Snowmass2021 - Letter of Interest

IceCube and IceCube-Gen2 Machine Learning

Thematic Areas: (check all that apply □/■)
□ (CompF1) Experimental Algorithm Parallelization
□ (CompF2) Theoretical Calculations and Simulation
■ (CompF3) Machine Learning
□ (CompF4) Storage and processing resource access (Facility and Infrastructure R&D)
□ (CompF5) End user analysis
□ (CompF6) Quantum computing
□ (CompF7) Reinterpretation and long-term preservation of data and code

Contact Information:
Claudio Kopper (Michigan State University) [kopppercl@msu.edu]

Authors (alphabetical):
Brian A. Clark (Michigan State University) [baclark@msu.edu],
Theo Glauch (Technische Universität München) [theo.glauch@tum.de],
Mirco Hünnefeld (TU Dortmund University) [mirco.huennefeld@tu-dortmund.de],
Claudio Kopper (Michigan State University) [kopppercl@msu.edu],
Hieu Le (Michigan State University) [lehieu1@msu.edu],
Jessica Micallef (Michigan State University) [micall12@msu.edu],
Maria Prado Rodriguez (University of Wisconsin–Madison) [maria.pradorodriguez@icecube.wisc.edu],
Johannes Wagner (Drexel University) [jmw464@drexel.edu]
on behalf of the IceCube[1] and IceCube-Gen2[2] Collaboration [analysis@icecube.wisc.edu]

Abstract:
The IceCube Neutrino Observatory is a cubic kilometer neutrino detector deployed at the South Pole, focused on detecting GeV to EeV neutrinos. IceCube measures neutrinos by detecting the optical Cherenkov photons produced in neutrino-nucleon interactions. To study the the properties of incident neutrinos, IceCube employs numerous machine learning algorithms, including convolutional, recurrent, and graph neural networks for purposes of reconstruction, classification, and uncertainty estimation. In this letter, we summarize the priorities for the collaboration moving forward, emphasizing the need to (1) cultivate expertise on how to adopt and evaluate ML methods to IceCube data, (2) develop ML algorithms which leverage domain knowledge, and (3) integrate and utilize various ML accelerator technologies. These new, more flexible methods coupled with increased computing capabilities will be important as upgrades to the IceCube detector are deployed in the next decade, including the Icecube-Upgrade and IceCube-Gen2.

[1] Full author list available at https://icecube.wisc.edu/collaboration/authors/snowmass21_icecube
[2] Full author list available at https://icecube.wisc.edu/collaboration/authors/snowmass21_icecube-gen2

The IceCube Neutrino Observatory [1] is designed to measure neutrinos from GeV to EeV energies. IceCube does this by deploying a hexagonal grid of Digital Optical Modules (DOMs) along 86 vertical strings deep in the glacier at South Pole. Each DOM contains a photomultiplier tube, and detects the Cherenkov radiation from relativistic charged particles emitted during neutrino interactions. IceCube leverages machine learning for a variety of tasks to extract information about the incident neutrinos from these detected photons—including reconstruction of energy and direction, classification of interaction type and neutrino flavor, and uncertainty estimation.

IceCube will benefit from a strong community focus on keeping up to date with developments in ML theory, ML applications and tools coming from industry and coordination of ML efforts across the various frontiers. Current key goals in ML for IceCube are:

1. **Expertise Cultivation:** a continued effort to adopt and evaluate ML methods to IceCube data, both at the detector level and at the analysis levels; user education on ML methods to make sure our students and scientists are aware of the most efficient methods available for their particular problems;
2. **Algorithms and Methods Development:** development of new ML methods specific to the problems in high-energy physics where techniques from industry applications do not apply (e.g. high-quality simulation data for detectors is abundant and can be generated easily).
3. **Hardware Integration:** the integration of various ML accelerators, such as FPGAs, TPUs, and modern GPU architectures in on-premises and cloud settings; integration of data pipeline/movement tools to allow for efficient ML training.

