Generative Models for Inverse Imaging Problems

Duff, Margaret

Award date:
2023

Awarding institution:
University of Bath

Generative Models for Inverse Imaging Problems

Margaret Anne Georgina Duff

Thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Mathematical Sciences

December 2022

COPYRIGHT

DECLARATION OF ANY PREVIOUS SUBMISSION OF THE WORK

The material presented here for examination for the award of a higher degree by research has not been incorporated into a submission for another degree.

Candidate's signature .....

Margaret Anne Georgina Duff

DECLARATION OF AUTHORSHIP

I am the author of this thesis, and the work described therein was carried out by myself personally, without exception.

Candidate's signature .....

Margaret Anne Georgina Duff

Summary

This thesis sits at the intersection of deep learning and inverse imaging problems and looks at the use of generative models in the framework of variational regularisation.

The first chapter, the introduction, motivates why inverse problems and specifically deep learning approaches to inverse problems are exciting and relevant before setting out key research questions that this thesis hopes to contribute to. A more detailed summary of the work presented in this thesis sets out a narrative that runs through the research. The chapter is concluded with a discussion of how the research will be evaluated.

The background of this thesis is set out in chapters 2 and 3. Chapter 2 starts with a brief introduction to inverse problems, the variational regularisation approach and the optimisation methods used in this work. The chapter then moves on to introduce deep learning and provides a critical analysis of deep learning applied to inverse problems. This sets the scene for the learned regularisation methods presented in this thesis. Chapter 3 looks at generative models, deep learning approaches for producing images similar to some training set. The chapter provides an introduction to a range of state-of-the-art models, including autoencoders, variational autoencoders and generative adversarial networks, with a unified notation and approach. The derivations of the variational autoencoder, in particular, will be useful for later chapters.

Chapter 4 starts with an introduction to the main theme of this thesis: *generative regularisers*. Generative regularisers penalise solutions to the inverse problem that are far from the range of a trained generative model. A literature review highlights and unifies key emerging themes from the literature. The success of generative regularisers depends on the quality of the generative model and one of the main contributions of this chapter is a set of desired criteria for generators that will go on to be used in an inverse problem context. These criteria are not sufficient for success, but they are a useful starting point that could guide future generative model research. The chapter finishes with a range of numerical experiments to test and compare generative models and generative regularisation

methods.

The direction of the thesis now forks into two possible extensions of the generative regularisers set out in chapter 4. Chapter 5 looks closer at how generative regularisers measure the distance from the range of a generative model and uses an adapted variational autoencoder to learn this distance in a way that is adaptive across an image and throughout the set of images. Numerical experiments demonstrate the flexibility of this approach and compare it to other learned and unlearned reconstruction methods. Chapter 6 considers the training of the generative model and how to train generators, specifically variational autoencoders, without access to ground truth or high-quality reconstructed data. The learned generators are tested for their ability to regularise inverse imaging problems.

Finally, chapter 7 gives a summary of the contributions of this thesis, linking to the key areas highlighted in the introduction, and outlines directions for future work following on from this thesis.

Acknowledgements

When I started this PhD I got a wide variety of advice and wisdom, two memorable pieces of advice that I would pass on included that "If you don't regularly feel stupid during your PhD, then you aren't doing it right!" and that a PhD is like a roller coaster ride. Both gave comfort and motivation during the inevitable twists and turns of research. Indeed, when starting in 2018, we couldn't have predicted the madness of 2020 and the dramatic changes that came with this. However, although many times the thesis felt like an insurmountable obstacle and something only other people did, here it is!

The first people to thank are of course my supervisors, Matthias Ehrhardt and Neill Campbell, who provided great guidance, encouragement and support. They provided great scientific insight and expertise but also let me forge my own path and set the direction for my own research. They consistently encouraged me to travel, present and talk to new people and develop new ideas. I must also thank Stephen Cowley and Carola-Bibiane Shonlieb, from Cambridge, without whom I would not have thought a PhD would be possible. I was fortunate enough to be able to attend conferences and interact with countless brilliant mathematicians and computer scientists. Discussing my work with others is always a great opportunity. I especially thank Vinay Namboodiri and Andreas Hauptmann for examining this thesis, giving up their time to read and discuss my thesis and provide valuable advice and feedback. I also thank the Bath Imaging Group: Claire, Jarrod, Baruch, Seb, Sadegh and Pablo who provided interesting discussion, a place to talk about my PhD and who all helped proof read abstracts and papers throughout the years.

The maths department at Bath University has universally been friendly and supportive, something I am continually surprised by and greatly appreciate! I want to thank the community of PhD students who were always available for distractions and support and especially for Wednesday afternoon cake and Friday morning coffee. I thank all the inhabitants of 4 West level 1, for putting up with me and my messy desk, but also for the companionship and friendship. I owe a lot to Susie, Lou, Lindsay, Helena and Jess in the SAMBa office who were amazing problem solvers and worked tirelessly on behalf of the PhD students. A special

mention goes to Miriam who kept me organised and provided consistent support and a listening ear.

Personally, I must thank my Mum, who was always available to listen, and learned to nod along every time I discussed maths! I spent long periods working from home, during lockdowns, and am always grateful for the company, home cooked meals and walks in the woods. I thank my Nan, who never really understood what I was doing, but always checked I was eating enough, and once told me that printing an academic poster was cheating and I should draw it instead! I must thank my friends including Danielle, who was always at the end of the phone; Sam, who always provides good advice and helped proof read parts of the thesis; Eloise, who got dragged on some very long walks round Bath and always has a cake recipe to share; and many many more! I was kept grounded and busy during my PhD by 10th Bath Combe Down Cubs and 39th Bath Guides, HCPT groups 37 and 104 and Girlguiding's inclusion network and GOLD team. I thank all volunteers, staff and parents at these charities.

Contents

List of Figures

Notation

This section does not aim to comprehensively list all notation used in this thesis but just key pieces of notation than runs through the work.

- Let $\mathcal{X}$ be the *image space* with *target* probability density distribution $p^*_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}_{\geq 0}$. Usually take $\mathcal{X} = \mathbb{R}^d$ a d-dimensional vector space where vectors in $\mathcal{X}$ can be arranged in a 2D grid of dimensions $d_1 \times d_2$, where $d_1 d_2 = d$.
- The *measurement or observation* space $\mathcal{Y}$ is usually taken to be $\mathcal{Y} = \mathbb{R}^m$.
- Define ϵ to be the additive noise in an inverse problem. The exact vector is unknown but we usually assume the noise model is known or can be approximated.
- Define $\mathcal{D} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq} 0$ to be a similarity measure on the measurement space, which we will assume smooth in its first argument and non-negative.
- Define $\mathcal{L} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq} 0$ to be a similarity measure on the image space, which we will assume smooth in its first argument and non-negative.
- Generally take $P_{\mathcal{A}}$ to be a probability distribution on the set $\mathcal{A}$ and $p_{\mathcal{A}}$ to be a corresponding density.
- Define $\mathcal{R} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ to be a regularisation function.
- Let $\lambda, \mu \in \mathbb{R}_{\geq 0}$ be regularisation parameters.
- Let $\mathcal{Z}$ be the *latent space* with probability density function $p_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}_{\geq 0}$. Usually take $\mathcal{Z} = \mathbb{R}^n$.
- Take $G : \mathcal{Z} \to \mathcal{X}$ to be a *generator*, usually parameterised by θ.
- Let P_G be the distribution in the image space induced by the generator.
- Let $E : \mathcal{X} \to \mathcal{Z}$ be an *encoder*, usually parameterised by ψ.
- Let $D : \mathcal{X} \to \mathbb{R}$ be a *discriminator*, usually parameterised by ϕ.
- Let $\Sigma : \mathcal{Z} \to \mathbb{R}^{d \times d}$, and $\Sigma(z)$ be the covariance associated with the generated image $G(z)$. Note that Σ is usually parameterised by θ. Often

$\Sigma \equiv \rho^2 I$, for some $\rho \in \mathbb{R}$ but in chapter 5, Σ is dense, and we define the precision Λ where $\Lambda^{-1} = \Sigma$ and has Cholesky decomposition $LL^T = \Lambda$.

- Define $\mathcal{N}(\cdot; \mu, \Sigma) : \mathcal{A} \to \mathbb{R}$ to be the probability density function of a normal distribution with mean μ and covariance Σ^2 over the space $\mathcal{A}$.
- Define $Po(\cdot; \lambda) : \mathcal{A} \to \mathbb{R}$ to be to be the probability density function of a Poisson distribution with mean λ over the space $\mathcal{A}$.
- Define the VAE encoded distribution $\mathcal{N}_{x,\psi}(z) := \mathcal{N}(z; \mu_\psi(x), \text{diag}(\sigma^2_\psi(x)))$, a distribution over the latent space $\mathcal{Z}$.
- Define the VAE generator distribution $p_{G,\Sigma}(\cdot|z; \theta) = \mathcal{N}(G_\theta(z), \Sigma_\theta(z))$.
- Denote $\mathcal{J}$ to be the AE or VAE objective function that we wish to minimise. It is usually a function of θ and ψ.
- Denote $\mathcal{K}$ to be the saddle point objective of a GAN usually a function of θ and ϕ.
- Let $\mathcal{F} : \mathcal{X} \to [0, \infty]$ be a distance used in a generative regulariser to measure the distance from the range of the generator.
- Define $d(\cdot, \cdot)$ to be a distance between two probability distributions.
- The *support* of a function $f : \mathcal{A} \to \mathcal{B}$ is given as

$$supp(f) = \{a \in \mathcal{A} | f(a) \neq 0\}. \tag{1}$$

- The *range* of a function $f : \mathcal{A} \to \mathcal{B}$ is given as:

$$Range(f) = \{b \in \mathcal{B} \mid \exists a \in \mathcal{A} \; s.t. \; f(a) = b\}. \tag{2}$$

- The characteristic function, defined for an arbitrary set $\mathcal{C}$ is defined as:

$$\iota_{\mathcal{C}}(t) = \begin{cases} 0 & \text{for} t \in \mathcal{C} \\ \infty & \text{for} t \notin \mathcal{C} \end{cases}. \tag{3}$$

- The Kullback-Leibler (KL) divergence is defined to be

$$d_{\text{KL}}(p\|q) = \mathbb{E}_{x \sim p(x)} \left[\log \left(\frac{p(x)}{q(x)} \right) \right]. \tag{4}$$

  The Kullback-Leibler divergence is defined only if for all x, $q(x) = 0$ implies $p(x) = 0$.
- Throughout we use the p-norm, for $p \geq 1$ and $x \in \mathcal{X}$, $\|x\|_p := (\sum_i |x_i|^p)^{1/p}$.

- We define the adjoint of the operator A, denoted A^* as

$$(Ax)^T y = x^T A^* y, \forall x \in \mathcal{X}, y \in \mathcal{Y}. \tag{5}$$

- Define $A^\dagger : \mathcal{Y} \to \mathcal{X}$ to be some mapping from the measurement to the image space, usually a 'rough' inverse.
- Define the weighted norm $\|x\|_M^2 = x^T M^{-1} x$.
- Denote the determinant of a matrix M by $|M|$.

Chapter 1

Introduction

1.1 Motivation

Solving an inverse problem is the task of computing an unknown quantity, $x \in \mathcal{X}$, from observed measurements, $y \in \mathcal{Y}$. The unknowns and the observed measurement are related by a forward model, $A : \mathcal{X} \to \mathcal{Y}$. We consider the problem of finding x, given y and A, connected by

$$Ax \approx y. \tag{1.1}$$

The equation is approximate due to potential noise in the measurements.

Inverse problems are ubiquitous in modern science and engineering. In image processing, many of the imaging tools smartphones now implement automatically are inverse problems: deblurring images when our hand wobbles; denoising images when we take them in low light; or providing super-resolution when zooming in further than the camera's physical capability. Taking black and white images and colourising them or filling in missing patches in images are also examples of inverse problems. Inverse problems tell us about parameters that we cannot directly observe. In geophysics, seismic inversion uses seismic measurements to infer physical properties below the earth's surface. Similarly, sonar systems on ships can be used to create images of the ocean floor and RADAR is a detection system that uses the scattering of radio waves to find objects in a region. This thesis will mainly focus on image reconstruction, for example, in computed tomography (CT) 1D line measurements are taken in a ring around the outside of a patient and the challenge is to create a 2D slice of the patient, useful for interpretation by medical professionals. In magnetic resonance imaging (MRI) strong magnets and computer-generated radio waves produce faint signals which are detected outside the body and again can be used to produce 2D slice images. In 2019, we saw the first image of a black hole [8] reconstructed from observations

from the event horizon telescope conducted over 4 nights using eight stations in six geographic sites across the world.

Inverse problems are useful and applicable but they are also challenging in a mathematical sense. Interesting inverse problems are often ill-posed, there does not exist a solution, the solution may not be unique or you may see large variations in the solution in response to small changes in measured data. For example in the case of inpainting, filling in missing pixels in an image, there could be many possible solutions. Blurring is a smoothing operator, so deblurring can be very sensitive to changes in the measured data.

Traditional approaches to inverse problems involve incorporating additional information in the form of a regulariser function that encourages solutions to the inverse problem to have certain hand-crafted properties. Examples include total variation regularisation [192, 172] or functions that encourage sparsity in some basis [88, 42, 35]. The latter is closely linked to theories on compressed sensing [37, 60]. An alternative approach uses a Bayesian framework, providing additional information to the inverse problem in the form of a prior [213].

Increasingly available datasets, hardware and software have led to a boom in deep learning in recent years. In deep learning for imaging, there has been fast growth from the first photo-realistic images of non-existent people [121, 122] to Dalle-2, which can create original, realistic images and art from a text description [182]. See figure 1-1 for an example generated image from Dalle-2. Language models can now produce realistic text given short prompts on a vast range of topics [33] and Github co-pilot can turn natural language prompts into coding suggestions in a range of programming languages [236]. Google DeepMind has created AlphaFold that predicts a protein's 3D structure from its amino acid sequence and has created a database for nearly all catalogued proteins known to science [114, 220]. There has also been fast growth in the area of inverse problems; between 2018 and 2021 the number of papers on the topic "deep learning and inverse problems" measured by Web of Science increased by a factor of 4.5 [9]. As we shall see in section 2.4, there are a range of approaches for incorporating deep learning in inverse problem reconstruction, including supervised, unsupervised and unlearned approaches.

The work presented in this thesis sits at the intersection of generative machine learning models and inverse imaging problems. Generative models are trained to output images close to some training distribution and are then incorporated into *generative regularisers*, a learned regularisation approach that penalises images far from the range of the trained generative model.

1.2 Active Research Areas

Despite the fast growth and excitement in the field, there are still problems left to solve in the area of deep learning and inverse problems. This subsection discusses a range of current research areas.

Data and training requirements Deep learning often has large data requirements. For example, one supervised method for low dose CT reconstruction, learned primal dual [3], uses 2168 training images, each 512×512 pixels in size from the AAPM Low Dose CT Grand Challenge [151]. Another unsupervised method [160] for compressed sensing MRI reconstruction uses 14513 slices, each 640×368 pixels from the fastMRI dataset [238]. A deep equilibrium method [76] for supervised learning with some convergence guarantees uses 10000 celebrity faces [144] for a deblurring problem and 2000 fastMRI knee images [238] resized to a window of size 320×320, for MRI reconstruction. AutoMAP, a supervised approach for image reconstruction was trained on 50,000 images from 131 subjects taken from the MGH-USC HCP public database [71].

Data is expensive to collect, for example, MRI takes 15-90 mins per subject [161]. Depending on the method, any changes in the MRI set-up, such as the sampling patterns or coil sensitivities, may require more data to be collected. Once data is collected, then it may need to be pre-processed, curated, centred, anonymised and labelled. This can be time-consuming but may also require the loss of data. For example, the Low Dose Parallel Beam dataset (LoDoPaB-CT) [140] takes images from the LIDC/IDRI dataset [14]. They manually removed scans that were of poor quality or had an unusual image size or patient orientation. In total 1010 individual scans were reduced to 812. For another example, a recent paper on equivariant neural networks [38] took 10000 fastMRI [238] brain DICOM images and sorted out just 6200 for training, test and validation, removing images based on quality.

Future growth in deep learning and inverse problems could consider methods that can learn and generalise from smaller training sets. Large scale benchmarking datasets, such as the Low Dose CT Grand Challenge and the fastMRI dataset, form important testing sets for new methods and allowing benchmarking and comparisons between methods. Sharing data in this way is also important for the future of the field.

Even with sufficient time and budget, there may be cases where ground truth or high-quality reconstructions are not available. For the first image of a black hole [8], there was no previous data to draw on. Designing algorithms with lower data requirements is important when data is difficult or expensive to collect and collate.

Mathematical convergence guarantees For safety-critical applications such as medical imaging, it is important to consider what, if any, guarantees can be given for reconstruction algorithms. Mathematically, there are many possible properties we could ask for. Stability results could allow us to control and estimate the error in the reconstruction given small deviations in the measured data. For iterative methods or methods that involve iterative optimisation schemes, we could require convergence in the sense that the iteration converges to a fixed point. We would probably also require that fixed point to be a "good" reconstruction. In section 2.1.3 we will define a regularisation scheme and require that, if there is noise in the data, a reconstructed solution converges to the noise-free solution, as the noise level decreases. A recent review paper [159] presents an overview of recent deep learning-based image reconstruction methods that fit into the different notions of convergence. There are some interesting trade-offs here, for example, a reconstruction method that always outputted the same constant image would be stable to changes in the measured data but would not be accurate. Alternatively, as discussed in [159], for a forward operator with a non-trivial null space, high accuracy requires large Lipschitz constants for the reconstruction operator and thus makes the reconstruction less stable.

It is well known for image classification that small, imperceptible changes in the original image can lead to confident erroneous classifications [82, 214, 69, 133]. From this, a whole field of research has emerged in adversarial machine learning *e.g.* [240, 237, 183, 149]. A model being stable to small deviations is sometimes called 'adversarial robustness'. The authors of [12] investigate well-known deep learning methods for image reconstruction [243, 232, 113, 197, 93] and give empirical examples of tiny, almost undetectable perturbations, both in the image and sampling domain, resulting in artefacts in the reconstruction, some of which are not obvious and could be missed. They also show how small structural changes, for example, a tumour, may not be captured in the reconstructed image and finally show how more samples may yield poorer performance. A subsequent paper [48] demonstrated a trade-off between stability and accuracy, with limits on how well a stable NN can perform in inverse problems. However, a later paper responds to this, finding that in the scenarios they consider deep-learning-based methods that are at least as robust as a popular non-learned method, Total Variation regularisation. They consider both random noise in the measurements and adversarial noise, where small deviations to the original measurements are chosen specifically to have the largest effect on the reconstruction [75].

The speed of research into generative models and deep learning means there is a gap between state-of-the-art methods and available mathematical results and theoretical guarantees. Future work in this area will continue to bridge this gap as well as providing insight and results that will guide future research.

Explainability Convergence and stability guarantees are steeped in the language of mathematics however, in the safety-critical applications of inverse problems, methods must be convincing to a wider audience. The idea of 'explainability' encompasses thoughts of a model being trustworthy, logical and reliable. All stakeholders need not understand the technical details but the concepts and motivations of the method should be understandable. For example, when classifying an image of a tissue sample as cancerous or not, a model could be made more explainable if it could highlight the area of the image that produced the decision [242]. Another avenue for explainability would be if the model could also output an equivalent healthy sample or, linking to the previous paragraphs on robustness, if it could output the smallest change in the tissue image that would change the classification. Finally, the model could also output a probability distribution, rather than a binary classification. Outputting 60% 'yes' and 40% 'no' is more informative than a simple 'yes' decision. Explainability is less well explored for image reconstruction. One common complaint is that methods are 'black boxes' and that we don't understand the implicit assumptions the models and architectures bring. Another common discussion is how to incorporate expert physical knowledge and modelling alongside data-driven methods. For this section, the conversation about future work in this area is perhaps less focused on "what" but "how". The research communities will need to engage with stakeholders to understand the context and subtleties of the real-world problems to design effective and explainable solutions.

Generalisability When designing inverse problem reconstruction algorithms, there is a trade-off between generalisability, the extent to which methods designed for one use case can be applied to other situations, and specificity, the accuracy of a method to a particular use case. A highly specific application could be acquiring a series of images of the same patient over time and using the first image as a guide for reconstructing subsequent images. This method is not general; if we used the image from one patient to guide the reconstruction of another, the reconstruction accuracy could decrease. In contrast, Total Variation (TV) regularisation [192] is an unlearned method that favours cartoon-like images with constant areas separated by sharp edges. TV is a general method with a wide range of applications but is not specific.

This trade-off between specificity and generalisability is similar to the 'bias-variance trade-off', a common phrase in machine learning. High-variance methods get a low error on the training dataset but at the risk of overfitting to noise or small fluctuations in the training dataset. In contrast, algorithms with high bias typically produce simpler models that may under-fit the training data. The high bias low variance case is likely to perform similarly on training and test data whereas the low bias high variance case can reproduce the training data so well

that it doesn't generalise.

Models that are more able to generalise may have reduced data requirements. For example, if the model can generalise to a new setting it may not need to be retrained on new data. More general models may also be able to overcome flaws in the data. Sample bias occurs when the realities of the environment in which a model will run do not reflect the dataset. For example, the creators of text-to-image software Dalle and Dalle-2 [182] have had to purposefully implement new techniques to ensure the generated image reflects the diversity of the world's population [185]. Their original results reflected the biases in the data it was trained on *e.g.* there were more images of male over female firefighters. In medical image reconstruction, in practice, imaging is done when there is a clinical need, such as injury or illness, and so we should be careful that models have either not just seen healthy examples in training or are capable of generalisation.

Recent high-profile deep learning methods such as GPT4 [171] and Segment Anything Model (SAM) [126] have focused on this breadth of applications. However, for scientific applications, where image acquisition methods, noise type, artefacts and datasets are so different to natural images, we are a long way off these large models. I do, however, expect though that future demand will be for more general models.

1.3 Overview of Research Presented in this Thesis

Generative models Suppose we want to train a model to take points sampled from some known distribution and output images similar to a training dataset. We make a 'low-dimensional manifold' assumption, that the desired images, the images in the training dataset, lie on some lower dimensional manifold of the full image space. The full image space $\mathcal{X}$ has a high dimension: if we treated $\mathcal{X}$ as a function on a 2D domain, then we would have infinite dimension, or in the discretised space, where $\mathcal{X}$ is defined on a 2D grid, we have a finite but high dimension given by the number of pixels. However, we are only interested in a subset of desired images, for example, those of a knee MRI ground truth images. This space of images has fewer degrees of freedom: the pixels in the background are all black; the underlying knee structures are always present and can only vary in size and location; tumours or growths have similar presentations, sizes shapes and locations; and there are similar textures, colours and repeating patterns. The space of knee MRI ground truth images is smaller than the full image space and we assume that these images lie on some lower dimensional manifold in the image space. See figure 1-2 for a pictorial representation of this and figure 1-1 for an artistic rendition by Dalle-2 of a low-dimensional cat manifold. The generator

Figure 1-1: A low dimensional cat manifold. Obtained from the Dalle-2 prompt "three cats walking along a spiral in space with dogs floating in the background"

$G : \mathcal{Z} \to \mathcal{X}$ takes points from the latent space $\mathcal{Z}$, which is lower in dimension than the image space, out outputs images in $\mathcal{X}$. The idea is that the generator G parametrises the lower dimensional manifold.

In addition, many generative models place a prior on the latent space which is a known distribution, usually a standard normal $\mathcal{N}(0, I)$ distribution. We can further ask, not only that the generator outputs images close to some training set, but that high probability points in the latent space are mapped to high probability images in the image space. The generator takes the distribution over $\mathcal{Z}$ and induces a probability distribution over the manifold in $\mathcal{X}$.

A generative model consists of a generator, a prior on the latent space and some mechanism for inducing a distribution of desirable images in the image space. By changing the mechanism for inducing a distribution, and the strategy for learning the generator parameters, we obtain different generative models. In chapter 3, we introduce some common generative models and formalise some of these ideas mathematically. Chapter 4 compares autoencoders (AEs), variational autoencoders (VAEs) and generative adversarial networks (GANs), and chapters 5 and 6 focus on VAEs in particular.

Generative regularisers In this thesis, we use generative models in a variational regularisation approach to inverse problems. The considered regularisers

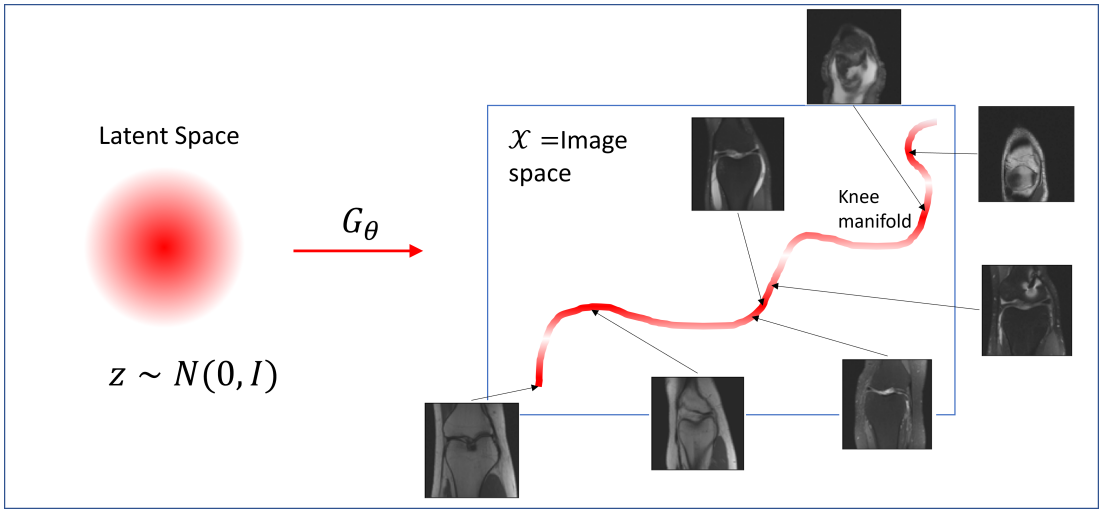


Figure 1-2: This diagram considers the low dimensional manifold assumption and generative models. We show the example Knee MRI ground truth images lying on the red lower dimensional manifold in the latent space. The generative model, G_θ, takes points in the latent space, z, and outputs points along the manifold. With a probability distribution on the latent space, this can induce a distribution on the image space. The generator G is taken to be a parametrised function, with parameters θ.

penalise images that are far from the range of a generative model that has learned to produce images similar to a training dataset. We name this family *generative regularisers*. Consider a general form,

$$x^* \in \arg\min_x \left\{ \mathcal{D}(Ax, y) + \lambda \left(\min_{z \in \mathcal{Z}} \mathcal{F}(G(z) - x) + \mathcal{R}_{\mathcal{Z}}(z) \right) \right\}, \tag{1.2}$$

where $\mathcal{D} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, $\lambda \in \mathbb{R}_{\geq 0}$, $\mathcal{F} : \mathcal{X} \to [0, \infty]$ and $\mathcal{R}_{\mathcal{Z}} : \mathcal{Z} \to [0, \infty]$. The first term is a data discrepancy term that encourages the reconstructed image to match the observed data. The scalar λ is a regularisation parameter and the bracketed section, $\mathcal{R}_G(x) = \min_{z \in \mathcal{Z}} F(G(z) - x) + R_{\mathcal{Z}}(z)$, is a regulariser, designed to be small if the reconstructed image, x, satisfies some desired property. In the case of generative regularisers, the desired property is that the reconstructed image is close, in the sense of the function $\mathcal{F}$, to something in the range of the generator, G. The final term $\mathcal{R}_{\mathcal{Z}}$ term contains extra information about the latent space and the locations in the latent space that maps to feasible images. In chapter 5 we also derive (1.2) from a Bayesian approach to inverse problems.

The benefit of retaining a variational regularisation approach is that we can take advantage of previous work mathematically modelling the forward and noise model, and tuning regularisation parameters, see *e.g.* [23] for more information on

variational regularisation. We can balance the terms of the objective (1.2), taking into account information from the prior and observed data. Another benefit of a generative model prior is that it can be explicitly sampled, and thus we can visually analyse the information it provides to the inverse problem.

By training unsupervised, without knowledge of the forward problem, there is no requirement for paired training data where both high-quality reconstructions and measured data need to be available. It also means we do not have to retrain the generative model for small deviations in the forward model. For example, changes in the sampling pattern in MRI or changes in the noise model or level. This makes the approach more general.

In chapter 4, we consider different choices for $\mathcal{F}$ in the literature and perform a literature review. We test three different choices of generative regularisers on the inverse problems of deblurring, deconvolution, and tomography. We show that restricting solutions of the inverse problem to lie exactly in the range of a generative model ($\mathcal{F}(x) = \iota_{\{0\}}(x)$ a characteristic function) can give good results either in the case where the observed data is particularly poor, or when the image to be reconstructed is sufficiently simple. However, we find that when the observed data improves, *e.g.* the noise is reduced or more measurements are taken, the reconstruction does not improve much or at all. Allowing small deviations from the range of the generator ($\mathcal{F}$ is the 2-norm, 1-norm or TV-norm) produces more consistent results on more complex datasets, but also, with a suitable choice of regularisation parameter, λ, the results continue to improve as data quality improves. Allowing small deviations gives flexibility for the generative regularisers to generalise to cases not seen during training.

The success of generative regularisers depends on the quality of the generative model. For example, if the generative model can output just a small range of images, and not the full range of potential inverse problem solutions, then restricting close to this range might not lead to good solutions. Practically, we also need our generators to be differentiable so we can differentiate (1.2) with respect to z. We propose a set of desired criteria to assess generative models, that are necessary but not sufficient for generative regularisers to be successful. In numerical experiments, we evaluate three common generative models: autoencoders, variational autoencoders and generative adversarial networks, against our desired criteria.

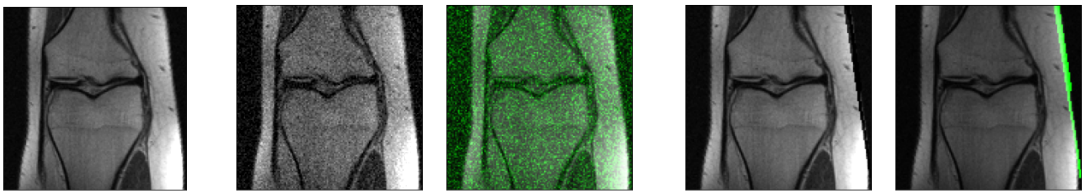
VAEs with structured covariance applied to inverse problems Figure 1-3 shows three example MR images: an original ground truth image (a), an image with additive Gaussian noise (b), and an image with the right edge of the knee moved (c). Both of the changed images are the same distance in the 2-norm to the original. Image (b) is not a desirable solution to an MRI inverse problem,

whereas (c) could be a possible solution. Any prior that we put on the inverse problem should be able to distinguish between the two.

In chapter 4 we discuss restricting solutions to be close to the range of a generative model trained on example ground truth images. In chapter 5, we will focus more on what it means to be 'close to' and investigate one avenue for learning a suitably weighted norm for the function $\mathcal{F}$ in (1.2). For each point on the manifold of generated images, $G_\theta(z)$ we also learn a symmetric positive definite covariance matrix $\Sigma_\theta(z)$, with height and width given by the number of image pixels, that induces a weighted norm $\|x\|_\Sigma^2 = x^T\Sigma^{-1}x$. Both G and Σ are parametrised functions, with parameters θ. For some intuition, consider first that $\Sigma_\theta(z)$ is a diagonal matrix. For each pixel, the corresponding value in the diagonal gives a weighting as to the allowable deviation from the range of the generator. The generative model may be very certain about a black pixel in the background of the image, giving a low value in the covariance matrix and high weighting in the norm, but be uncertain around an edge, giving a higher variance and lower weighting in the norm. In this way, the added noise in the background of figure 1-3(b) could be penalised highly, while the edge changes in (c), less so.

Extending this analogy, we consider not just areas of the image with high or low uncertainty, but also the structures in the image. For example, there may be a high variance along an edge, but we might also want to take into account the structure of the edge. We might wish to penalise moving the whole edge one pixel to the left much less than breaking up the edge completely. This can be achieved with a dense covariance matrix, $\Sigma_\theta(z)$. See figure 1-4 for a toy example of an L2 norm weighted by a dense matrix being able to differentiate between moving a whole edge and the breaking up of an edge. In chapter 5 we embed these ideas into a Bayesian approach to inverse problems and use a variational autoencoder with structured uncertainty to learn this covariance matrix. We can visualise the generated images and learned covariances, making explicit the priors we placed on the inverse problem. We compare against other state-of-the-art deep learning approaches to inverse problems with favourable results.

Generative regularisers without ground truth data As discussed in section 1.3, data can be difficult and expensive to obtain or simply unavailable. In chapter 6, we consider learning a generative model for high-quality reconstructions, from a dataset of noisy or partial measurements, where the forward model for the observations is known. We base our model on a VAE, optimising a bound on the Kulback-Lielbler, in the measurement space, between the training dataset and generated images observed under the known forward model. We are careful with our choice of VAE generative model and training loss in order to keep an explicit probability distribution over the generated images in the image space. We demonstrate our proposed *noisyVAE* on hand-built datasets, learning from noisy



(a) Original (b) Additive Gaussian Noise (c) Changing edge location

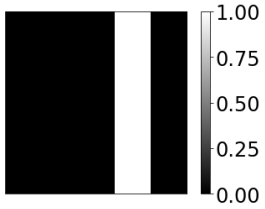
Figure 1-3: The left-hand side of (b) and (c) are deviations from the original image (a). The deviations are highlighted in green. Both (b) and (c) are the same distance from (a) when measured in the 2-norm. Being able to tell these two options apart is a challenge in evaluating the quality of reconstructed images and when designing priors for inverse problems.

and blurred data, and demonstrate their effectiveness for use in inverse problems. One example inverse problem where data is difficult to obtain is Positron Emission Tomography (PET) due to inherent noise in the measurement procedure. In addition, in this chapter, we demonstrate how the PET forward problem, with a Poisson noise model, could be effectively approached. Finally, we set out avenues for future work, including learning generators with structured covariance models from noisy and incomplete data.

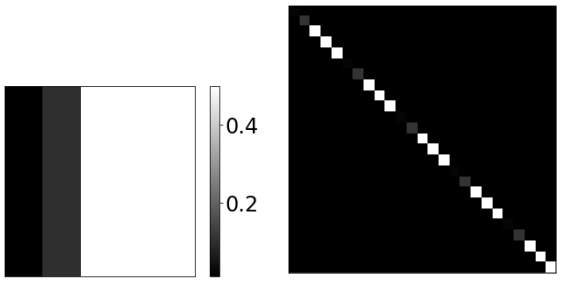
Which of the identified problems does it tackle? In this work, we will consider learned regularisation methods, where the learning is unsupervised, without the need for paired training data, that do not need to be re-trained with small changes in the forward model or noise level. This reduces data and training requirements and also makes these methods more generalisable. Using generative models as priors to inverse problems has the added benefit of being able to sample from the generative model, and thus visualise explicitly the prior information being provided to the reconstruction. This benefits explainability. Finally, we consider learning generative models without ground truth data, for use in inverse problems, again reducing data requirements. We do not tackle convergence guarantees although we will signpost to the literature, where available.

1.4 Evaluation Methods

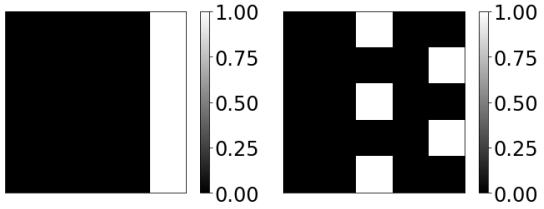
Some of this work will be evaluated on small, hand-built test datasets, such as a dataset of rectangles and circles in chapter 4 and single and overlapping ellipses in chapter 6. The benefit of such a dataset is that it can be specifically designed to test certain properties. For example, the rectangles and circles dataset tests the ability of generative models to produce sharp edges and corners, which can be difficult for VAEs. We will also evaluate our work against existing deep learning



(a) Original, x



(b) Diagonal covariance matrix, Σ_D. The left shows the individual pixel variances and, on the right, the corresponding diagonal covariance matrix.



(c) Two variations on the original, x_1 (left) and x_2 (right)

(d) A dense covariance matrix, Σ, which has pixels in the same column as highly correlated.

Figure 1-4: Take an image x, a 5×5 array containing one vertical edge in the 4th column, given in (a). Assign each pixel a variance with the highest variance around the edge (left of (b)). This gives a diagonal covariance matrix for the image, Σ_D, (right of (b)). The image in (c) gives the case of the edge moving one pixel to the right (x_1 on the left) or breaking up (x_2 on the right). With the covariance model given by Σ_D both x_1 and x_2 are the same distance from x, *i.e.* $(x_1 - x)^T\Sigma_D^{-1}(x_1 - x) = (x_2 - x)^T\Sigma_D^{-1}(x_2 - x)$. The dense covariance matrix in (d) represents the case where pixels in the same column are highly correlated but pixels in different columns are independent. Under this matrix, $(x_1 - x)^T\Sigma^{-1}(x_1 - x) << (x_2 - x)^T\Sigma^{-1}(x_2 - x)$.

datasets such as MNIST [136], in chapter 4, knee fastMRI in chapters 4 and 5 and PET data in chapter 6. With these, we can compare our work on realistic images, more reflective of real-world scenarios, and with other methods. These datasets, although 'real' in that they are real digits or real brain MR images, as discussed above, have been nicely curated, cropped and centred for us. As discussed in the relevant chapters, we also do some resizing and rescaling, to better suit our architectures and the size of our memory. Our datasets can't be considered 'real' in the sense of 'raw'. This is not to say that our results are

worthless, learning and insight can still be gained and indeed if methods failed on these curated datasets they would certainly fail on a more 'raw' dataset.

Evaluating generative models is complex and we will discuss this in more detail in chapter 4. For a known forward problem and observed data, inverse problem solutions can be evaluated on the extent to which the reconstructed image matches the observed data, *e.g.* by taking just the first term of (1.2). In addition, when using existing or hand-built datasets, we usually have access to the ground truth to compare and can measure the distance between reconstructed and ground truth images. As seen in figure 1-3, measures such as the 2-norm can be tricky. If a radiologist were evaluating figure 1-3, if the ground truth image were (a), it may be that they prefer (b), which although noisy is anatomically correct, in comparison to (c) with the structural change in the edge, but both have the same error in the 2-norm. There exist other measurement metrics, e.g. peak signal-to-noise ratio (PSNR) or structural similarity index measure (SSIM), but none are a perfect measure of image quality. Given the issues with quantitative measures, qualitative evaluation is important. We will endeavour to provide example images for analysis throughout the thesis.

As discussed, inverse problems is an established research area and deep learning and inverse problems is a fast-growing domain. Our proposed methods do not exist in a vacuum and so we compare against both unlearned variational regularisation approaches, such as TV; other learned regularisation approaches that don't require paired data; and other learned methods, that require paired training data. We consider both accuracy and generalisability. It is important to note, however, that in most cases one can't just take the most recent paper and off-the-shelf apply it to your data. The website "papers with code" reports that on average, so far in 2022, less than 30% of papers were published with code [175]. Even if the code is available, well maintained, well documented and runs off-the-shelf on your specific hardware, you still may need to adapt the architectures for your dataset and forward problem. You may also have to retune parameters such as learning rates or initialisations. Some models may also be too large or expensive to train [201]. As good scientists, we endeavour to get the best results to provide a fair comparison. However, with limited resources and time, it is likely that with more effort and expertise, results could be improved.

1.5 Thesis Structure

This thesis has two background and three core chapters and we list their contributions below.

- Chapter 2 contains background information on inverse problems and deep learning before a brief review of current deep learning approaches to inverse

problems.

- Chapter 3 introduces common generative models including autoencoders, variational autoencoders and generative adversarial networks, that will be referenced in subsequent chapters. It also discusses some more recent models.
- Chapter 4 introduces generative regularisers, providing a literature review and then numerical examples and comparisons. This chapter also sets out qualitative desired criteria for generative models for use in inverse imaging problems. The majority of this chapter is under review and available as a preprint [64].
- Chapter 5 considers VAEs with structured covariance and provides detailed numerical experiments for compressed sensing MRI on the fastMRI dataset. The majority of this chapter is available as a pre-print [65].
- Chapter 6 looks at learning generative models without ground truth data using only observed data, where the forward model is known. It introduces a loss function for training a VAE on observed data and demonstrates its efficacy on datasets of ellipses. It has some early proof of concept results on positron emission tomography (PET) images.
- Finally, chapter 7 concludes this thesis by revisiting the key themes set out in the introduction and identifies areas for future work.

Chapter 2

Inverse Problems and Deep Learning

In this chapter we will introduce background information on inverse problems and deep learning. We also include a review of deep learning approaches to inverse problems in order to set generative regularisers within a wider context.

2.1 Inverse Problems

Recall from equation (1.1) that solving an inverse problem is the task of computing an unknown quantity, $x \in \mathcal{X}$, from observed (potentially noisy) measurements, $y \in \mathcal{Y}$, related by a forward model, $A : \mathcal{X} \to \mathcal{Y}$. In this work, we assume that the forward model A is known and easy to compute. Measurement errors or additional, poorly understood physical processes mean that the observations come with additional noise and we generally assume that this noise model is known or can be approximated. For example, the case of additive noise, ϵ, gives

$$Ax + \epsilon = y. \tag{2.1}$$

2.1.1 Example Applications

The applications we focus on in this report are inverse problems in imaging. Some examples running through this thesis include:

- *Image denoising* - Take the forward model, A, in (2.1) to be the identity operator and we can define the inverse problem to be $y = x + \epsilon$. Here we wish to recover $x \in \mathcal{X}$ on observing the noisy $y \in \mathcal{Y} = \mathcal{X}$.

- *Deconvolution* - Movement during the image capture process or out-of-focus optics can be modelled using a convolution. Take $\mathcal{X} = \mathbb{R}^{d_1 \times d_2}$ and $\mathcal{Y} = \mathbb{R}^{m_1 \times m_2}$ and define a forward model

$$y_{u,v} = \sum_{s=1}^{a} \sum_{t=1}^{b} K_{s,t} x_{u-s,v-t}$$

  where x is the original image, y the observed image and K is a $a \times b$ matrix. The image x is either padded to ensure that $\mathcal{Y}$ has the same dimension as $\mathcal{X}$, or $\mathcal{Y}$ has a smaller dimension than $\mathcal{X}$.

  If the convolution kernel is known then the inverse problem could be tackled using a Fourier transform, however, this is always ill-posed (in infinite dimensions). The case of an unknown kernel is called 'blind convolution' and is even more challenging.

- *Compressed sensing* - Consider $y = Ax$ to be an under-determined linear system where A is an $\mathbb{R}^{m \times d}$ Gaussian random matrix, $x \in \mathbb{R}^d$ is a vectorised image and $d \gg m$, see for example [55]. The matrix A will have a non-trivial kernel and so, if a solution exists, there will be infinitely many solutions. There are usually two conditions under which recovery of the solution is possible, the first is that the solution is sparse under some domain and the second is a condition on the matrix A to behave like an isometry, a distance-preserving map, restricted to the set of possible sparse solutions. In chapter 4, we take A to be a random matrix, with entries drawn independently from a Gaussian distribution, as many classes of random matrices have been shown to be suitable for compressed sensing.

- *Tomography* - Tomography is the technique of imaging by sections through the use of a penetrating wave. For many tomography applications, the underlying forward problem is the X-ray transform, given by

$$y(\theta, s) = (Ax)(\theta, s) = \int_{t \in \mathbb{R}^2:\ \ t \cdot \theta = s} x(t) dt. \tag{2.2}$$

  Integrals are taken along lines in the image that satisfy $t \cdot \theta = s$ for a given θ, the x-ray angle, and s, the distance from the object centre. In this thesis, we usually consider discrete data where x is defined on a grid and so a discrete set of θ and s are chosen and the integral is approximated.

  In X-ray computed tomography (CT), the unknown quantity x represents a spatially varying density that is exposed to radiation from different angles and absorbs radiation according to its material or biological properties.

  For positron emission tomography (PET), a short-lived radioactive tracer is injected into the patient. As the tracer decays it emits a positron which

travels in the tissue for a short distance before interacting with an electron. The encounter annihilates both the electron and positron producing a pair of photons which travel in opposite directions. By measuring the locations of the photons hitting a scanner surrounding the patient, a line can be drawn between the two detected positions, which contains the location of the original event. The process can be approximated using (2.2) with a suitable noise model. For more detail see *e.g.* [184] and chapter 6.

- *Compressed sensing MRI* - MRI machines employ powerful magnets which produce a strong magnetic field that forces protons in the body to align with that field. Stimulating these protons with radio waves breaks the alignment and sensors detect the energy released when protons realign with the magnetic field. It is a non-damaging process, in contrast to the X-ray radiation dose in CT, and can obtain clear images of soft tissue such as muscles, ligaments, and tendons. Mathematically, let $\mathcal{X} \subset \mathbb{C}^d$, where the $d = d_1 d_2$ is the image dimension and complex vectors in $\mathcal{X}$ can be arranged into a $d_1 \times d_2$ complex image. Let $\mathcal{Y} \subset \mathbb{C}^{qm}$, be the measurement space, often called *k-space*. Consider $A = S \circ F \circ Q$ where Q performs point-wise multiplication with q coil sensitivity maps; F is a Fourier transform, and M is an under-sampling mask that returns only values at specified locations in k-space, according to a selected sampling pattern. MRI acquisition is slow (typically 15-90 minutes according to the NHS [161]) and the idea is by taking fewer k-space samples, hence fewer measurements, acquisition can be quicker and therefore cheaper and more conformable for the patient. For a recent review see *e.g.* [234]. Note that, in chapter 5, we deal with real magnitude images, and so define $\mathcal{X} \subset \mathbb{R}^d$ and define an extra operator $B : \mathbb{R}^d \to \mathbb{C}^d$ which maps into the complex domain by setting the complex component to be equal to zero, $B(x) = x + 0i$. For simplicity, we also ignore coil sensitivity maps, replacing Q with an identity operation. Thus $A = S \circ F \circ B$.

