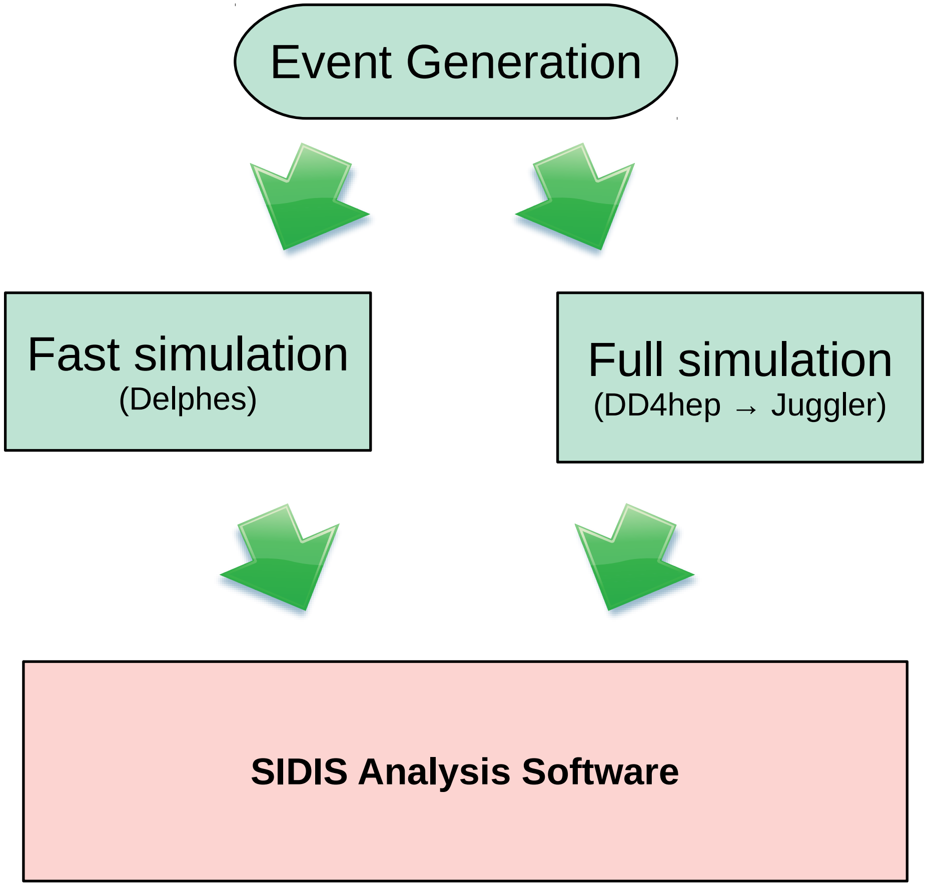
SIDIS Analysis Software for EIC

- ATHENA Software
- SIDIS Analysis Software
- Support for the Future

Christopher Dilks
27 April 2022

Duke
UNIVERSITY

Big Picture

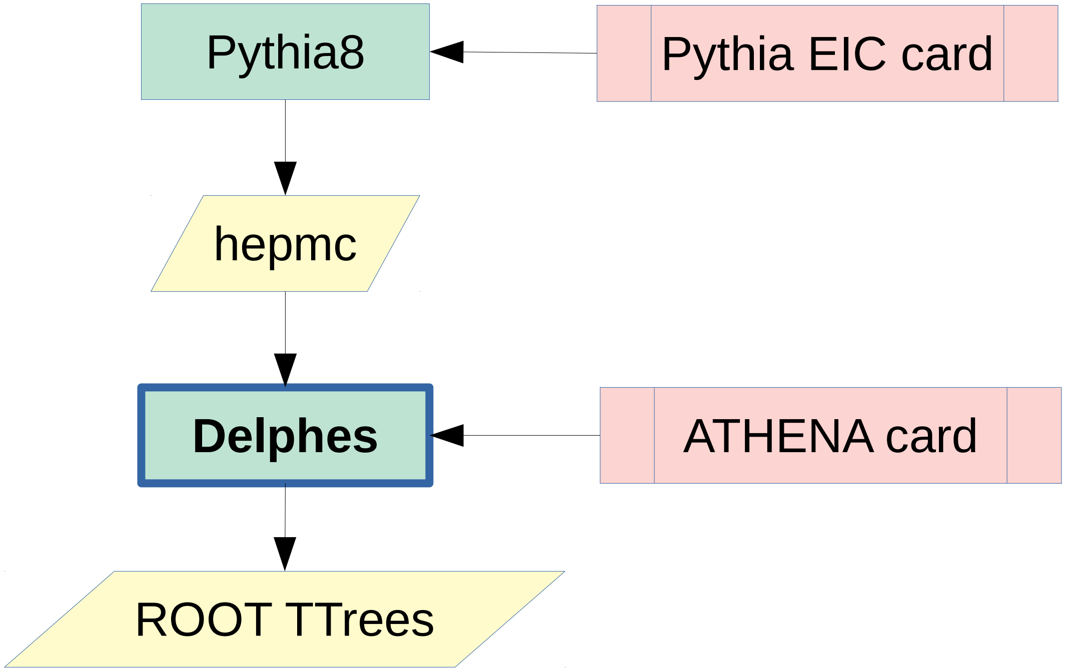

Event Generation
Fast simulation
