Learning-based predictive models: a new approach to integrating large-scale simulations and experiments

BOUT++ Workshop, August 14-17, 2018

Brian K. Spears

LLNL-PRES-740570

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Data science advances offer an opportunity to change how we do predictive science at the Laboratory

Today, I'd like to do a few things

- Identify a key challenge in predictive science at the lab
- Show some ways we've tried to meet that challenge with deep learning
- Inspire you to consider these techniques for your own work
- Entertain you a bit

At LLNL, and everywhere, we love our models

At LLNL, and everywhere, we love our models

At LLNL, and everywhere, we love our models

beautiful simulation result

theorist

designer

code developer

At LLNL, and everywhere, we love our models

beautiful simulation result

theorist

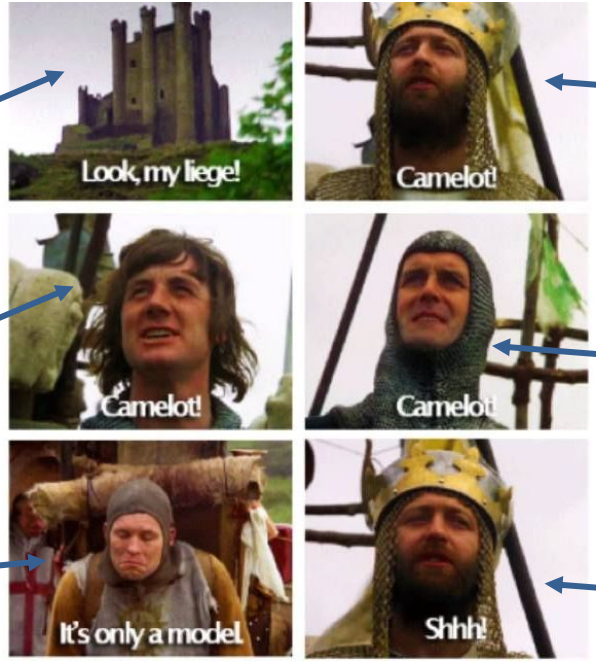
designer

code developer

experimentalist equipped with data

At LLNL, and everywhere, we love our models

beautiful simulation result



theorist

designer

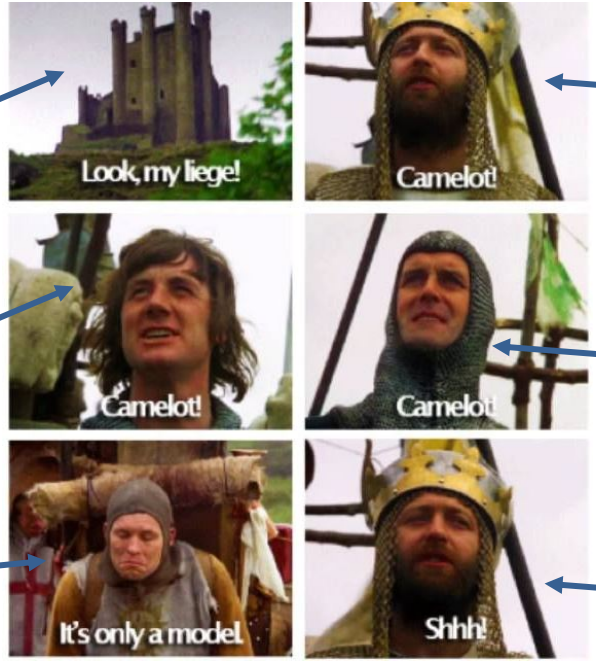
code developer

experimentalist equipped with data

typical response

At LLNL, and everywhere, we love our models

beautiful simulation result



theorist

designer

code developer

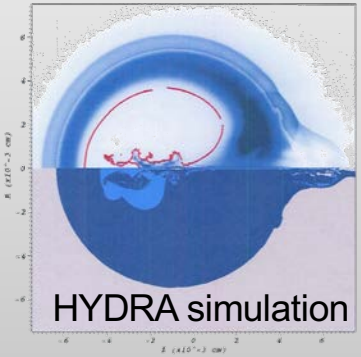
experimentalist equipped with data

typical response

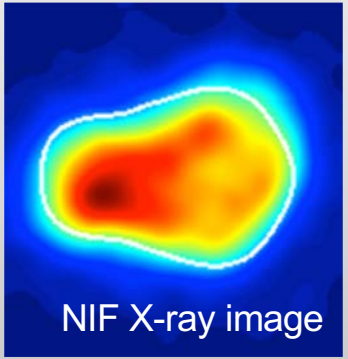
Experiments are not always kind to our models

We strive to advance our predictive capability by challenging simulation with experiment

Traditional pillar
High-performance computing

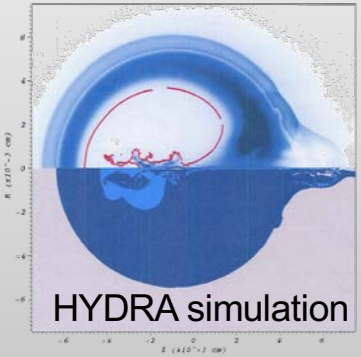


Traditional pillar
Large-scale experiments



We strive to advance our predictive capability by challenging simulation with experiment

Traditional pillar
High-performance computing

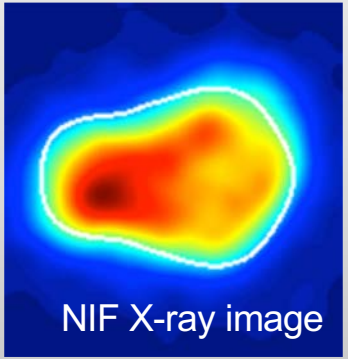


We compare a few key scalars – leaving our models less constrained

Y_{sim}
$T_{ion,sim}$
$P_{2,sim}$

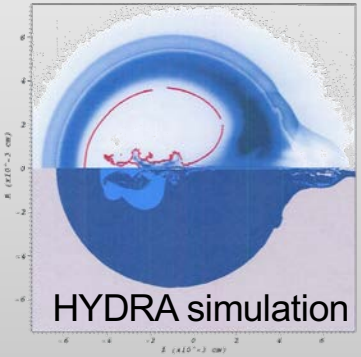
Y_{expt}
$T_{ion,expt}$
$P_{2,expt}$

Traditional pillar
Large-scale experiments



We strive to advance our predictive capability by challenging simulation with experiment

Traditional pillar
High-performance computing

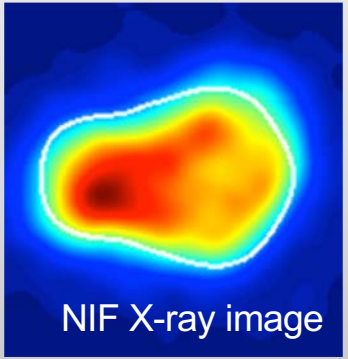


We compare a few key scalars – leaving our models less constrained

Y_{sim}
$T_{ion,sim}$
$P_{2,sim}$

Y_{expt}
$T_{ion,expt}$
$P_{2,expt}$

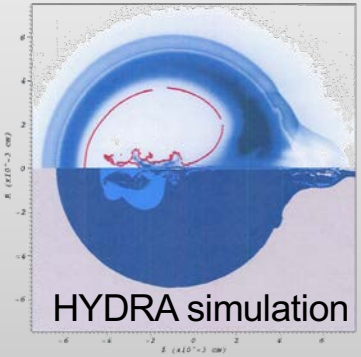
Traditional pillar
Large-scale experiments



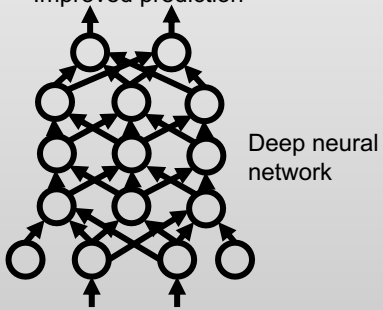
We need new techniques that improve our predictions in the presence of experimental evidence

Machine learning can dramatically improve predictive modeling across the Laboratory

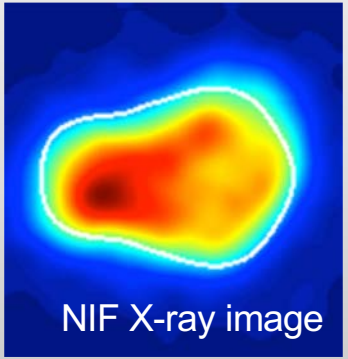
Traditional pillar
high-performance computing



New pillar
Machine learning to compare simulation and experiment



Traditional pillar
Large-scale experiments



Machine learning will allow us to use our full data sets to make our models more predictive

We need three technological advances to transform how we do predictive science at the Laboratory

- We need improved models
  - That fully utilize available data sets
  - That estimate uncertainty
  - That improve by exposure to experimental data
- We need software tools to develop and guide those models
- We need computational platforms that support the advancement of these predictive tools

