Machine learning for cosmology: from consistency tests to simulation based inference

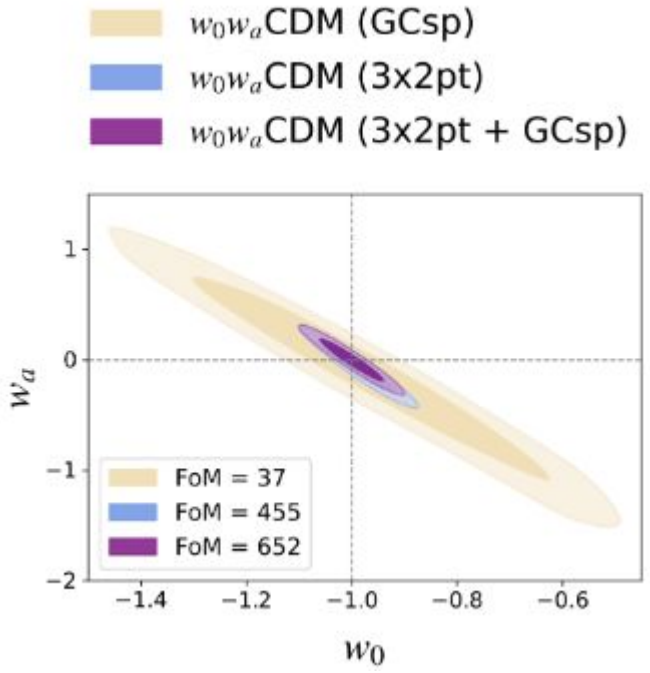
CASTLE 20/09/2024

Matteo Martinelli
(INAF - Osservatorio Astronomico di Roma)

Precision cosmology

After the Planck releases it became common to say that we are in the era of precision cosmology.

Such a precision is destined to improve thanks to Stage IV survey, and we will be able to measure parameters of our standard model with extreme precision

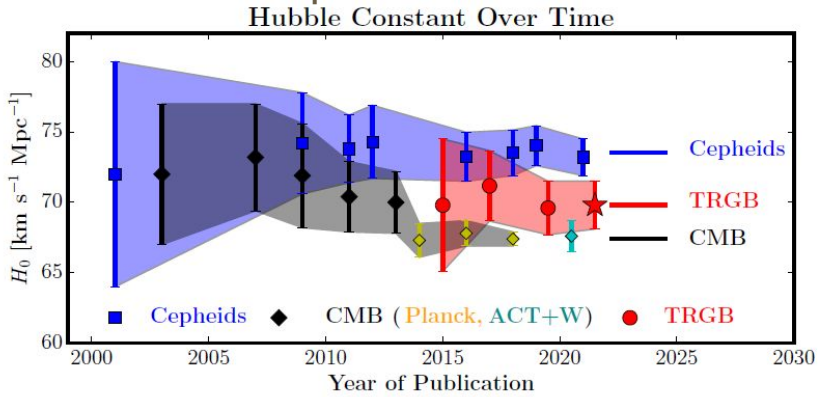


Euclid Collaboration: Mellier et al. (2024)

Cracks in the ΛCDM model

We are obtaining precise constraints, but are they accurate?
Our model might be wrong!

- Dark components (matter and energy) are still unknown
- Fine tuning and coincidence problem for Λ
- Tensions in measurements of parameters



Freedman ApJ (2021)

