Benchmark User Experience

.. and UX for ePIC in general

Torri Jeske | roark@jlab.org

Jefferson Lab

ePIC

Background

I'm a trained physicist. I stumbled upon this work as part of my role in EPSCI. As a physicist, I primarily spent time writing analysis software (ROOT macros) that only I would ever use.

I now spend a significant amount of time writing software for other people to use and *working with them* to ensure our software is usable, accessible, reliable, and solves our users problems.

Importantly -- I'm not an expert in UX and am still learning everyday.

Outline

What is User Experience?

Why should we care? How can we improve?

Validation Working Group

People, charges, lots of problems

How can we improve?

Where do we go from here?

User experience (UX)

how a user interacts with and experiences a product, system, or service. It includes a person’s perceptions of utility, ease of use, and efficiency.

User Experience for Software

How a person feels and performs while trying to accomplish a task. It's not just how it looks (user interface, or UI); it's the sum of the workflow: finding the tool, installing, learning, running, debugging, and *sharing* it.

Ignoring UX directly contributes to a feedback loop where only few people can do the work.

Higher barrier to entry → fewer users → knowledge concentrates into a tiny core → burnout → slower fixes by fewer people → rinse and repeat

Empathy

You *must* be able to put yourself in someone else's shoes and try to understand how they will use your software.

How does this relate to ePIC software?

Validation Working Group

Responsible for the validation of the simulations via a suite of detector and physics performance plots
Development of autonomous checks and verification (CI)
Co-convener with Dmitry Kalinkin, I had at a maximum of ~15-20% time dedicated to these efforts.

Priorities for FY23/24

- Implement and document simulation production strategy
- ***Develop and maintain a collection of plots that showcase the performance of the ePIC detector, its physics reach, and enable comparisons to a baseline or previous simulation campaigns***
- Drive the development of unit tests for the ePIC software, together with the Development WGs

Priorities for FY25/26

- Extending workflow automation
- Tools for comparisons

Validation Working Group

Responsible for the validation of the simulations via a suite of detector and physics performance plotsDevelopment of autonomous checks and verification (CI)
Co-convener with Dmitry Kalinkin

Priorities for FY23/24

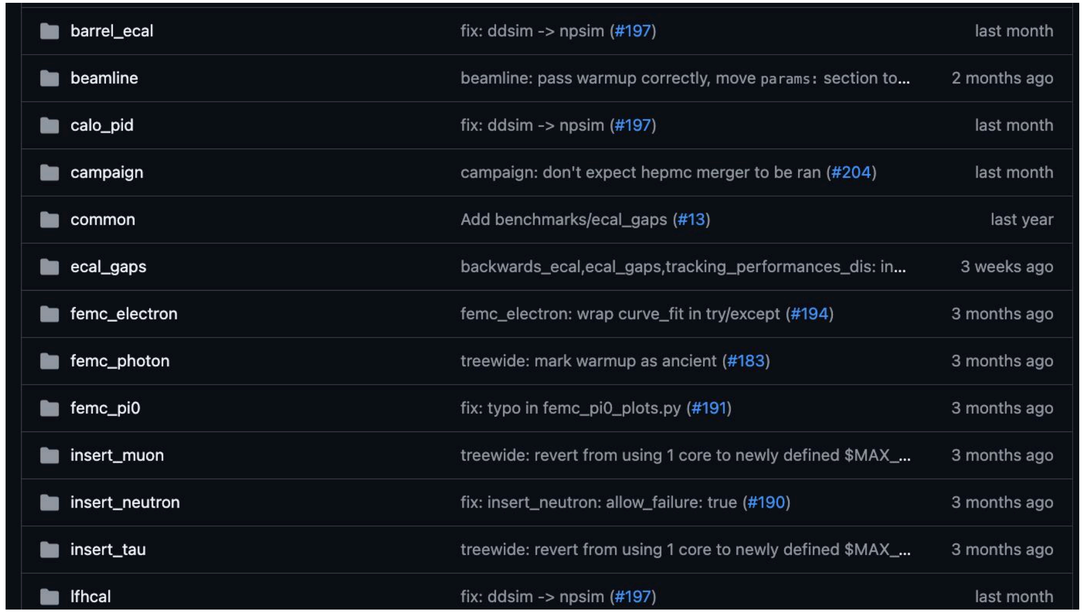
- Implement and document simulation production strategy
- ***Develop and maintain a collection of plots that showcase the performance of the ePIC detector, its physics reach, and enable comparisons to a baseline or previous simulation campaigns***
- Drive the development of unit tests for the ePIC software, together with the Development WGs

