

Just enough MadGraph to be dangerous.

Low-multiplicity dark signal modes generation

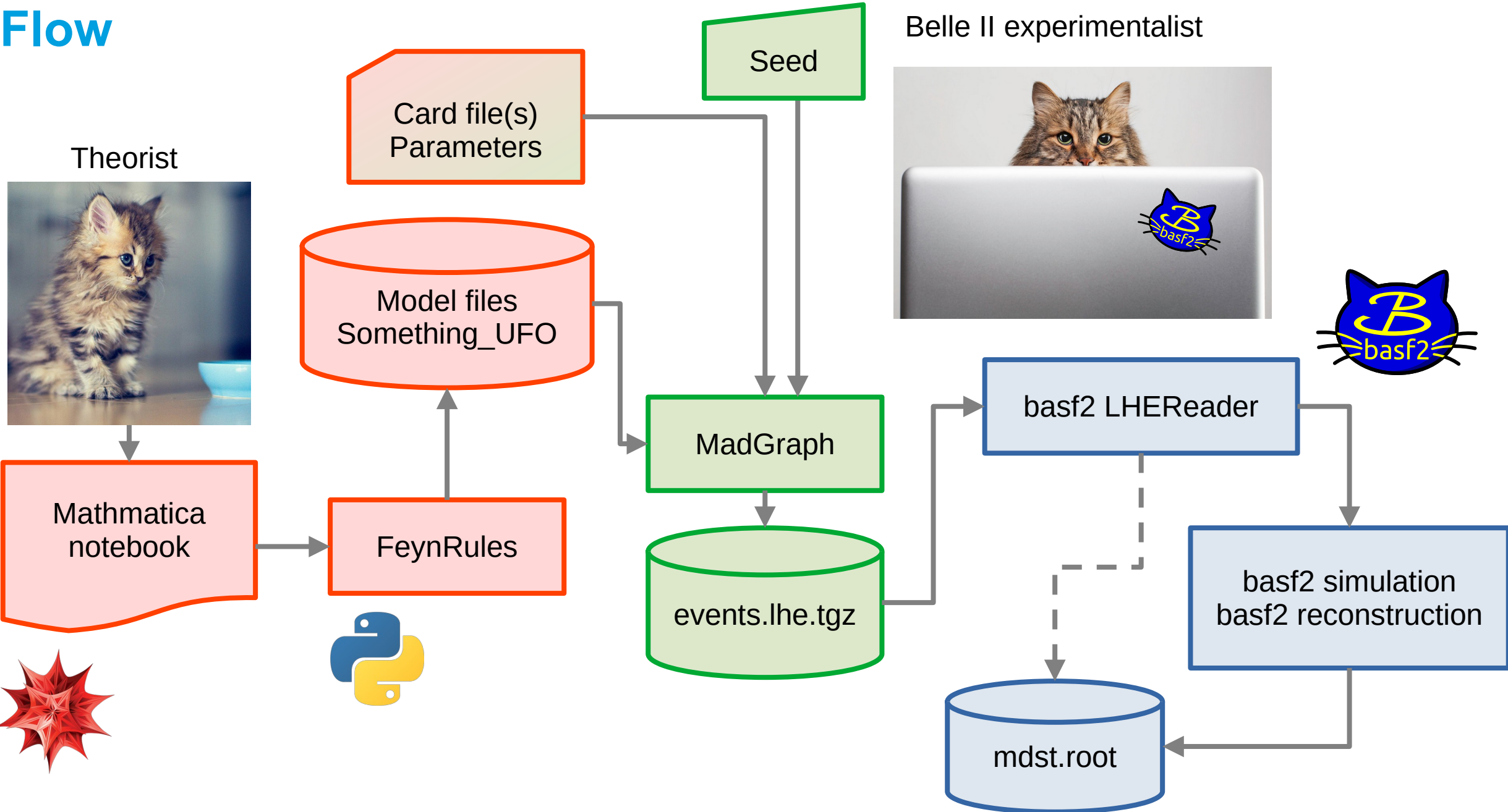
Sam

Hamburg – Karlsruhe – Victoria, 27.07.2021

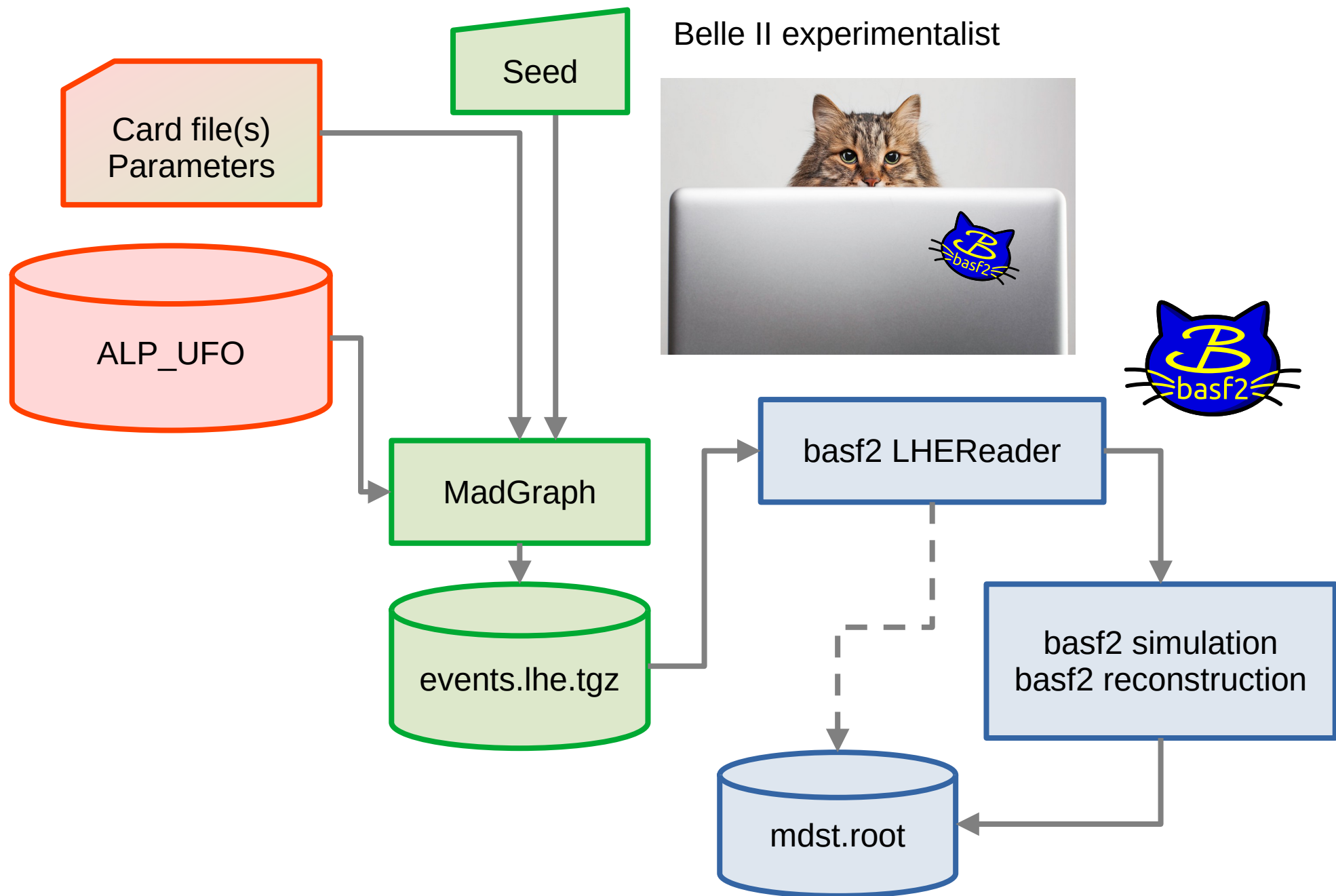
MadGraph

- An event generator [included in basf2 as an external](#) software.
- Has a [terrible website](#) which does not use https (I'll remind you it's 2021).
- Code hosted on [launchpad](#) (bug reports there too).
 - Also various model files kicking around. Some in [basf2/generators/madgraph](#) some in [B2DARK](#), many on the internet.
- Cite: “*The automated computation of tree-level and next-to-leading order...*”
[[JHEP07\(2014\)079](#)]
- tl;dr:
Does QFT matrix element calculations for you and generates particles and their 4-momenta (at a given collision process/energy).

