UC SANTA CRUZ

MaPyDe

REUSE, REPRODUCE, REINTERPRET

mapyde

Reduce, Reuse, Reinterpret

Dr. Giordon Stark

and Prof. Mike Hance, Camila Ots

August 29th, 2023

indico.dur.ac.uk://event/1178/

arXiv:2306.11055

if you can read this, you're too close

Event: 2418777995

2016-06-04 03:47:03 C

Overview of Today

How to reuse a public ATLAS analysis

- The **typical workflow** for published analyses
- Reinterpretation of **ATLAS EWK Compressed**
- **Technical details** of mapyde
- **Implementing** custom workflows

⌛ **Let’s get started**

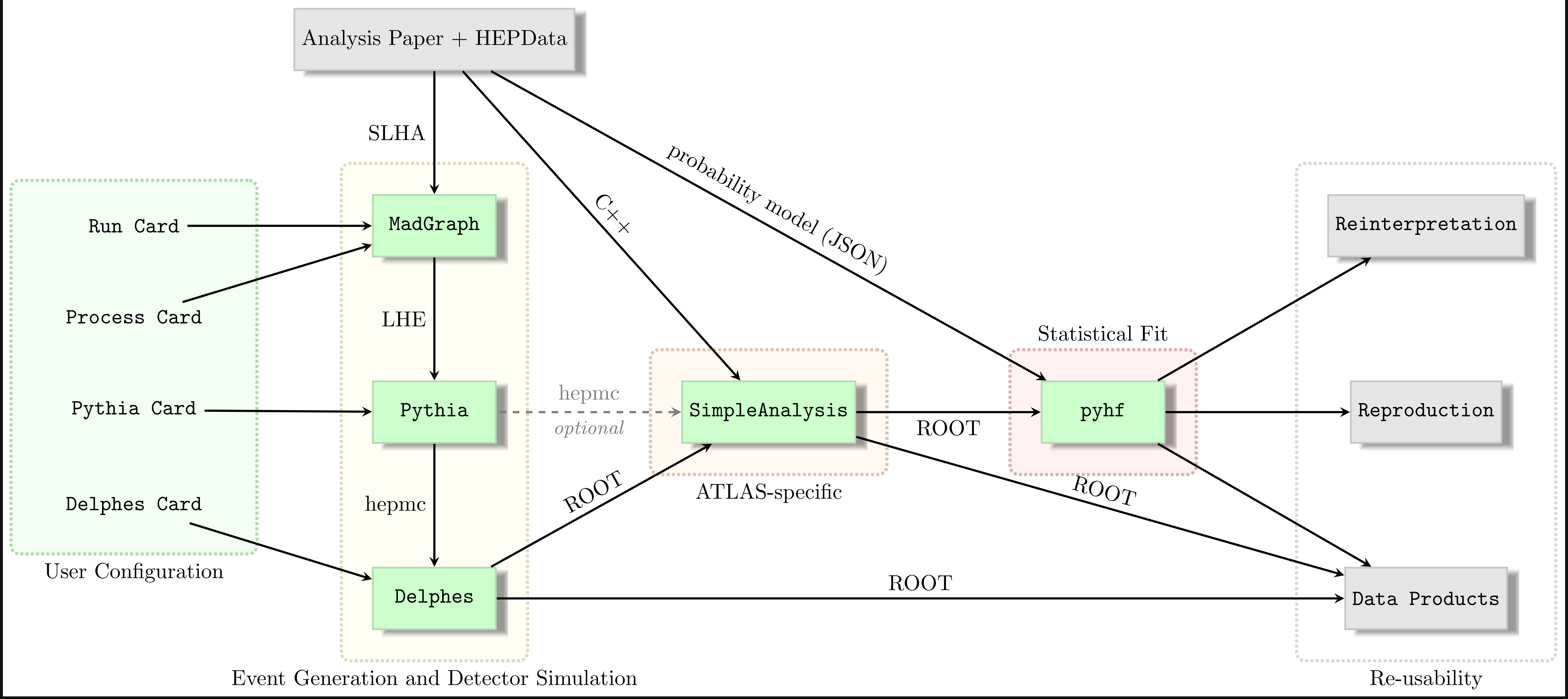
What is mapyde?

- MaPyDe: a highly configurable pipelining framework for recycling public analyses
  - named after common event generation and simulation toolkits: MadGraph, Pythia8, and Delphes
- SW dependencies: python, docker/singularity/apptainer
- Additional tools exist (and new ones can be supported)
  - SimpleAnalysis: ATL-PHYS-PUB-2022-017
  - pyhf: doi://10.21105/joss.02823

ℹ **See my tutorial last year at RiF2022**

approachable pheno tooling for students(!)

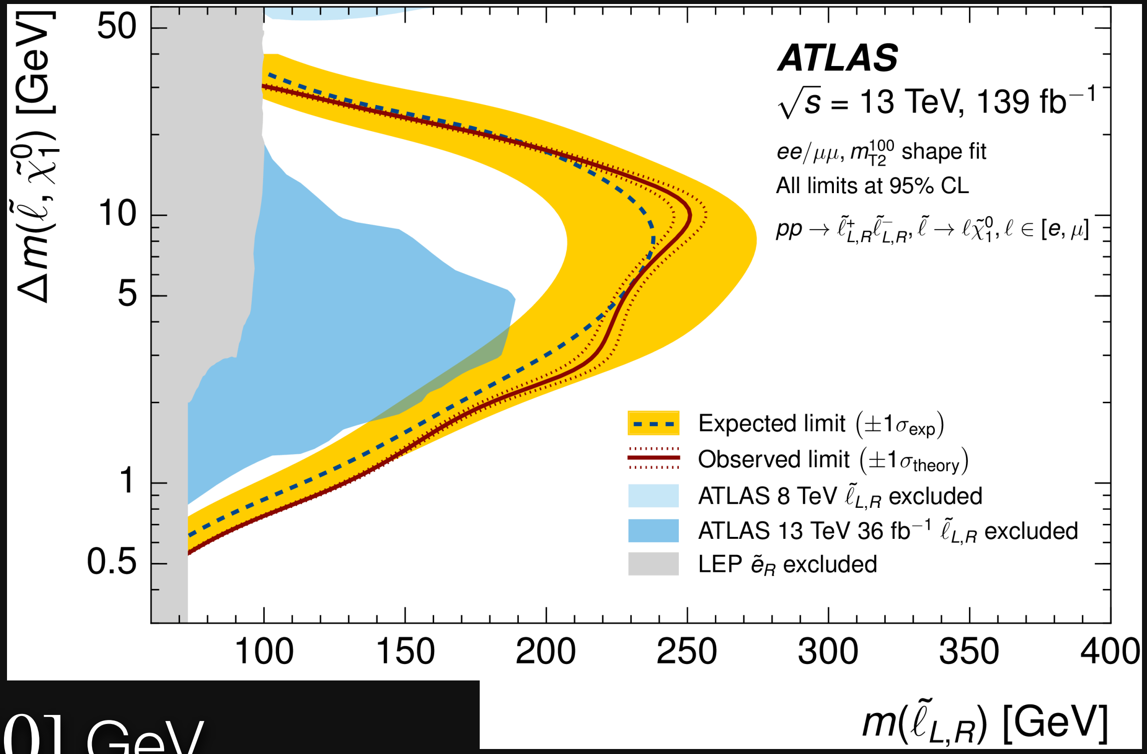
Workflow


Analysis Paper + HEPData
SLHA
C++
probability model (JSON)
Run Card
Process Card
Pythia Card
Delphes Card
User Configuration
MadGraph
LHE
Pythia
hepmc
Delphes
Event Generation and Detector Simulation
hepmc
optional
SimpleAnalysis
ATLAS-specific
ROOT
ROOT
Statistical Fit
pyhf
ROOT
ROOT
Reinterpretation
Reproduction
Data Products
Re-usability

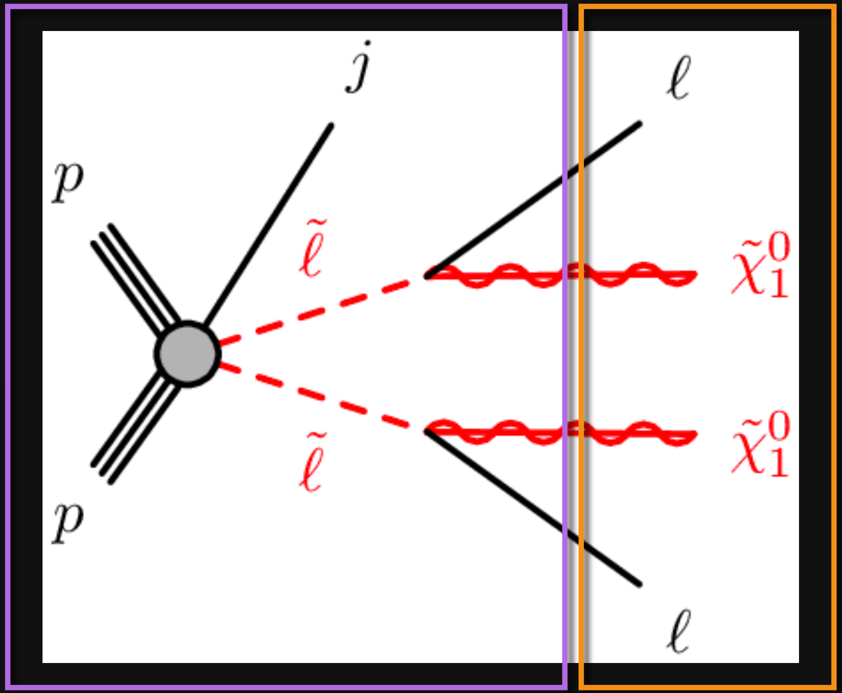
arXiv:1911.12606

A reminder…

(degenerate) sleptons



sleptons

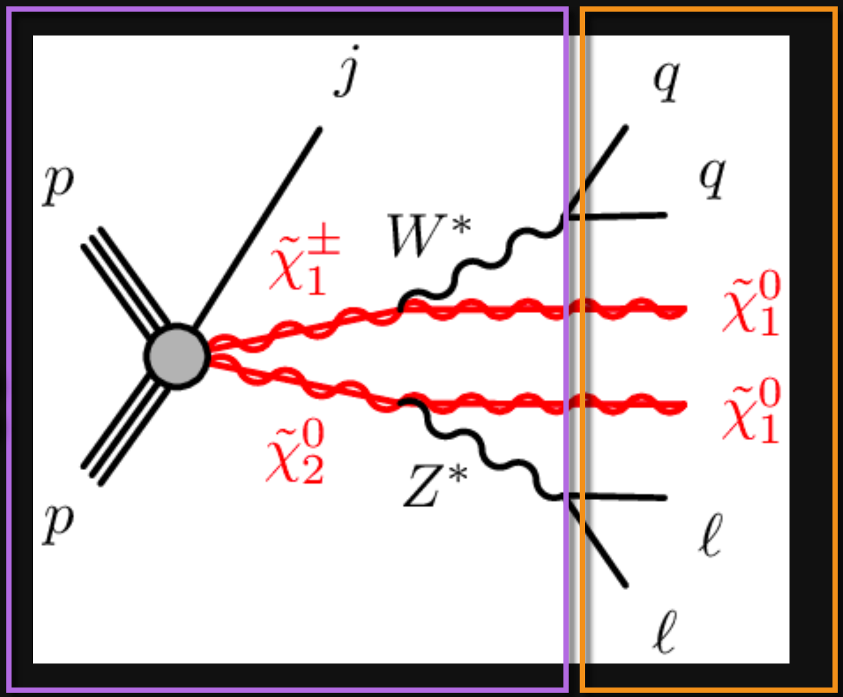


$\Delta m(\tilde\chi_2^0,\tilde\chi_1^0) \in [1,60]$ GeV

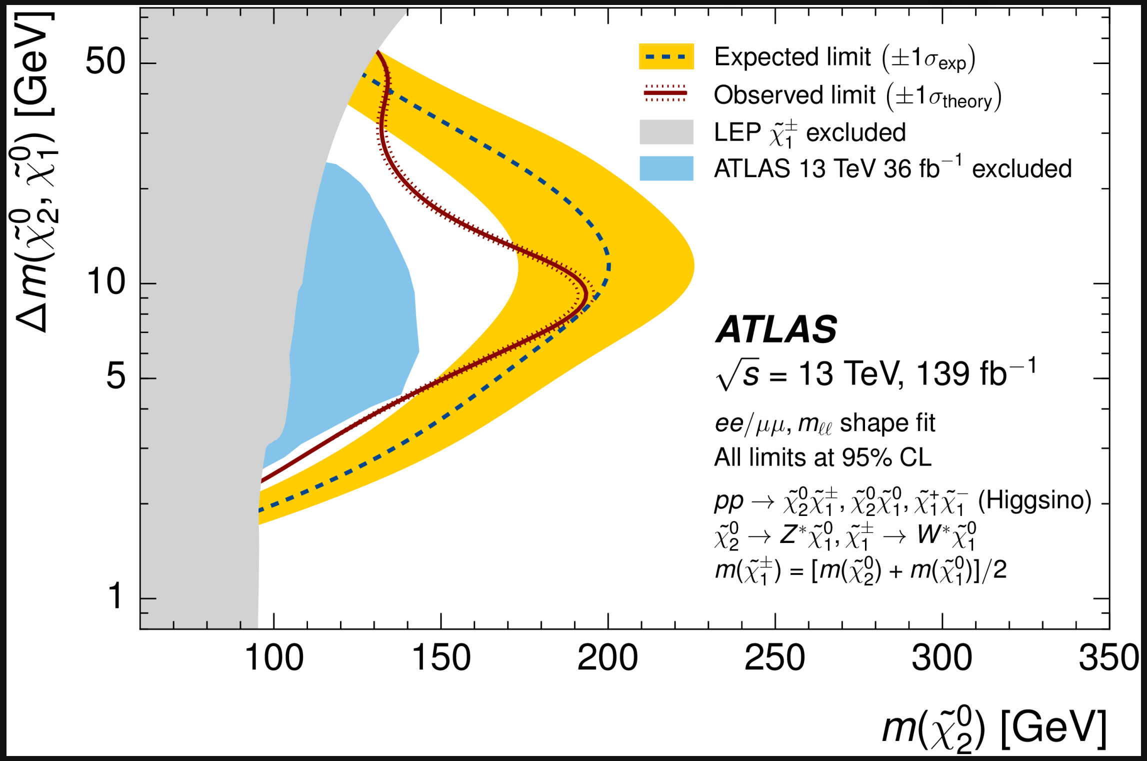
signature: $2\ell + 3j + E_T^{miss}$

"pure-Higgsino"

