BNL NPP Retreat for Postdoctoral Research Associates

Christian Weber

August 25th, 2023

Brookhaven National Laboratory

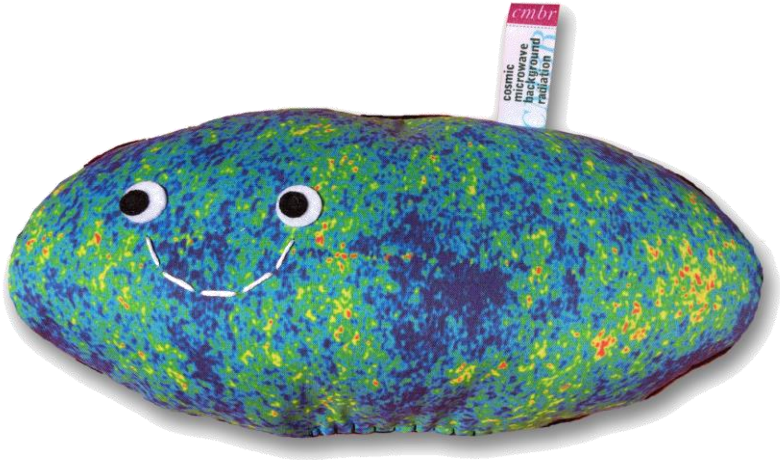
U.S. DEPARTMENT OF ENERGY

@BrookhavenLab

The Matter Universe

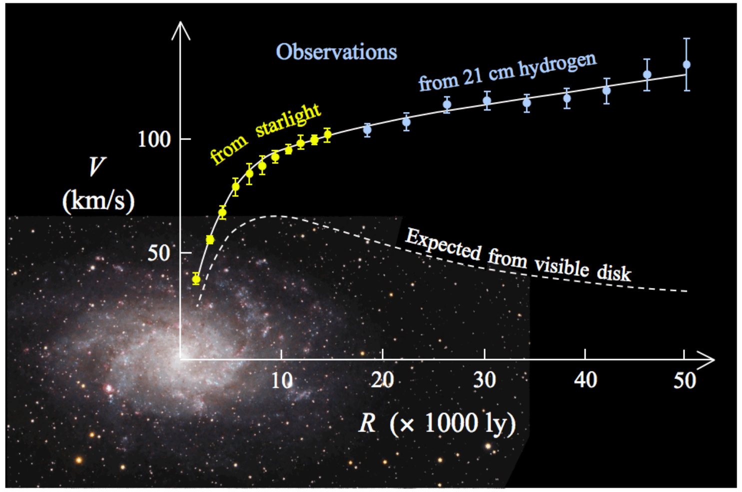
■ 14% Regular Matter
■ 86% Dark Matter

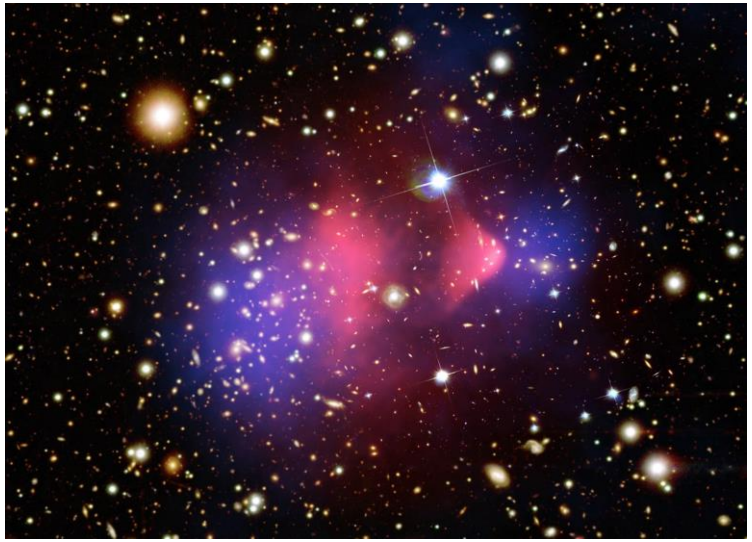
M. Tanabashi *et al.* (Particle Data Group), Phys. Rev. D **98**, 030001 (2018)



