Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

AI assisted Detector Design for EIC
-- Distributed AI workflow with PanDA/iDDS

Wen Guan, Tadashi Maeno, Xin Zhao and Torre Wenaus on behalf of the PanDA team

Sept 27, 2023

A Scalable and Distributed AI-assisted Detector Design for the EIC

- A 2 year project from Sep 15 2023 supported by DOE NP, ~$700k/yr total
- Lead PI: Cristiano Fanelli (William & Mary)
- Collaborating institutions:
  - BNL: Physics (NPPS - T Wenaus) and CSI (HPC - Meifeng Lin)
    - Supported participants: Wen Guan (35%) and similar fraction of experienced CSI person Tianle Wang
  - JLab: Experimental Nuclear Physics - Markus Diefenthaler
  - Unis: Duke - Anselm Vossen, Catholic University of America - Tanja Horn
- Wen will give a project overview and discuss our involvement and what we're bringing ('scalable and distributed')
- Ancillary benefits are important also
  - Close collaboration with our sister EIC host lab on
    - AI/ML and PanDA based complex workflows
      - Discussing with JLab how/where to establish a PanDA instance for the project and more generally EIC PanDA investigation
    - EIC simulation
      - Complements an LDRD project on this topic we're just now starting, also a collaboration with JLab

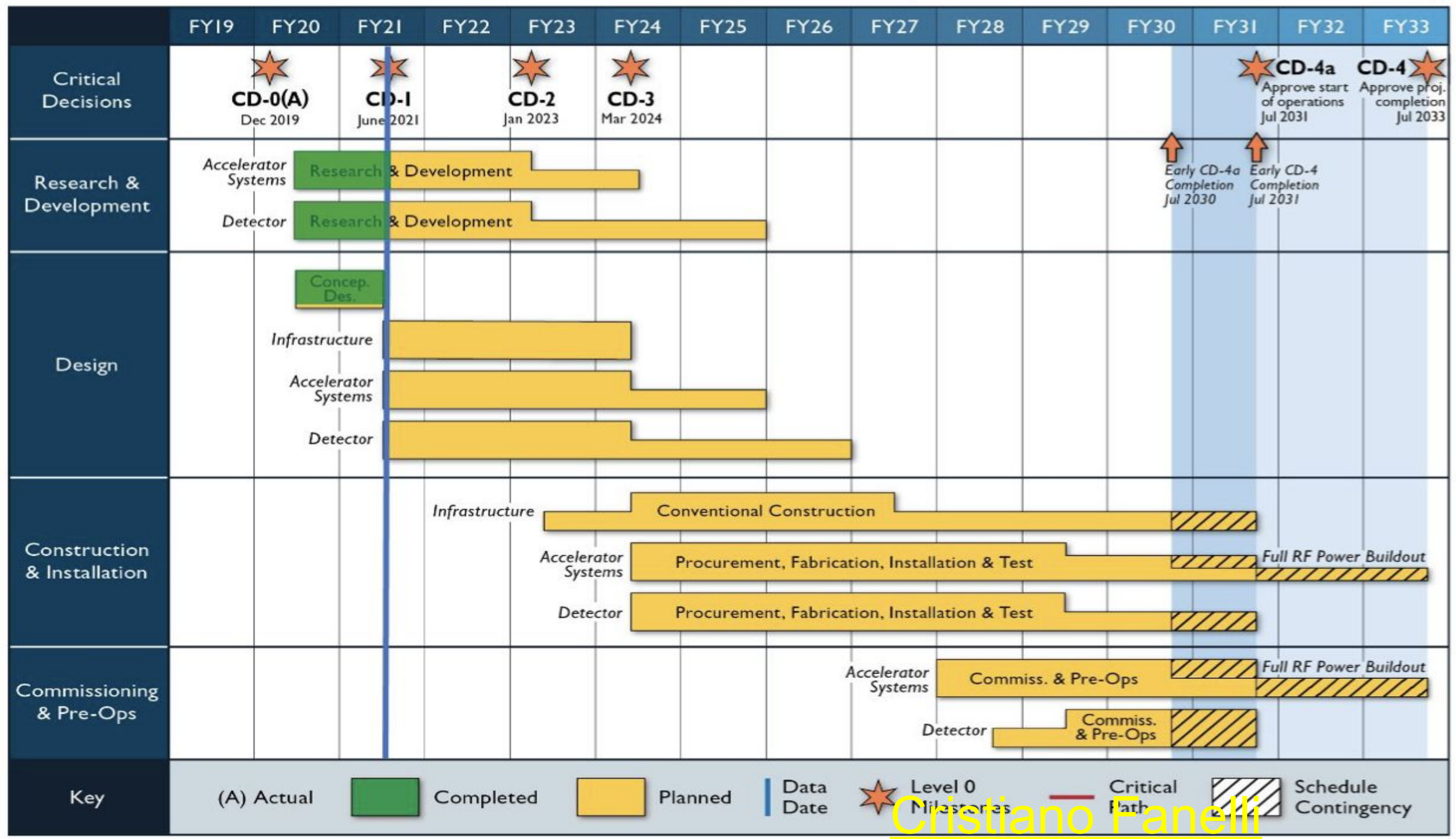
Introduction -- AI for EIC Detector (from Cristiano Fanelli)

