AI Benchmarking for Science: Efforts from the MLCommons Science Working Group

Jeyan Thiyagalingam[1♠], Gregor von Laszewski[2], Junqi Yin[3], Murali Emani[4], Juri Papay[1], Gregg Barrett[5], Piotr Luszczek[6], Aristeidis Tsaris[3], Christine Kirkpatrick[7], Feiyi Wang[3], Tom Gibbs[8], Venkatram Vishwanath[4], Mallikarjun Shankar[3], Geoffrey Fox[2♣], Tony Hey[1]

[1] Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK
[2] University of Virginia, Charlottesville, VA 22904-4298, USA
[3] Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[4] Argonne National Laboratory, Lemont, IL 60439, USA
[5] Cirrus AI, Johannesburg, RSA
[6] University of Tennessee, Knoxville, TN 37996, USA
[7] SDSC, 10100 Hopkins Dr, La Jolla, CA 92093
[8] NVIDIA
Corresponding Authors: ♠t.jeyan@stfc.ac.uk, ♣vxj6mb@virginia.edu

Abstract. With machine learning (ML) becoming a transformative tool for science, the scientific community needs a clear catalogue of ML techniques, and their relative benefits on various scientific problems, if they were to make significant advances in science using AI. Although this comes under the purview of benchmarking, conventional benchmarking initiatives are focused on performance, and as such, science, often becomes a secondary criteria.

In this paper, we describe a community effort from a working group, namely, MLCommons Science Working Group, in developing science-specific AI benchmarking for the international scientific community. Since the inception of the working group in 2020, the group has worked very collaboratively with a number of national laboratories, academic institutions and industries, across the world, and has developed four science-specific AI benchmarks. We will describe the overall process, the resulting benchmarks along with some initial results. We foresee that this initiative is likely to be very transformative for the AI for Science, and for performance-focused communities.

Keywords: Machine Learning · Benchmarks · Science and AI for Science.

1 Introduction

Recently, owing to the advances in deep learning, the AI, has been transformational in various aspects of our life. These advances have resulted in machine learning being one of the effective techniques for scientific data analysis, covering a number of domains of sciences, such as material, life, and environmental

sciences, particle physics and astronomy [9,10,24,11,1,13,22]. With AI becoming one of the underpinning technologies for science, there is a considerable amount of attention on several aspects of AI, including, but not limited to, understanding the general applicability of AI/ML to various scientific problems, role of high performance computing on AI/ML, datasets, explainability and robustness of AI/ML techniques, role of small-scale devices on AI/ML, AI/ML-specific algorithms, and scalability of AI/ML techniques with varying volumes of data or varying computational capabilities. With each of these areas being considerably large, it is a substantial undertaking for any single organization or community for developing an overall understanding of various initiatives and their corresponding impacts, particularly across different domains of applications. Ideally, multiple communities should join forces to understand these issues and to make relevant progresses in AI.

MLCommons is one such global initiative with the mission being *accelerate machine learning innovation and increase its positive impact on society.* Although MLCommons™ initiatives were legally setup in 2020, the initiatives originated along with the MLPerf™ benchmarking efforts in 2018. The overarching strands are: benchmarks, datasets, and best practice systems and usage. The current MLCommons initiatives retain the core activities of MLPerf across six distinct focus areas: Training, Training HPC, Inference Datacenter, Inference Edge, Inference Mobile, and Inference Tiny. With application and impact of AI being rather broad, MLCommons is setup along with a number of research working groups with the vision of creating an open "*AI for Research*" ecosystem that is driven by the community for the community[9]. These groups are open to the public, including academics and researchers. The philosophy of MLCommons is to support open-source "AI for Research". The MLCommons Research organization is responsible for overseeing new activities that can lead to new scientific methods in ML, as well as new applications of ML, and currently houses a number of working groups that focus on various areas of ML. These include: ML algorithms (Algorithms), dataset benchmarking (DataPerf), building shared resource infrastructure (Dynabench), benchmarking and best practices for healthcare (Medical), storage benchmarking for ML (Storage),and AI benchmarking for science (Science) [6].

In this paper, we describe the benchmarking initiatives of the Science Working Group, covering our initial set of benchmarks, datasets, policies that govern our benchmarks and benchmarking, rules around submitting new benchmarks or datasets, and some initial results on the evaluation of these benchmarks.

The rest of this paper is organized as follows: In Section 2, we describe the working group, goals of the group, and policies adopted by the working group towards science benchmarking. This is then followed by Section 3, where we describe the initial set of benchmarks curated by the working group. In Section 4, we provide some initial evaluations and discuss the results, and we conclude the paper with future directions in Section 5.

[9] https://mlcommons.org/en/groups/research/

2 MLCommons Science Working Group

2.1 About the Working Group

The Science working group [6] was an early member of MLCommons Research, created by the international community working on AI for Science, such as various national laboratories, large-scale experimental facilities, universities and commercial entities, to advance AI for Science along with other national and international level initiatives (e.g., [2]). The overarching drive of the WG is to support various scientific communities that are trying to leverage AI for advancing scientific discoveries. Since the inception, the WG has expanded to include almost 120 members, located across various organizations. The group also works with a number of other working groups within MLCommons, such HPC WG [5], where there are a number of overlapping issues of interest. The overall mission of the group entails collaborative engagements across different domains of sciences.