Cosmic Microwave Background

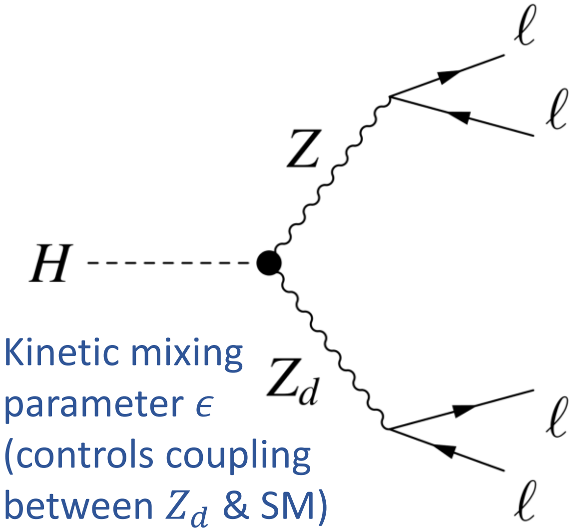
M 33 rotation curve



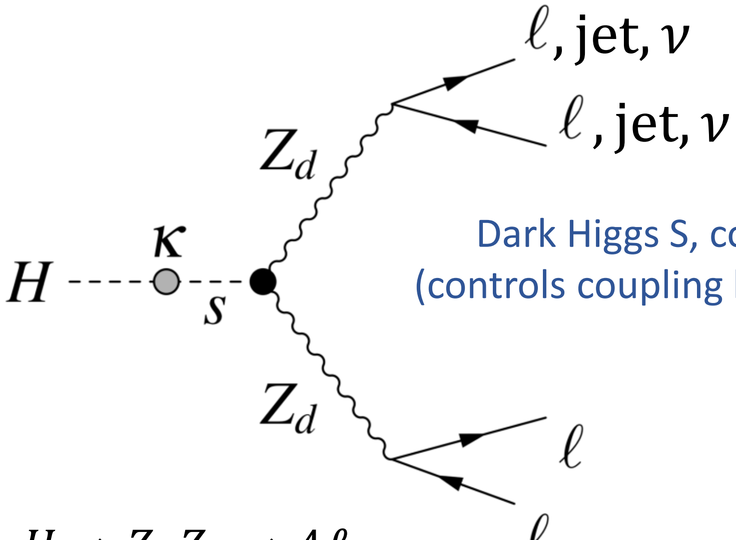


Bullet Cluster

Dark scalar S and dark vector boson Z_d searches

