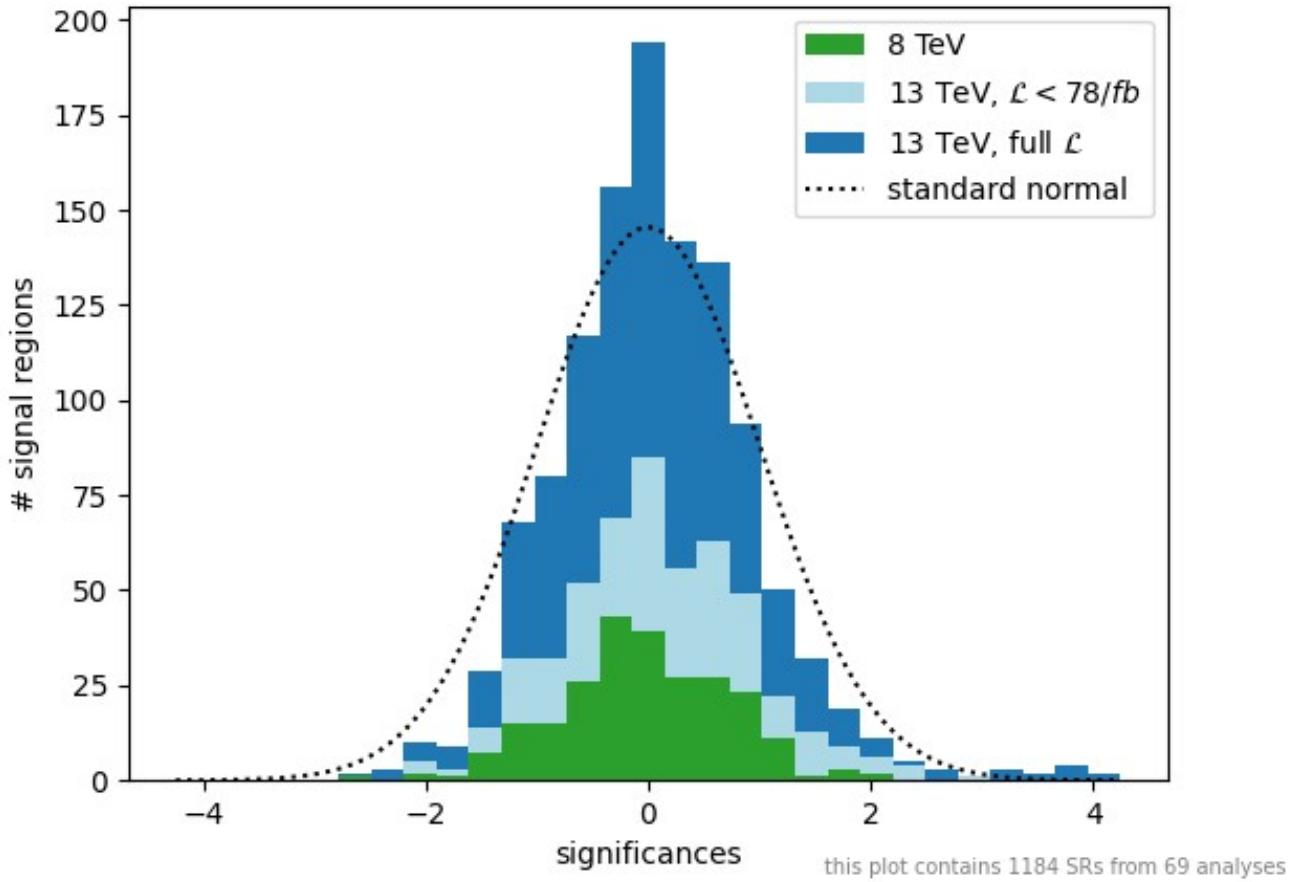
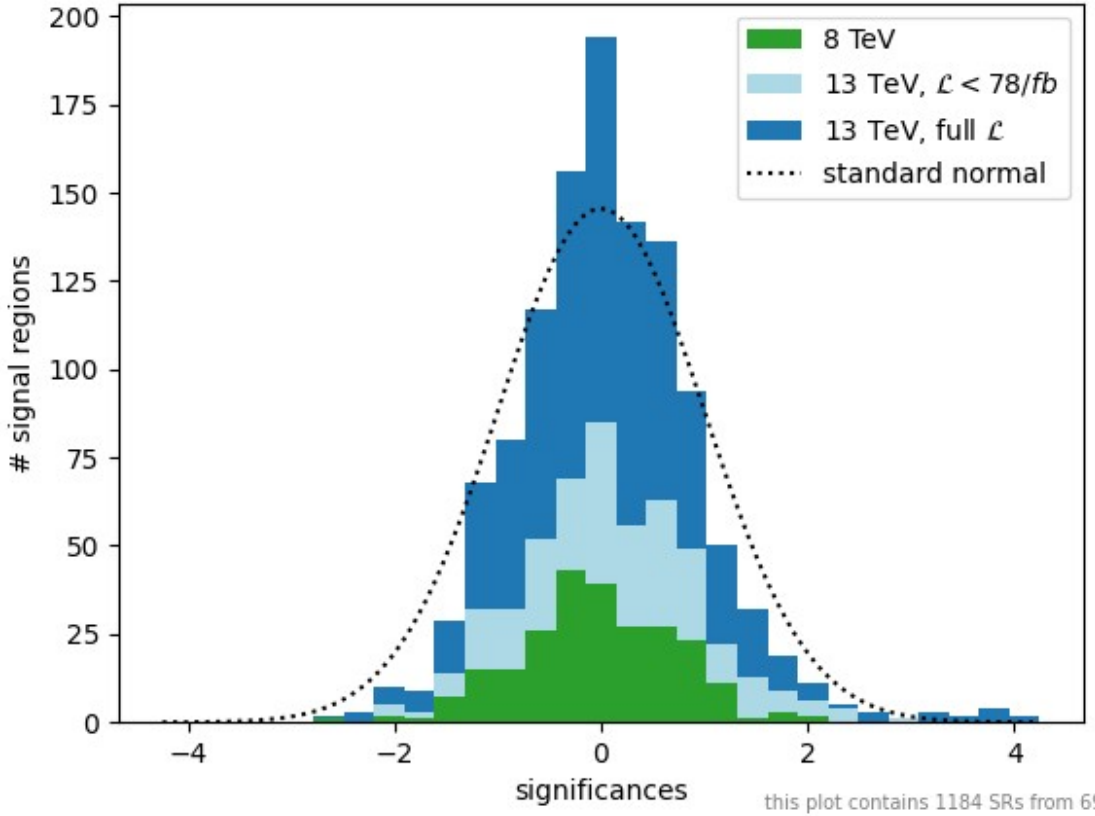
Experimental Data and SModelS



Wolfgang Waltenberger (ÖAW and Uni Wien)
for the SModelS collaboration

Invisibles Data Days, May 2024



In SModelS v3.0.0 – to be released ~ next week, we will have the results of a total of 160 publications in our database (if we count really everything). They are all ATLAS and CMS searches for new (particle) physics.

