SAND2021-6130C

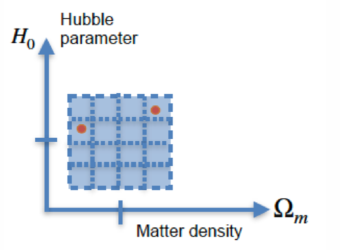
ECP

ExaLearn: Co-Design Center for Exascale Machine Learning Technologies

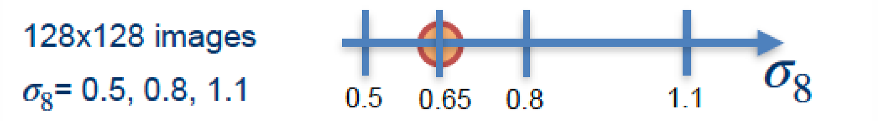
PI: Frank Alexander, Brookhaven National Laboratory (falexander@bnl.gov)
Partner PIs and Institutions: Ian Foster, Argonne National Laboratory; Christine Sweeney, Los Alamos National Laboratory; Peter Nugent, Lawrence Berkeley National Laboratory; Brian Van Essen, Lawrence Livermore National Laboratory; Sudip Seal, Oak Ridge National Laboratory; James A. Ang, Pacific Northwest National Laboratory; Michael Wolf, Sandia National Laboratories

SURROGATES

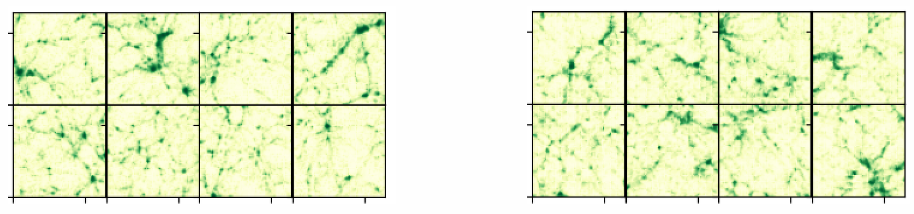
- **Definition:** Create a surrogate model (or emulator) to replace computationally expensive simulations through machine learning (ML), cheaply.
- **Method Used:** Generative adversarial networks (GAN) and hybrid autoencoders.
- **Initial Problem:** Train on existing cosmological simulations from simple *n*-body to full-physics hydrodynamical sims interpolating cosmological parameters. (https://petreldata.net/exalearn)
- **Software:** CosmoGAN, CosmoFlow, LBANN, and Lya-demo.
- **Results:** Accurately build conditional GANs to interpolate.
- **Next Steps:** Incorporate CosmoGAN into LBANN and work with larger three-dimensional sims while exploring other simulation capabilities: Combustion-Pele, ExaStar, etc.



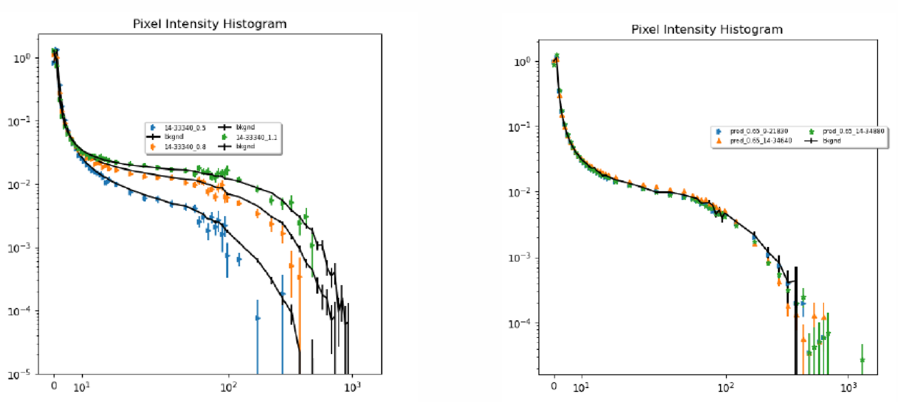
Typical cosmological simulations look at ~7 parameters. Two are seen to the left, the Hubble constant (current expansion rate) and Ω_M (matter density). For each set of parameters, we need to generate ~1000 images to train our GANs, can we interpolate on this grid?



