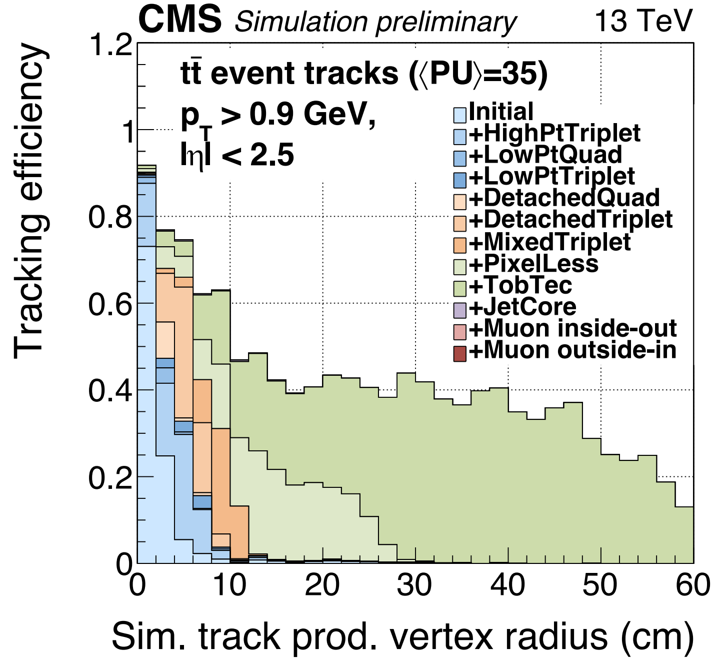
Advanced Methods for Data Processing and Reconstruction

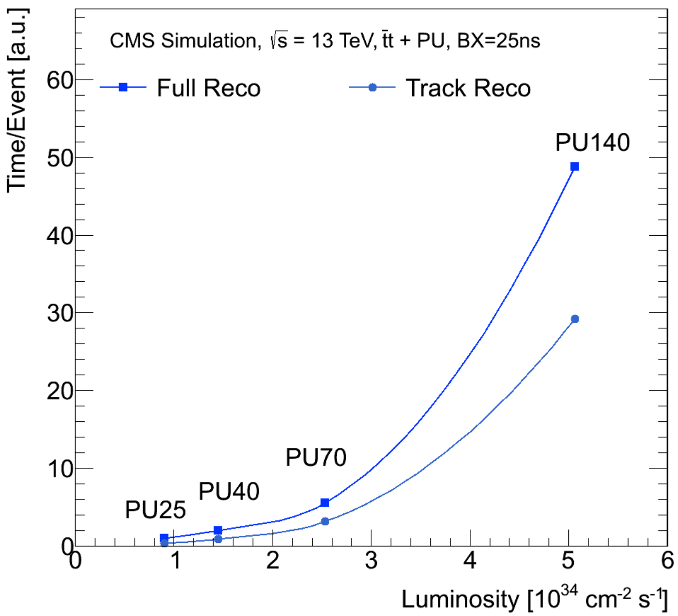
Accelerating Reconstruction on advanced hardware architectures:
Tracking on accelerators
Graph Neural Networks for reconstruction
Accelerating ML inference

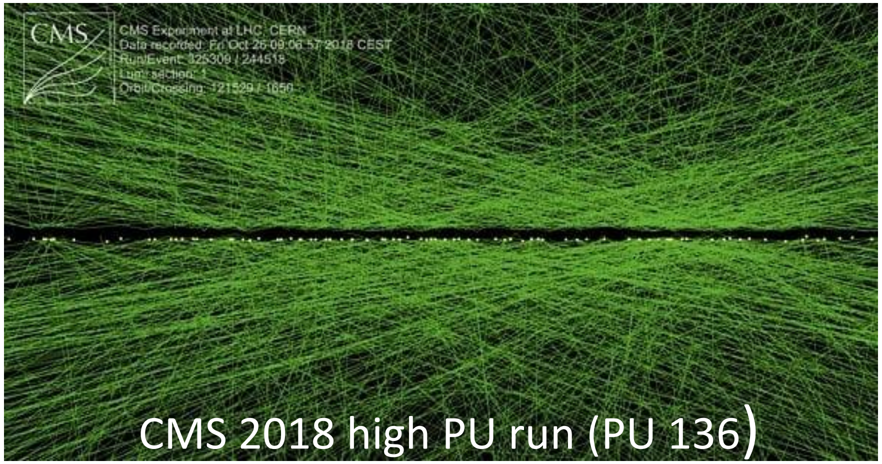
Allison Reinsvold Hall (FNAL), Lindsey Gray (FNAL), Nhan Tran (FNAL)

Charged particle track reconstruction in CMS

- Tracking takes up 58% of offline reconstruction time per event
- Performed using Kalman filter algorithm: well-understood and excellent performance
- Time to reconstruct tracks grows exponentially with pileup