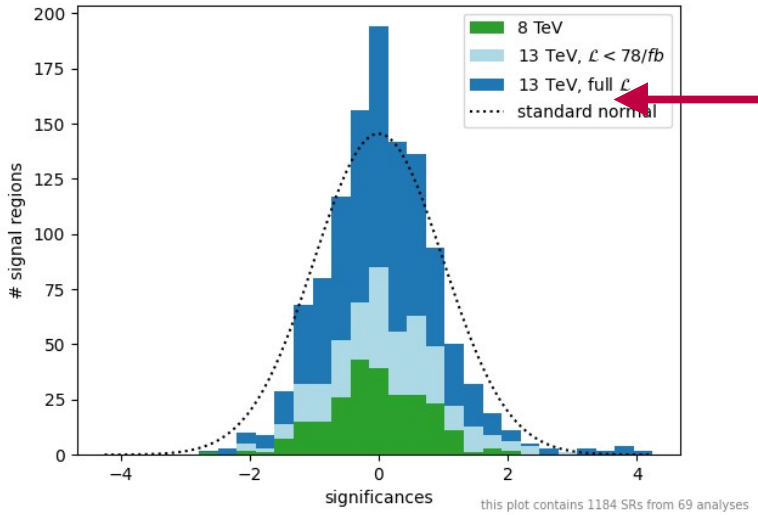
For (only) 69 of them do we have enough information to construct at least a simplified statistical model. Usually, a single result has multiple “signal regions”.

The plot above presents significances with respect to the Standard Model hypothesis, of all signal regions within these 69 publications.

If all is done correctly, and no new physics is in the data, the distribution should follow a standard normal. New physics would manifest itself as an overabundance of positive values.

And What Do We Do With The Information Contained In These Statistical Models?

A Next Standard Model



from this ...

… we wish to infer this:

i.e. identify the One True Point [*] (a.k.a. the Next Standard Model) in Hitoshi Murayama's landscape of theories

[*] obviously in reality we would actually want a posterior density, not one point

SModelS

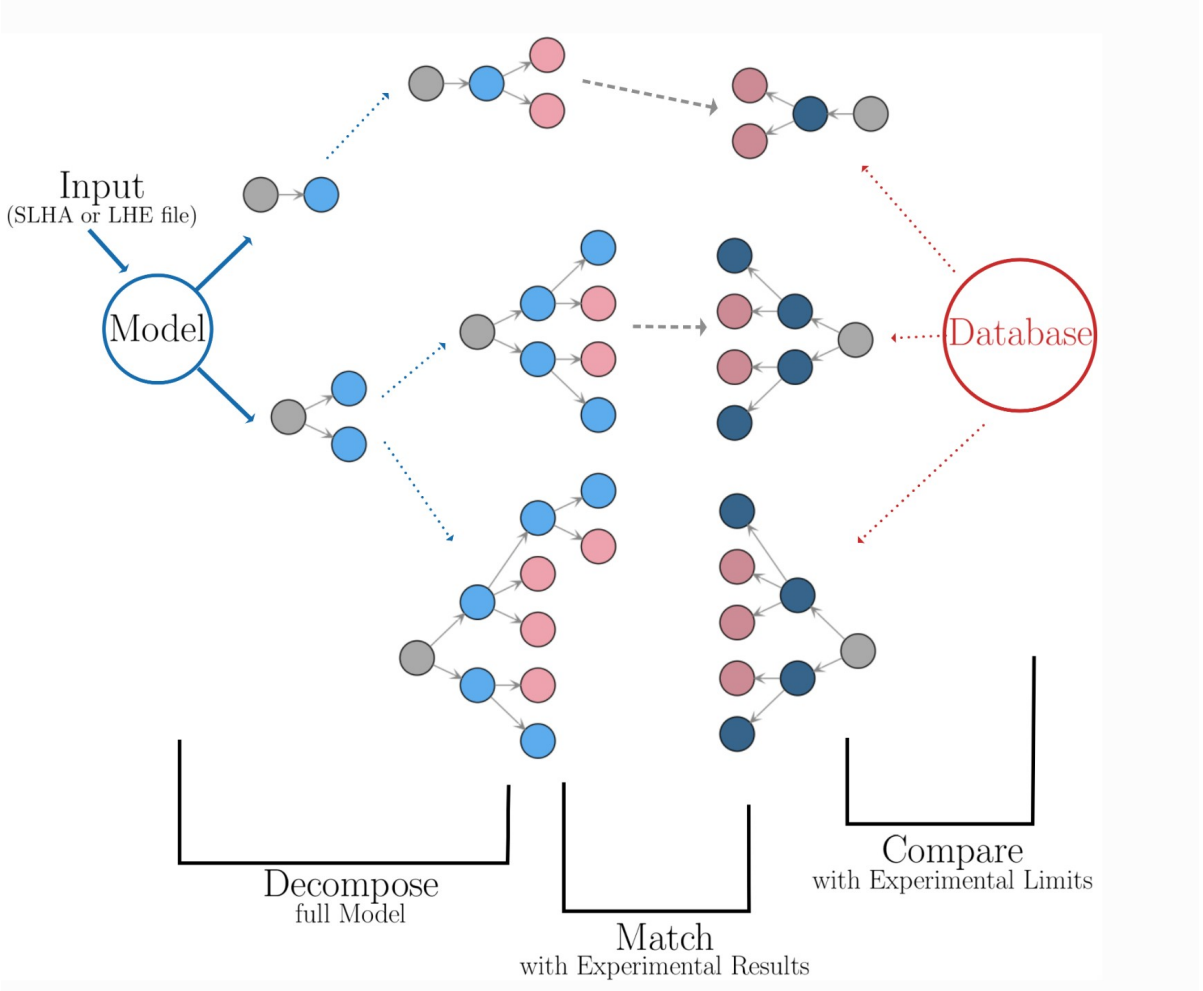
GitHub | pypi package 2.3.3 | Open in Colab | docs main

https://smodels.github.io

19 December 2023: SModelS version 2.3.3 available (what's new)

Paper for version 2.3: arXiv:2306.17676

- a tool to quickly compare a theory against a database of experimental results
- decomposes theory into its "simplified model spectrum"
- No expensive Monte Carlo event generation required!
- Database contains results from > 100 CMS and ATLAS publications



