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, machine learning, or in general, 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 and experimental methods, covering various domains of

sciences, such as material, life, and environmental sciences, particle physics and astronomy [17,18,42,19,43,47,4,21,41]. With AI and ML becoming underpinning technologies for science, there is a considerable amount of attention on several aspects, 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 of those AI/ML techniques, 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 progress 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 [29]. 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 [15]. These groups are open to the public, including academics and researchers from other institutions. 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) [16]. Each of these research working groups, as highlighted, focuses on a specific domain where AI can be transformational.

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, our overall experiences and lessons in developing these initial set of benchmarks, and how we intend to maintain these initiatives over the coming years.

The rest of this paper is organised 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.

2 MLCommons Science Working Group

2.1 About the Working Group

The Science working group [16] was an early member of MLCommons Research created by an active 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 (for example [6]). 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 international organizations. The WG meets on a fortnightly basis, with well over 50 recorded meetings until May 2022. The group also works with a number of other groups, such as MLCommons HPC WG [32,31], where there are a number of overlapping issues of interest.

The overall mission of the group entails collaborative engagements across different domains of sciences, including material, life, environmental, and earth sciences as well as particle physics and astronomy, to mention a few.

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,

1. identifying a number of representative scientific problems where AI can make a difference,
2. engineering at least one ML solution to the problem, to be considered as a baseline implementation,
3. identifying relevant datasets upon which the ML models can be trained or tested,
4. identifying a scientifically-driven metric that can help recognizing the scientific advancement to the problem,
5. curating and publishing those relevant datasets,
6. publishing the scientific results that can help the communities to develop improve these solutions, and
7. 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. These are,

1. Cloud masking (`cloud-masking`) [20] — atmospheric sciences.
2. Space group classification of solid state materials from Scanning Transmission Electron Microscope (STEM) data using Deep Learning (DL) (`stemdl`) [35] — solid state physics.
3. Time evolution operator (`tevelop`) [11] exemplified using predicting earthquakes — earth sciences.
4. predicting tumor response to single and paired 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 [46] to distribution to versioning of the datasets. These benchmarks have a multitude of purpose, which are discussed at length in [42,19]. 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 results, in particular in science, can be interpreted in different ways, and in rather subjective manners, leading to the whole initiative not serving the intended purpose. As such, establishing a policy around the rules and guidelines for evaluating and reporting results for the benchmarks from the WG is an important step. Other WGs have their own policies, for example [27,30]. The 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. This is essential, as no two benchmarks are the same, nor their functional behavior or scientific goals. As such, tailoring the policies for each and every benchmark is unavoidable. In general, the policies will cover the evaluation of benchmarks under two divisions, namely, Open and Closed divisions. Benchmark evaluation under the Open division will primarily focus on achieving better scientific results or outperforming existing performance (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 for evaluation and often will list permissible changes. The exact list of permissible changes is likely to vary across benchmarks, but in general, pre- and/or post-processing, and data are often kept fixed, with freedom to change or fine-tuning the underling ML model. The same line of argument applies for policies around submission of results. 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. The policy will also likely to cover the general format of the results to facilitate automation or maintaining a league-table.

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. For instance, large variations of the SST in the Pacific can cause anything from severe drought, to heavy rainfall, to tropical cyclones. Estimation of Sea Surface Temperature (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 on board the Sentinel-3 satellite, a mission operated jointly by the European Space Agency and by the European Organization for the Exploitation of Meteorological Satellites. 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 methods. An example input and output images are given in Figure 1. We also summarize the key features of this benchmark in Table 1. The overarching scientific objective, objective of the benchmark, description of the relevant dataset, and reference implementation are given below.

