Introduction to the Jefferson Lab Data Science Department and its Capabilities

CLAS Collaboration Meeting March 2024

Daniel Lersch & Malachi Schram

Thursday, March 17, 2024

Jefferson Lab

U.S. DEPARTMENT OF ENERGY | Office of Science

JSA

The Team

Malachi Schram
Department Head

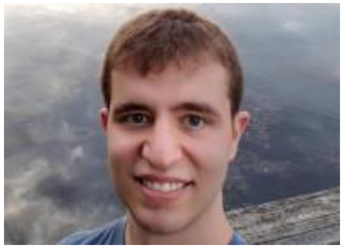
- 8 Researchers: 5 Staff + 3 Postdocs
- 4 Researchers from ODU / JLab joint institute
- And we keep growing

Kishansingh Rajput
Staff

Diana McSpadden
Staff

Armen Kasparian
Staff

Daniel Lersch
Staff



Steven Goldenberg
Postdoc

Ahmed Mohammed
Postdoc

Zhenyu Dai
Postdoc

Data Science at Jefferson Lab

Mission:

- Provide solutions to advance research across the Department of Energy complex
- Work with the subject matter experts at Jefferson Lab, partnering laboratories, and universities
- Provide solutions to scientific applications relevant to the <u>regional scientific community</u>

Vision:

- Expand the <u>capability</u> and <u>capacity</u> of data science at JLab
- Create a <u>collaborative</u> data science research hub to:

1. Work with regional partners on challenging scientific problems
2. Champion education and research opportunities with regional universities and industry
3. Reduce the carbon footprint by optimizing the data science workflow and algorithms

Building Collaborations

University of Virginia

Oak Ridge National Laboratory

Brookhaven National Laboratory

University of California · Santa Barbara · Let There Be Light

EVMS Eastern Virginia Medical School

Old Dominion University

Fermilab 50 Years of Discovery

SLAC National Accelerator Laboratory

CHKD

Cornell University Founded A.D. 1865

Pacific Northwest National Laboratory

Argonne National Laboratory

W&M

VT Virginia Tech

NREL National Renewable Energy Laboratory

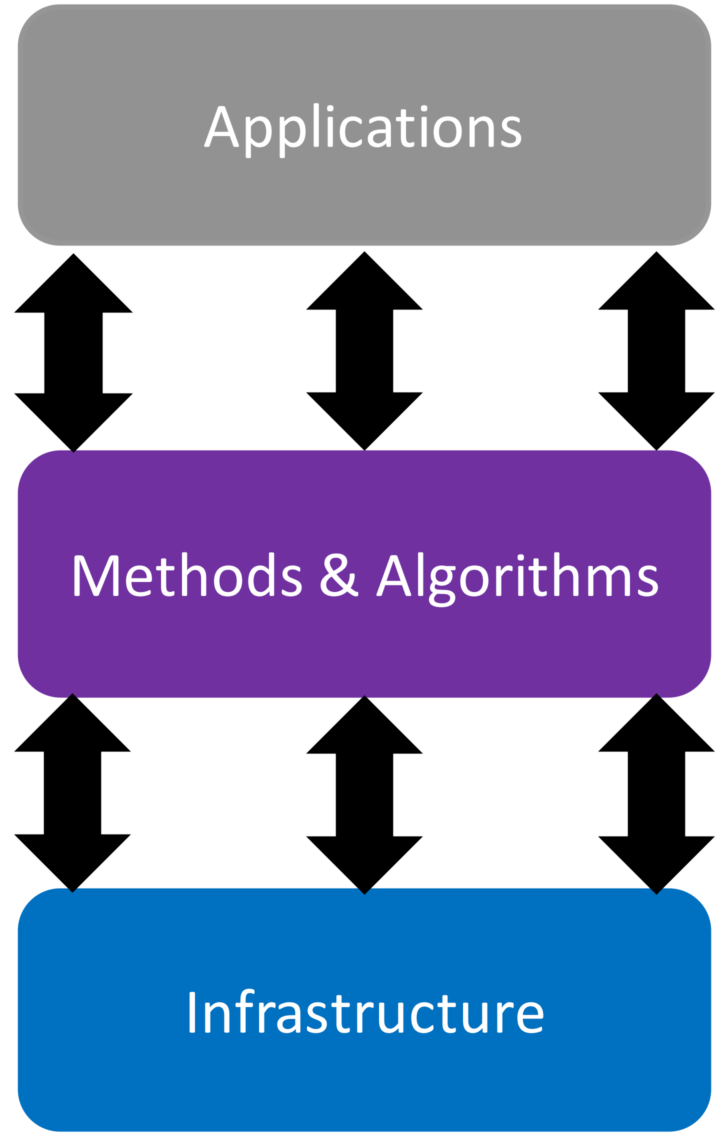
DESY.

The Jefferson Lab Data Science Pillars

- Nuclear Physics (**NP**), High Energy Physics (**HEP**), Advanced Scientific Computing Research (**ASCR**), Basic Energy Sciences (**BES**)
- Health & Climate

- AI based optimization & Controls
- Explainability and Robustness
- Generative AI
- Scalable AI

- JLab Data Science Composable Workflow
- JLab ML & Data Hub



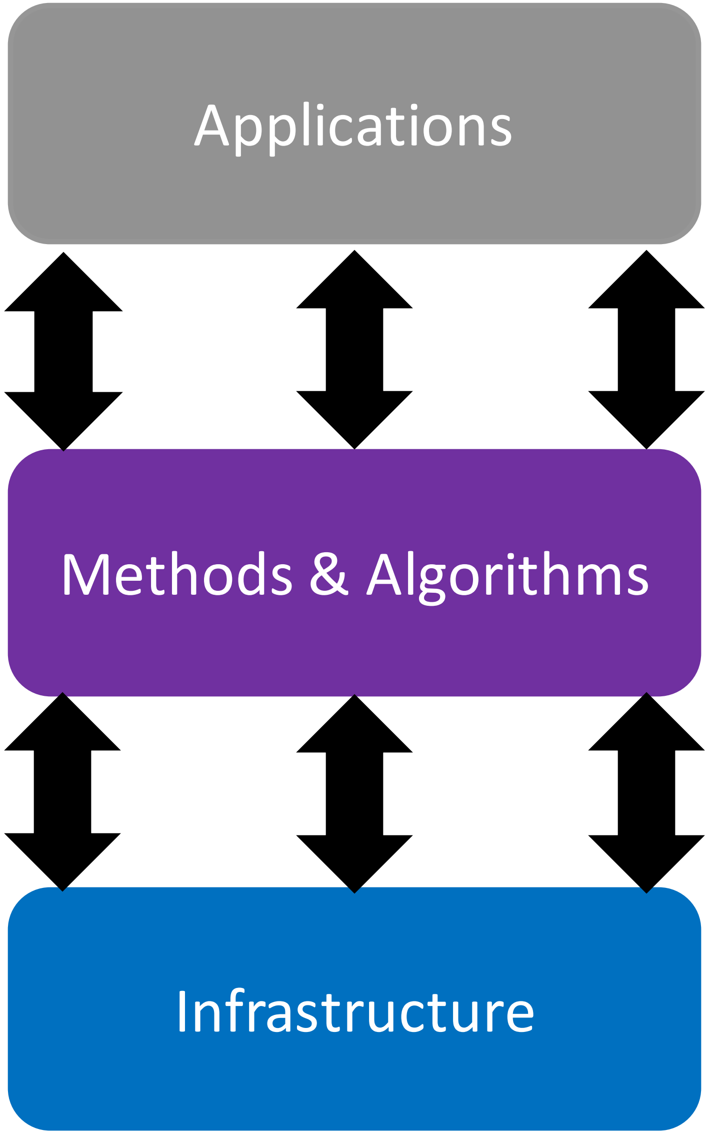
The Jefferson Lab Data Science Pillars

- Nuclear Physics (**NP**), High Energy Physics (**HEP**), Advanced Scientific Computing Research (**ASCR**), Basic Energy Sciences (**BES**)
- Health & Climate

Today's Focus

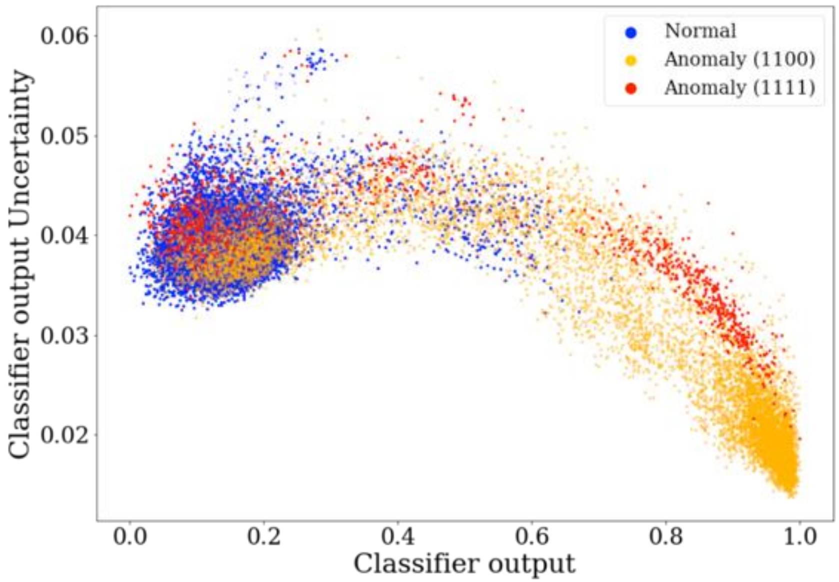
- AI based optimization & Controls
- **Explainability and Robustness**
- **Generative AI**
- **Scalable AI**

