Machine Learning Inference: SOFIE

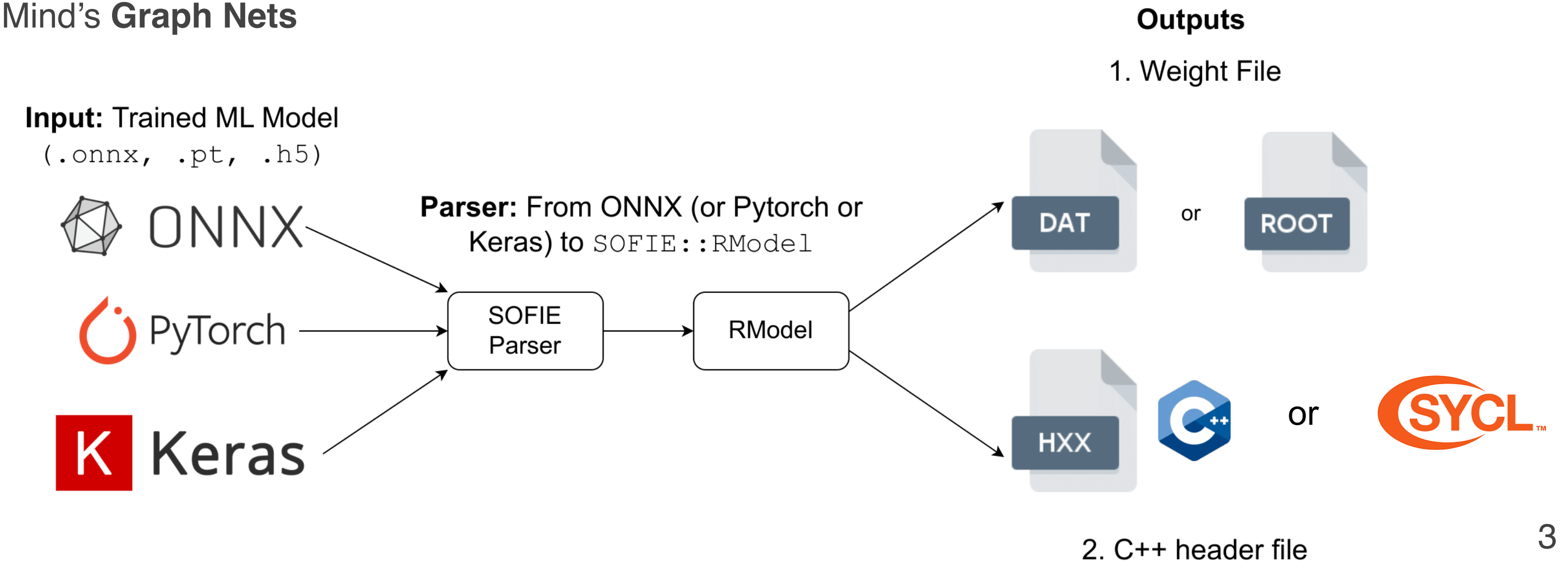
Motivation

- **Fast Evaluation of Machine Learning models** is more and more relevant
- ML tools like Tensorflow/PyTorch have functionality for inference
  - can run only for their models
  - usage in a C++ environment can be cumbersome
  - require heavy dependence
- A standard for describing deep learning models:
  - ONNX ("Open Neural Network Exchange")
  - cannot describe all possible deep learning models (e.g. GNN) fully
- ONNXRuntime: an efficient inference engine based on ONNX
  - can work in both C++ and Python
  - supporting both CPU and GPU
  - can be challenging to integrate in the HEP ecosystem
    - control of threads, dependencies, etc..
    - not optimised for single-event evaluation

Machine Learning Inference in ROOT

SOFIE : System for Optimised Fast Inference code Emit

- **Input**: trained ML model file
  - **ONNX**: Common standard for ML models
  - **Tensorflow/Keras** and **PyTorch** models (with reduced support than ONNX)
  - Since 6.32 support message passing **GNNs** from DeepMind's **Graph Nets**
- **Output**: **generated C++ code**
  - Easily **invokable directly** from C++ (plug-and-use)
  - **Minimal dependency** (on BLAS only)
  - Can be **compiled at run time** using ROOT Cling JIT and can be **used in Python.**



