Utrecht University

Faculty of Natural Sciences
Master of Science in Artificial Intelligence

Master Thesis

Emulating effective field theory predictions for galaxy alignments

Presented by:
Ignacio Montes-Álvarez

Academic course:
2022-2023

Emulating effective field theory predictions for galaxy alignments

Ignacio Montes-Álvarez

Ignacio Montes-Álvarez
Emulating effective field theory predictions for galaxy alignments.

Master of Science in Artificial Intelligence.
Faculty of Natural Sciences.
University of Utrecht.

Master Thesis. Academic course 2022-2023.

Advisor(s)

Dr. Nora-Elisa Chisari
Department of Physics

Dr. Siamak Mehrkanoon
Department of Information and Computing Sciences

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Cosmology advances in the last decades due to recent influxes of observational data have posed to be a challenge for the current numerical methods used. Due to the nature of analysis of this data, speed and accuracy constraints need to be strongly considered. Statistical methods that leverage Artificial Intelligence have started to populate Cosmology literature as an efficient and optimal option. In this work, we will apply these kind of techniques, and more specifically Gaussian Processes for regression to create a fast-prediction framework of the components of the Effective Field Theory model in order to enable prediction of the information provided by intrinsic alignments and how these act as contaminants in cosmological surveys. The aim of this project is to train AI models that are on-par with numerical methods but reduce the computational complexity in several orders of magnitude. We have created a novel emulator for the shape correlations of intrinsic alignments under the EFT theory by implementing a Python package for Gaussian Process emulation with GPU support using the GPyTorch library. We managed to accurately predict the EFT correlators along the power spectrum with relative errors in the sub-percent order. Sequential execution for a single cosmology takes a few seconds while in the parallel case this is reduced to the order of just milliseconds.

1 Introduction

Galaxies contain systems of stars, dust, gas and dark matter among others, held altogether by the force of gravity. Their size ranges from a few kilo-parsecs[1] to more than a hundred and they can contain up to hundreds of billions of stars. Galaxies are the perfect candidates to study the evolution of the Universe because they are the amplest objects outside our own galaxy, the Milky Way [45].

Furthermore, in the past decades, several optical[2] and spectroscopic[3] sky surveys have been conducted providing us with images of unprecedented resolution of the most distant patches of our Universe. Among these studies (and chronologically) we can find the early Hubble missions [7, 75], the BOSS [25], the KiDS-450 [41] and 1000 [40] surveys, the SDSS-IV [9] or the LSST mission [44]. It is also worth mentioning that there are many other missions ongoing or planned such as the Euclid [31] whose data will be able to answer a wide range of astrophysical questions, specially if combined with other powerful surveys such as LSST [61].

It comes with the increase in quality and amount of data, the need for using better methods to analyze the influx of information to draw accurate and efficient[4] conclusions. All of this combined the above-mentioned observational efforts, have led the field of Cosmology to a new scientific era.

1.1 Preliminaries

To bring context to the project it is useful to draw attention on why we will use galaxies as candidates for our research.

1.1.1 Galaxies and the redshift

Galaxies can help explain how the Universe is evolving based on some of their fundamental properties. Using their position and orientation in the sky, how we perceive their size or their color can help to understand how galaxies organize or cluster into what is called the large-scale structure of the Universe. This structure formation takes

[1] See **Parsec** in the glossary.
[2] Common imagery.
[3] Study of the radiation spectra emitted by matter to infer characteristics such as chemical composition or velocities of objects.
[4] In terms of computational complexity, both in size and in time.

place on an expanding, homogeneous[5] and isotropic[6] spacetime. The latter two concepts are the key to what is called the Cosmological Principle, which states that all systems of reference (whichever position of an observer in the Universe) would perceive the Universe's expansion equally. But what is this expansion and how do we know that is taking place?
Galaxies move with really small velocities when compared with the expansion, that is, the intrinsic force of gravity of the galaxies (holding all their components together) is stronger that the expansion but it is the space in which they are embedded which really cause them to move. But at the same time, this can be detected using the light that we perceive from this distant objects and it was firstly discovered by Edwin Hubble in the early 20th century. He compared the distance of a set of galaxies with respect to some standard candles[7] and found some deviations on their spectra towards the red, the so-called redshift[8] effect [42].

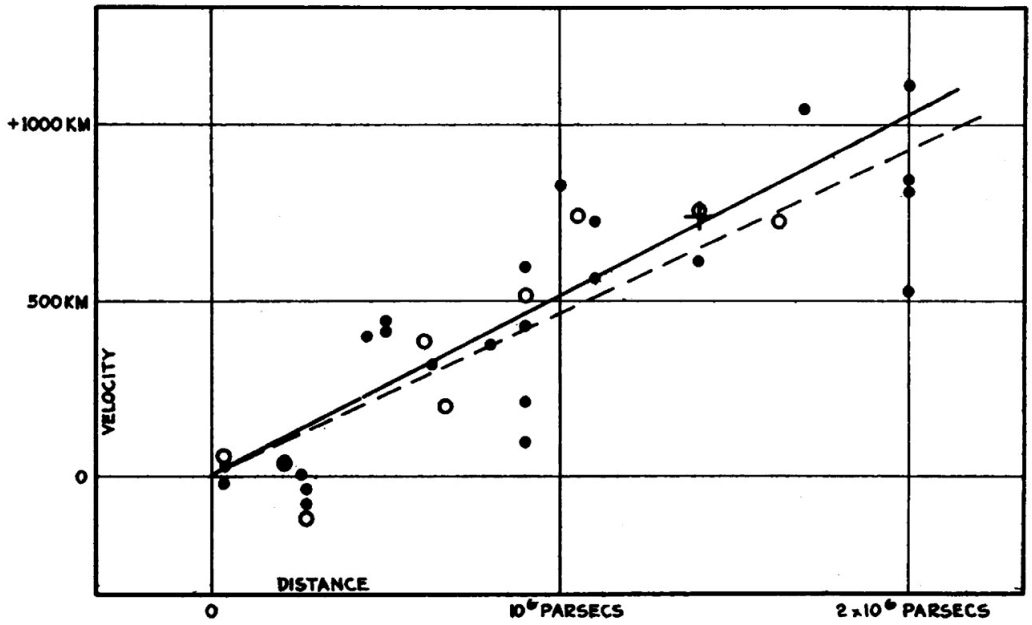

Figure 1.1: "Velocity-Distance Relation among Extra-Galactic Nebulae". The relation displayed in the plot shows that the further away a galaxy is receding from us, the greater the receding speed is.

Source: [42]

The relation shown in Figure 1.1 is know as Hubble's law and is mathematically stated as:

[5]Over large enough areas ($\gtrsim 100 Mpc$), it is very difficult to distinguish one area from another.
[6]The Universe has the same properties no matter where we look.
[7]An astronomical object with some known absolute luminosity magnitude which can be used to derive the distance to the same. Among the different types, the most common are the Cepheid Variables (which Hubble used in his study). They are used when the **parallax** (see in the glossary) method cannot be longer applied accurately.
[8]See **Redshift** in the glossary.

$$v = H_0 d$$

If the value of H_0, namely the Hubble's constant, can be properly calibrated (making use of objects with known distances and velocities), one can directly infer the distance to any galaxy given its velocity, which is in turn given by its redshift.

1.1.2 The Cosmic Microwave Background

It is mandatory to introduce the concept of the Cosmic Microwave Background (CMB) as it is the limit on how far we can look back in time. When the Big Bang happened, it was so hot that electrons and protons were not able to interact and it was not until they cooled down to a temperature of around 3000K after 3.79×10^5 years that these particles entered in a recombination period to form the first atoms. The photons[9] that did not interact with these atoms began travelling freely and gave birth to a radiation body which nowadays is observed homogeneously no matter where we look. There are some patterns present in the CMB[10] (Figure 1.2) and there has been several missions to study them such as the Planck missions [57, 58]. These have helped to adjust the parameters that govern the energy density present in the Universe (the Friedmann equation) which has led to conclude that the matter that we can observe, baryonic matter, makes up 5% of the total whilst the remaining 27% and 68% are dark matter and dark energy respectively. The nature of the latter two is yet unknown as they do not interact with electromagnetic fields but are needed to account for some gravitational effects that influence the Universe's structure and evolution.

[9] Packets of electromagnetic radiation.

[10] Denominated Baryonic Acoustic Oscillations (BAO), which correspond to perturbations that propagated in the way of acoustic waves.

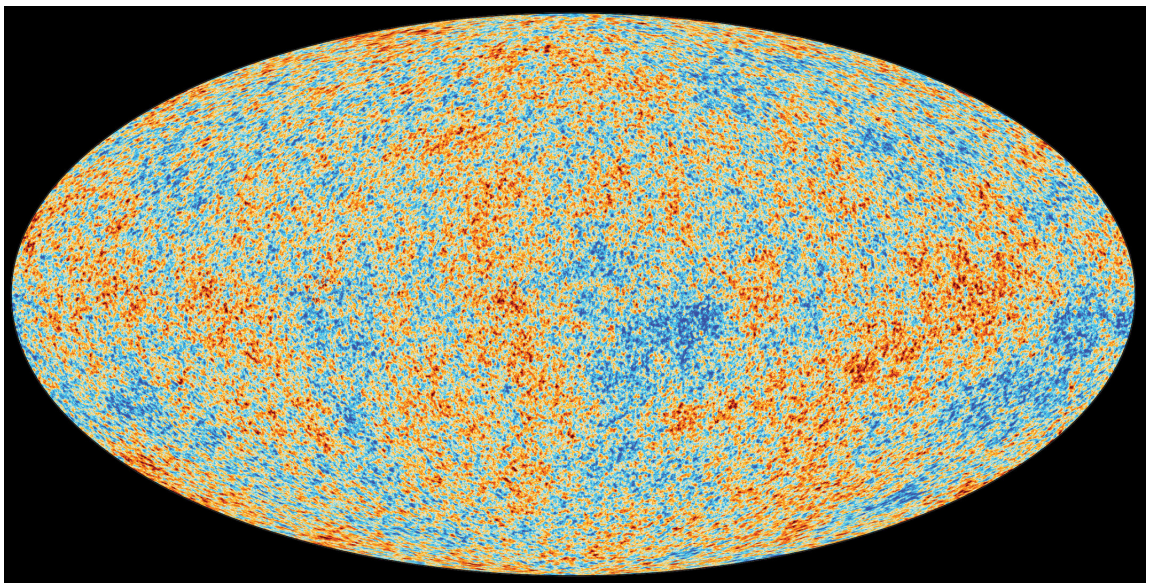

Figure 1.2: The Cosmic Microwave Background as measured by the Planck'18 mission

Source: [58], https://sci.esa.int/s/wQdrX4A

What is most interesting is the relation of the CMB with the structure formation as in smaller scales. We can see that patterns in this radiation wall have led to the emergence of structures of galaxy clusters bound by filaments and surrounded by voids forming a 'cosmic web'[11]. The galaxy distribution that we see today has grown from the anisotropies (changes) in the CMB in combination with force of gravity. However, the latter dilutes these patterns over time but still, they can via studied by measuring the correlation between galaxy positions in the sky.

1.1.3 Galaxy distribution measurements

Galaxy imagery is a key component in Cosmology, via this observations we are able to define the large-scale of the Universe, how they organize in the 'cosmic web'. These images cannot be directly used as data-sources as there are certain effects that introduce distortions in them but at the same time acting as source of information worth of study as it carries cosmological information. One of them is gravitational lensing (γ). This artifact occurs when the light emitted by a galaxy passes near an over-density such as cluster of galaxies or a dark matter field, making the rays appear distorted from our point of view; the gravitational field creates a divergent 'lense' which deviates the light from its normal path and makes the background object appear as if there were multiple (Figure 1.3. This effect is most notable over long distances.

[11] See **Cosmic web** in the glossary.

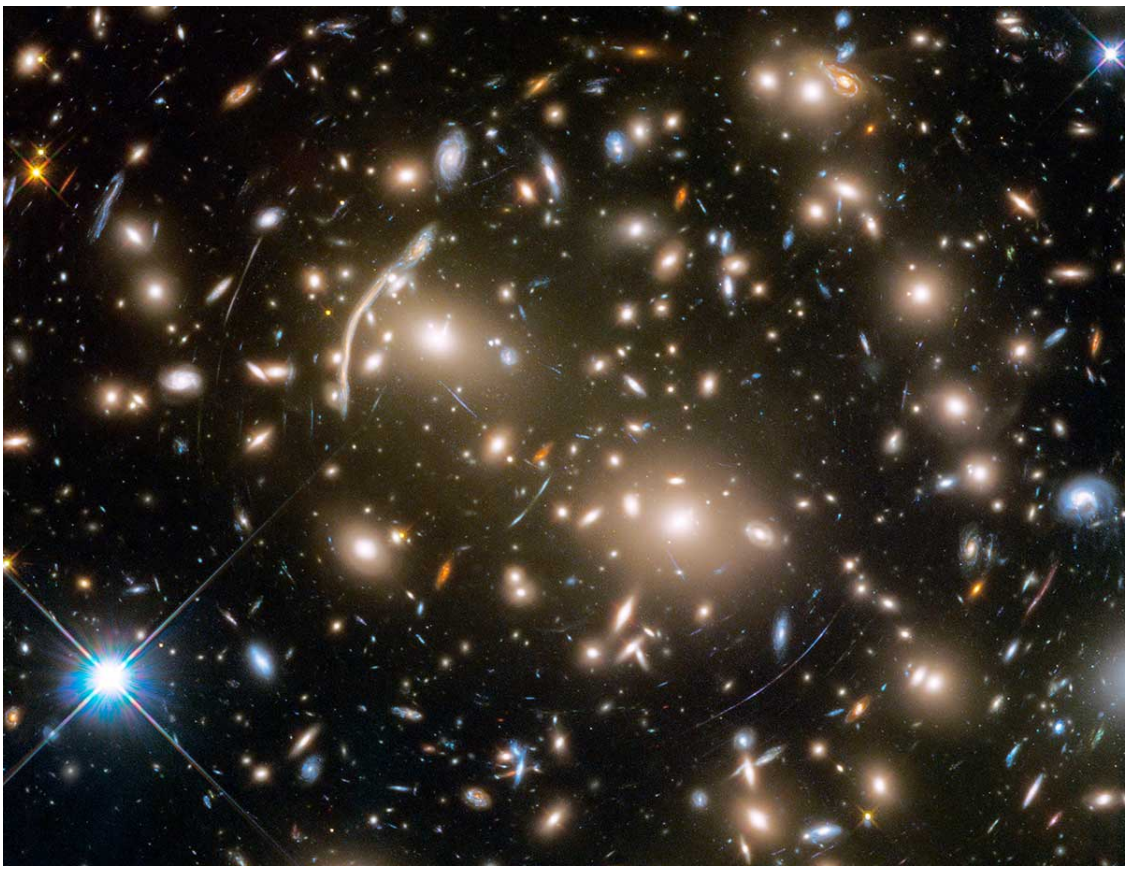

Figure 1.3: The cluster of galaxies denominated Abell 370 is located 4 billion light-years away and acts as a lense to the galaxies in the background; this effect cause galaxies to appear multiplied and distorted. One of the most notable examples is "the Dragon": several duplicated images of a single background spiral galaxies that are stretched along an arc (it can be appreciated on the upper left part of the image).

Source: https://hubblesite.org/contents/news-releases/2017/news-2017-20.html

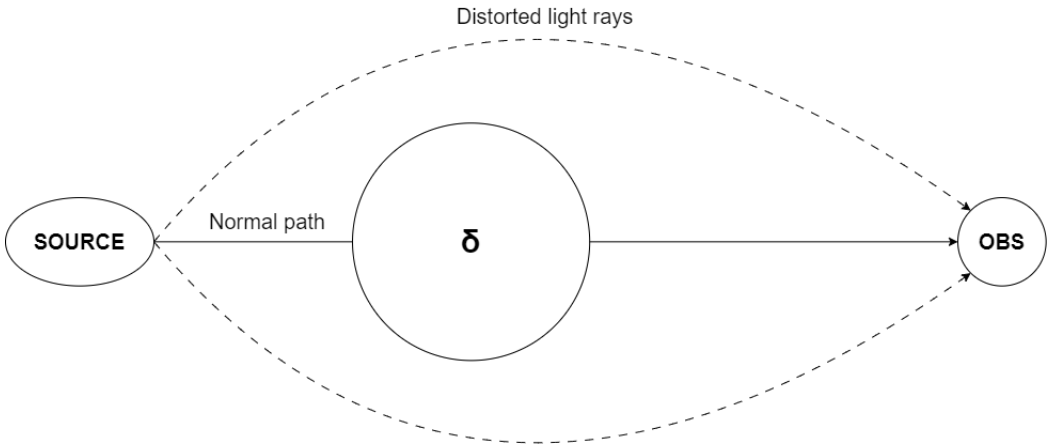

Figure 1.4: Schematic representation of gravitational lensing in which a background source object emits lights which is diverged by an over-density δ in such a way that we perceive multiple distorted objects around the foreground one.

There is another source of deformation, the so-called intrinsic alignments (I). This is directly related to how gravitational fields deform the galaxies themselves. This effect is caused by tidal forces generated by over-densities such as dark matter fields and cause changes in the orientation of galaxies. It is most notable when galaxies are too close or when there is not enough structure to produce strong gravitational lensing such in the closer Universe ($z < 0.2 - 0.3$).

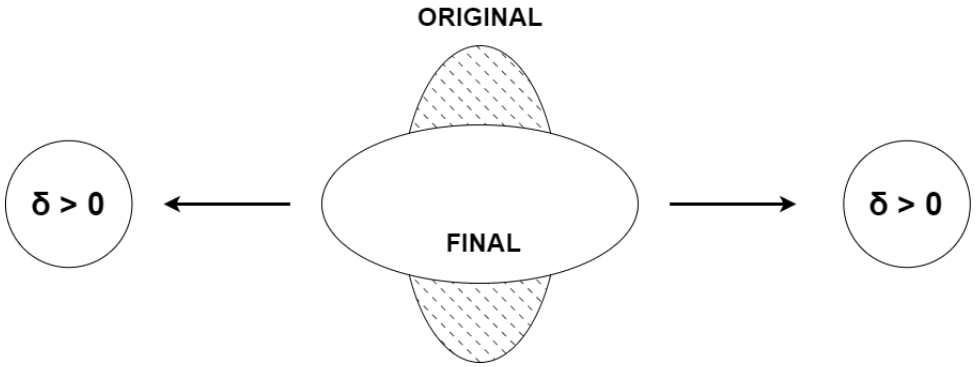

Figure 1.5: Schematic representation of an intrinsic alignments in which two over-densities δ change the original orientation of a galaxy to a final state in which its axis is pointing in the direction of the gravitational fields.

Lensing and alignments are tightly coupled, so it is difficult to know which one caused the deformation. Then, the total shape of a galaxy is given by the following equation:

$$e_+ \propto \gamma + I + Noise \tag{1.1}$$

There are other sources of noise attributed to the intrinsic characteristics of the galaxy such as the shape of the galaxy (disk, elliptical...), whether the galactic core is active or not (increasing luminosity), the fusion with other galaxies. But in reality what we want to measure is the correlation between galaxies which is given by the generalization of Equation (1.1):

$$\xi_{++} \propto \gamma\gamma + \gamma I + II + Noise \tag{1.2}$$

By studying this correlations we are able to constrain the cosmological parameters of the model which defines how energy and matter distributes in the Universe, the Lambda cold dark matter (ΛCDM) model [27]. Among this parameters we can find the density of baryonic (ω_b) and dark matter (ω_c), the Hubble constant (H_0), the spectral index (n_s) or the growth rate (f) among many others. Some of these parameters will be our target in this study for defining ξ_{++}:

- The addition of ω_b and ω_c.
- The spectral index, n_s.
- The primordial amplitude, A_s, when inflation occurred (or in turn σ_8 which corresponds to the amplitude of the density fluctuations).
- And the dark energy state equation.

The latter item is defined as:

$$w(z) = w_0 + w_z(1 - z)$$

Where w_z is defined as $w'(z = 0)$ and w_0 represents the type of fluid that the dark energy is and if its value is -1, the dark energy is a constant which is often referred as Λ. By combining results from different surveys and studies it is expected that the value of this constant would be bounded. The inclusion of such a (non-zero) constant would agree with Einstein's general relativity and support the theory of an expanding Universe as defined in the ΛCDM model.

1.1.4 The matter power spectrum

The matter power spectrum is the density contrast of the Universe (i.e., the difference between local density and average density) as a function of scale. It can be thought as the Fourier transform of the matter-matter correlation function (ξ_{mm}, same as Equation (1.2) but for matter correlation) or, in other words, the probability of finding another galaxy within a given distance. The spectrum is different given the scale:

- Small: the regime (density contrast) is non-linear and alignments are really strong. Usually, the modelling approach used here is the Halo Occupation Distribution (HOD) [8, 78].
- Intermediate: the regime is quasi-linear (Gaussian) and because it is Fourier independent, the power spectrum itself is enough to describe the density field.
- Large: the regime is linear, gravity competes with the expansion and it depends just on a scalar parameter. It can be used to model inflation, where shapes correspond to anisotropies in the tidal forces.

The matter power spectrum describes how much structure there is in different scales described above, where high values correspond to more structure and vice versa (Figure 1.6).