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

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 Resolution: [2400 × 3000 × 6] and [1200 × 1500 × 3] Size: 180 GB Source: CEDA Location: STFC Servers [20]
Reference implementation	SciML-Bench Cloudmask Benchmark [38]

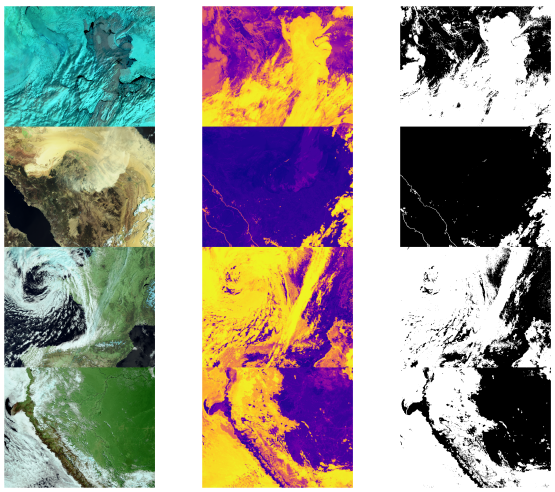


Fig. 1: Cloud mask example. The left column shows the raw images from the Sentinel-3 satellite while the images on the right column shows the predicted probability that a particular pixel is cloud.

a cloud or to a clear sky. Historically this has been performed using Bayesian techniques [28], which can lead to sub-optimal outputs in a number of cases. The scope of the `cloud-mask` benchmark is to explore whether ML-driven algorithms can outperform the Bayesian techniques or even can be a replacement technique.

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, in the absence of any ground truth, this is a valid and perfect

choice. However, with Bayesian masks not always being accurate or offering a gold-standard for labeling masks, the resulting model is likely to suffer from learnability issues, which sets the perfect challenge 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, especially when trained across a number of Graphical Processing Units (GPUs).

Data: The masking can be performed across different satellite imaging modalities. This particular benchmark relies on satellite imagery obtained from the Sentinel-3 satellites, particularly from the Sea and Land Surface Temperature Radiometer (SLSTR) 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 CEDA archive [10], the curated datasets are available as part of this benchmark, located in object store within the Science and Technology Facilities Council (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 [37], implemented using TensorFlow and Keras [1,5], with the support for distributed training using TensorFlow's native library, Distribute 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 [20].

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 nanoscale 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 [23] and [35]
Reference Implementation	AAIMS repository [39] Model: ResNet-50 Run Instructions: [39] Time-to-solution: 40 minutes on 60 V100 GPUs
References	[25,23,22,36,24]

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 data sample [35] 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 [23]. 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 [36] and [24] along with the reference implementation deposited into the AAIMS repository [39].

3.3 CANDLE-UNO (`candle-uno`)

The CANDLE (Exascale Deep Learning and Simulation Enabled Precision Medicine for Cancer) project [3] aims to implement deep learning architectures that are relevant to problems in cancer research. These architectures address problems at three biological scales: cellular (Pilot1 or P1), molecular (Pilot2 or P2), and population (Pilot3 or P3). The CANDLE initiative has three mainstreams of benchmarks to cover these pilots. In summary:

- The Pilot1 (P1) benchmarks are 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.
- Pilot2 (P2) benchmarks are formed out of problems and data originating at the molecular level. The high level goal of the problems behind the P2 benchmarks are molecular dynamic simulations of proteins involved in cancer, specifically the RAS protein.
- Pilot3 (P3) benchmarks are formed out of problems and data originating at the population level. The high level goal of the problems behind the P3 benchmarks is to predict cancer recurrence based on patient-related data.

The UNO version of the CANDLE suite is a P1 benchmark. 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 of Uno 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 effectiveness of drugs and how they can be developed to cure the tumor cells. The ML component aims to accelerate this part through being able to predict the 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.