Using smaller images, 128^2, trained at three values of σ_8(0.5, 0.8, and 1.1) which measures the amplitude of the linear power spectrum on the scale of 8*H_0/100 Mpc, we will try to use a CGAN to interpolate at 0.65 with fixed H_0 and Ω_M.

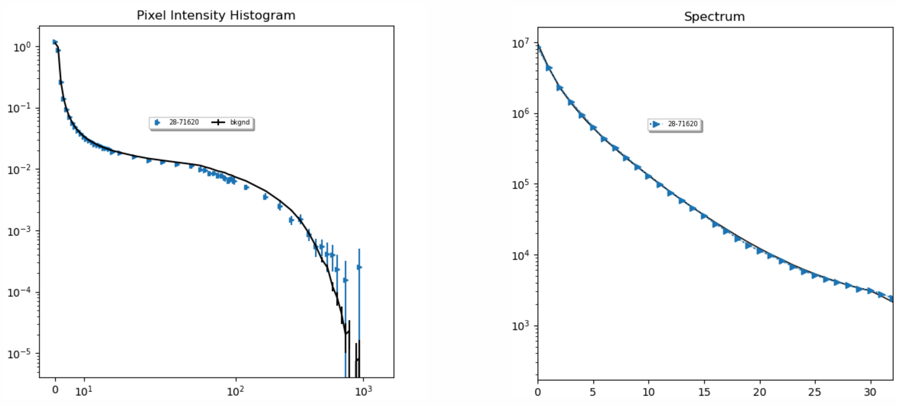


pyCOLA (Tassev et al., 2014) simulated (left) and GAN-generated (right) images based on training from more than 10,000 pyCOLA n-body cosmology simulations.



Pixel intensity histograms of the fixed values for the simulations we trained on (right) and the interpolated σ_8 = 0.65 (left). The interpolated CGAN-trained cosmological simulation matches nicely with the blinded set of pyCOLA simulations.

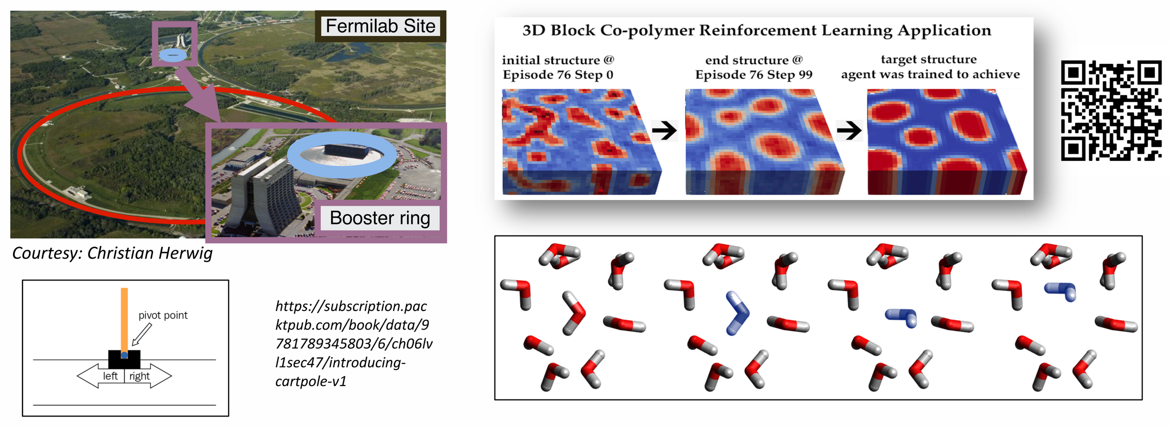
Next steps: Move from two- to three-dimensional using LBANN and expand the parameters we interpolate on from 1 to 2-3.



