NETL Workshop on Multiphase Flow Science
August 13th – 14th, 2024

Pioneering Real-Time (In-Situ) Machine Learning Integration for Multiphase Flow Analysis: A First-of-Its-Kind Workflow Demonstration with MFIX-Exa

Aytekin Gel[1] , Andrew Shao[2],
Jordan Musser[3] , William Fullmer[3]

1 ALPEMI Consulting, L.L.C., Phoenix, AZ, U.S.A.
2 Hewlett Packard Enterprise, Seattle, WA, U.S.A
3 National Energy Technology Laboratory , Morgantown, WV, U.S.A

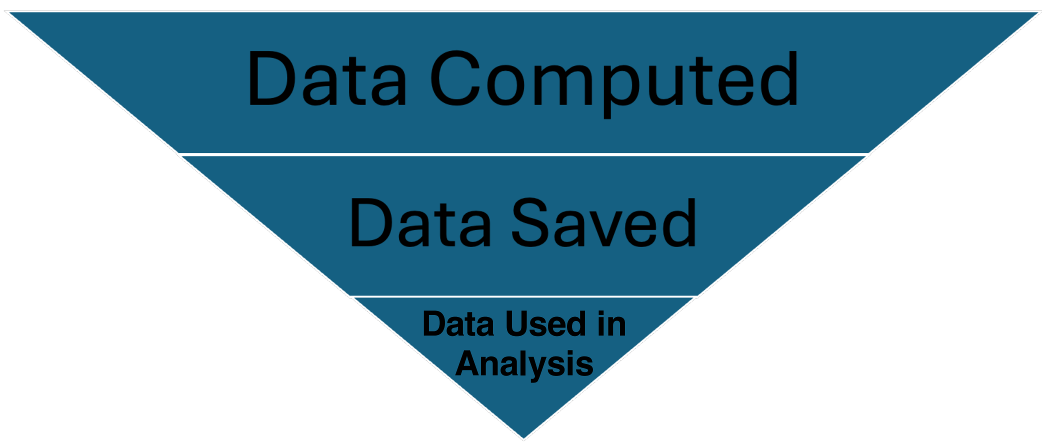
ALPEMI
CONSULTING

Outline

- Motivation and Objectives
- Demonstration Case Overview
- Overview of MFIX-Exa
- Machine Learning (ML) Model Development
- Overview of the Dataset from the Simulation Campaign & Postprocessing
- In-Situ Machine Learning Training Workflows
- Intelligent Downsampling
- Proof-of-Concept Implementation of the In-Situ ML Workflow
- Concluding Remarks

Motivation and Objectives

- **Computational Science and Simulation-Based Engineering (SBE)**: SBE has become an indispensable tool for solving complex problems through simulation.
- **Integration of Machine Learning**: Machine learning-based models are increasingly becoming essential components of SBE, and even replacing parts of it.
- **Advancements in High-Performance Computing**: The advent of Exascale supercomputers has enabled tackling more challenging problems.
- **Data Management Challenge**: There is a substantial gap where the amount of data computed greatly exceeds the data saved and the data used in analysis.



ALPEMI CONSULTING

Motivation and Objectives

- **I/O Overhead and Storage Limitations**: I/O overhead and data storage limitations force researchers to discard some computed data without knowing its potential usefulness, leading to inefficiencies and waste of resources.
- **Data Demand of Machine Learning (ML):** Machine learning models such deep learning require large amounts of data to achieve high accuracy, which highlights the importance of efficient data management and utilization.

Hypothesis:

Does a real-time (in-situ) machine learning workflow significantly enhance data utilization, model accuracy, and the quality of insights derived from simulation results compared to traditional offline (batch) machine learning methods?

Motivation and Objectives

- **Objectives:**
  - Develop a Real-Time (In-Situ) ML Workflow:
    - **Goal**: Construct a flexible and scalable in-situ ML workflow as a proof-of-concept demonstration.
    - **Focus**: Ensure that the workflow integrates seamlessly with existing simulation tools and scales effectively with data size and computational resources.
  - Identify Bottlenecks:
    - **Task:** Diagnose and address any performance or integration issues within the in-situ ML workflow**.**
    - **Outcome:** Provide insights into the limitations and areas for improvement.
  - Demonstrate Proof-of-Concept Workflow Operation:
    - **Integration:** Showcase the complete workflow with MFIX-Exa, a state-of-the-art multiphase flow CFD code.
    - **Scalability:** Validate the workflow's capability to handle large-scale simulations across multiple nodes.
  - Conduct Comparative Studies:
    - **Model Accuracy:** Assess and compare the accuracy of ML models developed in-situ versus offline.
    - **Data Write-Out Frequency:** Develop guidelines on optimal data write-out frequency to balance between ML model accuracy and computational efficiency.

Demonstration Case Overview

- U.S. Department of Energy's Office of Science's competitively selected ASCR Leadership Computing Challenge (ALCC) program award (7/1/23-6/30/24*)

- "**Machine Learning-Enhanced Multiphase CFD for Carbon Capture Modeling**"

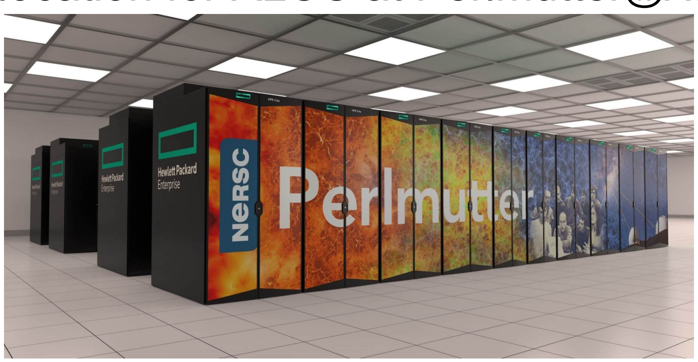
  PI: Jordan Musser (NETL),

  Co-PIs : Aytekin Gel (ALPEMI Consulting),

  William Fullmer (NETL)

- **Award: 100,000 GPU node hours**

  ≈ 1/10th of the annual total available GPU allocation for ALCC at Perlmutter@NERSC



Fact Sheet from 2023-2034 ALCC Awards

U.S. DEPARTMENT OF ENERGY | Office of Science

Title: Machine Learning-Enhanced Multiphase CFD for Carbon Capture Modeling

Principal Investigator: Jordan Musser (National Energy Technology Laboratory)

Co-investigators: Aytekin Gel (ALPEMI Consulting, LLC),
William Fullmer (National Energy Technology Laboratory)

ALCC Allocation:

Site(s): National Energy Research Scientific Computing Center (NERSC)

Allocation(s): 100,000 node-hours on Perlmutter-GPU

Research Summary:

This project leverages machine learning (ML) to enhance simulation-based engineering models that analyze how solid particles move and interact with a carrier fluid. Scientists use these tools to evaluate designs and help troubleshoot advanced particle-based reactors; however, because of long simulation times, researchers commonly choose faster running, less accurate models to reduce the time to solution. MFIX-Exa, a newly developed software designed to efficiently run on modern GPU-accelerated supercomputers, will generate high-fidelity datasets for training ML derived surrogate models. The created ML models that characterize phenomena like the interaction force between particles and the fluid can be incorporated into the faster running simulation tools to improve accuracy. This effort supports the Department of Energy's mission by enabling scientists to rapidly evaluate novel gas-solid reactor designs for advanced CO2 capture technologies.

