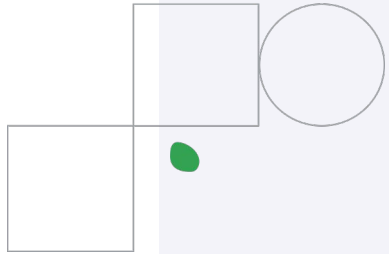
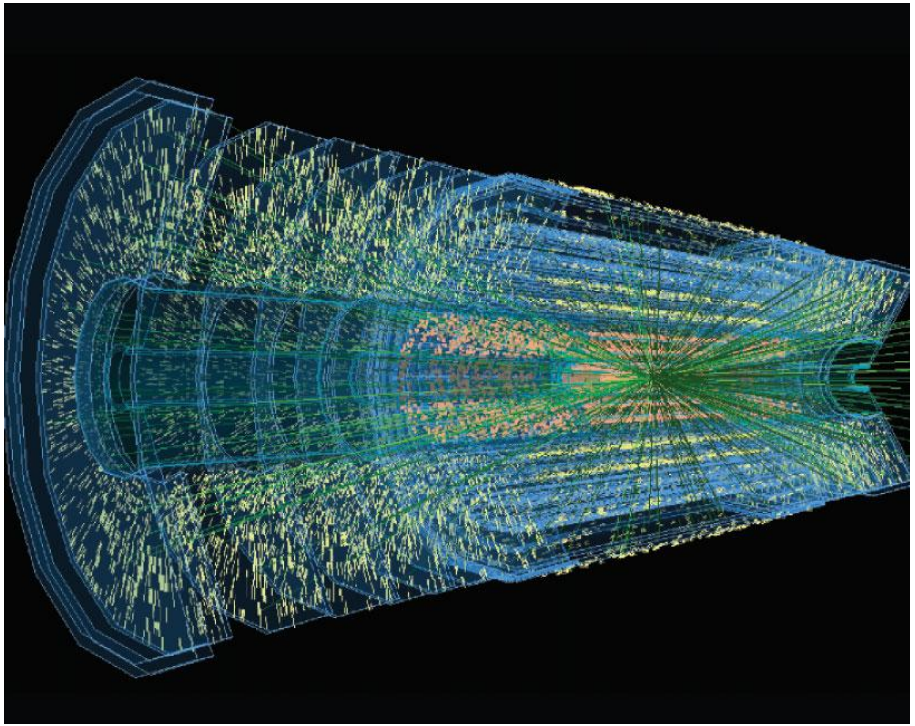
TPU vs. GPU for GNN training

Xiangyang Ju

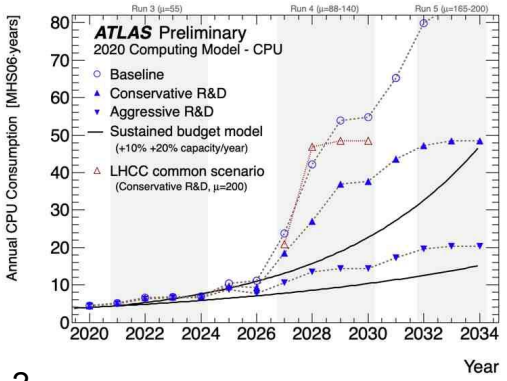
Lawrence Berkeley National Lab



Tracking at High-Luminosity LHC



- Each proton-proton collision contains ~10k tracks left by charged particles
- Each track on average has ~10 space points recorded by the detector
- The combinatorial complex of current track reconstruction algorithm grows quadratically as the number of collisions grows.
- New algorithm is needed.



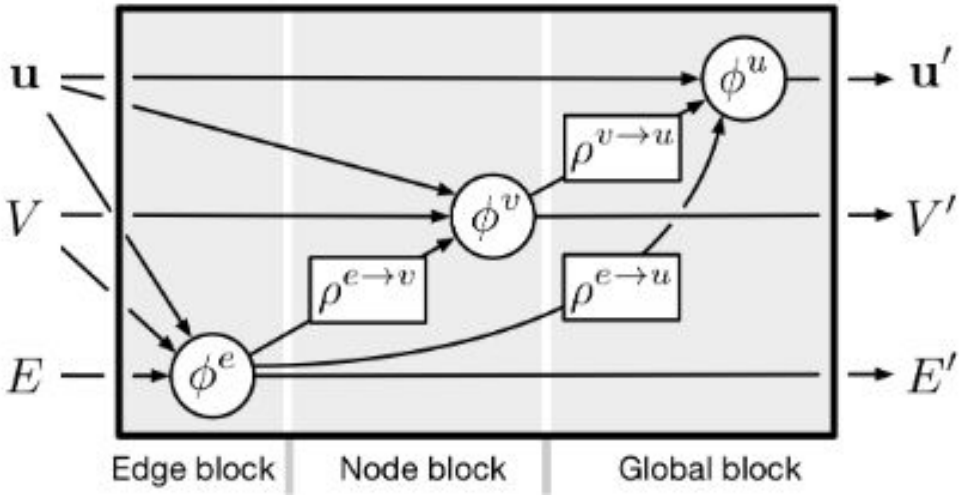
A more technical review of the GNN

arXiv:1806.01261

Graph contains nodes and edges, and node-, edge- and global-level attributes.