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 [44,45]
Reference implementation	Github [8] Model: Multi-task Learning-based custom model Code: [8] Instructions: [9] Ideal performance: 10,667 samples/sec on a single A100 GPU for a batch size of 64

Data: Combined dose response data relies on a number of sources that are specific drug responses to cancer conditions. We summarise these sources in Table 4. 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 5. The training can be accelerated by using a pre-staged dataset file. This static dataset can, however, be prebuilt. The datasets are publicly available from the CANDLE site [44]. 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 [8]. It can be run in both training and inference modes. However, this benchmark is defined to be training focussed. A dedicated script in this repository downloads all required datasets. The primary metric to evaluate for this application is the model throughput (samples

Table 4: Data sources of the dose response information.

#	Source	Parameters
1	The Cancer Therapeutics Response Portal	CTRP
2	The Genomics of Drug Sensitivity in Cancer	GDSC
3	The NCI Sarcoma	SCL
4	The NCI Small Cell Lung Cancer	SCLC
5	The NCI-60 Human Cancer Cell Line Screen single drug response	NCI60
6	A Large Matrix of Anti-Neoplastic Agent Combinations drug pair response	ALMANAC.FG ALMANAC.FF ALMANAC.1A
7	The Genentech Cell Line Screening Initiative	gCSI
8	The Cancer Cell Line Encyclopedia	CCLE

Table 5: 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

per 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 reaches convergence. With the required packages in the software stack, Uno can be run on diverse systems. More details on running Uno can be found in [9,8].

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 these evolutions, using earthquake as the driving example. We summarise the key features of the benchmark in Table 6.

Table 6: Summary of the `tevelop` Benchmark

Description	Earthquake Forecasting.
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 [7]
Reference Implementation	[14]
References	[11,13,12,26,14,7]

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) [33]. It is also possible to qualitatively asses prediction by comparing the observed earthquake, if one desires, but the benchmarks relies on the former [11,13].

Data: The United States Geological Survey (USGS) supplies earthquake data for the entire world, based on various measurements. The benchmark relies on a very small subset of the data from USGS focused between the regions of four degrees of latitude (32 degN to 36 degN) and six degrees of longitude (-120 degS

to -114 degS) region, effectively covering Southern California. 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. Southern California is divided into a grid of 40×60 with each each pixel covering actual zone of 0.1 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) [26]-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 [26]. Details of the current reference models can be found in [13,11].

4 Results from Initial Evaluations

As mentioned in previous sections, the Science WG is focused on developing benchmarks for advancing AI for Science, and hence scientific discoveries. To this end, we presented the benchmarks that the WG has consolidated since the formation of the WG. Each benchmark is accompanied by at least one reference implementation, with the aim of setting the trend for open competition.

In this section, we present some of the early results obtained initial evaluations of these benchmarks. As this is the first instance 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. Furthermore, as these benchmarks are in the process of being evaluated on different platforms, and as such, the results presented here may not appear to be uniform across benchmarks. We outline these aspects in Table 7. Additional details around Pearl, Summit and Theta systems can be found at [40], [34] and [2], respectively.

Table 7: Summary of the Evaluation.

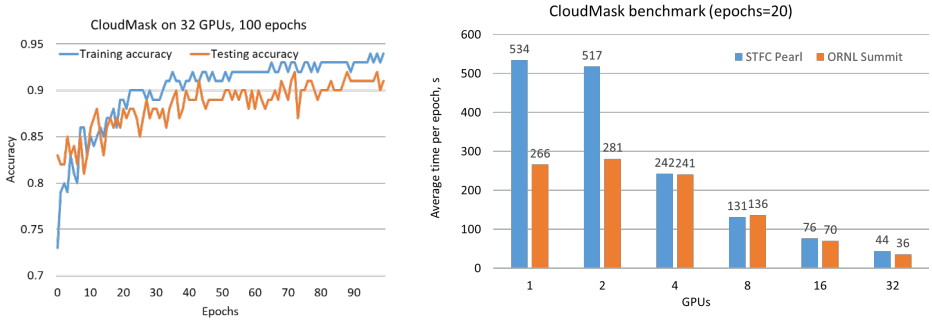
Benchmark	**Platforms /(Architectures)**	**Science Metric(s)**	**Performance Metric(s)**
`cloud-mask`	Pearl (V100) Summit (V100)	Accuracy	Scalability
`stemdl`	Summit (V100)	Accuracy, F1	-
`candle-uno`	Theta (A100)	-	Throughput
`tevelop`	K80, P100 V100, A100 RTX3080, RTX3090	NNSE	Training Time

