Exa.TrkX & GPU Acceleration with Inference as-a-Service

Yongbin Feng[1], Shih-Chieh Hsu[2], Xiangyang Ju[3], Alina Lazar[4]

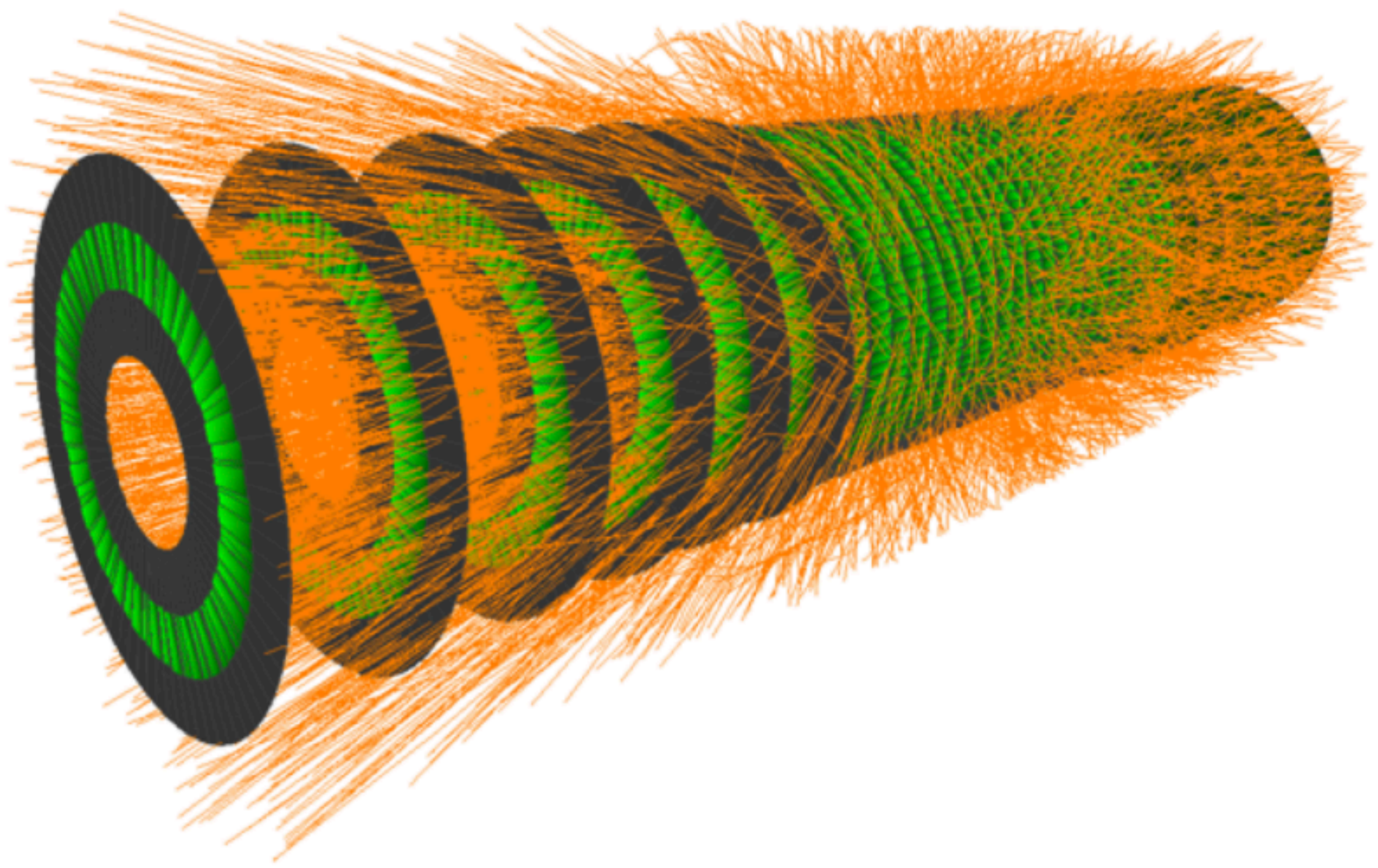
[1]Fermilab, [2] Univ. Washington, [3] LBNL, [4] Youngstown State Univ.

