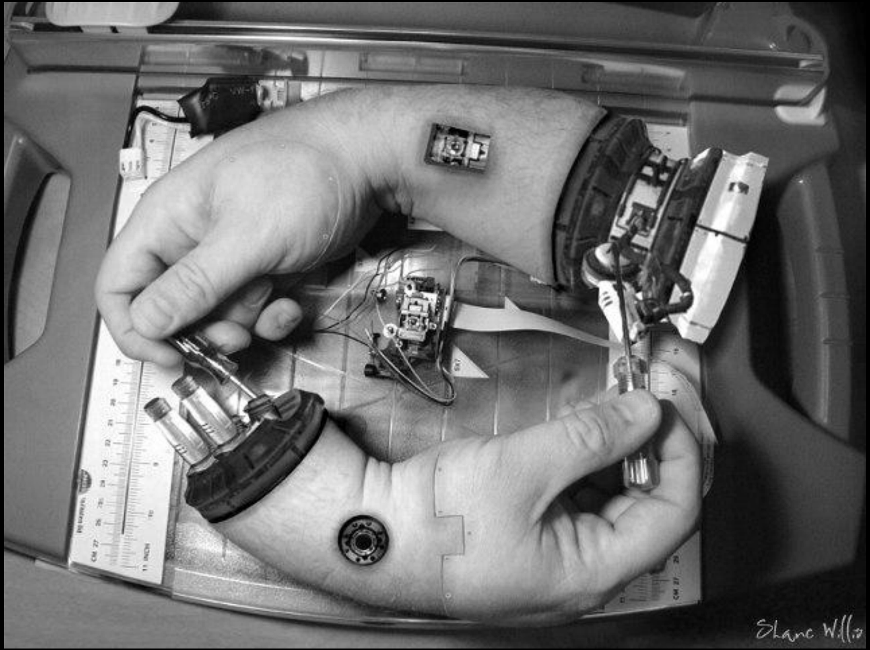
AI Working Group



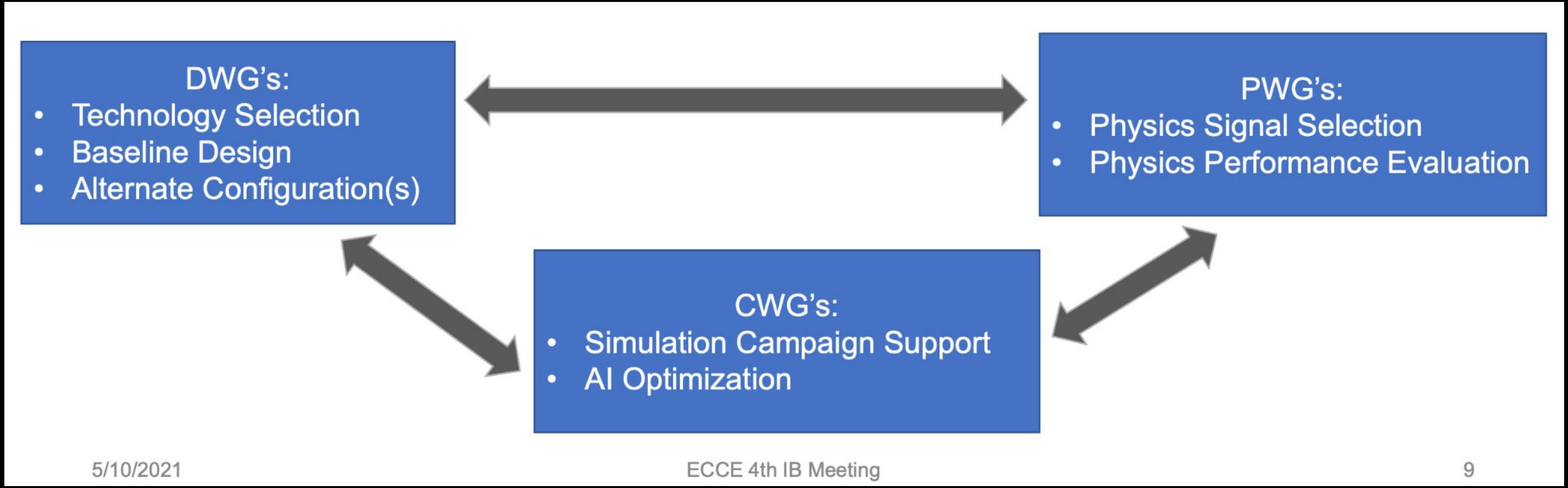
Cristiano Fanelli

Outline

- Origin and scope of AI WG
- Introduction (concrete examples of what can be done and how)
- Action items / plans / activities

Scope

- During the proposal phase we will work with other working groups (physics and detector) to assist in detector design optimization
- In the future this scope could be expanded to include other AI applications as well (AI assisted tracking, etc.)



DWG's:
- Technology Selection
- Baseline Design
- Alternate Configuration(s)

PWG's:
- Physics Signal Selection
- Physics Performance Evaluation

CWG's:
- Simulation Campaign Support
- AI Optimization

5/10/2021 ECCE 4th IB Meeting 9

The ECCE AI-WG - Platforms, Contacts

- Institutions actively involved at present and participating to our slack channel (created before ECCE mattermost perhaps we will migrate there at some point): Brunel, CNU, CUA, Duquesne, JLab, MIT, Regina

- Contact cfanelli@mit.edu, wphelps@jlab.org

- Anyone welcome to reach out to us and/or join the A.I. WG

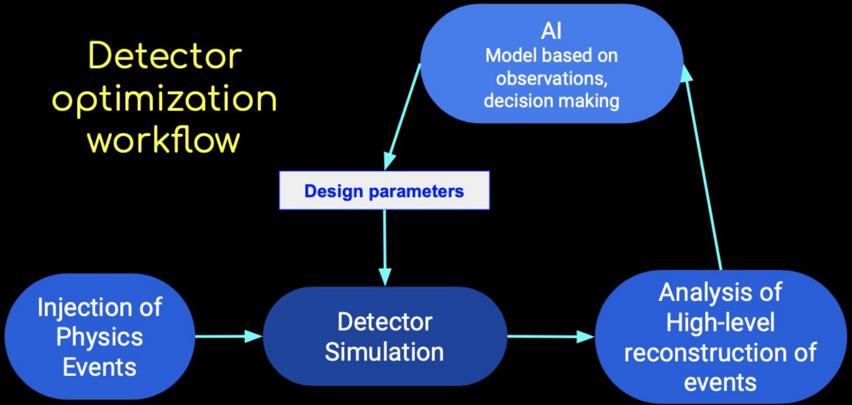
Why AI to improve the design at this phase?

- AI necessary to study/characterize problems with high dimensionality like the design of a complex sub-detector or a system of sub-detectors

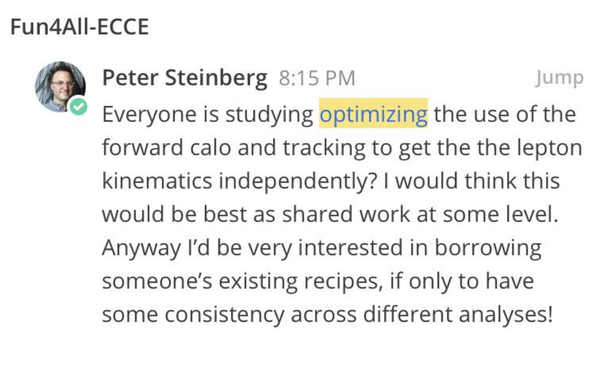
- Optimization does not mean necessarily “fine-tuning”. In a complex problem with multiple design criteria (e.g., performance, cost, material) it helps identifying/approximate the best set of trade-off solutions (Pareto frontier) and decisions can be made based on that.

