

Chapter 1

Methodology

The main goal of a forecast is to figure out how well some experiment will be able to measure some set of parameters. In our case, the experiment will be the different probes of the *Euclid* mission while the parameters that we want to measure are cosmological parameters. For this, we generate some fake data that the experiment could have measured given an underlying fiducial model of the universe, free parameters that show up in our modelling of the data, and the different survey specifications.

The two main methods of forecasting we will deploy in this work are the Markov chain Monte Carlo (MCMC) method and the Fisher Information (FI) method. They are both based in the framework of Bayesian statistics as they are methods of obtaining the posterior distributions, i.e. the probability, $p(\boldsymbol{\theta}|\mathbf{D})$, of observing the model parameters, $\boldsymbol{\theta}$, given the data, \mathbf{D} . The posterior is really what we are after, from it we can extract the mean and the measurement uncertainties of the model parameters. We can group all of our model parameters into 3 groups, cosmological parameters, nuisance parameters, and fixed parameters. While cosmological parameters are really what we are after the nuisance parameters show up during modelling and represent uncertainties in the modelling of the observed data or the observation process itself. In the end, they are not of interest to us, so we often only look at the marginal posterior. That is the posterior when the different values of nuisance parameters have been integrated over, i.e.

$$p(\boldsymbol{\theta}_c|\mathbf{D}) = \int p(\boldsymbol{\theta}|\mathbf{D}) d\boldsymbol{\theta}_n \quad (1.1)$$

In this work, we will employ Bayesian statistics, where the posterior is related to the likelihood, \mathcal{L} , i.e. the probability of measuring the data given the parameters. For this we employ the central theorem of Bayesian statistics called the Bayes' theorem. It reads

$$p(\boldsymbol{\theta}|\mathbf{D}) = \frac{p_{\text{prior}}(\boldsymbol{\theta})}{E_{\text{evidence}}(\mathbf{D})} \mathcal{L}(\mathbf{D}|\boldsymbol{\theta}). \quad (1.2)$$

The prior encompasses all of our previous knowledge. It could be either an experimental prior when there was already a measurement of the model parameter done in the past, or it is a theoretical prior. A theoretical prior tells us that the theory requires parameters to be within some bounds, this could be for example the requirement that the neutrino mass is higher than zero or that there are more than zero neutrino species.

The Evidence can be understood as the normalization of the likelihood function. It is not accessible with MCMC methods or FI methods, so we will not go into further details of its interpretation. We just want to note that it is used when comparing different models to explain the same data. If we obtain the Evidence its value represents the goodness of the fit while also punishing adding additional parameters.

In the next sections, we will very briefly go over the two different forecasting methods.

1.1 The Markov Chain Monte Carlo Method

If we wanted to get the posterior from the likelihood we could just use the equation 1.1. In reality, this is not feasible as most cosmological problems do not have analytical solutions. Even the most simple problems like comoving distances have no analytical expression once you add dark energy into the mix. This is why we resort to numerical methods like the MCMC method. In this method the goal is to obtain a sequence of random variables, we call them $\{\theta_i | i = 1, \dots, N\}$. In our case, these variables will be a vector of our model parameters, but the idea is much more general. When this sequence or chain is marcovian it can be shown that the distributions of the random variables converge to the distributions of the parameters, i.e. the posterior. The requirement Marcovian means that each element of the chain has a probability that is only a function of the last element of the chain.

There are different algorithms to generate such a chain but the simplest one and the one that we use in this work is the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm starts with a random point in parameter space that is our first chain element θ_1 . It then proposes a new point θ' . For the proposal of the new point, we typically use some multivariate Gaussian that was our initial guess of the distributions of the parameters θ . We denote the distribution that we sample the point from as $q(\theta' | \theta_1)$.