2.1.2 Ill-posedness

For arbitrary spaces $\mathcal{X}$ and $\mathcal{Y}$, a problem is *well-posed*, in the sense of Hadamard [91], if it satisfies the following

- **Existence**:- For all observed measurements there exists at least one solution to the problem.
- **Uniqueness**:- For all observed measurements there is at most one solution.
- **Stability**:- The solution of the problem depends continuously on the observed data.

Problems that are not well-posed, in the sense of Hadamard, are termed *ill-posed*.

For example, in the discrete case for $Y = \mathbb{R}^m$, $A \in \mathbb{R}^{m \times d}$ and $\mathcal{X} \in \mathbb{R}^d$, if $d < m$, there exists points in the measurement domain $\mathcal{Y}$ for which there are no solutions to $Ax = y$. Conversely, if $m < d$ and the system is under-determined, the null space of A will be non-trivial, and solutions may not be unique. Finally, if the condition number of A, the ratio between the biggest and smallest eigenvalues of A, is very large, then the problem will be sensitive to even very small changes in the observed measurements. In the previous examples, deconvolution and tomography inverse problems can be unstable to small changes in the measured data. Both the compressed sensing and compressed sensing MRI inverse problems have a non-trivial kernel and thus if there exists a solution, it will not be unique.

Hadamard argued in that original paper [91] that well-posed problems correspond to physical scenarios and are therefore worth studying. However, in the introduction, we gave many examples of ill-posed inverse problems that are worth studying and indeed I believe that the additional challenge of ill-posedness makes inverse problems such an interesting research area for mathematicians.

Inverting the Forward Operator The ill-posedness caused by the existence or uniqueness of solutions could be alleviated by re-writing the problem as a least squares problem, searching for solutions x that minimise

$$\|Ax - y\|_{\mathcal{Y}}^2. \tag{2.3}$$

In the case of multiple minima of this least squares formulation, one could define a solution uniquely, for example, as the minimum solution, as measured by a norm in $\mathcal{X}$. With suitable choices of operators, spaces and norms, this minimum norm solution to the least squares problem coincides with the Moore-Penrose pseudoinverse (see *e.g.* [22]). This is now an inverse problem reconstruction approach that gives both existence and uniqueness for the inverse problem.

However, this inverse is not necessarily stable or of acceptable quality and often further work is required.

2.1.3 Variational Regularisation

Regularisation replaces the original problem with a closely related problem that has better stability properties. Variational regularisation, sometimes called Tikhonov-type regularisation, looks for solutions to the inverse problem as solutions to the minimisation problem:

$$x^* \in \arg\min_x \mathcal{D}(Ax, y) + \lambda \mathcal{R}(x), \tag{2.4}$$

where $\mathcal{D} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq} 0$ is a similarity measure, which we will assume smooth in its first argument and non-negative; the constant λ is a regularisation parameter

and $\mathcal{R} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a regularisation function. Note that we could extend to the case where $\mathcal{D}$ is non-smooth in the first argument *e.g.* if $\mathcal{D}$ were the 1-norm.

Theoretical Properties As discussed in [196] there are a number of desired theoretical results that are of interest in regards to variational regularisation methods:

- *Existence*: For a fixed regularisation parameter, λ, and observed data, y, there exists (unique) minimisers of (2.4).
- *Stability*: For a fixed regularisation parameter, λ, the regularised solution, x^*, to (2.4) depends continuously on the observed data,y.
- *Convergence*: As the regularisation parameter tends to zero, $\lambda \to 0$, and the noise in the observed data tends to zero, $y \to Ax^\dagger$ for some $x^\dagger \in \mathcal{X}$, then x^* in (2.4) converges to a solution of $y = Ax$.

One might then go on to consider both convergence rates and bounds on stability.

Choosing a Regularisation Function The idea of the regularisation function is that it is small when some desired property of the reconstructed image is satisfied. Some examples include:

- Tikhonov regularisation, $\mathcal{R}(x) = \|x\|^2$, encourages the recovered unknown to be small in the 2-norm. More generally, Tikhonov-Phillips regularisation [216, 179], $\mathcal{R}(x) = \|Bx\|^2$, where $B : \mathcal{X} \to \mathcal{Z}$ is a linear operator, penalises some features of x.
- Total variation (TV) regularisation [192] aims to denoise images while preserving edges, favouring images that have a sparse gradient. The regularisation functional is the one norm of the gradient of the image, $\mathcal{R} = \|\nabla x\|_1$. For an image $x \in \mathbb{R}^{d_1 \times d_2}$, in matrix form, the total-variation can be approximated

$$\mathcal{R} = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2}. \tag{2.5}$$

- Sparsity penalties encourage sparsity of the reconstructed image under a certain representation. For example, for $x \in \mathbb{R}^d$, the 1-norm is often used as a regulariser to enforce sparse solutions. For example take $R(x) = \|Bx\|_1$ where $B : \mathcal{X} \to \mathcal{Z}$.

As discussed in the introduction, choosing a suitable regularisation term is a trade-off between specificity and generalisability. TV regularisation can give reasonable results on a wide range of datasets at a cost of providing cartoon-like

images and blob-like artefacts in some cases. Hand-crafting a regularisation strategy for a given dataset can require expert knowledge and risks overfitting to only a specific special case. A natural question to ask is: given a set of images, which regulariser would work well? Alternatively, how can we produce regularisers that are tailored to specific data or tasks? This thesis attempts to answer this question through learned regularisation.

For this approach, we still need to choose the regularisation parameter λ. One common a-posterior parameter choice rule is the Morozov discrepancy principle [155]. If $Ax = y$ and noisy data y^δ is observed, with $\|y - y^\delta\| \leq \delta$, then the Morozov discrepancy principle chooses the largest regularisation parameter λ such that $\|Ax^*(\lambda, y^\delta) - y^\delta\| \leq \delta$, where $x^*(\lambda, y^\delta)$ is the regularised solution with observed data $Y = y^\delta$ and regularisation parameter λ.

Note that variational regularisation is not the only possible form of regularisation. For a review of modern regularisation methods see [23].

2.1.4 Bayesian Approach

An alternative approach to solve the inverse problem given in (2.1) is through Bayesian statistics. With a known probability distribution on the additive noise, define a *likelihood* distribution, a probability distribution on data y given image x. The *posterior* probability of x given data y is given by Bayes' theorem:

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}. \tag{2.6}$$

Here, $p(x)$ is a *prior* model that represents the known information about the model parameter before observing data y. A prior could be uninformative, *e.g.* uniform over an image space, or could be very specific, *e.g.* centred on a particular example image. The benefit of this approach is that it allows for a complete statistical analysis including quantification of the uncertainty. Sampling from the posterior, for example through Monte Carlo methods, allows for not just one reconstructed image but a range of images and an idea of uncertainty. However, there are several hurdles to overcome. Except for a small number of carefully chosen conjugate priors, the posterior is intractable, with no closed-form expression. As we shall see below, choosing a 'good' prior is also analogous to choosing a good regularisation parameter and, as discussed, this choice can be challenging. For a textbook on Bayesian inverse problems see [213].

Bayesian and Variational Equivalences Consider calculating a mode of the density $p(x|y)$, also called the maximum a posteriori (MAP) estimate. From

Bayes' theorem this gives

$$x \in \arg\max_x \frac{p(x)p(y|x)}{p(y)} \tag{2.7}$$

$$\iff x \in \arg\max_x p(x)p(y|x) \tag{2.8}$$

$$\iff x \in \arg\min_x \{-\log p(x) - \log p(y|x)\}. \tag{2.9}$$

Consider $-\log p(x) := \mathcal{R}(x)$ to be the regularisation term. For example if we put a standard normal prior on $\mathcal{X} = \mathbb{R}^d$, $p(x) \propto \exp(-\frac{1}{2}\|x\|_2^2)$ and $-\log p(x) = \frac{1}{2}\|x\|_2^2$ and we have Tikhonov regularisation. Similarly, a Laplace prior will give regularisation in the 1-norm. The second term, $-\log p(y|x)$ corresponds to the data discrepancy term. For example, if we assume that the added noise, ϵ is Gaussian, with zero mean, and (potentially unknown) standard deviation γ *i.e.* $\epsilon \sim \mathcal{N}(0, \gamma^2)$ we have that $p(x|y) \propto \exp\left(-\frac{\|Ax-y\|_2^2}{2\gamma^2}\right)$. This gives $-\log p(y|x) = \frac{\|Ax-y\|_2^2}{2\gamma^2}$. Incorporating this into (2.9) gives

$$x_{\text{MAP}} \in \arg\min_x \|Ax - y\|_2^2 + \gamma^2 \mathcal{R}(x) \tag{2.10}$$

which can be directly compared to (2.4), the variational regularisation framework. The noise level in the data, γ, acts as a regularisation parameter, balancing between the prior and the data. Other possible likelihood distributions could be derived from other noise models. For example, see chapter 6 for a case of Poisson noise.

2.1.5 Inverse Problem Discretisation

A two-dimensional image could be given as a two-dimensional grid of pixels or it could be defined as an infinite-dimensional function over a two-dimensional domain. A thorough discussion of inverse problem discretisation is out of the scope of this thesis. Instead, in this work, we choose the former and define $\mathcal{X} = \mathbb{R}^d$ a d-dimensional vector space where vectors in $\mathcal{X}$ can be arranged in a 2D grid (size $d_1 \times d_2$ where $d_1 d_2 = d$). Also take $\mathcal{Y} = \mathbb{R}^m$ ($\mathbb{C}^m$ for the compressed sensing MRI problem) and so $A \in \mathbb{R}^{m \times d}$ ($\in \mathbb{C}^{m \times d}$).

2.2 Optimisation Algorithms

To optimise (2.4) we utilise several different optimisation schemes. First consider gradient descent with backtracking, algorithm 1 (see *e.g.* Algorithm 9.2 of [30]). The method initially takes a large step size in the direction of the negative gradient and checks to see if this decreases the objective sufficiently. If not, the method

iteratively shrinks the step size (i.e., "backtracking") until the update reduces the objective sufficiently. Backtracking gives some convergence guarantees, ensuring, for example, that the iterates don't oscillate around a critical point.

Algorithm 1 Gradient Descent with Backtracking to solve $\min_z f(z)$.

Initialise z_0, $L > 0$, $0 < \eta_0 < 1$, $\eta_1 > 1$.
for $i = 1, ..., K$ **do**
    Let $\tilde{z}(L) := z_{i-1} - \frac{1}{L}\nabla f(z_{i-1})$
    while $f(\tilde{z}(L)) \geq f(z_{i-1}) - \frac{1}{2L}\|\nabla f(z_{i-1})\|_2^2$ **do**
        $L = L\eta_1$
    $z_i = \tilde{z}$ and $L = L\eta_0$.

When optimising over two or more variables, we can use alternating gradient descent with backtracking, algorithm 2, which takes a backtracking gradient step over each variable in turn, allowing a different step size for each variable.

Algorithm 2 Alternating gradient descent with backtracking to solve $\min_{z,x} f(z, x)$.

Initialise z_0 and x_0, $L_z > 0$,$L_x > 0$, $0 < \eta_0 < 1$ and $\eta_1 > 1$
for $i = 1, ..., K$ **do**
    Let $\tilde{z}(L_z) := z_i - \frac{1}{L_z}\nabla f(z_i, x_i)$
    while $f(\tilde{z}(L_z), x_i) \geq f(z_i, x_i) - \frac{1}{2L_z}\|\nabla f(z_i, x_i)\|_2^2$ **do**
        $L_z = L_z\eta_1$
    Let $z_{i+1} = \tilde{z}(L_z)$ and then $L_z = L_z\eta_0$
    Let $\tilde{x}(L_x) := x_i - \frac{1}{L_x}\nabla f(z_{i+1}, x_i)$
    while $f(z_{i+1}, \tilde{x}(L_x)) \geq f(z_{i+1}, x_i) - \frac{1}{2L_x}\|\nabla f(z_{i+1}, x_i)\|_2^2$ **do**
        $L_x = L_x\eta_1$
    Let $x_{i+1} = \tilde{x}(x_L)$ and $L_x = L_x\eta_0$

Consider instead that (some part of) the function you wish to minimise is not differentiable. For example, if it contained the 1-norm or TV semi-norm. We can instead use the proximal, $\text{prox}_h(z) = \arg\min_x\{h(x) + \frac{1}{2}\|x - z\|_2^2\}$, which projects solutions onto an admissable space given by the regulariser, h in the case. We take a Proximal Alternating Linearised Minimisation (PALM) approach, algorithm 3, [25]. Note that in the version given in algorithm 3, we do not assume we know the Lipschitz coefficients of the terms in our objective and so estimate them using a line search.

Algorithm 3 PALM with backtracking to solve $\min_{z,u} f(z,u) + g_1(z) + g_2(u)$.

Initialise z_0, u_0, $L_z > 0$, $L_x > 0$, $0 < \eta_0 < 1$ and $\eta_1 > 1$.
for $i = 1, ...K$ **do**
  Let $\tilde{z}(L_z) := \text{prox}_{\frac{1}{L_z} g_1}(z_i - \frac{1}{L_z}\nabla_z f(z_i, u_i))$
  while $f(\tilde{z}(L_z), u_i) > f(z_i, u_i) + \nabla_z f(z_i, u_i)^T(\tilde{z}(L_z) - u_i) + \frac{L_z}{2}\|\tilde{z}(L_z) - z_i\|_2^2$ **do**
    $L_z = L_z\eta_1$
  Let $z_{i+1} = \tilde{z}(L_z)$ and then $L_z = L_z\eta_0$
  Let $\tilde{u}(L_u) := \text{prox}_{\frac{1}{L_u} g_2}(u_i - \frac{1}{L_u}\nabla_u f(z_{i+1}, u_i))$
  while $f(z_{i+1}, \tilde{u}(L_u)) > f(z_{i+1}, u_i) + \nabla_u f(z_{i+1}, u_i)^T(\tilde{u}(L_u) - u_i) + \frac{L_u}{2}\|\tilde{u}(L_u) - u_i\|_2^2$ **do**
    $L_u = L_u\eta_1$
  Let $u_{i+1} = \tilde{u}(L_u)$ and then set $L_u = L_u\eta_0$

2.3 Deep Learning

When giving talks to mathematicians, I find it is often worth avoiding the words "machine learning". It generally causes a not insignificant portion of the audience to switch off, believing it to be some form of dark magic, or dismissing the work as immediately uninteresting. Using the phrase 'function approximation' or 'parameterised function' generally gives a better response. This is not misleading, in essence, deep learning aims to estimate a function f that maps input u to output v. Using training data, some loss function and a suitable optimisation scheme, one hopes to choose some parameters, θ, such that the parametrised function f_θ approximates f well. Finally, we hope that f_θ makes accurate predictions based on unseen data.

2.3.1 Choosing a Parameterised Function

Feed forward neural networks The basic unit of a neural network is a *neuron* (or *node*). Given a vector input $u_i \in \mathbb{R}^{n_i}$, weights $W_{i+1} \in \mathbb{R}^{n_{i+1} \times n_i}$, bias term $b_{i+1} \in \mathbb{R}^n_{i+1}$ and non-linear activation function, σ_{i+1}, the neuron outputs:

$$u_{i+1} = \sigma_{i+1}\left(W_{i+1}u_i + b_{i+1}\right). \tag{2.11}$$

The activation function is usually applied element-wise and activation functions used in this work include: Sigmoid function: $\sigma(x) = \frac{1}{1+\exp(-x)}$, Rectified Linear Unit: $\text{ReLU}(x) = \max(0, x)$ and Leaky Rectified Linear Unit: $\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$ where α is some small positive constant.

The basic structure of a neural network contains multiple nodes arranged in layers. Given layer dimensions n_0, n_1 and n_2, input $u = u_0 \in \mathbb{R}^{n_0}$ and output $v = u_2 \in \mathbb{R}^{n_2}$, an example of a 2-layer network is given in the following equation:

$$v = (f_\theta(u)) = \sigma_2 \left(W_2 \left[\sigma_1 \left(W_1 u + b_1 \right) \right] + b_2 \right), \quad i = 1, ..., n_2. \tag{2.12}$$

This is an example of a fully connected feed-forward neural network that first takes a weighted sum of the input variables, adds a bias and then applies an activation function, before repeating for the second layer. In the case of (2.12), the parameters θ would be the set $\{W_1, W_2, b_1, b_2\}$. Note that throughout this work, the parameters θ, are also called *weights* and this refers to any trainable neural network parameters including biases.

Modern architectures Fully connected feed-forward neural networks have been surpassed by other architectures in most domains. For example, in imaging, convolutional neural networks [136] learn convolution kernels (or filters), which have a lower dimension than the full image and are passed over the image, transforming it based on the kernel values. It has two major benefits: the focus on local structures and repeated calculations across the image seems to work well with natural images and, also, there is a significant parameter, and therefore memory, reduction compared to fully connected layers.

Layers are also commonly built into structures, such as residual neural networks [96] which mimic the shape of an iterative update of a discretised ordinary differential equation solver and UNETs [189] which contract and then expand in dimension, causing an information bottleneck. Both require the use of skip connections [96] to pass data between layers.

There are a growing number of choices for architectures with new and exciting possibilities and applications. Transformer architectures [221], for example, were originally introduced for text-to-text challenges, such as translation, but have fast become crucial in large image models [68]. Transformers can take an input (*e.g.* an image) and, in parallel, attention layers try to capture the relations between the different input values. The resulting information could be used, for example, to predict the value of missing pixels in an image [45].

2.3.2 Determining Suitable Weights

Neural networks, with as few as one hidden layer, are universal approximators. This means that, under some conditions on the activation function and with sufficient nodes, they can approximate any continuous function on a compact set-up to arbitrary precision [56, 73, 102, 139]. The now pertinent question is: a given network may be expressive enough to estimate the function we want, but can the parameters required be found?

Data Sets The type of data determines the type of learning we can do. There are three broad types: supervised, unsupervised and semi-supervised learning. The definitions generally refer to the following:

- *Supervised*: Input data and the corresponding output data are known for the function we wish the neural network to approximate *i.e.* training set $\{u^j, v^j\}_{j=1}^N$, where $f(u_j) = v_j$ for $j = 1 \dots N$.
- *Unsupervised*: Datasets consists of input data without corresponding output data *i.e.* $\{u^j\}_{j=1}^N \subset \mathcal{X}$, where $\mathcal{X}$ is some space we are interested in investigating.
- *Semi-supervised:* Datasets are any mixture of the above. For example, there could be a selection of input and output data but they may not necessarily correspond. Else there may be a small amount of corresponding input and output data alongside a larger amount of either input or output data.

Loss function The network is trained through the minimisation of a loss function, usually averaged over the training data with parameters consisting of the network weights. For example, in supervised learning, one could take the objective as to minimise $\sum_{j=1}^N \|v^j - f_\theta(u^j)\|^2$ with respect to θ for some norm.

Learning algorithms Minimisation of the loss function is usually done through gradient descent methods, usually stochastic gradient descent [204] and its variants *e.g.* Adam [123]. Stochastic gradient descent computes gradient calculations over just an individual sample or small batch of training data with each iteration, reducing the computational cost.

In addition to the weights, there are often a large number of parameters that are not learnt but set before training, such as learning rate, number of layers, number of nodes per layer, iteration numbers, activation functions, regularisation parameters and potentially many more. It is common practice to use validation sets to tune these parameters but this can be time-consuming.

Automatic differentiation Optimising the loss function using gradient descent requires calculating gradients of the network outputs with respect to the weights, $\frac{\partial f_\theta}{\partial \theta}$. In addition, throughout this work, we require the gradient of the network output with respect to its input, $\frac{\partial f_\theta(u)}{\partial u}$. If the gradients of the activation function are known, these gradients can be calculated by repeated use of the chain rule, sum rule and product rule. This is efficiently calculated using automatic differentiation packages, such as TensorFlow.

2.3.3 Learning Neural Network Weights is an Inverse Problem

Learning the weights of a neural network is an inverse problem. The unknowns are the weights of the network, the observed data is the training dataset and the forward model is the application of the neural network, with given weights. This inverse problem is ill-posed. For example, the choice of f_θ could not be expressive enough to approximate the function f and a solution to the inverse problem wouldn't exist. Even if a solution exists, the underlying structure of most neural networks means this solution will not be unique. For example, swapping two nodes in the same layer in (2.12) would lead to the same solution.

In addition to the ill-posedness, the loss functions are usually non-convex so gradient descent-based methods can get stuck in local minima. Suboptimal local minima do exist *e.g.* [77] shows that poor local minima can readily be found with a poor choice of hyperparameters. We would usually expect neural networks with a large number of parameters to overfit to a training dataset and thus see a large deterioration in performance when tested on an unseen dataset. As discussed in the introduction, we desire neural networks to be generalisable: the network should give good results on data not seen in training.

There are a variety of 'tricks' available that can be utilised to improve network properties, for example, initialisation strategies and regularisation techniques such as weight penalties (*e.g.* section 5.2.2 of [80]), drop out layers [211] or batch normalisation [106]. The choice of network architecture is also assumed to have a regularising effect, *e.g.* the bottleneck layer in an autoencoder [130] forces the network to develop a compact representation of the input data and it thus may not be possible to learn the details of high-frequency noise and this information is lost. Stochastic gradient descent methods may also implicitly regularise: often SGD will converge to solutions with smaller norm [239]. Data augmentation can also improve accuracy and stability. Again, often the behaviour of these is not well understood. For example, [239] shows that although explicit regularisation, such as weight penalties, can improve generalisation, regularisation is neither necessary or sufficient for generalisation. They also show how popular deep learning architectures can fit random labels successfully and quickly, suggesting implicit regularisation by architecture is not always sufficient.

2.4 Deep Learning and Inverse Problems

In this section, we provide a brief overview of approaches to image reconstruction using deep learning. We try and split our overview based on the graph given in figure 2-1. On the x-axis, we have the type of training data: no data, ground truth only data or paired training data. On the y-axis, we have the role of the forward

Method	Training data needs	Training data amounts	Forward model known during training	Image Reconstruction method	Generalisation Capabilities	Execution Speed	Theoretical Guarantees	Examples
Deep Image Priors	None	None	N/a	Iterative	Completely general: no training	Slow	None in basic form [222] but variations can have good properties [90]	[222, 219, 59]
Naiive Direct	Supervised: Paired data and good reconstructions	Extremely large	No	One step	Low: generalisability limited by training data	Fast	None	[243]
Learned post processing		Medium	Yes	Non-learned reconstruction + one step	Some: forward model can be varied	Depends on non-learned reconstruction	Generally none but can ensure data consistency[199].	[241, 44, 113, 119]
Learned Unrolled methods			Sometimes	Fixed number of iterations	Some: forward model can be varied	Fast	None	[10, 4, 3, 93]
Deep equilibrium methods				Iterative		Can be chosen by user	In the limit, converges to a fixed point	[18, 76]
Learned regularisation						Slow	Depends on choice of regulariser	[132, 186, 51]
	Semi-supervised: good and bad reconstruction				Some: depends on training data			[146, 158, 92]
	Unsupervised: example good reconstructions		No		High: trained without forward model			See chapter 4
Plug and Play methods							Convergence guarantees exist *e.g.* [194]	[223, 117, 169]

Table 2.1: Summary: Deep Learning and Inverse Problems

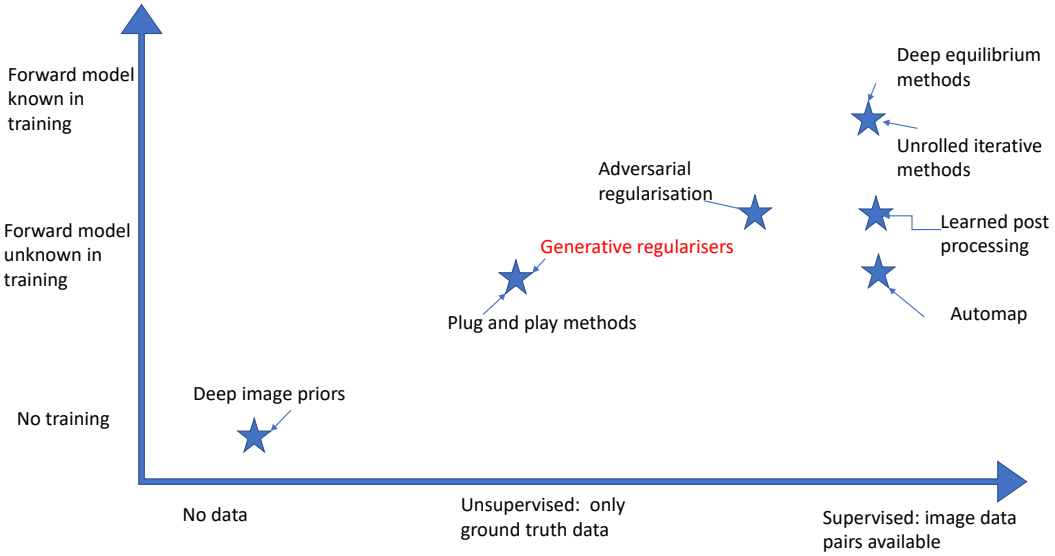


Figure 2-1: A graph comparing various deep learning approaches to inverse problems.

model during the deep neural network training. Some methods decouple the forward model and image reconstruction from the modelling of the image space with a deep neural network and others utilise information about the forward model in the deep neural network architecture. A summary of the analysis is also given in table 2.1. For other reviews consider [16, 159].

2.4.1 Untrained Methods

Deep Image Priors (DIP)[222, 219, 59] take an untrained convolutional neural network and use the weights of the neural network parameterise the image space *i.e.* for a fixed z, $x = f_\theta(z)$ maps the weights, θ, to images, x. Given some observed measurements, y, and a fixed z, the inverse problem can be reformulated as

$$\theta^* \in \arg\min_\theta \mathcal{D}(Af_\theta(z), y), \qquad x^* = f_{\theta^*}(z), \tag{2.13}$$

where $\mathcal{D}(\cdot,\cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is some loss function. The idea is that the network provides implicit regularisation as the convolutional networks favour 'natural images'. Exactly what 'natural' means is hard to define.

The success of DIP usually relies on further regularisation of (2.13). If the generator network is sufficiently wide and given sufficiently many iterative updates, then gradient descent will solve the non-convex optimisation problem in (2.13) to fit any signal, y, however noisy or corrupted [222], which is not desirable. Addi-

tional regularisation could be in the form of early stopping, regularisation on the network output, *e.g.* TV regularisation, or regularisation on the network weights [222]. In one particular example, Holler and Habring [90] adapt the DIP idea, restricting their reconstructions to be close in the TV norm to the output of an untrained network, with carefully defined mathematical properties.

DIP is also not quick to evaluate, as for each new image the weights of a neural network must be learned. This however makes the method very general, as each new image is reconstructed from scratch.

2.4.2 Supervised Methods: Paired Training Data

On the opposite end of the data spectrum, we consider a training set of paired data $\{x_i, y_i\}_{i=1}^N$, such that $y_i \approx Ax_i$ for all $i = 1, \dots n$ up to some noise in the observations. The learning challenge for an end-to-end method is now to learn $f_\theta : \mathcal{Y} \to \mathcal{X}$ which minimises the objective

$$\theta^* \in \arg\min_\theta \sum_{i=1}^{N} \mathcal{L}(f_\theta(y_i), x_i) \tag{2.14}$$

where $\mathcal{L}(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ is again some suitable loss function.

The most naive approach would be to ignore the forward model, A, define some deep architecture for f_θ going from space $\mathcal{Y}$ to $\mathcal{X}$, and optimise for θ. Although there have been some successes with this simple approach, *e.g.* AUTOMAP [243], this requires an extremely large amount of training data and incorporating domain knowledge from A can lead to improved results. Linking back to our aims in the introduction, incorporating expert knowledge, in the form of physical modelling for A, and encouraging solutions to be consistent with the measured data, can make approaches more trustworthy. In addition, if the forward model is included as an input to the network explicitly, then any small changes in the forward model can also be passed to the network and can make the learned network more generalisable.

Learned post processing Learned post-processing uses knowledge of the forward model to provide a rough reconstruction from measured data, before using a neural network to clean up the resulting image. For example, we decompose $f_\theta = h_\theta \circ A^\dagger$, for $A^\dagger : \mathcal{Y} \to X$ a rough inverse mapping the measurements back to the image space and $h_\theta : \mathcal{X} \to \mathcal{X}$ a neural network acting in the image space [241, 44, 113, 119]. The hope is that this method could provide the best of both worlds: the rough reconstruction could be a best-case unlearned method and the learned part could provide small improvements. In addition, the rough reconstruction maps back to the image domain, and the neural network, therefore,

acts only in the domain $\mathcal{X}$. This allows for the use of architectures like convolutional neural networks built into U-Nets[189], that are known to be successful on images. The speed of this approach depends on the rough reconstruction, the learned component is generally quick to evaluate.

This is approach is limited by the quality of the reconstruction. For example, if the rough reconstruction is unstable to noise in the observed data, then it may be difficult for the network to rectify this. The authors of [12] investigate [113] and note that these learned post-processing networks are unstable to small structural perturbations and also to any changes in the forward model sampling pattern. There is also some danger that the rough reconstruction could, in some way, destroy information, *e.g.* by providing an over-smoothed result. On the other hand, we note that, if the 'rough reconstruction' $A^\dagger y$ is consistent with the observed data, *i.e.* $AA^\dagger y = AA^\dagger Ax = Ax = y$, such is the case when $A^\dagger$ is the pseudo-inverse, then the additional post-processing can destroy this data consistency. One solution is to ensure that the neural network acts only in the null space of the forward operator [199, 198]. However, this places limitations on the network and the final solution will still depend on the quality of the initial reconstruction.

Learned unrolled methods Learned unrolled methods are architectures inspired by traditional iterative schemes. Consider that we wish to find $x^* \in \arg\min_{x\in\mathcal{X}}\{D(x) = \mathcal{D}(Ax, y)\}$. A common approach would be to consider gradient descent, with updates

$$x_{t+1} = x_t - \alpha_t \nabla D(x_t). \tag{2.15}$$

In the 'Learning to learn by gradient descent' paper [10], the authors replace the gradient step with a learned update rule

$$x_{t+1} = x_t - f_{\theta_t}(\nabla D(x_t)). \tag{2.16}$$

where f_{θ_t} is a network with parameters θ_t. In addition, the number of iterations is fixed and usually small. The parameters θ_t could be allowed to vary for each iteration number t or be fixed throughout the network.

The name 'unrolled' comes from this fixed number of iterations. By considering the initialisation of the optimisation scheme as an input and the result after k iterations the output, the iterative updates can be viewed as k modules of one large network. The method is trained like any other end-to-end method, choosing weights so that reconstructed and ground truth images match on a training dataset. Also, like other end-to-end approaches, the method is usually fast to evaluate once trained, and the small number of iterations is quick to evaluate.

There are many extensions to this work, for example, by providing the network with more inputs, such as information from previous updates, or through network choices and initialisation [4]. There is also the choice of other iterative schemes, other than basic gradient descent, that could be adapted. One very popular approach is Learned Primal-Dual Hybrid-Gradient (LPDHG)[3] where learned networks replace the proximal operators in the standard primal-dual hybrid-gradient algorithm.

Variational networks Variational networks [93] are an instance of an unrolled iterative scheme where gradient descent iterations for optimising a variational regularisation objective are unrolled. We include it in preparation for chapter 5. Consider the minimisation

$$\min_x \mathcal{R}(x) + \frac{\lambda}{2}\|Ax - y\|_2^2 \tag{2.17}$$

and note that optimisation by gradient descent will lead to updates of the form

$$x_{t+1} = x_t - \alpha_t \left(\lambda A^*(Ax_t - y) + \nabla\mathcal{R}(x_t)\right), \tag{2.18}$$

where A^* is the adjoint and α_t is a step-size. Replacing the $\nabla\mathcal{R}(x_t)$ term with a learned component, inspired by the Fields of Experts model [190], and again fixing iterations, gives an end-to-end method which includes information from the forward and its adjoint. For each iteration, the parameters of the regulariser, the step-sizes α_t and the regularisation parameter λ are learned. One pleasing feature is that if the network were to just output zeros, this leaves update $x_{t+1} = x_t - \alpha_t\lambda A^*(Ax_t - y)$, which is gradient descent for the least squares solution.

Deep equilibrium methods Note that even though the unrolled methods may be inspired by mathematically well-understood iterative optimisation techniques, the learned unrolled network does not necessarily inherit any of these properties. Deep equilibrium models [18, 76] are also inspired by iterative optimisation methods, but instead, consider iterating to convergence at a fixed point. Consider a network f_θ included as part of an iterative method

$$x_{t+1} = f_\theta(x_t; y). \tag{2.19}$$

Note here that f_θ is independent of t and the weights are the same with each iteration. The architecture of f_θ may depend on A, as in unrolled iterative methods or variational networks. For example, Deep Equilibrium Gradient Descent[76], takes

$$f_\theta(x; y) = x + \eta A^T(y - Ax) - \eta S_\theta(x) \tag{2.20}$$

for some trainable network S_θ. Note that by setting $S_\theta = \nabla\mathcal{R}$, (2.20) is identical in form to (2.18).

In comparison to the unrolled approaches, deep equilibrium methods consider iteration to convergence. The idea is that, if the limit, $x_t \to x^*$, exists, as $t \to \infty$, then it should be the solution to the inverse problem, $y = Ax^*$. The convergence aspect is straightforward and can be guaranteed for a suitable choice of f_θ, but the hard part is ensuring that the limit is a good solution to the inverse problem.

Consider that we wish to optimise $\sum_{i=1}^{N} \mathcal{L}(x_\infty(y_i, \theta), x_i)$, where $\mathcal{L}(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$. Taking gradients with respect to θ is not straightforward as it depends on $\frac{\partial x_\infty}{\partial \theta}$, and back-propagating through the potentially infinite number of iterations to reach the limit is impossible. Instead, note that the limit is a fixed point and so satisfies $x_\infty = f_\theta(x_\infty; y)$. It is now possible to differentiate both sides with respect to θ and rearrange to find $\frac{\partial x_\infty}{\partial \theta}$.

Deep equilibrium networks give an iterative method that will converge, hopefully to a good reconstruction of the inverse problem, although there are no guarantees for the latter. One benefit of the convergence result is that you expect stability with repeated applications of (2.19). The iterations should not become unstable. This allows the user to trade off computational budget and desired accuracy at test time. For example, you could use just a few iterations for a rough reconstruction or iterate to convergence to try and aim for the best possible prediction.

Learned regularisers - supervised Recall the variational regularisation framework (5.5) and how difficult it can be to choose a regulariser or choose regularisation parameters. One option is to set the regulariser to be a parameterised function, $\mathcal{R}_\theta(x)$, where θ could be just one or two parameters, such as a regularisation parameter, or paramaterise a more complex function. One could then learn θ in an end-to-end fashion, for example in a bi-level optimisation problem

$$\arg\min_\theta \sum_{i=1}^{N} \mathcal{L}(x^*(\theta, y_i), x_i) \text{ s.t. } x^*(\theta, y_i) \in \arg\min_x \mathcal{D}(Ax, y_i) + \lambda\mathcal{R}_\theta(x). \quad (2.21)$$

This is computationally challenging, especially as there is usually no analytic solution to the inner optimisation problem. See *e.g.* [132, 186, 51], for examples of learning parameters in the regularisation term.

2.4.3 Semi-supervised Methods: Unpaired Data

We consider a recent example of a semi-supervised method for inverse problems. Consider a dataset of ground truth images with some additional information, in

this case, some measured data, that need not be paired with the ground truth images.

Adversarial regularisation One possible interpretation of a regulariser is not just that the regularisation term is small for some desired property of the image, but that it is large for some set of undesired properties. For adversarial regularisation [146], the regulariser is chosen to maximise the Kantorovich formulation of the Wasserstein distance between the ground truth images and the poor reconstructions. This is very similar to the Wasserstein GAN formulation (see chapter 3) and indeed the regulariser plays the role of a discriminator, outputting small values for images similar to the ground truth training set, and large values for images produced from rough or poor reconstructions from the measured data dataset. The idea is that the critic should learn both an idea of the ground truth data distribution, and also common unwanted reconstruction artefacts. Once trained, the regulariser can be used in a variational regularisation scheme. Although inspired by the Wasserstein distance, there are no guarantees that learning the regulariser approximates the Wasserstein distance well, see *e.g.* [212].

The original paper has been extended to convex critics [158] and incorporated into an unrolled iterative method [157].

2.4.4 Unsupervised Methods: Ground Truth Training Data Only

Collecting and curating paired training data can be time-consuming and expensive. In addition, relying on paired data means that whenever something changes in the forward model, the methods trained in the previous regime either need to be retrained or need to be sufficiently general so that they can be applied to unseen data. Unsupervised methods, trained without a forward operator, are naturally more robust to forward model changes.

Plug and Play methods For a purely unsupervised approach, plug-and-play methods (PnP)[223] use denoisers trained on ground truth images and 'plug' them into iterative reconstructions for inverse problems. For example, consider the variational regularisation objective (2.4) and introduce a new variable $u \in \mathcal{X}$ to give the the constrained minimisation problem

$$x^* \in \arg\min_{x,u} \mathcal{D}(Ax, y) + \lambda \mathcal{R}(u) \qquad \text{s.t} \qquad x = u. \tag{2.22}$$

Reformulating this using an alternating direction method of multipliers (ADMM)

[29] approach, adding in another variable $v \in \mathcal{X}$, leads to an optimisation scheme

$$x_k \in \arg\min_x \left\{ \frac{1}{2\sigma^2}\mathcal{D}(Ax, y) + \frac{1}{2\eta}\|x - u_{k-1} + v_{k-1}\| \right\} \tag{2.23}$$

$$u_k \in \arg\min_x \left\{ \frac{1}{2\eta}\|x_k - v_{k-1} - x\|_2^2 + \lambda\mathcal{R}(x) \right\} \tag{2.24}$$

$$v_k = v_{k-1} + (x_k - u_k). \tag{2.25}$$

If the distance $\mathcal{D}$ is the 2-norm then (2.23) is solvable analytically. Equation (2.24) is the denoising of $x_k + v_{k-1}$ under the regulariser $\mathcal{R}$. This is the only place $\mathcal{R}$ appears and so instead of explicitly defining it, PnP methods implicitly regularise the optimisation by plugging in a 'good' denoiser to replace step (2.24). This denoiser acts on the image space $\mathcal{X}$ and requires no forward model knowledge.

There is a huge body of work on PnP methods, including on convergence guarantees [194], and on applications to other schemes other than ADMM *e.g.* fast iterative shrinkage/thresholding algorithm [117] or primal-dual splitting [169]. For a review paper focused on compressed sensing MRI, see [7].

Learned regularisers - unsupervised Learning a regulariser, or regularisation parameters, in an end-to-end fashion was covered in section 2.4.2. However, as set out in the introduction, the focus of this thesis is on learning the regulariser independently of the forward problem, using ground truth training data only.

Similar to PnP methods, a denoiser trained to remove added noise from ground truth data can be used as a regulariser for example $R(x) = \langle x, x - f_\theta(x)\rangle$ where $f_\theta : \mathcal{X} \to \mathcal{X}$ is a denoiser [188, 223, 143, 152]. Learning a basis or sparse coding of the ground truth dataset can also provide an avenue for regularisation [6]. For example, take a sparse basis $\{\phi_i\}$ spanning the space of ground truth data then $R(x) = \min_{\gamma,\phi} \left[\iota_{\{0\}}(\sum_i \gamma_i\phi_i - x) + \|\gamma\|_1\right]$ encourages reconstructions, x, to be sparse in this basis. *Generative regularisers*, the focus of this thesis, can be seen as a non-linear version of this. Instead of encoding the ground truth images as a set of weights and then reconstructing them by linearly combining basis elements we encode them in a latent space and reconstruct them using a non-linear function. A detailed overview of generative regularisers will be provided in subsequent chapters *e.g.* section 4.2.

Chapter 3

Generative Models

In this short chapter, we introduce generative models in a unified notation, including a variational autoencoder derivation that will be referred to in other chapters. We provide background on generators and generative models, focusing in particular on autoencoders (AEs), variational autoencoders (VAEs), generative adversarial networks (GANs), normalising flows and diffusion-based generative models. Note that this chapter is not meant to be exhaustive and other good review papers include [193, 129, 227].

3.1 Autoencoders

An AE has two parts, an encoder and a decoder. The encoder encodes an image in some latent space and the decoder takes a point in this latent space and decodes it, outputting an image. The lower dimensional latent space forms a 'bottleneck' that forces the network to learn representations of the input with reduced dimensionality and prevents the model from simply learning the identity mapping.

Take the encoder to be $E_\psi \colon \mathcal{X} \to \mathcal{Z}$ and decoder $G_\theta \colon \mathcal{Z} \to \mathcal{X}$, neural networks with parameters ψ and θ. The network is trained by minimising a reconstruction loss

$$\mathcal{J}(\psi, \theta) = \mathbb{E}_{x \sim P^*_{\mathcal{X}}} \left\| x - G_\theta(E_\psi(x)) \right\|_2^2 , \tag{3.1}$$

with respect to θ and ψ. Post training, the decoder can be used as a generator, taking values in $\mathcal{Z}$ and outputting points in $\mathcal{X}$.

The encoder is not required to cover a certain area of the latent space, or indeed restricted to a certain area, thus generating from points in the latent space may not lead to outputs similar to the training set. Furthermore, similar images

may not have a similar latent encoding. Nevertheless, this method of training is simple and has recently been used in learned singular valued decomposition and for applications in sparse view CT [24, 165].

3.2 Probabilistic Models

In the introduction, generative models were introduced as having three parts: a generator, a prior on the latent space, and some mechanism for inducing a distribution in the image space. As such, the AE doesn't fit this definition, as there are no distributions on the $\mathcal{X}$ or $\mathcal{Z}$ space. In order to move on to a probabilistic approach we first discuss some mathematical preliminaries.

Consider the image and latent spaces to be measurable spaces, $(\mathcal{X}, \mathcal{A}_X)$ and $(\mathcal{Z}, \mathcal{A}_Z)$, respectively. As $\mathcal{X}$ and $\mathcal{Z}$ are Euclidean spaces, we choose $\mathcal{A}_Z$ and $\mathcal{A}_X$ to be their Borel σ-algebras, the smallest σ-algebra containing every open set [66]. We consider a *prior probability distribution on the latent space*, $P_Z \colon \mathcal{A}_Z \to [0, +\infty]$ with corresponding probability density function p_Z (with respect to the Lebesgue measure). Recall the unknown *target probability distribution on the image space* $P^*_{\mathcal{X}} \colon \mathcal{A}_X \to [0, +\infty]$. Consider a measurable (e.g. continuous) *generator*, $G : \mathcal{Z} \to \mathcal{X}$.

We build a generated distribution by taking the prior distribution P_Z transformed by the generator, G, to give a generated distribution P_G on $\mathcal{X}$. Imposing a prior on the latent space sets out the locations in the latent space where we would like to be able to generate from. The generator, G, is chosen to minimise the distance between P_G and $P^*_{\mathcal{X}}$,

$$d(P^*_{\mathcal{X}}, P_G). \tag{3.2}$$

We now consider two different options for defining P_G using the generator G and the prior $P_{\mathcal{Z}}$. These correspond to the approaches taken by GANs and VAEs, respectively:

- Define $G_{\#}P_Z \colon \mathcal{A}_X \to [0, +\infty]$ to be the *push-forward distribution* of G applied to P_Z so that for any $B \in \mathcal{A}_X$, $G_{\#}P_Z(B) := P_Z(\{z \in \mathcal{Z} | G(z) \in B\})$. For f a real-valued $\mathcal{A}_X$-measurable function on $\mathcal{X}$ that $\mathbb{E}_{G_{\#}P_Z}[f] = \mathbb{E}_{P_Z}[f \circ G]$ [66, Proposition 2.6.8]. This gives generated distribution

  $$P_G := G_{\#}P_Z. \tag{3.3}$$

  There may not exist a corresponding density function with respect to the Lebesgue measure, as G may not be invertible, see for example [225, Theorem 12.7]. A random sample x from P_G can be obtained as $x = G(z)$, where z is a random sample from the prior P_Z.

- In some modelling regimes, it may be beneficial to be able to define a continuous density distribution for P_G and so instead of the push-forward operator we use a likelihood distribution $P(\mathcal{X}|\mathcal{Z};G)$ which depends on the generator, G. We would then define the *generated distribution* P_G to have density

$$p_G(x) = \int_{\mathcal{Z}} p(x|z;G)p_{\mathcal{Z}}(z)dz. \tag{3.4}$$

3.3 Generative Adversarial Networks

The choice of Wasserstein distance in (3.2) leads to the Wasserstein GAN, a popular generative model [13]. Taking $\Pi(P^*_{\mathcal{X}}, P_G)$ to be the set of joint probability distributions with marginals $P^*_{\mathcal{X}}$ and P_G, the Wasserstein distance is given by

$$\begin{aligned} \mathcal{W}(P^*_{\mathcal{X}}, P_G) &= \inf_{\nu\in\Pi(P^*_{\mathcal{X}},P_G)} \int \|x-\tilde{x}\|_2 \mathrm{d}\nu(x,\tilde{x}) &(3.5)\\ &= \inf_{\nu\in\Pi(P^*_{\mathcal{X}},P_G)} \mathbb{E}_{(x,\tilde{x})\sim\nu}\|x-\tilde{x}\|_2. &(3.6)\end{aligned}$$

Here we have used the 2-norm as a distance measure on $\mathcal{X}$, but other alternatives such as p-norms or Sobolev norms are possible [5].