IceCube adopts a variety of ML techniques in order to maximize scientific output. Traditional ML such as tree-based learners and shallow neural networks are predominantly used for classification tasks. These methods enhance the efficiency of event selections and as such constitute core contributions to many IceCube analyses. Other applications range from regression tasks (energy, stochasticity, and uncertainty estimation) to analysis method development [2; 3]. In contrast to maximum likelihood estimation (MLE) techniques [4; 5; 6; 7], the application of deep learning-based techniques can enhance the capabilities of the detector and usually provide a vastly superior reconstruction speed, and will thus become a crucial tool to operate the detector, particularly for time critical applications such as the real-time alert system [8].

Expertise Cultivation

IceCube seeks to adopt and evaluate machine learning methods to IceCube data, both at the detector level and at the analysis levels. This includes user education on ML methods to make sure our students and scientists are aware of the most efficient methods available for their particular problems.

The physics of neutrino interactions is invariant under translation and rotation, and deep learning architectures utilize these symmetries. The majority of IceCube's existing works focuses on the applications of convolutional neural networks (CNNs). Applications for CNNs on IceCube include both neutrino interaction reconstruction and event classification [9; 10; 11], with networks trained and optimized separately for events in the 100 GeV-PeV range and the range below 200 GeV. The speed of these CNNs is several orders of magnitude faster than previous MLE methods, while their performance is usually comparable or better.

However, CNN reconstructions require nontrivial preprocessing and conditioning of IceCube data. Formatting IceCube's hexagonal array into a uniform grid that a CNN expects requires a transformation to a rectangular grid structure and/or reducing the number of strings used. Future detectors such as the IceCube Upgrade and IceCube-Gen2 use more complicated geometries that would be difficult to adapt to current networks. Furthermore, regardless of geometry, information is lost in transforming the data for use by the network since summary statistics are used to account for all hits on an optical module in a time window or during the entire event. Thus, for both current and future applications, the collaboration has been exploring other deep learning methods, for example, recurrent neural networks (RNNs) and graph neural networks (GNNs).

In RNNs, recurrent nodes are represented as time series, and as such are a natural representation of the IceCube pulse data. This allows a pulse-based RNN to utilize the entire pulse series instead of relying on summary variables as used in the previously described CNN approach. An alternative approach utilizes a series of event "snapshots" by accumulating the measured charge at each DOM within certain time windows. CNNs are used to extract features from these three dimensional snapshots which are then further processed by an RNN. IceCube is also exploring other, new network architectures, such as "WaveNet"-like algorithms based on 1D dilated causal convolutions [12].

Graphs are independent of spatial geometries, and so their abstract nature gives GNNs freedom to handle irregular geometries and flexibility in weight propagation, which is useful for the more irregular geometries of future detector upgrades. IceCube currently has several ongoing efforts based on GNN tools, including particle identification, low energy event reconstruction, and uncertainty estimation. These networks operate on various IceCube graph representations with nodes as DOMs, singular signal pulses and signal point clouds respectively, and are currently being developed using graph attention network (GAT) [13; 14], graph convolutional network (GCN) [15] and dynamic graph CNN (DGCNN) [16; 17] frameworks with comparable performance to current baseline reconstruction algorithms.

Algorithms and Methods Development

IceCube continues to focus on development of new ML methods specific to the problems in high-energy physics where techniques from industry applications do not always directly apply.

One place where direct application of "industry" deep-learning methods often suffers is the difficulty in including domain knowledge—for example, the linear scaling of light yield with deposited energy. One approach IceCube is developing to tackle these issues is in the development of joint MLE/DL methods. These methods aim to combine strengths of MLE and DL by utilizing neural networks in a maximum-likelihood setting. A neural network is employed to approximate the computationally complex and often intractable step of computing the likelihood. This is performed in an implicit likelihood-free approach based on [18; 19] as well as an approach that explicitly defines the likelihood [10]. In this approach, the likelihood can utilize gradients to speed up convergence and the Hessian may be used to approximate uncertainties on the reconstruction.

Another unusual feature of applying ML in the HEP physics setting is the abundance of high-quality Monte Carlo training samples. Often in industry applications, the size of training samples is relatively small, or very costly to obtain [20]. For a detector like IceCube, large, statistically independent samples can be produced, with careful control of systematic uncertainties such as the ice properties. This has the potential to enable use of complex network architectures which would otherwise be intractable because of limited training, testing, and validation samples.