Higgsinos



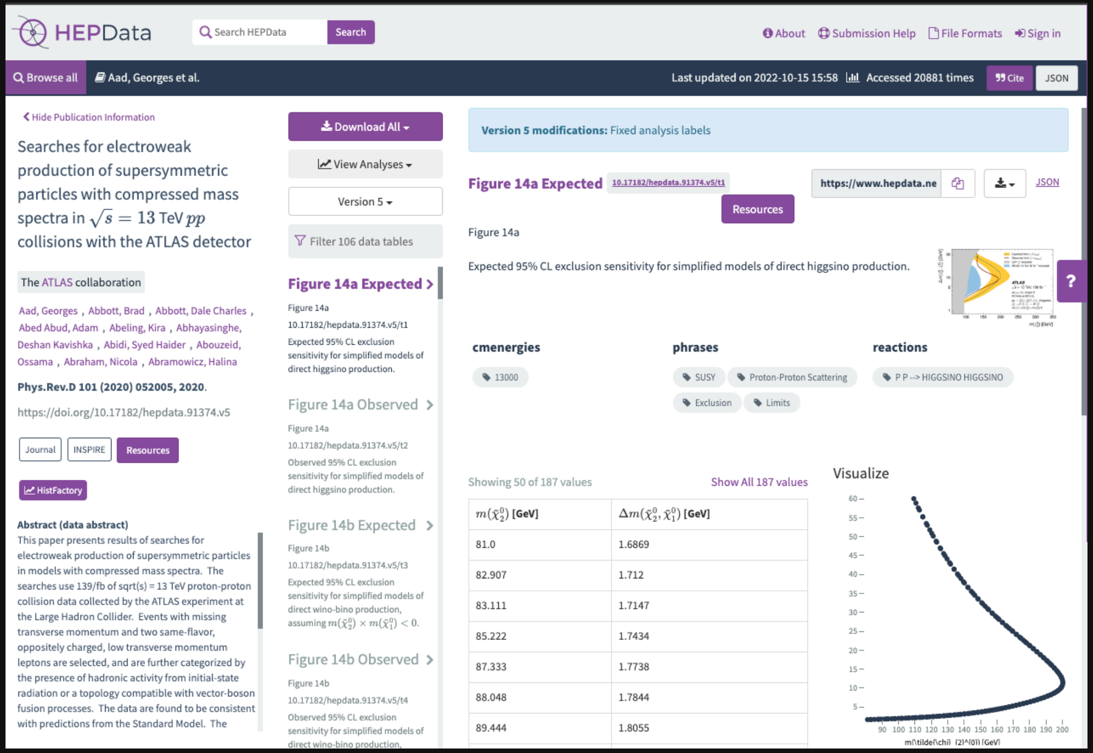
signature



model

Let's use new models and reinterpret!

Public data



Aad, Georges et al.

Last updated on 2022-10-15 15:58 Accessed 20881 times

Hide Publication Information

Searches for electroweak production of supersymmetric particles with compressed mass spectra in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector

The ATLAS collaboration

Aad, Georges , Abbott, Brad , Abbott, Dale Charles , Abed Abud, Adam , Abeling, Kira , Abhayasinghe, Deshan Kavishka , Abidi, Syed Haider , Abouzeid, Ossama , Abraham, Nicola , Abramowicz, Halina

Phys.Rev.D 101 (2020) 052005, 2020.

https://doi.org/10.17182/hepdata.91374.v5

Abstract (data abstract)

This paper presents results of searches for electroweak production of supersymmetric particles in models with compressed mass spectra. The searches use 139/fb of sqrt(s) = 13 TeV proton-proton collision data collected by the ATLAS experiment at the Large Hadron Collider. Events with missing transverse momentum and two same-flavor, oppositely charged, low transverse momentum leptons are selected, and are further categorized by the presence of hadronic activity from initial-state radiation or a topology compatible with vector-boson fusion processes. The data are found to be consistent with predictions from the Standard Model. The

Figure 14a Expected

Figure 14a

10.17182/hepdata.91374.v5/t1

Expected 95% CL exclusion sensitivity for simplified models of direct higgsino production.

Figure 14a Observed

Figure 14a

10.17182/hepdata.91374.v5/t2

Observed 95% CL exclusion sensitivity for simplified models of direct higgsino production.

Figure 14b Expected

Figure 14b

10.17182/hepdata.91374.v5/t3

Expected 95% CL exclusion sensitivity for simplified models of direct wino-bino production, assuming $m(\tilde{\chi}_2^0) \times m(\tilde{\chi}_1^0) < 0$.

Figure 14b Observed

Figure 14b

10.17182/hepdata.91374.v5/t4

Observed 95% CL exclusion sensitivity for simplified models of direct wino-bino production,

Version 5 modifications: Fixed analysis labels

Figure 14a Expected 10.17182/hepdata.91374.v5/t1 https://www.hepdata.ne

Figure 14a

Expected 95% CL exclusion sensitivity for simplified models of direct higgsino production.

cmenergies

13000

phrases

SUSY Proton-Proton Scattering Exclusion Limits

reactions

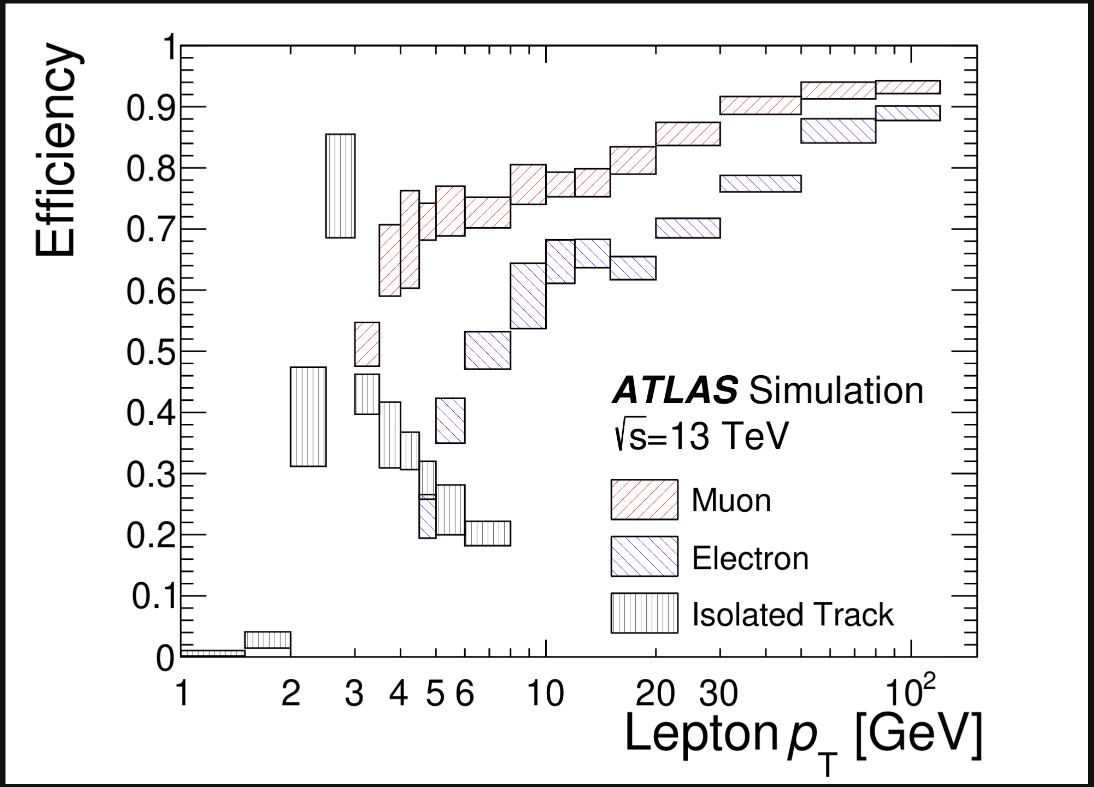
P P --> HIGGSINO HIGGSINO

Showing 50 of 187 values Show All 187 values

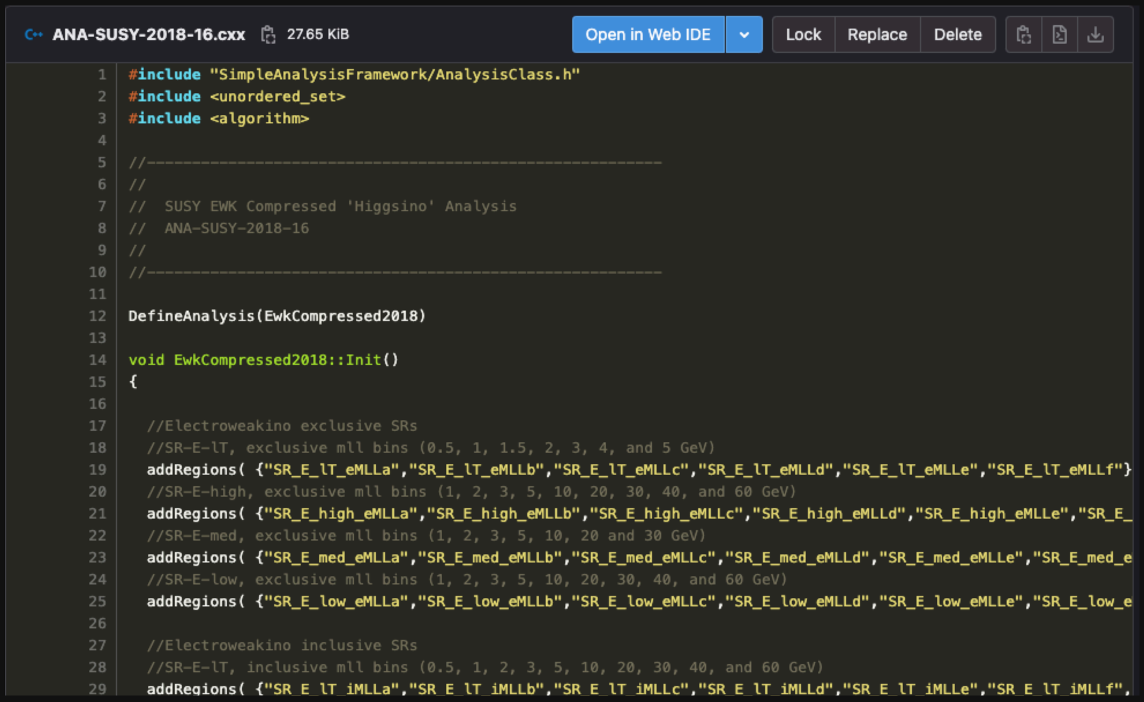
$m(\tilde{\chi}_2^0)$ [GeV]	$\Delta m(\tilde{\chi}_2^0, \tilde{\chi}_1^0)$ [GeV]
81.0	1.6869
82.907	1.712
83.111	1.7147
85.222	1.7434
87.333	1.7738
88.048	1.7844
89.444	1.8055

Visualize

- HEPData (right) contains:
  - selection acceptances and detector efficiencies for soft leptons (below)
  - HistFactory probability models in a JSON format
  - Exclusion contours
  - Cutflows/Yields



- In addition, SimpleAnalysis contains the generator-level selections (below) for this analysis
  - https://simpleanalysis.docs.cern.ch/analyses/
  - Provided as a C++ file, pre-compiled docker images available from ATLAS



ANA-SUSY-2018-16.cxx 27.65 KiB

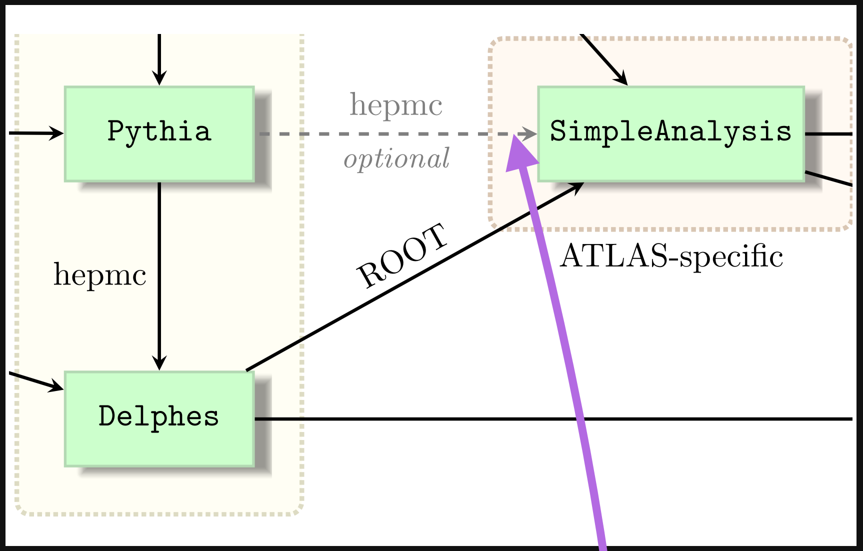
```
#include "SimpleAnalysisFramework/AnalysisClass.h"
#include <unordered_set>
#include <algorithm>

//-----------------------------------------------------------
//
//  SUSY EWK Compressed 'Higgsino' Analysis
//  ANA-SUSY-2018-16
//
//-----------------------------------------------------------

DefineAnalysis(EwkCompressed2018)

void EwkCompressed2018::Init()
{

  //Electroweakino exclusive SRs
  //SR-E-lT, exclusive mll bins (0.5, 1, 1.5, 2, 3, 4, and 5 GeV)
  addRegions( {"SR_E_lT_eMLLa","SR_E_lT_eMLLb","SR_E_lT_eMLLc","SR_E_lT_eMLLd","SR_E_lT_eMLLe","SR_E_lT_eMLLf"}
  //SR-E-high, exclusive mll bins (1, 2, 3, 5, 10, 20, 30, 40, and 60 GeV)
  addRegions( {"SR_E_high_eMLLa","SR_E_high_eMLLb","SR_E_high_eMLLc","SR_E_high_eMLLd","SR_E_high_eMLLe","SR_E_
  //SR-E-med, exclusive mll bins (1, 2, 3, 5, 10, 20 and 30 GeV)
  addRegions( {"SR_E_med_eMLLa","SR_E_med_eMLLb","SR_E_med_eMLLc","SR_E_med_eMLLd","SR_E_med_eMLLe","SR_E_med_e
  //SR-E-low, exclusive mll bins (1, 2, 3, 5, 10, 20, 30, 40, and 60 GeV)
  addRegions( {"SR_E_low_eMLLa","SR_E_low_eMLLb","SR_E_low_eMLLc","SR_E_low_eMLLd","SR_E_low_eMLLe","SR_E_low_e

  //Electroweakino inclusive SRs
  //SR-E-lT, inclusive mll bins (0.5, 1, 2, 3, 5, 10, 20, 30, 40, and 60 GeV)
  addRegions( {"SR_E_lT_iMLLa","SR_E_lT_iMLLb","SR_E_lT_iMLLc","SR_E_lT_iMLLd","SR_E_lT_iMLLe","SR_E_lT_iMLLf",
```

So how do we proceed?