Equation (3.6) is intractable as, for example, $P^*_{\mathcal{X}}$ is unknown. We instead rewrite (3.6) using the Kantorovich–Rubinstein dual characterisation (see Remark 6.5 [224])

$$\mathcal{W}(P^*_{\mathcal{X}}, P_G) = \sup_{D\in\mathrm{Lip}(\mathcal{X},\mathbb{R})} \left\{\mathbb{E}_{x\sim P^*_{\mathcal{X}}} D(x) - \mathbb{E}_{\tilde{x}\sim P_G} D(\tilde{x})\right\}, \tag{3.7}$$

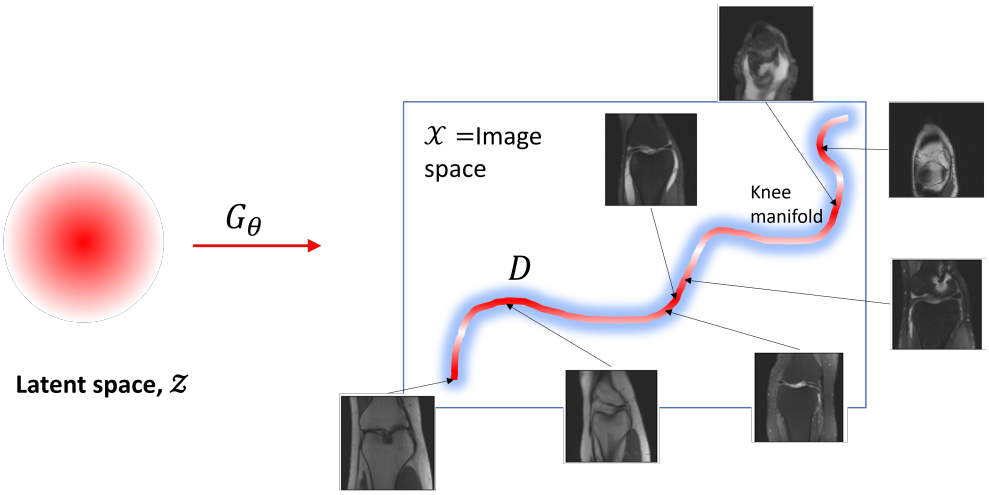
where $\mathrm{Lip}(\mathcal{X})$ denotes the space of real-valued 1-Lipschitz functions on $\mathcal{X}$. For any $D \in \mathrm{Lip}(\mathcal{X})$, for almost all $x \in \mathcal{X}$, the gradient $\nabla D(x)$ exists and $\|\nabla D(x)\|_2 \leq 1$. We call the function D a *discriminator*. Comparing this formulation to (3.6), (3.7) can now be approximated using samples from the two distributions, without knowledge of the full distribution.

Take G_θ and D_ϕ to be parameterised functions, e.g. neural networks, with parameters θ and ϕ, respectively. Using (3.3), the task of minimising the Wasserstein distance becomes

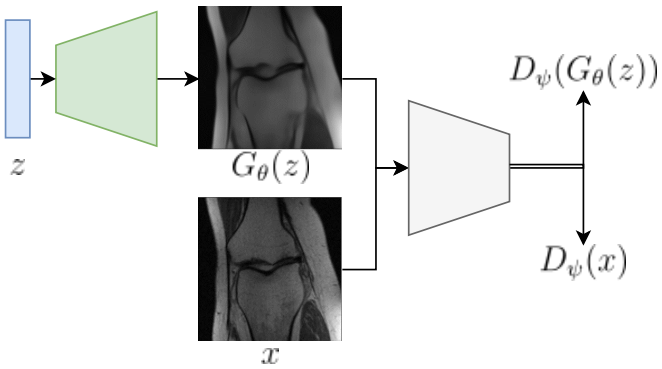
$$\min_\theta \max_\phi \left\{\mathcal{K}(\theta,\phi) \colon = \mathbb{E}_{x\sim P^*_{\mathcal{X}}} D_\phi(x) - \mathbb{E}_{z\sim P_z} D_\phi(G_\theta(z))\right\}. \tag{3.8}$$

To enforce the Lipschitz constraint a penalty term, $\mathbb{E}_{x\sim\hat{P}}\left(\|\nabla D_\phi(x)\|_2 - 1\right)^2$, is added to the loss function of the discriminator. Samples of $\hat{P}$ are calculated as $\hat{x} = tG_\theta(z) + (1-t)x$ for $t \sim \mathcal{U}[0,1]$, $x \sim P^*_{\mathcal{X}}$ and $z \sim P_Z$ [86].

The objective (3.8) is a saddle point problem. We say that (θ^*, ϕ^*) is a local saddle point if $\mathcal{K}(\theta^*,\phi) \leq \mathcal{K}(\theta^*,\phi^*) \leq \mathcal{K}(\theta,\phi^*)$ for all ϕ and θ in neighbourhoods of ϕ^*



(a) A GAN consists of a generator and a discriminator. The generator aims to take points in the latent space and output points along a lower dimensional of desired images. The discriminator tries to learn a decision boundary for this manifold. The generator is successful if the discriminator believes its generated images to be on this manifold. The discriminator is successful if it can label real example images as on the manifold and generate images as fake and not part of this manifold.



(b) Another pictorial representation of a GAN. Latent vectors, z are inputted into the generator to generate images and both real and generated images are tested by the discriminator.

Figure 3-1: Two pictorial representations of a GAN.

and θ^*, respectively. If $P_G = P^*_{\mathcal{X}}$, the desired result, then with $D_\phi \equiv$ constant, the result is a saddle point. However, there are many other, non-optimal critical points. For example, if P_G collapses down to just a Dirac delta distribution around a single image from $P^*_{\mathcal{X}}$ and again D_ϕ is constant, then this would also be a critical point. This example is often called *mode collapse*, when the generator just outputs a subset of the full training distribution.

See figure 3-1 for a pictorial view of a GAN. In the game-theoretic interpretation of (3.8) [81], a generative model is pitted against a discriminative model. The generator produces fake data and the discriminative model determines real from fake. The discriminator aims to accurately identify real images, maximising $\mathbb{E}_{x\sim P^*_{\mathcal{X}}} D_\phi(x)$, and generated (fake) images, maximising $-\mathbb{E}_{z\sim P_z} D_\phi(G_\theta(z))$. The discriminator forms a decision boundary around the manifold in 3-1(a). The generator tries to force the discriminator to label generated images as real, minimising $-\mathbb{E}_{z\sim P_z} D_\phi(G_\theta(z))$.

Although motivated by the Wasserstein distance, the quantity calculated by $\mathcal{K}(\theta, \phi)$ is unlikely to be an accurate approximation of the Wasserstein distance [13, 74, 212]. There are several reasons for this. Instead of optimising over all Lipschitz functions D in (3.6) we restricted to a family of functions parameterised by ϕ. The Lipschitz constraint is also not guaranteed to hold, enforced only by the addition of the gradient penalty. Additionally, (3.6) is calculated based on finite samples, not full distributions.

Other GAN variations Other choices of distance measure, in (3.2), may lead to other GAN variations. For example, choosing d in (3.2) to be an f-divergence leads to a range of GAN options [162]. Taking d to be the Jensen Shannon Divergence, an f-divergence, gives the original, 'vanilla', GAN [81]

$$\mathcal{K}(\theta, \phi) = \mathbb{E}_{z\sim P_Z}\left[\log(1 - D_\phi(G_\theta(z)))\right] + \mathbb{E}_{\tilde{x}\sim P^*_{\mathcal{X}}}\left[\log(D_\phi(\tilde{x}))\right]. \tag{3.9}$$

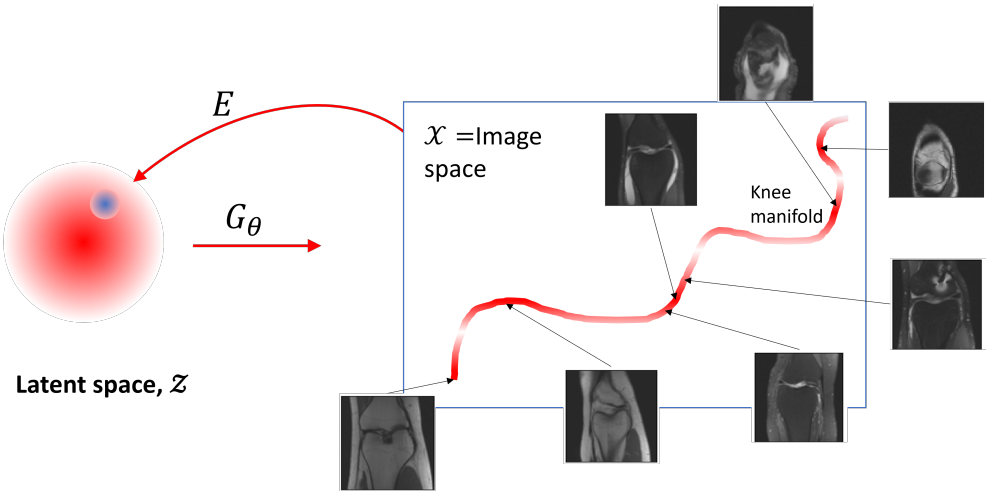
For definitions and details of the derivation see for example [227].

In the numerical experiments in chapter 4, we choose to use a Wasserstein GAN (3.8) as it is often more robust to a range of network designs and there is less evidence of mode collapse compared to the 'vanilla' GAN [13].

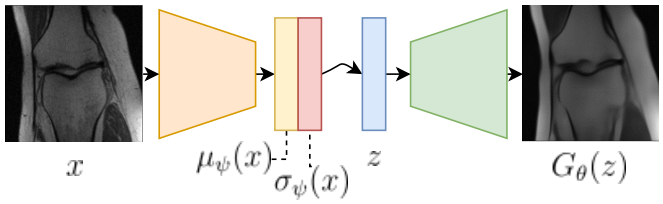
Note that there has been extensive research into GANs over the past few years to extend the models and to and this chapter is not meant to be exhaustive.

3.4 Variational Autoencoders

In the case of a VAE [125], we choose d, in (3.2) to be the Kullback–Leibler (KL) divergence and choose to induce a distribution P_G using (3.4).



(a) A VAE consists of two parts, an encoder (E here) and a generator. The encoder takes images and outputs a mean and a diagonal covariance of a normal distribution in the latent space, depicted in blue. The KL term in the VAE loss 3.23 encourages the encoded (blue) and prior distribution (red) to be similar. The generator takes samples from the latent space and outputs images.



(b) An image x is passed to the encoder, outputting mean $\mu_\psi(x)$ and diagonal variance $\sigma_\psi(x)$. This distribution is sampled to get a vector z which is then passed to a generator. The second term in (3.23) encourages input and output images to be similar.

Figure 3-2: Two pictorial representations of a standard VAE

For P, Q, probability distributions on $\mathcal{X}$, with probability density functions p and q, where P is absolutely continuous with respect to Q, the *KL divergence from Q to P* is defined as

$$d_{\mathrm{KL}}(P\|Q) = \mathbb{E}_{x\sim P} \log\left(\frac{p(x)}{q(x)}\right). \tag{3.10}$$

Parametrising the generator, G, with values θ, we take the KL divergence from P_{G_θ} to $P^*_{\mathcal{X}}$, and using (3.4), we get

$$\begin{aligned} d_{KL}(P^*_{\mathcal{X}}\|P_{G_\theta}) &= \mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log \frac{p^*_{\mathcal{X}}(x)}{p_{G_\theta}(x)} && (3.11)\\ &= \mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log p^*_{\mathcal{X}}(x) - \mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log p_{G_\theta}(x). && (3.12) \end{aligned}$$

The first term does not depend on θ and so minimising the KL divergence is equivalent to maximising $\mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log p_{G_\theta}(x)$ with respect to the parameters θ.

In most cases, the integral for $p_{G_\theta}(x)$, defined in (3.4), is intractable, except by expensive sampling. Using variational inference (note this is distinct to variational regularisation discussed in section 2.1.3) we introduce a new density distribution on $\mathcal{Z}$, $q(\cdot;\psi)$, from a family of functions parameterised by ψ. Starting with the definition of p_G in (3.4) we multiply and divide by q giving

$$p_{G_\theta}(x) = \int q(z;\psi)\frac{p(x|z;\theta)p_Z(z)}{q(z;\psi)}dz = \mathbb{E}_{z\sim q(\cdot;\psi)}\left[\frac{p(x|z;\theta)p_Z(z)}{q(z;\psi)}\right]. \tag{3.13}$$

Since the logarithm is concave, Jensen's inequality results in

$$\begin{aligned} \log p_{G_\theta}(x) &\geq \mathbb{E}_{z\sim q(\cdot;\psi)} \log\left[\frac{p(x|z;\theta)p_Z(z)}{q(z;\psi)}\right] && (3.14)\\ &= \mathbb{E}_{z\sim q(\cdot;\psi)}\left[\log p(x|z;\theta) - \log\left(\frac{q(z;\psi)}{p_Z(z)}\right)\right] && (3.15)\\ &= \mathbb{E}_{z\sim q(\cdot;\psi)}\left[\log p(x|z;\theta) - d_{KL}\left[q(\cdot;\psi)\|p_Z\right]\right]. && (3.16)\\ &:= -\mathcal{J}(\psi,\theta) \end{aligned}$$

For the KL divergence in the second term to be well defined, $q(z;\psi)$ must vanish whenever $p_Z(z) = 0$, but as $p_{\mathcal{Z}}$ is usually a Gaussian distribution, this is not a concern here. Equation (3.16) gives a lower bound for $\log p_{G_\theta}(x)$. By Baye's rule,

$$p(z|x,\theta) = \frac{p(x|z;\theta)p_Z(z)}{p_{G_\theta}(x)}, \tag{3.17}$$

and therefore we can make the lower bound more obvious:

$$-\mathcal{J}(\psi,\theta) = \mathbb{E}_{z\sim q(\cdot|x,\psi)} \log \frac{p(x|z;\theta)p_Z(z)}{q(z|\psi)} \tag{3.18}$$

$$= \mathbb{E}_{z\sim q(\cdot|x,\psi)} \log \frac{p_\theta(z|x)p_{G_\theta}(x)}{q(z|\psi)} \tag{3.19}$$

$$= \log p_{G_\theta}(x) - d_{KL}[q(z|\psi)\|p(z|x,\theta)]. \tag{3.20}$$

Line (3.20) along with the non-negativity of the KL divergence highlights again that $-\mathcal{J}(\psi,\theta)$ is a lower bound for $\log p_G(x|\theta)$. Moreover, one can see that this lower bound can be maximised when $q(z|\psi) = p(z|x,\theta)$.

The previous calculations were standard variational inference results. We now make choices particular to the VAE. Define the prior on $\mathcal{Z}$ to be a standard normal distribution. Introduce $\mu_\psi, \sigma^2_\psi \colon \mathcal{X} \to \mathcal{Z}$, neural networks parameterised by ψ, and hence let $q(\cdot;\psi)$ on $\mathcal{Z}$ be the Gaussian distribution

$$q(\cdot|x;\psi) = \mathcal{N}(\mu_\psi(x), \mathrm{diag}(\sigma^2_\psi(x))) := \mathcal{N}_{x,\psi}. \tag{3.21}$$

In addition to the assumption that q is Gaussian, we have also made the modelling choice that the distribution depends on x through neural networks. This choice is sometimes called 'amortised inference'. Also, let $p(x|z;\theta)$ be normally distributed with mean $G_\theta(z)$ and covariance matrix, Σ_θ,

$$p(\cdot|z;\theta) = \mathcal{N}(G_\theta(z), \Sigma_\theta). \tag{3.22}$$

Maximising the bound in (3.16) can now be written as a minimisation of

$$\mathcal{J}(\psi,\theta) = \mathbb{E}_{x\sim P^*_{\mathcal{X}}} \left(\mathbb{E}_{z\sim\mathcal{N}_{x,\psi}} \left[\log(|\Sigma_\theta|) + \frac{1}{2}(x - G_\theta(z))^T \Sigma_\theta^{-1}(x - G_\theta(z)) \right] \right.$$
$$\left. + d_{KL}\left(\mathcal{N}_{x,\psi}\|p_Z\right) \right) \tag{3.23}$$

ignoring constant terms.

In most cases, Σ_θ is taken to be some multiple of the identity

$$\Sigma_\theta = \rho^2 I \tag{3.24}$$

where ρ is either fixed or learned and it is this derivation that we use in chapter 4. However, in chapter 5 we consider a more general $\Sigma_\theta(z)$, where the covariance is learned, with weights θ, and dependent on the latent space vector, z.

The expectation over $P^*_{\mathcal{X}}$ is calculated empirically using the training set but the expectation over $\mathcal{N}_{x,\psi}$ is more complicated. To minimise $\mathcal{J}$ one might want to differentiate $\mathcal{J}$ with respect to ψ. Estimating gradients of the form $\nabla_\psi \mathbb{E}_{z\sim\mathcal{N}_{x,\psi}} f(z)$ is possible using Monte–Carlo methods but the estimator tends to exhibit high variance [173]. A significant contribution of the original VAE paper [125], is the so-called 'reparameterisation step'. Instead consider $z \sim \mathcal{N}_{x,\psi}$ implies $z = \mu_\psi(x) +$

$\epsilon\sigma_\psi(x)$ where $\epsilon \sim \mathcal{N}(0, I)$. Now $\nabla_\psi \mathbb{E}_{z\sim\mathcal{N}_{x,\psi}} f(z) \approx \frac{1}{L}\sum_{i=1}^{L} \nabla_\psi f(\mu_\psi(x) + \epsilon^{(i)}\sigma_\psi)$ where $\epsilon^{(i)} \sim \mathcal{N}(0, I)$. This approximation is experimentally seen to be more stable and values as small as $L = 1$ are regularly used.

For a pictorial interpretation of a VAE, see figure 3-2. We can interpret the two terms in (3.23) as a reconstruction term and a regularisation term, respectively. In the first term, an image is encoded to a distribution, the distribution is sampled from and the expected reconstruction and the original image are compared. See figure 3-2(a). Encoding to a distribution in the latent space, rather than just a single vector, enforces that points close to each other in the latent space should produce similar images. In the second term, the KL divergence encourages the encoded distributions to be close to standard normal. The $\mathcal{N}(0, I)$ prior also encourages independence between the latent dimensions. The balance between the two terms is determined by the noise level, ρ.

Note that the original KL objective, to maximise $\mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log p_{G_\theta}(x)$, describes maximising the probability, under the generated distribution, of samples from the training or desired distribution. In this objective, there is no requirement for the reciprocal relationship, that samples from the generated distribution should be likely under the desired distribution. Thus VAEs are often criticised for producing blurry or unrealistic images.

Updates in VAEs There have been several recent improvements in VAEs including vector quantised VAEs (VQVAEs) [170] that utilise a discrete latent representation. The VQ method is designed to prevent posterior collapse when the decoder starts outputting values independent of the latent space value z. In addition, Hierarchical VAEs [187, 205] consist of VAEs effectively stacked on top of each other. The idea is that this could improve the lower bond in (3.16) and decrease reconstruction error while learning a feature hierarchy for the images, similar to those learned by a convolutional neural network. In this thesis, chapters 5 and 6 also discuss variations on the standard VAE.

3.5 Invertible Neural Networks/Normalising Flows

Invertible neural networks or normalising flows [78, 174, 129, 108] train an invertible generator that forms a bijection between a latent space and the image space. Consider x as a transformation of a real vector z, under the transform $G_\theta : \mathcal{Z} \to \mathcal{X}$. The weights θ parameterise the transformation. Ensuring that G_θ is invertible and both G_θ and G_θ^{-1} are differentiable, we have that, using the change of variables formula,

$$p_{\mathcal{X}}(x;\theta) = p_{\mathcal{Z}}(G_\theta^{-1}(x))\|J_{G_\theta}(G_\theta^{-1}(x))\|^{-1} = p_{\mathcal{Z}}(G_\theta^{-1}(x))\|J_{G_\theta^{-1}}(x)\|^{-1} \tag{3.25}$$

where J_{G_θ} and $J_{G_\theta^{-1}}$ are the Jacobian of G_θ and G_θ^{-1}, respectively. Recall that if x (and z) is d dimensional then the Jacobian is a $d \times d$ matrix of partial derivatives. The formula can be interpreted as the probability of an image x is a multiplication of the probability of the latent vector that corresponds to the image x, $z = G_\theta^{-1}(x)$, and a scaling factor given by the determinant of Jacobian, that describes the change in volume moving from $p_\mathcal{Z}$ to $p_\mathcal{X}$.

Usually, z is sampled from an standard normal distribution and G_θ is split up into a series of transformations *e.g.* if $G_\theta = G_{k,\theta_k} \circ G_{k-1,\theta_{k-1}} \circ \cdots \circ G_{1,\theta_1}$, then $G_\theta^{-1} = G_{1,\theta_1}^{-1} \circ G_{2,\theta_2}^{-1} \circ \cdots \circ G_{k,\theta_k}^{-1}$ and $\det J_{G_\theta}(z) = \det J_{G_{k,\theta_k}}(G_{k-1,\theta_{k-1}}(\ldots(G_{1,\theta_1}(z)))) \cdot \cdots \cdot \det J_{G_{2,\theta_2}}(G_{1,\theta_1}(z)) \cdot \det J_{G_{1,\theta_1}}(z)$. Thus complex transformations can be built out of simpler transformations.

To choose parameters, θ, one possible choice of loss is the KL divergence between the target distribution $p^*_\mathcal{X}$ and the distribution generated by the invertible neural network

$$\mathcal{L}(\theta) = D_{KL}(p^*_\mathcal{X}(x) \| p_\mathcal{X}(x;\theta)). \tag{3.26}$$

In general, the distribution $p^*_\mathcal{X}$ is unknown, and so instead the (3.26) is calculated empirically using the change of variables formula (3.25):

$$\mathcal{L}(\theta) = -\mathbb{E}_{p*_\mathcal{X}(x)}\left[\log p_\mathcal{Z}(G_\theta^{-1}(x)) + \log|\det J_{G_\theta^{-1}}(x)|\right] + \text{const.} \tag{3.27}$$

The feed-forward and convolutional architectures discussed in section 2.3 are not necessarily invertible. There are several approaches for choosing invertible architectures, one is called the Non-linear Independent Components Estimation (NICE) [58] which partitions the input to a layer, z into vectors z_1 and z_2. The forward mapping sends z_1 to x_1 with an identity mapping and calculates $x_2 = z_2 + f_\theta(z_1)$ where f_θ can be any, non-invertible neural network. To calculate the inverse, $z_1 = x_1$ again remains unchanged and $z_2 = x_2 - f_\theta(x_1)$. The partition is free to change with each application of a bijective function and there is complete freedom in the choice of f_θ and this allows complex functions to be built up. Note that one benefit of this formulation is that the Jacobian is lower triangular and the determinant is easy to calculate.

3.6 Diffusion/Score Based Generative models

A generative model takes a known prior probability distribution and acts to transform it into the desired distribution. Continuous time Stochastic Differential Equations (SDEs) can smoothly transition between complex distributions and can be used as generative models. Much of this section is based on the derivations in [210].

Take $p^*_{\mathcal{X}}$ to be the desired data distribution and $p_{\mathcal{Z}}$ the known prior distribution over a latent space with the same dimension as the image space. Let $x(t)$, $0 \leq t \leq T$ be a diffusion process, define $p_t(x)$ be the probability density of $x(t)$, and take $p_0 = p^*_{\mathcal{X}}$ and $p_T = p_Z$. The diffusion process can be modelled as the solution to the SDE

$$dx = f(x,t)dt + g(t)dw \tag{3.28}$$

where w is the standard Brownian motion, $f(\cdot,t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued drift coefficient, and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is a scalar-valued diffusion coefficient. With suitable choices of f and g the SDE will diffuse the initial desired distribution, $p^*_{\mathcal{X}}$ into the fixed prior distribution, p_Z.

Taking images and transforming them into noise is not so useful. Instead, we wish to model the other direction, starting with samples from the prior and output images and so consider the reverse time SDE. This can be expressed analytically

$$dx = [f(x,t) - g(t)^2 \nabla_x p_t(x)]dt + g(t)d\bar{w} \tag{3.29}$$

where $\bar{w}$ is the reverse time standard Brownian motion. The term $\nabla_x p_t(x)$ is called a score function that can be approximated by a neural network [105, 209, 208]. This is where the learning comes in, and what makes this a deep learning-based approach. With an approximation of the score function, numerical SDE solvers can be used to generate new samples. As discussed in [210], different choices of f and g will give different SDE schemes, for example $f(x,t) = -\frac{1}{2}\beta(t)x$ and $g(t) = \sqrt{\beta(t)}$ where $0 < \beta(t) < 1$ [206, 101] or $f(x,t) = 0$ and $g(t) = \sqrt{\frac{d(\sigma^2(t))}{dt}}$ for σ an increasing function [208].

Sampling from the SDE can be slow. As discussed in [210], it is possible to define an ODE that induces the same marginal probability density as the SDE. This allows for fast sampling, with standard ODE solvers. This deterministic scenario also allows each image to be encoded to a point in the latent space and each latent space vector to generate an image. Conditional generation is also possible, in some cases without any retraining of the score function [210].

An alternative formulation to this score-based SDE approach is diffusion modelling. The above case where $f(x,t) = -\frac{1}{2}\beta(t)x$ and $g(t) = \sqrt{\beta(t)}$, can alternatively be formulated with a parameterised Markov chain trained using variational inference to produce samples matching the data after finite time [101]. In one direction, samples from the image space repeatedly have noise added until eventually, it reaches a sample from a standard normal distribution. The reverse direction reverses the diffusion process and allows for image generation.

3.7 Low Dimensional Manifold Assumption

AEs, VAEs and GANs work with the idea of a low dimensional manifold, such as depicted in figure 1-2. The newer diffusion-based models and invertible neural networks, choose their latent space $\mathcal{Z}$ to be the same as the image space. In the rest of this thesis, we will focus primarily on probabilistic models with a lower dimensional manifold assumption, such as VAEs and GANs. This is partly for reasons of timing. When we were scoping and planning for this thesis, VAEs and especially GANs were state-of-the-art and invertible neural networks and diffusion models were in their infancy and considered difficult to train for any meaningfully sized images. It is also because we hoped that the lower dimensional manifold would have a regularising effect. One concern, especially for use of invertible neural networks for inverse problems is whether they might add more instabilities to the problem, rather than mitigate the original ill-posedness. There has been some work looking at the instabilities of invertible neural networks [21]. Finally, is also worth noting that, especially in section 4, the training and application of the generative model are separate, so that any function G_θ can be included.

Chapter 4

Generative Regularisers

4.1 Introduction

Generally, ill-posed problems are solved by incorporating some prior information. As discussed in section 2.1.3 this is often given in the form of a regulariser in a variational regularisation framework [196, 107, 23], see equation (2.4). Hand-built regularisers are better suited to some types of images over others, *e.g.* TV is tailored to piece-wise smooth images. A natural question to ask is: given a set of images, which regulariser would work well?

There is a wide body of research into learning regularisers. Approaches include using a regulariser to force reconstructions to be sparse in some learned basis for feasible images [6]. Others have included a network trained for denoising [223, 152, 188] or removing artefacts [141, 164, 79], favouring images that are unchanged by the network. More recently, 'adversarial regularisation' [146] uses a neural network trained to discriminate between desired images and undesired images that contain artefacts. For a recent overview of approaches to using deep learning to solve inverse problems, see for example [16].

In this chapter, we consider the case where the regulariser depends on a learned generative model. We investigate regularisers [26, 57, 90, 218] that penalise values of $x \in \mathcal{X}$ that are far from the range of the generator, G, and call these *generative regularisers*. A popular example [26], revisited in section 4.2.1, limits solutions to those that are exactly in the range of the generator,

$$x^* = G(z^*), \qquad z^* \in \arg\min_z \|AG(z) - y\|_2^2 + \lambda\|z\|_2^2. \tag{4.1}$$

Generative regularisers combine the benefits of both a variational regularisation and a data-driven approach. The variational approach builds on the advancements in model-based inverse problems over the last century, while the data-

driven approach will provide more specific information than a hand-crafted regulariser. The method remains flexible as the machine learning element is unsupervised and therefore independent of the forward model and the noise type. In this chapter, we test different generative regularisers, inspired by the literature, on deconvolution, compressed sensing and tomography inverse problems. The success of generative regularisers will depend on the quality of the generator. We propose a set of criteria that would be beneficial for a generative model destined for use in inverse problems and demonstrate possible methods of testing generative models against these criteria.

4.2 Generative Regularisers for Inverse Problems

In this section, we bring together current approaches in the literature that penalise solutions of an inverse problem that are far from the range of the generator G. We consider variational regularisation (2.4) and regularisers of the form

$$\mathcal{R}_G(x) = \min_{z \in \mathcal{Z}} \mathcal{F}(G(z) - x) + \mathcal{R}_{\mathcal{Z}}(z) \tag{4.2}$$

where $\mathcal{F} : \mathcal{X} \to [0, \infty]$ and $\mathcal{R}_{\mathcal{Z}} : \mathcal{Z} \to [0, \infty]$. We consider different choices for $\mathcal{F}$.

4.2.1 Choices of $\mathcal{F}$

Restricting solutions to the range of the generator

The characteristic function of an arbitrary set $\mathcal{C}$ is defined as

$$\iota_{\mathcal{C}}(t) = \begin{cases} 0, & \text{for } t \in \mathcal{C} \\ 1, & \text{for } t \notin \mathcal{C} \end{cases}$$

Taking $\mathcal{F}(x) = \iota_{\{0\}}(x)$ and $\mathcal{R}_{\mathcal{Z}}(z) = \|z\|_2^2$ in (4.2) gives (4.1) and describes searching over the latent space for the encoding that best fits the data. Their choice $R_{\mathcal{Z}}(z)$ reflects the Gaussian prior placed on the latent space. Bora *et al.* [26] first proposed this strategy, applying it to compressed sensing problems. There are a number of interesting applications using this method, such as denoising [218], semantic manipulation [85], seismic waveform inversion [156], light field reconstruction [41], blind deconvolution [17] and phase retrieval [94]. Bora *et al.* [26] assume the existence of an optimisation scheme that can minimise (4.1) with small error and from this probabilistically bound the reconstruction error. However, the non-convexity introduced by the generator makes any theoretical

guarantees on the optimisation extremely difficult. Assuming the forward operator is a Gaussian matrix (the generator weights have independent and identically distributed Gaussian entries) and the layers of the generator are sufficiently expansive in size, there exist theoretical results on the success of gradient descent for optimising (4.1) [95, 138, 54].

This formulation can also be optimised by projected gradient descent [200, 109]:

$$\begin{aligned} w_{t+1} &= x_t - \eta A^T(Ax_t - y) \\ z_{t+1} &= \arg\min_z \|w_{t+1} - G(z)\|_2 \\ x_{t+1} &= G(z_{t+1}). \end{aligned} \tag{4.3}$$

With analogies to the restricted isometry property in compressed sensing [36], Shah and Hegde [200] introduce the *Set Restricted Eigenvalue Condition (S-REC)*. If the S-REC holds, then the operator A preserves the uniqueness of signals in the range of G. Theoretical work considers the case where A is a random Gaussian matrix, and shows, under some assumptions, it satisfies the S-REC with high probability. In addition, if the generator is an untrained network, then the projected gradient descent approach with sufficiently small step size converges to x^*, where $Ax^* = y$ [200, 109, 178].

Relaxing the Constraints

Returning to Bora *et al.* [26], the authors note that as they increase the number of compressed sensing measurements, the quality of the reconstruction levels off rather than continuing to improve. They hypothesise that this occurs when the ground truth is not in the range of the generator. One could consider relaxing the constraint that the solution is in the range of the generator, for example setting $\mathcal{F}(x) = ||x||_2^2$ allows for small deviations from the range of the generator. One could also encourage the deviations to be sparse, for example by taking $\mathcal{F}(x) = ||x||_1$ [57, 97]. Some theoretical considerations for this softly constrained approach is given in [79]. This approach is similar to the approaches of [141, 164] where they take $G \circ E : \mathcal{X} \to \mathcal{X}$ an encoder–decoder network and define $\mathcal{R}_G(x) = \|x - G(E(x))\|_2^2$. The idea is that this regulariser approximates the distance between x and the ideal data manifold. Less explicitly, there are a number of approaches that extend the range of the original generator, through optimisation of intermediate layers of the network [153, 53, 87] or tweaking the generative model training in response to observed data [160, 103].

4.2.2 Additional Regularisation

Additional regularisation on $\mathcal{Z}$ is given by $\mathcal{R}_\mathcal{Z}$ in (4.2). The most common choice is $\mathcal{R}_\mathcal{Z}(z) = \|z\|_2^2$ [26, 17] but there are other possibilities, for example

$\mathcal{R}_{\mathcal{Z}}(z) = \iota_{[-1,1]^d}(z)$ [218], where $d = \dim\mathcal{Z}$. Often, the regularisation matches the prior on the latent space used in generator training. Menon *et al.* [153] discuss that $\mathcal{R}_{\mathcal{Z}}(z) = \|z\|_2^2$ forces latent vectors towards the origin. However, most of the mass of the d-dimensional standard normal prior on their latent space is located near the surface of a sphere of radius d. Instead, they use a uniform prior on $d\mathcal{S}^{d-1}$. This idea has also been explored for interpolations in the latent space [229]. In addition, the prior on the latent space may not be a good model for the post-training subset of z that maps to feasible images. For a VAE there may be areas of the latent space that the generator has not seen in training and for a GAN, there could be mode collapse. A few recent papers consider how to find the post-training latent space distribution [52, 20].

Other regularisation choices could be based on features of the image, $x = G(z)$. For example VanVeen *et al.* [222] use $\mathcal{R}_{\mathcal{Z}}(z) = \mathrm{TV}(G(z))$. For a GAN generator, it is possible to take the regularisation term to be the same as the generator loss $\mathcal{R}_{\mathcal{Z}}(z) = \log(1 - D(G(z)))$. This regulariser utilises the discriminator, D, which has been trained to differentiate generated from real data. Examples include inpainting [235, 134] and reconstruction from an unknown forward model [11].

4.2.3 Other Approaches

There are a number of ideas that are linked to earlier discussions in this section but we will not cover them in detail. A major benefit of generative regularisers is the flexibility to changes in the forward model. We have therefore ignored conditional generative models [2, 176, 232, 148, 244, 167, 203] and those that train with a specific forward model in mind [141, 115, 89]. We also exclude work that uses an untrained neural network, for example, Deep Image Priors [222, 219, 59] or [90].

4.3 Regularisation Analysis

We briefly analyse here the `soft` regularisation functional given by $\mathcal{R}_G(x) = \min_z \|x - G(z)\|_2^2 + \mu\|z\|_2^2$, with $\mathcal{D}(Ax, y) = \|Ax - y\|_2^2$ in (2.4). We note that, in general, it will not satisfy the desired theoretical properties for variational regularisation as set out in section 2.1.3. Consider for example, $G : \mathbb{R} \to \mathbb{R}^2$ where $G(z) = (z, z^2)^T$. In this case, the generative model parameterises a lower dimensional manifold, a parabola in the larger 'image' space $\mathbb{R}^2$. Note that the range of the generative model is a non-convex set.

Consider that $A = I : \mathbb{R}^2 \to \mathbb{R}^2$. Consider that we observe data point $y = (\delta, 4)^T$. The solution to our regularised problem is now:

$$\min_{z,x} \left[L(x_1, x_2, z) := (x_1 - \delta)^2 + (x_2 - 4)^2 + \lambda(x_1 - z)^2 + \lambda(x_2 - z^2)^2 + \mu\lambda z^2 \right] \quad (4.4)$$

We have that

$$\frac{\partial L(x_1, x_2, z)}{\partial x_1} = 0 \iff x_1^* = \frac{\lambda z + \delta}{\lambda + 1} \tag{4.5}$$

$$\frac{\partial L(x_1, x_2, z)}{\partial x_2} = 0 \iff x_2^* = \frac{\lambda z^2 + 4}{\lambda + 1} \tag{4.6}$$

Substituting these gives

$$L\left(x_1^*, x_2^*, z\right) = \frac{\lambda}{(1+\lambda)}\left((z-\delta)^2 + (z^2-4)^2\right) + \lambda\mu z^2 \tag{4.7}$$

For the case $\delta = 0$

$$\frac{\partial}{\partial z} L\left(x_1^*, x_2^*, z\right) = 0 \iff z = 0 \text{ or } 2z^2 = 7 - (1+\lambda)\mu \tag{4.8}$$

and we have that $z = 0$ is a local maximum of L and, provided $(1+\lambda)\mu < 7$, we have two possible minimisers for L at

$$x_1 = \pm\frac{\lambda}{\lambda+1}\sqrt{\frac{7-(1+\lambda)\mu}{2}} \tag{4.9}$$

$$x_2 = \frac{\lambda(7-(1+\lambda)\mu)+8}{2(\lambda+1)} \tag{4.10}$$

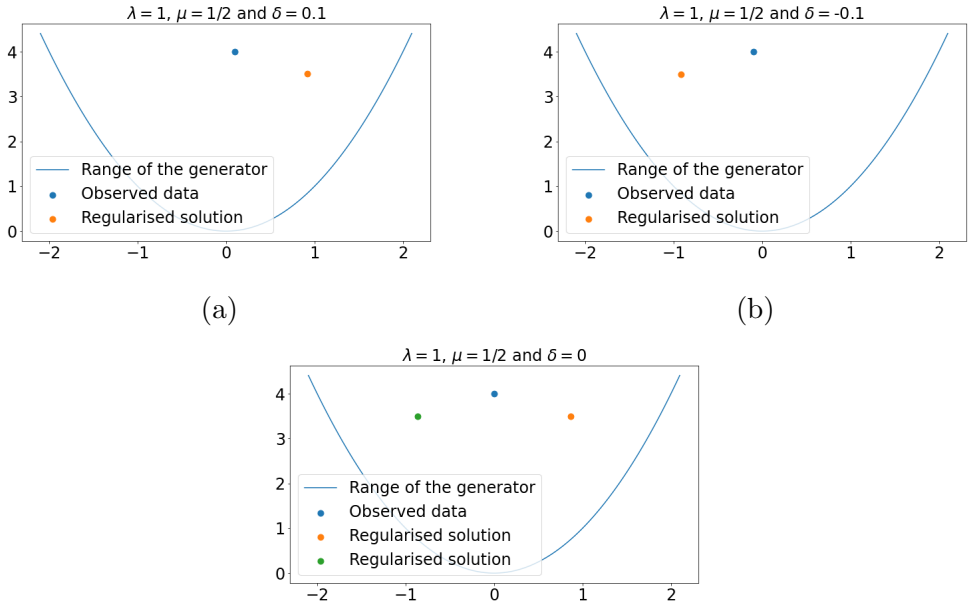
These both have the same value in the objective, L and hence there is not a unique minimiser for L.

Consider the case where $\lambda = 1, \mu = 1/2$ and now we vary δ. We have that

$$L\left(x_1^*, x_2^*, z\right) = \frac{1}{2}\left((z-\delta)^2 + (z^2-4)^2 + z^2\right) \tag{4.11}$$

which has two local minima sandwiched by local maximum as long as $|\delta| < 4$. As δ varies from small and positive to small and negative, the location of the global minima jumps from a positive x_1^* to a negative x_1^*. See figure 4-1 for a visualisation. Thus we do not have stability, the regularised solution does not depend continuously on the observed data.

We have seen that when minimising the objective L with respect to z, the non-convex nature of this minimisation and multiple local minima, give rise to situations where there are no unique global minima and/or the global minima can jump locations with just small perturbations in the observed data. We note that we demonstrated this for the case when $\mathcal{F}(x) = ||x||_2^2$ but there would be the same non-uniqueness issue for $\mathcal{F}(x) = \iota_{\{0\}}(x)$ or $\mathcal{F}(x) = ||x||_1$, for sufficiently small μ.



(a) (b)

(c)

Figure 4-1: Reconstruction results using 4.4 with $\lambda = 1$, $\mu = 1/2$ showing that this generative regulariser is not stable. The solution may not be unique, and small changes in the observed data, $(\delta, 4)$, may lead to large changes in the solution.

The possibility of a lack of stability is something to be aware of but does not mean that the approach cannot be useful or effective. Indeed, many of the approaches set out in chapter 2, do not meet these theoretical guarantees.

Future work could consider what properties of the generator might satisfy the properties of uniqueness, stability and convergence as set out in section 2.1.3. This counterexample uses a generator whose range is non-convex. If μ was zero and the range of the generator was convex then $\mathcal{R}_G(x) = \min_z \|G(z) - x\|_2^2$ would also be convex and this could lead to an avenue for theoretical consideration. Ensuring convexity in the range of the generator might require careful thought, for example, a linear generator would have a convex range, but would not be expressive enough for most of our needs.

We note that the augmented NETT approach [163], gets around the problem of multiple local minima in the minimisation for $\mathcal{R}_G(x)$ by removing the minimisation over z but instead using the encoder, which can be made continuous by construction. They are thus able to give a provable regularisation scheme. The work of [163] also considers the importance of a non-zero μ to ensure that their regulariser is coercive. In our case, take, for example, if $G = I$, the identity, then $\mathcal{R}_G(x) = \frac{\mu}{\mu+1}\|x\|_2^2$ which is coercive, only if $\mu \neq 0$. In general, where the first term of $\mathcal{R}_G(x)$ fails to be coercive, the second term might be able to compensate. Coercivity is important for the existence of minimisers and says that if $\|x\|_2^2 \to \infty$ then $\mathcal{R}_G(x) \to \infty$.

4.4 Generative Model Evaluation

Typically the aim of a generator has been to produce high fidelity images. However, the success of (4.2) relies not just on the ability of the generator to produce a few good images but to be able to produce every possible feasible image. In this section, we discuss desired properties for a generator trained for use in inverse problems and numerically explore methods to test these properties.

4.4.1 Desired Properties

To evaluate a generative model, in the context of inverse problems, we consider two overall aims which we will go on to further decompose:

A Samples from the generator are similar to those from the target distribution.

B Given a forward model and an observation, the image in the range of the generator that best fits the observation can be recovered using descent methods.

We split aim A into a set of properties:

A1 The generator should be able to produce every possible image in the target distribution. That is, for all $x \in \mathcal{X}$ such that x is similar to images in the training dataset, there exists $z \in \mathcal{Z}$ such that $G(z) = x$.

A2 The generator should not be able to produce images far from the target distribution. That is, for all $x \in \mathcal{X}$ such that x is not similar to images in the training dataset, then there does not exist $z \in \mathcal{Z}$ such that $G(z) = x$.

A1 includes that the generator should be robust to mode collapse and that the model should not trivially over-fit to the training data.

In the probabilistic case, with a prior over the latent space, property A becomes:

A That samples from the latent space, when mapped through the generator, will produce samples that approximate a target distribution. We should have that $d(P^*, P_G)$ is small for some distance measure d.

We also note that in the probabilistic case, A1 and A2 are not independent. By assigning probability mass to parts of the image space close to the target distribution, it is less likely that images far from the target distribution can be generated. In the probabilistic case, a third property is added:

A3 The generator should map high-probability vectors in the latent space distribution to high-probability images in the target distribution.

It is possible that A1 and A2 are satisfied but not A3. Note that these properties may not be possible to achieve for a given dataset.

We define two properties for Property B, these are

B1 The generator should be smooth with respect to the latent space, $\mathcal{Z}$.

B2 The area of the latent space, $\mathcal{Z}$, that corresponds to images similar to those in the training set should be known.

B1 ensures that gradient–based optimisation methods can be used. Continuity is also desirable: we wish that, in some way, points close together in the latent space should produce similar images. B2 considers that we need to have a distribution on or subset of $\mathcal{Z}$ to sample from in order to use the generator to sample images. This distribution may not necessarily be equal to any priors on the latent space used during training. We recognise that B1 and B2 are perhaps vague, and are not sufficient for Property B. It is an area for future work to consider making these statements precise enough to support theoretical work.

4.4.2 Generative Model Evaluation Methods

There is a wide range of existing generative model evaluation methods [28], focused mostly on Property A. We assume the availability of some test data drawn

from the same distribution as the training data and unseen by the generative model. The *average log likelihood* [81] of test data under the generated distribution is a natural objective to maximise. There is evidence, however, that the likelihood is generally unrelated to image quality and is difficult to approximate in higher dimensions [215]. To calculate a distance between generated and desired distributions, one possibility is the earth movers distance (EMD) [191], a discretised version of the Wasserstein distance. One could also encode the generated and unseen data in a lower dimensional space before taking distance calculations, for example by taking the outputs of one layer of any neural network trained for classification [99, 84]. A model that overfits the training data would perform perfectly in these distance measures. Also, the low dimensional representation used for the evaluation is likely to have the same inherent problems and drawbacks as the embedding learnt by the generative model. Similarly, a number of tests train an additional, separate, neural network discriminator to distinguish between test data and generated data [13, 145]. Failure to classify the two is a success of the generative model. For testing a GAN, the new discriminator is unlikely to be able to pick up failures that the original discriminator, used in training, missed. Finally, Arora *et al.* [15] estimate the size of the support of the generated distribution. A low support size would suggest mode collapse. Their technique depends on manually finding duplicate generated images which can be time-consuming and require expert knowledge.

Property B is less explored in the literature. One approach is to directly attempt to reconstruct test data by finding a latent space vector that when pushed through the generator, matches the data. With these found latent vectors, analysing their locations could check Property B2. To test the smoothness of the generator with respect to the latent space, Property B1, many previous papers, including the original GAN and VAE papers [81, 125], interpolate through the latent space, checking for smooth transitions in the generated images.

4.4.3 Numerical Experiments

In this section, we evaluate AE, VAE and GAN models against the desired properties given in section 4.4.1. We consider experiments on two datasets. Firstly, a custom-made `Shapes` dataset with 60,000 training and 10,000 tests 56×56 grey-scale images. Each image consists of a black background with a grey circle and rectangle of constant colour. The radius of the circle; the height and width of the rectangle; and the locations of the two shapes are sampled uniformly with ranges chosen such that the shapes do not overlap. This dataset is similar to the one used in [176]. Secondly, the `MNIST` dataset [136] consists of 28×28 grey-scale images of handwritten digits with a training set of 60,000 samples and a test set of 10,000 samples. For examples of both datasets, see the ground truth images in figure 4-3.

