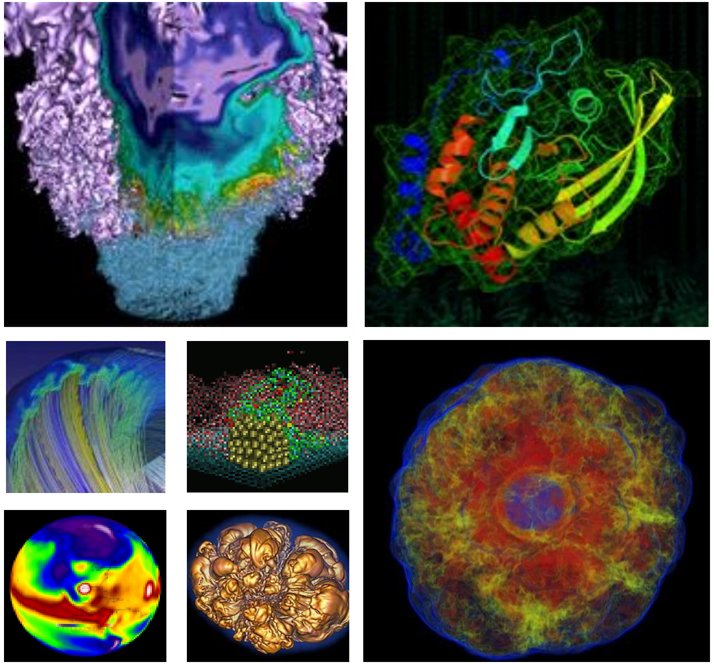
Deep500

Thoughts on Scientific DL Benchmarks

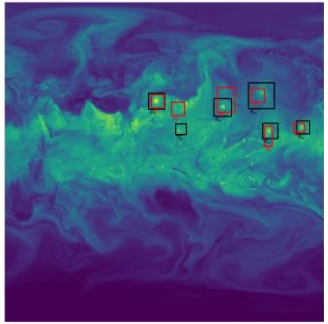


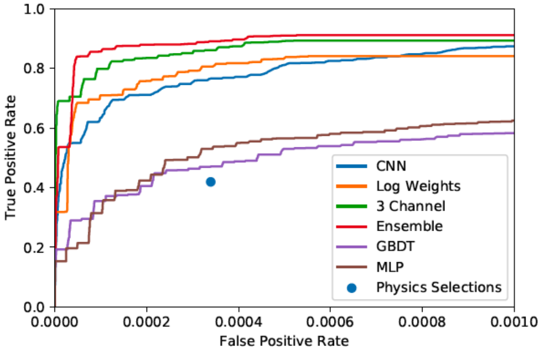
Thorsten Kurth, Mustafa Mustafa,
Steve Farrell, Prabhat

SC18 Deep500 BoF
Nov. 14, Dallas, TX

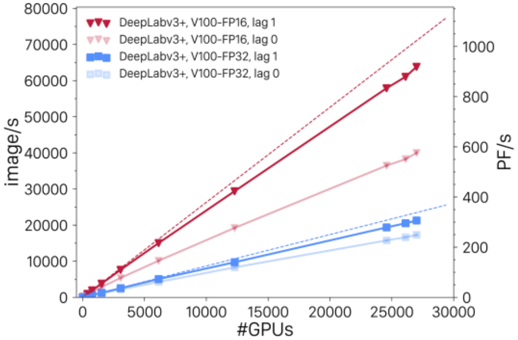
Deep Learning Works for Scientific Problems