EIC and Timeline - How AI come into play?

- AI basically present in all phases of the EIC schedule
- The EIC R&D program can be one of the first to systematically leverage on AI during the detector design phase
- AI can advance research, design, and operation of the EIC. In the Yellow Report, Sec. 11.12 (Artificial Intelligence for the EIC detector), we individuate specific aspects that can be potentially tackled with AI.
- Supported by new approaches like Streaming RO, the EIC can become one of the first largely automated experiments (e.g., calibration)

See talk R. Ent, Overview of the EIC Program

EIC operations are a decade away



Cristiano Fanelli

4

Introduction -- AI for EIC Detector Design (from Cristiano Fanelli)

Detector Design with AI

- Designing detectors "with" AI is a new area of research at its infancy that can have a tremendous impact across many fields (NP, HEP, Astro-Phys). See lectures https://github.com/cfteach/AI4NP_detector_opt given at the AI4NP winter school https://indico.jlab.org/event/409/.
- It includes a broad range of approaches, from "optimizing" an existing expert-drawn baseline detector concept, to in principle letting AI design completely "new" and unseen configurations.
- New field, not many examples... Many applications in other fields in recent years, e.g., industrial material, molecular and drug design [1, 2].
- AI-driven design is not limited to "interfacing" AI with existing advanced simulation platforms used in our community (Geant). It also (and principally) entails establishing a procedural body of instructions to encode efficiently the optimal design requirements and validate the results in a self-consistent way [3].
- As far as optimization is concerned, the choice of a suitable algorithm is a challenge itself (no free lunch theorem [4]) and the full potential of certain algorithms always requires some degree of customization. First thing to do is to study and characterize the properties of the problem.

[1] A. Mosavi, T. Rabczuk, and A. R. Varkonyi-Koczy, "Reviewing the novel machine learning tools for materials design," in Int. Conference on Global Research and Education, pp. 50–58, Springer, 2017
[2] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley, "Optimization of molecules via deep reinforcement learning," Scientific Reports, vol. 9, no. 1, pp. 1–10, 2019
[3] CF et al. "AI-optimized detector design for the future Electron-Ion Collider: the dual-radiator RICH case." *JINST* 15.05 (2020): P05009.
[4] Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. Trans. Evol. Comp 1, 67–82

Cristiano Fanelli

Introduction -- Parameter Optimization (from Cristiano Fanelli)

