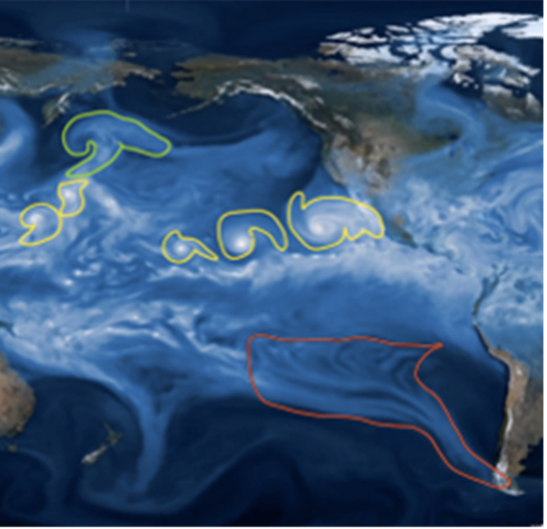
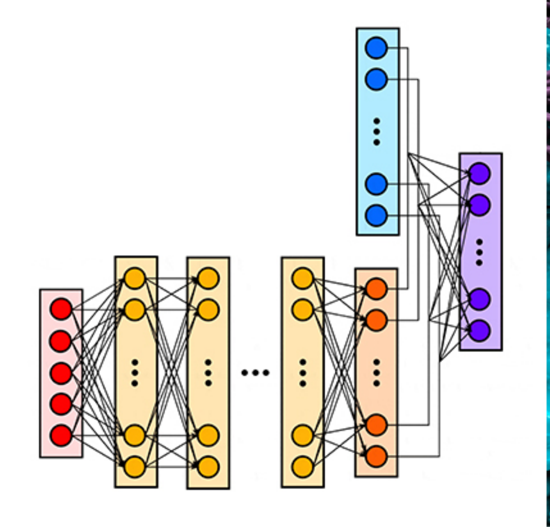
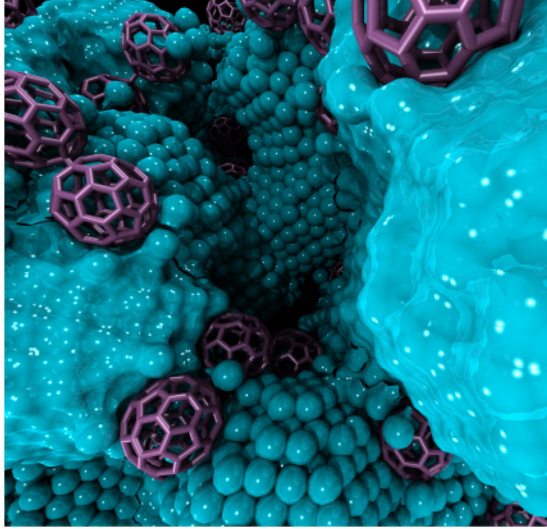
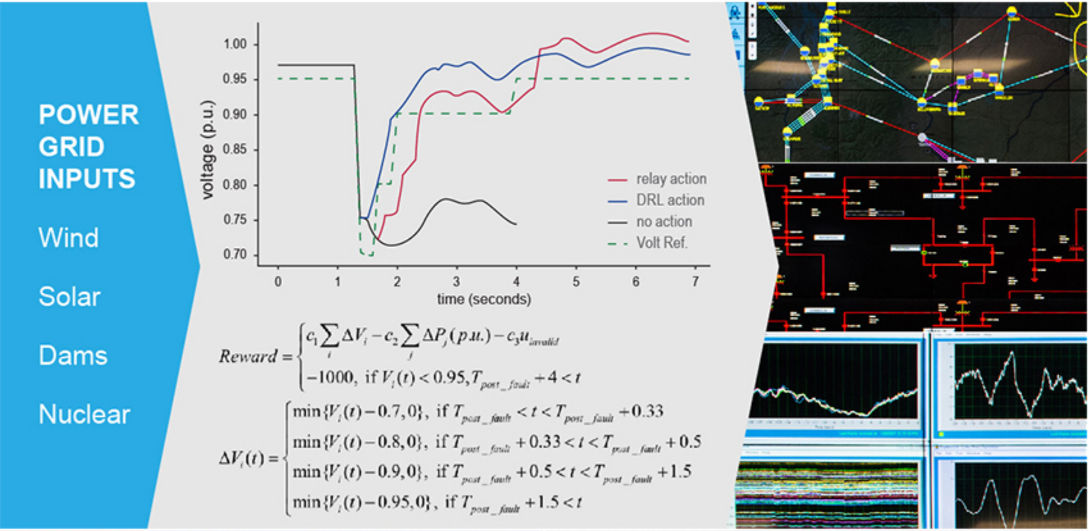


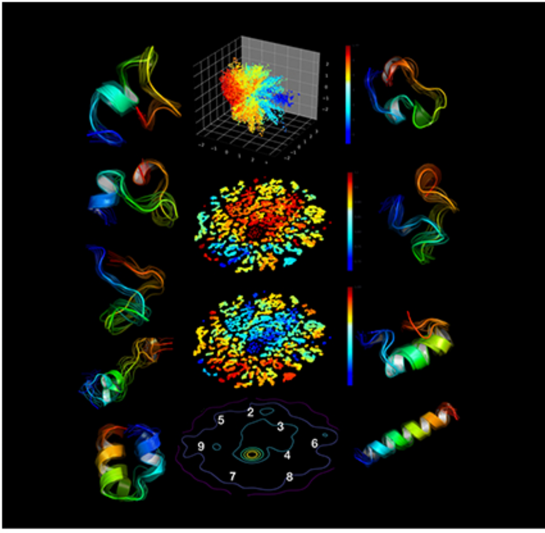


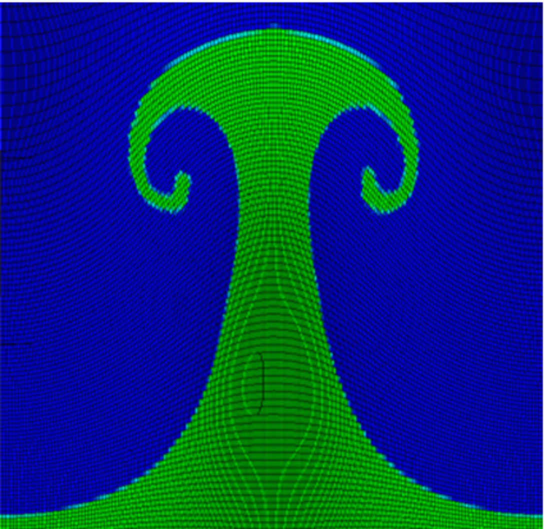


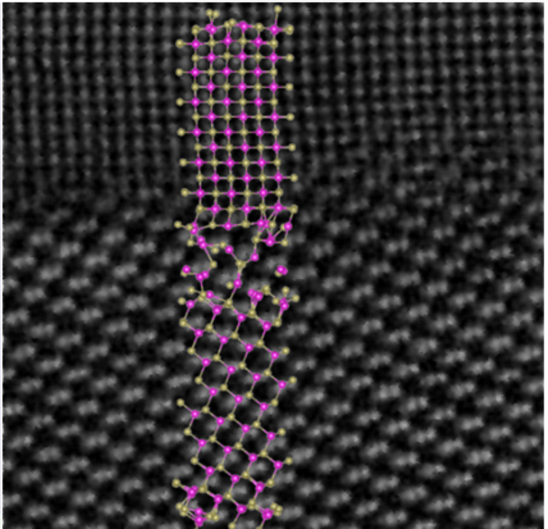
BASIC RESEARCH NEEDS FOR Scientific Machine Learning

Core Technologies for Artificial Intelligence









Prepared for U.S.
Department of Energy
Advanced Scientific
Computing Research

U.S. DEPARTMENT OF ENERGY

Cover image details, First row, left to right: detecting extreme weather events in climate datasets (Lawrence Berkeley National Laboratory), neural network for a turbulence closure model (Sandia National Laboratories), metal organic material design with machine learning (Argonne National Laboratory). Middle row, left to right: neural network for power grid control (Pacific Northwest National Laboratory), unsupervised machine learning for protein folding analysis (Oak Ridge National Laboratory). Last row, left to right: mesh relaxation with machine learning (Lawrence Livermore National Laboratory), machine learning analysis of electron microscopy materials images (Argonne National Laboratory).

Basic Research Needs Workshop for

Scientific Machine Learning

Core Technologies for Artificial Intelligence

Prepared for Department of Energy Advanced Scientific Computing Research

February 10, 2019*

*DOI: 10.2172/1478744

Contributors

Chair	Nathan Baker	Pacific Northwest National Laboratory
DOE ASCR Applied Mathematics lead	Steven Lee	Advanced Scientific Computing Research
Organizing committee and breakout leads	Frank Alexander	Brookhaven National Laboratory
	Timo Bremer	Lawrence Livermore National Laboratory
	Aric Hagberg	Los Alamos National Laboratory
	Yannis Kevrekidis	Johns Hopkins University
	Habib Najm	Sandia National Laboratories
	Manish Parashar	Rutgers University
	Abani Patra	State University of New York Buffalo
	James Sethian	University of California Berkeley
	Stefan Wild	Argonne National Laboratory
	Karen Willcox	University of Texas at Austin
Other contributors	Courtney Corley	Pacific Northwest National Laboratory
	Nathan Hilliard	Pacific Northwest National Laboratory
	Jeff Hittinger	Lawrence Livermore National Laboratory
	Ana Kupresanin	Lawrence Livermore National Laboratory
	Cosmin Petra	Lawrence Livermore National Laboratory
	Prabhat	Lawrence Berkeley National Laboratory
	Laura Pullum	Oak Ridge National Laboratory
	Laura Swiler	Sandia National Laboratories
	Chris Symons	Oak Ridge National Laboratory
	Xun Huan	Sandia National Laboratories
	Alex Tartakovsky	Pacific Northwest National Laboratory
	Stephen Young	Pacific Northwest National Laboratory
Workshop support	Jody Crisp	Oak Ridge Associated Universities
	Bruce Warford	Oak Ridge Associated Universities
	Julie Webber	Oak Ridge Associated Universities

Contents

List of Figures

List of Tables

List of Examples

Executive Summary

Introduction and Motivation

Scientific machine learning (SciML) is a core component of artificial intelligence (AI) and a computational technology that can be trained, with scientific data, to augment or automate human skills. Across the Department of Energy (DOE), scientific machine learning (SciML) has the potential to transform science and energy research. Breakthroughs and major progress will be enabled by harnessing DOE investments in massive data from scientific user facilities, software for predictive models and algorithms, high-performance computing platforms, and the national workforce. The crosscutting nature of machine learning and artificial intelligence provides a strong incentive for formulating a prioritized research agenda to maximize the capabilities and scientific benefits for the DOE.

The report summarizes the outcomes of a January 2018 basic research needs (BRN) workshop, which identified six Priority Research Directions (PRDs) for Advanced Scientific Computing Research (ASCR) in developing greater SciML-based capabilities for DOE mission challenges. The workshop considered the status, recent trends, and broad use of SciML. This information then was used to examine the opportunities, barriers, and potential for high scientific impact through fundamental advances in its mathematical, statistical, and computational research foundations.

SciML and AI will have broad use and transformative effects across the DOE. The first three PRDs describe foundational research themes that are common to the development of all SciML methods and corresponds to the need for domain-awareness, interpretability, and robustness. The other three PRDs describe capability research themes and correspond to the three major use cases for massive scientific data analysis, machine learning-enhanced modeling and simulation, and intelligent automation and decision-support for complex systems. This section describes each of the PRDs in more detail.

Priority Research Directions (PRDs)

The SciML BRN workshop identified six PRDs that build on the strengths of the ASCR Applied Mathematics program and address the challenges herein. Figure 1 (page vii) illustrates how PRDs on domain-aware, interpretable, and robust SciML support progress against the foundational research challenges. Similarly, Figure 2 (page vii) outlines a strategy to address the capability research challenges through PRDs on data-intensive SciML, machine learning-enhanced modeling and simulation, and intelligent automation and decision support.

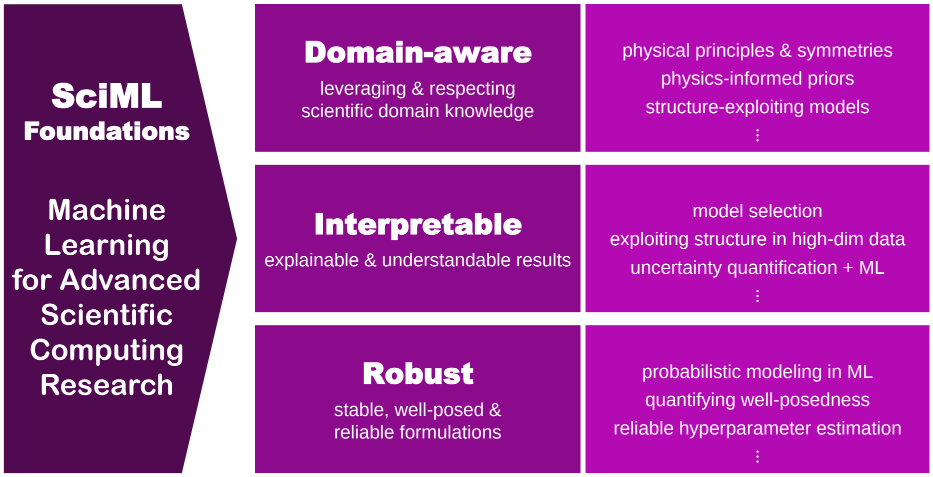

SciML
Foundations
Machine Learning for Advanced Scientific Computing Research
Domain-aware
leveraging & respecting scientific domain knowledge
physical principles & symmetries
physics-informed priors
structure-exploiting models
⋮
Interpretable
explainable & understandable results
model selection
exploiting structure in high-dim data
uncertainty quantification + ML
⋮
Robust
stable, well-posed & reliable formulations
probabilistic modeling in ML
quantifying well-posedness
reliable hyperparameter estimation
⋮

Figure 1: Foundational research themes of SciML must tackle the challenges of creating domain-aware, interpretable, and robust ML formulations, methods, and algorithms.

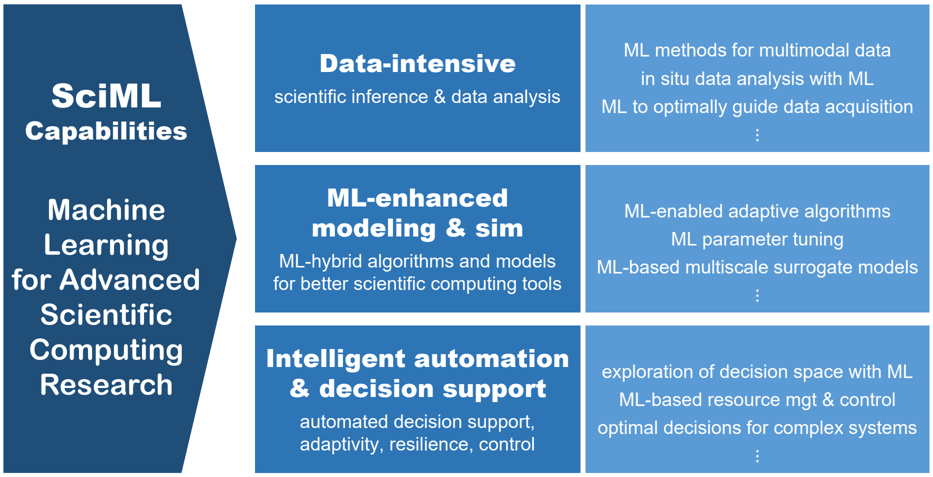

SciML
Capabilities
Machine Learning for Advanced Scientific Computing Research
Data-intensive
scientific inference & data analysis
ML methods for multimodal data
in situ data analysis with ML
ML to optimally guide data acquisition
⋮
ML-enhanced modeling & sim
ML-hybrid algorithms and models for better scientific computing tools
ML-enabled adaptive algorithms
ML parameter tuning
ML-based multiscale surrogate models
⋮
Intelligent automation & decision support
automated decision support, adaptivity, resilience, control
exploration of decision space with ML
ML-based resource mgt & control
optimal decisions for complex systems
⋮

Figure 2: Opportunities for SciML impact arise in scientific inference and data analysis; in ML-enhanced modeling and simulation; in intelligent automation and decision support; and in related applications.

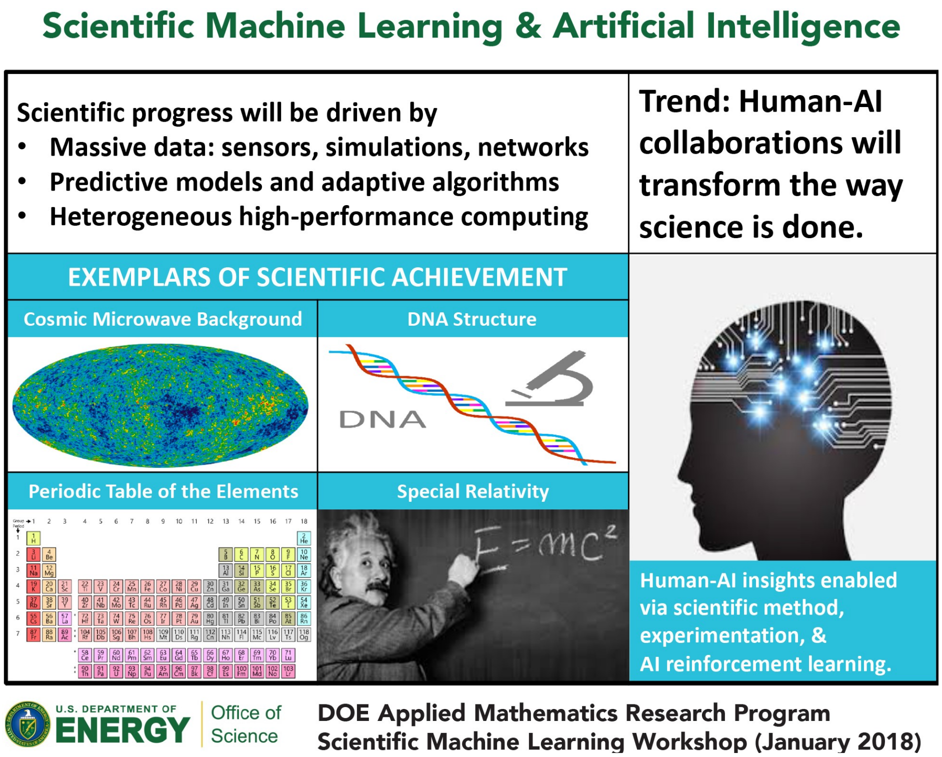
PRD 1. Domain-Aware Scientific Machine Learning. SciML methods are unlikely to ever replace established domain models based on physical mechanisms and scientific knowledge; however, there is a significant opportunity for SciML to *complement* traditional domain models. Domain knowledge includes physical principles, symmetries, constraints, expert feedback, computational simulations, uncertainties, etc. The focus of this PRD is to integrate such domain knowledge with SciML methods. Such integration is expected to improve accuracy, interpretability, and defensibility of SciML models while simultaneously reducing data requirements and accelerating SciML model training. Progress in this PRD will require new mathematical methods to learn improved model features that are constrained by domain knowledge, including fusion of multimodal and heterogeneous data sources to extract features.

PRD 2. Interpretable Scientific Machine Learning. Traditionally, physical understanding has been the bedrock of modeling. A user's confidence in a model's predictions is directly linked to the conviction that the model accounts for her/his domain knowledge; e.g., the right variables, parameters, and physical laws. In general, a tension exists between the need for increased complexity in ML models to improve results and the need for users to interpret the models and derive new insights and conclusions. This challenge has been widely recognized. However, SciML applications have unique challenges and opportunities to use existing domain knowledge to increase ML model interpretability. Progress in this PRD will require developing new exploration and visualization approaches to interpret complex models using domain knowledge, as well as new metrics to quantify model differences.

PRD 3. Robust Scientific Machine Learning. To take its place as a scientific methodology and be accepted for common use in domain sciences and high-regret applications, SciML methods must be robust and reliable. While ML methods are much used, the integration of protocols for verification, validation, and reproducibility are in their infancy. The credibility of research based on SciML requires that outcomes come from a process that is not sensitive to perturbations in training data, modeling choice, and/or computational errors. Progress in this PRD will require research for showing that SciML methods and implementations are well-posed, stable, and robust.

PRD 4. Data-Intensive Scientific Machine Learning. SciML in large-scale complex models and data faces a range of challenges, including high-dimensional, noisy, and uncertain input data, as well as limited information about model validity. Incorporating statistics, uncertainty quantification, and probabilistic modeling into SciML will provide a framework for managing some of these challenges. In particular, these approaches can address ill-conditioning, non-uniqueness, and over-fitting and allow for requisite uncertainty quantification in ML predictions. In addition, statistical and probabilistic methods can help uncover structure in data to improve scientific insight. At the same time, applying these methods in SciML is challenged by large data volume and complexity, as well as the high-dimensional structure of probabilistic SciML models. Progress in this PRD requires developing improved methods for statistical learning in high-dimensional SciML systems with noisy and complex data, for identifying structure in complex high-dimensional data, and for efficient sampling in high-dimensional parametric and model spaces.

PRD 5. Machine Learning-Enhanced Modeling and Simulation. DOE simulation codes model complex physical phenomena, often with dramatic variations in scale and behavior even within a single simulation. For performance, robustness, and fidelity, human expertise typically is integral in the simulation process to obtain quality solutions. The growing trend is for the models,



Scientific Machine Learning & Artificial Intelligence

Scientific progress will be driven by
- Massive data: sensors, simulations, networks
- Predictive models and adaptive algorithms
- Heterogeneous high-performance computing

Trend: Human-AI collaborations will transform the way science is done.

EXEMPLARS OF SCIENTIFIC ACHIEVEMENT

Cosmic Microwave Background

DNA Structure

Periodic Table of the Elements

Special Relativity

Human-AI insights enabled via scientific method, experimentation, & AI reinforcement learning.

U.S. DEPARTMENT OF ENERGY | Office of Science

DOE Applied Mathematics Research Program
Scientific Machine Learning Workshop (January 2018)

Figure 3: The ASCR Applied Mathematics program has laid the groundwork to harness machine learning and artificial intelligence for scientific purposes.

discretizations, and numerical solvers at the heart of DOE application codes to be more adaptive, usually through the use of simple theoretical controls and/or heuristics. There still are tremendous gains to be realized through the judicious use of SciML algorithms to better adapt aspects of the numerical models and their interactions with increasingly complex computer hardware. Similarly, traditional numerical algorithms are at the core of SciML algorithms, so SciML can be made more efficient, robust, and scalable by leveraging the extensive knowledge of the DOE scientific computing community. Catalyzing the interaction and interplay of scientific computing and ML algorithms has the potential to improve the throughput of both, but progress in this PRD will require developing new methods to quantify trade-offs and optimally manage the interplay between traditional and ML models and implementations.

PRD 6. Intelligent Automation and Decision Support. Applications that iterate around a forward simulation (e.g., in optimization, uncertainty quantification, inverse problems, data assimilation, and control) constitute a significant target for many of DOE's simulation and modeling capabilities, in many cases in support of decisions. Several major challenges loom at the frontier of simulation-based decisions in science and engineering: how to make the task of evaluating a complex, expensive simulation model over a high-dimensional parameter space tractable; how to best combine experimental and simulation data to inform decisions; how to validate the resulting evaluations and translate their uncertainty into quantifiable confidence for a decision-maker; and how to manage the interplay between automation and human decision-making. These challenges are particularly acute for SciML (as described in the preceding PRDs). In addition to progress in the previous PRDs, this PRD requires new mathematically and scientifically justified methods to

Table 1: DOE research initiatives and ASCR Applied Mathematics Funding Opportunity Announcements since 2005 that underpin each of the PRDs for SciML capabilities.

PRD 4. Data-Intensive Scientific Machine Learning
2009–2012: Mathematics for Analysis of Petascale Data 2010–2013: Resilient Extreme-Scale Solvers 2013–2016: DOE Data-Centric Science at Scale
PRD 5. Machine Learning-Enhanced Modeling and Simulation
2005–2008: Multiscale Mathematics Research and Education 2008–2012: Multiscale Mathematics for Complex Systems 2013–2016: Uncertainty Quantification for Extreme-Scale Science
PRD 6. Intelligent Automation and Decision Support
2009–2012: Mathematics for Complex, Distributed, Interconnected Systems 2010–2013: Uncertainty Quantification for Complex Systems 2012–2017: Mathematical Multifaceted Integrated Capability Centers I 2017–2022: Mathematical Multifaceted Integrated Capability Centers II

guide data acquisition and assure data quality and adequacy, improved SciML methods for multimodal data encountered in scientific applications, and new methods to quantify trade-offs and optimally manage resources used in decision support and related tasks.

Scientific Machine Learning and Artificial Intelligence

The six SciML PRDs provide a sound basis for implementing a coherent, long-term research and development strategy. Fortunately, over the last decade and more, DOE investments in applied mathematics and scientific computing have laid the groundwork for the type of basic research that will underpin key advances in the PRDs. For example, since 2005, the Applied Mathematics program has issued Funding Opportunity Announcements (FOAs) for university and DOE national lab projects that correspond to the SciML capability research themes as summarized in Table 1.

Scientific machine learning and artificial intelligence are rapidly developing areas of research. Recent reports have emphasized the need for developing a national strategy and preparing for the future [1, 2]. A previous report on a national AI strategy [3] will updated in 2019 to reflect the most recent developments and plans across federal agencies. This workshop report has considered these important challenges through the lens of applied mathematics and scientific computing research. In doing so, the six PRDs will be a useful guide in formulating an effective research agenda to maximize the capabilities and scientific benefits for the DOE.

Chapter 1

Introduction

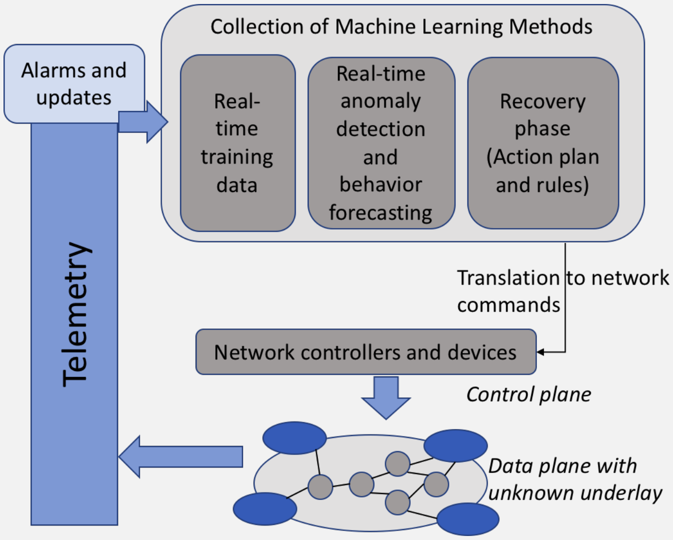
1.1 Scope

Recently, interest in ML-based approaches for science and engineering has increased rapidly. This growing enthusiasm for SciML stems from the combined development and use of efficient data analysis algorithms; massive amounts of data available from scientific instruments, scientific computations, and other sources; advances in high-performance computing; and the successes reported by industry, academia, and research communities. A conventional notion of ML involves training an algorithm to automatically find patterns, signals, or structure that may be hidden within massive data sets whose exact nature is unknown and therefore cannot be programmed explicitly. The algorithm's predictive capability is a learned skill. We seek to improve upon and harness the analytical and predictive power of ML to maximize its impact on DOE mission and science/engineering applications (for an overview of DOE missions, refer to recent reports [4, 5]). The SciML 2018 BRN Workshop attracted more than 120 experts in applied mathematics, computing, and data sciences from national laboratories, academia, industry, and federal agencies.

Workshop participants identified opportunities and gaps, thereby defining PRDs for applied mathematics in SciML. In particular, discussions centered on challenges and opportunities for increasing the *scale, rigor, robustness, and reliability* of SciML necessary for routine use in DOE science and engineering applications. Such developments also will greatly enhance the capabilities of many ML methods in common use. For example, some challenges may include rigorous analysis methods for developing and testing SciML algorithms and understanding SciML method approximation power, as well as bounding data and compute complexity of SciML approaches. These challenges were discussed in the context of existing methods with demonstrated use in scientific applications. The workshop also engaged the attention of the world-leading ASCR mathematics and computer science communities to establish the new research priorities presented as part of this report. The workshop principally focused on the mathematical challenges in SciML theory and application. These challenges were discussed in the context of existing methods with demonstrated use in scientific applications.

This report summarizes the outcomes of the SciML workshop and identifies current strengths and weaknesses of SciML for scientific discovery to inform the community and DOE about PRDs for SciML.

Example 1: Optimizing Network Traffic with Machine Learning


Collection of Machine Learning Methods
Alarms and updates
Real-time training data
Real-time anomaly detection and behavior forecasting
Recovery phase (Action plan and rules)
Telemetry
Translation to network commands
Network controllers and devices
Control plane
Data plane with unknown underlay

Exascale and increasingly complex science applications are exponentially raising demands from underlying DOE networks, such as traffic management, operation scale, and reliability constraints. Networks are the backbone to complex science workflows, ensuring data are delivered securely and on time for important computations to happen. To optimize these distributed workflows, networks are required to understand end-to-end performance needs in advance and be faster, efficient, and more proactive, anticipating bottlenecks before they happen. However, to manage multiple network paths intelligently, various tasks, such as pre-computation and prediction, must be done in near real time. ML provides a collection of algorithms that can add autonomy and assist in decision making to support key facility goals without increased device costs and inefficiency. In particular, by focusing on time-series and real-time statistics at all network levels (such as logs, traffic, security, usage, and more), ML can be used to predict potential anomalies in current traffic patterns and raise alerts before network faults develop. Unsupervised feature extraction methods can be used to tag samples from telemetry data, raising alarms and projecting behaviors of "normal" versus "abnormal" network scenarios. At this point, given learned traffic patterns and engineering decisions, certain network modules can be automated to induce correctional actions to rectify "bad" behavior, ensuring mission critical tasks are maintained. The latter part uses ML with decision support systems to offload some tasks, freeing engineers to focus on optimizing operations so science applications run as reliably as possible. Image credit: *Mariam Kiran, Energy Sciences Network (ESnet), and Lawrence Berkeley National Laboratory (LBNL).*

1.2 Definitions

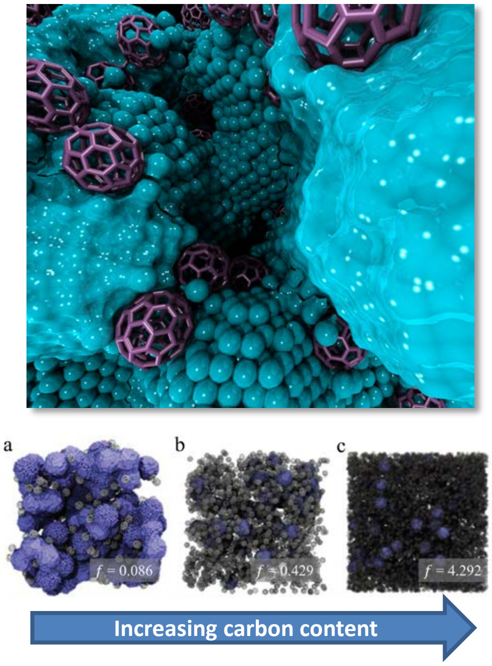
Perhaps the simplest definition of ML was provided in an apocryphal quote by Arthur Samuel, often attributed to his 1959 paper on ML for the game of checkers [6]: "Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." An equally laconic definition[*] was adopted in a recent report of the Royal Society [7].

This report employs a more detailed definition similar to that offered in a review paper by Jordan et al. [8]: "learning" is the process of transforming information into expertise or knowledge; "machine learning" is automated learning. The input information to ML includes training data and *a priori* knowledge or information (such as physical conservation principles). The output information from ML is "knowledge" or "expertise" used to make predictions or perform some activity. The *ML process* assumes the form of an algorithm that takes information as input and produces knowledge as output (as already described).

The literature often characterizes ML according to its different tasks, such as the canonical problems called out in [7] of classification, regression, clustering, dimension reduction, semi-supervised learning, and reinforcement learning. We recognize that SciML poses a rich set of

[*] "Machine Learning: A set of rules that allows systems to learn directly from examples, data and experience" – Royal Society [7].

Example 2: Designing Metal-Organic Hybrid Material Properties with Machine Learning



Images of three carbon-cobalt (C/Co) nanostructures with C/Co ratios of 0.086, 0.429 and 4.292 (a–c).

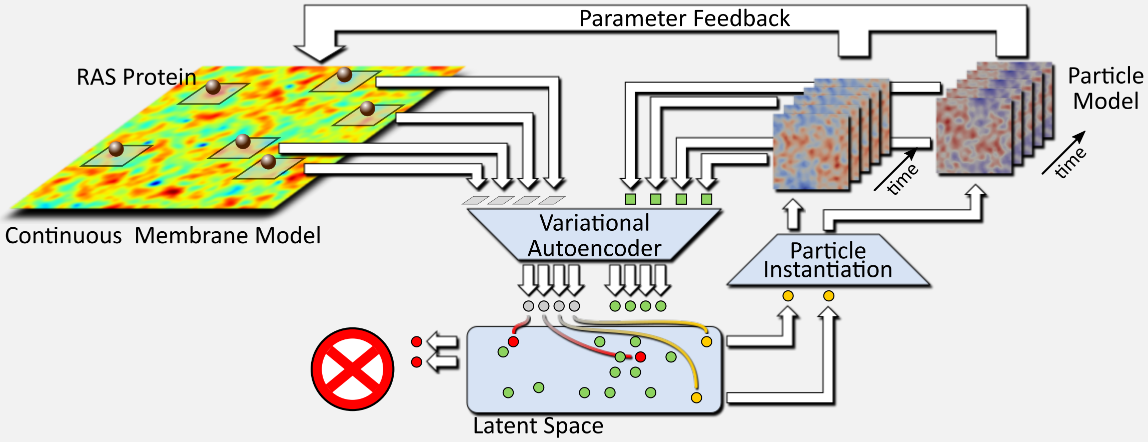
A single bond order potential trained using an ML approach predicts properties of metal-organic hybrid materials, accounting for complex atomic interfacial interactions ranging from highly metallic to highly organic. The new model is computationally cheap yet makes accurate predictions of hybrid material properties, reaching new possibilities in advanced design. The method uses supervised ML to train the model against actual values for structure, mechanical properties, and energies derived from quantum calculations for pure cobalt, pure carbon, and cobalt carbide in various compositions. Density functional theory calculations were used to generate the training set of material properties; genetic algorithms were used to optimize parameters. This ML approach has been applied to a range of systems, from clusters [12], oxides [13, 14], and nitrides [15] to two-dimensional materials [16]. Image adapted from Narayanan et al. [17].

challenges that span a broad set of potential use cases—and, in contrast to many existing applications of ML, a deeper consideration of the structure of the problem at hand is critical. Indeed, simulation supports a tremendously broad range of activities within the DOE portfolio, some of which support scientific goals of discovery and understanding and others that underpin decisions through prediction, optimization, and uncertainty quantification (UQ) [9]. The nature of SciML's potential impact varies across these activities, just as the need for new SciML methodology varies. We draw an analogy with the past decades of advancements in optimization: these advancements have been driven by an explicit and clear realization of the diverse structure of different optimization problem classes (linear programs, integer programs, mixed-integer programs, partial differential equation (PDE)-constrained optimization, etc.) and the need for structure-exploiting techniques in each case [10, 11]. Similar investments are required to advance domain-specific, structure-exploiting SciML.