Search for new physics

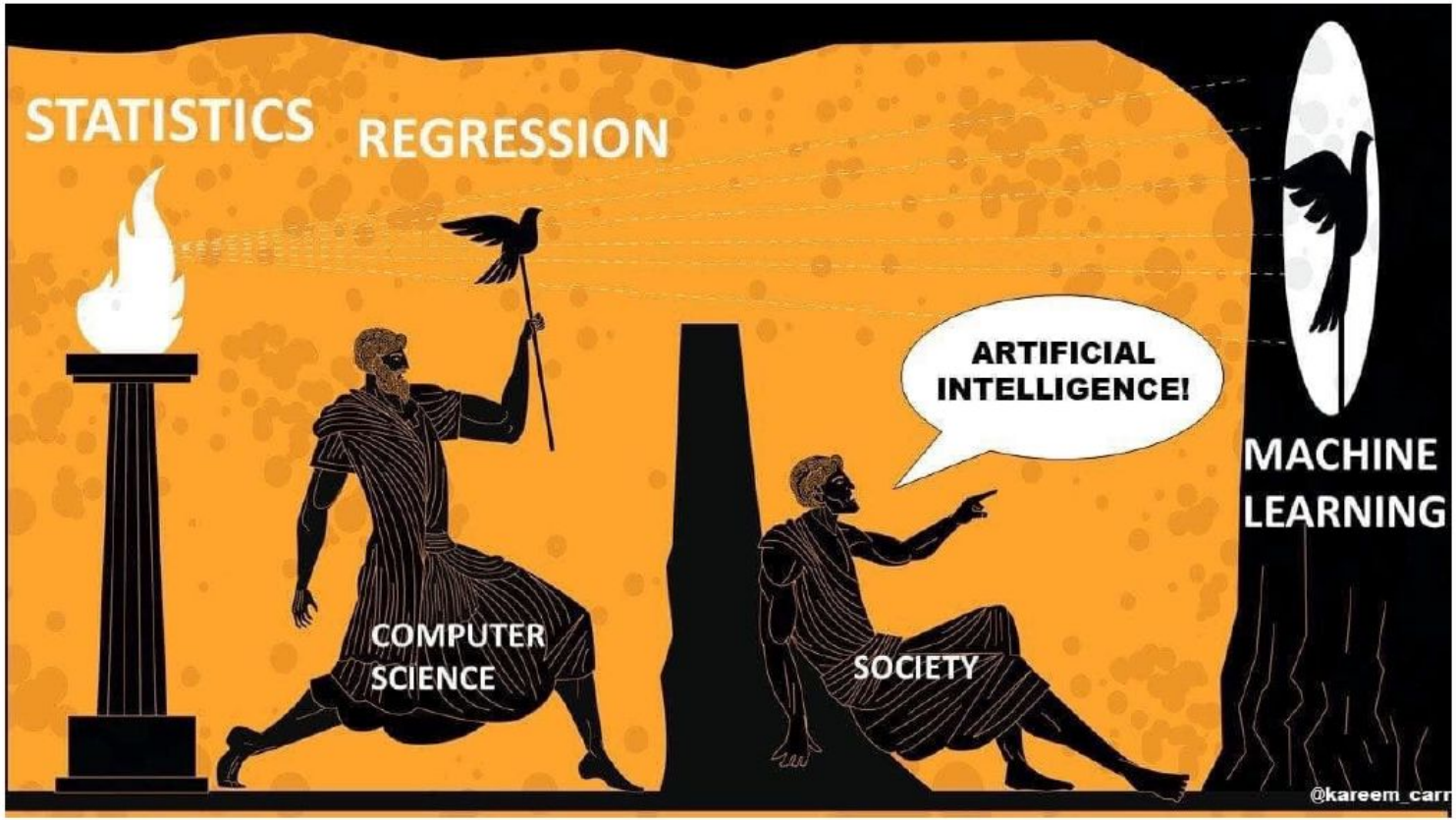
Tensions and issues of our standard model prompt us to test for alternatives.

- we have way too many model to think of testing them one by one looking for what's better than ΛCDM

- while modelling observables in our alternative model we are required to have extreme accuracy to not bias results (huge number of parameters)

- the choices we do for theoretical models and parameterizations can significantly affect results

What can machine learning do for us?



What can machine learning do for us?

- Reconstruction:
  obtain trend for cosmological functions without assuming any model

What can machine learning do for us?