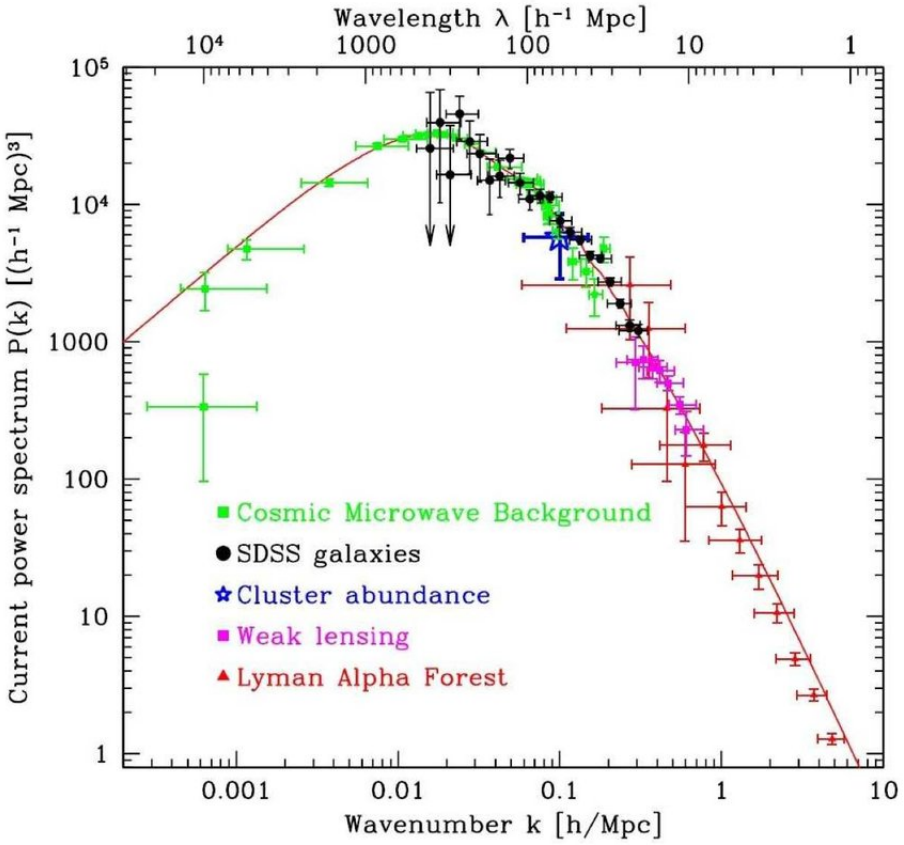

Figure 1.6: Several measurements of the power spectrum, in this case highlighting the SDSS results (in black) that lie in the quasi-linear regime where the BAO wiggles are present.

Source: [69]

1.2 Motivation

Current and future missions such as Euclid [31] will provide an immense amount of information to study the large-scale structure formation and evolution. How to process this data requires of a methodology which needs to be accurate (at the percent level) and more importantly fast. Direct numerical computation, despite being adequate for the first constraint, is too computationally intensive.

Current theoretical mechanisms (such as Effective Field Theory, EFT; more on Section 2.2) for extracting information about galaxy correlations require the calculation of components such as loop integrals[12] which are not scalable and need to be sped up. Statistical approximations with methods that leverage Artificial Intelligence (AI) are one of the best options, as they could provide results with almost no error when compared with the ground truth data and at the same time saving up computation resources.

AI models can be constrained in the same way as any other classical model provided that they get to see representative data of the problem at hand. This enables easy customization of the model at training time just by modifying the input data (e.g., adding more independent variables or features that describe or condition the dependent variables or outputs).

1.3 Aim of the project

The main goal of this research is to create a framework to train AI models to provide a quick and accurate (we will target absolute relative errors at the sub-percent level, this metric will be extensively elaborated on Section 4) interface for inference of several characteristics related to galaxies' intrinsic alignments. The models will be based on the Gaussian Process (GP) architecture as it has been shown to be useful in other studies (more on Section 2.3). It will leverage GPU[13] computing capabilities, much in line with the recent AI developments that take advantage of their hardware characteristics to enable fast matrix multiplications which in turns helps dealing with vast amounts of data. For this purpose will make use of the Python library GPyTorch [33]. We will aim to achieve inference speeds of the order of milliseconds.

The setup will be that of a supervised multi-output regression problem in which for a given set of cosmological parameters we will need to infer a set of factors representing different components of an EFT model. These factors correspond to a set of one-loop integrals extracted from a power spectrum which in turn was generated from a fiducial (ground-truth) cosmology.

[12] See **Loop integral** in the glossary.
[13] Graphics Processing Unit.

Generating a dataset of fiducial cosmologies from different combinations of cosmological parameters is also needed as a preliminary step in order to have a representative sample of our problem. Furthermore, the fiducial set of integrals needs to be numerically extracted to be used as comparison baseline to the predictions made by the model.

We will train different models constrained to different ranges of data and cosmological parameters (different datasets) to serve increasing accuracies at the expense of size and inference speed; presenting the user with diverse pre-trained options that will tailor to their needs.

2 Related work

2.1 General overview

There have been some previous studies related to galaxy intrinsic alignments. In general, what we study is spatial correlations of ellipticities not of single galaxies but of larger galaxy samples via the matter density contrast, i.e., the matter power spectrum. Extracting the contaminant factor that pose alignments is closely linked to the analysis of gravitational lensing and galaxy clustering. Several models have been proposed to study this effect [45]:

- Halo[14] shape models: applied to elliptical galaxies (usually red), measure the alignment of the baryonic matter with dark matter halo which hosts the structure due to alike properties.
- Tidal alignment models: applied to elliptical galaxies (usually blue), measure how the gravitational tidal field generated by the large-scale structure, in which the galaxies are embedded, acts and distorts the baryonic matter.
- Quadratic alignment models: applied to spiral galaxies, measure the perturbative effect of the angular momentum of the host halo and how it changes the galaxies' symmetry axis.
- Vorticity models: through simulations, measure the strong alignments between dark matter haloes and the large-scale structure non-linear vorticity.
- Acceleration models: measure the shape and orientation, based on the pattern of accretion of matter onto the galaxy.
- Merging models: applied to spiral galaxies, measure the angular momentum of the galaxy caused by the infall of other galaxies (fusion). The total angular momentum results from the combination of the angular momenta of the initial galaxies.

All of the above-mentioned models share the fact that the alignments are determined by the interaction of the galaxies with the fields in which they are embedded.

Some other studies have been carried to observe intrinsic alignments, such an early one that used photographic data and considered very low redshifts in which lensing is minuscule, thus correlations should be caused by intrinsic alignments [15] or later ones that used instead spectroscopy applied again to low redshifts [12, 21]; all of them

[14] See **Dark matter halo** in the glossary.

agreeing in the fact that intrinsic alignments are present and account for a few percents of the contaminants.

Models for early (elliptical) and late-type (spirals) galaxies use different free parameters to study them. In summary, the former state that the amplitude of the intrinsic alignments for a galaxy scale linearly with its luminosity [46, 67] whilst the later state that the ellipticity of a galaxy depends on the angular momentum direction; in this case it is the large-scale structure the one that exerts a torque on the halo in which the galaxy is embedded [23, 24]. For this discussed kind of models one needs to model the dynamics of the dark matter haloes and take into account the changes induced by the process of merging haloes.

Finally, there are other theories to explain alignments such as ones that simply pose that the stellar component (galaxy) follows the shape and orientation of the halo in which they are embedded; this could be determined by the initial (and random) conditions of the emerging structure or, as found in simulations, be determined by the vorticity fields or the cold gas accretion streams.

2.2 Modelling intrinsic alignments

As stated in Section 1.1.3, measurements of ξ_{++} are used to constrain the cosmological parameters. Whenever we want to model these, we usually have some observed data and a model that depends on the (cosmological) parameters of interest. Any other extra parameters used in the model are denominated 'nuisance' and just increase the parameter-space which can lead to an increase in the error on the measurements of our parameters of interest.

Constraining the cosmological parameters requires modelling of gravitational lensing but at the same time, and as mentioned in Section 2.1, intrinsic alignments cannot be ignored as they are a contaminant and thus, can bias the parameter estimation. Among the aforementioned methods, the linear tidal alignment model [10, 16] has been commonly adopted in literature. This model works relatively well for alignments which are separated by large distances (of approximately $10h^{-1}Mpc$ or more) and for red galaxies; but where this theory starts to fail is at smaller separations and for blue galaxies [71]. For this non-linear and quasi-linear regimes, methods such the direction of the angular momenta are used, but recent studies have proposed other alternatives to measure intrinsic alignments on this scales in order to obtain accurate and unbiased parameter constraints beyond linear scales. Approximations of the linear power spectra by its nonlinear analogue or HOD have been used and validated via cosmological simulations independently. Since the nonlinear/HOD cannot be approximated, and alternative by combining these two mechanisms has been created under what is called Standard Perturbation Theory (SPT) [11, 13, 65], in order to obtain accurate analytical predictions in the non-linear regimes where the perturbations are small. This model

studies the small perturbations caused by gravity that grow, seeded and eventually give rise to the current structure that we can observe in the Universe.

A variation of SPT that can describe the statistics of any tensorial field in the large-scale structure is called Effective Field Theory (EFT) [71]. It is based on Eulerian (point) perturbation theory and on large-scales describes the matter density distribution as an effective fluid [6] whereas on small-scales all the physics are integrated out and can be compared with simulations or observations in like manner. Furthermore, it solves the issue of convergence at smaller scales due to physical independence thanks to the inclusion of contributions of higher spatial derivatives of the density and tidal fields. So all in all, with EFT, the modelling of intrinsic alignments is extended from the linear regime to the quasi-linear one. It is worth to note that this approach can be applied to both three-dimensional shapes of galaxies as well as to the two-dimensional projection of those that can be encountered in galaxy surveys [73]. Under this framework and using symmetries of trace-free[15] tensors (meaning that size is not taken into account, just the shape) and scalar biased tracers[16] (e.g., a generic count of galaxy population), several characteristics from galaxies can be obtained such their shape, size and more importantly for our research, the effects of intrinsic alignments in biased tracers as observed in galaxy redshift surveys.

Galaxy shapes (their ellipticities) are affected by both gravitational lensing (which is a two-dimensional projection effect) and by intrinsic alignments which arise from physical interactions (thus, happening in a three-dimensional space). The latter, can be derived from the power spectrum (Section 1.1.4) using a one-loop perturbative expression (Feynman diagram[17]). Several terms contribute to this expression and among those there are thirteen independent integral correlators for Equation (1.2) ([71], Section 4.3 and Appendix B.1) and which are the numerical results of an integration routine (Figure 2.1). We will use those as ground-truth as well as target for emulation in our research.

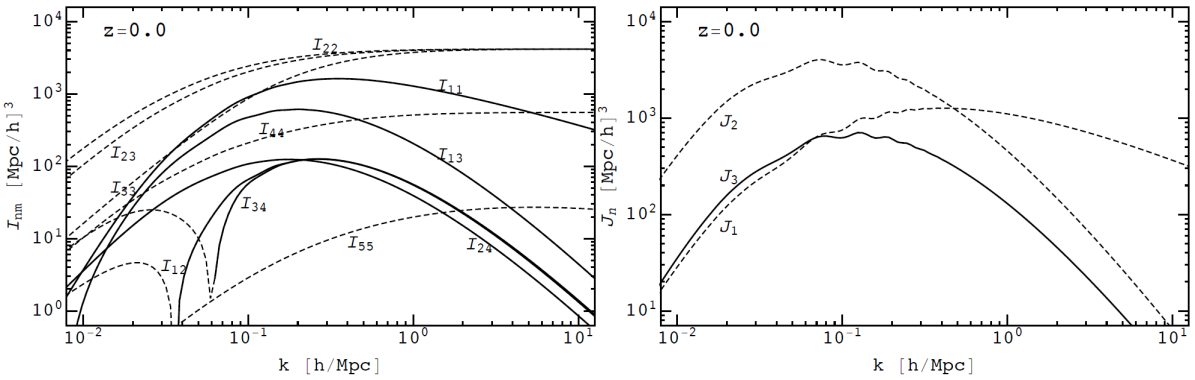

Figure 2.1: One-loop correlators from EFT

[15] $Tr[A] = 0$, i.e., the trace of a given matrix (A) is zero. Being the trace the sum of the elements in the main diagonal.

[16] A tracer is any object or group of objects that help detecting other objects (such as dark matter).

[17] See **Feynman diagram** in the glossary.

2.3 Cosmological emulation

As higher quality and complete datasets from different surveys come online, the cost of running large-volume simulations becomes prohibitive (e.g., running a MCMC[18] could require up to the order of $\mathcal{O}(10^6)$ evaluations). Statistical analysis of cosmological quantities has proven to be an extremely useful tool to sped up computations in contrast to direct numerical calculation; it can retrieve inferred values with errors at the percent and even sub-percent level. This approximation methodology relies on building 'emulators' which are nothing more than AI models.

Several of these models have been constructed over the past years with different aims such as emulating the matter linear power spectra [38], the non-linear power spectra [49, 50, 54], both linear and non-linear [55, 68] or CMB statistics (temperature, polarization...) [68]. The general trend for choosing an architecture for these emulators is that of Gaussian Processes (GPs) [74], specifically GP regression. On the other hand there have been other models based on Neural Networks (NNs) as architecture for the interpolation process [1, 2, 4].

However, not many of this emulators are based on perturbation theory. Under this category fall one-loop tracers emulators [18, 22] which are able to perform inference in the order of 1s, or more closely related to our aim, COMET, an EFT/VDG (Velocity Difference Generating function) emulator [30] which reduces the time to the order of 10ms. COMET uses an approach close to that of neural emulators [26, 28] despite using GPs; it models positions of observables whereas in our case want to model the intrinsic alignments of those. We will closely follow COMET's approach to keep a flexible, but at the same time, small range of cosmological parameters to help keeping computational costs as small as possible without compromising the key advantage of perturbation theory, accuracy.

[18] Markov chain Monte Carlo

3 Methodology

3.1 General overview

In this section we will describe the architecture and process of the emulator design (Figure 3.1), including the parameters chosen for training (with the corresponding justification), the generation methods for the datasets and an overview of the AI pipeline.

3.1.1 Cosmological parameters

The range of parameters currently being sampled in the emulator are in line with the KiDS-1000 priors [40]. This is the fourth public release of the KiDS project and it is a weak lensing survey that covers an area of 1000 $degrees^2$ with images. Statistical analysis of the shearing, clustering and lensing was used in order to constraint the cosmological parameters.

Parameter	Min	Max
ω_c	0.051	0.255
ω_b	0.019	0.026
n_s	0.84	1.1

Table 3.1: Cosmological parameter ranges. ω_c: physical density in cold dark matter, ω_b: physical density in baryons and n_s: scalar spectral index.

Furthermore, $H_0 = 69.5\, km\, s^{-1}\, Mpc^{-1}$ (the Hubble constant) and $A_s = 2.2078559 \times 10^{-9}$ (the comoving curvature power) are fixed in order to be able to set the initial power spectra (values taken from COMET [30]). The redshift value (z) is set to zero.

As we are not sampling H_0 ($h = H_0/100$), the emulator predictions could be exactly re-scaled provided that the input linear power spectra (P_{lin}) are in units of Mpc, that is, removing all the dependencies to h [64]. This technique is called 'evolution mapping' [63] and with it not only the predictions could be scaled but other parameters such as Ω_K, w_0, w_a, A_s and redshift evolution. This approach involves including an extra quantity denominated σ_{12} which represents the variance of the linear density field in spheres of 12 Mpc [64]. This quantity can be absorbed by the norm of the linear power spectrum as in Equation (48) of COMET, thus it can be either be emulated separately or computed numerically a posteriori in the re-scaling phase.

The space of generation for the power spectra is $k'_{min} = 10^{-4}$ and $k'_{max} = 100$ (both in units of Mpc^{-1}) with a total of 700 points, in order to get a fine grained set of spectra in units of Mpc^3.

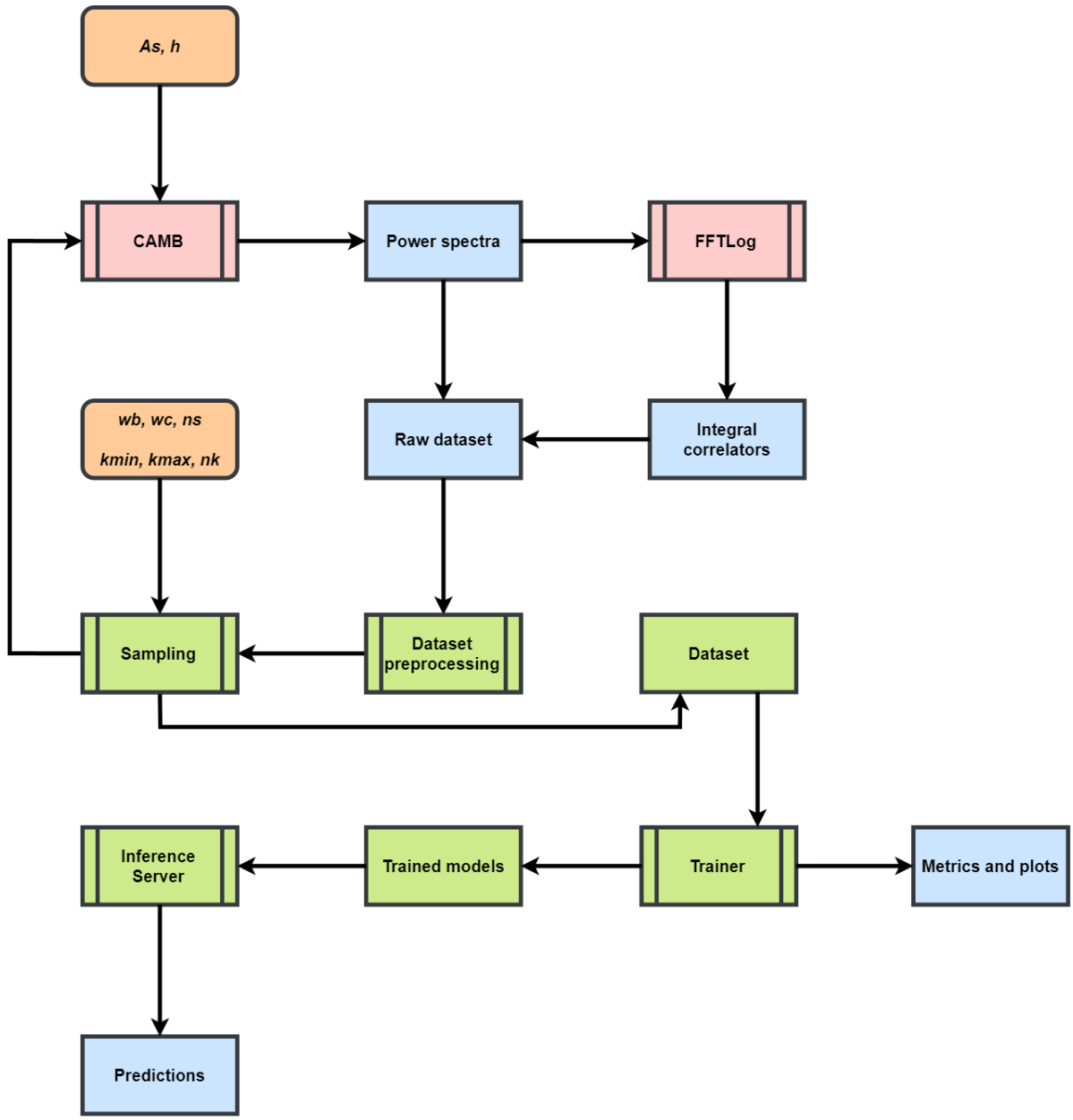

Figure 3.1: General overview of the project's flow, orange represents input parameters (being A_s and h fixed and the rest sampled accordingly in the ranges defined in Section 3.1.1), green represents the AI pipeline components and blue outputs of the different steps. The aforementioned outputs are the product of the double squared boxes, being those in pink external processes independent to the AI pipeline.

3.1.2 Dataset structure

50,000 different cosmologies will be created by applying Latin Hypercube Sampling (more on Section 3.5) over the parameters defined in Table 3.1 and the k'_{min} and k'_{max}. Afterwards, FFTLog [66] will be applied over each one of them (represented by a

sample of just three parameters, namely, ω_c, ω_b and ns) to obtain the 13 one-loop components as per defined in the perturbative effective field theory (EFT) [72].
For each cosmology, 13 components will be generated for each one of the 700 k-bins and in order to downscale the data, we sample again just 100 points between $k_{min} = 5{\times}10^{-3} Mpc^{-1}$ and $k_{max} = 2 Mpc^{-1}$ enforcing an over-density in the Baryonic Acoustic Oscillations (BAO) wiggles as done in COMET (we use 100 samples instead of 106).

We decided to define k_{max} as a few times $k_{\rm NL}$ ([19], Equation (12)):

$$k_{\rm NL}^{-2}(z) = \frac{1}{12\pi^2}\int_0^\infty P_{\rm lin}(k,z)dk$$

At $z = 3$, $k_{\rm NL}(z = 3) \simeq 0.5$ Mpc^{-1}. So to be on the safe side, we decided to use $k_{\rm max} = 2$ Mpc^{-1}. The rationale for increasing this a bit is that we know that at $z = 0$ the EFT fits the data up to $k_{\rm max} \simeq 0.3$ Mpc^{-1} and that the non-linear scale at this redshift is $k_{\rm NL}(z = 0) \simeq 0.15$ Mpc^{-1}, so it makes sense to increase it a bit.

The cosmologies file will be structured as follows: each row will represent each one of the 700 k-bins components where each column will represent the different power spectra as generated with the sampled parameters. Thus the shape will be 700 x 50000. The former will be used as input for FFTLog and will output 13 values for each one of the components, resulting in a shape of 650,001 x 100: where the first row contains the k-bins as re-sampled to the $k_{\rm min}$ - $k_{\rm max}$ followed by 13 blocks of 50,000 vectors corresponding to each one of the correlators per cosmology. Finally, the ready-to-use dataset model will have a shape of 50,000 x 14 x 100. The extra value added to the second dimension, corresponds to the power spectra that were used to generate each one of the 50,000 cosmologies as we also will emulate them.

The naming conventions for the correlators are in line with the original EFT paper, namely:

- I correlators: I_{11}, I_{12}, I_{13}, I_{22}, I_{24}, I_{I33}, I_{34}, I_{44} and I_{55}.
- J correlators: J_1, J_2 and J_3.

To those we will add the denomination for the power spectra, $P_{\rm lin}$, that we will reuse later when discussing the modeling approach.

3.1.3 Training objective

The Gaussian Process model will be trained to fit the 13 integral correlators and the power spectra values at each one of the k-bins for each one of the samples or features (set of cosmological parameters). The end goal is that the user just inputs the amount of cosmologies to emulate, and the AI pipeline will randomly draw the three cosmological parameters (within the ranges defined in Section 3.1.1) to perform inference in order

to get the 14 predictions.
Note that the emulator has been trained with a fixed k-range of 100 values between $5\times10^{-3}Mpc^{-1}$ and 2 Mpc^{-1}, so we will provide also a splining method to be able to provide prediction at both different k-bins within the range and outside of it[19].