1.3 Motivation and Impact

Traditional methods of scientific discovery proceed sequentially from data collection to analysis to construction of theories that codify new insights. In recent years, using high-end computation for complex model evaluation and data-driven methods for model enhancement and optimization have greatly enhanced and accelerated progress down this path, creating what are sometimes labeled as "third" and "fourth" paradigms for scientific discovery and engineering innovation [18,

Example 3: Multiscale Machine Learning Workflows for Cancer Research



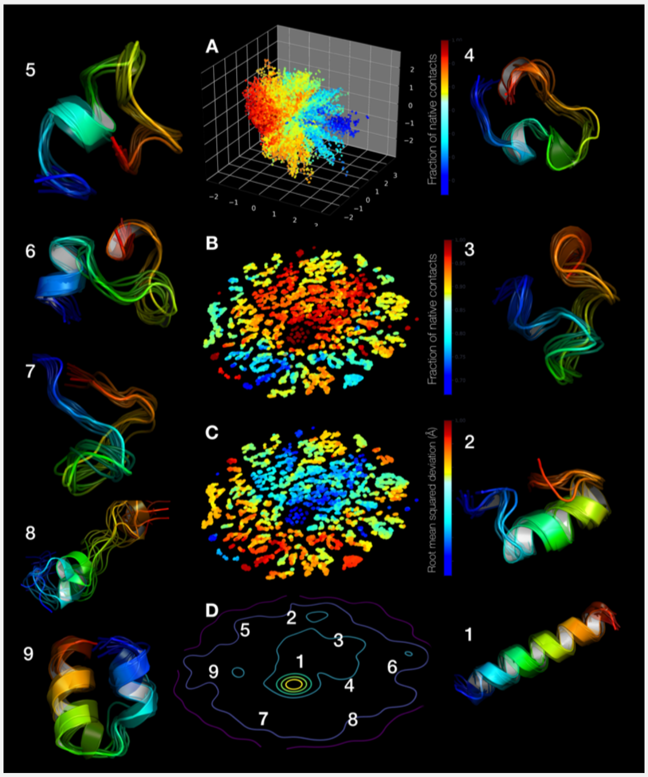
The RAS protein is known to be instrumental in the signaling pathways of many forms of cancer. To understand its interactions with the cell membrane, scientists require computational models because many characteristics of this interaction are not accessible experimentally. However, predictive modeling requires detailed, particle-based simulations only feasible for small portions of the membrane and at relatively short timescales. Unfortunately, simulating biologically relevant time and length scales is only possible with approximate continuum models. ML provides a means to combine the best of both aspects to ultimately create a simulation at relevant scales with the required accuracies. Given a continuum model (left) with single particles approximating the RAS protein, the system continuously examines small membrane patches underneath the RAS using a variational autoencoder to compare membrane configurations. Patches that differ from previous executions (yellow circles) are tagged and used to instantiate particle-based simulations. Patches deemed too similar to previous configurations are discarded. In this manner, more and more particle-based information is collected until the latent space of patch configurations is densely sampled. At this point, the system has produced equivalent particle-based simulations for each patch of the continuum model and, therefore, has created a simulation of biologically relevant time and length scales at particle-based precision. Image credit: *Timo Bremer and National Cancer Institute (NCI) Pilot 2 Team, Lawrence Livermore National Laboratory (LLNL).*

19].

Historically, the majority of scientific research effort has been on data collection; e.g., starting with the drive to create telescopes to observe planets and ending with large-scale, multi-physics simulations to gain a window into otherwise inaccessible processes, such as supernovae. However, while early analysis techniques sometimes were as simple as linear regressions, modern solutions for both experiments and simulations often are highly sophisticated combinations of mathematics, statistics, computer science, and a host of other disciplines. Even as the complexity of the analysis process has grown tremendously, it remains largely driven by human expectations and hypotheses. Guided by their deep subject matter expertise, extensive experience, and well-honed intuition, scientists will develop hypotheses and tailor analysis approaches to verify or disprove them.

However, this existing paradigm is rapidly being stretched to its limits because of dramatic growth in scientific data; e.g., collecting billions of particle collisions [20], simulating thousands of climate scenarios [21], or observing millions of stars [22]. No single researcher or even a large team can directly sift through such staggering amounts of data. Yet data reduction techniques—e.g., for finding "interesting" collisions or "unusual" stars—can fall victim to various confirmation biases, where rare events outside of the current theory may be overlooked or misinterpreted. As the size and complexity of data increase, the likelihood of missing opportunities for breakthroughs also

Example 4: Simplifying High-Dimensional Protein Folding with Machine Learning



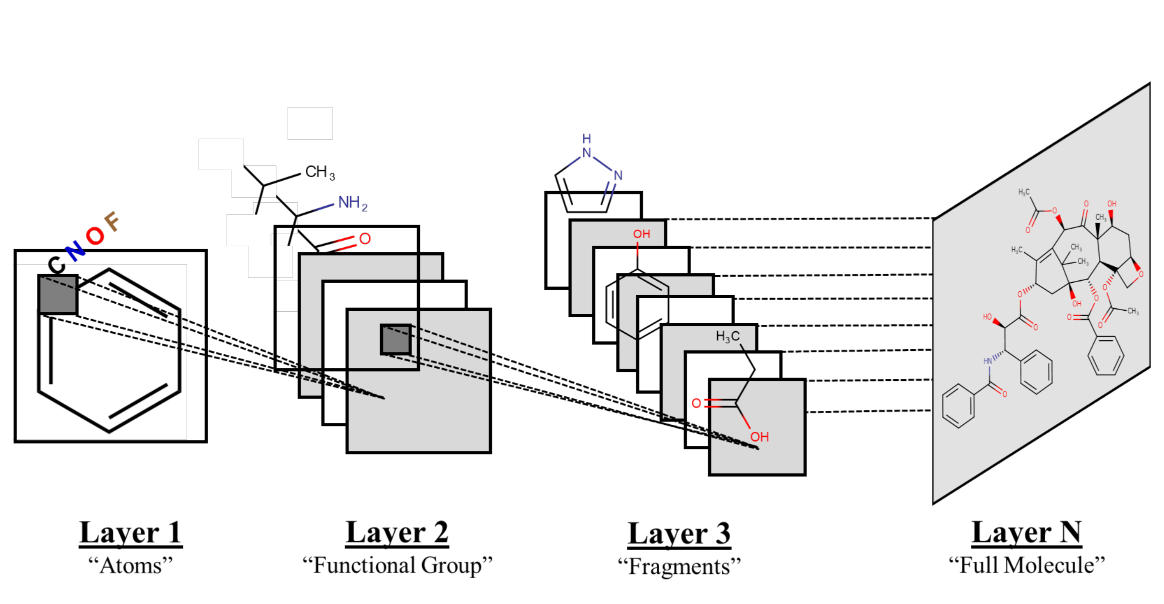
ML techniques reveal Fs-peptide folding events from long timescale molecular dynamics simulations. A low-dimensional embedding of the simulation events reveal transitions from fully unfolded states (blue) to fully folded states (red). In the picture, a two-dimensional embedding using *t*-test stochastic neighborhood embedding shows the presence of near-native states (labeled state 1) versus partially unfolded (2-7) and fully unfolded states (8-9). Developed as part of the Exascale Computing Project (ECP) cancer distributed learning environment (CANDLE) project. Image credit: *Arvind Ramanathan, Oak Ridge National Laboratory (ORNL).*

will escalate and may ultimately delay progress. To address this challenge, many areas of science are moving toward more data-driven techniques that ultimately aim to augment the need for prior assumptions and meticulously tailored hypotheses with massive data collections.

This data explosion is not limited to scientific applications. In fact, the digital revolution and its corresponding increase in easy data collection—from financial transactions to online purchases and social media interactions—have led to a revolution in ML. This revolution has been driven by the availability of large labeled data sets that are propelling the development of new ML methods. These methods allow computers to accomplish tasks once feasible only to humans; e.g., driving a car [23], playing checkers [6], or playing chess [24]. Especially in areas that require processing large data volumes (a task not well suited for human brains), these techniques have produced stunning results thought to be unachievable even a decade ago.

There are many potential benefits to DOE missions from engaging ML in computational science, including the examples herein. ML methods are particularly useful for discovering correlations in high-dimensional data and, thus, can be useful in analyzing computational results. ML methods also can construct surrogates for complex forward models; e.g., with neural network (NN), Gaussian process (GP), and related methods [25–27]. ML can also assist in dimension reduction for high-dimensional data; e.g., by learning/discovering low-dimensional manifolds underlying the data [28]. Such dimension-reduction methods can be employed for understanding the dynamical structure behind the data. ML-based dimension reduction also can help define effective distance measures between data sets, thereby providing paths toward effective likelihoods for complex model calibration and parameter estimation from observational data. Already, ML methods have been used in various science applications, including analyzing turbulent flow

Example 5: Predicting Chemical Properties with Machine Learning



Unlike traditional ML algorithms, deep neural networks (DNNs) can perform representation learning, which is the ability to automatically engineer relevant features for the task it is being trained to predict. In this illustration, convolutional neural network (CNN) models originally designed for computer vision applications can be adapted to analyze molecular drawings. In the training process, these CNN models construct hierarchical representations that can mirror the conceptual understanding of chemistry, which then can be used to construct even more sophisticated representations for predicting complex chemical properties that otherwise would be difficult to predict from first principles. Image credit: *Garrett Goh, ChemNet software, Pacific Northwest National Laboratory (PNNL).*

computations [29–36], subsurface flow modeling [37], solid mechanics modeling of diverse materials [38–46], geophysics [47–50], and combustion modeling [30].

Still, despite these successes, numerous challenges remain in the path toward routinely adopting ML in scientific applications. In many of the most successful ML examples, such as image recognition, system developers know the "ground truth" sufficiently well to check the results, often even while training the models. Almost by definition, the most interesting scientific applications of SciML are those, such as materials discovery or high-energy physics, where the answers are unknown beforehand or the results of an automated system are not easily verified. Instead, familiar questions from scientific computing are, for example:

- How reliably will a given algorithm work; e.g., for what type and quantity of data do we expect results?
- How robust is a certain solution; e.g., how might slightly different data or the addition of noise change the results?
- How rigorously have the assumptions and underlying theories been defined and validated? For classical techniques, such as analyzing PDE-based models, these questions lead to familiar concepts, including well-posedness, stability, numerical approximation, and UQ.

These questions have led to a long history of relevant research and to reliable and robust outcomes from PDE-based models used in DOE applications. On the other hand, equivalent concepts for SciML-based models are not well established, and the lack of precise definitions and clearly expressed assumptions often leads to the failure of ML-based methods. Finally, the ultimate goal of analysis is for a *scientist* to gain new insights, adding a human dimension to the scientific learning problem. This process requires both integrating the existing body of human knowledge into the SciML approach and providing help for users to understand how a given approach works.

Independent of these fundamental (and somewhat abstract) differences between ML as seen in the media and scientific data analysis, significant practical and technical distinctions also exist. The PRDs presented in this workshop report describe some of the most significant differences and resulting challenges. Nevertheless, even acknowledging these challenges, ML has the potential to significantly advance diverse scientific areas and will transform the way science is done. It is likely that before long, various experiments and simulations will no longer be primarily limited by what data they can collect but by how well they are able to extract insights from the data they have. However, to take full advantage of the combination of massive data collections and SciML for scientific discovery, we must understand the current state of the art, where it may not meet the demand of various scientific applications, and what the key open research directions are to address the shortcomings.

Chapter 2

Priority Research Directions in Scientific Machine Learning

2.1 Domain-Aware Scientific Machine Learning

Incorporating scientific domain knowledge in the ML process is a task unique to SciML. Awareness of domain knowledge can enhance domain-agnostic data in terms of accuracy, interpretability, and defensibility of SciML models. Furthermore, incorporating scientific domain knowledge has the potential to dramatically reduce data requirements, as well as to accelerate training and prediction.

Domain knowledge is found in many forms, such as physical principles, constraints [51], symmetries [52, 53], conservation laws, and other knowledge gained from theoretical or computational studies. Scientific domain knowledge can be expressed in many forms, including physical models (e.g., *ab initio* or first-principles physics), physical constraints (e.g., symmetries, invariances, conservation laws, asymptotic limits), computational simulations, uncertainties, correlations in space and time, and structural forms (e.g., discrete, graph-like, non-smooth data). For such domain knowledge, both theoretical foundations and computational infrastructure exist (e.g., solvers and simulations) that can benefit SciML.

Domain knowledge has been proven to help supervised and unsupervised ML, as well as in generating synthetic data (e.g., with constrained generative adversarial networks (GANs) [54]) and reinforcement learning [55, 56]. Although scientific data may satisfy (e.g., modulo various types of errors and noise) underlying laws of physics, directly leveraging such domain knowledge can allow the learning process to focus on modeling more challenging and computationally impractical phenomena with less labeled data. Domain knowledge can be incorporated for various objectives, including improved interpretability (Section 2.2) and robustness (Section 2.3), as well as in a multitude of ways; e.g., both ML-enhanced modeling-simulation (Section 2.5) as well as intelligent automation and decision support tasks (Section 2.6).

Thrust A: How Should Domain Knowledge Be Incorporated into Unsupervised Machine Learning and Model Feature Selection?

Features are important inputs to ML algorithms. In unsupervised ML, features are selected or constructed from raw input data to represent the important attributes of the data. The choice of features can dramatically affect the ML algorithm's algorithmic efficiency, the quality of the classifier or predictor output, and the model's interpretability or defensibility. Effective features for SciML are *representative*, capturing the relevant scientific information in data; *interpretable*, recognizable by human experts; and *generalizable*, producing the same results using different learning techniques. Ideal features will span the data efficiently and be connected to measurable physical quantities. However, the ultimate determination of whether or not features are effective is in the performance of the ML algorithm that uses them.

Research is needed regarding methods that incorporate domain knowledge into feature selection. Progress in this thrust will require new mathematical methods to learn improved model features that are constrained by domain knowledge, including fusion of multimodal and heterogeneous data sources to extract features.

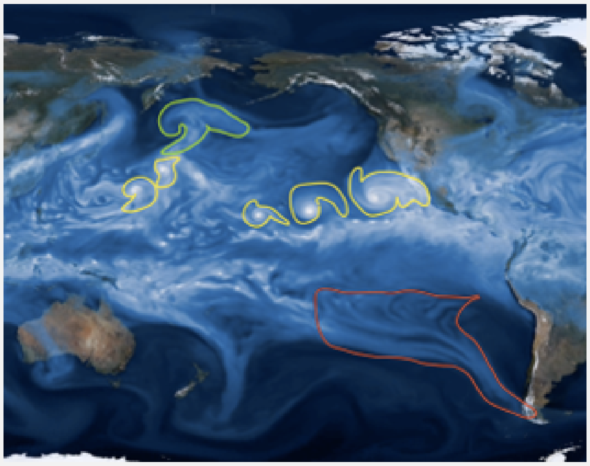
Thrust B: How Should Domain Knowledge Be Incorporated into Supervised Machine Learning?

The central question for this thrust is "which knowledge should be leveraged in SciML, and how should this knowledge be included?" Any answers will naturally depend on the SciML task and computational budgets, thus mirroring standard considerations in traditional scientific computing.

Hard Constraints. One research avenue involves incorporation of domain knowledge through imposition of constraints that cannot be violated. These hard constraints could be enforced during training, replacing what typically is an unconstrained optimization problem with a constrained one. In general, such constraints could involve simulations or highly nonlinear functions of the training parameters. Therefore, there is a need to identify particular cases when constraint qualification conditions can be ensured as these conditions are necessary regularity conditions for constrained optimization [57–59]. Although incorporating constraints during training generally makes maximal use of training data, there may be additional opportunities to employ constraints at the time of prediction (e.g., by projecting predictions onto the region induced by the constraints).

Soft Constraints. A similar avenue for incorporating domain knowledge involves modifying the objective function (soft constraints) used in training. It is understood that ML loss function selection should be guided by the task and data. Therefore, opportunities exist for developing loss functions that incorporate domain knowledge and analyzing the resulting impact on solvability and generalizability. Training objectives also increasingly contain several different components; e.g., regularization terms that can encapsulate soft constraints [60], and these components usually are composed in a weighted sum. Opportunities exist for analyzing the effect of such components; for example, research that draws on areas such as compressed sensing, Bayesian inference, and multi-objective optimization to incorporate domain knowledge into the components. A particular challenge is the need to incorporate uncertain or incomplete domain-specific knowledge, as well as multiple physics and data sources that have different time or space fidelity and/or are multiscale. These challenges are similar to those evident in the mathematical areas of reduced-

Example 6: Detecting Extreme Weather Patterns in Climate Data with Deep Learning



Characterizing the frequency and intensity of extreme weather patterns is an important research priority for the climate science community. To objectively extract patterns from large, complex climate data sets, the climate science community has predominantly used heuristics comprising multivariate threshold conditions [67]. Recent work has demonstrated that it is possible to use modern ML methods—such as deep learning (DL)—for classification and localization tasks [68, 69]. While these DL architectures provide high predictive accuracy, they are completely oblivious of the fact that the methods are being applied to a fluid flow system. The filter weight initialization and update schemes do not restrict the learned features to a subspace that would subscribe to conservation and physical consistency laws. Explicitly incorporating such constraints will further enhance the accuracy and, almost certainly, the interpretability of DL methods. Image credit: *Michael Wehner and Prabhat, LBNL.*

order modeling, multifidelity optimization, and UQ, but with an acute need for techniques (e.g., quantifiable error metrics and objectives) that facilitate the automation required in an ML pipeline.

Model Form. Another research avenue incorporates domain knowledge through the chosen form (e.g., basis employed) of the ML model. For example, incorporating algebraic invariances, such as symmetries and scaling in kernel approaches, have been shown to improve generalization performance [61–64]. Similarly, CNNs can incorporate knowledge in vision processing domains through convolutional filters that exploit locality, whereas the recurrent nature of recurrent neural networks (RNNs) can incorporate knowledge for speech recognition. An opportunity exists for principled selection of model forms across broad scientific domains, as well as for understanding the associated computational burden and effect on accuracy.

Thrust C: How Should Domain Knowledge Be Modeled and Represented in Scientific Machine Learning?

An additional opportunity for domain-aware SciML research is in constructing modeling languages and frameworks that facilitate the inclusion of domain knowledge into the training process. Often, modeling languages and frameworks (e.g., [65, 66]) are designed to lower the barrier of entry for users by facilitating rapid and robust problem formulation. Extending the ways that SciML can express and incorporate domain knowledge could have far-reaching implications in much the same way that these tools now are regularly used for implicit features, such as algorithmic differentiation.

2.2 Interpretable Scientific Machine Learning

One of the challenges in applying ML is the inherent complexity of many of its techniques. The canonical examples for ML complexity are DL-based approaches [70, 71]. DL promises unprecedented advances in dealing with a range of data types but relies on millions of degrees of freedom, connected in complex arrangements and trained through hand-tuned optimizations. In ML, significant expertise exists for selecting architectures, tuning optimization procedures, etc. However, current understanding is limited regarding how and/or why these techniques work and why they can be predictive. As a result, ML approaches may provide excellent classification performance while struggling to afford insight into the solution. Because novel insights are essential to science and engineering, the interpretability of ML methods must be improved.

Broadly speaking, the interpretability challenge in SciML touches all stages of the processing pipeline and all other PRDs. However, while the ultimate goals are similar, the focus of this PRD is on the humans "in-the-loop" with ML process. Furthermore, while more mature theories will diminish the need for interpretability in some areas, the frontier of ML research will depend on human understanding and intuition. This need is especially acute when considering the target audience to include ML experts, domain science practitioners, non-scientific stakeholders, and the public at large.

This PRD is closely coupled to the "domain-aware ML" PRD described in Section 2.1. General ML interpretability currently is the focus of several efforts, including the Defense Advanced Research Projects Agency (DARPA) eXplainable Artificial Intelligence (XAI) program [72]. However, unlike the problem of general ML interpretability, SciML interpretability has the advantage of rich domain (scientific) knowledge. Increased integration of domain knowledge into SciML methods (see Section 2.1) also offers the opportunity for improved interpretability of these methods.

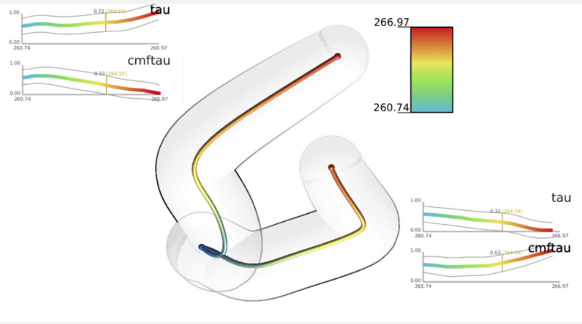
The fundamental challenges for interpretability can be thought of in terms of two dimensions: 1) data complexity and 2) target users. The former encapsulates the fact that in ML, one virtually always deals with high-dimensional data and parameter spaces, complex inputs and outputs, and highly sophisticated models. The latter dimension acknowledges the fact that depending on the intended audience, ideal techniques may differ widely according to the internal reference frame of the different user groups. To describe individual challenges and research directions in more detail, the thrusts presented in this section follow the typical steps in the ML processing pipeline.

Thrust A: How Should High-Dimensional Complex Data for Machine Learning Applications Be Explored?

Typically, one of the first steps in any data-processing pipeline is an initial exploration of data set structure. For example, questions regarding how smooth a given regression function should be necessitates some understanding of the geometry of both spaces, as well as the mapping between them. In general, understanding data characteristics has the potential to contribute significantly to the entire SciML pipeline—from understanding the distribution of the input data, to analyzing the model fitting through its path on the optimization landscape, to interpreting the output. In traditional scientific data analysis, this step often involves common visualization approaches; e.g., rendering slices through three-dimensional (3D) data, computing level sets, etc. Visualization provides a quick and intuitive way to understand data smoothness, the range of values, or if the data should be scaled. However, SciML data often are high-dimensional and/or complex, which complicates visualization. There is a need for methods that provide (human) users with SciML insights into data characteristics beyond traditional statistical indicators or other integrated measures.

Ideally, data exploration would be fulfilled by using well-defined, broadly accepted, and thoroughly tested indicators. Well-known examples of such indicators are statistical distribution parameters and the number of clusters, to name a few. Unfortunately, in many ML applications, these indicators may not exist (e.g., the underlying distributions may not be known), may not easily be defined (e.g., distributions of images), or may not lend themselves to a simple interpretation

Example 7: Exploring High-Dimensional Parameter Spaces with Machine Learning



Abstract representation of high-dimensional data can provide novel insights in an intuitive manner. The figure represents the Morse-Smale complex [73, 74] of the long-wave flux computed from a global climate model in a 21-dimensional parameter space [75]. Each of the two lines illustrates one connected region in parameter space where the flux varies monotonically between the minimum (shown in blue) and the two local maxima (shown in red). The lines form as the core lines of these regions are computed in high-dimensional space and are embedded into two dimensions using a common dimension reduction technique. The side plots show how one pair of input parameters (ordinate) varies with respect to the local flux (abscissa) in the corresponding region. Strong anti-correlation is evident in both regions yet in different directions. A global regression analysis (not shown) averages both behaviors and shows no discernible dependencies between the flux and these parameters. However, the high-dimensional plot clearly illustrates that strong dependencies exist, along with two different regions in the parameter space that lead to a high flux. Both parameters are related to cloud formation and an imbalance prevents clouds from forming, leading to a large flux. Image credit: *Samuel Gerber and Peer-Timo Bremer, LLNL.*

(e.g., higher-order Betti numbers). Research is needed to develop such quantitative measures or replace them with interpretable qualitative characteristics. These may include new types of descriptors to convey the geometry, topology, or general information content of high-dimensional data. This challenge is directly related to developing effective features and incorporating domain knowledge as discussed in Section 2.1.

Thrust B: How Should Scientific Machine Learning Models Be Explored and Understood (Model Introspection)?

Once an ML model has been selected, it is important to interpret the process by which it is fit/optimized to the data. That is, can the relationship between the model input, operation, and output be rationalized or explained? Such interpretability allows users to understand the model's results along with their robustness and sensitivity.

Traditionally, physical understanding has been the bedrock of modeling. A user's confidence in model predictions is linked directly to the conviction that the model accounts for the *right* variables (e.g., temperature, pressure, or material density), parameters (e.g., inflows or reaction rates), and physical laws (e.g., heat-mass balance, energy, or Arrhenius mass action kinetics). Very simple models are readily interpretable. However, once the problem dimensions extend beyond a few and the model's complexity increases slightly, then model understanding, particularly for stakeholders, becomes significantly more difficult. This problem is greatly exacerbated for nonlinear models. Models, such as DNNs, have sufficient nonlinearity and complexity to complicate routine interpretation. Although the performance of these complex models is impressive, their lack of interpretability makes them insufficient for high-regret and safety-critical systems [76]. For verification and for developing trust in the model, intuitive model introspection and interactive exploration of the solution space are vital for convincingly conveying results to stakeholders.

Given this objective of decomposing the model and decision process into interpretable human-meaningful and human-manageable steps, research is needed to provide a decision process decomposition for complex SciML models. In doing so, the human-meaningful steps may include connecting abstract representations to known laws (e.g., of physics) or interpretable concepts. This challenge is directly related to the domain-aware SciML PRD discussed in Section 2.1. Research is needed to provide model exploration and interpretation capabilities, as well as to enable the trade-off between model interpretability, flexibility, and accuracy for use in model selection.

Thrust C: How Should Differences between Scientific Machine Learning Models, Inputs, and Results Be Expressed?

SciML interpretability also requires the ability to describe differences between models, inputs, and results. The SciML workflow consists of gathering scientific data (that may be complex and high dimensional) and applying models (ranging from simple linear models to DL) to produce results (that can range from simple binary classification output to predicting dense spatio-temporal fields). This thrust area's scientific objective is to assist in the characterization of complex data sets, models, and output with the eventual goal of enhancing the interpretability of the overall SciML workflow. To compare and evaluate multiple SciML processes, it will be necessary to express differences between objects at each stage of the process.

Some of the required methods already exist for such comparisons. Techniques exist for input/output (I/O) data set comparisons, although additional research is required to ensure these methods can meet the scale and complexity of data found in SciML applications. A similar challenge exists in comparing ML models. Concepts of model complexity; e.g., Akaike information criterion (AIC) and Bayes information criterion (BIC), can be useful for simple models but do not scale to the depth or complexity of modern ML models. Therefore, research will be needed to develop new methods for model comparison, perhaps using dynamical systems or probabilistic frameworks.

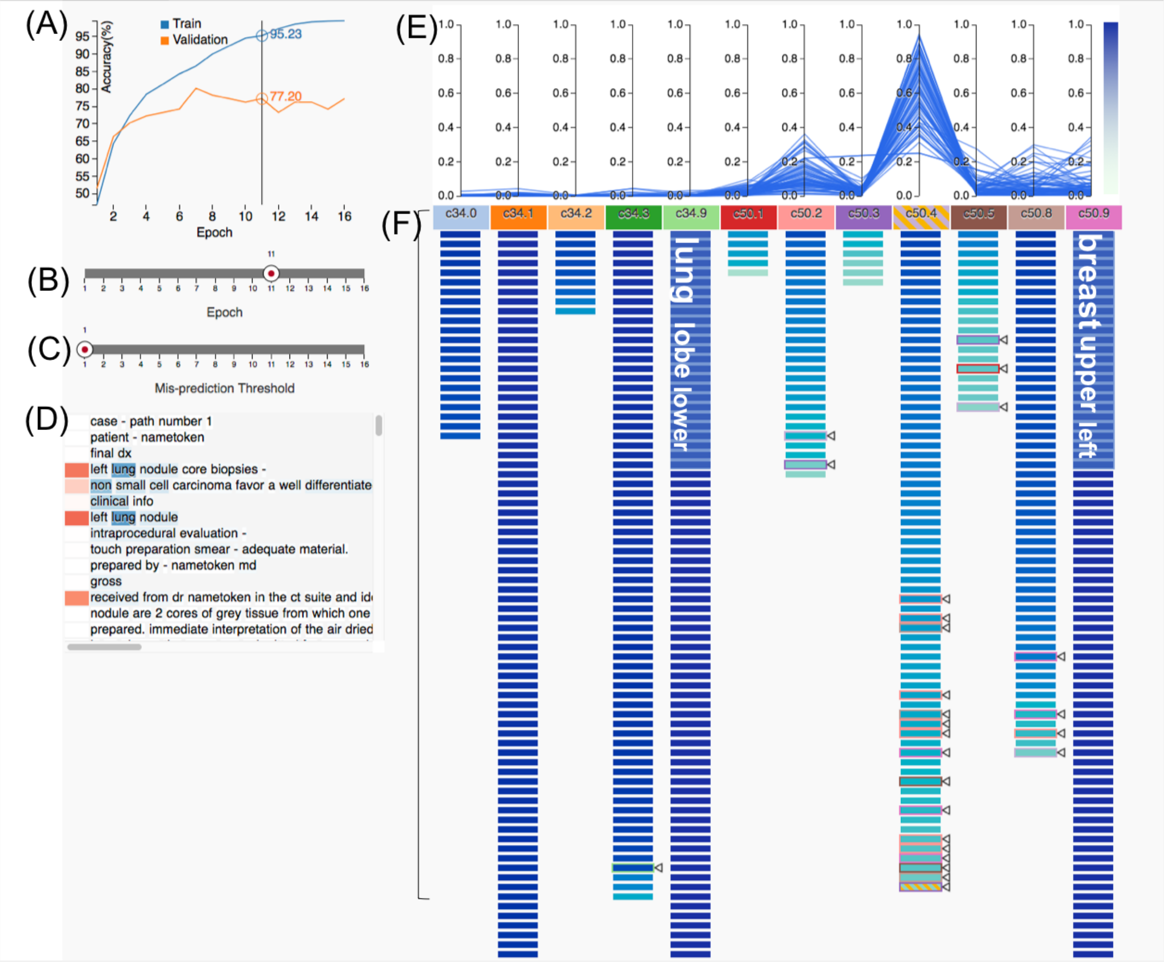
2.3 Robust Scientific Machine Learning

ML already has made a significant impact in a variety of high-regret applications, including clinical diagnoses [77, 78], security [79, 80], metal fatigue in aviation [81], and environmental contamination [82, 83]. An excellent review of some of these applications and associated risks is provided by Shneiderman [84] and references therein. Such applications exemplify the importance of robustness and rigor in ML to minimize the risks associated with its use. As discussed earlier (Section 2.2), these applications also point out the need for interpretability in SciML predictions [85]. The potential negative impacts of misused ML has led to calls for policies to "anticipate, monitor, and retrospectively review operations" for managing algorithms in high-impact applications [84] and to initiate research that explores ways to manage algorithm behavior [86, 87].

To gain acceptance as a legitimate scientific methodology, SciML must achieve the same level of scientific rigor expected of established methods deployed in science and applied mathematics. Basic requirements include validation and limits on inputs and context implicit in such validations, as well as verification of the basic algorithms to ensure they are capable of delivering known prototypical solutions exactly (cf., the use of manufactured solutions to test numerical algorithms [88]). In essence, these properties encapsulate a requirement for the scientific methodology to be reproducible and for the basic techniques to be well-posed and stable.

With its emphasis on well-defined analytic processes for stability and error analysis, applied mathematics can provide a mechanism for developing SciML methods with robustness. Such robustness will address issues related to sensitivity to training set size, choice of data in training and test sets, numerical instability in learning algorithms, scalability, and parallelization with complicated and heterogeneous hardware. While the need to investigate these issues has been expressed

Example 8: Visualizing Deep Neural Network Classification



case - path number 1
patient - nametoken
final dx
left lung nodule core biopsies -
non small cell carcinoma favor a well differentiate
clinical info
left lung nodule
intraprocedural evaluation -
touch preparation smear - adequate material.
prepared by - nametoken md
gross
received from dr nametoken in the ct suite and id
nodule are 2 cores of grey tissue from which one
prepared. immediate interpretation of the air dried

Components for visualizing the classification results during the iterative development pipeline of a DNN model include a training accuracy graph (A) that shows the accuracy changes of training and validation sets according to each epoch. In addition, the classification view (F) visualizes each categorized sample according to the predicted classes and calculated predicted scores for a selected epoch using the slider control (B). The parallel coordinates of the predicted probability distribution view (E) reveal the predicted score distributions over classes of user-selected samples. The detail view (D) shows features of samples learned by DNN; for example, important sentences and words in text analytics. The incorrect-prediction threshold slider (C) enables filtering of the samples that are consistently misclassified during the training process. In the classification view (F), box colors represent predicted scores with outlined boxes indicating incorrectly predicted samples. The outline color indicates the sample's labeled class. Small triangles denote samples with incorrect predictions greater than the threshold value. Image credit: *Junghoon Chae, ORNL.*

since the early days of ML and AI [89–91], more research is still needed. For instance, to set scientifically based rather than heuristic guidelines on acceptable classifiers that process experimental data from beamlines or predict failure in major components. It also is extremely unlikely that existing applied mathematics methodology will extend automatically to SciML. For example, classical linear algebra approaches are designed to optimize computations with sparsity structures arising from discretizations of PDEs. However, SciML needs are likely to be quite different. Similarly, classical von Neumann stability analysis of numerical approximations is unlikely to be adequate for establishing stability of complex learning schemes. Thus, complementary research in applied mathematics methodology is necessary.

Thrust A: How Should Scientific Reproducibility Be Implemented in Applications of Scientific Machine Learning?

Although "reproducibility" is a basic tenet of the scientific method, the definition of what reproducibility actually means is a current topic of debate [92]. Nevertheless, the key principle is that an independent research group should be able to replicate the findings of another whether they be experimental, theoretical, or computational. To state that a scientific result "holds" implicitly means that any independent researchers repeating the same process will be able to confirm the results for themselves. Without reproducibility, a scientific result loses any value as a foundation on which others can build. ML must be reproducible in order for SciML to assume its place as a trusted scientific technique.

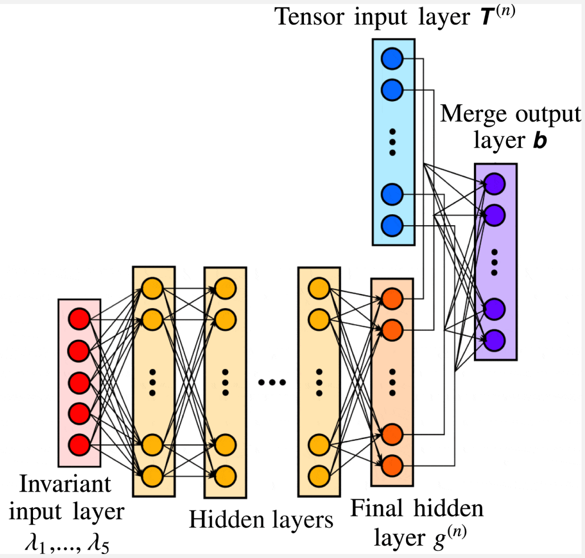
Best practices and computational resources are needed to accelerate the adoption of reproducible SciML research practices. Currently, there is a growing awareness that many of the results obtained using ML are *not* reproducible [93], including the case where one research group in ML spent nearly two months attempting to reproduce another group's reported findings without success [93]. It may be expected that results obtained using a particular computer code are guaranteed to be reproducible, but the reality in ML is that this is "far from [being the case]" [93]. Recent concerns about reproducibility in science are not confined to ML: a 2016 survey [94] of over 1,500 researchers by *Nature* reported that more than 70% of respondents reported having tried and failed to reproduce another scientist's findings. In a similar vein, [95] found that less than a 50% of academic research by drug companies is reproducible. Developing open frameworks for both theoretical and practical SciML research will foster an environment where reproducible research can be promoted and conducted. Preliminary steps in this direction already are underway in the ML community through initiatives such as Gym (`gym.openai.com`) and OpenML (`www.openml.org`).

Thrust B: Under What Conditions Is a Scientific Machine Learning Algorithm "Well-Posed"?

There is a basic need to establish a solid mathematical foundation for studying properties of the underlying implicit model, the algorithms used to analyze the models, and the sensitivities of outcomes to training data. For effective use in advancing and testing scientific hypotheses, SciML must be insensitive to the effects of intrinsic perturbations—in data and the model—that are not symptomatic of the underlying system. There are substantial links between the concept of SciML "stability" and the broader concept of "well-posedness". These foundations should formulate concepts and definitions that pave the way for a deeper understanding of SciML.

A mathematical framework provides the option to develop models and algorithms that are insensitive to the effect of perturbations not intrinsic to the underlying system. Appropriate reg-

Example 9: Embedding Rotational Invariance in Neural Network Models


Tensor input layer $\boldsymbol{T}^{(n)}$
Merge output layer $\boldsymbol{b}$
Invariant input layer $\lambda_1,\ldots,\lambda_5$
Hidden layers
Final hidden layer $g^{(n)}$

In the context of Reynolds-averaged incompressible turbulence modeling, an NN has been used in an eddy viscosity turbulence closure model [97]. The eddy viscosity model specifies the normalized Reynolds stress anisotropy tensor b as a linear combination of the tensors $\{T^{(n)}, n = 1, \cdots, 10\}$, which are given as analytical functions of the strain rate S and rotation rate R tensors. The coefficients in this linear combination, $\{g^{(n)}, n = 1, \cdots, 10\}$, depend on five tensor invariants $\lambda_1, \cdots, \lambda_5$, which also are given as analytical functions of (S, R). The NN, trained based on high-fidelity flow computations, is used to model the dependence of the $g^{(n)}$ on $\{\lambda_1, \cdots, \lambda_5\}$ and the subsequent linear combination to provide b. From physical arguments, the model for b needs to satisfy rotational invariance, ensuring that the physics of the flow is independent of the orientation of the coordinate frame of the observer. A special network architecture, a tensor basis neural network (TBNN), is proposed in [97]. The architecture shown embeds rotational invariance by construction. Without this guarantee, the NN model evaluated on identical flows with the axes defined in different directions can yield different predictions. Image adapted from *Ling et al. [97]*.

ularization enables useful information to be gleaned from an inverse problem in a clear and reproducible fashion. There is a challenge to identify appropriate techniques that can play a similar role in SciML.

