CLAS12 data preservation

Harut Avakian *)

"CLAS12 collaboration meeting, July 24 , 2020"

- Preserving data: scope, possible users
- Format of the data
- Preservation forms and goals
- Scope of possible applications
- Preservation content and policies
- CLAS6 example (N. Tyler)
- Summary

TF on preservation

*) N. Baltzel, G. Gavalian, V. Mokeev, M. Ungaro, A. Vossen

Data Preservation TF

Charge

- Identify methodologies to guarantee data preservation
- Asses existing technologies finding the most suited for preservation data related to perturbative and non-perturbative physics
- Define a work plan to test the proposed solutions with a time chart and milestones for a study-case
  1. identification
  2. implementation and test
  3. generalization to the full set
- Estimate costs and identify resources needed for each option
- Evaluate synergies with other projects at the lab providing a list of shared resources and common goals

LHC originally didn't foresee funding for preservation

- Reproducible analysis provides an immediate scientific benefits (data mining,…) !
- MC preservations plays important role in overall preservation/reproduction chain
- Closer integration with experiment data management should reduce duplication between open data and experiment resources
- There is a significant overlap between software development and preservation

Requirements from HEP community

https://indico.cern.ch/event/444264/contributions/1950398/attachments/1167572/1775833/DPHEP-status_report-16dec.pdf

DPHEP (data preservation in HEP) Collaboration:

it is never early to consider data preservation: early planning is likely to result in cost savings that may be significant. Furthermore, resources (and budget) beyond the data-taking lifetime of the projects must be foreseen from the beginning.

all **archived data** – should be easily **findable** and fully **usable** by the **designated communities** with clear (Open) access policies and possibilities to annotate further statement from the US Office of Science

All proposals submitted to the Office of Science (after 1 October 2014) for research funding must include a Data Management Plan (DMP) that addresses the following requirements:

- ***DMPs should describe whether and how data generated in the course of the proposed research will be shared and preserved.***

  If the plan is not to share and/or preserve certain data, then the plan must explain the basis of the decision

  At a minimum, DMPs must describe how data sharing and preservation will enable validation of results, or how results could be validated if data are not shared or preserved

Analysis Preservation Efforts

Broadly there are two themes in Analysis Preservation.

D.Duelmann

"The Museum"

long-term, descriptive, archival, historical record of scientific activity

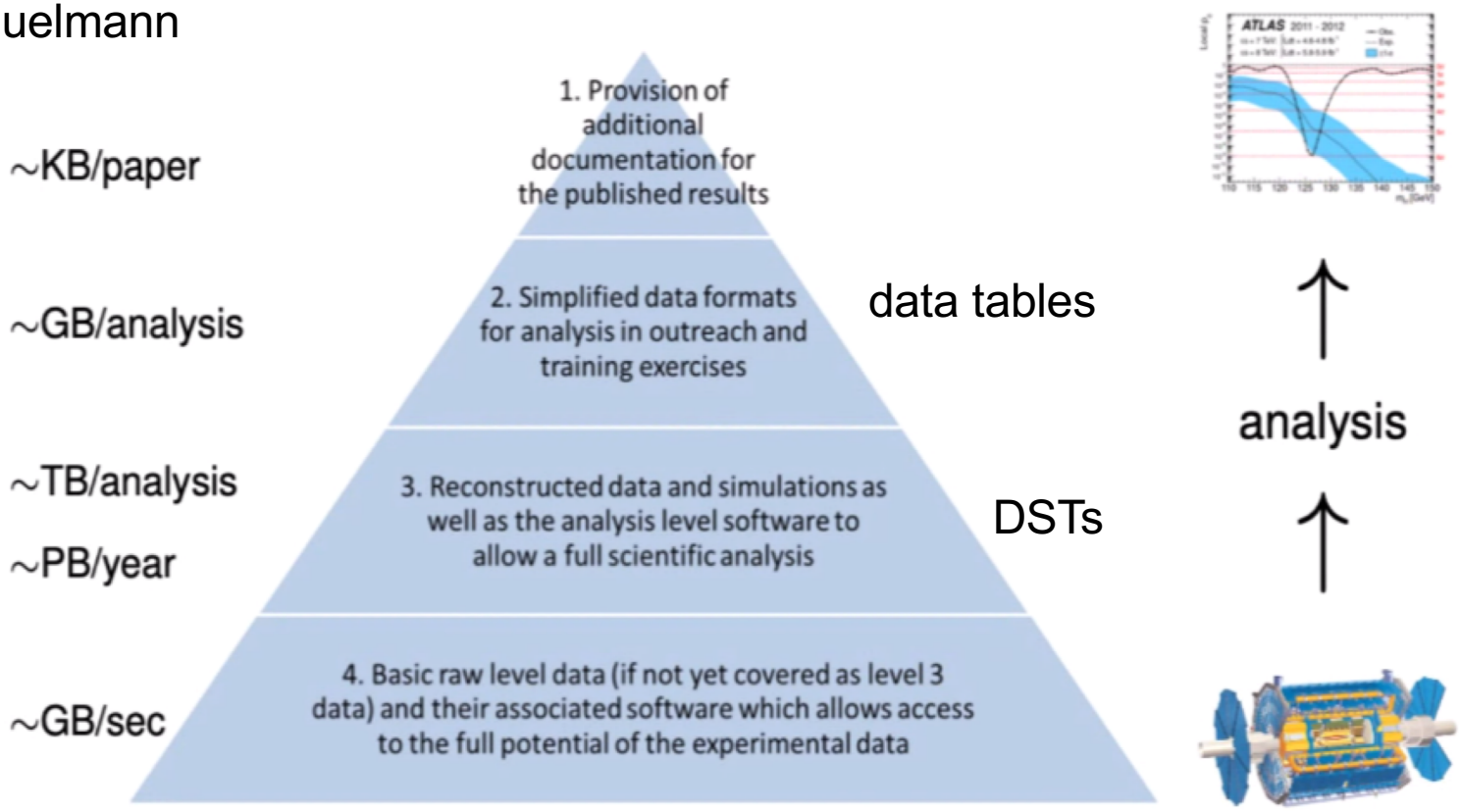
"The Hangar"