Hardware Integration

IceCube aims to use and integrate various ML accelerators, such as FPGAs, TPUs, and modern GPU architectures in on-premises and cloud settings, as well as integration of data pipeline/movement tools to allow for efficient ML training.

The availability of new and faster hardware is important, especially as the size of available data sets grow, and models grow in complexity. IceCube has experience leveraging distributed computing infrastructures to enhance on-premises resources, as demonstrated through a recent "Cloudburst" experiment on the Open Science Grid, where nearly 52k GPUs across three continents were used simultaneously for IceCube simulation [21]. Use of dedicated hardware such as Tensor Processing Units (TPUs), especially those in the cloud, have the potential to accelerate training by orders of magnitude [22]. As datasets grow, tools to efficiently move data into the memory of GPUs will also be important—for example, by connecting GPUs and FPGAs via PCIe [23], or through "GPUdirect"-like technology, where GPUs communicate directly with Infiniband servers to load data, eliminating the CPU as an intermediary [24].

In conclusion, while machine learning tools are already a crucial part of the IceCube toolchain, with increasing detector complexity and better understanding and calibration of the glacial ice and other detector properties, ML-based methods will become indispensable tools for IceCube data analysis.

References

[1] IceCube, M. G. Aartsen *et al.*, JINST **12**, P03012 (2017), arXiv:1612.05093.

[2] M. Bunse, N. Piatkowski, K. Morik, T. Ruhe, and W. Rhode, in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2018.

[3] M. Börner *et al.*, , Astronomical Society of the Pacific Conference Series Vol. 522, p. 431, 2020.

[4] M. Aartsen *et al.*, JINST **9**, P03009 (2014), arXiv:1311.4767.

[5] D. Chirkin, in *33rd International Cosmic Ray Conference*, p. 0581, 2013.

[6] C. Haack, L. Lu, and T. Yuan, EPJ Web Conf. **207**, 05003 (2019).

[7] F. Bradascio and T. Glüsenkamp, EPJ Web Conf. **207**, 05002 (2019), arXiv:1905.09612.

[8] IceCube, M. Aartsen *et al.*, Astropart. Phys. **92**, 30 (2017), arXiv:1612.06028.

[9] IceCube, M. Huennefeld, PoS **ICRC2017**, 1057 (2018).

[10] IceCube, M. Huennefeld, EPJ Web Conf. **207**, 05005 (2019).

[11] IceCube, M. Kronmueller and T. Glauch, PoS **ICRC2019**, 937 (2020), arXiv:1908.08763.

[12] A. van den Oord *et al.*, (2016), arXiv:1609.03499.

[13] N. Choma *et al.*, (2018), arXiv:1809.06166.

[14] P. Veličković *et al.*, (2017), arXiv:1710.10903.

[15] T. N. Kipf and M. Welling, (2016), arXiv:1609.02907.

[16] Y. Wang *et al.*, (2018), arXiv:1801.07829.

[17] H. Qu and L. Gouskos, Phys. Rev. D **101**, 056019 (2020), arXiv:1902.08570.

[18] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Proc. Nat. Acad. Sci. **117**, 5242 (2020), arXiv:1805.12244.

[19] J. Hermans, V. Begy, and G. Louppe, (2019), arXiv:1903.04057.

[20] G. M. Weiss and F. Provost, Journal of artificial intelligence research **19**, 315 (2003).

[21] I. Sfiligoi, F. Würthwein, B. Riedel, and D. Schultz, Running a pre-exascale, geographically distributed, multi-cloud scientific simulation, in *High Performance Computing*, pp. 23–40, Cham, 2020, Springer International Publishing.

[22] N. Jouppi, C. Young, N. Patil, and D. Patterson, IEEE Micro **38**, 10 (2018).

[23] R. Bittner, E. Ruf, and A. Forin, Cluster Computing **17**, 339 (2014).

[24] G. Shainer *et al.*, Computer Science-Research and Development **26**, 267 (2011).