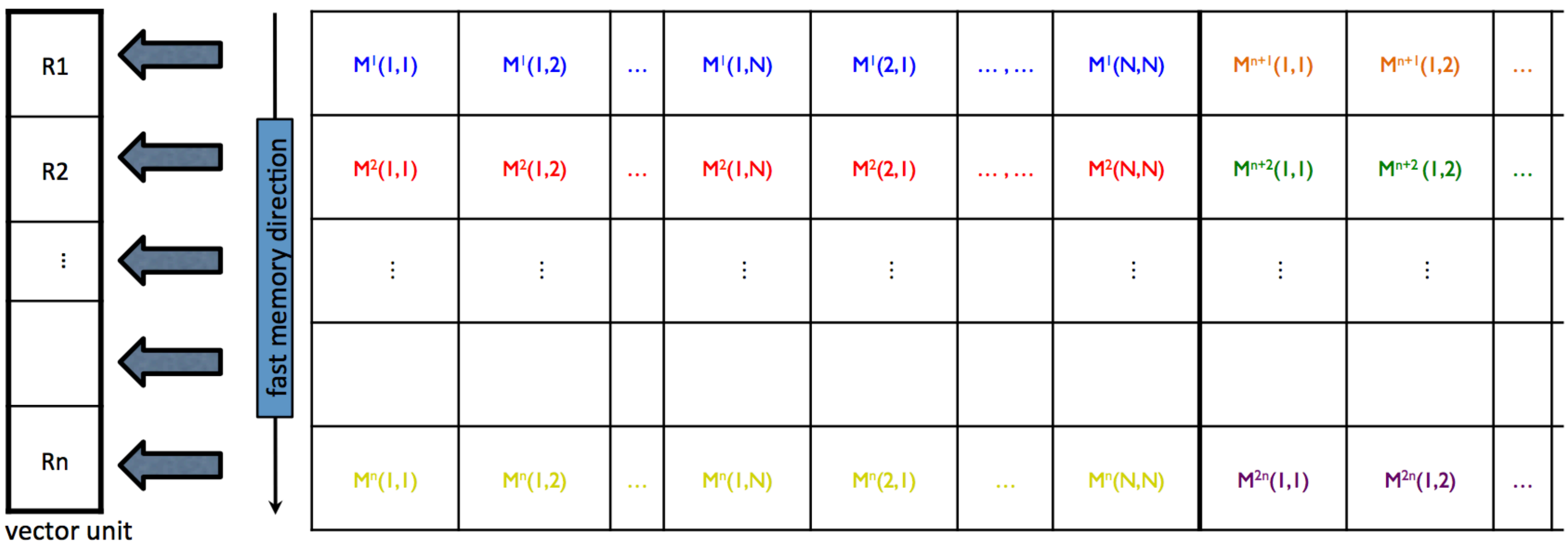




Sci-DAC4: HEP Event Reconstruction with Cutting Edge Computing Architectures

Fermilab, U. of Oregon, UC San Diego, Cornell, Princeton

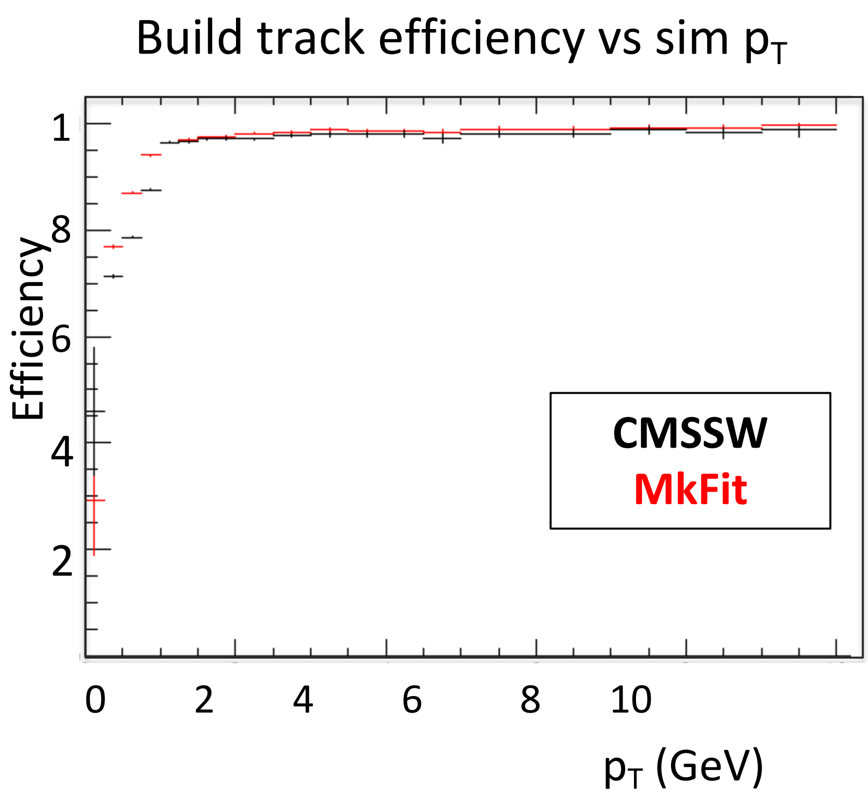
- 3 year SciDAC project to speed up HEP event reconstruction, collaborating with group funded by IRIS-HEP
- Kalman filter is **hard to optimize:** branching required to explore multiple candidates, different numbers of tracks/event and hits/track, requires complex data management and bookkeeping
- Custom “Matriplex” library to efficiently vectorize small matrix operations



Matrix size **NxN**, vector unit size **n**

Physics Results

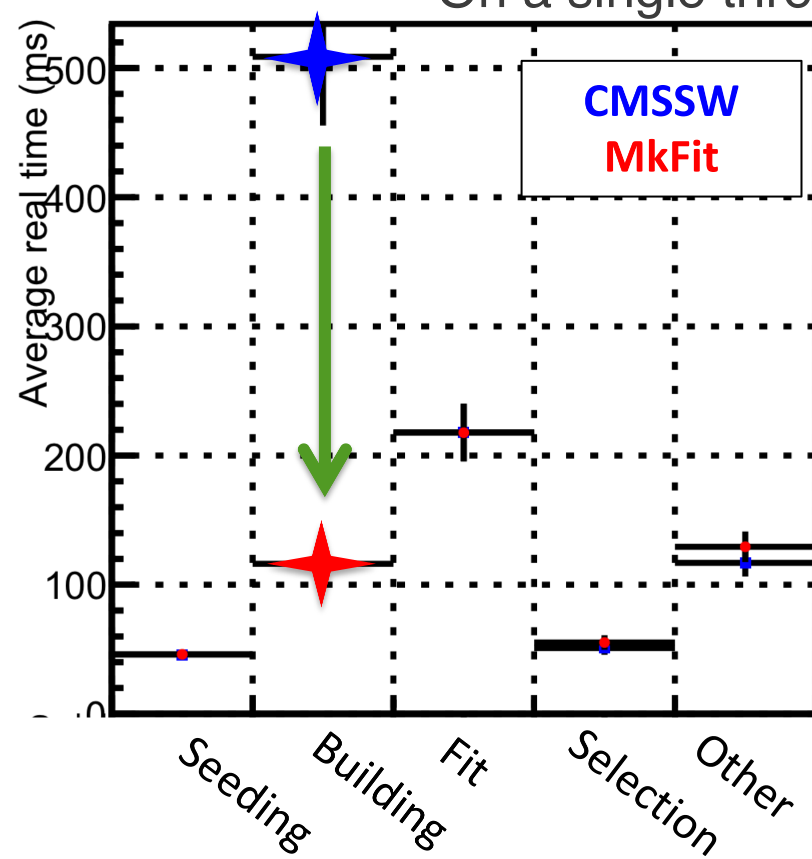
Equal or better track building efficiency than nominal CMSSW



