

10 TeV MuCol

MEMORY CONSUMPTION ISSUES



TOVA HOLMES, U. OF TENNESSEE
10 TEV MUCOL STUDIES
OCTOBER 7, 2023

Memory consumption issues

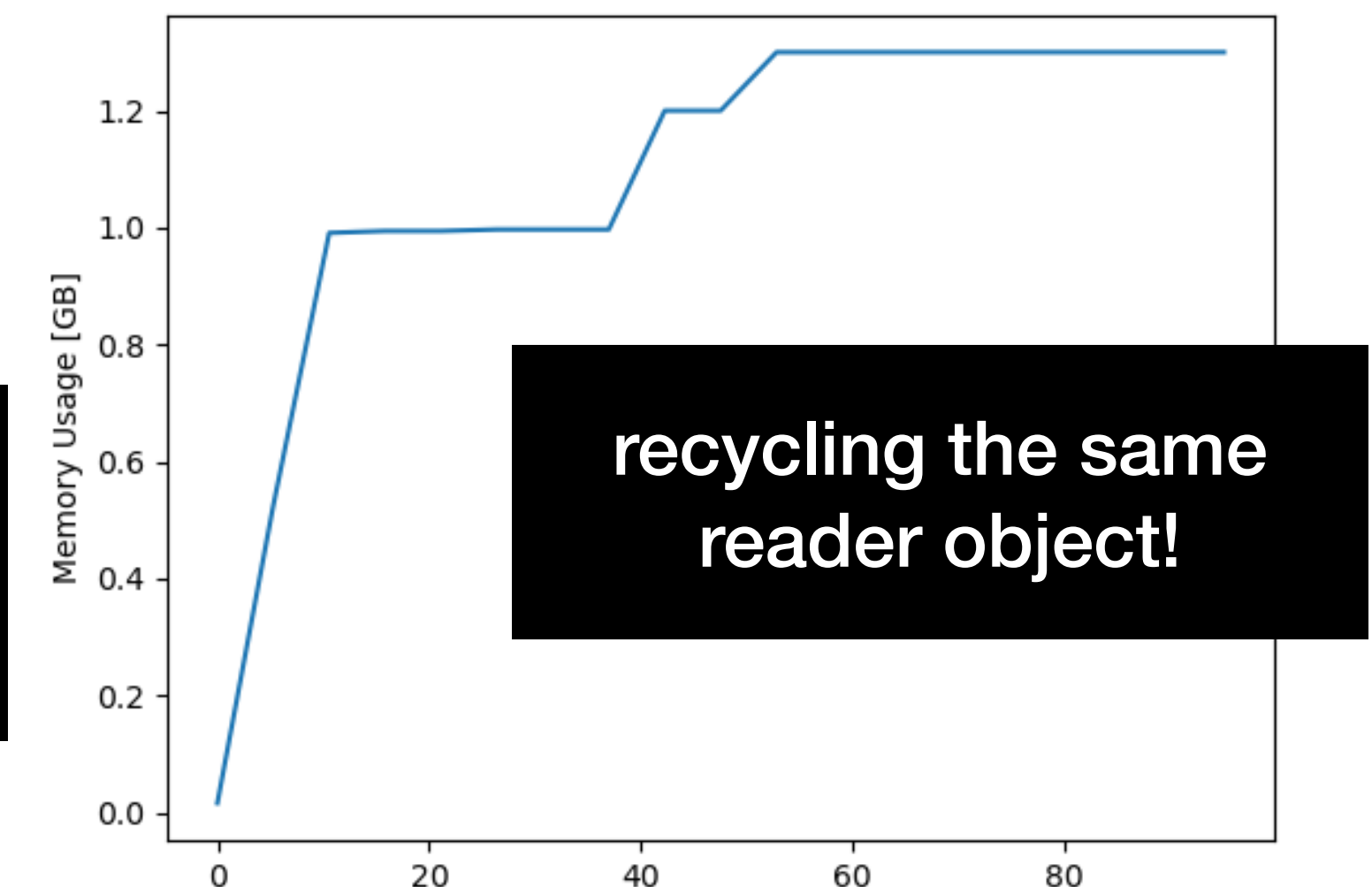
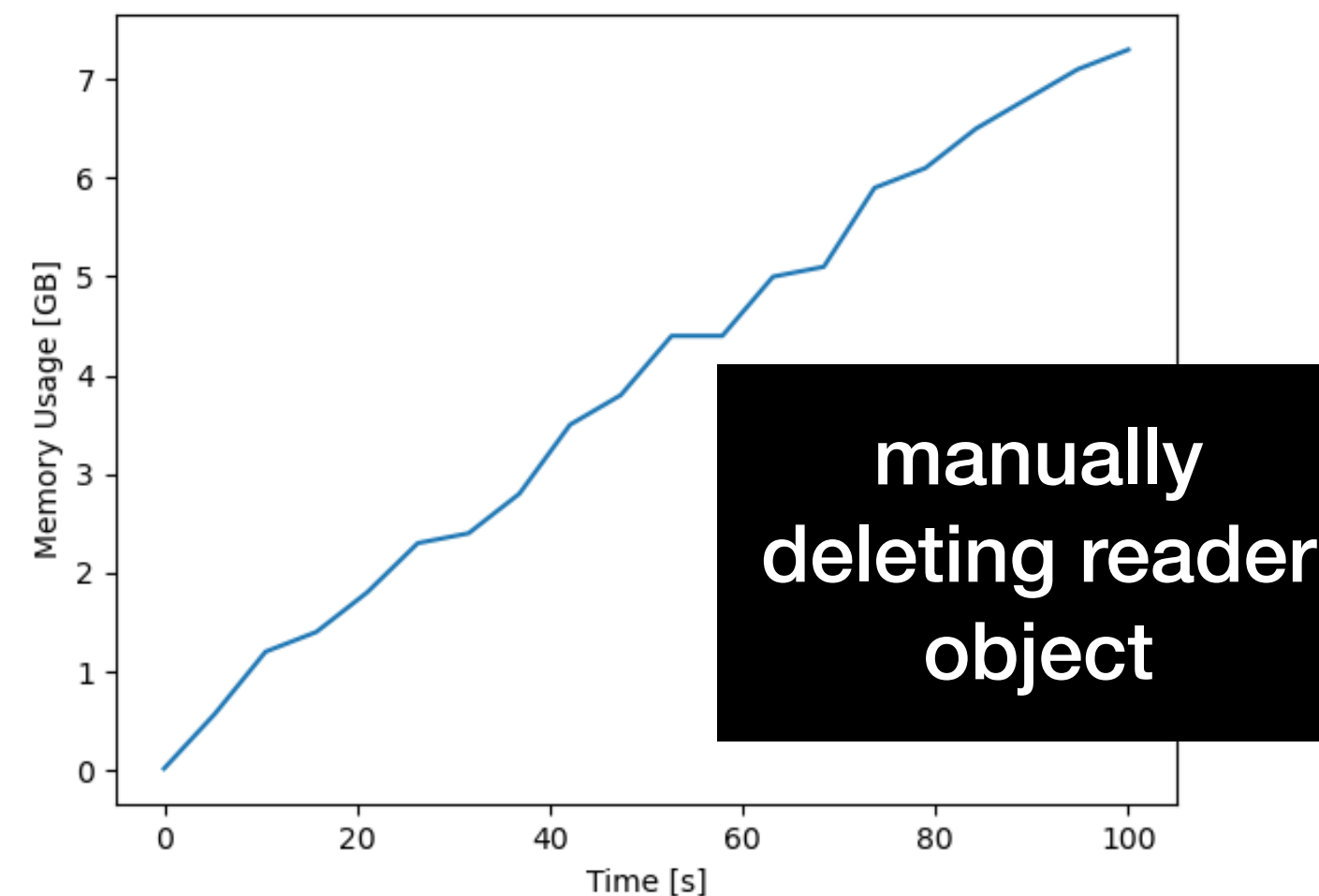
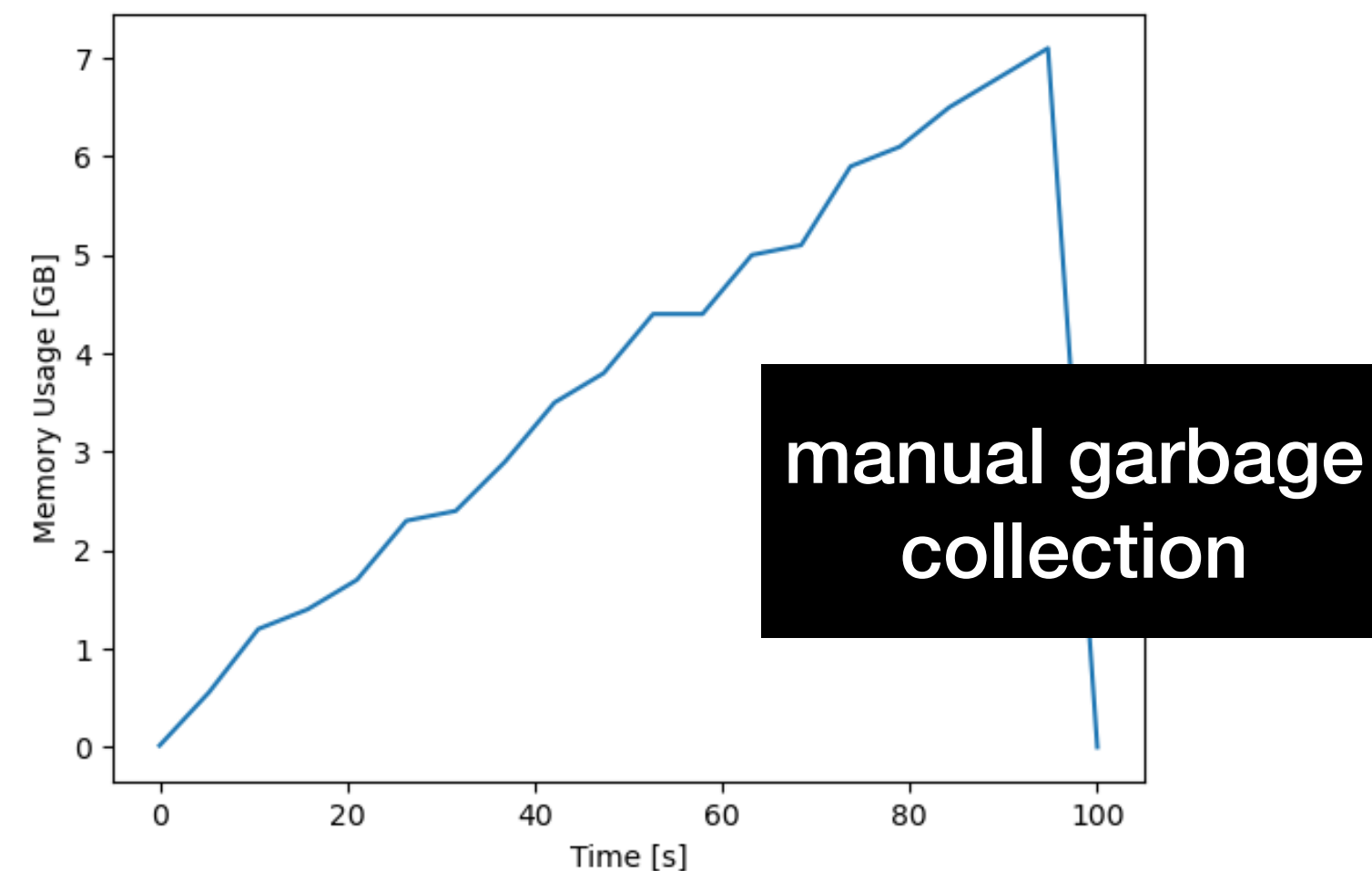
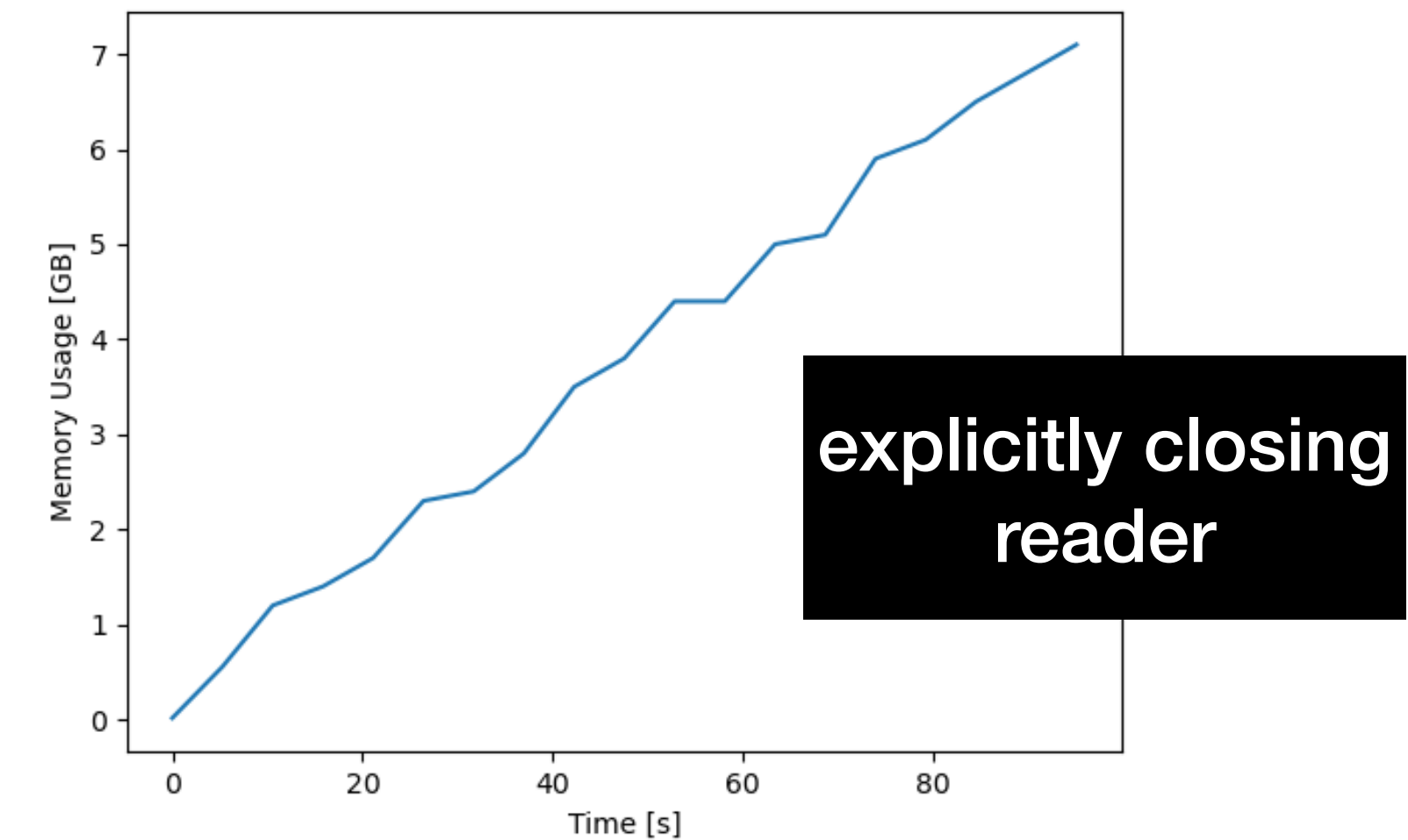
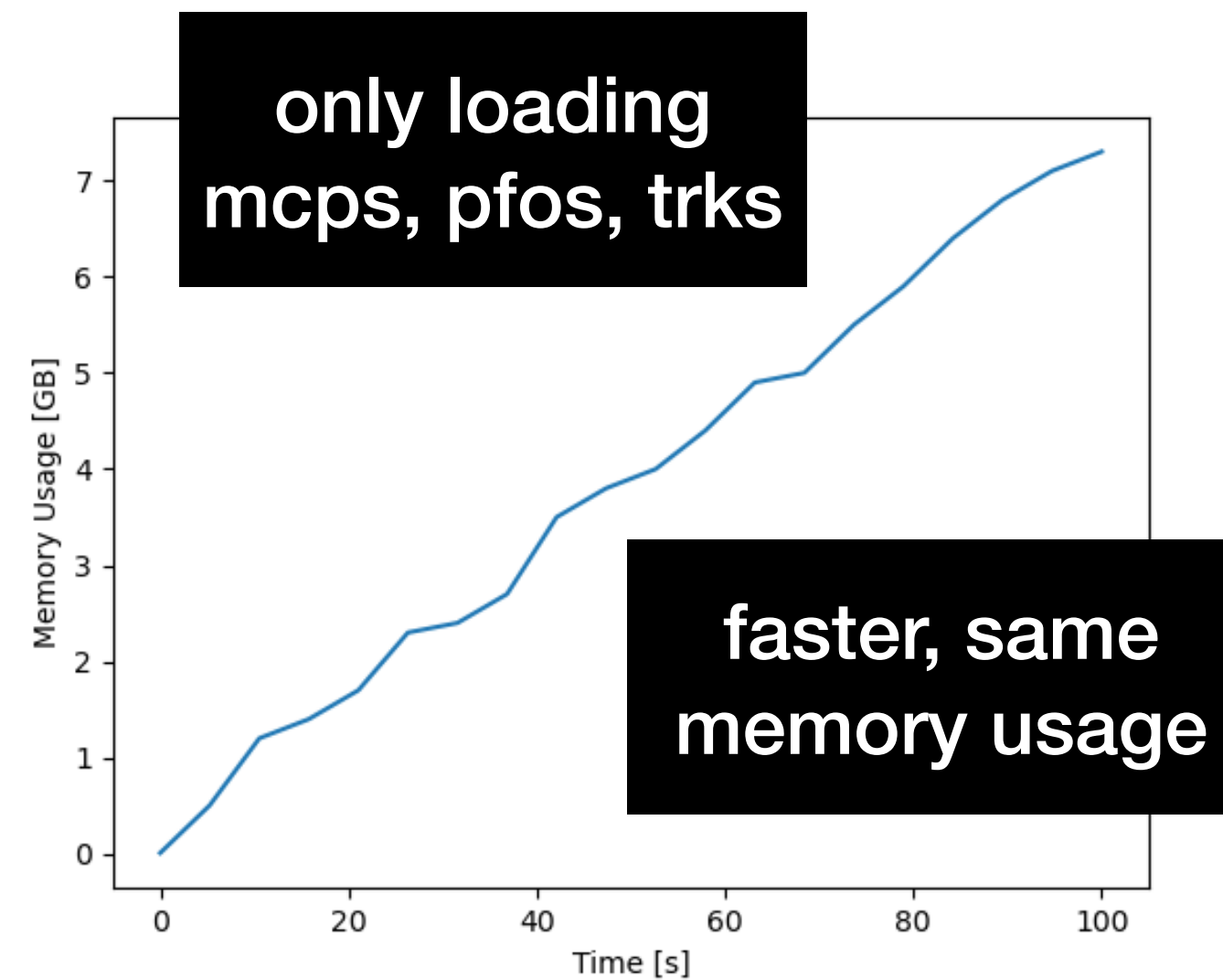
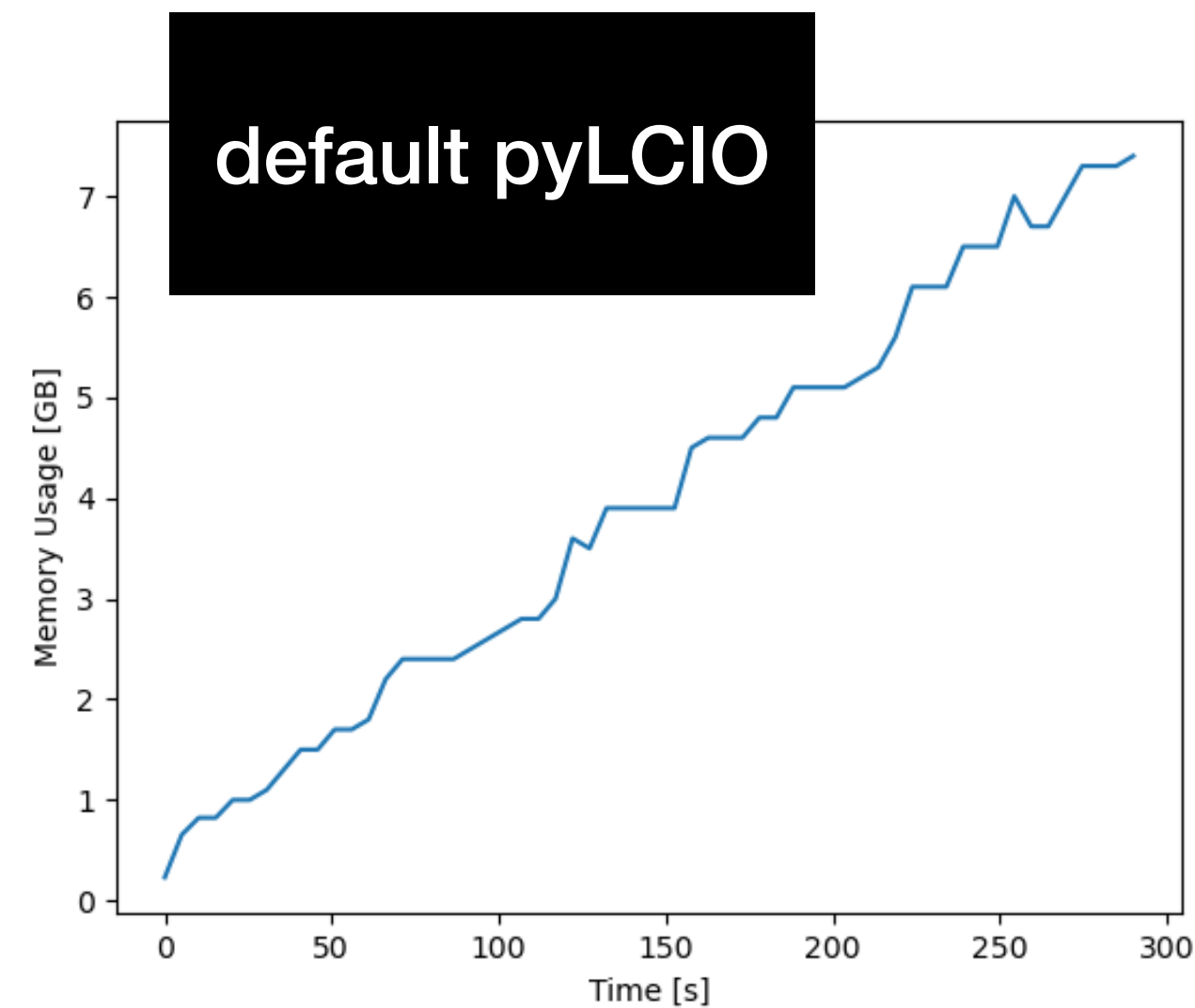
