Support for displaced signatures in RIVET

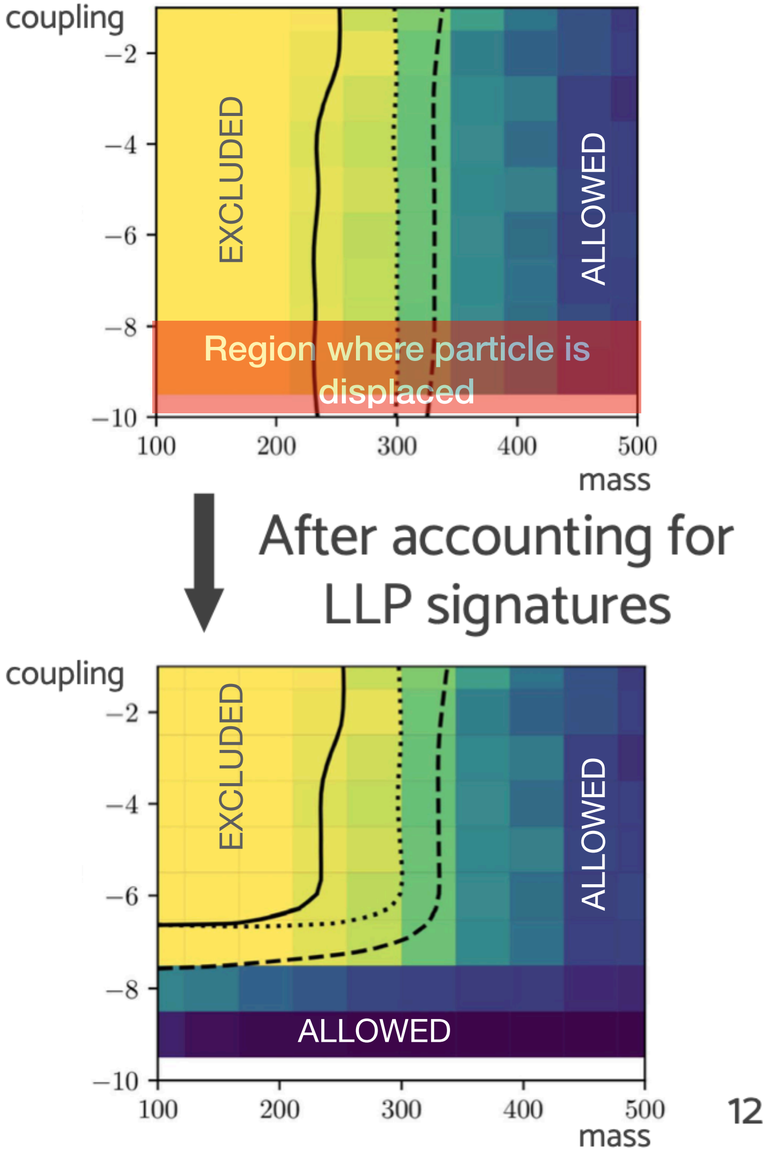
Louie Corpe (UCA/LPCA), MC Support workshop, 2 April 2025

What's the problem?