This new point is then accepted with an acceptance probability

$$A(\theta_1 \rightarrow \theta') = \min \left(1, \frac{\mathcal{L}(D|\theta')}{\mathcal{L}(D|\theta_1)} \frac{q(\theta_1|\theta')}{q(\theta'|\theta_1)} \right). \quad (1.3)$$

The next element of the chain becomes then either θ' or stays θ_1 with a probability given by the acceptance probability. This procedure is then iterated until the element of the chain have converged to follow a static distribution.

When the distribution of points, $\pi(\theta)$, has become static, the probabilities of going from one point to another need to fulfil the condition of reversibility. This means that the points fulfil

$$\pi(\theta') P(\theta' \rightarrow \theta) = \pi(\theta) P(\theta \rightarrow \theta'), \quad (1.4)$$

where P denotes the transition probability of going from one point to the other. This condition is called detailed balance and is an expression of the condition that equilibrium states should not change under time reversal.

If we use our acceptance probability and the suggestion probability to calculate the transition probability we find

$$\pi(\theta') q(\theta|\theta') A(\theta' \rightarrow \theta) = \pi(\theta) q(\theta'|\theta) A(\theta \rightarrow \theta'). \quad (1.5)$$

Next we realize that either $A(\theta' \rightarrow \theta) = 1$ or $A(\theta \rightarrow \theta') = 1$ as the fraction on the right side of equation 1.3 is reciprocal under the exchange of the states. Without loss of generality we chose that

$A(\theta' \rightarrow \theta) = 1$ this leads to

$$\begin{aligned} \pi(\theta') q(\theta|\theta') &= \pi(\theta) q(\theta'|\theta) \frac{\mathcal{L}(\mathbf{D}|\theta')}{\mathcal{L}(\mathbf{D}|\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \\ \frac{\pi(\theta')}{\mathcal{L}(\mathbf{D}|\theta')} &= \frac{\pi(\theta)}{\mathcal{L}(\mathbf{D}|\theta)}. \end{aligned} \quad (1.6)$$

This means that the static distribution that the points tend to converge to is the likelihood up to some constant factor. Inserting this into equation 1.1 gives us our posterior up to some factor. This is precisely why when doing the Metropolis-Hastings algorithm there is no way of obtaining the evidence.

Some detail needs to be said about the good convergence of Markovian chains. Firstly to define what we mean by a converged chain. Different metrics are being used but in our case, we use the Gelman-Rubin test. In this test, we launch multiple chains from different starting points, and they all converge to the final distribution of the likelihood. Then we calculate two quantiles. The first quantity is the variance of the mean of the chains, i.e. how far apart the different chains centres from one another. The second quantity is the mean of the variance of each chain, i.e. how much are the chains themselves vary from their centre. For a converged chain, we would expect that the ratio of these two quantities is close to one, so we use this ratio to measure chain convergence.

Often denoted as the R-1 test we often require a chain to fulfill

$$R - 1 := \sqrt{\frac{\text{Var}(\text{chain means})}{\text{mean}(\text{chain variances})}} - 1 \stackrel{!}{<} 0.1 \quad (1.7)$$

to be considered converged. This requires our distribution q from which we propose points should be already quite close to target distribution \mathcal{L} . The reasoning is the follows, if the suggested points would be way off of the bulk of the likelihood then they would never be accepted and the chain could not move very much. In the end, the individual chains would have very low variances. The opposite effect happens when the proposal distribution would only propose points in the very centre of the likelihood, then the chains would accept every suggestion. This would make it that firstly the variance of the chain means would be quite high as the random walk would not converge to the center of the likelihood. Secondly, this would also have the effect that there could be regions of the likelihood that would never be explored by the chains.

To help with the convergence during each run when the chains had enough time to burn in and reached some milestones of convergence we update the proposal distribution. For this, we calculate the covariance of the chains and use a multivariate Gaussian with the sample covariance to propose points out of. This has the effect that the main requirement of a Markovian chain gets broken, this is because now the probability of going from one point to the next depends not only on the last point but also on previous points. This means that if we want to analyze the chains we need to always throw away the points of the chains before the last update of the proposal distribution.