Architecture details are given in the appendix. We chose to use the same generator network for all three models, for comparison. Architecture choices were guided by [125, 86, 195]. All models have gone through a similar amount of optimisation of hyperparameters, including: the noise level ρ in the VAE decoder (3.24); the latent dimension; the number of layers; choice of convolution kernel size; drop out probability; leaky ReLU coefficient and learning rate. In order to select hyperparameters we manually inspected generated images. Models were built and trained using Tensorflow [83] in Python and made use of the Balena High-Performance Computing Service at the University of Bath. The models were trained using a single Dell PowerEdge C8220X node, with two Intel E5-2650 v2 CPUs, 64 GB DDR3-1866 MHz Memory and an Nvidia K20X GPU, 6 GB memory. The `MNIST` and `Shapes` VAE models take approximately 25 and 45 minutes to train, respectively.

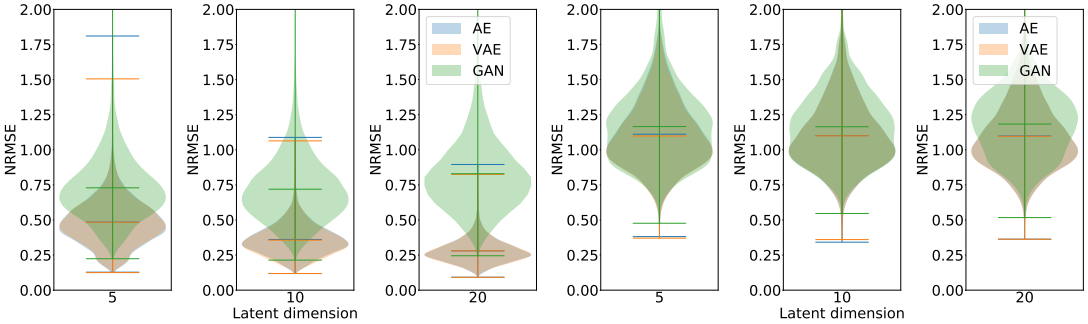
Reconstructing a Test Dataset

Property A1 asks that the generator is able to produce every image in the target distribution. Gradient descent with backtracking line search (Algorithm 1, in 2) is used to approximate

$$z^*(x) \in \arg\min_z \|G(z) - x\|_2^2, \tag{4.12}$$

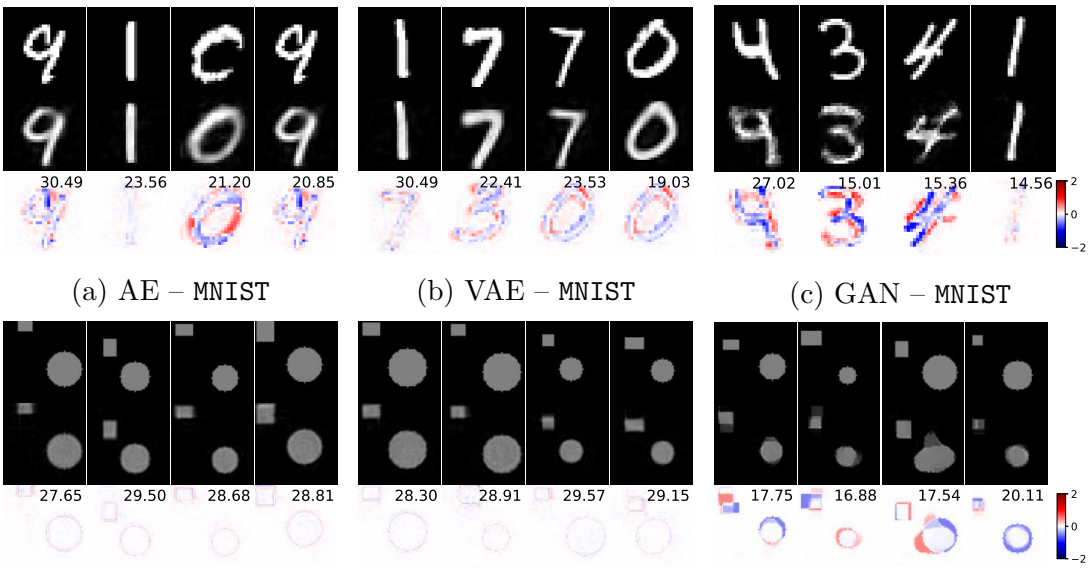
for each $x \in \mathcal{X}_{\text{test}}$, an unseen test dataset. For the AE and VAE, the algorithm is initialised at the (mean) encoding of the test image, $E_\psi(x)$ and $\mu_\psi(x)$, respectively. For the GAN, we take 4 different initialisations, drawn from a standard normal distribution, and take the best result. We find empirically, especially for the GAN, that different initialisations lead to different solutions.

Figure 4-2 shows $\|G(z^*(x)) - x\|_2/\|x\|_2$, the normalised root mean squared error (NRMSE) for reconstructions on `Shapes` and `MNIST` for the three different generator models. We see that, for both datasets, the AE and VAE have almost identical reconstruction results and the GAN results are comparatively worse. For the `Shapes` dataset the difference in results between the three generative models is less stark. In addition, NRMSE values are given for three different latent dimensions to show that the results are not sensitive to small changes in the latent dimension. Latent dimensions of 8 and 10 for `MNIST` and `Shapes`, respectively, are used in the rest of this paper. Figure 4-3 also shows reconstruction examples providing context to the results in figure 4-2. Numerical values on the image use the Peak-Signal-to-Noise-Ratio (PSNR, see definition 3.5 in [31]). The non-circular objects in the GAN results for `Shapes` could be a failure of the discriminator to detect circles.



(a) `MNIST` dataset (b) `Shapes` dataset

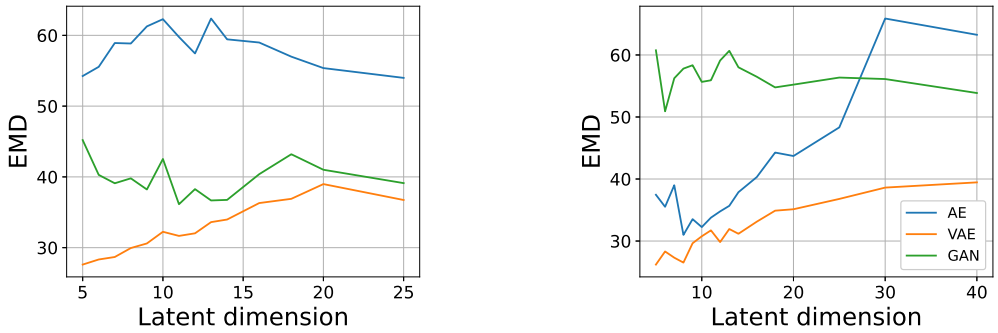
Figure 4-2: NRMSE between values of $G\left(\arg\min_z \|G(z) - x\|_2\right)$ and x and plotted as a histogram for all $x \in \mathcal{X}_{\text{test}}$. The horizontal lines show the median and range and the shaded area is a histogram. Note the brown colour is the result of the overlapping orange (VAE) and blue (AE).



(a) AE – `MNIST` (b) VAE – `MNIST` (c) GAN – `MNIST`

(d) AE – `Shapes` (e) VAE – `Shapes` (f) GAN – `Shapes`

Figure 4-3: Example reconstructions for the `MNIST` and `Shapes` dataset with eight ten-dimensional generative models respectively. In each sub-figure, the top row shows the ground truth, the second row the reconstruction and the third row the difference between the two.



(a) `MNIST` (b) `Shapes`

Figure 4-4: EMD between the test dataset and samples from a trained generator.

Distance Between P_G and P^*

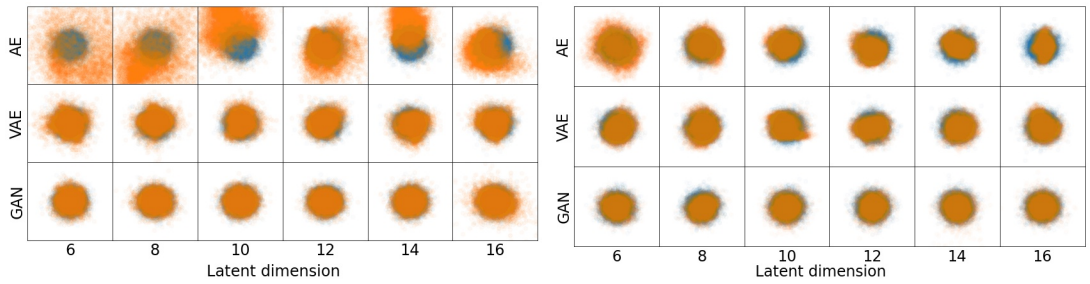
To investigate Property A3, the EMD [191] is calculated between empirical observations of the generated and the data distributions P_G and P^*. For the sets of test and generated images, $\mathcal{X}_{\text{test}} = \{x_1, \ldots, x_N\}$ and $\{G(z_1), \ldots, G(z_N) : z_i \sim \mathcal{N}(0, I)\}$, the EMD between their empirical distributions is defined as

$$\min_f \left\{ \sum_{i,j=1}^{N} f_{i,j} \|x_i - G(z_j)\|_2^2 : 0 \leq f_{i,j} \leq 1, \sum_{i=1}^{N} f_{i,j} = 1, \sum_{j=1}^{N} f_{i,j} = 1 \right\}. \quad (4.13)$$

The EMD is calculated using the Python Optimal Transport Library [72] with $N = 10,000$, the full test set. The results are given in figure 4-4. In both the `MNIST` and `Shapes` examples, the VAE has a lower EMD across the latent dimensions. The AE is added to this plot for comparison purposes but, as there is no prior on the latent space, $z_i \sim \mathcal{N}(0, I)$ may not be a suitable choice to sample from.

Visualisations of the Latent Space

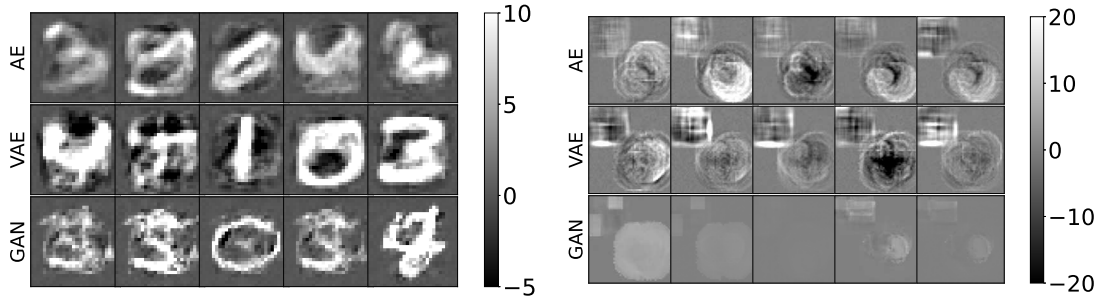
Property B2 requires that the area of the latent space that maps to feasible images is known. There is no prior on the latent space enforced for AEs and a $\mathcal{N}(0, I)$ prior is imposed for VAEs and GANs. In figure 4-5, gradient descent with backtracking (Algorithm 1 in chapter 2) is used to approximate (4.12), finding a latent vector $z^*(x)$ for each $x \in \mathcal{X}_{\text{test}}$. For comparison, the values $z^*(x)$ for the test set and 10,000 vectors drawn from a standard normal distribution are randomly projected into 2 dimensions. The encodings in the latent space match the prior $\mathcal{N}(0, I)$ for VAEs and GANs. For AEs, there are examples in lower latent dimensions, where the area covered by the encodings does not match a standard normal distribution.



(a) `MNIST` (b) `Shapes`

Figure 4-5: Comparisons of the latent space encodings of a test dataset with a standard normal distribution by projecting the vectors into 2 dimensions. Encodings of the test dataset are in orange and the standard normal vectors are in blue.

Generating Far from the Latent Distribution



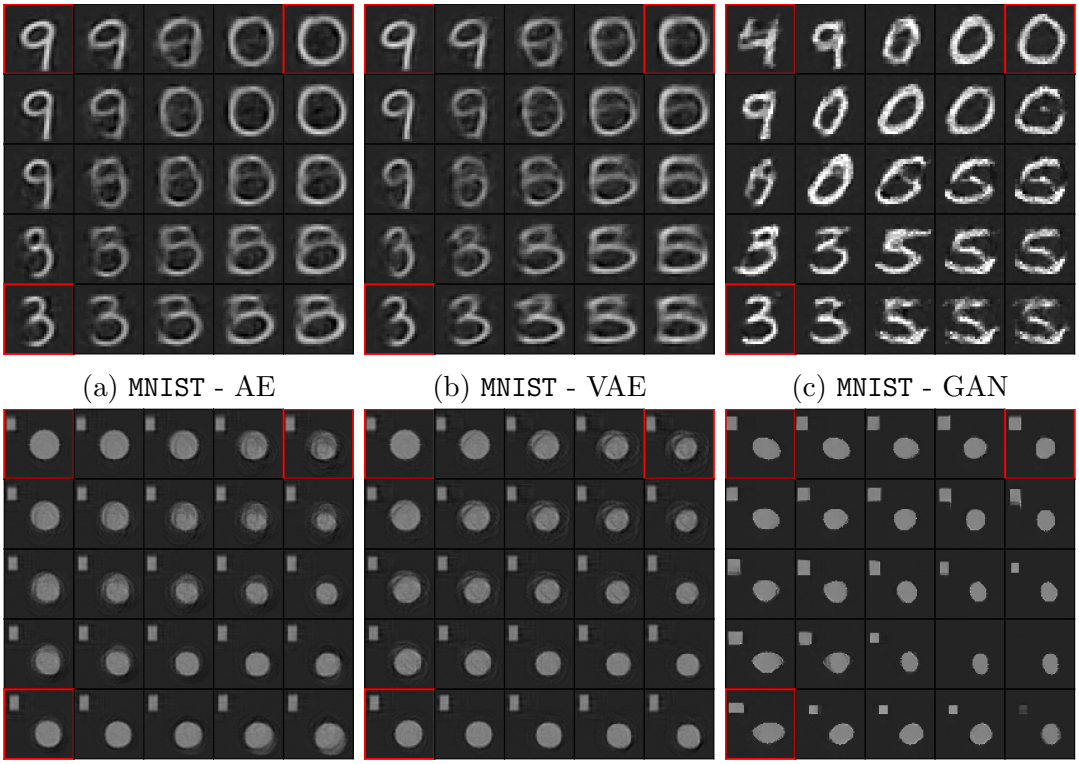
(a) `MNIST` (b) `Shapes`

Figure 4-6: Images generated far from the high-probability region of the prior distribution.

A known latent space gives known areas to sample from to produce new images. Figure 4-6 shows image examples generated far from a standard normal distribution. The images are not recognisable as similar to the training datasets. This emphasises the importance of Property B2, that the area of the latent space that corresponds to images similar to those in the training set should be known.

Interpolations in the Latent Space

We consider interpolating between points in the latent space, testing property B1. We hope to see smooth transitions between interpolated images, and that generated images are similar to those seen in training. We take three images from the test data, x_1, x_2 and x_3, find z_1, z_2 and z_3, their encodings in the latent space, using (4.12) and then plot interpolations $G(z_1 + \alpha_1(z_2 - z_1) + \alpha_2(z_3 - z_1))$



(a) `MNIST` - AE (b) `MNIST` - VAE (c) `MNIST` - GAN

(d) `Shapes` - AE (e) `Shapes` - VAE (f) `Shapes` - GAN

Figure 4-7: Interpolation ability of an AE, VAE and GAN. The highlighted top left, bottom left and top right latent space values were chosen close to the test dataset and the other images are computed via linear combinations in the latent space.

for $\alpha_1, \alpha_2 \in [0, 1]$. Figure 4-7 shows one example for each model and dataset for $\alpha_1, \alpha_2 = \{0, 0.25, 0.5, 0.75, 1\}$. In the AE and VAE, you see transitions that are smooth but blurry. The GAN images appear sharper but some outputs are not similar to training data examples, for example, in figure 4-7f there are a set of images that contain no rectangle. These images could be evidence of a discriminator failure: the discriminator has not yet learnt that these images are not similar to the training set.

Discussion

As expected, none of the three generator models, AE, VAE and GAN, fulfil Property A and B fully. For A, the GAN does poorly in the reconstruction results of figures 4-2 and 4-3. The lack of an encoder makes this more challenging. There is evidence of mode collapse, where parts of the training data are not well reconstructed and discriminator failure, where the images produced are not

realistic, see figures 4-3 and 4-7. The VAE does consistently better, demonstrated by the lower EMD between generated and test data in figure 4-4. The lack of prior on the AE, and thus a known area of the latent space to sample from, is a problem. Figure 4-6 demonstrates that sampling from the wrong area of the latent space gives poor results.

Pulling apart the cause of a failure to recover an image is difficult. It could be that the image is not in the range of the generator, a failure of Property A, or that the image is in the range of the generator but the image cannot be recovered using descent methods, a failure of Property B. For property B1, the mathematical properties of continuity or differentiability of a network, depending on the architecture. The interpolations in figure 4-7 show some evidence of large jumps between images in the GAN cases, but in general, the interpolations are reasonable. For both the GAN and the VAE, in figure 4-5, the encodings of the test images in the latent space seem to match the prior, Property B2.

4.5 Numerical Results for Inverse Problems

In this section, we apply AE, VAE and GAN models, evaluated in the previous section, on three inverse problems. Firstly *tomography*, the X-ray transform [39] with a parallel beam geometry. Secondly, *deconvolution* with a 5×5 Gaussian kernel. Lastly, *compressed sensing* where $y = Ax$ is an under-determined linear system where A is an $\mathbb{R}^{m \times d}$ Gaussian random matrix, $x \in \mathbb{R}^d$ is a vectorised image and $d \gg m$, see for example [55]. In each case, zero-mean Gaussian noise with standard deviation σ is added to the data. The forward operators were implemented using the operator discretisation library (ODL) [1] in Python, accessing scikit–learn [177] for the tomography back-end.

We consider variational regularisation methods in the form of (2.4) and (4.2) with $\mathcal{D}(Ax, y) = \|Ax - y\|_2^2$. To match the literature themes, we compare three different methods: hard, $\mathcal{F}(u) = \iota_{\{0\}}(u)$ and $\mathcal{R}_{\mathcal{Z}}(z) = \|z\|_2^2$; relaxed, $\mathcal{F}(u) = \|u\|_2^2$ and $\mathcal{R}_{\mathcal{Z}}(z) = \mu\|z\|_2^2$; and sparse, $\mathcal{F}(u) = \|u\|_1$ and $\mathcal{R}_{\mathcal{Z}}(z) = \mu\|z\|_2^2$, where μ is an additional regularisation parameter. We compare with regularisers independent of the generator: Tikhonov regularisation, $\mathcal{R}_G(x) = \|x\|_2^2$, for the convolution and tomography examples and TV regularisation [192], for the compressed sensing example.

The optimisation algorithms are given in the appendix. Hard and Tikhonov are optimised using gradient descent with backtracking line search, algorithm 1. TV regularisation is implemented using the Primal-Dual Hybrid Gradient method [40]. For relaxed, alternating gradient descent with backtracking is used, see algorithm 2. Finally, for sparse, the 1-norm is not smooth, and so Proximal Alternating Linearised Minimisation (PALM) [25] with backtracking, algorithm

3, is used to optimise the equivalent formulation $\min_{u\in\mathcal{X},z\in\mathcal{Z}} \|A(G(z)+u)-y\|_2^2+\lambda\left(\|u\|_1+\mu\|z\|_2^2\right)$. In all cases, initial values are chosen from a standard normal distribution.

4.5.1 Deconvolution

Figure 4-8 shows solutions to the deconvolution inverse problem with added Gaussian noise (standard deviation $\sigma = 0.1$) on the `MNIST` dataset. We test the relaxed, hard and sparse methods with a GAN against Tikhonov and try a range of regularisation parameters. Each reconstruction used the same realisation of noise affecting the data. Hard gives good PSNR results despite not reaching the Morozov discrepancy value, the expected value of the 2-norm of the added Gaussian noise [217]. For the hard constraints reconstructions are restricted to the range of the generator and we do not expect the data discrepancy to go to zero as λ decreases. In the relaxed and sparse constrained reconstructions, for smaller values of λ the solutions tend towards a least squares solution which fits the noise and is affected by the ill-posedness of the inverse problem. The additional variation in the choice of μ, as shown by the additional coloured dots, has little effect for the smaller values of λ.

Figure 4-9 again shows a deconvolution problem with added Gaussian noise (standard deviation $\sigma = 0.1$) on the `MNIST` dataset. We choose the hard reconstruction for the three different generator models and show three random initialisations in $\mathcal{Z}$. Regularisation parameters were chosen to maximise PSNR. The best results are given by the AE and the VAE. The GAN has failed to find a good value in the latent space to reconstruct the number three. The choice of initial value of z significantly affects the outcome of the reconstruction in the GAN case.

4.5.2 Compressed Sensing

Consider the compressed sensing inverse problem ($m = 150$ measurements) with added Gaussian noise (standard deviation $\sigma = 0.05$) on `MNIST` images. We choose regularisation parameters that optimise PSNR over 20 test images. Figure 4-10 includes a table with the PSNR results on an additional 100 test images. Due to the cartoon-like nature of the `MNIST` digits, TV regularisation is particularly suitable, however, VAE and AE hard and VAE relaxed are competitive with TV. For more context, example plots for the VAE and TV reconstructions are given in figure 4-10.

To give an indication of computational cost, Tikhonov reconstruction on the compressed sensing inverse problem on the `MNIST` dataset took on average 32 iterations of backtracking until the relative difference between iterates was less than 10^{-8}. In comparison, the hard and relaxed took on average 54 and 325 backtrack-

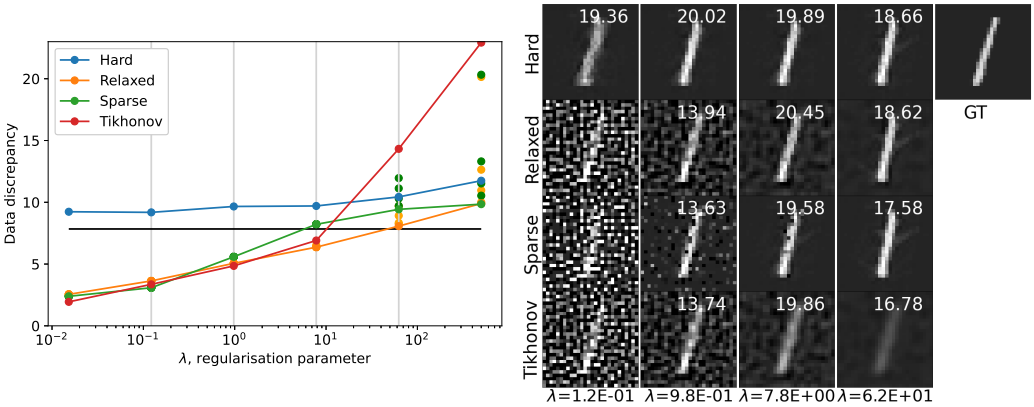
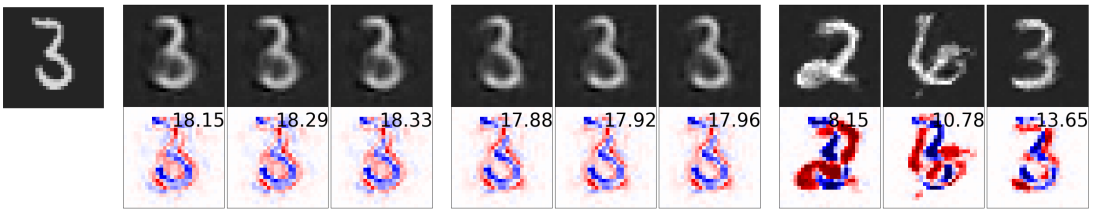


Figure 4-8: Solution of the deconvolution problem on `MNIST` with an eight-dimensional GAN. The plot shows the 2-norm reconstruction loss against regularisation parameter choice λ in comparison with the Morozov discrepancy value in black. Differing choices for μ are plotted as additional markers. The image plots correspond to the parameter values shown by the grey lines and include the PSNR values.



(a) GT (b) AE (c) VAE (d) GAN

Figure 4-9: Comparisons between the three generators, with eight-dimensional latent space, for the deconvolution problem. Reconstructions use the hard method. The plot shows 3 different initialisations for each generator. The ground truth (GT) is given on the left, the top line shows the reconstruction and the bottom line the residuals with the PSNR values.

Method	Generative Model			
	AE	VAE	GAN	None
Relaxed	19.31 ± 2.26	20.07 ± 1.66	17.12 ± 1.67	
Sparse	19.52 ± 2.72	21.08 ± 3.16	17.06 ± 2.5	
Hard	21.84 ± 4.09	**22.33 ± 4.17**	16.93 ± 2.57	
TV				21.54 ± 1.32

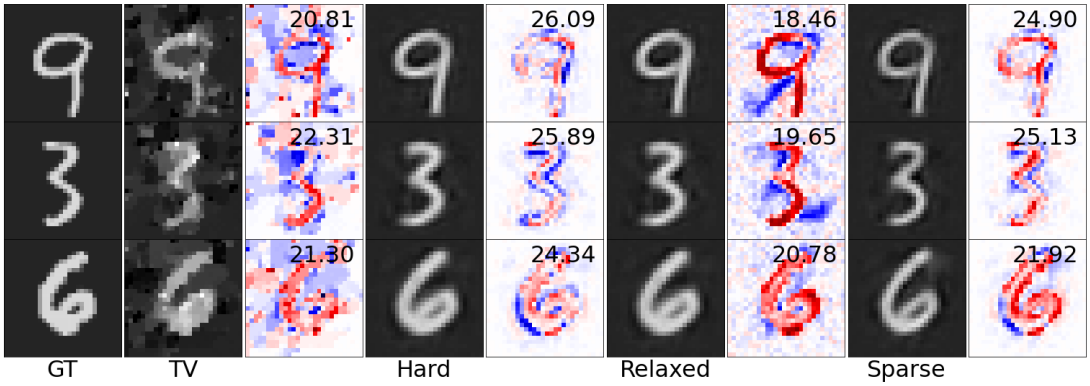


Figure 4-10: Results compare the three different regularisers and three different methods against the unlearned TV reconstruction on the compressed sensing inverse problem. The table shows the mean and standard deviations of PSNR values of 100 reconstructions. The plots show three example solutions, comparing the VAE reconstructions to TV reconstruction. The left column shows the ground truth, even columns the reconstructed images and odd columns the residuals with the PSNR values.

ing steps, respectively, without random restarts. The algorithms took up to 1 second for Tikhonov, 5 seconds for hard and 10 seconds for relaxed.

4.5.3 Tomography

Taking the tomography inverse problem with added Gaussian noise (standard deviation $\sigma = 0.1$), figure 4-11 includes a table which gives the average and standard deviation for the PSNR of 100 reconstructed `Shapes` images. The regularisation parameters were set to maximise the PSNR over a separate validation set of 20 test images. The GAN has a particularly poor performance but the AE and VAE results are all competitive with TV. Example reconstructions for the AE methods and TV reconstruction are given in figure 4-11. The generative regulariser gives a clear rectangle and circle while the TV reconstruction gives shapes with unclear outlines and blob-like artefacts. In terms of computational cost, Tikhonov took on average 157 iterations of backtracking until the relative difference between iterates was less than 10^{-8}. In comparison, the hard and relaxed took on average 37 and 255 iterations, respectively, without random restarts. The algorithms took

Method	Generative Model AE	VAE	GAN	None
Relaxed	30.70 ± 1.59	28.79 ± 0.71	24.20 ± 0.49	
Sparse	33.12 ± 0.80	33.09 ± 0.89	22.72 ± 1.80	
Hard	**33.17 ± 0.80**	32.92 ± 0.95	22.92 ± 2.10	
TV				29.94 ± 0.75

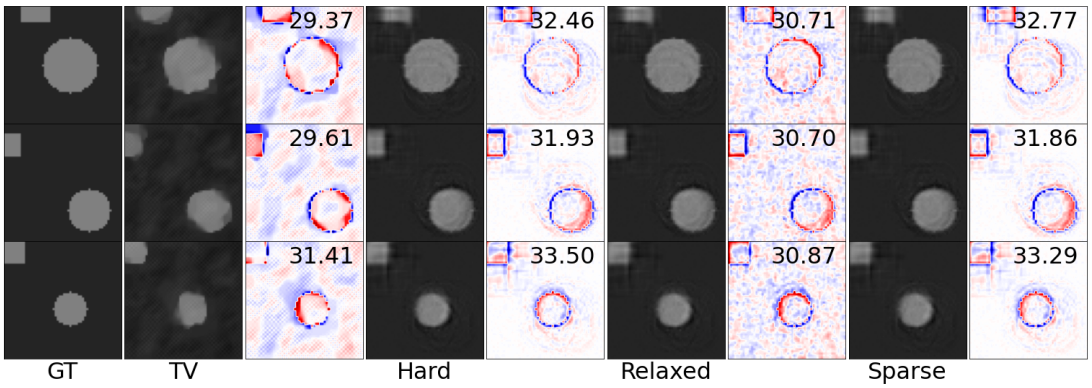


Figure 4-11: Results compare the three different regularisers and three different methods against the unlearned TV reconstruction on the tomography inverse problem. The table shows the mean and standard deviations of PSNR values of 100 reconstructions. The plots show five example reconstructions, comparing the AE reconstructions to TV reconstructions. The left column shows the ground truth, the even columns the reconstructions and the odd columns the residuals with the PSNR values.

up to 40 seconds for Tikhonov, up to 12 seconds for hard and up to 60 seconds for relaxed.

4.5.4 Out-of-Distribution Testing

We augment the `Shapes` dataset, creating a `shapes+` dataset, with the addition of a bright spot randomly located in the circle. We then take the Tomography inverse problem on the `shapes+` dataset with added Gaussian noise (standard deviation $\sigma = 0.05$). For a generative regulariser we use sparse, with $\mathcal{F}(x) = \|\nabla x\|_1$, the TV norm. Crucially, the VAE generator used was trained only on the standard `Shapes` dataset, without bright spots. We compare with standard TV reconstruction. The regularisation parameters were chosen to maximise the PSNR on 20 ground truth and reconstructed images. The mean PSNR over 100 test images for the sparse case is 32.83 with a standard deviation of 0.65 and for the TV reconstruction is 32.01 with a standard deviation of 0.67. Figure 4-12 show five reconstructions. The sparse deviations allow reconstruction of the bright spot demonstrating that generative regularisers can also be effective on

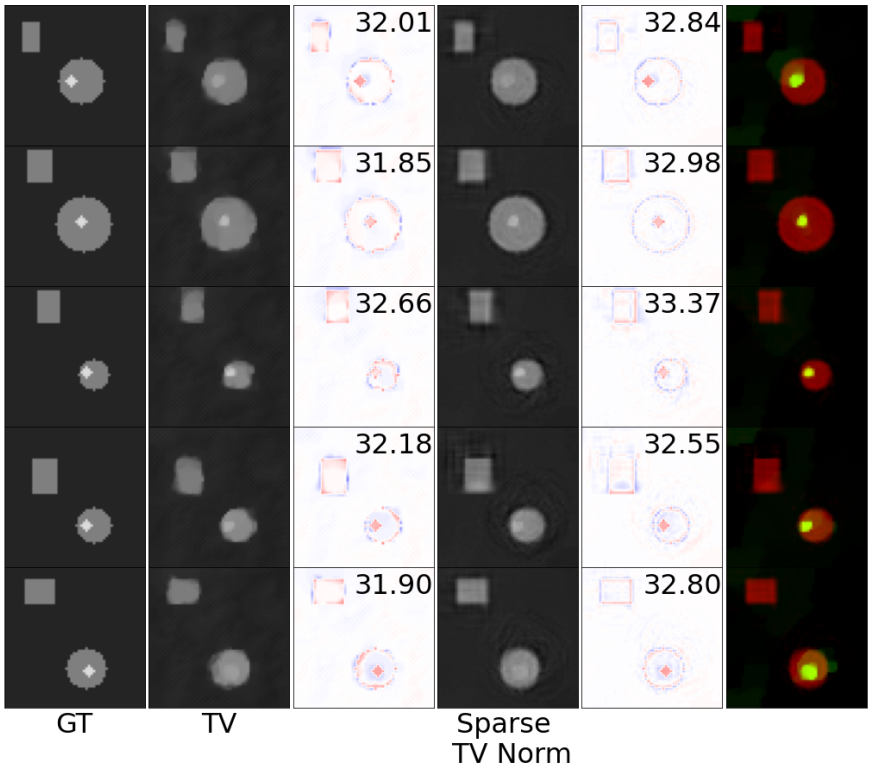


Figure 4-12: Tomography inverse problem on random images from the `shapes+` dataset. It compares the use of `sparse` method, where sparsity is measured in the TV-norm, with a standard TV reconstruction. The generator is a 10-dimensional VAE trained on `Shapes` images. In the final column, the part of the reconstruction lying in the range of the generator is coloured red and the sparse addition is green.

images close to, but not in, the training distribution.

4.5.5 `FastMRI` Dataset

We also train a VAE to produce knee `FastMRI` images. The VAE architecture is based on Narnhofer *et al*. The `FastMRI` knee dataset contains data 796 fully sampled knee MRI magnitude volumes [128, 238], without fat suppression. We extract 3,872 training and 800 test ground truth image slices from the dataset, selecting images from near the centre of the knee, resizing the images to 128×128 pixels and rescaling to the pixel range $[0, 1]$. The `FastMRI` VAE models took approximately 12 hours to train on the same system as above.

Results for the tomography inverse problem with added Gaussian noise of varying standard deviation are given in figure 4-13. For each image and noise level, the same noise instance is used for each reconstruction method, and additionally, for each method, a range of regularisation parameters are tested, and the reconstruction with the best PSNR value is chosen. This is to evaluate best achievable performance. The plot shows how the PSNR values, averaged over 50 test images, vary with the noise level. The relaxed method gives the best PSNR values, outperforming Tikhonov although with a larger variance. The sparse method curve has a similar shape to Tikhonov, but performs consistently worse suggesting that this choice of deviations from the generator is not suited to this dataset, generator or inverse problem. We see that for the hard method the results are consistent across the range of noise levels, not improving with reduced noise. The example images reflect the data with the hard reconstruction doing comparatively better with larger noise levels, but the relaxed method captures more of the fine details at the lower noise level.

4.6 Summary, Conclusions and Future Work

We considered the use of a generator, from a generative machine learning model, as part of the regulariser of an inverse problem, *generative regularisers*. Generative regularisers link the theoretically well-understood methods of variational regularisation with state-of-the-art machine learning models. The trained generator outputs data similar to training data and the regulariser restricts solutions (close) to the range of the generator. The cost of these generative regularisers is in the need for generative model training, the requirement for a large amount of training data and the difficulty of the resulting non-convex optimisation scheme. Weighing up the costs and benefits will depend on the inverse problem and the availability of data.

We compared three different types of generative regularisers which either restrict solutions to exactly the range of the generator or allow small or sparse deviations.

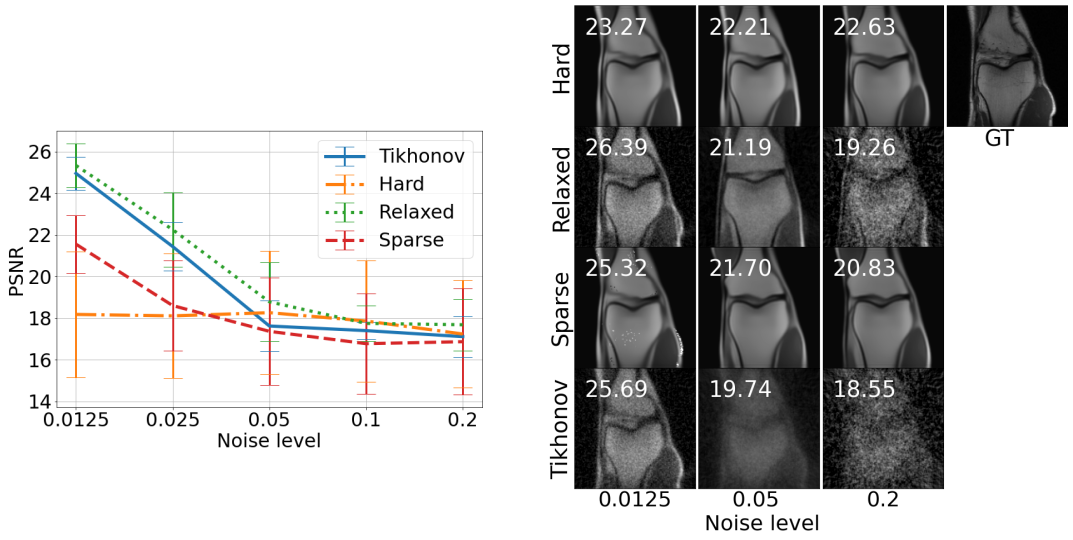


Figure 4-13: Reconstructions of the tomography inverse problem with additive Gaussian noise of varying standard deviations. Regularisation parameters are chosen to maximise the PSNR value for each image. The left shows the average PSNR values, taken over 50 test images, with the standard deviation in the error bars and the right shows one particular example reconstruction with PSNR values.

We found that in simpler datasets the restriction to the range of the generator was successful. Where the ground truth was more complex, then allowing small deviations produced the best results. A key benefit of generative regularisers over other deep learning approaches is that paired training data is not required, making the method flexible to changes in the forward problem. We demonstrated the use of generative regularisers on deconvolution and fully sampled tomography problems, both with gradually decaying singular values of the forward operator; and compressed sensing, with a large kernel and non-unique solutions.

The training of the generator is crucial to the success of generative regularisers, and a key contribution of this chapter is a set of desirable properties for a generator. Numerical tests linked to these properties were discussed and applied to three generative models: AEs, VAEs and GANs. None of these models fulfils the criteria completely. We observed known issues such as mode collapse and discriminator failure in the GAN, blurry images in the VAE and the lack of a prior in the AE. In the inverse problem experiments in this paper, the AE and the VAE yielded the most consistent results. The success of the AE, despite the lack of prior on the latent space, surprised us. We suspect the implicit regularisation on the model from the architecture and initialisations helped make the AE a usable generator. The GAN models did worst in the inverse problem examples: they generally seemed more sensitive to the initialisation of the non-convex optimisation, making the optimal point in the latent space difficult to recover.

4.6.1 Future Outlook

We identify three key areas for future growth in the field. Firstly, the desired criteria set out in section 4.4 for a generative model to be an effective regulariser are useful guidelines but they are not mathematically precise. The field of generative modelling is growing quickly and generators are improving every day. For safety-critical inverse problems, such as medical imaging, being able to quantitatively assess the quality of a given trained generator, would be hugely beneficial. Future work could consider refining these criteria into mathematically precise statements that could be used to quantitatively assess a given regulariser.

Secondly, we noted in section 4.3 that the `soft` method does not always lead to a convergent regularisation method in the strict mathematical definition. Future work could consider conditions on the generator needed to indeed yield a convergent regularisation method. Due to the non-linear nature of the generator G, we could also instead consider work from the non-linear inverse problems literature (*e.g.* [116]) which could be more able to deal with non-linear behaviour such as the sensitivity of the regularisers to different initialisations.

Finally, generative models are currently trained first and subsequently applied to an inverse problem. The benefit of this split approach is that the model does not

need retraining if there are changes in the forward problem. However, future work could consider how to train generative models with inverse problems in mind. For example, linking to the desired criteria, if B1 was refined to require Lipschitz continuity with a small coefficient, then with suitable choices of architecture and loss function, this could be ensured. Similarly, if convexity of the generator was required, then convex architectures could be considered. With a specific inverse problem in mind, one could also consider training the generator with a penalty for producing images with artefacts or choosing a loss function to encourage a smooth transition between images from the same subject.

4.A Generative Model Architectures

The architectures for the three different generative models, for the different datasets are given in this Appendix.

We first define a architecture for a convolution block, that consists of a convolution (or convolution transpose) with a given kernel size, stride length and number of filters, an activation function and a drop out layer. This pattern can be repeated to build up network expressivity.

Down-conv: $[f_1, f_2, \dots, f_l]$, $[s_1, s_2, \dots, s_l]$, $[k_1, k_2, \dots, k_l]$, activation, $[d_1, d_2, d_3]$	Convolution with f_1 filters, $k_1 \times k_1$ kernel, stride s_1, activation Dropout layer (prob=0.8) ⋮ Convolution with f_l filters, $k_l \times k_l$ kernel, stride s_l, activation Dropout layer (prob=0.8) Output size=$[d_1, d_2, d_3]$
Up-conv: $[f_1, f_2, \dots, f_l]$, $[s_1, s_2, \dots, s_l]$, $[k_1, k_2, \dots, k_l]$, activation, $[d_1, d_2, d_3]$	Convolution transpose with f_1 filters, $k_1 \times k_1$ kernel, stride s_1, activation Dropout layer (prob=0.8) ⋮ Convolution transpose with f_l filters, $k_l \times k_l$ kernel, stride s_l, activation Dropout layer (prob=0.8) Output size= $[d_1, d_2, d_3]$

Figure 4-14: Definitions used in Figure 4-15, 4-16 and 4-17.

For the `MNIST` dataset, the network is relatively simple with 3 convolution layers and a dense layer for the encoder/discriminator, and two dense layers followed by 3 convolution layers and a dense layer for the decoder/generator. The input image is size 28×28 and the latent dimension can vary, up to 25.

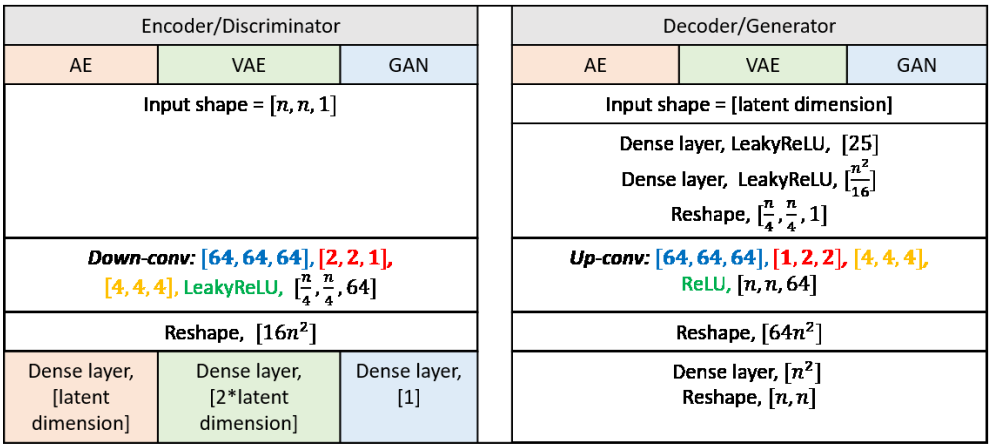


Figure 4-15: The architectures for the 3 generative models, AE, VAE and GAN, for the `MNIST` dataset. The convolution block definitions are given in figure 4-14.

For the `shapes` dataset, the network is very similar to the `MNIST` network, however the input images are size 56×56. The latent dimension can vary up to size 59.

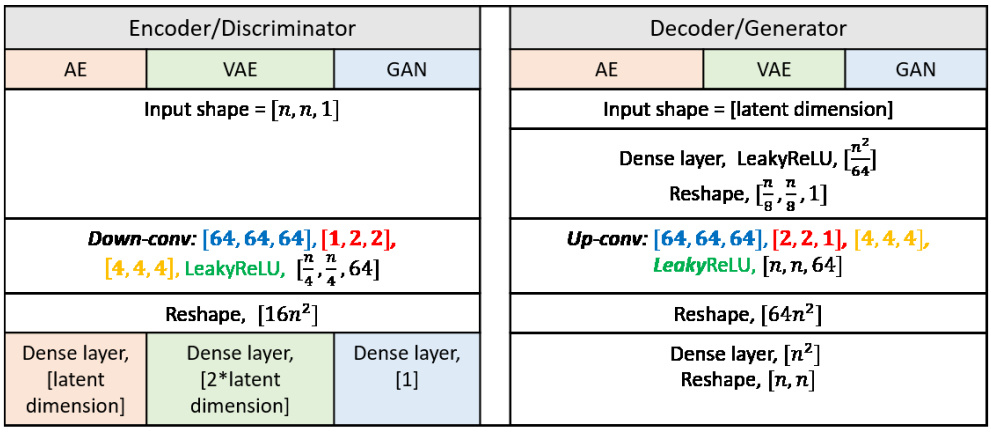


Figure 4-16: The architectures for the 3 generative models, AE, VAE and GAN, for the `Shapes` dataset. The convolution block definitions are given in figure 4-14.

For the `FastMRI` knee the network is is deeper and, to deal with the increased memory requirements of the 128×128 images, the dense layers are removed.