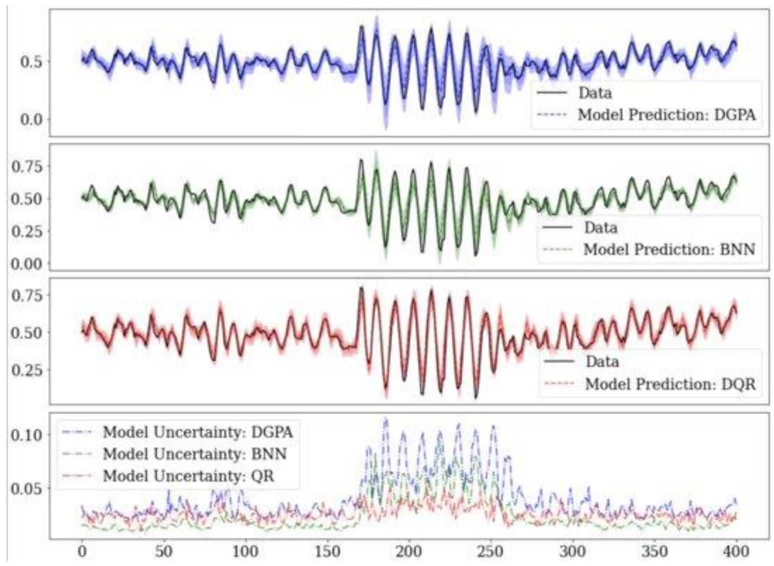
- **JLab Data Science Composable Workflow**
- JLab ML & Data Hub



Explainability and Robustness (1)

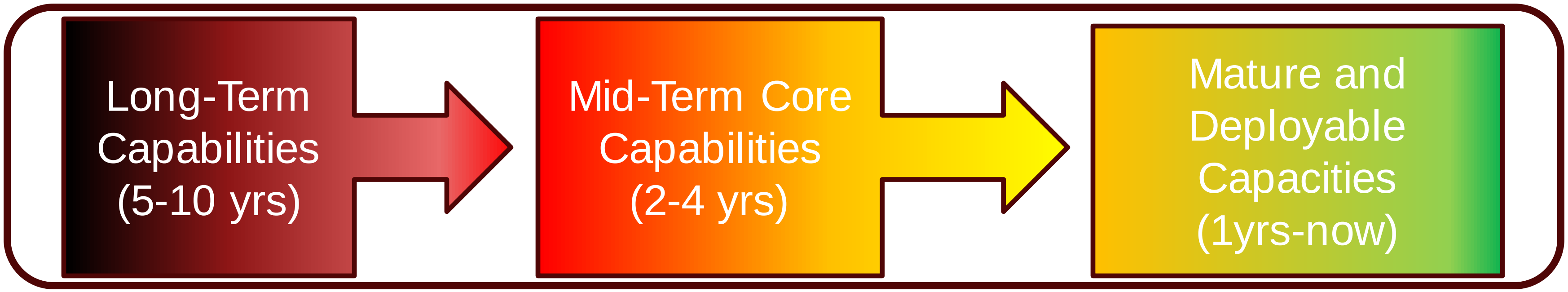
Tool Developed	Project	Collaborator
Uncertainty Quantification	Errant Beam	SNS w/ ORNL
Uncertainty Quantification	Data driven surrogate models	FNAL Booster
Uncertainty Quantification	Data driven regression for HVCM degradation capacitor models	SNS w/ ORNL
Loss Landscape	Conditional VAE models	SNS w. ORNL
Uncertainty Quantification	Norfolk flood surrogate models	ODU





Jefferson Lab

Explainability and Robustness (2)



Capability & Readiness

- **Mature & Deployable:** Integrated in majority of projects; Validation and migration of new algorithms into framework is ongoing
- **Mid-Term:** Working on new algorithms to handle boundary condition and scaling
- **Long-Term:** Working on understanding uncertainty quantification in generative AI

Uncertainty Quantification

A single model, without specific modifications, has no uncertainty!

What is often quoted: mean squared error, confusion matrix,.. ROC-Curve, …

- Deduced from data with known truth (or something close to it)
- No applicable to single prediction

Example: Mean Squared Error

$$\text{MSE}= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,\ y_i : \text{Known truth for } x_i,\ \hat{y}_i = \text{model}(x_i)$$

==> Gives an idea how good / bad the model performs on the entire data set

$$\hat{y}_i = \text{model}(x_i) \text{ holds NO information about uncertainty of } \hat{y}_i$$

Uncertainty Quantification

A single model, without specific modifications, has no uncertainty!

What is often quoted: mean squared error, confusion matrix,.. ROC-Curve, …

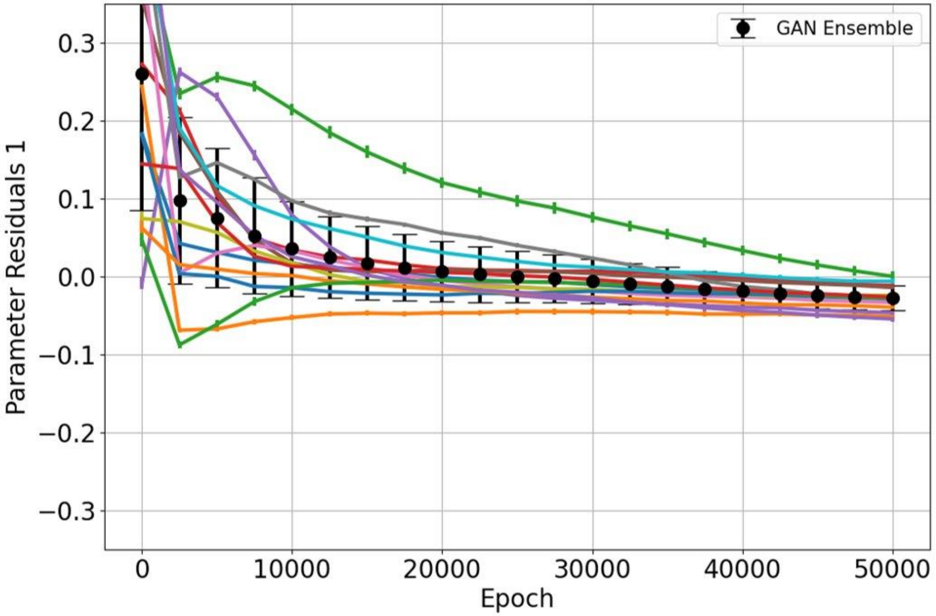
- Deduced from data with known truth (or something close to it)
- No applicable to single prediction

Common Techniques (just 2 out of many techniques)

1.) Ensemble: M models, independently trained on same data, but different initialization for internal parameters

$$\hat{y}_i = \frac{1}{M}\sum_{k=1}^{M} \mathrm{model}_k(x_i)$$

$$\sigma_i = \sqrt{\frac{1}{M}\sum_{k=1}^{M} (\mathrm{model}_k(x_i) - \hat{y}_i)^2}$$



Jefferson Lab

Uncertainty Quantification

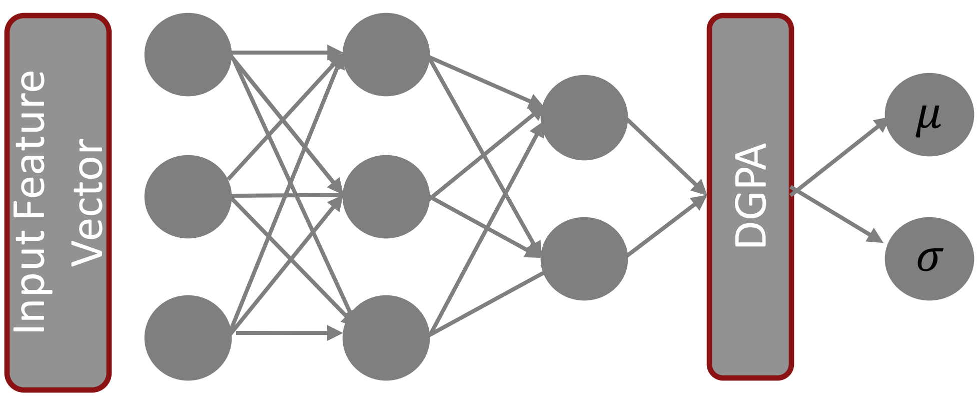
A single model, without specific modifications, has no uncertainty!

What is often quoted: mean squared error, confusion matrix,.. ROC-Curve, …

- Deduced from data with known truth (or something close to it)
- No applicable to single prediction

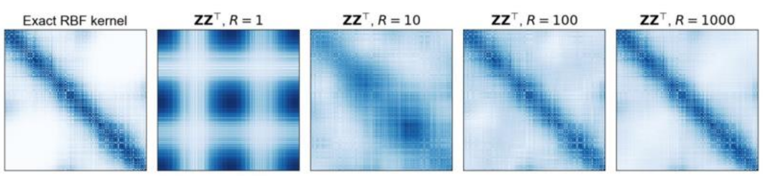
Common Techniques (just 2 out of many techniques)

2.) Deep Gaussian Process Approximation (DGPA): Approximate kernel k(x,y) to reduce computational cost. Model directly predicts uncertainty.



