AI Technologies Overview

Francis J. Alexander

BROOKHAVEN NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

@BrookhavenLab

ExaLearn: Co-design Center for Exascale Machine Learning Technologies

Project PI: Francis J. Alexander, Brookhaven Lab

Partner PIs and Institutions:

- Ian Foster, ANL
- Aric Hagberg, LANL
- Peter Nugent, LBNL
- Brian Van Essen, LLNL
- David Womble, ORNL
- James A. Ang, PNNL
- Michael Wolf, SNL

EXASCALE COMPUTING PROJECT

NNSA National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

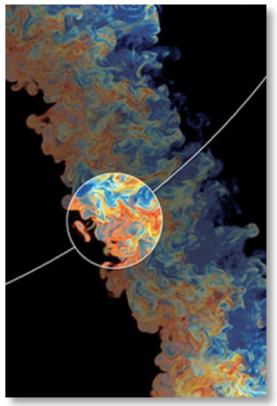
AD subprojects target national problems in DOE mission areas

National security

Next-generation, **stockpile stewardship** codes

Reentry-vehicle-environment simulation

Multi-physics science simulations **of high-energy density physics** conditions



Energy security

Turbine **wind plant** efficiency

Design and commercialization of **SMR**s

Nuclear fission and fusion reactor **materials design**

Subsurface use for **carbon capture**, petroleum extraction, waste disposal

High-efficiency, low-emission **combustion engine** and gas turbine design

Scale up of **clean fossil fuel** combustion

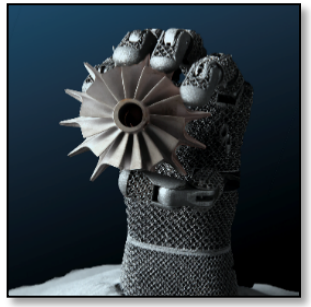
Biofuel catalyst design

Economic security

Additive manufacturing of qualifiable metal parts

Reliable and efficient planning of the **power grid**

Seismic hazard risk assessment



Scientific discovery

Cosmological probe of the standard model of particle physics

Validate fundamental laws of nature

Plasma wakefield accelerator design

Light source-enabled **analysis of protein and molecular structure** and design

Find, predict, and control materials and properties

Predict and control **magnetically confined fusion plasmas**

Demystify **origin of chemical elements**

Earth system

Accurate regional impact assessments in **Earth system models**

Stress-resistant crop analysis and catalytic conversion of **biomass-derived alcohols**

Metagenomics for analysis of biogeochemical cycles, climate change, environmental remediation

Health care

Accelerate and translate **cancer research** (partnership with NIH)

This is a diverse portfolio of applications!

Overarching Goals for ExaLearn

- Provide exascale ML software for use by:
  - ECP Applications Projects
  - Other ECP Co-design Centers
  - DOE Experimental Facilities
  - DOE Leadership Class Computing Facilities

- Establish multidisciplinary collaborations in learning technologies that cross-cut ECP projects:
  - AD projects that share an interest in ML methods
  - ST projects
  - HI/PathForward projects
- Key idea is to leverage ongoing ML efforts at labs, extend them to new projects, and find a way to draw new ideas out of their original projects without requiring unfunded mandates.

Guiding Principles

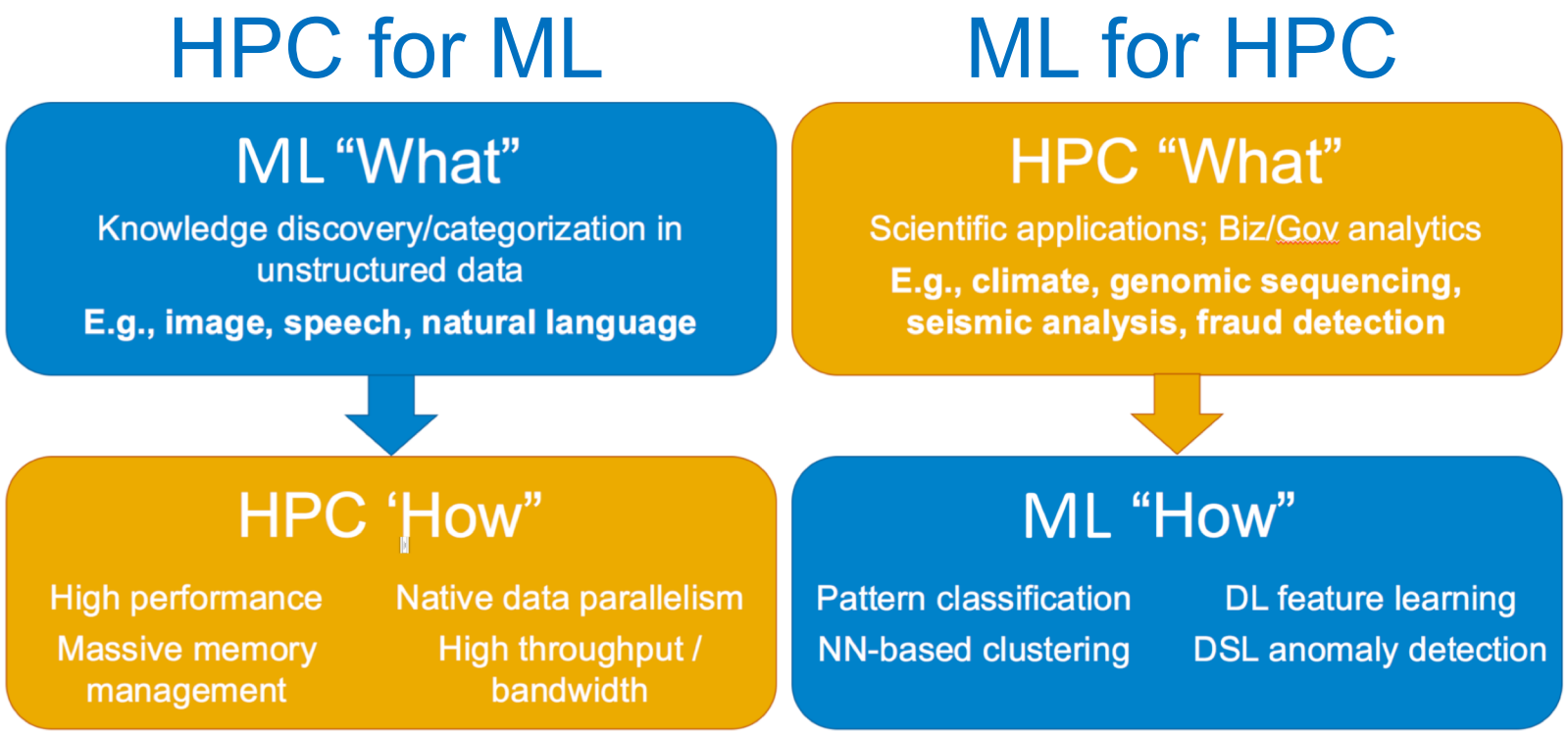
- ExaLearn produces a **Software Toolset** that:
  - Is applicable to multiple problems within the DOE mission
  - Has a line-of-sight to exascale computing, e.g., uses exascale platforms directly or provides essential components to an exascale workflow
  - Does not replicate capabilities easily obtainable from existing, widely available packages
  - Builds in domain knowledge where possible (not often done industry, although efforts beginning at IBM, GE, etc.); “physics”-based ML and AI are recurring themes
  - Quantifies uncertainty in a predictive capacity
  - Is both interpretable and reproducible
  - Is based on mathematically well-grounded methods
    - For example, some nice theory now for GANs, but more work needs to be done.

Application Priorities Determine Machine Learning Methods

ExaLearn focuses on employing the "right tool for the job"

- Deep Learning (CNN, RNN, etc.)
- Ensemble Methods and Random Forest Methods
- Reinforcement Learning
- Kernel Methods
- Tensor Methods
- Graph-Based Learning
- Large-scale System Integration (combining traditional HPC workloads with machine learning)

Relationships between Machine Learning and HPC



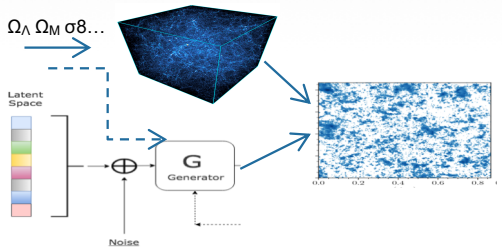
Courtesy of: Gadi Singer

- HPC for Machine Learning: HPC technologies are applied to learning tasks to accelerate computation and/or solve larger problems.
- Machine Learning for HPC: Learning technologies are applied to HPC computations to improve their performance in some way, e.g., by choosing the next simulation(s) to perform.

ExaLearn Application Pillars

Surrogates

- ML-created models
- Faster and/or higher fidelity models
- Generative networks
- Using ML to replace complicated physics
- Cosmology



Control

- ML-controlled experiments
- Efficient exploration of complex space
- Reinforcement Learning
- Use RL agent to control light source experiments
- Temperature control for Block Co-Polymer (BCP) experiments

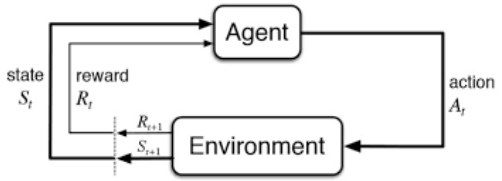
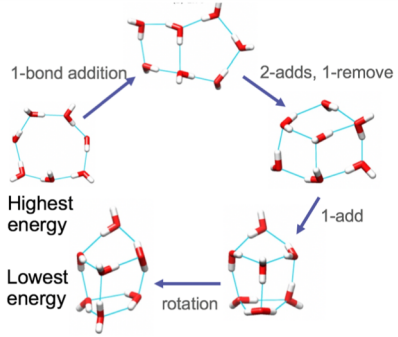


Image courtesy Sutton, Barto, Reinforcement Learning 2017

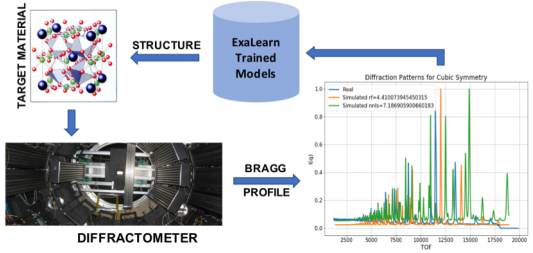
Design

- ML-created physical structures
- Optimized proposal for desired behavior of structure within complex design space
- Graph-Convnets
- Use Graph-CNN to propose new structures that respect chemistry
- Molecular Design



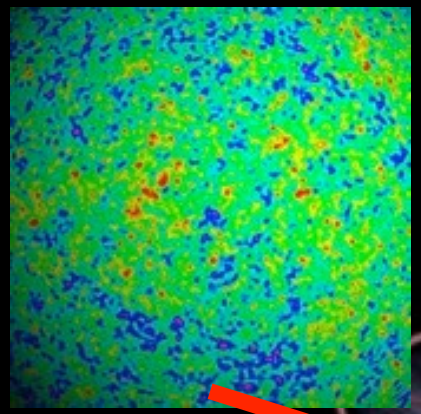
Inverse

- ML projection from observation to original form
- Back-out complex input structure from observed data
- Regression models
- Predicting crystal structure from light source imaging
- Material structure from neutron scattering

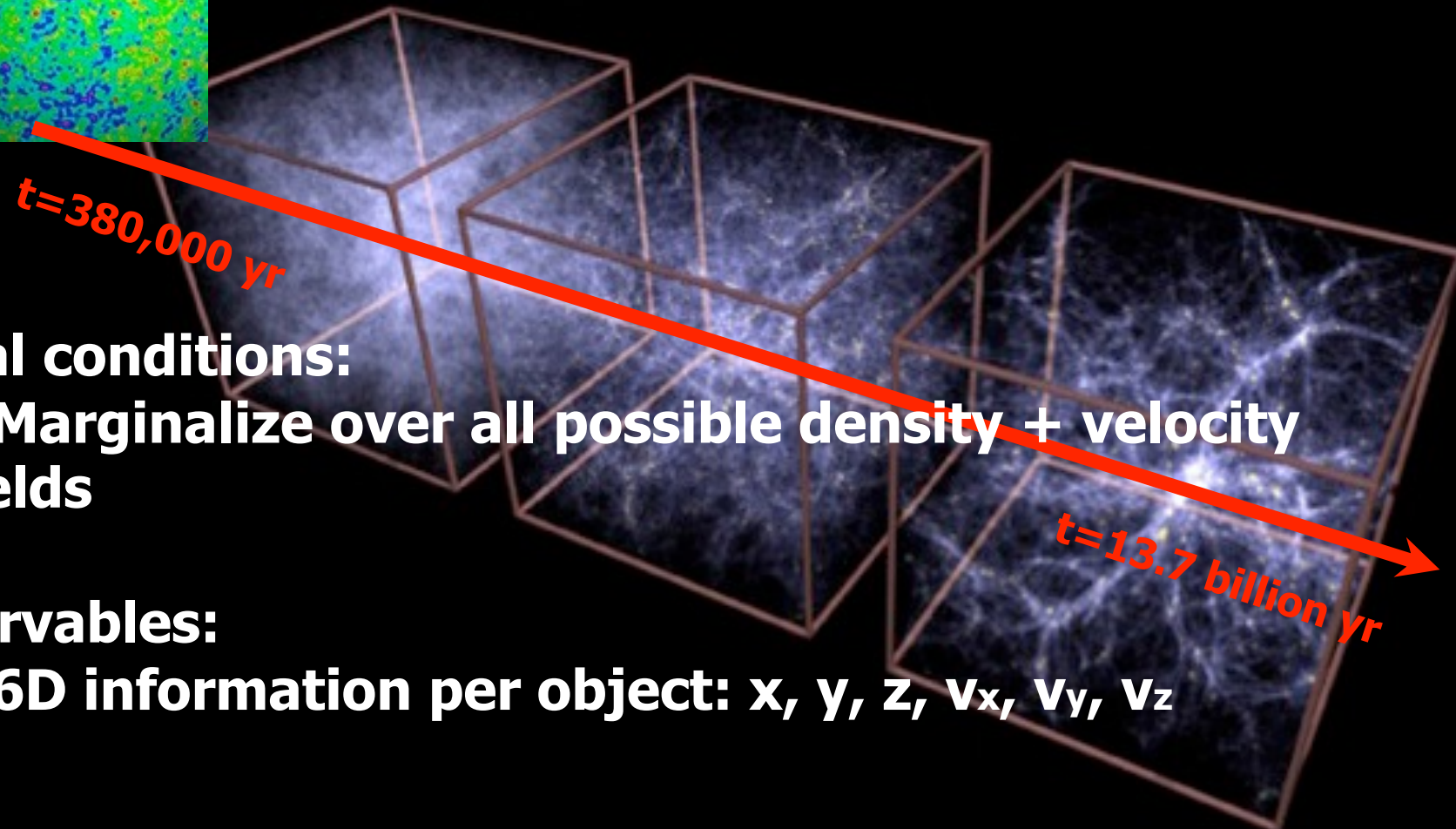


EXASCALE COMPUTING PROJECT

Fitting the Universe



$\mathcal{L}$(Our Universe| initial conditions, forces)



