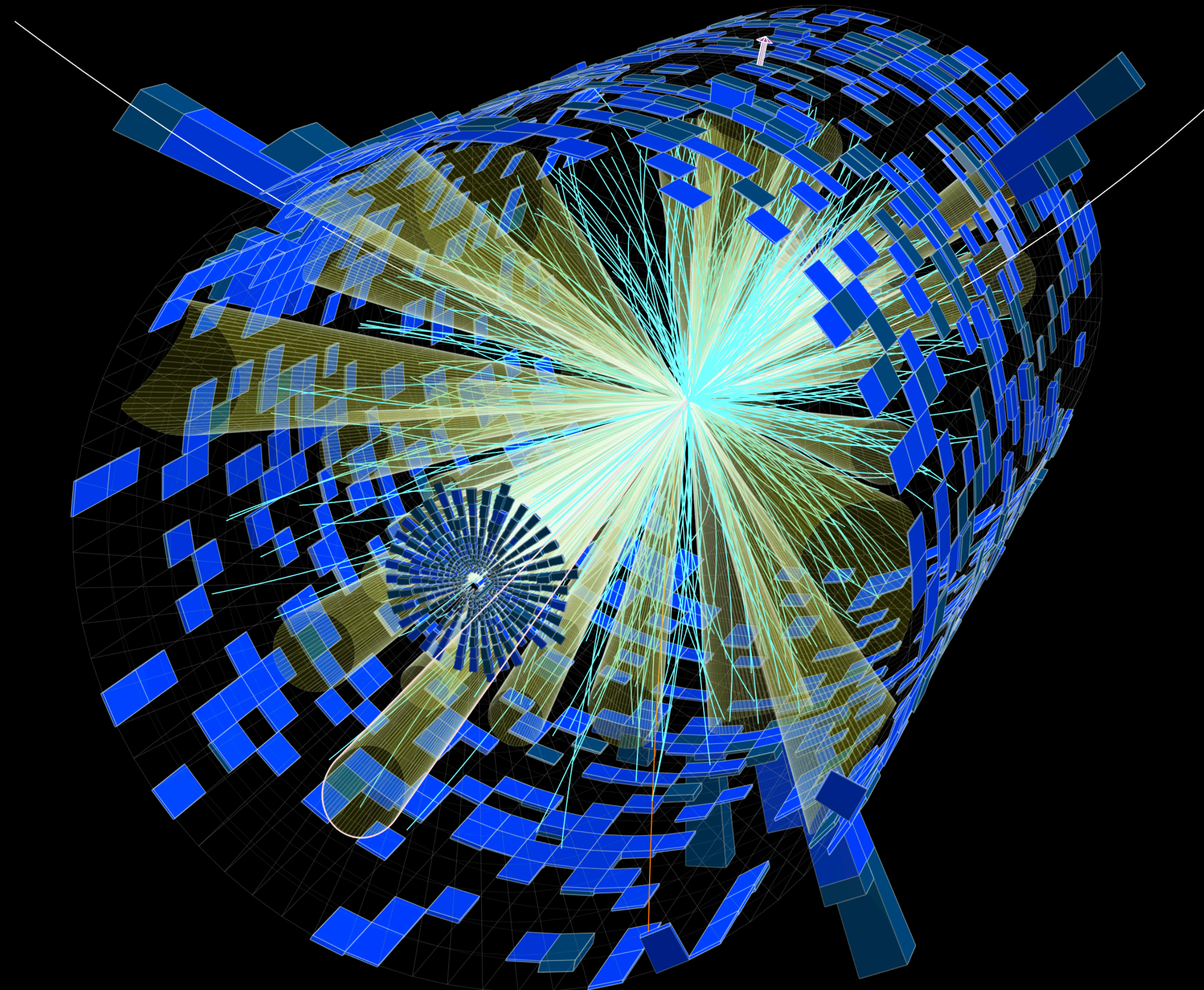


RADICALLY DIFFERENT FUTURES FOR HEP

ENABLED BY AI/ML

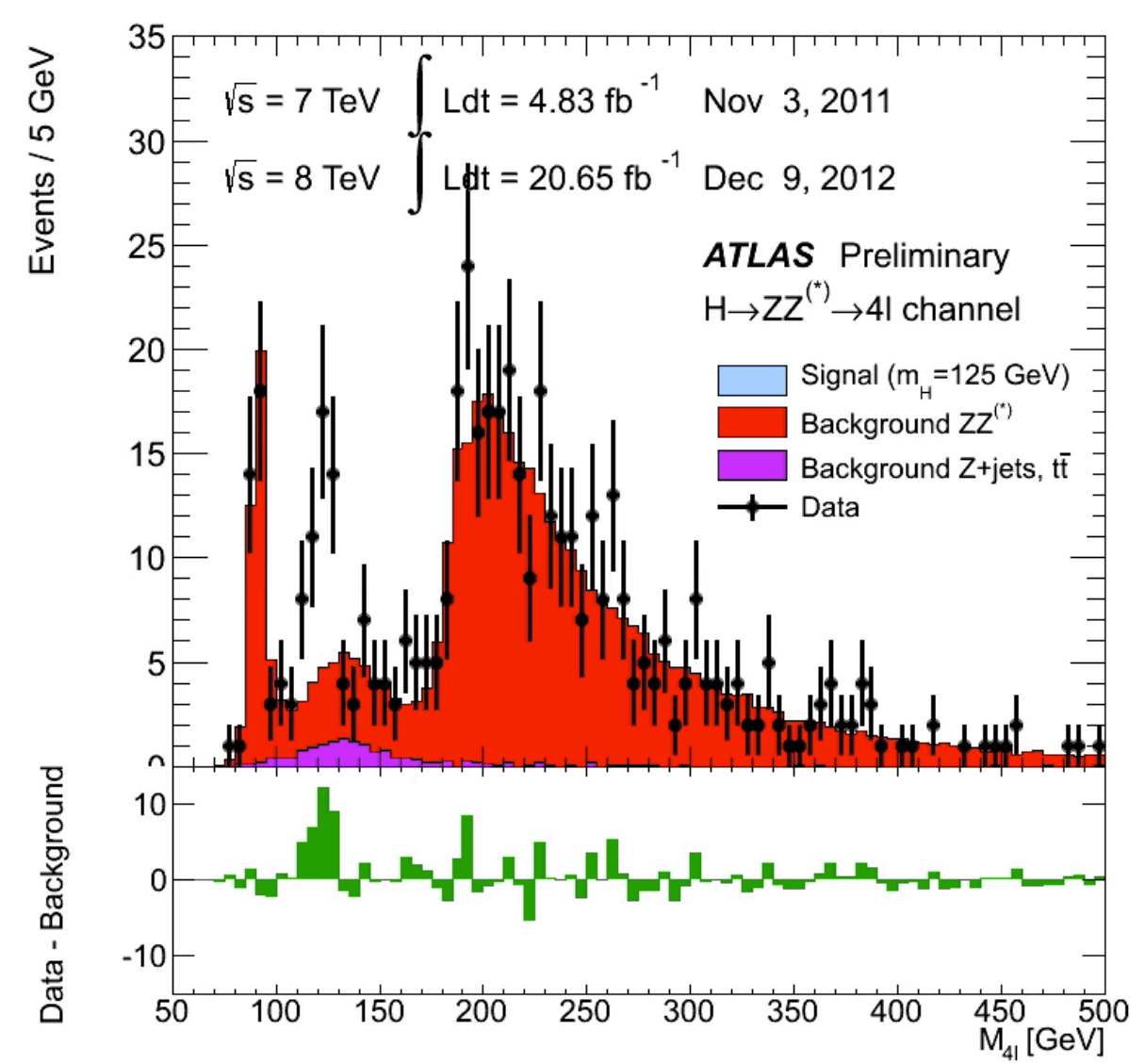
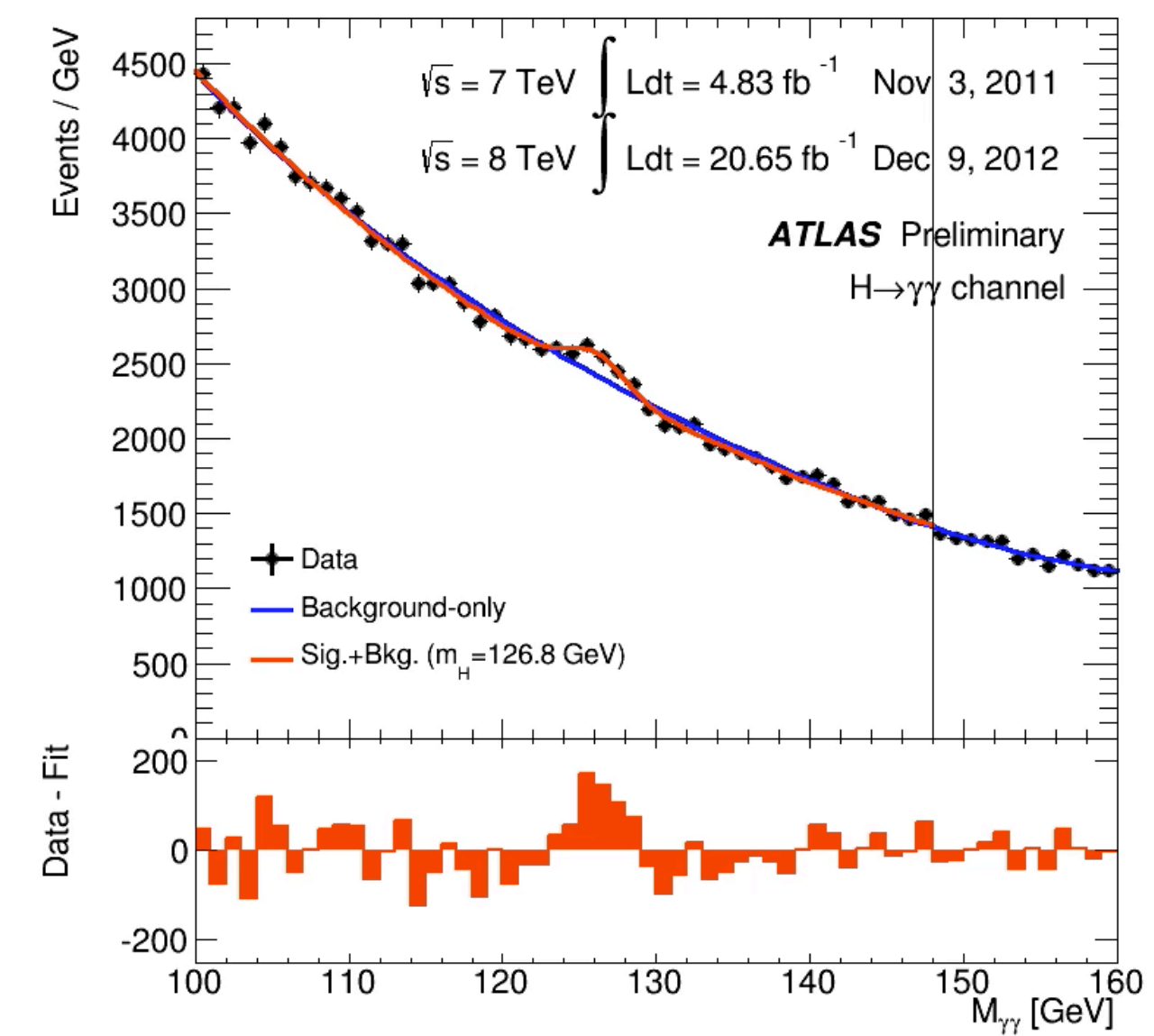


@KyleCranmer

University of Wisconsin-Madison

Data Science Institute, Physics, Statistics, Computer Science

July 2012





ImageNet Classification with Deep Convolutional Neural Networks

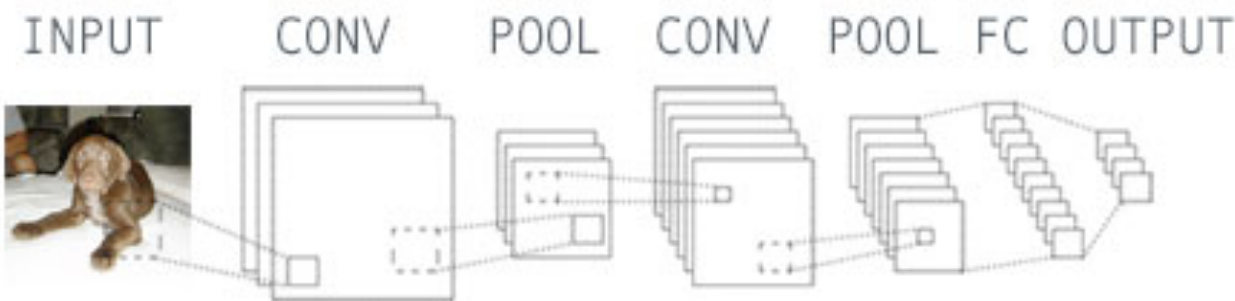
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

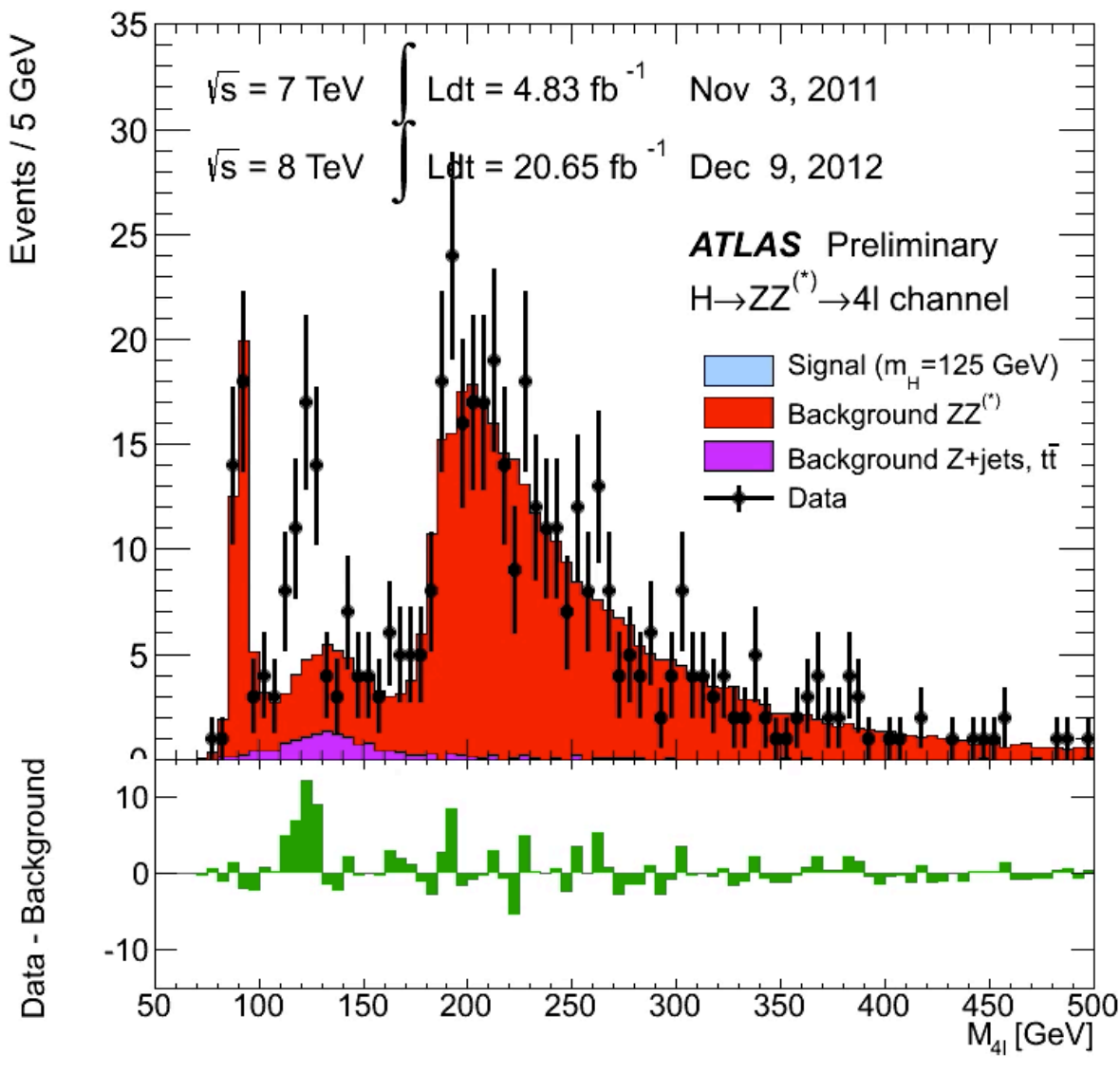
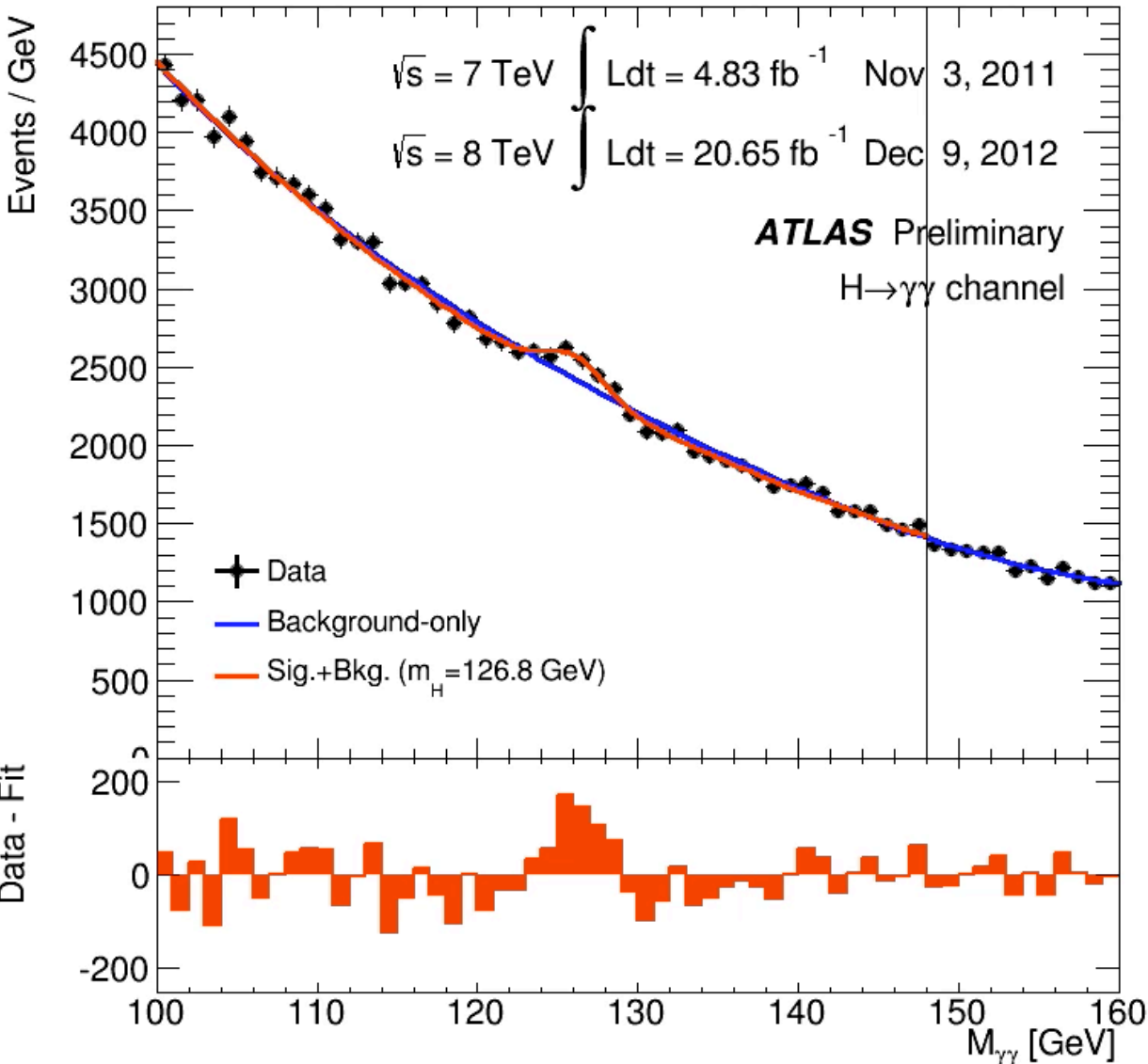
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.



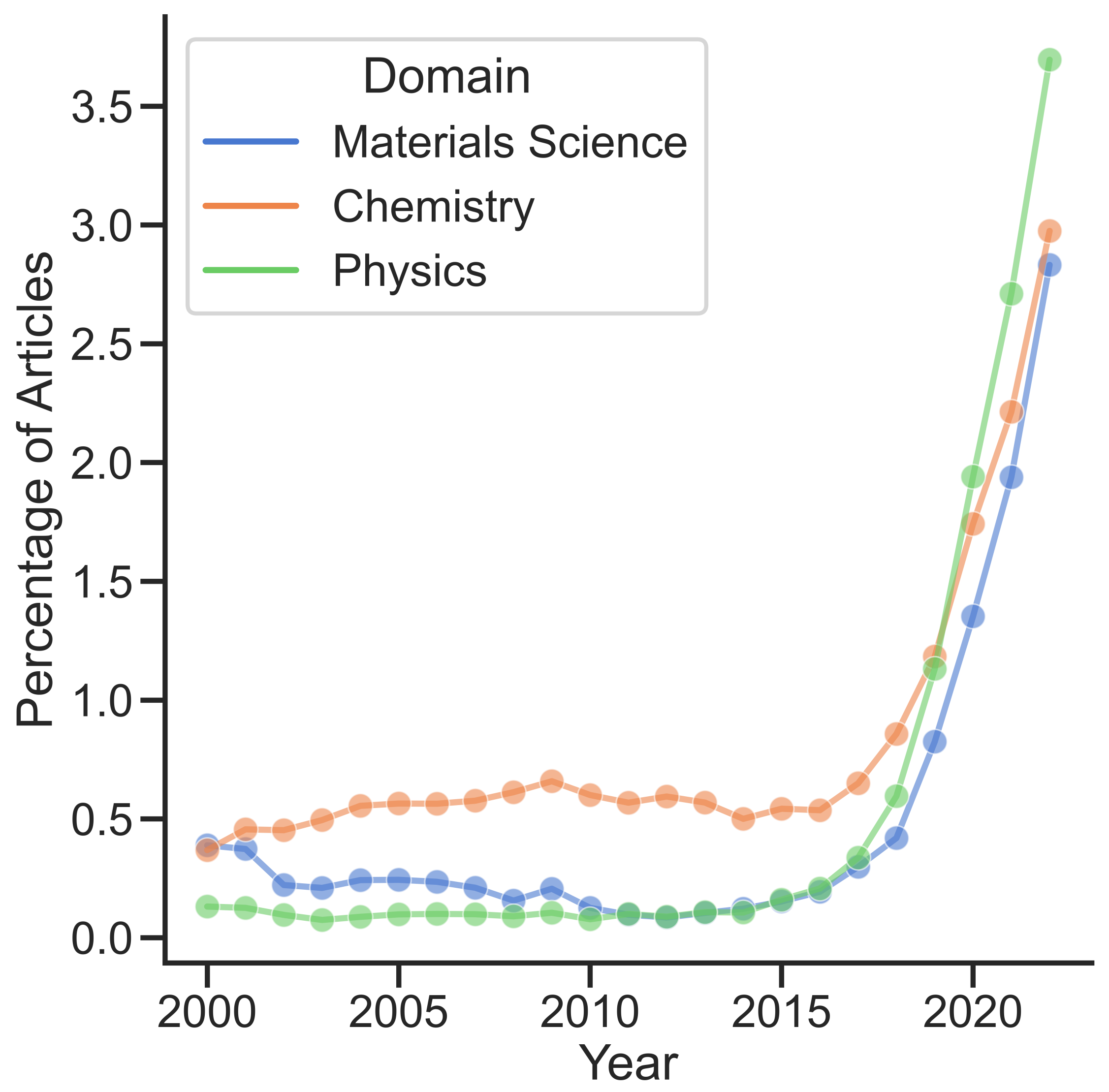
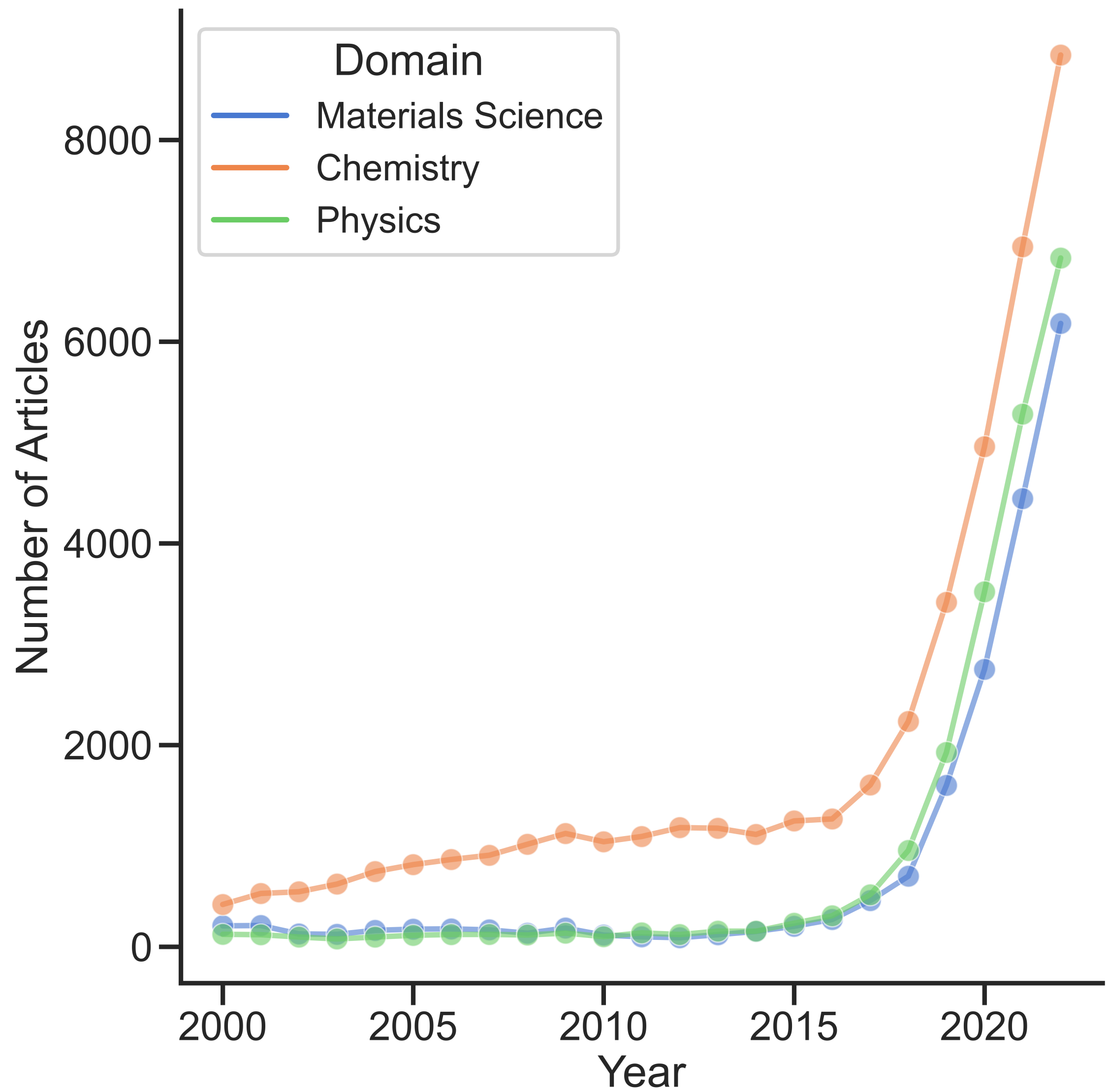
Dog:	94%
Cat:	31%
Bird:	2%
Boat:	0%



Dog:	37%
Cat:	91%
Bird:	21%
Boat:	1%



ML Publications in Science



“New directions in science are launched by new tools much more often than by new concepts. The effect of a concept-driven revolution is to explain old things in new ways. The effect of a tool-driven revolution is to discover new things that have to be explained.”

– FREEMAN DYSON



Abstract

Instead of focusing on the concrete challenges of incremental changes to HEP driven by AI/ML, it is perhaps a useful exercise to think through more radical, speculative changes.

- What might be enabled if we embraced a dramatically different approach?
- What would we lose?
- How would those changes impact the computational, organizational, and epistemological nature of the field?

What do I mean by “radically different” (for the purpose of this talk)?

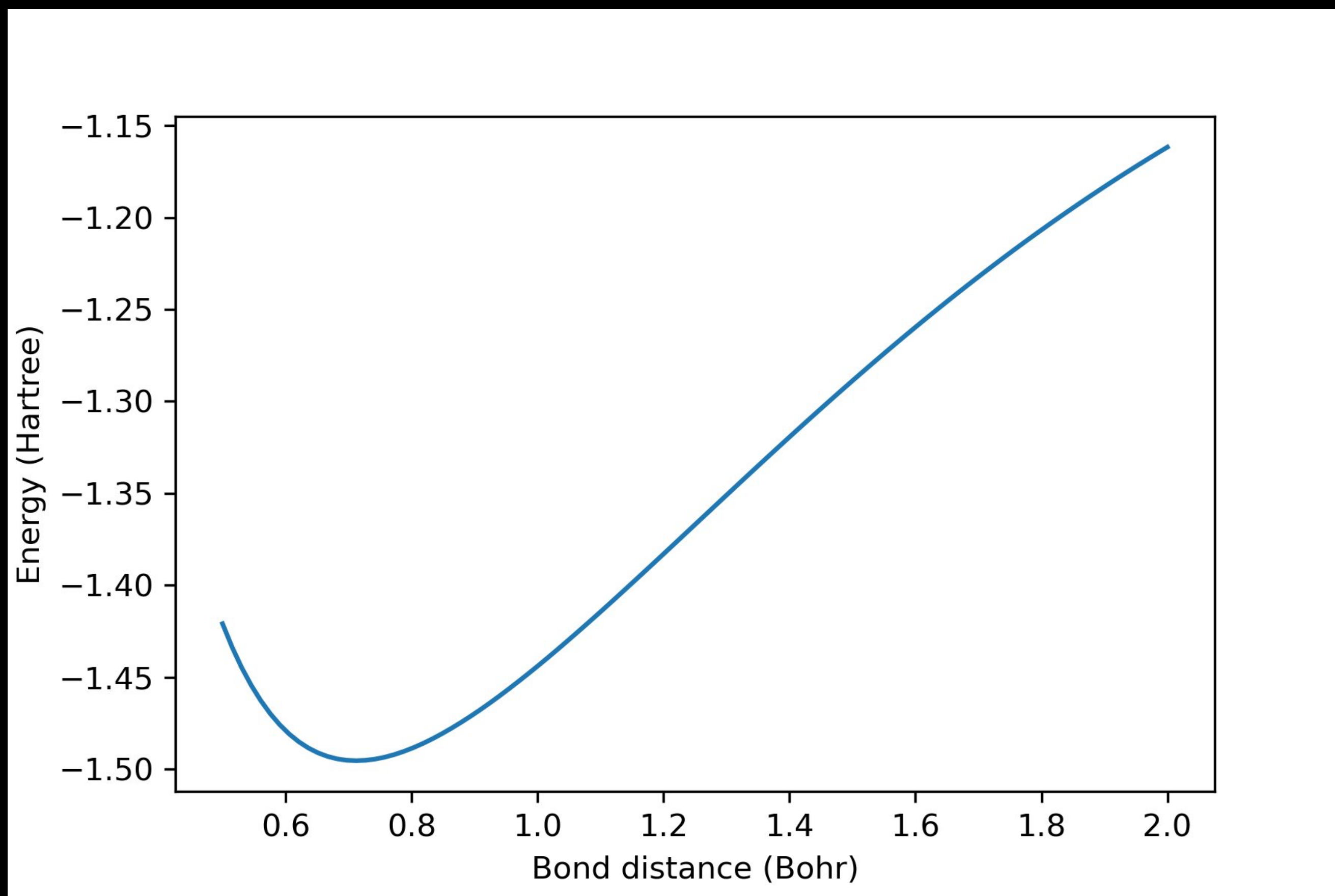
- Major reorganization of workflow or change to our tools.
- Some common components may not continue to exist (in the current form).

