SAND2019-5645PE

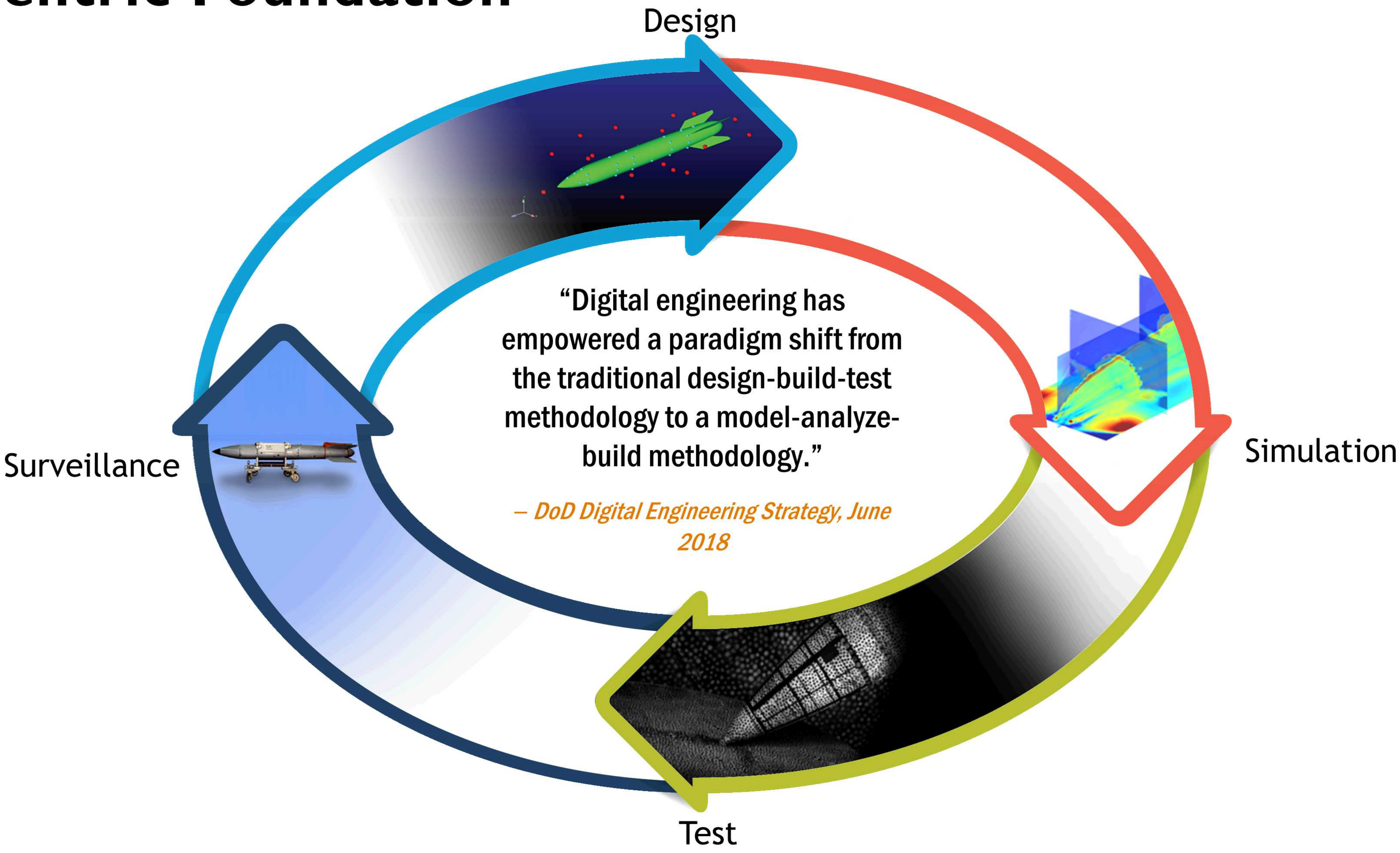
Exceptional service in the national interest

An Overview of Machine Learning for Scientific and High Performance Computing at Sandia

John Feddema, Sr. Manager, Enhanced Decision Making Group
Sandia National Laboratories
jtfedde@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

A Digital Engineering Strategy requires a Data-Centric Foundation



A Data-Centric Foundation will Enable Transformative Goals

Reduce the design cycle time: 10-12 years to 3 years	Increased production throughput	Re-think the surveillance program for the 21st century: improved assessment at lower cost
DATA STREAM	DATA STREAM	DATA STREAM
Agile, model-based design workflow Rapid cycles of learning	Credible mod/sim results Maintain International HPC/HPDA leadership	Risk-, physics-, and age-aware surveillance by design Synthesis of experimental and mod/sim results for critical decision making

National ASC Advanced Machine Learning (AML) Focus Areas

Improved efficiency in design process	Data-driven physics models	Enhanced experimental design
Anticipatory stockpile decision making	Reduced computational cost	

National ASC AML Priorities

Physics-constrained ML	Employ ML with sparse data	Invest in credible ML
Learning HPC hardware systems	Improve data specifications	Build talent at our labs

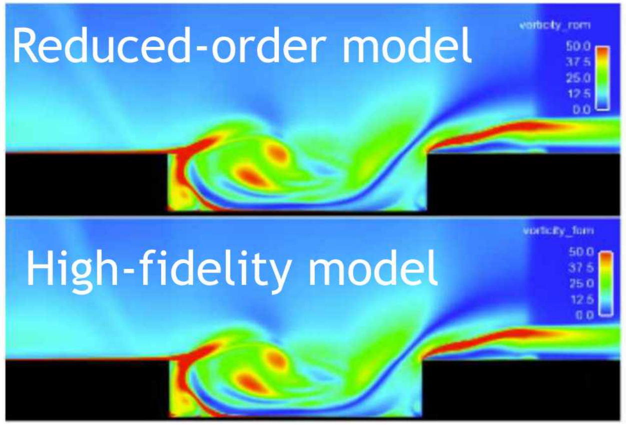
Sandia Portfolio of FY19 ASC AML Projects