Priorities for FY25/26

- Extending workflow automation
- Tools for comparisons

***Physics and Detector working group members would actually develop the plots.**

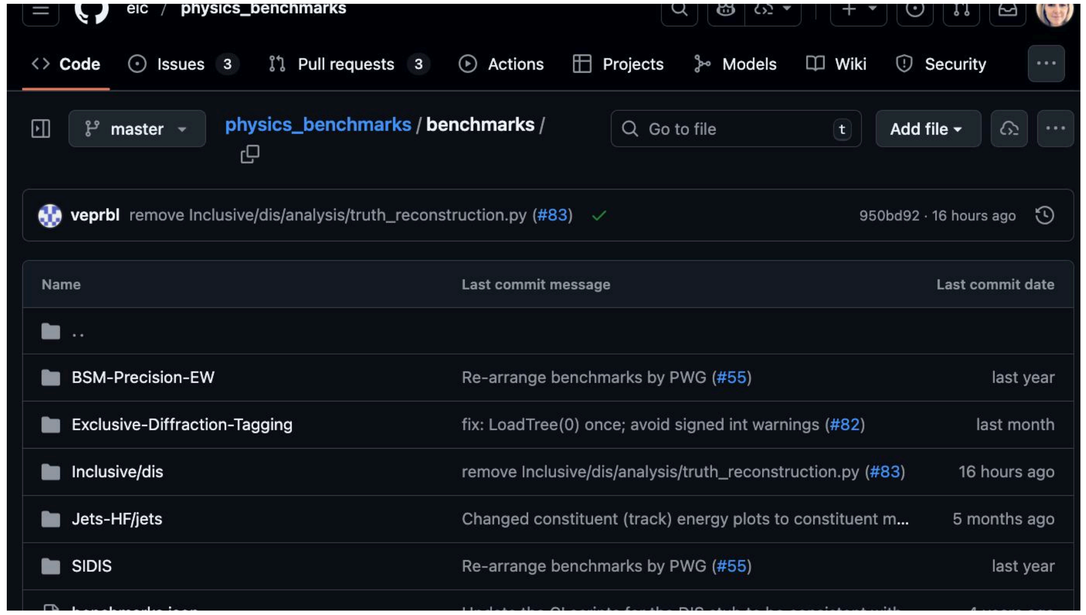
Benchmarks - as of yesterday



barrel_ecal	fix: ddsim -> npsim (#197)	last month
beamline	beamline: pass warmup correctly, move `params:` section to...	2 months ago
calo_pid	fix: ddsim -> npsim (#197)	last month
campaign	campaign: don't expect hepmc merger to be ran (#204)	last month
common	Add benchmarks/ecal_gaps (#13)	last year
ecal_gaps	backwards_ecal,ecal_gaps,tracking_performances_dis: in...	3 weeks ago
femc_electron	femc_electron: wrap curve_fit in try/except (#194)	3 months ago
femc_photon	treewide: mark warmup as ancient (#183)	3 months ago
femc_pi0	fix: typo in femc_pi0_plots.py (#191)	3 months ago
insert_muon	treewide: revert from using 1 core to newly defined $MAX_...	3 months ago
insert_neutron	fix: insert_neutron: allow_failure: true (#190)	3 months ago
insert_tau	treewide: revert from using 1 core to newly defined $MAX_...	3 months ago
lfhcal	fix: ddsim -> npsim (#197)	last month

github.com/eic/detector_benchmarks

30 unique benchmarks for different detector components



github.com/eic/physics_benchmarks

Handful of different benchmarks based on different physics here.