short-term, actionable, re-usable, deployable analysis implementation

9

Data Abstraction Hierarchy

D.Duelmann



Four data levels for capture, preservation and opening

adapted from slide by Tibor Simko

At which level we share the data (level 3?)

Lower the level, more relevant is coordination with software development

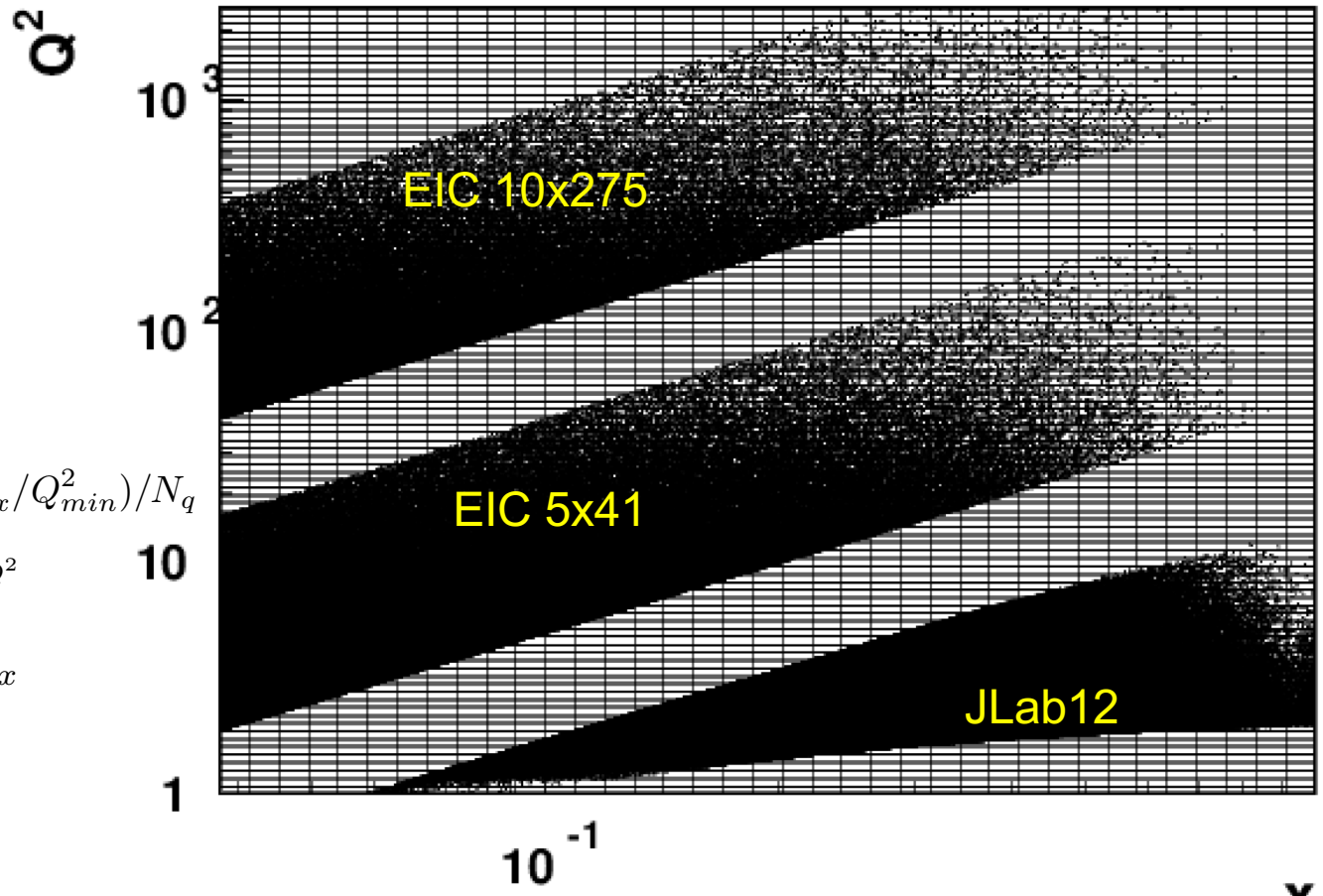
Requirements

- Identify the possible processes of interest
  - 1) SIDIS
  - 2) DVMP
  - 3)……..
- Identify the possible user:
  - A) CLAS12 collaboration
  - B) JLab
  - C) Theorists in collaboration with CLAS12
  - All
- Identify format for data
  - Data sample
  - Format to store
  - Tools to retrieve and analyze
- Defining clear policies for preservation