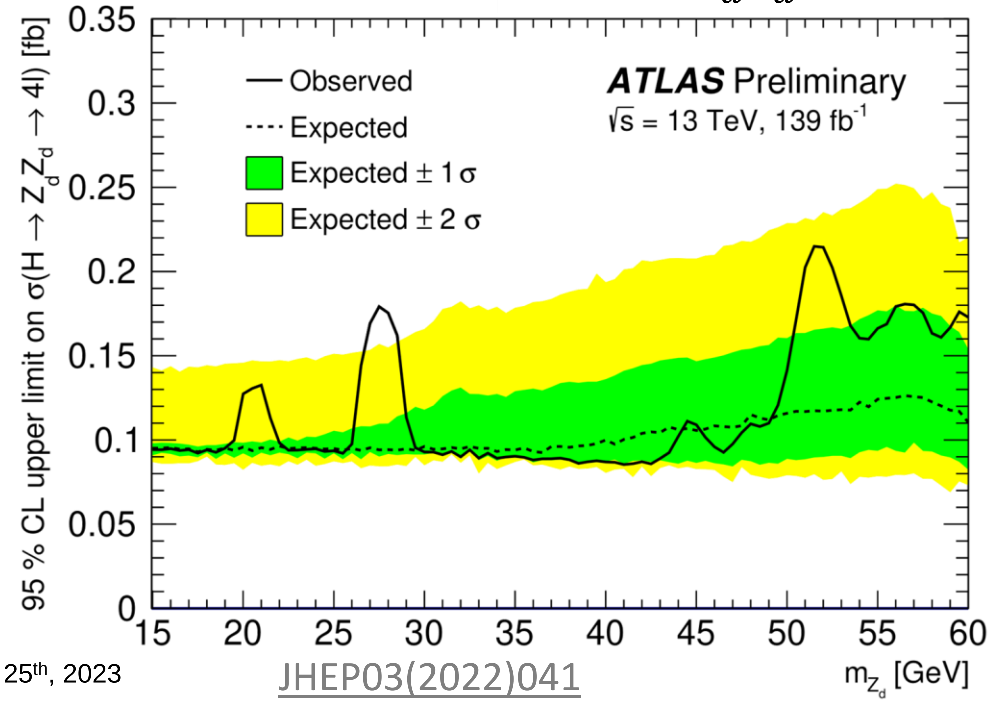


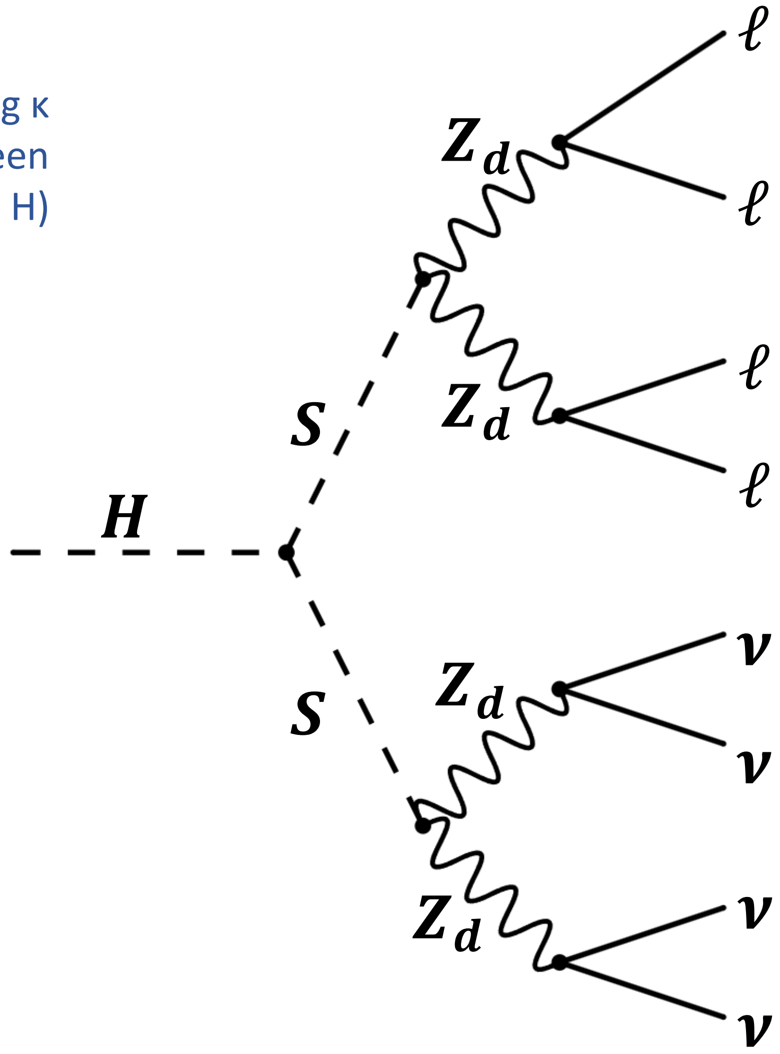
Kinetic mixing parameter ϵ (controls coupling between Z_d & SM)



Dark Higgs S, coupling κ (controls coupling between S and H)