- We want to use these algorithms to: (1) steer the design and suggest combinations of parameters that a “manual”/brute-force optimization will likely miss to identify; (2) further optimize some particular detector technology (see d-RICH paper, e.g., optics properties)

Detector optimization workflow


AI
Model based on observations, decision making
Design parameters
Injection of Physics Events
Detector Simulation
Analysis of High-level reconstruction of events

Optimization involves analysis of high-level reconstructed observables. Interaction with the other working groups and sharing of information beneficial.

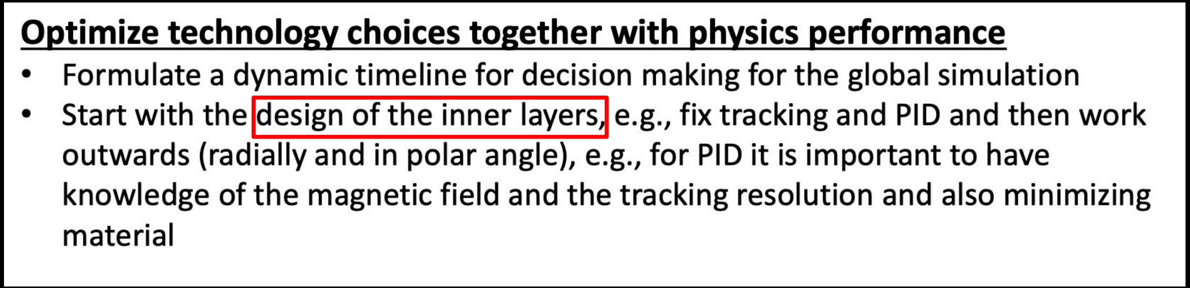

Fun4All-ECCE

Peter Steinberg 8:15 PM

Everyone is studying optimizing the use of the forward calo and tracking to get the the lepton kinematics independently? I would think this would be best as shared work at some level. Anyway I'd be very interested in borrowing someone's existing recipes, if only to have some consistency across different analyses!

AI WG Activities List

- Keeping in mind the "inner to outer" design process/strategy from Tanja's talk at the 5th IB:



Optimize technology choices together with physics performance

- Formulate a dynamic timeline for decision making for the global simulation
- Start with the design of the inner layers, e.g., fix tracking and PID and then work outwards (radially and in polar angle), e.g., for PID it is important to have knowledge of the magnetic field and the tracking resolution and also minimizing material

- Meeting have been organized in the past weeks with the goal to identify specific design tasks for the AI WG. The question for the Detector WGs: *where "optimization" is needed/useful*?

- Suggested/Identified the following activities (discussion later):
  - Tracking (Brunel, MIT, Regina, work in progress)
  - PID --- DIRC (CNU, MIT), d-RICH (MIT d-RICH paper)
  - Calorimetry (CUA, MIT, Regina, work started within eRD1, link to presentation)
  - Far Forward --- ZDC (Duquense, JLab)
  - Other?

Examples and Methods

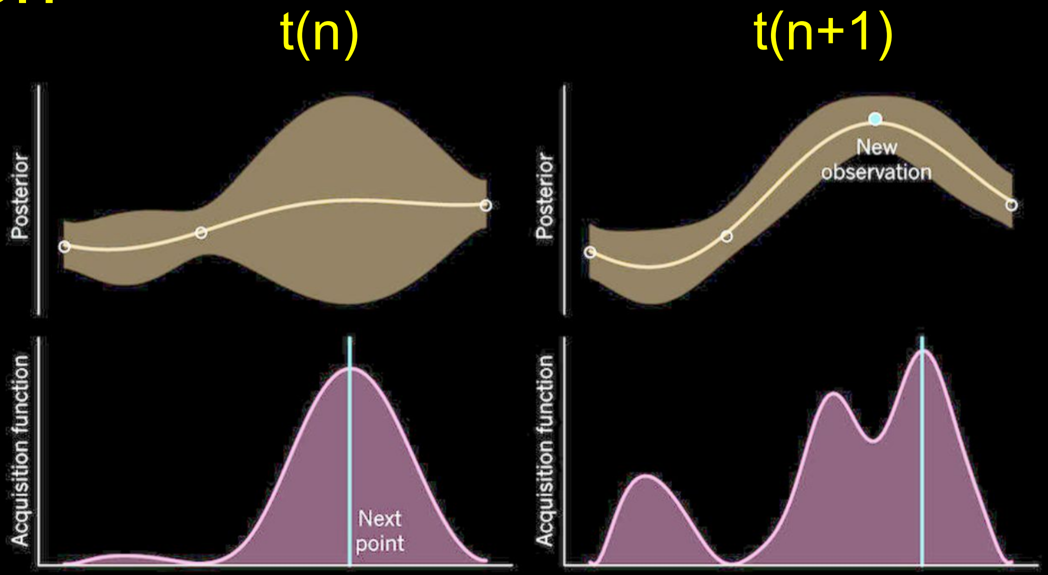
- It’s actually plenty of methods to use and it really depends on the problem you are tackling which one is the most suitable.
- We focus in the following on some example rather than on technical aspects of AI (we have expertise with multiple methods (bayesian optimization, evolutionary methods, deep learning, single/multi-objectives, etc, this discussion can happen offline on slack)
- Two examples:
  - dRICH
  - Si Tracker
- References
  - https://github.com/cfteach/AI4NP_detector_opt (lectures at AI4NP Winter School)
  - https://iopscience.iop.org/article/10.1088/1748-0221/15/05/P05009/meta (d-RICH/ BO)

Bayesian Optimization

- BO is a sequential strategy developed for global optimization.

- After gathering evaluations we builds a posterior distribution used to construct an **acquisition function.**

- This cheap function determines what is **next query point.**



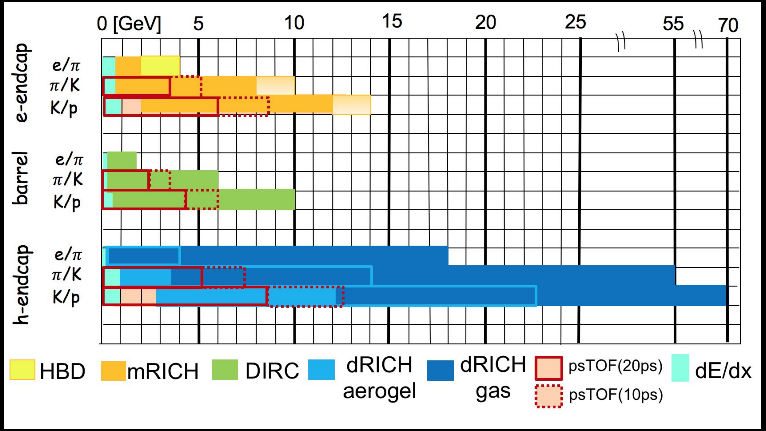
1. Select a Sample by Optimizing the Acquisition Function.
2. Evaluate the Sample With the Objective Function.
3. Update the Data and, in turn, the Surrogate Function.
4. Go To 1.

