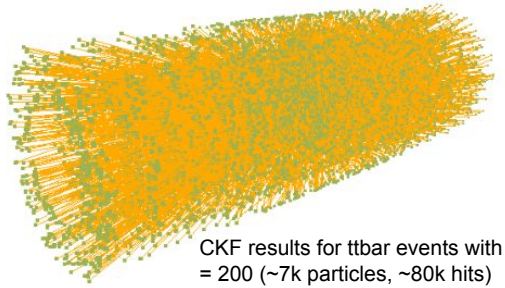
UC Berkeley/Stanford ACTS

Project + Personnel Overview

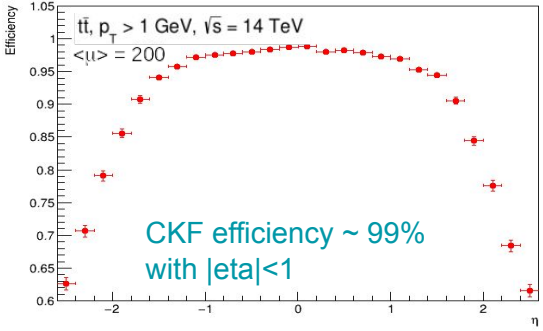
- Stanford
  - Lauren Tompkins (PI), Rocky Garg (postdoc), Marcelo Vicente (FPGA Engineer), Peter Chatain (Undergraduate Summer 2020), Cole Mocyemba (Undergraduate Summer 2020)
- UC Berkeley
  - Heather Gray (PI), Xiaocong Ai (postdoc), Ralf Farkas (fellow, also Bonn),
- Collaborators, some examples
  - ACTS (Andreas Salzburger, Paul Gessinger-Befurt, Moritz Kiehn, ...)
  - Charles Leggett, Beomki Yeo (LBL)
  - Nick Styles, Georgiana Mania (DESY)
  - EIC (Barbara Jacek, Markus Diefenthaler)
  - sPhenix (Joe Osborne, Anthony Frawley)
  - Belle-2 (Florian Bernlochner, Ralf Farkas)

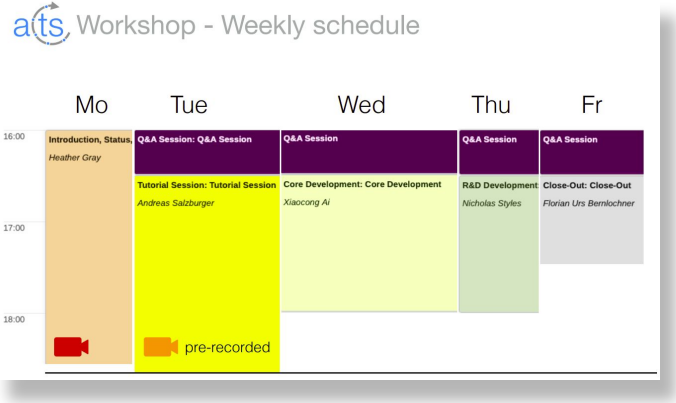
Achievements over the past year

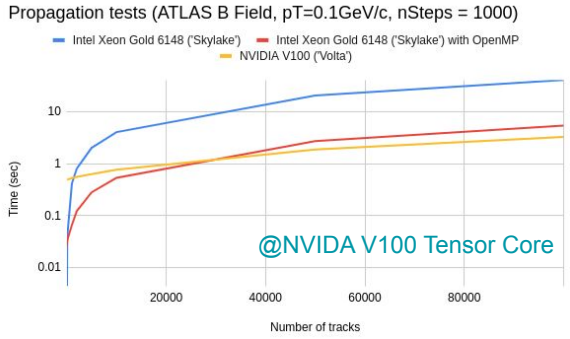
- Kalman Filter and Combinatorial Kalman Filter (CKF) Development
  - First for track fitting, then for track finding
- ACTS components on GPU
  - Track seeding
  - First pass of track fitting (in progress)
- Prototype alignment algorithms in ACTS (in progress)
- ACTS talk from X. Ai at Connecting the Dots
- Ongoing (virtual) ACTS workshop
  - 54 participants during the first day (!)
- Prototyped puttingTrkX GNN on Alveo -- terminated after it wasn't a good fit