Finally, the optimal goal will be to have some pre-trained models with a reasonable size, that maintain the expected accuracy and inference speed. This models will be exposed via a user-friendly interface in which all the inner processes are abstracted.

3.1.4 Software considerations

All the power spectra are generated with the Python wrapper of CAMB [51].

The FFTLog computation will be performed by leveraging a native Mathematica [43] implementation.

The Gaussian Process model will be implemented in GPyTorch [33] using support libraries such as the Python implementation of KeOps [17] for large-scale GPU training and inference.

3.1.5 Hardware considerations

All the experiments were run in a mobile workstation with a six cores (5.10 GHz) Intel i7 CPU, 64GB of DDR4 RAM and an NVIDIA Quadro RTX 3000 GPU with 6GB of GDDR6 VRAM.

3.2 Cosmology generation

Using CAMB, two different functionalities were implemented, one to generate cosmologies using Latin Hypercube Sampling[20] over the cosmological parameter space and another one to generate them just based on already created parameters. This decision was made based on the fact that the correlators generation process needs a broader and finer grained range (700 k-bins) of values for the cosmologies (using the first method) but the power spectra for this same parameters need to be down-scaled for the emulator range (using the second method) of 100 k-bins. Although, for the latter splining could have been performed, it was decided to regenerate from scratch as the overhead was not excessive and accuracy was then ensured.

[19]Caveats: the inference in this case will not be as accurate as we depend on the quality of the spline and the user will be warned accordingly.
[20]See **Latin Hypercube Sampling** in the glossary.

All the parameters not mentioned in Section 3.1.1 are set to their default values. Along these values is worth to remark that the default transfer function is set to the Boltzmann one.

3.3 Correlators' generation

All the integration routine is decoupled from our Python's pipeline and leverages Mathematica [43] implementing the FFTLog algorithm [66]. The calculations are performed on a notebook which makes use of CPU multi-threading. The code is divided in two main parts:

1. Cosmology independent: computes some known fixed quantities for the (13)-type and (22)-type integrals used to calculate the loop matrices plus initializes the FFTLog module as per the original paper.
2. Cosmology dependent: first it pre-computes the FFT [14] of the power spectra over the provided k-bins range and then it actually performs the integrals' calculation to obtain the final correlators' values by means of matrix multiplications.

3.4 Preprocessing

All inputs to the model are processed such that they do not have any h dependencies, we do this by multiplying by h the k-values and diving by h^3 the actual correlator and power spectra values. Both of those have different re-scaling techniques that will ease the learning process of the models, the latter being the most straightforward one as it is just a logarithmic transformation[21]. For the correlators there are several steps involved:

1. Removal of negative values: as the final step of the preprocessing pipeline is to apply a logarithm, is a must that there are only positive values so that the transformation is defined. There are three different cases that need to be considered.
   - No change of sign: all correlators always positive, such as I_{11}, I_{24} or J_3[22].
   - Add a constant: if for some of the correlator values there is a positive/negative or negative/positive change (i.e., the correlator crosses zero at some point) we will add the minimum value (minus a small value, such as 0.1, to ensure a proper logarithmic transformation) across the whole range of correlator values as a constant to all of them. This happens for correlators such as I_{12}, I_{13} or I_{34}.

[21] Natural logarithm.

[22] $P_{\rm lin}$ is not mentioned here as this is a different process, but it is worth to note that is always positive too.

- Flip the sign: this happens when all the values for a correlator are negative. The correlators not mentioned in the previous two points undergo this transformation.

$$f(X) = \begin{cases} -X & \text{if } all(X) < 0 \\ X + (min(X) - 0.1) & \text{if } any(X) < 0 \\ X & \text{otherwise} \end{cases}$$

The negative scaling f takes as input X which is the whole set of values for a single correlator, namely, the 100 k-bins for each one of the cosmologies associated to it.

2. Normalization using the power spectra values: the ratio of the correlators by the corresponding power spectra value is taken. This is done since the correlators' amplitude can greatly vary depending on the parameters used for the generation so using ratios reduces the dynamical range of the quantities, making emulation easier.
   - I factors: are scaled by a smoothened fiducial power spectrum that is shared for all of them.
   - J factors: are scaled by the corresponding power spectra that were used to generate them.
3. Logarithmic transformation: for extra reduction of the dynamical range and improve numerical stability.

This whole process will look as follows in terms of equations:

$$I_{norm} = \ln\left(\frac{f(I)}{P_{\mathrm{lin}_{fiducial}}}\right)$$

$$J_{norm} = \ln\left(\frac{f(J_i)}{P_{\mathrm{lin}_i}}\right)$$

Finally, the last step taken is to normalize all data per input dimension (i.e., correlator and power spectrum). Again, this helps training performance and stability by making the features to be in the same scale. In our case, for each set of samples per dimension we make the data have zero mean and unit variance by applying the following:

$$X_{norm} = \frac{\bar{X}}{\sigma_X}$$

3.5 Sampling

For both the selection of the cosmological parameters and the split into train and test sets Latin Hypercube Sampling was used.

Additionally, since the output from CAMB returns the k-values in an evenly log-spaced and interpolated range between k_{min} and k_{max}, we need to capture the BAO wiggles in order to apply evolution mapping. To do this we take a different approach: we apply the algorithm used in COMET for sampling the k-bins with an over-density around the BAO wiggles for the power spectra which linearly interpolates given a minimum, a maximum, a mode or center in which to generate more samples and the strength or density to apply over the center. The most important value to capture this region of the spectrum is the mode, which in units of k/h is 0.22 and after applying the proper transformation with our value of h (0.695 in units of 100 $km\ s^{-1}\ Mpc^{-1}$) is 0.153.

3.6 Gaussian Process emulation

The approach chosen is that of exact regression using Gaussian Processes ([74], Algorithm (2.1)) implemented leveraging state-of-the-art libraries for reducing the computational complexity of inference from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. This overhead mostly comes for the matrix inversion needed for computing the GP posterior mean ($\bar{f}_*$) at a single point x_*:

$$\bar{f}_* = k_*^\top (K + \sigma_n^2 I)^{-1} y \tag{3.1}$$

Where $K = (X, X^*)$ is the covariance matrix (kernel) of n x n training points and $k_* = k(x_*) = K(X, x^*)$ is the test vector.

In the same way, the posterior variance can be calculated as follows:

$$\mathbb{V}[f_*] = k(x_*, x_*) - k_*^\top (K + \sigma_n^2 I)^{-1} k_*$$

This inversion is mostly performed using a Cholesky decomposition which works by splitting a given matrix A into the product of two Cholesky factors: $A = LL^\top$ being L the lower triangular matrix of A. Since the matrix is known to be symmetric and positive definite, its determinant can be calculated by:

$$|A| = \prod_{i=1}^{n} L_{ii}^2$$

It has been commonly attributed to be the method of choice for solving GP problems as it is extremely stable but it comes at the expense of run time. The whole block of

improvement comes from the GPyTorch library [33] which uses matrix-matrix multiplications which fully utilize GPU in its inference engine. They developed an algorithm called 'Blackbox Matrix-Matrix Multiplication' (BBMM) which uses a modified batched version of linear conjugate gradients (mBCG) inspired mainly by the so-called iterative Krylov subspace methods; the latter is based on matrix access by just using matrix-vector multiplies (MVM). The most important aspect of this is how they compute the conjugate gradients: they solve eigenvalues without explicitly computing a matrix, they follow an adapted version of the calculation of matrix estimators via matrix-vector products [29] without using the Lanczos tridiagonalization algorithm [77] but instead preconditioning with a pivoted Cholesky decomposition [5, 37] to accelerate the convergence of conjugate gradients without losing accuracy [59].

All in all, BBMM efficiently uses actual hardware (GPUs) to perform all of these large-scale matrix multiplications circumventing space complexity and stability problems whilst, and most importantly, reducing the time complexity of exact GP inference to $\mathcal{O}(n^2)$.

Finally, we also make use PyKeOps [17, 32] for support of symbolic matrices in order to speed up tensor programs. The main advantage over traditional dense (large, heavy memory load and time-consuming) arrays/tensors and even sparse (with few non-zero coefficients, thus less memory intensive but which do not stream well on GPU) is that they are represented by a function, $F(x_i, y_j)$, that given two vectors is able to compute the location (i, j) on an array $M_{i,j}$. This symbolic tensor representations are fast and memory-efficient providing up to 10x-100x speed-up in many GPU tensorial applications.

3.6.1 Model architecture

Much in line with COMET and as we are treating the correlators and power spectra outputs independently, we will construct 100 sub-models per output dimension (i.e., 100 models times 14 dimensions, for a total of 1400 models). Each one of these sub-models groupings is wrapped in an `IndependentModelList` (as per GPyTorch) which in turn optimizes all the sub-models at the same time. The construction of this model looks as follows:

Algorithm 1 Setup model list

Input: *x_train* ← *torch.tensor*: the cosmological parameters array
Input: *y_train* ← *torch.tensor*: the features for the specific correlator/power spectrum
Input: *correlator_index* ← *int*: the identifier of the dimension

1: *models* ← *list*
2: *likelihoods* ← *list*
3: *k_bins_size* ← 100
4: **for** $i \in range(k\_bins\_size)$ **do**
5: $\quad likelihoods.append(GaussianLikelihood)$
6: $\quad models.append(ExactGPModel(x\_train, y\_train[:, correlator\_index, i])$
7: *model* ← $IndependentModelList(*models)$
8: *likelihood* ← $LikelihoodList(*likelihoods)$
9: *loss* ← $SumMarginalLogLikelihood(likelihood, model)$

Output: *model, likelihood, loss*: the corresponding wrapper model with its likelihood and associated loss function for the correlator provided

Algorithm 1 takes as input the cosmological parameters array (*wc*, *wb*, *ns*) for the selected cosmologies which is shared for all the tasks, the vectors corresponding to the correlator at hand and the identifier for the same. In lines (7) and (8) the star operator means broadcasting, i.e., unpacking the list. This algorithm populates two lists for each specific *k*-bin with:

- Model: an exact GP model which based on the platform and the hardware available decides which kernel to use. In the case of Linux and a GPU available it will use the PyKeOps version of the kernel. In any case, it is an additive kernel, i.e., a kernel created by adding other two, which in this case are a Radial Basis Function and Matérn function.

$$k_{rbf}(x_1, x_2) = exp\Big(-\frac{1}{2}(x_1 - x_2)^\top \Theta^{-2}(x_1 - x_2)\Big)$$

  Where Θ is a length-scale parameter.

$$k_{matern}(x_1, x_2) = \frac{2^{1-\nu}}{\Gamma(\nu)}\big(\sqrt{2\nu d}\big)^\nu K_\nu\big(\sqrt{2\nu d}\big)$$

  Where $d = (x_1 - x_2)^\top \Theta^{-2}(x_1 - x_2)$ is the distance between the two inputs with a scale factor given by the length-scale parameter Θ. Γ is the gamma function[23], ν is a smoothing parameter in which higher values imply more smoothness (it usually takes values of 0.5, 1.5 or 2.5 and in our case, we use 2.5) and K_ν is a modified Bessel function[24].

[23] See **Gamma function** in the glossary.
[24] See **Modified Bessel function** in the glossary.

The (characteristic) length-scale parameter Θ, in plain words, represents how far you need to move on a particular axis in the input space so that the function values become uncorrelated. In our case we signal the kernels to learn a different value for each one of the input dimensions using the paradigm of Automatic Relevance Determination (ARD) [56]. ARD determines how relevant an input is, in other words, if the length-scale for a given dimension has a large value the covariance will be almost independent of that particular input, meaning that it will be removed from the inference process.

In addition, the model will use a `ConstantMean` which is not more than a non-zero constant prior mean function, $\mu(x) = C$, in which the value of C will be learned during training.

- Likelihood: the mapping from latent (predicted by the model) function values $f(x)$ to observed or true values y. It is a one-dimensional distribution of scalar functions, in other words, for a specific x we expect that $f(x) \in \mathbb{R}$. In our case, a `GaussianLikelihood` which translates to:

$$y(x) = f(x) + \epsilon$$

$$p(y|f) = f + \epsilon$$

Where ϵ is the Gaussian (standard homoskedastic) noise, i.e., it follows a Gaussian distribution $\epsilon \sim N(0, \sigma^2)$.

These two lists are then wrapped into the corresponding convenience classes which treat all the sub-models as a single one.

Additionally we define the loss function which in our case is based on computing the `ExactMarginalLogLikelihood` (evidence) for each one of the sub-models ($f \sim GP(\mu, K)$). The marginal likelihood is the integral of the likelihood times the prior:

$$p(y|X) = \int p(y|f(X))p(f(X)|X)df$$

And the marginal log likelihood:

$$\mathcal{L}_m = \log(p(y|X)) = -\frac{1}{2}f^\top K^{-1} f - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi) \tag{3.2}$$

Finally, the loss across the whole wrapper of models (`SumMarginalLogLikelihood`) is given by generalizing Equation (3.2):

$$\mathcal{L} = \frac{\sum_{i=0}^{n} \mathcal{L}_{m_i}}{n} \tag{3.3}$$

Where n is the number of losses corresponding to the amount of sub-models, in our case, 100.

3.6.2 Training

We can comprise the full training pipeline into one algorithm that acts upon the models, likelihoods and losses defined for each one of the correlators (plus power spectra) in the previous section.

Algorithm 2 Training loop

Input: $model \leftarrow IndependentModelList$: the model to optimize
Input: $likelihood \leftarrow LikeLihoodList$: the likelihood associated to the model
Input: $loss \leftarrow SumMarginalLogLikelihood$: the function to compute the loss
Input: $iterations \leftarrow int$: number of epochs to train the model for
Input: $\eta \leftarrow float$: the learning rate of the model

1: $optimizer \leftarrow AdamW(model.parameters(), \eta)$
2: $scheduler \leftarrow CosineAnnealingWarmRestarts(optimizer, T_0 = 10, T_{mult} = 1)$
3: $losses \leftarrow list$
4: **for** $i \in range(iterations)$ **do**
5: $\quad optimizer.zero\_grad()$
6: $\quad predictions \leftarrow model(*model.train\_inputs)$
7: $\quad$ **if** CG convergence fails **then**
8: $\quad\quad break$
9: $\quad loss \leftarrow -loss(predictions, model.train\_targets)$
10: $\quad losses.append(loss)$
11: $\quad loss.backward()$
12: $\quad optimizer.step()$
13: $\quad scheduler.step()$

Output: *model, likelihood, losses*: the trained model and likelihood with the corresponding losses for each epoch

There is a few considerations regarding Algorithm 2:

- In line (1) we define the optimizer to use during training. This object keeps track of the current state of the parameters of the model (e.g., length-scale) and is also in charge of update those based on the computed gradients. In this case we use an AdamW [52] optimizer which is in turn a modification of the commonly used optimizer for deep learning applications, Adam [48]. The latter is extensively used as it is extremely flexible (it can be applied to a lot of different optimization objectives, with noisy and/or sparse gradients), adapts really well to large-scale setups in terms of data and/or parameters, has favorable convergence properties when compared with classical stochastic optimization methods (such as Stochastic

Gradient Descent, SGD [62]) and, more importantly, while keeping computation and memory costs as low as possible. The decision to use AdamW as our optimizer comes from the inherent flexibility of Gaussian Processes which leads them to be prone to overfitting[25]. There are some techniques to counter this effect, such as as L_2 regularization and weight decay [36] in SGD, however the weight decay is not applicable in the same way for adaptive gradient optimizers such as Adam. In the case of AdamW, the process of decaying the weights of the parameters is decoupled from the loss-based gradient updates ([52], Algorithm 2, line (12)) and it has been experimentally shown to be effective both to prevent overfitting and increase generalization performance as a whole. The learning rate (η) in a model determines the size of the steps that an optimization algorithm takes when moving towards the minimum of a loss function; if it is too big, it might overshoot and converge too quickly in a non-optimal local minimum, whereas in the opposite case, it might cause the process to be too slow and get stuck in sub-optimal local minimum. So it is important to determine an adequate learning rate so that the optimizer converges to the best solution available: common values for deep learning algorithms are in the range 10^{-1} to 10^{-5} but these values are really a guideline since the learning rate is a really problem-dependent parameter. GPyTorch recommends a η of 10^{-1} for most problems solved with Adam, but in our case with AdamW we chose 5×10^{-2} to give a bit more of room for the optimization routine. Finally, we use the default value of 10^{-2} for the weight decay rate.

- In line (2) we instantiate a learning rate (η) scheduler. The decision to use a scheduler here comes from the optimizers' experiments which effectively shown that static learning rates tend to converge to worse solutions while taking more time than if an scheduler is used. This function controls how the learning behaves during the optimization process and in our case we follow the approach of step-based warm restarts[26], in other words, decay the value of the learning rate until a η_{min} and after some time reset it to its initial value of η_{max}. Our choice of algorithm is that of Cosine Annealing [53] (same as in AdamW's paper) and updates the learning rate as follows:

$$\eta_t = \eta^i_{min} + \frac{1}{2}(\eta^i_{max} - \eta^i_{min})\left(1 + cos\left(\frac{T_{current}}{T_i}\pi\right)\right)$$

  $T_{current}$ is updated at each iteration, when $T_i = T_{current} = 0$ it will set η_t to η^i_{max} and once $T_i = T_{current}$ the value of η_t will be η^i_{min} and both T's will be reset. In addition, the authors include a T_{mult} parameter that increases by a factor the

[25] This term is commonly used in AI terminology and refers to the situation in which a model learns too well the training distribution but it is not able to generalize to unseen (test) data. It is the opposite of underfitting, a situation in which the model lacks expressiveness at all and performs equally bad with train and test data.

[26] There are other approaches to scheduling such as time-based or exponential.

value of T_i at every restart. Since the specifics of this method are also problem dependent and we did not have time to test different combinations, we opted for conservative values of $T_i = 10$ and $T_{mult} = 1$ so after each 10 epochs, the learning rate will reach zero and restart to the initial value (5×10^{-2}).

- In line (6) a forward pass to the model is invoked by broadcasting the cosmological parameters to each one of the sub-models, this will compute the predicted values which are then fed to the loss function (Equation (3.3)) in line (9) to be compared with the ground-truth vectors.
- Line (11) takes care of computing the gradients of the loss function with respect to each one of the parameters. Then, the optimizer in line (12) takes a step in the opposite direction of the gradients updating accordingly the parameters of the model. At the same time, the scheduler in line (13) increases $T_{current}$ by one. Finally, is is worth pointing out that before each new iteration in line (5), the gradients need to be removed so they can be recomputed from scratch with the newly updated parameters.
- As an extra measure to prevent overfitting and unnecessary extra training time we introduce a check similar to that of early-stopping[27] in line (7). It terminates the training loop after the model signals that the conjugate gradients (CG) relative residual expected to terminate is larger than the defined value (in our case 0.5). In other words, when the model starts failing for convergence after some time it is because there is no more updates to do after having found the most optimal solution with the current parameters.

After each model has been trained, either because the convergence has been reached or because the error tolerance has not been reached[28], we save the parameters of both the model and the likelihood for further replication in the inference step.

Not less important, the train and test splits generated for optimizing and validating the models were of size 1750 and 2500 respectively. Usually in other deep learning approaches the number of samples tends to be way higher but due to the associated memory overhead in GPs this had to be kept conservative. However, by using PyKeOps this limit can be increased without impacting the runtime. Adding more samples would be beneficial so that the the models can capture finer-grained aspects of the distribution of cosmologies.

In Table 3.2 we show a summary of the main quantities involved during the training process.

[27] It is a process that stops training after the model fails to decrease the loss after a few iterations, i.e., the optimizer reached a minimum, either local or global.

[28] After the determined cycles/iterations the minimum value configured as tolerance for a successfully accurate model epoch, is not achieved, thus the model, starts failing to converge.

Parameter	Value
Training size	1750
Validation size	2500
η	0.05
Iterations	500
CG tolerance	0.5
T_0 (T_i)	10
T_{mult}	1

Table 3.2: Training parameters

It is also worth to note that the models learn normalized and scaled predictions as per Section 3.4. So in order to check the performance of the models, we need to apply the inverse process. Furthermore we define our own metric to measure quality:

$$Error = \left| \frac{corr_{nm_{pred}}}{corr_{nm_{true}}} - 1 \right| * 100 \tag{3.4}$$

Where we compute the absolute ratio of the predicted value by the true one for whatever correlator expressed in terms of a percentage. In our case the prediction that we will use is the mean value as drawn from the posterior distribution as given by Equation (3.1). Taking advantage of this, we can also calculate the median error at each of k-bins for each model using the mean values.

3.6.3 Inference

Once the models are trained, those are reconstructed by initializing each one of the sub-models for each correlator in a process similar to that of Algorithm 1 but this time using the dictionaries containing the trained hyperparameters such as the length-scale or the mean constant prior. The advantage of following this approach instead of using the whole model with the cached weights is that the memory footprint goes down from the order of a few GB to that of MB. This comes at the expense of losing the cached inference computations that speed up subsequent requests.

The main two functionalities exposed to the user in this pipeline are:

1. Single cosmology emulation: in which the user specifies the three cosmological parameters (*wb*, *wc* and *ns*) and gets the values for all the correlators and the corresponding power spectrum (which was in turn also predicted and used for re-scaling the models' predictions).

2. Multiple random parallel cosmology emulation: the user specifies how many cosmologies wants to generate and the framework taking advantage of its parallel inference capabilities quickly generates the predictions for a set of random and uniformly distributed cosmological parameters within the training ranges of the emulator (Table 3.1). This utility makes use of PyTorch's `DataLoader` which is in charge of batching the data into parallel streams so that the models can take advantage of parallel distribution.

In addition, the user can specify whether to spline the k-values to a set of values different to the specific 100 values for which the model was trained. For this, a univariate spline is used and can accurately map new k-bins to correlator vectors within the models' k_{min} - k_{max} range. Splining outside of this range is also allowed but the user will be warned about the potential inaccuracy of the process.

4 Results

4.1 Training convergence

It is worth to include a plot of the training loss to visualize how the scheduler affects the optimization process.

