
Weak Lensing Cosmology with Scattering Transform

Sijin Chen



München 2025

Schwache Linsen-Kosmologie mit Streutransformationen

Sijin Chen

Masterarbeit an der Universitätssternwarte München
Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Sijin Chen
aus Suizhou, China

München, 08.05.2025

Erstgutachter: Prof. .Dr. Ralf Bender

Zweitgutachter: Dr. Stella Seitz

Declaration of authorship

I hereby declare that this thesis is my own work, and that I have not used any sources and aids other than those stated in the thesis.

Declaration of AI Usage

I declare that I have exclusively used ChatGPT 4 (*and/or any other AI tool, if applicable*) in my thesis/paper for the purposes listed below:

- *Brainstorming on my topic*
- *Correction of grammar, proofreading and editing*

Prompts are available in the AI Usage Reflection.

I confirm that I have checked my thesis/paper and its text for unintended plagiarism and that I have verified all facts and references used from the ChatGPT output.

München, 08.05.2025

Abstract

From observations of the Cosmic Microwave Background (CMB), we know that the initial density fluctuations in the early Universe followed a nearly Gaussian distribution, with small-amplitude perturbations distributed homogeneously across space. As the Universe evolved, gravitational instability amplified these perturbations, leading to the formation of large-scale structures. Over time, the initially Gaussian density field became increasingly non-Gaussian due to nonlinear gravitational evolution, making the extraction of non-Gaussian information both crucial and challenging, requiring advanced statistical methods.

In this project, we employ the scattering transform—a statistical tool inspired by convolutional neural networks—to analyze weak lensing convergence maps and constrain cosmological parameters. The scattering transform applies wavelet filters consisting of Gaussian envelopes modulated by harmonic oscillations of varying orientations and scales, performing convolutions on the input convergence maps. By taking the modulus of the convolved maps and averaging over spatial domains, we obtain a compact set of scattering coefficients that serve as robust summary statistics encoding non-Gaussian features.

This method is applied to the CosmoGridV1 simulations, a large suite of N -body simulations spanning a wide range of cosmologies. Cosmological parameter forecasts are first derived using the Fisher information matrix, providing a theoretical estimate of the constraining power of the scattering coefficients. In addition, we develop a deep learning-based emulator trained on simulation outputs across different cosmological parameters, including the effects of observational noise and systematic uncertainties. This emulator enables rapid predictions of scattering transform coefficients without the need for repeated expensive simulations.

We further perform Markov Chain Monte Carlo (MCMC) analyses to sample posterior distributions of cosmological parameters, using the emulator predictions in combination with realistic mock weak lensing data. Special attention is given to the impact of intrinsic alignments, shape noise, multiplicative bias and photometric uncertainty. Our results confirm that a tomographic analysis—including both auto- and cross-correlations between redshift bins—significantly enhances the precision of cosmological constraints, particularly on parameters such as Ω_m and σ_8 .

Contents

Abstract	II
1 Introduction	1
1.1 Basic knowledge of cosmology	2
1.2 Distance	7
1.3 Λ CDM model	9
1.4 w CDM model	12
2 Two points statistics	13
2.1 Inhomogeneity	13
2.2 Power spectrum of density perturbation	15
2.3 Gaussian and non-Gaussian information	18
2.4 Angular power spectrum	21
3 Weak lensing cosmology	23
3.1 Gravitational lensing basic	23
3.2 Lensing formalism	25
3.3 Projected overdensity	29
3.4 Shear measurement	32
3.5 Kaiser-Squires relation	33
3.6 E - and B -modes	34
4 Scattering transform	35
4.1 Wavelet transform	35
4.1.1 Extension of power spectrum	36
4.1.2 Formalism	36
4.2 Understanding the operations	41
4.2.1 Translation-invariant descriptors	41
4.2.2 Wavelet convolution and modulus	42
4.2.3 Hierarchy: information extraction beyond the power spectrum	44
4.2.4 Scattering operation in Fourier space	47
4.2.5 Wavelet power spectrum	48
5 Neural Network-based emulator	51
5.1 Terminology of ML	52
5.2 Deep learning	52

5.3	Bias-Variance tradeoff	55
5.4	Structure of deep neural networks	59
5.5	Optimization	63
6	Validation on CosmoGridV1 simulation	70
6.1	CosmoGridV1 simulation	70
6.2	Lightcone construction	71
6.3	Systematic effects	76
6.3.1	Baryonic feedback	76
6.3.2	Intrinsic alignment	78
6.3.3	Multiplicative bias	79
6.3.4	Shape noise	79
6.3.5	Photometric redshift uncertainty	80
6.4	Projected maps	81
6.4.1	Fibonacci sphere sampling	81
6.4.2	Convergence maps	81
7	Cosmological parameter constraints with Fisher matrix and MCMC	85
7.1	Summary statistics	85
7.1.1	Scattering coefficients	85
7.2	Fisher matrix analysis	88
7.2.1	Covariance estimation	92
7.2.2	Fisher forecast	94
7.2.3	Impact of resolution and shear	95
7.3	Emulator	99
7.4	MCMC sampling	100
7.4.1	Prior	103
7.4.2	Likelihood	104
7.4.3	Posterior	106
7.5	Handling masked data in scattering transform framework	110
8	Conclusion	118
A	Hankel transform	120
B	Discrete Fourier transform and power spectrum	123
C	Translation Invariance	128
D	Kaiser-Squire inversion	131
D.1	Spin 2 fields	131
D.2	Flat sky Kaiser-Squire inversion	132
D.3	Spin weighted spherical harmonics	133
D.4	Full sky Kaiser-Squire inversion	133

E	Systematic effects	136
E.1	Intrinsic alignment	136
E.2	Shape noise	140
F	Fibonacci sphere sampling	142
	Acknowledgements	153

Chapter 1

Introduction

Cosmology is the scientific study of the universe as a whole, including its origin, evolution, and structure formation. It tries to answer fundamental questions about the universe, such as how it began, what it is made of, and how it changes over time. The modern view of cosmology is that the universe started with the Big Bang about 13.8 billion years ago, a very hot and dense state from which space and time emerged. Shortly after the Big Bang, the universe experienced a very rapid and exponential expansion, called *inflation* [1], which is believed to be driven by some scalar fields [2]. During inflation, quantum fluctuations were stretched to cosmic scales, becoming the seeds for the growth of structures in the universe. These fluctuations also left imprints on the *Cosmic Microwave Background* (CMB), which we can observe today.

After the inflation era, the universe cooled down, allowing protons and neutrons to combine and form light atomic nuclei, such as hydrogen and helium. This process is called *Big Bang Nucleosynthesis*. The relative abundances of these elements provide strong evidence for the Big Bang model and match well with observations of the early universe [3]. After the CMB was emitted, the universe entered a period called the dark age. During this time, the universe was filled with neutral hydrogen and helium, and no stars or galaxies had formed yet. Because there were no light-emitting sources, this era is called “dark”. However, small density fluctuations in the matter distribution, seeded during inflation, began to grow under the influence of gravity. These fluctuations eventually led to the formation of the first stars and galaxies, marking the end of the dark age.

The formation of the first stars, known as Population III stars, started the process of reionization. Ultraviolet radiation from these stars ionized the surrounding neutral hydrogen. Galaxies continued to evolve through mergers and interactions, leading to the diverse structures we observe today. The large-scale structure of the universe, including galaxy clusters, cosmic voids, and filaments, also gradually formed mainly due to gravitational interactions.

In the present epoch, the universe’s expansion is accelerating again, a phenomenon attributed to dark energy. Observations of distant Type Ia supernovae [4] provide evidence for this accelerated expansion. Dark energy, often modeled as a cosmological constant Λ ,

now makes up about 70% of the universe’s total energy density.

To study the observed structures in the universe, many statistical methods have been proposed, such as N -point correlation functions, peak count statistics, and Minkowski functionals. In this thesis, we will focus on a novel method called the *scattering transform*, which generates a series of coefficients to help us understand cosmic structures on different scales.

In this chapter, we will first introduce some basic concepts of cosmology. In Section 1.1, we will discuss the expansion and components of the universe, as well as the Friedmann equations that connect matter density, curvature, and the evolution of the universe. In Section 1.2, we will introduce some frequently used cosmological distances. Finally, in Section 1.3, we will discuss the Λ CDM model, which is the standard model of cosmology.

In the later chapters, we will delve into the formalism of the two-point correlation function in Chapter 2. Chapter 3 will introduce the weak lensing effect, its properties, and its application in exploring the matter distribution of the universe. In Chapter 4, we will present the concept of the scattering transform, detailing its use in extracting information from weak lensing maps and its connection to traditional N -point correlation functions. Chapter 5 will focus on the neural network-based emulator, which predicts scattering transform coefficients for different cosmologies. Chapter 6 will describe the simulation framework employed for the scattering transform analysis. Finally, in Chapter 7, we will present the Fisher forecast using the scattering transform on various weak lensing maps and provide MCMC constraints on cosmological parameters.

1.1 Basic knowledge of cosmology

A lot of evidence has shown that the universe is expanding [5, 6, 7, 8]. This means that the distance in nowadays is bigger than it was in the past. So, it is convenient to express the distance by introducing the scale factor a . The value of scale factor is set to be $a_0 = 1$ today for convenience, where the subscript zero here means present time. At earlier times, a was smaller than it is today. We can imagine a grid being placed in space, similar to Fig. 1.1, that expands uniformly over time. Points on the grid, representing observers at rest, keep their coordinates fixed, meaning the comoving distance between two points—which simply reflects the difference in their coordinates and can be measured by counting grid cells as shown in Fig. 1.1—remains unchanged. However, the physical distance is proportional to the scale factor and changes as time progresses [9]. The mathematical relation between physical distance, d_p and comoving distance, χ , is usually expressed as:

$$d_p \propto a(t) \chi . \tag{1.1}$$

As a result of the universe’s expansion, the physical wavelength of light emitted by distant objects is stretched in proportion to the scale factor. Consequently, the observed wavelength is longer than it was at the time of emission. This is called *redshift*, defined as Eq. (1.2) [9]:

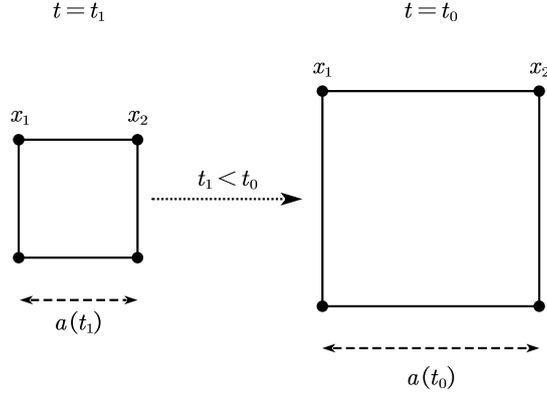


Figure 1.1: Expansion of the universe: The comoving distance between points x_1 and x_2 on the coordinate grid remains unchanged during expansion, meaning $|x_1 - x_2|$ stays constant. However, the physical distance is proportional to the scale factor and the comoving distance, $d_p = a(t) |x_1 - x_2|$, and thus increases as time progresses.

$$1 + z \equiv \frac{\lambda_{obs}}{\lambda_{emit}} = \frac{a_{obs}}{a_{emit}} = \frac{1}{a_{emit}}, \quad (1.2)$$

where λ_{obs} is the observed wavelength, λ_{emit} is the wavelength when light was emitted, a_{obs} is the scale factor at the time when it is observed, which is usually set to 1, and a_{emit} is the emission scale factor. So, the redshift and scale factor relation could be written as:

$$1 + z(t) = \frac{1}{a(t)}. \quad (1.3)$$

Apart from the scale factor and its evolution, the universe's smoothness is defined by another parameter: its geometry. There are three possibilities, described by curvature parameter k (also see in Fig. 1.2):

- Flat (Euclidean): the particles remain parallel as they continue their motion, $k = 0$
- Closed: the initially parallel particles slowly converge, similar to how lines of constant longitude meet at the poles on a 2-sphere, $k > 0$
- Open: the initially parallel particles diverge, akin to two marbles rolling off a saddle, $k < 0$

Before using mathematical equations to express the distance between two points in the universe, we have to introduce the two basic properties for matter distribution of the universe, usually called *cosmological principle* [11]:

- Homogeneity: Homogeneity means that the universe has a uniform composition and structure when averaged over sufficiently large distances ($> 100 \text{ Mpc}$).
- Isotropy: Isotropy means that the universe appears the same in all directions on large scales ($> 100 \text{ Mpc}$).

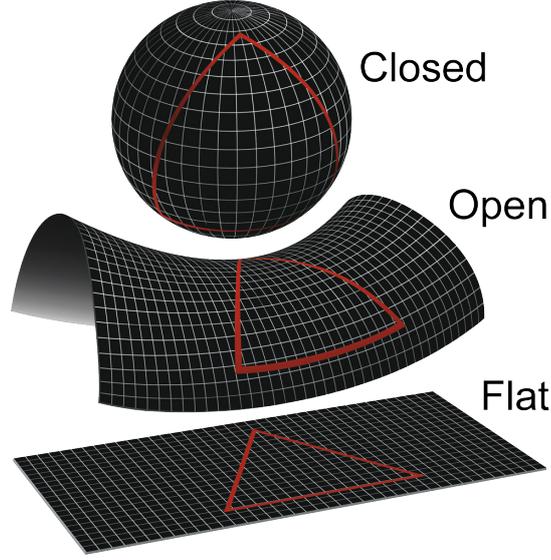


Figure 1.2: The geometry of the universe is characterized by curvature parameter k . From top to bottom: a closed universe with $k > 0$, open universe with $k < 0$ and flat universe with $k = 0$. Figure from [10]

Homogeneity shows that on large scales, the distribution of matter and energy is roughly the same everywhere, regardless of the location of the observer. And isotropy means that no matter which direction an observer looks, the large-scale properties of the universe (such as the distribution of galaxies or the Cosmic Microwave Background radiation) are uniform. Essentially, there is no preferred direction in the universe.

With these two properties, it is natural for us to measure the distance between points using spherical coordinate in an expanding curved universe:

$$dl^2 = a^2(t) \left[\frac{dr^2}{1 - kr^2} + r^2 d\vartheta^2 + r^2 \sin^2 \vartheta d\varphi^2 \right], \quad (1.4)$$

where r is the comoving distance in radial direction, $r \in [0, \infty]$, ϑ is the polar angle, $\vartheta \in [0, \pi]$ and φ is the azimuth angle, $\varphi \in [0, 2\pi)$.

To handle all three curvature cases simultaneously, we define:

$$f_k(\chi) \equiv \begin{cases} k^{-1/2} \sin(k^{1/2}\chi), & (k > 0) \\ \chi, & (k = 0) \\ |k|^{-1/2} \sinh(|k|^{-1/2}\chi), & (k < 0) \end{cases}. \quad (1.5)$$

Using Einstein summation convention, we could express Eq. (1.4) as:

$$dl^2 = \gamma_{ij} dx^i dx^j, \quad (1.6)$$

where both i and j should go through the summation over 1 to 3, meaning the sum in all directions of three-dimensional space. γ is called metric tensor, which could also be

expressed using a diagonal matrix in this case:

$$\gamma_{ij} = \begin{pmatrix} \frac{a^2(t)}{1-kr^2} & 0 & 0 \\ 0 & a^2(t) r^2 & 0 \\ 0 & 0 & a^2(t) r^2 \sin^2(\vartheta) \end{pmatrix}. \quad (1.7)$$

However, according to general relativity, due to the influence of gravity, time is no longer independent of space but is closely connected to it, with both being affected by gravity. To ensure the invariance of the spacetime interval, we need to introduce a time component in the metric tensor, which makes the metric tensor introduced above a four dimensional tensor:

$$\begin{aligned} ds^2 &= g_{\mu\nu} dx^\mu dx^\nu \\ &= -dt^2 + a^2(t) \left[\frac{dr^2}{1-kr^2} + r^2 d\vartheta^2 + r^2 \sin^2 \vartheta d\varphi^2 \right], \end{aligned} \quad (1.8)$$

where we use the natural unit system, so the speed of light is set to 1, $c = 1$. The index, Greek letter, μ and ν range from 0 to 3, where 0 represents time component and 1 to 3 represent spatial component. The four dimensional metric tensor, $g_{\mu\nu}$, which is called *Friedmann–Lemaître–Robertson–Walker (FLRW) metric*, is used to describe the homogenous and isotropic universe. So, with the time component included, the matrix form of the tensor is written as:

$$g_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & \frac{a^2(t)}{1-kr^2} & 0 & 0 \\ 0 & 0 & a^2(t) r^2 & 0 \\ 0 & 0 & 0 & a^2(t) r^2 \sin^2 \vartheta \end{pmatrix}. \quad (1.9)$$

Using FLRW metric, we could compute the metric connection:

$$\Gamma^\alpha_{\lambda\mu} = \frac{1}{2} g^{\alpha\nu} (\partial_\lambda g_{\mu\nu} + \partial_\mu g_{\lambda\nu} - \partial_\nu g_{\mu\lambda}), \quad (1.10)$$

with the partial derivatives ∂_μ with respect to the coordinates $(t, r, \vartheta, \varphi)$. And from this, calculating Ricci tensor, which represents gravitational effects due to matter, is possible:

$$R_{\mu\nu} = \partial_\lambda \Gamma^\lambda_{\mu\nu} - \partial_\nu \Gamma^\lambda_{\mu\lambda} + \Gamma^\lambda_{\lambda\rho} \Gamma^\rho_{\mu\nu} - \Gamma^\rho_{\mu\lambda} \Gamma^\lambda_{\nu\rho}. \quad (1.11)$$

With Ricci tensor $R_{\mu\nu}$ and metric tensor $g_{\mu\nu}$, we could define *Einstein tensor*, $G_{\mu\nu}$, which describes the curvature of spacetime due to the presence of matter and energy, in the following way:

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu}, \quad (1.12)$$

where R is the Ricci scalar, defined as $R = g^{\mu\nu} R_{\mu\nu}$ and represents an overall ‘‘average’’ curvature of spacetime, providing a scalar measure of how space is curved at a particular point.

Having introduced the Einstein tensor, which encapsulates the curvature of spacetime due to the presence of mass and energy, it is now essential to connect this geometric

framework with the physical sources of gravity. The relationship between the curvature of spacetime and the distribution of matter and energy is formalized in Einstein's field equations in Eq. (1.13). These equations express how the Einstein tensor is directly related to the energy-momentum tensor, $T_{\mu\nu}$, which describes the density and flow of energy and momentum in spacetime. By linking geometry to physics, the Einstein field equations provide the foundation for understanding how matter influences the structure of the universe.

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = 8\pi G T_{\mu\nu} , \quad (1.13)$$

where G is gravitational constant and Λ , also called *cosmological constant*, represents a constant energy density that fills space homogeneously, often associated with dark energy. Dark energy is responsible for the observed accelerated expansion of the universe. It acts as a repulsive force, pushing spacetime apart.

The general expression for energy-momentum tensor is written as:

$$T_{\mu\nu} = (\rho + p) U_\mu U_\nu + p g_{\mu\nu} , \quad (1.14)$$

where ρ is energy density, p is pressure and U is four-velocity. In the case of a universe depicted by FLRW metric, Eq. (1.14) can be expressed in a simpler form as a diagonal matrix, in Eq. (1.15), where only the energy density term ρ and the pressure term p are present. In this form, it can be interpreted as the energy-momentum tensor of a perfect fluid.

$$T_{\mu\nu} = \begin{pmatrix} -\rho & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix} . \quad (1.15)$$

According to Eq. (1.13), we could further derive equations that detail the evolution of the expanding universe. By setting $\mu = \nu = 0$, we could get the time-time component of Eq. (1.13):

$$G_{00} = \Lambda g_{00} = 8\pi G T_{00} . \quad (1.16)$$

By substituting the tensor components on both sides of the equation with their corresponding expressions, we arrive at the first Friedmann equation:

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G \rho}{3} - \frac{k}{a^2} + \frac{\Lambda}{3} , \quad (1.17)$$

where H here denotes *Hubble parameter*, \dot{a} is the time derivative of scale factor and ρ represents the energy density of the universe's constituents, such as matter and radiation. Equating the spatial terms on both sides of Eq. (1.13) with $\mu = \nu = i$ where $i = 1, 2, 3$, we can get:

$$G_{ii} = \Lambda g_{ii} = 8\pi G T_{ii} . \quad (1.18)$$

For all three indices of the spatial terms, we are able to derive:

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} (\rho + 3p) + \frac{\Lambda}{3} , \quad (1.19)$$

where \ddot{a} is the acceleration of scale factor and p is the pressure. Since the second Friedmann equation includes both the pressure term and the energy density term, we will use the so-called equation of state to link these two quantities:

$$p = w\rho , \quad (1.20)$$

where w is known as the equation-of-state parameter. It characterizes the relationship between the pressure p and the energy density ρ of the various components of the universe, such as matter, radiation, and dark energy.

- Non-relativistic matter: $w_m = 0$.
- Radiation: $w_r = \frac{1}{3}$.
- Dark energy: $w_\Lambda = -1$.

Combining two Friedmann equations Eq. (1.17) and Eq. (1.19), we could derive the equation governing the evolution of energy density:

$$\dot{\rho} + 3\frac{\dot{a}}{a}(p + \rho) = 0 , \quad (1.21)$$

By applying Eq. (1.21), we could further derive the evolution of energy density with in scale factor, where the general expression is given:

$$\rho \propto a^{-3(1+w)} , \quad (1.22)$$

Then the energy density of different constituents is shown:

- Matter: $\rho \propto a^{-3}$.
- Radiation: $\rho \propto a^{-4}$.
- Cosmological constant: $\rho = const.$

1.2 Distance

Having established the fundamental dynamics of the universe's expansion using the Friedmann equations, we now shift our focus to the concept of distance in cosmology. Understanding how distances are defined and measured in an expanding universe is essential for interpreting observations and testing cosmological models. In this section, we will explore various types of cosmological distances. The first concept is the *physical distance* $d_p(t)$, defined as the separation between two objects measured on a hypersurface of constant cosmic time. To determine the physical distance between two galaxies (one located at $r = 0$ and the other at $r = r_A$) at a given time t , we apply the metric, since $d_p(t) = \int_0^{r_A} ds$. This requires integrating along the path where $t, \vartheta, \varphi = const$, leading $ds^2 = a^2(t) d\chi^2 = a^2(t) \frac{dr^2}{1-kr^2}$, and then we get:

$$\begin{aligned}
 d_p(t) &= a(t) \int_0^{r_A} \frac{dr}{\sqrt{1-kr^2}} \\
 &= a(t) \int_0^{\chi_A} d\chi \\
 &= \begin{cases} k^{-1/2} a(t) \arcsin(k^{1/2} r_A), & (k > 0) \\ a(t) r_A, & (k = 0) \\ |k|^{-1/2} a(t) \operatorname{arcsinh}(|k|^{1/2} r_A), & (k < 0) \end{cases} \quad (1.23) \\
 &\equiv a(t) f_k^{-1}(r_A) = a(t) \chi_A,
 \end{aligned}$$

where $f_k^{-1}(r_A)$ is the inverse function of Eq. (1.5), defined as:

$$f_k^{-1}(r_A) = \int_0^{r_A} \frac{dr}{\sqrt{1-kr^2}} = \int_0^{\chi_A} d\chi = \begin{cases} k^{-1/2} \arcsin(k^{1/2} r_A), & (k > 0) \\ r_A, & (k = 0) \\ |k|^{-1/2} \operatorname{arcsinh}(|k|^{1/2} r_A), & (k < 0) \end{cases}. \quad (1.24)$$

With the mathematical expression for physical distance, we could now describe comoving distance using Eq. (1.1):

$$\chi_A = \frac{d^p}{a} = f_k^{-1}(r_A). \quad (1.25)$$

However, in actual observations, measuring the radial distance r directly is challenging. Instead, redshift is a more readily measurable quantity, making it a common and practical tool for expressing distances in cosmological contexts. We will then use redshift to express the distance. For light, it travels along null geodesics, which corresponds to a zero value of the FLRW metric interval (let's choose the coordinate so that light travels along radial direction):

$$ds^2 = -dt^2 + a^2(t) \frac{dr^2}{1-kr^2} = -dt^2 + a^2(t) d\chi^2 = 0. \quad (1.26)$$

Then, the comoving distance is shown:

$$\chi = \int_{t_1}^{t_0} \frac{dt}{a(t)} = \int_{\frac{1}{1+z}}^1 \frac{da}{a} \frac{1}{da/dt} = \int_0^z \frac{dz'}{H(z')}. \quad (1.27)$$

The distance-redshift relation in Eq. (1.27) would be helpful if $a(t)$ were known. However, we can use the observed relation to infer $a(t)$ or $H(z)$, but this requires a version where “distance” is replaced by a directly observable quantity. One such observable quantity is the angular diameter distance, which relates an object's physical size to its observed angular size.

In Euclidean geometry (see Fig. 1.3), it's easy to see that:

$$S = \vartheta d_A \quad \text{or} \quad d_A = \frac{S}{\vartheta}. \quad (1.28)$$

Accordingly, we define:

$$d_A \equiv \frac{S_p}{\vartheta}, \quad (1.29)$$

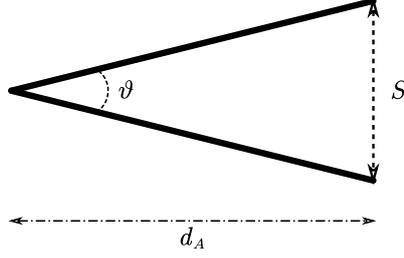


Figure 1.3: Defining the angular diameter distance

where S_p is the proper diameter of the object when the light we see left it, and ϑ is the observed angle. From the FLRW metric, the physical length S_p corresponding to an angle ϑ is, from $ds^2 = a^2(t) r^2 d\vartheta^2 \implies S_p = a(t) r \vartheta$. Thus, the general expression is given:

$$d_A(z) = a(t) r = \frac{r}{1+z} = \frac{f_k(\chi)}{1+z} = \frac{1}{1+z} f_k \left(\int_0^z \frac{dz'}{H(z')} \right). \quad (1.30)$$

1.3 Λ CDM model

After establishing key concepts like physical and comoving distances, as well as the universe's expansion, it is crucial to introduce the model that most accurately describes the universe's large-scale structure and evolution. The widely accepted framework in modern cosmology is the Lambda Cold Dark Matter (Λ CDM) model. This model offers a robust explanation for the cosmological observations, such as the Cosmic Microwave Background (CMB) radiation, the distribution of galaxies, clusters and cosmic voids in the large-scale structures of the universe, baryonic acoustic oscillations and so on. What's more important is that this model provides a comprehensive account of the universe's composition, like its name, cold dark matter and dark energy. Next, we will introduce dark matter and dark energy from both a theoretical and conceptual perspective. Based on Eq. (1.17), we could reformulate the equation in the case of a flat universe by setting $k = 0$:

$$\frac{8\pi G}{3H^2} \sum_i \rho_i = 1, \quad (1.31)$$

where we have absorbed dark energy term, $\Lambda/3$, into energy density ρ_i , so ρ_i here refers to radiation, matter and dark energy with different indices. Here the factor $\frac{8\pi G}{3H^2}$ is the reciprocal of critical density, as defined in Eq.(1.32), serves as the threshold distinguishing between closed, open and flat universe.

$$\rho_{cr} \equiv \frac{3H^2}{8\pi G}. \quad (1.32)$$

The current value of critical density is $\rho_{cr} = \frac{3H_0^2}{8\pi G} = 1.8788 \times 10^{-29} h^2 \text{g}\cdot\text{cm}^{-3}$ [9]. By dividing all energy densities with the critical density, we could define the *density parameters*, which indicates the percentage of different constituents in the universe:

$$\Omega_i \equiv \frac{\rho_i}{\rho_{cr}}. \quad (1.33)$$

Thus, we have:

$$\sum_i \Omega_i = 1 . \quad (1.34)$$

After introducing matter and dark energy through the Friedmann equation, we now turn to a conceptual explanation of these components, focusing on their roles in structure formation and cosmic expansion.

Cold dark matter: The concept of cold dark matter was first proposed by Fritz Zwicky [12] to explain the discrepancy between the theoretical prediction of the rotation speed of galaxies and the observational values. Subsequent observations, like gravitational lensing and CMB radiation, further confirm the existence of dark matter. Unlike normal matter, dark matter does not interact with the electromagnetic force. This means it does not absorb, reflect or emit light, making it invisible. Therefore, we could only study the distribution of dark matter using visible ordinary matters. The word “cold” indicates that dark matter particles move at non-relativistic speed, meaning they are slow-moving compared to the speed of light. Cold dark matter allows small-scale structures, such as galaxies, to form and leads to a “bottom-up” structure formation order. Besides the cold dark matter, the ordinary matter is called baryonic matter, which is made up of proton and neutrons, and also constitutes the total matter. According to Dark Energy Survey (DES) Year 3 Result [13], the percentage of total matter is $\Omega_m = 0.339^{+0.032}_{-0.031}$ with 68% confidence limits using Λ CDM model.

Dark energy: The concept of dark energy was introduced to explain the accelerated expansion of the universe, first observed through the study of distant supernovae in the late 1990s [4]. These observations revealed that the expansion rate of the universe is not slowing down, as initially expected, but rather accelerating. This unexpected behavior could not be explained by ordinary matter or dark matter alone, leading to the proposal of dark energy as a new form of energy that permeates space and exerts a repulsive gravitational effect. Unlike dark matter, which interacts gravitationally with ordinary matter, dark energy is thought to act uniformly throughout the universe, causing the accelerated expansion. It does not interact with light or other particles, making it detectable only through its influence on cosmic expansion and large-scale structures. Dark energy is often modeled as a cosmological constant (Λ) in the Λ CDM model, representing a constant energy density filling space homogeneously. According to the Planck 2018 Results [14], dark energy constitutes approximately $\Omega_\Lambda = 0.6847 \pm 0.0073$ of the total energy density of the universe, with a density parameter of $\Omega_\Lambda \approx 0.7$, making it the dominant component driving the universe’s expansion.

According to Eq.(1.19), the expansion of the universe requires the equation of state of dark energy, $w_{DE} < -\frac{1}{3}$ to make the second Friedmann equation greater than zero:

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} (\rho + 3p) = -\frac{4\pi G}{3} (1 + 3w_{DE}) \rho > 0 . \quad (1.35)$$

Fluids for which

$$\rho + 3p > 0 , \quad (1.36)$$

satisfy the **strong energy condition**. This means that, for dark energy to drive the accelerated expansion of the universe, it must violate this condition. The simplest approach to modeling dark energy is to set $w_{DE} = -1$, resulting in an exponential expansion $a(t) \propto \exp(H_0 t)$, where H_0 is the current value of Hubble parameter. Another alternative for dark energy is the **quintessence** model, where the dark energy adheres to the **weak energy condition**:

$$\rho + p > 0 , \quad (1.37)$$

which implies $w_{eff} > -1$. The case where $w_{DE} < -1$, known as **phantom energy**, has also been explored. It violates not just the strong energy condition of Eq.(1.36), but also the weak energy condition of Eq.(1.37). By applying Eq. (1.22), we can easily observe that in this model, ρ_{DE} increases as the universe expands. Now, consider a flat universe containing only two components: matter and phantom energy. The scale factor evolves as described by Eq. (1.38) for $t > t_{eq}$, where t_{eq} marks the time when the densities of matter and phantom energy become equal.

$$a(t) \simeq a(t_{eq}) \left[(1 + w_{DE}) \frac{t}{t_{eq}} - w_{DE} \right]^{\frac{2}{3(1+w_{DE})}} . \quad (1.38)$$

Eq. (1.38) indicates that the universe diverges at a finite time t_{BR} given by:

$$t_{BR} = \left(\frac{w_{DE}}{1 + w_{DE}} \right) t_{eq} \quad (1.39)$$

The time t_{BR} is referred to as the **Big Rip**. The scale factor's divergence is linked to the divergence of the phantom energy density, which evolves as:

$$\rho_{ph} \propto \left[(1 + w_{DE}) \frac{t}{t_{eq}} - w_{DE} \right]^{-2} . \quad (1.40)$$

After discussing the roles of dark matter and dark energy in shaping the universe, it is essential to consider another significant component of the cosmos: relativistic matter. This category includes radiation and neutrinos, both of which played a crucial role in the early universe and continue to influence its evolution. In the early universe, radiation and neutrinos, as forms of relativistic matter, played a dominant role. During the radiation-dominated era, radiation, primarily photons, governed the expansion of the universe, with its energy density scaling as $a(t) \propto t^{-4}$. As the universe expanded and cooled, photons decoupled from matter, leading to the Cosmic Microwave Background (CMB). Neutrinos, nearly massless particles, decoupled early and became relic particles that still exist today. Though weakly interacting, neutrinos influenced the formation of cosmic structures and left imprints on the CMB. As the universe evolved, the energy density of radiation decreased faster than matter, leading to the matter-dominated era. Despite their reduced role today, radiation and neutrinos continue to influence cosmic expansion and structure formation, providing crucial insights into the universe's early evolution. And based on the Planck 2018 Results [14], $\Omega_r \approx 5 \times 10^{-5}$.

1.4 w CDM model

The w CDM model is an extension of the standard Λ CDM model, in which the dark energy equation of state parameter w is treated as a constant, but is allowed to deviate from the cosmological constant value $w = -1$. When $w \neq -1$, the energy density of dark energy evolves differently from that of a cosmological constant, potentially becoming more or less dominant at different epochs.

Beyond this, dynamical dark energy models further generalize the framework by allowing w to vary with time or redshift. A commonly used example is the CPL (Chevallier–Polarski–Linder) parameterization [15], in which the equation of state takes the form $w(a) = w_0 + w_a(1 - a)$, capturing possible evolution of dark energy across cosmic history.

Chapter 2

Two points statistics

In order to probe LSS in the Universe, the statistical properties of the matter distribution are of great importance. The two-point correlation functions (2PCF) and the power spectrum are two of the most widely used statistical tools in cosmology. They provide a means to quantify the spatial distribution of matter, allowing us to understand the large-scale structure of the universe and its evolution over time. By studying the two-point correlation function, we can gain insights into the clustering properties of galaxies, the nature of dark matter, and the underlying cosmological parameters that govern the universe's dynamics.

In Section 2.1, we will discuss the inhomogeneity of the universe and the statistical properties of density perturbations. We will introduce the concept of the two-point correlation function and its relation to the power spectrum in Section 2.2. In Section 2.3, we will explain how power spectrum extract Gaussian information and why we need other tools to gain non-Gaussian information. Finally, in Section 2.4, we will show the 2D projection of the 3D power spectrum, the angular power spectrum, and its relation to the 2PCF.

2.1 Inhomogeneity

Although the universe is statistically homogeneous on large scale, observations have shown that some fluctuations still exist on small scales. One of the most significant evidence is the CMB, where temperature fluctuations at the level of $10^{-5} K$ provides a window into the early fluctuations of the early universe. The theoretical framework of inflation not only supports this view but also provides a compelling mechanism to explain the origin of these initial inhomogeneities, leading to the observed structure formation. We can not predict the specific location of over- ($\delta > 0$) or underdensities ($\delta < 0$), but instead rely on their statistical properties.

We will approach the statistical properties using two types of averages: the *volume average* and the *ensemble average*. The ensemble average is a theoretical construct, while the volume average is a more practical concept used in observations.

For ensemble average, we assume that our universe is just one of an ensemble of an infinite number of possible universes (realizations of random process) that could have resulted from random process producing the initial perturbations. Here we use the probability distribution $\text{Prob}(\gamma)$ to denote the random process. Thus the ensemble average of a quantity f depending on the initial random process is written as:

$$\langle f \rangle \equiv \int d\gamma \text{Prob}(\gamma) f(\gamma) . \quad (2.1)$$

Here f could be, e.g., the value of $\rho(\mathbf{x})$ at some location \mathbf{x} .

However, we can observe only one universe from that ensemble. So, it would be reasonable that the statistics we get by comparing different parts of the universe should be similar to the statistics of a given part of the universe over different realizations. In this case, volume average is more useful:

$$\bar{f} \equiv \frac{1}{V} \int_V d^3x f(\mathbf{x}) . \quad (2.2)$$

The fields $f(\mathbf{x})$ that satisfy

$$\bar{f} = \langle f \rangle , \quad (2.3)$$

for an infinite volume V (for \bar{f}) and an arbitrary location \mathbf{x} (for $\langle f \rangle$) are called *ergodic*. The equality does not hold for a finite volume V ; the difference is called *cosmic variance*. We express the inhomogeneous quantities as the combination of a homogeneous *background* value and a *perturbation* component, representing a deviation from the uniform background. Thus, the energy density can be written as:

$$\rho(\mathbf{x}, t) = \bar{\rho}(t) + \delta\rho(\mathbf{x}, t) , \quad (2.4)$$

where $\bar{\rho}$ is the background density, \mathbf{x} is the comoving 3D space coordinate and $\delta\rho$ is the density perturbation. We could further define the relative perturbation as:

$$\delta(\mathbf{x}, t) \equiv \frac{\rho(\mathbf{x}, t) - \bar{\rho}(t)}{\bar{\rho}(t)} , \quad (2.5)$$

where we could easily see that $\langle \delta(\mathbf{x}, t) \rangle = \langle \rho(\mathbf{x}, t) - \bar{\rho}(t) \rangle / \langle \bar{\rho}(t) \rangle = 0$. Thus we can not use $\langle \delta \rangle$ as a measure of inhomogeneity, instead, we can use the square of δ (in the following, we will omit t for simplicity), which is necessarily non-negative everywhere:

$$\langle \delta^2 \rangle = \frac{\langle \delta\rho^2 \rangle}{\bar{\rho}^2} . \quad (2.6)$$

However, it only informs us about the magnitude of the inhomogeneity, without providing any details regarding its shape or size. To gain deeper insights, we introduce the *correlation function* ξ . Specifically, we define the *2-point density correlation function* as:

$$\xi(\mathbf{x}_1, \mathbf{x}_2) \equiv \langle \delta(\mathbf{x}_1) \delta(\mathbf{x}_2) \rangle . \quad (2.7)$$

The correlation function is positive if the density perturbations at both \mathbf{x}_1 and \mathbf{x}_2 are expected to have the same sign, and negative if there is an overdensity at one point and an underdensity at the other. Because of statistical homogeneity, $\xi(\mathbf{x}_1, \mathbf{x}_2)$ depends only on the separation $\mathbf{r} \equiv \mathbf{x}_2 - \mathbf{x}_1$. Therefore, we can redefine ξ as:

$$\xi(\mathbf{r}) \equiv \langle \delta(\mathbf{x}) \delta(\mathbf{x} + \mathbf{r}) \rangle \quad (2.8)$$

Due to the property of isotropy, $\xi(\mathbf{r})$ does not depend on direction, behaving in a spherically symmetric manner. As a result, we can express it as:

$$\xi(\mathbf{r}) = \xi(r) . \quad (2.9)$$

The correlation function is significantly positive when r is smaller than the characteristic size of an over- or underdense region, but decreases for larger distances.

The correlation function at zero separation gives the variance of the density perturbation:

$$\langle \delta^2 \rangle = \langle \delta(\mathbf{x}) \delta(\mathbf{x}) \rangle = \xi(0) . \quad (2.10)$$

Also, we could define the correlation function $\hat{\xi}(\mathbf{r})$ for a single realization as a volume average:

$$\hat{\xi}(\mathbf{r}) \equiv \frac{1}{V} \int d^3x \delta(\mathbf{x}) \delta(\mathbf{x} + \mathbf{r}) . \quad (2.11)$$

Integrating over \mathbf{r} and assuming periodic boundary conditions, we get the integral constraint:

$$\int d^3r \hat{\xi}(\mathbf{r}) = \frac{1}{V} \int d^3r d^3x \delta(\mathbf{x}) \delta(\mathbf{x} + \mathbf{r}) = \frac{1}{V} \int d^3x \delta(\mathbf{x}) \int d^3r \delta(\mathbf{x} + \mathbf{r}) = 0 . \quad (2.12)$$

As a result, $\xi(r)$ must eventually turn negative, indicating that at a certain distance from an overdense region, it is more probable to encounter an underdense region. As the separation increases further, ξ may oscillate around zero, with the amplitude of these oscillations gradually diminishing.

2.2 Power spectrum of density perturbation

Fourier transform

Usually we will decompose the correlation function into the summation of difference wavelength by Fourier transform, so the following gives a short review about the properties of Fourier transform and also the convention we use.

For mathematical convenience, we assume the observable part of the universe lies within a fiducial cubic box, volume $V = L^3$, with periodic boundary conditions. We can now expand any function of space $f(\mathbf{x})$ in the following way:

$$f(\mathbf{x}) = \frac{V}{(2\pi)^3} \int f(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3x , \quad (2.13)$$

and the inverse Fourier transform is:

$$f(\mathbf{k}) = \frac{1}{V} \int f(\mathbf{x}) e^{-i\mathbf{k}\cdot\mathbf{x}} d^3x . \quad (2.14)$$

The Fourier transform representation of $f(\mathbf{x})$ is derived from its expansion in a Fourier series [11], where the set of functions $\{e^{i\mathbf{k}\cdot\mathbf{x}}\}$ with discrete wave vectors \mathbf{k} provides a complete and orthogonal basis for the expansion. These properties of Fourier series also extend to the Fourier transform, leading to the expression:

$$\int (e^{i\mathbf{k}'\cdot\mathbf{x}})^* \times (e^{i\mathbf{k}\cdot\mathbf{x}}) d^3x = \int e^{i(\mathbf{k}-\mathbf{k}')\cdot\mathbf{x}} d^3x = (2\pi)^3 \delta_D(\mathbf{k} - \mathbf{k}') , \quad (2.15)$$

for the orthogonality relation where \mathbf{k}, \mathbf{k}' are distinct wave vectors and $\delta_D(\mathbf{k} - \mathbf{k}')$ is the Dirac delta function. The symbol $*$ represents the complex conjugate. This relation holds true also for Fourier space:

$$\int e^{i(\mathbf{x}-\mathbf{x}')\cdot\mathbf{k}} d^3k = (2\pi)^3 \delta_D(\mathbf{x} - \mathbf{x}') . \quad (2.16)$$

For the real field $f(\mathbf{x})$, we could easily see from Eq. (2.14) that:

$$f(-\mathbf{k}) = f^*(\mathbf{k}) . \quad (2.17)$$

Another key result related to the Fourier transform is the convolution theorem. This theorem states that the convolution of two functions in real space corresponds to the Fourier transform of their product in Fourier space. The convolution of two functions or random continuous density fields, ρ and g , is defined as:

$$(f * g)(\mathbf{x}) = \int f(\mathbf{y}) g(\mathbf{x} - \mathbf{y}) d^3y , \quad (2.18)$$

and then the convolution theorem is written as:

$$(f * g)(\mathbf{x}) = \frac{V^2}{(2\pi)^3} \int f(\mathbf{k}) g(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3k , \quad (2.19)$$

where usually the volume V is absorbed into the Fourier mode and the convolution theorem could be reformulated as:

$$(f * g)(\mathbf{x}) = \frac{1}{(2\pi)^3} \int f(\mathbf{k}) g(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3k , \quad (2.20)$$

where the new Fourier mode $f(\mathbf{k})$ and $g(\mathbf{k})$ on the right hand side are the product of volume V and the original Fourier counterpart.

Power spectrum

Now we can expand the density perturbation using the integral of Fourier modes:

$$\delta(\mathbf{x}) = \frac{V}{(2\pi)^3} \int_{\mathbf{k}} \delta(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3k . \quad (2.21)$$

Therefore, the two point correlation function can be represented by their corresponding Fourier modes:

$$\xi(\mathbf{r}) = \langle \delta(\mathbf{x}) \delta(\mathbf{x} + \mathbf{r}) \rangle \quad (2.22)$$

$$= \frac{V^2}{(2\pi)^6} \left\langle \int_{\mathbf{k}} \delta(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3k \int_{\mathbf{k}'} \delta(\mathbf{k}') e^{i\mathbf{k}'\cdot(\mathbf{x}+\mathbf{r})} d^3k' \right\rangle \quad (2.23)$$

$$= \frac{V^2}{(2\pi)^6} \frac{1}{V} \int_V d^3x \int_{\mathbf{k}} \delta(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d^3k \int_{\mathbf{k}'} \delta(\mathbf{k}') e^{i\mathbf{k}'\cdot(\mathbf{x}+\mathbf{r})} d^3k' \quad (2.24)$$

$$= \frac{V}{(2\pi)^6} \int_V \int_{\mathbf{k}} \int_{\mathbf{k}'} \delta(\mathbf{k}) \delta(\mathbf{k}') e^{i(\mathbf{k}+\mathbf{k}')\cdot\mathbf{x}} e^{i\mathbf{k}'\cdot\mathbf{r}} d^3x d^3k d^3k' \quad (2.25)$$

$$= \frac{V}{(2\pi)^6} \int_{\mathbf{k}} \int_{\mathbf{k}'} \delta(\mathbf{k}) \delta(\mathbf{k}') e^{i\mathbf{k}'\cdot\mathbf{r}} (2\pi)^3 \delta_D(\mathbf{k} + \mathbf{k}') d^3k d^3k' \quad (2.26)$$

$$= \frac{V}{(2\pi)^3} \int_{\mathbf{k}'} \delta(-\mathbf{k}') \delta(\mathbf{k}') e^{i\mathbf{k}'\cdot\mathbf{r}} d^3k' \quad (2.27)$$

$$= \frac{V}{(2\pi)^3} \int_{\mathbf{k}} \delta(\mathbf{k}) \delta^*(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} d^3k \quad (2.28)$$

$$= \frac{V}{(2\pi)^3} \int_{\mathbf{k}} |\delta(\mathbf{k})|^2 e^{i\mathbf{k}\cdot\mathbf{r}} d^3k, \quad (2.29)$$

where from Eq. (2.23) to Eq. (2.24), we use the concept of volume average to calculate the ensemble average, then from Eq. (2.25) to Eq. (2.26) we use property of orthogonality of plane waves. And this gives the Fourier transform of the two point correlation function. By taking ensemble averages on both sides of Eq. (2.29), we can define the *power spectrum* as follows [11]:

$$P(\mathbf{k}) \equiv \langle |\delta(\mathbf{k})|^2 \rangle. \quad (2.30)$$

Unlike the correlation function, the power spectrum $P(\mathbf{k})$ is positive everywhere. And from statistical isotropy

$$\xi(\mathbf{r}) = \xi(r) \quad \Rightarrow \quad P(\mathbf{k}) = P(k). \quad (2.31)$$

An alternative definition of the power spectrum is given by [9]:

$$\langle \delta^*(\mathbf{k}') \delta(\mathbf{k}) \rangle \equiv (2\pi)^3 \delta_D(\mathbf{k} - \mathbf{k}') P(\mathbf{k}). \quad (2.32)$$

From Eq. (2.28) and Eq. (2.32), we could easily see that 2PCF is the inverse Fourier transform of the power spectrum:

$$\begin{aligned} \xi(\mathbf{r}) &= \frac{V}{(2\pi)^3} \int_{\mathbf{k}} \delta(\mathbf{k}) \delta^*(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} d^3k \\ &= \frac{V}{(2\pi)^3} \int_{\mathbf{k}} P(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} d^3k. \end{aligned} \quad (2.33)$$

Using spherical coordinates and doing the angular integrals, we could further simplify the Eq. (2.33) and get the relation between the 1D correlation function $\xi(r)$ and 1D

power spectrum $P(k)$,

$$\begin{aligned}
 \xi(\mathbf{r}) &= \frac{V}{(2\pi)^3} \int_{\mathbf{k}} P(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} d^3k \\
 &= \frac{V}{(2\pi)^3} \int_0^{2\pi} \int_0^\pi \int_k P(k) e^{ikr \cos \theta} k^2 \sin \theta d\varphi d\theta dk \\
 &= \frac{V}{(2\pi)^3} 2\pi \int_0^\pi \int_k P(k) e^{ikr \cos \theta} k^2 \sin \theta d\varphi d\theta dk \\
 &= -\frac{V}{(2\pi)^2} \int_0^\pi \int_k P(k) k^2 \frac{1}{ikr} d e^{ikr \cos \theta} dk \\
 &= -\frac{V}{(2\pi)^2} \int_k \frac{P(k) k}{ir} [e^{ikr \cos \pi} - e^{ikr \cos 0}] dk \\
 &= -\frac{V}{(2\pi)^2} \int_k \frac{P(k) k}{ir} (-2i \sin(kr)) dk \\
 &= \frac{V}{(2\pi)^3} \int_k P(k) \frac{\sin(kr)}{kr} 4\pi k^2 dk \\
 &= \frac{V}{(2\pi)^3} \int_k 2j_0(kr) P(k) 2\pi k^2 dk,
 \end{aligned} \tag{2.34}$$

where $j_0(x)$ is zeroth order spherical Bessel function defined as :

$$j_0(x) = \frac{\sin x}{x}. \tag{2.35}$$

In the small-angle (flat-sky) limit, the spherical symmetry reduces to a two-dimensional symmetry, and the corresponding Bessel transform becomes a Hankel transform involving J_0 . The mathematical expression between two-point correlation function and power spectrum in 2D flat coordinate could be written as:

$$\xi(\theta) = \int_0^\infty \frac{\ell d\ell}{2\pi} C_\ell J_0(\ell\theta). \tag{2.36}$$

where J_0 is Bessel function of first kind.

2.3 Gaussian and non-Gaussian information

In the later stages of the universe, as structure formation progresses, density perturbations evolve and become increasingly non-Gaussian due to nonlinear gravitational interactions. However, understanding the properties of a Gaussian field remains fundamental, as many early-universe models assume Gaussian initial conditions, and the statistical tools developed for Gaussian fields provide a crucial foundation. In this subsection, we will explore the key characteristics of Gaussian fields, which continue to play an important role in cosmological analysis even in the presence of non-Gaussianity.

