Towards Interpretable Machine Learning in High-Energy Physics: Development of an Explainability Framework

Applicant: Bornik Nag
Mentor: Dr. Liv Helen Våge
Project duration: 10 Weeks (8 June 2026 - 17 August 2026)
Proposed Start date: 8 June 2026

Overview

Complex machine learning (ML) algorithms such as graph neural networks and particle transformers have played a significant role in modern high-energy physics for extracting signals from large and complicated datasets, and will be an essential tool used by scientists in the High Luminosity Large Hadron Collider upgrade. Yet despite their remarkable predictive power, the inner workings of these models remain largely opaque due to the high-dimensional and nonlinear spaces they operate in. This has resulted in a "black-box" approach to statistical analyses, which raises a fundamental concern: a model's success on test data does not guarantee that it has learned the underlying physics, and it may be correct in only a subset of all possible physical states that the provided system can explore [1]. Knowing how and why a model reaches a given prediction is therefore essential to ensure that its results are based on solid physical theories. The need for such interpretability is even more important in searches for new physics beyond the Standard Model, where the laws may have to be inferred from the data collected at experiments.

However, interpretability is not a single well-defined property. Lipton (2016) argues that it encompasses at least two distinct ideas: transparency, which refers to the degree to which a model's mechanics can be directly understood by humans, and post-hoc explanation, which aims to rationalize the output of a model without requiring insight into its internal structure [2]. While such ideas have been explored across a range of tasks in HEP, these efforts have remained largely isolated, using different techniques on a case-by-case basis. No unified framework exists for benchmarking and comparing these methods across the different models being used in the field [3]. Developing such a framework is the central goal of this project.

Objectives

- **Survey, categorize and benchmark interpretability methods relevant to HEP**
    - Beginning from existing trained models such as jet classifiers, systematically apply both post-hoc explanation tools such as SHAP, LIME, and attention visualization, as well as transparency techniques. Assess whether methods agree on feature importance, whether explanations are stable across various training runs, and if results can be expressed in terms of known physics.
    - Apply and compare explanation methods starting from simpler, more transparent models like boosted decision trees to more complex models such as feedforward networks and particle transformers, and assess where methods agree or diverge.
    - Use techniques like linear probes at each transformer layer to identify where relevant information is encoded, and perform ablation studies to assess whether specific layers can be modified to improve the explainability of model behavior.

- **Develop practical guidelines and a GitHub repository of interpretability tools for the HEP community**
    - Produce a structured document covering when to use which interpretability approach, limitations of each method, performance on different models, and open problems.
    - Release a collection of reusable, model-agnostic Python modules built on PyTorch and scikit-learn for understanding ML models, as well as tutorial notebooks and reproducible examples using open-source data.

Timeline

Weeks 1-2: Examine and read the existing literature regarding ML interpretability in HEP to build a knowledge base. Configure a Python setup to work on the project, starting with the training of a baseline BDT classifier on the JetClass dataset, and apply SHAP to produce a variable importance hierarchy. Then compare it against the BDT's native feature importance as a first benchmark.

Weeks 3-5: Train an MLP on JetClass and systematically compare methods such as SHAP, LIME, integrated gradients and saliency maps. Document where they agree and where they diverge. Then extend the analysis to an attention-based model such as a particle transformer, contrasting attention visualization against the other techniques. Apply linear probes at each transformer layer to identify where physics-relevant information is encoded, and whether performing targeted ablation can improve misclassification rates.

Weeks 6-8: Apply the existing toolkit to a different architecture such as a GNN using PyTorch Geometric, identifying which methods can transfer cleanly and which require tweaking. Use this information to begin abstracting the codebase into small, reusable modules. Attempt to map model behavior onto known physics observables, and use random seeds and bootstrapped datasets to characterize the uncertainty of these methods by performing statistical analysis.

Weeks 9-10: Compile the findings into a practical guidelines document and prepare a presentation for the fellowship committee. Finalize the GitHub repository with tutorial notebooks, comprehensive documentation, sample loaders and reproducible examples, ready for community evaluation.

References and Resources

[1] Grojean, C., Paul, A., Qian, Z., & Strümke, I. (2022). Lessons on interpretable machine learning from particle physics. Nature Reviews Physics, 4(5), 284–286. https://doi.org/10.1038/s42254-022-00456-0

[2] Lipton, Z. C. (2017). The Mythos of Model Interpretability. arXiv [Cs.LG]. Retrieved from http://arxiv.org/abs/1606.03490

[3] Wetzel, S. J., Ha, S., Iten, R., Klopotek, M., & Liu, Z. (2025). Interpretable Machine Learning in Physics: A Review. arXiv [Physics.Comp-Ph]. Retrieved from http://arxiv.org/abs/2503.23616

SHAP: https://github.com/shap/shap
LIME: https://christophm.github.io/interpretable-ml-book/lime.html
JetClass: https://zenodo.org/records/661976