The code that we use to run our MCMC is called MontePython (MP) and has two modes that are of interest to us. The first mode is the Metropolis-Hastings mode, it uses the same strategy that was discussed earlier to generate the chains where we also update the proposal distribution to have fast convergence. The second mode is the Fisher mode which is discussed in the next section. The reason why we use both modes of MP will become clearer in the Chapter about Validation.

1.2 The Fisher Information Method

We observe that for well-constrained parameters the posterior looks often like a multivariate Gaussian. The FI method employs this in that it is an approximation where the Gaussian is tailored to the likelihood and then the error is extracted from the Gaussian. The Idea of the formalism is the following: To parametrize a multivariate Gaussian you only need the covariance matrix, C , and the mean of the variables, $\boldsymbol{\mu}$. In our case, the variables would be cosmological parameters and the covariance then would contain the measurement uncertainties.

Assuming that the posterior can be approximated as a Gaussian we can write it as

$$p(\boldsymbol{\theta}|\mathbf{D}) \propto p_{\text{prior}}(\boldsymbol{\theta}) \mathcal{L}(\mathbf{D}|\boldsymbol{\theta}') \approx \mathcal{N} \exp \left[-\frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\mu}) \cdot C^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}) \right]. \quad (1.8)$$

For a Gaussian posterior, the mean is also at the best fit, this is why when we do a forecast we directly know the value of $\boldsymbol{\mu}$. It is the value of the model parameters at the fiducial point where we generated our fake data. We can now use that the covariance can be extracted from the log of the posterior by calculating the second derivative. It is

$$-\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log(p(\boldsymbol{\theta}|\mathbf{D})) \Big|_{\boldsymbol{\theta}=\boldsymbol{\mu}} \approx (C^{-1})_{ij}. \quad (1.9)$$

If we assume flat priors in the region of interest we can directly calculate the derivative of the likelihood. This leads to the definition of the Fisher Information matrix as

$$F_{ij} := -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log \mathcal{L}(\mathbf{D}|\boldsymbol{\theta}) \quad (1.10)$$

For Gaussian priors, we can just add the term later. We will discuss later on how to do that when we discuss adding different experiments. Non-Gaussian or flat priors are harder to approximate. The easiest method of dealing with exponential priors is to make it flat prior by transforming the relevant parameters into log space. This might make the contours less Gaussian depending on how well-constrained the parameters are.

The difficulty of the FI method is that often times it is not so clear how to obtain the second-order derivatives of the likelihood. Some codes were purposefully built to be differentiable i.e. that you can obtain the derivatives for free. The typical Einstein-Boltzmann solvers (EBS) `CLASS`[?] and `CAMB`[?] do not have that functionality, so we need to obtain the derivatives numerically. This is what `MP` does in Fisher mode. We calculate the second-order derivatives using a double-sided finite differences scheme. We note numerical derivatives with the symbol D , thus we can write

$$\frac{D^2 \log \mathcal{L}}{D\theta_i D\theta_j} = \frac{1}{4 h_i h_j} [\log \mathcal{L}(\theta_i + h_i, \theta_j + h_j) - \log \mathcal{L}(\theta_i + h_i, \theta_j - h_j) - \log \mathcal{L}(\theta_i - h_i, \theta_j + h_j) + \log \mathcal{L}(\theta_i - h_i, \theta_j - h_j)] \quad \text{for } i \neq j \quad (1.11)$$

$$\frac{D^2 \log \mathcal{L}}{D^2 \theta_i} = \frac{1}{h_i^2} [\log \mathcal{L}(\theta_i + h_i) + \log \mathcal{L}(\theta_i - h_i) - 2 \log \mathcal{L}(\theta_i)], \quad (1.12)$$

where we have left out the arguments of the log-likelihood that do not vary in each line. Here the advantage of using the FI method becomes clear, while the MCMC often needs multiple thousand calls of the EBS the FI method needs much less. In the Implementation we use for N parameters

we have to call the likelihood $2N$ times for the diagonal elements and $2N(N-1)$ times for the off-diagonal elements, making our total calls $2N^2$. Even then we can shave off some more time if we use that when varying nuisance parameters there is no need for calling the EBS anymore.