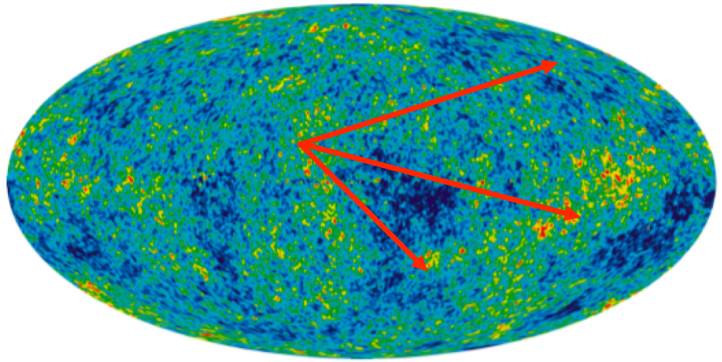
Initial conditions:

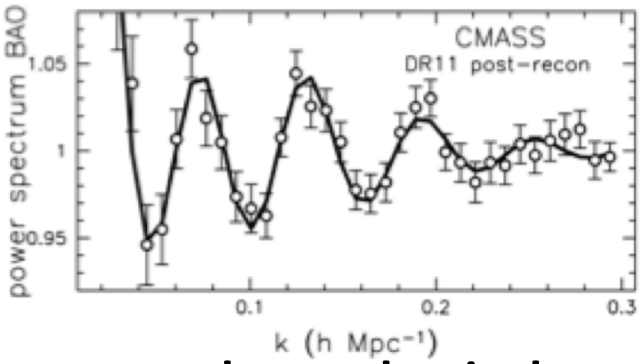
Marginalize over all possible density + velocity fields

Observables:

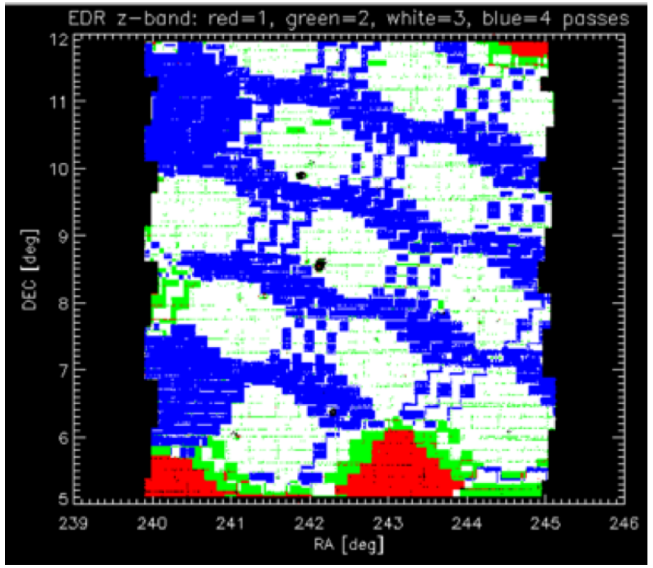
6D information per object: x, y, z, v_x, v_y, v_z

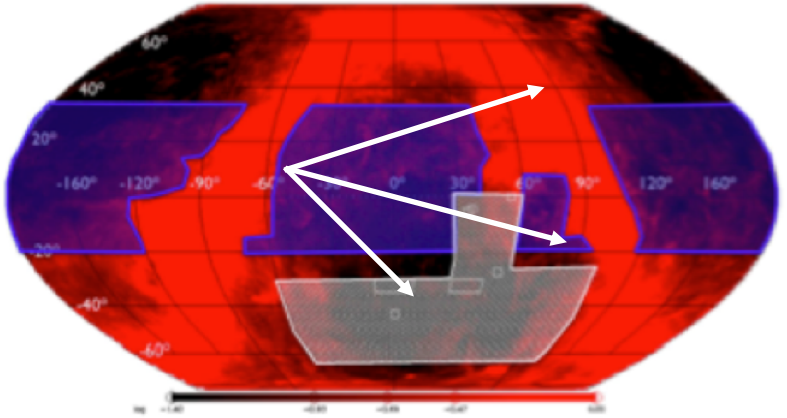
Why do we need simulations?



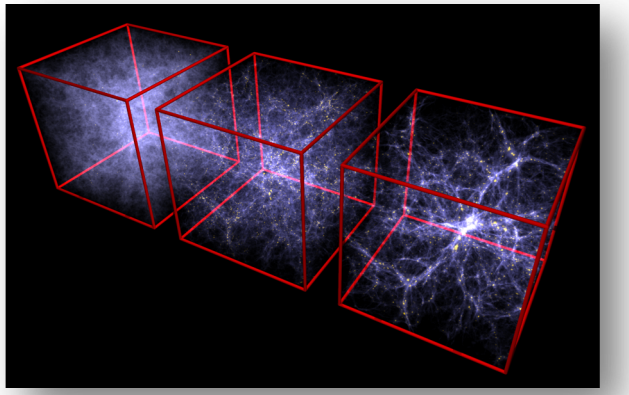


Two point correlation functions are a way to measure cosmology: what is the power spectrum of distances between every galaxy and every other galaxy as a function of time (redshift)? It requires a deep understanding of galaxy selection, completeness and the systematics of each.





Surrogates: Realistic Simulations on the Cheap

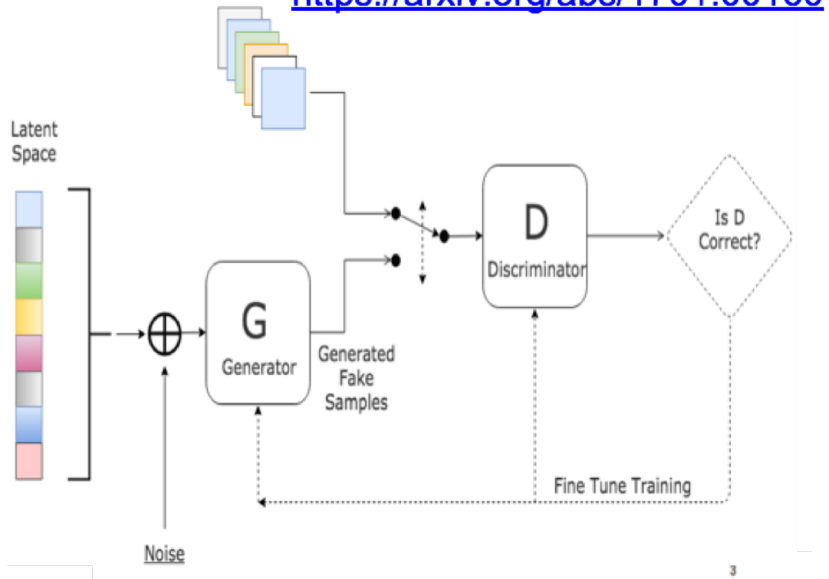


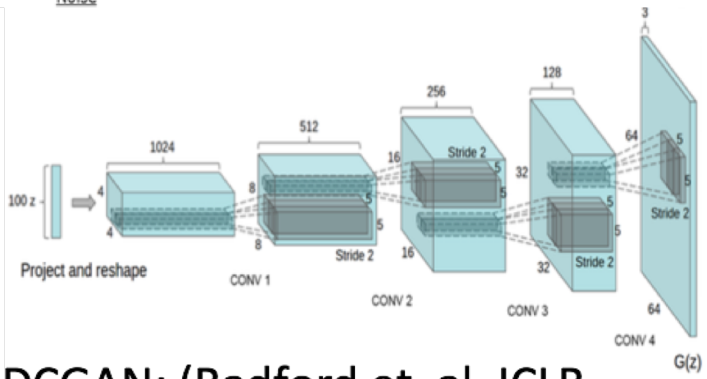
- **Challenge and Importance**: Many DOE simulation efforts could benefit from having realistic surrogate models in place of computationally expensive simulations. These can be used to quickly flesh out parameter space, help with real-time decision making and experimental design, and determine the best areas to perform additional simulations. We are targeting large-scale structure simulations of the universe. As the field is well developed, the scale can easily be ramped up to an exascale ML challenge, and the field is robust enough to explore systematics at the sub-percent level.
- **ML impact**: Neural-networks-based generative models can make reliable surrogate models of expensive simulations for data augmentation purposes. Such surrogate models can be used to aid in cosmological analysis to reduce systematic uncertainties in observations.
- **Timeliness**: The ExaSky application project is producing the largest LSS simulations now, the DESI experiment starts next year, and LSST takes its first science images in 2021.
- **Urgency**: All cosmological measurements today are limited by systematics, not statistics. To reduce these uncertainties and make the most of these future experiments, thousands (if not millions) of exascale-sized simulations will need to be carried out. Surrogate models are a viable path forward to achieve this goal—but only if their limitations are fully understood.
- **Benefit to ECP-Large DOE Experiments**: Once demonstrated, this software framework can be easily adapted to other fields and simulation areas, such as combustion.

Deep Learning to the Rescue?

- Jointly optimize Discriminator (D) and Generator (G) NNs
  - ✦G architecture like decoder in ConvAE
  - ✦Loss for G/D in opposition
- On 'natural images' GANs can be unstable, our problems have advantages:
  - ✦underlying physics structure
  - ✦existing, labeled simulation samples
  - ✦metrics to evaluate
- Build on industry research – e.g. convolutional DCGAN

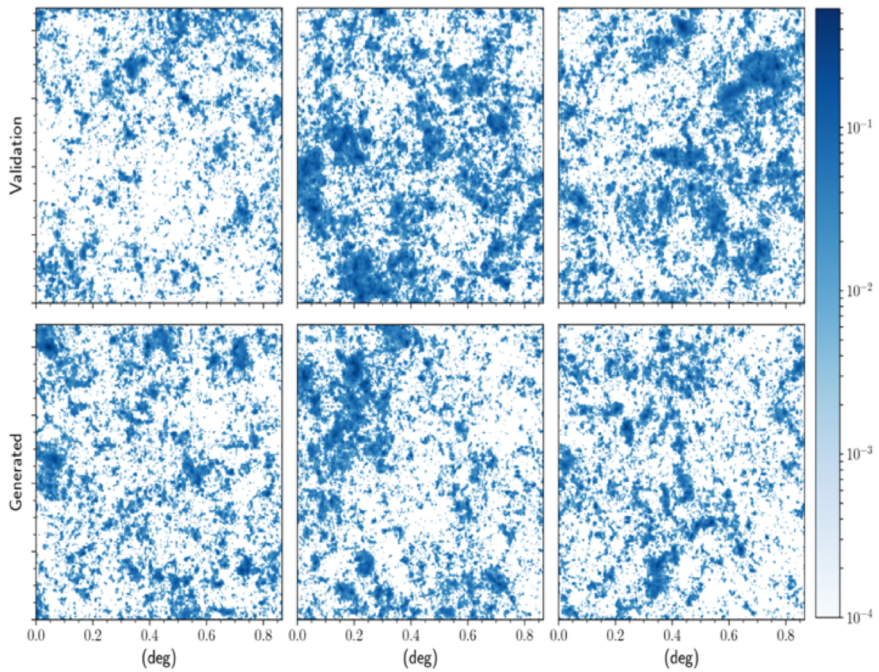
Goodfellow *et al.*
https://arxiv.org/abs/1406.2661
https://arxiv.org/abs/1701.00160

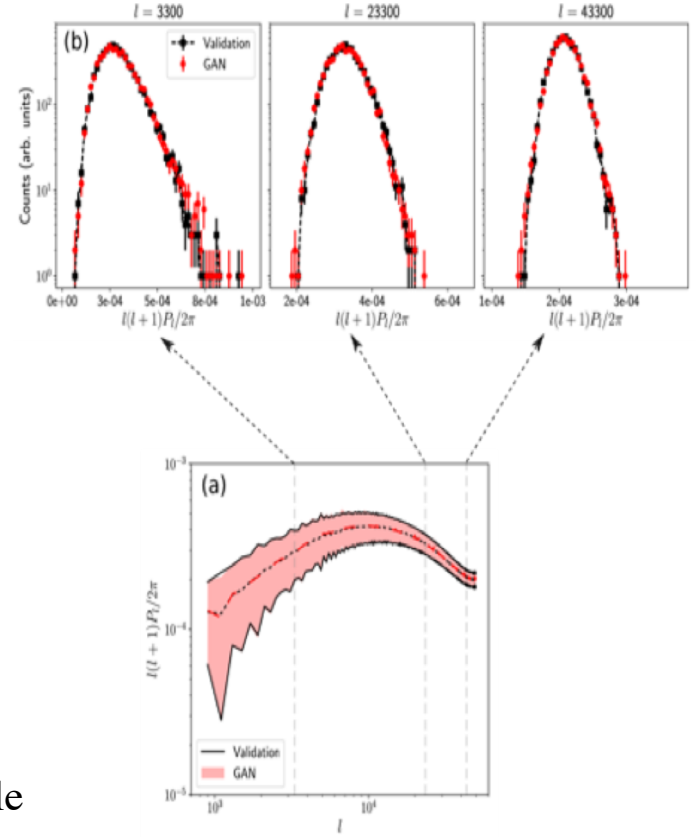




DCGAN: (Radford et. al. ICLR 2016)https://arxiv.org/abs/1511.06434

CosmoGAN

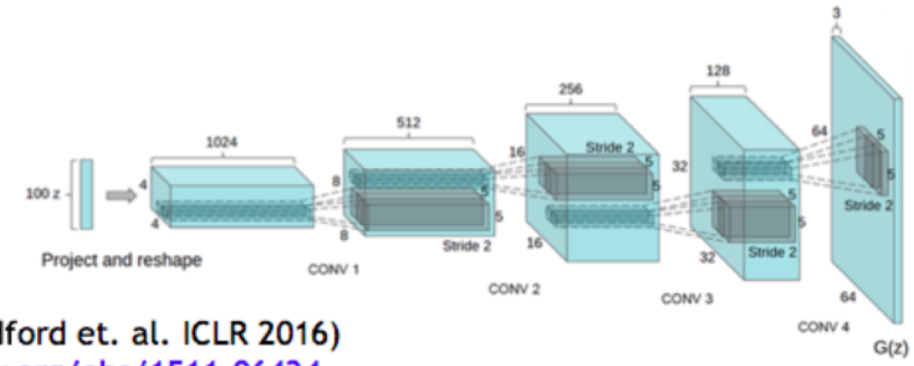




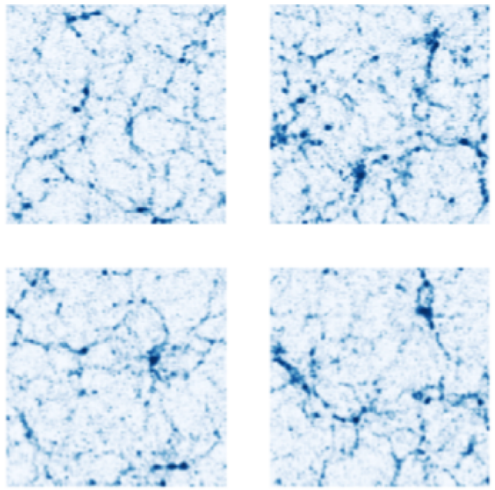
- Calculate power spectrum for generated images and validation sample
- Excellent agreement (K-S p_value > 0.995 for 246/248 moments)
- GAN not explicitly trained to reproduce these distributions
- *Also higher-order Minkowski functionals are reproduced*