(Delphes)
Full simulation
(DD4hep → Juggler)
SIDIS Analysis Software

Delphes Fast Simulation

https://cp3.irmp.ucl.ac.be/projects/delphes

DELPHES
fast simulation

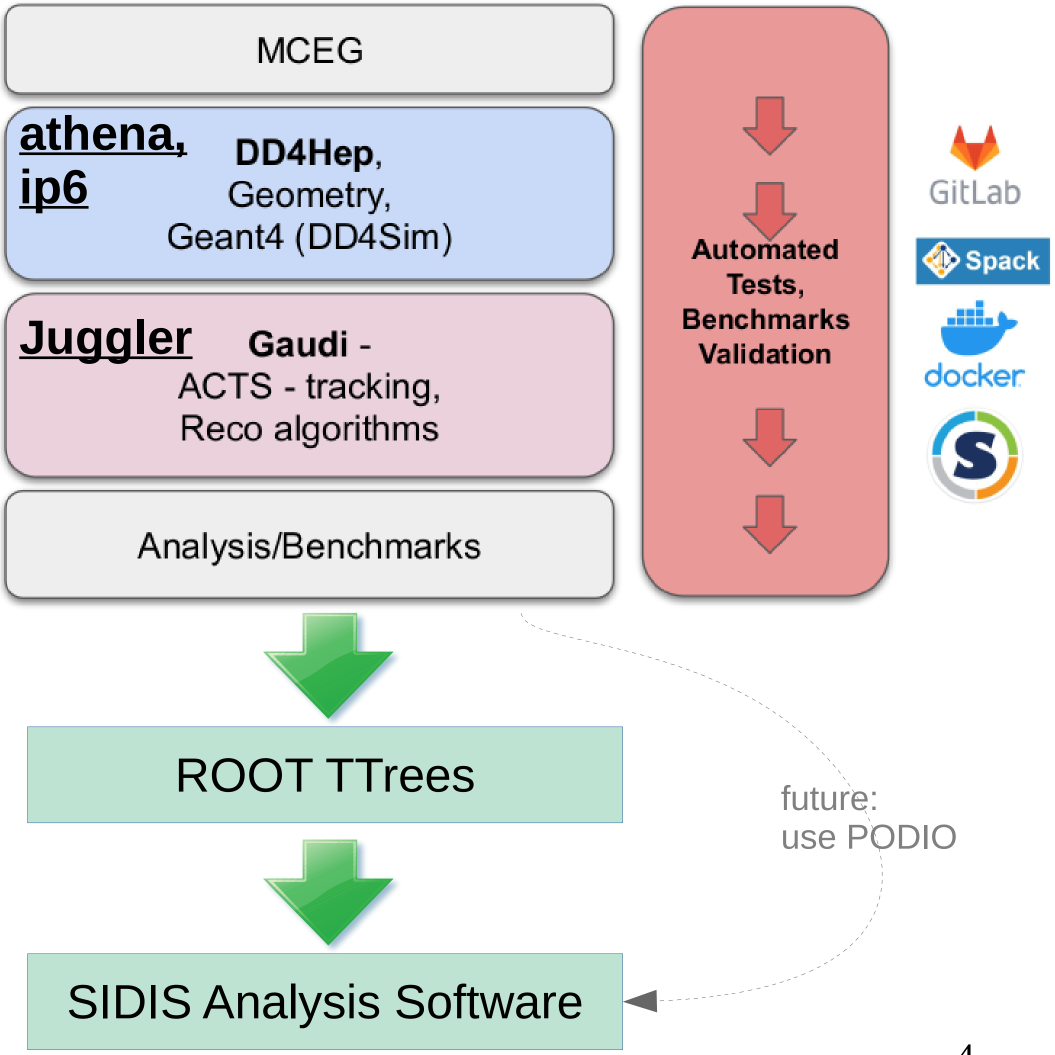
ATHENA configuration (card):
https://github.com/eic/delphes_EIC/tree/master