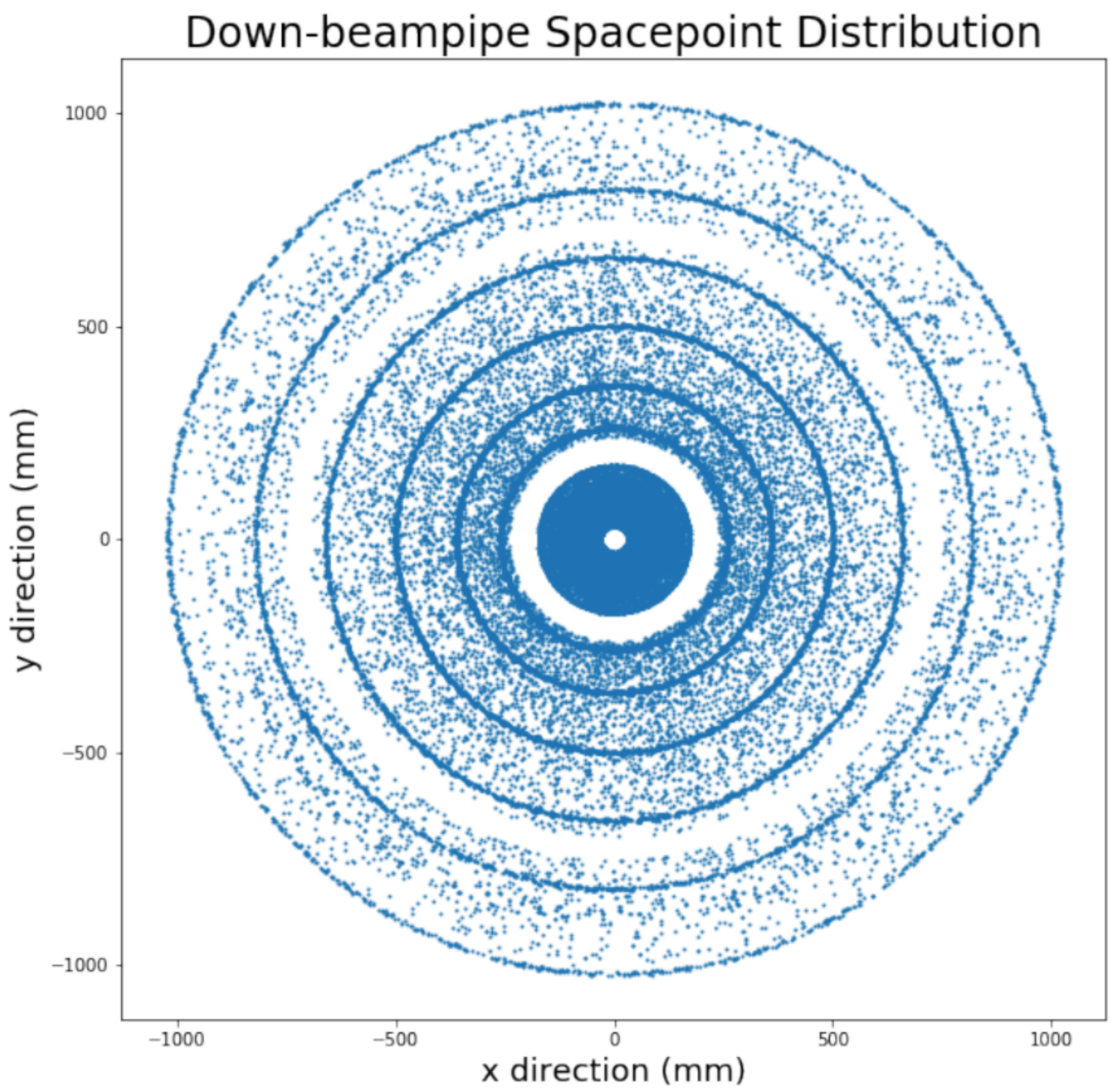
2022 Fast Machine Learning Workshop

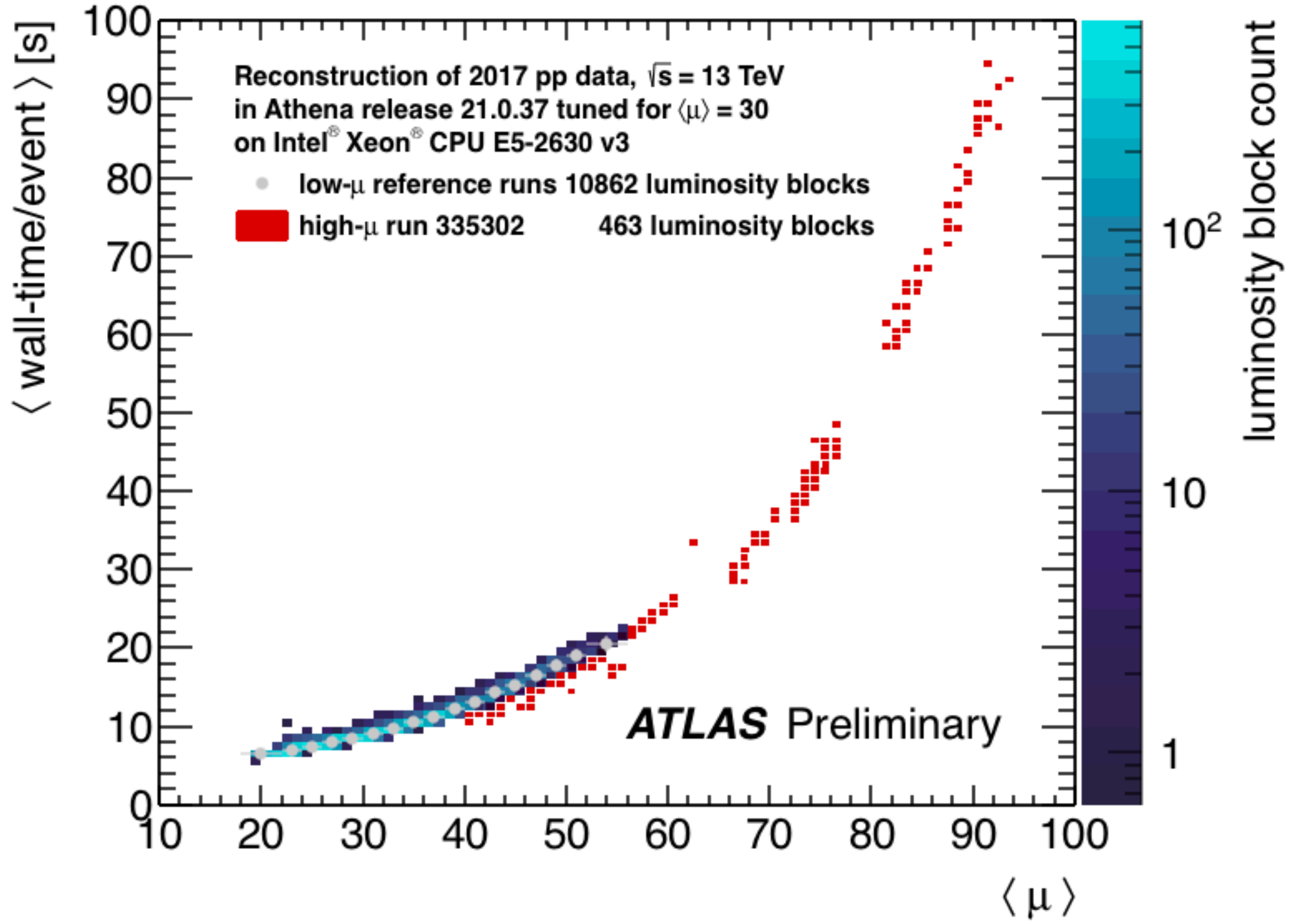
Dallas, Texas

October 3rd, 2022

Track Reconstruction at the HL-LHC

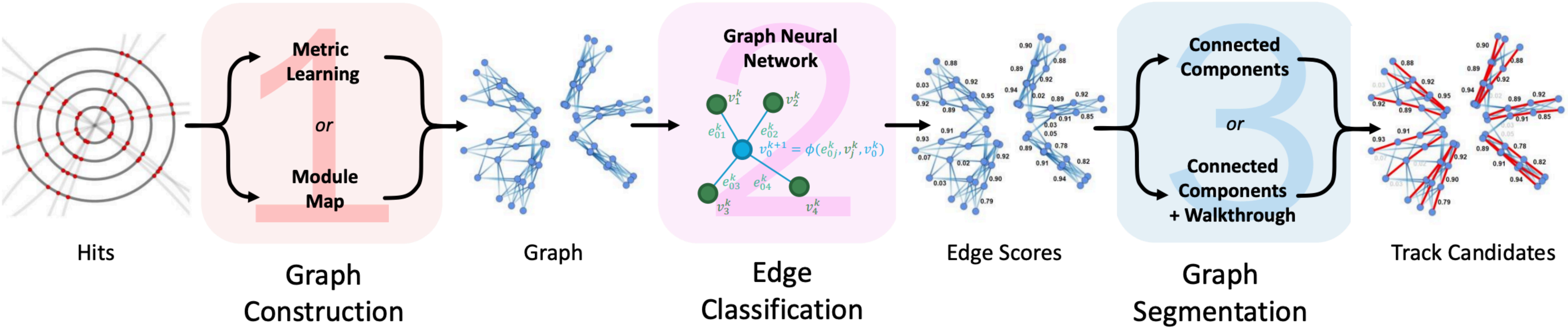