Reconstructed Data table: details

The same binning covers JLab and EIC

y>0.05



$$\Delta Q^2 = ln(Q^2_{max}/Q^2_{min})/N_q$$

$$Q^2_i = Q^2_{min} e^{i\Delta Q^2}$$

$$x_i = x_{min} e^{i\Delta x}$$

Resolutions defining the minimum sizes of bins are different for different experiments, but smaller bins will allow to sum over bins (ex. 5x5 bins in x&Q^2 to one to compare with EIC)

Std output: Reconstructed Data

"MC-sets": "Set1,set2" (Sets of reconstructed MCs used to define uncertainties for acc,RC, and kinematics) — MC

"Beam Energy": 10.600
"lepton-polarization": "0"
"nucleon-polarization": "0"
"observable":"e'X Counts"
"lepton": "e-"
"integrated Lumi"," 2fb^{-1}"
"D_integrated Lumi"," 0.05fb^{-1}"
"Detector":"CLAS12"

Detector Configuration

"Configuration":"Inbending" (Outbending, Torus 75%,.....)
"Reconstruction":"Rec-software v.4.1"
"Calibration":"Calib-dbase RGA-production"
"Fiducial region":"Tight" (kinematical cuts like "theta_min":"10 degrees", "W_min":"2 GeV",.. can be separated

Reconstruction

Data

"variables":["N_{Rec}","$\langle x \rangle_{Rec}$","$\langle y \rangle_{Rec}$","$\langle Q^2 \rangle_{Rec}$","$\langle W \rangle_{Rec}$",'N_e+',"RC","D_RC", "Acc","D_Acc","D_<X>","D_<y>","D_<Q2>","D_Sys"

"axis":[
{"name":"x","bins": 30,"min": 0.06, "max": 0.96,"scale":"lin","description":"x_bj"}
{"name":"y","bins": 30,"min": 0.2,"max": 0.95,"scale":"lin","description":"y"}
],

MC

.....

2	20	4717.0	0.10488	0.68755	1.43442	3.62238	20.0	1.020	0.05	0.819	0.005	0.001	0.003	0.005,...
2	21	4562.0	0.10427	0.71249	1.47774	3.68436	15.0	1.031	0.05	0.875	0.005	0.001	0.003	0.005,...

Data tables keep info on raw counts and averages of kinematical variables col 1-5 in "fiducial region" (defined in header) and additional info from MC to account for uncertainties from acceptance, RC, definitions of <> values

Providing the data set

Providing input for AI or Mesh type reproduction of sets

Transformed Generative Adversarial Network
Alanazi et al, http://arxiv.org/abs/arXiv:2001.11103

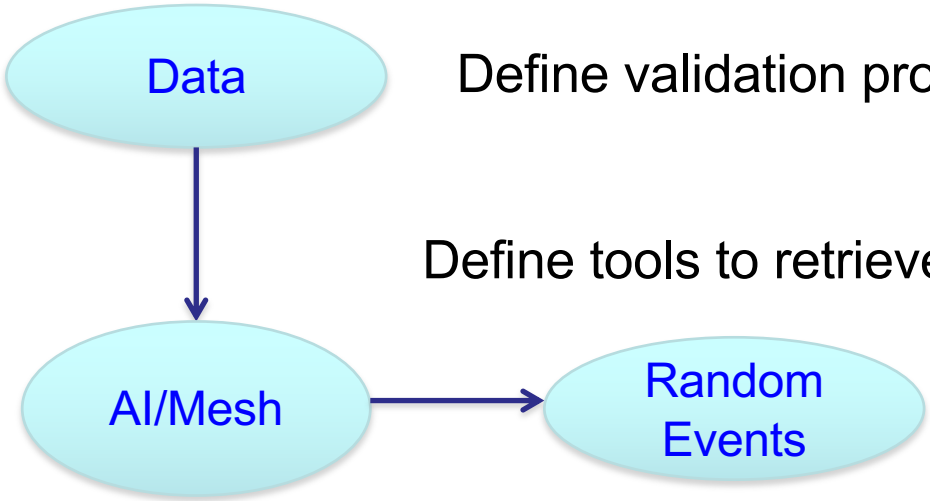
Nikos Chrisochoides et al

Synthetic Gaussian Data

- Input distribution size : 1,000,000 cells
- Uniform case: 30,949 triangles
- Adapted case: 3,788 triangles

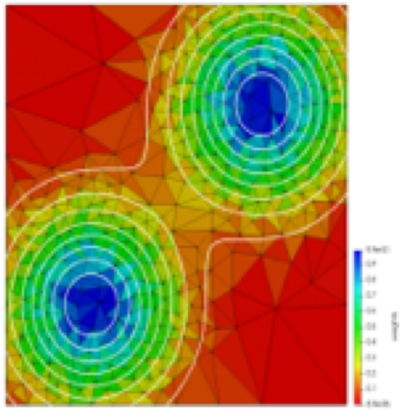
Define the input set in all details (ex. Model of RC)