ATHENA Full Simulation Software Stack

- Detailed detector geometry description in **DD4HEP**, which steers the Geant4 simulations
- Reconstruction framework (**JUGGLER**) built on top of **GAUDI**, leveraging **ACTS** for tracking and **Tensorflow** for AI.
- Modular components communicate through a robust, flat data model (**EICD**, implemented using **PODIO**).
- Leverage dedicated GitLab server (**eicweb**) with CI backend for reproducible container builds (using **Spack**), and automated tests and benchmarks.

https://eicweb.phy.anl.gov/EIC

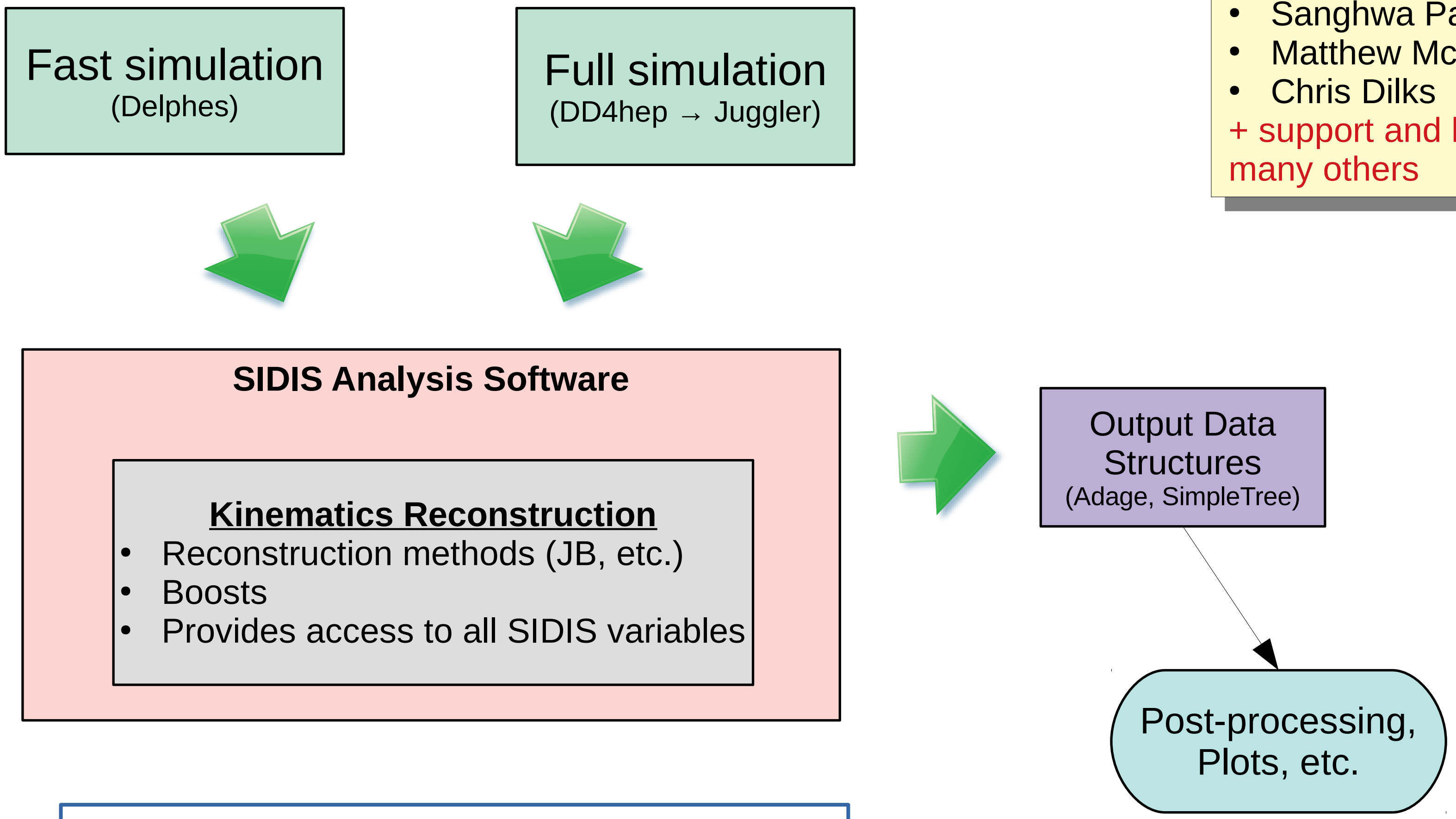


Slide adapted from Sylvester Joosten
https://indico.bnl.gov/event/14719/contributions/59799/attachments/39665/65790/2022-02-03%20Bi-weekly%20Collaboration%20Meeting.pdf

ATHENA SIDIS Analysis Software

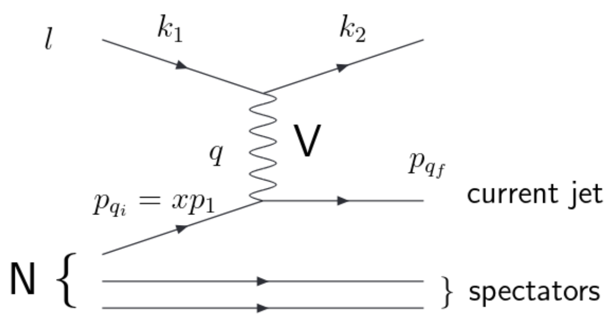
Contributors
- Duane Byer
- Connor Pecar
- Sanghwa Park
- Matthew McEneaney
- Chris Dilks

+ support and help from many others



https://github.com/c-dilks/largex-eic

Kinematics Reconstruction Methods



- SIDIS kinematics depends on what is used to reconstruct quantities such as x and Q^2
  - Scattered electron