Allows to formulate uncertainties

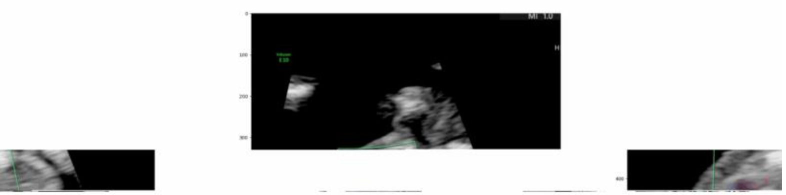
$$\boxed{k(x,y)} \approx z^T(x)z(y)$$



Generative AI (1)

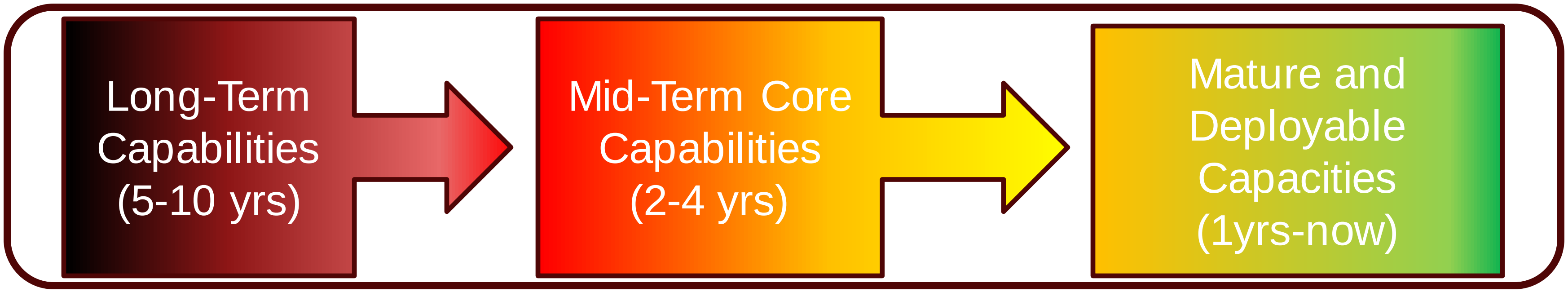
Tool Developed	Project	Collaborator
Anomaly detection and data generation	Errant Beam	SNS w/ ORNL
Anomaly detection and data generation	Analysis of ultrasound images	EVMS & ODU
Data driven parameter generation	Event-level analysis of deep inelastic scattering experiments	ANL / ODU / VTECH
Scientific generative AI	Event-level analysis of photoproduction data in CLAS	Hall B

Original



Generated

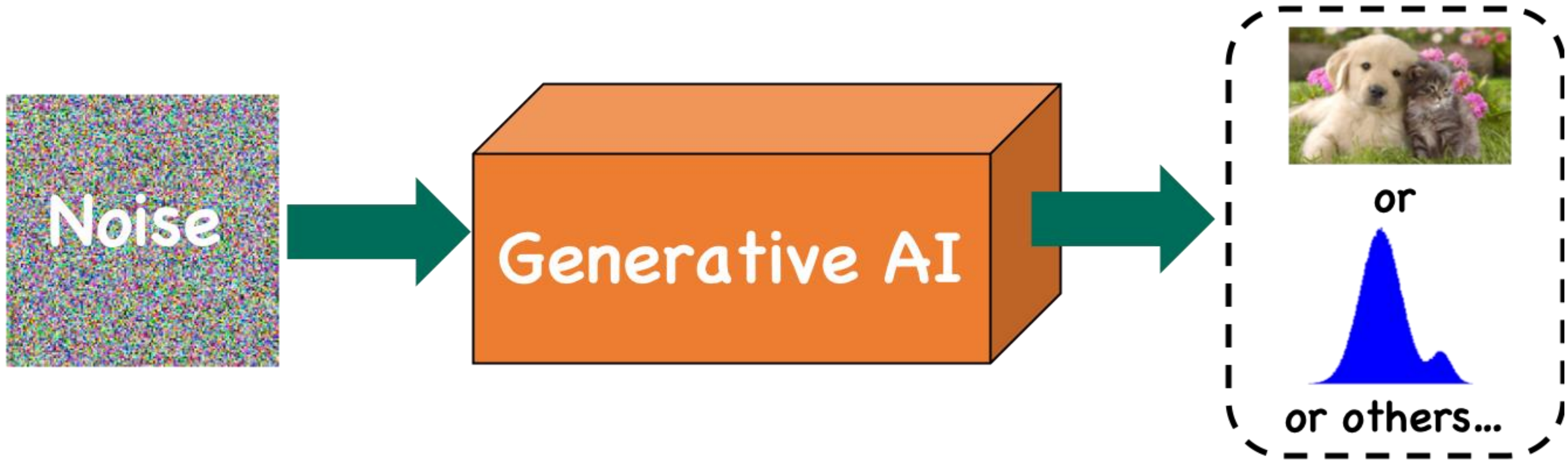
Generative AI (2)



Capability & Readiness

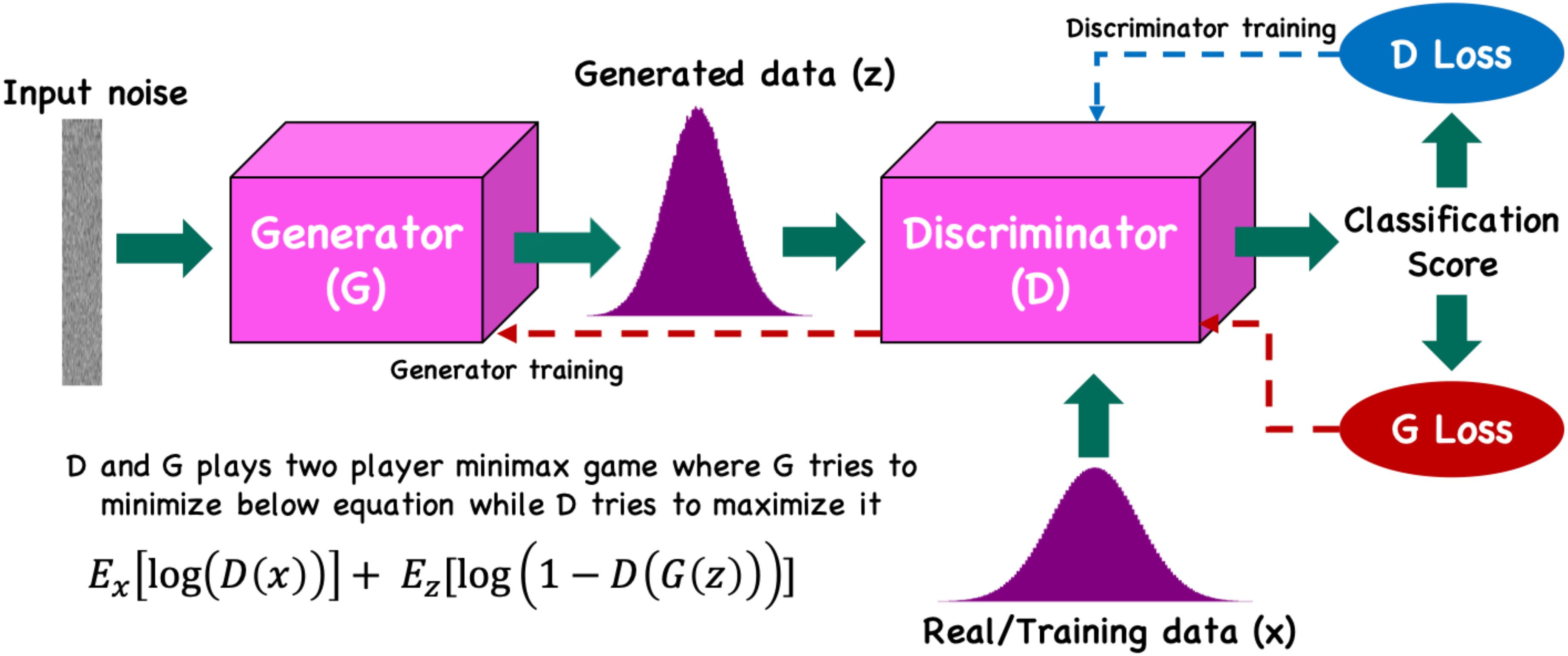
- **Mature & Deployable:** Basic techniques such as GAN, VAE available
- **Mid-Term:** Composable workflow for scientific generative AI
- **Long-Term:** Scalable, composable workflow for scientific generative I

Generative AI in a Nutshell



	Anomaly Detection	Data Generation	Training Difficulty	Image Quality & Diversity
Generative Adversarial Network (GAN)	⊘	✓		★★★★
Variational Autoencoder (VAE)	✓	✓		★★★
Diffusion Model	✓	✓		★★★★★

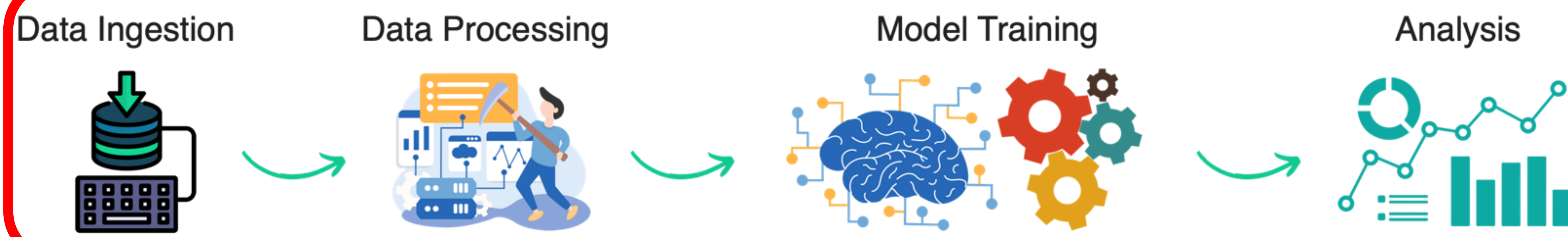
Generative Adverserial Networks (GANs)



$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

- Successfully utilized in multiple projects
- Always used in combination with diagnostic tools (e.g. gradient monitor, loss landscape,…)

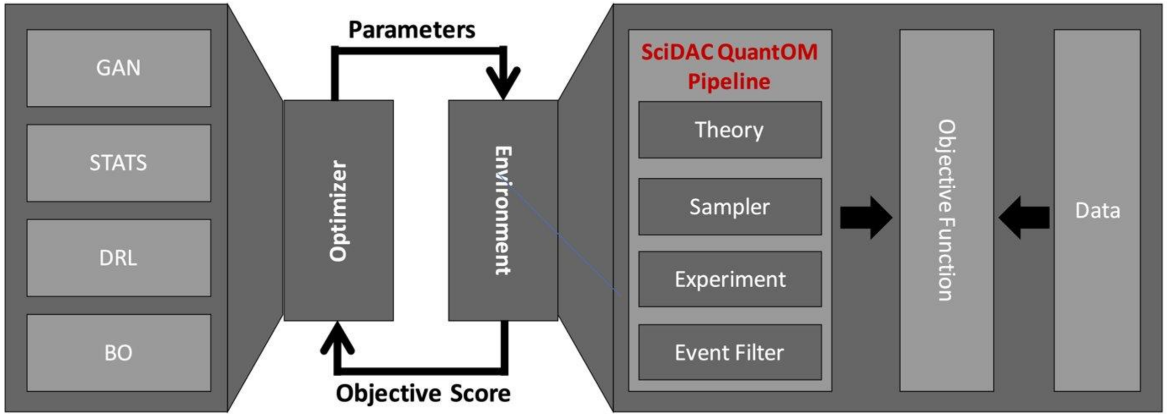
What is a Workflow and why should I use one?



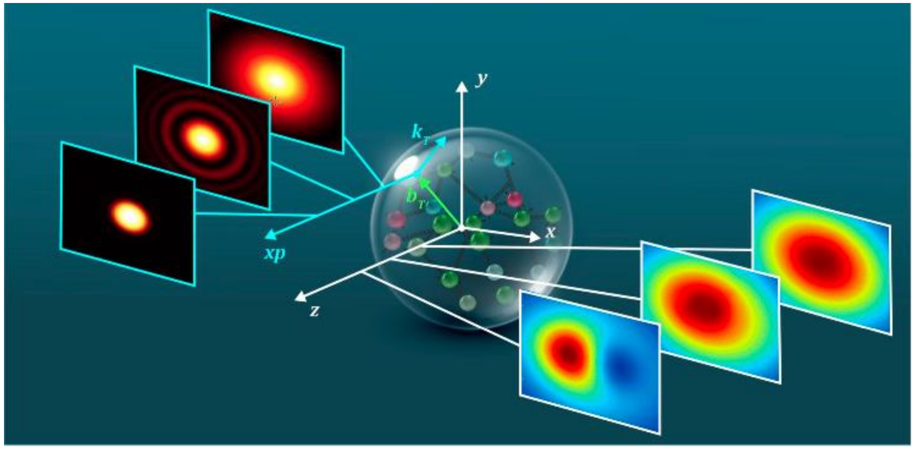
- **Workflow:** Chain of independent modules
- **Common denominator for every analysis**
- Replace / swap out modules, depending on analysis
- Key features
  - Work on modules independently --> Support collaborative efforts
  - Each module comes with a unit-test --> Easy debugging
  - Reproducibility and efficiency --> Everything runs from a configuration file
  - Profit from multiple ML / DL frameworks

TensorFlow Keras PyTorch scikit learn

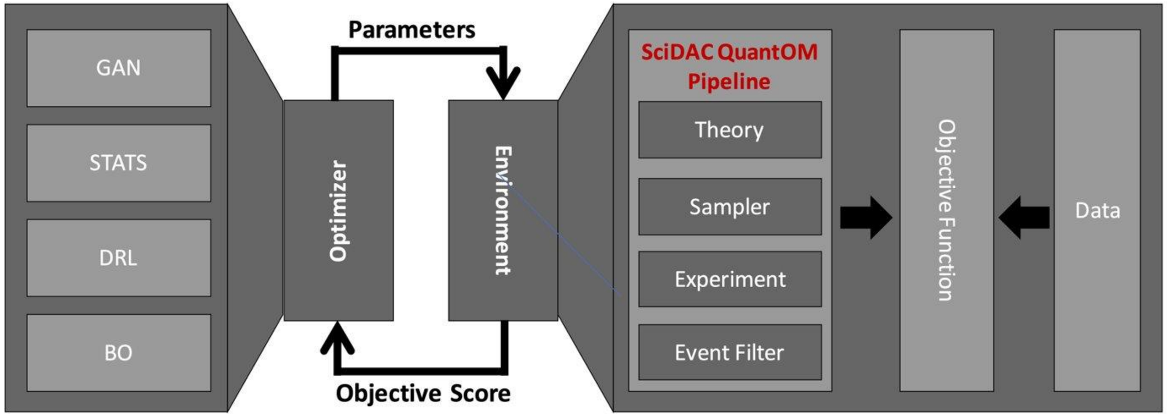
The Generative Inverse Problem Solver: GIPS

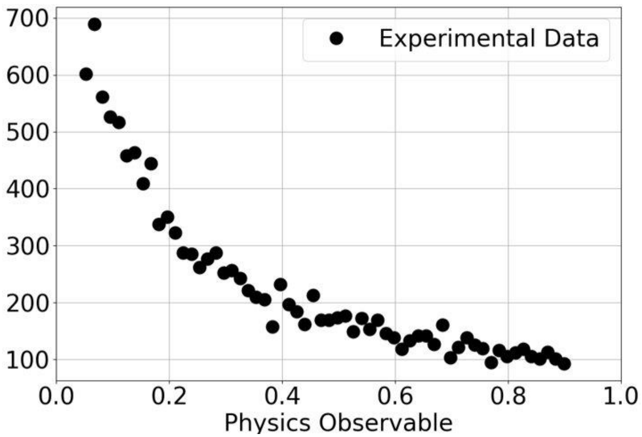


- Developed within the SciDAC QuantOM project
- Event-level toolkit for deep inelastic scattering data
- 3D imaging of the proton

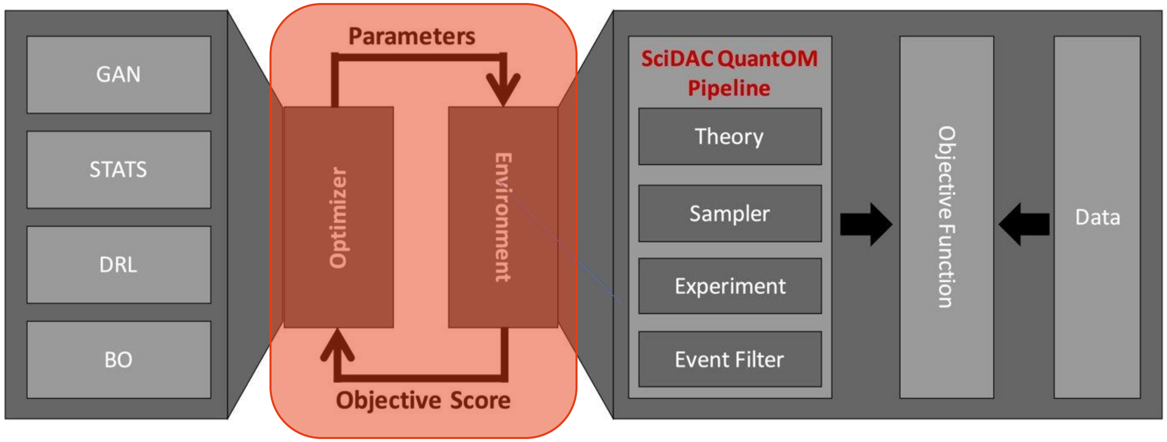


The Generative Inverse Problem Solver: GIPS

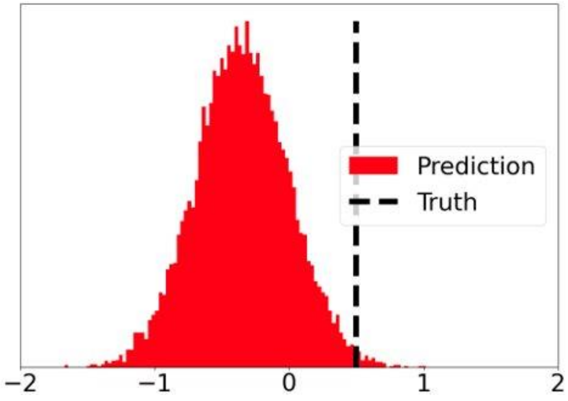


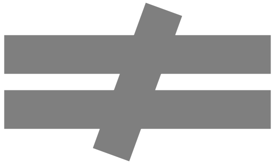


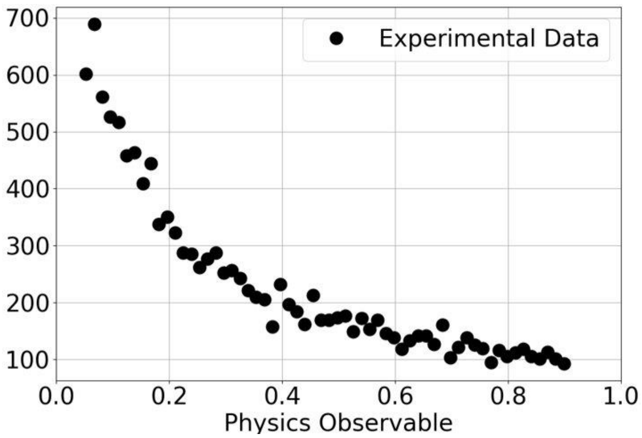
The Generative Inverse Problem Solver: GIPS