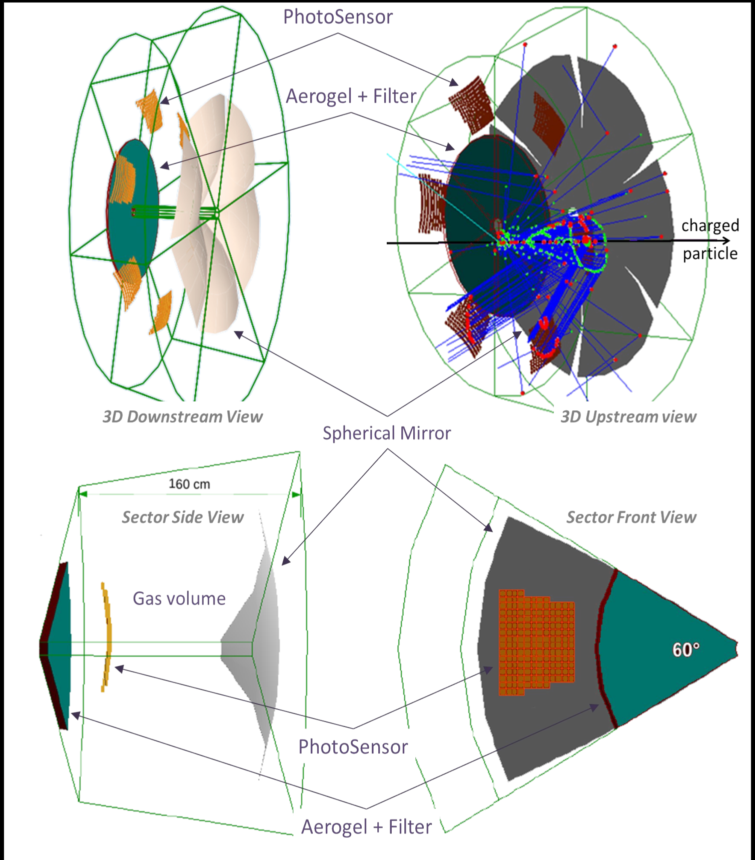
http://krasserm.github.io/2018/03/21/bayesian-optimization/
http://krasserm.github.io/2018/03/19/gaussian-processes/

Dual RICH: case study

E. Cisbani, A. Del Dotto, CF*, M. Williams et al.
"AI-optimized detector design for the future Electron-Ion Collider: the dual-radiator RICH case."
Journal of Instrumentation 15.05 (2020): P05009.

aerogel (4 cm, n(400 nm): 1.02)
+ 3 mm acrylic filter
+ gas (1.6 m, n(C_2F_6): 1.0008)



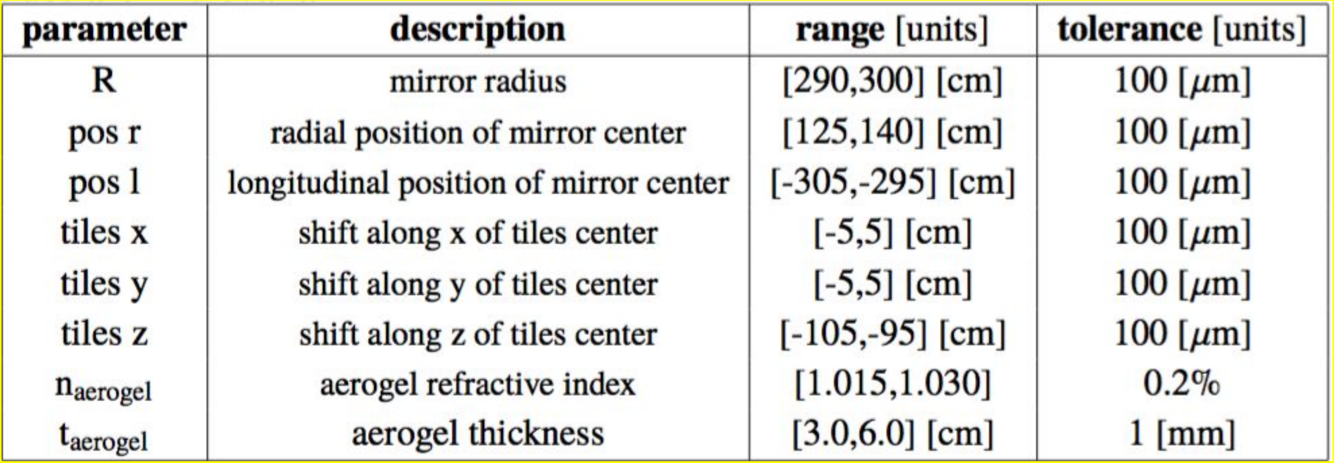


- Continuous momentum coverage.
- Simple geometry and optics, cost effective.
- Legacy design from INFN, see EICUG2017

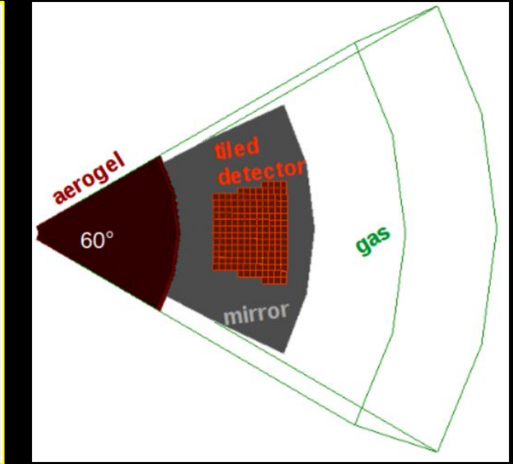
- 6 Identical open sectors (petals)
- Optical sensor elements:
  8500 cm^2/sector, 3 mm pixel
- Large focusing mirror

Construction Constraints on Design Parameters

The idea is that we have a bunch of parameters to optimize that characterize the detector design.
We know from previous studies their ranges and the construction tolerances.



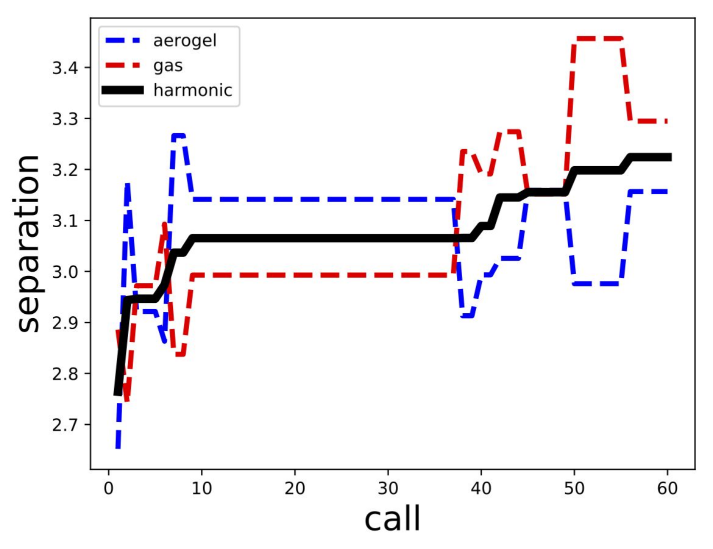
parameter	**description**	**range** [units]	**tolerance** [units]
R	mirror radius	[290,300] [cm]	100 [μm]
pos r	radial position of mirror center	[125,140] [cm]	100 [μm]
pos l	longitudinal position of mirror center	[-305,-295] [cm]	100 [μm]
tiles x	shift along x of tiles center	[-5,5] [cm]	100 [μm]
tiles y	shift along y of tiles center	[-5,5] [cm]	100 [μm]
tiles z	shift along z of tiles center	[-105,-95] [cm]	100 [μm]
$n_{aerogel}$	aerogel refractive index	[1.015,1.030]	0.2%
$t_{aerogel}$	aerogel thickness	[3.0,6.0] [cm]	1 [mm]