Projects	Physics Constrained ML	Sparse Data ML	Credible ML	Learning Hardware	Data Specification	Talent Development
ML for Meshing	■					
Adaptive Solvers	■					
ML for SysSoft				■		
Physics-Con. ML	■					■
Variational Autoencoders		■			■	
Credible ML			■	■		
ML for UQ	■	■	■			■
Physics Informed ML-High-Dim Data	■	■	■			
ML Surrogate Models	■					■
ML Trajectories	■	■	■			
ML Interatomic Potentials	■		■			■

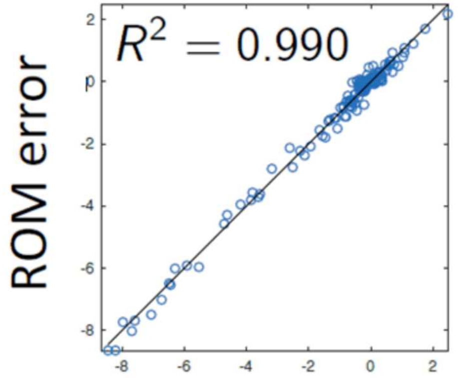
ML for Reduced Order Models

Physics Constrained ML
Credible ML

- **Problem**
  - High-fidelity computational physics simulations on HPC systems can take hours or days to execute
  - Lengthy execution time limits the design space explored during conceptual design
  - Need a faster, more efficient means of simulating complex physics problems
- **Technical Approach**
  - Create Reduced Order Model (ROM) from high-fidelity simulation data that
    - Executes faster via dimensionality reduction using autoencoders without significant reduction in accuracy
    - Preserves important physical properties (e.g., conversation laws)
    - Uses Machine Learning Error Models (MLEM) to quantify uncertainty
- **Results/Accomplishments**
  - Reduced order surrogate models and theory have been developed for turbulent flow simulations
  - Runtimes are 100-1000 times faster and are only 1% less accurate than the high-fidelity simulations
  - MLEM can predict errors with validated statistical properties



Turbulent flow vorticity field



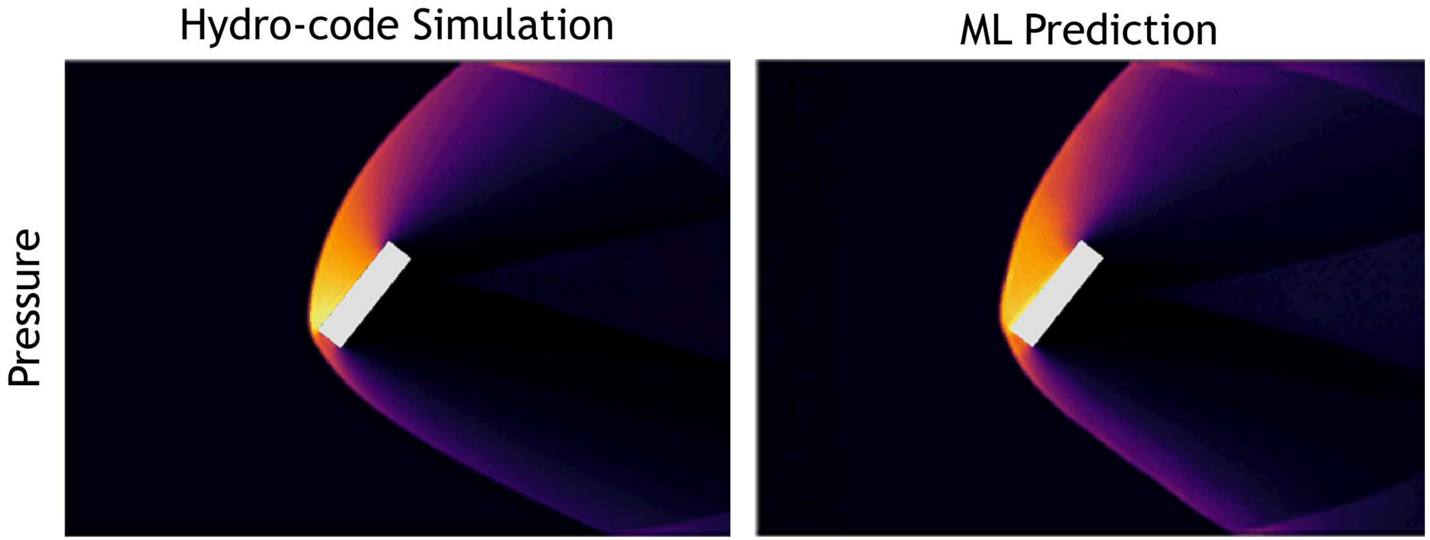
Support vector machine error prediction

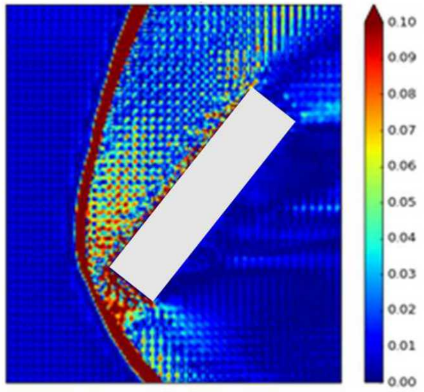
Focus Areas: Data-driven physics models and Reduced computational cost