  - Hadrons
  - Some mixture

i)	*Leptonic variables*	$q \equiv q_l = k_2 - k_1, \quad y_l = p_1.(k_1 - k_2)/p_1.k_1$
ii)	*Hadronic variables* [81]	$q \equiv q_h = p_2 - p_1, \quad y_l = p_1.(p_2 - p_1)/p_1.k_1$
iii)	*Jacquet-Blondel variables* [82]	$Q^2_{JB} = (\vec{p}_{2,\perp})^2/(1 - y_{JB}), \quad y_{JB} = \Sigma/(2E(k_1))$ $\Sigma = \sum_h (E_h - p_{h,z})$
iv)	*Mixed variables* [81]	$q = q_l, y_m = y_{JB}$
v)	*Double angle method* [83]	$Q^2_{DA} = \frac{4E(k_2)^2 \cos^2(\theta(k_2)/2)}{\sin^2(\theta(k_2)/2) + \sin(\theta(k_2)/2)\cos(\theta(k_2)/2)\tan(\theta(p_2)/2)},$ $y_{DA} = 1 - \frac{\sin(\theta(k_2)/2)}{\sin(\theta(k_2)/2) + \cos(\theta(k_2)/2)\tan(\theta(p_2)/2)},$
vi)	*θy method* [84]	$Q^2_{\theta y} = 4E(k_2)^2(1 - y_{JB})\frac{1 + \cos(\theta(k_2))}{1 - \cos(\theta(k_2))}, \quad y_{\theta y} = y_{JB}$
vii)	*Σ method* [85]	$Q^2_\Sigma = \frac{(\vec{k}_{2,\perp})^2}{1 - y_\Sigma}, \quad y_\Sigma = \frac{\Sigma}{\Sigma + E(k_2)[1 - \cos(\theta(k_2))]}$
viii)	*$e\Sigma$ method* [85]	$Q^2_{e\Sigma} = Q^2_l, \quad y_{e\Sigma} = \frac{Q^2_l}{s x_\Sigma}$

Prog.Part.Nucl.Phys. 69 (2013) 28-84, 1208.6087 [hep-ph]