GPU Extension of SOFIE

- Extended SOFIE functionality to produce **GPU** code using **SYCL**

```
// generate SYCL code internally
model.GenerateGPU();
// write output header and data weight file
model.OutputGeneratedGPU();
```

- **Minimise overhead of data transfers** between host and device
- **Manage buffers efficiently, declaring them at the beginning**
- Use libraries for **GPU Offloading**: GPU BLAS from Intel one API and PortBLAS for other GPUs
- **Fuse operators** when possible in a single kernel
- **Replace conditional check** with relational functions

model.hxx

```
namespace TMVA_SOFIE_Linear_event{

struct Session {

Session(std::string filename ="") {
   if (filename.empty()) filename =
"Linear_event.dat";
   std::ifstream f;
   f.open(filename);
   // read weight data file
   ………………..
}
std::vector<float> infer(float*
tensor_input1){
```

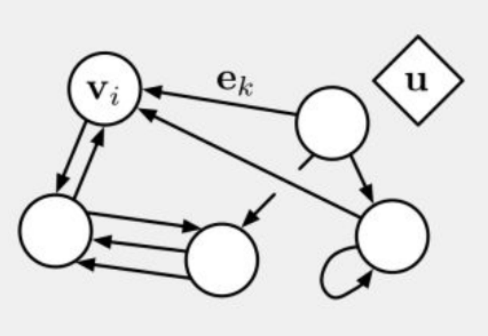
with SYCL code

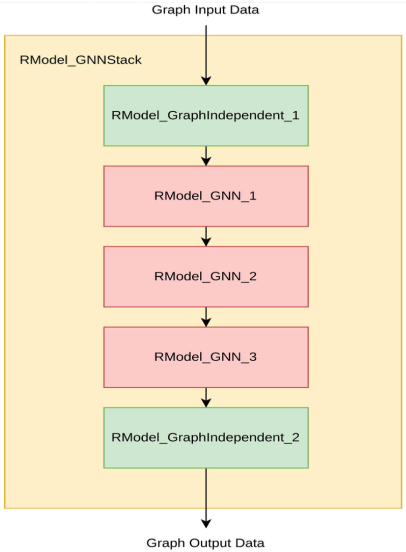
```
#include "Model.hxx"
// create session class
TMVA_SOFIE_Model::Session
ses("model_weights.dat");
//-- event loop
for (ievt = 0; ievt < N; ievt++) {
   // evaluate model: input is a C float array
   float * input = event[ievt].GetData();
   auto result = ses.infer(input);
```

Inference code needs to be linked against oneAPI MKL libraries and compiled using SYCL compiler

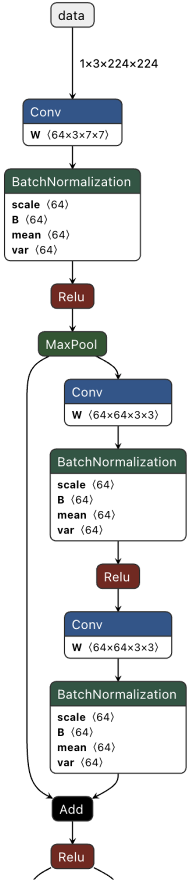
SOFIE GNN Support

- Since ROOT version 6.32 support inference of **GNN**s
  - parsing available for GNNs built from DeepMind's Graph Net library
  - supporting a LHCb model for full event interpretation (arXiv:2304.08610)
- Developed **C++ classes** for representing **GNN structure**.
  - based on SOFIE **`RModel`** and the **`ROperator`** classes which provide the functionality to generate C++ inference code
- **Python code** (based on PyROOT) for parsing from the Graph Nets models