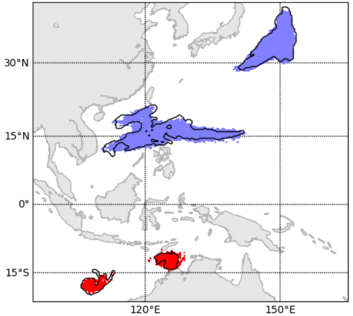
DL@15 PF, SC17



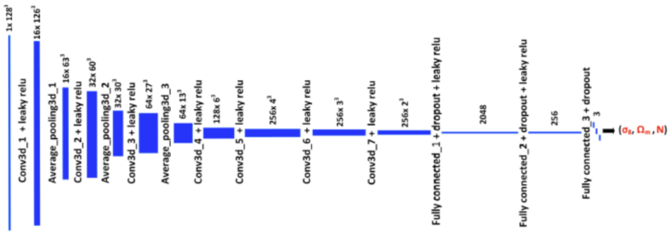


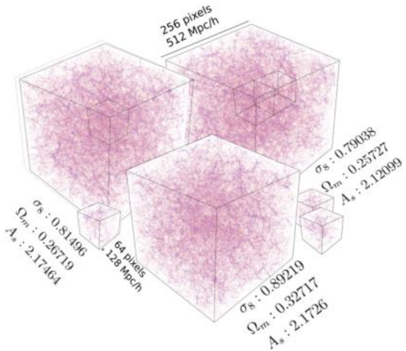
Exascale DL, SC18 GB



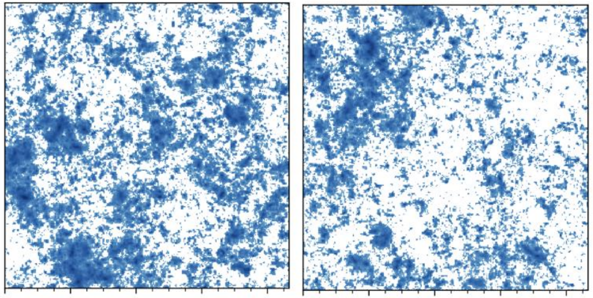


CosmoFlow, SC18



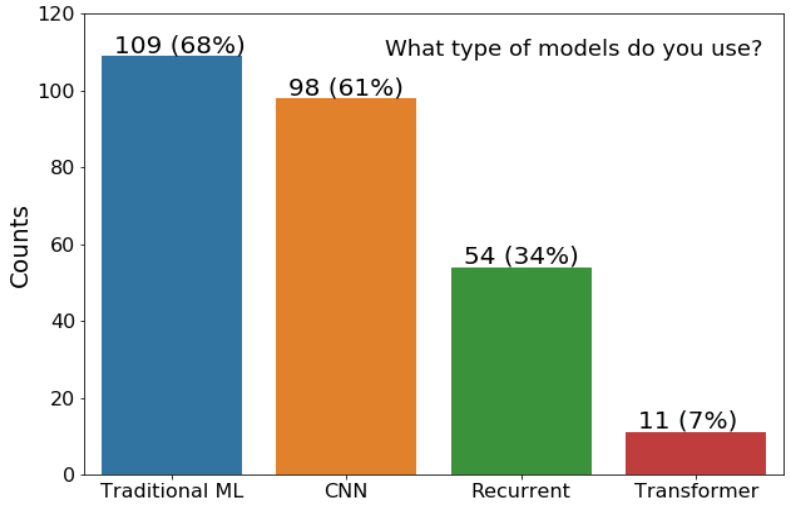


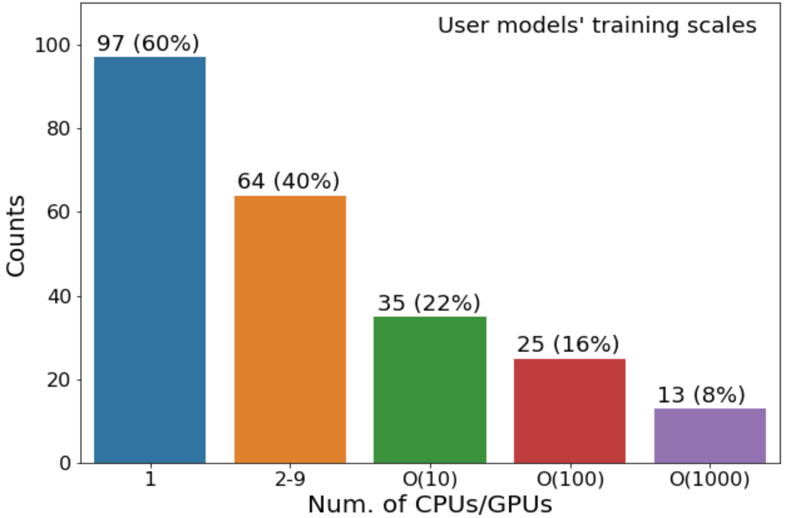
CosmoGAN



Simulation GAN

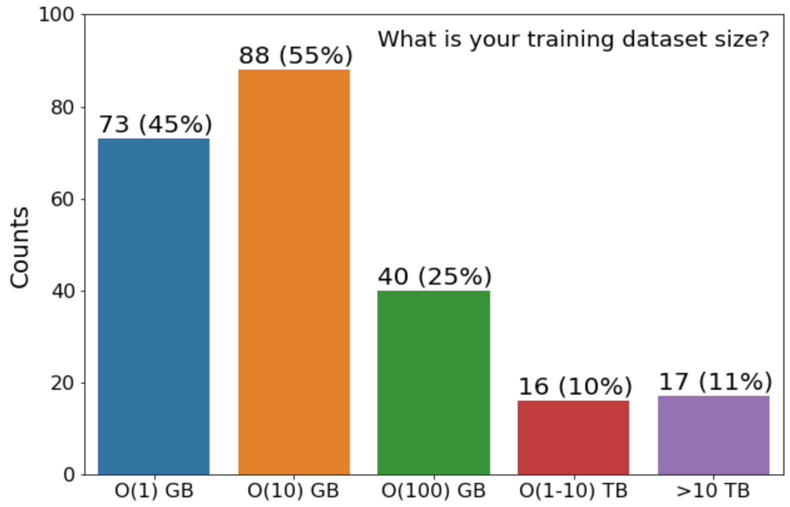
