Synergies between nuclear physics and AI/ML: A focused Quantom view

Malachi Schram
Head of the Data Science Department
On behalf of the research from the JLab Data Science Department
Newport News, Virginia
June 13th, 2023

TJNAF is managed by Jefferson Science
Associates for the US Department of Energy

Data Science at Jefferson Lab

Mission:

- Provide world-class data science solutions to advance research in nuclear physics by working with the subject matter experts at Jefferson Lab, partnering universities and Labs, and the Department of Energy.
- Provide world-class data science solutions to scientific applications relevant to the regional scientific community

Vision:

- Expand the capability and capacity of data science at JLab
- Create a collaborative 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

Current Portfolio

DOE Nuclear Physics:

- **Quantom SciDAC** (with ANL, VTech, ODU)
- Working with the experimental Halls (Tracking, etc.)
- Data Science contributing effort for AIEC (lead by EPSCI)

DOE Basic Energy Science:

- Machine Learning for Improving Accelerator and Target Performance (with ORNL)
- Collaborating with SLAC on application of ML-based controls for accelerators

DOE Advanced Scientific Computing Research:

- Data-Driven Decision Control for Complex Systems (with PNNL, ORNL, UC)

Non-DOE:

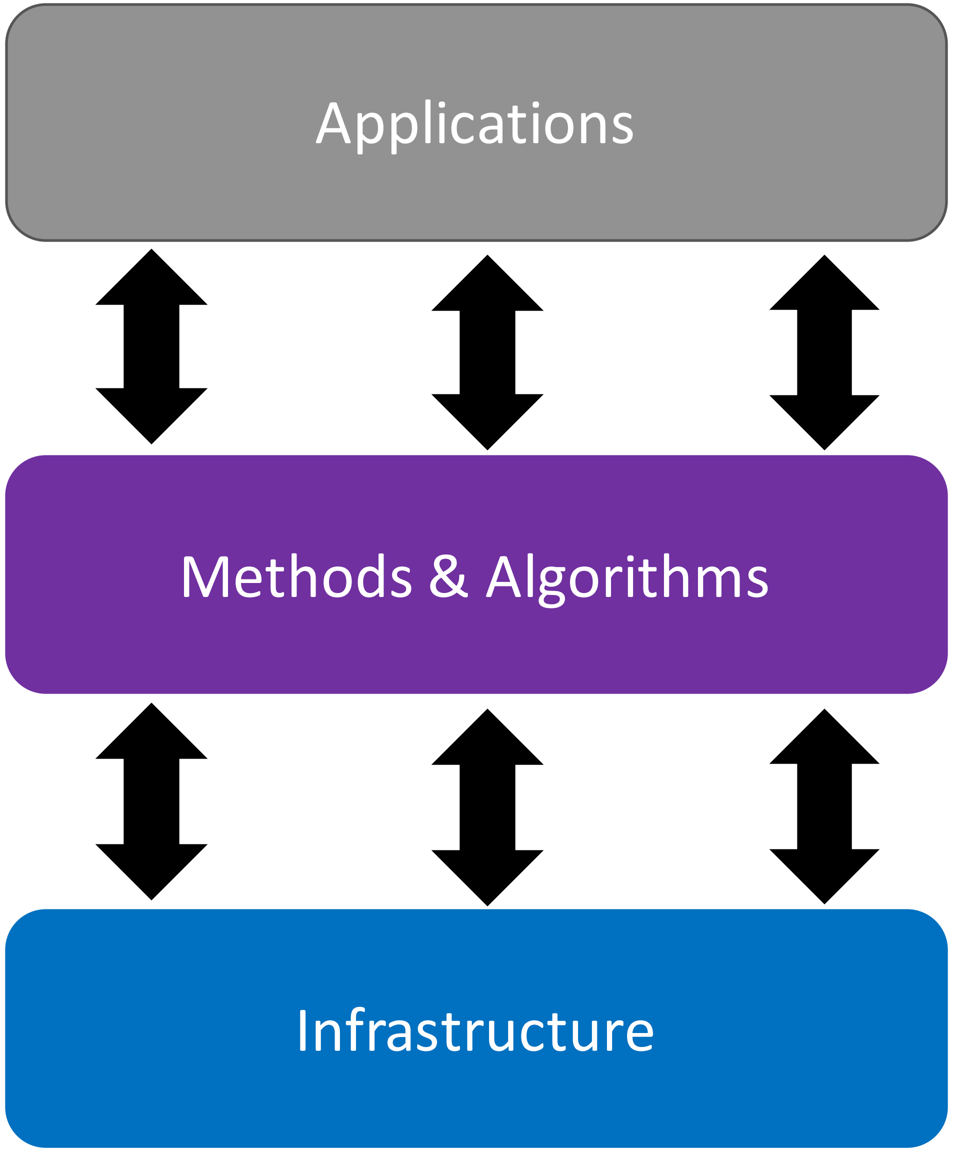
- Hampton Roads Digital Twin (with ODU)

Laboratory Directed Research & Development (LDRD):

- Multi-objective Optimization of Heat Load and Trip Rates in CEBAF (FY22)
- Adaptive Strategies for Optimal Computing Availability (FY23)

JLab Data Science Pillars

- **Applications:**
  - Nuclear Physics
  - Advanced Scientific Computing
  - Health & Climate
- **Focused Methods & Algorithms:**
  - Uncertainty Quantification
  - Interpretability and Explainability
  - Design & Control
- **Infrastructure:**
  - JLab ML & Data Hub
  - JLab Data Science software



DOE ASCR - BRN for SciML

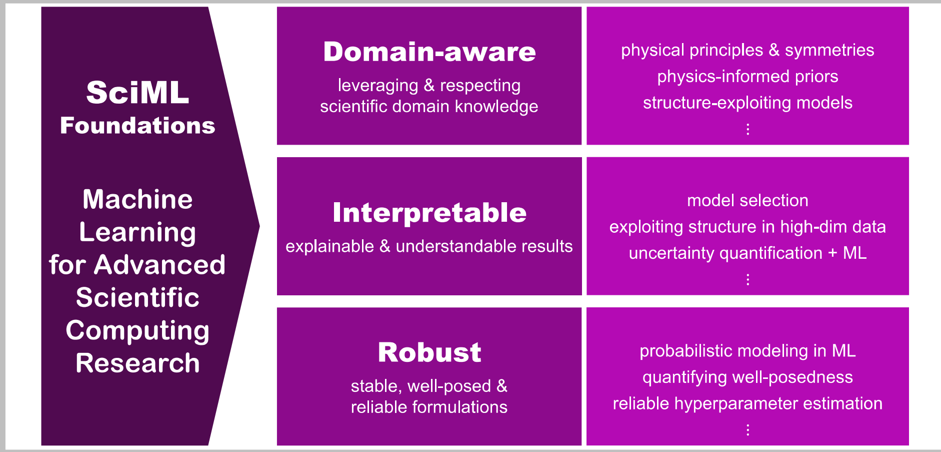


Figure 1: Foundational research themes of SciML must tackle the challenges of creating domain-aware, interpretable, and robust ML formulations, methods, and algorithms.

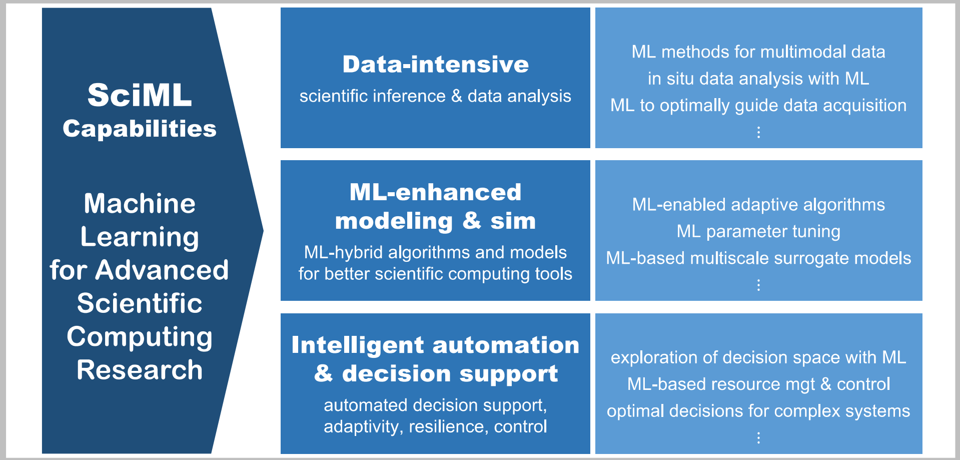
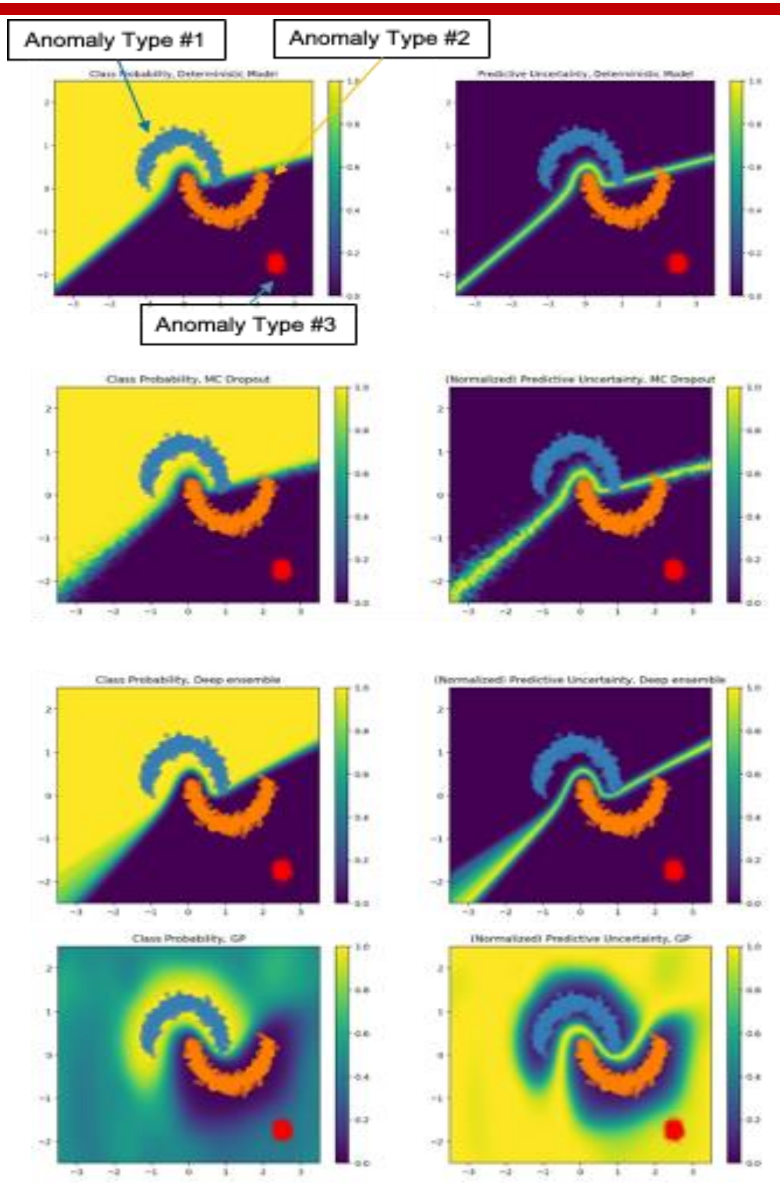