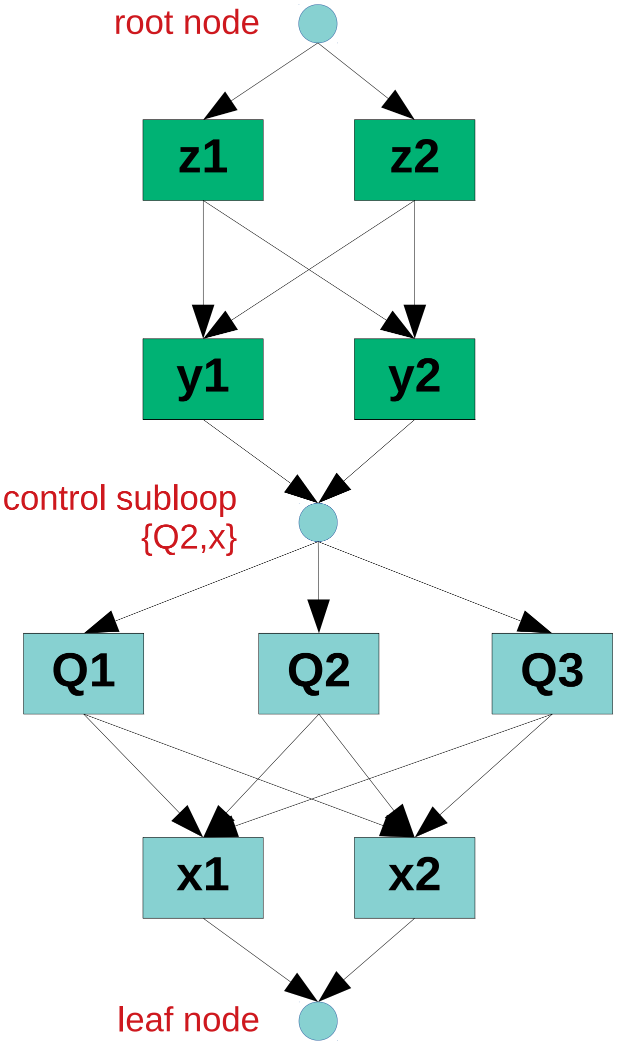
Data Structures

<u>Adage</u> – Analysis in a Directed Acyclic Graph Environment

- Graph data structure that stores:
  - <u>Data</u> in arbitrary multi-dimensional bins and cuts
    - 1 multi-dimensional bin == 1 full graph path
    - Anything can be stored in each bin; currently we store a large set of histograms
  - <u>Algorithms</u>, executable during graph traversal
    - No nested *for* loops: algorithms can be executed on every bin or any subset of bins
    - Allows for “binning agnostic” code
- Prototype developed within Largex-eic

<u>In practice:</u>
1) Define your bins
2) Define your control nodes (algorithms)
3) Run

4D Binning in (z,y,Q^2,x)



Data Structures

- **<u>Simple Tree</u>** – flat TTree, useful for quick tests etc.
  - Reconstructed SIDIS variables
  - Straightforward to connect to other analysis libraries
    - Asymmetry projections
    - Brufit (extension of Roofit)

- ***<u>Support for User Data Structures and Algorithms</u>***
  - Existing data structures may not suit our future needs
  - Implement your own ideas:
    - We could add "plugin" support, where the plugin would need:
      - A data structure class
      - Class methods Prepare(), Action(), Finish() = before all events, for each event, after all events
        - Similar to Juggler's initialize(), execute(), finalize()
        - Similar to Fun4all's Init(), process_event(), End()
  - Or add your own data structure to Adage, for multi-dimensional binning support

