A novel workflow of generator tunings for LHC new physics searches using HPC

Jim Kowalkowski, Paolo Calafiura, Zachary Marshall, Juliane Mueller, Stephen Mrenna, Tom Peterka, Holger Schulz and **Xiangyang Ju**

Supported by SciDAC4 project: HEP data analytics on HPC

Physics Event Generator Computing Workshop
Nov. 27 2018, @CERN

Introduction

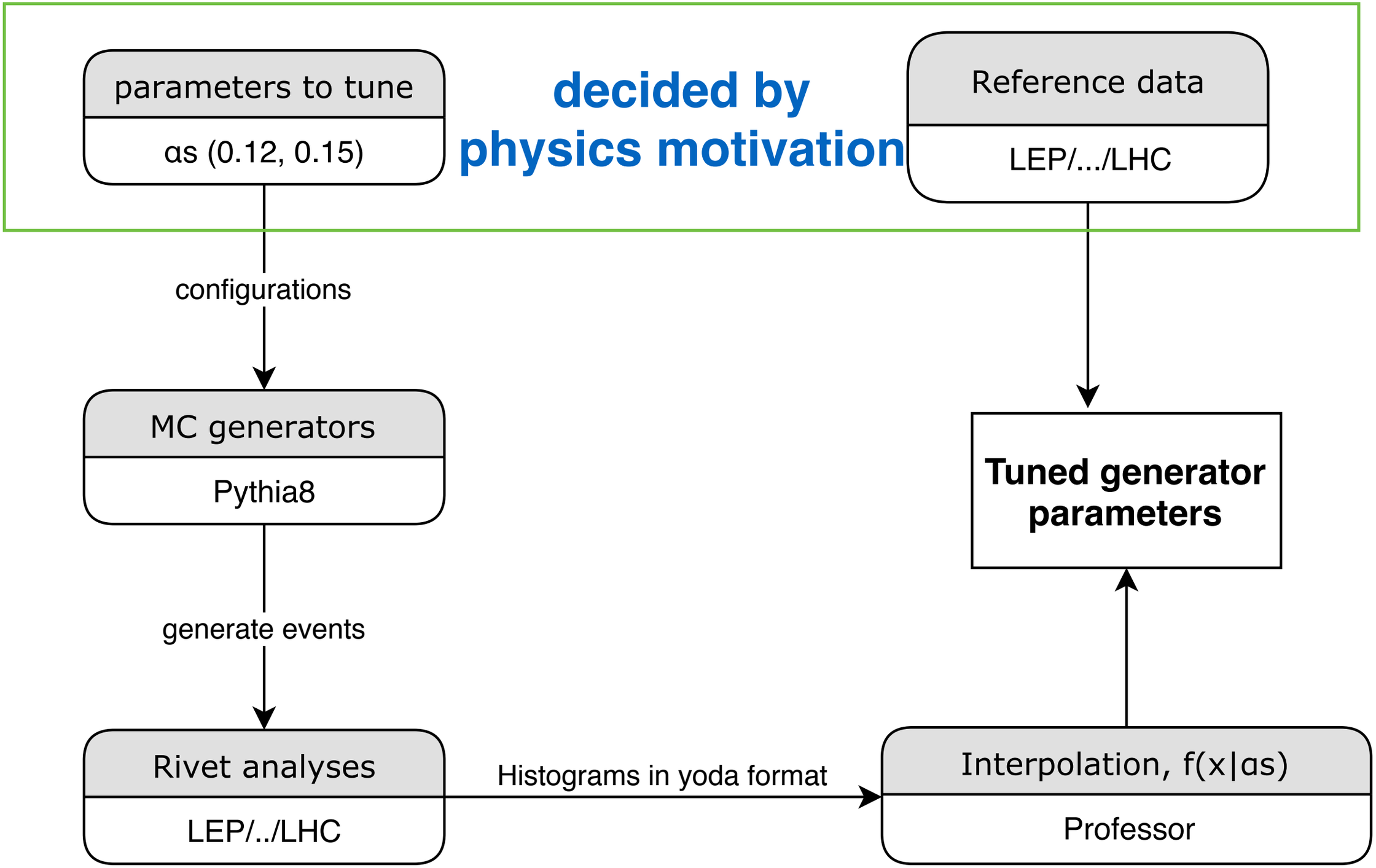
- Tuning of event generator models is needed to describe the structure of events not fixed by usual perturbative QCD calculations.
- Current workflow follows closely that in **Professor** developed by Andy, Holger and others in 2009, inspired by work at LEP in 1984.
  - **Can we improve the workflow?**

