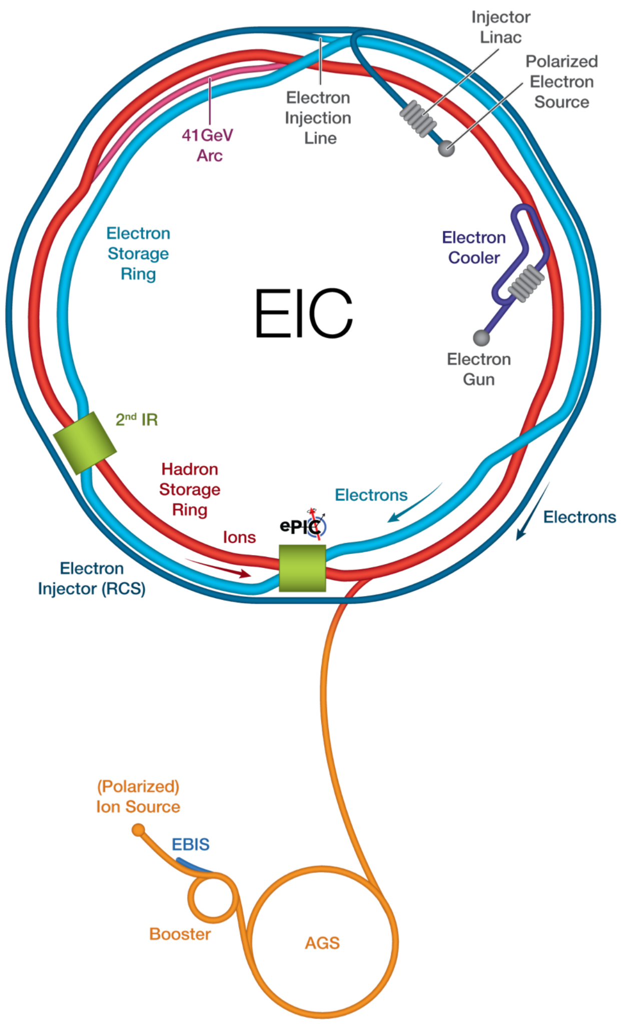
Markus Diefenthaler (Jefferson Lab)

Overview: ePIC Software & Computing

The Electron-Ion Collider (EIC)



Frontier accelerator facility in the U.S.

- **World's first collider of**:
  - Polarized electrons and polarized protons,
  - Polarized electrons and light ions (d, ^{3}He),
  - Electrons and heavy ions (up to Uranium).
- **High luminosity (**100 to 1000 times HERA luminosity**)** and **versatile range of center of mass energies** (20 GeV – 140 GeV), **beam polarizations** (longitudinal, transverse, tensor), and **beam species** (e; p → U).
- The EIC will enable us to embark on a **precision study of the nucleon and the nucleus at the scale of sea quarks and gluons**, over all of the kinematic range that is relevant.
- The **EIC Yellow Report (**Nucl.Phys.A 1026 (2022) 122447) describes the physics case, the resulting detector requirements, and the evolving detector concepts for the experimental program at the EIC.
- BNL and Jefferson Lab will be host laboratories for the EIC Experimental Program. Leadership roles in the EIC project are shared.

ePIC Collaboration

Formed in 2022/2023

ePIC Collaboration Meeting at Jlab in January 2023.

ePIC Collaboration Meeting in Warsaw in July 2023.

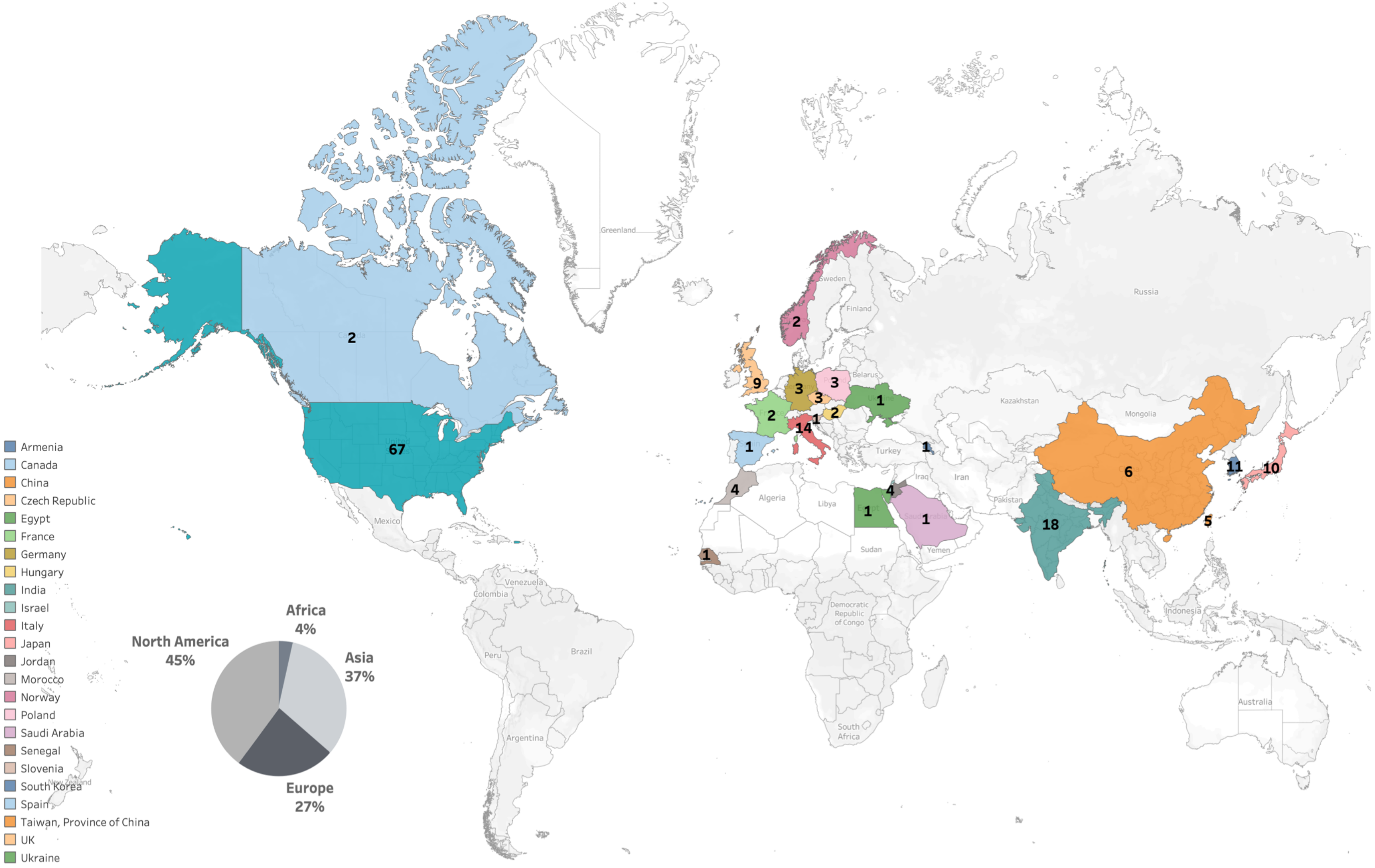
ePIC Collaboration Meeting at ANL in January 2024.

ePIC Members
> 850

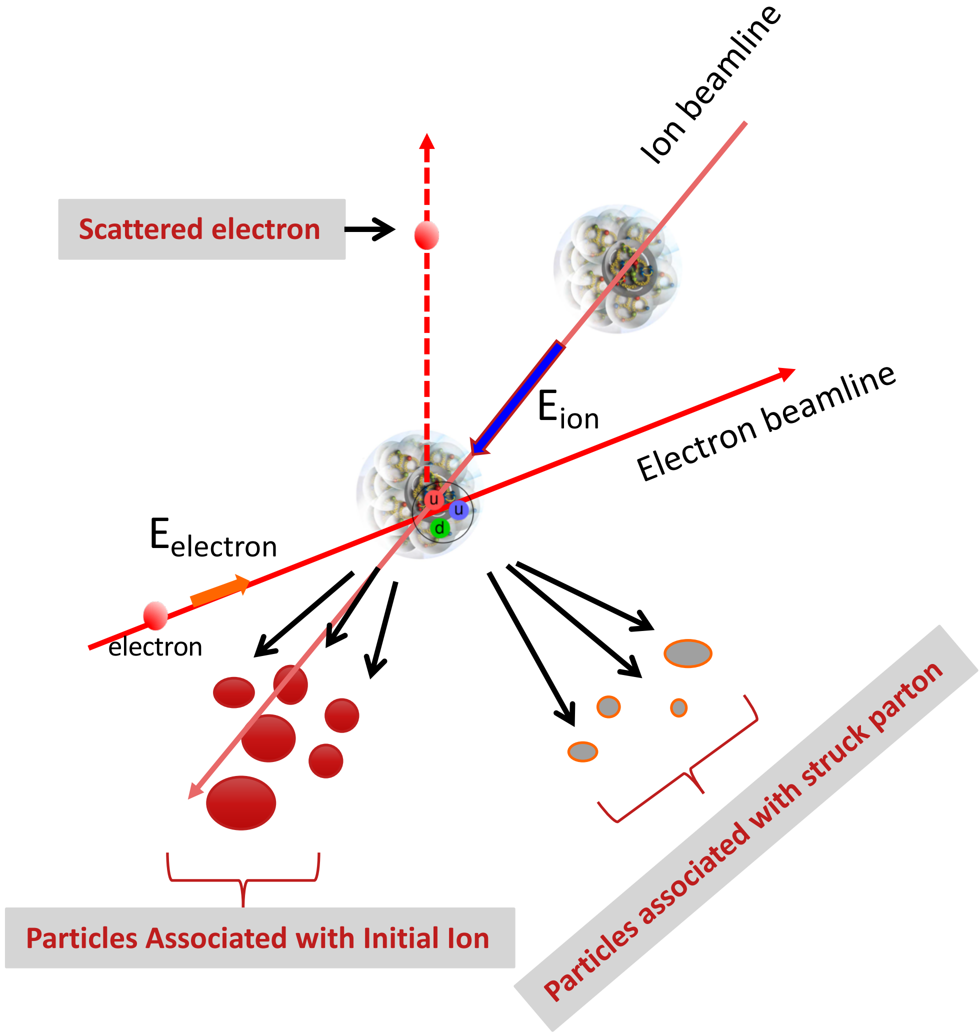
ePIC Institutions
173

ePIC Countries
25

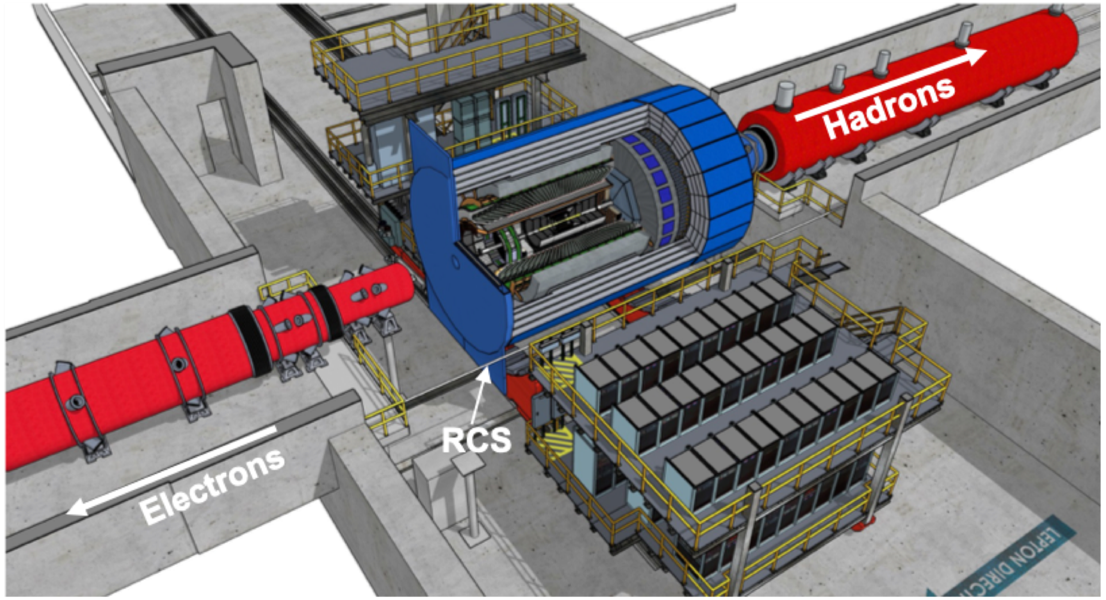
ePIC World Regions
4



General Purpose Detector for ePIC



Integrated interaction and detector region (+/- 45 m) to get ~100% acceptance for all final state particles, and measure them with good resolution.