2.2 Science Benchmarking

Achieving the overall goals of the working group requires a number of sub-aspects to be covered by the WG, such as, (a)identifying a number of representative scientific problems where AI can make a difference, (b) engineering at least one ML solution to the problem, to be considered as a baseline implementation, (c) identifying relevant datasets upon which the ML models can be trained or tested, (d) identifying a scientifically-driven metric that can help recognizing the scientific advancement to the problem, (e) curating and publishing those relevant datasets, (f) publishing the scientific results that can help the communities to develop improve these solutions, and (g) fostering collaborations and scientific achievements across multidisciplinary communities. All these activities are akin to conventional benchmarking, but with a major difference of focusing on scientific merits than pure performance, and hence the notion of science benchmarking. Since the formation, the WG has consulted a large number of scientific organizations, and worked with scientists in achieving some of the sub-aspects listed above. In particular, the WG has succeeded in identifying four science benchmarks derived from different branches of sciences, namely, (a) Cloud masking (`cloud-masking`) [23] — atmospheric sciences, (b) Space group classification of solid state materials from Scanning Transmission Electron Microscope (STEM) data using Deep Learning (`stemdl`) [14] — solid state physics, (c) Time evolution operator (`tevelop`) [7] exemplified using predicting earthquakes — earth sciences and (d) predicting tumor response to drugs (`candle-uno`) — healthcare.

We discuss these benchmarks in detail in Section 3. The key aspect here is that a single benchmark is actually a combination of a baseline or reference implementation and one or more datasets. The scientific data here requires a special attention. Although scientific datasets are widespread and common, curating, maintaining, and distributing large-scale, scientific datasets for public consumption is a challenging process, covering various aspects, from abiding by the FAIR

principles [26] to distribution to versioning of the datasets. These benchmarks have a multitude of purpose, which are discussed at length in [24,11]. However, it is worth highlighting that these scientific benchmarks serve one important purpose to the wider AI community: offering an unprecedented pedagogical value across domain boundaries.

2.3 Policies for Benchmarking

Benchmarking is an art and can be very subjective. Without clear policies, the benchmarking results can be subjectively and differently interpreted, leading to the whole initiative not serving the intended purpose. As such, establishing a set of policies, rules and guidelines for evaluating and reporting results for the benchmarks is an important step. The Science WG is in the process of drafting a detailed policy statement, and, here, we mention some of the key points for the reasons of brevity. The overarching policy will cover training and inference benchmarks, with a number of sub-policies focusing on each and every benchmark, as no two benchmarks are the same. In general, the policies will cover the evaluation of benchmarks under two divisions, namely, Open and Closed divisions. Benchmark evaluation under the former will focus on achieving better scientific results (using the established scientific metric). As such, the community has considerable amount of freedom to enhance the underlying ML models or pre- or post-processing aspects of the benchmarks, including data augmentation, wherever that is possible or sensible. Evaluation under the Closed division, on the other hand, limits the freedom and often will list permissible changes for each and every benchmark. In general, pre- and/or post-processing, and data aspects are often kept fixed, with flexibility to change or fine-tune the underling ML model. Similarly, policies around submission of results may also vary across benchmarks. For example, some benchmarks may insist on certain set of measurements to be submitted, such as power or network performance, while some may rely on generic details along with scientific metrics.

3 Benchmarks for the First Release

As outlined in Section 2, the WG has consolidated four different benchmarks from four different branches of sciences, namely, `cloud-mask`, `stemdl`, `candle-uno` and `tevelop`. We describe each of these benchmarks in detail, covering the science case, objectives, metrics, data and outline the baseline reference implementation. The aim here is to ensure that the community is aware of these challenges, and can develop techniques outperforming the baseline cases.

3.1 Cloud Masking (`cloud-mask`)

Sea and land surface temperatures (SST and LST), have a significant influence on the Earth's weather, and as such, estimation of SST from space-borne sensors, such as satellites, is crucial for a number of applications in environmental

sciences. Satellites are often equipped with special sensors for this purpose, such as the Sea and Land Surface Temperature Radiometer (SLSTR) on board the Sentinel-3 satellite. In principle, it is possible to make direct measurements of surface temperature from these satellites everywhere, except when clouds are present. Clouds can really affect the signals measured by satellites. making it much harder to retrieve the temperature measurements. One of the aspects that underpins the derivation of SST is cloud screening, which is a step that marks each pixel of thousands of satellite images as containing cloud or clear sky. This has been, historically, performed using either thresholding or Bayesian method. The purpose of this benchmark is to perform this using machine learning. An example input and output images are given in Figure 1. We also summarize the key features of this benchmark in Table 1. Details around objective of the benchmark, description of relevant datasets, and reference implementation are given below.

Table 1: Summary of the `cloud-mask` Benchmark.

