LEIDEN UNIVERSITY

MASTER THESIS

Data Compression for Weak Lensing Studies with Information Maximizing Neural Networks

Author:
Erik OSINGA

Supervisor:
prof. dr. Henk HOEKSTRA
Dr. Mohammadjavad VAKILI

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Weak lensing group
Sterrewacht Leiden

June 30, 2019

“There are no gains without pains”

Benjamin Franklin

LEIDEN UNIVERSITY

Abstract

Sterrewacht Leiden

Master of Science

Data Compression for Weak Lensing Studies with Information Maximizing Neural Networks

by Erik OSINGA

Over the next decade, stage IV lensing surveys such as the Euclid mission will revolutionize our view of the Universe by providing some of the tightest constraints on cosmological parameters via clustering and weak lensing. To accomplish this, these surveys will observe billions of galaxies. A crucial step in the inference of cosmological parameters is compressing this large volume of data to a manageable amount of statistical summaries. This work presents the first application of a novel data compression method to weak lensing data. We study the capabilities of a machine learning technique, the information maximizing neural network, applied to a Euclid-like survey. We find that the neural network provides an informative mapping from highly dimensional data to a single summary statistic in the test case of inferring the cosmological parameter Ω_m from mock data. This mapping is used to infer accurate posterior distributions without assuming the form of the likelihood. For multiple parameters, Ω_m and σ_8, the network provides unbiased informative summaries, but does not extract information added by tomographic bins. To lower the computational costs of the likelihood-free inference step, we show that a Gaussian process provides a promising emulator that can be used to predict summary statistics directly from the parameters. This is achieved by fitting only a small set of forward simulations that optimally sample the parameter space, providing a massive speedup of the inference step. While the current implementation still has some identified flaws, we conclude that the information maximizing neural network, rather than replacing classic analysis methods, can be an important addition to the analysis of future surveys.

Acknowledgements

First of all, I want to thank my supervisors, Mohammadjavad and Henk. I have learned so much from you over the past year and greatly enjoyed our bi-weekly discussions. In particular, Mohammadjavad, thank you for being so involved in the project. You were available anytime that I needed you even when I would come looking for you unannounced. I enjoyed our 'power programming' sessions in which we both just sat behind the computer and tried to get the code to work.

I also want to thank Tom Charnock, for being very helpful whenever I was stuck with a bug or was confused about a part of the neural network. You have come up with a brilliant idea and written a great piece of software. Like you have said 'cheers for your effort on the IMNN' I would like to say 'cheers for your effort on me' right back at you.

I too want to thank Jonah and *Couzy*. I am so glad that you guys decided to do a research project in cosmology as well, so that we could all be confused together at the same time. Our discussions and jokes about our problems have made the past year very fun, and I will certainly miss those.

Finally, I want to thank Tjissa. It has been probably the busiest year of both our lives, and it is easy to get lost in all the work that one has to do and forget to actually have a life. However, you always put things into perspective for me and made sure that we have our fair share of fun as well.

This research has also made used of many software tools. The most frequently used are the Python packages SciPy, Tensorflow, NumPy, Matplotlib and Corner, the CosmoSIS software, the text editor Sublime Text and the LaTeX editor Overleaf.

Contents

Chapter 1

Introduction

Cosmology is the study of the Universe as a whole. It aims to answer deeply fundamental questions about the origin, evolution and fate of the Universe. These questions have been asked since the dawn of mankind and for the larger part of history could only be answered in a philosophical or religious framework. For only a few centuries now have attempts been made to answer these questions within the framework of physical sciences.

A large advance in the field of cosmology was made in 1917, when Einstein applied his theory of general relativity to the Universe (Einstein, 1917). By assuming that the universe is dominated by matter and should be static he found that he had to introduce an additional term into his general relativistic equations, which we now know as the *cosmological constant*. However, Hubble's paper on the famous relation between distance and velocity (Hubble, 1929) indicated that the Universe is expanding rather than static. Einstein could then dispose of the cosmological constant in favor of an expanding matter-filled Universe. The debate between whether the Universe was steady-state or expanding was finally settled with the discovery of the cosmic microwave background radiation (Penzias and R. W. Wilson, 1965). Ironically, the cosmological constant is in recent years favored again as the explanation for the observed *accelerated* expansion of the Universe inferred from distant type Ia supernovae (Riess, Filippenko, et al., 1998; Perlmutter et al., 1999).

Currently, the model which is found to describe our Universe extremely well is called the ΛCDM model. The ΛCDM model is a model with only 6 parameters that correctly predicts (to very high accuracy) many observations. A few examples are the cosmic microwave background (e.g., Hinshaw et al., 2013; Planck Collaboration, Ade, et al., 2016), the formation of large scale structure (e.g., Burenin and Vikhlinin, 2012), the cosmic elemental abundance and baryonic acoustic oscillations (e.g., Samushia, Reid, White, Percival, Cuesta, Zhao, et al., 2014). In this concordance model, the baryons make up only about 5% of the Universe, while cold dark matter (CDM) and dark energy (Λ) make up about 25% and 70% respectively. The two main components, dark energy and dark matter, are very poorly understood. There is to date no observed particle that could be responsible for the CDM and the cosmological constant is even more inexplicable. The current 'best' explanation for the cosmological constant is vacuum energy, predicted from quantum mechanics. However, the value of the vacuum energy is found to be at least 60 orders of magnitude greater than the observed value of Λ, a problem that is called the cosmological constant problem (Weinberg, 1989).

Apart from the theoretical problems that ΛCDM faces, assuming ΛCDM produces unexplained tensions between parameters inferred from high-redshift observations (i.e., the Planck cosmic microwave background data) (Planck Collaboration, Ade, et al., 2016), and low-redshift observations. The most notable tension is the so-called Hubble tension, which refers to the now 4.4σ (Riess, Casertano, et al.,

2019) tension between the Hubble constant inferred by type Ia supernovae (Riess, Casertano, et al., 2019) and the Hubble constant inferred by the Planck collaboration (Planck Collaboration, Aghanim, et al., 2018). The tension is corroborated by several independent measurements, such as strong lensing constraints on the Hubble constant from the low-redshift universe (Birrer et al., 2019) and baryon acoustic oscillation data from the high-redshift universe (Addison et al., 2018).

A slightly less significant tension with Planck is also found by large scale structure observations such as weak lensing and cluster number counts (Heymans et al., 2013; Hildebrandt et al., 2017; Macaulay, Wehus, and Eriksen, 2013). The recent results from the KiDS weak lensing survey show for example a value for $S_8 \equiv \sigma_8\sqrt{\Omega_m/0.3}$ that is in 2.3σ tension with Planck (Hildebrandt et al., 2017).

Since the parameter constraints seem to be a function of redshift, the tensions could be indicative of the need for a dynamical dark energy model, modified gravity theories (see e.g., Amendola, Appleby, Avgoustidis, et al., 2018, for a review) or even more exotic explanations (e.g., Di Valentino, Linder, and Melchiorri, 2018; Hooper, Krnjaic, and McDermott, 2019; Vattis, Koushiappas, and Loeb, 2019).

Future cosmological experiments will be able to distinguish between at least some of these models, should the tensions persist. The upcoming Euclid mission (Laureijs et al., 2011) and the planned Large Synoptic Survey Telescope (LSST Science Collaboration et al., 2009) aim to increase cosmological parameter constraints through weak lensing and galaxy clustering by an order of magnitude. Weak lensing is a powerful cosmological probe and is explained in more detail in the next section.

1.1 Weak lensing

According to Einstein's theory of general relativity (GR), gravity can be viewed as a geometric property of space-time. Matter and energy alter the geometry of space-time by curving it. In this curved space-time, photons follow null geodesics and thus follow perturbed paths around massive objects. The first empirical evidence of this effect being well described by GR was the observed deflection of light by the Sun (Dyson, Eddington, and Davidson, 1920). This curving of light around massive objects is similar to the propagation of light through different media, which gives this effect its name 'gravitational lensing'.

There are three separate gravitational lensing regimes: strong lensing, microlensing and weak lensing. Strong lensing considers easily measurable distortions such as arcs or multiple images, which allow us to infer properties of the mass and mass profile of the lensing source. Strong lensing also amplifies the brightness of a source, which allows for detection of sources that are otherwise too faint to be observed (e.g., sub-mm galaxies; Negrello et al. (2010)). Microlensing is a special case of strong gravitational lensing where the separation between multiple images is too small to be resolved. As the lens and source move across the field of view, properties of both the lens and lensed source can be deduced from the time variability. Microlensing can be used to discover planets with Earth-like masses or even planets in other galaxies (see e.g., Tsapras (2018) for a recent review).

Finally, weak lensing, the regime of interest in this thesis, is the regime that encompasses any lensing effect that is not directly visible by eye, but can be inferred from statistics. Weak lensing is an extremely rich cosmological probe as it is a function of both the geometry of the universe and the growth of structure along the line of sight. This allows us to make estimates about the initial perturbations in the Universe, the abundance of (dark) matter, the evolution of large scale structure and potentially distinguish between dark energy and modified gravity theories (see Kilbinger, 2015, for a review).

As weak lensing is by definition not visible by eye, this effect is observed as a correlation of galaxy shapes as a function of distance to the lens. To observe this correlation, one must assume that the shape of galaxies in the Universe is randomly distributed or make some assumption about the intrinsic alignment (IA) of galaxies (see e.g., Kiessling et al., 2015; Troxel and Ishak, 2015, for reviews on IA).

In the weak lensing regime, the lens can be either a galaxy cluster, a single galaxy or the entire large scale cosmic structure along the line of sight towards the lens. If the lens is the latter, then the effect is called cosmic shear. Cosmic shear is difficult to measure, which can be appreciated from the fact that it was only first detected in 2000 (A. R. Refregier, Ellis, and D. J. Bacon, 2000; Van Waerbeke et al., 2000; Wittman et al., 2000; Kaiser, G. Wilson, and Luppino, 2000).

Since the first detection of cosmic shear, the field of weak cosmological lensing has evolved rapidly. Current surveys study cosmic shear as as a function of redshift with a technique called tomography (Hu, 1999), where the signal is divided into bins of redshift. To measure shapes and redshifts at the same time, multi-band surveys such as KiDS (Kuijken et al., 2015; Hildebrandt et al., 2017; Köhlinger et al., 2017; van Uitert et al., 2018), CFHTLens (Heymans et al., 2013; Kitching et al., 2014) and DES (The Dark Energy Survey Collaboration, 2005; Troxel, MacCrann, et al., 2018) are being carried out. In the next decade, larger surveys such as Euclid (Laureijs et al., 2011) and LSST (LSST Science Collaboration et al., 2009) will allow even better constraints on cosmological parameters.

As we enter into what is being called the era of precision cosmology, there is a growing need for more powerful statistical methods to analyze the massive amounts of data that future surveys will provide. Euclid will obtain multi-band images of a billion galaxies, which are noisy and convolved with a PSF. The shapes of these galaxies need to be measured to at least $\sim 1\%$ level accuracy (Cropper et al., 2013; Hoekstra, Viola, and Herbonnet, 2017). The photometric redshifts need to have a scatter $\sigma_z < 0.05(1+z)$ (Bordoloi, Lilly, and Amara, 2010), for which methods such as template fitting (e.g., Bolzonella, Miralles, and Pelló, 2000; Salvato et al., 2009) or machine learning (e.g., D'Isanto and Polsterer, 2018; Laurino et al., 2011) are being explored. The errors have to be propagated in a robust way, which requires the unfeasibly large calculation of the covariance matrix, given that Euclid is estimated to have about 10^4 summary statistics (Heavens et al., 2017).

Additionally, the theory to predict the statistics of the weak lensing surveys is still poorly developed at scales smaller than few Mpc due to the non-linear collapse of matter on these scales. In current cosmological analyses, Gaussian likelihoods are often assumed. While the initial matter field is indeed Gaussian, and the central limit theorem causes combinations of random variables to be Gaussian distributed, non-linear functions of Gaussian variables are in general not Gaussian distributed. It has been found that non-Gaussian correlations will significantly affect future weak lensing surveys (Sellentin and Heavens, 2018), so a likelihood-free analysis may be a better approach for future surveys.

To tackle the problems that current and future cosmological surveys face, machine learning techniques are being explored more and more. Oftentimes, an artificial neural network is trained to replace a complicated part of the analysis process or to avoid making certain assumptions (e.g., about the likelihood) (e.g., Fluri, Kacprzak, A. Refregier, et al., 2018; Alsing, B. Wandelt, and Feeney, 2018; Fluri, Kacprzak, Lucchi, et al., 2019).

1.2 Artificial neural networks

The term 'artificial neural networks' refers to a type of machine learning framework that is inspired by biological neural networks. The simplest framework consists of a set of connected neurons (i.e., nodes) grouped into layers, where each layer takes some number of inputs and produces an output for each neuron in the layer. Each connection is parameterized by two parameters, the weight and the bias, and each output of a neuron is activated by a non-linear activation function. Series of layers can be trained to learn complex non-linear functions through supervised learning (Goodfellow, Bengio, and Courville, 2016). By adding neurons and layers more complexity is added to the network, allowing for greater levels of abstraction but at the risk of overfitting the data.

There are many extensions of the simplest framework. Perhaps in astronomy the most interesting variant is the convolutional neural network (CNN) (see e.g., Gu et al., 2015, for a review). This type of network is commonly used in cases where the input data are images, or can be represented as images. CNNs have for example been used to classify and associate radio galaxies (Alhassan, Taylor, and Vaccari, 2018; Wu et al., 2019), to mitigate radio frequency interference (Akeret et al., 2017) and to differentiate weak lensing maps from different cosmological models (Schmelzle et al., 2017).

In the area of cosmology, machine learning is being explored as a possible solution to model complex physical processes such as structure formation (e.g., S. He et al., 2018) or the cosmic microwave background gravitational lensing potential (e.g., Caldeira et al., 2018). Some recent successes are the following. Galaxy cluster masses can be inferred with lower scatter than traditional methods (Ho et al., 2019; Armitage, Kay, and Barnes, 2019). Peel et al. (2018) have shown that CNNs can use weak lensing maps to distinguish between standard and modified gravity cosmologies. The non-Gaussian information in the smallest scales of weak lensing maps might be extracted with machine learning techniques to produce tighter constraints on cosmological parameters (Ribli, Pataki, and Csabai, 2019; Gupta et al., 2018).

Although these successes sound promising, results obtained through machine learning are difficult to interpret. While it can provide seemingly accurate models for the input data that it was trained on, no insight is given into how the model works. This problem is often called the black box problem (Zednik, 2019). Due to the nature of machine learning algorithms, no guarantee is provided that the output of the network is robust against different realizations of input data. Extreme examples of this are state of the art networks that classify unrecognizable images with very high certainty (Nguyen, Yosinski, and Clune, 2014) or misclassify images completely when just a few pixels are adjusted in the image (Szegedy et al., 2013). The black box or robustness problem is not discussed in many of the above mentioned papers.

1.3 This thesis

It is clear that there are some issues with the currently accepted standard cosmological model. Future weak lensing surveys such as Euclid and the Large Synoptic Telescope Survey will probe the nature of dark energy, map out dark matter and hopefully gain more insight in the tensions between parameters inferred from large scale structure and cosmic microwave background studies.

To have a chance to tackle the panoply of challenges discussed in Section 1.1 that future surveys will come across, it is clear that it is important to be able to extract every piece of available information from the data. Currently, the number of estimated summary statistics for Euclid is too large, and there is still no good method to preserve information of the non-linear regime of structure formation.

A method is required to compress the raw data to a small number of informative summaries, while losing minimal information. This thesis aims to do so with information maximizing neural networks (Charnock, Lavaux, and B. D. Wandelt, 2018). The small number of summaries are then used to generate constraints on cosmological parameters with likelihood-free inference.

The structure of this thesis is as follows. Chapter 2 will start with an overview of definitions that are used throughout this thesis and explain the formalism of weak lensing. In Chapter 3 the methods are discussed. First, likelihood-free inference and techniques to perform this efficiently are explained. Then, the information maximizing neural network is described. Finally, the software used to generate cosmological simulations is discussed. In Chapter 4 the information maximizing neural network is used to compress Gaussian signals. This allows for a study of the versatility and robustness of the network in a case where analytical posteriors are still available. Then, in Chapter 5 weak lensing data is generated for a Euclid-like survey. This data is compressed by the network, and posteriors are generated for cosmological parameters. Finally, in Chapter 6 we discuss the results and conclude in Chapter 7.

Chapter 2

Definitions and Derivations

This chapter will explicate some definitions that are used throughout this thesis. It starts with a short description of statistical inference, and introduces some concepts that accompany inference. Then a short description of the cosmic shear formalism is given, which serves mainly as additional background to the interested reader.

2.1 Statistics

Much of the general descriptions are introduced by Wasserman (2010). This book is followed in this section.

2.1.1 Inference

Statistical inference is the process of deducing the the probability distribution function (PDF) that generated the observed data. Oftentimes, the PDF is assumed, and only the parameters of the PDF are inferred. In that case, a parametric model is constructed, which takes the form

$$\mathcal{F} = \{f(x;\theta) : \theta \in \Theta\} \tag{2.1}$$

where θ is the vector of unknown parameters which can take values in the parameter space Θ. For example, if we would assume the parametric model of the data is a normal distribution then θ contains the mean and standard deviation, $\theta = (\mu, \sigma)$, and the parameter space Θ would be $(\mu \in \mathcal{R}, \sigma > 0)$. Statistical inference tries to estimate as best as possible the true parameters θ, this estimate is denoted by $\hat{\theta}$. The two dominant approaches for this estimation are called *frequentist* and *Bayesian*.

Frequentist The frequentist approach assumes that parameters are fixed unknown constants and our measurements sample noisy draws from the model with the true parameters. The most common method to estimate the parameters is from the *likelihood function*. If we let $X_1, ..., X_n$ be independent and identically distributed (i.i.d.) draws from a PDF $f(x;\theta)$ then the likelihood function is defined by

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(X_i;\theta). \tag{2.2}$$

Thus, the likelihood is simply the joint density of the data, but treated as a function of the parameter θ. The maximum likelihood estimate (MLE) $\hat{\theta}$ is the value of θ that maximizes $\mathcal{L}(\theta)$. This is the same value that maximizes the natural logarithm of the likelihood, so often times it is easier to maximize the log-likelihood.

Bayesian The Bayesian approach incorporates a prior belief about the parameter θ. This prior is taken into account according to Bayes' theorem

$$f(\theta|x^n) = \frac{f(x^n|\theta)f(\theta)}{\int f(x^n|\theta)f(\theta)d\theta} \tag{2.3}$$

where $f(x^n|\theta)$ is simply the likelihood function (Eq. 2.2) and thus we can write

$$f(\theta|x^n) = \frac{\mathcal{L}(\theta)f(\theta)}{\int \mathcal{L}(\theta)f(\theta)d\theta}. \tag{2.4}$$

Since the denominator is simply a normalizing constant, we can see that the *posterior* $f(\theta|x^n)$ is proportional to the likelihood times the prior. From this posterior we can then get point estimates (by for example taking the mean or maximum value) or obtain confidence intervals for θ.

In this work, and in much of recent cosmology, the Bayesian approach is taken. In this approach, every new experiment adds extra information which should be incorporated into the larger picture. The beliefs about model parameters from the previous experiments and theoretical considerations are taken into account via the prior. Often, Gaussian or log-normal priors are used (e.g., Jasche and Lavaux, 2015; Alsing and B. Wandelt, 2019).

2.1.2 Statistical summaries

A statistic is a function $T(X^n)$ of the data. It is a random variable since the input to the function are random variables. Most statistics are lossy operations, meaning that some information will be lost when applying the function to the data. When a statistic contains all information that is in the data, it is called a *sufficient statistic*.

For example, if we have N i.i.d. data points drawn from a normal distribution with unkown mean and variance: $X_1, ..., X_n \sim \mathcal{N}(\mu, \sigma)$, then the likelihood function (Eq. 2.2) is given by

$$\mathcal{L}(\theta) = \prod_i^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(X_i-\mu)^2}{2\sigma^2}\right] \tag{2.5}$$

which can be written as

$$\mathcal{L}(\theta) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left[-\frac{nS^2}{2\sigma^2}\right] \exp\left[-\frac{n(\bar{X}-\mu)^2}{2\sigma^2}\right], \tag{2.6}$$

where $\bar{X}$ are S^2 are the sample mean and sample variance respectively. We can see that Equation 2.6 only depends on the statistic $T = (\bar{X}, S)$ and as such T is a sufficient statistic. In fact, here T is the *minimal* sufficient statistic, since T captures the information needed to compute the likelihood function as concise as possible.

2.1.3 Information

In statistics, information often refers to the amount of information that a random variable X carries about the model parameters θ. The Fisher information (Fisher, 1925) is a way to quantify this. The Fisher information is defined as the second moment of the score. The score is defined as the derivative of the logarithmic likelihood function with respect to θ. The score measures how sensitive a likelihood function $\mathcal{L}(\theta)$ is to changes in the parameter θ. By calculating the variance of the score, intuitively we measure how accurate this sensitivity is. For multiple parameters θ_α and θ_β the Fisher information matrix is calculated as

$$F_{\alpha\beta}(\theta) = \int \left(\frac{\partial \log \mathcal{L}(\theta)}{\partial \theta_a} \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta_b} \right) \mathcal{L}(\theta) dx. \tag{2.7}$$

A different way to look at it is that the Fisher information measures the overall sensitivity of the likelihood function to changes in θ by weighing the sensitivity by the likelihood of observing the outcome x.

When the likelihood is twice continuously differentiable, the Fisher information can be rewritten to a simpler form (Charnock, Lavaux, and B. D. Wandelt, 2018), using the fact that the expectation value of the score is zero. We will first show that the expectation value of the score is zero.

$$\begin{aligned} E\left[\frac{\partial}{\partial \theta} \log \mathcal{L}(\theta)|_\theta \right] &= \int \frac{\frac{\partial}{\partial \theta}\mathcal{L}(\theta)}{\mathcal{L}(\theta)} \mathcal{L}(\theta) dx \\ &= \int \frac{\partial}{\partial \theta} \mathcal{L}(\theta) dx. \end{aligned} \tag{2.8}$$

The likelihood as a function of the parameter θ does not in general integrate to unity, but here the integral is a function of the data. Thus we get

$$E\left[\frac{\partial}{\partial \theta} \log \mathcal{L}(\theta)|_\theta \right] = \frac{\partial}{\partial \theta} 1 = 0. \tag{2.9}$$

Which proves that the expectation value of the score is zero.

Taking the derivative of the expectation value of the score must thus equal zero as well

$$\frac{\partial}{\partial \theta} \int \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \mathcal{L}(\theta) dx = \int \frac{\partial^2 \log \mathcal{L}(\theta)}{\partial \theta^2} \mathcal{L}(\theta) dx + \int \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \frac{\partial \mathcal{L}(\theta)}{\partial \theta} dx. \tag{2.10}$$

The second term on the right-hand side can be written as

$$\int \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \frac{\partial \mathcal{L}(\theta)}{\partial \theta} dx = \int \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \frac{\frac{\partial \mathcal{L}(\theta)}{\partial \theta}}{\mathcal{L}(\theta)} \mathcal{L}(\theta) dx = \int \left(\frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \right)^2 \mathcal{L}(\theta) dx \tag{2.11}$$

using the chain rule. We can see that Equation 2.11 is simply the variance of the score. As Equation 2.10 must equal zero, the variance of the score must be equal to the first term on the right-hand side of Equation 2.10. Therefore

$$V\left[\frac{\partial \log \mathcal{L}(\theta)}{\partial \theta} \right] = -\int \frac{\partial^2 \log \mathcal{L}(\theta)}{\partial \theta^2} \mathcal{L}(\theta) dx = -E\left[\frac{\partial^2 \log \mathcal{L}(\theta)}{\partial \theta^2} \right] \tag{2.12}$$

Finally, Eq. 2.12 proves that the Fisher information matrix entries α and β can be written as

$$F_{\alpha\beta}(\theta) = -\left\langle \frac{\partial^2 \ln \mathcal{L}(\mathbf{d}|\theta)}{\partial \theta_\alpha \partial \theta_\beta} \right\rangle. \tag{2.13}$$

where we now explicitly include $\mathbf{d}$ as some observed data.