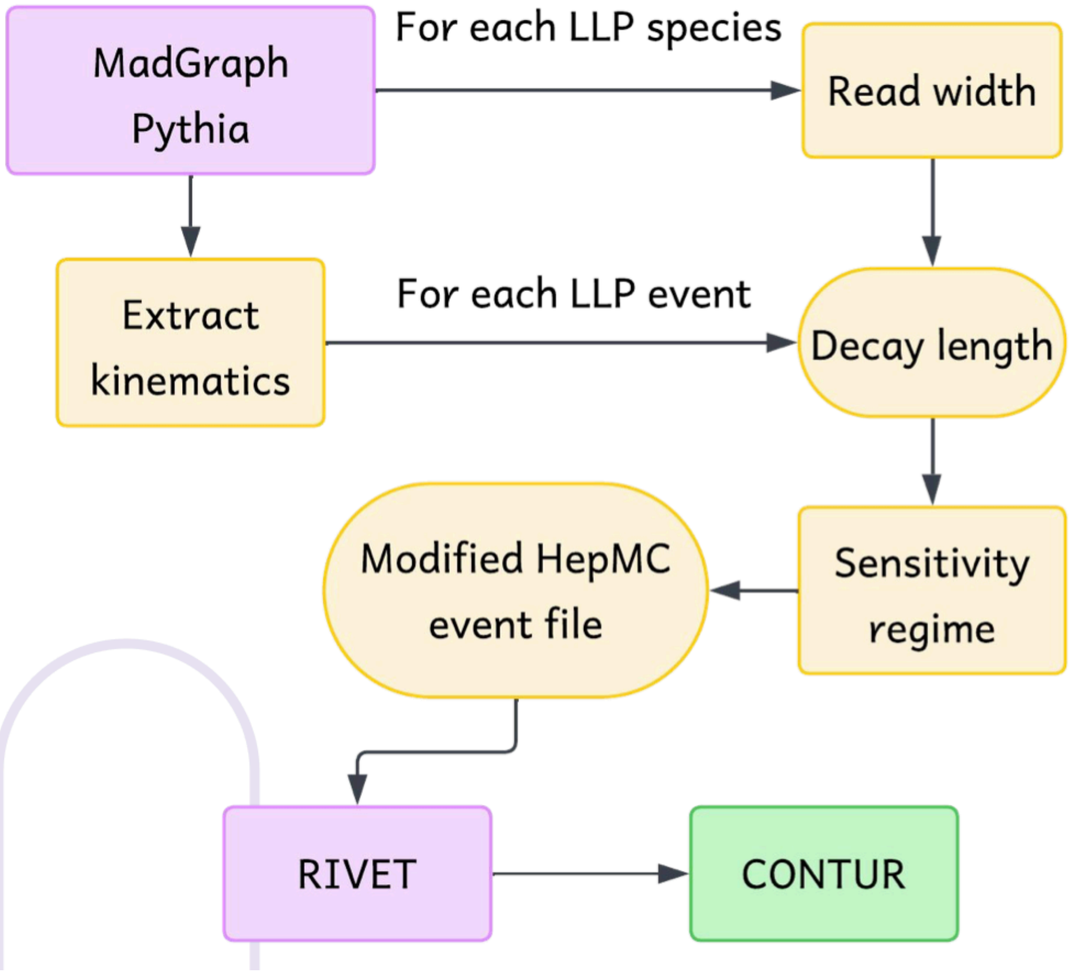
- RIVET assumes prompt behaviour for particles which it extracts from HEPMC
  - For SM particles which are known to have long lifetimes: restricted according to their average decay length? (RIVET devs can confirm?)
  - For BSM particles: all assumed to be prompt
  - As we shall see, this is problematic for long-lived particles.
  - Indeed, in some parts of parameter space, BSM particles can acquire long lifetimes. How to handle those safely in Rivet? And in client tools like CONTUR ?
  - Need a variable primary particle definition

Visualising the problem

- RIVET is blind to BSM particle lifetime. Hence, selection efficiency (and hence exclusion) is currently agnostic to lifetime
- But in many models, various combinations of lifetime and phase space can lead to long lifetimes...
  - Clearly there exists a lifetime dependence for the selection efficiency of RIVET routines: depending on where the decay takes place, the same event could give MET, displaced activity in the detector, or regular prompt objects....
  - This means incorrect selection efficiencies from RIVET can give misleading/incorrect results for downstream tools
- More details presented by Simon Jeannot at the LHC Reinterpretation Workshop https://indico.cern.ch/event/1466101/contributions/6287843/attachments/3020577/5328992/LLP%20CONTUR.pdf



An ad-hoc solution



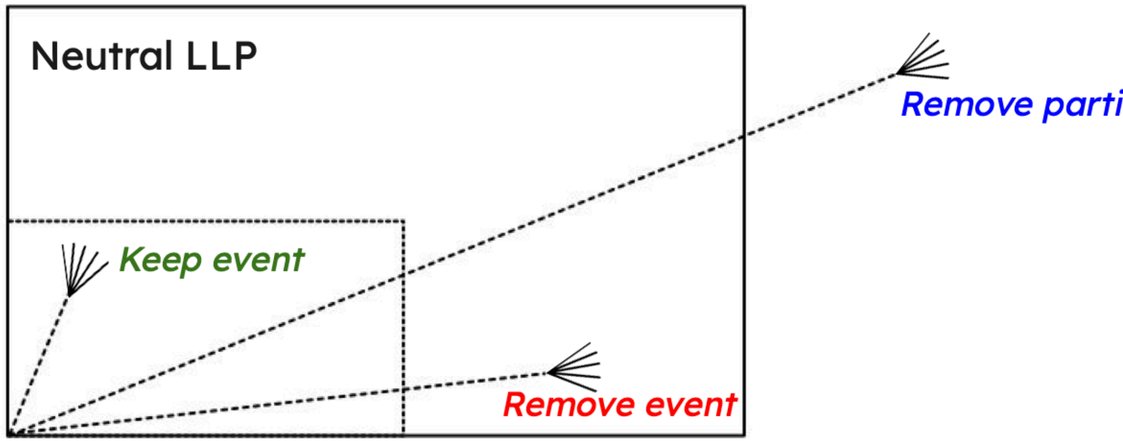
- An Ad-Hoc solution is to manually modify the HEPMC files before passing them to RIVET, such that the events containing LLPs can be removed or modified to be more realistic.
- All we need is the particle's decay position (accounting for exponential decay probability, boost/time dilation effects). Can calculate ad-hoc or get the event-generators give that to us.
- PyHEPMC module can do that for us. Have the possibility to remove individual particles (and all their descendants), or entire events
- But what algorithm to follow to handle these particles?