Description	Image classification at pixel level of satellite imagery.
Objective	Classification of pixels of satellite images into cloud and clear sky categories using machine learning.
Challenge Stream	Image Segmentation
Domain	Atmospheric Sciences
Metrics	Classification accuracy
Data	Type: Images ([$2400\times3000\times6$] and [$1200\times1500\times3$]) Size: 180 GB Source: CEDA Location: STFC Servers [23]
Reference implementation	SciML-Bench Cloudmask Benchmark [12]

Benchmarking Objectives and Metrics: The scientific objective of the problem is to develop a segmentation model for classifying the pixels in satellite images. This classification allows determining whether the given pixel belongs to a cloud or to a clear sky. The Bayesian techniques [17] used conventionally can lead to sub-optimal outputs in a number of cases, and hence the scope of the `cloud-mask` benchmark is to explore whether ML-driven algorithms can outperform the Bayesian techniques. Although various options are there, in its present form, the `cloud-mask` benchmark is set as a supervised learning problem, with cloud images are treated as inputs. However, like all science cases, the "true" ground truth (or labels), are never available for this case. Hence, the benchmark uses the Bayesian masks, supplied by the provider of the satellite images, as the ground truth. While this is arguable, we believe that in the absence of any ground truth, this is a valid and perfect choice. However, with Bayesian masks not always being accurate or not offering a gold-standard for masks, the resulting model is likely to suffer from learnability issues, which sets the perfect challenge

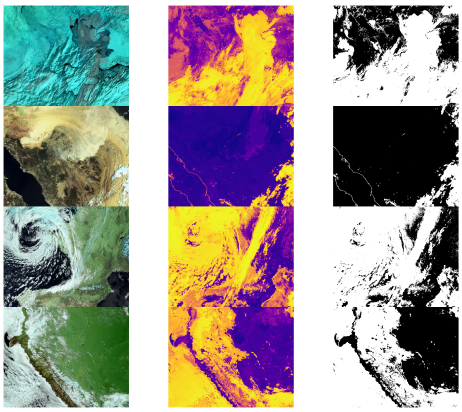


Fig. 1: Cloud mask example. The left column shows the raw images from the Sentinel-3 satellite while the images on the right column shows the predicted probability that a particular pixel is cloud.

for an ML-driven case. The benchmark can be considered as both training and inference focused, where the science metric is same as the classification accuracy — number of pixels classified correctly. The performance metric, can be inference timing and scalability on the training across a number of GPUs.

Data: The masking can be performed across different satellite imaging modalities. This particular benchmark relies on satellite imagery obtained from the SLSTR sensors equipped as part of the Sentinel-3 satellite. More specifically, the benchmark operates on multi-spectral image data. The overall dataset identified for this benchmark is split into two distinct sets: training set (163 GB) and an inference set (1.7GB). Each dataset inside these sets is made up of two parts: reflectance and brightness temperature. The reflectance is captured across six channels with the resolution of 2400×3000 pixels, and the brightness temperature is captured across three channels with the resolution of 1200×1500 pixels. Although the raw satellite images are free to download from the CEDA archive[10], the curated datasets are available as part of this benchmark, located in object store within the STFC servers. The exact instructions for securing these datasets are outlined in the WG pages.

Reference Implementation: The current reference implementation is variation of the U-Net deep neural network [20], implemented using TensorFlow and Keras, with the support for distributed training using TensorFlow's native library, Distributed Mirrored Strategy. The model represents a U-Net network and consists of 39 layers with two million trainable parameters. Further details can be found in [23].

[10] https://www.ceda.ac.uk/

3.2 STEMDL (`stemdl`)

State of the art Scanning Transmission Electron Microscopes (STEM) produce focused electron beams with atomic dimensions, and allow capturing diffraction patterns arising from the interaction of incident electrons with nano-scale material volumes. Backing out the local atomic structure of said materials requires compute- and time-intensive analyses of these diffraction patterns (known as convergent beam electron diffraction or CBED). Traditional analyses of CBED requires iterative numerical solutions of partial differential equations and comparison with experimental data to refine the starting material configuration. This process is repeated anew for every newly acquired experimental CBED pattern and/or probed material.

Table 2: Summary of the `stemdl` benchmark.

Description	Classification and reconstruction of convergent beam electron diffraction, CBED.
Objectives	Classification for crystal space groups and reconstruction for local electron density using machine learning.
Challenge Stream	Classification
Domain	Solid-state Physics
Metrics	Classification accuracy and/or F1-score
Data	Type: Images $[512 \times 512 \times 3]$, label: [200] (Classification) $[256 \times 256 \times 256]$, label: $[256 \times 256]$ (Reconstruction) Size: 548.7 GB for Classification Training samples: 138.7K Validation samples: 48.4 Reconstruction: 10 TB Source: Oak Ridge National Laboratory (ORNL) Location: OSTI Servers [14]
Reference Implementation	AAIMS repository [21] Model: ResNet-50 Run Instructions: [21] Time-to-solution: 40 minutes on 60 V100 GPUs
References	[14,19,15]

Benchmark Objectives and Metrics: The scientific objective of the benchmark is to develop a universal classifier for space group of solid state materials, and reconstruction of local electron density. As stated before, this is conventionally performed using expensive simulations. The goal here is to use explore the suitability of ML algorithms for performing advanced analysis of CBED. This benchmark aims to quantify this using a classification task. As such, the benchmark is set with the supervised learning focus where both the scientific metric is reflected by the classification accuracy of the ML model. The benchmark also

desires to achieve better top-1 classification accuracy and/or F1-score compared to the reference implementation.

Data: A single data sample [14] from this dataset is given by a three-dimensional array formed by stacking various CBED patterns simulated from the same material at different distinct material projections (i.e. crystallographic orientations). Each CBED pattern is a two-dimensional array with 32-bit floating-point image intensities. Associated with each data sample in the dataset is a host of material attributes or properties which are, in principle, retrievable via analysis of this CBED stack. The dataset has (1) 200 crystal space groups out of 230 unique mathematical discrete space groups and (2) local electron density which governs material's property. A more detailed description of the data can be found in CBED database [14]. The dataset is divided into three distinct sets, split across training (148,006 files), testing (18,749 files), and development (20,400 files). The distinct nature of these sets ensures that the model learns the generic symmetry based on space groups instead of memorizing a particular pattern for a material.