The calculation of the second derivative is numerically quite sensitive to the choice of stepsize. It was shown before in [?] that the optimal stepsize to calculate the derivatives is of the order of a few per cent of the marginalized error. And even then one needs to use very high precision settings of the EBS to be able to have stable results. This does defeat the purpose of the FI method as to obtain the marginalized error we would need to run an MCMC.

To make the computation of the FI matrix numerically stable we can use the explicit form of the photometric and spectroscopic likelihoods to write it using only the first derivatives of the observables. For the photometric likelihood we find

$$F_{ij} = \frac{1}{2} \sum_{\ell=\ell_{\min}}^{\ell_{\max}} (2\ell+1) f_{\text{sky}} \text{Tr} \left[(C_{\ell}^{\text{fid}})^{-1} (\partial_i C_{\ell}^{\text{th}}|_{\text{fid}}) (C_{\ell}^{\text{fid}})^{-1} (\partial_j C_{\ell}^{\text{th}}|_{\text{fid}}) \right]. \quad (1.13)$$

We have replaced the observed angular power spectrum with the theoretical angular power spectrum calculated at the fiducial like in the spectroscopic probe. There are two advantages of calculating the FI matrix like this, firstly the derivatives that show up are first order and thus much easier to calculate. To calculate them we use either of two finite differences stencils

$$\frac{DC_{\ell}^{\text{th}}}{D\theta_i} = \frac{1}{2h_i} [C_{\ell}^{\text{th}}(\theta_i + h_i) - C_{\ell}^{\text{th}}(\theta_i - h_i)] \quad (1.14)$$

$$\text{or} = \frac{1}{6h_i} [-11 C_{\ell}^{\text{th}}(\theta_i) + 18 C_{\ell}^{\text{th}}(\theta_i + h_i) - 9 C_{\ell}^{\text{th}}(\theta_i + 2h_i) + 2 C_{\ell}^{\text{th}}(\theta_i + 3h_i)]. \quad (1.15)$$

By default, we will stick to the double-sided finite difference and not to the one-sided one but to check the numerical stability of our derivatives we have tested both methods. First-order derivatives have the property to be more numerically stable. That is why we do not need to tune our stepsize as much as in the case of the second-order derivatives. We always use a relative stepsize of 1% for the cosmological parameters except for the neutrino mass, where we vary it by a relative 10%. The choice for the neutrinos stems from the fact, that the neutrino mass only has a small effect on our observables. For a 1% change of the neutrino mass, the change of the observables would be dominated by numerical noise, so we had to choose a higher stepsize. We will also test that the choice of a 10% stepsize for massive neutrinos produces stable results.

The second advantage of this method is that to compute all derivatives one only needs to call the code $2N$ since the first-order derivatives of the observables can then be multiplied to find the second-order derivatives of the likelihood.

For the spectroscopic likelihood, we find a similar expression for the likelihood

$$F_{ij} = \frac{1}{8\pi^2} \sum_i \int k_{\text{fid}}^2 \partial_i [\log P_{\text{th}}(k_{\text{fid}}, \mu_{\text{fid}}, z_i)]_{\text{fid}} \partial_i [\log P_{\text{th}}(k_{\text{fid}}, \mu_{\text{fid}}, z_i)]_{\text{fid}} V_i^{\text{fid}} dk_{\text{fid}} d\mu_{\text{fid}}. \quad (1.16)$$

Again we find that the FI matrix can be written with products of first-order derivatives. To calculate the FI matrix for our forecast we will use the code Cosmicfish (CF) that we have modified such that our changes to the recipes of the observables are consistent with our MP Implementation. Since CF never calculates the actual likelihood this gives us a great test for our likelihood Implementation if the FI matrix obtained by MP in Fisher mode agrees with the FI matrix from CF. This is further

discussed in the chapter ?? about the Validation.