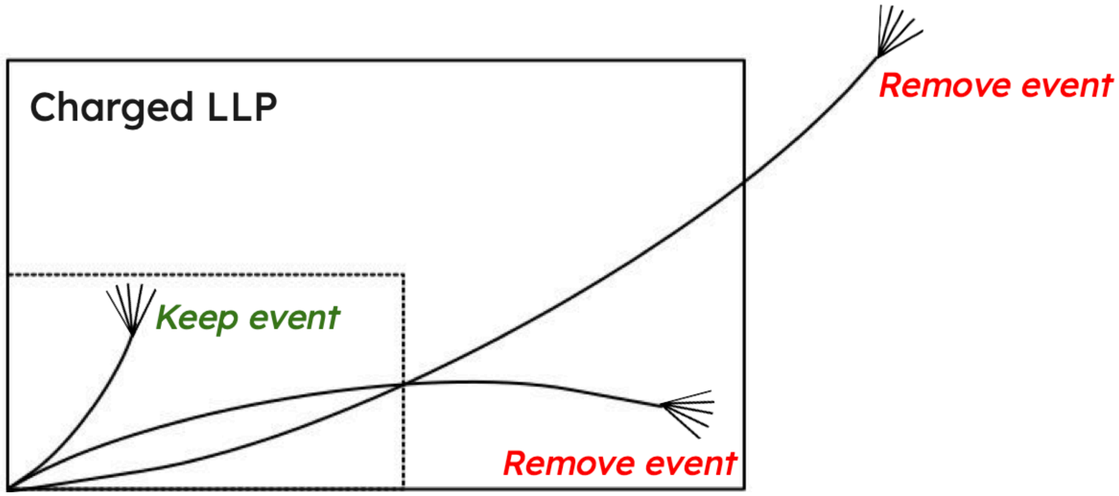
An ad-hoc solution

Low lifetime: extrapolation of the prompt sensitivity

Long lifetime: neutral LLP can yield to a recoil effect on the p_T of prompt particles which they are produced with

Intermediate regime: dip in sensitivity, regime which direct searches are designed to probe





When to keep event?

d0 cut at 0.5mm for both ATLAS and CMS is the limiting factor in track/ vertex reconstruction. Anything decaying with a transverse radius of **0.5mm** can be considered prompt at the LHC experiments

When to keep event but discard particle?

Neutral LLPs decaying outside of the detector will show up as MET. In that case all we need to do is remove that particle and all its descendants from the event

When to discard the whole event?

If a particle decays inside the detector, we have no general idea how detector (and unfolded measurements) would react. Same for a charged LLP decaying anywhere after d0 cut. Only choice is to remove the whole event

What radius to pick ? 20m to be safe?