Encoder/Discriminator		Decoder/Generator
VAE		VAE
Input shape = $[n.n, 1]$		Input shape = [latent dimension]
		Dense layer, ReLU , $[\frac{n^2}{8}]$ Reshape, $[\frac{n}{16}, \frac{n}{16}, 16]$
Down-conv: $[8,16,32]$, $[1,1,1]$, $[3,3,3]$, LeakyReLU, $[n, n, 32]$ ***Down-conv:*** $[64,64,64]$, $[2,1,1]$, $[3,3,3]$, LeakyReLU, $[\frac{n}{2}, \frac{n}{2}, 64]$ ***Down-conv:*** $[128,128,128]$, $[2,1,1]$, $[3,3,3]$, LeakyReLU, $[\frac{n}{4}, \frac{n}{4}, 128]$ ***Down-conv:*** $[256,256,256]$, $[2,1,1]$, $[3,3,3]$, LeakyReLU, $[\frac{n}{8}, \frac{n}{8}, 256]$ ***Down-conv:*** $[64,32,8]$, $[1,1,1]$, $[3,3,3]$, LeakyReLU, $[\frac{n}{8}, \frac{n}{8}, 8]$		***Up-conv:*** $[32,64,128,256]$, $[1,1,1,1]$, $[3,3,3,3]$, ReLU, $[\frac{n}{16}, \frac{n}{16}, 256]$ ***Up-conv:*** $[512,512,512,512]$, $[1,1,1,1]$, $[3,3,3,3]$, ReLU, $[\frac{n}{16}, \frac{n}{16}, 512]$ ***Up-conv:*** $[256,256,256,256]$, $[2,1,1,1]$, $[3,3,3,3]$, ReLU, $[\frac{n}{8}, \frac{n}{8}, 256]$ ***Up-conv:*** $[128,128,128,128]$, $[2,1,1,1]$, $[3,3,3,3]$, ReLU, $[\frac{n}{4}, \frac{n}{4}, 128]$ ***Up-conv:*** $[64,64,64,64]$, $[2,1,1,1]$, $[3,3,3,3]$, ReLU, $[\frac{n}{2}, \frac{n}{2}, 64]$ ***Up-conv:*** $[32,16,8,4]$, $[2,1,1,1]$, $[3,3,3,3]$, ReLU, $[n, n, 8]$
Dense layer, [2*latent dimension]		***Up-conv:*** $[1]$, $[1]$, $[3]$, ReLU, $[n.n.1]$

Figure 4-17: The architectures for the `FastMRI` knee dataset VAE. The convolution block definitions are given in figure 4-14.

Chapter 5

Compressed Sensing MRI Reconstruction Regularised by VAEs with Structured Image Covariance

Learned regularisation for MRI reconstruction can provide complex data-driven priors to inverse problems while still retaining the control and insight of a variational regularisation method. Moreover, unsupervised learning, without paired training data, allows the learned regulariser to remain flexible to changes in the forward problem such as noise level, sampling pattern or coil sensitivities. One such approach uses generative models, trained on ground-truth images, as priors for inverse problems, penalising reconstructions far from images the generator can produce. In this chapter, we utilise variational autoencoders (VAEs) that generate not only an image but also a covariance uncertainty matrix for each image. The covariance can model changing uncertainty dependencies caused by structure in the image, such as edges or objects, and provides a new distance metric from the manifold of learned images. We demonstrate these novel generative regularisers on radially sub-sampled MRI knee measurements from the fastMRI dataset and compare to other unlearned, unsupervised and supervised methods. The results show that the proposed method is competitive with other state-of-the-art methods and behaves consistently with changing sampling patterns and noise levels.

5.1 Introduction

In this chapter, we choose to explore the inverse problem of compressed sensing Magnetic Resonance Imaging (MRI). A challenging inverse problem, that, if successful, provides the benefits of MRI with faster acquisition times. Sophisticated mathematical reconstruction techniques allow for high-quality images to be produced with just a subset of the full MRI measurements, which are quicker to obtain. This reduces costs but can also improve image quality by reducing motion artefacts. Mathematically, we seek to recover an image, $x \in \mathcal{X}$, from observed measurements, $y \in \mathcal{Y}$. The two are related by a linear forward model, $A : \mathcal{X} \to \mathcal{Y}$ as in (1.1). In MRI, A is composed of a Fourier transform and a sub-sampling operator that takes just a subset of the Fourier data. Measurements are incomplete and so multiple solutions may exist and the problem is ill-posed.

The ill-posedness can be mitigated by incorporating prior information; we consider this to be given in the form of a regulariser, $\mathcal{R}$, in a variational regularisation (5.5) framework [147, 127, 230].

In this chapter, we build on the previous chapter and the introduced *generative regularisers* [26, 57, 90, 218, 64] that consider the use of a generative machine learning model as part of the regulariser. As in the case of the previous chapter, the generative model is trained on high-quality data, obtained from fully sampled measurements, with no knowledge of the forward model, A, or any sub-sampled data. The idea of generative regularisers is to penalise inverse problem solutions that are dissimilar to the learned distribution on ground truth, or high-quality reconstructed, images.

As we saw in 3.4, the generative part of a VAE consists of a network, such that for any point $z \in \mathcal{Z}$, within the learned latent space, the network outputs a distribution on the image space $\mathcal{X}$,

$$p_{G,\Sigma}(x|z) = \mathcal{N}(x; G_\theta(z), \Sigma_\theta(z)) \tag{5.1}$$

for parameterised functions $G_\theta : \mathcal{Z} \to \mathcal{X}$ and $\Sigma_\theta : \mathcal{Z} \to \mathbb{R}^{d\times d}$, $d := \dim(\mathcal{X})$. In a standard VAE model, Σ_θ is taken to be a multiple of the identity matrix, *i.e.* $\Sigma_\theta(z) := \rho^2 I$, where the predicted variance ρ is either fixed or learned. This assumes that the reconstruction error of the generated image is independent and identically distributed for all pixels. In reality, parts of an image, such as the background, will be easy to model whereas sharp edges are harder to model accurately. Similarly, as images are commonly piece-wise smooth, there are often local correlations in the errors made by the model. A well-known issue with VAEs is their tendency to produce smooth images (see *e.g.* [193]), missing the sharp edges that exist in real data. In this chapter we consider the effect of a more expressive covariance matrix, Σ_θ, in a generative regulariser.

Contributions We propose a highly adaptive generative regulariser where edge and correlations in the data are modelled with a structured covariance network. We demonstrate the strength of this model by reconstructing knee MRI images from retrospectively sub-sampled real-valued data from the fastMRI dataset [238]. Contributions include:

- Adaption and tuning of the structured covariance model introduced by Dorta *et al.* [63] for the fastMRI dataset of knee MR images.
- An extension of the denoising example from Dorta *et al.* [63] for use in inverse problems with a non-trivial forward model, producing a novel generative regulariser.
- A demonstration of how the prior provided by the VAE with structured covariance can be explicitly visualised. We see that the structured covariance model has learned long-range correlations between pixels, taking into account image structure.
- An ablation study to compare three different options for the decoder covariance, $\Sigma_\theta(z)$: Σ is a fixed constant multiplying the identity matrix, Σ is a diagonal matrix with a learned diagonal and the proposed structured covariance where Σ is dense. We show the most flexible, dense covariance method, produces the best inverse problem results.
- Comparisons of our proposed regulariser with a variety of regularisers: the unlearned Total Variation (TV) [192] and least squares methods. Also with two unsupervised methods: a deep image prior approach with a pre-trained generator from Narnhofer *et al.* [160] and the original Compressed Sensing using Generative Models work by Bora *et al.* [26]. Finally, with a state-of-the-art, learned, end-to-end method, variational networks [93]. We demonstrate that our method is competitive with the state-of-the-art neural networks, yet offers superior generalisation to other noise statistics and sampling patterns.

5.2 Related Work

As discussed in chapter 2, deep learning approaches to image reconstruction in inverse problems is a growing field. For example, there are several supervised deep learning approaches [243, 93, 166, 226, 181, 104, 150] that require datasets of sub-sampled measurements paired with high-quality reconstructions. With any change in the forward model, *e.g.* a different k-space sampling pattern, or noise level, new data needs to be acquired and models retrained. Furthermore, with these methods, care needs to be taken to ensure the image reconstruction is consistent with the observed measurements and these methods can also be

unstable to small perturbations in the measured data [12]. In contrast, deep image priors [219, 160], seen in section 2.4.1, have no data requirements and instead use an untrained convolutional neural network as a prior for the inverse problem. The prior is implicit and comes from the architecture choices, another choice to make, but also requires regularisation in the form of early stopping to prevent over-fitting to the potentially noisy data.

We choose to take an unsupervised approach, where we have example ground truth images but no paired data. The image modelling, for training the regulariser, and the forward modelling for the inverse problem reconstruction, are completely decoupled. We model the images using a generative model incorporating it as part of a regulariser for the inverse problem reconstruction. There has been previous work in this area [26, 57, 90, 218, 64], discussed in chapter 4, and we extend it with the addition of a structured image covariance network.

We discuss a range of generative models in section 3. When choosing and training a generative model for use in inverse problems, the generative model must be able to produce a whole range of possible solutions. A common issue with GANs [81] is, that they do not generate across the range of images they were trained on. This failure can be subtle [15] and therefore difficult to identify. GANs also do not have an encoder, and finding a latent space value that corresponds to a particular image is a non-convex optimisation problem with multiple local minima. In comparison, a VAE is more suitable for use in generative regularisers because they are able to reconstruct images across the span of the training distribution although with the consequence of fewer high frequencies. We consider VAEs over other generative models such as normalising flows or invertible neural networks, both of which have been applied to inverse problems [110, 124], because of the regularising effect of a lower dimensional latent space in the VAE. Dorta *et al.* [63] proposed the use of a structured covariance as part of a VAE for denoising, and the novel addition of this work is the application to the non-trivial forward problem of MRI reconstruction.

5.3 Method

We build a probabilistic model for the reconstructed image, x, given an observation, y. First, consider the likelihood of the measurement, y, given image, x, denoted $p(y|x)$, and usually taken to be $\mathcal{N}(y; Ax, \gamma^2 I)$ for additive Gaussian noise over the observations with standard deviation, γ. We choose a prior on the images x given by a pre-trained generative model, $p_{G,\Sigma}(x|z)$ as in (5.1). Finally, let $p_{\mathcal{Z}}$ be the prior on the latent space used to train the generator, usually $\mathcal{N}(z; 0, I)$. Combining these parts, we have that

$$p(x, z|y) \propto p(y|x, z)\, p_{G,\Sigma}(x|z)\, p_{\mathcal{Z}}(z) \tag{5.2}$$

$$= p(y|x)\, p_{G,\Sigma}(x|z)\, p_{\mathcal{Z}}(z). \tag{5.3}$$

Ideally, we would seek to marginalise out the latent vector, z, however this integral is intractable, except by expensive sampling; instead, we take a maximum a posteriori (MAP) estimate. By taking logarithms, maximising (5.3) with respect to x and z is equivalent to variational regularisation (2.4) with

$$\mathcal{D}(Ax, y) = \frac{1}{2\gamma^2}\|Ax - y\|_2^2 \tag{5.4}$$

and

$$\mathcal{R}(x) = \min_z \left(\log(|\Sigma_\theta(z)|) + \frac{\|x - G_\theta(z)\|^2_{\Sigma_\theta(z)}}{2} + \frac{\|z\|_2^2}{2} \right), \tag{5.5}$$

where we denote the weighted norm by $\|x\|^2_M := x^T M^{-1} x$ and the determinant of a matrix Σ by $|\Sigma|$. Equation (5.4) ensures that x explains the observation y, while the second term in (5.5) constrains x to be close to images in the range of the generator. The first term in (5.5) is a log-determinant term and ensures that the second term in (5.5) is not made arbitrarily small by choosing a large variance across all pixel values. The final term in (5.5) matches the prior on the latent space and should encourage $G_\theta(z)$ to be similar to images seen during generative model training. We use $\mathcal{N}(z; 0, I)$ because it is easily accessible, however, it may not match the post training distribution in the latent space that maps to feasible inverse problem solutions.

VAEs with Structured Covariance The generative model is trained to minimise the distance between the generated, $p_{G,\Sigma}(\cdot;\theta)$, and training, $p^*_{\mathcal{X}}$, distribution. A VAE is derived by choosing the distance measure to be a Kullback–Leibler divergence and then maximising a lower bound approximation to this distance [125]. The intractable distribution over the latent space, $p_{G,\Sigma}(z|x;\theta)$, is approximated by an encoder

$$q(z|x;\psi) = \mathcal{N}(z; \mu_\psi(x),\ \mathrm{diag}(\sigma^2_\psi(x))) =: \mathcal{N}_{x,\psi}(z) \tag{5.6}$$

with neural networks μ_ψ, σ^2_ψ, parameterised with weights, ψ. Following the derivation in [125] or in section 3.4, training a VAE becomes a minimisation with respect to ψ and θ of

$$\mathbb{E}_{x\sim p_{\mathrm{Im}}} \mathbb{E}_{z\sim\mathcal{N}_{x,\psi}} \left(\mathrm{nll}(x, G_\theta(z), \Sigma_\theta(z)) + d_{KL}(\mathcal{N}_{x,\psi} \| p_{\mathcal{Z}}) \right) \tag{5.7}$$

where we write the negative log-likelihood as

$$\mathrm{nll}(x, G, \Sigma) = \log(|\Sigma|) + \frac{1}{2}\|x - G\|^2_\Sigma + \text{constants} \tag{5.8}$$

and the expectation over x is calculated empirically over the training set. The expectation over z is also calculated empirically and for more details see chapter 3.

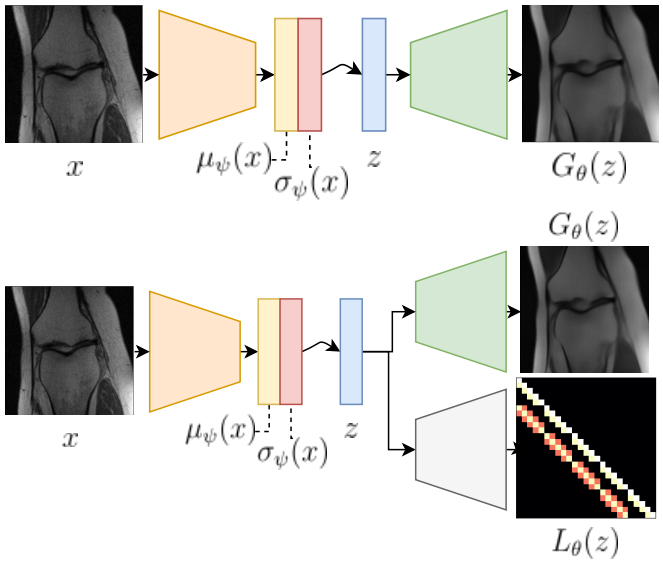


Figure 5-1: Comparison of the standard VAE (top) and the Structured Uncertainty Prediction Networks as developed by Dorta *et al.* [63] (bottom). The VAE has an encoder network that outputs a distribution, $\mathcal{N}(z; \mu_\psi(x),\ \mathrm{diag}(\sigma^2_\psi(x)))$, which is then sampled from to get a latent vector, z, which is passed to the generator network, $G_\theta(z)$. In addition to the usual generator, the proposed network includes a network that takes latent vectors, z, and outputs the weights of a sparse lower triangular matrix, $L_\theta(z)$. This matrix corresponds to the Cholesky decomposition of the inverse covariance for the generated image.

Structure of $\Sigma_\theta(z)$ Using a dense matrix Σ_θ is computationally infeasible for even moderate-sized images as it is expensive to calculate both the log determinant and the inverse required for the norm. We use the computationally efficient Structured Uncertainty Prediction Networks as developed by Dorta *et al.* [63]. For each point in the latent space, an additional decoder, parameterised by θ, outputs a sparse lower triangular matrix, $L_\theta(z)$, with the diagonal constrained to be positive. This forms the Cholesky decomposition of precision matrix, Σ_θ^{-1}, such that $\Sigma_\theta = (L_\theta L_\theta^T)^{-1}$. The Cholesky decomposition is taken to be sparse: $(L_\theta)_{i,j} = 0$ if pixels i, j are not in a small neighbourhood. This leads to a sparse precision matrix, Σ_θ^{-1}. A zero value at entry i and j in Σ_θ^{-1} means these pixels are independent, conditioned on all other image pixels. The pixels could still be correlated and thus this still allows for a dense covariance matrix. Indeed we would not choose to make Σ_θ sparse because zero values in the covariance indicate two uncorrelated pixels and we wish to allow complex dependencies across images. The local sparsity structure is also amenable to parallelisation on the GPU and for more details see [62]. A pictorial view of the full network including an example sparse Cholesky matrix is given in figure 5-1.

Objective functions This sparse Cholesky decomposition of the precision makes predicting a dense covariance matrix computationally feasible. The negative log-likelihood from (5.7) becomes

$$\text{nll}(x, G, \Sigma) = -2\sum_{i=1}^{d} \log(L_{ii}) + \frac{1}{2}\|L(x - G)\|_2^2 \tag{5.9}$$

up to constant terms and with, as above, $\Sigma^{-1} = LL^T$. The log determinant is now just a summation over the diagonals of the Cholesky matrix, we have removed the dense matrix inversion and there is no need to build the full covariance matrix. This also simplifies the regulariser from (5.5), giving

$$\mathcal{R}(x) = \min_z \left(\text{nll}(x, G_\theta(z), \Sigma_\theta(z)) + \frac{1}{2}\|z\|_2^2 \right). \tag{5.10}$$

Moreover, sampling from the extended VAE is possible by solving a sparse system of equations to get a sample $x = G_\theta(z) + (L_\theta(z)^T)^{-1}u$ where $u \sim \mathcal{N}(0, I)$.

We note that the minimisation problems in (5.7) and (2.4) with (5.10) are not well defined as they are not bounded from below. Pixel values that can be determined with high accuracy *e.g.* those of a consistent black background, can have extremely low variance and the log determinant term may become large and negative. We both bound the size of the diagonals of L_θ using a scaled tanh activation and added a very small amount of noise to the black background to deal with this. Investigating other priors on L_θ is an interesting area for future work.

5.4 Experiments

Dataset The NYU fastMRI knee dataset contains, amongst other data, 796 fully sampled knee MRI magnitude volumes [128, 238], without fat suppression. This data was acquired on one of three clinical 3T systems (Siemens Magnetom Skyra, Prisma and Biograph mMR) or one clinical 1.5T system, using a 15-channel knee coil array and conventional Cartesian 2D TSE protocol employed clinically at NYU School of Medicine. We use the ground truth data from the fastMRI single coil challenge, where the authors used emulated single-coil methodology to simulate single-coil data from the multi-coil acquisition. In addition, in the fastMRI dataset, the images are cropped to a square, centring the knee. We extract 3,872 training and 800 test ground truth images from this dataset, selecting images from near the centre of the knee, resize the images, using the Python imaging library [46] function with an anti-aliasing filter, to 128×128 pixels and linearly rescale each image to the range $[0, 1]$. The training and test datasets correspond to the training and validation sets of the fastMRI dataset and images from the same volume are always contained in the same dataset.

Forward Problem Our forward problem is inspired by the fastMRI single-coil reconstruction task; reconstructing images approximating the ground truth from under-sampled single-coil MR data [238]. The ground truth images are Fourier transformed and sub-sampled. To this end, a mask selects the points in k-space corresponding to a sampling pattern. We use both radial and Cartesian sampling patterns, selecting radial spokes and horizontal lines in k-space, respectively. We use the operator discretisation library (ODL) [1] in Python and take the same radial sub-sampling MRI operators as in [34]. Note that this is a relatively simple MRI model, yet a good starting point to test the feasibility of the proposed approach. We discuss its limitations in section 5.6.

Model Architecture See figure 5-1 for a comparison of the VAE and the VAE with structured image covariance. All the networks are built of resnet-style blocks which consist of a convolutional layer with stride 1, a resizing layer, followed by two more convolutional layers, and then a ReLU activation function. The output of this process is then added to a resized version of the original input to the block. The resizing layer is either a bilinear interpolation for an up-sampling layer, increasing width and height by a factor of 2; convolutions with stride 2 for a down-sampling layer or a convolution with stride 1 for a resnet block that maintains image size. We choose a latent space of size 100. The generative network consists of a single dense layer outputting 8x8 images with 16 channels before a resnet block without resizing to give 8x8 images with 256 channels, then four up-sampling blocks are applied giving image sizes 16x16, 32x32, 64x64 and 128x128 with channels 512, 256, 128 and 64 and final another resnet block without resizing to reduce the channels down to one output image. The covariance network is identical but outputs a 128x128 image with 5 channels, from which the sparse Cholesky matrix is formulated, based on code from Dorta *et al.* [63]. The sparsity pattern is chosen such that the Cholesky matrix $L_\theta(z)_{i,j}$ is non-zero only if pixels j is contained in a 5×5 patch, centred at pixel i. The encoder is reversed copy of the generator, with down-sampling layers replacing the up-sampling layers. We include drop-out layers during training.

Model Training Training the VAE with structured covariance is done in a two-stage process. First the generated mean, G_θ, and the encoder, $\mathcal{N}_{x,\psi}$, are trained with a standard VAE loss *i.e.* $\Sigma_\theta = \rho I$ where ρ is fixed [125]. The weights for the mean and encoder are then fixed before the covariance model is trained using (5.7). The choice of this two-stage training is two-fold: firstly, it forces as much information as possible to be stored in the weights of the mean, and not the covariances, and secondly, it allows us to compare the effect of just changing the covariance models in the ablation study. Models were built and trained in TensorFlow using an NVIDIA RTX 2080 - 8GB GPU. It took approximately 8 hours to train the means and then 24 hours and 30 hours for the diagonal and

structured covariance models.

Ablation Study We consider three variations on the structure of $\Sigma_\theta(z)$: Σ_θ is a fixed constant multiplying the identity matrix, Σ_θ is a diagonal matrix with a learned diagonal and our proposed method where Σ_θ is a dense matrix. We call these options: *mean+identity*, *mean+diagonal* and *mean+covar**. The *mean+identity* model is just taken to be the output of the first part of the *mean+covar** training described above. For *mean+diagonal*, we again take the learned generated mean, G_θ, and the encoder, $\mathcal{N}_{x,\psi}$ from the *mean+identity* model and then optimise (5.7) for the covariance network, but choose the off diagonals of the Cholesky matrix, $L_\theta(z)$, to be zero, so that the final covariance matrix is diagonal.

Proposed reconstruction method For the proposed method *mean+covar** and the versions *mean+identity* and *mean+diagonal* we choose a variation on the above regulariser (5.5)

$$\mathcal{R}(x) = \min_z \lambda \left(\text{nll}(x, G_\theta(z), \Sigma_\theta(z)) + \frac{\mu}{2}\|z\|_2^2 \right). \tag{5.11}$$

The addition of the two regularisation parameters λ and μ is in recognition that this is a non-convex problem and that our modelling assumptions are imperfect, for example, the prior on $\mathcal{Z}$ may not match the posterior after VAE training. We use alternating gradient descent with backtracking line search, see algorithm 2, where gradient descent steps are taken, alternating in the x and z space, with step size chosen to insure the objective decreases. We initialise at a rough reconstruction, given by the adjoint of the forward operator, and the encoding of the adjoint for x and z, respectively. Regularisation parameters were selected via a grid search to maximise the peak signal–to–noise ratio (PSNR). This is to demonstrate the best achievable results. An alternative approach would be to use a validation set to set the regularisation parameters but this is an approach for future work.

Unlearned method comparisons We compare against Total Variation (TV) regularisation [192] implemented using the Primal-Dual Hybrid Gradient (PDHG) method [40], with regularisation parameter chosen to maximise PSNR. As a baseline, we also calculate the least squares solution, $\min_x \|Ax - y\|_2^2$, optimised using gradient descent with backtracking line search and regularised with early stopping. The stopping point is optimised to give the best PSNR value for each image, the best achievable result.

Data driven prior comparisons For another example of a generative regulariser, we take our learned mean generator G_θ and implement the method of

Bora *et al.* [26], minimising with respect to z the objective

$$L(z) = \frac{1}{2}\|AG(z) - y\|_2^2 + \mu\|z\|_2^2, \tag{5.12}$$

searching for an image in the *range* of the generator that best matches the measurements. The regularisation parameter, μ, is chosen to maximise PSNR. We also implement the approach of [160] which takes a trained generator but then, after observing data, tweaks the weights of the network, optimizing $z^*, \theta^* \in \arg\min \|AG_\theta(z) - y\|_2^2$, we call this *Narnhofer19*. We use the same mean generator as in our other experiments, gradient descent with backtracking for the z optimisation and TensorFlow's inbuilt Adam optimiser for the network weight update, θ. We choose 1000 iterations to find an initial value of z and then select the number of iterations for the alternating z and θ updates via a grid search to maximise the PSNR to give the best achievable result.

Supervised end-to-end method comparison Finally, we also demonstrate comparisons with a variational network (VN) [93]. This is a end-to-end approach which takes (5.5) with $d(Ax, y) = \frac{\lambda}{2}\|Ax - y\|_2^2$ and treats the regulariser $\mathcal{R}$ as some unknown function. Optimising (5.5) by gradient descent will lead to updates of the form

$$x_{t+1} = x_t - \alpha_t \lambda A^T(Ax_t - y) + \nabla\mathcal{R}(x_t), \tag{5.13}$$

where A^T is the adjoint of A and α_t is a step-size. The authors replace the unknown $\nabla\mathcal{R}(x_t)$ term with a learned component, inspired by a Fields of Experts model [190]. Fixing the number of iterations and unrolling leads to an end-to-end method that includes information from the forward and its adjoint. We use the same network design and parameters as the original paper. The iterations are first initialised with a rough reconstruction given by the adjoint. The gradient of the regulariser consists of a convolution with 24 filters, kernel size of 11 and stride 1; a component-wise activation function; and then a convolution transpose with the same dimensions to return an image with the same dimensions as the input. The activation function consists of a weighted sum of 31 radial basis functions with learnable means, variance and weights. All learnable parameters are allowed to vary independently in each layer and the step-size α_t and parameter λ are also learned. We train three variations on our data: 25 radial spokes with 0.05 and 0.0125 added noise and 125 radial spokes with 0.05 added noise.

5.5 Results

Ablation Study To compare the covariance models *mean+identity*, *mean+diagonal* and *mean+covar**, visualisations of samples from the learned covariances are

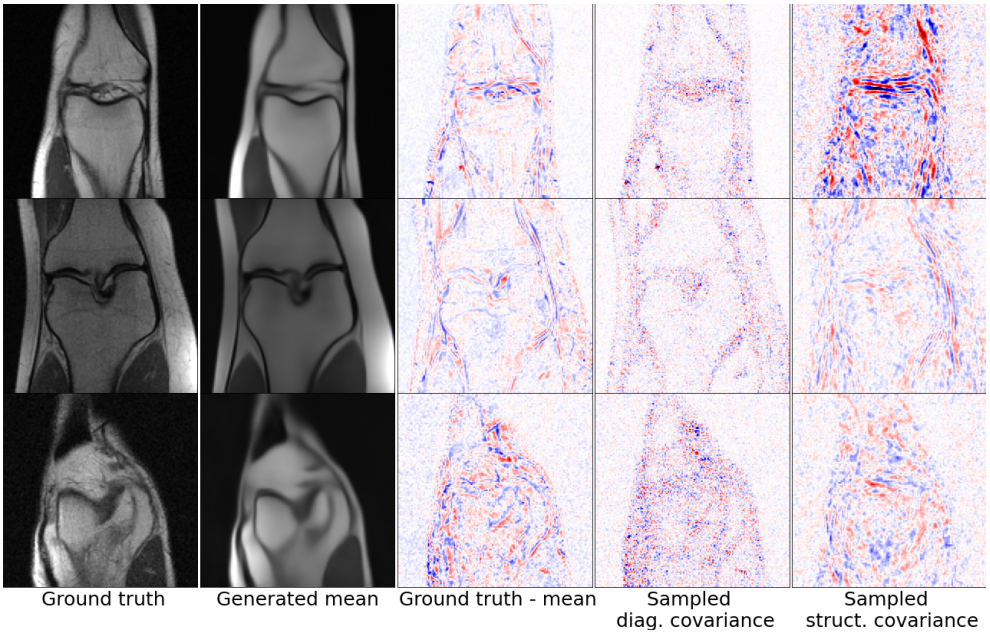


Figure 5-2: Comparing the covariance models *mean+diagonal* and *mean+covar**. The first column gives the ground truth, and the second column a reconstruction in the *range* of the VAE generator. The third gives the residual which can be considered as one sample of a zero mean Gaussian distribution with unknown covariance. The final two columns give single residual samples from our learned covariances *mean+diagonal* and *mean+covar** models. Columns are rescaled for better visualisation.

given in figure 5-2. The first three columns compare the original images, the generated mean images and the residual. This residual can be considered as one sample of a zero mean Gaussian distribution with unknown covariance. The final two images show single residual samples from our learned covariances. We see that the *mean+diagonal* places uncertainty in the right locations, but without structure, the samples are speckled and grainy. The *mean+covar** models can match some of the missing structures from the generated mean images. We note that these are random samples and therefore not expected to match the residual precisely but rather illustrate statistical similarity.

Figure 5-3 compares the effectiveness of the three covariance models as generative regularisers. The same observed data was used for each reconstruction method for each image. It is clear that the results for *mean+covar** give a consistently higher PSNR value than *mean+diag* and *mean+identity* and we use this method throughout the rest of the results section. These results support our hypothesis that learning a more flexible distance measure allows us to better fit the data and gives a better regulariser for our inverse problem.

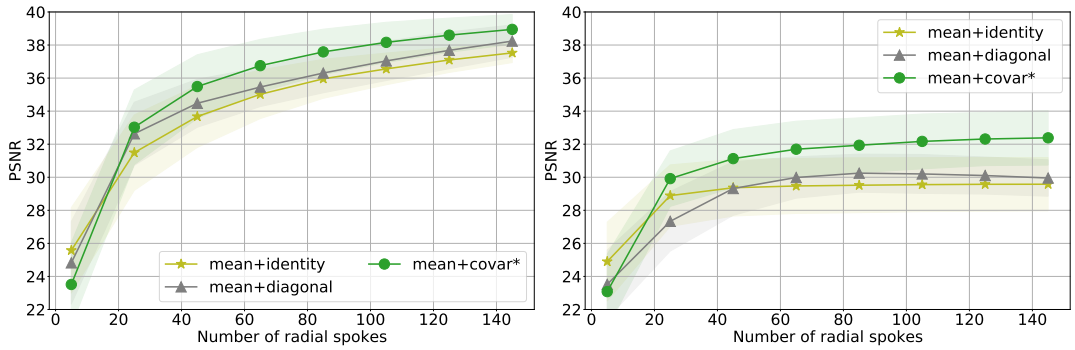


Figure 5-3: Ablation study comparing generative regularisers based on VAEs with different noise models *mean+identity*, *mean+diagonal* and *mean+covar**. The plots show PSNR values average over reconstructions of 50 test images. The mean is given by the solid line and standard deviation in the shaded box. The x-axis measures the number of radial spokes in the sampling pattern and the measured data was corrupted with additive Gaussian noise of standard deviation 0.0125 on the left and 0.05 on the right.

Prior introspection We can explicitly examine the learned structured covariance, to visually assess the prior information passed to the generative model. To do this, we take a test image, x, and use the encoder to give a latent vector, z, that corresponds to the test image. From this we can calculate the structured covariance, $\Sigma_\theta(z) \in \mathbb{R}^{d\times d}$, where $d = \dim \mathcal{X}$. Each row of the covariance matrix corresponds to a pixel in the generated image, and the row can be reshaped into an image, showing the correlations between the chosen pixel and all others. Two example images with chosen pixels highlighted with a star are shown in figure 5-4. Positive correlations are given in red, and negative correlations are in blue. We can see that the structure of the edges is present, and that, despite the local structure of the precision matrix, longer-range correlations have been learned.

Comparison with unlearned methods Figure 5-7 shows comparisons with TV and least squares reconstructions for a range of noise levels and the number of radial spokes in the sampling pattern. Image examples can be seen in columns 2 and 3 of figures 5-5 and 5-6. We see the results of *mean+covar** track TV but with improvements across the range of radial spokes and noise levels tested. Especially in the examples in figure 5-6, you can see the piece-wise constant shapes typical of a TV reconstruction, whereas for the same measured data *mean+covar** has managed to reconstruct more of the fine detail. As expected, due to the difficult nature of the inverse problem, the least squares reconstruction does poorly. As a rough indicator, our un-optimised implementation took 1.8 seconds for reconstructing one image using least squares, and 6.4 seconds using TV, for one choice of regularisation parameter. For one choice of regularisation parameter,

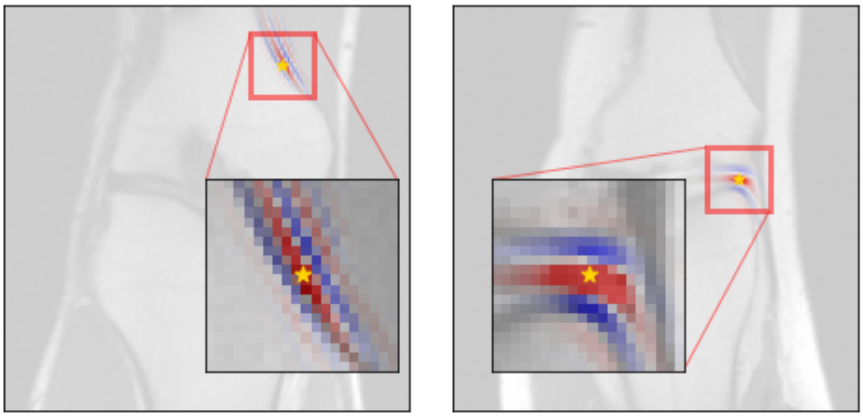


Figure 5-4: Visualisation of learned covariances between example pixels (yellow stars) and other pixel locations for the *mean+covar* model. Red indicates a high positive correlation, and blue is a strong negative correlation.

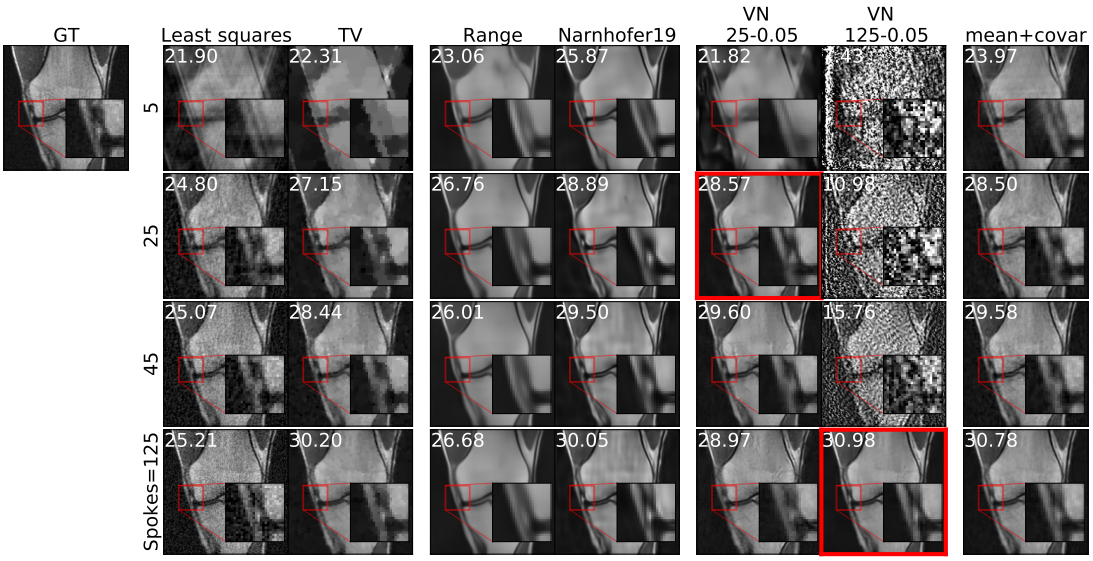


Figure 5-5: Example reconstructions of a test image for different amounts of measured data: 5, 25, 45 and 125 radial spokes, all with additive Gaussian noise with standard deviation 0.05. The columns give different reconstruction methods. The PSNR values are added in white and the red boxes indicate the settings the highlighted variational network has been trained on.

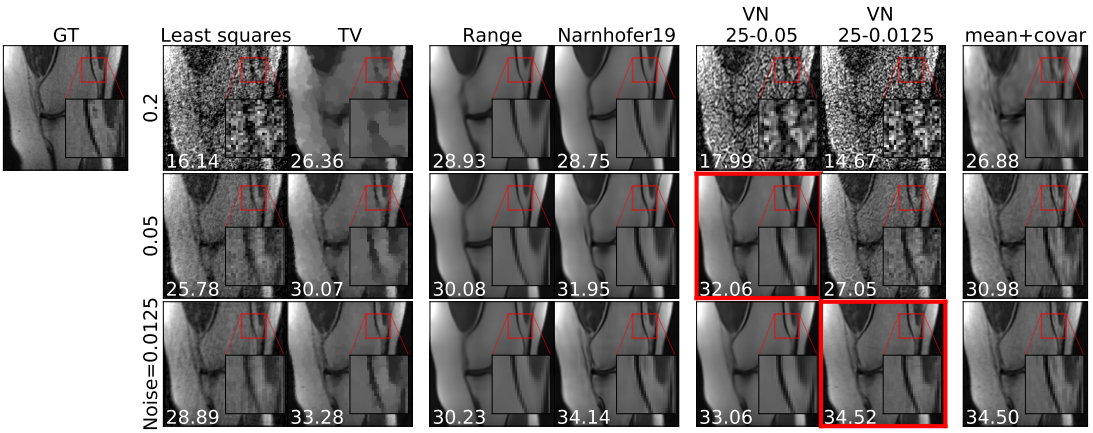


Figure 5-6: Example reconstructions of a test image with additive noise of different standard deviations: 0.2, 0.05 and 0.0125, all with a sampling pattern with 25 radial spokes. The layout is as in figure 5-5.

*mean+covar** took 78.2 seconds.

Comparison with other data-driven priors Comparisons with *Narnhofer19* and *range* [26] are given in figure 5-8 and columns 4 and 5 of figures 5-5 and 5-6. We see that the results of searching in the *range* of the generator saturate so that even with increased data in the form of radial spokes, or better data in the form of less noise, the reconstructions don't see significant improvement. The example image reconstructions reflect this; they show overly smoothed images that, although have the right structure, don't contain any of the fine detail of the ground truth. This could be because the generative model is not expressive enough to match this fine detail. Alternatively, it could be because the non-convex optimisation has failed to find a good enough minimum to match the measured data. The results of *Narnhofer19* are much more detailed and very similar in PSNR values to our *mean+covar**. In the images, there is some evidence of smoothing of details *e.g.* in figure 5-6. Our un-optimised implementation took 106.7 seconds for *Narnhofer19* and 23.9 seconds for the *range* [26] method.

Comparison with supervised end-to-end methods We now compare *mean + covar** with variational networks [93], trained end-to-end on particular noise and sampling settings. Figure 5-9 shows the PSNR results for 50 test images, with vertical lines highlighting the settings where the networks were trained. Example images are shown in columns 6 and 7 of figures 5-5 and 5-6 with the trained for setting highlighted in red. We note that the variational networks achieve the best results, in comparison to the other methods, for the settings they were trained on. This is as expected for end-to-end reconstructions. We see that the results

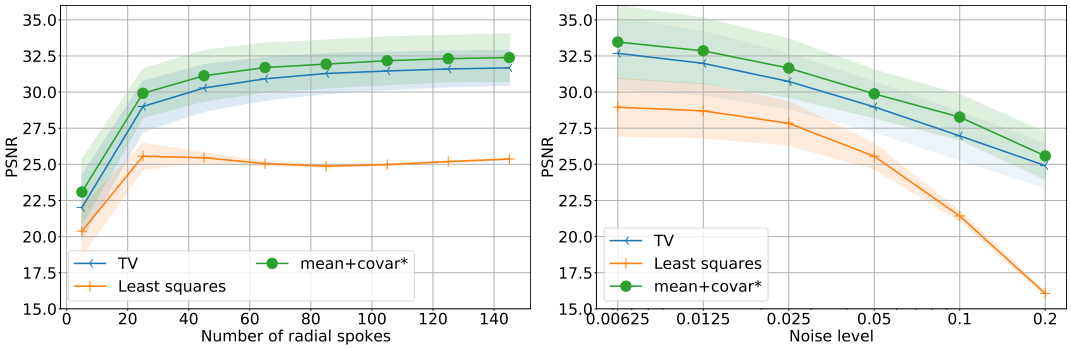


Figure 5-7: Comparison of the *mean+covar** method with the training-free methods TV and least squares. The plots show the mean PSNR values for reconstructions averaged over 50 test images, with the standard deviations given by the shaded area. The left plot shows results from measured data with additive Gaussian noise of standard deviation 0.05 and differing numbers of measured radial spokes. The right plot gives results where 25 radial spokes are used for each reconstruction but the standard deviation of the added noise varies.

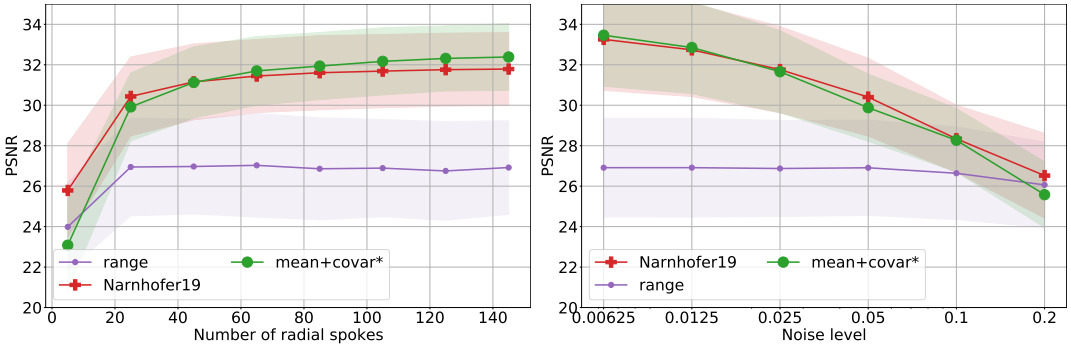


Figure 5-8: Comparison of the *mean+covar** method with *range* [26] and *Narnhofer19* [160], generative regulariser approaches. The experimental set-up is as figure 5-7.

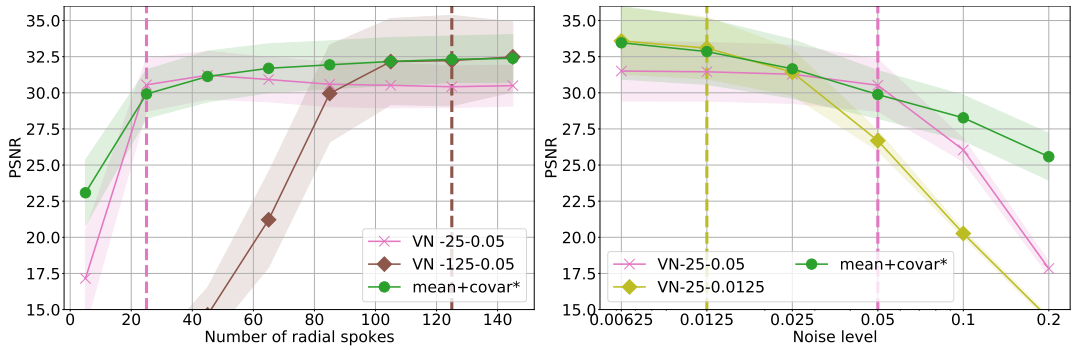


Figure 5-9: Comparison of the unsupervised *mean+covar** method with the supervised variational networks [93]. The experimental setup is as figure 5-7 but in addition, the vertical lines depict the experimental settings the variational networks were trained on.

are less optimal the further from the trained for setting, with some particularly poor results, *e.g.* in figure 5-5, while our unsupervised method remains consistent. Also as expected, the variational networks are quick to implement once trained, taking 3.4 seconds for one image

Generalisation to a Different Sampling Pattern Finally, we test the generalisation ability of the methods by changing the sampling pattern. We mask horizontal rows in the k-space, taking 16 (out of 128) centre rows and a uniformly random selection of other rows with a given probability, p. Fully sampled images correspond to $p = 1$. Example sampling patterns and one example reconstruction is given in figure 5-10. We see an improvement over TV for *mean+covar**. *Narnhofer19* gives good PSNR values but the images are potentially over-smoothed. For example, the vertical lines in the top left part of the zoomed-in region are consistently missing from the *Narnhofer19* reconstructions. The variational network has not been trained on this horizontal sampling pattern and although gives a reasonable reconstruction, seems to have some artefacts, for example in the top right and bottom left of the image.

5.6 Discussion

The numerical results show that the generative regulariser *mean+covar** consistently outperforms TV, figure 5-7, and is competitive with other unsupervised generative model regularisation schemes, figure 5-8, over a range of noise levels and sampling patterns. This is an important contribution, demonstrating a generative model, trained on an MRI dataset, provides an effective prior for image reconstruction. As part of the ablation study, figures 5-2 and 5-3, we demonstrated that models *mean+diag* and *mean+identity* provided some regularisation

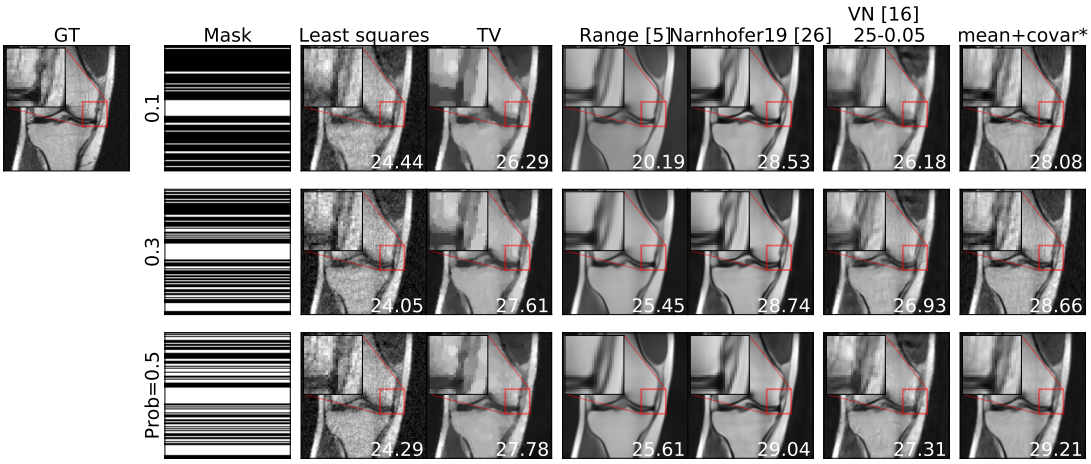


Figure 5-10: Example reconstructions of an image using measured data taken with different horizontal random sampling unseen in the training of the variational networks. The first column gives the k-space mask used to sample the data. The remaining columns give reconstructions for the different methods.

but *mean+covar** provided the most flexibility to fit the data and thus the best results. Searching in the *range* of the generator we saw evidence that the generator was not expressive enough to fit the data and that the reconstructed images did not continue to improve with more or higher-quality data, this matches with similar results in [26, 64].