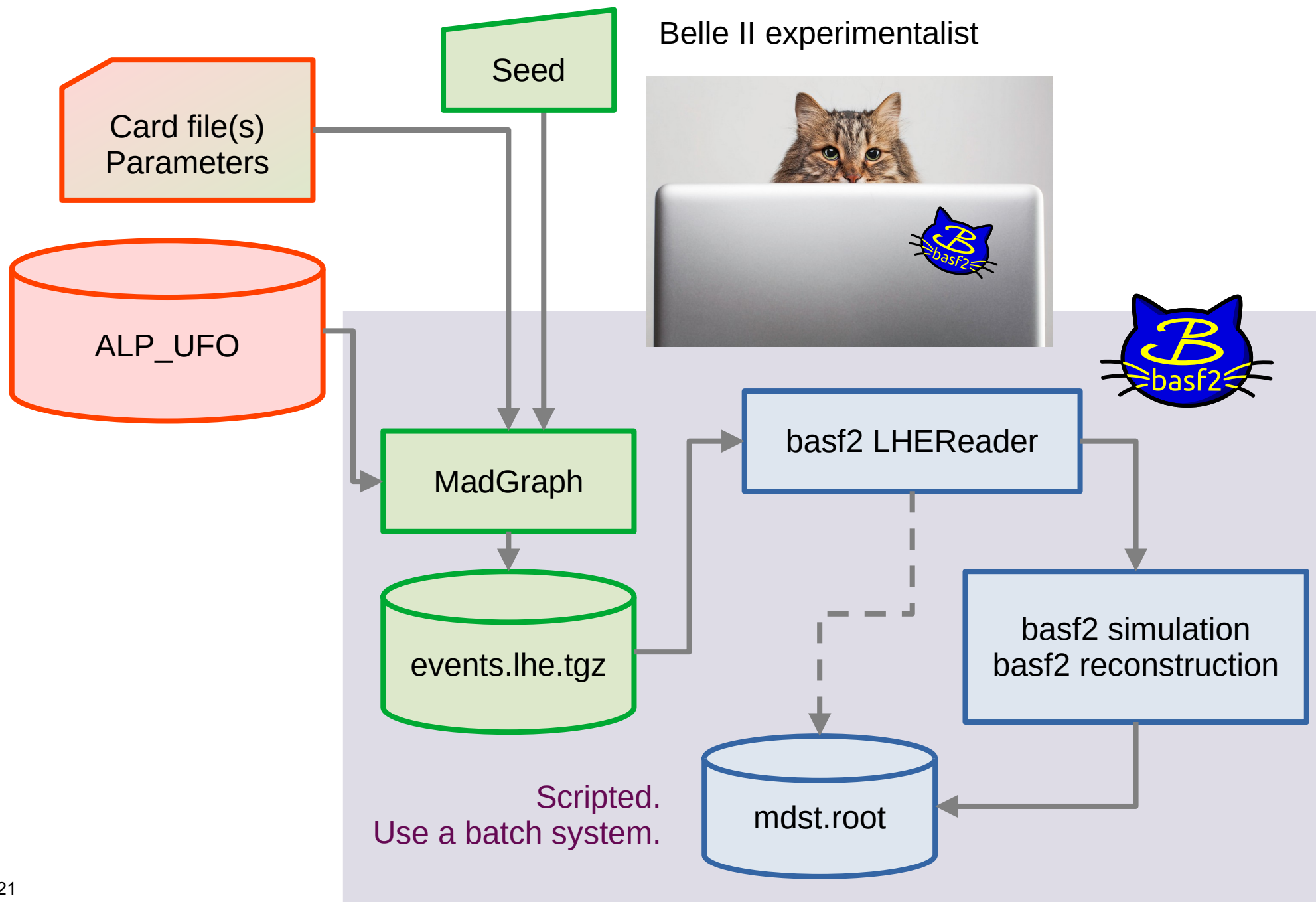
Flow



Flow



Flow



Things

- UFO stands for “Universal FeynRules Output” [[arXiv:1108.2040](#)]
- LHE stands for “Les Houches Events” (it’s just an event file format) [[arXiv:hep-ph/0609017](#)]
 - A better reference is [this documentation from some other generator](#) (scroll to the section titled: “**The <event> block**”).
 - It’s “just” XML: you can read what is happening.
- There is an extra piece. The ISR plugin [[arXiv:1804.00125](#)] [[github](#)].
 - Sometimes we have “fun” incompatibility problems.
 - I believe it stops working in release-06. (But you should check).
 - This modifies the lineshape to account for ISR.
But! It doesn’t add a photon particle.

The <event> block

Each event block contains the full description of a single event, which basically consists of a list of produced particles (with their corresponding four-momentum vectors etc). An example event might look like this:

```
<event>
  3  0  1.  0.  0.  0.
  2212  0  0  0  0  0 -0.264  0.275  3.468  3.613  0.938  0.  9.
  2112  0  0  0  0  0  0.267 -0.052  0.138  0.986  0.938  0.  9.
  211  0  0  0  0  0 -0.003 -0.222  0.730  0.776  0.138  0.  9.
</event>
```

The first line in the <event> block has a special meaning. The columns of these first lines have the following meaning:

- 1: the number of particles (N)
- 2: (unused in GiBUU)
- 3: the weight (w) of the event
- 4-6: (unused in GiBUU)

For real events, the weight w equals 1 (i.e. all events have the same weighting). For perturbative events, on the other hand, the weight is usually related to the total cross section for the process (and can be different for each event).

Each of the following lines represents one particle. The columns of these lines have the following meaning:

- 1: ID code of the particle (in [PDG numbering scheme](#), which differs from our internal [GiBUU numbering scheme](#)). This number determines the identity and charge of the particle.
- 2-6: Unused (always 0).
- 7-9: 3-momentum (px, py, pz) of the particle in GeV. Usually the z-axis corresponds to the beam direction.
- 10: Energy E of the particle in GeV.
- 11: Mass m of the particle in GeV.
- 12: Unused (0).
- 13: Spin of the particle ('9'=unknown).

The example event shown above contains a three-body final state, consisting of a proton, neutron and pi+ (produced in a pp collision).

Taken from [here](#).