- Reconstruction:
  obtain trend for cosmological functions without assuming any model

- Inference:
  new parameter inference methods, avoiding issues of current approaches and obtaining results faster

What can machine learning do for us?

- Reconstruction:
  obtain trend for cosmological functions without assuming any model

- Inference:
  new parameter inference methods, avoiding issues of current approaches and obtaining results faster

- Classification:
  use NN classifiers to distinguish different progenitors of astrophysical events (e.g. PBH in GW catalogues, DM events in gamma rays observations)

ML and function reconstruction

Reconstructing cosmological functions

A first, straightforward, application of Machine Learning for our purposes is to reconstruct cosmological functions without any model assumptions.

This allows to extract information directly on functions rather than on parameters.

Reconstructing techniques also allow you to obtain derivatives of cosmological and combine all reconstructions together to obtain non observable quantities.

Gaussian processes

Reconstructing a function with GP means assuming that your data are a Gaussian realization of the function with a Gaussian noise provided by the data covariance

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} K(X,X) + C & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix} \right)$$

Gaussian processes

Reconstructing a function with GP means assuming that your data are a Gaussian realization of the function with a Gaussian noise provided by the data covariance

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} K(X,X) + C & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix} \right)$$

The Kernel function contains the “hyper-parameters”. These depend on the choice of the Kernel. Once these are determined by feeding data to the GP, you can obtain the function

$$\begin{aligned} \mathrm{mean}(f^*) &= \mu^* + K(X^*,X)[K(X,X) + C]^{-1}(y - \mu) \\ \mathrm{cov}(f^*) &= K(X^*,X^*) \\ &\quad - K(X^*,X)[K(X,X) + C]^{-1} K(X,X^*) \end{aligned}$$

Application: consistency check for GR (and other stuff)

We assume (flat) w(z)CDM and related the EoS of DE to observable quantities, trying to obtain it from background and perturbation equations.

Application: consistency check for GR (and other stuff)

We assume (flat) w(z)CDM and related the EoS of DE to observable quantities, trying to obtain it from background and perturbation equations.

Looking at the background expansion we find

$$\frac{\mathrm{d}\ln H(z)}{\mathrm{d}\ln(1+z)} \equiv -\frac{\dot{H}(z)}{H^2(z)} = \frac{3}{2} + \frac{3}{2}\left[1-\Omega_{\mathrm{m}}(z)\right] w_{\mathrm{de}}(z)$$

$$w_{\mathrm{bg}}(z) = \frac{1}{1-\Omega_{\mathrm{m}}(z)}\left[\frac{2}{3}\frac{\mathrm{d}\ln H(z)}{\mathrm{d}\ln(1+z)} - 1\right]$$

Perenon, MM et al. Phys. Dark. Univ. (2022)

Application: consistency check for GR (and other stuff)

We assume (flat) w(z)CDM and related the EoS of DE to observable quantities, trying to obtain it from background and perturbation equations.

Looking at the background expansion we find

$$\frac{\mathrm{d}\ln H(z)}{\mathrm{d}\ln(1+z)} \equiv -\frac{\dot{H}(z)}{H^2(z)} = \frac{3}{2} + \frac{3}{2}\left[1-\Omega_{\mathrm{m}}(z)\right] w_{\mathrm{de}}(z)$$

$$w_{\mathrm{bg}}(z) = \frac{1}{1-\Omega_{\mathrm{m}}(z)}\left[\frac{2}{3}\frac{\mathrm{d}\ln H(z)}{\mathrm{d}\ln(1+z)} - 1\right]$$

while from the linear evolution of perturbations we get

$$\ddot{\delta}_{\mathrm{m}}(z) + 2H(z)\dot{\delta}_{\mathrm{m}}(z) - \frac{3}{2}\Omega_{\mathrm{m}}(z)H^2(z)\delta_{\mathrm{m}}(z) = 0$$

$$f(z) \equiv \frac{\mathrm{d}\ln\delta}{\mathrm{d}\ln(1+z)}$$

$$w_{\mathrm{gr}}(z) = \frac{1}{1-\Omega_{\mathrm{m}}(z)}\left[\frac{2}{3}\left(f(z) - \frac{\mathrm{d}\ln f(z)}{\mathrm{d}\ln(1+z)}\right) - \frac{\Omega_{\mathrm{m}}(z)}{f(z)} + \frac{1}{3}\right]$$