$H \to Z_d Z_d \to 4\ell$

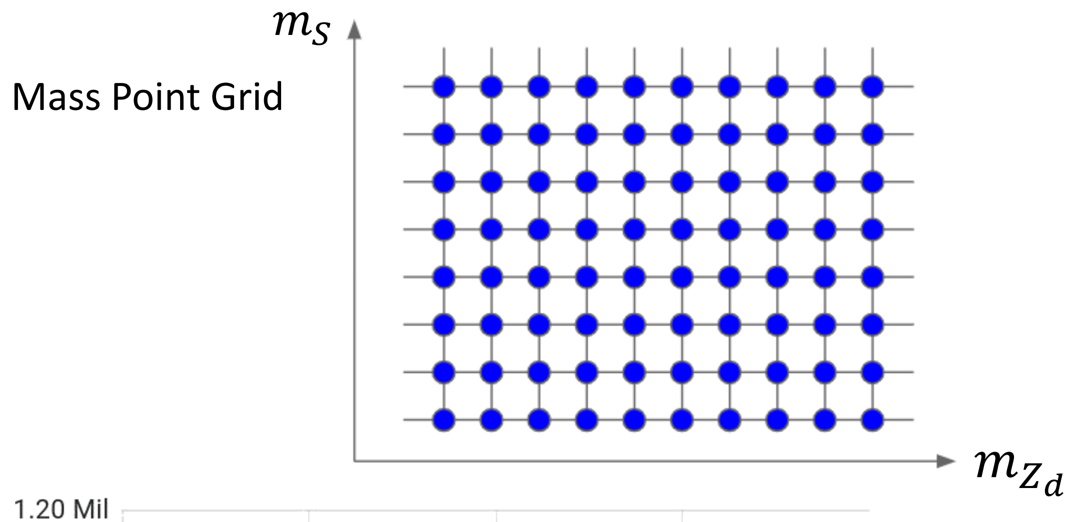


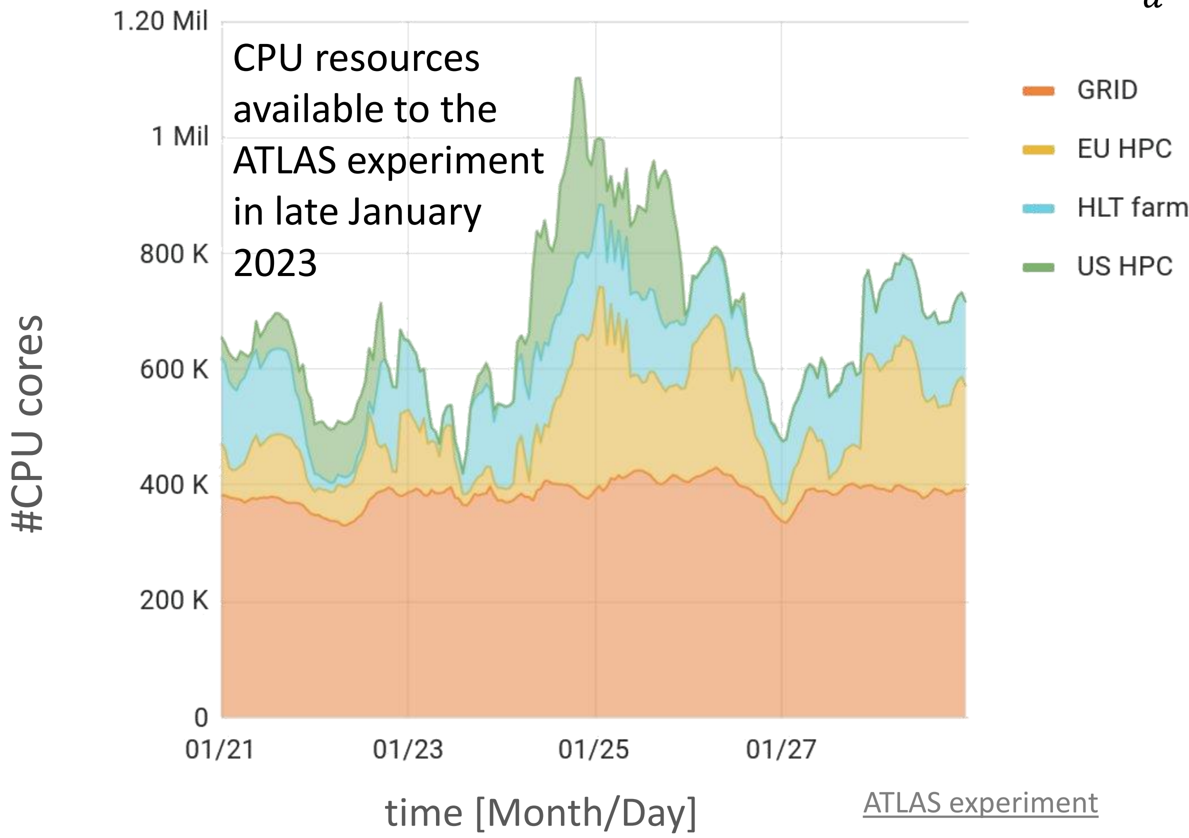


Monte Carlo Simulation

Monte Carlo (MC) Event simulation is very computationally intensive

- $\sim$15h calculation time for 100 simulated events on a single CPU core
- Might need around 10^5 events or more per signal sample
- Using a grid of 10 values in two dimensions would require 10^7 events, about $\sim$2k CPU months
- Estimate is for fast calorimeter simulation, full calorimeter simulation requires even more computational resources