A general Gaussian random field $\delta_0(\mathbf{x})$ with vanishing mean is completely specified by its 2PCF, $\langle \delta_0(\mathbf{x}_1) \delta_0(\mathbf{x}_2) \rangle = \xi(\mathbf{x}_1 - \mathbf{x}_2)$. The expectation value of three fields vanishes:

$$\langle \delta_0(\mathbf{x}_1) \delta_0(\mathbf{x}_2) \delta_0(\mathbf{x}_3) \rangle = 0 . \quad (2.37)$$

The expectation value with four fields is non-zero, but completely determined by $\xi(\mathbf{r})$:

$$\begin{aligned} \langle \delta_0(\mathbf{x}_1) \delta_0(\mathbf{x}_2) \delta_0(\mathbf{x}_3) \delta_0(\mathbf{x}_4) \rangle = & \xi(\mathbf{x}_1 - \mathbf{x}_2) \xi(\mathbf{x}_4 - \mathbf{x}_3) + \xi(\mathbf{x}_1 - \mathbf{x}_3) \xi(\mathbf{x}_4 - \mathbf{x}_2) \\ & + \xi(\mathbf{x}_1 - \mathbf{x}_4) \xi(\mathbf{x}_3 - \mathbf{x}_2) , \end{aligned} \quad (2.38)$$

where the three terms arise from the three distinct possibilities of combining the four fields into two pairs. The expansion by pairing the fields similarly works for any higher, even number of fields, and it is known as *Wick's theorem*. For the general case, any ensemble average of products of δ_0 can be written in terms of products of ensemble average of pairs as:

$$\langle \delta_0(\mathbf{x}_1) \dots \delta_0(\mathbf{x}_{2n+1}) \rangle = 0 , \quad (2.39)$$

$$\langle \delta_0(\mathbf{x}_1) \dots \delta_0(\mathbf{x}_{2n}) \rangle = \sum_{\text{all partitions}} \prod_{\text{pairs}(i,j)} \langle \delta_0(\mathbf{x}_i) \delta_0(\mathbf{x}_j) \rangle , \quad (2.40)$$

where the $\sum \prod$ notation indicates a sum over all possible partitions of the set of points into pairs. Wick's theorem works in Fourier space as well, the Fourier-space counterpart of Eq. (2.37) and Eq. (2.38)

$$\langle \delta_0(\mathbf{k}_1) \delta_0(\mathbf{k}_2) \delta_0(\mathbf{k}_3) \rangle = 0 , \quad (2.41)$$

$$\begin{aligned} \langle \delta_0(\mathbf{k}_1) \delta_0(\mathbf{k}_2) \delta_0(\mathbf{k}_3) \delta_0(\mathbf{k}_4) \rangle = & (2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{k}_2) \delta_D(\mathbf{k}_3 + \mathbf{k}_4) P(\mathbf{k}_1) P(\mathbf{k}_3) \\ & + (2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{k}_3) \delta_D(\mathbf{k}_2 + \mathbf{k}_4) P(\mathbf{k}_1) P(\mathbf{k}_2) \\ & + (2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{k}_4) \delta_D(\mathbf{k}_2 + \mathbf{k}_3) P(\mathbf{k}_1) P(\mathbf{k}_2) . \end{aligned} \quad (2.42)$$

With Wick's theorem, it is easy to see that the 2PCF is a full description of Gaussian information. As the universe evolves, linear perturbation theory is not enough to explain the growth of structures, some non-linear perturbation theory has to be utilized. In the regime of standard perturbation theory (SPT), the non-linear field is expanded as:

$$\delta_m(\mathbf{k}) = \delta^{(1)}(\mathbf{k}) + \delta^{(2)}(\mathbf{k}) + \dots + \delta^{(n)}(\mathbf{k}) , \quad (2.43)$$

where the source terms for $\delta^{(n)}$ involve n power of the linear fields, and so each term in series Eq. (2.43) is smaller than the previous one. Since $\delta^{(1)}$ is a Gaussian density field, then δ_m is no longer a Gaussian field with non-zero additional terms included. The coupling between the n th order and linear density field is given by:

$$\delta^{(n)}(\mathbf{k}) = \left[\prod_{i=1}^n \int \frac{d^3 k_i}{(2\pi)^3} \right] (2\pi)^3 \delta_D\left(\mathbf{k} - \sum_{i=1}^n \mathbf{k}_i\right) \times F_n(\mathbf{k}_1, \dots, \mathbf{k}_n) \delta^{(1)}(\mathbf{k}_1) \dots \delta^{(1)}(\mathbf{k}_n) , \quad (2.44)$$

where $F_n(\mathbf{k}_1, \dots, \mathbf{k}_n)$ is the kernel connecting n linear density field with different wavelength. The important effect of non-linear evolution is that statistics involving an odd

number of matter density fields no longer vanish. Using only two terms in Eq. (2.43), then the expansion of bispectrum could be written as:

$$\begin{aligned}
 \langle \delta_m(\mathbf{k}_1)\delta_m(\mathbf{k}_2)\delta_m(\mathbf{k}_3) \rangle &= \langle (\delta^{(1)}(\mathbf{k}_1) + \delta^{(2)}(\mathbf{k}_1)) (\delta^{(1)}(\mathbf{k}_2) + \delta^{(2)}(\mathbf{k}_2)) (\delta^{(1)}(\mathbf{k}_3) + \delta^{(2)}(\mathbf{k}_3)) \rangle \\
 & \quad (2.45) \\
 &= \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(1)}(\mathbf{k}_3) \rangle + \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle \\
 &+ \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(2)}(\mathbf{k}_2)\delta^{(1)}(\mathbf{k}_3) \rangle + \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(2)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle \\
 &+ \langle \delta^{(2)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(1)}(\mathbf{k}_3) \rangle + \langle \delta^{(2)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle \\
 &+ \langle \delta^{(2)}(\mathbf{k}_1)\delta^{(2)}(\mathbf{k}_2)\delta^{(1)}(\mathbf{k}_3) \rangle + \langle \delta^{(2)}(\mathbf{k}_1)\delta^{(2)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle , \quad (2.46)
 \end{aligned}$$

where the first term in Eq. (2.46) vanishes according to Wick's theorem, the second, third and fourth terms are the leading contributions to the bispectrum, and the rest are higher order terms whose contribution is less. Since the three leading terms are cyclic permutation, we will choose the first one to do further calculation.

$$\begin{aligned}
 \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle &= \left\langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2) \int \frac{d^3q}{(2\pi)^3} \int \frac{d^3l}{(2\pi)^3} \right. \\
 & \quad (2\pi)^3 \delta_D(\mathbf{k}_3 - \mathbf{q} - \mathbf{l}) F_2(\mathbf{q}, \mathbf{l}) \delta^{(1)}(\mathbf{q})\delta^{(1)}(\mathbf{l}) \rangle \quad (2.47)
 \end{aligned}$$

$$\begin{aligned}
 &= \int \frac{d^3q}{(2\pi)^3} \int d^3l \delta_D(\mathbf{k}_3 - \mathbf{q} - \mathbf{l}) \\
 & \quad F_2(\mathbf{q}, \mathbf{l}) \langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(1)}(\mathbf{q})\delta^{(1)}(\mathbf{l}) \rangle \quad (2.48)
 \end{aligned}$$

$$\begin{aligned}
 &= \int \frac{d^3q}{(2\pi)^3} \int d^3l \delta_D(\mathbf{k}_3 - \mathbf{q} - \mathbf{l}) F_2(\mathbf{q}, \mathbf{l}) \\
 & \quad \times ((2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{k}_2) \delta_D(\mathbf{q} + \mathbf{l}) P(\mathbf{k}_1) P(\mathbf{q})) \\
 & \quad + (2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{q}) \delta_D(\mathbf{k}_2 + \mathbf{l}) P(\mathbf{k}_1) P(\mathbf{k}_2) \\
 & \quad + (2\pi)^6 \delta_D(\mathbf{k}_1 + \mathbf{l}) \delta_D(\mathbf{k}_2 + \mathbf{q}) P(\mathbf{k}_1) P(\mathbf{k}_2)) , \quad (2.49)
 \end{aligned}$$

where we use the expansion relation with F_2 kernel, shown in Eq. (2.50), from Eq. (2.47) to Eq. (2.48), and we utilize Wick's theorem from Eq. (2.48) to Eq. (2.49).

$$F_2(\mathbf{k}_1, \mathbf{k}_2) = \frac{5}{7} + \frac{2(\mathbf{k}_1 \cdot \mathbf{k}_2)^2}{7 k_1^2 k_2^2} + \frac{1}{2} \mathbf{k}_1 \cdot \mathbf{k}_2 \left(\frac{k_1}{k_2} + \frac{k_2}{k_1} \right) . \quad (2.50)$$

As for Eq. (2.49), we could further simplify it by integrating over \mathbf{q} and \mathbf{l} . Then for the first term in Eq. (2.49), we have $\mathbf{q} = -\mathbf{l}$, which makes $F_2(\mathbf{l}, -\mathbf{l})$ vanish. Integrating the second and third term, we finally get:

$$\langle \delta^{(1)}(\mathbf{k}_1)\delta^{(1)}(\mathbf{k}_2)\delta^{(2)}(\mathbf{k}_3) \rangle = (2\pi)^3 \delta_D^{(3)}(\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3) 2F_2(\mathbf{k}_1, \mathbf{k}_2) P_L(\mathbf{k}_1) P_L(\mathbf{k}_2) , \quad (2.51)$$

where P_L refers the linear power spectrum. In the end, Eq. (2.45) is expressed as:

$$\begin{aligned}
 \langle \delta_m(\mathbf{k}_1, \eta)\delta_m(\mathbf{k}_2, \eta)\delta_m(\mathbf{k}_3, \eta) \rangle &= (2\pi)^3 \delta_D^{(3)}(\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3) \\
 & \quad \times [2F_2(\mathbf{k}_1, \mathbf{k}_2) P_L(\mathbf{k}_1, \eta) P_L(\mathbf{k}_2, \eta) + 2 \text{ perm.}] , \quad (2.52)
 \end{aligned}$$

where ‘‘perm’’ indicates cyclic permutation. The above derivation and equation show the existence of non-Gaussian matter density field mathematically. This highlights the limitations of Gaussian-based statistics, and motivates the use of higher-order estimators and nonlinear summary statistics to capture the full cosmological information content.

2.4 Angular power spectrum

Instead of \mathbf{r} , the angular separation vector $\boldsymbol{\theta}$ (also expressed as (θ, ϕ)) is typically used as the displacement vector for correlation functions in actual observations conducted on the celestial sphere. In analogy with the Fourier expansion in 3D space, we separate out the contributions of different angular scales by doing a multipole expansion,

$$\delta(\boldsymbol{\theta}, \phi) = \sum_{\ell \geq 0} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\boldsymbol{\theta}, \phi), \quad (2.53)$$

where ℓ is the multipole number and it takes only discrete integers, and m represents different phases with given separation scales ℓ . Functions $Y_{\ell m}(\boldsymbol{\theta}, \phi)$ are *spherical harmonics*, which form an orthonormal set of functions over the sphere, so that we can calculate the multipole coefficients $a_{\ell m}$ from

$$a_{\ell m} = \int_0^\pi \int_0^{2\pi} Y_{\ell m}^*(\boldsymbol{\theta}, \phi) \delta(\boldsymbol{\theta}, \phi) d\Omega. \quad (2.54)$$

Spherical harmonics functions are related to the *associated Legendre functions* $P_\ell^m(x)$ by

$$Y_{\ell m}(\boldsymbol{\theta}, \phi) = (-1)^m \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta) e^{im\phi}. \quad (2.55)$$

Thus, the θ -dependence is in $P_\ell^m(\cos\theta)$ and the ϕ -dependence is in $e^{im\phi}$. The functions P_ℓ^m are real and

$$Y_{\ell, -m} = (-1)^m Y_{\ell, m}^*, \quad (2.56)$$

so that

$$Y_{\ell 0} = \sqrt{\frac{2\ell+1}{4\pi}} P_\ell(\cos\theta), \quad (2.57)$$

is also real. The functions $P_\ell \equiv P_\ell^0$ are called *Legendre polynomials*. Summing over the m corresponding to the same multipole number ℓ gives the *addition theorem* [16],

$$\sum_m Y_{\ell m}^*(\boldsymbol{\theta}', \phi') Y_{\ell m}(\boldsymbol{\theta}, \phi) = \frac{2\ell+1}{4\pi} P_\ell(\cos\vartheta), \quad (2.58)$$

where ϑ is the angle between $\hat{\mathbf{n}} = (\theta, \phi)$ and $\hat{\mathbf{n}}' = (\theta', \phi')$, i.e., $\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}' = \cos\vartheta$.

The quantity we want to calculate from theory is the variance $\langle |a_{\ell m}|^2 \rangle$ to get a prediction for a typical size of the $a_{\ell m}$. From statistical isotropy, it also follows that these expectation

values depend only on ℓ and not on m . The ℓ values are associated with the angular scale of the anisotropy pattern, while the m values pertain to its ‘‘orientation’’ or ‘‘pattern’’. In order to get the scale information, we could calculate the mean value of $\langle |a_{\ell m}|^2 \rangle$ at different orientations with given scale,

$$C_\ell \equiv \langle |a_{\ell m}|^2 \rangle = \frac{1}{2\ell + 1} \sum_m \langle |a_{\ell m}|^2 \rangle, \quad (2.59)$$

and altogether we have

$$\langle a_{\ell m}^* a_{\ell' m'} \rangle \equiv \delta_{\ell\ell'} \delta_{mm'} C_\ell, \quad (2.60)$$

where $\delta_{mm'}$ and $\delta_{\ell\ell'}$ are Kronecker delta functions, and C_ℓ is called *angular power spectrum*. It is analogous to the power spectrum of density perturbations.

Additionally, the definition of angular correlation function could be derived using angular power spectrum, similarly as the power spectrum and 2 point correlation function in 3D space:

$$\xi(\vartheta) = \langle \delta(\boldsymbol{\theta}_1) \delta(\boldsymbol{\theta}_2) \rangle \quad (2.61)$$

$$= \sum_{\ell m} \sum_{\ell' m'} \langle a_{\ell m}^* a_{\ell' m'} \rangle Y_{\ell m}^*(\boldsymbol{\theta}_1) Y_{\ell' m'}(\boldsymbol{\theta}_2) \quad (2.62)$$

$$= \sum_{\ell} C_\ell \sum_m Y_{\ell m}^*(\boldsymbol{\theta}_1) Y_{\ell m}(\boldsymbol{\theta}_2) \quad (2.63)$$

$$= \sum_{\ell} \frac{2\ell + 1}{4\pi} C_\ell P_\ell(\cos \vartheta), \quad (2.64)$$

where we expand density perturbation using spherical harmonics in Eq. (2.62), then use the definition of angular power spectrum to simplify the summation in Eq. (2.63) and finally apply addition theorem in Eq. (2.64) to get the final mathematical relation between angular power spectrum and angular correlation function.

Chapter 3

Weak lensing cosmology

All observed light from distant galaxies is bent because of the inhomogeneity caused by the ubiquitous density fluctuations at large scales according to the theory of general relativity. These fluctuations create a tidal gravitational field that causes light bundles to be deflected differentially. This is called gravitational lensing effect. As a result, images of light-emitting galaxies that we observe are distorted. The direction and amount of distortion is directly related to the size and shape of the matter distribution projected along the line of sight, providing a powerful tool for us to study the matter distribution. In this chapter, we are going to introduce the concept and formalism of gravitational weak lensing. In Section 3.1, we will introduce some general concepts about weak lensing. In Section 3.2, we will show the mathematical formalism of weak gravitational lensing, and how convergence and shear are defined. In Section 3.3, we will present the physical meaning of convergence field.

3.1 Gravitational lensing basic

Gravitational lensing effect occurs when the gravitational field of a massive object, such as a galaxy or galaxy clusters, bends the path of light coming from a more distant source. This phenomenon is a direct consequence of Einstein's General Relativity, which predicts that massive objects warp the spacetime around them, causing light to follow curved trajectories. Fig. 3.1 illustrates a lensing system with only one source and lensing object, where it shows how a lensed image appears at a different angular position in the sky compared to their actual position.

Besides the shift of position of lensed image on the sky, the shape and size of the source would be changed as well, which is called *distortion*. Distortion elongates the images in one direction and compresses it in another direction and also adds some shear to the original shape. Meanwhile, the observed images are magnified as well due to the distortion effect. In the next subsection, we will show the distortion effect mathematically. Based on the magnitude of distortion, gravitational lensing is divided into *strong lensing*, *weak lensing* and *microlensing*.

- **Strong lensing:** is a gravitational lensing effect that is strong enough to produce multiple images, arcs, or Einstein rings. For this to occur, the projected lens mass must be greater than the critical density, Σ_{cr} .

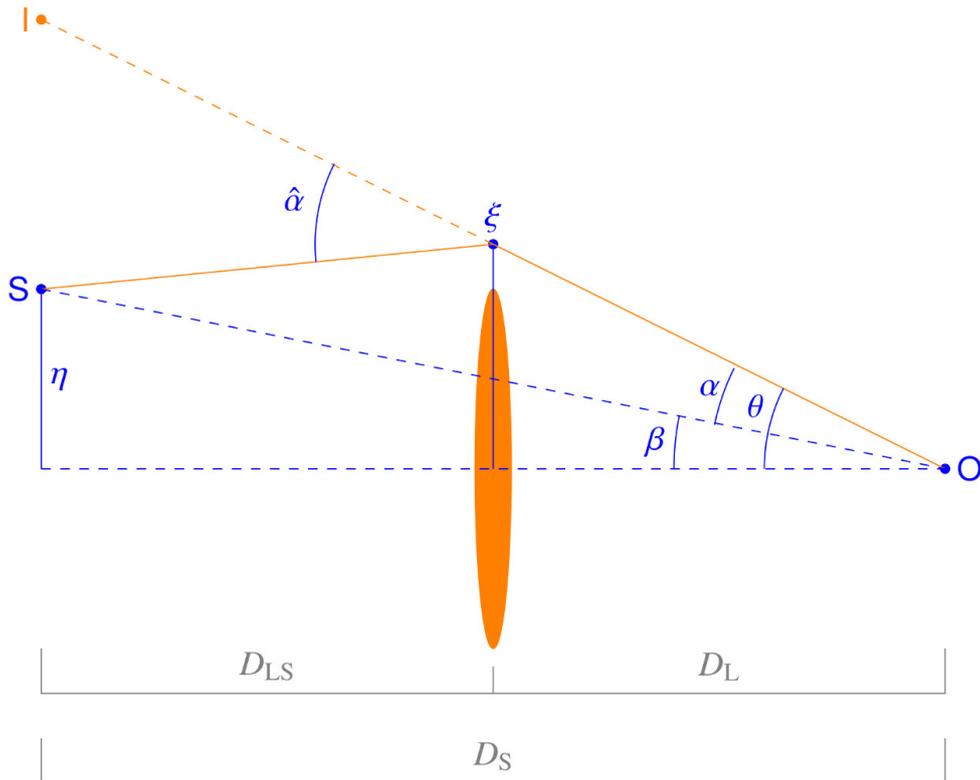


Figure 3.1: Lensing system illustration. Point I , S and O denote deflected image, source and observer. In the bottom, D_S , D_{ds} and D_d refer the distance between source and observer, source and lens, lens and observer. Light emitted from a source located at a distance η from the central optical axis and traveling parallel to it gets deflected by an angle α before reaching the observer. The resulting image appears at an angular separation θ from the optical axis in the sky, which differs from the actual separation of the source, β . Image from [17].

- **Microlensing:** occurs when the lensing object is much smaller in mass, such as a star or a planet. In microlensing, the lens does not produce multiple distinct images due to the small angular separation, but it temporarily magnifies the light from a background source as the lens passes between the observer and the source. Microlensing is often used to detect objects that emit little or no light themselves, such as exoplanets, brown dwarfs, or black holes.
- **Weak lensing:** takes place when the gravitational field of the lensing object is not strong enough to produce multiple images or significant arcs but still distorts the shapes of background galaxies slightly. These distortions are so small that they cannot be easily detected in individual galaxies. Instead, the effect is measured statistically by analyzing the collective distortions of many galaxies across a wide area of the sky.

Before going to the mathematical description of gravitational lensing, we should remember that Fig. 3.1 is just a schematic illustration of effect of only one lens on the source galaxy. But in reality, light emitted from the source experiences a lot of lenses, which results in a continuous deflection when the light travels to observers. And then the accumulated result contains the bending from all the matter along the propagation of light.

3.2 Lensing formalism

In order to quantify gravitational lensing, it is necessary to account for light propagation in an inhomogeneous universe. For a general metric that represents an expanding universe with first-order perturbations, the line element ds is expressed as:

$$ds^2 = (1 + 2\Psi) dt^2 - a^2(t) (1 - 2\Phi) dl^2 , \quad (3.1)$$

where spatial part of the metric is given by the comoving coordinate l , two Bardeen gravitational potentials Ψ and Φ are used to describe weak field, $\Psi, \Phi \ll 1$. The potential of a lens with mass M and radius R can be approximated by

$$\frac{GM}{R} = \frac{1}{2} \frac{R_S}{R} , \quad (3.2)$$

where R_S is the Schwarzschild radius. The weak-field condition, Eq. (3.2), is satisfied for the majority of mass distributions, except for extremely compact objects where the size R is nearly equal to their Schwarzschild radius. In General Relativity, and in the absence of anisotropic stress which is the case on large scales, the two potentials are equal, $\Phi = \Psi$. The density contrast is related to gravitational potential via Poisson equation, representing in comoving coordinate as

$$\nabla^2 \Phi = 4\pi G a^2 \bar{\rho} \delta . \quad (3.3)$$

Photons propagating on the null geodesics gives a vanishing line element ds . Then according to Eq. (3.1), we could get:

$$t = \int (1 - 2\Phi) dr , \quad (3.4)$$

where the integral is along the light path in **physical** or **proper** coordinate dr . Similar to geometrical optics, the potential functions like a medium with a refractive index of $n = 1 - 2\Phi$ (where $\Phi < 0$), altering the direction of the light's trajectory. Using Fermat's principle by setting $\delta t = 0$, we get Euler-Lagrange equation for the refractive index:

$$\frac{d}{dr} \left(\frac{\partial n}{\partial r'} \right) - \frac{\partial n}{\partial r} = 0 . \quad (3.5)$$

Integrating the equations along the light path gives the total *deflection angle* $\hat{\alpha}$ defined as the difference between the directions of emitted and received light rays:

$$\hat{\alpha} = -2 \int \nabla_{\perp}^p \Phi dr , \quad (3.6)$$

where the gradient ∇_{\perp}^p is taken perpendicular to the light path, with respect to the physical coordinates. With this, we could quantify the connection between light deflection and gravitational potential on cosmological scale. To describe how rays within an infinitesimally narrow light bundle propagate, we consider the separation between two nearby geodesics, which is governed by geodesic deviation equation. In a homogeneous FLRW Universe, the transverse comoving separation \mathbf{x}_0 between two light rays, as a function of χ from observer, is proportional to the comoving angular distance:

$$\mathbf{x}_0(\chi) = f_K(\chi) \boldsymbol{\theta} , \quad (3.7)$$

where the separation vector \mathbf{x}_0 is seen by the observer under the angle $\boldsymbol{\theta}$. [18]. While in perturbed Universe, the light is deflected by an amount $d\hat{\alpha} = -2\nabla_{\perp} \Phi(\mathbf{x}, \chi') d\chi'$ in the presence of potential Φ at distance χ' from the observer (expressed in comoving coordinate). When deflector exits, the change in the separation vector at the source's comoving distance χ is given by $d\mathbf{x} = f_K(\chi - \chi') d\hat{\alpha}$ (see Fig. 3.2 for an illustration). The total separation is determined by the integrating along the line of sight over χ' . Lensing deflection changes the path of both light rays, and we denote the fiducial ray with the superscript (0) . Then the separation at distance χ between two light rays is observed angular diameter distance minus the deflected angle multiplied by the traveled distance:

$$\mathbf{x}(\chi) = f_K(\chi) \boldsymbol{\theta} - 2 \int_0^{\chi} d\chi' f_K(\chi - \chi') [\nabla_{\perp} \Phi(\mathbf{x}, \chi') - \nabla_{\perp} \Phi^{(0)}(\chi')] . \quad (3.8)$$

Without lensing, the observer would see the separation vector \mathbf{x} under an angle $\boldsymbol{\beta} = \mathbf{x}(\chi) / f_K(\chi)$. The difference between the apparent angle $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ represents the total, *scaled deflection angle* $\boldsymbol{\alpha}$, which defines the lens equation:

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \boldsymbol{\alpha} , \quad (3.9)$$

with

$$\boldsymbol{\alpha} = 2 \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi')}{f_K(\chi)} [\nabla_{\perp} \Phi(\mathbf{x}, \chi') - \nabla_{\perp} \Phi^{(0)}(\chi')] . \quad (3.10)$$

However, the integral of Eq. (3.10) over the actual deflected light path might be complicated. Since the typical deflection angles are on the order of seconds or smaller [20],

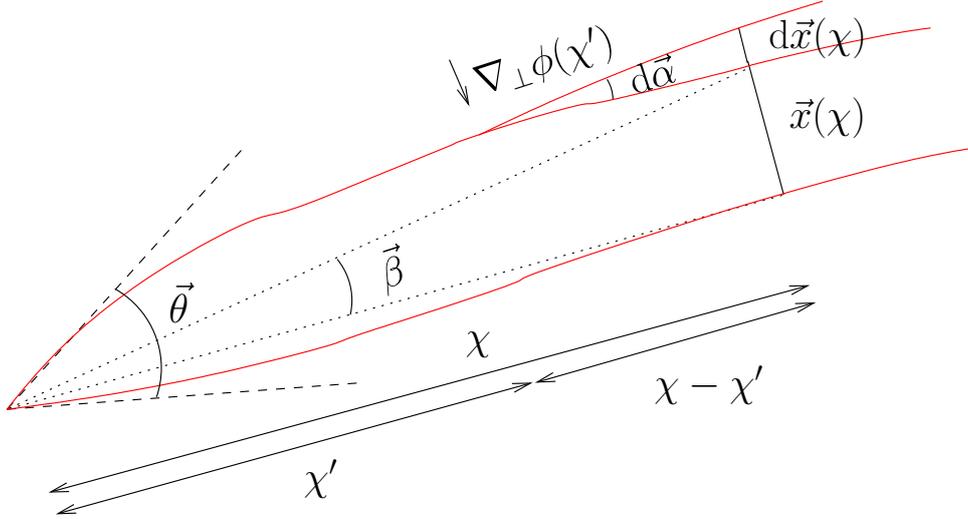


Figure 3.2: Two light rays (represented by red solid lines) are shown converging toward an observer on the left. The rays are separated by a transverse comoving distance $\mathbf{x}(\chi)$, which changes with the distance ξ from the observer. A deflector located at distance ξ' disturbs the geodesics in proportion to the transverse gradient $\nabla_{\perp}\phi$ of the potential. The dashed lines represent the apparent paths of the light rays, which appear to converge at an angle θ to the observer. The dotted lines illustrate the unperturbed geodesics, defining the angle β , at which the unperturbed transverse comoving distance \mathbf{x} is observed. Image from [19].

the integration path could be approximated by an unperturbed straight light path in the absence of lensing, i.e. directly over θ instead of β . This approximation is known as *Born approximation*, which greatly simplifies the calculation by neglecting higher-order corrections. The Born approximation is valid in the weak lensing regime, where the deflections are small, and the light rays do not deviate significantly from their original trajectories. However, for strong lensing or highly nonlinear structures, corrections beyond the Born approximation may be necessary to account for the cumulative effects of multiple deflections.

We linearize the lens equation Eq. (3.9) and introduce the (inverse) amplification matrix, defined as the Jacobian $\boldsymbol{\beta} = \mathbf{x}(\chi) / f_K(\chi)$, which describes the linear relationship between the lensed (image) coordinates $\boldsymbol{\theta}$ and the unlensed (source) coordinates $\boldsymbol{\beta}$,

$$\begin{aligned} A_{ij} &= \frac{\partial \beta_i}{\partial \theta_j} = \delta_{ij} - \frac{\partial \alpha_i}{\partial \theta_j} \\ &= \delta_{ij} - 2 \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \frac{\partial^2}{\partial x_i \partial x_j} \Phi(f_K(\chi') \boldsymbol{\theta}, \chi'), \end{aligned} \quad (3.11)$$

where the second term in Eq. (3.10) drops out because it is independent on $\boldsymbol{\theta}$. The second term in Eq. (3.11) represents the integral of the gradient of the three-dimensional gravitational potential in the transverse direction, taken along the line of sight. This results in the gradient of the projected gravitational potential in two-dimensional space.

Thus, the deflection angle can be written as the gradient of a 2D potential, called the *lensing potential* ψ ,

$$A_{ij} = \delta_{ij} - \partial_i \partial_j \psi , \quad (3.12)$$

where the partial derivatives are understood with respect to $\boldsymbol{\theta}$ and the expression of ψ is given as

$$\psi(\boldsymbol{\theta}, \chi) = 2 \int_0^\chi d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \Phi(f_K(\chi') \boldsymbol{\theta}, \chi') . \quad (3.13)$$

The symmetric matrix \mathbf{A} is expressed in terms of the scalar *convergence* κ and the two-component spin-two *shear*, $\gamma = (\gamma_1, \gamma_2)$ as follows:

$$\mathbf{A} = \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix} , \quad (3.14)$$

where the convergence κ and shear γ are defined as the second derivatives of the lensing potential:

$$\kappa = \frac{1}{2} (\partial_1 \partial_1 + \partial_2 \partial_2) \psi = \frac{1}{2} \nabla_\perp^2 \psi , \quad (3.15)$$

$$\gamma_1 = \frac{1}{2} (\partial_1 \partial_1 - \partial_2 \partial_2) \psi , \quad (3.16)$$

$$\gamma_2 = \partial_1 \partial_2 \psi . \quad (3.17)$$

Convergence, being the diagonal component of the matrix, represents an isotropic enlargement or reduction of the observed size of a source image. In contrast, shear, which is the trace-free part, describes anisotropic stretching, transforming a circular light distribution into an elliptical one. In the case of cosmological lensing by large-scale structures, images experience very weak lensing, with κ and γ values being only a few percent or smaller. Each source corresponds to a single, unique image, with no occurrence of multiple images, and the matrix \mathbf{A} is invertible. Based on Eq. (3.11), we could see that:

$$\delta \boldsymbol{\theta} = \mathbf{A}^{-1} \delta \boldsymbol{\beta} , \quad (3.18)$$

where \mathbf{A}^{-1} is the inverse of the Jacobian matrix, describing the local mapping of the source light distribution to image coordinates, which is represented as:

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{pmatrix} 1 - \kappa + \gamma_1 & \gamma_2 \\ \gamma_2 & 1 - \kappa - \gamma_1 \end{pmatrix} , \quad (3.19)$$

where $\det(\mathbf{A}) = (1 - \kappa)^2 - \gamma^2$, with $\gamma^2 = \gamma_1^2 + \gamma_2^2$, is the Jacobi determinant. Mathematically, shear is also written as a complex number

$$\gamma = \gamma_1 + i\gamma_2 = |\gamma| \exp(2i\varphi) , \quad (3.20)$$

where γ is the modulus of the shear and φ is the orientation angle of the shear. The prefactor in Eq. (3.19) shows that the solid angle covered by the image differs from the solid angle subtended by the source by the *magnification factor*

$$\mu = \frac{1}{\det \mathbf{A}} = \frac{1}{(1 - \kappa)^2 - \gamma^2} \approx 1 + 2\kappa , \quad (3.21)$$

where the final approximation is a first-order Taylor expansion. Therefore, in the weak lensing regime, the magnification of an image is primarily governed by the convergence κ (to first-order in a Taylor expansion), rather than by the shear. When $\kappa > 0$, the observed image is magnified (shown in Fig. 3.3), whereas when $\kappa < 0$, the observed image appears smaller than the source image. Since κ only affects the size of the image rather than the shape of the source, we could factor out $(1 - \kappa)$ from Eq. (3.14). Cosmic shear is based on the measurement of galaxy shapes and the real observable is not shear but *reduced shear*:

$$g = \frac{\gamma}{1 - \kappa}. \quad (3.22)$$

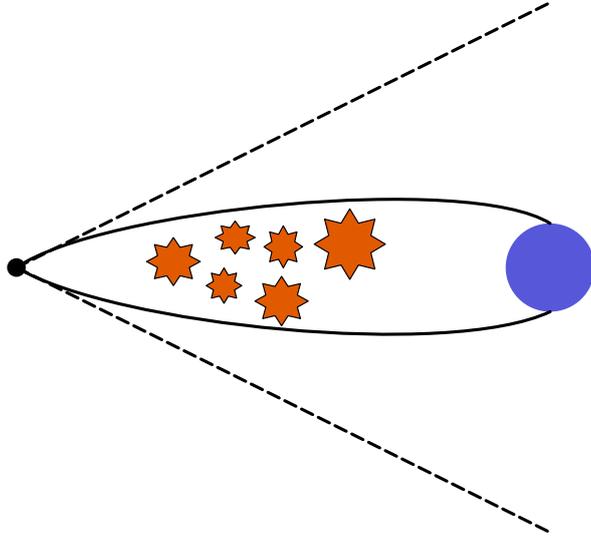


Figure 3.3: Illustration of magnification. This is the case of convergence that is greater than zero. Orange objects represent the foreground galaxies and purple circle is the source galaxy. Solid line shows the real light path and dashed line indicates the observed image.

Regarding the effect of shear on observations, Eq. (3.19) shows that when only the γ_1 term is present ($\gamma_2 = 0$), a circular source image is stretched in one direction and compressed in the perpendicular direction. However, when $\gamma_1 = 0$ and $\gamma_2 \neq 0$, the source image is rotated and stretched within this coordinate system, as depicted in Fig. 3.4.

The total distortion effects, with convergence and shear, described by Eq. (3.14) are illustrated schematically in Fig. 3.5, where we demonstrate how gravitational lensing distorts the image of a source galaxy, which is assumed to have an intrinsically circular shape when projected onto the sky.

3.3 Projected overdensity

Since convergence κ is linked to the lensing potential ψ through a 2D Poisson equation, Eq. (3.15), it can be understood as a (projected) surface density. To prove it mathematically, let's start with the 2D Poisson equation, $\nabla_{\perp}^2 \psi$. If we replace the perpendicular

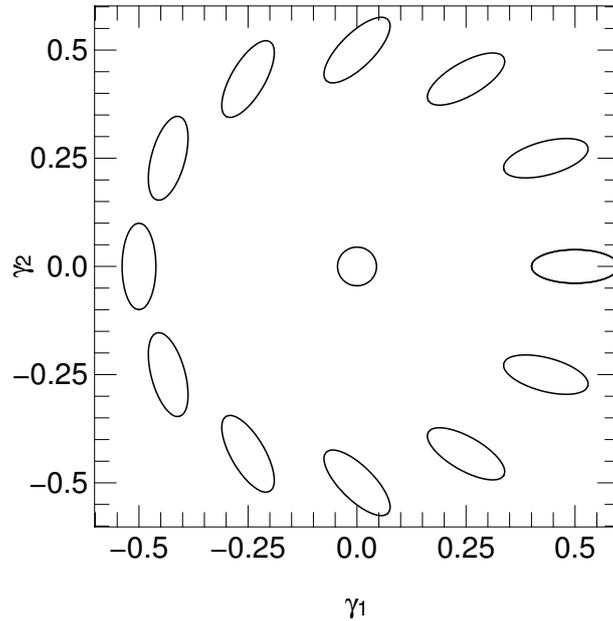


Figure 3.4: The orientation of the shear-induced ellipses in gravitational lensing, represented by the Cartesian components γ_1 and γ_2 . As the polar angle φ varies from 0 to 2π , the orientation of the shear ellipses rotates by π , illustrating the directional stretching and compression effects on background source images due to the lensing shear. Image from [19].

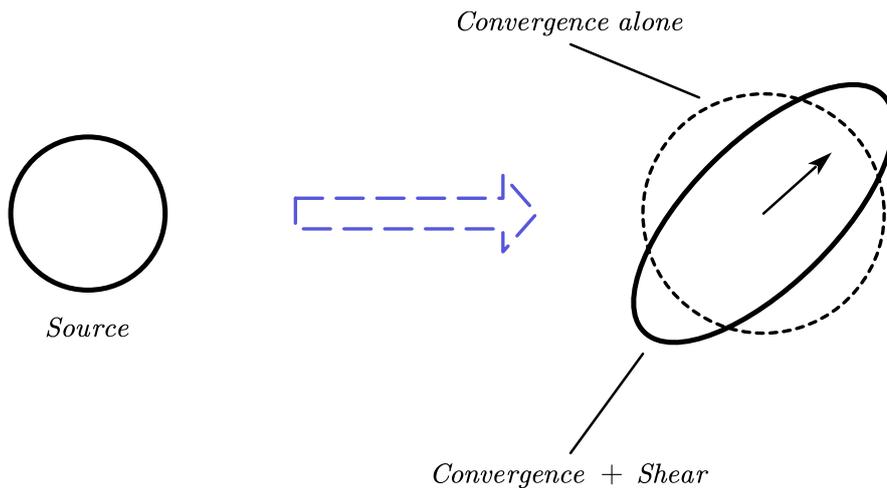


Figure 3.5: Total lensing distortion effects: For a galaxy with an intrinsically circular shape, convergence will isotropically increase its size, while shear will distort it into an elliptical shape, introducing anisotropy. Image adapted from [17].

Laplacian ∇_{\perp}^2 with the complete Laplacian (**in comoving coordinate**),

$$\nabla^2 = \nabla_{\perp}^2 + \frac{\partial^2}{\partial \chi^2} . \quad (3.23)$$

We could insert Eq. (3.23) into Eq. (3.15), resulting in

$$\kappa(\boldsymbol{\theta}, \chi) = \frac{1}{2} \left(\nabla^2 - \frac{\partial^2}{\partial \chi^2} \right) \psi \quad (3.24)$$

$$= \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \nabla^2 \Phi(f_K(\chi') \boldsymbol{\theta}, \chi') \quad (3.25)$$

$$- \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \frac{\partial^2}{\partial \chi^2} \Phi(f_K(\chi') \boldsymbol{\theta}, \chi') . \quad (3.26)$$

As the positive and negative contributions approximately cancel each other out when integrated along the line of sight, we have the integral, Eq. (3.26), equal to zero

$$\int \frac{\partial^2 \Phi}{\partial \chi^2} d\chi = \frac{\partial \Phi}{\partial \chi} \Big|_{\text{end points}} = 0 . \quad (3.27)$$

Thus, the convergence κ could be formulated using the density contrast by taking Poisson equation Eq. (3.3) into Eq. (3.25):

$$\begin{aligned} \kappa(\boldsymbol{\theta}, \chi) &= \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \nabla^2 \Phi(f_K(\chi') \boldsymbol{\theta}, \chi') \\ &= 4\pi G \int_0^{\chi} d\chi' \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} a^2 \bar{\rho} \delta(f_K(\chi') \boldsymbol{\theta}, \chi') . \end{aligned} \quad (3.28)$$

To deal with $\bar{\rho}$ in above equation, we utilize first Friedmann equation:

$$\begin{aligned} \bar{\rho}(t) &= \Omega_m(t) \rho_{\text{crit}}(t) = \Omega_m(t) \frac{3H^2(t)}{8\pi G} \\ &= \Omega_{m,0} \frac{3H_0^2}{8\pi G} a^{-3}(t) , \end{aligned} \quad (3.29)$$

where H_0 and $\Omega_{m,0}$ is the present value of Hubble parameter and matter density parameter. Therefore, the equation connecting the 2D surface-mass density and 3D matter density contrast is

$$\kappa(\boldsymbol{\theta}, \chi) = \frac{3H_0^2 \Omega_{m,0}}{2} \int_0^{\chi} \frac{d\chi'}{a(\chi')} \frac{f_K(\chi - \chi') f_K(\chi')}{f_K(\chi)} \delta(f_K(\chi') \boldsymbol{\theta}, \chi') . \quad (3.30)$$

This formulation represents the projection of the density at position χ to the observers along comoving coordinates, influenced by geometrical factors related to the distances between the source, deflector, and observer. The mean convergence from a population of source galaxies is calculated by weighting the above expression with the galaxy *probability distribution* in comoving distance, $n(\chi) d\chi$,

$$\kappa(\boldsymbol{\theta}) = \int_0^{\chi_{\text{lim}}} d\chi n(\chi) \kappa(\boldsymbol{\theta}, \chi) , \quad (3.31)$$

The integral extends out to the limiting comoving distance χ_{lim} of galaxy sample. The physical meaning of this equation indicates all the matter density contrast at position $\boldsymbol{\theta}$ with distance $[0, \chi_{lim}]$ to the observer are projected to that point. By inserting Eq. (3.30) into Eq. (3.31) and interchanging the integral order, we get the following expression,

$$\kappa(\boldsymbol{\theta}) = \frac{3H_0^2\Omega_{m,0}}{2} \int_0^{\chi_{lim}} \frac{d\chi}{a(\chi)} q(\chi) f_K(\chi) \delta(f_K(\chi)\boldsymbol{\theta}, \chi), \quad (3.32)$$

where $q(\chi)$ is *lens efficiency*, defined as

$$q(\chi) = \int_{\chi}^{\chi_{lim}} d\chi' n(\chi') \frac{f_K(\chi' - \chi)}{f_K(\chi')}. \quad (3.33)$$

The lens efficiency function $q(\chi)$ describes how effective a matter overdensity at comoving distance χ is at lensing the background galaxies. It weights the contribution of matter at different distances to the observed convergence $\kappa(\boldsymbol{\theta})$. Intuitively, if the mass distribution is closer to the observer, it has less lensing effect while if it's closer to the background galaxies, it has a stronger lensing effect. Zero efficiency represents the matter is beyond the source galaxy.

3.4 Shear measurement

The eigenvalues of the inverse Jacobi matrix, Eq. (3.19), are

$$\lambda_{\pm} = \frac{1 - \kappa \pm \lambda}{\det \mathbf{A}} = \frac{1}{1 - \kappa \mp \gamma}. \quad (3.34)$$

A hypothetical circular source, when subjected to weak gravitational lensing, is distorted into an ellipse with semi-major and semi-minor axes, denoted as a and b , which correspond to the proportionality of the eigenvalues λ_{\pm} . By the common definition, the *ellipticity* ε of such an image is

$$\varepsilon \equiv \frac{a - b}{a + b} = \frac{\lambda_+ - \lambda_-}{\lambda_+ + \lambda_-} = \frac{\gamma}{1 - \kappa}, \quad (3.35)$$

which just the reduced shear. In the cases of weak lensing where $\kappa \ll 1$, $\varepsilon = g \approx \gamma$.

In observations of weak lensing, however, the sources are usually not circular but have an inherent source elliptical shape, ε_S . Cosmic shear modifies this ellipticity as a function of the complex reduced shear. Assuming no further distortions arise from observational systematic errors, the relationship between the observed ellipticity, ε and the source ellipticity, ε_S is given by [21],

$$\varepsilon = \frac{\varepsilon_S + g}{1 + g^* \varepsilon_S}, \quad (3.36)$$

where the asterisk “*” denotes complex conjugation. And also in weak lensing regime, this relation is approximated by

$$\varepsilon \approx \varepsilon_S + g, \quad (3.37)$$

When the intrinsic ellipticity of galaxies has no preferred orientation, the expectation value of ε_S becomes zeros, $\langle \varepsilon_S \rangle = 0$. As a result, the observed ellipticity serves an unbiased estimator of the reduced shear, with

$$\langle \varepsilon \rangle = g . \quad (3.38)$$

3.5 Kaiser-Squires relation

In real observations, convergence cannot be observed since we have no way to know the intrinsic size and then compare the magnification with the observed one. The only thing we can observe is the shape of the field. Since both convergence and shear are the second order derivatives with respect to the lensing potential, the relation between them must exist. And this relation is called, *Kaiser-Squires relation* [22]. And the mathematical expression is given by [23]:

$$\tilde{\gamma}(\boldsymbol{\ell}) = \frac{(\ell_1 + i\ell_2)^2}{\ell^2} \tilde{\kappa}(\boldsymbol{\ell}) = e^{2i\beta} \tilde{\kappa}(\boldsymbol{\ell}) . \quad (3.39)$$

In the case of full sky maps, we have to use the spherical harmonics to represent the convergence and shear. Since convergence maps are scalar fields, we could use the scalar spherical harmonics to represent them, which could be expressed as:

$$\begin{aligned} \kappa &= \frac{1}{2} \nabla^2 \psi \\ &= \frac{1}{2} \nabla^2 \sum_{\ell, m} \psi_{\ell, m} Y_{\ell, m} \\ &= \frac{1}{2} \sum_{\ell, m} -\ell(\ell+1) \psi_{\ell, m} Y_{\ell, m} , \end{aligned} \quad (3.40)$$

where we used the relation $\nabla^2 Y_{\ell, m} = -\ell(\ell+1) Y_{\ell, m}$. The shear is a spin-2 field, so we could use the spin-2 spherical harmonics to represent it. The shear could be expressed as:

$$\begin{aligned} \gamma &= \gamma_1 + i\gamma_2 \\ &= \partial_1 \partial_1 \psi - \partial_2 \partial_2 \psi + 2i\partial_1 \partial_2 \psi \\ &= \frac{1}{2} \bar{\partial} \bar{\partial} \psi \\ &= \frac{1}{2} \bar{\partial} \bar{\partial} \sum_{\ell, m} \psi_{\ell, m} {}_2 Y_{\ell, m} \\ &= \frac{1}{2} \sum_{\ell, m} \sqrt{\ell(\ell+2)(\ell+1)(\ell-1)} \psi_{\ell, m} Y_{\ell, m} , \end{aligned} \quad (3.41)$$

where $\bar{\partial}$ is the differential operator used to lower the spin weight, and we have used the relation $\bar{\partial}({}_s Y_{\ell, m}) = -\sqrt{(\ell+s)(\ell-s+1)} {}_{s-1} Y_{\ell, m}$ and $\psi_{\ell, m}$ are the coefficients under spherical harmonics expansion at the basis of $Y_{\ell, m}$. For further derivation, we could see in

Appendix D. Joining Eq. (3.40) and Eq. (3.41) into Eq. (3.39), we could get the relation between the convergence and shear in spherical harmonics representation:

$$\gamma_{\ell,m} = -\sqrt{\frac{(\ell+2)(\ell-1)}{\ell(\ell+1)}}\kappa_{\ell,m} . \quad (3.42)$$

3.6 E - and B -modes

We can define a vector field \mathbf{u} as the gradient of the 'potential' κ , $\mathbf{u} = \nabla\kappa$. Then, the curl of this gradient vanishes since $\nabla \times \nabla\kappa = 0$. This motivates the decomposition of the shear fields into E -mode (gradient-like) and B -mode (curl-like) components:

- **E -mode:** are associated with the convergence field κ and represent the true lensing signal caused by mass fluctuations
- **B -mode:** correspond to a curl component and should vanish for pure gravitational lensing under standard assumptions (no systematic effects)

Therefore, the detection of nonzero b -modes can signal the presence of systematic errors in the data, such as imperfect shape measurements, intrinsic alignments or residual PSF effects. Mathematically, the shear field γ can be expressed in terms of E - and B -modes as:

$$\gamma = \nabla^2\phi_E + i\nabla^2\phi_B , \quad (3.43)$$

where ϕ_E and ϕ_B are scalar potentials for E - and B -modes respectively.

Chapter 4

Scattering transform

In cosmology, understanding the non-Gaussian information embedded in cosmological fields is crucial for probing the underlying physics. However, traditional analytical approaches, like perturbation theory to derive higher order N -point correlation functions, often struggle to capture this information due to their modeling complexity and computational challenges. On the other hand, machine learning techniques like convolutional neural networks (CNNs) have shown promise in extracting non-Gaussian features but require extensive training and large datasets, which can be resource-intensive and less interpretable.

The scattering transform [24] offers a compelling alternative. It provides a framework that bridges the gap between analytical methods and machine learning by extracting hierarchical, interpretable features without the need for training. By leveraging wavelet transforms and non-linear operations, the scattering transform captures non-Gaussian information in a computationally efficient and robust manner. This makes it particularly well-suited for cosmological applications, where both interpretability and precision are essential.

4.1 Wavelet transform

Wavelet transform: Bridging Scales and Features

The wavelet transform is a powerful mathematical tool that bridges the gap between spatial and frequency domains, offering a localized analysis of signals. Unlike the Fourier transform, which provides global frequency information, the wavelet transform captures both spatial and frequency characteristics, making it particularly useful for analyzing non-stationary signals. In the context of cosmology, where structures exist across a wide range of scales, the wavelet transform serves as a foundation for extracting hierarchical features and understanding the multi-scale nature of cosmological fields.

Before delving into the scattering transform, which builds upon the wavelet transform, let us revisit the concepts of the power spectrum to establish a broader context and gain deeper intuition.

4.1.1 Extension of power spectrum

In Eq. (2.32), we defined the power spectrum as $P(k) \equiv \langle \tilde{I}(k) \tilde{I}(-k) \rangle$, where \tilde{I} here could be any cosmological field, such as convergence field or shear field, and the tilde denotes Fourier space. It emphasizes the strength of the clustering at separation $\propto \frac{1}{k}$ in Fourier space. And through the relation between two-point correlation function and power spectrum, we could see that we are using a planar wave-like kernel to do the convolution, where these kernels start from minus infinity to plus infinity, shown in Fig. 4.1. This kind of kernel is a good way to decompose signals into different frequencies and we can get the amplitude of the signal at each frequency.

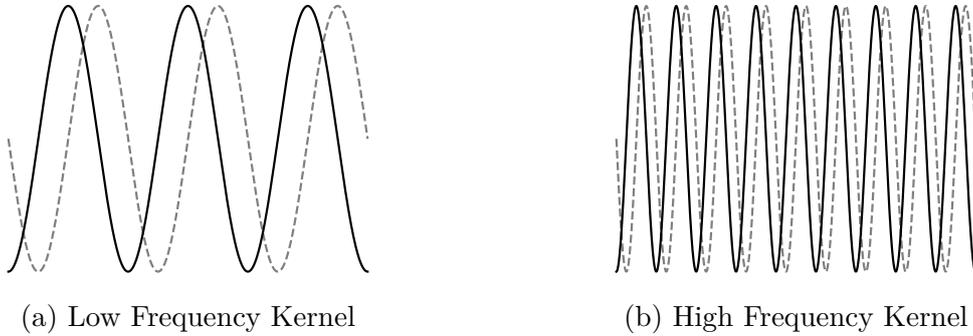


Figure 4.1: Fourier transform kernels with different frequency

In some cases, we cannot always get the amplitude at a single frequency but the average value over a frequency bin. Then, the binned power spectrum could be expressed by utilizing a window function:

$$P_{\text{binned}}(\mathbf{k}) \equiv \int P(\mathbf{k}') w(\mathbf{k}') d\mathbf{k}' \propto \langle |I(\mathbf{r}) \star \psi_{\mathbf{k}}(\mathbf{r})|^2 \rangle, \quad (4.1)$$

where $w(\mathbf{k})$ is the binning window function in Fourier space, and the convolution kernel $\psi_{\mathbf{k}}$ whose Fourier counterpart is defined by $\tilde{\psi}_{\mathbf{k}} = w^{1/2}(\mathbf{k})$, with the tilde denoting Fourier transform. The angular bracket refers to the spatial average of all position \mathbf{r} and “ \star ” represents convolution. We will further discuss this in Section 4.2.5. The comparison between the Fourier kernel and the localized kernel can be seen in Fig. 4.2.

By using localized kernel, we could interpret the operation in the following way: the convolution selects fluctuations and features around a scale, and then by taking modulus, we could convert the selected fluctuations into their local strength, finally, the average measures the overall intensity around this scale.

4.1.2 Formalism

Conceptually, the scattering transform is a sequence of operations that could be taken hierarchically:

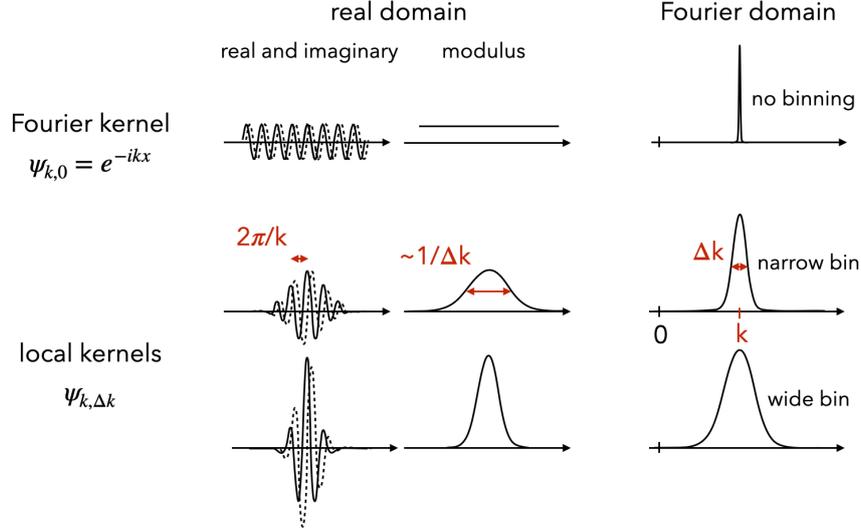


Figure 4.2: Comparison between Fourier kernel and localized kernel. Fourier kernels contain all information about frequency space but no information from real space, while localized kernels contain information in both real and Fourier space. Image from [25].

1. Wavelet Convolution: Use localized oscillating wave as kernels to do convolution on cosmological fields.
2. Modulus: Convert the selected fluctuations (clustering) into their local strength.
3. Average: Calculate the spatial average of the convolved field to get the fluctuation level around that scale.

$$I_{n-1} \rightarrow I_n \equiv |I_{n-1} \star \psi^j| \quad (\text{scattering operation}), \quad (4.2)$$

$$S_n \equiv \langle I_n \rangle, \quad (4.3)$$

where ψ^j stands for a wavelet indexed by j , its logarithmic scale. When the dimension of the input field $I(\mathbf{x})$ is higher than 1, we should add another parameter, l , to describe the orientation of the wavelet kernel.

Fig. 4.3 shows the successive applications of the scattering transform method, which forms a hierarchy structure, with wavelets connecting different layers and convolved fields $I_n(x)$ as nodes. Each I_n represents the intensity map at a specific scale derived from the previous-order field I_{n-1} , resembling another convolution on that field. The averaging operation at each node is similar to the average pooling operation in convolutional neural networks. The 0th-, 1st-, and 2nd-order scattering coefficients can be explicitly expressed as:

$$S_0 \equiv \langle I_0 \rangle, \quad (4.4)$$

$$S_1(j_1, l_1) \equiv \langle |I_1^{j_1, l_1}| \rangle = \langle |I_0 \star \psi^{j_1, l_1}| \rangle, \quad (4.5)$$

$$S_2(j_1, l_1, j_2, l_2) \equiv \langle |I_2^{j_1, l_1, j_2, l_2}| \rangle = \langle | |I_0 \star \psi^{j_1, l_1}| \star \psi^{j_2, l_2} | \rangle, \quad (4.6)$$

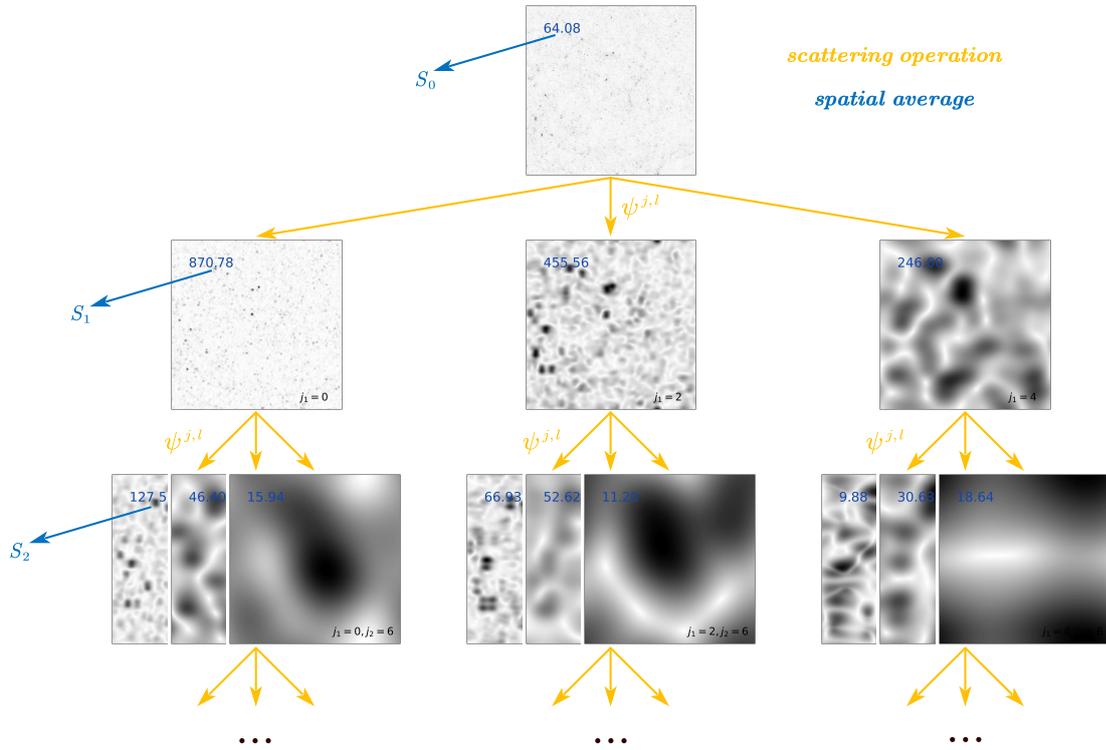


Figure 4.3: Illustration of hierarchy structure of scattering transform. The orange arrow means the scattering operation, which consists of wavelet convolution and pixel-wise modulus, and the blue arrow means the spatial average. Here we only plot two layers of convolution for illustration.

where the indices j and l mean the scale the orientation of the wavelet filter as mentioned before.

Number of scattering coefficients

The number of scattering coefficients depends on the number of wavelet filters used. These filters are defined over a range of logarithmic scales, η^j , where $J_{min} \leq j \leq J_{max}$ and η represents the wavelet dilation parameter. In this project, we adopt dyadic scaling ($\eta = 2$) and set $J_{min} = 1$. For simplicity, J is used to denote J_{max} . The signal length must not exceed the maximum wavelet filter scale, 2^j . With J scale options for each wavelet, the n -th order of convolution generates J^n possible combinations, though only a subset contains meaningful information. Since with higher and higher scale parameter j , the wavelet filter covers more and more areas in real space. This means that if we perform a wavelet convolution with a smaller filter compared to previous one, we are doing convolution inside previous convolved field, and this does not take new area into account, which could get any new information. Therefore, in order to get new information after each order of convolution, we have to choose wavelet kernels that are larger than previous ones. As a result, only combinations with $j_2 > j_1$ are important, which significantly reduced the number of informative coefficients by a factor of 2^{n-1} in n -th order. Thus, the number of useful scattering coefficients at each order is:

$$\begin{aligned} 0\text{th order} &: 1 \text{ coefficient.} \\ 1\text{st order} &: J \text{ coefficients.} \\ 2\text{nd order} &: J(J-1)/2 \text{ coefficients.} \end{aligned}$$

Since J represents the logarithm of the dynamic range of scales in the field, the number of scattering coefficients grows slowly as the field size increases. This results in a relatively compact set of descriptors.