Iteration 0



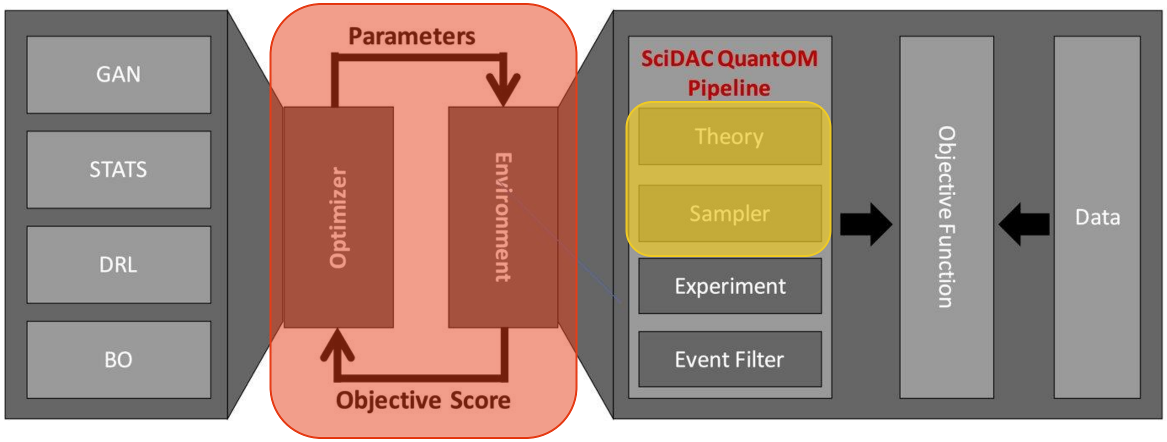




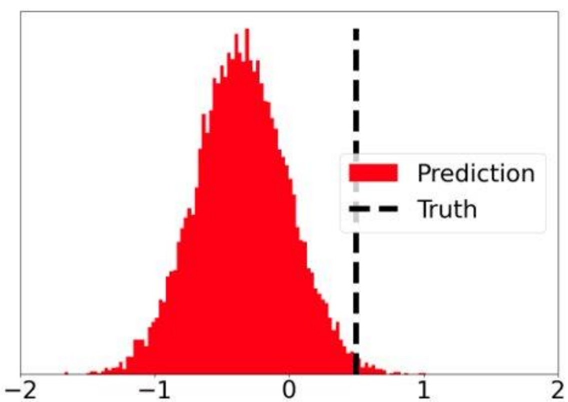
Parameters p

Can not directly compare

The Generative Inverse Problem Solver: GIPS

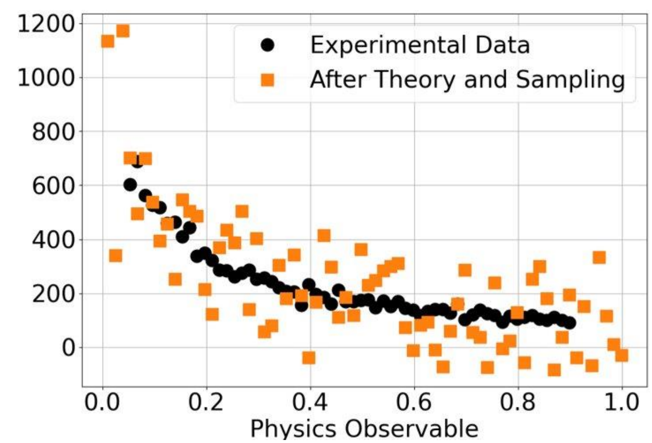


Iteration 0

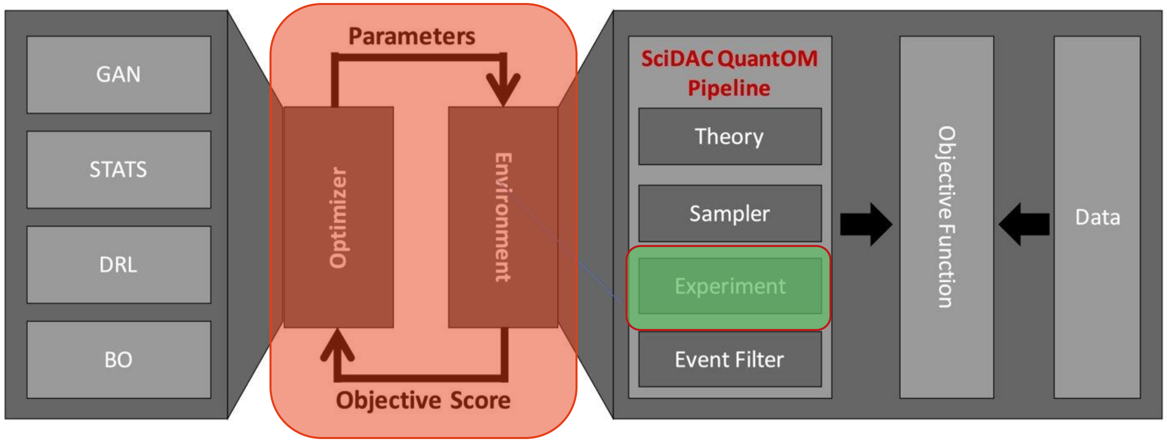


Parameters p

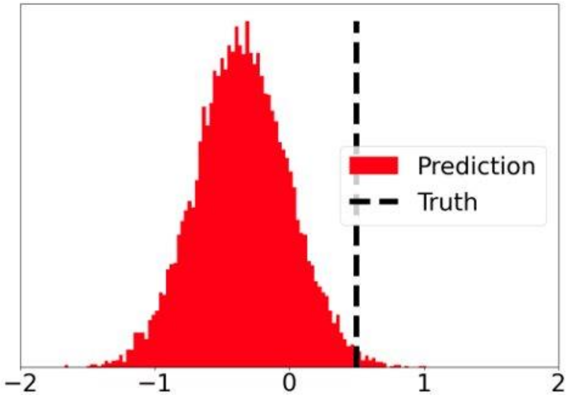
- Convert parameters to parton density functions (PDFs)
- Include higher order and radiate corrections
- Sample events from PDFs (e.g. via MCMC)



The Generative Inverse Problem Solver: GIPS

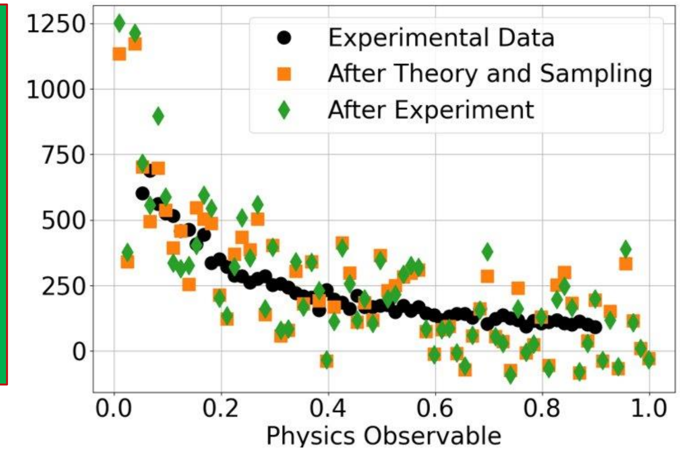


Iteration 0

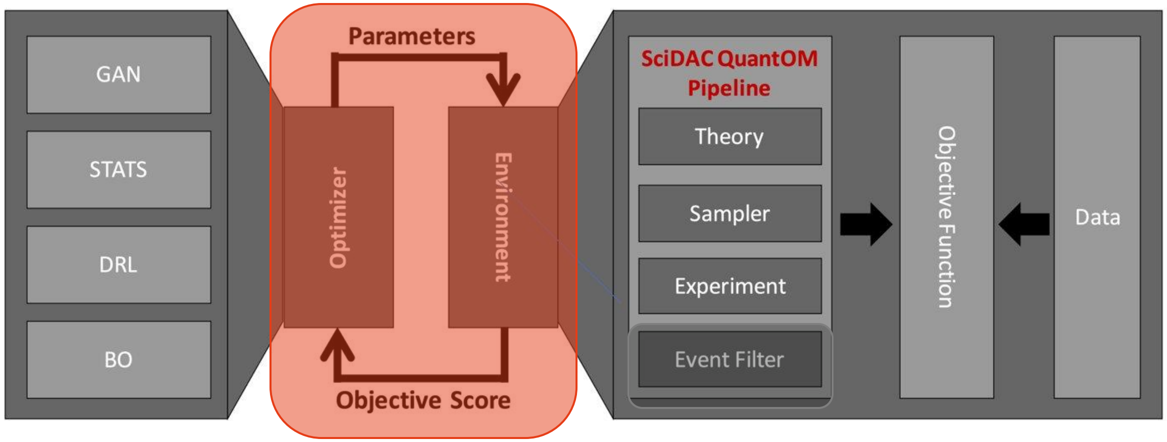


Parameters p

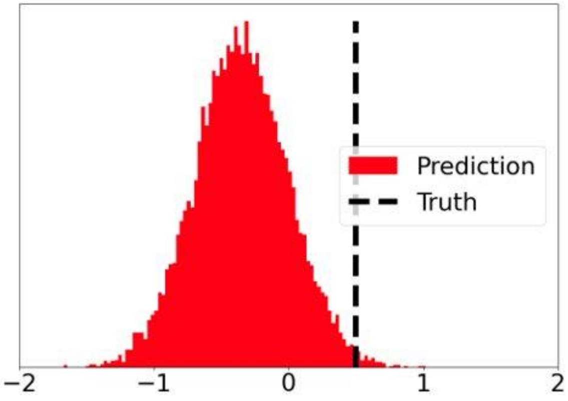
- Apply experimental effects (e.g. resolution, acceptance)
- Handle background contributions
- Use surrogate for detector



