

Faster (SM)EFT fits with a large number of free parameters

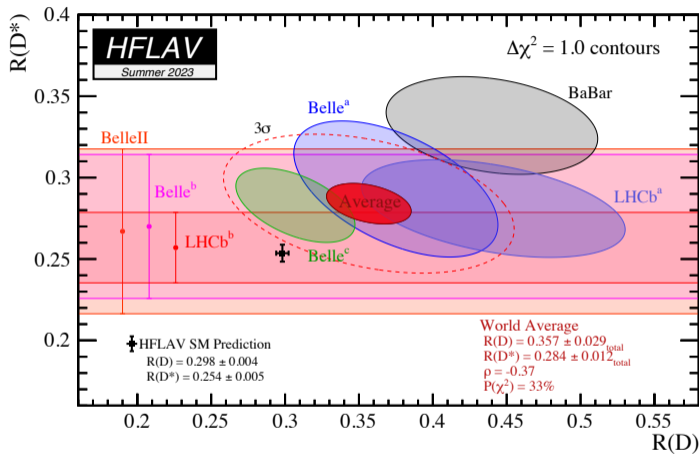
Philip Bechtle



color meets flavor

20. March 2024

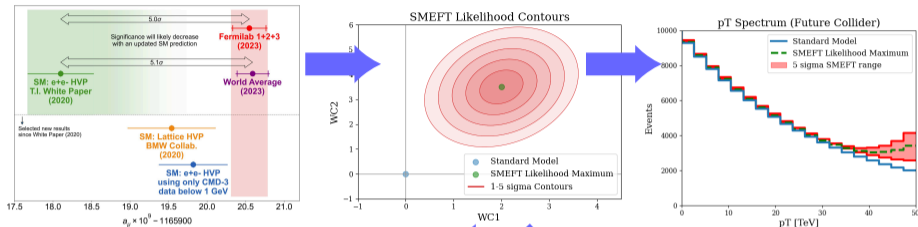
The Quest for a No-Lose-Theorem at Future Colliders



Some measurements at *low* energies don't fit perfectly to the Standard Model...

The Quest for a No-Lose-Theorem at Future Colliders

Illustration of the logic:



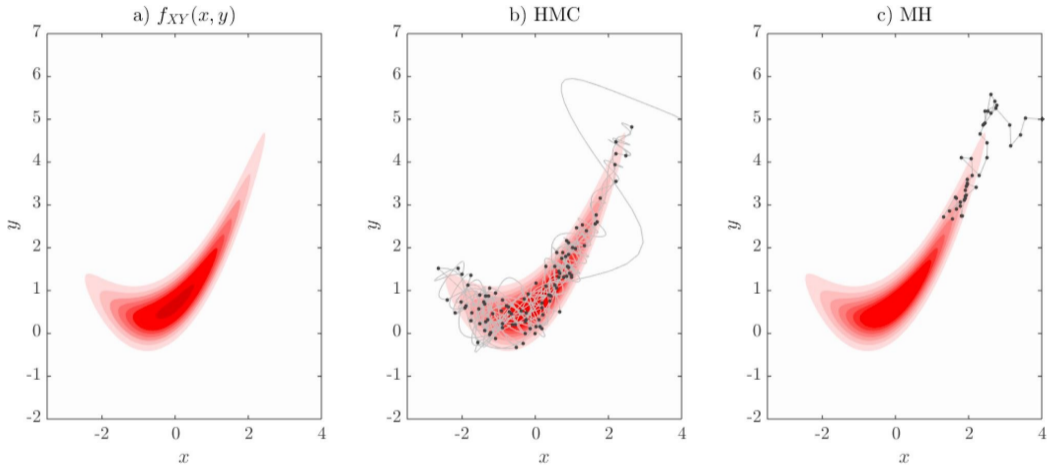
The $g - 2$ Scientific Seminar (2023)
[muong-2.2023]

Measurements done so far
(e.g. $(g_\mu - 2)$) restrict the
SMEFT parameter space

Presence of non-zero Wilson
Coef. (might) affect kinematic
distributions in future colliders