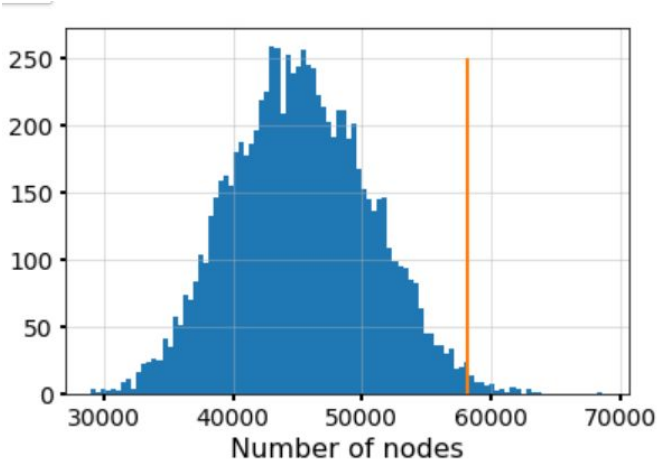
GNN are trainable functions operating on a graph.

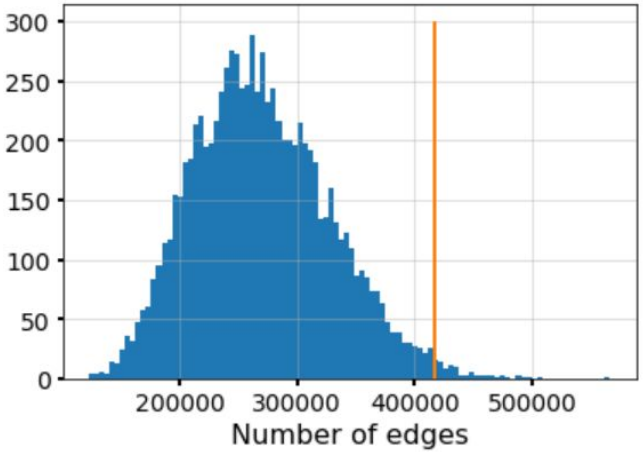
Those functions are neural networks.



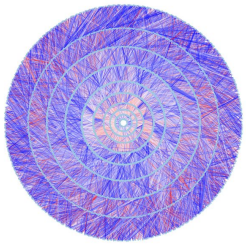
Graph size

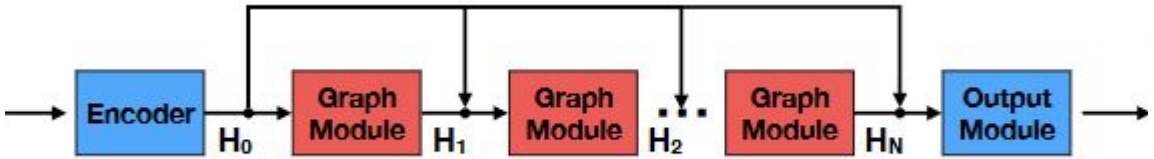
On average: 45,000 nodes and 250,000 edges.

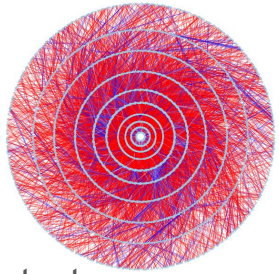


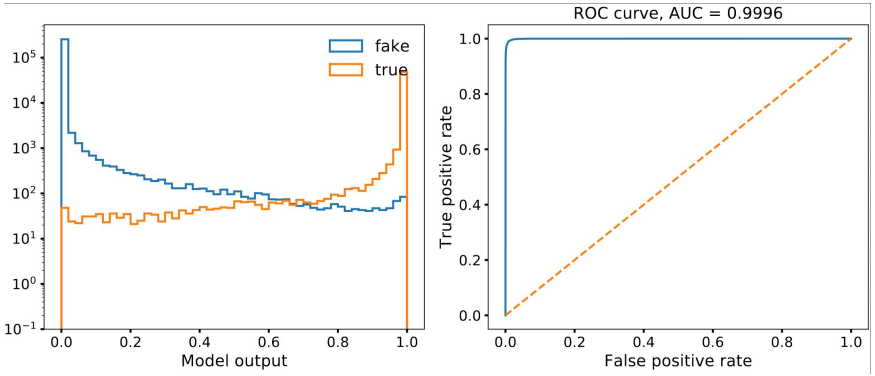


GNN-based solution









Current study is based on a simplified detector geometry.

One epoch containing ~7800 events for training

With a threshold of 0.5, it achieves a precision of 97.5% and a recall of 98.6%.

AI accelerators in this study

- GPU V100 at NERSC Cori, each node has 40 skylake CPUs and 8 V100
- GPU A100 at google cloud
- TPU: us-central1-a, TPU-v3-8 and TPU-v2-32

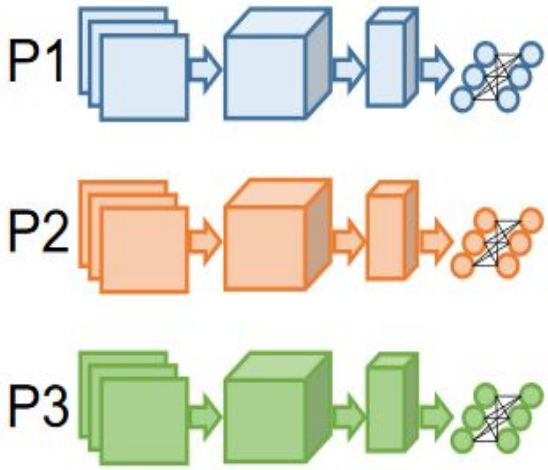
Device	Accelerator architecture	# of chips	Peak Flops [TFLOPS]	High-Bandwidth memory [GiB]	Price with 1 year commitment [$/hour]	Thermal design power [W]
GPU	Nvidia V100	1	14 (fp32)	16	1.56	250
GPU	Nvidia A100	1	19.5 (fp32)	40	N/A	250
TPU-v2-32	TPU v2	32	180*4=720	8*32=256	15.33	75*32=2400
TPU-v3-8	TPU v3	8	420	16*8=128	8	75*8=600

arXiv:1907.10701 Google Cloud TPU Nvidia V100 datasheet In-Datacenter Analysis for TPU