  - **Can we use the workflow to do something more?**

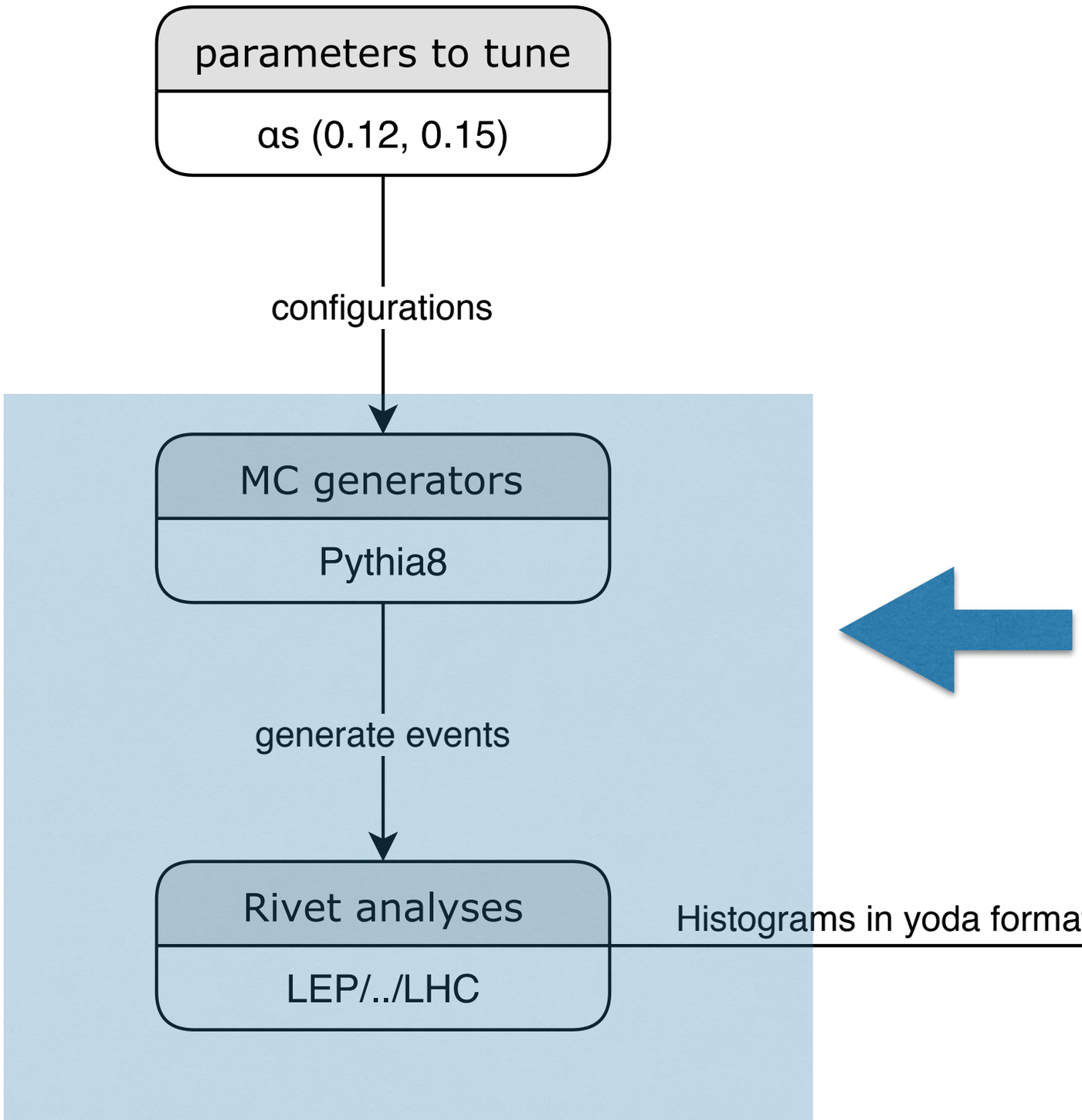
Introduction

- Tuning of event generator models is needed to describe the structure of events not fixed by usual perturbative QCD calculations.
- Current workflow follows closely that in **Professor** developed by Andy, Holger and others in 2009, inspired by work at LEP in 1984.
  - **Can we improve the workflow?**
    - use HPC to expand the number of parameters that can be tuned,
    - propagate uncertainties associated with MC events to tuning results,
    - automatically select the most relevant parameters to tune,
    - automatically adjust the importance of different observables.
  - **Can we use the workflow to do something more?**
    - tune parametric detector simulation,
    - estimate backgrounds for new physics searches at LHC,
    - tune a leading-order MC to high-order MC or Data.

Current workflow



Current workflow, challenges

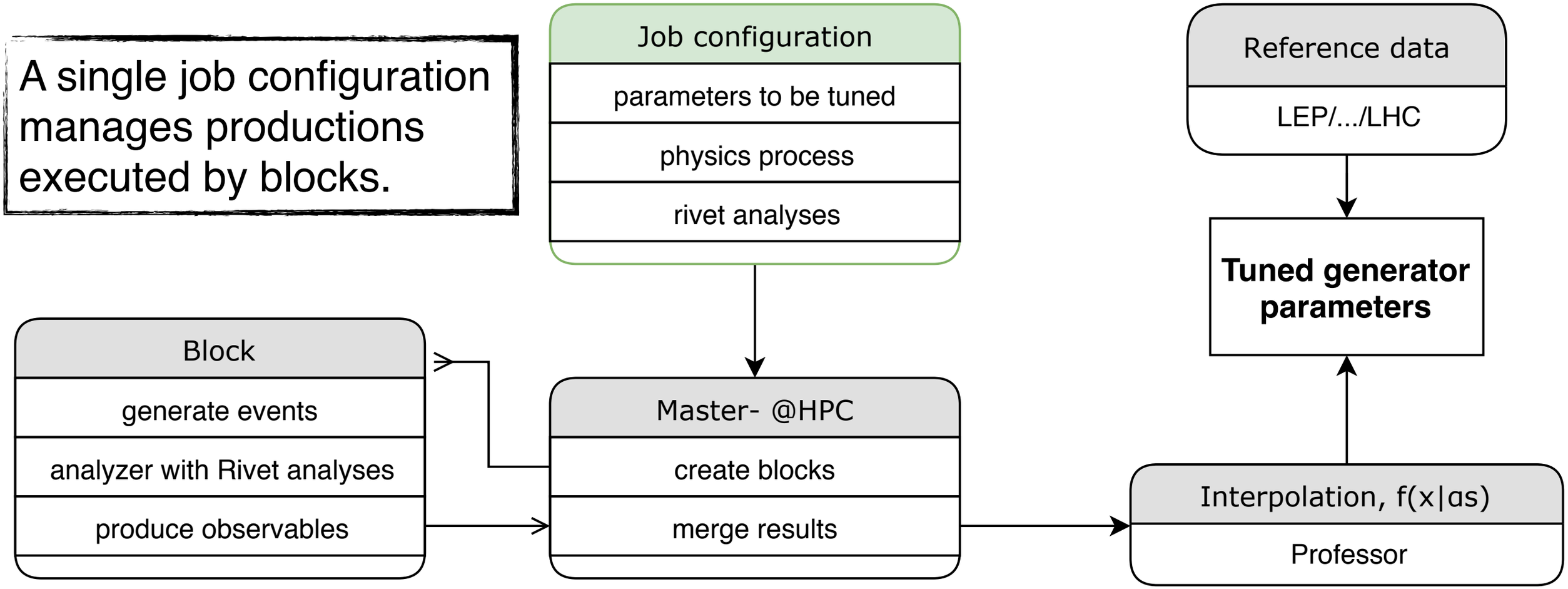


Takes most of CPU time. (numbers are taken from ATLAS A14 tune, a global tune, as an example.)

1. many parameters to tune [>=10]
2. many configurations [>=500]
3. each configuration needs large amount of MC events [>=1million]
4. many physics processes [>=4]
5. many Rivet analysis [>=11]

New workflow for HPC

A single job configuration manages productions executed by blocks.