The generative regularisers presented are not end-to-end methods and still require the use of an iterative optimisation scheme to reconstruct an image given an observation, furthermore the optimisation objective is non-convex. However, this optimisation is practically straightforward as the gradients can be calculated automatically, using automatic differentiation, and the optimisation can be initialised with the adjoint reconstruction. The benefit of the unsupervised approach is that retraining is not required for different forward models. Figure 5-9 shows that the variational network end-to-end method [93], although provides the best results on the forward model it was trained on, sees its success drop off as you test it on other forward model settings. An additional benefit of *generative regularisers* is that you can visually inspect the learned image prior and we demonstrate this in figure 5-4.

Training the Cholesky weights for the precision matrix proved tricky. As shown in figure 5-2, the structured covariance model has learned some of the residual structure that we expect but at the expense of applying higher variances across the whole image. This may make the structured covariance model too permissive, allowing it to fit noise or artefacts from the measurements. With an improved model, we expect that the regularisation parameters in (5.10) would not be re-

quired, they could be set by the hierarchical Bayesian derivation. Future work could consider what additional priors or alternative modelling schemes could help the learning of this structured covariance model, such as described in [62].

Although this work demonstrates a data-driven approach to inverse problems that is flexible and powerful there are a number of ways the MRI simulation could be made more realistic. As discussed in section 5.4, our experiments are inspired by the single coil fastMRI challenge which uses real-valued images and, as the name suggests, a single-coil acquisition. In reality, modern MRI scans use a multi-coil acquisition and usually produce complex images with a non-trivial phase. A multi-coil acquisition could be incorporated into our framework with a change of the forward model, *e.g.* via a SENSE formulation[180]. Modelling the structures and correlations between the real and complex parts of an MR image is an interesting open problem and subject to future work, potentially guided by the approach to colour images in [62]. Finally, the radial masks we used to simulate a radial sampling pattern, were just an approximation of the actual MR physics and future work could consider bringing this closer to real-world applications.

As in most machine learning approaches, we have assumed that the images we wish to reconstruct are similar to those in the training dataset used to train the generative model. This is not obvious in medical imaging, where damage, tumours, and illness can lead to different image presentations that may be far from those of healthy volunteers used to create datasets. Modelling this, *e.g.* by sparse deviations [57, 64] away from the range of a generative model, is an interesting area of future development.

5.7 Conclusion

Generative regularisers provide a bridge between black-box deep learning methods and traditional variational regularisation techniques for image reconstruction. We propose a highly adaptive generative regulariser that models image correlations with a structured noise network. The proposed method is trained unsupervised, using only high-quality reconstructions and is thus adaptable to different noise and k-space sampling levels. Our results show that generative regularisers are most effective when the underlying generative model outputs both an image but also a non-trivial covariance matrix for each point in the latent space. The covariance provides a learned metric that guides where the reconstruction can or cannot vary from the learned generative model. We demonstrated the success of this approach through comparisons with other unsupervised and supervised approaches.

5.A Sparsity in the Precision and Consequences for the Variance

As discussed in section 5.3, for reasons of computational efficiency we restrict the Cholesky decomposition of the precision matrix to be sparse. This restricts the type of covariance matrices we can learn. A natural question is thus: is what effect does this restriction have? Which covariance models are we not able to model with this parameterisation?

We first note that the precision matrix $\Lambda = LL^T$ is also sparse if L is sparse. We will drop the θ parameters in this appendix. Recall the precision matrix

$$\Lambda_{i,j} = \sum_l L_{i,l} L_{j,l} \tag{5.14}$$

and see that for this summation to be non-zero, there must exist a pixel $l \leq \max(i,j)$ such that $L_{i,l} \neq 0$ and $L_{j,l} \neq 0$. In this work, we take $L_{i,j}$ to be non-zero only if $j \leq i$ and pixel j lies in a $k \times k$ patch centred on pixel i, where k is odd. See figure 5-11 for possible values of j for a given highlighted i, for $k = 3$, *i.e.* it shows for $i = 36$ the positions of pixels j such that $L_{i,j}$ can be non-zero. Again, from this sparsity pattern, see figure 5-11 for the resulting values of j for a given i that $\Lambda_{i,j}$ can be non-zero.

(a) Support of $L_{i,j}$ for fixed i (b) Support of $\Lambda_{i,j}$ for fixed i

Figure 5-11: For $k = 3$ and a fixed i, given by the yellow square, the red squares correspond to the pixels j where the matrix is allowed to be non-zero.

Now consider inverting this precision matrix Λ. Think about Λ as the connectivity of a graph: $\Lambda_{i,j} \neq 0$ if there is a connection between graph nodes (pixels) i and j. In a similar multiplication as in (5.14), $\Lambda^2_{i,j}$ can be non-zero, if there exists a node l such that $\{i,l\}$ and $\{l,j\}$ are connected. Repeating this up to d, where Λ is a $d \times d$ matrix, we have $\Lambda^d_{i,j}$ can only be non-zero if there exists any path between the nodes i and j. Now consider using Cayley Hamilton theorem (see *e.g.* theorem 3.16 in [233]), $\Sigma = \Lambda^{-1} = \frac{(-1)^{n-1}}{\det \Lambda}(\Lambda^{n-1} + c_{n-1}\Lambda^{n-2} + \cdots + c_1 I_n)$, for

some constants $c_1, ..., c_n$, and so again $\Sigma_{i,j}$ can be non-zero if there exists a path between the nodes i and j.

We can thus see that

- If Λ is diagonal, then no element is connected to any other, each power of Λ is diagonal and so Σ is also diagonal.
- If Λ is a block diagonal matrix, then there are unconnected regions of the graph. Each power of Λ has the same diagonal block structure and Σ is also a block diagonal matrix with $\Sigma_{i,j} = 0$ if nodes i, j are in different regions.
- If the graph corresponding to Λ is connected, then there exists a path between all of the nodes, then Σ is dense.

As we discussed in section 5.3, a zero value in the precision matrix means that two pixels are independent, conditioned on all the other pixels in the image. From this discussion, we determine that the restriction to small neighbourhoods in the Cholesky decomposition means that we are unable to model long-range correlations with no dependent path between them. For example, consider eye colour. The eyes are two regions in an image of a face that should be highly correlated. However, between the eyes, is the nose, with skin colour and texture that is independent of the eyes, conditioned on all other pixels in the image. We can't draw a path, between the two eye regions, that moves between neighbouring pixels that have a dependent relationship between the two eye regions. As they are unconnected in the precision matrix, they remain unconnected and thus uncorrelated in the covariance matrix.

Experimentally, consider a set of 10×10 images, with striped rows. The pixel value of each of the top 5 rows is chosen independently from a uniform distribution. The top five rows are then repeated for the bottom five rows. Each pixel is identical to the other pixel in the row and all the pixels in the row 5 below and independent of pixels in any other row. An example image is given in figure 5-12a. We take 10,000 such images and numerically produce a covariance matrix, given in figure 5-12b. In this matrix, you can see the correlations between pixels in the same row, the main diagonal squares, and those in the row five rows above or below, in the off-diagonal squares. We then choose three different connectivity parameters for the Cholesky decomposition, $k = 3$, $k = 5$, and $k = 11$. The last was chosen such that pixels five rows apart, can be connected. We see that for $k = 3$, the learned covariance matrix, with the restricted parametrisation, has failed to learn the longer range connections, instead only reproducing the local correlations. For $k = 5$, we see that the long-range correlations between the rows have been modelled, however, this is at the expense of some spurious correlations between rows that should be independent. Although, as discussed above, these two separate regions can't be highly correlated with independent regions in be-

tween, with the extra flexibility given by more non-zero values in the Cholesky decomposition, we can approximate the covariance matrix well. Finally, when the connectivity in the precision can cover the jumps between rows, we can reproduce the covariance matrix without additional spurious correlations.

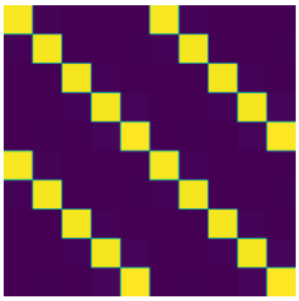
5.B Priors on the Sparse Cholesky Decomposition

In section 5.3, we discuss how the objective functions we use are not bounded from below and noted the need to use a sigmoid activation function on the diagonal of the Cholesky decomposition in order to bound the diagonals and give a well-posed minimisation problem. Also in section 5.4, we discussed the need to split the training between first training the means and then the covariance matrix. Finally, we discussed in the conclusion, that the learned network is still too permissive and that training the covariance network was challenging. An effective prior over the covariance matrix might help to mitigate all three of these issues. However, priors are tricky as we generally do not have access to the full covariance matrix, except from slow, and computationally demanding, matrix multiplication and then inversion. This means that calculating a log prior over the covariance for each iteration of training a VAE is not currently possible.

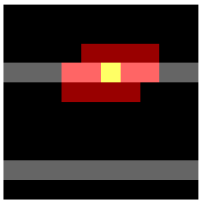
There are several possible future directions:

- Determine an efficient calculation of the covariance matrix that can be implemented on a GPU. This could allow priors to be defined on the full matrices and the log prior evaluated at each iteration of training.
- Consider priors that we could place on the Cholesky decomposition of the precision or the precision matrix. For example, the conjugate prior for the covariance matrix of a Gaussian distribution is the inverse Wishart distribution and there is a corresponding Wishart distribution over precision matrices. This, and other options, are discussed in [61], but they note the difficulty in choosing a prior that was broad enough to cover the desired covariance matrices but informative enough to be useful. This is the same trade-off that we discussed when choosing priors for inverse problems and led to us considering learned rather than hand-crafting priors. One could consider learning a prior for the covariance matrices, *e.g.* using a hierarchical model to determine some hyperparameters, but this could be difficult to train [228].
- Investigating the architecture of the generator and thus the implicit priors placed on the network. Currently, as shown in figure 5-1, we use one generator network to output both the values for the mean and the covariance

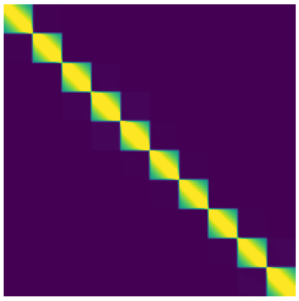
(a) Example random image



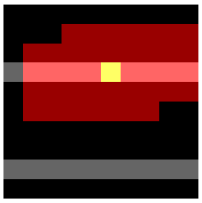
(b) Covariance matrix



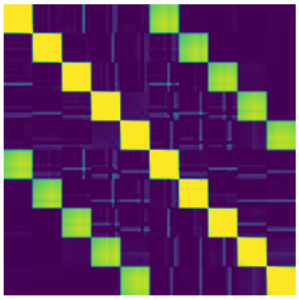
(c) Precision connectivity example $k = 3$



(d) Fitted covariance matrix, $k = 3$

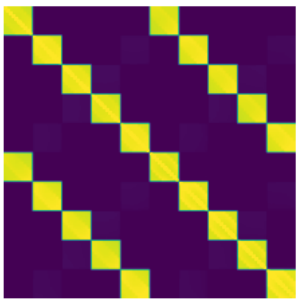


(e) Precision connectivity example $k = 5$



(f) Fitted covariance matrix, $k = 5$

(g) Precision connectivity example $k = 11$



(h) Fitted covariance matrix, $k = 11$

Figure 5-12: In (a) you see an example sample of an image from which the covariance matrix in (b) is calculated empirically. In (d),(f) and (h) you see approximations of the covariance matrix in (b), limited by a sparse parameterisation of the Cholesky decomposition of the precision matrix. In (c), (e) and (g) you can see, for the yellow pixel highlighted, the other pixels it is identical to (and thus highly correlated), in grey, and in red the possible connections that can be described by the precision matrix.

network for a single latent vector. It could be that this is not the correct approach and this puts undesirable implicit priors on the covariance or mean networks. An architecture search could include: decoupling the two networks but still generating from a single latent vector; having a completely different latent space and generator for mean and covariance, or perhaps using a generated mean image and input to a network that outputs the covariance values.

Chapter 6

Training a VAE Without Ground Truth Images for use in Inverse Problems

6.1 Introduction

In this chapter, we again take the approach of *generative regularisation* ([26, 57, 90, 218, 64]) which penalises solutions that are far from the range of a generative model. In previous work, the generator was trained to generate images similar to some training set of ground truth or high-quality reconstructed images. The challenge of this approach is the requirement for large amounts of high-quality ground truth training data. For some inverse problems, such as MRI it is difficult and expensive to collect the necessary datasets: patient confidentiality, cost and time of data acquisition, differences between hardware and imaging systems and the rarity of some diseases. In other inverse problems, such as Positron Emission Tomography (PET), high-quality ground truth images may be difficult to obtain because of the inherent noise in the measurement procedure. In this work, we consider an approach to learning a generative model for images only based on observed measurements.

We assume that instead of a dataset of high-quality reconstructions, we have access to both observed data and the forward model that produced the observations. For example we have samples $(y^i, A^i)_{i=1,\cdots,n}$ from a distribution $P_{\mathcal{Y},\mathcal{A}}$, such that there exists $x^i \in \mathcal{X}$ and that $A^i x^i \approx y^i$ up to some additive noise. We also assume the existence of some pseudo-inverse for A^i given by $A^{i\dagger} : \mathcal{Y} \to \mathcal{X}$, which gives a rough reconstruction. For inverse problems that map between the same spaces, this could just be the identity, for MRI it could be a zero-filled reconstruction, or for X-ray tomography a filtered back-projection. We use this data

to train a variation on a variational autoencoder (VAE) [125], we call *noisyVAE*. As in a standard VAE, the VAE generator, $G : \mathcal{Z} \to \mathcal{X} \subseteq \mathbb{R}^d$, takes points in the latent space and outputs a mean in the image space, with associated covariance $\Sigma \in \mathbb{R}^{d \times d}$, which gives a conditional distribution on $\mathcal{X}$. For the noisyVAE, the encoder takes rough reconstructions of observed data, $A^\dagger y$, and outputs a distribution in the latent space. For each forward model, A^i, we seek to minimise the Kullbach-Leibler (KL) divergence between the generated distribution, with A^i applied, and the measured data, observed under A^i. The KL divergence is considered in the measurement space, $\mathcal{Y}$. This gives a powerful and flexible formulation and we will demonstrate its ability to learn characteristics of the desired image distribution from only observed data.

We will also demonstrate the ability of these learned generative models to act as a generative regulariser. We then go on to test these generative models as priors to inverse problems and consider both inverse problems that match the forward model used to create the training set and inverse problems independent from the training set.

6.1.1 Contributions

- We propose a training loss for a noisyVAE, which learns from a training dataset containing only observed data, in space $\mathcal{Y}$, and the related forward model. The noisyVAE generates images in the image space $\mathcal{X}$.

- We demonstrate effective training of the noisyVAE on two hand-built ellipses datasets.

- We perform an ablation study looking at a second additional reparameterisation step used to calculate the noisyVAE loss in the domain $\mathcal{Y}$.

- We demonstrate that the VAE is effective in providing regularisation for inverse problem reconstruction. We consider the inverse problems of denoising, deconvolution and PET and first compare against other VAE models including standard VAEs trained on ground truth data or rough reconstructions of ground truth data. We compare against the generative regulariser approach of [26] and against unlearned TV, Tikhonov and maximum likelihood expectation minimisation.

- A brief literature review provides context for our work and we then set out clear avenues for future work.

6.2 Method

Standard VAE derivation Consider a standard VAE model, as discussed in chapters 3 and 5, and consider that for any point $z \in \mathcal{Z}$, in the latent space, the network outputs a conditional distribution on the image space, $\mathcal{X}$, given by $p_{G,\Sigma}(\cdot|z;\theta) = \mathcal{N}(x; G_\theta(z), \Sigma_\theta(z))$ given by (5.1) or (3.22). The VAE is trained by minimising an upper bound on the KL divergence between the generated, $p_{G,\Sigma}$, and data distribution, which is equivalent to maximising $\mathbb{E}_{x\sim p^*_{\mathcal{X}}} \log p_{G,\Sigma}(x;\theta)$, with respect to the neural network parameters θ.

In order to obtain $p_{G,\Sigma}(x;\theta)$ from the conditionally generator $p_{G,\Sigma}(x|z)$, equation (3.4) integrates our the latent vector, z. However, this is intractable, except by expensive sampling and instead the intractable distribution over the latent space, $p_{G,\Sigma}(z|x;\theta)$ is approximated by an encoder $q(z|x;\psi) = \mathcal{N}(z; \mu_\psi(x), \mathrm{diag}(\sigma^2_\psi(x))) =: \mathcal{N}_{x,\psi}(z)$ with mean and variance given by neural networks μ_ψ and σ^2_ψ. Following the derivation in chapter 3, we maximise a lower bound to the KL divergence given by

$$\log p_{G,\Sigma}(x;\theta) \geq \mathbb{E}_{z\sim q(z|x;\psi)} \left[\log p_{G,\Sigma}(x|z;\theta)\right] - d_{KL}(q(z|x;\psi)\|p_{\mathcal{Z}}) \tag{6.1}$$

After incorporating the Gaussian distribution choices for p and q, training a VAE becomes a minimisation of

$$\mathbb{E}_{x\sim p^*_{\mathcal{X}}} \left[\mathbb{E}_{z\sim\mathcal{N}_{x,\psi}} \left(\log(|\Sigma_\theta(z)|) + \frac{1}{2}\|x - G_\theta(z)\|^2_{\Sigma_\theta(z)}\right) + d_{KL}(\mathcal{N}_{x,\psi}\|p_{\mathcal{Z}})\right] \tag{6.2}$$

with respect to ψ and θ. Note that the KL divergence is taken over the latent space $\mathcal{Z}$. The expectation over $x \sim p^*_{\mathcal{X}}$ is approximated empirically over the training dataset. Similarly, the expectation over $z \sim \mathcal{N}_{x,\psi}$ is estimated empirically although this must be done carefully because we wish to optimise over ψ and hence take derivatives with respect to ψ. The *reparamaterisation trick* writes $z = \mu_\psi(x) + \epsilon \cdot \mathrm{diag}(\sigma_\psi(x))$ where $\epsilon \sim \mathcal{N}(0,1)$. It is straightforward to sample from a standard normal distribution, and in practice, only one sample is taken to approximate the expectation.

NoisyVAE Consider now that we do not have samples $x \sim p^*_{\mathcal{X}}$ but instead only have samples $\{y^i, A^i\}_{i=1,\cdots,n}$ from a distribution $p_{\mathcal{Y},A}$. Thus we have both the observed data and the forward model used to make the observation, *i.e.* $A^i x^i \approx y^i$ for $x^i \in \mathcal{X}$. We also assume the existence of some pseudo-inverse to A^i, given by $A^{i\dagger} : \mathcal{Y} \to \mathcal{X}$. For a pictorial representation of the noisyVAE, see figure 6-1. As above, set $p_{G,\Sigma}(x|z;\theta) = \mathcal{N}(x; G_\theta(z), \Sigma_\theta(z))$. We consider that the posterior of the observed data given latent vector z is

$$p_{G,\Sigma|\mathcal{A}}(y|z,A,\theta) = \int p(y|x,z,A) p_{G,\Sigma}(x|z;\theta) dx \tag{6.3}$$

$$= \int p(y|x;A)p_{G,\Sigma}(x|z;\theta)dx \tag{6.4}$$

$$= \mathbb{E}_{x\sim p_{G,\Sigma}(x|z:\theta)}p(y|x;A). \tag{6.5}$$

We consider the objective of a VAE is to minimise the KL divergence, in the measurement space $\mathcal{Y}$, between the distribution of generated data, under a forward model A, and the training data distribution, conditioned on the same forward model.

$$\min_{\theta} \mathbb{E}_A d(P_{\mathcal{Y}|A}, P_{G,\Sigma|A}) \tag{6.6}$$

where both are distributions over $\mathcal{Y}$. The expectation over A allows for information from multiple forward models to be combined.

We take $q(z|y;\psi) = \mathcal{N}(z;\mu_\psi(A^\dagger y), \mathrm{diag}(\sigma^2_\psi(A^\dagger y))) =: \mathcal{N}_{A,y,\psi}(z)$. Hence, for the encoder, we first calculate a rough reconstruction of the measured data before using neural networks μ_ψ and σ_ψ to encode a distribution in the latent space. This allows us to use the same architectures for the encoder as the standard VAE and we will discuss this choice further in the chapter conclusion.

Hence, by replacing x with y in (6.1), and using (6.5) we consider the maximisation of an objective:

$$\log p_{G,\Sigma}(y|A;\theta) \geq \mathbb{E}_{y,A}\left(\mathbb{E}_{z\sim\mathcal{N}_{y,\psi}} \log\left[\mathbb{E}_{x\sim N(x;G_\theta(z),\Sigma_\theta(z))}\left(p(y|x,A)\right)\right] - d_{KL}\left(\mathcal{N}_{A,y,\psi}\|p_Z\right)\right) \tag{6.7}$$

$$\geq \mathbb{E}_{y,A}\left(\mathbb{E}_{z\sim\mathcal{N}_{y,\psi}}\left[\mathbb{E}_{x\sim N(x;G_\theta(z),\Sigma_\theta(z))}\left(\log p(y|x,A)\right)\right] - d_{KL}\left(\mathcal{N}_{A,y,\psi}\|p_Z\right)\right) \tag{6.8}$$

where we have ignored terms independent of ψ and θ. Note that the KL divergence term is in the latent space $\mathcal{Z}$ as before.

The hope is that we can use the same re-parameterisation trick to calculate the expectation over x as we do over z. Thus for the expectation over z set $z = \mu_\psi(x) + \epsilon \cdot \mathrm{diag}(\sigma_\psi(x)$ and instead take empirical samples of $\epsilon \sim \mathcal{N}(0, I)$. Again the expected value can now be calculated empirically, taking samples of $\nu \sim \mathcal{N}(0, I)$. This is a standard variational inference strategy.

Example 1: Inverse Problem with Gaussian noise and $\Sigma_\theta(z) = \rho^2 I$ For a simple example where the inverse problem noise model is additive Gaussian noise and the generated distribution has a spherical noise model, take:

- $p(y|x,A) = \mathcal{N}(y;Ax,\gamma^2 I)$ where γ is a fixed constant. Thus $\log p(y|x,A) = -\frac{\|Ax-y\|_2^2}{2\gamma^2} +$ terms independent of x and z.

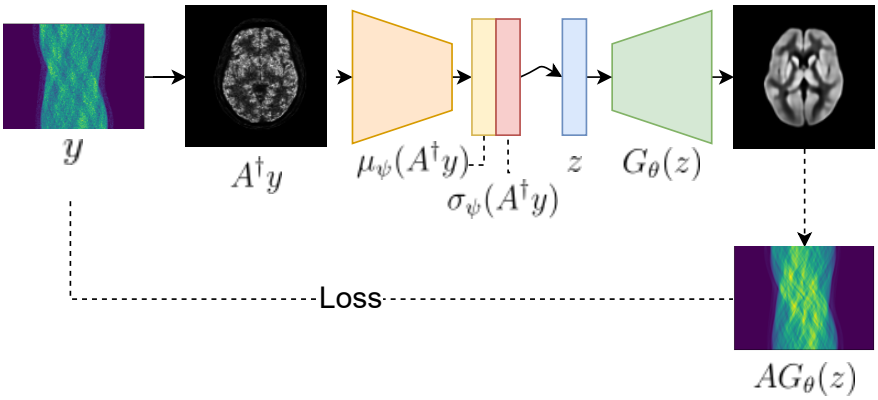


Figure 6-1: Pictorial representation of a noisyVAE. Observed data is roughly reconstructed, using $A^\dagger$, before encoding to a mean and variance in the latent space. A sample from the resulting distribution is passed to the generator, which outputs an image in the space $\mathcal{X}$. The first part of the noisyVAE objective (6.8) measures the loss between the original measured data, and the output of the forward model applied to the generated image, given by the dotted line labelled 'Loss' in the diagram.

- $\Sigma_\theta(z) = \rho^2 I$ so that $p_{G,\Sigma}(x;\theta) = \mathcal{N}(x; G_\theta(z), \rho^2 I)$ and $x \sim \mathcal{N}(x; G_\theta(z), \rho^2 I)$ if $x \sim G_\theta(z^j) + \rho\nu^k$ and $\nu^k \sim \mathcal{N}(0, I)$.

We thus have that maximising (6.8) becomes a minimisation of:

$$\mathbb{E}_{y,A}\left(\mathbb{E}_{z\sim\mathcal{N}_{y,\psi}}\left[\mathbb{E}_{x\sim N(x;G_\theta(z),\rho^2 I)}\left(\frac{\|Ax-y\|_2^2}{2\gamma^2}\right)\right] + d_{KL}\left(\mathcal{N}_{A,y,\psi}\|p_Z\right)\right) \tag{6.9}$$

with respect to ψ, θ and ρ. Using the reparameterisation trick we find that (6.9) can be estimated

$$\sum_i^N \sum_j^J \sum_k^K \frac{\|A^i x(\epsilon^j, \nu^k, y^i, A^i, \theta, \psi) - y^i\|_2^2}{2\gamma^2} + d_{KL}\left(\mathcal{N}_{A^i,y^i,\psi}\|p_Z\right) \tag{6.10}$$

where $x(\epsilon^j, \nu^k, y^i, A^i, \theta, \psi) = G_\theta(z^j) + \rho\nu^k$, $z^j = \mu_\psi(A^{i\dagger} y^i) + \epsilon^j \sigma(A^{i\dagger} y^i)$ and $\nu^k, \epsilon^j \sim \mathcal{N}(0, I)$ and J and K are the number of samples, usually some small positive integer.

Example 2: Inverse Problem with Poisson noise and $\Sigma_\theta(z) = \rho^2 I$ For the case where measured data is an instance of a Poisson distribution with mean given by Ax and the generated distribution has a spherical Gaussian noise model, take:

- $p(y|x, A) = Po(y; Ax)$. We choose that

$$\log p(y|x, A) = \sum_{l=1}^{m} (Ax - y)_l + y_l \log\left(\frac{y_l}{(Ax)_l}\right) \tag{6.11}$$

  up to constant terms independent of x. Note that this is defined only for $(Ax)_l \geq 0, \forall l$ and that $y_l \geq 0, \forall l$ due to the data being measured counts.

- $\Sigma_\theta(z) = \rho^2 I$ so that $p_{G,\Sigma}(x; \theta) = \mathcal{N}(x; G_\theta(z), \rho^2 I)$ and $x \sim \mathcal{N}(x; G_\theta(z), \rho^2 I)$, if $x \sim G_\theta(z^j) + \rho\nu^k$ and $\nu^k \sim \mathcal{N}(0, I)$.

We thus have that maximising (6.8) becomes a minimisation of:

$$\mathbb{E}_{y,A}\left(\mathbb{E}_{z\sim\mathcal{N}_{y,\psi}}\left[\mathbb{E}_{x\sim N(x;G_\theta(z),\rho^2 I)}\left(\sum_{l=1}^{m}(Ax-y)_l + y_l\log\left(\frac{y_l}{(Ax)_l}\right)\right)\right] + d_{KL}\left(\mathcal{N}_{A,y,\psi}\|p_Z\right)\right) \tag{6.12}$$

with respect to ψ, θ and ρ. Using the reparameterisation trick we find that (6.12) is equivalent to

$$\sum_{i}^{N}\sum_{j}^{J}\sum_{k}^{K}\sum_{l=1}^{m}(A^i x(\epsilon^j, \nu^k, y^i, A^i, \theta, \psi)) - y^i)_l + y_l^i \log\left(\frac{y_l^i}{(A^i x(\epsilon^j, \nu^k, y^i, A^i, \theta, \psi))_l}\right) + d_{KL}\left(\mathcal{N}_{A^i, y^i, \psi}\|p_Z\right) \tag{6.13}$$

where $x(\epsilon^j, \nu^k, y^i, A^i, \theta, \psi) = G_\theta(z^j) + \rho\nu^k$, $z^j = \mu_\psi(A^{i\dagger}y^i) + \epsilon^j \sigma(A^{i\dagger}y^i)$ and $\nu^k, \epsilon^j \sim \mathcal{N}(0, I)$.

Priors to inverse problems With this generative model as a prior, we can derive a generative regulariser using Bayes' formula. As above, consider an observed noise model, $p(y|x; A)$, the likelihood of the observation y, given data x and with a forward model, A, where A could be the from the same distribution as used to train the generative model but need not be. Take the prior on $\mathcal{Z}$ to be given by $p(z) = \mathcal{N}(z; 0, I)$. With our learned noisyVAE, $p_{G,\Sigma}$, we have the posterior of the reconstructed image given the observed data given by:

$$p(x, z|y) \propto p(y|x, z)\, p_{G,\Sigma}(x|z)\, p_{\mathcal{Z}}(z) \tag{6.14}$$

$$= p(y|x)\, p_{G,\Sigma}(x|z)\, p_Z(z). \tag{6.15}$$

A MAP estimate for x can be found by minimising the following objective with respect to x and z

$$\log p(y|x; A) + \log(|\Sigma_\theta(z)|) + \frac{1}{2}\|x - G_\theta(z)\|^2_{\Sigma_\theta(z)} + \frac{1}{2}\|z\|_2^2. \tag{6.16}$$

Compared to a variational regularisation framework, *e.g.* equation (2.4), we can treat the final three terms of (6.16) as a regularisation function.

We recognise that (6.16) is only an estimate for the MAP, as we should integrate out z in (6.15), before maximising over x. Currently integrating over z is intractable, except by expensive sampling. Note also that the prior on z here should really be the amalgamated posterior over z after training, *i.e.* $p_{G,\Sigma}(x|z)$ should be $p_{G,\Sigma}(x|z, \left\{y_{i,A^i}\right\}_{i=1,\dots,N})$. It is dependent on the data and training. We instead use the $\mathcal{N}(0, I)$ prior used in training because we have easy access to it.

6.3 NoisyVAE Training and Evaluation

6.3.1 Datasets

We test on three datasets. The first two are custom-built datasets of ellipses, called *single ellipse* and *ellipses*. In the *single ellipse* dataset there is just one ellipse on a black background with a solid fill of intensity drawn from a uniform distribution between 0 and 1. In the *ellipses* dataset, there are three ellipses in each image. Each ellipse is added onto a black background with an intensity drawn from a uniform between 0.16 and 0.33. Their centres, major and minor axis and rotation angle are also chosen from a uniform distribution. Both datasets have image size 128×128 and have 5000 and 10000 training examples for the *single ellipse* and *ellipses* datasets, respectively. Example images are given in the first column of figures 6-4 and 6-5 for *ellipses* and *single ellipse*, respectively. In addition, we use a set of positron emission tomography images based on 20 brain MRI phantoms obtained from BrainWeb [50, 47]. Each phantom is segmented into grey matter, white matter and skin and are then assigned a constant intensity reflecting an expected PET radiotracer, 18F-FDG uptake. The constant uptake in the regions has been modelled as 1, 0.25 and 0.125, respectively. We extract 665 2D slices from 19 3D brain scans and reserve one scan, of 35 images for testing. Each slice is cropped to the centre 128x128 pixels. Two example images are given in figure 6-3.

6.3.2 Forward models

Denoising For denoising experiments, we take $A = I$ and consider additive Gaussian noise. Note that $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$. We don't use a rough reconstruction method and instead take $A^\dagger = I$.

Deconvolution We consider a Gaussian blur with convolution kernel $K \in \mathbb{R}^{k \times k}$ given by

$$K_{i,j}(\alpha) \propto \exp\left(-\frac{(i-\frac{k}{2})^2 + (j-\frac{k}{2})^2}{\alpha^2}\right) \tag{6.17}$$

which is then normalised by the sum of all the entries of the kernel. The measured data is a result of a convolution with this kernel, with zero padding, to ensure that $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{d_1 \times d_2}$. We fix $k = 40$ and allow α to vary. The larger the value of α the more blurred the image. We take 50 iterations of the TV reconstruction algorithm as a rough reconstruction. The regularisation parameter was chosen for the rough reconstruction by visual inspection.

Positron Emission Tomography The goal of PET imaging is to reconstruct the concentration of a radioactive tracer from measurements along line integrals. We estimate a PET 2D forward problem with the following elements:

- The Radon transform, denoted S. We choose a parallel beam geometry and choose the detector shape and number of measured angles to ensure that this is injective. The tomographic modelling is done via the open-source package ODL [1].
- Factors, F, a combination of attenuation from the simulated CT and normalisation factors.
- Background, b, consisting of random observations, which are modelled as spatially constant, and scatter of which magnitude varies smoothly across the domain, resembling the shape of the X-ray transform of the ground truth.
- A scaling factor β, which determines the number of counts observed. We vary the noise level between 100,000-1,000,000 counts, the higher the count the 'easier' the problem, with another 100,000 counts contributing to the background.

See figure 6-2 for example values for each of these parts. Thus for ground truth image, x, we observe data, y, as an instance of Poisson noise with mean $\beta FSx+b$. We consider that for observed data, y, the components of the forward model, F, S, β and b, are known. In PET reconstruction, these values are usually pre-computed (for example see [67]). It might be an interesting future research direction to use machine learning or generative model approaches to predict these quantities from observed data.

We use a maximum likelihood expectation minimisation (MLEM) method to give a rough reconstruction for the PET forward problem [202, 135]. MLEM is

defined by the following iterative updates, where all operations are completed element-wise:

$$x^{k+1} = \frac{x^k}{A^T\mathbf{1}} A^T \left(\frac{y}{Ax^k + b} \right). \tag{6.18}$$

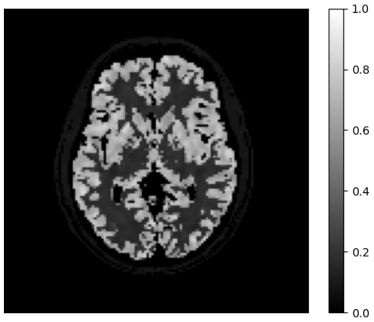
In each update, the current estimate of the image is forward projected using the weighted sinogram, and the background is added. This is then compared to the measured data to obtain a ratio of correction factors. These factors are then back projected using the adjoint of the forward operator, and normalised by the adjoint applied to an image of constant magnitude one. The result of this is then used to correct the current estimate. We use 50 updates to provide a rough reconstruction. Example MLEM reconstructions are given in figure 6-3.

6.3.3 Models and Training

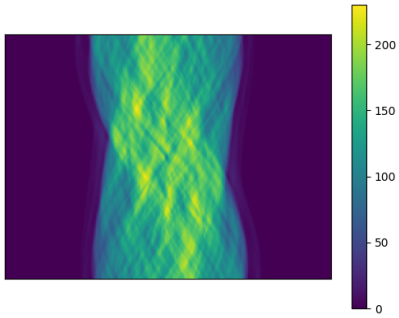
NoisyVAE Objective We consider the proposed noisyVAEs with $\Sigma_\theta = \rho^2 I$, where ρ is a learnable parameter. For the *single ellipse* and *ellipses* datasets, we consider denoising and deconvolution inverse problems with Gaussian noise. For the PET dataset we consider a simulated PET forward problem. We minimise a slight variation of the objective given in (6.13) or (6.10) for measurements observed with Poisson or Gaussian noise, respectively. We include a factor in front of the KL term, which is tuned as a hyperparameter. This is similar to a β-VAE [100]. This extra parameter encompasses the noise level in the data, γ, in (6.10).

Comparisons In this work, we train three different types of VAEs. The first is a standard VAE [125], trained on ground truth images (VAE-GT). With a dataset of observations and the forward model that produced the observations, it is possible to use existing methods to reconstruct image data. The second comparison method, VAE-TV, trains a standard VAE on TV reconstructed images.

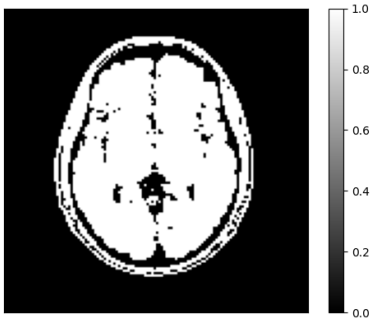
Architectures All the VAE networks are built of a combination of three types of resnet-style blocks: up-sampling, down-sampling or identity. Each block consists of a convolutional layer with stride 1, a resizing layer, followed by two more convolutional layers, and then a ReLU activation function. The output of this process is then added to a resized version of the original input to the block. The resizing layer is either a bilinear interpolation for an up-sampling block, increasing width and height by a factor of 2; convolutions with stride 2 for a down-sampling block or a convolution with stride 1 for an identity block. We choose a latent space of size 20, 100 and 100 for the *single ellipse*, *ellipses* and PET datasets, respectively. The generative network consists of a single dense layer outputting



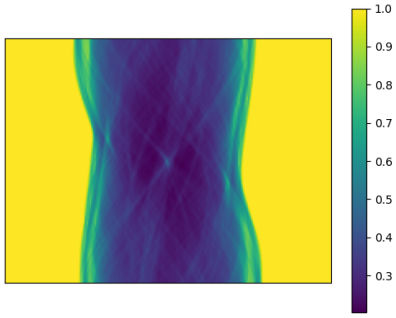
(a) Ground truth image, x



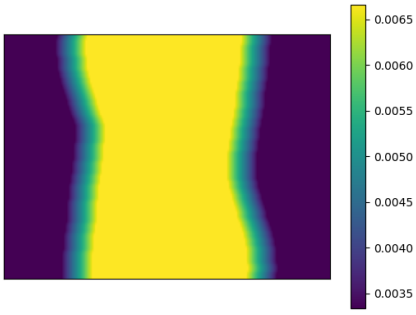
(b) Sinogram, Sx



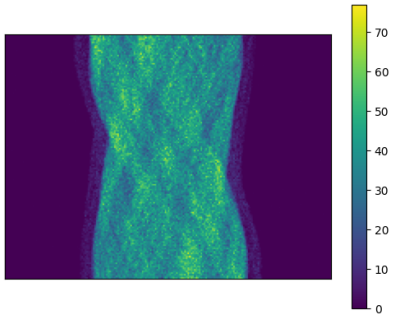
(c) The support of the ground truth image



(d) Factors, F, where attenuation is modelled from the forward projection, S applied to the support of the ground truth image.

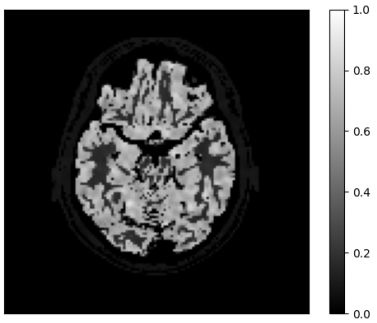


(e) Background, b, modelled as a gaussian blur, with reflective boundary conditions, of the support of the sinogram data Sx.

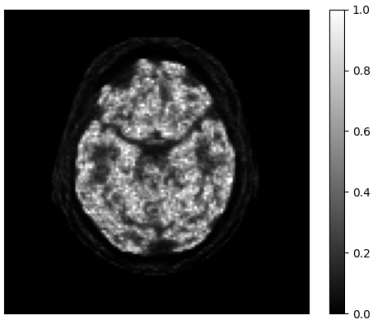


(f) Observed data, $Po(\beta FSx + b)$.

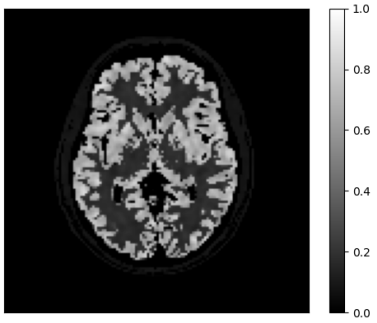
Figure 6-2: Example PET forward problem - 500,000 counts.



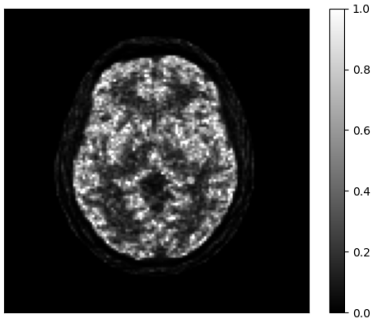
(a) Ground truth image, x



(b) MLEM reconstruction - 1,000,000 counts



(c) Ground truth image, x



(d) MLEM reconstruction - 500,000 counts

Figure 6-3: Example PET MLEM reconstructions.

8x8 images with 16 channels before a resnet block without resizing to give 8x8 images with 256 channels, then four up-sampling blocks are applied giving image sizes 16x16, 32x32, 64x64 and 128x128 with channels 512, 256, 128 and 64 and finally another resnet block without resizing to reduce the channels down to one output image.

Training Models were built and trained in TensorFlow using an NVIDIA RTX 2080 - 8GB GPU. After calculating rough reconstructions over the dataset, training the noisyVAEs took approximately 24 hours on *single ellipse* and *ellipse* and 13 hours on the PET dataset, due to its smaller size.

Generation examples For an example of how a noisyVAE has learned from data with added Gaussian noise, see figures 6-4 and 6-5. These plots show images from the test set and for each test image, x, we have used gradient descent with a random initialisation to output $G(\arg\min_z \|G(z) - x\|_2^2)$. Linking to criteria A in chapter 4, we would hope that our learned generators can reconstruct all possible images in the test dataset. We can see that the noisyVAE has achieved very good results for the *ellipses* dataset in figure 6-4, obtaining better PSNR values than the results trained on the ground truth data (VAE-GT) and TV reconstructions (VAE-TV), even managing to reconstruct the small white triangle indicated by the red arrow. In particular, the edges are much sharper than those generated from VAE-TV, which has inherited the blurred ragged edge from the rough reconstructions, see e.g. the green arrow. On the *single ellipse* dataset, figure 6-5, there is even more noise in the data, and the noisyVAE has gained some added texture in its generated images, compared to those generated by VAE-GT. In the slice plot, in figure 6-5b you can see that the VAE-TV has lost both sharp boundaries, at the base and top of the 'hat' shape, giving the purple smoothed line. VAE-GT has managed both sharp boundaries while the noisyVAE has managed the sharp boundaries at the bottom, correctly learning the black background, but has added texture in the middle of the ellipse.

One has to be careful when making conclusions from these plots. As discussed in chapter 4, a failure to reconstruct a test image could be a failure to find a suitable point in the latent space to map to a test image and does not necessarily mean a test image is not in the range of the generator The encoder and latent spaces for each of these generators were trained independently and, due to the stochastic nature of training, one could be more or less suitable for this type of reconstruction. However, the noisyVAE can learn something from some quite degraded data.

