FERMILAB-POSTER-24-0198-STUDENT

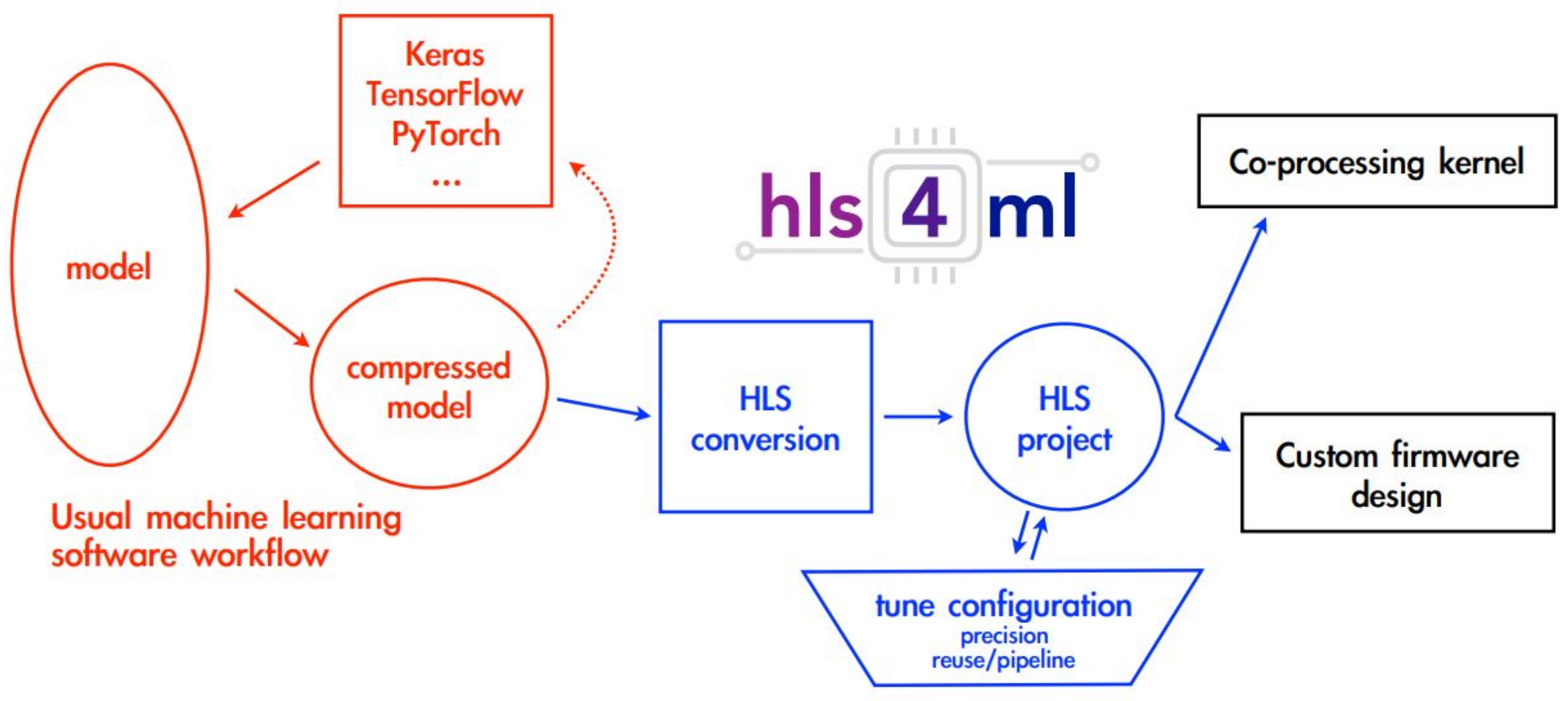
wa-hls4ml: A GNN Surrogate Model for hls4ml

Dennis Plotnikov[1 2], Benjamin Hawks[1], and Nhan V. Tran[1]

[1] Fermi National Accelerator Laboratory [2] Johns Hopkins University

Overview of hls4ml

hls4ml is a pipeline used to convert machine learning models to a form that can be run on a field-programmable gate array (FPGA) or inscribed into an application-specific integrated circuit (ASIC). This has strong applications in high-energy physics, where detector triggers require latency on the scale of nanoseconds, but would benefit greatly from the power of machine learning.