Identify format for data for a given scope



Define validation procedures

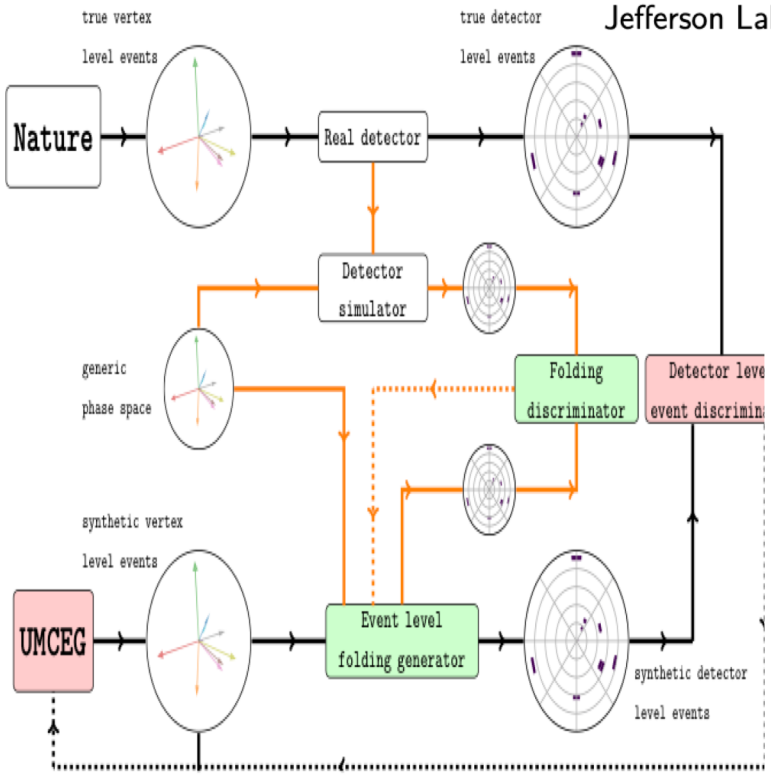
Define tools to retrieve and analyze

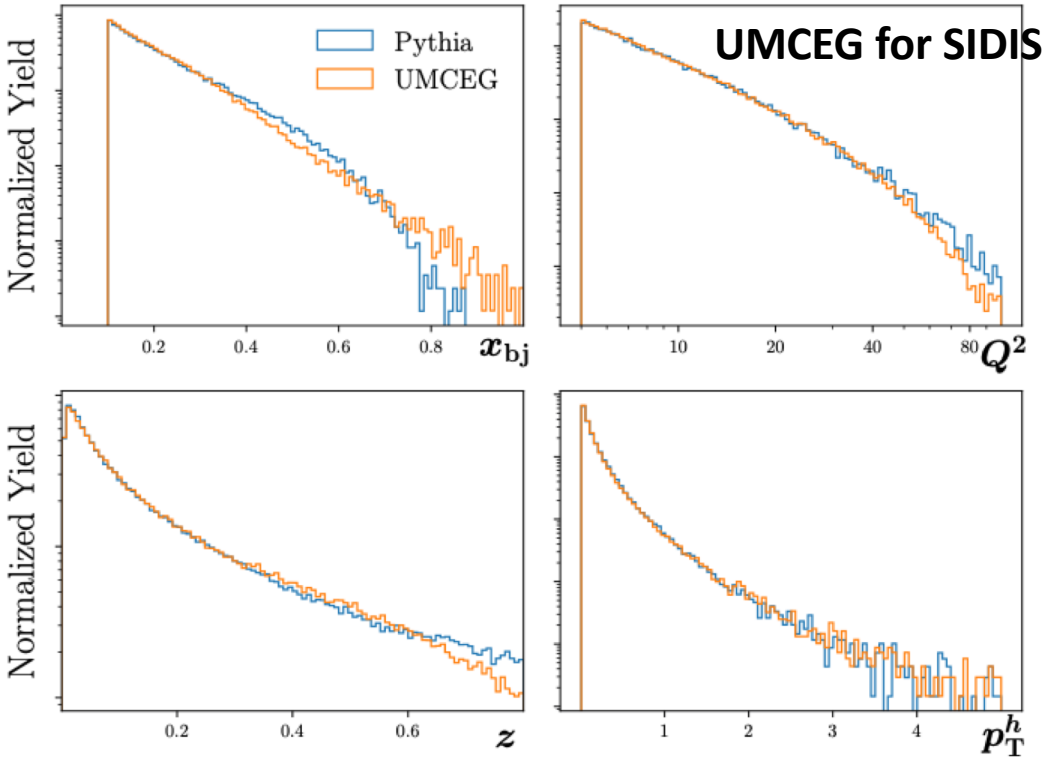


https://crtc.cs.odu.edu/CNF_Example_Meshes

AI production based on data/MC input

Nobuo Sato
Jefferson Lab



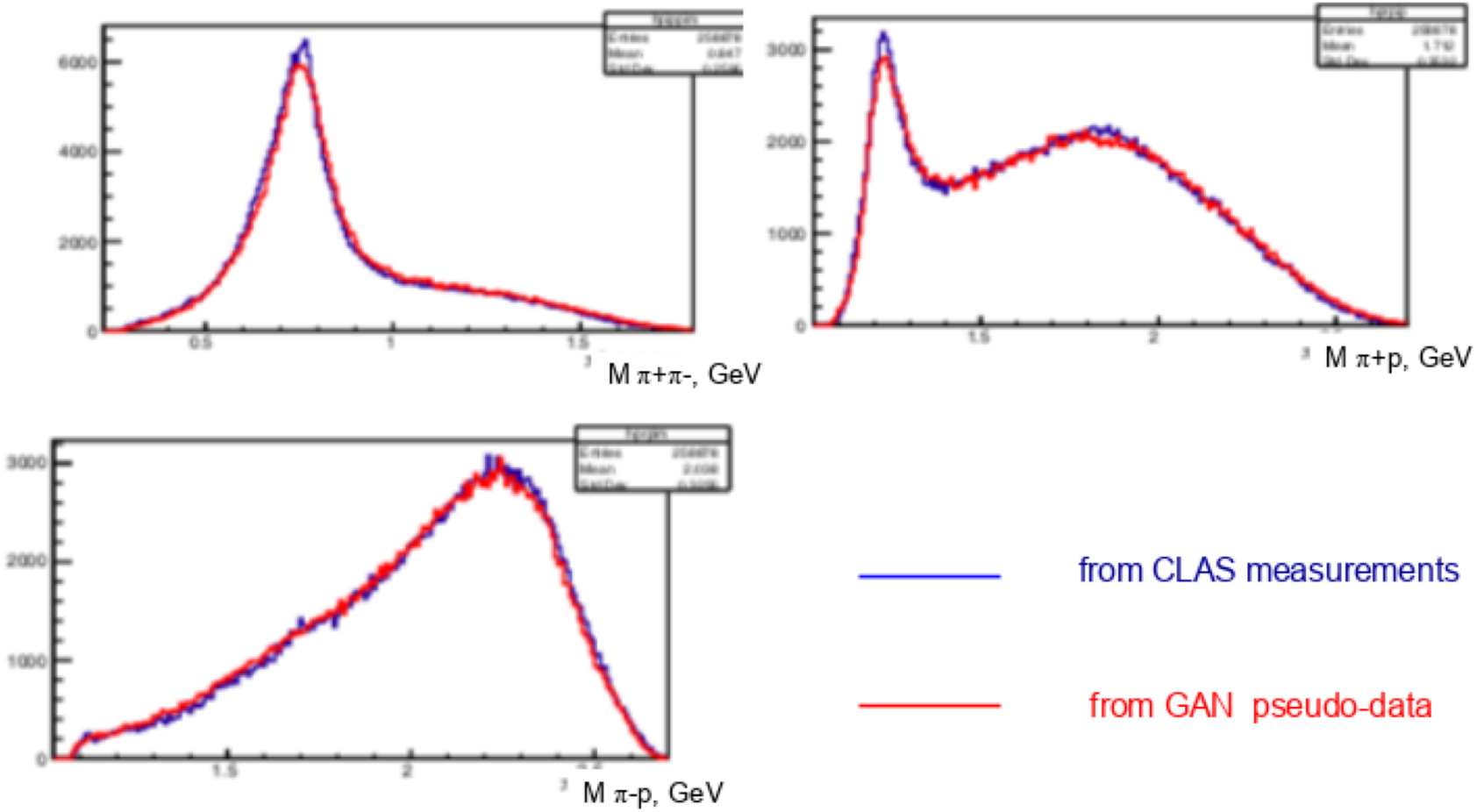


UMCEG for SIDIS

- The colored boxes are build using NNs
- Each pair of colored boxes corresponds to a GAN

The MC as well as real data can be reproduced
Can produce x10 more data from the input (good for multidimensional analysis
How we can check/validate the AI data, and what we can do with this data?

Comparison between nine 1D-distributions for $\pi^+\pi^-p$ events from the CLAS experimental data and from ML pseudo-data



The pseudo-data well reproduce the experimental results from CLAS on nine 1d distributions, M.Battaglieri et al., Phys. Rev. D80, 072005 (2009)

Preserving and Sharing the CLAS12 data

Two ways to preserve:

1) Preserve the full set of events needed to reproduce the input for any given set of observables (DIS,SIDIS,DVMP,….)
Pros: Flexible, can perform analysis not foreseen in past (good for large acceptance detectors) change kinematical phase space by cuts
Cons: Require Maintenance of full set of info needed for handling (including generators, simulation, reconstruction software analysis software, calibration constants,….).
Technology: Hardware virtualization (such as VMware and VirtualBox) and container virtualization (such as Docker)

2) Creating model for the data for specific final states of interest and produce/reproduce the set with AI, mesh,grids…
Will take input from experiment (ex. based on grids for different processes)

- Pros: Fast, dedicated
- Cons: Only for a given set, should be redone with new cuts
- May be challenging for multiparticle/multidimensional processes

1) & 2) do not exclude each other, their combination will be most efficient

All **archived data** – should be easily **findable** and fully **usable** by the **designated communities**

Defining requirements

Any option for preservation will require strict rules and full documentation to provide validation and reproducibility

All software used in the data chain should be available in git and well documented

- Generators used in acceptance and RC (OSG friendly)
- gemc version
- Calibration software version
- Calibration database (sqlite)
- Reconstruction version
- Set of PID and fiducial cuts
- Momentum and other corrections applied
- Physics and detector backgrounds (ex. high lumi backgrounds)
- Production of the full list of systematic errors (cross check important)
- ……

Release for final publication should be allowed only after fulfilling all above requirement

M. Ungaro

Preservation requirements

OSG Containers

- Produced by (automated) build from github changes to dockerfile
- Propagated automatically to CVMFS by OSG:
  /cvmfs/singularity.opensciencegrid.org