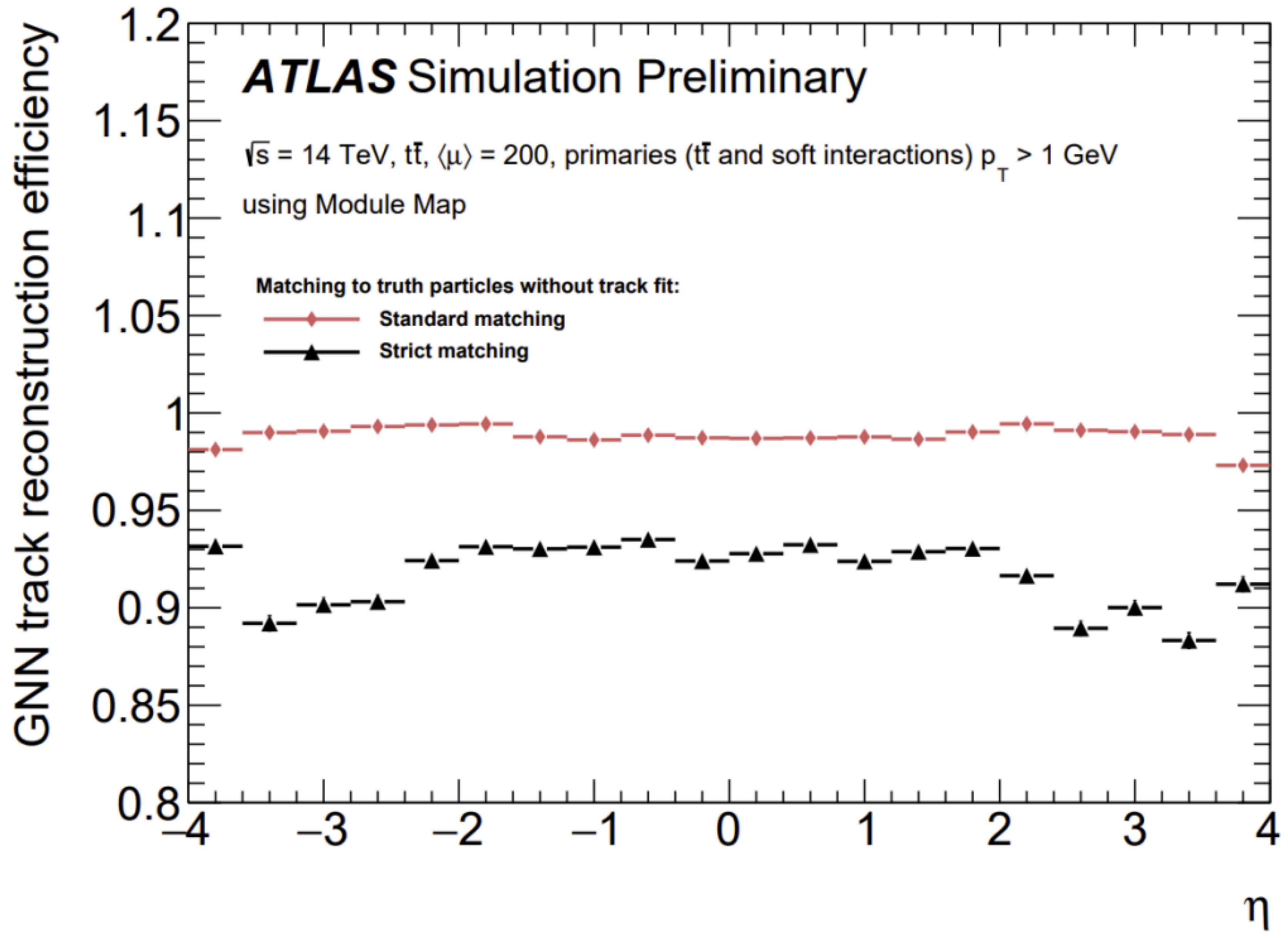


- Track reconstruction is expected to be very challenging in the future, especially at the HL-LHC:
  - A ttbar event with 150-200 pileup at the HL-LHC will produce O(5K) charged particles, and O(100K) spacepoints
- Computing cost does not scale linearly with number of pileup. Track reconstruction takes the major fraction of time among all the reconstruction steps