Similar to power spectrum, the scattering transform method could also be applied to higher dimensional data, such as images or 3D physical fields. The only thing that we need to change is the parameter space, where we not only use scale parameter j , but also the orientation parameter l . As a sequence, all scale indices should be replaced by that of oriented scales: $j \rightarrow j, l$. The angular size (J) and angular sampling rate (L) remain a free choice. For example, for the two-dimension case, one could probe L orientations by using angular sampling rate π/L , whose position angles are $\pi l/L$, with $0 \leq l < L$, shown in Fig. 4.4. Compared to 1D case, there are L^n times more coefficients at n -th order. Fortunately, these coefficients could be further reduced.

Reduction of orientations

When dealing with isotropic fields, we could simply average the scattering coefficients over all orientation indices, which reduces the number of scattering coefficients by a factor of

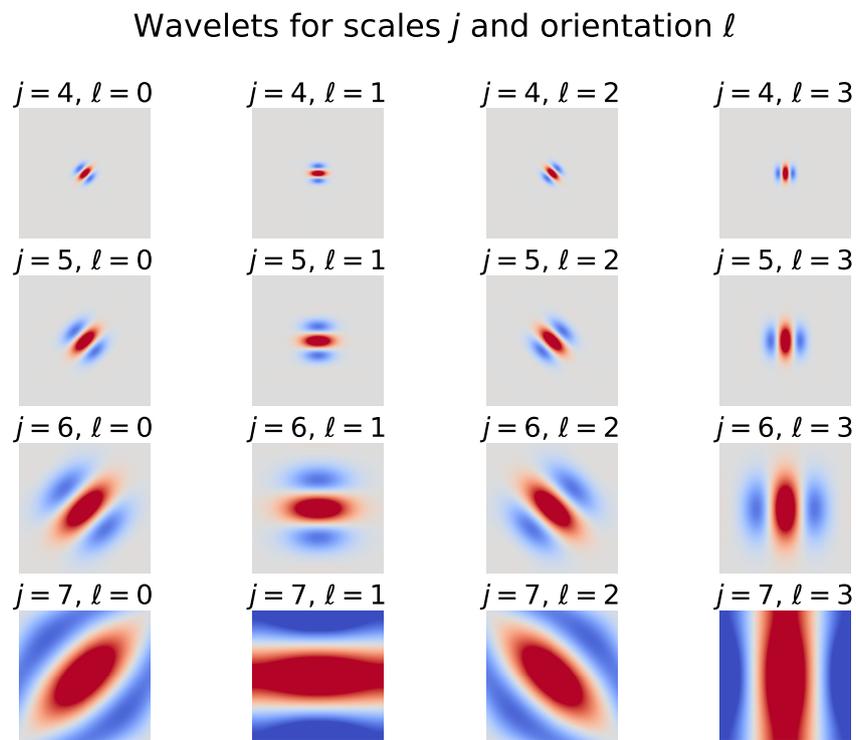


Figure 4.4: Wavelets for each scales j and orientations l used ($J = 8$ and $L = 4$ here). Color saturation and color here respectively denote complex magnitude and complex phase.

L^n and creates a much more compact set of statistical descriptors:

$$s_1(j_1) \equiv \langle S_1(j_1, l_1) \rangle_{l_1} , \quad (4.7)$$

$$s_2(j_1, j_2) \equiv \langle S_2(j_1, l_1, j_2, l_2) \rangle_{l_1, l_2} , \quad (4.8)$$

where $\langle \cdot \rangle_l$ represents the average over orientation indices, lowercase letters denote *reduced scattering coefficients* and uppercase are original scattering coefficients. For example, if we set $J = 8$, then we have $1 + 8 + 28$ reduced scattering coefficients (s_0, s_1, s_2) in total.

Normalization

Since the wavelets filters are dimensionless, the scattering coefficients retain the same units as the input field I_0 across all orders. However, sometimes we need to work with dimensionless statistics. Additionally, since the coefficients S_n are just once more convolution and modulus operation based on field I_{n-1} , they are strongly correlated. To address these issues—making these coefficients unitless and reducing their correlation—one can utilize normalized scattering coefficients, defined as:

$$s_{n,decor} = \frac{s_n}{s_{n-1}} , \quad (4.9)$$

where s_n and s_{n-1} should belong to the same branch of the scattering tree, i.e., having the same $\{j_1, \dots, j_{n-1}\}$, and then $s_{n,decor}$ should have the same number of features as s_n . The necessity of using the 1st-order ratio s_1/s_0 depends on the physical significance of the field and is not always required.

4.2 Understanding the operations

4.2.1 Translation-invariant descriptors

Translation-invariance is a fundamental property of many physical laws. Its features do not depend on absolute position but rather on the statistical structure of the field. However, embedding this invariance into a descriptor is not straightforward. In this section, we take a step back to outline the big picture of translation-invariant descriptor, providing insights into the scattering transform and a variety of other statistical descriptors.

First, due to their simplicity and interpretability, one might prefer to rely solely on linear operation to construct translation-invariant descriptors. However, the only descriptor that satisfies the requirement is the global mean of the field. The reason is as follows. All translation operators share the same eigenvectors: the Fourier modes, $e^{i\mathbf{k}\cdot\mathbf{x}}$, with eigenvalues $e^{-i\mathbf{k}\cdot\mathbf{c}}$. The mathematical representation shows in the following equation:

$$L_{\mathbf{c}} e^{i\mathbf{k}\cdot\mathbf{x}} = e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{c})} = e^{-i\mathbf{k}\cdot\mathbf{c}} e^{i\mathbf{k}\cdot\mathbf{x}} , \quad (4.10)$$

where $L_{\mathbf{c}}$ is the translation operator and \mathbf{c} here is a constant denoting the shift. If we want the translation-invariance property for any displacement \mathbf{c} , then the eigenvalue,

$e^{-i\mathbf{k}\cdot\mathbf{c}}$, should be unity, which means that $\mathbf{k} = \mathbf{0}$, corresponding to the mean value of the field.

So, to extract translation-invariant information beyond the global mean value, the estimator we use must involve *non-linearity* (see Appendix C for mathematical details). A typical way to achieve this is to first non-linear transform the field, and then take the mean of the transformed field:

$$\text{descriptor} \equiv \langle \text{transformed field} \rangle . \quad (4.11)$$

The role of the non-linear operator, such as the modulus, is to convert fluctuations at given frequency ranges into their strengths. Without this step, the mean value vanishes for zero-mean fields, implying that direct averaging does not retain any high-frequency ($\mathbf{k} \neq 0$) information. By taking the modulus, we capture the strength of the fluctuations in those frequency bins, effectively transferring high-frequency information into the $\mathbf{k} = 0$ mode through averaging. Additionally, the non-linear transformation must preserve translation-invariance, meaning it should commute with the translation operator.

4.2.2 Wavelet convolution and modulus

The scattering operation $I \rightarrow |I \star \psi^{j,l}|$ comprises two steps: a convolution by wavelet filter and a pixel-wise modulus. Then let's decompose the two processes and explain them step by step.

Wavelets

Locality: The wavelet transform is a linear transform that combines properties of both the Fourier transform, which contains all the frequency information but no spatial information, and the pixel decomposition, which includes all spatial information without any frequency information. The wavelet transform is localized in both real space and Fourier space. Its reaction to translations smaller than the wavelet scale resembles Fourier coefficients (pure phase shift), whereas for large displacements, it behaves like pixels.

Regularity: Wavelets belonging to the same family—where “family” here refers to a specific wavelet type, such as Morlet wavelet in this project and Mexican hat wavelet [26]—share the same shape but differ in orientations and sizes. They offer a broad, logarithmic tiling in Fourier space, enabling logarithmic sub-sampling of scales. This reduces the number of coefficients that need to be computed. While wavelet convolution is a linear decomposition, its extensive logarithmic tiling of scales underpins the deformation stability of scattering coefficients.

Morlet wavelets

In this project, we use the Morlet wavelet as the convolution kernel, since it provides a good compromise between spatial and frequency localization. A Morlet wavelet con-

sists of a plane wave modulated by a Gaussian envelope, resulting in a function that is simultaneously localized in real and Fourier space. The Gaussian envelope ensures that the wavelet is compact in space, while modulation defines its central frequency in Fourier space. We adopt a dyadic scaling scheme, where the wavelet is sampled at scales increasing by factors of two. In Fourier space, each wavelet is centered around a characteristic frequency, with its bandwidth roughly matched to that frequency, providing a natural multiscale decomposition.

$$G(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma|}} e^{-\mathbf{x}^T \Sigma^{-1} \mathbf{x} / 2} e^{i \mathbf{k}_0 \cdot \mathbf{x}} , \quad (4.12)$$

where Σ is the covariance matrix that defines the Gaussian envelope's size and shape, and \mathbf{k}_0 specifies the frequency and direction of the oscillation. For maximum symmetry, Σ is typically chosen as a diagonal matrix with one eigenvalue distinct from the others, and \mathbf{k}_0 aligns with that eigen-direction. The eigenvalue along \mathbf{k}_0 is set to σ^2 , while the other eigenvalues are set to σ^2/s^2 , where s is the ratio of the wavelet's transverse to radial width in Fourier space. In Fourier space, the Gabor function corresponds to a Gaussian filter centered at \mathbf{k}_0 :

$$\tilde{G}(\mathbf{k}) = \frac{1}{\sqrt{|\Sigma|}} e^{-(\mathbf{k}-\mathbf{k}_0)^T \Sigma (\mathbf{k}-\mathbf{k}_0) / 2} . \quad (4.13)$$

Wider envelopes calculated the weighted average over larger areas in real space, and they have narrower profiles in Fourier space, corresponding to the lower k -mode information. However, there is a problem for Gabor filter: its profile does not go to zero at zero frequency. This contradicts the admissibility of wavelet, which requires wavelets to strictly be band-pass filter rather than low-pass filter. To satisfy the admissibility condition, we subtract a correction term β , leading to the modified Gabor filter [24]. And this is equivalent to subtract another Gaussian function centered at 0 to cancel the 0-frequency contribution. Then the corrected Gabor function becomes:

$$\tilde{\psi}(\mathbf{k}) = \frac{1}{\sqrt{|\Sigma|}} e^{-(\mathbf{k}-\mathbf{k}_0)^T \Sigma (\mathbf{k}-\mathbf{k}_0) / 2} (e^{i \mathbf{k}_0 \cdot \mathbf{x}} - \beta) , \quad (4.14)$$

where $\beta = e^{-\mathbf{k}_0^T \Sigma \mathbf{k}_0 / 2}$, ensuring the wavelet meets the admissibility criterion. The expression in Fourier space is:

$$\tilde{\psi}(\mathbf{k}) = \tilde{G}(\mathbf{k}) - \beta e^{-\mathbf{k}^T \Sigma \mathbf{k} / 2} . \quad (4.15)$$

Here, we follow the convention in kymatio, [27], (<https://www.kymatio.io>),

$$\begin{aligned} \sigma &= 0.8 \times 2^j , \\ k_0 &= \frac{3\pi}{4 \times 2^j} , \\ s &= 4/L , \end{aligned} \quad (4.16)$$

where σ is measured in pixels, j starts from 0, and k_0 ranges from 0 to 2π . This configuration ensures that the Morlet wavelet family effectively spans the entire Fourier space using a dyadic scale sequence (2^j). Mathematically, we could also see that with increasing j , k_0 is becoming smaller and wavelet in position space is becoming larger and larger, just

capturing large scale information. On the contrary, it captures small-scale information. In this project, $J = 8$ and $L = 4$ are used. Examples of Morlet wavelets are illustrated in Fig. 4.5.

Modulus

Wavelet convolution is a linear operation, preserving the first-order properties of the field. However, for a homogeneous field, linear convolution alone cannot extract information beyond the mean value $\langle I \rangle$, as the expectation operator commutes with all linear transformations. To capture information beyond the mean, non-linear operations are required. In the scattering transform, this is achieved using the modulus operation. The pointwise modulus is a non-linear transformation that converts selected fluctuations into their local strengths. Typically, the envelope of these fluctuations has a lower frequency than the original fluctuations, allowing the modulus to scatter high-frequency information to lower frequencies.

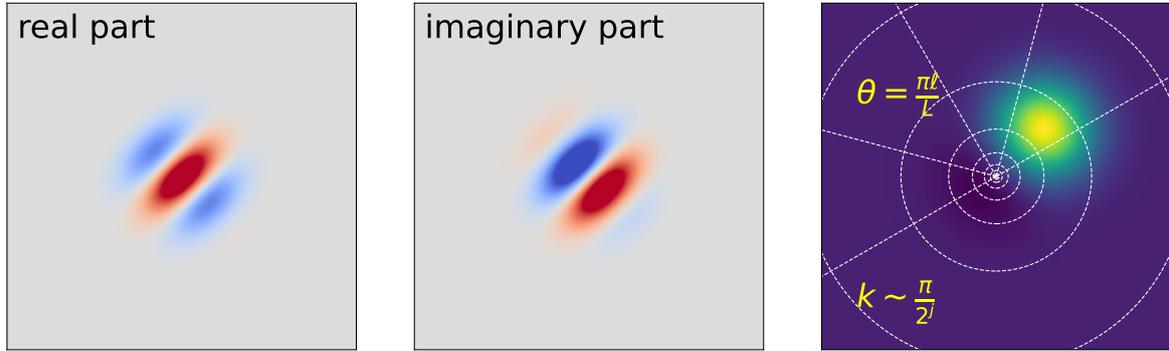
4.2.3 Hierarchy: information extraction beyond the power spectrum

There are some similarities between power spectrum and scattering transform. If we express the mathematical form of the power spectrum in terms of the convolution operation, we have:

$$P(\mathbf{k}) \propto \langle |I \star \psi'|^2 \rangle \quad \text{with } \psi' = e^{-i\mathbf{k}\cdot\mathbf{x}}. \quad (4.17)$$

Both of power spectrum and scattering transform describe the fluctuation strength as a function of scales. The difference between scattering transform and power spectrum lies on the choice of convolution kernels, of which one uses localized wavelet and L_1 norm while another applies global Fourier modes and L_2 norm. Because of the features of the convolution kernels, power spectrum's version of I_1 fields, $|I \star \psi'|^2$, contains all frequency information, like we know the amplitude at each frequency, but it does not include any spatial information. In contrast, the localized wavelet kernel preserves some spatial information, as shown in Fig. 4.6. Analogous to the power spectrum, the mean of I_1 field represents the average amplitude of Fourier modes chosen by wavelets. However, unlike the power spectrum, the spatial distribution of fluctuations in scattering transform also captures the phase interactions between these Fourier modes. This information could be further extracted by doing another scattering transform operation, $I_1 \rightarrow I_2 = |I_1 \star \psi_2| = ||I_0 \star \psi_1| \star \psi_2|$, and then we could simply get the second order scattering coefficients, S_2 , by taking the average of I_2 . This process could be repeated to get higher order scattering coefficients.

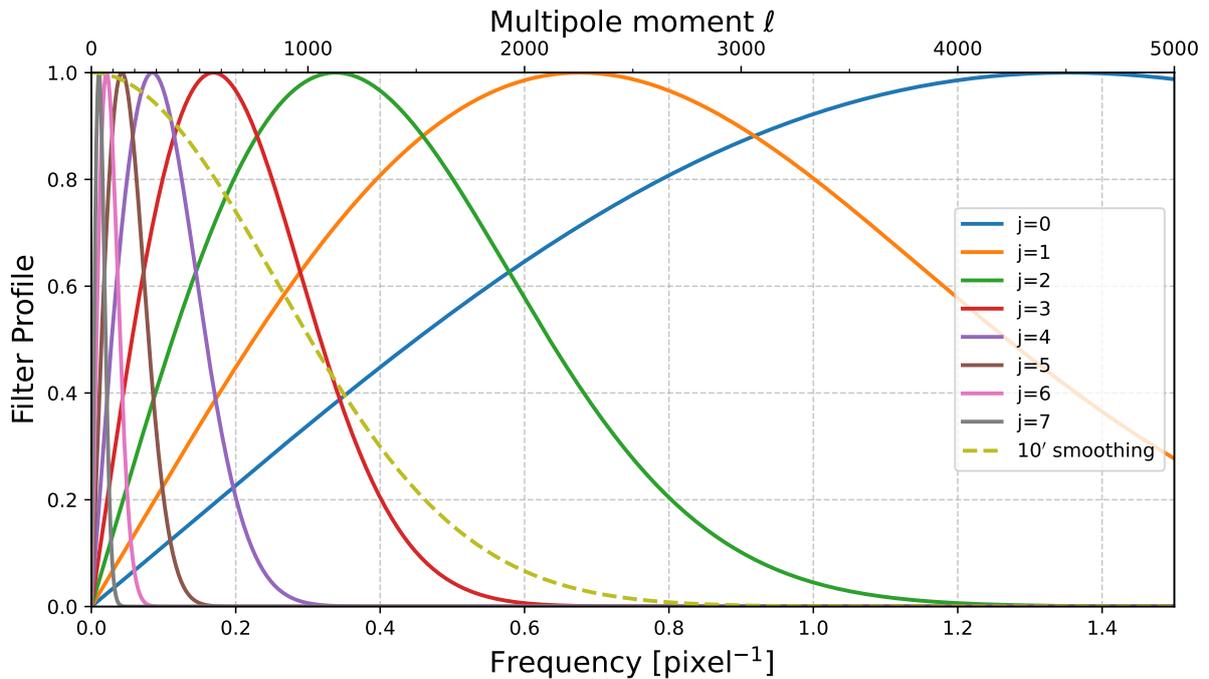
Similarly, the S_2 coefficients behave like the power spectrum of the I_1 fields and characterize how the localized features extracted by I_1 are distributed across space. Since I_1 already captures the scale-dependent variations of the original field, S_2 can be viewed as quantifying the spatial organization or coherence of these variations, thus providing



(a) Real space real part

(b) Real space imag part

(c) Fourier space



(d) Radial frequency profiles

Figure 4.5: *Upper panel:* profile of a Morlet wavelet ($j = 5$, $\ell = 0$, image size 512×512 pixels) in the real space and Fourier space. The central point in Fourier space represent zero frequency. *Lower panel:* Radial frequency profiles of Morlet wavelets with different scales. Dilating or contracting the wavelet by a factor of 2 gives the whole family of wavelets. $J = 8$ and $L = 4$ for these plots.

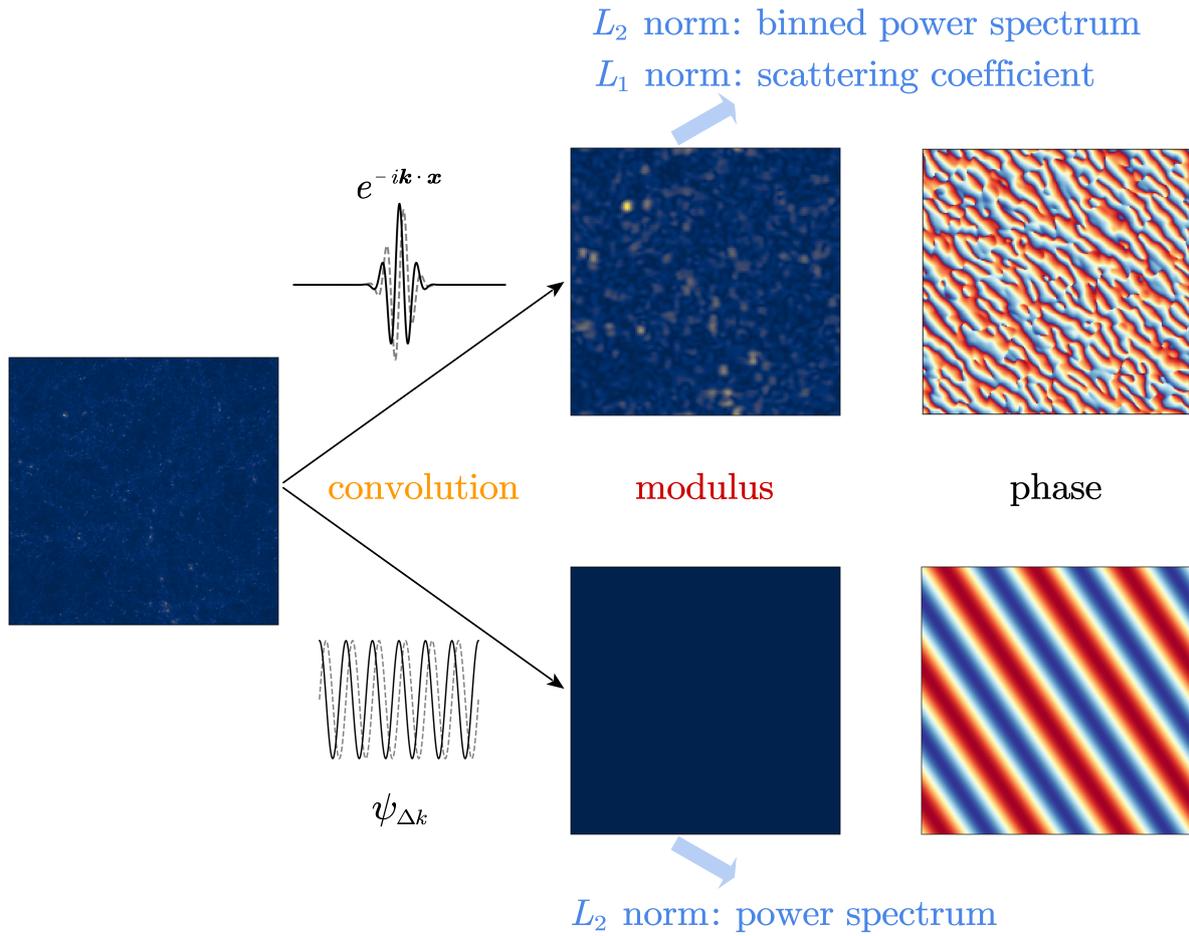


Figure 4.6: Comparison between scattering transform and power spectrum. We have some input cosmological field I and we apply two different convolution kernels on it. The upper two are modulus and phase part of scattering transform while the lower two are power spectrum.

a higher-order statistical description of the input field. This enables S_2 to probe interactions between structures across different spatial scales, beyond what is captured by S_1 alone.

Intuitively, we could think of the scattering transform as a hierarchical method to extract clustering information. Through iteration, we could get n th-order scattering coefficients S_n . By analogy with the N -point correlation function, we could interpret the physical meaning behind this. Let's simply start from the first order. Since we need at least two points to describe the scale of some structures, the first order scattering coefficients S_1 could be understood as two-point correlation function by doing one time of scattering transform operation on the original fields. Therefore, the 2nd-order scattering coefficients S_2 measures another two points based on previous two points, including information up to four points. In general, n th-order scattering coefficients S_n contains information up to 2^n points. Thus, scattering transform quickly extract information from higher-order statistics.

4.2.4 Scattering operation in Fourier space

In general, the modulus operation in real space, being non-linear, will mix Fourier modes and scatter information across different frequencies. By applying the modulus to $I \star \psi$, where ψ has a single peak in Fourier space, we can represent the information of I around the frequency of ψ in terms of lower frequencies. Essentially, the frequency content of $|I \star \psi|$ is typically lower than that of $I \star \psi$.

Intuitively, this is because the modulus converts the oscillation behavior of the convolution into a smooth envelope. Mathematically, we could re-write the modulus as $\sqrt{(I \star \psi)(I \star \psi)^*}$, where $*$ means complex conjugate. By Taylor expanding the square root in terms of $(I \star \psi)(I \star \psi)^* - C$, we get:

$$\sqrt{(I \star \psi)(I \star \psi)^* - C + C} \propto (I \star \psi)(I \star \psi)^* - C, \quad (4.18)$$

where C is the mean of $(I \star \psi)(I \star \psi)^*$ over all pixels. The leading term of Fourier transform corresponds to the auto-correlation of $I \star \psi$ in Fourier space. When the power spectrum of I is a smooth function, the frequency of $I \star \psi$ is similar to ψ . For the Morlet wavelet, the central wavenumber of the wavelet ψ is defined as k_0 , shown in Eq. (4.16). As we can see in Eq. (4.13), the half-width of the envelope in Fourier space is about $1/\sigma$ (we simply approximate the standard deviation of the Gaussian envelope as the half-width of the wavelet here). And the formula of auto-correlation in Fourier space approximates as:

$$\begin{aligned} & \tilde{I}(\mathbf{k}) \cdot \tilde{I}(\mathbf{k}) \cdot \tilde{\psi}(\mathbf{k}) \cdot \tilde{\psi}(\mathbf{k}) \\ & \propto \tilde{I}^2(\mathbf{k}) \cdot \tilde{\psi}^2(\mathbf{k}) \\ & \propto \tilde{I}^2(\mathbf{k}) \cdot e^{-(\mathbf{k}-\mathbf{k}_0)^T \Sigma (\mathbf{k}-\mathbf{k}_0)}. \end{aligned} \quad (4.19)$$

Thus, the half-width of this auto-correlation is $\sqrt{2}/\sigma$. And from $|\tilde{I}(\mathbf{k}) \cdot \tilde{\psi}(\mathbf{k})|^2$, we could see that the auto-correlation have a centroid at 0. As $\sqrt{2}/\sigma < k_0$, this shows that typical

frequency of $|I \star \psi|$ is lower than $I \star \psi$. As a result, the scattering operation $I \rightarrow |I \star \psi|$ re-expresses the high frequency information in terms of lower frequency modes including 0-frequency component. As the 0-frequency component is translation invariant, it can be directly used as a statistical descriptor of the original field.

4.2.5 Wavelet power spectrum

In Eq. (4.2), we introduced the scattering transform by making three adjustments to the standard power spectrum: removing the squared modulus, incorporating localized kernels, and applying repeated transformations through deeper layers. If we insist on using the squared modulus and keep other modifications, we could get exactly the 2^n -point correlation functions. To make it clearer, we could re-write the squared modulus as $|I|^2 \rightarrow I \cdot I^*$, which is just the definition of two-point correlation function. The mathematical details follow. Let's first define the coefficients gained from the squared modulus as *pseudo scattering coefficients* [28]. Then the general formula of pseudo scattering coefficients are:

$$\begin{aligned}
 S_{n+1}^{\text{pseudo}} &\equiv \langle I_{n+1}^{\text{pseudo}} \rangle \\
 &\equiv \int |I_n(\mathbf{x}) \star \psi_{n+1}(\mathbf{x})|^2 d\mathbf{x} \\
 &= \int \left| \tilde{I}_n(\mathbf{k}) \cdot \tilde{\psi}_{n+1}(\mathbf{k}) \right|^2 d\mathbf{k} \\
 &= \int \tilde{I}_n(\mathbf{k}) \tilde{I}_n^*(\mathbf{k}) \cdot \left| \tilde{\psi}_{n+1}(\mathbf{k}) \right|^2 d\mathbf{k} \\
 &= \int \tilde{I}_n(\mathbf{k}) \tilde{I}_n(-\mathbf{k}) \cdot \left| \tilde{\psi}_{n+1}(\mathbf{k}) \right|^2 d\mathbf{k} \\
 &= \int P_n(\mathbf{k}) \cdot \tilde{\psi}_{n+1}^2(\mathbf{k}) d\mathbf{k} ,
 \end{aligned} \tag{4.20}$$

where we used the property that $\tilde{\psi}_n = \tilde{\psi}_n^*$ in Fourier space, and applied *Parseval's theorem* here, $\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |\tilde{f}(k)|^2 dk$ so that the integral of squared modulus in real space equals to the integral in Fourier space. And $P_n(\mathbf{k})$ here represents the power spectrum of I_n . This equation shows that *each pseudo scattering coefficient is a weighted average (binned) of the power spectrum of the previous order of pseudo scattering field.*

By applying this equation to the first order, we obtain that S_1^{pseudo} are averaged power spectrum (2-point correlation function) of the input field, weighted by wavelets:

$$\begin{aligned}
 S_1^{\text{pseudo}} &= \int P_0(\mathbf{k}) \cdot \tilde{\psi}_1(\mathbf{k})^2 d\mathbf{k} \\
 &= \text{averaged (binned) power spectrum of } I_0 .
 \end{aligned} \tag{4.21}$$

Meanwhile, the first order scattering fields are expressed as:

$$\begin{aligned}
 I_1^{\text{pseudo}}(\mathbf{x}) &= |I_0(\mathbf{x}) \star \psi_1(\mathbf{x})|^2 \\
 &= [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})] \cdot [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})]^* .
 \end{aligned} \tag{4.22}$$

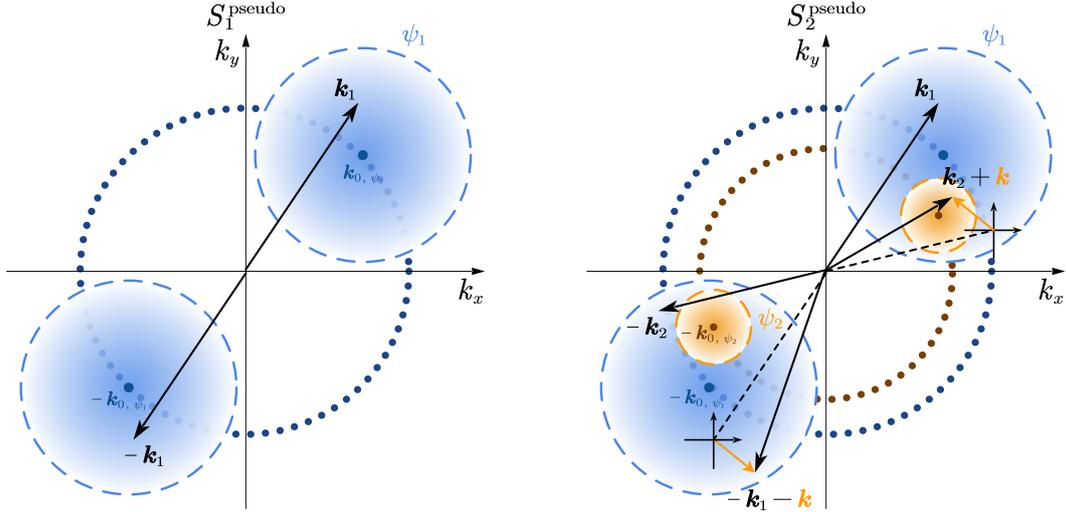
Applying Fourier transform on both sides, we get the Fourier space expression of $I_1^{\text{pseudo}}(\mathbf{k})$:

$$\begin{aligned}
 \widetilde{I_1^{\text{pseudo}}}(\mathbf{k}) &= \mathcal{F} \{ [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})] \cdot [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})]^* \} \\
 &= \mathcal{F} [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})] \star \mathcal{F} [I_0(\mathbf{x}) \star \psi_1(\mathbf{x})]^* \\
 &= [\tilde{I}_0(\mathbf{k}) \cdot \tilde{\psi}_1(\mathbf{k})] \star [\tilde{I}_0(\mathbf{k}) \cdot \tilde{\psi}_1(\mathbf{k})]^* \\
 &= \int [\tilde{I}_0(\mathbf{k}') \cdot \tilde{\psi}_1(\mathbf{k}')] \cdot [\tilde{I}_0(\mathbf{k} + \mathbf{k}') \cdot \tilde{\psi}_1(\mathbf{k} + \mathbf{k}')]^* d\mathbf{k}' \\
 &= \int \tilde{I}_0(\mathbf{k}') \tilde{I}_0^*(\mathbf{k} + \mathbf{k}') \cdot [\tilde{\psi}_1(\mathbf{k}') \tilde{\psi}_1(\mathbf{k} + \mathbf{k}')] d\mathbf{k}' ,
 \end{aligned} \tag{4.23}$$

where we used convolution theorem. This is exactly the same as the two-point correlation function weighted by some window function. Therefore, the first order scattering fields are just the two-point correlation function of the input field, weighted by wavelets. Again, we have second order pseudo scattering coefficients expressed as:

$$\begin{aligned}
 S_2^{\text{pseudo}} &= \int P_1(\mathbf{k}) \cdot \tilde{\psi}_2(\mathbf{k}) d\mathbf{k} \\
 &= \int \widetilde{I_1^{\text{pseudo}}}(\mathbf{k}) \cdot \widetilde{I_1^{\text{pseudo}}}^*(\mathbf{k}) \cdot \tilde{\psi}_2^2 d\mathbf{k} \\
 &= \iiint \tilde{I}_0(\mathbf{k}_1) \tilde{I}_0(\mathbf{k} + \mathbf{k}_1)^* \tilde{I}_0(\mathbf{k}_2)^* \tilde{I}_0(\mathbf{k} + \mathbf{k}_2) \\
 &\quad \cdot [\tilde{\psi}(\mathbf{k}_1) \tilde{\psi}(\mathbf{k} + \mathbf{k}_1) \tilde{\psi}(\mathbf{k}_2) \tilde{\psi}(\mathbf{k} + \mathbf{k}_2) \tilde{\psi}_2^2(\mathbf{k})] \\
 &\quad \cdot d\mathbf{k}_1 d\mathbf{k}_2 d\mathbf{k} \\
 &= \text{weighted averaged trispectrum of } I_0 .
 \end{aligned} \tag{4.24}$$

The integrand consists of four Fourier coefficients of the input field, each modulated by the first filter $\tilde{\psi}_1$, while their difference \mathbf{k} is modulated by the second filter $\tilde{\psi}_2$. From Fig. 4.7, we could see the illustration of first- and second-order pseudo scattering coefficients and the corresponding two- and four-point correlation functions. Higher order cases could be derived similarly. showing that S_n^{pseudo} behaves exactly the same way as the 2^n -point correlation function, but weighted by $\psi_1, \psi_2, \dots, \psi_n$.



(a) 1st order pseudo scattering coefficients (b) 2nd order pseudo scattering coefficients

Figure 4.7: Illustration of the pseudo scattering coefficients and their connection to the N -point correlation function in Fourier space. Since we set parameters such that Morlet wavelets have isotropic Gaussian profiles in Fourier space, we approximate their regions of influence using circles. The dashed circles, filled with a gradient color, represent Morlet wavelet filters at different scales, with darker regions indicating higher filter weights. The dots at the center of these circles correspond to the peak locations of the wavelet filters, given by $k_0 = 0.75\pi \times 2^{-j}$. Because Morlet wavelets depend only on the frequency magnitude and not its direction, their center points can lie anywhere along the dotted circles centered at the origin. For clarity, we illustrate only one such choice of center in this figure. *Left panel:* First-order pseudo scattering coefficients, which depend on the two-point correlation function at \mathbf{k} and $-\mathbf{k}$. *Right panel:* Second-order pseudo scattering coefficients, which depend on the four-point correlation function at \mathbf{k}_1 , $-\mathbf{k}_1$, $-\mathbf{k}_2 - \mathbf{k}$, and $\mathbf{k}_2 + \mathbf{k}$. Since the second wavelet filter, ψ_2 , captures larger-scale features than the first wavelet filter, ψ_1 , the second-order pseudo scattering coefficients provide information at lower frequencies.

Chapter 5

Neural Network-based emulator

As machine learning (ML) reshapes technological landscapes across various sectors, its influence is profoundly felt in the field of cosmology. This branch of astrophysics, which deals with the origins and evolution of the universe, benefits greatly from the data-driven insights provided by ML. Given the vast amount of data generated by astronomical observations and simulations, traditional analysis methods are often insufficient. ML with its ability to process and interpret large datasets efficiently, is ideally suited to help cosmologists tackle these challenges.

In cosmology, ML is primarily used to analyze large-scale cosmic structures and phenomena that are otherwise too mathematically complicated or computationally expensive for traditional analytical methods. With the advent of extensive sky surveys and simulations generating petabytes of data, conventional data analysis techniques have become inadequate. ML models, especially those involving neural networks, are adopted at handling such voluminous and complex datasets, providing insights that are both deeper and computationally feasible. Further more, ML algorithms have been pivotal in refining methods for dark matter and dark energy research, two of the most exclusive aspects of modern cosmology. By applying these algorithms to gravitational lensing data, researchers can map dark matter distribution more accurately and infer the properties of dark energy.

The integration of ML into cosmology represents a paradigm shift in how data from the universe is interpreted. It not only enhances the precision of our existing knowledge but also paves the way for novel discoveries that push the boundaries of our understanding of the universe.

In Section 5.1 and 5.2, some ML related terminologies will be introduced. Then in Section 5.3, we will discuss the bias-variance tradeoff, a fundamental concept in ML that highlights the balance between model complexity and generalization. Following that, Section 5.4 will delve into the structure of deep neural networks, exploring their architecture and components. Finally, Section 5.5 will cover optimization techniques, including gradient descent and regularization methods, which are essential for training ML models effectively.

5.1 Terminology of ML

First, let's clarify some terminologies that are frequently used in ML and artificial intelligence field. Artificial intelligence, ML and deep learning are usually used interchangeably, but there exist some distinctions between these three terminologies. Fig. 5.1 shows the hierarchical relationship between AI, ML and deep learning. And the following gives both the concept and distinctions of these terminologies:

- **Artificial Intelligence (AI):** A broad field of computer science dedicated to creating systems capable of performing tasks that typically require human intelligence. The primary distinction of AI is its ability to mimic human decision-making processes and undertake tasks like reasoning, visual perception, and language understanding. AI systems are not necessarily designed to learn from their interactions; they can simply execute predefined tasks based on programmed rules and logic.
- **ML:** ML is a subset of AI that specially focuses on algorithms that enable machines to improve at tasks over time without being explicitly programmed for each task. The key differentiator for ML is its reliance on data and statistical methods to “learn” information directly from data without relying on a predetermined equation as a model. ML systems adapt and improve based on experience, primarily through exposure to more data.
- **Deep Learning:** Deep learning is a further specialization within ML, distinguished by its use of deep neural networks—networks with multiple layers of processing units, allowing them to learn complex patterns in large amount of data. Deep learning differentiates itself by the depth of its neural networks, which are capable of learning from a vast array of features and variables in massive amounts of data, often outperforming other ML techniques on tasks that involve such complexities.

5.2 Deep learning

Deep learning architectures are based on deep neural networks, which consist of multiple layers of nodes, also known as neurons. These networks mimic the layered structures of the human brain and are designed to progressively extract increasingly abstract and informative representations from the raw input.

- **Layers and Nodes:** The input layers receives the raw data, which is then processed through one or more hidden layers, as shown in Fig. 5.2. Each layer consists of nodes that perform various computations through a set of learned weights and biases. The output of each node is transformed by a nonlinear activation function, which helps the networks learn complex patterns.
- **Learning Process:** In a deep learning neural networks, each subsequent layer uses the output from the previous layer as its input, refining the abstraction and complexity of the information. The final output layer produces the predictions or classifications based on the learned features.

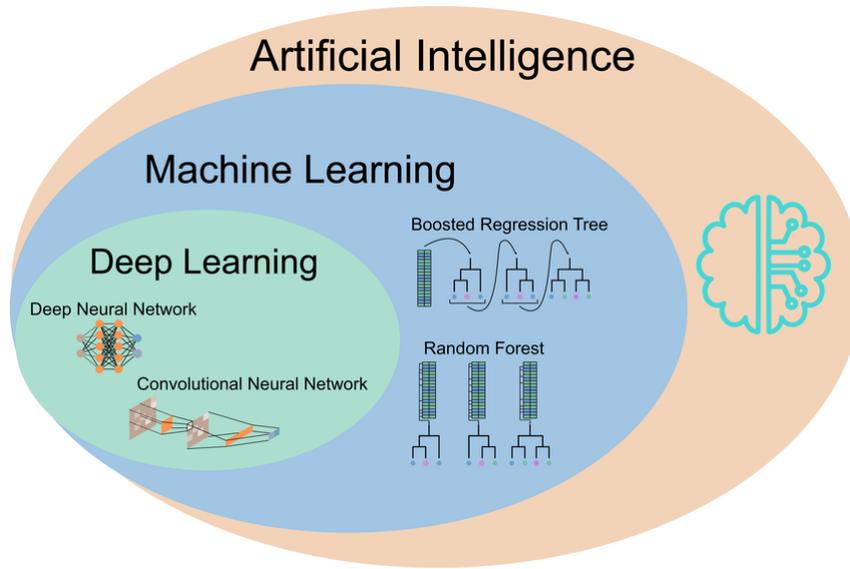


Figure 5.1: Relationship between AI, ML and deep learning. This diagram illustrates the nested structures of AI, ML and deep learning. AI is the overarching field that employs algorithms to perform tasks requiring human-like intelligence. ML is a subset of AI where algorithms learn from data to make decisions or do regression. Deep learning, nested within ML, utilizes complex neural networks (such as convolutional and deep neural networks) to process layered computations and patterns in data, advancing capabilities beyond traditional ML methods like boosted regression trees and random forests. Image from [29].

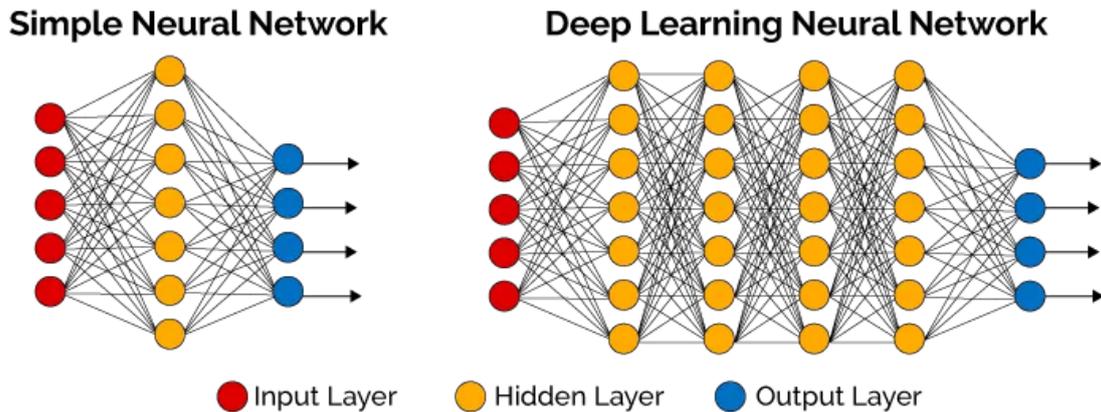


Figure 5.2: Illustration of deep learning neural networks. The left diagram represents a simple neural network that consists of only one hidden layer. This hidden layer receives raw data from the input layer, processes it by multiplying with a weight matrix and adding biases, and then applies an activation function. The processed data is then passed to the output layers. While the right one shows the multi-layer neural networks. Each layer might contain different number of nodes, and then each layer receives data from previous layer, do the matrix multiplication, add bias and then take the nonlinear operation. And finally pass the data to the output layer. Image from [30].

Having outlined the fundamental concepts of layers, nodes, and the learning process in deep learning, it's natural to delve deeper into how these models are actually trained to make predictions and decisions. This training can broadly be categorized into two main types: supervised and unsupervised learning. Each method uses different approaches to teach machines from data, shaping how they predict and analyze information. In the following, we will introduce the concept of these two type of learning:

- **Supervised Learning:** In supervised learning, the model is trained using a labeled dataset, which provides both the input features and the corresponding target outputs. The goal is to teach the model to accurately predict the target output from the input data. This method is commonly used for classification tasks, where the output is a category, and regression tasks, where the output is a continuous value. Fig. 5.3 shows the example of supervised learning.
- **Unsupervised Learning:** Unsupervised learning involves training a model on data without any labels, allowing the model to discover the inherent structures and patterns within the dataset on its own. The most common technique is clustering, where the model groups similar data points together. Fig. 5.4 gives unsupervised learning.

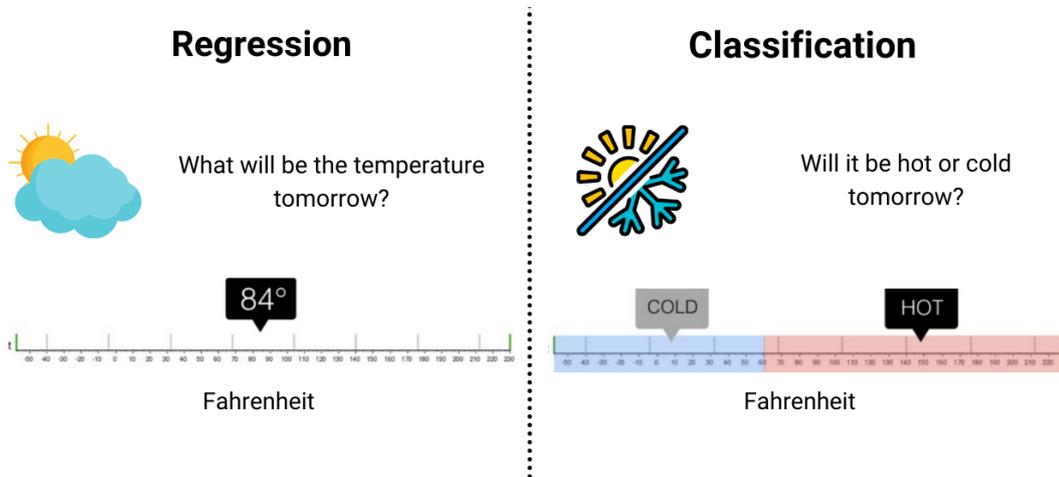


Figure 5.3: Supervised learning. The left one is regression task where neural network need to predict the value of temperature. While the right diagram represents classification, where a temperature value is also provided. Additionally, a threshold is set such that temperatures below this value are classified as “cold”, and those above it are considered “hot”. Image from [31].

After exploring the concepts of supervised learning and unsupervised learning, it's important to address common challenges such as overfitting and underfitting, which can significantly impact the performance of ML models. Overfitting occurs when a model learns the training too well, including the noise and outliers, making it perform poorly on new data. It is more like memorizing the behavior of the training data, rather than learning from the data. Underfitting, on the other hand, happens when a model is too simple to capture the underlying pattern of the data, leading to inadequate performance both



Figure 5.4: Unsupervised learning. This diagram demonstrates the K-means clustering algorithm [32], a method of unsupervised learning. Before clustering, the data points are unlabeled and dispersed. After applying K-means, the data points are grouped into clusters (shown in different colors), each representing a collection of data points with similar characteristics. Image from [33].

on the training data and new data. Fig. 5.5 shows the result of overfitting and underfitting.

5.3 Bias-Variance tradeoff

Usually, the complexity of models directly affects the fitting effect. When the model is too simple, meaning the low complexity, it's very difficult to capture the complicated characteristics of given data. Just like the underfitting in Fig. 5.5, it's impossible to fit the data with only a linear function. While very complex models often correspond to overfitting. In this case, the model seems to be specially customized for this set of data and cannot be used for other data at all, like the overfitting part in Fig. 5.5. Therefore, we need to adjust the complexity of our models so that they are just right for the data we are dealing with. And there is a terminology for this problem, *Bias-Variance Tradeoff*. The concepts of bias and variance in this context follows:

- **Bias:** Bias refers to the errors introduced when a complex real-world problem is approximated using a model that is too simplistic. In ML, high bias occurs when the model fails to capture the underlying patterns of the data, often resulting in underfitting. This is evident even if we train our models on an extremely large dataset—potentially infinite in size—the error remains significant. For instance, employing a linear model to fit data derived from a quadratic function will lead to substantial bias. The linear model, due to its simplicity, underfits the data regardless of the absence of noise.
- **Variance:** Variance refers to an algorithm's sensitivity to minor fluctuations in the

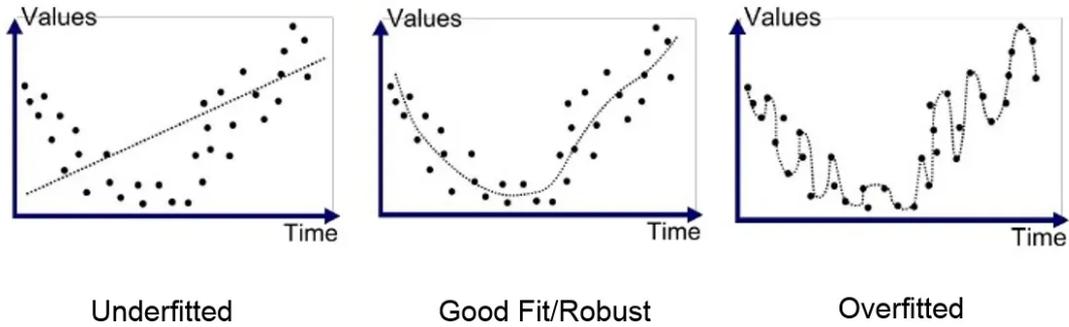


Figure 5.5: Illustration of underfitting, good fit and overfitting. For underfitting, the model is too simplistic to effectively represent the data. In cases of a good fit, the model is appropriately balanced—not too simple or complex—and can generalize well to represent the typical case. For overfitting, the model is excessively complex, fitting the training data points perfectly but performing poorly on new datasets. Image from [34].

training set. High variance can lead an algorithm to model the random noise within the training data rather than the intended outputs, resulting in overfitting. This issue is commonly seen in overly complex models that have an excessive number of parameters relative to the number of observations. In simpler terms, the model learned from the training set does not *generalize* well to new, unseen data—the test error is high. There is a large risk that we’re fitting patterns in the data that happened to be present in our *small, finite* training set, but that do not reflect the wider pattern of the relationship between input and output. For example, fitting a training dataset generated from a quadratic function with an n^{th} order polynomial (where $n + 1$ is equal to the number of data points in the dataset) might result in a model that perfectly passes through every data point, achieving 0 error on this training set. However, such a model is unlikely to perform well on any other dataset. The variance can be intuitively characterized by the amount of variations across models learned on multiple different training datasets.

Fig. 5.7 graphically demonstrates the bias and variance. Briefly speaking, bias measures how close the model approximates the “true” function and variance indicates how stable the model’s stability in response to the noise in the data. To mathematically state the bias-variance tradeoff for regression problems, we consider the following setup [37]:

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^n$ such that $y^{(i)} = h^*(x^{(i)}) + \xi$, where $h^*(\cdot)$ is the ground truth function and ξ is the error with the Gaussian distribution of zero mean and σ^2 variance: $\xi^{(i)} \sim \mathcal{N}(0, \sigma^2)$. And the distribution of this error is independent of the model (say, $\mathbb{E}[\xi h^*] = \mathbb{E}[\xi] \mathbb{E}[h^*]$).
- Train a model on the dataset S , denoted by $\hat{h}_S(\cdot)$.
- Consider a test example point (x, y) such that $y = h^*(x) + \xi$, and measure the expected test mean square error (MSE, this will be introduced in Section 5.4) (average

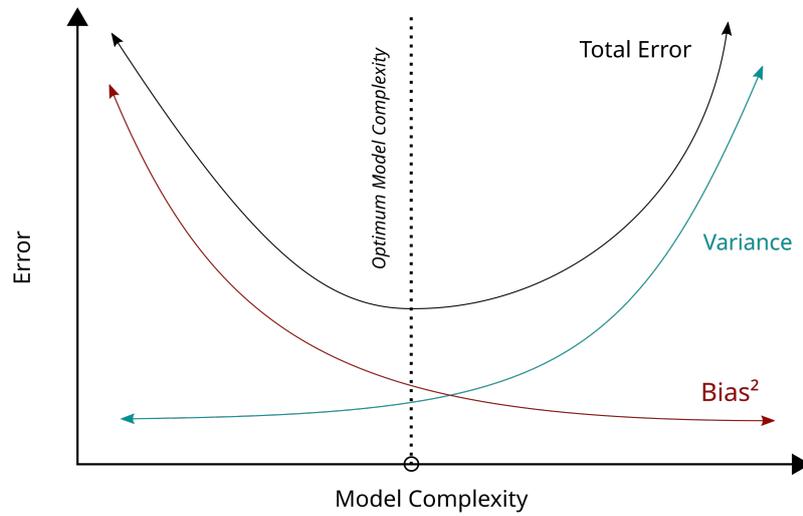


Figure 5.6: Bias-Variance tradeoff. With increasing model complexity, the bias is always decreasing while the variance keeps increasing. To find the point where the total error is minimum, we have to balance the variance and bias. Image from [35].

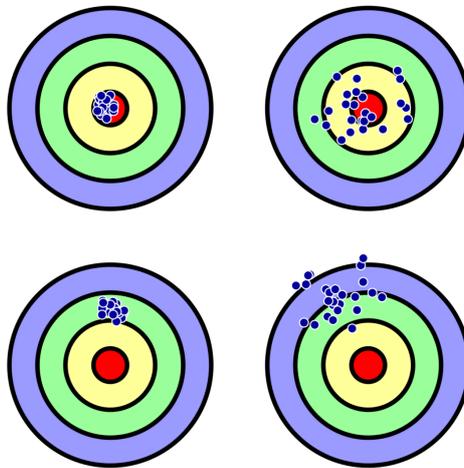


Figure 5.7: Bias and Variance. This diagram depicts the predictions of a model, where each dot represents a single prediction and the center of all circles represents the ground truth value. The closer the dots are to the center, the more accurate the model. The top left one is the model with low bias and low variance, which is the model we like. The top right one represents low bias and high variance model, since the dots are all near the center but they are all scattered. The bottom left one means high bias and low variance, because the predictions are far away from the ground truth and are very concentrated. The bottom right one illustrates high bias and high variance, given that the dots are all far from the center and also scattered. Image from [36].

over the random draw of the training set S and the randomness of ξ)

$$\text{MSE}(x) = \mathbb{E}_{S,\xi} [(y - h_S(x))^2] . \quad (5.1)$$

We now decompose the MSE into a bias and variance term.

$$\begin{aligned} \text{MSE}(x) &= \mathbb{E} [(y - h_S(x))^2] \\ &= \mathbb{E} [(\xi + (h^*(x) - h_S(x)))^2] \\ &= \mathbb{E} [\xi^2] + 2\mathbb{E} [\xi (h^*(x) - h_S(x))] + \mathbb{E} [(h^*(x) - h_S(x))^2] \\ &= \mathbb{E} [\xi^2] + 2\mathbb{E} [\xi] \mathbb{E} [(h^*(x) - h_S(x))] + \mathbb{E} [(h^*(x) - h_S(x))^2] \\ &= \sigma^2 + \mathbb{E} [(h^*(x) - h_S(x))^2] . \end{aligned} \quad (5.2)$$

We define $h_{\text{avg}}(x) = \mathbb{E}_S[h_S(x)]$ as the ‘‘average model’’—the theoretical model created by generating an infinite number of datasets, training models on each, and then averaging their predictions for x . Note that h_{avg} is a hypothetical model used for analytical purposes and is not attainable in reality, as acquiring an infinite number of datasets is practically impossible. However, in many instances, h_{avg} is found to be approximately equivalent to the model developed by training on a *single* dataset that contains an infinite number of samples. We then can decompose $\text{MSE}(x)$ by letting $c = h^*(x) - h_{\text{avg}}(x)$ (which is a constant that does not depend on the choice of S , since h^* is a deterministic function and h_{avg} is defined as the average over infinite of training sets).

$$\begin{aligned} \text{MSE}(x) &= \sigma^2 + \mathbb{E} [(h^*(x) - h_S(x))^2] \\ &= \sigma^2 + \mathbb{E} [(h^*(x) - h_{\text{avg}}(x) + h_{\text{avg}}(x) - h_S(x))^2] \\ &= \sigma^2 + \mathbb{E} [(h^*(x) - h_{\text{avg}}(x))^2] + \mathbb{E} [(h_{\text{avg}}(x) - h_S(x))^2] \\ &\quad + 2\mathbb{E} [(h^*(x) - h_{\text{avg}}(x)) (h_{\text{avg}}(x) - h_S(x))] \\ &= \sigma^2 + \mathbb{E} [(h^*(x) - h_{\text{avg}}(x))^2] + \mathbb{E} [(h_{\text{avg}}(x) - h_S(x))^2] \\ &\quad + 2c\mathbb{E} [(h_{\text{avg}}(x) - h_S(x))] . \end{aligned} \quad (5.3)$$

Based on the definition of h_{avg} , we have

$$\mathbb{E} [(h_{\text{avg}}(x) - h_S(x))] = \mathbb{E} [h_{\text{avg}}(x)] - \mathbb{E} [h_S(x)] = 0 . \quad (5.4)$$

Thus, the final expression of $\text{MSE}(x)$ becomes:

$$\begin{aligned} \text{MSE}(x) &= \sigma^2 + \mathbb{E} [(h^*(x) - h_{\text{avg}}(x))^2] + \mathbb{E} [(h_{\text{avg}}(x) - h_S(x))^2] \\ &= \underbrace{\sigma^2}_{\text{unavoidable}} + \underbrace{(h^*(x) - h_{\text{avg}}(x))^2}_{\triangleq \text{bias}^2} + \underbrace{\text{Var}(h_S(x))}_{\triangleq \text{variance}} . \end{aligned} \quad (5.5)$$

We call the second term the bias (square) and the third term variance. As previously mentioned, the bias reflects the portion of error that arises from the limited expressivity of the model. Recall that h_{avg} represents the optimal model that could be learned, even with infinite amount of training datasets. Therefore, the bias is not a result of insufficient

data; instead, it stems from the inherent inability of the model family to adequately approximate h^* .