Perenon, MM et al. Phys. Dark. Univ. (2022)

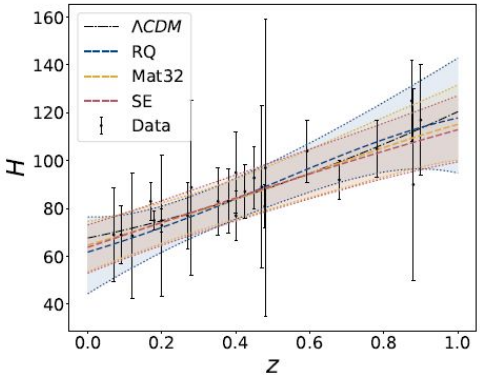
A promising approach

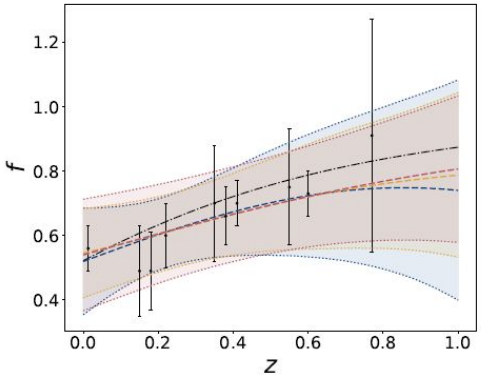
$$w_{\rm bg}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\frac{{\rm d}\ln H(z)}{{\rm d}\ln(1+z)} - 1\right]$$

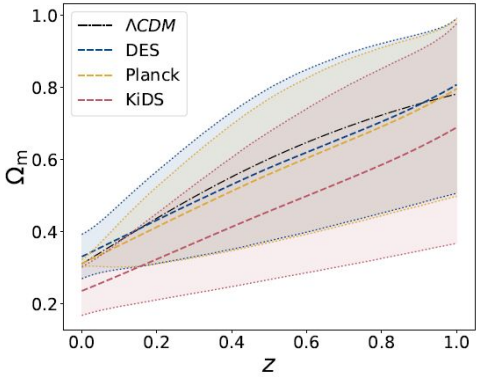
$$w_{\rm gr}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\left(f(z) - \frac{{\rm d}\ln f(z)}{{\rm d}\ln(1+z)}\right) - \frac{\Omega_{\rm m}(z)}{f(z)} + \frac{1}{3}\right]$$

By exploiting Machine Learning algorithms (GP), one can minimize assumptions and reconstruct functions in a differentiable way.

$$\Omega_{\rm m}(z) = \Omega_{\rm m,0}\frac{H_0^2}{H^2(z)}$$





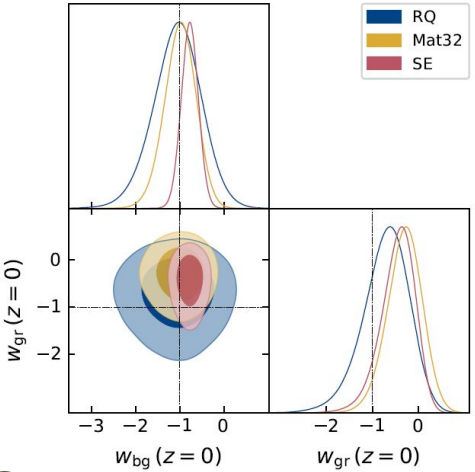


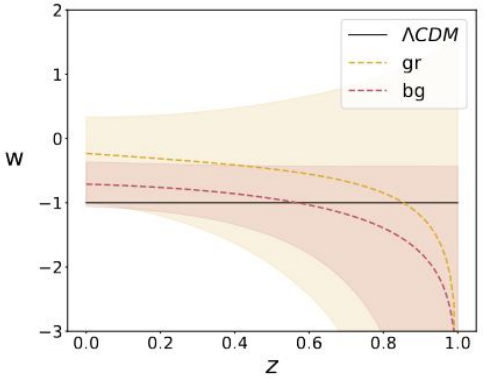
Perenon, MM et al. Phys. Dark. Univ. (2022)