Finally, the FI formalism gives us a guiding principle to make educated guesses for the proposal distributions. Firstly when we have already a covariance matrix of a previous MCMC, and we want to launch another MCMC but with fewer parameters, then just deleting the file's corresponding rows is not optimal but was the Implementation in MP. We have added a new method of fixing that leads to a better initial guess.

We calculate a matrix from the covariance of the MCMC that approximates the FI matrix by just inverting it. The matrix now has all the parameters that the actual Fisher would have. Now if we remind ourselves of the definition of the FI matrix as the second derivate of the log-likelihood then fixing a parameter is the same as never adding the corresponding lines to the matrix. Inversion then leads us to the desired covariance matrix. We illustrate this procedure below.

$$\begin{matrix} & \boldsymbol{\theta}_o & \boldsymbol{\theta}_f \\ \boldsymbol{\theta}_o & \begin{pmatrix} C_{oo} & C_{of} \\ C_{fo} & C_{ff} \end{pmatrix} \end{matrix} \xrightarrow{\text{invert}} \begin{matrix} & \boldsymbol{\theta}_o & \boldsymbol{\theta}_f \\ \boldsymbol{\theta}_o & \begin{pmatrix} F_{oo} & F_{of} \\ F_{fo} & F_{ff} \end{pmatrix} \end{matrix} \xrightarrow{\text{extract}} F_{oo} \xrightarrow{\text{invert}} \tilde{C}_{oo}$$

We have denoted the parameters we want to keep open with $\boldsymbol{\theta}_o$ and the parameters we want to fix with $\boldsymbol{\theta}_f$. In the Fisher approximation the new covariance \tilde{C}_{oo} is the covariance of the MCMC if the parameters had been fixed from the beginning. This also gives us an interpretation of what the diagonal elements of the Fisher mean. If we were to fix all parameters except one then the matrix F_{oo} is just a number. The inversion is then just its reciprocal value but the interpretation of this number would be the variance when every other parameter was fixed. This is precisely what the unmarginalized error is thus the diagonal of a Fisher contains the reciprocal unmarginalized errors. Next, we can ask ourselves the question of how to add multiple experiments together. On the level of the likelihood when neglecting cross-correlation the likelihood of a combined probe, \mathcal{L} , factorizes into the likelihoods of the individual probes, \mathcal{L}^1 and \mathcal{L}^2 . For the Fisher, we thus can write

$$F_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log \mathcal{L} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} [\log \mathcal{L}^1 + \log \mathcal{L}^2] = F_{ij}^1 + F_{ij}^2. \quad (1.17)$$

This means that combining two uncorrelated probes is the same as adding their fisher matrix. We can again use this to make an educated guess on the proposal distribution of an MCMC with combined probes, like the spectroscopic and the photometric probes when there are previous runs. Firstly we have to separate all of our parameters into parameters that are common to both likelihoods $\boldsymbol{\theta}_c$, i.e. the cosmological parameters, and then the nuisance parameters of both likelihoods $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Calculating the derivatives of one likelihood with respect to the nuisance parameters of the other likelihood makes little sense but could be done mathematically. The derivatives would just vanish so the Fisher Matrix would have empty rows of zeros in these and would be not invertible. But this helps us in the formulation of our recipe.

We start by inverting our MCMC covariance matrices to have the approximate FI matrices. We then add columns and rows of zeros that correspond to the nuisance parameters of the other likelihood. Finally, we can add the matrices and invert the result to get our new guess for the proposal covariance. Typically, this works very well for complimentary probes since breaking degeneracies makes the final posterior Gaussian enough for the approximation. We illustrate the final combination