Timing Results

4.3x speedup* compared to CMSSW. **7x speedup** if data conversions are ignored

* On a single thread



Next steps and future work

Exploring two approaches for GPU implementation:

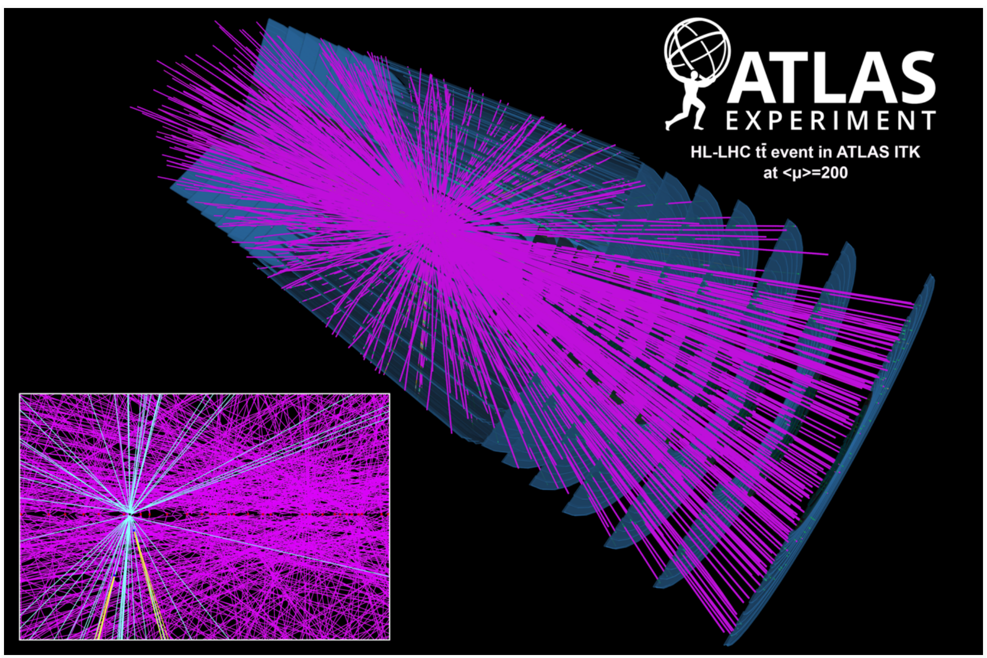
- Option 1: Write algorithm using CUDA
- Option 2: Code portability tools such as OpenACC
  - Collaborating with ORNL and the SciDAC RAPIDS Institute

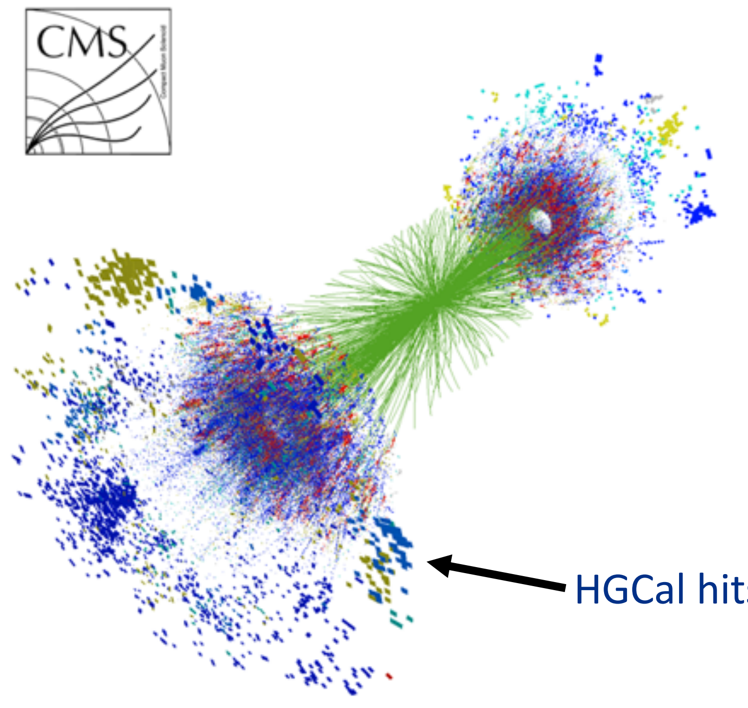
Next steps:

- Continue to improve algorithm’s timing performance
- Finishing optimizing physics performance, particularly for difficult-to-reconstruct tracks such as those with fewer hits
- Integrate algorithm into CMS High Level Trigger and test algorithm online during Run 3 of the LHC

