UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

ML-based reweighting of BSM signal grids: an example in Rivet

Tomasz Procter,
LHC Reinterpretation Forum,
February 2025

with thanks to S. Jiggins, A. Buckley

Grant CHIST-ERA 2022/04/Y/ST2/00186 National Science Centre, Poland

Outline

- Why reweight signal grids, and what is CARL?
- Introducing the case study
- ATLAS-SUSY-2018-30 refresher
- Setup
- YODA 2 magic
- Results

CARL & Reweighting

Reweighting & CARL-torch

- Signal grids are big; event generation is expensive.
- Solution: generate events at a subset of points, *reweight* to reach the rest of the grid.
- Train an ML-classifier, and use “likelihood ratio trick”:
  - From classification score *s* to likelihood ratio *r*

$$s(\vec{x}) = \frac{p_0(\vec{x})}{p_0(\vec{x}) + p_1(\vec{x})}$$

$$r(\vec{x}) = \frac{s(\vec{x})}{1 - s(\vec{x})}$$

- CARL-torch package[1] - open source package for training the classifier and applying the weights.
  - Builds on ideas from 1506.02169
  - Pub-note coming shortly
    - Focusses on reweighting technique and disk-space savings.

[1] https://zenodo.org/records/4062049

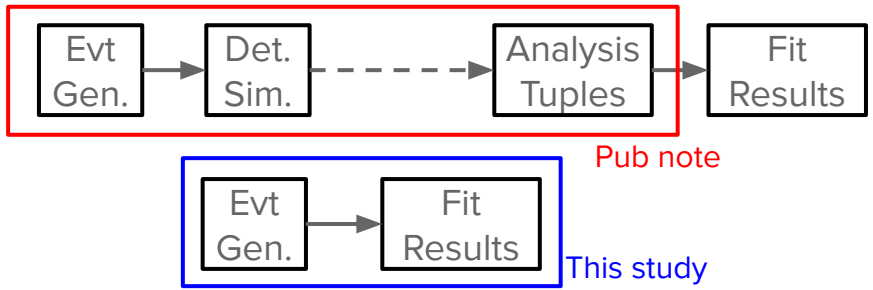
CARL-torch – on a grid: how to reweight to an arbitrary point?

- Use several points from across the grid in the alternate sample, and include the co-ordinates (m_g,m_χ) in the training features (“prametrised neural likelihood ratio estimator”)
- For the nominal sample, fill these inputs with *random* (m_g,m_χ) values from the set of points used.
  - Gives the network a “proposal” distribution from which to sample.
  - (also, otherwise the classifier can discriminate perfectly between nominal and alternate).
  - In practice a nominal sample using several points across the grid is necessary to ensure support.
- Then, at inference-time, process a nominal sample, with (m_g,m_χ) set to the *target* values.
- Obtain classifier scores, convert to weights.

$$r(\vec{x}|m_{\tilde{g}}, m_{\tilde{\chi}}) = \frac{s(\vec{x}|m_{\tilde{g}}, m_{\tilde{\chi}})}{1 - s(\vec{x}|m_{\tilde{g}}, m_{\tilde{\chi}})}$$

Task at hand

- Forthcoming ATLAS pub-note verifies that individual physical distributions are well reproduced.
- But what about the practicalities of a real analysis:
  - BSM Signal Regions tend to be defined from the (extreme) tails of multiple observables – poor reweighting there may not be captured by statistical distance measures
- Can we reproduce the most important outputs of an analysis:
  - CR/SR yields
  - Limits and exclusions

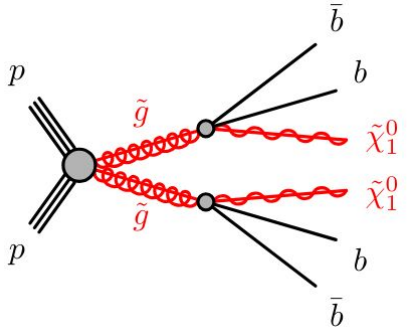


- Testing this is going to mean a lot of MC generation (for comparison across the grid).
- Let's do a study in Rivet to check how things work!