recipe below

$$\begin{array}{c}
 \begin{array}{cc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_c \begin{pmatrix} C_{cc}^1 & C_{c1}^1 \\ C_{1d}^1 & C_{11}^1 \end{pmatrix} & \boldsymbol{\theta}_1 \begin{pmatrix} C_{cc}^1 & C_{c1}^1 \\ C_{1d}^1 & C_{11}^1 \end{pmatrix} \end{array}, \quad \begin{array}{cc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_c \begin{pmatrix} C_{cc}^2 & C_{c2}^2 \\ C_{2c}^2 & C_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_2 \begin{pmatrix} C_{cc}^2 & C_{c2}^2 \\ C_{2c}^2 & C_{22}^2 \end{pmatrix} \end{array} \xrightarrow{\text{invert}} \begin{array}{cc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_c \begin{pmatrix} F_{cc}^1 & F_{c1}^1 \\ F_{1d}^1 & F_{11}^1 \end{pmatrix} & \boldsymbol{\theta}_1 \begin{pmatrix} F_{cc}^1 & F_{c1}^1 \\ F_{1d}^1 & F_{11}^1 \end{pmatrix} \end{array}, \quad \begin{array}{cc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_c \begin{pmatrix} F_{cc}^2 & F_{c2}^2 \\ F_{2c}^2 & F_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_2 \begin{pmatrix} F_{cc}^2 & F_{c2}^2 \\ F_{2c}^2 & F_{22}^2 \end{pmatrix} \end{array} \\
 \\
 \begin{array}{ccc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_c \begin{pmatrix} F_{cc}^1 & F_{c1}^1 & 0 \\ F_{1d}^1 & F_{11}^1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \boldsymbol{\theta}_1 \begin{pmatrix} F_{cc}^1 & F_{c1}^1 & 0 \\ F_{1d}^1 & F_{11}^1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \boldsymbol{\theta}_2 \begin{pmatrix} F_{cc}^1 & F_{c1}^1 & 0 \\ F_{1d}^1 & F_{11}^1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad \begin{array}{ccc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_c \begin{pmatrix} F_{cc}^2 & 0 & F_{c2}^2 \\ 0 & 0 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_1 \begin{pmatrix} F_{cc}^2 & 0 & F_{c2}^2 \\ 0 & 0 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_2 \begin{pmatrix} F_{cc}^2 & 0 & F_{c2}^2 \\ 0 & 0 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} \end{array} \xrightarrow{\text{add}} \begin{array}{ccc} \boldsymbol{\theta}_c & \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_c \begin{pmatrix} F_{cc}^1 + F_{cc}^2 & F_{c1}^1 & F_{c2}^2 \\ F_{1c}^1 & F_{11}^1 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_1 \begin{pmatrix} F_{cc}^1 + F_{cc}^2 & F_{c1}^1 & F_{c2}^2 \\ F_{1c}^1 & F_{11}^1 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} & \boldsymbol{\theta}_2 \begin{pmatrix} F_{cc}^1 + F_{cc}^2 & F_{c1}^1 & F_{c2}^2 \\ F_{1c}^1 & F_{11}^1 & 0 \\ F_{2d}^2 & 0 & F_{22}^2 \end{pmatrix} \end{array} \\
 \\
 \xrightarrow{\text{invert}} \tilde{C}
 \end{array}$$

This also explains how to add a Gaussian prior to a Fisher afterwards when looking at the Bayes theorem 1.1 again. The prior can be pulled into the likelihood and act like an effective second experiment. The covariance matrix of the multivariate Gaussian prior directly defines an FI matrix that can be just added like an experiment afterwards. This of course needs the prior to be centred on the fiducial cosmology of the forecast.

A script to merge covariance matrices for MP was added by us. To show how well this works we have combined two test MCMCs of the two main *Euclid* probes and present the results in figure ?? . We generated this plot with the use of GetDist ¹ [?].