Overall detector requirements:

- Large rapidity (-4 < h < 4) coverage; and far beyond in far-forward detector regions.
- Large acceptance solenoid of 1.7 T (up to 2 T).
- High control of systematics: luminosity monitor, electron and hadron polarimetry.

Compute-Detector Integration to Maximize Science

Broad ePIC Science Program:

- Plethora of observables, with less distinct topologies where every event is significant.
- High-precision measurements: Reducing systematic uncertainties of paramount importance.

Streaming Readout Capability Due to Moderate Signal Rate:

- **Capture every collision signal**, including background.
- Event selection using all available detector data for **holistic reconstruction**:
  - **Eliminate trigger bias** and provide accurate estimation of uncertainties during event selection.
- Streaming background estimates ideal to **reduce background** and related systematic uncertainties.

	EIC	RHIC	LHC → HL-LHC
Collision species	$\vec{e}+\vec{p}, \vec{e}+A$	$\vec{p}+\vec{p}/A, A+A$	$p+p/A, A+A$
Top x-N C.M. energy	140 GeV	510 GeV	13 TeV
Peak x-N luminosity	10^{34} cm^{-2} s^{-1}	10^{32} cm^{-2} s^{-1}	$10^{34} \rightarrow 10^{35}$ cm^{-2} s^{-1}
x-N cross section	**50 µb**	**40 mb**	**80 mb**
Top collision rate	500 kHz	10 MHz	1-6 GHz
$dN_{ch}/d\eta$	0.1-Few	~3	~6
Charged particle rate	**4M N_{ch}/s**	**60M N_{ch}/s**	**30G+ N_{ch}/s**

Compute-Detector Integration to Accelerate Science

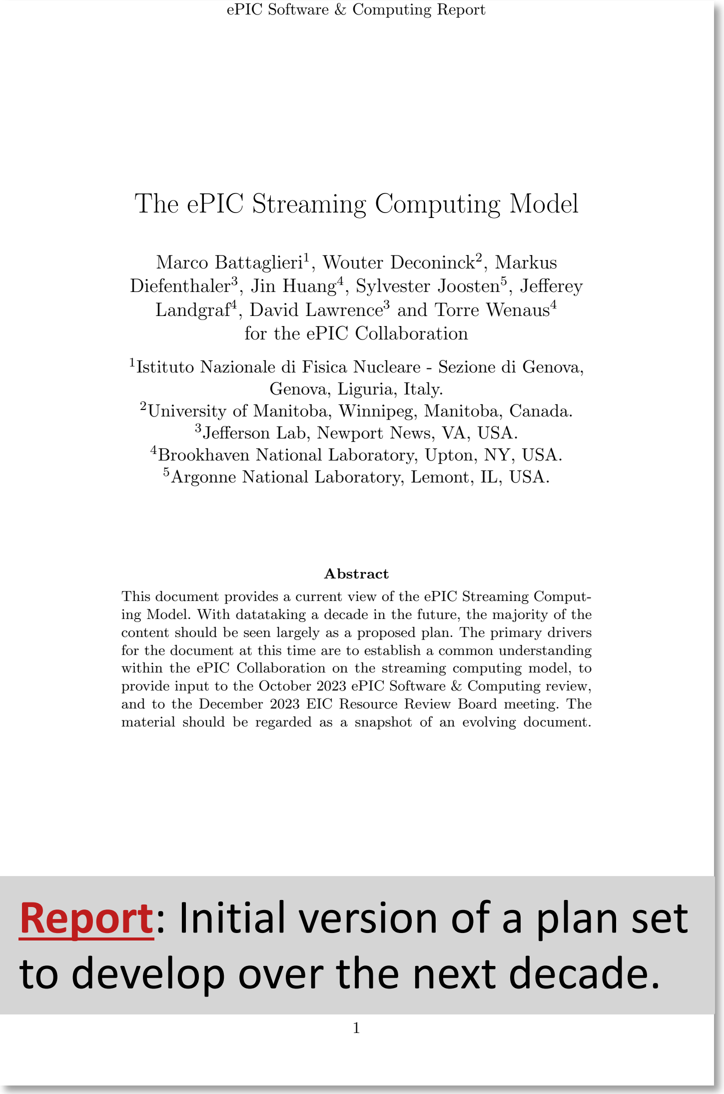
- **Problem** Data for physics analyses and the resulting publications available after O(1year) due to complexity of NP experiments (and their organization).
  - Alignment and calibration of detector as well as reconstruction and validation of events time-consuming.
- **Goal** **Rapid turnaround of 2-3 weeks for data for physics analyses**.
  - Timeline driven by calibrations.
- **Solution** Compute-detector integration using:

AI for autonomous alignment and calibration as well as reconstruction and validation for rapid processing.

Streaming readout for continuous data flow of the full detector information.

Heterogeneous computing for acceleration.

ePIC Streaming Computing Model



ePIC Software & Computing Report

The ePIC Streaming Computing Model

Marco Battaglieri[1], Wouter Deconinck[2], Markus Diefenthaler[3], Jin Huang[4], Sylvester Joosten[5], Jefferey Landgraf[4], David Lawrence[3] and Torre Wenaus[4]
for the ePIC Collaboration

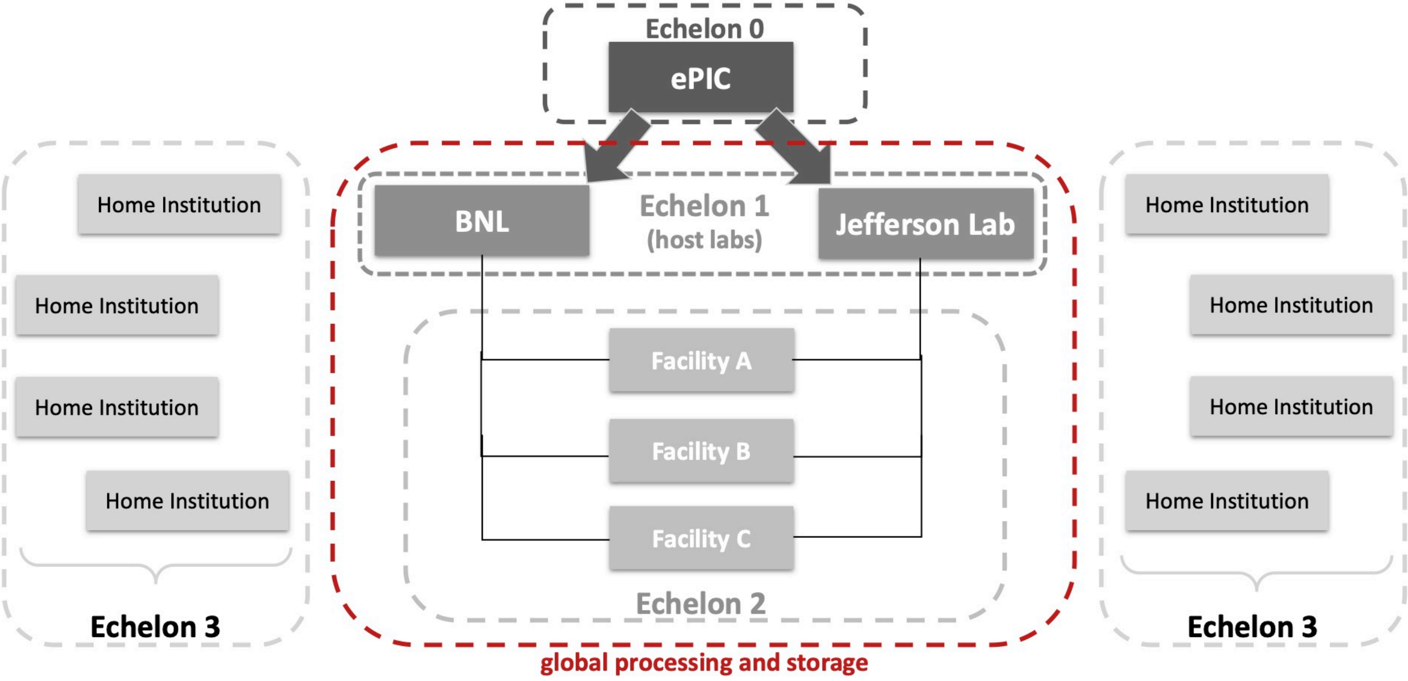
[1]Istituto Nazionale di Fisica Nucleare - Sezione di Genova, Genova, Liguria, Italy.
[2]University of Manitoba, Winnipeg, Manitoba, Canada.
[3]Jefferson Lab, Newport News, VA, USA.
[4]Brookhaven National Laboratory, Upton, NY, USA.
[5]Argonne National Laboratory, Lemont, IL, USA.

Abstract

This document provides a current view of the ePIC Streaming Computing Model. With datataking a decade in the future, the majority of the content should be seen largely as a proposed plan. The primary drivers for the document at this time are to establish a common understanding within the ePIC Collaboration on the streaming computing model, to provide input to the October 2023 ePIC Software & Computing review, and to the December 2023 EIC Resource Review Board meeting. The material should be regarded as a snapshot of an evolving document.

1

Report: Initial version of a plan set to develop over the next decade.



