Scale factors and calibration tools in CMS [for nanoAOD]

Maria D'Alfonso (MIT) & Loukas Gouskos (CERN)
for the Cross POG group(*)

HSF Data analysis WG
January 19, 2021

(*) Physics Office for the coordination of activities related to development, maintenance and use of analysis data-formats

Overview of the CMS data-tiers

RAW (~1MB/ev)
- Full event information directly from T0 containing “raw” detector info
- Not used for analysis

↓

RECO (~2-3MB/ev)
- **RECO**nstructed data; contains physics objects with many details stored
- Mainly for low-level developments

↓

AOD (~500kb/ev)
- **A**nalysis **O**bject **D**ata: a subset of RECO. Used for physics analyses in Run 1
- Run2: Searches w/ non-std signatures

↓

mini-AOD (~50kb/ev)
- Default data-tier for the Run 2 analyses
- Covers ~95% of CMS analyses

↓

nano-AOD (~1-5kb/ev)
- New (i.e., ~2017) development
- Used in a handful of Run 2 analyses
- Target: cover>50% of the CMS analyses

MiniAOD based analysis workflow

- Based on the CMS Event Data Model (EDM)/ requires CMSSW
  - Rich event content (~50kb/evt)
    - High-level physics objects with all necessary info for object development and calibration
    - Full list of PF candidates; most tracks and generated particles
    - All trigger objects and bits, generator level information: LHE weights, etc..
  - Then: analysis groups develop a framework to process miniAOD and produce flat ROOT trees

- Cons:
  - Duplication of effort & resources
    - multiple sets of ntuples with very similar content
  - code maintainability
  - Stress in computing infrastructure during rush periods

- Clearly, this approach is not sustainable in the long term

NanoAOD

- A light-weight O(kb/evt) flat root ntuple read by bare-root
- Content:
  - High-level physics objects and precompute variables/IDs/etc.. [instead of storing the necessary inputs]
  - All trigger bits
  - Subset of GEN particles [hard scatter, leptons, heavy flavour] and a selected set of LHE/GEN weights [lots of effort to improve content]
  - Store all variables with reduce precision
- Drop:
  - PF Candidates and tracks [stored in miniAOD]
  - Detector-level info [calo cells, rechits, etc…]
- Initial goal: cover 30-50% of CMS Physics analyses
  - A more aggressive target [~80-90% of the analyses] seems feasible
    - Some [limited] room for increase in size/evt if adopted by many analyses
    - Huge gain in computing resources

Analysis Tools

- A few frameworks/libraries [e.g., NanoAOD-Tools] that collect tools to assist the analyzers
  - Modular and flexible
    - python-based modules [but C++ implementations also supported for computationally expensive tasks]
    - Multiple modules runs can be run in one go
- In a nutshell:
  - Gen-level corrections: LHE, PDF weights
    - cross sections are stored in db
  - Non-event data: e.g., trigger prescales, LHC info, ..
  - Tools to apply physics object calibrations, evt-level weights, …
    - event-level corrections/SF; more involved corrections are propagated at the NanoAOD production stage using Global Tags
  - Routines to evaluate systematic uncertainties
  - Tools for skimming and/or pruning of the output content
  - Functions to compute complex variables; shared among analyses
  - Grid submission tools
- Main idea: centrally develop & validate tools common in most analyses

Analysis Tools: Calibrations & SF

- One of the main efforts currently in the group
  - Improve physics object calibration workflow
    - produce needed samples -> run calibration analysis -> Scale Factors
      - sample production takes significant fraction of the whole process
  - Bookkeeping and application of SF
    - Tools in place, yet room for improvement:
      - Unify across different groups, extend functionality
      - More versatile:
        - Simple and decoupled from official CMS software
        - support both C++ and python
      - Profit from new tools and technologies
        - Functions that works for both **row** and **columnar** type implementations
      - Improve analysis preservation

NanoAOD: “custom NanoAOD”

- NanoAOD data-tier has lots of flexibility
  - Develop custom nanoAOD workflows for **very** specific cases with tailored event content

- **NanoAOD workflow for jet calibration:**
  - Designed to aid the calibration workflow of jets
    - Sample production [started from heavier data-tiers] results to a very significant fraction of the total calibration procedure
    - Yet: keep evt/size under control: ~5-6 kb/evt on a limited set of samples
  - Derive JEC/JER, SF for taggers/puid/q-g..