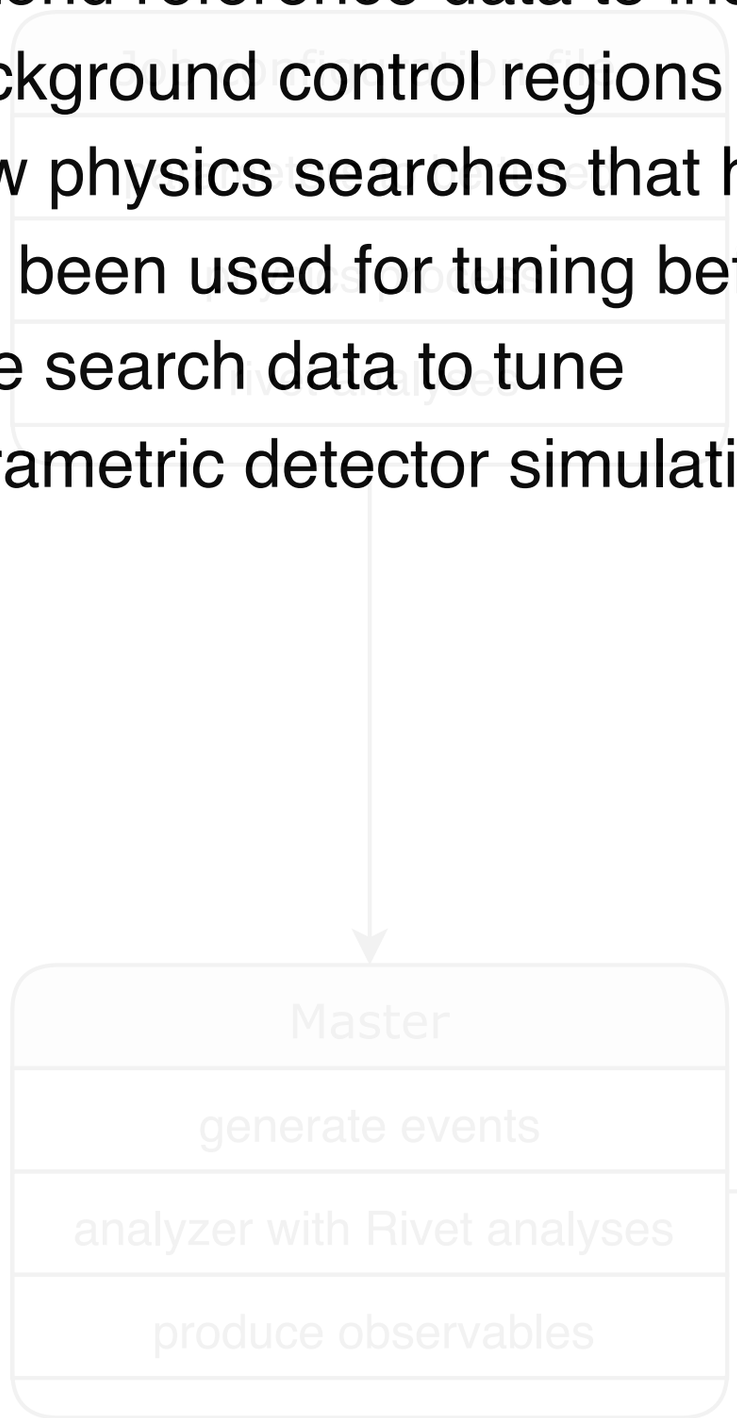


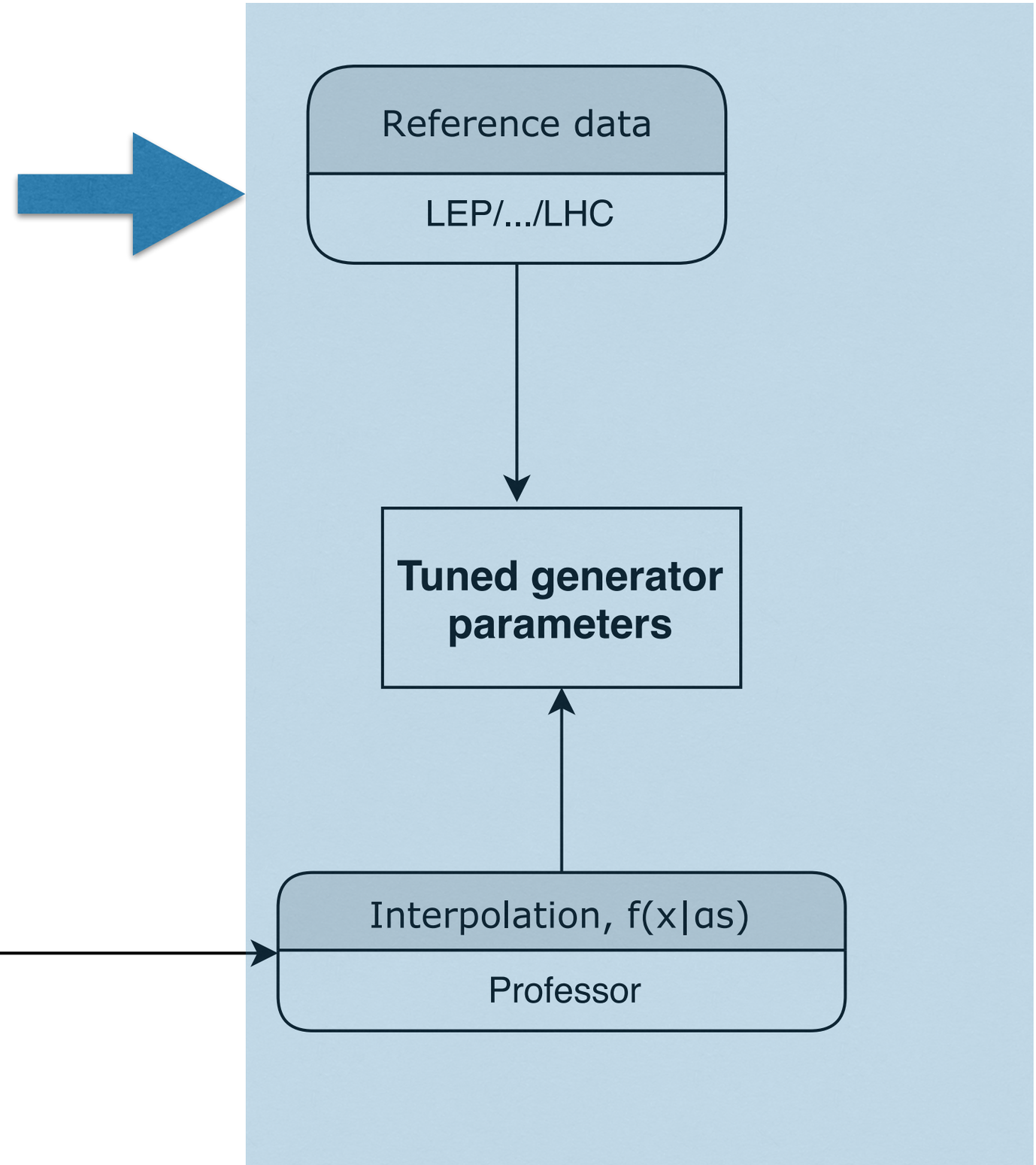
Advantages include:

1. Events processed by all Rivet analyses in a single run.
2. Less intermediate steps
3. Events generated in parallel using full computation power of HPC
4. Allowing automated tuning.

New workflow, data, Professor

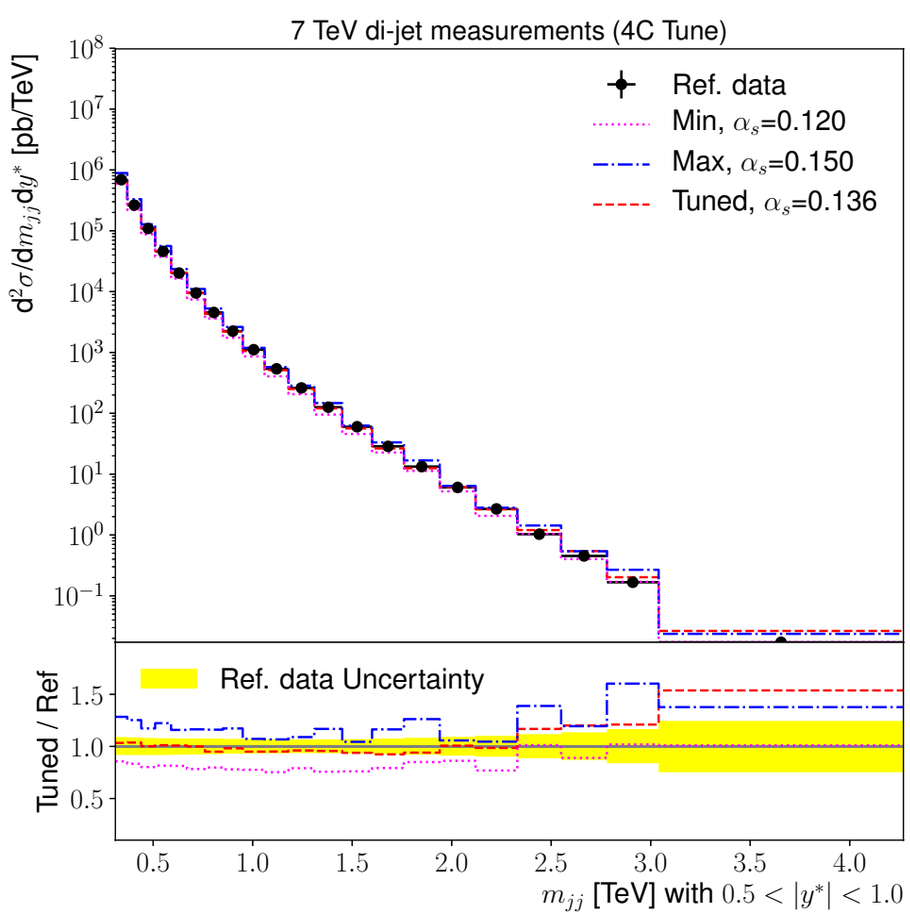
1. Extend reference data to include background control regions in new physics searches that has not been used for tuning before.
2. Use search data to tune parametric detector simulation.