As an example, if the likelihood is Gaussian we can write the log-likelihood as

$$\ln \mathcal{L}(\mathbf{d}|\theta) = -\frac{1}{2}\left((\mathbf{d} - \mu(\theta))^T C^{-1}(\mathbf{d} - \mu(\theta)) + \ln|2\pi C|\right). \tag{2.14}$$

where μ and C indicate the mean and covariance. This allows us to calculate the Fisher information matrix as (Charnock, Lavaux, and B. D. Wandelt, 2018)

$$F_{\alpha\beta}(\theta) = Tr\left[\left(\frac{\partial}{\partial\theta_\alpha}\mu(\theta)\right)^T C^{-1}\left(\frac{\partial}{\partial\theta_\beta}\mu(\theta)\right)\right]. \tag{2.15}$$

2.2 Weak lensing

This section serves as an introduction to the definitions used in cosmological weak lensing. This is the branch of weak lensing that considers the perturbation of light rays due to the cosmological large scale structure. First we derive the formalism used in cosmic shear studies and then derive the lensing power spectrum.

2.2.1 Cosmic shear formalism

We mainly follow the review by Kilbinger (2015) for many of these derivations. For a more in-depth review, see M. Bartelmann and Schneider (2001).

To describe the distortion of light due to the matter distribution in the universe, we need to consider an inhomogeneous universe. In an inhomogeneous universe, the metric that describes weak ($\ll c^2$) potential perturbations (e.g., in the case of weak lensing) Ψ and Φ is given by

$$ds^2 = \left(1 + \frac{2\Psi}{c^2}\right)c^2dt^2 - a^2(t)\left(1 - \frac{2\Phi}{c^2}\right)dl^2 \tag{2.16}$$

where $a(t)$ is the scale factor as a function of time, c the speed of light and l the comoving coordinate. In general relativity (GR) and in the absence of anisotropic stress, which is the case for late-time ΛCDM cosmologies, the potentials Ψ and Φ are equal. The element dl^2 can be separated into a radial and angular part which for a flat universe $dl^2 = d\chi^2 + \chi^2 d\omega$, where χ is the comoving radial coordinate and ω the angular coordinate.

We then need an equation that describes how the scale factor of the Universe changes. This is the Friedmann equation. For a homogeneous and isotropic Universe it is given by

$$H^2(t) = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{\kappa c^2}{R_0^2 a^2} + \frac{\Lambda}{3}, \tag{2.17}$$

where H is the Hubble parameter, a is again the expansion factor as a function of time, G is the gravitational constant, ρ the total energy density, Λ the cosmological constant, R_0 the radius of curvature and κ the curvature, which is +1 for a positively curved universe, -1 for a negatively curved universe and 0 for a perfectly flat universe. In this thesis, we will consider a flat universe, which is predicted by inflation and confirmed by measurements (e.g., Sánchez et al., 2012; Samushia, Reid, White, Percival, Cuesta, Lombriser, et al., 2013). The density ρ can be split up into separate components of the universe: matter, radiation and dark energy, and usually these components are expressed as density parameters through division by the critical density

$$\rho_{cr} = \frac{3H_0^2}{8\pi G}, \tag{2.18}$$

where the critical density describes the density for which the universe is flat. Assuming an equation of state $P = w\rho$, where $w \sim 0$ for the matter component, $w = 1/3$ for the relativistic component (e.g., radiation) and $w = -1$ for the dark energy, we can write Equation 2.17 as

$$H^2(t) = H_0^2 \left(\Omega_{r,0} a^{-4} + \Omega_{m,0} a^{-3} + \Omega_\Lambda \right), \tag{2.19}$$

where H_0 is the Hubble parameter at the current expansion factor.

We now have the tools to derive the weak lensing formalism. We will consider an example system consisting of a source plane and lens plane, as shown in Figure 2.1. The light travel time along the path from the source to the observer is obtained from Equation 2.16 by setting $ds^2 = 0$, since light rays travel along null geodesics

$$t = \frac{1}{c} \int \left(1 - \frac{2\Psi}{c^2} \right) d\lambda, \tag{2.20}$$

where $d\lambda$ is now the proper coordinate along the (perturbed) light path. This is analogous to the light travel time of a ray propagating trough a medium with refractive index $n = 1 - \frac{2\Psi}{c^2}$. Using Fermat's principle of light taking the shortest optical path we can obtain the Euler-Lagrange equations and after integrating these we obtain the angular difference between the emitted and observed light ray. This deflection angle $\hat{\alpha}$ is given by

$$\hat{\alpha} = -\frac{2}{c^2} \int \nabla_\perp^p \Phi d\lambda, \tag{2.21}$$

where the gradient is taken perpendicular to the path of the light ray. This angle is twice the Newtonian prediction. Equation 2.21 is impractical written in this way, since we have to integrate over the perturbed light path. However, we can simplify Equation 2.21 since $\Phi/c^2 \ll 1$, thus we expect the deflection angle to be small. Therefore we can simply integrate radially over the unperturbed light path dr, obtaining

$$\hat{\alpha} = -\frac{2}{c^2} \int_{-\infty}^{\infty} \nabla_\perp^p \Phi dr. \tag{2.22}$$

The somewhat subtle approximation to integrate over the unperturbed light path dr is called the Born approximation (see e.g., Petri, Haiman, and May, 2017, for the validity of this approximation).

We now define the transverse comoving separation between the source and optical axis as x (see Fig. 2.1). In a flat universe, and under the small angle (θ) approximation, x is given by

$$x(\chi) = \chi\theta. \tag{2.23}$$

From Equation 2.22 we have seen that the deflection of a light ray in the presence of a potential Φ is given by

$$d\hat{\alpha} = -\frac{2}{c^2} \nabla_\perp^p \Phi(x, \chi') d\chi' \tag{2.24}$$

at distance χ' from the observer. From the point of view of the lens, the change in the transverse separation vector at the distance to the source χ_S is

$$dx = (\chi_S - \chi_L) d\hat{a}, \tag{2.25}$$

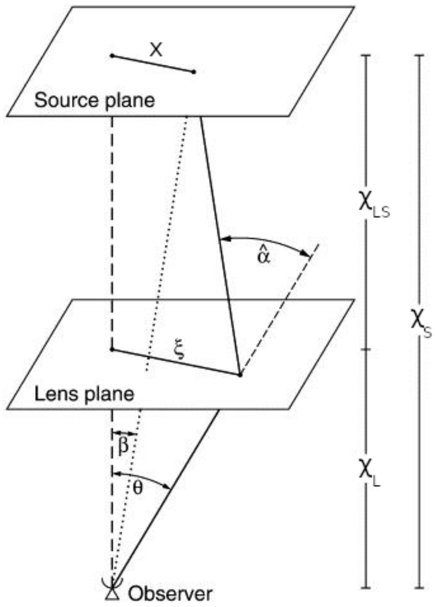


FIGURE 2.1: Example of a gravitational lens system. The unlensed source angle is denoted by β, the observed source angle is denoted by θ. ξ is called the impact factor and x is the comoving separation between the optical axis and the source position. Figure adapted from M. Bartelmann and Schneider (2001)

which is proportional to the distance between the lens and the source. Integrating this expression gives the transverse comoving separation as a function of χ_S

$$x(\chi_S) = \chi_S\theta - \frac{2}{c^2}\int_0^{\chi_S}\frac{\chi_S - \chi'}{\chi_S}\nabla_\perp\Phi(x,\chi')d\chi'. \tag{2.26}$$

The angle $\beta = x/\chi_S$ is the angle of the source which would be observed in the absence of a lens. The reduced deflection angle α, defined as the angle between the unlensed source position and the apparent source position is then

$$\alpha = \theta - \beta. \tag{2.27}$$

Equation 2.27 is called the lens equation. Using the lens equation, we arrive to the expression of the reduced deflection angle α

$$\alpha = \frac{2}{c^2}\int_0^{\chi_S}\frac{\chi_S - \chi'}{\chi_S}\nabla_\perp\Phi(x,\chi')d\chi'. \tag{2.28}$$

In the Born approximation, we can write the reduced deflection angle α as the gradient of a two-dimensional potential ψ, called the lensing potential.

$$\psi(\theta,\chi_S) = \frac{2}{c^2}\int_0^{\chi_S}\frac{\chi_S - \chi'}{\chi_S\chi'}\Phi d\chi'. \tag{2.29}$$

This lensing potential holds two important properties. Firstly, of course, the gradient of the lensing potential equals the reduced deflection angle

$$\nabla_x\psi = \alpha(x). \tag{2.30}$$

Secondly, the Laplacian gives twice the convergence

$$\nabla_x^2 \psi(x) = 2\kappa(x), \tag{2.31}$$

where the convergence is the dimensionless surface density

$$\kappa(x) = \frac{\Sigma(x)}{\Sigma_{cr}}. \tag{2.32}$$

Σ_{cr} is the critical surface density, a characteristic quantity that describes the lens system

$$\Sigma_{cr} = \frac{c^2}{4\pi G} \frac{\chi_S}{\chi_L(\chi_L - \chi_S)}. \tag{2.33}$$

As a side note, Σ_{cr} is found by solving the lens equation for axisymmetric lenses for a source at $\theta = 0$ (directly behind the lens). If the average surface density of the lens is Σ_{cr} inside a radius R, an Einstein ring can appear. Note that this implies we are in the strong lensing regime.

We have shown that the first order effect of gravitational lensing is to distort the image position as given by the lens equation. This distortion is not measurable in the case of weak lensing, since since multiple images (i.e., strong lensing) are needed to infer the original source position. The second order effects of gravitational lensing are more important in the weak lensing regime. These effects distort the shape and size of the sources. Formally, the distortion can be quantified by solving the lens equation for all the points within an extended source. However, often the source is much smaller than the angular size on which the properties of the lens change, and thus a locally linear transformation can be defined to map from source to image coordinates. This is parametrized with the Jacobi matrix, expressed as

$$A_{ij} = \delta_{ij} - \partial_i \partial_j \phi, \tag{2.34}$$

where the partial derivatives $\partial_{i,j}$ are defined with respect to $\theta_{i,j}$. From the definition of the lensing potential (2.29), A can be written as

$$A = \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix}. \tag{2.35}$$

Written in this way, the convergence κ and shear $\gamma = (\gamma_1, \gamma_2)$ are defined as combinations of second derivatives of the lensing potential. Convergence multiplies all directions by $(1 - \kappa)^{-1}$, isotropically changing the observed size. Shear stretches the image, by multiplying for example both diagonal directions by $(1 - \gamma_2)^{-1}$ and $(1 + \gamma_2)^{-1}$. An example of these effects is given in Figure 2.2. These second order effects are not measurable on a source-by-source basis either, but can be measured by averaging many observed galaxies and computing the two-point correlation function or power spectrum of the observed galaxy ellipticities.

2.2.2 Lensing power spectrum

In general, the power spectrum or its real space equivalent: the correlation function, is a statistical tool that can be used to quantify correlations in data. We will start by defining the two-point correlation function and then show that the power spectrum is the Fourier transform of the correlation function.

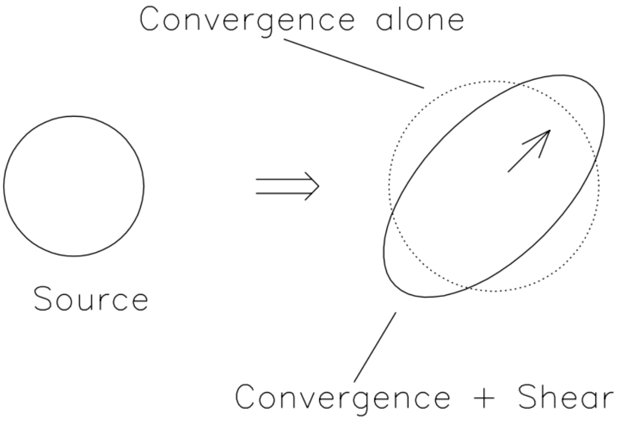


FIGURE 2.2: Distortion due to convergence and shear from gravitational lensing. Figure credit: Narayan and Matthias Bartelmann (1996).

The two-point correlation function of the convergence κ is defined as

$$\xi_\kappa(\phi) = \langle \kappa(\theta)\kappa(\theta+\phi)\rangle \tag{2.36}$$

where the brackets denote the ensemble average. By assuming the ergodic hypothesis, the ensemble average can be replaced by a spatial average over positions θ. Assuming that the Universe is statistically homogeneous and isotropic on large scales, the density field, and thus the convergence are also homogeneous and isotropic. The two-point correlation function is then only a function of the modulus of ϕ and not the direction of ϕ. The two-point correlation function measures the excess probability over random of observing the value of the convergence κ at some point $\theta+\phi$ given that we know the convergence κ at the point θ. If there is no spatial correlation, ξ will be zero for all values of ϕ.

Typically, it is more convenient to work in Fourier space and to use the power spectrum. The Fourier transform of the convergence is

$$\hat{\kappa}(l) = \int \kappa(\theta)\exp(il\theta)d\theta. \tag{2.37}$$

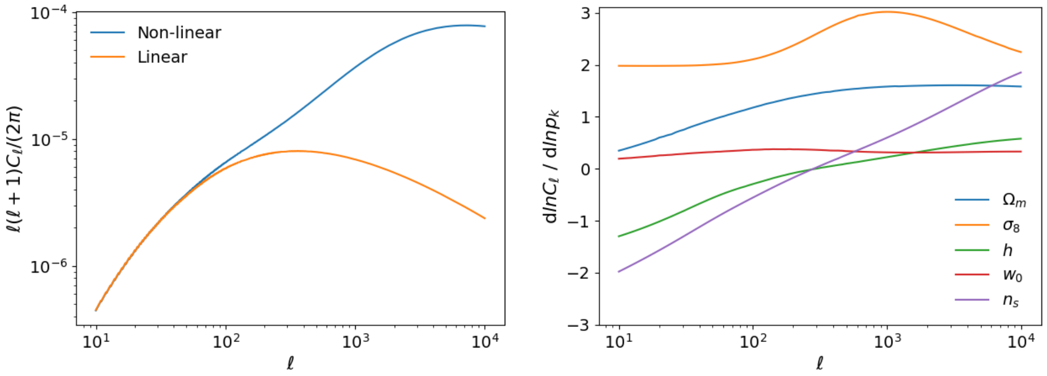
Thus the correlation function in Fourier space is

$$\begin{aligned}\langle \hat{\kappa}(l)\hat{\kappa}^*(l')\rangle &= \left\langle \int \kappa(\theta)\exp(il\theta)d\theta \int \kappa(\theta')\exp(-il'\theta')d\theta' \right\rangle \\ &= \int \exp(il\theta)d\theta \int \langle \kappa(\theta)\kappa(\theta')\rangle \exp(-il'\theta')d\theta'.\end{aligned} \tag{2.38}$$

With $\theta' = \theta + \phi$, and using the assumption that the correlation function is isotropic,

$$\begin{aligned}\langle \hat{\kappa}(l)\hat{\kappa}^*(l')\rangle &= \int \exp(i(l-l')\theta)d\theta \int \xi(\phi)\exp(-il'\phi)d\phi' \\ &= (2\pi)^2\delta_D(l-l')P_\kappa(l).\end{aligned} \tag{2.39}$$

We can see that the power spectrum of the convergence P_κ is the Fourier transform of the two-point correlation function. The same holds for the power spectrum of the shear or any other variable where the same assumptions hold as given above.



(A) (B)

FIGURE 2.3: (A): The theoretical weak lensing power spectrum, with linear and non-linear collapse models. (B): Numerically calculated derivatives of the weak lensing power spectrum with respect to different parameters p.

To get a more physically interesting expression for the power spectrum, we express the convergence κ in terms of the over-density $\delta = \rho/\overline{\rho} - 1$ with the following expression (Kilbinger, 2015)

$$\kappa(\theta, \chi) = \frac{3H_0^2\Omega_m}{2c^2}\int_0^{\chi}\frac{\chi - \chi'}{a(\chi')\chi}(\chi')\delta(\chi\theta, \chi')d\chi'. \tag{2.40}$$

Weighing this expression by a distribution of source galaxies in redshift space $n(\chi)d\chi = n(z)dz$ (up to some χ_{lim}), we obtain the mean convergence

$$\begin{aligned}\kappa(\theta) &= \int_0^{\chi_{lim}} n(\chi)\kappa(\theta, \chi)d\chi \\ &= \frac{3H_0^2\Omega_m}{2c^2}\int_0^{\chi_{lim}}\frac{q(\chi)}{a(\chi)}\chi\delta(\chi\theta, \chi)d\chi.\end{aligned} \tag{2.41}$$

Where we have defined the lensing efficiency q as

$$q(\chi) = \int_{\chi}^{\chi_{lim}} n(\chi')\frac{\chi' - \chi}{\chi'}d\chi'. \tag{2.42}$$

Taking Equation 2.41 and putting it into Equation 2.39 to solve for the convergence power spectrum, the matter power spectrum appears on the right-hand side. Thus the convergence power spectrum can be expressed as (Kilbinger, 2015)

$$C_\kappa(l) = \frac{9}{4}\Omega_m^2\left(\frac{H_0}{c}\right)^4\int_0^{\chi_{lim}}\frac{q^2(\chi)}{a^2(\chi)}P_\delta(k = \frac{l}{\chi}, \chi)d\chi. \tag{2.43}$$

A couple of approximations have been made to obtain Equation 2.43. The Limber approximation (Limber, 1953; Kaiser, 1992) is used to project the 3D power spectrum to a 2D power spectrum. To do this projection, the flat sky and small angle approximation are used. Finally, source-source and source-lens clustering are ignored (see Schneider, van Waerbeke, and Mellier, 2002; Hamana et al., 2002).

The power spectrum of the shear γ is identical to the power spectrum of the convergence. Thus, when measuring the weak lensing power spectrum, the shear is often calculated instead of the convergence. This is simplest in real-space, in which the shear two-point correlation function can be estimated. Conversely, when simulating data it is often simpler to work in Fourier space and to calculate the power spectrum.

The calculations of the weak lensing power spectra are done using the *CosmoSIS* package, as explained in Section 3.3. With this code, we calculate an example angular power spectrum with the parameter values from Planck Collaboration, Aghanim, et al. (2018), which is plotted in Figure 2.3a. The figure shows both the power spectrum calculated with the linear collapse model and the non-linear collapse model, which becomes important at smaller scales (large ℓ). The sensitivity of the power spectrum with respect to various cosmological parameters is shown in Figure 2.3b. This figure shows the numerically calculated derivatives with respect to the matter density Ω_m, the amplitude of the linear power spectrum σ_8, the Hubble parameter h, the dark energy equation of state w and the scalar spectral index n_s. The derivatives are calculated with the central difference method

$$\frac{\partial C_\ell}{\partial p} \approx \frac{C_\ell(p+h) - C_\ell(p-h)}{2h}. \tag{2.44}$$

$C_\ell(p)$ denotes the power spectrum calculated at parameter p. We have used $h = 0.002$ for Ω_m and σ_8, since the power spectrum is most sensitive to these parameters, and $h = 0.01$ for the other parameters. While the central difference method is not very accurate with these step sizes, it does serve as a good first order approximation.

Figure 2.3b shows that the parameter that the spectrum is most sensitive to is σ_8. The parameter σ_8 is defined as the root mean square mass fluctuations in $8h^{-1}$Mpc spheres. The matter density parameter Ω_m is the second most influential parameter. Both of these parameters introduce an amplitude change of the power spectrum. Therefore, usually $S_8 = \sigma_8\sqrt{\Omega_m/0.3}$ is the inferred parameter in cosmic shear studies, since σ_8 and Ω_m are degenerate (e.g., Hildebrandt et al., 2017). The amplitude change of the matter power spectrum can intuitively be understood as follows. As the matter density of the universe increases, gravity will be more influential, increasing the amplitude of the power spectrum on all scales. An increase in σ_8 increases the 'clumpiness of the Universe'. This is because for higher σ_8, the initial density perturbations in the Universe are more likely to reach the overdensity threshold for collapse, as the fluctuations are stronger. This will increase the number of collapsed halos at all scales almost equally, also producing a scaling of the matter power spectrum. The dark energy equation of state parameter also causes a slight change in amplitude, but as can be seen in Figure 2.3b, the power spectrum is not very sensitive to this. Finally, a tilt in the power spectrum is introduced by the parameters n_s and h, as these change the power spectrum as a function of scale.

It is possible to observe angular power spectra in different redshift bins, which is a technique called tomography. This improves the constraints on cosmological parameters inferred from power spectra, allows for measurement of a varying dark energy equation of state, and partly breaks the degeneracy between Ω_m and σ_8.

2.2.3 Tomography

Thus far we have only considered the two dimensional power spectrum and correlation function, which can be used to infer the projected large scale mass distribution. However, more information is available if the redshifts of the galaxies are known. With known redshifts, it is possible to split observed galaxies into redshift bins to obtain multiple two dimensional power spectra. The lowest redshift bin will then only be lensed by the local large scale structure, while galaxies in higher redshift bins will be lensed by structure over a growing range of redshifts. The combination of these lensing effects can then be used to recover information about the 3D distribution of matter. This also probes the time-evolution, since redshift involves both distance and time. Splitting the galaxy distribution up into different redshift bins is called power spectrum tomography.

Defining tomographic redshift bins allows for the calculation of cross and auto power spectra. Splitting the distribution up into n_z redshift bins defines n_z auto spectra and $n_z \times (n_z - 1)/2$ cross spectra, for a total of $n_z(n_z + 1)/2$ power spectra. Similar to Equation 2.43, the weak lensing power spectrum between tomographic redshift bins i and j is given by (Hu, 1999; Takada and Jain, 2004; Alsing and B. Wandelt, 2019)

$$C_{\ell,ij} = \int W_i(\chi) W_k(\chi) \chi^{-2} P_\delta(k = \frac{\ell}{\chi}; \chi) \tag{2.45}$$

where W now denotes the lensing weight function, given by

$$W_i(\chi) = \frac{3\Omega_m H_0^2}{2} \chi \int_\chi^{\chi_H} n_i \frac{\chi' - \chi}{\chi'} d\chi' \tag{2.46}$$

where χ_H is the distance to the Hubble horizon, and $n_i(\chi)$ the galaxy redshift distribution in bin i.

We should note that there are some limitations to this method due to the redshift uncertainties, since the galaxy redshifts are often determined photometrically. If the redshift uncertainties are too large or biased, the information is washed out over different redshift bins (Ma, Hu, and Huterer, 2006). Additionally, the possibility of catastrophic redshift outliers puts stringent constraints on the spectroscopic training samples (Bernstein and Huterer, 2010; Sun et al., 2009). Still, it has been shown that using just a small number of redshift bins significantly improves cosmological parameter constraints (Hu, 1999; Simon, King, and Schneider, 2004).

Chapter 3

Methods

This chapter will explain the methods used in this thesis. It starts by describing techniques often used for likelihood free inference, being Approximate Bayesian Computation and Population Monte Carlo sampling. Then, the information maximizing neural network is described. Finally, we describe shortly the software used for cosmological simulations.

3.1 Likelihood-free inference

In standard Bayesian analyses, a prior is constructed and the likelihood function is known analytically. However, often times in cosmology the exact likelihood function is intractable but forward simulations can be made to generate mock data. The unknown likelihood function can be bypassed by instead comparing forward simulations to the observed data. This section will introduce the methods used for likelihood free inference, which is Approximate Bayesian Computation at its simplest form. Since a large number of simulations is often needed to approximate the likelihood function, techniques such as Population Monte Carlo sampling are often used, which we will explain in this section as well.

3.1.1 Approximate Bayesian Computation

The inference problem can be simply posed as Bayes' theorem (Eq. 2.4), where the probability of the parameters θ given the data x^n is to be evaluated.

$$f(\theta|x^n) = \frac{\mathcal{L}(\theta)f(\theta)}{\int \mathcal{L}(\theta)f(\theta)d\theta}. \tag{3.1}$$

However, the analytical likelihood $\mathcal{L}(\theta)$ is often unknown. Approximate Bayesian Computation (ABC) can be used when the following essential elements are available: a simulator that can generate mock data from a forward model given some parameters, a prior probability distribution over the input parameters, and a distance function between datasets, which often uses summary statistics of the datasets.