Solving HL-LHC Detector Challenges with ML

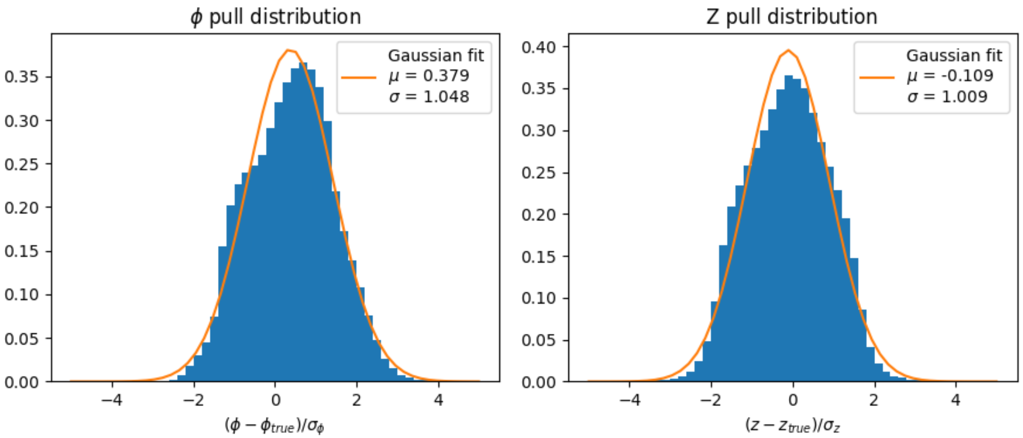
- HL-LHC provides enormous instantaneous luminosity (~1e35/cm2/s)
  - Challenges for radiation tolerance, bandwidth, and pattern recognition
  - Pattern recognition difficult due to many overlapping patterns
  - Particle density & detector segmentation increase ~order of magnitude
  - **need a new arsenal of reconstruction tools**

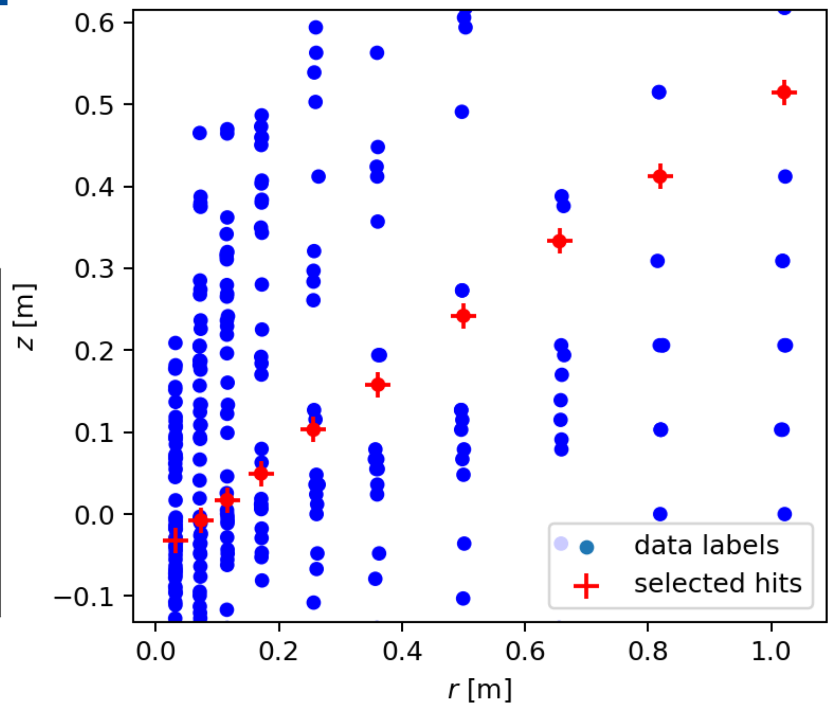




Using ML for Reconstruction

Reconstruction task: associate detector hits into usable physics objects





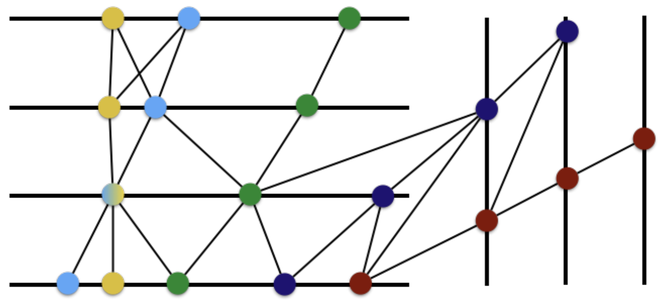
[arXiv:1810.06111](https://arxiv.org/abs/1810.06111)

- Finding an ML algorithm that can perform a reconstruction task is not straightforward
  - Fully connected networks, CNNs not well adapted to irregular detector geometries (gaps, cracks, etc…)
    - Spend valuable resources encoding ‘dead’ space or otherwise impertinent information
  - The ‘representation’ of the detector is hidden from these networks because of their strange geometries
  - Networks still function well but could be improved

Graph Neural Networks in Tracking

GNNs only care about data received and associations, directly exposing representations

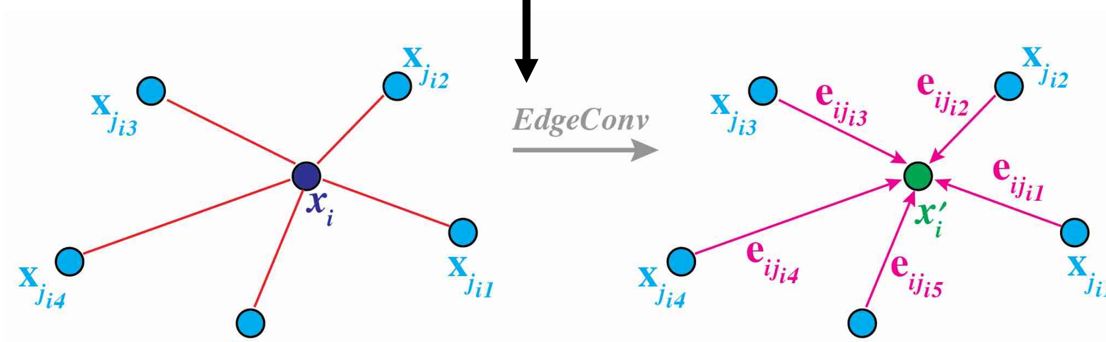
hits in a tracker



O(Nlog(N))