Analysis Tools: Calibration tools

- Each physics-object group provides the necessary ingredients to propagate the corrections to the analysis
  - Values collected in different data-formats
    - json
    - CSV
    - root files (TH1, TH2, Tformula)
    - databases, Global tags

  - Use a common format across all groups:
    - Develop a JSON format using a centralize schema
      - clear and attractive format
      - easy to navigate and implement
      - self-documenting
      - Developed outside LHC and HEP: lots of support and tools to aid the analyzer

  - JSON files stored in a central area:
    - “write-once” mode in cvmfs [final decision pending]
    - easy access, allows versioning..

Calibration Tools: Common JSON

- Example:

Self-documenting

+ Tag of the conditions “Vxx”

```
{
    "schema_version": 1,
    "corrections": [
        {
            "name": "EIDISO_WH_out",
            "description": "An electron scale factor",
            "version": 1,
            "inputs": [
                { "name": "eta", "type": "real" },
                { "name": "pt", "type": "real" },
                { "name": "systematic", "type": "string" }
            ],
            "output": { "name": "weight", "type": "real" },
            "data": {
                "nodetype": "binning",
                "edges": [ -2.5, -2.17, -1.8, -1.57, -1.44, -0.8, 0.0, 0.8, 1.44, 1.57, 1.8, 2.17, 2.5 ],
                "content": [
                    {
                        "nodetype": "binning",
                        "edges": [ 25.0, 27.0, 30.0, 32.0, 35.0, 40.0, 50.0, 200.0 ],
                        "content": [
                            {
                                "nodetype": "category",
                                "keys": [ "nominal", "up", "down" ],
                                "content": [ 0.903, 0.9540000000000001, 0.852 ]
                            },
                            {
                                "nodetype": "category",
                                "keys": [ "nominal", "up", "down" ],
                                "content": [ 0.921, 0.9630000000000001, 0.879 ]
                            },
                            {
                                "nodetype": "category",
                                "keys": [ "nominal", "up", "down" ],
                                "content": [ 0.924, 0.9550000000000001, 0.893 ]
                            },
                            {
                                "nodetype": "category",
                                "keys": [ "nominal", "up", "down" ],
                                "content": [ 0.926, 0.9470000000000001, 0.905 ]
                            },
```

Can be extended to event-level corrections
– top/W/Z-p_T reweighting
– ISR jet reweighting …