PI: Kevin Carlberg

Accelerating Calculations of Fluid Flow via Physics-Informed Machine Learning (ML) Models

- **Problem**
  - High-fidelity simulations on HPC systems are too expensive
- **Technical Approach**
  - Train a neural network to predict the steady-state flow field
  - Guide the prediction with physical constraints (conservation laws) and aerodynamic forces (drag, lift, torque)
- **Results/Accomplishments**
  - Demonstrated >100x speed increase in 2D with < 6% average error
  - Predict > 1000x speed increase in 3D

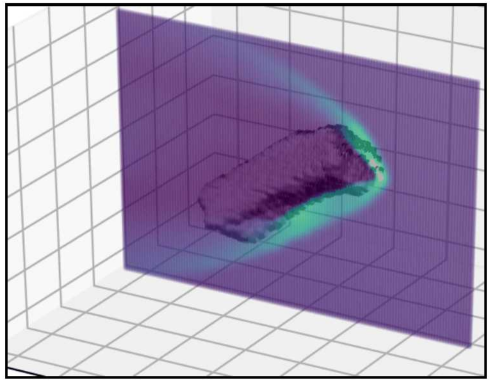




Relative error map of ML prediction

2D force	Avg Error
Drag	1.87%
Lift	5.63%
Torque	2.29%

ML model successfully predicts flow field and aerodynamic coefficients



ML prediction of pressure field around complex 3D object

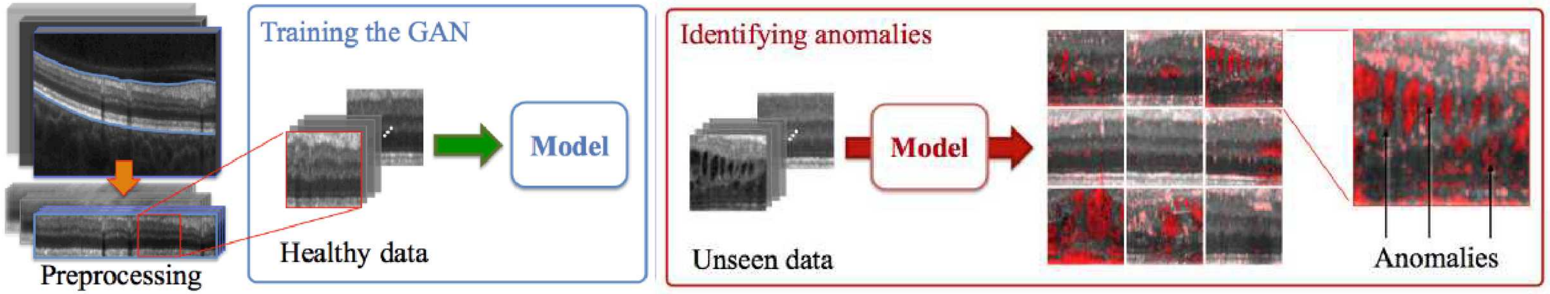
PI: John Korbin

Focus Areas: Data-driven physics models and Reduced computational cost

Deep Learning Enables Discovery of Anomalies in High-Reliability Components

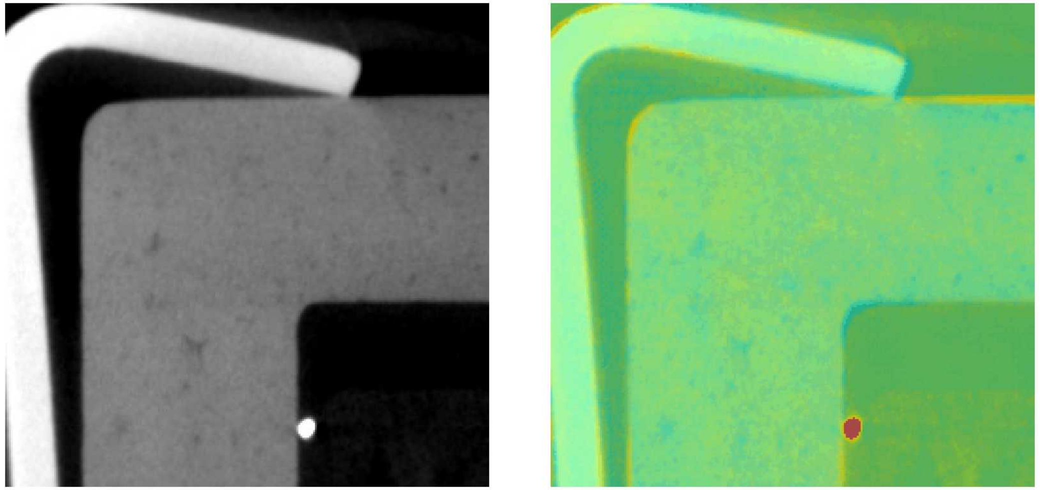
ML with Sparse Data

- **Problem**
  - It is difficult to train ML networks to identify rare/never-before-seen features with high confidence
- **Technical Approach**
  - Train a neural network (with 3000 images) to understand only "good" features
- **Results/Accomplishments**
  - Deep learning model finds anomalies in seconds that would take a human hours to find
  - Entirely unsupervised anomaly detection