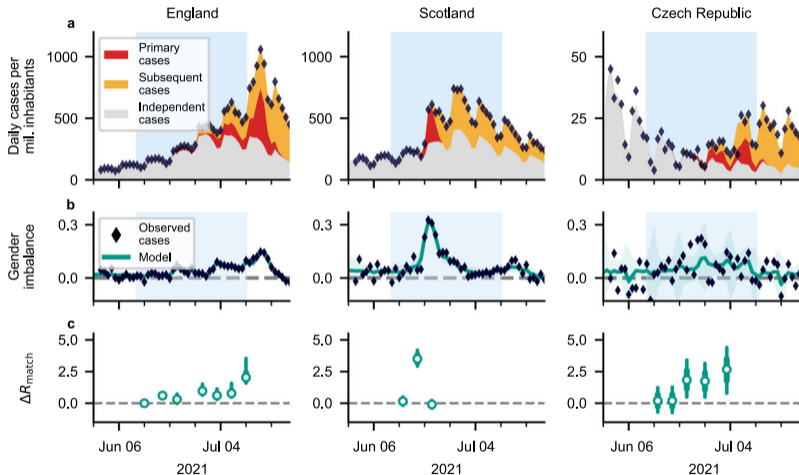
Can we use them to predict which discoveries a new (expensive) *high* energy collider should see?

Scanning Large Parameter Spaces



Would like to use Hamiltonian Monte Carlo

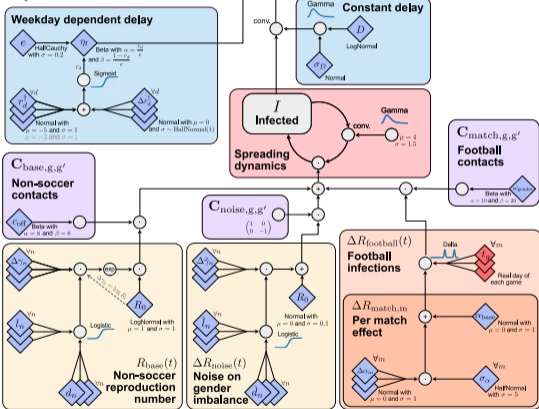
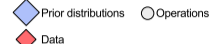
Scanning Large Parameter Spaces



We are using HMC as a community (invented in Lattice QCD) and even myself

Scanning Large Parameter Spaces

Model Overview



docs stable docs dev license MIT CI passing codecov 53% DOI 10.5281/zenodo.2587213

Welcome to BAT, a Bayesian analysis toolkit in Julia.

BAT.jl offers a variety of posterior sampling, mode estimation and integration algorithms, supplemented by plotting recipes and I/O functionality.

BAT.jl originated as a rewrite/redesign of [BAT](#), the Bayesian Analysis Toolkit in C++. BAT.jl now offer a different set of functionality and a wider variety of algorithms than its C++ predecessor.

We are successfully scanning parameter spaces of 250-500 parameters in an afternoon

Essential for Scanning Large Parameter Spaces in an Afternoon

Enzyme Overview

The Enzyme project is a tool that takes arbitrary existing code as LLVM IR and computes the derivative (and gradient) of that function. This allows developers to use Enzyme to automatically create gradients of their source code without much additional work. By working at the LLVM level Enzyme is able to differentiate programs in a variety of languages (C, C++, Swift, Julia, Rust, Fortran, TensorFlow, etc) in a single tool and achieve high performance by integrating with LLVM's optimization pipeline.

```
#include <stdio.h>

double square(double x) {
    return x * x;
}

double __enzyme_autodiff(void*, double);
int main() {
    double x = 3.14;
    // Evaluates to 2 * x = 6.28
    double grad_x = __enzyme_autodiff((void*)square, x);
    printf("square'(%f) = %f\n", x, grad_x);
}
```

By differentiating code after optimization, Enzyme is able to create substantially faster derivatives than existing tools that differentiate programs before optimization.

Diverse Tools for HEP

🔗 **smelli** – a global likelihood for precision constraints

`smelli` is a Python package providing a global likelihood function in the space of dimension-six Wilson coefficients in the Standard Model Effective Field Theory (SMEFT). The likelihood includes contributions from quark and lepton flavour physics, electroweak precision tests, and other precision observables.

The package is based on [flavio](#) for the calculation of observables and statistical treatment and [wilson](#) for the running, translation, and matching of Wilson coefficients.