Code writing

AI writing code, using tools (written by humans)

Prompt:

"Compute the dissociation curve of H2 using pyscf."



```
----- PROPMT -----  
"""  
Compute the dissociation curve of H2 using pyscf  
"""  
----- END PROPMT -----  
import numpy as np  
import matplotlib.pyplot as plt  
from pyscf import gto, scf  
  
# Define the molecule  
mol = gto.M(atom='H 0 0 0; H 0 0 1.4', basis='sto-3g')  
  
# Perform the SCF calculation  
scf_eng = scf.RHF(mol)  
scf_eng.conv_tol = 1e-9  
scf_eng.conv_tol_grad = 1e-7  
scf_eng.kernel()  
  
# Plot the dissociation curve  
dissociation_curve = np.arange(0.5, 3.0, 0.1)  
energies = []  
for r in dissociation_curve:  
    mol = gto.M(atom='H 0 0 0; H 0 0 {}'.format(r), basis='sto-3g')  
    scf_eng = scf.RHF(mol)  
    scf_eng.conv_tol = 1e-9  
    scf_eng.conv_tol_grad = 1e-7  
    scf_eng.kernel()  
    energies.append(scf_eng.e_tot)  
  
plt.plot(dissociation_curve, energies)  
plt.xlabel('Length of bond (Angstrom)')  
plt.ylabel('Total energy (Hartree)')  
  
plt.savefig('h2_curve.pdf')  
plt.show()
```


AI-assisted code generation for HEP


Already useful in HEP I

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

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

Code output continued:

c++

```
#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, double phi2) {
    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;
```

It is ~guaranteed that AI is going to dramatically change how HEP physicists write code


- & Migrate, convert code to new platforms

Already useful in HEP IV

Generate ATLAS analysis template

ZH

Write me a template for an AthAnalysis based ATLAS analysis



Sure! Here's a basic template for an AthAnalysis-based ATLAS analysis:

c++

Copy code

```
#include "xAODRootAccess/Init.h"
#include "xAODRootAccess/TEvent.h"
#include "xAODRootAccess/TStore.h"
#include "xAODEventInfo/EventInfo.h"
#include "xAODJet/JetContainer.h"

#include <EventLoop/Job.h>
#include <EventLoop/StatusCode.h>
#include <EventLoop/Worker.h>

class MyAnalysis : public EL::AnaAlgorithm {
public:
    MyAnalysis() {}
```

```
private:
    xAOD::TEvent* m_event; ///
    xAOD::TStore* m_store; ///

    // Add any other member variables or functions as needed
};

EL::StatusCode MyAnalysis::initialize()
{
    // Initialize xAOD access
    xAOD::Init("MyAnalysis").ignore();

    // Get the input TEvent object
    m_event = wk()->xAODEvent();
    m_store = wk()->xAODStore();

    return EL::StatusCode::SUCCESS;
}

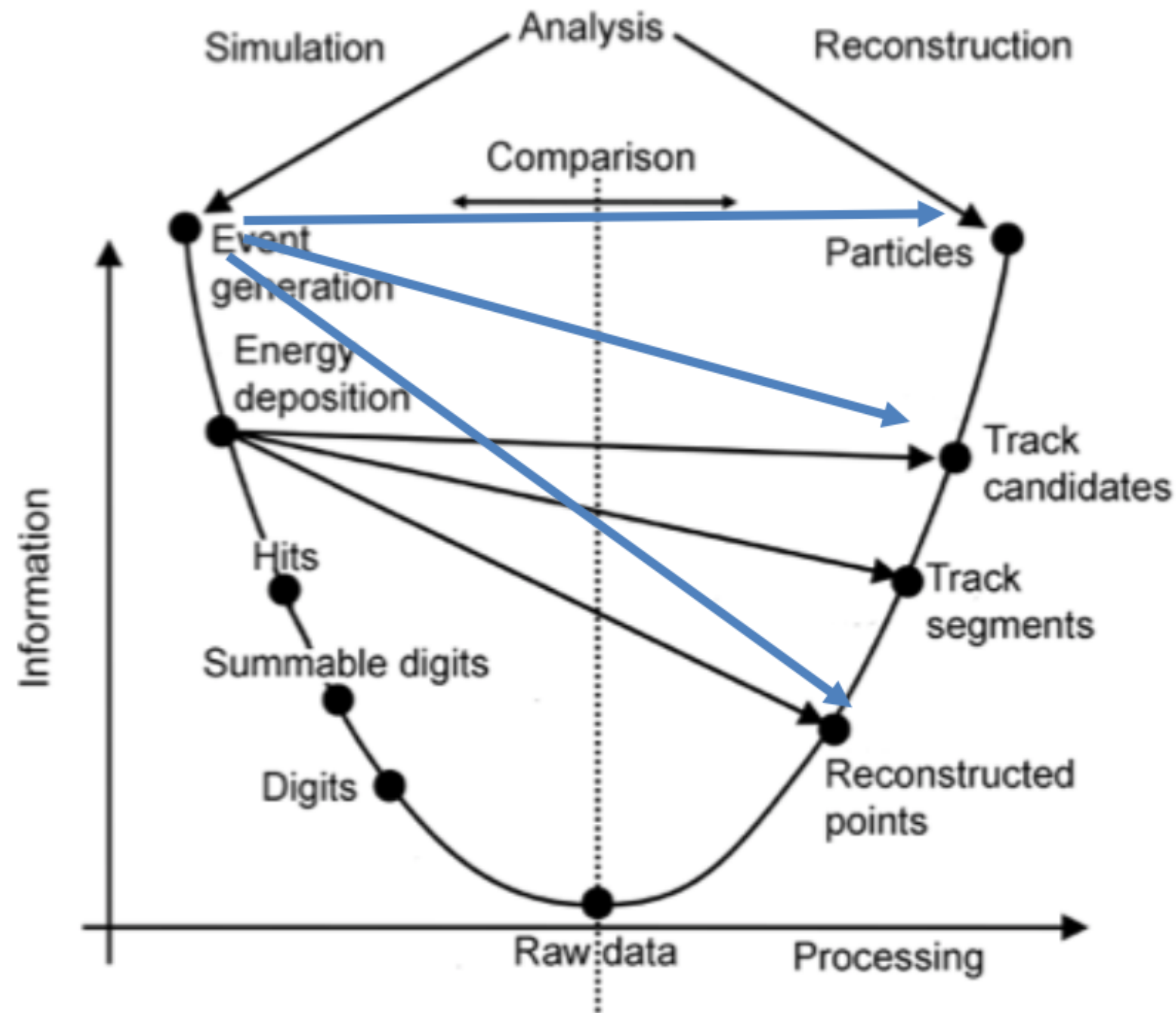
EL::StatusCode MyAnalysis::execute()
{
    // Get the event information
    const xAOD::EventInfo* evtInfo = 0;
    m_event->retrieve(evtInfo, "EventInfo").ignore();

    // Get the jets from the event
    const xAOD::JetContainer* jets = 0;
    m_event->retrieve(jets, "AntiKt4EMTopoJets").ignore();

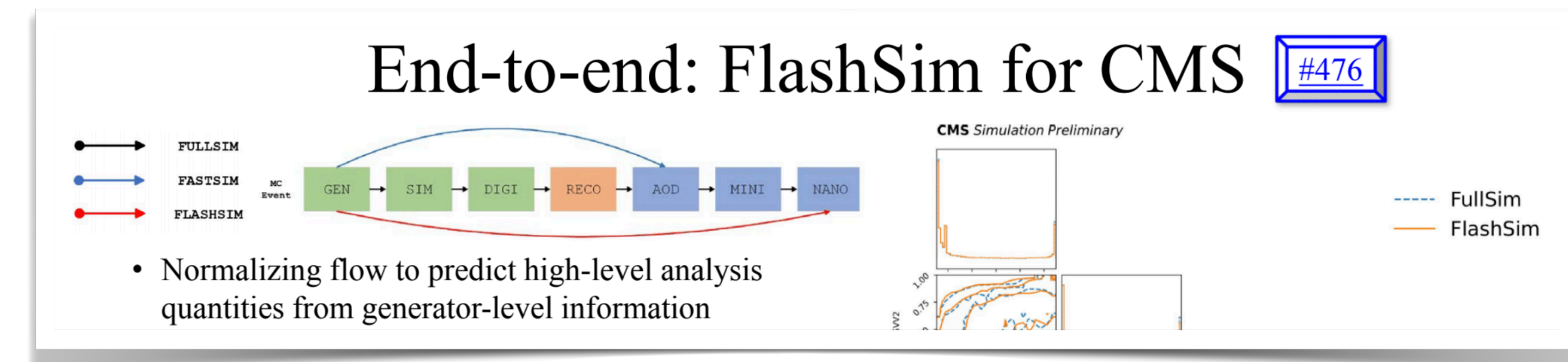
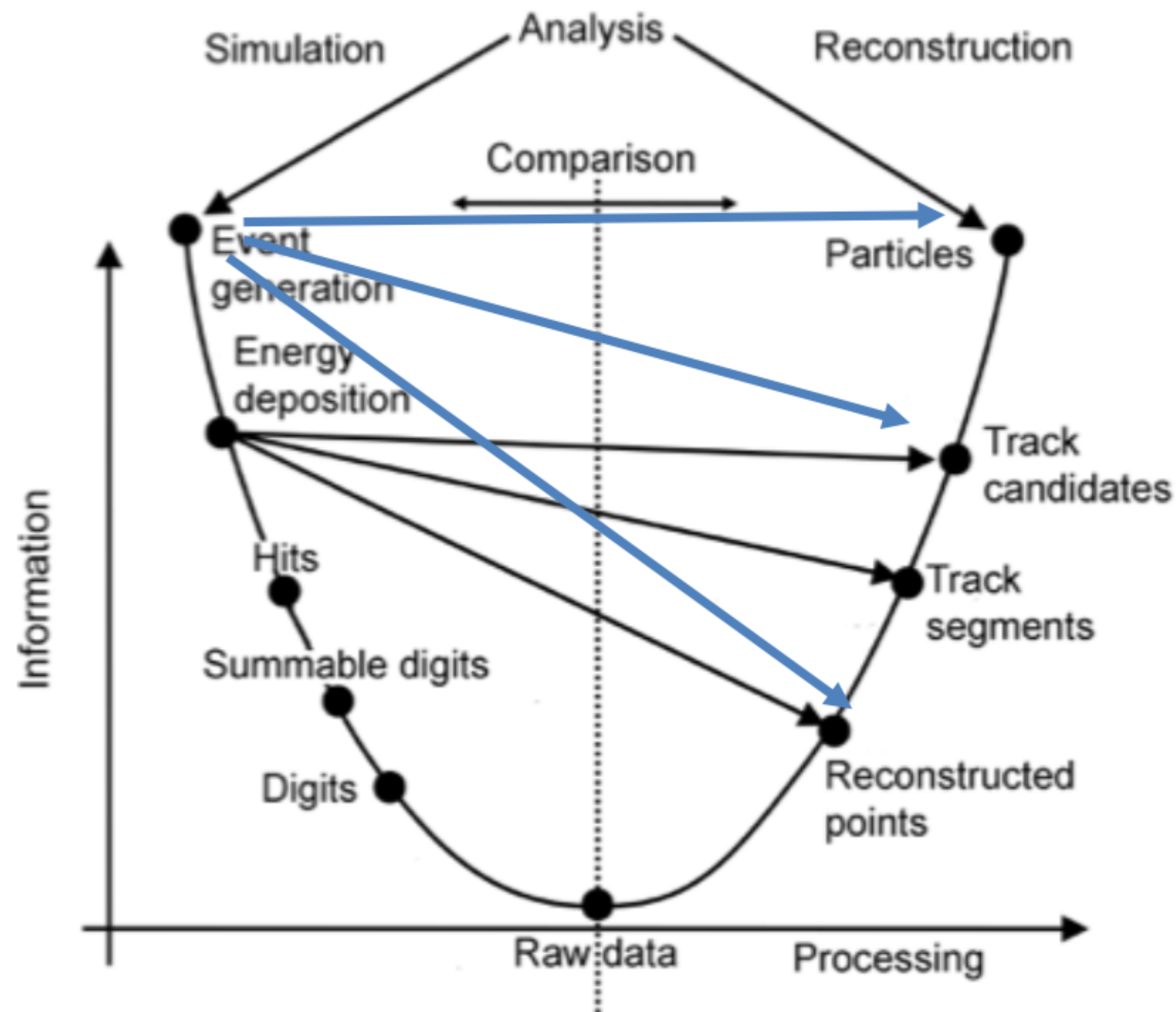
    // Loop over the jets and do something
    for (const xAOD::Jet* jet : *jets) {
        double pt = jet->pt();
        double eta = jet->eta();
        double phi = jet->phi();
```


Simulation, Reconstruction, and Analysis

The standard HEP workflow & shortcuts



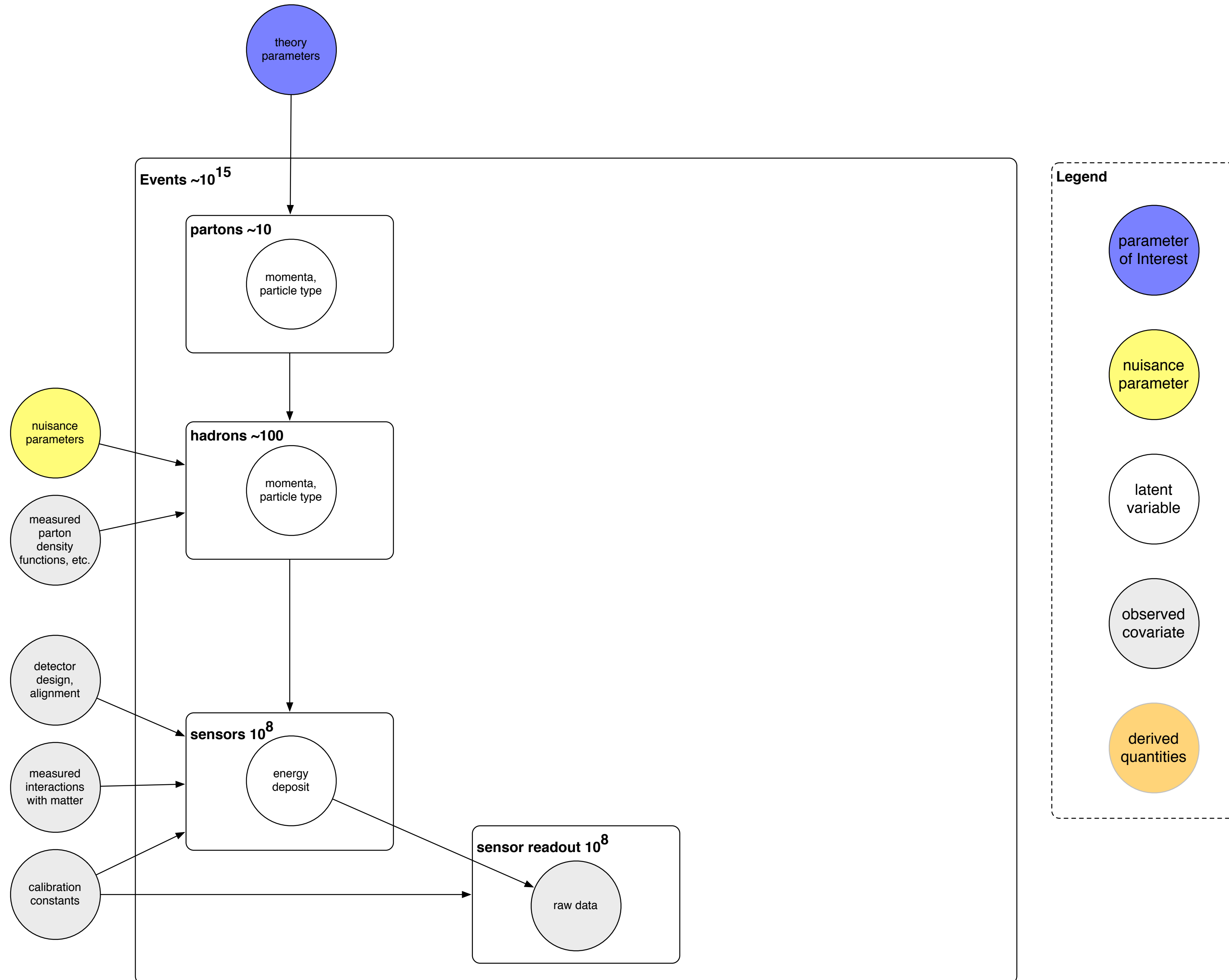
The standard HEP workflow & shortcuts



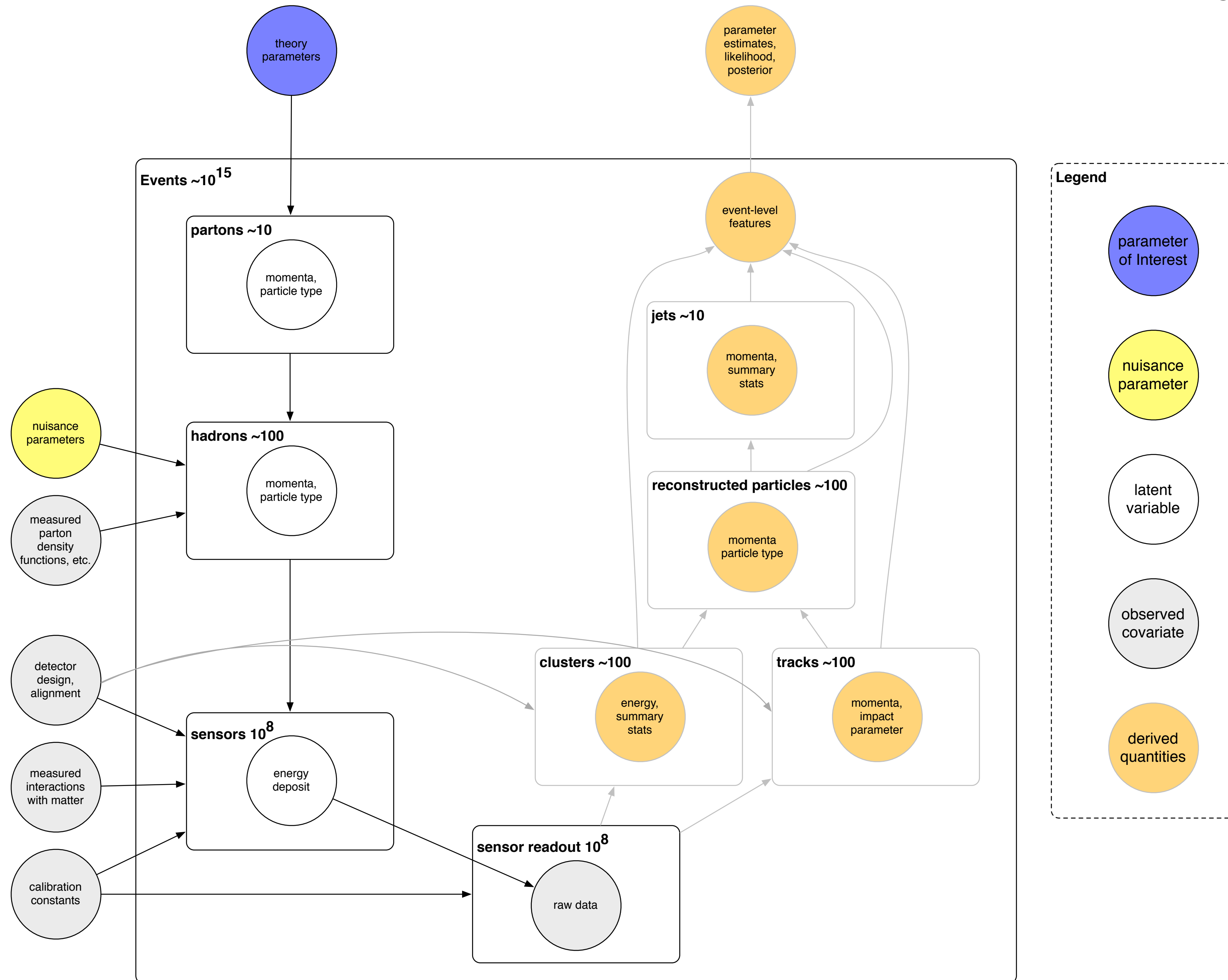
Generative AI will radically change fast simulation

See Kevin Pedro's talk next

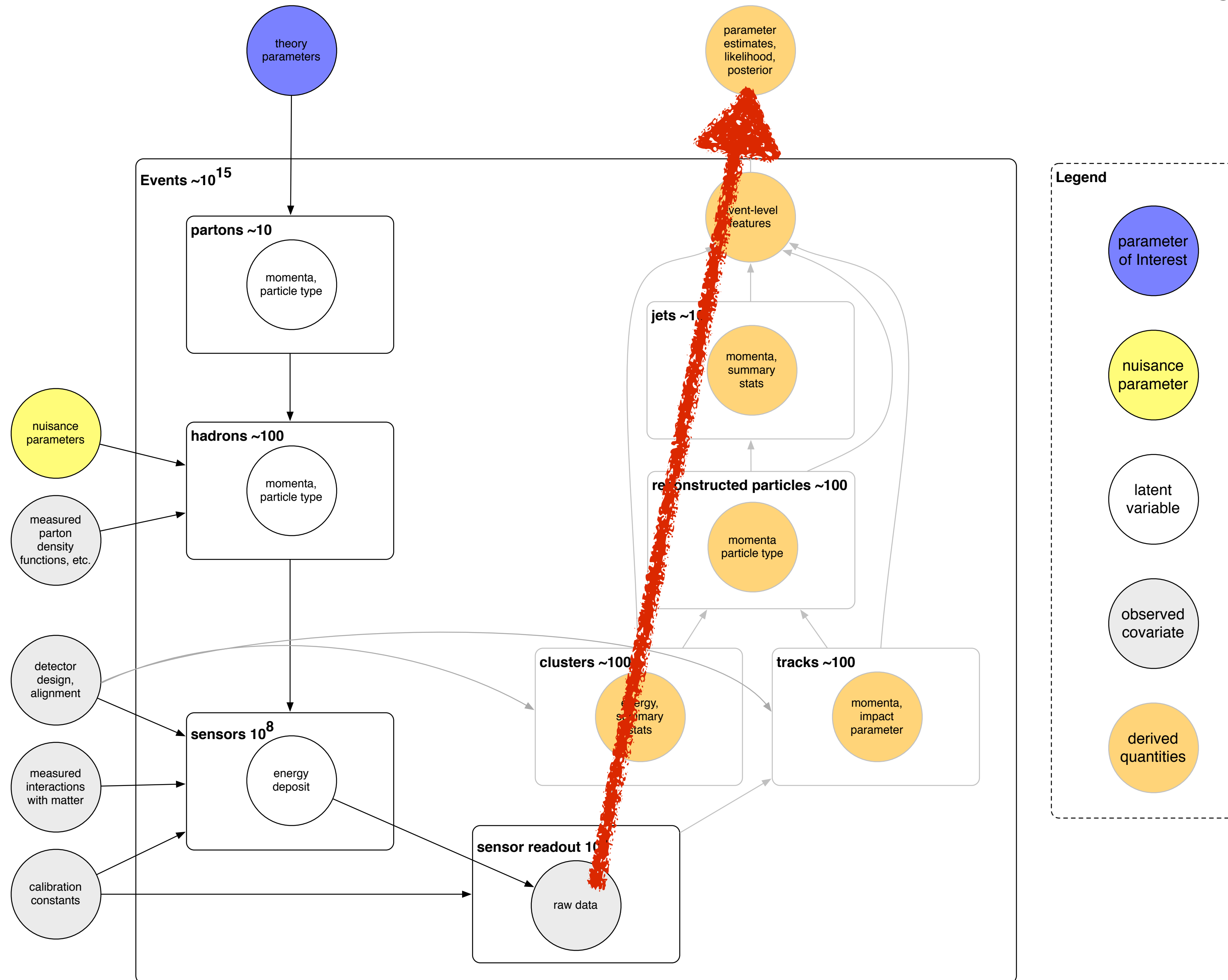
Full Simulation



Full Simulation + reconstruction + analysis



Full Simulation + reconstruction + analysis



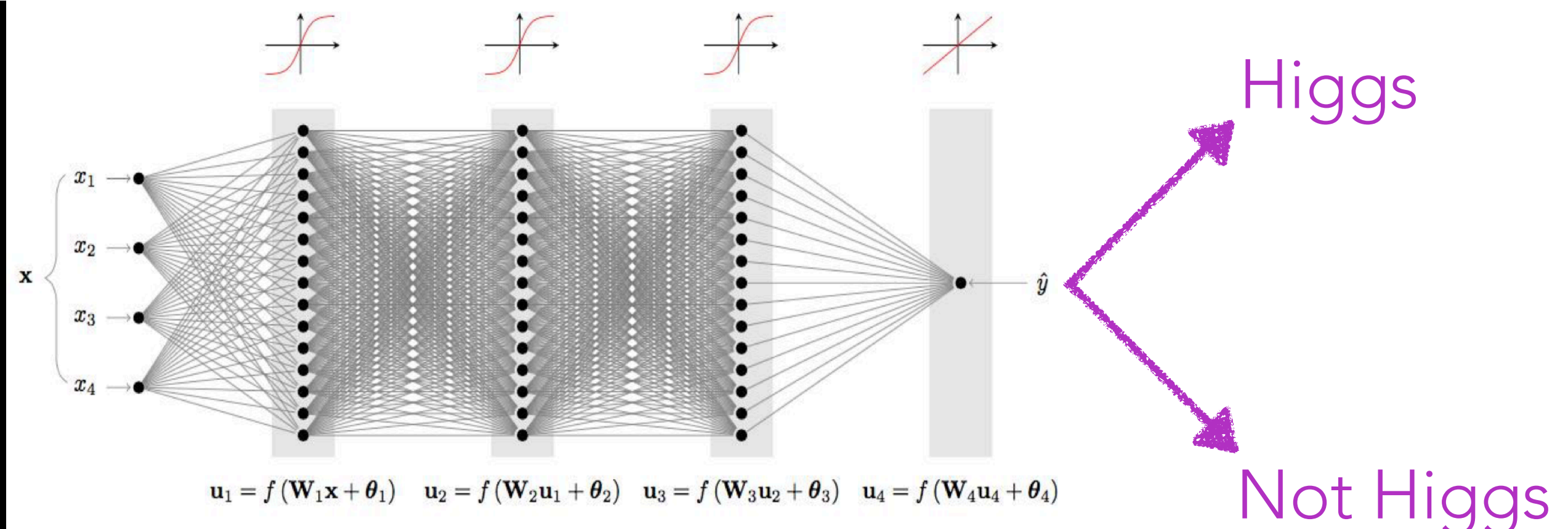
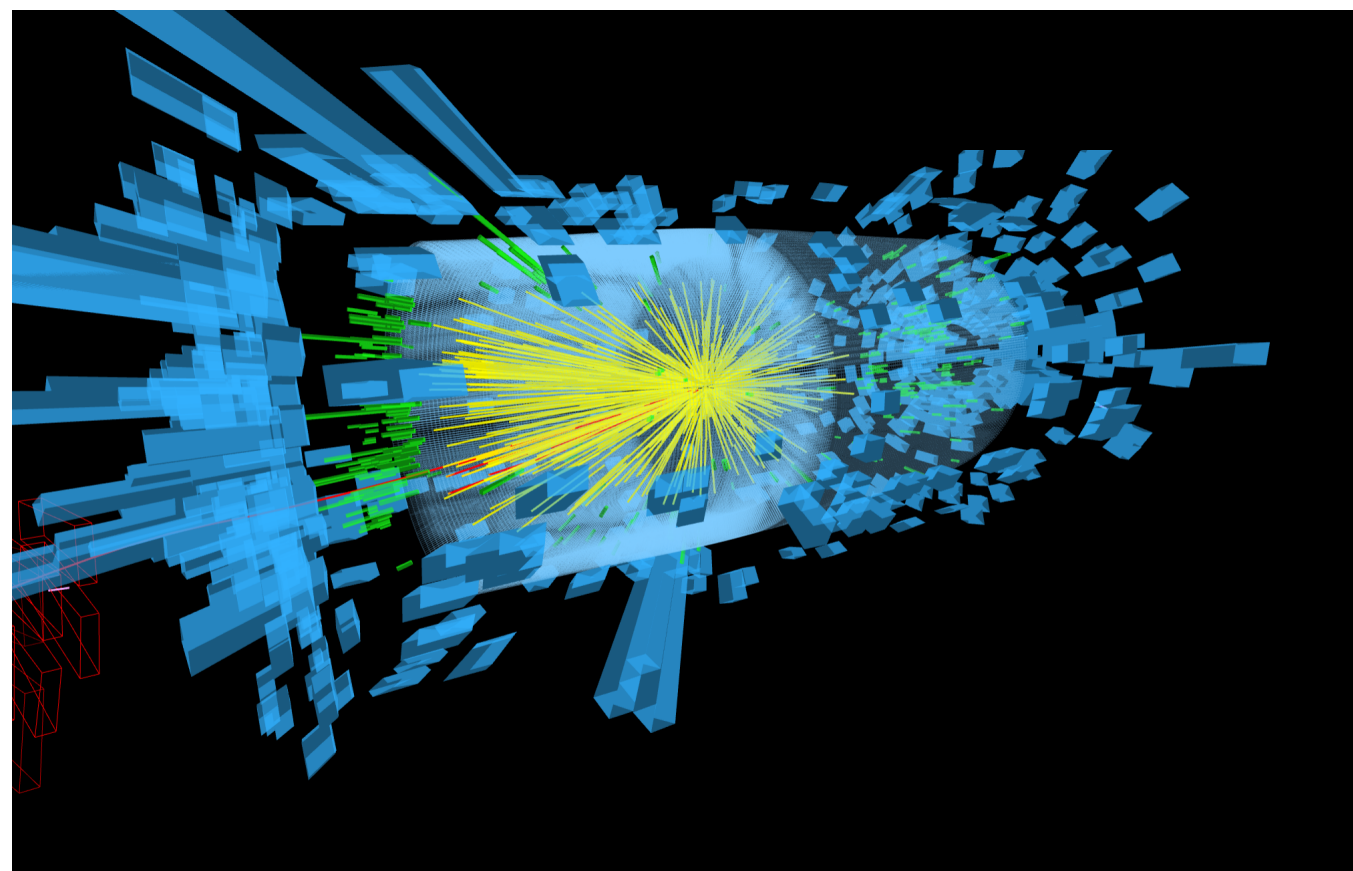
An example of radical change

End-to-End Physics Event Classification with CMS Open Data
Applying Image-Based Deep Learning to Detector Data for the Direct Classification
of Collision Events at the LHC

M. Andrews, M. Paulini, S. Gleyzer, B. Poczos

In end-to-end approach, bypass reconstruction, go straight to signal candidates

- Could also bypass event-level classification and go straight to physics result (measurement, claim of discovery) based on a sample of events
- Loose interpretation of intermediate objects. New approaches would be needed to evaluate object-level systematics and establish trust in result.
- Would require a very different organization of computing resources