Dmitry did the overwhelming majority of this work, including developing tutorials to explain how to make a benchmark.

Problems!

What plots?

“We don’t know what we want to look at yet”

Immediate Pushback

- We can’t display these images on publicly available page
- Pushback from the working groups, analysis coordinators, and even at the TIC level (according to DK)

Supplied benchmarks once..

- Most users developed using the simulation campaigns but on their local machine
- There is no effective enforcement policy to ensure that users push their analysis codes back to github
- We do not know what the end users use benchmarks for (if anything at all)

More problems..

Using images for CI/CD

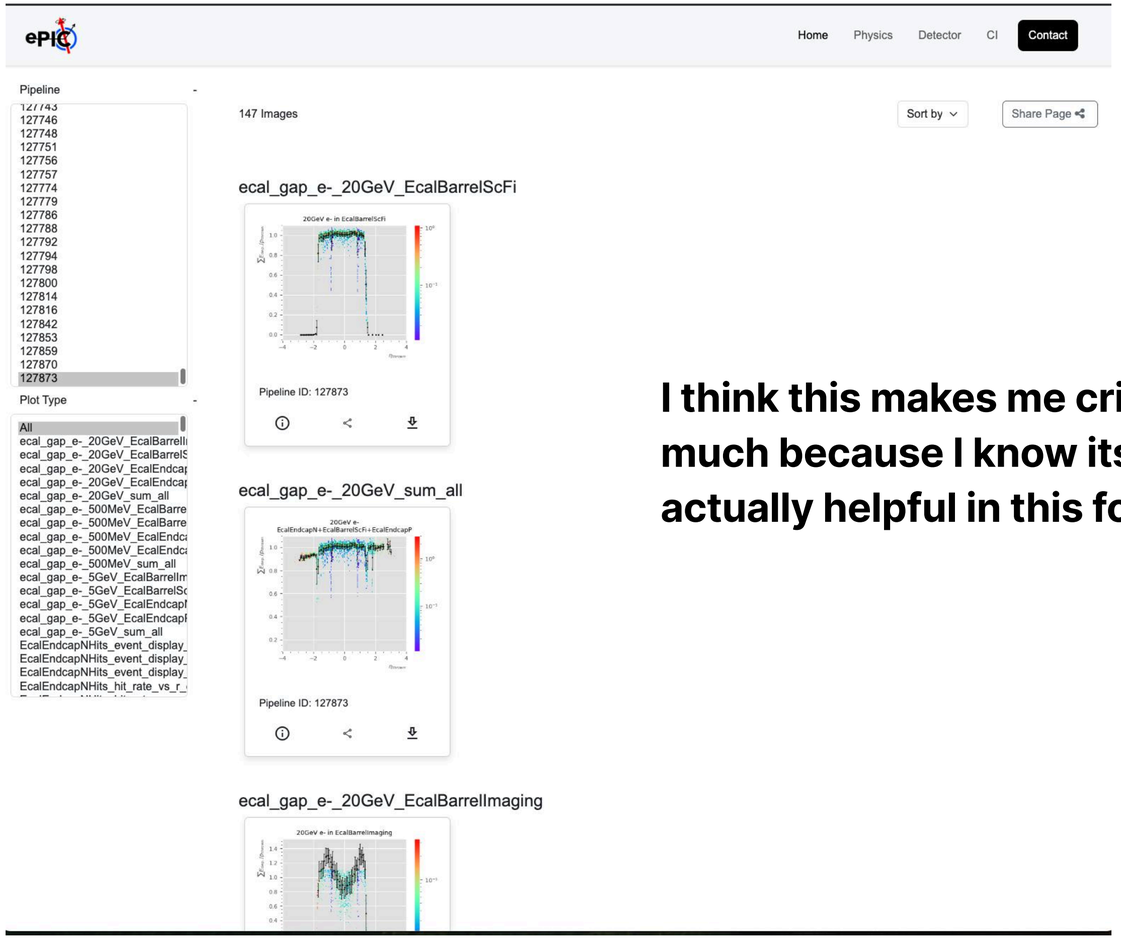
- Wouter and Dmitry seemed to really want image-based artifacts from the CI/CD system
  - To this day, I'm not sure I am convinced image-based artifacts (as opposed to unit tests) are effective.
- We worked together to set up a way to write the images to a specific jlab directory visible to a jlab hosted web server
- Our end users do not use this. They do not understand pipelines, jobs etc
- **There was and is still no good mapping that would guide a user to their benchmarks after they were submitted.**