2023 ASCR Leadership Computing Challenge Award

High Level Overview of MFIXExa

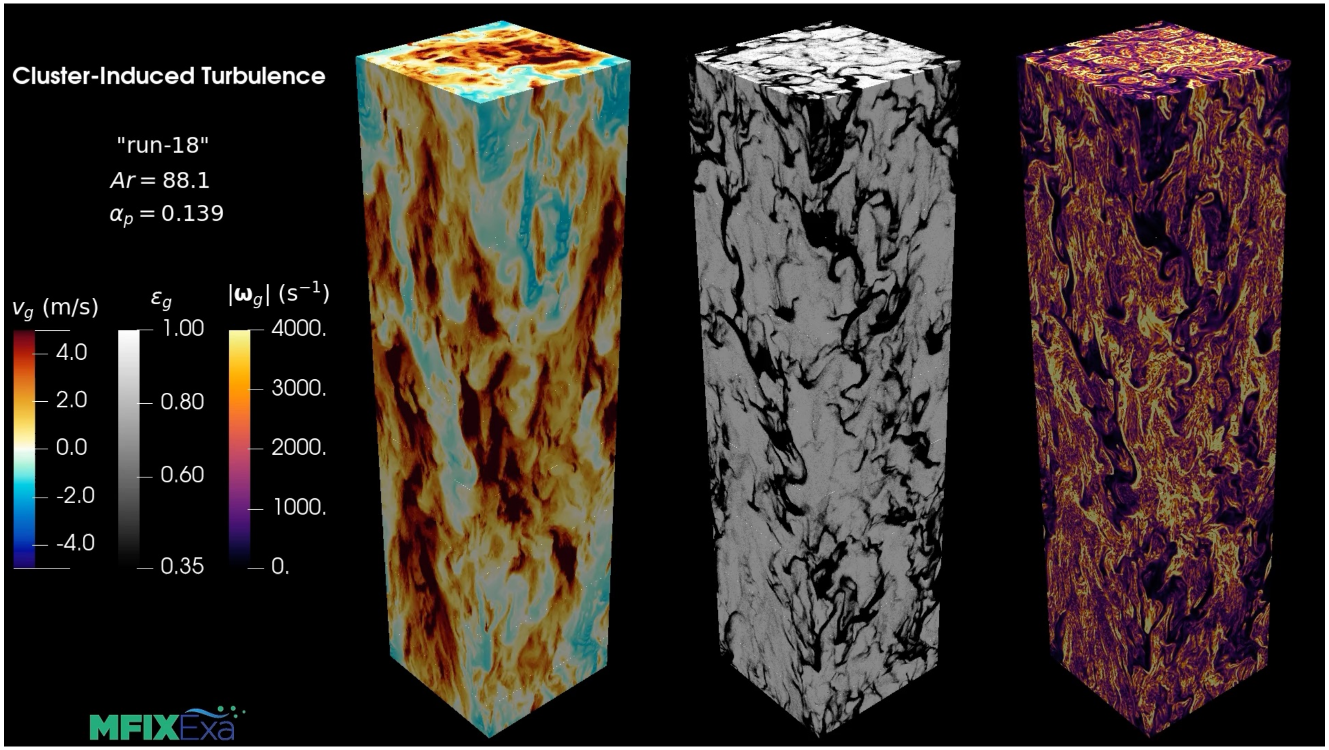
- MFIX-Exa was developed under DOE's Exascale Computing Project (ECP) and build using the AMReX framework for performance portability
- Fluid solver is a low-Mach fluid formulation using an explicit update with an approximate projection to enforce incompressibility
- Contains two Lagrangian particle models that are fully coupled to the fluid
  - DEM – Tracks every particle and resolves collisions using the soft-sphere model
  - PIC – Parcels represent collections of particles and only interact via a solids pressure
- MFIX-Exa has been tested on numerous DOE Leadership class facilities including Frontier and Summit at ORNL, Perlmutter at NERSC, and Polaris and Aurora at ALCF.
- For more information, please see Wednesday 9 AM talk on MFIX-Exa or visit MFIX-Exa website:

https://mfix.netl.doe.gov/products/mfix-exa/

ALPEMI CONSULTING

Demonstration Case Overview

- **Simulated Problem:** An unbounded (triply periodic) system of particle sedimentation or, equivalently, fluidization.
- System size twice as large (in each direction) of the previous state-of-the-art study*



Domain Size:
Lx = 0.1536 m
Ly = 0.0384 m
Lz = Ly

I/O frequency:
Every 0.025s for 100 times

- Resolution of fluid cells: 1024 × 256 × 256 (approximately 67 million cells)
- Total number of simulated particles ranging from 10 to 410 million particles as part of the simulation campaign.

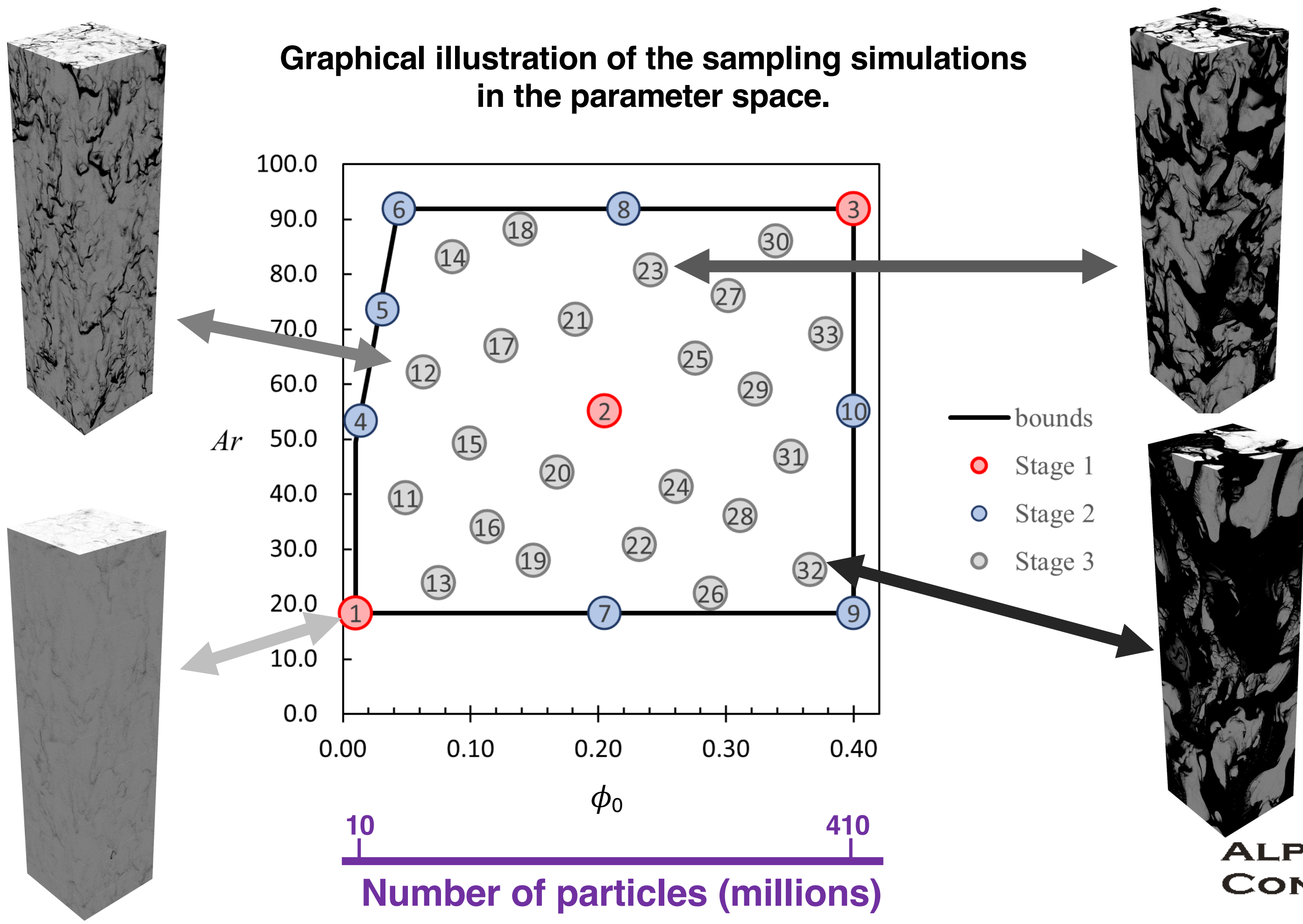
*** S. Beetham, R. O. Fox and J. Capecelatro, "Sparse identification of multiphase turbulence closures for coupled fluid–particle flows,"** ***Journal of Fluid Mechanics*****, 914, A11, 2021**

ALPEMI CONSULTING

Demonstration Case Overview

- Simulation campaign: carefully designed 33 samples employing space-filling design
- Two key parameters are systematically varied:
  Archimedes number (18 ≤ Ar ≤ 92) and initial solids concentration (0.01≤ ϕ_0 ≤0.4)

Graphical illustration of the sampling simulations in the parameter space.