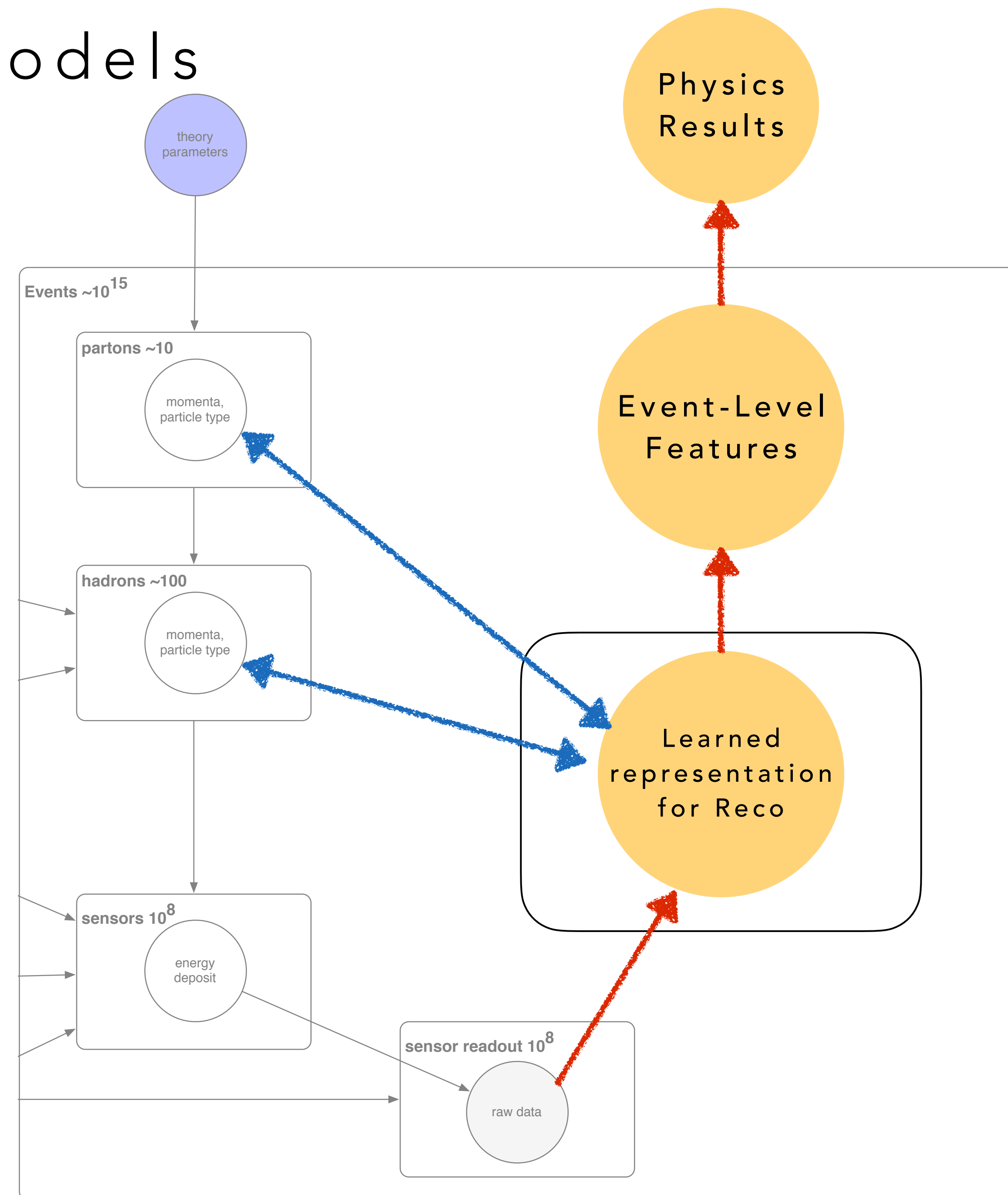
Reconstruction & Foundation Models

Instead of bypassing reconstruction, replace or augment the standard reconstructed objects with a rich, learned (black box?) representation

- e.g. the hidden state in the middle of a DNN
- Could use self-supervised learning to find it
- Basically a compression of the data

A common, general-purpose representation would serve as a **foundation model**

- Could have fast sim target this common black box representation of data
- Could share across experiments
- Could serve as a common data format or as a target for unfolding



Deterministic → Probabilistic Reconstruction

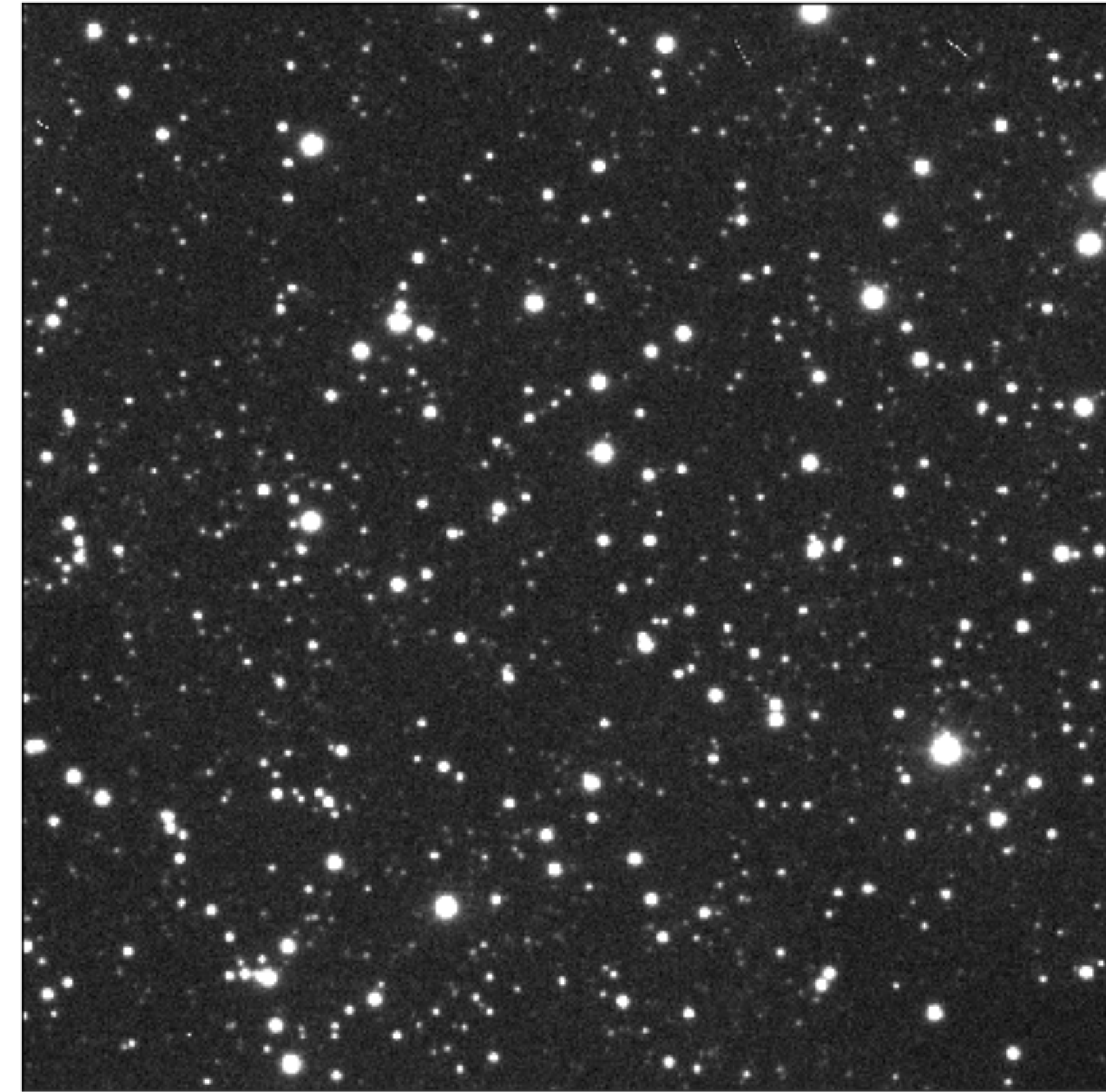
Our standard reconstruction produces a **deterministic estimate** of particle ID, kinematics, etc. Typically, we process one interpretation of an event.

Runjing Liu, et. Al. [arXiv:2102.02409](https://arxiv.org/abs/2102.02409)

- This is analogous to a catalog in astronomy

Astronomy is increasingly exploring the idea of **probabilistic catalogues**

- Address ambiguities in reconstruction pipeline (e.g. merged objects, photo-z's, ...)
- Analysis of individual events at LIGO is probabilistic

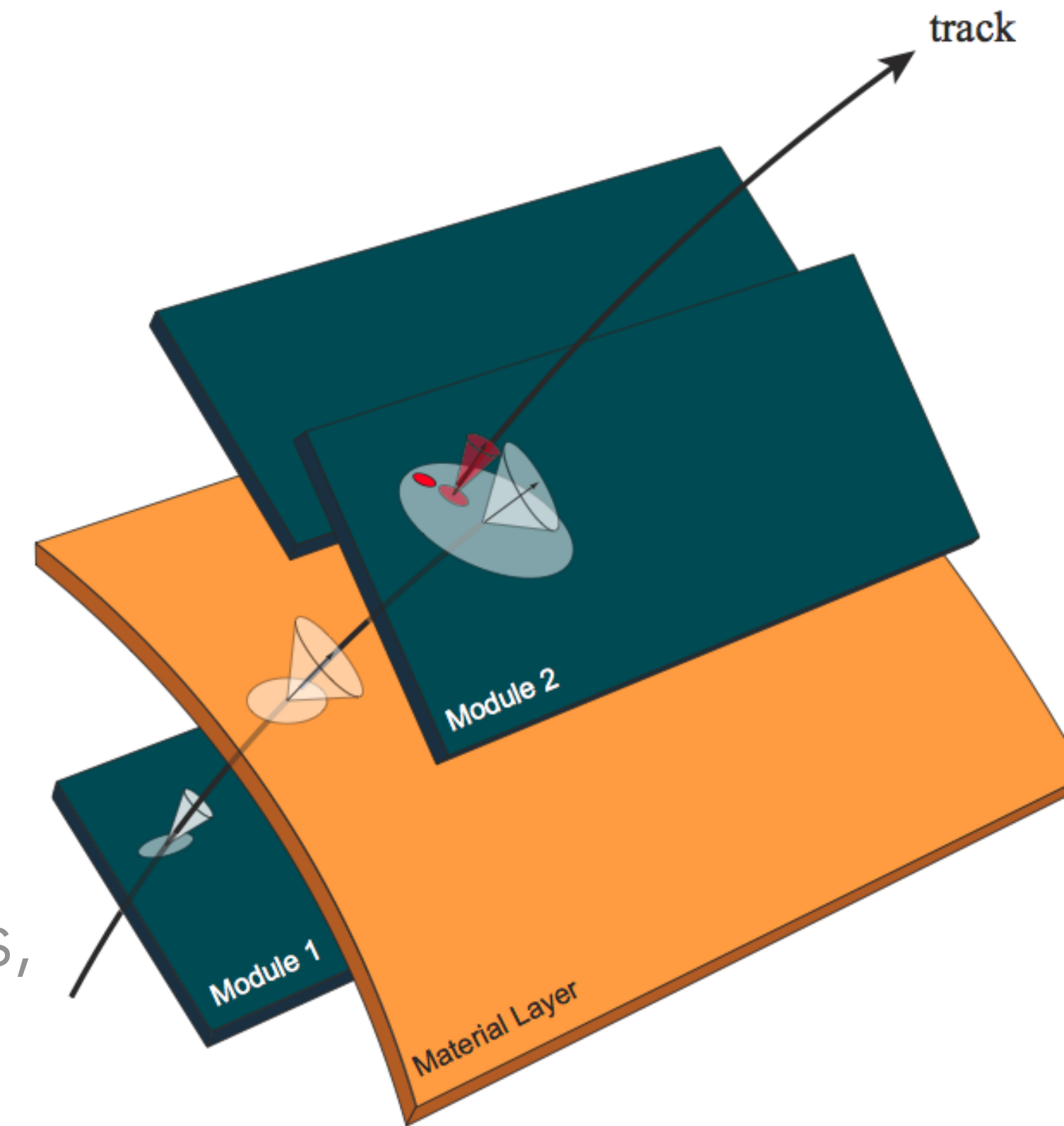


Conceptually appealing and could improve some analyses, but consequences

Probabilistic Reconstruction

Conceptually appealing... but how?

- Attach N-Dim probability distributions to each object
 - Similar to what we do for covariance matrix on track parameters, but want something more flexible
 - How will they be used downstream?
 - Once you move away from simple operations or distributions, it gets hard, e.g. Kalman-Filter -> Gaussian Sum Filter etc.
- Sample different interpretations & process them the standard way
 - Increases computational, storage, & book-keeping requirements
 - Similar to systematic variations on simulation, but for data too



Both approaches complicate downstream statistical procedures

Publishing a model for the data distribution

If Generative AI techniques were good enough, we could use the model as a dramatically compressed representation of the entire dataset

- Conceptually equivalent to publishing mean and std. dev. of a Gaussian
- Could generate synthetic data that follows data distribution without reference to theory or simulation
- Would enable comparisons between data and predictions (but need simulation)
- Would be very hard to validate for complex, high-dimensional data
- Could use same idea in the context of the trigger to represent data we don't keep

Related, high-dimensional unfolding enabled by AI/ML

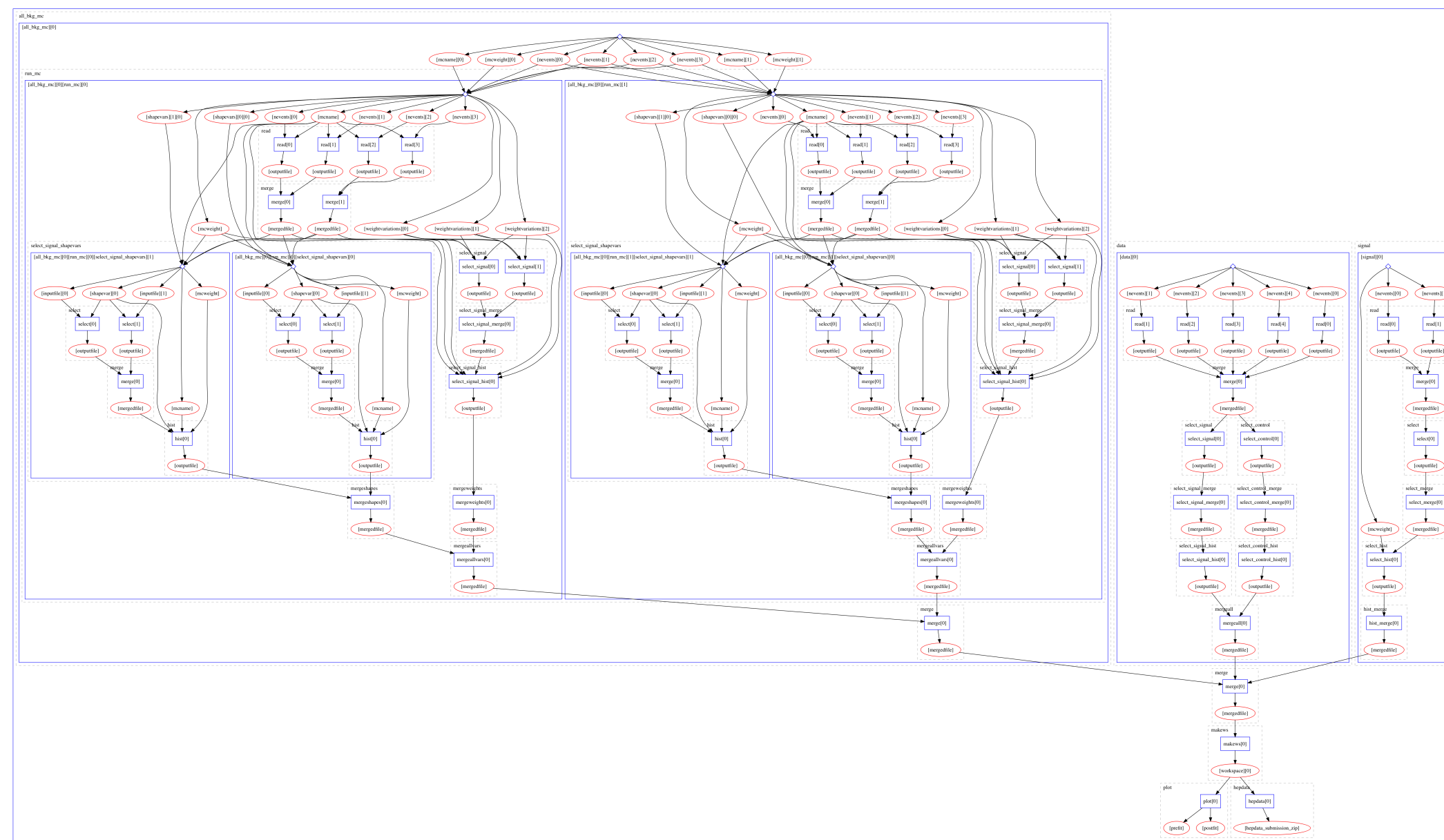
- e.g. **OmniFold** [Andreassen, Komiske, Metodiev, Nachman, Thaler [arXiv:1911.09107](https://arxiv.org/abs/1911.09107)]

Reinterpretation & Connection to Theory

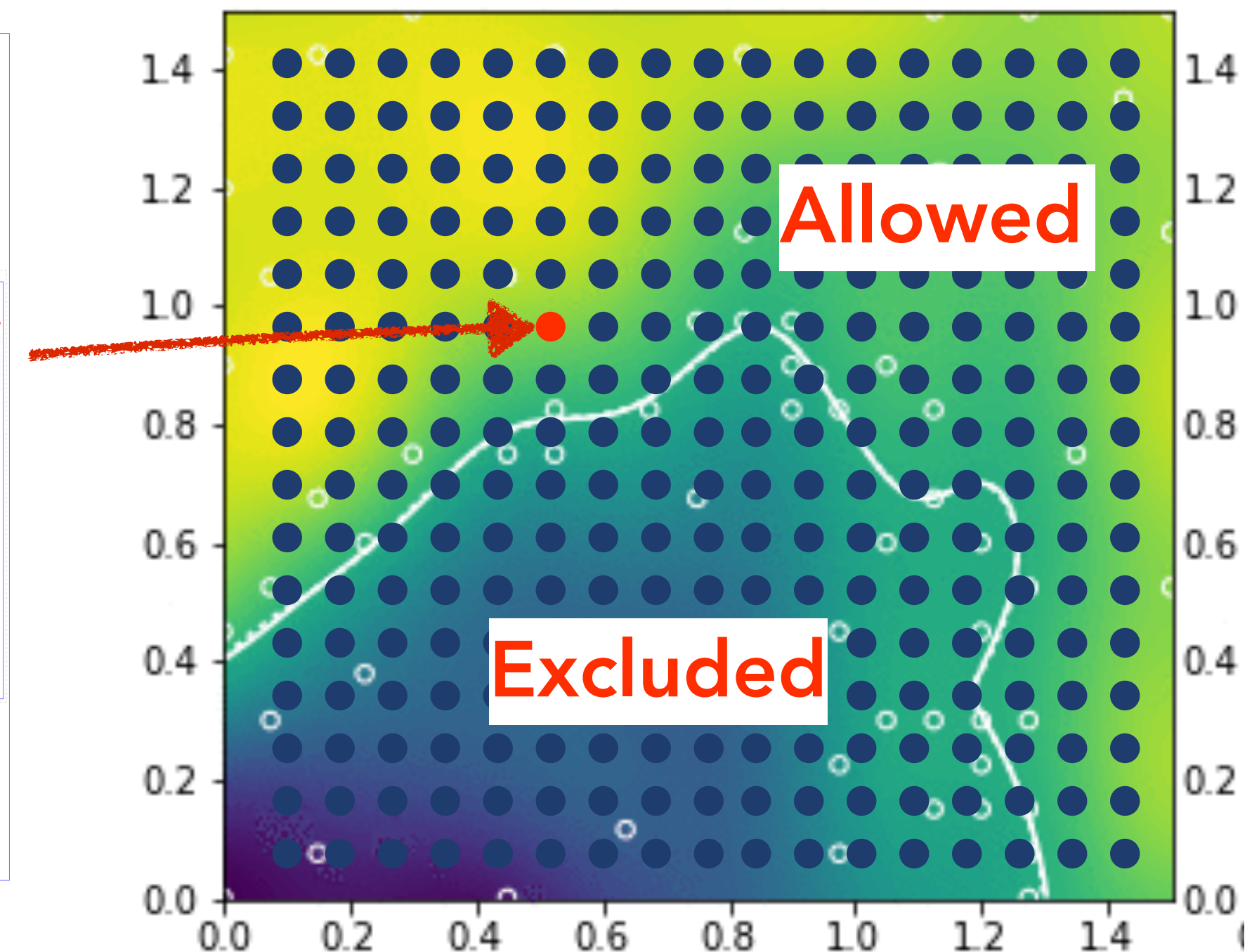
Reinterpretation / RECAST

It is common in particle physics to:

- scan parameter space of theory, simulate signal for each point
- execute complex workflow that implements analysis for each parameter point
- determine which regions of parameter space are excluded



Complex computational workflow



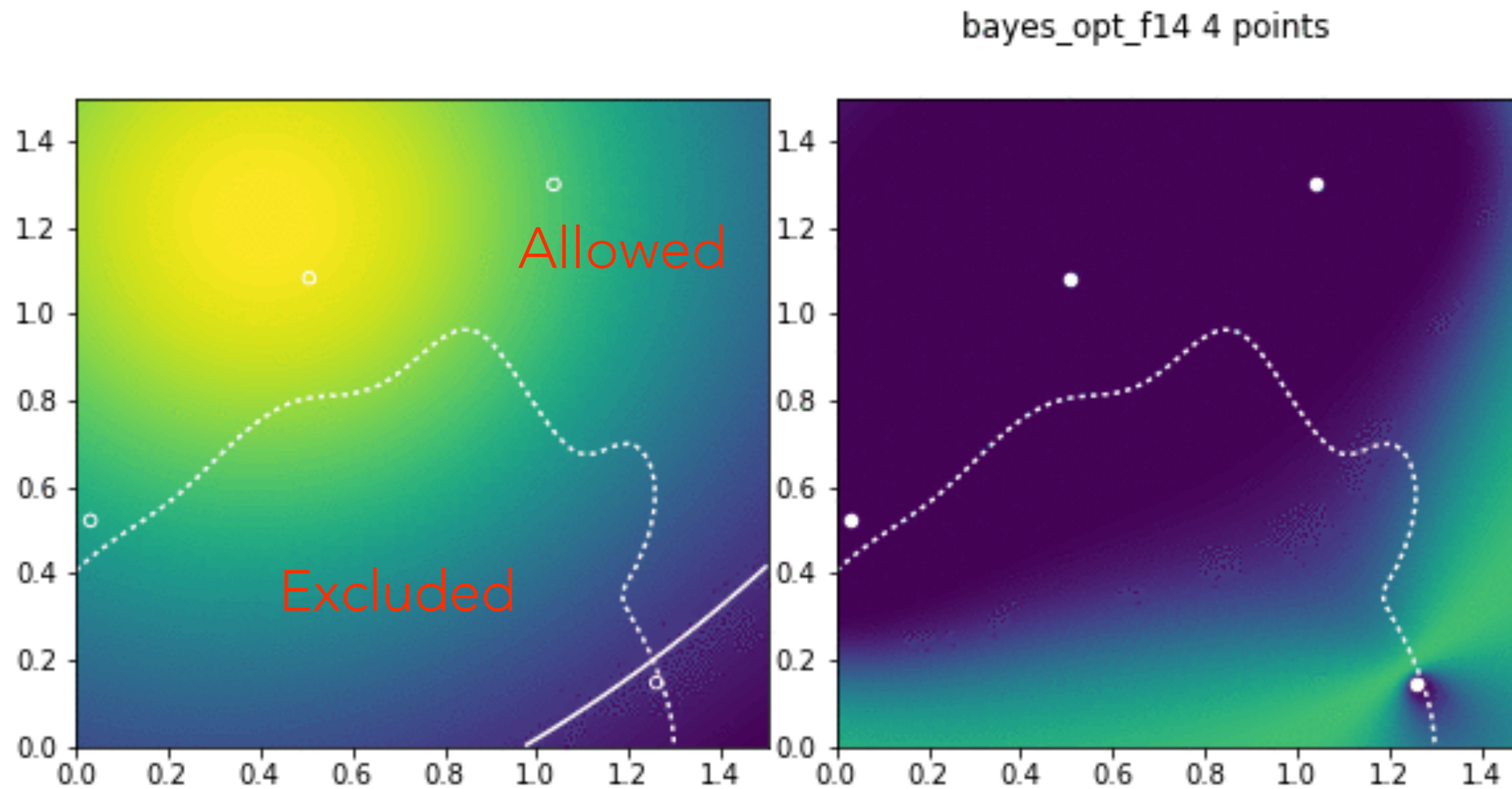
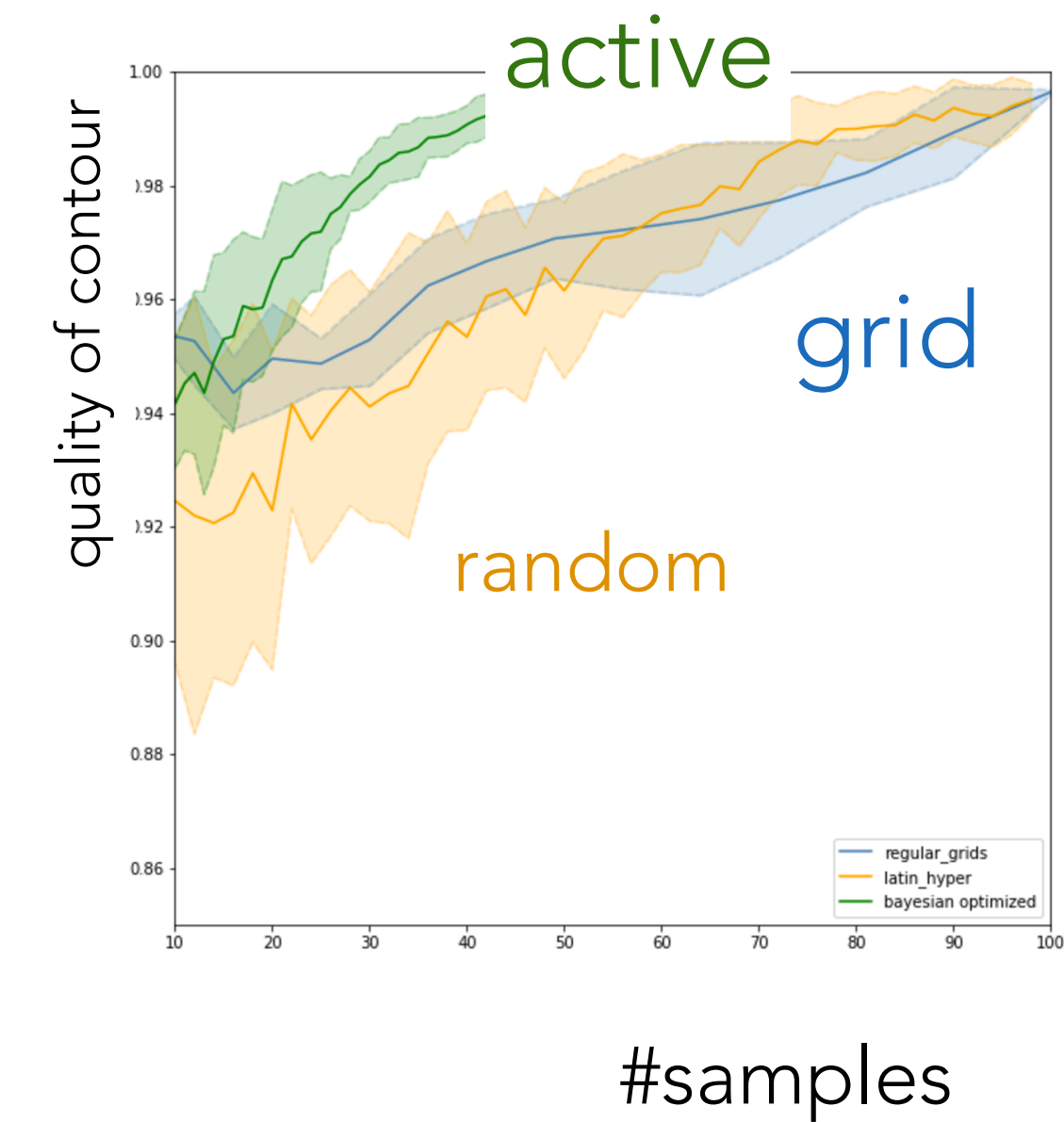
Active Learning for reinterpretation

↓ Instead of generating MC a priori, generate it on demand where it is relevant!

Drastically more efficient use of computing resources →

Changes traditional relationship between production system & analysis

Analysis drives production system



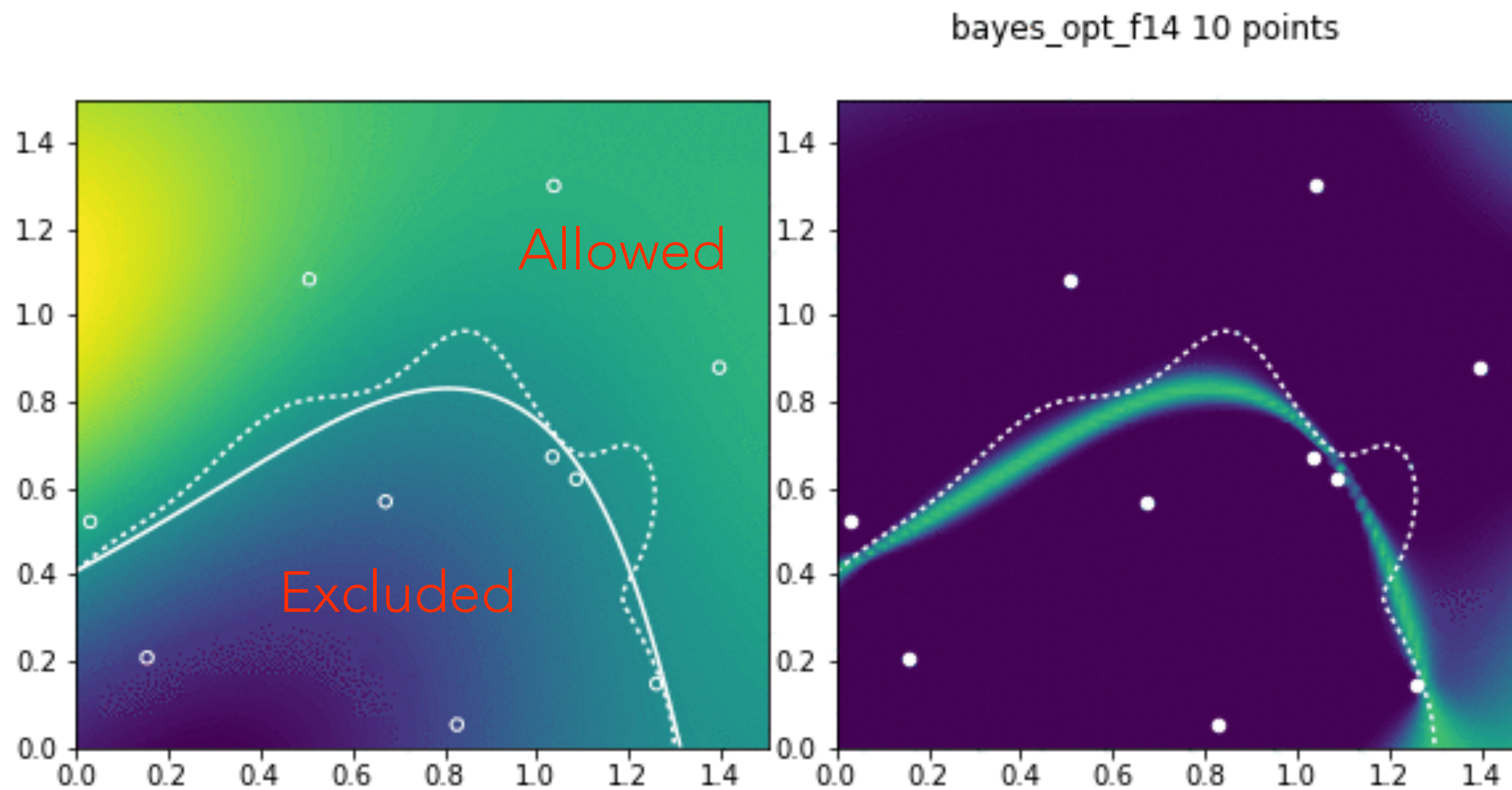
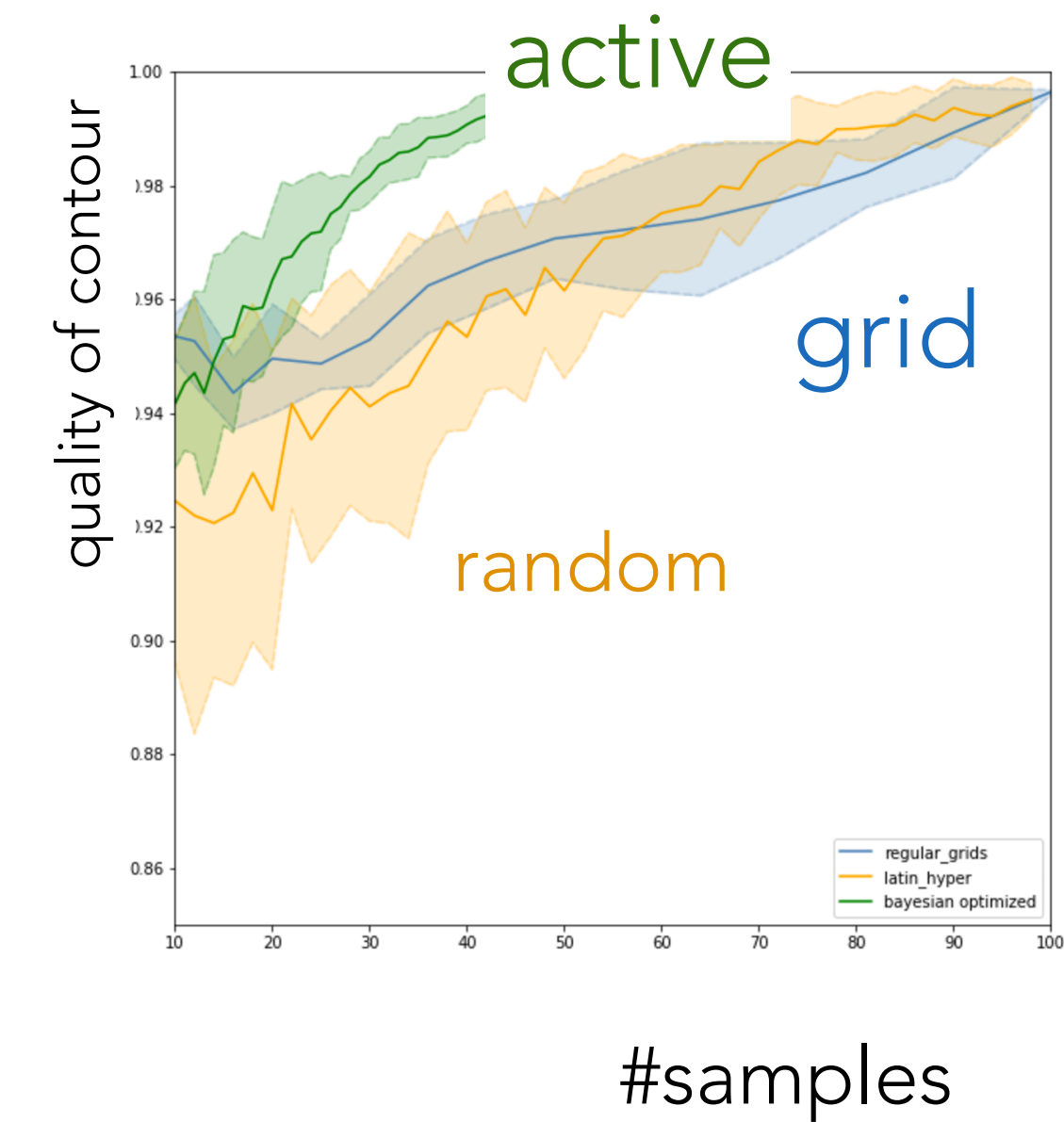
Active Learning for reinterpretation

↓ Instead of generating MC a priori, generate it on demand where it is relevant!