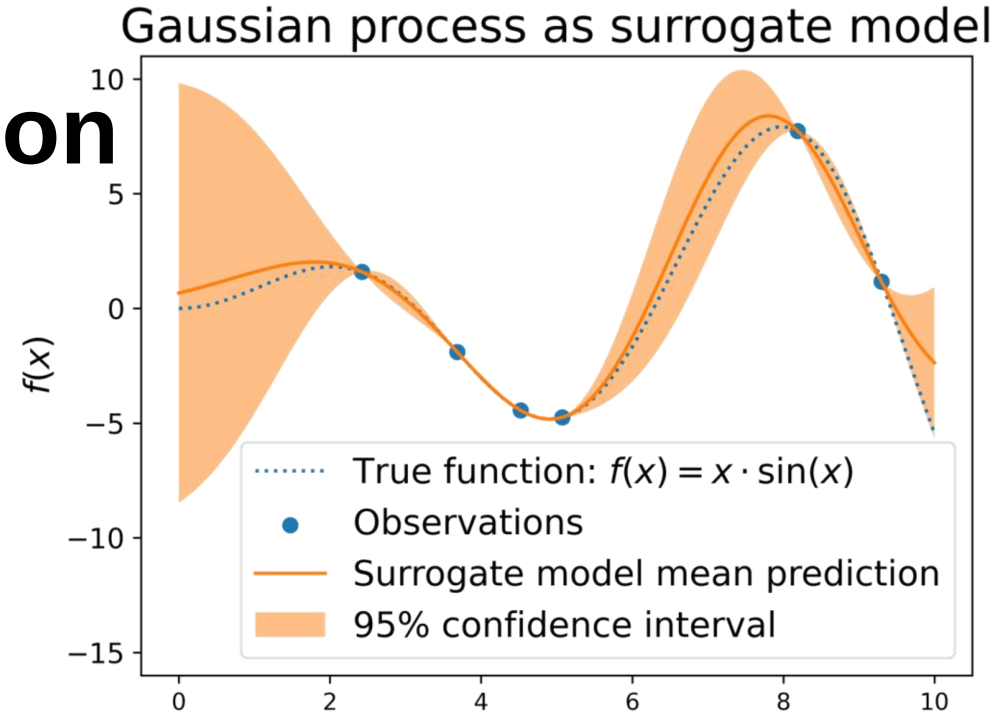




ATLAS experiment

Active Learning for reinterpretation

An iterative process to collect new labelled data for optimization tasks



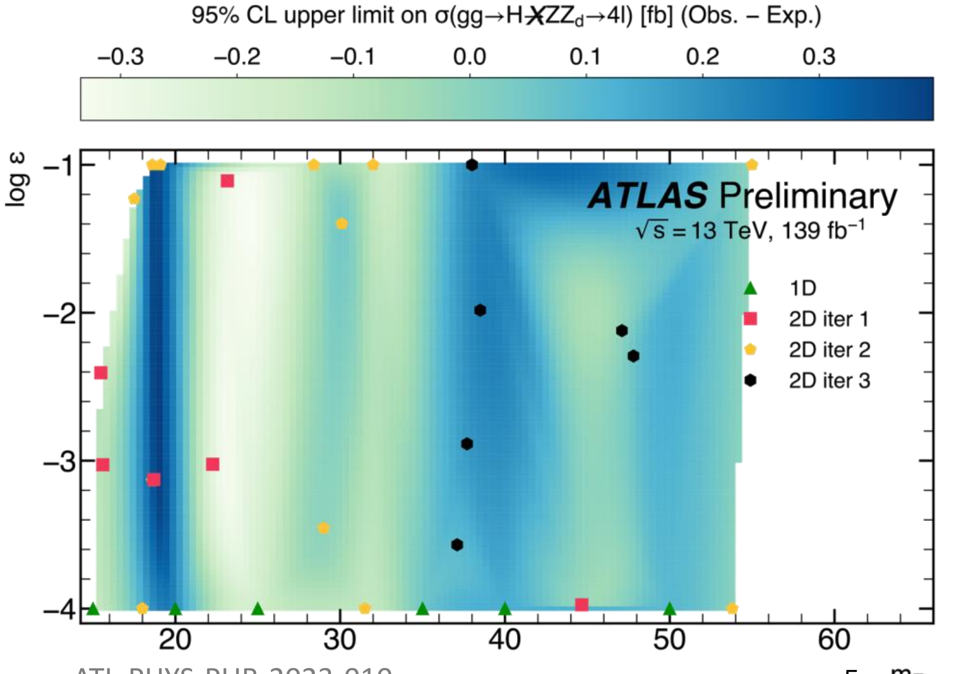
We developed a system to work around the computational constraints by using an Active Learning approach that includes