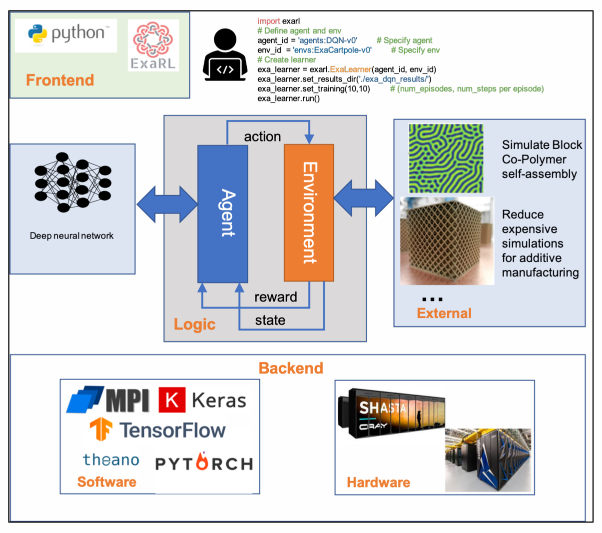
A histogram of the pixel intensity (left) and the power spectrum or 2-pt correlation function (right) for a pyCOLA simulated and GAN generated images for values a single set of cosmological parameters.

CONTROL

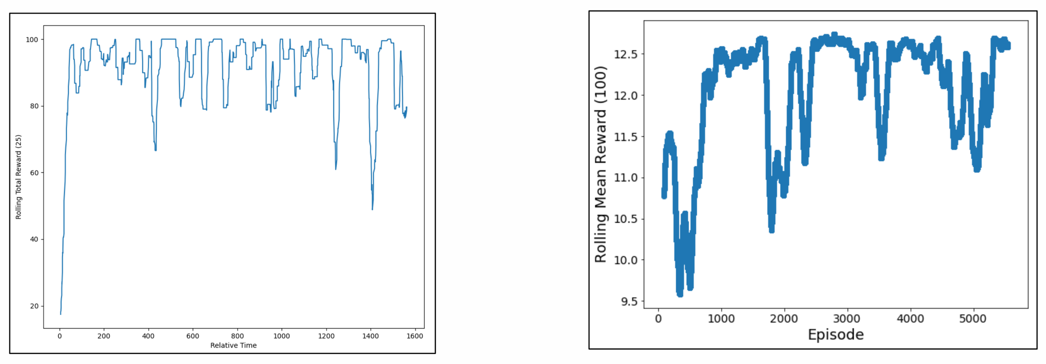
- **Definition:** Efficient exploration of complex problem spaces
- **Methods Used:** Reinforcement learning (RL) and surrogate models
- **Problems:** 1) Accelerator control for Booster at Fermi National Laboratory (FNAL), 2) Block copolymer (BCP) self-annealing control, 3) Water cluster molecular design, and 4) Scalable version of proxy application for balancing pole on cart (ExaCartPole).
- **Software:** EXARL scalable RL framework AND applications: 1) Neural network (NN)-based digital twin of FNAL Booster, 2) BCP partial differential equation (PDE)-based simulations, 3) NN-based environment for water cluster, 4) ExaCartPole multi-MPI-rank physics-based environment (scalable "Hello world" for RL).
- **Results:** Functioning RL applications using scalable EXARL framework: 1) ExaBooster, 2) ExaCH (BCP control), 3) ExaWaterCluster, and 4) ExaCartPole proxy application. EXARL scalable framework. Prototype RL application performance monitoring tools.
- **Next Steps:** Continued scaling of EXARL, proxy application distribution (discrete and continuous action space), continued integration of ExaWaterCluster into EXARL.



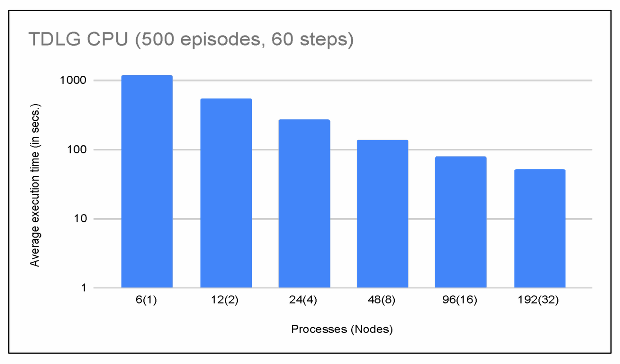
RL applications: FNAL Booster control (upper left), cart pole proxy application (lower left); BCP self-annealing control and QR code for BCP demo (upper right); water cluster design (lower right).



EXARL scalable RL system components and workflow.