Generating images from campaigns

- Markus really wanted these benchmark plots to be produced using the entire data set from the campaigns
  - There is still not widespread agreement for this.
- This wasn't possible to do via gitlab
- I wrote a script to submit jobs to the jlab farm that would run the jet benchmarks over the simulation campaigns.
  - This directly conflicted with what Wouter and Dmitry wanted (perhaps rightfully so), because I became a sole maintainer of that script.
  - "It wasn't open source", "We don't know what you're doing"

Image Browser (CI/CD)

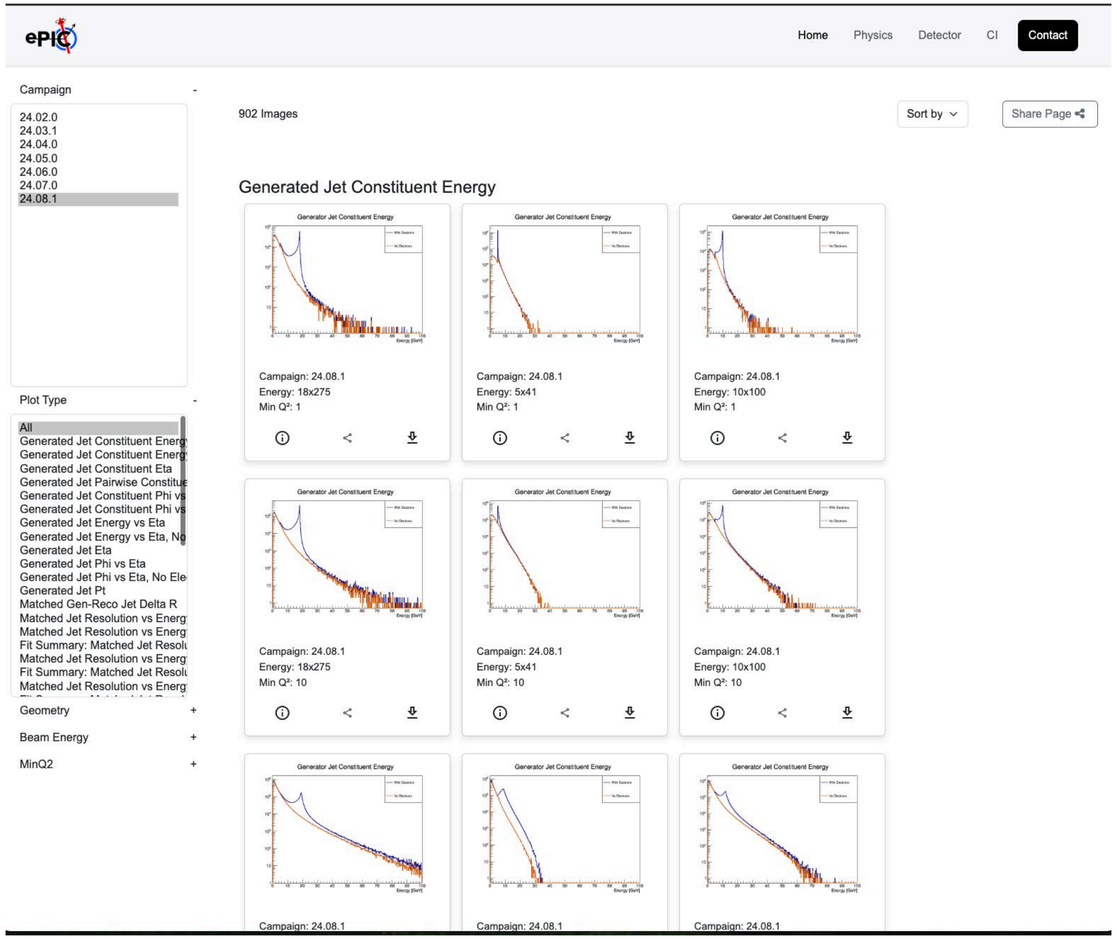
- It’s a simple UI with basic navigation to find images given an image name and (initially) a pipeline ID
- Issue 1: ”What’s a pipeline”
- Issue 2: Some people don’t want these public
- *A useful tool that doesn’t solve our end users problems.*
  - How can you be so sure, Torri?
  - The lack of scream tests when I stopped developing and Dmitry stopped sending image artifacts.