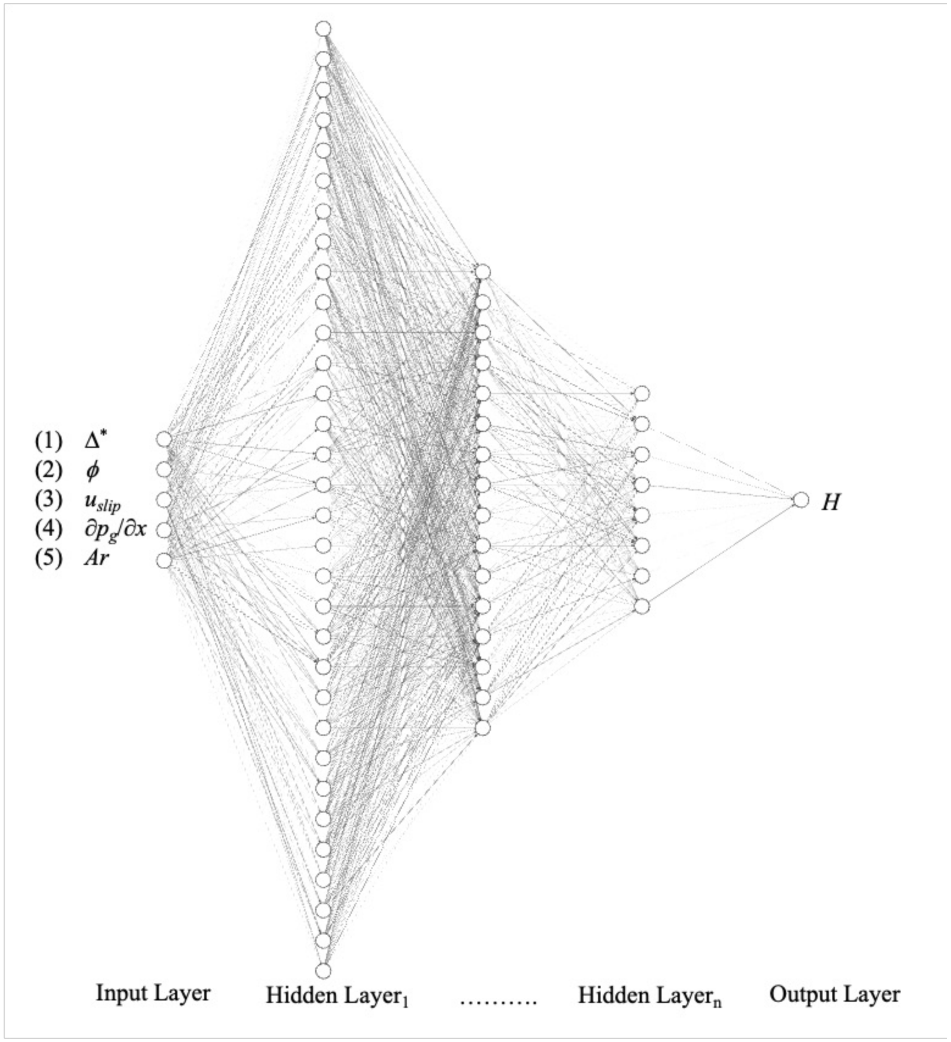
ALPEMI CONSULTING

Machine Learning Model Development

- Construct an ML surrogate model to characterize heterogenous index, *H* :

$$H = f\left(\Delta^*, \quad \hat{\phi}, \quad \hat{u}^*_{slip}, \quad \left(\widehat{\frac{\partial p_g}{\partial x}}\right)^*, \quad Ar\right)$$

- Constructed ML model can be plugged into lower-fidelity CFD code (e.g., MFIX-Exa PIC or MFiX-PIC) to characterize the governing physics without the need for high-fidelity resolution runs.
- Dataset: Post-process the data generated from the simulation campaign with 33 samples.



ALPEMI CONSULTING

Overview of the Dataset from the Simulation Campaign & Data Postprocessing

- Raw dataset:
  - 33 samples of simulations (run-01…. run-33)
  - Each MFIX-Exa simulation had 100 unique timesteps saved in AMReX native file format (/plt*)

Sample No	Time step	Folders	Size (GB)
	1	/plt19909	4.7
	2	/plt20450	4.7
run-01	….	…..	
	99	/plt72481	4.7
	100	/plt72981	4.7
	Total run-01 size:		471
Zip Compressed Total size:			412

Sample No	Time step	Folders	Size (GB)
	1	/plt25166	17
	2	/plt28180	17
run-02	….	…..	
	99	/plt308910	17
	100	/plt311974	17
	Total run-02 size:		1700
Zip Compressed Total size:			1400

1.4 Terabytes

Sample No	Time step	Folders	Size (GB)
	1	/plt027725	29
	2	/plt031095	29
run-03	….	…..	
	99	/plt310054	29
	100	/plt313316	29
	Total run-03 size:		2900
Zip Compressed Total size:			2400

2.4 Terabytes

……………

	Time step	Folders
	1	/plt28102
	2	/plt32106
run-33	….	…..
	99	/plt302597
	100	/plt323079
	Total run-33 size:	
Zip Compressed Total size:		

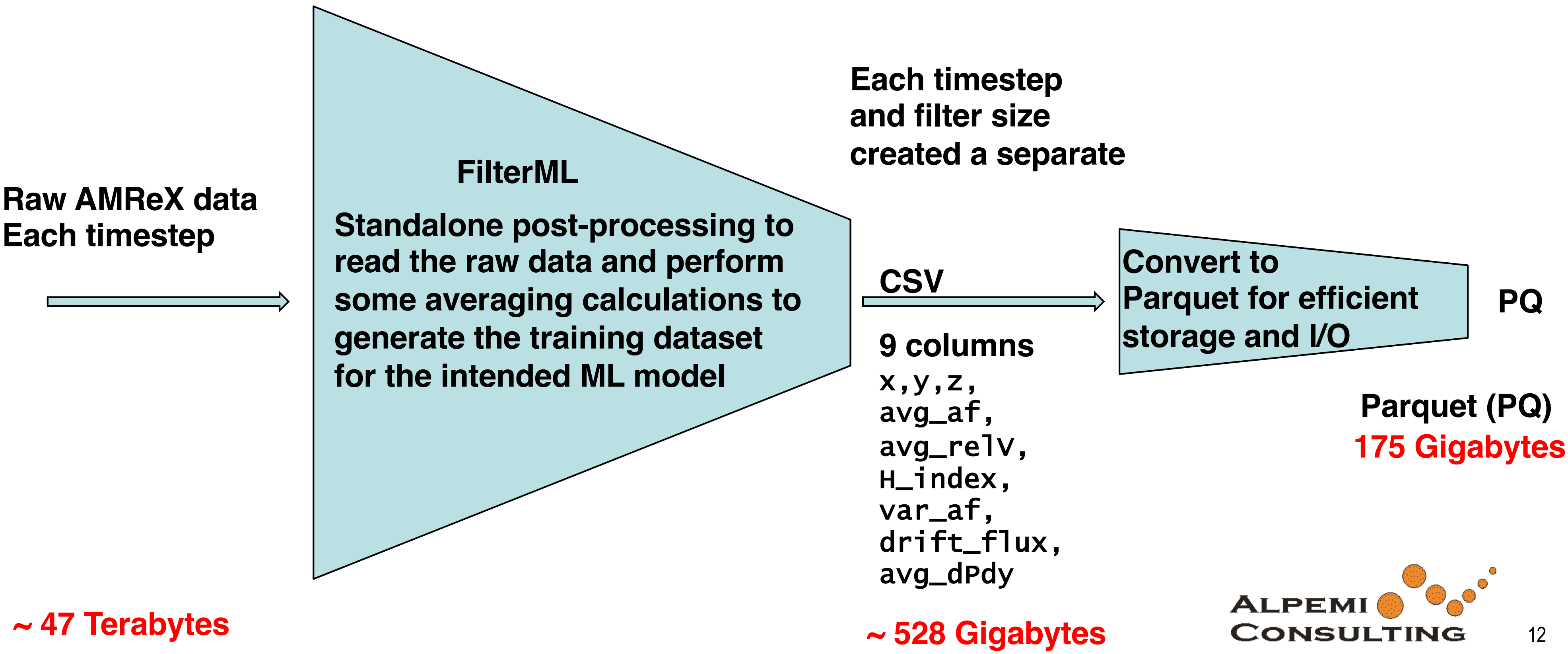
Total Size of 33 Samples Simulation Campaign Results
(Zip Compressed AMReX native format) = 46.8 Terabytes

