HEP Simulator issues tackled with Bayesian Inference

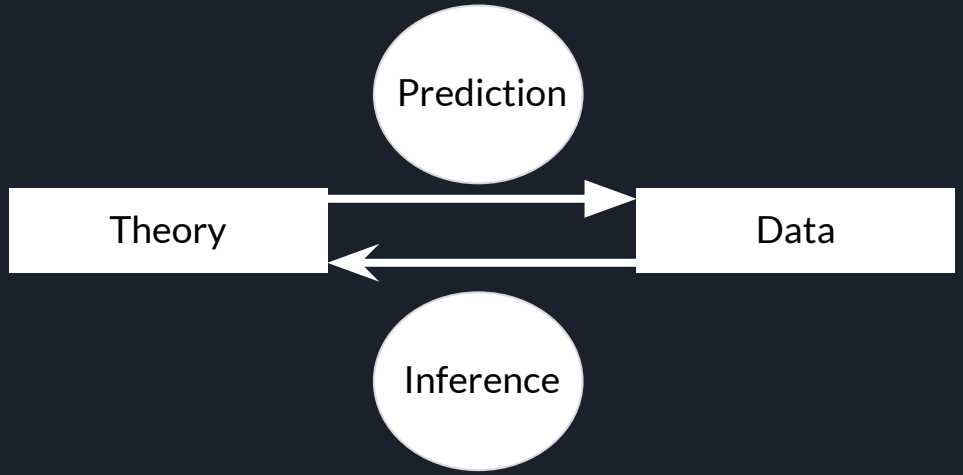
Manuel Szewc
IJS
Bayesian Inference in High Energy Physics,
IPPP, Durham 26/05/2022

Event generators are fundamental tools for all tasks involved in HEP experiments

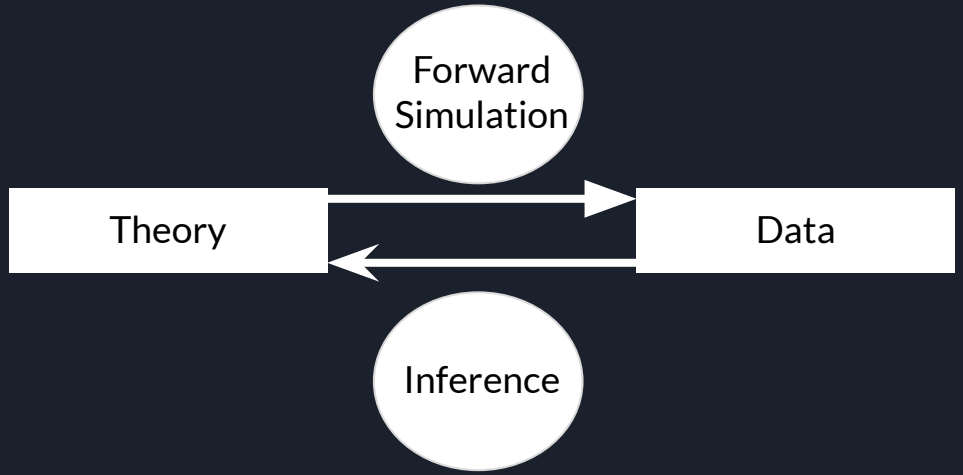
Two recent Reviews provide a very clear picture

- arxiv:2203.11110 is a very clear review of state of the art tools, their impressive results, their main issues and current “traditional” ideas to solve them
- arxiv:2203.07460 provides a complementary viewpoint from a ML perspective