I think this makes me cringe so much because I know its not actually helpful in this form.

Image Browser (Campaigns)

- It’s a simple UI with basic navigation to find images given an image name and a campaign
- Some people still don’t want these images public
- *A useful tool that is really helpful - Brian Page*
  - Downside: The “back end” software clones repositories to get the latest versions of macros and updates the page with new images. Brian did not automatically push his updated macro to the repo. (Remember the lack of enforcement I mentioned earlier?)



Validation Working Group

Responsible for the validation of the simulations via a suite of detector and physics performance plotsDevelopment of autonomous checks and verification (CI)
Co-convener with Dmi

Priorities for F 25/26

- Implement and doc automation production strategy s
- ***Develop and maint that showcase the detector, its physics reach, and enable comparisons to a baseline or previous simulation campaigns***
- Drive the development of unit tests for the ePIC software, together with the Development WGs

validation working group was dissolved.

***Physics and Detector working group members would actually develop the plots.**

Okay, so where do we go from here?

Standing request for campaign-level images

- Markus wants Derek Anderson and myself to work together to produce these performance plots automatically when a new campaign is finished.
- Dmitry still wants something for the CI/CD system because this functionality is essentially not present from GitLab.
- **Reality: I have so little time.**

Dmitry and I see “eye to eye”!

- We got to meet in person, in my office, and we both agreed that there is no good way for any user (that isn’t him) to navigate finding benchmark images from the CI/CD system.
- We cannot develop any new features to make this easier without first solving this problem.

Summary

Failure to properly understand our users

We asked them for benchmarks they would not use, maintain, or bother to look at. Probably no one asked them about their own workflows.

Provided documentation for what *we* wanted

There was no incentive for our end users to not develop on their own machines.

Lack of engagement

Dmitry did a lot of work in terms of getting benchmarks, but there was no consistent communication between us and our users.

I personally don't think people understood the importance of validation at the same level they understood the importance of *production*.

Solutions?

Meet our users where they're at

We know people are hesitant to share their code, especially when its under rapid development.

Disincentivize local development

This is really hard because *everyone* does this at some level, especially for "quick studies".

Detail use cases for benchmarks

Dmitry uses benchmarks for tracking, to figure out if new material maps are good, or ACTS version updates are good.

Obviously, Dmitry is only a single user who is not representative of the majority of those in ePIC.

Coherent support across leadership

This feels obvious, but Markus was really our biggest and seemingly only supporter.