Our Inverse Problem

To connect our observations with contender theories for the Next Standard Model,

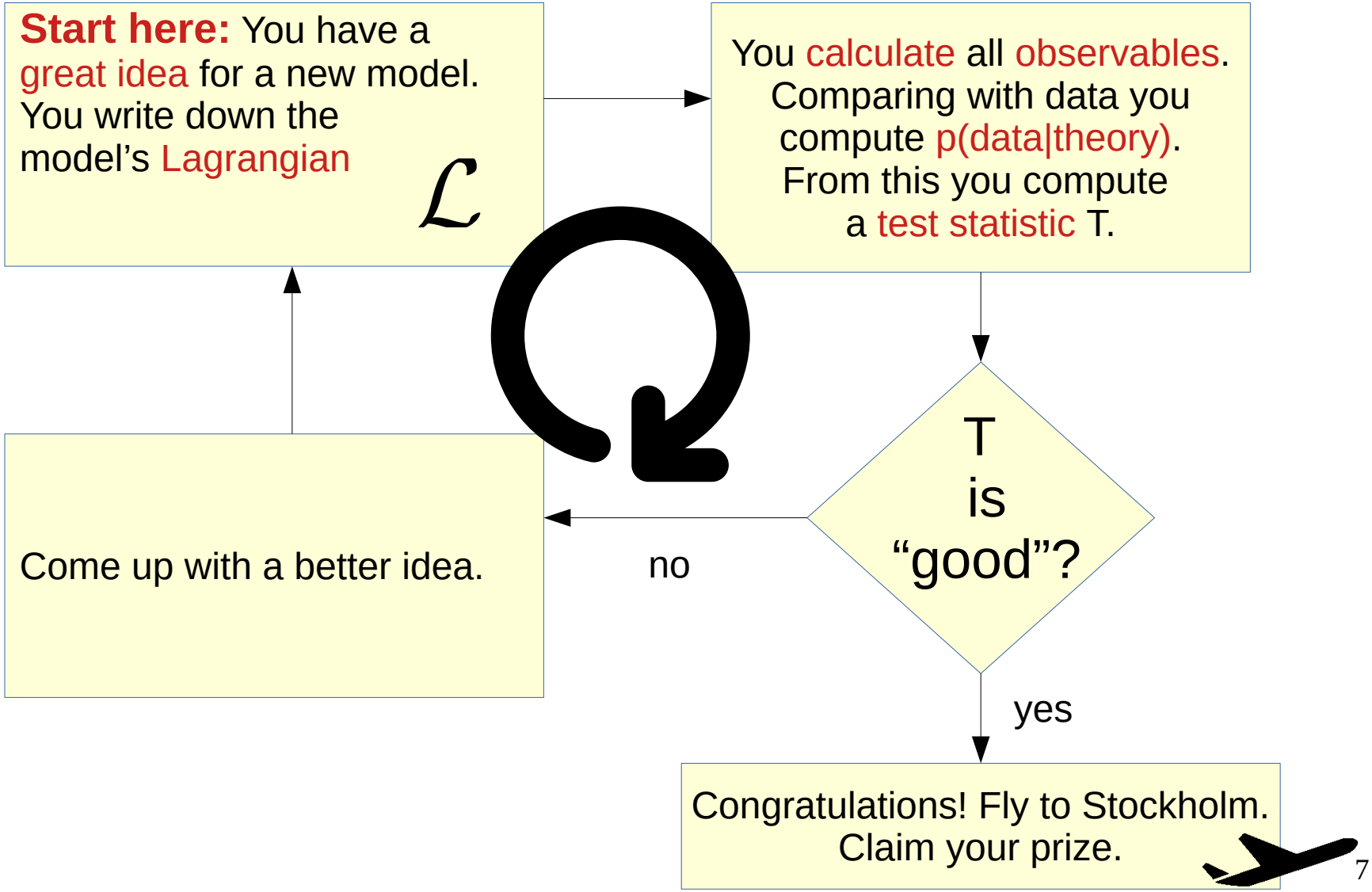
- In addition to lots of other information, we need statistical models (in one way or another)

- we can either start by:
  postulating a theory, then fit its parameters
  ("top-down")

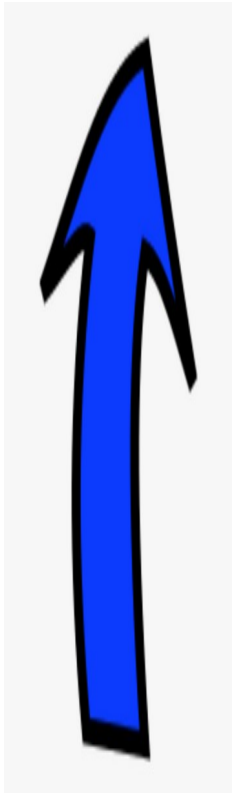
- or:
  try to build our model directly from data
  ("bottom-up")

Top-down versus bottom-up

Top-Down:



Top-down versus bottom-up



Bottom-Up:

Only now do you think about symmetries, gauge groups, etc that may underlie all observations. Construct your Lagrangian.

$\mathcal{L}$

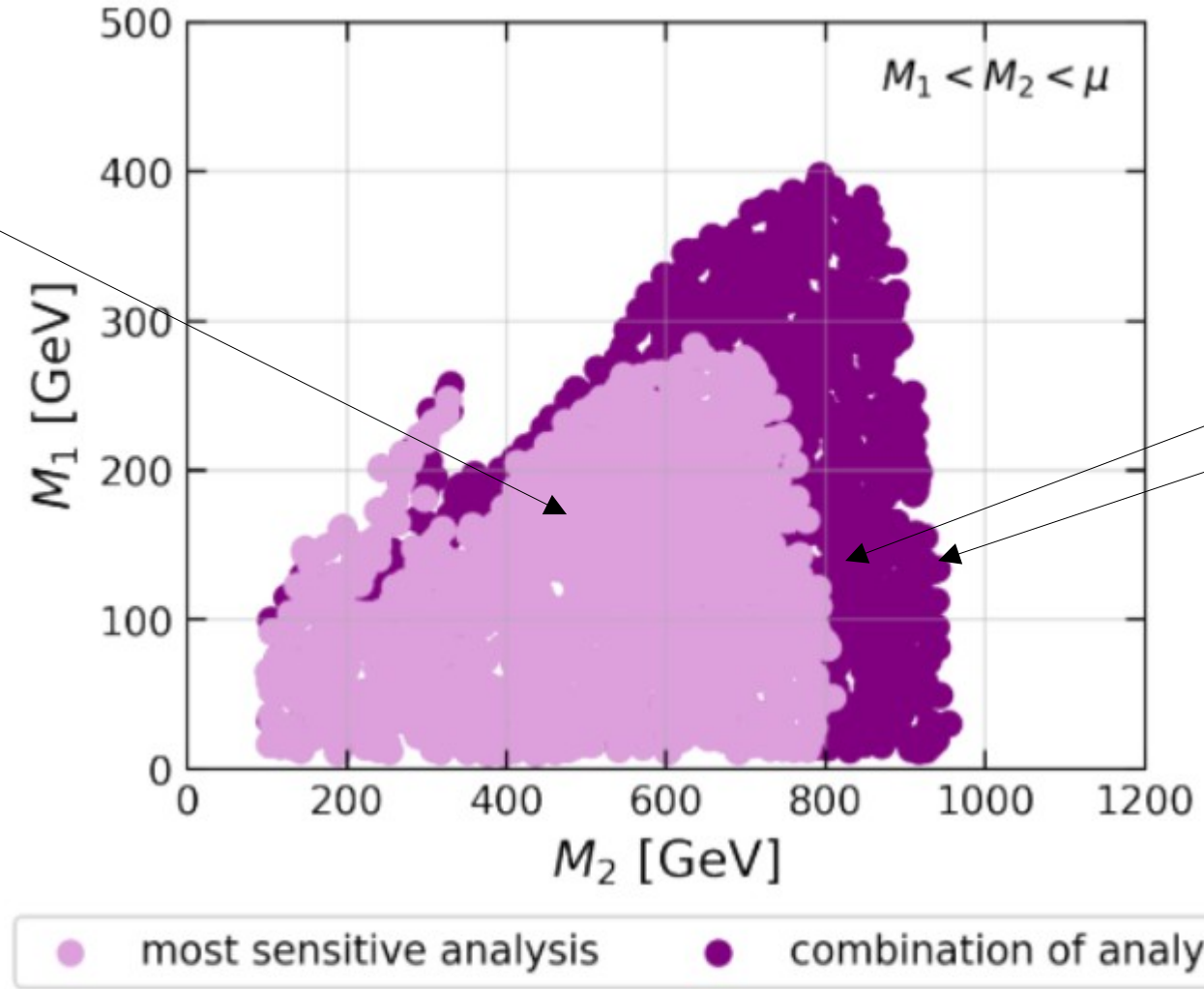
From the descriptions you try and construct precursor theories to the NSM that describe everything you really know about TeV-scale (and below) physics

Start here: You describe your experimental findings in a language amenable to theoretical physics, e.g. simplified models for on-shell effects (“searches”), effective field theories for off-shell effects (“measurements”).