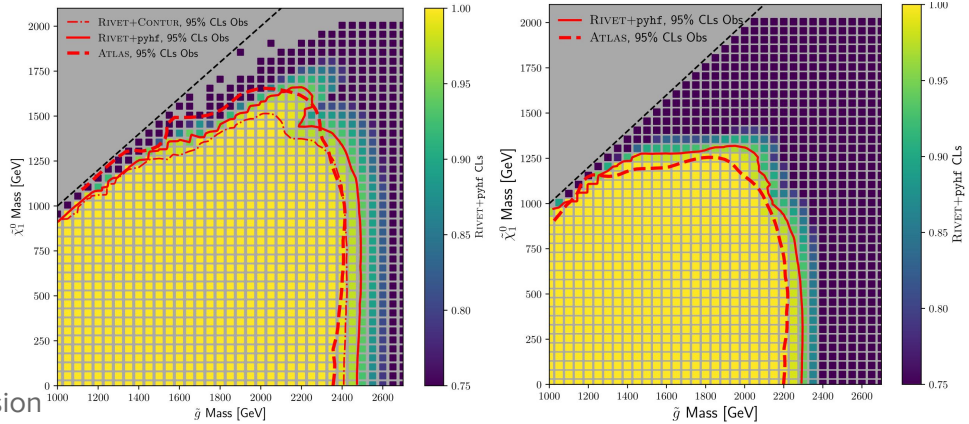
Setup and training

Refresher: ATLAS-SUSY-2018-30 (arXiv:2211.08028)

- Search for pair produced gluinos decay to squarks in multi-b jet final states.
- Much discussed at previous RiF's/similar
  - Last RiF, Durham
  - "Mini-workshop" Grenoble, June 2024 (Me, Krzysztof)
- (Partly) NN-based search for gluino pair production
  - NN public, reinterpretations have been largely successful (both NN & C&C regions).
  - Thanks (again) to the analysis team for providing so much material.
- So why use this analysis:
  - Validated rivet analysis
  - Range of observables and SRs
  - *Lots* of experience running simplified workflow (pythia Evt. gen.+Rivet ntuple dumping)



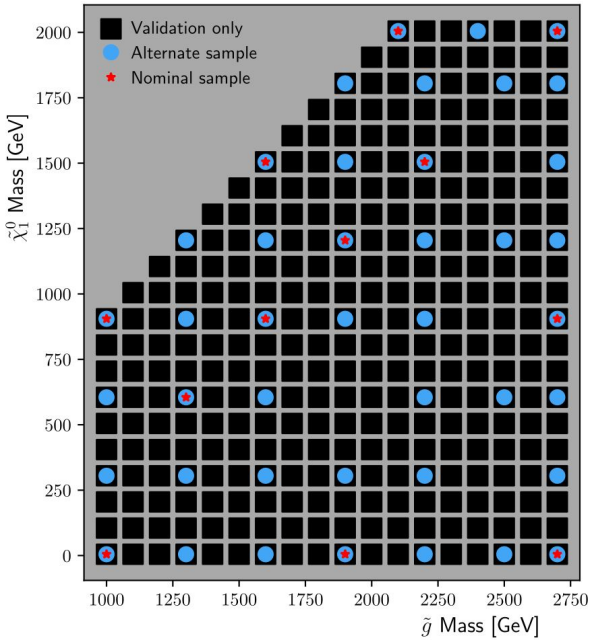
C&C Exclusion



NN Exclusion

Training and setup

- Generate training data at 12 (nominal) and 30 (alternate) points.
  - (This is probably overkill, but I had a thesis to write…)
  - Interesting future studies on best way to choose points.
- Generate validation MC at all the intermediate points.
- Train on 19 input features:
  - Small-jet kinematics, whole event-variables (MET, H_T, invariant masses etc)
  - Large-jet kinematics deliberately excluded to provide validation variables.
- Network setup is *deliberately* very simple.
  - This is a proof-of-concept study.
  - But also clear that even a small amount of optimisation would probably make results *even* better.



Aside: Rivet practicalities and YODA 2.0

See Andy's talk this afternoon for full description of YODA 2

- CARL weights are inferred "on-the-fly", event-by-event
  - Depend on observables (e.g. H_T) calculated by Rivet quite far into the event loop.
  - CARL outputs ONNX files – can reuse existing Rivet machinery.
    - As simple as two extra lines of code!
    - (See talks at Grenoble, last RIF and three RiFs ago)
- Rivet has a sophisticated multi-weight treatment only for **hepmc** weights.
  - **Not** "on-the-fly" weights
- Only solution: Rewrite the entire Rivet multiweight handling system.