- Trying to run over all the electron samples w/o BIB
 - Tried both key4hep and slcio and in both had memory issues
 - Tried a simple example with nothing in the loop, just opening files in slcio

```
+ test.py
import glob
import gc
max_events = 1000

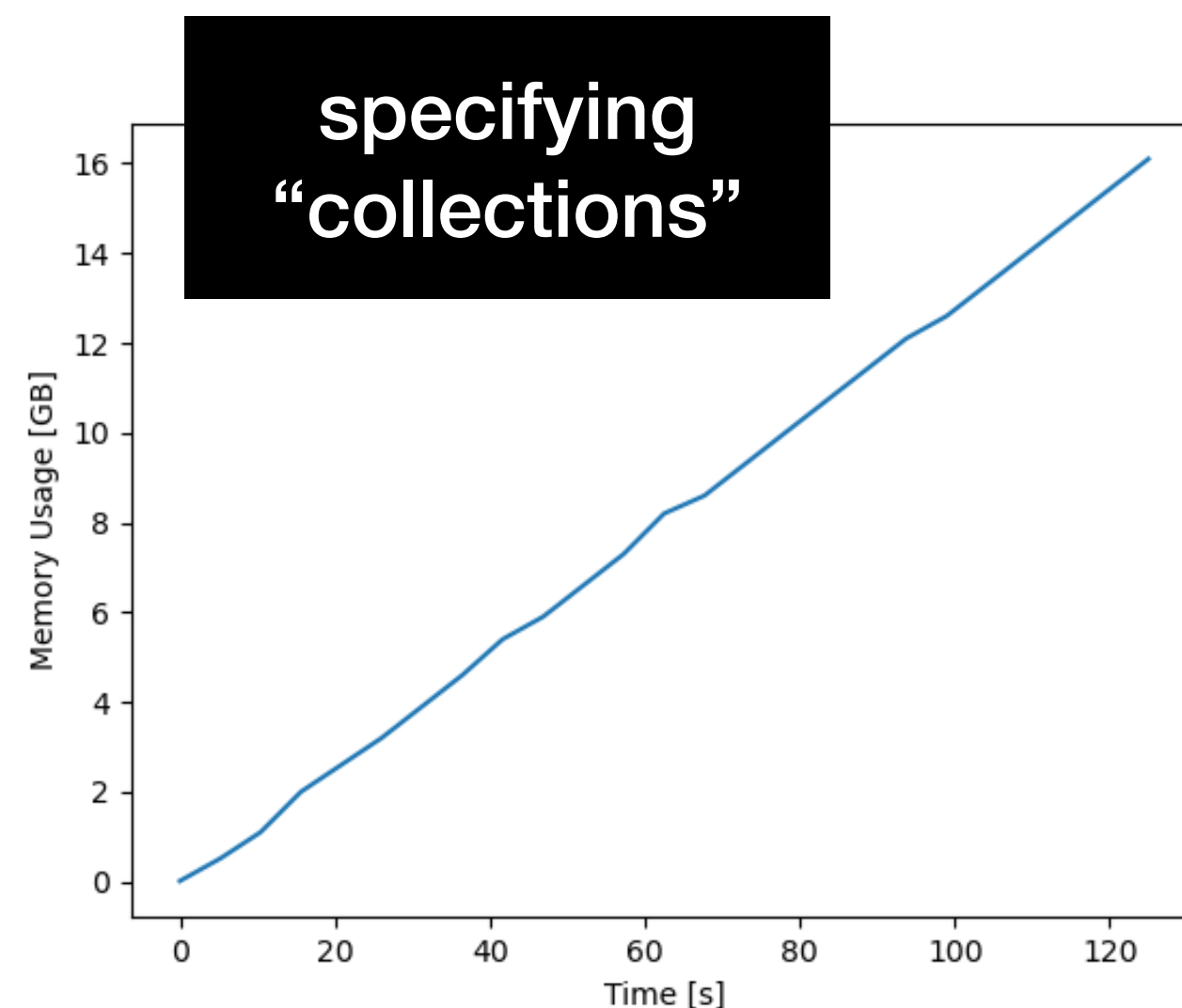
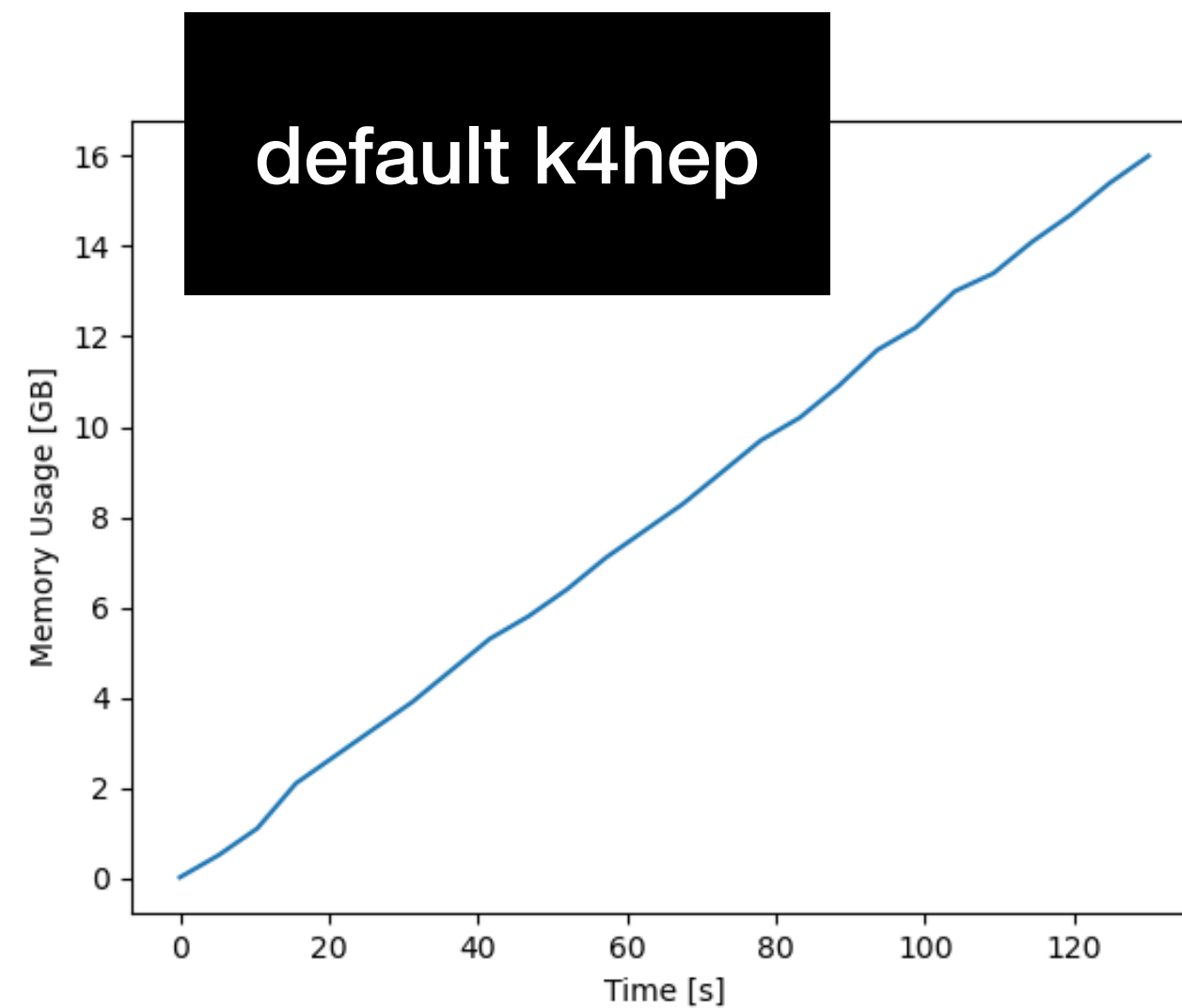
# pylcio simplified setup
import pyLCIO
