnetic Resonance in Medicine*, 3:823–833, 12 1986. ISSN 07403194. doi: 10.1002/mrm.1910030602. URL https://onlinelibrary.wiley.com/doi/10.1002/mrm.1910030602.

[24] D A Feinberg, J D Hale, J C Watts, L Kaufman, and A Mark. Halving mr imaging time by conjugation: demonstration at 3.5 kg. *Radiology*, 161:527–531, 11 1986. ISSN 0033-8419. doi: 10.1148/radiology.161.2.3763926. URL http://pubs.rsna.org/doi/10.1148/radiology.161.2.3763926.

[25] P. B. Roemer, W. A. Edelstein, C. E. Hayes, S. P. Souza, and O. M. Mueller. The nmr phased array. *Magnetic Resonance in Medicine*, 16:192–225, 11 1990. ISSN 07403194. doi: 10.1002/mrm.1910160203. URL https://onlinelibrary.wiley.com/doi/10.1002/mrm.1910160203.

[26] Emmanuel Candes, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, 9 2004. ISSN 00189448. doi: 10.48550/arxiv.math/0409186. URL http://arxiv.org/abs/math/0409186.

[27] Emmanuel J Candes and Justin K Romberg. Signal recovery from random projections. *Computational Imaging III*, 5674:76 – 86, 2005. doi: 10.1117/12.600722. URL https://doi.org/10.1117/12.600722.

[28] Ivana Tosic and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28:27–38, 3 2011. ISSN 1053-5888. doi: 10.1109/MSP.2010.939537. URL http://ieeexplore.ieee.org/document/5714407/.

[29] Jose Caballero, Anthony N. Price, Daniel Rueckert, and Joseph V. Hajnal. Dictionary learning and time sparsity for dynamic mr data reconstruction. *IEEE Transactions on Medical Imaging*, 33:979–994, 4 2014. ISSN 0278-0062. doi: 10.1109/TMI.2014.2301271. URL http://ieeexplore.ieee.org/document/6716020/.

[30] J.M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18:1395–1408, 7 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2022459. URL http://ieeexplore.ieee.org/document/5061489/.

[31] Andreas Greiser and Markus von Kienlin. Efficientk-space sampling by density-weighted phase-encoding. *Magnetic Resonance in Medicine*, 50:1266–1275, 12 2003. ISSN 0740-3194. doi: 10.1002/mrm.10647. URL https://onlinelibrary.wiley.com/doi/10.1002/mrm.10647.

[32] Chi-Ming Tsai and Dwight G. Nishimura. Reduced aliasing artifacts using variable-densityk-space sampling trajectories. *Magnetic Resonance in Medicine*, 43:452–458, 3 2000. ISSN 0740-3194. doi: 10.1002/(SICI)1522-2594(200003)43:3<452::AID-MRM18>3.0.CO;2-B. URL https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1522-2594(200003)43:3<452::AID-MRM18>3.0.CO;2-B.

[33] Wotao Yin, Stanley Osher, Donald Goldfarb, and Jerome Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1:143–168, 1 2008. ISSN 1936-4954. doi: 10.1137/070703983. URL http://epubs.siam.org/doi/10.1137/070703983.

[34] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1457, 11 2004. ISSN 0010-3640. doi: 10.1002/cpa.20042. URL https://onlinelibrary.wiley.com/doi/10.1002/cpa.20042.

[35] Jo Schlemper, Jose Caballero, Joseph V. Hajnal, Anthony N. Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE Transactions on Medical Imaging*, 37:491–503, 2 2018. ISSN 0278-0062. doi: 10.1109/TMI.2017.2760978. URL https://ieeexplore.ieee.org/document/8067520/.

[36] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19:221–248, 6 2017. ISSN 1523-9829. doi: 10.1146/annurev-bioeng-071516-044442.

[37] Avinash Hindupur. The gan zoo, 9 2018. URL https://github.com/hindupuravinash/the-gan-zoo.

[38] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35:53–65, 10 2017. ISSN 10535888. doi: 10.1109/MSP.2017.2765202. URL http://arxiv.org/abs/1710.07035http://dx.doi.org/10.1109/MSP.2017.2765202.

[39] Shahab Jozdani, Dongmei Chen, Darren Pouliot, and Brian Alan Johnson. A review and meta-analysis of generative adversarial networks and their applications in remote sensing. *International Journal of Applied Earth Observation and Geoinformation*,

108:102734, 4 2022. ISSN 15698432. doi: 10.1016/j.jag.2022.102734. URL https://linkinghub.elsevier.com/retrieve/pii/S0303243422000605.

[40] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 11 2014. URL http://arxiv.org/abs/1411.1784.

[41] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 11 2016. URL http://arxiv.org/abs/1611.07004.

[42] Augustus Odena. Semi-supervised learning with generative adversarial networks, 6 2016. URL http://arxiv.org/abs/1606.01583.

[43] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans, 10 2016. URL http://arxiv.org/abs/1610.09585.

[44] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 5 2016. doi: 10.48550/arxiv.1605.09782. URL http://arxiv.org/abs/1605.09782.

[45] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 1 2017. doi: 10.48550/arxiv.1701.04862. URL http://arxiv.org/abs/1701.04862.