Distributed training strategy

Performing data parallel distributed training:

Same model is replicated to different devices (GPUs, TPUs), different data are sent to devices for training, gradients are averaged among devices to update the weights

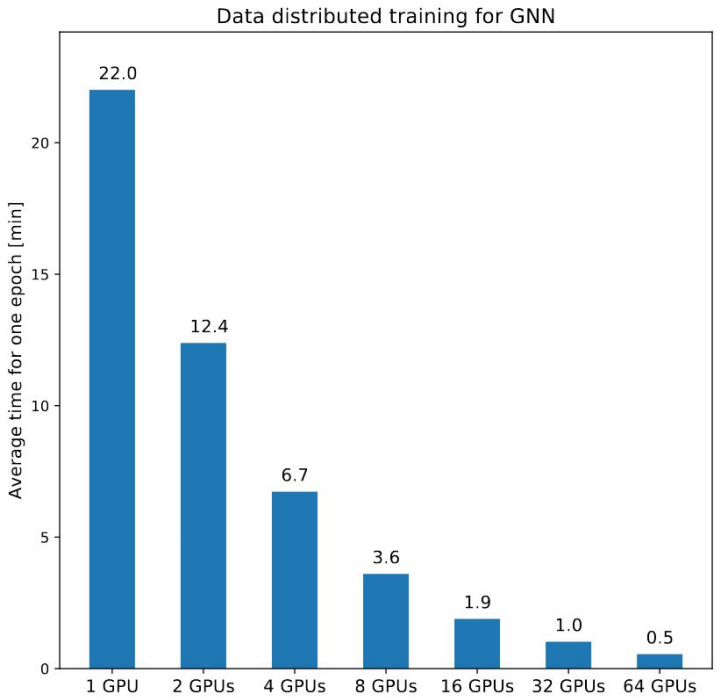


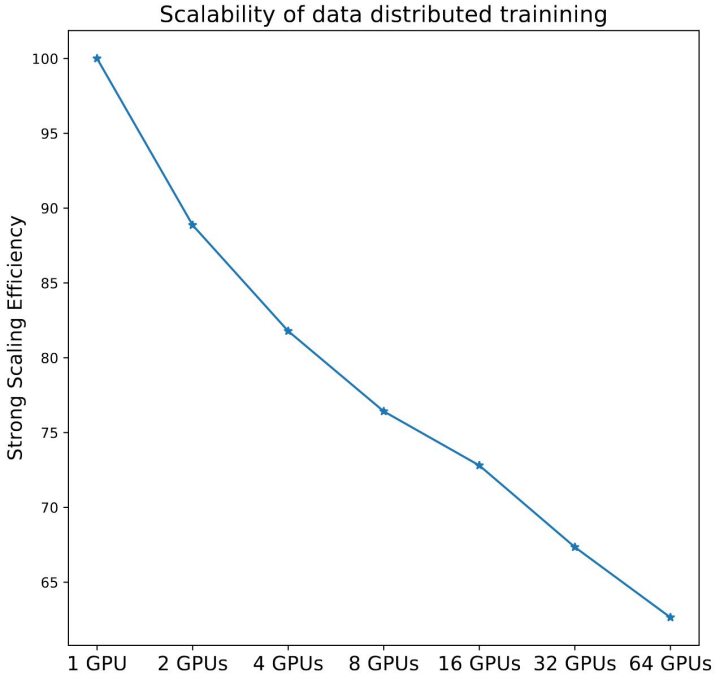
arxiv:1802.09941

Two implementations:

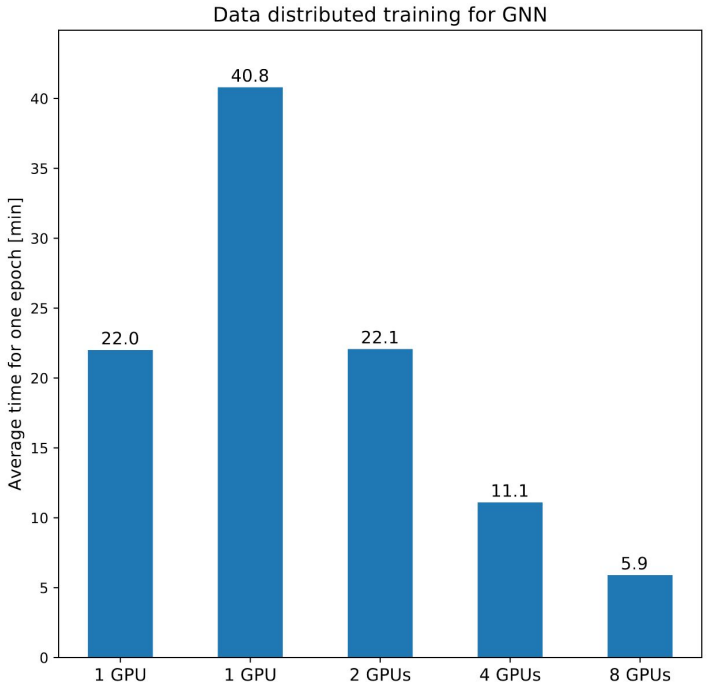
1. Horovod
   a. Good: MPI-based, HPC friendly
   b. Bad: not work for TPU, need extra coding
2. Distributed strategy in TensorFlow
   a. Good: same code runs on CPU, GPU, TPU. even IPU?
   b. Bad: need same graphs size, cannot across nodes

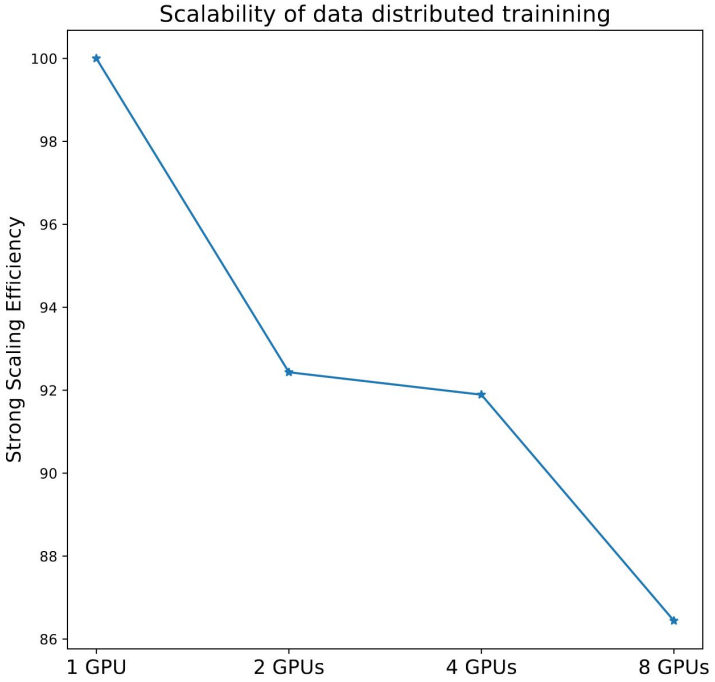
Distributed training for GPUs, with Horovod





Distributed training for GPUs, with TF distributed strategy

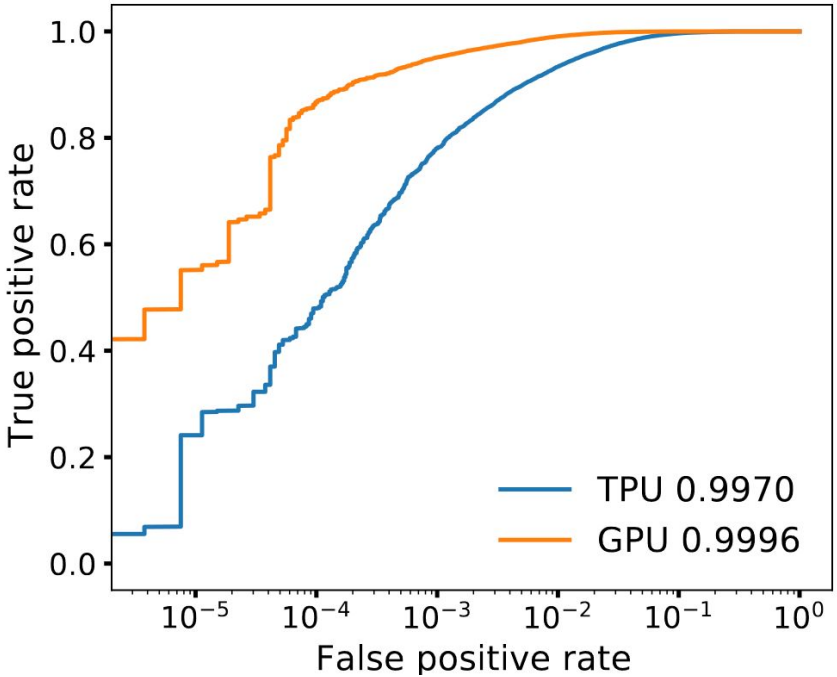




Key metrics for compare TPUs with GPUs

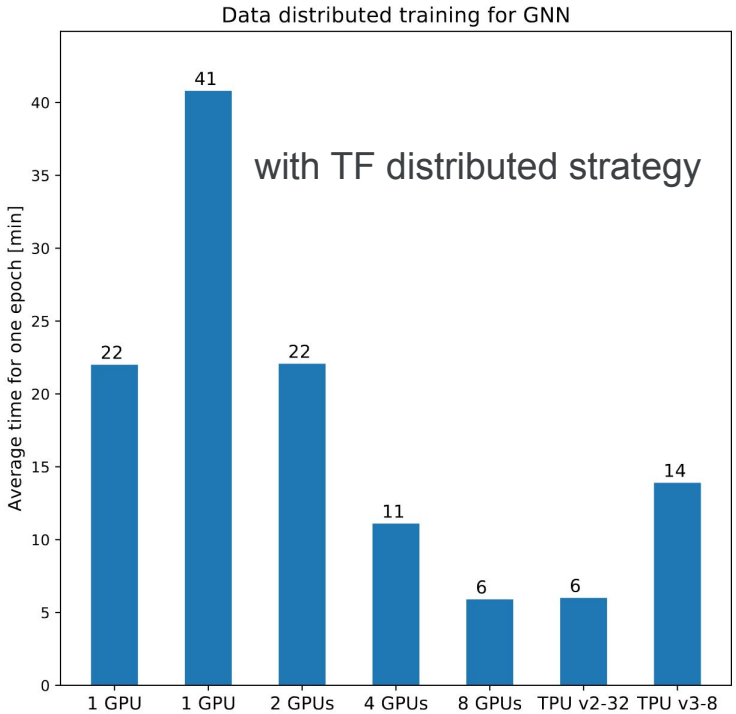
1. Accuracy → precision and recall on testing data
2. Latency → time it takes to finish training for one epoch
3. Cost → dollars per epoch
4. Heat dissipation → energy cost per epoch. = thermal design watt times the time it takes to finish one epoch, assuming device 100% busy during the training,