Ablation study - reparamaterisation step Figure 6-6 compares a VAE trained on ground truth *ellipses* data, with a VAE-TV and three noisyVAEs

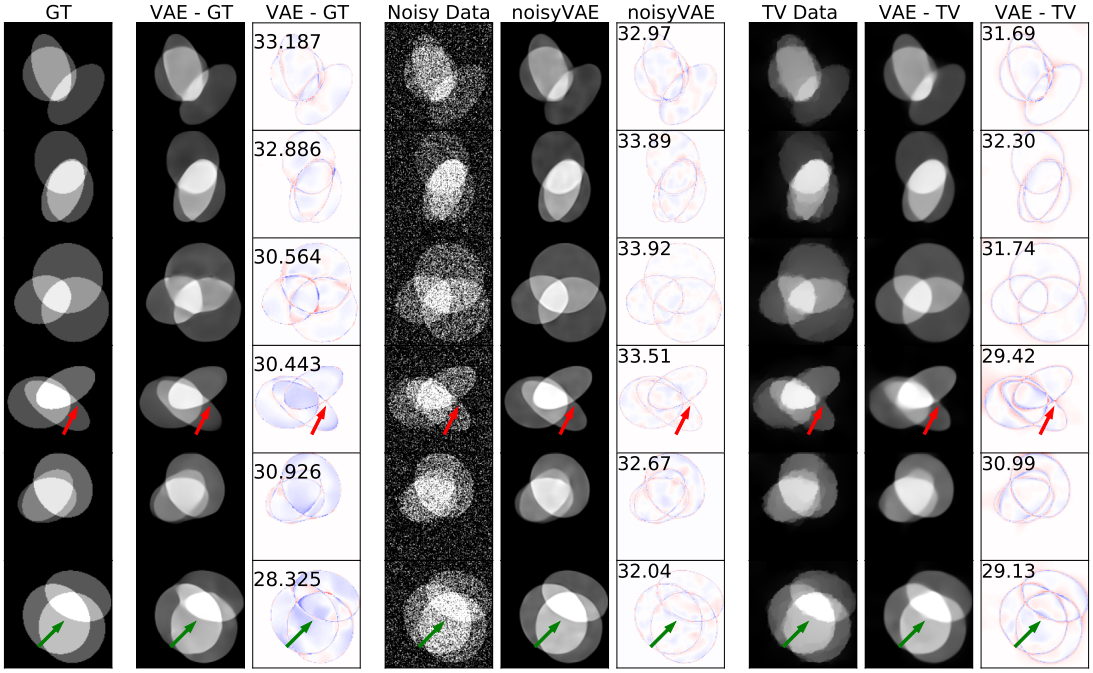
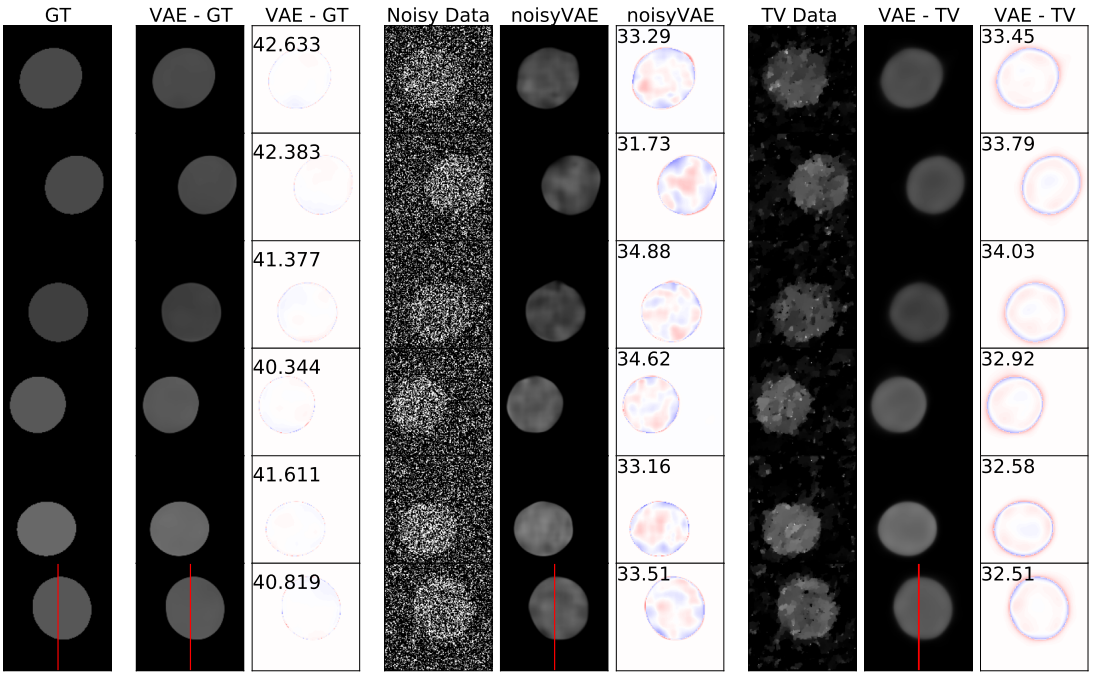
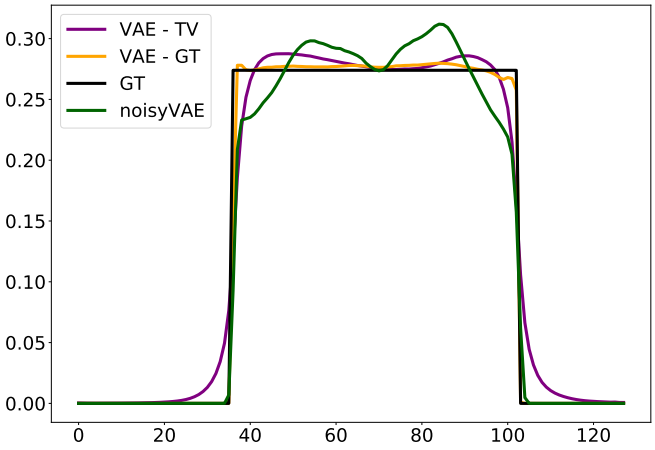


Figure 6-4: The first column gives ellipses from the test dataset we wish to reconstruct. The second column gives the result of $G(\arg\min_z \|G(z) - x\|_2^2)$, where G is a VAE trained on ground truth data. The third column gives the error. The next three columns demonstrate the data the noisyVAE was trained on (0.6 added noise), and the attempt to reconstruct the ground truth. The final three columns give the TV reconstructions of the noisy data used to train the VAE-TV and the ground truth data reconstructions.



(a) The first column gives an ellipse from the test dataset we wish to reconstruct. The second column gives the result of $G(\arg\min_z \|G(z) - x\|_2^2)$, where G is a VAE trained on ground truth data. The third column gives the error. The next three columns demonstrate the data the noisyVAE was trained on (0.6 added noise), and the attempt to reconstruct the ground truth. The final three columns give the TV reconstructions of the noisy data used to train the VAE-TV and the ground truth data reconstructions. The slice along the red line of the final column is plotted in figure 6-5b.



(b) Image slices along the red lines in figure 6-5a.

Figure 6-5: Testing the ability of the generative models to reconstruct examples from the *single ellipse* test dataset.

trained with data obtained with a convolution forward operator ($\alpha = 5$) and 0.1 standard deviation added noise. The plot has two key messages. The first is that we can learn a noisyVAE from observed data from an ill-posed inverse problem such as deconvolution. As shown by the green arrow, the generated images from the noisyVAEs have undesired learned texture, and all generative models have failed to get the point indicated by the red arrow, however, there are three ellipses in all the noisyVAE reconstructions.

The noisyVAEs were trained using a second reparameterisation step, in the image space. Recall, in (6.8), the expectation $\mathbb{E}_{x\sim N(x;G_\theta(z),\Sigma_\theta(z))}$ is calculated empirically and we now investigate the number of samples required for this estimation. Therefore, the second point of this plot, is that, in practice, we don't see any effect of changing the number of samples we take at this step. In all other experiments in this chapter, we take just one sample (K=1 in (6.13) and (6.10)).

6.4 Numerical Results - Inverse Problems

6.4.1 Reconstruction Methods

We test our proposed noisyVAE on its effectiveness as a prior for use in inverse problems. We use the variational regularisation framework given in (2.4). For the data-discrepancy term, $\mathcal{D}$, we use the 2-norm for the denoising and deconvolution inverse problem, and then the KL loss in (6.11) for the PET inverse problem. For the regularisation term, $\mathcal{R}$, we consider a *soft* restriction to the range of the generator, inspired by (6.16)

$$\mathcal{R}(x) = \min_z \lambda \left(\|G(z) - x\|_2^2 + \frac{\mu}{2}\|z\|_2^2 \right). \tag{6.19}$$

The addition of the two regularisation parameters λ and μ is in recognition that this is a non-convex problem and that our modelling assumptions are imperfect, for example, the prior on $\mathcal{Z}$ may not match the posterior after VAE training.

We compare to restricting solutions to be in the *range* of a VAE, a method introduced by [26], giving regulariser

$$\mathcal{R}(x) = \min_z \lambda \left(\iota(G(z) - x) + \frac{\mu}{2}\|z\|_2^2 \right). \tag{6.20}$$

For the *range* approach we use gradient descent with backtracking, see algorithm 1, initialised at a point in the latent space sampled from a standard normal distribution. For the *soft* approach, we use alternating gradient descent with backtracking line search, see algorithm 2, where gradient descent steps are taken, alternating in the x and z space. For the *ellipses* and *single ellipse* datasets, to initialise we first run the *range* approach, to find a suitable point in the latent

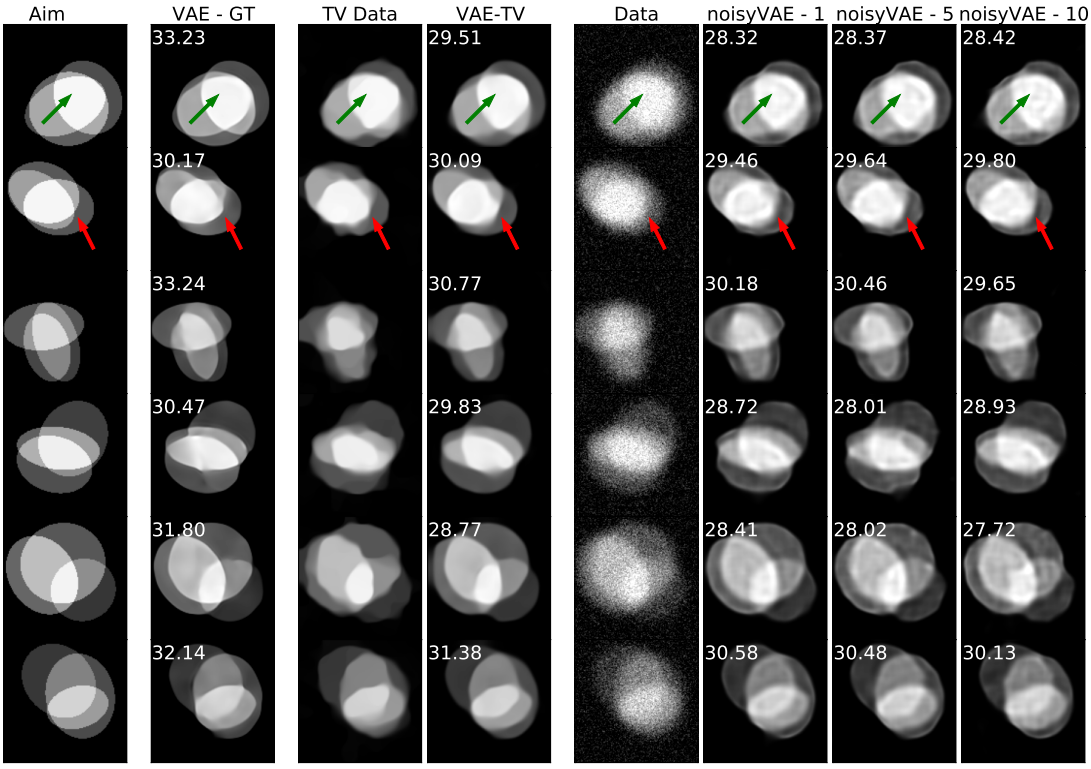


Figure 6-6: Plots to show the affect of the reparameterisation step in (6.8); taking different numbers of samples to approximate $\mathbb{E}_{x\sim N(x;G_\theta(z),\Sigma_\theta(z))}$. The first column gives ellipses from the test dataset we wish to reconstruct. The second column gives the result of $G(\arg\min_z \|G(z)-x\|_2^2)$, where G is a VAE trained on ground truth data. This is restricting to the range of the generative model. The third and fourth column gives examples of rough reconstructions used to train VAE-TV and the results of reconstructing the noise-free test images using VAE-TV. The fifth column gives the noise data used to train the noisyVAEs and the last three columns give the reconstruction results for noisyVAES trained with 1, 5 and 10 samples for the parametrisation step.

space that matches the data to initialise z_0, and initialise $x_0 = G(z_0)$ up to a small perturbation. For the PET dataset and forward problem, we initialise by running 50 iterations of MLEM and then use this rough reconstruction and an encoding of the rough reconstructions to initialise x and z, respectively. Additionally, PET images are restricted to be positive and so the gradient descent steps in x in algorithms 1 and 2 are followed by a projection onto the set of positive pixel values.

We compare these reconstructions to *range* and *soft* results obtained from VAEs trained on ground truth data (*VAE-GT*) and TV reconstructed data (*VAE-TV*). We will also compare against unlearned Tikhonov and TV reconstructions and, for PET, an MLEM approach. Regularisation parameters, and the iteration number for MLEM, were selected via a grid search to maximise the peak signal–to–noise ratio (PSNR) for each image, demonstrating best achievable results.

For rough timings, for one set of regularisation parameters and the denoising inverse problem, the *soft* method took on the order of 1-2 seconds for the *ellipses* and *single ellipse* dataset and a similar amount of time for the PET inverse problem on the PET dataset. TV denoising in comparison took between 1 and 2 minutes for the *single ellipse* and *ellipses* datasets and MLEM took on the order of 1-2 seconds.

6.4.2 Denoising

Single ellipse dataset We first consider denoising with a range of different noise levels. We use a noisyVAE trained on observations with 0.6 added noise for the *single ellipse* dataset. The VAE-TV model is trained on TV reconstructions from the same noisy datasets and VAE-GT is trained on clean data. Example reconstructed images for the *single ellipse* are given in figure 6-8 and figure 6-7 shows mean and standard deviation PSNR results. The left plot of figure 6-7 shows that the proposed generative regularisers do better than unlearned TV reconstruction for the mid-range noise levels. Indeed we can see in figure 6-8, that in the third row, while the TV reconstruction still has blob-like artefacts, all three generators produce good reconstructions. The right plot, of figure 6-7, compares the generators trained on the different types of data and notes that the VAE-GT and noisyVAE seem to perform better than VAE-TV for all but the highest noise levels. We anticipate that this is due to some blurring around the edges of the VAE-TV reconstructions. Figure 6-8, shows that in the very high noise cases, the standard VAE-GT struggles to find a point in the latent space to match the data, whereas the VAEs trained on noisy data, seem to perform visually better. We highlight the noise case that was seen in training for the VAE-TV and noisyVAE, in red in figure 6-8 and with a vertical grey line in figure 6-7 and note that there is no clear change in the behaviour of the regulariser around this point.

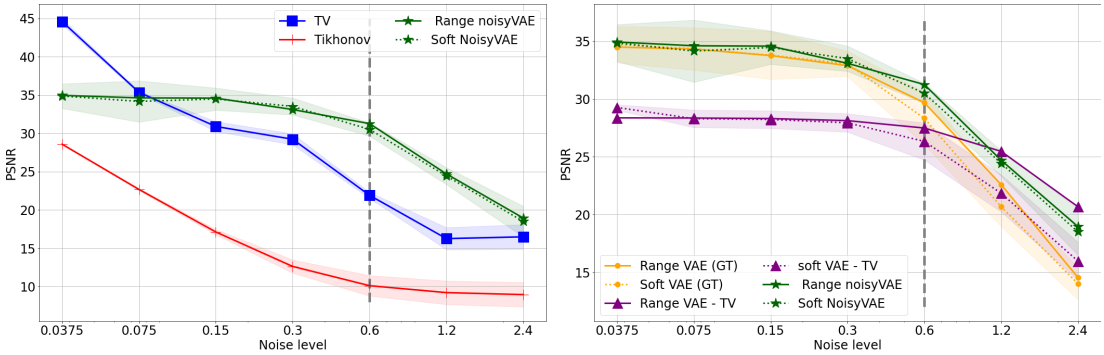


Figure 6-7: Reconstruction results for denoising on the *single ellipse* dataset. The x-axis gives amounts of added noise and reconstruction quality is assessed using PSNR. The bold lines denote the mean, and the shaded areas standard deviation of the *soft approach*, taken over 10 test images. The vertical grey line gives the noise level in the data used to train VAE-TV and VAE-GT.

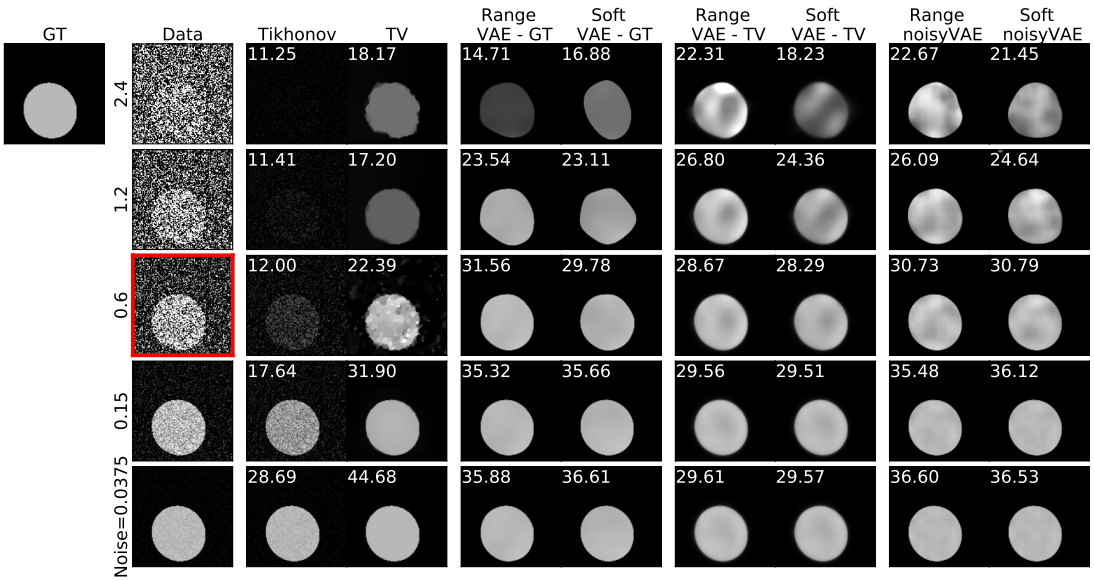


Figure 6-8: Reconstruction results for denoising on the *single ellipse* dataset. The rows give different amounts of added noise in the data. The columns give the ground truth image, the observed data, Tikhonov and TV reconstructions, reconstructions from a VAE trained on ground truth data, reconstructions from a VAE trained on TV reconstructed data and then the reconstructions from the proposed noisyVAE. The red boxes highlight the amount of noise in the data used to train VAE-TV and noisyVAE.

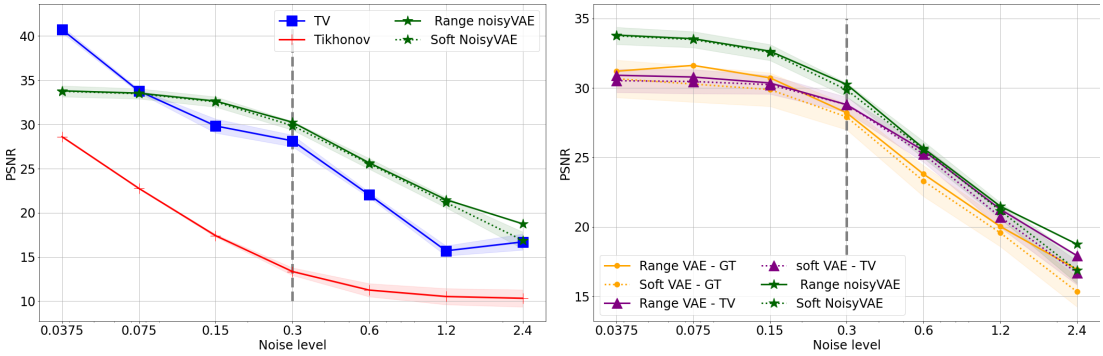


Figure 6-9: Reconstruction results for denoising on the *ellipses* dataset. The x-axis gives amounts of added noise and reconstruction quality is assessed using PSNR. The bold lines denote the mean, and the shaded areas standard deviation of the *soft* approach, taken over 10 test images. The grey line gives the noise level in the data used to train VAE-TV and VAE-GT.

***Ellipses* dataset** We consider a noisyVAE trained on observations of the *ellipses* dataset with 0.3 added noise. The VAE-TV model is trained on TV reconstructions from the same noisy datasets and again VAE-GT is trained on clean data. We then test their use as generative regularisers for the denoising inverse problem, for a range of noise levels. On the left of figure 6-9, you see again generative regularisers outperforming TV in the mid-range noise levels. On the right, you see that the generators trained on the different types of data behave very similarly and there doesn't seem to be a significant penalty when training with noisy data. Example images are given in figure 6-10 where you can see that, even with 0.6 added noise, a high noise case, the *soft* approach for the noisyVAE manages to reconstruct three ellipses and the green arrow indicates where it has managed to reconstruct a small overlapping triangle, where none of the other methods succeeded. The red arrow indicates where the noisyVAE has also managed to reconstruct a sharp outline successfully. Again the red box in figure 6-10 and the grey line in figure 6-9 indicate the noise level in the training data seen by the noisyVAE and VAE-TV but again there seems no indication of this point in the data.

Out of distribution testing on *ellipses* dataset Like the previous paragraph, consider a noisyVAE trained on observations of the *ellipses* dataset with 0.3 added noise. The VAE-TV model is trained on TV reconstructions from the same noisy datasets and again VAE-GT is trained on clean data.

We consider testing these learned generators on a dataset with a different range of ellipse sizes and shapes than those that produced the training data. In training, the ellipse axes were of length between 64-90 and 40-90 pixels and centred with

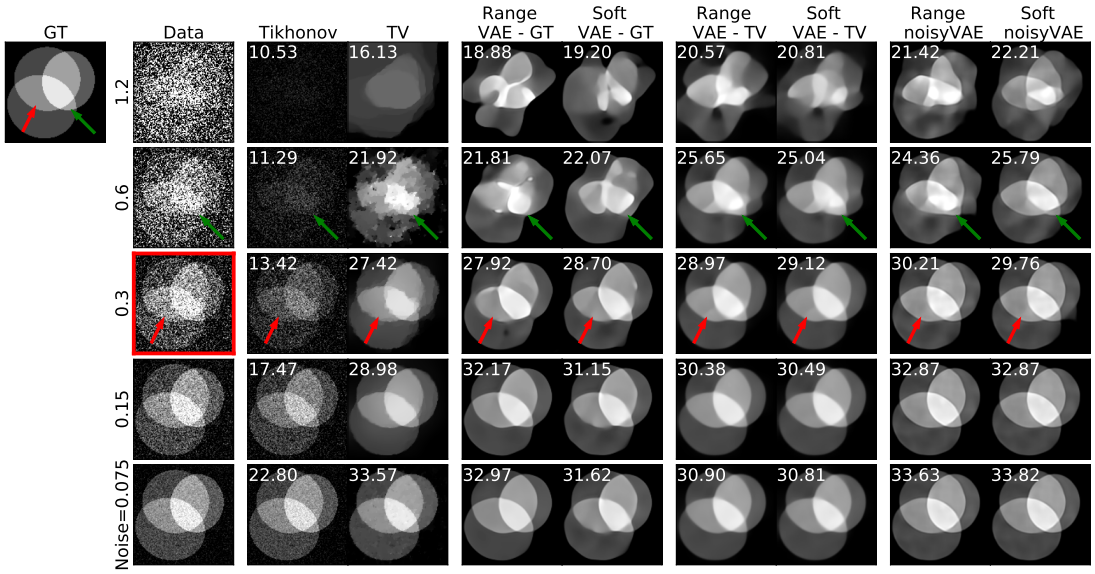


Figure 6-10: Reconstruction results for denoising on the *ellipses* dataset. The figure is set up as in figure 6-8

(x, y) coordinates taken between 40-90. For testing, we take ellipses with each axis between 40-80 and 13-40, centred with coordinates taken between 32-96. So the test ellipses are thinner and can be located in different places in the image. An example image is given in the ground truth of figure 6-8. The best example of the success of the generative regularisers, is the 4th row, with 0.15 noise, where the TV reconstruction has attempted to fill the area between the ellipse, but VAE models have better separation. In addition, as shown by the red arrow, the noisyVAE has some indication of the overlap, which is not seen in any of the other reconstructions.

In the high noise cases, all models struggle to reconstruct, although the VAE models attempt to fit ellipses, giving images with sharp boundaries and clear black backgrounds, for example, see the green arrow. If this were a medical imaging scenario, then further research into uncertainty quantification would be necessary. One avenue of research could consider looking at the latent space distribution. One might imagine that, for ground truth, x, not seen during training, the latent vector, z, required to ensure $G(z) \approx x$, could be far from the distribution over z used in training. Note, however, that this posterior distribution in z may not necessarily be the standard normal prior distribution.

PSNR results for this out-of-distribution test are given in figure 6-11. This is very similar to the results for the in-distribution test, figure 6-9, which suggests that these generative regularisers can generalise to this small shift in distribution.

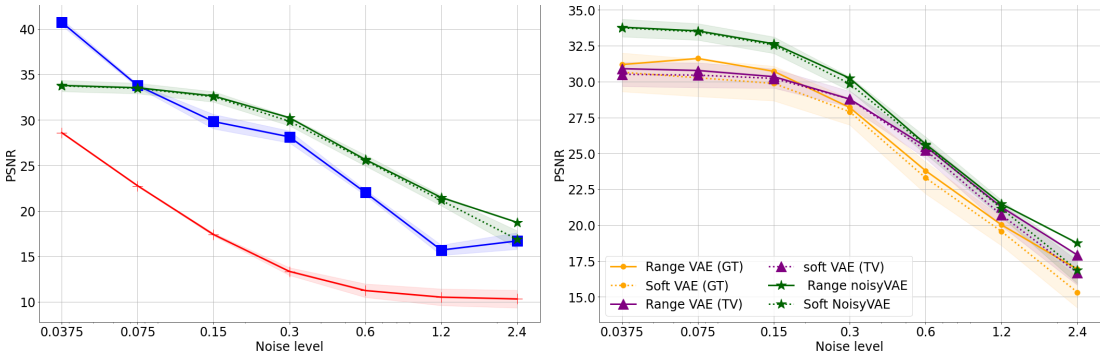


Figure 6-11: Reconstruction results for denoising on out of distribution *ellipses* dataset. The x-axis gives amounts of added noise and reconstruction quality is assessed using PSNR. The bold lines denote the mean, and the shaded areas standard deviation of the *soft* approach, taken over 10 test images.

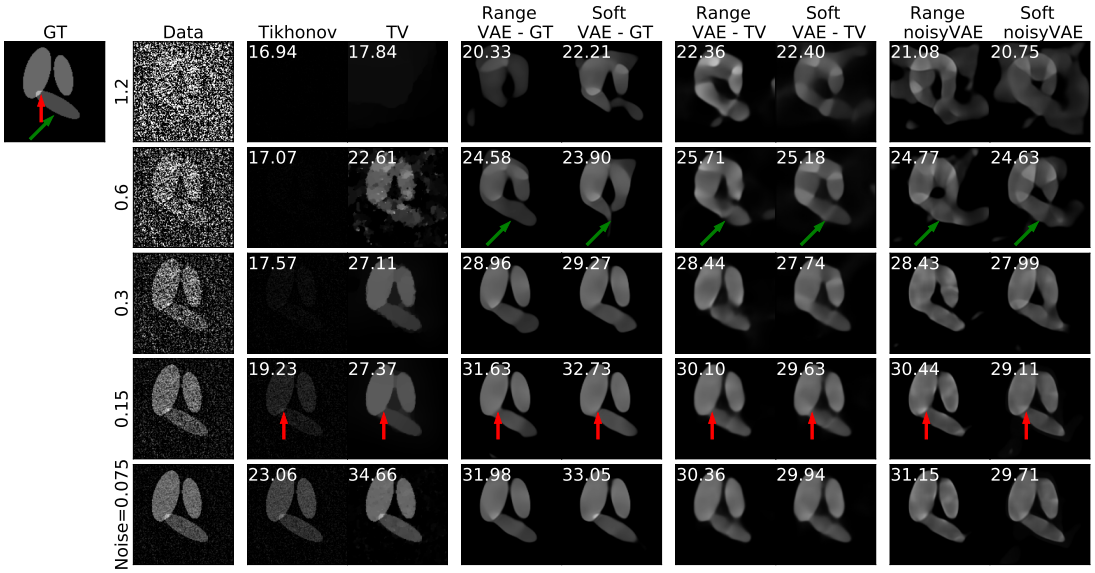


Figure 6-12: Reconstruction results for denoising on the out of distribution *ellipses* dataset. The figure is set up as in figure 6-8.

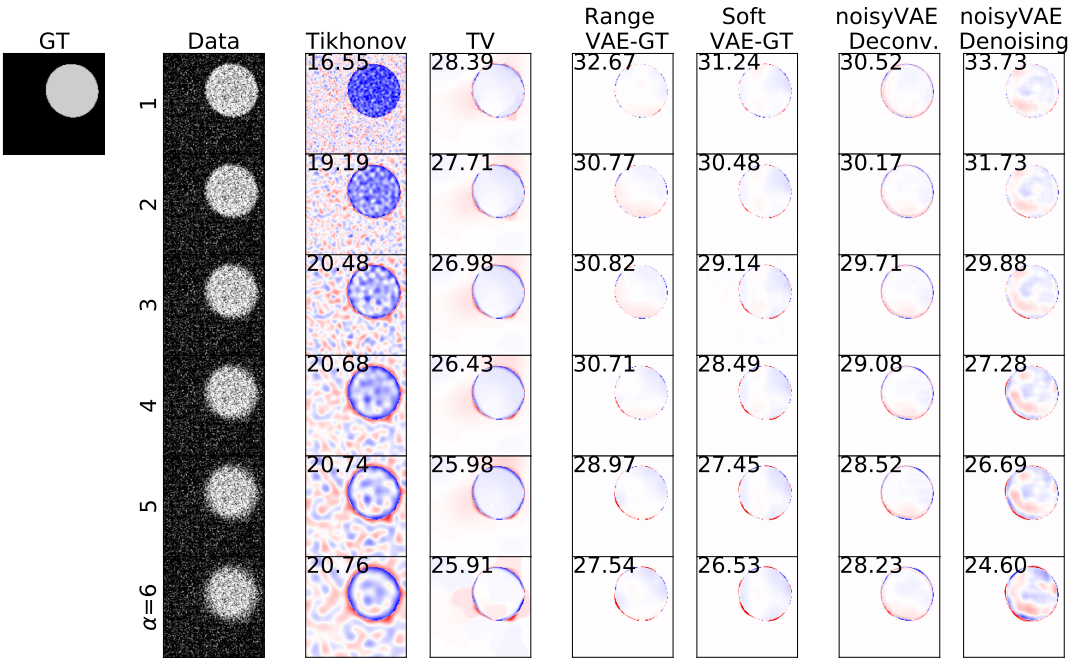


Figure 6-13: Reconstruction results for deconvolution on the *ellipses* dataset, for different scaling factors α in (see (6.17)).The plots show the reconstruction error. In the columns we show the ground truth image, the observed data, a Tikhonov and TV reconstruction, then a VAE trained on ground truth data for both the *range* and *soft* methods and finally the *soft* method for a noisyVAE trained on observed data from a deconvolution inverse problem ($\alpha = 3$, 0.1 added noise) and a noisyVAE trained on noisy data (0.6 added noise).

6.4.3 Deconvolution

Single Ellipse Dataset In figure 6-13, we demonstrate the success of two noisyVAEs on the deconvolution inverse problems. One model was trained on *single ellipse* denoising data with 0.6 added noise and the other was trained on Gaussian blurred data with 0.1 added noise and $\alpha = 3$. Each is applied as a regulariser for a convolution inverse problem, with varying scaling α in (6.17). The plot shows the reconstruction error and for this particular example, the *soft* approaches for all the VAEs behave similarly. The noisyVAE trained on the denoising problem has the most blob-like artefacts inside the ellipse, suggesting the model had more flexibility to overfit to the noise in the data, but produced good visual results. We have demonstrated how the noisyVAE is flexible to be applied as a generative regulariser for forward models not seen during training.

6.4.4 PET

In this final paragraph, we give preliminary results for PET reconstructions.

Two PET reconstruction examples are given in figure 6-14. We test two noisy-VAEs, one trained on 500,000 counts and the other on 1,000,000 counts, against a VAE trained on ground truth data and compare against Tikhonov and MLEM reconstructed images.

We can see that the *soft* approach for VAE-GT and noisyVAEs gives similar visual results to the MLEM iterations. However, it is clear from the *range* images that the generator is producing overly smoothed images without some of the detail seen in the ground truth images. The architecture is the same for all the VAE models and the current choice may just not be suited to the data.

6.5 Related Work

Deep Learning applied to inverse problems is a fast-growing field. See [16] or [168] for a review. Successful end-to-end approaches, such as [3] or [93], require large amounts of paired training data, and even unsupervised approaches, such as [26, 110, 146], require large amounts of high-quality ground truth data. As discussed, this data requirement can be challenging. If some data exists, there has been some previous work on using data augmentation to make the most of limited data [70, 120]. This can be challenging for inverse problems, as augmenting the image data also requires updating the corresponding measurement. In the case of no data at all, one can't pre-train a network. Instead, deep inverse priors [219], for example, train a convolutional neural network to output the solution to an inverse problem in response to measured data, using the network to implicitly regularise the solution. Similarly, CryoGANs take input projected measurements of the unknown object and use a discriminator to train a generator to output a reconstruction [89].

There are a variety of deep learning models for inverse problems that aim to denoise or remove artefacts and are trained without ground truth data. Noise2Noise [137] learns to denoise from a dataset consisting of two separate noise instances for each unknown ground truth and the methods. This is often not a realistic scenario and instead, Noise2Void[131] and Noise2Self[19] consider training from just one noise instance of each unknown but aren't able to handle correlated noise. Stein's Unbiased Risk Estimation provides another method for denoising without ground truth [207, 154]. The work of [43] applies Stein's Unbiased Risk Estimation to an ill-posed inverse problem, with a non-trivial forward operator, incorporating additional information in the form of image equivariances. Denoisers trained without ground truth can be used as part of iterative reconstruction schemes *e.g.*

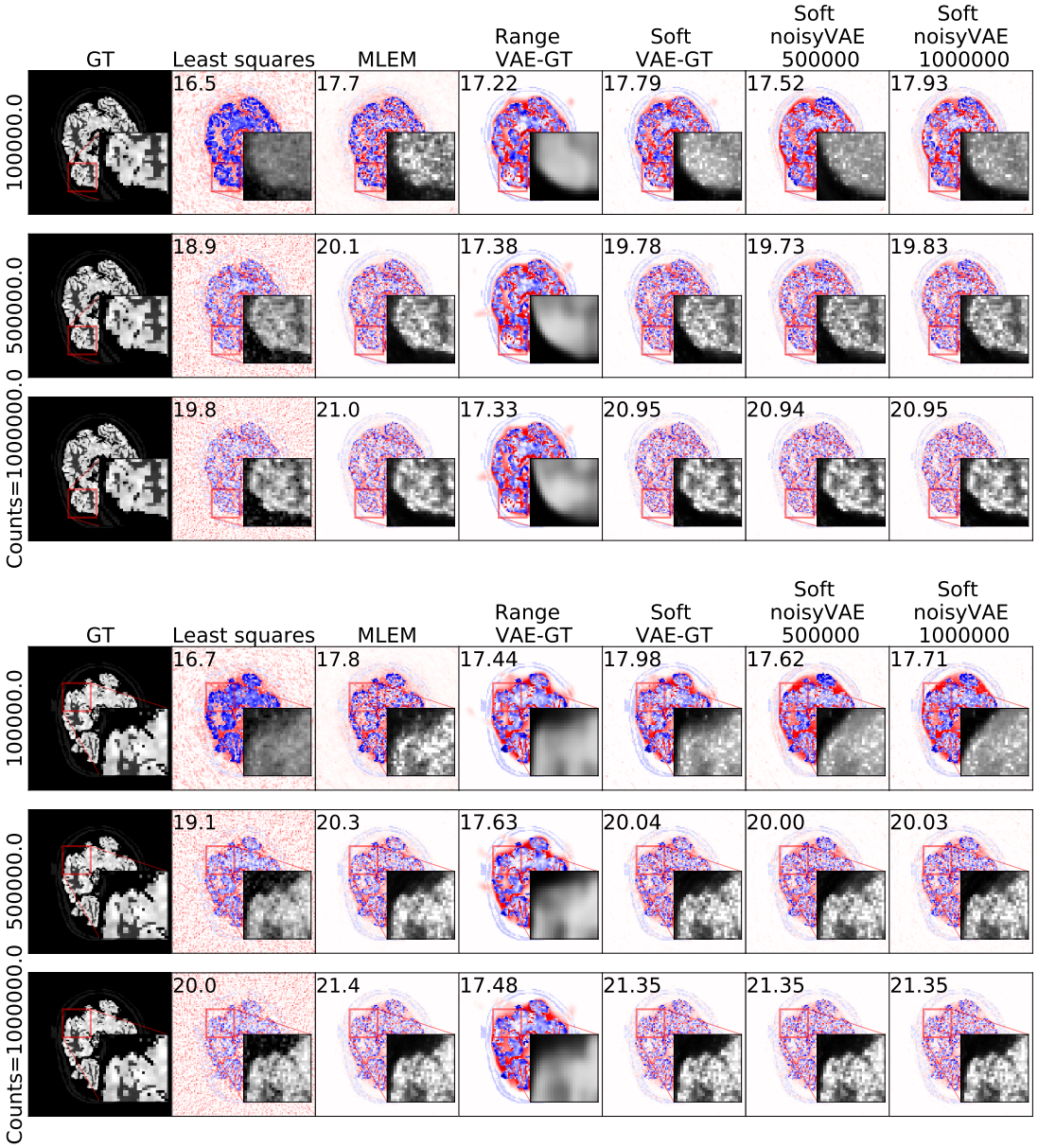


Figure 6-14: Two PET reconstruction examples. The background gives the loss and the zoomed-in region an indication of the reconstructed image. PSNR results are given in the top left-hand corner.

[245], plug and play methods (for a review see [7]) or as regularisers [143]. The Noise2Inverse method [98] deals with a non-trivial forward operator and learns to remove the effects of added noise in the observation domain, training without ground truth, however, is not designed to deal with ill-posedness or remove artefacts caused by the forward operator. The approach of [231] considers compressed sensing MRI and performs cross-validation using subsets of k-space data to learn a clean reconstruction from partial k-space observations.

In terms of learning generative models from observed data, ambientGAN[27] assumes the forward model and noise type is known. They learn generator from the latent to image space $G_\theta : \mathcal{Z} \to \mathcal{X}$, and treat this as a generator producing outputs on the observed data space $A \circ G_\theta : \mathcal{Z} \to \mathcal{Y}$. They train this generator by learning a discriminator $D_\phi : \mathcal{Y} \to \mathbb{R}$, acting in the observation space that differentiates between genuine observed data and observations of the results of the generator. They use the standard Wasserstein GAN loss (3.8). Several papers follow from this, using similar ideas. For example, MisGAN [142] focuses on inpainting but where the masks for producing the observed data are unknown. They learn a generator for both the images and the masks and also two discriminators: one acting on the masked images and another on the masks. Noise robust GANs [118] consider learning a generator from noisy data, where the noise model may be unknown and may be dependent on the image, learning a generator for both the image and the noise. A later paper [49] trains a conditional GAN, with a generator taking MRI k-space measurements and outputting a complex MRI image. The MRI forward problem is then applied to the image, with a random sampling mask and a discriminator comparing real and generated k-space data. The discriminator is being used as a tool for training an end-to-end method, the generator. All these approaches focus on GANs, which require saddle point optimisation and can be tricky to train and susceptible to mode collapse. For inverse problem applications, learned generators should be able to produce the whole range of possible image reconstructions.

One could also consider taking the AmbientGAN approach with the VAE, replacing the usual generator by $A \circ G_\theta$ and choosing an encoder that maps $\mathcal{Y}$ to $\mathcal{Z}$. However, in the VAE derivation, this would mean that we only have control of the distribution $p(y|z) = \mathcal{N}(y; A(G_\theta(z)), \Sigma_\theta(z))$ and not of the distribution we wish to learn, $p(x|z)$. There are a number of benefits of learning the latter. The first is that a distribution on $p(x|z)$ can be incorporated directly as a prior to the inverse problem reconstruction, in the Bayesian framework (6.16), giving an explainable regularisation function. The second is that we could include structured noise models, in the form of chapter 5. Finally, the third is that it is very flexible in combining different forward models and scenarios. For example, for an inverse problem scenario where different forward models produce measurements with different dimensions, such as inpainting with different image masks, forc-

ing the same noise model onto each measurement space might not make sense. Instead, by determining the noise model we are fitting, in the image space, the forward operator is free to vary, and information from different sources can be combined.

There has been previous work in incorporating generators as part of regularisers for inverse problems, for example [26, 57, 90, 218, 64]). These all took a two-step approach to solve inverse problems: first training a generator on ground truth data and then using the generator to regularise an inverse problem. Any generator could plug into this framework, including one trained on noisy partial measurements.

6.6 Conclusions

In this work, we have introduced a noisyVAE, a variation on the VAE objective that learns a generator outputting images, from observed data and the forward model that produces the observations. This is designed to deal with the case where high-quality or ground-truth images are not available.

We demonstrated the generation ability of noisyVAEs learned from noisy and convolved data and demonstrated their ability to reconstruct example ground truth images from a test dataset of ellipses. We were able to show that these learned generators were effective as generative regularisers for inverse problems both seen (figures 6-10 and 6-8) and unseen (figure 6-13) during training and flexible enough to deal with a small shift in test image distribution (figure 6-12).

The noisyVAE framework is flexible to different noise models in the observed data, including Poisson noise for PET imaging. Preliminary results show that learning a VAE from PET data is possible, but further work is required to improve the detail in the generated images and thus provide a more effective inverse problem prior.

6.7 Future Work

Numerical experiments and evaluation There are more numerical experiments and evaluation methods that we wish to investigate.

The flexibility of the set-up in (6.8) is that we could incorporate information from a variety of sources together, making use of the option of different forward models. For example, information could be obtained from different MRI scanners, with different sensitivities. We could also envision cases where there are a few high-quality reconstructions available, where the forward model is the identity,

alongside measured data, with non-trivial forward operators. Future work could investigate scenarios like this.

This chapter wished to address cases where ground truth or high-quality reconstructed data is difficult or expensive to obtain. The models that we train require just observed data and the corresponding forward model, removing the need for ground truth data. However, even measured data can be difficult to obtain, and it would be interesting to run experiments changing the amount of data used to train the generators and investigating the resulting properties of the learned generators.

Finally, the PET experiments given in this paper were preliminary and, as discussed, with a systematic architecture search, we would expect to get better results from all VAE models. In addition, we simulated PET data from ground truth images that were themselves simulated from MRI phantoms. To bring these experiments closer to reality, an obvious next set is to find suitable real-world datasets. PET scans are usually three-dimensional and this would be an interesting challenge for our generative models. Another useful qualitative experiment would be to test the ability of the generative regularisers to reconstruct abnormal lesions, not seen in the training data.

Extending noisyVAEs In the current formulation, we utilise a rough reconstruction to transform the observed data from the measurement space to the image space. For different forward models, this rough reconstruction could vary. We then consider just one encoder that takes these rough reconstructions as inputs in the image space and outputs a latent vector encoding. When data is gathered from a variety of sources (forward models), it may be beneficial to train a separate encoder for each forward model. For example, for a convolution forward problem, with different levels of blurring, the rough reconstruction may have different noise statistics and reconstruction artefacts and asking one encoder to deal with all cases may not be optimal. Future work could also consider the encoder(s) taking measurement data as an input. We choose encoders acting on the image space because architectures acting on images, such as CNNs, are reasonably well understood. If we were to define an encoder going from the measurement space, it could be that different architectures are more suitable. For example, if the forward model measures a physical quantity at a set of points in a domain, then those points may not be uniformly spaced and so one might want to consider architectures that could take into account the physical locations of the sensors.

Finally, allowing $\Sigma_\theta(z)$ to be non-trivial could allow greater flexibility of our models and a more effective prior, as seen in chapter 5. As discussed in 5, in order to parametrise a dense $\Sigma_\theta(z)$ in a computationally efficient way, we param-

eterised a sparse Cholesky decomposition of the precision matrix, $L_\theta(z)L_\theta(z)^T = \Sigma_\theta^{-1}(z)$. In the noisyVAE objective (6.8), $\Sigma_\theta(z)$ only appears in the expectation $\mathbb{E}_{x\sim N(x;G_\theta(z),\Sigma_\theta(z))}$. Note that sampling from $P_{G,\Sigma}(x|z)$ is possible through $x = G_\theta(z) + (L_\theta(z)^T)^{-1}u$ where $u \sim \mathcal{N}(0, I)$ and as $L_\theta(z)$ is lower triangular this should be computationally efficient. However, consider that in chapter 5 we were concerned about the diagonals of $L_\theta(z)$ becoming infinite, and thus capped the values of the diagonal, in (6.8) we are more likely to be concerned about $L_\theta(z)$ collapsing to zero and thus the discussions in section 5.B on priors will be important.

Chapter 7

Conclusions and Future Work

In this final chapter, we return to the four challenge areas identified in the introduction and also consider interesting directions for future work.

7.1 Summary

Returning to the challenge areas identified in the introduction, we summarise both the work presented in this thesis and our contributions to the challenge areas.

Data and training requirements The requirement for large amounts of high-quality paired training data needed for supervised learning can be punitive in regimes where data is expensive or impossible to obtain. In the supervised learning scenario, changes in the forward model can require re-training with new data. In chapter 4 and chapter 5 we focus on the case where we have a large amount of high-quality reconstructed images or ground truth images. We train independently of the forward operator and don't require paired training data. This means that the model remains flexible to changes in the forward model, for example changing the MRI sampling pattern in chapter 5, and extensive retraining is not required. A valid criticism of this approach is that we could easily create a dataset of paired training data, in order to train in a supervised manner, using the ground truth data and a known forward operator. In chapter 6, we consider the case where we don't have access to any high-quality reconstructed images and instead have just a dataset of observed data and the forward model that relates the two. We show that we are able to learn generative regularisers with similar properties and behaviour as those trained on ground truth data.

Explainability Trustworthiness is crucial for safety-critical applications, such as medical imaging. In chapter 4, we set out a set of desired criteria in order to evaluate generative models for use in inverse problems. We hope that these could guide future research into generative models for inverse problems and therefore produce better and more trustworthy generative regularisers. One benefit of using generative models as a prior to inverse problems is that by generating from them, one can explicitly visualise the prior it places on the inverse problem. For the MNIST digits in chapter 4, we use generative models to reconstruct example test images, and we can see that the VAE models can reconstruct the structure of the digit with some blurriness. Going on to use this model as a regulariser, one can be aware that the regulariser has a tendency to produce blurry images. In chapter 5, we go one step further and visualise the learned dependencies between pixels generated from a VAE with structured covariance. Thus, as well as being able to explicitly visualise images favoured by a prior, for each pair of pixels in that generated images, we were able to see the learned correlations between them. Throughout, we consider generative regularisers under the framework of variational regularisation, chapter 4, and Bayesian inverse problems, chapters 5 and 6. Both frameworks balance information from the prior and observed data and incorporate the physical information provided by the forward model. In chapter 4, we visualise this trade off, showing the data discrepancy loss and example images for a range of regularisation parameters.

Theoretical guarantees Of the four key areas, this area is the least explored in this thesis. In chapter 2 we introduced deep learning models and discussed how training a neural network is itself an ill-posed inverse problem. for generative regularisers, the objective, see *e.g.* (1.2), is non-convex with potentially many local minima. Theoretical guarantees are difficult. However, in chapter 4, we highlighted some areas where theoretical results exist and we have identified potential future work in this direction.