Top-down: example

"Global LHC constraints on electroweakinos with SModelS v2.3"
Model: MSSM, but looking at its charginos and neutralinos only

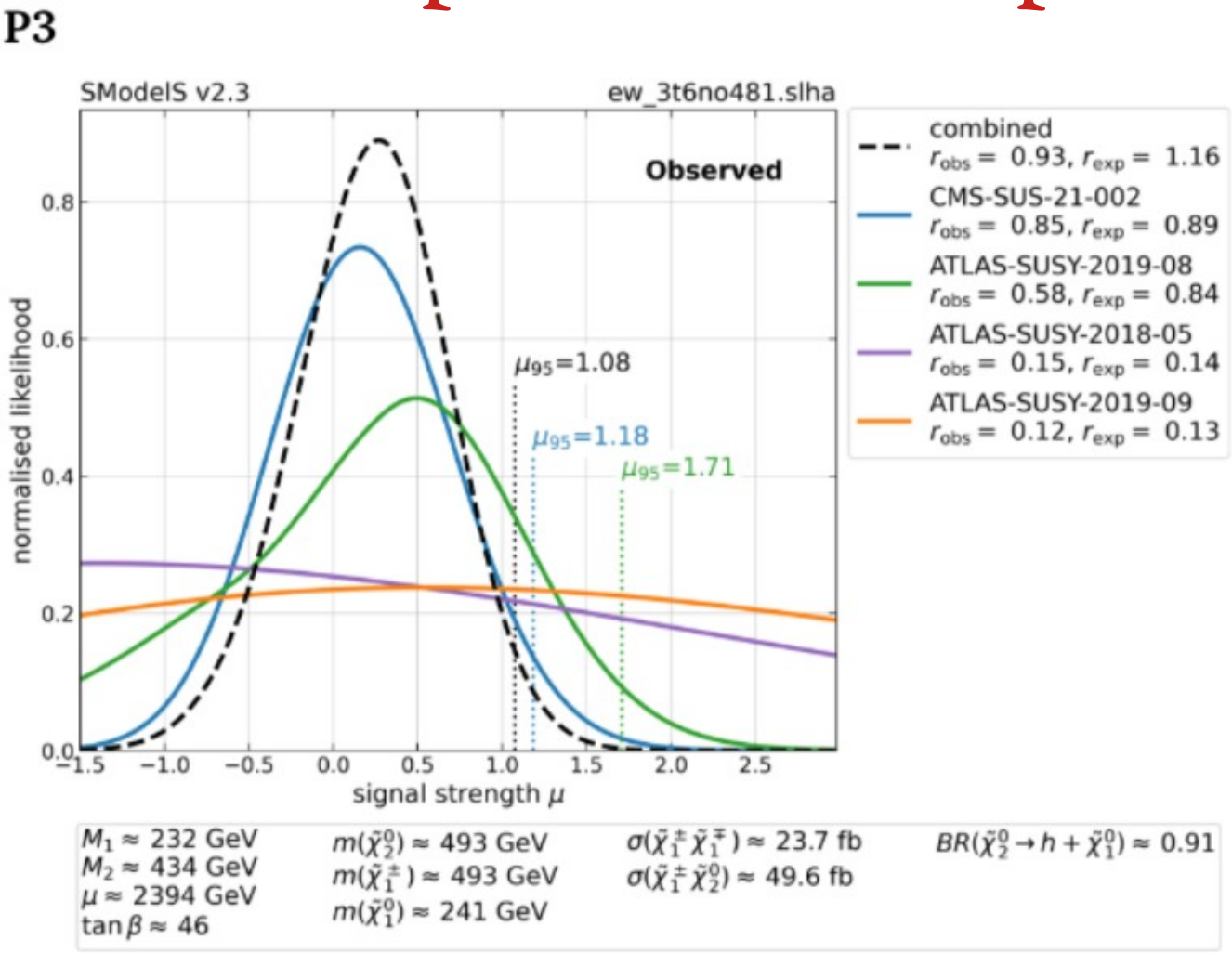
Excluded
Model points



gain in exclusion reach by combining (CMS and ATLAS) results

https://arxiv.org/abs/2312.16635

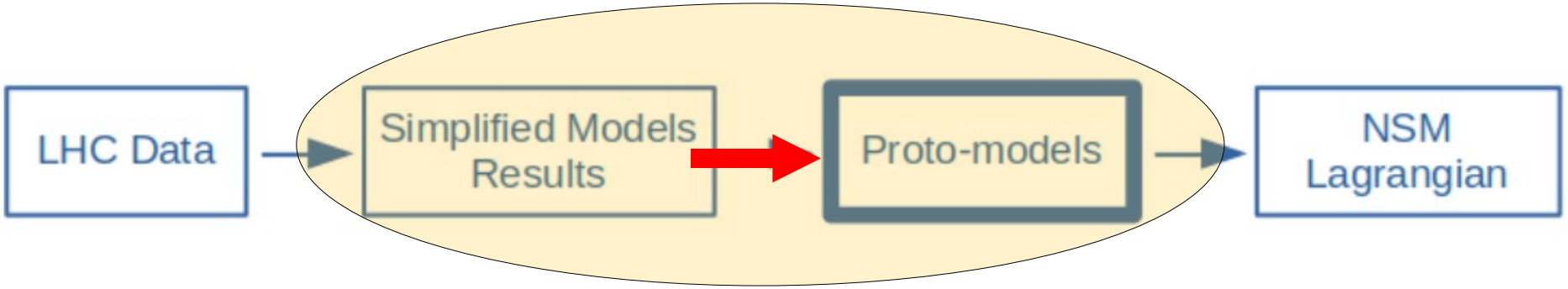
Top-down: example

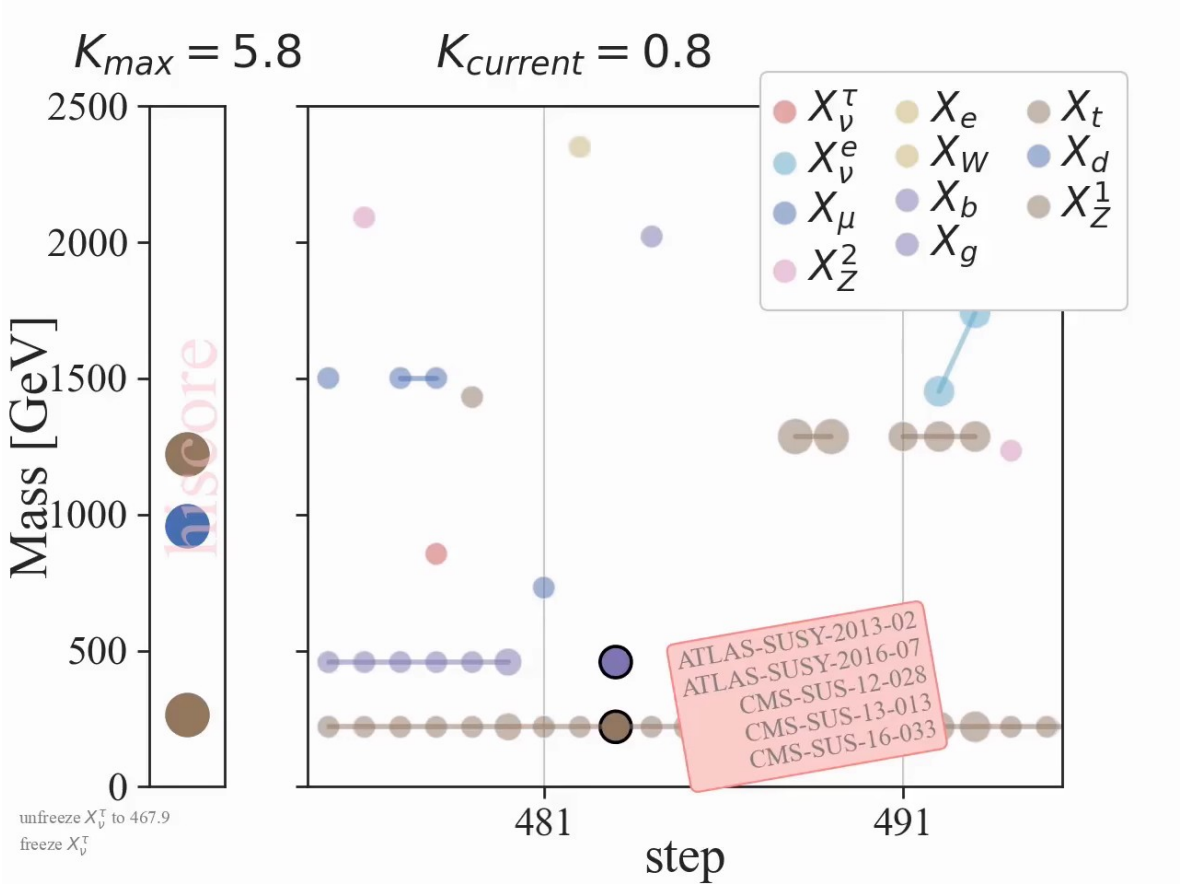


Combination at work, for one model point. Likelihoods here are functions of a single parameter (the overall “signal strength”). Combining several likelihoods (colored lines) results in a narrower combined likelihood (dashed line).

https://arxiv.org/abs/2312.16635

Bottom-up: Protomodels





"Likelihood-based automated model building" (in a wider sense of the words)

Statistical Models in the SModelS Database

ATLAS, Run-2, Full Lumi only:

ID	Short Description	$\mathcal{L}$ [fb^{-1}]	UL_{obs}	UL_{exp}	EM	comb.
ATLAS-EXOT-2018-48	di-top resonance	139.0	✓	✓		
ATLAS-EXOT-2019-03	dijet resonance	139.0	✓	✓		
ATLAS-SUSY-2018-04	2 hadronic τ	139.0	✓		✓	PyHF
ATLAS-SUSY-2018-05	2ℓ + jets, EWK	139.0	✓		✓	PyHF
ATLAS-SUSY-2018-06	3ℓ, EWK	139.0	✓	✓	✓	
ATLAS-SUSY-2018-08	2 OS ℓ	139.0	✓		✓	
ATLAS-SUSY-2018-09	2 SS ℓ	139.0	✓			
ATLAS-SUSY-2018-10	1ℓ + jets	139.0	✓		✓	
ATLAS-SUSY-2018-12	0ℓ + jets	139.0	✓	✓	✓	
ATLAS-SUSY-2018-13	displaced jets	139.0			✓	SLv1
ATLAS-SUSY-2018-14	displaced vertices	139.0			✓	PyHF
ATLAS-SUSY-2018-16	2 soft ℓ + jets, EWK	139.0	✓	✓	✓	PyHF
ATLAS-SUSY-2018-22	multi-jets	139.0	✓		✓	