Thrust C: How Should the Robustness, Performance, and Quality of Scientific Machine Learning Be Assessed?

The outcome of an ML process is either a decision (classification) or a prediction. For reliable and credible use of SciML, we need the ability to rigorously quantify ML performance in these outcomes. Performance measurement implies an assessment of quality, as well as a cost measure of computations and/or data preparation and management. Traditional measures of acceptable quality based on statistical cross-validation-type approaches often are heuristic. Measures of prediction quality such as *a priori* and *a posteriori* error estimates for numerical approximations of PDEs [96] (familiar to the finite element modeling community) will be transformative in allowing the development of optimal and reliable ML algorithms for different uses. Such error estimates also will enable SciML processes that allow iterative model improvement. Research establishing quantitative estimates of prediction quality, including effective confidence bounds, will greatly enhance the usefulness of SciML to decision makers and users. Finally, research is needed on algorithms that have proven convergence rates with weak dependence on bad data, especially in situations with a large amount of data of unproven quality or minimal availability of human expertise.

2.4 Data-Intensive Scientific Machine Learning

The central role of data in SciML suggests an associated fundamental role for statistics and UQ methods. Data used to train ML models are often noisy, uncertain, incomplete, sparse, and only partly informative. Similarly, ML models themselves are subject to uncertainty in their general form, internal structure, and associated parameters. Statistics provide an array of methods to address these data and model complexities and uncertainties in SciML. Ultimately, this enhances the ability to use extreme-scale computations and experimental data for scientific discovery in physical systems highly relevant to DOE.

Statistics and UQ methods can add significant robustness to SciML fitting/regression methods. Often, statistical methods are useful for dealing with the problem of over-fitting with DL and CNNs given small amounts of data [98]. In many places where multiple minima exist and the solution space is large, a high degree of ill-conditioning is present, considering the many nearly equivalent feasible solutions. Reformulating the issue as a statistical inverse problem can add significant conditioning, changing the question from determining the best solution to finding the set of solutions with significant probability. Further, this reformulation provides a solution with quantified uncertainty estimates in its parameters/weights and potentially its structure. For example, incorporating probabilistic modeling in ML [99] targets feasible estimation of uncertainty in ML predictions. A robust probabilistic/statistical analysis of noise, errors, and uncertainties in ML in high-dimensional systems enables reliable discovery of correlations and causal structures in large-scale systems of interest. Similarly, Bayesian integration is useful for general high-dimensional function applications [100] with relevance in ML. Bayesian modeling also is finding utility in Bayesian generative adversarial networks [101, 102]. Additionally, there are potential advantages for regression and SciML with non-parametric models, such as GPs, versus pre-determined NN structures in DL [103].

Despite the stated benefits, some challenges often increase upon setting the ML problem in a probabilistic context. These include increased dimensionality from probabilistic modeling, the need to deal with high computational costs in UQ sampling strategies, and the need for adequate error modeling in data and models. The non-parametric identification of structure in high-dimensional data is a significant challenge. High dimensionality can necessitate large numbers of samples to allow for reliable identification of underlying structure in high-dimensional computational/experimental data. This is particularly true when the system does not admit a sufficiently low-dimensional underlying structure. Likewise, UQ in ML in high dimensions presents a number of challenges. UQ for SciML involves the formulation of probabilistic machine learning (PML) models, inference of ML model structure and (hyper)-parameters with quantified uncertainty, and forward propagation of uncertainty to ML model predictions. Both the statistical inverse problem (PML model training) and the forward UQ problem (propagating uncertainty to ML model predictions) can become quite expensive in high-dimensional complex ML models.

Thrust A: How Should Structure Be Extracted from High-Dimensional Data and Complex Models?

Advances are necessary to improve methods for analysis and discovery of structure in scientific data and physical systems models. As described in other PRDs (refer to Sections 2.1 and 2.2), knowledge of underlying structure affords a means for interpreting the features of data and models. Structure includes a number of underlying properties of data and models. For example, correlation and causal structure in data and models are crucial ingredients. Similarly, structure includes the specification of low-dimensional manifolds and geometry underlying high-dimensional data and complex dynamical systems. Structure is a crucial component of some ML methods (e.g., GP models), and the accuracy of these methods depends on how well one can learn the structure from data or physics-based models. While there is a large body of work in this area, there are significant gaps in the state of the art when dealing with both high dimensionality and complexity in data and models.

Despite recent advances in methods for discovering low-dimensional structures in data, much work is needed in applying these methods to large-scale physical systems of interest to DOE. Continued research toward improved methods for discovering sparsity [104–109] and low-rank structure [110–115] in ML models will be useful for dealing with the high-dimensionality challenge. Additionally, extracting models from data [116–119] is an important area of work toward improved ML. Similar advances are needed to improve methods for discovering structure underlying data, including learning underlying geometry [120] beyond principal component analysis (PCA) [121–123] and building on work targeting discovery and representation of diffusion manifolds [124–128]. Knowledge of data structure is necessary for effective sampling on manifolds, potentially allowing for embedding physical constraints in SciML system training. There also are opportunities for advances in the definition and utilization of geodesics [129–133] and correlation metrics [134–136] for providing measures of distance that are useful for model comparison, calibration, selection, and validation.

Research opportunities are also abundant in the area of causality. Although "correlation does not imply causation," both are challenging to analyze in high dimensions. There is a need for approaches that apply data compression or dimension reduction before performing correlation analysis in large and complex data sets. Causation is more challenging to identify. If one has a known causal graph, there are statistical metrics to compare an inferred causal structure with the known ground truth causal structure. These metrics include precision (the fraction of inferred edges included in the ground truth) and recall (the fraction of edges from the ground truth that are identified in the inferred or ML approach). In most cases, however, a known causal graph does not exist. There are approaches for learning causal structure from data [137–139], including Bayesian hierarchical models [140], additive noise models [141], and information-geometric causal inference [142]. However, the reliability of these methods leaves much to be desired, and there are arguments for the need to address fundamental challenges in the current probabilistic framework to represent causality [143]. Developing and applying these ideas to higher dimensions in complex physical models is a significant research task with many methodological challenges.

Thrust B: How Can Complex and High-Dimensional Spaces Be Efficiently Sampled?

Exploring model spaces in SciML may involve sampling in categorical/discrete spaces. These are hard problems, particularly in high dimensions. Discrete structures arise in many contexts of SciML, including from non-Euclidean input data, models that can be represented as (or projected to) graphs (or networks), and representation of physical principles or constraints. When multiple options for computational modeling exist at each stage (e.g., mesh discretization, boundary condition representation, constitutive material model choices, or PDE solver options), the number of candidate models becomes quite large. Such model selection is expensive because of the requirement to calculate model evidence*: one needs to generate the likelihood function and integrate it with respect to the parameter density over all parameter values [144]. Finding cheaper alternatives to Monte Carlo sampling over large combinatorial model spaces is an open challenge [145], although variational Bayesian inference provides a useful alternative in a number of instances.

* "Model evidence" also is sometimes called "Bayesian marginal likelihood."

In addition to sampling in discrete spaces, there is a need for continued development of improved methods for sampling, optimization, and integration in high-dimensional spaces. Similarly, important research directions include development of effective methods for representing and constructing high-dimensional functions and for discovery and construction of low-dimensional manifolds in high-dimensional spaces. Determining probabilistic characterizations of the points on those manifolds and efficient methods for accurately generating sample realizations from manifolds are also needed. A key need is to do this efficiently, using minimal numbers of large-scale complex model samples, and to be computationally feasible. As such, there is a need for continued improvements in multifidelity methods to allow the use of models at varying degrees of fidelity—hence, cost—to optimally achieve the desired target accuracy.

Thrust C: How Can Robust Scientific Machine Learning Be Achieved with Noisy, Complex Data?

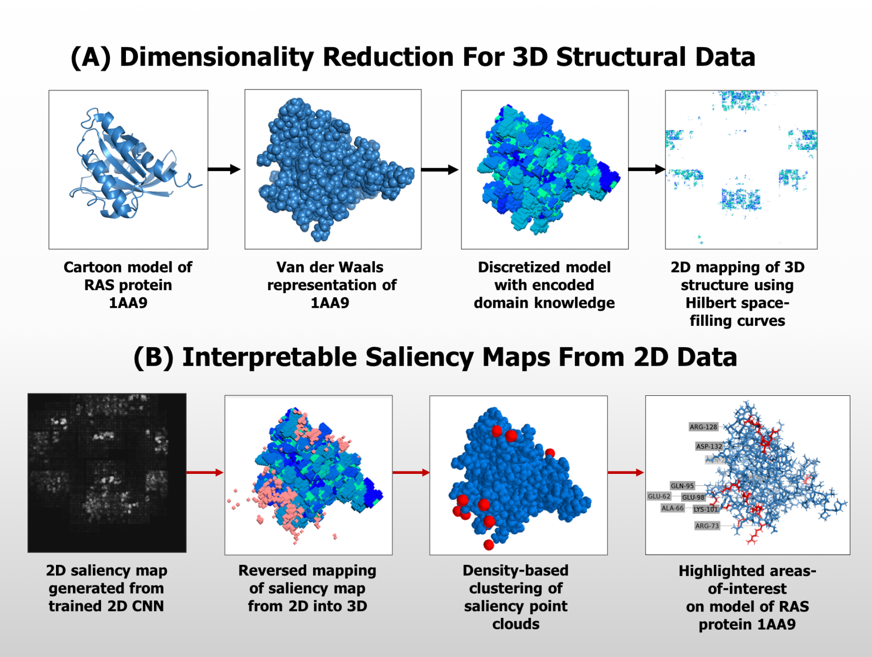
A crucial need in SciML is to maximize the amount of learning from the available scarce data and to actively choose data and/or new experiments wisely. An extreme case in this context is where data are simply not available, but one is limited to using available data summaries, such as statistics on specific functionals of the data. When data are available, there often are serious challenges associated with complexities in its structure. Data often are unrepresentative, whereby the available observations do not reflect the distribution of the measurements in general. Additionally, data can have gaps; i.e., non-uniformly spread over ranges of conditions of interest. Notably, not all data have the same relevance, quality, error and noise, and cost of acquisition.

In SciML, we deal with data from physical systems (observational and experimental data), as well as from computational simulations (of physical systems). Both data types may be expensive to collect or generate. Thus, it is of interest to be able to design these data collection or generation procedures so as to minimize the amount of data required and the associated costs. Active learning [146] and optimal experimental design [147] are research fields that aim to achieve this in a systematic and quantitative manner. Specifically, active learning is tightly connected to “sequential” (adaptive) experimental design, where the next decision depends on what has been observed and learned in the past. In active learning, there are opportunities for research to improve the design optimization process, especially for systems that are dynamically evolving in time. In particular, the availability of a physical model can provide predictions that enable look-ahead for future design choices.

2.5 Machine Learning-Enhanced Modeling and Simulation

Scientific computing within DOE traditionally has been dominated by complex, resource-intensive numerical simulations. However, the rise of data-driven SciML models and algorithms provides exciting new opportunities. Traditional scientific computing forward simulations often are referred to as “inner-loop” modeling (cf. “outer-loop” problems, such as sensitivity analysis and optimization, are discussed in Section 2.6). The combination of traditional scientific computing knowledge with ML-based adaptivity and acceleration has the potential to increase the performance and throughput of inner-loop modeling. Conversely, to address truly “big data” data using high-performance computing (HPC) resources, ML algorithms must be scalable and efficient. The DOE scientific computing community has decades of expertise involving numerical algorithms, especially dealing with the challenges of parallel computing, that can benefit the inner loop of training in ML. Therefore, an opportunity exists to advance ML, particularly at scale, by entraining more involvement from the computational mathematics community.

Example 10: Deep Learning Interpretability in Protein Coarse-Graining


(A) Dimensionality Reduction For 3D Structural Data
Cartoon model of RAS protein 1AA9
Van der Waals representation of 1AA9
Discretized model with encoded domain knowledge
2D mapping of 3D structure using Hilbert space-filling curves
(B) Interpretable Saliency Maps From 2D Data
ARG-128
ASP-132
GLN-95
GLU-62
GLU-98
ALA-66
LYS-101
ARG-73
2D saliency map generated from trained 2D CNN
Reversed mapping of saliency map from 2D into 3D
Density-based clustering of saliency point clouds
Highlighted areas-of-interest on model of RAS protein 1AA9

High-level data pipeline overview for dimensionality reduction of 3D protein structures (A) and interpretation of saliency maps from a trained CNN model (B). A Van der Waals representation of protein structure is generated in 3D space and discretized into a voxel representation. Supplementary biological channels of information are added to the discrete representation, such as residue hydrophobicity and charge. The dimensionality reduction algorithm takes the discrete 3D structure and maps it to a two-dimensional grid using Hilbert space-filling curves. The dimensionality reduction algorithm is a one-to-one mapping of points between 3D and two-dimensional space and preserves locality between points. The final two-dimensional data structure provides reduced two-dimensional CNN training times when compared against 3D CNN counterparts [148]. Saliency maps generated from trained two-dimensional CNN models can be interpreted by first applying the dimensionality reduction algorithm in reverse to map two-dimensional data back into 3D space. The resulting salient point clouds then can be clustered using density-based clustering to identify regions of high saliency. These areas along the 3D structure are regions that highly influence the output of the CNN model. From these salient regions, specific residues can be identified that fall in close proximity to the salient regions. Image credit: *Rafael Zamora-Resendiz and Silvia Crivelli, LBNL.*

Thrust A: How Can Machine Learning Be Used to Enable Adaptive Scientific Computing?

The overarching objective of this thrust is to use ML to improve the performance and throughput of numerical simulations through ML-enabled adaptivity. Success will result in a reduced need for human intervention and specialized expert knowledge to produce forward simulations efficiently and accurately. The ultimate benefit will be the ability to produce intelligent simulation capabilities that automatically provide solutions robustly with guaranteed accuracy and in the least amount of time—all within the user's prescribed constraints.

Generically, the "inner loop" refers to the computations within each time step of a simulation and/or within each iteration in an iterative solver. In the typical inner loop of a forward simulation, one or more numerical algorithms are used to advance or converge the solution of the discrete model. Typically, there are many choices for the numerical analyst: discretizations, linear solvers, nonlinear solvers, eigensolvers, interpolations, preconditioners, different models, etc. Even for a specific solver, for instance, there are additional choices, such as relaxation parameters, smoothing operators, and/or the number of Krylov subspace vectors. Expert knowledge is the primary tool by which these choices are made today, but the choices often are made and fixed at the start of the simulation. They do not change with the evolving behavior of the numerical solution. Furthermore, as demonstrated by adaptive step size control in ordinary differential equation (ODE) integrators [149], adaptive mesh refinement [150], and shock-capturing schemes [151–154], there can be great advantages to adapting the solution process to the solution, even as it is being computed. Adaptivity can present complex choices between method order and discretization or matching solution processes. Much of this adaptivity is driven by heuristics and simplified analysis, which still require a great deal of human experience and evaluation. There is a need for research to realize the opportunity that SciML presents in enabling even greater adaptivity, which will help address the growing complexity of applications and architectures; increase performance in terms of fidelity, robustness, and/or time to solution; and increase throughput by reducing the need for human guidance or intervention in the simulation workflow.

The software-related issues of compilation, implementation selection, and problem decomposition, which can occur across distributed resources or increasingly across a heterogeneous on-node mix of processors, co-processors, and memory types, are closely related to ML-enhanced modeling and simulation. Selection of the optimal choices is highly problem- and architecture-dependent. Today, such choices usually are made for convenience or by intuition. There are examples of mathematical libraries that auto-tune; e.g., ATLAS [155], but the auto-tuning procedure is a static optimization done at compilation without reference to a specific problem or changing resource constraints. Programming models such as RAJA [156] and Kokkos [157] provide a means to abstract away implementation details, and tools including Apollo [158] are being developed to determine higher performance policies (implementations) from measured data. However, these decisions are made statically, not on the fly, in response to actual performance for a specific simulation. SciML provides an opportunity to add dynamism not only in the choice and control of algorithms, but in the choices of data layouts and architecture-aware algorithm implementations.

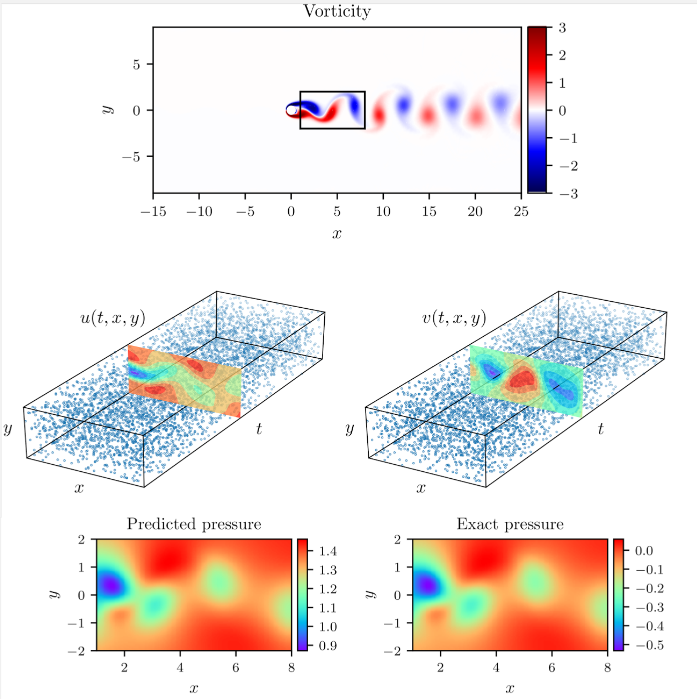
The possibilities for advances in ML-enhanced modeling and simulation are at least as numerous as the multitude of algorithms used in scientific computing. Such advances will benefit from research into adaptive numerical algorithms and adaptive implementation of numerical algorithms.

Thrust B: How Can Department of Energy Scientific Computing Expertise Help Scientific Machine Learning?

The DOE scientific computing community has decades of experience in scalable numerical algorithms that can benefit ML. SciML using HPC resources will require ML algorithms that are scalable and efficient. The inner loop of the SciML training process involves mathematical optimization algorithms and the linear algebra solvers used within these optimization techniques. A key consideration is that the numerical problems presented by ML will, in general, have different structures than the traditional PDE-based problems, which affords an opportunity to develop novel solver techniques specifically designed for ML.

The performance of an optimization algorithm used in training is a core design consideration in ML. As learning models (and their training) have become more diverse and complex, algorithms have been adopted from diverse areas, such as convex optimization [160], non-smooth optimization [161], robust optimization [162], semi-definite programming [163, 164], stochastic optimization [165], derivative-free optimization [166, 167], and global optimization [168]. A majority

Example 11: Improving Fluid Dynamics Simulations with Machine Learning



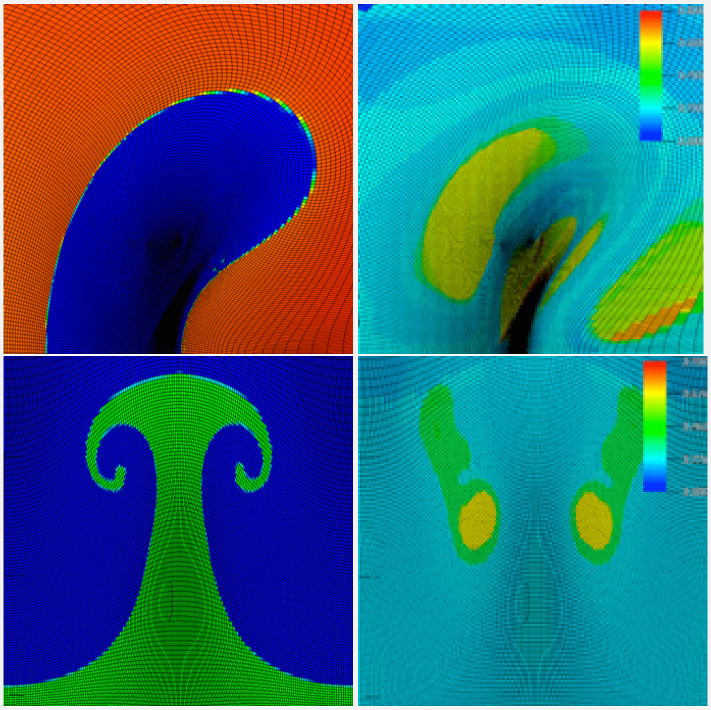
Correct PDE	$u_t + (uu_x + vu_y) = -p_x + 0.01(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.01(v_{xx} + v_{yy})$
Identified PDE (clean data)	$u_t + 0.999(uu_x + vu_y) = -p_x + 0.01047(u_{xx} + u_{yy})$ $v_t + 0.999(uv_x + vv_y) = -p_y + 0.01047(v_{xx} + v_{yy})$
Identified PDE (1% noise)	$u_t + 0.998(uu_x + vu_y) = -p_x + 0.01057(u_{xx} + u_{yy})$ $v_t + 0.998(uv_x + vv_y) = -p_y + 0.01057(v_{xx} + v_{yy})$

This example illustrates the capabilities obtained by incorporating domain knowledge into a DNN. Given scattered and potentially noisy data on the streamwise $u(t, x, y)$ and transverse $v(t, x, y)$ velocity components of an incompressible fluid flow in the wake of a cylinder, we can employ a *physics-informed neural network* that is constrained by the Navier-Stokes equation to identify unknown parameters (e.g., the Reynolds number of the flow) and reconstruct a velocity field that is guaranteed to be incompressible and satisfy any boundary conditions, as well as to recover the entire pressure field, which by definition can only be identified up to a constant. The algorithm is trained in absence of any data on the pressure itself. This theme of consistently inferring a continuous quantity of interest from a small set of auxiliary measurements by leveraging the underlying laws of physics demonstrates the capabilities that physics-informed ML can offer. Figure adapted from *Raissi et al. [159].*

of the ML frameworks deployed today use some variant of the stochastic gradient descent method for this purpose [169]. Many of these algorithmic variants have been rigorously analyzed [170–172] and differ based on their parallelism and access to training data, including cases where very few passes through the data are allowed or where the data are distributed in such a way that synchronous access is infeasible [173]. Significant research has been performed by the ML and optimization communities, leading to improved mini- and multi-batch [174–176] as well as asynchronous [177, 178] algorithms. At the same time, accelerated and momentum-based techniques have been studied to reduce the number of iterations required by an optimization algorithm [179, 180]. Advances in non-convex optimization methods also have benefited ML in terms of the ability to incorporate loss functions and training objectives with favorable learning and regression properties [181–183]. Examples include methods for variance reduction [184, 185] and globalization techniques, such as trust-region methods [186, 187]. Depending on problem size and solution requirements, first-order [188, 189], second-order [190, 191], and secant [192] methods have been successfully employed in ML.

Despite ongoing research, much work remains to develop numerical algorithms that improve the speed and efficiency of ML training, particularly at large scale. A closer collaboration between the numerical algorithms and ML communities could prove to be a fruitful research area that advances the capabilities of ML with HPC. There is a research need for scalable and efficient ML on HPC resources by leveraging the expertise of the scientific computing community to develop

Example 12: Machine Learning-Enhanced Mesh Relaxation



The arbitrary Lagrangian-Eulerian (ALE) method is used in various engineering and scientific applications to enable multi-physics simulations. Unfortunately, the ALE method can suffer from simulation failures, such as mesh tangling, that require users to adjust parameters throughout a simulation just to reach completion. Identifying which parameters to adjust, when, and by how much is as much art as science, and finding the right ALE strategy often requires many iterations (simulation fails, adjust parameters, and rollback) that can be disruptive and time consuming.

In recent work [193], a supervised learning framework for predicting conditions leading to ALE simulation failures was developed using a Random Forest learning algorithm. The learning and inference algorithms were integrated into a production ALE code for modeling high-energy-density physics. Specifically, a mesh relaxation strategy was developed that incorporates the prediction values by relaxing zones more if they have a higher risk of failure. Models were trained on a variety of test problems. This learned relaxation strategy was applied to standard test problems without encountering any mesh tangling failures. Here, we show results from a shock-bubble interaction (Bubble Shock) and a shock-interface interaction (Shock Tube), which models a planar shock traveling through an interface and generating a Richtmyer-Meshkov instability. The left-side plots show the visualization, using pseudo-coloring of materials, of the Bubble Shock (top) and Shock Tube (bottom). The right-side plots are a pseudo-coloring of the respective risk-of-failure prediction values derived from the trained models from each test problem. For the first time, it shows that ML can be a viable approach to predict simulation failures when trained with a sufficient number of examples. Image credit: *M. Jiang, LLNL [193].*

appropriate scalable solver algorithms. Success in this research will lead to faster, more robust training of learned models, particularly at large scale, that will allow training over larger data sets and enable *in situ* training in scientific applications.

The scientific computing community has a well-established history of developing advanced, scalable solvers. As such, there is an obvious opportunity to leverage the existing knowledge from high-performance scientific computing to address SciML method costs and achieve scalability. The close collaboration on optimization techniques for ML (as already described) is merely the beginning. Other techniques from computational mathematics; e.g., multilevel solvers, multifidelity solvers, floating-point compression, and domain decomposition techniques, also may contribute to improved ML training performance. Finally, an opportunity exists for the co-design of new and adaptation of existing SciML algorithms for different computer architectures.

2.6 Intelligent Automation and Decision Support

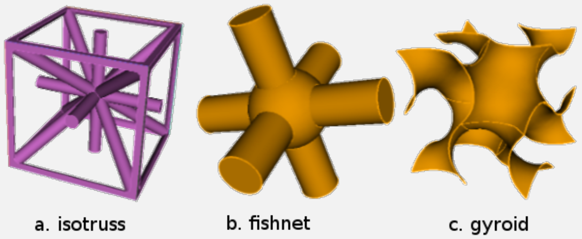
The term "outer loop" is used increasingly to describe computational applications that form outer loops around a forward simulation [194]. Examples of outer-loop applications include optimization, UQ, inverse problems, data assimilation, and control. These applications constitute a significant target of many of DOE's simulation and modeling capabilities—in many cases, for automation or in support of decisions. There are a variety of different outer-loop forms, all tasks that could benefit from SciML. In particular, *design optimization*, *optimal control*, and *optimization under uncertainty* approaches seek to explore high-dimensional parameter space to determine optimal data acquisition decisions according to (possibly multi-objective) success metrics. The thrusts featured in this section describe these challenges and their potential application to DOE facilities.

Thrust A: How Should Machine Learning Be Used to Intelligently Guide Data Acquisition?

Often, scientists face severe constraints (budget, time, computational resources) in running more physical experiments or computational simulations to support outer-loop applications. To achieve an objective, the question of "where do I sample next?" is critical. Objectives may include providing the "best" experimental data to most inform model parameters in a calibration process, minimizing prediction uncertainty bounds, identifying a descent direction in an optimization search, or choosing a few high-fidelity simulations that, when combined with corrected low-fidelity simulations, achieve an accurate prediction at a much smaller cost than running a large number of high-fidelity simulations. This question relates closely to the challenge of active learning (discussed in Section 2.4). Here, we focus on guiding data acquisition in outer-loop settings, such as optimization, UQ, and sensitivity analysis. Research is needed to help identify what data (e.g., physical experiments or computational simulations) should be queried or gathered to most improve our achievement of an outer-loop result. ML approaches can bring new, helpful ideas to this data acquisition challenge, especially with respect to high-dimensional problems. There is a need to develop mathematically and scientifically justified methods to guide data acquisition and assure data quality and adequacy in outer-loop algorithms. For DOE applications, these outer-loop algorithms are specially crafted to exploit the forward problem structure, such as in PDE-constrained optimization formulations, adjoint methods for sensitivity analysis, and scalable UQ formulations for stochastic PDEs. We expect this research area will involve methods that combine and leverage statistical inference, experimental design, optimization, and ML, as well as methods related to specific problem classes (PDEs, networks, etc.).

The complexity of computational models has grown enormously over the past few decades with routine analyses involving coupled physics or multiscale models. With the increasing complexity of the models, the ability to "start small" and adaptively add data as necessary and at locations that can most improve model fidelity and scientific understanding becomes more important in pursuit of an outer-loop goal. Much research already has been done on adaptive schemes for computer experiments that maximize information gain [195] or balance exploitation and exploration of a data space [196, 197]. Bayesian optimization is a related class of methods that are becoming increasingly popular for solving global optimization problems governed by expensive functions [198–200]. This optimization approach can be turned to reflect problem structure and goals, such as maximizing the expected improvement [196], information gain [201–203], and utility over a finite budget of remaining evaluations [204]. The utility function for the optimization also can be modified to leverage gradient information [205] and to draw information from multiple sources, such as approximate models and noisy data sources [206]. Recent research in optimal

Example 13: Enabling the Automated Design of Multiscale Materials with Machine Learning



Today's additive manufacturing technologies are successful in fabricating microscale metamaterials unit cells that can be combined to obtain materials with novel properties. The figure features examples of metamaterial unit cells: a) isotrusses have a high stiffness/density ratio, b) fishnets possess interesting electromagnetic band gap properties, and c) gyroids have large surface areas that are of interest in batteries and supercapacitors. Guided by data generated from high-resolution HPC physics simulations, SciML potentially can provide low-dimensional surrogate models to better understand the behavior of metamaterials unit cells. Image credit: *C. G. Petra and D. White, LLNL.*

experimental design has focused on using adaptively selecting high-fidelity model runs to supplement many low-fidelity model runs in order to achieve a particular objective, such as an overall variance bound on an expected response quantity [207, 208].

We expect research opportunities in several areas for intelligent automation and decision support for outer-loop applications. One need focuses on using ML methods to identify optimal data acquisition strategies within outer-loop applications of optimization, UQ, and sensitivity analysis. A second research direction focuses on engaging SciML methods to optimally manage the complicated workflows of outer-loop algorithms; e.g., using ML to adaptively select algorithmic and simulation parameters. The fields of Bayesian optimization, statistical inference, experimental design, and ML are intertwined. Our expectation is that this tight interrelationship will be able to substantially improve state-of-the-art data acquisition strategies for outer-loop applications.

Thrust B: How Can Scientific Machine Learning Be Used to Improve Science Outcomes from Science Facilities?

DOE has developed and manages numerous experimental and computational scientific facilities: x-ray light sources, neutron scattering facilities, magnetic fusion facilities, particle accelerators, Argonne Leadership Computing Facility (ALCF), Oak Ridge Leadership Computing Facility (OLCF), National Energy Research Scientific Computing Center (NERSC), ESnet, etc. While these facilities generate considerable amounts of scientific data, their operations also are driven by data. With increasing numbers of available sensors and associated data streams, the ability for real-time monitoring of the state and performance of facilities continues to increase dramatically. However, the ability to make decisions based on these multimodal data streams has not kept pace. If we could make better use of these data, we could better optimize the use of scientific facilities. In this sense, managing a scientific facility is a significant outer-loop challenge and SciML has an important decision support role to play if we aim to translate the available operational data into actions.

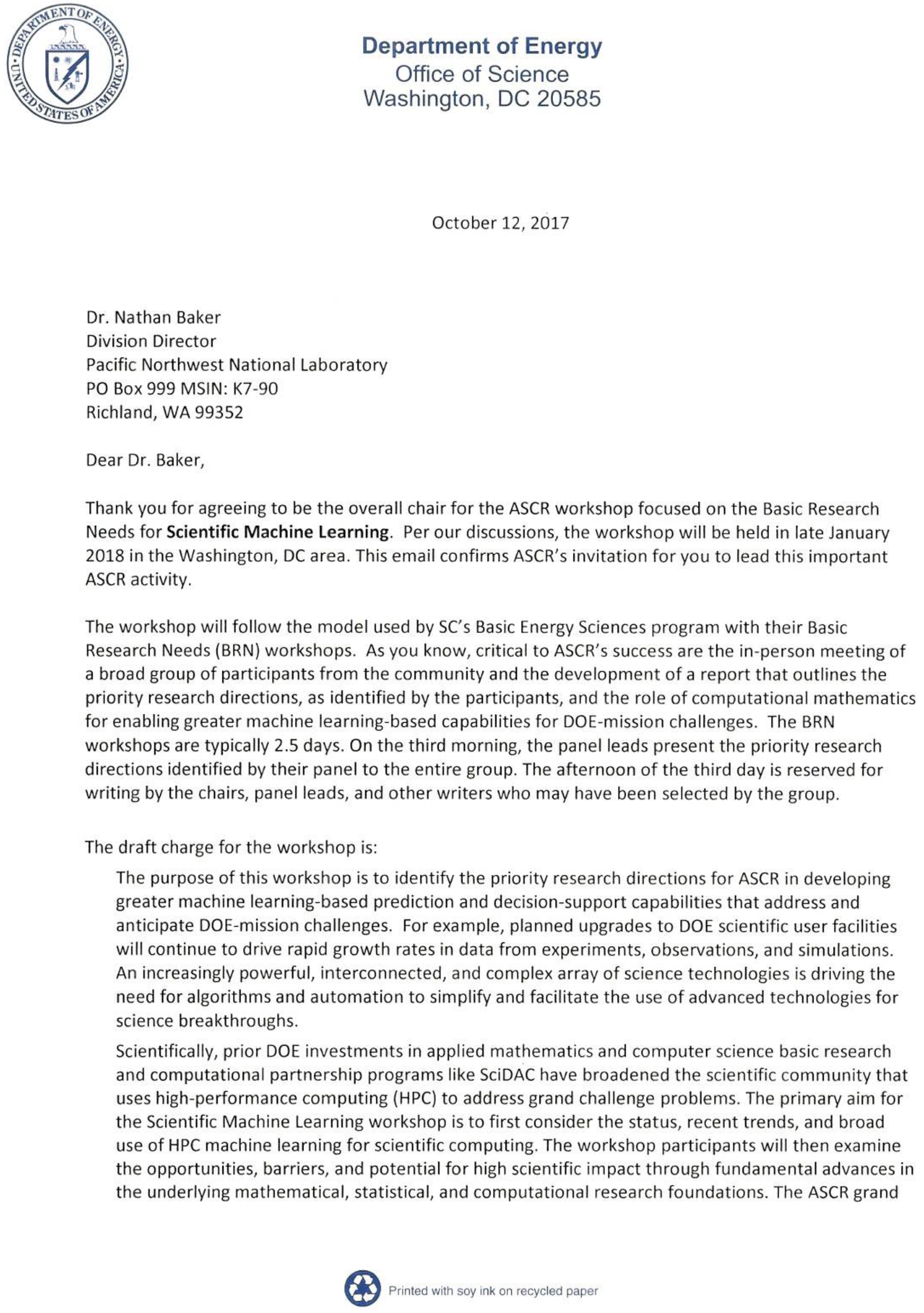
Research is already underway to exploit scientific facility data. At the National Ignition Facility (NIF), small defects can focus laser light into locally extreme intensities, causing further damage and growth of the defects and ultimately destroying the optical glass [209]. Therefore, between each laser shot, a robotic camera images each NIF optical component. The resulting data are used with image recognition software to identify and track defects. Another example comes from computational facilities (such as ALCF, NERSC, and OLCF). Currently, fault prediction, detection, and recovery are fairly brittle on pre-exascale systems deployed at these facilities, and we only expect that reliability issues will become more challenging on future exascale systems. The time is ripe for data-driven, ML-based systems to track real-time telemetry data and predict if computational nodes are poised to fail. This information can feed into an adaptive runtime and scheduling system that can dynamically migrate jobs to new computational resources. For example, machine-learning-based silent data corruption detection (MACORD) [210] presents a SciML-based framework for detecting silent errors in HPC machines. In their work, Tuncer et al. [211] employ historical data on resource usage and performance counter data to diagnose performance anomalies of HPC applications. SciML approaches also have been used to model the energy efficiency of modern HPC data centers [212]. There is ample opportunity for further development of intelligent automation for scheduling and system diagnostics.