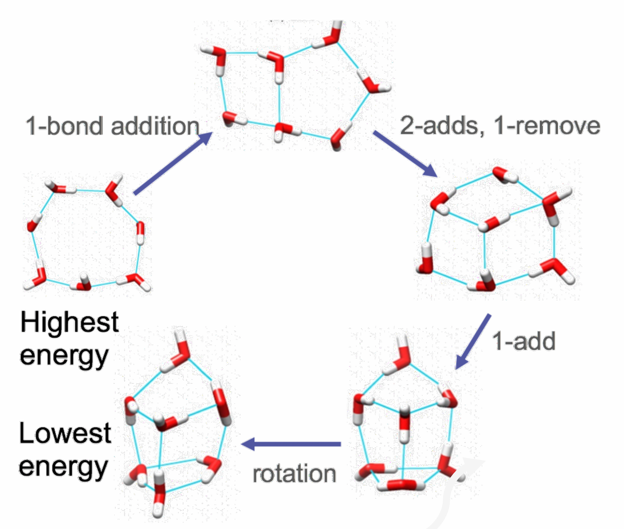
RL convergence plots for ExaCartPole (left, 6-rank environment) and ExaWaterCluster (right) problems using EXARL. Reward increases with training episodes.

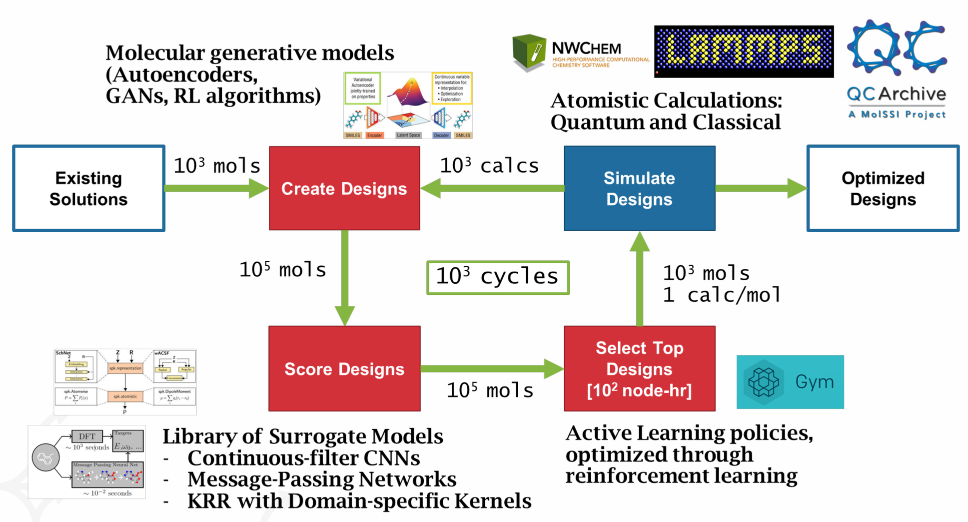


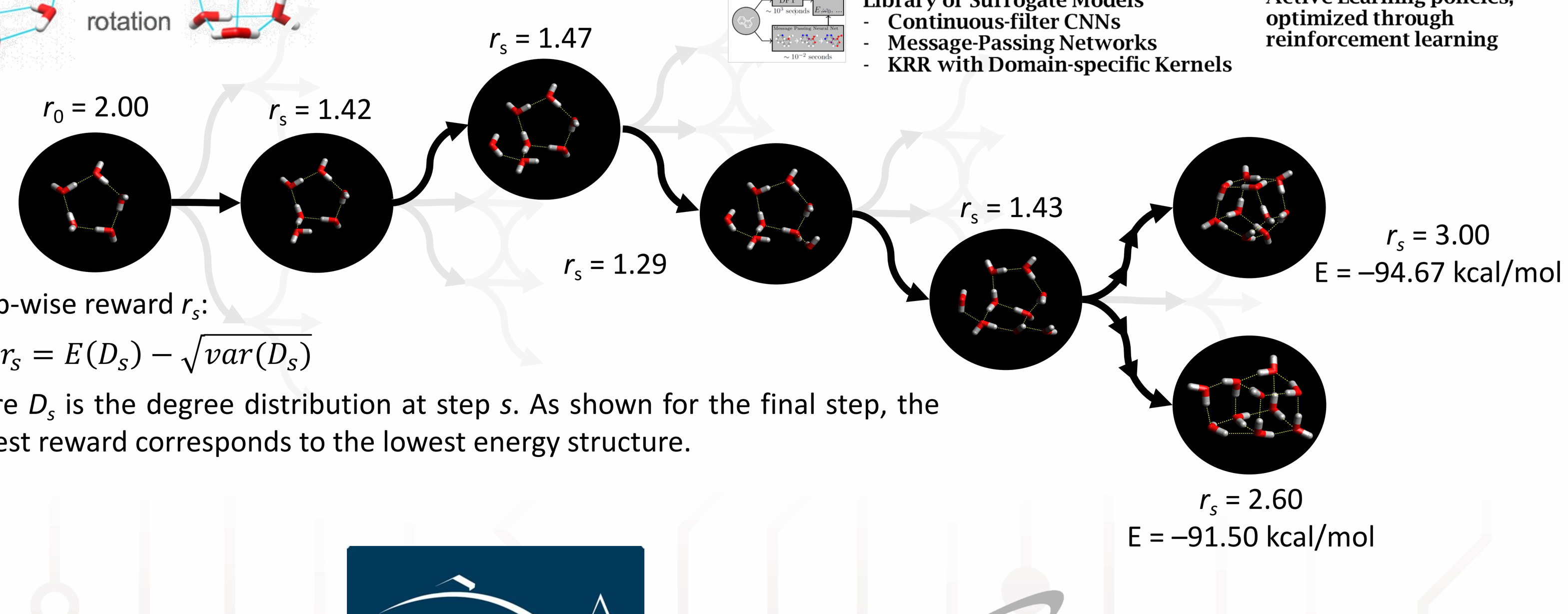
EXARL used by 3D BCP app on 32 nodes Summit. Training on GPU, environment on CPU.