Installation

The package requires Python version 3.6 or above. It can be installed with



```
python3 -m pip install smelli --user
```













(Compiled) Python Code

Diverse Tools for HEP

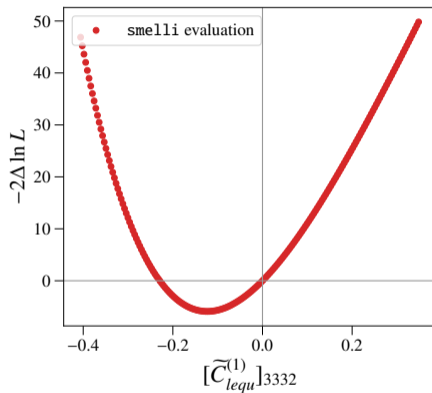
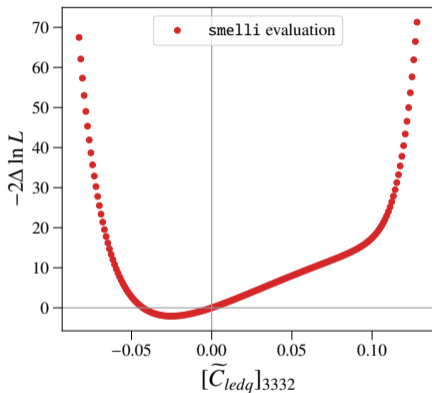
mg5amcnlo / MadSpin / src / 

 **Rikkert Frederix** Fixed a bug in the phase-space generation for MadSpin (relevant only) 

Name	Last commit message
 ..	
 driver.f	Fixed a bug in the phase-space generation for MadSpin (relevant only)
 driver_decay.f	first running version with PS generation in fortran
 driver_prod.f	first running version with PS generation in fortran
 initialize.f	1. converted many of the INFO to DEBUG messages
 lha_read_ms.f	forgot a log check in madspin
 makefile_decay	first running version with PS generation in fortran
 makefile_full	the seed in the madspin_card.dat is now passed properly to the fortr...
 makefile_prod	first running version with PS generation in fortran
 ranmar.f	Fortran executable is unable to write down the state of ranmar, mayb...

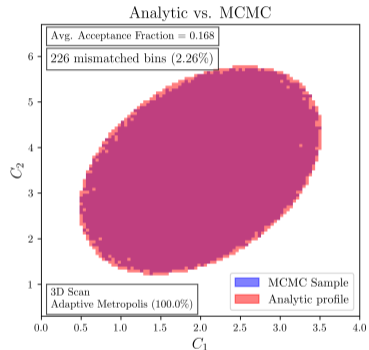
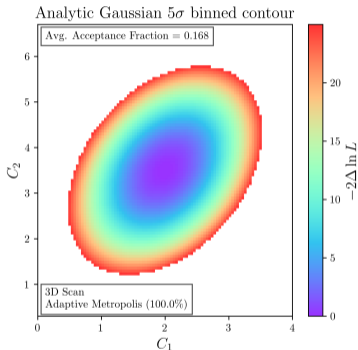
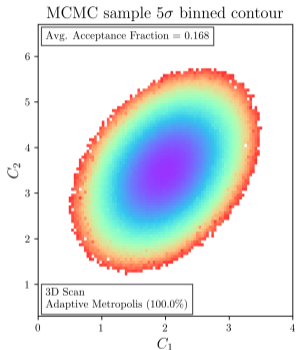
Fortran Code, C(++), ... mostly written by diverse (only loosely connected) teams

Need very high density scans for high significance (5σ)



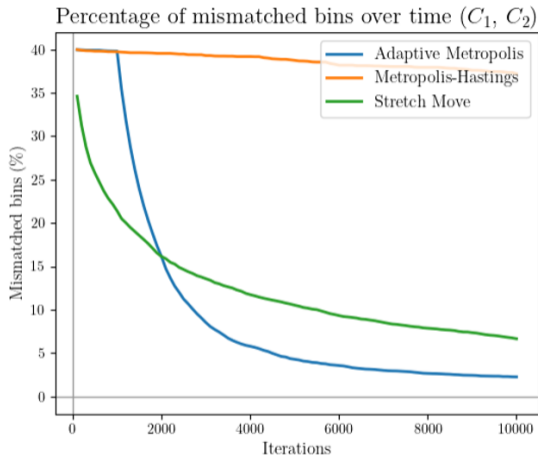
Non-Gaussian Statistics

Need very high density scans for high significance (5σ)



Need dense coverage

Need very high density scans for high significance (5σ)



Adaptive Markov Chain much faster than other algorithms, but not fast enough

Completely Exaggerated Wishlist

- ▶ Use any code written by anybody in any language
- ▶ Differentiate this code
- ▶ Completely crazy question: Could we even differentiate through a random random number generator?
- ▶ ...
- ▶ Of course we're happy for every more reasonable proposal of how to speed up these scans