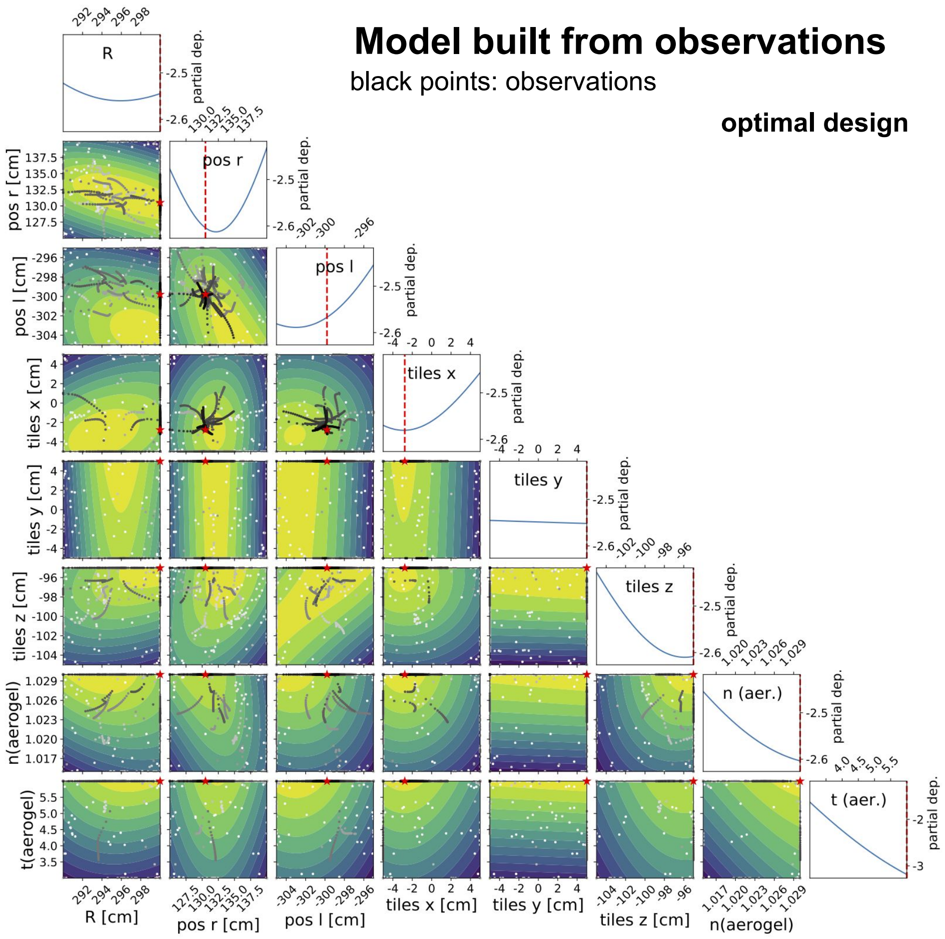


Ranges depend mainly on mechanical constraints and optics requirements.
These requirements can change in the next future based on inputs from prototyping.

The Model and the Optimized FoM

$$N\sigma = \frac{||\langle\theta_K\rangle - \langle\theta_\pi\rangle||\sqrt{N_\gamma}}{\sigma_\theta^{1p.e.}}$$