The variance component accounts for the errors introduced by the random characteristics of a finite dataset. It quantifies the learned model's sensitivity to the randomness of dataset. Typically, variance decreases with increasing data size.

5.4 Structure of deep neural networks

Having discussed the fundamental concepts of bias and variance as depicted in above sections, it's essential to explore how these factors are influenced by the structure of deep learning neural network. The architecture of a neural network-ranging from the number of layers to the type of layers used-plays pivotal role in determining the balance between bias and variance. A well-designed network structure can effectively minimize bias by capturing complex patterns in the data, while also managing variance to ensure robust performance on new, unseen data. The following section will delve into the architecture of execution of deep learning networks and its critical role in optimizing this balance to enhance model performance and generalization, as shown in Fig. 5.8.

The training process encompasses a series of crucial steps, each aiding the network in its capacity to discern intricate relationships and patterns:

- **Weight Matrix and Bias:** Weights are numerical values associated with neurons in two adjacent layers. They determine the strength of these connections and, in turn, the influence that one neuron's output has on another neuron's input. They can increase or decrease the importance of specific information. While biases are essentially constants associated with each neuron and added to the neuron's output. Biases serve as a form of offset or threshold, allowing neurons to activate even when the weighted sum of their inputs is not sufficient on its own. They introduce a level of adaptability that ensures the network can learn and make predictions effectively. The mathematical expression could be written as follows, with w being the weight matrix, b the bias, X the input, Y the output:

$$Y = w \cdot X + b . \quad (5.6)$$

- **Activation Function:** Activation functions introduce non-linearity to the model, making it learn the complex decisions and predictions. Without this non-linearity feature, a neural network would behave like a linear regression model, no matter how many layers it has. The activation function decides whether a neuron should be activated by calculating the weighted sum of inputs and adding a bias term. Activation functions also enable the backpropagation by providing gradients that are essential for updating the weights and bias. The types of activation function could be seen in Section 5.4. The expression for the activation function, σ , is given by:

$$Z = \sigma (w \cdot X + b) . \quad (5.7)$$

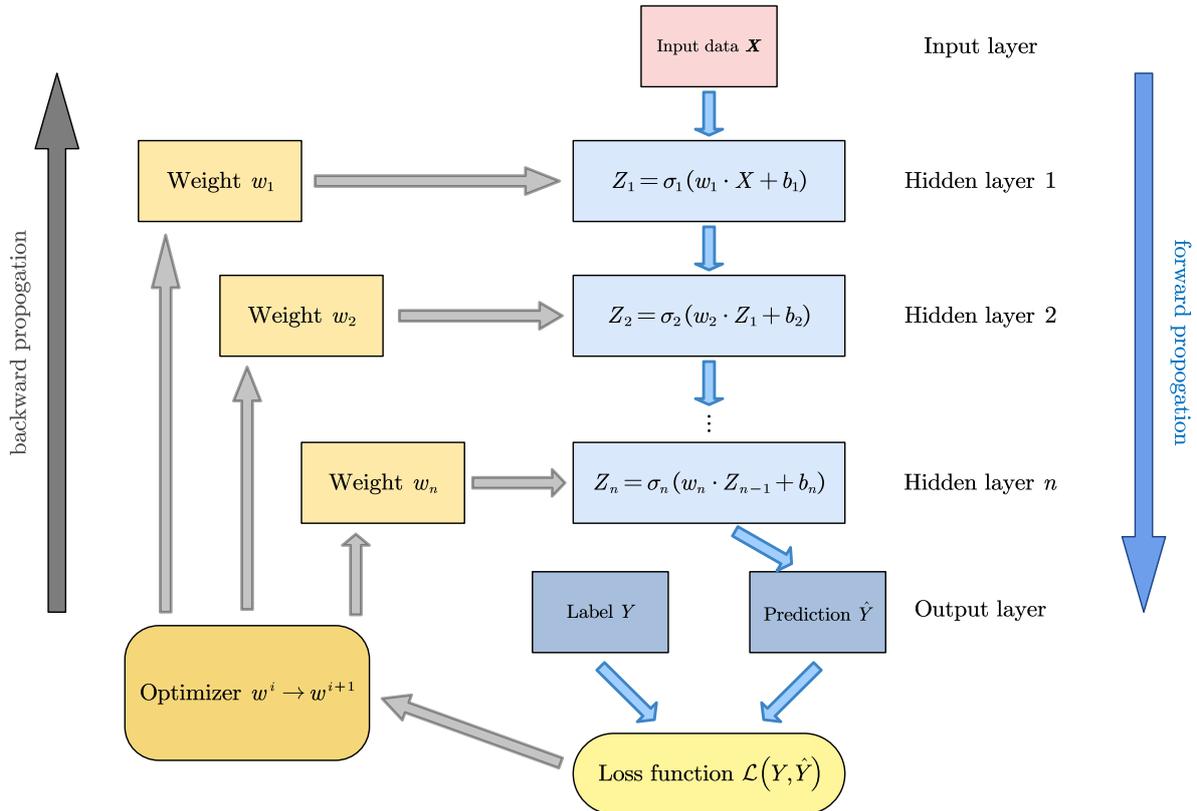


Figure 5.8: The architecture of deep learning neural networks. The raw data is forwarded to the hidden layer, where it is multiplied by a weight matrix and then added to a bias. This is followed by the application of a non-linear activation function, after which the processed data is sent to the next layer. After passing through n hidden layers, the architecture produces final predictions, which are compared to the actual label values. The loss function, which varies depending on the specific problem being addressed, is calculated. We then use the *gradient descent* method to update the weight matrix and proceed with another training iteration.

- **Loss Function:** The loss function, also referred to as the error function, quantifies the discrepancy between the network's prediction \hat{Y} and actual target values Y . The resulting value, the loss, reflects the accuracy of the model's predictions. Usually we want to find the global minimum of the loss function, corresponding to the most accuracy model during the process of training.
- **Backpropagation:** Backpropagation is an algorithm used to train neural networks. Usually, it typically employs the gradient descent method, detailed in Section 5.5, to minimize the loss function and update weights and biases. The update equation could be written as:

$$w_i = w_i - \eta \cdot \frac{d\mathcal{L}}{dw_i} , \quad (5.8)$$

where \mathcal{L} represents the value of loss function, w_i denotes an element in the weight matrix, and η is the learning rate.

Activation Function

Having introduced the structures of deep learning neural networks, here we are going to discuss one of the most crucial elements of deep neural networks: non-linear activation functions. These functions significantly enhance the model's expressiveness according to the universal approximation theorem [38]. Each node in a neural layer performs merely affine so this limitation confines the space to linear relations. On the other hand, non-linear activation functions leverage the capacity of multiple layers, thus broadening the potential range. Typically, these functions are designed to be simple to address computational limitations. Here we will introduce some non-linear activation functions that are frequently used in deep learning, also seen in Fig. 5.9:

- **Sigmoid:** The sigmoid function is one of the most frequently used activation functions in the early days of deep learning. It is a smooth function that is easy to take derivatives. The mathematical equation follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} . \quad (5.9)$$

However, the sigmoid function has some drawbacks. First, the gradient vanishes when x is very large or very small, resulting in long steps to converge, shown in Fig. 5.9. Also, the output is not zero-centered and always greater than zero, leading to zig-zagging dynamics in the gradient updates for the weight.

- **ReLU (Rectified Linear unit):** ReLu is currently one of the most popular activation functions, and basically it's just a function that takes the maximum.

$$\text{ReLU} = \max(0, x) . \quad (5.10)$$

ReLU solves the gradient vanishing problem (in the positive regime). The computation is very vest, requiring only a check if the input is greater than zero. And it converges much faster than sigmoid and tanh function.

- **Tanh**: The Tanh function is also a very popular non-linear activation function. It solves the zero-centered problem, but the gradient vanishing problem still exists, described in Fig. 5.9. It is defined as:

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} . \quad (5.11)$$

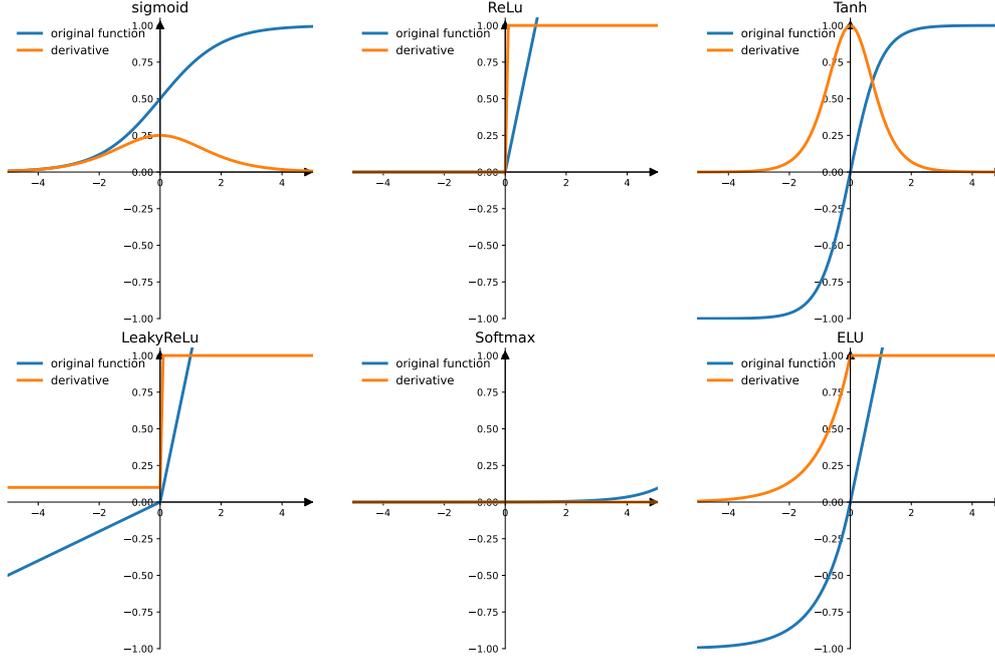


Figure 5.9: Non-linear activation functions. The activation functions and their derivatives comprising Sigmoid, ReLu, Tanh, LeakyReLu, Softmax and Exponential Linear Unit (ELU) were generated using Torch, where the slope of negative regime for LeakyReLu was set to 0.1.

Loss Function

In supervised learning, the loss function plays an important role in the training process. This function serves as a measure to quantify the difference between the predictions made by the model and the actual data. By doing so, it allows the model to incrementally improve its performance through the optimization process. Common loss functions include MSE, cross-entropy loss, Kullback-Leibler Divergence (KL Divergence), negative loglikelihood (NLL) and so on. In this project, we use MSE to quantify the error. So we will introduce MSE in the following.

MSE is a widely used loss function in regression problem tasks. The MSE quantifies the average squared deviation between the model's prediction, denoted as \hat{y} , and the observed values, written as y , within the dataset. In this case of a dataset containing n data points,

the MSE is calculated as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 , \quad (5.12)$$

where \hat{y}_i represents the predicted value for the i th data point, while y_i corresponds to the i th observed data in this dataset. It's crucial to understand that employing the MSE loss function does not ensure that the trained networks will provide an unbiased estimation of the model parameters. As depicted in Eq. (5.5), the MSE comprises both a bias term and a variance component.

5.5 Optimization

In this section, we will introduce the optimization, a pivotal aspect of ML that ensures the effective training of models. This section delves into gradient descent [39], a fundamental algorithm for minimizing the loss function, highlighting its role in driving the convergence of model parameters towards optimal values. Additionally, regularization [37] techniques will be discussed, which are crucial for preventing overfitting and enhancing the model's generalization capabilities. Lastly, the section covers hyperparameters, which, unlike model parameters that are modified during training process, are set prior to the training process and significantly influence the learning trajectory and performance of the model. Together, these components form the backbone of successful ML outcomes, balancing model complexity with predictive accuracy.

Gradient Descent

The gradient descent method is the most widely used algorithm to train ML models by minimizing errors between predicted and actual results. Its basic idea is to minimize the loss function by iteratively moving in the direction of the steepest descent as defined by the negative of the gradient of the model parameters, such as weight matrices and biases, shown in Fig. 5.10. And we hope that the iteration stops when it reaches the global minimum position. The general equation of gradient descent for multiple dimensions follows:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) , \quad (5.13)$$

where $\boldsymbol{\theta}$ represents parameters of the model with subscript n and $n + 1$ referring to the number of iterations, η denotes the learning rate (also called step length), \mathcal{L} is the loss function, and $\nabla_{\boldsymbol{\theta}}$ means the derivative with respect to parameters $\boldsymbol{\theta}$.

Type of Gradient Descent

Usually there are three popular types of gradient descent that mainly differ in the amount of data we use:

- **Stochastic Gradient Descent:** Stochastic Gradient Descent (SGD) calculates the gradient and updates the model parameters using just **one** randomly chosen

training sample. This random selection adds an element of unpredictability to the optimization process, hence the term “stochastic”. It is computationally more efficient than the conventional gradient descent method because it processes only one data point at a time. However, SGD tends to be noisier than standard gradient descent and often requires more iterations to achieve the minimum due to this inherent randomness.

- **Batch Gradient Descent:** Batch gradient descent sums the error for each point in a training dataset, updating the model only after all the training samples have been evaluated. And this process refers to as a *training epoch*. The advantage of this method is that it generates a stable error gradient and a stable convergence. However, it could be slow and computationally expensive when dealing with large datasets and complex models. Besides, it can get stuck in shallow local minima or saddle points where the gradient is either very small or zero.
- **Mini-Batch Gradient Descent:** Mini-batch gradient descent uses a subset of a training set, called mini-batch, to compute the gradient at each iteration. This approach strikes a balance between the computational efficiency of batch gradient descent and the speed of stochastic gradient descent. The size of the mini-batch can vary, acting as a hyperparameter. Note that we should tell the difference between epoch and iteration here. One epoch refers to one complete pass through the entire training dataset, while iteration denotes the process of using one mini-batch to compute the gradient. For example, there are 95 training samples in total and we choose 10 as the size of mini-batch. Then each epoch contains 10 iterations.

Learning Rate

Learning rate is an important hyperparameter that affects the speed of convergence and stability. Traditionally, people set learning rate as a fixed value, meaning we need to start at high learning rate and manually change it by steps or by some learning schedule. A small learning step at the onset would lead to very slow convergence, while a very high rate at the start might miss the minima. Then, *Adam* [40] optimization algorithms are proposed to improve convergence speed and stability. Let’s explain this algorithm step by step, starting from the momentum to the final Adam.

1. **SGD:** SGD method follows the constant learning rate and it’s described by Eq. (5.13), moving towards the opposite direction of greatest gradient. The effect of different learning rates are shown in Fig. 5.10.
2. **Momentum:** When using SGD, we might meet the situations where the learning process follows a zigzag motion, causing the model to converge too slowly. This is because the direction of gradient changes greatly after each iteration. By introducing momentum [41], we take previous gradients into account, which largely cancels out the oscillation part and moves as directly as possible in one direction. If we are lucky to have that a gradient has been consistently pointing in the same direction, and then the momentum term proportional to the previous gradients will also accumulate and

accelerate the optimization in that direction.

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta \mathbf{v}_n , \quad (5.14)$$

$$\mathbf{v}_n = \beta \mathbf{v}_{n-1} + (1 - \beta) \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) , \quad (5.15)$$

where \mathbf{v}_n and \mathbf{v}_{n-1} represent current and previous momentum, and β serves as a hyperparameter that controls the extent to which the previous momentum affects the current update.

3. **AdaGrad**: Adaptive Gradient (AdaGrad) [42] is the algorithm allowing parameters to have their own distinct learning rates. This method adjusts the learning rate dynamically during the training process, based on the historical accumulation of the squared gradients for each parameter. Mathematically, this could be described by:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \frac{\eta}{\sqrt{G_n + \epsilon}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_n) , \quad (5.16)$$

$$G_n = G_{n-1} + (\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_n))^2 , \quad (5.17)$$

where G_n represent the sum of squares of all gradients in history, and ϵ is a very small constant preventing the denominator to be zero.

4. **RMSprop**: AdaGrad faces a significant issue due to its accumulation of all past gradients, which results in an increasingly smaller learning rate. This continual reduction can dramatically slow down the convergence rate as training progresses. To solve this problem, *RMSprop* [43] progressively forgets past gradients by multiplying them with a factor less than 1, effectively performing an “exponential moving average”, which exponentially decreases the influence of past gradients. The equation is basically the same as Eq. (5.16), but changes the term G_n into:

$$G_n = \alpha G_{n-1} + (1 - \alpha) (\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_n))^2 , \quad (5.18)$$

where α is the decay rate, similar to the momentum term, representing that we gradually forget the contribution from earlier gradients.

5. **Adam**: The Adaptive Momentum (Adam) algorithm integrates the benefits of both momentum and RMSprop. It employs momentum to enhance the directionality of the gradient updates and utilizes the principles of RMSprop to optimize the learning rate. This combination allows Adam to effectively adjust its updates, promoting faster and more stable convergence in training neural networks. The equation is represented as follows:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \frac{\eta}{\sqrt{G_t + \epsilon}} \mathbf{v}_n , \quad (5.19)$$

$$G_n = \alpha G_{n-1} + (1 - \alpha) (\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_n))^2 , \quad (5.20)$$

$$\mathbf{v}_n = \beta \mathbf{v}_{n-1} + (1 - \beta) \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_n) . \quad (5.21)$$

In this project, we utilized Adam algorithm to do the optimization.

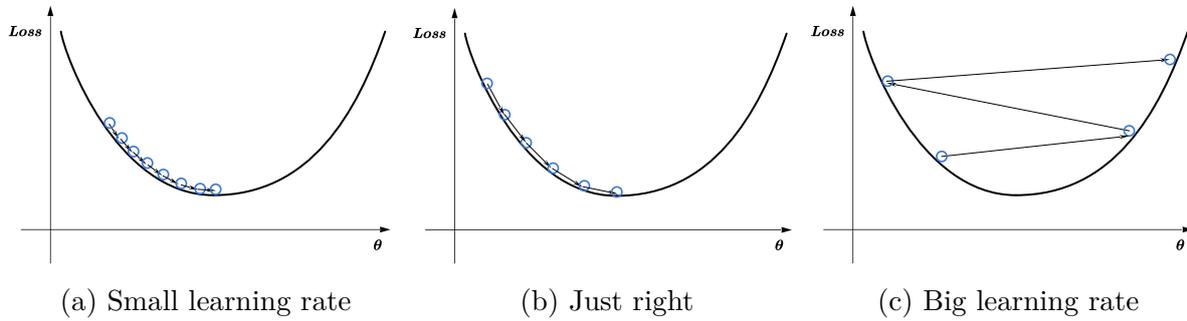


Figure 5.10: Gradient descent with different learning rates. (a) illustrates the gradient descent with a very small learning rate, resulting in a long time to reach minima. (b) uses just right step length and reaches minima with fast speed. (c) demonstrates the effect of big learning rate, which causes the divergence.

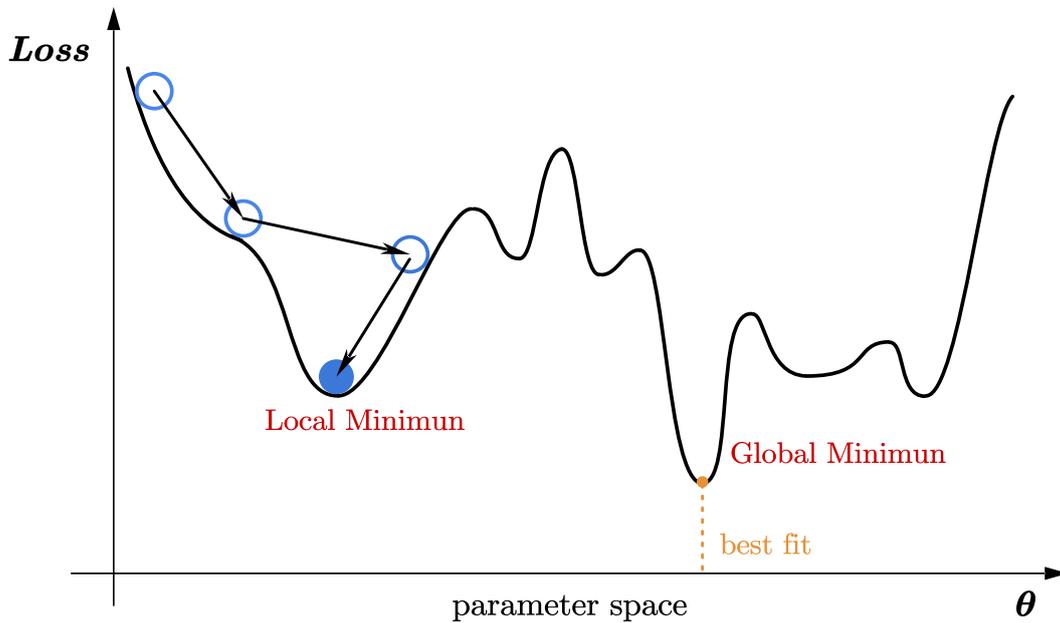


Figure 5.11: Illustration of the loss function landscape in parameter space. The optimization process (e.g., gradient descent) can get stuck in a local minimum (blue) instead of reaching the global minimum (orange), which corresponds to the best fit to the data. The vertical dashed line indicates the location of the global minimum in parameter space.

Regularization

Regularization is a method employed in ML to curb overfitting and keep, at the same time, the training error as low as possible, enhancing models' ability to generalize to new, unseen data. It discourages the model from fitting the training data too closely and promotes simpler or more regular patterns in the learned parameters. By doing so, regularization helps the model avoid learning noise or insignificant details present in the training dataset, ultimately improving its accuracy on test data. Here we will introduce three widely used regularization methods: dropout [44], batch normalization [45] and early stopping [46].

- **Dropout:** Dropout involves randomly ignoring or dropping out some layer outputs during training. Dropout is implemented per-layer in a neural network and each node in the layer will be ignored with probability p for every iteration, and this essentially removes them temporarily from the network. The reasons why dropout can solve overfitting follow.
 - There is an averaging effect. First, let's return to the standard model without dropout. We use the same training set to train different neural networks, and we could get different predictions with these models. Then we could take the average of these predictions to gain the final result. Because different models might have different overfittings, and the operation of average could help to cancel out the overfittings with opposite directions, which contributes to the generalization. Dropout method ignores different nodes during the training process, which is similar to training different models. Because removing the nodes randomly has greatly changed the neural networks, the entire process of dropout is almost equivalent to taking the average of different neural network models.
 - Reduce complex co-adaptations between neurons. The dropout process means that two neurons are not necessarily present in the same dropout network every time. As a result, the update of weights no longer depends on the joint action of hidden nodes with a fixed relationship, preventing the situation where certain features are only effective under specific other features. This forces the networks to learn more robust features, which exist in random subsets of other neurons. In other words, if our neural network is making a certain prediction, it should not be overly sensitive to specific clues or fragments. This shows that we should not give a node too much weight, meaning that each node needs to make similar contributions, rather than some certain nodes making excessive contributions. Even if it loses specific clues, it should be able to learn some common features from many other clues.
- **Batch Normalization:** Batch normalization process involves calculating the mean and variance of each feature in a mini-batch and then scaling and shifting the features using these statistics so that they have *zero* mean and *one* variance value.

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}, \quad (5.22)$$

where μ is the mean value, $\mu = \frac{1}{N} \sum_i x_i$, σ^2 is the variance of this feature, $\sigma^2 = \frac{1}{N} \sum_i (x_i - \mu)^2$, and ϵ is a small constant that prevents division by 0. Since the original input data might cover many magnitudes, this leads to a zigzag path when implementing gradient descent. Batch normalization ensures that the input to each layer remains roughly the same order of magnitude, regardless of changes in the distribution of earlier layers' output. Consequently, this helps in stabilizing the training process, enabling higher learning rates and faster convergence.

- **Early Stopping:** Early stopping involves stopping the training process before the model starts to overfit, usually referring to stop before the given number of epoch. When the training reaches a certain level, the error on the training dataset continues to decrease while increase on the testing dataset. Ideally, we hope to stop at that point. In practice, we terminate the training when the testing error has not decreased for a certain number of consecutive steps.

Hyperparameter

This section, we will delve into hyperparameters, which are crucial settings established before training phase begins, distinct from model parameters which evolve during training. Hyperparameters are pivotal as they shape the learning path and overall outcomes of the model, affecting how quickly and effectively a model can learn from data. These settings are fundamental to the structure and optimization of learning algorithms, influencing everything from convergence speed to model accuracy and robustness. In the following, we are going to give some brief introductions about the hyperparameters we optimized in this project.

- **Batch Size:** Batch size refers to the number of training samples used in one iteration. It affects model's convergence rate, memory efficiency and training stability. A larger batch size provides a more accurate estimate of the gradient, but it requires more memory and is less efficient. Conversely, a smaller batch size offers a faster computations and can help the model escape local minima, though it may lead to a noisier gradient estimate, affecting the stability of the convergence.
- **Number of Layers:** The number of hidden layers in a neural network, often referred to as the depth of the network, determines its capacity to learn complex patterns. More hidden layers allow the network to capture higher levels of abstraction in the data, which can be crucial for tasks involving complex data structures. However, increasing the number of hidden layers can also make the network prone to overfitting, especially if there is not enough data to support the increased model complexity.
- **Number of Nodes:** Nodes, also known as neurons, are the fundamental units of computation in a neural network. The number of nodes in each layer impact the capacity of the layer to represent different features of the input data. More nodes can allow for a more detailed and nuanced learning of data intricacies but can also lead to overfitting and increased computational demand. Balancing the number of

nodes is crucial for optimizing performances without unnecessarily complicating the model.

- **Learning Schedule:** The concept of learning schedule, often referred to as learning rate schedule or learning rate decay, adjusts the learning rate according to a predefined schedule or set of rules. It modifies the learning rate dynamically based on either the number of epochs completed or the performance of the model on the training or validation set. Early in training, a larger learning rate can expedite convergence by taking larger steps. As training progresses, reducing the learning rate can help the model fine-tune its parameters more delicately, avoid overshooting minima. If the learning schedule is too short, then we might not get the optimized model but the one with error oscillating around the minima. And if the learning schedule is too long, it takes us too much time on useless training since the learning rate would be super small and it takes tons of time to reach the minimal position of the error.

Each of these hyperparameters must be carefully tuned to balance the trade-off between training efficiency, model complexity and generalization ability. In this project, we employ `optuna` [47], a python package, to tune the hyperparameters and find the optimal configuration that offers the best performance on unseen data.

Chapter 6

Validation on CosmoGridV1 simulation

Having introduced the conceptual framework of weak lensing and the scattering transform, we now move on to describe the simulation dataset used throughout this work. In particular, we will detail how weak lensing convergence fields are constructed from N -body simulation and how we incorporate realistic observational effects such as intrinsic alignments, shape noise, and multiplicative bias into the simulated data. This simulation dataset provides the basis for testing our scattering transform method and validating the cosmological parameter constraints derived from weak lensing observables.

This chapter is organized as follows. In Section 6.1, we describe the basis setting for CosmoGridV1 simulation. In Section 6.2 we will demonstrate how the weak lensing fields are constructed from N -body simulation. And in Section 6.3, some systematic effects would be discussed and we will include these effects for later calculation. Finally, we will present how to extract square maps from the full sky maps in Section 6.4.

6.1 CosmoGridV1 simulation

In this project, we use the public available CosmoGridV1 Simulations [48], which are a set of N -body simulations with baryonic feedback. Specifically, the baryonic feedback was introduced by modifying the particle positions inside a dark matter-only N -body simulation snapshot and the detailed will be discussed in Section 6.3.1. The boxsize for this simulation is $V = 900^3 (\text{Mpc}/h)^3$ and number of particles is $n_{\text{part}} = 832^3$ for each box. The simulations vary six parameters:

- Ω_m : matter density parameter today, which is the sum of the baryon, cold dark matter, and neutrino densities, $(\Omega_m + \Omega_b + \Omega_\nu)$. The neutrino mass is fixed to three degenerate neutrinos, each with $m_\nu = 0.2 \text{ eV}$.
- σ_8 : matter clustering amplitude with a top-hat filter of radius $8 \text{ Mpc}/h$.
- w_0 : dark energy equation of state.

- n_s : spectral index of the primordial power spectrum.
- Ω_b : baryon density parameter today.
- H_0 : Hubble constant.

The distributions of these parameters are shown in Fig. 6.1. Most of the distributions have square shapes, but for $\sigma_8 - \Omega_m$, $w_0 - \Omega_m$ and $w_0 - \sigma_8$ planes, there are additional restrictions on the prior, which remove some combinations of these parameters from the simulation set. This is because the w CDM model effectively accounts for the relativistic fields, including neutrinos, and is inherently similar to the phantom crossing in dark energy models. The boundaries of the prior are shown in Table 6.1.

The variable parameters were sampled on a 6-dimensional Sobol sequence. Further, the grid is divided into *wide* and *narrow* priors, split evenly. The fiducial cosmology is listed in Table 6.1. And Fig. 6.2 shows the redshift distribution [49] used for the projection in CosmoGridV1.

	fiducial	Δ fid.	wide grid prior	narrow grid prior
Ω_m	0.26	± 0.01	$\in [0.10, 0.50]$	$\in [0.15, 0.45]$
σ_8	0.84	± 0.015	$\in [0.40, 1.40]$	$\in [0.50, 1.30]$
w_0	-1	± 0.05	$\in [-2.00, -0.33]$	$\in [-1.25, -0.75]$
n_s	0.9649	± 0.02	$\in [0.87, 1.07]$	$\in [0.93, 1.00]$
Ω_b	0.0493	± 0.001	$\in [0.03, 0.06]$	$\in [0.04, 0.05]$
H_0	67.3	± 2.0	$\in [64.0, 82.0]$	$\in [65.0, 75.0]$

Table 6.1: CosmoGridV1 simulation parameters and their priors. Parameter Ω_m , σ_8 and w_0 have their additional restrictions beyond the box prior here, as shown in Fig. 6.1.

6.2 Lightcone construction

The simulation can be ran in a *snapshot* or *lightcone* mode, where the first one requires to store the full particle position at a set of given timestep and the latter one is based on thin particle shells at the given cosmic redshift [52, 53]. After the simulation are completed, a set of tomographic survey maps can be created either using Born approximation or ray tracing technique [54, 55]. These maps are created by integrating the shell particles density against the relevant probe kernel, which is typically a function of the redshift distribution of selected galaxy sample $n(z)$ and cosmological parameters. In CosmoGridV1, lightcone construction and Born approximation are adopted for full sky map generation.

The lightcone is constructed by concentrically stacking the shells at different redshift of the replicated density field around the observer who is standing at $z = 0$. In order to construct a lightcone spanning a large survey volume, CosmoGridV1 utilizes a *shell permutation* scheme to create a high resolution of the simulated density field across the full survey volume. This is achieved by dividing the lightcone into groups of shells, with

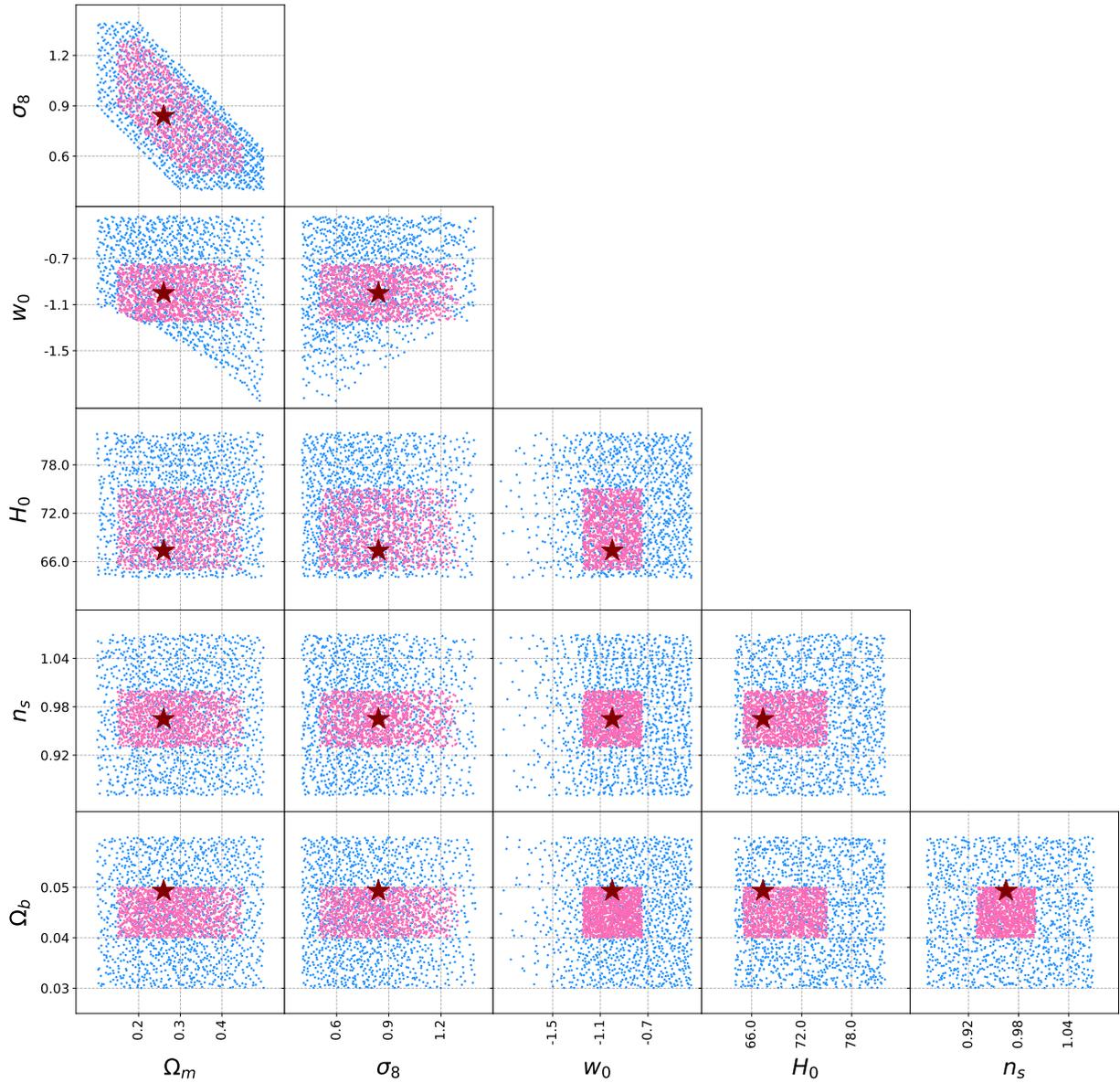


Figure 6.1: Distribution of the parameters of 2500 different cosmology simulations. For each cosmology, there are 7 different realizations. The blue and pink parameters belong to the “wide” and “narrow” grid respectively. The red stars represent the fiducial cosmology, which contains 200 realizations.

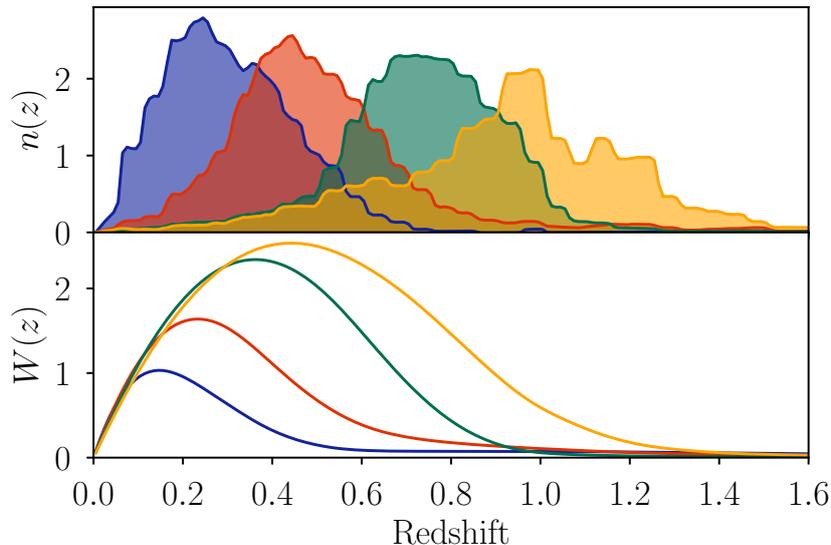


Figure 6.2: Estimated redshift distributions and lensing kernels for the fiducial source galaxy sample used in this analysis. The DES Y3 cosmic shear signal is primarily sensitive to large-scale structure in the redshift range $z = 0.1$ – 0.5 , where the individual lensing kernels peak. Each distribution is normalized independently over the range $z = 0$ – 3 . The total effective number density of source galaxies is $n_{\text{eff}} = 5.59 \text{ arcmin}^{-2}$ [50], divided approximately equally among the four redshift bins. Credit from [51].

each group taken from an independent simulation boxes with different initial conditions. Before putting the simulation boxes into the lightcone, CosmoGridV1 adopt a randomization scheme to increase the number of realizations from one underlying simulation run [56, 57]. It applies random operations to the particle positions or to the δ field on the 3D grids consisting of rotations by 90° (interchanging the axes), translation and parity flips [58]. Such a randomization of the simulation boxes avoids the repetition of the same structure present in the density field along the line of sight, bringing the variance of the maps closer to the true cosmic variance. The groups of the shells are chosen such that the outer z border of each shell in the group is crossing the boundary of the box, making the width of each shell approximately equal to the length of the box, about $900 \text{ Mpc}/h$. The redshift range of this lightcone covers from $z = 0$ to $z = 3.5$ and it typically consists of around 6 replicated boxes. Since there are 7 independent realizations for each cosmology, this further allows for creating lightcones with all the boxes within the same shell coming from unique initial conditions, which ensures that no line of sight will experience the same points in the simulation volume across the multiple replicas. The illustration of this construction is shown in Fig. 6.3.

For the CosmoGridV1 simulations, the methodology follows the framework outlined in [56, 59], where the Born approximation is deemed sufficiently accurate for weak lensing studies on intermediate scales using UFALCON¹ code, as demonstrated by [54, 60]. The 2D projected fields, such as convergence and intrinsic alignment maps, are computed using

¹<https://cosmology.ethz.ch/research/software-lab/UFalcon.html>

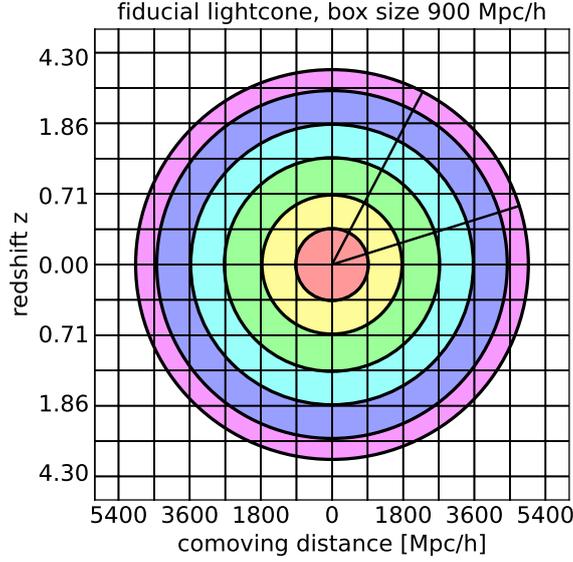


Figure 6.3: Illustration of the full-sky lightcone construction achieved by replicating the density field and applying shell permutation. The observer is positioned at the center of the lightcone, with the density field at discrete redshifts used to form the shells. The square grid represents the periodic boundaries of the simulation boxes for the fiducial cosmology. The lightcone volume covered by different shell groups, each derived from separate simulations with distinct initial conditions, is depicted in various colors. Adapted from [56].

the following expression:

$$m_{2\text{D}}^{\text{pix}} \approx \sum_b W^m \int_{\Delta z_b} \frac{dz}{E(z)} \delta_{3\text{D}} \left[\frac{c}{H_0} \mathcal{D}(z) \hat{n}^{\text{pix}}, z \right], \quad (6.1)$$

where W^m represents the relevant probe kernels, \hat{n}^{pix} is the unit vector pointing to the pixel center, $\mathcal{D}(z)$ is the dimensionless comoving distance, $E(z)$ satisfies $d\mathcal{D} = dz/E(z)$, and Δz_b is the thickness of the shell b . The kernels W for weak lensing (WL) and intrinsic alignment (IA) are defined as in [61]:

$$W^{\text{WL}} = \frac{3}{2} \Omega_m \frac{\int_{\Delta z_b} \frac{dz}{E(z)} \int_z^{z_s} dz' n(z') \frac{\mathcal{D}(z)\mathcal{D}(z,z')}{\mathcal{D}(z')} \frac{1}{a(z)}}{\int_{\Delta z_b} \frac{dz}{E(z)} \int_{z_0}^{z_s} dz' n(z')}, \quad (6.2)$$

$$W^{\text{IA}} = \frac{\int_{\Delta z_b} dz F(z) n(z)}{\int_{\Delta z_b} \frac{dz}{E(z)} \int_z^{z_s} dz' n(z')}, \quad (6.3)$$

where $n(z)$ is the galaxy redshift distribution, z_s and z_0 are the source and observer redshifts, respectively. The function $F(z)$, which depends on cosmology and redshift, is given by:

$$F(z) = -C_1 \rho_{\text{crit}} \frac{\Omega_m}{D_+(z)}, \quad (6.4)$$

where $C_1 = 5 \times 10^{-14} h^{-2} M_\odot \text{Mpc}^3$ is a normalization constant, ρ_{crit} is the critical density at $z = 0$, and $D_+(z)$ is the normalized linear growth factor with $D_+(0) = 1$.

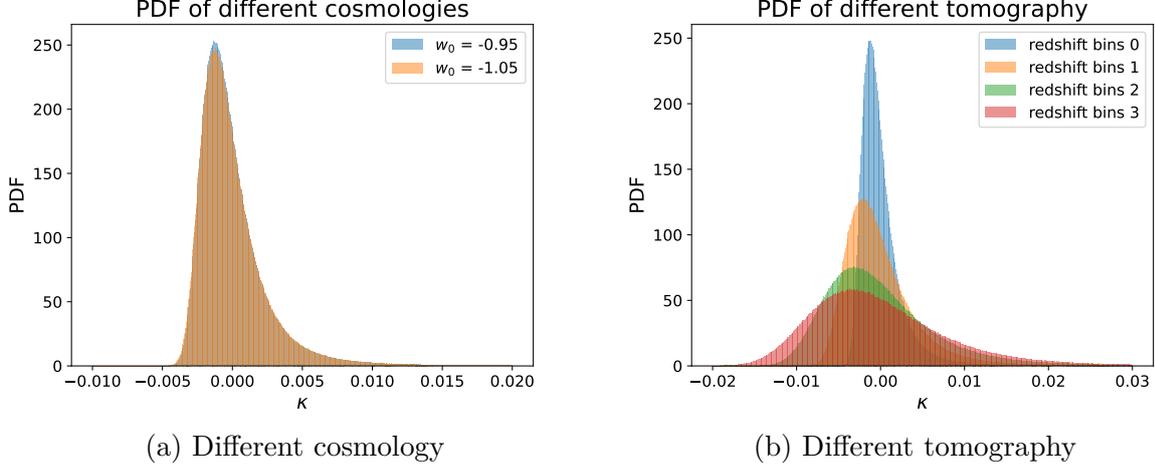


Figure 6.4: Illustration of the probability density distribution (PDF) of the convergence maps. The left panel shows the PDF of the convergence maps for different cosmologies of same initial random seed at latest redshift bin, and the cosmological parameters are basically the same as fiducial cosmology, with only w_0 being slightly changed. While the right panel shows the PDF of the same convergence maps but different tomographic bins. From tomographic bins 0 to 3, the redshift increases.

Fig. 6.5 shows the full-sky map of weak lensing convergence and intrinsic alignment projected under this lightcone construction. And Fig. 6.4 shows an example of how the PDF changes with different cosmological parameters. As we can see, with different parameter of the equation of state for dark energy, the peak shifts slightly towards the underdense regions, while the height of the long tail on the right becomes lower for the blue histogram. For different tomographic bins, the PDF of the lowest redshift bin has a higher peak and more localized distribution, while the largest redshift one corresponds to a lower peak and a more extended distribution. Two reasons might account for these:

- At lower redshifts, the distance between the observer and the sources is smaller, meaning that the convergence at low redshift integrates over a shorter portion of the large-scale structure compared to higher-redshift bins. A shorter integration path implies that fluctuations in the mass distribution have less chance to accumulate, leading to smaller fluctuations in κ .
- The lensing efficiency function, which weights how much a mass fluctuation contributes to κ , is lower for lenses close to the observer (low redshift), especially if the sources are not far behind them. This weaker efficiency kernel further suppresses the amplitude of κ fluctuations.

Overall, these results highlight the interplay between redshift, lensing efficiency, and the integration path in shaping the convergence PDF.

Another thing we should pay attention to in Fig. 6.4b is the non-Gaussianity of the convergence maps. As the redshift increases, the PDF becomes broader and more symmetric, indicating that the field approaches Gaussianity at higher redshifts. In contrast, the lowest-redshift bin exhibits a sharper peak and noticeable skewness toward low κ , reflecting stronger non-Gaussianity due to the enhanced nonlinear structure formation at late times. This trend is consistent with the expected time evolution of gravitational clustering [62].

6.3 Systematic effects

6.3.1 Baryonic feedback

On small scales, the distribution of matter density is significantly influenced by baryonic feedback effects [64], which include processes such as gas cooling, star formation, and feedback from Active Galactic Nuclei (AGN). These effects introduce uncertainties that grow more pronounced at smaller scales. Simulating these effects using traditional hydrodynamic methods is computationally intensive and time-consuming. To address this, a method called *baryonification* [65] has been developed. This approach models baryonic feedback by transforming the density distribution from dark matter-only simulations into one that accounts for baryonic effects. The baryonic feedback is incorporated by adjusting particle positions in snapshots of dark matter-only n -body simulations. The density field is described as a combination of dark matter halos influenced by 1-halo and 2-halo terms. The dark matter-only (dmo) field is expressed as:

$$\rho_{\text{dmo}}(r) = \rho_{\text{NFW}}(r) + \rho_{2\text{h}}(r) , \quad (6.5)$$

where ρ_{NFW} represents a generalized Navarro-Frenk-White (NFW) profile [66], which depends on the halo's virial mass $M \equiv M_{200}$ and concentration $c \equiv c_{200}$, defined within a radius where the density is 200 times the average density. The baryonification model (dmb) modifies the 1-halo term to include contributions from collisionless matter (clm), gas (gas), and the central galaxy (cga). The ρ_{clm} term primarily represents dark matter but also includes satellite galaxies and intercluster stars. The total baryonified density is then given by:

$$\rho_{\text{dmb}}(r) = \rho_{\text{clm}}(r) + \rho_{\text{gas}}(r) + \rho_{\text{cga}}(r) + \rho_{2\text{h}}(r) . \quad (6.6)$$

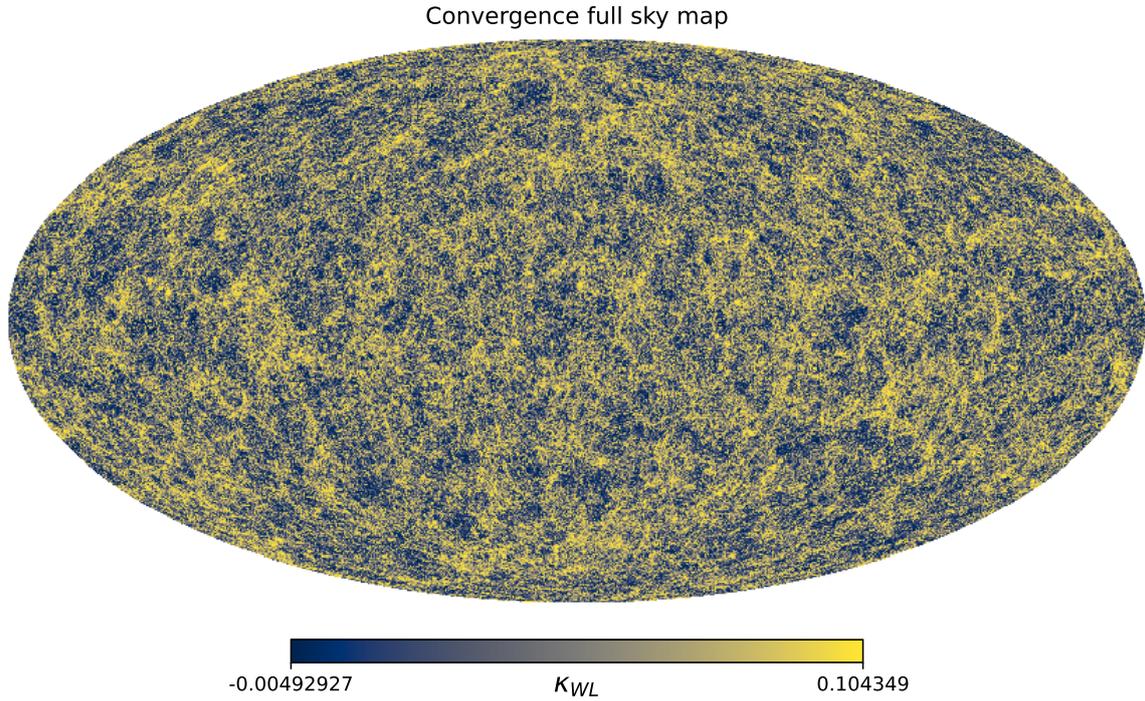
The baryonification process transforms the density ρ_{dmo} into ρ_{dmb} by first calculating the integrated mass profile within a radius r :

$$M_{\chi}(r) = \int_0^r s^2 \rho_{\chi}(s) ds . \quad (6.7)$$

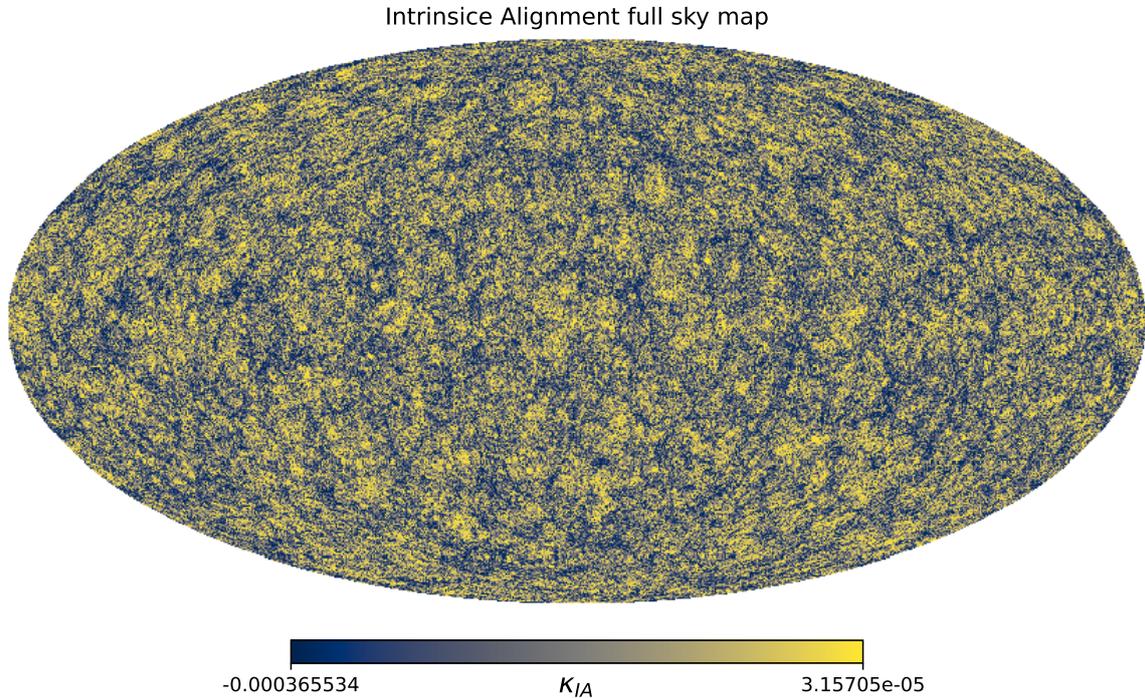
This profile is a bijective function, enabling the definition of a displacement function as:

$$d(r_{\text{dmo}}|M, c) \equiv r_{\text{dmb}}(M) - r_{\text{dmo}}(M) . \quad (6.8)$$

To apply the baryonification model to projected data, CosmoGridV1 employs an efficient method called *shell baryonification* [48]. Instead of modifying particle positions in 3D,



(a) Convergence map



(b) Intrinsic Alignment map

Figure 6.5: Projected fields of the convergence and intrinsic alignment full-sky maps. The maps are generated using lightcone construction and the Born approximation based on CosmoGridV1 N -body simulations of resolution $NSIDE = 512$. Visualization is done using the `healpy` package [63], with a histogram-based color normalization to enhance contrast.

this method calculates a projected displacement function based on the difference between baryonified and dark matter-only profiles and applies it to 2D shell particle count maps. The projected integrated mass profile is computed as:

$$M_{\chi}^p(r) = 2\pi \int_0^r s \int_0^{z_{\max}} \rho_{\chi}(s, z) dz ds , \quad (6.9)$$

with $z_{\max} = 50r$. The projected displacement is then given by:

$$d^p(r_{\text{dmo}}|M) = r_{\text{dmb}}(M^p) - r_{\text{dmo}}(M^p) . \quad (6.10)$$

This displacement is applied to pixel positions using interpolation, effectively redistributing mass to account for baryonic effects, such as the suppression of central halo densities and the redistribution of matter to larger radii. In CosmoGridV1 simulation, the baryonic feedback is already incorporated into the simulation and thus we do not have to modify this manually.