DESIGN

- **Definition:** Solving optimization problems with simulations steered by machine learning (ML) and optimal experimental design methods
- **Methods Used:** Bayesian optimization, message passing neural networks, Reinforcement learning.
- **Initial Problems:**
  1) Generate clusters of water molecules for quantitative examination of the nature and magnitude of intermolecular interactions in liquid water.
  2) Designing molecules for performant and safe electrolytes in next-generation Li-ion batteries out of trillions of candidates.
- **Software:** Library of ML methods for graph generation, Colmena—an HPC toolkit for steering ensemble simulations with machine learning.
- **Results:** Early EXARL implementation for water clusters; Bayesian optimization for oxidation-resistant electrolytes on 512 Theta nodes.
- **Next Steps:** Surrogate models for NWChemEx; water cluster optimization with EXARL.







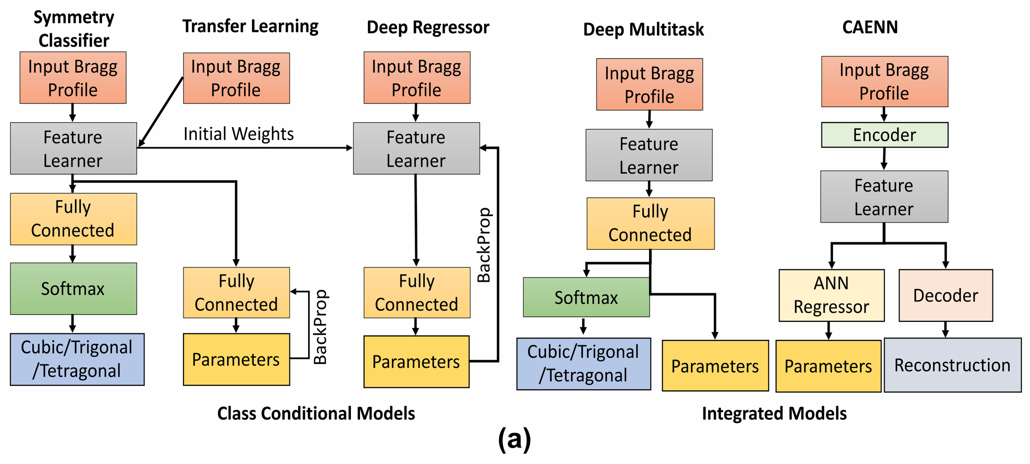
Step-wise reward r_s:

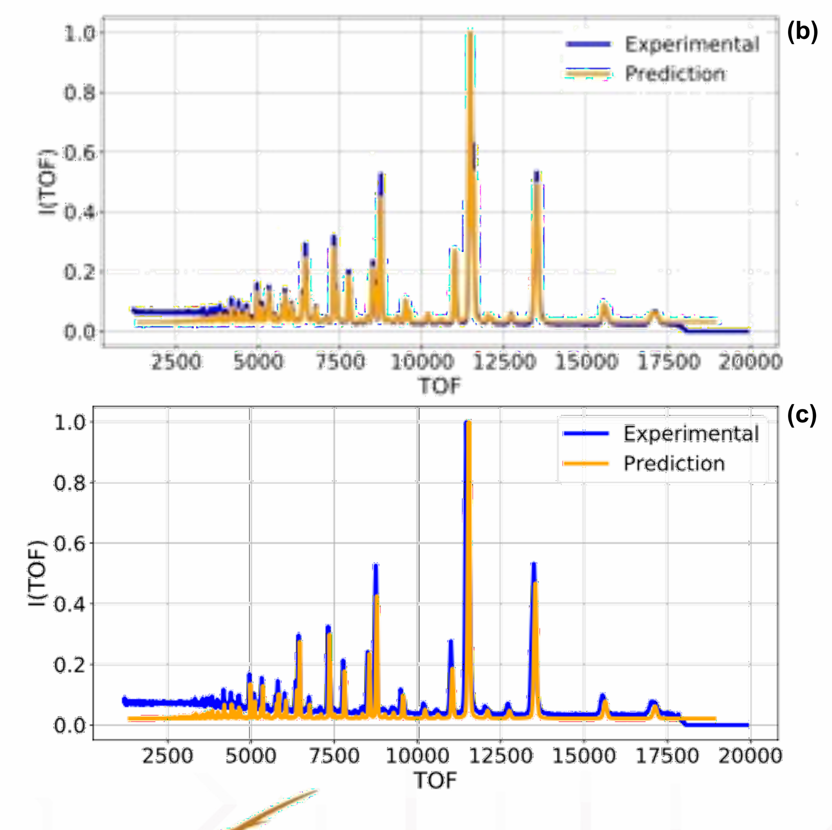
$$r_s = E(D_s) - \sqrt{var(D_s)}$$

where D_s is the degree distribution at step s. As shown for the final step, the highest reward corresponds to the lowest energy structure.

INVERSE PROBLEMS

- **Definition:** Use machine learning (ML) methods to solve the inverse problem of predicting material structures from X-ray or neutron scattering profiles**.**
- **Methods Used:** Transfer Learning, Multitask Networks, Convolutional Autoencoder.
- **Initial Problem:** Design a classifier to determine crystallographic symmetry and a regressor to predict unit cell parameters of a known perovskite material from its neutron scattering (Bragg) profiles.
- **Software:** GSAS-II for generation of labeled examples, Keras; Scikit-learn.
- **Results:** Two categories of models–**class-conditiona**l and **integrated**–were trained and evaluated. The former relies on a two-stage inference pipeline in which a crystallographic class label is first predicted followed by regression to predict the length/angle parameters. In the latter category, the classification and regression tasks are performed as a single learning task. These models were trained on synthetically generated data of three different symmetry classes, validated against experimental observations, shown that integrated models outperform class-conditional models and predicted with MSE ~ $O(10^{-3})$.
- **Next Steps:** Build labeled examples of Bragg profiles that sample complete parameter space of all seven crystallographic symmetry classes; build deep learning models that predict symmetry classes and cell parameters of all seven crystallographic symmetries.





(a) Classifier (left), class-conditional and integrated models. (b) CAENN integrated model prediction compared with experimentally observed tetragonal sample with cell lengths a = 3.9851 and c = 4.0358. (c) CAENN integrated model prediction compared with experimentally observed tetragonal sample with cell lengths a = 4.0196 and c = 4.0210.

For more details: **C. Garcia-Cardona, R. Kannan, T. Johnston, T. Proffen and S. K. Seal, "Structure Prediction from Neutron Scattering Profiles: A Data Sciences Approach," 2020 IEEE International Conference on Big Data, pp. 1147-1155, 2020.**

Argonne National Laboratory · Berkeley Lab · Brookhaven National Laboratory · Lawrence Livermore National Laboratory · Los Alamos National Laboratory EST. 1943 · Oak Ridge National Laboratory · Pacific Northwest National Laboratory · Sandia National Laboratories

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.