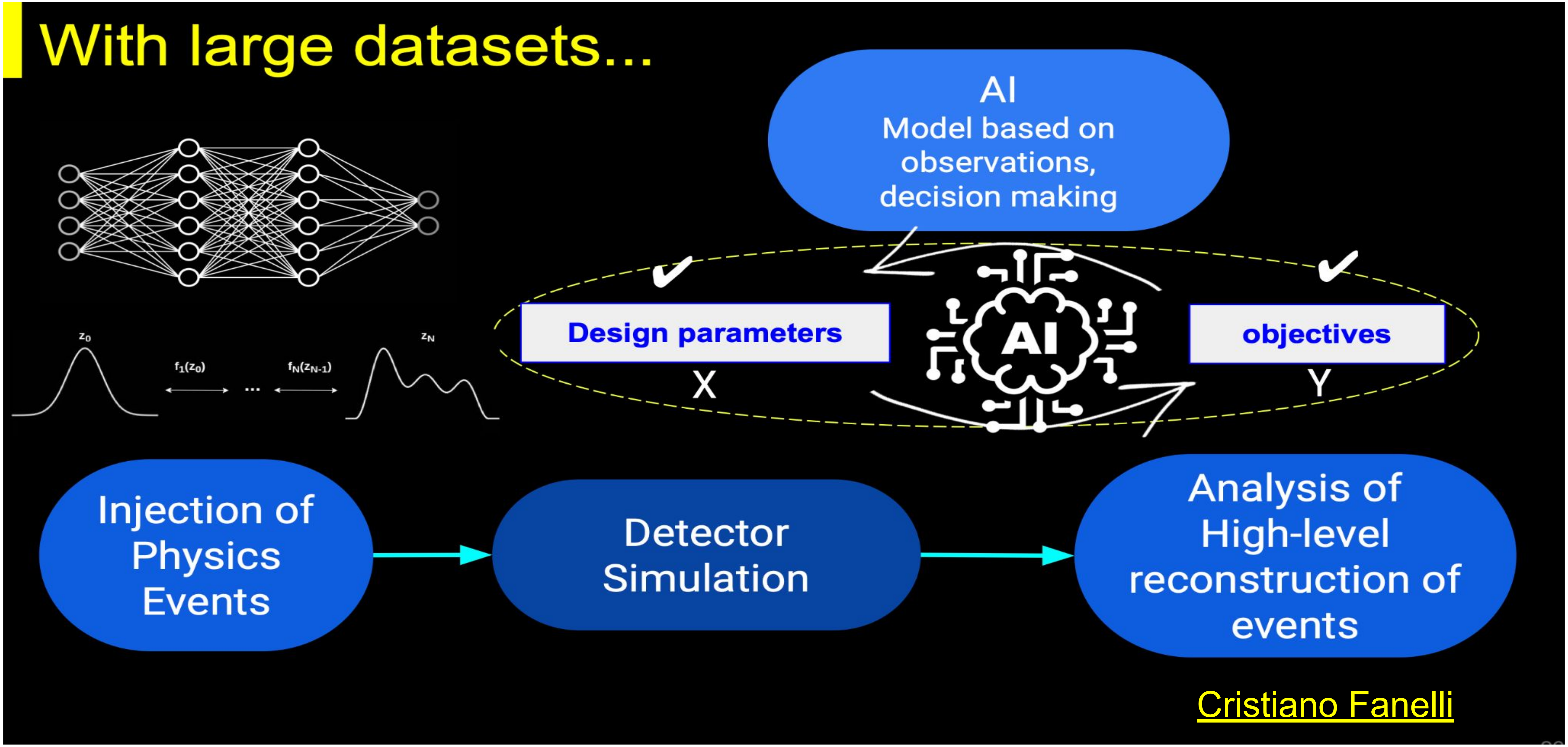
How do we design and optimize detectors?

- Typically full detector design is studied once the subsystem prototypes are ready.
- In the subsystem design phase constraints from the full detector or outer layers are taken into consideration.
- Actually many parameters (mechanics, geometry, optics) characterize the design of each sub-detector, hence the full design represents a large combinatorial problem. A well known phenomenon observed in optimization problems with high-dimensional spaces is the so-called "curse of dimensionality" [1], introduced for the first time by Bellman when considering problems in dynamic programming.
- In addition to that, more objective functions often need to be considered at the same time in the design of each sub-detector (e.g., resolution, efficiency, cost, distinguishing power, etc).
- In this context, AI offers SOTA solutions to solve complex optimization problems in an efficient way.

Cristiano Fanelli

[1] Bellman, Richard. *Dynamic programming*. Vol. 295. RAND CORP SANTA MONICA CA, 1956.