- Surrogate model that approximates the function mapping physics parameters onto exclusion limits

$$m_S, m_{Z_d}, \Gamma_{Z_d}, \epsilon, \kappa \rightarrow \sigma_{m_S, m_{Z_d}, \Gamma_{Z_d}, \epsilon, \kappa}$$

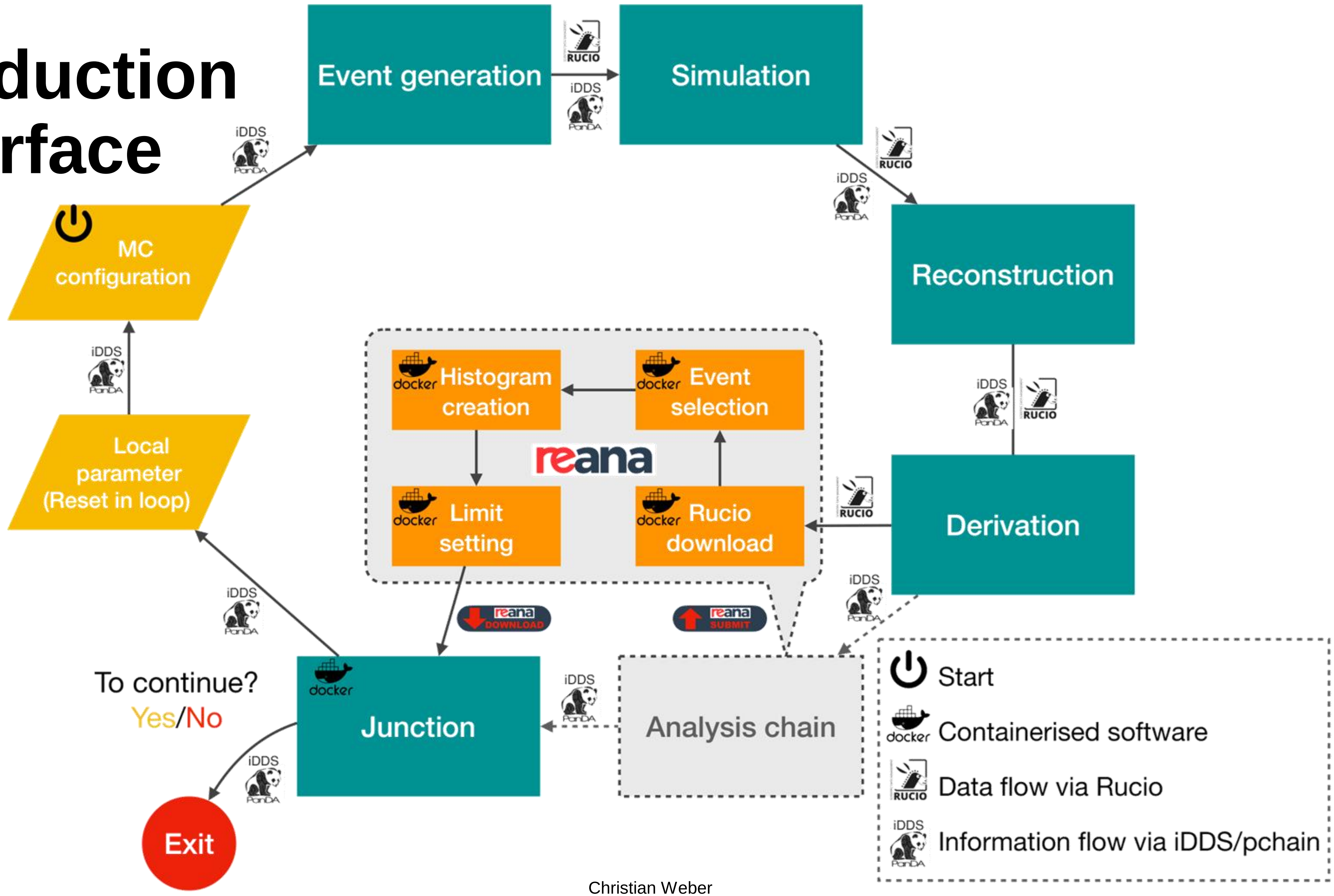
- An acquisition function that determines which points in the physics parameter space, e.g. $m_S, m_{Z_d}, \Gamma_{Z_d}, \epsilon, \kappa$, to explore next based on the surrogate model