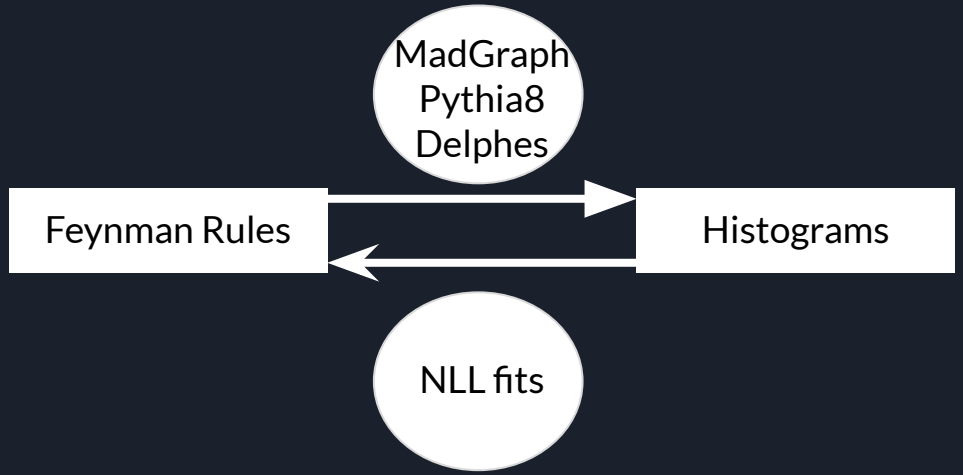
Ideally



Why we need Event generators



An example


MadGraph
Pythia8
Delphes
Feynman Rules
Histograms
NLL fits

Specific needs from event generators

- Physical simulations: we need the simulated events to follow as close as possible the modelling hypothesis (e.g. QFT predictions at a particle collider from Standard Model Lagrangian)
- Speed: We need simulations to be reasonably fast
- Size: Ideally they should be storable and shareable

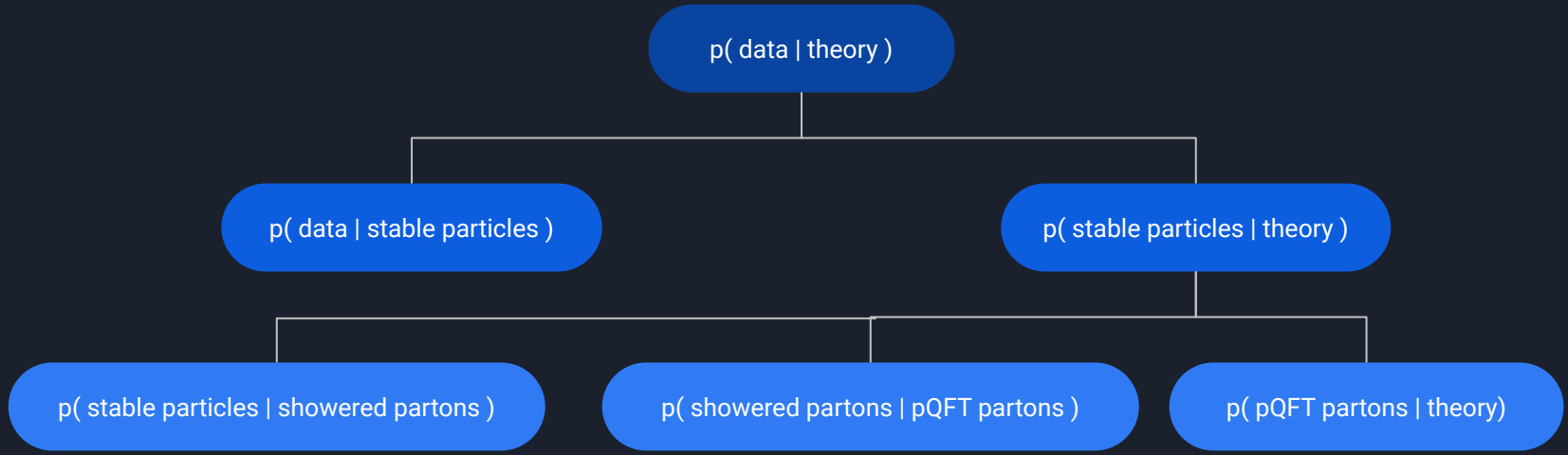
Focusing on collider experiments

Event generators need to consider many different physical phenomena, with different scales and different tools:

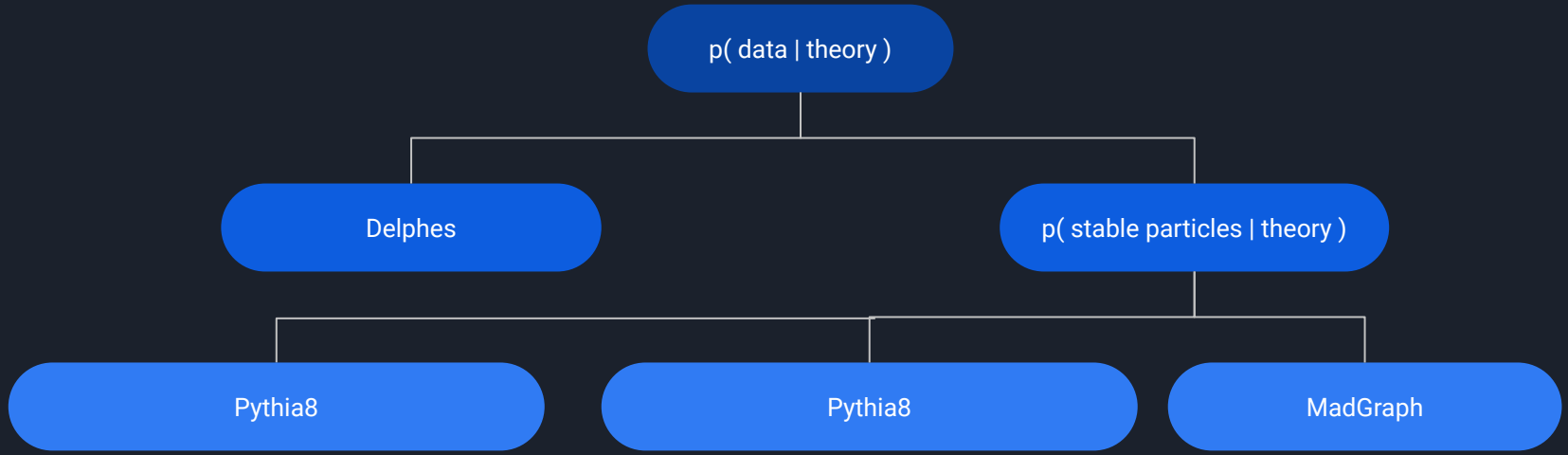
- Partons originating from colliding hadrons sampled through PDFs
- Hard scattering amplitudes calculation and phase space integration
- ISR/FSR
- Hadronization
- Final state interactions
- Underlying event effects
- Detector effects

Making use of factorization theorems, different dedicated softwares have achieved incredible sophistication but still face difficulties

An example of factorization theorems (a modelling assumption?)



An example of factorization theorems (a modelling assumption?)



Main issues:

High-dimensional parameter space that models empirically different non-perturbative effects plus assume exact factorization theorems:

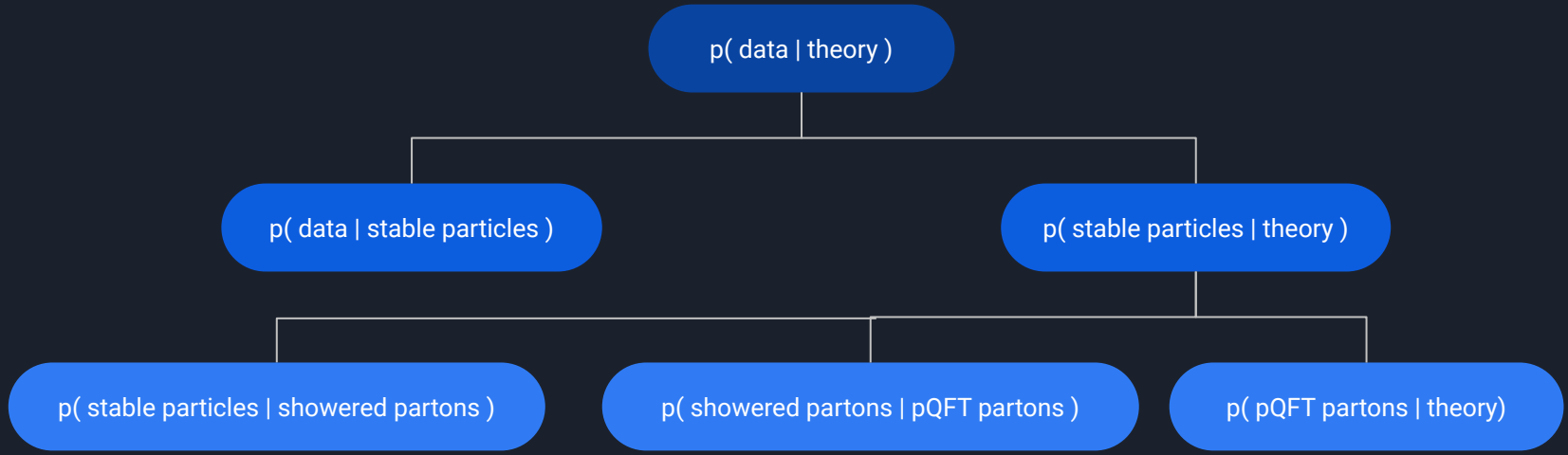
- Expensive tunes
- Expensive and difficult treatment of uncertainties
- Additional systematics due to modelling
- Computational bottlenecks
- Numerical instabilities
- Cross-cutting from factorization theorems' breakdowns
- Self-consistency of parameter tunes

Surrogate models

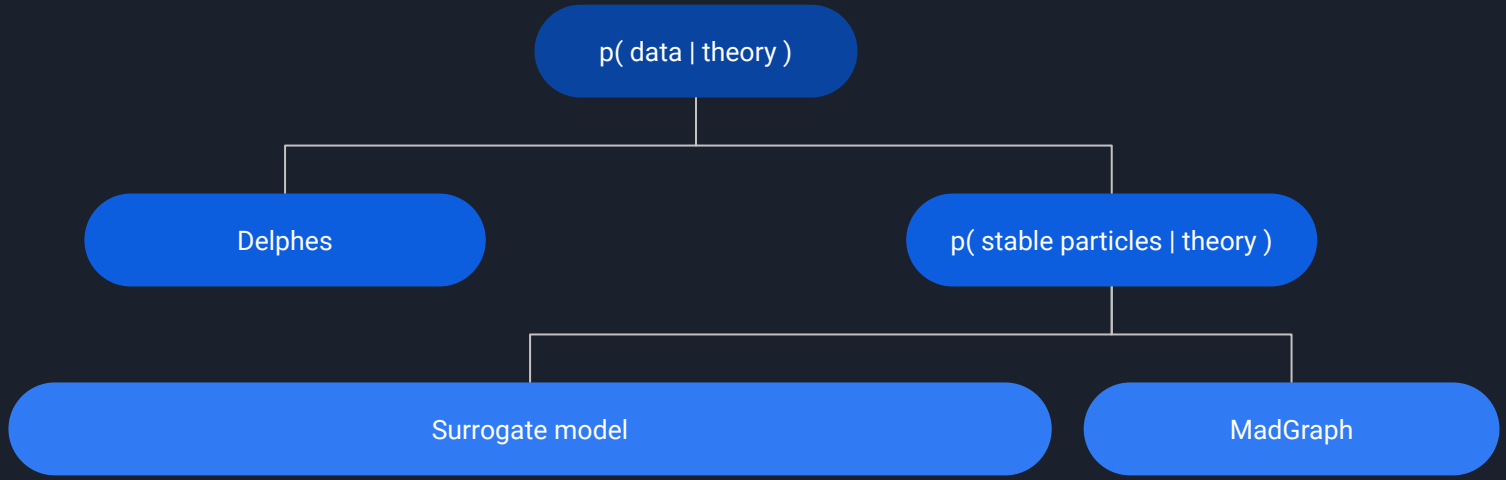
Empirical models are necessary for non-perturbative MC simulations (mainly, hadronization) but introduce systematic uncertainties and unphysical behavior

