Logic Gate Neural Networks for Jet Substructure Classification

IRIS-HEP Fellowship Project Proposal

Applicant: Alan Chen

Mentors: Liv Våge and Lino Gerlach

Duration: June 22, 2026 – August 30, 2026 (10 weeks)

1. Motivation

Machine learning is increasingly important for processing the large volume of data produced at the Large Hadron Collider (LHC). At the hardware-trigger level, models must operate with very low latency and limited resources. Previous work has demonstrated FPGA-based neural-network inference for jet classification with latencies of approximately 100 ns [1]. However, conventional neural networks still rely on arithmetic operations and require careful quantization and hardware optimization.

Differentiable logic neural networks (LNNs) offer an alternative based on Boolean operations such as AND, OR, and XOR. During training, continuous relaxations allow the choice of logic gate at each node to be learned with gradient descent. The trained network is then discretized into a Boolean circuit that can be compiled for highly efficient inference [2]. This project will investigate LNNs for fast jet classification using *torchlogix*, a PyTorch-based package for training, discretizing, and compiling logic networks [3].

2. Objectives

The project will use the Jet Substructure Classification (JSC) dataset, which contains 16 high-level features for five jet classes: gluon, light quark, W boson, Z boson, and top quark [1]. The objectives are:

1. Develop a reproducible JSC pipeline with validated preprocessing, fixed training/validation/test splits, a conventional neural-network baseline, and torchlogix training and evaluation.
2. Compare relaxed LNN performance during training with the discrete performance of the deployable Boolean circuit, and verify that compiled predictions agree with the discrete PyTorch model.
3. Study the trade-off among classification accuracy, inference speed, and hardware resource usage by varying network depth, width, connectivity, and feature binarization.

3. Procedure

First, existing *torchlogix* examples will be reproduced to validate the workflow from model training to discrete evaluation and compiled C inference. The JSC data pipeline and a compact conventional baseline will then be established under the same data splits and evaluation metrics.

LNN models will be evaluated using overall and per-class accuracy, confusion matrices, the relaxed-to-discrete performance difference, Boolean gate count, compiled CPU inference speed, and available FPGA-oriented resource estimates. A focused architecture study will identify compact models with favorable accuracy–resource–latency trade-offs.

The multiclass output stage will also be examined from a hardware-aware perspective. Current LNNs form class scores by summing groups of Boolean outputs, which may require substantial counter and adder logic. The project will assess whether direct class selection, such as an argmax or tournament-style comparison, could reduce this cost. Compiled C and Verilog-like outputs will be used to connect model design with deployability.

If progress on the core benchmark permits, the project will begin an exploratory study of the much larger COLLIDE-2V physics dataset. This extension will focus on selecting a suitable classification task, developing the required data pipeline, and obtaining preliminary LNN results. The project may also compare with related logic-based approaches. Strong benchmark or software results may contribute to a short publication or a future *torchlogix* software paper, although the primary outcome remains reproducible research software.

4. Deliverables

The main deliverable will be a public GitHub repository containing the data pipeline, baseline and LNN training code, reproducible configurations, evaluation scripts, benchmark results, and documentation. Results will cover relaxed and discrete accuracy, compiled inference, resource estimates, and architecture comparisons. Problems or improvement opportunities found in *torchlogix* will be reported through issues and, where practical, documentation updates, examples, benchmark scripts, or focused pull requests. Results will be summarized in the final IRIS-HEP report and presentation.

5. Timeline

Weeks 1–2: Review the recommended literature, reproduce torchlogix examples, and complete the JSC data pipeline and conventional baseline.

Weeks 3–4: Train initial LNNs; compare relaxed, discrete, and compiled predictions; and establish the initial JSC benchmark.

Weeks 5–7: Optimize JSC architectures and binarization choices with respect to accuracy, compiled inference speed, and FPGA-oriented resource estimates, including a hardware-aware study of multiclass output aggregation.

Weeks 7–9: If progress on the core benchmark permits, begin an exploratory COLLIDE-2V study by selecting a classification task, developing the data pipeline, and obtaining preliminary LNN results.

Week 10: Validate selected configurations, finalize documentation and software contributions, and prepare the report and presentation.

References

[1] J. Duarte et al., “Fast inference of deep neural networks in FPGAs for particle physics,” JINST 13, P07027 (2018).
[2] F. Petersen et al., “Deep Differentiable Logic Gate Networks,” NeurIPS 35 (2022).
[3] L. Gerlach, T. Gerlach, E. Kauffman, and L. Våge, “torchlogix,” software (2026), doi:10.5281/zenodo.18800427.