ATLAS-SUSY-2018-23	$Wh(\gamma\gamma)$, EWK	139.0	✓	✓		
ATLAS-SUSY-2018-31	2 b-jets + 2 h	139.0	✓		✓	PyHF
ATLAS-SUSY-2018-32	2 OS ℓ	139.0	✓		✓	PyHF
ATLAS-SUSY-2018-40	2 b-jets + 2 h	139.0	✓	✓	✓	
ATLAS-SUSY-2018-41	hadr. EWK	139.0	✓	✓	✓	SLv1
ATLAS-SUSY-2018-42	charged LLPs, dE/dx	139.0	✓	✓	✓	
ATLAS-SUSY-2019-02	2 soft ℓ, EWK	139.0	✓		✓	SLv1
ATLAS-SUSY-2019-08	$1\ell + h(bb)$, EWK	139.0	✓		✓	PyHF
ATLAS-SUSY-2019-09	3ℓ, EWK	139.0	✓	✓	✓	PyHF

Type of statistical model

Statistical Models

- **Only exclusions for particular models**
  If only “exclusion lines” are given, without upper limits, we can do nothing

- **Observed 95% CL upper limits on production cross sections only:**
  cannot construct likelihood, binary decision “excluded” / “not-excluded” only (“critic”)

- **Expected and observed 95% CL upper limits**
  can construct an approximate likelihood with truncated Gaussian, cannot combine topologies, very crude approximation

- **Signal Efficiency “maps” for “simplified models”**
  can construct a likelihood as Gaussian (for the nuisances) * Poissonian (for yields), can work per SR, and combine topologies in each SR [*]

- **Efficiency maps + correlation matrices**
  can combine signal regions via multivariate Gaussian * Poissonians

- **Efficiency maps + full likelihoods**
  full realism, correct statistical model

- **Efficiency maps + full likelihoods + naming convention + knowledge about “data overlaps”**
  can combine publications at full realism