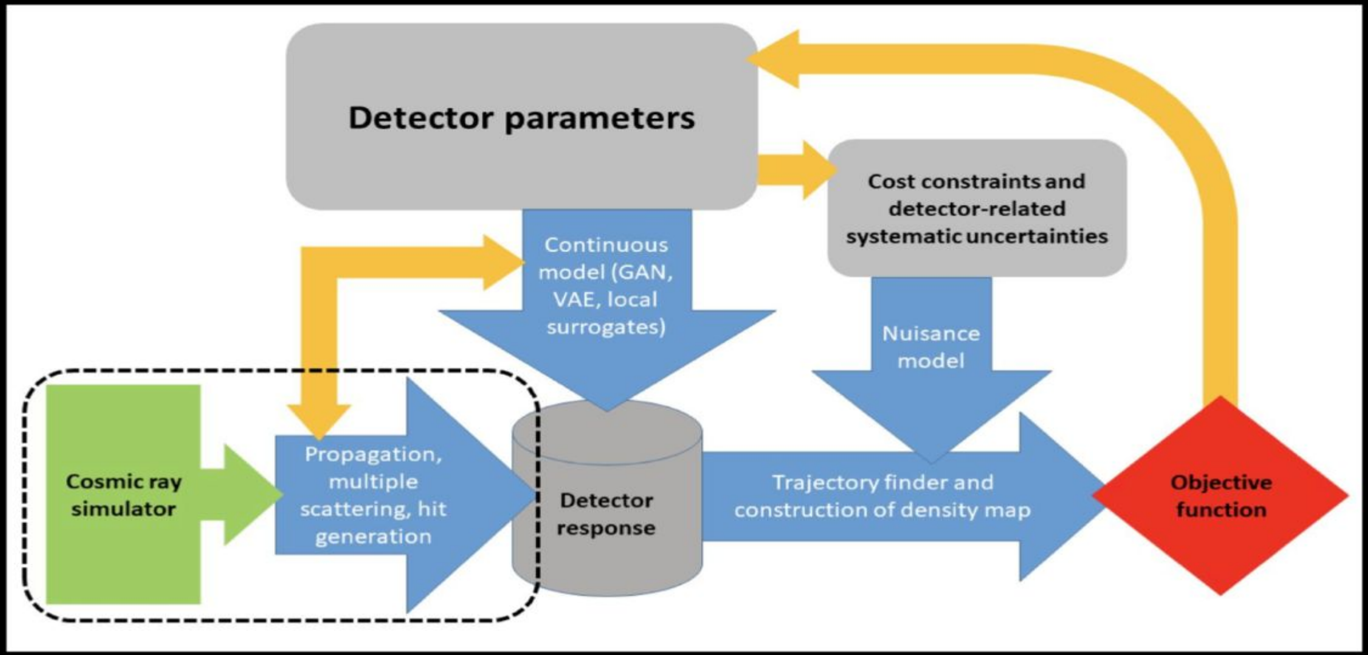
Introduction -- AI4EIC Dectector Opt workflow (from Cristiano Fanelli)



Introduction -- AI4EIC Dectector Mode (from Cristiano Fanelli)

MODE

- Detectors design with AI is gaining a lot of interest.
- MODE is a recently formed collaboration of physicists and computer scientists who target the use of differentiable programming in design optimization of detectors for particle physics applications A. G. Baydin et al. Nuclear Physics News 31.1 (Mar 30, 2021): 25-28.
- Ambitious project: develop a modular, customizable, and scalable, fully differentiable pipeline for the end-to-end optimization of articulated objective functions that model in full the true goals of experimental particle physics endeavours, to ensure optimal detector performance, analysis potential, and cost-effectiveness.



Conceptual layout of an optimization pipeline for a muon radiography apparatus.

An end to end optimization requires modeling of simulations. Requires collect reference data to train the surrogate models ML implementations.

Cristiano Fanelli

27

Parameter Optimization with AI

- Objectives
  - In this project, we will focus on the part of parameter optimization with AI.
  - Especially employ PanDA/iDDS to provide a Distributed Machine Learning (DML) platform, with also DML R&D.
- Special requirements:
  - Many Parameters
    - Search space is big, many parallel jobs are required.
  - Multiple Objectives
    - Multiple Objective Optimization
    - Multiple Objective Bayesian Optimization
- Solutions
  - Many CPU intensively ---- Distributed with PanDA
  - Multiple steps workflow orchestration ---- iDDS