6.3.2 Intrinsic alignment

In weak lensing studies, the shapes of distant galaxies are used to infer the underlying matter distribution through the shear field. However, the observed shear is not only affected by the foreground matter distribution, but also contaminated by the intrinsic ellipticities of galaxies. This arises from the fact that galaxies do not orient themselves randomly with respect to the large-scale structure gravitational potential in which they form — a phenomenon known as *intrinsic alignment* (IA). Tidal gravitational fields can induce correlations in galaxy orientations, independent of lensing. These alignments introduce spurious correlations that can mimic or bias the cosmic shear signals, posing a significant challenges for precision cosmology.

To mitigate IA contamination, several theoretical IA models have been proposed in the literature, such as *Linear Alignment (LA) Model* [67], *Non-Linear Alignment (NLA) Model* [68, 69, 70] and *Perturbation theory-based model* [71, 72]. In LA model, intrinsic shape of a galaxy is assumed to be linearly related to the large-scale tidal field present during its formation. By incorporating the non-linear matter power spectrum, the NLA model is able to account for the non-linear evolution of the tidal field. A more complicated perturbation theory-based model uses perturbation theory of structure formation and the bias expansion for cosmological tracers, like *Tidal Alignment Tidal Torque (TATT) Model* [73]. The TATT model further extends this framework by combining two physical mechanisms: tidal alignment, which dominated for elliptical galaxies, and tidal torquing, which affects disk galaxies through angular momentum acquisition. This more flexible formulation allows the TATT model to capture a broader range of galaxy alignment behaviors and has become increasingly important for modeling IA in tomographic cosmic shear studies.

When applying the IA model to the convergence maps, we directly add the IA convergence maps into the simulation convergence with a random amplitude A_{IA} :

$$\kappa_{\text{obs}} = \kappa_{\text{sim}} + A_{IA} \cdot \kappa_{IA} , \quad (6.11)$$

where κ_{obs} is the observed convergence, κ_{sim} is the simulated convergence, κ_{IA} is the IA convergence, and A_{IA} is drawn from a uniform distribution, $\mathcal{U}[-5, 5]$, with $A_{\text{IA}} = 0$ for fiducial cosmology.

A more mathematical explanation of how IA affects the observed shear is provided in Appendix E. In the case of CosmoGridV1, it utilized NLA model to create IA convergence maps using the W_{IA} kernel as described in Eq. (6.3) and we need to add IA maps to weak lensing convergence maps manually if we want to include this systematic effect.

6.3.3 Multiplicative bias

In weak lensing studies, the observed shear measurements can be systematically biased due to the inaccuracies in the measurement process. One such systematic effect is the *multiplicative bias*, which arises when the measured shear is systematically scaled by a factor relative to the true shear. This bias can originate from various sources, including:

- **Noise Bias:** Measurements noise in galaxy shapes can lead to biased shear estimates, especially when the noise is correlated with the galaxy’s ellipticity.
- **Model Bias:** Simplifications or inaccuracies in the methods used to fit galaxy shapes can introduce systematic errors in the shear estimates.
- **Selection Bias:** The process of selecting galaxies for shear measurement can preferentially include or exclude certain types of galaxies, leading to a biased sample.

The multiplicative bias is typically parameterized as:

$$\gamma_{\text{obs}} = (1 + m)\gamma_{\text{true}} , \quad (6.12)$$

where γ_{obs} is the observed shear, γ_{true} is the true shear and m is the multiplicative bias parameter. Accurate calibration of m is essential for precision cosmology, as even small biases can affect the inferred cosmological parameters.

Since the CosmoGridV1 simulation does not account for this bias, we manually incorporate it in this project. To make the convergence maps more realistic and comparable to DES Y3 data, we adopt the multiplicative bias parameters from the DES Y3 survey [50]. All the multiplicative bias coefficients follow the Gaussian distribution, $\mathcal{N}(0.0063, 0.0091)$, $\mathcal{N}(0.0198, 0.0078)$, $\mathcal{N}(0.0241, 0.0076)$, $\mathcal{N}(0.0369, 0.0076)$ for tomographic bins from the latest to the earliest redshift.

6.3.4 Shape noise

The convergence maps generated by CosmoGridV1 are free of noise, implying that the galaxies in real observations are assumed to have perfectly circular intrinsic shapes, and measurement uncertainties are ignored. However, in reality, galaxies possess intrinsic ellipticities and are randomly oriented across the sky. This introduces what is known as *shape noise* into the shear measurements, which subsequently affects the convergence

maps.

In order to make the convergence maps more realistic and make the covariance matrix more accurate for scattering transform, we need to add shape noise to the convergence maps. The shape noise is generated by randomly sampling the intrinsic ellipticities of galaxies, which are assumed to follow a Gaussian distribution with zero mean and a standard deviation as follow:

$$\sigma_{\text{pix}} = \frac{\sigma_e}{\sqrt{A \cdot n_{\text{eff}}}}, \quad (6.13)$$

where σ_e is the corresponding shape noise parameter, A represents the area of each pixel and n_{eff} is the effective galaxy number density (gal/arcmin²) in the pixel. The total value is $n_{\text{eff}} = 5.590$ and $\sigma_e = 0.268$, and the values for each tomographic bin are $n_{\text{eff}} = [1.476, 1.479, 1.484, 1.461]$ and $\sigma_e = [0.243, 0.262, 0.259, 0.301]$ from DES Y3 data [50] for redshift from latest to earliest. And because of the resolution of CosmoGridV1 simulation, $NSIDE = 512$, the area A of each pixel is approximately 6.9×6.9 arcmin².

Since we could only observe shear fields and get the ellipticity data from galaxies, the conclusion that the noise on shear field is the same on convergence field is not that obvious. The detailed mathematical explanation could be found in Appendix E.2.

6.3.5 Photometric redshift uncertainty

Photometric redshift uncertainty is another systematic effect in weak lensing studies. In observational cosmology, the redshifts of galaxies are often estimated using photometric methods, which rely on measurement of galaxy fluxes in multiple broad-band filters. These photometric redshifts (photo- z) are less practice than spectroscopic redshifts, leading to uncertainty that can propagate into weak lensing analyses. The uncertainty in photometric redshifts arises from several factors:

- **Template Mismatch:** The galaxy spectral energy distribution (SED) templates used for photo- z estimation may not perfectly match the observed galaxies.
- **Degeneracies:** Different combinations of galaxy properties, such as age, metallicity, and dust content, can produce similar observed colors, leading to redshift degeneracies.
- **Photometric Errors:** Measurements noise in the observed fluxes can introduce scatter in the estimated redshifts.
- **Catastrophic Outliers:** In some cases, galaxies are assigned redshifts that are significantly different from their true values due to misclassification or extreme degeneracies.

Photometric redshift uncertainties affect weak lensing studies in several ways:

- **Source Redshift Distribution:** The uncertainty in photo- z leads to errors in the estimated redshift distribution of source galaxies, which is critical for calculating lensing kernels.

- **Tomographic Binning:** Errors in photo- z can cause galaxies to be assigned to incorrect tomographic bins, leading to biases in the inferred cosmological parameters.
- **Bias in Shear Measurements:** The lensing efficiency depends on the relative distances between the lens and source galaxies. Photo- z errors can bias this efficiency, affecting the shear signal.

In the context of CosmoGridV1, photometric redshift uncertainties are already included for grid cosmologies so we do not need to model this systematic effect manually. Their distributions also follow the Gaussian distribution, $\mathcal{N}(0, 0.018)$, $\mathcal{N}(0, 0.015)$, $\mathcal{N}(0, 0.011)$, $\mathcal{N}(0, 0.017)$ for small redshift bin to large redshift bin.

6.4 Projected maps

6.4.1 Fibonacci sphere sampling

In cosmology, the `healpy` [63] package is commonly used to visualize full-sky maps. These maps are generated by arranging one-dimensional data in a specific order to produce a two-dimensional representation on the sphere. However, for the scattering transform method, which operates on two-dimensional maps, it becomes necessary to extract rectangle or square patches from these full-sky maps.

To make full use of the simulation as much as possible, using more data and leaving less gaps, we use *Fibonacci sphere* method [74, 75] to generate evenly distributed samples on the sphere. This method is based on the Fibonacci sequence and the golden ratio, and we use this irrational number to generate angular steps, ensuring quasi-uniform and non-repetitive coverage of the sphere. The detailed discussion about Fibonacci sphere is shown in Appendix F.

Using the evenly distributed points on the sphere, we can extract square maps from the full-sky maps. The method employed here is the *Gnomonic Projection* [76], an azimuthal map projection that maps the surface of a sphere onto a tangent plane from a single point at the sphere's center. The centers of these projections correspond to the points sampled from the Fibonacci sphere. Fig. 6.6 demonstrate the extracted square maps from the sphere.

6.4.2 Convergence maps

All the full sky maps, including convergence, IA and shape noise maps, are generated using the same method. Fig. 6.7 shows the extracted maps from different sources. For Fig. 6.7d, the mock map is generated by including multiplicative bias and adding IA map and shape noise:

$$\kappa_{\text{mock}} = (1 + m) \cdot (\kappa_{\text{sim}} + A_{IA} \cdot \kappa_{IA}) + \kappa_{\text{shape noise}} . \quad (6.14)$$

Fig. 6.8 shows the distribution of convergence maps from weak lensing, intrinsic alignment and shape noise. As we can see, the noiseless weak lensing convergence map, the

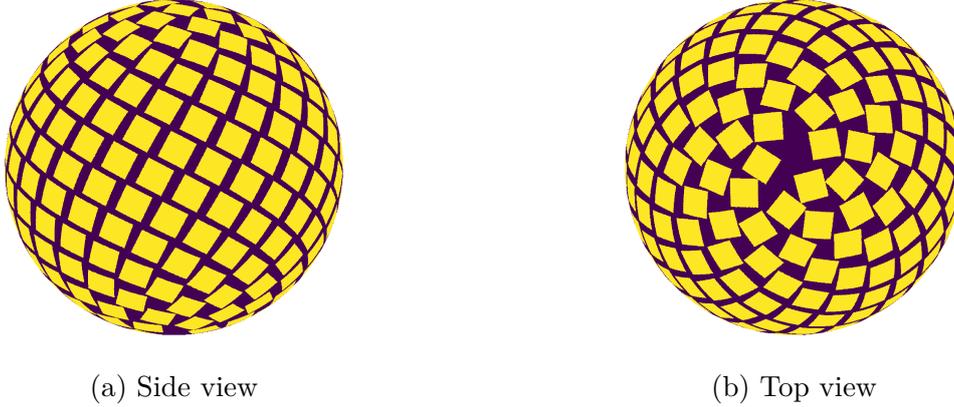


Figure 6.6: Illustration of the Gnomonic projection method. The left panel shows the side view of the projection, where the right panel shows the top view. The yellow squares represent the projected maps we extract from the full-sky maps. Each pixel equals to the resolution of $NSIDE = 512$, which is about 6.9 arcmin. There are 86 pixels for each side of the square. The total size of the square map is $9.85^\circ \times 9.85^\circ$. The side of the square makes an angle of 27.7° with the horizontal axis or the equator. These parameter settings make maximal use of the simulation data, with 260 squares in each full-sky map and covered area of 61.1% without any overlapping.

distribution peaks at some underdense region, with a long tail on the overdense region. While the intrinsic alignment holds an opposite distribution, with a peak at the overdense region and a long tail on the underdense part. What's more, the magnitude of intrinsic alignment is much smaller compared to weak lensing maps. For Fig. 6.8c, it demonstrate the distribution of the Gaussian shape noise and the mock data. Due to the fact that multiplicative bias and intrinsic alignment are included, the distribution of the mock data is slightly shifted compared to the Gaussian field. But their distributions are still similar, indicating that the shape noise dominates in all systematic effects.

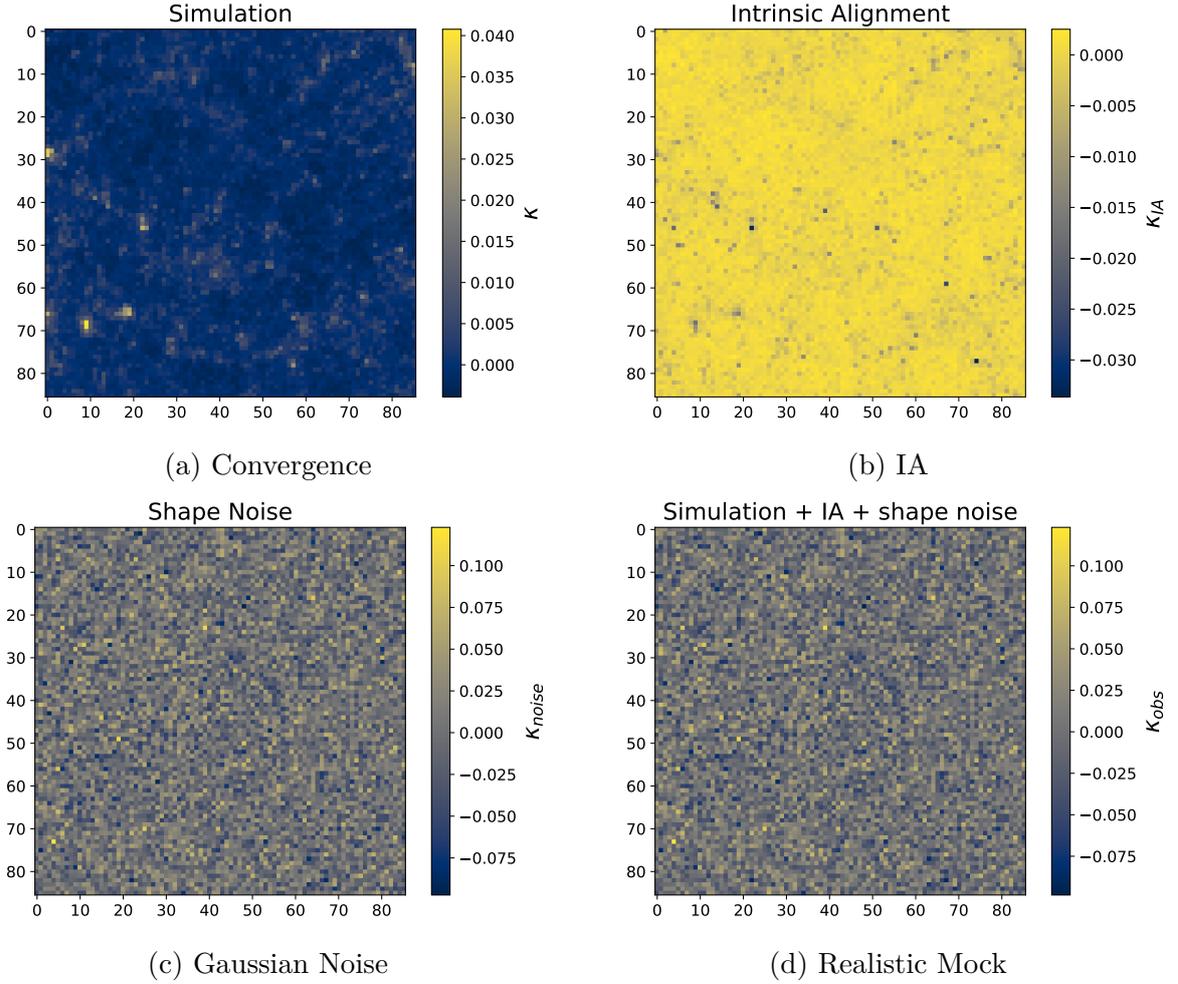


Figure 6.7: Demonstration of the square maps extracted from different sources. (a) is the noiseless weak lensing convergence map from simulation fiducial cosmology, (b) is the intrinsic alignment map, (c) is the Gaussian noise map, and (d) is the map with shape noise, intrinsic alignment included on the simulation mass map, where the strength of the intrinsic alignment is selected to be $A_{IA} = 3$. All the maps are generated within the latest redshift bin.

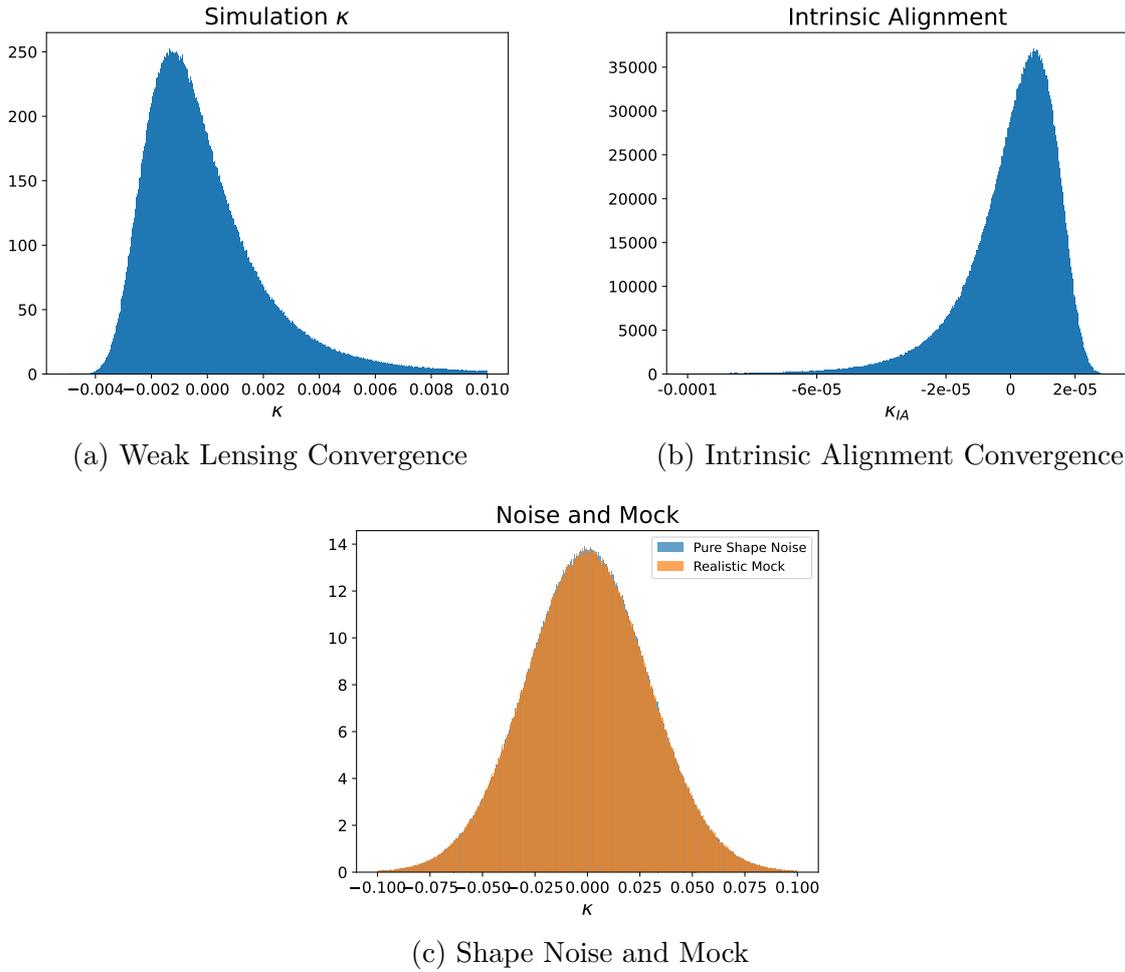


Figure 6.8: Distribution of the different maps. (a) is the distribution of the noiseless weak lensing convergence directly from simulation of fiducial cosmology, (b) is the distribution of the intrinsic alignment convergence, and (c) is the distribution of the pure shape noise and mock map. For the mock map, multiplicative bias, intrinsic alignment and shape noise are all included. And the strength for the intrinsic alignment is chosen to be $A_{IA} = 3$.

Chapter 7

Cosmological parameter constraints with Fisher matrix and MCMC

Having established the theoretical framework and prepared the simulated datasets, we now turn to the core data analysis, aiming to constrain cosmological parameters based on weak lensing convergence field. Our focus is on leveraging the scattering transform to extract high-order statistical information from convergence maps, which serves as the basis for cosmological inference. A key object is to assess how effectively the scattering coefficients, combined with emulator predictions, can constrain the parameters.

This chapter is structured as follows. In Section 7.1, we will show scattering coefficients as well as scattering transform based power spectrum. In Section 7.2, we demonstrate the parameter uncertainty with Fisher forecast and in Section 7.4 we will show the cosmological constraints with scattering transform.

Before delving into the specifics of the aforementioned methods, we first define some common notations used throughout this chapter: the n -dimensional vector of data is represented as \mathbf{x} , with components x_1, x_2, \dots, x_n . This data set is modeled by a theoretical function that depends on a vector of model parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ in a known manner. The model M , which is employed to fit the data \mathbf{x} , produces predictions represented by another n -dimensional vector $\boldsymbol{\mu}$, with components $\mu_1, \mu_2, \dots, \mu_n$.

7.1 Summary statistics

7.1.1 Scattering coefficients

With the square maps of weak lensing convergence well prepared, we can now apply scattering transform operation on them, which generates a series of scattering coefficients under different cosmology scenarios. The visualization of these coefficients below demonstrates the scattering transform's ability to capture the underlying structure of the data.

First, we applied scattering transform operation on square maps of size 300 by 300 pixel². Each pixel has an area of 8×8 arcmin², which is slightly lower than the resolution of the

simulation. Thus, for each square map, it covers $40^\circ \times 40^\circ$ area. We could maximally extract 15 non-overlapping square maps from one full-sky map. 7 realizations (since there are only 7 realizations for each grid cosmology) of full-sky maps are used to generate 105 square maps for the scattering transform. All the maps here are noiseless weak lensing convergence maps generated directly from simulations without including any systematic effects. The scattering transform is performed with $J = 8$, $L = 4$, meaning that we are extracting information from 8 different scales, and the 4 orientations of the wavelets.

Fig. 7.1 shows the scattering coefficients calculated from maps mentioned above with different cosmological parameters. As we can see, the scattering coefficients have similar patterns in both figures, but the magnitudes of the coefficients are different. This indicates that the scattering coefficients are sensitive to the cosmological parameters.

The blue dots are the zeroth order coefficients. Since it represents the spatial average of the square maps, it is supposed to be zero theoretically. However, due to the finite size of the maps, the zeroth order coefficient is not exactly zero. Besides, the error bar of the zeroth order is relatively large, because we could only calculate one zeroth order coefficient from a single map, then limited number of realizations leads to such large error bar.

For the first order coefficients, seen as orange dots in the figure, they are the spatial average of the first order convolved field with filter size increasing from left to right. Since the scattering convolution is a pixel level weighted average, bigger filters cover more pixels, which leads to a smoother field after convolution, gradually erasing the large clustering and underdense. Therefore, the first order coefficients are expected to be smaller and smaller as the filter size increases. The error bars of the first order coefficients are relatively small compared to the zeroth order coefficient, because first order coefficients are the mean value of first order convolved fields, and each value in the pixel is a weighted average of many pixels in the original map, causing a smaller difference of pixels compared to the zeroth order coefficient. The magnitude of first order coefficients in the second figure is smaller than the first one but its error bar is larger. This might be due to the fact that the second cosmology, because of greater value of σ_8 , has a greater clustering effect, which leads to larger clusterings or underdense in the map. The absolute value at each pixel, even though after the filter, still varies a lot from regions to regions.

Then, for the second order scattering coefficients, they are calculated from the second order convolved fields which are just another convolution of the first order convolved fields. The second order coefficients also have similar patterns in both figures, but the magnitudes here are quite similar. This is because some extreme values in the original maps have been smoothed out by the first order convolution, then the second order convolution further smooths the first order convolved fields, which is the reason why the second order coefficients have similar magnitude. This also accounts for second order coefficients are expected to be smaller than the first order coefficients. Besides, the error bars of the second order coefficients are much smaller than previous, this could be explained by the smoothed fields as well.

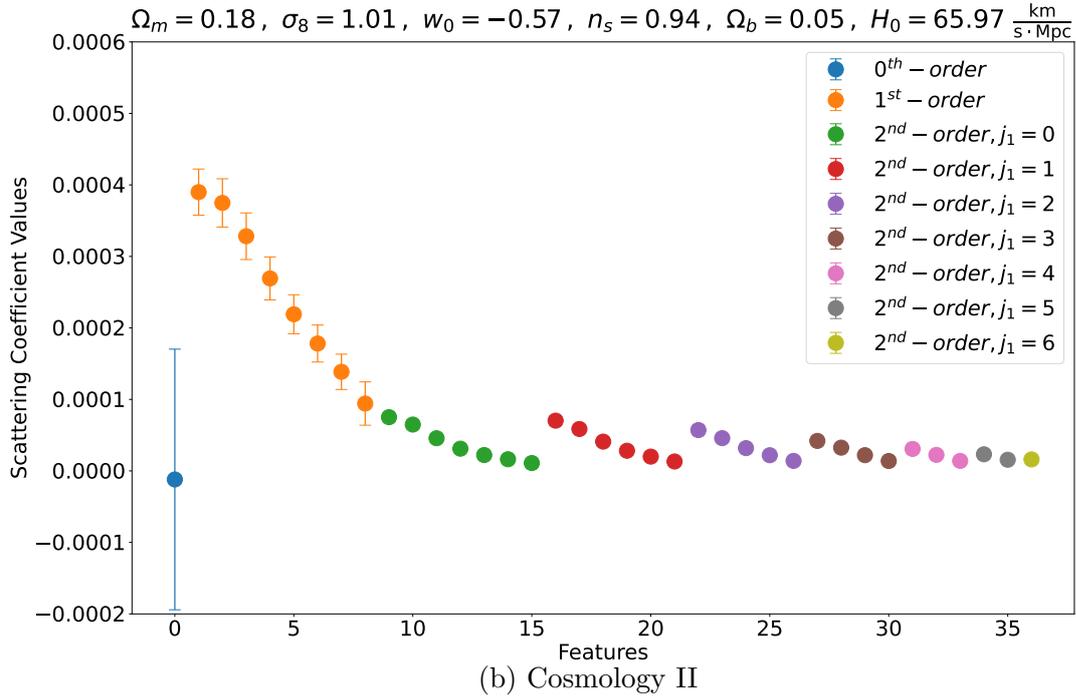
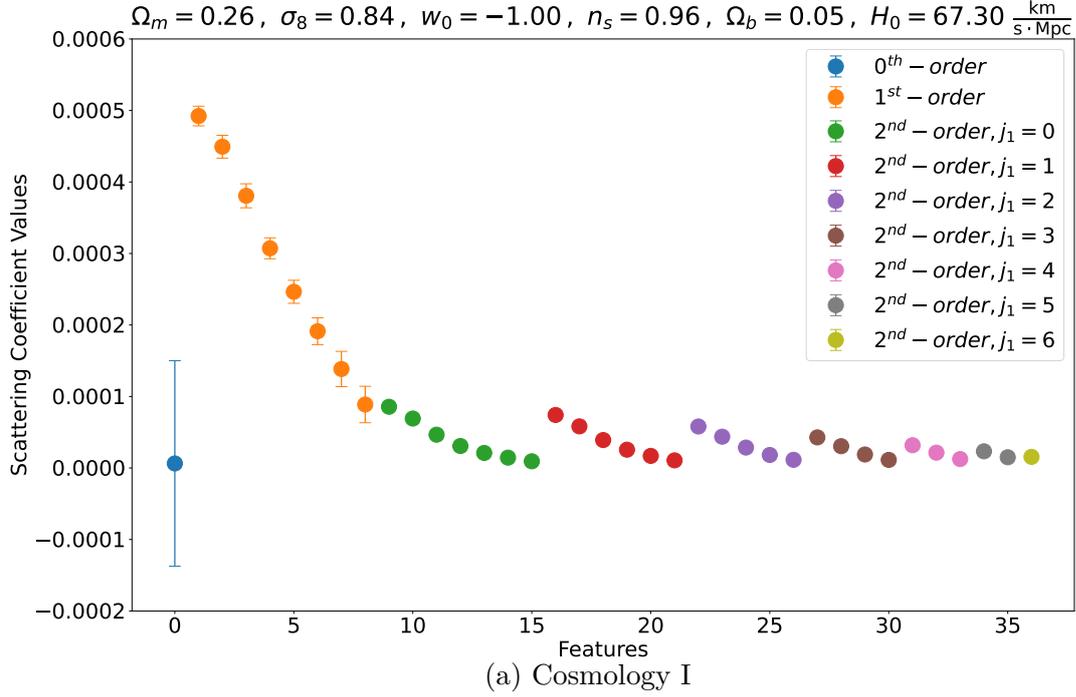


Figure 7.1: Scattering coefficients for two different cosmologies, shown in the top and bottom panels, respectively. Cosmological parameters are listed above each plot. All coefficients are computed from noiseless weak lensing convergence maps of the same tomographic bin. The x -axis indexes the scattering features: one zeroth-order (blue), eight first-order (orange), and 28 second-order coefficients (colored by j_1). Second-order coefficients are grouped by the scale of the first filter (j_1), and vary by the second filter j_2 . For example, green points correspond to $j_1 = 0$ with varying j_2 . The y -axis shows the coefficient magnitude, with error bars denoting standard deviations. Both panels share the same y -range.

Fig. 7.2 demonstrates the scattering coefficients of fiducial cosmology in different tomographic bins using 200 realizations. As expected, the zeroth order coefficients from all tomographic bins are much more closer to zero than the previous figures with only 7 realizations. For higher redshift data, they have higher values of scattering coefficients. This could be explained by the fact that the projection of distribution of matter in higher tomographic bin has a wider profile in their PDF than late universe, shown in Fig. 6.4b.

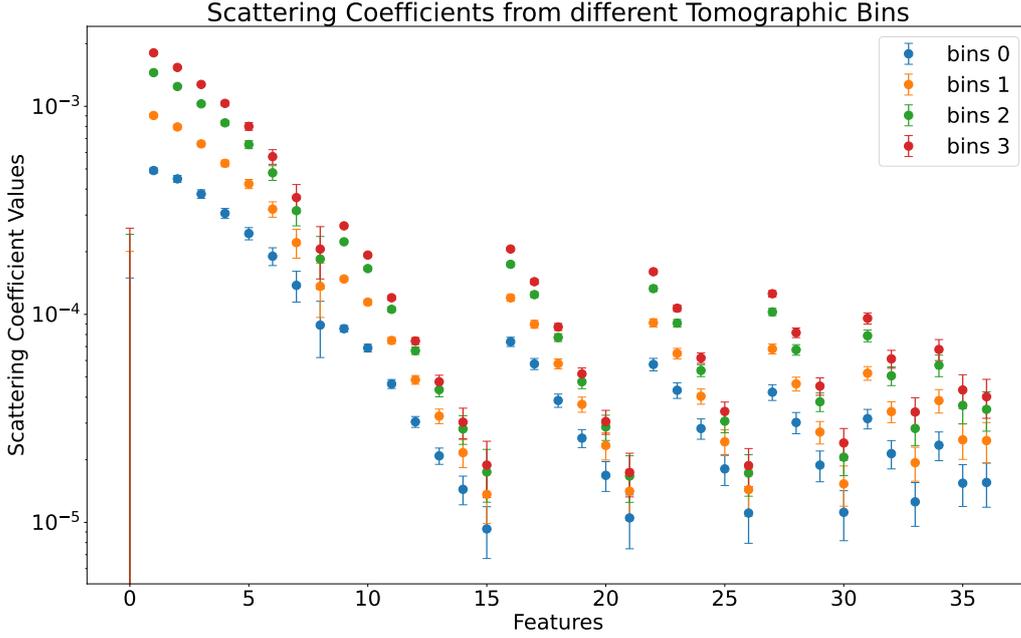


Figure 7.2: Scattering coefficients from different tomographic bins, shown on a logarithmic y -axis due to their wide dynamic range. Bins from 0 to 3 correspond to increasing redshift. The x -axis represents the number of scattering coefficients. $J = 8$ is chosen here, we have 1 for zeroth order, 8 for the first order and 28 for the second order coefficients. Error bars represent the standard deviation of the coefficients.

7.2 Fisher matrix analysis

The objective of parameter estimation is to infer the probability distribution of the model parameters $\boldsymbol{\theta}$ based on a given data set \boldsymbol{x} . In the framework of Bayesian statistics, this probability density reflects the degree of confidence in a specific parameter vector. This posterior probability distribution is given by [77]:

$$P(\boldsymbol{\theta}|\boldsymbol{x}), \quad (7.1)$$

which can be used to compute expectation values, maximum posterior estimates, and parameter uncertainties.

With Bayes's theorem, the posterior probability distribution can be expressed as [77]:

$$P(\boldsymbol{\theta}|\mathbf{x}) = \frac{P(\mathbf{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{x})} , \quad (7.2)$$

where $P(\mathbf{x}|\boldsymbol{\theta})$ is the likelihood function, $P(\boldsymbol{\theta})$ is the prior probability distribution, and $P(\mathbf{x})$ is the evidence. The evidence is a normalization constant that ensures the posterior distribution integrates to one. Usually, the evidence is computationally expensive to calculate due to the high dimensionality of the parameter space:

$$P(\mathbf{x}) = \int P(\mathbf{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta} , \quad (7.3)$$

and it is not necessary for parameter estimation since it just rescales the posterior distribution and has nothing to do with the shape of the posterior. Thus, we can ignore the evidence and just focus on the likelihood function and the prior probability distribution. Then prior is the probability distribution of the parameters before observing the data. It represents our beliefs about the parameters based on previous knowledge or assumptions. If we have no prior knowledge about the parameters, we can use a uniform prior, which assigns equal probability to all possible values of the parameters within a certain range. And this is usually uninformative and called *flat prior*.

Likelihood is the probability density function describing the data distribution given the parameters. It is also a function of the parameters when observed data is given, and it measures how well the model with parameters $\boldsymbol{\theta}$ explains the data \mathbf{x} . The likelihood function is defined as [78]:

$$P(\mathbf{x}|\boldsymbol{\theta}) \equiv L(\mathbf{x}; \boldsymbol{\theta}) . \quad (7.4)$$

If flat prior is applied, the posterior distribution is proportional to the likelihood function:

$$P(\boldsymbol{\theta}|\mathbf{x}) \propto L(\mathbf{x}; \boldsymbol{\theta}) , \quad (7.5)$$

which implies that we can focus solely on the likelihood function to analyze the posterior distribution, as this does not alter the relative probabilities of the parameters. Therefore, inside the range of prior, the posterior distribution is proportional to the likelihood while outside the range of prior, the posterior distribution is zero.

An very important method to probe likelihood distribution is *Fisher information matrix*, which is a measure of the amount of information that an observed random variable set carries about the parameters $\boldsymbol{\theta}$, written as:

$$F_{ij} \equiv \left\langle \frac{\partial^2 \mathcal{L}}{\partial \theta_i \partial \theta_j} \right\rangle , \quad (7.6)$$

where $\mathcal{L} = -\ln L$, and the indices i and j correspond to the parameters. The ensemble average is computed with respect to the likelihood in the data space. To better understand the physical significance of the Fisher matrix, we first define the maximum likelihood estimator (MLE) $\hat{\boldsymbol{\theta}}$, which represents the parameter vector $\boldsymbol{\theta}_{\text{ML}}$ that maximizes the likelihood

function within the parameter space for a given data set \mathbf{x} . Additionally, we focus on the regions surrounding $\boldsymbol{\theta}_{\text{ML}}$, as these regions coincide with the peak areas of the posterior distribution under the assumption of a flat prior.

To show some good properties of Fisher matrix, we can expand the log-likelihood function around the maximum likelihood parameter $\boldsymbol{\theta}_{\text{ML}}$ using Taylor expansion:

$$\mathcal{L}(\mathbf{x}; \boldsymbol{\theta}) \approx \mathcal{L}(\mathbf{x}; \boldsymbol{\theta}_{\text{ML}}) + \frac{1}{2} \sum_{i,j} (\theta_i - \theta_{i,\text{ML}}) \frac{\partial^2 \mathcal{L}}{\partial \theta_i \partial \theta_j} (\theta_j - \theta_{j,\text{ML}}) |_{\boldsymbol{\theta}_{\text{ML}}} + \mathcal{O}(\boldsymbol{\theta}^3), \quad (7.7)$$

where the first order derivative, $\partial \mathcal{L} / \partial \theta_i$, with respect to each parameter θ_i vanishes at the maximum likelihood point because the likelihood reaches the maximum at $\boldsymbol{\theta}_{\text{ML}}$ the slope goes to zero at that point. Thus, the log-likelihood function is basically characterized by the quadratic term of $\boldsymbol{\theta}$. Changing the log-likelihood back to the likelihood, we have:

$$L(\mathbf{x}; \boldsymbol{\theta}) \approx \exp(-\mathcal{L}) \quad (7.8)$$

$$\approx L(\mathbf{x}; \boldsymbol{\theta}_{\text{ML}}) \exp \left[-\frac{1}{2} \sum_{i,j} (\theta_i - \theta_{i,\text{ML}}) \frac{\partial^2 (-\ln L)}{\partial \theta_i \partial \theta_j} (\theta_j - \theta_{j,\text{ML}}) |_{\boldsymbol{\theta}_{\text{ML}}} + \mathcal{O}(\boldsymbol{\theta}^3) \right], \quad (7.9)$$

which is a Gaussian distribution with respect to the parameters $\boldsymbol{\theta}$ centered at $\boldsymbol{\theta}_{\text{ML}}$ with $L(\mathbf{x}; \boldsymbol{\theta}_{\text{ML}})$ acting as a scaling factor. Furthermore, we could re-formulate this equation using the standard expression of multivariate Gaussian distribution by setting: and the inverse covariance matrix is given by the Fisher matrix:

$$\mathbf{C}_{ij}^{-1} = -\frac{\partial^2 (\ln L)}{\partial \theta_i \partial \theta_j}. \quad (7.10)$$

Therefore, the Fisher matrix can be interpreted as the expectation value of the inverse parameter covariance matrix at $\boldsymbol{\theta}_{\text{ML}}$. Consequently, the inverse of the Fisher matrix serves as an estimate of the parameter covariance matrix. The diagonal elements of this covariance matrix represent the variance of individual parameters, while the off-diagonal elements capture the covariance between different parameters, indicating the degree of correlation among them.

When the parameter uncertainties are small, corresponding to a small covariance matrix and large Fisher matrix values, Eq. (7.9) shows that even a slight deviation of $\boldsymbol{\theta}$ from $\boldsymbol{\theta}_{\text{ML}}$ results in a significant reduction in the likelihood function. This implies that the posterior distribution is well-approximated by a multivariate Gaussian distribution. The inverse parameter covariance matrix reflects the curvature of the likelihood surface at $\boldsymbol{\theta}_{\text{ML}}$. A large Fisher matrix value indicates that the likelihood is highly sensitive to changes in the parameter, signifying that more information is available. Conversely, a small Fisher matrix value suggests that the likelihood remains relatively constant across the parameter space, indicating less information. Furthermore, Eq. (7.9) demonstrates that under the Gaussian approximation, the likelihood surface is effectively characterized by the parameter covariance matrix, making it a key determinant of the uncertainty in

parameter estimation [79].

We assume that the data \mathbf{x} follow a multivariate Gaussian distribution with the data covariance matrix \mathbf{C} and the mean $\boldsymbol{\mu}$ predicted by a model M . This is a reasonable assumption since \mathbf{x} is usually the average over large amount of observational result, each of these independent observations should follow an identical distribution and should follow a Gaussian distribution by central limit theorem. Thus, the likelihood function can be expressed as:

$$L(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{C}|}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]. \quad (7.11)$$

By taking the logarithm of the likelihood function, and ignoring the constant term, we have:

$$\ln L(\mathbf{x}; \boldsymbol{\theta}) = -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}). \quad (7.12)$$

Then, take the derivative of the log-likelihood function with respect to one parameter θ_i :

$$\frac{\partial \ln L}{\partial \theta_i} = \frac{1}{2} \left(\frac{\partial \boldsymbol{\mu}}{\partial \theta_i} \right)^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) + \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \theta_i}. \quad (7.13)$$

By taking another derivative with respect to θ_j , we have:

$$\begin{aligned} \frac{\partial^2 \ln L}{\partial \theta_i \partial \theta_j} &= \frac{1}{2} \left(\frac{\partial^2 \boldsymbol{\mu}}{\partial \theta_i \partial \theta_j} \right)^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &\quad - \frac{1}{2} \left(\frac{\partial \boldsymbol{\mu}}{\partial \theta_i} \right)^T \mathbf{C}^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \theta_j} \\ &\quad - \frac{1}{2} \left(\frac{\partial \boldsymbol{\mu}}{\partial \theta_j} \right)^T \mathbf{C}^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \theta_i} \\ &\quad + \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} \frac{\partial^2 \boldsymbol{\mu}}{\partial \theta_i \partial \theta_j}. \end{aligned} \quad (7.14)$$

After taking the average over the data space, the first and last term vanish because the average of the data \mathbf{x} is equal to the mean $\boldsymbol{\mu}$, and due to the symmetry of the covariance matrix, the final expression for Fisher matrix is [80]:

$$F_{ij} = \left\langle \frac{\partial^2 \ln L}{\partial \theta_i \partial \theta_j} \right\rangle = - \left(\frac{\partial \boldsymbol{\mu}}{\partial \theta_i} \right)^T \mathbf{C}^{-1} \left(\frac{\partial \boldsymbol{\mu}}{\partial \theta_j} \right). \quad (7.15)$$

As we can see in Eq. (7.15), Fisher matrix depends on only the theoretical modeling and its parameters. So no actual data is required to calculate the Fisher matrix and the goal of Fisher matrix is to obtain a forecast of parameter uncertainties before any data is collected. Fisher forecast gives the best possible estimate according to the Cramer Rao bound.

7.2.1 Covariance estimation

In the framework of Fisher analysis, the data set utilized consists of scattering coefficients calculated from the CosmoGridV1 simulations. The parameter vector includes six distinct parameters, $\boldsymbol{\theta} = (\Omega_m, \sigma_8, w_0, n_s, H_0, \Omega_b)$. Predictions $\boldsymbol{\mu}$ for the data vectors are generated through scattering transform modeling. Based on Eq. (7.15), we now need to estimate the covariance matrix of the scattering coefficients.

Given the critical importance of an accurate covariance matrix for parameter estimation, various methods have been developed to compute it, tailored to the characteristics of different statistical techniques. Some approaches rely on analytical methods, such as model-independent modeling [81] or Gaussian and non-Gaussian decomposition [82]. Others estimate the covariance matrix directly from observational data [83]. In this study, we adopt a simulation-based approach to calculate the covariance matrix.

The estimator of covariance matrix is given by:

$$\hat{\mathbf{C}} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T, \quad (7.16)$$

where N is the number of independent realizations. For each realization, it produces a data vector \mathbf{x}_i of scattering coefficients, and the average of the data vector is given by:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (7.17)$$

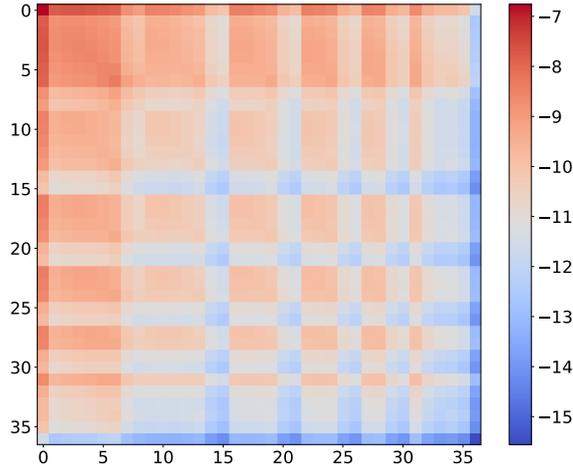


Figure 7.3: Data covariance matrix of scattering coefficient at fiducial cosmology within latest tomographic redshift bin. Here the matrix is shown in logarithmic scale. The underlying cosmological parameters are $\Omega_m = 0.26$, $\sigma_8 = 0.84$, $w_0 = -1$, $n_s = 0.9649$, $H_0 = 67.3$ km/s/Mpc, and $\Omega_b = 0.0493$. This data covariance matrix is estimated from noiseless weak lensing convergence maps.

In this Fisher analysis section, we estimate the data covariance matrix from scattering coefficients calculated from square maps generated from CosmoGridV1 simulation. The square maps here have a size of 86 by 86 pixel², and each pixel has an area of 6.9×6.9 arcmin². So, the area of each square map is $9.85^\circ \times 9.85^\circ$. We can extract 260 non-overlapping square maps from one full-sky map. There are 200 different realizations of full-sky maps with the same cosmological parameters. Thus, we have 52,000 square maps in total for scattering coefficients calculation. The parameters for scattering transform are set to $J = 8$ and $L = 4$, which means that we are extracting information with filters of 8 different scales and 4 different orientations. So the total scattering coefficients have 1 zeroth order coefficient, 8 first order coefficients and 28 second order coefficients. Fig. 7.3 shows the estimated data covariance matrix. Since we have assumed that the data covariance matrix does not depend on parameters, we choose to compute this data covariance matrix at the fiducial cosmology without ant noise or systematic effects included.

In Fig. 7.3, each pixel represents one entry in the data covariance matrix. Index 0 denotes the zeroth order scattering coefficients, indices from 1 to 8 are the first order scattering coefficients, and indices from 9 to 36 are the second order scattering coefficients. So the matrix has a size of 37×37 . Since the scattering coefficients cover different orders of magnitude, we show the covariance matrix in logarithmic scale. The diagonal elements represent the variance of each scattering coefficient, while the off-diagonal elements represent the covariance between different coefficients. To better show the correlation between different scattering coefficients, we calculate the Pearson correlation coefficients matrix which is based on data covariance matrix but normalized by the variance of each coefficient:

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}, \quad (7.18)$$

where X and Y are two random variables from the data vector, $\text{cov}(X, Y)$ represents their covariance, and σ_X and σ_Y denote their respective standard deviations. The correlation coefficient quantifies the degree to which changes in one variable are associated with changes in the other. Fig. 7.4 shows the how strong the correlation is between different scattering coefficients under noiseless and noisy condition. For noiseless data, we could see the strong correlation between different scattering coefficients. Since the second order scattering transform is just another convolution based on the first order scattering transform, it is expected that the second order coefficients are highly correlated with the first order coefficients under the same first order wavelet filters. If we add systematics to the maps, the correlation between different coefficients is still strong for those with big first and second order wavelet filters. This is because big filters cover large areas when doing convolution, and they are more stable to the systematic effects and Gaussian shape noise, while the small filters capture only very localized information, and they are more sensitive to the systematic errors.

After computing the data covariance matrix, the next step is to estimate its inverse, known as the *precision matrix*. However, directly inverting the data covariance matrix introduces bias, as discussed in [84]. The sample variance obtained from the direct inversion of \hat{C} fluctuates around the true variance but is constrained to remain above zero. When \hat{C}

is inverted, some sample variance values that are close to zero can become arbitrarily large. Since there are no corresponding compensating small values, the resulting precision matrix estimator becomes biased upwards [79]. The bias can be corrected by the following equation:

$$\widehat{C}^{-1} = \frac{N - D - 2}{N - 1} \hat{C}^{-1}, \quad (7.19)$$

where \widehat{C}^{-1} is the unbiased estimator in the inverse covariance matrix, N is the number of realizations, D is the dimension of each data vector, and \hat{C} is the sample covariance matrix.

In [80], the authors show that the inversion of Fisher matrix also introduce some bias to the uncertainty of the covariance matrix of parameters. In order to reduce this bias, another correction factor should be applied to the inverse Fisher matrix:

$$\mathbf{C}_{\text{param}} = \left(1 + \frac{(D - N_p)(N - D - 2)}{(N - D - 1)(N - D - 4)} \right) \cdot \mathbf{F}^{-1}, \quad (7.20)$$

where N_p is the number of parameters to be constrained.

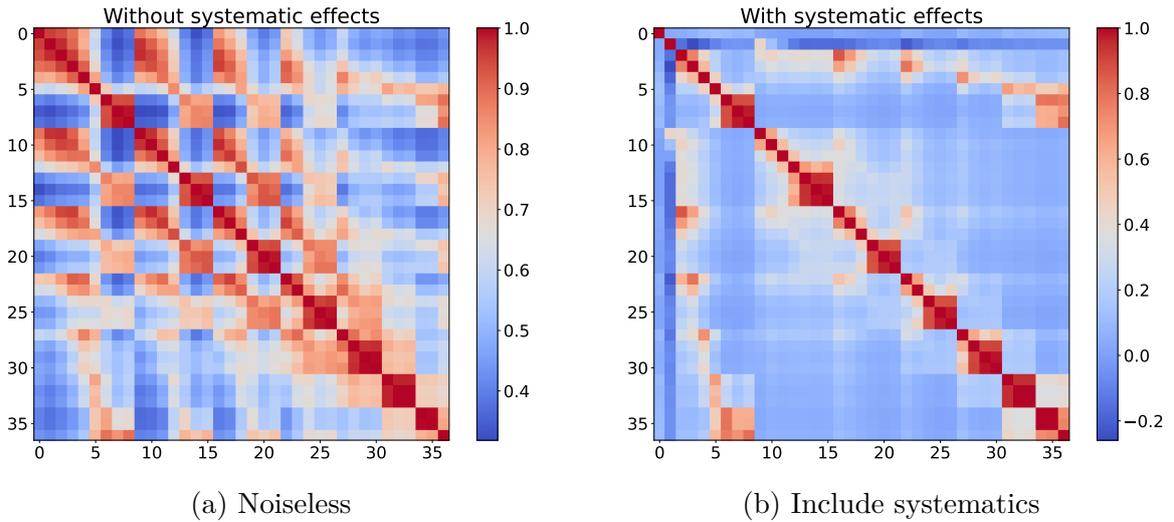


Figure 7.4: Left panel is the correlation matrix calculated from Fig. 7.3. The right panel is the correlation matrix calculated from maps that include intrinsic alignment, multiplicative bias, and shape noise with the same cosmological parameters as the left one. Both of these are calculated from tomographic bin 0.

7.2.2 Fisher forecast

Together with Eq. (7.15), (7.16) and (7.19), we could compute the Fisher matrix. In the case of this project, we calculate the derivatives with respect to the parameter using the following equation:

$$\frac{\partial \boldsymbol{\mu}}{\partial \theta_i} = \frac{\boldsymbol{\mu}(\boldsymbol{\theta}_{\text{fid}} + \boldsymbol{\delta}\theta_i) - \boldsymbol{\mu}(\boldsymbol{\theta}_{\text{fid}} - \boldsymbol{\delta}\theta_i)}{2\boldsymbol{\delta}\theta_i}, \quad (7.21)$$

where θ_{fid} represents the parameters of fiducial cosmology, θ_i is the change of i -th parameter with respect to the fiducial cosmology while keeping other parameters fixed at fiducial values, and μ denotes the scattering coefficients at the corresponding cosmology. The value of $\delta\theta$ is listed in Table 6.1.

Fig. 7.5 illustrates the 1σ Fisher forecast for the parameters Ω_m and σ_8 . The contours in the two-dimensional parameter space are elliptical, consistent with the assumption of a Gaussian likelihood function in the parameter space. The gray contour represents the forecast using the power spectrum, while the green contour corresponds to the wavelet power spectrum defined in Section 4.2.5. Their similar orientation and size indicate that the wavelet power spectrum is consistent with the traditional power spectrum. The blue contour, derived from the first-order scattering coefficients, exhibits a similar orientation to the gray contour, suggesting that the first-order coefficients capture comparable information to the power spectrum. The orange contour, based on de-correlated second-order scattering coefficients, shows a distinct orientation due to the de-correlation operation defined in Eq. (4.9). This implies that the second-order coefficients provide additional information beyond the first-order coefficients. This could also be seen in the marginalized distribution. First order coefficients capture more information about Ω_m while the de-correlated second order coefficients extract the slightly more information about σ_8 . Finally, the red contour, which incorporates all scattering coefficients, yields the tightest constraint, demonstrating the combined information extracted by all three orders of scattering coefficients. Each square map has an area of $9.85^\circ \times 9.85^\circ$, 86×86 pixels². The parameters of scattering transform operation is chosen as $J = 8, L = 4$.

However, sometimes we are not interested in all the parameters that vary, and we only want to focus on a few of them. In this case, we can marginalize the Fisher matrix over the parameters we are not interested in. Suppose the full parameter space vector set is \mathbf{p} , which is a union of two parameter sets: $\mathbf{p} = \mathbf{q} \cup \mathbf{r}$, where \mathbf{q} is the parameter set we are interested in and \mathbf{r} is the parameter set we want to marginalize. The full Fisher matrix could be written as:

$$F = \begin{bmatrix} F_{\mathbf{q}\mathbf{q}} & F_{\mathbf{q}\mathbf{r}} \\ F_{\mathbf{r}\mathbf{q}} & F_{\mathbf{r}\mathbf{r}} \end{bmatrix}. \quad (7.22)$$

Then the marginalized Fisher matrix is given by:

$$F_{\mathbf{q}\mathbf{q}}^{\text{marg}} = F_{\mathbf{q}\mathbf{q}} - F_{\mathbf{q}\mathbf{r}} F_{\mathbf{r}\mathbf{r}}^{-1} F_{\mathbf{r}\mathbf{q}}. \quad (7.23)$$

7.2.3 Impact of resolution and shear

In the above discussion, we used scattering transform to extract information on noiseless weak lensing convergence field at the resolution level of the simulation data. However, in real observations, we could usually only obtain shear fields. Thus, in this subsection, we will discuss the impact of resolution and shear field on Fisher forecast using scattering transform.