Batch and pragmatics

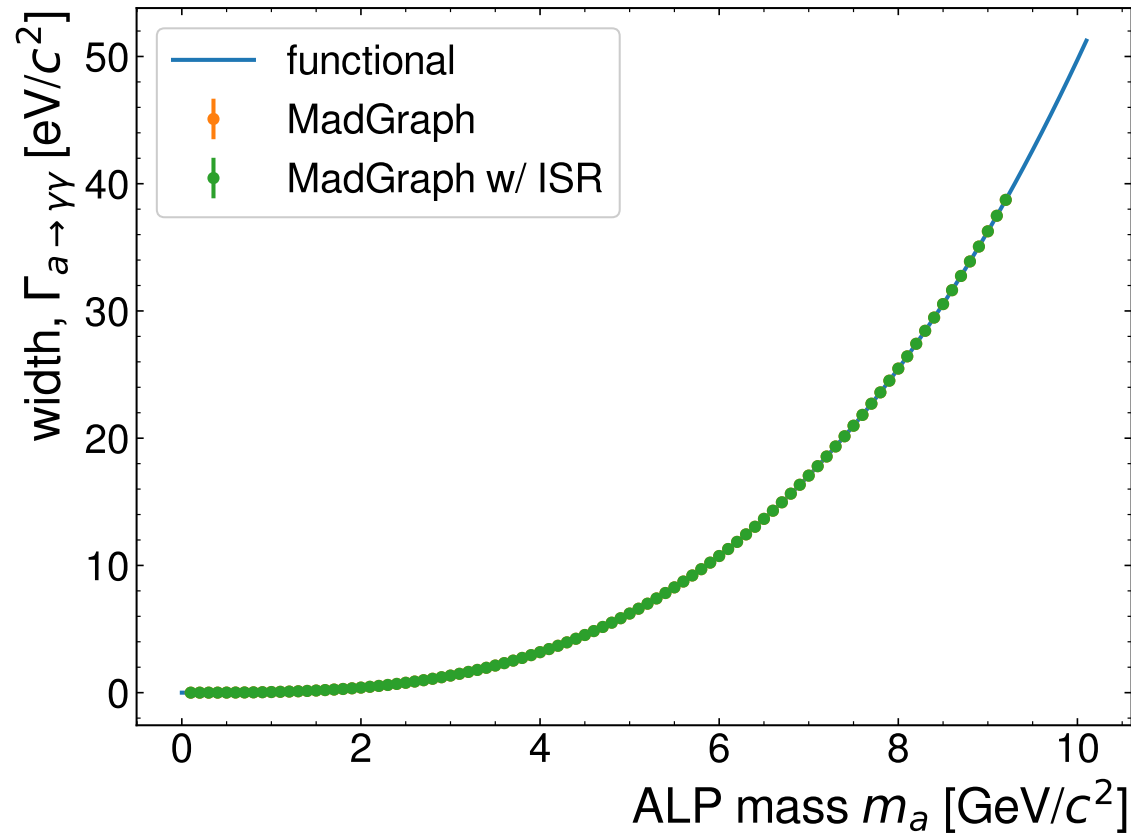
- You might see a lot few ways to script things up.
- *I* use a standalone MadGraph+ISR plugin then setup a specific basf2 version (following the simulation/reconstruction conditions of an MC campaign).
 - The generation is fast, so can be done interactively even (or in a batch).
 - The basf2 steps, I send to HTCondor @ DESY NAF.
- **Torben** has python scripts that call the MadGraph executable, untar the output, and then run simulation/reconstruction (since basf2 is just python).
- **Savino** has even more complicated python scripts because he has a lot of model parameter.
- This is screaming for someone to make a nice workflow thing in b2luigi.
- By the time I realised b2luigi was so cool, already had a handcooked setup working.

Misc. tips

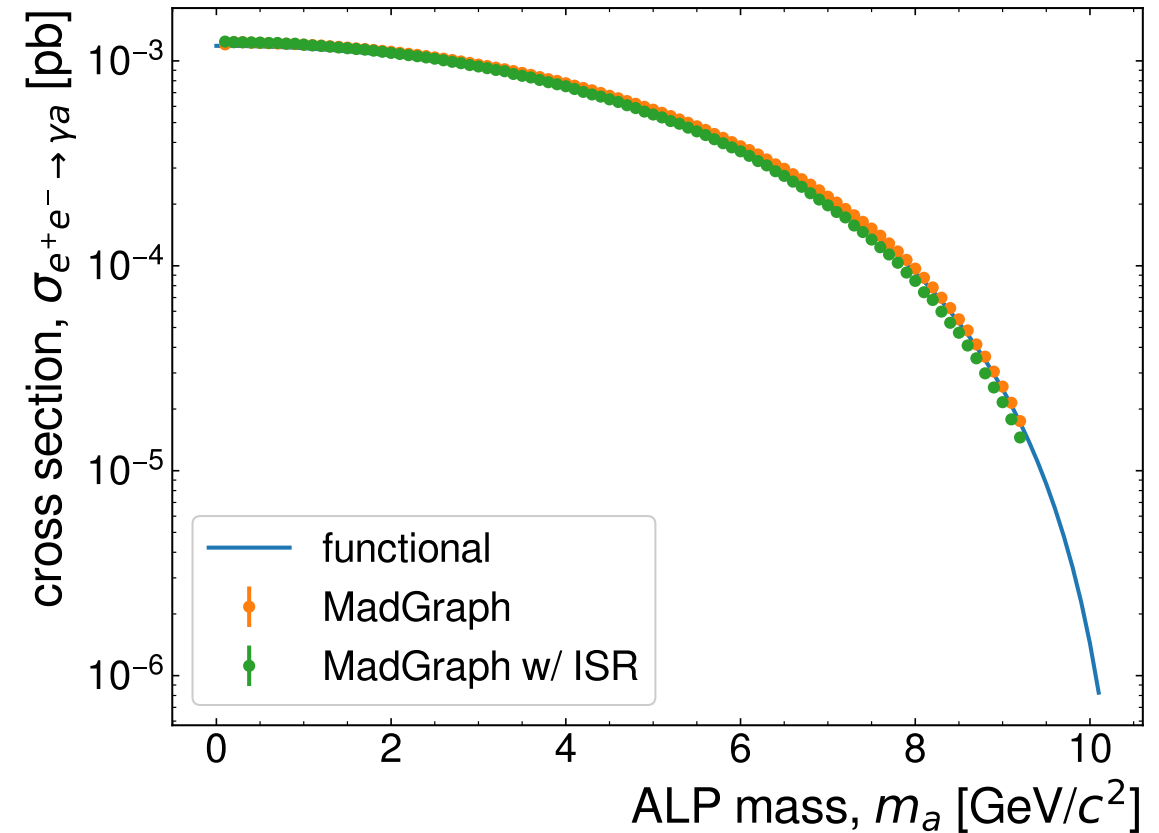
- Look at the Feynman diagrams and check they make sense.
- Eyeball-scan a few events in the LHE file.
- Read the model paper and see if they give you some helpful/testable formulae.
 - For the ALP analysis, they did.
- Run a gen-level check before running sim and reco.
- Set different run numbers for your runs.
- Set different run numbers for your runs.
- Set different run numbers for your runs.

Examples of some sanity tests.

ALP-model-specific checks

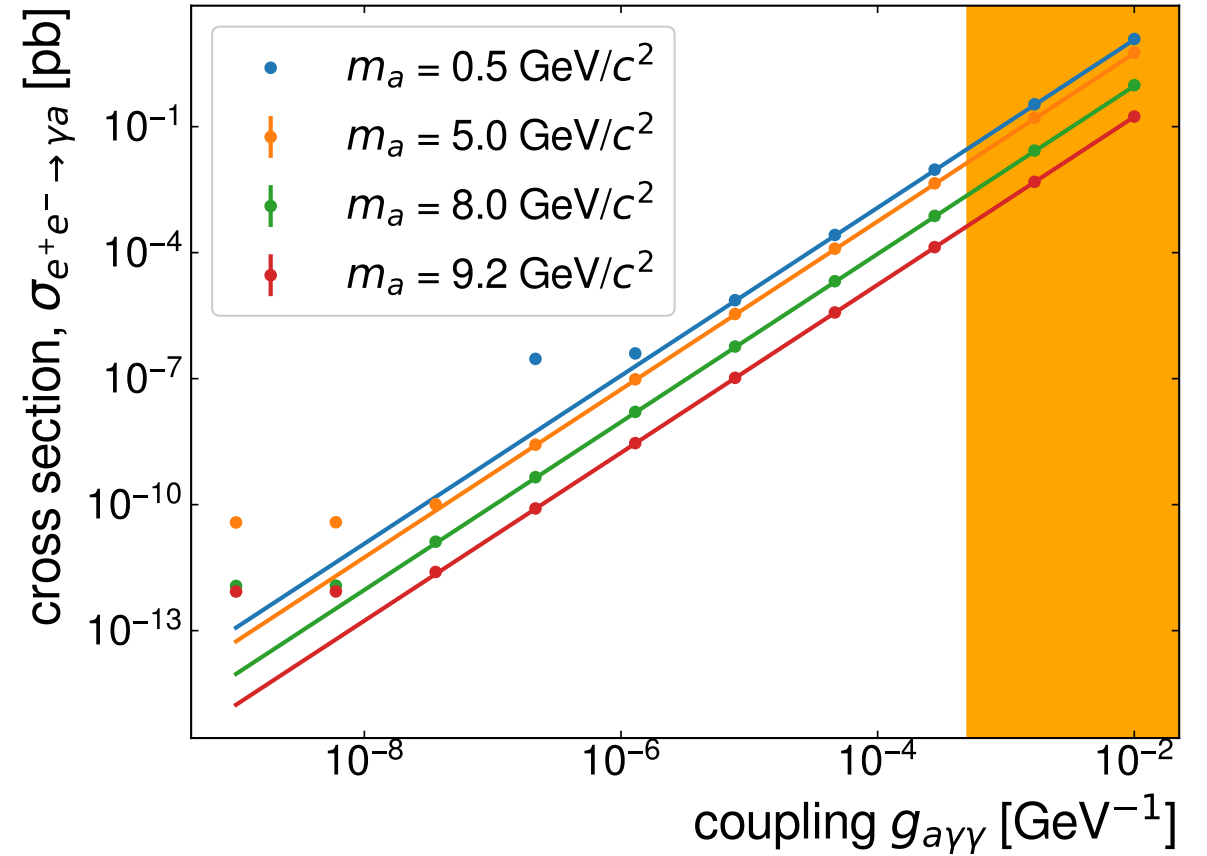
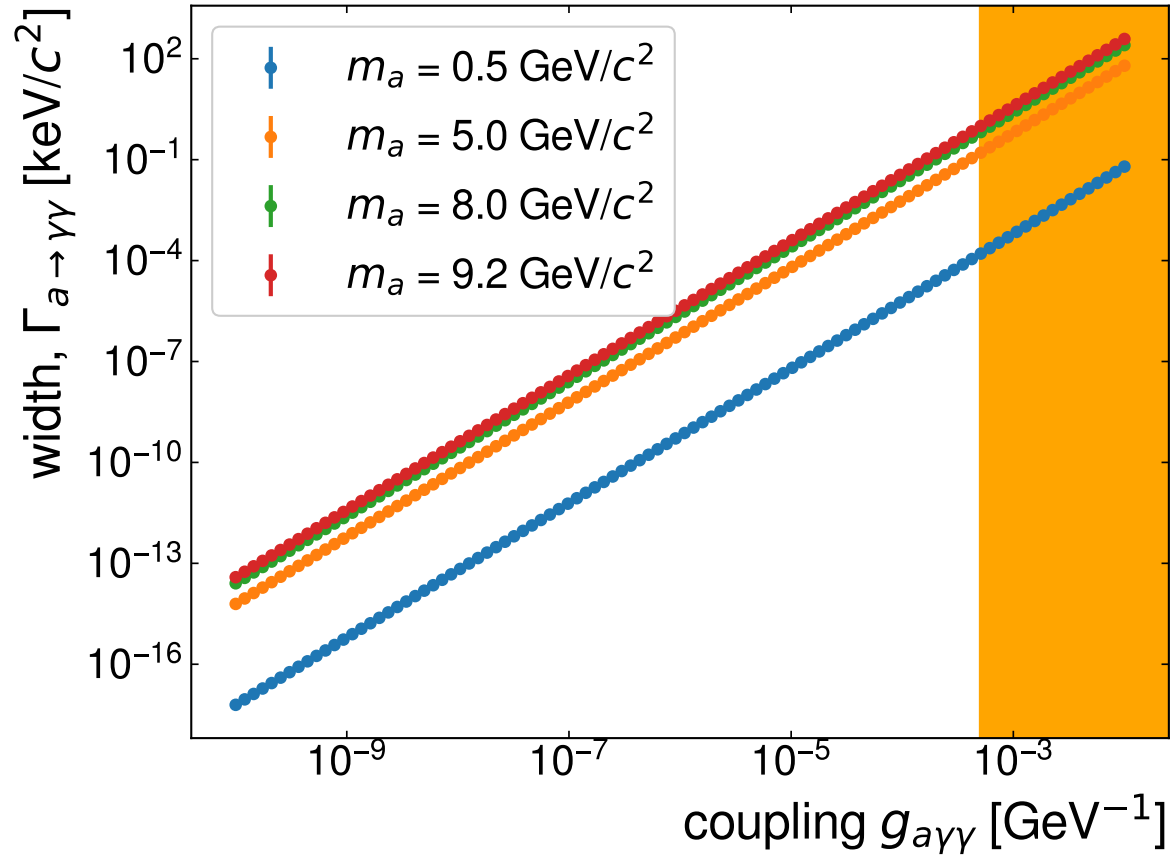


$$\Gamma_a = \tau_a^{-1} = \frac{g_{a\gamma\gamma}^2 m_a^3}{64\pi}$$



$$\sigma_{e^+e^- \rightarrow a\gamma} = \frac{g_{a\gamma\gamma}^2 \alpha}{24} \left(1 - \frac{m_a^2}{s}\right)^3$$

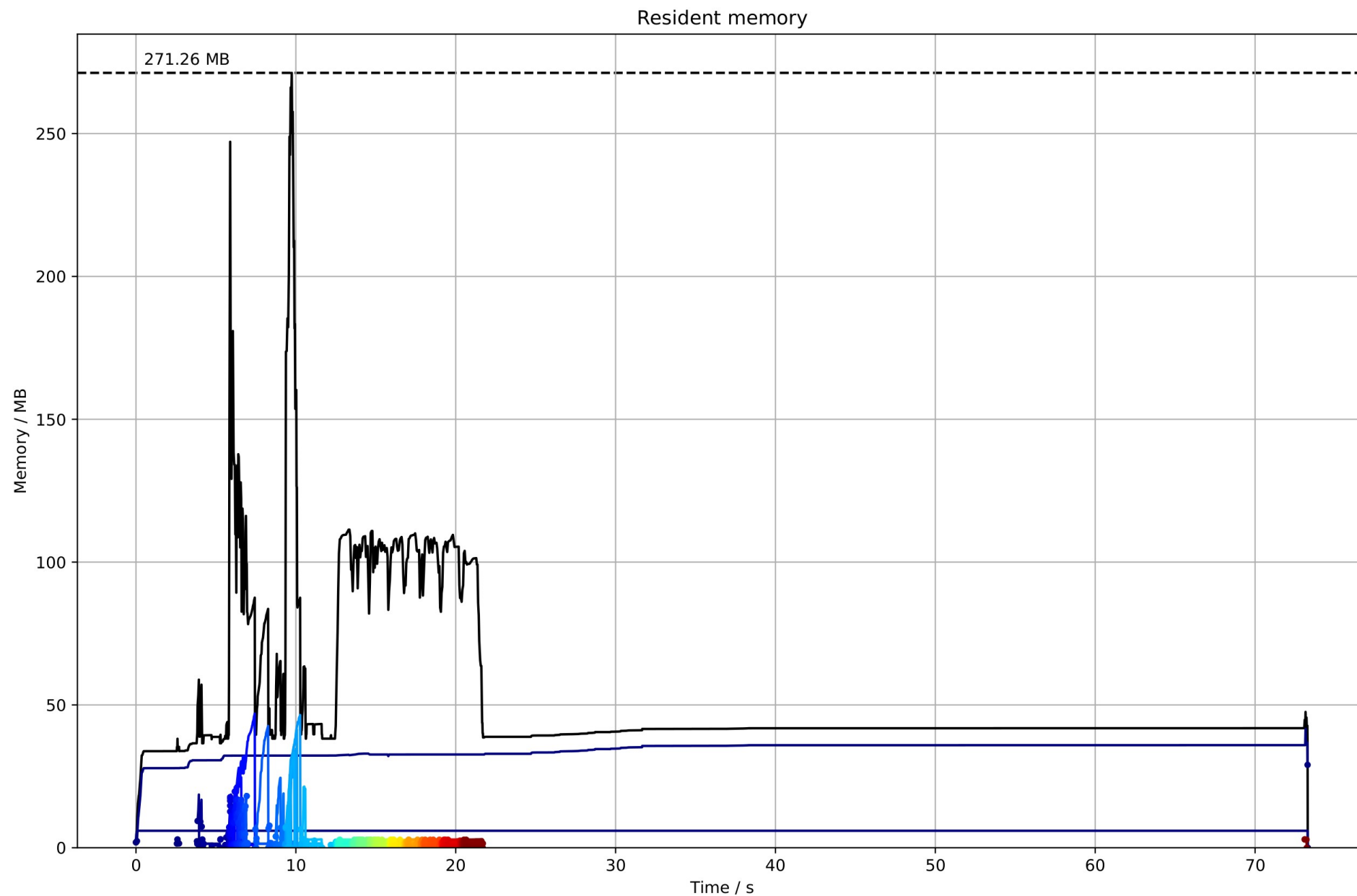
ALP-model-specific checks



FYI. MadGraph memory usage.

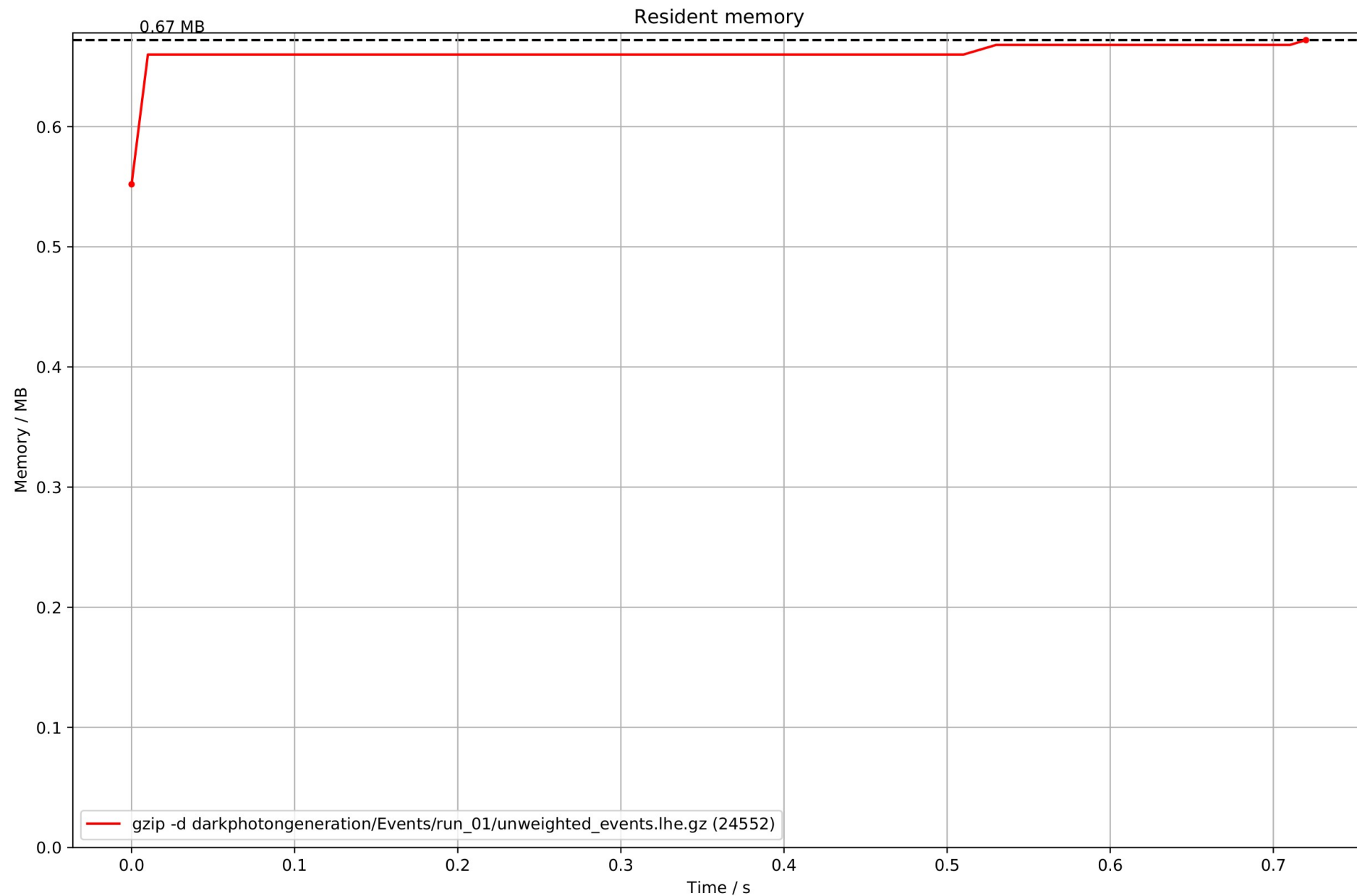
MadGraph memory

Generating 100k dark
photon events with
Madgraph v2.7.2 in
the externals
(no ISR plugin)



Unzipping LHE files

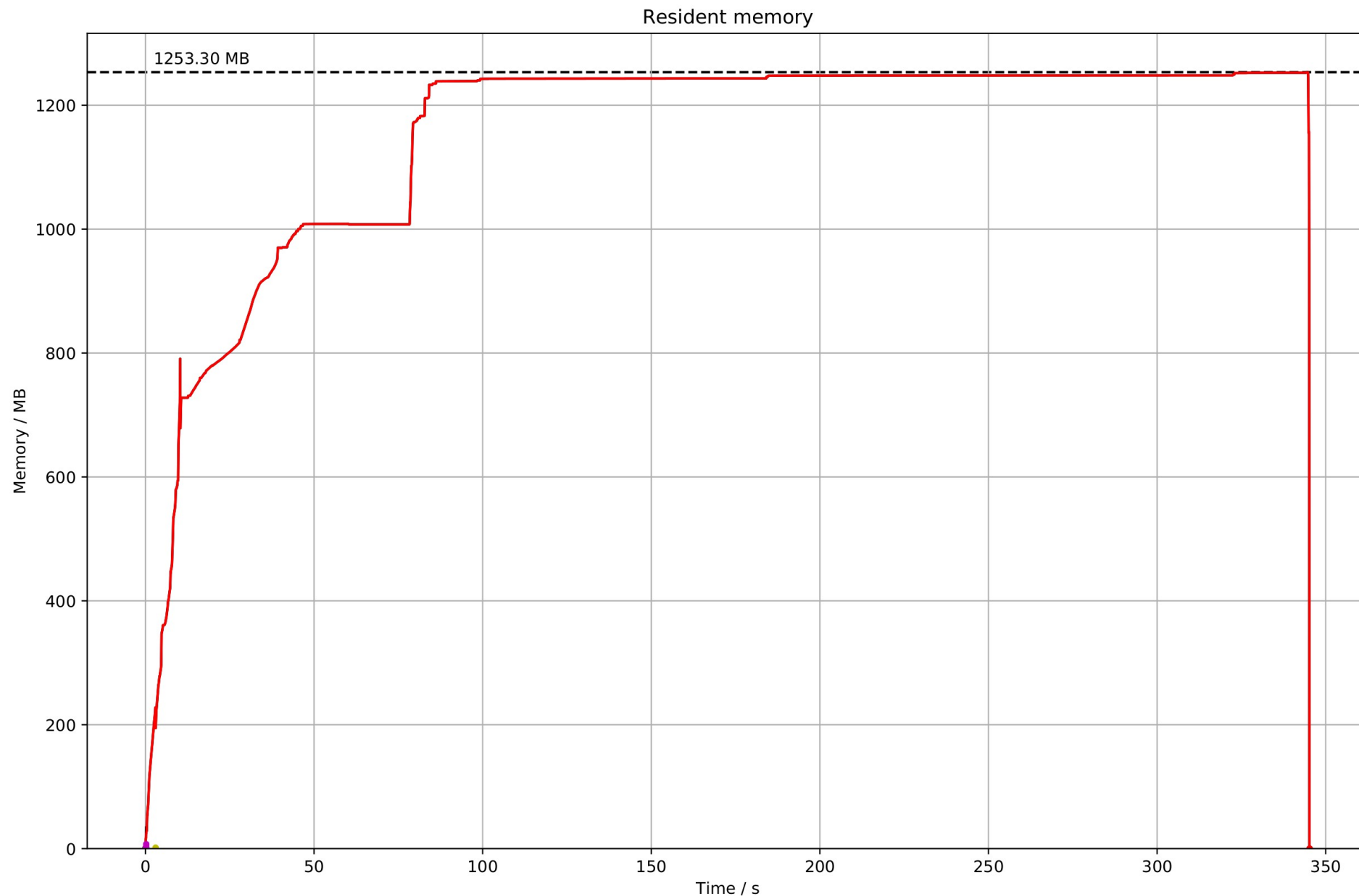
Also not a problem



basf2 sim. & reco

Reading in LHE file,
running normal
processing

(1k events)



Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Sam Cunliffe

sam.cunliffe@desy.de

orcid: [0000-0003-0167-8641](https://orcid.org/0000-0003-0167-8641)