$$\min_{x \in X}(f_1(x), f_2(x), \ldots, f_k(x))$$

$$f : X \to \mathbb{R}^k$$

$$x \mapsto \begin{pmatrix} f_1(x) \\ \vdots \\ f_k(x) \end{pmatrix}$$

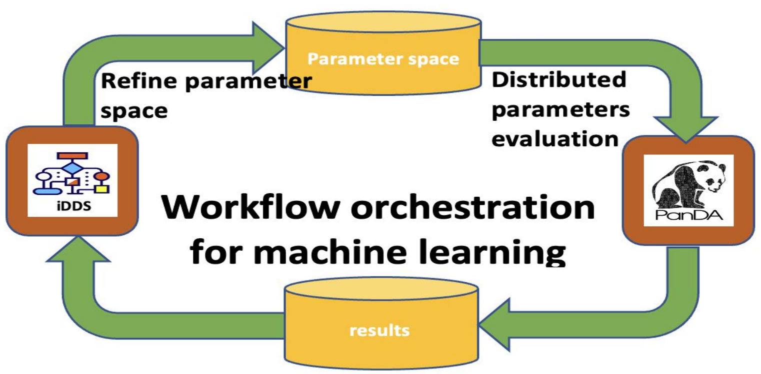
Current experience

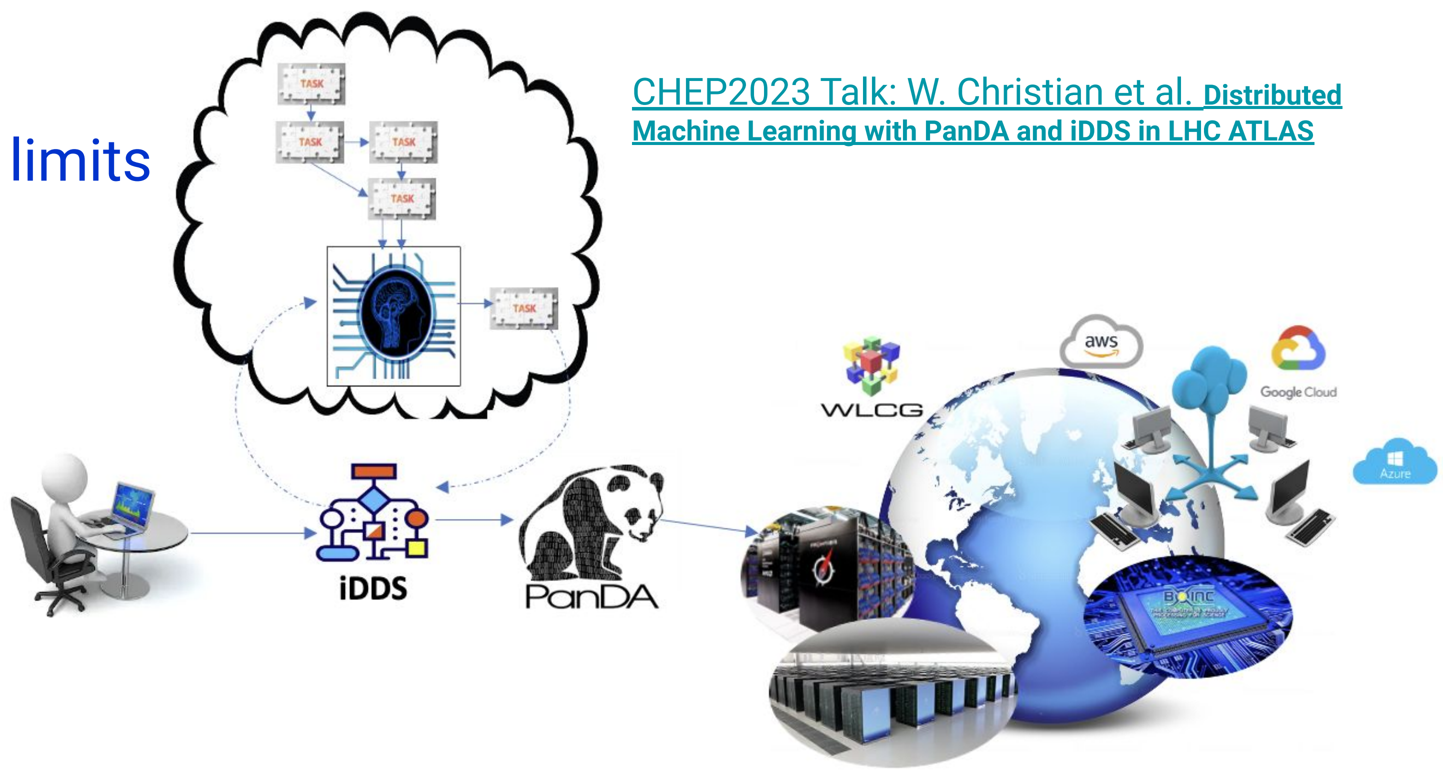
Distributed ML with PanDA and iDDS in ATLAS