files = glob.glob("/data/fmeloni/DataMuC_MuColl10_v0A/reco/electronGun_pT_250_1000/*.slcio")
i = 0
for f in files:
    if i >= max_events: break
    reader = pyLCIO.IOIMPL.LCFactory.getInstance().createLCReader()
    reader.setReadCollectionNames(["MCParticle", "PandoraPF0s", "SiTracks_Refitted"])
    reader.open(f)
    for event in reader:
        if i >= max_events: break
        if i%100 == 0: print("Processing event ", i)
        i += 1
    reader.close()
    gc.collect()

print("Done.")
~
~
~
~
~
~
~
```

To run 1000 events, nothing done inside loop, pyLCIO



To run 1000 events, nothing done inside loop, k4hep



- Default is using the `podio.root_io.Reader(filename)` function
 - No ability to load files separately from initializing reader (I think?) so cannot do the same fix
 - Can specify collections
 - `reader.collections = ["MCParticle", "PandoraPFOs", ...]`
 - Doesn't do anything (maybe needs to be called before opening a file, but can't because initialization is the only way to pass a filename AFAIK)
 - Can still run code that accesses un-specified collections

Without fix, runs faster but consumes more memory compared to pyLCIO
Possible solution is to work with the `ROOTFrameReader()` directly and have more tunability

Memory consumption issues

- In the end, was able to get OK memory performance with pyLCIO
 - Didn't have access to the same knobs in key4hep
 - Still very slow!! (around 10 events/s)
 - Not loading in MCPs halves memory but not much speed change (around 13 events/s)
 - Merging files may help with this

```
+ test.py
import glob
#import gc
max_events = 10000

# pylcio simplified setup
import pyLCIO
files = glob.glob("/data/fmeloni/DataMuC_MuColl10_v0A/reco/electronGun_pT_250_1000/*.slcio")
i = 0

reader = pyLCIO.IOIMPL.LCFactory.getInstance().createLCReader()
#reader.setReadCollectionNames(["MCPParticle", "PandoraPF0s", "SiTracks_Refitted"])
reader.setReadCollectionNames(["PandoraPF0s", "SiTracks_Refitted"])

for f in files:
    if i >= max_events: break
    reader.open(f)
    for event in reader:
        if i >= max_events: break
        if i%100 == 0: print("Processing event ", i)
        i += 1
    reader.close()
    #del(reader)
    #gc.collect()

print("Done.")
```