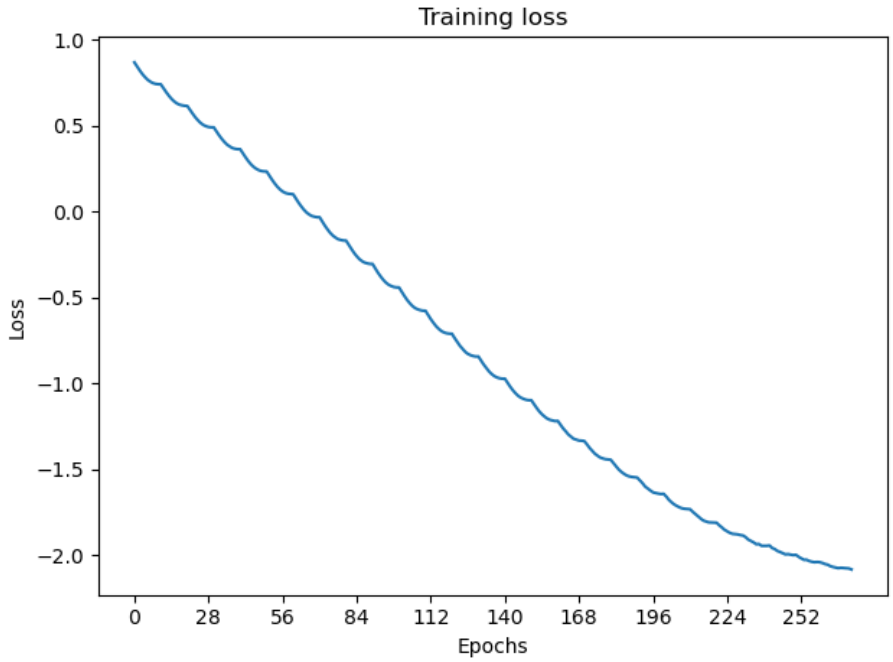

Figure 4.1: Loss over time for I_{11}

As it can be seen from the small bumps on the curve, the scheduler resets the learning rate each 10 iterations; when the learning rate reaches zero before a restart the loss increases a little bit before continuing towards the minimum. Furthermore, all models converged around an average value of -2 for the marginal log likelihood (Equation (3.3)).

4.2 Relative errors of training

As per Equation (3.4) the results are presented for a extract of the correlators and the power spectrum. Figures 4.2, 4.4, 4.6 and 4.8 include the results for the whole 2500

validation cosmologies[29].

Making it easier to visualize (due to, among others, presence of outliers) Figures 4.3, 4.5, 4.7 and 4.9 include the median absolute relative errors across all validation cosmologies for the corresponding correlators and the power spectrum in the above-mentioned figures. The shaded region corresponds to the models' confidence in the 84th percentile[30].

[29]For the complete list of figures refer to the Appendix B.1.
[30]For the complete list of figures refer to the Appendix B.2.

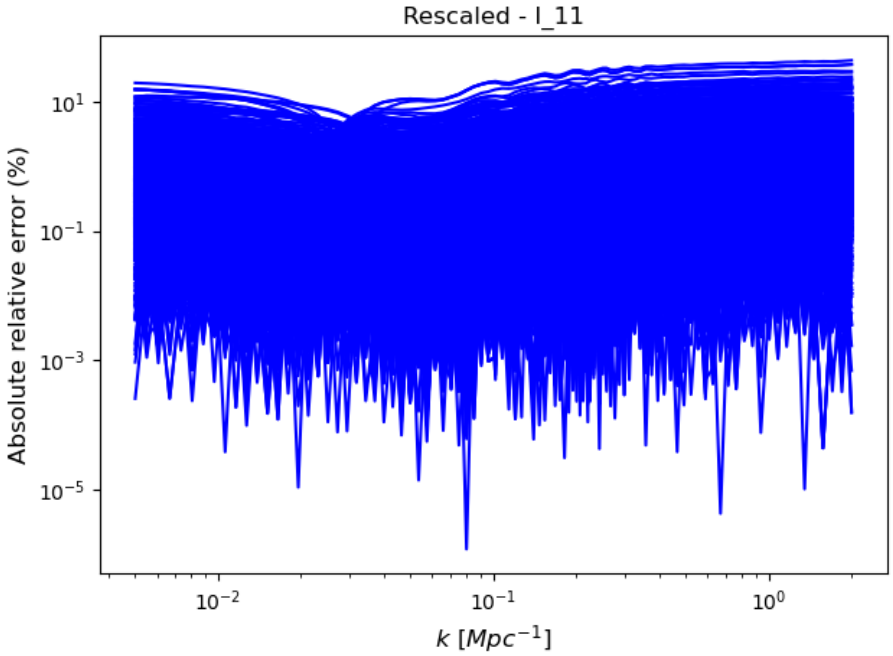

Figure 4.2: Absolute relative errors for I_{11}

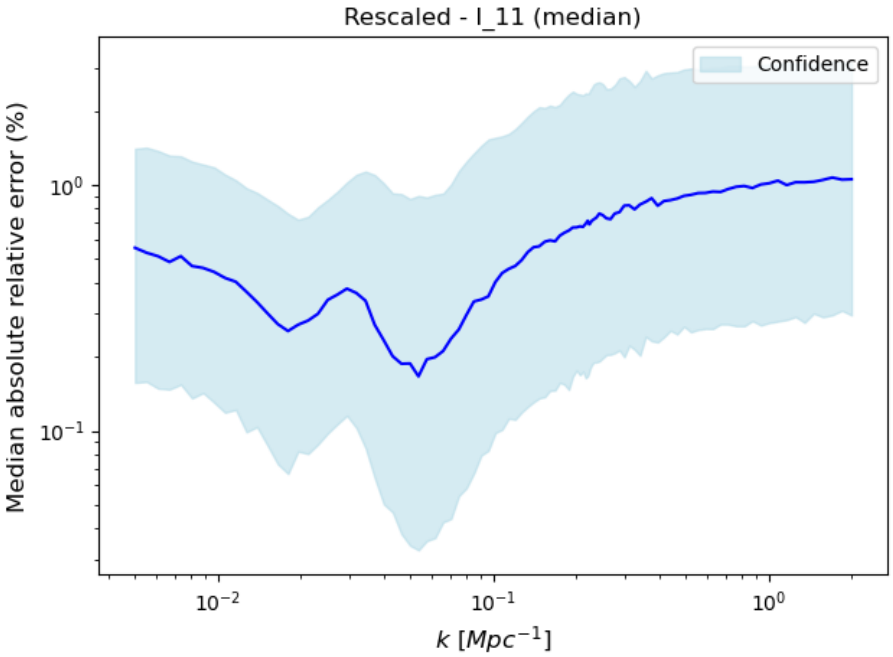

Figure 4.3: Median absolute relative error for I_{11}

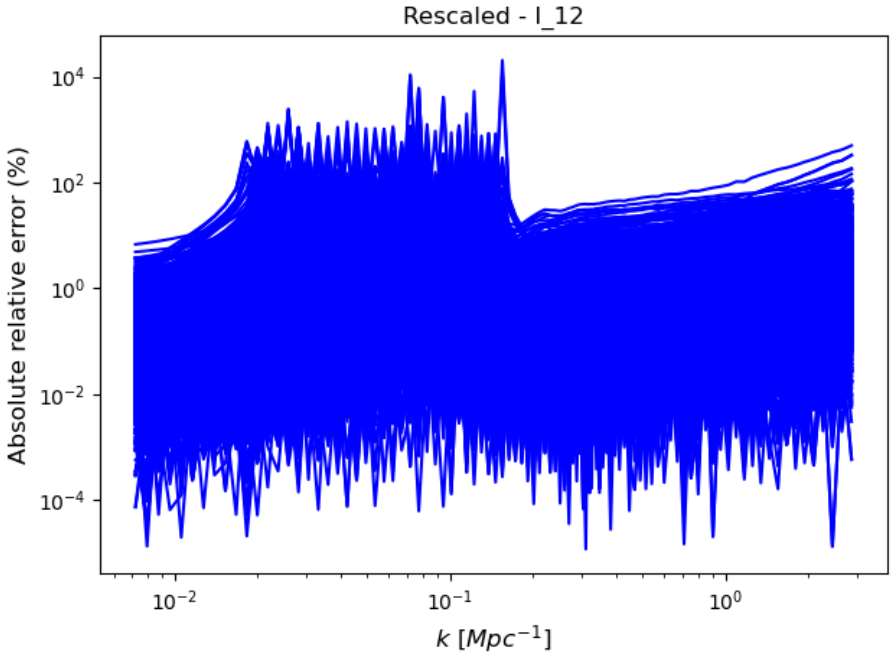

Figure 4.4: Absolute relative errors for I_{12}

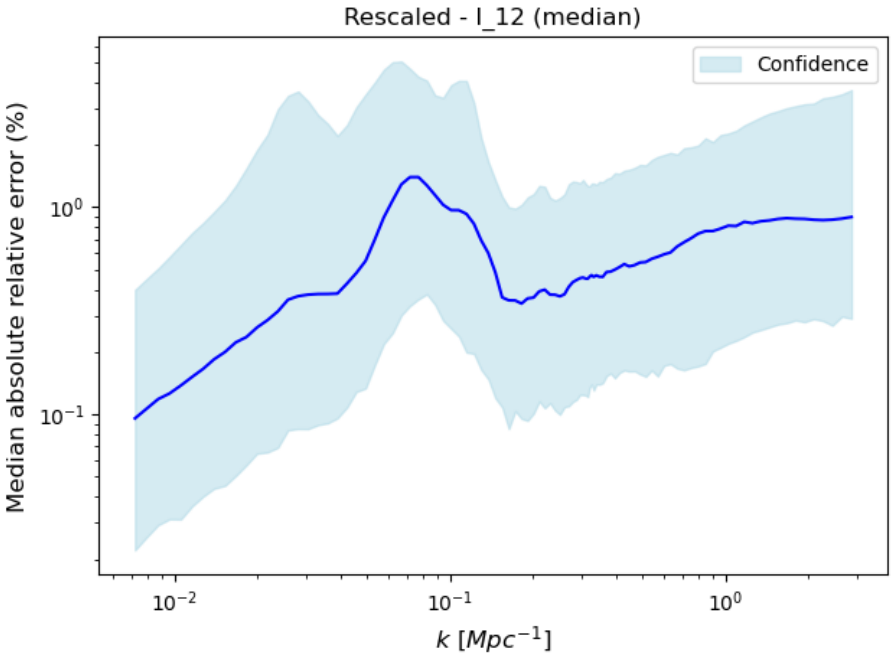

Figure 4.5: Median absolute relative error for I_{12}

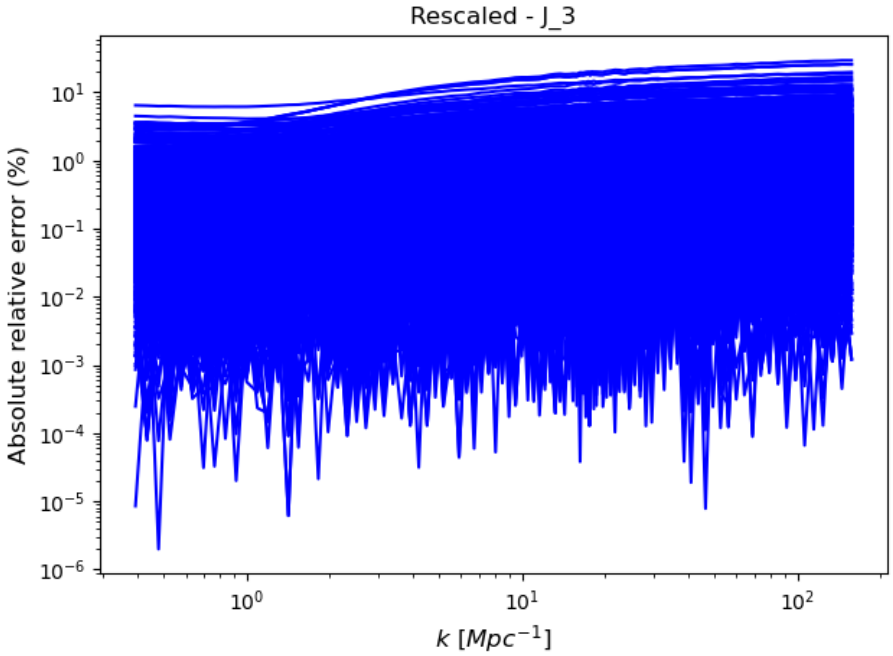

Figure 4.6: Absolute relative errors for J_3

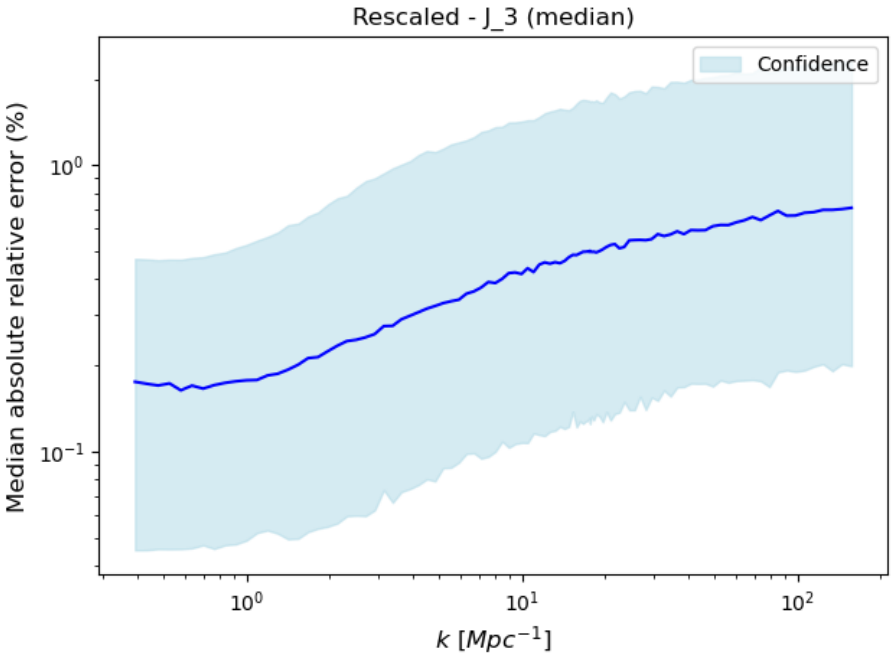

Figure 4.7: Median absolute relative error for J_3

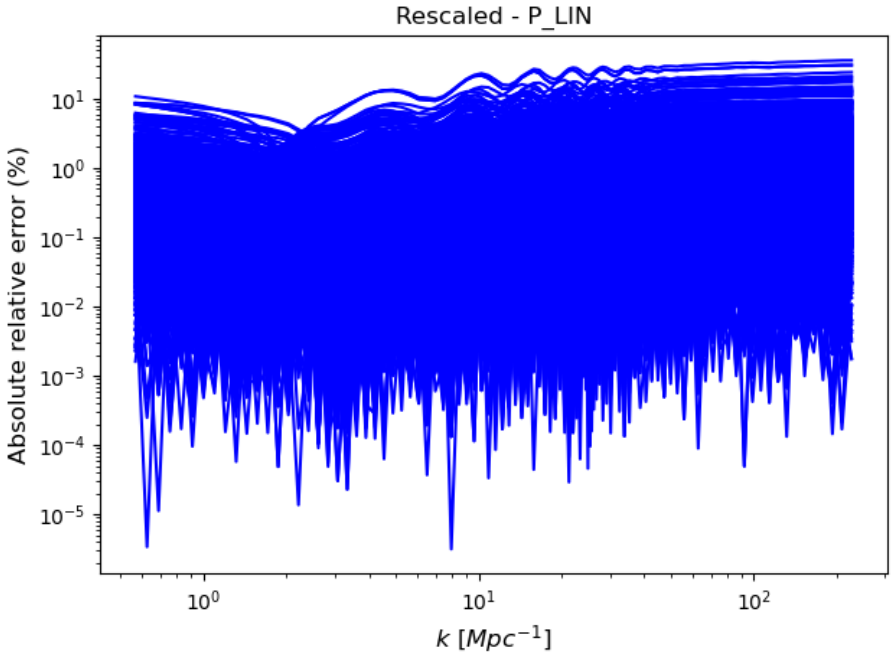

Figure 4.8: Absolute relative errors for P_{lin}

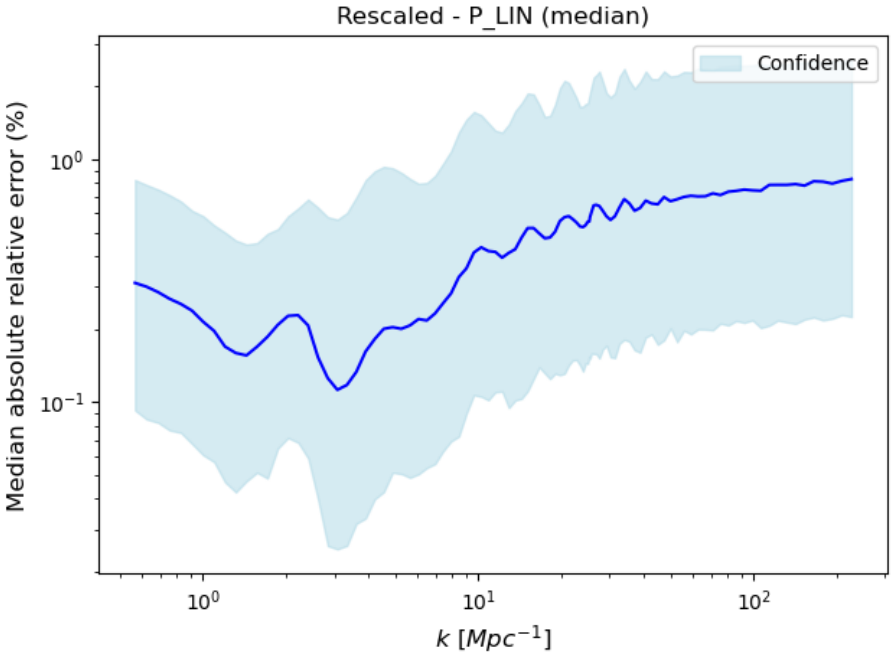

Figure 4.9: Median absolute relative error for P_{lin}

Firstly, it is worth to note that in this project we are not using as ground-truth values exact mathematical computations of the correlators' values but an approximation of those coming from FFTLog, thus, the maximum accuracy can only be as good as that

of the method. FFTLog states a percent and sub-percent level error depending on the k-bin range.

We can see that the absolute relative errors in general range from the percent level to several orders of sub-percent level (up to 10^{-4}) as it can be seen in Figures 4.2 and 4.6 for two of the correlators and in Figure 4.8 for the power spectra. However, there are some correlators that contain quite a few outliers that bump the errors to several orders of magnitude in the positive side (up to 10^3 - 10^4) as it can be seen in Figure 4.4. Over those regions, the models cannot make as accurate predictions.

However, it is easier to analyze the **errors'** outcome by looking at the **median** from those as the outliers are mitigated. We can see that we **always stay at the 10^{-1} error level** with a pretty decent confidence interval. For correlators that cross zero such as I_{12}, we can still see an increase to the percent level around those regions[31]. Interestingly, we can see that for all correlators and the power spectrum, **the error tends to increase as we also increase the k-bins**.

4.3 Outlier cosmologies

In Figures 4.10 and 4.11 we plot the correlation between the top 100 cosmologies with the most error (outliers) to see whether this anomalies are caused by any specific combination of parameters[32].

[31]This also happens for I_{13} (Figure B.11) and I_{34} (Figure B.16).
[32]For the complete list of figures refer to the Appendix B.3.

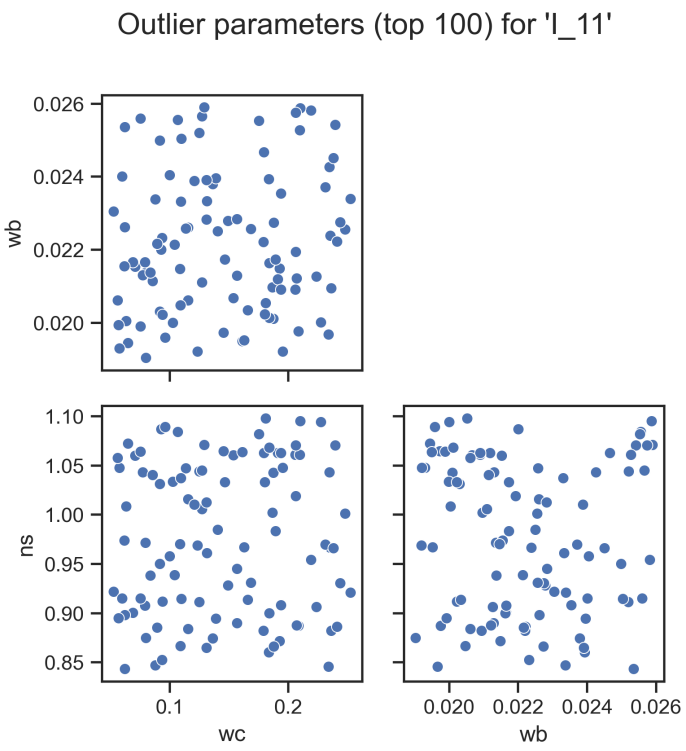

Figure 4.10: Correlations between the cosmological parameters for the outliers of I_{11}

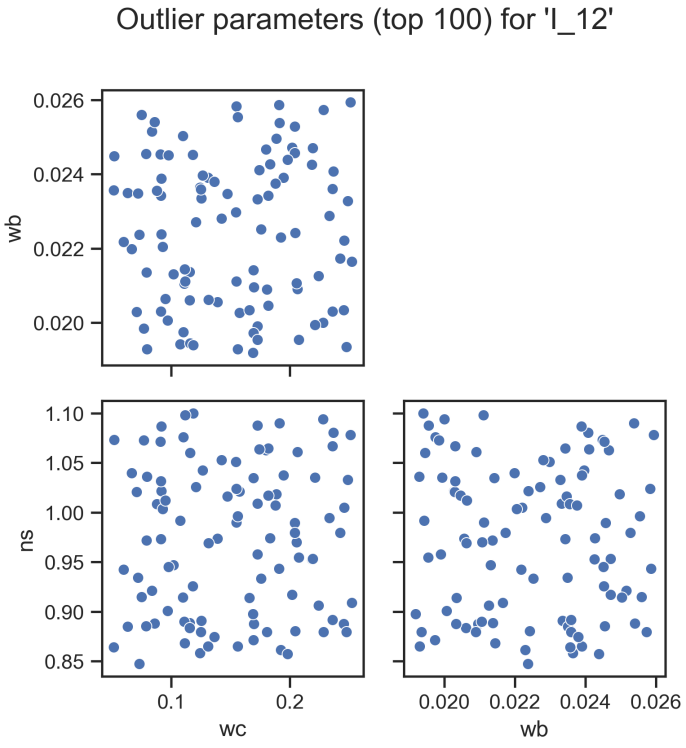

Figure 4.11: Correlations between the cosmological parameters for the outliers of I_{12}

Here we include two plots from two very different cosmologies, I_{11} which does not have extreme outliers (Figure 4.10) and I_{12} that contains a few of them (Figure 4.11) and as we can see, the distribution is pretty uniform so no conclusions can be extracted from this. We can conclude that outliers are not directly related to the parameters' combinations.