Based on the noiseless simulation convergence full sky maps, we could use Eq. (3.42) to generate full sky shear maps. Of course, the generated shear maps have the same cosmo-

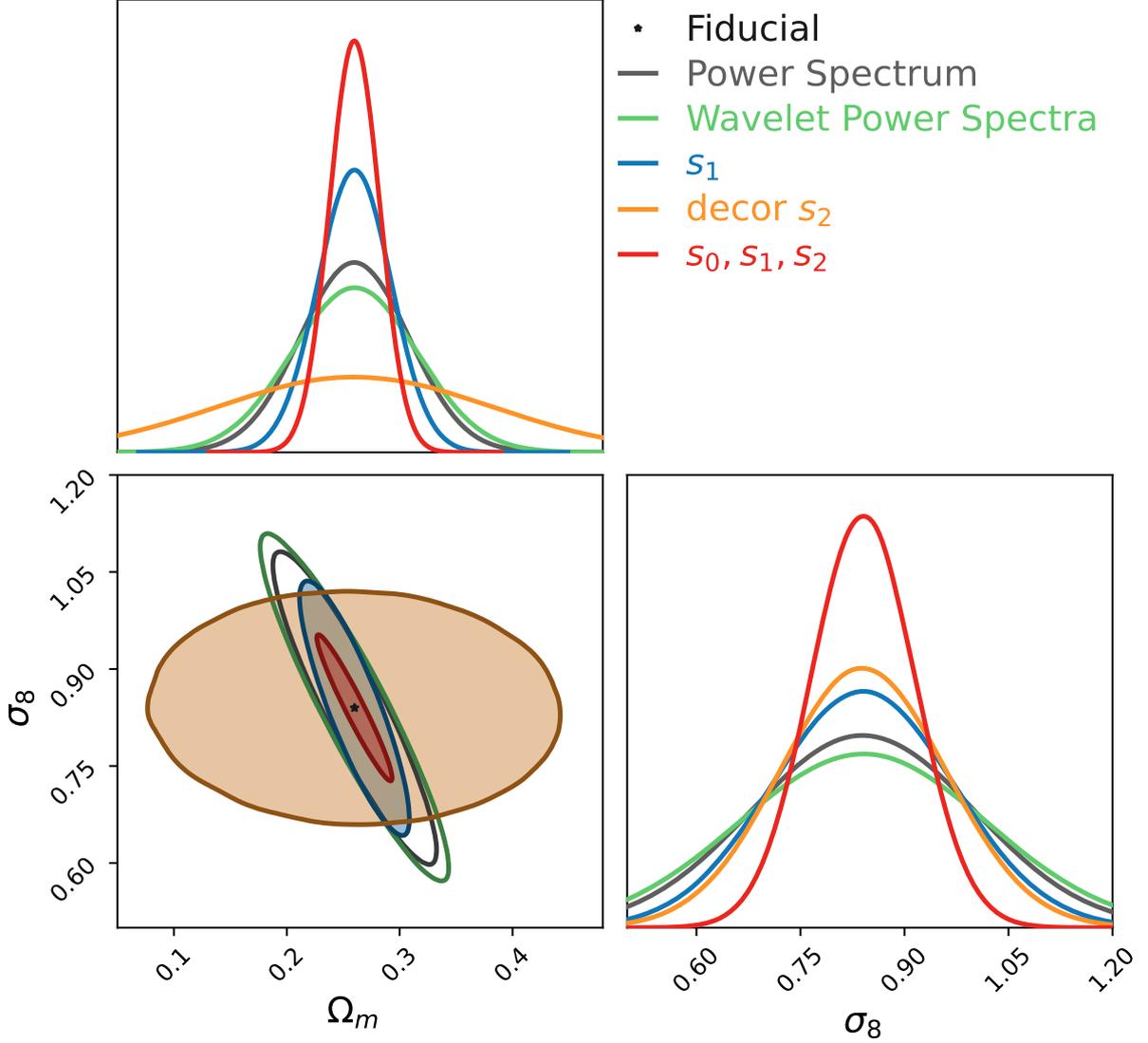


Figure 7.5: 1σ contour of Fisher forecast of Ω_m and σ_8 . Black star is the fiducial values of these two parameters. Grey contour is the forecast using power spectrum mentioned in Appendix B. Green elliptical is the binned power spectrum using Eq. (4.22). Blue contour is the forecast using only first order scattering coefficients. Orange Contour shows the result using de-correlated second order scattering coefficients defined in Eq. (4.9). Red contour demonstrates the forecast using all three orders of scattering coefficients. All these data are calculated from noiseless weak lensing convergence maps in the latest redshift bin.

logical parameters as the corresponding convergence maps. Then we use the same method to cut square maps from two shear full-sky maps and compute scattering coefficients on both shear maps to get the scattering coefficients for shear:

$$s_{\gamma,\text{total}} = \text{concatenate}(s_{\gamma,1}, s_{\gamma,2}) , \quad (7.24)$$

where $s_{\gamma,\text{total}}$ represents the total shear scattering coefficients that we will use to generate Fisher forecast, and $s_{\gamma,1}$ and $s_{\gamma,2}$ are the scattering coefficients calculated from two shear maps respectively. Besides, the scattering coefficients on the are also computed on the magnitude shear maps:

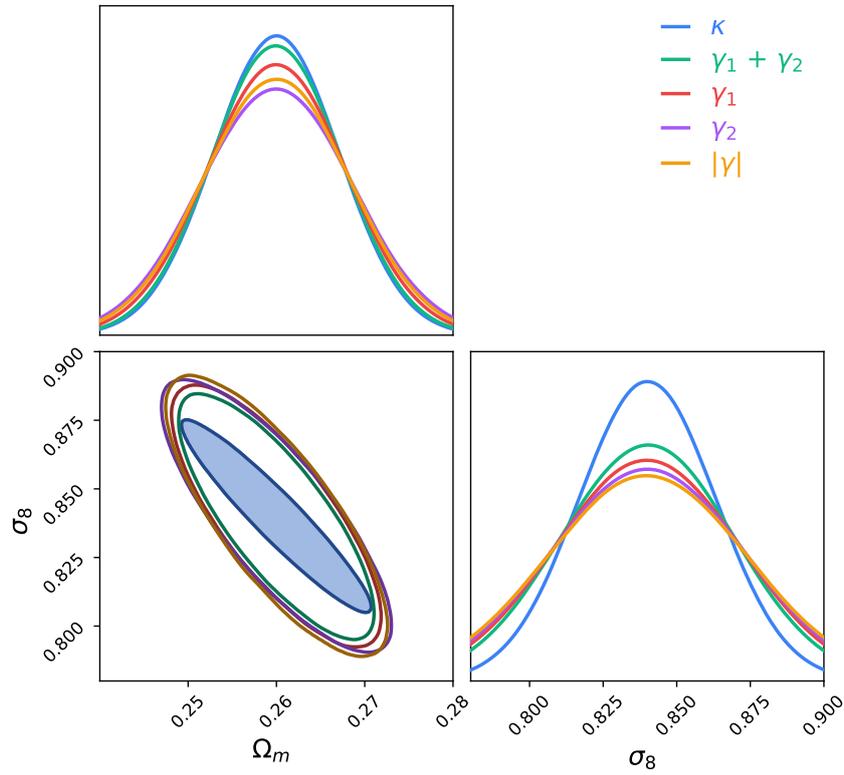
$$\boldsymbol{\gamma} = |\gamma| e^{2i\phi} . \quad (7.25)$$

Fig. 7.6a illustrates the 1σ Fisher forecast derived from scattering coefficients of various maps. All square maps used in this analysis have dimensions of $40^\circ \times 40^\circ$, with a resolution of 300 pixels by 300 pixels, where each pixel corresponds to 8×8 arcmin². The results indicate that the combined shear fields yield a tighter contour compared to a single shear field, as the information from the original convergence field is distributed across the two shear fields. However, when comparing the contours of the convergence field and the shear field, it is evident that the convergence field provides a more constrained forecast in the Ω_m and σ_8 parameter space. This suggests that the scattering transform method is more effective on convergence maps, even though the shear maps are directly derived from the convergence maps.

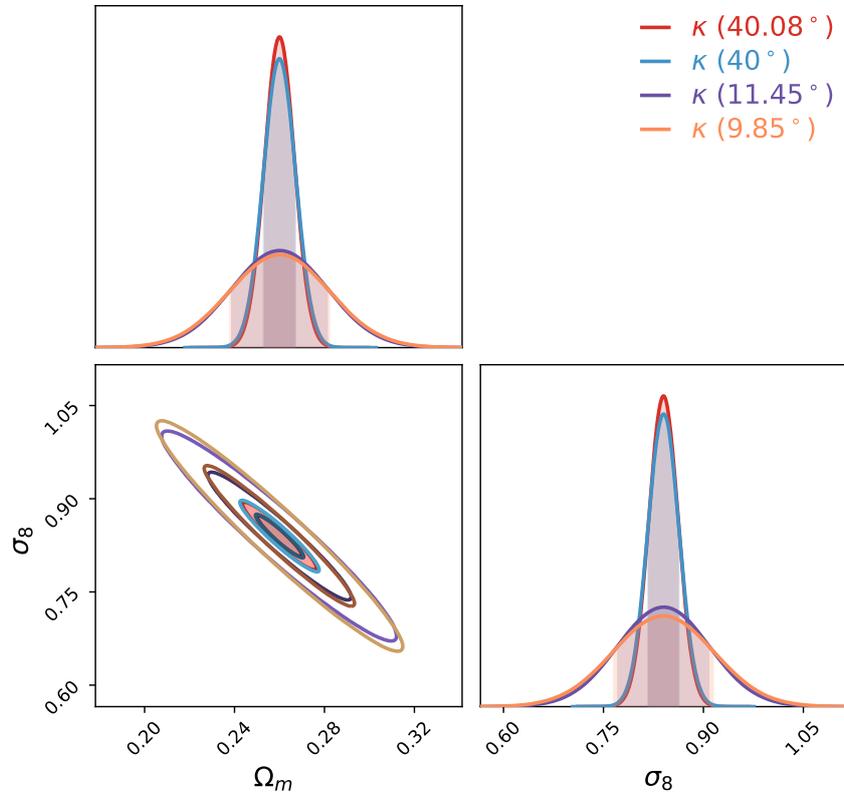
We further investigate the impact of pixel resolution and map area size on the scattering transform, as illustrated in Fig. 7.6b. To ensure a fair comparison, we keep the total area of the square maps constant while varying the pixel resolution, resulting in different pixel counts per map. The red contour corresponds to square maps of $40.08^\circ \times 40.08^\circ$ with a resolution of 6.9×6.9 arcmin², yielding 350 pixels per side. The blue contour represents square maps of $40^\circ \times 40^\circ$ with a resolution of 8×8 arcmin², resulting in 300 pixels per side. This comparison demonstrates that, for maps covering the same area, increasing the resolution or pixel count improves the forecast accuracy.

Additionally, we examine the effect of total map area on the scattering transform. The red, purple, and orange contours correspond to maps with a resolution of $N_{\text{SIDE}} = 512$ but varying total areas. These maps have pixel counts of 350, 100, and 86, corresponding to areas of $40.08^\circ \times 40.08^\circ$, $11.45^\circ \times 11.45^\circ$, and $9.85^\circ \times 9.85^\circ$, respectively. The results clearly show that larger map areas yield tighter constraints.

However, when applying the scattering transform, the maps must be square, which necessitates the validity of the flat-sky approximation to ignore curvature effects. According to the ‘‘Flat-sky approximation’’ section in [85], the largest area size suitable for this approximation is approximately $10^\circ \times 10^\circ$. Therefore, performing the scattering transform on convergence maps of size $10^\circ \times 10^\circ$ is the most reliable approach for extracting information in weak lensing cosmology.



(a) convergence and shear comparison



(b) different size comparison

Figure 7.6: Illustration of Fisher forecast on σ_8 and Ω_m plane using convergence field, shear field, and different pixel numbers and resolutions.

7.3 Emulator

Since now there is no analytical model for scattering transform and we can not calculate scattering coefficients use some equations directly but use machine learning. Here we will use a neural network-based emulator to predict cosmological parameters with given cosmological parameters so that we have the model predicted value $\mu(\theta)$ for the likelihood.

For the emulators in this project, we are using fully-connected neural network, with several hidden layers. There are 2500 grid simulations with different cosmological parameters in 4 tomographic redshift bins, we thus use scattering coefficients calculated from these simulations for machine learning work. For each tomographic redshift bin, we will set one emulator for scattering coefficients prediction. 90% are set to be training sets and 10% are set to be validation set for each tomographic bin. Since scattering coefficients are very small and cover many different orders of magnitudes, seen in Fig. 7.2, we first take the logarithm of the coefficients with base of 10 to make them have the same order before putting them into trainings. To simplify the training process and make the emulators more accurate, we do not use zeroth-order scattering coefficients, the mean value of the square maps, as the training features. Before the training starts, we further standardize the training features so that they have 0 mean and standard deviation of 1.

$$s_{\text{stand}} = \frac{s - \bar{s}}{\sigma_s}, \quad (7.26)$$

where s represents the scattering coefficients here. For the activation function, we choose Parametric Swish functions:

$$h(x) = (\beta + \sigma(\alpha x)(1 - \beta))x, \quad (7.27)$$

with

$$\sigma(\alpha x) = \frac{1}{1 + e^{-\alpha x}}, \quad (7.28)$$

where x is the output from previous layer and β and α both trainable parameters, just like weight W and bias b in each layer. During the training process, we utilize the mini-batch gradient descent method, choosing a small batch from the whole training set randomly to calculate the gradient at each epoch. From the training set, we split 10% as the validation set to calculate the loss function. To prevent overfitting, we set early stopping strategy and save the model parameter whenever it gets a new minimum loss value from validation sets.

Fig. 7.7 shows the prediction of the scattering transform of one cosmology. The emulator achieves high accuracy in reproducing the scattering coefficients, with most relative errors remaining at the level of 10^{-3} and only a few features reaching up to 10^{-2} . And this is also understandable, since those features have very low magnitude, meaning that a little deviation from the true values (here we assume the coefficients from simulations are the true values) leads to a lot of difference. Across a dynamic range spanning five orders of magnitude, the emulator closely matches the real data, demonstrating its robustness and reliability for subsequent cosmological parameter inference.

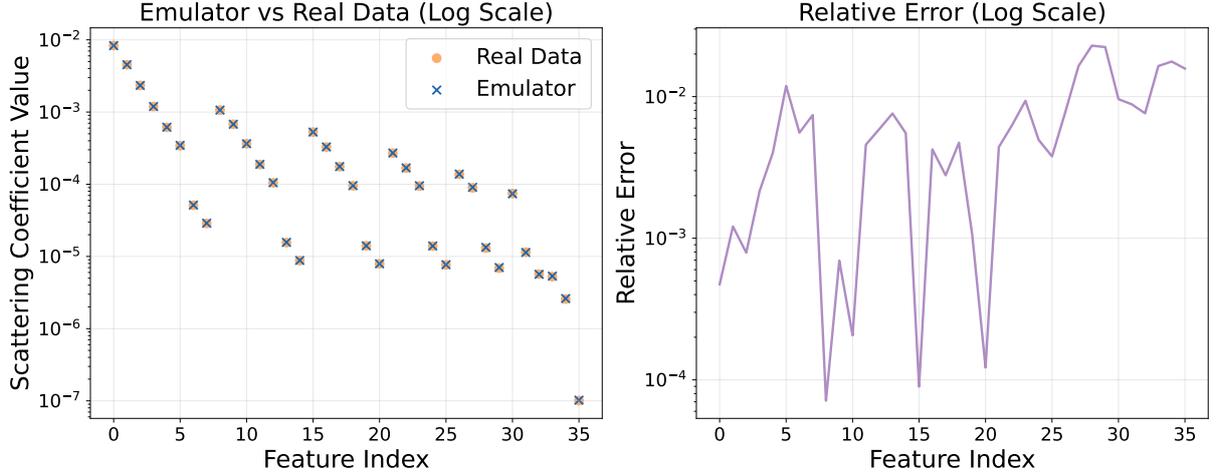


Figure 7.7: Demonstration of the emulation prediction on noisy maps. The left panel shows the scattering coefficients calculated from simulation and that predicted from emulator. The right panel exhibits the relative error of prediction compared to the real value.

To further quantify the error, we calculated the reduced χ^2 [86, 87] for testing sets:

$$\chi_{\text{red}}^2 = \frac{(\mathbf{s}_{\text{pred}} - \mathbf{s}_{\text{sim}}) \cdot \mathbf{C}^{-1} (\mathbf{s}_{\text{pred}} - \mathbf{s}_{\text{sim}})^T}{N_{\text{dof}}}, \quad (7.29)$$

where \mathbf{s}_{pred} is the prediction from emulator, \mathbf{s}_{sim} is the value calculated from simulation, \mathbf{C} is the covariance matrix estimated from fiducial cosmology with all systematic effects included and N_{dof} is the degrees of freedom, which is 36 in this case. The reduced chi-squared measures how well the predicted values match the true values, taking into account the uncertainties via the covariance matrix. It tells us, on average per degree of freedom, how large the normalized residuals are. A value of $\chi_{\text{red}}^2 \sim 1$ indicates that the emulator predictions are consistent with the simulation within the expected uncertainties. Values significantly greater than 1 suggests that the emulator systematically deviates from the true values, whereas values much smaller than 1 may indicate that the uncertainties are overestimated.

Fig. 7.8 shows the values of reduced χ_{red}^2 trained on four tomographic bins with scattering coefficients calculated from noisy maps. All of these four emulators have the mean value of χ_{red}^2 very close to 1 tested on testing sets. Because different random seeds are used when splitting the whole data into training set and testing set, the distributions of the dots are different in Fig. 7.8.

7.4 MCMC sampling

The Fisher forecast assumes a Gaussian likelihood distribution in the parameter space, allowing for the prediction of parameter uncertainties without determining the maximum likelihood location. Typically, the center of the Fisher forecast is chosen to coincide with

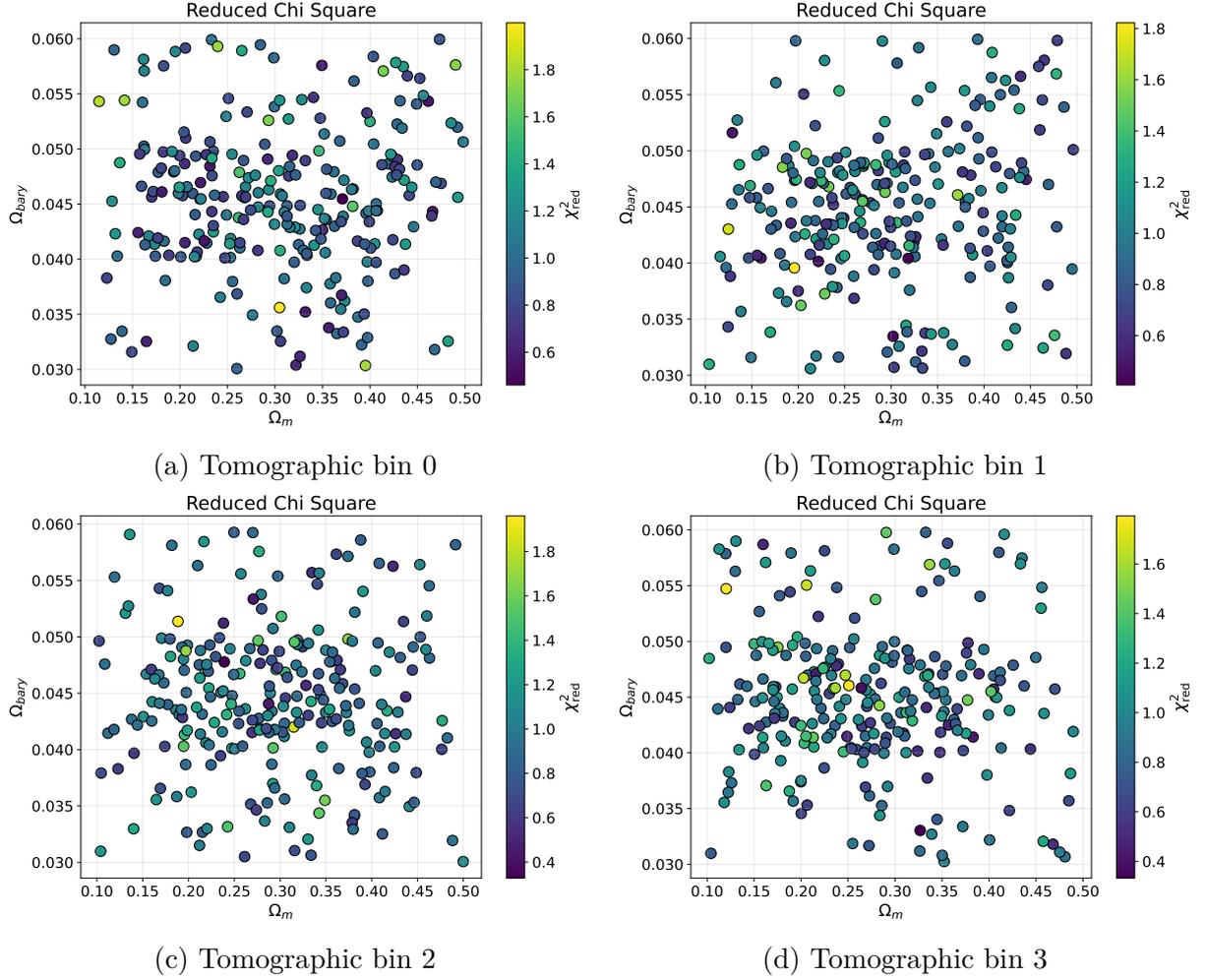


Figure 7.8: Reduced chi-squared values of emulators for four tomographic redshift bins on testing sets projected on $\Omega_{\text{bary}} - \Omega_m$ plane. The color shows the values of χ_{red}^2 defined in Eq. (7.29).

the fiducial parameter values, as illustrated in Fig. 7.5.

Once observational data is available, numerical methods can be employed to explore the posterior distribution of cosmological parameters, $p(\boldsymbol{\theta}|\mathbf{x})$. This provides a comprehensive view of the parameter distribution, enabling the identification of the maximum posterior parameters and their corresponding uncertainties. In this section, we introduce a widely used method for sampling the posterior distribution, known as *Markov Chain Monte Carlo* (MCMC). We will first explain the concept and advantages of MCMC, followed by its application in conjunction with a neural network-based emulator.

As a probability distribution, the posterior satisfies the following properties [88]:

$$p(\boldsymbol{\theta}|\mathbf{x}) > 0 \quad \text{for all } \boldsymbol{\theta} , \quad (7.30)$$

$$\int p(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta} = 1 , \quad (7.31)$$

where the first property ensures that the posterior probability is always non-negative across the parameter space, and the second property guarantees that the total probability over all possible parameter values sums to 1. However, due to the high dimensionality of the parameter space in models with multiple free parameters, calculating posterior statistics, such as the expectation value, can be computationally demanding:

$$\mathbb{E}[\boldsymbol{\theta}] = \int \boldsymbol{\theta} \cdot p(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta} , \quad (7.32)$$

where the integration spans the entire parameter space.

The MCMC method provides a solution to this challenge by generating a sequence of samples from the posterior distribution. The key idea is to use these samplings to approximate the posterior distribution, allowing us to compute posterior statistics without the need for direct integration.

$$\mathbb{E}[\boldsymbol{\theta}] \approx \frac{1}{N} \sum_{i=1}^N \boldsymbol{\theta}_i , \quad (7.33)$$

where N is the number of samples drawn from the posterior distribution, and $\boldsymbol{\theta}_i$ represents the i -th sample from N .

A Markov chain is a stochastic process that describes a sequence of random variable evolving over discrete steps, where the distribution of the next state depends only on the current state and not on the full history. Mathematically, this property, known as *Markov property*, can be expressed as:

$$p(\boldsymbol{\theta}_{n+1}|\boldsymbol{\theta}_n, \boldsymbol{\theta}_{n-1}, \dots, \boldsymbol{\theta}_1) = p(\boldsymbol{\theta}_{n+1}|\boldsymbol{\theta}_n) , \quad (7.34)$$

where $\boldsymbol{\theta}_n$ denotes the state of the system at sampling step n . Here p is the transition probability that governs the evolution between consecutive states. It defines the probability of moving from the current parameter vector $\boldsymbol{\theta}_n$ to a new vector $\boldsymbol{\theta}_{n+1}$. The Markov

property implies that, after a sufficiently large number of steps, the influence of the initial state $\boldsymbol{\theta}_1$ becomes negligible, and the chain's behavior depends only on the current state. This feature is crucial because it ensures that once the Markov Chain reaches equilibrium, the samples generated are effectively independent of the starting point and can be used to explore the target probability distribution efficiently.

In practice, sampling from a complex distribution is achieved by evolving a Markov Chain according to a carefully designed transition probability. The aim is to ensure that, regardless of the initial distribution of the random variable $\boldsymbol{\theta}_1$, the distribution of later samples $\boldsymbol{\theta}_n$ converges to the desired target distribution. This convergence relies on certain properties of the transition probability, among which detailed balance plays a fundamental role. The detailed balance condition is defined as:

$$p(\boldsymbol{\theta}_n | \boldsymbol{\theta}_{n+1} \pi(\boldsymbol{\theta}_{n+1})) = p(\boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_n \pi(\boldsymbol{\theta}_n)) , \quad (7.35)$$

where $\pi(\boldsymbol{\theta})$ is the target distribution we wish to sample from. When the transition kernel satisfies detailed balance and other conditions such as ergodicity, the stationary distribution of the Markov Chain becomes exactly the target distribution $\pi(\boldsymbol{\theta})$. In other words, if the chain has already reached the target distribution at step n , it will remain in equilibrium under the transition dynamics.

In the context of Bayesian inference, the posterior distribution $p(\boldsymbol{\theta} | \mathbf{x})$ is of primary interest, where \mathbf{x} denotes the observed data. According to Bayes's theorem:

$$p(\boldsymbol{\theta} | \mathbf{x}) = \frac{p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{x})} , \quad (7.36)$$

where $p(\mathbf{x} | \boldsymbol{\theta})$ is the likelihood, $p(\boldsymbol{\theta})$ is the prior and $p(\mathbf{x})$ is the evidence. MCMC methods utilize Markov Chain to generate samples distributed according to $p(\boldsymbol{\theta} | \mathbf{x})$. By running the chain for a sufficient number of steps and ensuring appropriate transition dynamics, the sequence of samples produced can be used to estimate expectations, credible intervals, and other posterior properties of parameters under study. Thus, the combination of the Markov property, careful design of transition probabilities, and the Monte Carlo approach enable MCMC to serve as a powerful tool for posterior inference in complex, high-dimensional parameter spaces.

7.4.1 Prior

As mentioned before, many MCMC methods are proposed for sampling. The simplest one is Metropolis-Hastings algorithm [89]. Hamiltonian Monte Carlo (HMC) [90] is also an efficient MCMC method using Hamiltonian dynamics to do the sampling and leapfrog for integration. And in this work, we will use *affine invariant sampling* MCMC method based on a python package `emcee` [91] but a torch and GPU accelerated version.

For the posterior distribution, since we focus only on the shape of this distribution and the evidence term is only used for normalization and scaling the posterior, we will ignore this term and work on only prior and likelihood. Usually we work with the logarithm of

probabilities to improve numerical stability and to simplify the calculation of acceptance ratios, as difference of log-probabilities are easier and more accurate to compute than ratios of very small numbers:

$$\ln p(\boldsymbol{\theta}|\mathbf{x}) = \ln p(\boldsymbol{\theta}) + \ln L(\mathbf{x}|\boldsymbol{\theta}) . \quad (7.37)$$

For the prior, we adopt the flat distribution for the parameters varying in CosmoGridV1 simulations. The detailed distributions are listed in Table 7.1. For $n_s, \Omega_b, H_0, A_{IA}$, we have a square flat distribution as also shown in Fig. 6.1, while for Ω_m, σ_8, w_0 , we have some cuts from the square flat distribution to match the sampling space of CosmoGridV1 grid simulations as mentioned in Chapter 6.

Parameter	Fiducial Value	Prior Distribution	Additional Condition
Ω_m	0.26	$\mathcal{U}(0.10, 0.50)$	See note (a)
σ_8	0.84	$\mathcal{U}(0.40, 1.40)$	See note (a)
w_0	-1.00	$\mathcal{U}(-2.00, -0.33)$	See note (b)
n_s	0.9649	$\mathcal{U}(0.87, 1.07)$	-
Ω_b	0.0493	$\mathcal{U}(0.03, 0.06)$	-
H_0	67.30	$\mathcal{U}(64, 82)$	-
A_{IA}	0	$\mathcal{U}(-5, 5)$	-
Δz^1	0.0	$\mathcal{N}(0.0, 0.018)$	-
Δz^2	0.0	$\mathcal{N}(0.0, 0.015)$	-
Δz^3	0.0	$\mathcal{N}(0.0, 0.011)$	-
Δz^4	0.0	$\mathcal{N}(0.0, 0.017)$	-
m_1	-0.006	$\mathcal{N}(-0.006, 0.009)$	-
m_2	-0.020	$\mathcal{N}(-0.020, 0.008)$	-
m_3	-0.024	$\mathcal{N}(-0.024, 0.008)$	-
m_4	-0.037	$\mathcal{N}(-0.037, 0.008)$	-

Table 7.1: Priors and additional constraints for cosmological and nuisance parameters. $\mathcal{U}(a, b)$ denotes a uniform prior from a to b , and $\mathcal{N}(\mu, \sigma)$ denotes a Gaussian prior with mean μ and standard deviation σ . (a) (Ω_m, σ_8) are constrained to lie within a polygon \mathcal{P} defined by the vertices: $(0.10, 1.40)$, $(0.20, 1.40)$, $(0.50, 0.65)$, $(0.50, 0.40)$, $(0.30, 0.40)$, $(0.10, 0.90)$. (b) $w_0 \geq w_{0,\min}(\Omega_m)$ where $w_{0,\min}(\Omega_m)$ satisfies $\left(\frac{\Omega_m-1}{\Omega_m}(1+w_0)\right)^{1/(3w_0)} = 1$.

7.4.2 Likelihood

When the data involve the sum or average of many independent random variables, the Central Limit Theorem says the result will tend to be approximately Gaussian, regardless of the underlying distributions. Therefore, the mathematical expression of the likelihood could be written as:

$$\ln L(\mathbf{x}|\boldsymbol{\theta}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})) \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})) , \quad (7.38)$$

where \mathbf{x} denotes the observed data, which, in our case, corresponds to the scattering coefficients measured from the fiducial cosmology. The quantity $\boldsymbol{\mu}$ represents the theoretical prediction that depends on the cosmological parameters to be constrained, and it is obtained by emulator predictions. The matrix \mathbf{C} denotes the data covariance matrix, which is estimated from simulations of the fiducial cosmology and is assumed to be independent of cosmological parameters. Of course, we need to multiply the correction factor shown in Eq. (7.19) [84] when calculating the inverse covariance matrix.

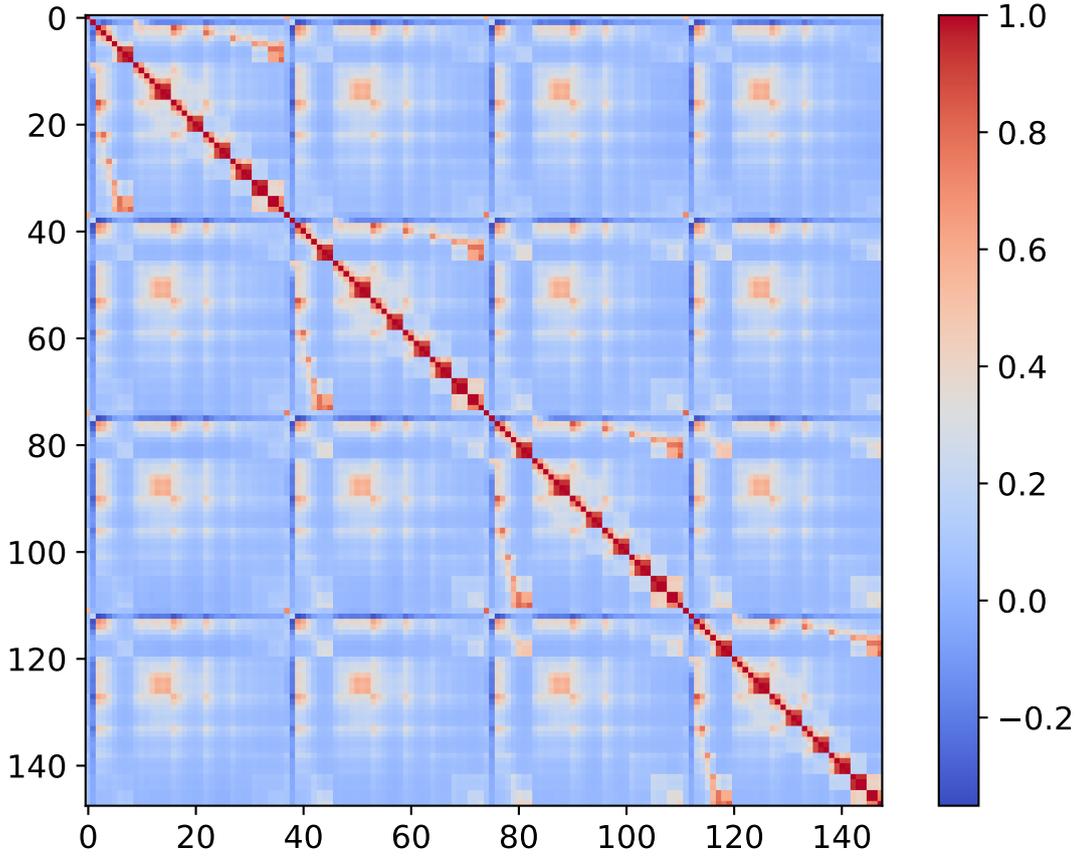


Figure 7.9: Correlation matrix of the data vector $\mathbf{s} = (\mathbf{s}_{\text{tomo0}}, \mathbf{s}_{\text{tomo1}}, \mathbf{s}_{\text{tomo2}}, \mathbf{s}_{\text{tomo3}})$, constructed from 200 realizations of the fiducial cosmology. Each block of 36 features corresponds to the first- and second-order scattering coefficients computed from maps in a specific tomographic redshift bin. The indices are ordered by increasing redshift.

In a tomographic analysis combining four redshift bins, we obtain four sets of scattering coefficients, leading to a data covariance matrix of larger dimensionality. Denoting $\mathbf{s}_{\text{tomon}}$ as the set of first- and second-order scattering coefficients corresponding to the n -th tomographic bin, we construct the full data vector by concatenating the four sets sequentially as $\mathbf{s} = (\mathbf{s}_{\text{tomo0}}, \mathbf{s}_{\text{tomo1}}, \mathbf{s}_{\text{tomo2}}, \mathbf{s}_{\text{tomo3}})$. Consequently, the total length of the data vector becomes 144, and the size of the associated data covariance matrix increases to 144×144 .

As expected, combining multiple redshift bins enhances the amount of information encoded in the resulting data covariance matrix, thereby enabling tighter parameter constraints compared to using the covariance from a single redshift bin alone. The correlation matrix corresponding to the data vector from the tomographic analysis is shown in Fig. 7.9. In this matrix, the four diagonal 36×36 submatrices correspond to the standard correlation matrices for $\mathbf{s}_{\text{tomo}0}$, $\mathbf{s}_{\text{tomo}1}$, $\mathbf{s}_{\text{tomo}2}$, $\mathbf{s}_{\text{tomo}3}$ respectively. The off-diagonal blocks capture the correlations between features originating from different tomographic bins. And this is the covariance matrix for the likelihood.

7.4.3 Posterior

For the logarithm posterior distribution, it gives $-\infty$ if the sampled parameter vector is outside the bounds of the prior and return the sum according to Eq. (7.38) if the parameters are within the boundaries.

A critical aspects of MCMC analysis is verifying chain convergence to ensure reliable posterior distributions. However, assessing convergence is not straightforward, especially when the posterior contains multiple isolated high-probability regions connected by low-probability bridges. In such cases, a sampler may require an impractically long time to explore all significant modes of the distribution.

To address this, we use the *integrated autocorrelation time* τ_{int} , which estimates the number of steps needed for the Markov chain to produce an independent sample. When τ_{int} is sufficiently small relative to the chain length, the sampled distribution reliably approximates the posterior. Following the `emcee`, we implement the following convergence checks:

1. Evaluate the integrated autocorrelation time τ_{int} every 50 steps.
2. Denote the total length of the Markov Chain as τ_{chain} , and use a superscript i on τ_{int} to indicate each computed value.

The Markov Chain is considered to have converged when the following two conditions are met:

$$50 \times \tau_{\text{int}}^i < \tau_{\text{chain}} , \quad (7.39)$$

$$\frac{\tau_{\text{int}}^{i-1} - \tau_{\text{int}}^i}{\tau_{\text{int}}^i} < 0.01 . \quad (7.40)$$

The first condition ensures that the chain is sufficiently long, allowing the MCMC walkers to traverse the high-probability regions multiple times. The second condition requires that the fluctuation in the estimated τ_{int} between successive measurements is small, indicating that the chain has stabilized and convergence has been achieved.

We run 1000 independent Markov Chains whose initial positions are uniformly distributed within the prior bound mentioned in Table 7.1 and end the sampling only when all the

Trace plots for different parameters

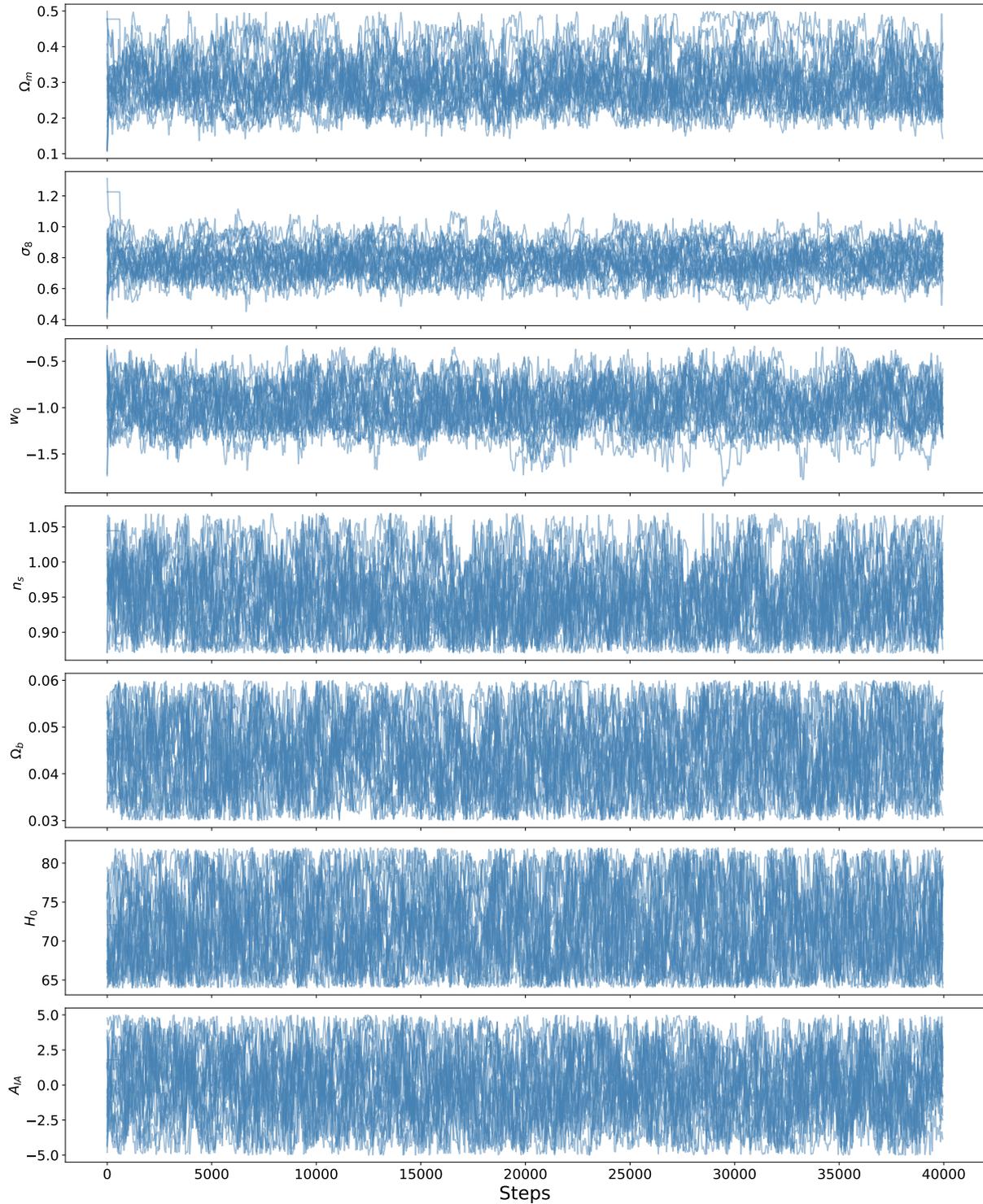


Figure 7.10: Trace plots of the MCMC chains for all four tomographic bins without multiplicative bias and photometric uncertainty parameters. For each parameter, 20 independent Markov Chains are displayed simultaneously. The evolution of all chains is terminated once the convergence criteria given in Eq. (7.39) and (7.40) are satisfied.

chains converge.

Fig. 7.10 shows the trace plots of 20 chains for 7 cosmological parameters using all four tomographic bins. In this plot, the values of parameters are plotted as a function MCMC step numbers. From this figure, we can see that for most parameters, the chains begin to converge after the first few thousand steps. And then the chains began to oscillate in those regions. However, for some parameters such as σ_8 , the convergence is clearer and more stable. Some chains appear stuck in the beginning before transitioning to the high-probability region. This behavior is likely caused by unfavorable initializations in low-likelihood regions, resulting in poor acceptance and preventing them from evolving.

This also reflects the limited constraining power on some parameters, especially when systematic effects are included and all parameters are sampled simultaneously. Overall, the trace plot shows good convergence behavior across bins.

The MCMC corner plots of all four tomographic bins and only tomographic bin 0 are shown in Fig. 7.11. For the prediction of the scattering coefficients from the tomographic bin 0, we used the same emulator in these two analysis. We chose 30000 steps as burn-in phase for the 40000 total steps during which the MCMC chains do not depend on their initial conditions gradually. The burn-in phase would be discarded before shown in the corner plot.

From this figure, we can see that the constraints using four tomographic bins are much tighter than using only one tomographic bin, indicating that more information is extracted by using tomographic analysis. Also, distributions for Ω_m and σ_8 are much tighter compared to other parameters and the fiducial values are lying within the 1σ credible regions of the posterior distribution. This behavior is expected, as weak lensing primarily probes the large-scale matter distribution and the growth of cosmic structures, which are most directly governed by the matter distribution parameter Ω_m and the fluctuation parameter σ_8 . In contrast, parameters such as w_0 , n_s , H_0 and Ω_b mainly influence the cosmic expansion history or the detailed scale dependence of structure formation, leading to much weaker direct effects on the lensing observables.

Moreover, the intrinsic alignment amplitude A_{IA} affects the observed shear signal through galaxy alignment effect rather than through gravitational lensing itself, which further reduces the sensitivity of weak lensing measurement to this parameter. That's why some of these parameters have very flat posterior distributions. As for the nuisance systematic effects, scattering transform does not have any constraining power on them. Besides, we could also see that there are two cuts in $\Omega_m - w_0$ and $w_0 - \sigma_8$ plane, and this is caused by the prior. Since we set the prior values outside the boundary to be zero and no walkers could explore regions there.

In the following discussions, particularly regarding observational effects such as survey masking, we will therefore focus on the parameters to which the scattering transform is most sensitive: Ω_m and σ_8 .

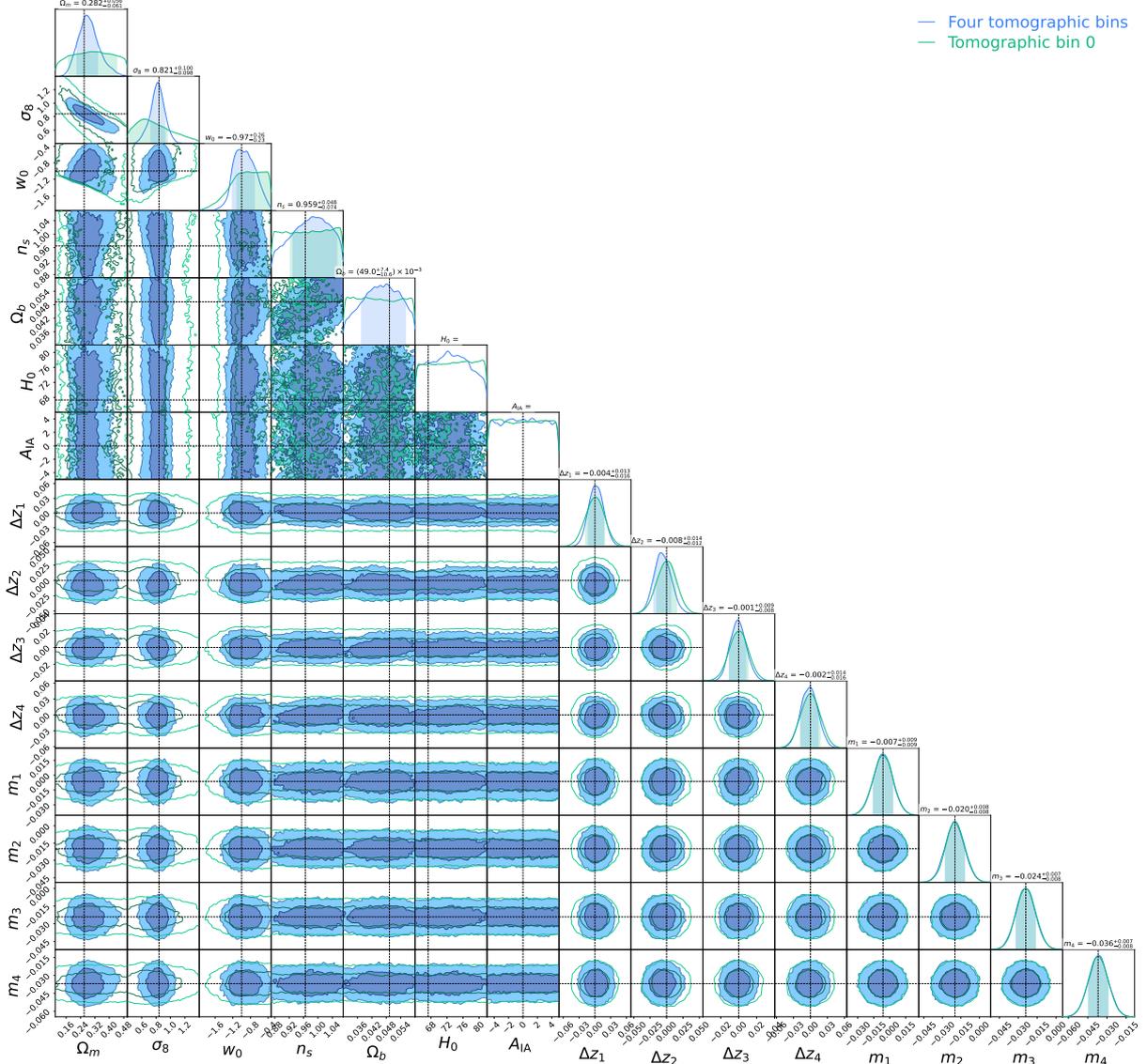


Figure 7.11: Comparison of the MCMC constraints between multi tomographic bins and single redshift bin. The blue contour is the result with all four tomographic bins while the green contour shows the constraints using only earliest redshift bin. All emulators used in MCMC are trained on maps with systematic effects. The parameter values and uncertainties displayed above each marginalized posterior correspond to the blue contours and histograms, which represent the results from the “Four tomographic bins” analysis.

7.5 Handling masked data in scattering transform framework

In the previous analysis, cosmological parameters constraints were obtained based on simulation data incorporating systematic effects such as shape noise and intrinsic alignment. However, in practical observations, an additional complication arises due to the presence of masks. Masks are applied to the data to exclude contaminated or unusable regions, such as area affected by bright stars, foreground objects, or instrumental defects. These masks introduce spatial incompleteness, which can significantly impact the measured weak lensing signal and, consequently, the derived cosmological constraints. To ensure the robustness and applicability of our analysis to realistic survey conditions, it is crucial to account for the effects of masking. In the following, we describe the methodology for incorporating masked regions into the simulation data and analyze their influence on the scattering transform and the resulting parameter constraints.

To better emulate the conditions of real surveys, we adopt a mask consistent with the footprint of the Dark Energy Survey Year 3 (DES Y3) survey covers approximately 5000 square degrees of the southern sky with a complex and irregular geometry [50]. Applying a realistic survey footprint allows us to mimic the spatial inhomogeneity encountered in observational data. In Fig. 7.12, we show the footprint of DES Y3, where the masked regions are clearly visible as gaps within the survey area.

Since scattering transform has to be operated on square maps, we need to extract square maps from DES-like foot print as well. To reduce the bias caused by the masked pixels, we carefully select non-overlapping square maps within the footprint, ensuring that at least 98% of each selected region remains unmasked. This criterion helps to preserve the integrity of the extracted features and minimizes contamination from incomplete areas, leading to more reliable and unbiased results in the subsequent analysis. Fig. 7.13 indicates the regions where we extract square maps from the DES Y3 footprint. And Fig. 7.14 demonstrates the extracted square map and its binary mask, where the yellow regions are useful pixels and purple regions are masked pixels.

The scattering transform can be viewed as a type of convolution, or equivalently, a weighted summation where the weights decay with increasing distance from the center pixel. This means that even if the center pixel itself is unmasked, masked neighboring pixels can still affect its value. Therefore, to obtain more accurate scattering coefficients in the presence of masking, we aim to exclude contaminated pixels. Specifically, we select pixels according to the following criteria:

- The center pixel must not be masked since the wavelet gives the highest weight for the center pixels.
- If more than 5% of the neighboring pixels around a center pixel are masked, the center pixel is also considered masked.

Here, the definition of “neighbor” requires further clarification:

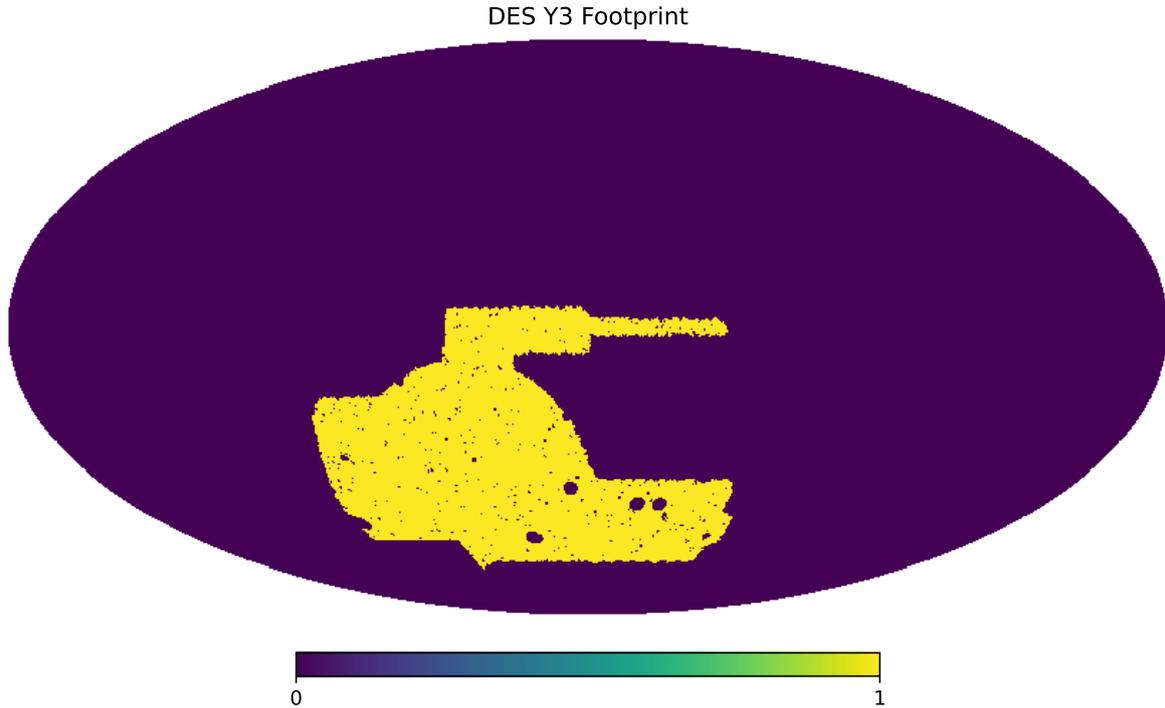


Figure 7.12: DES Y3 Footprint of Tomographic bin 3. The yellow part represents the observed regions while the purple part is the masked or unobserved regions.

1. We first sum the modulus of all wavelet filters of the same scale, which are basically the Gaussian envelopes, but different orientations, as shown in Fig. 7.15, to obtain an approximately isotropic filter that depends only on the distance from the center pixel, rather than the direction.
2. We define the effective area of this combined wavelet filter as the region where the filter values lie between the maximum value and 10% of the maximum value, shown in Fig. 7.16.

As we can see, the wavelet filter becomes approximately isotropic after summing over all orientations, and its effective region maintains a shape similar to that of the original directional filters. Based on this observation, we present the final mask settings for square maps at different scales, as shown in Fig. 7.17. These mask settings are derived by extending the original mask shown in Fig. 7.14b according to the effective support of each wavelet filter.

From the figure, we observe that for $j = 0$, the updated masked regions are nearly identical to the original. This is because the $j = 0$ wavelet filter has a very small effective area, involving only the four immediate neighboring pixels around the center. As a result, the mask extension is minimal.