- ✓ **Have public data for the analysis, and a 🐍🐳–enabled workflow**
  - images/containers for MadGraph, Pythia, MadSpin, Delphes, SimpleAnalysis, pyhf, and intermediate output conversions
  - ATLAS documented SimpleAnalysis (for truth-level selections + deriving detector acceptances/selection efficiencies)
  - pyhf for statistical fitting
  - mapyde for chaining all of this together
- ✓ **Reproduction of existing results**
  - tune differences between our setup and the ATLAS setup (event generation, reconstruction, acceptances and efficiencies, cross-section)
- ✓ **Reinterpretation of (new) alternate theories**
  - for each model considered (sleptons and Higgsinos), identify a new model we want to explore current experimental sensitivity

ATL-PHYS-PUB-2022-017

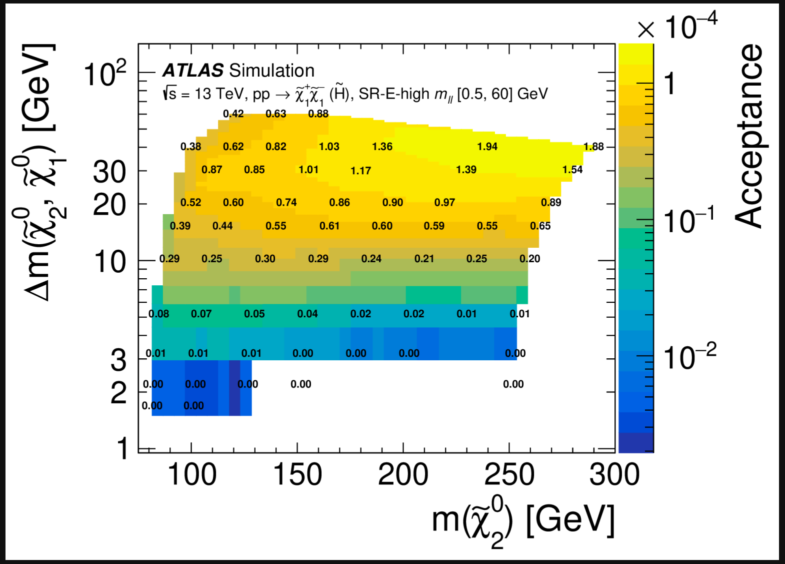
Acceptance Comparisons



- ATLAS has a different setup for producing their signal models: MadGraph with ≲ **2-jets + MadSpin**
  - Detector Reconstruction/Simulation: **ATLAS Internal**

- Our paper uses: MadGraph with ≲ **1-jet + Pythia**
  - *note: mapyde supports MadSpin (we opted not to use it)*
  - Detector Reconstruction/Simulation: **Delphes** (with default ATLAS card)

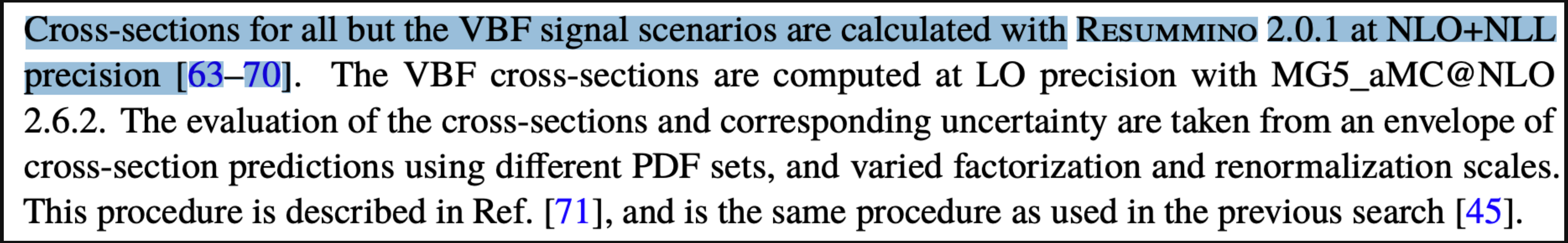
- Compare the acceptances from our workflow to acceptances published by ATLAS — at most 10% difference (in the compressed-mass region)



$$\underset{\text{detector acceptance}}{A} \times \underset{\text{selection efficiency}}{\varepsilon} \times \underset{\text{branching ratio}}{\mathrm{BR}} \times \underset{\text{cross section}}{\sigma} \times \underset{\text{integrated luminosity}}{\mathscr{L}_{\mathrm{int}}}$$

arXiv:1911.12606

Tuning Cross-Section



Cross-sections for all but the VBF signal scenarios are calculated with Resummino 2.0.1 at NLO+NLL precision [63–70]. The VBF cross-sections are computed at LO precision with MG5_aMC@NLO 2.6.2. The evaluation of the cross-sections and corresponding uncertainty are taken from an envelope of cross-section predictions using different PDF sets, and varied factorization and renormalization scales. This procedure is described in Ref. [71], and is the same procedure as used in the previous search [45].

- In this analysis, ATLAS published NLO cross-sections calculated with **Resummino**
- Our cross-sections are calculated at LO ($\lesssim$ 1 jet) with **MadGraph**
- Derive a ***k*-factor** for each signal point that **we apply** in order to adjust for quantum (loop) corrections

$$k \equiv \frac{\sigma_{\mathrm{NLO}}}{\sigma_{\mathrm{LO,\ 0\,jet}}}$$

ATLAS provides

Special mapyde run with 0-jet at LO

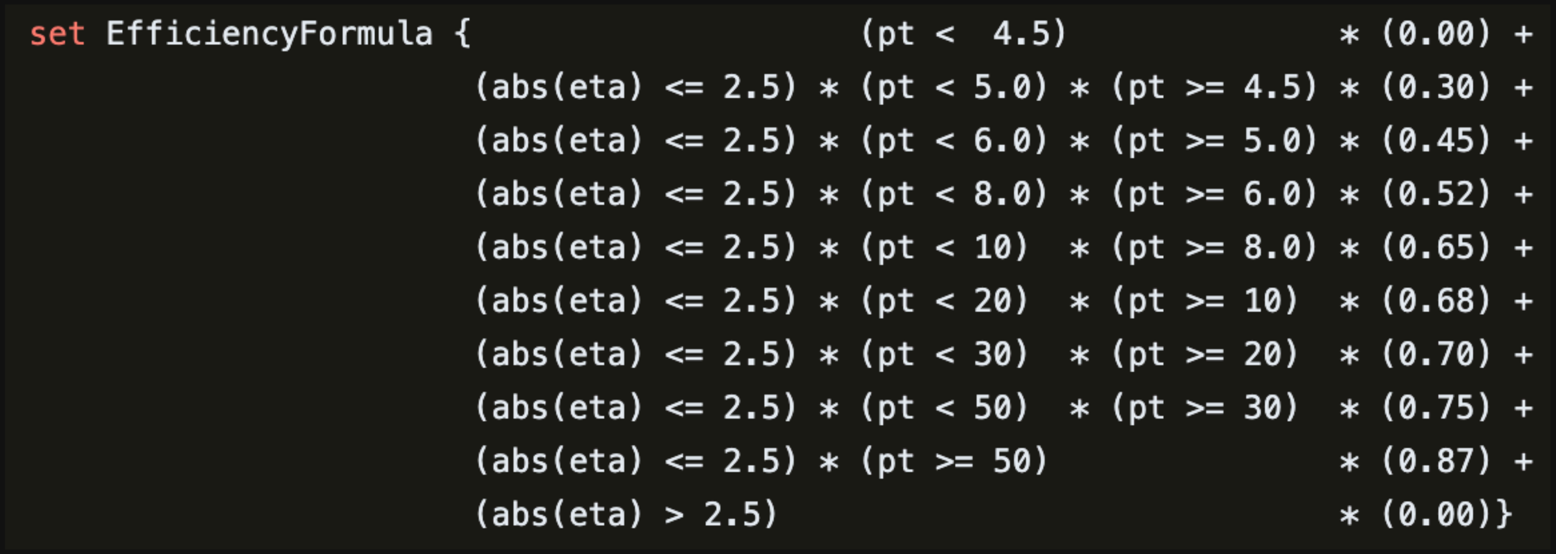
$$\sigma_{\mathrm{eff}} = k \times \sigma_{\mathrm{LO,\ 1\,jet}}$$

mass-independent $k = 1.18$

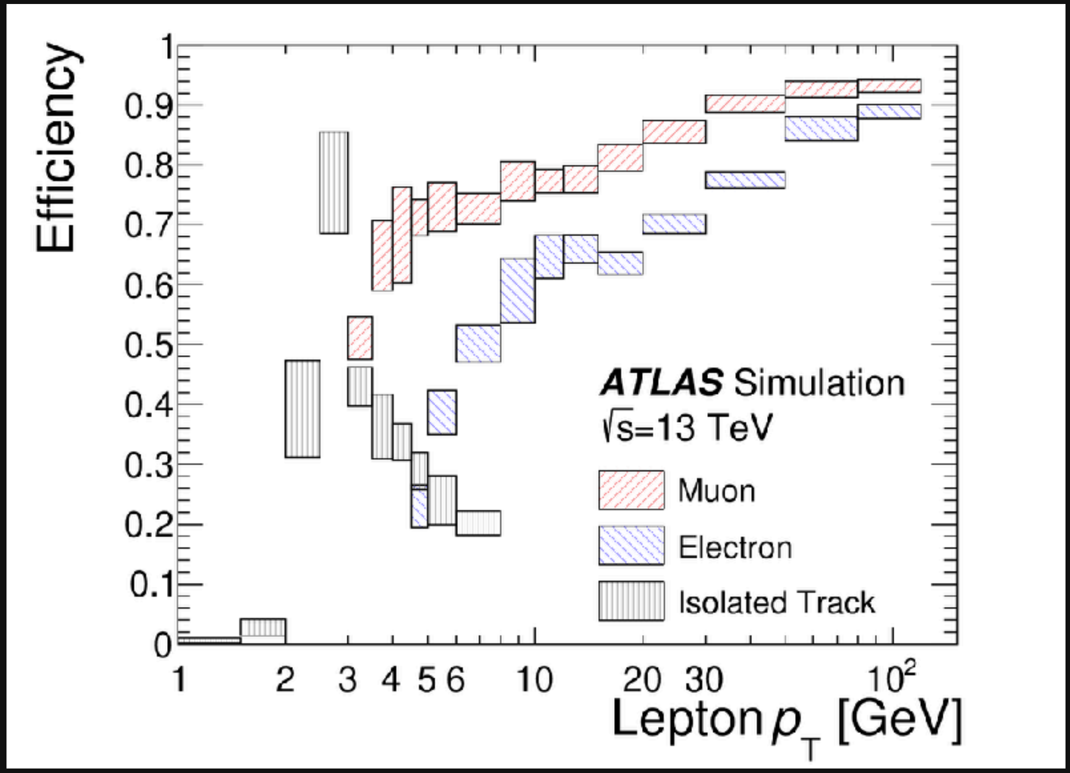
scipp-atlas/mapyde@e4da3779/cards/delphes/delphes_card_ATLAS_lowptleptons_sleptons_notrackineffic.tcl#L472-L554

Tuning Reconstruction

- Then, tune default ATLAS card by incorporating the (soft) lepton reconstruction efficiencies



```
set EfficiencyFormula {                        (pt <  4.5)              * (0.00) +
                        (abs(eta) <= 2.5) * (pt < 5.0) * (pt >= 4.5) * (0.30) +
                        (abs(eta) <= 2.5) * (pt < 6.0) * (pt >= 5.0) * (0.45) +
                        (abs(eta) <= 2.5) * (pt < 8.0) * (pt >= 6.0) * (0.52) +
                        (abs(eta) <= 2.5) * (pt < 10)  * (pt >= 8.0) * (0.65) +
                        (abs(eta) <= 2.5) * (pt < 20)  * (pt >= 10)  * (0.68) +
                        (abs(eta) <= 2.5) * (pt < 30)  * (pt >= 20)  * (0.70) +
                        (abs(eta) <= 2.5) * (pt < 50)  * (pt >= 30)  * (0.75) +
                        (abs(eta) <= 2.5) * (pt >= 50)                * (0.87) +
                        (abs(eta) > 2.5)                              * (0.00)}
```