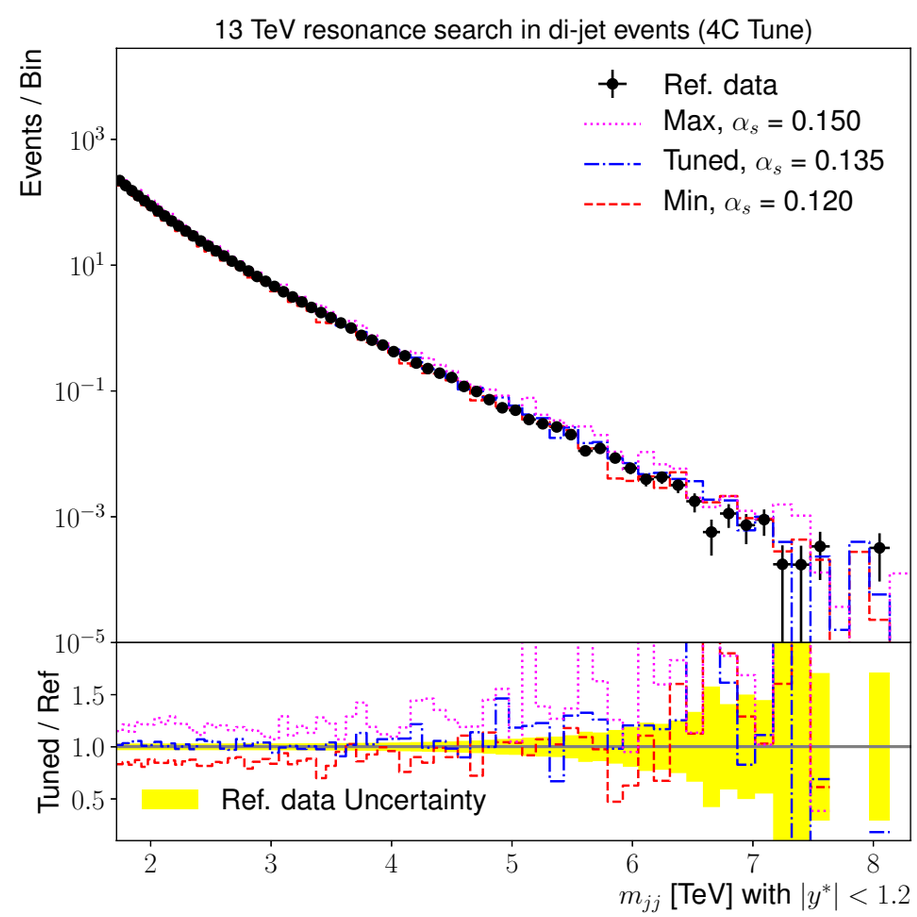




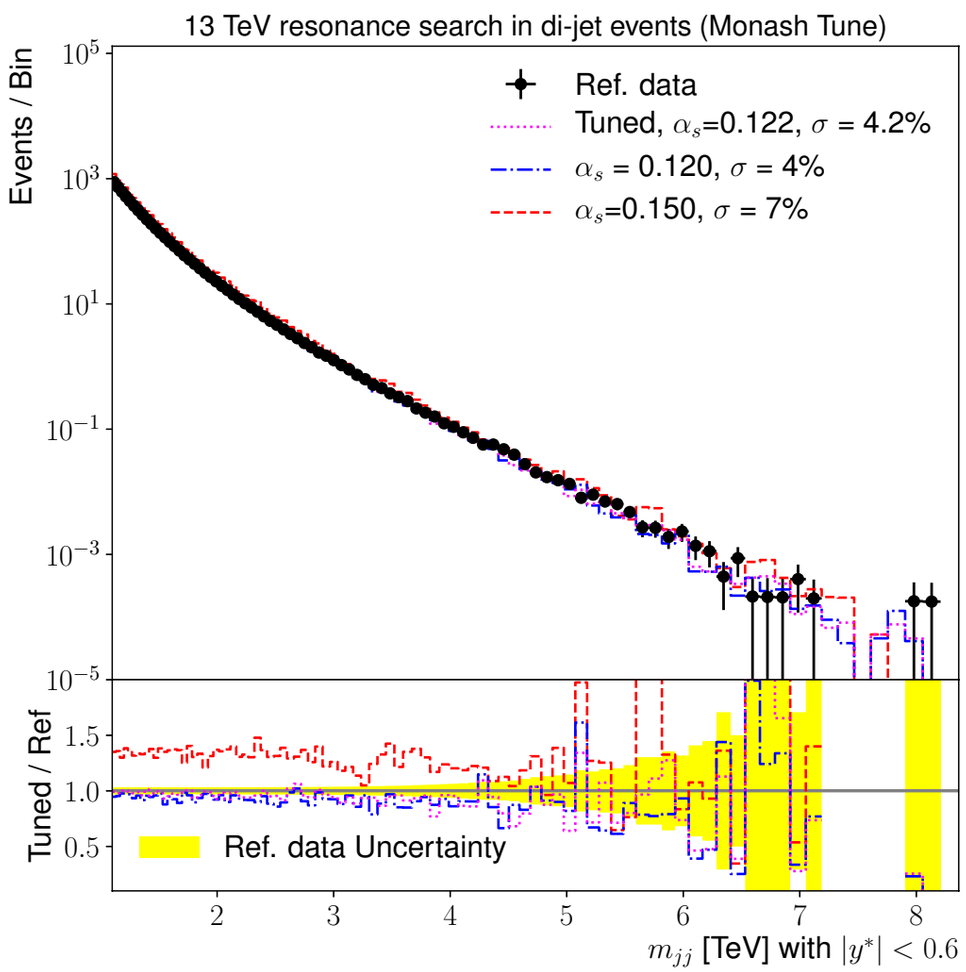
Generator tuning with detector data

- Generator tuning normally performed with ‘unfolded’ measurement data
  - Measurements often take years to complete and are limited in kinematic range
  - Search data available much faster in wider kinematic range
- **Use fast simulation to model detector effects, and tune to search data**
  - **First results compare well with measurements**
- Expands data that can be used and kinematic range
  - Tune event generators in phase space regions most interesting to LHC searches.