Schlegl et. al. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. 2017

Real query image   Anomaly detection



X-Ray CT scan of faulty electronic component

PI: Emily Donahue

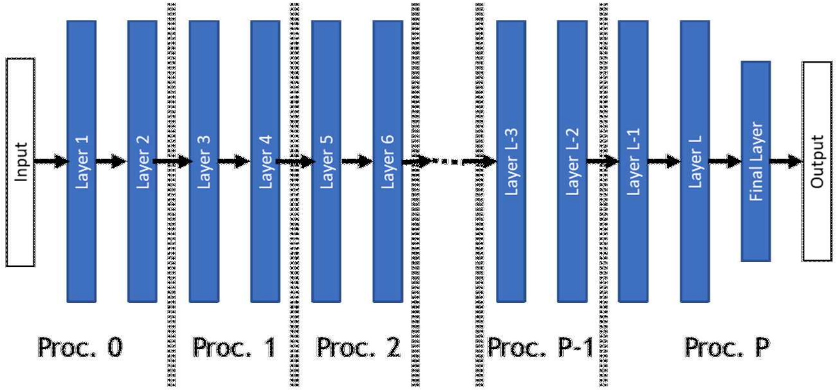
Focus Area: Anticipatory stockpile decision making

Parallel Training of Deep Residual Neural Networks

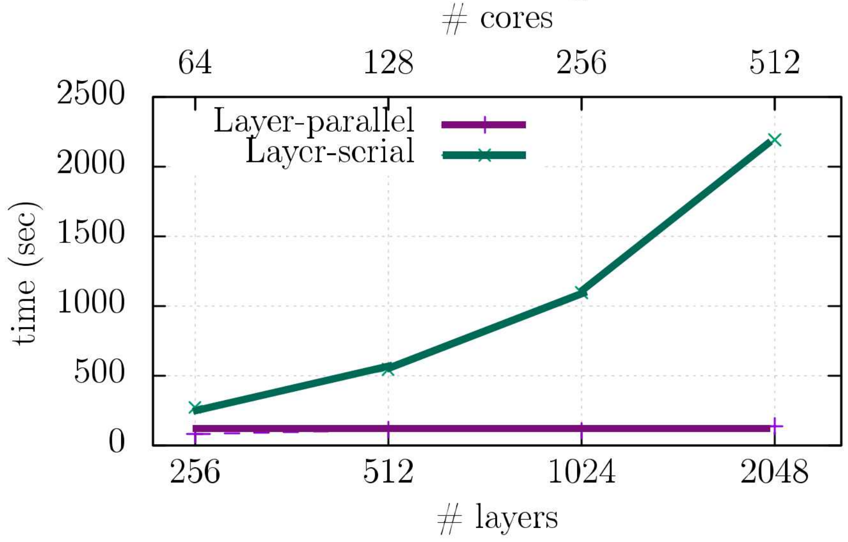
Credible ML

- **Problem**
  - Deep Learning is often viewed as a black box that requires large training sets and significant training time
  - High-consequence decisions made with ML analysis need to be explainable and credible
- **Technical Approach**
  - Training a Residual Neural Network (ResNet) is cast as an optimal control problem subject to nonlinear dynamics
  - Classical forward and backward propagation through network layers are replaced by a parallel MultiGrid Reduction In Time (MGRIT) iteration in the layer domain.
- **Results/Accomplishments**
  - Unique "Layer-Parallel" approach provides scalable speed-up over serial stochastic gradient descent approach to training (~16x at 2048 layers)
  - Theoretical basis to learning provided by dynamic system optimization methodology

Layer-Parallel training for ResNet



Weak Scaling



S. Guenther, L. Ruthotto, J. B. Schroder, E. C. Cyr, N. R. Gauger, Layer-Parallel Training of Deep Residual Neural Networks, preprint arXiv:1812.04352, 2018.

PI: Eric Cyr, ASCR funded project

Focus Area: Reduced computational cost

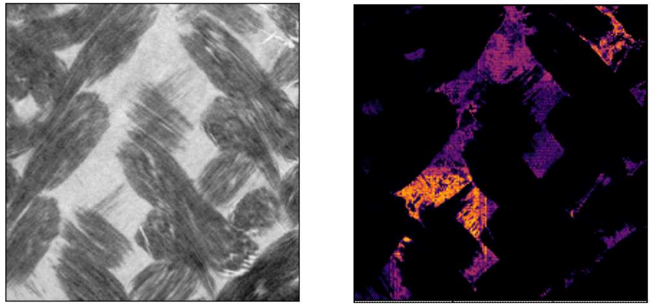
Credible ML

Uncertainty Quantification for High-Consequence Deep Learning Applications

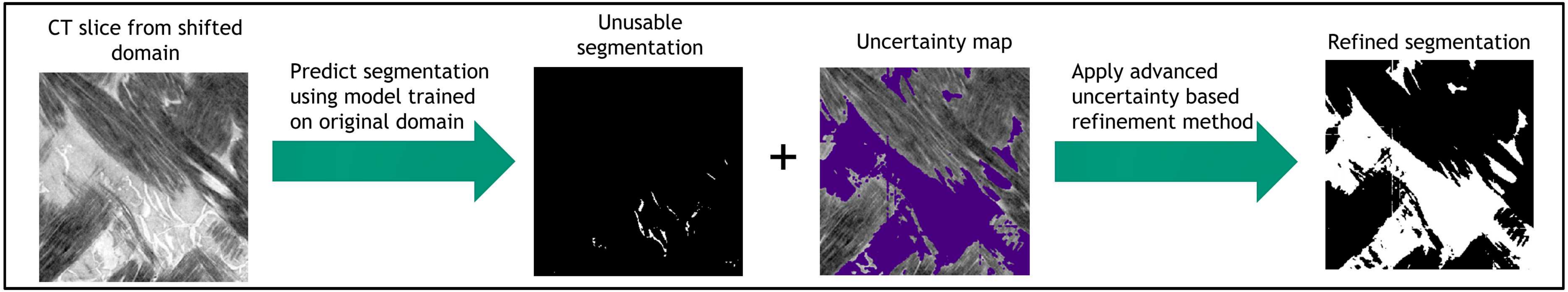
- **Problem**
  - High-consequence decisions made with ML analysis need to be informed by uncertainty measures
- **Technical Approach**
  - Dropout enables neural networks to measure uncertainty
  - Averaging the results over multiple inferences with an ML network reveals the uncertainty of a prediction
- **Results/Accomplishments**
  - Uncertainty metrics provide a measure of the model's credibility on a particular task
  - Uncertainty enables neural networks to overcome domain shift

