FERMILAB-POSTER-25-0130-STUDENT

Artificial Intelligence Benchmarking

Anjay Krishnan, CCI Intern

What is AI benchmarking?

AI benchmarking is a method for evaluating the effectiveness of an AI model using a set of standardized metrics, for example, high school-level math exams. These benchmarks and their results will enable ranking various AI models based on their effectiveness in performing a specific task.

Purpose

The software that we design allows for the metadata for each benchmark to be output in a clear and easy-to-read format. The output of our software can be used to determine the effectiveness of a benchmark, so that the user can know which benchmarks to use for evaluations.

Methods

1. We initially collected data on 75 different benchmarks and stored it in a YAML file format, as shown below.

```yaml
-
  - date: '2024-05-01'
    description: The date of availability of the benchmark. If an official release date is not available, use the date of
      adding the entry.
    condition: required
  - version: TODO
    description: The version number of the benchmark
    condition: optional
  - last_updated: 2024-05
    description: 'The date when the entry was last updated. Format: YYYY-mm-dd'
    condition: optional
  - expired: null
    description: An indication if the benchmark is no longer valid.
    condition: optional
  - valid: 'yes'
    description: Identifies if the benchmark is valid at the time of review.
    condition: required
  - name: Jet Classification
    description: The name of the benchmark.
    condition: required
  - url: https://github.com/fastmachinelearning/fastml-science/tree/main/jet-classify
    description: The main URL for this benchmark.
    condition: required
  - doi: TODO
    description: A DOI number that may be associated with the benchmark.
    condition: optional
  - domain: Particle Physics
    description: The scientific domain this benchmark belongs to.
    condition: required
  - focus: Real-time classification of particle jets using HL-LHC simulation features
    description: Short summary of the focus of this benchmark.
    condition: required
  - keywords:
    - classification
    - real-time ML
    - jet tagging
    - QKeras
    description: List of keywords relevant for the benchmark.
    condition: '>=1'
```

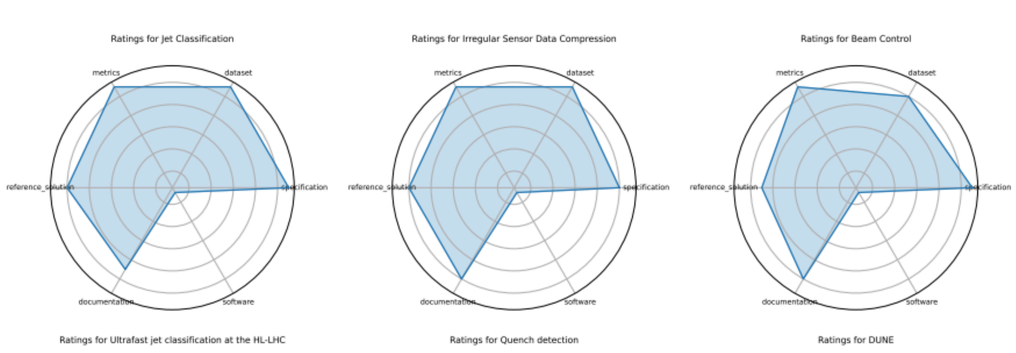
An example entry from the input YAML file is shown to the left.

2. We wrote code to convert the YAML file into Markdown and LaTeX formats. This included a combined file containing all the benchmarks, as well as individual files for each benchmark.

3. We then added additional features to the command line arguments and then added ratings based on the criteria provided by the MLCommons team.

Ratings

We rated each benchmark on a scale of 0-10 in 5 different categories of metrics, dataset, specification, software, documentation, and reference solution.



Example radar plots that show the ratings for three benchmarks in an easily readable manner. These radar plots are implemented into the Markdown and LaTeX tables for reference.

- This manuscript has been authored by FermiForward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.
- This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Community College Internship (CCI)

Results

We were able to successfully build the software and implement the list of command line prompts or features which are listed below.

- --files / -i
  Specifies the YAML file(s) to be processed. This argument is required and can accept one or more files.
- --format / -f
  Sets the desired output format: either md for Markdown or tex for LaTeX. This is a required argument.
- --outdir / -o
  Indicates the directory where output files should be saved. This allows users to control where their processed files go.
- --authortruncation
  Truncates the number of authors displayed in index or summary tables to keep the output clean and concise. Useful for long author lists.
- --columns
  Lets users specify a subset of columns to include in the output, using a comma-separated list (e.g., --columns name,date,domain). This supports custom views.
- --check
  Runs validation checks on the YAML input files to ensure all required fields and formatting rules are met. This mode does not produce an output file.
- --index
  Generates individual pages for each benchmark entry. If Markdown format is selected, it also creates an index.md page with links to all entries.
- --noratings
  Removes the rating columns from the output file. This is useful if the user wants a simpler view of the benchmark information.
- --withcitation
  Adds a BibTeX citation row to the Markdown output. This is helpful for researchers who want to quickly find citation info.
- --required
  When used with --columns, it treats all listed columns as required and checks that they are present in every YAML file.
- --standalone / -s
  For LaTeX output, includes the full LaTeX document structure (preamble, document environment), making it ready to compile directly.