- A software stack using ATLAS Grid tools (PanDA and the iDDS) to automate the iterative Monte Carlo sample production, their analysis, and the use of the analysis results in the active learning step



ATL-PHYS-PUB-2023-010

Production Interface



Large Language Models for High energy Physics

Generate algorithm to match reconstructed jets to quarks.

Prompt and response:

ZH

I do have two c++ objects, quark1 and quark2, and a c++ vector called myJets, whose elements are called jet. For each element of myJets and quark1 or quark2 I can calculate a metric deltaR: deltaR(quark1, jet) and deltaR(quark2, jet). How can I find the two jets that minimize the metric deltaR for each, quark1 and quark2.

You can find the two jets in the `myJets` vector that minimize the metric deltaR for each `quark1` and `quark2` by iterating over each element in `myJets` and computing the deltaR for each `quark`. Here is an example code snippet that demonstrates how you can do this:

c++ Copy code

```cpp
#include <iostream>
#include <vector>
```

Code output continued:

c++

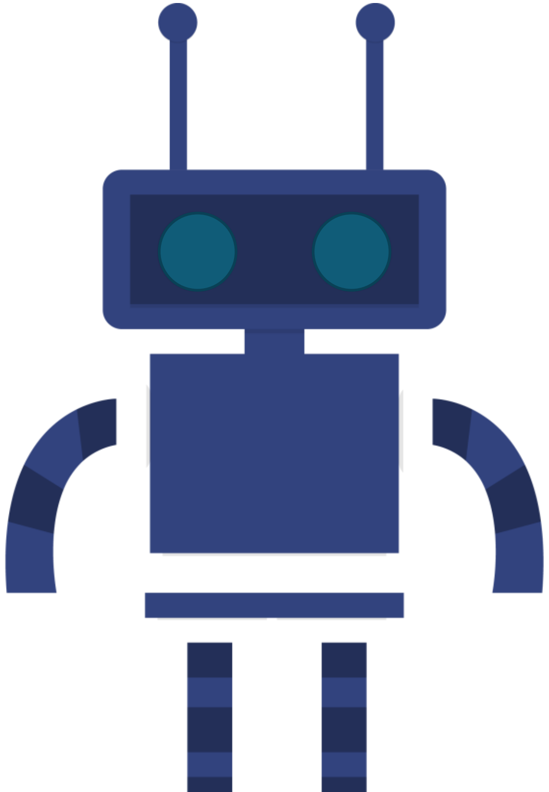
```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;

// Define a struct to hold the jet and its corresponding deltaR
struct JetDeltaR {
    double deltaR;
    int jetIndex;
};

// Define a function to calculate the deltaR between two objects
double calculateDeltaR(double eta1, double phi1, double eta2, doubl
    double dEta = eta1 - eta2;
    double dPhi = abs(phi1 - phi2);
    if (dPhi > M_PI) {
        dPhi = 2 * M_PI - dPhi;
    }
    return sqrt(dEta * dEta + dPhi * dPhi);
}

int main() {
    // Define the quarks and jets
    double quark1Eta = 1.0;
    double quark1Phi = 2.0;
    double quark2Eta = -1.0;
    double quark2Phi = 3.0;
```

ATLAS GPT

Large Language Models like ChatGPT can already provide utility for work in High Energy Physics. Particularly for less experienced collaborators.