As part of the ALCC project outcome, raw dataset is available at NETL EDX: https://edx.netl.doe.gov/dataset/mfix-exa-alcc2324-run-data

ALPEMI CONSULTING

Overview of the Dataset from the Simulation Campaign & Data Postprocessing

- Explored various options to handle the dataset
  - Conversion to HDF5 or directly write out in HDF5
  - Leverage Python libraries (e.g., DASK, Arkouda)
- Standalone post-processing tool FilterML employed



ALPEMI CONSULTING

ML Model Development to compute H-index

- Preliminary Analysis: Used automated ML tools (PyCaret) to explore and eliminate the traditional ML methods.

Automated ML Model Construction Results for the transient dataset for 33 samples with 100 timesteps each case

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.0662	0.0305	0.1739	0.5356	0.0855	0.3151	1.394
gbr	Gradient Boosting Regressor	0.0685	0.0313	0.1763	0.5225	0.0872	0.3380	7.662
catboost	CatBoost Regressor	0.0673	0.0320	0.1780	0.5131	0.0869	0.3184	5.008
rf	Random Forest Regressor	0.0663	0.0328	0.1804	0.5003	0.0892	0.3162	0.730
et	Extra Trees Regressor	0.0676	0.0340	0.1837	0.4819	0.0911	0.3251	0.322
xgboost	Extreme Gradient Boosting	0.0673	0.0345	0.1849	0.4742	0.0896	0.3163	0.237
knn	K Neighbors Regressor	0.0691	0.0352	0.1869	0.4637	0.0938	0.3217	0.092
lar	Least Angle Regression	0.1253	0.0514	0.2263	0.2131	0.1286	0.6419	0.018
br	Bayesian Ridge	0.1253	0.0514	0.2263	0.2131	0.1286	0.6419	0.023
ridge	Ridge Regression	0.1253	0.0514	0.2263	0.2131	0.1286	0.6419	0.018
lr	Linear Regression	0.1253	0.0514	0.2263	0.2131	0.1286	0.6419	0.036
omp	Orthogonal Matching Pursuit	0.1217	0.0524	0.2285	0.1977	0.1285	0.6079	0.017
dt	Decision Tree Regressor	0.0878	0.0603	0.2449	0.0767	0.1176	0.4022	0.421
huber	Huber Regressor	0.1060	0.0611	0.2468	0.0636	0.1382	0.3961	0.127
lasso	Lasso Regression	0.1352	0.0652	0.2550	-0.0001	0.1485	0.7470	0.022
en	Elastic Net	0.1352	0.0652	0.2550	-0.0001	0.1485	0.7470	0.017
llar	Lasso Least Angle Regression	0.1352	0.0652	0.2550	-0.0001	0.1485	0.7470	0.017
dummy	Dummy Regressor	0.1352	0.0652	0.2550	-0.0001	0.1485	0.7470	0.015
ada	AdaBoost Regressor	0.1466	0.0927	0.3001	-0.4129	0.1670	0.9593	0.912
par	Passive Aggressive Regressor	0.3326	0.3390	0.4967	-3.8829	0.2332	1.8604	0.042

Filter Size = 128, 3 input features

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.1061	0.0525	0.2288	0.4488	0.1143	0.4349	2.060
xgboost	Extreme Gradient Boosting	0.1054	0.0530	0.2298	0.4438	0.1145	0.4266	0.514
catboost	CatBoost Regressor	0.1073	0.0535	0.2309	0.4385	0.1157	0.4368	5.282
gbr	Gradient Boosting Regressor	0.1092	0.0540	0.2320	0.4331	0.1167	0.4632	71.024
rf	Random Forest Regressor	0.1087	0.0562	0.2367	0.4101	0.1196	0.4424	6.541
et	Extra Trees Regressor	0.1114	0.0588	0.2422	0.3824	0.1227	0.4543	1.914
knn	K Neighbors Regressor	0.1139	0.0616	0.2478	0.3531	0.1266	0.4530	0.660
lar	Least Angle Regression	0.1740	0.0830	0.2878	0.1270	0.1625	0.8086	0.059
br	Bayesian Ridge	0.1740	0.0830	0.2878	0.1270	0.1625	0.8086	0.080
ridge	Ridge Regression	0.1740	0.0830	0.2878	0.1270	0.1625	0.8086	0.056
lr	Linear Regression	0.1740	0.0830	0.2878	0.1270	0.1625	0.8086	0.080
omp	Orthogonal Matching Pursuit	0.1688	0.0853	0.2919	0.1023	0.1625	0.7526	0.056
huber	Huber Regressor	0.1574	0.0927	0.3042	0.0245	0.1680	0.5588	0.718
lasso	Lasso Regression	0.1794	0.0950	0.3080	-0.0000	0.1737	0.8377	0.074
en	Elastic Net	0.1794	0.0950	0.3080	-0.0000	0.1737	0.8377	0.080
llar	Lasso Least Angle Regression	0.1794	0.0950	0.3080	-0.0000	0.1737	0.8377	0.056
dummy	Dummy Regressor	0.1794	0.0950	0.3080	-0.0000	0.1737	0.8377	0.047
dt	Decision Tree Regressor	0.1457	0.1081	0.3286	-0.1395	0.1603	0.5642	4.564
par	Passive Aggressive Regressor	0.3079	0.2272	0.4615	-1.3544	0.2518	1.3010	0.257
ada	AdaBoost Regressor	0.7169	1.8142	1.2263	-18.6689	0.5074	4.1059	18.496