In summary, the goal of this thrust is to emphasis the need for research into capabilities that lead to more data-driven optimized management of scientific facilities. By harnessing information about the facility in the available data, more rapid and informed decisions can be made about its health and performance, leading to improved operational decisions and, ultimately, increased scientific throughput by decreased downtime and better scheduling of calculations or experiments.

Appendix A

Scientific Machine Learning Workshop

A.1 Workshop Charge



Department of Energy
Office of Science
Washington, DC 20585

October 12, 2017

Dr. Nathan Baker
Division Director
Pacific Northwest National Laboratory
PO Box 999 MSIN: K7-90
Richland, WA 99352

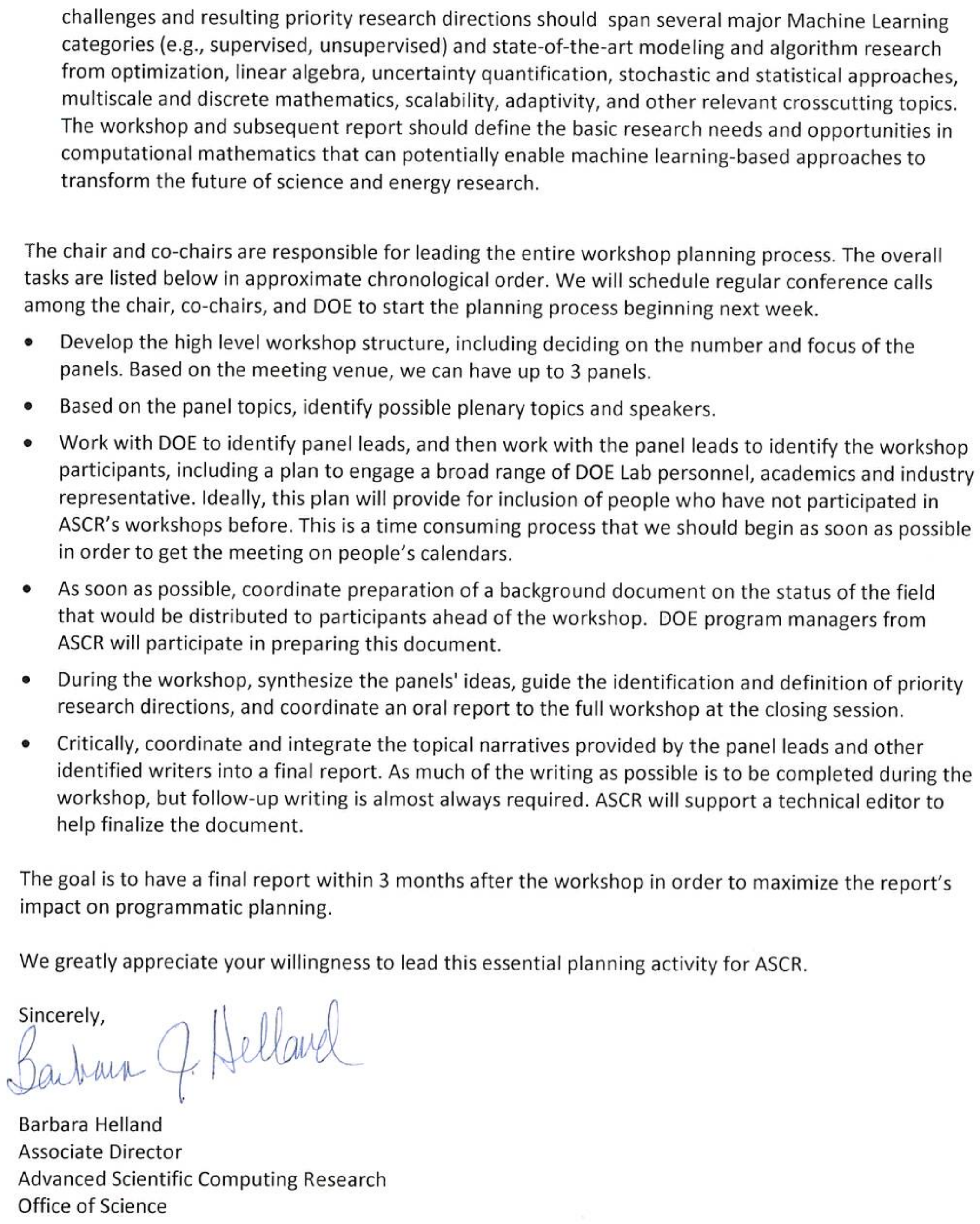
Dear Dr. Baker,

Thank you for agreeing to be the overall chair for the ASCR workshop focused on the Basic Research Needs for **Scientific Machine Learning**. Per our discussions, the workshop will be held in late January 2018 in the Washington, DC area. This email confirms ASCR's invitation for you to lead this important ASCR activity.

The workshop will follow the model used by SC's Basic Energy Sciences program with their Basic Research Needs (BRN) workshops. As you know, critical to ASCR's success are the in-person meeting of a broad group of participants from the community and the development of a report that outlines the priority research directions, as identified by the participants, and the role of computational mathematics for enabling greater machine learning-based capabilities for DOE-mission challenges. The BRN workshops are typically 2.5 days. On the third morning, the panel leads present the priority research directions identified by their panel to the entire group. The afternoon of the third day is reserved for writing by the chairs, panel leads, and other writers who may have been selected by the group.

The draft charge for the workshop is:

> The purpose of this workshop is to identify the priority research directions for ASCR in developing greater machine learning-based prediction and decision-support capabilities that address and anticipate DOE-mission challenges. For example, planned upgrades to DOE scientific user facilities will continue to drive rapid growth rates in data from experiments, observations, and simulations. An increasingly powerful, interconnected, and complex array of science technologies is driving the need for algorithms and automation to simplify and facilitate the use of advanced technologies for science breakthroughs.
>
> Scientifically, prior DOE investments in applied mathematics and computer science basic research and computational partnership programs like SciDAC have broadened the scientific community that uses high-performance computing (HPC) to address grand challenge problems. The primary aim for the Scientific Machine Learning workshop is to first consider the status, recent trends, and broad use of HPC machine learning for scientific computing. The workshop participants will then examine the opportunities, barriers, and potential for high scientific impact through fundamental advances in the underlying mathematical, statistical, and computational research foundations. The ASCR grand



challenges and resulting priority research directions should span several major Machine Learning categories (e.g., supervised, unsupervised) and state-of-the-art modeling and algorithm research from optimization, linear algebra, uncertainty quantification, stochastic and statistical approaches, multiscale and discrete mathematics, scalability, adaptivity, and other relevant crosscutting topics. The workshop and subsequent report should define the basic research needs and opportunities in computational mathematics that can potentially enable machine learning-based approaches to transform the future of science and energy research.

The chair and co-chairs are responsible for leading the entire workshop planning process. The overall tasks are listed below in approximate chronological order. We will schedule regular conference calls among the chair, co-chairs, and DOE to start the planning process beginning next week.

- Develop the high level workshop structure, including deciding on the number and focus of the panels. Based on the meeting venue, we can have up to 3 panels.
- Based on the panel topics, identify possible plenary topics and speakers.
- Work with DOE to identify panel leads, and then work with the panel leads to identify the workshop participants, including a plan to engage a broad range of DOE Lab personnel, academics and industry representative. Ideally, this plan will provide for inclusion of people who have not participated in ASCR's workshops before. This is a time consuming process that we should begin as soon as possible in order to get the meeting on people's calendars.
- As soon as possible, coordinate preparation of a background document on the status of the field that would be distributed to participants ahead of the workshop. DOE program managers from ASCR will participate in preparing this document.
- During the workshop, synthesize the panels' ideas, guide the identification and definition of priority research directions, and coordinate an oral report to the full workshop at the closing session.
- Critically, coordinate and integrate the topical narratives provided by the panel leads and other identified writers into a final report. As much of the writing as possible is to be completed during the workshop, but follow-up writing is almost always required. ASCR will support a technical editor to help finalize the document.

The goal is to have a final report within 3 months after the workshop in order to maximize the report's impact on programmatic planning.

We greatly appreciate your willingness to lead this essential planning activity for ASCR.

Sincerely,

Barbara J. Helland

Barbara Helland
Associate Director
Advanced Scientific Computing Research
Office of Science

A.2 Pre-Workshop Report – Factual Status Document

Executive Summary

This pre-workshop report about SciML has been prepared on behalf of the ASCR program in the DOE Office of Science (SC) and ASCR Program Manager Dr. Steven Lee. The ASCR program has a long tradition of providing scientific computing leadership for transforming science and energy research. ASCR accomplishes this by developing and maintaining world-class scientific computing and network facilities, advancing research in applied mathematics and computer science, and working in partnership with a broad range of researchers to solve increasingly complex challenges in computational and data sciences.

Recently, interest in ML-based approaches for science and engineering has increased rapidly. This growing interest stems from the combined development and use of efficient analysis algorithms; massive amounts of data available from scientific instruments and other sources; advances in HPC; and the successes reported by industry, academia, and research communities. A conventional notion of ML involves training an algorithm to automatically find patterns, signals, or structure that may be hidden within massive data sets whose exact nature is unknown and, therefore, cannot be programmed explicitly. The predictive capability of the algorithm is a learned skill. We seek to improve upon and harness the predictive power of ML to maximize its impact on DOE mission and science/engineering applications.

The purpose of the workshop will be to identify opportunities and gaps, thereby defining priority research directions for applied mathematics in SciML. The workshop will specify the challenges and opportunities for increasing the rigor, robustness, and reliability of ML necessary for routine use in DOE science and engineering applications. In particular, the focus will be on identifying the unique ML needs and requirements of scientific applications. In terms of examples, these challenges may include rigorous analysis methods for developing and testing ML methods, understanding ML method approximation power, and bounding data and compute complexity of ML approaches. These challenges will be discussed in the context of existing methods with demonstrated use in scientific applications. We also seek to engage the attention of the world-leading ASCR mathematics and computer science communities in these new research priorities. The workshop will focus principally on the mathematical challenges in ML theory and application. These challenges will be discussed in the context of existing methods with demonstrated use in scientific applications.

The purpose of this pre-workshop report is to provide sufficient information about the landscape of ML research—as it relates to DOE SC missions and ASCR capabilities—to inform the upcoming workshop. Because of the rapid pace of development in ML, as well as its roots in many different fields, a complete review is not feasible for this report. Instead, we focus on three main areas of SciML: examples of potential impact on DOE SC missions, major areas of ML research for scientific discovery, and the intersection of existing ASCR capabilities with ML research.

Introduction

Scope

Recently, interest in ML-based approaches for science and engineering has increased rapidly. This growing interest stems from the combined development and use of efficient analysis algorithms;

massive amounts of data available from scientific instruments and other sources; advances in HPC; and the successes reported by industry, academia, and research communities. A conventional notion of ML involves training an algorithm to automatically find patterns, signals, or structure that may be hidden within massive data sets whose exact nature is unknown and, therefore, cannot be programmed explicitly. The algorithm's predictive capability is a learned skill. We seek to improve upon and harness the predictive power of ML to maximize its impact on DOE mission and science/engineering applications (for an overview of DOE missions, refer to recent reports [4, 5]).

The workshop's purpose will be to identify opportunities and gaps, thereby defining HPC for applied mathematics in SciML. The workshop will specify the challenges and opportunities for increasing the rigor, robustness, and reliability of ML necessary for routine use in DOE science and engineering applications. In particular, the focus will be on identifying the unique ML needs and requirements of scientific applications. In terms of examples, these challenges may include rigorous analysis methods for developing and testing ML methods, understanding ML method approximation power, and bounding data and compute complexity of ML approaches. These challenges will be discussed in the context of existing methods with demonstrated use in scientific applications. We also seek to engage the attention of the world-leading ASCR mathematics and computer science communities in these new research priorities. The workshop will focus principally on the mathematical challenges in ML theory and application. These challenges will be discussed in the context of existing methods with demonstrated use in scientific applications.

This pre-workshop report describes the state of ML in 2017 as it relates to DOE missions and ASCR capabilities. The goal of the report is to identify the current strengths and weaknesses of ML for scientific discovery and help inform the workshop discussion on grand challenges and PRDs for SciML. To achieve this goal, the report is structured in specific sections:

- The "Motivation and Impact" section provides examples of how ML already is influencing DOE missions and identifies areas for future impact.
- The "Scientific Computing and Machine Learning" section offers a brief overview of the most popular ML methods with relevance to scientific discovery.
- The "Computational Foundations for Scientific Machine Learning" section describes the intersection between existing ASCR applied mathematics capabilities and current ML methods.

Definitions

The simplest definition of ML was provided in an apocryphal quote by Arthur Samuel, often attributed to his 1959 paper on ML for the game of checkers [6]:

> Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

This workshop will use a more detailed definition, similar to that offered in a review paper by Jordan et al. [8]. *Learning* is the process of transforming information into expertise or knowledge. *Machine Learning* is automated learning. The input information to ML includes training data and *a priori* knowledge or information (such as physical conservation principles). The output information from ML is "knowledge" or "expertise" used to make predictions or perform some activity. The *ML process* assumes the form of an algorithm that takes information as input and produces as output knowledge as just defined.

Motivation and Impact

Scientific discovery fundamentally relies on two pillars: data collection, which aims to create new measurements of complex phenomena using computations, experiments, and/or observations; and data analysis, which aims to extract new insights from the data. Historically, most of the effort has been on data collection, for example, starting with the drive to create telescopes to observe planets and ending with large-scale, multi-physics simulations to gain a window into otherwise inaccessible processes, such as supernovae. Yet, where early analysis techniques were as simple as linear regressions, modern solutions for both experiments and simulations are highly sophisticated combinations of mathematics, statistics, computer science, and a host of other disciplines. Nevertheless, even as the complexity of the analysis process has grown tremendously, it still has been largely driven by human expectations and hypotheses. Guided by their deep subject matter expertise, extensive experience, and well-honed intuition, scientists will develop hypothesis and customize analysis approaches to verify or disprove them. However, as we collect billions of particle collisions [20], simulate thousands of climate scenarios [21], or observe millions of stars [22], this existing paradigm is rapidly being stretched to its limits. No single researcher or even a large team can directly sift through such staggering amounts of data. Yet data reduction techniques, i.e., finding "interesting" collisions or "unusual" stars, may fall victim to various confirmation biases, where rare events outside of the current theory may be overlooked or misinterpreted. As the size and complexity of data increase, the likelihood of missing opportunities for breakthroughs also will escalate and may ultimately delay progress significantly. To address this challenge, many areas of science are moving toward more data-driven techniques that ultimately aim to substitute the need for prior assumptions and meticulously tailored hypotheses with massive data collections.

This data explosion is not limited to scientific applications. In fact, the digital revolution and the corresponding capability to log everything, from financial transactions to online purchases and social media interactions, has led to a revolution in ML. This revolution has been driven by the availability of large labeled data sets that are propelling the development of new ML methods. These methods allow computers to accomplish many seemingly complicated tasks, e.g., driving a car [23], playing checkers [6], or playing chess [24]. Especially in areas that require processing large volumes of data (a task not well suited for human brains), these techniques have produced stunning results thought to be unachievable merely a decade ago. However, adapting these approaches to scientific applications has proven challenging. In many of the most successful examples of ML, such as image recognition, system developers know the ground truth at least sufficiently well enough to check the results, often even while training the models. Almost by definition, the most interesting scientific applications of ML are those, such as cancer or high energy physics, where we do not know the answers beforehand nor are we necessarily able to easily verify the answer of an automated system. Consequently, scientific analysis solutions often cannot be validated directly by checking for success or failure. Instead, familiar questions from scientific computing are, for example:

- How reliably will a given algorithm work, e.g., for what type and quantity of data do we expect results?
- How robust is a certain solution, e.g., how might slightly different data or the addition of noise change the results?
- How rigorously have the assumption and underlying theories been defined and validated? For classical techniques, such as the analysis of PDE-based models, these questions lead to

familiar concepts, such as well-posedness, stability, numerical approximation, and UQ.

These questions all have led to a long history of relevant research and to reliable and robust outcomes from PDE-based models used in DOE applications. On the other hand, the equivalent concepts for ML-based models are not well established, and the lack of precise definitions and clearly expressed assumptions often leads to failure of ML-based methods. Finally, the ultimate goal of analysis is for a scientist to gain new insights, adding a human dimension to the problem. This requires both integrating the existing body of human knowledge into the ML approach and providing help for users to understand how a given approach works. In particular, the learned variables—or quantities extracted through a black box model—should be convertible to corresponding quantities from another box. Lagrangian mechanics express Newton's law in an arbitrary coordinate system; there should be a similar equivalence in the context of learning. Systems that generate intrinsic variables and enable translation and matching among hidden discovered variables would be even better. These variables are tautologically natural and become or translate to human knowledge.

Independent of these fundamental and somewhat abstract differences between ML, as seen in the media and by scientific data analysis, significant practical and technical distinctions also exist. First, rather than huge collections of rather small and simple data items (e.g., images), scientific applications usually deal with far smaller collections of much bigger and more complex data sets (e.g., time-dependent flow fields, spectra, or time series). This may invalidate assumptions about how densely a phenomena has been sampled and create computational challenges in managing and processing the data that are distinct from typical usage patterns. Even with these challenges, ML has the potential to significantly advance a variety of scientific areas and may fundamentally alter how we approach data analysis. For example, it is likely that before long, various experiments and simulations will no longer be primarily limited by what data they can collect but by how well they are able to analyze and quantify the data they generate as a way to self-assess and reorganize algorithms. However, to take full advantage of the combination of massive data collections and ML for scientific discovery, one must understand the current state of the art, where it may not meet the demand of various scientific applications, and what the key open research directions are to address the shortcomings.

Use Cases and Examples

One way to view potential use cases is by considering particular classes of problems. This approach is at the heart of decades of advancements in optimization: an explicit and clear realization of the diverse structure of different optimization problems classes (linear programs, integer programs, mixed-integer programs, PDE-constrained optimization, etc.) and the need for structure-exploiting techniques in each case [10, 11]. A complete statement of problem classes for ML is beyond the scope of this report (and is itself a topic that requires research) However, we note a few directions of initial thinking.

Simulation supports a broad range of activities within the DOE portfolio, some of which support scientific goals of discovery and understanding and others that support decisions through prediction, optimization, and UQ [9]. At the frontier of simulation-based science and engineering lies a common challenge among all of these use cases: how to make the task of evaluating a complex, expensive simulation model over a high-dimensional parameter space tractable and, importantly, how to validate the resulting evaluations.

This is an incredible opportunity for both ML and numerical simulations as there are massive amounts of numerical simulations that can be viewed as ground truth where the true model is known and for which we may develop purely empirical ML models to match the true dynamics. The resulting models do not have the true parameters embedded, but they discover them through observations of the simulations. This kind of environment confronts most issues of precision validation and trust.

Systems governed by PDEs and large-scale ordinary differential equations (ODEs) form a significant part of DOE's portfolio of target application problems [9]. Just as these systems require the development and analysis of specialized structure-exploiting methods for optimization, inverse problems, and UQ, realizing the benefits of ML will require similar recognition of problem structure. Within this general class, we also note several situations that may distinguish themselves in terms of ML modeling goals:

- The "truth" PDEs/ODEs are known but are too expensive to simulate directly, e.g., direct numerical simulation (DNS) in turbulence.
- The PDEs/ODEs are known but have unknown parameters, e.g., subsurface models.
- The PDEs/ODEs are an approximation of the truth with closure parameters, e.g., turbulence and force-field models.
- The PDEs/ODEs are partly known but also have unknown terms, e.g., biological models. This is where there is the highest potential for advancing the science of modeling.

Design optimization and *optimization under uncertainty* approaches seek to explore high-dimensional parameter space to determine optimal designs according to (possibly multi-objective) success metrics. *Sensitivity analysis* is used to explore parameter sensitivities around a solution (local sensitivity analysis), often in support of gradient computations for scalable optimization, and to apportion uncertainty in a quantity of interest (QoI) among uncertain inputs and their interactions (global sensitivity analysis), which in turn can be used for factor prioritization and factor fixing. *Uncertainty analysis* seeks to propagate the effects of high-dimensional uncertain parameters through a forward model to estimate statistics such as mean, variance, and probability of failure. These estimates may be used to support validated predictions, scenario exploration, and optimization under uncertainty. *Inverse problems* seek to estimate unknown and unobservable parameters from (often indirect and noisy) data. More generally, inverse problems provide a systematic framework for learning from data through the lens of models under both data and model uncertainty. In this sense, there are clear connections between inverse problems and ML, as well as a clear opportunity to exploit the considerable existing research base of large-scale inverse problems in complex physical systems. Finally, *regression modeling* especially of complex (e.g., non-scalar) multimodal data using DL-type approaches may lead to a new generation of surrogates that accounts for the inherent correlations between variables and provides predictions of experimentally relevant quantities (e.g., images, time series, or distributions).

Other uses of ML techniques in the high-performance and scientific computing communities include improving the performance of HPC machines. For example, MACORD [210] presents an ML-based silent data corruption detection framework for detecting silent errors in HPC machines. Tuncer et al. [211] used historical data on resource usage and performance counter data to diagnose performance anomalies of HPC applications. ML approaches also have been used to model the energy efficiency of modern HPC data centers [212].

Another significant growth area for ML in the DOE portfolio involves using ML techniques to improve existing paradigms, i.e., *multiscale modeling*, and the convergence of experimental and simulation data. A wide range of applications use multiscale modeling to couple highly accurate but computationally expensive simulations, e.g., first-principles molecular dynamics, with coarse-grained approximation to reach longer simulation times or consider larger systems. In such cases, it is crucial to understand when and where a high-fidelity simulation is required, how to interpolate between existing results, and how to understand the "manifold" of high-fidelity solutions. All of these questions can benefit greatly from improved ML techniques. The second aspect, sometimes referred to as *cognitive computing*, involves merging of simulations and experimental results. How can one optimally combine the knowledge gained from (potentially large numbers of) imperfect simulations with sparse experimental data while taking expert intuition into account? Emerging concepts in ML, such as transfer learning (e.g, training a model on simulation data then refining it based on experiments), have the potential to revolutionize how scientific computing is used.

In the remainder of this section, we briefly discuss a few concrete application examples and use cases. These examples are not intended to be an exhaustive list or indicate DOE priorities. Several other uses cases are presented in the numerous review articles on ML [8, 99, 213–217]. Instead, they indicate relevant examples drawn from the authors' collective experience to cover a broad range of potential science areas.[*]

Turbulent Combustion. Modeling turbulent combustion processes is essential for designing more efficient power generation systems and next-generation transportation systems. Because of the inherent flow and chemistry complexities and range of spatial and temporal timescales involved in combustion, computational modeling has played—and will continue to play—a key role in understanding the underlying physical phenomena. Furthermore, computational modeling provides a rigorous platform on which to conduct parametric exploration and optimization in support of design and decision making.

The fundamental challenges in turbulent combustion derive from the complexity of 3D turbulent flow, the complexity of oxidation chemistry of complex fuels, and the large range of length and time scales in high-Reynolds-number turbulence [218]. Further, experimentally observable minimum time and length scales are rather large compared to those present in turbulent flows of practical relevance and similarly large compared to the scales accessible to reliable physics-based models on current computational resources. Advancing the state of the art in this context requires synergistic melding of physics-based modeling of the governing equations and data-driven modeling of experimental data. DNS provides highly reliable physics-based models—3D time-dependent Navier-Stokes equations—for turbulence. These must be coupled with chemical kinetic mechanisms and models for thermodynamics and molecular transport; realistic equations of state; and, depending on the system at hand, models for liquid spray, radiation, and soot. On the other hand, these DNS models are intractable for use at any reasonable system scale. Thus, they do not allow design or parameter exploration. Instead, turbulence modeling is often used, e.g., large-eddy simulation (LES) [219] or Reynolds-averaged Navier Stokes (RANS) [220], together with data-driven modeling closures that attempt to model the unresolved scales and missing physics. Existing approaches to combine both ends of this modeling spectrum rely on expert-driven heuristics and struggle to adequately account for the very high dimensional parameter spaces and complex multimodal data provided by experiments.

With high dimensionality and complex data handling at the core of ML techniques in general, turbulent combustion research is poised to significantly benefit from research in and the application of ML. In particular, better inverse modeling and feature selection could improve data-driven model closure [34, 35]. Better sensitivity analysis and UQ could provide new data-driven strate-

[*] ...or author availability on a sunny day.

gies to characterize model error [27], and new regression techniques may result in more powerful surrogates of more tractable quantities, such as images or spectra. In all cases, it is clear that ML will not replace decades of research in principled physics-based approaches; rather, it can bring a toolbox of methods to enrich, improve, and accelerate current modeling approaches.

Department of Energy Biomedical Partnerships. DOE has a long record of making scientific contributions across the spectrum of biological sciences. In particular, SC has been a leader in genomics with the initiation of the Human Genome Project. Currently, the Office of Biological and Environmental Research is tasked with the mandate of "understanding complex biological and environmental systems by coupling theory, observations, experiments, models, and simulations" [221]. All of these challenges stress test the latest capabilities in ML. Recently, DOE has initiated a high-profile biological research project, partnering with the National Cancer Institute (NCI) [222]. For the DOE/NCI partnership (some of which is supported directly by SC), there are three targeted areas: pre-clinical models predicting drug response, understanding the structure and dynamics of RAS/RAF-related proteins, and development of predictive models of patient health trajectories. All three of these areas will stress test the latest methods in ML, including natural language processing (NLP). Other examples of application usage are described in Refs. [223–226], where ML techniques are leveraged to populate relational databases with information from unstructured data sources. These methods are further optimized in [227] to leverage the effectiveness and efficiency of statistical inference and ML for difficult extraction tasks while not requiring users to directly write any probabilistic inference algorithms. The DL approach has had many successes, but as discussed elsewhere in this document, success in DL will require a deeper understanding of the mathematical foundations.

The application of ML methods to biological and health sciences is well established and has benefited from the standard suite of conventional tools, including support vector machines (SVMs), random forests, and graphical models (among others). The scientific literature about the biomedical applications of these methods is enormous. Multiple crowd-sourcing competitions have led to creative ML solutions for building predictors. Entire industries and institutes have grown up around providing the ML needs for biosciences, including NLP. With the growth of DL methods, the landscape is even more complicated.

Despite the proliferation of papers about ML in health and genomics, the application of ML to biological sciences still has considerable room for growth. To reach its full potential, there are several challenges that must be overcome. First, despite the often stated existence of "too much data," there actually are few independent samples, especially for problems as complex as cancer. As a result, it can be difficult to calculate the accuracy (error rate) of the classifiers/predictors and to know how well they will perform in real-world environments. Given the aforementioned complexity, it is likely that classifier performance would be greatly improved by integrating as much domain knowledge as possible. Therefore, advances in transfer learning and the integration of multiple types of data are required. Finally, improved computational algorithms are required to determine what data (i.e., experiments) to carry out next.

ML has the potential to improve health care, reduce unnecessary treatments, and better target limited resources. Provably good methods will shorten the overall development time (from lab to bedside); reduce experimental burden; and present the possibility to save money, time, and resources.

Understanding RAS-Related Proteins. As mentioned earlier, understanding RAS/RAF-related proteins is one element of the DOE/NCI collaboration. A mutated RAS protein is found in more than 30% of all known cancers and 95% of all pancreatic cancers. Nevertheless, the exact mechanism by which RAS is involved in cancer progression is unknown, and it has remained untargetable with existing drugs. The way RAS interacts with different cell membranes likely depends on the local structure of the membrane, as well as the presence or absence of other proteins (e.g., RAF), yet it happens on time and length scales largely inaccessible experimentally. However, the next generation of molecular dynamics promises predictive simulations of realistic cell membranes interacting with RAS molecules that for the first time will allow researchers to observe the relevant processes in detail.

Modeling a realistic cell membrane involves hundreds of thousands of different lipids and millions of atoms [228]. Even the next generation of supercomputers only will allow rather short simulations (at the microsecond scale) at full resolution [229]. However, biological processes often are stochastic in nature and relevant local structures may take much longer timescales to evolve. This may make the biologically relevant events, i.e., a particular RAS-membrane interaction, exceedingly rare with respect to the *in silico* timescales. One solution is to explore many different initial conditions in the hope of observing an unknown phenomenon. The relevant parameter space includes both lipid composition and the spatial distribution of lipids, which is impossible to explore fully. Furthermore, even given unlimited computing resources, it is unclear which events may be important and what configurations are "special."

As computational resources grow, molecular dynamics simulations can produce massive numbers of lipid distributions and RAS configurations. Without a clear understanding of which configuration might be biologically relevant, identifying these currently relies on brute force statistics. Unfortunately, even the massive data collections being created are not remotely sufficient to cover the parameter space reliably. Instead, to find the relevant outliers, new unsupervised learning techniques are needed to identify and classify lipid compositions and spatial distributions. While this has been successfully demonstrated to identify protein conformations [230], applying the same concepts to unordered collections of millions of atoms will require new classes of learning algorithms. The same ideas may enable a multiscale modeling approach that promises to reach biologically relevant timescales at atomic resolution—assuming a system capable of classifying membrane-RAS configurations into biologically equivalent, set, full-scale atomistic simulations could be restricted to explore yet unknown configurations while using continuum approaches to advance much larger systems across longer timescales than previously possible.

Materials Design. Materials science also has seen the rapid adoption of ML, from modeling crystal structures to materials discovery [231–234]. For example, combining ML automation with crowd-sourcing for automatic data extraction for materials science applications has been studied [235]. In recent years, some of the greatest advances in areas such as batteries, solar cells, or carbon capture are due to new materials being discovered with new properties. The traditional approach to materials science is based primarily on an intuition-guided experimental approach, where a large number of new compounds are synthesized and tested. This is time-consuming and costly, which severely limits the number and types of materials that can be explored. More recently, first-principles molecular dynamics codes have been used to predict material properties. While these may be more flexible and easier to set up than physical experiments, the corresponding simulations remain too costly and the parameter space too big to reliably explore all possible materials. The holy grail of materials science is to leverage the ability to simulate virtually arbitrary materials to quickly design new compounds custom tailored to exhibit desired properties, e.g., certain band-gap, dopant concentration, melting point.

Even accounting for the rapid growth in computational power, the parameter space of possible materials is too high dimensional (i.e., different atoms, arrangements, and/or dopants) to make exhaustive exploration feasible. Instead, the goal is to simulate some number of materials covering a range of input parameters and from these predict which other combinations are promising to try next. Fundamentally, this can be thought of as a *regression* or *surrogate* modeling problem. However, the dimensions are high, many input parameters are categorical, and the outputs highly nonlinear, which makes solving such challenges difficult.

Materials design touches on a number of core ML approaches and significant advances are underway. Under a large umbrella, the Materials Genome Initiative [236] brings together a diverse group of researchers, employing various tools to accelerate design. Another aspect of "searching through a space of possibilities" is to identify property-determining substructure features. Searching material databases for geometrically similar arrangements of framework atoms can allow one to screen databases an order of magnitude larger than what has been examined [237]. One of the central challenges is to determine adequate mathematical representations, whether they develop adequate surrogate models in high-dimensional parameter spaces, identify appropriate similarity metrics, measure "closeness," or discover how to link similarities with predictive properties about functional performance. One advantage to ML approaches is they may more easily lend themselves to predict multivariate and/or multimodal outputs [238], e.g., jointly predicting multiple output properties, some of which may not be scalar: expected correlations between outputs may represent a crucial constraint on the problem that can make it tractable. Determining how to build in correct physical constraints will be a crucial part of making these approaches work. Finally, connecting functionality and manufacturability to the search and design process will likely require a coupling of experiment, advanced simulation, ML, and human intuition.

Multiscale Modeling. A common problem in many application areas is that while highly accurate predictive models (e.g., first-principles molecular dynamics simulation) exist, these typically cannot cover the time and/or length scales required to address high-level challenges (e.g., optimize an engine, simulate a cell, or deform a material). One attractive solution is to build multiscale approximations, where a cheaper *coarse-grain* model is used to cover the macroscales required, while a more accurate *fine-grain* model is used only when necessary or to inform parameter settings for the coarse grain [239–243]. A typical example is deforming materials, where fine-scale models are used to compute the stress tensor of the material given the current stresses or material hardness [244, 245]. However, evaluating stress with the fine-scale model at every grid point of the finite element grid is computationally infeasible. Instead, there is a need to efficiently cache fine-scale evaluations to recompute stresses only in configurations too dissimilar from previous evaluations. Similar ideas apply to the previously discussed RAS studies, where lipid membranes are evolved at either continuous or atomistic scale or the combustion problems where fine-scale kinetics are only evaluated selectively.

These ideas presents a number of challenges. First, the parameter space defining a fine-grain simulation is often high dimensional, may include parameters on different units and scales, or may not lend itself to any explicit encoding (e.g., the lipid bilayers already discussed). Adding to this challenge is the likelihood that not all combinations of parameters are physically meaningful: the problem resolves to sampling an embedded sub-manifold in some high-dimensional space without an inherent metric. Similar challenges exist in the output space, which may be as simple as a single stress tensor or as complex as a molecular dynamics simulation. Nevertheless, it is crucial to define a meaningful distance in this space to evaluate metrics, such as those that inform when two fine-grain simulations should be considered similar enough to not require an additional evaluation. Finally, avoiding unnecessary evaluation of the fine-grain model implies an interpolation scheme that allows for accurately predicting results in the vicinity of existing data.

Depending on the specific application, addressing the challenges of multiscale modeling can be virtually arbitrarily complex; however, many of the fundamental problems are directly related to core research areas in ML. For example, latent spaces, such as those created by auto-encoders, can provide both dimension reduction and a metric in otherwise abstract spaces [214, 246, 247]. Furthermore, an appropriate latent space will naturally express correlations between different variables, i.e., encode any embedded lower-dimensional structures, and may lead to more efficient sampling and better surrogate models. In general, dimension reduction [248–250], metric learning [251], and multivariate, multimodal modeling [251] are all central to ML. Any advances in these areas could significantly improve the performance of multiscale approaches.

Image Analysis: Detection, Recognition, and Classification. The ability to analyze experimental and simulation data automatically is essential to support extraction of scientific information from high-throughput data generators. On the experimental side, data are being collected at faster rates and higher resolution from sophisticated scientific facilities supported by DOE, including the synchrotron light sources, nanoscience centers, and genomic and biological institutes. Obtaining information from these data can help classify structures, reveal patterns, connect the output of one experiment to another, and guide which experiments to perform next. On the simulation side, with the help of extreme-scale computing, high-resolution calculations from varied fields, such as climate modeling, combustion simulations, and fusion calculations, are poised to predict time-evolving physical variables with tremendous fidelity. Detecting, recognizing, and measuring structures from these data have the potential to reveal correlations not immediately apparent from pointwise (in space and time) quantities, track key integrated variables, identify outliers and rare events, and provide mechanisms to compress data when isolating crucial information.

Analyzing image data typically is done manually, which is far too time-consuming to deal with the current onslaught of data, let alone the coming acceleration of data rates from experiment and simulation. Classical image processing aims to automate much of this process by using an assortment of techniques, ranging from thresholding and scaling to more advanced techniques, such as anisotropic diffusion, wavelet transformations, level set methods, Markov random field methods, and other graph-based methods for image segmentation. These methods are powerful but often require tremendous customization and tuning for specific applications.

One of the most successful applications of ML has been in image and video analysis, particularly for natural images [252, 253]. The methods combine an array of techniques including DL using architectures such as CNNs and statistical methods such as cascaded random forests or SVMs. Applying these methods to scientific data poses some significant challenges, partly because of the lack of reliable training data and the need to prospect for scientific information engineered into the network architecture so that physical quantities can be used to monitor and drive the learning/training process. An important component when tackling large data sets is the scalability of ML methods to very large systems as in Kurth et al. [254].

Need for Robust Scientific Machine Learning

Already, ML has made a significant impact in a variety of high-regret applications, including clinical diagnoses [77, 78], security [79, 80], metal fatigue in aviation [81], and environmental contamination [82, 83]. An excellent review of some of these applications and associated risks are provided by Shneiderman [84] and references therein. Such applications exemplify the importance of robustness and rigor in ML to minimize the risks associated with its use. These applications also point to the need for explainability in ML predictions [85]. The potential negative impacts of misused ML has led to calls for policies to "anticipate, monitor, and retrospectively review operations" for managing algorithms in high-impact applications [84], as well as to initial research in exploring ways to manage algorithm behavior [86, 87].