Figure 2: Opportunities for SciML impact arise in scientific inference and data analysis; in ML-enhanced modeling and simulation; in intelligent automation and decision support; and in related applications.

Uncertainty Quantification for ML

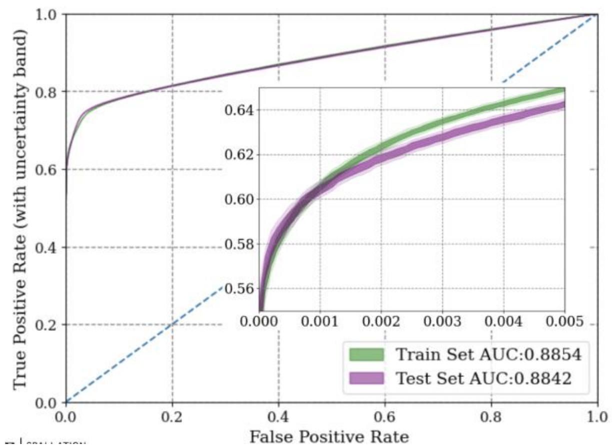
Develop methods that include uncertainty estimates in machine learning models

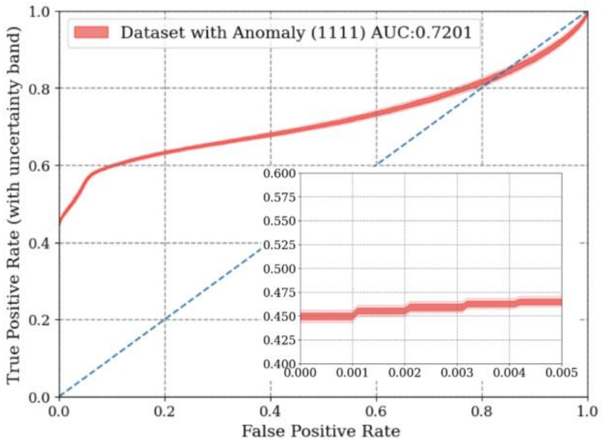
- <u>Applications</u>:
  - Data driven ML-based surrogate models
  - Real time controller
  - Anomaly detections
- <u>Requirements</u>:
  - Out-of-distribution uncertainties
  - Auto-calibration
  - Single inference
- <u>Hardware considerations:</u>
  - Memory
  - Inference time
  - Performance trade-off due to approximations

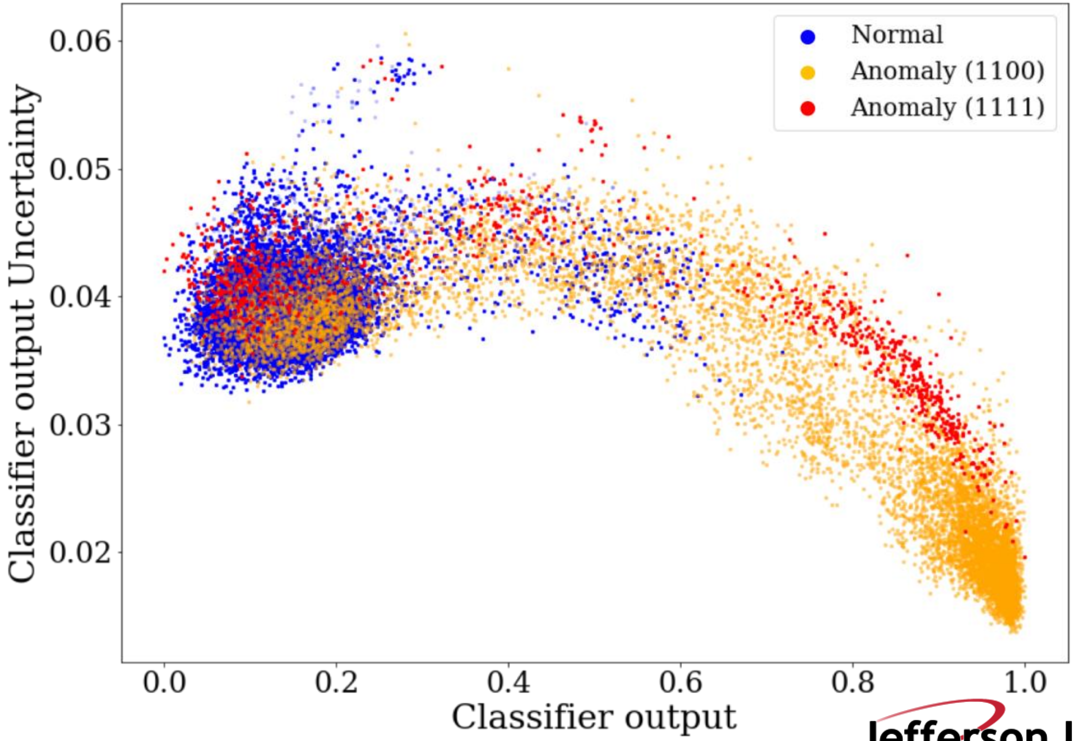


Uncertainty Aware Siamese Model (“Classification”)

- We enhanced our models by adding GP approximation layer which provides the uncertainty estimate
- Results from similarity model showed a ~4x improvement in performance over previously published results, it is also much better than a vanilla Auto-encoder
- The ROC curves show true fault detection rate above 60% while keeping the false alarms below 0.5% (not optimized)
- We introduced an out-of-domain anomaly, labelled 1111 (red), the UQ-based model performed similar in classifying the anomalies and indicated high uncertainty (as expected)