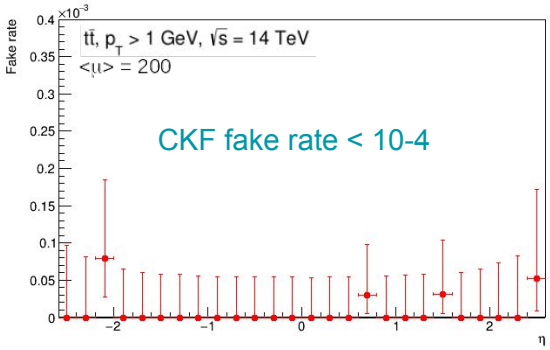


CKF results for ttbar events with mu = 200 (~7k particles, ~80k hits)









Milestones

Year 1 + 2

- Hosted ACTS workshops in 2019 and 2020 (2019 was a milestone)
- Benchmarking of ACTS components on GPU (done)
- Machine learning implementation for ambiguity resolution (prototype done, missing student to complete)

Proposed for Year 3 (wording to be fine-tuned)

- ACTS demonstration on ATLAS and/or ATLAS-ITK detector
  - Comparison to current ATLAS algorithms
  - Integration and comparison of ACTS algorithms in Athena
- Kalman Filter prototype on GPU
  - Full KalmanFilter should be on GPU by this September
- Alignment implementation in ACTS
  - Algorithm prototype should be done by this July
- Evaluation of novel tracking algorithms (end 2020?)
- Benchmark I/O and sparse memory access for Alveo (FPGA accelerator)

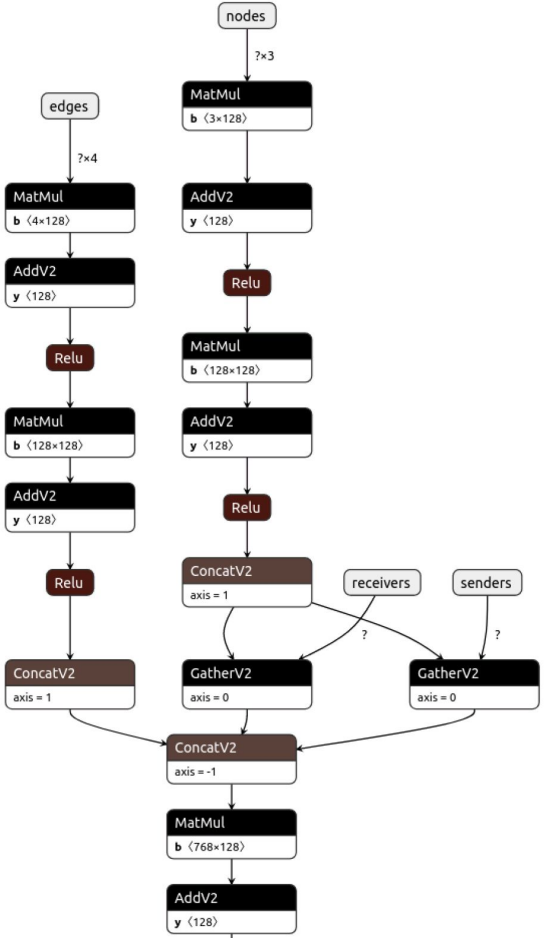
Feedback + Outcomes

- Overall goal: A complete, modern and experiment-independent track reconstruction package
  - Deployed within the athena framework
  - Used by other high-energy and nuclear physics experiments
- Performance evaluation and optimization (e.g. GPU or FPGA accelerated) of tracking algorithms

Back-up

Mapping GNNs into FPGAs

- Focus on “Connecting The Dots” from HEP.TrkX
- ~650k hits/event on TrackML
- Pre-processing before GNN:
  - Initial cut applied to reduce data volume
    - ~250k hits after selecting barrel
    - ~30k hits after 1GeV cut (unclear how to do this IRL)
    - ~15M segments (hit to hit connections/edges)
  - Second cut reduces data volume even further
    - Cut in depth, angle between hits, etc.
    - **~5k hits and ~10k segments**
    - Binary connection matrices (very sparse)
      - Known as receivers/senders in graph input
      - Represents edge/node connectivity



Mapping GNNs into FPGAs

- Actual GNN needs several passes, e.g. 15-passes, for proper accuracy
- Gathering in FPGA cannot be easily parallelized
  - Scales well with CPU/GPUs, but not FPGA
  - Pipelining is difficult since previous and forward stages need to wait or buffer until gather completes
  - Memory bandwidth and frequency dependent
    - Internal memory likely not enough
    - Complicated internal routing
  - Latency and throughput impacted