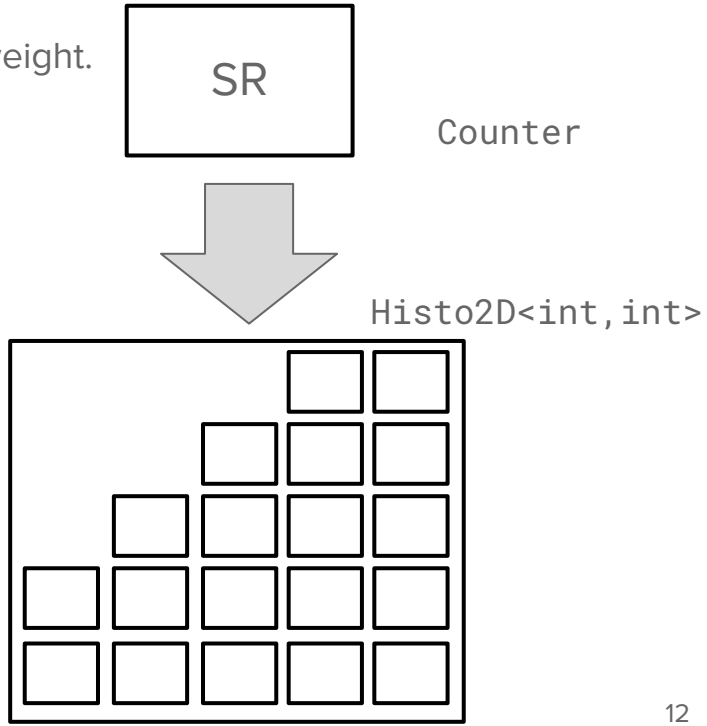
Aside: Rivet practicalities and YODA 2.0

See Andy's talk this afternoon for full description of YODA 2

- CARL weights are inferred "on-the-fly", event-by-event
  - Depend on observables (e.g. H_T) calculated by Rivet quite far into the event loop.
  - CARL outputs ONNX files – can reuse existing Rivet machinery.
    - As simple as two extra lines of code!
    - (See talks at Grenoble, last RIF and three RiFs ago)
- Rivet has a sophisticated multi-weight treatment only for **hepmc** weights.
  - **Not** "on-the-fly" weights
- ~~Only solution: Rewrite the entire Rivet multiweight handling system.~~
- Use new features from YODA 2!
  - We can construct n-dimensional histograms from virtually any C/C++ float or integral type.

Aside: Rivet practicalities and YODA 2.0

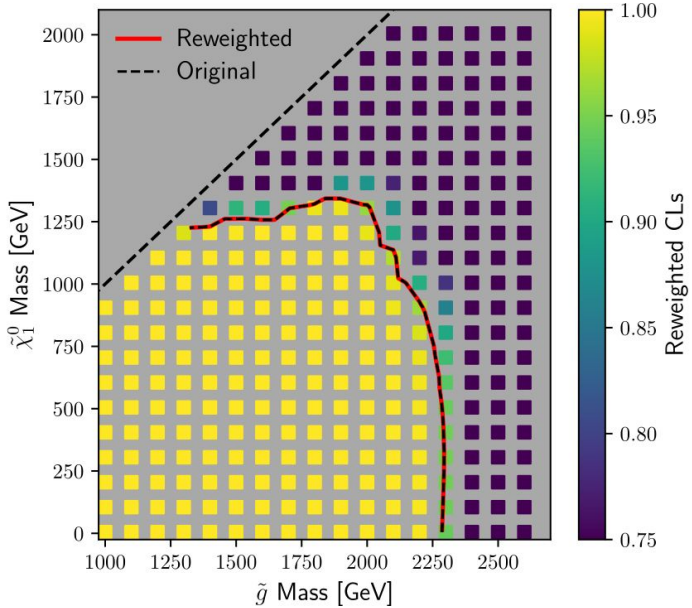
- Infer weights across the entire (pre-planned) grid as soon as all necessary info obtained.
  - In order to correct back to 139 fb^{-1}, all events need a weight.
- Store weights for every grid point.
- Promote every 0D bin to a 2D (int) histogram
  - One entry per point in parameter space
- Define an analysis-specific fill method
- Total changes to analysis:
  - < 20 SLoC added (mainly just to get inputs for the CARL network)
  - 3 lines removed (early event vetoes)