4.4 Timings

4.4.1 Cosmology generation

The generation of the cosmologies with CAMB was done in batches of 2500 cosmologies and took 3000 seconds on average. This equates to 1.2 seconds per power spectrum. Furthermore, if at some point in the future the model needs to be updated with more data (e.g., new cosmological parameters, different ranges...), new cosmologies can be generated fairly quick.

4.4.2 FFTLog execution

The generation of the correlators was done in batches of 2500 cosmologies and took 1500 seconds on average for the pre-computation of the cosmologies' shared values and 900 seconds for the actual integration. This roughly equates to a second per cosmology. This latter fact also helps in generating new extended or updated ground-truth data as mentioned at the end of the previous section.

4.4.3 Training pipeline

Each one of the models took an average of 2022.15 seconds for training, 22.57 seconds for caching and 266 ms for cached validation (1750 training and 2500 validation cosmologies, GPU, no PyKeOps).

4.4.4 Inference pipeline

Loading and setting up the cache for all models takes an average 20.28 seconds in all cases. Once loaded, the inference for the single cosmology case takes an average of 3.7 seconds taking into account model inference, predictions' saving and splining. On the other hand, the results for the random parallel generation are showed in Table 4.1.

Total cosmologies	Batch size	Runtime (s)	Runtime/cosmology (ms)
8192	2048	39	4.8
	4096	29	3.5
1,048,576	8192	3300	3.1
	65,536	2400	2.3

Table 4.1: Parallel inference times

The timings on the table, again, as the single cosmology case, consider the same extra processes aside from inference. However, a one time computation needs to be added, which involves the batching of the data and causes some overhead. The smaller case totals to roughly one second whereas the 10^6 case totals 120 seconds.

4.4.5 Time analysis

As a baseline we should take the timings for both generating the correlators with FFTLog (Section 4.4.2) and the power spectra with CAMB (Section 4.4.1), both of them being in the 10^3 ms average. We can see from 4.4.3 that each model takes around 266 ms per model for 2500 cosmologies, yet, if we look at the timings for a single cosmology prediction in Section 4.4.4 we see that the average run is 3.7 seconds which equates again to the above-mentioned value for each model. The main takeaway here is that the whole setup of this framework revolves around the idea of parallel inference and that is why the timings for just one cosmology versus multiple of them at the same time take the same amount of time. The explanation behind this is how hardware computations are performed on the GPU as it does not matter whether we fit just a single value or several: as long as they fit on the memory bus, they will be calculated in parallel and the timing will be the same.

From this **single cosmology** example we can still draw some insights, it is **slower than FTTLog for the correlators** as our approach takes 3.5 seconds for all the correlators instead of a second, however, it is **an order of magnitude decrease in runtime for the power spectrum** model.

Where the emulator shines is when it comes to **distributed inference**, from Table 4.1 we can see that the whole inference process is decreased in several orders of magnitude, taking just a **few ms on average per cosmology**: this includes the emulation of the 13 correlators plus the power spectrum. Some preliminary experiments show that increasing the size of the data that is passed in parallel to the models (bath size) decreases the runtime. This latter fact is extremely dependent of the hardware available as it is directly correlated with the amount of memory available.

5 Discussion

As mentioned in Section 2.3 there are a lot of different products with different emulation objectives and architectures. It is worth to highlight COMET as it is the emulator that our framework resembles the most: specifically, this emulator is able to achieve a single prediction runtime of 10ms using the GPy [34] library. However, our approach is the first of its kind and creates the baseline for intrinsic alignments studies under the EFT theory. We have created a fast-inference framework that is able to estimate the components of the EFT theory without sacrificing accuracy.

However, the main limitation of the emulator would come if it was to be used as part of a MCMC, since the single prediction speed is not too good, sequential execution for long chains would be not be efficient. However, there are some samplers that already implement distributed calls on the CPU [47] which could allow to increase the number of parallel emulations done with the models.

Furthermore, the framework is heavily focused on GPU for inference, so not having this hardware available would impact performance negatively[33].

[33]No further experiments were conducted to evaluate the CPU performance.

6 Conclusions and future work

6.1 Summary

We have created a novel emulator for the shape correlations of intrinsic alignments under the EFT theory by implementing a Python package for Gaussian Process emulation with GPU support using the GPyTorch library. We managed to accurately predict the EFT correlators along the power spectrum with relative errors in the sub-percent order. Sequential execution for a single cosmology takes a few seconds while in the parallel case this is reduced to the order of just milliseconds.

6.2 Code availability

The implementation was organized by using GIT version control and it can be found at https://git.science.uu.nl/uucosmo/ima-eftemulator[34].

The code will be publicly released upon publication of the research and the aim is to fully integrate it in CCL's [20] pipeline.

6.3 Future work

The majority of suggested improvements are mentioned in the repository. However, it is worth to remark some of them here:

1. Evaluate the feasibility and relative performance in terms of accuracy versus speed trade-off of Variational/Approximate Gaussian Processes. Some examples include Scalable Variational Gaussian Process (SVGP) [39], SVGP plus Contour Integral Quadrature (CIQ) [60], Spare Gaussian Process Regression (SGPR) [70] or Structured Kernel Interpolation (SKI/KISS) [76].
2. Same as in the previous point but for Neural Network approaches by using Multilayer Peceptrons (MLPs) or Generative Adversarial Networks (GANs). There has been some efforts in developing emulators using the former architecture [2, 3].
3. Increase the speed of the single cosmology prediction routine (as COMET [30] does using GPy) or integrate the parallel one into a MCMC distributed sampler.

[34]For requesting access contact the author at *i.montesalvarez@students.uu.nl*

4. Replace the Mathematica integration routine and efficiently implement it in Python so there is no need for an external intermediary outside of the package. An option for this would be to implement CUBA with Cuhre [35] using symbolic Python.

Bibliography

[1] Shankar Agarwal et al. "pkann – II. A non-linear matter power spectrum interpolator developed using artificial neural networks". In: *Monthly Notices of the Royal Astronomical Society* 439.2 (Feb. 2014), pp. 2102–2121. ISSN: 0035-8711. DOI: 10.1093/mnras/stu090. eprint: https://academic.oup.com/mnras/article-pdf/439/2/2102/18473172/stu090.pdf. URL: https://doi.org/10.1093/mnras/stu090.

[2] Raul E Angulo et al. "The BACCO simulation project: exploiting the full power of large-scale structure for cosmology". In: *Monthly Notices of the Royal Astronomical Society* 507.4 (July 2021), pp. 5869–5881. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2018. eprint: https://academic.oup.com/mnras/article-pdf/507/4/5869/40419371/stab2018.pdf. URL: https://doi.org/10.1093/mnras/stab2018.

[3] Giovanni Aricò et al. "The BACCO simulation project: a baryonification emulator with neural networks". In: *Monthly Notices of the Royal Astronomical Society* 506.3 (July 2021), pp. 4070–4082. DOI: 10.1093/mnras/stab1911. URL: https://doi.org/10.1093%5C%2Fmnras%5C%2Fstab1911.

[4] T. Auld et al. "Fast cosmological parameter estimation using neural networks". In: *Monthly Notices of the Royal Astronomical Society: Letters* 376.1 (Mar. 2007), pp. L11–L15. ISSN: 1745-3925. DOI: 10.1111/j.1745-3933.2006.00276.x. eprint: https://academic.oup.com/mnrasl/article-pdf/376/1/L11/2820739/376-1-L11.pdf. URL: https://doi.org/10.1111/j.1745-3933.2006.00276.x.

[5] Francis Bach. *Sharp analysis of low-rank kernel matrix approximations*. 2013. arXiv: 1208.2015 [cs.LG].

[6] Daniel Baumann et al. "Cosmological non-linearities as an effective fluid". In: *Journal of Cosmology and Astroparticle Physics* 2012.07 (July 2012), pp. 051–051. DOI: 10.1088/1475-7516/2012/07/051. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2012%5C%2F07%5C%2F051.

[7] Steven V. W. Beckwith et al. "The Hubble Ultra Deep Field". In: *The Astronomical Journal* 132.5 (Sept. 2006), p. 1729. DOI: 10.1086/507302. URL: https://dx.doi.org/10.1086/507302.

[8] Andreas A. Berlind et al. "The Halo Occupation Distribution and the Physics of Galaxy Formation". In: *The Astrophysical Journal* 593.1 (July 2003), p. 1. DOI: 10.1086/376517. URL: https://dx.doi.org/10.1086/376517.

[9] Michael R. Blanton et al. "Sloan Digital Sky Survey IV: Mapping the Milky Way, Nearby Galaxies, and the Distant Universe". In: *Astronomic Journal* 154.1, 28 (July 2017), p. 28. DOI: 10.3847/1538-3881/aa7567. arXiv: 1703.00052 [astro-ph.GA].

[10] Jonathan Blazek, Matthew McQuinn, and Uroš Seljak. "Testing the tidal alignment model of galaxy intrinsic alignment". In: *Journal of Cosmology and Astroparticle Physics* 2011.05 (May 2011), pp. 010–010. DOI: 10.1088/1475-7516/2011/05/010. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2011%5C%2F05%5C%2F010.

[11] Jonathan Blazek, Zvonimir Vlah, and Uroš Seljak. "Tidal alignment of galaxies". In: *Journal of Cosmology and Astroparticle Physics* 2015.08 (Aug. 2015), pp. 015–015. DOI: 10.1088/1475-7516/2015/08/015. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2015%5C%2F08%5C%2F015.

[12] Jonathan Blazek et al. "Separating intrinsic alignment and galaxy-galaxy lensing". In: *Journal of Cosmology and Astroparticle Physics* 2012.05 (May 2012), pp. 041–041. DOI: 10.1088/1475-7516/2012/05/041. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2012%5C%2F05%5C%2F041.

[13] Jonathan A. Blazek et al. "Beyond linear galaxy alignments". In: *Physical Review D* 100.10 (Nov. 2019). DOI: 10.1103/physrevd.100.103506. URL: https://doi.org/10.1103%5C%2Fphysrevd.100.103506.

[14] E. O. Brigham and R. E. Morrow. "The fast Fourier transform". In: *IEEE Spectrum* 4.12 (1967), pp. 63–70. DOI: 10.1109/MSPEC.1967.5217220.

[15] M. L. Brown et al. "Measurement of intrinsic alignments in galaxy ellipticities". In: *Monthly Notices of the Royal Astronomical Society* 333.3 (July 2002), pp. 501–509. DOI: 10.1046/j.1365-8711.2002.05354.x. URL: https://doi.org/10.1046%5C%2Fj.1365-8711.2002.05354.x.

[16] P. Catelan, M. Kamionkowski, and R. D. Blandford. "Intrinsic and extrinsic galaxy alignment". In: *Monthly Notices of the Royal Astronomical Society* 320.1 (Jan. 2001), pp. L7–L13. DOI: 10.1046/j.1365-8711.2001.04105.x. URL: https://doi.org/10.1046%5C%2Fj.1365-8711.2001.04105.x.

[17] Benjamin Charlier et al. "Kernel Operations on the GPU, with Autodiff, without Memory Overflows". In: *Journal of Machine Learning Research* 22.74 (2021), pp. 1–6. URL: http://jmlr.org/papers/v22/20-275.html.

[18] Shi-Fan Chen et al. "Redshift-space distortions in Lagrangian perturbation theory". In: *Journal of Cosmology and Astroparticle Physics* 2021.03 (Mar. 2021), p. 100. DOI: 10.1088/1475-7516/2021/03/100. URL: https://dx.doi.org/10.1088/1475-7516/2021/03/100.

[19] Nora Elisa Chisari and Andrew Pontzen. "Unequal time correlators and the Zel'dovich approximation". In: *Phys. Rev. D* 100 (2 July 2019), p. 023543. DOI: 10.1103/PhysRevD.100.023543. URL: https://link.aps.org/doi/10.1103/PhysRevD.100.023543.

[20] Nora Elisa Chisari et al. "Core Cosmology Library: Precision Cosmological Predictions for LSST". In: *The American Astronomical Society* 242.1, 2 (May 2019), p. 2. DOI: 10.3847/1538-4365/ab1658. arXiv: 1812.05995 [astro-ph.CO]. URL: https://github.com/LSSTDESC/CCL.

[21] Nora Elisa Chisari et al. "Intrinsic alignments of group and cluster galaxies in photometric surveys". In: *Monthly Notices of the Royal Astronomical Society* 445.1 (Sept. 2014), pp. 726–748. DOI: 10.1093/mnras/stu1786. URL: https://doi.org/10.1093%5C%2Fmnras%5C%2Fstu1786.

[22] Anton Chudaykin et al. "Nonlinear perturbation theory extension of the Boltzmann code CLASS". In: *Phys. Rev. D* 102 (6 Sept. 2020), p. 063533. DOI: 10.1103/PhysRevD.102.063533. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.063533.

[23] Robert G. Crittenden et al. "Discriminating Weak Lensing from Intrinsic Spin Correlations Using the Curl-Gradient Decomposition". In: *The Astrophysical Journal* 568.1 (Mar. 2002), pp. 20–27. DOI: 10.1086/338838. URL: https://doi.org/10.1086%5C%2F338838.

[24] Robert G. Crittenden et al. "Spin-induced Galaxy Alignments and Their Implications for Weak-Lensing Measurements". In: *The Astrophysical Journal* 559.2 (Oct. 2001), p. 552. DOI: 10.1086/322370. URL: https://dx.doi.org/10.1086/322370.

[25] Kyle S. Dawson et al. "The Baryon Oscillation Spectroscopic Survey of SDSS-III". In: *Astronomic Journal* 145.1, 10 (Jan. 2013), p. 10. DOI: 10.1088/0004-6256/145/1/10. arXiv: 1208.0022 [astro-ph.CO].

[26] Joseph DeRose et al. "Neural network acceleration of large-scale structure theory calculations". In: *Journal of Cosmology and Astroparticle Physics* 2022.04 (Apr. 2022), p. 056. DOI: 10.1088/1475-7516/2022/04/056. URL: https://dx.doi.org/10.1088/1475-7516/2022/04/056.

[27] Nathalie Deruelle and Jean-Philippe Uzan. "605The Lambda-CDM model of the hot Big Bang". In: *Relativity in Modern Physics*. Oxford University Press, Aug. 2018. ISBN: 9780198786399. DOI: 10.1093/oso/9780198786399.003.0059. eprint: https://academic.oup.com/book/0/chapter/369328301/chapter-pdf/45462909/oso-9780198786399-chapter-59.pdf. URL: https://doi.org/10.1093/oso/9780198786399.003.0059.

[28] Jamie Donald-McCann, Kazuya Koyama, and Florian Beutler. "matryoshka II: accelerating effective field theory analyses of the galaxy power spectrum". In: *Monthly Notices of the Royal Astronomical Society* 518.2 (Jan. 2023), pp. 3106–3115. DOI: 10.1093/mnras/stac3326. arXiv: 2202.07557 [astro-ph.CO].

[29] Kun Dong et al. *Scalable Log Determinants for Gaussian Process Kernel Learning*. 2017. arXiv: 1711.03481 [stat.ML].

[30] Alexander Eggemeier et al. "COMET: Clustering observables modelled by emulated perturbation theory". In: *Monthly Notices of the Royal Astronomical Society* 519.2 (Dec. 2022), pp. 2962–2980. ISSN: 0035-8711. DOI: 10.1093/mnras/stac3667. eprint: https://academic.oup.com/mnras/article-pdf/519/2/2962/48521235/stac3667.pdf. URL: https://doi.org/10.1093/mnras/stac3667.

[31] Euclid Collaboration et al. "Euclid preparation - I. The Euclid Wide Survey". In: *A&A* 662 (2022), A112. DOI: 10.1051/0004-6361/202141938. URL: https://doi.org/10.1051/0004-6361/202141938.

[32] Jean Feydy et al. "Fast geometric learning with symbolic matrices". In: *Advances in Neural Information Processing Systems* 33 (2020).

[33] Jacob R. Gardner et al. "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 7587–7597.

[34] GPy. *GPy: A Gaussian process framework in python*. http://github.com/SheffieldML/GPy. since 2012.

[35] T. Hahn. "Cuba—a library for multidimensional numerical integration". In: *Computer Physics Communications* 168.2 (2005), pp. 78–95. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2005.01.010. URL: https://www.sciencedirect.com/science/article/pii/S0010465505000792.

[36] Stephen Hanson and Lorien Pratt. "Comparing Biases for Minimal Network Construction with Back-Propagation". In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky. Vol. 1. Morgan-Kaufmann, 1988. URL: https://proceedings.neurips.cc/paper_files/paper/1988/file/1c9ac0159c94d8d0cbedc973445af2da-Paper.pdf.

[37] Helmut Harbrecht, Michael Peters, and Reinhold Schneider. "On the low-rank approximation by the pivoted Cholesky decomposition". In: *Applied Numerical Mathematics* 62.4 (2012). Third Chilean Workshop on Numerical Analysis of Partial Differential Equations (WONAPDE 2010), pp. 428–440. ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2011.10.001. URL: https://www.sciencedirect.com/science/article/pii/S0168927411001814.

[38] Katrin Heitmann et al. "The Mira-Titan Universe: Precision Predictions for Dark Energy Surveys". In: *The Astrophysical Journal* 820.2 (Mar. 2016), p. 108. DOI: 10.3847/0004-637X/820/2/108. URL: https://dx.doi.org/10.3847/0004-637X/820/2/108.

[39] James Hensman, Alex Matthews, and Zoubin Ghahramani. *Scalable Variational Gaussian Process Classification*. 2014. arXiv: 1411.2005 [stat.ML].

[40] Heymans, Catherine et al. "KiDS-1000 Cosmology: Multi-probe weak gravitational lensing and spectroscopic galaxy clustering constraints". In: *A&A* 646 (2021), A140. DOI: 10.1051/0004-6361/202039063. URL: https://doi.org/10.1051/0004-6361/202039063.

[41] H. Hildebrandt et al. "KiDS-450: cosmological parameter constraints from tomographic weak gravitational lensing". In: *Monthly Notices of the Royal Astronomical Society* 465.2 (Feb. 2017), pp. 1454–1498. DOI: 10.1093/mnras/stw2805. arXiv: 1606.05338 [astro-ph.CO].

[42] Edwin Hubble. "A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae". In: *Proceedings of the National Academy of Science* 15.3 (Mar. 1929), pp. 168–173. DOI: 10.1073/pnas.15.3.168.

[43] Wolfram Research Inc. *Mathematica, Version 13.2*. Champaign, IL, 2022. URL: https://www.wolfram.com/mathematica.

[44] Željko Ivezić et al. "LSST: From Science Drivers to Reference Design and Anticipated Data Products". In: *The Astrophysical Journal* 873.2 (Mar. 2019), p. 111. DOI: 10.3847/1538-4357/ab042c. URL: https://dx.doi.org/10.3847/1538-4357/ab042c.

[45] Benjamin Joachimi et al. "Galaxy Alignments: An Overview". In: *Space Science Reviews* 193.1-4 (Nov. 2015), pp. 1–65. DOI: 10.1007/s11214-015-0177-4. arXiv: 1504.05456 [astro-ph.GA].

[46] Joachimi, B. et al. "Constraints on intrinsic alignment contamination of weak lensing surveys using the MegaZ-LRG sample". In: *A&A* 527 (2011), A26. DOI: 10.1051/0004-6361/201015621. URL: https://doi.org/10.1051/0004-6361/201015621.

[47] Minas Karamanis, Florian Beutler, and John A Peacock. "zeus: A Python implementation of Ensemble Slice Sampling for efficient Bayesian parameter inference". In: *arXiv preprint arXiv:2105.03468* (2021).

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

[49] Juliana Kwan et al. "Cosmic Emulation: Fast Predictions for the Galaxy Power Spectrum". In: *Astronomic Journal* 810.1, 35 (Sept. 2015), p. 35. DOI: 10.1088/0004-637X/810/1/35. arXiv: 1311.6444 [astro-ph.CO].

[50] Earl Lawrence et al. "The Mira-Titan Universe. II. Matter Power Spectrum Emulation". In: *The Astrophysical Journal* 847.1 (Sept. 2017), p. 50. DOI: 10.3847/1538-4357/aa86a9. URL: https://doi.org/10.3847%5C%2F1538-4357%5C%2Faa86a9.

[51] Antony Lewis and Anthony Challinor. *CAMB: Code for Anisotropies in the Microwave Background*. Astrophysics Source Code Library, record ascl:1102.026. Feb. 2011. ascl: 1102.026.

[52] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[53] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017. arXiv: 1608.03983 [cs.LG].

[54] A J Mead et al. "HMcode-2020: Improved modelling of non-linear cosmological power spectra with baryonic feedback". In: *Monthly Notices of the Royal Astronomical Society* 502.1 (Jan. 2021), pp. 1401–1422. DOI: 10.1093/mnras/stab082. URL: https://doi.org/10.1093%5C%2Fmnras%5C%2Fstab082.

[55] Kelly R Moran et al. "The Mira–Titan Universe – IV. High-precision power spectrum emulation". In: *Monthly Notices of the Royal Astronomical Society* 520.3 (Dec. 2022), pp. 3443–3458. ISSN: 0035-8711. DOI: 10.1093/mnras/stac3452. eprint: https://academic.oup.com/mnras/article-pdf/520/3/3443/49228233/stac3452.pdf. URL: https://doi.org/10.1093/mnras/stac3452.

[56] Radford M Neal. *Bayesian Learning for Neural Networks*. 1996.