reconstructing (very) soft leptons

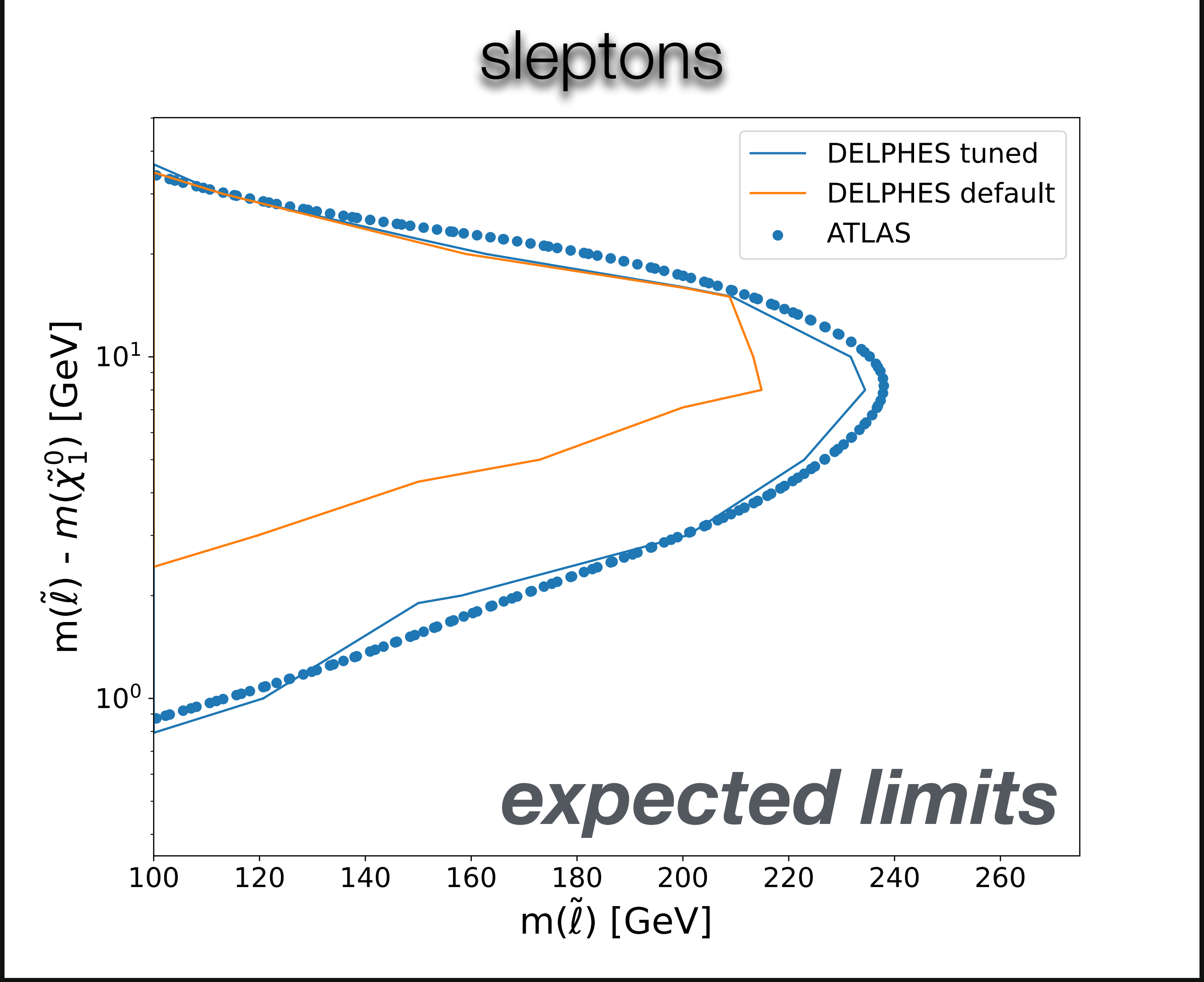
- This means updating the TCL (config) that Delphes uses

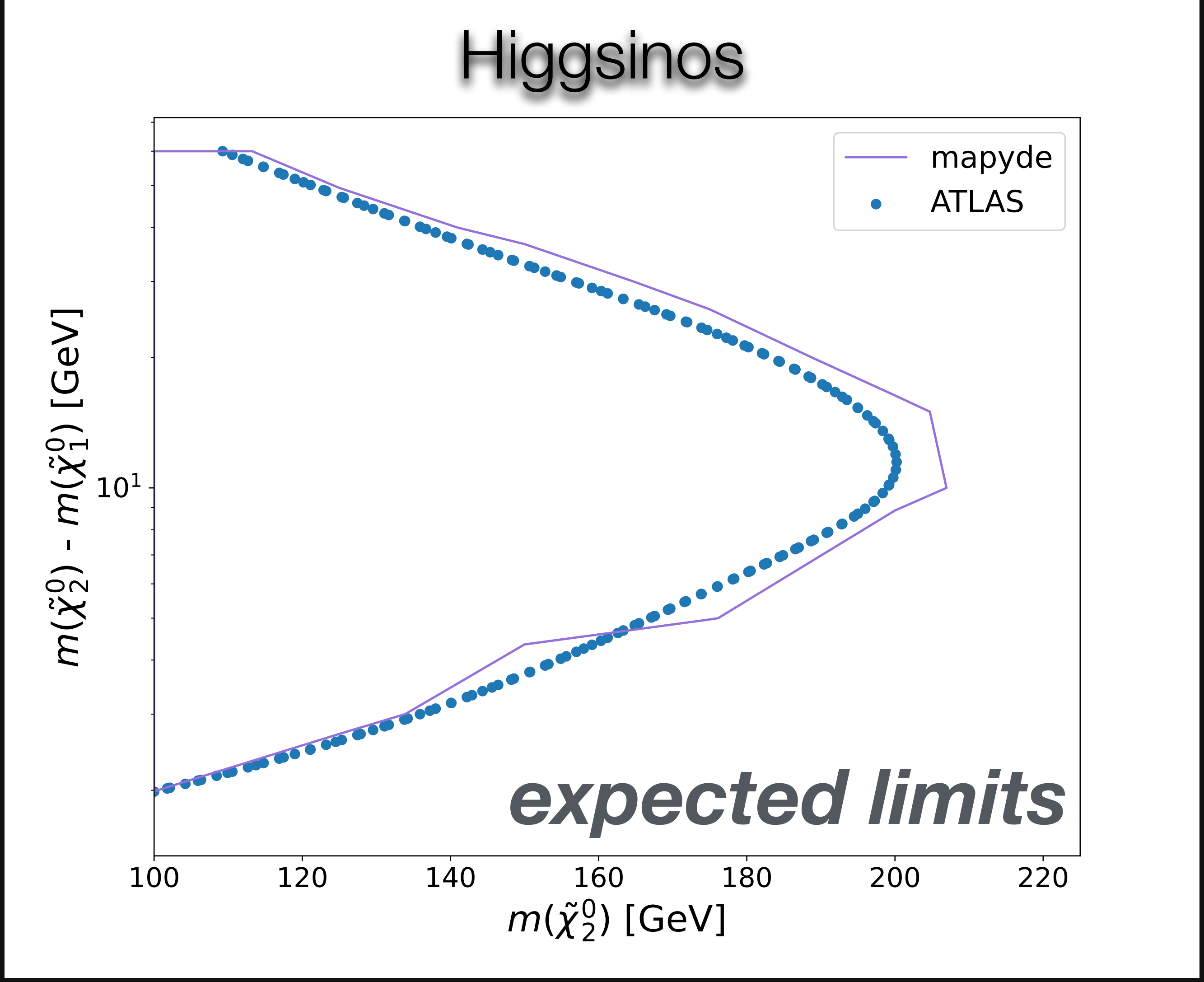
include comment on upper bound

arXiv:2306.11055 arXiv:1911.12606

mapyde

Reproduction





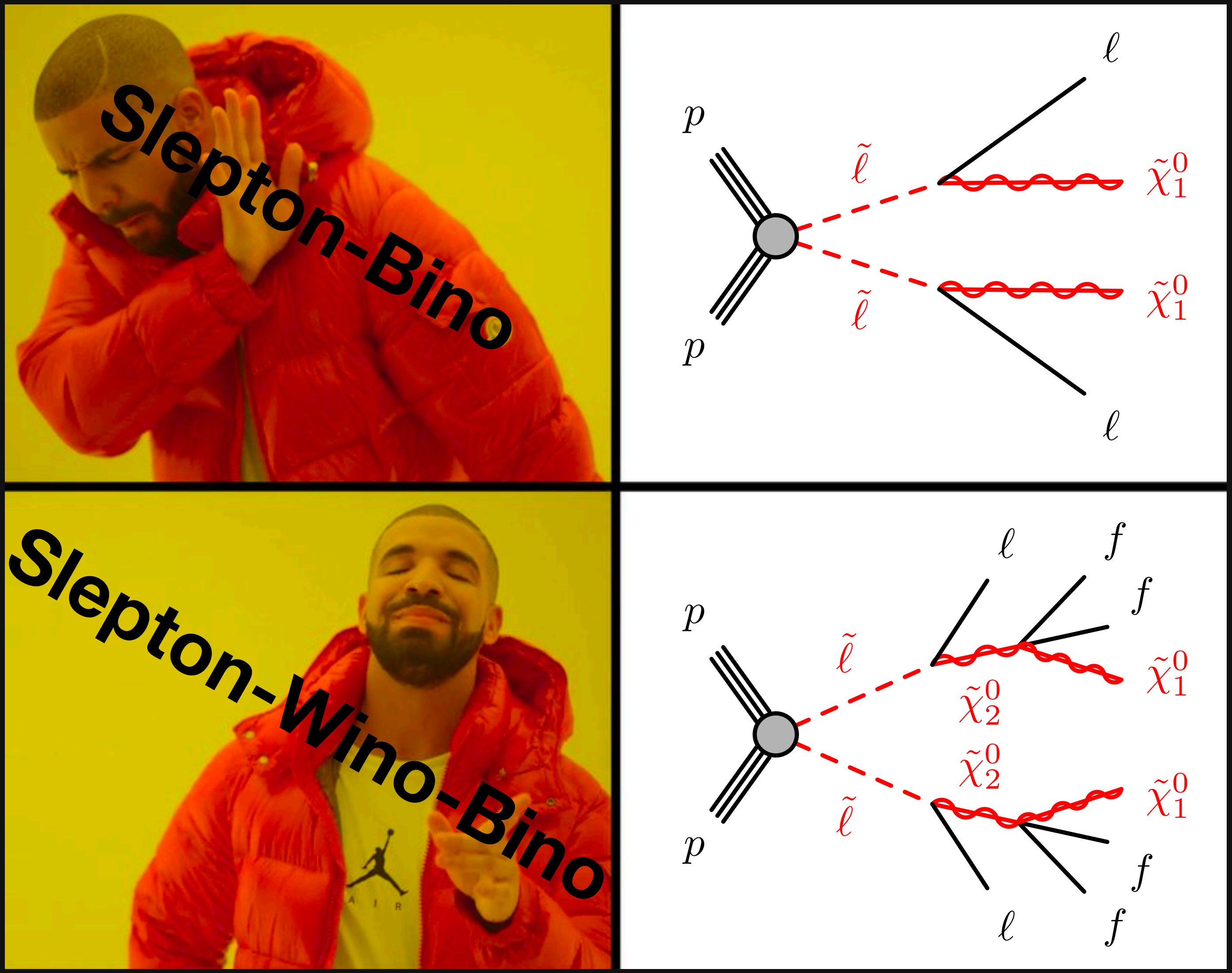
- Sleptons: lepton reconstruction efficiencies are an essential piece of the workflow
- Higgsinos: also includes lepton reconstruction efficiencies (just not overlaid)

✓ Our workflow… works.

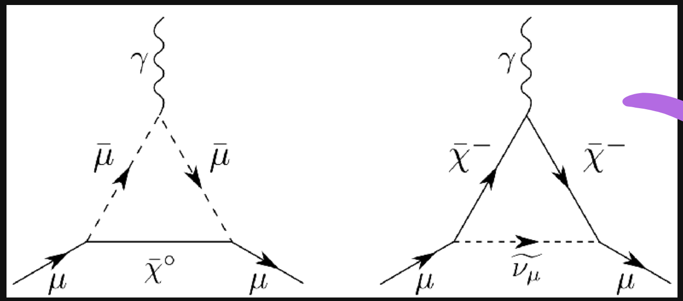
arXiv:2306.11055 arXiv:1911.12606 arXiv:2303.01523

MaPyDe
REUSE, REPRODUCE, REINTERPRET
mapyde

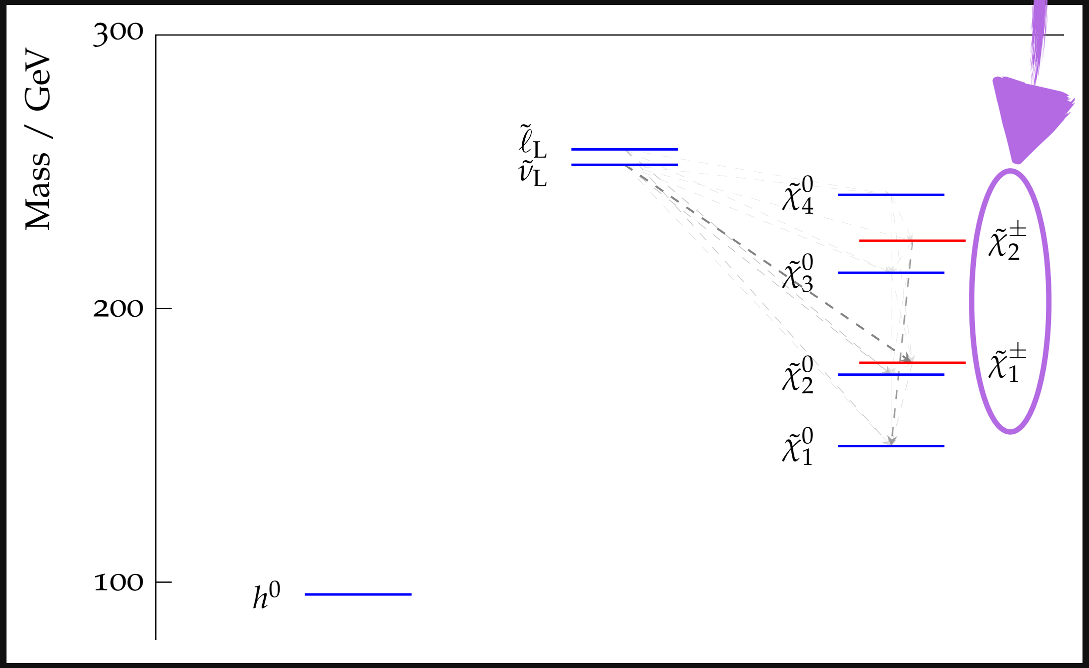
Reinterpretation (sleptons)



- ATLAS did a “simplified model” search — is it too simple?