GANs for cosmology

- Building on success of CosmoGAN “1.0” (weak lensing convergence maps), use the widely-used DCGAN network architecture
  - Simple CNN setup works well for scaling up network size and parallelizing the model for large inputs



DCGAN: (Radford et. al. ICLR 2016)
https://arxiv.org/abs/1511.06434

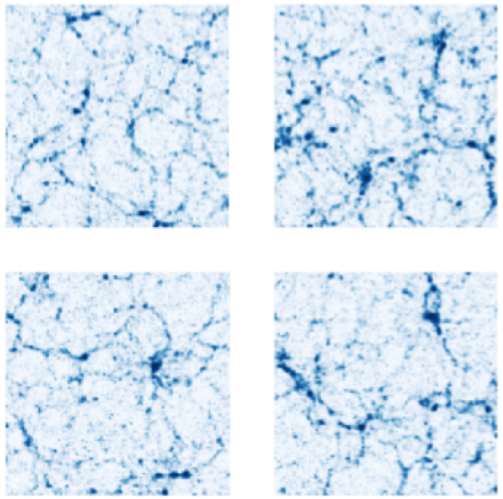


Looking beyond the good press…

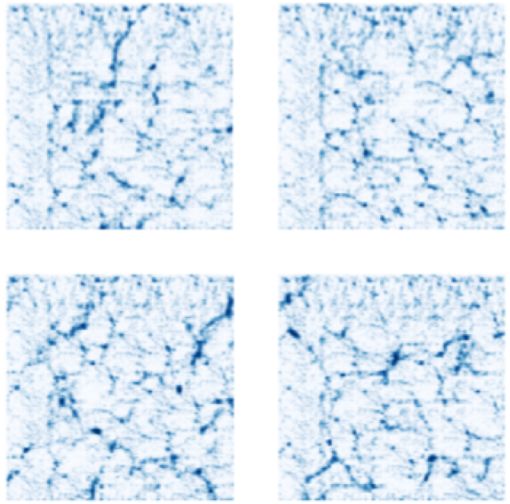
GANs for cosmology: pitfalls & challenges

- GANs can achieve high sample quality but are notoriously brittle
  - As G gets closer to the real data manifold, gradients from D are unbounded
  - Loss functions do not strongly correlate with sample quality, **mode collapse** is common

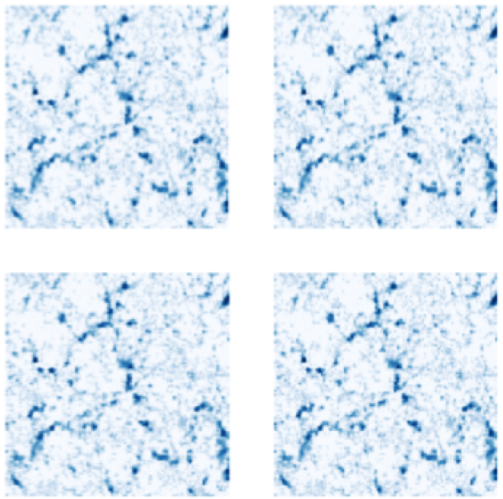
Real samples



Partial mode collapse



Complete mode collapse



Making things Stable

GAN regularizers to improve stability

Mitigate instability by **adding regularization terms** to the objective functions

From "non-science DL" literature (dataset agnostic):

- Gradient penalties ("R1 regularization") $R_1(\psi) := \frac{\gamma}{2} \mathrm{E}_{p_{\mathcal{D}}(x)} \left[\|\nabla D_\psi(x)\|^2 \right]$
  - DL theory: R1 stabilizes training dynamics close to Nash equilibrium
- Feature matching $||\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \mathbf{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \mathbf{f}(G(\boldsymbol{z}))||_2^2$
  - Penalize generator to match the statistics of the intermediate feature maps in the discriminator

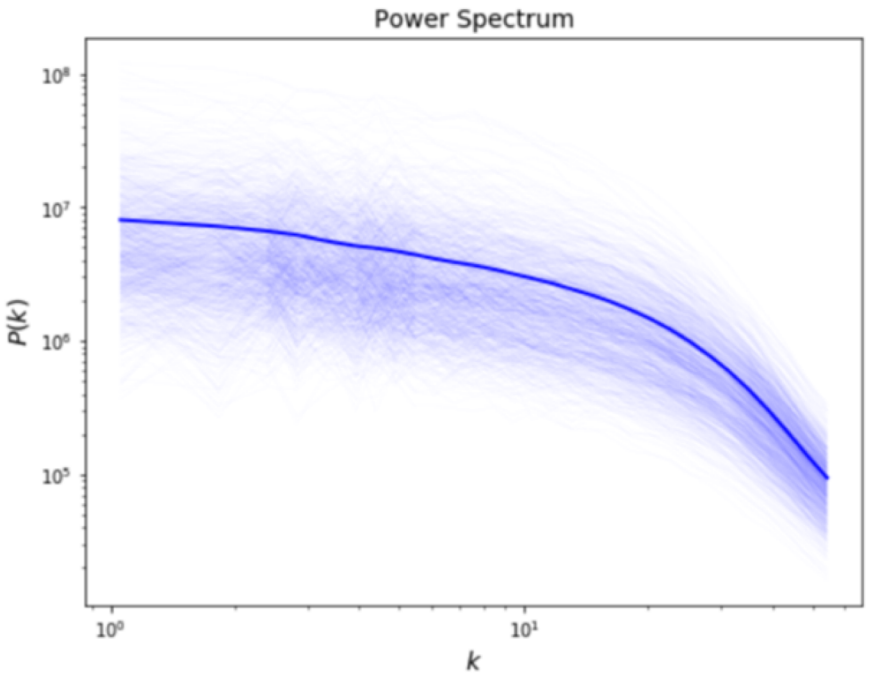
Adding Physics to the GANs

Physically-motivated constraints

- Additional loss term(s) to push generator towards generating samples with a **realistic power spectrum** *P(k)*, one of the target statistics
  - Define $\mathcal{L}_{\text{spec}}$ such that the mean and variance of *P(k)*, per *k* bin, matches expected distribution

$$\mathcal{L}_{\text{spec}} = \log ||Q(P_{\text{generated}}(k)) - Q(P_{\text{target}}(k))||_2^2$$

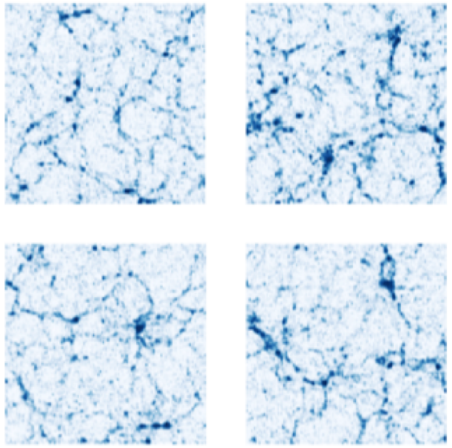
  - Backprop through power spectrum computation (2D FFTs + binning over $|k|$) so generator gets useful gradients

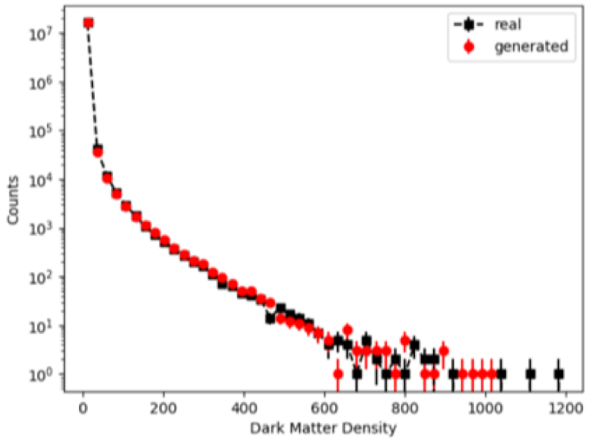


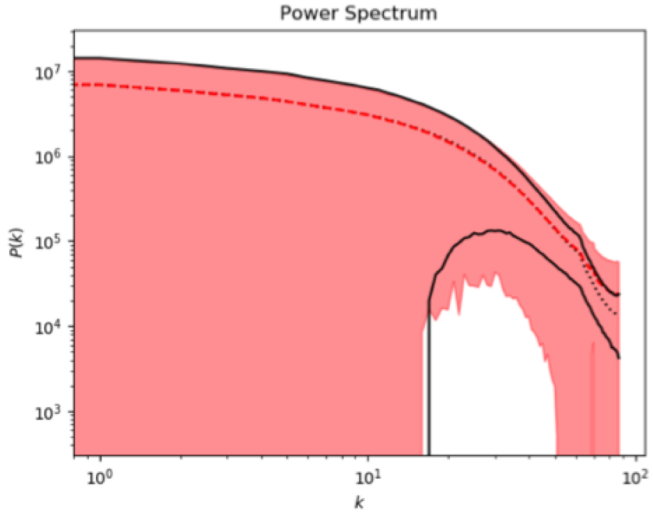
Current 2-D State-of-the-Art

Results: ML regularizers + *P(k)* constraint

- With constraints in place, generated samples tightly match target distributions for summary statistics (mass density histogram, power spectrum)



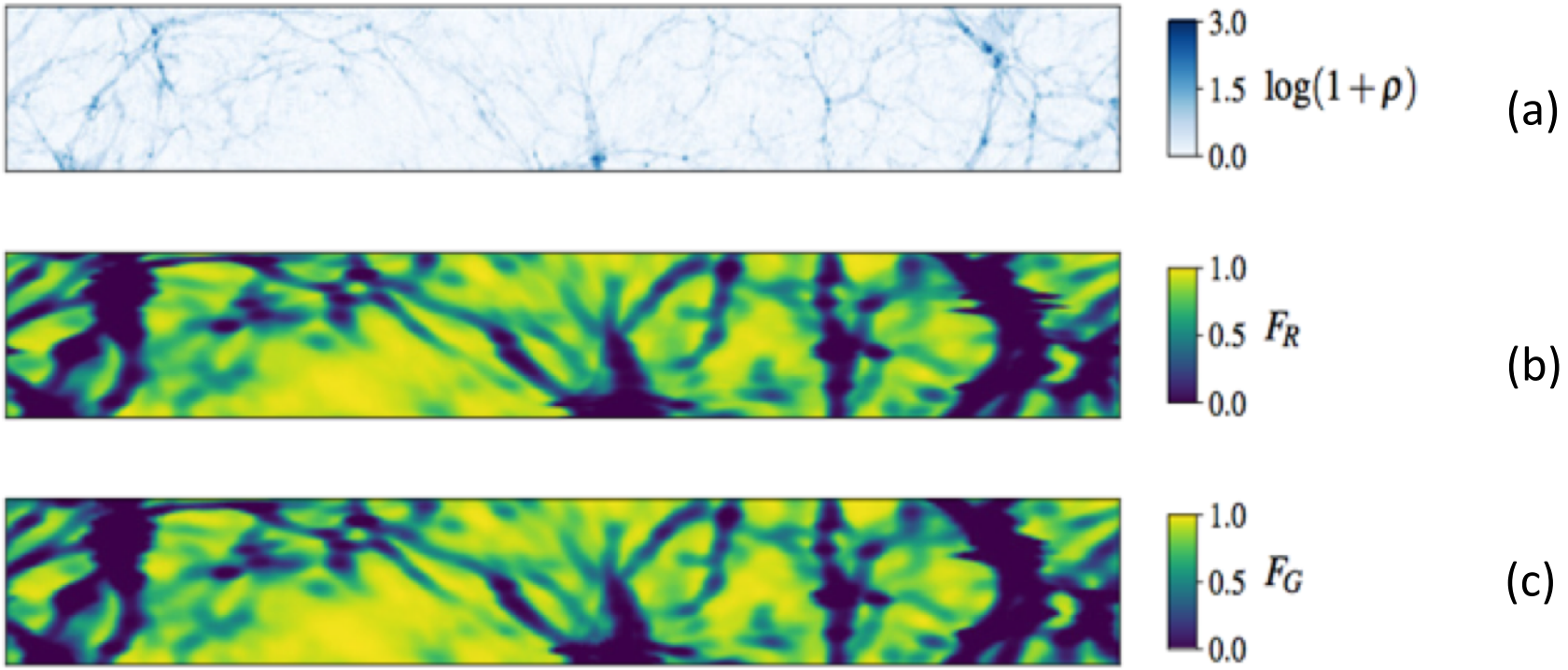




- Now, onto **scaling up** these techniques for larger sample sizes, in 3D, etc

LBANN to the rescue!

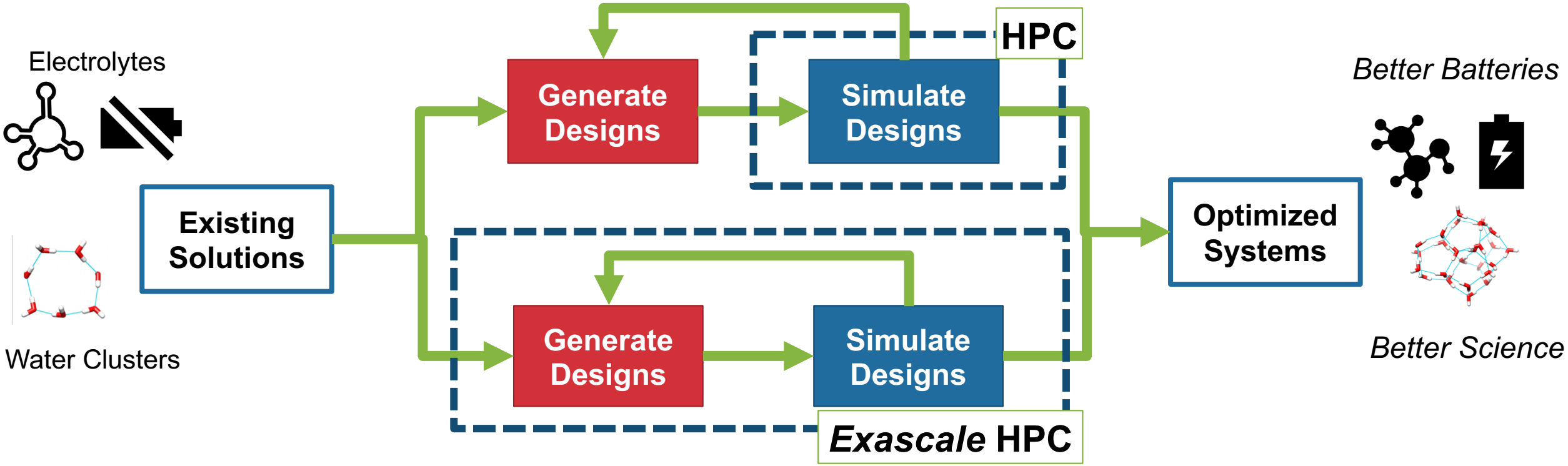
VAE for simple to full-physics models to observations



Visualization of the pipeline output. A 3D dark matter distribution (of which a 2D slice is shown in panel (a)) is the principal input to the workflow, which tries to produce the corresponding Ly-alpha flux field FR (b). The prediction FG is shown in panel (c). Generally, structures at both large and small scales, as well as the distortions that warp them in redshift space, are captured well.