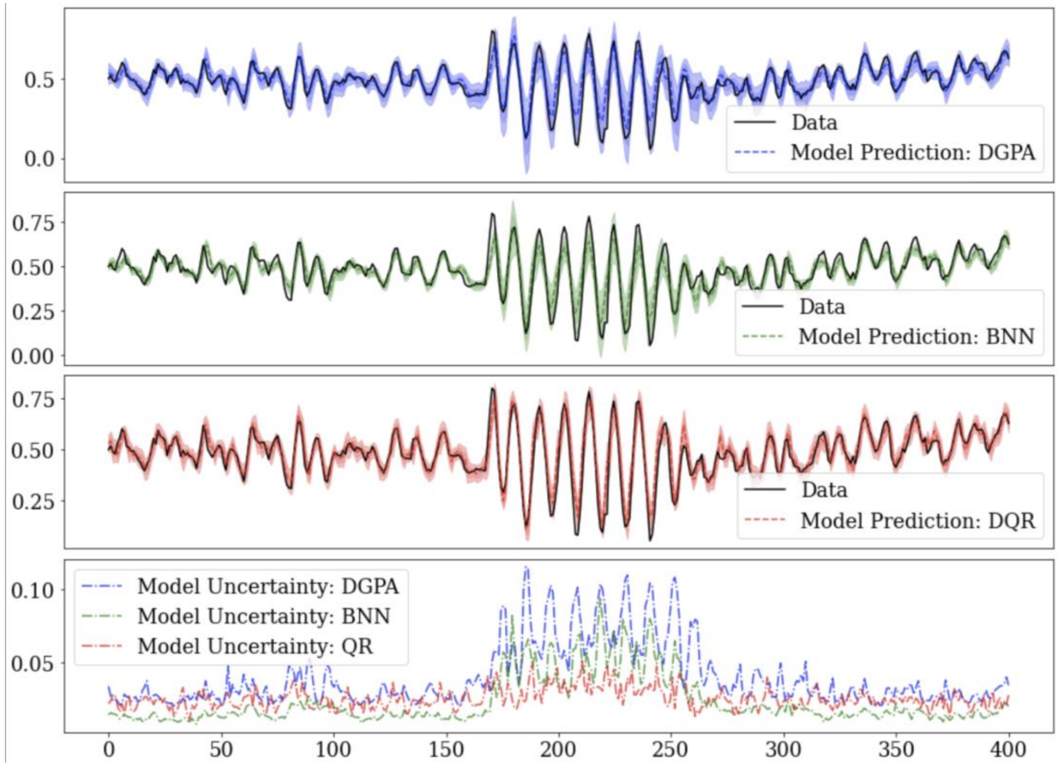


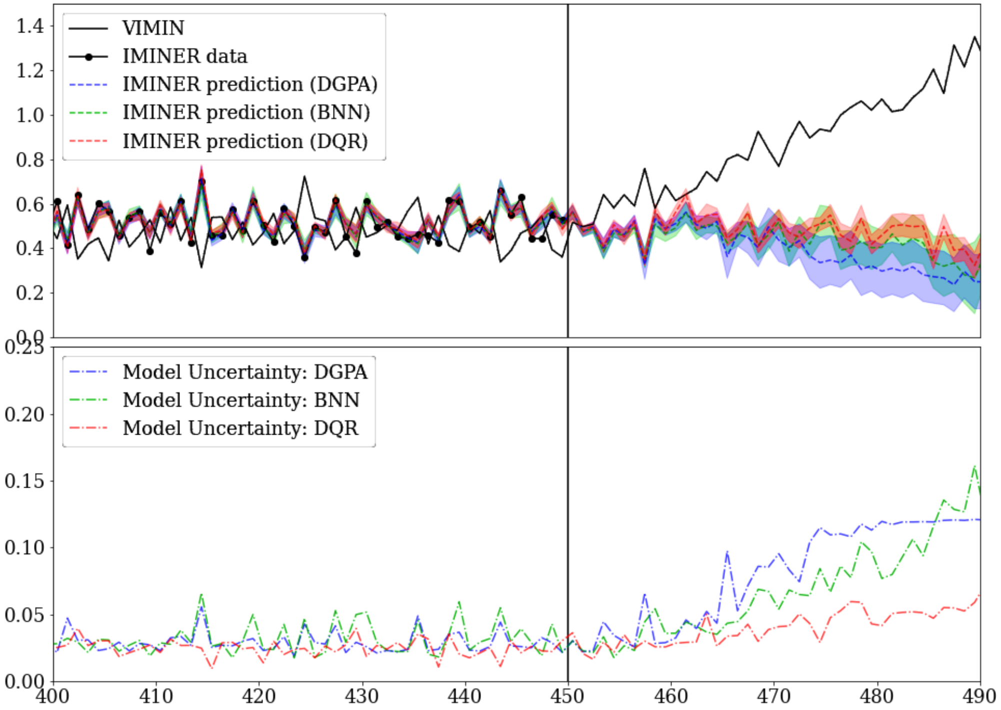




Data Driven UQ ML-based Surrogate Models (Regression)

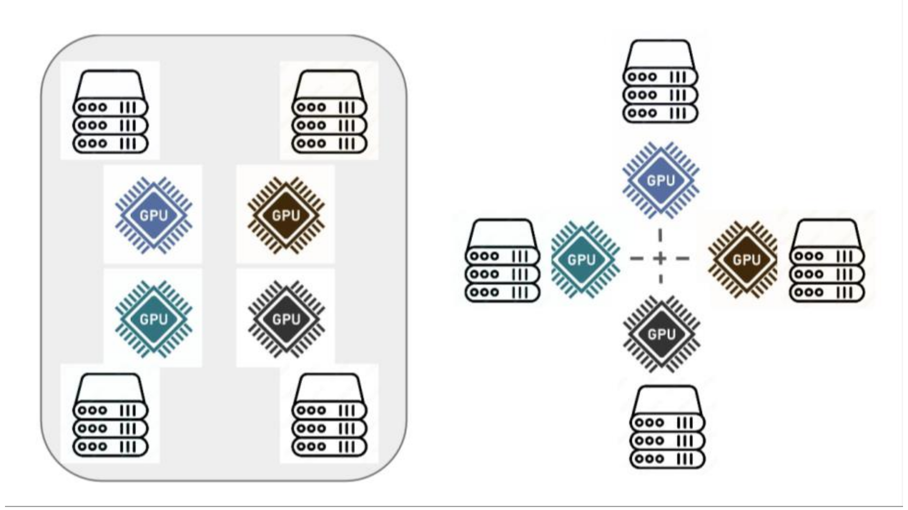
- Compare different techniques: DQR, BNN, DGPA
- DQR models have great performance for training distribution but not for OOD
- BNN models do a better job to estimate OOD
- DGPA models are distance aware by design resulting in better OOD estimation



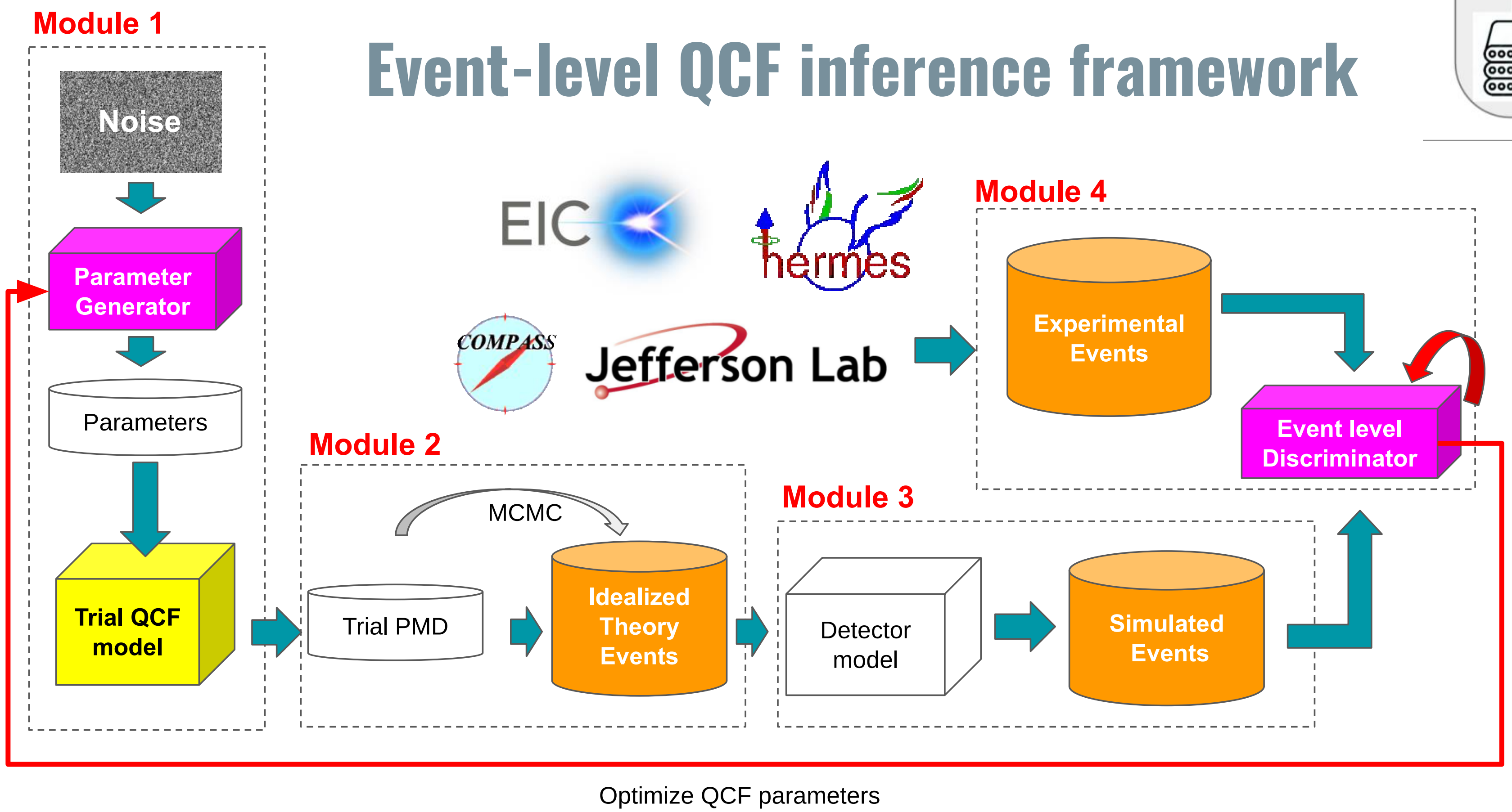


SCIDAC QUANTOM

- The goal is to extract a quark and gluon tomography of nuclei and answer important questions on the nature of visible matter at the femtoscale
- Develop modular components to dynamically compose workflows
- Multiple AI/ML components that need to scale to LCFs



Event-level QCF inference framework



TensorFlow

PyTorch

HOROVOD

Jefferson Lab

FRAMEWORK

- Developed a common and modular framework
- Includes:
  - Core base classes
  - Proxy App, GPDs Theory
  - Experimental (filter, detector, etc.)
  - GAN workflows

cfg	Updated visualization tools
core	Implemented visulation tools and helper scripts
demos	Implement requested changes and information capture
discriminator_module	fix useBias bug in disc
eventselection_module	Implemented visulation tools and helper scripts
expdata_module	Update with code from pub_demo that appeared to be missing.
experimental_module	Add first readme version for the experimental module
generator_module	configurable-N events per parameter set
sample_data	Add sample data for test use cases.
theory_module	Updated visualization tools
utils	Updated visualization tools
workflow	Updated visualization tools

master · 36 branches · 0 tags · Go to file · Add file · Code

schr476 Include MacOS env builder (testing). 8e78e1f 2 weeks ago · 111 commits

tomography_toolkit_dev	Updated visualization tools	last month
utests	Implement requested changes and information capture	last month
.gitignore	add scripts dir to ignore	last month
HOWTO_CONTRIBUTE.md	Write documentation file: HOWTO_CONTRIBUTE.md	last month
README.md	Merge branch 'master' into 80-update-conda-requirements-file	last month
env-metal-arm64.yaml	Include MacOS env builder (testing).	2 weeks ago
env.yaml	Update the conda setup file and the requirements.txt	last month
requirements.txt	Update the conda setup file and the requirements.txt	last month
setup.py	Implemented visulation tools and helper scripts	last month