[46] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 2017-December:5768–5778, 3 2017. ISSN 10495258. doi: 10.48550/arxiv.1704.00028. URL http://arxiv.org/abs/1704.00028.

[47] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *34th International Conference on Machine Learning, ICML 2017*, 1:322–349, 3 2017. doi: 10.48550/arxiv.1703.00573. URL http://arxiv.org/abs/1703.00573.

[48] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Con-*

ference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 11 2015. doi: 10.48550/arxiv.1511.06434. URL http://arxiv.org/abs/1511.06434.

[49] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 11 2014. ISSN 01628828. doi: 10.48550/arxiv.1411.4038. URL http://arxiv.org/abs/1411.4038.

[50] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, 12 2014. doi: 10.48550/arxiv.1412.6806. URL http://arxiv.org/abs/1412.6806.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 5 2015. URL http://arxiv.org/abs/1505.04597.

[52] Tran Minh Quan, Thanh Nguyen-Duc, and Won-Ki Jeong. Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss. *IEEE Transactions on Medical Imaging*, 37:1488–1497, 6 2018. ISSN 0278-0062. doi: 10.1109/TMI.2018.2820120. URL https://ieeexplore.ieee.org/document/8327637/.

[53] José Salvador, João Oliveira, and Mauricio Breternitz. Reinforcement learning: A literature review (september 2020), 10 2020.

[54] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 5 1992. ISSN 0885-6125. doi: 10.1007/BF00992698. URL http://link.springer.com/10.1007/BF00992698.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–33, 2 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL http://www.ncbi.nlm.nih.gov/pubmed/25719670.

[56] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*,

pages 2094–2100, 9 2015. ISSN 2159-5399. doi: 10.48550/arxiv.1509.06461. URL http://arxiv.org/abs/1509.06461.

[57] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. *33rd International Conference on Machine Learning, ICML 2016*, 4:2939–2947, 11 2015. doi: 10.48550/arxiv.1511.06581. URL http://arxiv.org/abs/1511.06581.

[58] Jonathan Connell and Sridhar Mahadevan. Robot learning. *Robotica*, 17:229–235, 3 1999. ISSN 0263-5747. doi: 10.1017/S0263574799271172.

[59] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2 2015. URL http://arxiv.org/abs/1502.05477.

[60] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 7 2017. URL http://arxiv.org/abs/1707.06347.

[61] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms, 1999.

[62] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2 2016. URL http://arxiv.org/abs/1602.01783.

[63] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2 2018.

[64] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 12 2018.

[65] Douglas C Montgomery and John Wiley. Design and analysis of experiments eighth edition, 2013. URL www.wiley.com/go/permissions.

[66] Matthias Feurer and Frank Hutter. Hyperparameter optimization, 2019. URL http://link.springer.com/10.1007/978-3-030-05318-5_1.

[67] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. URL http://scikit-learn.sourceforge.net.

[68] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6683 LNCS:507–523, 2011. ISSN 03029743. doi: 10.1007/978-3-642-25566-3_40/COVER. URL https://link.springer.com/chapter/10.1007/978-3-642-25566-3_40.

[69] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104:148–175, 1 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2494218. URL https://ieeexplore.ieee.org/document/7352306/.

[70] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, 5 2013. ISSN 15206149. doi: 10.1109/ICASSP.2013.6639346. URL http://ieeexplore.ieee.org/document/6639346/.

[71] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw bayesian optimization, 6 2014. URL http://arxiv.org/abs/1406.3896.

[72] Xueqiang Zeng and Gang Luo. Progressive sampling-based bayesian optimization for efficient and automatic machine learning model selection. *Health Information Science and Systems*, 5:1–21, 12 2017. ISSN 20472501. doi: 10.1007/S13755-017-0023-Z/TABLES/5. URL https://link.springer.com/article/10.1007/s13755-017-0023-z.

[73] Russell C. Eberhart and Yuhui Shi. Comparison between genetic algorithms and particle swarm optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1447: 611–616, 1998. ISSN 16113349. doi: 10.1007/BFB0040812/COVER. URL https://link.springer.com/chapter/10.1007/BFb0040812.

[74] Nikolaus Hansen. The cma evolution strategy: A tutorial, 4 2016. URL http://arxiv.org/abs/1604.00772.

[75] Ilya Loshchilov and Frank Hutter. Cma-es for hyperparameter optimization of deep neural networks, 4 2016. URL http://arxiv.org/abs/1604.07269.

[76] A. Ali Heydari, Craig A. Thompson, and Asif Mehmood. Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions, 12 2019. URL http://arxiv.org/abs/1912.12355.

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014. doi: 10.48550/arxiv.1409.1556. URL http://arxiv.org/abs/1409.1556.

[78] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 6 2016. URL http://arxiv.org/abs/1606.01540.

[79] Johannes Langguth, Konstantin Pogorelov, Stefan Brenner, Petra Filkuková, and Daniel Thilo Schroeder. Don't trust your eyes: Image manipulation in the age of deepfakes. *Frontiers in Communication*, 6, 5 2021. ISSN 2297-900X. doi: 10.3389/fcomm.2021.632317. URL https://www.frontiersin.org/articles/10.3389/fcomm.2021.632317/full.