Nyx with only gravity and particle and Nyx with full hydro and gas physics.

Design: Expanding Computational Design to the Exascale

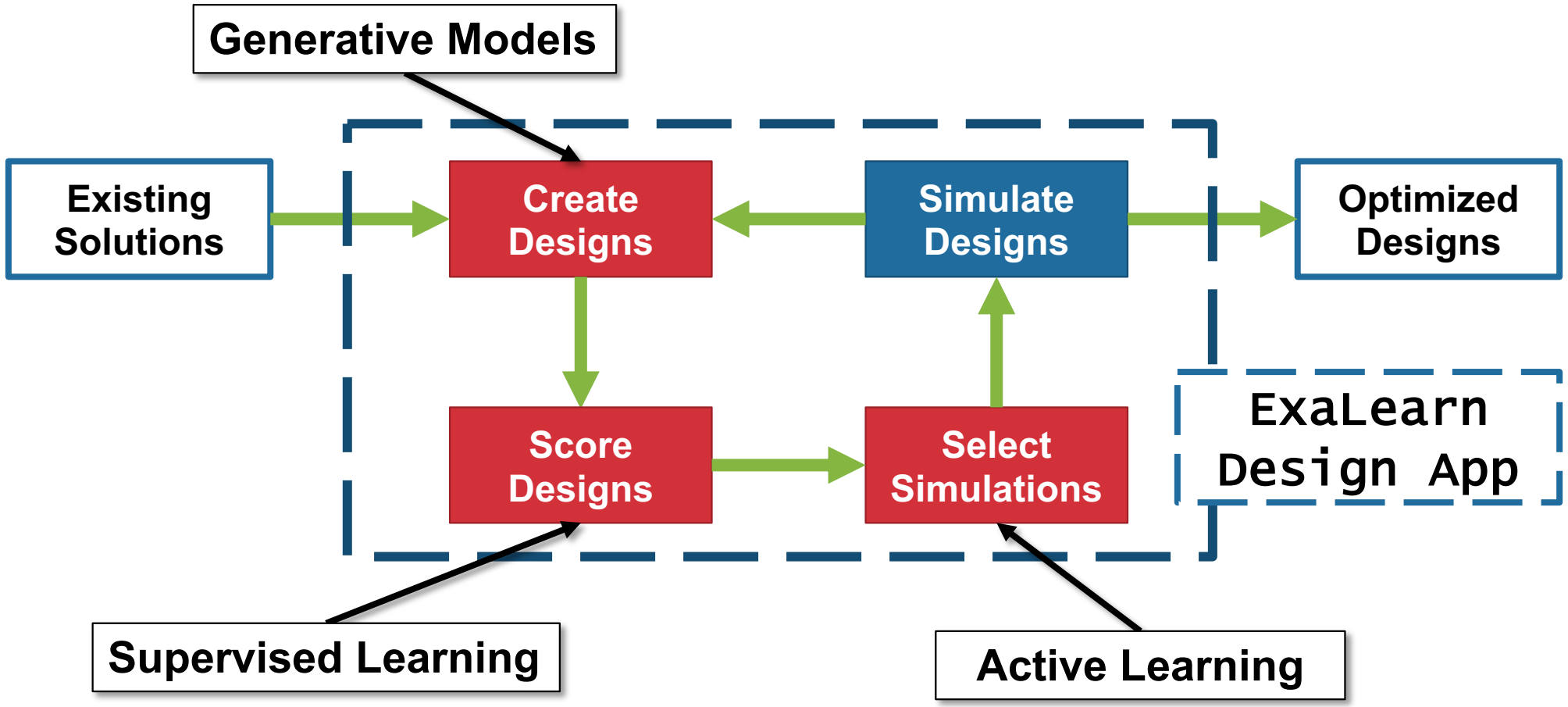


Current Model: Humans steer HPC, HPC performs simulations *(Months-Years)*

Why is this limited? Humans are **slow.** Slow decisions, slow to learn

Needed Solution: **HPC steering itself** **(Days-Weeks)!**

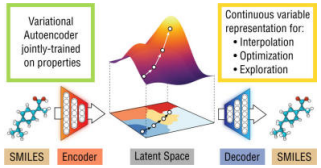
Steering HPC Requires Extensive Machine Learning



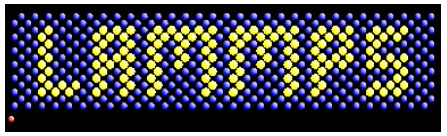
- Our goal is to provide a generalized framework for common data abstractions.
- For example, data in multiple chemistry applications is modeled as graphs.
- Our API will require users to focus on implementing a few functions.

Example Workflow: Molecular Design

Graph/Molecular generative models (Autoencoders, GANs, RL algorithms)



NWCHEM
HIGH-PERFORMANCE COMPUTATIONAL CHEMISTRY SOFTWARE



Atomistic Calculations: Quantum or Classical

Existing Solutions

10^3 mols

Create Designs [10^1 node-hr]

10^3 calcs

Simulate Designs [10^4 node-hr]

Optimized Designs

10^5 mols

10^3 cycles

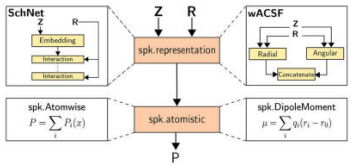
10^3 mols
1 calc/mol

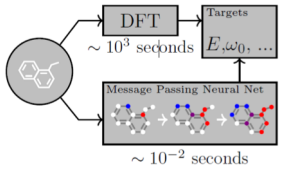
Score Designs [10^2 node-hr]

10^5 mols

Select and Explanation of Top Designs [10^2 node-hr]

Gym





Ensemble of Deep Neural Networks
- **Continuous-filter CNNs**
- **Message-Passing Networks**
- **[…]**

Active Learning policies driven by graph descriptors, tuned with reinforcement learning

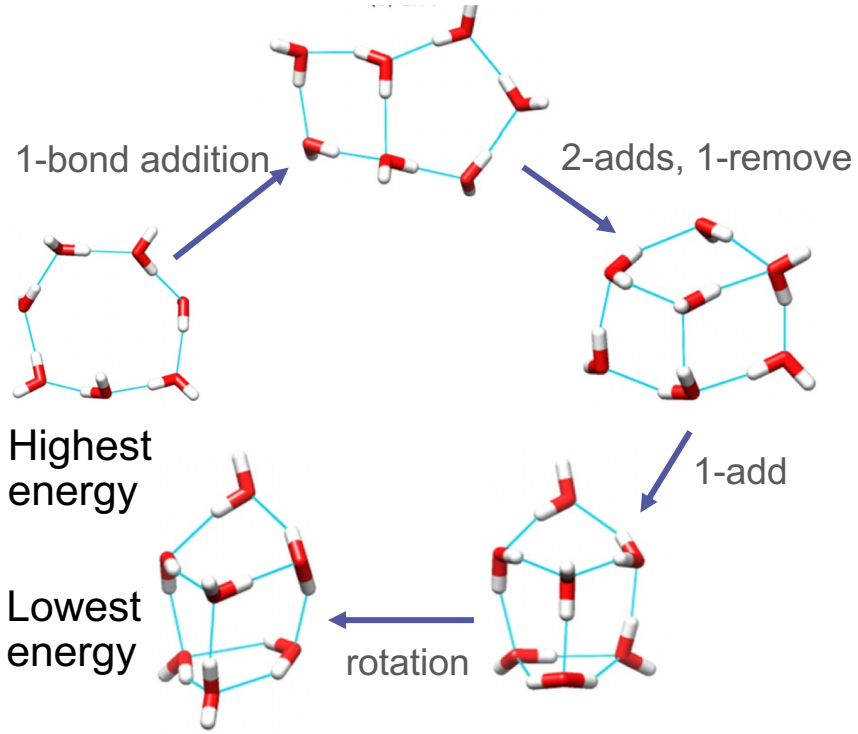
ECP EXASCALE COMPUTING PROJECT

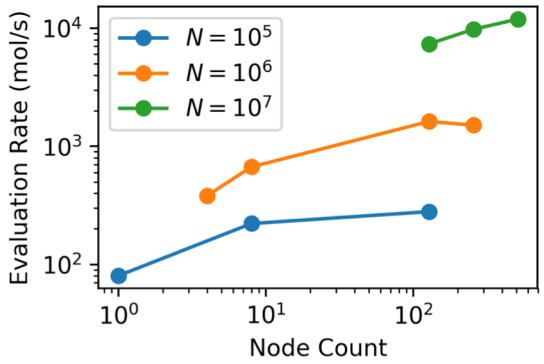
FY19 Results: Building Problem-Specific Components

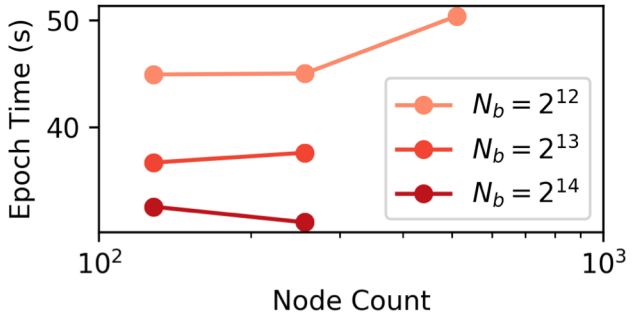
RL-based implementation of structure design

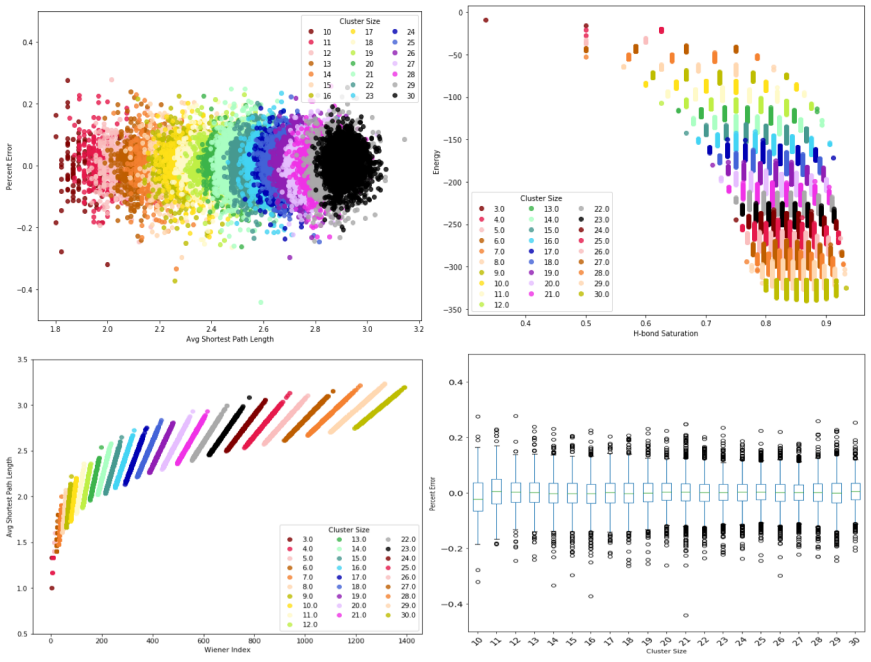
Scalability studies with Graph-CNNs (on Theta and Summit)

General-purpose descriptors for interpretable models





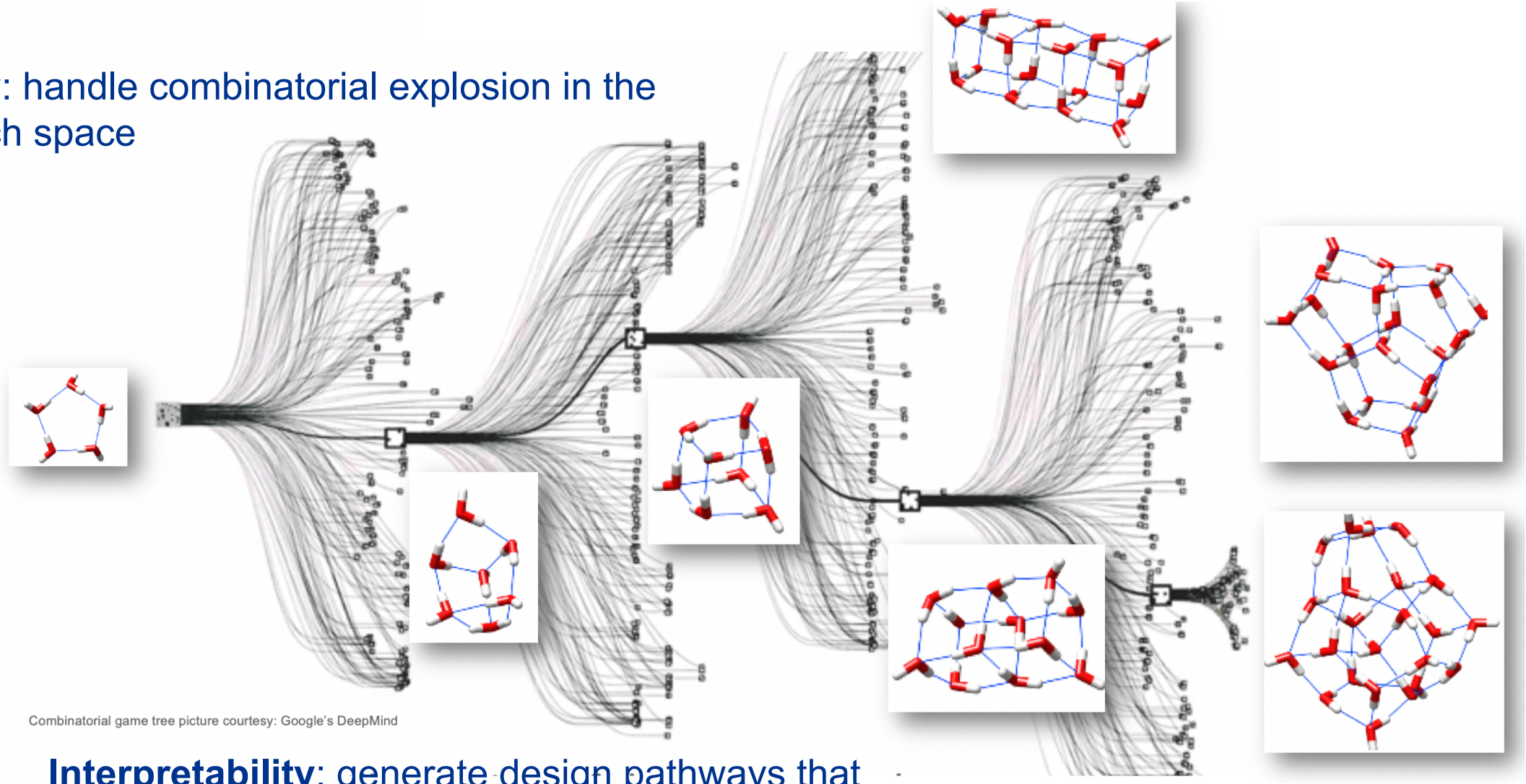




1. Develop Reinforcement Learning pipeline for graph-based networks (with ExaLearn control)
2. Tailor Reinforcement Learning algorithms with physics-aware ML algorithms
3. Develop interpretable ML models for graph-based models of atomic/molecular structure
4. Generate novel electrolyte molecules and water clusters