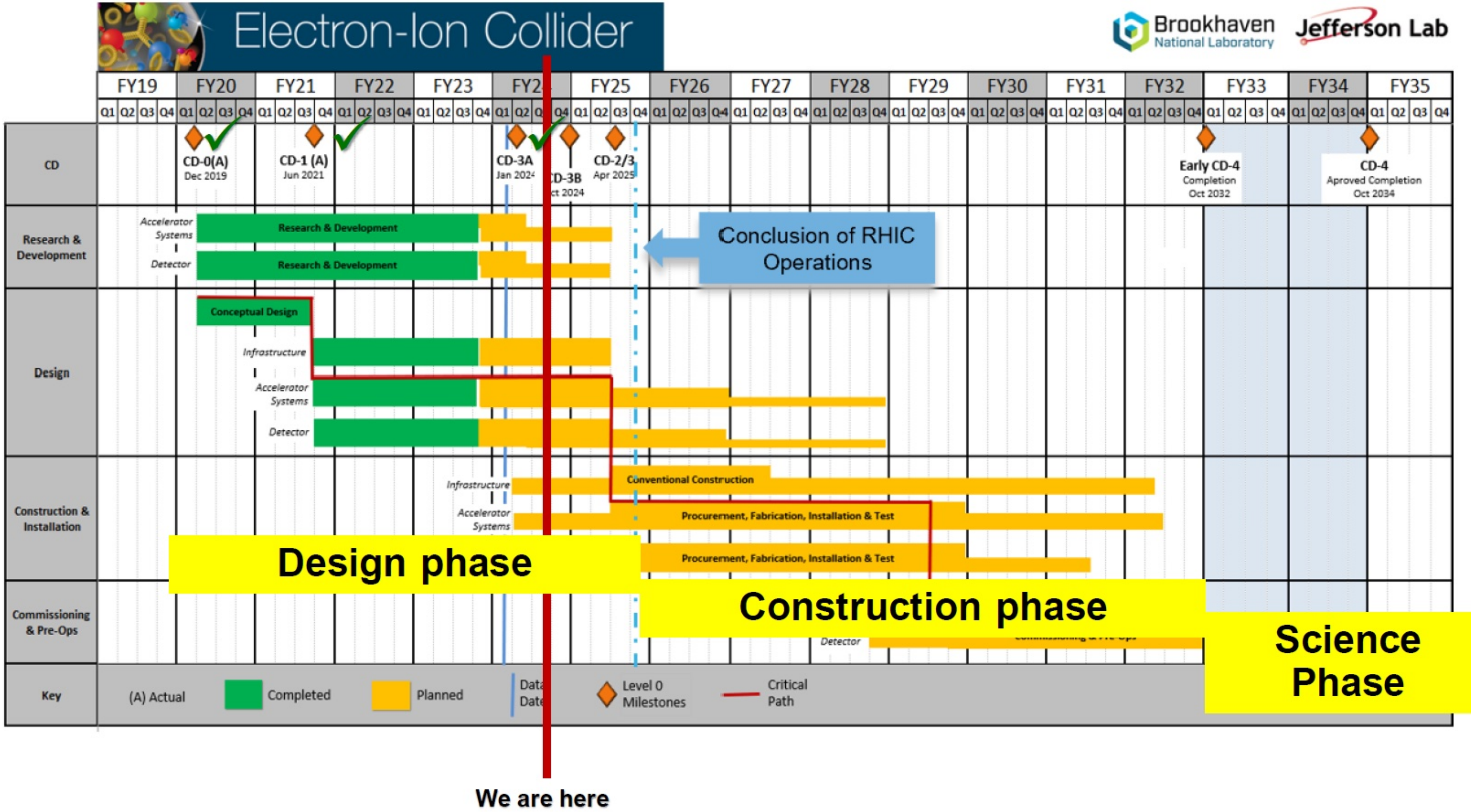
Echelon 0: ePIC experiment.

Echelon 1: Crucial and innovative partnership between host labs.

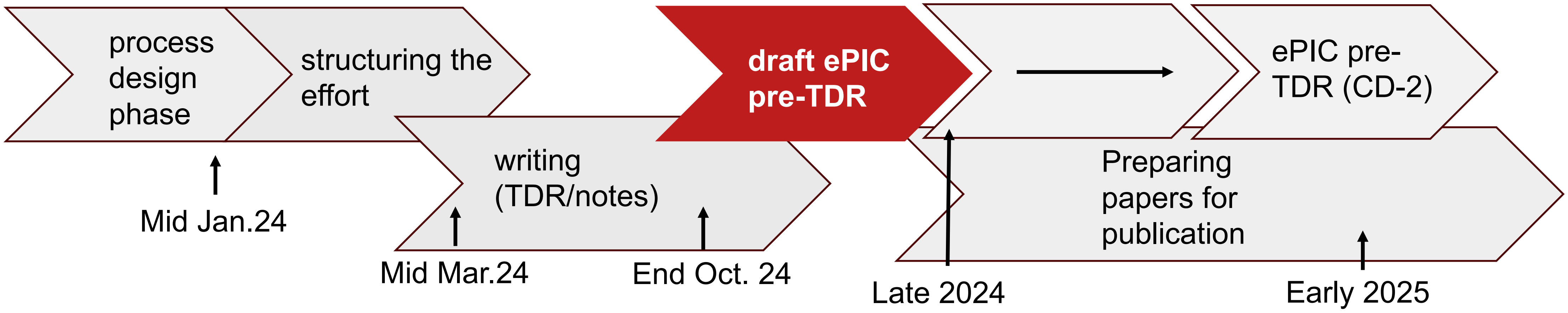
Echelon 2: Essential global contributions.

Echelon 3: Full support of the analysis community.

EIC Operations in About a Decade



TDR Strategy and Publications



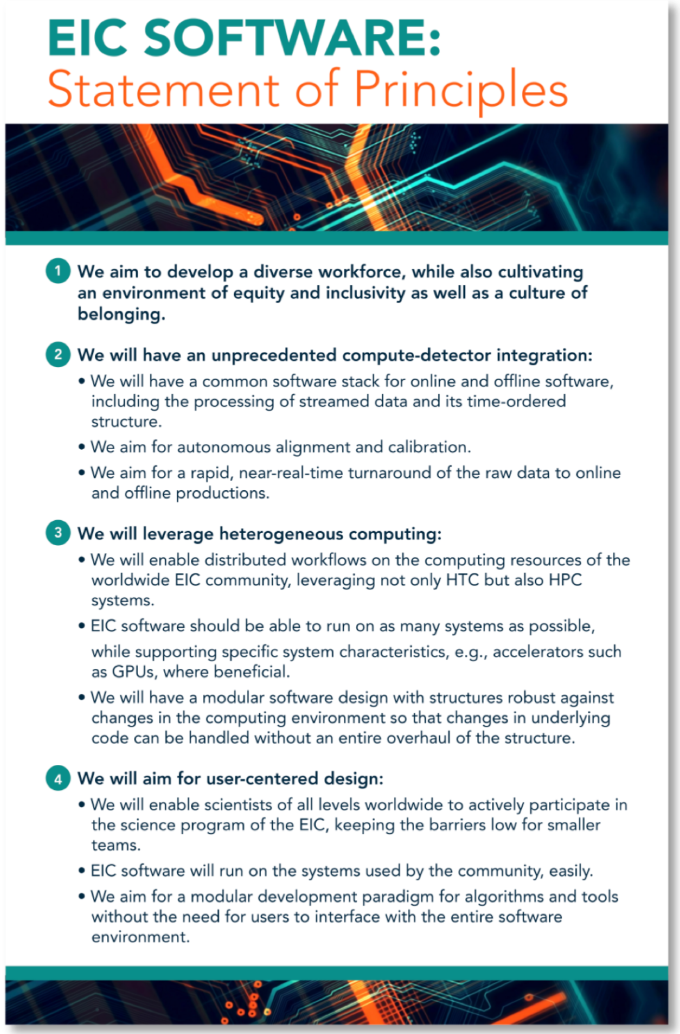
- ePIC Software & Computing is essential to the TDR, providing **advanced software** and **simulation productions** that are the **input for detector and physics studies**:
  - ***"Software and Simulation Readiness for TDR"*** parallel session and plenary discussion at the ANL collaboration meeting, where we extensively defined the remaining development tasks, drawing on significant input from the collaboration at large.
  - Good progress since collaboration meeting, improving the accuracy of the simulations and building up the reconstruction in shared priorities with Detector Subsystem Collaborations and Physics WGs.

ePIC Software: Introduction

Our Philosophy

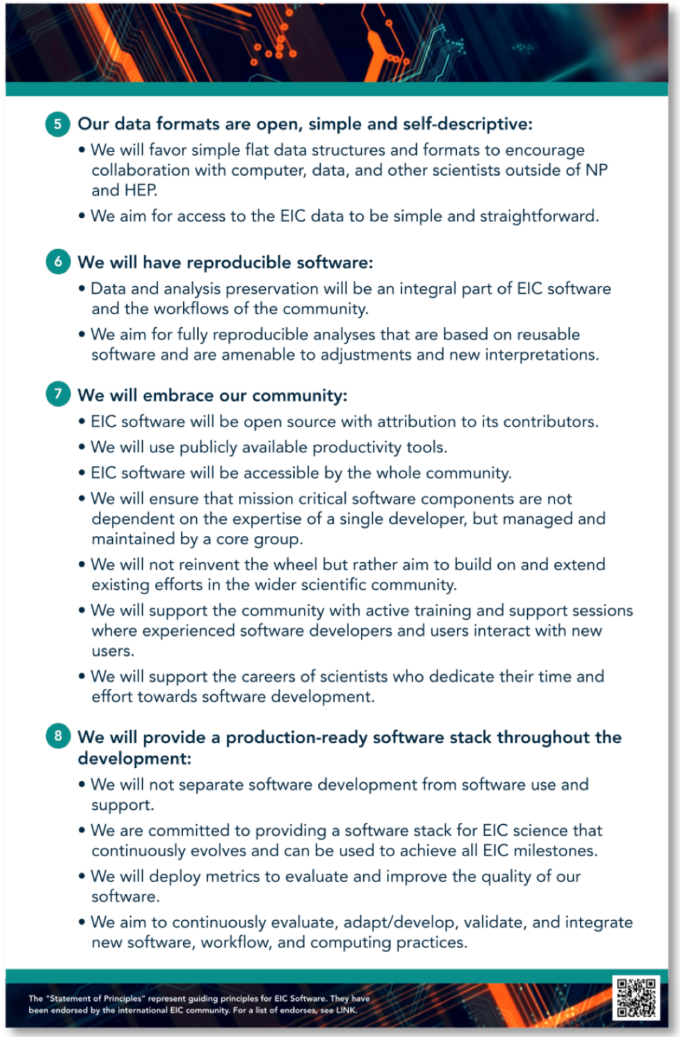
- We focus on **modern scientific software & computing practices** to ensure the **long-term success of the EIC scientific program** throughout all CD milestones.
  - Strong emphasis on modular, orthogonal tools.
  - Integration with HTC/HPC, CI workflows, and enable use of standard data science toolkits.
- We **leverage cutting edge sustainable community software** where appropriate, **avoiding the "not invented here" syndrome**.
  - Can build our software on top of a mature, well-supported, and actively developed software stack by using modern community tools, e.g. from CERN, the HPC community, and the data science community.
  - Actively collaborate with external software projects, while externalizing some support burden to external projects.
- We embrace these practices today to avoid starting our journey to EIC with technical debt.
- **We are writing software for the future, not the lowest common denominator of the past!**

Foundation of ePIC Software Stack



EIC SOFTWARE:
Statement of Principles

1. **We aim to develop a diverse workforce, while also cultivating an environment of equity and inclusivity as well as a culture of belonging.**
2. **We will have an unprecedented compute-detector integration:**
   - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
   - We aim for autonomous alignment and calibration.
   - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.
3. **We will leverage heterogeneous computing:**
   - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
   - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
   - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.
4. **We will aim for user-centered design:**
   - We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
   - EIC software will run on the systems used by the community, easily.
   - We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.