Slide of scan of woven composite material

Uncertainty map - brighter pixel values indicate higher uncertainty



High uncertainty indicates this model should not be trusted in this domain

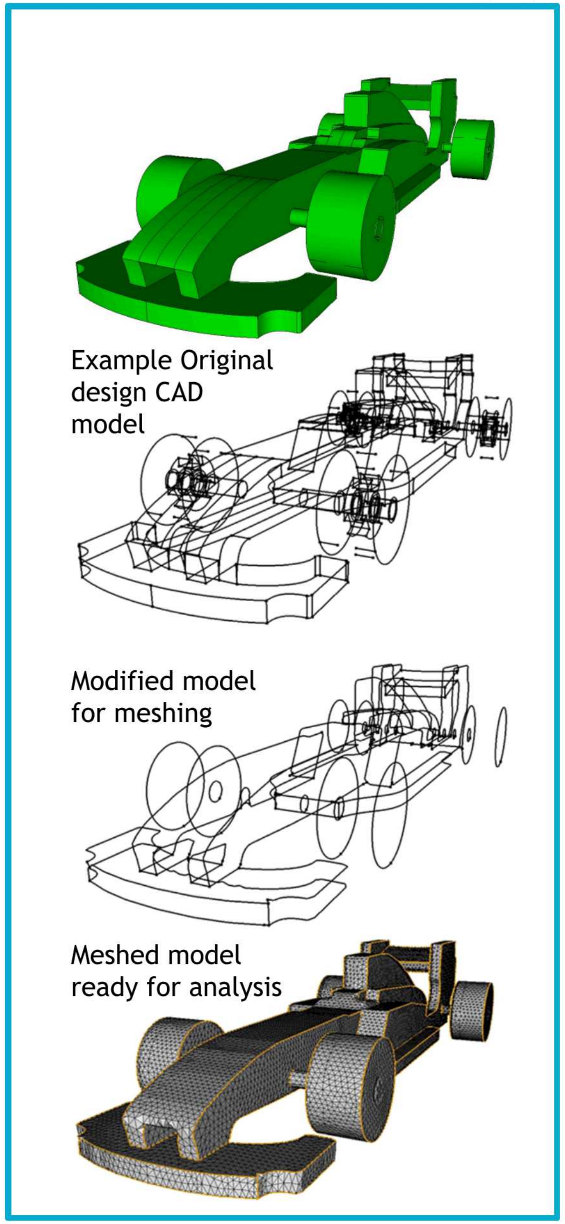


PI: John Korbin

Focus Areas: Anticipatory stockpile decision making and Enhanced experimental design

ML Enables Automatic Mesh Generation

- **Problem**
  - Geometry preparation and meshing for computational simulation is bottleneck (consuming 70%+ of analyst time)
  - Analyst/engineer must have extensive domain-specific expertise to manage many individual complex problems and tasks
  - Must produce verifiably accurate physics appropriate mesh ready for simulation
- **Technical Approach**
  - Identify tasks currently done by analysts to train machine learning models
  - Capture and label operations performed by expert using existing software
  - Build a feature library of geometric characteristics commonly encountered in CAD models and identify solutions for effectively modifying CAD for best resulting mesh
  - Explore machine learning models that provide best solutions for CAD features with associated solution labels
- **Results/Accomplishments**
  - Developed ML techniques to rank geometry-modification operations by their likelihood of yielding a meshable model
  - Provides insight on which geometric features are most useful for machine leaning, and would be relatively easy to integrate into the analyst workflow if successful



PI: Steve Owens

Focus Areas: Improved efficiency in design process and Reduced computational cost

Example Meshing Application

Power Tools

Tasks | Import or Create Geometry | Setup FEA Model | Prepare Geometry | Mesh | Validate Mesh | Define BC's | Export Mesh

Prepare Geometry

Remove Small Features
Small features can over-constrain your mesh and result in poor elements. First enter a size below which helps Cubit indentify small features.

Volume List: all

Small Curve Length 0.0074997 ...

☑ Auto Update — Detect Small Features

☑ Use Machine Learning
○ Scaled Jacobian ◉ In-Radius

Consider correcting small features listed below. Select a small feature to view the possible solutions.