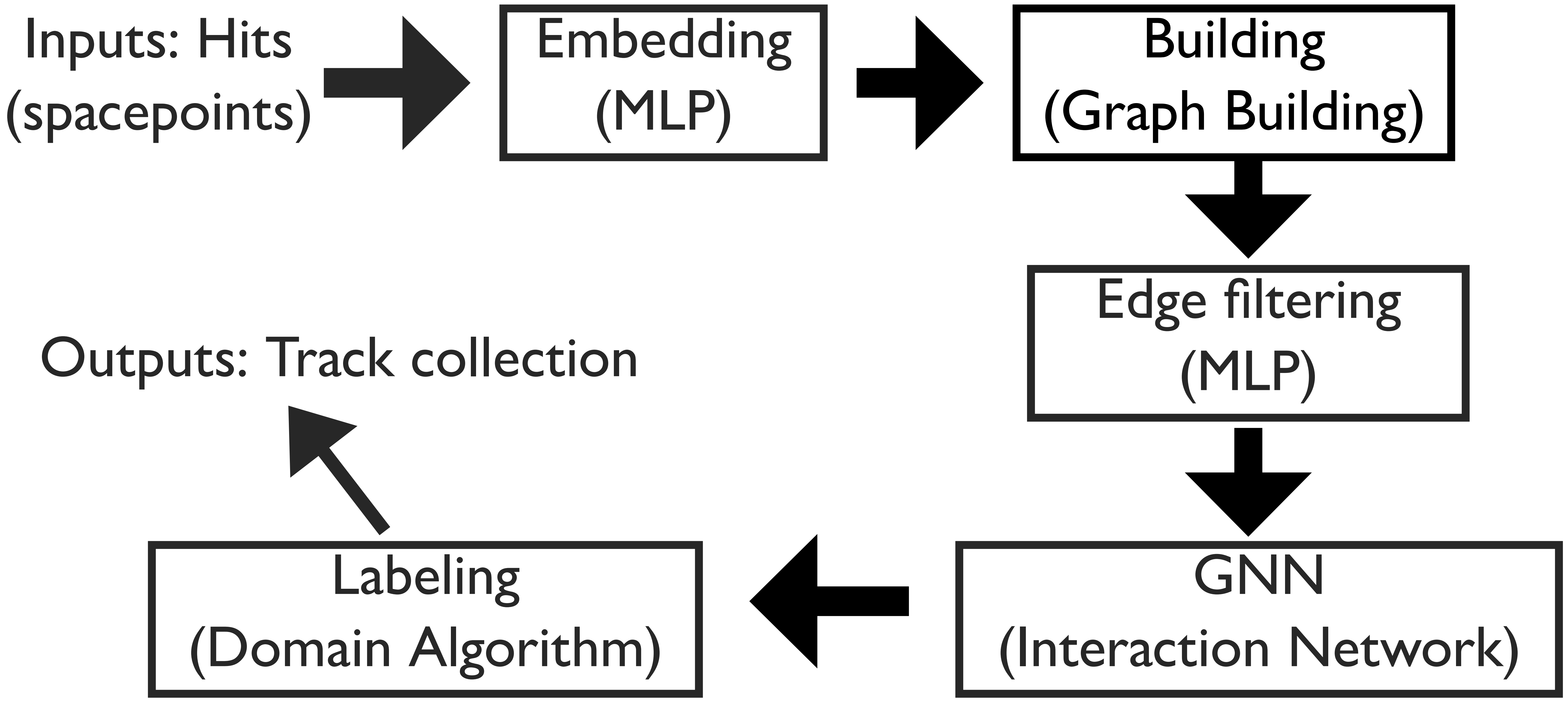
ML-based Track Reconstruction

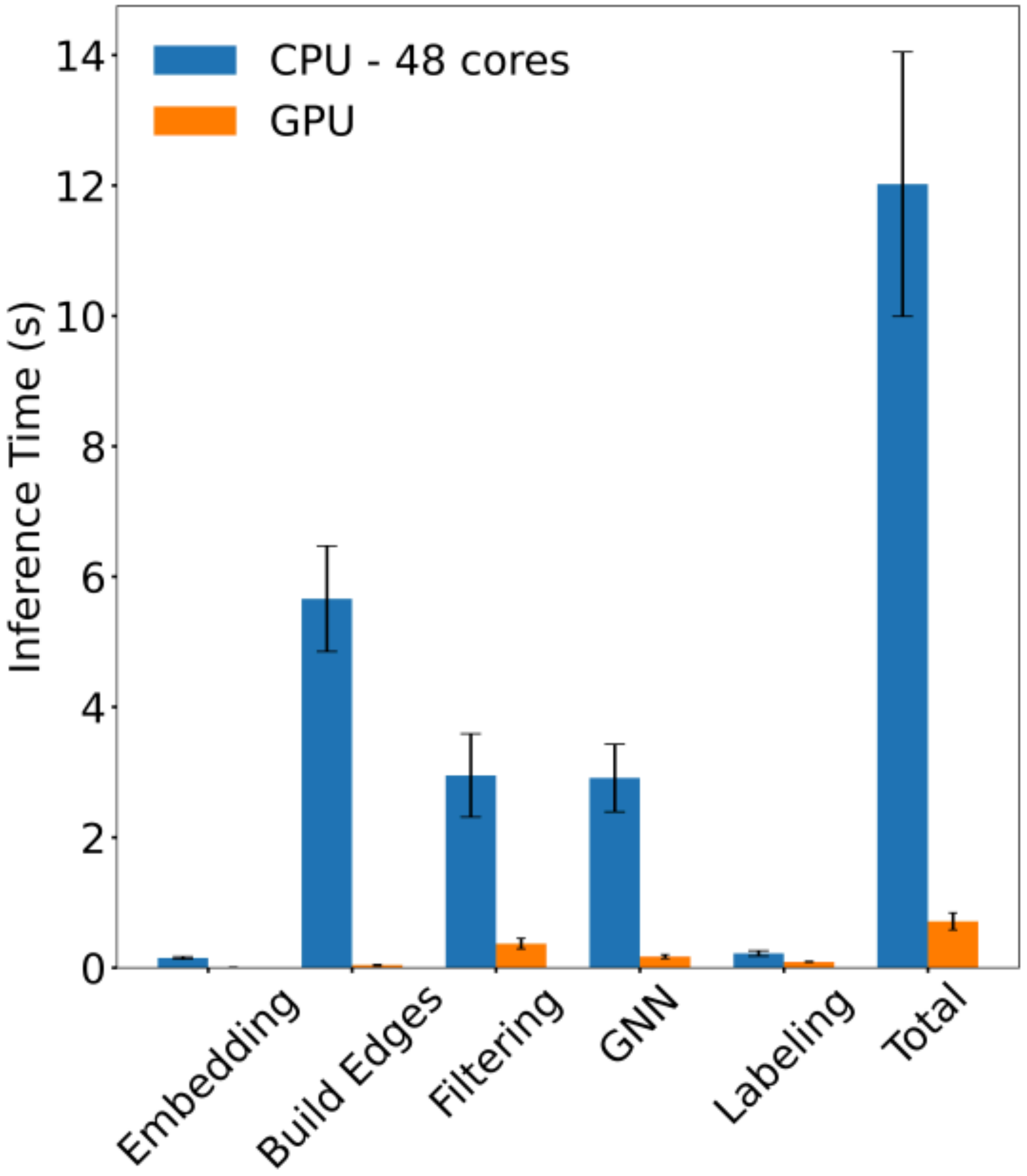


- ML-based track reconstruction with GraphNN could be a promising solution:
  - ML algorithms can run fast, easy to optimize, and easily accelerated on different coprocessors to get faster
- Good performances on the 200 pileup simulation datasets: similar efficiency as the classical algorithm, and O(10^{-3}) fake rates