Generalisability Lastly, it is important to ensure that our proposed algorithms and regularisers remain flexible and are able, to some extent, to be effective outside of the datasets used during training. By keeping most of the training of our generative models unsupervised, without knowledge of the forward operator, we demonstrated in chapter 5, how the model can remain flexible to some changes in the forward model. In chapter 4 and 5 we discussed how restricting solutions to lie exactly in the range of the generator, is limited by the quality of the generator which is determined by the training data, the expressiveness of the generator, and the success of the training procedure. Allowing small deviations in the 2-norm, 1-norm or TV norm (chapter 4) or measured by some learned weighted norm (chapter 5) allows more flexibility in the prior. In chapter 6, we tested models trained on data for one inverse problem applied as a genera-

tive regulariser for a different problem. In chapters 4 and 6, we also consider out-of-distribution testing on data not seen during training.

7.2 Future Work

Generative models One of the visions of chapter 4 was to answer the question "if a colleague sends me a generative model, how can I determine its suitability for use in inverse problems?". We were able to give some 'must have' criteria, for example, the generative model should be able to reconstruct a range of possible solutions to the inverse problem (A1), and it must be differentiable (B1), for use in gradient-based optimisation schemes. These criteria are neither complete, they might all be satisfied but a good reconstruction is not guaranteed, or mathematically precise enough to test definitively. Future work in formulating these criteria would be beneficial to the field. For example, with precise criteria, we could consider designing generative models with inverse problems in mind. For instance, if the criteria set out that the network should be able to interpolate between points in the latent space, with all intermediate images being realistic, then this could be added to the loss function of a GAN, with the discriminator assessing the intermediate images. Similarly, if it was required that the range of the generator should be convex, then this might also be ensured by a suitable choice of the generative model architecture.

One aspect of VAEs and GANs that we would have liked to have researched in more detail, is the role of the latent space and, for VAEs, the encoder. In chapter 4, we consider a range of latent dimensions but found no real difference between different choices. In chapters 5 and 6, for the knee MRI and brain PET images, we kept the latent dimension relatively small at just 100 for 128×128 images. This was for several reasons, the first was that it seemed to help with the structured covariance network training. The second was that we found that the samples taken from random points in the latent space were of better quality and it was easier to find points in the latent space that matched test images. It is not quite so straightforward to consider that a higher latent dimension for a VAE means less of a network bottleneck and thus and a more expressive generator because often the encoder and decoder networks will also have a regularising effect. Similarly, a smaller latent dimension may still be able to produce impressive samples *e.g.* if each training image was encoded and decoded from a unique scalar-valued range, the latent dimension could be as small as one. We also mention, *e.g.* in chapter 4, that the prior on $\mathcal{Z}$ we use in the Bayesian formulation of the generative regularisers (6.12) might not be an accurate reflection of the post-training distribution in the latent space that maps to desired images. In general, we found that varying the regularisation parameter in front of this term had little difference, as long as it wasn't too large. Perhaps with a more accurate

post-training distribution, this term would have more of an effect and we could try larger latent dimensions.

As discussed in chapter 3, during the period of this PhD research several new generative models have emerged, including Normalising Flows and Diffusions/Score Based generative models. They differ from the AEs, VAEs and GANs that we have primarily focused on as their latent and generated space have equal dimensions, removing the low dimensional manifold assumption. With this comes potential benefits, for example, invertible generators allow for a well-defined and tractable probability density function over the image domain. This allows, for example, the learned probability of an image to be used directly as a prior for the inverse problem. However, it could also come with potential drawbacks, such as the instabilities of invertible neural networks. Additionally, in the past few years, we have also seen new architectures emerge, such as transformers [221] which were originally introduced for text-to-text challenges, such as translation, but have fast become crucial in large image models [68]. New architectures are interesting because of the implicit priors they place on networks and may lead to generative regularisers with different behaviours. Future work could consider incorporating these architectures and generators into a generative regularisation framework and analysing their behaviour.

Geometric machine learning [32] is a fast-growing research area that includes the study of data and machine learning from the perspectives of symmetry and invariance. In the context of this thesis, we could ask that our generator and hence generative regulariser had some desired symmetry. If there is no prior knowledge of the orientation of an object we wish to reconstruct, then we might wish that the generative model is somehow invariant to rotations, it produces all possible orientations with equal probability. Equivariant neural networks can have equivariance or invariance properties built-in [38]. Other symmetries could be considered *e.g.* scale and contrast. On the other hand, a dark spot on the lung might be much more significant than a dark spot in the background of a chest CT scan, and we do not want our network to be translationally invariant, at least globally. Indeed, in chapter 5 we explicitly allowed the loss function in our generative regulariser to vary across an image. When considering learning generative models effectively, perhaps with limited data, then explicitly building networks with some symmetry, might be an interesting avenue for future work.

Geometric machine learning could also open up new inverse problems over non-Euclidean data structures. We view images as Euclidean, as 2 or 3-dimensional arrays of equal-sized square pixels, but often we are imaging or measuring over non-Euclidean domains. For example, meshes of electrodes are used to measure the electrical activity of the heart (electrocardiogram) or the brain (electroencephalogram) and are often considered as just a vector or grid at a given time point, rather than taking into account the spatial information. Graph convolu-

tional networks open up avenues for taking into account more of the geometric information *e.g.* [112, 111] and it could be possible to design generators suited to non-Euclidean inverse problems.

NoisyVAEs In the conclusion of chapter 6, we set out a range of further work for investigating the success of the noisyVAE objective for learning to generate without access to ground truth data. This included looking into incorporating information measured from different forward models, experimenting with the architecture of the encoder and assessing the amount of training data required to still give good generated images. Positron Emission Tomography (PET) is an important imaging modality that suffers from high noise and thus high-quality reconstructed images are difficult and expensive to obtain. Future work could consider a detailed architecture search to find a generator suited for PET images and assessing noisyVAEs on real-world PET datasets.

Structured Covariance Models As discussed in the summary and conclusions of chapter 5, although the structured covariance model learned the correlations within the images and was able to be used as an effective prior, training was challenging. The trained model that we used for the numerical results was too permissive, applying higher variances across the whole image and we would hope to do better. As discussed in 5.B, future work in this direction could consider priors on the structured covariance, which could be computationally challenging, or further investigations to determine the best architecture and training regime for the network. To make the MRI images scenario used in chapter 5 more realistic, complex valued images with could be considered, and modelling the correlations between real and imaginary parts, would be an interesting avenue for future work.

Another future research direction is to combine the power and flexibility of the learned covariance networks of chapter 5 with the generative networks learned from noisy or partially observed data in chapter 6. The formulation of the noisyVAE was chosen with this in mind, and it would be interesting to see what correlations and structure can be learned from only observed data.

Generative regularisers In chapter 4, we discuss how generative regularisers are not a convergent regularisation method. At the end of the chapter, we then go on to discuss directions for future theoretical research, including considering what the minimum properties of the generator are required to give a convergent regularisation method. The non-linear nature of the generator might also suggest looking at the non-linear inverse problem literature.

Practically, there are also a few directions for future work. The first considers uncertainty quantification. As seen in some of the inverse problems in chapter 6, where the observed data is poor, the generative regularisers can still output

'confident' results *i.e.* clear plausible images. In this case, you might expect that the data discrepancy term is still high, suggesting that the results do not fit the observed data well. However, there may be other avenues for exploring uncertainty. One could consider the latent encoding and compare the optimised z with a post-training learned distribution over the latent space. We would anticipate that images far from the training distribution will require latent vectors from obscure parts of the latent space. Another avenue for uncertainty quantification could be in multiple sampling from the generative model, perhaps around a found latent vector, or by initialising optimisation schemes at multiple locations. This fits well with our discussion on 'explainability'.

The second considers the regularisation parameters. For generative regularisers (1.2), the regularisation parameter λ is crucial for determining how strictly solutions are restricted to be near the range of the generator. A small regularisation parameter leads to little or no regularisation and high values restrict solutions to be in the range of the generator. The Bayesian framework suggests that if the noise level in the observed data is known, then these regularisation parameters should also be known, provided the probabilistic modelling is sufficiently accurate. Future work could consider different strategies for optimally setting these regularisation parameters or if, with improved structured covariance models, how they could be removed.

Consider an inverse problem reconstructed with TV regularisation and that for large regularisation parameters, blob-like cartoon images are obtained, but, for small parameters, more natural images are possible. The regularisation parameter can be tuned in response to different problems or scenarios. If one just needed the rough structure of a knee from MRI data to perform a measurement or segmentation a large regularisation parameter and a block-like reconstruction would be sufficient. However, if one needed to look carefully and a small structure, a more detailed image is required. It might be interesting to consider how different generators could be combined to produce a similar effect, for example, one could have a patch-based generator trained to produce texture and a global generator trained to produce image structure and combining the two with different weights could give different results depending on need.

Bibliography

[1] Jonas Adler, Holger Kohr, and Ozan Öktem. *Operator Discretization Library (ODL)*. 2017.

[2] Jonas Adler and Ozan Öktem. "Deep Bayesian Inversion". In: *ArXiv Preprint* (2018).

[3] Jonas Adler and Ozan Öktem. "Learned Primal-Dual Reconstruction". In: *IEEE Transactions on Medical Imaging* 37 (6 2018), pp. 1322–1332.

[4] Jonas Adler and Ozan Öktem. "Solving ill-posed inverse problems using iterative deep neural networks". In: *Inverse Problems* 33 (12 2017), p. 124007.

[5] Jonas Adler and Sebastian Lunz. "Banach Wasserstein GaN". In: *NeurIPS* (2018).

[6] Michal Aharon, Michael Elad, and Alfred Bruckstein. "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation". In: *IEEE Transactions on Signal Processing* 54 (11 2006), pp. 4311–4322.

[7] Rizwan Ahmad, Charles A Bouman, Gregery T Buzzard, Stanley Chan, Sizhuo Liu, Edward T Reehorst, and Philip Schniter. "Plug-and-Play Methods for Magnetic Resonance Imaging: Using Denoisers for Image Recovery". In: *IEEE Signal Processing Magazine* 37 (1 2020), pp. 105–116.

[8] Kazunori Akiyama, Antxon Alberdi, Walter Alef, Keiichi Asada, Rebecca Azulay, Anne-Kathrin Baczko, David Ball, Mislav Baloković, John Barrett, and Dan Bintley. "First M87 event horizon telescope results. IV. Imaging the central supermassive black hole". In: *The Astrophysical Journal Letters* 875 (1 2019), p. L4.

[9] *Analyze Results: inverse problems deep learning (Topic)*. URL: `https://www.webofscience.com/wos/woscc/analyze-results/ef71ea9e-ade2-4dd7-acc6-ded622d642e0-50b441e3`.

[10] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent". In: *NeurIPS* 29 (2016).

[11] Rushil Anirudh, Jayaraman J. Thiagarajan, Bhavya Kailkhura, and Timo Bremer. "MimicGAN: Robust Projection onto Image Manifolds with Corruption Mimicking," in: *IJCV* (2020).

[12] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C. Hansen. "On instabilities of deep learning in image reconstruction and the potential costs of AI". In: *Proceedings of the National Academy of Sciences* (2020), p. 201907377.

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *ICML* (2017), pp. 298–321.

[14] Samuel G. Armato et al. "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans". In: *Medical Physics* 38 (2 2011), pp. 915–931.

[15] Sanjeev Arora, Andrej Risteski, and Yi Zhang. "Do GANs Learn the Distribution? Some Theory and Empirics". In: *ICLR* (2018), pp. 1–16.

[16] Simon Arridge, Peter Maass, Ozan Öktem, and Carola Bibiane Schönlieb. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174.

[17] Muhammad Asim, Fahad Shamshad, and Ali Ahmed. "Blind Image Deconvolution Using Deep Generative Priors". In: *IEEE Transactions on Computational Imaging* 6 (2020).

[18] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. "Deep equilibrium models". In: *NeurIPS* 32 (2019).

[19] Joshua Batson and Loic Royer. "Noise2self: Blind denoising by self-supervision". In: *ICML* (2019), pp. 524–533.

[20] Matthias Bauer and Andriy Mnih. "Resampled priors for variational autoencoders". In: PMLR, 2020, pp. 66–75.

[21] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. "Understanding and Mitigating Exploding Inverses in Invertible Neural Networks". In: *AISTATS, PMLR* 130 (2021), pp. 1792–1800.

[22] Adi Ben-Israel and Thomas N E Greville. *Generalized inverses: theory and applications*. Vol. 15. Springer Science Business Media, 2003.

[23] Martin Benning and Martin Burger. "Modern regularization methods for inverse problems". In: *Acta Numerica* 27 (27 2018), pp. 1–111.

[24] Yoeri E. Boink and Christoph Brune. "Learned SVD: solving inverse problems via hybrid autoencoding". In: *ArXiv Preprint* (2019).

[25] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146 (1-2 2014), pp. 459–494.

[26] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. "Compressed sensing using generative models". In: *ICML* (2017), pp. 822–841.

[27] Ashish Bora, Eric Price, and Alexandros G Dimakis. "AmbientGAN: Generative models from lossy measurements". In: *ICLR* (2018).

[28] Ali Borji. "Pros and cons of GAN evaluation measures". In: *Computer Vision and Image Understanding* 179 (2019), pp. 41–65.

[29] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine learning* 3 (1 2011), pp. 1–122.

[30] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. 2004.

[31] Kristian Bredies and Dirk Lorenz. *Mathematical image processing*. Springer International Publishing, 2018, pp. 1–469.

[32] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges". In: *ArXiv Preprint* (2021).

[33] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. "Language models are few-shot learners". In: *NeurIPS* 33 (2020), pp. 1877–1901.

[34] Leon Bungert and Matthias J. Ehrhardt. "Robust Image Reconstruction with Misaligned Structural Information". In: *IEEE Access* 8 (2020), pp. 222944–222955.

[35] E Candes, David L. Donoho, Emmanuel J. Candès, and David L. Donoho. "Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges". In: *Curves and Surface Fitting* C (2 2000).

[36] Emmanuel J. Candes and Terence Tao. "Decoding by linear programming". In: *IEEE Transactions on Information Theory* 51 (12 2005), pp. 4203–4215.

[37] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. "Stable signal recovery from incomplete and inaccurate measurements". In: *Communications on Pure and Applied Mathematics* 59 (8 2006), pp. 1207–1223.

[38] Elena Celledoni, Matthias J Ehrhardt, Christian Etmann, Brynjulf Owren, Carola-Bibiane Schönlieb, and Ferdia Sherry. "Equivariant neural networks for inverse problems". In: *Inverse Problems* 37 (8 2021), p. 85006.

[39] Yair Censor. *The mathematics of computerized tomography*. Vol. 18. SIAM, 2002, p. 283.

[40] Antonin Chambolle and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of Mathematical Imaging and Vision* 40 (1 2011), pp. 120–145.

[41] Paramanand Chandramouli, Kanchana Vaishnavi Gandikota, Andreas Goerlitz, Andreas Kolb, and Michael Moeller. "Generative Models for Generic Light Field Reconstruction". In: *TPAMI* (2020).

[42] Caroline Chaux, Patrick L. Combettes, Jean Christophe Pesquet, and Valérie R. Wajs. "A variational formulation for frame-based inverse problems". In: *Inverse Problems* 23 (4 2007).

[43] Dongdong Chen, Julián Tachella, and Mike E Davies. "Robust Equivariant Imaging: a fully unsupervised framework for learning to image from noisy and partial measurements". In: *CVPR* (2022), pp. 5647–5656.

[44] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. "Low-dose CT with a residual encoder-decoder convolutional neural network". In: *IEEE transactions on medical imaging* 36 (12 2017), pp. 2524–2535.

[45] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. "Generative pretraining from pixels". In: PMLR, 2020, pp. 1691–1703.

[46] A. Clark. *Pillow (Python Imaging Library Fork)*. 2015.

[47] Chris A Cocosco, Vasken Kollokian, Remi K.-S. Kwan, and Alan C Evans. "BrainWeb: Online Interface to a 3D MRI Simulated Brain Database". In: *NeuroImage* (1997).

[48] Matthew J Colbrook, Vegard Antun, and Anders C Hansen. "The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and Smale's 18th problem". In: *Proceedings of the National Academy of Sciences* 119 (12 2022), p. 1119.

[49] Elizabeth K Cole, Frank Ong, Shreyas S Vasanawala, and John M Pauly. "Fast unsupervised MRI reconstruction without fully-sampled ground truth data using generative adversarial networks". In: *ICCV* (2021), pp. 3988–3997.

[50] Casper da Costa-Luis. *PET-MR - BrainWeb-based multimodal models of 20 normal brains*. 2020. URL: `https://pet-mr.github.io/brainweb/`.

[51] Caroline Crockett and Jeffrey A Fessler. "Bilevel Methods for Image Reconstruction". In: *Foundations and Trends in Signal Processing* 15 (2-3 2022), pp. 121–289.

[52] Bin Dai and David Wipf. "Diagnosing and enhancing VAE models". In: *ICLR* (2019).

[53] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G. Dimakis. "Intermediate Layer Optimization for Inverse Problems using Deep Generative Models". In: *ICML* (2021).

[54] Constantinos Daskalakis, Dhruv Rohatgi, and Manolis Zampetakis. "Constant-expansion suffices for compressed sensing with generative priors". In: *NeurIPS* (2020).

[55] Mark A. Davenport, Marco F. Duarte, Yonina C. Eldar, and Gitta Kutyniok. *Introduction to compressed sensing*. Ed. by Y Eldar and G Kutyniok. 2012.

[56] Chen Debao. “Degree of approximation by superpositions of a sigmoidal function”. In: *Approximation Theory and its Applications* 9 (3 1993), pp. 17–28.

[57] Manik Dhar, Aditya Grover, and Stefano Ermon. “Modeling Sparse Deviations for Compressed Sensing using Generative Models”. In: *ICML* 3 (2018), pp. 1990–2005.

[58] Laurent Dinh, David Krueger, and Yoshua Bengio. “Nice: Non-linear independent components estimation”. In: *arXiv preprint arXiv:1410.8516* (2014).

[59] Sören Dittmer, Tobias Kluth, Peter Maass, and Daniel Otero Baguer. “Regularization by Architecture: A Deep Prior Approach for Inverse Problems”. In: *Journal of Mathematical Imaging and Vision* 62 (3 2020), pp. 456–470.

[60] David L. Donoho. “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52 (4 2006), pp. 1289–1306.

[61] Garoe Dorta. “Learning models for intelligent photo editing”. In: *Student Thesis - Doctor of Engineering (EngD)* (2020).

[62] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill D F Campbell, and Ivor Simpson. “Training vaes under structured residuals”. In: *arXiv preprint arXiv:1804.01050* (2018).

[63] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill D.F. Campbell, and Ivor Simpson. “Structured Uncertainty Prediction Networks”. In: *CVPR* (2018), pp. 5477–5485.

[64] Margaret Duff, Neill D. F. Campbell, and Matthias J. Ehrhardt. “Regularising Inverse Problems with Generative Machine Learning Models”. In: *ArXiv Preprint* (2021).

[65] Margaret Duff, Ivor J. A. Simpson, Matthias J. Ehrhardt, and Neill D. F. Campbell. “Compressed Sensing MRI Reconstruction Regularized by VAEs with Structured Image Covariance”. In: (2022).

[66] Rick Durrett. *Measure Theory*. Springer, 2012, pp. 1–40.

[67] Matthias J. Ehrhardt, Pawel Markiewicz, and Carola Bibiane Schönlieb. “Faster PET reconstruction with non-smooth priors by randomization and preconditioning”. In: *Physics in Medicine and Biology* 64 (22 2019).

[68] Patrick Esser, Robin Rombach, and Bjorn Ommer. “Taming transformers for high-resolution image synthesis”. In: *CVPR* (2021), pp. 12873–12883.

[69] K Eykholt, I Evtimov, E Fernandes, B Li, A Rahmati, C Xiao, A Prakash, T Kohno, and D Song. “Robust Physical-World Attacks on Deep Learning Visual Classification”. In: *CVPR* (2018), pp. 1625–1634.

[70] Zalan Fabian, Reinhard Heckel, and Mahdi Soltanolkotabi. “Data augmentation for deep learning based accelerated MRI reconstruction with limited data”. In: PMLR, 2021, pp. 3057–3067.

[71] Qiuyun Fan et al. “MGH-USC Human Connectome Project datasets with ultra-high b-value diffusion MRI.” In: *NeuroImage* 124 (Pt B 2016), pp. 1108–1114.

[72] Rémi Flamary and Nicolas Courty. *POT python optimal transport library.* 2017.

[73] Ken-Ichi Funahashi. “On the approximate realization of continuous mappings by neural networks”. In: *Neural Networks* 2 (3 1989), pp. 183–192.

[74] Aude Genevay, Gabriel Peyré, and Marco Cuturi. “GAN and VAE from an Optimal Transport Point of View”. In: *ArXiv Preprint* (2017).

[75] Martin Genzel, Jan Macdonald, and Maximilian Marz. “Solving Inverse Problems With Deep Neural Networks - Robustness Included”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[76] Davis Gilton, Greg Ongie, and Rebecca Willett. “Deep Equilibrium Architectures for Inverse Problems in Imaging”. In: *IEEE Transactions on Computational Imaging* (2021).

[77] Micah Goldblum, Jonas Geiping, Avi Schwarzschild, Michael Moeller, and Tom Goldstein. “Truth or Backpropaganda? An Empirical Investigation of Deep Learning Theory”. In: *ICLR* (2020).

[78] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. “The reversible residual network: Backpropagation without storing activations”. In: *NeurIPS* 30 (2017).

[79] Mario González, Andrés Almansa, Mauricio Delbracio, Pablo Musé, and Pauline Tan. “Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior”. In: *SIAM Journal on Imaging Science* 15 (2 2022), pp. 822–859.

[80] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[81] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets”. In: *NeurIPS* (2014), pp. 2672–2680.

[82] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[83] GoogleResearch. *TensorFlow: Large-scale machine learning on heterogeneous systems.* Software available from tensorflow.org. 2015.

[84] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Alexander Smola, Bernhard Schölkopf, and Alexander Smola GRETTON. “A Kernel Two-Sample Test”. In: *JMLR* 13 (2012), pp. 723–773.

[85] Jinjin Gu, Yujun Shen, and Bolei Zhou. “Image processing using multi-code GaN prior”. In: *CVPR* (2020), pp. 3009–3018.

[86] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. "Improved training of wasserstein GANs". In: *NeurIPS* (2017), pp. 5768–5778.

[87] Sean Gunn, Jorio Cocola, and Paul Hand. "Regularized Training of Intermediate Layers for Generative Models for Inverse Problems". In: *ArXiv Preprint* (2022).

[88] Kanghui Guo and Demetrio Labate. "Optimally sparse multidimensional representation using shearlets". In: *SIAM Journal on Mathematical Analysis* 39 (1 2007).

[89] Harshit Gupta, Michael T. McCann, Laurene Donati, and Michael Unser. "CryoGAN: A New Reconstruction Paradigm for Single-Particle Cryo-EM Via Deep Adversarial Learning". In: *IEEE Transactions on Computational Imaging* (2021).

[90] Andreas Habring and Martin Holler. "A Generative Variational Model for Inverse Problems in Imaging". In: *SIAM Journal on Mathematics of Data Science* 4 (1 2022), pp. 306–335.

[91] Jacques Hadamard. "Sur les problèmes aux dérivées partielles et leur signification physique". In: *Princeton University Bulletin* (1902), 49–52.

[92] Markus Haltmeier, Linh Nguyen, Daniel Obmann, and Johannes Schwab. "Sparse Lq-regularization of inverse problems with deep learning". In: *ArXiv Preprint* (2019).

[93] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P. Recht, Daniel K. Sodickson, Thomas Pock, and Florian Knoll. "Learning a variational network for reconstruction of accelerated MRI data". In: *Magnetic Resonance in Medicine* 79 (6 2018), pp. 3055–3071.

[94] Paul Hand, Oscar Leong, and Vladislav Voroninski. "Phase retrieval under a generative prior". In: *NeurIPS* (2018), pp. 9136–9146.

[95] Paul Hand and Vladislav Voroninski. "Global Guarantees for Enforcing Deep Generative Priors by Empirical Risk". In: *IEEE Transactions on Information Theory* 66 (1 2020), pp. 401–418.

[96] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *CVPR* (2016), pp. 770–778.

[97] Chinmay Hegde. "Algorithmic Aspects of Inverse Problems Using Generative Models". In: *56th Annual Allerton Conference on Communication, Control, and Computing* (2019), pp. 166–172.

[98] Allard Adriaan Hendriksen, Daniel Maria Pelt, and K. Joost Batenburg. "Noise2Inverse: Self-Supervised Deep Convolutional Denoising for Tomography". In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1320–1335.

[99] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs trained by a two time-scale update rule

converge to a local Nash equilibrium". In: *NeurIPS* (2017), pp. 6627–6638.

[100] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. "B-VAE: Learning basic visual concepts with a constrained variational framework". In: *ICLR* (2017).

[101] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *NeurIPS* 33 (2020), pp. 6840–6851.

[102] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2 (5 1989), pp. 359–366.

[103] Shady Abu Hussein, Tom Tirer, and Raja Giryes. "Image-Adaptive GAN based Reconstruction". In: *AAAI* (2019), pp. 3121–3129.

[104] Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. "Deep learning for undersampled MRI reconstruction". In: *Physics in Medicine and Biology* 63 (13 2018).

[105] Aapo Hyvärinen and Peter Dayan. "Estimation of non-normalized statistical models by score matching." In: *JMLR* 6 (4 2005).

[106] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: PMLR, 2015, pp. 448–456.

[107] Kazufumi Ito and Bangti Jin. *Inverse Problems: Tikhonov Theory And Algorithms (Applied Mathematics)*. 2014.

[108] Jörn Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. "I-RevNet: Deep invertible networks". In: *ICLR* (2018).

[109] Gauri Jagatap and Chinmay Hegde. "Algorithmic guarantees for inverse imaging with untrained network priors". In: *NeurIPS* 32 (2019).

[110] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G. Dimakis, and Jonathan I. Tamir. "Robust Compressed Sensing MRI with Deep Generative Priors". In: *NeurIPS* (2021).

[111] Manhua Jia, Wenjian Liu, Junwei Duan, Long Chen, C L Philip Chen, Qun Wang, and Zhiguo Zhou. "Efficient graph convolutional networks for seizure prediction using scalp EEG." In: *Frontiers in Neuroscience* 16 (2022), p. 967116.

[112] Xiajun Jiang, Sandesh Ghimire, Jwala Dhamala, Zhiyuan Li, Prashnna Kumar Gyawali, and Linwei Wang. "Learning geometry-dependent and physics-based inverse image reconstruction". In: *MICCAI* (2020), pp. 487–496.

[113] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. "Deep convolutional neural network for inverse problems in imaging". In: *IEEE Transactions on Image Processing* 26 (9 2017), pp. 4509–4522.

[114] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (7873 2021), pp. 583–589.

[115] Maya Kabkab, Pouya Samangouei, and Rama Chellappa. “Task-aware compressed sensing with generative adversarial networks”. In: *AAAI* (2018), pp. 2297–2304.

[116] Barbara Kaltenbacher, Frank Schöpfer, and Thomas Schuster. “Iterative methods for nonlinear ill-posed problems in Banach spaces: Convergence and applications to parameter identification problems”. In: *Inverse Problems* 25 (6 2009).

[117] U S Kamilov, H Mansour, and B Wohlberg. “A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems”. In: *IEEE Signal Processing Letters* 24 (12 2017), pp. 1872–1876.

[118] Takuhiro Kaneko and Tatsuya Harada. “Noise Robust Generative Adversarial Networks”. In: *CVPR* (2020), pp. 8401–8411.

[119] Eunhee Kang, Junhong Min, and Jong Chul Ye. “A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction”. In: *Medical physics* 44 (10 2017), e360–e375.

[120] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. “Training generative adversarial networks with limited data”. In: *NeurIPS* 33 (2020), pp. 12104–12114.

[121] Tero Karras, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *CVPR* (2019), pp. 4401–4410.

[122] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. “Analyzing and improving the image quality of stylegan”. In: *CVPR* (2020), pp. 8107–8116.

[123] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *ICLR* (0205).

[124] Diederik P. Kingma and Prafulla Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *NeurIPS* (2018), pp. 10215–10224.

[125] Diederik P. Kingma and Max Welling. “Auto-encoding variational bayes”. In: *ICLR* (2014).

[126] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. “Segment Anything”. In: (2023). URL: `http://arxiv.org/abs/2304.02643`.

[127] Florian Knoll, Kristian Bredies, Thomas Pock, and Rudolf Stollberger. “Second order total generalized variation (TGV) for MRI”. In: *Magnetic Resonance in Medicine* 65 (2 2011), pp. 480–491.

[128] Florian Knoll et al. “fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning”. In: *Radiology: Artificial Intelligence* 2 (1 2020).

[129] Ivan Kobyzev, Simon J.D. D Prince, and Marcus A. Brubaker. "Normalizing flows: An introduction and review of current methods". In: *IEEE transactions on pattern analysis and machine intelligence* 43 (11 2020), pp. 3964–3979.

[130] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37 (2 1991), pp. 233–243.

[131] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. "Noise2void-learning denoising from single noisy images". In: *CVPR* (2019), pp. 2129–2137.

[132] Karl Kunisch and Thomas Pock. "A Bilevel Optimization Approach for Parameter Learning in Variational Models". In: *SIAM Journal on Imaging Sciences* 6 (2013).

[133] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial examples in the physical world". In: *ICLR* (2019).

[134] Avisek Lahiri, Arnav Kumar Jain, Divyasri Nadendla, and Prabir Kumar Biswas. "Faster Unsupervised Semantic Inpainting: A GAN Based Approach". In: *ICIP* (2019), pp. 2706–2710.

[135] Kenneth L Lange and Richard E Carson. "EM reconstruction algorithms for emission and transmission tomography." In: *Journal of computer assisted tomography* 8 2 (1984), pp. 306–316.

[136] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86 (11 1998), pp. 2278–2323.

[137] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. "Noise2Noise: Learning image restoration without clean data". In: *ICML* 7 (2018).

[138] Qi Lei, Ajil Jalal, Inderjit S. Dhillon, and Alexandros G. Dimakis. "Inverting deep generative models, one layer at a time". In: *NeurIPS* 32 (2019).

[139] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6 (6 1993), pp. 861–867.

[140] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maaß. "The LoDoPaB-CT Dataset: A Benchmark Dataset for Low-Dose CT Reconstruction Methods". In: *Scientific Data* 8 (1 2021), pp. 1–12.

[141] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. "NETT: Solving inverse problems with deep neural networks". In: *Inverse Problems* 36 (6 2020).

[142] Steven Cheng Xian Li, Benjamin M. Marlin, and Bo Jiang. "Misgan: Learning from incomplete data with generative adversarial networks". In: *ICLR* (2019).

[143] Jiaming Liu, Yu Sun, Cihat Eldeniz, Weijie Gan, Hongyu An, and Ulugbek S. Kamilov. "RARE: Image Reconstruction Using Deep Priors Learned without Groundtruth". In: *IEEE Journal on Selected Topics in Signal Processing* 14 (6 2020), pp. 1088–1099.

[144] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep learning face attributes in the wild". In: *ICCV* (2015), pp. 3730–3738.

[145] David Lopez-Paz and Maxime Oquab. "Revisiting classifier two-sample tests". In: *ICLR* (2017).

[146] Sebastian Lunz, Ozan Öktem, and Carola Bibiane Schönlieb. "Adversarial regularizers in inverse problems". In: *NeurIPS* (2018), pp. 8507–8516.

[147] Michael Lustig, David Donoho, and John M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine* 58 (6 2007), pp. 1182–1195.

[148] Jun Lv, Jin Zhu, and Guang Yang. "Which GAN? A comparative study of generative adversarial network-based fast MRI reconstruction". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379 (2200 2021).

[149] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards deep learning models resistant to adversarial attacks". In: *ICLR* (2018).

[150] Morteza Mardani, Enhao Gong, Joseph Y. Cheng, Shreyas S. Vasanawala, Greg Zaharchuk, Lei Xing, and John M. Pauly. "Deep generative adversarial neural networks for compressive sensing MRI". In: *IEEE Transactions on Medical Imaging* 38 (1 2019), pp. 167–179.

[151] C. McCollough. "TU-FG-207A-04: Overview of the Low Dose CT Grand Challenge". In: *Medical Physics* 43 (6Part35 2016), pp. 3759–3760.

[152] Tim Meinhardt, Michael Moeller, Caner Hazirbas, and Daniel Cremers. "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *ICCV* (2017), pp. 1799–1808.

[153] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models". In: *CVPR* (2020), pp. 2437–2445.

[154] Christopher A Metzler, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk. "Unsupervised learning with Stein's unbiased risk estimator". In: *arXiv preprint arXiv:1805.10531* (2018).

[155] Vladimir Alekseevich Morozov. *Methods for solving incorrectly posed problems*. Springer Science Business Media, 2012.

[156] Lukas Mosser, Olivier Dubrule, and Martin J. Blunt. "Stochastic Seismic Waveform Inversion Using Generative Adversarial Networks as a Geological Prior". In: *Mathematical Geosciences* 52 (1 2020), pp. 53–79.

[157] Subhadip Mukherjee, Marcello Carioni, Ozan Öktem, and Carola-Bibiane Schönlieb. "End-to-end reconstruction meets data-driven regularization for inverse problems". In: *NeurIPS* (2021).

[158] Subhadip Mukherjee, Sören Dittmer, Zakhar Shumaylov, Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. "Learned convex regularizers for inverse problems". In: *ArXiv Preprint* (2020).

[159] Subhadip Mukherjee, Andreas Hauptmann, Ozan Öktem, Marcelo Pereyra, and Carola-Bibiane Schönlieb. "Learned reconstruction methods with convergence guarantees". In: *ArXiv Preprint* (2022).

[160] Dominik Narnhofer, Kerstin Hammernik, Florian Knoll, and Thomas Pock. "Inverse GANs for accelerated MRI reconstruction". In: SPIE-Intl Soc Optical Eng, 2019, p. 45.

[161] *NHS - MRI Scans*. 2022. URL: `https://www.nhs.uk/conditions/mri-scan/`.

[162] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. "f-GAN: Training generative neural samplers using variational divergence minimization". In: *NeurIPS* (2016), pp. 271–279.

[163] Daniel Obmann, Linh Nguyen, Johannes Schwab, and Markus Haltmeier. "Augmented nett regularization of inverse problems". In: *Journal of Physics Communications* 5 (10 2021), p. 105002.

[164] Daniel Obmann, Linh Nguyen, Johannes Schwab, and Markus Haltmeier. "Sparse Anett for Solving Inverse Problems with Deep Learning". In: *International Symposium on Biomedical Imaging Workshops, Proceedings* (2020).

[165] Daniel Obmann, Johannes Schwab, and Markus Haltmeier. "Deep synthesis regularization of inverse problems". In: *ArXiv Preprint* (2020).

[166] Changheun Oh, Dongchan Kim, Jun Young Chung, Yeji Han, and Hyun Wook Park. "Eter-net: End to end mr image reconstruction using recurrent neural network". In: *Lecture Notes in Computer Science* (2018).

[167] Gyutaek Oh, Byeongsu Sim, Hyung Jin Chung, Leonard Sunwoo, and Jong Chul Ye. "Unpaired Deep Learning for Accelerated MRI Using Optimal Transport Driven CycleGAN". In: *IEEE Transactions on Computational Imaging* (2020), pp. 1285–1296.

[168] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. "Deep learning techniques for inverse problems in imaging". In: *IEEE Journal on Selected Areas in Information Theory* 1 (1 2020), pp. 39–56.

[169] S Ono. "Primal-Dual Plug-and-Play Image Restoration". In: *IEEE Signal Processing Letters* 24 (8 2017), pp. 1108–1112.

[170] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning". In: *NeurIPS* 2017-Decem (2017), pp. 6307–6316.

[171] OpenAI. "GPT-4 Technical Report". In: (2023). URL: http://arxiv.org/abs/2303.08774.

[172] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. "An Iterative Regularization Method for Total Variation-Based Image Restoration". In: *Multiscale Modeling Simulation* 4 (2 2005), pp. 460–489.

[173] John Paisley, David M. Blei, and Michael I. Jordan. "Variational Bayesian inference with stochastic search". In: *ICML* 2 (2012), pp. 1367–1374.

[174] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "Normalizing flows for probabilistic modeling and inference". In: *JMLR* 22 (57 2021), pp. 1–64.

[175] *Papers with code - trends*. URL: https://paperswithcode.com/trends.

[176] Hyoung Suk Park, Jineon Baek, Sun Kyoung You, Jae Kyu Choi, and Jin Keun Seo. "Unpaired image denoising using a generative adversarial network in x-ray CT". In: *IEEE Access* 7 (2019), pp. 110414–110425.

[177] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. "Scikit-learn: Machine Learning in Python". In: *JMLR* 12 (2011), pp. 2825–2830.

[178] Pei Peng, Shirin Jalali, and Xin Yuan. "Auto-encoders for compressed sensing". In: *NeurIPS* (2019).

[179] David L. Phillips. "A Technique for the Numerical Solution of Certain Integral Equations of the First Kind". In: *Journal of the ACM (JACM)* 9 (1 1962).

[180] K P Pruessmann, M Weiger, M B Scheidegger, and P Boesiger. "SENSE: sensitivity encoding for fast MRI." In: *Magnetic resonance in medicine* 42 (5 1999), pp. 952–962.

[181] Tran Minh Quan, Thanh Nguyen-Duc, and Won Ki Jeong. "Compressed Sensing MRI Reconstruction Using a Generative Adversarial Network With a Cyclic Loss". In: *IEEE Transactions on Medical Imaging* 37 (6 2018), pp. 1488–1497.

[182] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* (2022).

[183] Jonas Rauber, Wieland Brendel, and Matthias Bethge. "Foolbox: A Python toolbox to benchmark the robustness of machine learning models". In: *arXiv preprint* (2017).

[184] Andrew J. Reader, Guillaume Corda, Abolfazl Mehranian, Casper da Costa-Luis, Sam Ellis, and Julia A. Schnabel. "Deep Learning for PET

Image Reconstruction". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 5 (1 2020).

[185] *Reducing Bias and Improving Safety in DALL·E 2*. 2022. URL: https://openai.com/blog/reducing-bias-and-improving-safety-in-dall-e-2/.

[186] Juan los Reyes and Carola-Bibiane Schönlieb. "Image denoising: Learning the noise model via nonsmooth PDE-constrained optimization". In: *Inverse Problems and Imaging* 4 (2013).

[187] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *ICML* 4 (2014), pp. 3057–3070.

[188] Yaniv Romano, Michael Elad, and Peyman Milanfar. "The little engine that could: Regularization by Denoising (RED)". In: *SIAM Journal on Imaging Sciences* 10 (4 2017), pp. 1804–1844.

[189] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *MICCAI* (2015), pp. 234–241.

[190] Stefan Roth and Michael J Black. "Fields of experts". In: *International Journal of Computer Vision* 82 (2 2009), pp. 205–229.

[191] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. "Metric for distributions with applications to image databases". In: *ICCV* (1998), pp. 59–66.

[192] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60 (1-4 1992), pp. 259–268.

[193] Lars Ruthotto and Eldad Haber. "An introduction to deep generative modeling". In: *GAMM-Mitteilungen* 44 (2 2021).

[194] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. "Plug-and-play methods provably converge with properly trained denoisers". In: *International Conference on Machine Learning* (2019), pp. 5546–5557.

[195] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training GANs". In: *NeurIPS* (2016), pp. 2234–2242.

[196] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, and Frank Lenzen. *Variational regularization methods for the solution of inverse problems*. 2009.

[197] 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". In: *IEEE Transactions on Medical Imaging* 37 (2 2018), pp. 491–503.

[198] Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. "Big in Japan: Regularizing Networks for Solving Inverse Problems". In: *Journal of Mathematical Imaging and Vision* 62 (3 2020), pp. 445–455.

[199] Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. "Deep null space learning for inverse problems: Convergence analysis and rates". In: *Inverse Problems* 35 (2 2019), p. 025008.

[200] Viraj Shah and Chinmay Hegde. "Solving Linear Inverse Problems Using Gan Priors: An Algorithm with Provable Guarantees". In: *ICASSP* (2018), pp. 4609–4613.

[201] Or Sharir, Barak Peleg, and Yoav Shoham. "The Cost of Training NLP Models: A Concise Overview". In: *ArXiv Preprint* (2020).

[202] L A Shepp and Y Vardi. "Maximum likelihood reconstruction for emission tomography." In: *IEEE Transactions on Medical Imaging* 1 (2 1982), pp. 113–122.

[203] Byeongsu Sim, Gyutaek Oh, and Jong Chul Ye. "Optimal Transport Structure of CycleGAN for Unsupervised Learning for Inverse Problems". In: *ICASSP* (2020), pp. 8644–8647.

[204] Naresh K. Sinha and Michael P. Griscik. "Stochastic Approximation Method". In: *IEEE Transactions on Systems, Man and Cybernetics* 1 (4 1971), pp. 338–344.

[205] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. "Ladder variational autoencoders". In: *NeurIPS* (2016), pp. 3745–3753.

[206] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning* (2015), pp. 2256–2265.

[207] Shakarim Soltanayev and Se Young Chun. "Training deep learning based denoisers without ground truth data". In: *NeurIPS* 31 (2018).

[208] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *NeurIPS* 32 (2019).

[209] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. "Sliced score matching: A scalable approach to density and score estimation". In: *Uncertainty in Artificial Intelligence* (2020), pp. 574–584.

[210] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *ICLR* (20201).

[211] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". In: *JMLR* 15 (2014), pp. 1929–1958.

[212] Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. "Wasserstein GANs Work Because They Fail (to Approximate the Wasserstein Distance)". In: *ArXiv Preprint* (2021).

[213] A. M. Stuart. "Inverse problems: A Bayesian perspective". In: *Acta Numerica* 19 (2010), pp. 451–459.

[214] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *ICLR* (2014).

[215] Lucas Theis, Aäron Van Den Oord, and Matthias Bethge. "A note on the evaluation of generative models". In: *ICLR* (2016).

[216] A N Tikhonov. "On the stability of inverse problems". In: *Doklady Akademii Nauk Sssr* 39 (5 1943).

[217] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. 1995.

[218] Subarna Tripathi, Zachary C. Lipton, and Truong Q. Nguyen. "Correction by Projection: Denoising Images with Generative Adversarial Networks". In: *ArXiv Preprint* (2018).

[219] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *International Journal of Computer Vision* (2020), pp. 1867–1888.

[220] Mihaly Varadi et al. "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models". In: *Nucleic Acids Research* 50 (D1 2022), pp. D439–D444.

[221] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *NeurIPS* 30 (2017).

[222] Dave Van Veen, Ajil Jalal, Mahdi Soltanolkotabi, Eric Price, Sriram Vishwanath, and Alexandros G. Dimakis. "Compressed Sensing with Deep Image Prior and Learned Regularization". In: *ArXiv Preprint* (2018).

[223] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. "Plug-and-Play priors for model based reconstruction". In: *GlobalSIP* (2013), pp. 945–948.

[224] Cédric Villani. *Optimal Transport: Old and New*. Number 338. Springer Verlag, 2009.

[225] Paul Vos and Qiang Wu. *Probability Essentials*. Vol. 38. Springer Science Business Media, 2018, pp. 75–109.

[226] Shanshan Wang, Zhenghang Su, Leslie Ying, Xi Peng, Shun Zhu, Feng Liang, Dagan Feng, and Dong Liang. "Accelerating magnetic resonance imaging via deep learning". In: *Proceedings - International Symposium on Biomedical Imaging* 2016-June (2016), pp. 514–517.

[227] Yang Wang. "A Mathematical Introduction to Generative Adversarial Nets (GAN)". In: *ArXiv Preprint* (2020).
[228] Bradley a. Warner and Radford M Neal. "Bayesian Learning for Neural Networks". In: *Journal of the American Statistical Association* 92 (1997), p. 791. URL: http://www.jstor.org/stable/2965731?origin=crossref.
[229] Tom White. "Sampling Generative Networks". In: *ArXiv Preprint* (2016).
[230] Jingyan Xu and Frédéric Noo. "Convex optimization algorithms in medical image reconstruction—in the age of AI". In: *Physics in Medicine and Biology* 67 (7 2022), 07TR01.
[231] Burhaneddin Yaman, Seyed Amir Hossein Hosseini, Steen Moeller, Jutta Ellermann, Kâmil Uğurbil, and Mehmet Akçakaya. "Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data". In: *Magnetic resonance in medicine* 84 (6 2020), pp. 3172–3191.
[232] 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". In: *IEEE Transactions on Medical Imaging* 37 (6 2018), pp. 1310–1321.
[233] Yisong Yang. *A concise text on advanced linear algebra*. 2014.
[234] Jong Chul Ye. "Compressed sensing MRI: a review from signal processing perspective". In: *BMC Biomedical Engineering* 1 (1 2019).
[235] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. "Semantic image inpainting with deep generative models". In: *CVPR* (2017), pp. 6882–6890.
[236] *Your AI pair programmer*. URL: https://github.com/features/copilot.
[237] X Yuan, P He, Q Zhu, and X Li. "Adversarial Examples: Attacks and Defenses for Deep Learning". In: *IEEE Transactions on Neural Networks and Learning Systems* 30 (9 2019), pp. 2805–2824.
[238] Jure Zbontar et al. "fastMRI: An Open Dataset and Benchmarks for Accelerated MRI". In: *ArXiv Preprint* (2018).
[239] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64 (3 2021), pp. 107–115.
[240] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. "Theoretically Principled Trade-off between Robustness and Accuracy". In: *ICML* 97 (2019). Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov, pp. 7472–7482.
[241] Ji Zhao, Zhiqiang Chen, Li Zhang, and Xin Jin. "Few-view CT reconstruction method based on deep learning". In: *2016 IEEE Nuclear Science*

Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD) (2016), pp. 1–4.

[242] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning Deep Features for Discriminative Localization". In: *CVPR* (2016), pp. 2921–2929.

[243] Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. "Image reconstruction by domain-transform manifold learning". In: *Nature* 555 (7697 2018), pp. 487–492.

[244] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *ICCV* (2017), pp. 2242–2251.

[245] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. "Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior". In: *CVPR* (2019), pp. 10247–10256.