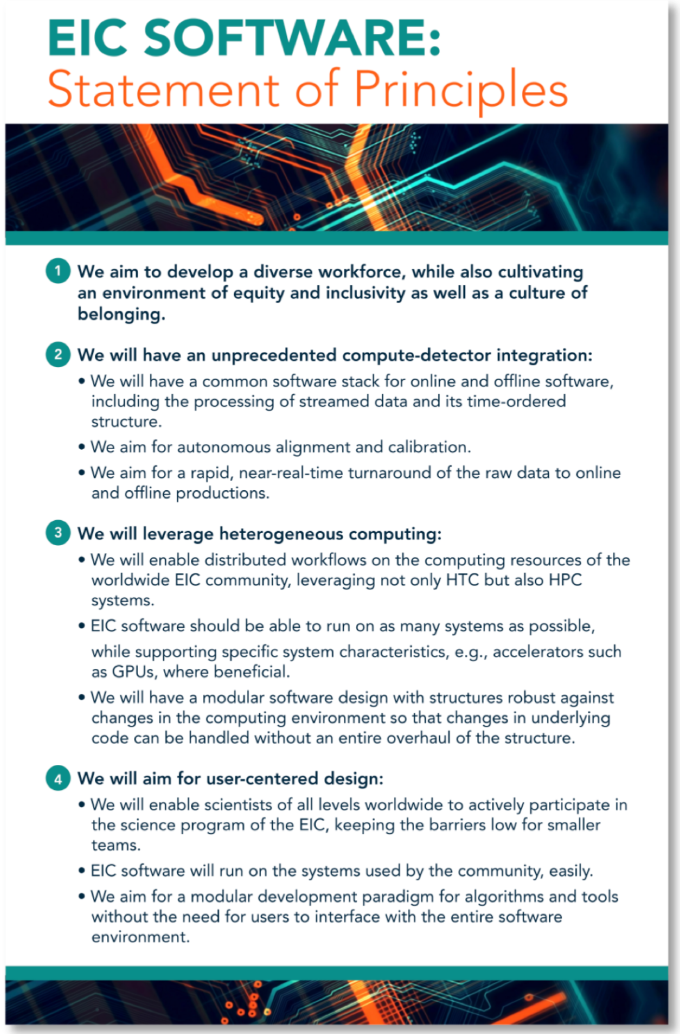


5. **Our data formats are open, simple and self-descriptive:**
   - We will favor simple flat data structures and formats to encourage collaboration with computer, data, and other scientists outside of NP and HEP.
   - We aim for access to the EIC data to be simple and straightforward.
6. **We will have reproducible software:**
   - Data and analysis preservation will be an integral part of EIC software and the workflows of the community.
   - We aim for fully reproducible analyses that are based on reusable software and are amenable to adjustments and new interpretations.
7. **We will embrace our community:**
   - EIC software will be open source with attribution to its contributors.
   - We will use publicly available productivity tools.
   - EIC software will be accessible by the whole community.
   - We will ensure that mission critical software components are not dependent on the expertise of a single developer, but managed and maintained by a core group.
   - We will not reinvent the wheel but rather aim to build on and extend existing efforts in the wider scientific community.
   - We will support the community with active training and support sessions where experienced software developers and users interact with new users.
   - We will support the careers of scientists who dedicate their time and effort towards software development.
8. **We will provide a production-ready software stack throughout the development:**
   - We will not separate software development from software use and support.
   - We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
   - We will deploy metrics to evaluate and improve the quality of our software.
   - We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

The "Statement of Principles" represent guiding principles for EIC Software. They have been endorsed by the international EIC community. For a list of endorses, see LINK.

- Community document that encodes our aspirations (technical and cultural) for software and computing at the EIC.
- Co-written and endorsed by a large group representing the international EIC community.

EIC Software: Statement of Principles [PDF]



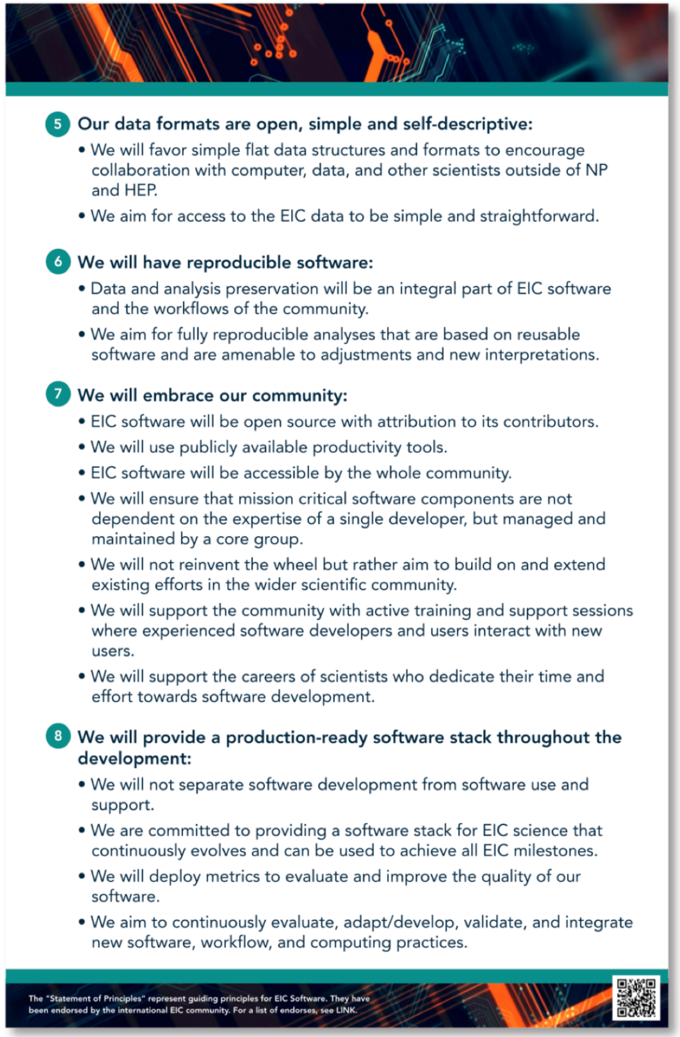
EIC SOFTWARE:
Statement of Principles

1. **We aim to develop a diverse workforce, while also cultivating an environment of equity and inclusivity as well as a culture of belonging.**

2. **We will have an unprecedented compute-detector integration:**
   - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
   - We aim for autonomous alignment and calibration.
   - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.

3. **We will leverage heterogeneous computing:**
   - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
   - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
   - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.

4. **We will aim for user-centered design:**
   - We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
   - EIC software will run on the systems used by the community, easily.
   - We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.



5. **Our data formats are open, simple and self-descriptive:**
   - We will favor simple flat data structures and formats to encourage collaboration with computer, data, and other scientists outside of NP and HEP.
   - We aim for access to the EIC data to be simple and straightforward.

6. **We will have reproducible software:**
   - Data and analysis preservation will be an integral part of EIC software and the workflows of the community.
   - We aim for fully reproducible analyses that are based on reusable software and are amenable to adjustments and new interpretations.

7. **We will embrace our community:**
   - EIC software will be open source with attribution to its contributors.
   - We will use publicly available productivity tools.
   - EIC software will be accessible by the whole community.
   - We will ensure that mission critical software components are not dependent on the expertise of a single developer, but managed and maintained by a core group.
   - We will not reinvent the wheel but rather aim to build on and extend existing efforts in the wider scientific community.
   - We will support the community with active training and support sessions where experienced software developers and users interact with new users.
   - We will support the careers of scientists who dedicate their time and effort towards software development.

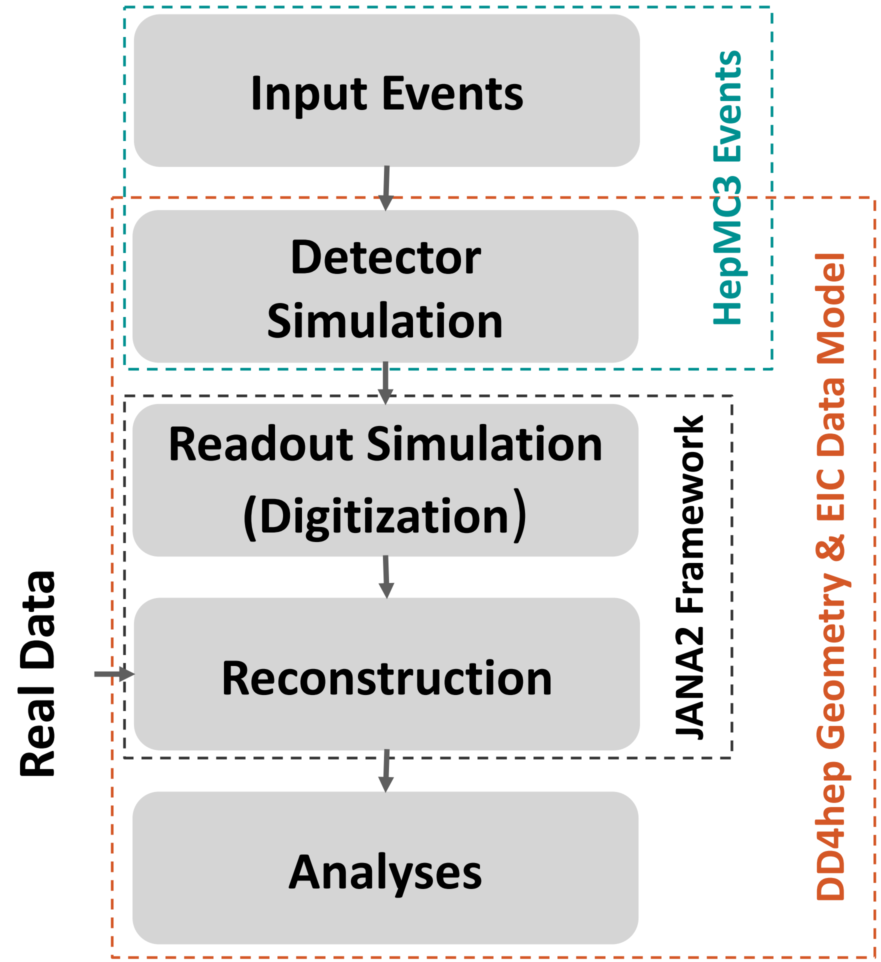
8. **We will provide a production-ready software stack throughout the development:**
   - We will not separate software development from software use and support.
   - We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
   - We will deploy metrics to evaluate and improve the quality of our software.
   - We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

The "Statement of Principles" represent guiding principles for EIC Software. They have been endorsed by the international EIC community. For a list of endorses, see LINK.

EIC Software is:

1. Diverse
2. Integrative
3. Heterogeneous
4. User-centered
5. Accessible
6. Reproducible
7. Collaborative
8. Agile

ePIC Software Stack: A Modular Simulation, Reconstruction, and Analysis Toolkit



Input events from **MC event generators** or particle guns, with optional background merging.

Geant4 simulations with **DD4hep** for geometry description and exchange, output data in the **EIC Data Model** (EDM4hep + EDM4eic, described in Podio).

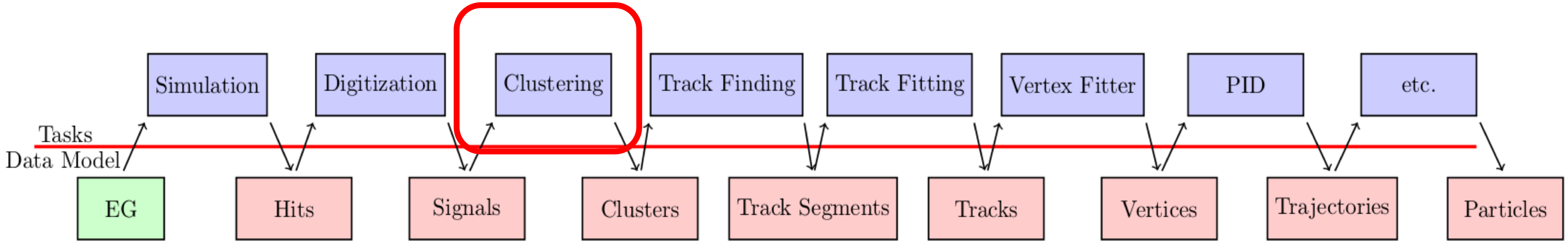
Algorithms to transform the GEANT4 hits to mimic real detector readout, including background stacking, "pileup", DAQ frames.

Realistic reconstruction algorithms starting from raw detector output (from digitization or real data).

User analyses in plain C++/ROOT or Python/uproot, facilitated by using a flat data model.

Continuous integration for detector and physics benchmarks and monthly production campaigns ensure a production-ready software stack at any time.

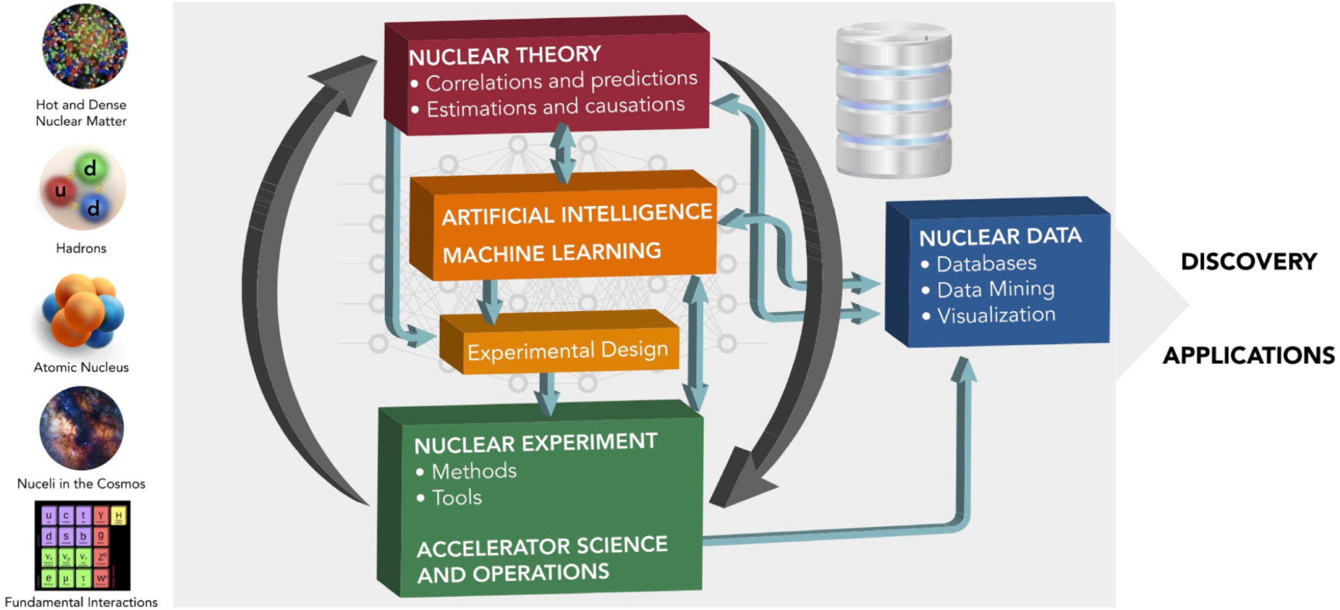
Data-Driven API Design and the EIC Data Model



- Use of **standard interfaces** between individual simulation, reconstruction, and analysis tasks **creates modularity** that enables **easy exchange of components**
- Example: multiple clustering algorithms can be swapped out, as long as they adhere to the data model interfaces
- We standardized on EDM4eic (an extended version of EDM4hep from the Key4HEP project) for our entire software stack, and **HepMC3 for Monte Carlo input**.
- This modularity extends beyond the EIC community, since many data structures are common across NP and HEP experiments worldwide

Standardized interfaces between components ensure modularity.

Artificial Intelligence at ePIC



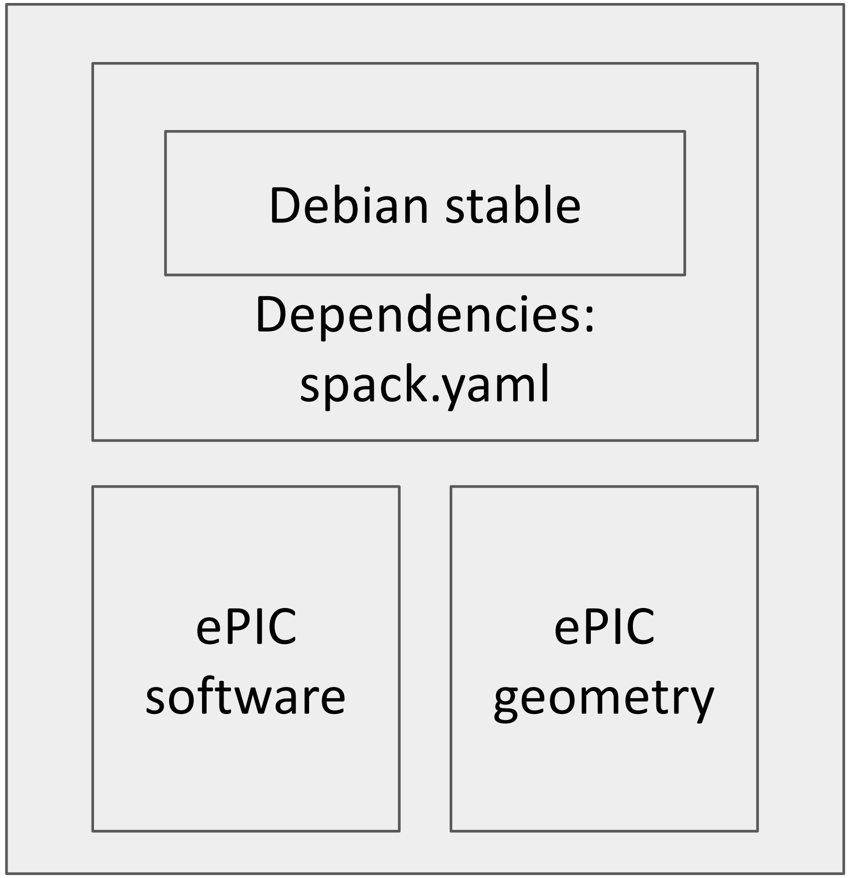
- AI already has an important presence in ePIC, with many usages in the prototyping stage, e.g., for detector surrogates and reconstruction methods.
- To explore and develop the full potential of AI for ePIC, we will move from prototyping to production, adding powerful AI approaches into our workflows.

First step: Integrate existing AI reconstruction routines, e.g., for calorimeter reconstruction, into the reconstruction framework:

- Currently using TensorFlow and Torch for learning and standalone inference.
- Integrated ONXX framework for inference.
- First production-ready AI reconstruction algorithms foreseen for August simulation campaign.

ePIC Software: Development & Deployment

Deployment with Containers



- Provide a single curated software build "**eic-shell**" for local development, CI, and production campaigns
  - Multiple architecture-specific versions of images where needed (e.g. amd64 and aarch64).
  - Build docker image and converted singularity image.
- **Different flavors**:
  - **nightly:** All master branches, built every night.
  - **stable/tagged:** Release versions.
  - **unstable**: Temporary containers for Pull Requests.
- **Distribution**:
  - DockerHub & Github Registry: Sll docker images.
  - eicweb: Internal docker images, all singularity images
  - CVMFS: OSG ~6 hour synchronizations t: /cvmfs/singularity.opensciencegrid.org

Local Software Deployment with eic-shell

Step 1: curl -L get.epic-eic.org | bash

Step 2: ???

Step 3: Profit

Easy to get started locally…

in only one line!

- Uses deployed images on /cvmfs when available, downloads singularity sifs otherwise.
- Rolling out seamless container updates to end users.
- At the same time basis of scalable computing on OSG: same containers are used everywhere.
- Note: In principle not even needed to look at data (flat format!).

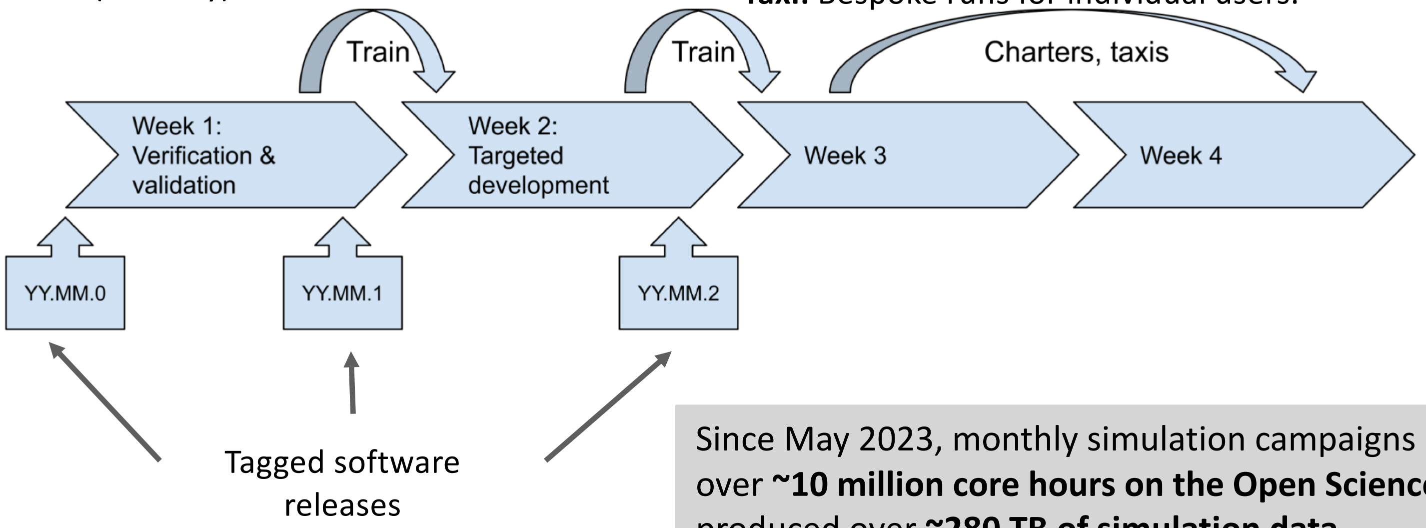
Monthly Simulation Productions

1. **Continuous deployment** of the software used for detector and physics simulations.
2. **Regular updates** of simulation productions for detector and physics studies in preparation for the TDR (and subsequent CD milestones).
3. **Timely validation and quality control** for simulation productions on datasets that require substantial time and resources. Focus on **benchmarks driven by Continuous Integration** (CI), a process that automates the testing and building of software.

Train: Major central campaign at a fixed (monthly) schedule.

Charter: Special interest runs for working groups.

Taxi: Bespoke runs for individual users.



Broad science program for the EIC: The selection of physics processes and associated Monte Carlo simulations for the TDR has been finalized with the Physics Working Groups.

Since May 2023, monthly simulation campaigns have used over **~10 million core hours on the Open Science Grid** and produced over **~280 TB of simulation data**.

ePIC Simulation Campaigns

Details on https://eic.github.io/epic-prod/

Monthly Production Updates:

- Provided at the end of each campaign on the main mailing list of the collaboration.
- Version Format: Year.Month

Live Updates:

- Follow the **firehose** Mattermost channel for real-time information.

Previous Campaign Information:

- Access **reconstructed output files** and **full Geant4 simulation output files** on our campaign history pages.

The list of datasets are displayed in a tree structure with:

- **XrootD server** and **base address** at the top,
- followed by **RECO** for reconstructed output files or **FULL** for full Geant4 simulation files and the **version**.
- Then they are organized by **detector config**, **physics process**, and **beam properties**.