First, parameters values are sampled from the prior $f(\theta)$, and these are rejected with a probability that is proportional to $\mathcal{L}(\theta)$. This can be achieved by generating forward simulations with the parameter values and comparing the distance between the observed and simulated data. The fraction of parameter values that have the smallest associated distances from the observed data are then retained, which is the approximation of the posterior distribution function. In practice, a distance threshold ϵ is usually set where simulations are accepted if the distance is smaller than ϵ and rejected otherwise. The likelihood function is thus approximated as (Dutta et al., 2016)

$$\mathcal{L}(\theta) \propto P(\rho(x^s, x) \leq \epsilon). \tag{3.2}$$

where ρ is the distance function, x^s the simulated datasets and x the observed dataset. If the distance ϵ is chosen too small, then an impractically large number of simulations are needed to approximate the posterior. Conversely, if ϵ is chosen too large, then the likelihood is simply approximated by the probability that running the simulator with a value θ produces data within the distance threshold of the observed data. The distance ϵ is usually chosen such that there is a good balance between the computation time and the accuracy of ABC. We should note that because the threshold ϵ cannot be set exactly to zero, the posteriors inferred by ABC are always broader than the exact posteriors, but will be unbiased with small enough ϵ (Alsing, B. Wandelt, and Feeney, 2018).

The distance function is also of crucial importance. Due to the curse of dimensionality, an exponentially increasing number of simulations is needed as the dimension of the data increases. Therefore, the distance between highly dimensional datasets is often calculated as the distance between a few informative summary statistics. Summary statistics are explained in Section 2.1.2. The distance function we use use throughout this thesis is defined by the distance between the summary of some generated dataset x^s and the observed dataset x, weighed by the Fisher information of the simulated summaries.

$$\rho(x^s, x) = \sqrt{(x^s - x)^T F(x^s - x)}, \tag{3.3}$$

This is the optimal, but not unique choice for a distance measure (Alsing and B. Wandelt, 2018).

3.1.2 Population Monte Carlo Sampling

There are a few problems with the simple ABC method. The algorithm rejects many proposed samples when ϵ is small, leading to a very inefficient algorithm. The algorithm does not use the information of the previously accepted samples to update the proposal distribution (initially the prior). Beaumont et al. (2008) proposed a weighing of the *particles* (a dataset realization with parameters θ) that are accepted in the previous step of ABC to update the proposal distribution for the current step of ABC. This approach is called Population Monte Carlo ABC (PMC-ABC).

Following the approach by Hahn et al. (2017), the first step ($t = 1$) of PMC-ABC is identical to the ABC, where a set of particles is drawn from the prior and accepted if the data is within the distance threshold ϵ set by the user. This process is repeated until N particles are accepted. Then equal weights are assigned to the particles $w_1^i = 1/N$. For the following steps ($t > 1$), the distance threshold is decreased. Several approaches can be taken to decrease ϵ. Lin and Kilbinger (2015), for example, set ϵ_t to the median of the distances of the previous step $t-1$, while Charnock, Lavaux, and B. D. Wandelt (2018) set it to the 75th percentile of the previous distances. New particles are drawn from the previous set of particles with probability proportional to their weights. These particles are now perturbed by a kernel $K(\theta_t)$. Like the decreasing of the distance threshold, there is no general consensus on the perturbation kernel. Some examples are setting the kernel as a multivariate Gaussian with zero mean and the covariance of the previous set of particles or uniform kernels centered on the previous set of particles with a certain width σ. For a review on setting the perturbation kernels, see Filippi et al. (2011). The weights of the particles are then updated according to the following rule (Hahn et al., 2017):

$$w_t^i = f(\theta) / \left(\sum_{j=1}^{N} w_{t-1}^i K(\theta_{t-1}^j) \right) \tag{3.4}$$

The process of drawing new particles, perturbing them with a kernel, updating their weights and the distance threshold is repeated until a large number of draws is needed to obtain N accepted particles, as this is a sign the posterior has stopped changing considerably. The ratio of draws needed to draws accepted before stopping the algorithm is called the *criterion* and is an important parameter that has to be defined by the user.

3.2 Information Maximizing Neural Network

This thesis is centered around a novel concept introduced by Charnock, Lavaux, and B. D. Wandelt (2018). The concept Information Maximizing Neural Networks (IMNN) combines data compression with the ideas of deep neural networks. A network is trained to find a (nonlinear) data compression function $f : d \to x$ that maps the data d to compressed summaries x while maximizing the Fisher information content. The function f transforms the original (unknown) likelihood function, to a Gaussian-like likelihood (cf. Eq. 2.14) of the form

$$\ln \mathcal{L}(\mathbf{d}|\theta) = -\frac{1}{2} \left((\mathbf{d} - \mu_f(\theta))^T C_f^{-1} (\mathbf{d} - \mu_f(\theta)) + \ln |2\pi C| \right). \tag{3.5}$$

where $\mu_f(\theta)$ and C_f now denote the mean and covariance of summaries obtained by feeding simulated data, generated at some fiducial parameter values, to the network.

$$\begin{aligned} \mu_f(\theta) &= \frac{1}{n_s} \sum_{i=1}^{n_s} x_i^s \\ (C_f)_{\alpha\beta} &= \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (x_i^s - \mu_f)_\alpha (x_i^s - \mu_f)_\beta. \end{aligned} \tag{3.6}$$

Since f is now a function of the weights and biases of the network, this modification of the likelihood allows for the calculation of an analog to the Fisher information (cf. Eq. 2.15), as a function of the weights and biases of the network

$$F_{\alpha\beta}(\theta) = Tr \left[\left(\frac{\partial}{\partial \theta_\alpha} \mu_f(\theta) \right)^T C_f^{-1} \left(\frac{\partial}{\partial \theta_\beta} \mu_f(\theta) \right) \right]. \tag{3.7}$$

This Fisher information is optimized by simple back-propagation, as will be explained briefly. First, to calculate the Fisher information, the derivative of the mean network output with respect to the input parameters $\frac{\partial}{\partial \theta_\alpha} \mu_f(\theta)$ has to be calculated. As partial derivatives commute with sums, the derivative of the mean output is simply the mean of the derivatives of the network outputs. The task is then to calculate the derivative of the network output with respect to the input parameters.

Two approaches to the calculation of the derivative can be taken. The first approach is calculating the numerical derivative by providing the network with two additional sets of simulations, one just above ($d_i^{s,fid+}$) and one just below ($d_i^{s,fid-}$) the fiducial parameter values. The output of the network at these simulations can then be used to calculate the numerical derivative using the central difference method

$$\left(\frac{\partial x}{\partial \theta_\alpha}\right)_i^{s,fid} \approx \frac{x_i^{s,fid+} - x_i^{s,fid-}}{\Delta\theta_\alpha^+ - \Delta\theta_\alpha^-}, \tag{3.8}$$

where $\Delta\theta^\pm$ are the deviations from the fiducial parameter values. In this approach it is important to set the random seed to the same value when generating the upper and lower simulations to suppress the sample variance. This drastically reduces the amount of simulations needed to approximate the derivative.

The second approach that can be taken is to apply the chain rule, splitting the derivative into the derivative of the output with respect to the input multiplied by the derivative of the input simulations.

$$\mu_{f,\alpha} = \frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{n_d} \frac{\partial x_{ik}^{s,fid}}{\partial d_k}\frac{d_{ik}^{s,fid}}{\partial \theta_\alpha}, \tag{3.9}$$

where k now labels the data point of a simulation. We have found during testing of Gaussian signals (Section 4) that the second method is more stable, but the same seed now has to be set for the upper and lower as well as the central simulations. If all input simulations are not generated at the same seed, the network learns prohibitively slow.

To optimize the Fisher information, we must redefine the output of the network slightly. By construction, the output of the network is the summary x of some input simulation d. However, since the quantity that we want to maximize is in this case the Fisher information, a new 'true' network output is defined. The true network output is defined as the determinant of the Fisher information of n_s simulations fed through n_s identical networks, calculated using the additional simulations needed for the derivatives. A schematic of the network architecture can be viewed in Figure 3.1. The Fisher information content can then be maximized by minimizing the loss function as a function of the true network output $a^L = |F|$. The simplest choice for the loss function is to minimize the negative determinant of the Fisher matrix

$$\frac{\partial \Lambda}{\partial a^L} = -|F|. \tag{3.10}$$

However, a regularization factor is often needed in practice, since the Fisher information is invariant under linear scaling of the summary. To prevent the output summaries artificially increasing to very large values, the determinant of the covariance matrix is a good regularization factor. This penalizes the network when the determinant of the covariance becomes too large. The loss function can now be defined as

$$\frac{\partial \Lambda}{\partial a^L} = -\log|F| + \lambda\sum_{\alpha,\beta}\left(C_{\alpha,\beta} - I_{\alpha,\beta}\right)^2, \tag{3.11}$$

where λ is now a hyper-parameter that sets the strength of the regularization, C the covariance matrix of the output summaries and I the identity matrix. Furthermore, we have switched to maximizing the log determinant of the Fisher matrix since this is often more stable than maximizing the Fisher information directly. An additional advantage of using the covariance matrix in this way is that the regularization term imposes that the summaries are more symmetric and less correlated by weighing the off-diagonal terms of the covariance of the summaries more than the diagonal terms.

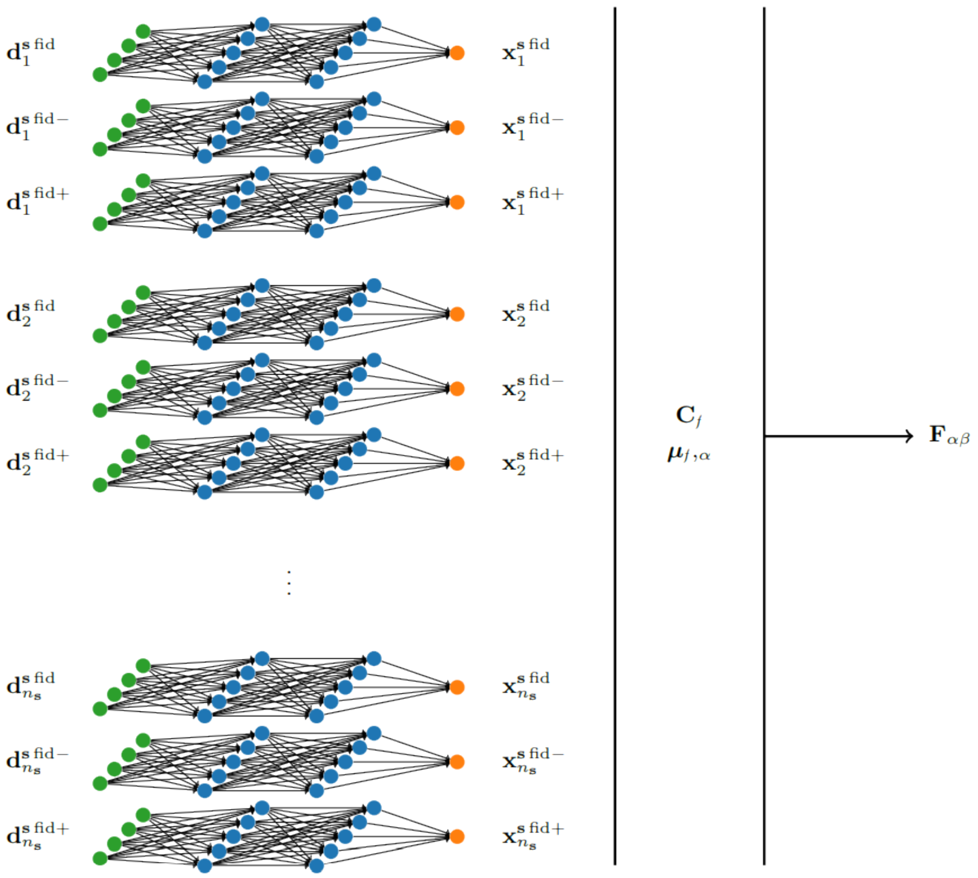

$\mathbf{d}_1^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{d}_1^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{d}_1^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{d}_2^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{d}_2^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{d}_2^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{d}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{d}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{d}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{x}_1^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{x}_1^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{x}_1^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{x}_2^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{x}_2^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{x}_2^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{x}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}}$
$\mathbf{x}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}-}$
$\mathbf{x}_{n_\mathbf{s}}^{\mathbf{s}\,\mathrm{fid}+}$
$\mathbf{C}_f$
$\boldsymbol{\mu}_{f,\alpha}$
$\mathbf{F}_{\alpha\beta}$

FIGURE 3.1: Schematic view of the IMNN. A set of n_s simulations $d_i^{s,fid}$ are passed to the network on each training step, producing n_s output summaries $x_i^{s,fid}$. These summaries are combined to calculate the covariance C_f and the upper and lower simulations are used to calculate the derivative of the mean $\mu_{f,\alpha}$. Finally, the true output of the network is the Fisher information $F_{\alpha,\beta}$. Figure adopted from Charnock, Lavaux, and B. D. Wandelt (2018).

The loss function is minimized using simple back-propagation until the network has converged. The trained network then represents the function $f : d \rightarrow x$ that compresses the data d to the summaries x. The statistical summary of the data can be obtained by feeding it to a network with the final weights and biases. A modified maximum likelihood estimate (MLE) of the parameters of some new input data can be calculated without any additional forward simulations, by interpolating the output summary of the network at the trained parameters, $f(d^{fid}) = x^{fid}$, to the output summary of the network from the new input data. This can be done since we already know $f(d^{fid}) = x^{fid}$ and the derivative $\frac{\partial x}{\partial \theta}|_{\theta_{fid}}$ from the training data. This will of course only be a good approximation when the new input data is close to the data that the network is trained at, due to the fact that we only use the local first derivative to calculate this estimate. The statistical summaries can then be used to carry out for example Approximate Bayesian Computation to infer the full posterior of the parameters that generated the data.

3.3 Cosmological simulations

To generate cosmological observables, in particular the weak lensing angular power spectrum, we make use of *CosmoSIS*[1] (Zuntz et al., 2015). CosmoSIS is a modular code that connects various pieces of cosmological software.

For the calculation of the shear angular power spectra in this thesis we take the following steps. First, we use the *Code for Anisotropies in the Microwave Background*[2] (CAMB) (Lewis, Challinor, and Lasenby, 2000; Howlett et al., 2012). This code evolves initial density perturbations in the Universe to the three dimensional matter power spectrum. It then uses the *Halofit* (Smith et al., 2003, originally) fitting model from Takahashi et al. (2012) to compute the non-linear part of the matter power spectrum. Secondly, we construct a Smail redshift distribution

$$dn/dz \propto z^{\alpha} e^{-(z/z_0)^{\beta}} \tag{3.12}$$

parameterized by α, β and z_0. The photometric redshift error of the observed galaxies is assumed to be Gaussian, with a scatter σ_z and possibly an additive bias b. This distribution is constructed in CosmoSIS. The redshift distribution is then split into n_z tomographic redshift bins. The calculated three dimensional matter power spectrum is projected into the 2D tomographic redshift bins by another native CosmoSIS code. This code assumes the Limber approximation (Limber, 1953) and the flat-sky approximation. Kilbinger et al. (2017) have shown that these approximations are a good approximation, even for stage IV surveys such as Euclid, when $\ell > 10$. The steps as explained here are followed for all the shear angular power spectra shown throughout this thesis.

[1] `http://bitbucket.org/joezuntz/cosmosis`

[2] `https://camb.info/`

Chapter 4

Summarizing Gaussian Signals

This chapter will investigate some relatively simple cases where the information maximizing neural network is trained. This allows us to find cases where the network fails, how to resolve those cases and which problems can in general be expected when trying to summarize complicated datasets. Additionally, since these simple cases can be solved analytically, we can see how well the network performs.

In this chapter, the network is trained to summarize different types of Gaussian signals. We start by summarizing one unknown parameter, the variance, of a one dimensional Gaussian and then add a second parameter, the mean.

4.1 One unknown parameter

One of the simplest toy models we can construct is where we have n data points drawn from a normal distribution with zero mean and unknown variance: $X_i, ..., X_n \sim \mathcal{N}(0, \theta)$. It should be noted that this example is also explored in (Charnock, Lavaux, and B. D. Wandelt, 2018), but in less detail, and we repeat the example deliberately so we can verify that the network works.

The likelihood function (Eq. 2.2) of this model is given by

$$\begin{aligned} \mathcal{L}(\theta) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{X_i^2}{2\theta}\right) \\ &= \frac{1}{(2\pi\theta)^{n/2}} \exp\left(-\frac{1}{2\theta}\sum_{i=1}^{n} X_i^2\right). \end{aligned} \tag{4.1}$$

The log-likelihood is given by

$$\ln \mathcal{L}(\theta) = -\frac{n}{2}\ln(2\pi\theta) - \frac{1}{2\theta}\sum_{i=1}^{n} X_i^2. \tag{4.2}$$

Thus in this case the minimal sufficient statistic is given by $x = \sum_{i=1}^{n} X_i^2$, since this is the only term that depends on the data. The value that maximizes the score relates this sufficient statistic to the variance:

$$0 = \frac{\partial \ln \mathcal{L}(\theta)}{\partial \theta} = \frac{x}{2\theta^2} - \frac{n}{2\theta^2}. \tag{4.3}$$

Thus the sufficient statistic in terms of the variance is given by

$$x = n\theta. \tag{4.4}$$

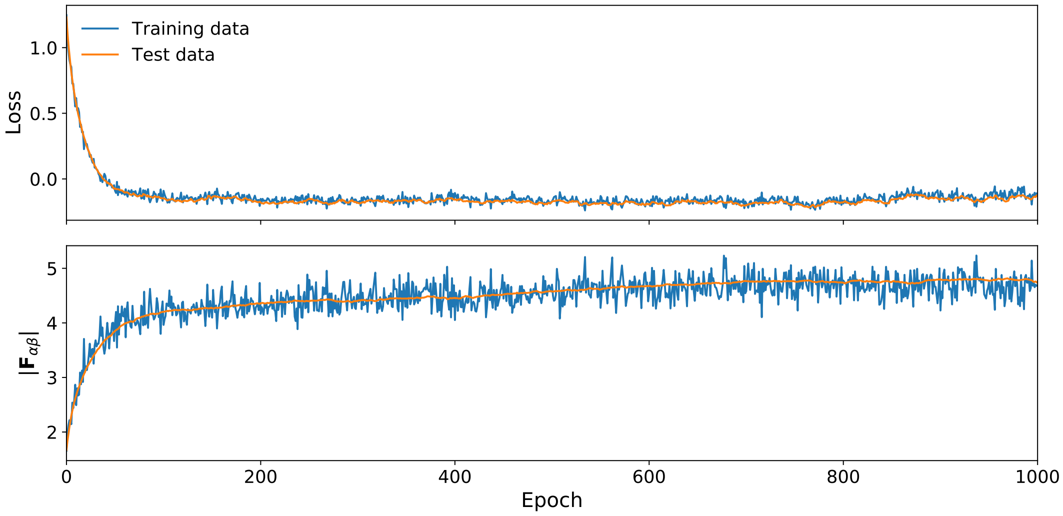


FIGURE 4.1: Fisher information and value of the loss function as a function of training epoch.

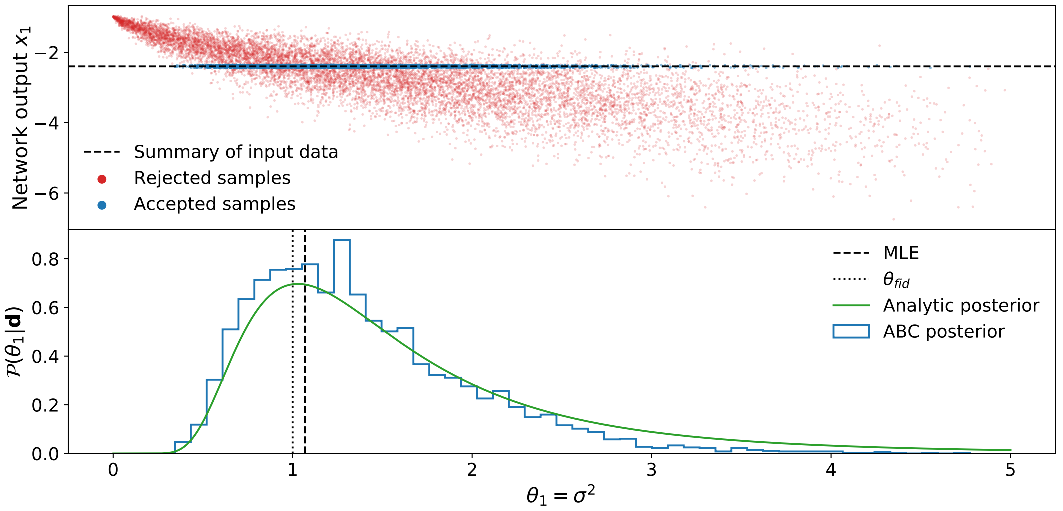


FIGURE 4.2: **top**: Network output summary as a function of the unknown parameter θ. The red points show a random subset of all generated output values, and the blue points show summaries that are closer than some value $\epsilon = 0.1$ to the network output of the input data, shown as the dashed line. **bottom**: Histogram of the approximate posterior. The variance of the input data is shown as the dotted line and the MLE from the network summary as the dotted line

From Equation 2.13, we compute the Fisher information:

$$F = -\left\langle \frac{-x}{\theta^3} + \frac{n}{2\theta^2} \right\rangle = \frac{n}{2\theta^2}. \tag{4.5}$$

which is 5 in the case we will consider, where $n = 10$ and $\theta_{fid} = 1$.

We then move to training the information maximizing network. The network setup that is used consists of two densely connected hidden layers both with 128 nodes. This setup is inspired from Charnock, Lavaux, and B. D. Wandelt (2018), who also used two hidden layers, but we have found that half the number of nodes per layer works equally well. The weights and biases are initialized from a normal distribution with zero mean and standard deviation $\sqrt{2/k^{l-1}}$ where k^l is the number of neurons in layer l (K. He et al., 2015). This is a popular initialization method as it keeps the size of the layers in mind, allowing for more efficient convergence. The network uses 5000 training simulations, but only 1000 simulations at a time to calculate the covariance matrix. To calculate the derivative of the network mean, we generate 1000 upper and lower simulations at $\theta \pm \Delta\theta$, with $\Delta\theta = 0.1$. No dropout is used, but a test set of 1000 simulations is given to the network to track whether the network is overfitting. After training the network for 1,000 epochs with a learning rate of 10^{-2} we obtain a final Fisher information of 4.68 on the training set and 4.75 on the test set, as shown in Figure 4.1. The network has converged to almost the analytical Fisher information of 5 within 200 epochs.

To verify that the network has indeed obtained an informative summary for the variance of the data we perform the ABC method described in Section 3.1.1. The input data, which we want to infer the posterior from, is a noisy realization of 10 data points from a Gaussian with zero mean and unit variance. We use a Gaussian prior with a mean of 1 and variance of 2 truncated at 0 and 5. From this prior, 10^5 samples are drawn and the output summaries of the network are plotted in Figure 4.2. The approximate posterior is generated by accepting samples that are within a distance (defined in Sect. 3.1.1) $\epsilon = 0.1$ from the network output of the input data. This value of ϵ is found by dividing the network output by 20, which we empirically found to be a good first guess of ϵ. The calculated posterior agrees well with the analytical posterior, but we are drawing a lot of points that are far away from the network output of the input data and the ϵ value is chosen quite arbitrarily. To improve on this, we now use the population Monte Carlo method (Section 3.1.2) with the same prior, and imposing that we stop when only 10^4 samples are kept in the approximate posterior (i.e., a *criterion* of 0.1). The result is shown in Figure 4.3. An interesting thing that can be taken away from this figure is that the PMC method finds a slightly worse shape for the posterior than the ABC only algorithm (Fig. 4.2). However, the maximum a posteriori probability (i.e., the mode of the posterior) is still the same in both cases. The exact shape of the posterior is dependent on the value of ϵ when doing ABC, and dependent on the stopping criterion (the ratio of draws needed to draws accepted) when doing PMC. This implies that if a good approximation of the full posterior function is warranted, the criterion and distance thresholds are still important hyper-parameters which have to be tuned by hand.

We found that the loss function steadily declines for the learning rates 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5}, with the only difference being the number of epochs required to converge to a final Fisher information near 5. The $\Delta\theta$ parameter was varied in [0.001, 0.01, 0.1, 0.3, 0.5] and we found that all of these parameters work about equally well. For various other values of the variance, we explored $\theta_{fid} = [2, 3, 4, 5]$. This of course changes the analytical Fisher information available to [1.25, 0.56, 0.32, 0.2] respectively, which we found the network approximates correctly as well.

We also investigated different values of the number of input data points n. When changing the number of input data points, the Fisher information changes too. Since the variance is chosen to be unity, the Fisher information scales linearly with the number of input data points (Eq. 4.5). The number of data points was varied in $n = [10, 100, 1000, 10000]$. For $n = 100$, the network already struggles to learn the problem. It finds a Fisher information on the training set of around 50 (the analytical value), but only approaches around $F = 26$ on the test set. After more testing, we found that an increasing amount of simulations is needed for the network to learn the problem with more data points. This is because providing more data points does not add much information, and thus we are actually increasing the difficulty of the

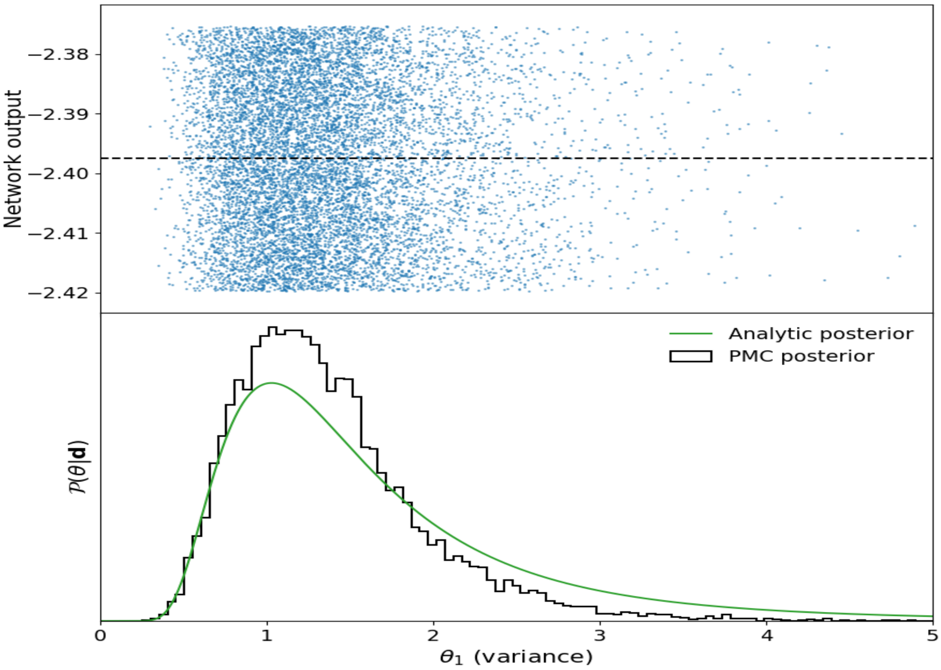


FIGURE 4.3: Results of PMC-ABC using the network to summarize a Gaussian signal with unit variance. The top panel shows the input data summary as the dashed line and the generated datasets by the PMC algorithm as the blue dots. The bottom panel shows the approximate posterior inferred from binning the generated datasets in the top panel.

problem. Therefore, the network needs more training examples to distinguish the important features from the data and learn the summary statistic correctly.

The number of simulations used to approximate the covariance of the network outputs n_s was found to be an important parameter. We varied $n_s = [100, 200, 400, 600, 1000]$ and we found that for less than 400 simulations, the network is not able to learn the problem due to not being able to approximate the covariance matrix correctly. When splitting the calculation of the covariance matrix up into 5 smaller sets of simulations, the network is able to learn the problem with 100 simulations at a time. This splitting provides variations in the derivative of the mean $\mu_{f,\alpha}$ and the covariance C of the output of the network, which allows the network to learn the problem slightly faster with less amounts of total simulations.

The most important parameter was found to be the number of simulations used to calculate the derivative of the network outputs, this was varied in [50, 100, 200, 500, 1000] and we found that for less than 1000 simulations to calculate the derivative, the derivative is not approximated correctly and the network is unable to learn the summary. This is result is quite unexpected, considering that we set the same random seed for the central, upper and lower simulations. One would expect that since these simulations are seeded, the derivative is easily approximated and less perturbed simulations are needed than the central simulations, but apparently this is not the case. Interestingly, due to the way the network is set up, the number of derivative simulations was also lower when lowering the amount of simulations used to approximate the covariance of the output. However, in these versions of the network (See Table 4.1, version 40-44), the number of derivative simulations did not hinder the ability to learn. From these two setups, it seems that the number

TABLE 4.1: Results of the training the network as function of various parameters. n_s refers to the number of central simulations and n_d the number of simulations slightly above and below the fiducial parameters.

Version	Learning rate	num epochs	splits	Δ_θ	n_s	θ_{fid}	n_d	input shape	activation	Final detF train	Final detF test
1	0.1	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	NaN	NaN
2	0.01	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.92	4.93
3	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.81	4.32
4	0.0001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.51	4.29
5	1E-05	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	3.41	3.16
10	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.53	4.86
11	0.001	5000	5	[0.1]	1000	[1.0]	1000	[100]	leaky_relu	50.17	27.34
12	0.001	5000	5	[0.1]	1000	[1.0]	1000	[1000]	leaky_relu	3335.04	38.44
13	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10000]	leaky_relu	1904.67	14.01
20	0.001	5000	5	[0.001]	1000	[1.0]	1000	[10]	leaky_relu	5.12	4.54
21	0.001	5000	5	[0.01]	1000	[1.0]	1000	[10]	leaky_relu	4.93	4.42
22	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.89	5.03
23	0.001	5000	5	[0.3]	1000	[1.0]	1000	[10]	leaky_relu	4.99	5.09
24	0.001	5000	5	[0.5]	1000	[1.0]	1000	[10]	leaky_relu	5.54	5.38
30	0.001	1000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.08	4.55
31	0.001	10000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.66	4.37
40	0.001	5000	5	[0.1]	100	[1.0]	100	[10]	leaky_relu	7.81	5.16
41	0.001	5000	5	[0.1]	200	[1.0]	200	[10]	leaky_relu	7.05	4.33
42	0.001	5000	5	[0.1]	400	[1.0]	400	[10]	leaky_relu	5.77	4.28
43	0.001	5000	5	[0.1]	600	[1.0]	600	[10]	leaky_relu	5.01	5.04
44	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.69	4.69
50	0.001	5000	1	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.84	4.34
51	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.1	4.77
52	0.001	5000	10	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.47	4.62
53	0.001	5000	20	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.8	4.31
54	0.001	5000	50	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	4.81	4.95
60	0.001	5000	5	[0.1]	1000	[1.0]	50	[10]	leaky_relu	1.24	0.19
61	0.001	5000	5	[0.1]	1000	[1.0]	100	[10]	leaky_relu	0.73	0.45
62	0.001	5000	5	[0.1]	1000	[1.0]	200	[10]	leaky_relu	0.85	0.62
63	0.001	5000	5	[0.1]	1000	[1.0]	500	[10]	leaky_relu	1.18	0.63
64	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.2	4.68
70	0.001	5000	5	[0.1]	1000	[1.0]	1000	[10]	leaky_relu	5.03	5.17
71	0.001	5000	5	[0.1]	1000	[2.0]	1000	[10]	leaky_relu	1.3	1.2
72	0.001	5000	5	[0.1]	1000	[3.0]	1000	[10]	leaky_relu	0.61	0.55
73	0.001	5000	5	[0.1]	1000	[4.0]	1000	[10]	leaky_relu	0.3	0.28
74	0.001	5000	5	[0.1]	1000	[5.0]	1000	[10]	leaky_relu	0.21	0.18

of derivative simulations cannot be a small fraction of the number of simulations to approximate the covariance, which contradicts the statement made in Charnock, Lavaux, and B. D. Wandelt (2018) that relatively few extra simulations are needed to approximate the covariance. This might be explained by the fact that we use the chain rule (Eq. 3.9) to calculate the derivative of the network output, and Charnock, Lavaux, and B. D. Wandelt (2018) use the central difference method (Eq. 3.8). With the chain rule method, the derivative of the output summaries with respect to the data $\frac{\partial x}{\partial d}$, is calculated at the central simulations, while the derivative of the data with respect to the parameters $\frac{\partial d}{\partial \theta}$ is calculated by using the upper and lower simulations. If these two terms are not calculated with same amount of simulations, the uncertainty in the terms will differ, which might cause the incorrect approximation of the derivative.

Finally, as Table 4.1 shows, for many different hyper-parameters and random initializations of the network, the analytical Fisher information of 5 was indeed approached closely. This indicates that not much fine-tuning is needed to find a network that summarizes the information correctly.

4.2 Two unknown parameters

As a second step, we move to multiple unknown parameters. In this case, we draw n data points drawn from a Gaussian distribution with unknown mean and variance to see if the network can learn to summarize both parameters. We define $\theta = (\theta_1, \theta_2)^T$, where $\theta_1 = \mu$ and $\theta_2 = \sigma^2$. This problem is also easily solved analytically, but we investigate it to see how the network performs on multiple parameters. The likelihood is given by (Eq. 2.2)

$$\mathcal{L} = \prod_i^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(d_i - \theta_1)^2}{2\theta_2} \right]. \tag{4.6}$$

From which the log-likelihood can be found

$$\ln \mathcal{L} = -\frac{n}{2} \ln(2\pi\theta_1) - \sum_i^n \frac{(d_i - \mu)^2}{2\theta_2}. \tag{4.7}$$

There is a single sufficient statistic which describes the mean and a single sufficient statistic which describes the variance. These summaries can be found by finding the maximum of the likelihood. We can calculate the analytical Fisher information by taking the negative of the second derivative of the likelihood function (Eq. 4.6), and then taking the expectation by evaluating this at the fiducial parameter values (cf. Eq. 2.13). The first derivative of the log-likelihood with respect to the mean is given by

$$-\sum_i^n \frac{2\mu - 2d_i}{2\theta_2}. \tag{4.8}$$

If we differentiate this with respect to θ_2, we obtain a function of $(d_i - \mu)$ of which the expectation value is zero and therefore the diagonal elements of the Fisher information matrix are zero. The upper left element of the matrix is obtained by again differentiating the score with respect to the mean, and the lower right element of the matrix was already calculated in Equation 4.5. Thus, we obtain a Fisher information matrix of

$$F = \begin{bmatrix} \frac{n}{\theta_2} & 0 \\ 0 & \frac{n}{2\theta_2^2} \end{bmatrix} \tag{4.9}$$

If we choose $n = 10$ data points drawn from a fiducial mean of $\theta_1^{\text{fid}} = 0$ and variance of $\theta_2^{\text{fid}} = 1$ the analytical Fisher information is $n^2/2\theta_2^3 = 50$.

This problem was also explored as a function of learning rate, input shape, $\Delta\theta$, number of simulations to approximate the covariance, number of splits, and number of derivative simulations. The most important parameter was again number of simulations used to approximate the derivative. 1000 simulations were needed to approximate the derivative correctly. The same network setup is used as in the previous section. After training for 500 epochs, the learning curve is shown in Figure 4.4, where the analytical Fisher information of 50 is well approached, but again not quite reached on the test set. The output of using the summaries to do approximate Bayesian computation is shown in Figure 4.5, inferring the parameters of an input dataset generated at $\mu = 0, \sigma^2 = 1$. We have drawn 10^5 datasets from a multivariate Gaussian prior with mean $[1, 2]$, and diagonal covariance $[2, 2]$, truncated in the first dimension as $-3 < \mu < 3$ and in the second dimension as $0 < \sigma^2 < 5$. The prior is purposely offset from the input parameters now, to see whether the ABC method still

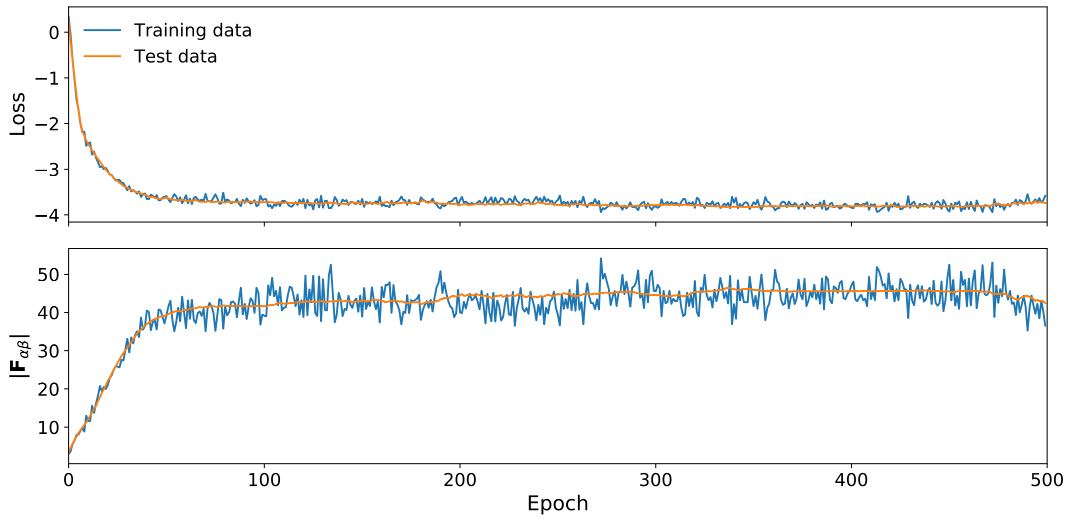


FIGURE 4.4: Fisher information and value of the loss function as a function of training epoch for a network that summarizes Gaussian data with unknown mean and variance.

works even when the prior is chosen away from the parameters, and to make sure we are not just sampling the prior. The prior mean is still within 1σ of the input parameter values however, since we would otherwise not be sampling the interesting part of the parameter space. It is interesting to see that in Figure 4.5, the first output summary is a quadratic function of the mean. This shows that the network does indeed find non-linear mappings between the input data and the summary statistics. The 2D posterior is also plot in Figure 4.6, which shows that the input parameters are in good agreement with the posterior inferred from ABC, and that we are indeed not simply sampling the prior distribution.

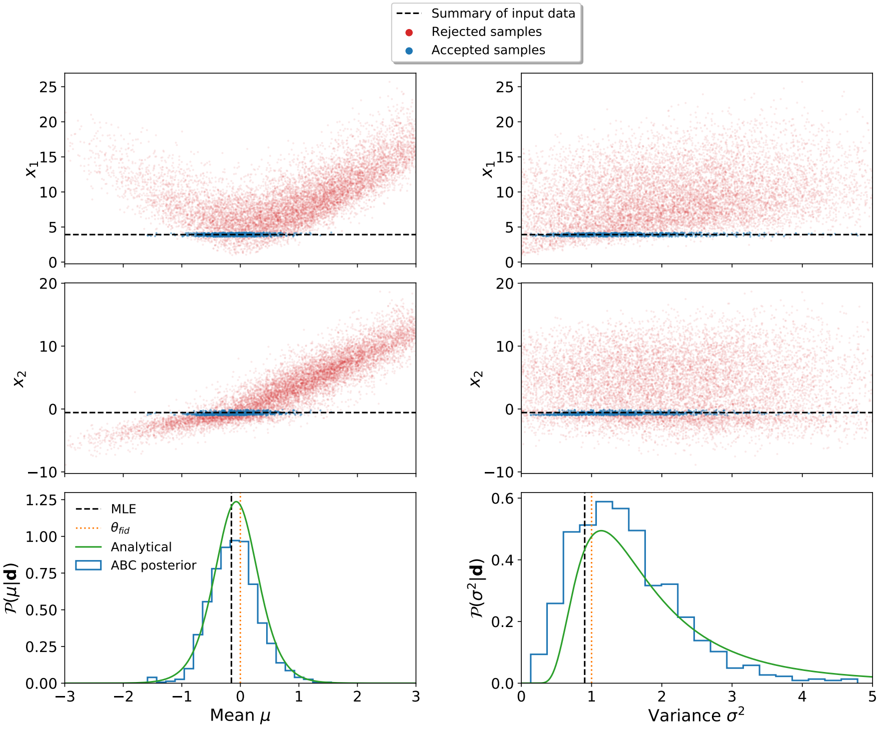


FIGURE 4.5: Equivalent to Fig. 4.2, but for two unknown parameters and thus two output summaries. A random subset of a tenth of the 10^5 generated summaries is plotted here. Both output summaries are strong functions of the mean μ (left), while a large scatter is visible as function of the variance σ^2 (right).

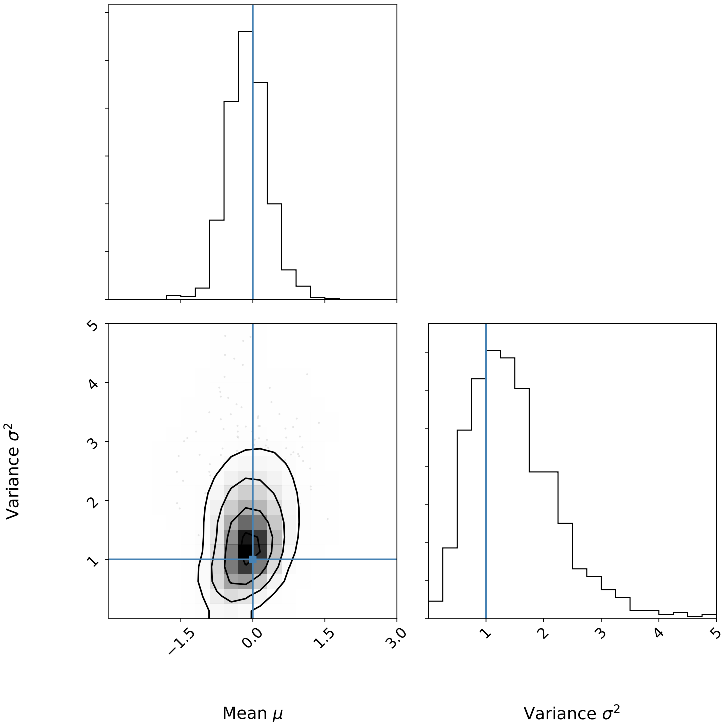


FIGURE 4.6: Two dimensional posterior inferred from approximate Bayesian computation using the summaries generated by the network. The input parameter values are indicated as blue lines.

Chapter 5

Summarizing Weak Lensing Data Vectors

5.1 One parameter

In this chapter, we build up the complexity of the problem gradually. We start simple in this section by not yet employing tomographic redshift bins and investigate the behaviour of the network for a single free parameter: Ω_m. The weak lensing power spectra, C_ℓ, are generated with *CosmoSIS*, as explained in Section 3.3 and the definition of the weak lensing power spectrum was derived in Section 2.2. We assume a Euclid-like survey without any complex masking or boundary effects, for simplicity.

To calculate the weak lensing power spectrum, we must specify the source redshift distribution $n(z)$. For Euclid, the source redshift distribution is known (Laureijs et al., 2011; Amendola, Appleby, D. Bacon, et al., 2013; Schaan et al., 2017) and given by the following equation

$$dn/dz \propto z^\alpha e^{-(z/z_0)^\beta} \tag{5.1}$$

with $\alpha = 1.3$, $\beta = 1.5$ and $z_0 = 0.65$. The total number of sources observed per area of the sky is $n = 30$ arcmin^{-2} and the total area Euclid will cover is 15,000 deg^2. We use redshift bounds $0 \leq z \leq 2$ and normalize the distribution to match the expected number of observed sources. The shape noise, due to observing a finite sample of galaxy ellipticities, is assumed to be $\sigma_e = 0.26$, following Schaan et al. (2017). We define the observed power spectrum as the theoretical power spectrum contaminated by shape noise as follows

$$C^{obs}(\ell) = C(\ell) + \frac{\sigma_e^2}{n} \tag{5.2}$$

where n is the observed number of galaxies in the survey. The redshift of the sources is in practice not known exactly, but is determined via photometric redshifts. It is common in the literature to assume the galaxies have a photometric redshift z_{ph} distribution $\mathcal{P}(z_{ph}|z)$ that is a Gaussian function of the true redshift z. This distribution is usually assumed to be a Gaussian with mean z, a scatter $\sigma_z(z)$ and possibly a bias $b_z(z)$. For a Euclid-like survey, typical parameters that are used are $\sigma_z = 0.05(1+z)$ and $b_z(z) = 0$, which we adopt here as well.

The observed power spectra will be noisy realizations of the theoretical power spectrum. To simulate this effect, we calculate the expected covariance matrix of the observed lensing power spectrum, which is given by (Scoccimarro, Zaldarriaga, and

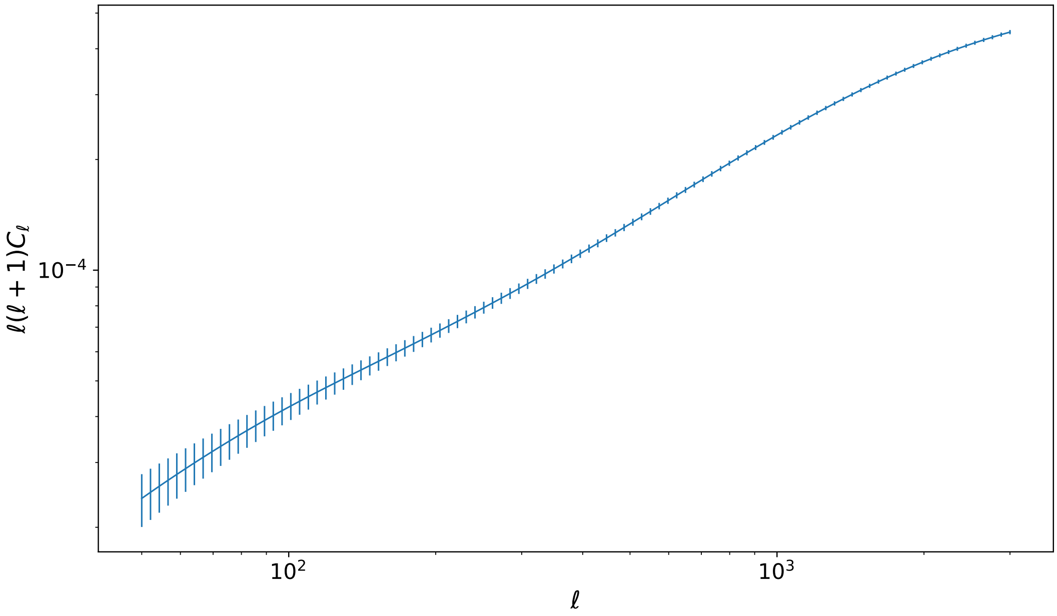


FIGURE 5.1: Theoretical power spectrum calculated for a Euclid-like survey. The error bars show the magnitude of the diagonal Gaussian covariance matrix.

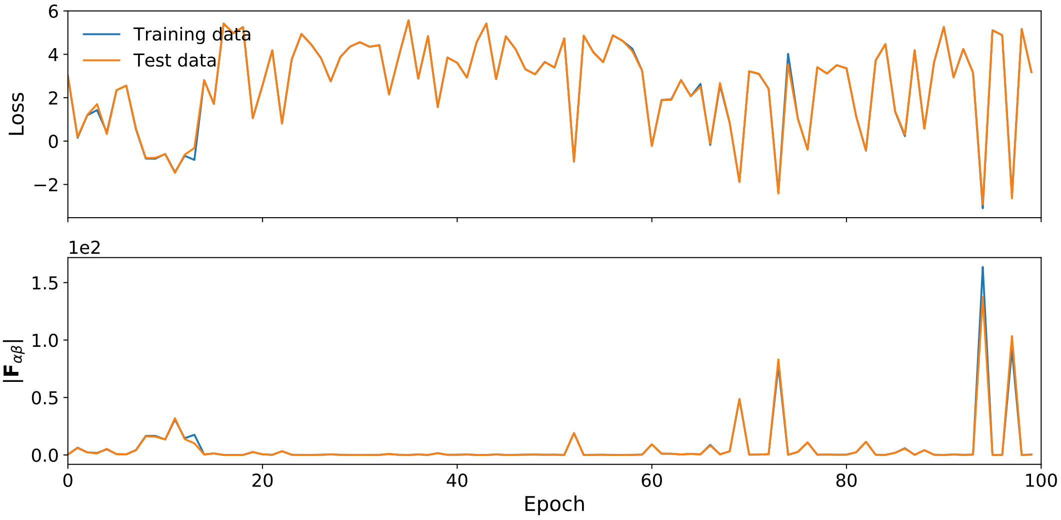


FIGURE 5.2: Initial behaviour of the loss function and Fisher information when training the information maximizing neural network without modifying the input data. It is obvious that the network is not learning.