ONNX Supported Operators

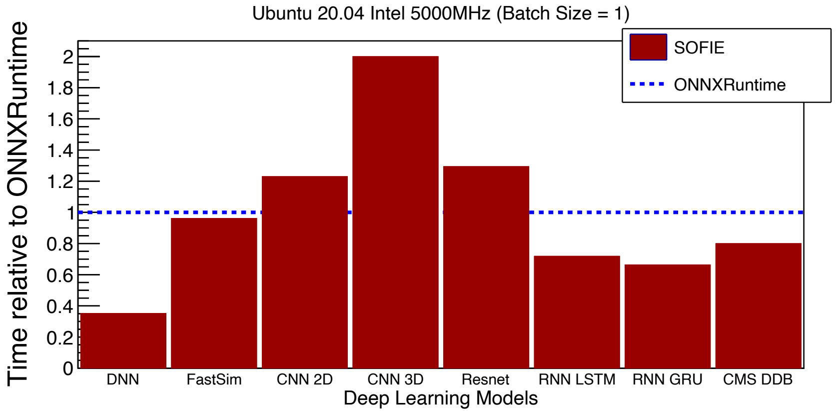


Operators implemented in ROOT	**CPU**	**GPU**
Perceptron: Gemm	✓	✓
Activations: Relu, Selu, Sigmoid, Softmax, Tanh, LeakyRelu, Swish	✓	✓
Convolution and **Deconvolution** (1D, 2D and 3D)	✓	✓
Pooling: MaxPool, AveragePool, GlobalAverage	✓	✓
Recurrent: RNN, GRU, LSTM	✓	✓
Layer Unary operators: Neg, Exp, Sqrt, Reciprocal, Identity	✓	✓
Layer Binary operators: Add, Sum, Mul, Div	✓	✓
Other Layer operators: Reshape, Flatten, Transpose, Squeeze, Unsqueeze, Slice, Concat, Reduce, Gather	✓	✓
BatchNormalization, LayerNormalization	✓	✓
Custom operator	✓	

- current CPU support available in **ROOT 6.30**
- GPU/SYCL is implemented in a ROOT PR

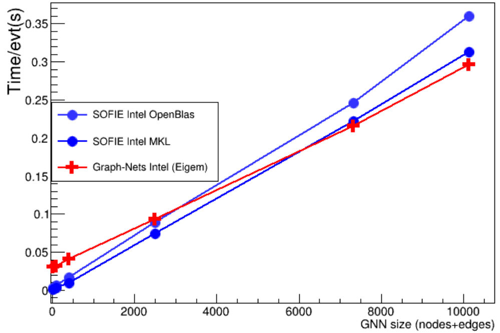
Benchmarking Time of Inference

CPU event performance of **SOFIE** vs **ONNXRuntime**



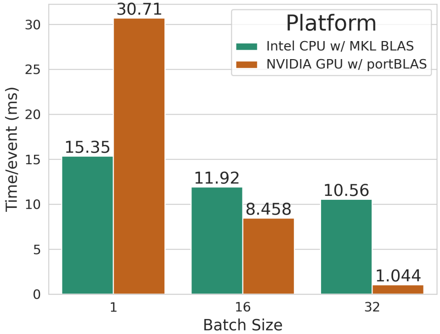
CPU time for **GNN inference**

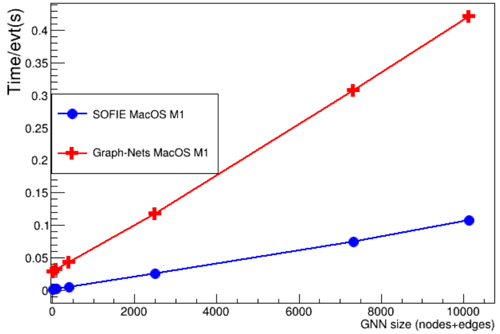
- varying GNN size (node + edges)