Fermilab

7 4 Nov 2020 Nick Smith I corrections.json

Analysis Tools: Application of SF

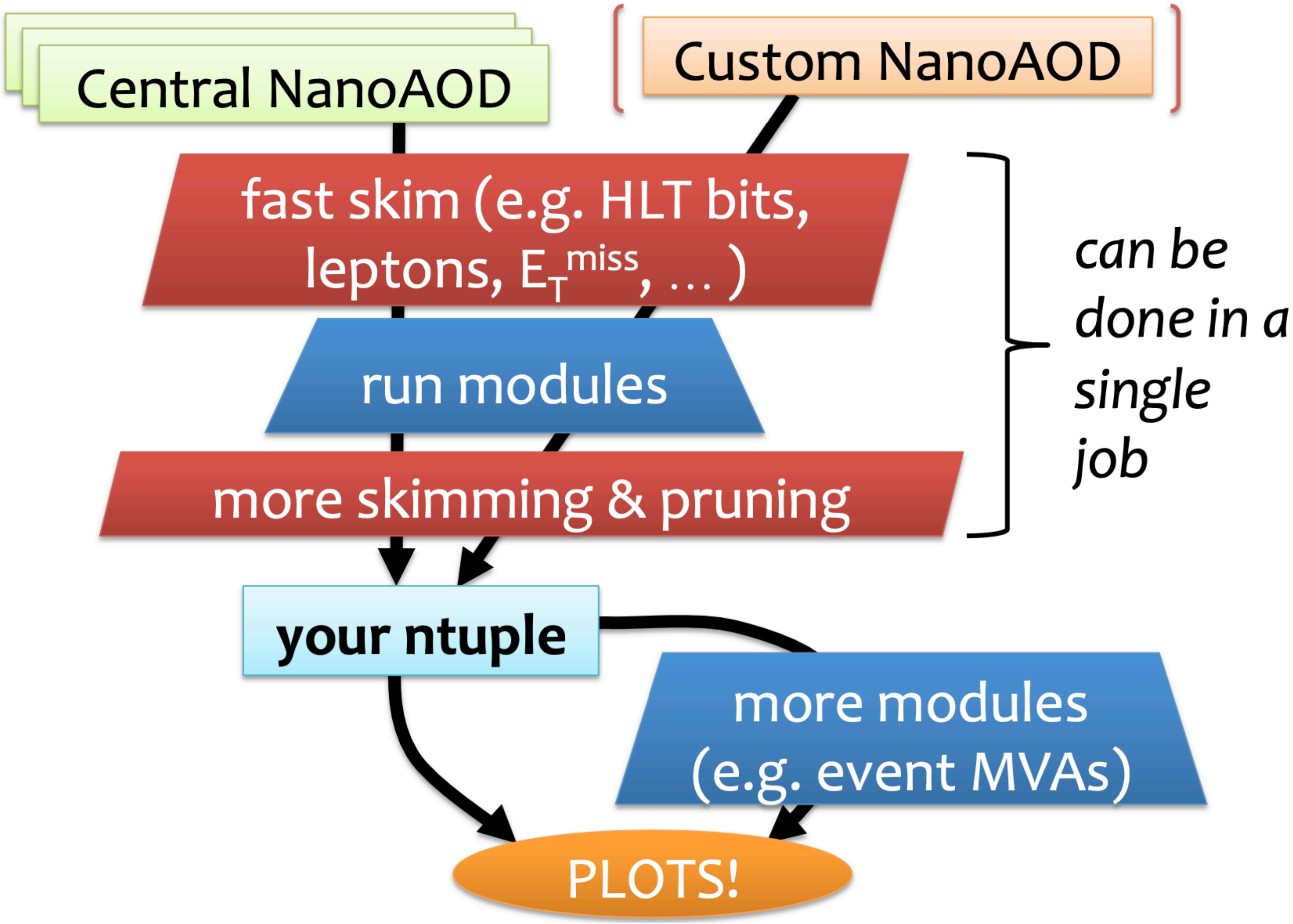
- Based on simple and flexible functions
  - avoid duplication of code/effort
  - centrally maintained
    - less error-prong, broader feedback, share expertise
    - Physics object groups share the same functions
  - Decouple from official CMS software and NanoAOD-Tools
    - Support both C++ and python based analysis codes
    - Traditional (row) and more recent (columnar) analysis frameworks: RootDataFrame, Coffea, …