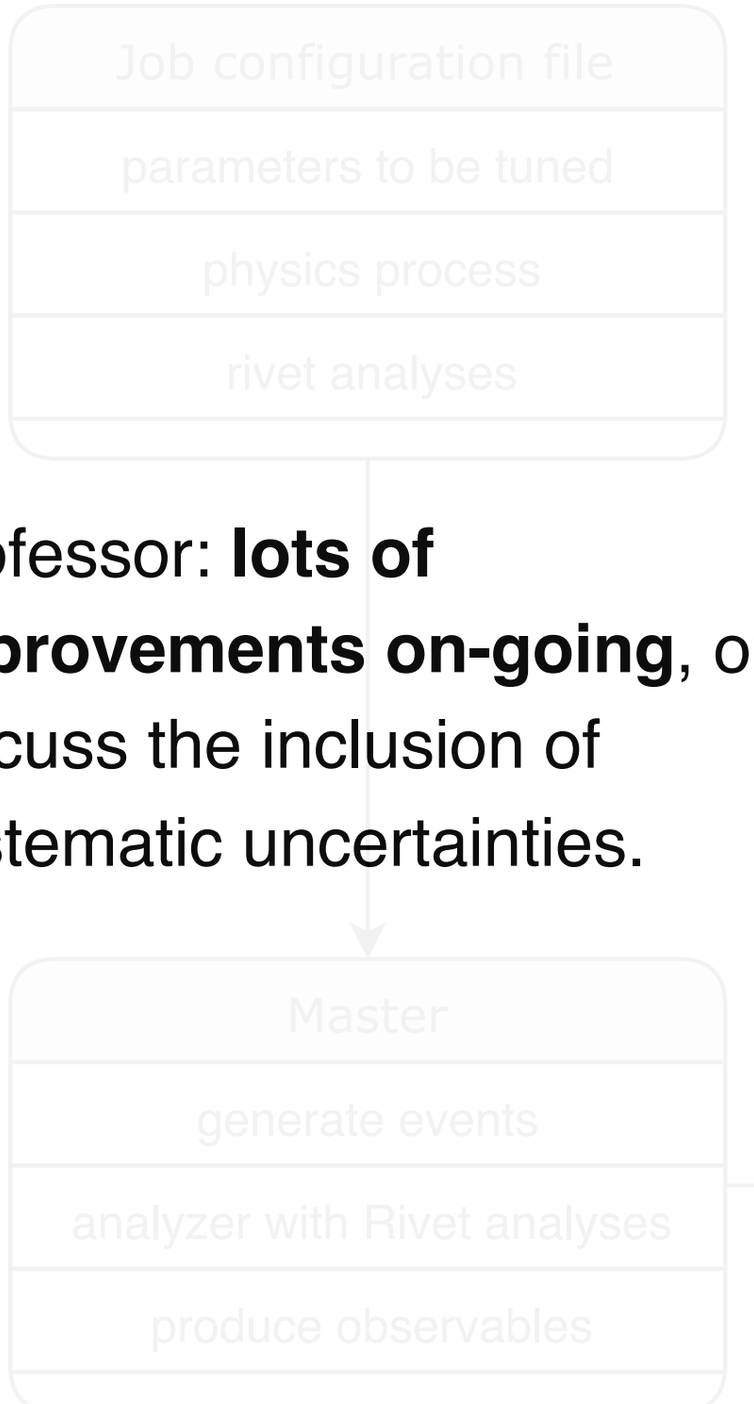
Simulation tuning



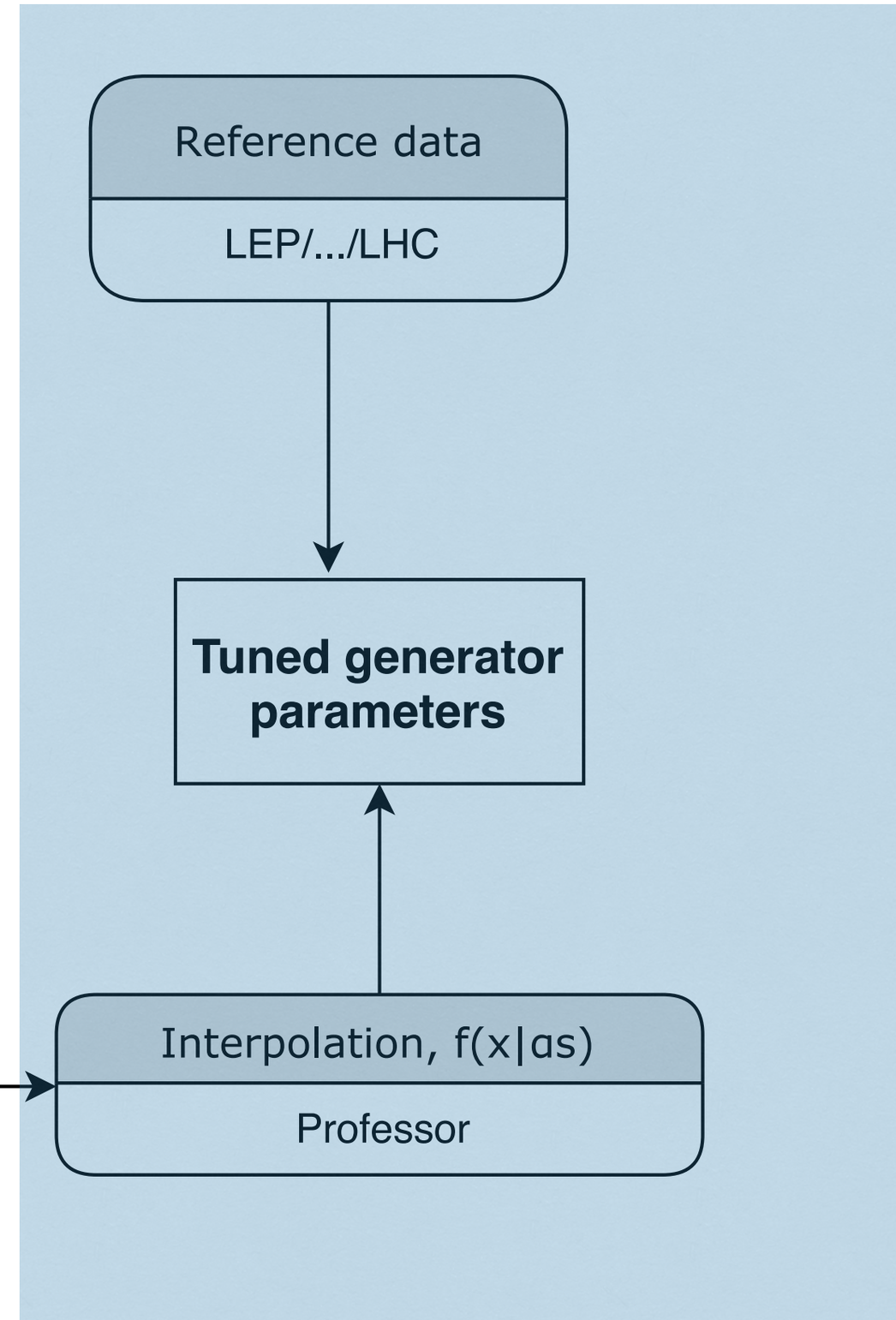
- The mechanism used for tuning event generators can be applied to tune parametric simulation.
  - **Proof of principle with Rivet fast simulation, using the same tools and workflows as for event generation tuning**.
- Parameters normally taken from papers published by the experiments
  - not always applicable to search regions, where these fast simulation are most used
- **Can be extended to provide an LHC search-data based 'tune' of efficiencies and resolutions**.

Above two results are only an introductory exercise to show **what can be done in principle for a more difficult and useful case.**