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 (STFC) and Summit (ORNL) when the number of GPUs are varied up to 32. There are a number of observations here:

- Although the accuracy improves with the number of epochs (both testing and training), 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 [28]. Hence the sub-optimal outputs does not mean, the ML model is not being effective. There are two possible avenues to verify the real accuracy of the model. One is to compare this against LIDAR data (obtained from ground sensors). However, the region where the Sentinel-3 and LIDAR sensors overlap is very limited, and hence the available data is considerably limited. This means, we need to evaluate the model only for a small subset of the overall data. Second option is to use this to estimate the SST values, and compare this against the real readings measured by the ground sensors (such as those obtained from buoys).
- As for scalability, there are a number of different observations. 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. The former tightly integrates two DGX-2 nodes, with each node housing 16 GPUs, while the latter has six GPUs per node. 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)

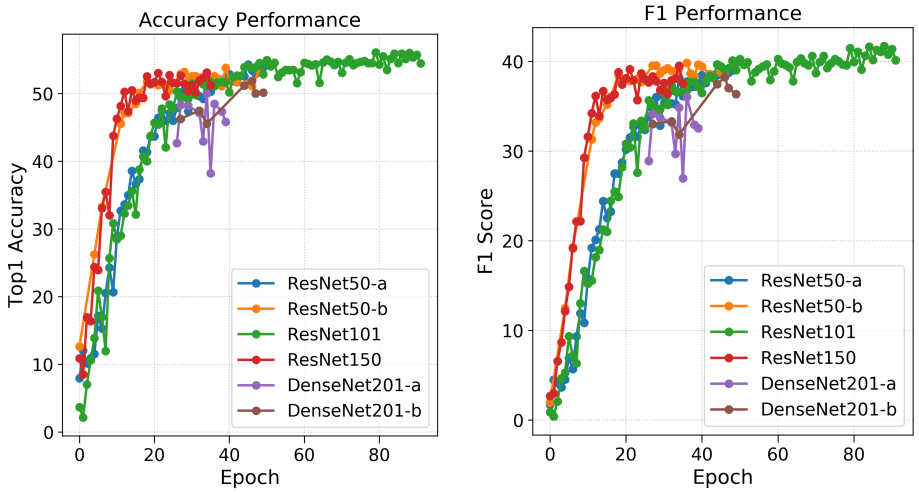
Fig. 2: Performance of `cloud-mask` on the PEARL platform. In the figure (a) shows the variation of classification accuracy against the number of epochs, and (b) shows the training scalability of benchmark compared on Pearl and Summit. See text for more details.

4.2 Results for the `stemdl` Benchmark

With the reference implementation [39], we used newly developed multi-GPU and multi-node electron scattering simulation codes [25] on the Summit to generate CBED patterns for well over 60,000 solid-state materials, representing nearly every known crystal structure. Although the classification accuracy is the ultimate metric, this is influenced by a number of hyperparameters that underpin our network architecture. As such, it is important to ensure that the the best classification is achieved through hyperparameter search. Although various techniques exist for hyperparameter 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, hyperparameters have an overall influence on the rate and best performance of the benchmark, and secondly the performance converges rapidly for some of the hyperparameter settings, namely, for the ResNet-101 model. We also show how the accuracy can further be improved from baseline performance in Figure 4, 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.