The Generative Inverse Problem Solver: GIPS

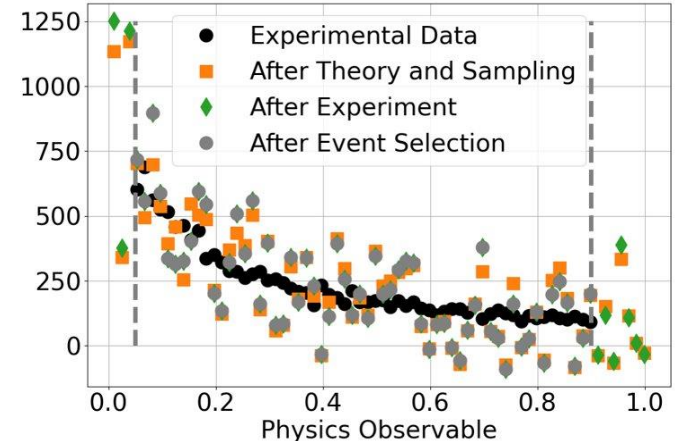


Iteration 0

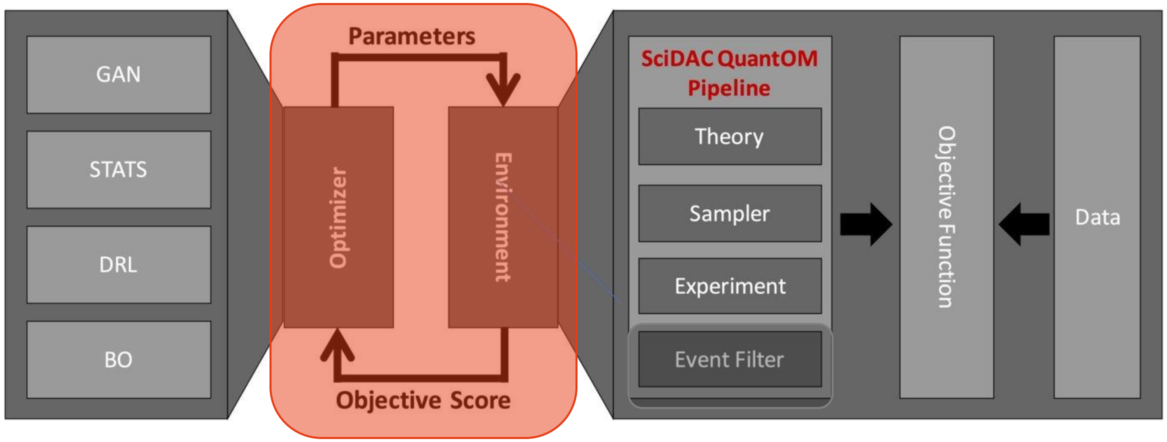


Parameters p

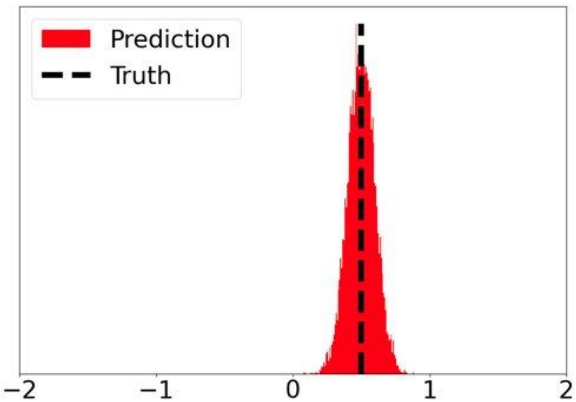
- Exclude un-physical data points
- Match experimental and synthetic data



The Generative Inverse Problem Solver: GIPS

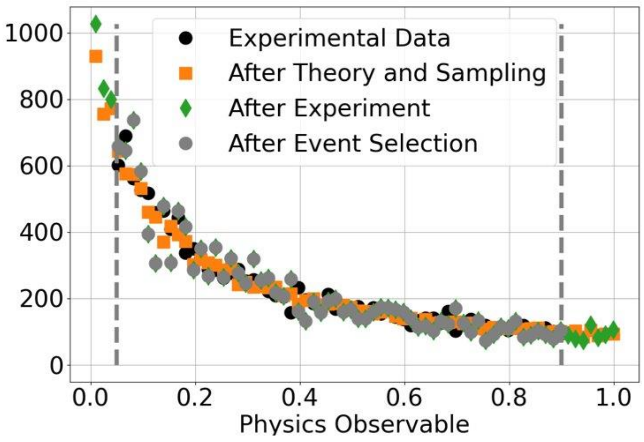


Iteration 100



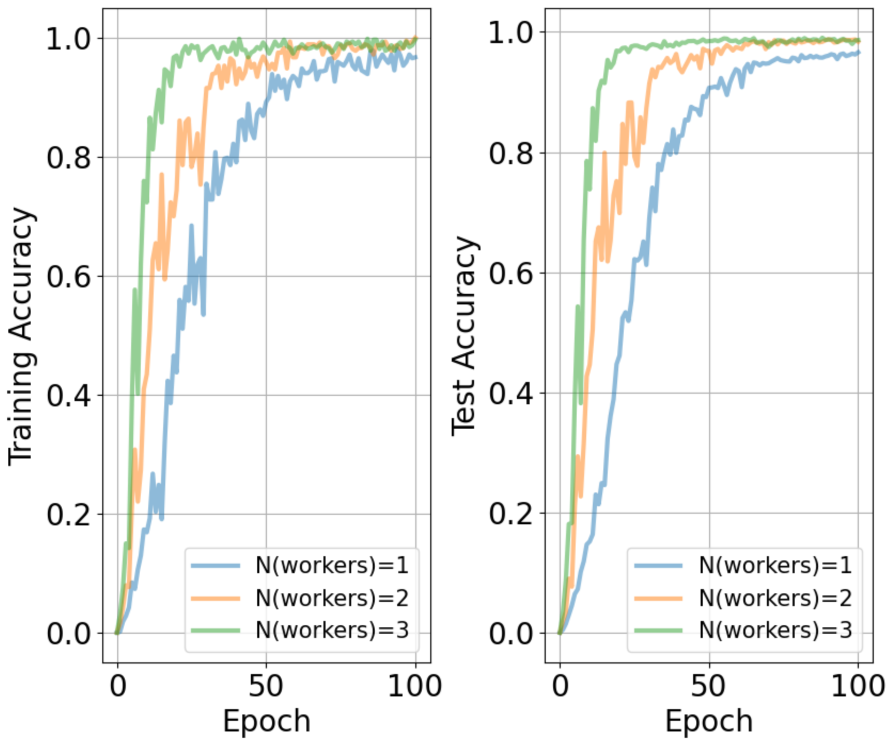
Parameters p

- Run workflow iteratively
- Use objective score to update optimizer



What is Scaling and do I need it?

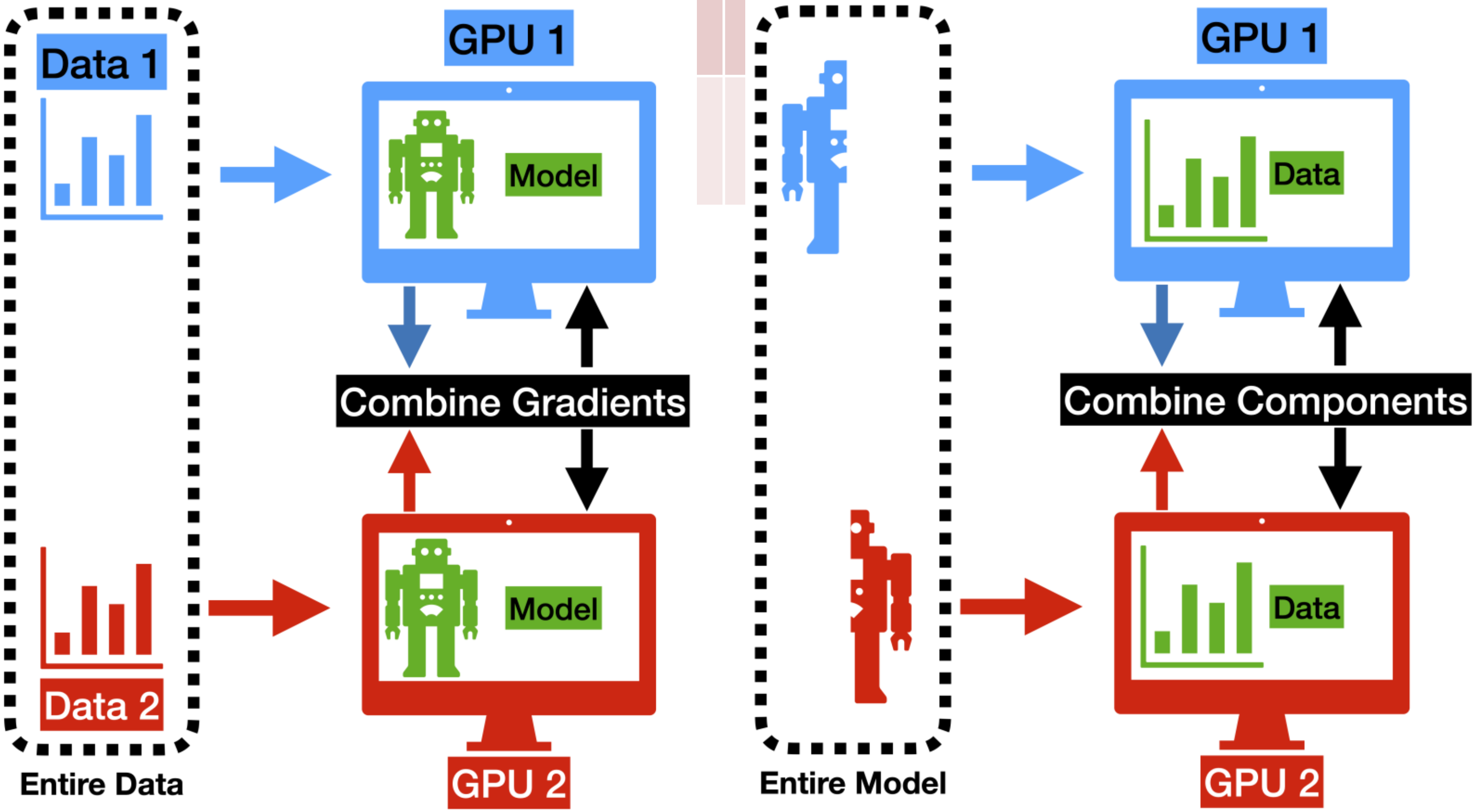
Data Format	Model Complexity (Number of trainable Parameters)
Digits	~1k - 100k
Images & Videos	~100k - 10000k
Text & Language	>> 10000k