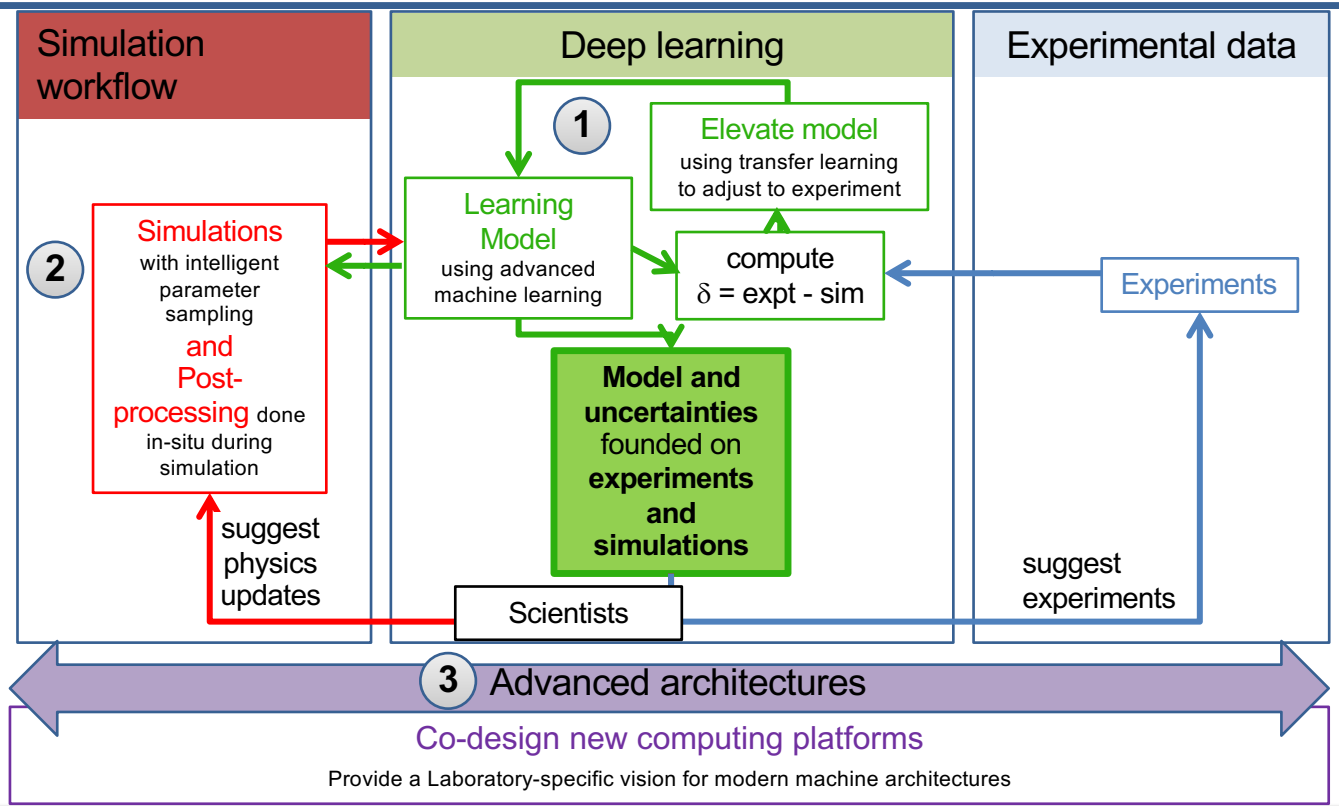
Simulation, experiment, and deep learning

Computational workflows and big data

Heterogeneous exascale computers

These advances can improve the modeling chain across programs and missions

We need a framework that couples simulation and experiment using machine learning to improve prediction



I am proud to present the work of a wonderful team

Machine-learned Predictive Models
PI, Brian Spears

Machine Learning Element
Timo Bremer

Architectures
Jay Thiagarajan
Rushil Anirudh
Shusen Liu

Elevation and UQ
Jim Gaffney
Bogdan Kustowski (new hire)
Gemma Anderson (new hire)
Francisco Beltran
Michael Kruse

Large-scale Learning
Sam Ade Jacobs
Brian Van Essen
David Hysom
Jae-Sung Yeom

Workflow Element
Luc Peterson

Workflow Tools
Peter Robinson
Jessica Semler
Luc Peterson
Ben Bay (new hire)
Scott Brandon

Intelligent Sampling
Vic Castillo
Bogdan Kustowski
Kelli Humbird (LG scholar)
David Domyancic
Richard Klein

In-situ tools
John Field
Steve Langer
Joe Koning

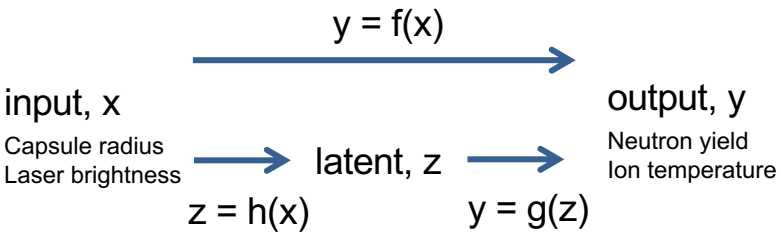
Data Harvesting
Michael Kruse
Dave Munro
Robert Hatarik

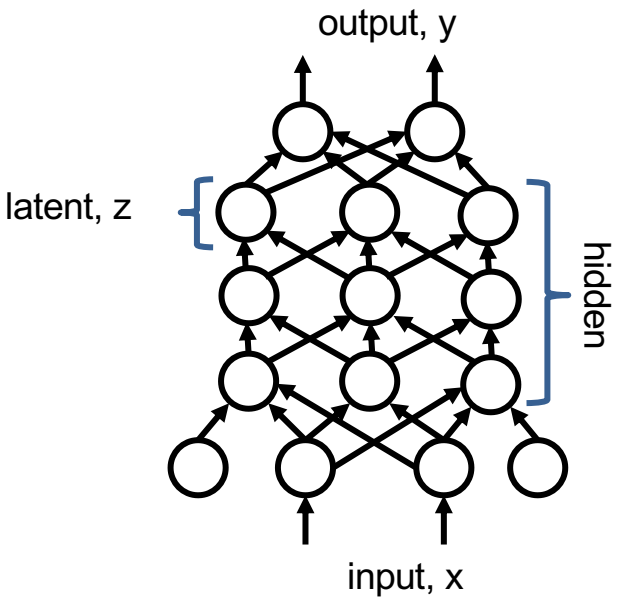
We are developing a learning code to bridge the gap between simulation and experiment

- **Multi-modal deep learning** that harnesses powerful correlations in physics data
- **Uncertainty estimation** that provides confidence measures for both outputs and inputs to explain data
- **Model elevation** that improves predictions and uncertainties in face of experiment

Deep learning allows us to learn structure in data

- We use deep neural net models to map inputs to outputs
- Deep neural networks better capture rich data structure
  - Hidden layers build representation of data
  - Called a latent (or feature) space spanned by latent variables
  - Learn by minimizing the loss function (prediction matches truth)

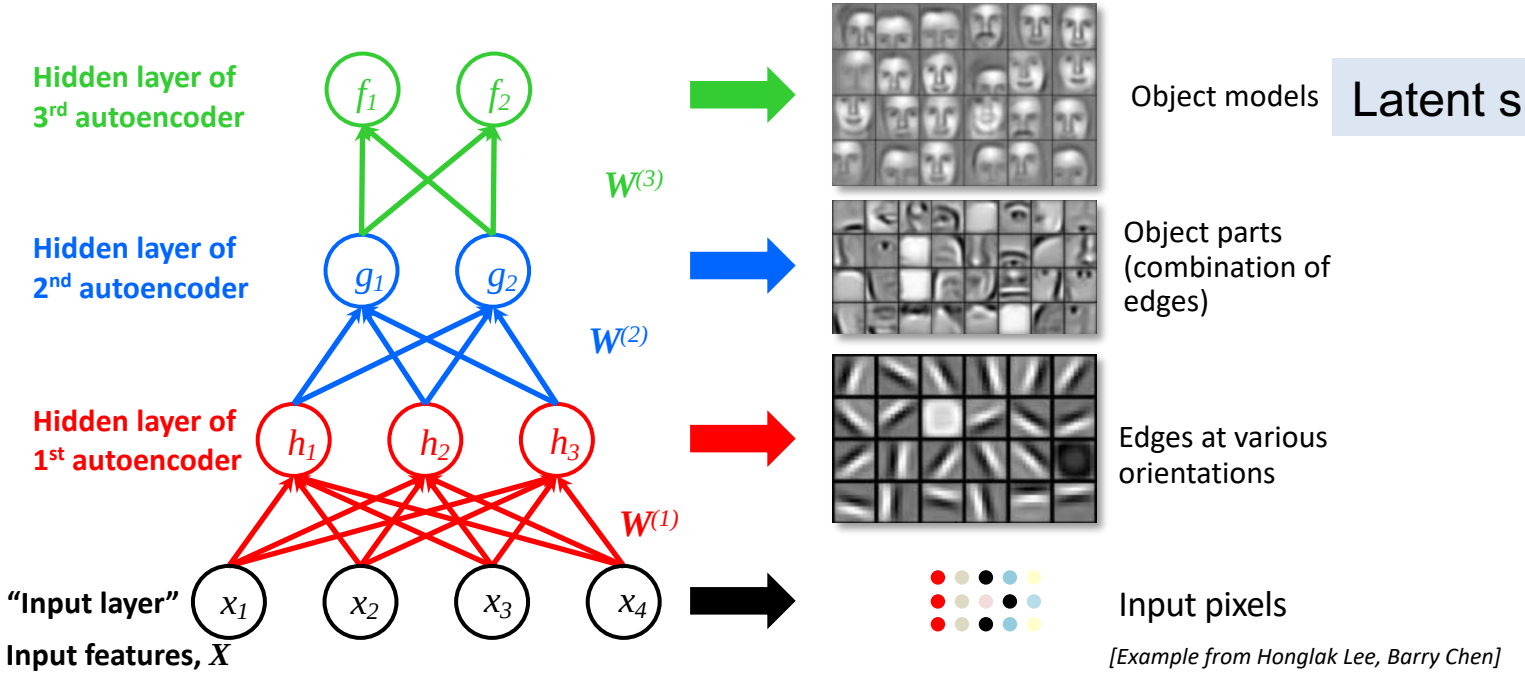




Engineering and exploiting the latent space is one of our key strategies

Contemporary deep learning: a guide for practitioners in the physical sciences arXiv:1712.08523