Example coverage plot: η vs. p in (x,Q^2) bins, with PID limits

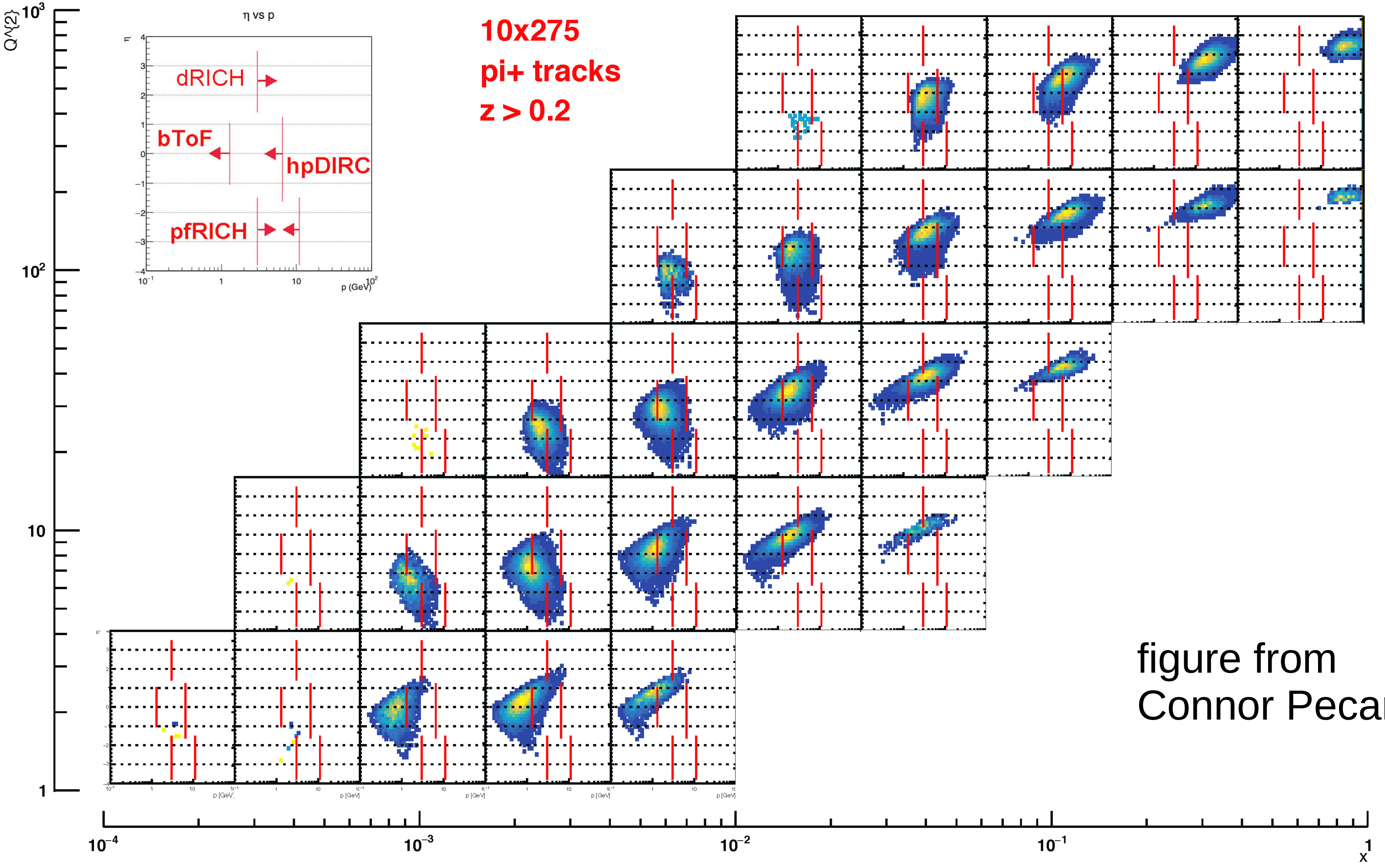


figure from
Connor Pecar

Example benchmark plot: pion z_{rec}–z_{gen}, from fast and full simulations, in (x,Q^2) bins