Key Goals: Scalability and Interpretability

Scalability: handle combinatorial explosion in the state-search space



Combinatorial game tree picture courtesy: Google's DeepMind

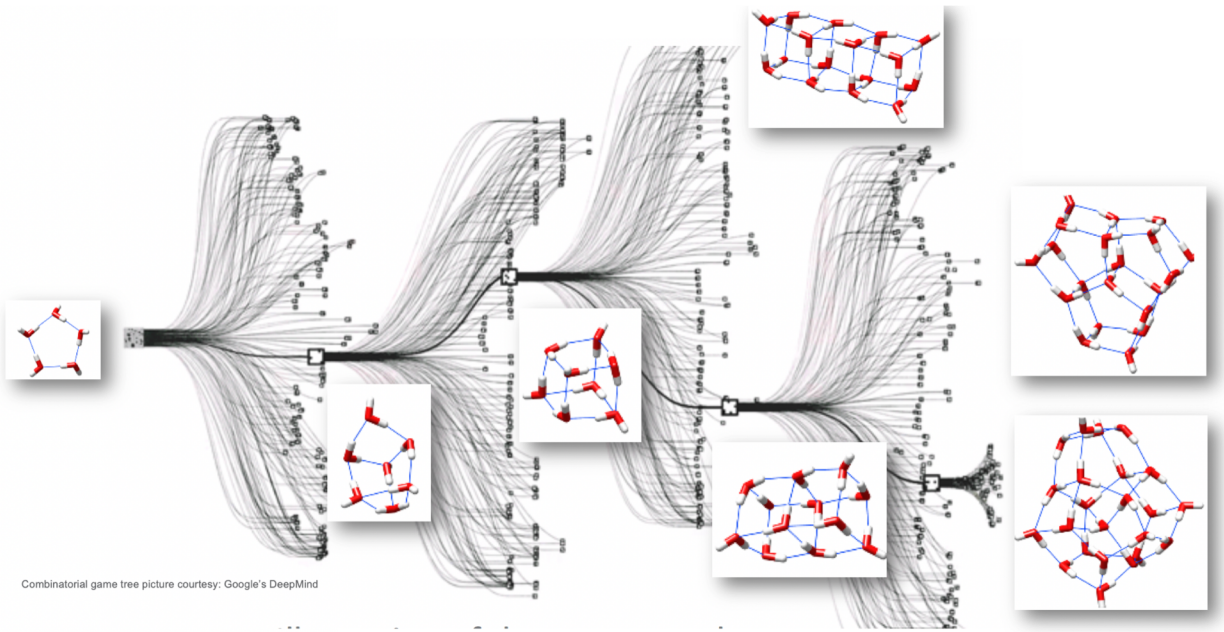
Interpretability: generate design pathways that scientists can reason about

How will we find our Target Structures?

Algorithm 1 A search algorithm to find a target molecular structure starting from an initial structure G_{mol}^{init} using a reinforcement learning model M_{RL} and branching factor b.

1: **procedure** SEARCH($G_{mol}^{init}, b, M_{RL}$)
2: INIT($results, \emptyset$)
3: INIT($\mathbb{P}_{cand}$, EXPLORE-NEXT($M_{RL}, [G_{mol}^{init}], b$)))
4: **while** $size(\mathbb{P}_{cand}) > 0$ **do**
5: $pathway$ = POP($\mathbb{P}_{cand}$)
6: N_{cand} = EXPLORE-NEXT($M_{RL}, pathway, b$)
7: **for all** $G_{new} \in N_{cand}$ **do**
8: $P_{new} = pathway \cup \{G_{new}\}$
9: **if** MATCH-TARGET(G_{new}) **then**
10: $results = results \cup \{G_{new}\}$
11: **else**
12: PUSH($\mathbb{P}_{cand}, P_{new}$)
13: RETURN $results$
14: **end procedure**

Learning M_{RL} is the focus on this tutorial



Combinatorial game tree picture courtesy: Google's DeepMind

Machine Learning for Inverse Problems

Problem Definition

Use ML methods to solve the inverse problem of predicting material structures from experimental data
(diffraction patterns called Bragg profiles in this demo)

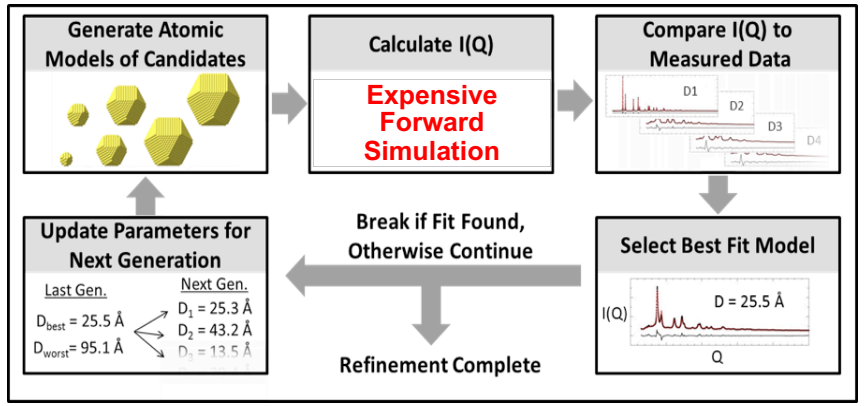
Main FY19 Outcomes

- Scalable distributed framework for generation of labelled simulated neutron diffraction data set implemented; Framework potentially generalizable to other simulators.
- Multiple labelled data sets of modest sizes generated; Initial design of a classifier for structural symmetry prediction tested.
- Multiple shallow regressor models for parameter predictions trained; Using best model, about 90% prediction accuracy demonstrated.
- Together, these constitute one of the first evidences of the feasibility of material structure prediction using neutron scattering data.
- All efforts in the inverse problem application built ground up from scratch; Paper reporting this effort to be submitted in October, 2019

Team

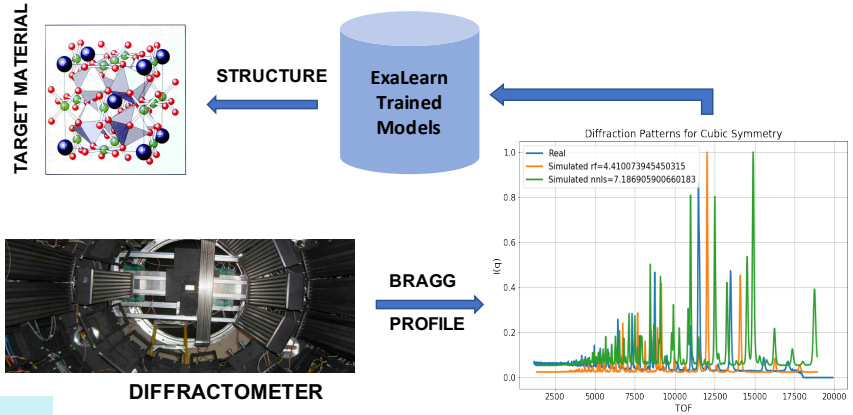
Cristina Garcia Cardona (LANL), Ramakrishnan Kannan (ORNL), Thomas Proffen (ORNL), Travis Johnston (ORNL), Katherine Page (ORNL/UTK), David Womble (ORNL), Sudip K Seal (ORNL, **POC**)

Current Loop Refinement Method



Inferences are time-consuming, less accurate and model-driven

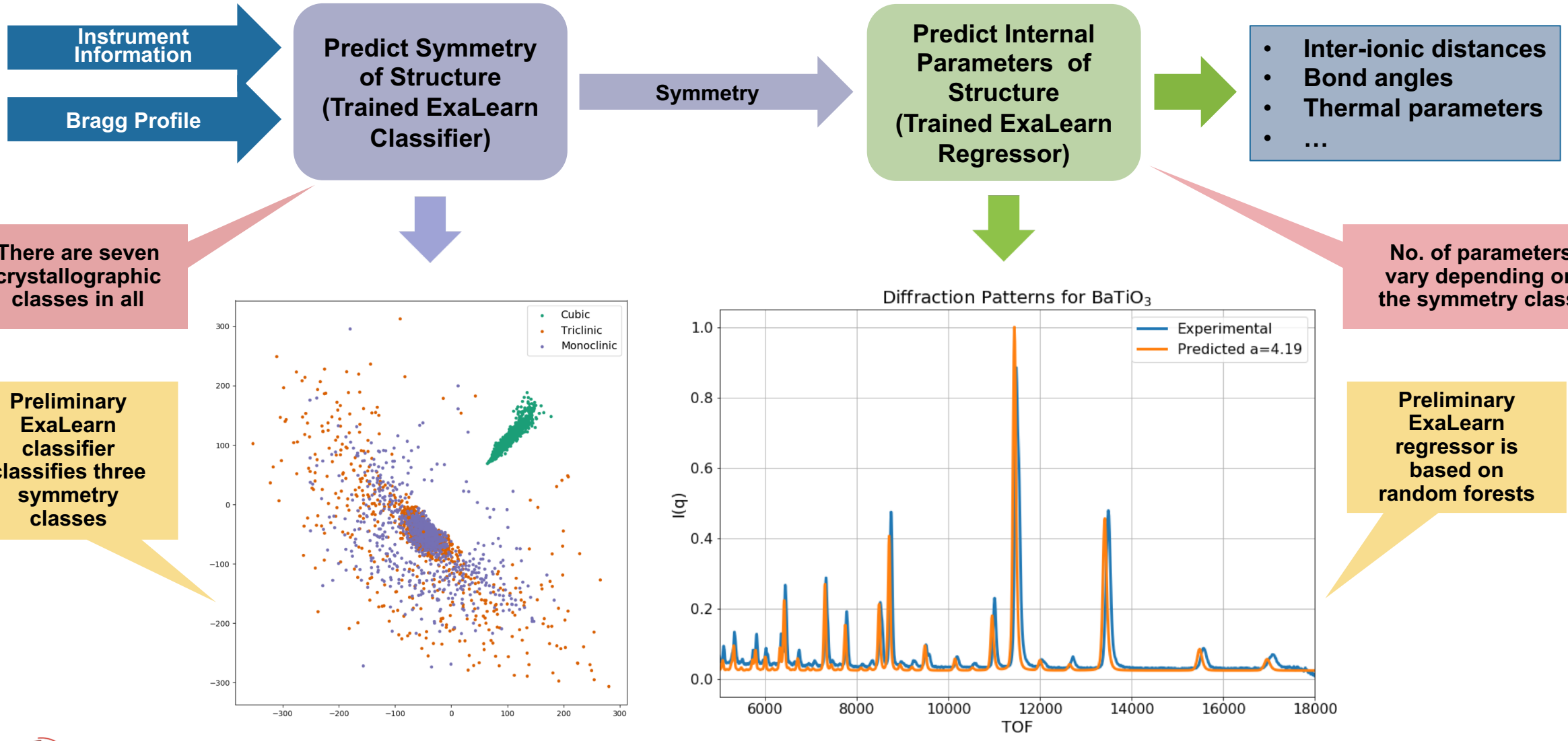
Proposed ExaLearn Method



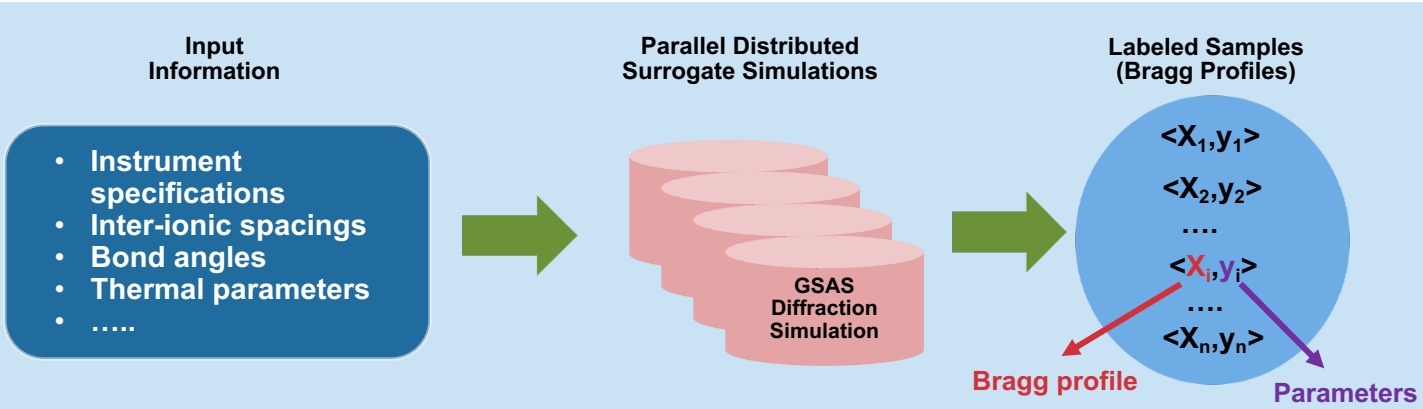
Once trained, predictions are fast, more accurate and data-driven

Inverse Problems in Materials Science : ML Pipeline

ExaLearn Pipeline for Material Structure Determination from Neutron Scattering Data



Inverse Problems in Materials Science : Status

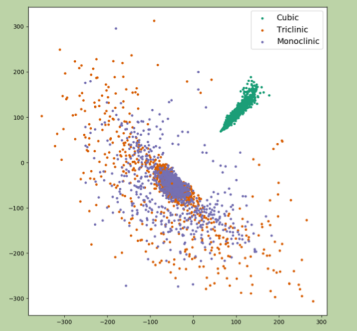


Training Data Generation

Class Label	Data Size	Time Taken (in hr)	No. of Cores
Cubic	125,000	120	2
Monoclinic	297,910	32	24
Triclinic	804,357	10	48
Tetragonal	998,584	96	48

Generation of Large-scale Training Data

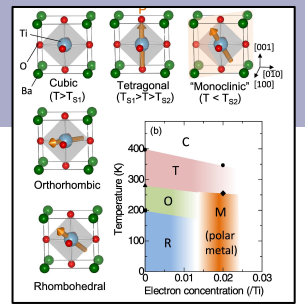
Initial Design of Classifier



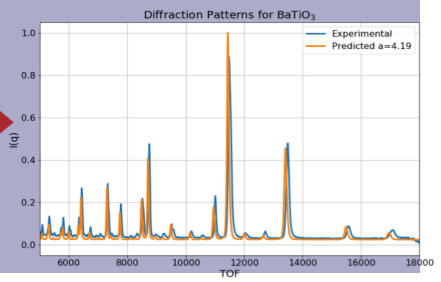
- 1D-Convolution (16 kernels, width 3)
- 1D-Max Pooling (kernel width 2)
- 1D-Convolution (32 kernels, width 4)
- 1D-Max Pooling (kernel width 2)
- Fully Connected Layer (256 hidden neurons)
- ReLU (non-linear activation)
- Fully Connected Layer (3 output neurons, one for each class)
- Softmax (probability distribution over the 3 classes).