Deep neural networks build efficient descriptions of data to improve predictions

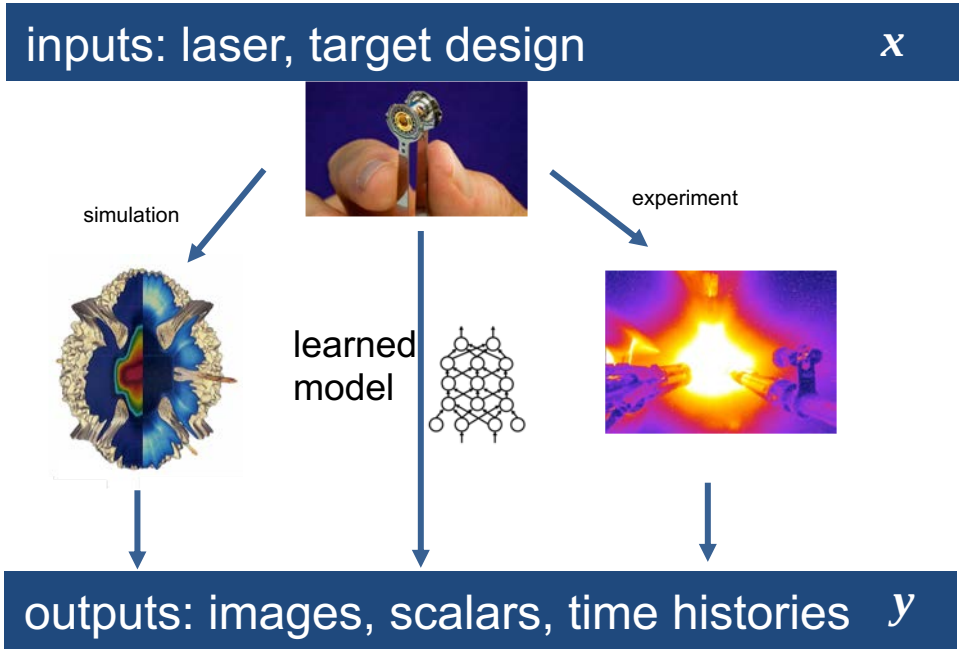


[Example from Honglak Lee, Barry Chen]

We need these efficient descriptions for ICF, Weapons, and other programs

For scientific prediction, we want our learned model to meet a few key requirements

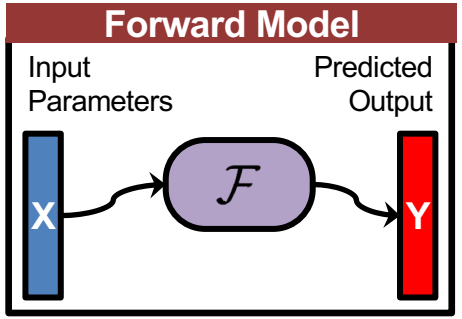
Inertial Confinement Fusion (ICF)
as a testbed problem



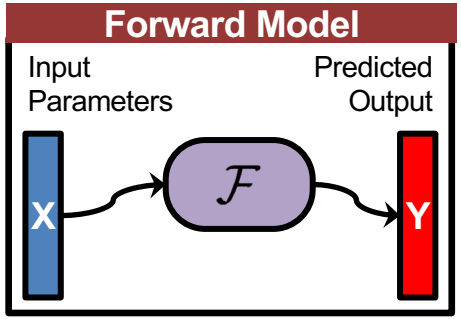
<u>Design requirements for learned models</u>

1. Use all the available data signatures
2. Be physically consistent
3. Be self consistent

We developed a cyclic system of sub-networks to engineer required performance features

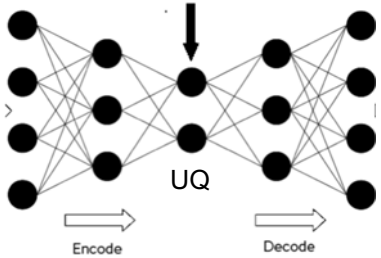


We developed a cyclic system of sub-networks to engineer required performance features



① **Surrogate Fidelity Loss**

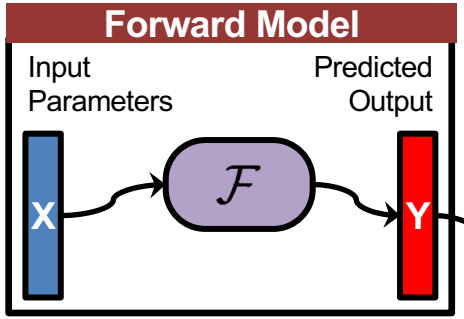
Learn a metric for comparing outputs



Performance features

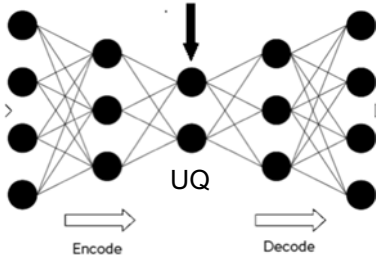
1. **Uses all the data**
   engineers the latent space

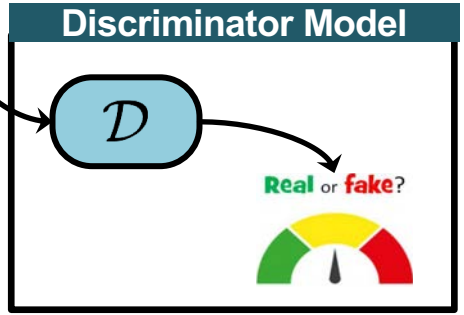
We developed a cyclic system of sub-networks to engineer required performance features



① **Surrogate Fidelity Loss**

Learn a metric for comparing outputs



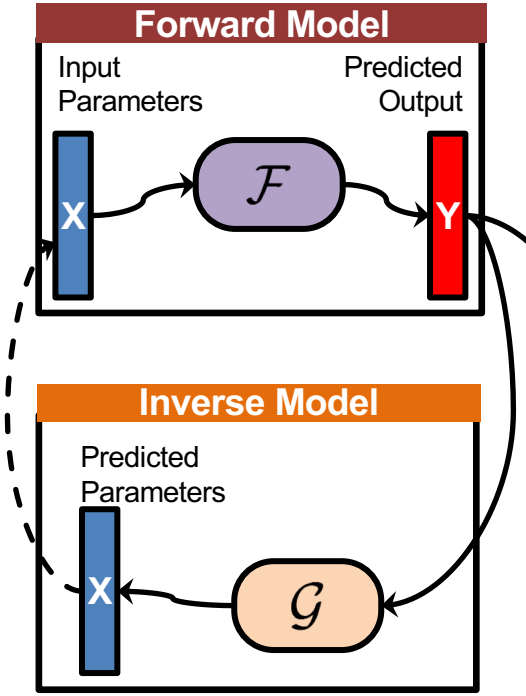


② **Physical Consistency Loss**

Performance features

1. **Uses all the data**
   engineers the latent space
2. **Enforces physical consistency**
   predictions look like training examples

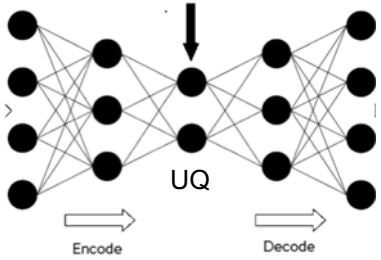
We developed a cyclic system of sub-networks to engineer required performance features