$\tilde{\mu}^\pm/\tilde{\chi}^0$ or $\tilde{\chi}^\pm/\tilde{\nu}_\mu$

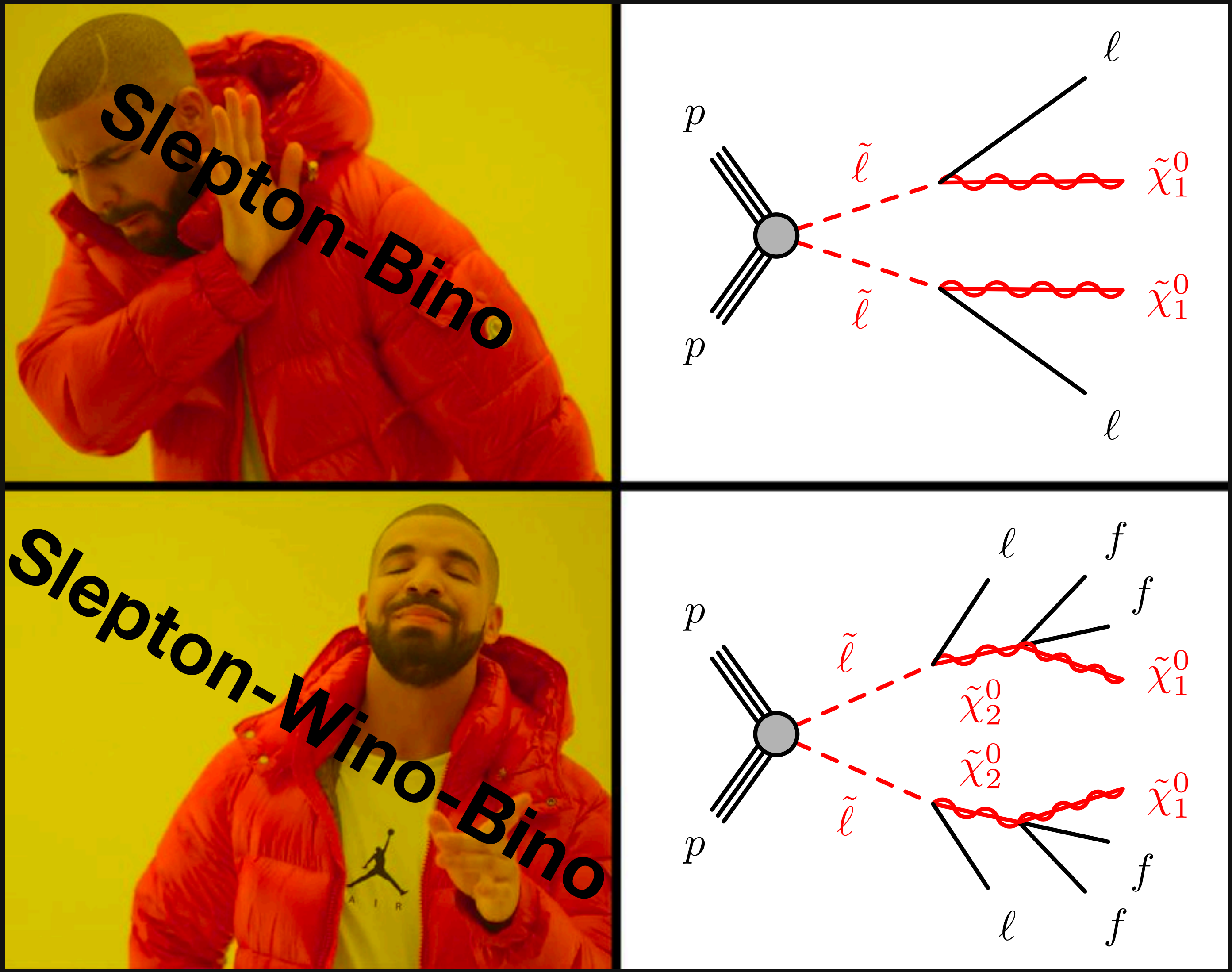


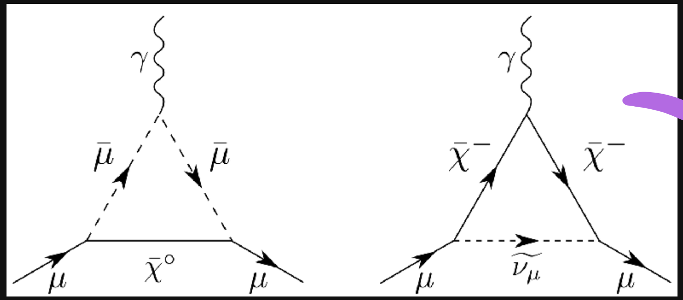
- Including wino expands parameter-space, perhaps contributions to g-2?

arXiv:2306.11055 arXiv:1911.12606 arXiv:2303.01523

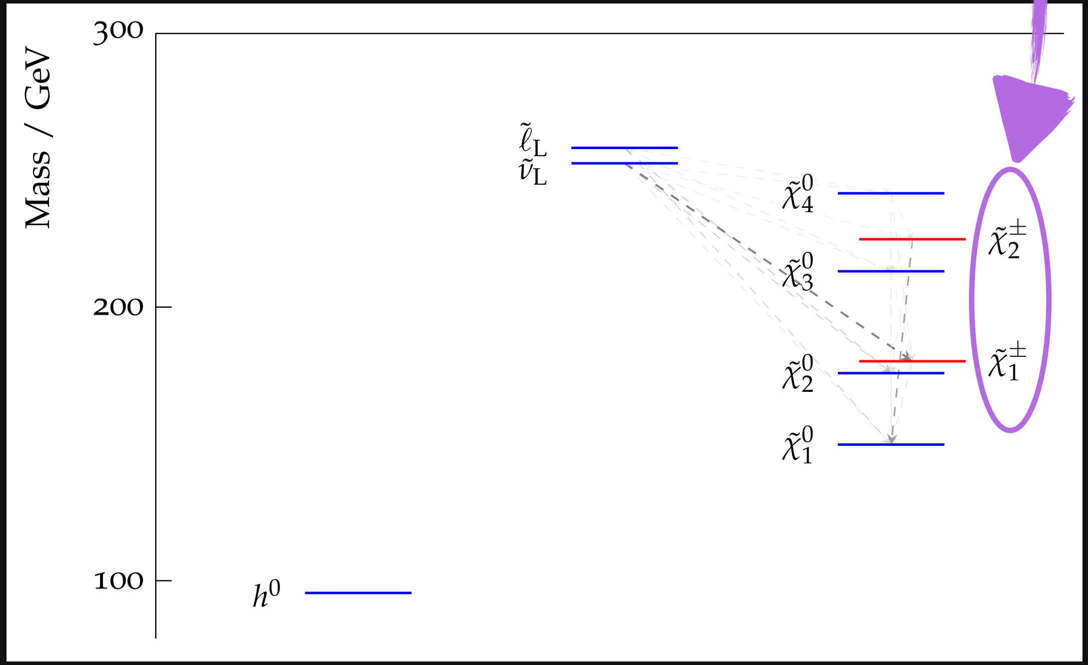
MaPyDe
REUSE, REPRODUCE, REINTERPRET
mapyde

Reinterpretation (sleptons)





$\tilde{\mu}^{\pm}/\tilde{\chi}^0$ or $\tilde{\chi}^{\pm}/\tilde{\nu}_\mu$



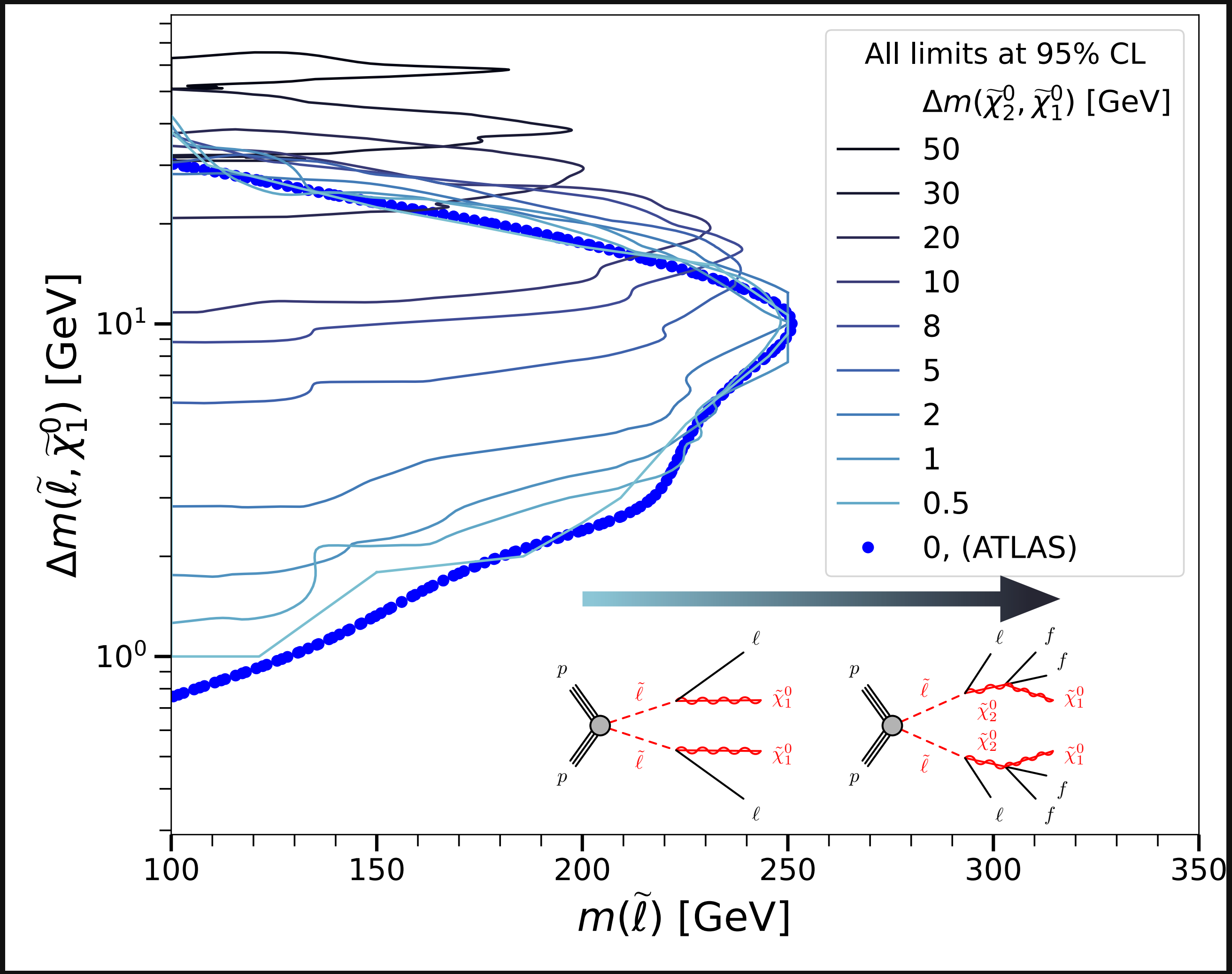
- Including wino expands parameter-space, perhaps contributions to g-2?

arXiv:2306.11055 arXiv:1911.12606 arXiv:2303.01523

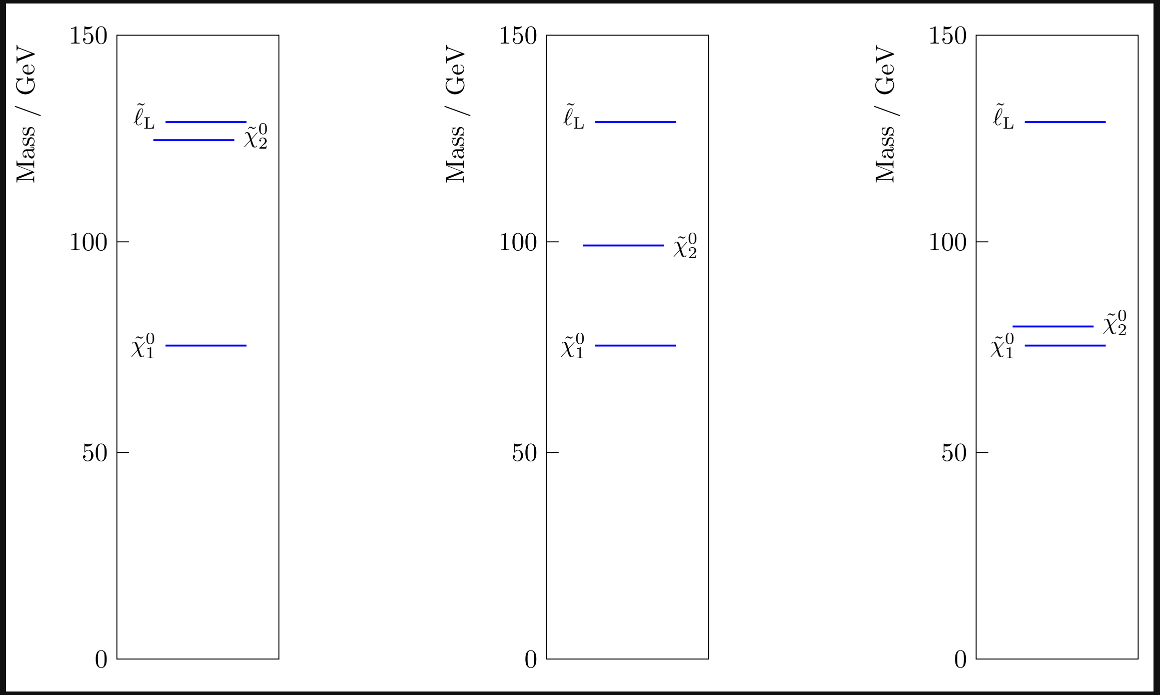
Reinterpretation (sleptons)

mapyde

observed limits



- lepton p_T determined by gap of:
  - slepton-bino: $\tilde{\ell} - \tilde{\chi}_1^0$
  - slepton-wino-bino: $\tilde{\ell} - \tilde{\chi}_2^0$



- each contour is bounded:
  - **below**: weakened sensitivity at smaller $\tilde{\ell} - \tilde{\chi}_2^0$ due to softer SM leptons/lower lepton efficiencies
  - **above**: reduced σ due to larger slepton masses, and selection efficiency (optimized for softer)

- at $\tilde{\chi}_2^0 - \tilde{\chi}_1^0$ splittings that are:
  - **small**: resembles slepton-bino ATLAS results (degeneracy of neutralinos)
  - **large**: stronger constraints at large $\tilde{\ell} - \tilde{\chi}_1^0$

arXiv:2306.11055 arXiv:1911.12606 arXiv:2303.01523

Reinterpretation (WBH)