README.md

SciDAC Quantom Collab Dev [version 0.1]

Dev repository for SciDAC Quantom project

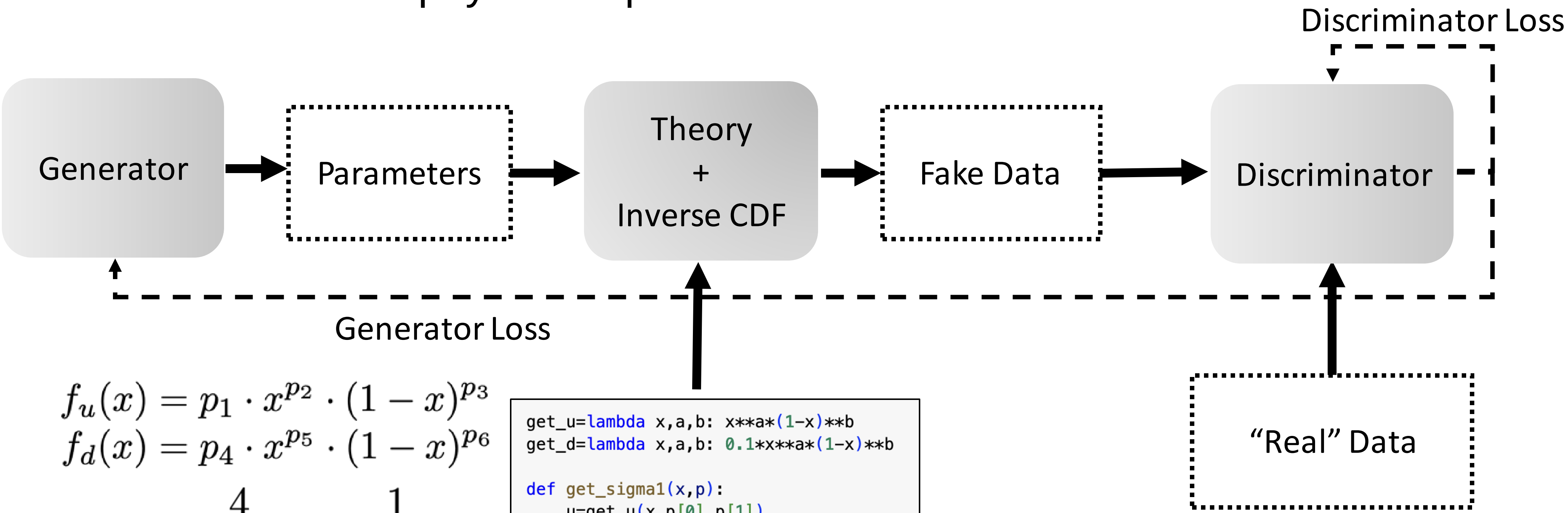
Code contribution proceedure

1. Create an issue or access assigned issue
2. Create a new branch using the issue id
3. When your code is complete and updated in the branch then make a pull request

More details are found here: HOWTO_CONTRIBUTE.md

PROXY APP (NOT REAL PHYSICS)

- Goal: Recover the parameter of the QCFs for u and d contribution
- WARNING: this has no physical equivalent

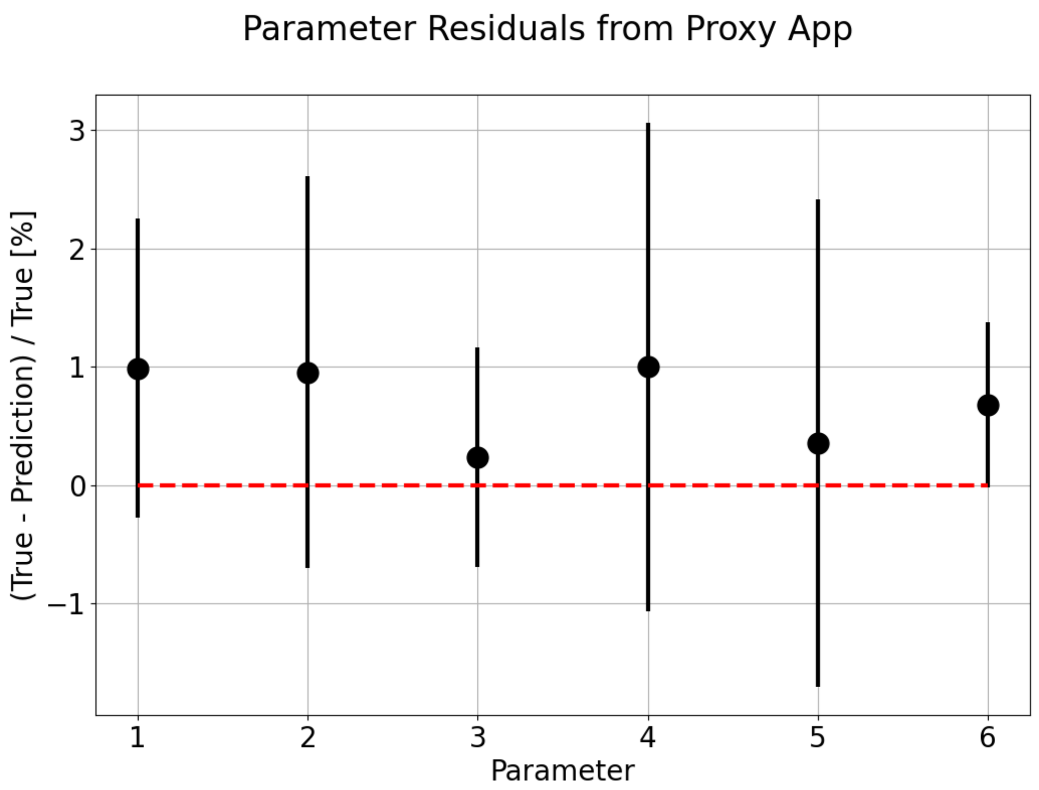


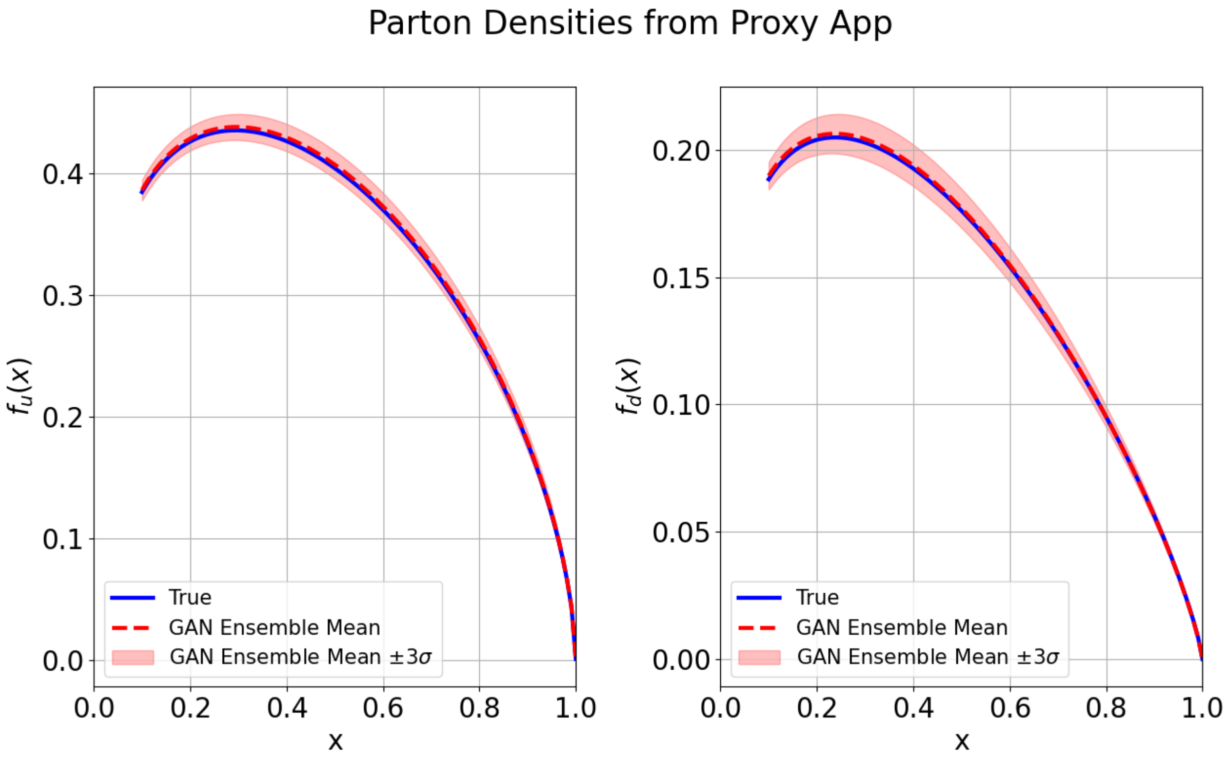
$$f_u(x) = p_1 \cdot x^{p_2} \cdot (1-x)^{p_3}$$
$$f_d(x) = p_4 \cdot x^{p_5} \cdot (1-x)^{p_6}$$
$$\sigma_p = \frac{4}{9} \cdot f_u + \frac{1}{9} \cdot f_d$$
$$\sigma_n = \frac{1}{9} \cdot f_u + \frac{4}{9} \cdot f_d$$

```
get_u=lambda x,a,b: x**a*(1-x)**b
get_d=lambda x,a,b: 0.1*x**a*(1-x)**b

def get_sigma1(x,p):
    u=get_u(x,p[0],p[1])
    d=get_d(x,p[2],p[3])
    return 4*u+d

def get_sigma2(x,p):
    u=get_u(x,p[0],p[1])
    d=get_d(x,p[2],p[3])
    return 4*d+u
```