For $j = 1$ to $j = 3$, the mask settings depend on the spatial distribution of the originally

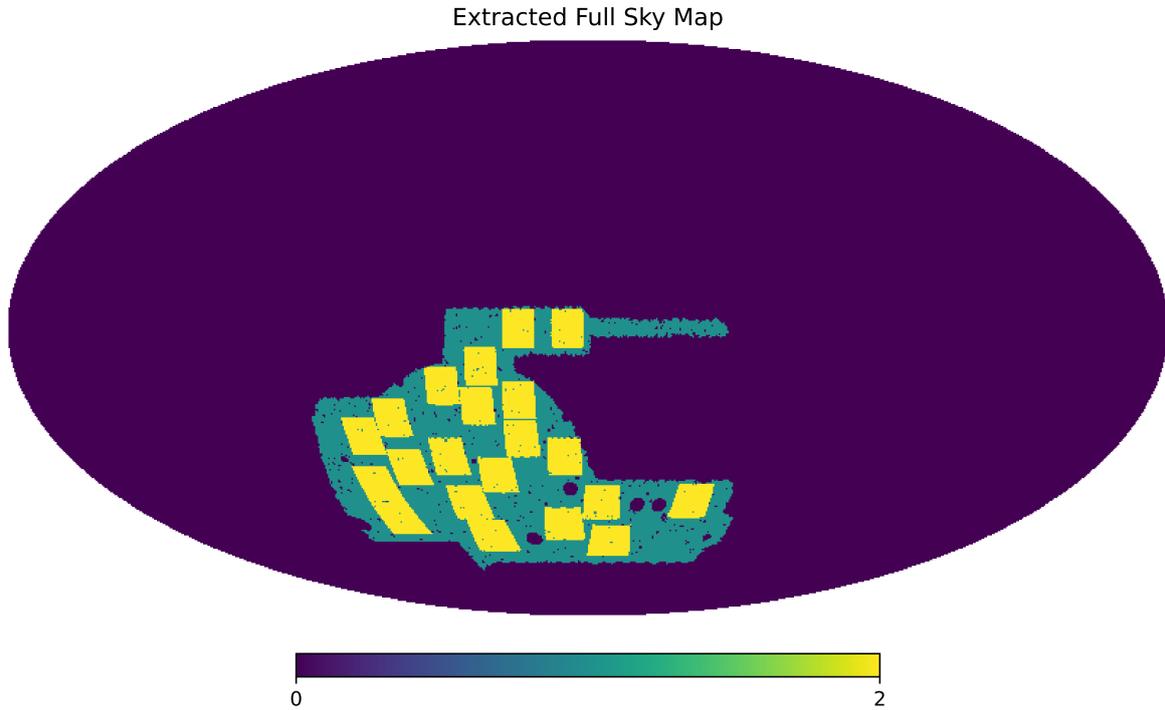


Figure 7.13: Extracted Full Sky Map. Purple indicates unobserved or masked regions, while green and yellow together show the DES Y3 footprint. The yellow areas are the square maps that have been extracted.

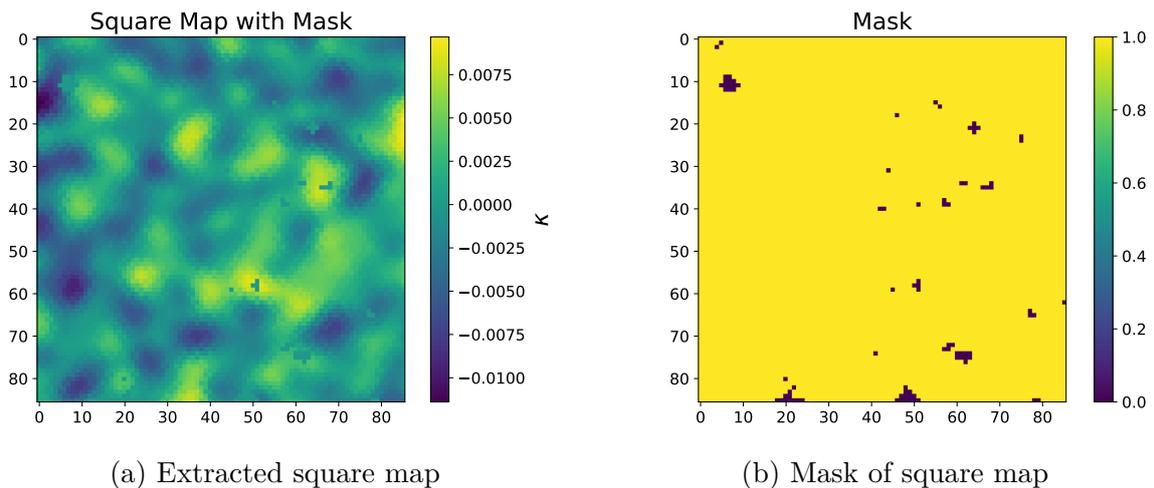


Figure 7.14: Left panel: the extracted κ map from the DES Y3 footprint, with missing regions masked. Right panel: the corresponding binary mask indicating observed pixels. The yellow areas correspond to the extracted square maps within the DES Y3 footprint, while the dark purple regions denote masked pixels.

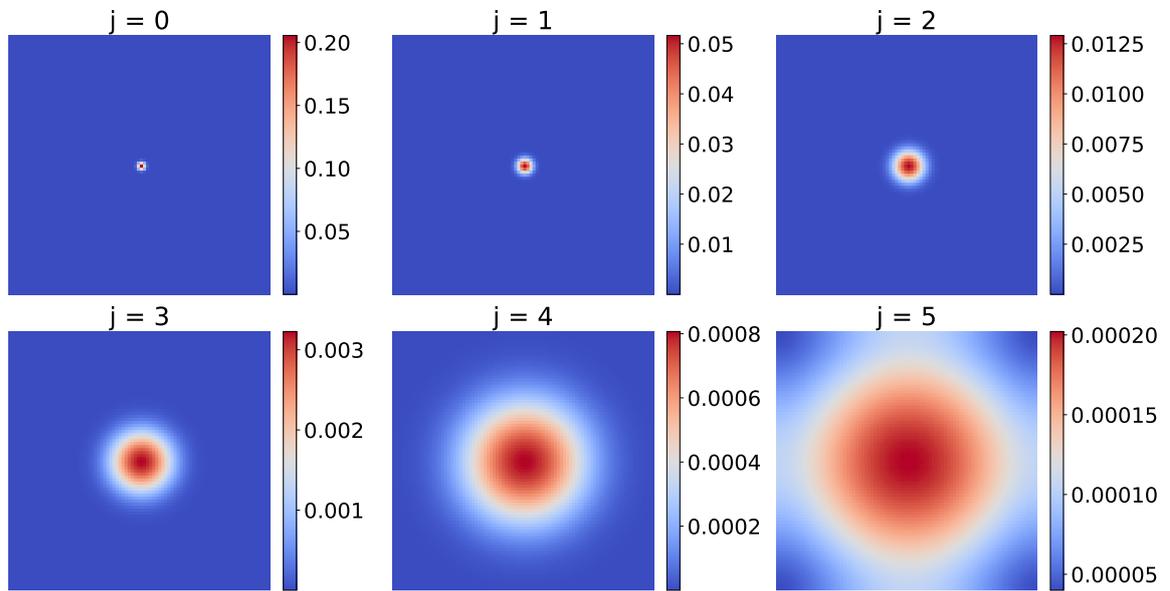


Figure 7.15: Symmetric wavelet filters in real space.

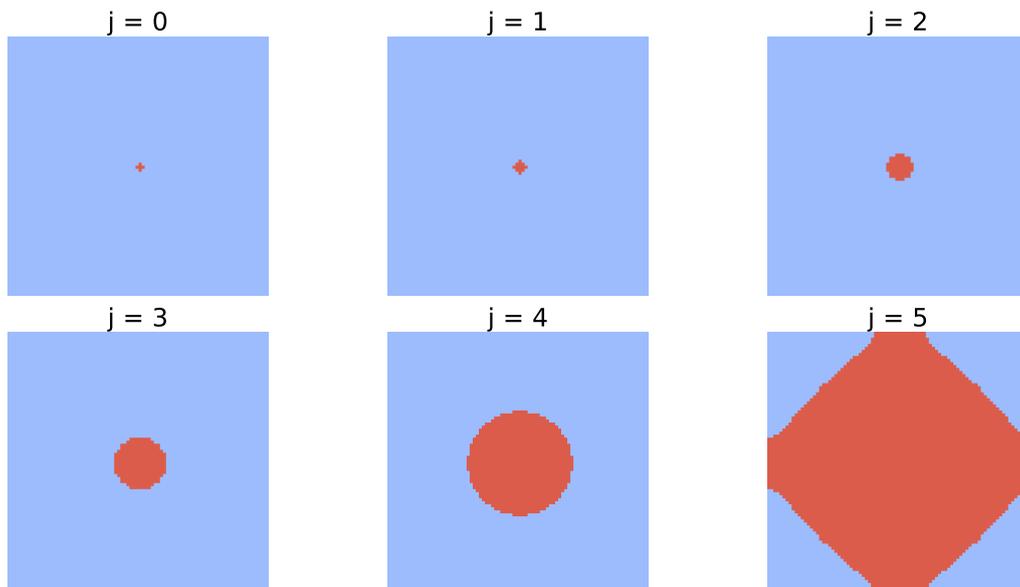


Figure 7.16: Effective regions of wavelet filters shown in Fig. 7.15

masked pixels. If the masked pixels are clustered, the effective mask region becomes significantly larger due to overlap in the filter support. In contrast, if the originally masked pixels are spatially isolated, the new mask remains nearly the same as the original.

For higher scales ($j \geq 4$), the mask settings are exactly the same as the original mask. This is because the wavelet filters at these scales have effective support that covers nearly the entire square map, and the overall masking fraction is less than 2%, which falls below the 5% threshold for excluding a pixel.

With all mask settings prepared, we can now proceed to the calculation of scattering coefficients. In the original case without masking, the scattering coefficients are computed by simply taking the spatial average of the modulus of the convolved field. However, with masked pixels present, it is necessary to account for them by excluding masked regions when performing the spatial averaging. A method proposed in Ref. [92] addresses this by removing only the originally masked pixels. To further improve the accuracy of the scattering coefficient calculation, particularly for small-scale wavelet filters, we additionally exclude pixels that are surrounded by masked pixels, as discussed above. Also, when calculating the second order scattering coefficients, we need to use two different wavelet filters. If the pixel is set to be masked by any of the wavelet filter, then we assume it's to be masked when calculating the second-order scattering coefficients.

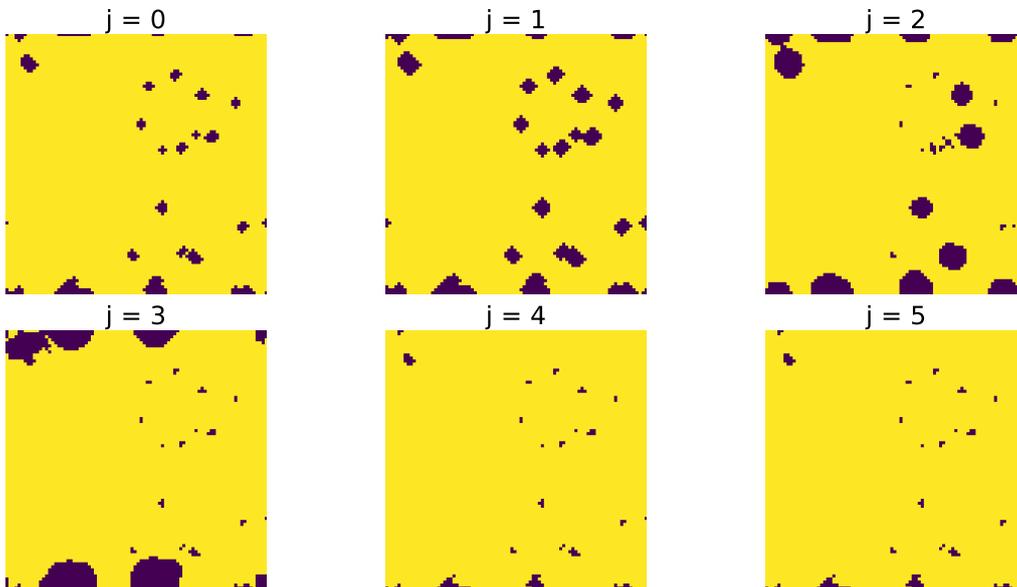


Figure 7.17: Mask settings for different wavelet filters.

Fig. 7.18 shows the comparison of scattering coefficients between true values and the values calculated from different methods. From the figure, we could see that the relative error decreases a lot for the small scales comparing to the method that only removes the originally masked pixels when taking the spatial average, this is again because we remove the unmasked pixels that are close to the originally masked pixels. To further quantify

the error caused by the mask, we calculate the χ^2 of these two methods:

$$\chi^2 = (\mathbf{s}_{mask} - \mathbf{s}_{true}) \mathbf{C}^{-1} (\mathbf{s}_{mask} - \mathbf{s}_{true}) / \text{D.O.F} , \quad (7.41)$$

where \mathbf{s}_{mask} is the scattering coefficients computed from masked maps while the \mathbf{s}_{true} is the true value or the coefficients from unmasked square maps, and D.O.F is the degree of freedom. The reduced χ^2 for the blue dots are only about 1/7 of the green dots, indicating the great improvement of the accuracy of the scattering coefficients calculation.

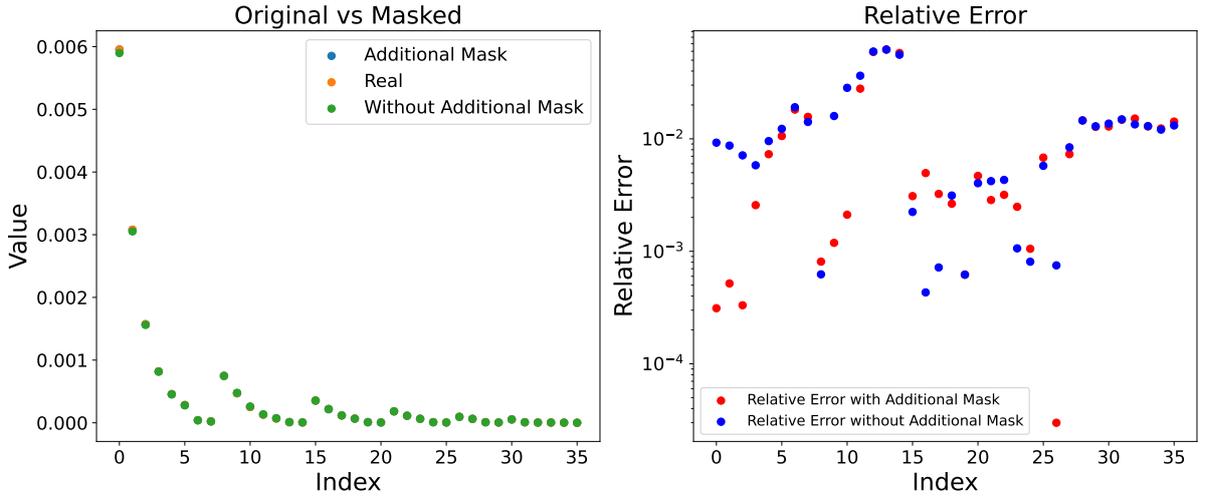


Figure 7.18: Comparison of the scattering coefficients under different mask dealing method. The left panel: blue dots are the mask setting method discussed in this section, orange dots are assumed to be the real scattering coefficients that are calculated from square maps without any mask and green dots are the result. The “Additional Mask” here means that the pixels that are surrounded by masked pixels are set to be masked as well.

Fig. 7.19 shows the MCMC constraints the marginalized posterior distributions of three cosmological parameters, Ω_m , σ_8 and S_8 , obtained from MCMC sampling, where S_8 is defined in the following equation:

$$S_8 = \sigma_8 \left(\frac{\Omega_m}{0.3} \right)^{0.5} . \quad (7.42)$$

Two cases are compared: one using convergence maps that contain masked pixels (green contours and lines), and one using unmasked maps (blue). The masking is based on the DES Y3 footprint and applied to the simulated maps to mimic realistic survey conditions. The results are listed in Table 7.2.

We observe that the impact of masking is relatively minor. The overall shape and location of the posteriors remain consistent between the two cases, with only a light shift of the maximum a posterior position in the contours when masked pixels are included.

Case	Ω_m	σ_8	S_8
Fiducial	0.260	0.840	0.782
Without mask	$0.282^{+0.056}_{-0.061}$	$0.821^{+0.100}_{-0.098}$	$0.795^{+0.045}_{-0.053}$
With mask	$0.275^{+0.049}_{-0.060}$	$0.825^{+0.081}_{-0.092}$	$0.776^{+0.045}_{-0.049}$

Table 7.2: Summary of cosmological constraints for Ω_m , σ_8 , and S_8 with and without masking.

This suggests that the masking treatment in our pipeline is effective, and the presence of masked regions does not significantly bias the cosmological parameter constraints.

Unexpectedly, we find that the posterior contours obtained from masked data are in some cases slightly tighter than those obtained from unmasked data, particularly for σ_8 . This is counter-intuitive, as masking typically removes information and increases statistical uncertainty. It is important to note that the emulators were trained on unmasked data only, which makes this observation even more surprising.

One possible explanation is that masking may incidentally suppress certain noisy, leading to a smoother or more “regularized” parameter likelihood. Another possibility is that the apparent tightening arises from statistical fluctuations, particularly if the sample variance in the masked maps happens to align more coherently with the emulator’s prediction space. In the masked case, the scattering coefficients are computed as the average over approximately 20 realizations (corresponding to the number of “good” square maps that can be extracted from the DES Y3 footprint), whereas in the unmasked case, the coefficients are averaged over 260 realizations. Further investigation is required to fully understand this effect, but overall, the difference is small and does not significantly alter the inferred cosmological constraints.

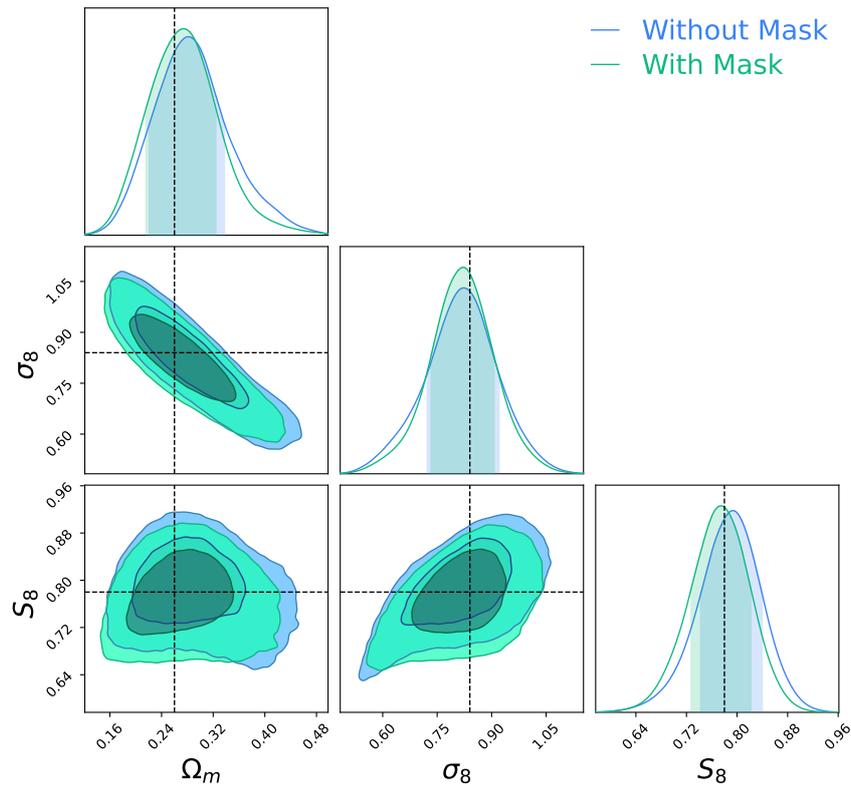


Figure 7.19: Marginalized posterior distributions for Ω_m , σ_8 , and S_8 from MCMC sampling using four tomographic bins. Blue contours and lines correspond to the case without masking, while green represents the case with masked pixels in the input convergence maps. The masks are based on the DES Y3 footprint. **Note:** this is still a simulated likelihood analysis, using realistic survey masking but not real observational data.

Chapter 8

Conclusion

In this thesis, we have explored the use of scattering transform coefficients as a summary statistic for weak gravitational lensing maps, with the aim of constraining cosmological parameters beyond what is achievable using conventional two-point statistics. The scattering transform is a hierarchical, nonlinear, and translation-invariant feature extractor that is structurally inspired by convolutional neural networks but fully deterministic and interpretable. It preserves multiscale information in the input field by cascading wavelet modulus operators and averaging, thereby encoding non-Gaussian features and complex structures in a way that is robust to noise and local deformation.

The motivation for employing the scattering transform in the context of cosmology stems from the growing interest in non-Gaussian statistics. While the power spectrum (or two-point correlation function) captures all the information in Gaussian fields, the weak lensing convergence field is inherently non-Gaussian on small and intermediate scales due to nonlinear structure formation. Traditional approaches such as higher-order moments, bispectrum, and peak statistics attempt to capture this non-Gaussian information, but often come with challenges related to noise sensitivity and complex theoretical modeling. The scattering transform offers a promising alternative by encoding a broad range of non-Gaussian features in a structured and physically interpretable manner.

Throughout this work, we computed the first- and second-order scattering coefficients on simulated weak lensing convergence maps from the CosmoGridV1 simulation suite. These coefficients were computed for various tomographic bins and angular scales, allowing us to retain both redshift and spatial information. We examined the behavior of the scattering coefficients under variations in cosmological parameters, intrinsic alignment amplitude, and several nuisance systematics. We then constructed emulators that map cosmological parameters to scattering coefficients using neural networks, and used these to perform parameter inference via MCMC.

Our results demonstrate that the scattering transform is particularly sensitive to the parameters Ω_m and σ_8 , which directly control the amplitude and shape of the matter power spectrum. The parameter $S_8 = \sigma_8 (\Omega_8/0.3)^{0.5}$, which is well-constrained by weak lensing observables, is also effectively constrained by the scattering statistics. On the other hand,

parameters such as w_0 , n_s , H_0 and Ω_b , which primarily affect background evolution or detailed shape changes of the power spectrum, are less constrained by scattering transform coefficients. Intrinsic alignment parameters and shear calibration systematics also show limited sensitivity.

Importantly, we carried out masking-aware preprocessing to handle real survey conditions, such as those mimicked from the DES Y3 footprint. By applying a conservative masking rule based on the wavelet effective support, we ensured that the scattering coefficients were computed only from reliable, unmasked regions. Our masking strategy introduces a small information loss but avoids the risk of contamination from missing pixels. The robustness of the method was validated through a comparison between masked and unmasked maps, with posterior contours remaining largely consistent.

We also assessed the accuracy and reliability of the emulator and inference pipeline through convergence tests and visual inspection of MCMC posterior. The fiducial cosmology was found to lie well within 68% credible regions for Ω_m , σ_8 and S_8 in all tomographic bins, indicating that the emulator was sufficiently accurate for the parameter space explored. The sharp boundaries seen in certain parameter planes reflect the imposed hard prior ranges rather than emulator artifacts.

In conclusion, the scattering transform provides a powerful and interpretable summary statistic for weak lensing analysis. It enables the extraction of non-Gaussian information in a way that complements and extends traditional two-point statistics. Our work confirms that scattering statistics can deliver competitive constraints on key cosmological parameters such as Ω_m , σ_8 and S_8 with robustness to moderate levels of noise and masking. As future large-scale surveys like Euclid deliver higher-fidelity lensing maps with greater sky coverage and tomographic resolution, we anticipate that methods like the scattering transform will play an increasingly important role in extracting the full statistical power of the data.

Looking forward, several avenues for improvement and extension remain. First, reducing the dimensionality of the scattering coefficients using PCA or autoencoders could improve emulators performance and inference speed. Second, we are currently applying the developed pipeline to real DES Y3 data, which will allow us to evaluate the practical effectiveness of the scattering transform in a realistic survey setting.

Appendix A

Hankel transform

In this appendix, we will derive Hankel transform, a special kind of Fourier transform in the context of isotropy.

First, we have a brief introduction to the Bessel function, in the form of integral. The mathematical expression for n th order Bessel function (n is an integer here) is expressed as:

$$J_n(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(n\tau - x \sin \tau)} d\tau. \quad (\text{A.1})$$

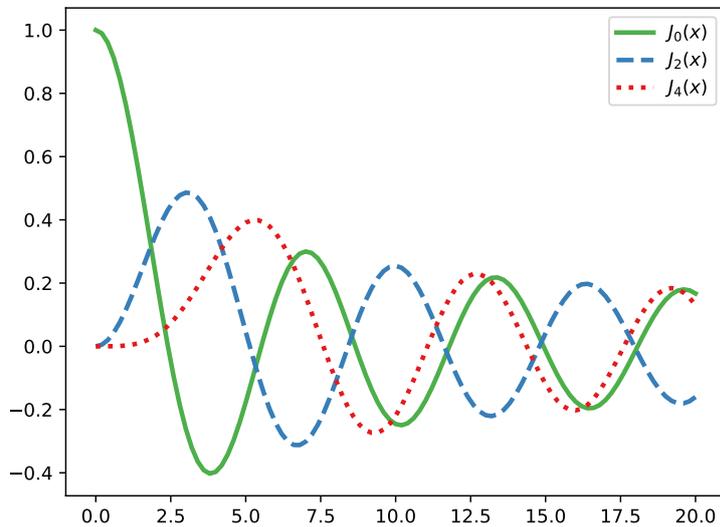


Figure A.1: The filter functions $J_0(x)$ and $J_4(x)$ play a role in the relationships between the cosmic shear correlation functions $\xi_{\pm}(\theta)$ and the convergence power spectrum $P_{\kappa}(\ell)$. For a given angular scale θ , the filter $J_0(x) = J_0(\theta\ell)$ places more emphasis on lower frequencies ℓ compared to the filter $J_4(x) = J_4(\theta\ell)$. As a result, the function $\xi_{+}(\theta)$ is more responsive to large-scale density fluctuations than $\xi_{-}(\theta)$ at the same angular scale θ .

We consider the two-dimensional Fourier transform of a function $\varphi(\mathbf{x})$, which shows a circular symmetry. This means that $\varphi(r \cos \theta, r \sin \theta) \equiv f(r, \theta)$ is independent of θ . The Fourier transform of φ is

$$\Phi(\zeta, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i(x\zeta + y\eta)} dx dy . \quad (\text{A.2})$$

where we introduce two polar coordinates:

$$x = r \cos \theta \quad y = r \sin \theta , \quad (\text{A.3})$$

and

$$\zeta = \rho \cos \varphi \quad \eta = \rho \sin \varphi . \quad (\text{A.4})$$

Because of the circular symmetry, we have:

$$\begin{aligned} \Phi(\rho \cos \varphi, \rho \sin \varphi) &\equiv F(\rho, \varphi) \\ &= \int_0^{\infty} \int_0^{2\pi} f(r) e^{-ir\rho \cos(\theta - \varphi)} r dr d\theta \\ &= \int_0^{\infty} r f(r) dr \int_0^{2\pi} e^{-ir\rho \cos(\theta - \varphi)} d\theta \\ &= \int_0^{\infty} r f(r) J_0(\rho r) dr . \end{aligned} \quad (\text{A.5})$$

This shows that $F(\rho, \phi)$ is independent of ϕ so that we can write $F(\rho)$ instead of $F(\rho, \phi)$. Therefore, the two-dimensional Fourier transform of a circularly symmetric function is, in fact, a Hankel transform of order zero.

In general, Hankel transform could be regarded as the circularly symmetric form of Fourier transform. And we could, therefore, write down the Hankel transform in the following way:

$$\begin{aligned} F(\rho) &= 2\pi \int_0^{\infty} r f(r) J_n(\rho r) dr = 2\pi H_n[f(r)] , \\ f(r) &= \frac{1}{2\pi} \int_0^{\infty} \rho F(\rho) J_n(\rho r) d\rho = \frac{1}{2\pi} H_n^{-1}[F(\rho)] . \end{aligned} \quad (\text{A.6})$$

If we apply this into the shear-shear correlation function, the derivation follows. First, let's expand shear with Fourier transform,

$$\gamma(\boldsymbol{\theta}) = \int \frac{d^2\ell}{(2\pi)^2} \gamma(\boldsymbol{\ell}) e^{i\boldsymbol{\ell} \cdot \boldsymbol{\theta}} , \quad (\text{A.7})$$

$$\gamma(\boldsymbol{\theta} + \boldsymbol{\alpha}) = \int \frac{d^2\mathbf{k}}{(2\pi)^2} \gamma(\mathbf{k}) e^{i\mathbf{k} \cdot (\boldsymbol{\theta} + \boldsymbol{\alpha})} . \quad (\text{A.8})$$

With Kaiser-Squire relation, Eq. (3.39), we can reformulate above equations as:

$$\gamma(\boldsymbol{\theta}) = \int \frac{d^2\ell}{(2\pi)^2} \kappa(\boldsymbol{\ell}) e^{i2\phi\boldsymbol{\ell}} e^{i\boldsymbol{\ell} \cdot \boldsymbol{\theta}} , \quad (\text{A.9})$$

$$\gamma(\boldsymbol{\theta} + \boldsymbol{\alpha}) = \int \frac{d^2\mathbf{k}}{(2\pi)^2} \kappa(\mathbf{k}) e^{i2\phi\mathbf{k}} e^{i\mathbf{k} \cdot (\boldsymbol{\theta} + \boldsymbol{\alpha})} . \quad (\text{A.10})$$

Then the correlation function between these two shear is given:

$$\begin{aligned}
 \langle \gamma(\boldsymbol{\theta}) \gamma(\boldsymbol{\theta} + \boldsymbol{\alpha}) \rangle &= \frac{1}{(2\pi)^4} \left\langle \int d^2\ell \int d^2k \kappa(\boldsymbol{\ell}) \kappa(\mathbf{k}) e^{i2\phi_\ell} e^{i2\phi_k} e^{i\boldsymbol{\ell} \cdot \boldsymbol{\theta}} e^{i\mathbf{k} \cdot (\boldsymbol{\theta} + \boldsymbol{\alpha})} \right\rangle \\
 &= \frac{1}{(2\pi)^4} \int d^2\ell \int d^2k \langle \kappa(\boldsymbol{\ell}) \kappa(\mathbf{k}) \rangle e^{i2\phi_\ell} e^{i2\phi_k} e^{i\boldsymbol{\ell} \cdot \boldsymbol{\theta}} e^{i\mathbf{k} \cdot (\boldsymbol{\theta} + \boldsymbol{\alpha})} \\
 &= \frac{1}{(2\pi)^4} \int d^2\ell \int d^2k (2\pi)^2 \delta_D(\mathbf{k} + \boldsymbol{\ell}) P_\kappa(\boldsymbol{\ell}) e^{i2\phi_\ell} e^{i2\phi_k} e^{i\boldsymbol{\ell} \cdot \boldsymbol{\theta}} e^{i\mathbf{k} \cdot (\boldsymbol{\theta} + \boldsymbol{\alpha})} \\
 &= \frac{1}{(2\pi)^2} \int d^2\ell P_\kappa(\boldsymbol{\ell}) e^{i4\phi_\ell} e^{-i\boldsymbol{\ell} \cdot \boldsymbol{\alpha}} \\
 &= \int \frac{d\ell}{2\pi} \ell P_\kappa(\ell) J_4(\ell\alpha) .
 \end{aligned} \tag{A.11}$$

Appendix B

Discrete Fourier transform and power spectrum

In this appendix, we will introduce the Discrete Fourier Transform (DFT). In this project, we used `get_power_spectrum`¹ function to measure the binned power spectrum.

In Section 2.2, we introduced the continuous Fourier transform and its formalism. However, since real-world signals are not of infinite length, the Fourier transform cannot be expressed as an integral over the entire real line from negative infinity to positive infinity. Instead, it is represented as a summation given by:

$$F(k_n) = \sum_{n=-N}^N f(x) e^{-ik_n x} , \quad (\text{B.1})$$

where $F(k_n)$ represents the amplitude at frequency k_n and the summation consists of $2N + 1$ terms. For a function $f(x)$ defined on the interval $[0, L]$, we assume it is periodically extended over the entire real line $(-\infty, +\infty)$. The function $f(x)$ could be extended as:

$$f(x) = \sum_{n=-N}^N F(k_n) e^{ik_n x} , \quad x \in [0, L] . \quad (\text{B.2})$$

Because of the periodic property, we have:

$$f(x \pm L) = f(x) . \quad (\text{B.3})$$

Using Eq. (B.2), we could rewrite Eq. B.3 as

$$\sum_{n=-N}^N F(k_n) e^{ik_n(x \pm L)} = \sum_{n=-N}^N F(k_n) e^{ik_n L} . \quad (\text{B.4})$$

This holds for any $x \in [0, L]$. So, we could easily see that

$$e^{\pm ik_n L} = 1 \quad \text{for all } n . \quad (\text{B.5})$$

¹https://github.com/SihaoCheng/scattering_transform/blob/master/ST.py

Therefore, we have to choose the frequency that satisfies the following equation:

$$k_n L = 2m\pi \quad m \in Z . \quad (\text{B.6})$$

So, k_n could be written as $k_n = \frac{2\pi n}{L}$, where $n = 0, \pm 1, \pm 2, \dots, \pm N, \dots$. For an interval $[0, L]$, we assume that $T = 2N + 1$ points are evenly spaced across it.

$$f(x_m) = f\left(\frac{L}{T} * m\right) \quad \text{where } m = 0, 1, 2, \dots, 2N . \quad (\text{B.7})$$

To match the sampling rate, we need $2N + 1$ frequencies as well. In the following, we will introduce another property of DFT. By making $n = N + 1$ and $n = -N$, we have:

$$k_{N+1} - k_{-N} = \frac{2\pi}{L} \times (N + 1) - \frac{2\pi}{L} \times (-N) = \frac{2\pi}{L} (2N + 1) . \quad (\text{B.8})$$

This leads the equivalent property of Fourier kernel:

$$\begin{aligned} e^{ik_{N+1}x_m} &= e^{i[k_{-N} + \frac{2\pi}{L} \times (2N+1)]x_m} \\ &= e^{ik_{-N}x_m} \times e^{i\frac{2\pi}{L} \times (2N+1)x_m} \\ &= e^{ik_{-N}x_m} \times e^{i\frac{2\pi}{L} \times (2N+1)\frac{L}{2N+1}m} \\ &= e^{ik_{-N}x_m} \times e^{i2\pi m} \\ &= e^{ik_{-N}x_m} \times 1 . \end{aligned} \quad (\text{B.9})$$

With this property, we can easily see that there are two equivalent choices of frequency sets:

$$k_n = \frac{2\pi}{L} [0, 1, 2, \dots, N, N + 1, N + 2, \dots, 2N] , \quad (\text{B.10})$$

$$k_n = \frac{2\pi}{L} [0, 1, 2, \dots, N, -N, -(N - 1), \dots, -2, -1] . \quad (\text{B.11})$$

In this thesis, we will take the Eq. (B.11) convention.

We assume that the input image has a dimension of (M, N) , where M is an even number and N is an odd number. Then, in `numpy.fft.ffft2`, the elements of the resulting Fourier-transformed matrix are distributed across the following frequency components:

$$\left[\begin{array}{cccc|cccc} (0, 0) & (0, 1) & \cdots & (0, \frac{N-1}{2}) & (0, -\frac{N-1}{2}) & (0, -(\frac{N-1}{2} - 1)) & \cdots & (0, -2) & (0, -1) \\ (1, 0) & (1, 1) & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\frac{M}{2} - 1, 0) & \vdots & \ddots & (\frac{M}{2} - 1, \frac{N-1}{2}) & (\frac{M}{2} - 1, -\frac{N-1}{2}) & \vdots & \ddots & \vdots & (\frac{M}{2} - 1, -1) \\ \hline (-\frac{M}{2}, 0) & \vdots & \ddots & (-\frac{M}{2}, \frac{N-1}{2}) & (-\frac{M}{2}, -\frac{N-1}{2}) & \vdots & \ddots & \vdots & (-\frac{M}{2}, -1) \\ (-\frac{M}{2} - 1, 0) & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & (-\frac{M}{2} - 1, -1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (-2, 0) & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & (-2, -2) & (-2, -1) \\ (-1, 0) & \cdots & \cdots & (-1, \frac{N-1}{2}) & (-1, -\frac{N-1}{2}) & (-1, -(\frac{N-1}{2} - 1)) & \cdots & (-1, -2) & (-1, -1) \end{array} \right] . \quad (\text{B.12})$$

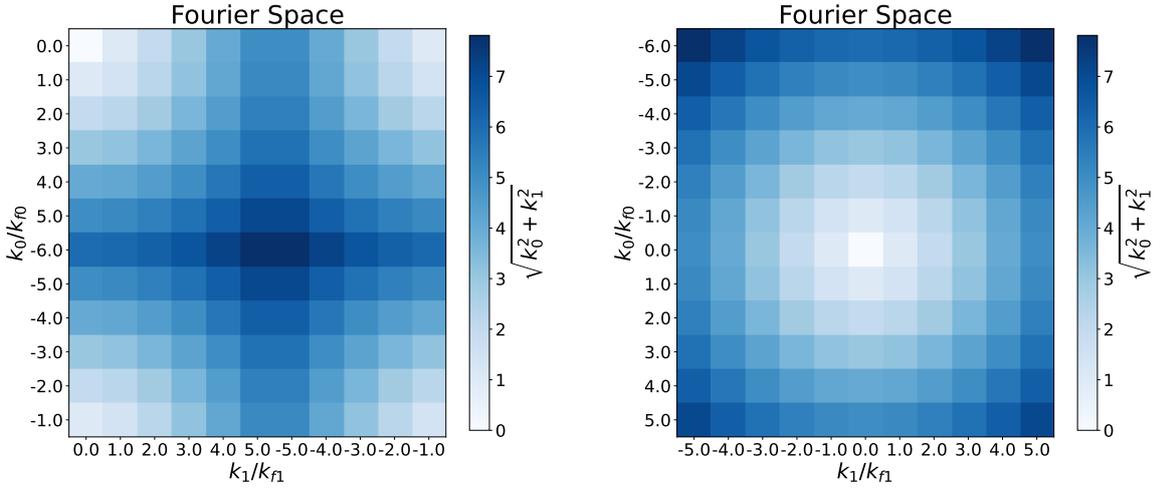
where the first number in the parentheses represents the frequency along axis-0, and the second number corresponds to the frequency along axis-1. And the vertical and horizontal

lines here divide the matrix into four pieces which could be used in the following.

When calculating the power spectrum, we are concerned only with the magnitude of the frequency, not the direction of the frequency vector. To achieve this, the matrix is divided into four sections, and these sections are shifted so that the low-frequency components from each section are aligned at the center of the matrix:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \rightarrow \left[\begin{array}{c|c} D & C \\ \hline B & A \end{array} \right]. \quad (\text{B.13})$$

where A , B , C , D represent the top-left, top-right, bottom-left and bottom-right sections of the matrix in Eq. (B.12). To compute the magnitude of the frequency, we first calculate the distance of each point in the shifted matrix Eq. (B.13) to its center. Then, we define a logarithmic range of k , and for each interval $[k[i], k[i + 1])$, we compute the mean value of the Fourier-transformed modes whose distances fall within this range. This mean value is taken as the power spectrum at the frequency $k[i]$. Fig. B.1 and (B.2) shows the process of calculating the binned power spectrum with a specific example.



(a) Before shifting frequency to center

(b) After shifting frequency to center

Figure B.1: These two figures show the frequency distribution after using `numpy.fft.fftn` to do DFT. Left figure is the frequency distribution without using `numpy.fft.fftshift` to move the low frequency to the center, while right one used. They both follow the convention of python, which means that the vertical direction is axis-0 starting from top to bottom, and horizontal direction is axis-1 starting from left to right. Here I set vertical length to be $M = 12$ and horizontal length to be $N = 11$. Two k_f shown in the x/y labels are the basis frequency so that we have $k_{f0} = \frac{2\pi}{M}$ and $k_{f1} = \frac{2\pi}{N}$, and then the number of two axes are the relative value corresponding to their basis frequency, following the convention described in Eq. (B.11). So after the DFT, the total frequency is decreasing from the center to four corners. Then, we shift the sections, like Eq. (B.13), so that the lowest total frequency, $\sqrt{k_0^2 + k_1^2} = 0$, is located in the center of the matrix.

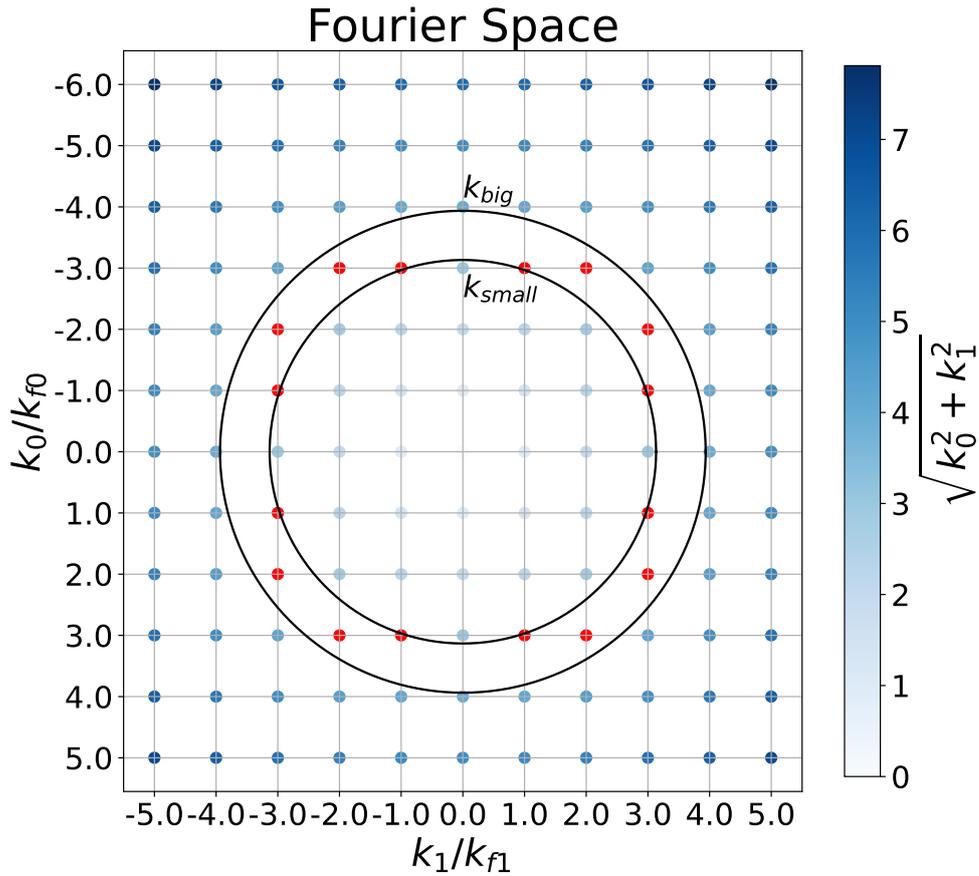


Figure B.2: Calculate the average power spectrum in each frequency bins. The Fourier transformed value at each point is calculated using the frequency value corresponding to the coordinates (pair of frequency) of that point. After shifting the zero frequency to the center, we set evenly distributed logarithm scale range starting from 0 to maximum total frequency. Then we calculate the binned power spectrum by averaging the DFT value whose total frequency lies in the rings of logarithm scale range (regions between k_{small} and k_{big}), shown as red points in the plot.

Appendix C

Translation Invariance

In this appendix, we will introduce some basic properties of translation-invariance and show that why we need non-linearity to construct the estimators for scattering transform.

First, let's begin with the linear operator. The following two properties are the definition of linear operator:

$$h(\mathbf{u} + \mathbf{v}) = h(\mathbf{u}) + h(\mathbf{v}) , \quad (\text{C.1})$$

$$h(c\mathbf{u}) = ch(\mathbf{u}) , \quad (\text{C.2})$$

where $h(\cdot)$ is a linear operator, \mathbf{u} and \mathbf{v} are the arbitrary vectors, c is a constant.

Another operation is *orthogonal transformation*. An orthogonal operator is a linear operator T on a finite-dimensional or infinite-dimensional inner product space (usually over real space \mathbb{R} or complex space \mathbb{C}) that preserves the inner product of vectors. Mathematically, for any vector \mathbf{u} and \mathbf{v} in the space, we have:

$$\langle T(\mathbf{u}), T(\mathbf{v}) \rangle = \langle \mathbf{u}, \mathbf{v} \rangle , \quad (\text{C.3})$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Now, we could introduce translation transformation. A translation operator is a mathematical operator that shifts a function, vector or field by a fixed amount in a given direction, without altering its shape or other intrinsic properties. Let $L_{\mathbf{c}}$ denote the translation operator that shifts a function $f(\mathbf{x})$ by \mathbf{c} . The action of $L_{\mathbf{c}}$ on $f(\mathbf{x})$ is defined as:

$$L_{\mathbf{c}}f(\mathbf{x}) = f(\mathbf{x} - \mathbf{c}) . \quad (\text{C.4})$$

Translation operator is also an orthogonal operator, which means:

$$\langle L_{\mathbf{c}}f, L_{\mathbf{c}}g \rangle = \langle f, g \rangle . \quad (\text{C.5})$$

This property could be proved in the following way:

$$\begin{aligned}
 \langle L_{\mathbf{c}}f, L_{\mathbf{c}}g \rangle &= \int_{-\infty}^{\infty} f(\mathbf{x} - \mathbf{c}) g(\mathbf{x} - \mathbf{c}) d\mathbf{x} \\
 &= \int_{-\infty}^{\infty} f(\mathbf{x}') g(\mathbf{x}') d\mathbf{x}' \\
 &= \langle f, g \rangle .
 \end{aligned} \tag{C.6}$$

The third important operator is *convolution operator*, which is also a kind of linear operation. The convolution of two functions f and g is defined as:

$$(f \star g)(\mathbf{x}) = \int f(\mathbf{v}) g(\mathbf{x} - \mathbf{v}) d\mathbf{v} . \tag{C.7}$$

Then its linearity follows easily:

$$\begin{aligned}
 (f \star (\alpha g_1 + \beta g_2))(\mathbf{x}) &= \int f(\mathbf{v}) (\alpha g_1(\mathbf{x} - \mathbf{v}) + \beta g_2(\mathbf{x} - \mathbf{v})) d\mathbf{v} \\
 &= \int \alpha f(\mathbf{v}) g_1(\mathbf{x} - \mathbf{v}) + \int \beta f(\mathbf{v}) g_2(\mathbf{x} - \mathbf{v}) d\mathbf{v} \\
 &= \alpha (f \star g_1)(\mathbf{x}) + \beta (f \star g_2)(\mathbf{x}) .
 \end{aligned} \tag{C.8}$$

In this project, we used wavelet convolution, and this operator commutes with translation operator. The mathematical expression can be written as:

$$L_{\mathbf{c}}(f \star \psi) = (L_{\mathbf{c}}f) \star \psi , \tag{C.9}$$

where ψ here denotes the convolution kernel and \star is the convolution operator. The proof of this property follows. First, the expression of convolution between ψ and $f(\mathbf{x})$ is given by:

$$(f \star \psi)(\mathbf{x}) = \int f(\mathbf{v}) \psi(\mathbf{x} - \mathbf{v}) d\mathbf{v} . \tag{C.10}$$

Then, the translation of the convolution is:

$$\begin{aligned}
 L_{\mathbf{c}}(f \star \psi)(\mathbf{x}) &= (f \star \psi)(\mathbf{x} - \mathbf{c}) \\
 &= \int f(\mathbf{v}) \psi((\mathbf{x} - \mathbf{c}) - \mathbf{v}) d\mathbf{v} \\
 &= \int f(\mathbf{v}) \psi(\mathbf{x} - \mathbf{c} - \mathbf{v}) d\mathbf{v} .
 \end{aligned} \tag{C.11}$$

And we do the convolution with translated $f(\mathbf{x})$:

$$\begin{aligned}
 ((L_{\mathbf{c}}f) \star \psi)(\mathbf{x}) &= \int (L_{\mathbf{c}}f)(\mathbf{v}) \psi(\mathbf{x} - \mathbf{v}) d\mathbf{v} \\
 &= \int f(\mathbf{v} - \mathbf{c}) \psi(\mathbf{x} - \mathbf{v}) d\mathbf{v} \\
 &= \int f(\mathbf{w}) \psi(\mathbf{x} - \mathbf{c} - \mathbf{w}) d\mathbf{w} \\
 &= \int f(\mathbf{v}) \psi(\mathbf{x} - \mathbf{c} - \mathbf{v}) d\mathbf{v} ,
 \end{aligned} \tag{C.12}$$

where we used $\mathbf{v} = \mathbf{w} + \mathbf{c}$ from the second line to third line, and then $\mathbf{v} = \mathbf{w}$ for the variable substitution from the third line to the fourth line. Comparing Eq. (C.11) and (C.12), we could find that translation operator commutes with convolution operator.

If R is a linear or non-linear operator which commutes with translation, $R(L_{\mathbf{c}}f(\mathbf{x})) = L_{\mathbf{c}}(Rf(\mathbf{x}))$, then the integral $\int Rf(\mathbf{x}) d\mathbf{x}$ is translation invariant. The proof is given:

$$\begin{aligned}
 L_{\mathbf{c}}\left(\int Rf(\mathbf{x}) d\mathbf{x}\right) &= \int L_{\mathbf{c}}Rf(\mathbf{x}) d\mathbf{x} \\
 &= \int RL_{\mathbf{c}}f(\mathbf{x}) d\mathbf{x} \\
 &= \int Rf(\mathbf{x} - \mathbf{c}) d\mathbf{x} \\
 &= \int Rf(\mathbf{x}) d\mathbf{x} .
 \end{aligned} \tag{C.13}$$

If R here is the wavelet convolution kernel, ψ , we use in this project, then $\int f(\mathbf{x}) \star \psi(\mathbf{x}) d\mathbf{x}$ gives a translation invariant value.

The second thing we need to prove now is that why we need non-linearity to construct the estimators for scattering transform. Suppose we have a cosmological field, like convergence field, $\kappa(\mathbf{x})$, which has vanishing mean. And we apply some linear operator like convolution or Fourier transform to this field, and calculate the spatial average directly without any non-linear transform, like modulus. The result of this value should be zero. But first, we have to prove that spatial average operation commutes with convolution operators. The proof is given as:

$$\begin{aligned}
 \langle(\kappa \star \psi)(x)\rangle &= \int (\kappa \star \psi)(x) dx \\
 &= \int \left(\int \kappa(y) \psi(x-y) dy\right) dx \\
 &= \int \left(\int \kappa(x-u) \psi(u) du\right) dx \\
 &= \int \psi(u) \left(\int \kappa(x-u) dx\right) du \\
 &= \int \psi(u) \cdot 0 du \\
 &= 0 .
 \end{aligned} \tag{C.14}$$

Thus, a non-linear operation is needed to construct the estimators for scattering transform. And we chose the modulus operation here. Intuitively, it's very easy to see that modulus and spatial average operation are both translation invariant. Thus, the scattering coefficients are translation invariant.

Appendix D

Kaiser-Squire inversion

In this appendix, we are going to explain why the shear field is a spin 2 field and derive the Kaiser-Squire inversion in a more detailed way.

D.1 Spin 2 fields

A spin- s field is a field that transforms by a phase factor $e^{is\alpha}$ under a local rotation of the coordinate system by an angle α . If a field has spin weight s , then under the local rotation of the coordinate system by an angle α , the field transforms as

$$f' = e^{is\alpha} f , \tag{D.1}$$

which means that:

- A spin-0 field ($s = 0$) returns to the same value after any rotation.
- A spin-1 field ($s = 1$) returns to the same value after a 2π rotation.
- A spin-2 field ($s = 2$) returns to the same value after a π rotation.

To prove that the shear field is a spin-2 field, we start with the local Cartesian coordinates system (x, y) in a small patch of the sky. The shear components as defined as:

$$\gamma_1 = \frac{1}{2} \left(\frac{\partial^2}{\partial^2 x} - \frac{\partial^2}{\partial^2 y} \right) \psi , \tag{D.2}$$

$$\gamma_2 = \frac{\partial^2}{\partial x \partial y} \psi . \tag{D.3}$$

To see how shear fields transform under a rotation, we consider a rotation of the coordinate system by an angle α :

$$x' = x \cos \alpha - y \sin \alpha , \tag{D.4}$$

$$y' = x \sin \alpha + y \cos \alpha . \tag{D.5}$$

Then the first order derivatives of ψ with respect to the new coordinates are:

$$\begin{aligned}\frac{\partial}{\partial x'} &= \frac{\partial x}{\partial x'} \frac{\partial}{\partial x} + \frac{\partial y}{\partial x'} \frac{\partial}{\partial y} = \cos \alpha \frac{\partial}{\partial x} - \sin \alpha \frac{\partial}{\partial y} , \\ \frac{\partial}{\partial y'} &= \frac{\partial x}{\partial y'} \frac{\partial}{\partial x} + \frac{\partial y}{\partial y'} \frac{\partial}{\partial y} = \sin \alpha \frac{\partial}{\partial x} + \cos \alpha \frac{\partial}{\partial y} .\end{aligned}\tag{D.6}$$

And the second order derivatives are:

$$\begin{aligned}\frac{\partial^2}{\partial x'^2} &= \cos^2 \alpha \frac{\partial^2}{\partial x^2} - 2 \sin \alpha \cos \alpha \frac{\partial^2}{\partial x \partial y} + \sin^2 \alpha \frac{\partial^2}{\partial y^2} , \\ \frac{\partial^2}{\partial y'^2} &= \sin^2 \alpha \frac{\partial^2}{\partial x^2} + 2 \sin \alpha \cos \alpha \frac{\partial^2}{\partial x \partial y} + \cos^2 \alpha \frac{\partial^2}{\partial y^2} .\end{aligned}\tag{D.7}$$

Also, for the second order mixed derivatives:

$$\frac{\partial^2}{\partial x' \partial y'} = \cos \alpha \sin \alpha \frac{\partial^2}{\partial x^2} + (\cos^2 \alpha - \sin^2 \alpha) \frac{\partial^2}{\partial x \partial y} - \cos \alpha \sin \alpha \frac{\partial^2}{\partial y^2} .\tag{D.8}$$

With these transformations, we can rewrite the shear components in the new coordinates:

$$\begin{aligned}\gamma_1 &= \frac{1}{2} \left(\frac{\partial^2}{\partial x'^2} - \frac{\partial^2}{\partial y'^2} \right) \psi = -\sin 2\alpha \frac{\partial^2}{\partial x \partial y} \psi , \\ \gamma_2 &= \frac{\partial^2 \psi}{\partial x' \partial y'} = \frac{1}{2} \left(\sin 2\alpha \left(\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right) + \cos 2\alpha \frac{\partial^2}{\partial x \partial y} \right) ,\end{aligned}\tag{D.9}$$

which shows that the shear field goes back to its original state after rotating by π and thus is a spin-2 field.