To list the reconstructed output files for a particular dataset, first start eic-shell and execute:

```
xrdfs root://dtn-eic.jlab.org
```

XRootD server

```
ls /work/eic2/EPIC/RECO/23.12.0/epic_craterlake/DIS/NC/10x100/minQ2=10
```

Base Address | TYPE and Version | Detector Config | Physics | Beam Properties

GitHub: EIC Organization and Managed Runners

- **Recommended standard interface for all source code projects in ePIC.**
- Modest computational resources:
  - 20 dual-core job slots for all projects under github.com/eic
- **User management and workflow**:
  - Everyone can get a GitHub account and every EIC user can get EIC organization membership.
  - All new contributions happen through a pull request (PR),
  - Code can only be merged if it passess all CI checks and passes expert review.
  - All PRs are squash-merged into the main branch to maintain a clear history.
  - We strongly encourage users to make small incremental changes to most effectively develop software in a collaborative context.
- **Additional features**:
  - GitHub actions model of development: easily shared across all of GitHub.
  - GitHub pages for presentation (e.g. https://eic.github.io/epic/craterlake_views).
  - Also: GitHub pages for documentation (e.g. https://eic.github.io/epic/).

Open, Collaborative Software Development

Encourage Upstream Contributions

- Requirements of well-formed HepMC as input has resulted in real improvements to multiple MCEGs used by EIC community.
- Various upstream contributions to DD4hep, ACTS, Spack, uproot,...

Encourage Social Coding

- CI platform provides the incentive for developers to commit code frequently: achieving data management and analysis preservation goals.
- Pull request reviews to ensure higher quality code and build developer skills.

Enable Access Without Restrictions

- ePIC collaboration members include over 170 institutions worldwide
- Data ‘publicly’ available through BNL S3 and publicly available through JLab xrootd.
- Flat data structures (i.e. could be a csv), stored as ubiquitous ROOT trees without need for data structure libraries.
- Support for uproot using numpy library (awkward not needed).

Approaches Under Evaluation

- Rucio for data management (moving into production).
- Reproducible analysis workflow tools.

ePIC Software: Organization

ePIC Software & Computing Organization

Spokesperson's Office

Detector | **Software & Computing** | Physics

Guiding Principles:

- *Diversity, Equity, and Inclusion*
- *Statement of Software Principles*
- *Sustainability.*

Software & Computing Coordinator
Markus Diefenthaler (Jefferson Lab)

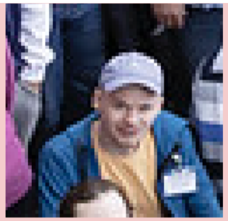
Cross-cutting Working Group:

- *Data and Analysis Preservation*

Deputy Coordinator (Operations)
Wouter Deconinck (U. Manitoba)

Operation Working Groups:

- Production
- User Learning
- Validation



Deputy Coordinator (Development)
Dmitry Kalinkin (U. Kentucky)

Development Working Groups:

- Physics and Detector Simulation
- Reconstruction
- *Analysis Tools*

Deputy Coordinator (Infrastructure)
Torre Wenous (BNL)

Infrastructure Working Groups:

- Streaming Computing Model
- *Multi-Architecture Computing*
- *Distributed Computing*

Community Building

Regular meetings to drive forward priority targets and provide an avenue for new collaboration members to engage.

12 pages of detailed notes that enabled software progress, pushed the review preparations, and informed our planning.

Discussed: Status and plans; software and simulations for TDR, tutorials; streaming computing; software projects with HEP.

Onboarding

Landing Page for Onboarding New Users:

- This page includes a continuously updated and improved list of **useful links**, **software tutorials**, and **frequently asked questions**.
- Any member of the collaboration can **directly contribute by submitting change requests**.

Landing Page

Get started

ePIC Tutorials

HEP Software Training Center

FAQ

Welcome to the **ePIC Landing Page**!

Our mailing list: eic-projdet-compsw-l@lists.bnl.gov

Active HelpDesk channel on Mattermost for **user support**.

User Learning

Software tutorial series, most recent during CERN meeting, covering four key topics:

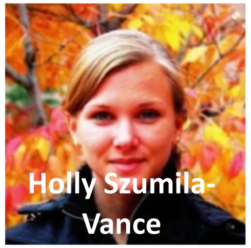
1. Overview of ePIC Software (Holly Szumila-Vance, Jefferson Lab)

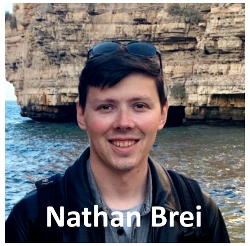
   Eic-shell Easy to get started locally… in only 1 line!

   ```
   curl -L get.epic-eic.org | bash
   ```

   Based on container images, the same images are used for simulation campaigns.

2. Working with Simulation Output (Stephen Kay, University of York)
3. Simulating Detectors and Their Readout (Simon Gardner, University of Glasgow)
4. Reconstruction Algorithms (Nathan Brei, Jefferson Lab)





ePIC Software: MCEGs

Criteria for MCEGs to be Included in Production

1. Must not duplicate effort. Need to have reference generator for each process.
2. Must be in hepmc3.tree.root format.
3. Must be version-tracked in a publicly accessible repository: Source code, steering files, run cards, etc. Follow the input preprocessing guidelines.

Working with the PWGs to revise the **reference list of physics processes** and **related MC samples** to be included in the simulation campaigns for the TDR: [Preliminary list], additional feedback on [exclusive, diffractive, and tagging processes].

File Nomenclature and Organization	**Example**
<physics processes>/<generator repository release tag>/<electron momentum>x<proton momentum>/q2_<minimum q2>to<maximum q2>/<generator repository release tag>_<physics processes>_<electron momentum>x<proton momentum>_q2_<minimum q2>to<maximum q2>_run<index>.hepmc3.tree.root **Currently being revised while defining Rucio naming schema!**	DIS/NC/pythia6.428-1.0/10x100/q2_10to100/pythia6.428-1.0_DIS-NC_10x100_q2_10to100_run001.hepmc3.tree.root

Overview of MCEGs Currently in Use

General-Purpose MCEGs

pythia6-eic 1.0.0

pythia8.306-1.0

MCEGs from NP Community

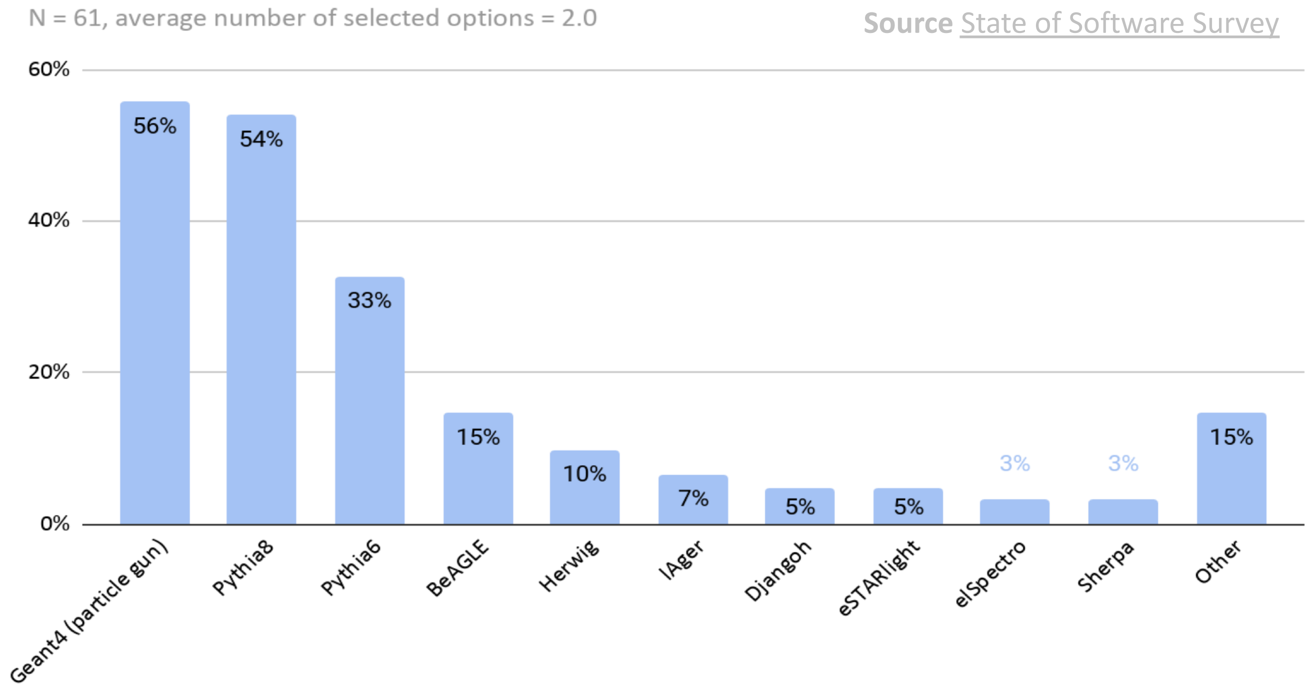
EpIC1.0.0-1.0

lAGER

DEMPGen 1.0.0

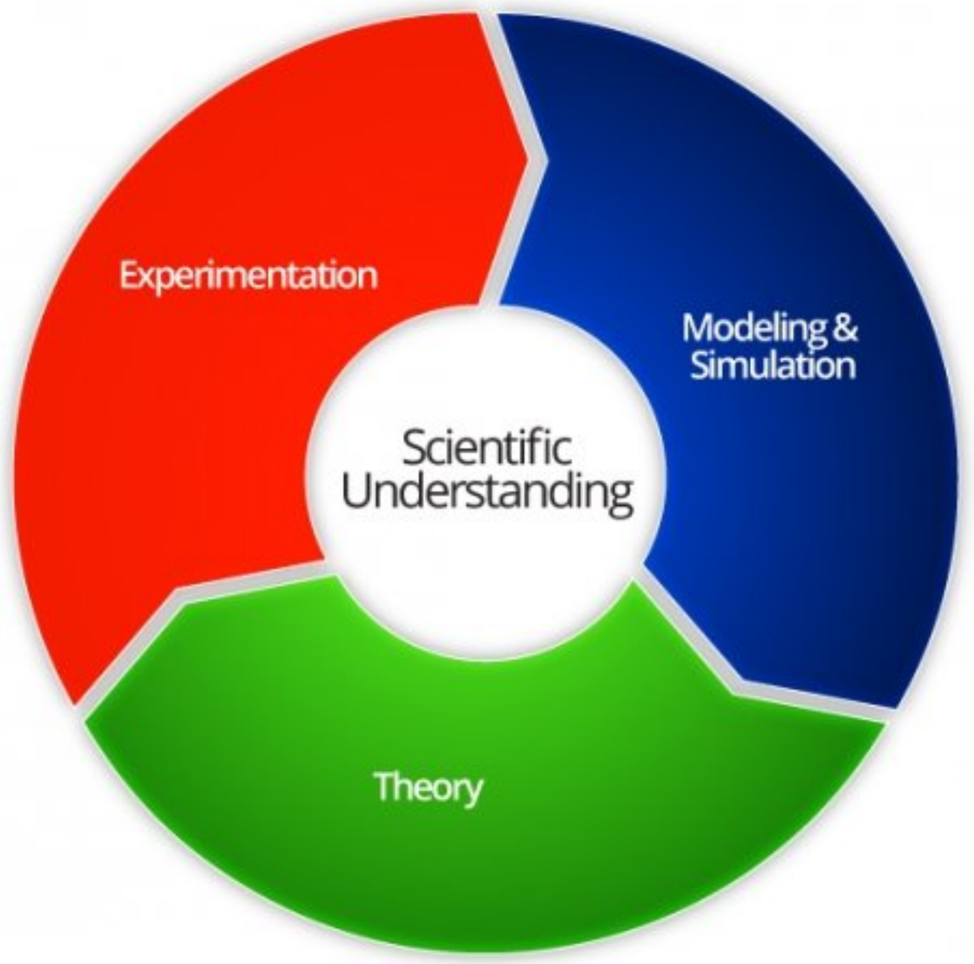
SARTRE sartre-1.39-1.0

Drastic change from MCEGs using during Yellow Report effort:



Other (N = 9): personal computer codes (N = 2), ACT, CLASDIS, ComptonRad, GRAPE-DILEPTON, MADX, MILOU, OPERA, RAYTRACE, Sartre, Topeg, ZGOUBI

MCEGs for the EIC: Requirements and Challenges



Monte Carlo Simulation of

- electron-proton (ep) collisions,
- electron-ion (eA) collisions, both light and heavy ions,
- including higher order QED and QCD effects,
- including a plethora of spin-dependent effects.