CLOSURE TEST

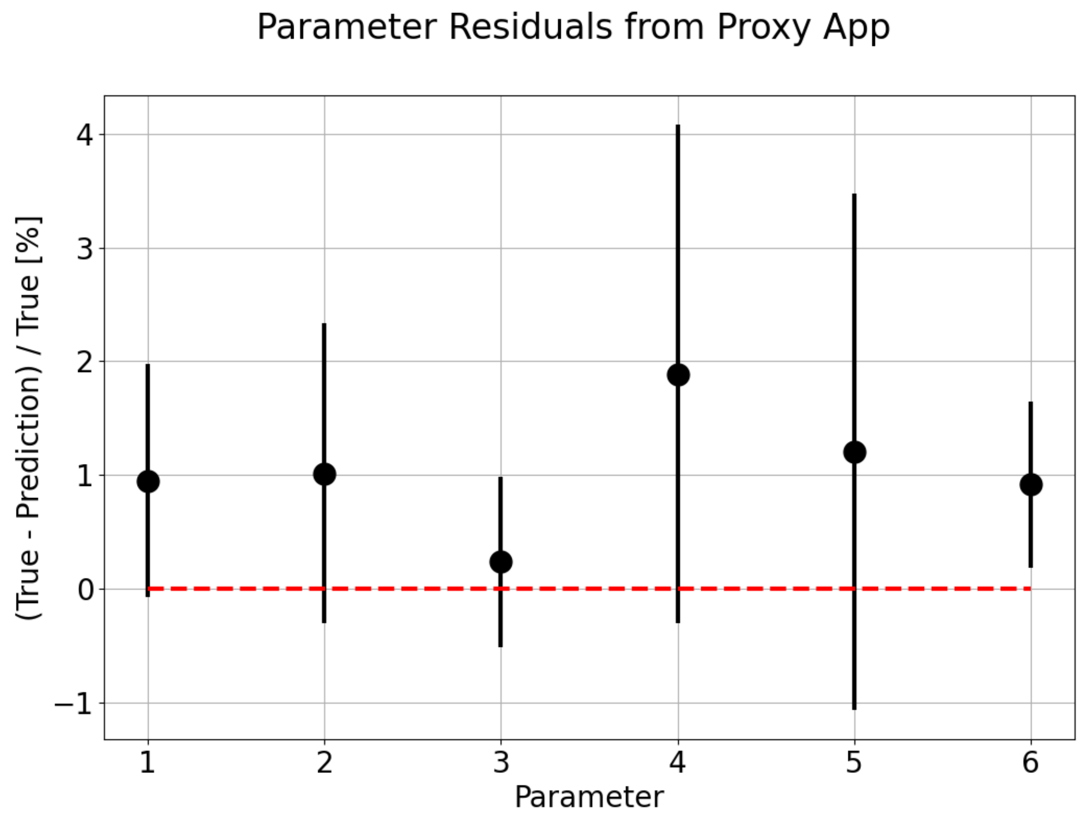
- Use the Proxy App for closure tests and scaling
- We generate toy data (1M events) using fix parameters in the theory module
- We train an ensemble of 15 GAN workflows
- Results are from ideal setup

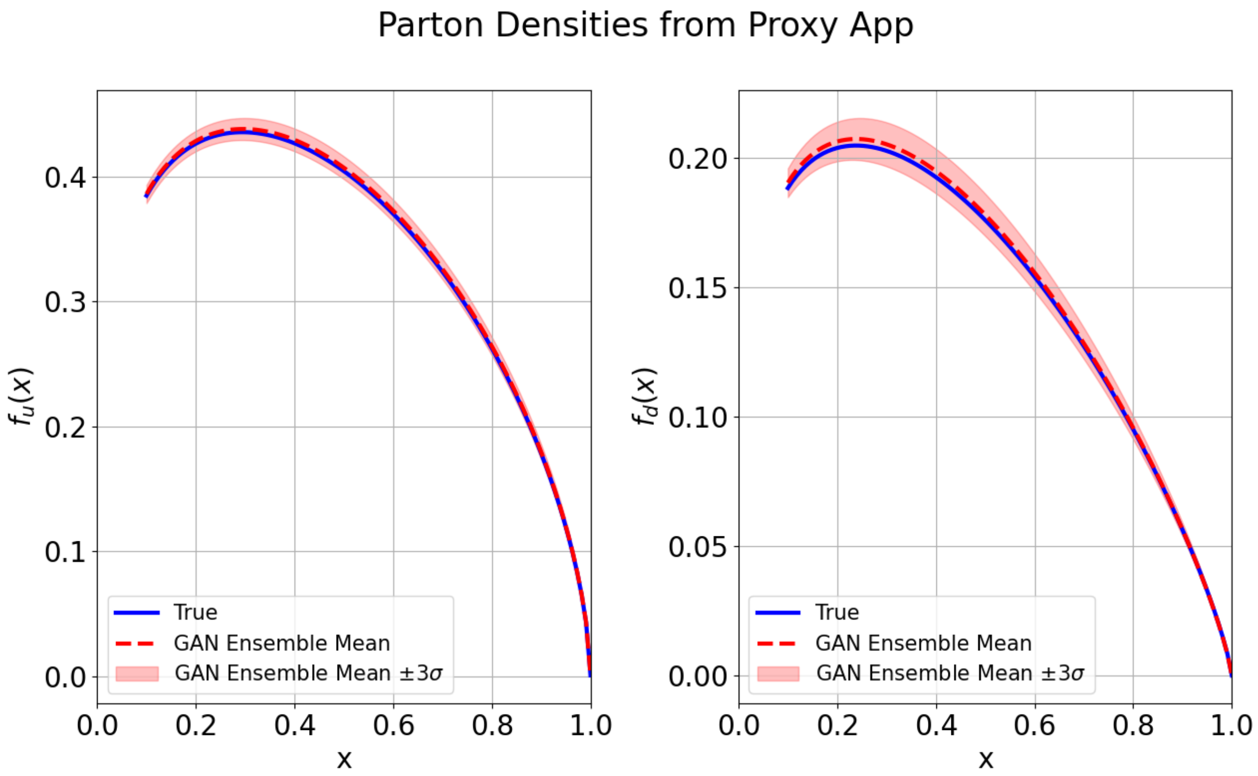




ADDITIONAL PROXY APP RESULTS

- Including some detector effects:
  - Detector with 5% / 12% resolution effect on sigma1 / sigma2
  - Added correlations between sigma1 / sigma2 (less than 10%)



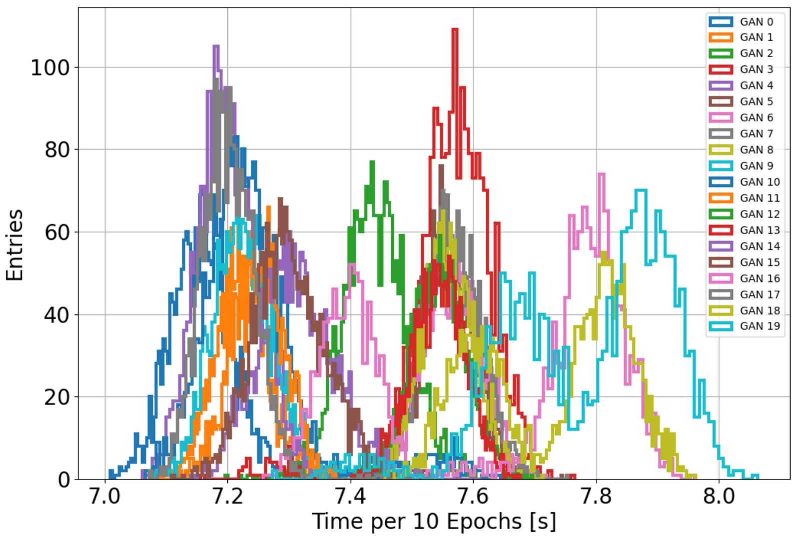


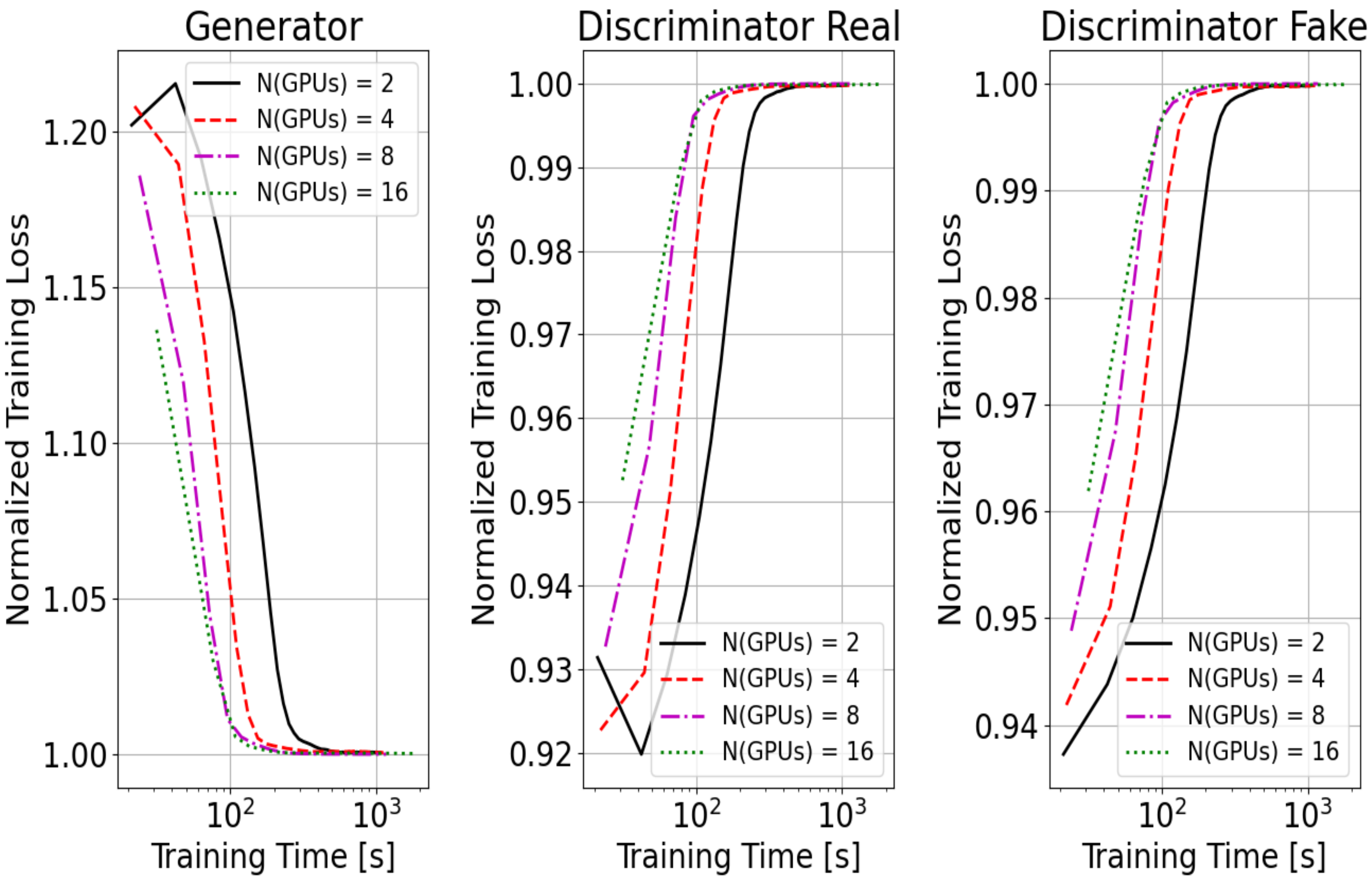
SCALING USING COMMON TOOLS

- Initial scaling studies were based on Horovod and Fairscale

- Both didn't scale very well (Horovod results on right)

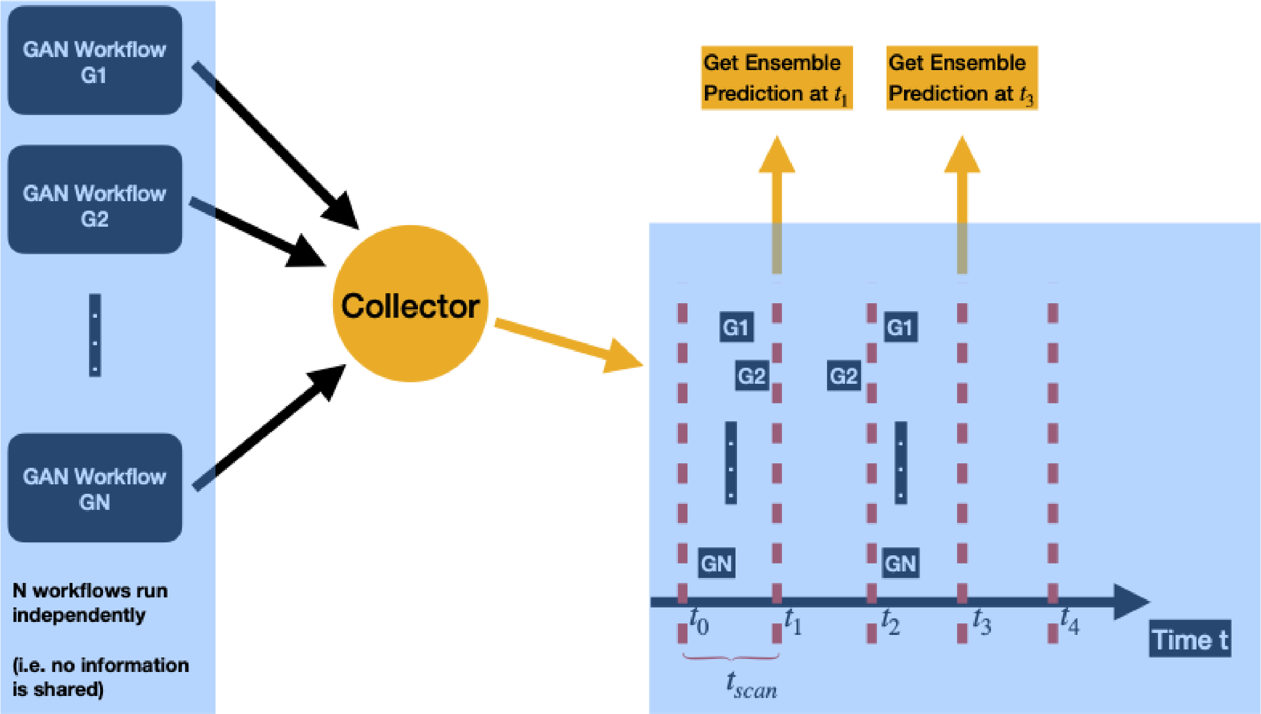
- The stochastic nature of the workflow doesn't work well with allreduce techniques

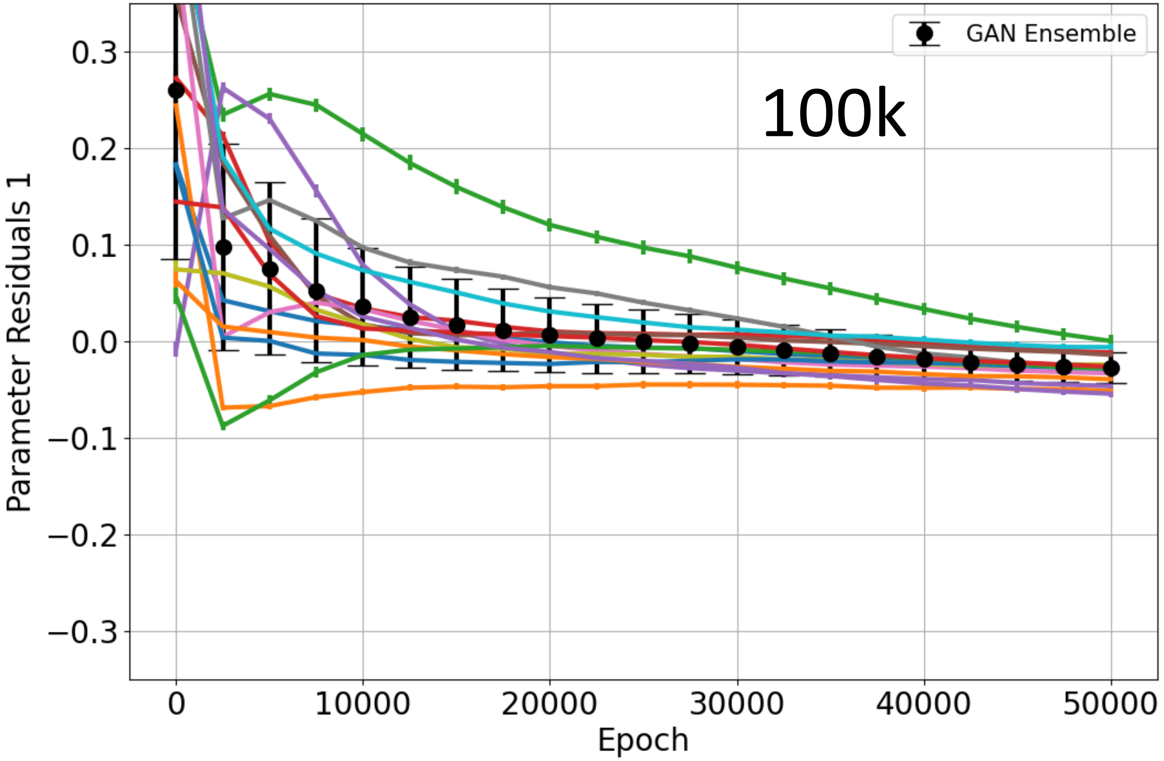




EXPLORING SCALING USING ENSEMBLES

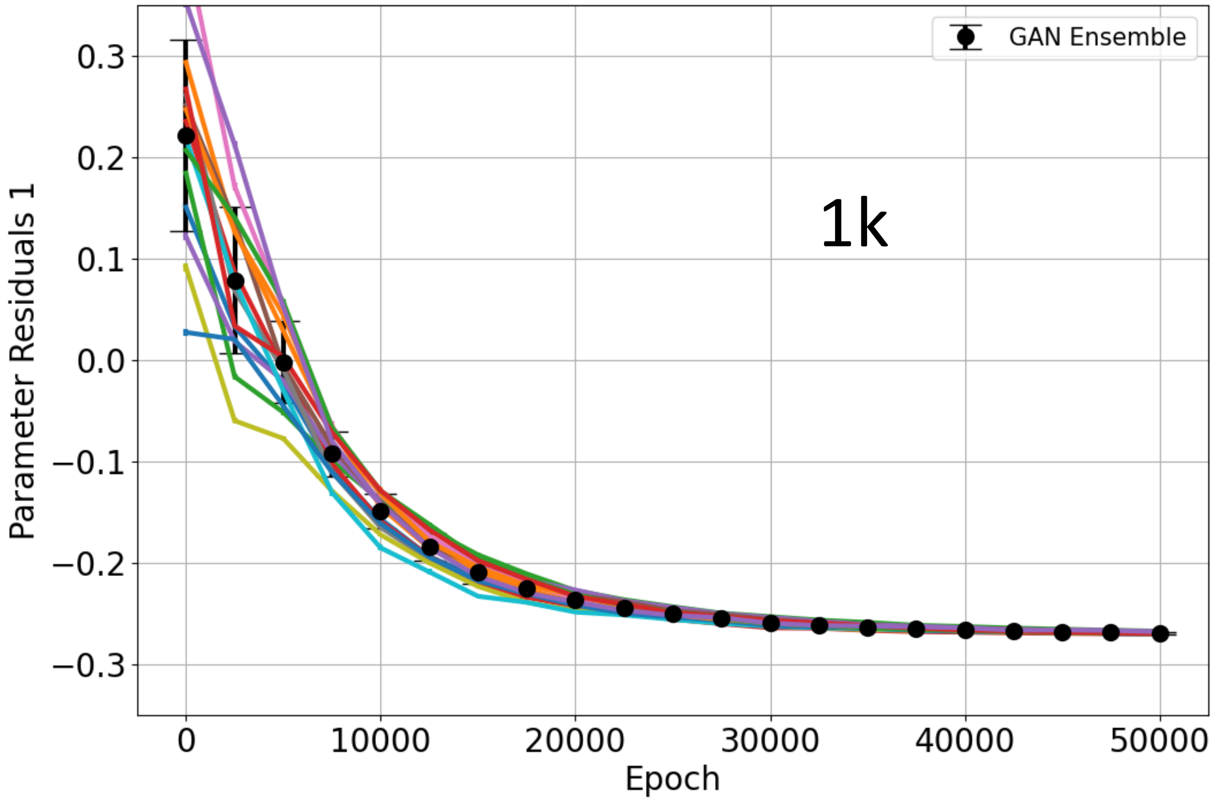
- Simplest approach is to have completely decoupled learning and take aggregate predictions at fixed time intervals
- This allows us to study the model stability and accelerate convergence
- Aggregated learning using asynchronous MPI will be explore next

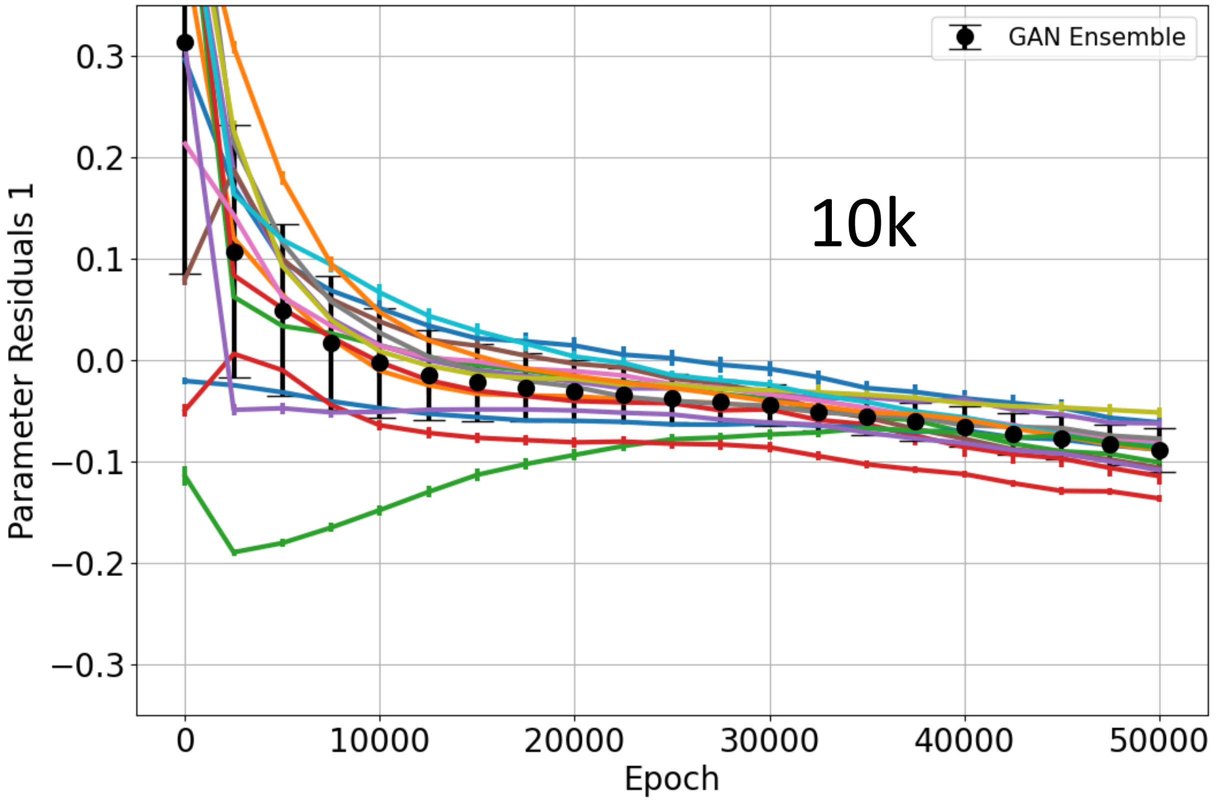




IMPACT FROM SAMPLE STATISTICS

- Statistical impact in results: 1k samples (left) and 10k samples (right)
- Clear bias that need to be understood (difference between parent and sampled distributions)
- Training samples are fixed per ensemble. We are including the sample statistic effects now.