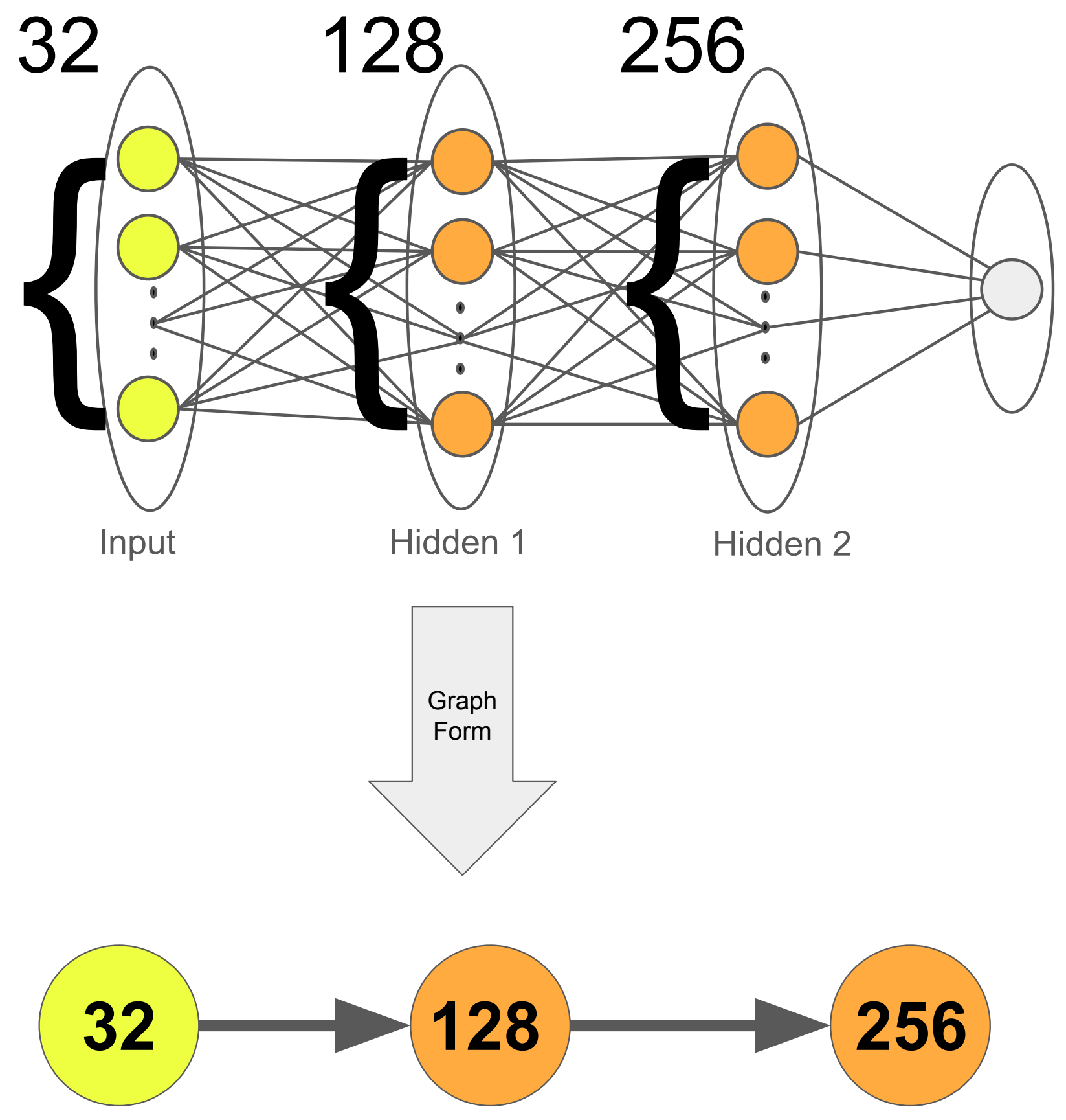


A typical workflow to translate a model into a FPGA implementation using hls4ml (J. Duarte et al 2018 JINST 13 P07027)

Surrogate Modeling

A surrogate model is a time- or resource-efficient model, which can be used to get a rough estimate of a more sophisticated model.

For hls4ml, converting a given input network could take on the order of days. After that, the result could be unuseable, either due to a failed synthesis, or due to the resource consumption being untenable for the application.