4.3 Results for the `candle-uno` Benchmark

We used the reference implementation on the ThetaGPU platform [2] (ALCF, Argonne), consisting of NVIDIA A100 GPUs. 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



(a) (b)

Fig. 3: Performance of `stemdl` on the Summit platform. Figures (a) and (b) shows the variation of classification accuracy and F1-Score against the number of epochs for various hyperparameter settings, respectively. See text for more details.

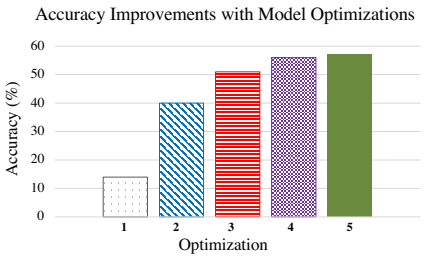


Fig. 4: Accuracy improvements.

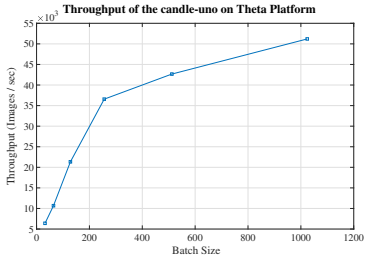


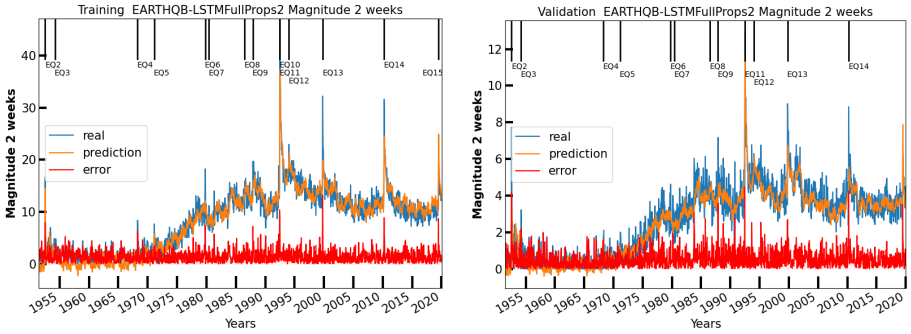
Fig. 5: Throughput of `candle-uno`.

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

As stated in previous sections, we will be using the benchmark to predict earthquakes over the Southern Californian region. As stated before, earthquake data is often binned to generate the spatial time series, and for this evaluation we have considered the bin size of two-weeks. With this, we used our reference

implementation for this evaluation. There are 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 [11].



(a) (b)

Fig. 6: Evaluation of the `tevelop` to predict earthquakes since 1955, for two-week window periods over the Californian region. Figure (a) shows the training performance while (b) shows the validation accuracy. We show both real and predicted values along with the error.

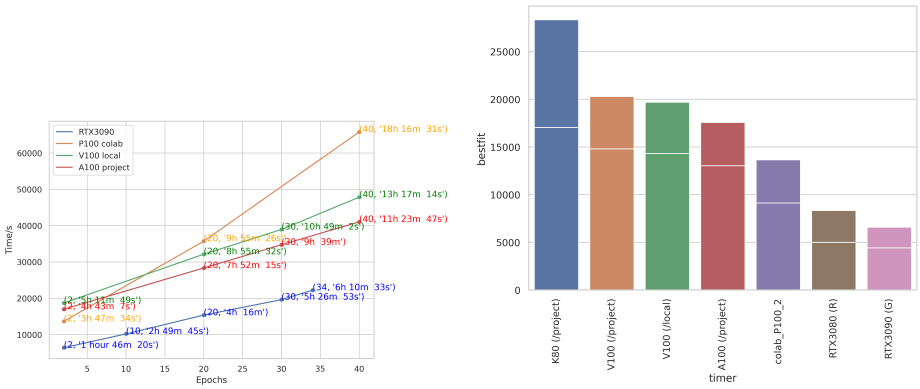
Next, to compare and contrast the performance of different baseline models, we use a subset of the full dataset contained (which has $2,400$ pixels) consisting of 500 most active pixels, with 400 of them for training and 100 for validation. We then compare these models (with the same bin size), across a number of time periods, ranging from two-weeks to four years, and compare their normalized NSE (NNSE) values, with the NNSE value zero signifying the worst and value of unity signifying the best possible prediction. We show the resulting performance in Table 8. A more detailed set of examples, and illustrations can be found in [11]. 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 benchmarks. 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. These include, `cloud-mask` from atmospheric sciences, `stemdl` from condensed matter physics, `candle-uno` from healthcare, and