D.2 Flat sky Kaiser-Squire inversion

In the flat sky limit, the shear field can be written in terms of the Fourier transform of the potential field ψ as follows:

$$\tilde{\gamma}_1(\ell_1, \ell_2) = \frac{1}{2} (-\ell_1^2 + \ell_2^2) \tilde{\psi}(\ell_1, \ell_2) ,\tag{D.10}$$

$$\tilde{\gamma}_2 = -\ell_1 \ell_2 \tilde{\psi}(\ell_1, \ell_2) ,\tag{D.11}$$

whereas the convergence field in Fourier space is:

$$\tilde{\kappa}(\ell_1, \ell_2) = -\frac{1}{2} (\ell_1^2 + \ell_2^2) \tilde{\psi}(\ell_1, \ell_2) .\tag{D.12}$$

Thus, the shear field can be written in terms of the convergence field as:

$$\begin{aligned}\tilde{\gamma}(\boldsymbol{\ell}) &= \tilde{\gamma}_1(\boldsymbol{\ell}) + i\tilde{\gamma}_2(\boldsymbol{\ell}) \\ &= \frac{\ell_1^2 - \ell_2^2}{\ell_1^2 + \ell_2^2} \tilde{\kappa} + i \frac{2\ell_1 \ell_2}{\ell_1^2 + \ell_2^2} \tilde{\kappa} \\ &= \left(\frac{\ell_1 + i\ell_2}{\ell} \right)^2 \tilde{\kappa}(\boldsymbol{\ell}) ,\end{aligned}\tag{D.13}$$

where $\ell = \sqrt{\ell_1^2 + \ell_2^2}$.

D.3 Spin weighted spherical harmonics

Spin-weight spherical harmonics ${}_s Y_{\ell,m}$ are a generalization of the standard spherical harmonics $Y_{\ell,m}$ and are used to describe fields with intrinsic spin on the sphere. For example, spin-0 spherical harmonics, just the usual spherical harmonics, could be used to describe the scalar field, like convergence field. Spin-1 spherical harmonics are able to depict vector fields, like electric and magnetic fields. Spin-2 spherical harmonics are used to describe the tensor field, like CMB, gravitational wave and shear field.

Spin-weighted spherical harmonics also have the orthogonality property:

$$\int_{S^2} {}_s Y_{\ell,m,s'} \bar{{}_s Y_{\ell',m'}} d\Omega = \delta_{ss'} \delta_{\ell\ell'} \delta_{mm'} , \quad (\text{D.14})$$

where $d\Omega = \sin\theta d\theta d\phi$ is the solid angle element on the sphere and δ here represents Kronecker delta function. This property enables us to expand the fields on the sphere in terms of the spin-weighted spherical harmonics like normal spherical harmonics.

The spin could be raised or lowered by the following two operators:

Spin-raising operator:

$$\bar{\partial} ({}_s Y_{\ell,m}) = +\sqrt{(\ell-s)(\ell+s+1)} {}_{s+1} Y_{\ell,m} . \quad (\text{D.15})$$

Spin-lowering operator:

$$\bar{\partial} ({}_s Y_{\ell,m}) = -\sqrt{(\ell+s)(\ell-s+1)} {}_{s-1} Y_{\ell,m} . \quad (\text{D.16})$$

D.4 Full sky Kaiser-Squire inversion

With the equation mentioned above, Eq. (D.15) and (D.16), we could rewrite the convergence field in terms of the raising and lowering operators as [93] :

$$\kappa = \frac{1}{4} (\bar{\partial}\bar{\partial} + \bar{\partial}\bar{\partial}) \psi . \quad (\text{D.17})$$

Also, the expression of shear field under these operators is:

$$\gamma = \gamma_1 + i\gamma_2 = \frac{1}{2} \bar{\partial}\bar{\partial}\psi . \quad (\text{D.18})$$

Expanding the projected potential field, spin-0 convergence field, as well as the spin-2 complex shear field in terms of the spin-weight spherical harmonics, ${}_0 Y_{\ell,m}$ and ${}_2 Y_{\ell,m}$, leads to the following expressions:

$$\psi(\theta, \phi) = \sum_{\ell,m} \tilde{\psi}_{\ell,m} {}_0 Y_{\ell,m}(\theta, \phi) , \quad (\text{D.19})$$

$$\kappa = \kappa_E + i\kappa_B = \sum_{\ell,m} (\tilde{\kappa}_{E,\ell,m} + i\tilde{\kappa}_{B,\ell,m}) {}_0 Y_{\ell,m} , \quad (\text{D.20})$$

$$\gamma = \gamma_1 + i\gamma_2 = \sum_{\ell,m} (\tilde{\gamma}_{E,\ell,m} + i\tilde{\gamma}_{B,\ell,m}) {}_2 Y_{\ell,m} . \quad (\text{D.21})$$

Taking the raising operator and lowering operator into Eq. (D.17) and (D.18), we can expand κ and γ field using spin-0 spherical harmonics as:

$$\begin{aligned}
 \kappa &= \frac{1}{4} (\bar{\partial}\bar{\partial} + \partial\partial) \psi \\
 &= \frac{1}{4} (\bar{\partial}\bar{\partial} + \partial\partial) \sum_{\ell,m} \tilde{\psi}_{\ell,m} {}_0Y_{\ell,m} \\
 &= \frac{1}{4} \sum_{\ell,m} \tilde{\psi}_{\ell,m} (\bar{\partial}\bar{\partial} + \partial\partial) {}_0Y_{\ell,m} \\
 &= \frac{1}{4} \sum_{\ell,m} \tilde{\psi}_{\ell,m} (-\ell(\ell+1) - (\ell+1)\ell) {}_0Y_{\ell,m} \\
 &= -\frac{1}{2} \sum_{\ell,m} \ell(\ell+1) \tilde{\psi}_{\ell,m} {}_0Y_{\ell,m} \\
 &= \sum_{\ell,m} (\tilde{\kappa}_{E,\ell,m} + i\tilde{\kappa}_{B,\ell,m}) {}_0Y_{\ell,m} ,
 \end{aligned} \tag{D.22}$$

$$\begin{aligned}
 \gamma &= \frac{1}{2} \bar{\partial}\partial\psi \\
 &= \frac{1}{2} \bar{\partial}\partial \sum_{\ell,m} \tilde{\psi}_{\ell,m} {}_0Y_{\ell,m} \\
 &= \frac{1}{2} \bar{\partial} \sum_{\ell,m} \tilde{\psi}_{\ell,m} \sqrt{(\ell-1)(\ell+2)} {}_1Y_{\ell,m} \\
 &= \frac{1}{2} \sum_{\ell,m} \tilde{\psi}_{\ell,m} \sqrt{(\ell-1)(\ell+2)} \sqrt{\ell(\ell+1)} {}_2Y_{\ell,m} \\
 &= \sum_{\ell,m} (\tilde{\gamma}_{E,\ell,m} + i\tilde{\gamma}_{B,\ell,m}) {}_2Y_{\ell,m} .
 \end{aligned} \tag{D.23}$$

Then, we can get the relation between convergence field, potential field and shear field as:

$$\kappa_{E,\ell,m} + i\kappa_{B,\ell,m} = -\frac{1}{2} \ell(\ell+1) \psi_{\ell,m} , \tag{D.24}$$

$$\gamma_{E,\ell,m} + i\gamma_{B,\ell,m} = \frac{1}{2} \sqrt{\ell(\ell-1)(\ell+1)(\ell+2)} \psi_{\ell,m} , \tag{D.25}$$

$$\kappa_{E,\ell,m} + i\kappa_{B,\ell,m} = -\sqrt{\frac{\ell(\ell+1)}{(\ell+2)(\ell-1)}} (\gamma_{E,\ell,m} + i\gamma_{B,\ell,m}) . \tag{D.26}$$

Eq. (D.26) is the harmonic space spherical-sky generalisation of the Kaiser-Squire (KS) inversion formula. An inversion spherical-harmonics transform on the full-sky maps of the convergence and shear fields. We have to mention here that when the convergence field comes from simulation usually without noise, then the convergence field is real, and the B-mode vanishes. So, we have $\kappa_{sim} = \kappa_{E,sim}$ and $\kappa_{B,sim} = 0$. However, this does not imply that the expansion coefficients are necessarily zero, i.e., $\kappa_{E,\ell,m} \neq 0$ and $\kappa_{B,\ell,m} \neq 0$,

since the spherical basis are complex functions. This also implies that $\gamma_{E,\ell,m}$, $\gamma_{B,\ell,m}$, γ_1 and γ_2 do not always vanish. In fact, γ_1 and γ_2 should be non-zero since both of them could be generated from the full-sky κ map.

Appendix E

Systematic effects

In this appendix, we will introduce some systematic effects and explain how they affect the observed shear fields.

E.1 Intrinsic alignment

Intrinsic alignment [67, 94, 95, 96] refers to the physical alignment of galaxy shapes due to local gravitational interactions, rather than distortions caused by weak gravitational lensing. This effect can introduce biases [69, 97, 98, 99, 100] in the measurement of cosmic shear, as it correlates galaxy shapes in a way that is independent of the lensing signal. Understanding and modeling intrinsic alignment is crucial for accurate cosmological parameter estimation. To introduce how intrinsic alignment affects the observed shear fields, let's first define the shape of galaxies and halos:

$$I_{ij}(\mathbf{x}_g) = \frac{1}{N} \int d\mathbf{r} \rho_g(\mathbf{r}) r_i r_j, \quad (\text{E.1})$$

where ρ_g refers to the mass density of galaxies at position \mathbf{x}_g (usually the center of the galaxy), and $\mathbf{r} \equiv \mathbf{x} - \mathbf{x}_g$, the normalization factor $N = \int d\mathbf{r} \rho_g(\mathbf{r})$. Because of the statistical isotropy, the shape of an object after the ensemble average must be spherically symmetric, implying that the off-diagonal elements of the moment of inertia tensor are zero and the eigenvalues must be equal in all directions:

$$\langle I_{11} \rangle = \langle I_{22} \rangle = \langle I_{33} \rangle = \frac{1}{3} \langle \text{Tr} I \rangle. \quad (\text{E.2})$$

Thus, the total moment of inertia tensor can be written as:

$$\langle I_{ij} \rangle = \frac{1}{3} \langle \text{Tr} I \rangle \delta_{ij}. \quad (\text{E.3})$$

Then we could define the fluctuations of shape, S_{ij} , as the deviation from the ensemble average. Just like the scalar density contrast $1 + \delta_g(\mathbf{x}) \equiv \frac{n_g(\mathbf{x})}{\langle n_g \rangle}$, the mathematical expression for this tensor fluctuation is:

$$\delta_{ij} + S_{ij}(\mathbf{x}) \equiv \frac{I_{ij}(\mathbf{x})}{\langle \text{Tr} I \rangle / 3}. \quad (\text{E.4})$$

By separating the trace and trace-free parts, the shape fluctuation tensor can be written as:

$$S_{ij}(\mathbf{x}) = \frac{\text{Tr}S(\mathbf{x})}{3}\delta_{ij} + S_{(ij)}(\mathbf{x}) , \quad (\text{E.5})$$

where δ_{ij} is Kronecker delta and this term depicts the scaling, and the second term $S_{(ij)}$ is the trace-free part of the shape fluctuation tensor, which describes the distortion. Similar to shear, we could also write the shape fluctuation tensor in terms of the second order derivative of the potential $\partial_i\partial_j\Phi(\mathbf{x})$. The linear theory, *linear alignment (LA) model*, assumes that the three-dimensional tensor S_{ij} is linearly related to the second derivative of the potential [69] (similar to the linear bias relation in galaxy clustering $\delta_g(\mathbf{x}) = b_1\delta(\mathbf{x})$):

$$S_{ij}(\mathbf{x}) = b_K K_{ij}(\mathbf{x}) , \quad (\text{E.6})$$

where K_{ij} is the rescaled, dimensionless tidal field tensor, and b_K is the linear alignment amplitude. The tidal field tensor is defined as:

$$\begin{aligned} K_{ij}(\mathbf{x}) &\equiv \left(\partial_i\partial_j - \frac{\delta_{ij}}{3}\nabla^2 \right) \frac{\Phi(\mathbf{x})}{4\pi G a^2 \bar{\rho}_m(a)} \\ &= \left(\frac{\partial_i\partial_j}{\nabla^2} - \frac{\delta_{ij}}{3} \right) \delta(\mathbf{x}) , \end{aligned} \quad (\text{E.7})$$

where we get the second line by using Poisson equation $\nabla^2\Phi = 4\pi G a^2 \bar{\rho}_m\delta$. In Fourier space, the tidal field is given by:

$$K_{ij}(\mathbf{k}) = \left(\hat{k}_i\hat{k}_j - \frac{\delta_{ij}}{3} \right) \delta(\mathbf{k}) , \quad (\text{E.8})$$

where \hat{k}_i and \hat{k}_j are the unit vectors in the direction of i and j . We consider a Fourier mode aligned with the z -axis, $\mathbf{k} = k\hat{z}$. Then the contribution of this mode to $S_{ij}(\mathbf{x})$ can be written as

$$S_{ij}(\mathbf{x}) \supset b_K \left(\hat{k}_i\hat{k}_j - \frac{\delta_{ij}}{3} \right) \delta(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} \Big|_{\mathbf{k}=k\hat{z}} = \begin{bmatrix} -\frac{1}{3} & 0 & 0 \\ 0 & -\frac{1}{3} & 0 \\ 0 & 0 & +\frac{2}{3} \end{bmatrix} b_K \delta(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} . \quad (\text{E.9})$$

Thus, the LA model predicts that a Fourier mode \mathbf{k} introduces a distortion pattern in the form of a plane wave with amplitude $b_K\delta(\mathbf{k})$, shown in Fig. E.1. Measurements from simulation and observed galaxies show that $b_K < 0$ [101]. The above discussion is based on three-dimensional patterns. In actual observations, we could only have projected two-dimensional shapes on the plane perpendicular to the line of sight. The two dimensional trace-free components of the projected shape tensor γ_{ij} can be obtained by applying the projection to the linear alignment model, yielding:

$$\gamma_1(\mathbf{x}) = \frac{b_K}{2} \frac{\partial_1^2 - \partial_2^2}{\nabla^2} \delta(\mathbf{x}) , \quad (\text{E.10})$$

$$\gamma_2(\mathbf{x}) = b_K \frac{\partial_1\partial_2}{\nabla^2} \delta(\mathbf{x}) . \quad (\text{E.11})$$

Or equivalently, in Fourier space:

$$\gamma_1(\mathbf{k}) = \frac{b_K}{2} \left(\frac{\hat{k}_1^2 - \hat{k}_2^2}{k^2} \right) \delta(\mathbf{k}) = \frac{b_K}{2} (1 - \mu^2) \cos 2\phi_{\mathbf{k}} \delta(\mathbf{k}) , \quad (\text{E.12})$$

$$\gamma_2(\mathbf{k}) = b_K \left(\frac{k_1 k_2}{k^2} \right) \delta(\mathbf{k}) = \frac{b_K}{2} (1 - \mu^2) \sin 2\phi_{\mathbf{k}} \delta(\mathbf{k}) , \quad (\text{E.13})$$

$$\pm 2\gamma(\mathbf{k}) \equiv \gamma_1(\mathbf{k}) \pm i\gamma_2(\mathbf{k}) = \frac{b_K}{2} (1 - \mu^2) e^{\pm 2i\phi_{\mathbf{k}}} \delta(\mathbf{k}) , \quad (\text{E.14})$$

where $\mu \equiv \hat{k} \cdot \hat{z} = \hat{k}_3$ and $\phi_{\mathbf{k}} \equiv \tan^{-1}(\hat{k}_2/\hat{k}_1)$. Thus, the intrinsic alignment is already anisotropic due to the projection effect. We define the E -mode and B -mode fields by rotating $\pm 2\gamma$ in Fourier space to cancel the phase part factor as:

$$E(\mathbf{k}) + iB(\mathbf{k}) \equiv_{\pm 2} \gamma(\mathbf{k}) , \quad (\text{E.15})$$

which leads to the expression of E -mode and B -mode:

$$E(\mathbf{k}) = \frac{b_K}{2} (1 - \mu^2) \delta(\mathbf{k}) , \quad (\text{E.16})$$

$$B(\mathbf{k}) = 0 . \quad (\text{E.17})$$

Thus, calculating the power spectra of E -mode and galaxy number density contrast with linear bias, $\delta_g(\mathbf{x}) = b_1 \delta(\mathbf{x})$, we have:

$$P_{gE}(k, \mu) = b_1 \frac{b_K}{2} (1 - \mu^2) P_{\text{lin}}(k) , \quad (\text{E.18})$$

$$P_{EE}(k, \mu) = \frac{b_K^2}{4} (1 - \mu^2) P_{\text{lin}}(k) . \quad (\text{E.19})$$

Ref.[68] proposed an empirical model based on the linear alignment where the linear matter power spectrum appear in Eq. (E.18) and (E.19) is replaced by the non-linear matter power spectrum, $P_{\text{lin}}(k) \rightarrow P_{\text{NL}}(k)$. And this model is called the *non-linear alignment (NLA)* model. The NLA model is widely used in the literature to model the intrinsic alignment effect.

Since the IA correlation of galaxies mimics the part of shear correlation, so if we assume the observed shape of galaxies is purely due to the gravitational lensing effect, the constraints on cosmological parameters will be biased. Rather, the observed galaxy shape should include both weak gravitational lensing (G) and intrinsic alignment (I) effect:

$$\gamma^{\text{obs}} = \gamma^{\text{G}} + \gamma^{\text{I}} , \quad (\text{E.20})$$

where we assumed that both of the distortions are very small [102]. Then the angular power spectrum of the observed galaxy includes three terms:

$$C_{\ell}^{\text{obs}} = C_{\ell}^{\text{GG}} + C_{\ell}^{\text{II}} + C_{\ell}^{\text{GI}} . \quad (\text{E.21})$$

Fig. E.2 illustrates how the weak lensing and IA effects are correlated.

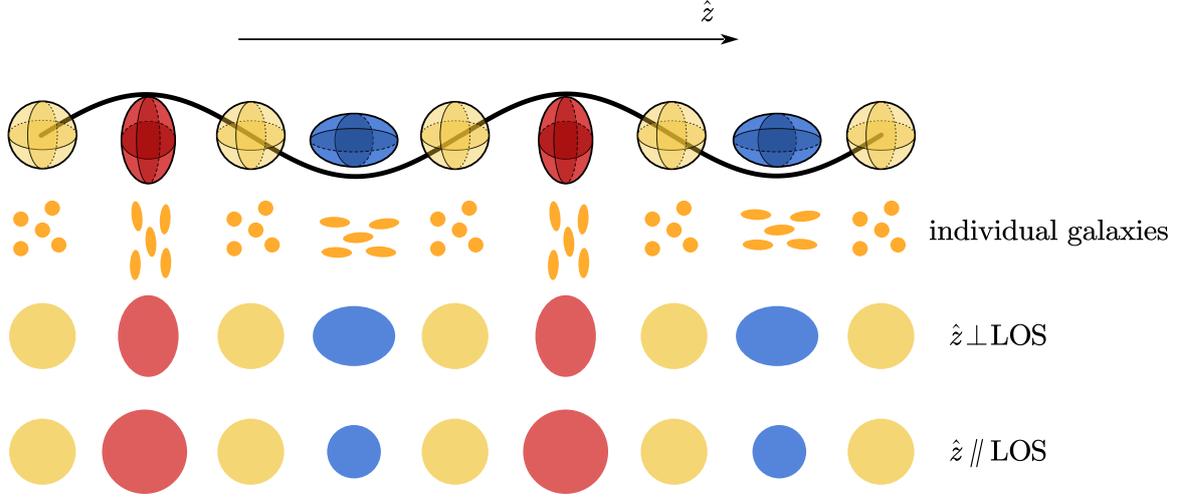


Figure E.1: The IA model predicts a distortion pattern in galaxy shapes. The black curve illustrates large-scale Fourier modes along the z -axis (horizontal in the figure) derived from the density field $\delta(\mathbf{x})$. The ellipsoids (blue and red) and the sphere (yellow) depict the distortion patterns induced by this mode based on the LA model with $b_K < 0$ [101]. The first row shows statistical predictions, indicating that while individual galaxies (orange shapes in the second row) exhibit scattered shapes, the large-scale pattern is statistically aligned. If the \hat{z} direction is perpendicular to the line of sight, the observed distortion pattern manifests as tangential stretching and squeezing, revealing the distortion in galaxy shapes (third row). Conversely, if the \hat{z} direction aligns with the line of sight, the observed (projected) distortion pattern appears as radial scaling effect (fourth row).

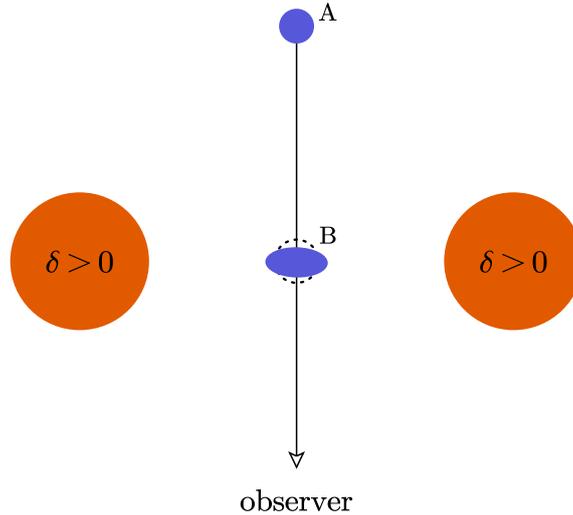


Figure E.2: Illustration of weak lensing and intrinsic alignment effects. Purple color (A and B) represents the observed galaxies. A distant galaxy (A) is lensed by a foreground mass distribution, shown as orange circles. The intrinsic alignment effect (B) is due to the local gravitational interaction of galaxies, which introduces IA effect. The GI correlation corresponds to the correlation between A and B.

E.2 Shape noise

In this section, we are going to introduce the shape noise from shear and convergence maps. Since the shear field and convergence field are connected through Kaiser-Squire Inversion, we can start from the shape noise from shear fields. Because of the Central Limit Theorem, the shape noise from the observed shear field is Gaussian distributed, and can be expressed as:

$$N = N_1 + iN_2 , \quad (\text{E.22})$$

where N_1 and N_2 are both Gaussian random variables with zero mean and the same standard deviation for the real part and imaginary part of the shear field. The standard deviation of the shape noise is given by:

$$\sigma_x = \frac{\sigma_e}{\sqrt{A \cdot n_{eff}}} , \quad (\text{E.23})$$

where σ_x represents the value in position space (where we could express it to be σ_k in Fourier space), σ_e is the shape noise parameter and n_{eff} is the galaxy effective number density.

Based on Kaiser-Squire Inversion under flat sky approximation in Eq. (3.39), we could express it in position space as:

$$\kappa = \nabla^{-2} \partial^* \partial^* \gamma , \quad (\text{E.24})$$

where $\partial = \partial_1 + i\partial_2$ and then $\partial^* = \partial_1 - i\partial_2$, and $\nabla^{-2} = \frac{1}{\partial^* \partial}$ is the inverse Laplacian operation. By expanding this equation and the shear field, we could have:

$$\begin{aligned} \kappa &= \nabla^{-2} \partial^* \partial^* \gamma \\ &= \nabla^{-2} (\partial_1 - i\partial_2) (\partial_1 - i\partial_2) (\gamma_1 + i\gamma_2) \\ &= \nabla^{-2} [(\partial_1 \partial_1 - \partial_2 \partial_2) + 2i\partial_1 \partial_2] (\gamma_1 + i\gamma_2) . \end{aligned} \quad (\text{E.25})$$

Since κ field is a real-valued field, we need to ignore the imaginary part, which leads to:

$$\kappa = \nabla^{-2} [(\partial_1 \partial_1 - \partial_2 \partial_2) \gamma_1 - 2\partial_1 \partial_2 \gamma_2] . \quad (\text{E.26})$$

This equation is the general expression for the correlation between convergence and shear fields. For the real observed data, we should include the noise term mentioned in Eq. (E.23) to the shear:

$$\gamma_{\text{obs}} = (\gamma_{1,\text{true}} + N_1) + i(\gamma_{2,\text{true}} + N_2) , \quad (\text{E.27})$$

where the subscript obs refers to the observed shear field, true refers to the true shear field and N is the shape noise from $\mathcal{N} \sim (0, \sigma_x)$. By taking Fourier transform on both sides of Eq. (E.26), we have:

$$\tilde{\kappa}_{\text{obs}} = \frac{1}{\ell^2} \left[(\ell_1^2 - \ell_2^2) (\tilde{\gamma}_1 + \tilde{N}_1) - 2\ell_1 \ell_2 (\tilde{\gamma}_2 + \tilde{N}_2) \right] , \quad (\text{E.28})$$

where the $\tilde{\cdot}$ represents the Fourier transform of the field. Since the shape noise is Gaussian distributed, the Fourier transform of the shape noise is also Gaussian distributed. And the relation of the std of the shape noise in position space and Fourier space is:

$$\sigma_{\tilde{x}} \cdot \sigma_k = \frac{1}{2\pi} , \quad (\text{E.29})$$

where σ_k is the std of the shape noise in Fourier space. Here we care only about how the shape noise contributes to the real convergence field so that we could add this directly to the simulation convergence field. With the property that the summation of two Gaussian random variables is also a Gaussian random variable, we could calculate the variance of the Gaussian field on convergence maps:

$$\begin{aligned} \tilde{\sigma}_{\kappa,k}^2 &= \left(\frac{(\ell_1^2 - \ell_2^2)}{\ell^2} \right)^2 \tilde{\sigma}_{1,k}^2 + \left(\frac{2\ell_1\ell_2}{\ell^2} \right)^2 \tilde{\sigma}_{2,k}^2 \\ &= \frac{(\ell_1^2 + \ell_2^2)^2}{\ell^4} \tilde{\sigma}_k = \tilde{\sigma}_k , \end{aligned} \quad (\text{E.30})$$

which is exactly the same as the shape noise in the shear field. Thus, we could conclude that the shape noise in convergence maps is also Gaussian distributed with the same standard deviation as the shear field.

Appendix F

Fibonacci sphere sampling

Fibonacci sphere is a method [74, 75] to generate evenly distributed samples on the sphere. This method is based on the Fibonacci sequence and the golden ratio, and we use this irrational number to generate angular steps, ensuring quasi-uniform and non-repetitive coverage of the sphere. The azimuthal angle is sampled using the following equation:

$$\phi = \left(\frac{2\pi i}{g} \right) \bmod 2\pi , \quad (\text{F.1})$$

where mod is the modulo operation, $g = \frac{1+\sqrt{5}}{2}$ is the golden ratio, and i is the index of the sample, which ranges from 0 to $N - 1$ if we want to sample N points. The modulo operation ensures that all the values of ϕ are between 0 and 2π . The use of the golden angle, an irrational number, makes sure that the sampled points are not the same in which the case of using integers. By using more samples, the azimuthal angle will be more evenly distributed. The polar angle θ is sampled using the following equation:

$$\theta = \arccos \left(1 - \frac{2(i + 0.5)}{n} \right) , \quad (\text{F.2})$$

where N is the number of samples. The polar angle θ is sampled in such a way that the points are evenly distributed in the vertical direction, from north to south pole. Fig. F.1 shows an example of the Fibonacci sphere sampling method. These samplings are not strictly uniform, but are quasi-uniform and non-repetitive.

Using the evenly distributed points on the sphere, we can extract square maps from the full-sky maps. The method employed here is the *Gnomonic Projection* [76], an azimuthal map projection that maps the surface of a sphere onto a tangent plane from a single point at the sphere's center. The centers of these projections correspond to the points sampled from the Fibonacci sphere. A key property of this projection is that all great circles on the sphere are represented as straight lines on the plane. This is analogous to placing a flat sheet tangent to a globe and projecting the surface onto the sheet from the globe's center. This type of projection is valid only for points within a 90° angular distance from the center, effectively capturing a hemispherical region without introducing distortion singularities.

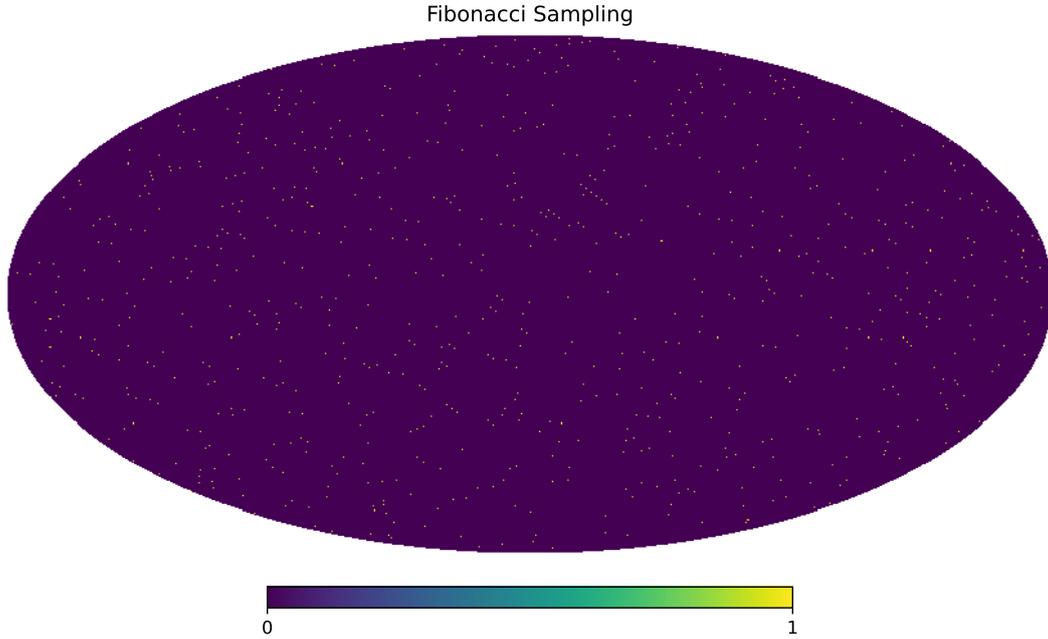


Figure F.1: Illustration of the Fibonacci sphere sampling method. In order to deliver a better illustration, 10000 points are chosen on a sphere with $NSIDE = 512$, this number is much smaller in practice. The yellow dots represent the sampled points on the sphere, which is approximately uniform distribution.

The purpose of sampling points on the sphere is to ensure that the projection centers are sufficiently spaced apart. This avoids overlap and ensures efficient use of the data while maximizing map coverage. The size of the square maps is determined by the number of pixels to be sampled, and the rotation angle of the squares is carefully chosen to prevent overlapping regions.

Bibliography

- [1] Alan H Guth. Inflationary universe: A possible solution to the horizon and flatness problems. *Physical Review D*, 23(2):347, 1981.
- [2] Andrei D Linde. The inflationary universe. *Reports on Progress in Physics*, 47(8):925, 1984.
- [3] Gary Steigman. Primordial nucleosynthesis in the precision cosmology era. *Annu. Rev. Nucl. Part. Sci.*, 57(1):463–491, 2007.
- [4] Adam G Riess, Alexei V Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M Garnavich, Ron L Gilliland, Craig J Hogan, Saurabh Jha, Robert P Kirshner, et al. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *The astronomical journal*, 116(3):1009, 1998.
- [5] Saul Perlmutter. Supernovae, dark energy, and the accelerating universe: The status of the cosmological parameters. In *Proceedings of the XIX International Symposium on Lepton and Photon Interactions at High Energies. Stanford, California*, 1999.
- [6] Pierre Astier and Reynald Pain. Observational evidence of the accelerated expansion of the universe. *Comptes Rendus Physique*, 13(6-7):521–538, 2012.
- [7] Balakrishna S Haridasu, Vladimir V Luković, Rocco D’Agostino, and Nicola Vittorio. Strong evidence for an accelerating universe. *Astronomy & Astrophysics*, 600:L1, 2017.
- [8] Saul Perlmutter, Goldhaber Aldering, Gerson Goldhaber, Richard A Knop, Peter Nugent, Patricia G Castro, Susana Deustua, Sebastien Fabbro, Ariel Goobar, Donald E Groom, et al. Measurements of ω and λ from 42 high-redshift supernovae. *The Astrophysical Journal*, 517(2):565, 1999.
- [9] Scott Dodelson and Fabian Schmidt. *Modern cosmology*. Academic Press, San Diego, CA, 2 edition, July 2020.
- [10] Wikipedia. Shape of the universe — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Shape%20of%20the%20universe&oldid=1241516259>, 2024. [Online; accessed 12-September-2024].
- [11] J. A. Peacock. *Cosmological physics*. 1999.

- [12] F Zwicky. Die rotverschiebung von extragalaktischen nebeln. *Helv. Phys. Acta*, 6:110–127, 1933.
- [13] Timothy MC Abbott, Michel Aguena, Alex Alarcon, S Allam, O Alves, A Amon, F Andrade-Oliveira, James Annis, S Avila, D Bacon, et al. Dark energy survey year 3 results: Cosmological constraints from galaxy clustering and weak lensing. *Physical Review D*, 105(2):023520, 2022.
- [14] Nabila Aghanim, Yashar Akrami, Mark Ashdown, Jonathan Aumont, Carlo Bacigalupi, Mario Ballardini, Anthony J Banday, RB Barreiro, Nicola Bartolo, S Basak, et al. Planck 2018 results-vi. cosmological parameters. *Astronomy & Astrophysics*, 641:A6, 2020.
- [15] Michel Chevallier and David Polarski. Accelerating universes with scaling dark matter. *International Journal of Modern Physics D*, 10(02):213–223, 2001.
- [16] George B Arfken and Hans J Weber. Mathematical methods for physicists 6th ed. *Mathematical methods for physicists 6th ed. by George B. Arfken and Hans J. Weber. Published: Amsterdam; Boston: Elsevier*, 2005.
- [17] Ramesh Narayan and Matthias Bartelmann. Lectures on gravitational lensing. *arXiv preprint astro-ph/9606001*, 1996.
- [18] Peter Schneider, Jürgen Ehlers, Emilio E Falco, Peter Schneider, Jürgen Ehlers, and Emilio E Falco. *Gravitational lenses as astrophysical tools*. Springer, 1992.
- [19] Martin Kilbinger. Cosmology with cosmic shear observations: a review. *Reports on Progress in Physics*, 78(8):086901, 2015.
- [20] Matthias Bartelmann and Peter Schneider. Weak gravitational lensing. *Physics Reports*, 340(4-5):291–472, 2001.
- [21] Carolin Seitz and Peter Schneider. Steps towards nonlinear cluster inversion through gravitational distortions: Iii. including a redshift distribution of the sources. *arXiv preprint astro-ph/9601079*, 1996.
- [22] Nick Kaiser and Gordon Squires. Mapping the dark matter with weak gravitational lensing. *Astrophysical Journal, Part 1 (ISSN 0004-637X)*, vol. 404, no. 2, p. 441-450., 404:441–450, 1993.
- [23] Peter Schneider, Christopher Kochanek, and Joachim Wambsganss. *Gravitational lensing: strong, weak and micro: Saas-Fee advanced course 33*, volume 33. Springer Science & Business Media, 2006.
- [24] Sihao Cheng, Yuan-Sen Ting, Brice Ménard, and Joan Bruna. A new approach to observational cosmology using the scattering transform. *Monthly Notices of the Royal Astronomical Society*, 499(4):5902–5914, 2020.
- [25] Sihao Cheng and Brice Ménard. How to quantify fields or textures? a guide to the scattering transform. *arXiv preprint arXiv:2112.01288*, 2021.

- [26] Aparna Rawat and Abhishek Singh. Mexican hat wavelet transform of generalized functions in g spaces. *Proceedings-Mathematical Sciences*, 131(2):31, 2021.
- [27] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, et al. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020.
- [28] Sihao Cheng, Rudy Morel, Erwan Allys, Brice Ménard, and Stéphane Mallat. Scattering spectra models for physics. *PNAS nexus*, 3(4):pgae103, 2024.
- [29] Maximilian Pichler and Florian Hartig. Machine learning and deep learning—a review for ecologists. *Methods in Ecology and Evolution*, 14(4):994–1016, 2023.
- [30] Favio Vázquez. Deep learning made easy with deep cognition. <https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351>, Jan 2018.
- [31] Arnab Mukhrjee. Ai part 3 regression vs classification models. <https://www.linkedin.com/pulse/aipart3regression-vs-classification-models-arnab-mukherjee/>, Mar 2023.
- [32] Abiodun M Ikotun, Absalom E Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023.
- [33] Olga Bolgurtseva Inna Logunova. Means clustering algorithm in ml. <https://serokell.io/blog/k-means-clustering-in-machine-learning>, Jan 2023.
- [34] Anup Bhande. What is underfitting and overfitting in machine learning and how to deal with it. <https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it>, Mar 2018.
- [35] Zach BobbittHey. What is the bias-variance tradeoff in machine learning? <https://www.statology.org/bias-variance-tradeoff/>, Nov 2020.
- [36] Wikipedia. Verzerrung-Varianz-Dilemma — Wikipedia, the free encyclopedia. <http://de.wikipedia.org/w/index.php?title=Verzerrung-Varianz-Dilemma&oldid=236879265>, 2024. [Online; accessed 23-December-2024].
- [37] Andrew Ng. Cs229 lecture notes - supervised learning. 2012.
- [38] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

- [39] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [40] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [41] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.
- [42] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [43] Geoffrey Hinton. Lecture 6e – rmsprop: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning, 2012. Available online: <https://www.coursera.org/lecture/neural-networks/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude-QjdUd>.
- [44] GE Hinton. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [45] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [46] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002.
- [47] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [48] Tomasz Kacprzak, Janis Fluri, Aurel Schneider, Alexandre Refregier, and Joachim Stadel. Cosmogridv1: a simulated Λ cdm theory prediction for map-level cosmological inference. *Journal of Cosmology and Astroparticle Physics*, 2023(02):050, 2023.
- [49] Silvan Fischbacher, Tomasz Kacprzak, Jonathan Blazek, and Alexandre Refregier. Redshift requirements for cosmic shear with intrinsic alignment. *Journal of Cosmology and Astroparticle Physics*, 2023(01):033, 2023.
- [50] Alexandra Amon, Daniel Gruen, Michael A Troxel, Niall MacCrann, Scott Dodelson, Ami Choi, Cyrille Doux, Lucas F Secco, Simon Samuroff, Elisabeth Krause, et al. Dark energy survey year 3 results: Cosmology from cosmic shear and robustness to data calibration. *Physical Review D*, 105(2):023514, 2022.
- [51] Lucas Frozza Secco, Simon Samuroff, Elisabeth Krause, Bhuvnesh Jain, J Blazek, Marco Raveri, Andresa Campos, Alexandra Amon, Anqi Chen, Cyrille Doux, et al. Dark energy survey year 3 results: Cosmology from cosmic shear and robustness to modeling uncertainty. *Physical Review D*, 105(2):023515, 2022.

- [52] P Fosalba, E Gaztañaga, FJ Castander, and M Crocce. The mice grand challenge light-cone simulation–iii. galaxy lensing mocks from all-sky lensing maps. *Monthly Notices of the Royal Astronomical Society*, 447(2):1319–1332, 2015.
- [53] Pablo Fosalba, Enrique Gaztanaga, Francisco J Castander, and Marc Manera. The onion universe: all sky lightcone simulations in spherical shells. *Monthly Notices of the Royal Astronomical Society*, 391(1):435–446, 2008.
- [54] Andrea Petri, Zoltán Haiman, and Morgan May. Validity of the born approximation for beyond gaussian weak lensing observables. *Physical Review D*, 95(12):123503, 2017.
- [55] Chi Tian, Matthew F Carney, James B Mertens, and Glenn Starkman. Accurate relativistic observables from postprocessing light cone catalogs. *Physical Review D*, 105(6):063511, 2022.
- [56] Raphaël Sgier, Janis Fluri, Jörg Herbel, Alexandre Réfrégier, Adam Amara, Tomasz Kacprzak, and Andrina Nicola. Fast lightcones for combined cosmological probes. *Journal of Cosmology and Astroparticle Physics*, 2021(02):047, 2021.
- [57] Carmelita Carbone, Volker Springel, Carlo Baccigalupi, Matthias Bartelmann, and Sabino Matarrese. Full-sky maps for gravitational lensing of the cosmic microwave background. *Monthly Notices of the Royal Astronomical Society*, 388(4):1618–1626, 2008.
- [58] Volker Springel, Martin White, and Lars Hernquist. Hydrodynamic simulations of the sunyaev-zeldovicheffect (s). *The Astrophysical Journal*, 549(2):681, 2001.
- [59] RJ Sgier, Alexandre Réfrégier, Adam Amara, and Andrina Nicola. Fast generation of covariance matrices for weak lensing. *Journal of Cosmology and Astroparticle Physics*, 2019(01):044, 2019.
- [60] Janis Fluri, Tomasz Kacprzak, Aurelien Lucchi, Alexandre Refregier, Adam Amara, Thomas Hofmann, and Aurel Schneider. Cosmological constraints with deep learning from kids-450 weak lensing maps. *Physical Review D*, 100(6):063514, 2019.
- [61] Tomasz Kacprzak and Janis Fluri. Deeplss: Breaking parameter degeneracies in large-scale structure with deep-learning analysis of combined probes. *Physical Review X*, 12(3):031029, 2022.
- [62] Alexandre Barthelemy, Sandrine Codis, Cora Uhlemann, Francis Bernardeau, and Raphael Gavazzi. A nulling strategy for modelling lensing convergence in cones with large deviation theory. *Monthly Notices of the Royal Astronomical Society*, 492(3):3420–3439, 2020.
- [63] Andrea Zonca, Leo Singer, Daniel Lenz, Martin Reinecke, Cyrille Rosset, Eric Hivon, and Krzysztof Gorski. healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in python. *Journal of Open Source Software*, 4(35):1298, March 2019.

- [64] Nora Elisa Chisari, Alexander J Mead, Shahab Joudaki, Pedro Ferreira, Aurel Schneider, Joseph Mohr, Tilman Tröster, David Alonso, Ian G McCarthy, Sergio Martin-Alvarez, et al. Modelling baryonic feedback for survey cosmology. *arXiv preprint arXiv:1905.06082*, 2019.
- [65] Aurel Schneider, Romain Teyssier, Joachim Stadel, Nora Elisa Chisari, Amandine MC Le Brun, Adam Amara, and Alexandre Refregier. Quantifying baryon effects on the matter power spectrum and the weak lensing shear correlation. *Journal of Cosmology and Astroparticle Physics*, 2019(03):020, 2019.
- [66] Julio F Navarro. The structure of cold dark matter halos. In *Symposium-international astronomical union*, volume 171, pages 255–258. Cambridge University Press, 1996.
- [67] Paolo Catelan, Marc Kamionkowski, and Roger D Blandford. Intrinsic and extrinsic galaxy alignment. *Monthly Notices of the Royal Astronomical Society*, 320(1):L7–L13, 2001.
- [68] Sarah Bridle and Lindsay King. Dark energy constraints from cosmic shear power spectra: impact of intrinsic alignments on photometric redshift requirements. *New Journal of Physics*, 9(12):444, 2007.
- [69] Christopher M Hirata and Uroš Seljak. Intrinsic alignment-lensing interference as a contaminant of cosmic shear. *Physical Review D—Particles, Fields, Gravitation, and Cosmology*, 70(6):063526, 2004.
- [70] Benjamin Joachimi, Rachel Mandelbaum, FB Abdalla, and SL Bridle. Constraints on intrinsic alignment contamination of weak lensing surveys using the megaz-*lrg* sample. *Astronomy & Astrophysics*, 527:A26, 2011.
- [71] Francis Bernardeau, Stéphane Colombi, Enrique Gaztañaga, and Román Scocimarro. Large-scale structure of the universe and cosmological perturbation theory. *Physics reports*, 367(1-3):1–248, 2002.
- [72] Vincent Desjacques, Donghui Jeong, and Fabian Schmidt. Large-scale galaxy bias. *Physics reports*, 733:1–193, 2018.
- [73] Jonathan A Blazek, Niall MacCrann, MA Troxel, and Xiao Fang. Beyond linear galaxy alignments. *Physical Review D*, 100(10):103506, 2019.
- [74] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical geosciences*, 42:49–64, 2010.
- [75] Helmut Vogel. A better way to construct the sunflower head. *Mathematical bio-sciences*, 44(3-4):179–189, 1979.
- [76] José Luis Amorós, Martin J. Buerger, and Marisa Canut de Amorós. 3. the gnomonic projection. In *The Laue Method*, pages 55–81. Academic Press, 1975.

- [77] Alan Heavens. Statistical techniques in cosmology. *arXiv preprint arXiv:0906.0664*, 2009.
- [78] Max Tegmark, Andy N Taylor, and Alan F Heavens. Karhunen-loeve eigenvalue problems in cosmology: How should we tackle large data sets? *The Astrophysical Journal*, 480(1):22, 1997.
- [79] Andy Taylor, Benjamin Joachimi, and Thomas Kitching. Putting the precision in precision cosmology: How accurate should your data covariance matrix be? *Monthly Notices of the Royal Astronomical Society*, 432(3):1928–1946, 2013.
- [80] Scott Dodelson and Michael D Schneider. The effect of covariance estimator error on cosmological parameter constraints. *Physical Review D—Particles, Fields, Gravitation, and Cosmology*, 88(6):063537, 2013.
- [81] Luca Amendola, Simone Fogli, Alejandro Guarnizo, Martin Kunz, and Adrian Vollmer. Model-independent constraints on the cosmological anisotropic stress. *Physical Review D*, 89(6):063538, 2014.
- [82] Alexandre Barreira, Elisabeth Krause, and Fabian Schmidt. Accurate cosmic shear errors: do we need ensembles of simulations? *Journal of Cosmology and Astroparticle Physics*, 2018(10):053, 2018.
- [83] Stéphanie Escoffier, M-C Cousinou, A Tilquin, A Pisani, A Aguichine, S de La Torre, A Ealet, W Gillard, and E Jullo. Jackknife resampling technique on mocks: an alternative method for covariance matrix estimation. *arXiv preprint arXiv:1606.00233*, 2016.
- [84] J Hartlap, Patrick Simon, and P Schneider. Why your model parameter confidences might be too optimistic. unbiased estimation of the inverse covariance matrix. *Astronomy & Astrophysics*, 464(1):399–404, 2007.
- [85] Martin Kilbinger, Catherine Heymans, Marika Asgari, Shahab Joudaki, Peter Schneider, Patrick Simon, Ludovic Van Waerbeke, Joachim Harnois-Déraps, Hendrik Hildebrandt, Fabian Köhlinger, et al. Precision calculations of the cosmic shear power spectrum projection. *Monthly Notices of the Royal Astronomical Society*, 472(2):2126–2141, 2017.
- [86] Kai Lehman, Nico Schuster, Luisa Lucie-Smith, Nico Hamaus, Christopher T Davies, and Klaus Dolag. Cosmological inference with cosmic voids and neural network emulators. *arXiv preprint arXiv:2502.05262*, 2025.
- [87] Zhengyangguang Gong, Anik Halder, Alexandre Barreira, Stella Seitz, and Oliver Friedrich. Cosmology from the integrated shear 3-point correlation function: simulated likelihood analyses with machine-learning emulators. *Journal of Cosmology and Astroparticle Physics*, 2023(07):040, 2023.
- [88] David W Hogg and Daniel Foreman-Mackey. Data analysis recipes: Using markov chain monte carlo. *The Astrophysical Journal Supplement Series*, 236(1):11, 2018.

- [89] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.
- [90] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.
- [91] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The mcmc hammer. *PASP*, 125:306–312, 2013.
- [92] Sihao Cheng, Gabriela A Marques, Daniela Grandón, Leander Thiele, Masato Shirasaki, Brice Ménard, and Jia Liu. Cosmological constraints from weak lensing scattering transform using hsc y1 data. *arXiv preprint arXiv:2404.16085*, 2024.
- [93] Alexandre Barthelemy, Anik Halder, Zhengyangguang Gong, and Cora Uhlemann. Making the leap. part i. modelling the reconstructed lensing convergence pdf from cosmic shear with survey masks and systematics. *Journal of Cosmology and Astroparticle Physics*, 2024(03):060, 2024.
- [94] Rupert AC Croft and Christopher A Metzler. Weak-lensing surveys and the intrinsic correlation of galaxy ellipticities. *The Astrophysical Journal*, 545(2):561, 2000.
- [95] Robert G Crittenden, Priyamvada Natarajan, Ue-Li Pen, and Tom Theuns. Discriminating weak lensing from intrinsic spin correlations using the curl-gradient decomposition. *The Astrophysical Journal*, 568(1):20, 2002.
- [96] YP Jing and Yasushi Suto. Triaxial modeling of halo density profiles with high-resolution n-body simulations. *The Astrophysical Journal*, 574(2):538, 2002.
- [97] Benjamin Joachimi, Marcello Cacciato, Thomas D Kitching, Adrienne Leonard, Rachel Mandelbaum, Björn Malte Schäfer, Cristóbal Sifón, Henk Hoekstra, Alina Kiessling, Donnacha Kirk, et al. Galaxy alignments: An overview. *Space Science Reviews*, 193:1–65, 2015.
- [98] Alina Kiessling, Marcello Cacciato, Benjamin Joachimi, Donnacha Kirk, Thomas D Kitching, Adrienne Leonard, Rachel Mandelbaum, Björn Malte Schäfer, Cristóbal Sifón, Michael L Brown, et al. Galaxy alignments: theory, modelling & simulations. *Space Science Reviews*, 193:67–136, 2015.
- [99] Donnacha Kirk, Michael L Brown, Henk Hoekstra, Benjamin Joachimi, Thomas D Kitching, Rachel Mandelbaum, Cristóbal Sifón, Marcello Cacciato, Ami Choi, Alina Kiessling, et al. Galaxy alignments: Observations and impact on cosmology. *Space Science Reviews*, 193:139–211, 2015.
- [100] MA Troxel and Mustapha Ishak. The intrinsic alignment of galaxies and its impact on weak gravitational lensing in an era of precision cosmology. *Physics Reports*, 558:1–59, 2015.
- [101] Toshiki Kurita. The intrinsic alignment of galaxies: The good, the bad, and the ugly. March 2025.

- [102] GM Bernstein and M Jarvis. Shapes and shears, stars and smears: optimal measurements for weak lensing. *The Astronomical Journal*, 123(2):583, 2002.

Acknowledgements

First, I would like to express my heartfelt gratitude to my supervisor Dr. Stella Seitz for your insightful guidance, constant support, and encouragement throughout this thesis project. Your expertise, feedback, and enthusiasm toward science during discussions always motivate me to do it better. I am especially grateful for your patience in our conversations—even when I struggled to clearly express my thoughts, you always listened attentively and helped me clarify and improve my ideas. I would be super thankful for letting me join Gravitational Lensing Group and be a part of OPINAS group—not just as research group, but as communities where I could learn, grow, and experiment freely without fear. I would also want to express my gratitude for Prof. Dr. Ralf Bender for the continued support.

I am deeply thankful for Zhengyangguang Gong (Laurence). Words cannot express my gratitude. You have always been the person I can rely on the most. I want to thank you for providing such an interesting idea for my master thesis and I benefited a lot from this project. I am super grateful for your guidance throughout this project, from which I learned not only knowledge related to this project, I gained insights in other interesting and important topics as well. I always enjoy the discussions during the lunch and dinner time when we came up with a lot of ideas. You are more like a brother than academic supervisor and we had a great time in the past two years. I hope we could have more interesting ideas that we could work together.

Thank you Dr. Anik Halder for inviting me to Gravitational Lensing Group. You are the first star illuminating my path toward cosmology. You are an inspiration to me. You showed me how interesting science is especially when people do it together and how to do science. You are not only the tutor of my cosmology, but also my life. You taught me how to be a good person, how to move forward when facing troubles and how to keep an open mind. Thank you so much, Anik.

I would like to sincerely thank David Gebauer for constant support whenever I encountered questions. I'm especially grateful for your support when I was practicing my talk—you listened, gave thoughtful feedback, and helped me improve at my very first workshop. You are always a good friend to me.

To Leon Ecker, thank you for always inspiring me to try new things and for helping me step out of my comfort zone—especially at times when I did not have the confidence to take the first step. You brought me into conversations when I hesitated, and your pres-

ence make those moments feel safe. Your support has always been a source of strength, especially during stressful times. I am also deeply grateful for all the helpful advice and information you shared during my PhD applications. I am very thankful for your company.

Thank you, Yunhe Wang, for sharing so many interesting courses and engaging discussions—they made the learning experience more enjoyable and meaningful. During times when I felt especially low, your willingness to reach out and remind me that I could talk anytime truly meant a lot. Your presence made many of my commutes to Garching feel far less lonely, and I am sincerely thankful for that.

Many thanks to Kai Lehmann, Luis Thomas, Raphael Zoeller, Dr. Dylan Britt, Dr. Michael Walther, Jan-Niklas Pippert, Jaydeep, Xu Han, Sarah Joswig, Dr. Shankar Srinivasan and Naman Pujari for making the Friday afternoon tea times such a memorable part of my experience. Thank you for all the fun conversations, the entertaining gossip, and the thought-provoking research discussions. Special thanks for making the 007 office such a unique place—a space filled with warmth, laughter, and a wonderfully relaxed atmosphere. It wouldn't have been the same without you.

I would like to sincerely thank Dr. Ariel Sanchez, who taught my first cosmology course and sparked my interest in the field. I'm also grateful to Prof. Jochen Weller, whose cosmology lectures further deepened my understanding. Many thanks to Prof. Daniel Gruen, Dr. Oliver Friedrich for the *From Data to Insight* course and exercise sessions—through them, I gained a solid foundation in statistical methods that continue to shape how I approach scientific problems.

Also, I want to thank my bachelor friends: Haojun Sun (TienChiao Tsao), Yu Hua, He Sun, Ziyu Li, Jiarong Guo, Zhu Tao, Junrei Ma, Shijie Yuan (Yvan Shih Chieh), Kai Chen, Yumo Zhao, Guangxian Li, Junkai Liu and Yiju Yang for the interesting stories and nice travels when I was back in China.

Finally, I would like to express my deepest gratitude to my parents. Thank you for your unwavering support and for always giving me the freedom to pursue my own path, no matter what choices I made. Your love, patience, and trust have been the foundation that carries me through every step of this journey.