Small Features

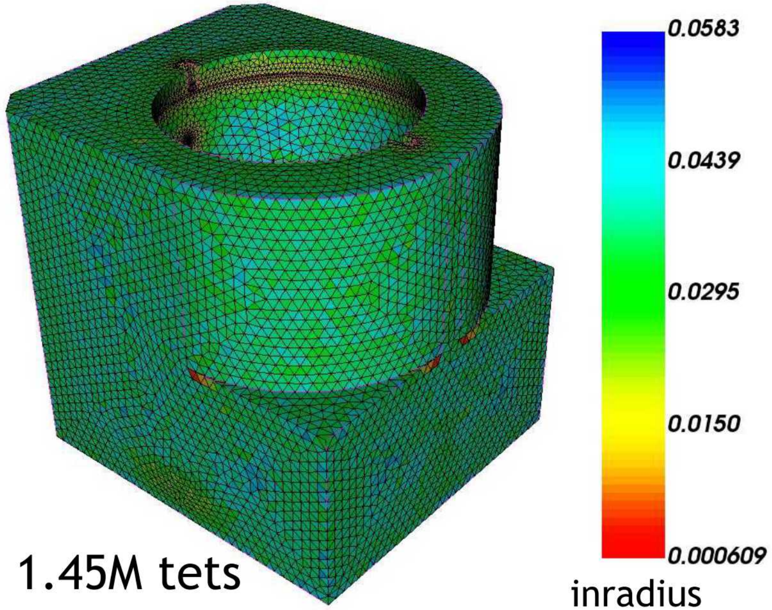
Entity ID	Entity Data
▼ Worst Entities (18)	In-Radius
▶ Curve 492	0.0046
▶ Curve 638	0.0049
Vertex 385	0.0083
Vertex 387	0.0142
Vertex 467	0.0147
Vertex 643	0.1549
Vertex 636	0.1557
Vertex 284	0.1785
Vertex 321	0.2107

Solutions

Possible Solutions
- Collapse (all virtual) to Vertex 386 [SJ: -0.076 IR: 0.108]
- Collapse to Vertex 386 [SJ: -0.063 IR: 0.107]
- Collapse to Vertex 385 [SJ: -0.058 IR: 0.104]
- Collapse (all virtual) to Vertex 385 [SJ: -0.125 IR: 0.094]
- Remove Surface 121 [SJ: -0.533 IR: -0.048]

Execute

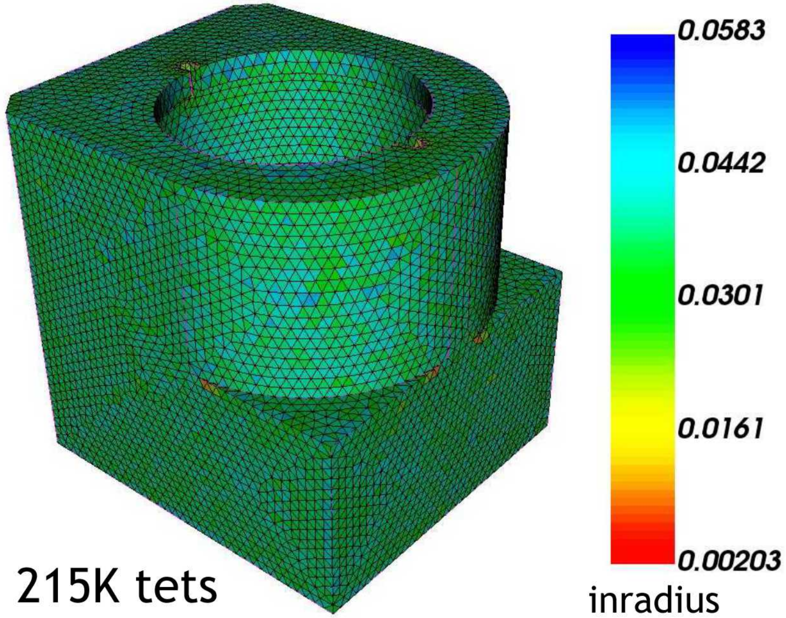
Before Executing ML Solutions



Small Features

Entity ID	Entity Data
▼ Worst Entities (12)	In-Radius
▶ Curve 132	0.1306
▶ Curve 129	0.1306
▶ Curve 131	0.1331
▶ Curve 78	0.1331
▶ Surface 24	0.2288
▶ Surface 21	0.2289
Vertex 59	0.2297
Vertex 62	0.2306

After Executing ML Solutions



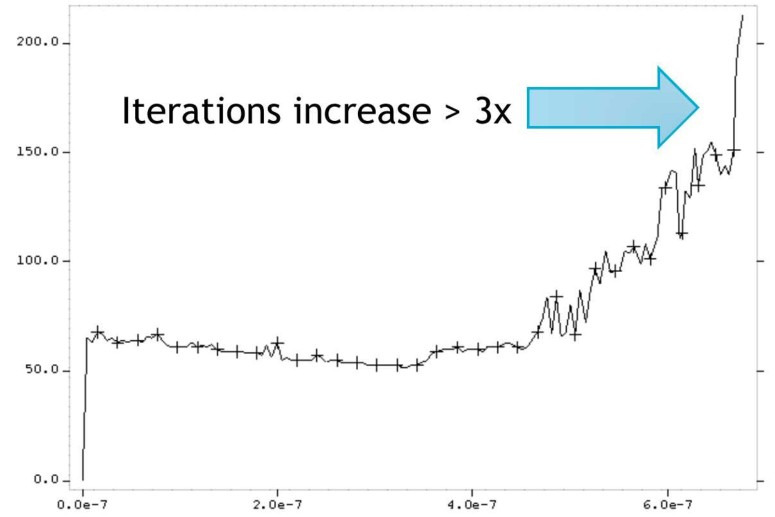
Small Features

Entity ID	Entity Data
Worst Entities (0)	In-Radius
Sharp Angle Vertices (0)	Angle
Small Curves (0)	Length
Small Surfaces (0)	Area
▶ Narrow Surfaces (1)	Area
Cone Surfaces (0)	Area