Drastically more efficient use of computing resources →

Changes traditional relationship between production system & analysis

Analysis drives production system



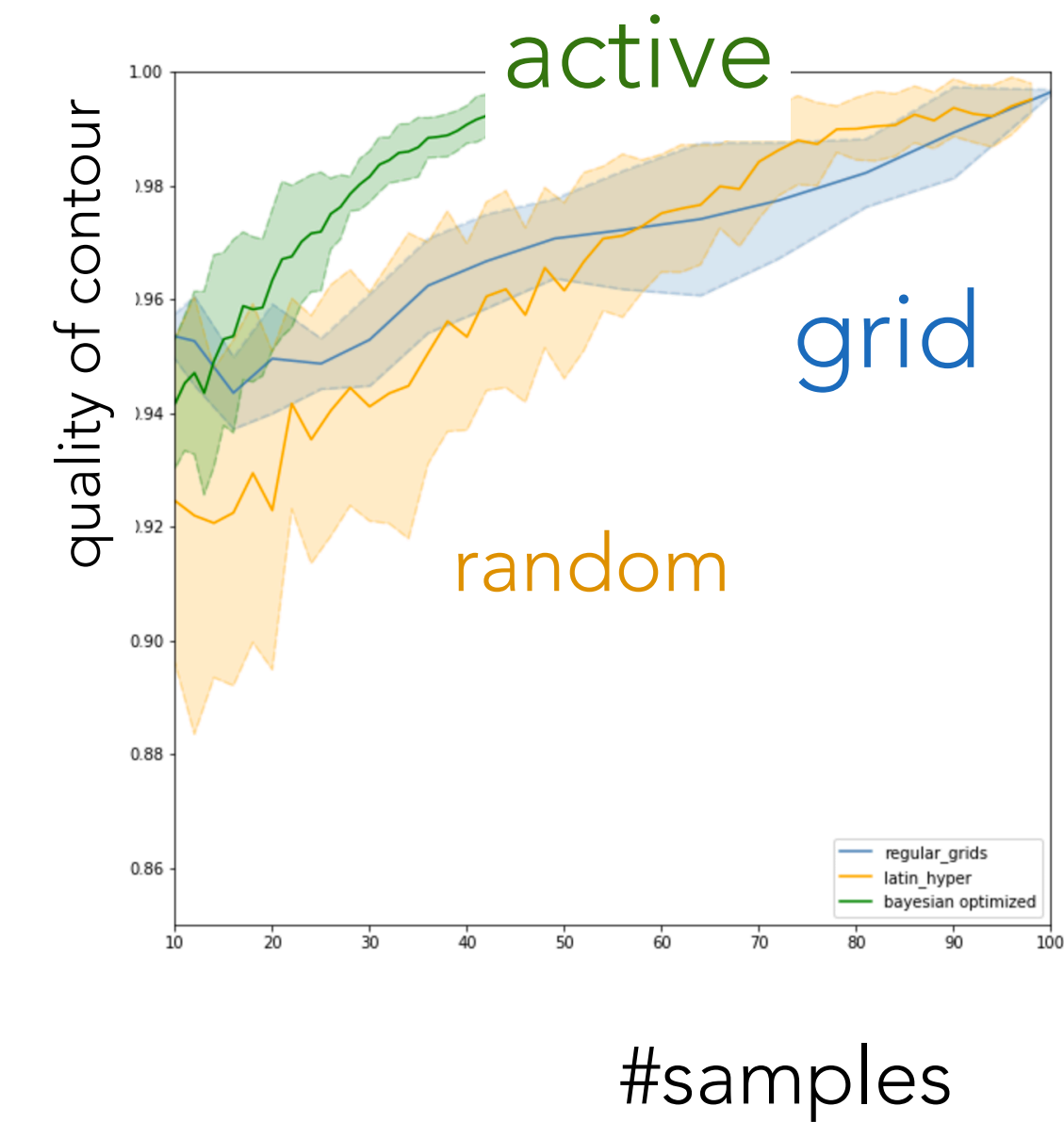
Active Learning for reinterpretation

↓ Instead of generating MC a priori, generate it on demand where it is relevant!

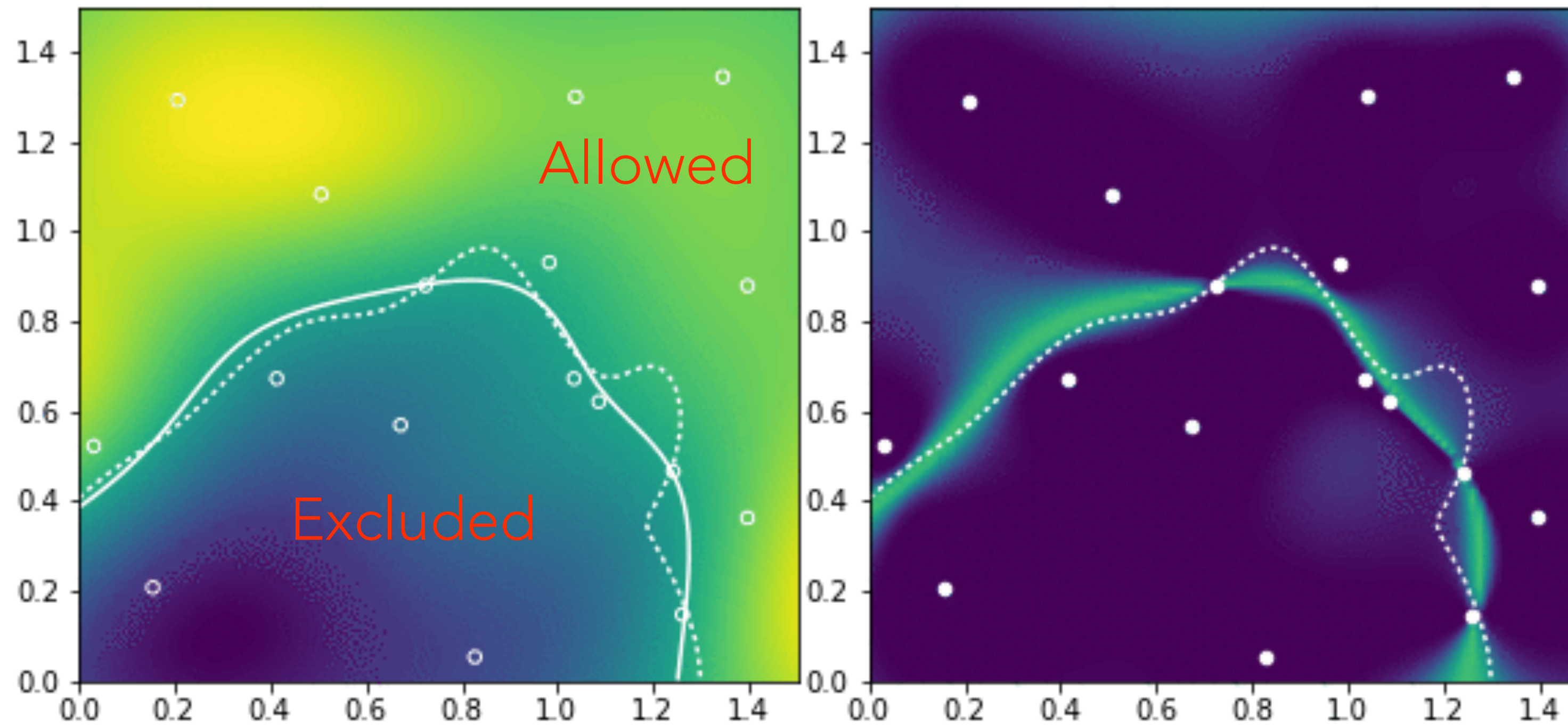
Drastically more efficient use of computing resources →

Changes traditional relationship between production system & analysis

Analysis drives production system



bayes_opt_f14 17 points



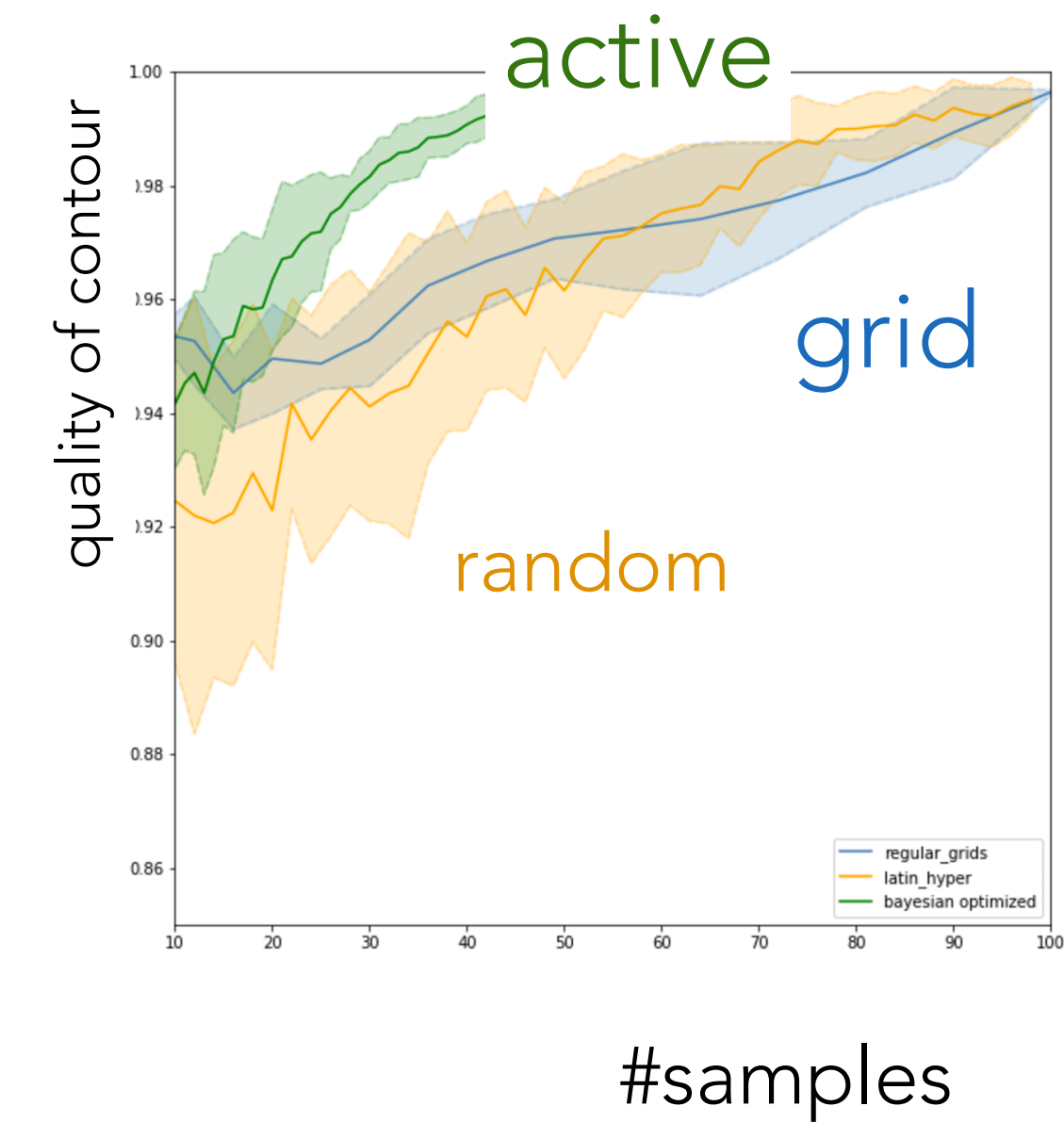
Active Learning for reinterpretation

↓ Instead of generating MC a priori, generate it on demand where it is relevant!

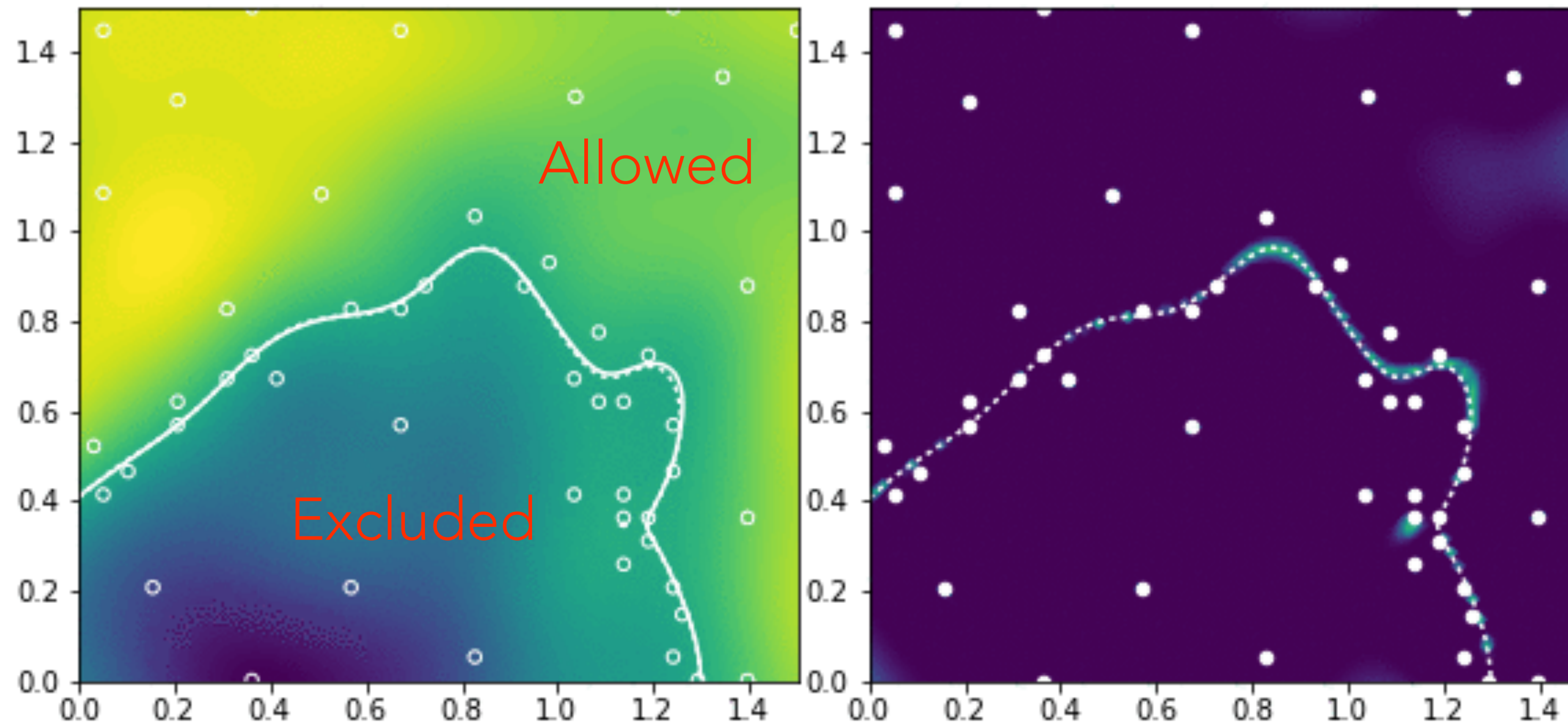
Drastically more efficient use of computing resources →

Changes traditional relationship between production system & analysis

Analysis drives production system



bayes_opt_f14 50 points



Anomaly detection

Lots of interest recently in anomaly detection — fueled by machine learning

- Driven by a desire to be model-independent

The LHC Olympics 2020

A Community Challenge for Anomaly
Detection in High Energy Physics



Gregor Kasieczka (ed),¹ Benjamin Nachman (ed),^{2,3} David Shih (ed),⁴ Oz Amram,⁵
Anders Andreassen,⁶ Kees Benkendorfer,^{2,7} Blaz Bortolato,⁸ Gustaaf Brooijmans,⁹
Florenca Canelli,¹⁰ Jack H. Collins,¹¹ Biwei Dai,¹² Felipe F. De Freitas,¹³ Barry M.
Dillon,^{8,14} Ioan-Mihail Dinu,⁵ Zhongtian Dong,¹⁵ Julien Donini,¹⁶ Javier Duarte,¹⁷ D.
A. Faroughy,¹⁰ Julia Gonski,⁹ Philip Harris,¹⁸ Alan Kahn,⁹ Jernej F. Kamenik,^{8,19}
Charanjit K. Khosa,^{20,30} Patrick Komiske,²¹ Luc Le Pottier,^{2,22} Pablo
Martín-Ramiro,^{2,23} Andrej Matevc,^{8,19} Eric Metodiev,²¹ Vinicius Mikuni,¹⁰ Inês
Ochoa,²⁴ Sang Eon Park,¹⁸ Maurizio Pierini,²⁵ Dylan Rankin,¹⁸ Veronica Sanz,^{20,26}
Nilai Sarda,²⁷ Uroš Seljak,^{2,3,12} Aleks Smolkovic,⁸ George Stein,^{2,12} Cristina Mantilla
Suarez,⁵ Manuel Szwec,²⁸ Jesse Thaler,²¹ Steven Tsan,¹⁷ Silviu-Marian Udrescu,¹⁸
Louis Vaslin,¹⁶ Jean-Roch Vlimant,²⁹ Daniel Williams,⁹ Mikaeel Yunus¹⁸

3	Unsupervised	11
3.1	Anomalous Jet Identification via Variational Recurrent Neural Network	11
3.2	Anomaly Detection with Density Estimation	16
3.3	BuHuLaSpa: Bump Hunting in Latent Space	19
3.4	GAN-AE and BumpHunter	24
3.5	Gaussianizing Iterative Slicing (GIS): Unsupervised In-distribution Anomaly Detection through Conditional Density Estimation	29
3.6	Latent Dirichlet Allocation	33
3.7	Particle Graph Autoencoders	38
3.8	Regularized Likelihoods	42
3.9	UCluster: Unsupervised Clustering	46
4	Weakly Supervised	51
4.1	CWoLa Hunting	51
4.2	CWoLa and Autoencoders: Comparing Weak- and Unsupervised methods for Resonant Anomaly Detection	55
4.3	Tag N' Train	60
4.4	Simulation Assisted Likelihood-free Anomaly Detection	63
4.5	Simulation-Assisted Decorrelation for Resonant Anomaly Detection	68
5	(Semi)-Supervised	71
5.1	Deep Ensemble Anomaly Detection	71
5.2	Factorized Topic Modeling	77
5.3	QUAK: Quasi-Anomalous Knowledge for Anomaly Detection	81
5.4	Simple Supervised learning with LSTM layers	85

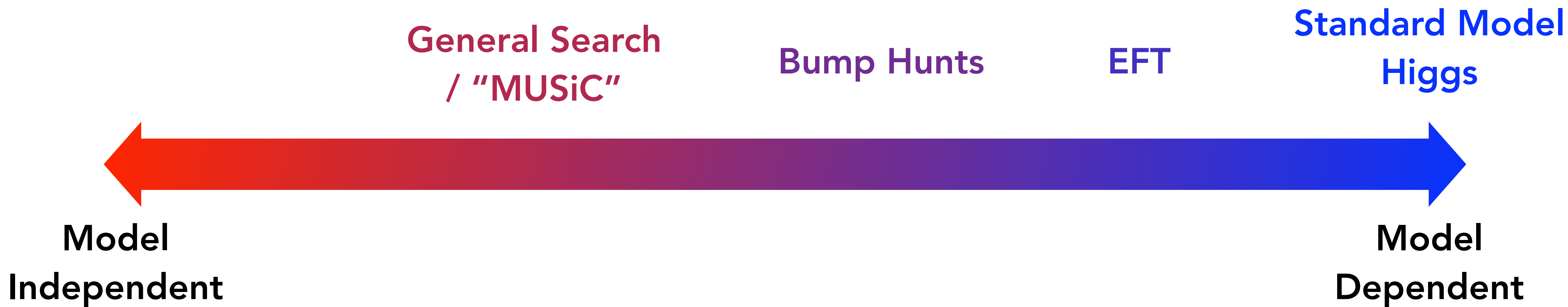
A spectrum

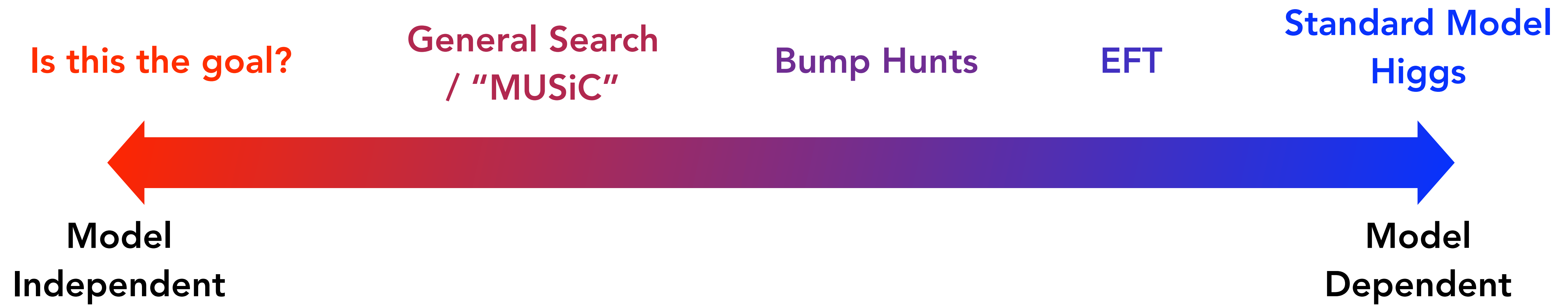














A spectrum



Is a fully model-independent approach our goal?

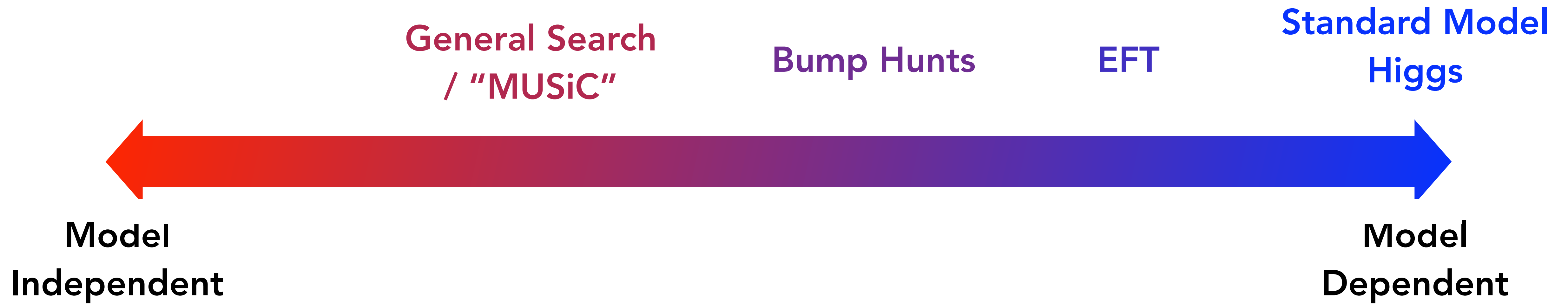
- What does that even mean?
- Is it the right goal? Is it a reasonable goal?
- How do we connect back to theory?

Summary and Outlook

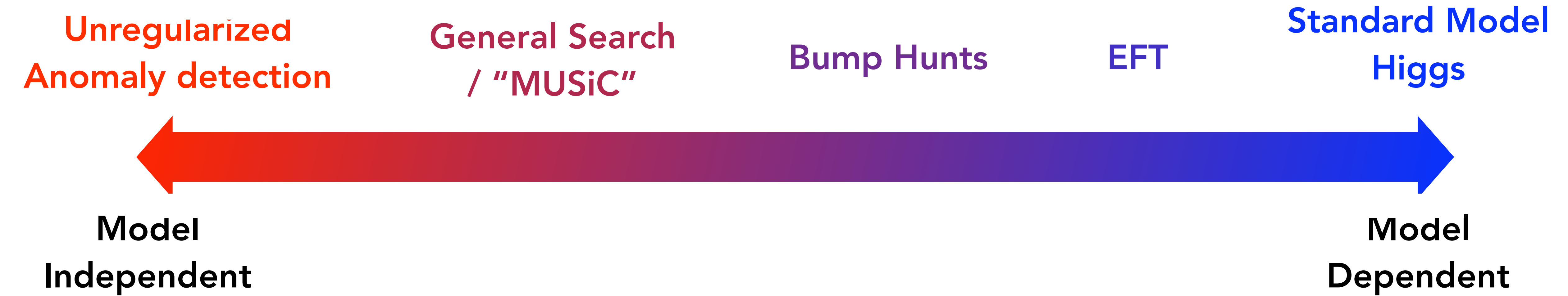
- Computing technology, built on silicon, has enabled a revolution in almost every facet of life and science; this 70-year trend will continue although the underlying technology is reaching maturity (linear growth)
- Physics requires experiments, data, and theory...working hand in hand to explain phenomena...
 - there is a valid, proven, way to practice science: ask a falsifiable question, construct a hypothesis, perform an experiment, refine hypothesis...repeat...
 - There is also an incorrect way to practice science which leads to alchemy: collect data, find patterns magically, forget theory, repeat
- Poincare': Science is built of facts the way a house is built of bricks: but an accumulation of facts is no more science than a pile of bricks is a house.

David Dean's keynote this morning

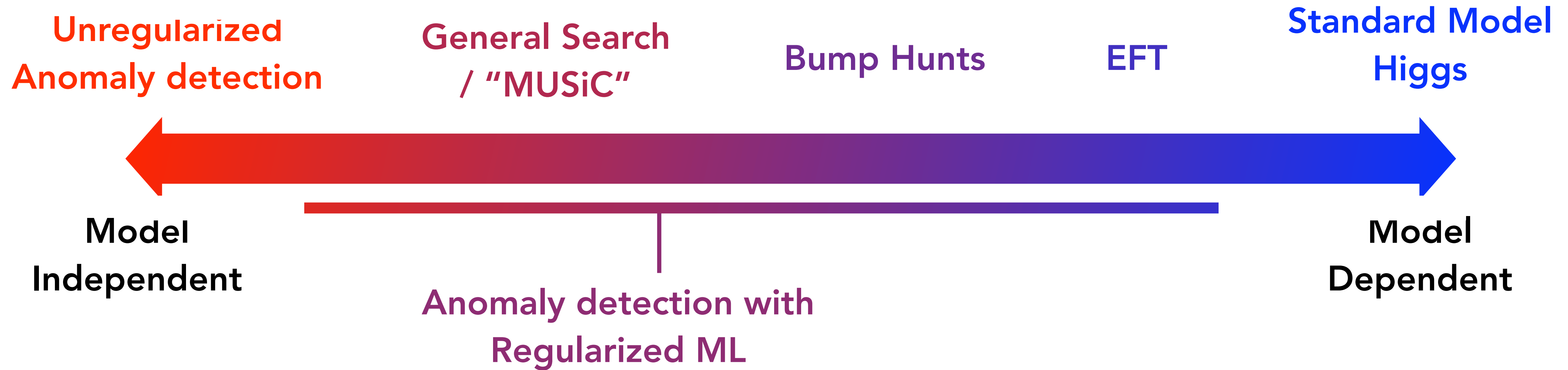
The spectrum revisited



The spectrum revisited

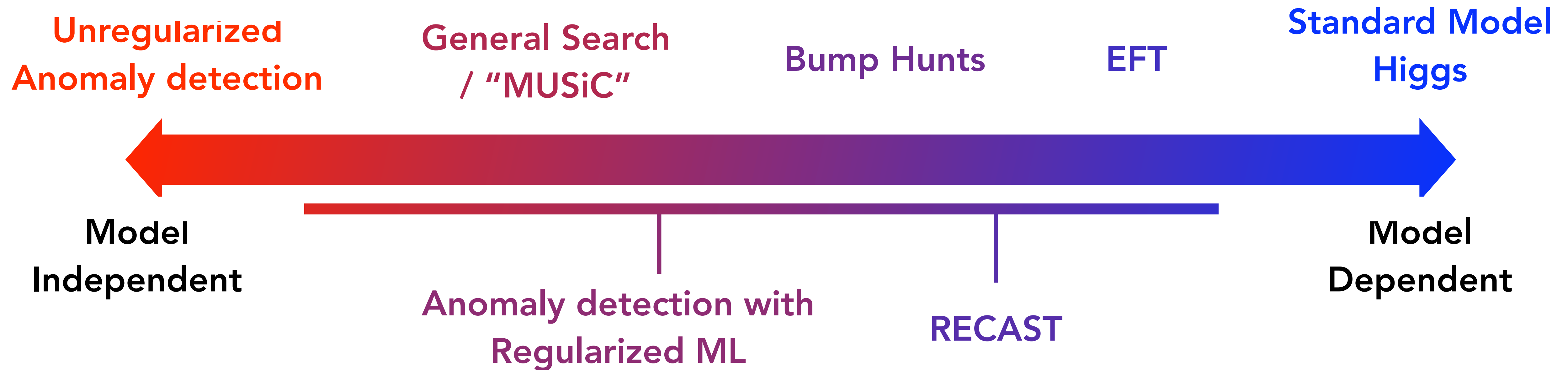


The spectrum revisited



Regularized ML allow us to specify "alternate hypothesis" in a language other than QFT that captures intuitive physics. Much more should be done here!