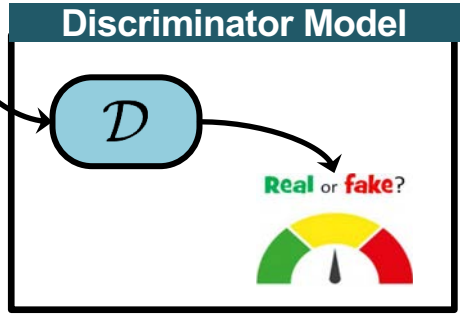


③ Cycle Consistency Loss

① Surrogate Fidelity Loss

Learn a metric for comparing outputs





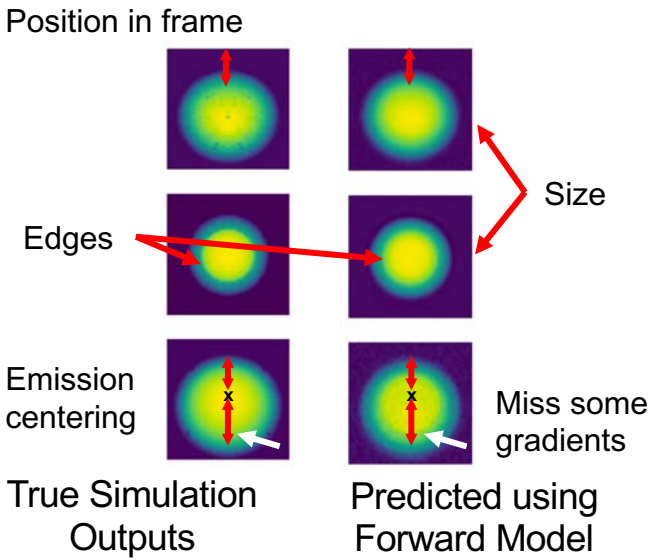
② Physical Consistency Loss

Performance features

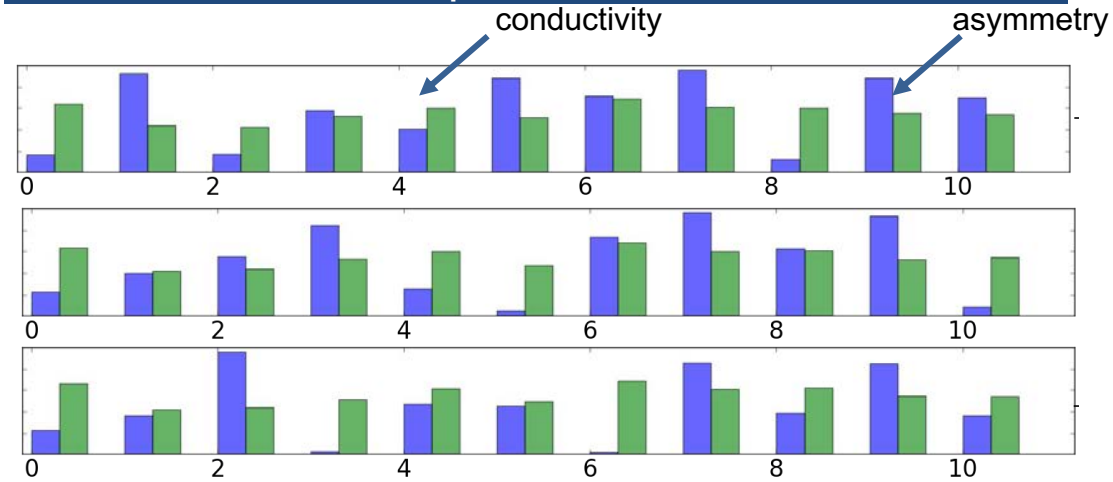
1. **Uses all the data**
   engineers the latent space
2. **Enforces physical consistency**
   predictions look like training examples
3. **Enforces self consistency**
   regularizes ill-posed inverse

The learning system reproduces and recovers key physics information

Integral features are predicted without explicit programming



Cyclic system infers input parameters from output observations



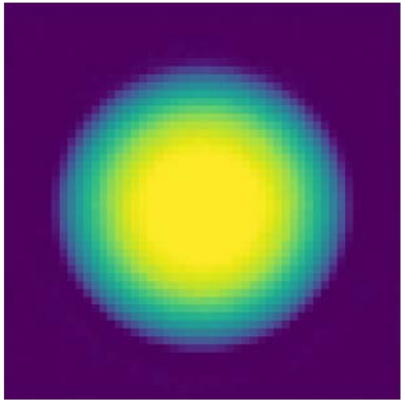
True input parameters (Blue)
Inferred input parameters (Green)

Sensitive parameters are recovered with higher fidelity

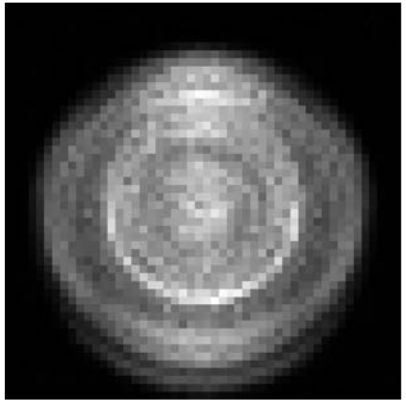
The neural network performance requirements have led to successful prediction

Variational extensions have equipped all output quantities with uncertainty measures

Predicted image and its error map

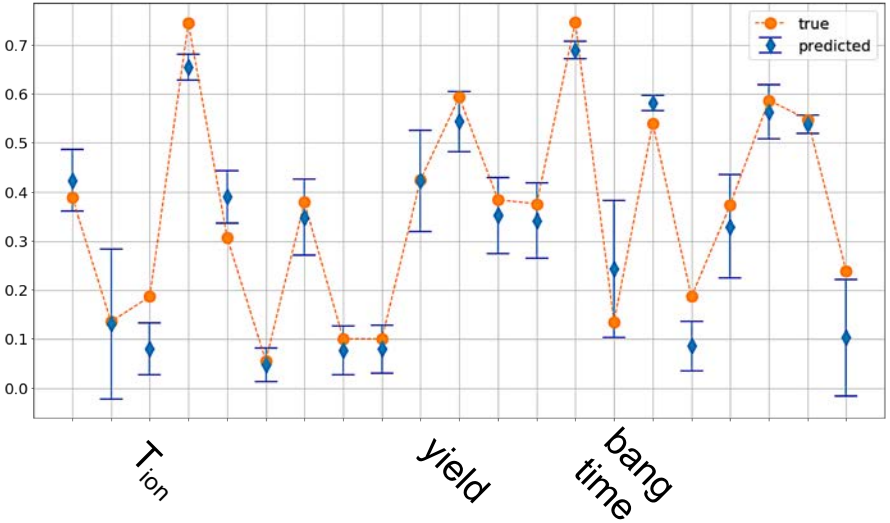


Predicted image
(pixel value)



Error map
(pixel-wise variance)

Predicted scalars and their error bars



Our predictive tools are prepared for statistical comparison with experiment

Depending on the situation, networks can avoid or inherit human bias

- Learned models know what they're taught, and only what they're taught
- Humans (even scientists and engineers) can be distracted by context

Audience interaction

Find the toothbrush in 1 second!

From Heather Murphy Oct. 6, 2017 NYTimes

Neutrogena
Beach Defender
70

Audience interaction

Is there a parking meter present?

ICA

Audience interaction

Trained neural nets recognize the meter with very high accuracy.

Humans often miss the meter*.

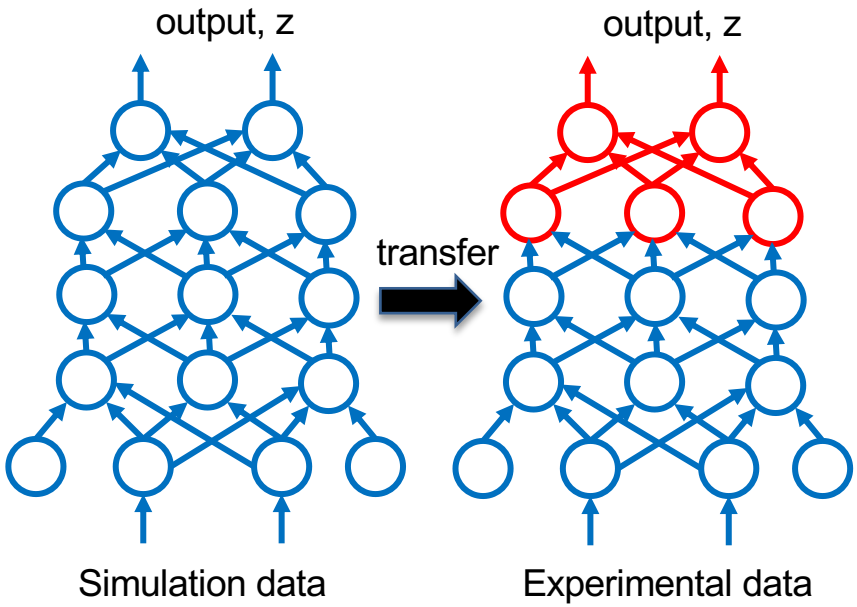
Expectations (e.g., about scale) sometimes prevent us from finding obvious patterns.

But, what if we've used our simulations to build in bias?