With its emphasis on well-defined analytic processes for stability and error analysis, applied mathematics can provide a mechanism for developing ML methods with robustness. Such robustness will address issues related to sensitivity to training set size, choice of data in training and test set, numerical instability in the learning algorithms, scalability, and parallelization with complicated and heterogeneous hardware. While the need to investigate these issues has been expressed since the early days of ML and AI [89–91], much research is needed, for instance to set scientifically based rather than heuristic guidelines on acceptable classifiers that process experimental data from beamlines or predict failure in major components. It is also extremely unlikely that existing mathematical methodology will extend automatically to ML. For example, classical linear algebra libraries are designed to optimize computations with sparsity structures arising from discretizations of PDEs, while ML needs are likely to be quite different. Similarly, classical von Neumann stability analysis of numerical approximations is unlikely to be adequate for establishing stability of complex learning schemes. Thus, complementary research in applied mathematics methodology is necessary.

Need for Interpretable Scientific Machine Learning

Another often overlooked aspect of choosing a model is the *interpretability* of the results. Traditionally, physical understanding has been the bedrock of modeling. A user's confidence in model predictions is directly linked to the conviction that the model accounts for the right variables (e.g., temperature, pressure, or material density); the right parameters (e.g., inflows or reaction rates); and, most importantly, the right physical laws (e.g., heat-mass balance, energy, or Arrhenius mass action kinetics). Consider something as simple as principal components used to transform the data. These *data-driven* variables (linear combinations of the observables) may yield quantitatively better results, such as lower regression errors. However, the corresponding physical laws cannot easily or intuitively be expressed in terms of these "strange" combinations. While the equations can be adjusted to accommodate a change of variables, the *understanding* of the model is lost. This problem is greatly exacerbated for nonlinear models (e.g., nonlinear dimension reduction [250]), and models such as DNNs that become entirely opaque black boxes, which even their creators may not be able to fully explain.

With the success of so-called *DL approaches* in areas such as computer vision [70, 71], there has been a push toward relinquishing the practical human constraint of understanding a model. Instead of focusing on what variables physically mean and which physical laws are driving their evolution, the analysis paradigm is shifting to trust in the correctness of data mining algorithms. These algorithms typically take in a finite set of physical observations and return predictions, classifications, or other forms of integrated results.

Effectively, serious thinking on choosing variables and understanding mechanisms is replaced by carefully designed data-driven ML approaches and properly calibrated assumptions on smoothness, data distributions, etc. In many of the most visible examples, such as self-driving cars [23] and face recognition [255], it can be argued that the lack of interpretability is an anthropic rather than a mathematical problem. However, this assessment crucially relies on the fact that the correctness of, for example, a face-recognition algorithm is easily checked, mistakes are typically obvious, and massive databases of examples can be accumulated for model testing.

Unfortunately, in many scientific applications, the "correct" results are not known. In fact, ML is being applied precisely to better understand what the right solution should be. In this context, many of the most prominent ML tools are difficult to apply to scientific computing problems. Consider the simple example of a scatter plot, e.g., showing the distribution of some high-dimensional point cloud with respect to two of its dimensions. A so-called *axis-aligned plot* (that uses two of the original variables as x and y axes) is easy to interpret; however, it is well known that such plots are bad at representing more complex structures. The traditional example is that of the "Swiss roll," a two-dimensional sheet rolled into a spiral. Any axis-aligned projection will result in artifacts and severely misrepresent the local neighborhoods of the points. To address this problem, a host of dimension reduction tools exist to create "better" embeddings [249, 250]. In our example, one can unroll the sheet and produce a perfect (relative to the local neighborhood structure) projection. However, the resulting plot no longer supports any meaningful axes labels. In the context of scientific applications, such an embedding may be useful to highlight clustering behavior. However, without the ability to relate directions in the plot to native attributes, users typically have great difficulty connecting the structure in the plots with their physical intuition.

In general, tension exists between increasingly complex models to improve results on particular data sets and the need for users to interpret the models to derive new insights and conclusions. This challenge has been widely recognized, and new programs have started to search for solutions [72]. Nevertheless, we expect that concerted and long-term research efforts will be required, spanning the entire gamut of integrating physical constraints into black box models to make them more dependable, developing new exploration and visualization approaches to interpret complex models, or deriving theoretical guarantees on model quality or consistency.

Scientific Computing and Machine Learning

This section provides a brief overview of popular ML methods with particular relevance to scientific computing. A number of reviews provide more in-depth discussions of these and other methods [8, 217, 256, 257].

Supervised Machine Learning Methods

Supervised ML is perhaps the area with the deepest connections to statistics and UQ in computational science. The training phase of supervised ML is equivalent to model calibration and parameter estimation in the physical sciences, while the testing/prediction phase is similarly present in predictive model testing with new data.

Regression is commonly used in both contexts, whether the forward model is an SVM [258, 259], a NN, or a physical system model with a governing set of differential equations. Statistical methods are used in physical systems modeling for inference of uncertain parameters or estimation of parameters of model surrogates, whether polynomial chaos (PC) expansions [260] or GP models [261]. Similarly, training ML forward models, e.g., NNs, with noisy data often is done in a probabilistic context, e.g., Bayesian neural networks (BNNs), estimating network weights with quantified uncertainty in the form of Bayesian posterior probability distribution functions (PDFs) [262]. Early work in this area [263–265] has found renewed popularity [266, 267]. A significant challenge in BNNs involves scaling the probabilistic approach to large data and network scales typical of CNNs [268, 269] and DL models [70, 270–274]. One practical approach to Bayesian DL [275] is potentially via variational inference (VI) [276, 277] and associated approximate procedures such as dropout VI [278]. Recent work suggests that while dropout VI may be useful for practical estimation of uncertainty in ML, it can severely underestimate model uncertainty [279], motivating ongoing work to improve uncertainty estimates [280].

There are opportunities for adapting these advances to the calibration of physical systems models. For example, there are potential advantages in adapting ML training methods, such as back propagation [281–285] and stochastic gradient descent [286], to the calibration of high-dimensional physical systems models. At the same time, there are potential advantages in adapting a range of advances in statistics and UQ, such as for statistical inference in high-dimensional spaces [287–290] and Bayesian non-parametrics [291–296], to the challenges of training Bayesian deep networks. Of course, computational challenges abound with learning for Bayesian networks [297]. Moreover, choosing good priors is a significant challenge in general, particularly in nonparametric models [298]. Typically, recourse is made to approximate inference algorithms. Further, frequentist algorithms generally are useful for solving Bayesian problems or in providing understanding the Bayesian model. Often, a Bayesian-frequentist middle ground is useful, where a Bayesian model is coupled with frequentist model-checking techniques [299].

An extensive body of research has established that NNs with quite simple structures (i.e., only one hidden layer) and wide variety of activation functions (such as the class of sigmoidal functions that includes the logit and hyperbolic tangent activation functions) can arbitrarily well-approximate multidimensional continuous functions [300–302]. Furthermore, by using smoothing techniques, these results can be extended to discontinuous (classification) tasks. Historically, these results have focused on the unrealistic “infinite” data limit; however, recent results have been providing quantitative upper bounds on the approximation error [301–304]. In particular, these techniques provide explicit error bounds (depending on the activation function and the smoothness of the target function) for interpolation over the space of NNs given regularly spaced samples from the domain. However, for many ASCR applications, the “curse of dimensionality” precludes gathering sufficient data to apply these results. In addition, these results offer little insight into NN behavior when extrapolating beyond the given data, which is particularly relevant in applications such as climate modelling and high-energy physics. Perhaps, the best way to understand the mathematical challenges presented by the state of DL is via analogy to polynomial approximation methods: while there currently exists an NN analogue of the Stone-Weierstrass theorem, what is missing is an analogue of smoothing splines and Taylor series that will allow for reasonable interpolation and principled extrapolation from observed data.

Unsupervised Machine Learning Methods

Unsupervised ML methods refer to approaches designed to learn features from unlabeled data. Common tasks for this branch of ML include: feature learning, dimensionality reduction, visualization, outlier/anomaly detection, and variable selection.

In contrast to its supervised counterpart, unsupervised learning lives at the opposite end of the data spectrum, where none of the training data are labeled (i.e., features without labels). Semisupervised learning lives between these two extremes, where some of the data are labeled and the remaining—usually much more—data are unlabeled. Despite the fact that the data points do not come with labels, they still can carry considerable information about the feature distribution (marginal of the feature-label distribution) and can greatly benefit the construction of effective learning algorithms.

Not surprisingly, unlabeled data are often considerably cheaper and easier to obtain than labeled data and, therefore, can be more prevalent than their labeled counterparts. As a result, it is vital to have a robust mathematical foundation for using these types of abundant data most effectively. Still, considerable open problems on the mathematical foundations of unsupervised methods remain, especially where prediction performance is concerned. As such, the need for effective unsupervised learning algorithms with known/guaranteed performance is quite valuable.

Traditional ML techniques involve identifying a core set of features from input data, such as in PCA [305]. Other methods, such as stochastic neighbor embedding (SNE) [306], accomplish dimensionality reduction while still preserving "closeness" between samples, the t-SNE variant is particularly useful for visualization [248].

More recent advances in unsupervised learning have come with the advent of DNNs. As described by Bengio [307], this kind of approach lends itself to problem spaces where there are few or no labels for classes of interest. Unsupervised ML methods can be used to match items between two unlabeled data sets [308]. Additionally, unsupervised approaches can be used as a "warm start" for supervised problems, where more unlabeled versus labeled data exist. Traditional autoencoders, in particular, operate by learning an encoder-decoder structure, which learns intermediate representations of the input data [309]. Recent advances known as *variational autoencoders* [310] have honed in on the generative properties of these methods. Furthermore, other recent approaches have used the reconstruction error of the output of these autoencoders as an unsupervised outlier or anomaly detector [311].

Broadly speaking, unsupervised ML methods are related to probabilistic and statistical methods for learning a model of the data. These include filtering, latent variable, Gaussian mixture, hidden Markov, and dimensionality reduction methods [312–316], as well as methods for building stochastic generative models for the data [317–319].

One challenge in dealing with unlabeled data is rigorously constructing and efficiently calculating metrics for algorithm performance. As a result, unsupervised learning methods are often subjective. Consider the representative class of unsupervised techniques involving clustering. The goal of clustering is to separate individual data points into groups. Within groups, there is some similarity of the points, while the points are less similar between groups. Questions arise regarding the "optimal" number of clusters. There is subjectivity in determining this number. Some theoretical work exists, but more is needed. Hence, a challenge for clustering and other unsupervised learning methods is the need to provide more rigorous foundations [320]. For more discussion of this problem, refer to recent work by Dalton and Dougherty [321, 322].

Commonly, unlabeled data are not independent and identically distributed (random variables) independent and identically distributed (random variables) (i.i.d.). Instead, they are data of opportunity, and the limited theory available does not apply in real-world situations. A theoretical understanding of unsupervised methods non-i.i.d. or non-randomly obtained data is lacking. The DOE workload often deals with extremely complex systems. For these systems, the labels may be difficult to obtain, and, where such labels exist, the number of samples may be quite small. How to effectively use unsupervised and semisupervised methods with small samples is another challenge.

Reinforcement Machine Learning Methods

Reinforcement learning refers broadly to a set of methods where no training data are used, and no suboptimal actions are corrected in an explicit manner. Popular examples include q-learning and temporal difference, which learn policies (actions) based on some notion of reward. The diverse set of methods for reinforcement learning forms the basis for many of the recent trends in modern AI, such as self-driving cars [323], autonomous robots [324], and game playing (e.g., poker [325, 326], Gō [24, 327, 328], chess [329], and video games [330–332]). However, the design of reinforcement learning methods is currently more of an art than a science and lacks rigorous foundations that would allow for applying reinforcement learning to complex problems. For example, there are significant opportunities for reinforcement learning to be applied for scientific discovery via experimental design and other search methods in the context of high-throughput experiments. In addition, there are connections between reinforcement learning and traditional numerical techniques that have not yet been fully explored [333]. However, many of the more well-known examples benefit from a large set of training data relative to the complexity of the problem at hand.

In the context of reinforcement ML, Markov decision process (MDP) and stochastic dynamical programming methods are often used, generally seeking a trade-off between exploration and exploitation [334]. There are good examples of probabilistic/statistical methods being applied. For example, Bayesian methods have been employed in reinforcement learning [335–337]. Other examples include data-efficient probabilistic methods for small noise systems [338]. For the large noise regime, methods such as [339] are being developed that rely on partially observed MDPs and Bayesian filtering based on assumptions of linearity (or small nonlinearity). Ongoing challenges involve dealing with more general belief distributions and policies as well as non-Gaussian noise and partial observations.

Current advances in reinforcement learning have come in conjunction with learning policies for deep networks. In particular, there have been improvements in areas using common architectures and learning algorithms for various problems with the deep Q network (DQN) described by Mnih et al. [332].

Some pitfalls of traditional deep reinforcement learning include how sensitive they are to parameter tuning, such as exploration and rewards, to facilitate convergence. This has been addressed, in part, by another recent method, trust region policy optimization (TRPO) [186], which has been shown to reduce the overhead of tuning these parameters and produces reasonable results when compared to DQN. Other approaches include asynchronous advantage actor critic (A3C) [340], which have shown to stabilize the learning process. Current research in this area has focused on using evolutionary strategies [341] as an alternative to MDP-based approaches, such as policy gradients.

Another interesting avenue of current work in this field has been in the application of meta-learning [342]. This has provided a mechanism for these networks to continuously adapt and improve to new situations, learning better policies. Further, meta-learning has demonstrated the ability to “extrapolate” between similar problems, potentially mitigating the issues posed by limited/small sample data [343–346]. Other state-of-the-art techniques include competitive self-play [24, 347], where agents play and adapt against other agents. In addition, learning through imitation has been shown to be quite useful in cases such as robotics [343, 348].

Several opportunities exist in the arena of modern deep reinforcement learning approaches. While environments with discrete action spaces have seen rapid improvement, work in continuous action spaces [349, 350] has not been as fruitful. Another issue still plaguing reinforcement learning is exploration. Current approaches force exploration by applying perturbations to the agent’s actions to force experimentation. While this works well for discrete action spaces, agent exploration of large continuous spaces can be costly.

Other Machine Learning Methods

The area of semi-supervised learning [351, 352], where only a subset of the data are labeled, is of practical relevance [353] and offers interesting statistics and UQ challenges and opportunities. In this context, similarity graphs that identify similarity between labeled and unlabeled samples are useful. Semi-supervised learning on graphs [354] has been cast in a Bayesian framework to capture uncertainty [355]. Semi-supervised learning methods include transductive support vector machines (TSVMs) [356–358], where unlabeled test data are labeled by the SVM classifier to maximize the margin, and related algorithms such as a spectral graph transducer [359]. However, self-training is error prone, and both TSVMs and graph-based methods are difficult to scale [247]. Probabilistic formulations have been employed to overcome these limitations [360, 361].

Another important area is active learning [362, 363], where improved accuracy may be achievable with smaller numbers of labeled training points if the algorithm is allowed to actively choose learning data in a greedy manner that maximizes information gain, placing training data points where uncertainty is maximal. Active learning is motivated when labeling data is challenging or requires significant time and/or expense. Active learning performance depends on having accurate estimates of uncertainty in model weights as well as on the level of noise in the data. These challenges have been dealt with in the difficult context of *cold-start* active learning, using a Bayesian framework [364].

State of the Art in Model-Free Methods. A model is a closed-form formula (an algebraic equation, a deterministic or stochastic ODE or PDE, possibly a nonlocal/integral one) that expresses relations between physical observables and allows us to make predictions. It could be a simple formula based on physical understanding (e.g., Newtonian gravity) or data driven, such as a large empirical response surface saved as a database or represented through generic basis functions like Gaussians or DNNs. All models, physical or data driven, arise from observations and measurements of the physical world. We first understand (a) what the relevant physical quantities are in terms of which to describe the phenomenon; (b) the interactions between these physical quantities, embodied in physical laws (e.g., chemical kinetics); and finally (c) we put the two together to obtain a closed-form expression that can be used to make predictions. Thus, human understanding of the right variables and relations between them is an especially important component of modeling/prediction.

Different paths exist from observations to predictions. Taylor series can be used locally to approximate smooth functions, giving convenient local surrogates of the (unknown) model equations. In this context, various versions of Kalman filtering, autoregressive moving average (ARMA), nonlinear autoregressive moving average (NARMA), or NN models can be used for *completely* data-driven predictions [365, 366]. In the early 1980s, Moore showed how singular value decomposition (SVD) can lead to I/O-based balanced realizations of linear systems solely from data [367]. Iterating the learned *black/gray box* can allow bootstrapping predictions into the future.

A well-established version of such a path from data to predictions, starting in meteorology with Ed Lorenz in the 1950s and blossoming in the 1980s, involved using PCA to obtain good collective reduced variables, known as proper orthogonal decomposition (POD) in the fluid mechanics literature [368]. If the behavior is low dimensional, we can use SVD to determine the right few variables and subsequently project the equations in the subspace they span,which is a POD-Galerkin method. If the detailed PDEs are lacking, ML can approximate the dynamics in the reduced subspace. Alternatively, one can leverage matrix-free methods to do *equation-free* computation. Manifold learning techniques that, starting with data clouds of observations, detect and provide parsimonious parameterizations of the intrinsic low dimensionality of the data (giving good reduced variables), become a hugely enabling technology for modern data-driven modeling. The goals may vary and include parsimoniously estimating the unavailable equations, usefully filtering/denoising the data, quantifying uncertainty, factoring our symmetries (translational, rotational, or scale invariance), or performing data fusion (to name a few).

These manifold learning tools (hinted at by bottleneck NN autoencoders in the 1980s) exploded with the publication of Isomap [369] and locally linear embedding (LLE) [370] in 2000, followed by developments such as diffusion maps, Laplacian eigenmaps, vector-diffusion maps, and other kernel-driven methods that have paved the way to *nonlinear* data-driven system realizations. Such manifold learning algorithms (often based on harmonic analysis, the Koopman operator, or Hilbert space reproducing kernels) do not *pre-accept* that the observations presented to them is the *right* one. Instead, they afford free movement across all possible deformations/diffeomorphisms of the observed response surface. Then, the question becomes: which is the *best* representation to work with? Until now, human understanding has been a tremendous constraint in modeling. We must be able to explain what we are modeling to each other and write down physical laws in a few lines so that the knowledge can be parsimoniously conveyed. This crucial precedence of *human* or *physical* understanding begins to recede as we increasingly rely on learning the intrinsic geometry of data and taking data-driven paths toward prediction.

Computational Foundations for Scientific Machine Learning

There are several aspects of existing ASCR applied mathematics research that can be extended to ML. This section outlines some of these potential overlapping areas between ASCR applied mathematics and ML.

Rigorous Analysis

While ML opens up exciting new opportunities for both applications and research, there are dangers in applying ML without an adequate understanding or awareness of possible pitfalls and technical limitations. This issue has particular relevance to areas of interest to ASCR, where an extremely high level of scientific rigor is maintained. An application that has adopted techniques from ML without a firm scientific basis risks compromising the scientific integrity of the entire application. Nevertheless, although the mathematical foundations of traditional ML are relatively mature, the current pace that new approaches are being developed and applied to expanded areas means the gap between theory and practice is widening rapidly. Additionally, the quickly growing field of DL lacks the robust mathematical foundations of more established approaches.

A fundamental concept with direct bearing on the fidelity of ML is the extent possible for a machine (or any other "learner") to learn from training data in a given domain. The notion of probably approximately correct (PAC) learning essentially is a mathematical framework for answering such questions [371]. Typically, one seeks to provide quantitative statements about the resulting model's probability for being capable of providing predictions within some guaranteed error bound. For instance, one may be able to provide estimates about how the amount of data needed to develop a model/forecast that meets a given tolerance scales with the data set size or, more appropriately, the information content of the data [89]. Such estimates, even when available, are often restrictive in the required assumptions and will not apply to the vast range of mission-critical DOE applications.

A related but quite different question concerns the selection of the most appropriate hypothesis class for a given application in terms of incorporating prior knowledge, physical principles, and pure data. Structural risk minimization is a learning framework that makes it possible to express preference for certain hypothesis classes in a mathematically precise way [372]. In particular, this paradigm can support principles that consider the complexity of a hypothesis space, such as the minimum description length (MDL) principle, along with other methods of incorporating prior knowledge and data.

The ultimate goal of applying ML to ASCR applications is to provide a decision maker with two pieces of information:

- prediction or information as to what may result from taking a particular course of action
- quantitative of the level to which the predictions can be trusted.

For example, one may want to make decisions about the parameters used to manufacture an aircraft wing. In principle, the answer or decision takes the form of a fixed set of numbers that are predicted to suffice. However, in practice, there are many sources of uncertainty, including not only the data used to inform ML but also issues in the sensitivity of the underlying model space and the effect on the extent it is even possible to make predictions. These issues contribute to uncertainty in the conclusions suggested by ML and it is vital that the decision makers are provided with information, ideally quantitative, regarding the uncertainty.

Of course, these issues are not peculiar to ML and are ubiquitous in science. However, the key difference is that in mainstream science, the underlying models are expressed in an explicit form amenable to analysis using techniques that have been developed over decades and are relatively well understood. Moreover, the assumptions that underpin a particular model usually have been debated and refined, and the resulting model's limitations have been explored extensively. In mainstream predictive computational science, the sources and effects of discretization and numerical errors have been equally well studied over many years and are widely appreciated and understood.

The current rapid pace at which ML is being applied to an ever-widening domain of applications means that such questions are at best poorly understood and at worst not even being asked. Consequently, there is an urgent need to develop mathematical tools that can provide the same levels of confidence and understanding to predictions that emerge as a result of applying ML as the levels of confidence and understanding expected from more traditional approaches.

The gaps in the current levels of understanding for basic components of ML are widely accepted. Fields medalist Martin Hairer (Imperial College) recently stated [373]:

> "Neural networks are interesting because we couldn't imagine them existing 20 years ago. Neural networks are the artificial information processing systems that allow for innovations like self-driving cars. Although we can make them, we don't exactly know how they work, or for what reasons they go wrong. It's still early days, and we can't be sure what the maths questions will be, but I know there will be some interesting ones as it develops."

In this context, ML provides many exciting opportunities for applied mathematics in terms of not only offering answers but also at the level of formulating what kind of mathematical question can reasonably be answered when there is no explicit model to analyze.

For ML to be adopted by ASCR, it is necessary that it meets the level of rigor required by ASCR applied mathematics. An analysis technique may be considered *rigorous* if it provides three essential pieces of information:

- Complete description of all steps in the analysis, from the raw input data through the final product, which enable the analysis to be independently reproduced
- Complete list of assumptions, parameters, and algorithmic choices used
- Verification and validation of the techniques and their implementation.

These features are characteristic of good science but are not always easy to achieve. Analysis techniques for ML pose their own special challenges. ML (e.g., SVMs) has well-developed theoretical frameworks [90]. Others, including the currently much-in-use DNNs, still lack a full and thorough framework. Much of the work is recent (e.g., Shaham et al. [374]) along with some decades-old, well-known results, such as those on the "universal approximation properties" of NNs or optimal approximation for smooth functions [301, 375–379]. While the early work does explore basic approximation properties, much of it will need to be revisited in the context of advances in network complexity and training methodologies deployed on currently practiced methods. Paraphrasing Shaham et al. [374], current challenges include specification of the network topology (i.e., depth; layer sizes) given a target function to obtain desired approximation properties, obtaining estimates of the amount of training data needed to generalize to test data with high accuracy, and development of training algorithms with performance guarantees. In addition, during a keynote address at Neural Information Processing Systems (NIPS) 2017 conference, Ali Rahimi highlighted the need for ML experiments and theorems that "chase down the explanation for a strange phenomenon you observed" in order to make the use and application of ML less "alchemy" and more science [380].

Complete and Reproducible Descriptions. As data sets become even larger and more complex, "raw" data are often processed significantly before the "core" analysis starts. Depending on the use case, this can include removing inconvenient outliers (disregarding data sets with less than ideal results), choosing training and testing sets, and any number of other "pre-processing" operations. Usually, these operations are barely discussed if they are mentioned at all; yet they can significantly alter the results. Furthermore, even the core algorithms can contain hidden choices, for example which normalization procedures are used, which clustering algorithms, which neighborhood graphs, all of which may influence the results. A rigorous analysis should provide a complete and reproducible description of all steps.

Complete Listing of Assumptions, Parameters, and Algorithmic Choices. Many algorithms and statistical techniques rely on various assumptions (e.g., data being i.i.d. and noise being normal-distributed). A rigorous approach will not only list these assumptions but demonstrate that they are (expected) to be met by the data.

Verification and Validation. A basic tenet of predictive science is the issue of the confidence that can be attached to predictions. Verification and validation has become a basic requirement for many areas of computation and modeling in science. Several papers [381–383] were prompted by the burgeoning of computational predictive science over the past two decades and the need to maintain high levels of scientific rigor if disasters are to be averted. The classic example is the Sleipner failure, an offshore oil platform [384, 385] in which ill-founded confidence in the predictions coming from a sophisticated finite element simulation code led to the failure and loss of the platform at an estimated cost of $180 million (in 1991). It is essential that ML embraces the lessons learned from this and other disasters about the importance of developing and applying appropriate verification and validation standards that must be met by ML to be accepted as *bona fide* predictive techniques suitable for adoption by ASCR.

Model Reduction and Multifidelity Modeling

The field of *model reduction* encompasses a broad range of methods that seek efficient low-dimensional representations of an underlying high-fidelity model [116]. The majority of model reduction methods have targeted the case where the high-fidelity model is a high-dimensional system of ODEs or a discretization of PDEs. A large class of model reduction methods derive the low-dimensional approximation by projecting the original model onto a low-dimensional subspace (or, more generally, a low-dimensional manifold). The projection framework can be combined with various ways of representing parametric dependence, where for example the parameters may describe material properties, system geometry, system configuration, initial conditions, and boundary conditions. A recent survey paper summarizes state of the art in projection-based model reduction methods and applications [386]. There also is a growing body of literature on data-driven methods that infer the reduced model directly from data (typically simulation data), avoiding the need for an explicit projection step.

Multifidelity methods recognize that the model reduction process introduces approximation errors. For the complex nonlinear multiscale problems of interest to DOE, these approximation errors cannot always be quantified and controlled. Yet, even an approximate reduced model of unknown quality can be of great use if employed within a rigorous multifidelity framework. The overall premise of these multifidelity methods is that approximate models are leveraged for speedup, while the high-fidelity model is kept in the loop to establish accuracy and/or convergence guarantees. A recent survey paper summarizes multifidelity methods for applications in uncertainty propagation, inverse problems, and optimization [194].

Model reduction has clear connections to ML. In fact, many of the methods used to determine the low-dimensional subspace are closely related to ML methods, even if the terminology differs. For example, POD, perhaps the most widely used model reduction method, is closely related to PCA. Moreover, many of the same ideas exist in the model reduction and ML communities, albeit under different names, such as active learning, which is akin to adaptive model reduction methods, and reinforcement learning, which is comparable to goal-oriented model reduction. The difference in fields is perhaps largely one of history and perspective: model reduction methods have grown from the scientific community with a focus on *reducing* high-dimensional models that arise from physics-based modeling; whereas ML has grown from the computer science community with a focus on *creating* low-dimensional models from black box data streams. Yet, in recent years, there has been an increased blending of the two perspectives and recognition of their associated opportunities.

One set of opportunities relates to the growing body of literature on data-driven model reduction methods. Methods such as the Loewner framework [387, 388], vector fitting [389, 390], dynamic mode decomposition [391–393], eigensystem realization algorithm [394], and operator inference [395] all offer ways to learn a reduced model while respecting some of the structure that arises through the underlying system governing equations. In particular, these methods are formulated from a dynamical systems perspective. This is a key element in establishing notions of stability and convergence, which is distinct from the black box perspective of ML. In the past, data-driven model reduction has been limited to linear time invariant systems with more recent contributions introducing nonlinear terms (via quadratic terms) and parametric dependencies. However, a large gap remains between rigorous data-driven model reduction methods and the highly nonlinear, multiscale applications of relevance to DOE. There are clear opportunities for ML methods to advance data-driven model reduction and in turn for the structure-exploiting perspective of data-driven model reduction to advance ML for dynamical systems. In particular, we note that the model reduction perspective is a way to embed physical constraints and move toward so-called *physics-based ML*. For example, these physical constraints could be enforced through the nature of projection or the definition of the low-dimensional manifold itself. In the model reduction community, this is known as *structure-preserving model reduction*. Again, there are promising existing examples (see [396–399] for applications across different fields), but many important open research questions remain.

A second area of opportunity at the intersection of model reduction and ML is through *a posteriori* error analysis. Rigorous error analysis has been a significant focus of the model reduction community in the past decade, particularly for reduction via the reduced basis method [400–403]. There are many opportunities for bringing this kind of thinking to bear on a more general class of approximate models, including those derived using ML methods. Other opportunities lie in the connections between model reduction and other approximation methods [116]. For instance, recent work has established connections between the performance of reduced basis methods and results from approximation theory that employ concepts such as Kolmogorov widths [404], while others have established connections between model reduction and compressive sensing [405].

These references are not intended to be comprehensive; rather, they indicate examples where making connections across fields has led to new insight, theoretical results, and methods. We believe that there are growing opportunities in making deeper connections among model reduction, approximation theory, and ML.

Computational Complexity

Current Status and Recent Advances. There is increasing interest within the large-scale scientific computing community in ML and statistical inference approaches and automated/hybrid learning tools [213, 406–408]. While ML techniques and tools, such as NNs, Bayesian methods, SVMs, hidden Markov models, and Q-Learning, have been employed for some time, their use on large-scale systems is gaining interest more broadly, especially with recent advances in DL [409]. Several emerging usage modes are presented in Section A.2.

Parallelization also has been used to accelerate the training time for various ML techniques. TensorFlow [410], Graphlab [411], and Petuum [412] are some popular techniques that perform distributed ML on big data. The DistBelief software framework [413] developed by Google also models computational parallelism using multithreading and message passing to accelerate the training time in large-scale distributed deep networks. It also supports data parallelism, where a single objective is optimized using multiple replicas of the model. Similarly, Jacobs et al. proposed a framework built around the Caffe DL framework in [414] for parallelizing DNN training that maximizes the amount of data ingested by the training algorithm and have applied it to image classification on HPC machines [415]. Frameworks such as Theano [416], Caffe [414], GPUMLib [417], Dadiannao [418], and Mxnet [419] have been explored to efficiently leverage the computing capability of graphics processing unit (GPU) clusters to reduce the training time of learning models. Recent work presented at SC17 successfully extracted weather patterns in a 15 terabyte (TB) climate data set, achieving 15 petaflop (PF) performance [254].

Another key factor contributing to the computational complexity is hyperparameter search, which can determine the quality of the DL network and is important when applying DL to a new problem. Li et al. [420] proposed a parameter server framework for distributed ML problems and Young et al. [421] presented an approach for searching the possible space of network hyperparameters and construction that can scale to 18,000 nodes. Deep^3 [422] explored the impact of resource constraint and model training time was reduced in such environments by running subroutines to find optimal NN size and performing data pre-processing to reduce data movement cost.

Scientific Challenges and Opportunities. The pervasive use and speed of integrating ML techniques into science workflows has been quite impressive. As a result, it is important to understand and quantify the computational complexity to the ML techniques used, as well as the overall application workflow that includes these ML techniques. This complexity includes several aspects:

- Computational complexity of the ML algorithm/method is defined by its dependence on the number of samples and size/dimension of each sample and its scalability. This dependence on problem size (large data sizes, large numbers of attributes/features, large numbers of coefficients/parameters) is measured in terms of quality (accuracy or confidence) and system resources (compute, storage, network) size.
- System complexity is defined by constraints that impact the design, execution, and scalability of the ML algorithm/method. These constrains include architectural and software solutions that can effectively and scalably support the ML methods and their integration as a part of an application workflow [423], as well as how algorithms should evolve/adapt to address this complexity.

Key aspects impacting the complexity of ML techniques are related to the large data sets involved and the associated training time and scalability. Approaches such as data sampling have been used to create a smaller version of the original data, which is then used to create and train the analytical model. Different sampling approaches [424–426] are proposed to show that the representative subset of the large data set can be used to effectively train the model without losing significant accuracy.

Computational complexity of an ML method involves dependence on the number of samples N and size/dimension of each sample d. Effective sample size n can be defined as the number of training samples required for achieving a desired accuracy with associated confidence bounds for the given model choice (SVM, DNN, etc.). Generally, the computational complexity of learning a chosen model with given accuracy and confidence is a function of d and n. Notably, this is not only about resource sensitivity, but the scalability of ML algorithms with increased problem size, e.g., large data sizes, large numbers of attributes/features, large numbers of coefficients/parameters to be estimated, and large numbers of classes/models, irrespective of available resources. This is a reasonably well-studied problem and there are several performance guarantees for the classification problem [89] based on the accuracy of a feed-forward net that depends on the network's size, amount of training data, and accuracy of the training data under the assumption that future data will be drawn independently from the same probability distribution as the training data. For the generic classification task, there are several results characterizing the behavior of NNs in terms of the Vapnik-Chervonekis (VC)-dimension of the architecture [427–430]. Specifically, these results provide asymptotic bounds on the VC-dimension based on the number of weights, number of inputs, type of network, and activation functions. However, as the VC-dimension only captures performance on an optimal input arrangement, these results leave open the question of how NNs perform on more realistically structured inputs. In particular, the typical performance of an NN classifier when trained on limited correlated data remains an open question.

Summarizing, we must understand the scaling of computational ML methods with increased problem size:

- Data size and number of data samples
- Number of attributes, features, and dimensionality of the input space
- Model complexity, such as the size of the NN, number of nodes, edges, and layers
- Number of classes.

Dimensionality must be defined to include:

- Intrinsic dimensionality

- Extrinsic/embedding dimension.

Techniques may be scalable in one but not the other or have vastly different limits. Key questions to analyze and establish good modeling and theory will be:

- At what point with respect to data or compute resources does the method become useful (resource dependence)?
- What are examples of "anytime" methods for ML?
- What is the relationship between resources (data, compute, time) available and the "quality" of the model?

Such work will initiate with simple analysis, as in [89], but must reflect complexity of the triplet (data, algorithm, platform). Another growing area is the establishment of theory that deals with streaming data, e.g., sufficiency proofs for data needed to establish specific hypotheses.

Extreme Scalability of Machine Learning Techniques. While there is growing adoption and a number of ongoing research efforts focused on using ML techniques on cloud and distributed systems, mapping these techniques to extreme-scale HPC machines is not as widely explored and presents new challenges. While sampling techniques in HPC systems may reduce the large data sets to achieve acceptable solutions, there will be reduction in the data quality and some of the hidden details could be overlooked. Hence, solutions that can reduce the data size without compromising the accuracy of the trained model will be required. Although distributed ML frameworks aim to parallelize the data movement among various nodes to minimize the overall data transfer time, they still suffer from large amounts of data movement. To amortize this cost, exploration of *in situ* ML algorithms is essential. In addition to this exploration, data staging techniques (e.g., enabled by DataSpaces [431]) may alleviate the costs of large-scale data distribution by training the learning models using in-transit workflows, but the impact of data staging on the execution time of ML algorithms has not been well studied. Effectively using heterogeneous resources in HPC systems for ML is another research challenge still being investigated. Thus, designing new HPC architectures for current ML techniques and innovative algorithms to fully utilize the HPC capability is an important future research direction.

Integrating Machine Learning within *in situ* Application Workflows. The growing data sizes and complexity associated with scientific applications, increasing performance gap between compute and I/O capabilities, and significant data management challenges and costs at extreme scales have led to significant interest in and support for *in situ* and in-transit data processing [432]. As ML techniques become an important part of scientific application workflows, scalable *in situ* and in-transit formulations and implementations of these techniques are also increasingly important. Such implementations present significant challenges at all levels, from algorithmic formulations and programming abstractions suitable to online and *in situ* and in-transit execution to runtime services to manage the control and data flow of the overall workflow. For example, integrating current ML libraries, such as Theano [416] and Caffe [414], requires significant code changes to scientific workflows and analysis applications. Enabling various ML algorithms inside data staging servers can decouple the scientific workflows from feature extraction, which can be as easy as calling an application programming interface with a target feature of interest as an input.

Optimization

The fields of mathematical optimization and ML have deep connections that span both modeling and algorithms. These connections are quickly evolving with advances in each area, leading to new problem classes, algorithms, and theoretical insights in the other field [8, 433–436].