GPU (SYCL) vs CPU performance

- using a Resnet model with varying batch size





Summary

- **SOFIE**, fast and easy-to-use inference engine for Deep Learning models, is available in ROOT
  - Can be easily integrated with other ROOT tools (*RDataFrame*) for ML inference in end-user analysis
  - Supporting several **ONNX** operators and also **GNN**s
  - A prototype implementation for **GPU** using **SYCL** has been developed
    - plan to extend to **CUDA** and/or **ALPAKA** following some interest by experiments to deploy in their GPU-based trigger system
- Future developments according to user needs and received feedback
  - aim to support the latest production model of experiments (GNN and transformers)
  - models used for fast simulations (GAN and VAE)

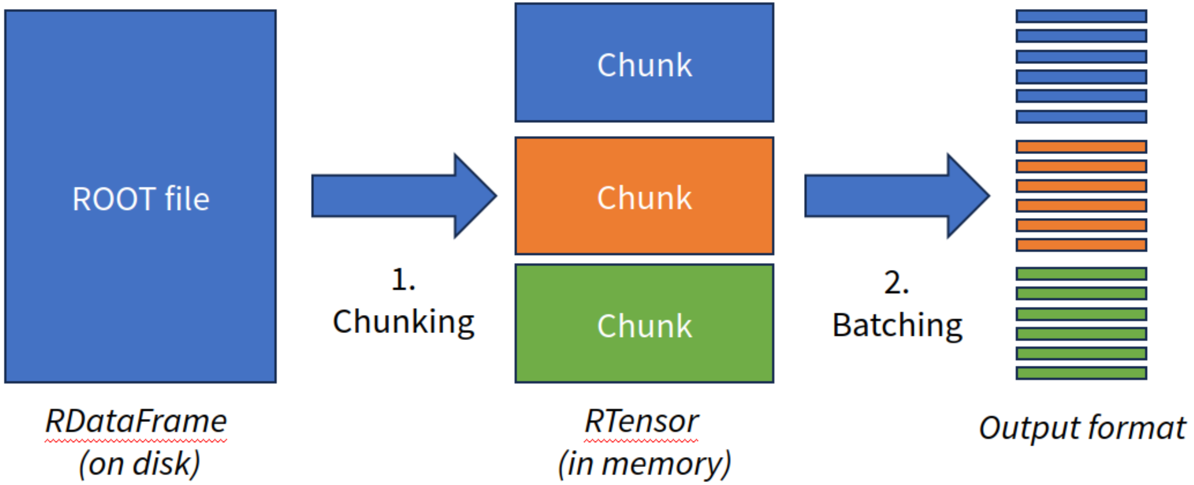
Other ML Activities

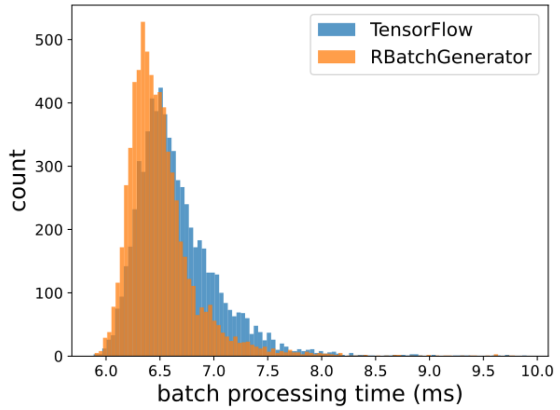
RBatchGenerator: Batching ROOT files

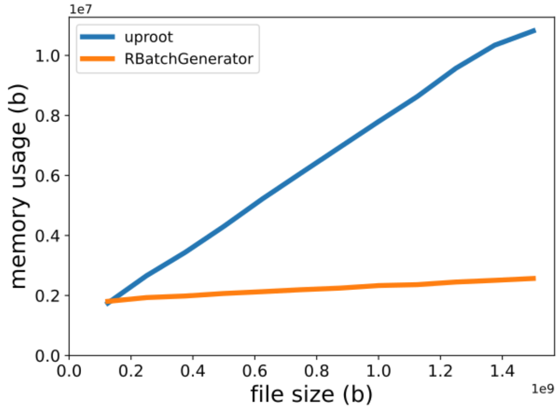
Serving tensors to ML training pipelines (ongoing R&