Accuracy



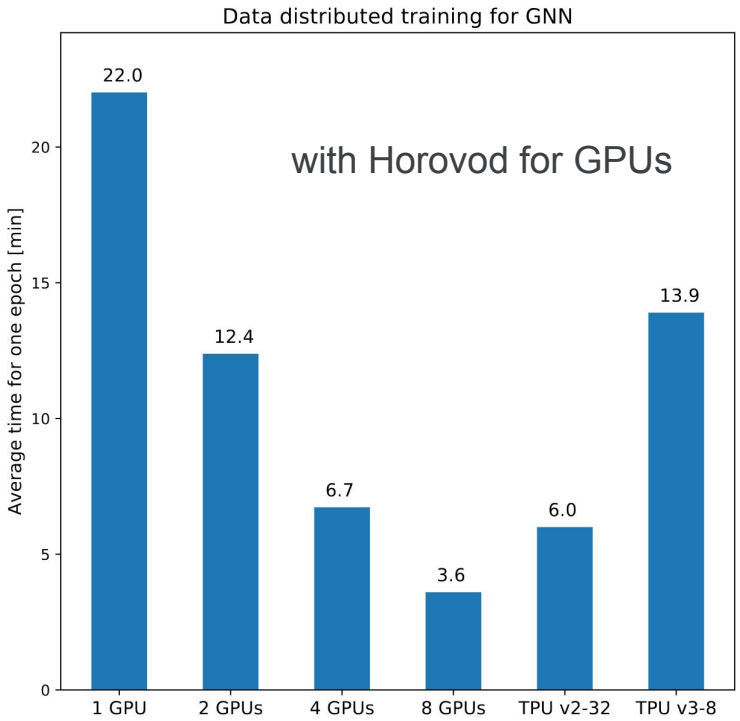
1. Hyperparameters of the model when trained in GPU are tuned to have good performance. The learning rate is found particularly important.
2. No detailed hyperparameter tuning is done for TPU

Latency



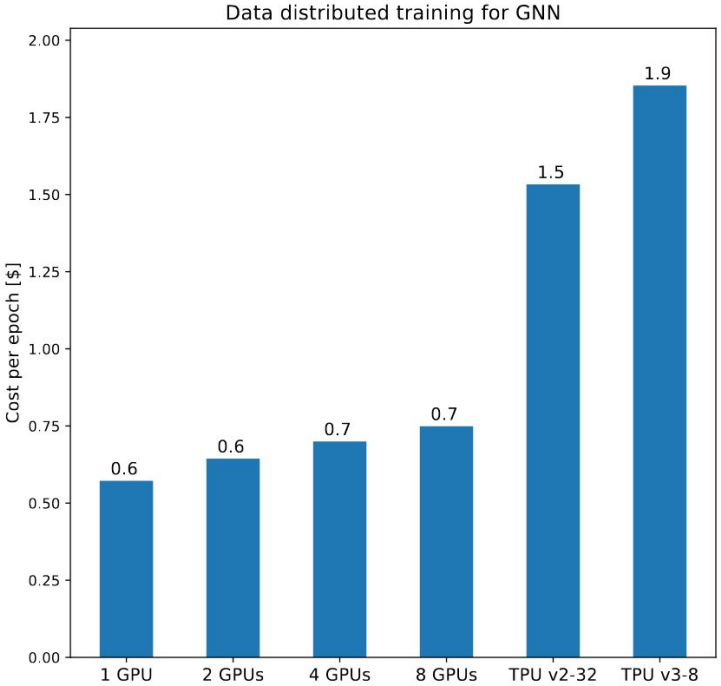
1. Padding graphs to the same size increases the training time by a factor of 2
2. TPU v2-32 equals 8 GPUs and TPU v3-8 is better than 2 GPUs, worse than 4 GPUs