Hui, 1999)

$$\begin{aligned} Cov[C^{obs}(\ell_a), C^{obs}(\ell_b)] &= \langle C^{obs}(\ell_a) C^{obs}(\ell_b)\rangle - \langle C^{obs}(\ell_a)\rangle \langle C^{obs}(\ell_b)\rangle \\ &= \frac{2C^{obs}(\ell_a)C^{obs}(\ell_a)\delta^K_{ab}}{(2\ell_a+1)\Delta\ell f_{\text{sky}}} + Cov^{NG} \end{aligned} \tag{5.3}$$

where the first term on the right-hand side is the Gaussian contribution and the second term the non-Gaussian contribution. In the Gaussian limit, all Fourier modes are independent random variables, and thus uncorrelated. This is imposed by the Kronecker delta. The number of modes at a given ℓ is $(2\ell+1)\Delta l f_{\text{sky}}$, where f_{sky} is the fraction of the sky observed by Euclid. For simplicity, we will only consider the Gaussian part of the covariance in this study. For an analysis of the impact of the non-Gaussian part of the covariance see e.g., Takada and Jain (2009). The general covariance matrix of the power spectrum C_ℓ, would be a $n_\ell \times n_\ell$ dimensional matrix, where n_ℓ indicates the amount of ℓ modes, but because in the Gaussian assumption different modes are uncorrelated, the off-diagonal terms of this matrix are zero, and we can realize the noisy power spectra from one-dimensional normal distributions with variance Cov_ℓ as given in Equation 5.3

$$C_\ell \sim \mathcal{N}(C_\ell^{obs}, Cov_\ell). \tag{5.4}$$

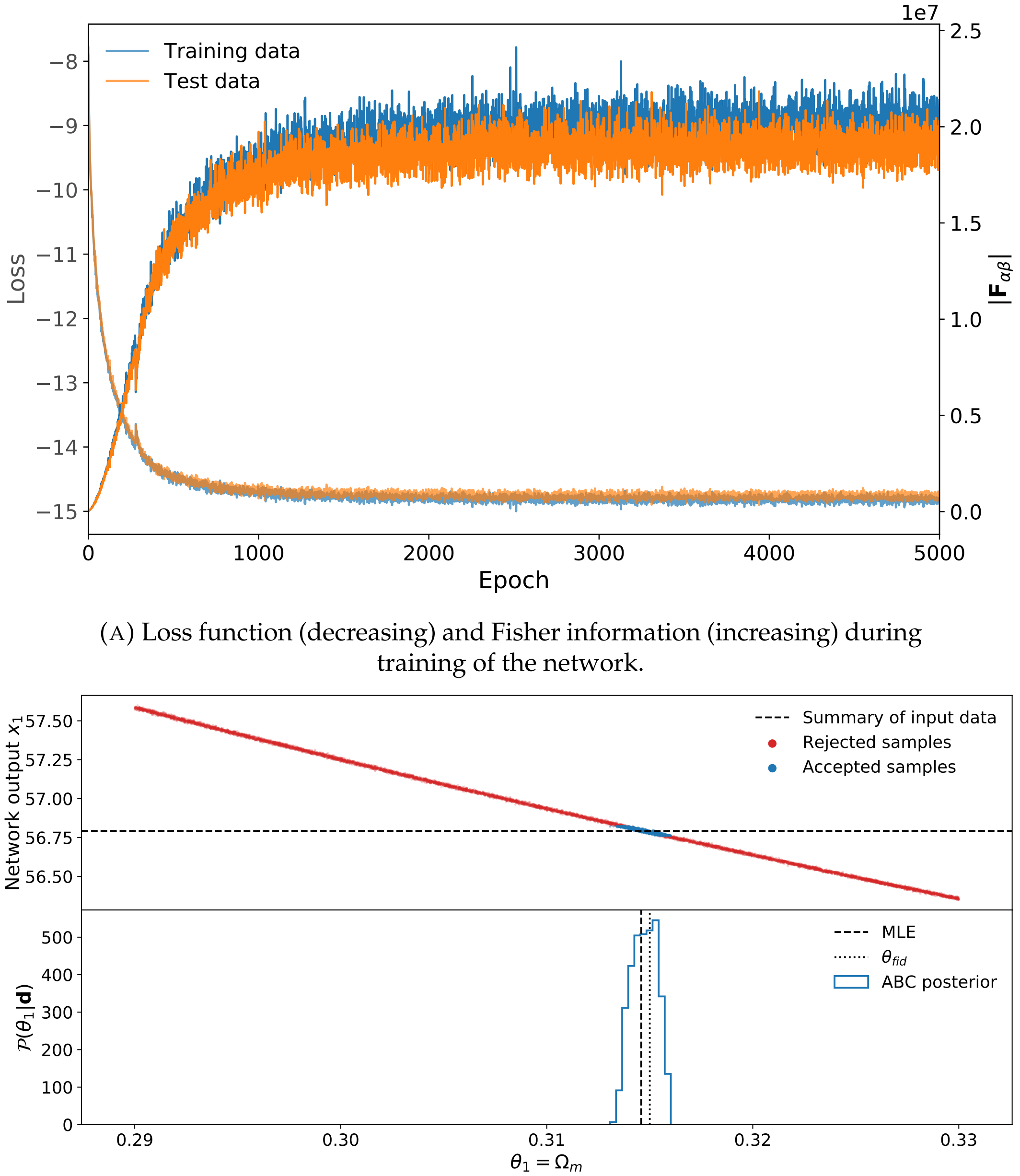
We compute the power spectrum at $n_\ell = 100$ equally log-spaced samples, between $\ell = 50$ and $\ell = 3000$ following Takada and Bridle (2007). We base our fiducial parameters at which the training data will be generated on Alsing and B. Wandelt (2019), but define the amplitude of the power spectrum in terms of A_s instead of σ_8. The fiducial cosmological parameters are $A_s = 2.1 \times 10^{-9}$, $\Omega_m = 0.315$, $\Omega_b h^2 = 0.0244$, $h = 0.674$, $n_s = 0.965$, $w_0 = -1.03$. The theoretical power spectrum calculated for these parameters is plotted in Figure 5.1. The figure shows the 1σ Gaussian error bars calculated with Equation 5.3. We can see that at small ℓ the sample variance dominates, and that we are not yet probing the small physical scales where the shape noise becomes important.

The setup for the information maximizing neural network is mainly the same as the setup that was used in Chapter 4. The number of simulations used at once to approximate the covariance of the network outputs is 1000, and we again use 5 different training splits, for a total of 5000 simulations. For the derivative of the network outputs, we feed 1000 simulations at $\Omega_m \pm \Delta\theta$ with $\Delta\theta = 0.02$. The test set totals 1000 simulations at the fiducial Ω_m and 200 simulations for the derivative. We use the same architecture of the network that was found to be suitable for summarizing Gaussian noise in Chapter 4, a densely connected network of two layers, but we now set 256 nodes per layer, as our input data-vector is larger. The learning rate parameter is set to 10^{-4} after testing the behaviour of the neural network for an initially small amount of epochs and the leaky *ReLu* activation function is used. Unless otherwise specified, this setup will be used throughout the rest of this work.

Initially, the Fisher information calculated by the network was not improving, and the behaviour of the loss function was quite random, as Figure 5.2 shows. This is because the data had not yet been standardized, so the values of the power spectrum, which are of the order $\sim 10^{-9}$, will be in the machine round-off error regime when combining them with the initial values of the weights and biases of the network, which have around values of order unity. Additionally, logarithmic space is a more natural space to define the power spectrum in, as is evident from Figure 5.1. We thus imposed the standardization of the simulated data d_i^s by taking the natural logarithm of the data.

$$\vec{d'}_i^s = \ln(\vec{d}_i^s). \tag{5.5}$$

This standardization resulted in the learning curve given in Figure 5.3a. This figure shows that the network converges to a Fisher information of around 2×10^7 within 2000 epochs, after which the loss function does not change significantly anymore. The trained network is now the function that compresses an input data vector of dimension 100 to a single number. We use the summaries calculated by the trained network to perform Approximate Bayesian Computation (ABC). The input data to infer the posterior of is a noisy simulation at $\Omega_m = 0.315$. A uniform prior over Ω_m in the range $[0.29, 0.33]$ is used because while ABC is rather inefficient, *CosmoSIS* allows

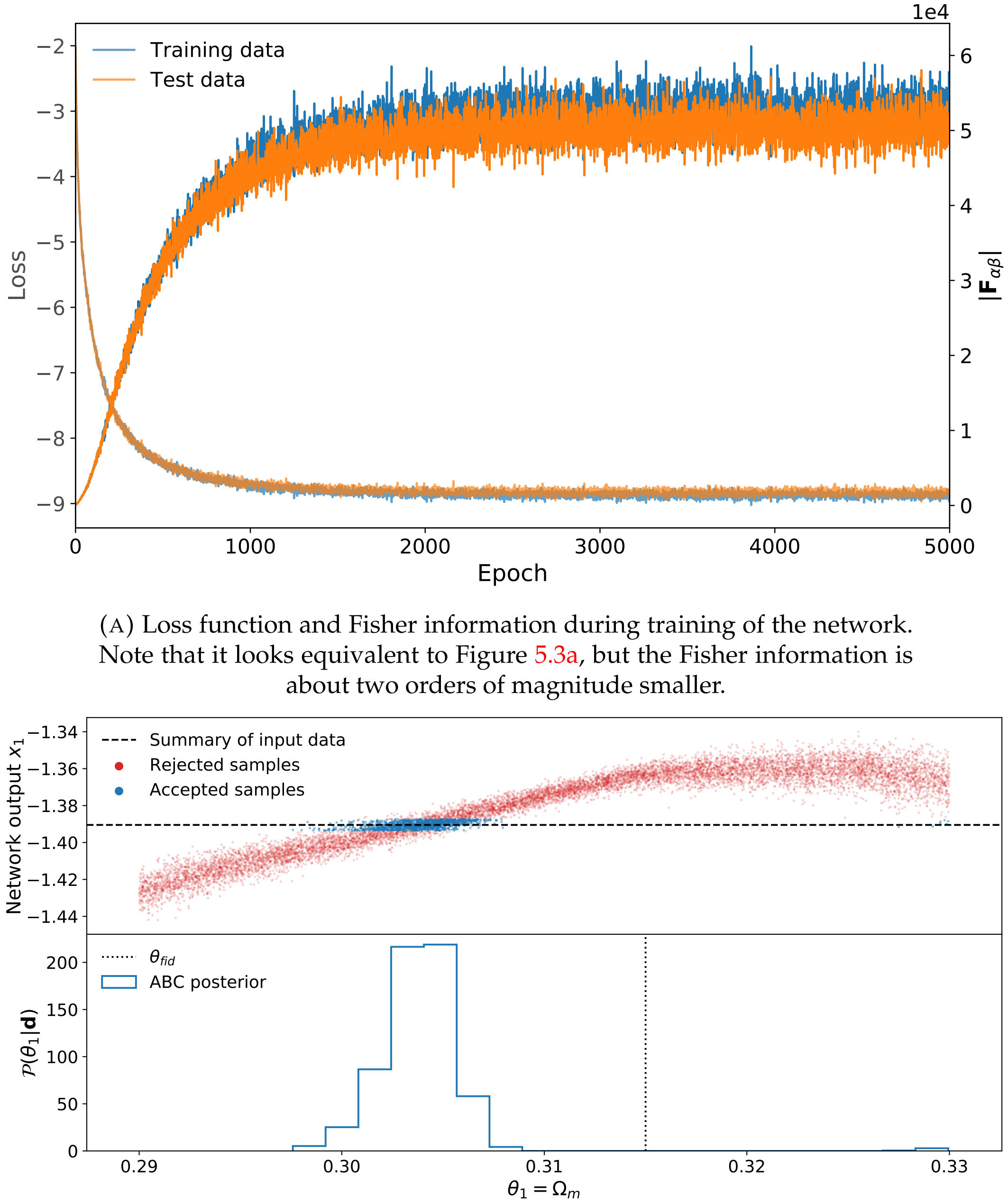


(A) Loss function (decreasing) and Fisher information (increasing) during training of the network.

(B) The top panel shows the network output summary as a function of the parameter at which a dataset was generated. The bottom panel shows the approximate posterior inferred from ABC. The dashed line shows the maximum likelihood estimate from interpolating the training summary and the dotted line shows the parameter at which the input data was generated.

FIGURE 5.3: Network results after re-scaling the input data by taking the negative logarithm. (A) shows the learning process and (B) shows the results of ABC.

the power spectra to be calculated in parallel for uniform priors. This turned out to be faster than doing Population Monte Carlo (PMC) sampling in one dimension, but it should be noted that this is simply an implementation problem and generally PMC will converge faster. We draw 20,000 simulations from the prior and set the threshold ϵ such that 5% of all simulations are accepted. The results of ABC are plotted in Figure 5.3b. Figure 5.3b shows good agreement of the inferred posterior and the input parameter $\Omega_m = 0.315$. The maximum likelihood estimate (MLE) in



(A) Loss function and Fisher information during training of the network. Note that it looks equivalent to Figure 5.3a, but the Fisher information is about two orders of magnitude smaller.

(B) Equivalent to Figure 5.3b, but for data standardized to zero mean and unit variance after the logarithm has been taken.

FIGURE 5.4: Network results after re-scaling the input data by taking the negative logarithm and subsequently mapping to zero mean and unit variance. (A) shows the learning process and (B) shows the results of ABC.

this figure is obtained from interpolating the output of the network on the training data to the output of network on the ‘real’ data using the derivative of the network at the trained simulations. Figure 5.3b shows that the MLE is in this case a good estimate of the true parameter, since the data is generated at the same parameter as the training data for the network.

In the field of machine learning, the data is often standardized to make the distribution of the data have zero mean and unit variance. In the ideal case, this should not impact the end results, but usually makes learning faster, as the network does not spend the first part of training performing some re-scaling to the scale of the features. To investigate whether this is also the case with weak lensing data vectors, we have trained another version of the network with data that was first standardized to logarithmic space according to Equation 5.5, and subsequently mapped to have zero mean and unit variance. The learning curve of this network is shown in Figure 5.4a. This Figure shows the same behaviour as Figure 5.3a but the Fisher information is two orders of magnitude less. The network does not converge significantly faster than the previous network, and from Figure 5.4a it seems that the network has maximized the Fisher information. However, when using this network as the compression function for ABC, we found that the summary statistic does not capture the right information about the parameter Ω_m. This becomes apparent from Figure 5.4b, which shows a large bias towards low Ω_m, and also a larger scatter in the output summary as function of Ω_m than in Figure 5.3b. This behaviour is likely due to the fact that the weak lensing data vector spans several orders of magnitude and re-scaling this vector to have zero mean and unit variance removes that information. Finally, standardizing the data without mapping it to logarithmic space first produced a behaviour of the loss function similar to that of Figure 5.2, where the network was unable to decrease its loss function. The optimal re-scaling of the data is thus taking the logarithm, this is the re-scaling that we use throughout the rest of this work.

5.2 Tomography and multiple parameters

In this section we generalize the approach of Section 5.1 to multiple tomographic redshift bins and multiple parameters. We choose to look at the two parameters that cosmic shear is most sensitive to, σ_8 and Ω_m, but we stress again that cosmic shear actually only probes a degenerate combination of these parameters (see Sect. 2.2.2). The distribution of source redshifts is now split into $n_z = 3$ tomographic redshift bins. This produces a total of $n_z \times (n_z + 1)/2 = 6$ power spectra, three cross-correlation and three auto-correlation spectra. The weak lensing power spectrum in redshift bin ij is calculated according to Equation 2.45:

$$C_{\ell,ij} = \int W_i(\chi) W_k(\chi) \chi^{-2} P_\delta(k = \frac{\ell}{\chi}; \chi) \tag{5.6}$$

The shape noise contamination now only applies to auto-correlation spectra, since the shapes of galaxies in different redshift bins are assumed to be uncorrelated. This is imposed by the Kronecker delta δ^K_{ij} in

$$C^{obs}_{(ij)}(\ell) = C_{(ij)}(\ell) + \delta^K_{ij} \frac{\sigma_e^2}{\overline{n_i}}, \tag{5.7}$$

here $\overline{n_i}$ refers to the average number of galaxies in redshift bin i. In the case of tomography, the Gaussian covariance between angular power spectra in different tomographic redshift bin combinations ij and $i'j'$ is given by (Takada and Bridle, 2007)

$$Cov_{\ell,(ij),(i'j')} = \frac{C^{obs}_{(ii')}(\ell) C^{obs}_{(jj')}(\ell) + C^{obs}_{(ij')}(\ell) C^{obs}_{(ji')}(\ell)}{(2\ell + 1)\Delta\ell f_{sky}}. \tag{5.8}$$

where f_{sky} is the sky coverage, and $(2\ell + 1)\Delta\ell f_{sky}$ is the number of modes at a given ℓ, which is adjusted for the partial sky coverage of Euclid. The general covariance

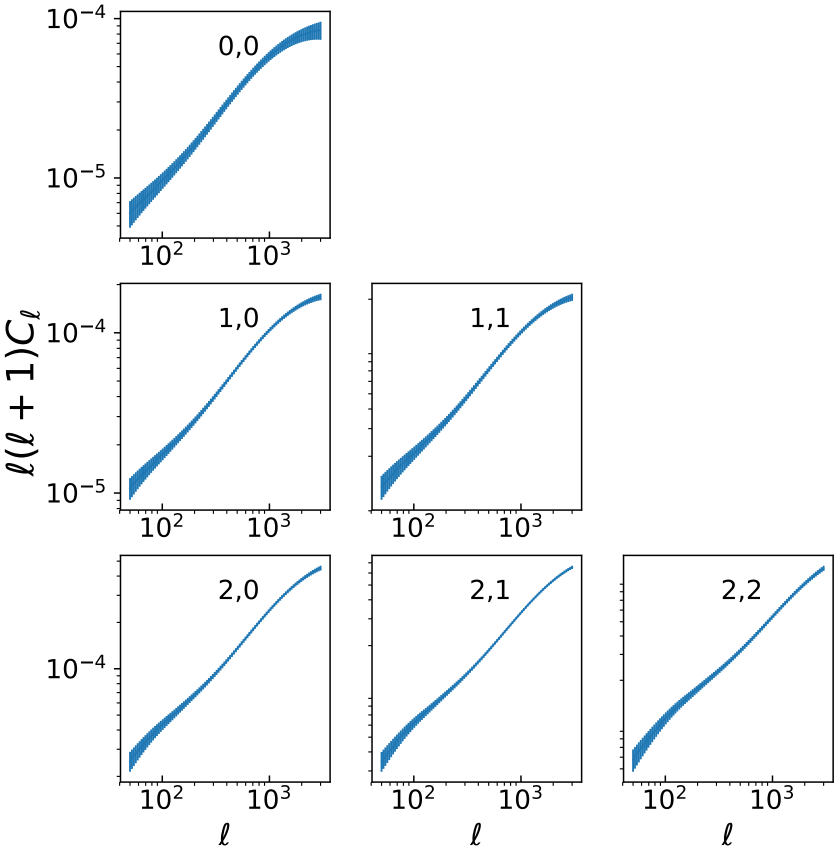
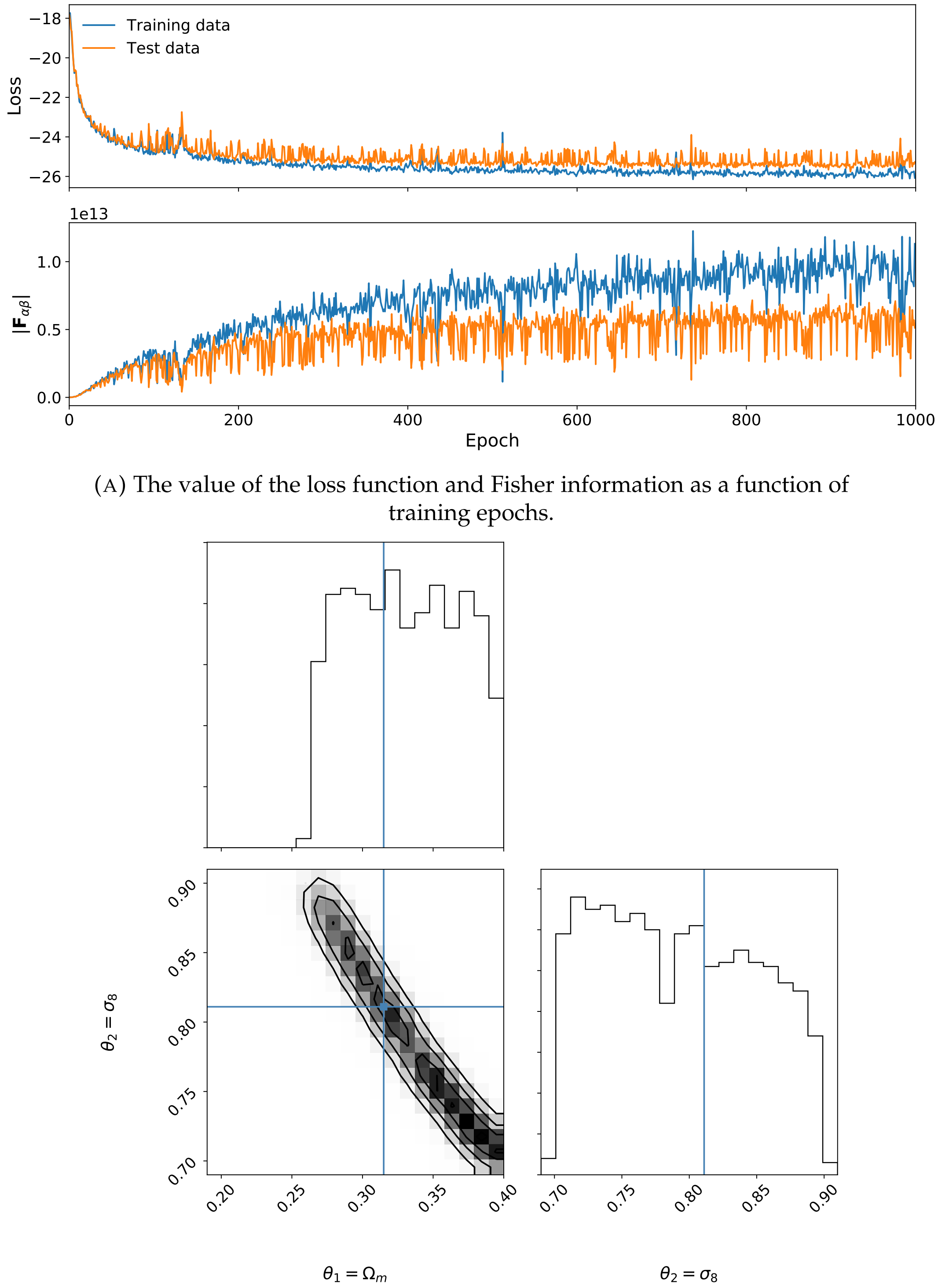


FIGURE 5.5: Theoretical power spectra calculated for a Euclid-like survey with three tomographic redshift bins. The error bars are given by the diagonal of the Gaussian covariance matrix.

matrix, including the non-Gaussian term, would have dimensions $n_l \times (n_z+1)/2$ by $n_l \times (n_z+1)/2$, with n_l being the number of sampled multipoles and n_z the number of redshift bins. This is due to the non-Gaussian correlations between different ℓ modes. The Gaussian part of the covariance matrix is diagonal in ℓ, which means there is correlation only between the same ℓs in different tomographic power spectra. This allows us to define n_l matrices of dimensions $(n_z+1)/2$ by $(n_z+1)/2$. Again, we set $n_l = 100$ multipole bins that are logarithmically spaced between $\ell = 50$ and $\ell = 3000$.

The fiducial cosmological parameters at which training data is generated are now defined as $\sigma_8 = 0.811$, $\Omega_m = 0.315$, $\Omega_b h^2 = 0.0244$, $h = 0.674$, $n_s = 0.965$, $w_0 = -1.03$, following Alsing and B. Wandelt (2019). The parameters of interest are now denoted by $\theta = [\Omega_m, \sigma_8]$, and we use $\Delta\theta = [0.02, 0.02]$ for the simulations just above and below the fiducial parameters. The six calculated power spectra with the diagonal part of the covariance matrix as the error bar are shown in Figure 5.5. This figure shows that because we split the distribution up into several redshift bins the shape noise now starts to become important in some of the power spectra at the high ℓ modes.

The network has the same setup as the previous section, but now the input data vector is the flattened vector of the six noisy power spectra shown in Figure 5.5. Thus, the input data vector is now a 600 dimensional vector, which will be compressed to two numbers. The learning curve of the information maximizing network is shown in Figure 5.6a. The network converged within a few hundred epochs, but



(A) The value of the loss function and Fisher information as a function of training epochs.