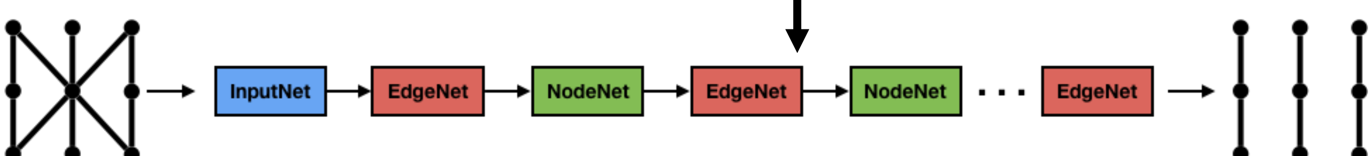
order of magnitude smaller network than previous slide

one network type usable on variety of detectors



arXiv:1801.07829

O(N)



initial guess

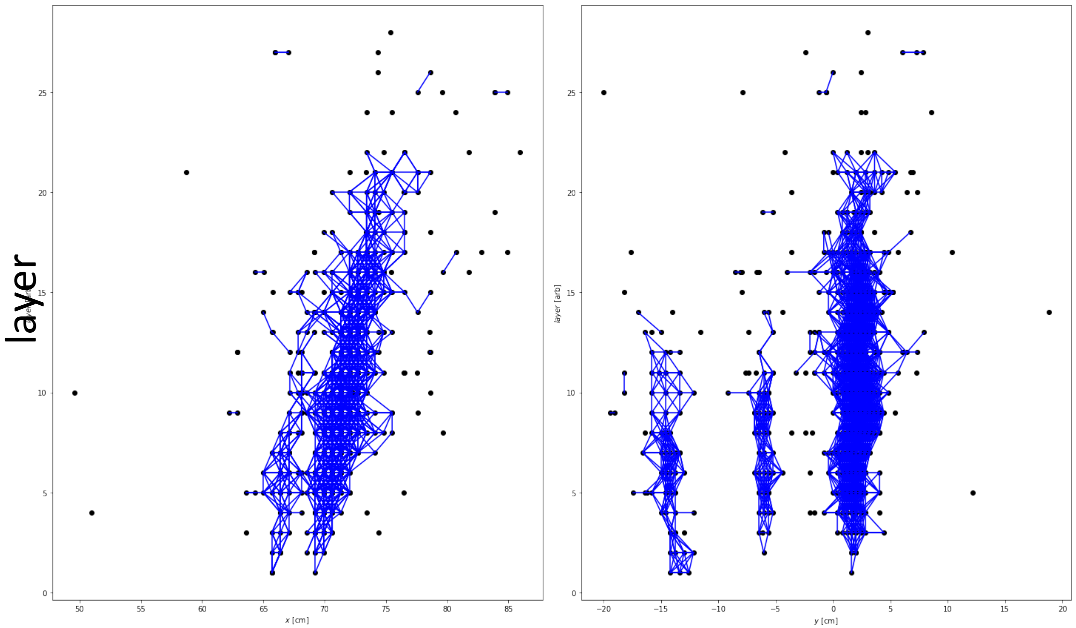
arXiv:1810.06111

final associations

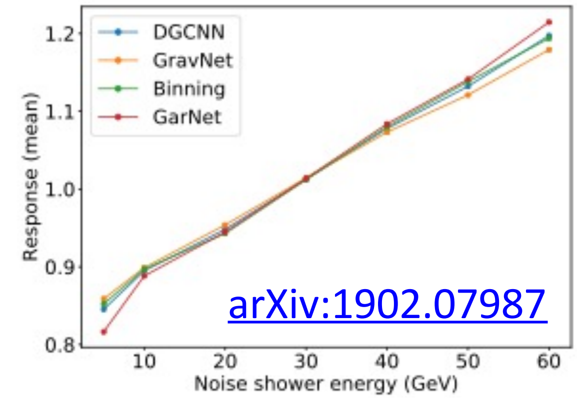
O(log*(N))

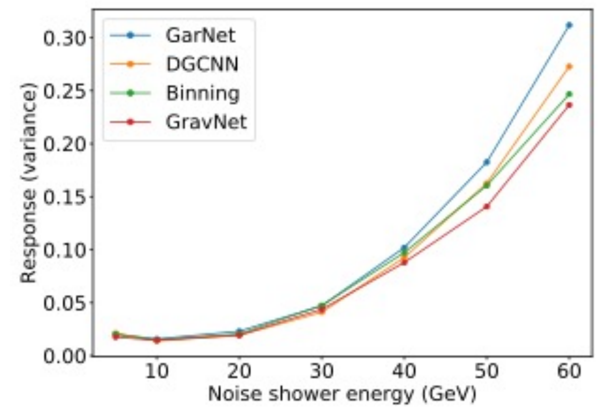
HL-LHC Calorimetry and Particle Flow

example of cluster graphs in HGCal



LDRD pilot work





- Graph-nets can be extended for use in calorimetry straightforwardly (same as tracking)
  - Same toolkit of fast algorithms can be used to build clusters from network outputs
  - Performance outclasses current human made algorithm for HGCal
- Particle Flow is also a graph segmentation task, next target after calorimetry
  - Associate tracks and calorimeter clusters best representation of collider event