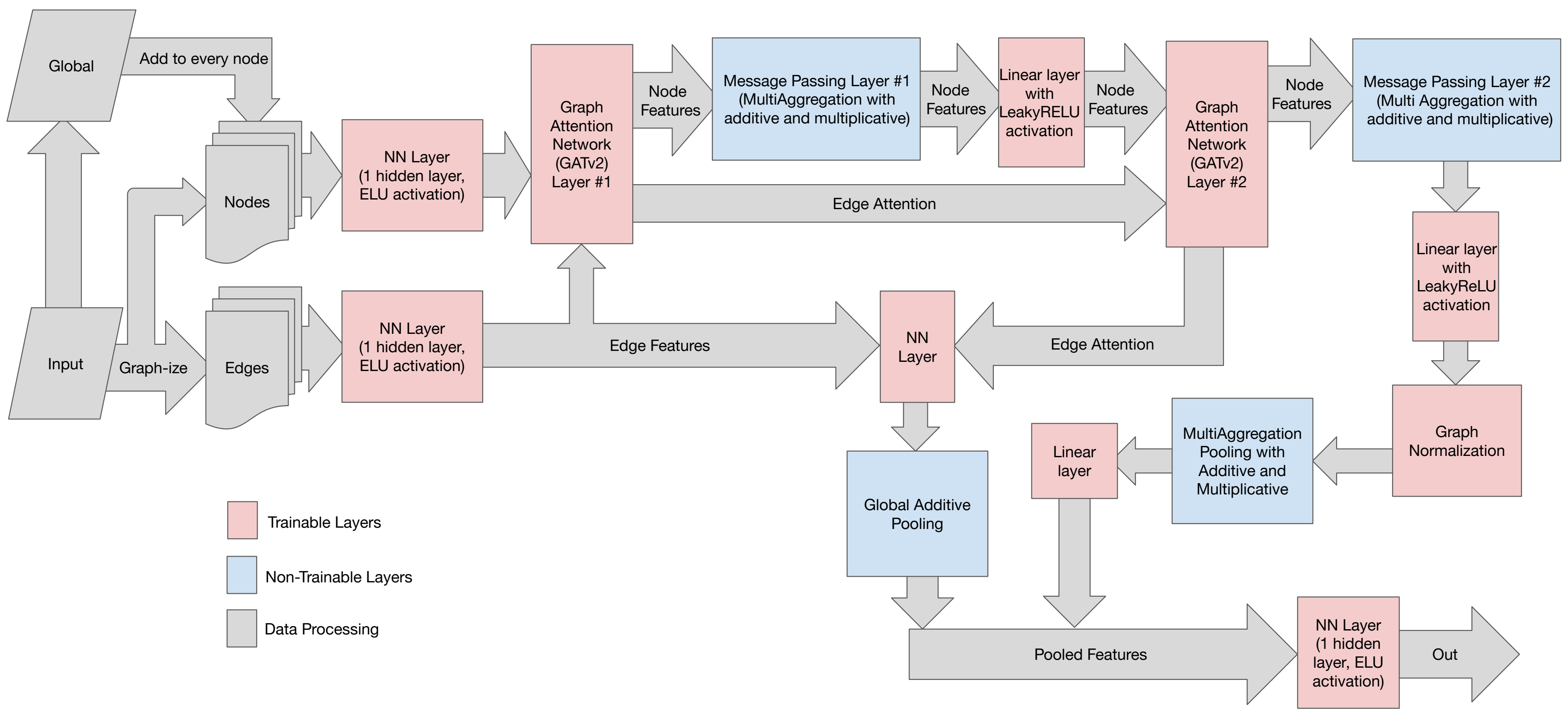


A simple two-hidden-layer feedforward neural network, represented as a three node directed graph.