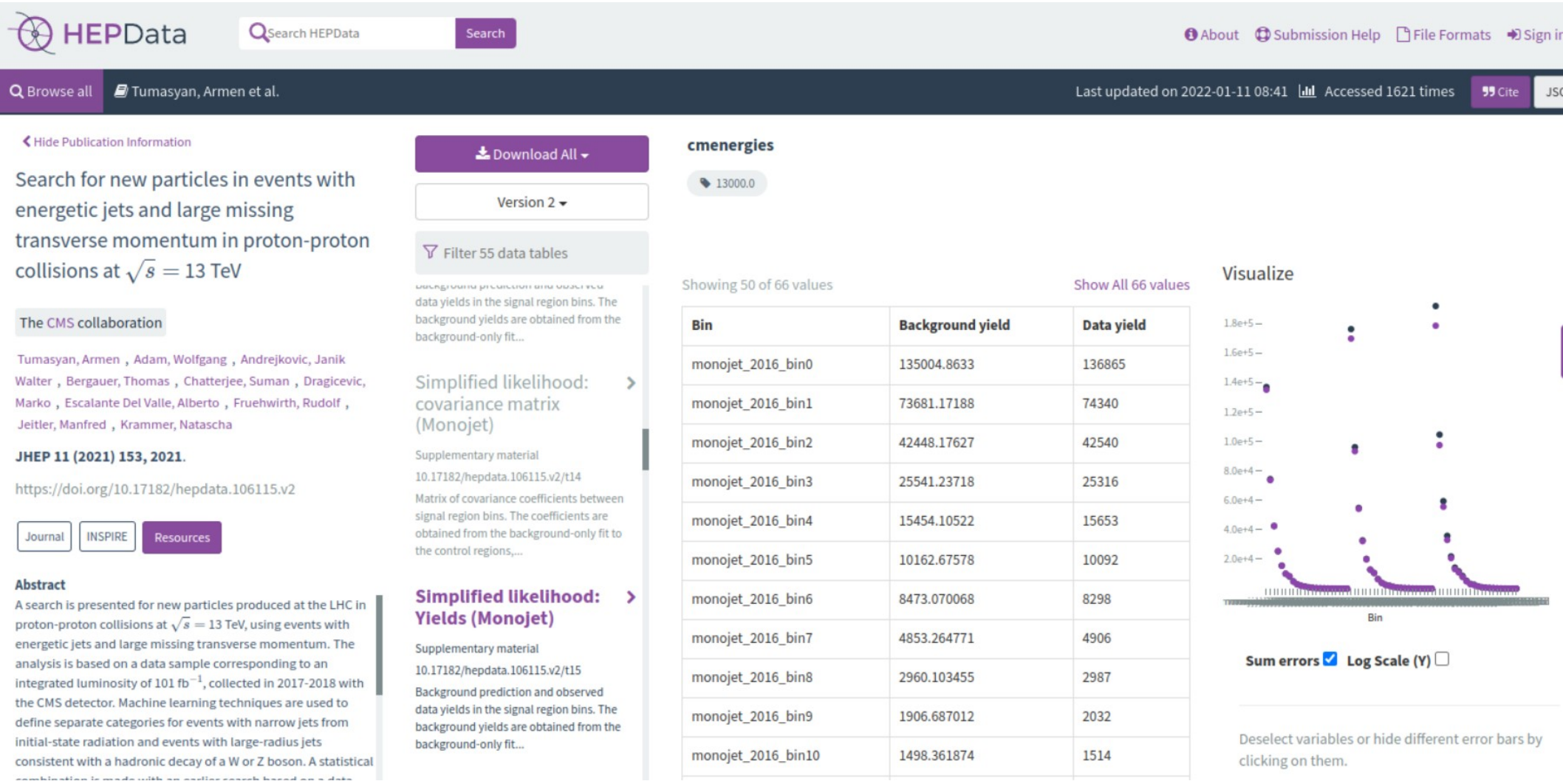
Combos

Likelihoods

BETTER

[*] if efficiency maps are not supplied, we can try to produce them with recasting frameworks

HEPDATA



(CMS monojet search)

Serialized Statistical Models

(Lukas is of course the authority here but) pyhf serializes the models into json files:

```
{
    "channel": {
        "type": "object",
        "properties": {
            "name": { "type": "string" },
            "samples": { "type": "array", "items": {"$ref": "#/definitions/sample"}, "minItems": 1 }
        },
        "required": ["name", "samples"],
        "additionalProperties": false
    },
}
```

Posterior distributions on nuisance parameters (such as jet energy scales, pdf uncertainties, reconstruction ineffciencies, etc etc) are described via “auxiliary measurements” –> keeping things frequentist (if we wish)!

Statistics Standardization comittee formed: **HS3**

https://github.com/hep-statistics-serialization-standard/hep-statistics-serialization-standard

These “closed world” approaches are general enough for users!

In "User Space": Machine Learning Surrogate Models

If speed and/or memory footprint become an issue, users can always also machine-learn surrogate models.

If only likelihood is needed: simple, fully connected neural networks (Bayesian neural networks for adding errors on the estimates), e.g.:

CERN-TH-2019-187

The DNNLikelihood: enhancing likelihood distribution with Deep Learning

Andrea Coccaro[a], Maurizio Pierini[b], Luca Silvestrini[b,c], and Riccardo Torre[a,b]

[a] *INFN, Sezione di Genova, Via Dodecaneso 33, I-16146 Genova, Italy*
[b] *CERN, 1211 Geneva 23, Switzerland*
[c] *INFN, Sezione di Roma, P.le A. Moro, 2, I-00185 Roma, Italy*

If full statistical model including toys is needed as well: generative network such as normalizing flows. e.g:

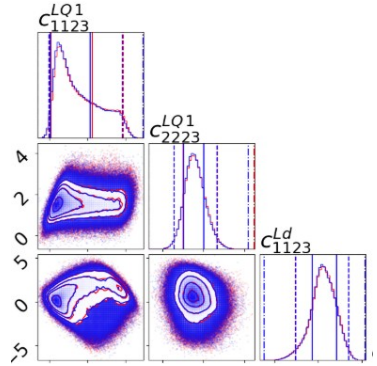
SciPost Physics Submission

The NFLikelihood: an unsupervised DNNLikelihood from Normalizing Flows

Humberto Reyes-González[1,2,⋆] and Riccardo Torre[2,†]

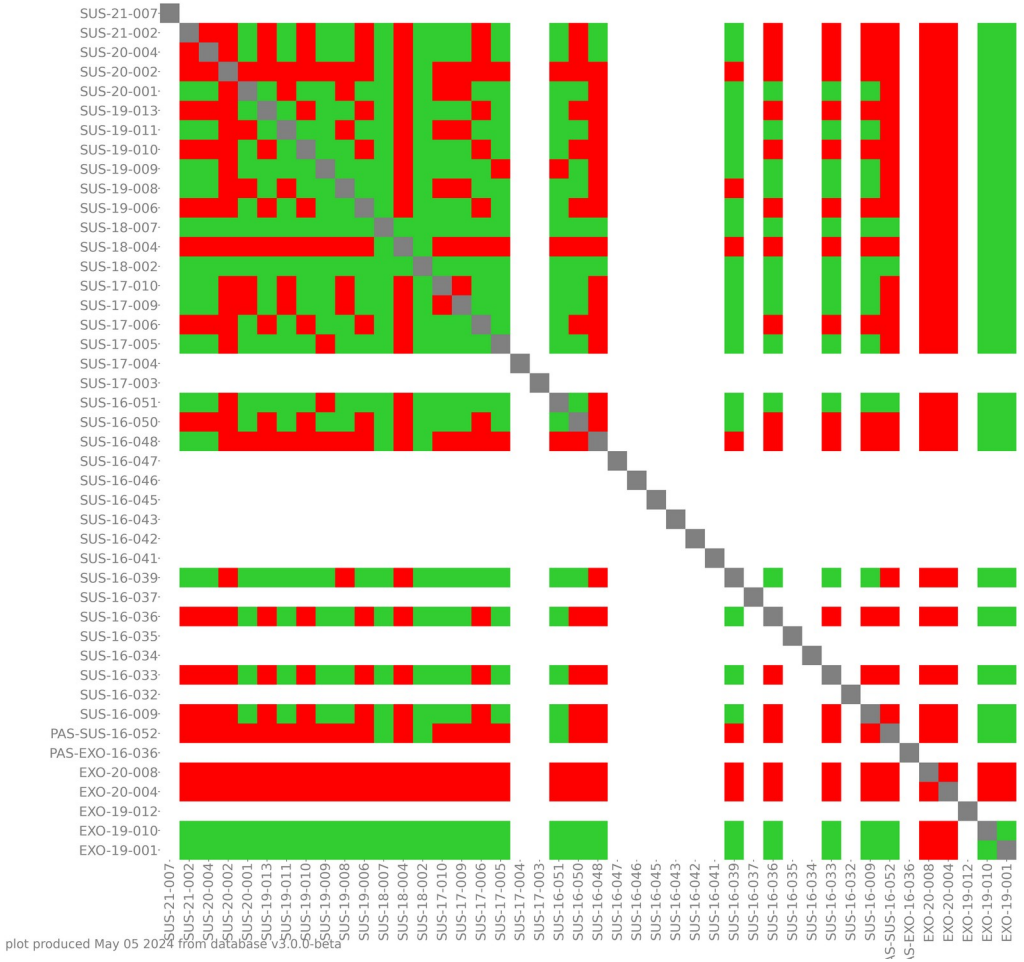
1 Department of Physics, University of Genova, Via Dodecaneso 33, 16146 Genova, Italy
2 INFN, Sezione di Genova, Via Dodecaneso 33, I-16146 Genova, Italy