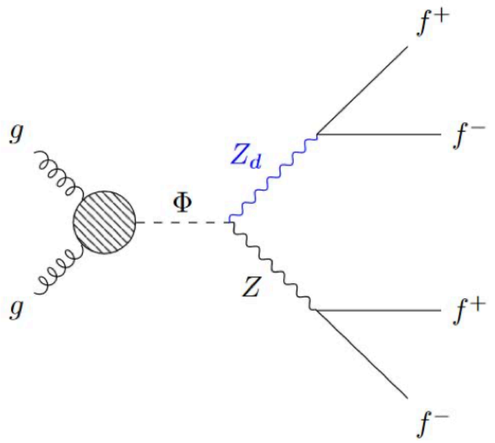
Ad Hoc-solution in action

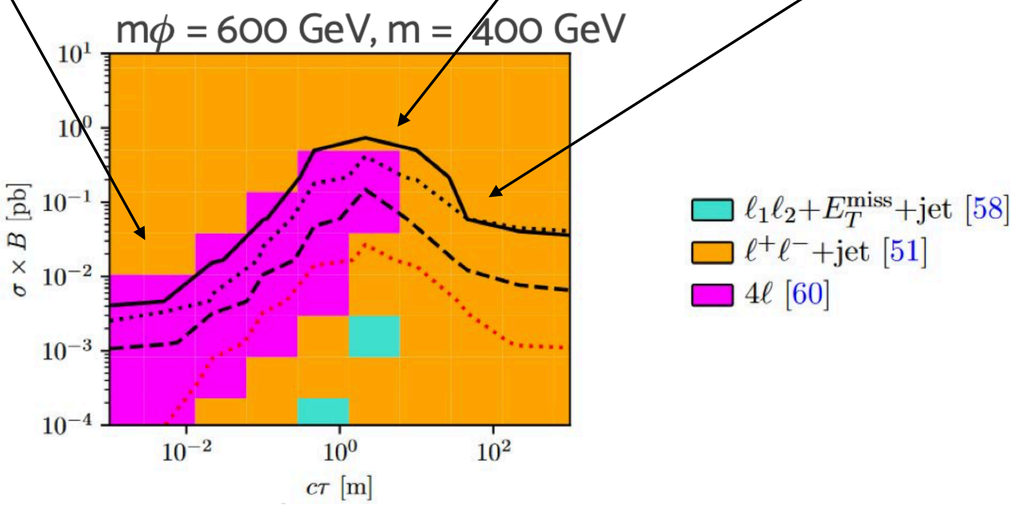
LLPs giving MET.
Sensitivity comes from recoil of prompt Z
in events which are kept

Sensitivity loss:
we throw the events away here
Because the decays happen in the detector

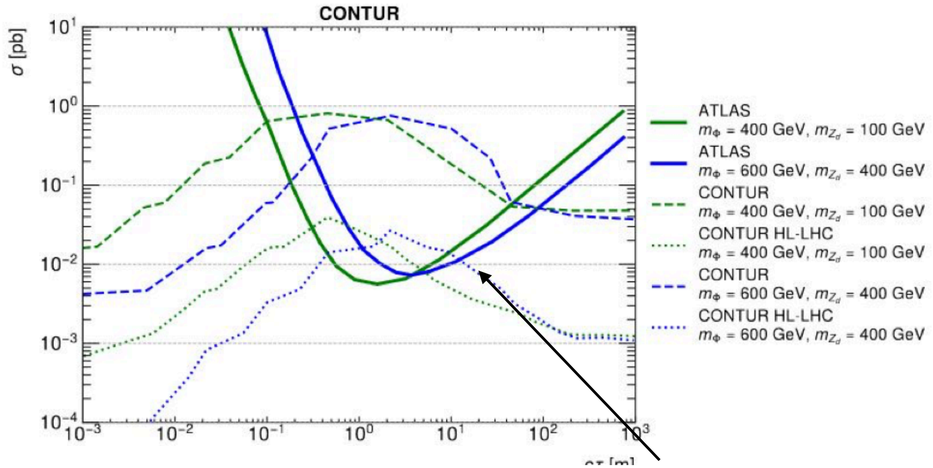
Many LLPs decaying before d0=0.5mm

2407.18710





Dark photon:



Competitive with direct searches... it's worth paying attention!

Towards a less artisanal solution

- How to make this work out of the box in RIVET?
- Some initial work around ParticleFinder class by Chris and Andy https://gitlab.com/hepcedar/rivet/-/issues/245#note_2385070787
- Would allow configurable options for ParticleFinder.
  - If we have access to the particle's creation vertex, should allow us to pick particles or not according to the decay position. Great!
  - What about vetoing entire events containing LLPs? How would this work?
  - Same outer radius for ATLAS, CMS? Other experiments ?
- Correct decay positions are not propagated by MAdGraph by default (need to explicitly add time_of_flight option). Can this be changed such that default behaviour is to return realistic decay position ?

Who is working on this

- Andy and Chris -- preliminary work on ParticleFinder class
- Louie Corpe with ε^2 FTE
- Simon Jeannot (masters student) with ε FTE
- Sophie McNeil (masters student) with O(1) FTE for 2 months
- You ?