However, they also make uncertainty-evaluation and tunes more efficient.

An example of Surrogate models



An example of Surrogate models



Surrogate models

This is usually where Machine Learning can be really helpful as it can learn very precise surrogate models for different modules.

See for example (there are many, many others):

- MLHAD / HADML for Pythia8 / Herwig surrogate models
- CaloGAN, CaloFlow for Geant4 calorimeter surrogate models
- OTUS, DijetGAN for End-to-end surrogate models

Surrogate models for inference

Surrogate models can make for a much easier parameter inference and unfolding.

See for example:

- MLPF for particle reconstruction from calorimeter and trackers
- OmniFold and cINN for unfolding
- MadMiner and the Matrix Element Method for parameter inference

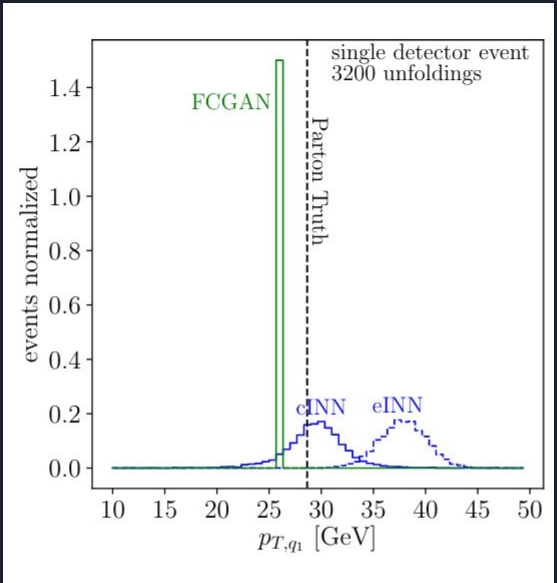
Bayesian techniques

They already permeate event generators and their surrogate models extensions: NNPDFs, Bayesian Neural Networks, cINN surrogate models

Bayesian inference allows for a better evaluation of uncertainties and for improved unfolding (as long as one can make probabilistic statements)

Intuitively, Bayesian always sounds good… Now it is becoming more possible. Differentiable programming is a key development. So are different approaches to inference like Black Box Variational Inference and simulation-based inference where the intractabilities are somewhat sidestepped.

Bayesian unfolding with cINN
(M. Bellagente et al, arxiv:2006.06685)



Bayesian techniques for modelling

There is a vast array of literature about Bayesian model building we can take advantage of. Already a lot of examples in HEP (e.g. GANs, VAEs, Shower Deconstruction, LDA, Topic models in general …)

Always need to keep in mind the same requirements as for event generators:

- Physically meaningful: harder to achieve than with event generators. Requires physical bias to be baked into the model.
- Speed + size: training should not be too hard nor require so much data as to render simulations preferable

Where could we go further?

Could we lessen the modularity assumption? Start from traditional MC approaches and treat cross-cuttings from a Bayesian perspective.

Treat our simulator as a non-parametric bayesian model? As a prior over an empirical model?

Start thinking more about marginal posteriors as swyft does for astro-cosmo. Likelihood-free inference.

What about Multilevel model emulators?

Could we implement model comparison / model combination Bayesian techniques to asses and combine different tunes and even different generators together and resolve inconsistencies?

Always one should decide how much to trust current simulation techniques and how to infuse them with more Bayesian methods

Thank you!