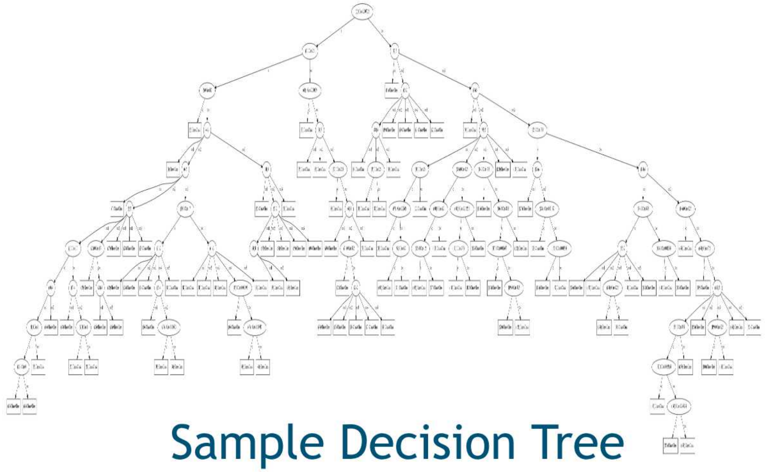
Machine Learning for Adaptive Solvers

Physics Constrained ML

- **Problem**
  - Analysts are not solver experts!
  - Changing physics may need changing solver settings
- **Technical Approach**
  - Gather mission-relevant problems
  - Sample solver settings
  - Currently have ~25k runs on synthetic problems
  - Use SNL's Avatar to generate decision trees
  - Research Question: What problem characteristics most affect solver performance?
- **Long-Term Vision**
  - App plugin to automatically pick solver settings
- **Results/Impact**
  - Decision tree looks good on smaller datasets... But still need to evaluate at scale!



Solver Iterations for an Example Problem



Sample Decision Tree

PI: Chris Siefert

Focus Area: Improved efficiency in design process

Diagnosing HPC Performance Variations

Learning Hardware

Problem

- **On HPC systems, the same job on the same system can vary in performance up to 100%**
  - Leads to poor scheduling
  - Results in reduced efficiency (costly on large-scale systems)
- **Much of the variations are caused by system anomalies**
  - E.g., Shared resource contention, firmware bugs, CPU throttling for thermal control, orphan processes from previous jobs

Technical Approach

- **Created an automated ML framework to detect and classify anomalies**
  - Based on resource usage data (CPU, memory, network)
  - Using concise features, low-overhead
  - Generally applicable

Results/Accomplishments

- **Framework outperforms existing methods**
  - Evaluated in two different HPC environments, over 0.97 F-Score
- **Easy-to-compute statistical feature extraction**
  - Storage overhead reduced to less than 10%
  - Computation overhead below 1% of a single core

PI: Vitus Leung

Focus Area: Reduced computational cost

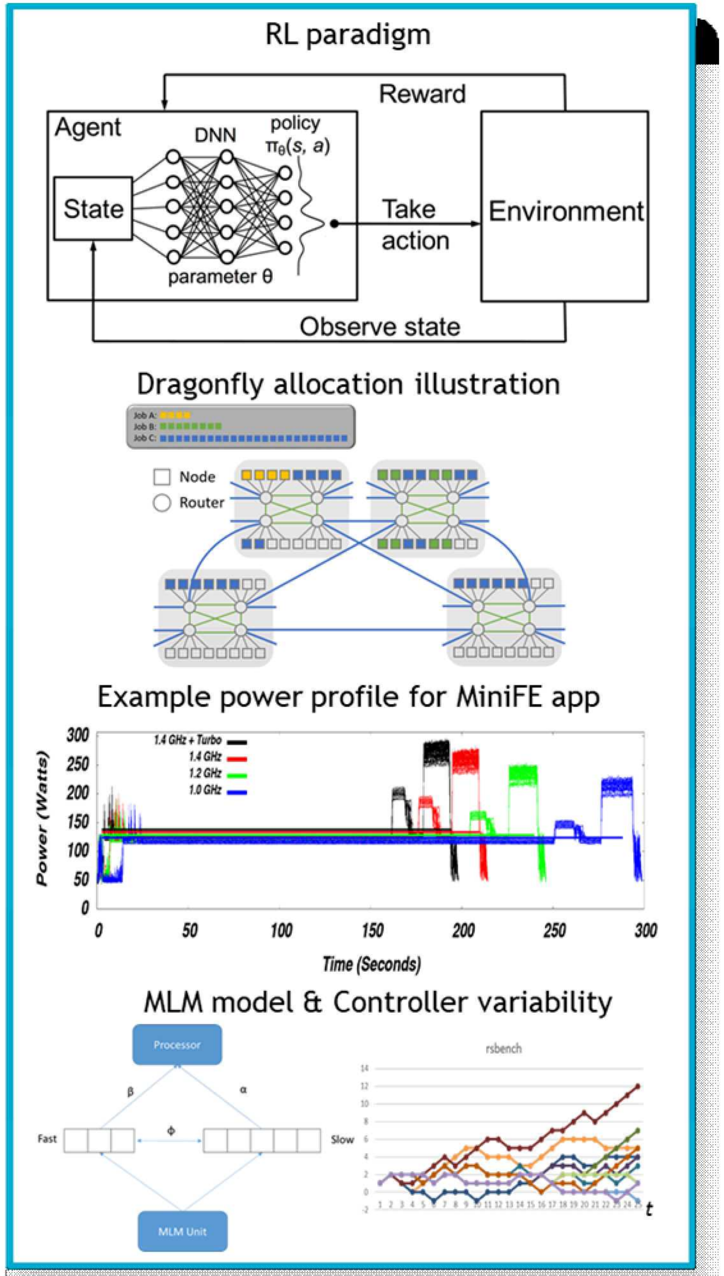
ML for System Software

- **Problem**
  - The growing complexity of computing systems, ranging from cell phones to supercomputers, is becoming difficult for developers to manage
  - More intelligent and automated mechanisms are needed to avoid unintended resource oversubscription and manage the placement and movement of data and computation in extremely heterogeneous systems