¹<https://getdist.readthedocs.io>

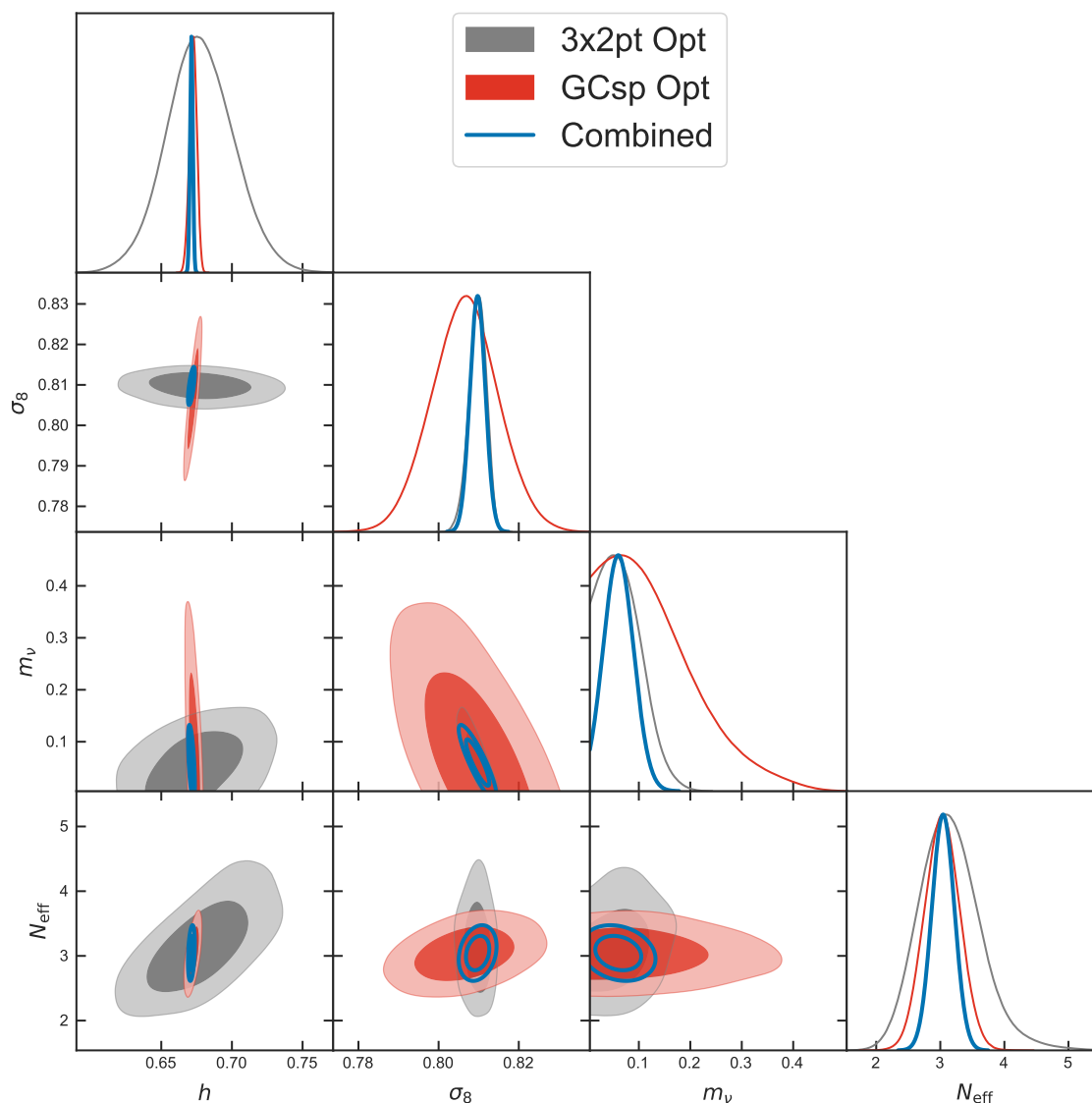


Figure 1.1: Demonstration of the covariance combination recipe. The MCMCs are done with the settings of the Validation section. The combined probe is not an MCMC, but the ellipses from the Gaussian approximation obtained with the recipe. This also demonstrates very well the complementarity of the *Euclid* probes.