- Depending on the model complexity, a single GPU is not suitable for training (Unless you are fine waiting months for your publication results)
- To speed up training time: Run your analysis across multiple GPUs
- **Scaling:** Total training time / Model performance vs. Number of GPUs
- **Example on the left:** MNIST Classifier trained on JLab GPUs, training times nearly identical for all runs

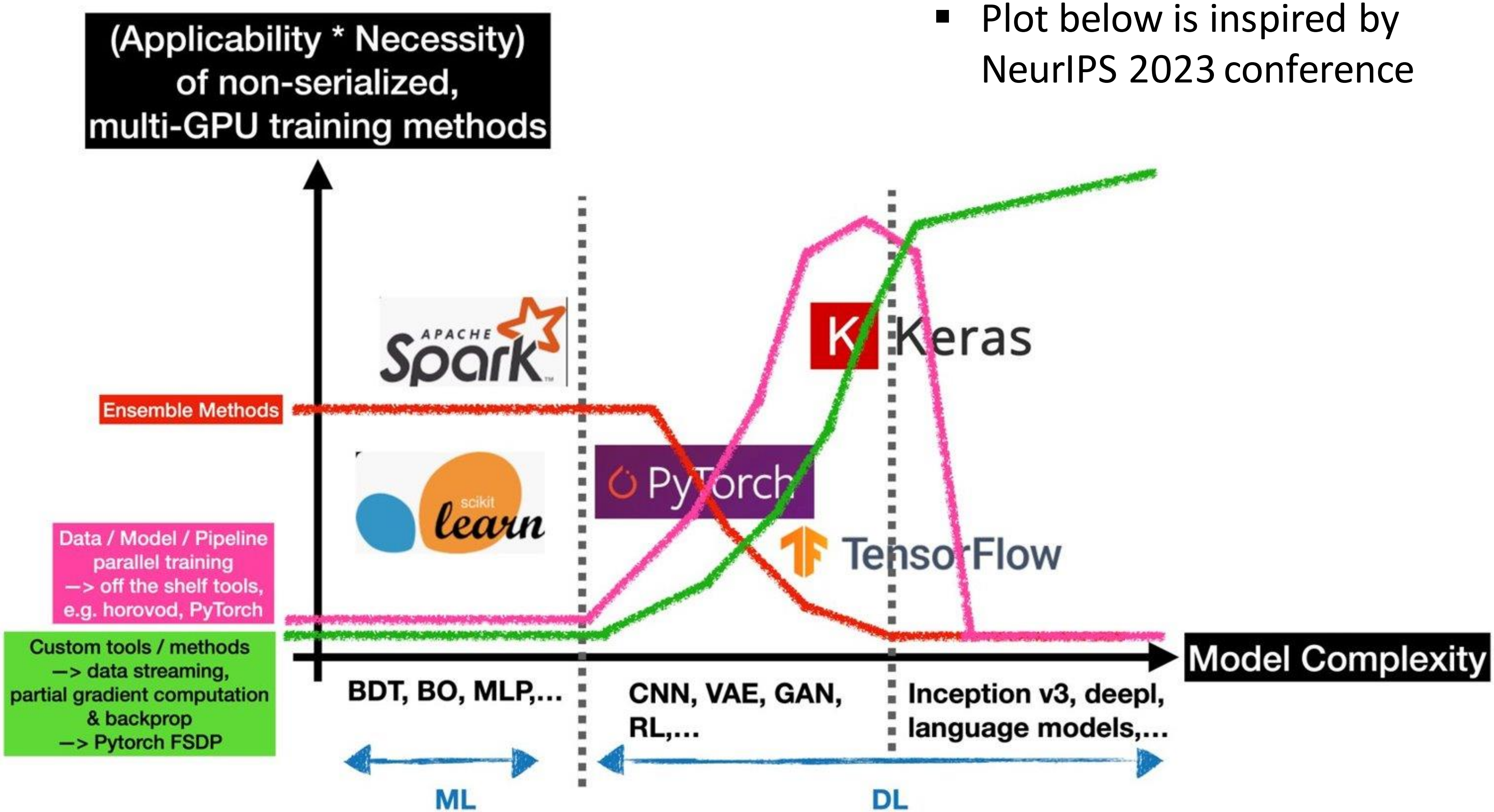
Basic Distributed Training Strategies

Training Method	Explanation
Data Parallel	Shard data across GPUs, each GPU sees full model --> Distribute gradients
Model Parallel	Shard model across GPUs, each GPU sees fraction of the model and full data

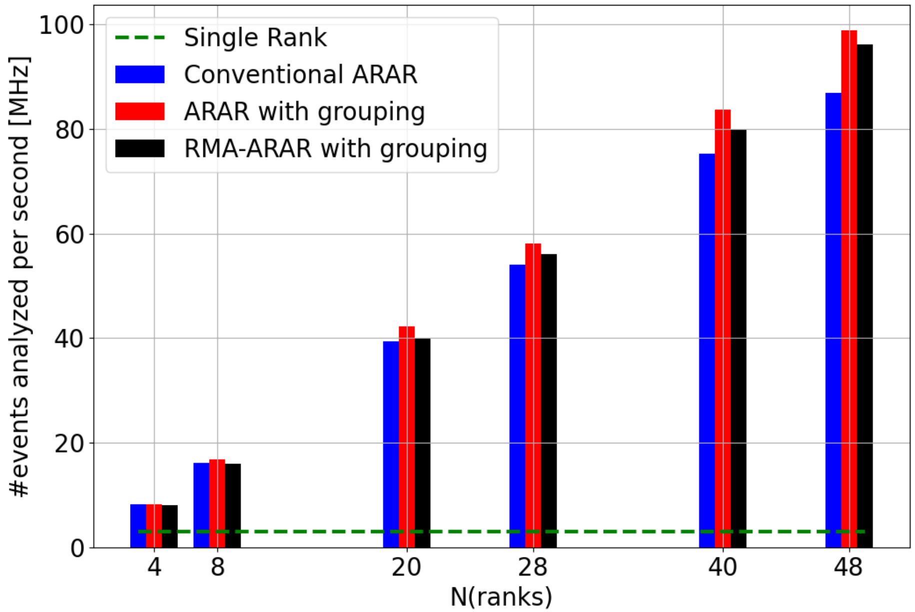


Distributed Training of Machine and Deep Learning Models

- **Applicability:** Can I do it?
- **Necessity:** Do I have to do it?
- Plot below is inspired by NeurIPS 2023 conference

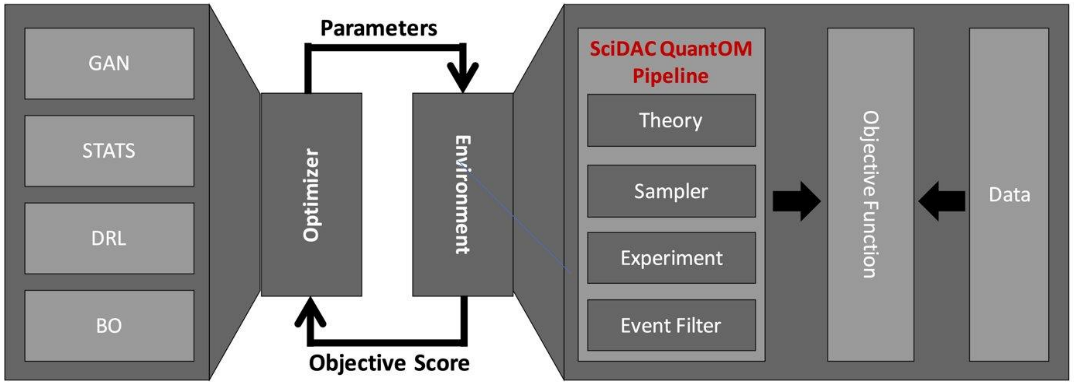


Scaling an entire Workflow

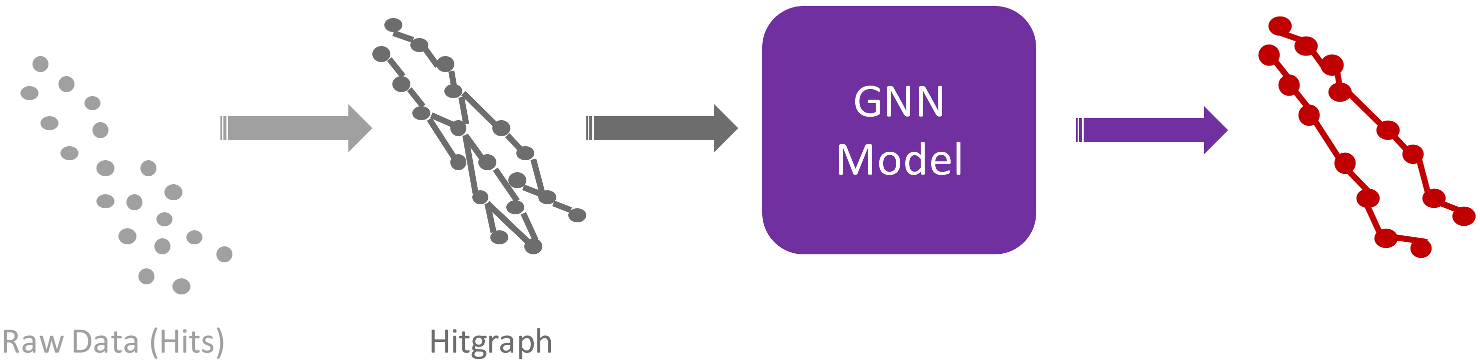


- Run QuantOM workflow on Polaris machine @ Argonne
- Utilize multiple GPUs to enhance analyzing power
- Test different methods for scaling the workflow
- Publication of current results in progress