⋆ humberto.reyes@rwth-aachen.de
† riccardo.torre@ge.infn.it



Combinability Matrix, and "recasts"

CMS, 13 TeV



- We keep track of which pairs of analyses have no overlaps in what LHC collision events enter the signal regions. In the "pheno community", we combine only if no overlap.
- Currently most reliable source for the combinations matrix: analysis recast frameworks: generating events, track which signal regions are populated by which events
- Requires detailed documentation about the analysis
- Experimental collaborations could in principle produce this information as well (and better)

Strength in numbers: Optimal and scalable combination of LHC new-physics searches

Jack Y. Araz, Andy Buckley, Benjamin Fuks, Humberto Reyes-Gonzalez, Wolfgang Waltenberger, Sophie L. Williamson, Jamie Yellen

SciPost Phys. 14, 077 (2023) · published 20 April 2023

doi: 10.21468/SciPostPhys.14.4.077 pdf BiBTeX RIS Submissions/Reports Check for updates

Reproducing an Analysis

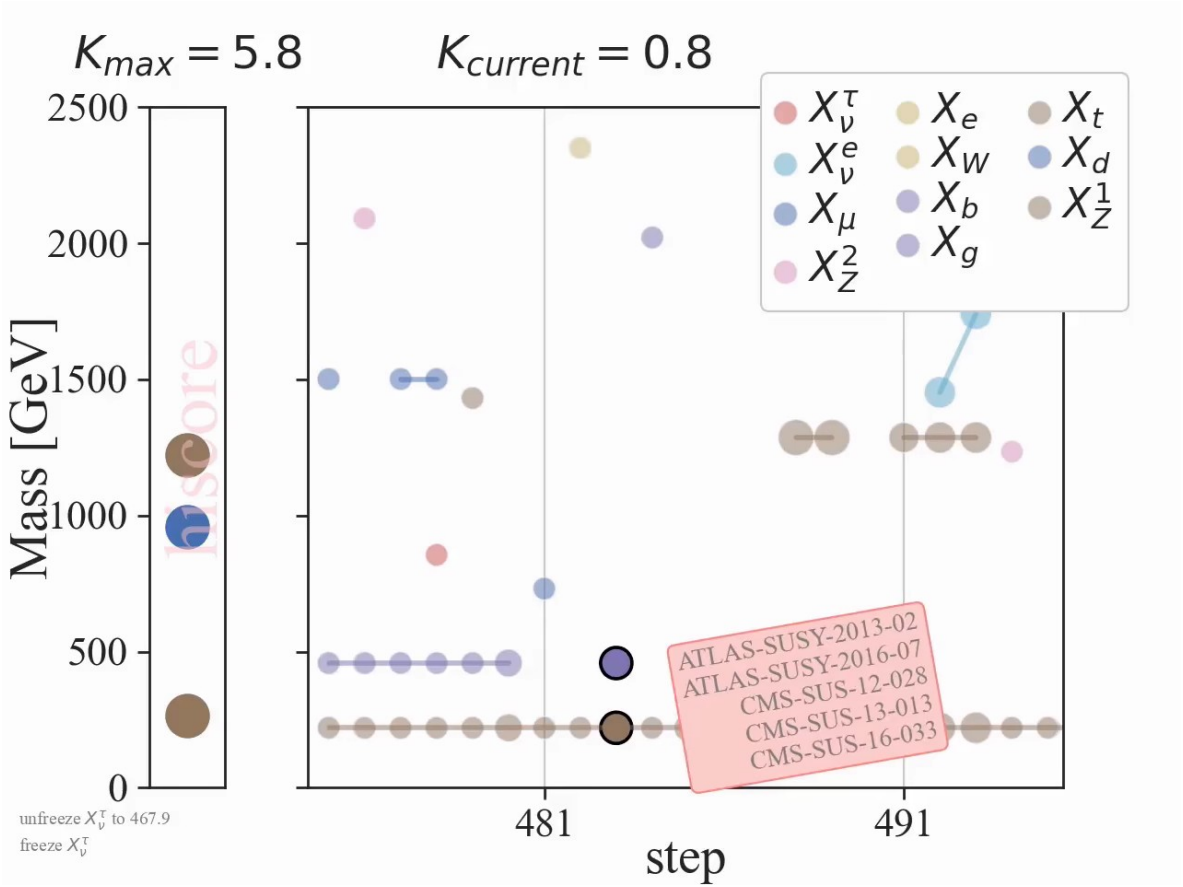
In order to facilitate reproduction of an entire analysis – to e.g. apply it to a new model, or to study analysis overap (previous slide), the analysis needs to be documented in great detail:

- Cutflow tables of the analysis cuts, reconstruction object efficiencies, machine-learning models (e.g. as onnx files) need to be given in digitized form, for e.g. recasts, studies of analysis overlaps.
- Implementation in a simple framework / language: e.g. SimpleAnalysis or Analysis Description Language (ADL)
- For Monte Carlo samples used in publication: generator datacards, etc
- Upper limit plots, signal efficiency maps (of simplified models) in digitized form
- **Statistical models in reusable digitized form**