What DL workloads do we need benchmarks for?

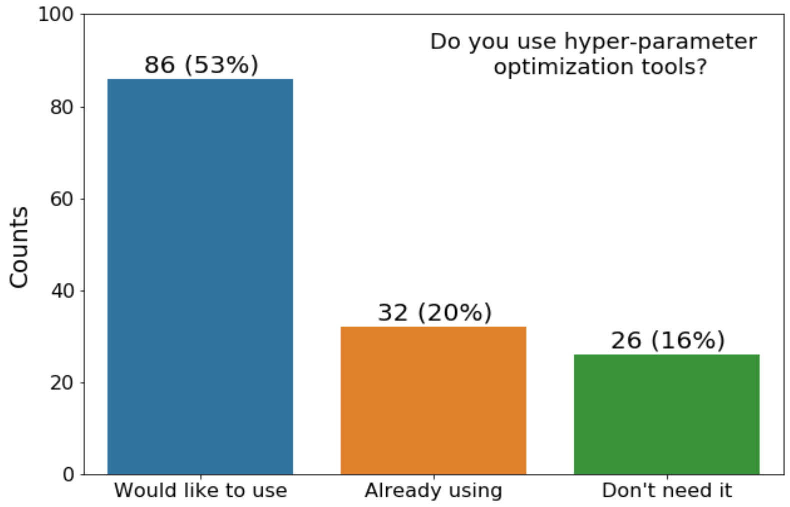




- results from ML@NERSC user survey
- various levels of sophistication
- large range of scales (with significant number of users training at more that 100 nodes)

Dataset sizes and HPO





- dataset sizes can be significant
- HPO-tools desired by large fraction of participants