Summary

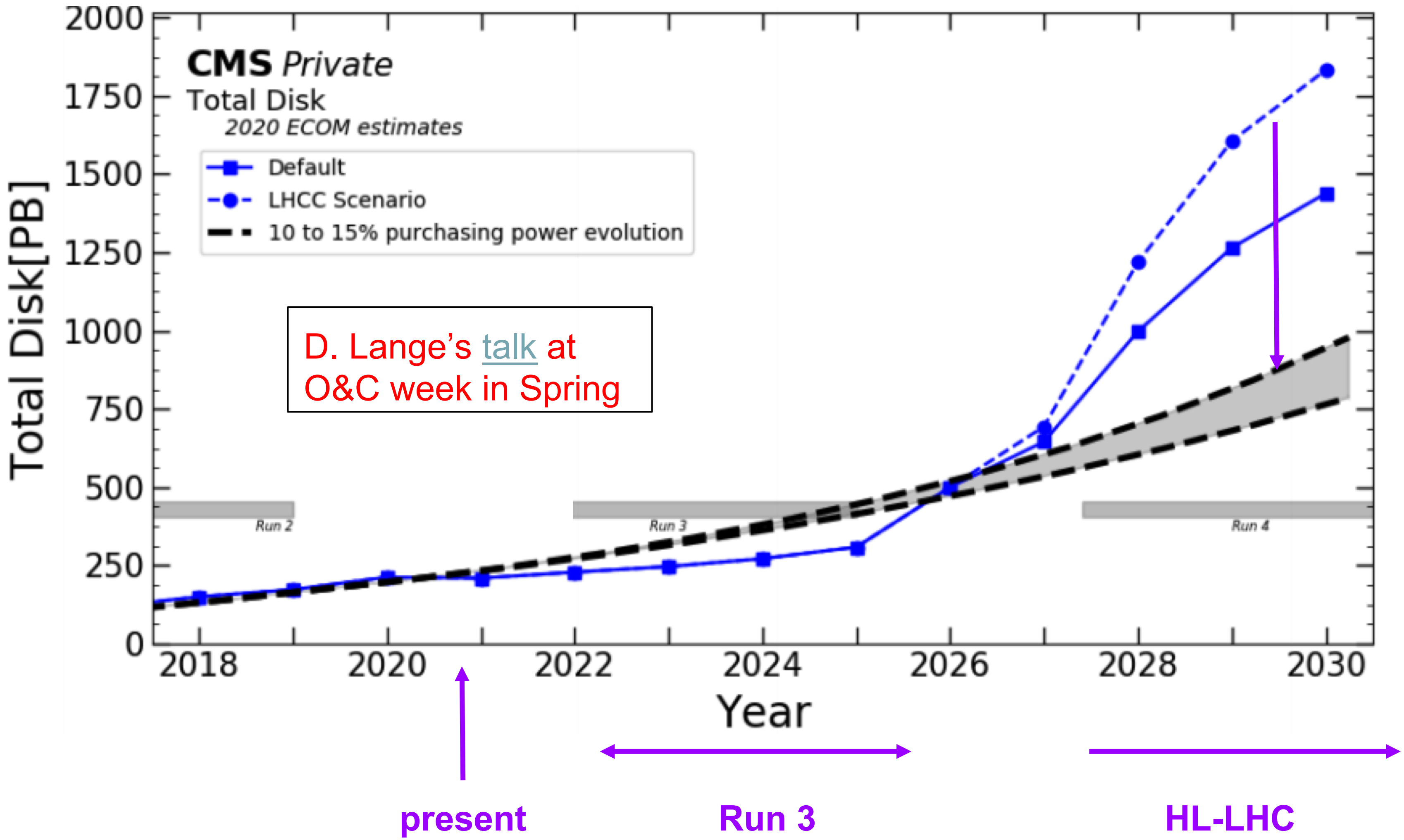
- LHC LS 2: period to improve based on the Run 1 & 2 experience
  - Ensure scalability of the CMS analysis format in Run 3 and beyond
  - Streamline operations, avoid duplication of code, share the load
    - More data -> increased complexity [e.g., year-dependent corections…]
    - Improving in these areas -> More time to produce important physics results

- CMS NanoAOD(+ Tools) has great potential
  - One of the main priorities is to improve the physics object calibration workflow and the tools to propagate the corrections
  - Synergies and exchange of experience between experiments very useful

NanoAOD-based workflow



Current and foreseen disk needs

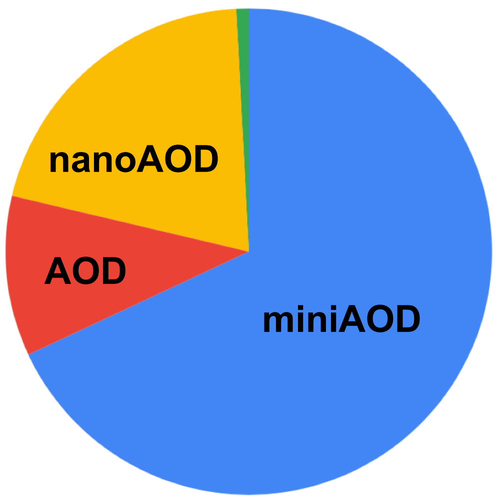


- Current situation not sustainable in the future
  - CMS Physics analyses must move to light-weight data-tiers

Data-tiers used in CMS analyses

- Using inputs collected ~year ago

Physics analyses in Run 2



- **AOD used in analysis with exotic signatures and/or precision physics**
- **Why not NanoAOD?**
  (a) Physics content
  (b) Missing SF/unc and functionalities in NanoAOD-Tools
  (a) Delays in nanoAOD production + analysis fwk already setup for miniAOD

Plan on using NanoAOD in Run 3: