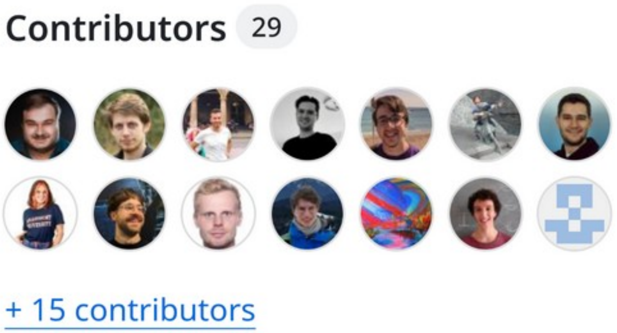
Interfacing EOS WET likelihoods to a SMEFT analysis

Matthew Kirk

SMEFT Tools 2025
27 Jan 2025

EOS

- Software for flavour physics phenomenology
- C++ backend, Python front-end
- Can be used via Jupyter notebook or CLI
- Developed by



What is EOS designed to do?

1) Calculate state of the art theory predictions

2) Fit theory parameters and/or observables from data

3) Produce Monte-Carlo pseudo-event samples

Theory predictions

- As of v1.0.13, more than 1300 (pseudo-)observables in a variety of processes

List of Observables

- Observables in (semi)leptonic b-hadron decays
- Observables in (semi)leptonic c-hadron decays
- Observables in rare (semi)leptonic and radiative b-hadron decays
- Observables in neutral meson mixing
- Pseudo-observables for the non-local form factors
- Form factors
- Observables in scattering processes

Theory predictions

- Most processes have a variety of underlying parameterisations available (e.g. for hadronic physics, …) which can be selected at run time

List of Observables

- Observables in (semi)leptonic b-hadron decays
- Observables in (semi)leptonic c-hadron decays
- Observables in rare (semi)leptonic and radiative b-hadron decays
- Observables in neutral meson mixing
- Pseudo-observables for the non-local form factors
- Form factors
- Observables in scattering processes

Fitting theory parameters and observables

- EOS uses Bayesian approach – specify prior probability and likelihood from experiment, and find posterior probability of parameters
- Analysis in EOS is done in the WET (/ LEFT), so parameters include WET coefficients

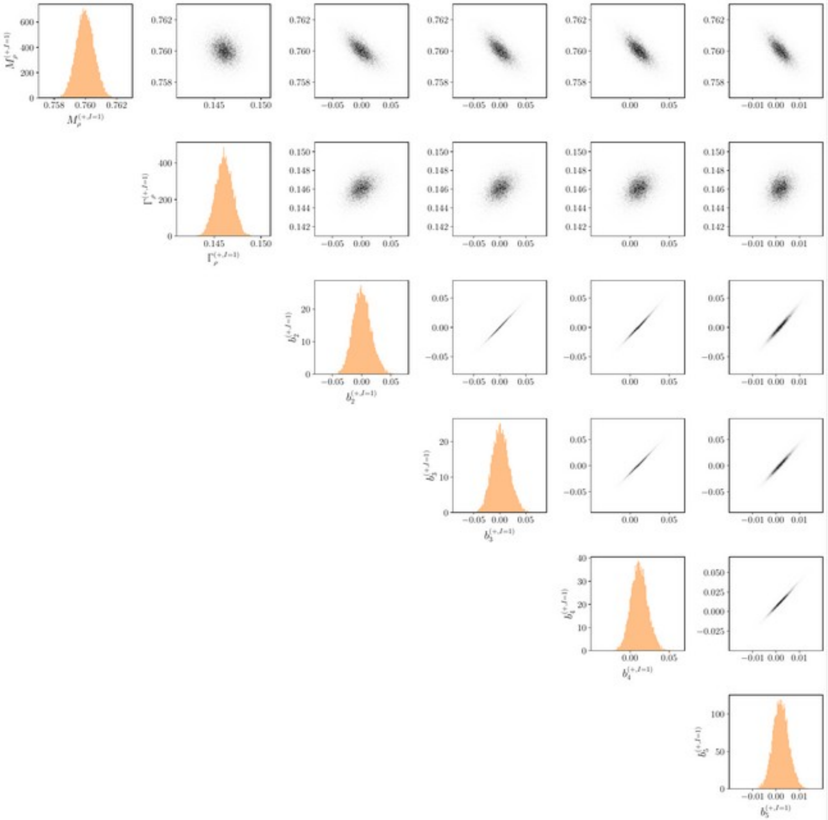
Fitting theory parameters and observables

- Specify priors on your parameters
- Specify likelihoods for observables that depend on your parameters

```yaml
likelihoods:
  - name: EXP-Belle-timelike
    constraints:
      - '0->pipi::Abs{f_+}^2@Belle:2008C;end=19'
  - name: EXP-CLEO-timelike
    constraints:
      - '0->pipi::Abs{f_+}^2@CLEO:2000B;end=29'
  - name: EXP-NA7-spacelike
    constraints:
      - '0->pipi::Abs{f_+}^2@NA7:1986A'
  - name: EXP-JLAB # From arXiv:0809.3052, table I, converted from f_pi to |f_pi|^2
    manual_constraints:
      "0->pipi::f+":
        type: "Gaussian"
        observable: "0->pipi::Abs{f_+}^2(q2);I=1"
        kinematics: {'q2': -2.45}
        options: {'form-factors': 'KKRvD2024'}
        mean: 0.027889
        sigma-stat: {"hi": 0.0033, "lo": 0.0033}
        sigma-sys:  {"hi": 0.0043, "lo": 0.0023}

priors:
  - name: FF-fp-I1-pole-params-order5
    descriptions:
      - { 'parameter': '0->pipi::M_(+,1)@KKRvD2024', 'min':  0.757, 'max':  0.763, 'type': 'uniform'}
      - { 'parameter': '0->pipi::Gamma_(+,1)@KKRvD2024', 'min':  0.141, 'max':  0.151, 'type': 'uniform'}
  - name: FF-fp-I1-expansion-order5
    descriptions:
      - { 'parameter': '0->pipi::b_(+,1)^2@KKRvD2024', 'min':  -0.08, 'max':  0.08, 'type': 'uniform'}
      - { 'parameter': '0->pipi::b_(+,1)^3@KKRvD2024', 'min':  -0.09, 'max':  0.09, 'type': 'uniform'}
      - { 'parameter': '0->pipi::b_(+,1)^4@KKRvD2024', 'min':  -0.05, 'max':  0.07, 'type': 'uniform'}
      - { 'parameter': '0->pipi::b_(+,1)^5@KKRvD2024', 'min':  -0.02, 'max':  0.02, 'type': 'uniform'}


posteriors:
  - name: FF-fp-I1-order5
    global_options:
      form-factors: KKRvD2024
    fixed_parameters:
      0->pipi::t_0@KKRvD2024: -1.0
    prior:
      - FF-fp-I1-pole-params-order5
      - FF-fp-I1-expansion-order5
    likelihood:
      - EXP-Belle-timelike
      - EXP-CLEO-timelike
      - EXP-NA7-spacelike
      - EXP-JLAB
```

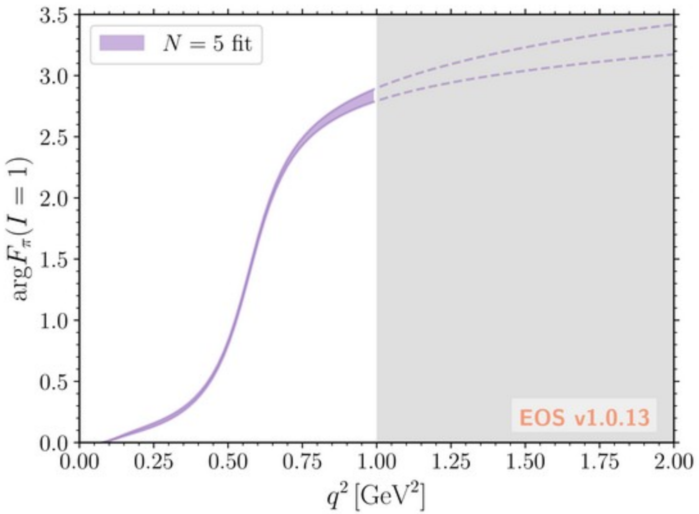
Fitting theory parameters and observables

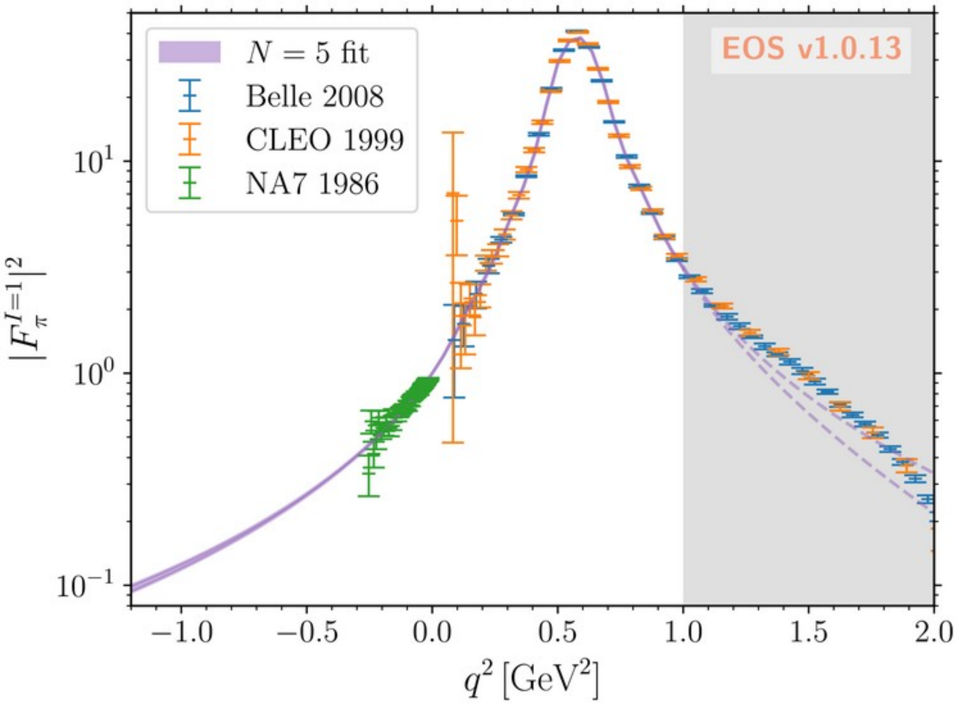
- Use nested sampling (specifically dynesty) to generate weighted samples corresponding to the posterior



Fitting theory parameters and observables

- Which can be used to predict observables





Monte-Carlo samples

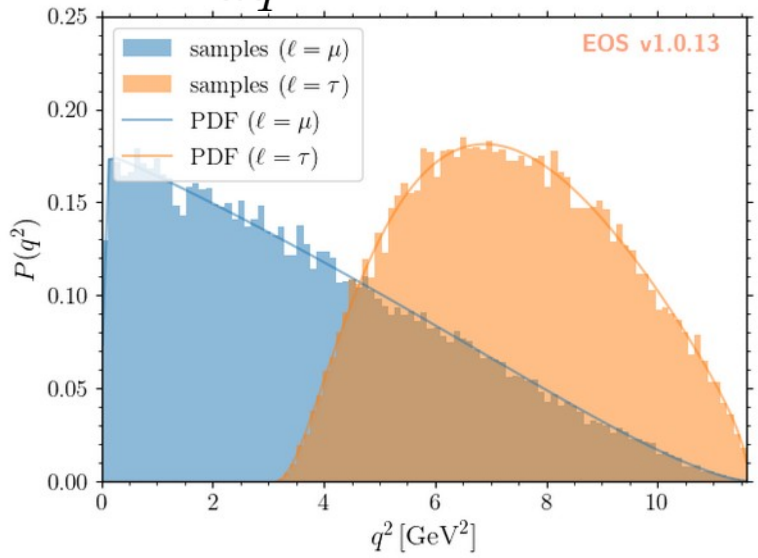
- 27 SignalPDFs predefined in EOS
- Built in functions to sample from them

- **List of Signal PDFs**
  - **Signal PDFs in (semi)leptonic b-hadron decays**
    - Signal PDFs in leptonic and photoleptonic B decays
    - Signal PDFs in semileptonic $B \to P\ell^-\bar{\nu}$ decays
    - Signal PDFs in semileptonic $B \to V\ell^-\bar{\nu}$ decays
    - Signal PDFs in semileptonic $B \to PP\ell^-\bar{\nu}$ decays
    - Signal PDFs in semileptonic $\Lambda_b \to 1/2^+\ell^-\bar{\nu}$ decays
    - Signal PDFs in semileptonic $\Lambda_b \to 3/2^-\ell^-\bar{\nu}$ decays
  - **Signal PDFs in rare (semi)leptonic b-hadron decays**
    - Signal PDFs in rare semileptonic $B \to P\{\bar{\nu}\nu, \ell^+\ell^-\}$ decays
    - Signal PDFs in rare semileptonic $B \to V\{\bar{\nu}\nu, \ell^+\ell^-\}$ decays

Monte-Carlo samples

- 27 SignalPDFs predefined in EOS
- Built in functions to sample from them
- E.g. look at differential distributions of $B \to D\ell\nu$

$$\frac{d\Gamma}{dq^2}(B \to D\ell\nu)$$



EOS

- More information? See https://eoshep.org/
- Development on Github @ https://github.com/eos/eos
  - Very happy to accept new contributions
- More questions? Or want to ask about what EOS could do for you? Join the Discord server: https://discord.gg/hyPu7f7K6W

Using EOS for SMEFT

- All constraint information from EOS is in the language of WET / LEFT
  - Available datasets can be found at: https://github.com/eos/data/
- So how can you use them in a SMEFT analysis?

Using EOS for SMEFT

- Let's say you are studying BSM (either specific model or SMEFT)
- After RG running, you expect effects in some low energy B physics observables
- E.g. $B \to \pi \ell \nu$

Using EOS for SMEFT

- You see / remember this paper:

Toward a complete description of $b \to u\ell^-\bar{\nu}$ decays within the Weak Effective Theory

Domagoj Leljak,[a] Blaženka Melić,[a] Filip Novak,[b] Méril Reboud,[c] Danny van Dyk[c]

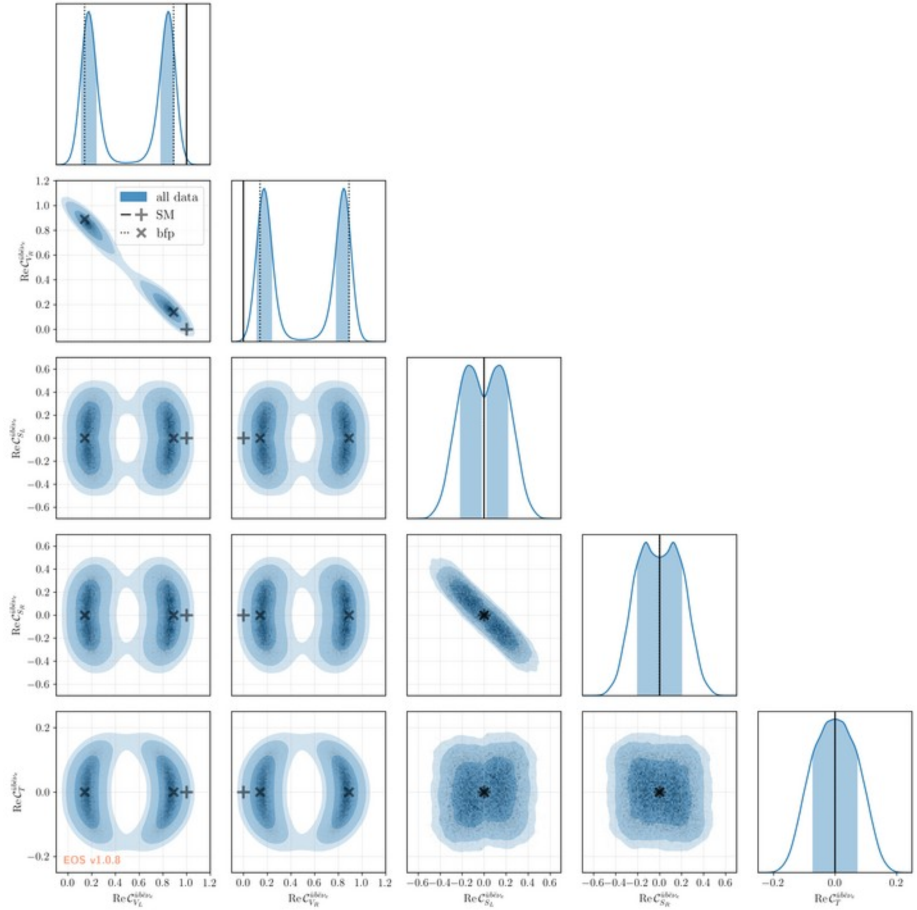
[a] *Rudjer Boskovic Institute, Division of Theoretical Physics, Bijenička 54, HR-10000 Zagreb, Croatia*
[b] *Physik Department T31, Technische Universität München, 85748 Garching, Germany*
[c] *Institute for Particle Physics Phenomenology and Department of Physics, Durham University, Durham DH1 3LE, UK*