- xrootd access to background files @JLAB for merging
- cvmfs access to CLAS12 software:
  - tagged CCDB SQL file
  - (magnetic field maps)

- conform all detectors to read DIGITIZATION_TIMESTAMP in the digitization
- self contained environment and configurations
- automatic (empty) tests

Production:

OSG Production Simulations through the portal

Development:

For tests of the new software versions

M. Ungaro

Preservation requirements

OSG Software Content

Tagged Installed Software:

> CCDB version: 1.07.00
> CLHEP version: 2.4.1.3
> GEANT4 version: 4.10.06.p02
> QT using system installation
> XERCESC version: 3.2.3
> EVIO version: 5.1
> MLIBRARY version: 1.4
> SCONS version: 1.9
> CLAS12 Tag: 4.4.0

Simulation Software Moving to CVMFS

Moving to 100% "module" environment

(currently: mix of scripts and module)

CVMFS Tagged Software:

> Coatjava
> Java

Plan: support multiple version of various libraries through CVMFS installations and module environment.

Preservation requirements: generators

- An executable with the same name as the github repository name, installed at the top level dir
- If libraries are needed, they should be put inside a lib directory, at the top level dir
- The generator output file name must be the same name as the exectuable + ".dat". For example, the output of clasdis must be clasdis.dat
- To specify the number of events, the option "--trig" must be used
- If necessary, an environment variable (name in its README) where the executable will look for data
- The optional argument --docker will be added by default to all executable. This option can be ignored or used by the executable to set conditions to run on the OSG container

If you are the maintainer of a package and made changes that you want to include here, send emails to ungaro@jlab.org, baltzell@jlab.org (Mauri or Nathan) requesting the update.

M. Ungaro

OSG Generator Requirements

https://github.com/JeffersonLab/clas12-mcgen

List of Generators

name	summary description	maintainer	executable name	output ok
clasdis	clas SIDIS MC based on PEPSI LUND MC	Harut Avakian	✅	✅
claspyth	SIDIS full event generator based on PYTHIA	Harut Avakian	✅	🔴
dvcsgen	DVCS/pi0/eta generator based on GPD and PDF parameterizations	Harut Avakian	✅	✅
genKYandOnePion	KY, pi0P and pi+N	Valerii Klimenko	✅	✅
inclusive-dis-rad	generates inclusive electron and optionally radiative photon using PDFs	Harut Avakian	✅	🔴
tcsgen	Timelike Compton Scattering	Rafayel Paremuzyan	🔴	🔴
jpsigen	J/Psi	Rafayel Paremuzyan	✅	🔴

- added at submodules to jeffersonlab/clas12-mcgen
- frozen version of each submodule in container
- requirements to access generators from OSG portal

CLAS6 data preservation

- Data mining of CLAS6 data demonstrated the relevance and great benefits of preservation
- For some processes the quality of CLAS6 was exceptional, and new publications will certainly , require some clas6 software chain to run

A container with a full chain of the clas6 data analysis has been prepared for the community by Nick Tyler (detailed presentation in backup slides)

- Project and documentation available on GitHub https://github.com/tylern4/clas6
- Available for running simulations on the farm with Singularity

- Also running on other farm/batch systems

Singularity image available on farm [/work/clas/clase1/tylern/clas6.img] (~416M)

CLAS12 containers ~2Gb

Summary

- Define the policies for preserving the CLAS12 data for internal benefits and for external use (”open data”)
- Finalize the list of items required for git and procedure for validation of docker containers, for clas12, and also clas6 containers, including the full analysis chain
- Define the scope of the data to be preserved, and possible user base
- Define the metadata input (YAML, JSON,..) to be used in generators or making compatible with existing standards (also phenomenology groups).

- Test clas6 container by Nick Tyler
- Test reproducing RGA-pass1 analysis (starting with DIS electron, with extension to inclusive and 2 pion final states
- Define criteria for validity and consistency of the real and virtual MCs based on new advanced approaches (AI, mesh,..) comparing with existing generators and data
- Check the applicability of concepts and tools developed by HEP community

Support slides…

CLAS6 SOFTWARE IN DOCKER/SINGULARITY

Nick Tyler 07/2020

N.Tyler

CLAS DOCKER/SINGULARITY

- Project and documentation available on GitHub
  https://github.com/tylern4/clas6
- CLAS software is from the clas svn trunk, full chain is built and running
  https://jlabsvn.jlab.org/svnroot/clas/trunk
- Available for running simulations on the farm with Singularity
  - An example and wapper scripts available on the GitHub page
- Also running on other farm/batch systems but non-trivial at this point
  [UofSC, OpenScienceGrid, CHTC]

CLAS DOCKER/SINGULARITY

- Project and documentation available on GitHub
  https://github.com/tylern4/clas6
- CLAS software is from the clas svn trunk, full chain is built and running
  https://jlabsvn.jlab.org/svnroot/clas/trunk
- Available for running simulations on the farm with Singularity
  - An example and wapper scripts available on the GitHub page
- Also running on other farm/batch systems but non-trivial at this point
  [UofSC, OpenScienceGrid, CHTC]