- **Generate batches directly from a ROOT file**
- As fast as traditional ML software
- Scales to very large file sizes
- Easy to add to workflow

- Working on direct integration with RDataFrame



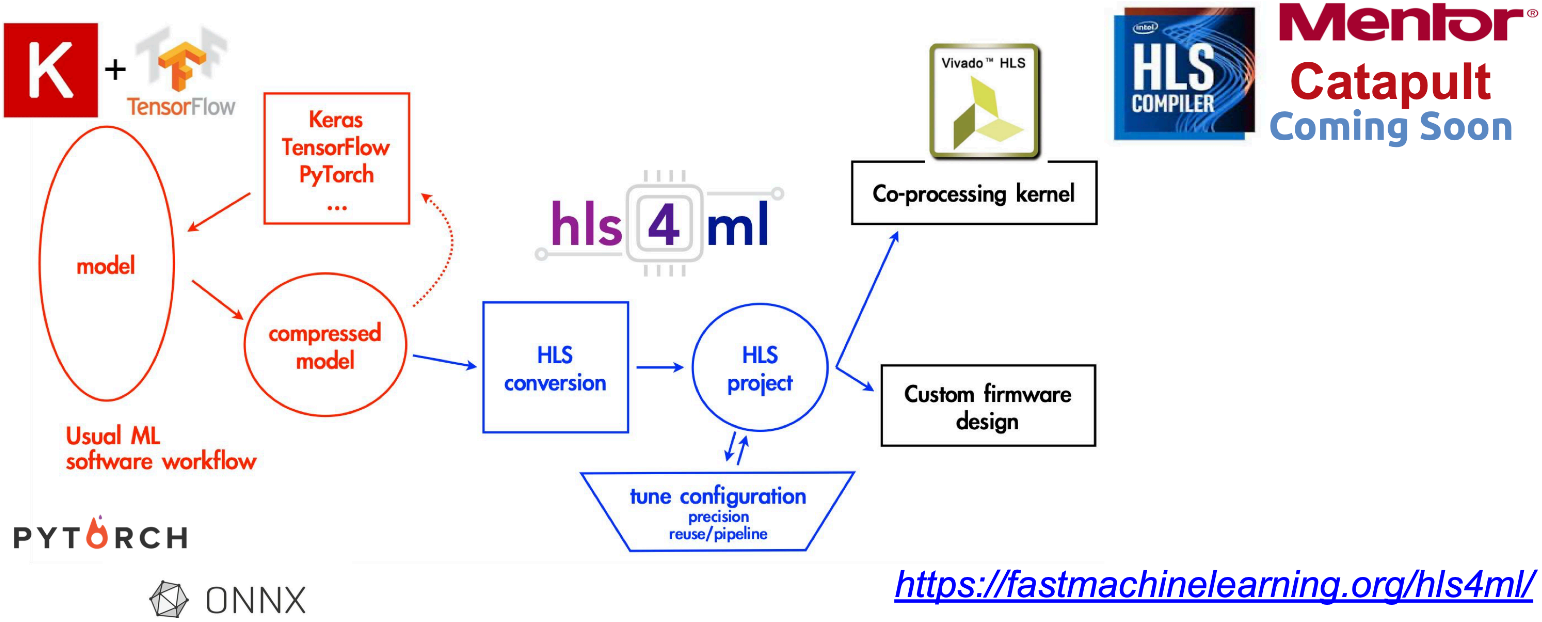




NGT Activities

- SFT is hosting common activities of Next Generation Trigger project
  - Support tools for fast ML in FPGA:
    - **hls4ml** (for DL) and **Conifer** (BDT)
    - develop new functionality to support experiment needs
  - Develop the software infrastructure for **hardware-aware neural network training workflows**.
    - enable the development and deployment of hardware-optimal AI-based real-time algorithms.

high level synthesis for machine learning



https://fastmachinelearning.org/hls4ml/

from V. Lonchar at 24th IEEE Real-Time Conference