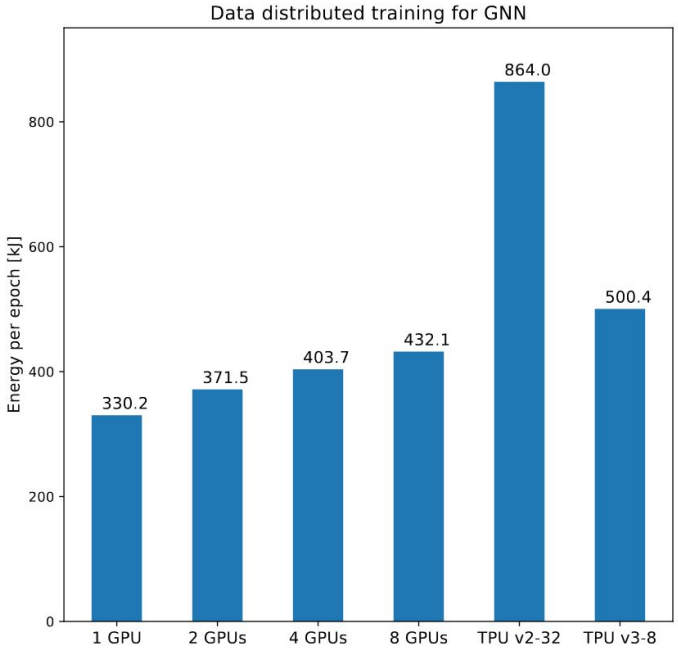
Latency



1. No padding required in Horovd,
2. TPU v2-32 equals ~4 GPUs and TPU v3-8 is better than 1 GPUs, worse than 2 GPUs

Cost and heat dissipation





Summary

Device	# of devices	Latency [minutes]	Cost [$]	Heat dissipation [kJ]
GPU V100	1	22.0	0.6	330
	2	12.4	0.6	371
	4	6.7	0.7	403
	8	3.6	0.7	432
TPU v2	32	6.0	1.5	864
TPU v3	8	13.9	1.9	500

Profiling TPU v3-8 and GPU V100

TPU v3-8

Performance Summary

Average Step Time — 823.2 ms
lower is better
(σ = 87.5 ms)
- Idle: 112.19 ms
- Input: 0.39 ms
- Compute: 710.60 ms

Host Idle Time — 98.2%
lower is better

TPU Idle Time — 13.6%
lower is better

FLOPS Utilization
(higher is better, why two numbers?)
- Utilization of TPU Matrix Units: 2.3%
- Compared to Program's Optimal FLOPS: 28.1%

Memory Bandwidth Utilization — 28.1%
higher is better

Run Environment

Number of Hosts used: 1

Device type: TPU v3

Number of device cores: 8 (Replica count = 1, num cores per replica = 1)

GPU v100

Performance Summary

Average Step Time — 169.2 ms
lower is better
(σ = 31.5 ms)

TF Op Placement
- Host: 13.9%
- Device: 86.1%

Op Time Spent on Eager Execution
lower is better
- Host: 0.1%
- Device: 0.0%

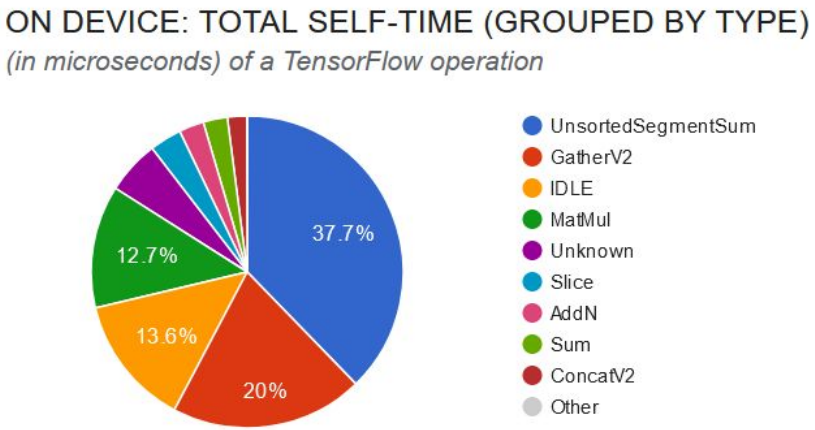
Device Compute Precisions
out of Total Device Time
- 16-bit: 0.0%
- 32-bit: 100.0%

GPU idle time 10%.

FLOPS Utilization: 30% (fp32 only)

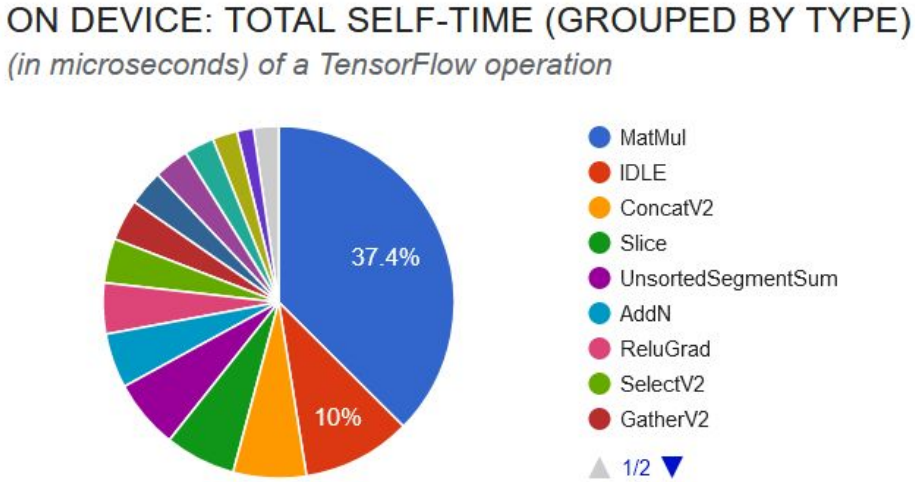
Profiling [continued]