Reference Implementation: A detailed description of the baseline implementation method can be found in [19] and [15] along with the reference implementation deposited into the AAIMS repository [21].

3.3 CANDLE-UNO (`candle-uno`)

The CANDLE (Exascale Deep Learning and Simulation Enabled Precision Medicine for Cancer) project[11] aims to implement deep learning architectures that are relevant to problems in cancer research, addressing problems at three biological scales: cellular (Pilot1 or P1), molecular (Pilot2 or P2), and population (Pilot3 or P3), resulting three mainstreams of benchmarks covering these pilots. The UNO version of the CANDLE suite is a P1 benchmark, which is formed out of problems and data at the cellular level. The high level goal of the problems behind the P1 benchmarks is to predict drug response based on molecular features of tumor cells and drug descriptors. We summarize the key aspects of this benchmark in Table 3, and a detailed description of the objectives, metrics, data and the reference implementation below.

Benchmarking Objectives and Metrics: The goal is to predict tumor response to single and paired drugs, based on molecular features of tumor cells across multiple data sources. It aims to accelerate the scientific goal of establishing the effectiveness of drugs. The ML component aims to accelerate this aspect using ML-based prediction of response values. As such, it is a regression problem, with the science metric being mean absolute error (MAE) between the predicted and ground truth values. On the performance front, the metric is responses predicted per second for a given batch size.

Data: Combined dose response data relies on a number of sources that are specific drug responses to cancer conditions. These include The Cancer Therapeutics Response Portal (CTRP), The Genomics of Drug Sensitivity in Cancer (GDSC),

[11] https://github.com/ECP-CANDLE/Benchmarks

Table 3: Summary of the `candle-uno` benchmark.

Description	The Pilot 1 Unified Drug Response Predictor benchmark, Uno to enable drug discovery, drug response prediction from cell lines.
Objectives	Predictions of tumor response to drug treatments, based on molecular features of tumor cells and drug descriptors
Challenge Stream	Regression
Domain	Healthcare
Metrics	Validation loss with a minimum score of 0.0054
Data	Type: Size: 6.4GB Training samples: 423,952 Validation samples: 52,994 Location : ALCF Servers [25]
Reference implementation	Model: Multi-task Learning-based custom model Code & Instructions: [4] (see README) Ideal performance: 10,667 samples/sec on a single A100 GPU for a batch size of 64

The NCI Sarcoma (SCL), The NCI Small Cell Lung Cancer (SCLC), The NCI-60 Human Cancer Cell Line Screen single drug response (NCI60), A Large Matrix of Anti-Neoplastic Agent Combinations drug pair response (ALMANAC.FG, ALMANAC.FF, ALMANAC.1A), The Genentech Cell Line Screening Initiative (gCSI) and The Cancer Cell Line Encyclopedia (CCLE). The ML model can be trained on any subset of a dataset obtained from these dose response data sources. The benchmark relies on a dataset that includes both single drug dose response measurements pair dose response measurements. More specifically, there are $27,769,716$ single drug dose response measurements and $3,686,475$ drug pair dose response measurements. The combined raw dose response data has $3,070$ unique samples and $53,520$ unique drugs. For the scope of this work, we used the AUC configuration of Uno that utilizes a single data source, namely, CCLE. We show the data distribution between the samples in Table 4. The training can be accelerated by using a pre-staged dataset file. This static dataset can, however, be pre-built. The datasets are publicly available from the CANDLE site [25]. These are directly downloadable with relevant download scripts, including a pre-built static dataset to simplify the deployment.

Reference implementation: The reference implementation implements a deep learning architecture with 21 M parameters in TensorFlow framework in Python. The code is publicly available on GitHub. It can be run in both training and inference modes. However, this benchmark is defined to be training focused. A dedicated script in the repository downloads all required datasets. The primary metric to evaluate for this application is the model throughput (samples per

Table 4: The data distribution between the single and pair drug samples.

	Growth	Sample	Drug1	Drug2	MedianDose
ALMANAC.1A	208,605	60	102	102	7.000000
ALMANAC.FF	2,062,098	60	92	71	6.698970
ALMANAC.FG	1,415,772	60	100	29	6.522879
CCLE	93,251	504	24	0	6.602060
CTRP	6,171,005	887	544	0	6.585027
GDSC	1,894,212	1,075	249	0	6.505150
NCI60	18,862,308	59	52,671	0	6.000000
SCL	301,336	65	445	0	6.908485
SCLC	389,510	70	526	0	6.908485
gCSI	58,094	409	16	0	7.430334

second). The model is said to converge when the validation loss reaches a certain threshold, for example 0.0054. The throughput is then measured for the last epoch when the model converges. With the required packages in the software stack, Uno can be run on diverse systems. More details on running Uno can be found in the repository.