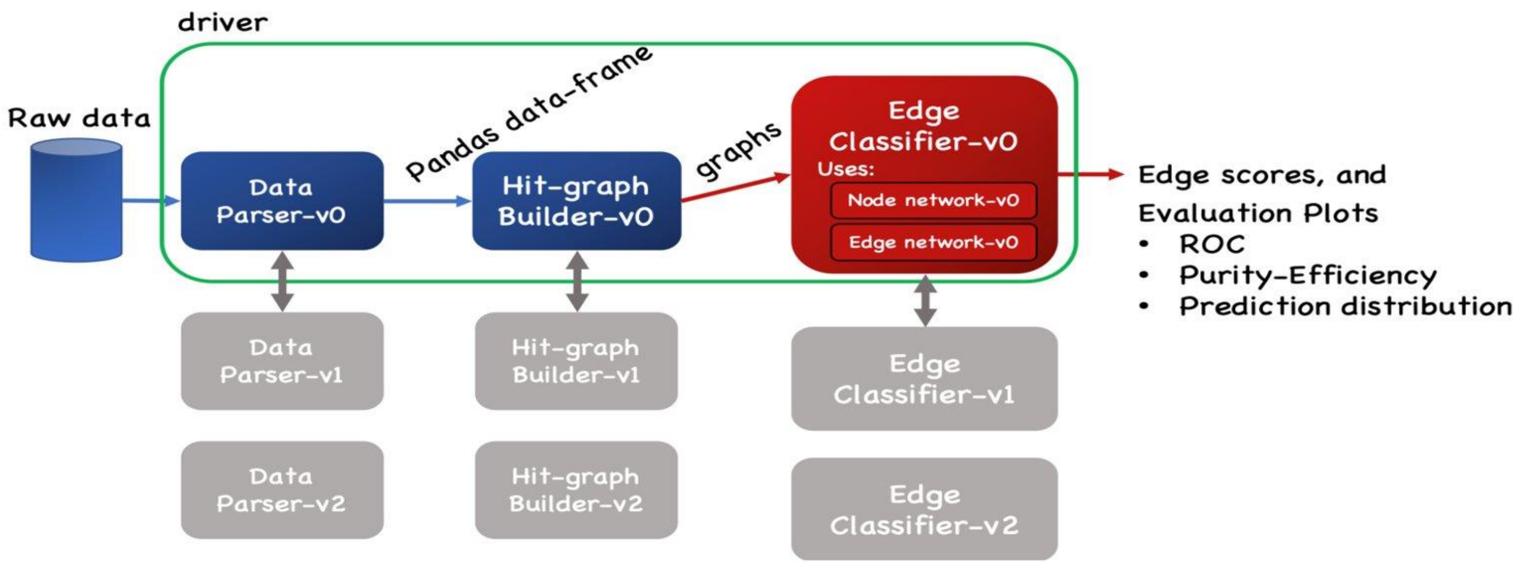
Polaris provides researchers with a powerful testbed to prepare applications and workloads for



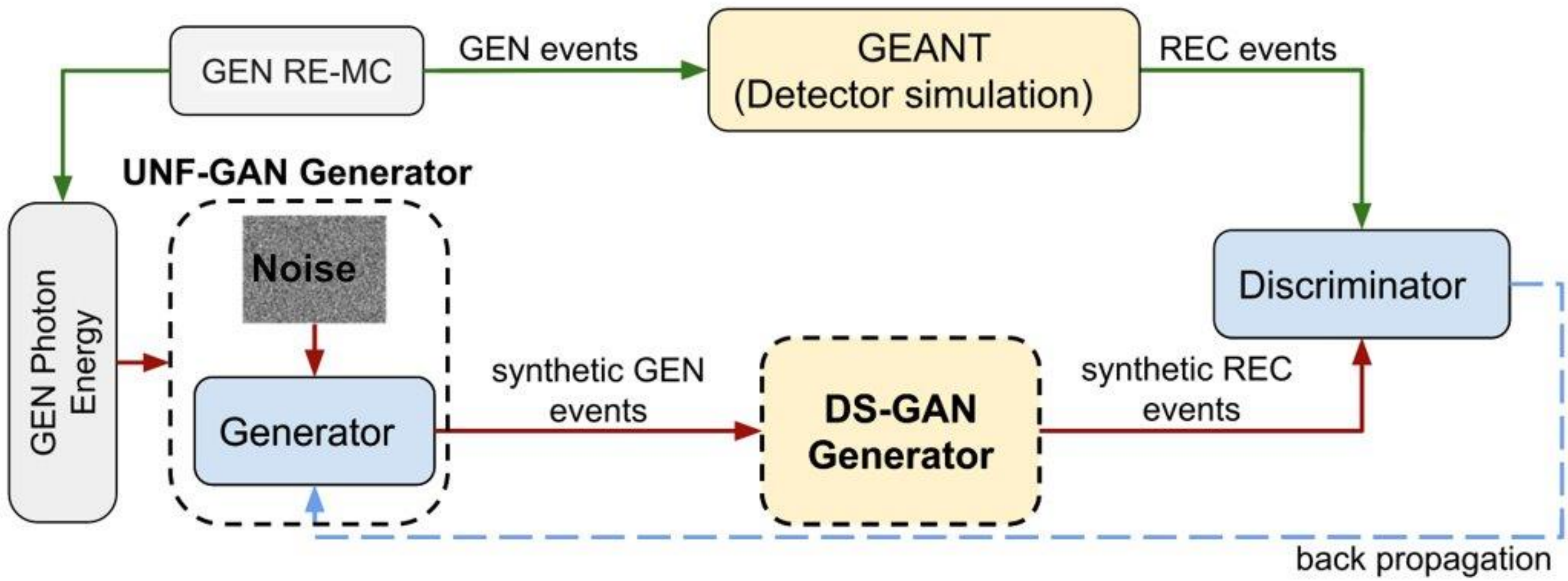
Charged Particle Tracking in Hall D



- Graph Neural Network (GNN) process raw detector hits
- Observe ~10 speed up compared to conventional method
- Deploy model of FPGAs



Generative Modeling Analysis in Hall B

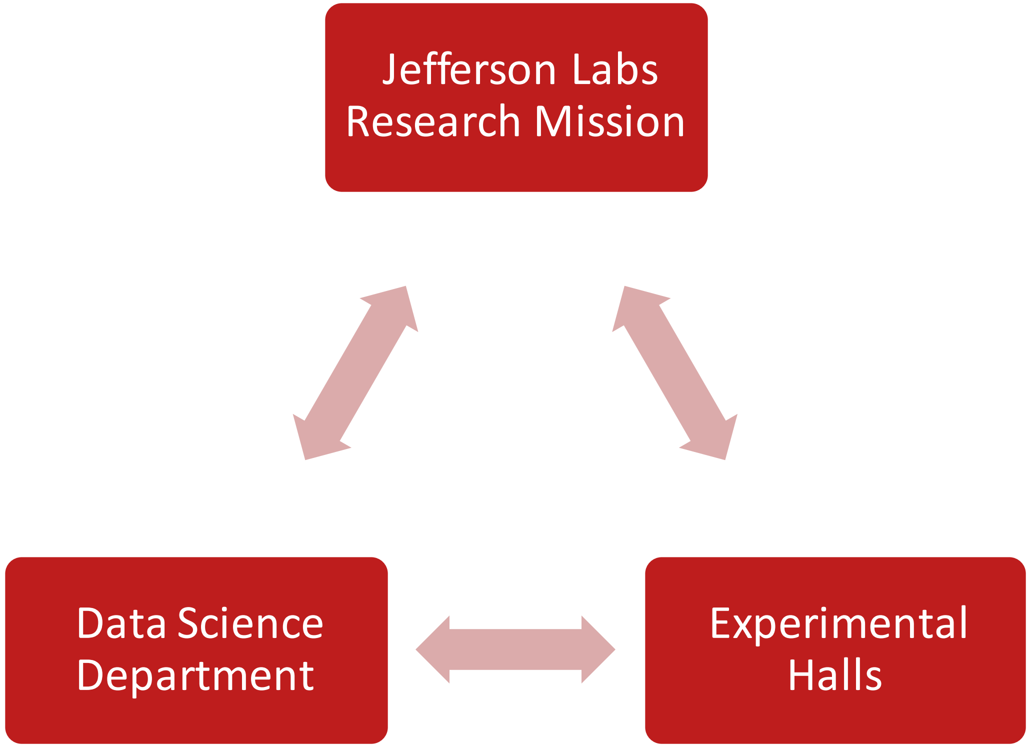


Plot taken from: Clas12 collab., Phys. Rev. D, 108, 094030 (2023)

- R & D already done by Y. Alanzi et al.
- Our contribution(s):
  - Implement existing code into generic, composable workflow
  - Add uncertainty quantification for GAN(s)
  - Provide scaling capability

```
Hall_B/AIDAPT/
├── data_parsers/
│   ├── __init__.py
│   └── aidapt_numpy_reader_v0.py
├── data_prep/
│   ├── __init__.py
│   ├── lab_variables_to_invariants.py
│   ├── numpy_minmax_scaler.py
│   └── numpy_standard_scaler.py
├── models/
│   ├── __init__.py
│   └── tf_mlp_gan_v0.py
└── utils/
    ├── config_utils.py
    └── math_utils.py
```

Let's Collaborate!



- Algorithmic Tools & Methods (e.g. uncertainty quantification, hyper parameter tuning,continuois learning...)
- Infrastructure for running scalable and robust workflows
- Experience and knowledge outside Nuclear Physics

- Expert knowledge (e.g. detectors, data formats, analyses,...)
- Interesting uses-cases (e.g. event-level fitting, tracking,...)
- Existing ML & DL tools

Please feel free to reach out

- Reached out to experimental hall leaders to initiate discussion(s) (I need to follow up on this)

- If you have questions, concerns, suggestions, please contact me: dlersch@jlab.org