In recent years, the advances made in optimization analysis have significantly improved theoretical insight for ML. The training process in supervised learning is almost exclusively posed as an optimization problem with an objective function defined by a loss function and training data. Depending on the form of the loss function and parameters/weights being optimized, a number of classes arise, including convex, non-convex, and non-smooth optimization problems. Selection of the loss function form is an active research area with implications for solvability, generalizability, and application in distributed, online, and other computational contexts. Training objectives also can include a regularization term with similar implications and may result in strongly convex training formulations for which theoretical analysis, such as optimality and rates of convergence, has been established for many classes of optimization algorithms [437–440]. Many optimization algorithms relevant to ML exploit explicit problem structure, such as partial separability of an objective [439] or linear algebraic properties [441].

Optimization also plays a role in the form that an ML model takes. For example, a model can depend linearly or nonlinearly on its training parameters. Similarly, computing training objective derivatives with respect to these parameters in an automatic and efficient manner is a task that frequently arises in optimization. Back-propagation-based approaches used for NNs are effectively a special case of the reverse mode of algorithmic differentiation [268, 442], for which rigorous results have been established regarding computational and memory operation complexity. Furthermore, tuning hyperparameters [200, 443], such as network configuration and learning rate, is vital to achieving success for many learning tasks. Hyperparameter tuning itself can be viewed as a bilevel optimization problem [444] with the lower level defined by the traditional model training problem. Other training strategies also have direct analogues in optimization, such as the equivalence of boosting and convex optimization formulations [445].

The performance of an optimization algorithm used in training commonly is a primary design consideration in ML. A majority of the ML frameworks deployed today use some variant of the stochastic gradient descent method for this purpose [169]. Many of these algorithmic variants have been rigorously analyzed (refer to [170–172]) and differ based on their parallelism and access to training data, including cases where very few passes through the data are allowed or where the data are distributed in such a way that synchronous access is infeasible [173]. Significant research has been performed by the ML and optimization communities, leading to improved mini-/multi-batch [174–176] and asynchronous [177, 178] algorithms. At the same time, accelerated and momentum-based techniques have been studied to reduce the number of iterations required by an optimization algorithm [179, 180]. Advances in non-convex optimization methods also have benefited ML in terms of the ability to incorporate loss functions and training objectives with favorable learning and regression properties [181–183]. Examples include methods for variance reduction [184, 185] and globalization techniques, such as trust-region methods [186, 187]. Depending on problem size and solution requirements, first-order [188, 189], second-order [190, 191], and secant [192] methods have been successfully employed in ML.

As learning models (and their training) become more diverse and complex, algorithms have been adopted from diverse areas, such as convex optimization [160], non-smooth optimization [161], robust optimization [162], semi-definite programming [163, 164], stochastic optimization [165], derivative-free optimization [166, 167], and global optimization [168].

Advances from the ML community have significantly contributed to and shifted mathematical optimization developments. Optimization methods also can be a consumer of ML models. Surrogate-based and multifidelity optimization algorithms can employ ML-based models, especially where they can be updated online. Canonical examples include Bayesian optimization [446], where for example a GP-based model (refer to [447]) is dynamically refined in the course of the optimization. The incorporation of constraints in such learning is an active area where optimization ideas can apply [448–450]. Such approaches generally occur in active learning [146, 451], where optimization and statistical approaches, such as sequential design of experiments, provide natural formalisms.

Present-day advances in the space of optimization for DL have focused on using *learned* optimizers. As posited by work done in [452], learning specialized update rules for specific problems may be the only way to achieve improved performance. Recent work has closely examined reducing computational and memory overhead constraints for these learned optimization strategies [453]. Several specific techniques have been developed for the few-shot regime, where networks are trained to learn from small amounts of data [280, 454–456]. Similar learned optimizers have been applied successfully in the hyperparameter optimization space, showing promising results against engineered Bayesian approaches [457].

Statistics and Uncertainty Quantification

The central role of data in ML is conducive to an associated fundamental role for statistics and UQ methods for a diversity of ML problems. Typically, data are noisy and uncertain as well as incomplete, sparse, and only partly informative. Similarly, models are generally uncertain in terms of their form and internal structure and associated parameters. This preponderance of noise and uncertainty is a strong motivation for using statistics and UQ in data analysis, modeling, and ML.

High dimensionality is ubiquitous in both forward propagation of uncertainty and statistical inference. An important feature of smooth observables in physical systems, namely their low intrinsic dimension, is key to the various avenues of dealing with this challenge. Similarly, data are often informative on only a subset of parameters of interest. Hence, the ongoing development of methods for identifying important subsets of model inputs in forward UQ; discovering low-dimensional, low-rank sparse functional representations; and adaptive basis and active subspace methods remains important. The demonstrated utility [458] of compressive sensing methods [459] and various methods for ℓ_1-norm constrained regression for identifying sparse constructions are particularly noteworthy. There has been significant development of methods for distinguishing the data/likelihood-informed subspaces in Bayesian inference, as well as the development of Markov chain Monte Carlo (MCMC) and other methods for robust inference in high dimensions. Model complexity and high computational cost are also major challenges that have seen significant development over recent years. Multilevel and multifidelity methods have been developed to afford optimal use of computational resources in forward UQ, surrogate construction, and optimization. Adaptive local surrogates have been employed within trust regions to provide optimal utilization of computational resources. Similarly, model error methods have evolved to deal with the challenges associated with complex physical systems and inherent structural errors in available models. Further, software with multilevel parallelism has been employed for effective management of parallel computations of many concurrent instances of large-scale computational models. Optimization methods also have evolved to address uncertainty and noise in model computations. Regression is employed with effective model surrogates to deal with noisy model evaluations. Uncertainties in model inputs and operating conditions are incorporated in optimization QoIs and various risk measures are employed to define utility functions of interest.

A range of open-source software products have been developed, benefiting from ASCR funding, to manage a range of UQ and statistical problems. These have evolved with Scientific Discovery through Advanced Computing (SciDAC) funding in directions targeting DOE SC applications.

Broadly speaking, there are many fruitful avenues for interaction between statistics, UQ, and ML. The developments (as discussed) in UQ/statistics methods and software provide significant capabilities for handling high-dimensional, computationally complex models with efficient use of computational resources. In addition, there is a long history of developments in applications of statistics and probabilistic modeling in ML. For example, the incorporation of probabilistic modeling in ML [99] targets feasible estimation of uncertainty in ML predictions. There also are potential advantages for regression and ML with non-parametric models, such as GPs, versus predetermined NN structures in DL [103]. Often, statistical methods are useful for dealing with the problem of overfitting with DL and CNNs given small amounts of data [98]. For example, effective cross-validation methods [460] have impact in ML. Similarly, Bayesian integration is useful for general high-dimensional function applications [100] with relevance in ML. Bayesian modeling also is finding utility in Bayesian generative adversarial networks [101, 102]. Notably, statistics and UQ methods can add significant robustness and realism to ML fitting/regression methods. In many places where multiple minima exist and the solution space is large, a high degree of ill-conditioning is present given the many, nearly equivalent feasible solutions. Reformulating the problem as a statistical inverse problem can add significant conditioning, changing the question from determining the best solution to finding the set of solutions with significant probability. This can provide a solution with quantified uncertainty estimates in its parameters/weights and structure. Furthermore, it is clear that ASCR-funded developments in computational methods and HPC platforms generally are of significant utility for scalable ML in a large-scale computational setting for big data [216].

Notably, there are many potential benefits to ASCR areas of interest from engaging ML in computational science. ML methods are especially useful for discovering correlations in high-dimensional data. Thus, they can find utility in analyzing computational results. ML methods also are useful for constructing NN surrogate surfaces for complex forward models [25, 26]. ML can additionally benefit learning/discovering low-dimensional manifolds underlying high-dimensional data [28]. These can be employed for understanding the dynamical structure behind the data. They also can help define effective distance measures between data sets, thereby providing paths toward effective likelihoods for complex model calibration and parameter estimation from observational data. ML methods, specifically clustering/classification, have been effective for representing high-dimensional functions exhibiting qualitatively different behavior/functional forms in distinct regions of phase/configuration space [27]. Stochastic gradient descent methods have been used successfully for exploring very high-dimensional spaces and discovering optimal solutions (in the ML literature) for training NN models. These experiences will find good use in numerous areas impacting science and applied math. ML methods have been used in analyzing turbulent flow computations [29–36], subsurface flow modeling [37], solid mechanics modeling of diverse materials [38–46], geophysics [47–50], and combustion modeling [30].

A.3 Workshop Agenda

Tuesday (January 30, 2018)

08:00 – 09:00	Breakfast & Registration
09:00 – 09:15	Welcome Steven Lee (Acting Research Division Director, ASCR)
09:15 – 09:30	Welcome from Co-Chairs Nathan Baker (PNNL) & Mark Ainsworth (Brown)
09:30 – 10:25	Scientific Machine Learning: ASCR Facilities Perspective Venkat Vishwanath (ALCF) Mariam Kiran (ESnet) Junqi Yin (OLCF) Prabhat (NERSC)
10:25 – 10:45	Break
10:45 – 11:25	Three Principles of Data Science: Predictability, Stability, and Computability Bin Yu (University of California (UC) Berkeley)
11:25 – 12:15	Scientific Machine Learning across Federal Agencies Elizabeth Hsu (National Institutes of Health (NIH)) Edward McLarney (National Aeronautics and Space Administration (NASA) Langley Research Center) Chris Boehnen (Intelligence Advanced Research Project Activity (IARPA)) Doug Kothe (ORNL, ECP)
12:15 – 13:30	Working lunch Summary of pre-workshop report & themes Reminder of breakout session goals
13:30 – 14:10	Physics, Structure, and Uncertainty: Probabilistic Learning for Risk Mitigation Roger Ghanem (University of Southern California)
14:10 – 16:10	Parallel breakout sessions & flash talks
16:10 – 16:25	Coffee break
16:25 – 18:10	Parallel breakout sessions
18:10 – 19:10	Dinner (on your own)
19:10 – 21:00	Parallel breakouts: preparation for preliminary reports

Wednesday (January 31, 2018)

08:00 – 09:00	Breakfast
09:00 – 09:40	Machine Learning in the Wild Jacob Shapiro (Princeton)
09:40 – 10:30	Preliminary reports & discussion
10:30 – 11:00	Break
11:00 – 12:00	Preliminary reports & discussion
12:00 – 13:30	Working lunch
13:30 – 14:10	Challenge and Scope for Empirical Modeling Ronald Coifman (Yale)

14:10 – 16:10	Parallel breakout sessions
16:10 – 16:25	Coffee break
16:25 – 18:10	Parallel breakout sessions
18:10 – 19:10	Dinner (on Your own)
19:10 – 21:00	Parallel breakouts: preparation for final reports

Thursday (February 1, 2018)

08:00 – 09:00	Breakfast
09:00 – 10:40	Final reports & discussion
10:40 – 10:55	Break
10:55 – 12:25	Final reports & discussion
12:25 – 12:50	Priority research directions – summary
12:50 – 13:00	Closing remarks Steven Lee
13:00	ADJOURN
13:00 – 14:00	Working lunch Chairs, breakout leads, & designated writers only
14:00 – 17:00	Writing session Chairs, breakout leads, & designated writers only

Breakout sessions

1. Numerical analysis for ML

2. ML, multifidelity, & reduced-order models

3. ML, optimization, & complexity

4. Probabilistic ML

5. ML interpretability

A.4 Workshop Participants

Organizers

Chair	Nathan Baker	Pacific Northwest National Laboratory
DOE ASCR Applied Mathematics lead	Steven Lee	Advanced Scientific Computing Research
Organizing committee and breakout leads	Frank Alexander	Brookhaven National Laboratory
	Timo Bremer	Lawrence Livermore National Laboratory
	Aric Hagberg	Los Alamos National Laboratory
	Yannis Kevrekidis	Johns Hopkins University
	Habib Najm	Sandia National Laboratories
	Manish Parashar	Rutgers University
	Abani Patra	State University of New York Buffalo
	James Sethian	University of California Berkeley
	Stefan Wild	Argonne National Laboratory
	Karen Willcox	University of Texas at Austin
Other contributors	Courtney Corley	Pacific Northwest National Laboratory
	Nathan Hilliard	Pacific Northwest National Laboratory
	Jeff Hittinger	Lawrence Livermore National Laboratory
	Ana Kupresanin	Lawrence Livermore National Laboratory
	Cosmin Petra	Lawrence Livermore National Laboratory
	Prabhat	Lawrence Berkeley National Laboratory
	Laura Pullum	Oak Ridge National Laboratory
	Laura Swiler	Sandia National Laboratories
	Chris Symons	Oak Ridge National Laboratory
	Xun Huan	Sandia National Laboratories
	Alex Tartakovsky	Pacific Northwest National Laboratory
	Stephen Young	Pacific Northwest National Laboratory
Workshop support	Jody Crisp	Oak Ridge Associated Universities
	Bruce Warford	Oak Ridge Associated Universities
	Julie Webber	Oak Ridge Associated Universities

Participants

The following is a list of registered participants in the "Scientific Machine Learning" workshop held January 30 to February 1.

- Mark Ainsworth – Brown University
- Rick Archibald – ORNL
- Paul Atzberger – UC Santa Barbara
- Nathan Baker – PNNL
- Prasanna Balaprakash – Argonne National Laboratory (ANL)
- Andrew Barker – LLNL
- Paul Bauman – State University of New York (SUNY) Buffalo
- Russell Bent – Los Alamos National Laboratory (LANL)
- Julie Bessac – ANL
- Laura Biven – United States (US) DOE
- Christopher Boehnen – IARPA
- Phil Bording – Alabama A&M University

- Peer-Timo Bremer – LLNL
- Robert Bridges – ORNL
- David Brown – LBNL
- Aydin Buluc – LBNL
- Richard Carlson – US DOE
- Lali Chatterjee – US DOE
- Barry Chen – LLNL
- Michael Chertkov – LANL
- Eric Chisolm – US DOE
- Leland Cogliani – Lewis Burke Associates
- Ronald Coifman – Yale University
- Bill Conley – Office of the Secretary of Defense
- Claire Cramer – US DOE
- Silvia Crivelli – LBNL
- Bert Debusschere – Sandia National Laboratories (SNL)
- Lori Diachin – LLNL
- David Etim – National Nuclear Security Administration
- Nicola Ferrier – ANL
- Fernanda Foertter – ORNL
- Cristina Garcia Cardona – LANL
- Roger Ghanem – University of Southern California
- Pieter Ghysels – LBNL
- Alex Gorodetsky – University of Michigan
- Elizabeth D Gregory – NASA Langley/-Computational AeroSciences & High Performance Computing Incubator
- Aric Hagberg – LANL
- Mahantesh Halappanavar – PNNL
- Charlotte Haley – ANL
- Alan Heirich – Stanford University & SLAC National Accelerator Laboratory (SLAC) National Accelerator Center
- Jan Hesthaven – Ecole Polytechnique Federale de Lausanne (EPFL)
- Hassan Hijazi – LANL
- Jeffrey Hittinger – LLNL
- Harsha Honnappa – Purdue University
- Paul Hovland – ANL
- Elizabeth Hsu – NCI
- Zhenyu Huang – PNNL
- Eliu Huerta – University of Illinois Urbana-Champaign (UIUC) & National Center for Supercomputing Applications
- Ilse Ipsen – North Carolina State University
- Elise Jennings – ANL
- Ben Kallen – Society for Industrial and Applied Mathematics (SIAM)
- Yannis Kevrekidis, Johns Hopkins University
- Kibaek Kim – ANL
- Mariam Kiran – ESnet & LBNL
- Kerstin Kleese van Dam – Brookhaven National Laboratory (BNL)
- Matthew Knepley – SUNY Buffalo
- Tamara Kolda – SNL
- Doug Kothe – ORNL
- Ana Kupresanin – LLNL
- Thorsten Kurth – LBNL
- Steven Lee – US DOE
- Eliane Lessner – US DOE

- Sven Leyffer – ANL
- Sherry Li – LLNL
- Faming Liang – Purdue University
- Youzuo Lin – LANL
- Dalton Lunga – ORNL
- Barney Maccabe – ORNL
- Ramana Madupu, US DOE
- Joseph Manzano – PNNL
- Sonia McCarthy – US DOE
- Ryan McClarren – University of Notre Dame
- Edward McLarney – NASA Langley Research Center
- Dmitriy Morozov – LBNL
- Habib Najm – SNL
- Thomas Ndousse–Fetter – US DOE
- Esmond Ng – LBNL
- Lucy Nowell – US DOE
- Manish Parashar – Rutgers University
- Vivak Patel – University of Chicago
- Abani Patra – SUNY Buffalo
- Benjamin Peherstorfer – University of Wisconsin–Madison
- Talita Perciano – LBNL
- Paris Perdikaris – University of Pennsylvania
- Cosmin Petra – LLNL
- Caleb Phillips – National Renewable Energy Laboratory (NREL)
- Ali Pinar – SNL
- Eric Pouyoul – ESnet
- Mr Prabhat – LBNL
- Laura Pullum – ORNL
- Arvind Ramanathan – ORNL
- Nageswara Rao – ORNL
- Daniel Ratner – SLAC
- Jaideep Ray – SNL
- Matthew Reynolds – NREL
- Baruch Schieber – IBM AI Research
- Michelle Schwalbe – National Academy of Sciences Board on Mathematical Sciences and Analytics
- Thiago Serra – Carnegie Mellon University
- James Sethian – UC Berkeley
- Phiala Shanahan – William & Mary – Thomas Jefferson National Accelerator Facility
- Jacob Shapiro – Princeton University
- Devanand Shenoy – US DOE
- Ted Slater – Cray Inc.
- Jim Stewart – SNL
- Panos Stinis – PNNL
- Sreenivas Sukumar – Cray Inc.
- Ceren Susut – US DOE
- Pieter Swart – LANL
- Laura Swiler – SNL
- Christopher Symons – ORNL
- Alexandre Tartakovsky – PNNL
- Valerie Taylor – ANL
- Jayaraman Thiagarajan – LLNL
- Christian Tjandraatmadja – Carnegie Mellon University
- Madeleine Udell – Cornell University

- Venkatram Vishwanath – ANL
- Bruce Warford – Oak Ridge Associated Universities – Oak Ridge Institute for Science and Education
- Julie Webber – Oak Ridge Associated Universities – Oak Ridge Institute for Science and Education
- Clayton Webster – ORNL
- Stefan Wild – ANL
- Karen Willcox – Massachusetts Institute of Technology (MIT)
- Brendt Wohlberg – LANL
- David Womble – ORNL
- Jinchao Xu – Pennsylvania State University
- Chao Yang – LBNL
- Ulrike Yang – LLNL
- Xiu Yang – PNNL
- Qian Yang – Stanford University
- Junqi Yin – ORNL
- Shinjae Yoo, BNL
- Bin Yu – UC Berkeley
- Zhizhen (Jane) Zhao – UIUC

A.5 Workshop Presentations

Plenary Presentation Abstracts

The following is a list of plenary talks at the "Scientific Machine Learning" workshop held January 30 to February 1.

Three Principles of Data Science: Predictability, Stability, and Computability. Bin Yu, University of California Berkeley

In this talk, I'd like to discuss the intertwining importance and connections of three principles of data science in the title. They will be demonstrated in the context of two collaborative projects in neuroscience and genomics, respectively. The first project in neuroscience uses transfer learning to integrate fitted CNNs on ImageNet with regression methods to provide predictive and stable characterizations of neurons from the challenging primary visual cortex V4. The second project proposes iterative random forests (iRFs) as a stabilized random forest (RF) to seek predictable and interpretable high-order interactions between biomolecules.

Physics, Structure, and Uncertainty: Probabilistic Learning for Risk Mitigation. Roger Ghanem, University of Southern California

Increasingly, critical decisions are demanded for situations where likelihoods are not sufficiently constrained by models. This could be caused by the lack of suitable mathematical models or the inability to compute the behavior of these models, or observe the associated physical phenomena, under a sufficient number of operating conditions. In many of these situations, the criticality of the decisions is manifested by the need to make inferences on high consequence events, which are typically rare. The setting is thus one of characterizing extreme events when useful models are lacking, computational models are expensive, or empirical evidence is sparse. We have found adaptation and learning to provide transformative capabilities in all of these settings. A key observation is that models and parameters are typically associated with comprehensive constraints that impose conservation laws over space and time, whose solution yields spatio-temporal fields, and that require comprehensive calibration with exhaustive data. Decisions typically depend on QoIs that are agnostic to this complexity and that are constructed through an aggregation process over space, time, or behaviors. A regularization is thus imposed by allowing the QoIs to drive the complexity of the problem. But then one has to learn the QoIs. This talk will describe recent procedures for probabilistic learning of QoIs using both orthogonal projections and manifold learning. Both approaches are applied to science and engineering problems where models are either too expensive to compute or too inconclusive to provide acceptable interpolation to data. In both cases, probabilistic inferences are possible as required by risk assessment and probabilistic-based design.

Machine Learning in the Wild. Jacob Shapiro, Princeton University

Tremendous advances in machine learning combined with increasingly pervasive sensing are enabling new modes of inference and new kinds of measurement across public policy and the social sciences. Opportunities for learning about the world and doing good in it extend from measuring poverty, to predictive policing, to healthcare cost management, and beyond. But these advances come with risks. Beyond the well-documented human costs of overfitting and algorithmic discrimination, the core functionality that made traditional statistics such a powerful tool for hypothesis driven research—transparent (albeit debatable) standards for distinguishing signal from noise, a rich toolkit for assessing the validity of causal inference, and the ability to conduct power calculations when planning data collection—does not yet exist for most machine learning approaches. This talk will cover exciting new applications, discuss pitfalls of using machine learning for policy, and lay out what practical users outside of engineering and the natural sciences need from the machine learning community.

Challenge and Scope for Empirical Modeling. Ronald Coifman, Yale University

We will describe examples demonstrating that true physical or biological models can be automatically learned from raw data. The methods are totally data and modality agnostic. We see a challenging paradigm shift emerging, due to various geometry and code, learning, and deep and recurrent neural nets. These are but the tip of the iceberg, and suffer from a variety of precision and uncertainty limitations. We will address some basic mathematical issues, related to stochasticity, nonlinearity and of course dimensionality and scalability.

Panel Presentation Titles

The following is a list of panel presentations at the "Scientific Machine Learning" workshop held January 30 to February 1.

Scientific Machine Learning: ASCR Facilities Perspective

- Venkat Vishwanath – ALCF
- Mariam Kiran – ESnet
- Junqi Yin – OLCF
- Prabhat – NERSC

Scientific Machine Learning across Federal Agencies

- Elizabeth Hsu – NIH
- Edward McLarney – NASA Langley Research Center
- Chris Boehnen – IARPA
- Doug Kothe – ECP

Flash Talk Titles

The following is a list of flash talks presented at the "Scientific Machine Learning" workshop held January 30 to February 1.

- *Importance of the Mathematical Foundations of Machine Learning Methods for Scientific and Engineering Applications*. Paul Atzberger, UC Santa Barbara.
- *Machine Learning for Novel Algorithm Discovery*. Andrew Barker, LLNL.
- *Derivative-Free Mixed-Integer Optimization for Automated Predictive Modeling using Machine Learning*. Prasanna Balaprakash, ANL.
- *Bayesian Machine Learning for High Velocity Streaming Data*. Bruce Bugbee, NREL.

- *Communication-Avoiding Computational Primitives in Machine Learning for Science*. Aydin Buluc, LBNL.
- *Mathematics for Physics-Informed Machine Learning*. Michael Chertkov, LANL.
- *Enabling Scalable Deep Learning in Complex Structural Domains via Feature-Preserving Dimensionality Reduction*. Silvia Crivelli, LBNL.
- *Predictive Fidelity of Machine Learning Methods Applied to Scientific Simulations*. Bert Debusschere, SNL.
- *Linear Complexity Hierarchical Matrix Algebra for Kernel-Based Machine Learning*. Pieter Ghysels, LBNL.
- *Low-Rank Continuous Tensor Decompositions for Machine Learning*. Alex Gorodetsky, University of Michigan.
- *A Graph Theoretical Approach to Deep Learning: Design, Optimization and Generalization of Deep Networks*. Mahantesh Halappanavar, PNNL.
- *Adaptive Nonparametric Spectra for Kernel Learning*. Charlotte Haley, ANL.
- *Inference and Optimal Estimation for Machine Learning*. Harsha Honnappa, Purdue University.
- *Direct Model Inversion, Bifurcations of Local Minima*. Alan Heirich, SLAC.
- *Enhancing Reduced Order Models through Machine Learning*. Jan Hesthaven, EPFL.
- *Accelerating Families of Simulations through Adaptive Learning*. Jeff Hittinger, LLNL.
- *Applicability of Scientific Machine Learning*. Henry Huang, PNNL.
- *Real-Time Regression Analysis with Deep Convolutional Neural Networks*. Eliu Huerte, UIUC.
- *Numerical Integrity of Scientific Machine Learning*. Ilse Ipsen, North Carolina State University.
- *Complexity Measures for Identifying Discrete Structures in Data*. Kibaek Kim, ANL.
- *The Kolmogorov Superposition Theorem for Machine Learning*. Matthew Knepley, SUNY Buffalo.
- *Understanding Randomized Algorithms in Machine Learning*. Tammy Kolda, SNL.
- *Interpretable Machine Learning for Validation and Uncertainty Quantification of Complex Models*. Ana Kupresanin, LLNL.
- *Optimization Models and Paradigms for Machine Learning*. Sven Leyffer, ANL.
- *Learning Algorithms for Hyperparameter Optimization*. Sherry Li, LBNL.
- *Hybrid Physics-Guided Data-Driven Inverse Analysis*. Youzou Lin, LANL.
- *Generalizing Semi-Supervised and Unsupervised Learning for Domain Adaption with Very Large Scientific Data*. Dalton Lunga, ORNL.
- *Quantifiable and Efficient Data Curation for Deep Learning*. Joseph Manzano, PNNL.

- *Machine Learning as a Closure Tool*. Ryan McClarren, ORNL.
- *Descriptive Precision in Scientific Machine Learning*. Dmitriy Morozov, LBNL.
- *Generalizable Scientific Machine Learning*. Vivak Patel, University of Chicago.
- *Model Reduction Meets Machine Learning Inferring Physically Consistent Models from Data*. Benjamin Peherstorfer, University of Wisconsin.
- *Mathematics in Pattern Recognition for Scientific Investigations*. Talita Perciano, LBNL.
- *Learning from Small Data*. Paris Perdikaris, University of Pennsylvania.
- *Unsupervised Learning Through Randomized Algorithms for High-Volume, High-Velocity Data (ULTRA-HV)*. Ali Pinar, SNL.
- *Mathematically Rigorous Verification and Validation of Scientific Machine Learning*. Laura L. Pullum, ORNL.
- *Finite-Sample Generalization Theory for Machine Learning Practice for Science*. Nageswara Rao, ORNL.
- *Bayesian Modeling of Neural Net Emulators*. Jaideep Ray, SNL.
- *Multi-fidelity Methods for Training Neural Networks*. Matthew Reynolds, NREL.
- *Symbolic Regression via Optimization*. Baruch Schieber, IBM.
- *How Could Polyhedral Theory Harness Deep Learning?*. Thiago Serra, Carnegie Mellon University.
- *Machine Learning the Structure of Matter: Systems with Symmetries*. Phiala Shanahan, William and Mary College.
- *Position Paper for DOE ASCR Workshop on Scientific Machine Learning: Materials Models*. Laura Swiler, SNL.
- *Merging HPC and ML for Data-driven Scientific Computing*. Alex Tartakovsky, PNNL.
- *Machine Learning for Accelerated Optimization Algorithms*. Nathan Urban, LANL.
- *A Mathematical Foundation Necessary for Advancing Machine Learning*. Clayton Webster, ORNL.
- *Dictionary Learning Methods for Scientific Data Analysis*. Brendt Wohlberg, LANL.
- *Convolutional Neural Networks and Multigrid Methods*. Jinchao Xu, Pennsylvania State University.
- *Machine Learning via Low-rank Tensor Networks*. Chao Yang, LBNL.
- *Scalable Multilevel Algorithms and Deep Learning*. Ulrike Yang, LLNL.
- *Addressing Unevenly Distributed Datasets and High Accuracy Requirements in Scientific Machine Learning through Improved Validation Methods and Representation Learning*. Qian Yang, Stanford University.

- *Integrating Machine Learning and Physical Modelling for Scientific Discovery*. Xiu Yang, PNNL.
- *Uncertainty Quantification in Statistical Machine Learning*. Zhizhen Zhao, UIUC.

Appendix B

List of Abbreviations

3D	three-dimensional
A3C	asynchronous advantage actor critic
AI	artificial intelligence
AIC	Akaike information criterion
ALCF	Argonne Leadership Computing Facility
ALE	arbitrary Lagrangian-Eulerian
ANL	Argonne National Laboratory
ARMA	autoregressive moving average
ASCR	Advanced Scientific Computing Research
BIC	Bayes information criterion
BNL	Brookhaven National Laboratory
BNN	Bayesian neural network
BRN	basic research needs
CANDLE	cancer distributed learning environment
CNN	convolutional neural network
DARPA	Defense Advanced Research Projects Agency
DL	deep learning
DNN	deep neural network
DNS	direct numerical simulation
DOE	Department of Energy
DQN	deep Q network
ECP	Exascale Computing Project
EPFL	Ecole Polytechnique Federale de Lausanne
ESnet	Energy Sciences Network
FOA	Funding Opportunity Announcement
GANs	generative adversarial networks
GP	Gaussian process
GPU	graphics processing unit
HPC	high-performance computing
i.i.d.	independent and identically distributed (random variables)

IARPA	Intelligence Advanced Research Project Activity
I/O	input/output
iRF	iterative random forest
LANL	Los Alamos National Laboratory
LBNL	Lawrence Berkeley National Laboratory
LES	large-eddy simulation
LLE	locally linear embedding
LLNL	Lawrence Livermore National Laboratory
MACORD	machine-learning-based silent data corruption detection
MCMC	Markov chain Monte Carlo
MDL	minimum description length
MDP	Markov decision process
MIT	Massachusetts Institute of Technology
ML	machine learning
NARMA	nonlinear autoregressive moving average
NASA	National Aeronautics and Space Administration
NERSC	National Energy Research Scientific Computing Center
NIH	National Institutes of Health
NCI	National Cancer Institute
NIF	National Ignition Facility
NIPS	Neural Information Processing Systems
NLP	natural language processing
NN	neural network
NREL	National Renewable Energy Laboratory
ODE	ordinary differential equation
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
PAC	probably approximately correct
PC	polynomial chaos
PCA	principal component analysis
PDE	partial differential equation
PDF	probability distribution function
PF	petaflop
PML	probabilistic machine learning
PNNL	Pacific Northwest National Laboratory
POD	proper orthogonal decomposition
PRD	Priority Research Direction
QoI	quantity of interest
RANS	Reynolds-averaged Navier Stokes
RF	random forest
RNN	recurrent neural network
SC	Office of Science
SciML	scientific machine learning
SIAM	Society for Industrial and Applied Mathematics
SLAC	SLAC National Accelerator Laboratory
SNE	stochastic neighbor embedding
SNL	Sandia National Laboratories
SUNY	State University of New York
SVD	singular value decomposition
SVM	support vector machine
TB	terabyte

TBNN	tensor basis neural network
TRPO	trust region policy optimization
TSVM	transductive support vector machine
UC	University of California
UIUC	University of Illinois Urbana-Champaign
US	United States
UQ	uncertainty quantification
VC	Vapnik-Chervonekis
VI	variational inference
XAI	eXplainable Artificial Intelligence

Bibliography

[1] Executive Office of the President, National Science and Technology Council, Committee on Technology. *Preparing for the Future of Artificial Intelligence*. https://go.usa.gov/xE9ex. Oct. 2016.

[2] J. New. *Why the United States Needs a National Artificial Intelligence Strategy and What It Should Look Like*. http://www2.datainnovation.org/2018-national-ai-strategy.pdf. Dec. 2018.

[3] National Science and Technology Council, Networking and Information Technology Research and Development Subcommittee. *National Artificial Intelligence Research and Development Strategic Plan*. https://go.usa.gov/xE9eZ. Oct. 2016.

[4] Department of Energy. *Quadriennial Technology Review. An Assessment of Energy Technologies and Research Opportunities*. Sept. 2015.

[5] Department of Energy. *Quadriennial Energy Review. Transforming the Nation's Electricity System: The Second Installment of the QER*. Jan. 2017.

[6] A. L. Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal* 7 (1959), pp. 211–229.

[7] The Royal Society. *Machine learning: the power and promise of computers that learn by example*. Tech. rep. DES4702. Apr. 2017.

[8] M. I. Jordan and T. M. Mitchell. "Machine learning: trends, perspectives, and prospects". In: *Science* 349.6245 (July 2015), pp. 255–260.

[9] *U.S. Department of Energy Strategic Plan 2014–2018*. 2014.

[10] J. Czyzyk, T. Wisniewski, and S. Wright. "Optimization Case Studies in the NEOS Guide". In: *SIAM Review* 41.1 (1999), pp. 148–163.

[11] *NEOS Guide: Case studies in optimization*. https://neos-guide.org/Case-Studies. 2017.

[12] B. Narayanan, A. Kinaci, F. G. Sen, M. J. Davis, S. K. Gray, M. K. Y. Chan, and S. K. R. S. Sankaranarayanan. "Describing the Diverse Geometries of Gold from Nanoclusters to Bulk—A First-Principles-Based Hybrid Bond-Order Potential". In: *The Journal of Physical Chemistry C* 120.25 (June 2016), pp. 13787–13800.

[13] F. G. Sen, A. Kinaci, B. Narayanan, S. K. Gray, M. J. Davis, S. K. R. S. Sankaranarayanan, and M. K. Y. Chan. “Towards accurate prediction of catalytic activity in IrO_2 nanoclusters via first principles-based variable charge force field”. In: *Journal of Materials Chemistry A* 3.37 (2015), pp. 18970–18982.

[14] K. Sasikumar, B. Narayanan, M. Cherukara, A. Kinaci, F. G. Sen, S. K. Gray, M. K. Y. Chan, and S. K. R. S. Sankaranarayanan. “Evolutionary Optimization of a Charge Transfer Ionic Potential Model for Ta/Ta-Oxide Heterointerfaces”. In: *Chemistry of Materials* 29.8 (Apr. 2017), pp. 3603–3614.

[15] B. Narayanan, K. Sasikumar, Z.-G. Mei, A. Kinaci, F. G. Sen, M. J. Davis, S. K. Gray, M. K. Y. Chan, and S. K. R. S. Sankaranarayanan. “Development of a Modified Embedded Atom Force Field for Zirconium Nitride Using Multi-Objective Evolutionary Optimization”. In: *The Journal of Physical Chemistry C* 120.31 (July 2016), pp. 17475–17483.

[16] M. J. Cherukara, B. Narayanan, A. Kinaci, K. Sasikumar, S. K. Gray, M. K. Chan, and S. K. R. S. Sankaranarayanan. “Ab Initio-Based Bond Order Potential to Investigate Low Thermal Conductivity of Stanene Nanostructures”. In: *The Journal of Physical Chemistry Letters* 7.19 (Sept. 2016), pp. 3752–3759.

[17] B. Narayanan, H. Chan, A. Kinaci, F. G. Sen, S. K. Gray, M. K. Y. Chan, and S. K. R. S. Sankaranarayanan. “Machine learnt bond order potential to model metal-organic (Co-C) heterostructures”. In: *Nanoscale* 9 (46 2017), pp. 18229–18239.

[18] T. Hey, S. Tansley, and K. Tolle. *The Fourth Paradigm – Data Intensive Scientific Discovery*. 2009.

[19] W. Michener. “Five new Paradigms for Science and an Introduction to DataOne”. In: *EducausE Review* March/April (2012).

[20] M. Mustafa, J. Balewski, J. Lauret, J. Porter, S. Canon, L. Gerhardt, L. Hajdu, and M. Lukascsyk. “STAR Data Reconstruction at NERSC/Cori, an adaptable Docker container approach for HPC”. In: *Journal of Physics: Conference Series* 898.8 (2017), p. 082023.

[21] C. Covey, D. D. Lucas, J. Tannahill, X. Garaizar, and R. Klein. “Efficient screening of climate model sensitivity to a large number of perturbed input parameters”. In: *Journal of Advances in Modeling Earth Systems* 5.3 (2013), pp. 598–610.