List of command line prompts/features of the software to make reading and creating the output files easier

We were also able to successfully create a Markdown and LaTeX table of any input YAML file(s) given. A few table entries from the Markdown and LaTeX are shown below.

date	name	domain	focus	keywords	task_types	metrics
2024-05-01	Jet Classification	Particle Physics	Real-time classification of particle jets using HL-LHC simulation features	classification, real-time ML, jet tagging, QKeras	Classification	Accuracy, AUC
2024-05-01	Irregular Sensor Data Compression	Particle Physics	Real-time compression of sparse sensor data with autoencoders	compression, autoencoder, sparse data, irregular sampling	Compression	MSE, Compression ratio
2024-05-01	Beam Control	Accelerators and Magnets	Reinforcement learning control of accelerator beam position	RL, beam stabilization, control systems, simulation	Control	Stability, Control loss

1. @article{hawks2022fastml, title={Fast Machine Learning for Science: Benchmarks and Dataset}, author={Hawks, Ben and Tran, Nhan and others}, year={2022}, url={https://arxiv.org/abs/2207.07958}

Output MD format (3 listings with a few columns shown, and one citation in footnote is shown to the left)

Date	Name	Domain	Focus	Task Types	Metrics	Models	Citation
2020-09-07	MMLU (Massive Multitask Language Understanding)	Multidomain	Academic knowledge and reasoning across 57 subjects	Multiple choice	Accuracy	GPT-4o, Gemini 1.5 Pro, o1, DeepSeek-R1	[1] ⇒
2023-11-20	GPQA Diamond	Science	Graduate-level scientific reasoning	Multiple choice, Multi-step QA	Accuracy	o1, DeepSeek-R1	[2] ⇒
2018-03-14	ARC-Challenge (Advanced Reasoning Challenge)	Science	Grade-school science with reasoning emphasis	Multiple choice	Accuracy	GPT-4, Claude	[3] ⇒
2025-01-24	Humanity's Last Exam	Multidomain	Broad cross-domain academic reasoning	Multiple choice	Accuracy		[4] ⇒
2024-11-07	FrontierMath	Mathematics	Challenging advanced mathematical reasoning	Problem solving	Accuracy		[5] ⇒
2024-07-18	SciCode	Scientific Programming	Scientific code generation and problem solving	Coding	Solve rate (percent)	Claude3.5-Sonnet	[6] ⇒
2025-03-13	AIME (American Invitational Mathematics Examination)	Mathematics	Pre-college advanced problem solving	Problem solving	Accuracy		[7] ⇒
2025-02-15	MATH-500	Mathematics	Math reasoning generalization	Problem solving	Accuracy		[8] ⇒
2024-04-02	CURIE (Scientific Long-Context Understanding, Reasoning and Information Extraction)	Multidomain Science	Long-context scientific reasoning	Information extraction, Reasoning, Concept tracking, Aggregation, Algebraic manipulation, Multimodal comprehension	Accuracy		[9] ⇒
2023-01-26	FEABench (Finite Element Analysis Benchmark)	Computational Engineering	FEA simulation accuracy and performance	Simulation, Performance evaluation	Solve time, Error norm	FEniCS, deal.II	[10] ⇒
2024-07-12	SPIQA (Scientific Paper Image Question Answering)	Computer Science	Multimodal QA on scientific figures	Question answering, Multimodal QA, Chain-of-Thought evaluation	Accuracy, F1 score	Chain-of-Thought models, Multimodal QA systems	[11] ⇒

11 listings shown and 3 references in BibTex format are shown to the left

References

[1] D. Hendrycks, C. Burns, S. Kadavath, *et al.*, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2021. [Online]. Available: https://arxiv.org/abs/2009.03300.

[2] D. Rein, B. L. Hou, A. C. Stickland, *et al.*, *Gpqa: A graduate-level google-proof q and a benchmark*, 2023. [Online]. Available: https://arxiv.org/abs/2311.12022.

[3] P. Clark, I. Cowhey, O. Etzioni, *et al.*, "Think you have solved question answering? try arc, the ai2 reasoning challenge," in *EMNLP 2018*, 2018, pp. 237–248. [Online]. Available: https://allenai.org/data/arc.