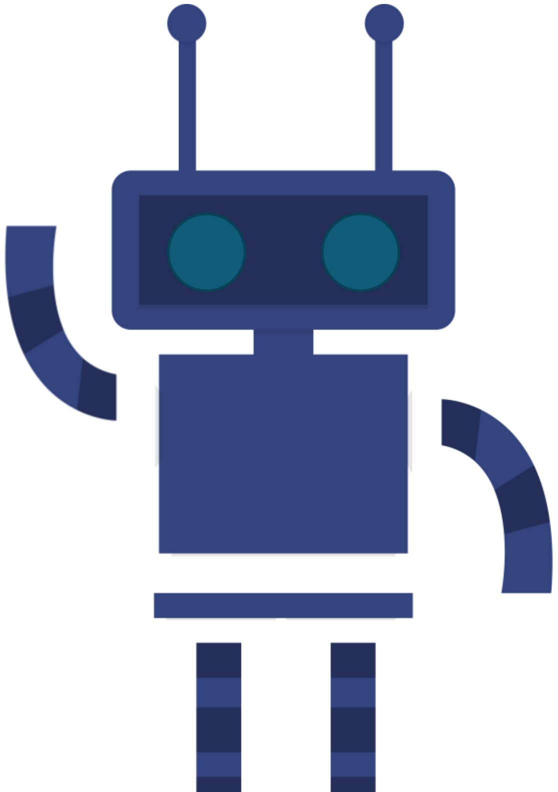
Ideally, we would like something that can also help more advanced collaborators, e.g.:

- Inform us about the current jet recommendations for ATLAS analyses
- Tell me how electrons are calibrated in my very specific analysis framework.
- Help with debugging
- Summarize research papers, documentation, etc.



Recently joined ATLAS GPT efforts
- Works by prompt engineering existing LLM, e.g. ChatGPT
- See atlasgpt.docs.cern.ch

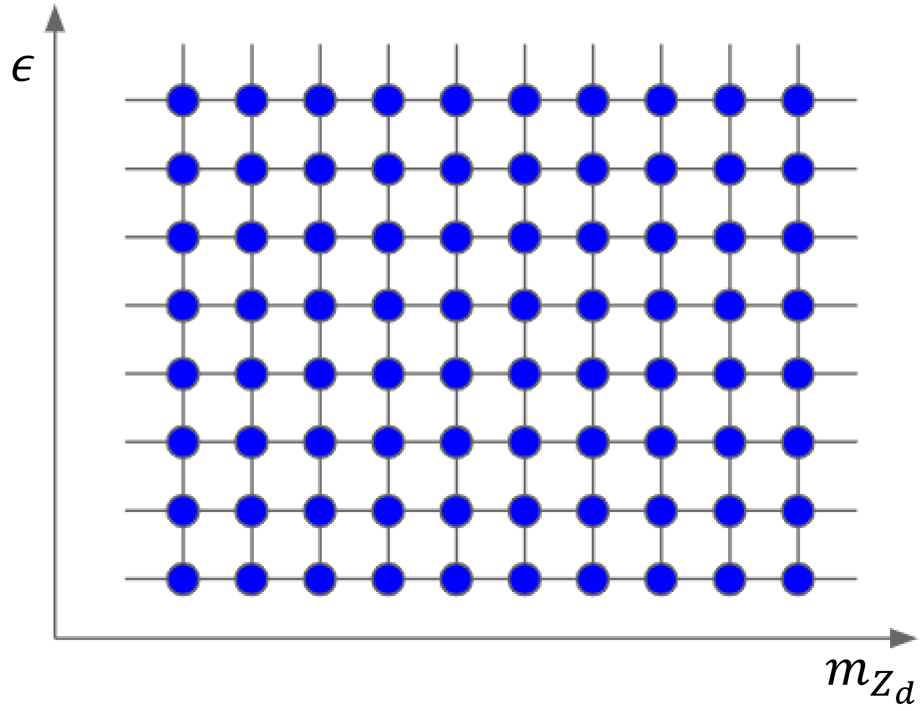
Thank you!



Limits in multidimensional parameter space

Process for calculation of limits in phase space conceptionally simple, e.g., do a ‘grid search’

- Select boundaries in m_S, m_{Z_d}, Γ_{Z_d}, ϵ, κ
- Generate grid of signal samples
- And calculate cross section limits for each sample
- Bonus: with theory predictions one can infer contours between excluded and not excluded regions of parameter space



Problem:

- Generating Monte Carlo samples is computationally expensive