Common challenges, e.g. with HL-LHC: **High-precision QCD measurements require high-precision simulations**.

Unique challenges MCEGs for electron-**ion** collisions and **spin-dependent** measurements, including **novel QCD phenomena** (e.g., 3D quark-gluon imaging in momentum (**TMDs**) and position space (**GPDs**)).

MCEG Priorities: Wishlist

- Training of the EIC community, similar to the MCEG tutorials during the EICUG Yellow Report:
  - **Rivet** Christian Bierlich (LUND)**, Pythia 8** by Stefan Prestel (then LUND), **Herwig 7** Simon Plätzer (Vienna), **Sherpa 2** Stefan Höche (FNAL).
- **Validation of existing MCEGs using Rivet**:
  - Rivet analyses from HERA and CEBAF. Rivet for physics analyses from ePIC simulations?
  - Build automated workflows (CI).
- Development of a DIS tune.

- **Merging of higher order QED and QCD effects**.
- Roadmap for spin-dependent parton showers .
- Roadmap for spin-dependent hadronization models.
- Roadmap for eA, both light and heavy ions.
- **Feasibility of shared elements between general-purpose MCEGs.**

- Guidance on how to compare EIC measurements with theory: Folding and unfolding approaches.

MCEG Validation Status

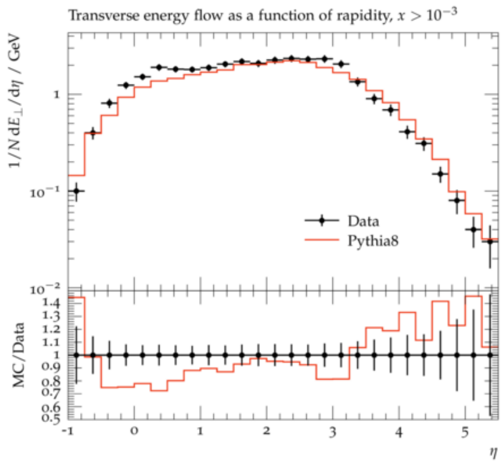
We are not starting from zero!

Tuning of MCEGs:

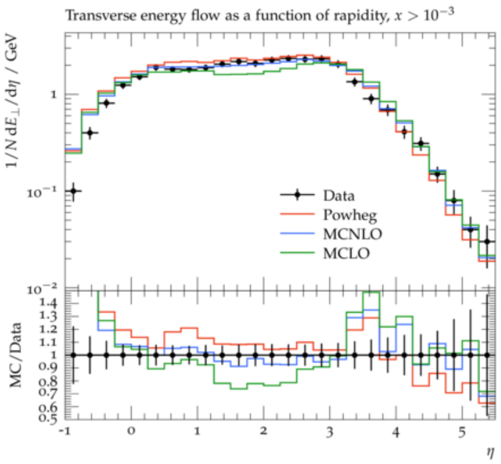
- MCEGs developed by EIC community, e.g., BeAGLE, has been compared to and tuned to selected ep and eA measurements.
- Pythia6 version used by EIC community has been tuned to HERMES and other experiments in detail. Modeling in interim region, 1 GeV^2 < Q^2 < 10 GeV^2, based on HERMES data.

Ongoing activity on validation of general-purpose MCEGs:

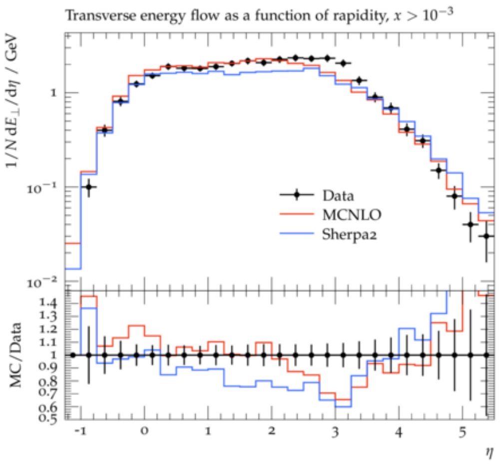
- Comparison to published DIS results using RIVET and understand differences.
- **Provide initial findings and results in publication (draft available):**
  - Overview of where we stand in understanding HERA data with current physics and models implement in MCEGs.



Pythia8

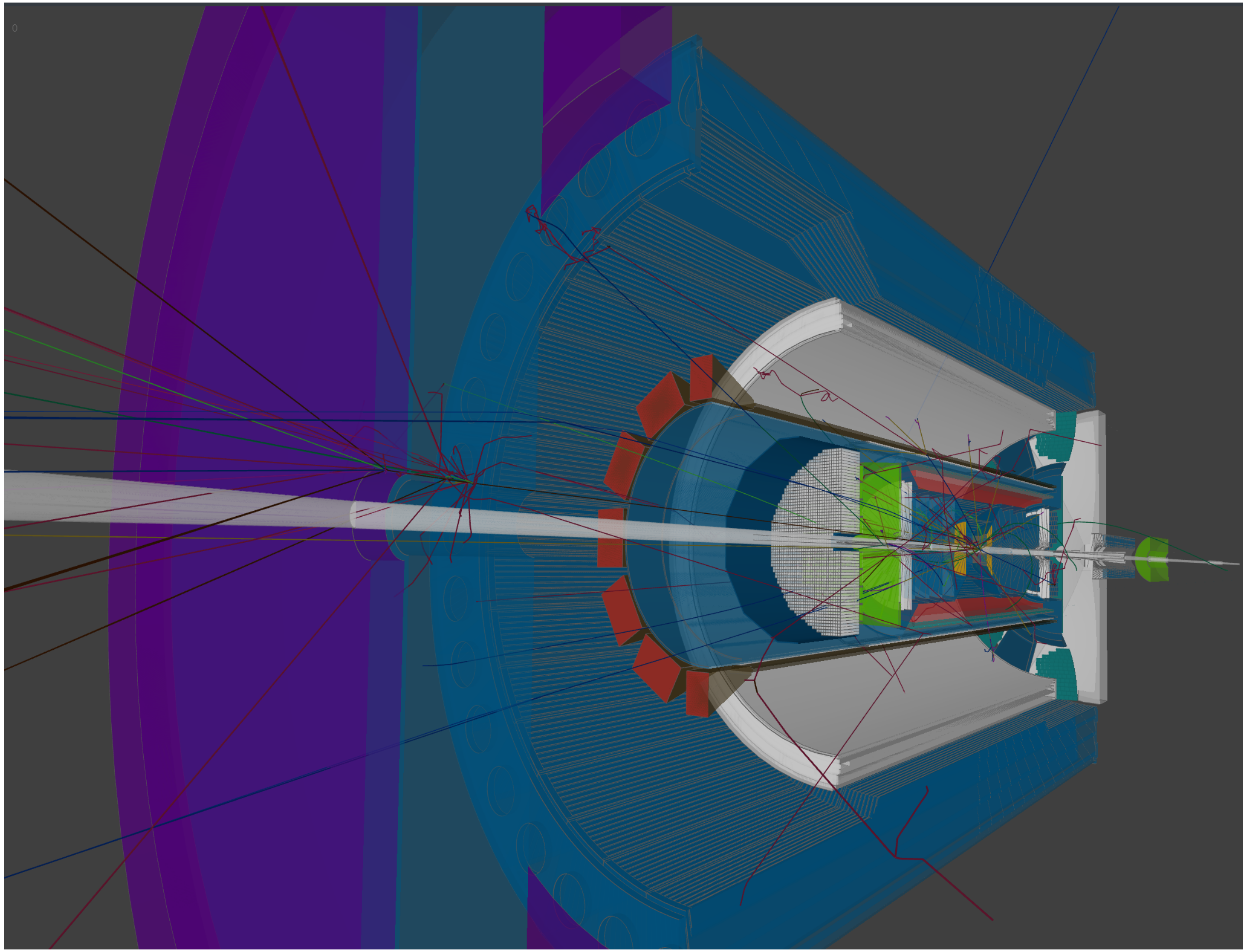


Herwig7



Sherpa2

Visualization via Firebird: Web-Based Event Display Based on Phoenix



Summary

- The ePIC software stack is a **modern** and **modular toolkit** built from NP/HEP community tools and components from HPC and Data Science; ePIC is an **active** member of the **NP/HEP software & computing community**.

- **High level milestones** ensures that the agile development process is continuously confronted with real world exercising of the software and the developing realization of the computing model:
    - Priority always given to meeting near-term needs. ePIC leverages monthly production campaigns, CI-driven benchmarks, and timeline-based prioritization to **ensure timely completion of the simulation studies for the Technical Design Report**.
    - Longer range timeline progressively exercising the streaming computing model to deliver for the needs of the CD process, for specific applications, e.g. test beams, for scaling and capability challenges, and ultimately for the phases of datataking.
    - The timeline, including the schedule for the Technical Design Report, allows for the needed **modernization of the software stack for physics simulation**, contingent upon receiving support from the MCEG community for comparison studies and subsequent validation.

Involvement from the MCEG community essential.
Need to consider organization and funding opportunities.

Backup

Deeply Learning DIS Kinematics https://inspirehep.net/literature/1912344

Slide from Abdullah Farhat (ODU)

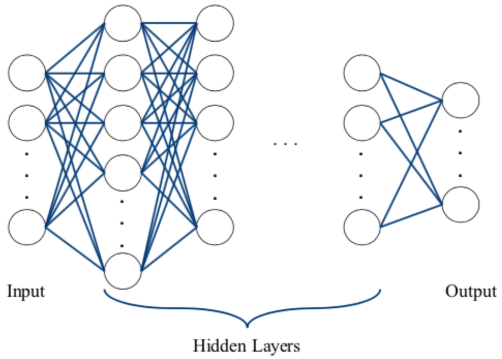
The idea is to build a physics-driven neural network method that combines the quantities determined with the classical methods in a way:

$$Q^2_{NN} = A_{Q^2}(Q^2_{EL}, Q^2_{DA}, Q^2_{JB}) + L_{Q^2}(A_{Q^2}, E_{l'}, \theta_{l'}) + H_{Q^2}(A_{Q^2}, P_{T,H}, \delta_H)$$

And reconstruct x, with Q^2_{NN} as an input, in the form:

$$x_{NN} = A_x(x_{EL}, x_{DA}, x_{JB}) + L_x(A_x, Q^2_{NN}, E_{l'}, \theta_{l'}) + H_x(A_x, Q^2_{NN}, P_{T,H}, \delta_H)$$

Neural networks can be used to reconstruct the kinematics by weighting classical reconstructions and using all four of the measured quantities as corrections.