Initial Evaluation of Regressors

- Unconstrained Least Squares
- Non-Negative Least Squares
- Multi-Label Regressor with Gradient Boosted Trees
- Support Vector Machine – Regression
- **Random Forests**



Best performance (about 90% accuracy with cubic symmetry)

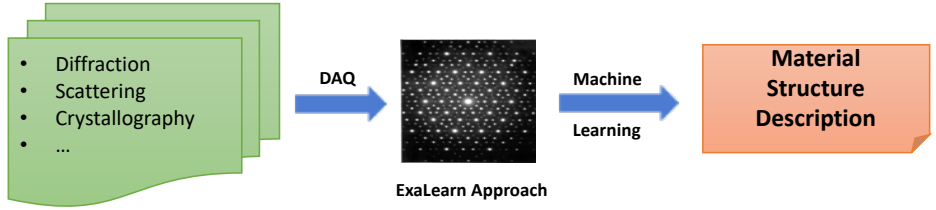


Models were tested against annotated experimental data from the NOMAD diffractometer in SNS

Labelled data generation for the full problem space and training deeper networks will require **extreme scale** compute time and resources – next steps!

Extreme-scale Machine Learning for Inverse Problems
Background

- **Long term goal:** develop and deploy ML-driven solutions of large-scale inverse problems that are directly relevant to DOE-related science and technology
- Given a set of observations, **inverse problems** seek to determine the parameters that produced those observations.
- Inverse problems arise in numerous DOE-related scientific application domains, e.g.,
  - Fusion physics: given plasma equilibrium profiles in tokamaks/stellarators, determine device diagnostics.
  - Microscopy: various kinds of microscopy -- electron, scanning tunneling, transmission electron and others; given a microscopy image, determine the material properties that produced the observed image.
  - X-Ray crystallography: determine structure of target from diffraction patterns produced by it upon bombardment by incident X-ray beam.
  - Additive manufacturing: determining thermal parameters from target solidification microstructures in powder-bed metal additive manufacturing.
- **Short term goal:** Develop extreme-scale ML framework to solve the inverse problem of material structure determination from neutron scattering experiments.



Predicting Material Structure from Neutron Scattering Data

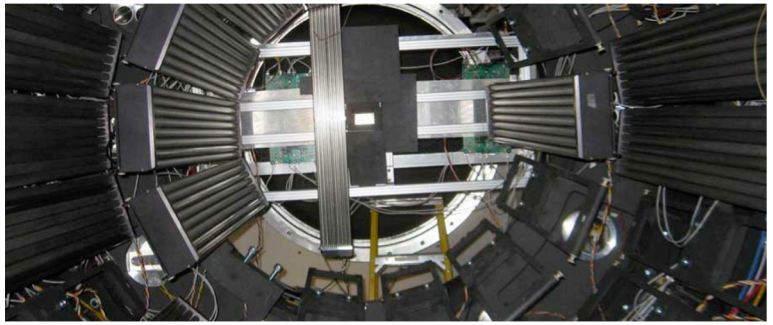
Problem Definition

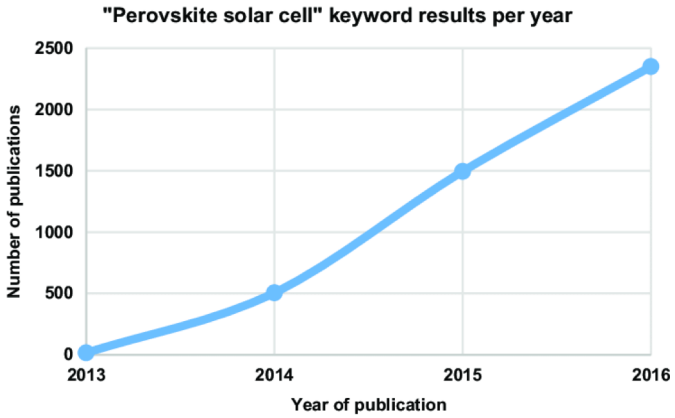
Use ML methods to solve the inverse problem of predicting material structures from neutron diffraction patterns called Bragg profiles.

Problem Scope

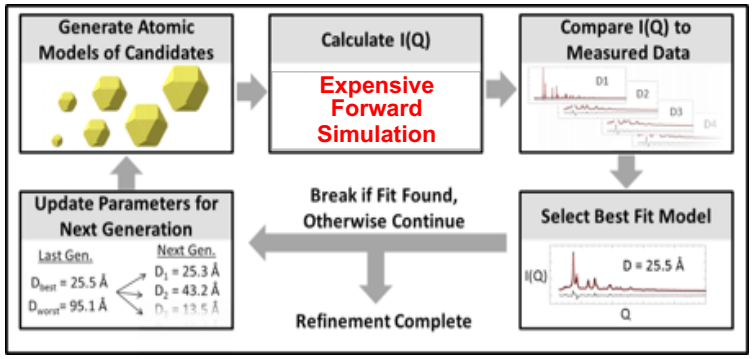
- The problem lends itself as a rich template that can be generalized to other problem domains.
- Access to experimental data (NOMAD) for model validation studies.
- Initial target material is a perovskite called barium titanate ($BaTiO_3$).
- Effort expected to be immediately impactful to a very large international user base at the Spallation Neutron Source (SNS).

Nanoscale-Ordered Materials Diffractometer (NOMAD)



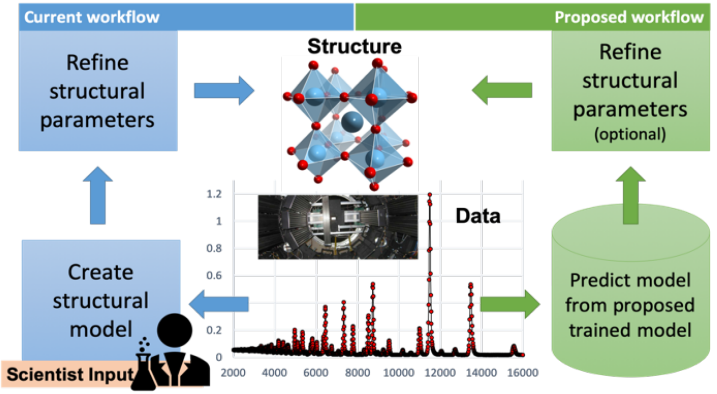


Current Loop Refinement Method



Inferences are time-consuming, less accurate and model-driven

Proposed ExaLearn Method



Once trained, predictions are fast, more accurate and data-driven

Perovskite solar cells: https://www.researchgate.net/figure/Number-of-publications-resulting-from-the-search-of-perovskite-solar-cell-on-Web-of_fig3_317569861

ExaLearn ML Pipeline

ExaLearn Pipeline for Material Structure Determination from Neutron Scattering Data

Instrument Information

Bragg Profile

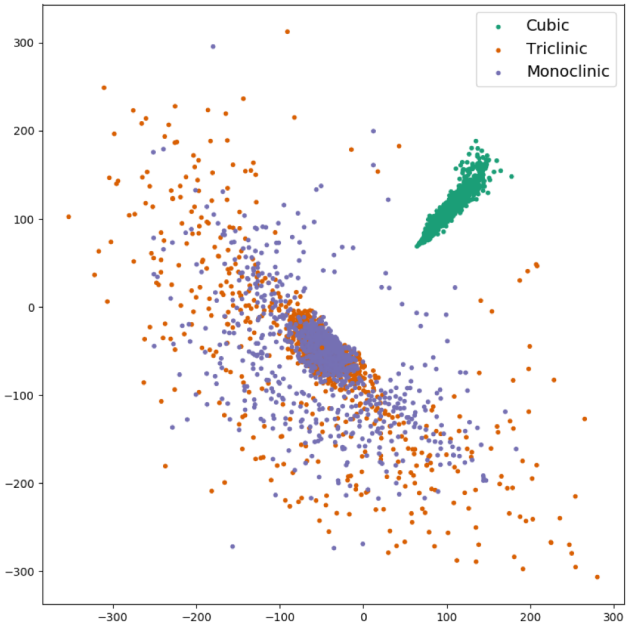
Predict Symmetry of Structure (Trained ExaLearn Classifier)

Symmetry

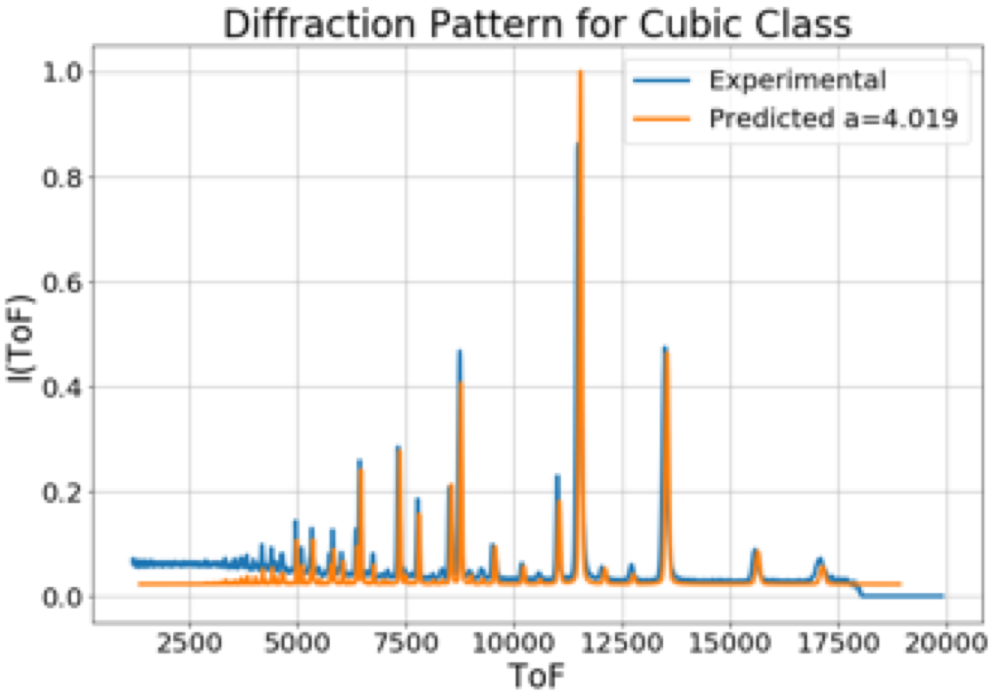
Predict Internal Parameters of Structure (Trained ExaLearn Regressor)

- Lattice lengths
- Lattice angles
- Thermal parameters
- …

There are seven crystallographic classes in all

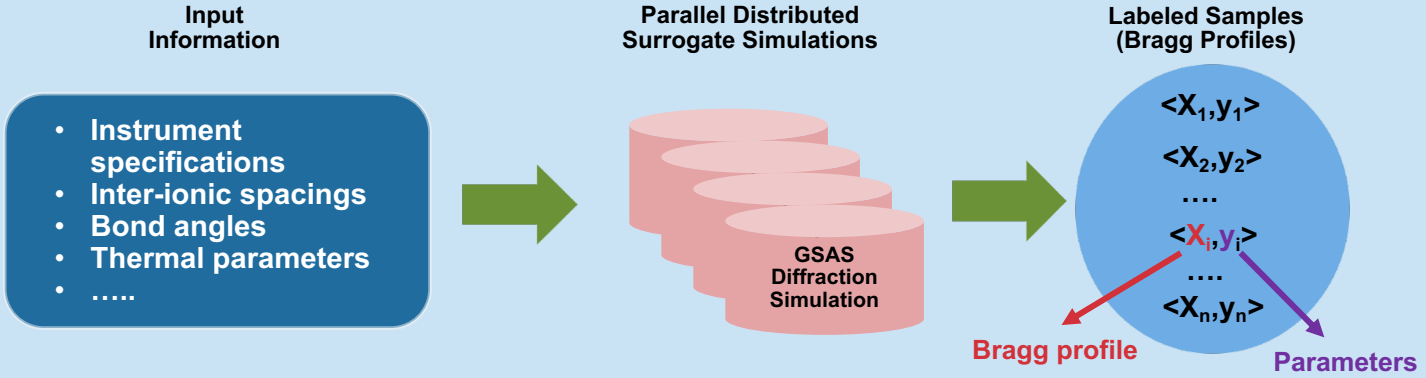


No. of parameters vary depending on the symmetry class.



Conclusions

Learning to Predict Material Structure from Neutron Scattering Data, *Workshop on Big Data, Tools and Methods (BTSD), IEEE Big Data 2019, Los Angeles, Dec 9-12, 2019.*



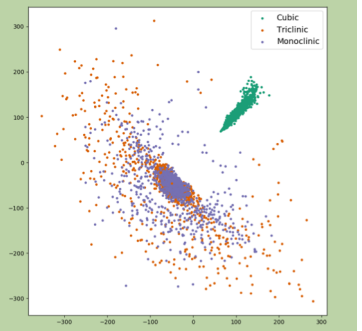
Generation of Large-scale Training Data

Training Data Generation

Class Label	Data Size	Time Taken (in hr)	No. of Cores
Cubic	125,000	120	2
Monoclinic	297,910	32	24
Triclinic	804,357	10	48
Tetragonal	998,584	96	48

Initial Design of Classifier

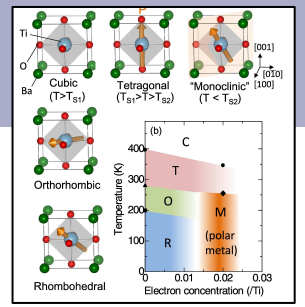
- 1D-Convolution (16 kernels, width 3)
- 1D-Max Pooling (kernel width 2)
- 1D-Convolution (32 kernels, width 4)
- 1D-Max Pooling (kernel width 2)
- Fully Connected Layer (256 hidden neurons)
- ReLU (non-linear activation)
- Fully Connected Layer (3 output neurons, one for each class)
- Softmax (probability distribution over the 3 classes).