[22] J. Najita, B. Willman, D. P. Finkbeiner, R. J. Foley, S. Hawley, J. A. Newman, G. Rudnick, J. D. Simon, D. Trilling, R. Street, A. Bolton, R. Angus, E. F. Bell, D. Buzasi, D. Ciardi, J. R. A. Davenport, W. Dawson, M. Dickinson, A. Drlica-Wagner, J. Elias, D. Erb, L. Feaga, W.-f. Fong, E. Gawiser, M. Giampapa, P. Guhathakurta, J. L. Hoffman, H. Hsieh, E. Jennings, K. V. Johnston, V. Kashyap, T. S. Li, E. Linder, R. Mandelbaum, P. Marshall, T. Matheson, S. Meibom, B. W. Miller, J. O’Meara, V. Reddy, S. Ridgway, C. M. Rockosi, D. J. Sand, C. Schafer, S. Schmidt, B. Sesar, S. S. Sheppard, C. A. Thomas, E. J. Tollerud, J. Trump, and A. von der Linden. *Maximizing Science in the Era of LSST: A Community-Based Study of Needed US Capabilities*. 2016. arXiv: `1610.01661 [astro-ph.IM]`.

[23] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. 1st. 2009.

[24] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. “Mastering the game of Go without human knowledge”. In: *Nature* 550.7676 (2017), pp. 354–359.

[25] B. A. Sen and S. Menon. "Representation of Chemical Kinetics by Artificial Neural Networks for Large Eddy Simulations". In: *43rd AIAA Joint Propulsion Conference*. Cincinatti, 2007.

[26] B. A. Sen and S. Menon. "Turbulent premixed flame modeling using artificial neural networks based chemical kinetics". In: *Proceedings of the Combustion Institute* 32.1 (2009), pp. 1605–1611.

[27] K. Sargsyan, C. Safta, H. N. Najm, B. J. Debusschere, D. Ricciuto, and P. Thornton. "Dimensionality Reduction for Complex Models via Bayesian Compressive Sensing". In: *International Journal for Uncertainty Quantification* 4.1 (2014), pp. 63–93.

[28] C. Soize and R. Ghanem. "Data-Driven Probability Concentration and Sampling on Manifold". In: *Journal of Computational Physics* 321 (2016), pp. 242–258.

[29] M. Milano and P. Koumoutsakos. "Neural Network Modeling for Near Wall Turbulent Flow". In: *Journal of Computational Physics* 182.1 (2002), pp. 1–26.

[30] B. Tracey, K. Duraisamy, and J. Alonso. *Application of supervised learning to quantify uncertainties in turbulence and combustion modeling*. AIAA Aerospace Sciences Meeting, AIAA-2013-0259. 2013.

[31] B. Tracey, K. Duraisamy, and J. Alonso. *A machine learning strategy to assist turbulence model development*. AIAA SciTech, AIAA-2015-1287. 2015.

[32] K. Duraisamy, Z. Shang, and A. Singh. *New approaches in turbulence and transition modeling using data-driven techniques*. AIAA SciTech, AIAA-2015-1284. 2015.

[33] Z. Zhang and K. Duraisamy. *Machine learning methods for data-driven turbulence modeling*. AIAA Aviation, AIAA-2015-2460, 2015. 2015.

[34] J. Ling and J. Templeton. "Evaluation of machine learning algorithms for prediction of regions of high RANS uncertainty". In: *Physics of Fluids* (2015), p. 085103.

[35] J. Ling, R. Jones, and J. Templeton. "Machine learning strategies for systems with invariance properties". In: *Journal of Computational Physics* 318 (2016), pp. 22–35.

[36] J.-L. Wu, J.-X. Wang, H. Xiao, and J. Ling. *A Priori Assessment of Prediction Confidence for Data-Driven Turbulence Modeling*. 2017. arXiv: `1607.04563 [physics.comp-ph]`.

[37] A. Elsheikh, R. Tavakoli, M. Wheeler, and I. Hoteit. "Boosting iterative stochastic ensemble method for nonlinear calibration of subsurface flow models". In: *Computer Methods in Applied Mechanics and Engineering* 259 (2013), pp. 10–23.

[38] J. Ghaboussi, J. G. Jr., and X. Wu. "Knowledge based modeling of material behavior with neural networks". In: *Journal of Engineering Mechanics* 117 (1991), pp. 132–153.

[39] J. Ghaboussi and D. Sidarta. "New Nested Adaptive Neural Networks (NANN) for constitutive modeling". In: *Computers and Geotechnics* 22 (1998), pp. 29–52.

[40] G. Ellis, C. Yao, R. Zhao, and D. Penumadu. "Stress-strain modeling of sands using artificial neural networks". In: *Journal of Geotechnical and Geoenvironmental Engineering* 121 (1995), pp. 429–435.

[41] T. Furukawa and G. Yagawa. "Implicit constitutive modelling for viscoplasticity using neural networks". In: *International Journal for Numerical Methods in Engineering* 43 (1998), pp. 195–219.

[42] Y. Sun, W. Zeng, Y. Zhao, Y. Qi, X. Ma, and Y. Han. "Development of constitutive relationship model of Ti600 alloy using artificial neural network". In: *Computational Materials Science* 48 (2010), pp. 686–691.

[43] S. Jung and J. Ghaboussi. "Neural network constitutive model for rate-dependent materials". In: *Computers & Structures* 84 (2006), pp. 955–963.

[44] X. Feng and C. Yang. "Genetic evolution of nonlinear material constitutive models". In: *Computer Methods in Applied Mechanics and Engineering* 190 (2001), pp. 5957–5973.

[45] Y. Shen, K. Chandrashekhara, W. Breig, and L. Oliver. "Finite element analysis of V-ribbed belts using neural network based hyperelastic material model". In: *International Journal of Non-Linear Mechanics* 40 (2005), pp. 875–890.

[46] G. Liang and K. Chandrashekhara. "Neural network based constitutive model for elastomeric foams". In: *Engineering Structures* 30 (2008), pp. 2002–2011.

[47] V. Demyanov, M. Kanevsky, A. Pozdnoukhov, and M. Christie. *Uncertainty quantification with support vector regression prediction models*. Accuracy 2010 Symposium. July 20-23, Leicester, UK. 2010.

[48] L. Durlofsky and Y. Chen. "Uncertainty quantification for subsurface flow problems using coarse-scale models". In: *Numerical Analysis of Multiscale Problems*. 2012, pp. 163–202.

[49] S. Trehan and L. J. Durlofsky. *Machine-learning-based estimation of upscaling error, with application to uncertainty quantification*. 2017.

[50] S. Trehan, K. Carlberg, and L. Durlofsky. *Error modeling for surrogates of dynamical systems using machine learning*. 2017. arXiv: `1701.03240`.

[51] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. "Constrained k-means clustering with background knowledge". In: *ICML*. Vol. 1. 2001, pp. 577–584.

[52] C. J. S. Webber. "Self-Organization of Symmetry Networks: Transformation Invariance from the Spontaneous Symmetry-Breaking Mechanism". In: *Neural Computation* 12.3 (2000), pp. 565–596.

[53] P. E. Shanahan, D. Trewartha, and W. Detmold. *Machine learning action parameters in lattice quantum chromodynamics*. 2018. arXiv: `1801.05784 [hep-lat]`.

[54] M. Fischetti and J. Jo. *Deep Neural Networks as 0-1 Mixed Integer Linear Programs: A Feasibility Study*. Dec. 2017. arXiv: `1712.06174 [cs.LG]`.

[55] G. Carleo and M. Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325 (2017), pp. 602–606.

[56] J. B. Hamrick, A. J. Ballard, R. Pascanu, O. Vinyals, N. Heess, and P. W. Battaglia. *Metacontrol for adaptive imagination-based optimization*. 2017. arXiv: `1705.02670`.

[57] Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. New York, NY, 1982.

[58] H. Hijazi and L. Liberti. "Constraint qualification failure in action". In: *Operations Research Letters* 44.4 (2016), pp. 503–506.

[59] J. Nocedal and S. J. Wright. *Numerical Optimization*. Second. 2006.

[60] W. Zhang, J. Chien, J. Yong, and R. Kuang. "Network-based machine learning and graph theory algorithms for precision oncology". In: *Precision Oncology* 1.1 (Aug. 2017).

[61] F. J. Király, A. Ziehe, and K.-R. Müller. *Learning with Algebraic Invariances, and the Invariant Kernel Trick*. Nov. 2014. arXiv: `1411.7817 [stat.ML]`.

[62] B. Schölkopf, C. Burges, and V. Vapnik. “Incorporating invariances in support vector learning machines”. In: *Artificial Neural Networks — ICANN 96: 1996 International Conference Bochum, Germany, July 16–19, 1996 Proceedings*. Ed. by C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff. 1996, pp. 47–52.

[63] D. Decoste and B. Schölkopf. “Training Invariant Support Vector Machines”. In: *Machine Learning* 46.1 (Jan. 2002), pp. 161–190.

[64] J. Ling, R. Jones, and J. Templeton. “Machine learning strategies for systems with invariance properties”. In: *Journal of Computational Physics* 318.Supplement C (2016), pp. 22–35.

[65] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[66] I. Dunning, J. Huchette, and M. Lubin. “JuMP: A Modeling Language for Mathematical Optimization”. In: *SIAM Review* 59.2 (2017), pp. 295–320.

[67] Prabhat, O. Rubel, S. Byna, K. Wu, F. Li, M. Wehner, and W. Bethel. “TECA: A Parallel Toolkit for Extreme Climate Analysis”. In: *Procedia Computer Science* 9 (2012), pp. 866–876.

[68] Y. Liu, E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, W. Collins, et al. *Application of deep convolutional neural networks for detecting extreme weather in climate datasets*. 2016. arXiv: 1605.01156.

[69] E. Racah, C. Beckham, T. Maharaj, S. Ebrahimi Kahou, M. Prabhat, and C. Pal. “ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. 2017, pp. 3402–3413.

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 1097–1105.

[71] M. D. Zeiler and R. Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. 2014, pp. 818–833.

[72] DARPA. *Explainable artificial intelligence (XAI)*. https://www.darpa.mil/program/explainable-artificial-intelligence. 2017.

[73] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. “Visual Exploration of High Dimensional Scalar Functions”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1271–1280.

[74] S. Gerber, O. Rübel, P.-T. Bremer, V. Pascucci, and R. Whitaker. “Morse-Smale Regression”. In: *Computational and Graphics Statistics* 22.1 (2013), pp. 193–214.

[75] C. Covey, D. D. Lucas, J. Tannahill, X. Garaizar, and R. Klein. “Efficient screening of climate model sensitivity to a large number of perturbed input parameters”. In: *Journal of Advances in Modeling Earth Systems* 5.3 (2013), pp. 598–610.

[76] F. Doshi-Velez and B. Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. Feb. 2017. arXiv: `1702.08608 [stat.ML]`.

[77] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. Nov. 2017. arXiv: `1711.05225 [cs.CV]`.

[78] S. Rhee, S. Seo, and S. Kim. *Hybrid Approach of Relation Network and Localized Graph Convolutional Filtering for Breast Cancer Subtype Classification*. Nov. 2017. arXiv: `1711.05859 [cs.CV]`.

[79] A. Kurakin, I. Goodfellow, and S. Bengio. *Adversarial examples in the physical world*. Feb. 2017. arXiv: `1607.02533 [cs.CV]`.

[80] J. Su, D. V. Vargas, and S. Kouichi. *One pixel attack for fooling deep neural networks*. Oct. 2017. arXiv: `1710.08864 [cs.LG]`.

[81] H. Wang, W. Zhang, F. Sun, and W. Zhang. "A Comparison Study of Machine Learning Based Algorithms for Fatigue Crack Growth Calculation". In: *Materials* 10.5 (2017), p. 543.

[82] M. Araujo, S. Siebenaler, S. Baldor, E. Dupont, D. Davila, and S. Blaisdell. "Automated Small Leak Detection from Hazardous Liquid Pipelines Using Multi-Platform Remote Sensing". In: *11th Pipeline Technology Conference 2016*. 2016.

[83] M. Kubat, R. C. Holte, and S. Matwin. "Machine learning for the detection of oil spills in satellite radar images". In: *Machine Learning* 30.2-3 (1998), pp. 195–215.

[84] B. Shneiderman. "Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight". In: *Proceedings of the National Academy of Sciences* 113.48 (Nov. 2016), pp. 13538–13540.

[85] J. Wang, J. Oh, and J. Wiens. *Learning Credible Models*. Nov. 2017. arXiv: `1711.03190 [cs.LG]`.

[86] M. A. Kocak, D. Ramirez, E. Erkip, and D. E. Shasha. *SafePredict: A Meta-Algorithm for Machine Learning That Uses Refusals to Guarantee Correctness*. Nov. 2017. arXiv: `1708.06425 [cs.LG]`.

[87] M. Hein and M. Andriushchenko. "Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. 2017, pp. 2263–2273.

[88] P. J. Roache. "Code Verification by the Method of Manufactured Solutions". In: *Trans. ASME, J. Fluids Engineering* 124 (2002), pp. 4–10.

[89] E. B. Baum and D. Haussler. "What Size Net Gives Valid Generalization?" In: *Neural Computation* 1.1 (Mar. 1989), pp. 151–160.

[90] T. Hofmann, B. Schulkopf, and A. J. Smola. "Kernel methods in machine learning". In: *The Annals of Statistics* 36.3 (June 2008), pp. 1171–1220.

[91] B. Yu. "Stability". In: *Bernoulli* 19 (4) (2013), pp. 1484–1500.

[92] R. Boisvert. In: *Communications of the ACM* 59 (2016).

[93] M. Hutson. "Artificial intelligence faces reproducibility crisis". In: *Science* 359.6377 (Feb. 2018), pp. 725–726.

[94] M. Baker. “1500 Scientists Lift the Lid on Reproducibility”. In: *Nature* 533.7604 (2016), pp. 452–454.

[95] In: *Nature Reviews Drug Discovery* 10 (2011), pp. 643–644.

[96] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. 2000.

[97] J. Ling, A. Kurzawski, and J. Templeton. “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance”. In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166.

[98] Y. Gal and Z. Ghahramani. *Bayesian convolutional neural networks with Bernoulli approximate variational inference*. 2016. arXiv: `1506.02158v6`.

[99] Z. Ghahramani. “Probabilistic machine learning and artificial intelligence”. In: *Nature* 521.7553 (2015), pp. 452–459.

[100] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. “Bayesian Optimization in a Billion Dimensions via Random Embeddings”. In: *Journal of Artificial Intelligence Research* 55 (2016), pp. 361–387.

[101] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680.

[102] Y. Saatchi and A. G. Wilson. *Bayesian GAN*. 2017. arXiv: `1705.09558v2`.

[103] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. “Deep Kernel Learning”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain*. 2016.

[104] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1966), pp. 268–288.

[105] J. H. Friedman. “Fast sparse regression and classification”. In: *International Journal of Forecasting* 28.3 (2012), pp. 722–738.

[106] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. 2015.

[107] D. L. Donoho. “For most large underdetermined systems of linear equations the minimal $\ell1$-norm solution is also the sparsest solution”. In: *Communications on Pure and Applied Mathematics* 59.6 (2006), pp. 797–829.

[108] D. L. Donoho. “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306.

[109] E. J. Candes, J. K. Romberg, and T. Tao. “Stable signal recovery from incomplete and inaccurate measurements”. In: *Communications on Pure and Applied Mathematics* 59.8 (2006), pp. 1207–1223.

[110] L. Grasedyck, D. Kressner, and C. Tobler. “A literature survey of low-rank tensor approximation techniques”. In: *GAMM-Mitteilungen* 36.1 (2013), pp. 53–78.

[111] L. Wang, M. T. Chu, and B. Yu. “Orthogonal Low Rank Tensor Approximation: Alternating Least Squares Method and Its Global Convergence”. In: *SIAM Journal on Matrix Analysis and Applications* 36.1 (2015), pp. 1–19.

[112] W. Dong, G. Li, G. Shi, X. Li, and Y. Ma. “Low-Rank Tensor Approximation with Laplacian Scale Mixture Modeling for Multiframe Image Denoising”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 442–449.

[113] A. Nouy. “Low-Rank Tensor Methods for Model Order Reduction”. In: *Handbook of Uncertainty Quantification*. Ed. by R. Ghanem, D. Higdon, and H. Owhadi. 2016, pp. 1–26.

[114] M. Chevreuil, R. Lebrun, A. Nouy, and P. Rai. “A Least-Squares Method for Sparse Low Rank Approximation of Multivariate Functions”. In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 897–921.

[115] D. Kressner, R. Kumar, F. Nobile, and C. Tobler. “Low-Rank Tensor Approximation for High-Order Correlation Functions of Gaussian Random Fields”. In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 393–416.

[116] S. L. Brunton, J. L. Proctor, and J. N. Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937.

[117] C. Gear, I. G. Kevrekidis, and C. Theodoropoulos. “‘Coarse’ integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods”. In: *Computers & Chemical Engineering* 26.7 (2002), pp. 941–963.

[118] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. “Equation-Free, Coarse-Grained Multiscale Computation: Enabling Mocroscopic Simulators to Perform System-Level Analysis”. In: *Communications in Mathematical Sciences* 1.4 (2003), pp. 715–762.

[119] D. Givon, R. Kupferman, and A. Stuart. “Extracting macroscopic dynamics: model problems and algorithms”. In: *Nonlinearity* 17 (2004), pp. 55–127.

[120] G. Hinton and R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science* 313 (2006), pp. 504–507.

[121] I. Jollife. *Principal Component Analysis*. New York, 1986.

[122] M. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 37 (1991), pp. 233–243.

[123] E. Candès, X. Li, Y. Ma, and J. Wright. “Robust principal component analysis?” In: *J. ACM* 58.3 (2011), pp. 1–37.

[124] S. T. Roweis and L. K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326.

[125] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. “Diffusion Maps, Reduction Coordinates, and Low Dimensional Representation of Stochastic Systems”. In: *Multiscale Modeling & Simulation* 7.2 (2008), pp. 842–864.

[126] A. V. Little, J. D. Lee, Y.-M. Jung, and M. Maggioni. “Estimation of Intrinsic Dimensionality of Samples from Noisy Low-Dimensional Manifolds in High Dimensions with Multiscale SVD”. In: *IEEE Workshop on Statistical Signal Processing (SSP)*. 2009, pp. 85–88.

[127] C. Soize and R. Ghanem. “Data-driven probability concentration and sampling on manifold”. In: *Journal of Computational Physics* 321 (2016), pp. 242–258.

[128] C. Soize and R. Ghanem. “Polynomial chaos representation of databases on manifolds”. In: *Journal of Computational Physics* 335 (2017), pp. 201–221.

[129] S. Byrne and M. Girolami. “Geodesic Monte Carlo on Embedded Manifolds”. In: *Scandinavian Journal of Statistics* 40.4 (2013), pp. 825–845.

[130] T. Craciunescu and A. Murari. “Geodesic distance on Gaussian manifolds for the robust identification of chaotic systems”. In: *Nonlinear Dynamics* 86.1 (Oct. 2016), pp. 677–693.

[131] B. Ganapathysubramanian and N. Zabaras. "A non-linear dimension reduction methodology for generating data-driven stochastic input models". In: *Journal of Computational Physics* 227.13 (2008), pp. 6612–6637.

[132] M. Girolami and B. Calderhead. "Riemann manifold Langevin and Hamiltonian Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (2011), pp. 123–214.

[133] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323.

[134] G. J. Szekely, M. L. Rizzo, and N. K. Bakirov. "Measuring and testing dependence by correlation of distances". In: *The Annals of Statistics* 35.6 (Dec. 2007), pp. 2769–2794.

[135] M. Abbas, A. Ilin, A. Solonen, J. Hakkarainen, E. Oja, and H. Järvinen. "Bayesian optimization for tuning chaotic systems". In: *Nonlinear Processes in Geophysics Discussions* 1 (2014), pp. 1283–1312.

[136] H. Haario, L. Kalachev, and J. Hakkarainen. "Generalized correlation integral vectors: A distance concept for chaotic dynamical systems". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.6 (2015), p. 063102.

[137] J. Pearl. "Causal inference in statistics: An overview". In: *Statistics Surveys* 3 (2009), pp. 96–146.

[138] J. Runge, V. Petoukhov, J. F. Donges, J. Hlinka, N. Jajcay, M. Vejmelka, D. Hartman, N. Marwan, M. Palus, and J. Kurths. "Identifying causal gateways and mediators in complex spatio-temporal systems". In: *Nature Communications* 6 (2015), pp. 1–10.

[139] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. "Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks". In: *Journal of Machine Learning Research* 17.32 (2016), pp. 1–102.

[140] A. Feller and A. Gelman. "Hierarchical Models for Causal Effects". In: *Emerging Trends in the Social and Behavioral Sciences*. Ed. by R. Scott and S. Kosslyn. 2015, pp. 1–16.

[141] P. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Scolkopf. "Nonlinear causal discovery with additive noise models". In: *Advances in Neural Information Processing Systems (Neural Information Processing Systems 2008)* 21 (2009), pp. 689–696.

[142] P. Daniusis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Scölkopf. "Inferring deterministic causal relations". In: *Proceedings of the 26th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2010)*. 2010, pp. 143–150.

[143] J. Pearl. *Causality: Models, Reasoning, and Inference*. 2009.

[144] K. P. Burnam and D. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. 2003.

[145] L. Wasserman. "Bayesian model selection and model averaging". In: *Journal of Mathematical Psychology* 44.1 (2000), pp. 92–107.

[146] B. Settles. *Active Learning*. 2012.

[147] K. Chaloner and I. Verdinelli. "Bayesian Experimental Design: A Review". In: *Statistical Science* 10.3 (1995), pp. 273–304.

[148] T. Corcoran, R. Zamora-Resendiz, X. Liu, and S. Crivelli. *A Spatial Mapping Algorithm with Applications in Deep Learning-Based Structure Classification*. Feb. 2018. arXiv: `1802.02532 [cs.CV]`.

[149] G. Söderlind. "Automatic Control and Adaptive Time-Stepping". In: *Numerical Algorithms* 31.1 (Dec. 2002), pp. 281–310.

[150] M. J. Berger and P. Colella. "Local adaptive mesh refinement for shock hydrodynamics". In: *Journal of Computational Physics* 82 (May 1989), pp. 64–84.

[151] J. P. Boris and D. L. Book. "Flux-Corrected Transport. III. Minimal-Error FCT Algorithms". In: *Journal of Computational Physics* 20 (Apr. 1976), pp. 397–431.

[152] B. van Leer. "Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method". In: *Journal of Computational Physics* 32 (July 1979), pp. 101–136.

[153] P. Colella and P. R. Woodward. "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations". In: *Journal of Computational Physics* 54 (Sept. 1984), pp. 174–201.

[154] C.-W. Shu and S. Osher. "Efficient Implementation of Essentially Non-oscillatory Shock-capturing Schemes". In: *Journal of Computational Physics* 77.2 (Aug. 1988), pp. 439–471.

[155] *Automatically Tuned Linear Algebra Software (ATLAS)*. `http://math-atlas.sourceforge.net`.

[156] *RAJA: Managing Application Portability for Next-Generation Platforms*. `https://go.usa.gov/xE9e4`.

[157] *Kokkos*. `https://github.com/kokkos`.

[158] *Apollo: Fast, Lightweight, Dynamic Tuning for Data-Dependent Code*. `https://computation.llnl.gov/projects/apollo`.

[159] M. Raissi, P. Perdikaris, and G. E. Karniadakis. *Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations*. 2017. arXiv: `1711.10566`.

[160] S. Bubeck. "Convex Optimization: Algorithms and Complexity". In: *Foundations and Trends in Machine Learning* 8.3-4 (2015), pp. 231–357.

[161] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". In: *Journal of Machine Learning Research* 12 (2011), pp. 2121–2159.

[162] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. "Oracle-based robust optimization via online learning". In: *Operations Research* 63.3 (2015), pp. 628–638.

[163] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi. "Dropping Convexity for Faster Semi-definite Optimization". In: *29th Annual Conference on Learning Theory*. Ed. by V. Feldman, A. Rakhlin, and O. Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, NY, 2016, pp. 530–582.

[164] B. Kulis, S. Sra, and I. Dhillon. "Convex Perturbations for Scalable Semidefinite Programming". In: *JMLR Workshop and Conference Proceedings 5*. Ed. by M. W. van Dyk D. Apr. 2009, pp. 296–303.

[165] R. Byrd, G. Chin, J. Nocedal, and Y. Wu. "Sample size selection in optimization methods for machine learning". In: *Mathematical Programming* 134.1 (2012), pp. 127–155.

[166] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. "Optimal Rates for Zero-Order Convex Optimization: The Power of Two Function Evaluations". In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2788–2806.

[167] O. Shamir. “On the Complexity of Bandit and Derivative-Free Stochastic Convex Optimization”. In: *Proceedings of the 26th Annual Conference on Learning Theory*. Ed. by S. Shalev-Shwartz and I. Steinwart. Vol. 30. Proceedings of Machine Learning Research. Princeton, NJ, USA, 2013, pp. 3–24.

[168] A. D. Bull. “Convergence Rates of Efficient Global Optimization Algorithms”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2879–2904.

[169] L. Bottou. “Large-Scale Machine Learning with Stochastic Gradient Descent”. In: *Proceedings of COMPSTAT 2010*. 2010, pp. 177–186.

[170] A. Agarwal and J. C. Duchi. “Distributed delayed stochastic optimization”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 873–881.

[171] A. Defazio, F. R. Bach, and S. Lacoste-Julien. “SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. 2014, pp. 1646–1654.

[172] M. Schmidt, N. L. Roux, and F. Bach. “Minimizing finite sums with the stochastic average gradient”. In: *Mathematical Programming* 162.1-2 (June 2016), pp. 83–112.

[173] M. A. Zinkevich, M. Weimer, A. Smola, and L. Li. “Parallelized Stochastic Gradient Descent”. In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*. Neural Information Processing Systems’10. Vancouver, British Columbia, Canada, 2010, pp. 2595–2603.

[174] M. Li, T. Zhang, Y. Chen, and A. J. Smola. “Efficient mini-batch training for stochastic optimization”. In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14, New York, NY, USA - August 24 - 27, 2014*. 2014, pp. 661–670.

[175] M. Takac, A. Bijral, P. Richtarik, and N. Srebro. “Mini-batch Primal and Dual Methods for SVMs”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA, 2013, pp. III-1022–III-1030.

[176] A. S. Berahas, J. Nocedal, and M. Takac. “A Multi-Batch L-BFGS Method for Machine Learning”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. 2016, pp. 1055–1063.

[177] B. Recht, C. Re, S. Wright, and F. Niu. “Hogwild: A lock-free approach to parallelizing stochastic gradient descent”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 693–701.

[178] J. Liu, S. J. Wright, C. Re, V. Bittorf, and S. Sridhar. “An Asynchronous Parallel Stochastic Coordinate Descent Algorithm”. In: *Journal of Machine Learning Research* 16 (2015), pp. 285–322.

[179] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. 2013.

[180] P. Tseng. “Approximation accuracy, gradient methods, and error bound for structured convex optimization”. In: *Mathematical Programming* 125.2 (Aug. 2010), pp. 263–295.

[181] A. Aravkin, J. V. Burke, A. Chiuso, and G. Pillonetto. “Convex vs. Non-Convex Estimators for Regression and Sparse Estimation: The Mean Squared Error Properties of ARD and GLasso”. In: *Journal of Machine Learning Research* 15 (2014), pp. 217–252.

[182] R. Collobert, F. Sinz, J. Weston, and L. Bottou. "Trading convexity for scalability". In: *Proceedings of the 23rd International Conference on Machine Learning*. 2006, pp. 201–208.

[183] P.-L. Loh and M. J. Wainwright. "High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity". In: *The Annals of Statistics* 40.3 (June 2012), pp. 1637–1664.

[184] S. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. "Stochastic Variance Reduction for Nonconvex Optimization". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. 2016, pp. 314–323.

[185] Z. Allen-Zhu and E. Hazan. "Variance Reduction for Faster Non-Convex Optimization". In: *International Conference on Machine Learning*. 2016.

[186] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. "Trust region policy optimization". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2015, pp. 1889–1897.

[187] H. Ghanbari and K. Scheinberg. *Black-Box Optimization in Machine Learning with Trust Region Based Derivative Free Algorithm*. 2017. arXiv: `1703.06925 [cs.LG]`.

[188] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (Jan. 2009), pp. 183–202.

[189] K. Gregor and Y. LeCun. "Learning Fast Approximations of Sparse Coding". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. 2010, pp. 399–406.

[190] Y. LeCun, I. Kanter, and S. A. Solla. "Second-order properties of error surfaces: Learning time and generalization". In: *Advances in Neural Information Processing Systems*. Vol. 3. 1991, pp. 913–924.

[191] S. Becker and Y. LeCun. "Improving the Convergence of Back-Propagation Learning with Second Order Methods". In: *Proceedings of the 1988 Connectionist Models Summer School*. Ed. by D. S. Touretzky, G. E. Hinton, and T. J. Sejnowski. 1989, pp. 29–37.

[192] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. "A Stochastic Quasi-Newton Method for Large-Scale Optimization". In: *SIAM Journal on Optimization* 26.2 (Jan. 2016), pp. 1008–1031.

[193] M. Jiang, B. Gallagher, J. Kallman, and D. Laney. "A Supervised Learning Framework for Arbitrary Lagrangian-Eulerian Simulations". In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2016, pp. 977–982.

[194] B. Peherstorfer, K. Willcox, and M. Gunzburger. "Survey of multifidelity methods in uncertainty propagation, inference, and optimization". In: *SIAM Review* 60.3 (2018), pp. 550–591.

[195] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. 2003.

[196] D. R. Jones, M. Schonlau, and W. J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions". In: *Journal of Global Optimization* 13.4 (1998), pp. 455–492.

[197] B. J. Bichon, M. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. "Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions". In: *AIAA Journal* 46 (Oct. 2008), pp. 2459–2468.

[198] J. Mockus, V. Tiesis, and A. Zilinskas. "The application of Bayesian methods for seeking the extremum". In: *Towards Global Optimization* 2.117-129 (1978), p. 2.

[199] D. R. Jones. "A taxonomy of global optimization methods based on response surfaces". In: *Journal of Global Optimization* 21.4 (2001), pp. 345–383.

[200] J. Snoek, H. Larochelle, and R. P. Adams. "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2951–2959.

[201] J. Villemonteix, E. Vazquez, and E. Walter. "An informational approach to the global optimization of expensive-to-evaluate functions". In: *Journal of Global Optimization* 44.4 (2009), pp. 509–534.

[202] P. Hennig and C. J. Schuler. "Entropy search for information-efficient global optimization". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 1809–1837.

[203] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. "Predictive entropy search for efficient global optimization of black-box functions". In: *Advances in Neural Information Processing Systems*. 2014, pp. 918–926.

[204] R. R. Lam, K. E. Willcox, and D. H. Wolpert. "Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach". In: *Advances in Neural Information Processing Systems*. 2016, pp. 883–891.

[205] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier. "Bayesian optimization with gradients". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5273–5284.

[206] M. Poloczek, J. Wang, and P. Frazier. "Multi-information source optimization". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4291–4301.

[207] B. Peherstorfer, K. Willcox, and M. Gunzburger. "Optimal Model Management for Multifidelity Monte Carlo Estimation". In: *SIAM Journal on Scientific Computing* 38.5 (2016), A3163–A3194.

[208] L. L. Gratiet, C. Cannamela, and B. Iooss. "A Bayesian Approach for Global Sensitivity Analysis of (Multifidelity) Computer Codes". In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1 (2014), pp. 336–363.

[209] L. M. Kegelmeyer, R. Clark, R. R. Leach, D. McGuigan, V. M. Kamm, D. Potter, J. T. Salmon, J. Senecal, A. Conder, M. Nostrand, and P. K. Whitman. "Automated optics inspection analysis for NIF". In: *Fusion Engineering and Design*. Proceedings of the 8th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research 87.12 (Dec. 2012), pp. 2120–2124.

[210] O. Subasi, S. Di, P. Balaprakash, O. Unsal, J. Labarta, A. Cristal, S. Krishnamoorthy, and F. Cappello. "MACORD: Online Adaptive Machine Learning Framework for Silent Error Detection". In: *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. Sept. 2017, pp. 717–724.

[211] O. Tuncer, E. Ates, Y. Zhang, A. Turk, J. Brandt, V. J. Leung, M. Egele, and A. K. Coskun. "Diagnosing Performance Variations in HPC Applications Using Machine Learning". In: *International Supercomputing Conference*. 2017, pp. 355–373.

[212] H. Shoukourian, T. Wilde, D. Labrenz, and A. Bode. "Using Machine Learning for Data Center Cooling Infrastructure Efficiency Prediction". In: *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*. 2017, pp. 954–963.

[213] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. “Machine Learning: A Review of Classification and Combining Techniques”. In: *Artificial Intelligence Review* 26.3 (Nov. 2006), pp. 159–190.

[214] Y. Bengio, A. Courville, and P. Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (Aug. 2013), pp. 1798–1828.

[215] M. Berry, T. E. Potok, P. Balaprakash, H. Hoffmann, R. Vatsavai, and Prabhat. *Machine learning and understanding for intelligent extreme scale scientific computing and discovery*. Workshop Report. DOE ASCR, 2015.

[216] P. Gupta, A. Sharma, and R. Jindal. “Scalable machine-learning algorithms for big data analytics: a comprehensive review”. In: *WIREs Data Mining and Knowledge Discovery* 6 (2016), pp. 194–214.

[217] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. “Efficient Processing of Deep Neural Networks: A Tutorial and Survey”. In: *Proceedings of the IEEE* 105.12 (Dec. 2017), pp. 2295–2329.

[218] F. Williams. *Combustion Theory*. 2nd. New York, 1985.

[219] H. Pitsch. “Large-Eddy Simulation of Turbulent Combustion”. In: *Annual Review of Fluid Mechanics* 38 (2006), pp. 453–482.

[220] P. A. Durbin and B. A. P. Reif. *Statistical Theory and Modeling for Turbulent Flows*. 2nd. 2011.

[221] *U.S. Department of Energy Office of Science Biological and Environmental Research (BER)*. `https://science.energy.gov/ber/`. 2017.

[222] R. Stevens. *CANDLE: Exascale Deep Learning and Simulation Enabled Precision Medicine for Cancer*. `http://candle.cels.anl.gov`.

[223] S. E. Peters, C. Zhang, M. Livny, and C. Re. “A machine reading system for assembling synthetic paleontological databases”. In: *PLoS One* 9.12 (2014), e113523.

[224] C. Zhang, V. Govindaraju, J. Borchardt, T. Foltz, C. Ré, and S. Peters. “GeoDeepDive: statistical inference using familiar data-processing languages”. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013, pp. 993–996.

[225] R. B. Altman. “Predicting cancer drug response: advancing the DREAM”. In: *Cancer Discovery* 5.3 (2015), pp. 237–238.

[226] M. Vidyasagar. “Identifying predictive features in drug response using machine learning: opportunities and challenges”. In: *Annual Review of Pharmacology and Toxicology* 55 (2015), pp. 15–34.

[227] C. Zhang, C. Re, M. Cafarella, C. De Sa, A. Ratner, J. Shin, F. Wang, and S. Wu. “DeepDive: Declarative Knowledge Base Construction”. In: *Communications of the ACM* 60.5 (May 2017), pp. 93–102.

[228] H. I. Ingolfsson, M. N. Melo, F. J. van Eerden, C. Arnarez, C. A. Lopez, T. A. Wassenaar, X. Periole, A. H. de Vries, D. P. Tieleman, and S. J. Marrink. “Lipid Organization of the Plasma Membrane”. In: *Journal of the American Chemical Society* 136.41 (2014), pp. 14554–14559.

[229] H. I. Ingolfsson, C. A. Lopez, J. J. Uusitalo, D. H. de Jong, S. M. Gopal, X. Periole, and S. J. Marrink. “The power of coarse graining in biomolecular simulations”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.3 (2014), pp. 225–248.

[230] A. Ramanathan, A. J. Savol, C. J. Langmead, P. K. Agarwal, and C. S. Chennubhotla. "Discovering Conformational Sub-States Relevant to Protein Function". In: *PLOS ONE* 6.1 (Jan. 2011), pp. 1–16.

[231] L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton. "Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations". In: *Physical Review B* 96 (2 July 2017), p. 024104.

[232] R. Liu, Y. C. Yabansu, Z. Yang, A. N. Choudhary, S. R. Kalidindi, and A. Agrawal. "Context Aware Machine Learning Approaches for Modeling Elastic Localization in Three-Dimensional Composite Microstructures". In: *Integrating Materials and Manufacturing Innovation* (2017), pp. 1–12.

[233] P. Raccuglia, K. C. Elbert, P. D. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist. "Machine-learning-assisted materials discovery using failed experiments". In: *Nature* 533.7601 (2016), pp. 73–76.