Filter Size = 64, 3 input features

Paradigm Shift: In-Situ ML Training Workflows

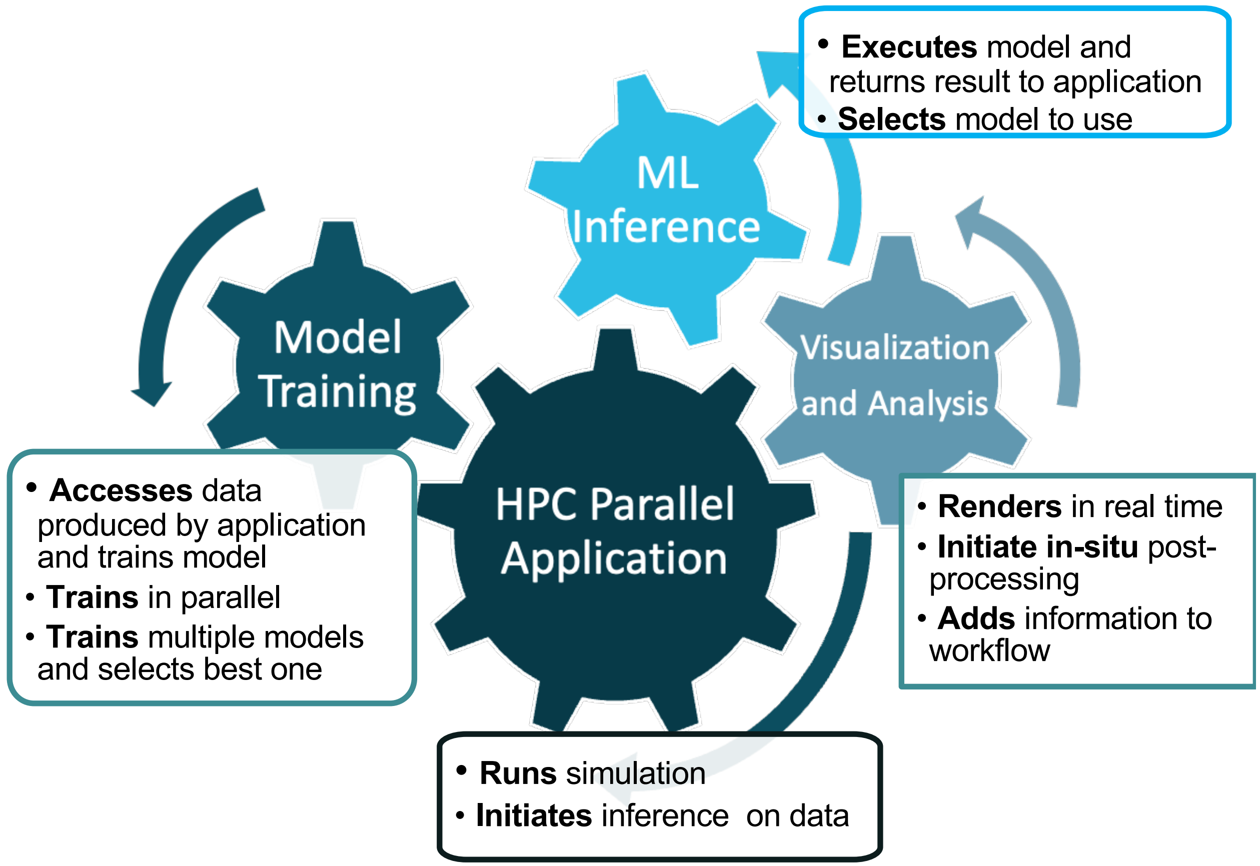
Typical Numerical Workflows

Input data
- Set of initial conditions
- File representing geometry
- Hard-coded values

Monolithic Application
- HPC native
- Parallel C/C++/Fortran
- Contains all needed logic
- Outputs to filesystem

Postprocessing
- Stored on filesystem
- Visualize or run analysis

New AI-Enhanced Numerical Workflows



Source: Adopted from "Smartsim: scientific worfklows with simulation and AI" Presentation by Andrew Shao at MMMHub Young Workshop June 2024

In-Situ ML Training Workflow Implementation with SMARTSIM

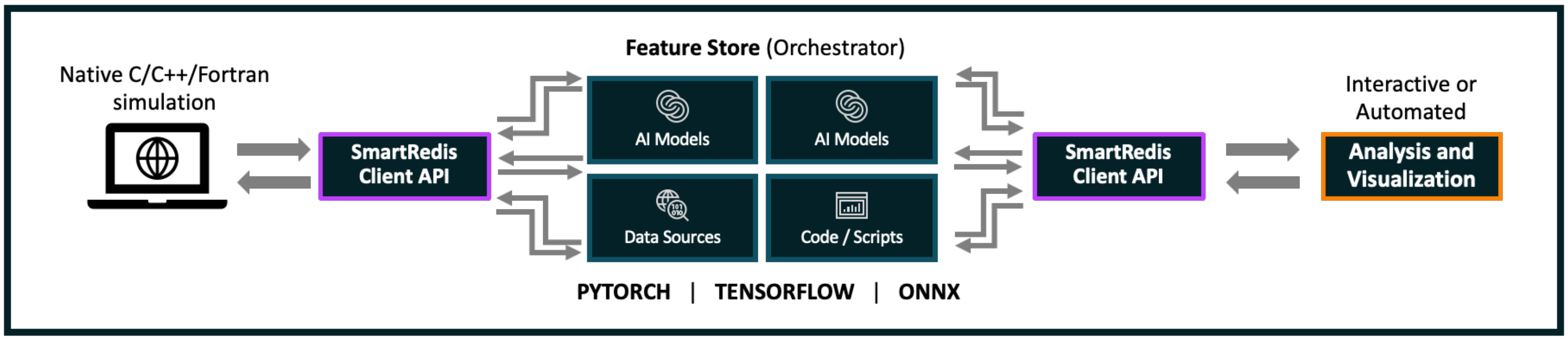
github.com/CrayLabs/SmartSim

- SmartSim
  - an open-source library
  - aims to bridge the divide between traditional numerical simulation and data science
  - enables scientists to create advanced and scalable workflows for scientific simulations integrated with machine learning without the complexity:
    - Call Machine Learning (ML) inference in existing Fortran/C/C++ simulations
    - Exchange data between C, C++, Fortran, and Python applications
    - Train ML models online and make predictions using TensorFlow, PyTorch, and ONNX
    - Analyze data streamed from HPC applications while they are running

Source: "Smartsim: scientific worfklows with simulation and AI" Presentation by Andrew Shao at MMMHub Young Workshop June 2024

In-Situ ML Training Workflow Implementation with SMARTSIM

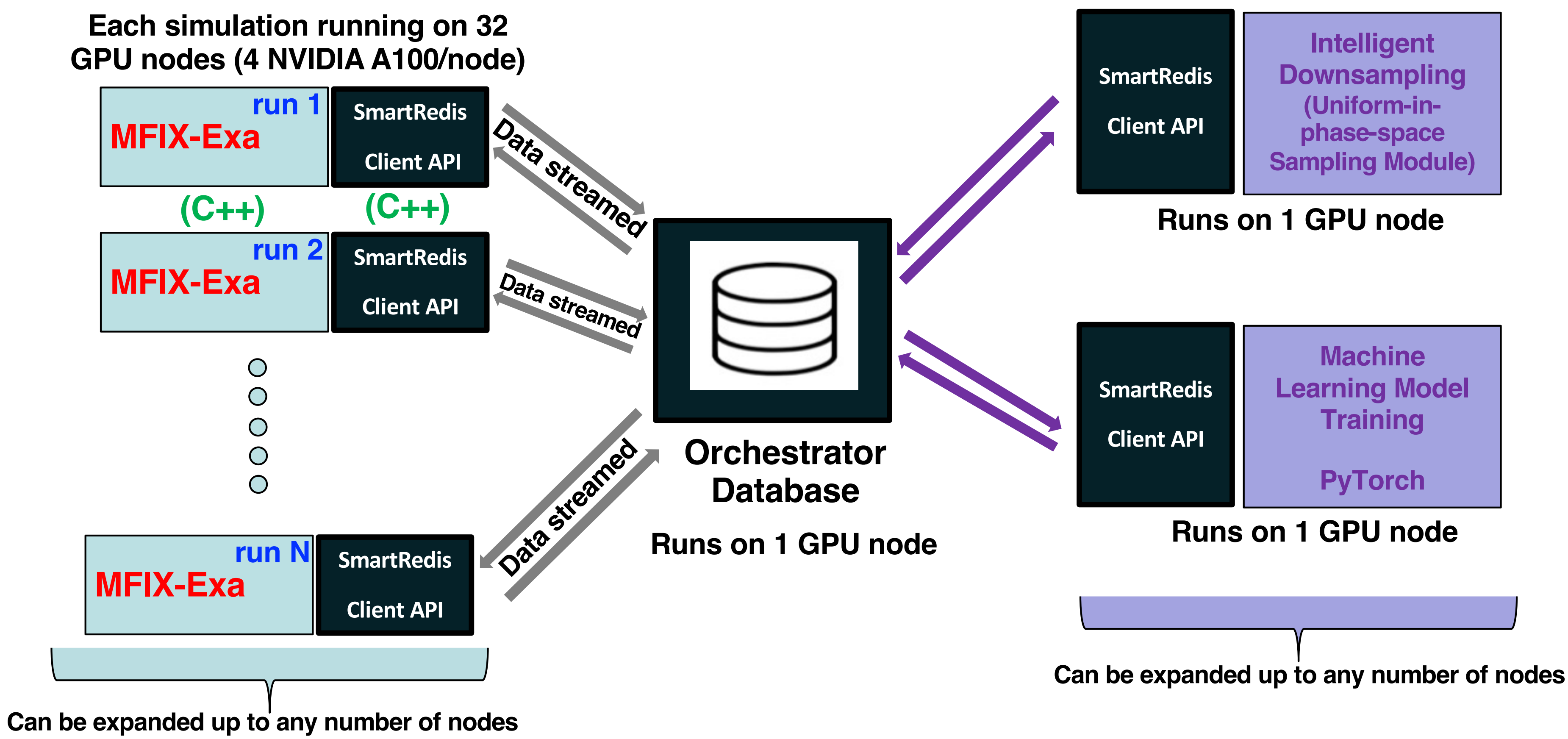
Typical SmartSim / SmartRedis Integrated Workflow:



All of these can be done *without touching the filesystem*

Source: Illustrations adapted from "Smartsim: scientific worfklows with simulation and AI" Presentation by Andrew Shao at MMMHub Young Workshop June 2024

Proof-of-Concept Implementation of the In-Situ ML Workflow



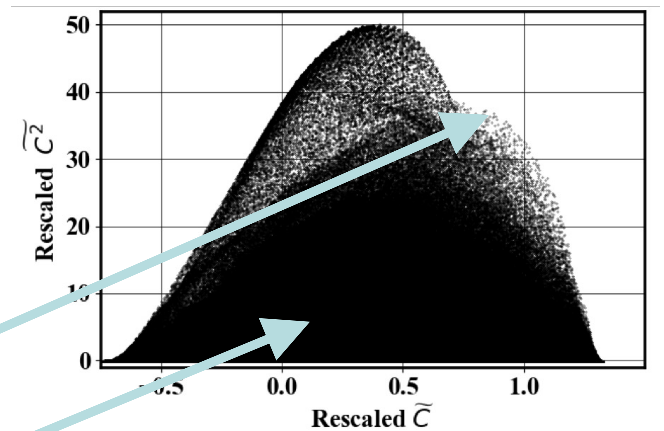
All in-memory, no read/writes to filesystem

Database can be scaled up to any number of nodes with clustered database setup

Intelligent Downsampling (Uniform-in-Phase-Space Sampling)

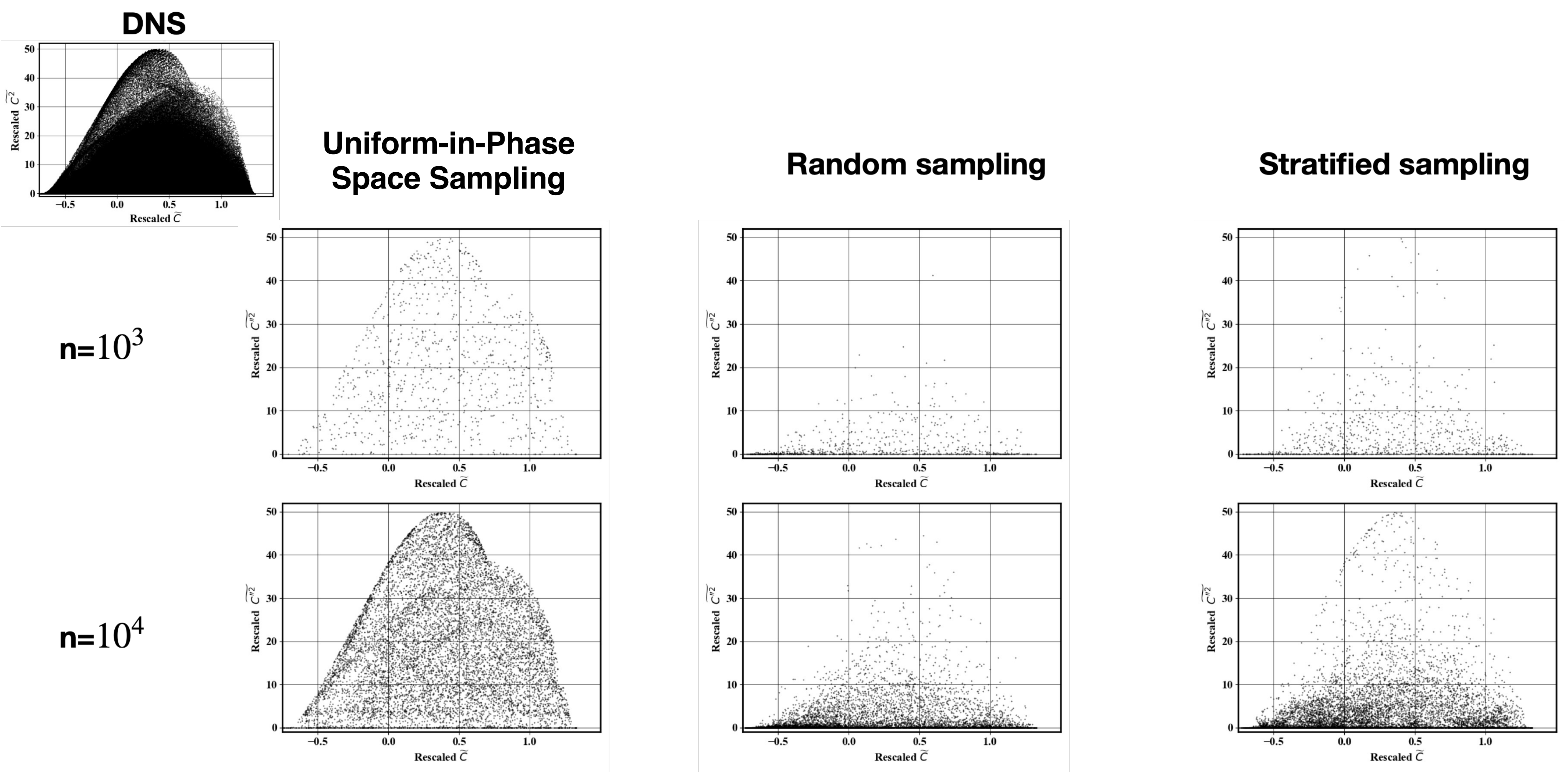
- Computationally efficient data selection method to reduce number of datapoints, which is applicable to any dataset
- Objective:
  - Select n datapoints s. t.
    - n << N
    - n data points are uniformly distributed in phase-space
    - n data points cover the full phase-space
- Why do that?
  - Ensure that **rare data points** are not discarded
  - Eliminate **redundant data**
- Open-source developed at NREL: github.com/NREL/Phase-space-sampling

DNS



Source: Adopted from "Data-Driven Reacting Flow Model Development: Data Sampling, Non-Linear Models and Uncertainty Quantification" Presentation at Stanford FLAME AI Workshop September 2023