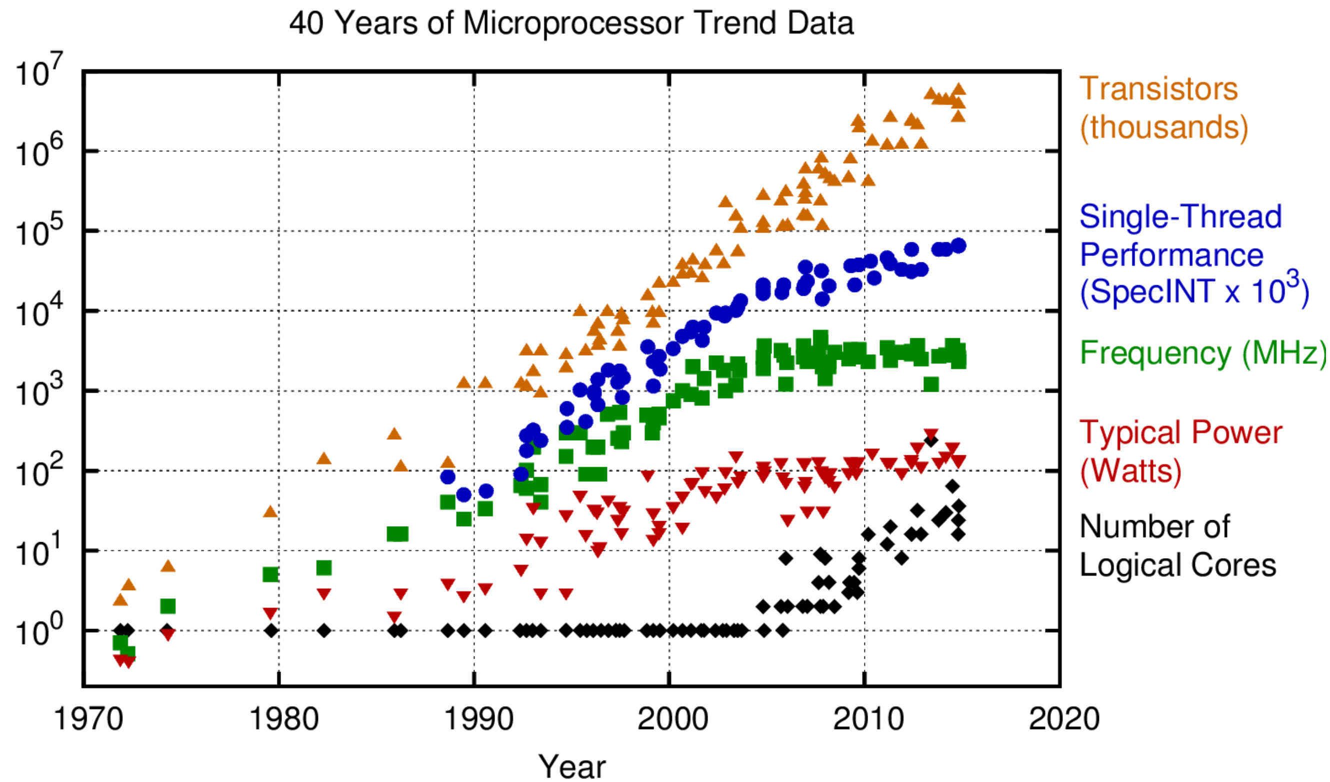
Next steps and future work

- Graph neural networks provide a powerful new toolkit for reconstruction
  - Same network architectures can be applied in tracking, calorimetry, higher-level event reconstruction
  - Combined with appropriate efficient algorithms to post-process the data, much faster than typical task-specific algorithms
  - Cross cutting through detector types and frontiers genuinely possible
- Next challenge is to make these tools available in experiment computing environments
  - Develop networks, integrate tools, accelerate inference
  - Target offline computing, software trigger, and hardware triggers to integrate graph networks and bring these powerful new algorithms to bear in every aspect of experiments

ML inference on heterogeneous computing architectures

Moore's Law falling off
…but Dennard Scaling ended in 2010

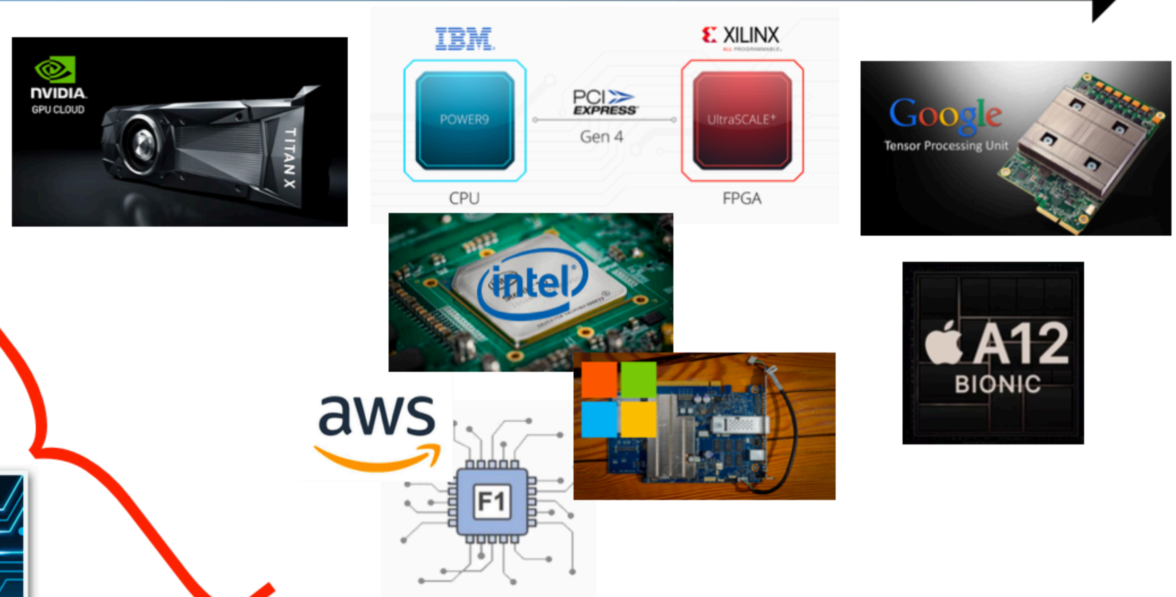
Single threaded performance not improving
Circa ~2005: "The Era of Multicore"



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

→ Today: Transition to the "Era of Specialization"? (c.f. Doug Burger)