E-mail: domagojleljak@gmail.com, melic@irb.hr, filip.novak@tum.de, merilreboud@gmail.com, danny.van.dyk@gmail.com

Abstract: We fit the available data on exclusive semileptonic $b \to u\ell^-\bar{\nu}$ decays within the Standard Model and in the Weak Effective Theory. Assuming Standard Model dynamics, we find $|V_{ub}| = 3.59^{+0.13}_{-0.12} \times 10^{-3}$. Lifting this assumption, we obtain stringent constraints on the coefficients of the $ub\ell\nu$ sector of the Weak Effective Theory. Performing a Bayesian model comparison, we find that a beyond the Standard Model interpretation is favoured over a Standard Model interpretation of the available data. We provide a Gaussian mixture model that enables the efficient use of our fit results in subsequent analyses beyond the Standard Model, within and beyond the framework of the Standard Model Effective Field Theory.

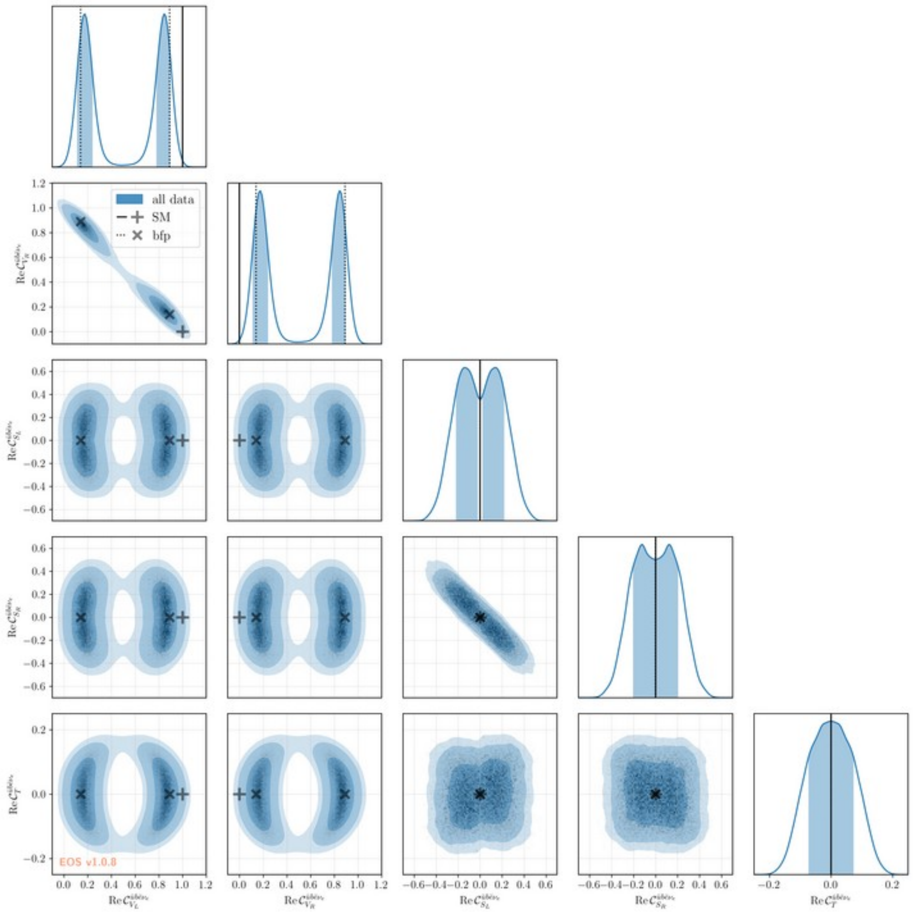
Using EOS for SMEFT

- You see this paper
- With this result for a likelihood on WET coefficients:



Using EOS for SMEFT

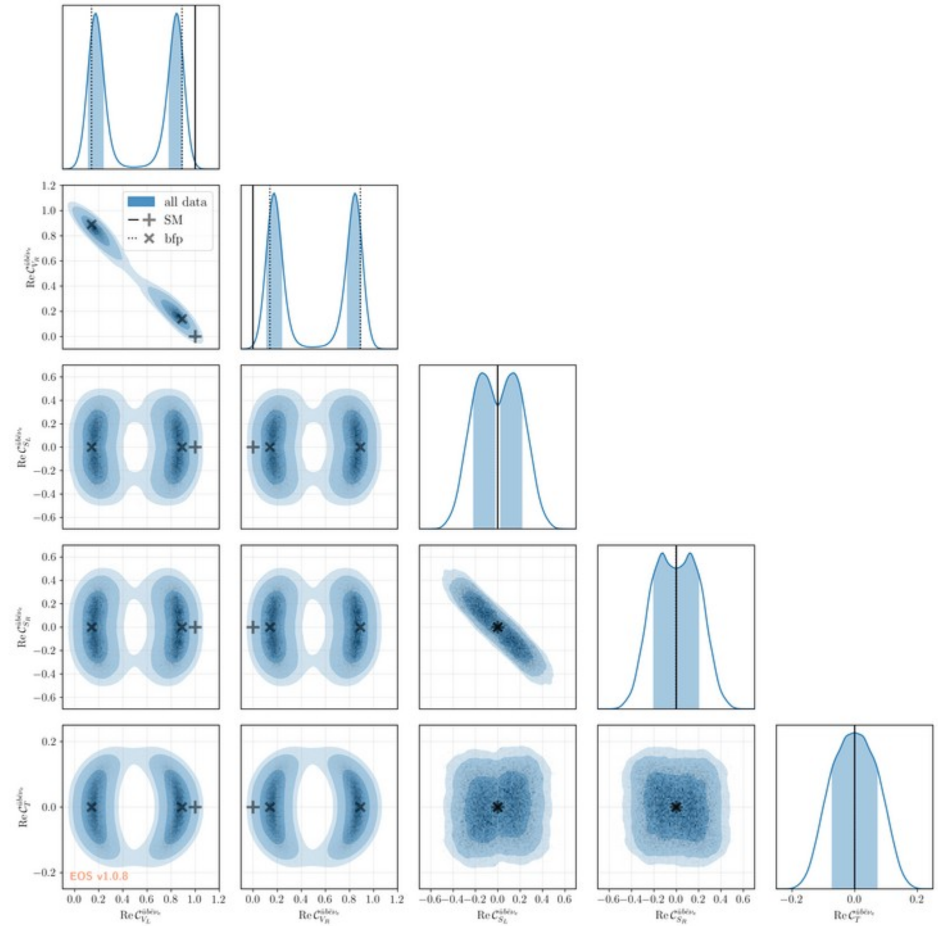
- You want to include this likelihood in your fit
- But it’s not a nice simple Gaussian or anything…