A promising approach

By exploiting Machine Learning algorithms (GP), one can minimize assumptions and reconstruct functions in a differentiable way.

Plugging the reconstructed functions into our estimators we can compare the results.





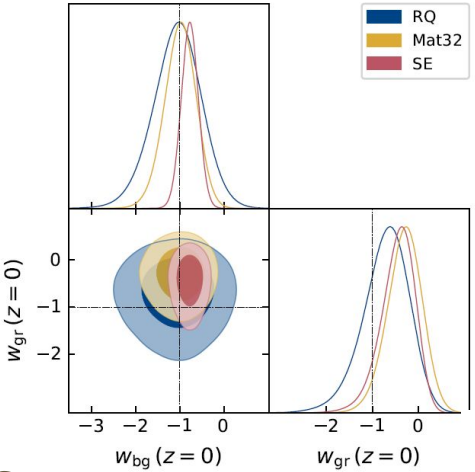
Perenon, MM et al. Phys. Dark. Univ. (2022)

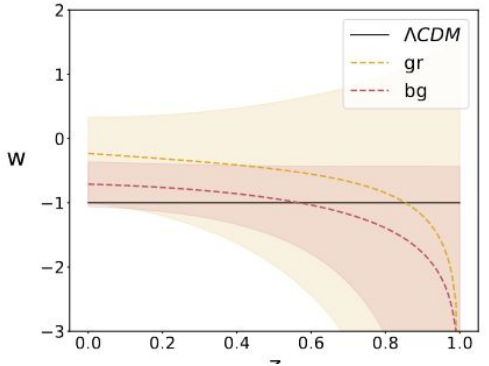
A promising approach

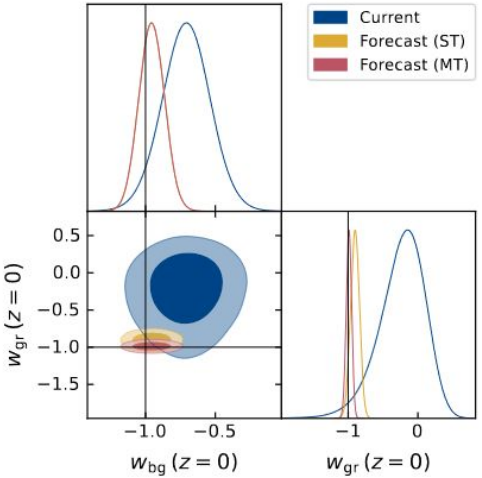
By exploiting Machine Learning algorithms (GP), one can minimize assumptions and reconstruct functions in a differentiable way.

Plugging the reconstructed functions into our estimators we can compare the results.

With the improvement of data from LSS survey, this approach could be significantly improved.







Perenon, MM et al. Phys. Dark. Univ. (2022)

Setbacks

ML reconstructions method as GP allow us to avoid model assumptions and obtain nice consistency checks... but:

- Be careful of leftover assumptions in your tests;
- No need to specify a model and/or choose a parameterization, however these methods depend on hyper-parameters that need to be handled with care (priors, kernels, etc...);
- Assumptions done in data extraction are completely hidden;
- ML methods stability can be problematic with few/bad data

Perenon, MM et al. Phys. Dark. Univ. (2021)

ML and parameter inference

Simulation based inference

SBI techniques allow to overcome some of the problems appearing with the increased precision of our experiments, e.g.