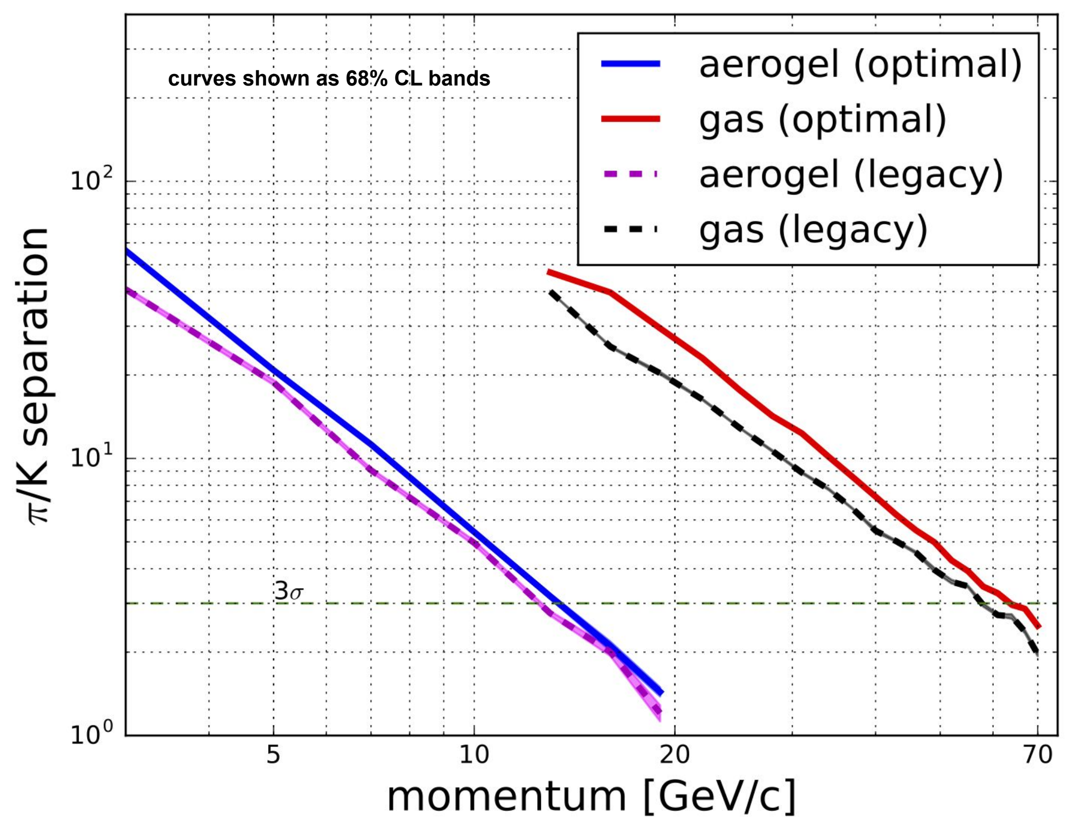



Model built from observations

black points: observations

optimal design

dRICH Performance @ the optimal design point



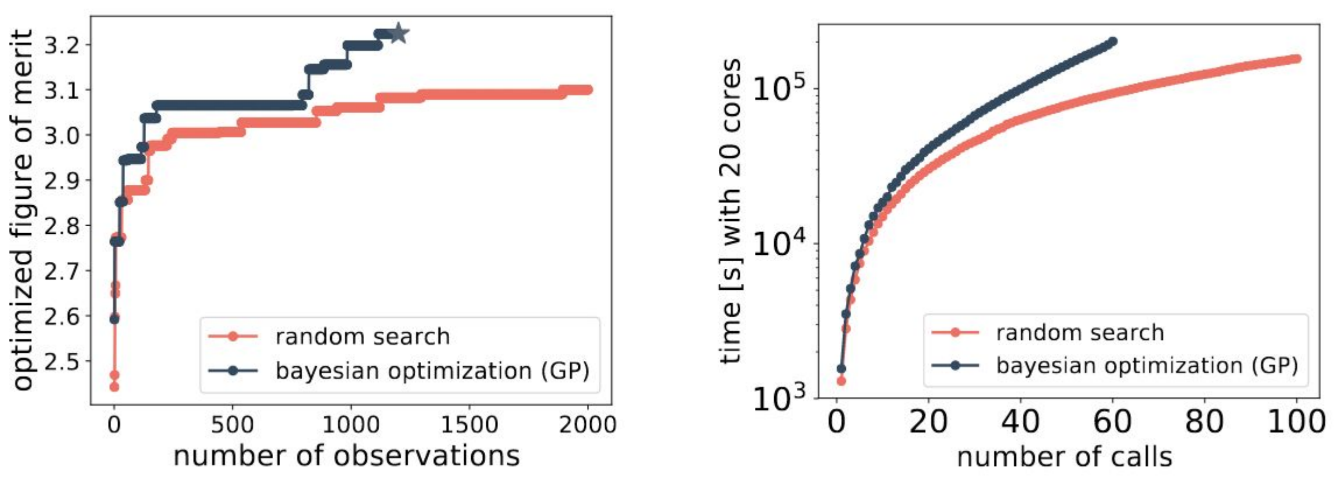
- Statistically significant Improvement in both parts.

- In particular in the gas region where the 5σ threshold shifted from 43 to 50 GeV/c and the 3σ one extended up to

- Notice that before this study we did not know “how well” the legacy design was performing.

E. Cisbani, A. Del Dotto, CF*, M. Williams et al.
JINST 15.05 (2020): P05009.

Comparison with Random Search



Each call:
400 tracks generated/core
20 cores

1 design point ~ 10 mins/CPU

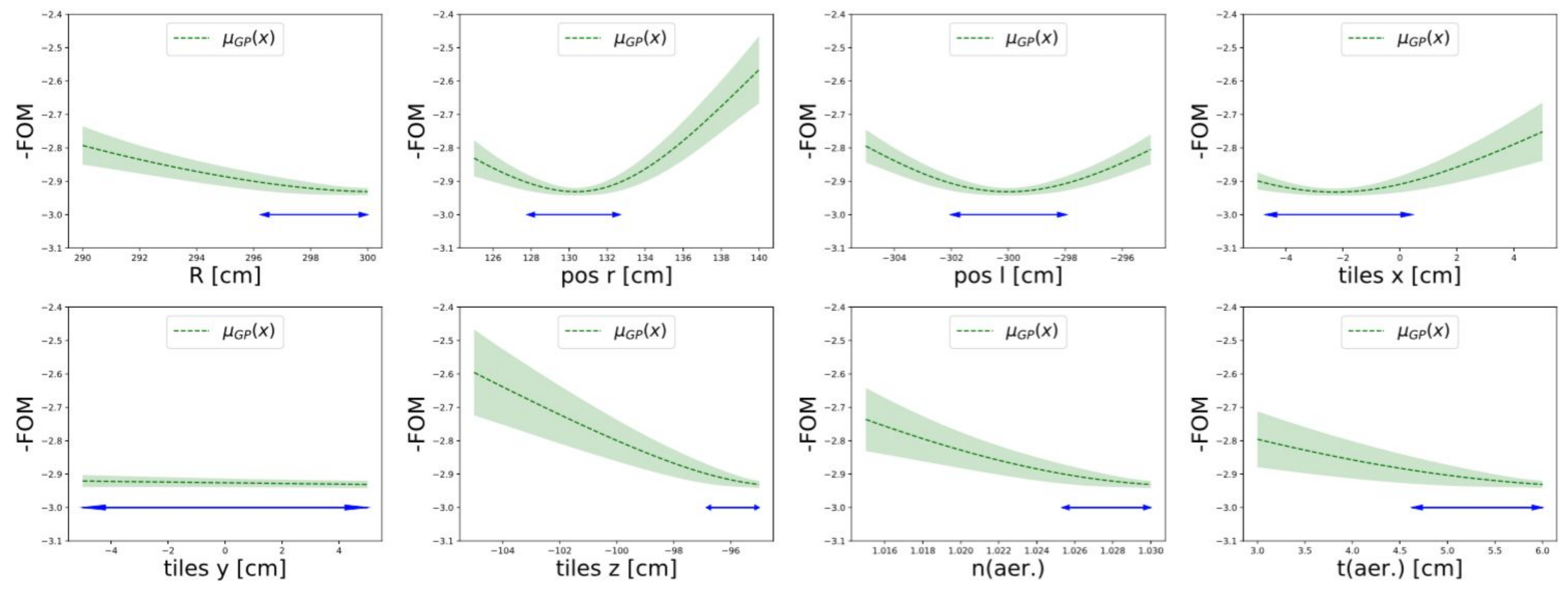
Budget: 100 calls

- BO with GP scales cubically with number of observations.
- Bayesian optimization methods are more promising because they offer principled approaches to weighting the importance of each dimension.
- For this 8D problem - even with 50 cores, RS looks unfeasible due to the curse of dimensionality.
    - Recall that the probability of finding the target with RS is $1-(1-v/V)^T$, where T is trials, v/V is the volume of target relative to the unit hypercube

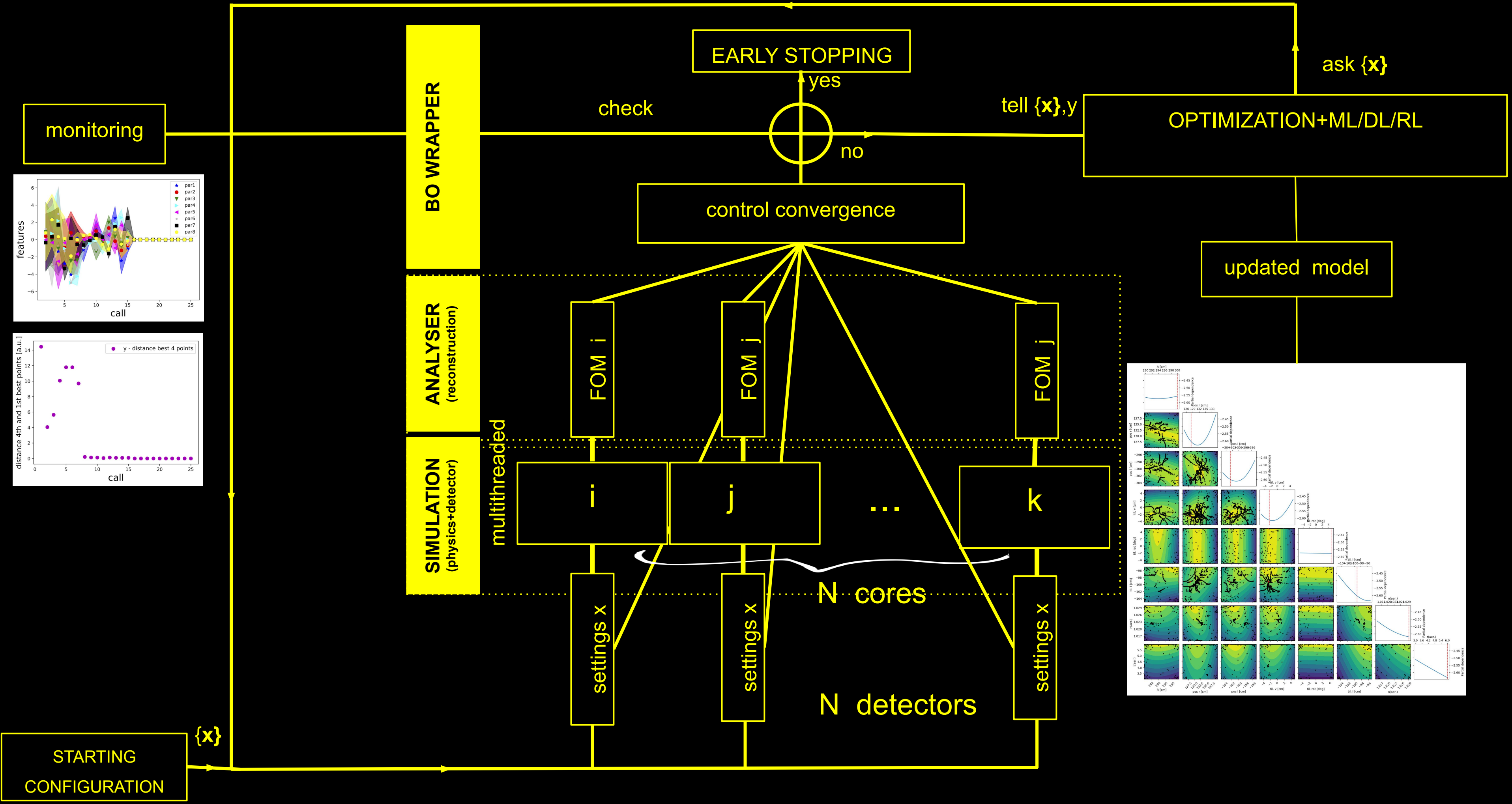
Bergstra, Bengio, “Random search for hyper-parameter optimization”, J. Mach. Learn. Res.13 (Feb) (2012) 281–305.