Heterogeneous Computing

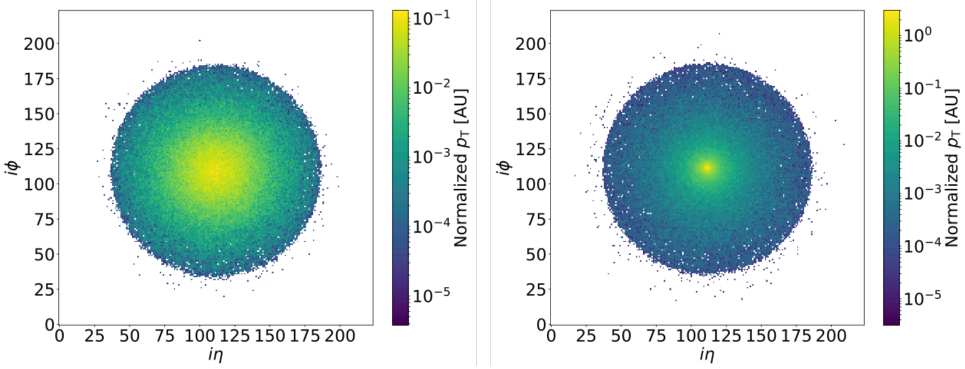


Advances in heterogeneous computing driven by **machine learning!**

BIG machine learning in physics

Open top quark dataset with ResNet50

https://arxiv.org/abs/1904.08986

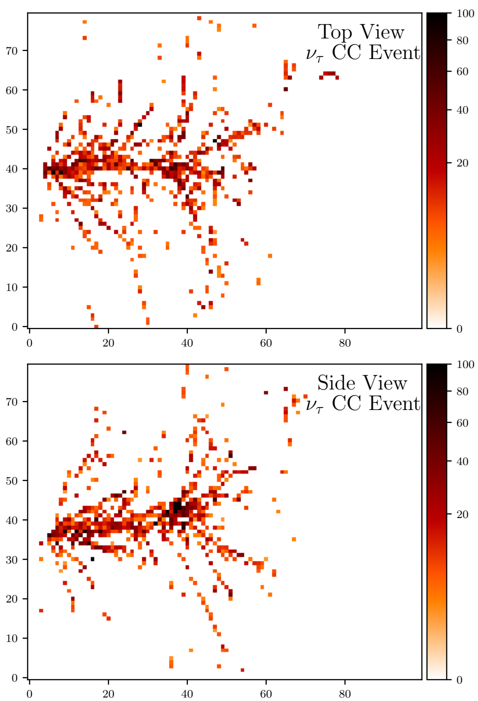


HEP.TrkX

Tracking and clustering with Graph NNs

https://arxiv.org/abs/1810.06111

https://arxiv.org/abs/1902.07987

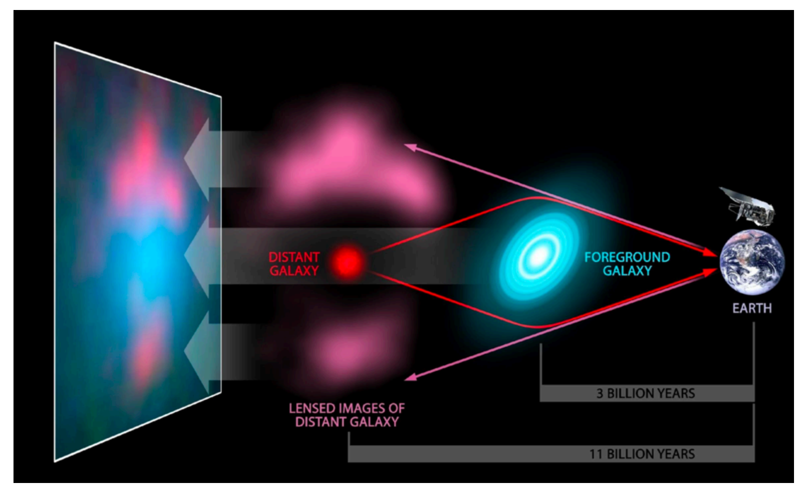


Nova event classification with CNNs

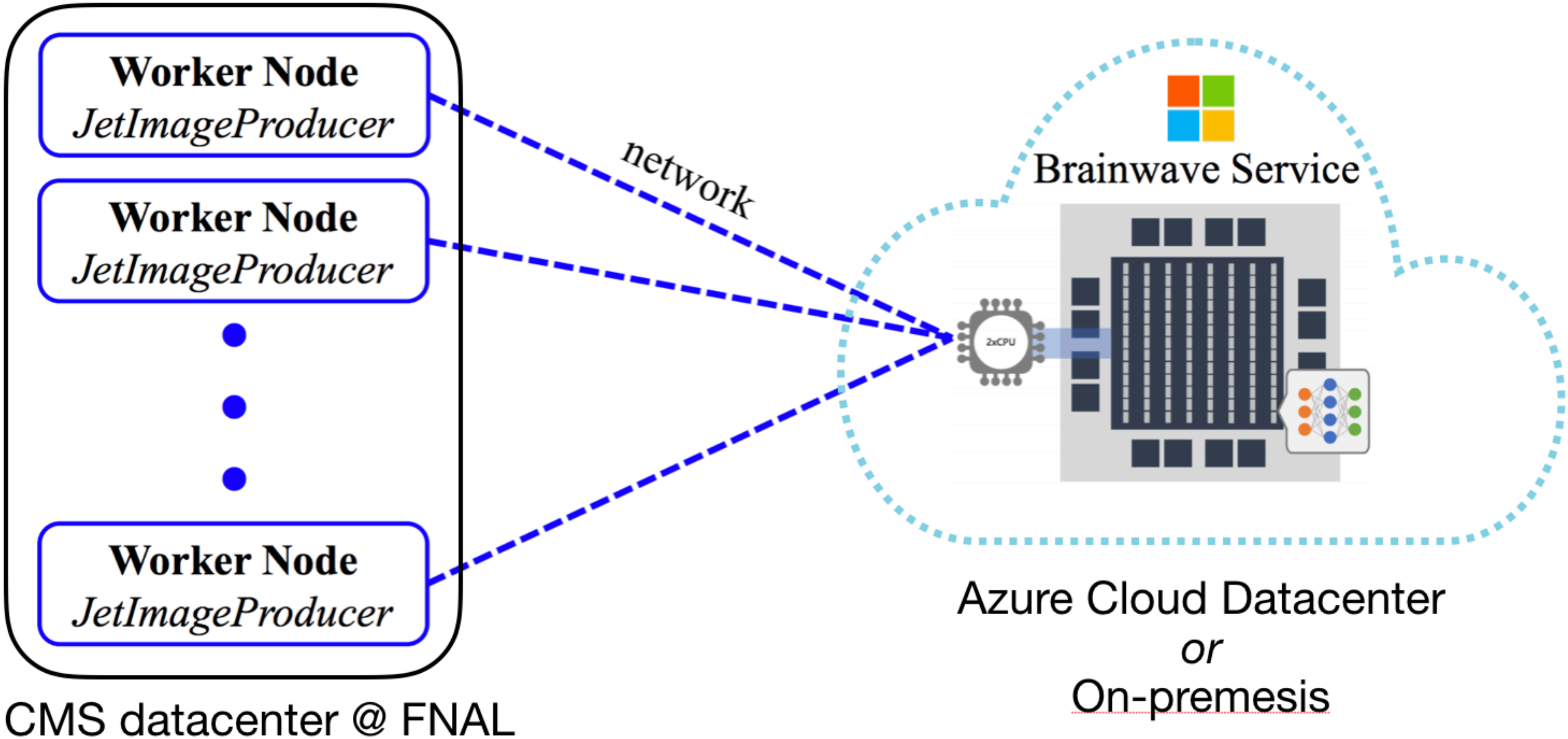
https://arxiv.org/abs/1604.01444

DES lensing with CNNs

https://arxiv.org/abs/1810.01483



Accelerated ML as a Service



Non-disruptive integration of heterogenous computing resources into the HEP computing model

Deploy as a service (many CPUs to few FPGAs) is much **more cost-effective**

Proof-of-concept study

https://arxiv.org/abs/1904.08986

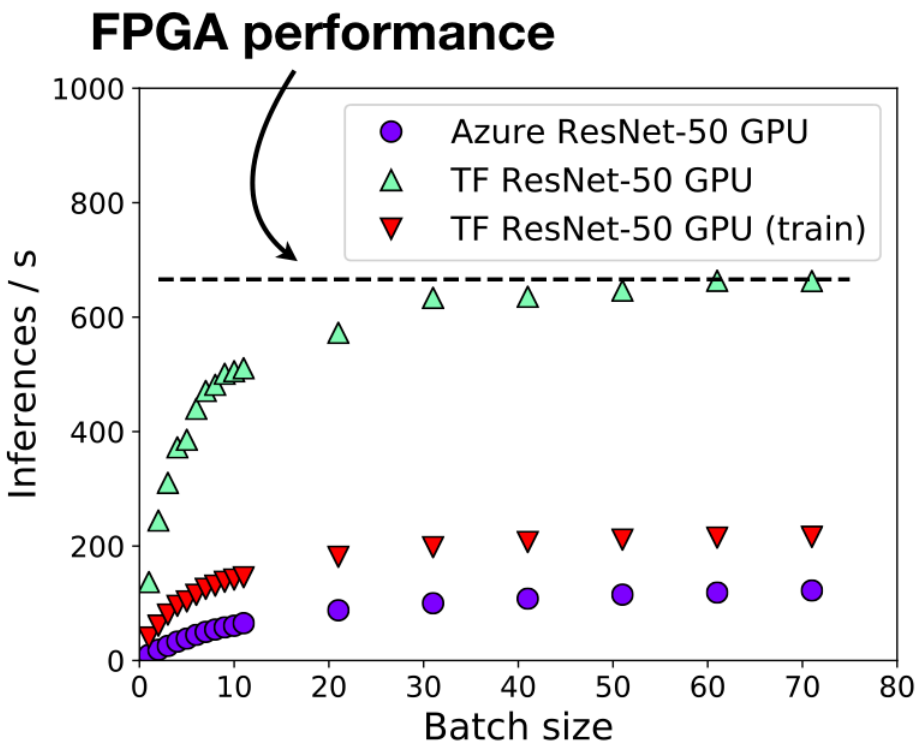
Integrate Microsoft Azure ML acceleration with Intel FPGAs into CMSSW

ResNet50 for top tagging at LHC and event classification at Nova

Measured latency of Azure ML as a service to be **30 (175) times faster** than inference in CPUs with CMSSW

Includes round trip time

Multi-threaded non-blocking CMSSW feature `ExternalWork`



Throughput competitive with locally-connected GPU with large batch size