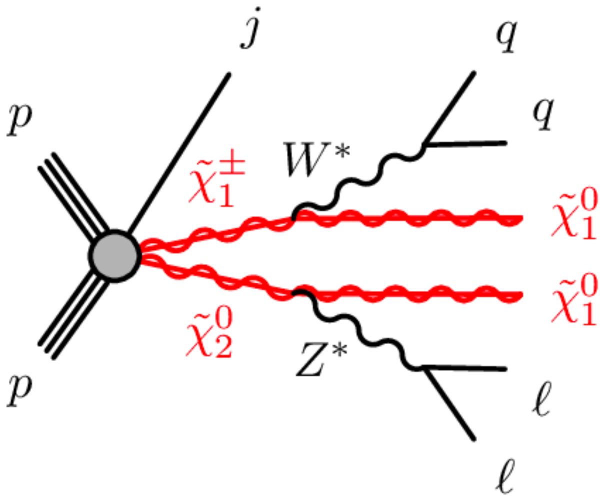
MaPyDe

REUSE, REPRODUCE, REINTERPRET

mapyde

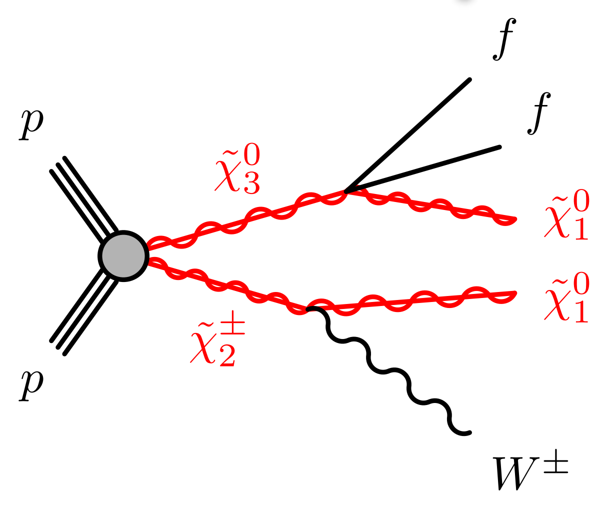
"higgsino"

ATLAS

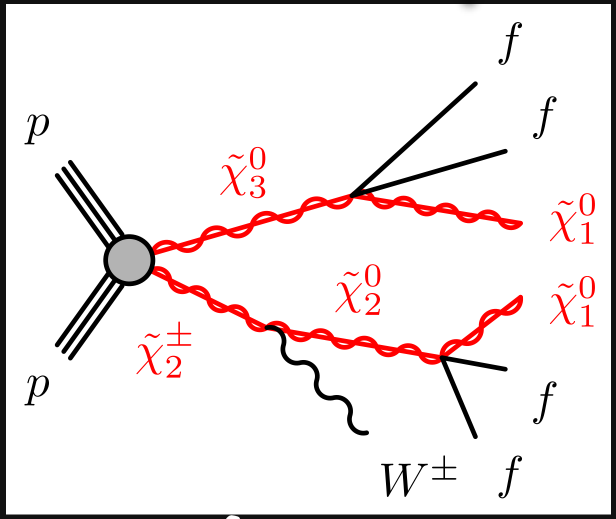


"wino-bino-higgsino" (WBH)

one-step



two-step



ATLAS looked Higgsinos ($\tilde{\chi}_1^0, \tilde{\chi}_1^\pm, \tilde{\chi}_2^0$) where the neutralinos were compressed…

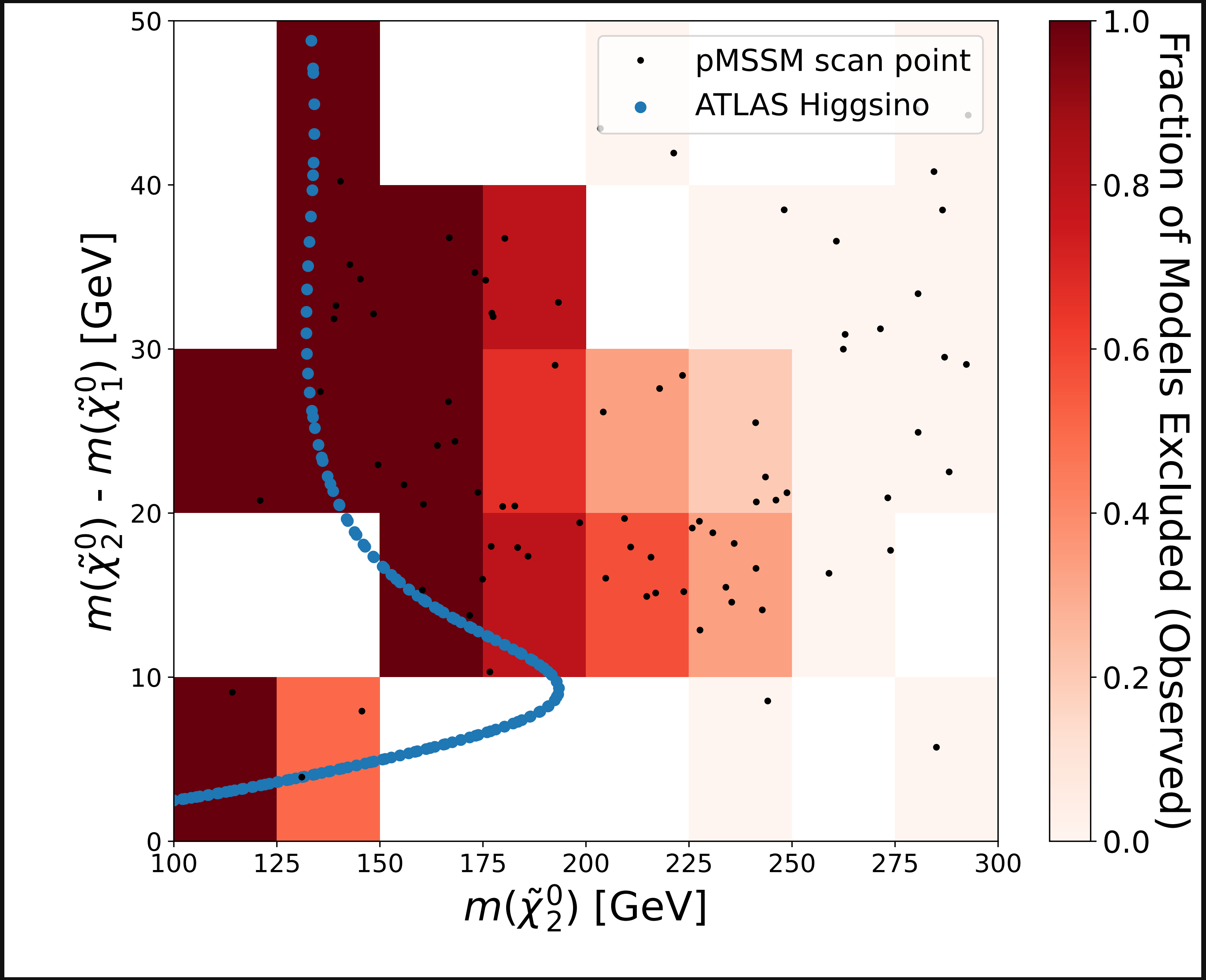
What if the wino-bino terms (M_1, M_2) were "close by" μ so that we have an additional neutralino and chargino ($\tilde{\chi}_2^\pm, \tilde{\chi}_3^0$)?

⚠ Let's run a pMSSM scan and see?

arXiv:2306.11055 arXiv:1911.12606 arXiv:2303.01523

MaPyDe
REUSE, REPRODUCE, REINTERPRET
mapyde

Reinterpretation (WBH)



- Scan 500k points, 81 models survived
  - $m(\tilde{\chi}^0_1) > 100$ GeV
  - $m(\tilde{\chi}^0_3) < 300$ GeV
  - $\Delta m(\tilde{\chi}^0_3, \tilde{\chi}^0_1) < 50$ GeV
  - SPheno: valid
  - FeynHiggs: Higgs mass
  - Micromegas: Dark matter
  - SuperIso: Flavor physics
  - GM2Calc: g-2
- Compared to a pure-Higgsino model, the presence of an additional neutralino $\tilde{\chi}^0_3$ with similar decays to $\tilde{\chi}^0_2$ might be helping

observed limits

⚠ **Stronger constraints at larger mass splittings!**

https://toml.io/en/

The tech stack

“pip”-installable (also exists on conda-forge)

docker/singularity/apptainer are the runtime environments (must exist on the host machine or node)

configuration (apart from individual cards used by various tools) is in TOML

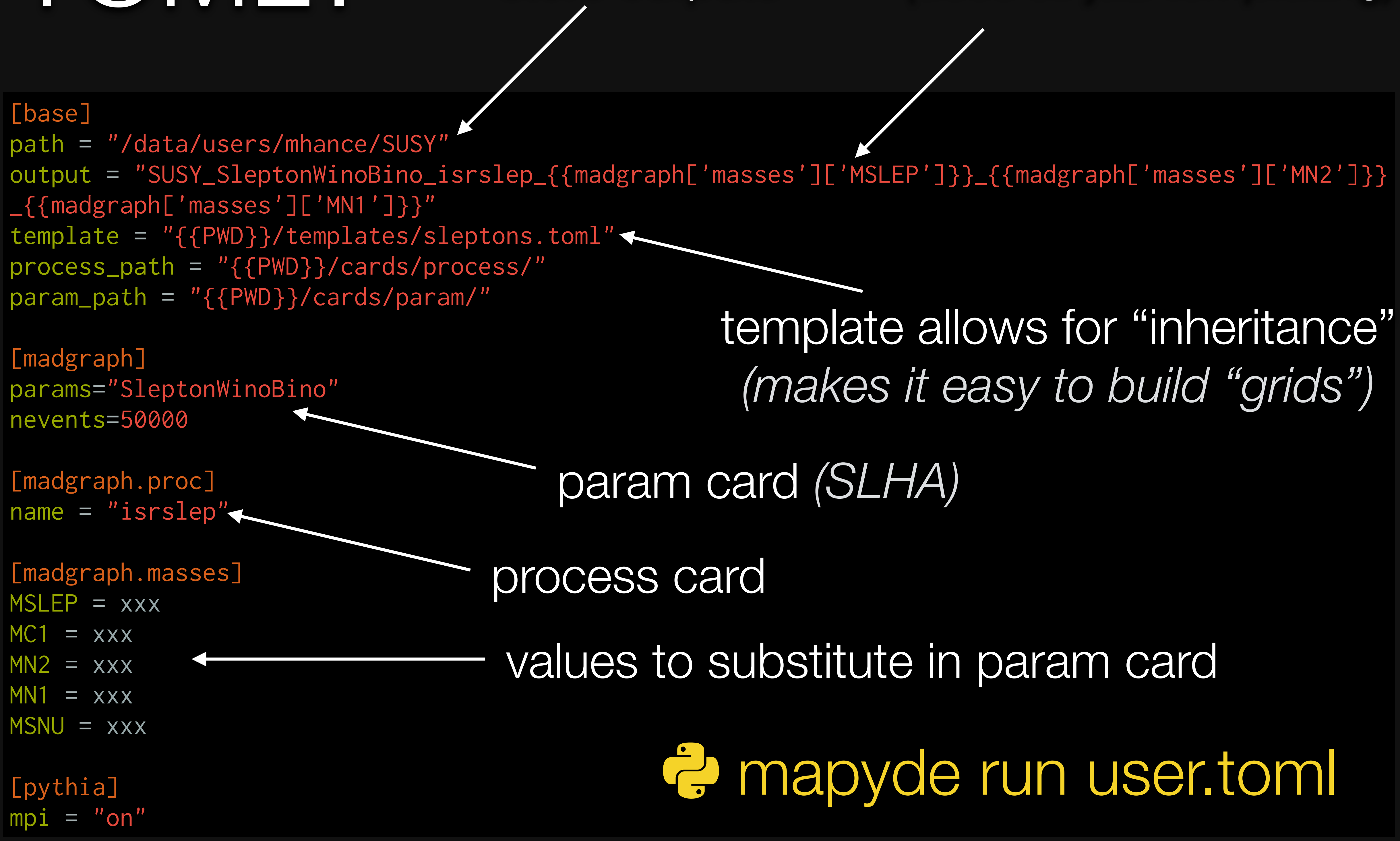
- modern configuration that’s human-readable

https://toml.io/en/

TOML?

top-level folder to store outputs

output folder name *(uses Jinja2 templating)*



```
[base]
path = "/data/users/mhance/SUSY"
output = "SUSY_SleptonWinoBino_isrslep_{{madgraph['masses']['MSLEP']}}_{{madgraph['masses']['MN2']}}
_{{madgraph['masses']['MN1']}}"
template = "{{PWD}}/templates/sleptons.toml"
process_path = "{{PWD}}/cards/process/"
param_path = "{{PWD}}/cards/param/"

[madgraph]
params="SleptonWinoBino"
nevents=50000

[madgraph.proc]
name = "isrslep"

[madgraph.masses]
MSLEP = xxx
MC1  = xxx
MN2  = xxx
MN1  = xxx
MSNU  = xxx

[pythia]
mpi = "on"
```

template allows for “inheritance”
(makes it easy to build “grids”)

param card *(SLHA)*

process card

values to substitute in param card

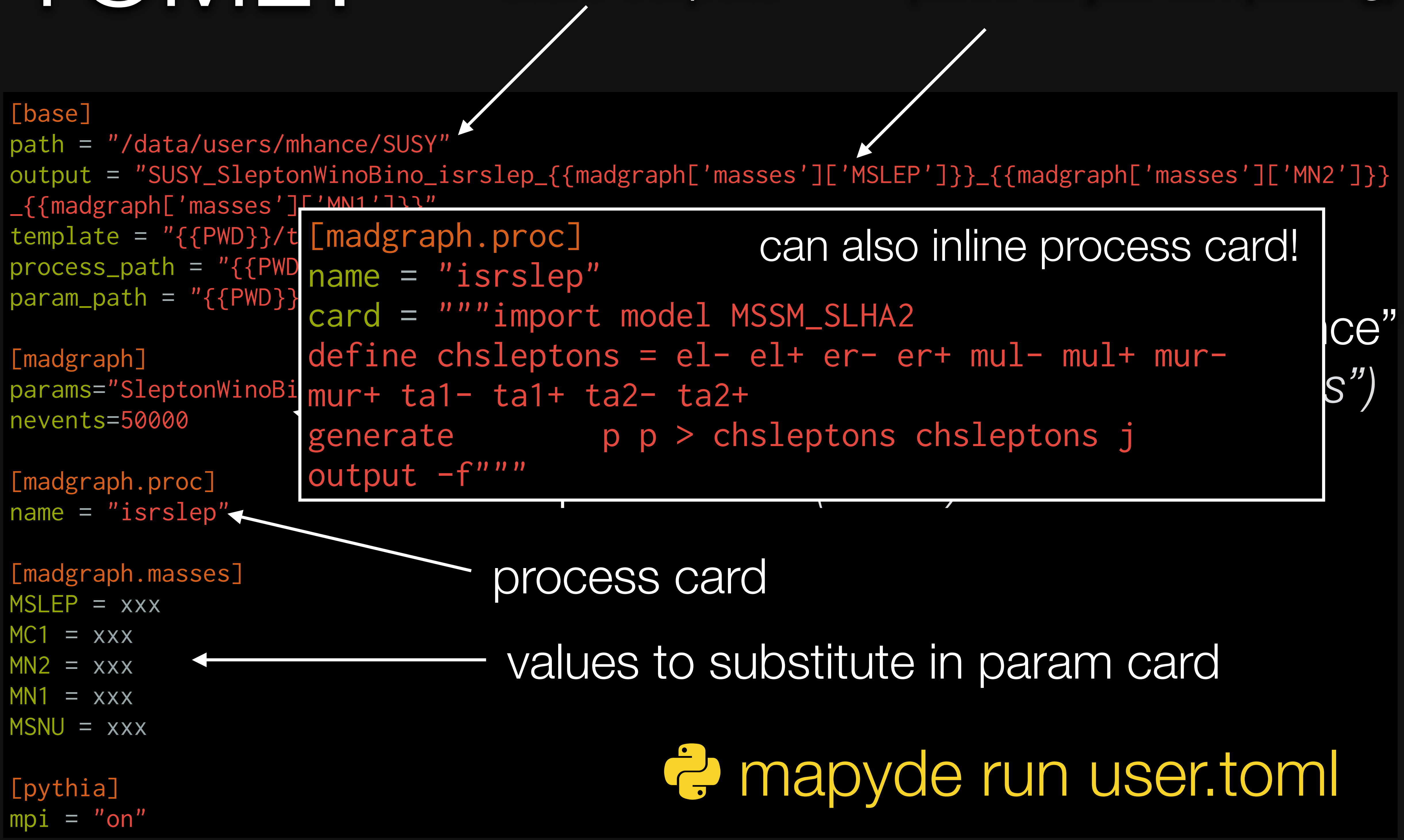
mapyde run user.toml

https://toml.io/en/

TOML?