[57] Planck Collaboration et al. "Planck 2015 results - XIII. Cosmological parameters". In: *A&A* 594 (2016), A13. DOI: 10.1051/0004-6361/201525830. URL: https://doi.org/10.1051/0004-6361/201525830.

[58] Planck Collaboration et al. "Planck 2018 results - VI. Cosmological parameters". In: *A&A* 641 (2020), A6. DOI: 10.1051/0004-6361/201833910. URL: https://doi.org/10.1051/0004-6361/201833910.

[59] Geoff Pleiss et al. *Constant-Time Predictive Distributions for Gaussian Processes*. 2018. arXiv: 1803.06058 [cs.LG].

[60] Geoff Pleiss et al. *Fast Matrix Square Roots with Applications to Gaussian Processes and Bayesian Optimization*. 2020. arXiv: 2006.11267 [cs.LG].

[61] Jason Rhodes et al. "Scientific Synergy between LSST and Euclid". In: *The Astrophysical Journal Supplement Series* 233.2 (Dec. 2017), p. 21. DOI: 10.3847/1538-4365/aa96b0. URL: https://dx.doi.org/10.3847/1538-4365/aa96b0.

[62] Herbert E. Robbins. "A Stochastic Approximation Method". In: *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.

[63] Ariel G Sánchez et al. "Evolution mapping: a new approach to describe matter clustering in the non-linear regime". In: *Monthly Notices of the Royal Astronomical Society* 514.4 (June 2022), pp. 5673–5685. ISSN: 0035-8711. DOI: 10.1093/mnras/stac1656. eprint: https://academic.oup.com/mnras/article-pdf/514/4/5673/44829949/stac1656.pdf. URL: https://doi.org/10.1093/mnras/stac1656.

[64] Ariel G. Sánchez. "Arguments against using h^{-1} Mpc units in observational cosmology". In: *Phys. Rev. D* 102 (12 Dec. 2020), p. 123511. DOI: 10.1103/PhysRevD.102.123511. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.123511.

[65] D.M. Schmitz et al. "Time evolution of intrinsic alignments of galaxies". In: *Journal of Cosmology and Astroparticle Physics* 2018.07 (July 2018), pp. 030–030. DOI: 10.1088/1475-7516/2018/07/030. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2018%5C%2F07%5C%2F030.

[66] Marko Simonović et al. "Cosmological perturbation theory using the FFTLog: formalism and connection to QFT loop integrals". In: *Journal of Cosmology and Astroparticle Physics* 2018.04 (Apr. 2018), pp. 030–030. DOI: 10.1088/1475-7516/2018/04/030. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2018%5C%2F04%5C%2F030.

[67] Sukhdeep Singh, Rachel Mandelbaum, and Surhud More. "Intrinsic alignments of SDSS-III BOSS LOWZ sample galaxies". In: *Monthly Notices of the Royal Astronomical Society* 450.2 (May 2015), pp. 2195–2216. ISSN: 0035-8711. DOI: 10.1093/mnras/stv778. eprint: https://academic.oup.com/mnras/article-pdf/450/2/2195/3086153/stv778.pdf. URL: https://doi.org/10.1093/mnras/stv778.

[68] Alessio Spurio Mancini et al. "CosmoPower: emulating cosmological power spectra for accelerated Bayesian inference from next-generation surveys". In: *Monthly Notices of the Royal Astronomical Society* 511.2 (Jan. 2022), pp. 1771–1788. ISSN: 0035-8711. DOI: 10.1093/mnras/stac064. eprint: https://academic.oup.com/mnras/article-pdf/511/2/1771/42440282/stac064.pdf. URL: https://doi.org/10.1093/mnras/stac064.

[69] Max Tegmark et al. "The Three-Dimensional Power Spectrum of Galaxies from the Sloan Digital Sky Survey". In: *The Astrophysical Journal* 606.2 (May 2004), pp. 702–740. DOI: 10.1086/382125. URL: https://doi.org/10.1086%5C%2F382125.

[70] Michalis Titsias. "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 16–18 Apr 2009, pp. 567–574. URL: https://proceedings.mlr.press/v5/titsias09a.html.

[71] Zvonimir Vlah, Nora Elisa Chisari, and Fabian Schmidt. "An EFT description of galaxy intrinsic alignments". In: *Journal of Cosmology and Astroparticle Physics* 2020.01 (Jan. 2020), p. 025. DOI: 10.1088/1475-7516/2020/01/025. URL: https://dx.doi.org/10.1088/1475-7516/2020/01/025.

[72] Zvonimir Vlah, Nora Elisa Chisari, and Fabian Schmidt. "An EFT description of galaxy intrinsic alignments". In: *Journal of Cosmology and Astroparticle Physics* 2020.01 (Jan. 2020), p. 025. DOI: 10.1088/1475-7516/2020/01/025. URL: https://dx.doi.org/10.1088/1475-7516/2020/01/025.

[73] Zvonimir Vlah, Nora Elisa Chisari, and Fabian Schmidt. "Galaxy shape statistics in the effective field theory". In: *Journal of Cosmology and Astroparticle Physics* 2021.05 (Apr. 2021), p. 061. DOI: 10.1088/1475-7516/2021/05/061. URL: https://dx.doi.org/10.1088/1475-7516/2021/05/061.

[74] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[75] Robert E. Williams et al. "The Hubble Deep Field: Observations, Data Reduction, and Galaxy Photometry". In: *Astronomical Journal* 112 (Oct. 1996), p. 1335. DOI: 10.1086/118105. arXiv: astro-ph/9607174 [astro-ph].

[76] Andrew Gordon Wilson and Hannes Nickisch. *Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP)*. 2015. arXiv: 1503.01057 [cs.LG].

[77] Guanghan Xu and Thomas Kailath. "Fast Estimation of Principal Eigenspace Using Lanczos Algorithm". In: *SIAM Journal on Matrix Analysis and Applications* 15.3 (1994), pp. 974–994. DOI: 10.1137/S0895479890183848. eprint: https://doi.org/10.1137/S0895479890183848. URL: https://doi.org/10.1137/S0895479890183848.

[78] Zheng Zheng et al. "Theoretical Models of the Halo Occupation Distribution: Separating Central and Satellite Galaxies". In: *The Astrophysical Journal* 633.2 (Nov. 2005), pp. 791–809. DOI: 10.1086/466510. arXiv: astro-ph/0408564 [astro-ph].

A Glossary

Arc-second Angular measurement equal to $\frac{1}{3600}$ of a degree.

Astronomical unit Abbreviated as AU, is the distance between the Earth and the Sun and equivalent to approximately 1.5×10^8 kilometers.

Cosmic web How galaxies organize to form the large-scale structure of the Universe.

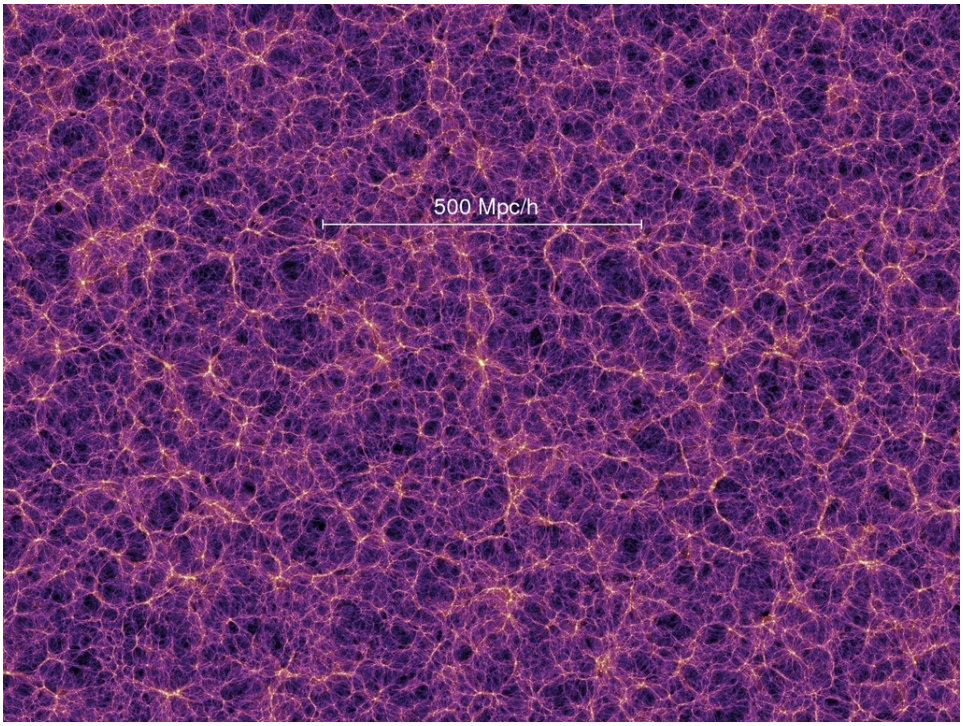

Figure A.1: A patch of 2 billion light-years on a side representing the galaxy distribution (measured at $z = 0$) from the Millennium Simulation Project. The purple color represent dark matter voids whereas the yellow colors represent galaxies which tend to cluster together due to gravity forming clumps and running towards filaments surrounded by the aforementioned voids.

Source: https://wwwmpa.mpa-garching.mpg.de/galform/virgo/millennium/

Dark matter halo Basic unit of cosmological structure which contains gravitationally bounded (baryonic) matter. A single halo, can contain multiple subhaloes. Models such as ΛCDM propose that this haloes can contain galaxies. Despite not being directly observable, it is thought that they exist because their interaction with the galaxies embedded in them, causing effect such as gravitational lensing or motion changes (e.g., angular momentum).

Doppler effect Or Doppler shift, is the change in frequency of a wave with respect of an observer who is moving relative to the source (Figure A.2). It is given by the following equation:

$$f_{obs} = \frac{f_e}{1 - \frac{v_e}{c}} \rightarrow f_{obs} = f_e \sqrt{\frac{1 - \frac{v_e}{c}}{1 + \frac{v_e}{c}}}$$

Where f_{obs} and f_e correspond to the observed and emitted frequencies respectively, v_e is the velocity of the source and c is the propagation speed of the waves in the medium. The above-mentioned equation can be equivalently expressed in terms of wavelength (λ):

$$\lambda_{obs} = \lambda_e \sqrt{\frac{1 + \frac{v_e}{c}}{1 - \frac{v_e}{c}}} \tag{A.1}$$

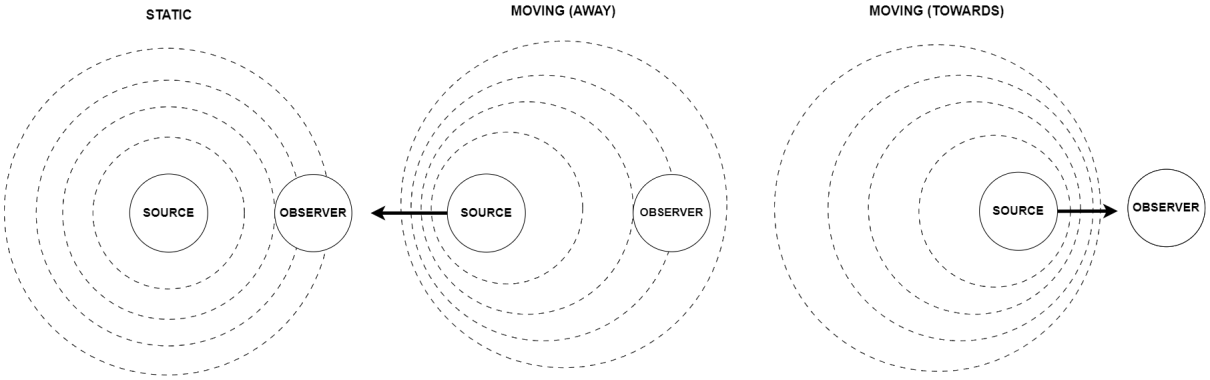

Figure A.2: Schematic representation of the Doppler effect as perceived by an observer if the source is static or moving (away/towards) from the observer's reference system. For the static case: $\lambda_{obs} = \lambda_e$, for the away: $\lambda_{obs} > \lambda_e$ and for the towards: $\lambda_{obs} < \lambda_e$.

Feynman diagram Pictorial representation of those mathematical expressions that describe the behavior and interaction of subatomic particles. They are used to model the perturbative contributions to a field. They are graphs that consist of points (vertices) and lines attached to those (edges). There are several interpretations and applications that one can make out of such a representation, one of them corresponds to the loop order. In the scope of our research, we are interested in one-loop diagrams which are no more that connected unicyclic diagrams; one of the most common examples is the "triangle diagram" (Figure A.3).

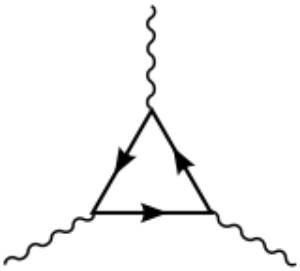

Figure A.3: An example of a one-loop Feynman diagram.

The general approach to compute this diagrams is by using loop integrals.

Gamma function Denoted by $\Gamma(n)$ is the generalization of the factorial function to complex numbers. Is defined for all positive integers as per $\Gamma(n) = (n-1)!$ and for all complex numbers (except the negative integers) as:

$$\Gamma(n) = \int_0^\infty t^{z-1} e^{-t} dt$$

Latin Hypercube Sampling Or LHS, is a quasi-random sampling method that produces samples that depict the underlying distribution of the data more accurately that a classic random sampling method. This enables to use smaller sample sizes as the data is evenly distributed across all dimensions. To sample a distribution of N variables, the range of each of them is divided into M (needed as input) equally probable intervals. Each sample taken is placed on the only one possible axis-aligned hyperplane containing it (Figure A.4).

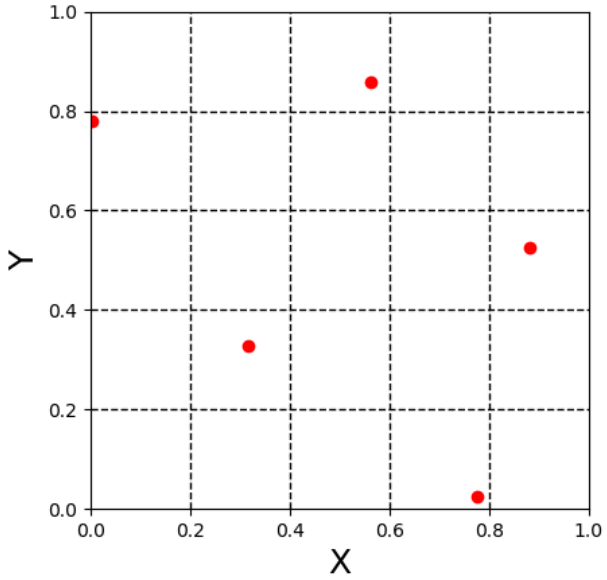

Figure A.4: Schematic representation of how samples (red dots) are taken for $N = 2$ and $M = 5$ where we can see that the distribution space has been divided into 5x5 equally probable regions. Note that none of the samples share the same hyper-plane, i.e., for this specific 2D example, no row-column intersection.

Light-year Distance that the light travels in one year, approximately 9.46×10^{12} kilometres or 0.3 *pc*.

Loop integral Integral used to evaluate Feynman diagrams with one or more loops and integrated over the internal momenta. Usually computationally intensive to calculated, specially for loop values greater than one.

Modified Bessel function Canonical solution of Bessel's differential equation which in this case is valid for complex arguments (even for a purely imaginary argument). It is defined as follows:

$$K_\alpha(x) = \frac{\pi}{2} \frac{I_{-\alpha}(x) - I_\alpha(x)}{sin(\alpha\pi)}$$

Where I_α is defined as:

$$I_\alpha(x) = \sum_{m=0}^{\infty} \frac{1}{m!\Gamma(m+\alpha+1)} (\frac{x}{2})^{2m+\alpha}$$

Parallax It is the observed displacement of an object caused by the change of the observers' system of reference. The distance between two objects can be calculated

using trigonometry (Figure A.5). This method cannot be applied to extremely distant objects since the parallax angle would be too small, this meaning that cannot be applied in our cosmological study.

$$tan(\alpha) = \frac{1AU}{d} \rightarrow d = \frac{1AU}{tan(\alpha)} \tag{A.2}$$

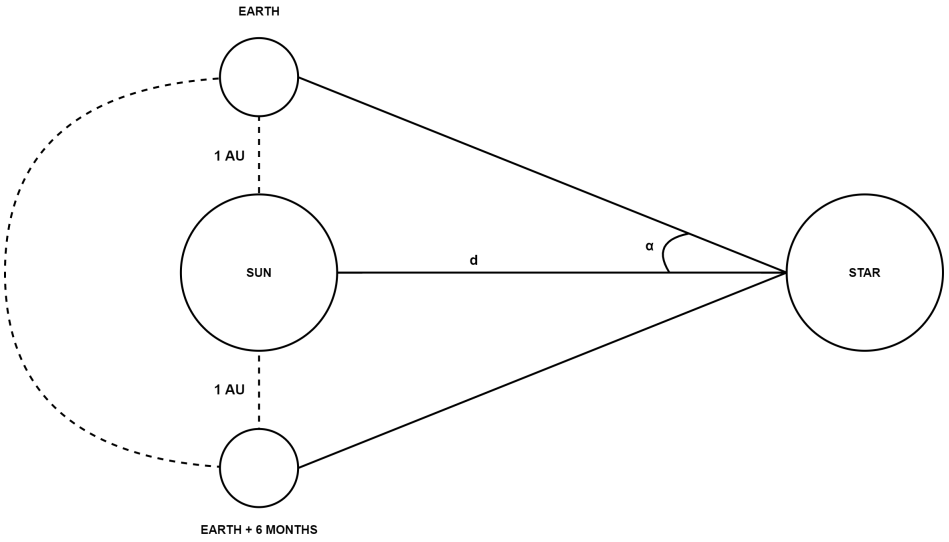

Figure A.5: Schematic representation on how to measure stellar parallax (from Earth).

Parsec Abbreviated as pc, Distance to an object whose parallax angle (α) is one arcsecond (1″). It is equivalent to $2.05 \times 10^5 AU$ or 3.26 light-years. In equation (A.2):

$$d = \frac{1AU}{1''} = 1pc$$

Redshift Abbreviated as z, is an application of the Doppler effect to cosmology, in which the waves are light (thus the propagation speed c will be equal to the speed of light, approximately 3×10^5 kilometers per second) and the sources are astronomical objects (Figure A.6). It relies on the decomposition of light spectrum, meaning that sources that are moving away from us will have their spectra shifted to the red (and vice versa to the blue). The magnitude of the velocity of the source is proportional to the shift (for small velocities when compared to that of the light):

$$1 + z = \frac{\lambda_{obs} - \lambda_e}{\lambda_e}$$

And applying (A.1) we get the proportionality:

$$1 + z = \sqrt{\frac{1 + \frac{v_e}{c}}{1 - \frac{v_e}{c}}} \rightarrow v_e \approx cz$$

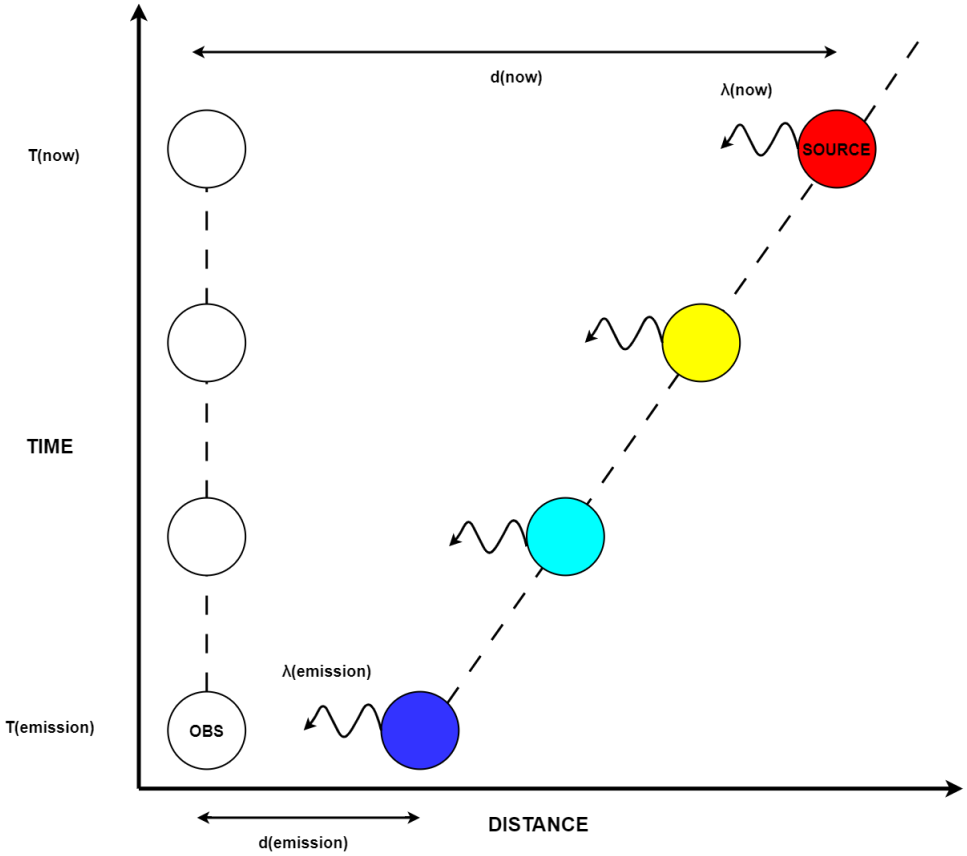

Figure A.6: Schematic representation of the redshift effect. When a source emits a wave of light with a given wavelength, $\lambda(emission)$, it travels through spacetime until it reaches us at $T(now)$. The received wave, $\lambda(now)$, has been shifted towards the red as the source is moving away from us. The strength or degree of this change in amplitude of the wavelength is given by the redshift (z) which increases as the velocity of the source does. Note that $\lambda(now) > \lambda(emission)$.