Tolerance Regions

- BO provides a model of how the FoM depends on the parameters, hence it is possible to use the posterior to define a tolerance on the parameters (regions ensuring improved PID, see previous slide).



- Larger than the construction tolerances on each parameter.
  Notice a small lateral shift of the tiles has negligible impact on the PID capability.



monitoring

BO WRAPPER

check

EARLY STOPPING

yes

no

tell {**x**},y

ask {**x**}

OPTIMIZATION+ML/DL/RL

updated model

control convergence

ANALYSER (reconstruction)

SIMULATION (physics+detector)

multithreaded

FOM i

FOM j

FOM j

i

j

...

k

N cores

settings x

settings x

settings x

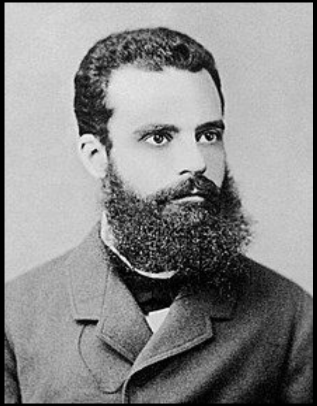
N detectors

STARTING CONFIGURATION

{**x**}

MOO and Pareto Efficiency

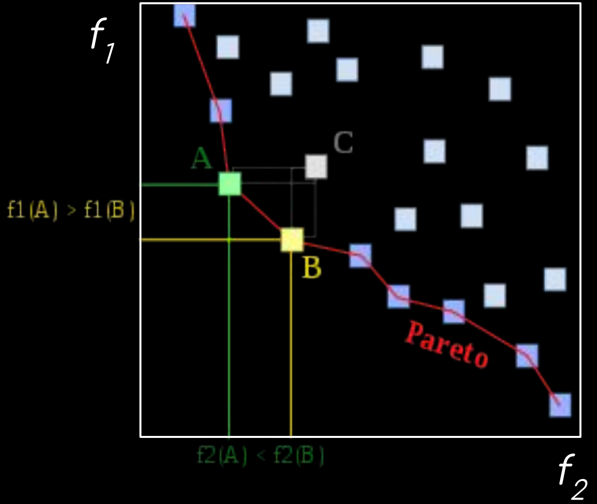
- The problem becomes challenging when the objectives are of conflict to each other, that is, the optimal solution of an objective function is different from that of the other. For example improving the resolution of a detector could imply increasing the costs for its realization.
- In solving such problems, with or without constraints, they give rise to a trade-off optimal solutions, popularly known as Pareto-optimal solutions.



V. Pareto, 1848-1923

- Due to the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms which use a population approach in its search procedure.
- Starting with parameterized procedures in early nineties, the so-called evolutionary multi-objective optimization (EMO) algorithms is now an established field of research.

Point *C* is not on the Pareto frontier because it is dominated by both point *A* and point *B*.



Evolutionary Optimization

[1] Deb, Kalyanmoy. "Multi-objective optimisation using evolutionary algorithms: an introduction." *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London, 2011. 3-34.

- Evolutionary optimization (EO) algorithms use a population based approach in which more than one solution participates in an iteration and evolves a new population of solutions in each iteration.
- The reasons for the popularity of EOs are many:

  (i) do not require any derivative information

  (ii) relatively simple to implement

  (iii) flexible and have a widespread applicability.
- The use of a population of solutions to solve multi-objective optimization problems an EO procedure seems a “natural” choice.
- The MOO problems give rise to a set of Pareto-optimal solutions which need a further processing to arrive at a single preferred solution. To achieve the first task, the use of population in an iteration helps an EO to simultaneously find multiple non-dominated solutions, which portrays a trade-off among objectives, in a single simulation run.

MO-based solutions are helping to reveal important hidden knowledge about a problem
– a matter which is difficult to achieve otherwise [1].

Frameworks

- Notice that MOO with dynamic/evolutionary algorithms (see, e.g., [1-3]) are probably the most utilized approaches on github, followed by more recent developments on multi-objective bayesian optimization (see, e.g., [4-7]). Using them has the advantage of having an entire community developing those tools.

  https://github.com/topics/multi-objective-optimization

- Agent-based approaches to MOO are also possible (see, e.g., [8]), but won't be discussed here.
- Remarkably these approaches can accommodate mechanical and geometrical constraints during the optimization process.

[1] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," Advances in Engineering Software, vol. 42, no. 10, pp. 760–771, 2011.

[2] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," The Journal of Machine Learning Research, vol. 13, no. 1, pp. 2171–2175, 2012.

[3] J. Blank and K. Deb, "pymoo: Multi-objective Optimization in Python," IEEE Access, vol. 8, pp. 89497–89509, 2020

[4] M. Laumanns and J. Ocenasek, "Bayesian optimization algorithms for multi-objective optimization," in International Conference on Parallel Problem Solving from Nature, pp. 298–307, Springer, 2002.

[5] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "Botorch: Programmable bayesian optimization in pytorch," arXiv preprint arXiv:1910.06403, 2019.

[6] P. P. Galuzio, E. H. de Vasconcelos Segundo, L. dos Santos Coelho, and V. C. Mariani, "MOBOpt—multi-objective Bayesian optimization," SoftwareX, vol. 12, p. 100520, 2020.

[7] A. Mathern, O. S. Steinholtz, A. Sjöberg, M. Önnheim, K. Ek, R. Rempling, E. Gustavsson, and M. Jirstrand, "Multi-objective constrained Bayesian optimization for structural design," Structural and Multidisciplinary Optimization, pp. 1–13, 2020.