top-level folder to store outputs

output folder name *(uses Jinja2 templating)*



```
[base]
path = "/data/users/mhance/SUSY"
output = "SUSY_SleptonWinoBino_isrslep_{{madgraph['masses']['MSLEP']}}_{{madgraph['masses']['MN2']}}
_{{madgraph['masses']['MN1']}}"
template = "{{PWD}}/t
process_path = "{{PWD
param_path = "{{PWD}}

[madgraph]
params="SleptonWinoBi
nevents=50000

[madgraph.proc]
name = "isrslep"

[madgraph.masses]
MSLEP = xxx
MC1 = xxx
MN2 = xxx
MN1 = xxx
MSNU = xxx

[pythia]
mpi = "on"
```

```
[madgraph.proc]
name = "isrslep"
card = """import model MSSM_SLHA2
define chsleptons = el- el+ er- er+ mul- mul+ mur-
mur+ ta1- ta1+ ta2- ta2+
generate        p p > chsleptons chsleptons j
output -f"""
```

can also inline process card!

ce”

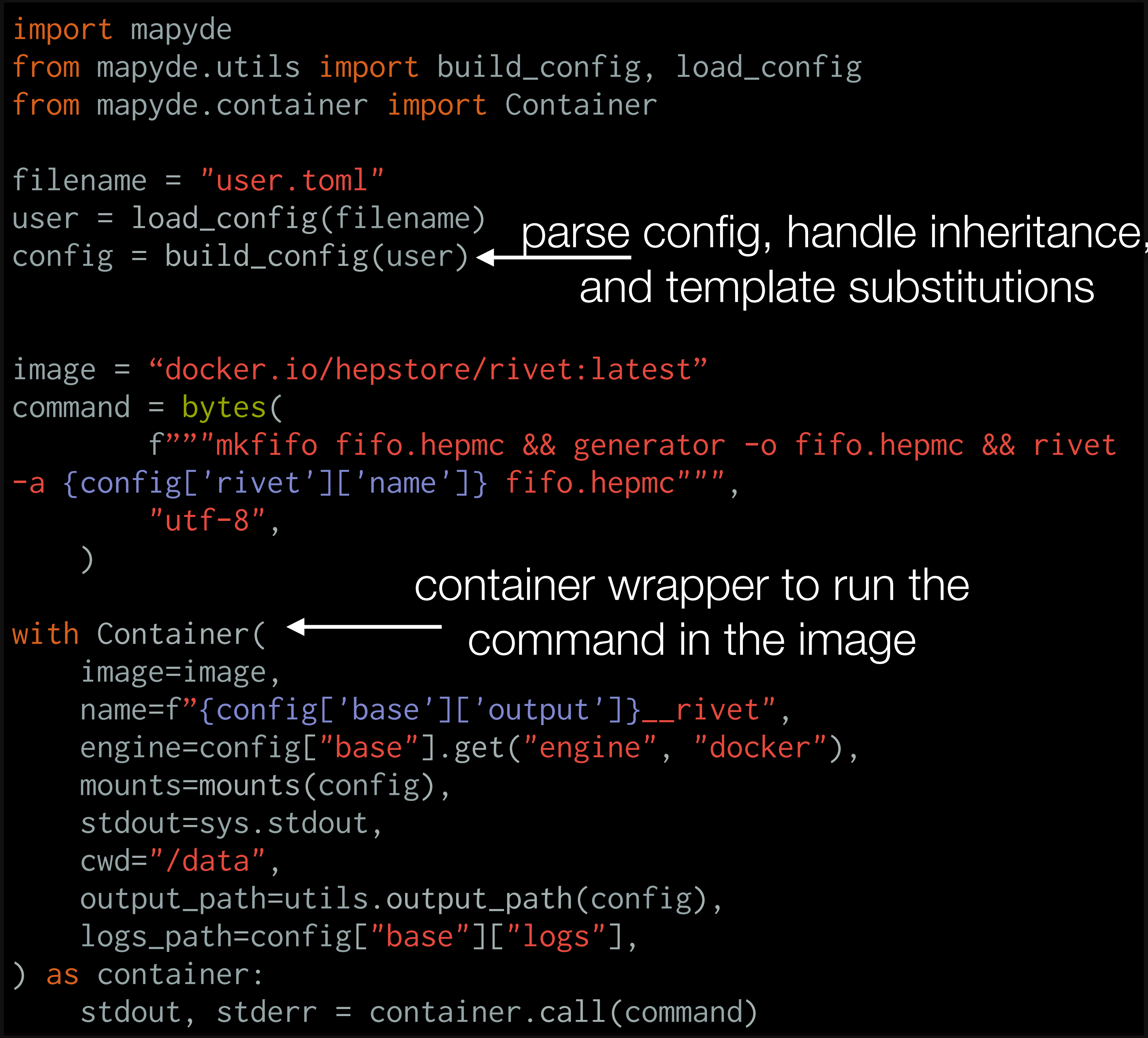
s”)

process card

values to substitute in param card

mapyde run user.toml

Custom usage: e.g. rivet



```
import mapyde
from mapyde.utils import build_config, load_config
from mapyde.container import Container

filename = "user.toml"
user = load_config(filename)
config = build_config(user)


image = “docker.io/hepstore/rivet:latest”
command = bytes(
        f””"mkfifo fifo.hepmc && generator -o fifo.hepmc && rivet
-a {config['rivet']['name']} fifo.hepmc""",
        "utf-8",
    )

with Container(
    image=image,
    name=f”{config['base']['output']}__rivet",
    engine=config["base"].get("engine", "docker"),
    mounts=mounts(config),
    stdout=sys.stdout,
    cwd="/data",
    output_path=utils.output_path(config),
    logs_path=config["base"]["logs"],
) as container:
    stdout, stderr = container.call(command)
```

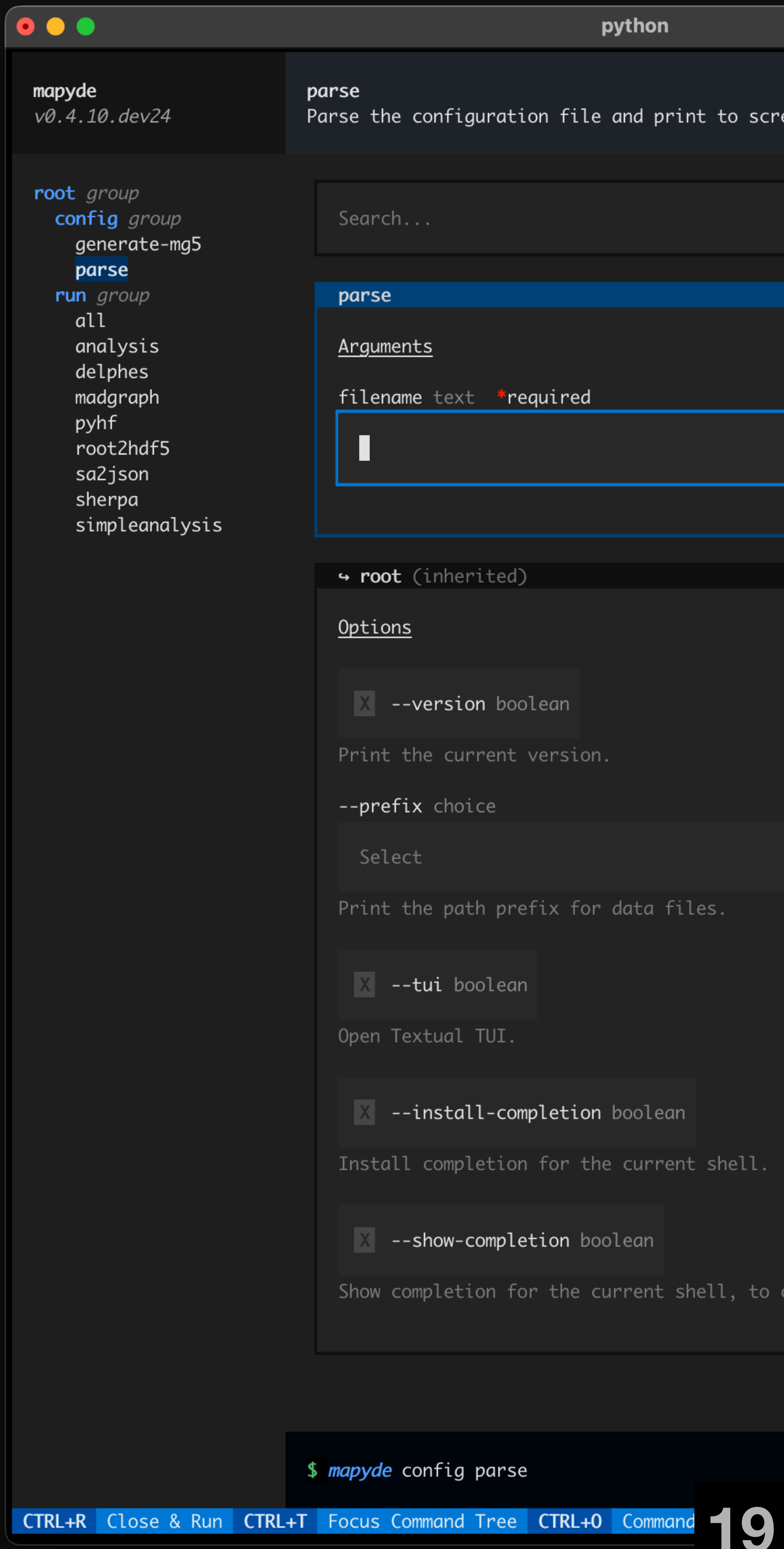
parse config, handle inheritance, and template substitutions

container wrapper to run the command in the image

- out of the box, mapyde comes with enough utilities to roll your own workflow
- maybe you want to use something that’s not currently provided

Other technical bits

- **mapyde utilities**: https://scipp-atlas.github.io/mapyde/latest/reference/mapyde/utils/
  - merging configs, rendering Jinja2 template strings, handling inheritance
- **mapyde prefix**: https://scipp-atlas.github.io/mapyde/latest/reference/mapyde/prefix/
  - “global” configuration paths for data, cards, scripts, templates
  - can be used in python within a context manager (to change paths temporarily)
- **mapyde runner**: https://scipp-atlas.github.io/mapyde/latest/reference/mapyde/runner/
  - hard-coded support for MadGraph, Pythia, Delphes, Sherpa, SimpleAnalysis, SimpleAnalysis->JSON, pyhf, ROOT->HDF5, and custom analysis scripts
  - can provide support for other tools upon request in GitHub issues
- **mapyde tui** (textual user interface) from command line



```
python

mapyde                  parse
v0.4.10.dev24           Parse the configuration file and print to scr

root group              Search...
  config group
    generate-mg5        parse
    parse
  run group             Arguments
    all
    analysis            filename text  *required
    delphes
    madgraph
    pyhf
    root2hdf5
    sa2json
    sherpa
    simpleanalysis
                        ↳ root (inherited)

                        Options

                        X --version boolean
                        Print the current version.

                        --prefix choice
                         Select
                        Print the path prefix for data files.

                        X --tui boolean
                        Open Textual TUI.

                        X --install-completion boolean
                        Install completion for the current shell.

                        X --show-completion boolean
                        Show completion for the current shell, to

                        $ mapyde config parse

CTRL+R  Close & Run  CTRL+T  Focus Command Tree  CTRL+O  Command
```