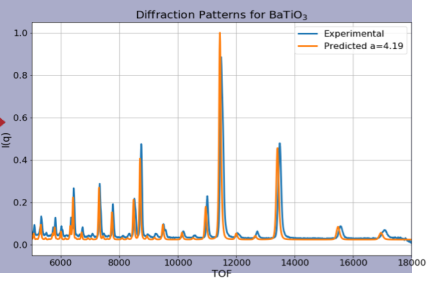


Initial Evaluation of Regressors

- Unconstrained Least Squares
- Non-Negative Least Squares
- Multi-Label Regressor with Gradient Boosted Trees
- Support Vector Machine – Regression
- **Random Forests**

Best performance (about 90% accuracy with cubic symmetry)



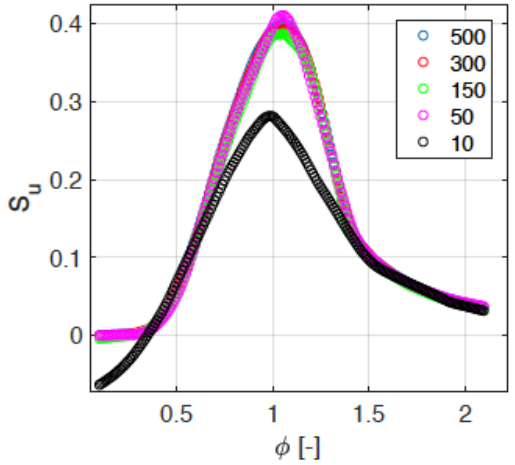


Models were tested against annotated experimental data from the NOMAD diffractometer in SNS

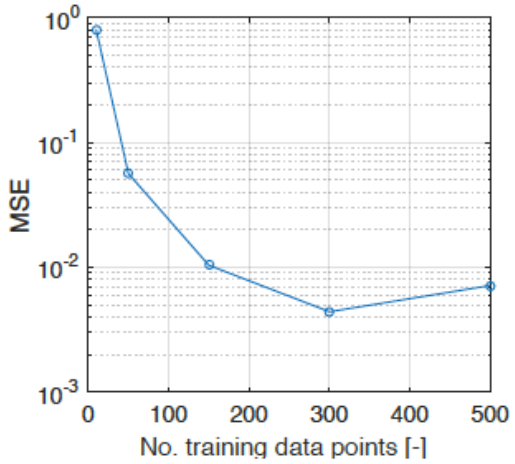
Labelled data generation for the full problem space and training deeper networks will require **extreme scale** compute time and resources – next steps!

UQ: A Critical Component of All Scientific Machine Learning Efforts

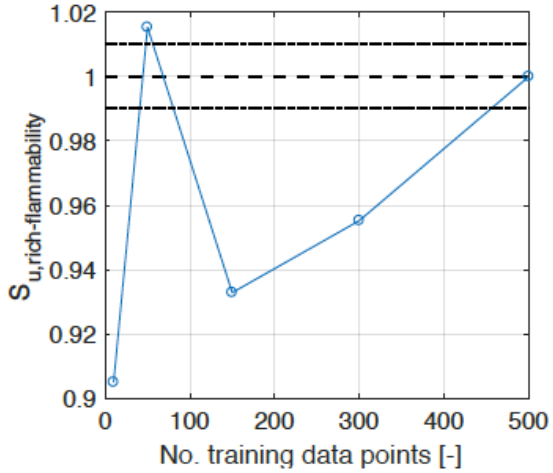
ExaLearn will work in collaboration with domain scientists to develop appropriate Uncertainty Quantification tools for the individual application pillars.



combustion NN surrogate for laminar flame speed



generalization error for increasing data volumes



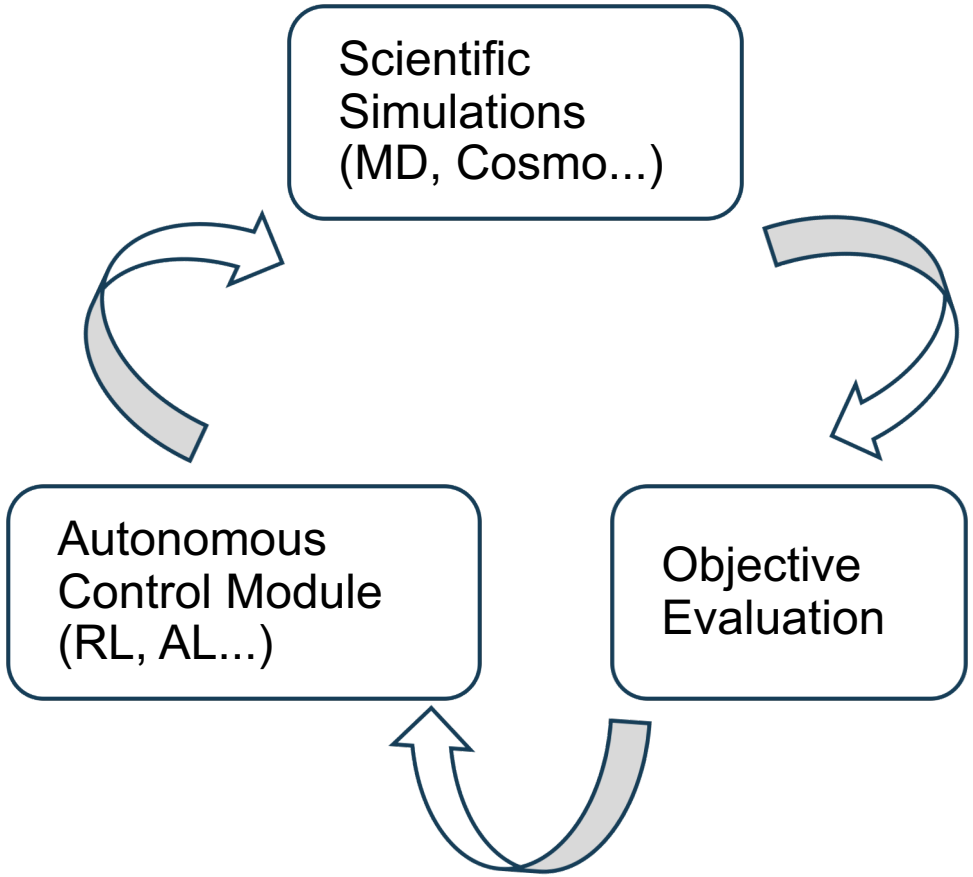
relative error on flame speed summary statistic for increasing data volume

- Analyzed surrogate accuracy for increasing training data volumes, assessing generalization error and error in specific summary statistics of interest
  - Preliminary results on combustion flame speed computations
- Planned: A-posteriori analysis of trained machine learning models, sensitivity analysis, robustness to data noise, NN architecture changes

Vision: Enabling probabilistic neural network training, with mean and variance estimates on network parameters, to assess predictive fidelity of surrogate models

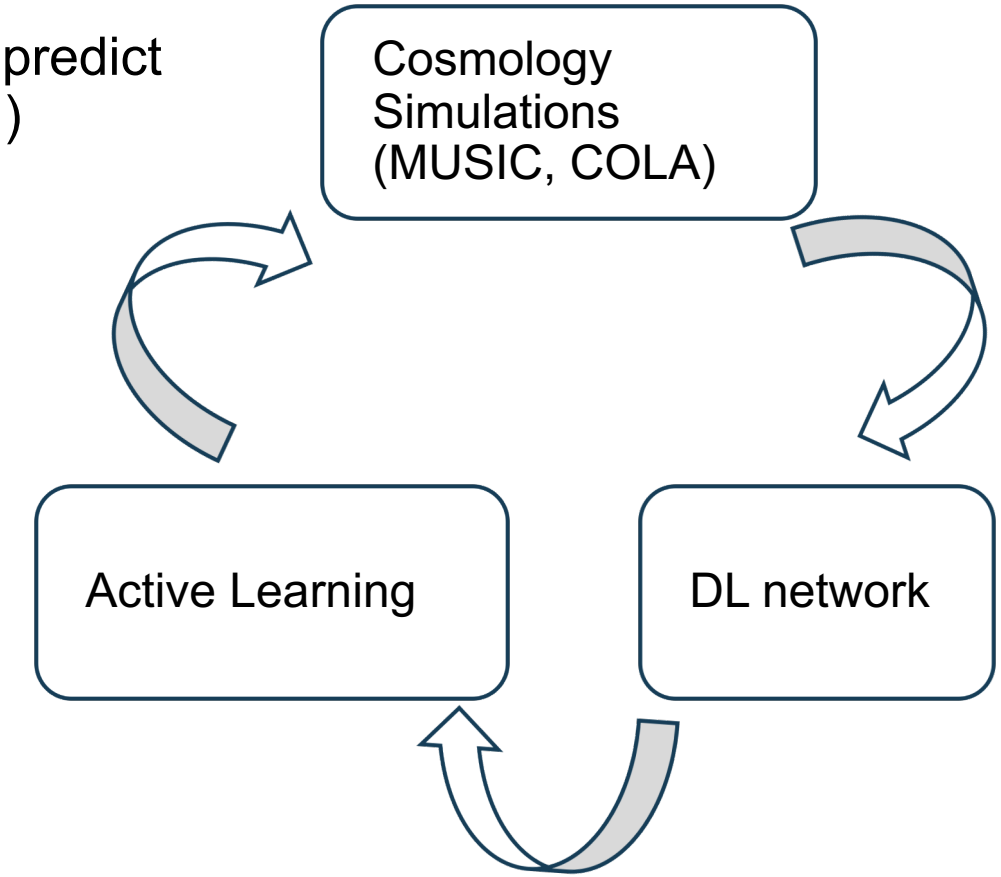
Objective-Driven Experimental Design

- **Motivation**: At large scales, simulation “efficiency” runs risk of being lower without careful steering. Or “wrong” “suboptimal” calculations may be performed without careful analysis of results generated.
- **Definition**: ODED is an ML-enabled autonomous system to design and execute computations.
- **End Results**:
  - To reach the objective earlier (with less computation)
  - To produce a better result (within the computation budget)
- **How It works**:
  - Iterative learning, simulation, and objective evaluation
  - Autonomous Control Module could be Reinforcement Learning, Active Learning, or other types of modules.

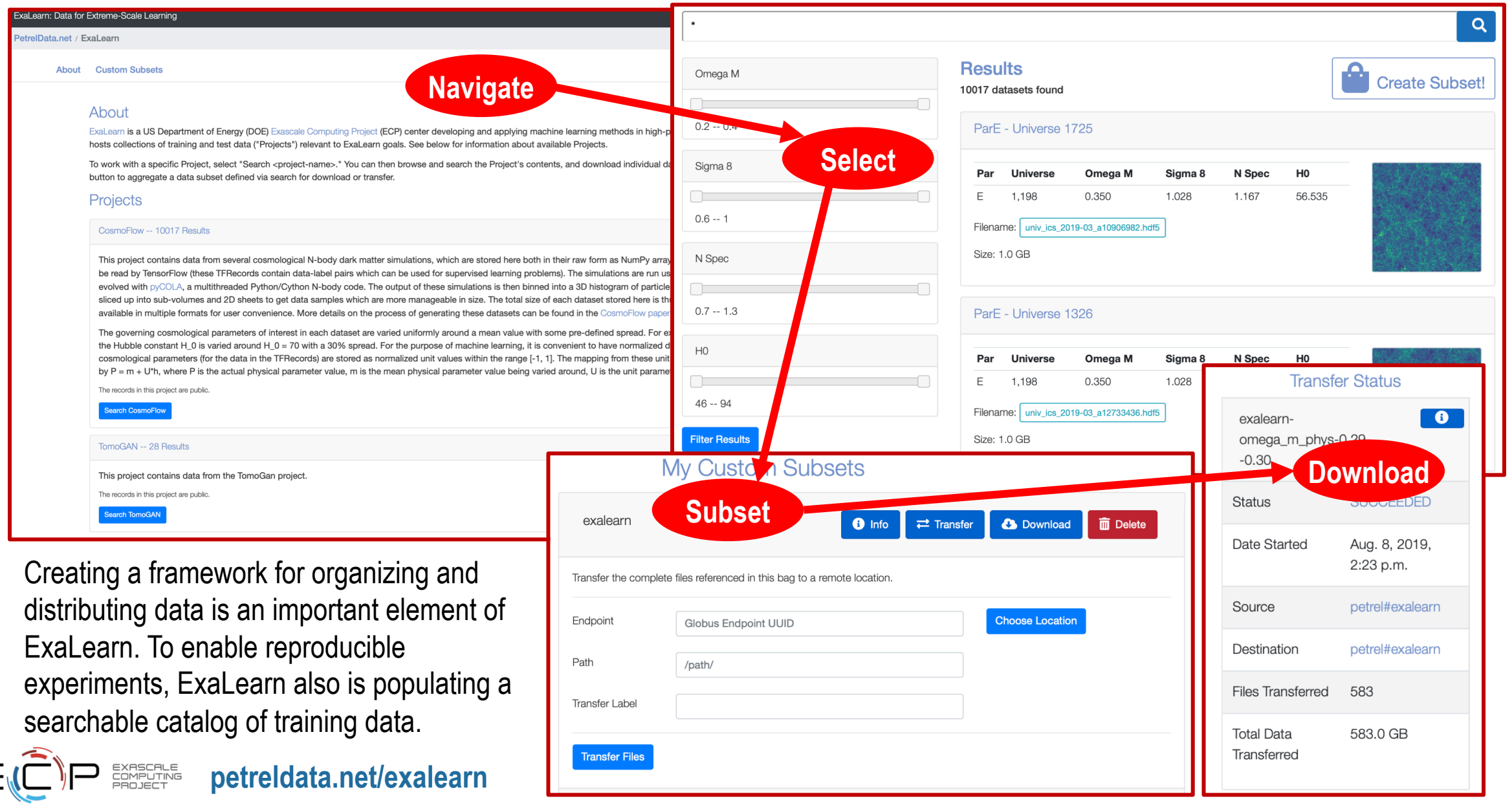


ODED Example: A Simplified Cosmology Problem–Parameter Estimation (Classification)

- Given a 3D sub-volume of the universe (from simulation), predict the parameter \omega-m \in [0.15, 0.40] (6 classes in total)
- **Objective: Train an estimator with as few simulation runs as possible**
- CosmoFlow CNN model is used.
- Current Stage:
  - Using PyCOLA as the simulation software
  - One-degree of freedom to control the simulation (\omega_0)
  - On local 8-GPU compute node
- Future:
  - Use a more expensive N-body simulation (e.g., GADGET)
  - Expand to more degrees of parameters on leadership machines
  - Provide a library to generalize the capability to other applications



Searchable Repository for Machine Learning Training Data



Creating a framework for organizing and distributing data is an important element of ExaLearn. To enable reproducible experiments, ExaLearn also is populating a searchable catalog of training data.