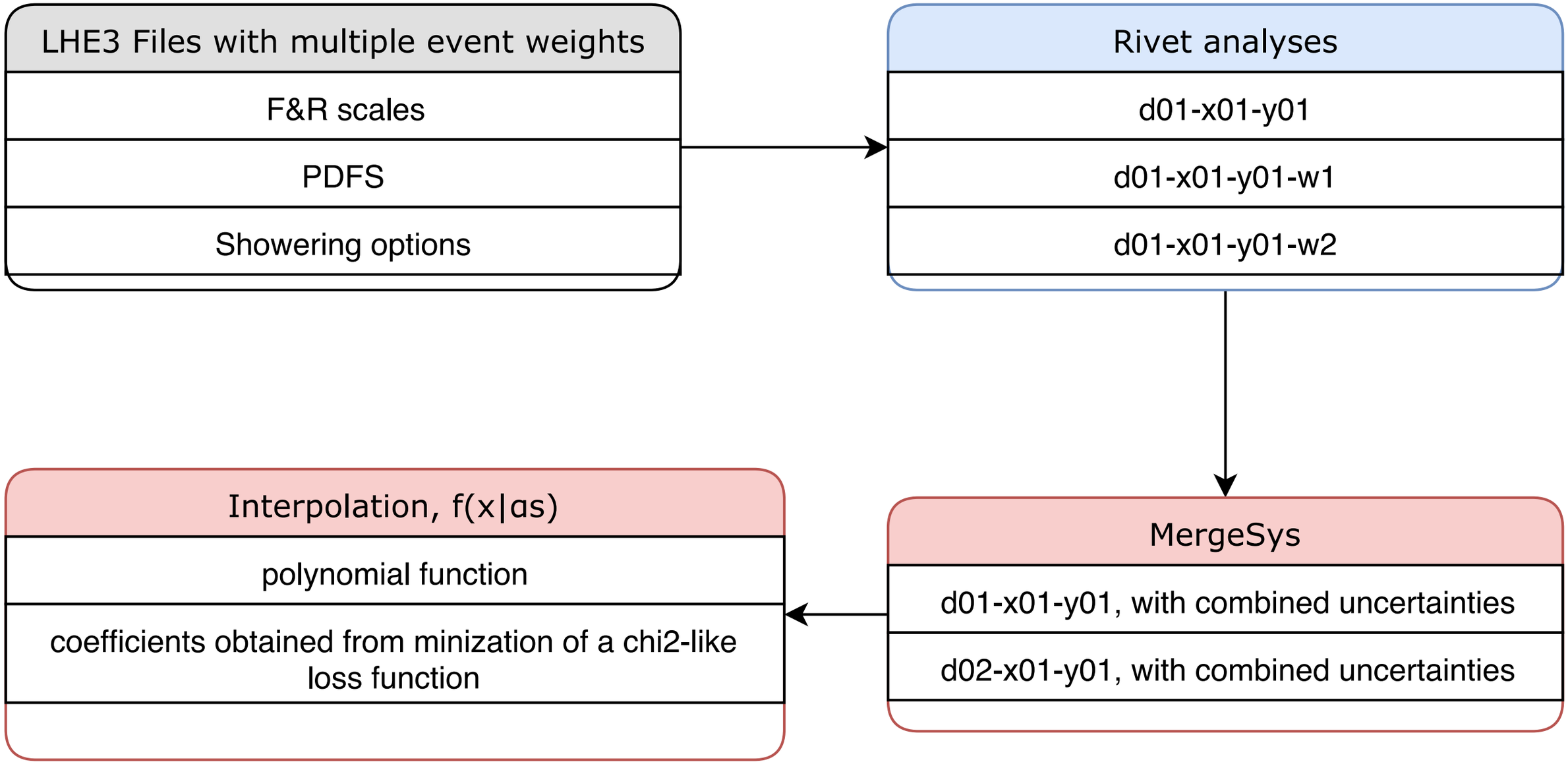
New workflow, data, Professor



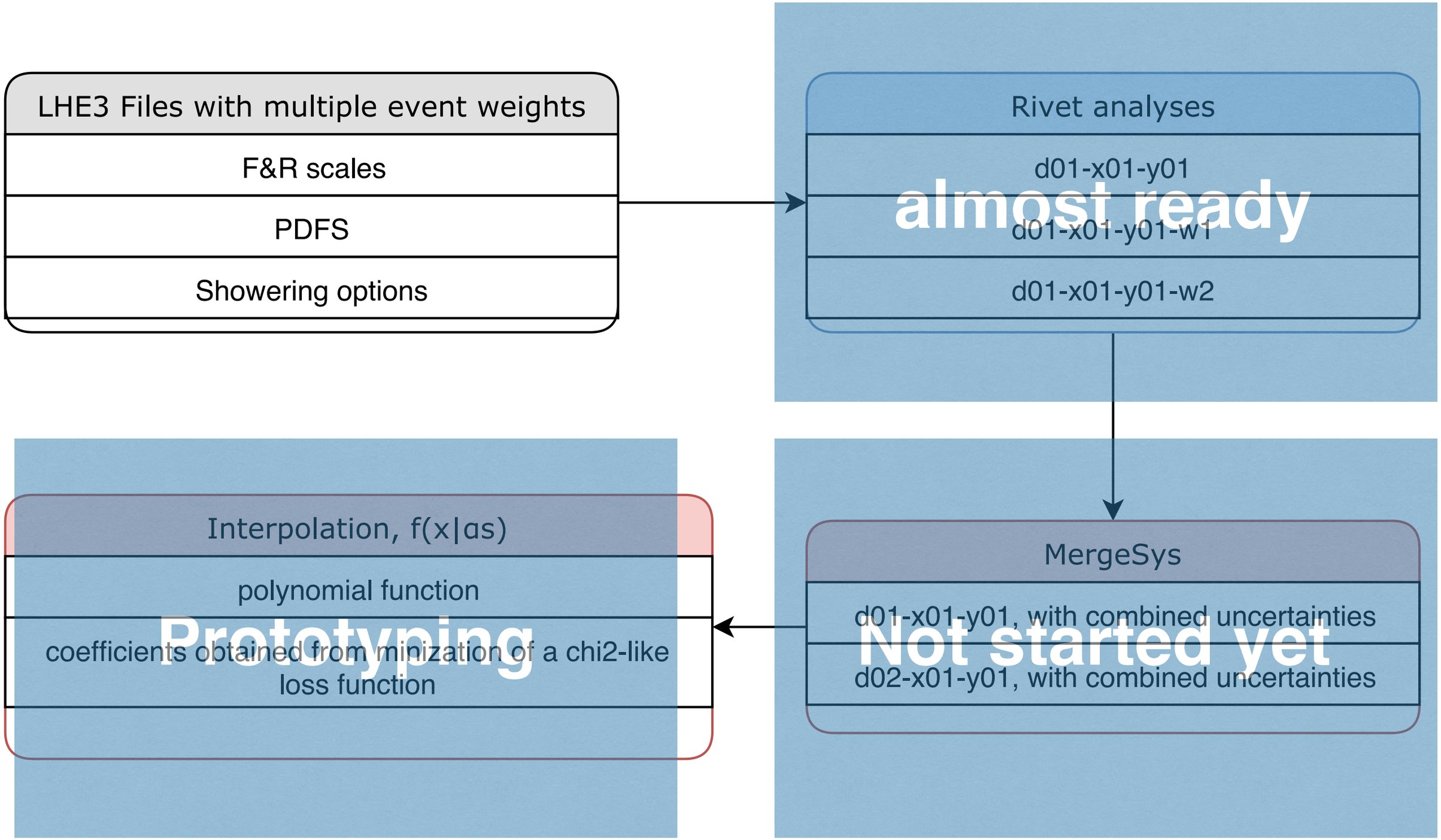
1. Professor: **lots of improvements on-going**, only discuss the inclusion of systematic uncertainties.