B Extra results

B.1 Relative errors of training

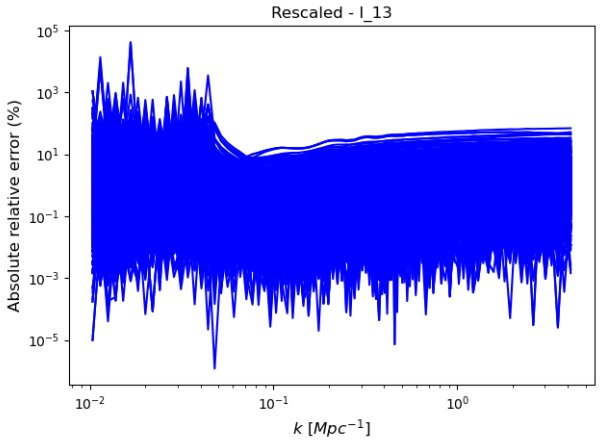

Figure B.1: Absolute relative errors for I_{13}

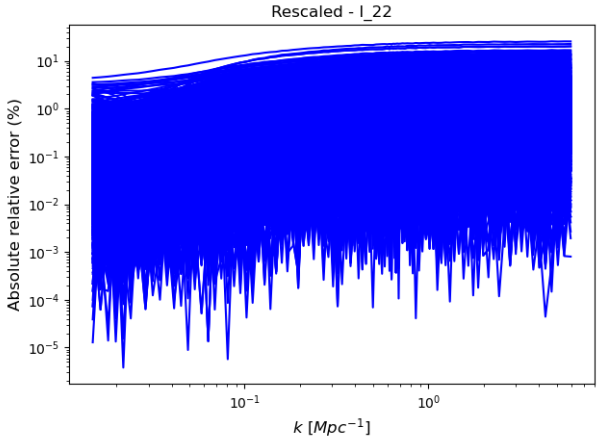

Figure B.2: Absolute relative errors for I_{22}

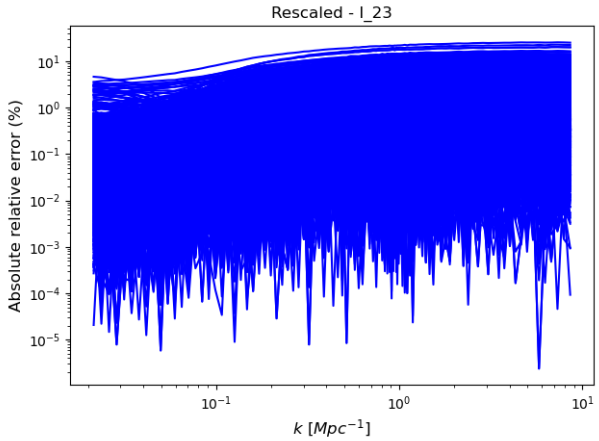

Figure B.3: Absolute relative errors for I_{23}

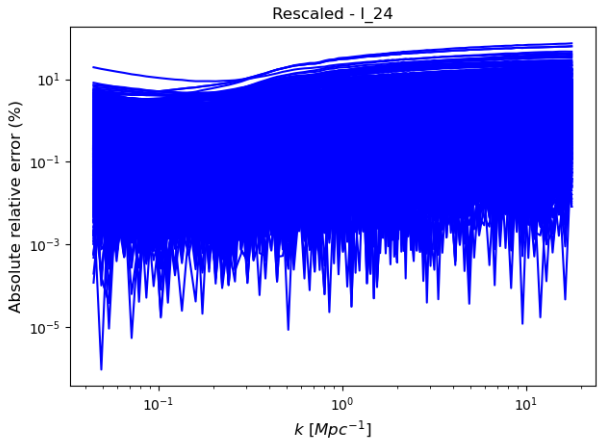

Figure B.4: Absolute relative errors for I_{24}

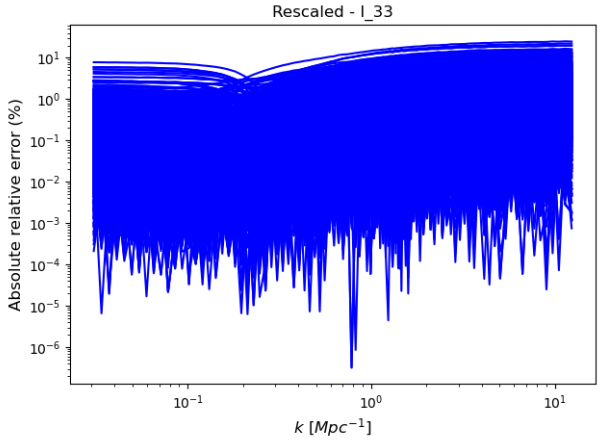

Figure B.5: Absolute relative errors for I_{33}

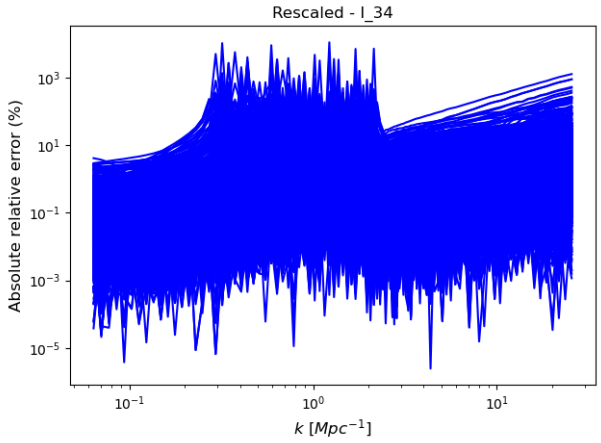

Figure B.6: Absolute relative errors for I_{34}

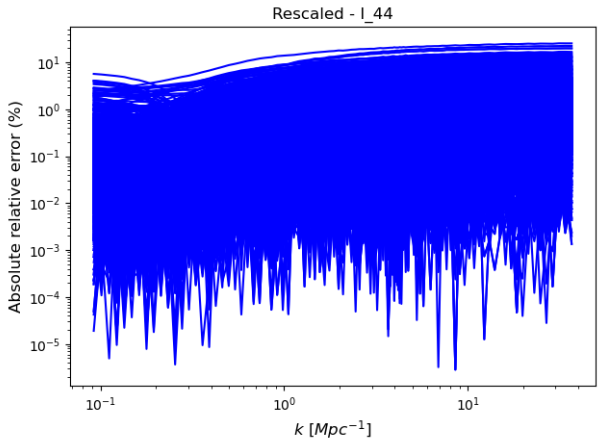

Figure B.7: Absolute relative errors for I_{44}

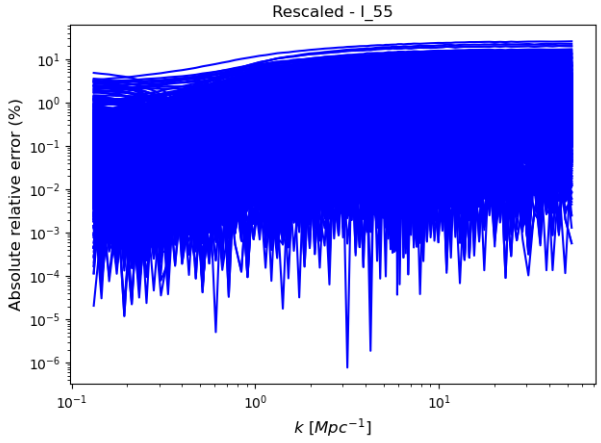

Figure B.8: Absolute relative errors for I_{55}

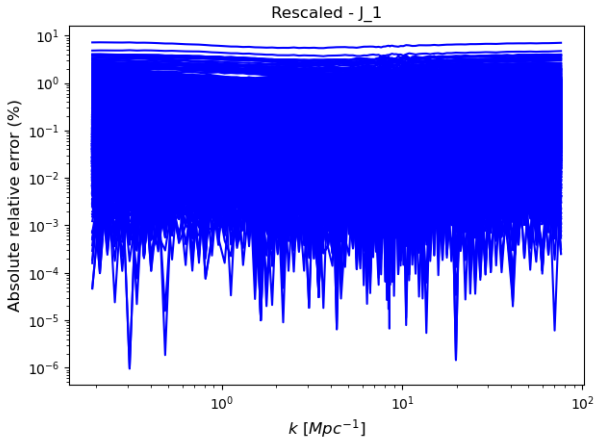

Figure B.9: Absolute relative errors for J_1

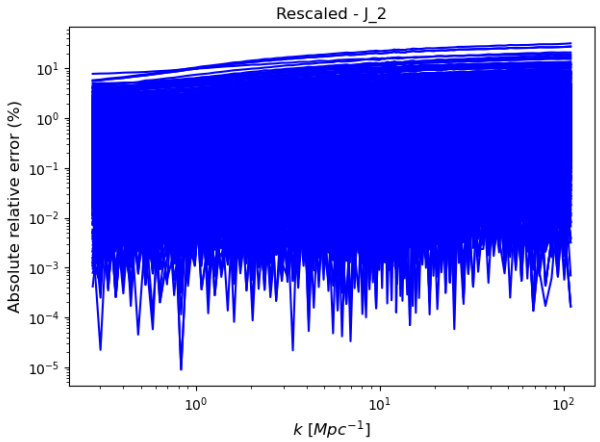

Figure B.10: Absolute relative errors for J_2

B.2 Median relative errors of training

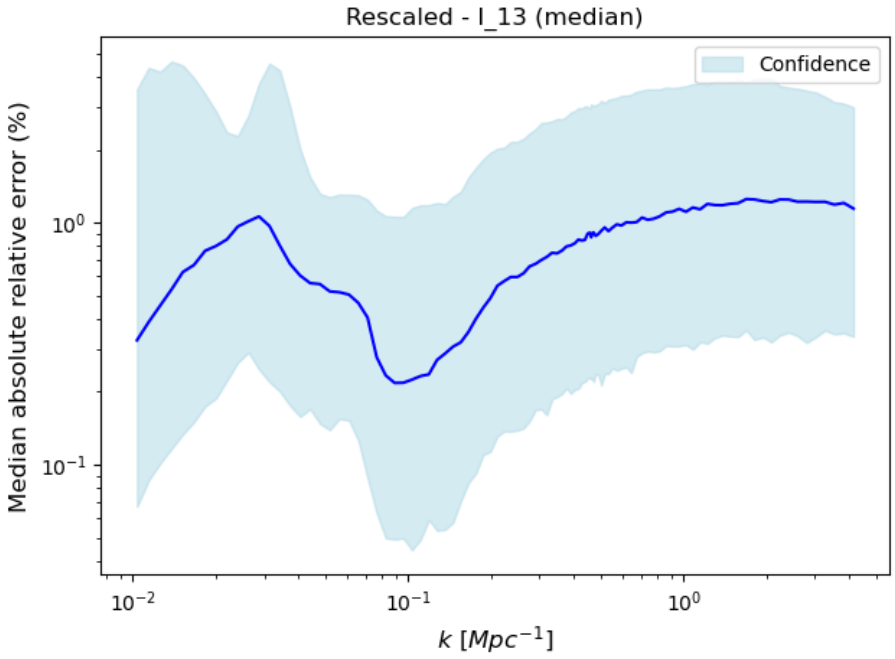

Figure B.11: Median absolute relative error for I_{13}

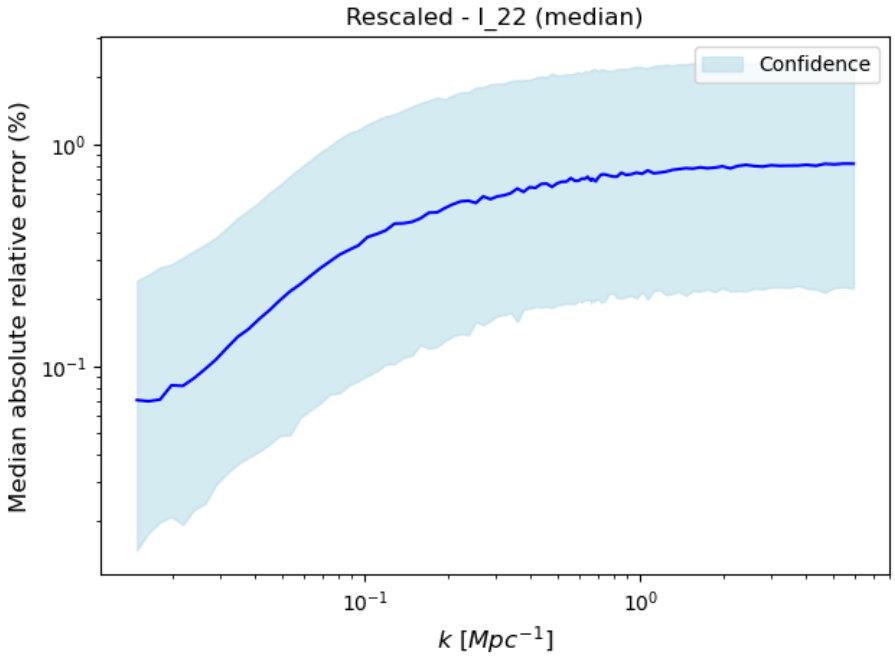

Figure B.12: Median absolute relative error for I_{22}

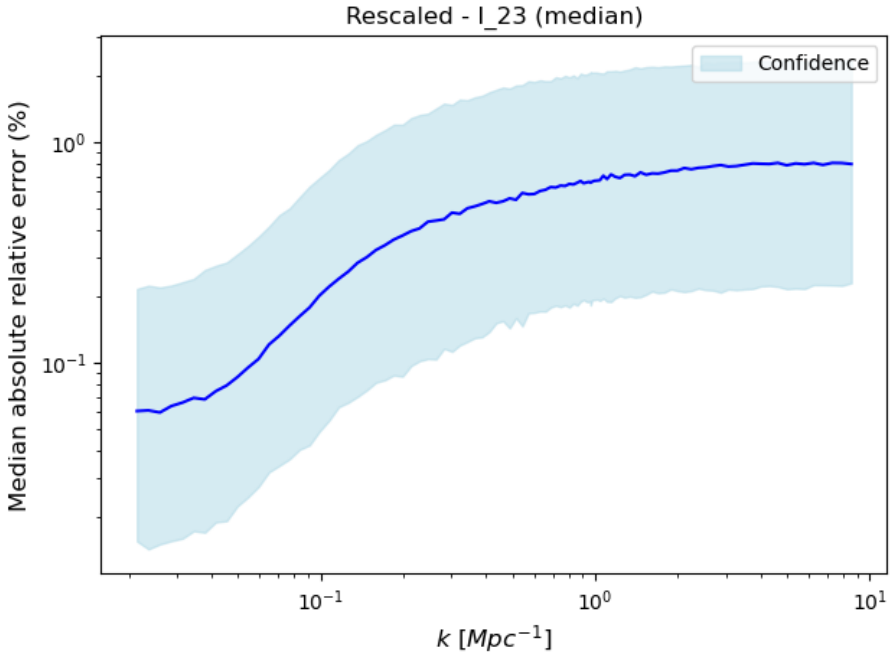

Figure B.13: Median absolute relative error for I_{23}

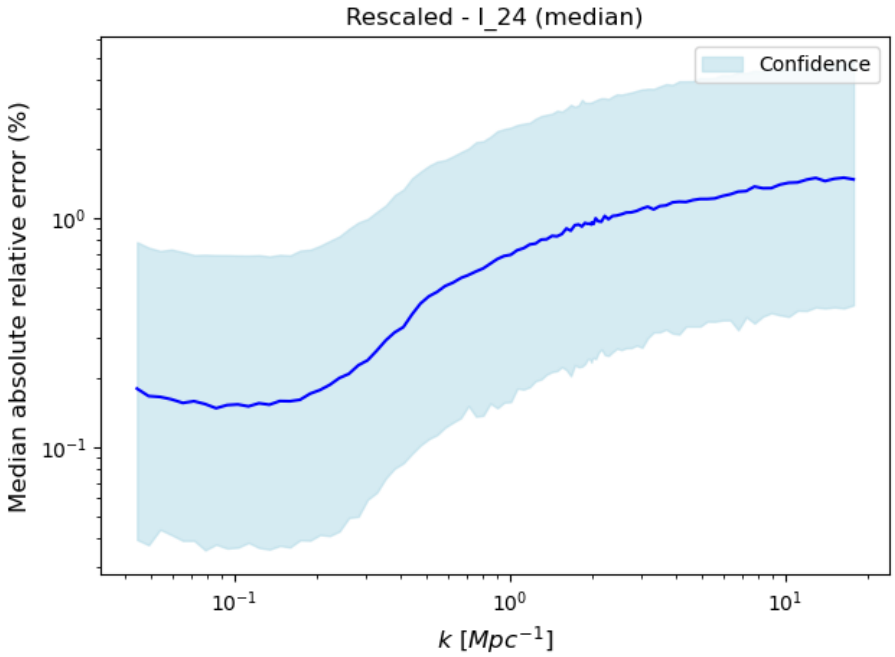

Figure B.14: Median absolute relative error for I_{24}

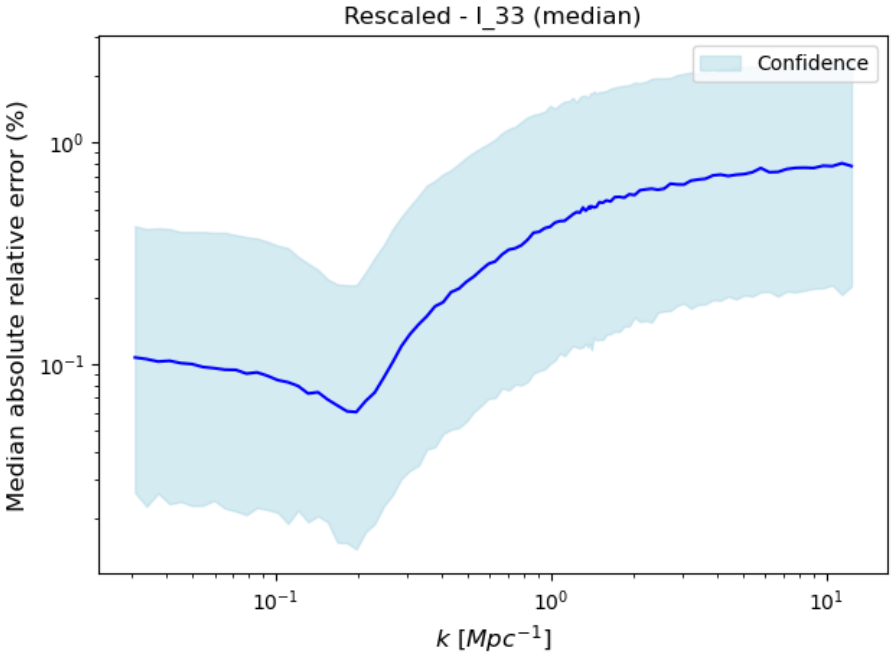

Figure B.15: Median absolute relative error for I_{33}

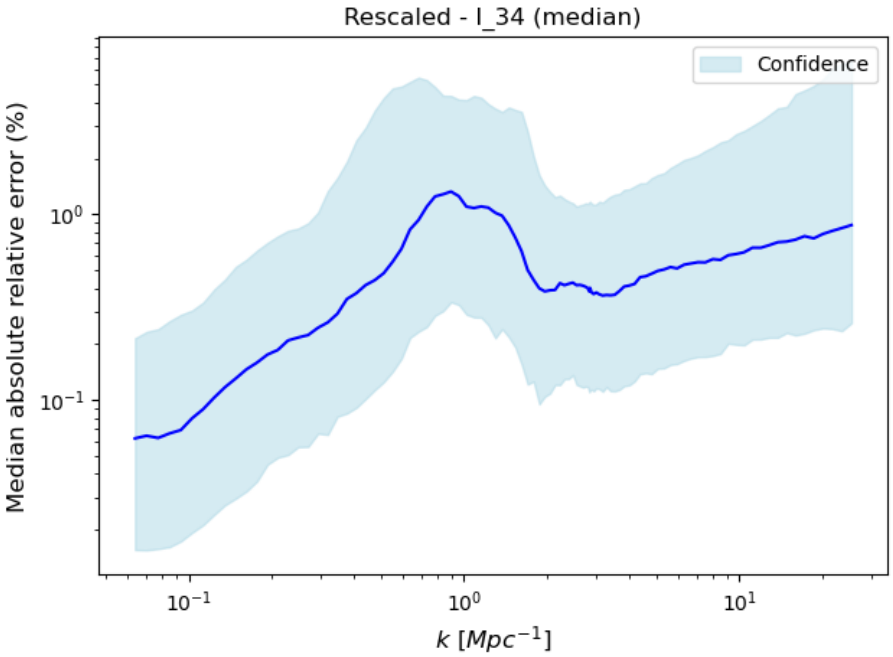

Figure B.16: Median absolute relative error for I_{34}

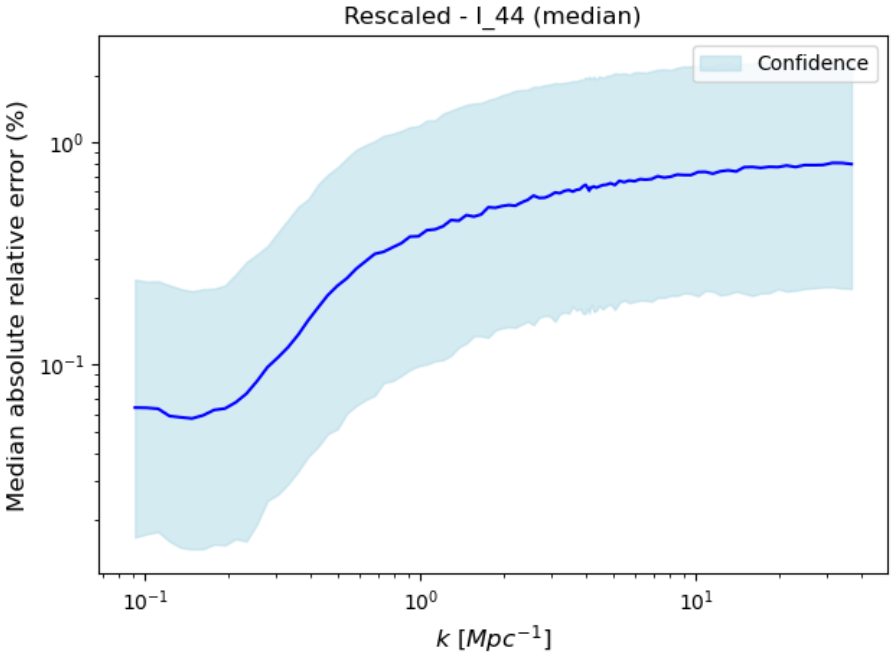

Figure B.17: Median absolute relative error for I_{44}

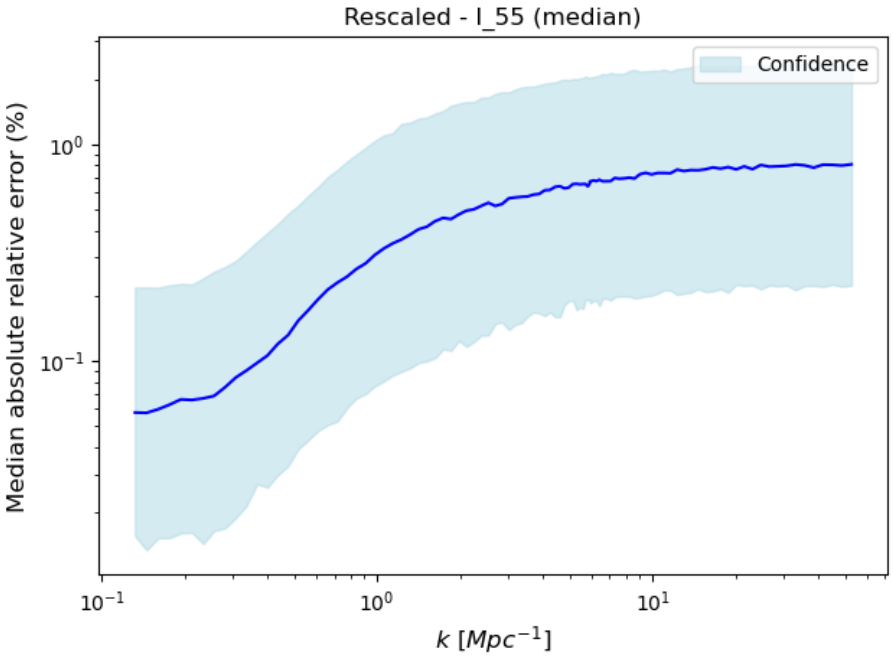