TPU v3-8



Most time spent in aggregating information between nodes and edges

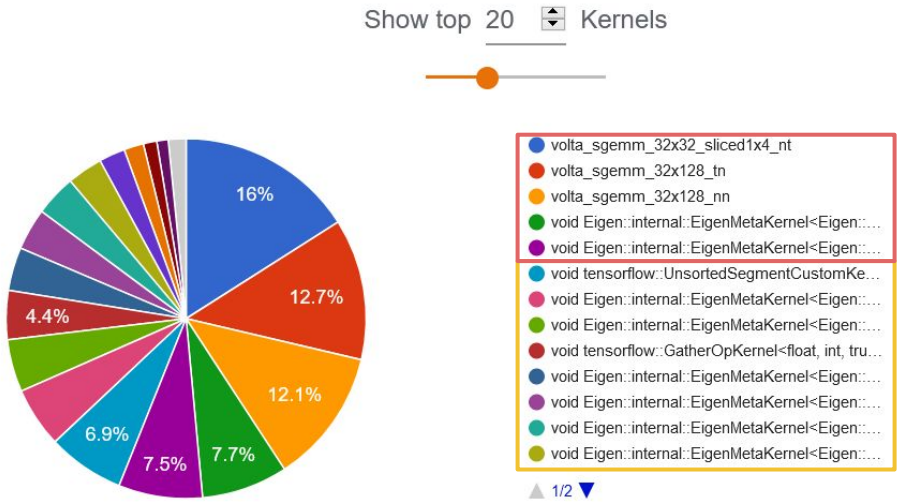
GPU v100



Most time spent in matrix multiplication

Profiling GPU kernels

Top 20 Kernels with highest Total Duration

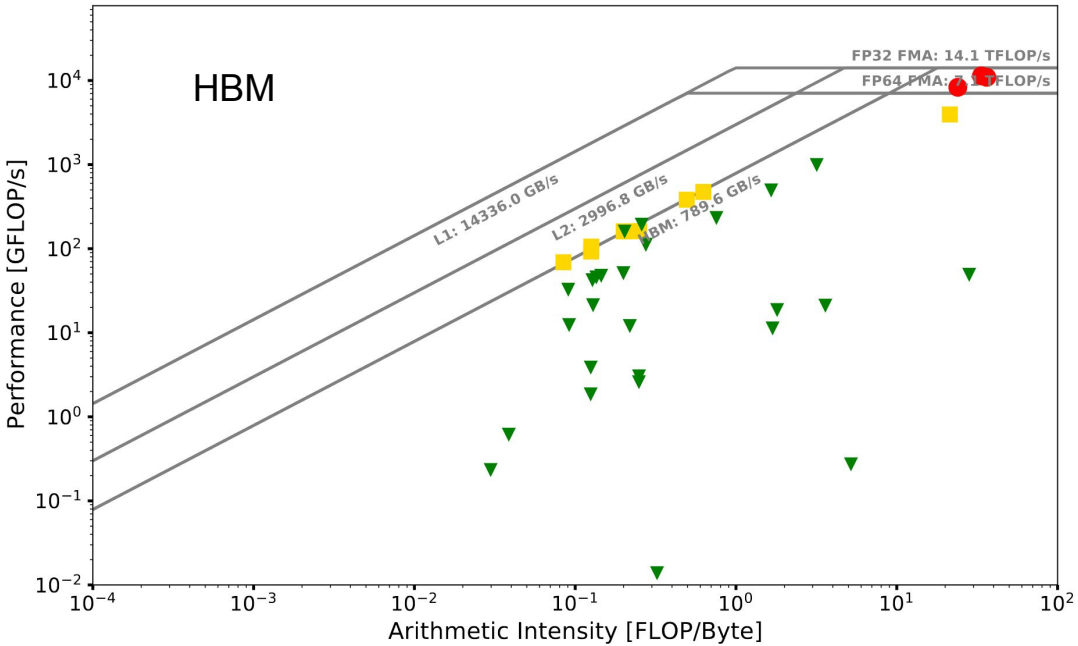


Top 5 kernels are mostly matrix multiplications and sweepers, taking **66%** of total computing time.

Top 5 to 20 kernels are led by the message passing operation: UnsortedSegment(sum)

Analyze with roofline model

With kind help from Yunsong Wang

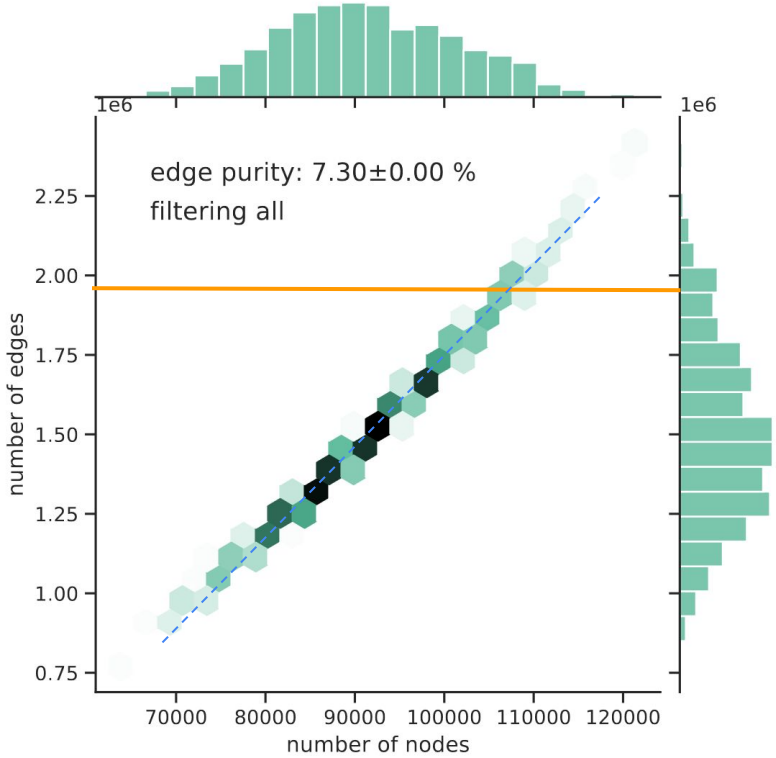


Red: the top 5 kernels
Yellow: the top 5 to 20 kernels
Green: the rest

Message passing Ops are limited by bandwidth

Profiling results for L1/L2 and overall are in backup.