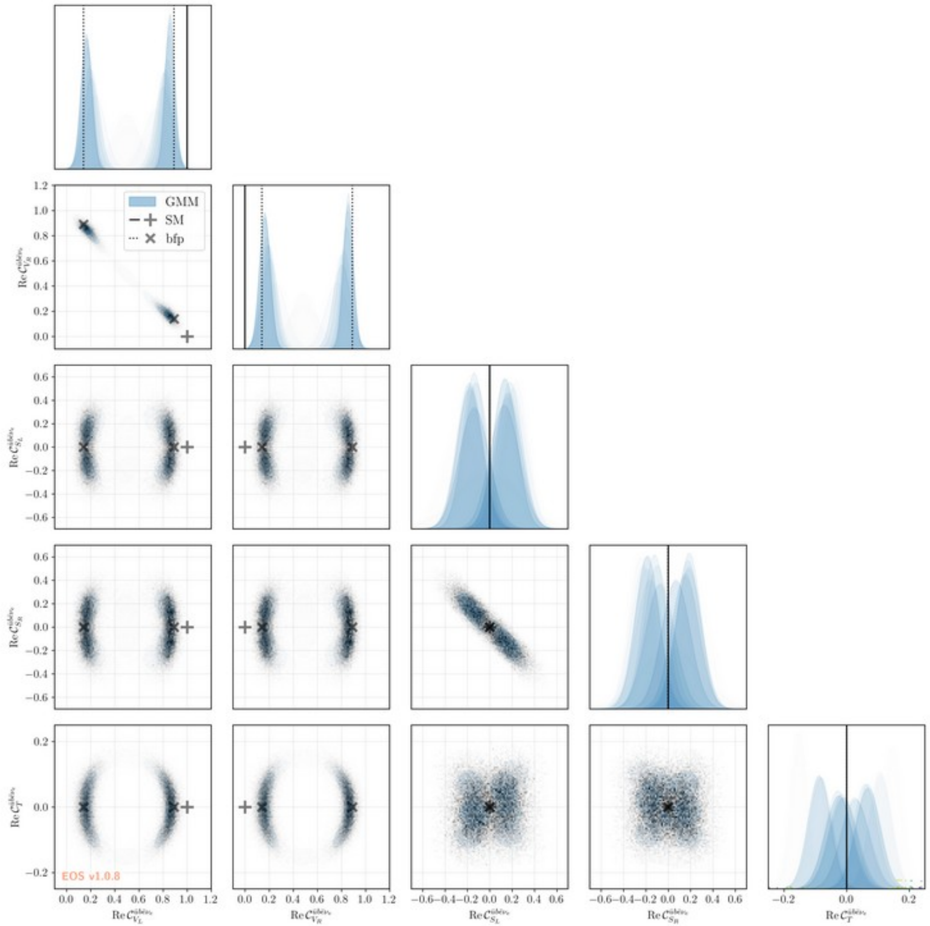
Gaussian Mixture Model

- Approximate the posterior distribution as a sum of normal distributions

Gaussian Mixture Model



Gaussian Mixture Model



Accessing through EOS

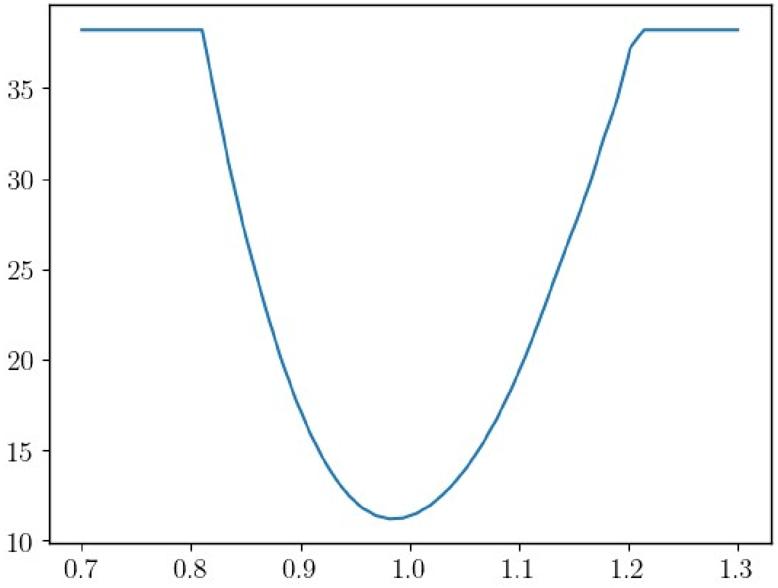
- https://github.com/eos/data/
- https://github.com/eos/data/tree/2023-01v2