* "Humans, but Not Deep Neural Networks, Often Miss Giant Targets in Scenes" Miguel P. Eckstein, Kathryn Koehler, Lauren E. Welbourne,Emre Akbas

Next, we turn to transfer learning to remove simulation bias and better match experimental data

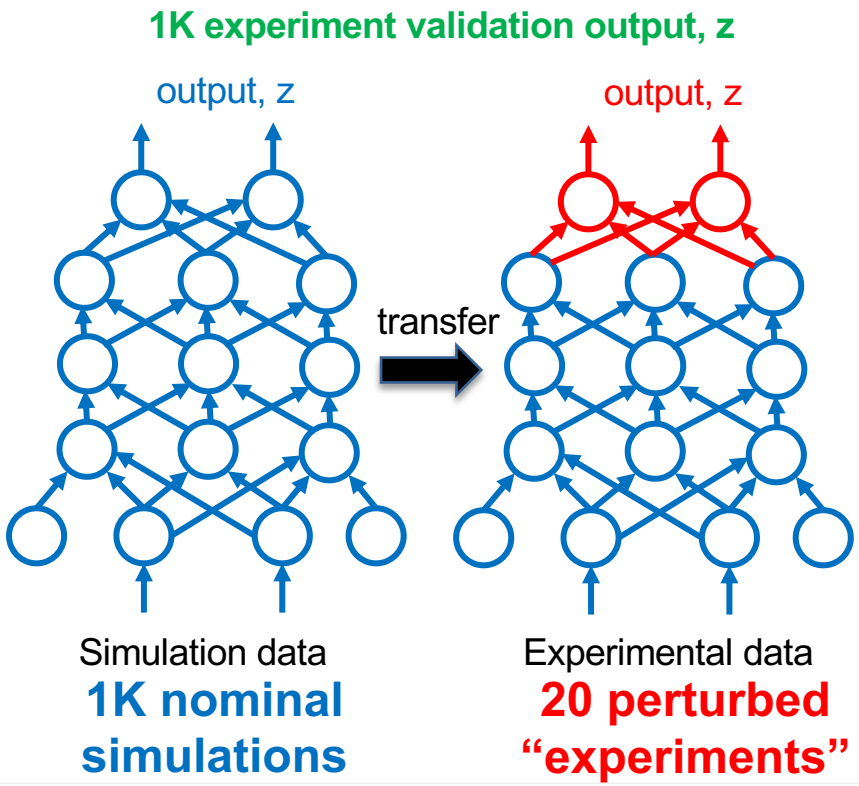
- Train the network on simulated data
- Re-train networks to predict experimental data
- Well-suited to ICF data
  - Improves prediction accuracy
  - Requires much less data than initial training
  - Measures discrepancy as a *function* of input parameters



Transfer learning produces elevated models that incorporate simulation and experiment

We've tested transfer learning with simplified data sets

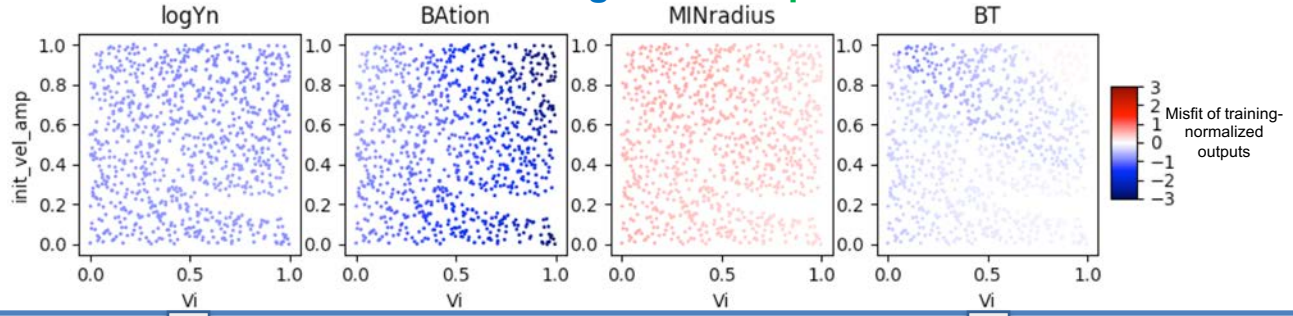
- **Simulation data set**
  - 1K sims with nominal physics
- **Experimental data set**
  - **Transfer data:** 20 "experiments" with altered physics
  - **Validation data:** 1K next "experiments" with altered physics
- **Compare predictions before and after transfer learning**
  - **Simulation surrogate vs validation data**
  - **Elevated surrogate vs validation data**



Numerical tests show successful transfer learning using few experiments

Large prediction error without transfer learning

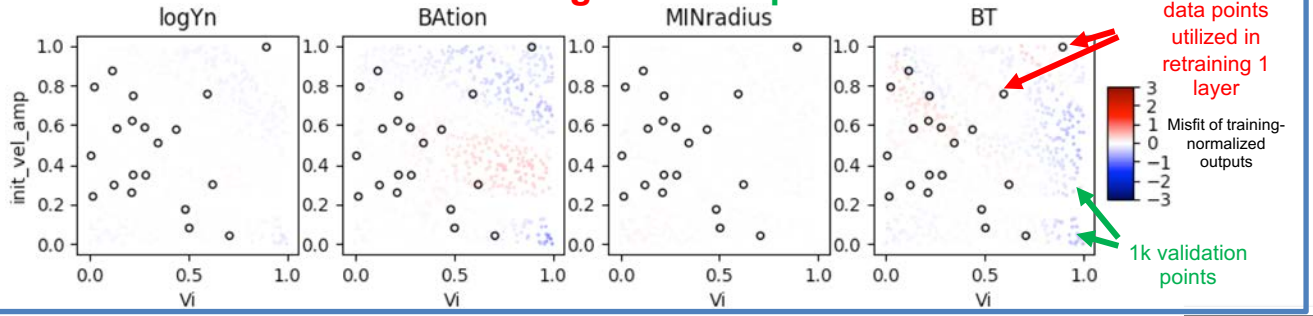
difference between **simulation surrogate** and **experiment validation**



Transfer learning

Smaller prediction error after transfer learning

difference between **elevated surrogate** and **experiment validation**

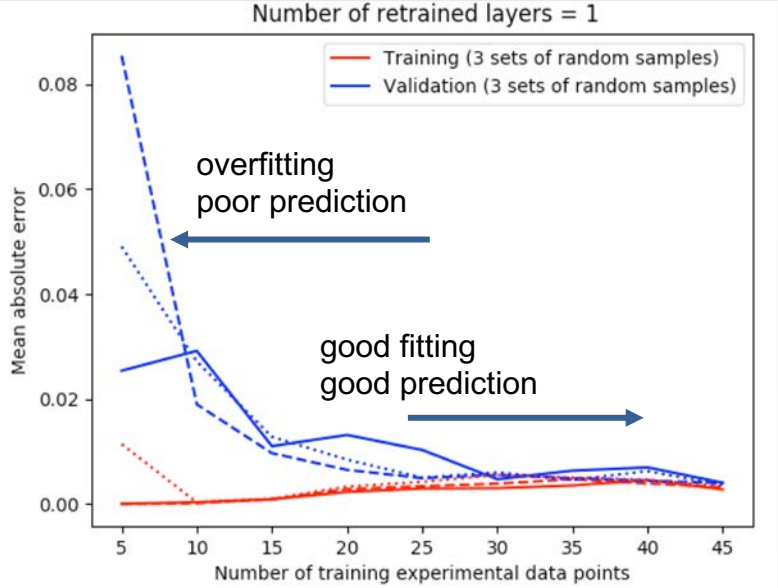


We are developing a framework to describe the applicability of transfer learning for NIF experiments and Laboratory missions

Three key features that control applicability of transfer learning

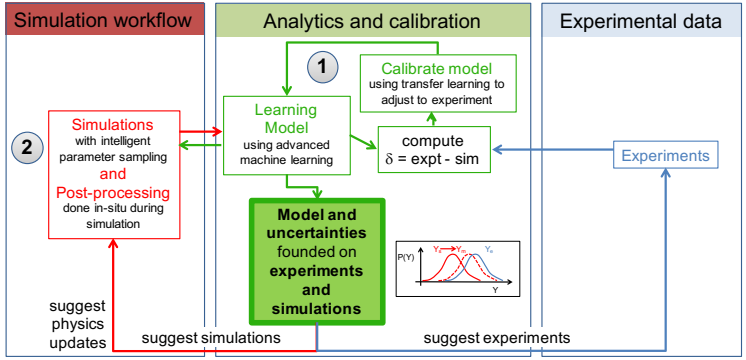
- **Size of discrepancy** between **simulation model** and **true experiment model**
- **Model capacity** needed to capture the discrepancy
- **Experimental data volume** needed to satisfy the model capacity

Amount of retraining data impacts predictive quality



Deep learning needs big data, so we'd better be able to produce it

- Management of simulation production and sampling
- Reduction of raw simulations to observables
- Driving the execution of the learning models

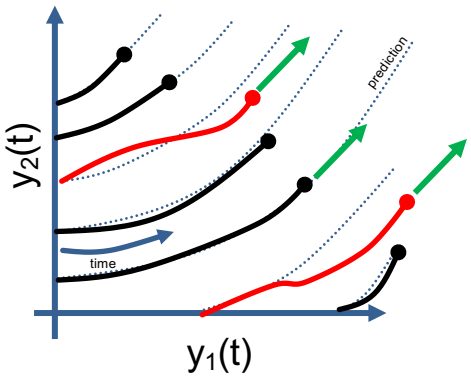


Even at very large scale, we must choose carefully which simulations to execute

Speculative sampling

- During the run, is the simulation evolving as predicted?
  - Yes? No new information. Terminate. Invest in a new simulation.
  - No? Unpredicted behavior! Continue.
- Speculate on many more simulations than we can finish.
- More completely probe parameter space for further cost reduction.