N.Tyler

LAYOUT OF SIMULATION SCRIPT

Example script of a full chain which runs on the farm

Sources functions to run singularity image →

Run function to copy in configuration files →

Runs generator with wrapper to run in singularity →

These functions are wrappers to in functions in singularity →

Convert/copy outputs from scratch as needed →

```
# Make a few environment variables for directories
export JOB_DIR=${PWD}
export SCRATCH=/scratch/${USER}/${SLURM_JOB_ID}/${SLURM_ARRAY_TASK_ID}

#######========== Load Singularity to Farm Node ==========#######
source clas6_function_singularity.sh

#######========== Setup Scartch Folders ==========#######
cd ${SCRATCH}

#######========== Copy in configuration files =====#######
cp-to-job ${JOB_DIR}/aao_rad.inp
cp-to-job ${JOB_DIR}/gsim.inp
cp-to-job ${JOB_DIR}/user_ana.tcl

#######========== Run Generator ==========#######
run-singularity-clas6 aao_rad < aao_rad.inp

#######========== Run gsim ==========#######
gsim_bat -nomcdata -ffread gsim.inp -mcin aao_rad.evt -bosout gsim.bos

#######========== Run gpp ==========#######
gpp -ouncooked.bos -a2.35 -b2.35 -c2.35 -f0.97 -P0x1b -R23500 gsim.bos

#######========== Run user_ana ==========#######
user_ana -t user_ana.tcl


#######========== Run h10maker ==========#######
run-singularity-clas6 h10maker -rpm cooked.bos all.root
#######========== Copy all the files to Work for output ==========#######
cp -r ${SCRATCH}/all.root ${JOB_DIR}/outputs/sim_${SLURM_JOB_ID}_${SLURM_ARRAY_TASK_ID}.root
```