Table 8: 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 (such as, local HDD, local NVMe, NFS)

`tevelop` from earth sciences. Here, each benchmark include a data set, science- and/or performance-based metrics, and one or more reference implementations. All these benchmarks support both Open and Closed divisions.

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 sub-

missions 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 of the Science and Technology Facilities Council (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 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-AC05-00OR22725, respectively, and from the PEARL AI resource at the Rutherford Appleton Laboratory, Science and Technology Facilities Council.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. (ALCF), A.L.C.F.: Theta/thetagpu. https://www.alcf.anl.gov/alcf-resources/theta, [Last accessed 27th May 2022]
3. Argonne National Laboratory: ECP CANDLE benchmarks, https://github.com/ECP-CANDLE/Benchmarks
4. Callaway, E.: It will change everything: Deepmind's ai makes gigantic leap in solving protein structures. Nature **588**, 203–204 (2020)
5. Chollet, F., et al.: Keras. https://keras.io (2015)
6. DOE OSTI: Artificial intelligence for science in the US department of energy. Web Page, https://science.osti.gov/Initiatives/AI, accessed: 2022-5-7
7. Earthquake Data. Google Drive (May 2022), https://github.com/laszewsk/mlcommons-data-earthquake, [Last accessed 27th May 2022]
8. ECP-CANDLE: Benchmarks. GitHub (May 2022), https://github.com/ECP-CANDLE/Benchmarks/tree/master/Pilot1/Uno, [Last accessed 27th May 2022]
9. ECP-CANDLE: Candle UNO Readme. GitHub (May 2022), https://github.com/ECP-CANDLE/Benchmarks/blob/develop/Pilot1/Uno/README.AUC.md, [Last accessed 27th May 2022]
10. for Environmental Data Analytics (CEDA), C.: Centrte for environmental data analytics (ceda). https://www.ceda.ac.uk/, [Last accessed 27th May 2022]
11. Fox, G., Rundle, J., Donnellan, A., Feng, B.: Earthquake nowcasting with deep learning. Geohazards **3**(2), 199 (Apr 2022)
12. Fox, G.C.: Science Data Processing: Examples in Jupyter Notebook (Apr 2010), https://ggle.io/58zu, [Last accessed 27th May 2022]
13. Fox, G.C.: Earthquakes for Real. Presentations (May 2022), https://ggle.io/58zv, [Last accessed 27th May 2022]
14. Fox, G.C., von Laszewski, G., Knuuti, R., Butler, T., Kolesar, J.: Mlcommons science benchmark earthquake code (may 2022), https://bityl.co/COro
15. Gennady Pekhimenko,Vijay Janapa Reddi: MLCommons research working group, home page. Web Page, https://mlcommons.org/en/groups/research/, accessed: 2022-5-7
16. Geoffrey Fox, Tony Hey, Jeyan Thiyagalingam: Science data working group of MLCommons research. Web Page, https://mlcommons.org/en/groups/research-science/, accessed: 2020-12-3
17. 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)
18. 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)
19. 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)
20. Jeyan Thiyagalingam, Kuangdai Leng, Samuel Jackson, Juri Papay, Mallikarjun Shankar, Geoffrey Fox, Tony Hey: sciml-bench: SciML benchmarking suite for AI for science. GitHub (2021), https://github.com/stfc-sciml/sciml-bench, accessed: 2022-5-7