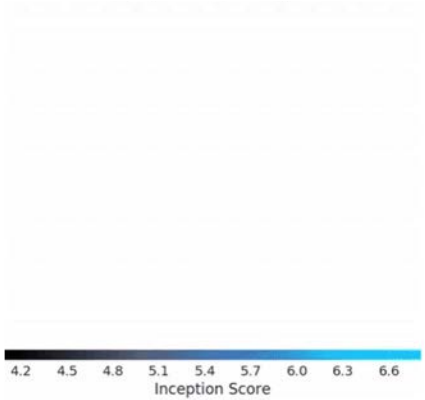
Novel approach



Speculative sampling may require far less data than random sampling

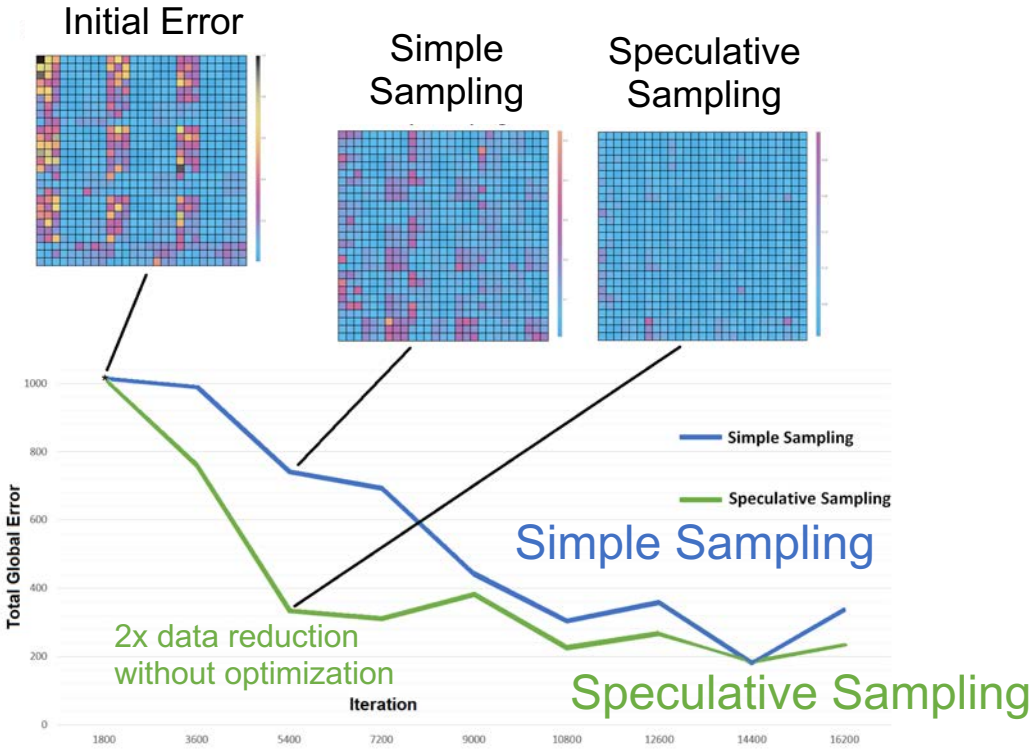
Initial speculative sampling experiments delivered better learned models for much less data

Agent-based exploration
Deep Mind, November, 2017



Model error in local regions

Total Global Model Error vs Iteration



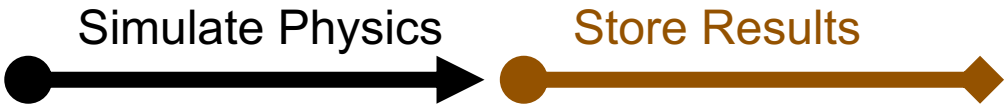
Castillo *et al.*

The potential benefits of speculative sampling come at a cost – a new simulation workflow paradigm

- Not all simulations complete; may restart
- An inherently multi-task workflow
  - Simulators
  - Learners
  - Predictors
  - Controllers
  - Archivers
- Tasks require different hardware
  - Simulate on CPUs
  - Learn on GPUs
- Many asynchronous decisions
  - But, need global coordination
- Current practices that pose obstacles to speculative sampling
  - File-system based task coordination
  - Single-machine ensembles
  - Pre-defined & dedicated resources for each ensemble & each sample
    - eg single batch job, dedicated nodes

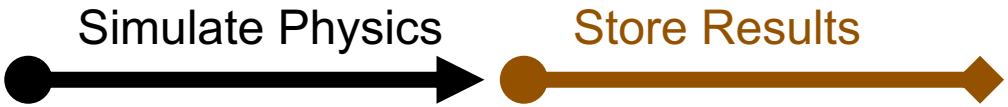
We are building a new kind of workflow to enable HPC speculative sampling