Figure B.18: Median absolute relative error for I_{55}

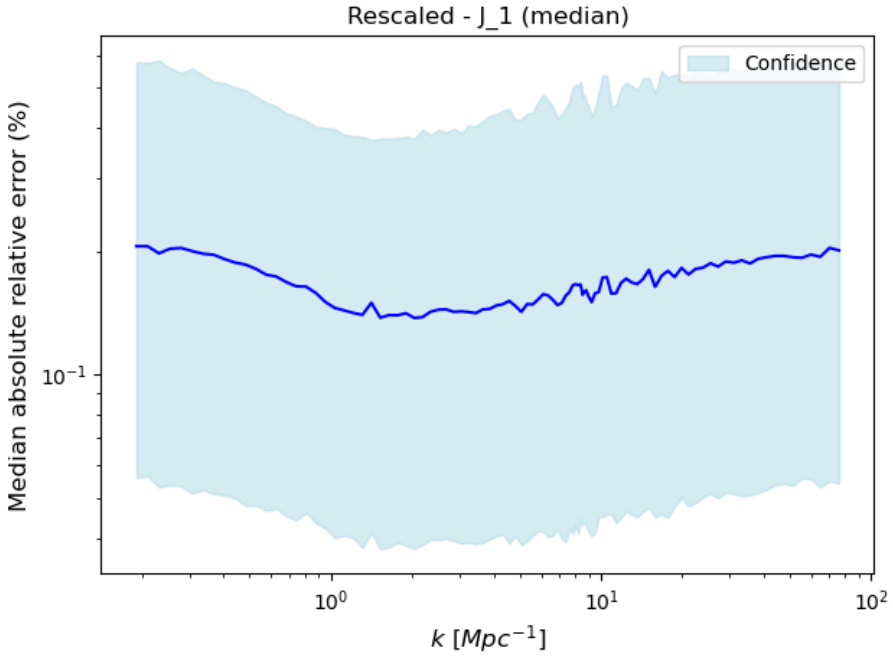

Figure B.19: Median absolute relative error for J_1

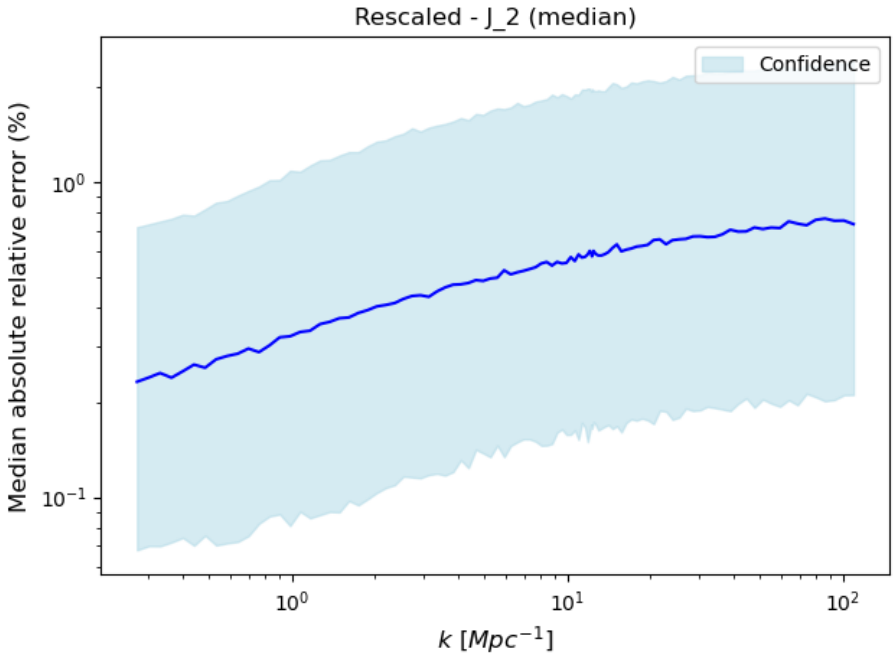

Figure B.20: Median absolute relative error for J_2

B.3 Outlier cosmologies

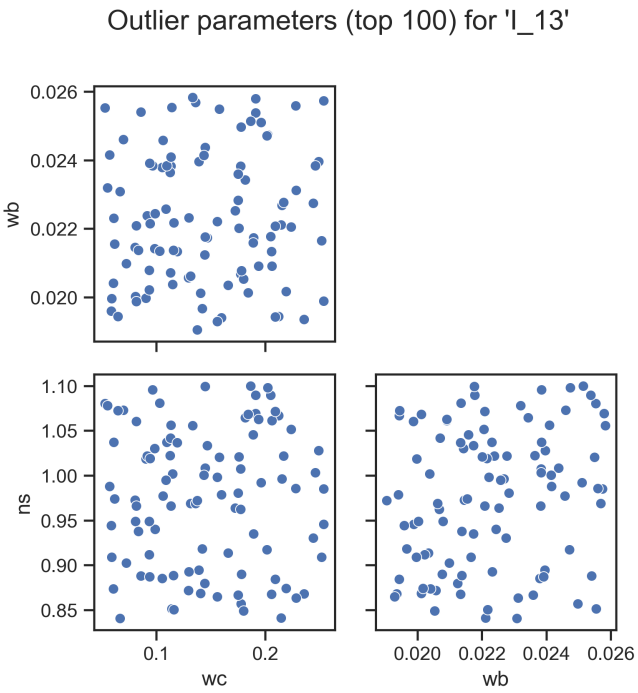

Figure B.21: Correlations between the cosmological parameters for the outliers of I_{13}

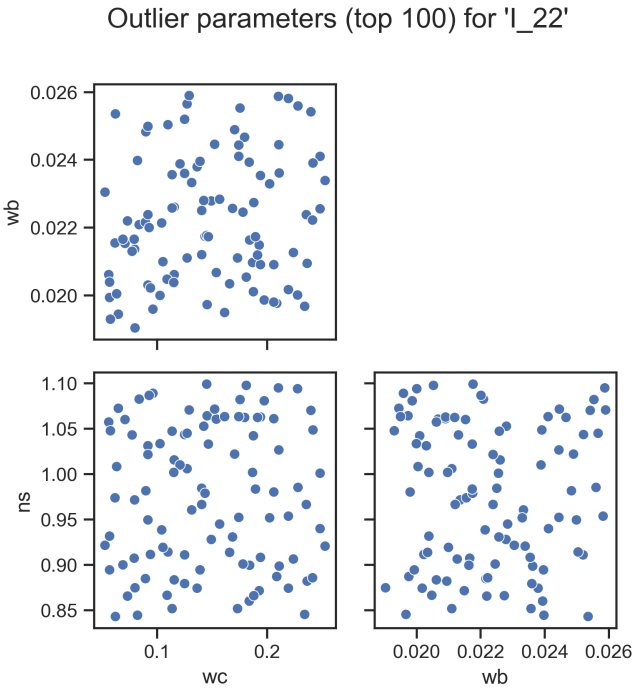

Figure B.22: Correlations between the cosmological parameters for the outliers of I_{22}

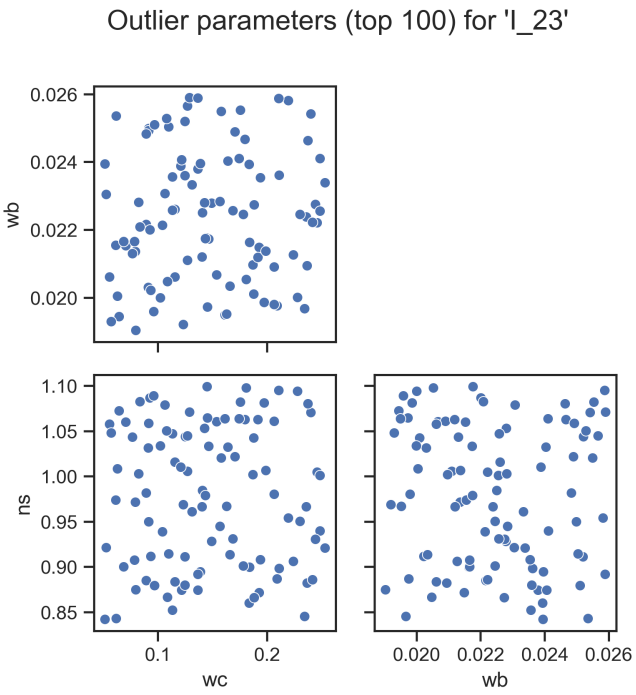

Figure B.23: Correlations between the cosmological parameters for the outliers of I_{23}

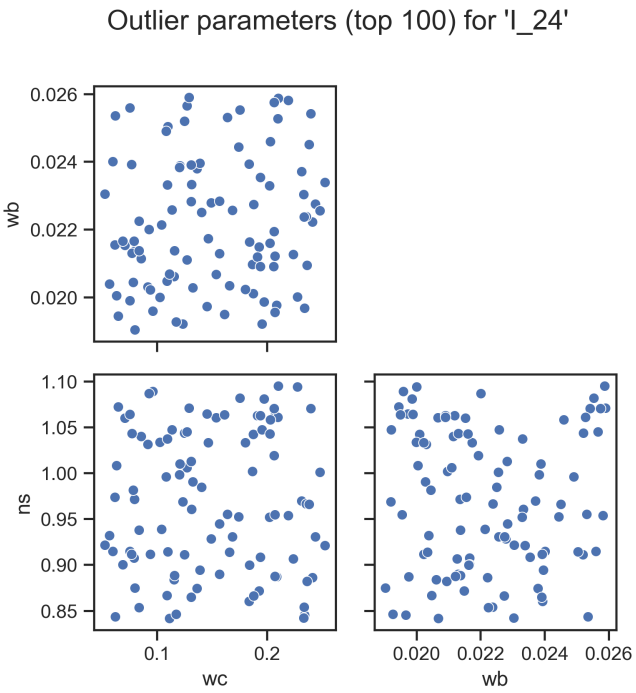

Figure B.24: Correlations between the cosmological parameters for the outliers of I_{24}

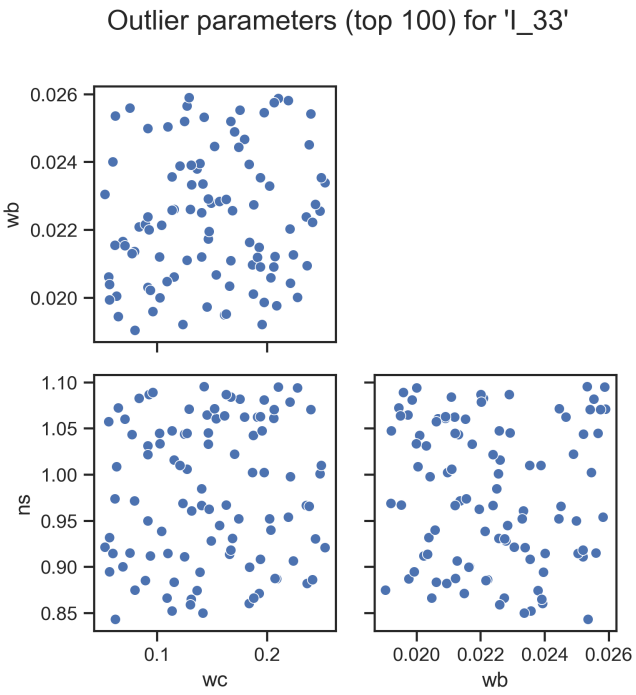

Figure B.25: Correlations between the cosmological parameters for the outliers of I_{33}

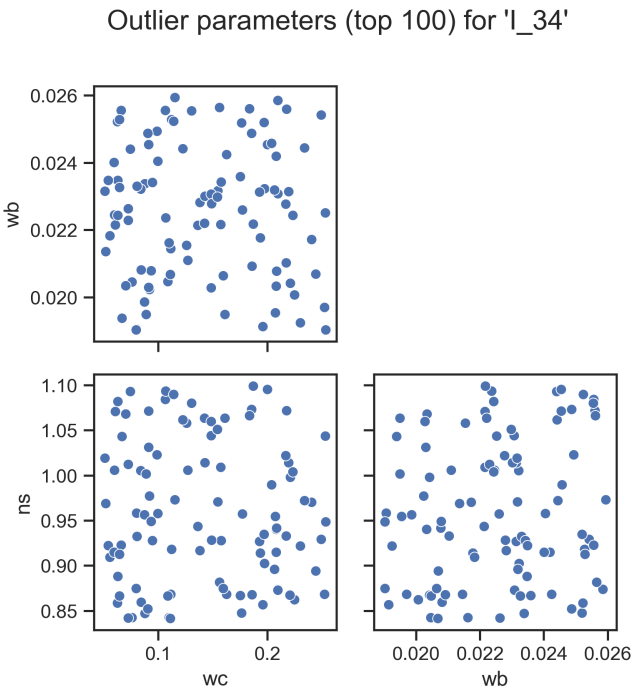

Figure B.26: Correlations between the cosmological parameters for the outliers of I_{34}

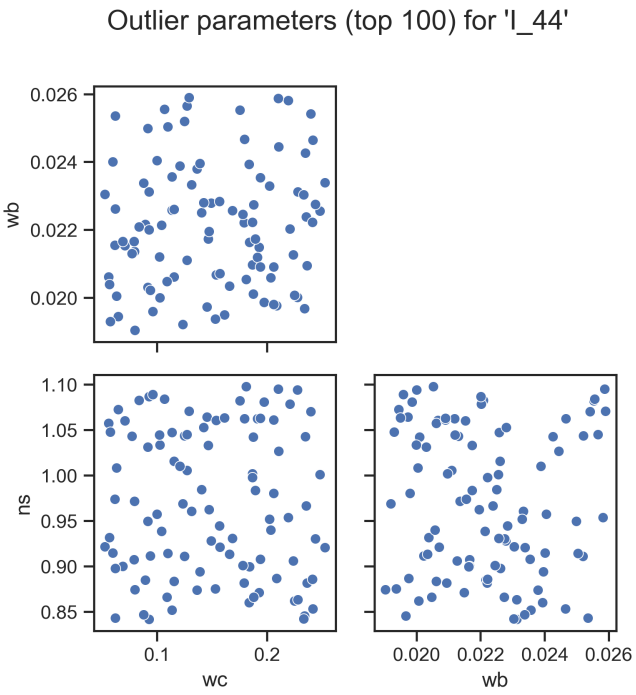

Figure B.27: Correlations between the cosmological parameters for the outliers of I_{44}

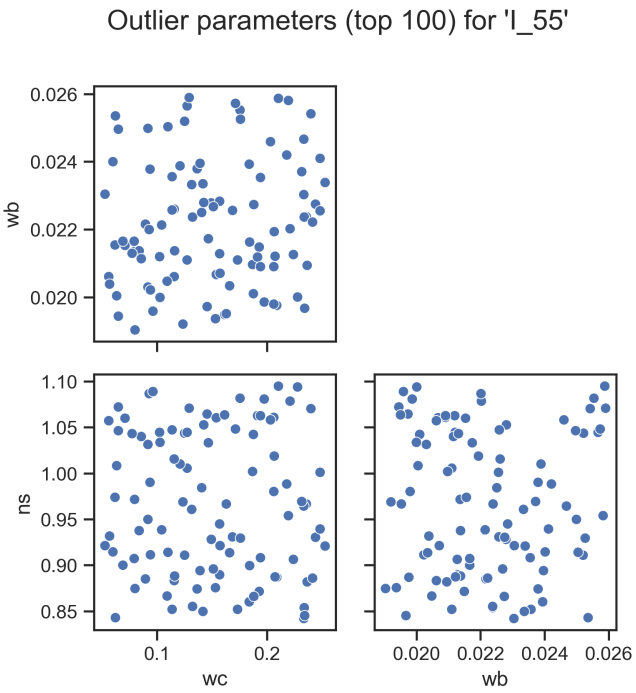

Figure B.28: Correlations between the cosmological parameters for the outliers of I_{55}

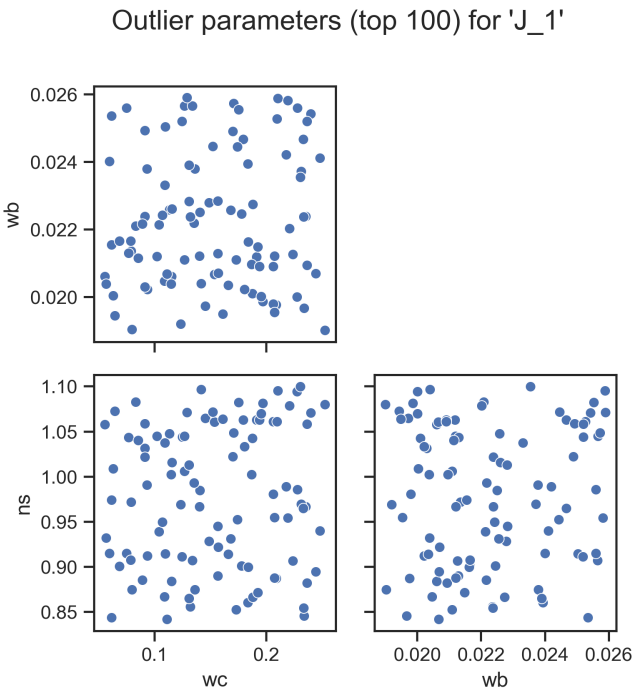

Figure B.29: Correlations between the cosmological parameters for the outliers of J_1

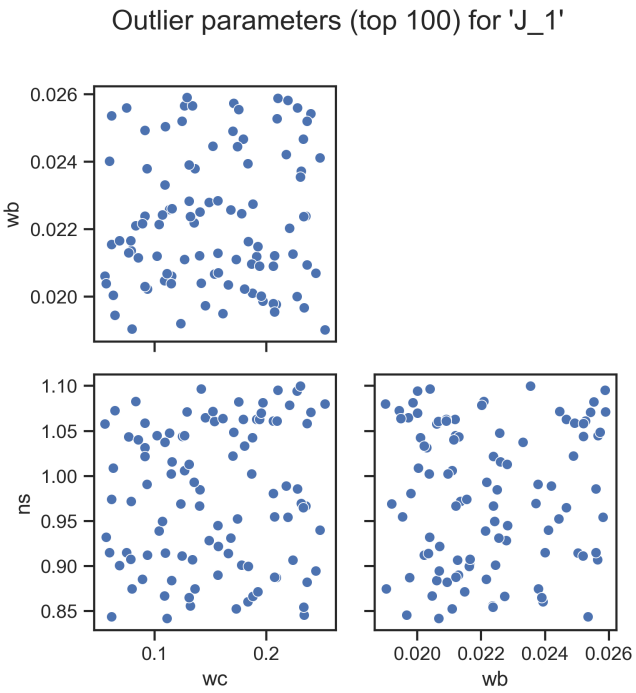

Figure B.30: Correlations between the cosmological parameters for the outliers of J_2

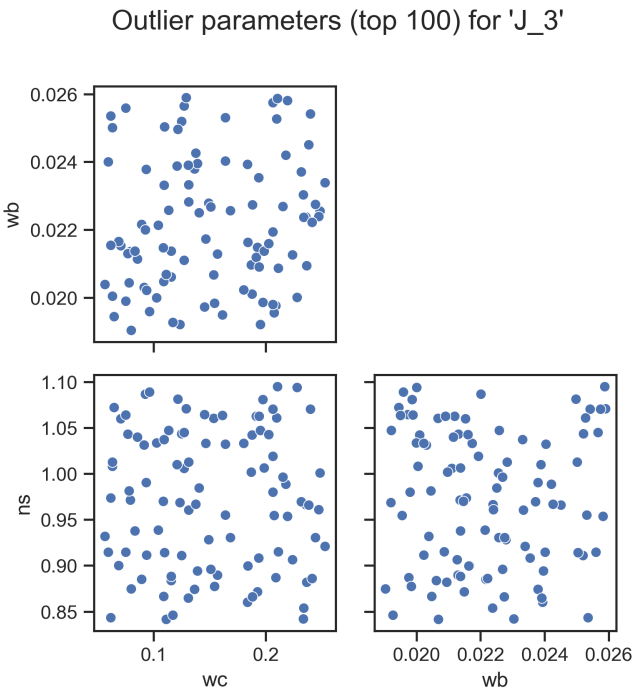

Figure B.31: Correlations between the cosmological parameters for the outliers of J_3