Summary

mapyde

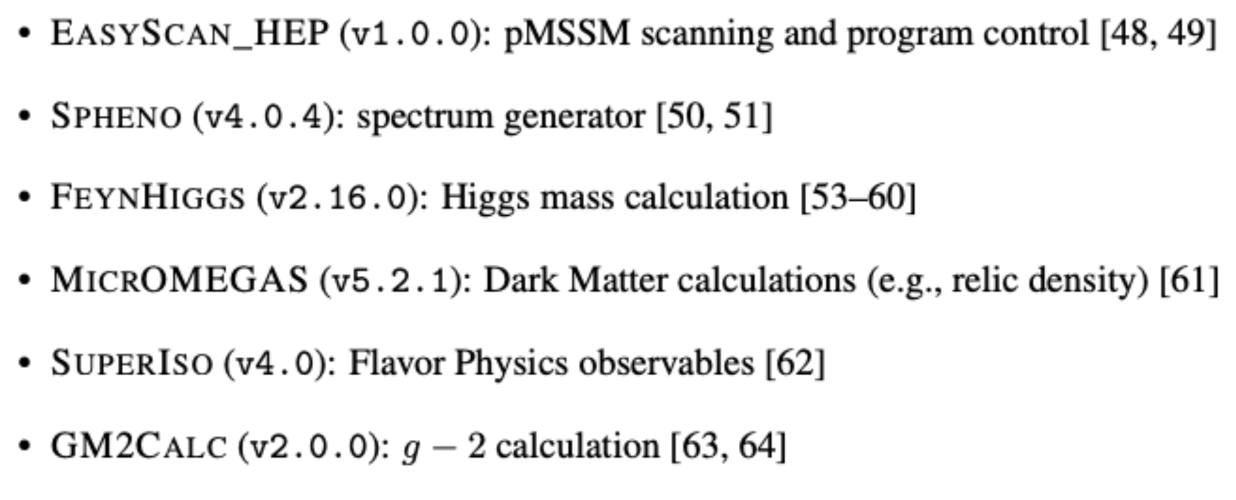
- A **reinterpretation** of compressed electroweak SUSY was performed using public ATLAS data and mapyde — **could guide Run 3 searches**
- Relies on many pieces in place from both experimental and theory communities, and from tech industry — **successful coordination**
- Low barrier to entry means mapyde can be used as a **pedagogic tool** for teaching new students particle physics through a hands-on approach
- **paper**: arXiv:2306.11055 (submitted to SciPost Phys)
  - **repo**: scipp-atlas/mapyde-paper
  - **data**: scipp-atlas/mapyde-paper-data

docs online | License Apache 2.0 | DOI 10.5281/zenodo.8040910

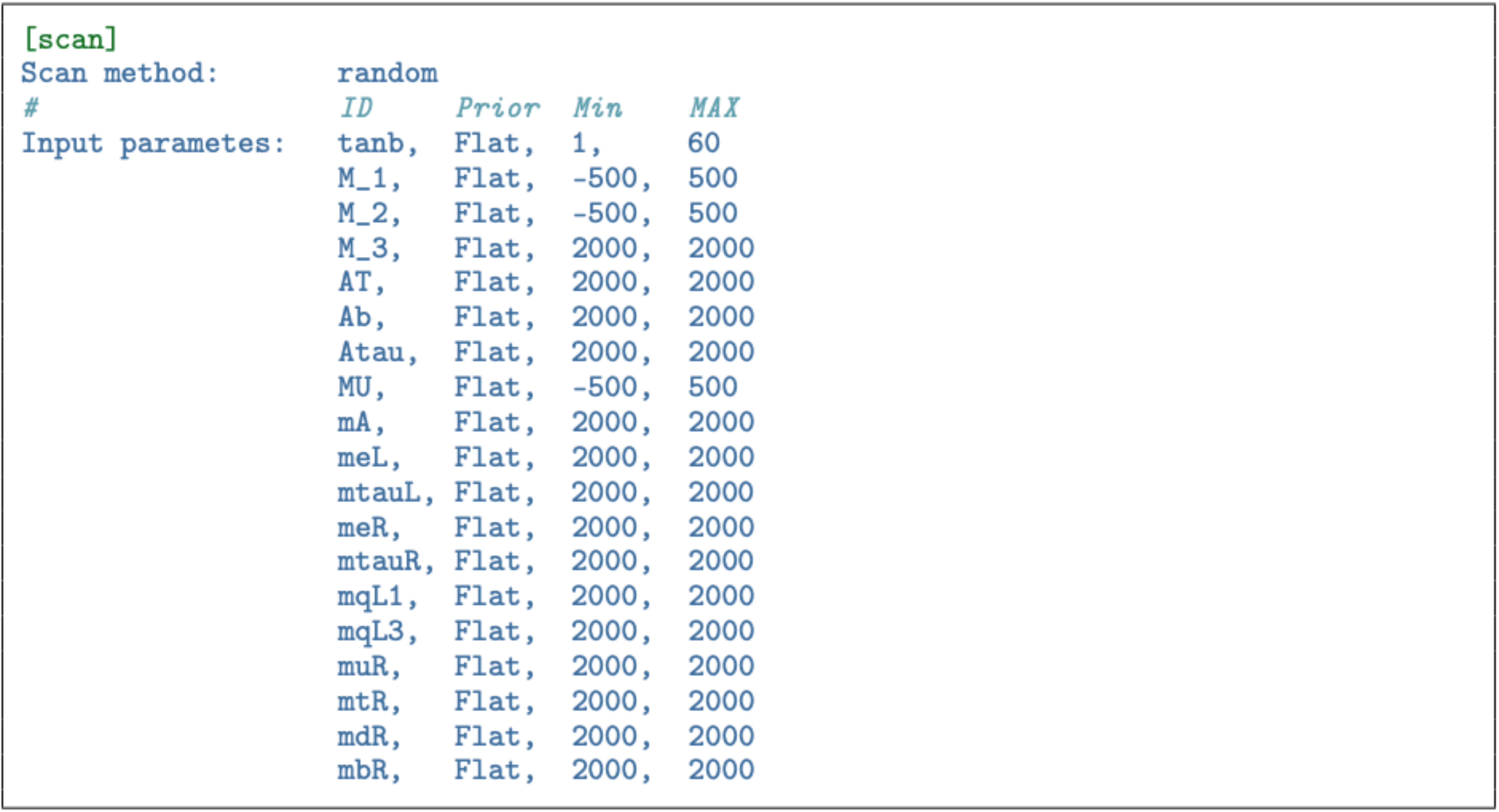
pypi package 0.5.0 | conda-forge v0.5.0

Backup

pMSSM Scan



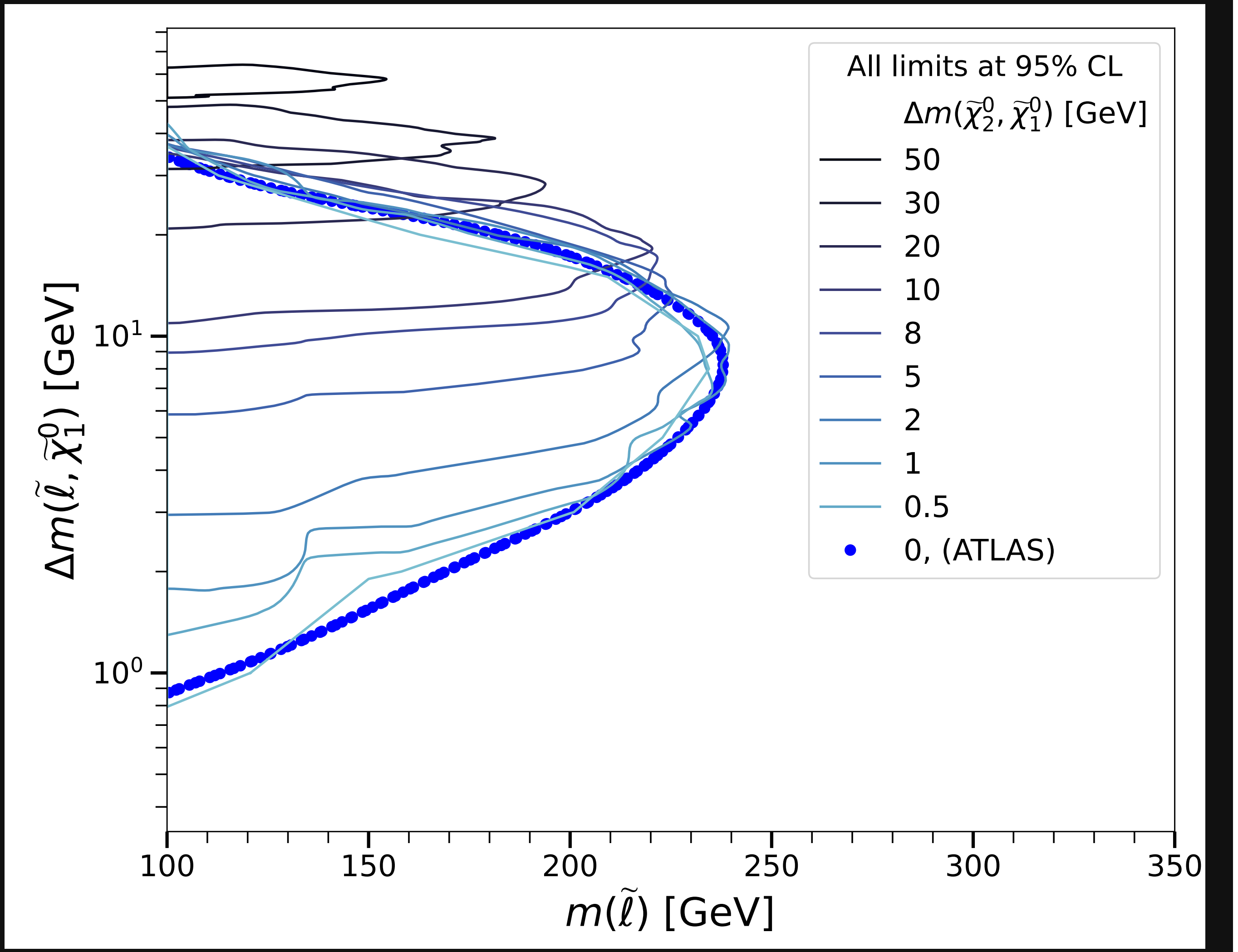
- EASYSCAN_HEP (v1.0.0): pMSSM scanning and program control [48, 49]
- SPHENO (v4.0.4): spectrum generator [50, 51]
- FEYNHIGGS (v2.16.0): Higgs mass calculation [53–60]
- MICROMEGAS (v5.2.1): Dark Matter calculations (e.g., relic density) [61]
- SUPERISO (v4.0): Flavor Physics observables [62]
- GM2CALC (v2.0.0): $g-2$ calculation [63, 64]



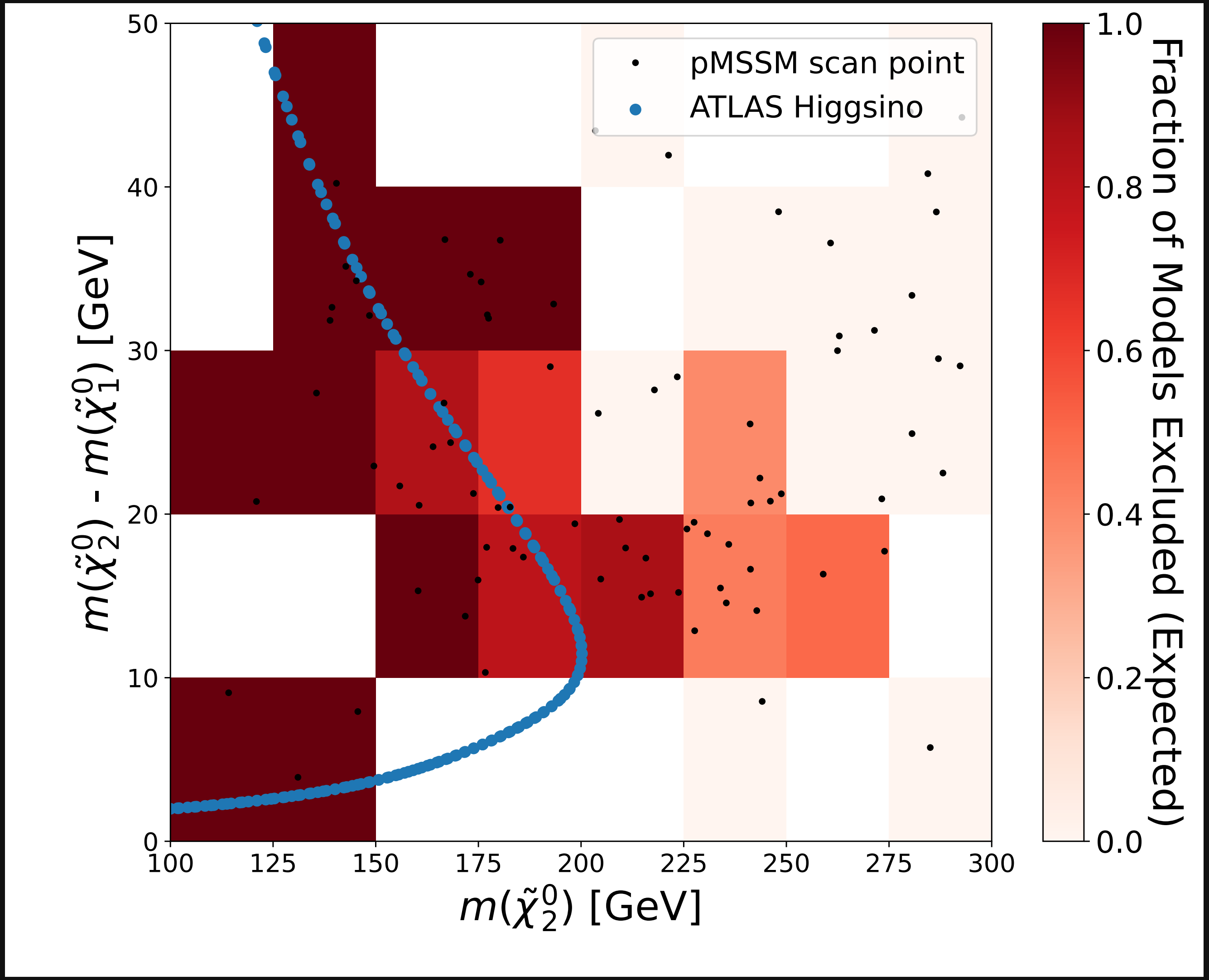
```
[scan]
Scan method:        random
#                   ID      Prior   Min     MAX
Input parametes:    tanb,   Flat,   1,      60
                    M_1,    Flat,   -500,   500
                    M_2,    Flat,   -500,   500
                    M_3,    Flat,   2000,   2000
                    AT,     Flat,   2000,   2000
                    Ab,     Flat,   2000,   2000
                    Atau,   Flat,   2000,   2000
                    MU,     Flat,   -500,   500
                    mA,     Flat,   2000,   2000
                    meL,    Flat,   2000,   2000
                    mtauL,  Flat,   2000,   2000
                    meR,    Flat,   2000,   2000
                    mtauR,  Flat,   2000,   2000
                    mqL1,   Flat,   2000,   2000
                    mqL3,   Flat,   2000,   2000
                    muR,    Flat,   2000,   2000
                    mtR,    Flat,   2000,   2000
                    mdR,    Flat,   2000,   2000
                    mbR,    Flat,   2000,   2000
```

Listing 1: A portion of the easyscan.ini configuration defining the random sampling for the electroweakinos scan.

Expected



sleptons



wino-bino-higgsino