Experimental Results → Protomodels

the test statistic

Particle spectra



A hiscore protomodel

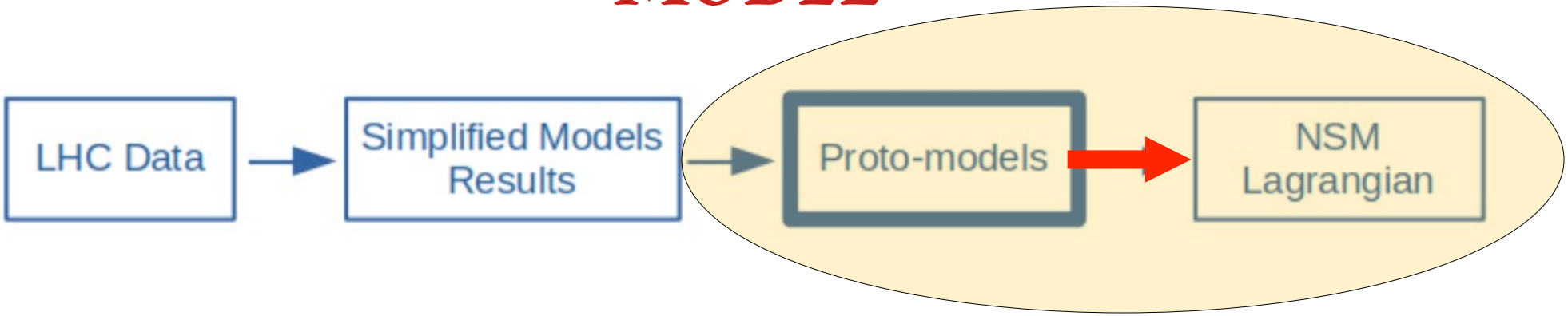
potential dispersed signals

Random modifications

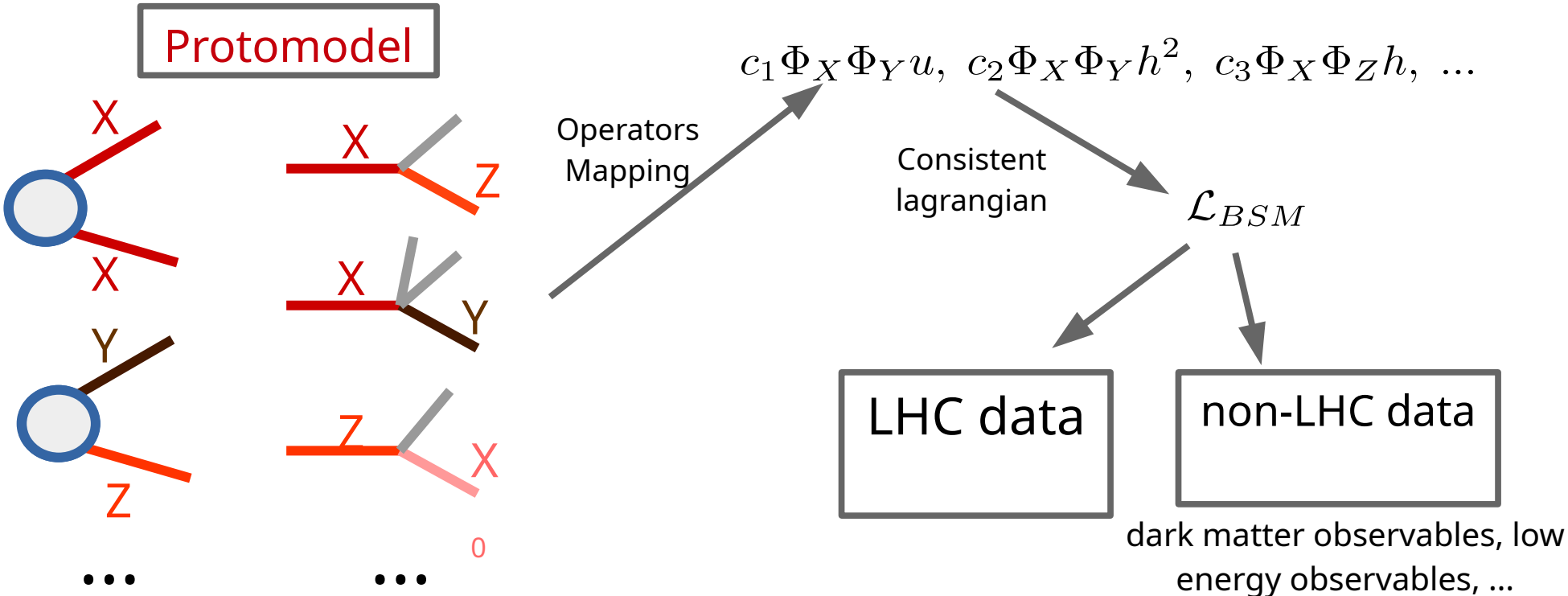
an MCMC-like walk

https://smodels.github.io/protomodels/videos

ProtoModels → Next Standard Model



Work from protomodels to UV complete theories has begun
(John Gargalionis, PostDoc in Valencia) – lot's of combinatorics!

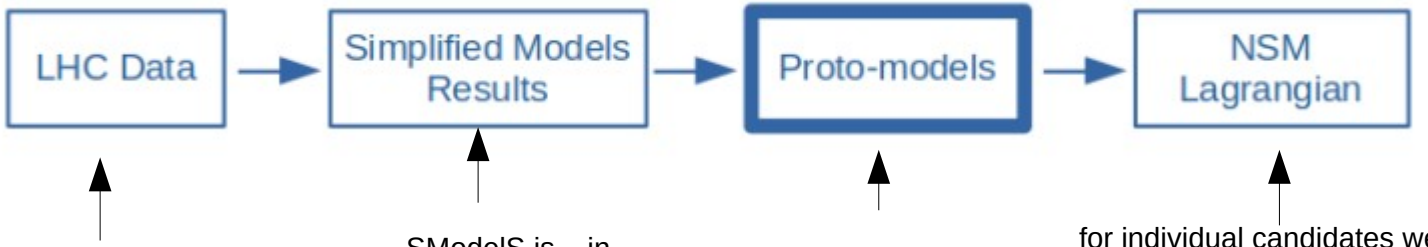


Summary

- Ideally, all important aspects of a physics result get published: cutflow tables, reconstruction object efficiencies, machine learned models, generator datacards, for SModelS: efficiency maps and: **serialized statistical models**!
- HEPdata is the collider physics results platform of choice
- DOIs for all digital objects
- Global fits are still an indispensible tool to obtain a “big picture” (see also gambit’s talk)
- The SModelS collaboration recently started to automate model building based on the statistical models

Differentiably if possible

If we had gradients we could perform gradient descent to find the best model, and we could use e.g. the Fisher information to infer the error on its parameters (or, alternatively we can then MCMC-sample).



described as likelihoods L that are differentiable with respect to the yields y_i

SModelS is – in principle -- differentiable

that's just a sum of simplified models → differentiable!

for individual candidates we can make this differentiable w.r.t fundamental parameters Θ_l, via neural networks

$$\frac{\partial L}{\partial \theta_l} = \frac{\partial L}{\partial y_i} \cdot \frac{\partial y_i}{\partial p_j} \cdot \frac{\partial p_j}{\partial (m_k, \Gamma_k, \sigma_k)} \cdot \frac{\partial (m_k, \Gamma_k, \sigma_k)}{\partial \theta_l}$$

Not yet required (theory space as well as space of measurements are still low-dimensional enough). Action item for now: keep choosing, supporting, developing technologies that support autograd

The Test Statistic

The test statistic K^c is a likelihood-ratio test that quantifies how much better the proto-model describes the data than the Standard-Model (plus a penalty for model complexity).

Quantifies violation of Standard Model hypothesis

Penalizes for model complexity

$$K^c := -2 \ln \frac{\mathrm{L}^c_{\mathbf{SM}}}{\mathrm{L}^c_{\mathbf{BSM}}(\hat{\mu})} \cdot \frac{\pi(\mathbf{SM})}{\pi(\mathbf{BSM})}$$

We search for proto-models and combinations of results / likelihoods that maximize K^c *while remaining compatible with all negative results in our database*.