GNN for High-Luminosity LHC



On average the number of nodes increases from 45k to 90k, the number of edges increases from 250k to 1500k.

Using 3300 training events, each epoch takes about 30 minutes. It would not be completely unreasonable to have 10k training events, in that case, it would take 1 hour to train one epoch.

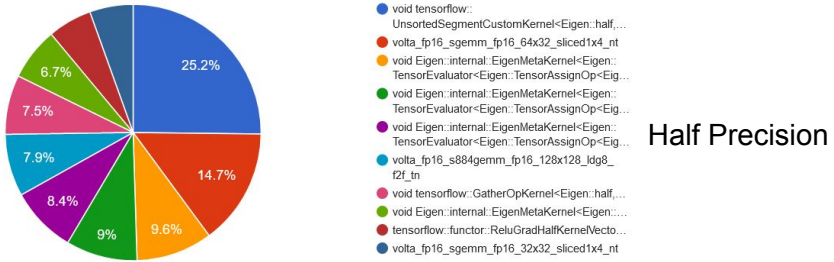
The memory consumption reaches the limitation of A100.

Future promising studies

Mixed precision looks promising in reducing the computing time and memory consumption. The following results on mixed precision represent an ideal scenario, which in practice do not work yet.

	Tensor dtype	time per event [ms]	Memory usage [GiB]
Single precision	Float 32	169	9.81
Half precision	Float 16	120	4.9

Single Precision



Half Precision

NVIDIA Tensor Core in V100 only supports half precision computations, however, it carries ~80% of total computing capability.

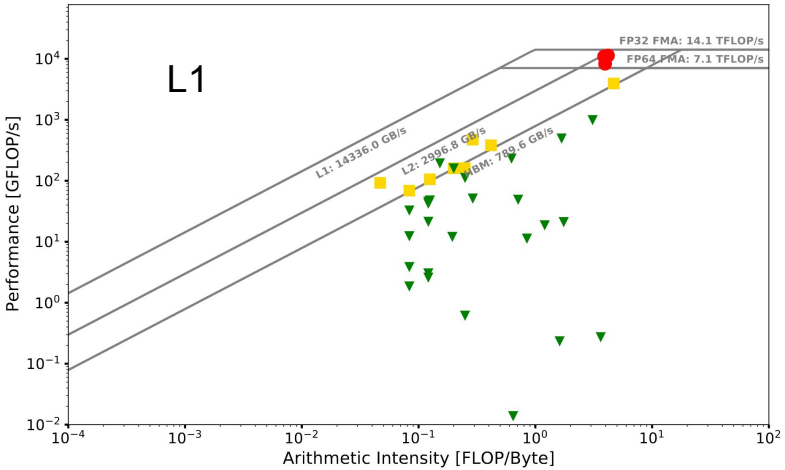
Summary

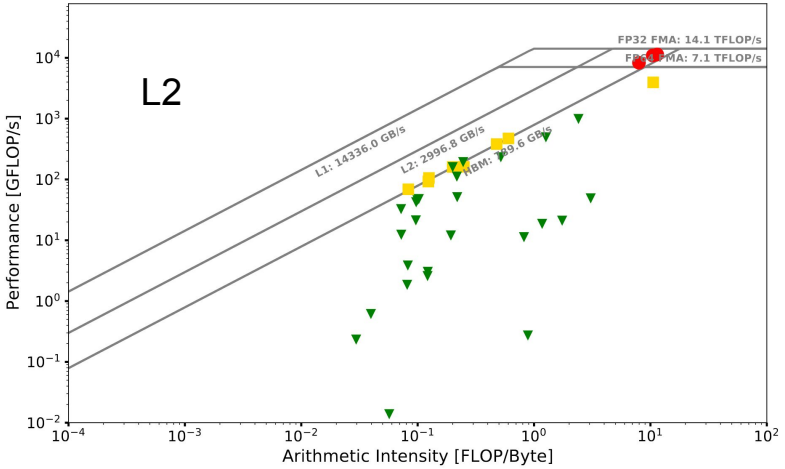
- Graph Neural Networks are a powerful tool for track reconstruction
- **With our GNN configuration** GPUs perform better than TPUs according to the three metrics described.
  - Distributed training strategy in TF partly to blame
- Next steps:
  - Study mixed precision and other optimizations
  - IPU

GPU V100 and A100

GPU Architecture	NVIDIA Volta	NVIDIA Ampere
NVIDIA CUDA Cores	5120	
FP64 [TFLOPS]	7	9.7, TensorCore: 19.5
FP32 [TFLOPS]	14	19.5, TF32: 312
GPU Memory	16 GB HBM2	40 GB HBM2
GPU clock	1245 MHz	765 MHz
Memory bandwidth	900 GB/sec	1.6 TB/sec
PCIe	32 GB/sec (Gen3)	64 GB/sec (Gen4)
NVLink	300 GB/sec (Gen2)	600 GB/sec (Gen3)

Analyze profiling with roofline model





Analyze profiling with roofline model