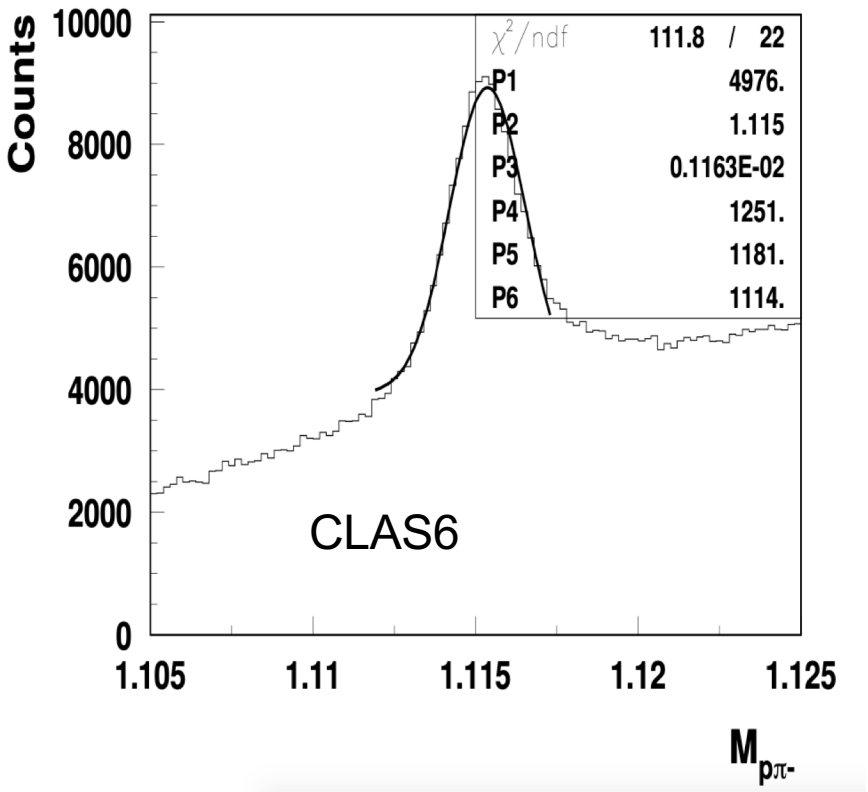
N.Tyler

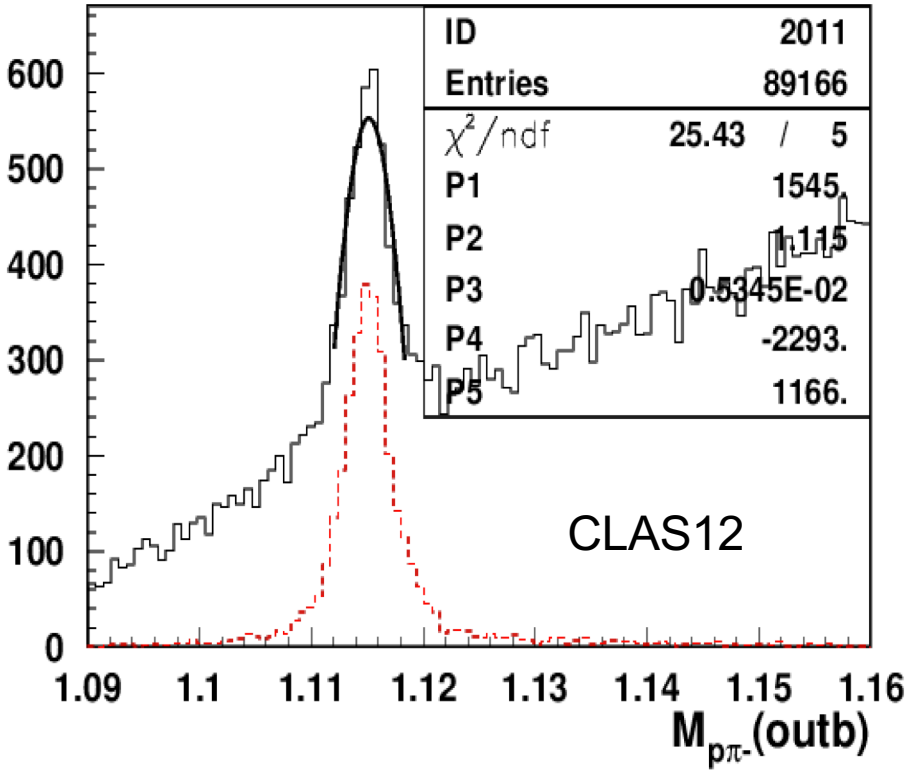
CURRENT COMPLICATIONS

- Need to have a copy of `/group/clas/parms` for all the software/configurations to work (~33GB)
  - On farm it is linked into the container but offsite can be a problem
- Need to have access to `clasdb.jlab.org`
  - Not accessible offsite but I've made copies which work where I have simulations running
- Not all generators/configurations are fully tested
  - Fully working and tested on e1 experiments [e1d/e1f/e1-6] with aao generators

Example with CLAS6

Data mining of CLAS6 data demonstrated the relevance of preserving it
For some processes the quality of CLAS6 was exceptional





Resolutions x5 better with CLAS6

M. Ungaro

Preservation requirements

OSG Software Versioning

Two solutions pros and cons

All in container

PROS	CONS
Only docker needed	Only one version of each library / container due to size limitations
Read only file system (reproducibility)	need one container for each version
No internet needed after image is downloaded	

Software from CVMFS

PROS	CONS
Supports multiple versions of software, i.e. coatjava	cvmfs and internet needed
lightweight container	need to make sure disk will never change once "tagged"

Preservation key words & concepts

Paradigm of systematic reusing

Concept of open data, requires preservation of analysis

- Data may or may not be directly linked to specific analysis
- In reusing the data, main user is the internal one
- For reproducing the actual analysis, important to define criteria
- Re-use and reproducibility, reinterpretation interconnected
- Important storing/capturing software
- Relevance of common tools
- Continuous integration pipeline involve: git and github, docker containers, environment storage, software capture, maintaining and retaining infrastructure

reana

REANA is a reusable and reproducible research data analysis platform.

Structure your analysis inputs, code, environments, workflows and run your analysis on remote containerised compute clouds.

Scalable

Support for remote containerised clouds.

kubernetes

Reusable

Containerise once, reuse elsewhere. Cloud-native.

Preservation: internal and external, may be deploying outside of CERN

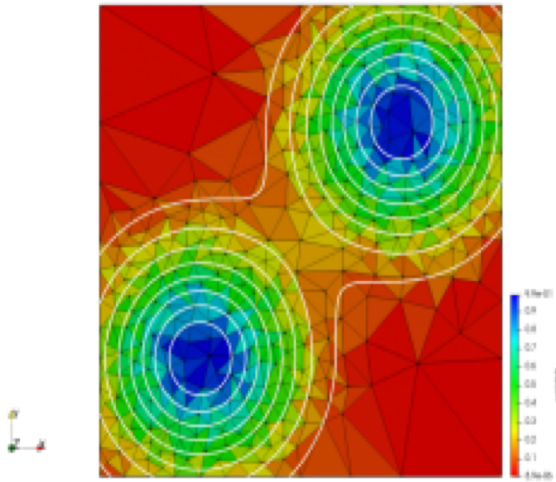
CERN preservation tools:

ReANA framework on different centers with HPC

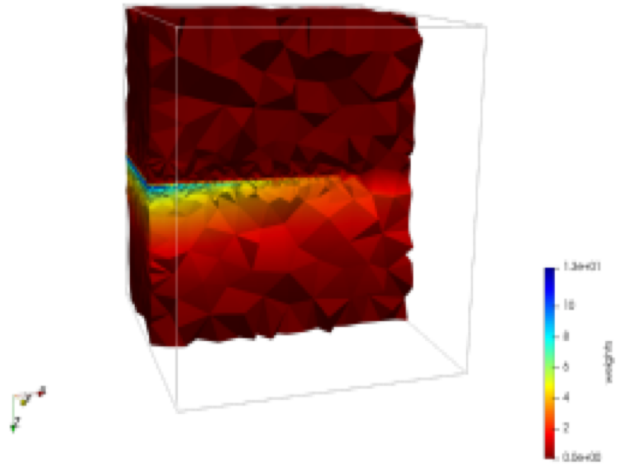
CAP (CERN Analysis Preservation) "preserve analysis" data mining

Data Preservation

- **Data Preservation**
  - Create a model of the data for specific final state.
  - Model the phase space distribution of the data.
  - Ability to generate random events according to the generation.
- **What can be achieved**
  - Analyze data again with different binning and cuts
  - Extract observables with different method
  - Ability to compare with theory
- **Method:**
  - Using adaptive mesh generation used in CNF project
  - 3D adaptive generation and MC generation currently works
  - 4D-5D coming soon (4D within a year, work is ongoing on 5D)



DVCS Cross Section
1M points Grid,
67K Tetrahedron



19

LHC tools

Kubernetes is a container orchestration system for **Docker** to coordinate clusters of nodes at scale in production in an efficient manner