- ❖ PanDA as an engine for large scale AI/ML
  - ➢ PanDA is powerful to schedule jobs to distributed heterogeneous resources
  - ➢ Large scale
  - ➢ Transparent to users for different computing resources
  - ➢ Smart workload routing

- ❖ iDDS (intelligent Data Delivery Service) orchestrates the workflow for automation
  - ➢ Complex workflow orchestration
    - ■ Collect results from previous tasks
    - ■ Analyze the results with user predefined jobs
    - ■ Generate new tasks/jobs based on the analyses

- ❖ Use Cases
  - ➢ HyperParameter Optimization
  - ➢ Monte Carlo Toy based confidence limits
  - ➢ Active Learning





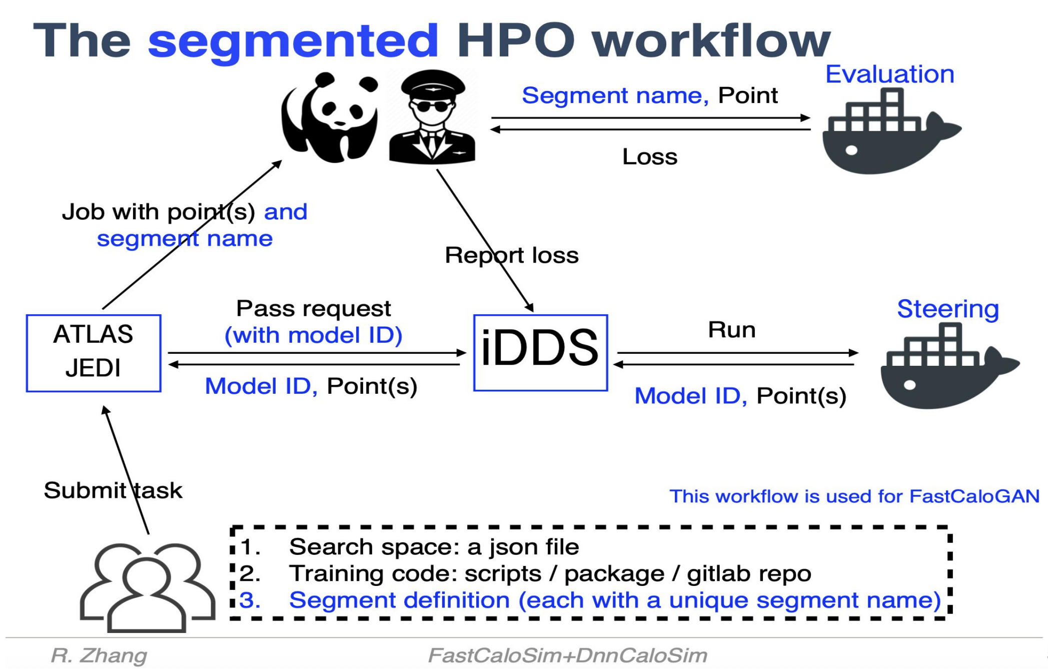
CHEP2023 Talk: W. Christian et al. **Distributed Machine Learning with PanDA and iDDS in LHC ATLAS**

Brookhaven National Laboratory

Current experience

HyperParameterOptimization (HPO) iDDS

- HPO includes two parts
  - Hyper parameter generating (steering)
    - Asynchronously ask-tell mode: when needing more parameters, this part is called to generate points
    - Bayesian method can be used here
  - Evaluation
    - Distributed jobs in parallel to evaluate different parameters
- iDDS HPO automates hyperparameter generation and evaluation with many iterations: new hyperparameters are generated automatically from previous evaluation results.
- PanDA distributes evaluation tasks to CPU/GPUs on potentially geographically distributed resources.



Challenges

- Multiple steps workflow orchestration in iDDS
  - Client
    - In the current iDDS HPO implementation, users need to explicitly convert operations/functions to iDDS Work/Task.
    - The AI4EIC workflow is very complicated. It's very inconvenient to do this explicitly conversion.
    - New methods are under investigation, for example, python decorator
  - Server
    - The current HPO implementation is based on an ask-tell mode.
    - In AI4EIC workflow, the main script is complicated. It not only generates new parameters, but also does other work. The ask-tell mode may not fit it. It may need to be running persistently.
    - Investigating whether it's possible to checkpoint/restart
- Inputs/outputs processing for different jobs
  - How to efficiently and seamlessly transfer inputs and outputs for different jobs in a very large scale, to construct the pipeline: output of one job can be input of another job
- Logs
  - It's important to see the logs as soon as possible locally for DML
  - Realtime logging?