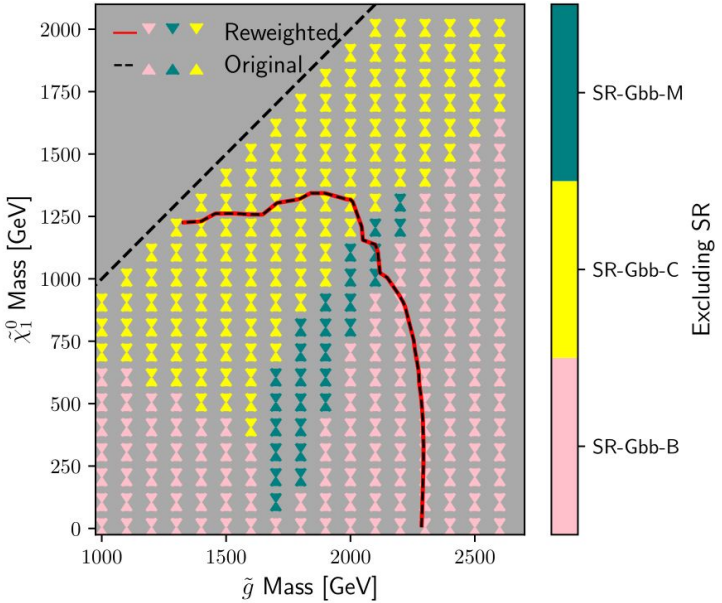


Results

Results: C&C SRs and exclusion

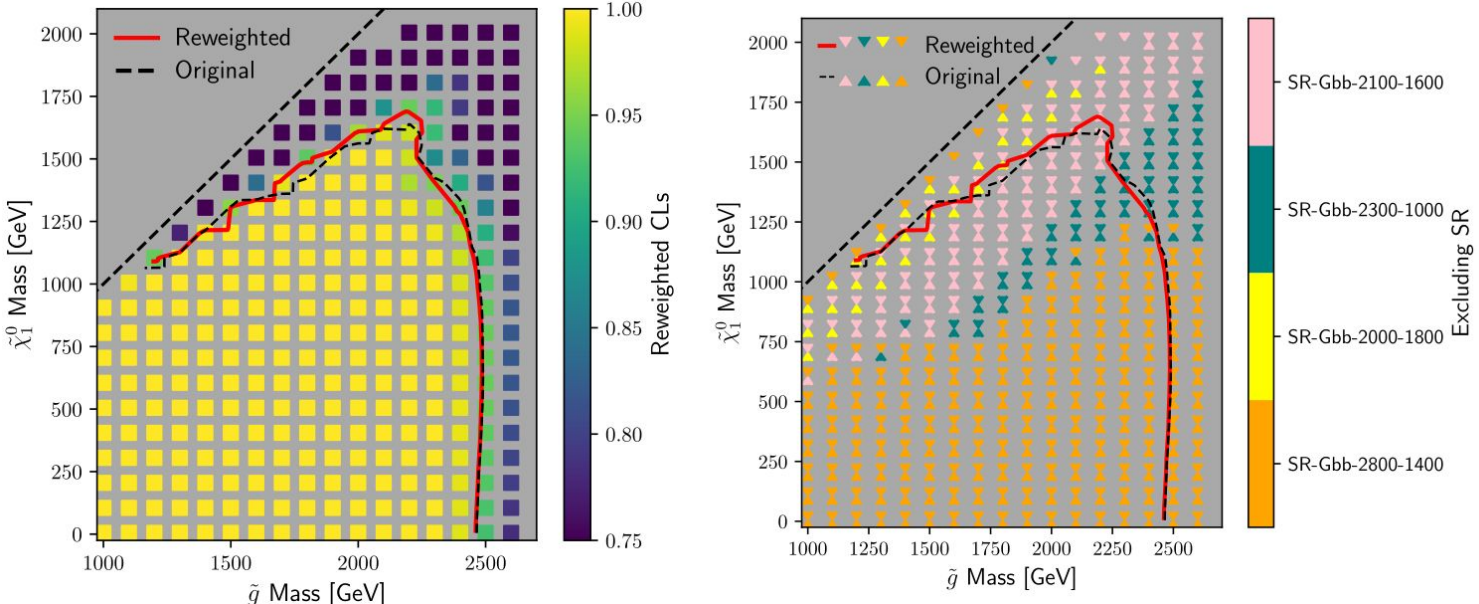
- Exclusion contours replicated almost perfectly.
- At every single point where events were generated, the same “most-exclusionary” SR was picked.