(B) Inferred two dimensional posterior after Approximate Bayesian Computation with a uniform prior over Ω_m and σ_8. The blue lines indicate the input parameters.

FIGURE 5.6: Results after training a network to compress weak lensing data vectors with two free parameters, Ω_m and σ_8 and three tomographic redshift bins. (A) shows the learning process and (B) shows the inferred posterior.

there is a small discrepancy between the training and the test loss. We have found

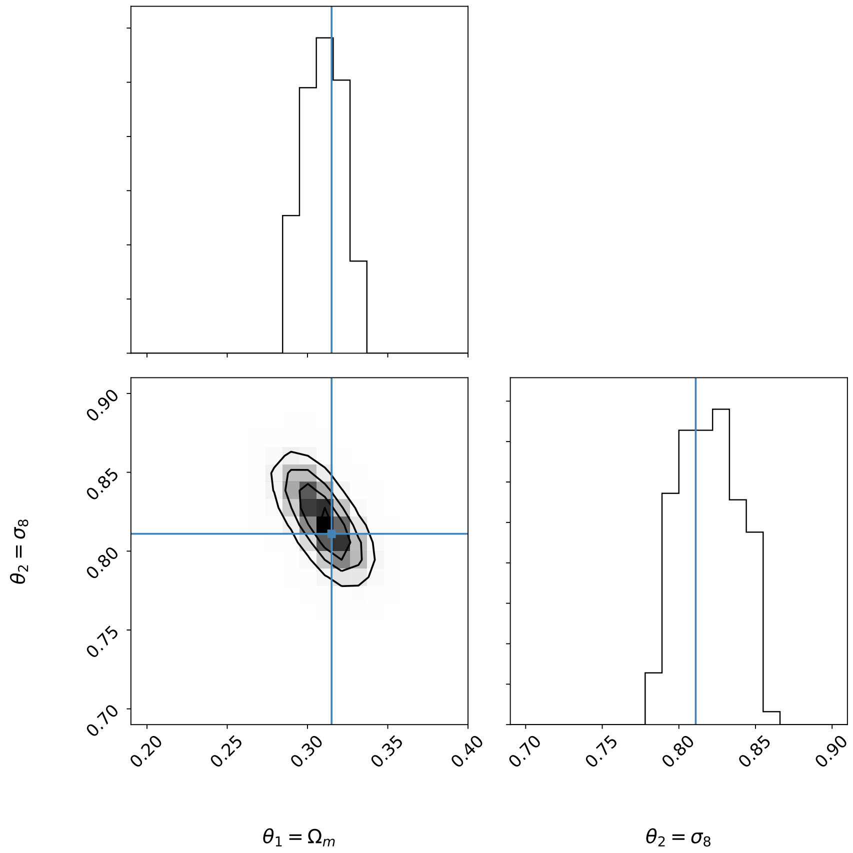
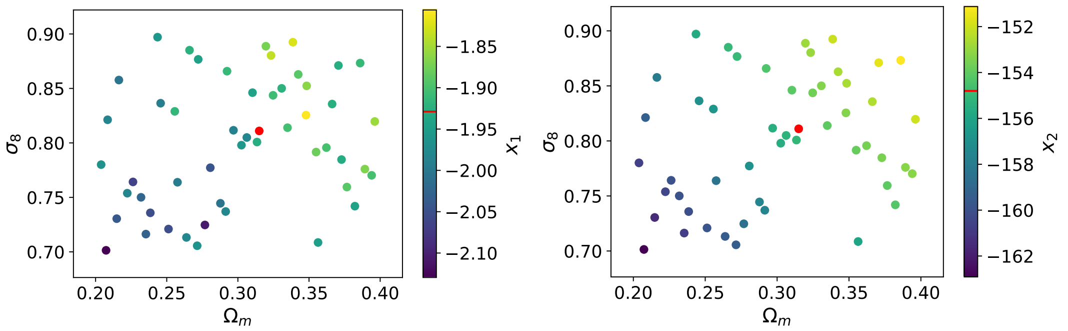


FIGURE 5.7: Equivalent to Figure 5.6b, but with a tighter prior on Ω_m.

that using architectures with more nodes or layers does not fix this small discrepancy. However, some discrepancy between the training and test set is to be expected for increasingly complex datasets since the network is trained to optimize the Fisher information for the training set, and will thus generally perform better on the training set. Since the test loss is not increasing, we can be reasonably sure that we are not overfitting.

To investigate the fidelity of the final trained network, we again perform likelihood free inference on some input data. The input data that is fed to the trained network is a noisy weak lensing data vector generated at $\Omega_m = 0.315$ and $\sigma_8 = 0.811$. For the ABC algorithm, we use a uniform prior over Ω_m in the range $[0.20, 0.40]$ and a uniform prior over σ_8 in the range $[0.70, 0.90]$. Again, 20,000 simulations are drawn from this prior and 5% of the simulations are accepted. The resulting 2D posterior is shown in Figure 5.6b. The degeneracy in Ω_m and σ_8 is visible as the posterior is elliptical, showing equiprobable contours going upwards in σ_8 with decreasing Ω_m. With a tighter prior on Ω_m we can lift some of the degeneracy, as an example, a uniform prior of Ω_m in the range $[0.29, 0.33]$ produces the posterior shown in Figure 5.7. We see that in all cases the posterior agrees well with the input parameters of Ω_m and σ_8, indicating the network has learned to summarize the weak lensing data vectors in an unbiased way, in the general case of multiple parameters and multiple tomographic redshift bins.



(A) First network output.

(B) Second network output.

FIGURE 5.8: Network output x_1 and x_2 as a function of the input parameters sampled at 50 points via a Latin hyper-cube design that maximizes the minimum distance between points in the parameter space. The fiducial parameters at which the network was trained and at which the input data is generated is indicated by the red dot.

5.3 Fitting the output summaries

The main computational bottleneck after training the network are the forward simulations of the weak lensing power spectra that we had to generate to compute the posterior via ABC. If an inexpensive model for this would be available, one could draw as much samples as is necessary to converge to the true posterior. From the behaviour of the output of the network as a function of the parameters (e.g., Figure 5.3b), it seems reasonable that we could fit a relatively simple function to approximate what the summary statistics the network would return given data generated from some input parameters. If this function is a good approximation, output summaries can be predicted immediately from the input parameters, allowing for computationally very cheap likelihood free inference. Building such a network emulator is the goal of this section.

5.3.1 Plane fit

We will begin with the first order approximation that the output summaries are a linear function of the input parameters, which essentially means fitting a plane to the network output. Before we can fit the output of the network as a function of the parameters, we must first sample the parameter space by simulating (ideally as few as possible) forward models and giving those to the trained network. Random sampling of the parameter space is sub-optimal, it is usually better to use a Latin hypercube design (McKay, Beckman, and Conover, 1979). Latin hyper-cubes partition the range of every parameter into M segments such that the entire N dimensional space is divided into M^N cells. A sample point is then drawn and accepted if no other sample points lie along the same partition in any dimension (i.e., samples are not in the same row or column). In this way, all defined sub-ranges of the possible parameter combinations are simulated. We impose the additional criterion that the minimum distance between the points has been maximized, the *maximin* criterion (Johnson, Moore, and Ylvisaker, 1990).

We generate 50 samples in total from the parameter space where $0.2 \leq \Omega_m \leq 0.4$ and $0.7 \leq \sigma_8 \leq 0.9$. The 50 samples are shown in Figures 5.8a and 5.8b which respectively show the first and second output of the network for the 50 input simulations. The figures show clearly that the two summaries are correlated. The correlation is

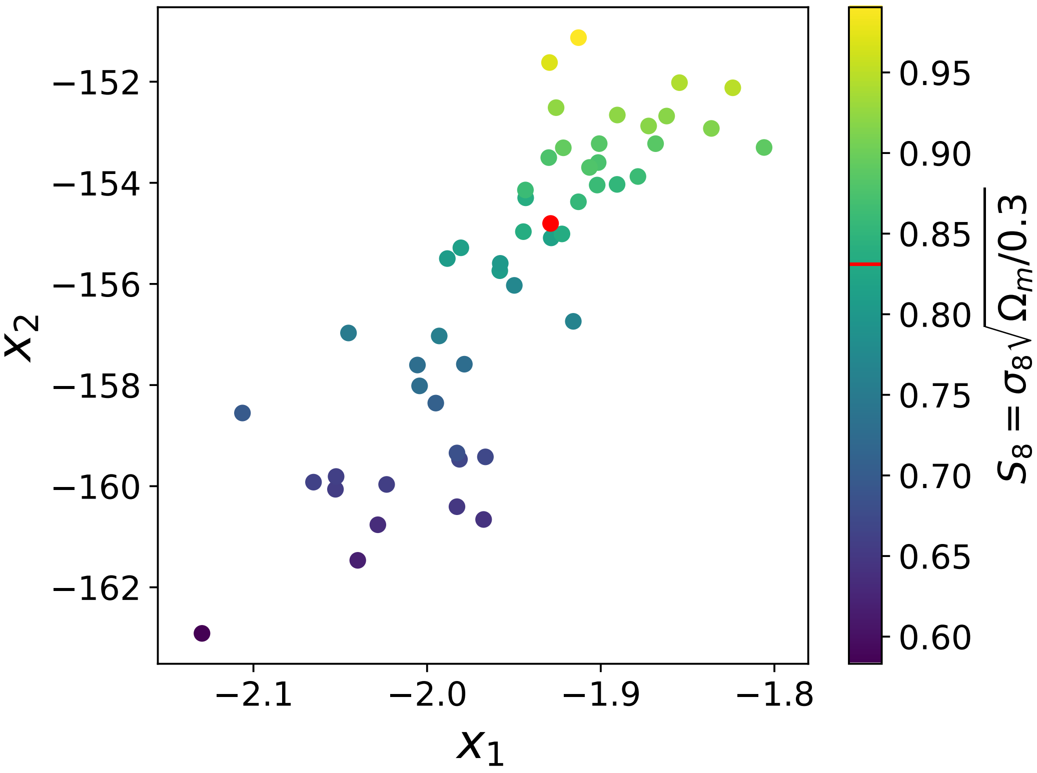
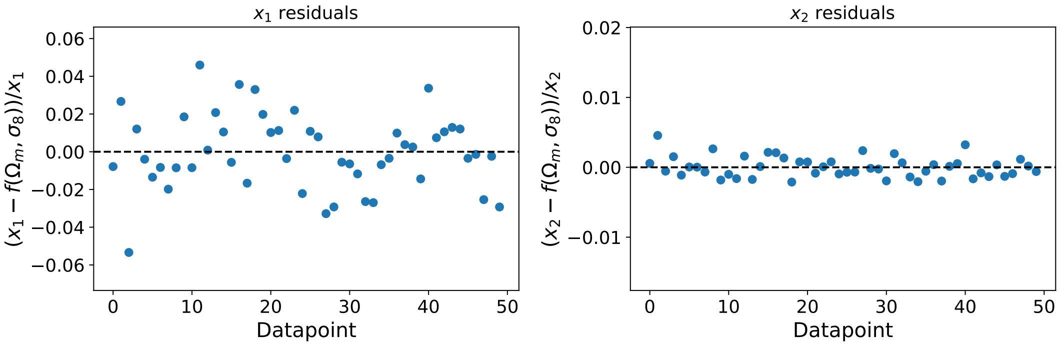


FIGURE 5.9: Scatter plot of both network output summaries colored by $S_8 = \sigma_8\sqrt{\Omega_m/0.3}$. The 50 points shown here are sampled in the $\Omega_m, \sigma 8$ plane via a Latin hyper-cube design. The fiducial parameters at which the network was trained and at which the input data is generated is indicated by the red dot.

even more evident in Figure 5.9, where we plot the output summaries colored by $S_8 = \sigma_8\sqrt{\Omega_m/0.3}$. The fact that the output summaries are correlated is not surprising, given that σ_8 and Ω_m are degenerate and the combination of both, S_8, is actually the only parameter that we can infer. Thus the minimum amount of sufficient statistics in this setup is actually only one, and it seems that the network has learned this as well. Therefore, even though the summaries are regularized against having correlation (Equation 3.11), one of the summaries is now partly redundant as both contain overlapping information. We still fit both summaries with a plane function, as we need two equations to infer both parameters.

$$\begin{aligned} x_1 &= a_1\Omega_m + b_1\sigma_8 + c_1 \\ x_2 &= a_2\Omega_m + b_2\sigma_8 + c_2. \end{aligned} \tag{5.9}$$

The fit is calculated with a robust least squares method, using the *soft L1* loss function $\rho(z) = 2\sqrt{(1+z)} - 1$. The soft L1 loss is the smooth approximation of the classic L1 loss function, we choose this loss function as it is more robust against outliers. The best fit parameters calculated with robust least squares differ less than a percent from the best fit parameters calculated with general least squares, which is expected as Figure 5.8 showed no clear outliers. The plane seems to be a reasonable first model, since the residuals shown in Figures 5.10a and 5.10b indicate a relatively good fit. The fit on x_2 better than on x_1, which is also to be expected since Figure 5.8 showed that x_2 has much lower scatter than x_1. The fit can now be viewed as the



(A) Residuals of the first output summary.

(B) Residuals of the second output summary.

FIGURE 5.10: Residuals of the plane fit $f(\Omega_m, \sigma_8)$, to the first and second output summary of the network. The residuals are plotted against the index of the 50 data points in the Ω_m, σ_8 plane.

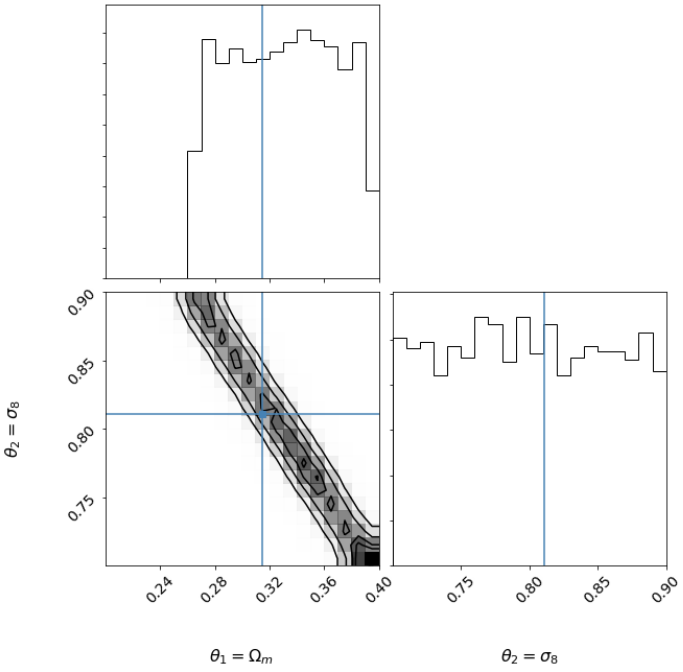
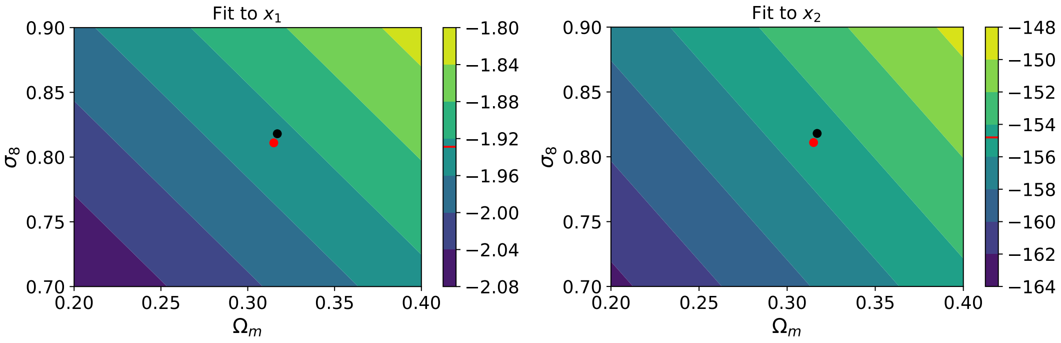


FIGURE 5.11: Posterior inferred by Approximate Bayesian Computation using the fitted plane model as the forward model. A slight bias towards higher S_8 is visible, as the posterior lies more to the upper right of the input parameters, given by the blue lines.

cheap model that calculates the 'forward simulations' and we use this model to do ABC.

We use the same uniform priors for ABC as in the previous section. Figure 5.11 shows the inferred posterior, which shows we have introduced a small bias towards higher Ω_m and σ_8. It also shows that the degeneracy in Ω_m and σ_8 is made slightly worse because the plane fit is only a linear approximation, which is only valid near the input parameters. By extrapolating this linear approximation, we impose a perfect degeneracy over Ω_m and σ_8 over the full range of possible values of Ω_m and



(A) Fit to the first output summary of the network x_1.

(B) Fit to the second output summary of the network x_2.

FIGURE 5.12: Plane fit to the output summaries of the network. The fiducial parameters at which the network is trained and the input data is generated are indicated by the red dot and the most likely value for the parameters of the input data given the plane fit are indicated by the black dot.

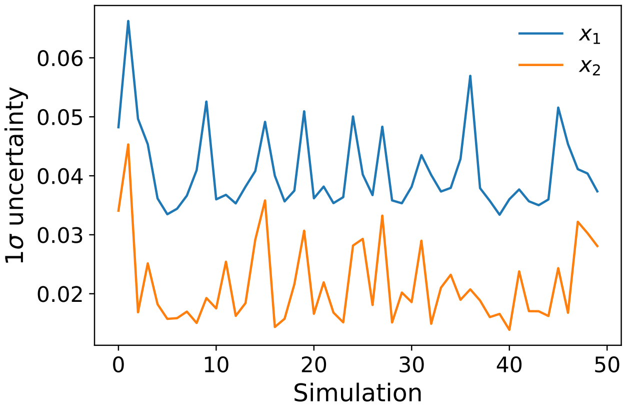


FIGURE 5.13: One sigma uncertainties on each of the 50 output summaries of the network, calculated by 1000 Monte Carlo realizations of the particles sampled by the Latin hyper-cube.

σ_8. The fits to x_1 and x_2 are plotted in Figures 5.12a and 5.12b. Here the fiducial input simulation is indicated by the red dot, and the most likely values, as will be explained later, are indicated by the black dot, which is indeed slightly biased upwards.

The most likely value of Ω_m and σ_8 of some input data can be inferred directly if we have the fitted model by setting Equation 5.9 equal to the output summaries of the input data. Solving these equations gives us the values $\Omega_m = 0.32$ and $\sigma_8 = 0.82$, which are quite close to the input values. To estimate the uncertainty on these parameters, we must estimate the uncertainty on the fit.

The uncertainty on the fitted parameters can be estimated if we do a simple least squares fit. We have already confirmed the difference with robust least squares to be small, so in this case outliers are not of concern. To quantify the uncertainty correctly, we must first have some idea about the uncertainty of the calculated summaries x_1 and x_2. For this, we use a Monte Carlo method. We draw 1000 noisy realizations of every of the 50 power spectra generated. This is still a cheap calculation, since we only have to add different realizations of Gaussian noise to the already calculated power spectra. These 1000 noisy realizations are propagated trough the trained network, and the standard deviation in the output of the network will be used as the uncertainty on the summaries. The 1σ uncertainty on each of the 50 data points is plotted in Figure 5.13, which shows the average 1σ uncertainty is 0.041 for x_1 and 0.022 for x_2, which are quite low given the values of the output summaries. Using this as the estimate for the error on the summaries, we find that the best fit least square parameters $p \in [a_1, b_1, c_1, a_2, b_2, b_3]$ have fractional uncertainties p/σ_p given by $[0.1, 0.2, 0.03, 0.0010.002, 0.0002]$ respectively, where we can immediately see again that x_1 does not follow a plane fit as well as x_2. For the most likely parameters we now propagate these uncertainties, taking into account the covariance of the fitted parameters. We find after solving Equation 5.9 with the simple least square parameters, the values of $\sigma_8 = 0.82 \pm 0.08$, $\Omega_m = 0.31 \pm 0.05$. This means that the input parameters of $\Omega_m = 0.315$ and $\sigma_8 = 0.811$ lie just outside of the 1σ boundary. This shows that the fitting the network with this simple function still retains much information about the parameters, although at the cost of a stronger degeneracy between Ω_m and σ_8 and a slight bias towards higher S_8 due to the linear approximation of the summary statistics.

5.3.2 Gaussian Process

The results of the previous section show that fitting the output summaries of the network can provide us with a promising network emulator which allows us to do very fast likelihood-free inference. The plane fit, however, is not in general a good option, and while the fit seemed to work relatively well given the limited complexity of the model, we can do better with more advanced methods.

In this section we fit the output of the network with a probabilistic model, a Gaussian process (see Rasmussen and Williams, 2005, for an in-depth description). The two main advantages of fitting a Gaussian process are that we do not make any assumption about the specific form of the process that we try to model and that there is a natural way to quantify the uncertainty in the data points inferred from the model.

Gaussian processes are recently becoming more popular in cosmology, as they can be used as accurate, but fast emulators, which will are becoming more and more important in the precision-era of cosmology. Gaussian processes have been used for example to emulate the halo mass function (T. McClintock et al., 2019), the matter power spectrum (Lawrence et al., 2017) and for reconstructing the Planck 2018 probability distributions with a massive speedup in computation time (Thomas McClintock and Rozo, 2019).

To define a Gaussian process, first we define a stochastic process where the points θ in parameter space are assigned random variables by some real function $f(\theta)$, in our case this function is the trained network. This stochastic process is called a Gaussian process if we assume the joint distribution of a subset of these random variables $\mathcal{P}(f(\theta_1), ..., f(\theta_N))$ is a multivariate Gaussian distribution. Gaussian processes can thus define a distribution over functions, allowing us to perform the fitting in function space, rather than in parameter space. In reality, we are not interested in modelling the actual input data itself, but only in inferring the relationship between the input data and target data.

While a multivariate Gaussian is completely determined by its mean and covariance, a Gaussian process is completely determined by its mean function $m(\vec{\theta})$ and covariance function $k(\vec{\theta},\vec{\theta}')$, defined as (Rasmussen and Williams, 2005)

$$\begin{aligned} m(\vec{\theta}) &= \mathbb{E}[f(\vec{\theta})] \\ k(\vec{\theta},\vec{\theta}') &= \mathbb{E}[(f(\vec{\theta}) - m(\vec{\theta}))(f(\vec{\theta}') - m(\vec{\theta}'))] \end{aligned} \tag{5.10}$$

A Gaussian process has no analytical representation of the probability density function, but is usually written as

$$f(\vec{\theta}) \sim \mathcal{GP}\left(m(\vec{\theta}), k(\vec{\theta},\vec{\theta}')\right). \tag{5.11}$$

where the mean is often set to zero, since this is simply a translation that can be added afterwards. The covariance function, or kernel k defines the smoothness of the Gaussian process. An often used kernel is the squared exponential kernel, which is considered a general good choice for a kernel function (Seikel and Clarkson, 2013). The squared exponential kernel is given by

$$k(\vec{\theta},\vec{\theta}') = \sigma_f^2 \exp\left(-\frac{(\theta-\theta')^2}{2l^2}\right). \tag{5.12}$$

where l is a hyper-parameter that defines the characteristic length scale of the problem and σ_f is a hyper-parameter that defines the signal variance. The squared exponential kernel imposes high covariance between outputs that are generated at close parameter $(\vec{\theta})$ values which falls off rapidly as the distance between the input parameters increases.

Formally, to determine the optimal hyper-parameters l and σ_f for our problem, we have to marginalize the marginal[1] Gaussian likelihood function over the hyper-parameters. This marginal likelihood function, neglecting terms not sensitive to hyper-parameters, is given by (Seikel and Clarkson, 2013)

$$\begin{aligned} \ln \mathcal{L} &= \ln p(f(\vec{\theta})|\vec{\theta},\sigma_f,l) \\ &= -\frac{1}{2}\ln|k(\vec{\theta},\vec{\theta}) + C| \end{aligned} \tag{5.13}$$

where k is given by equation 5.12, and thus depends on the hyper-parameters. However, marginalizing over the log-likelihood is expensive. If the likelihood is well-behaved and we take a reasonable starting point for the hyper-parameters, we can find the best hyper-parameters with a simple optimization routine that maximizes the marginal likelihood (e.g., Seikel and Clarkson, 2013).