3.4 Time Series Evolution Operator (`tevelop`)

Time series capture the variation of values against time, and common to a a number of scientific problems. Time series can be multiple dimensions. For example geospatial datasets are two-dimensional series, based both on time and spatial position. One of the common tasks when dealing with time series is the ability to predict or forecast them in advance. Such a task is considerably easier if the underlying time series has a clear evolution structure across dimensions. For example, if the evolution structure can be established on the spatial aspects (i.e. there is a strong correlation between nearby spatial points), estimating the evolution becomes relatively easier. The problem chosen is termed as a spatial bag where there is spatial variation, but it is not clearly linked to the geometric distance between spatial regions. In contrast, traffic-related time series have a strong spatial structure. As such, identifying the evolution in time series is a common problem across a number of domains. This particular benchmark focuses on extracting the evolution, using earthquake as the driving example. We summarize the key features of the benchmark in Table 5.

Benchmarking Objectives and Metrics: The scientific objective is to extract the evolution of a time series, exemplified using earthquake forecasting. To make the benchmarking exercise more focused, this forecasting is done on a subset of the overall earthquake dataset for the region of Southern California. Conventional methods for forecasting relies on statistical techniques. Here, the aim is to use ML for not only extracting the evolution, but also to test the effectiveness using forecasting. The exact scientific metric for quantifying the benefit of the forecasting is the Nash Sutcliffe Efficiency (NSE) [18]. It is also possible

Table 5: Summary of the `tevelop` Benchmark

Description	Earthquake Forecasting [7,16,8,3].
Objectives	Improve the quality of Earthquake forecasting in a region of Southern California.
Metrics	Normalized Nash-Sutcliffe model efficiency coefficient (NNSE)with $0.8 \leq NNSE \leq 0.99$
Data	Type: Richter Measurements with spatial and temporal information (Events). Input: Earthquakes since 1950. Size: 11.3GB (Uncompressed), 21.3MB (Compressed) Training samples: 2,400 spatial bins Validation samples: 100 spatial bins Source: USGS Servers [3]
Reference Implementation	[8]

to qualitatively asses prediction by comparing the observed earthquake, if one desires, but the benchmarks relies on the former [7].

Data: The benchmark relies on a very small subset of the earthquake data from United States Geological Survey (USGS) focused between the regions of Southern California (latitude: 32°N to 36°N, longitude: −120°S to −114°S). The subset of the data for this region covers all earthquakes in that region since 1950. There are four measurements per record, namely, magnitude, spatial location, depth from the crust, and time. The curated dataset is organized to cover this in different temporal and spatial bins. Although the actual time lapse between measurements is one day, we accumulate this into a fortnightly data. The region is then divided into a grid of 40×60 with each each pixel covering actual zone of $0.1\,\text{deg} \times 0.1$ or $11km \times 11km$ grid. The dataset also includes an assignment of pixels to known faults, and a list of the largest earthquakes in that region from 1950. We have chosen various samplings of the dataset to provide both input and predicted values. These include time ranges from a fortnight up to four years. Furthermore, we calculate summed magnitudes and depths and counts of significant quakes (magnitude < 3.29).

Reference Implementation: The benchmark includes three distinct deep learning-based reference implementations. These are Long short-term memory (LSTM)-based model, Google Temporal Fusion Transformer (TFT) [16]-based model, and a custom hybrid transformer model. The TFT-based model uses two distinct LSTMs, covering a an encoder a decoder with a temporal attention-based transformer. The custom model includes a space-time transformer for the Decoder and a two-layer LSTM for the encoder. Each model predicts NSE and generates visualizations illustrating the TFT for interpretable multi-horizon time series forecasting [16]. Details of the current reference models can be found in [7].

4 Results from Initial Evaluations

In this section, we present some of the early results obtained initial evaluations of our benchmarks. As this is the first time we are presenting these findings, it is worth noting that the initial evaluations are far from being complete or perfect, especially when lacking any relative measures to benchmark against. However, these initial evaluations are likely to provide more insight into how these evaluations should be tuned or scoped in future releases. We outline these aspects in Table 6. We relied on three different platforms, namely, Pearl[12], Summit[13] and Theta[14], along with other architectures, for our evaluations.

Table 6: Summary of the Evaluation.

Benchmark	**Platforms /(Architectures)**	**Science Metric(s)**	**Performance Metric(s)**
`cloud-mask`	Pearl (V100) Summit (V100)	Accuracy	Scalability
`stemdl`	Summit (V100)	Accuracy, F1	-
`candle-uno`	Theta (A100)	-	Throughput
`tevelop`	K80, P100, V100 A100, RTX3080, RTX3090	NNSE	Training Time

4.1 Results for the `cloud-mask` Benchmark

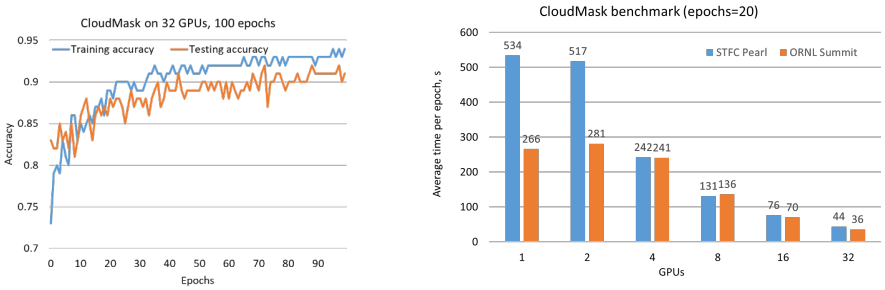
We show the masking accuracy for the training and validation cases in Figure 2a, and the scalability results in Figure 2b. We show two different performance results. In the former, we show how the accuracy of the classification varies against the number of epochs, either trained or tested. The latter shows how the benchmark training scales (average time per epoch) on the Pearl and Summit platforms when the number of GPUs are varied up to 32. There are a number of observations here:

- The accuracy improves with the number of epochs (both testing and training), but they do not exceed 95% of the accuracy shows by the Bayesian mask-based ground truth. However, this has to be interpreted very carefully. The Bayesian-based mask is not necessarily the best either [17]. Hence the sub-optimal outputs does not mean, the ML model is not being effective. We are exploring different means for verifying the real accuracy of the model (such as using data from LIDAR and ground sensors).