Existing simulation workflows from ICF and elsewhere are largely linear



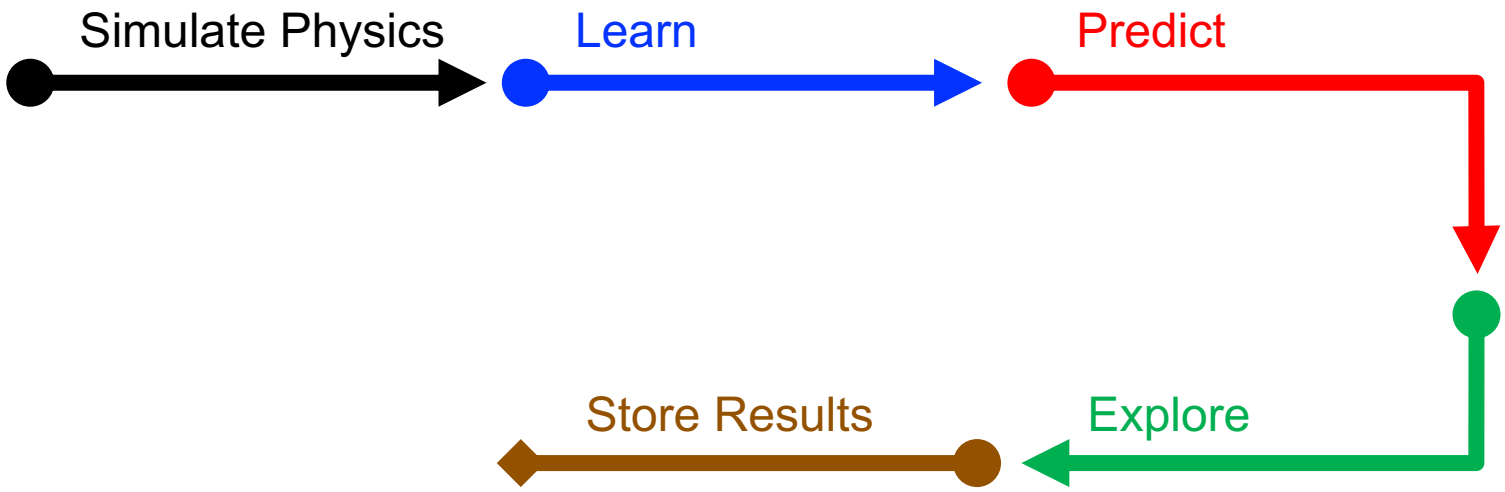
Robinson, Semler *et al.*

A speculative sampling workflow is different



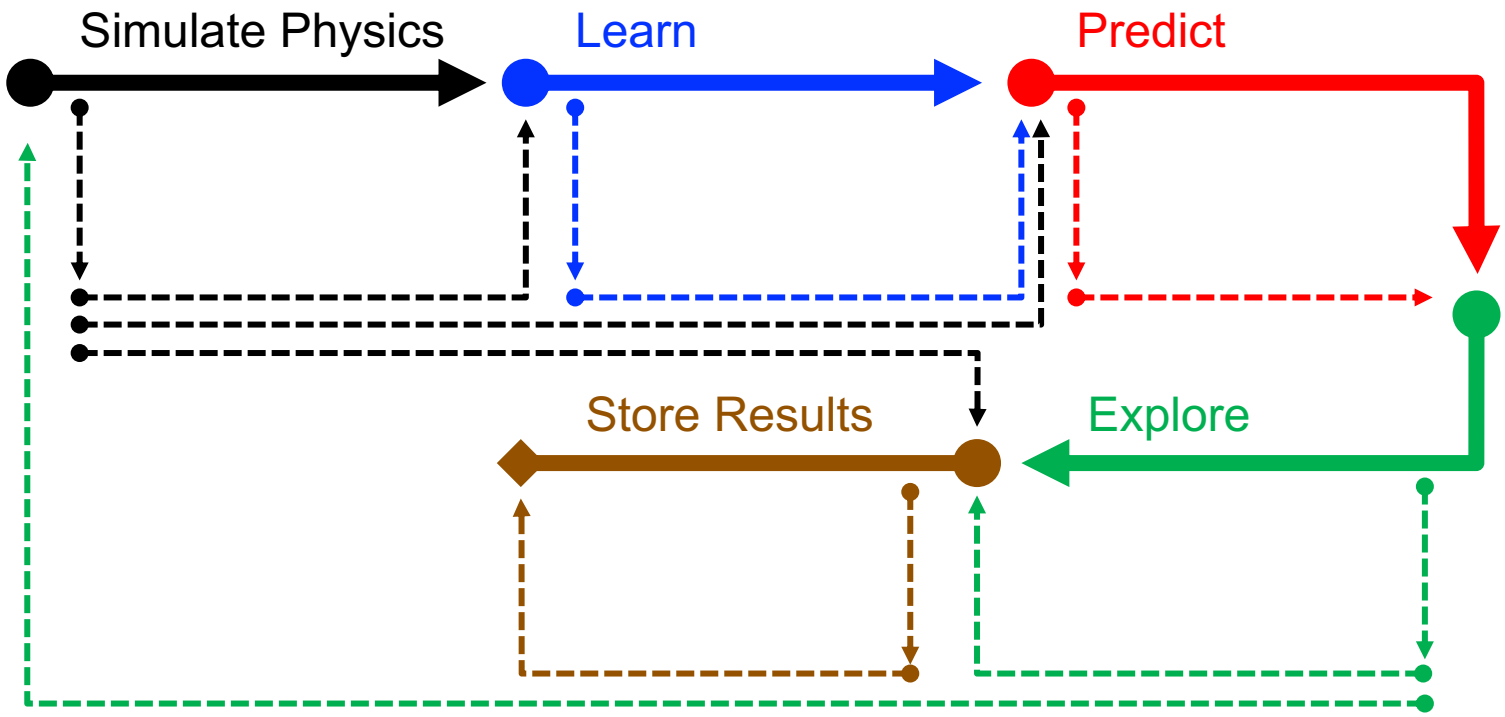
Robinson, Semler *et al.*

A speculative sampling workflow is multi-task



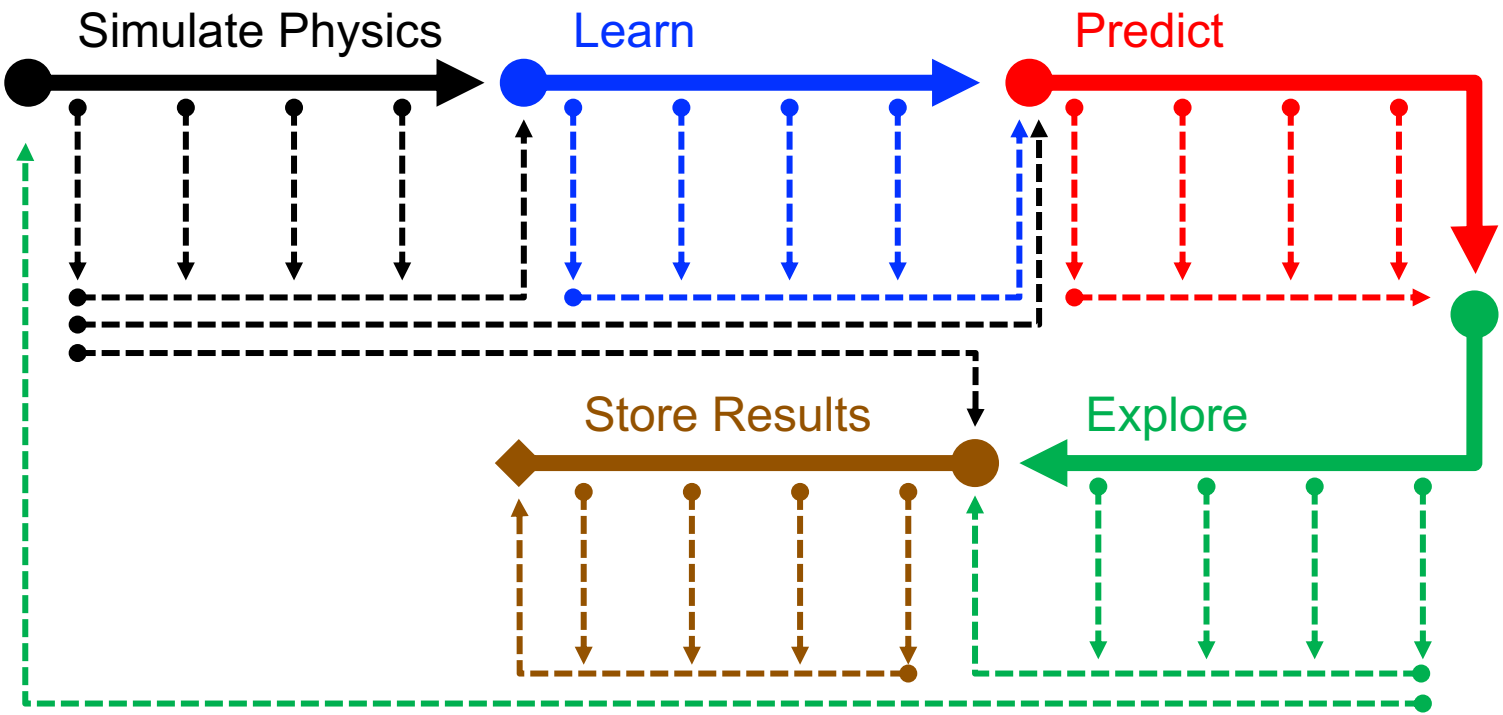
Robinson, Semler *et al.*

Speculative sampling isn't a linear workflow



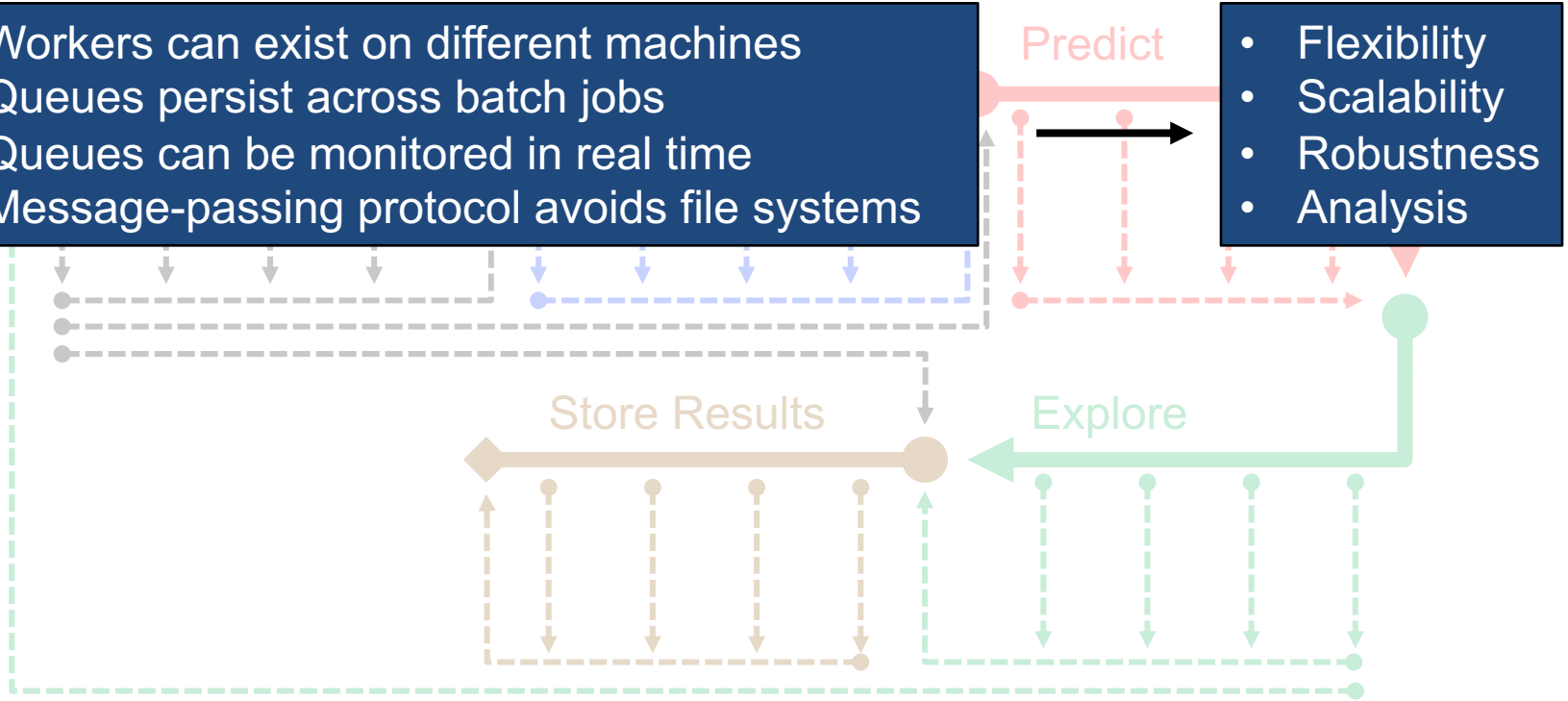
Robinson, Semler *et al.*

We've built a novel 'assembly line' workflow to provide speculative sampling with the required flexibility



We've built a novel 'assembly line' workflow to provide speculative sampling with the required flexibility

- Workers can exist on different machines
- Queues persist across batch jobs
- Queues can be monitored in real time
- Message-passing protocol avoids file systems