Results: NN SRs and exclusion

- Very low stats bins (0-5 events), defined by an extreme cut on a feature not in the training.
  - About “as hard as it gets”
- Still very promising results – likelihood contours very close.

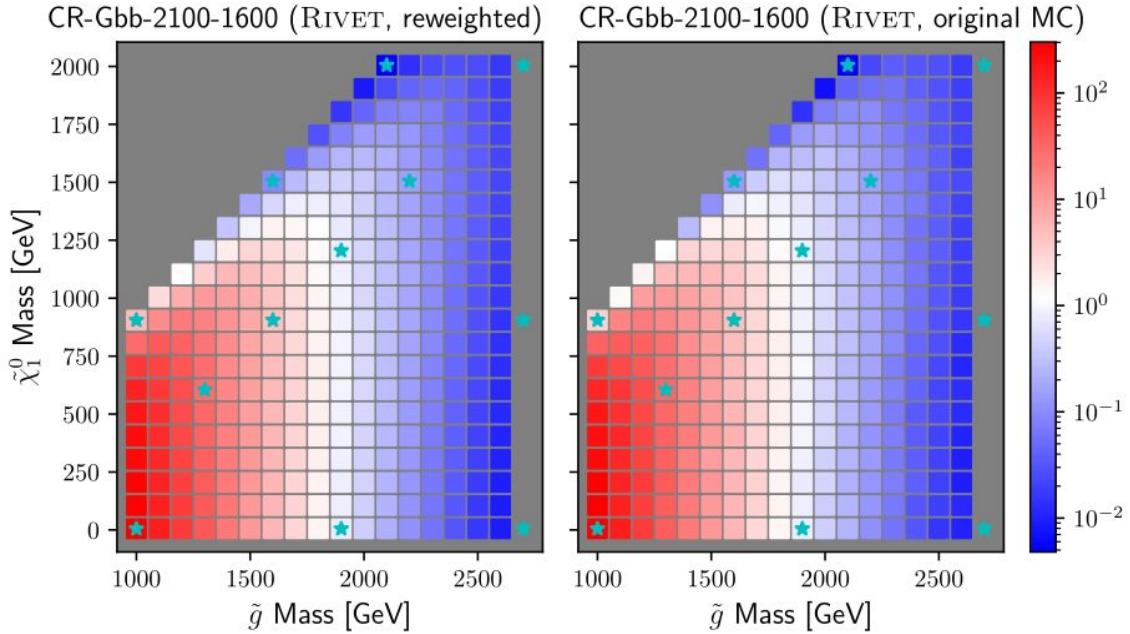
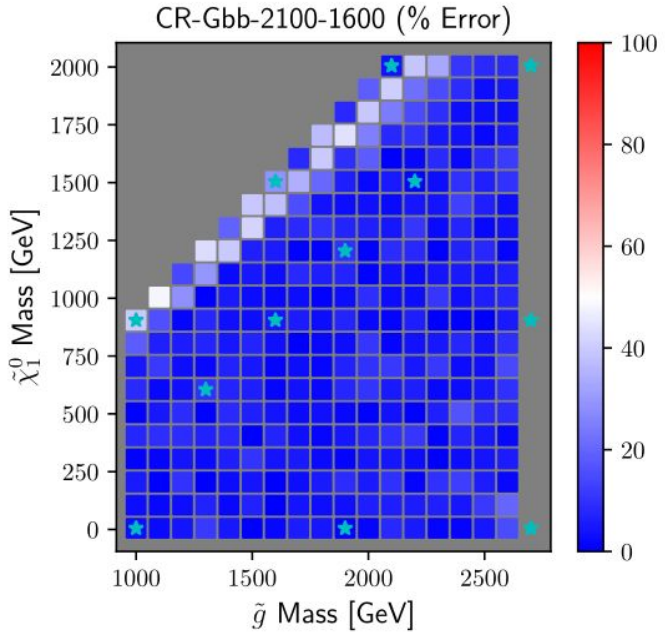


Conclusions

Conclusions

- These reweighting techniques work.
  - See upcoming ATLAS pub-note.
  - Not just in principle when comparing abstract distributions
  - But for “real” tasks like drawing exclusion curves, too.
  - Could have applications beyond the experiments?

- Having reliable reinterpretation material => Lots of really cool follow up studies.
  - Including by experimentalists! (perhaps better utilised historically by measurements)
- Easy Rivet interface to ONNX => it can be a playground for all sorts of cool ML studies, too.

BONUS

Results: CR yields