Preliminary work plans

- Environment setup
  - Deploy a PanDA/iDDS environment --- postgresql based PanDA, k8s?
  - Distributed jobs to site CPUs (OSG?)
- Workflow orchestration:
  - iDDS new workflow structure developments
  - Investigate and develop new methods to handle inputs/outputs seamlessly for jobs
- Executor in pilot
  - Inputs fetching and outputs forwarding
- Logs
  - Realtime logging
- More future work
  - DML Improvements and ML R&D

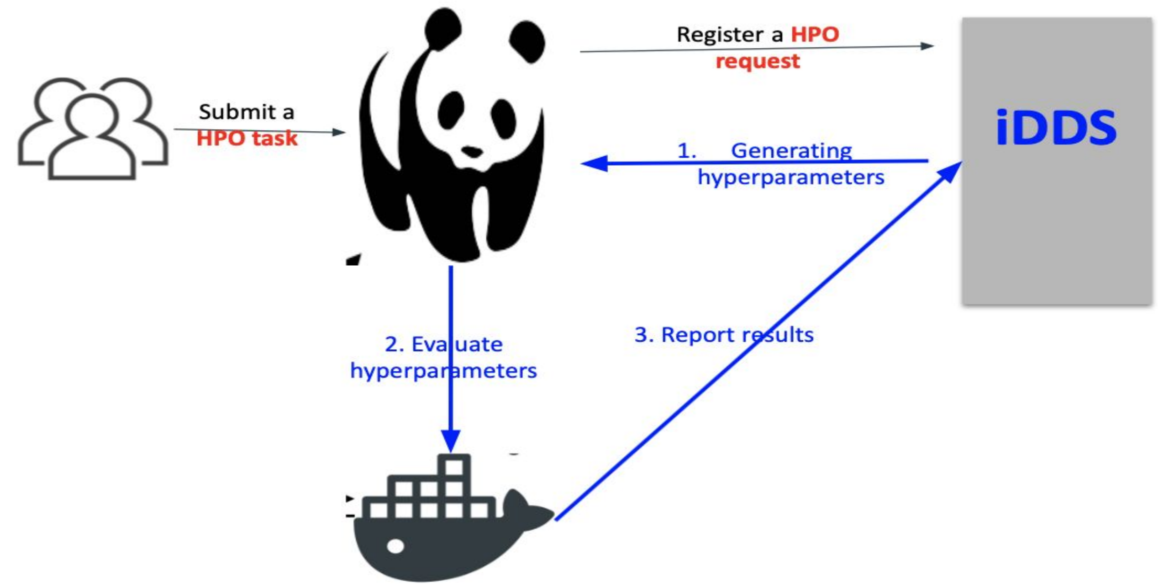
Thanks

Backups

iDDS HyperParameter Optimization (HPO)

iDDS HPO provides a fully-automated platform for hyperparameter optimization on top of geographically distributed CPU/GPU resources on the Grid, HPC and Clouds.

- ❖ A group of optimized hyperparameters can greatly improve the physics analysis performance. A lot of LHC analyses are using HPO to enhance the performance.
- ❖ iDDS HPO automates hyperparameter generation and evaluation with many iterations: new hyperparameters are generated automatically from previous evaluation results.
- ❖ iDDS HPO distributes ML tasks to CPU/GPUs on potentially geographically distributed resources.
- ❖ iDDS HPO has been **used by ATLAS ML users**, not specific to ATLAS.
- ❖ Different use cases are using the HPO framework to automate distributed tasks.
  - FastCaloGAN
  - Monte Carlo toy based confidence limits estimation (requiring multiple steps of grid scans, where current steps depend on previous steps)
  - AtlFast3 fast simulations



- ❖ **A HPO task should include two parts**
  - ➢ Hyperparameter generating:
    - ■ Option 1: define search space with predefined methods
    - ■ Option 2: develop user container
  - ➢ Evaluation
    - ■ User ML training/learning process