The spectrum revisited

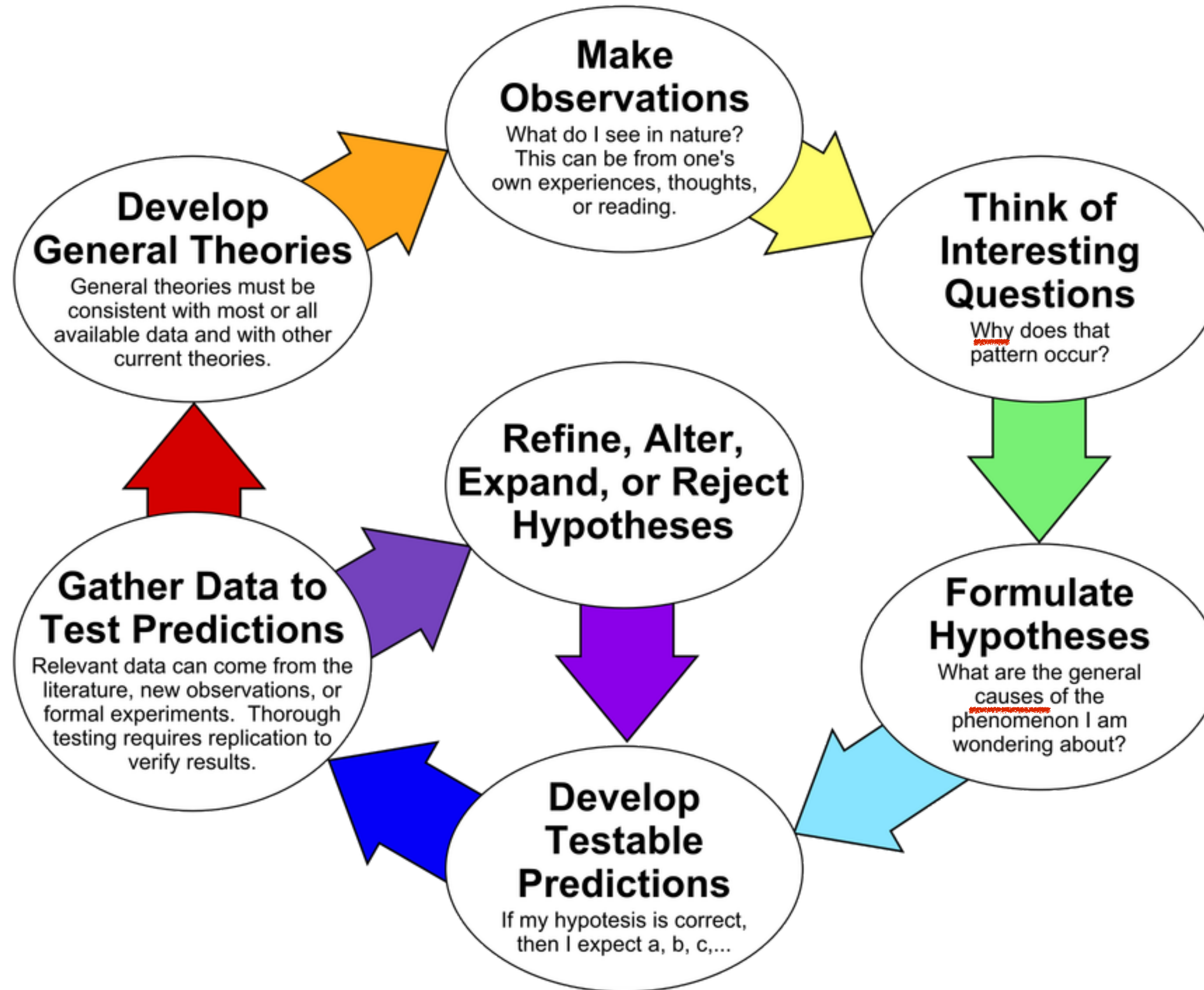


Regularized ML allow us to specify "alternate hypothesis" in a language other than QFT that captures intuitive physics. Much more should be done here!

RECAST allows us to interpret results and connect back to theory

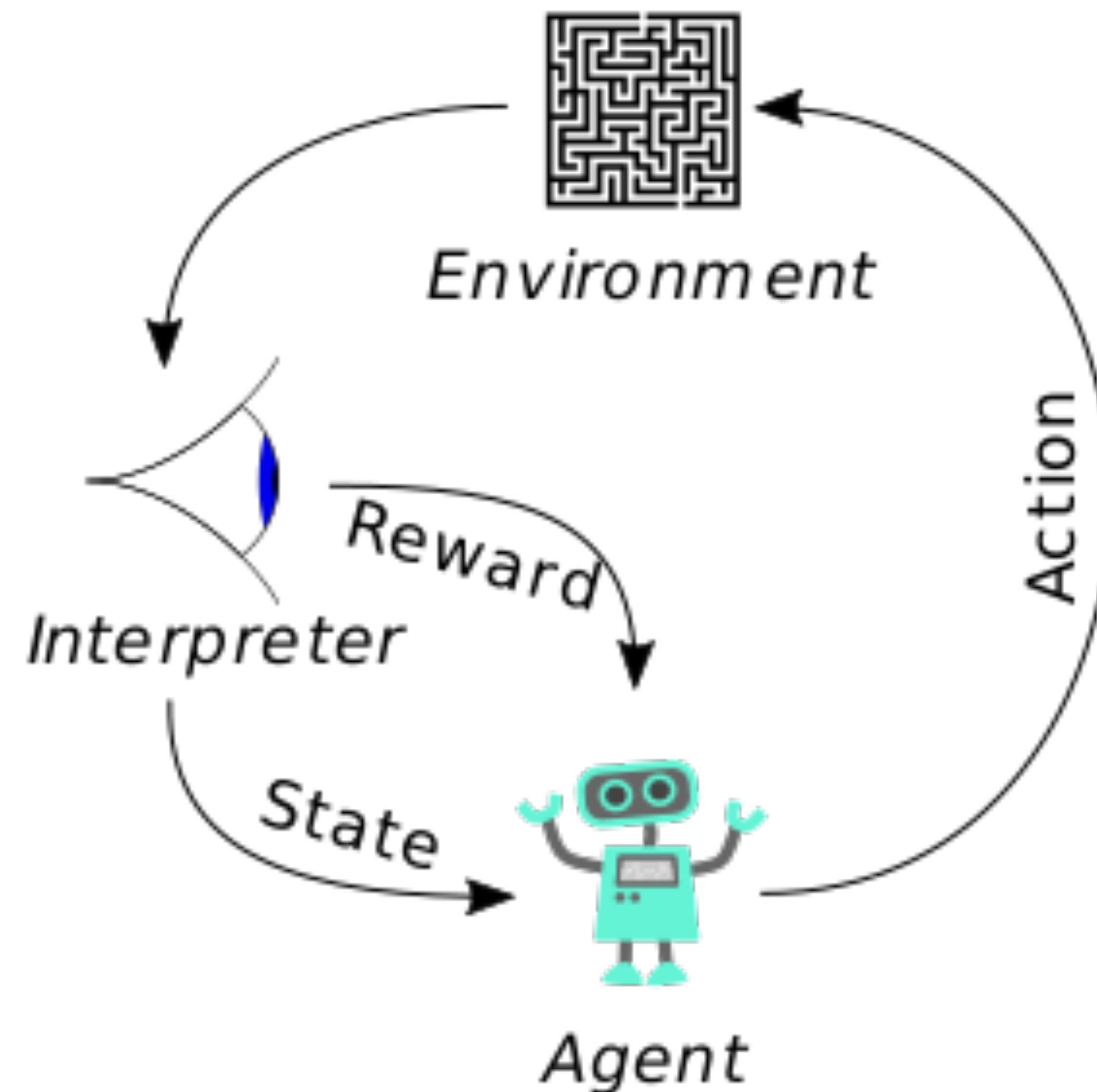
(Reinterpretation of anomaly detection analyses is subtle and computationally expensive)

The Scientific Method as an Ongoing Process



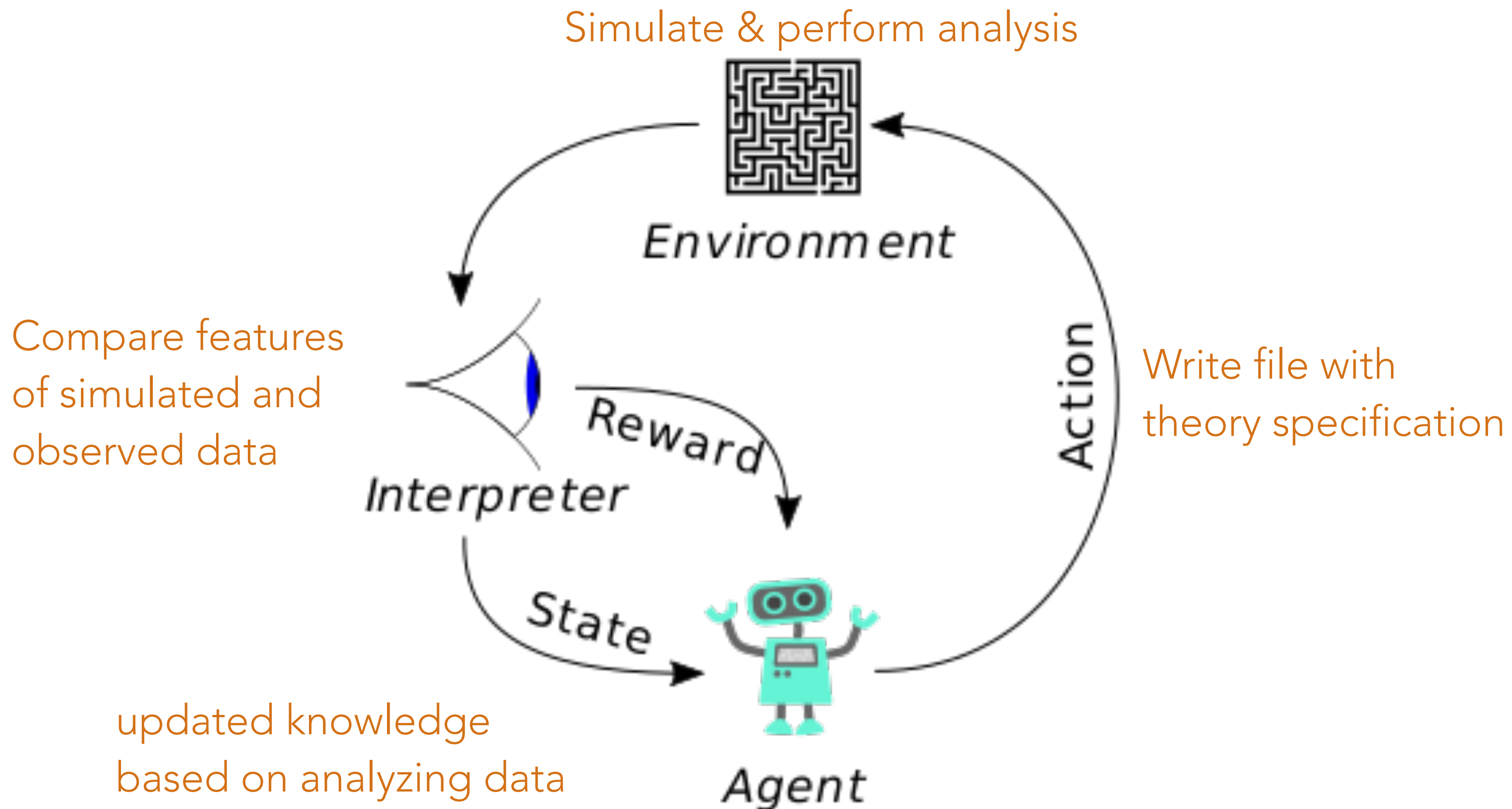
Closing the loop: Hypothesis generation

Conceivable that we could use LLMs + multimodal generative AI to write UFO / QRules files and explore space of theories based on features of an anomaly



Closing the loop: Hypothesis generation

Conceivable that we could use LLMs + multimodal generative AI to write UFO / QRules files and explore space of theories based on features of an anomaly



Trigger & DAQ

AI/ML for trigger & DAQ

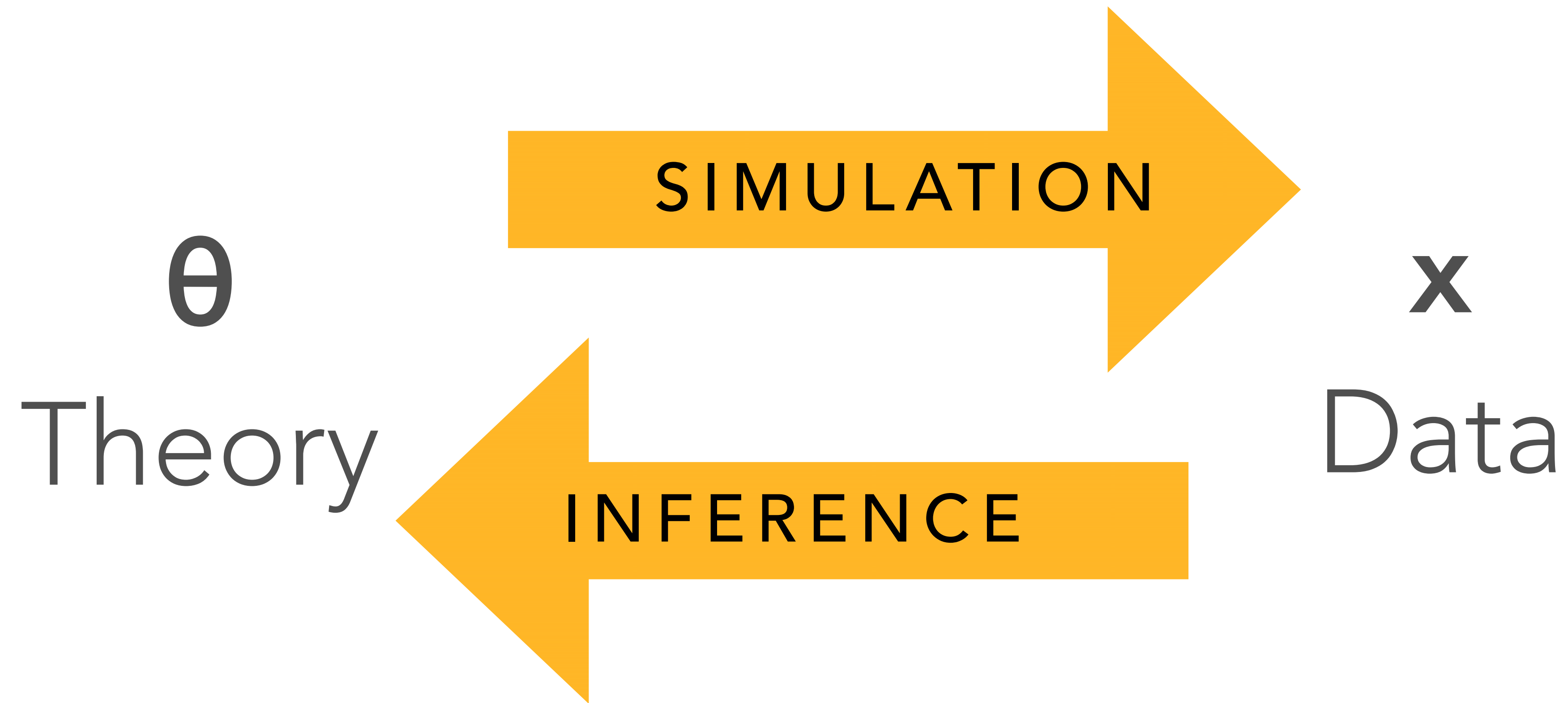
AI/ML for trigger & DAQ deserves a dedicated talk

- Largely skipping due to time

Major opportunities:

- Low-level anomaly detection to flag weird things that we might not trigger
- Real-time learning to adapt to run conditions
- New approaches to data compression
- Learn data distribution for what we do not trigger
- Assistant / automation for shifters, operations, data quality

Unifying Simulation and Inference



Simulating particle physics processes

Theory
parameters
 θ



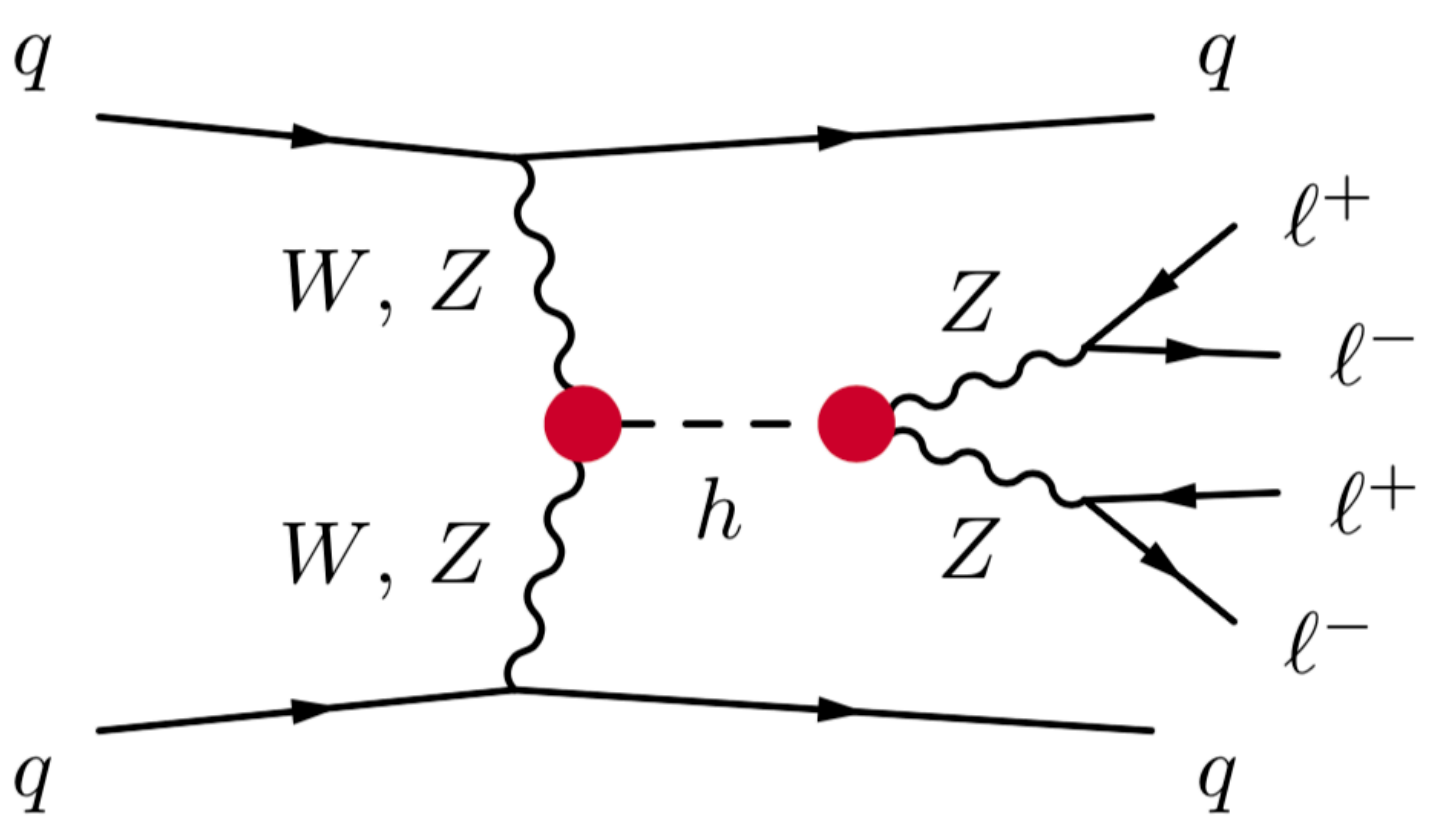
Evolution

Simulating particle physics processes

MC Truth / Latent variables

Parton-level momenta Theory parameters

$z_p \longleftarrow \theta$



Evolution

Simulating particle physics processes

MC Truth / Latent variables

Shower
splittings

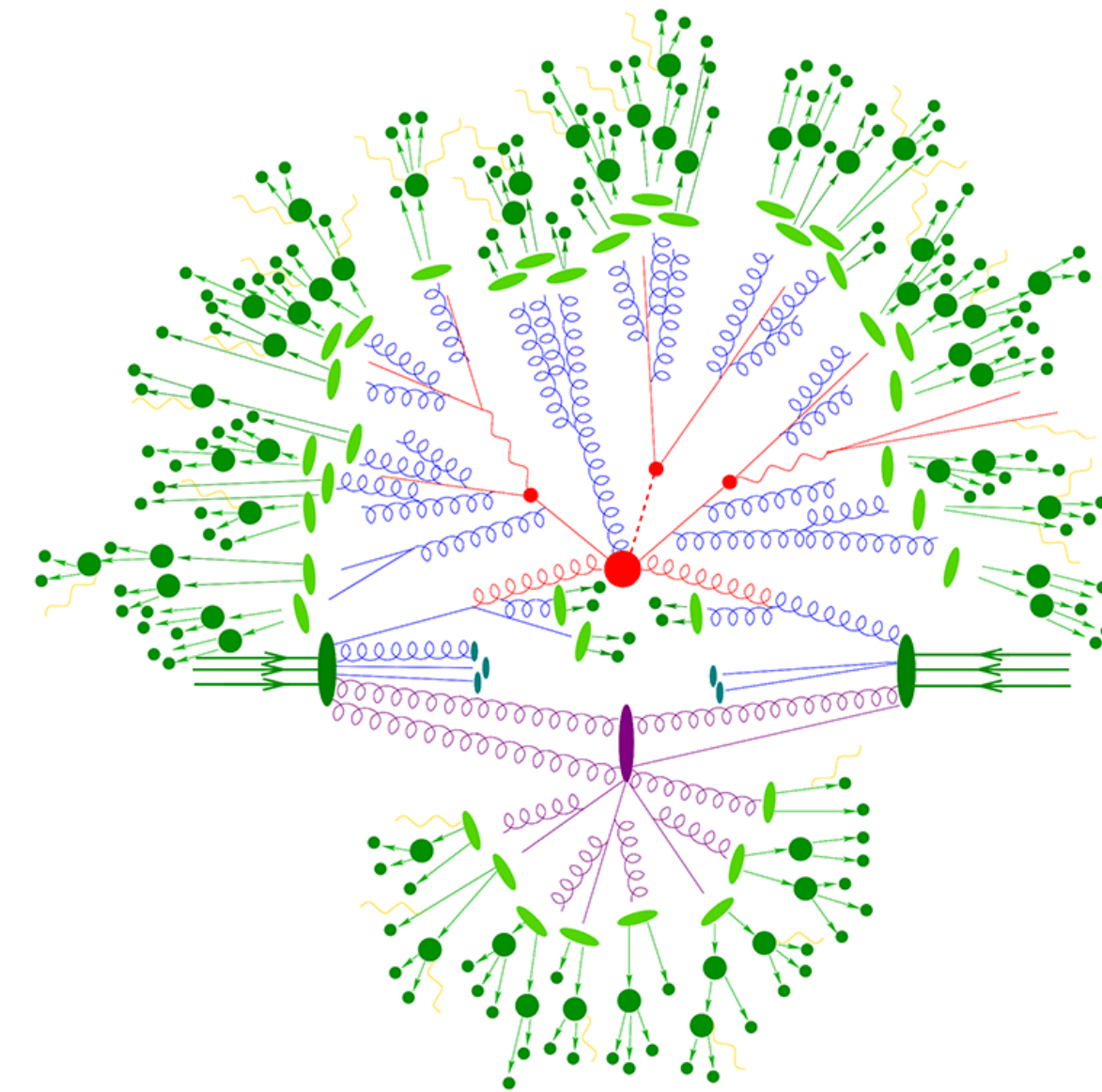
Parton-level
momenta

Theory
parameters

z_s

z_p

θ

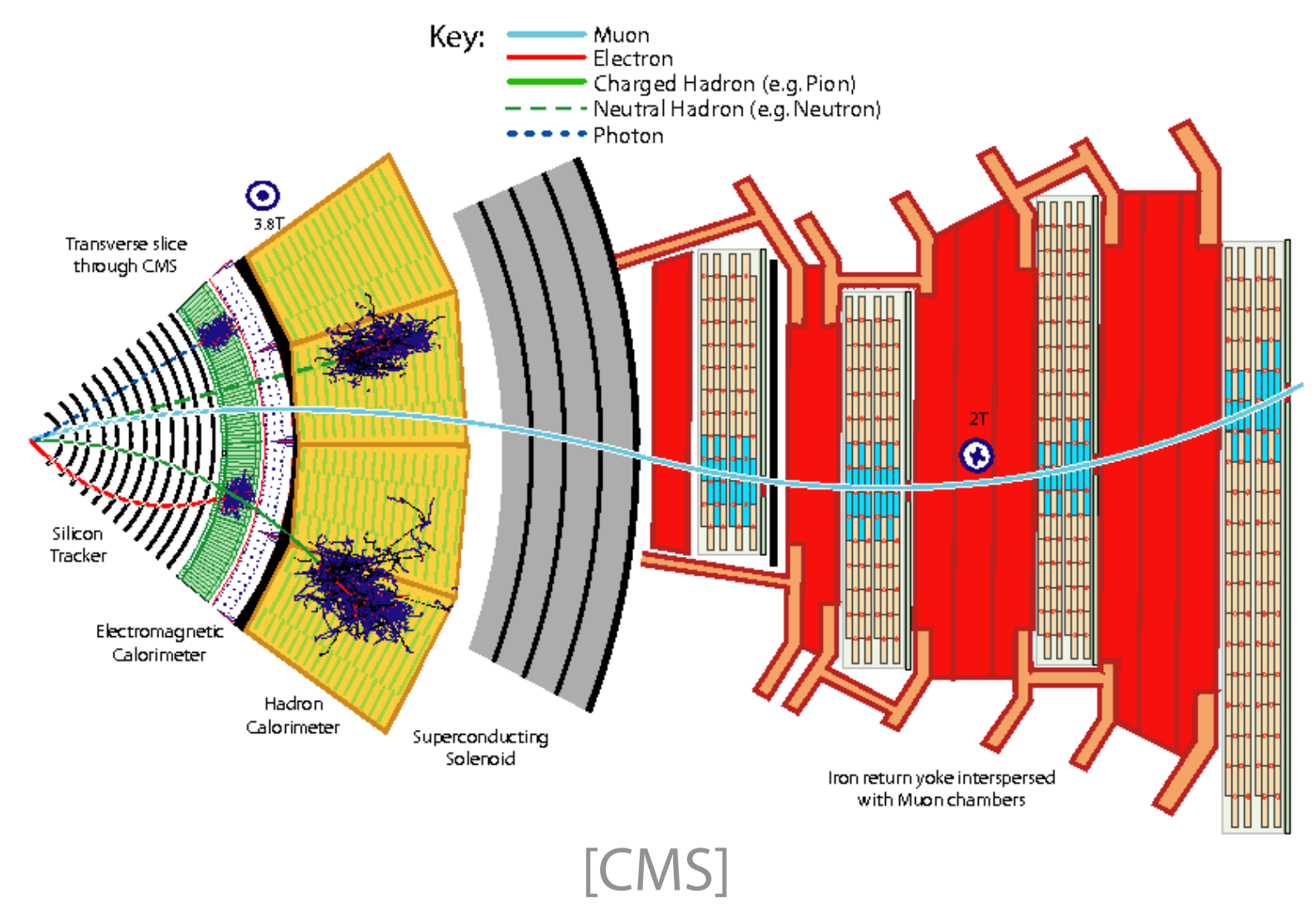
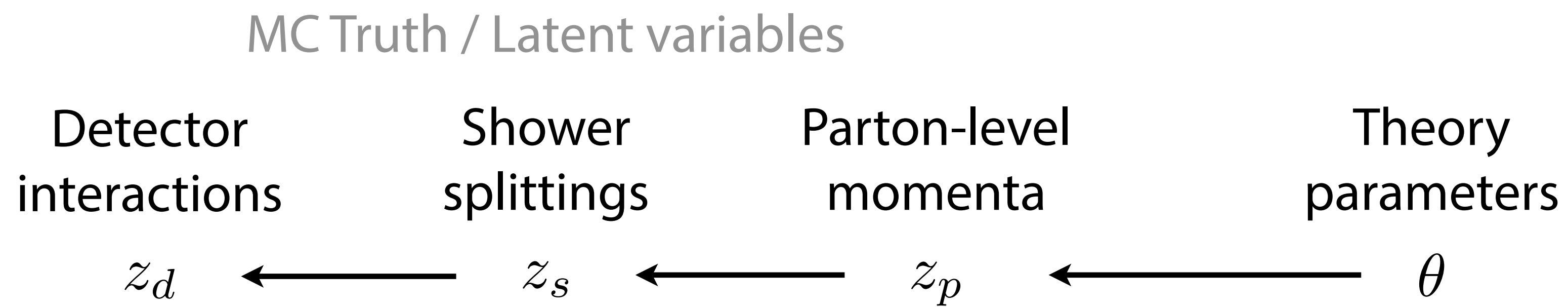


[F. Krauss]