Tuning with systematic uncertainties



Tuning with systematic uncertainties



Surrogate function

Surrogate function **f(x|α)**, **computationally cheap function replacing expensive MC generation**, is to model the dependence of observables on input parameters

Currently a polynomial function is used:

$$f(\boldsymbol{p}_0 + \delta\boldsymbol{p}, x) = a_0^{(0)} + \sum_{i=1}^{n} a_i^{(1)}(x)\delta p_i + \sum_{i=1}^{n}\sum_{j=i}^{n} a_{ij}^{(2)}(x)\delta p_i \delta p_j$$
$$\approx M(\boldsymbol{p}_0 + \delta\boldsymbol{p}, x)$$

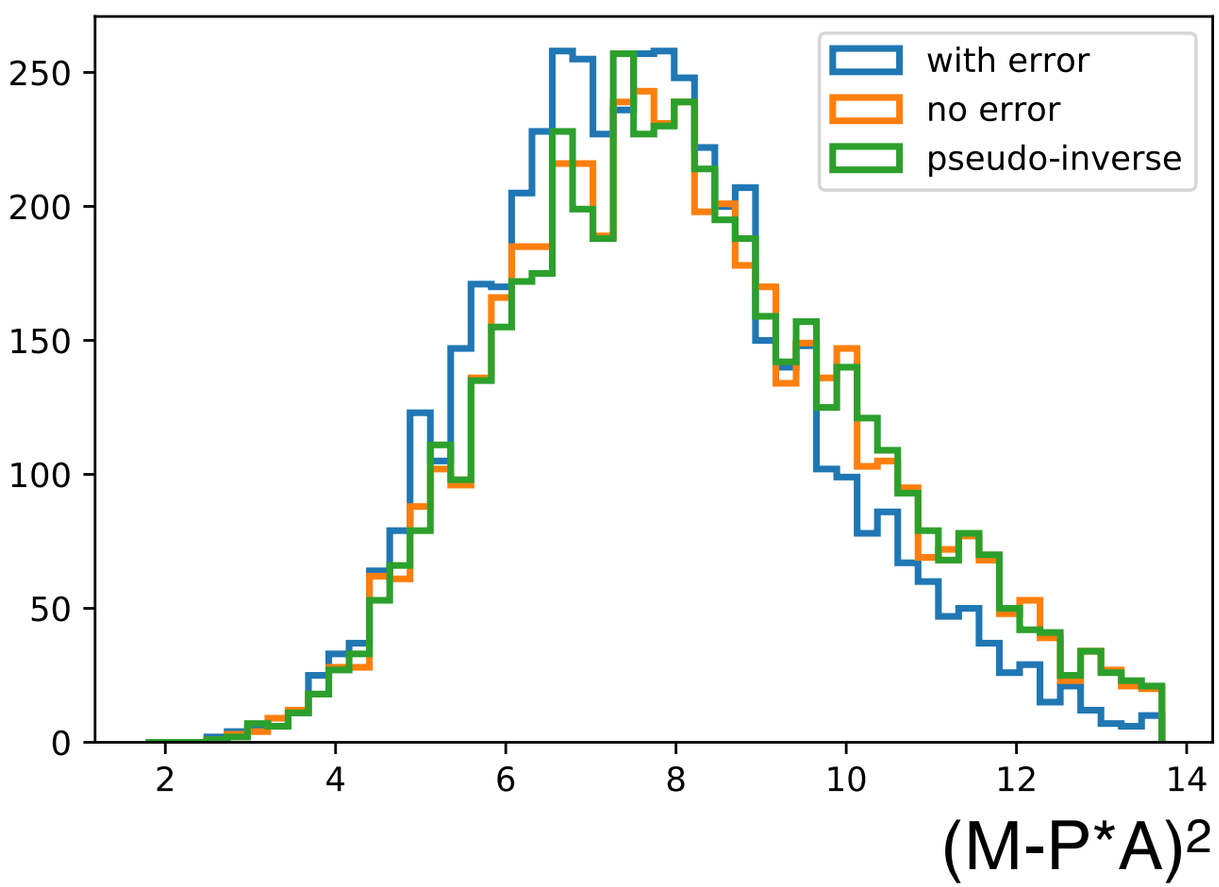
$$\downarrow$$

Effectively solve linear equations for A. $\boldsymbol{P} \cdot \boldsymbol{A} = \boldsymbol{M}$

Without errors associated with M, it can be solved by **pseudo-inverse** or **minimizing the loss function**:	$\mathcal{L} = \|\|\boldsymbol{M} - \boldsymbol{P} \cdot \boldsymbol{A}\|\|_2$
With errors associated with M, it can be solved by **minimizing the loss function**:	$\mathcal{L} = (\boldsymbol{M} - \boldsymbol{P} \cdot \boldsymbol{A})^T \Sigma^{-1} (\boldsymbol{M} - \boldsymbol{P} \cdot \boldsymbol{A})$

A dummy example

1. Assume we want to tune two parameters [x1, x2], use 2nd order polynomial function to model the dependance.
    1. 6 coefficients in the polynomial function to be determined.
2. Generate pseudo-data for each [x1, x2] for 30 times, called 30 scans. Each pseudo-data is artificially assigned a 5% uncertainty.
3. Use three methods to estimate the 6 coefficients.
4. To judge which one is better, generate new pseudo-data and calculate the difference squared.



f(x|α) obtained by the minimization of the loss function including MC uncertainties results in **smaller mean value** and **variance.**

Summary and outlook

- We showed proof-of-principle on two things tuning workflow can potentially do:
  - Generator tuning with detector data,
  - Parametric detector simulation tuning.
- Following improvements have been brought to the workflow:
  - Generating MC events in parallel in HPC,
  - Inclusion of systematic uncertainties.
- Extending and simplifying automatic tuning will allow many important tests:
  - tune universality
    - LO/multi-leg/NLO tune universality
    - Underlying events/Multiple Parton Interaction universality
  - difference between hadronization models.

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of HEP, Scientific Discovery through Advanced Computing (SciDAC) program.