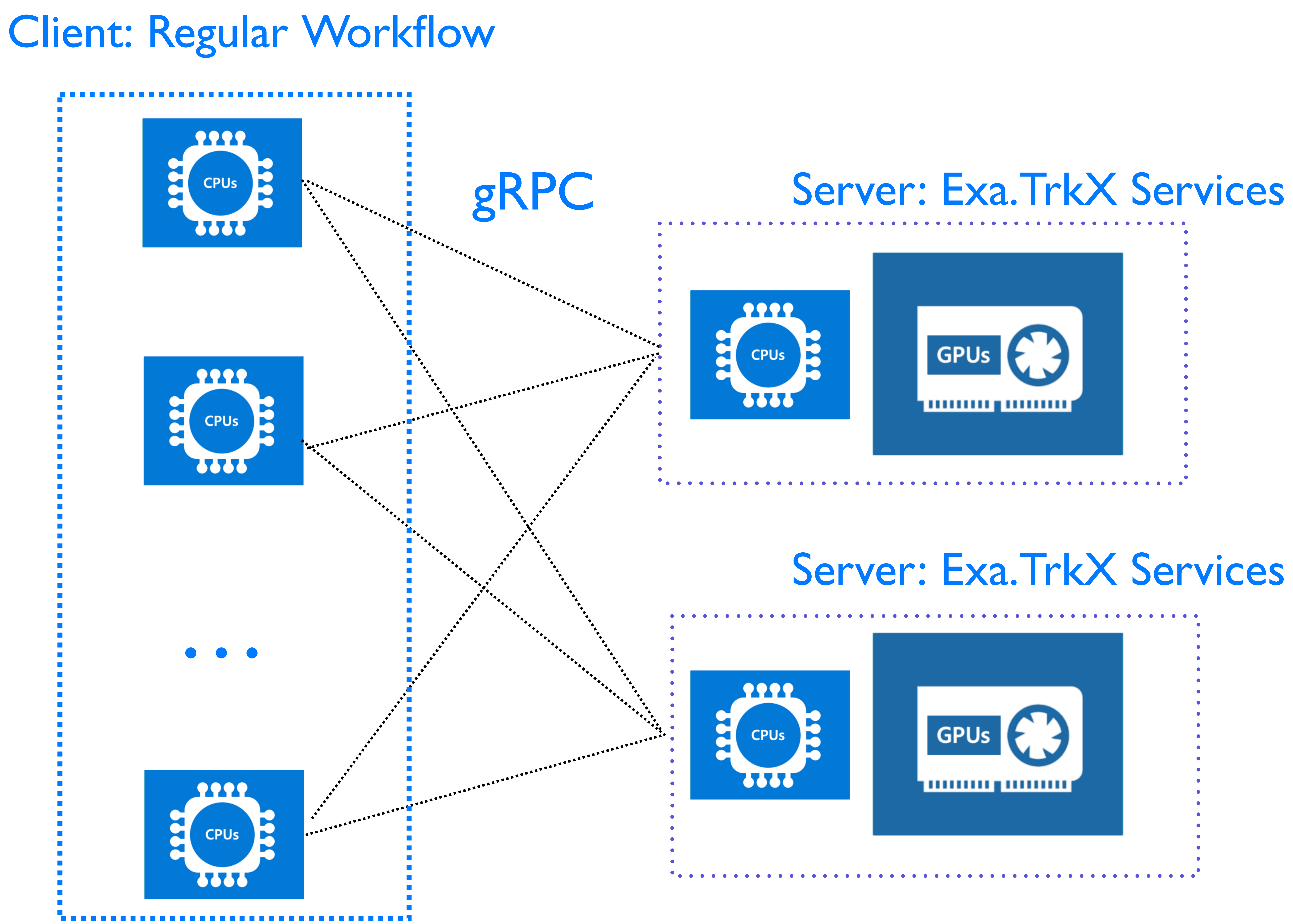
Inference Costs





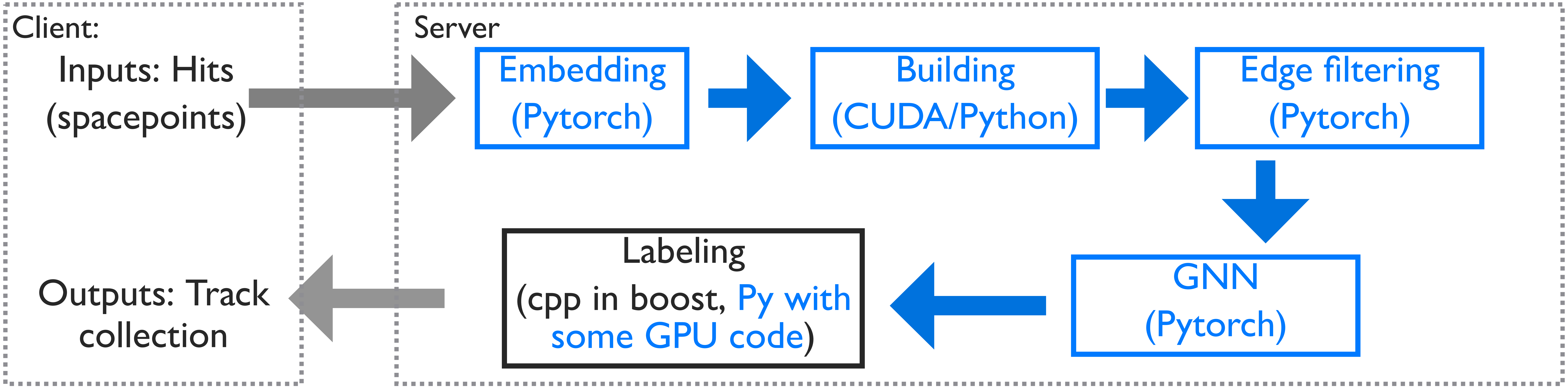
- Workflow runs much faster on GPUs compared with CPUs after optimizations: from O(20s) on 48-core Intel Xeon 8268s CPUs to <1s on NVIDIA V100. More details on Arxiv.2202.06929

Inference As-a-Service



- Inference as-a-Service provides lots of benefits, e.g.:
  - Separate ML inferences out of the main software, easy to maintain
  - Enables access to remote GPUs;
  - more flexibility of the CPU/GPU ratios;
  - Easy deployment on different types of coprocessors
  - Etc
- More in Patrick's talk and Dylan's talk

Current Exa.TrkX Workflow with as-a-Service



- Server side uses NVIDIA Triton Inference server. Various features and benefits:
  - Supports of different backends: ML including TF, Pytorch, ONNX; domain algorithms: CUDA, Python, Cpp
  - Ensemble model that can collect the whole inference modules together; reduce the IOs between client and server
- Pytorch models runs out of the box; CUDA and cpp implementations currently done with Python custom backend

Preliminary Results

Direct Inference	ms/evt
Embedding	0.5
Building	2.2
Filtering	27.6
GNN	31.7
Total	62

As a Service	ms/evt
Embedding	1.7
Building	7.3
Filtering	26.7
GNN	21.3
Total	64.4

- Benchmarked in the 0-PU dataset to start with.
- Time not including the labeling part (domain algorithm code; takes some efforts to prepare a custom backend for it)
- Similar inference time between CPU-GPU directly connected and CPU-Server with aaS:
  - Also checked the server-side metrics: the fraction of time to handle IOs are small. Most of the time are on computations.

Summary

- Track reconstruction is expected to be very challenging in the high-density environments. ML-based approaches are naturally nice candidates to solve such problems.

- Current Exa.TrkX models have very promising results, similar to domain algorithms and runs faster. These ML algorithms can easily be deployed on different hardwares and accelerated:
    - Preliminary results indicates that it can run 20-100 times faster on the GPUs compared with CPUs.

- As-a-Service version of the Exa.TrkX inference workflow implemented. Preliminary results show consistent behaviors with directly-connected, but more flexibilities. More studies and results in the future!

Back Up