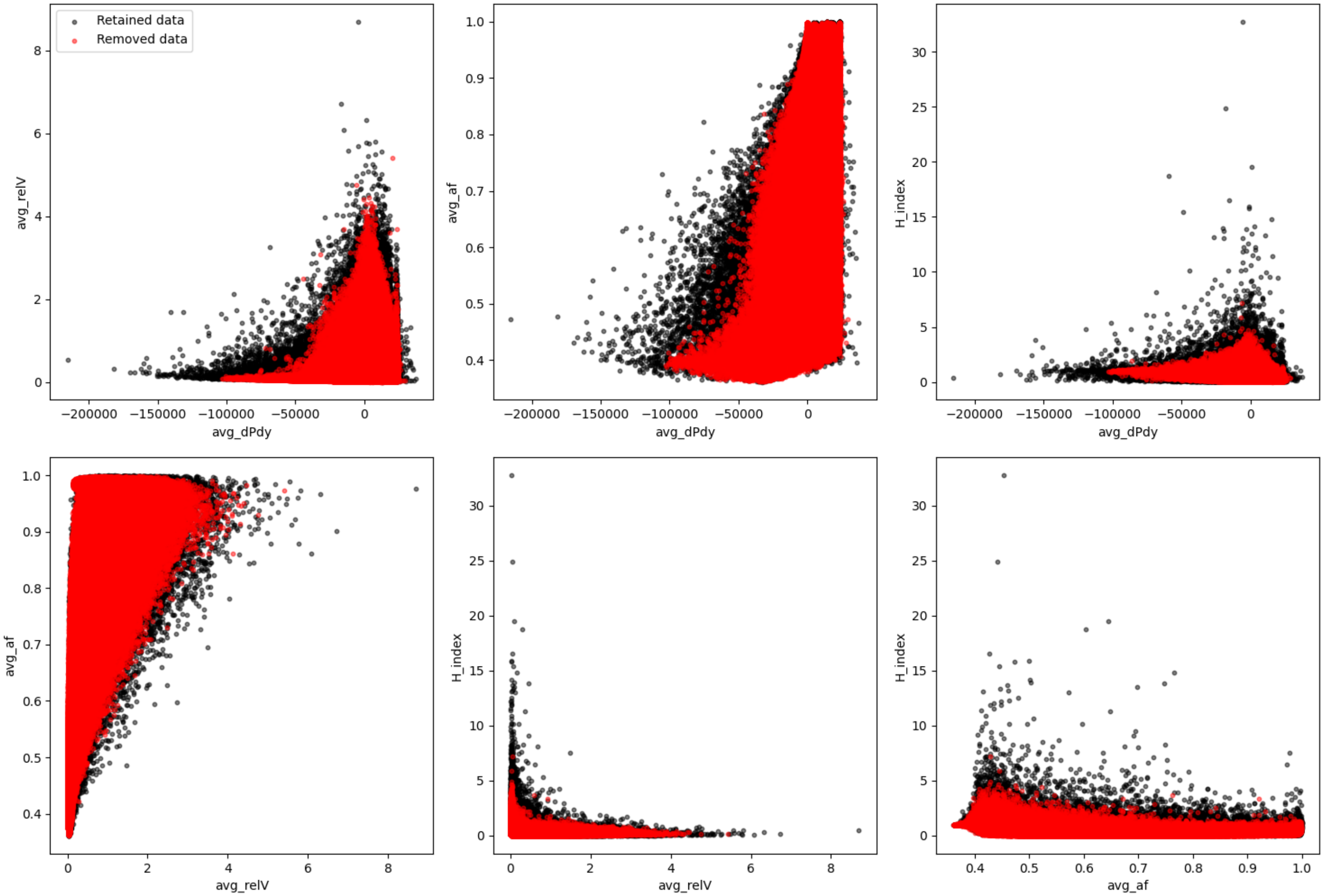
Intelligent Downsampling (Uniform-in-Phase-Space Sampling)



Source: Adopted from "Data-Driven Reacting Flow Model Development: Data Sampling, Non-Linear Models and Uncertainty Quantification" Presentation at Stanford FLAME AI Workshop September 2023

Intelligent Downsampling (Uniform-in-Phase-Space Sampling)

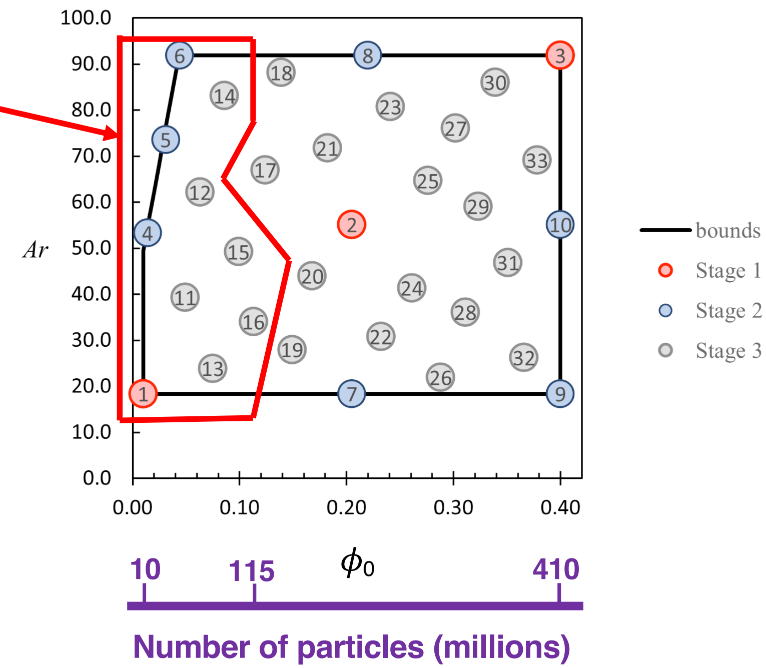
Example of Standalone Application of Uniform-in-Phase Space Sampling for Filter Size = 64



ALPEMI CONSULTING

Proof-of-Concept Demonstration of the In-Situ ML Workflow

- Selected 10 cases out of 33
  - Decision based on remaining ALCC allocation and end of allocation year deadline
- Restarted from the last saved timestep in each case
- FilterML integrated MFIX-Exa
  - Test runs with filter sizes 8, 16, 32, 64, 128
  - Streamed every single timestep
- Downsampler module:
  - Downsampled every 2 timesteps
  - Downsampled to 5% of original data
- ML Trainer module:
  - After initial training group data, training data grew as more data arrived from simulations.
  - PyTorch-based very simple neural network architecture (can be easily replaced with better one)
  - Max epoch set to 50



ALPEMI CONSULTING

Proof-of-Concept Demonstration of the In-Situ ML Workflow

Trial # 1

- Single MFIX-Exa (run-01) running 16 GPU nodes
- Filter size = 128
- 19 GPU nodes
- Total wall-clock = 20 minutes
- First successful operation of the in-situ ML workflow

Trial # y

- Ensemble MFIX-Exa runs (2 select run-xx), each running on 16 GPU nodes
- Filter size = 128
- 35 GPU nodes
- Total wall-clock = 60 minutes
- First successful operation of ensemble of MFIX-Exa simulations generating data for in-situ ML workflow

……..

Trial # yy

- Ensemble MFIX-Exa runs with number of cases going up to 10
- Decreasing Filter sizes (64,32,…) to test increasing data streamed
- Increasing GPU nodes allocated
- Total wall-clock = 1 to 12 hours
- Test stability of the In-Situ Workflow

ALPEMI CONSULTING

Proof-of-Concept Demonstration of the In-Situ ML Workflow

Trial # yyy

- Ensemble MFIX-Exa runs (10 select run-xx), each running on 16 GPU nodes
- Filter size = **16**
- **163** GPU nodes
- Total wall-clock = **20 hours execution**
- First prototype production run demonstration

```
INFO:root:Initializing Data Generator
INFO:root:Fitting scaler
INFO:root:Beginning training loop
INFO:root:Loss function: 0.09619668871164322
INFO:root:Loss function: 0.08910127729177475
INFO:root:Loss function: 0.0851670652627945
INFO:root:Loss function: 0.08261381089687347
INFO:root:Loss function: 0.0785333663225174
INFO:root:Loss function: 0.07413556426763535
INFO:root:Loss function: 0.06933750212192535
INFO:root:Loss function: 0.06570214033126831
INFO:root:Loss function: 0.06148453801870346
INFO:root:Loss function: 0.061030469834804535
INFO:root:Loss function: 0.056127648800611496
INFO:root:Loss function: 0.055097974836826324
INFO:root:Loss function: 0.048788271844387054
INFO:root:Loss function: 0.04844445735216141
INFO:root:Loss function: 0.044939033687114716
INFO:root:Loss function: 0.04389194771647453
INFO:root:Loss function: 0.04131445288658142
INFO:root:Loss function: 0.04006652534008026
INFO:root:Loss function: 0.037928592413663864
INFO:root:Loss function: 0.035487689077854156
INFO:root:Loss function: 0.03205966576933861
INFO:root:Loss function: 0.03273927420377731
INFO:root:Loss function: 0.0292130708694458
INFO:root:Loss function: 0.029631394892930984
INFO:root:Loss function: 0.02819298766553402
INFO:root:Loss function: 0.026997467502951622
INFO:root:Loss function: 0.02565888687968254
INFO:root:Loss function: 0.024672124534845352
INFO:root:Loss function: 0.02120424248278141
INFO:root:Loss function: 0.022206377238035202
INFO:root:Loss function: 0.0202163215726614
INFO:root:Loss function: 0.020730257034301758
INFO:root:Loss function: 0.01914270594716072
INFO:root:Loss function: 0.01719050668179989
INFO:root:Loss function: 0.018636707216501236
INFO:root:Loss function: 0.017125103622674942
INFO:root:Loss function: 0.016454549506306648
INFO:root:Loss function: 0.0157964788377285
INFO:root:Loss function: 0.01608632318675518
INFO:root:Loss function: 0.01502862386405468
INFO:root:Loss function: 0.014910456724464893
INFO:root:Loss function: 0.013407580554485321
INFO:root:Loss function: 0.014263911172747612
INFO:root:Loss function: 0.014529166743159294
INFO:root:Loss function: 0.013453206047415733
INFO:root:Loss function: 0.013202277943491936
INFO:root:Loss function: 0.01366801280528307
INFO:root:Loss function: 0.013635891489684582
INFO:root:Loss function: 0.013453234918415546
INFO:root:Loss function: 0.01204115990549326
INFO:root:TIME ELAPSED Training step 1.4363384540192783
INFO:root:Saving the model
```