- **Technical Approach**
  - We believe that ML can be utilized to infer intelligent approaches to resource management
  - In particular Reinforcement Learning (RL)
    - ML paradigm where an agent makes actions given state observations from an environment; the environment subsequently emits rewards and new state observations, based on the agent's actions

- **Technical Direction**
  - Job allocation for Dragonfly Networks
  - Adaptive P-State control for HPC workloads
  - Multi-level Memory Management

- **Results/Impact**
  - Developed RL MODL (Mathematical Optimizations for Deep Learning) library
  - Quantization study of RL showing results on continuous control problems

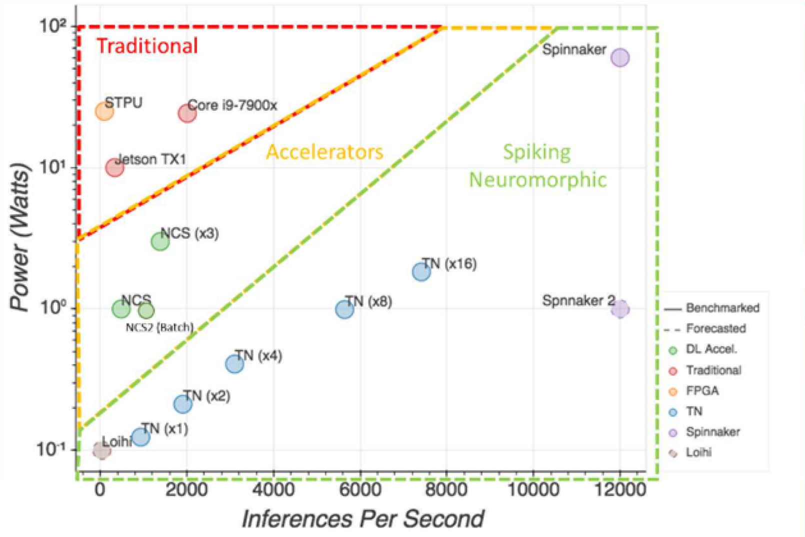


PI: Craig Vineyard

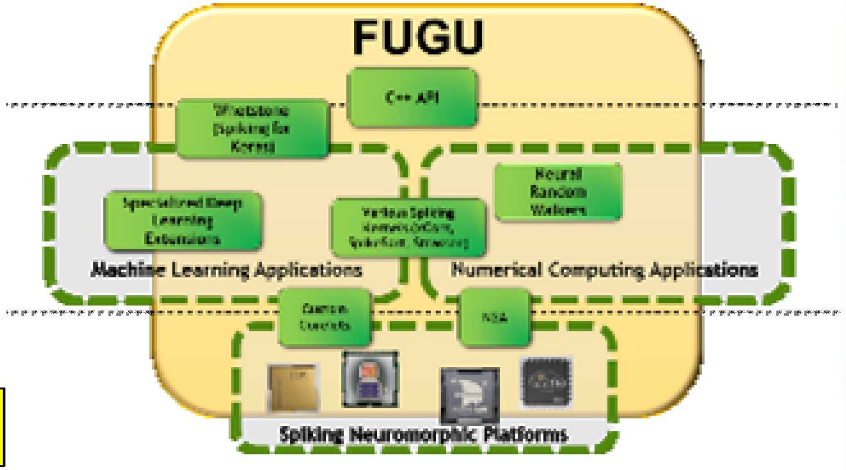
Focus Area: Reduced computational cost

Neuromorphic Software Stack

- **Problem**
  - Potential of large power savings (100-1000x) by using spiking neural networks instead of artificial neural networks
  - Programming spiking neural networks is difficult because of event-driven nature of computation
- **Technical Approach**
  - Develop Whetstone – a deep learning library focused on feed-forward networks trained through back propagation
    - Integrates with Keras (which in turn interfaces to Tensorflow)
  - Develop Fugu – a framework for linking existing spiking neural networks and expanding to solve scientific computing problems
    - Independent of the hardware that runs the neuron computation
- **Results/Impact**
  - Whetstone - Initial open-source prototypes have been created with a port to SpiNNaker hardware
  - FUGU – Shared with Tri-Lab team



WHETSTONE



PI: Brad Aimone

Focus Area: Reduced computational cost

Conclusions

- Machine learning will provide new capabilities for scientific and engineering applications
  - Reduced order surrogate models for scientific/engineering problems
  - Ability to identify anomalies and regions of interest in inspection and surveillance data
  - Correlating and certifying simulation and experimental results
  - As a supplement for physics models, ML might help us learn what is wrong/missing in physics models and aid in experimental design
  - For agility of application workflows (automating processes)

- Machine learning will provide new capabilities for HPC system administrators, facilities, and dev-ops
  - Help model complex behaviors (e.g., failures, degradation, energy)
  - Automate/adapt usage to comply with more complex policy (e.g., energy consumption)

- Machine learning will provide new capabilities for HPC systems software
  - Adaptable resource management (e.g., network, memory, storage, energy)
  - “Smart” data-movement for Exascale runtimes

Questions?