Modeling Networks with Graphs

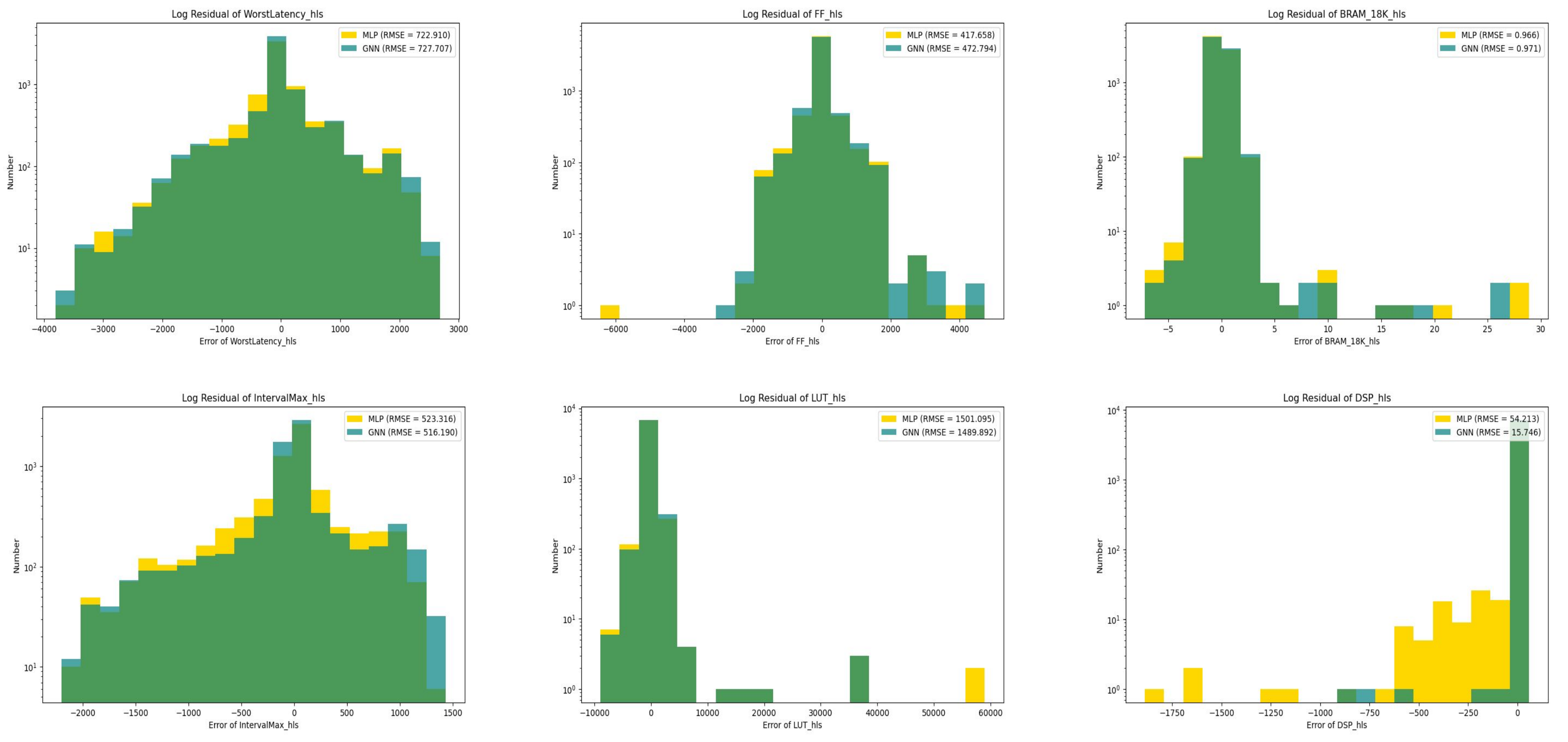
The structure of a typical neural network can be represented in the form of a directed graph. Each node represents a layer of the network, and each edge represents a feedforward connection.

This allows for modeling complex architectures, including skip connections and recurrent networks. Node features (e.g. number of connections in the layer) and edge features (e.g. sparsity of the connection) can both be included.


Global
Add to every node
Input
Graph-ize
Nodes
Edges
NN Layer (1 hidden layer, ELU activation)
NN Layer (1 hidden layer, ELU activation)
Graph Attention Network (GATv2) Layer #1
Node Features
Message Passing Layer #1 (MultiAggregation with additive and multiplicative)
Node Features
Linear layer with LeakyRELU activation
Node Features
Graph Attention Network (GATv2) Layer #2
Node Features
Message Passing Layer #2 (Multi Aggregation with additive and multiplicative)
Edge Attention
Edge Features
NN Layer
Edge Attention
Linear layer with LeakyReLU activation
Graph Normalization
MultiAggregation Pooling with Additive and Multiplicative
Linear layer
Global Additive Pooling
Pooled Features
NN Layer (1 hidden layer, ELU activation)
Out
Trainable Layers
Non-Trainable Layers
Data Processing

The structure of the graph neural network. Node and edge features are passed through graph attention networks and message-passing layers

The graph-based input data for the surrogate model is best handled by a graph neural network (GNN). The network takes data in the form of graphs (represented with an adjacency list), and runs them through two GATv2 graph attention networks (arXiv:2105.14491) and two message-passing graph convolution layers, before pooling all edge and node results with global features. A final neural network layer allows all three types of data to inform a prediction.



Residual histograms comparing the GNN (cyan) to a control multi-layer perceptron (yellow). On interval, LUT, and DSP prediction, the GNN is able to attain a lower RMSE even on the same data, showing one advantage of the graph representation

Results

The wa-hls4ml surrogate model is capable of simulating the resource consumption of the final synthesis results within a reasonable margin. On 3 out of 6 tested target regression features, the GNN has a lower RMS error rate than a standard multi-layer perceptron on the same collection of heterogeneous data (containing 2-layer and 3-layer input models) The GNN structure is able to predict DSP unit usage with a significantly lower RMS error. Additionally, the GNN is not limited by graph architectures, allowing it to extrapolate to unseen architectures. More data is needed on this extrapolation performance.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 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 Science Undergraduate Laboratory Internships Program (SULI).

Fermi National Accelerator Laboratory

This work was supported in part by National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC-2112167, CNS-2100237, CNS-2120019, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100Gbps networks.