[12] https://www.turing.ac.uk/research/asg/pearl
[13] https://www.olcf.ornl.gov/summit/
[14] https://www.alcf.anl.gov/alcf-resources/theta

– Pearl offers better scalability when more than two GPUs are used, while for Summit this has to be four GPUs. However, interestingly, both Pearl and Summit are based on V100 GPUs with totally two different configurations. However, there are performance differences between these platforms when a few GPUs are used. A more detailed investigation is needed both on the scalability and why few GPUs offer sub-optimal performance.

It is very important to note that these conclusions would not have been possible without these initial evaluations.



(a) (b)

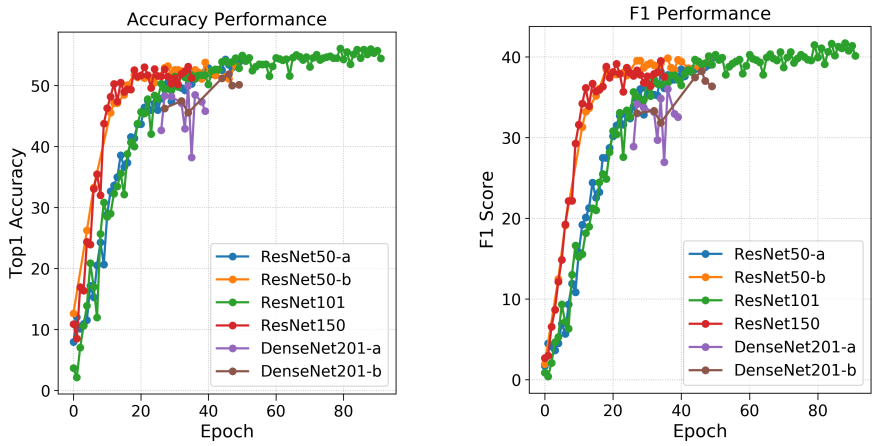
Fig. 2: Performance of `cloud-mask`. The classification accuracy against the number of epochs (on Pearl), and the training scalability of benchmark both on Pearl and Summit platforms are shown in (a) and (b), respectively.

4.2 Results for the `stemdl` Benchmark

We used the Namsa simulation code[15] on the Summit to generate CBED patterns for well over 60,000 solid-state materials, representing nearly every known crystal structure, on which we used the reference implementation. Although the classification accuracy is the ultimate metric, this is influenced by a number of hyper-parameters that underpin our network architecture. As such, it is important to ensure that the the best classification is achieved through hyper-parameter search. Although various techniques exist for hyper-parameter search, and that itself can be a separate benchmarking challenge, here we show the validation accuracy and F1-score for various hyper-parameter sets. There are a number of observations here, but to highlight two: first, as expected, hyper-parameters have an overall influence on the rate and best performance of the benchmark, and secondly the performance converges rapidly for some of the hyper-parameter settings, namely, for the ResNet-101 model. We also show how the accuracy can further be improved from baseline performance in Figure 4,

[15] https://www.osti.gov/biblio/1631694

where the raw performance is marked as (1), along with various optimizations, including, pre-processing (2), time augmentation (3), regularization (4), and by using deeper models (5). These optimizations improve the accuracy from 14% to 57% through these optimizations.



(a) (b)

Fig. 3: Performance of `stemdl` on the Summit platform. The classification accuracy and F1-Score against the number of epochs for various hyper-parameter settings are shown in (a) and (b), respectively. See text for more details.

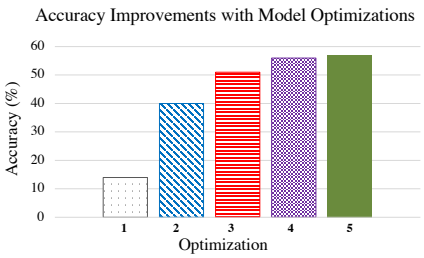


Fig. 4: Accuracy improvements.

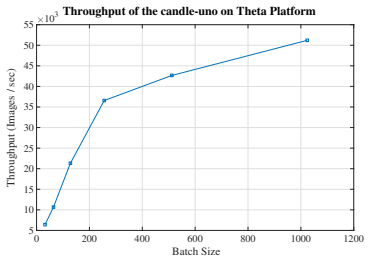


Fig. 5: Throughput of `candle-uno`.