Before we model the output summaries of the network as a Gaussian process, it is useful to re-map the summaries to a new space. As Section 5.3.1 showed, the summaries we sampled by a Latin hyper-cube are highly correlated, which is to be expected since we are probing the degenerate combination $S_8 = \sigma_8\sqrt{\Omega_m/0.3}$. Thus a single summary should suffice to carry all information about S_8. We re-map the output summaries by finding the unit eigenvectors $\vec{v}_1$ and $\vec{v}_2$ of the covariance matrix of the summaries. The eigenvectors define the mapping that transforms the covariance matrix to a diagonal matrix (with the corresponding eigenvalues on the diagonals), thus resulting in a space where the two summaries are uncorrelated. In this space, a single summary statistic carries all information about the parameters, and the other statistic is uninformative about the parameters. Applying this transformation to the

[1] The marginal Gaussian likelihood function here refers to the likelihood that is marginalized over the function space, not the hyper-parameter space.

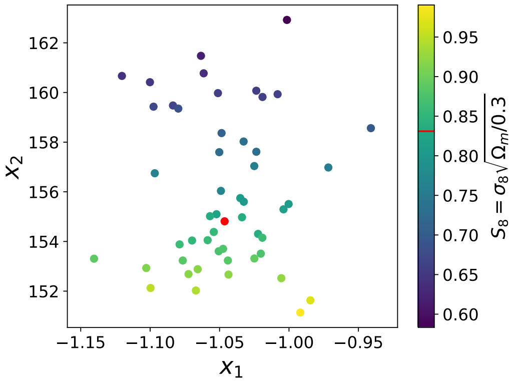


FIGURE 5.14: Equivalent to Figure 5.9, but after transforming to the space where the two summaries are uncorrelated.

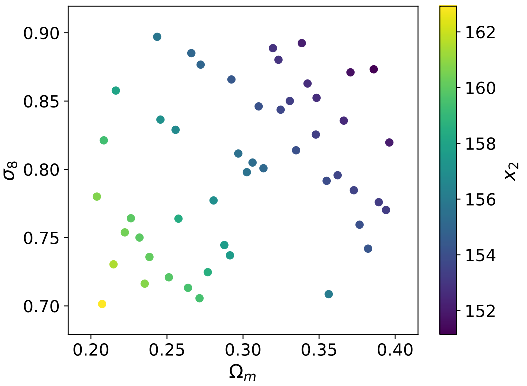


FIGURE 5.15: Re-mapped second output summary as a function of Ω_m and σ_8.

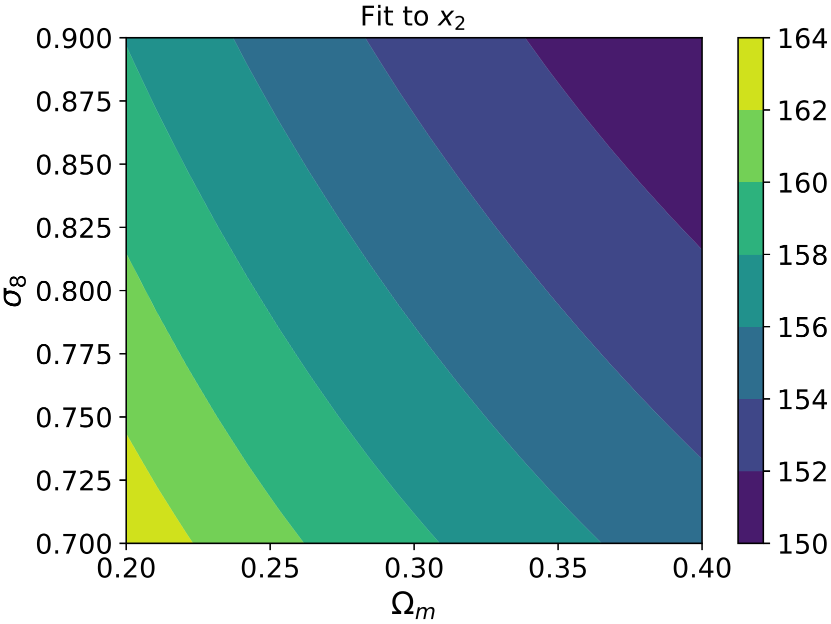


FIGURE 5.16: Gaussian Process fit to the re-mapped second output summary.

summaries previously shown in Figure 5.9 yields the summaries as shown in Figure 5.14. Figure 5.14 indeed shows that all information about S_8 is now contained in the second summary, as the second summary is now clearly correlated with S_8 while the first is uncorrelated.

To fit the Gaussian process to the re-mapped second output summary $\hat{x}_2$, we choose the value for the initial hyper-parameter σ_f to be equal to the sample variance of the 50 re-mapped output summaries $\sigma_{\hat{x}_2} = 9.4$. For the characteristic length scale l we take a large initial guess of $l = 5$, since the fitted function should be rather smooth, as the results of the plane fit were already decent, and the output summary seems to be a smooth function of both parameters, as is shown in Figure 5.15. This initial guess was found to be a good starting point after trying multiple scale lengths, lower scale lengths were found to over-fit the summary. We maximize the likelihood (Eq. 5.13) with a quasi-Newton optimizer and find that the optimal parameters are

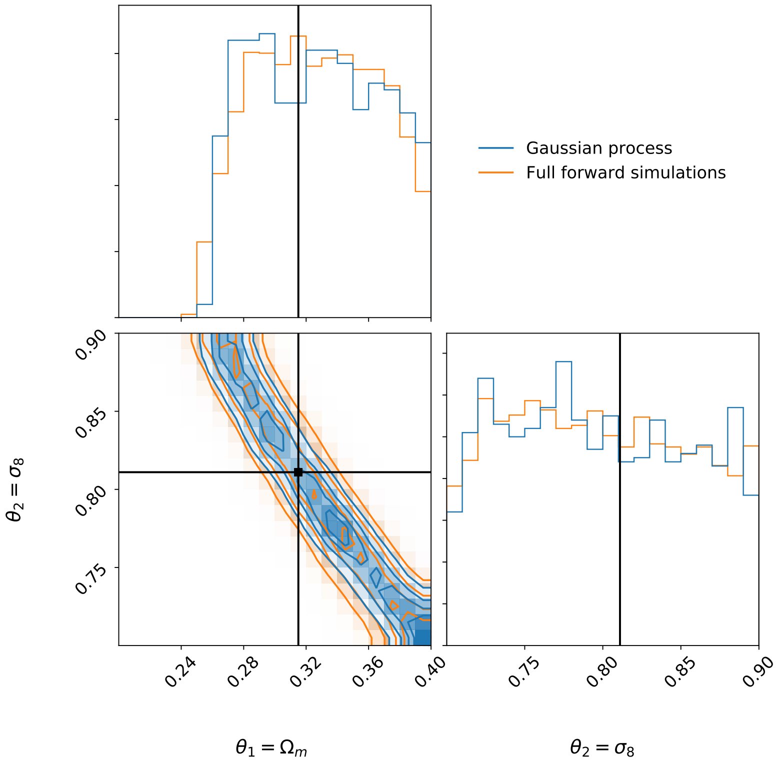


FIGURE 5.17: Posterior inferred from Approximate Bayesian Computation using the fitted Gaussian Process as an emulator for the forward simulations and the network. The posterior previously inferred from using 20,000 full forward simulations and propagating these through the network is shown in orange and the input parameters are indicated in black.

$\sigma_f = 12.1$ and $l = 1.9$. The resulting Gaussian process looks like a good fit, which can be appreciated from comparing Figures 5.15 and 5.16.

The Gaussian Process now defines our network emulator and we can use it to do Approximate Bayesian Computation. The setup of the ABC is the same as before, except that the Gaussian process now generates the summary statistics directly from the parameters. The resulting posterior is shown in Figure 5.17. Figure 5.17 shows that the posterior generated with the Gaussian process agrees very well with the posterior that was calculated with a full set of 20,000 forward simulations in Section 5.2. This shows that we can infer an accurate unbiased posterior by fitting just 50 simulations generated by a Latin hyper-cube design, which provides a massive reduction in computation time when compared to the 20,000 full forward simulations we had computed before.

Chapter 6

Discussion

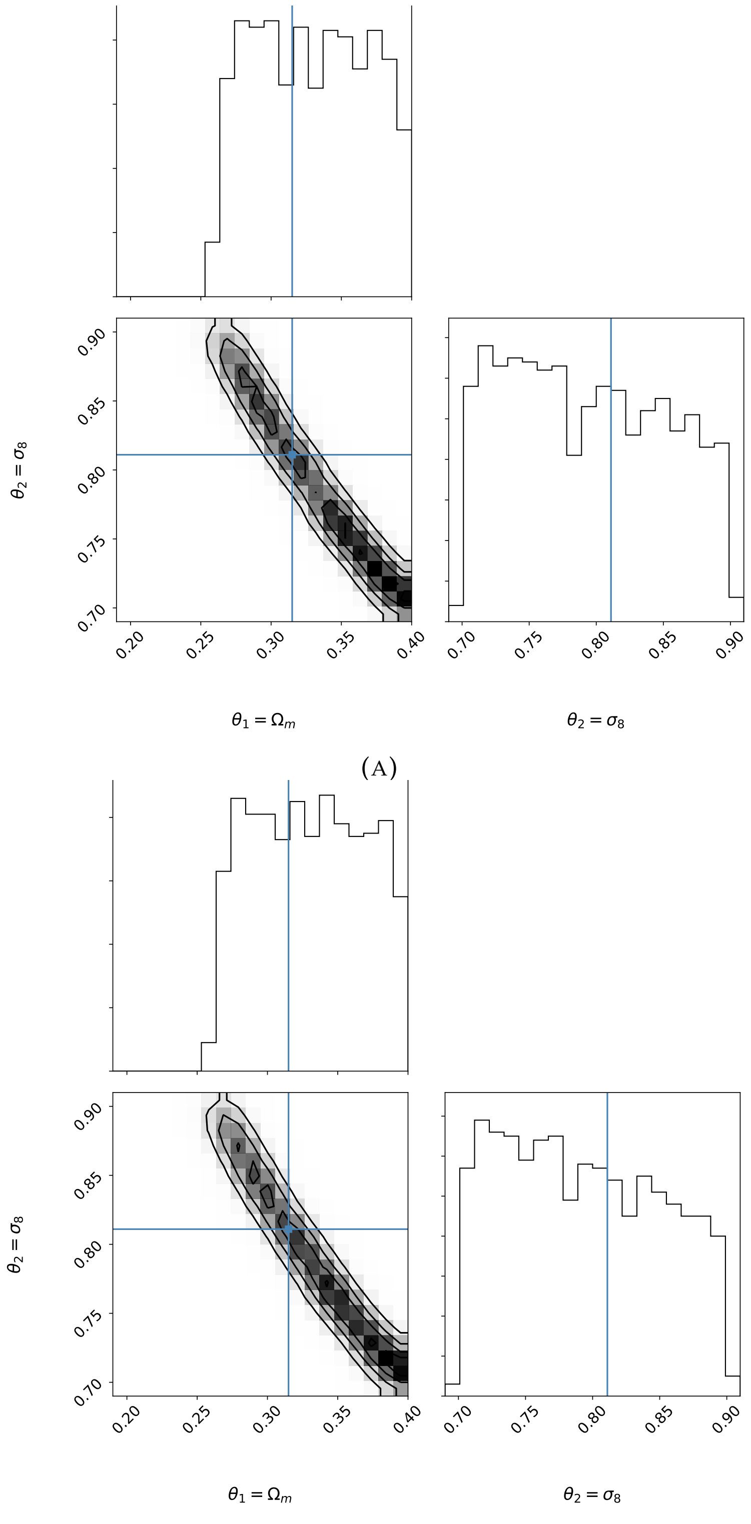
6.1 Robustness of the network

To discuss the results given in the previous chapter properly, we should first review the robustness of the information maximizing neural network. This is the aim of this section, where a few additional tests of the robustness are also performed.

Random processes It is important to realize that we are dealing with random processes, as the initialization of the information maximizing neural network is a random process, and the inference of the posterior through approximate Bayesian computation (ABC) is a random process. Ideally, these should not effect the final results, given that the network is trained to convergence and that enough samples are drawn for ABC. For the ABC we used 20,000 simulations with relatively small priors of width 0.2 in Ω_m and σ_8 space. This is an approximate step-size in parameter space of 0.001. The number of simulations is thus not of concern, as we actually over-sampled the parameter space somewhat heavily. During the inference of Gaussian signals, Table 4.1 showed that the final Fisher information converged to around the same value for a broad range of hyper-parameters and thus also various different initializations of the network. Additionally, since we did not have to re-run the network multiple times in Chapter 5, we can conclude that for the setups used throughout this thesis the output of the network is at least robust against the precise seeding of the random number generators that are used.

Training parameters An important thing to note is that thus far in this work, the values of the parameters at which the information maximizing neural network was trained were kept the same when generating some 'observed' data for which we inferred the posterior to check the fidelity of the output summaries. While we can use relatively tight priors on Ω_m and σ_8 obtained from previous experiments such as the Planck Collaboration, Aghanim, et al. (2018), it is unrealistic that the network will be trained at exactly the parameters of the Universe.

We investigate whether the network summaries are still informative about the input data when the network is trained at some fiducial parameters which are chosen away from the parameters with which the input data is generated. For this purpose, a new network is trained at the parameters $\Omega_m = 0.20$ and $\sigma_8 = 0.70$. After training, we propagate a power spectrum generated from $\Omega_m = 0.315$ and $\sigma_8 = 0.811$ through the network. Both output summaries are used in a full forward-simulation approximate Bayesian computation, which produces the posterior shown in Figure 6.1a. The figure shows clearly that the network is robust against being trained at parameter values that are away from the parameters at which the input data is generated. No bias is apparent, and the posterior is equivalent to the posterior generated with a network that was trained at $\Omega_m = 0.315$ and $\sigma_8 = 0.811$ (Figure 5.6b). This proves that the actual parameters at which the network is trained are not of much concern when inferring Ω_m and σ_8.



(A)

(B)

FIGURE 6.1: (A): Posterior generated by ABC of a network trained at $\Omega_m = 0.20$ and $\sigma_8 = 0.70$, while the input parameters of the input data are 0.315 and 0.811, which are indicated by the blue lines. (B): Equivalent, but with a network trained for only 10 epochs.

Network convergence In this work the neural networks were trained for many epochs after the loss function had flattened, to be sure that the loss function had indeed converged. The training was computationally quite cheap, because of the fact that simple fully-connected architectures with only a few hidden layers were used. If more effects, such as for example intrinsic alignment or photo-z biases, are included in the data a more complex network might be needed to maximize the information content. In fact, if the input data is much more complex, such as for example a highly dimensional shear map, some form of an (often more computationally intensive) convolutional neural network would likely be necessary to capture all available information. In this case, running the network for thousands of epochs beyond the initial flattening of the loss function might be prohibitively expensive.

To test the behaviour of a network that has not converged yet, we train the network for only 10 epochs, and use this network as the compression function to do ABC in the exact same way that was done before. The resulting posterior is shown in Figure 6.1b, which shows, surprisingly, that the posterior is almost exactly the same as the posterior generated with a fully trained network. We have found that the magnitude of the output summaries differs between the trained and untrained network, but the shape of the output summaries as a function of the parameters does not. Therefore, we see two possible explanations for this result. First, it could be that the network has not actually maximized the information content yet, but simply lowered the covariance between the initial summaries such that the regularization term in the loss function (3.11) becomes lower, and the Fisher information increases up to convergence. This converged Fisher information might not be the total amount of information available in the data however, but simply the total amount that the network can extract from the data given the architecture of the network. Second, it could be that the random initialization of the weights and biases of the network already creates a complex enough non-linear function that it provides a one-to-one mapping of the data to a summary statistic. The parameters Ω_m and σ_8 only introduce a scaling of the amplitude of the power spectrum, so a simple function such as the mean of the power spectrum would already be sufficient to distinguish power spectra generated at different parameters.

We have tried some more complex densely connected architectures by changing the number of hidden layers between 2 to 4 and tweaking the number of nodes, but we have not found significantly different performance by the network. In all cases the Fisher information was of comparable magnitude. This implies that a simple extension of the architecture is not enough to extract more information from the input data, but a more complex architecture might be able to.

6.2 Interpretation of the results

One parameter Following the layout of the previous chapter, we start by considering the results of the inference of Ω_m with as input data noisy power spectra generated without tomographic redshift binning. Figure 5.3b showed that the output summary is a strong linear function of Ω_m. A simple linear function is not unexpected because of the simple scaling of the power spectrum at all scales with changing Ω_m. Since σ_8 was not a free parameter in the initial setup, there is no degeneracy and the posterior on Ω_m was quite narrow. From the agreement of the posterior with the input parameter of $\Omega_m = 0.315$, we can conclude that the network used for the inference of Ω_m has successfully summarized all information available in the noisy power spectra.

From Figure 5.4a, which showed the Fisher information of the network on data mapped to zero mean and unit variance, and the discussion in the previous section, we can deduce that convergence of the Fisher information or loss function of the network does not imply that all the information in the data is captured. While Figure 5.4a showed that the network converges to a Fisher information of around 5×10^4, the output summaries are now very biased towards low Ω_m as Figure 5.4b showed. There is nothing that implies that something has gone wrong in the process of training the network, and the only way to know that the network was biased was in this case because we knew the value of the input parameter. This poses a problem for the application of the network to real data, since one might think from the convergence of the loss function that the network has captured all information, while a strong unknown bias is introduced and the wrong parameter value is inferred. It is therefore important in an analysis of real data to train various different setups and compare the Fisher information and posteriors of these setups.

Two parameters The results of Section 5.2, where we train the network to infer both Ω_m and σ_8 are mostly in line with the results of the one parameter case. The posteriors generated agree well with the input values of Ω_m and σ_8, but a strong degeneracy between these parameters is apparent. While this degeneracy is expected from theory, it should be partly broken by splitting the distribution up into three redshift bins and we would thus expect tighter constrains on the individual parameters as well (see e.g., Merkel and Schäfer, 2017, for Euclid forecasts). However, as the one dimensional posteriors from the histograms in Figure 5.6b show, the individual parameters are basically unconstrained as both parameters have almost uniform posteriors. To investigate this problem, we train three networks with data vectors generated without redshift bins, with three tomographic redshift bins and with six tomographic redshift bins respectively. We would expect the constraints on the parameters to improve as the number of redshift bins increases. Contrary to this expectation, Figure 6.2 shows that the constraints do not depend on the number of redshift bins used. This confirms that the network is not able to identify and learn the information that is added by adding more redshift bins. We suspect that this is due to the general simple feed forward architecture of the network. By this are not referring to the number of hidden layers or nodes, but the general setup that the first hidden layer takes the input power spectra as one large flattened data vector of dimension 600 in the case of three tomographic redshift bins. By concatenating different power spectra, we make it more difficult for the network to distinguish how many separate power spectra we are inputting. A possible way to solve this would be to train six different networks to infer the cosmological parameters for each power spectrum and then combine those posteriors to get the total constraints on Ω_m and σ_8. For the combination of these posteriors, the covariance of the power spectra should be carefully taken into account. Perhaps a better method is to define multiple separate input layers that are joined before the network output is given. In this way, it is clear that six distinct data-vectors are given, but the information is still combined inside the network.

Fitting summaries Section 5.3.1 showed that a simple model to fit the output summaries of the network was already capable of providing a promising emulator. This model could be fit to only 50 simulations sampled from a Latin hyper-cube design. However, it is likely that the simple model only worked because of the problems that were discussed in Section 6.1 regarding the convergence of the network and the

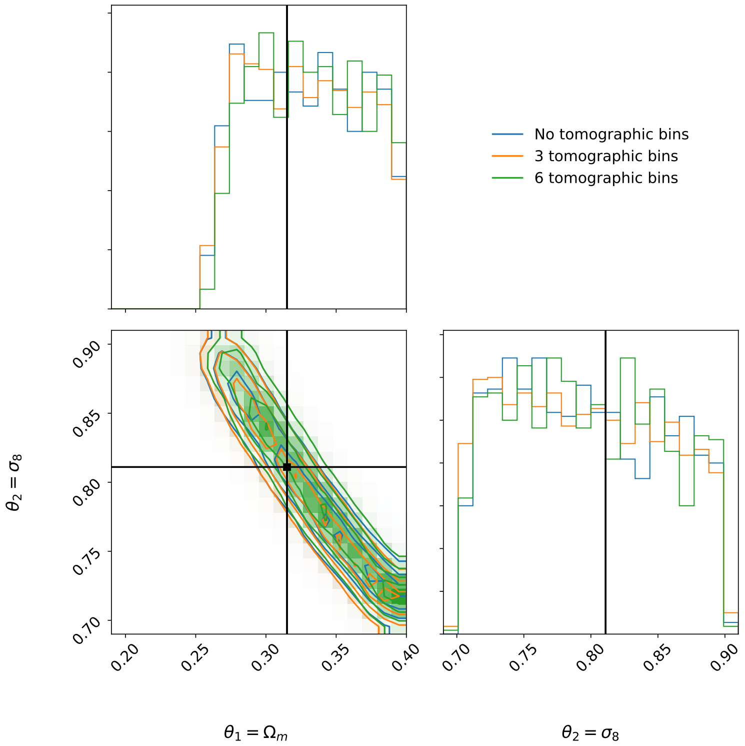


FIGURE 6.2: One and two dimensional posteriors inferred from the summary statistics of networks trained on weak lensing data vectors with one, three and six tomographic redshift bins.

inability to retrieve all available information from the data. The Gaussian process which was fit in Section 5.3.2 is a very versatile model without initial assumptions about the form of the model and is therefore likely still a good method to fit the output summaries in the case that all the information is extracted from the data by a more complex network. Figure 5.17 showed that the posterior generated from the Gaussian process agrees very well with the posterior generated from 20,000 full forward simulations. This provides a tremendous speedup in the case when the forward simulations are computationally very expensive. Further research has to determine whether the Gaussian process will be versatile enough to fit the probably more complex function that defines the output summaries given the parameters.

6.3 Comparison with other studies

In this section we will address how the information maximizing neural network compares to other data compression techniques used in cosmology. The three main data compression schemes that aim to compress data down to the number of parameters (n) are *approximate score compression*, *regression neural networks* and the information maximizing neural network (Alsing and B. Wandelt, 2019).

Approximate score compression (Alsing and B. Wandelt, 2018) finds the summaries by calculating the score function, which is the derivative of the log-likelihood with respect to the parameters, and thus a vector of length n. Alsing and B. Wandelt (2018) showed that the score function defines the compressed summary statistics. The main advantage of this method is that the compression is analytical, since the score function has an analytical expression, and that we thus know the mapping between data and summary statistics. The disadvantages are that we must assume the likelihood function to calculate the score function and that the score function is usually a function of some statistical properties of the data, such as the mean and covariance. While usually the Gaussian likelihood approximation may be justified under the central limit theorem, non-Gaussian terms may significantly affect current and future weak lensing surveys (Sellentin and Heavens, 2018). Additionally, calculating the statistical properties of the data may require many forward simulations of the data model, defeating the purpose of the data compression step.

Regression neural networks are deep neural networks that are trained to estimate the parameters from the data directly. Commonly in cosmology, convolutional neural networks (CNNs) are trained on shear/convergence maps or matter distribution maps (Schmelzle et al., 2017; Ravanbakhsh et al., 2016; Ribli, Pataki, and Csabai, 2019; Fluri, Kacprzak, Lucchi, et al., 2019). The output of the network can be seen as a direct prediction, or as a massively compressed summary statistic, which can then be used in a likelihood or likelihood-free inference setting. These neural networks do typically require the simulations to span a broad range of the parameter space, in comparison to the information maximizing neural network that is trained only at some fiducial point in parameter space. This is still an emerging area of research, as only recently have CNNs be applied to observed weak lensing data for the first time (Fluri, Kacprzak, Lucchi, et al., 2019). The advantage of this method is that little assumptions are made about the analysis process, which gives the neural network much freedom in building a model to extract the information from the data optimally. The disadvantage of this method is that the entire data analysis process becomes a black box and no guarantee is given that the output of the network is robust against different input datasets.