- high posterior dimensionality: in order to model observables accurately, nuisance effects must be included, introducing several parameters (O(100) for Stage IV), which can be complicated to sample

- likelihood modelling: SBI methods avoid assumptions on the shape of the likelihood function, e.g. Gaussianity. Likelihood information is accessed implicitly, with data realizations given different sets of parameters

SBI with Marginal Neural Ratio Estimation (MNRE)

What we want our analyses to provide us is the posterior of parameters

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

Our standard methods sample the parameter space to reconstruct the posterior:

- requires modelling the likelihood
- can get complicated and expensive when you have MANY parameters
- samples the whole parameter space, even things we want to throw away

SBI with Marginal Neural Ratio Estimation (MNRE)

What we want our analyses to provide us is the posterior of parameters

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

MNRE takes a different approach:

SBI with Marginal Neural Ratio Estimation (MNRE)

What we want our analyses to provide us is the posterior of parameters

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

MNRE takes a different approach:

- draw parameters from prior and simulate data from joint distribution

$$\{(\boldsymbol{x}^1, \boldsymbol{\theta}^1), ..., (\boldsymbol{x}^N, \boldsymbol{\theta}^N)\} \qquad p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

SBI with Marginal Neural Ratio Estimation (MNRE)

What we want our analyses to provide us is the posterior of parameters

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

MNRE takes a different approach:

- draw parameters from prior and simulate data from joint distribution

$$\{(\boldsymbol{x}^1, \boldsymbol{\theta}^1), ..., (\boldsymbol{x}^N, \boldsymbol{\theta}^N)\} \qquad p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- construct samples from product of marginal probabilities (pair shuffling)

$$p(\boldsymbol{x})p(\boldsymbol{\theta})$$

SBI with Marginal Neural Ratio Estimation (MNRE)

What we want our analyses to provide us is the posterior of parameters

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

MNRE takes a different approach:

- draw parameters from prior and simulate data from joint distribution

$$\{(\boldsymbol{x}^1, \boldsymbol{\theta}^1), ..., (\boldsymbol{x}^N, \boldsymbol{\theta}^N)\} \qquad p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- construct samples from product of marginal probabilities (pair shuffling)

$$p(\boldsymbol{x})p(\boldsymbol{\theta})$$

- use these samples to train a NN so that when present a set of data it can return

$$r(\boldsymbol{x}; \boldsymbol{\theta}) \equiv \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{p(\boldsymbol{\theta})}$$

Why marginal?

Once the NN returns the posterior-to-prior ratio, we can use this to weight samples from extracted from the prior, but we can do this only on a subset of parameters.

$$\boldsymbol{\vartheta} \subset \boldsymbol{\theta} = (\boldsymbol{\vartheta}, \boldsymbol{\eta})$$

- we can directly achieve 1- and 2-dimensional marginal posteriors, without needing to sample the full joint posterior: more efficient than traditional methods (and other SBI techniques)

- we can ignore large numbers of nuisance parameters, targeting only the parameters of interest

An example: LSS constraints with MNRE

By applying the MNRE approach, through the Swyft code, to LSS we can speed up parameter inference significantly.

We generate theoretical expectations of 3x2pt observables for parameters extracted from the prior and train the NN.

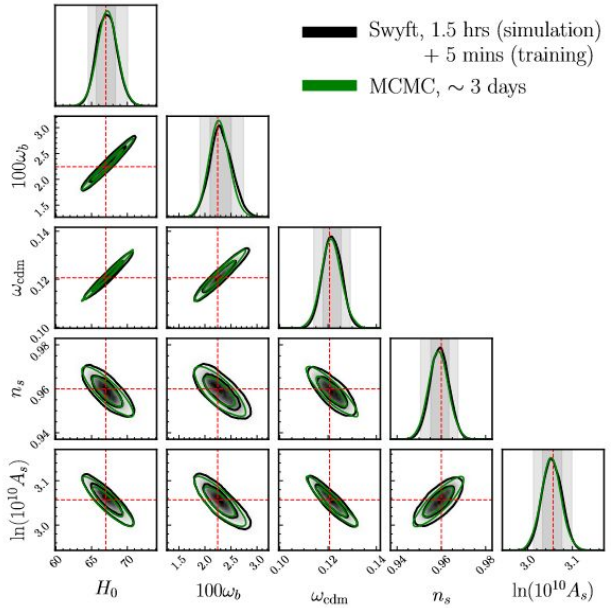
We then provide a synthetic dataset and use the NN to obtain marginal constraints.