FPGA running with batch-of-1

Proof-of-concept paper

https://arxiv.org/abs/1904.08986

FPGA-accelerated machine learning inference as a service for particle physics computing

Javier Duarte · Philip Harris · Scott Hauck · Burt Holzman · Shih-Chieh Hsu · Sergo Jindariani · Suffian Khan · Benjamin Kreis · Brian Lee · Mia Liu · Vladimir Lončar · Jennifer Ngadiuba · Kevin Pedro · Brandon Perez · Maurizio Pierini · Dylan Rankin · Nhan Tran · Matthew Trahms · Aristeidis Tsaris · Colin Versteeg · Ted W. Way · Dustin Werran · Zhenbin Wu

Collaborations and expertise growing:
CMS, ATLAS, Nova, DUNE, Industry

Special thanks for seed funding support:
US-CMS ops
FNAL LDRD

Summary

- Offline reconstruction is projected to dominate processing needs in HL-LHC

- Tracking largest competitor: mkFit project made significant process in vectorizing and speeding up pattern recognition
  - On the way to vectorized implementation of Kalman Filter on GPUs and other advanced architectures

- Machine Learning excellent candidate to speed up reconstruction by revolutionizing approach
  - Graph Neural Networks used for calorimetry and particle flow

- Processing needs for inference of large networks not small
  - Accelerated inference on FPGAs, run as a service, investigated to speed up reconstruction