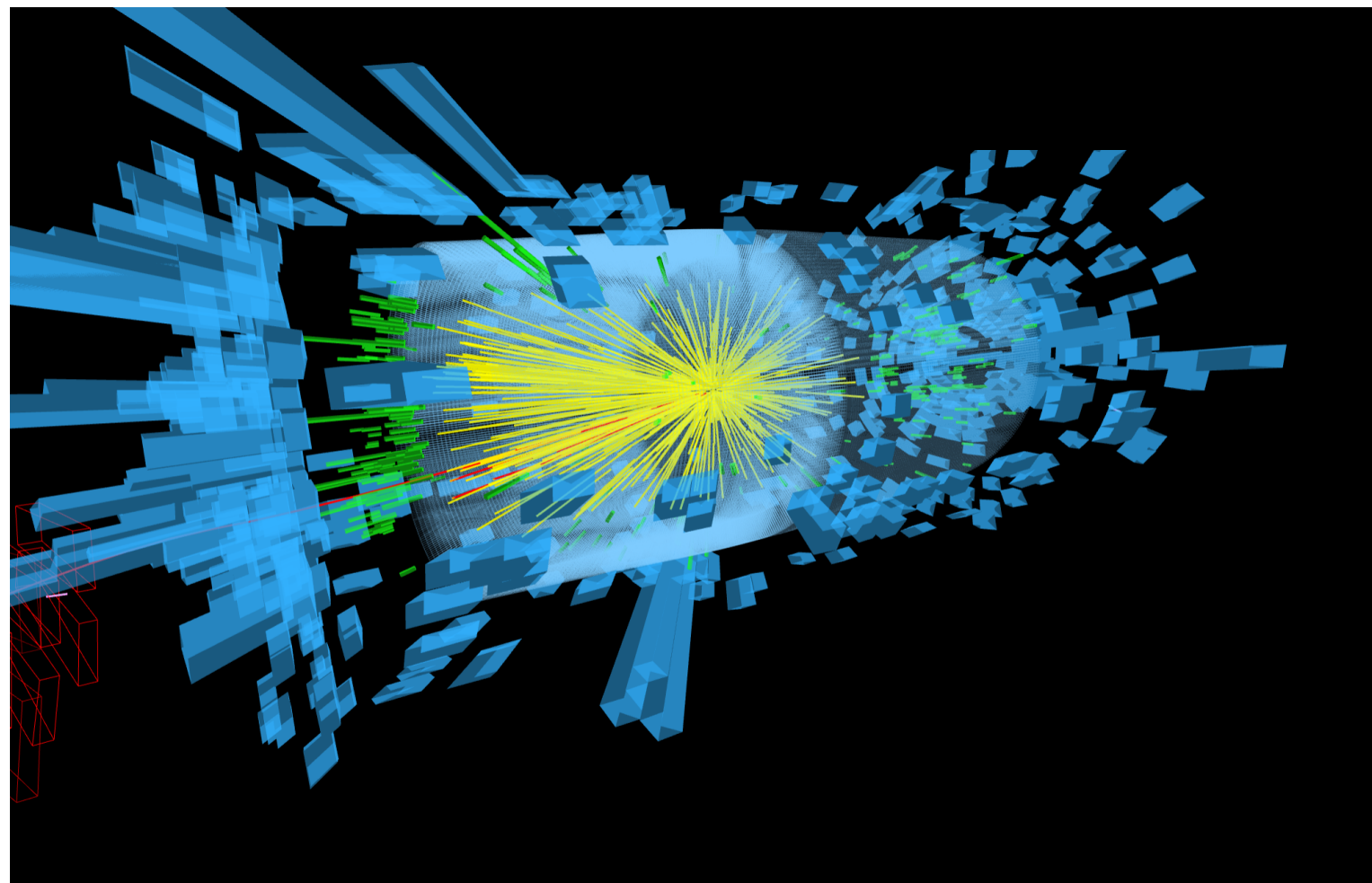
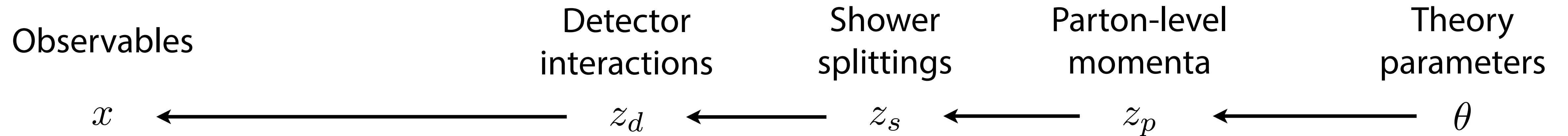
Evolution

Simulating particle physics processes



Simulating particle physics processes

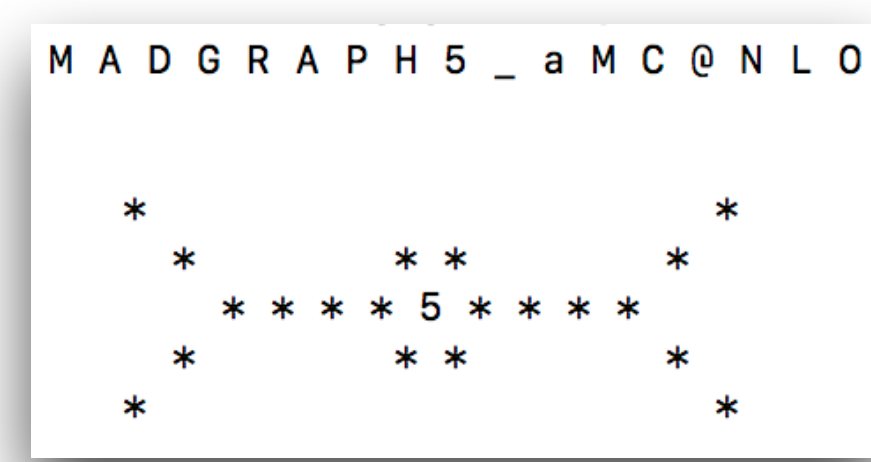
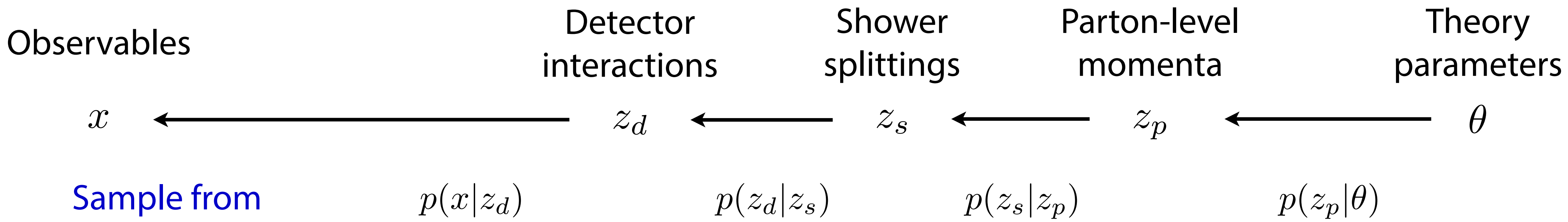
MC Truth / Latent variables




← Evolution

Simulating particle physics processes

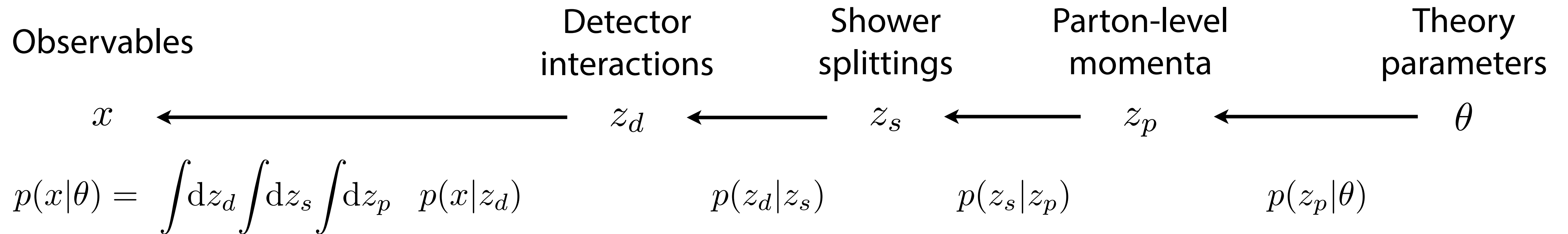
MC Truth / Latent variables



 Prediction (simulation)

Simulating particle physics processes

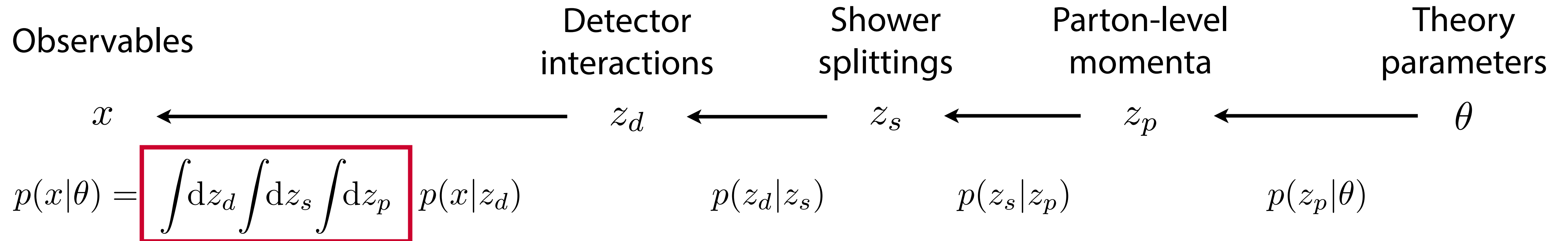
MC Truth / Latent variables



Inference

Simulating particle physics processes

MC Truth / Latent variables



It's infeasible to calculate the integral over this enormous space!

Inference



"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and **the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.**"

—PAUL DIRAC

ICML 2017 Workshop on Implicit Models

Workshop Aims

Probabilistic models are an important tool in machine learning. They form the basis for models that generate realistic data, uncover hidden structure, and make predictions. Traditionally, probabilistic models in machine learning have focused on prescribed models. Prescribed models specify a joint density over observed and hidden variables that can be easily evaluated. The requirement of a tractable density simplifies their learning but limits their flexibility --- several real world phenomena are better described by simulators that do not admit a tractable density. Probabilistic models defined only via the simulations they produce are called implicit models.

Arguably starting with generative adversarial networks, research on implicit models in machine learning has exploded in recent years. This workshop's aim is to foster a discussion around the recent developments and future directions of implicit models.

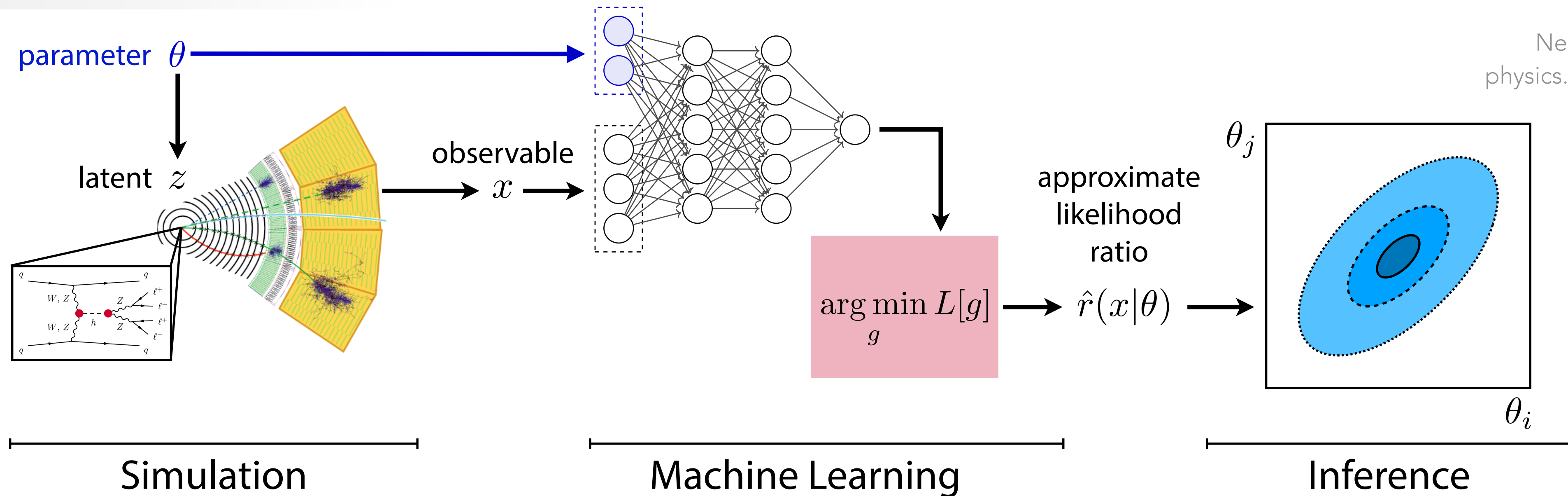
Implicit models have many applications. They are used in ecology where models simulate animal populations over time; they are used in phylogeny, where simulations produce hypothetical ancestry trees; they are used in physics to generate particle simulations for high energy processes. Recently, implicit models have been used to improve the state-of-the-art in image and content generation. Part of the workshop's focus is to discuss the commonalities among applications of implicit models.

Of particular interest at this workshop is to unite fields that work on implicit models. For example:

- **Generative adversarial networks** (a NIPS 2016 workshop) are implicit models with an adversarial training scheme.
- Recent advances in **variational inference** (a NIPS 2015 and 2016 workshop) have leveraged implicit models for more accurate approximations.
- **Approximate Bayesian computation** (a NIPS 2015 workshop) focuses on posterior inference for models with implicit likelihoods.
- Learning implicit models is deeply connected to **two sample testing, density ratio and density difference** estimation.

We hope to bring together these different views on implicit models, identifying their core challenges and combining their innovations.

Simulation-based inference

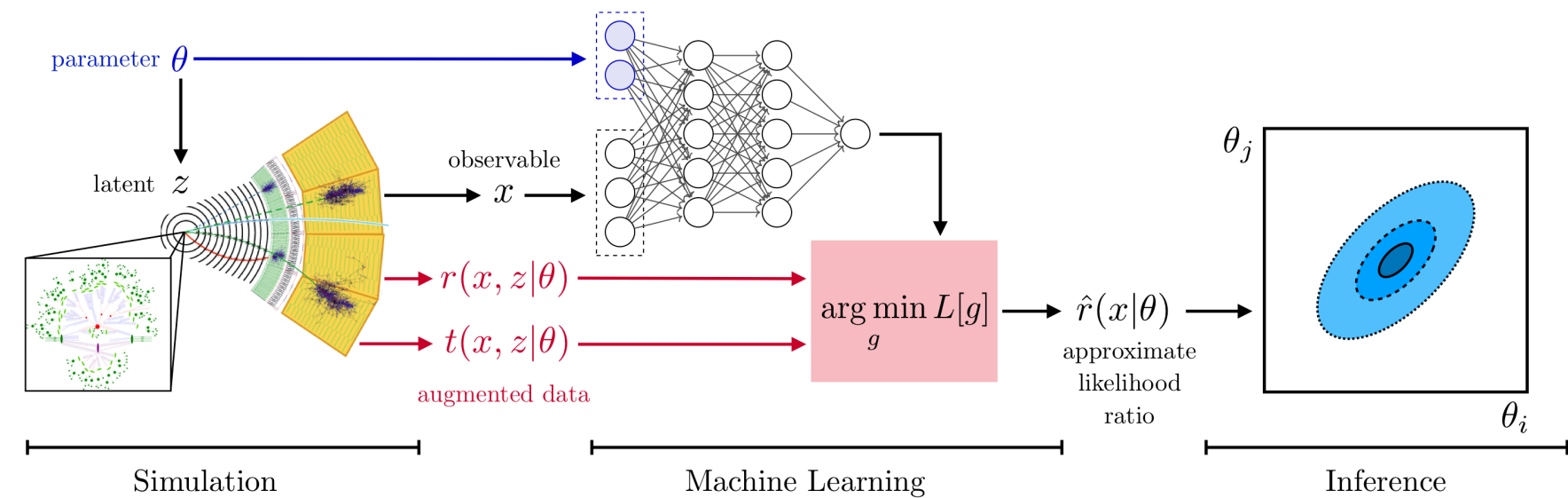


The NN is a **surrogate** for the likelihood (ratio) or posterior, which is used for inference
 Expert knowledge in simulator is transferred to surrogate via the learning process

What do we gain?

Massive gains in precision of a flagship measurement at the LHC !

Equivalent increasing data collected by LHC by several factors



MadMiner: Machine learning–based inference for particle physics

By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package

0.6.3

build passing

docs failing

chat on github

code style black

License MIT

DOI 10.5281/zenodo.1489147

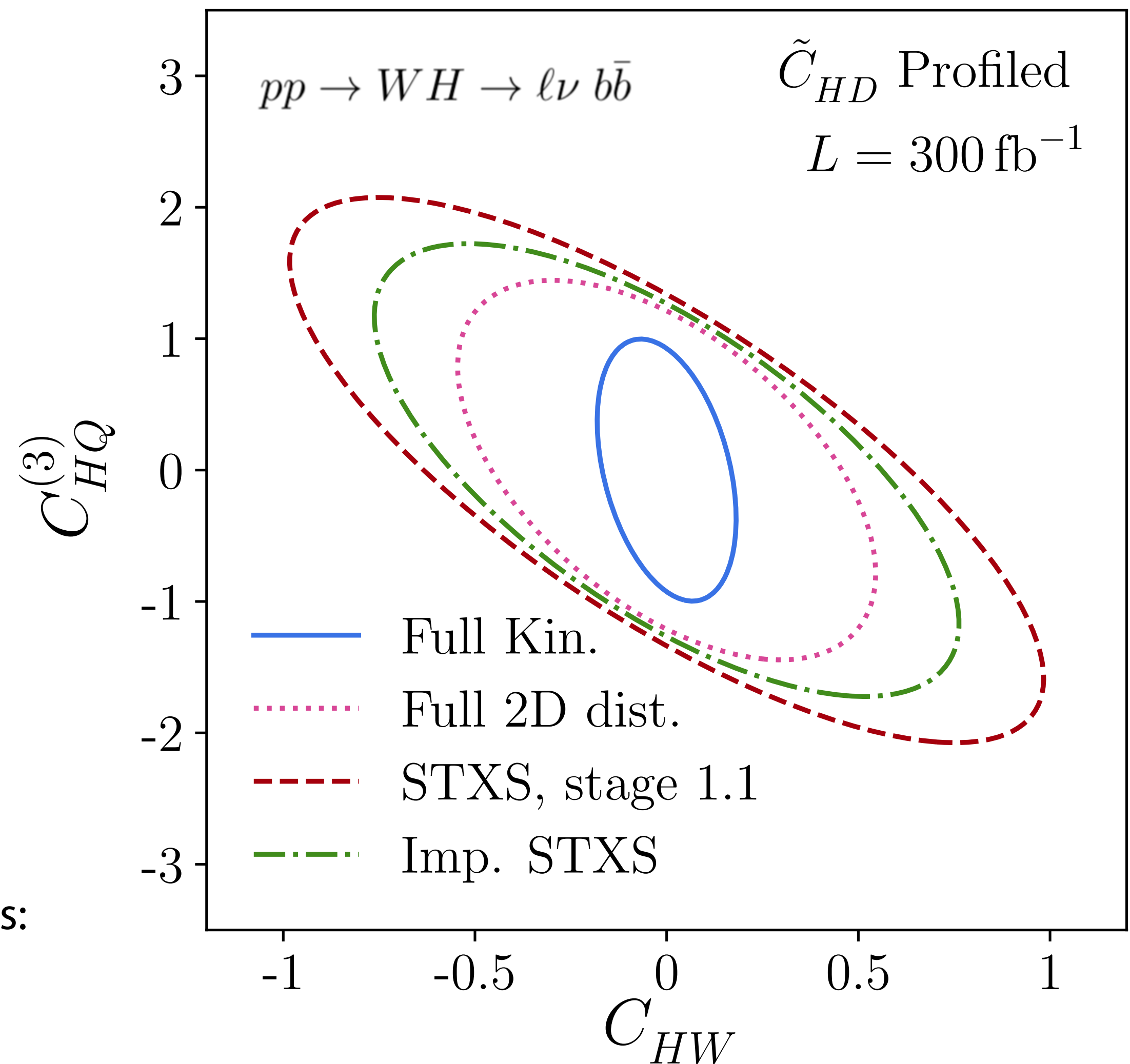
arXiv

1907.10621

Introduction

Particle physics processes are usually modeled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional

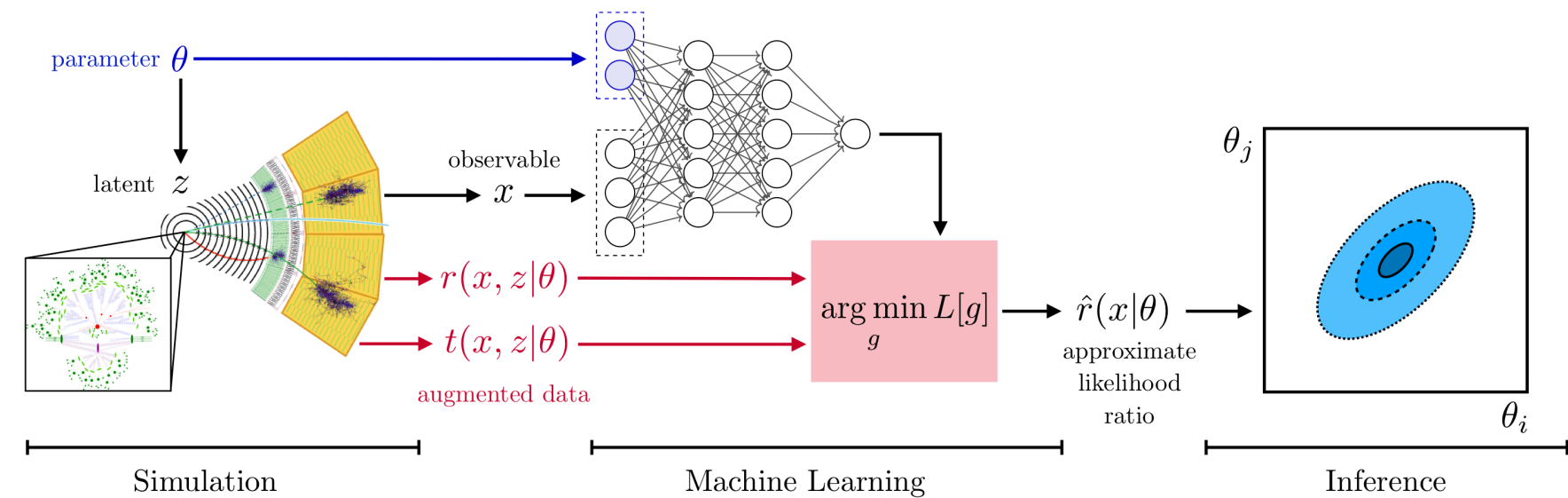
Dedicated software package interfacing with particle physics simulators:
github.com/johannbrehmer/madminer



What do we gain?

Massive gains in precision of a flagship measurement at the LHC !

Equivalent increasing data collected by LHC by several factors



MadMiner: Machine learning–based inference for particle physics

By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package

0.6.3

build

passing

docs

failing

chat

on

gitter

code style

black

License

MIT

DOI

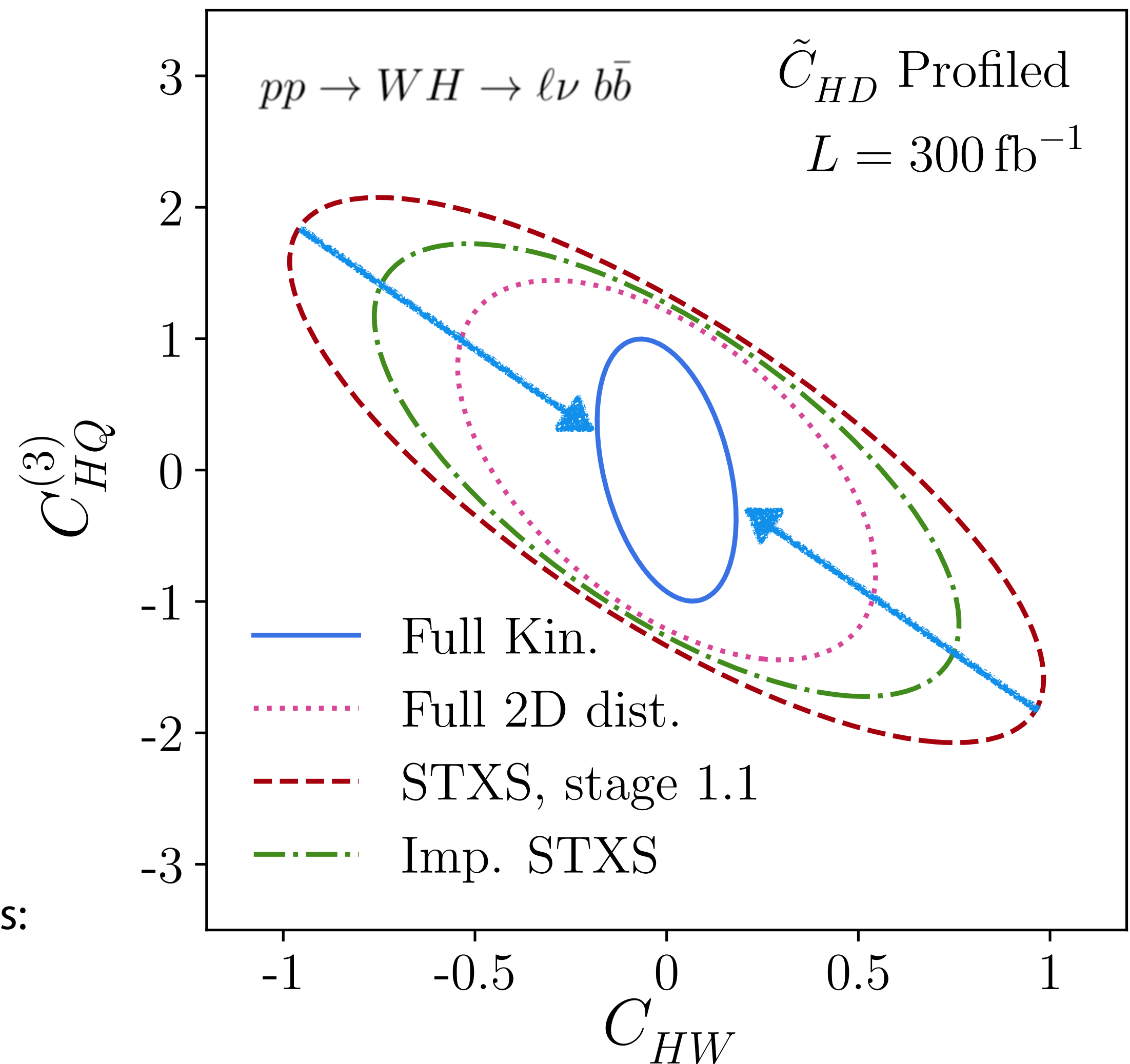
10.5281/zenodo.1489147

arXiv

1907.10621

Introduction

Particle physics processes are usually modeled with complex Monte-Carlo simulations of the hard process, parton shower, and detector interactions. These simulators typically do not admit a tractable likelihood function: given a (potentially high-dimensional) set of observables, it is usually not possible to calculate the probability of these observables for some model parameters. Particle physicists usually tackle this problem of "likelihood-free inference" by hand-picking a few "good" observables or summary statistics and filling histograms of them. But this conventional



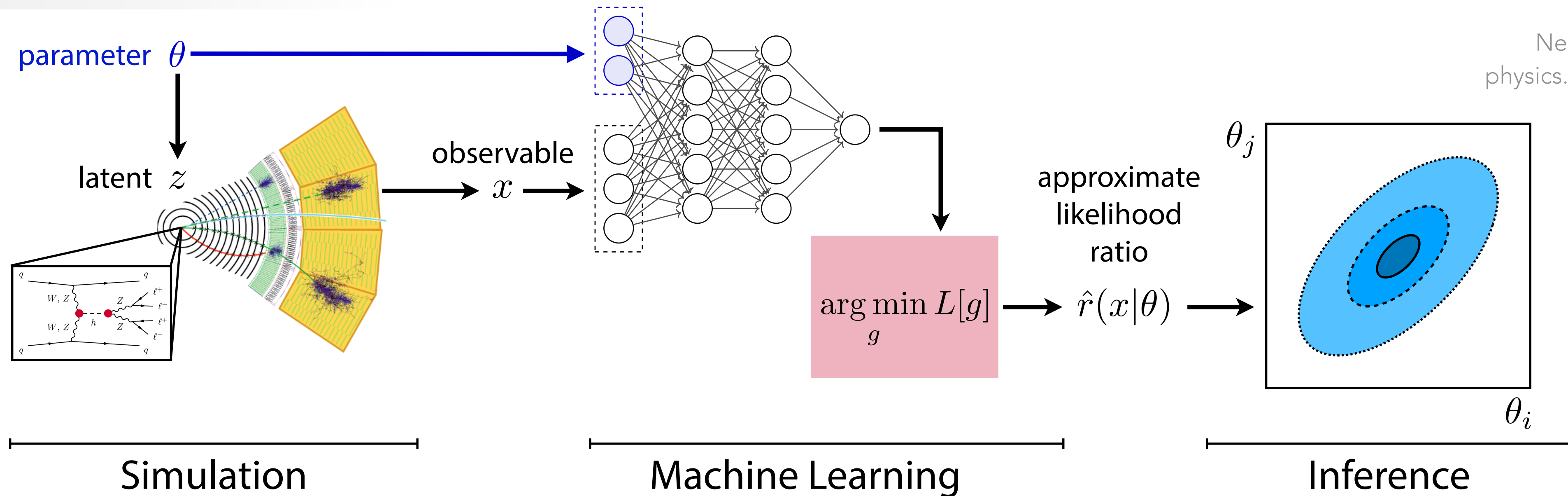
Dedicated software package interfacing with particle physics simulators:
github.com/johannbrehmer/madminer

[J. Brehmer, F. Kling, I. Espejo, K. Cranmer 1907.10621]

See also LHC EFT WG Report [arXiv:2211.08353](https://arxiv.org/abs/2211.08353)

[J. Brehmer, S. Dawson, S. Homiller, F. Kling, T. Plehn 1908.06980]

Is it radically different?



Allows us to work with low-level / “raw” data without summaries designed by experts

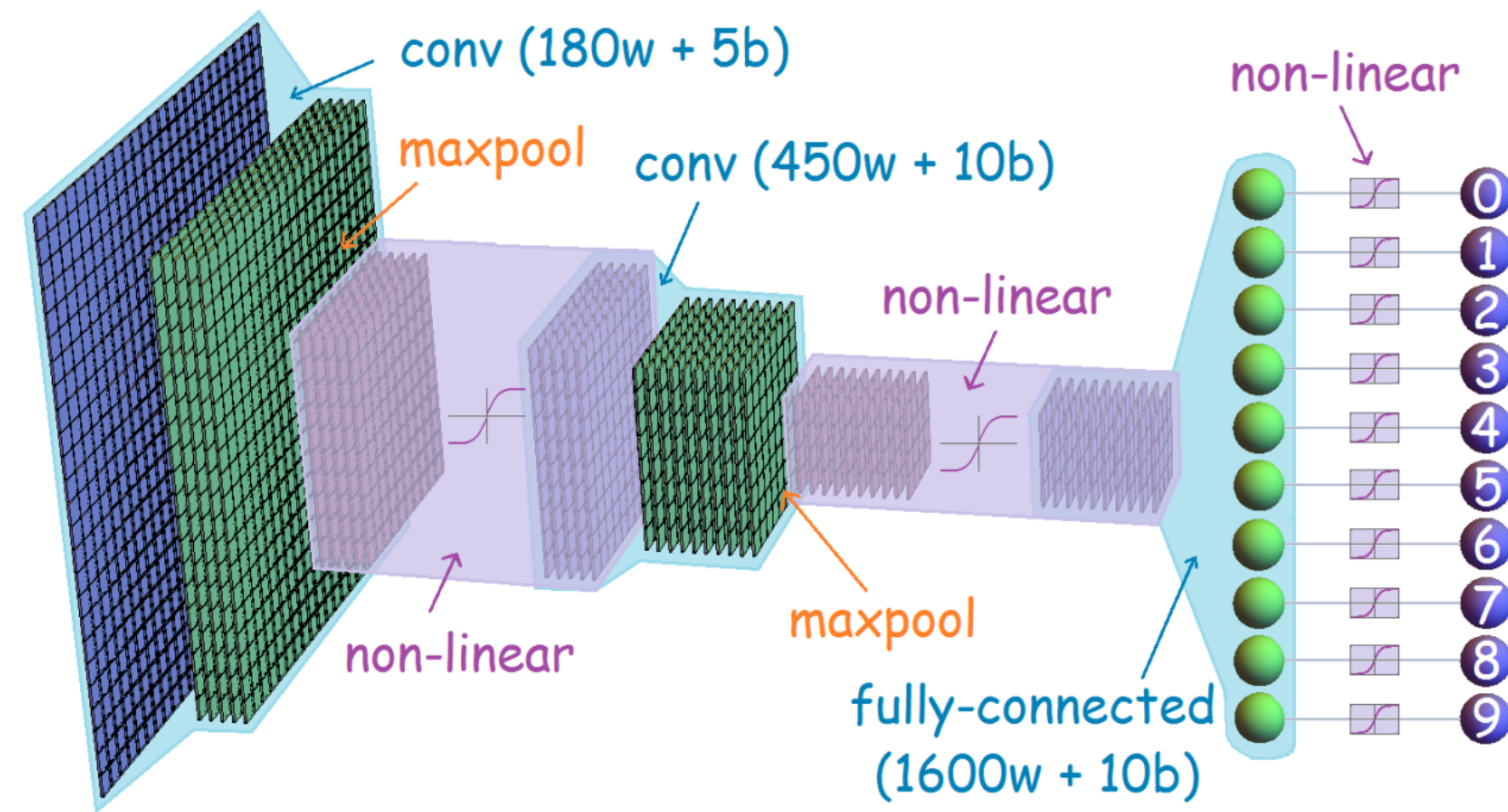
It is radical in the sense that the human role is different from traditional HEP analysis.