An example: LSS constraints with MNRE

By applying the MNRE approach, through the Swyft code, to LSS we can speed up parameter inference significantly.

We generate theoretical expectations of 3x2pt observables for parameters extracted from the prior and train the NN.

We then provide a synthetic dataset and use the NN to obtain marginal constraints.



Franco-Abellan et al. (2024)

Setbacks

While SBI allows us to overcome some analysis issues, e.g. the likelihood modelling, we still need to face some setbacks

- this approach still requires us to assume a model from which, given the parameters we can obtain a theoretical expectation for data. Are we accurate?
- we can still be limited by simulation speed and by the number of parameters
- Neural network architecture must be set up carefully

Conclusions

Take home messages

- The improvement in data quality from current and upcoming experiments will provide great precision in our measurement, but also new issues
- Model accuracy becomes crucial. This requires heavier calculation which could become problematic. We should also think of testing our models with more agnostic approaches.
- Machine learning approaches can provide a framework to tackle these issues
  - Reconstruction of fundamental cosmological function for consistency tests
  - Simulation based inference for faster constraints and to avoid likelihood modelling
- Still a long road ahead! We need to ensure the accuracy of these methods and the robustness for different data quality and type.

Extra slides

Comparing results

$$w_{\rm bg}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\frac{{\rm d}\ln H(z)}{{\rm d}\ln(1+z)} - 1\right]$$

$$w_{\rm gr}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\left(f(z) - \frac{{\rm d}\ln f(z)}{{\rm d}\ln(1+z)}\right) - \frac{\Omega_{\rm m}(z)}{f(z)} + \frac{1}{3}\right]$$

We can obtain measurements of the EoS from different datasets and compare the results. If our assumptions are valid, results need to coincide.

Comparing results

$$w_{\rm bg}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\frac{{\rm d}\ln H(z)}{{\rm d}\ln(1+z)} - 1\right]$$

$$w_{\rm gr}(z) = \frac{1}{1-\Omega_{\rm m}(z)}\left[\frac{2}{3}\left(f(z) - \frac{{\rm d}\ln f(z)}{{\rm d}\ln(1+z)}\right) - \frac{\Omega_{\rm m}(z)}{f(z)} + \frac{1}{3}\right]$$

We can obtain measurements of the EoS from different datasets and compare the results. If our assumptions are valid, results need to coincide.

- $w_{\rm bg} = w_{\rm gr} \neq -1$, the same mechanism breaks ΛCDM and we can use the reconstructions to trace its behaviour;
- $w_{\rm bg} \neq -1 \neq w_{\rm gr}$, competing effects at play in the two sectors (e.g. massive neutrinos vs MG);
- $w_{\rm bg} = -1, w_{\rm gr} \neq -1$ only perturbations affected (MG?), it can also hint for problems in how we obtain the data on f (z).

