Introduction

Nikolai Hartmann

LMU Munich

July 27, 2023, PyHEP.dev Workshop

About me

- Postdoc at LMU in the group of Thomas Kuhr (Belle II)
- Working on Software, Computing, ML
- Did PhD at ATLAS
- still involved to make columnar data analysis working with `DAOD_PHYSLITE` (small(ish) format for end-user analysis, similar to CMS NanoAOD)
- Big fan of columnar data analysis / array programming and awkward arrays (i also like fitting and statistics stuff, e.g. pyhf)

What i'm struggling with / want to talk about in this workshop ...

uproot reading / awkward representation of ATLAS `DAOD_PHYSLITE`

- Currently lots of not-so-nicely readable branches (`vector<vector<...` needs loops)
  → fortunately largely solved by `awkward forth`
- Currently cross references all over the place
  - can be represented with awkward `IndexedArray`
  - slightly more complicated than in NanoAOD due to not a-priori knowing where to link to
  - but: basics implemented in `PHYSLITE` schema in `coffea.nanoevents`
    → need to "daskify" the linking stuff (thank a lot to Lindsey for the help)

ML preprocessing bag of tricks

- Feeding "awkward" data into ML models becomes increasingly popular
- However, students struggle getting the preprocessing done efficiently
- I have learned a bag of tricks ... but is there some common functionality missing? (and have others similar use cases?)

Example 1: join several flat ntuples of particle lists

- have several flat TTrees of different particle candidates
- want to join into per-event lists of all particle candidates using a set of identifying columns (e.g. event number, production number)
- **Trick**
  - concatenate all flat candidate lists
  - use pandas groupby and `.indices` to get indices into flat array
  - `ak.run_lengths` might also work, but a bit cumbersome with multiple columns

Example 2: use variable length lists with masking or graph network libraries

- slow: loop over awkward array
- could use numba (does actually support generators)
- or flatten array + `ak.num` and loop over slices to produce list of numpy arrays
  $\rightarrow$ loop over this fast enough to produce padded batches and graph NN library representations