Log of the trainer module showing decreasing loss function progress after 50 epochs

Started showing up on NERSC Website Top Running Jobs List

Proof-of-Concept Demonstration of the In-Situ ML Workflow

Trial # Last

- Ensemble MFIX-Exa runs (10 select run-xx), each running on 32 GPU nodes
- Filter size = **8**
- **323** GPU nodes
- Total wall-clock = **12 hours**
- First time filter size= 8 tested for long duration

  ~1.3e6 rows /timestep

NERSC Website showing In-situ ML Training as the Top Running Job

ALPEMI CONSULTING

Concluding Remarks and Observations

- Proof-of-concept demonstration of a first-of-its-kind, scalable capability to perform in-situ ML training for multiphase flow simulations.
  - Leveraged all open-source tools and frameworks
    - MFIX-Exa(NETL), SmartSim (HPE), Uniform-in-Phase-Space Sampling (NREL)
  - Implemented an in-situ intelligent downsampling methodology coupled with the ML training, which can be on and off based on needs.
  - Demonstrated the scalability of the workflow from 16 to 323 GPU nodes of Perlmutter (could have gone higher but ran of time & space).
- As part of the ALCC project outcome, raw dataset for the 33 simulations is available for researchers interested to explore with their ML models at

NETL EDX: https://edx.netl.doe.gov/dataset/mfix-exa-alcc2324-run-data

ALPEMI CONSULTING

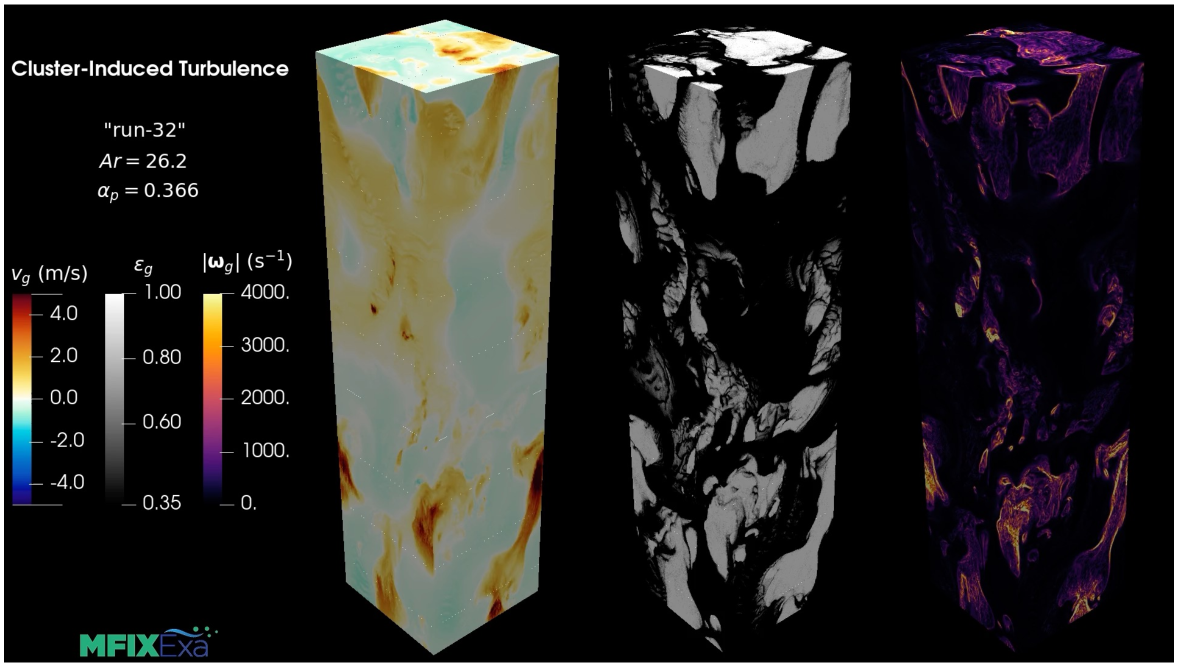
Motivation and Objectives

- **Objectives:**
  - Develop a Real-Time (In-Situ) ML Workflow: ✓
    - **Goal**: Construct a flexible and scalable in-situ ML workflow as a proof-of-concept demonstration.
    - **Focus**: Ensure that the workflow integrates seamlessly with existing simulation tools and scales effectively with data size and computational resources.
  - Identify Bottlenecks: ✓
    - **Task:** Diagnose and address any performance or integration issues within the in-situ ML workflow**.**
    - **Outcome:** Provide insights into the limitations and areas for improvement.
  - Demonstrate Proof-of-Concept Workflow Operation: ✓
    - **Integration:** Showcase the complete workflow with MFIX-Exa, a state-of-the-art multiphase flow CFD code.
    - **Scalability:** Validate the workflow's capability to handle large-scale simulations across multiple nodes.
  - Conduct Comparative Studies: (To be completed subject to allocation)
    - **Model Accuracy:** Assess and compare the accuracy of ML models developed in-situ versus offline.
    - **Data Write-Out Frequency:** Develop guidelines on optimal data write-out frequency to balance between ML model accuracy and computational efficiency.

Concluding Remarks and Observations

- Many interesting potential research directions to explore:
  - In-situ construction of multiple ML models concurrently with different architectures (e.g. PINNs, Neural Operators, mixed precision) for the same dataset and automatic decision support to identify the best one.
  - Identify the optimal frequency of I/O for off-line processing for a given level of ML model accuracy by comparing with in-situ based ML models.
  - Assess the effect of intelligent downsampling vs. direct use of raw data.
  - Automatic decision support to enable/disable intelligent downsampling (uniform-in-phase space sampling) to avoid stalls in the ML pipeline.
  - Phase-space sampling targeting multiple quantities of interest rather than one.
  - In-situ statistical outlier detection to detect anomalies during the simulations rather afterwards.
  - Integration with Nodeworks to lower the barrier and enable GUI based in-situ ML workflow construction and deployment.
  - Computational steering by leveraging the in-situ ML model constructed and reinforcement learning.

Thank you for your attention.
Questions?



Animation of one of the cases within ALCC simulation campaign showing velocity, void fraction and worticity (William Fullmer@NETL)

Acknowledgments & Credits:

Mr. Alper B. Gel, (BioEng/EECS Depts., UC Berkeley) for ML model development and implementation of Accelerate library from Hugging Face, which enabled the PyTorch code to run across any distributed GPU configurations.

Dr. Malik Hassanaly (NREL) for discussions and collaboration enabling the use of Uniform-in-phase-space data selection and sampling framework.

Dr. Aytekin Gel was supported by ALPEMI Consulting, LLC for the in-situ ML workflow development project using ALCC dataset.

The simulation campaign in this research was carried out with the competitively selected allocation award of 100,000 GPU node hours under the ASCR Leadership Computing Challenge (ALCC) program of Office of Science of the U.S. Department of Energy.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award ALCC-ERCAP0029910.

ALPEMI CONSULTING