Something more about MNRE

We want the posterior-to-prior ratio, so that we can wait samples

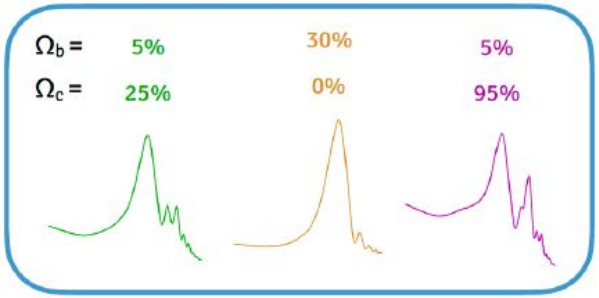
$$r(\boldsymbol{x};\boldsymbol{\theta}) \equiv \frac{p(\boldsymbol{x},\boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{p(\boldsymbol{\theta})}$$

we can train a binary classifier

$$d_\phi(\boldsymbol{x},\boldsymbol{\theta}) \simeq \begin{cases} 1 & \text{if} \quad (\boldsymbol{x},\boldsymbol{\theta}) \sim p(\boldsymbol{x},\boldsymbol{\theta}), \\ 0 & \text{if} \quad (\boldsymbol{x},\boldsymbol{\theta}) \sim p(\boldsymbol{x})p(\boldsymbol{\theta}). \end{cases}$$

$$\ell[d_\phi(\boldsymbol{x},\boldsymbol{\theta})] = -\int d\boldsymbol{x}d\boldsymbol{\theta}\,[p(\boldsymbol{x},\boldsymbol{\theta})\ln d_\phi(\boldsymbol{x},\boldsymbol{\theta}) + p(\boldsymbol{x})p(\boldsymbol{\theta})\ln(1-d_\phi(\boldsymbol{x},\boldsymbol{\theta}))]$$

$(\mathbf{x},\boldsymbol{\theta}) \sim p(\mathbf{x},\boldsymbol{\theta})$ (jointly drawn)



$(\mathbf{x},\boldsymbol{\theta}) \sim p(\mathbf{x})p(\boldsymbol{\theta})$ (marginally drawn)

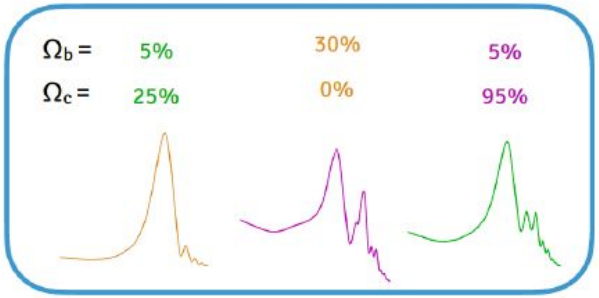


Figure courtesy of Guadalupe Cañas Herrera

Other SBI applications

What we did is mostly an attempt to speed up parameter inference, but SBI can have several other applications, e.g.

Other SBI applications

What we did is mostly an attempt to speed up parameter inference, but SBI can have several other applications, e.g.

- field-level inference: if we can obtain simulations of an observation field, we can use this to train the NN. No need to use summary statistics, we let the NN choose its own;

  e.g. Villaescusa-Navarro et al. (2021) , von Wietersheim-Kramsta et al. (2024)

Other SBI applications

What we did is mostly an attempt to speed up parameter inference, but SBI can have several other applications, e.g.

- field-level inference: if we can obtain simulations of an observation field, we can use this to train the NN. No need to use summary statistics, we let the NN choose its own;

  e.g. Villaescusa-Navarro et al. (2021) , von Wietersheim-Kramsta et al. (2024)

- distribution reconstruction: obtain cosmological observables from the distribution of a given observable, e.g. GW d_L with no redshift info

$$\ln L(\vec{O}|\vec{\theta}) = N_{\rm obs} \ln N_{\rm Th}(\vec{\theta}) - N_{\rm Th}(\vec{\theta}) + \sum_{i=1}^{N_{\rm obs}} \ln \left[\int \mathrm{d}\,\bar{O}_i \frac{N(\bar{O}_i|O_i,\sigma_i)\,T(\bar{O}_i|\vec{\theta})}{\int \mathrm{d}\,\vec{\theta}\,T(\bar{O}_i|\vec{\theta})P(\vec{\theta})} \right]$$

MM et al. *in preparation*