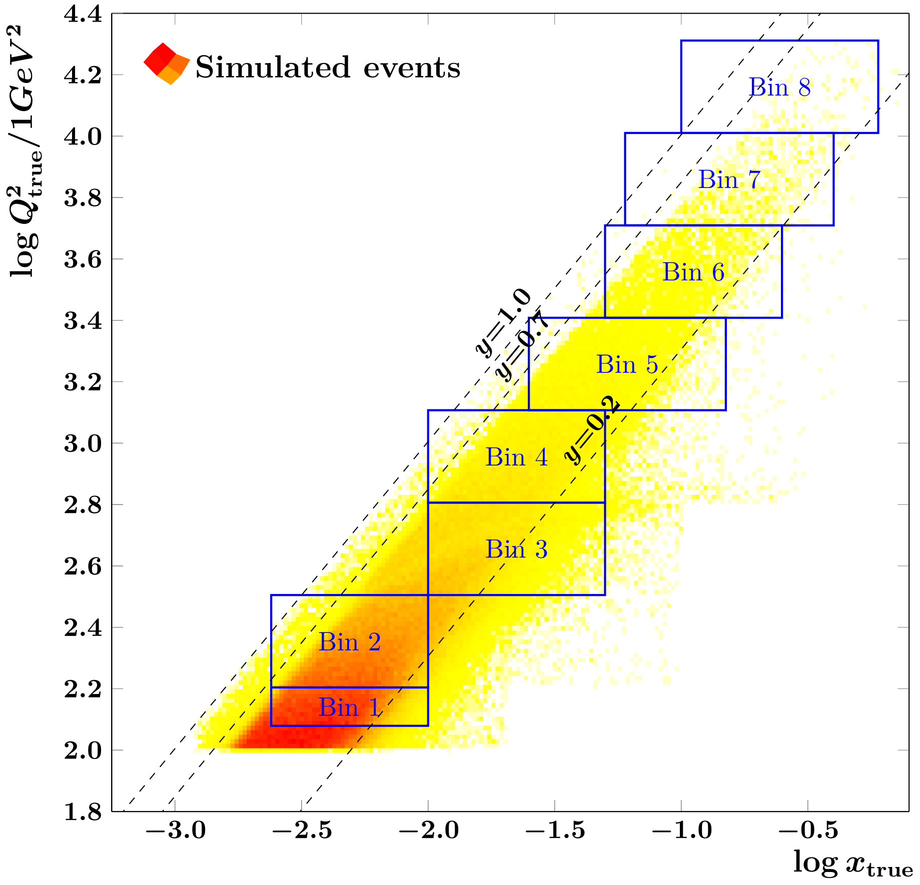


Properties of the DNN

- **Effective**: Universal approximation capability: can approximate any continuous function to arbitrary accuracy
- **Robust**: Increasing the depth of the network (i.e. the number of terms in the sum) necessarily reduces the error
- **Computationally Efficient**: Structure avoids ”vanishing” gradients arising in the backpropagation algorithm

Results for *x* and Q^2

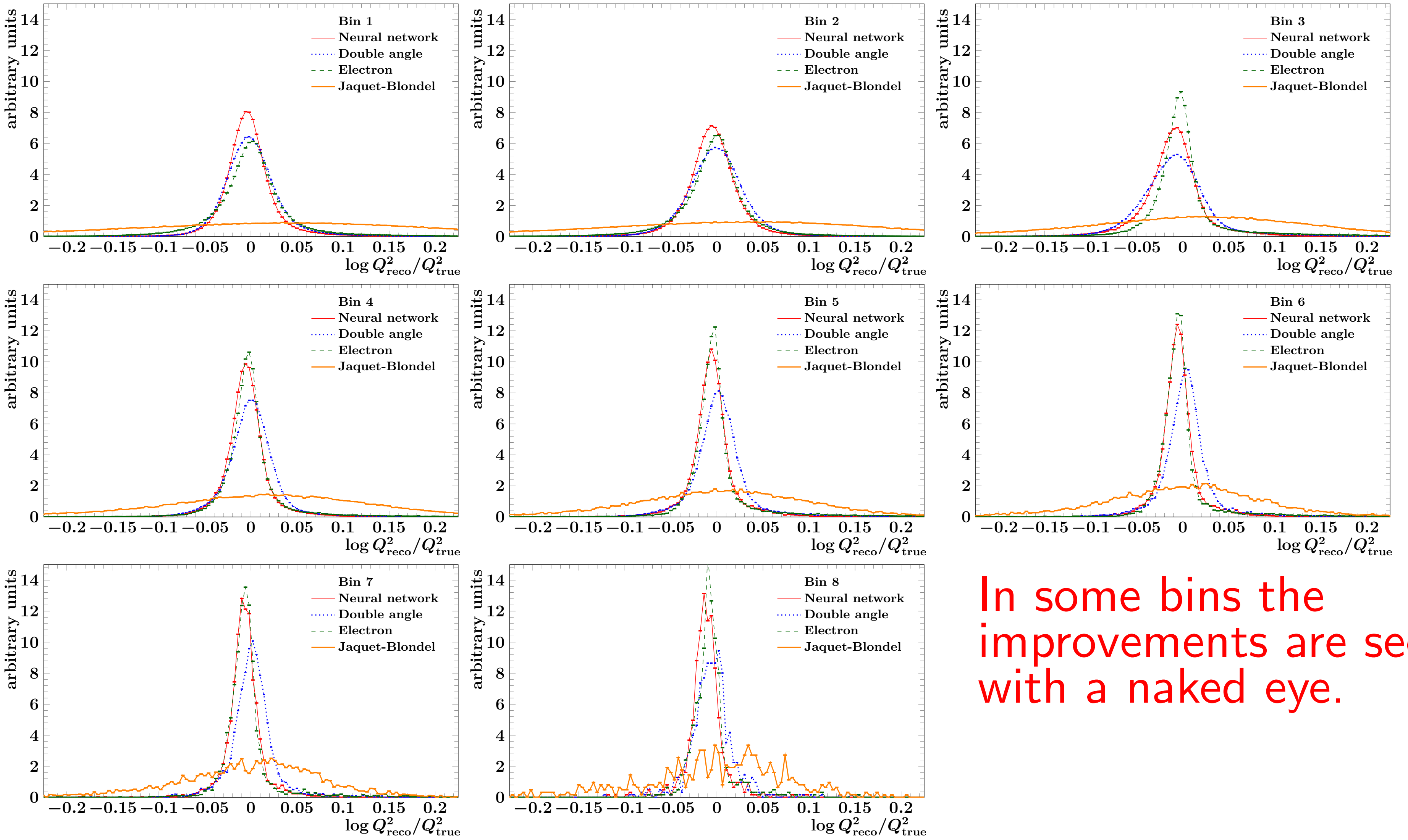
Kinematic phase space at the ZEUS experiment:



Bin	Events	Resolution of $\log x$		Resolution of $\log Q^2/1GeV^2$	
1	301780	**NN: 0.070**	EL: 0.083	**NN: 0.035**	EL: 0.035
		JB: 0.180	DA: 0.103	JB: 0.203	DA: 0.062
2	350530	**NN: 0.069**	EL: 0.082	**NN: 0.040**	EL: 0.043
		JB: 0.167	DA: 0.096	JB: 0.192	DA: 0.064
3	138456	**NN: 0.098**	EL: 0.130	NN: 0.055	**EL: 0.053**
		JB: 0.138	DA: 0.100	JB: 0.150	DA: 0.077
4	74844	**NN: 0.067**	EL: 0.084	**NN: 0.044**	EL: 0.046
		JB: 0.117	DA: 0.077	JB: 0.138	DA: 0.063
5	31043	**NN: 0.064**	EL: 0.091	**NN: 0.036**	EL: 0.041
		JB: 0.102	DA: 0.073	JB: 0.117	DA: 0.053
6	11475	**NN: 0.053**	EL: 0.079	**NN: 0.033**	EL: 0.036
		JB: 0.083	DA: 0.061	JB: 0.100	DA: 0.045
7	3454	**NN: 0.050**	EL: 0.069	**NN: 0.036**	EL: 0.038
		JB: 0.074	DA: 0.055	JB: 0.093	DA: 0.042
8	624	**NN: 0.036**	EL: 0.055	**NN: 0.033**	EL: 0.037
		JB: 0.067	DA: 0.045	JB: 0.095	DA: 0.041

Results for Q^2

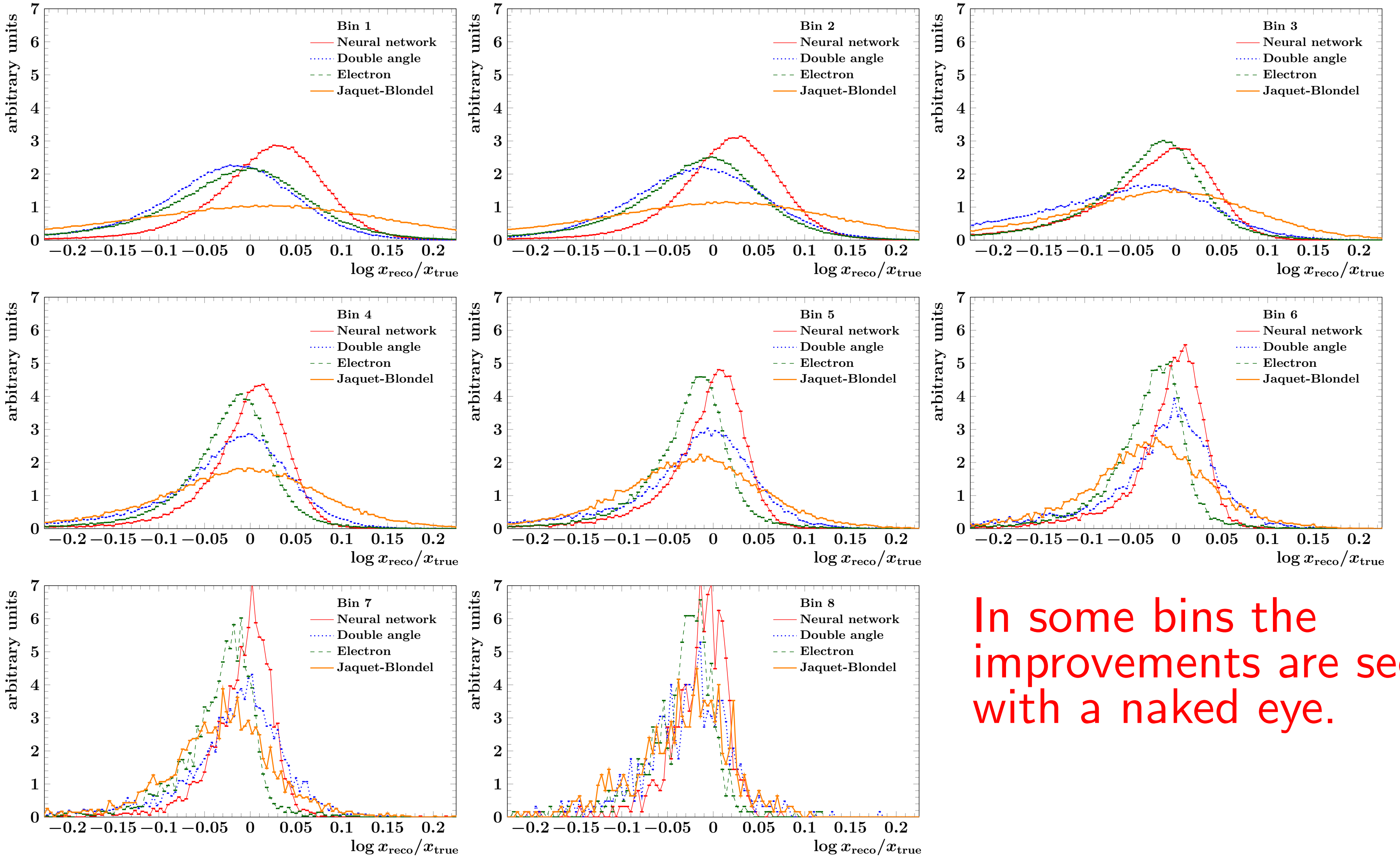
Slide from Abdullah Farhat (ODU)



In some bins the improvements are seen with a naked eye.

Results for *x*

Slide from Abdullah Farhat (ODU)



In some bins the improvements are seen with a naked eye.