[8] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," in Advances in Neural Information Processing Systems, pp. 14636–14647, 2019

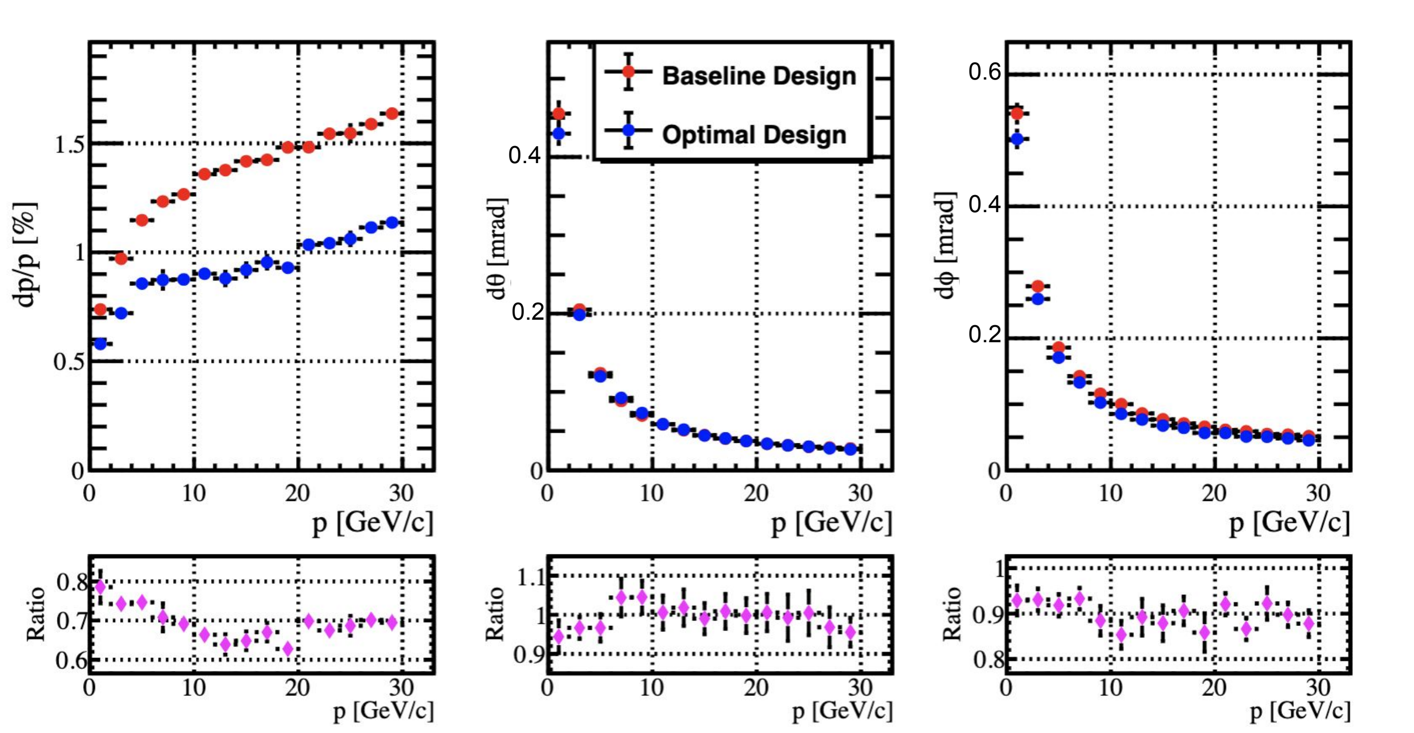
Si Tracker Exercise: Out of the box

K. Suresh, U. of Regina & CF

https://github.com/reynier0611/g4lblvtx/ (private repo)
Fun4All_G4_simplified_v2_DIRC_barrel_GEM.C

obj = 1/N(bins) * $\sum_i$ res_i(opt) / res_i(base)

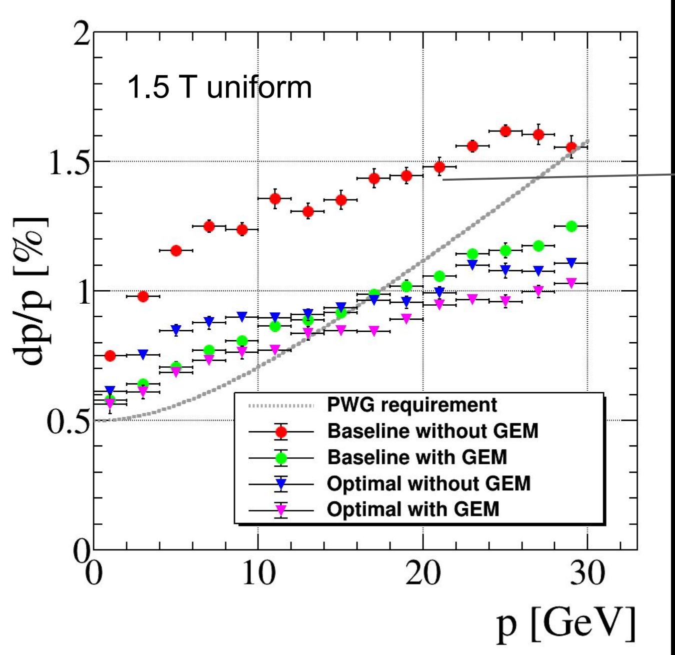
- Pipeline with 4 parameters (radii of barrel Si tracker), 3 objectives, 1 constraint.
- 30 k pions per design point (uncertainty on objectives of ~1-2%) generated flat in P in [0,30] GeV at |η|<0.5
- First test with few evaluations using an evolutionary algorithm (~ few hours) distributed on 4 cores.

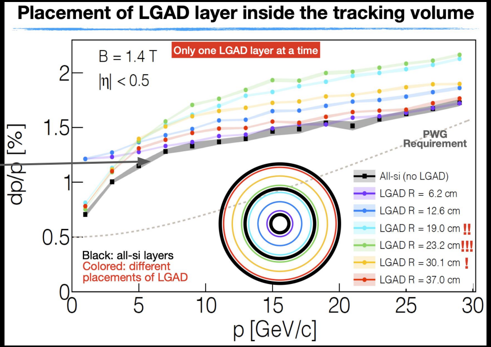


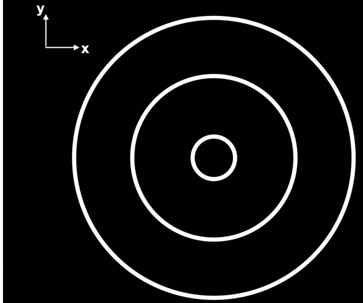
Si Tracker Exercise: Out of the box

K. Suresh, U. of Regina & CF

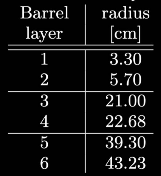
See very nice talk by R. Cruz , 05/21/2021