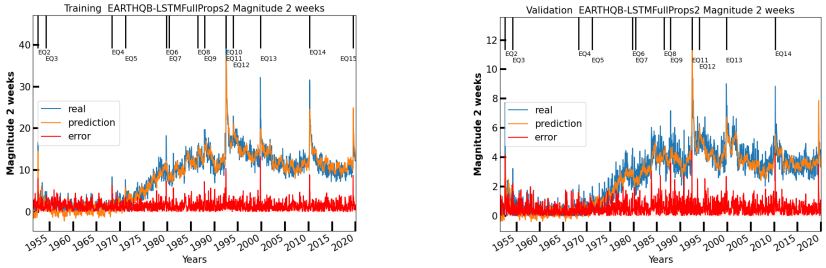
4.3 Results for the `candle-uno` Benchmark

We used the reference implementation on the ThetaGPU platform. As stated before, our metric is throughput (i.e., number of samples processed per second)

for varying batch sizes on a single GPU. We present the results in Figure 5. The results show that the the overall throughput increases with the batch size, showing a trend of saturation, and highlights that more investigation is needed to qualify future implementations, especially across different platforms.

4.4 Results for the `tevelop` Benchmark

The `tevelop` benchmark is evaluated by using it to predict earthquakes over the Southern Californian region. The earthquake data is often binned to generate the spatial time series, and for this evaluation, we consider the bin size of two-weeks. With this, we used our reference model with three baseline implementations, namely, LSTM, TFT and Transformer-based models. We first present the performance results of the LSTM-based model focused on science metric in Figure 6. The results show that ML can, indeed, offer significant benefits. Additional examples ranging from a week to a year are presented in [7].



(a) (b)

Fig. 6: Performance of the `tevelop` in predicting earthquakes, for two-week window periods. The training performance and the validation accuracy are shown in (a) and (b), respectively covering real and predicted values and the error.

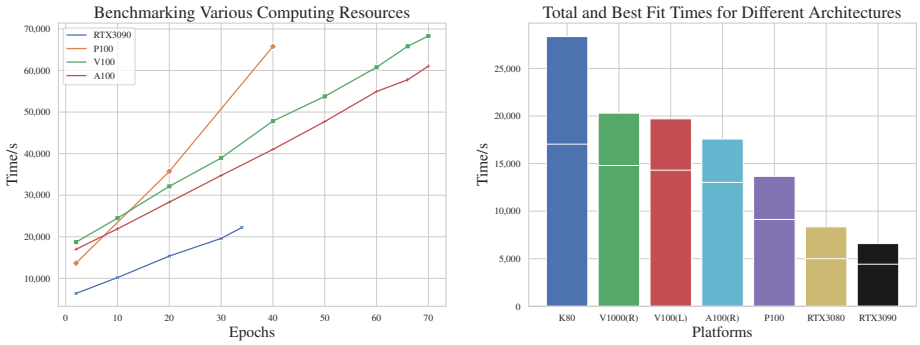
To compare and contrast the performance of different baseline models, we use a subset of the full dataset (which has 2,400 pixels) consisting of 500 most active pixels, divided at the ratio of 4:1 for training and and validation. We then compare these models, across a number of time periods, ranging from two-weeks to four years, and compare their normalized NSE (NNSE) values, with the interpretation of increasing NNSE values imply better predictions. We show the resulting performance in Table 7. A more detailed set of examples, and illustrations can be found in [7]. Finally, we compare the performance of this benchmark on different architectures, and show the results in Figure 7.

5 Conclusions

In this paper, we have discussed the initiatives of the MLCommons Science Working Group for advancing the AI for Science through science-specific bench-

Table 7: Comparison of different models for earthquake prediction.

	LSTM		TFT		Transformer	
Period	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
2 weeks	0.902	0.869	0.931	0.885	0.893	0.856
4 weeks	0.896	0.883	-	-	0.866	0.883
2 months	0.887	0.881	-	-	0.865	0.881
3 months	0.925	0.893	0.976	0.922	0.919	0.881
6 months	0.950	0.900	0.972	0.882	0.954	0.896
1 year	0.923	0.865	0.976	0.853	0.955	0.876
2 years	0.928	0.830	-	-	0.855	0.830
4 years	0.937	0.770	-	-	0.817	0.770



(a) (b)

Fig. 7: Evaluation of the `tevelop` benchmark across a range of architectures and storage systems. Figure (a) shows the training performance while (b) shows the impact of different storage systems (such as, local HDD, local NVMe, NFS).

marks. By collaboratively working with multiple communities, covering various international laboratories, academic institutes and industries, the working group has succeeded in identifying a number of key scientific problems, and developed benchmarks for them. While this is a notable step forward for AI benchmarking, it is significant step for AI benchmarking focused on science. The working group is also actively working on a number of future benchmarks, drawing expertise from various domains. These future benchmarks will cover additional domains, and will also include a variety of classes of ML algorithms, such as surrogate models, inference- and training-based evaluations, and generative models, to mention a few. The future work will also give emphasis to the FAIR aspects of the data, ensuring that all our datasets are FAIR compliant. The working group is aspiring to support submissions of evaluations, so that the community is aware of performance benefits of different systems.

We are very hopeful that this initiative becomes beneficial to the scientific community in a number of different ways, such as supporting easy selection of ML algorithms for a given scientific problem, or for pedagogical purposes. With such purposes, we are hopeful the combined effect of MLCommons is likely to make a significant difference in the AI community.