But, not any more radical than use of AI/ML for signal vs. background classifiers.

Interpretability requires new strategies and attitudes.

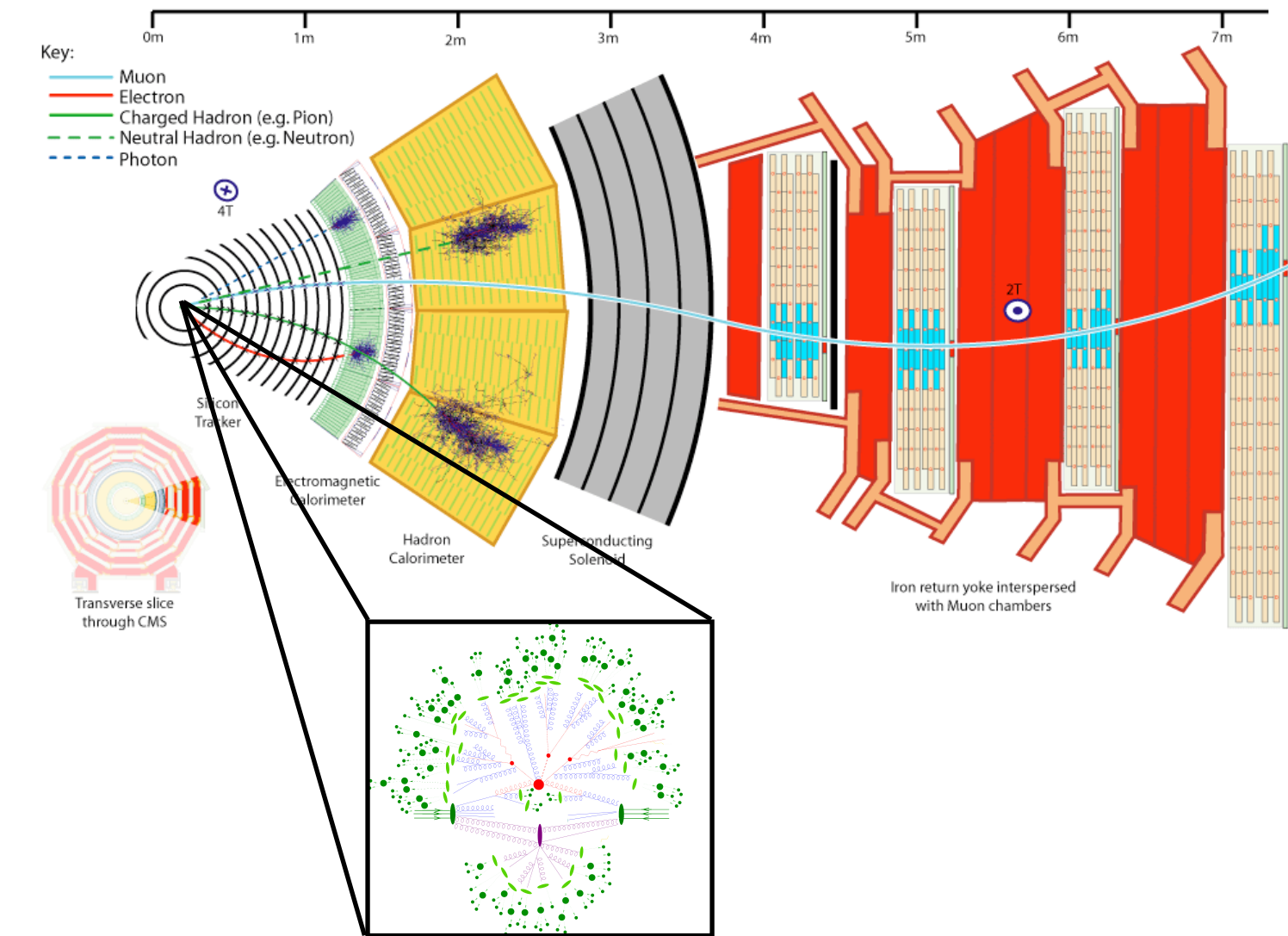
Two approaches simulation-based inference

Learn simulator (with deep learning)



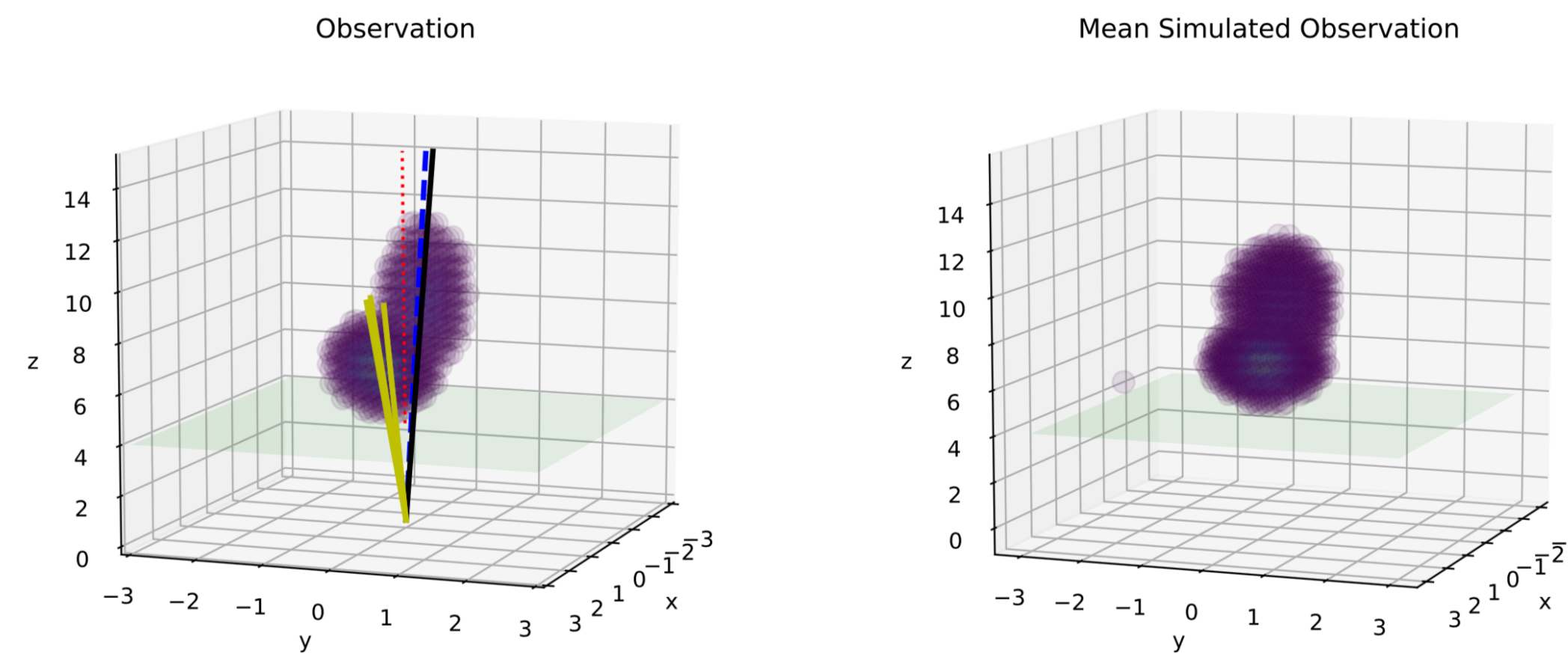
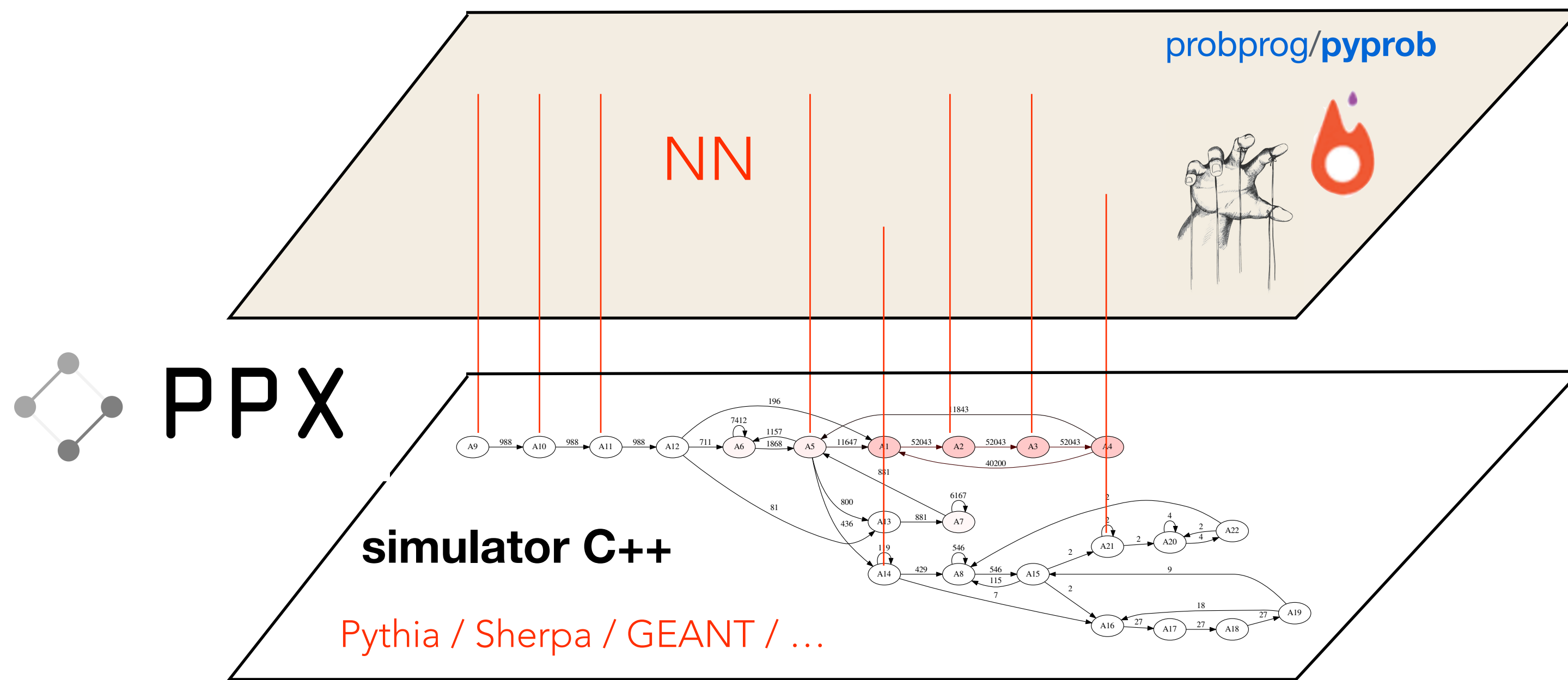
- Learn a NN surrogate
- avoid complexity of MC truth / latent variables in simulation

Control simulator (much more efficiently)



- Probabilistic programming
- Maintain ability to infer latent variables, but have to cope with that complexity

Probabilistic Programming: Use Generative AI to control the random numbers in the simulator to efficiently generate an output that matches the observed data.

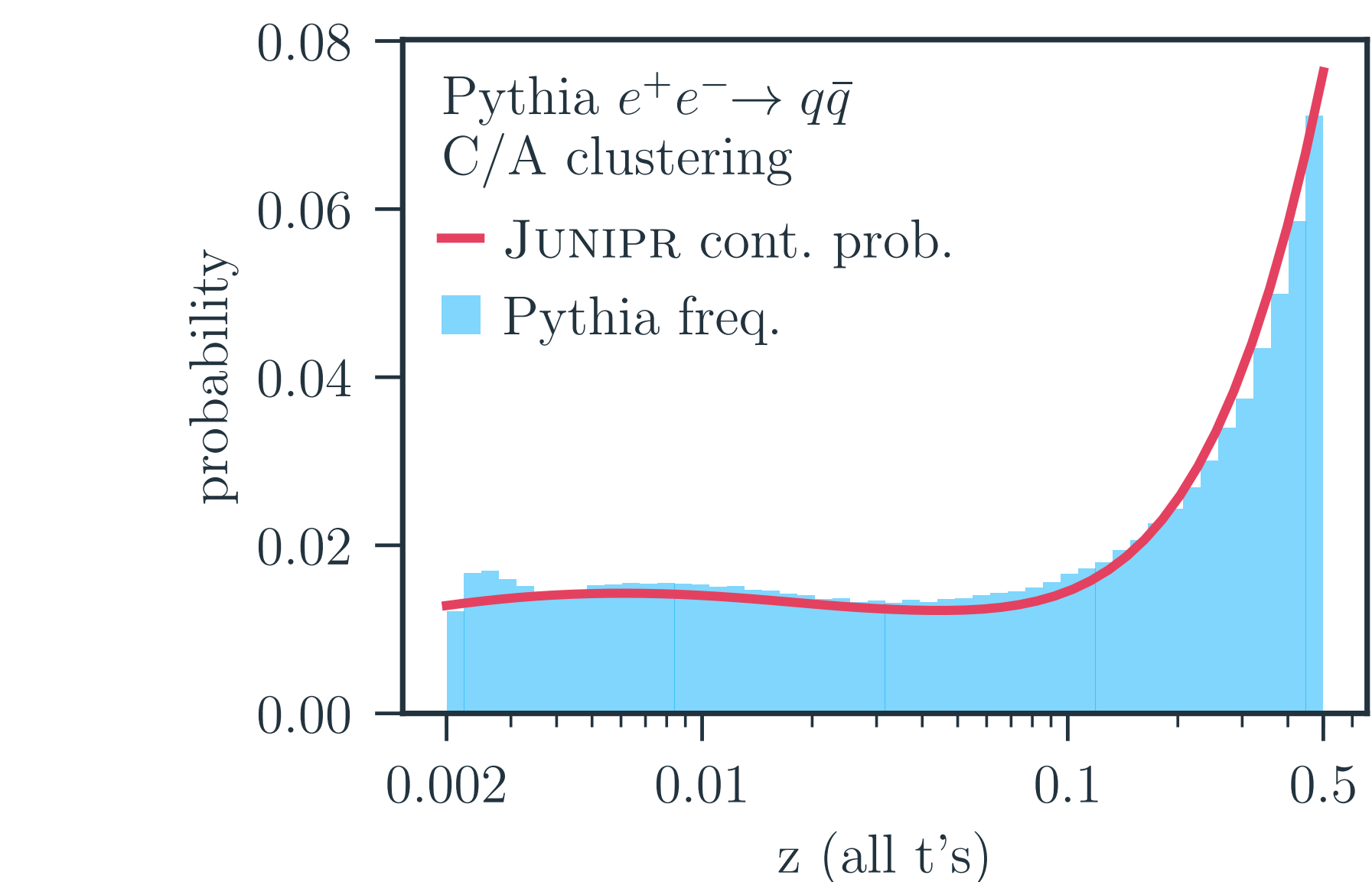
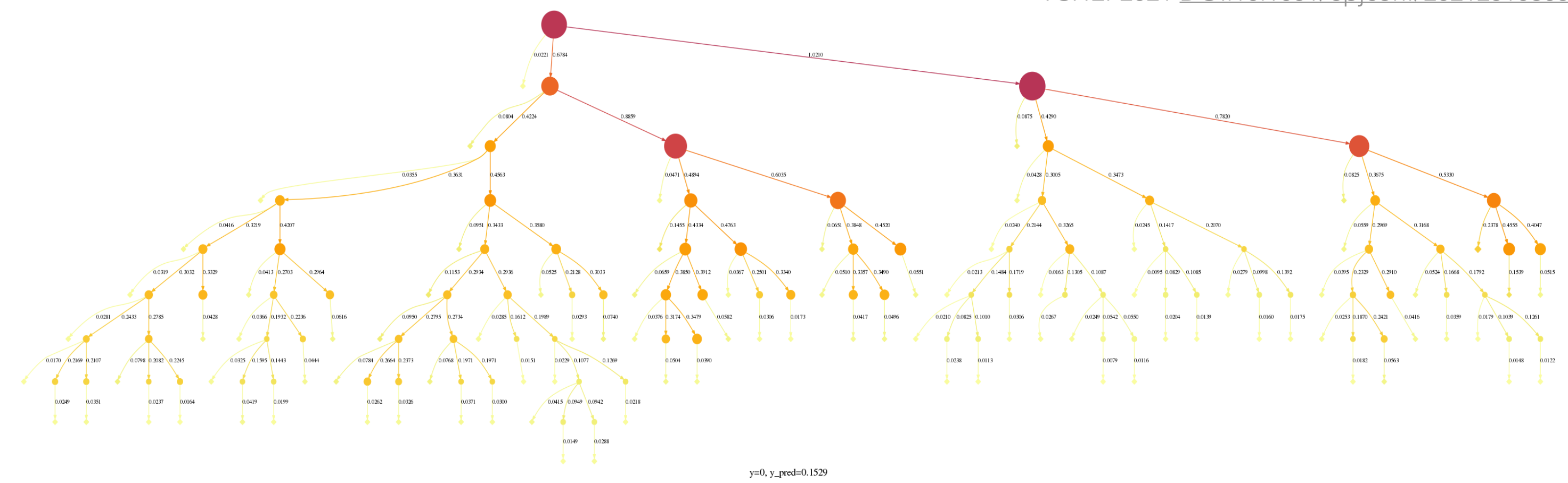


- **Augment** simulator so it can be controlled by external AI “guide” (C++, >1M lines of code)
- Use Generative AI techniques to control the internal randomness
- Radically different use of the simulator
- Still computationally expensive
- Enables “deep interpretability” & Bayesian Inference
- Preserves semantics of current simulation chain

JUNIPR is a generative model for jets

$$P_{\text{jet}}(\{p_1, \dots, p_n\}) = \left[\prod_{t=1}^{n-1} P_t(k_1^{(t+1)}, \dots, k_{t+1}^{(t+1)} | k_1^{(t)}, \dots, k_t^{(t)}) \right] \times P_n(\text{end} | k_1^{(n)}, \dots, k_n^{(n)}).$$

Autoregressive structure matches causal structure in traditional physics simulators



Latent variables are interpretable

- Can train on real data.
- Could imagine abandoning current parton shower algorithms with a more flexible data-driven shower.
- Loose interpretability?
- Would it extrapolate to different energies?
- There is a spectrum connecting Pythia and JUNIPR
- Training is conceptually equivalent to MC tuning.

Differentiable Programming

The core of modern AI is gradient-based optimization enabled by **automatic differentiation**. Incorporating this into our tools would lead to radical changes.

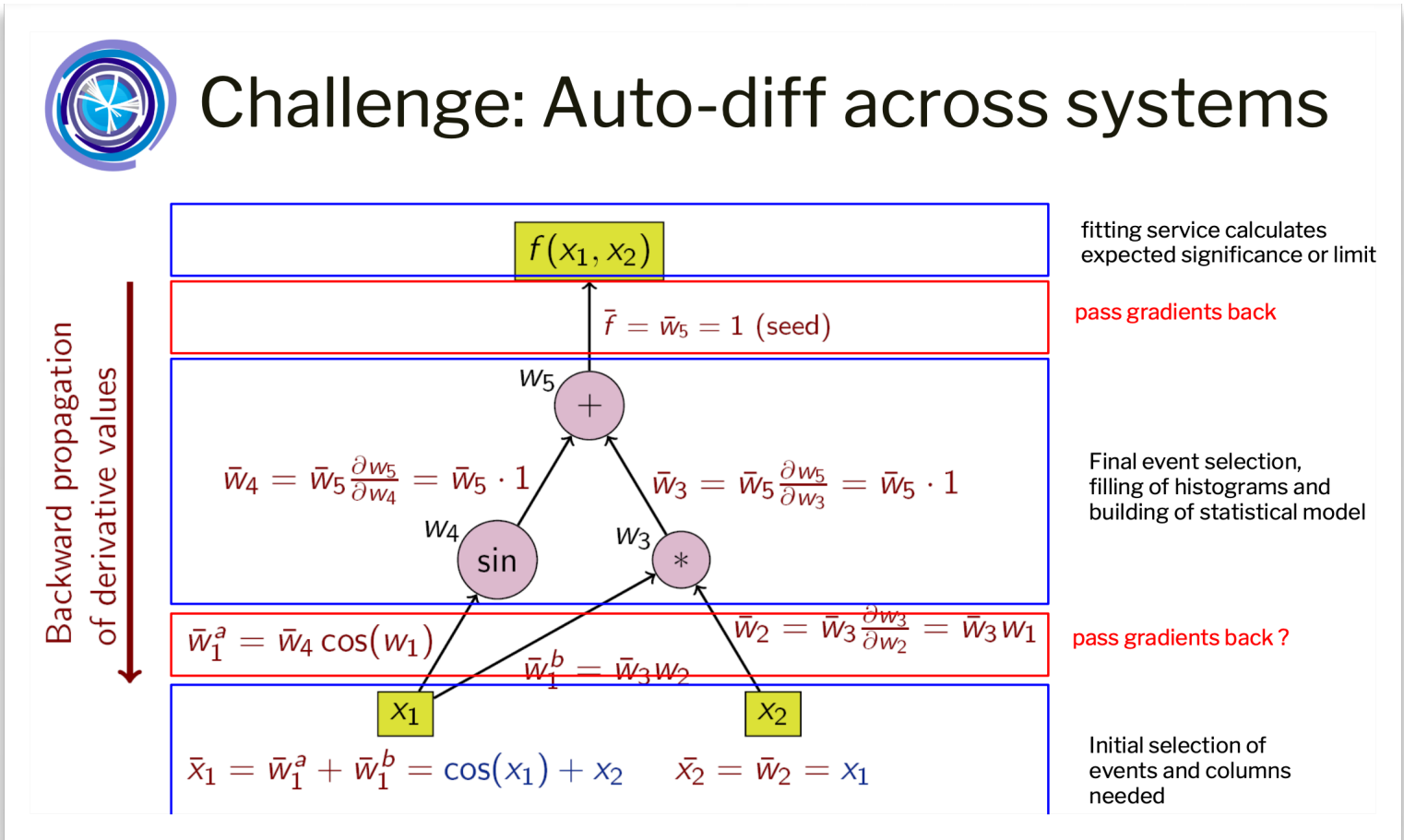
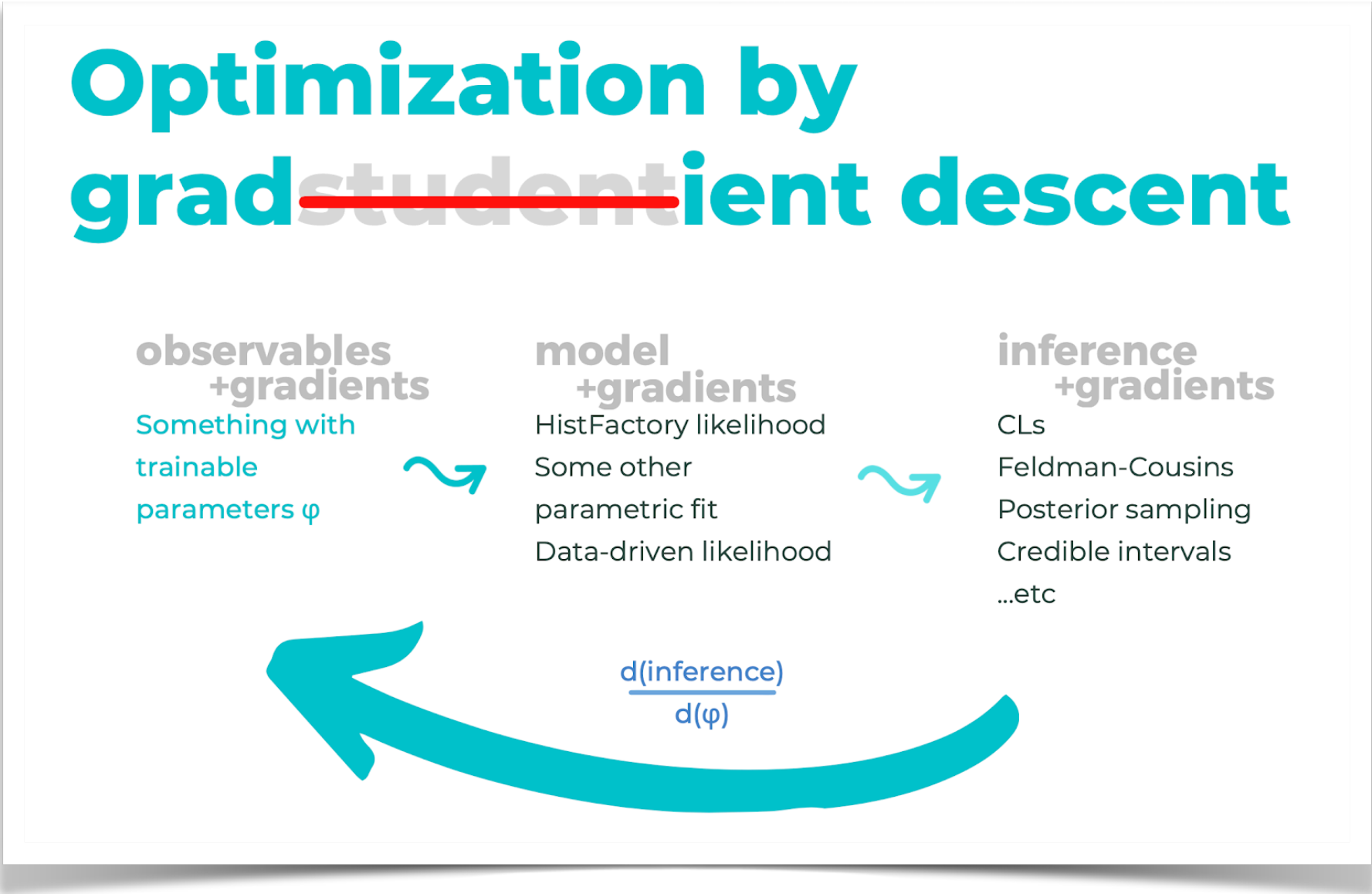
Differentiable Programming in High-Energy Physics

Atılım Güneş Baydin (Oxford), Kyle Cranmer (NYU), Matthew Feickert (UIUC), Lindsey Gray (FermiLab), Lukas Heinrich (CERN), Alexander Held (NYU) Andrew Melo (Vanderbilt) Mark Neubauer (UIUC), Jannicke Pearkes (Stanford), Nathan Simpson (Lund), Nick Smith (FermiLab), Giordon Stark (UCSC), Savannah Thais (Princeton), Vassil Vassilev (Princeton), Gordon Watts (U. Washington)

August 31, 2020

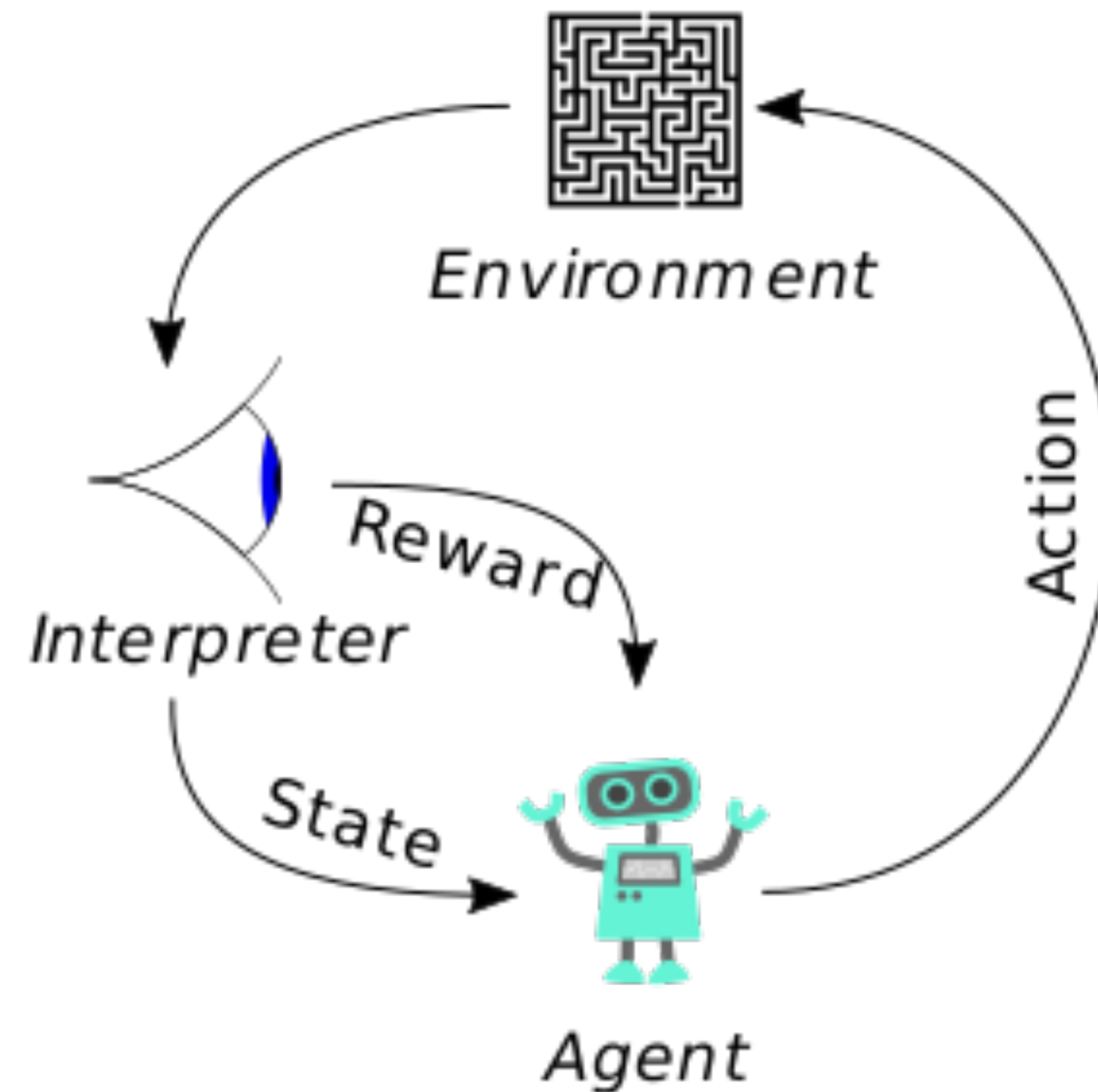
Abstract

A key component to the success of deep learning is the use of gradient-based optimization. Deep learning practitioners compose a variety of modules together to build a complex computational pipeline that may depend on millions or billions of parameters. Differentiating such functions is enabled through a computational technique known as automatic differentiation. The success of deep learning has led to an abstraction known as **differentiable programming**, which is being promoted to a first-class citizen in many programming languages and data analysis frameworks. This often involves replacing some common non-differentiable operations (eg. binning, sorting) with relaxed, differentiable analogues. The result is a system that can be optimized from end-to-end using efficient gradient-based optimization algorithms. A *differentiable analysis* could be optimized in this way — basic cuts to final fits all taking into account full systematic errors and automatically analyzed. This Snowmass LOI outlines the potential advantages and challenges of adopting a differentiable programming paradigm in high-energy physics.