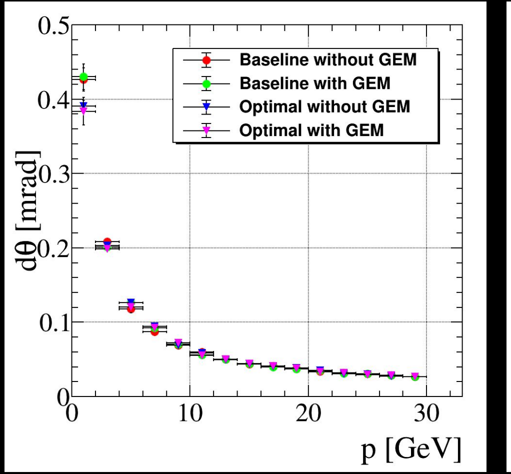

Placement of LGAD layer inside the tracking volume

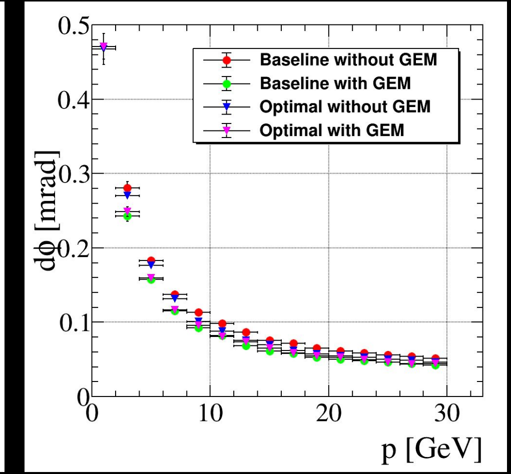


All-silicon tracker layers



Barrel layer	radius [cm]
1	3.30
2	5.70
3	21.00
4	22.68
5	39.30
6	43.23

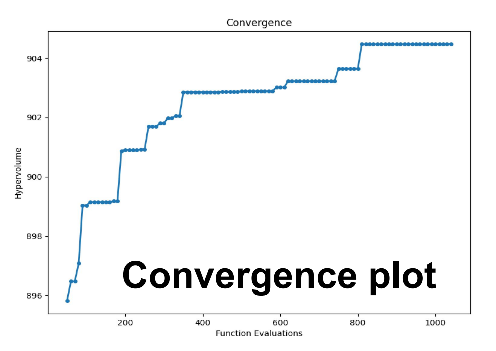


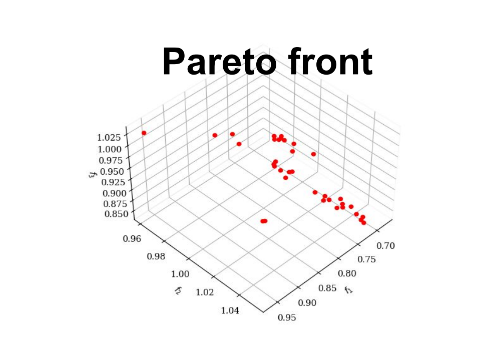


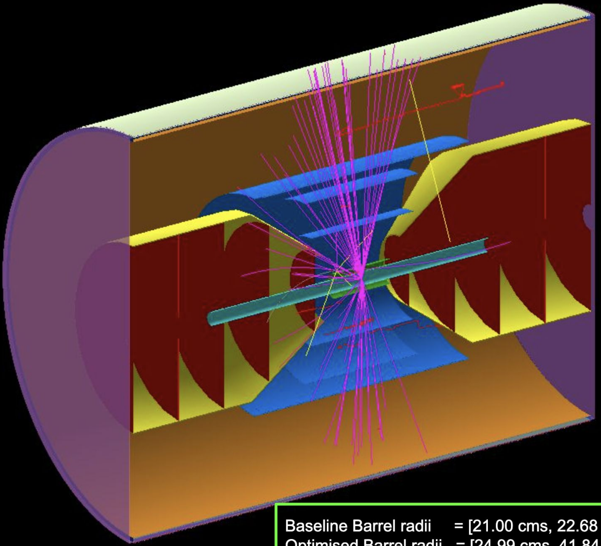
PWG curve: YR Detector Matrix

Si Tracker Exercise: Out of the box

K. Suresh, U. of Regina & CF







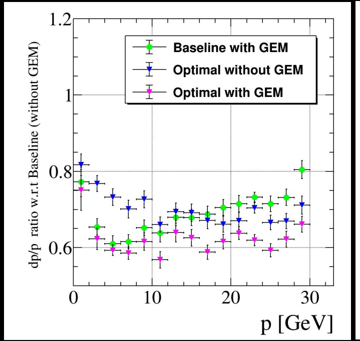
Baseline Barrel radii = [21.00 cms, 22.68 cms, 39.30 cms, 43,30 cms]
Optimised Barrel radii = [24.99 cms, 41.84 cms, 49.90 cms, 50.45 cms]

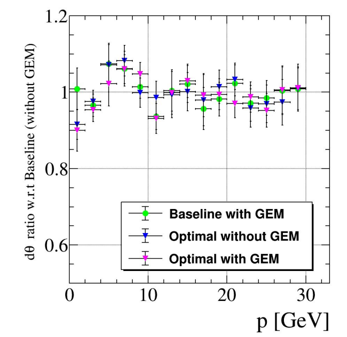
R1, R2, R3, R4 [cm]
baseline (21.0, 22.7, 39.3, 42.2)
“optimized” (25.0, 41.8, 49.9, 50.4)

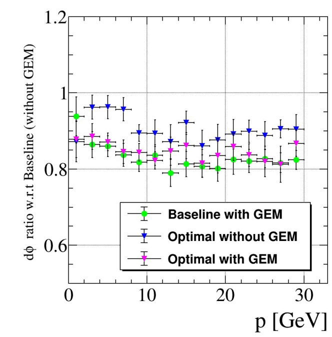
* running new optimization allowing for more space

(improvement in resolutions compared to baseline without GEM)

1-obj(dp/p)	1-obj(dθ)	1-obj(dφ)
30% (±2%)	2% (±2%)	11% (±2%)

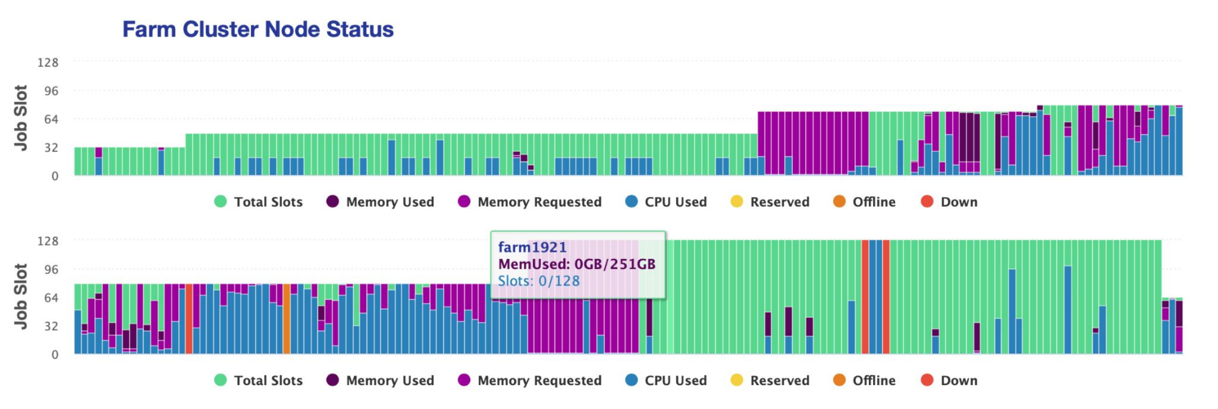






Jefferson Lab Computing Resources

- The scientific computing cluster known as the “farm” has 25k cores
  - EIC Projects are allocated 10%
- 1PB for EIC use



The “Farm” status on an unusually quiet day (scicomp.jlab.org)

- Batch use as well as interactive use supported with
  - Nodes with up to two 32 core AMD Epyc Processors (128 threads), 256GB Ram, 1TB SSD local storage
  - 3 Nodes with 4 Titan RTX Cards (24 GB Memory)
  - GPU nodes also available through jupyterhub.jlab.org
  - Additional GPU nodes arriving soon

Discussion on Activities / Projects, distribution of work

SPARES