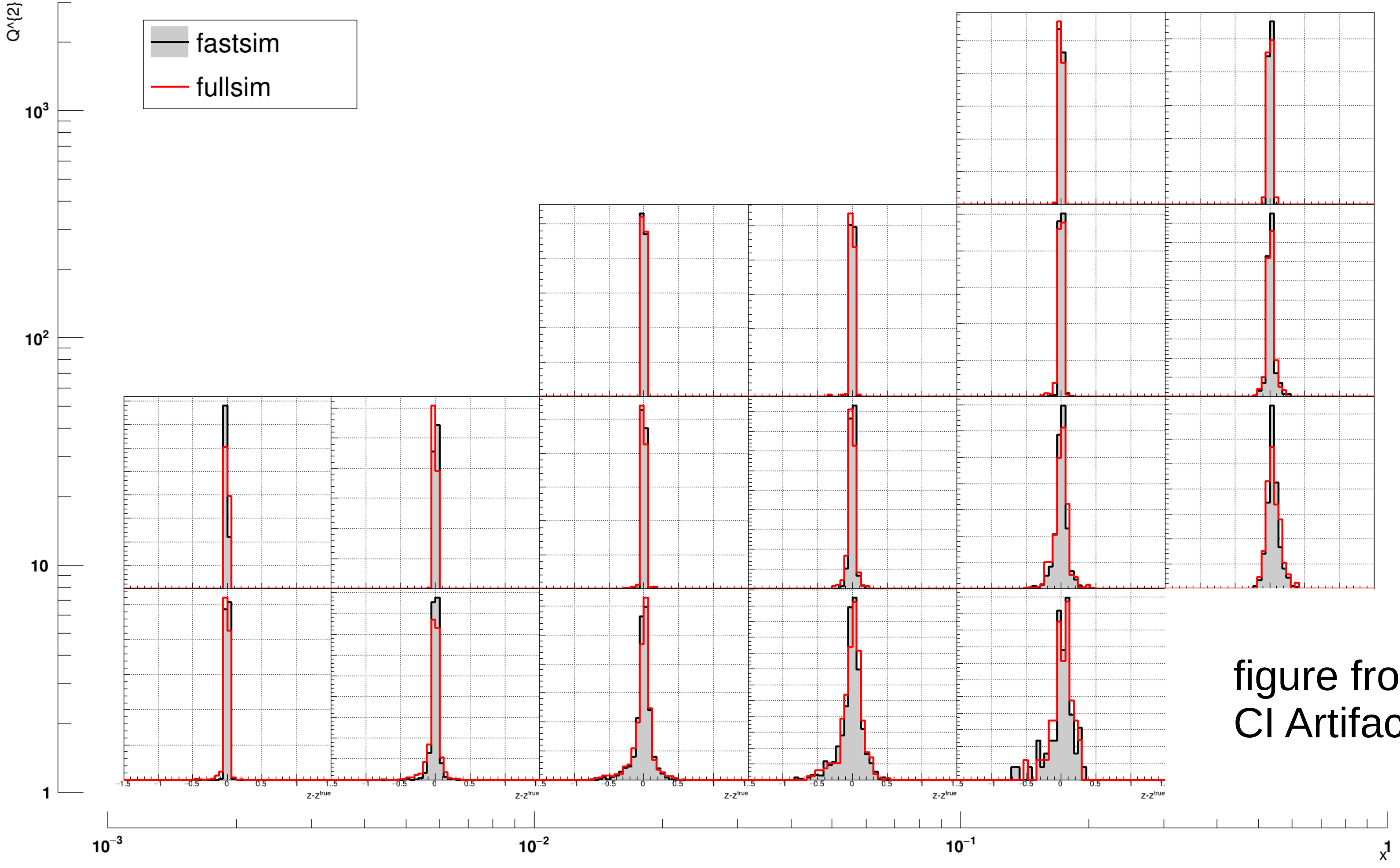


figure from
CI Artifacts

Example benchmark plot: y, from fast and full simulations, in pion (p,η) bins

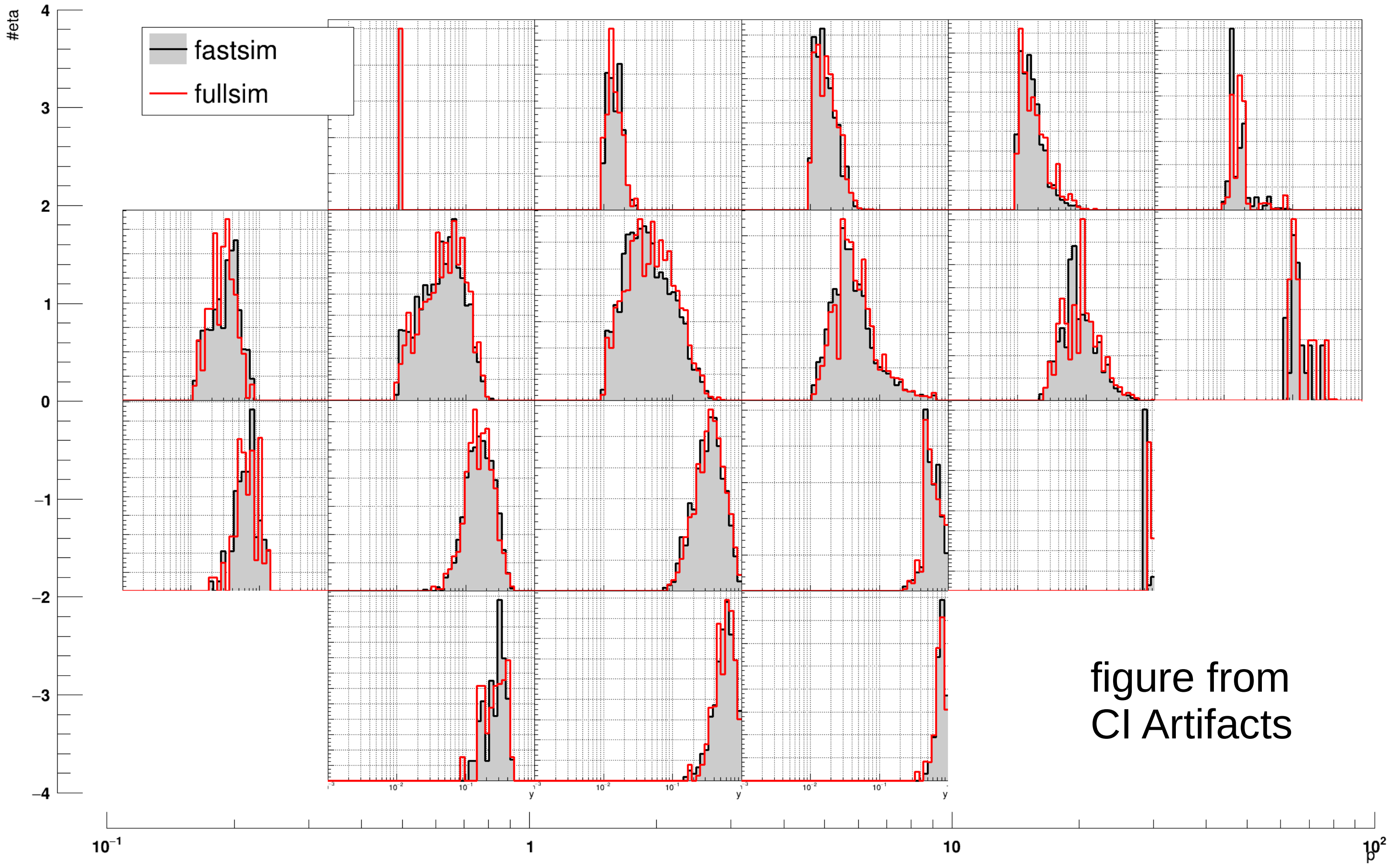


figure from
CI Artifacts

Software Design Principles adopted from ATHENA Software Group

- **Modularity**
  - One “task” = one “module”
  - Modules are mutually “orthogonal”
  - SIDIS SW itself is a module, reading output from fast and full simulations
  - Adaptable to upstream data structure changes → Analysis sub-classes
  - Adaptable to downstream needs → Edit existing or add new data structures

- **Continuous Integration (CI)**
  - Support development / testing
  - Automate generation of benchmark plots
  - Track evolution of any plot as development proceeds

- **Containerization**
  - Singularity / Docker image available, including dependencies such as Delphes and ROOT
  - Entry point for new contributors
  - Support CI

- **Version control (Git)**
  - Trunk-based development → pull requests and code reviews

Support for the Future

- **Upstream Integration:** migrate to EICweb (gitlab)

Contributions are welcome!

1) **Connect to upstream CI pipelines**
   - Example Scenario:
     - A change in detector design is being considered
     - Proposed change triggers detector CI pipelines and benchmarks
     - SIDIS analysis SW pipeline could also be triggered, providing immediate feedback of the effect of the proposed design change
2) **Improve Modularization and Integration**
   - Adage should be a separate module
   - Use PODIO
   - Kinematics calculations could be moved upstream (e.g., to a Juggler algorithm)
3) **Generalization**
   - We don’t have to limit ourselves to SIDIS
   - Already we have (some) support for jets
   - Support broader needs of the collaboration
   - Name change, since “largex-eic” is historical; our scope is much broader