[234] J. A. Gomberg, A. J. Medford, and S. R. Kalidindi. "Extracting knowledge from molecular mechanics simulations of grain boundaries using machine learning". In: *Acta Materialia* 133 (2017), pp. 100–108.

[235] R. B. Tchoua, K. Chard, D. J. Audus, L. T. Ward, L. Joshua, J. J. de Pablo, and I. Foster. "Towards a Hybrid Human-Computer Scientific Information Extraction Pipeline". In: *2017 IEEE 13th International Conference on e-Science*. 2017.

[236] *Materials Genome Initiative*. `https://www.mgi.gov/`. 2017.

[237] R. L. Martin, T. F. Willems, L.-C. Lin, J. Kim, J. A. Swisher, B. Smit, and M. Haranczyk. "Similarity-Driven Discovery of Zeolite Materials for Adsorption-Based Separations". In: *ChemPhysChem* 13.16 (Nov. 2012), pp. 3561–3561.

[238] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. "Multimodal deep learning". In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. 2011, pp. 689–696.

[239] A. Gooneie, S. Schuschnigg, and C. Holzer. "A Review of Multiscale Computational Methods in Polymeric Materials". In: *Polymers* 9.1 (2017).

[240] M. F. Horstemeyer. "Multiscale Modeling: A Review". In: *Practical Aspects of Computational Chemistry: Methods, Concepts and Applications*. Ed. by J. Leszczynski and M. K. Shukla. Dordrecht, 2010, pp. 87–135.

[241] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden. "Heterogeneous multiscale methods: A review". In: *Communications in Computational Physics* 2.3 (2007), pp. 367–450.

[242] E. B. Tadmor, M. Ortiz, and R. Phillips. "Quasicontinuum analysis of defects in solids". In: *Philosophical Magazine A* 73.6 (1996), pp. 1529–1563.

[243] F. F. Abraham, J. Q. Broughton, N. Bernstein, and E. Kaxiras. "Spanning the continuum to quantum length scales in a dynamic simulation of brittle fracture". In: *EPL (Europhysics Letters)* 44.6 (1998), p. 783.

[244] J. Knap, N. R. Barton, R. D. Hornung, A. Arsenlis, R. Becker, and D. R. Jefferson. "Adaptive sampling in hierarchical simulation". In: *International Journal for Numerical Methods in Engineering* 76.4 (2008), pp. 572–600.

[245] N. R. Barton, J. Knap, A. Arsenlis, R. Becker, R. D. Hornung, and D. R. Jefferson. "Embedded polycrystal plasticity and adaptive sampling". In: *International Journal of Plasticity* 24.2 (2008), pp. 242–266.

[246] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. “Deep Convolutional Inverse Graphics Network”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. 2015, pp. 2539–2547.

[247] D. Kingma, S. Mohamed, D. Rezende, and M. Welling. “Semi-supervised learning with deep generative models”. In: *Proceedings of Advances in Neural Information Processing Systems (Neural Information Processing Systems)*. 2014, pp. 3581–3589.

[248] L. v. d. Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.11 (2008), pp. 2579–2605.

[249] J. P. Cunningham and Z. Ghahramani. “Linear Dimensionality Reduction: Survey, Insights, and Generalizations”. In: *Journal of Machine Learning Research* 16 (2015), pp. 2859–2900.

[250] L. van der Maaten, E. O. Postma, and H. J. van den Herik. *Dimensionality Reduction: A Comparative Review*. 2008.

[251] B. Kulis. “Metric Learning: A Survey”. In: *Foundations and Trends in Machine Learning* 5.4 (2013), pp. 287–364.

[252] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. *Caffe: Convolutional Architecture for Fast Feature Embedding*. June 2014. arXiv: `1408.5093 [cs]`.

[253] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. Sept. 2014. arXiv: `1409.0575 [cs]`.

[254] T. Kurth, J. Zhang, N. Satish, E. Racah, I. Mitliagkas, M. M. A. Patwary, T. Malas, N. Sundaram, W. Bhimji, M. Smorkalov, J. Deslippe, M. Shiryaev, S. Sridharan, Prabhat, and P. Dubey. “Deep Learning at 15PF: Supervised and Semi-supervised Classification for Scientific Data”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC ’17. New York, NY, USA, 2017, 7:1–7:11. ISBN: 978-1-4503-5114-0.

[255] E. Learned-Miller, G. Huang, A. Roychowdhury, H. Li, G. Hua, E. Learned-miller, G. B. Huang, A. Roychowdhury, and H. Li. *Labeled Faces in the Wild: A Survey*. 2015.

[256] X.-W. Chen and X. Lin. “Big Data Deep Learning: Challenges and Perspectives”. In: *IEEE Access* 2 (2014), pp. 514–525.

[257] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521 (2015), pp. 436–444.

[258] B. E. Boser, I. M. Guyon, and V. N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT ’92. Pittsburgh, Pennsylvania, USA, 1992, pp. 144–152.

[259] C. Cortes and V. Vapnik. “Support-vector networks”. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297.

[260] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton, NJ, USA, 2010.

[261] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. 2005.

[262] M. Kanevski, A. Pozdnoukhov, and V. Timonin. “Probabilistic neural network Classification model based on maximum posterior decision rule”. In: *Machine Learning for Environmental Data* (2009).

[263] R. M. Neal. "Bayesian learning for neural networks". PhD thesis. University of Toronto, 1995.

[264] D. J. MacKay. "A practical Bayesian framework for backpropagation networks". In: *Neural Computation* 4.3 (1992), pp. 448–472.

[265] J. Denker and Y. LeCun. *Transforming neural-net output levels to probability distributions*. Advances in Neural Information Processing Systems 3. 1991.

[266] A. Graves. *Practical variational inference for neural networks*. Advances in Neural Information Processing Systems. 2011.

[267] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. "Weight Uncertainty in Neural Networks". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. 2015, pp. 1613–1622. arXiv: `1505.05424`.

[268] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*. Tech. rep. DTIC Document, 1985.

[269] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[270] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions*. 2014. arXiv: `1409.4842 [cs.CV]`.

[271] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to sequence learning with neural networks*. Neural Information Processing Systems. 2014.

[272] M. Sundermeyer, R. Schlüter, and H. Ney. INTERSPEECH. 2012.

[273] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur. *Recurrent neural network based language model*. Eleventh Annual Conference of the International Speech Communication Association. 2010.

[274] N. Kalchbrenner and P. Blunsom. *Recurrent continuous translation models*. EMNLP. 2013.

[275] H. Wang and D.-Y. Yeung. "Towards Bayesian Deep Learning: A Framework and Some Existing Methods". In: *IEEE Transactions on Knowledge and Data Engineering* 28.12 (2016), pp. 3395–3408. arXiv: `1604.01662`.

[276] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. "An introduction to variational methods for graphical models". In: *Machine Learning* 37.2 (1999), pp. 183–233.

[277] Y. Gal and Z. Ghahramani. "On Modern Deep Learning and Variational Inference". In: *Advances in Approximate Bayesian Inference workshop, Neural Information Processing Systems*. 2015, pp. 1–9.

[278] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors*. 2012. arXiv: `1207.0580`.

[279] Y. Gal and Z. Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2015. arXiv: `1506.02142v1 [stat.ML]`.

[280] Y. Li and Y. Gal. *Dropout Inference in Bayesian Neural Networks with Alpha-divergences*. Proceedings of the 34 th International Conference on Machine Learning. Sydney, Australia, PMLR 70. 2017.

[281] P. J. Werbos. "Backwards differentiation in AD and neural nets: Past links and new opportunities". In: *Automatic Differentiation: Applications, Theory, and Implementations*. 2006, pp. 15–34.

[282] A. E. Bryson. "A gradient method for optimizing multi-stage allocation processes". In: *Proc. Harvard Univ. Symposium on Digital Computers and their Applications*. 1961.

[283] H. J. Kelley. "Gradient Theory of Optimal Flight Paths". In: *ARS Journal* 30.10 (1960), pp. 947–954.

[284] S. E. Dreyfus. "The numerical solution of variational problems". In: *Journal of Mathematical Analysis and Applications* 5 (1962), pp. 30–45.

[285] Y. LeCun. "Une procedure d'apprentissage pour reseau a seuil asymetrique". In: *Proceedings of Cognitiva 85, Paris* (1985), pp. 599–604.

[286] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551.

[287] T. Cui, K. J. Law, and Y. M. Marzouk. "Dimension-independent likelihood-informed MCMC". In: *Journal of Computational Physics* 304 (2016), pp. 109–137.

[288] T. Cui, Y. Marzouk, and K. Willcox. "Scalable posterior approximations for large-scale Bayesian inverse problems via likelihood-informed parameter and state reduction". In: *Journal of Computational Physics* 315 (2016), pp. 363–387.

[289] Y. Chen, D. Keyes, K. Law, and H. Ltaief. "Accelerated dimension-independent adaptive Metropolis". In: *SIAM Journal on Scientific Computing* 38.5 (2016), S539–S565.

[290] A. Beskos, M. Girolami, S. Lan, P. E. Farrell, and A. M. Stuart. "Geometric MCMC for infinite-dimensional inverse problems". In: *Journal of Computational Physics* 335 (2017), pp. 327–351.

[291] M. Clyde and E. I. George. "Empirical Bayes Estimation in Wavelet Nonparametric Regression". In: *Bayesian Inference in Wavelet-Based Models*. Ed. by P. Müller and B. Vidakovic. 1999, pp. 309–322.

[292] J. K. Ghosh and R. V. Ramamoorthi. *Bayesian Nonparametrics*. New York, 2003.

[293] P. Müller and F. Quintana. "Nonparametric Bayesian data analysis". In: *Statistical Science* 19.1 (2004), pp. 95–110.

[294] K. Heller and Z. Ghahramani. "Bayesian Hierarchical Clustering". In: *Twenty Second International Conference on Machine Learning (ICML-2005)*. 2005.

[295] Y. Teh and M. Jordan. "Hierarchical Bayesian nonparametric models with applications". In: *Bayesian Nonparametrics*. Ed. by N. Hjort, C. Holmes, P. Müller, and S. Walker. 2010.

[296] F. Pinski, G. Simpson, A. Stuart, and H. Weber. "Kullback-Leibler Approximation for Probability Measures on Infinite Dimensional Spaces". In: *SIAM Journal on Mathematical Analysis* 47.6 (2015), pp. 4091–4122.

[297] D. Chickering. "Learning Bayesian networks is NP-complete". In: *Lecture Notes in Statistics*. 1996, pp. 121–130.

[298] J. Steinhardt and Z. Ghahramani. *Pathological properties of deep Bayesian hierarchies*. Neural Information Processing Systems Workshop on Bayesian Nonparametrics. 2011.

[299] A. Gelman. "Induction and deduction in bayesian data analysis". In: *Rationality Markets and Morals* 2 (2011), pp. 67–78.

[300] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (Jan. 1993), pp. 861–867.

[301] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals, and Systems* 2.4 (Dec. 1989), pp. 303–314.

[302] D. Costarelli. “Interpolation by neural network operators activated by ramp functions”. In: *Journal of Mathematical Analysis and Applications* 419.1 (Nov. 2014), pp. 574–582.

[303] G. A. Anastassiou. “Rate of Convergence of Some Neural Network Operators to the Unit-Univariate Case”. In: *Journal of Mathematical Anaylsis and Applications* 212 (1997), pp. 237–262.

[304] D. Costarelli. “Neural network operators: Constructive interpolation of multivariate functions”. In: *Neural Networks* 67 (July 2015), pp. 28–36.

[305] S. Wold, K. Esbensen, and P. Geladi. “Principal component analysis”. In: *Chemometrics and Intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[306] G. E. Hinton and S. T. Roweis. “Stochastic neighbor embedding”. In: *Advances in Neural Information Processing Systems*. 2003, pp. 857–864.

[307] Y. Bengio. “Deep learning of representations for unsupervised and transfer learning”. In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. 2012, pp. 17–36.

[308] S. Benaim and L. Wolf. “One-Sided Unsupervised Domain Mapping”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. 2017, pp. 752–762.

[309] G. E. Hinton, S. Osindero, and Y.-W. Teh. “A fast learning algorithm for deep belief nets”. In: *Neural Computation* 18.7 (2006), pp. 1527–1554.

[310] D. P. Kingma and M. Welling. *Auto-encoding variational Bayes*. 2013. arXiv: `1312.6114 [stat.ML]`.

[311] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller. “A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1996–2000.

[312] Z. Ghahramani. “Unsupervised Learning”. In: *Advanced Lectures on Machine Learning LNAI 3176*. Ed. by O. Bousquet, G. Raetsch, and U. von Luxburg. 2004.

[313] S. Mohamed, K. Heller, and Z. Ghahramani. *Bayesian and L1 Approaches for Sparse Unsupervised Learning*. 29th International Conference on Machine Learning (ICML). 2012.

[314] R. R. Coifman and S. Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30.

[315] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. “Diffusion maps, spectral clustering and reaction coordinates of dynamical systems”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 113–127.

[316] F. P. Kemeth, S. W. Haugland, F. Dietrich, T. Bertalan, Q. Li, E. M. Bollt, R. Talmon, K. Krischer, and I. G. Kevrekidis. *An Equal Space for Complex Data with Unknown Internal Order: Observability, Gauge Invariance and Manifold Learning*. arXiv: `1708.05406 [physics.data-an]`.

[317] G. Hinton and R. Zemel. “Autoencoders, minimum description length and Helmoltz free energy”. In: *Advances in Neural Information Processing Systems 6*. Ed. by J. Cowan, G. Tesauro, and J. Alspector. San Mateo, CA, 1994.

[318] R. Zemel and G. Hinton. “Learning population codes by minimizing description length”. In: *Neural Computation* 7.3 (1995), pp. 549–564.

[319] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. "The Helmholtz Machine". In: *Neural Computation* 7.5 (Sept. 1995), pp. 889–904.

[320] I. Molchanov. *Theory of random sets*. 2006.

[321] L. A. Dalton, M. Benalcázar, M. Brun, and E. R. Dougherty. "Analytic Representation of Bayes Labeling and Bayes Clustering Operators for Random Labeled Point Processes." In: *IEEE Transactions on Signal Processing* 63.6 (2015), pp. 1605–1620.

[322] L. A. Dalton. "On the optimality of k-means clustering". In: *Genomic Signal Processing and Statistics (GENSiPS), 2013 IEEE International Workshop on*. 2013, pp. 70–71.

[323] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani. "Deep reinforcement learning framework for autonomous driving". In: *Electronic Imaging* 2017.19 (2017), pp. 70–76.

[324] J. Kober, J. A. Bagnell, and J. Peters. "Reinforcement learning in robotics: A survey". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[325] F. Dahl. "A reinforcement learning algorithm applied to simplified two-player Texas Hold'em poker". In: *Machine Learning: ECML 2001* (2001), pp. 85–96.

[326] J. Heinrich and D. Silver. *Deep reinforcement learning from self-play in imperfect-information games*. 2016. arXiv: `1603.01121 [cs.LG]`.

[327] C. Moyer. "How Google's AlphaGo beat a Go world champion". In: *The Atlantic* 28 (2016).

[328] D. Silver, R. S. Sutton, and M. Müller. "Reinforcement Learning of Local Shape in the Game of Gō." In: *IJCAI*. Vol. 7. 2007, pp. 1053–1058.

[329] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. Dec. 2017. arXiv: `1712.01815 [cs.AI]`.

[330] C. Amato and G. Shani. "High-level reinforcement learning in strategy games". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. 2010, pp. 75–82.

[331] G. Lample and D. S. Chaplot. "Playing FPS Games with Deep Reinforcement Learning." In: *AAAI*. 2017, pp. 2140–2146.

[332] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[333] V. S. Borkar. "Reinforcement learning – a bridge between numerical methods and Monte Carlo". In: *Perspectives in Mathematical Sciences I: Probability and Statistics*. 2009, pp. 71–91.

[334] L. P. Kaelbling, M. L. Littman, and A. W. Moore. "Reinforcement Learning: A Survey". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285.

[335] M. Duff. "Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes". PhD thesis. Department of Computer Science, University of Massachusetts Amherst, 2002.

[336] S. Ross, B. Chaib-draa, and J. Pineau. "Bayesian reinforcement learning in continuous POMDPs with application to robot navigation". In: *In International Conference on Robotics and Automation*. 2008, pp. 2845–2851.

[337] P. Dallairea, C. Besse, S. Ross, and B. Chaib-draa. “Bayesian reinforcement learning in continuous POMDPs with Gaussian processes”. In: *International Conference on Intelligent Robots and Systems*. 2009, pp. 2604–2609.

[338] M. Deisenroth and C. Rasmussen. *PILCO: A model-based and data-efficient approach to policy search*. International Conference on Machine Learning. 2011.

[339] R. T. McAllister and C. E. Rasmussen. *Data-Efficient Reinforcement Learning in Continuous State-Action Gaussian-POMDPs*. 31st Conference on Neural Information Processing Systems (Neural Information Processing Systems 2017). 2017.

[340] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. “Asynchronous methods for deep reinforcement learning”. In: *International Conference on Machine Learning*. 2016, pp. 1928–1937.

[341] T. Salimans, J. Ho, X. Chen, and I. Sutskever. *Evolution strategies as a scalable alternative to reinforcement learning*. 2017. arXiv: `1703.03864 [stat.ML]`.

[342] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel. *Continuous adaptation via meta-learning in nonstationary and competitive environments*. 2017. arXiv: `1710.03641`.

[343] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. *One-shot visual imitation learning via meta-learning*. 2017. arXiv: `1709.04905 [cs.LG]`.

[344] Z. Li, F. Zhou, F. Chen, and H. Li. *Meta-SGD: Learning to Learn Quickly for Few Shot Learning*. 2017. arXiv: `1707.09835 [cs.LG]`.

[345] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. *Meta-Learning with Temporal Convolutions*. 2017. arXiv: `1707.03141 [cs.AI]`.

[346] D. Rezende, Shakir, I. Danihelka, K. Gregor, and D. Wierstra. “One-Shot Generalization in Deep Generative Models”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA, June 2016, pp. 1521–1529.

[347] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch. *Emergent Complexity via Multi-Agent Competition*. 2017. arXiv: `1710.03748 [cs.AI]`.

[348] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. *One-Shot Imitation Learning*. 2017. arXiv: `1703.07326 [cs.AI]`.

[349] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. *Continuous control with deep reinforcement learning*. 2015. eprint: `1509.02971` (cs.LG).

[350] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine. “Continuous Deep Q-Learning with Model-based Acceleration”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA, June 2016, pp. 2829–2838.

[351] X. Zhu and A. Goldberg. “Introduction to semisupervised learning”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3.1 (2009), pp. 1–130.

[352] M. Hady and F. Schwenker. “Semi-supervised learning”. In: *Handbook on Neural Information Processing*. 2013, pp. 215–239.

[353] V. Demyanov, A. Pozdnoukhov, M. Kanevski, and M. Christie. *Geomodelling of a Fluvial System with Semi-Supervised Support Vector Regression*. GEOSTATS 2008—VIII International Geostatistics Congress. 2008.

[354] A. Blum and S. Chawla. *Learning from labeled and unlabeled data using graph mincuts*. 2001.

[355] A. Bertozzi, X. Luo, A. Stuart, and K. Zygalakis. *Uncertainty quantification in the classification of high dimensional data*. 2017. arXiv: `1703.08816v1`.

[356] V. Vapnik. *Statistical Learning Theory*. 1998.

[357] T. Joachims. *Transductive inference for text classification using support vector machines*. International Conference on Machine Learning. 1999.

[358] R. Collobert, F. Sinz, J. Weston, and L. Bottou. "Large scale transductive SVMs". In: *Journal of Machine Learning Research* 7 (2006), pp. 1687–1712.

[359] T. Joachims. "Transductive Learning via Spectral Graph Partitioning". In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*. 2003.

[360] M. Loog. "Contrastive pessimistic likelihood estimation for semi-supervised classification". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.3 (2016), pp. 462–475.

[361] M. Trapp, T. Madl, R. Peharz, F. Pernkopf, and R. Trappl. *Safe Semi-Supervised Learning of Sum-Product Networks*. 2017. arXiv: `1710.03444v1`.

[362] B. Settles. *Active learning literature survey*. University of Wisconsin, Madison. 2010.

[363] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Marí. "A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification". In: *IEEE Journal of Selected Topics in Signal Processing* 5.3 (June 2011).

[364] N. Houlsby, J. M. Hernández-Lobato, and Z. Ghahramani. "Cold-start Active Learning with Robust Ordinal Matrix Factorization". In: *Proceedings of the 31st International Conference on Machine Learning*. Beijing, China, 2014.

[365] J. T. Connor, R. D. Martin, and L. E. Atlas. "Recurrent neural networks and robust time series prediction". In: *IEEE Transactions on Neural Networks* 5 (1994), pp. 240–252.

[366] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis*. 4th. 2008.

[367] U. Helmke, J. B. Moore, and J. E. Perkins. "Dynamical systems that compute balanced realizations and the singular value decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 15 (1994), pp. 733–754.

[368] V. Algazi and D. Sakrison. "On the optimality of the Karhunen-Loeve expansion". In: *IEEE Transactions on Information Theory* 15 (1969), pp. 319–321.

[369] J. B. Tenenbaum, V. de Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290 (2000), pp. 2319–2323.

[370] S. T. Roweis and L. K. Saul. "Nonlinear dimensionality reduction by locally linear embedding". In: *Science* 290 (2000), pp. 2323–2326.

[371] L. G. Valiant. "A theory of the learnable". In: *Communications of the ACM* 27 (1984), pp. 1134–1142.

[372] V. Vapnik. *Principles of risk minimization for learning theory*. Neural Information Processing Systems. 1992.

[373] H. Dunning. *Fields medallist mathematician joins Imperial*. `http://www3.imperial.ac.uk/newsandeventspggrp/imperialcollege/newssummary/news_2-10-2017-15-9-15`. Oct. 2017.

[374] U. Shaham, A. Cloninger, and R. R. Coifman. "Provable approximation properties for deep neural networks". In: *Applied and Computational Harmonic Analysis* (2016).

[375] A. R. Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information Theory* 39.3 (1993), pp. 930–945.

[376] H. T. Siegelmann and E. D. Sontag. "On the Computational Power of Neural Nets". In: *Journal of Computer and System Sciences* 50.1 (Feb. 1995), pp. 132–150.

[377] F. Girosi and T. Poggio. "Networks and the best approximation property". In: *Biological Cybernetics* 63 (1990), pp. 169–176.

[378] H. Mhaskar. "Neural networks for optimal approximation of smooth and analytic functions". In: *Neural Computation* 8 (1996), pp. 164–177.

[379] K. Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257.

[380] A. Rahimi. *Neural Information Processing Systems 2017 "Test of time" award presentation*. `https://www.youtube.com/watch?v=Qi1Yry33TQE&feature=youtu.be`.

[381] I. Babuska and J. T. Oden. "Verification and validation in computational engineering and science: basic concepts". In: *Computer Methods in Applied Mechanics and Engineering* 193.36-38 (2004), pp. 4057–4066.

[382] P. J. Roache. *Fundamentals of Verification and Validation*. Socorro, NM, 2009.

[383] C. H. W.L. Oberkampf T.G. Trucano. *Verification, validation, and predictive capabilities in computational engineering, and physics*. Tech. rep. 2003–3769. Sandia National Labs, 2003.

[384] M. Collins, F. J. Vecchio, R. G. Selby, and P. R. Gupta. "The failure of an offshore platform". In: *Concrete International - Detroit* 19 (1997), pp. 28–36.

[385] B. Jakobsen and F. Rosendahl. "The Sleipner platform accident". In: *Structural Engineering International* 4.3 (1994), pp. 190–193.

[386] P. Benner, S. Gugercin, and K. Willcox. "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems". In: *SIAM Review* 57.4 (2015), pp. 483–531.

[387] A. Mayo and A. Antoulas. "A framework for the solution of the generalized realization problem". In: *Linear Algebra and its Applications* 425.2–3 (2007), pp. 634–662.

[388] A. Ionita and A. Antoulas. "Data-Driven Parametrized Model Reduction in the Loewner Framework". In: *SIAM Journal on Scientific Computing* 36.3 (2014), A984–A1007.

[389] B. Gustavsen and A. Semlyen. "Rational approximation of frequency domain responses by vector fitting". In: *Power Delivery, IEEE Transactions on* 14.3 (July 1999), pp. 1052–1061.

[390] Z. Drmac, S. Gugercin, and C. Beattie. "Quadrature-Based Vector Fitting for Discretized $\mathcal{H}_2$ Approximation". In: *SIAM Journal on Scientific Computing* 37.2 (2015), A625–A652.

[391] P. Schmid and S. J. "Dynamic mode decomposition of numerical and experimental data". In: *Bulletin American Physical Society, 61st APS meeting*. 2008, p. 208.

[392] P. Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of Fluid Mechanics* 656 (Aug. 2010), pp. 5–28.

[393] C. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. Henningson. "Spectral analysis of nonlinear flows". In: *Journal of Fluid Mechanics* 641 (Dec. 2009), pp. 115–127.

[394] J.-N. Juang and R. S. Pappa. "An eigensystem realization algorithm for modal parameter identification and model reduction". In: *Journal of Guidance* 8.5 (1985), pp. 620–627.

[395] B. Peherstorfer and K. Willcox. “Data-driven operator inference for nonintrusive projection-based model reduction”. In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215.

[396] S. Lall, P. Krysl, and J. E. Marsden. “Structure-preserving model reduction for mechanical systems”. In: *Physica D: Nonlinear Phenomena* 184.1 (2003), pp. 304–318.

[397] C. Beattie and S. Gugercin. “Interpolatory projection methods for structure-preserving model reduction”. In: *Systems & Control Letters* 58.3 (2009), pp. 225–232.

[398] L. Peng and K. Mohseni. “An online manifold learning approach for model reduction of dynamical systems”. In: *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1928–1952.

[399] L. Peng and K. Mohseni. “Symplectic model reduction of Hamiltonian systems”. In: *SIAM Journal on Scientific Computing* 38.1 (2016), A1–A27.

[400] K. Veroy, C. Prud’homme, D. V. Rovas, and A. T. Patera. “A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations”. In: *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2003-3847. Orlando, FL, 2003.

[401] C. Prud’homme, D. Rovas, K. Veroy, Y. Maday, A. Patera, and G. Turinici. “Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods”. In: *Journal of Fluids Engineering* 124 (2002), pp. 70–80.

[402] M. Grepl and A. Patera. “*A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations”. In: *ESAIM-Mathematical Modelling and Numerical Analysis (M2AN)* 39.1 (2005), pp. 157–181.

[403] G. Rozza, D. Huynh, and A. Patera. “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics”. In: *Archives of Computational Methods in Engineering* 15.3 (2008), pp. 229–275.

[404] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. “Convergence rates for greedy algorithms in reduced basis methods”. In: *SIAM Journal on Mathematical Analysis* 43.3 (2011), pp. 1457–1472.

[405] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz. “Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems”. In: *SIAM Journal on Applied Dynamical Systems* 13.4 (2014), pp. 1716–1732.

[406] P. Simon. *Too big to ignore: The business case for big data*. Vol. 72. 2013.

[407] F. Sebastiani. “Machine Learning in Automated Text Categorization”. In: *ACM Computing Surveys* 34.1 (2002), pp. 1–47.

[408] M. Bkassiny, Y. Li, and S. K. Jayaweera. “A Survey on Machine-Learning Techniques in Cognitive Radios”. In: *IEEE Communications Surveys Tutorials* 15.3 (2013), pp. 1136–1159.

[409] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[410] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*. 2016. arXiv: `1603.04467 [cs.DC]`.

[411] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. “Distributed GraphLab: a framework for machine learning and data mining in the cloud”. In: *Proceedings of the VLDB Endowment* 5.8 (2012), pp. 716–727.

[412] E. P. Xing, Q. Ho, W. Dai, J. K. Kim, J. Wei, S. Lee, X. Zheng, P. Xie, A. Kumar, and Y. Yu. “Petuum: A new platform for distributed machine learning on big data”. In: *IEEE Transactions on Big Data* 1.2 (2015), pp. 49–67.

[413] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. “Large scale distributed deep networks”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 1223–1231.

[414] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. “Caffe: Convolutional architecture for fast feature embedding”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 675–678.

[415] S. Jacobs, N. Dryden, R. Pearce, and B. Van Essen. *Towards Scalable Parallel Training of Deep Neural Networks*. Tech. rep. Lawrence Livermore National Laboratory, 2017.

[416] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. “Theano: A CPU and GPU math compiler in Python”. In: *Proceedings of the 9th Python in Science Conf*. 2010, pp. 1–7.

[417] N. Lopes and B. Ribeiro. “GPUMLib: An efficient open-source GPU machine learning library”. In: *International Journal of Computer Information Systems and Industrial Management Applications* 3 (2011), pp. 355–362.

[418] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, et al. “Dadiannao: A machine-learning supercomputer”. In: *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 2014, pp. 609–622.

[419] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. *Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems*. 2015. arXiv: `1512.01274 [cs.DC]`.

[420] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. “Scaling Distributed Machine Learning with the Parameter Server.” In: *OSDI*. 10.4. 2014, p. 3.

[421] S. R. Young, D. C. Rose, T. Johnston, W. T. Heller, T. P. Karnowski, T. E. Potok, R. M. Patton, G. Perdue, and J. Miller. “Evolving Deep Networks Using HPC”. In: *Proceedings of the Machine Learning on HPC Environments*. Denver, CO, USA, 2017.

[422] B. D. Rouhani, A. Mirhoseini, and F. Koushanfar. “Deep3: Leveraging Three Levels of Parallelism for Efficient Deep Learning”. In: *Proceedings of the 54th Annual Design Automation Conference 2017*. DAC ’17. Austin, TX, USA, 2017.

[423] J. C. Bennett, A. Bhagatwala, J. H. Chen, A. Pinar, M. Salloum, and C. Seshadhri. “Trigger Detection for Adaptive Scientific Workflows Using Percentile Sampling”. In: *SIAM Journal on Scientific Computing* 38.5 (2016), S240–S263.

[424] P. Ma and X. Sun. “Leveraging for big data regression”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 7.1 (2015), pp. 70–76.

[425] H. Wang, R. Zhu, and P. Ma. “Optimal Subsampling for Large Sample Logistic Regression”. In: *Journal of the American Statistical Association* (2017).

[426] R. Zhu. “Gradient-based Sampling: An Adaptive Importance Sampling for Least-squares”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 406–414.

[427] P. Koiran and E. D. Sontag. “Vapnik-Chervonenkis dimension of recurrent neural networks”. In: *Discrete Applied Mathematics* 86 (1998), pp. 63–79.

[428] C. Zhang, W. Bian, D. Tao, and W. Lin. “Discretized-Vapnik-Chervonenkis Dimension for Analyzing Complexity of Real Function Classes”. In: *IEEE Transactions on Neural Networks and Learning Systems* 23.9 (Sept. 2012), pp. 1461–1472.

[429] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. “Learnability and the Vapnik-Chervonenkis dimension”. In: *Journal of the ACM* 36.4 (Oct. 1989), pp. 929–965.

[430] P. Kuusela, D. Ocone, and E. D. Sontag. “Remarks on the Sample Complexity for Linear Control Systems Identification”. In: *IFAC Proceedings Volumes* 34.14 (Aug. 2001), pp. 431–436.

[431] C. Docan, M. Parashar, and S. Klasky. “Dataspaces: an interaction and coordination framework for coupled simulation workflows”. In: *Cluster Computing* 15.2 (2012), pp. 163–181.

[432] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, et al. “Combining in-situ and in-transit processing to enable extreme-scale scientific analysis”. In: *International Conference for High Performance Computing, Networking, Storage and Analysis (SC’12)*. 2012, pp. 1–9.

[433] S. Sra, S. Nowozin, and S. J. Wright, eds. *Optimization for Machine Learning*. Neural Information Processing. 2011.

[434] K. P. Bennett and E. Parrado-Hernandez. “The Interplay of Optimization and Machine Learning Research”. In: *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 1265–1281.

[435] L. Bottou, F. E. Curtis, and J. Nocedal. *Optimization Methods for Large-Scale Machine Learning*. 2016. arXiv: `1606.04838 [stat.ML]`.

[436] F. E. Curtis and K. Scheinberg. “Optimization Methods for Supervised Machine Learning: From Linear Models to Deep Learning”. In: *The Operations Research Revolution*. Sept. 2017, pp. 89–113.

[437] A. Agarwal, O. Dekel, and L. Xiao. “Optimal algorithms for online convex optimization with multi-point bandit feedback”. In: *23rd Conference on Learning Theory*. 2010, pp. 28–40.

[438] A. Agarwal, S. Negahban, and M. J. Wainwright. “Fast global convergence of gradient methods for high-dimensional statistical recovery”. In: *The Annals of Statistics* 40.5 (Oct. 2012), pp. 2452–2482.

[439] P. Richtarik and M. Takac. “Parallel coordinate descent methods for big data optimization”. In: *Mathematical Programming* 156.1 (Mar. 2016), pp. 433–484.

[440] Y. Nesterov. “Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems”. In: *SIAM Journal on Optimization* 22.2 (2012), pp. 341–362.

[441] M. C. Ferris and T. S. Munson. “Interior-Point Methods for Massive Support Vector Machines”. In: *SIAM Journal on Optimization* 13.3 (2002), pp. 783–804.

[442] A. Gunes Baydin, B. A. Pearlmutter, A. Andreyevich Radul, and J. M. Siskind. *Automatic differentiation in machine learning: A survey*. Feb. 2015. arXiv: `1502.05767`.

[443] D. Horn and B. Bischl. “Multi-objective parameter configuration of machine learning algorithms using model-based optimization”. In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2016, pp. 1–8.

[444] K. Kunisch and T. Pock. “A Bilevel Optimization Approach for Parameter Learning in Variational Models”. In: *SIAM Journal on Imaging Sciences* 6.2 (2013), pp. 938–983.

[445] G. Lebanon and J. D. Lafferty. “Boosting and maximum likelihood for exponential models”. In: *Advances in Neural Information Processing Systems*. 2002, pp. 447–454.

[446] E. Brochu, V. M. Cora, and N. de Freitas. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. 2010. arXiv: `1012.2599v1`.

[447] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. 2006.

[448] J. R. Gardner, M. J. Kusner, Z. Xu, K. W. Weinberger, and J. P. Cunningham. “Bayesian Optimization with Inequality Constraints”. In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. 2014.

[449] J. M. Hernandez–Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani. “Predictive Entropy Search for Bayesian Optimization with Unknown Constraints”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. 2015.

[450] V. Picheny, R. B. Gramacy, S. M. Wild, and S. Le Digabel. “Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian”. In: *Thirtieth Annual Conference on Neural Information Processing Systems (Neural Information Processing Systems)*. Dec. 2016.

[451] M. A. Taddy, R. B. Gramacy, and N. G. Polson. “Dynamic Trees for Learning and Design”. In: *J. Amer. Statist. Assoc.* 106.493 (2011), pp. 109–123.

[452] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas. “Learning to learn by gradient descent by gradient descent”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3981–3989.

[453] O. Wichrowska, N. Maheswaranathan, M. W. Hoffman, S. G. Colmenarejo, M. Denil, N. de Freitas, and J. Sohl-Dickstein. *Learned Optimizers that Scale and Generalize*. 2017. arXiv: `1703.04813 [cs.LG]`.

[454] S. Ravi and H. Larochelle. *Optimization as a model for few-shot learning*. 2016.

[455] C. Finn, P. Abbeel, and S. Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. 2017. arXiv: `1703.03400 [cs.LG]`.

[456] J. Snell, K. Swersky, and R. S. Zemel. *Prototypical Networks for Few-shot Learning*. 2017. arXiv: `1703.05175 [cs.LG]`.

[457] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, M. Botvinick, and N. Freitas. “Learning to learn without gradient descent by gradient descent”. In: *International Conference on Machine Learning*. 2017, pp. 748–756.

[458] K. Sargsyan, C. Safta, H. N. Najm, B. J. Debusschere, D. Ricciuto, and P. Thornton. “Dimensionality Reduction for Complex Models Via Bayesian Compressive Sensing”. In: *International Journal for Uncertainty Quantification* 4.1 (2014), pp. 63–93.

[459] E. J. Candes. “Compressive sampling”. In: *Proceedings of the International Congress of Mathematicians*. 2006.

[460] T. Krueger, D. Panknin, and M. Braun. *Fast Cross-Validation via Sequential Testing*. 2016. arXiv: `1206.2248v6`.