ECP EXASCALE COMPUTING PROJECT

petreldata.net/exalearn

Online Access to ExaLearn Models

DLHub

Data and Learning Hub for Science

A simple way to find, share, publish, and run machine learning models and discover training data for science

Documentation

Read the Docs | Python SDK | CLI

Papers and Presentations

DLHub on ArXiV | DLHub Slides

Get Started

1 **Describe**

```
m = KerasModel()
m.create_model("p1b1-example.h5")

m.set_title("CANDLE Pilot 1 - Benchmark 1")
m.set_name("candle_p1b1") # short name
m.set_domains("genomics","biology","HPC")
```

2 **Publish**

```
from dlhub_sdk.client import DLHubClient

dl = DLHubClient()
dl.publish_servable(m)
```

3 **Run**

```
from dlhub_sdk.client import DLHubClient

dl = DLHubClient()

mid = dl.get_id_by_name("candle_p1b1")
data = np.load("pilot1.npy")
pred = dl.run(mid, data.tolist())
```

dlhub.org

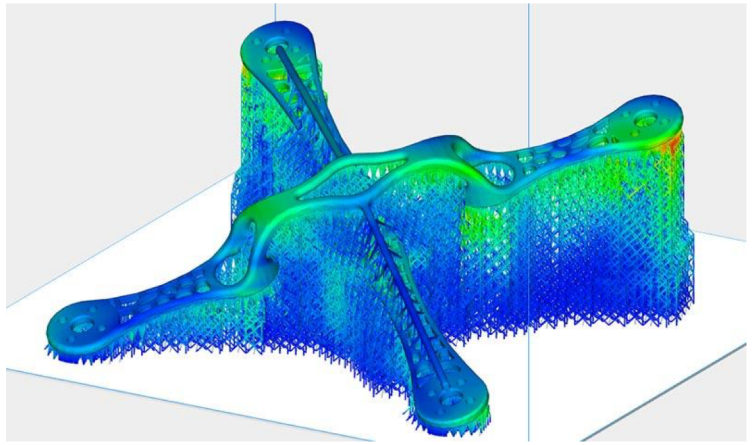
Brief Overview of ExaLearn Control Pillar

- Goals
  - Provide scalable control-related machine learning software for ECP applications
  - Implement use case applications for demonstration and testing
  - Run on exascale DOE Leadership class computing facilities
- Methods
  - Using primarily reinforcement learning for now, but could expand to other methods
  - Science use case: RL for temperature control for block copolymer self-annealing in light source experiments
  - EXARL software framework for exascale reinforcement learning for science and benchmarking
- Collaboration
  - Working toward adoption of ECP ExaAM (additive manufacturing) application
  - Leverage related ECP application software (eg CANDLE hyperparameter optimization)
  - ECP Proxy App project collaboration on RL proxy app

Control Problems in Science

Simulation

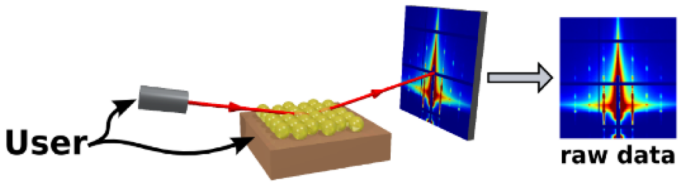
Accelerate sampling in a simulation via search, to reduce computation required for solution

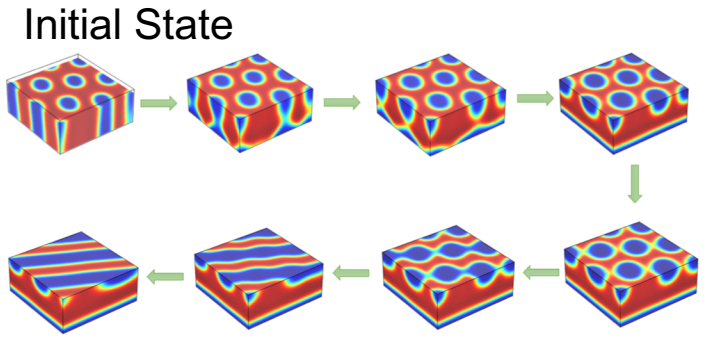


https://www.materialise.com/en/press-releases/materialise-brings-simulation-for-additive-manufacturing-to-production-floor

Experiment

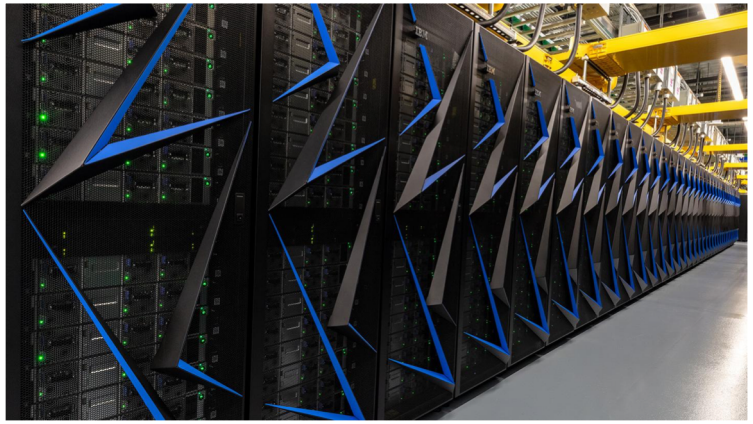
Guide scientific experiments - eg. block copolymer self-annealing





Operation

- Control HPC facility resource management
- Control experimental facilities (eg x-ray beam)
- Control air, land or space vehicles



https://www.olcf.ornl.gov/summit

Control Problems: Do You Have One For Us to Scale?

Characteristics of control problems:

actions to take to get to different states, target state

Complex control problems may have:

- many different possible actions and/or states
- complicated trade-offs
- conditional behavior
- complex goal and subgoal relationships
- nuances in the order of actions taken
- long-term rewards that may not be immediately obvious

(Reference: http://opendeeptech.com/alphago-googles-artificial-intelligence)

Examples of everyday complex control problems:

- Game playing (Go, Atari)
- Autonomous vehicle control
- Robotic control
- Factory control

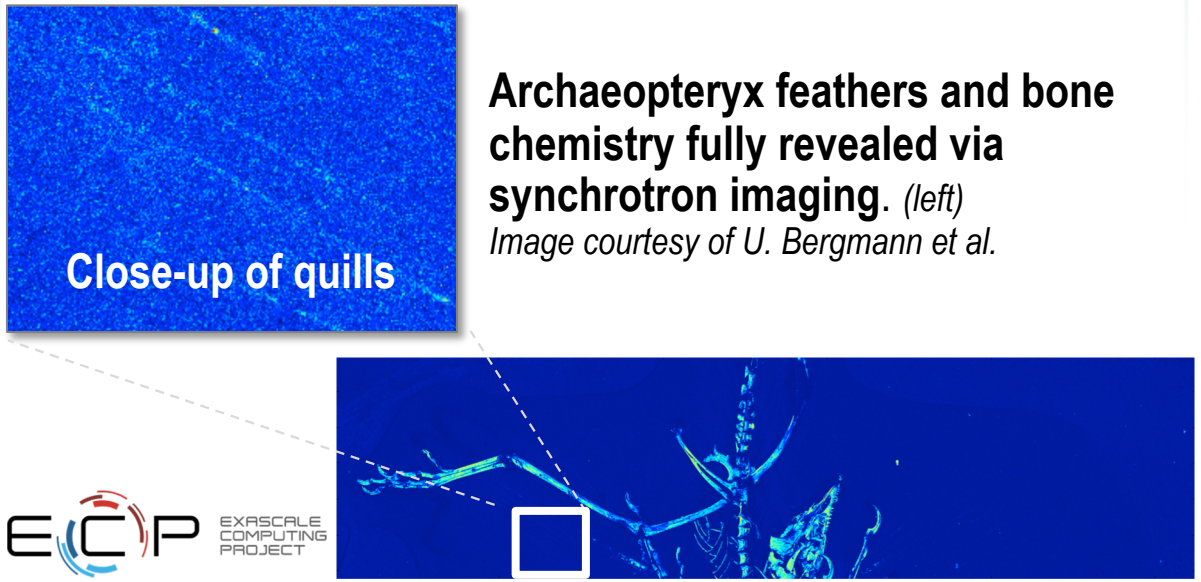
(Reference: ipam.ucla.edu/programs/workshops/autonomous-vehicles/)

(Reference: https://robohub.org/deep-learning-in-robotics/)

Upcoming Control Use Cases for FY20 and Beyond

Adaptive Illumination

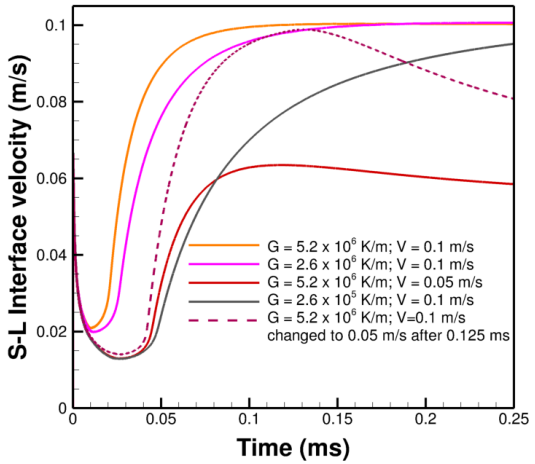
- **SLAC/Stanford use case**, training data and RL system (D. Ratner, J. Betterton, M. Kochendorfer)
- **AI use case**: exploit image sparseness to improve acquisition time/X-ray dose/resolution by adaptively controlling acquisition.
- **Impact:** Reduced imaging needed (shorter experiment time) at both synchrotron and XFEL light sources.
- **Example**:



Archaeopteryx feathers and bone chemistry fully revealed via synchrotron imaging. *(left) Image courtesy of U. Bergmann et al.*

Additive Manufacturing

- **ExaAM use case**, training data available (MEUMAPPS sim)
- **AI use case**: Predict next steady-state behavior in the solidification of a metal to guide real-time annealing in line with the experiment.
- **Impact**:
  - Reduce expensive simulations because can predict the next steady state response for a thermal stimulus.
  - Model system more dynamically to achieve desired end state.
- **Example**:



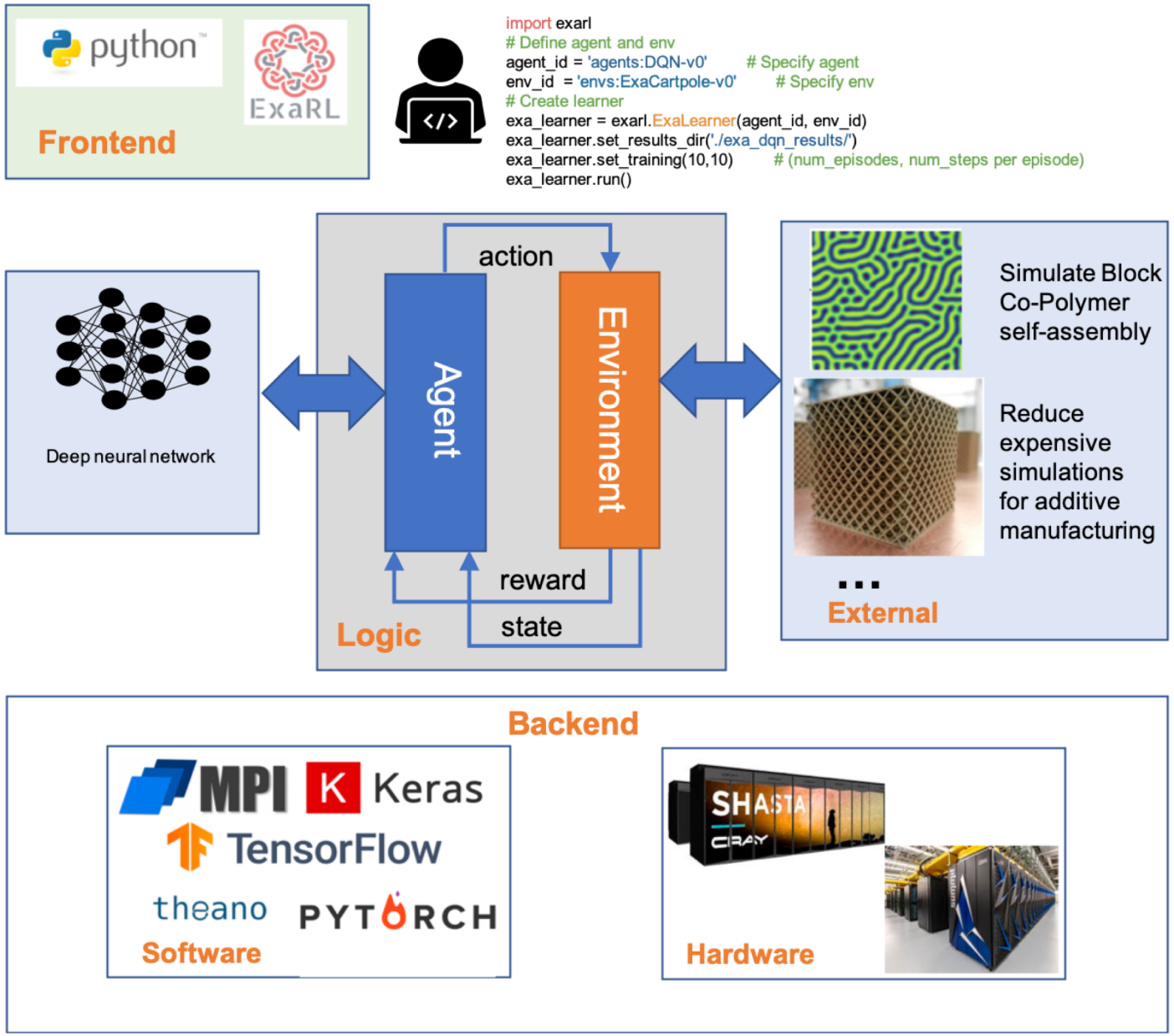
Simulation of directional solidification of a ternary Ni-Fe-Nb alloy. (left) The metal is in an unsteady state (dotted line) when transient behavior (slowed cooling rate) is introduced after the initial steady state was achieved.

Image courtesy of R. Balasubramaniam

LA-UR-19-27506

Easily eXtendable Architecture for Reinforcement Learning (EXARL)

- EXARL: scalable RL framework for scientific environments
- Extends OpenAI Gym's environment registry to agents
- Dynamic multi-node environments
- Abstract classes to mandate necessary functionality
- Easy to register new agents and environments
- Supports different hardware and software infrastructures
  - Use existing prevalent infrastructure



Future Vision: Integrate the Four Machine Learning Types in ExaLearn for a Single Application

Example: Tokamak Plasma Fusion

- Generate goal-driven surrogate models for dynamic processes to replace expensive whole device simulations (WDM)
- Use these surrogates to generate training data for a RL-based real-time controller
- Apply pipeline from the inverse problems pillar to predictions plasma equilibria configurations in tokamaks and stellarators
- Apply tools from the design pillar to optimize tokamak design and control policy

- EXAWIND: Exascale Predictive Wind Plant Flow Physics Modeling
- Combustion-Pele: Transforming Combustion Science and Technology with Exascale Simulations
- ExaSMR: Coupled Monte Carlo Neutronics and Fluid Flow Simulation of Small Modular Reactors
- MFIX-Exa: Performance Prediction of Multiphase Energy Conversion Device
- WDMApp: High-Fidelity Whole Device Modeling of Magnetically Confined Fusion Plasmas
- WarpX: Exascale Modeling of Advanced Particle Accelerators

Big Data Need Big Theory, Too

- **Coveney, Dougherty, Highfield**: “We point out the weaknesses of pure big data approaches…. No matter their ‘depth’ and the sophistication of data-driven methods, such as artificial neural nets, in the end they merely fit curves to existing data. Not only do these methods invariably require far larger quantities of data than anticipated by big data aficionados in order to produce statistically reliable results, but they can also fail in circumstances beyond the range of the data used to train them because they are not designed to model the structural characteristics of the underlying system. ***We argue that it is vital to use theory as a guide to experimental design for maximal efficiency of data collection and to produce reliable predictive models…***.”

Thank You