Finding a (good) performance metric

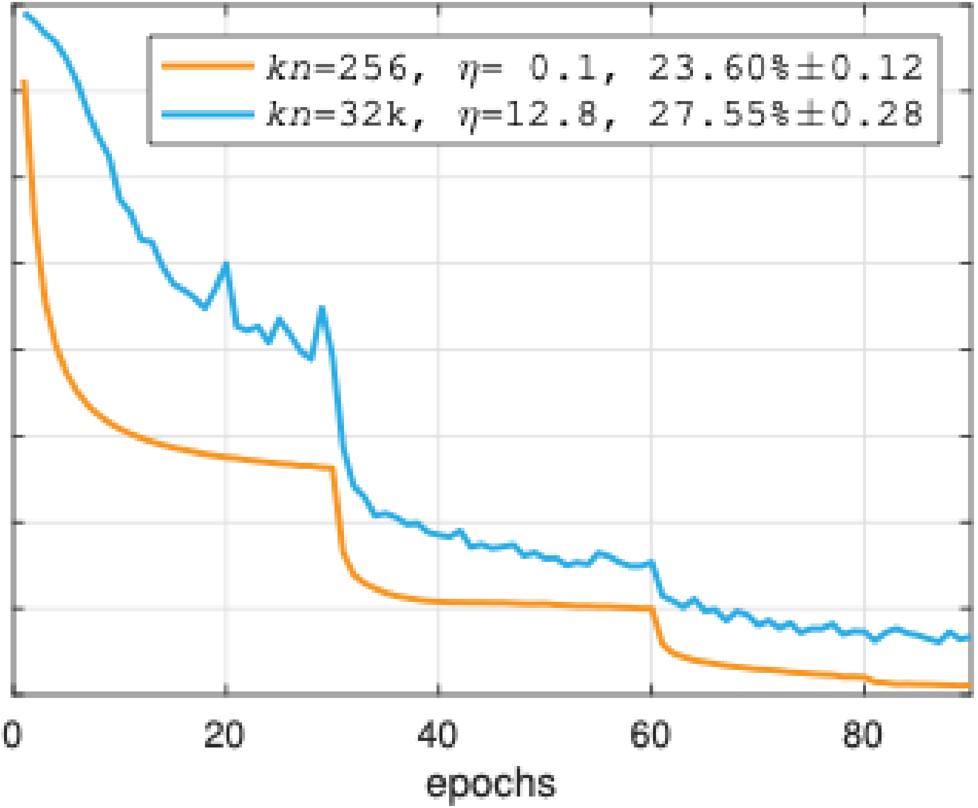
- throughput metric (e.g. samples/sec, time/sample, flops/s)
  - very easy to define and understandable
  - measures improvements in HW and SW stack (if training algorithm is kept fixed)
  - for inference workloads accurate in predicting speedup
  - for training workloads not necessarily related to time-to-solution

- time-to-solution (e.g. wallclock time to reach certain accuracy/loss)
  - relevant to DL practitioners, speedup numbers actually have a meaning
  - hard to define, e.g. what target score are we aiming at (problem dependent)
  - might mingle architectural advantages with HP optimization efforts and algorithmic advances/modifications

- time-to-solution+HPO (including architectural modifications, i.e. genetic algorithms)
  - includes important HPO and thus measures SW readiness/support
  - very hard to define target metric, e.g. what is the best network, best accuracy you can overall get, etc.

- energy/sample for inference workloads

Time-to-solution is challenging



Goyal et al. [arXiv:1706.02677](https://arxiv.org/abs/1706.02677)

Well designed benchmarks

- relevance: use state-of-the-art models/building blocks (DL community is very swift!)
- capacity (HPO), capability (batch-, domain/model-parallel training) and hybrid workloads
- measure IO performance of the file system
  - cover a variety of different input file and data formats
  - stress-test modern file system features (e.g. BurstBuffer, node-local NVMe, etc.)
- architectural coverage?
  - modern models are too big to fit into RAM of old GPUs and model/domain parallel frameworks are not very common
- framework-agnostic?
  - landscape changes quickly, enforce open exchange formats (ONNX)?
- define HPO selection guidelines/table for arbitrary batch size along with target score numbers on reference architectures
- DL training is non-deterministic: include tolerance/CI for scoring metrics

Thank You