Acknowledgements

We would like to thank Samuel Jackson from the Scientific Machine Learning Group at the Rutherford Appleton Laboratory (RAL) of the Science and Technology Facilities Council (STFC)(UK) for his contributions towards the Cloud Masking benchmark. This work was supported by Wave 1 of the UKRI Strategic Priorities Fund under the EPSRC grant EP/T001569/1, particularly the 'AI for Science' theme within that grant, by the Alan Turing Institute and by the Benchmarking for AI for Science at Exascale (BASE) project under the EPSRC grant EP/V001310/1, along with the Facilities Funding from Science and Technology Facilities Council (STFC) of UKRI, NSF Grants 2204115 and 2204115, and DOE Award DE-SC0021418. This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This research also used resources from the Oak Ridge and Argonne Leadership Computing Facilities, which are DOE Office of Science user facilities, supported under contracts DE-AC05-00OR22725 and DE-AC02-06CH11357 respectively, and from the PEARL AI resource at the RAL, STFC. This work would not have been possible without the continued support of MLCommons and MLCommons Research, and in particular, we thank Peter Mattson, David Kanter and Vijay Janapa Reddi for their leadership and help.

References

1. Callaway, E.: It will change everything: DeepMind's AI makes gigantic leap in solving protein structures. Nature **588**, 203–204 (2020)
2. Department of Energy: Artificial Intelligence for Science in the US Department of Energy, https://science.osti.gov/Initiatives/AI, [Last Accessed: 30th June, 2022]
3. Earthquake Data, https://github.com/laszewsk/mlcommons-data-earthquake, [Last accessed 30th June 2022]
4. ECP-CANDLE: Benchmarks. GitHub, https://github.com/ECP-CANDLE/Benchmarks/tree/master/Pilot1/Uno, [Last accessed 30th June 2022]
5. Farrell, S., et al.: MLPerf HPC: A Holistic Benchmark Suite for Scientific Machine Learning on HPC Systems (2021), https://arxiv.org/abs/2110.11466
6. Fox, G., Hey, T., Thiyagalingam, J.: Science data working group of MLCommons research. Web Page, https://mlcommons.org/en/groups/research-science/, [Last Accessed: 30th June, 2022]
7. Fox, G., Rundle, J., Donnellan, A., Feng, B.: Earthquake nowcasting with deep learning. Geohazards **3**(2), 199 (Apr 2022)

8. Fox, G.C., von Laszewski, G., Knuuti, R., Butler, T., Kolesar, J.: MLCommons Science Benchmark Earthquake Code, https://bityl.co/COro
9. Henghes, B., Pettitt, C., Thiyagalingam, J., Hey, T., Lahav, O.: Benchmarking and scalability of machine-learning methods for photometric redshift estimation. Monthly Notices of the Royal Astronomical Society **505**(4), 4847–4856 (May 2021)
10. Henghes, B., Thiyagalingam, J., Pettitt, C., Hey, T., Lahav, O.: Deep learning methods for obtaining photometric redshift estimations from images. Monthly Notices of the Royal Astronomical Society **512**(2), 1696–1709 (Feb 2022)
11. Hey, T., Butler, K., Jackson, S., Thiyagalingam, J.: Machine learning and big scientific data. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences **378**(2166), 20190054 (March 2020)
12. Jackson, S., Cox, C., Thiyagalingam, J., Hey, T.: sciml-bench: SciML benchmarking suite for AI for science: Cloud masking benchmark. GitHub (2021), https://github.com/stfc-sciml/sciml-bench/tree/master/sciml_bench/benchmarks/slstr_cloud, [Last accessed 30th June 2022]
13. Jumper, J., Evans, R., Pritzel, A., et al.: Highly accurate protein structure prediction with alphafold. Nature **596**, 583–589 (2021)
14. Laanait, N., Borisevich, A., Yin, J.: A database of convergent beam electron diffraction patterns for machine learning of the structural properties of materials (2019), https://www.osti.gov/servlets/purl/1510313/
15. Laanait, N., Romero, J., Yin, J., Young, M.T., Treichler, S., Starchenko, V., Borisevich, A., Sergeev, A., Matheson, M.: Exascale deep learning for scientific inverse problems (2019), https://arxiv.org/abs/1909.11150
16. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021)
17. Merchant, C.J., Harris, A.R., Maturi, E., Maccallum, S.: Probabilistic physically based cloud screening of satellite infrared imagery for operational sea surface temperature retrieval. Quarterly Journal of the Royal Meteorological Society **131**(611), 2735–2755 (2005)
18. Nash, J., Sutcliffe, J.: River flow forecasting through conceptual models part i – a discussion of principles. Journal of Hydrology **10**(3), 282–290 (1970)
19. Pan, J.: Probability flow for classifying crystallographic space groups. In: Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI. pp. 451–464. Springer (2020)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer (2015)
21. STEMDL Benchmark: STEMDL Benchmark. GitHub, https://github.com/at-aaims/stemdl-benchmark, [Last accessed 30th June 2022]
22. Tanaka, A., Tomiya, A., Hashimoto, K.: Deep Learning and Physics. Springer, Singapore (2021)
23. Thiyagalingam, J., Leng, K., Jackson, S., Papay, J., Shankar, M., Fox, G., Hey, T.: sciml-bench: SciML benchmarking suite for AI for science. GitHub (2021), https://github.com/stfc-sciml/sciml-bench, [Last accessed 30th June 2022]
24. Thiyagalingam, J., Shankar, M., Fox, G., Hey, T.: Scientific machine learning benchmarks. Nature Reviews Physics (April 2022)
25. Index of Pilot1 CANDLE-UNO Benchmark, https://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot1/combo, [Last accessed 30th June 2022]
26. Wilkinson, M.D., et al.: The FAIR guiding principles for scientific data management and stewardship. Scientific Data **3**(1) (Mar 2016)