The information maximizing neural network combines both techniques, to find summaries without assuming the likelihood by fitting the compression function only. The advantages are a combination of the advantages of the previous techniques. First, we do not require the entire parameter space to be sampled, only some fiducial point in the parameter space. Second, we do not have to make assumptions about the likelihood function that reduce the optimality of the compression. Third, the entire data analysis process is not a black box, but only the compression function is. Still, the compression function can be visualized by plotting the output of the network as a function of different input parameters. The main disadvantage is, as we have noted in the previous sections, that it is difficult to determine whether the information has been optimally compressed or simply compressed to the limit of the given network architecture, because the compression part of the process is still opaque.

We think it is likely that in the future the data compressing network used throughout this work will not replace the classic power spectrum analysis, but rather augment the analysis by finding additional summaries that the power spectrum might not capture. If the Gaussian likelihood function is found to be a good assumption for weak lensing data, or a correction is found for this assumption, we would recommend approximate score compression as a first-step compression summary, since the method is analytical. The information maximizing neural network can then be used to augment this summary, by extracting the extra information that is missed by the first-step compression.

6.4 Shortcomings and suggestions

This study presents the first application of the information maximizing neural network to weak lensing data. Therefore, many simplifying assumptions have been made throughout this work. Firstly, the weak lensing data vector that are generated are generated from calculated noisy realizations of angular power spectra, rather than noisy shear or convergence maps. This means a first-step compression is already implicitly made, and some information may already be lost. It would be interesting to see how the information maximizing neural network performs on simulated shear maps, and whether additional information can be extracted to augment the power spectrum summaries. Secondly, we have not included many effects present in real weak lensing data, and it is important to see how the network reacts to these effects. The things we have not modeled yet include, but are not limited to, the non-Gaussian covariance terms between different angular power spectra (Takada and Jain, 2009), a biased photometric redshift uncertainty model, catastrophic redshift outliers, survey masking or boundary effects, and the intrinsic alignment of galaxies.

Additionally, the approximate Bayesian computation method that we have used to do likelihood-free inference has some drawbacks. We have used the most naive approach, where many simulations are generated and a quite arbitrary distance threshold is set as the acceptance criterion. Even though we have also considered the Population Monte Carlo extension of this method, the stopping criterion for this method still has to be set quite arbitrarily. We have throughout the previous chapter set the threshold such that a small subset of the total amount of simulations were accepted, which is not a bad first guess. However, there are some better methods, such as setting the threshold such that only the region of parameter space is accepted where the output summary is approximately a linear function of the parameters, or by first performing simulations to calibrate the threshold value (see Bertorelle, Benazzo, and Mona, 2010, for a review). Implementing such a technique would make the posteriors more robust against different datasets and trained networks.

Lastly, it is clear that a better network architecture is needed. As concluded in section 6.2, the network is not sensitive to different numbers of tomographic bins, while increasing the number of bins should improve the constraints. To better investigate what information the network is missing, it would also be necessary to compare the posteriors with posteriors from a standard likelihood analysis of the power spectra, analogous to the analysis in Hildebrandt et al. (2017) for example.

Chapter 7

Conclusion

This work has presented the first application of the information maximizing neural network (IMNN) to weak lensing data vectors. We have implemented the IMNN and verified its functionality by first summarizing Gaussian signals, of which the resulting summaries and posteriors are known analytically. The Gaussian signals have shown that the network can reduce the dimensionality of input data to the a number of summaries equal to the number of free parameters. These summaries were still informative about the data, as the posteriors inferred by likelihood-free inference matched the analytical posteriors very well.

Mock weak lensing power spectra for a Euclid-like survey were generated with some simplifying assumptions and the IMNN was trained to summarize these. First, the only free parameter was set to be Ω_m. We found that data standardization to logarithmic space was needed for the network to extract the most informative summaries from the data. With this standardization, the summaries generated by the network were found to construct a convincing posterior for Ω_m in a likelihood-free inference setting. The approach was generalized to multiple parameters and multiple tomographic redshift bins. The network was found to create unbiased summaries informative about Ω_m and σ_8, but did not seem to extract the information that was added by splitting the source galaxies into tomographic redshift bins. We suggest that a more complex network architecture is needed to extract this information.

The computational bottleneck after training the network was found to be the large amount of forward simulations needed for the likelihood-free inference algorithm. As is becoming more and more common in cosmology, we attempted to avoid this bottleneck by fitting an emulator that returns the expected value of the summary statistics given the cosmological parameters. The emulator was fitted on a small number of network outputs, optimally sampling the parameter space via a Latin hyper-cube design. First, it was shown that a linear approximation of the summaries works relatively well, but may introduce a small bias in the final posterior. We then fitted a Gaussian process, which makes no assumptions about the functional form of the emulator. By transforming the summaries to the space where they are uncorrelated, we could fit a single Gaussian process. The posterior inferred by using this emulator agreed very well with the posterior obtained from 20,000 full forward simulations. This showed that the Gaussian process is a promising accurate and unbiased emulator for the summary statistics.

The results have been found to be robust as a function of different hyperparameters, but it is clear that there are still some problems with the current implementation. We have found that convergence of the loss function does not imply that all information has been extracted from the data. Future studies should aim to find a more complex network architecture that can extract the information from tomographic power spectra. We think it is likely that due to the opaque nature of the neural network, the IMNN will not replace classic power spectra analyses, but rather augment them by extracting the information classic analysis methods might miss.

Bibliography

Addison, G. E. et al. (2018). “Elucidating ΛCDM: Impact of Baryon Acoustic Oscillation Measurements on the Hubble Constant Discrepancy”. In: *Astrophysical Journal* 853.2, 119, p. 119.

Akeret, J. et al. (2017). “Radio frequency interference mitigation using deep convolutional neural networks”. In: *Astronomy and Computing* 18, pp. 35–39.

Alhassan, Wathela, A. R. Taylor, and Mattia Vaccari (2018). “The FIRST Classifier: compact and extended radio galaxy classification using deep Convolutional Neural Networks”. In: *Monthly Notices of the RAS* 480, pp. 2085–2093.

Alsing, Justin and Benjamin Wandelt (2018). “Generalized massive optimal data compression”. In: *Monthly Notices of the RAS* 476, pp. L60–L64.

— (2019). *Nuisance hardened data compression for fast likelihood-free inference*.

Alsing, Justin, Benjamin Wandelt, and Stephen Feeney (2018). “Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology”. In: *Monthly Notices of the RAS* 477.3, pp. 2874–2885.

Amendola, Luca, Stephen Appleby, Anastasios Avgoustidis, et al. (2018). “Cosmology and fundamental physics with the Euclid satellite”. In: *Living Reviews in Relativity* 21, 2, p. 2.

Amendola, Luca, Stephen Appleby, David Bacon, et al. (2013). “Cosmology and Fundamental Physics with the Euclid Satellite”. In: *Living Reviews in Relativity* 16.1, 6, p. 6.

Armitage, Thomas J., Scott T. Kay, and David J. Barnes (2019). “An application of machine learning techniques to galaxy cluster mass estimation using the MACSIS simulations”. In: *Monthly Notices of the RAS* 484, pp. 1526–1537.

Bartelmann, M. and P. Schneider (2001). “Weak gravitational lensing”. In: *Physics Reports* 340, pp. 291–472.

Beaumont, Mark A. et al. (2008). *Adaptive approximate Bayesian computation*.

Bernstein, Gary and Dragan Huterer (2010). “Catastrophic photometric redshift errors: weak-lensing survey requirements”. In: *Monthly Notices of the RAS* 401.2, pp. 1399–1408.

Bertorelle, G., A. Benazzo, and S. Mona (2010). “ABC as a flexible framework to estimate demography over space and time: some cons, many pros”. In: *Molecular Ecology* 19.13, pp. 2609–2625.

Birrer, S. et al. (2019). “H0LiCOW - IX. Cosmographic analysis of the doubly imaged quasar SDSS 1206+4332 and a new measurement of the Hubble constant”. In: *Monthly Notices of the RAS* 484.4, pp. 4726–4753.

Bolzonella, M., J.-M. Miralles, and R. Pelló (2000). “Photometric redshifts based on standard SED fitting procedures”. In: *Astronomy and Astrophysics* 363, pp. 476–492. eprint: astro-ph/0003380.

Bordoloi, R., S. J. Lilly, and A. Amara (2010). “Photo-z performance for precision cosmology”. In: *Monthly Notices of the RAS* 406, pp. 881–895.

Burenin, R. A. and A. A. Vikhlinin (2012). “Cosmological parameters constraints from galaxy cluster mass function measurements in combination with other cosmological data”. In: *Astronomy Letters* 38, pp. 347–363.

Caldeira, J. et al. (2018). *DeepCMB: Lensing Reconstruction of the Cosmic Microwave Background with Deep Neural Networks*.

Charnock, Tom, Guilhem Lavaux, and Benjamin D. Wandelt (2018). "Automatic physical inference with information maximizing neural networks". In: *Physical Review D* 97, 083004, p. 083004.

Cropper, Mark et al. (2013). "Defining a weak lensing experiment in space". In: *Monthly Notices of the RAS* 431, pp. 3103–3126.

Di Valentino, Eleonora, Eric V. Linder, and Alessand ro Melchiorri (2018). "Vacuum phase transition solves the H_0 tension". In: *Physical Review D* 97.4, 043528, p. 043528.

D'Isanto, A. and K. L. Polsterer (2018). "Photometric redshift estimation via deep learning. Generalized and pre-classification-less, image based, fully probabilistic redshifts". In: *Astronomy and Astrophysics* 609, A111, A111.

Dutta, Ritabrata et al. (2016). "Fundamentals and Recent Developments in Approximate Bayesian Computation". In: *Systematic Biology* 66.1, e66–e82.

Dyson, Frank Watson, Arthur Stanley Eddington, and C. Davidson (1920). "IX. A determination of the deflection of light by the sun's gravitational field, from observations made at the total eclipse of May 29, 1919". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 220.571-581, pp. 291–333.

Einstein, Albert (1917). "Cosmological Considerations in the General Theory of Relativity". In: *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)* 1917, pp. 142–152.

Filippi, Sarah et al. (2011). *On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo*.

Fisher, R.A. (1925). *Statistical Methods for Research Workers*. Edinburgh Oliver & Boyd.

Fluri, Janis, Tomasz Kacprzak, Aurelien Lucchi, et al. (2019). *Cosmological constraints with deep learning from KiDS-450 weak lensing maps*.

Fluri, Janis, Tomasz Kacprzak, Alexandre Refregier, et al. (2018). "Cosmological constraints from noisy convergence maps through deep learning". In: *Physical Review D* 98.12, 123518, p. 123518.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Gu, Jiuxiang et al. (2015). *Recent Advances in Convolutional Neural Networks*.

Gupta, Arushi et al. (2018). "Non-Gaussian information from weak lensing data via deep learning". In: *Physical Review D* 97, 103515, p. 103515.

Hahn, ChangHoon et al. (2017). "Approximate Bayesian computation in large-scale structure: constraining the galaxy-halo connection". In: *Monthly Notices of the RAS* 469, pp. 2791–2805.

Hamana, Takashi et al. (2002). "Source-lens clustering effects on the skewness of the lensing convergence". In: *Monthly Notices of the RAS* 330.2, pp. 365–377.

He, Kaiming et al. (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*.

He, Siyu et al. (2018). *Learning to Predict the Cosmological Structure Formation*.

Heavens, Alan F. et al. (2017). "Massive data compression for parameter-dependent covariance matrices". In: *Monthly Notices of the RAS* 472, pp. 4244–4250.

Heymans, Catherine et al. (2013). "CFHTLenS tomographic weak lensing cosmological parameter constraints: Mitigating the impact of intrinsic galaxy alignments". In: *Monthly Notices of the RAS* 432, pp. 2433–2453.

Hildebrandt, H. et al. (2017). "KiDS-450: cosmological parameter constraints from tomographic weak gravitational lensing". In: *Monthly Notices of the RAS* 465, pp. 1454–1498.

Hinshaw, G. et al. (2013). "Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Parameter Results". In: *The Astrophysical Journal Supplement Series* 208, 19, p. 19.

Ho, Matthew et al. (2019). *A Robust and Efficient Deep Learning Method for Dynamical Mass Measurements of Galaxy Clusters*.

Hoekstra, Henk, Massimo Viola, and Ricardo Herbonnet (2017). "A study of the sensitivity of shape measurements to the input parameters of weak-lensing image simulations". In: *Monthly Notices of the RAS* 468, pp. 3295–3311.

Hooper, Dan, Gordan Krnjaic, and Samuel D. McDermott (2019). *Dark Radiation and Superheavy Dark Matter from Black Hole Domination*.

Howlett, Cullan et al. (2012). "CMB power spectrum parameter degeneracies in the era of precision cosmology". In: *Journal of Cosmology and Astro-Particle Physics* 2012.4, 027, p. 027.

Hu, Wayne (1999). "Power Spectrum Tomography with Weak Lensing". In: *Astrophysical Journal* 522, pp. L21–L24.

Hubble, E. (1929). "A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae". In: *Proceedings of the National Academy of Science* 15, pp. 168–173.

Jasche, Jens and Guilhem Lavaux (2015). "Matrix-free large-scale Bayesian inference in cosmology". In: *Monthly Notices of the RAS* 447, pp. 1204–1212.

Johnson, M.E., L.M. Moore, and D. Ylvisaker (1990). "Minimax and maximin distance designs". In: *Journal of Statistical Planning and Inference* 26.2, pp. 131–148.

Kaiser, Nick (1992). "Weak Gravitational Lensing of Distant Galaxies". In: *Astrophysical Journal* 388, p. 272.

Kaiser, Nick, Gillian Wilson, and Gerard A. Luppino (2000). *Large-Scale Cosmic Shear Measurements*.

Kiessling, Alina et al. (2015). "Galaxy Alignments: Theory, Modelling &amp; Simulations". In: *Space Science Reviews* 193, pp. 67–136.

Kilbinger, Martin (2015). "Cosmology with cosmic shear observations: a review". In: *Reports on Progress in Physics* 78, 086901, p. 086901.

Kilbinger, Martin et al. (2017). "Precision calculations of the cosmic shear power spectrum projection". In: *Monthly Notices of the RAS* 472.2, pp. 2126–2141.

Kitching, T. D. et al. (2014). "3D cosmic shear: cosmology from CFHTLenS". In: *Monthly Notices of the RAS* 442, pp. 1326–1349.

Köhlinger, F. et al. (2017). "KiDS-450: the tomographic weak lensing power spectrum and constraints on cosmological parameters". In: *Monthly Notices of the RAS* 471, pp. 4412–4435.

Kuijken, K. et al. (2015). "Gravitational lensing analysis of the Kilo-Degree Survey". In: *Monthly Notices of the RAS* 454, pp. 3500–3532.

Laureijs, R. et al. (2011). *Euclid Definition Study Report*.

Laurino, O. et al. (2011). "Astroinformatics of galaxies and quasars: a new general method for photometric redshifts estimation". In: *Monthly Notices of the RAS* 418, pp. 2165–2195.

Lawrence, Earl et al. (2017). "The Mira-Titan Universe. II. Matter Power Spectrum Emulation". In: *Astrophysical Journal* 847, 50, p. 50.

Lewis, Antony, Anthony Challinor, and Anthony Lasenby (2000). "Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models". In: *Astrophysical Journal* 538.2, pp. 473–476.

Limber, D. Nelson (1953). "The Analysis of Counts of the Extragalactic Nebulae in Terms of a Fluctuating Density Field." In: *Astrophysical Journal* 117, p. 134.

Lin, Chieh-An and Martin Kilbinger (2015). "A new model to predict weak-lensing peak counts. II. Parameter constraint strategies". In: *Astronomy and Astrophysics* 583, A70, A70.

LSST Science Collaboration et al. (2009). *LSST Science Book, Version 2.0*.

Ma, Zhaoming, Wayne Hu, and Dragan Huterer (2006). "Effects of Photometric Redshift Uncertainties on Weak-Lensing Tomography". In: *Astrophysical Journal* 636.1, pp. 21–29.

Macaulay, E., I. K. Wehus, and H. K. Eriksen (2013). "Lower Growth Rate from Recent Redshift Space Distortion Measurements than Expected from Planck". In: *Physical Review Letters* 111.16, 161301, p. 161301.

McClintock, Thomas and Eduardo Rozo (2019). *Reconstructing Probability Distributions with Gaussian Processes*.

McClintock, T. et al. (2019). "The Aemulus Project. II. Emulating the Halo Mass Function". In: *Astrophysical Journal* 872, 53, p. 53.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2, pp. 239–245.

Merkel, Philipp M. and Björn Malte Schäfer (2017). "Parameter constraints from weak-lensing tomography of galaxy shapes and cosmic microwave background fluctuations". In: *Monthly Notices of the RAS* 469.3, pp. 2760–2770.

Narayan, Ramesh and Matthias Bartelmann (1996). *Lectures on Gravitational Lensing*.

Negrello, Mattia et al. (2010). "The Detection of a Population of Submillimeter-Bright, Strongly Lensed Galaxies". In: *Science* 330, p. 800.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2014). *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*.

Peel, Austin et al. (2018). *Distinguishing standard and modified gravity cosmologies with machine learning*.

Penzias, A. A. and R. W. Wilson (1965). "A Measurement of Excess Antenna Temperature at 4080 Mc/s." In: *Astrophysical Journal* 142, pp. 419–421.

Perlmutter, S. et al. (1999). "Measurements of Ω and Λ from 42 High-Redshift Supernovae". In: *Astrophysical Journal* 517, pp. 565–586.

Petri, Andrea, Zoltán Haiman, and Morgan May (2017). "Validity of the Born approximation for beyond Gaussian weak lensing observables". In: *Physical Review D* 95.12, 123503, p. 123503.

Planck Collaboration, P. A. R. Ade, et al. (2016). "Planck 2015 results. XIII. Cosmological parameters". In: *Astronomy and Astrophysics* 594, A13, A13.

Planck Collaboration, N. Aghanim, et al. (2018). *Planck 2018 results. VI. Cosmological parameters*.

Rasmussen, Carl Edward and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ravanbakhsh, Siamak et al. (2016). "Estimating Cosmological Parameters from the Dark Matter Distribution". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 2407–2416.

Refregier, Alexandre R., Richard S. Ellis, and David J. Bacon (2000). "Detection of weak gravitational lensing by large-scale structure". In: *Monthly Notices of the Royal Astronomical Society* 318.2, pp. 625–640.

Ribli, Dezső, Bálint Ármin Pataki, and István Csabai (2019). "An improved cosmological parameter inference scheme motivated by deep learning". In: *Nature Astronomy* 3, pp. 93–98.

Riess, Adam G., Stefano Casertano, et al. (2019). "Large Magellanic Cloud Cepheid Standards Provide a 1% Foundation for the Determination of the Hubble Constant and Stronger Evidence for Physics beyond ΛCDM". In: *Astrophysical Journal* 876.1, 85, p. 85.

Riess, Adam G., Alexei V. Filippenko, et al. (1998). "Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant". In: *Astronomical Journal* 116, pp. 1009–1038.

Salvato, M. et al. (2009). "Photometric Redshift and Classification for the XMM-COSMOS Sources". In: *Astrophysical Journal* 690, pp. 1250–1263.

Samushia, Lado, Beth A. Reid, Martin White, Will J. Percival, Antonio J. Cuesta, Lucas Lombriser, et al. (2013). "The clustering of galaxies in the SDSS-III DR9 Baryon Oscillation Spectroscopic Survey: testing deviations from Λ and general relativity using anisotropic clustering of galaxies". In: *Monthly Notices of the RAS* 429, pp. 1514–1528.

Samushia, Lado, Beth A. Reid, Martin White, Will J. Percival, Antonio J. Cuesta, Gong-Bo Zhao, et al. (2014). "The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: measuring growth rate and geometry with anisotropic clustering". In: *Monthly Notices of the RAS* 439, pp. 3504–3519.

Sánchez, Ariel G. et al. (2012). "The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: cosmological implications of the large-scale two-point correlation function". In: *Monthly Notices of the RAS* 425, pp. 415–437.

Schaan, Emmanuel et al. (2017). "Looking through the same lens: Shear calibration for LSST, Euclid, and WFIRST with stage 4 CMB lensing". In: *Physical Review D* 95.12, 123512, p. 123512.

Schmelzle, Jorit et al. (2017). *Cosmological model discrimination with Deep Learning*.

Schneider, P., L. van Waerbeke, and Y. Mellier (2002). "B-modes in cosmic shear from source redshift clustering". In: *Astronomy and Astrophysics* 389, pp. 729–741.

Scoccimarro, Román, Matias Zaldarriaga, and Lam Hui (1999). "Power Spectrum Correlations Induced by Nonlinear Clustering". In: *Astrophysical Journal* 527.1, pp. 1–15.

Seikel, Marina and Chris Clarkson (2013). *Optimising Gaussian processes for reconstructing dark energy dynamics from supernovae*.

Sellentin, Elena and Alan F. Heavens (2018). "On the insufficiency of arbitrarily precise covariance matrices: non-Gaussian weak-lensing likelihoods". In: *Monthly Notices of the RAS* 473.2, pp. 2355–2363.

Simon, P., L. J. King, and P. Schneider (2004). "The covariance of cosmic shear correlation functions and cosmological parameter estimates using redshift information". In: *Astronomy and Astrophysics* 417, pp. 873–885.

Smith, R. E. et al. (2003). "Stable clustering, the halo model and non-linear cosmological power spectra". In: *Monthly Notices of the RAS* 341.4, pp. 1311–1332.

Sun, Lei et al. (2009). "Catastrophic Photo-z Errors and the Dark Energy Parameter Estimates with Cosmic Shear". In: *Astrophysical Journal* 699.2, pp. 958–967.

Szegedy, Christian et al. (2013). *Intriguing properties of neural networks*.

Takada, Masahiro and Sarah Bridle (2007). "Probing dark energy with cluster counts and cosmic shear power spectra: including the full covariance". In: *New Journal of Physics* 9.12, p. 446.

Takada, Masahiro and Bhuvnesh Jain (2004). "Cosmological parameters from lensing power spectrum and bispectrum tomography". In: *Monthly Notices of the RAS* 348.3, pp. 897–915.

— (2009). "The impact of non-Gaussian errors on weak lensing surveys". In: *Monthly Notices of the RAS* 395.4, pp. 2065–2086.

Takahashi, Ryuichi et al. (2012). "Revising the Halofit Model for the Nonlinear Matter Power Spectrum". In: *Astrophysical Journal* 761.2, 152, p. 152.

The Dark Energy Survey Collaboration (2005). *The Dark Energy Survey.*

Troxel, M. A. and Mustapha Ishak (2015). "The intrinsic alignment of galaxies and its impact on weak gravitational lensing in an era of precision cosmology". In: *Physics Reports* 558, pp. 1–59.

Troxel, M. A., N. MacCrann, et al. (2018). "Dark Energy Survey Year 1 results: Cosmological constraints from cosmic shear". In: *Physical Review D* 98, 043528, p. 043528.

Tsapras, Yiannis (2018). "Microlensing Searches for Exoplanets". In: *Geosciences* 8, p. 365.

van Uitert, Edo et al. (2018). "KiDS+GAMA: cosmology constraints from a joint analysis of cosmic shear, galaxy-galaxy lensing, and angular clustering". In: *Monthly Notices of the RAS* 476, pp. 4662–4689.

Van Waerbeke, L. et al. (2000). "Detection of correlated galaxy ellipticities from CFHT data: first evidence for gravitational lensing by large-scale structures". In: *Astronomy and Astrophysics* 358, pp. 30–44. eprint: astro-ph/0002500.

Vattis, Kyriakos, Savvas M. Koushiappas, and Abraham Loeb (2019). "Dark matter decaying in the late Universe can relieve the H_0 tension". In: *Physical Review D* 99.12, 121302, p. 121302.

Wasserman, Larry (2010). *All of Statistics: A Concise Course in Statistical Inference.* Springer Publishing Company, Incorporated.

Weinberg, Steven (1989). "The cosmological constant problem". In: *Rev. Mod. Phys.* 61 (1), pp. 1–23.

Wittman, D. M. et al. (2000). "Detection of weak gravitational lensing distortions of distant galaxies by cosmic dark matter at large scales". In: *Nature* 405, pp. 143–148.

Wu, Chen et al. (2019). "Radio Galaxy Zoo: CLARAN - a deep learning classifier for radio morphologies". In: *Monthly Notices of the RAS* 482, pp. 1211–1230.

Zednik, Carlos (2019). *Solving the Black Box Problem: A General-Purpose Recipe for Explainable Artificial Intelligence.*

Zuntz, J. et al. (2015). "CosmoSIS: Modular cosmological parameter estimation". In: *Astronomy and Computing* 12, pp. 45–59.