Accessing through EOS

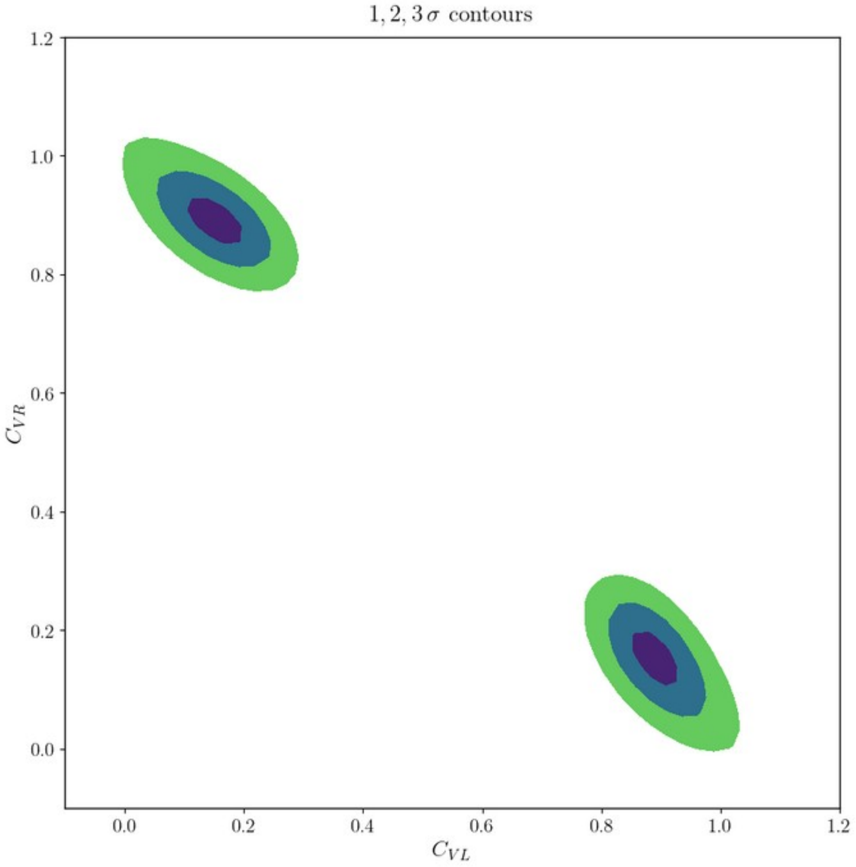
```python
import eos
import numpy as np
import matplotlib.pyplot as plt

ds = eos.DataSets()

for d in ds.datasets():
    print(d)

params, llh, chi2_func = ds.likelihood("2023-01v2", "WET-ublnu")

x_vals = np.linspace(0.7, 1.3)
chi2_data = np.zeros_like(x_vals)
for i, c in enumerate(x_vals):
    wc = np.array((c, 0.0, 0.0, 0.0, 0.0))
    chi2_data[i] = chi2_func(llh(wc))
plt.plot(x_vals, chi2_data)
```

Accessing through EOS

```
import eos
import numpy as np
import matplotlib.pyplot as plt

ds = eos.DataSets()

for d in ds.datasets():
    print(d)

params, llh, chi2_func = ds.likelihood("20

x_vals = np.linspace(0.7, 1.3)
chi2_data = np.zeros_like(x_vals)
for i, c in enumerate(x_vals):
    wc = np.array((c, 0.0, 0.0, 0.0, 0.0))
    chi2_data[i] = chi2_func(llh(wc))
plt.plot(x_vals, chi2_data)
```



Accessing through EOS



Going beyond GMMs

- Nabu: very much a WIP
  - A code to use machine learning to learn complicated (e.g non-gaussian) likelihoods.
- Once complete, EOS will use it to make likelihoods available
- Watch this (eoshep.org) space

EOS

- More information? See https://eoshep.org/
- Development on Github @ https://github.com/eos/eos
  - Very happy to accept new contributions
- More questions? Or want to ask about what EOS could do for you? Join the Discord server: https://discord.gg/hyPu7f7K6W