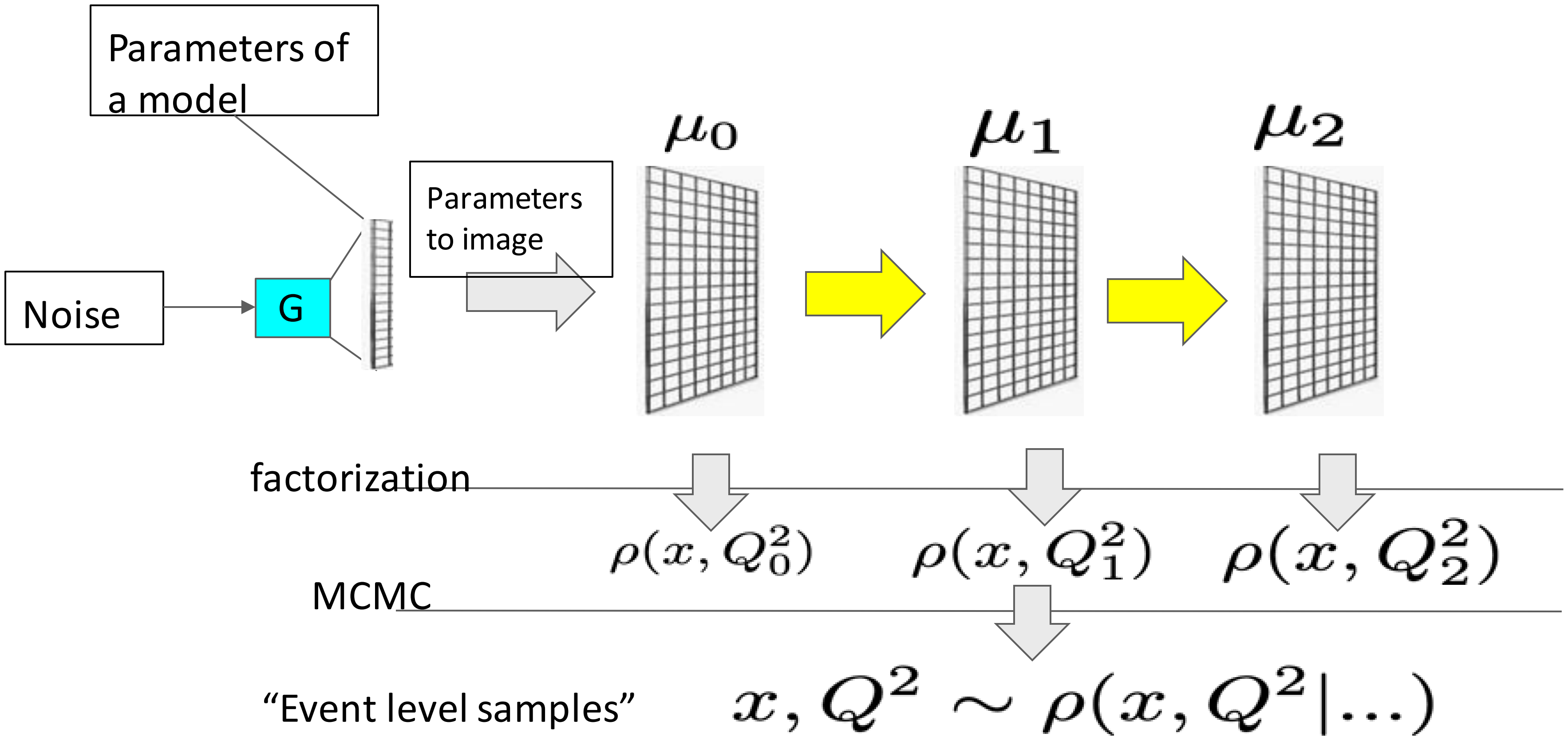




Jefferson Lab

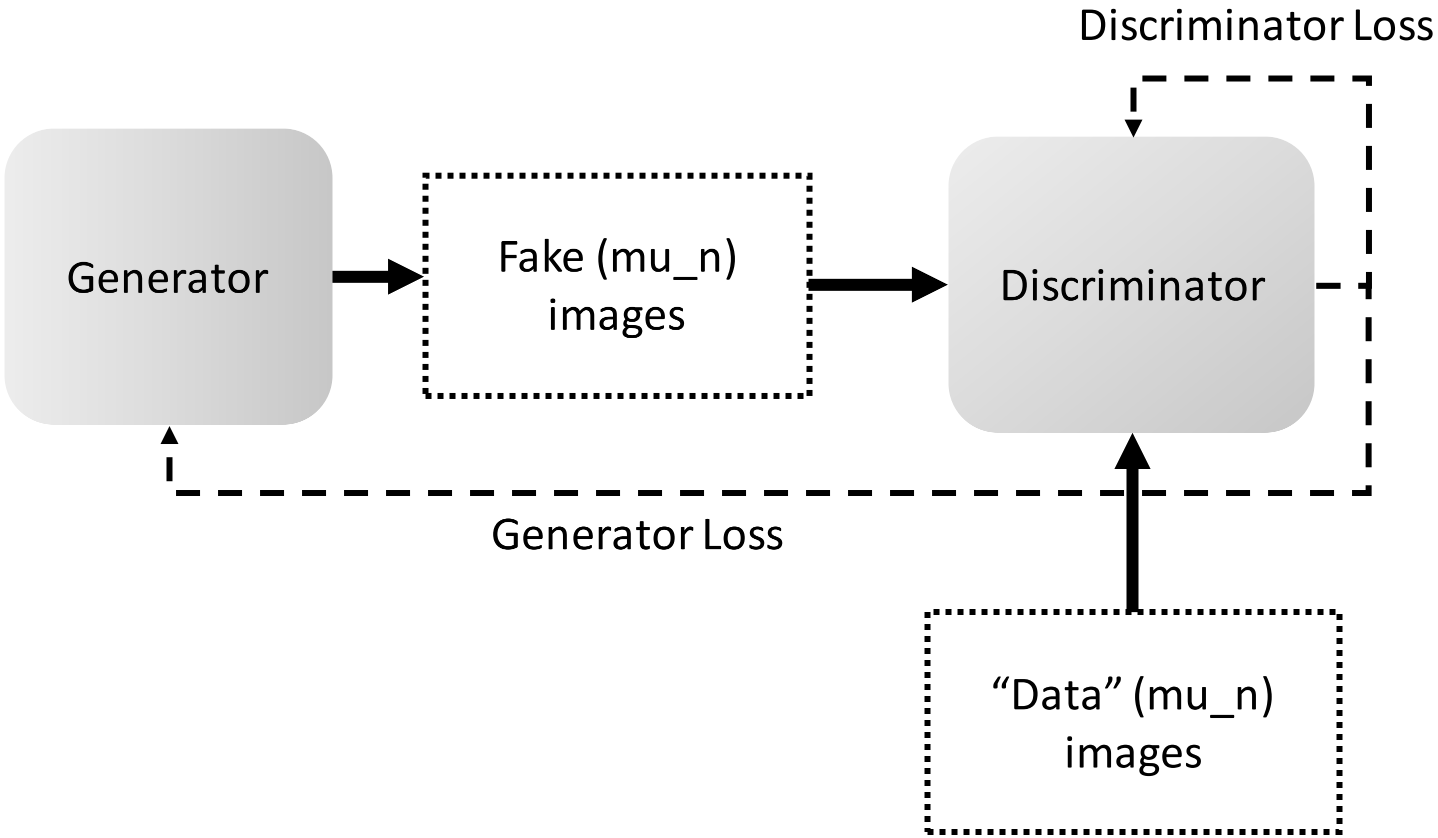
BEYOND THE PROXY APP

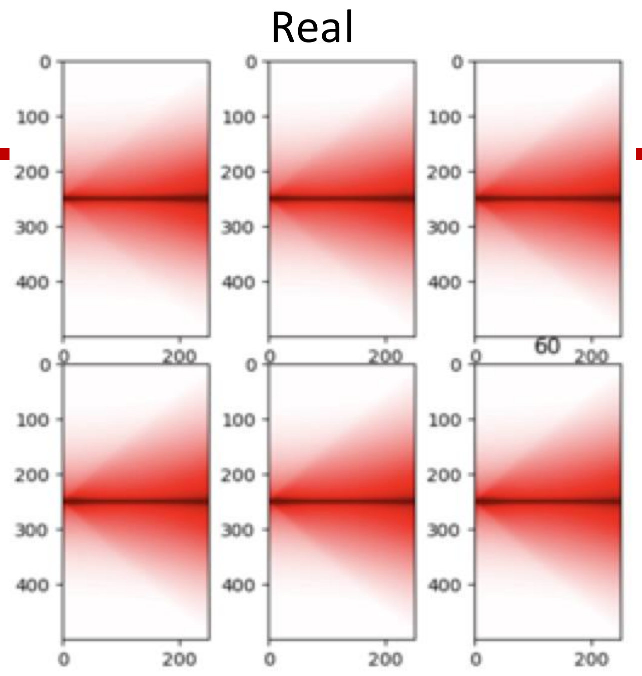
- The proxy app allows us to study the framework and scaling
- We need to also study potential real use case (inclusive DIS, etc.)

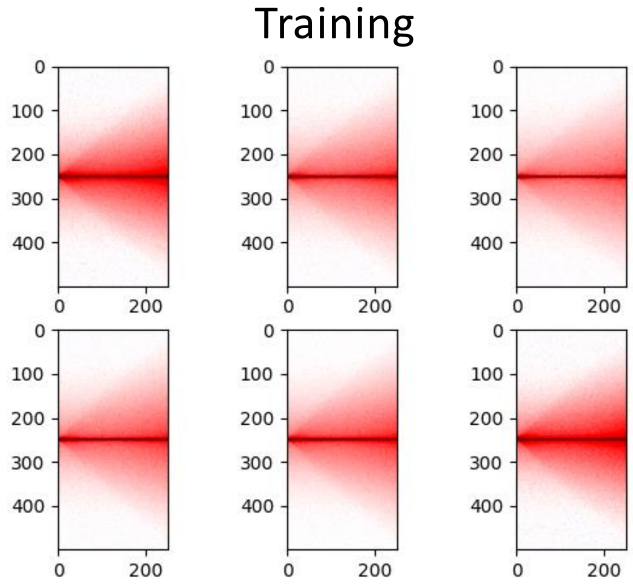


LEARNING FROM GPD IMAGES

- Goal: Recover the evolved "data" images for theory code
- WARNING: there is no physical equivalent
- This is the MNIST example for NP ☺

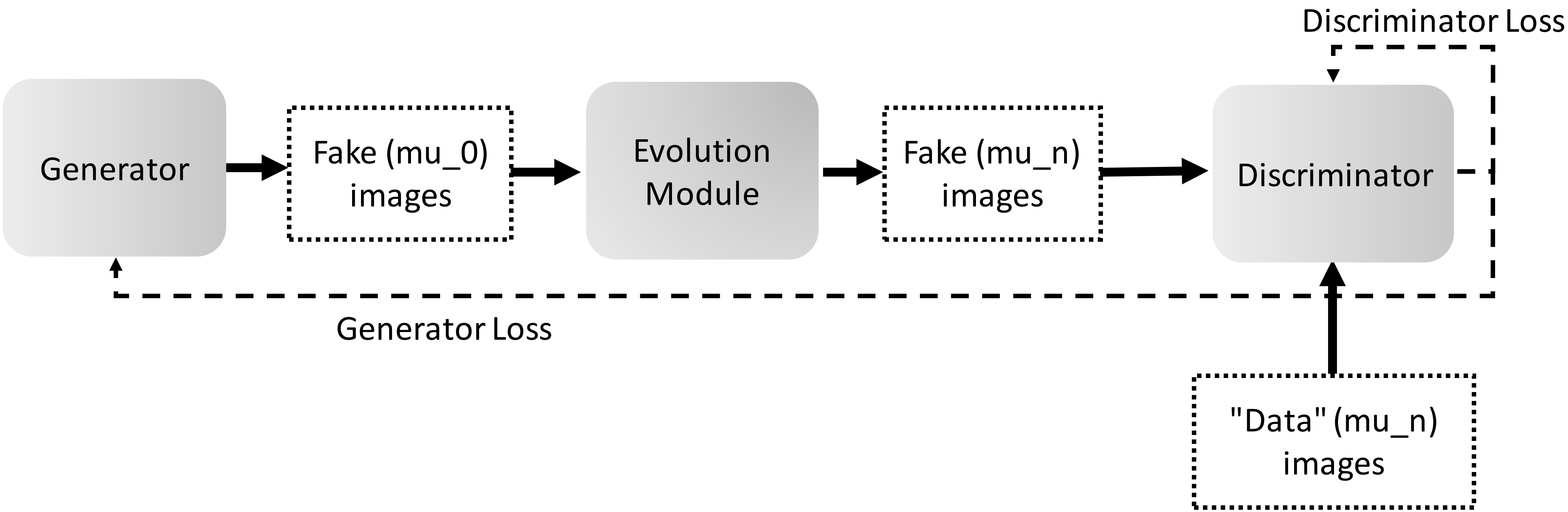






LEARNING FROM GPD IMAGES

- Next step is to add the evolution code in the mix
- The evolution code is slow and require a lot of memory
- In fact, we cannot run on GPU due to memory requirement (on A100)



STATUS & CHALLENGES

Status:

- Workflow is working and is easily configurable
  - Easy to add new theories (proxy app, images, etc)
  - Initial scaling is ongoing with modest improvements
  - Completed end-to-end closure test for proxy app … discovered several interesting challenges

Challenges:

- Gradient-based approach require all components to be differentiable for backpropagation
  - This is a challenge for traditional sampling techniques (e.g. MCMC) and other components within the workflow

- Gradient-based optimizers store large amount of information
  - This is a problem when backpropagating through evolution code, etc.

- Elements of the current workflow have stochastic execution times which doesn't scale using allreduce methods (e.g. Horovod, etc.)

Workflow is not limited to Quantom applications


Thank you