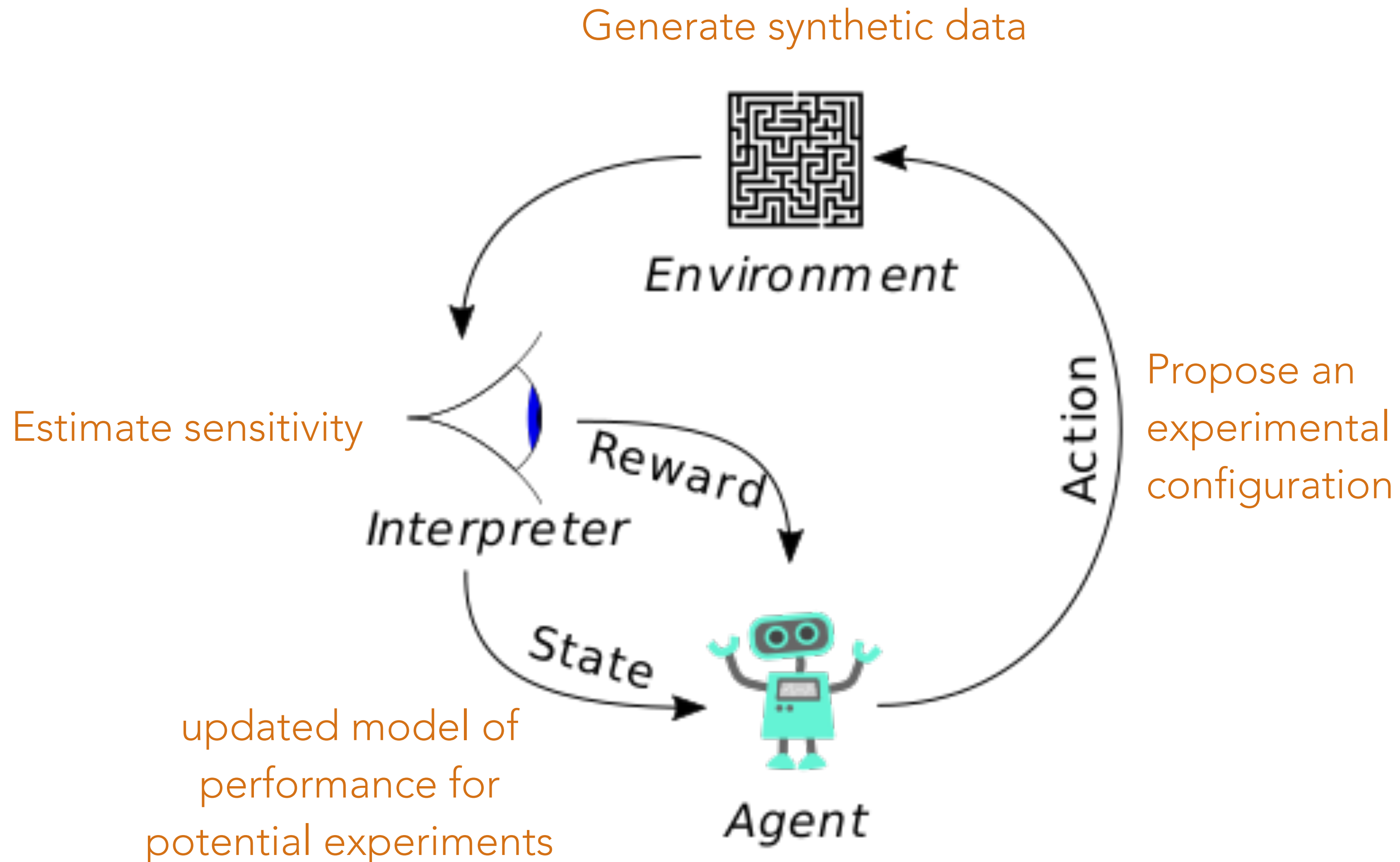
Closing the loop: Experimental Design

Scientist trying to decide what experiment to do next



Closing the loop: Experimental Design

Scientist trying to decide what experiment to do next



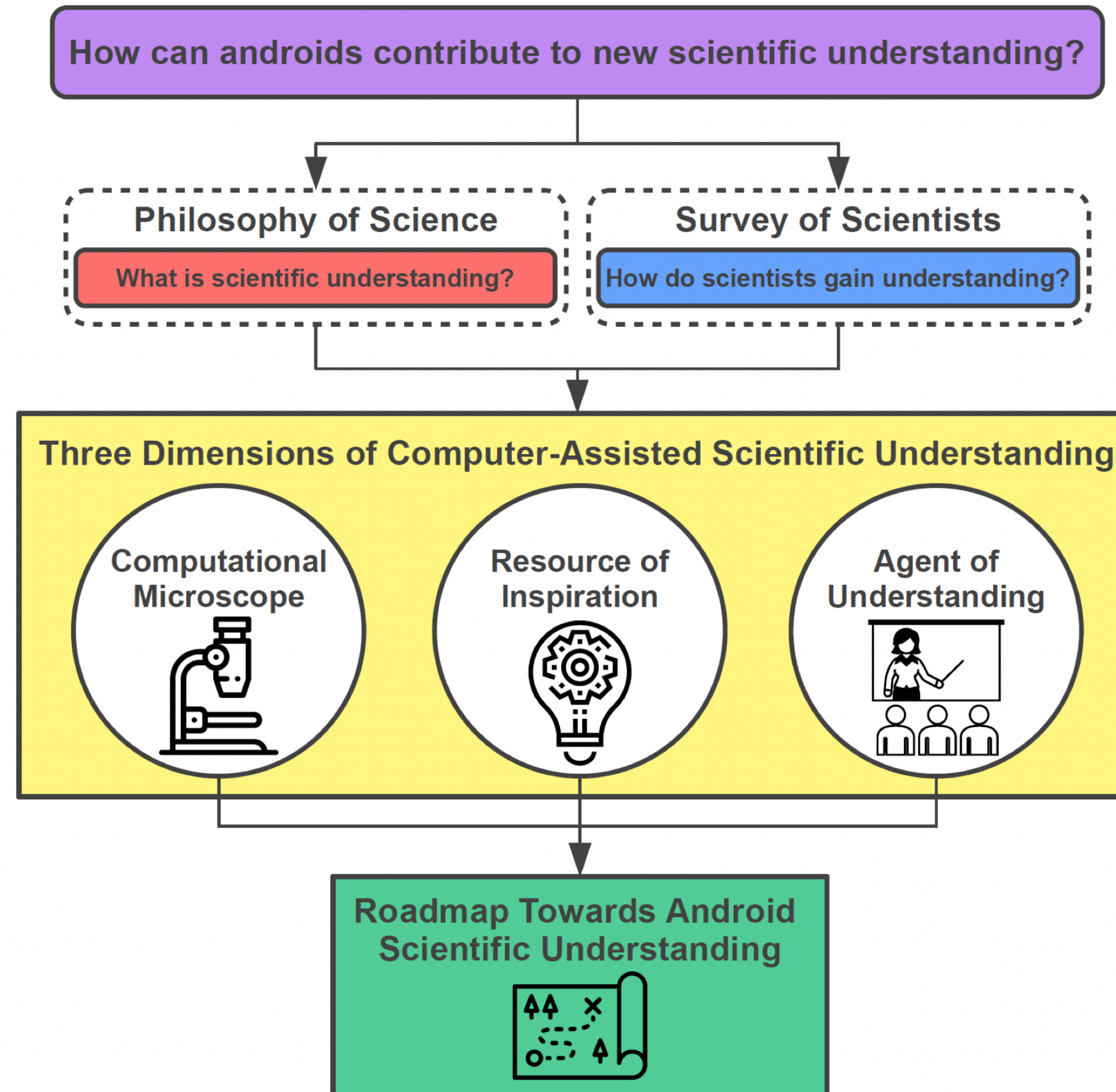
Scientific Understanding

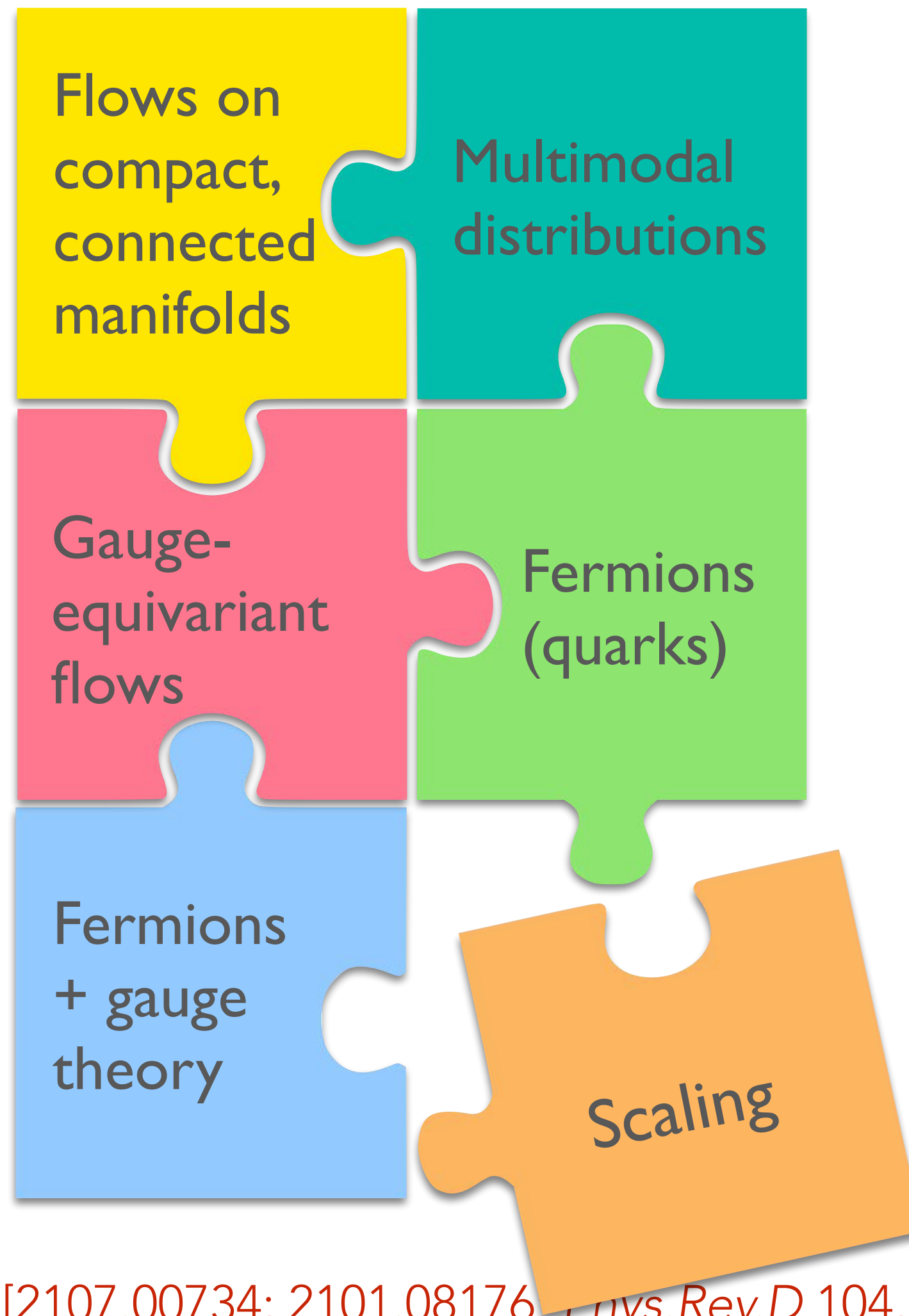
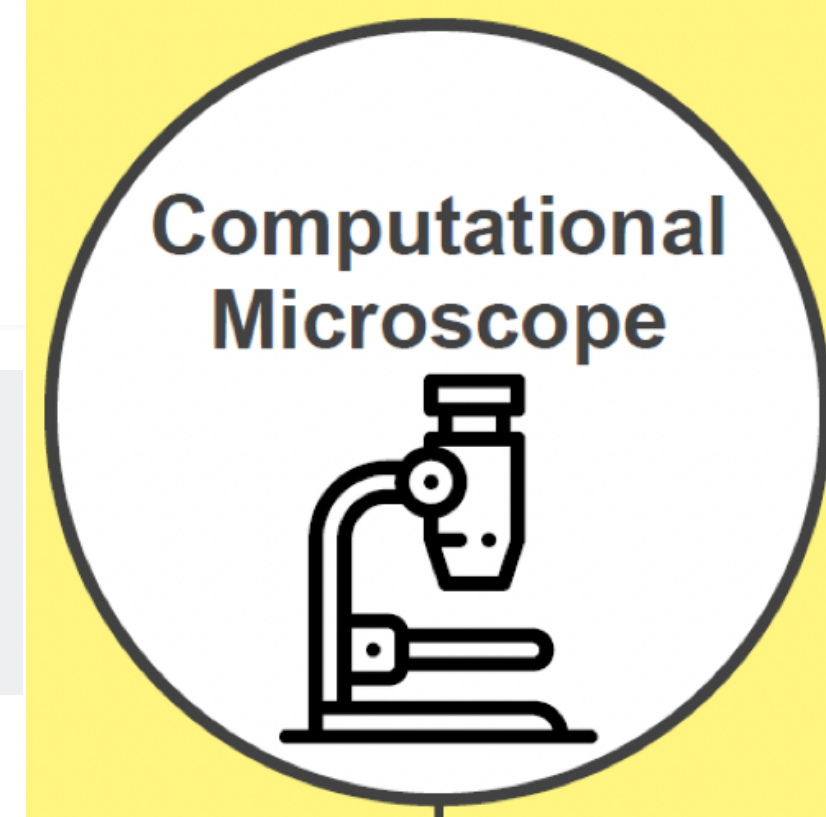
On scientific understanding with artificial intelligence

Mario Krenn,^{1, 2, 3, 4, *} Robert Pollice,^{2, 3} Si Yue Guo,² Matteo Aldeghi,^{2, 3, 4} Alba Cervera-Lierta,^{2, 3} Pascal Friederich,^{2, 3, 5} Gabriel dos Passos Gomes,^{2, 3} Florian Häse,^{2, 3, 4, 6} Adrian Jinich,⁷ AkshatKumar Nigam,^{2, 3} Zhenpeng Yao,^{2, 8, 9, 10} and Alán Aspuru-Guzik^{2, 3, 4, 11, †}

Imagine an oracle that correctly predicts the outcome of every particle physics experiment, the products of every chemical reaction, or the function of every protein. Such an oracle would revolutionize science and technology as we know them. However, as scientists, we would not be satisfied with the oracle itself. We want more. We want to comprehend how the oracle conceived these predictions. This feat, denoted as scientific understanding, has frequently been recognized as the essential aim of science. Now, the ever-growing power of computers and artificial intelligence poses one ultimate question: How can advanced artificial systems contribute to scientific understanding or achieve it autonomously?

Computer-Assisted Scientific Understanding





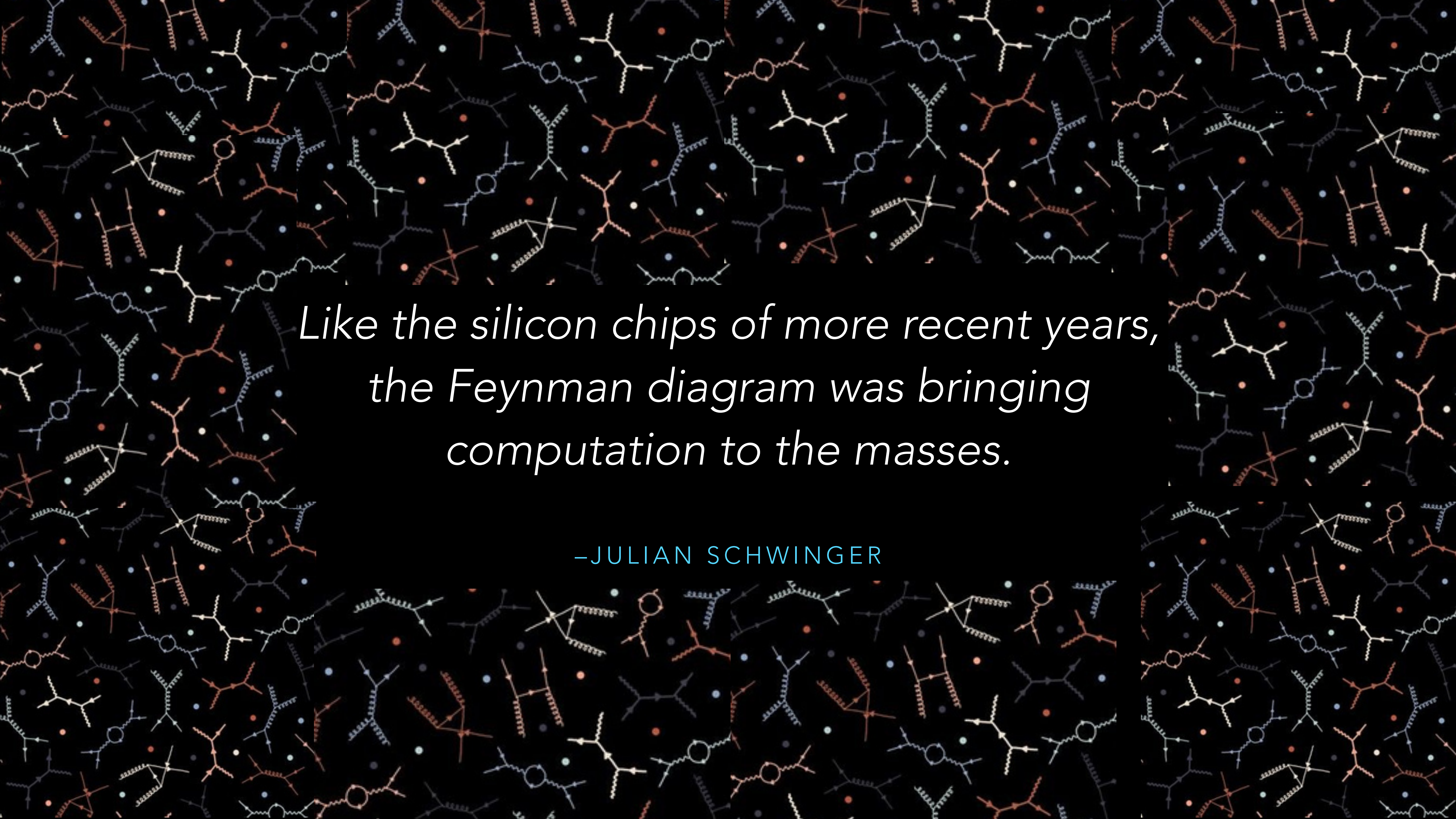
Machine learning for QCD

- Provably-exact machine-learning-accelerated sampling algorithm
- Orders of magnitude more **efficient** than conventional algorithms overcoming critical slowing-down
- **Unbiased** results where traditional approaches fail

Deployment for state-of-the-art QCD
scheduled for Aurora 2023 first science time

AURORA | EARLY SCIENCE
PROGRAM FOR
DATA & LEARNING

[2107.00734; 2101.08176, *Phys.Rev.D* 104, 114507; *Phys.Rev.D* 103, 074504 (2021); *Phys.Rev.Lett.* 125, 121601; PMLR 8083-8092 (2020); *Phys.Rev.D* 100, 034515 (2019); *Phys.Rev.D* 97, 094506 (2018)]



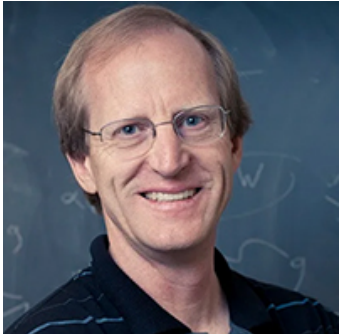
*Like the silicon chips of more recent years,
the Feynman diagram was bringing
computation to the masses.*

—JULIAN SCHWINGER

AI 4 Amplitudes



Kyle Cranmer



Lance Dixon



Garrett Merz



Tianji Cai

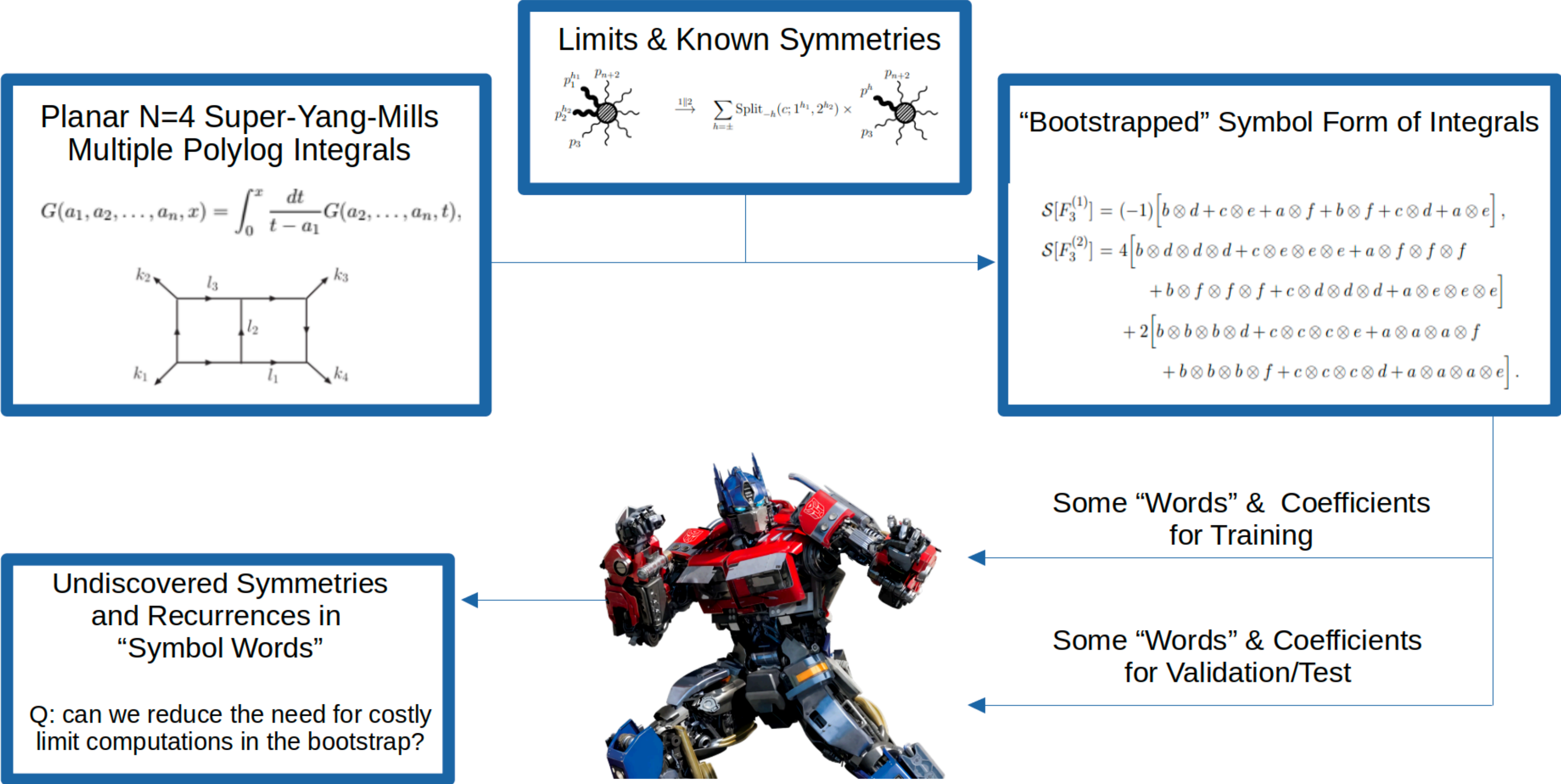


Matthias Wilhelm



François Charton

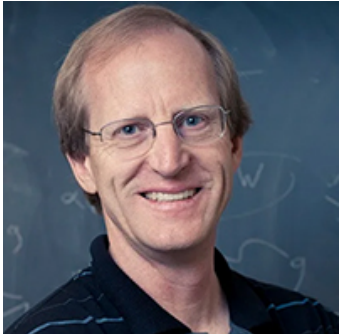
Like an NP-complete problem, the answer is hard to find, but easy to verify



AI 4 Amplitudes



Kyle Cranmer



Lance Dixon



Garrett Merz



Tianji Cai

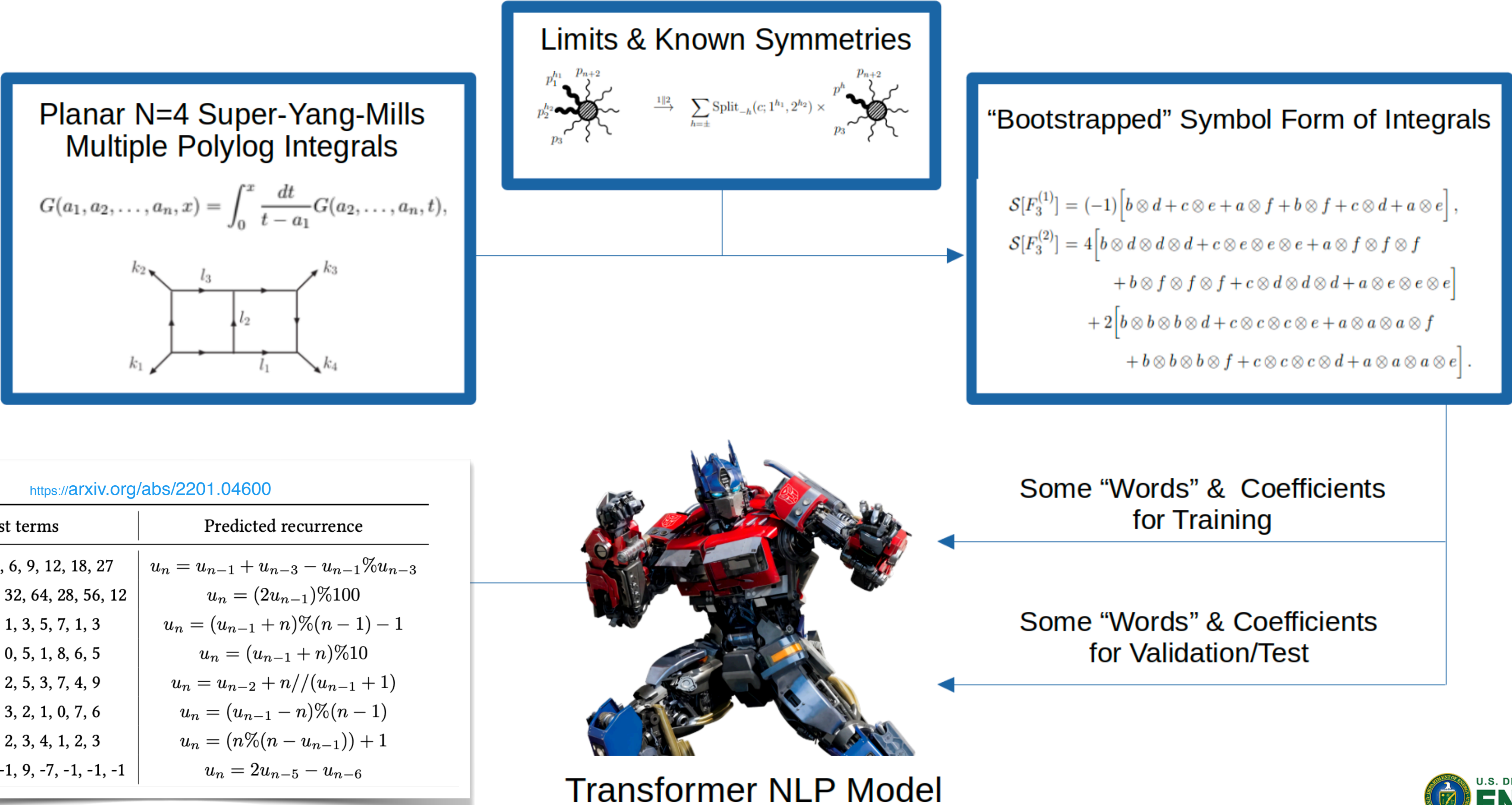


Matthias Wilhelm



François Charton

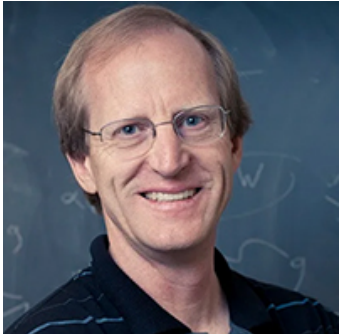
Like an NP-complete problem, the answer is hard to find, but easy to verify



AI 4 Amplitudes



Kyle Cranmer



Lance Dixon



Garrett Merz



Tianji Cai

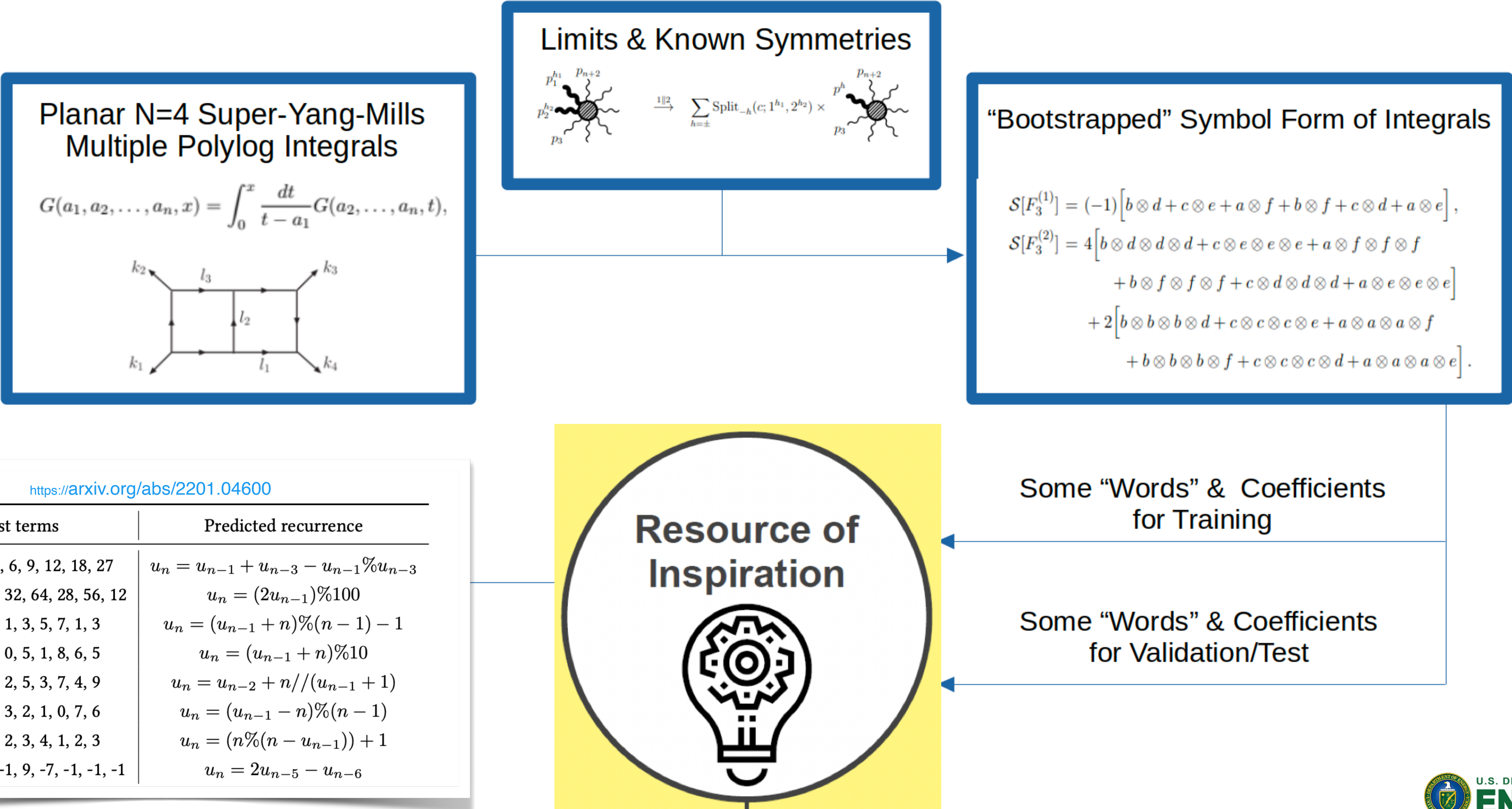


Matthias Wilhelm



François Charton

Like an NP-complete problem, the answer is hard to find, but easy to verify



Conclusions

Perhaps it is a useful **ongoing** exercise to think through more radical, speculative changes

- What might be enabled if we embraced a dramatically different approach?
- What would we lose?
- How would those changes impact the computational, organizational, and epistemological nature of the field?

There are many opportunities:

- some are easy, some are hard, some are unnecessarily hard
- some are worth it, some are not

Do they reveal overly rigid assumptions about our approach?

- We should attempt to be nimble as AI/ML will continue to surprise and disrupt us