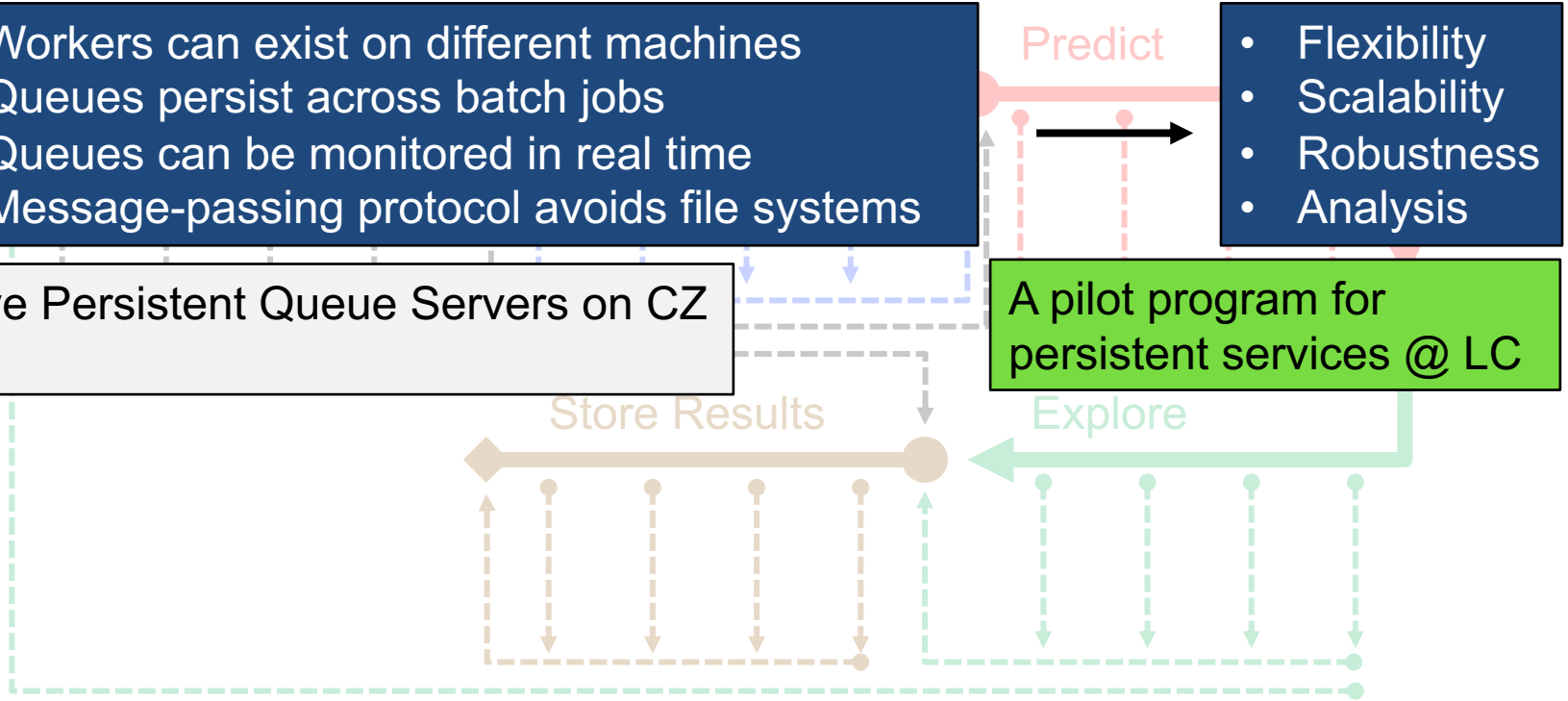


- Flexibility
- Scalability
- Robustness
- Analysis

We've built a novel 'assembly line' workflow to provide speculative sampling with the required flexibility

- Workers can exist on different machines
- Queues persist across batch jobs
- Queues can be monitored in real time
- Message-passing protocol avoids file systems

Live Persistent Queue Servers on CZ



- Flexibility
- Scalability
- Robustness
- Analysis

A pilot program for persistent services @ LC

We've built a novel 'assembly line' workflow to provide speculative sampling with the required flexibility

- Workers can exist on different machines
- Queues persist across batch jobs
- Queues can be monitored in real time
- Message-passing protocol avoids file systems

Predict

- Flexibility
- Scalability
- Robustness
- Analysis

Live Persistent Queue Servers on CZ

A pilot program for persistent services @ LC

Store Results

Message Broker
→ *RabbitMQ*

Communication meets LC security requirements

We've built a novel 'assembly line' workflow to provide speculative sampling with the required flexibility

- Workers can exist on different machines
- Queues persist across batch jobs
- Queues can be monitored in real time
- Message-passing protocol avoids file systems

Predict

- Flexibility
- Scalability
- Robustness
- Analysis

Live Persistent Queue Servers on CZ

A pilot program for persistent services @ LC

Store Results

Message Broker
→ *RabbitMQ*

Communication meets LC security requirements

Python Worker + Tasking API
→ *celery*

Widely-used open source software (e.g. Instagram)

We're aiming to make a "splash" for early Sierra access

- Generate 1 billion semi-analytic ICF implosion simulations
  - Oversample a high-dimensional space
  - Train a deep learning model on a physics problem
  - Develop new physics insight
- Shareable data for the machine learning community
  - Beyond MNIST and ImageNET
  - Several billion images, plus scalars, time histories
- Challenging and meaningful problems unique to the Laboratory

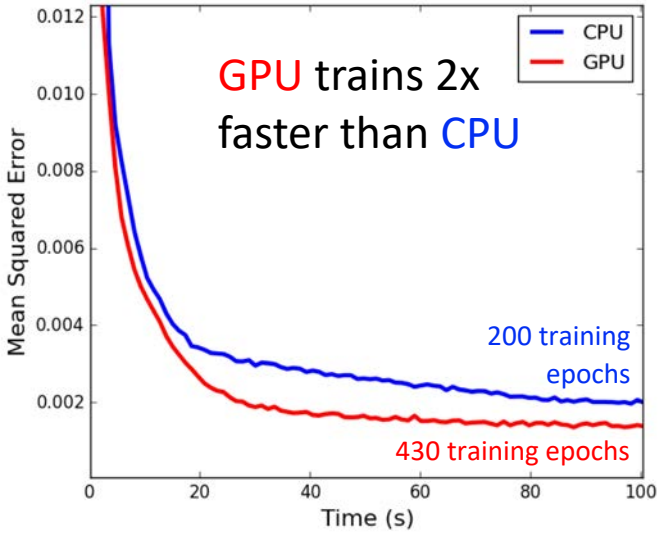
IMAGENET → Lawrence Livermore National Laboratory

An exciting opportunity to establish Lab leadership in cognitive computing and data science

Advanced machine learning workflows must navigate new advances in computer architectures

- CPU
  - Good branching control
  - Parallelism across large networks
- General purpose GPU
  - highly parallel
  - high memory bandwidth
- Inference accelerators
  - 16-bit precision
  - Low power consumption
- Next-gen machines, like Sierra, have all 3 … today!

Choose your processor wisely: DJINN in TensorFlow on CPU vs GPU



What commerce wants from a next-generation computer may not match what science wants

Google

amazon

TESLA MOTORS

Unilever

JAGUAR

Tens of billions of dollars

Infrequent, high-cost training
Frequent, low-cost evaluation

Lawrence Livermore National Laboratory

PPPL

PSFC
MIT Plasma Science & Fusion Center

UR LLE

OXFORD CENTRE FOR HIGH ENERGY DENSITY SCIENCE

Frequent, low-cost training
Less-frequent evaluation?

We must engage in co-design with vendor partners to develop Lab-appropriate machines

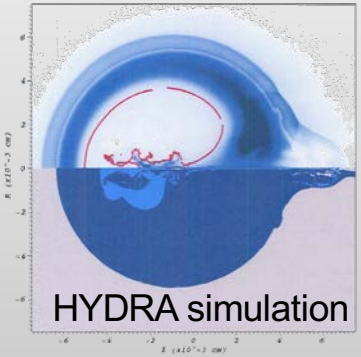
In most cases, we could substitute your application for ours

Problems only need a few essential ingredients

1. A reliable simulation code (you work at LLNL, after all)
2. A deep learning architecture, appropriately tuned
3. An advanced workflow code (you can have ours!)
4. Experiments (you'll have to get your own)
5. An interdisciplinary team with sensibilities in data science (see Data Science Institute)

We are advancing the way the Lab develops its predictive models across missions

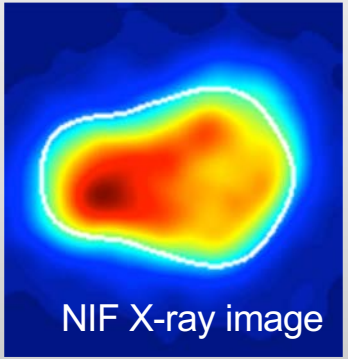
Traditional pillar
high-performance computing



New pillar
Machine learning to improve predictive science

- Deep learning to improve prediction
- Advanced workflows to support learning
- New architectures for modern computation

Traditional pillar
Large-scale experiments



These tools and techniques could work for you

Lawrence Livermore
National Laboratory