21. Jumper, J., Evans, R., Pritzel, A., et al.: Highly accurate protein structure prediction with alphafold. Nature **596**, 583–589 (2021)
22. Laanait, N., J Yin, A.B.: SMC data challenges: Towards a Universal Classifier for Crystallographic Space Groups (2020), https://bityl.co/COsc
23. 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/
24. 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
25. Laanait, N., Yin, J., USDOE: Namsa (Aug 2019), https://www.osti.gov/biblio/1631694
26. 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)
27. Mattson, P., Reddi, V.J., Cheng, C., Coleman, C., Diamos, G., Kanter, D., Micikevicius, P., Patterson, D., Schmuelling, G., Tang, H., Wei, G.Y., Wu, C.J.: MLPerf: An industry standard benchmark suite for machine learning performance. IEEE Micro **40**(2), 8–16 (Mar 2020)
28. 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)
29. History of MLCommons. Web Page, https://mlcommons.org/en/history/, accessed: 2022-5-7
30. MLCommons: MLPerf training rules. GitHub, https://github.com/mlperf/training_policies/blob/master/training_rules.adoc, accessed: 2020-5-1
31. HPC MLCommons training working group. Web Page (May 2022), https://mlcommons.org/en/groups/training-hpc/, accessed: 2022-5-7
32. Inside HPC, MLPerf-HPC working group seeks participation,. Web Page, https://insidehpc.com/2020/02/mlperf-hpc-working-group-seeks-participation/, accessed: 2020-5-1
33. Nash, J., Sutcliffe, J.: River flow forecasting through conceptual models part i – a discussion of principles. Journal of Hydrology **10**(3), 282–290 (1970)
34. (OLCF), O.R.L.C.F.: Summit. https://www.olcf.ornl.gov/summit/, [Last accessed 27th May 2022]
35. ORNL: 10.13139/OLCF/1510313 (May 2022), https://doi.ccs.ornl.gov/ui/doi/70, [Last accessed 27th May 2022]
36. Pan, J.: Probability flow for classifying crystallographic space groups. In: Nichols, J., Verastegui, B., Maccabe, A.B., Hernandez, O., Parete-Koon, S., Ahearn, T. (eds.) Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI. pp. 451–464. Springer International Publishing, Cham (2020)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
38. Samuel Jackson, Caroline Cox, Jeyan Thiyagalingam and Tony Hey: 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 27th May 2022]

39. STEMDL Benchmark: STEMDL Benchmark. GitHub (May 2022), https://github.com/at-aaims/stemdl-benchmark, [Last accessed 27th May 2022]
40. STFC PEARL Facility, S.C.D.S.: Pearl. https://www.turing.ac.uk/research/asg/pearl, [Last accessed 27th May 2022]
41. Tanaka, A., Tomiya, A., Hashimoto, K.: Deep Learning and Physics. Springer, Singapore (2021)
42. Thiyagalingam, J., Shankar, M., Fox, G., Hey, T.: Scientific machine learning benchmarks. Nature Reviews Physics (April 2022)
43. Tran, N.H., Xu, J., Li, M.: A tale of solving two computational challenges in protein science: neoantigen prediction and protein structure prediction. Briefings in Bioinformatics **23**(1) (Dec 2021)
44. Index of Pilot1 CANDLE-UNO Benchmark (May 2022), https://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot1/combo, [Last accessed 27th May 2022]
45. Index of Pilot1 CANDLE-UNO Benchmark (May 2022), https://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot1/uno, [Last accessed 27th May 2022]
46. Wilkinson, M.D., et al.: The FAIR guiding principles for scientific data management and stewardship. Scientific Data **3**(1) (Mar 2016)
47. Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., Baker, D.: Improved protein structure prediction using predicted interresidue orientations. Proceedings of the National Academy of Sciences **117**(3), 1496–1503 (2020)