Global fits beyond the standard Beyond-the-Standard-Model models

Anders Kvellestad, University of Oslo

on behalf of the GAMBIT Collaboration

Fysikermøtet 2021 — June 23, 2021

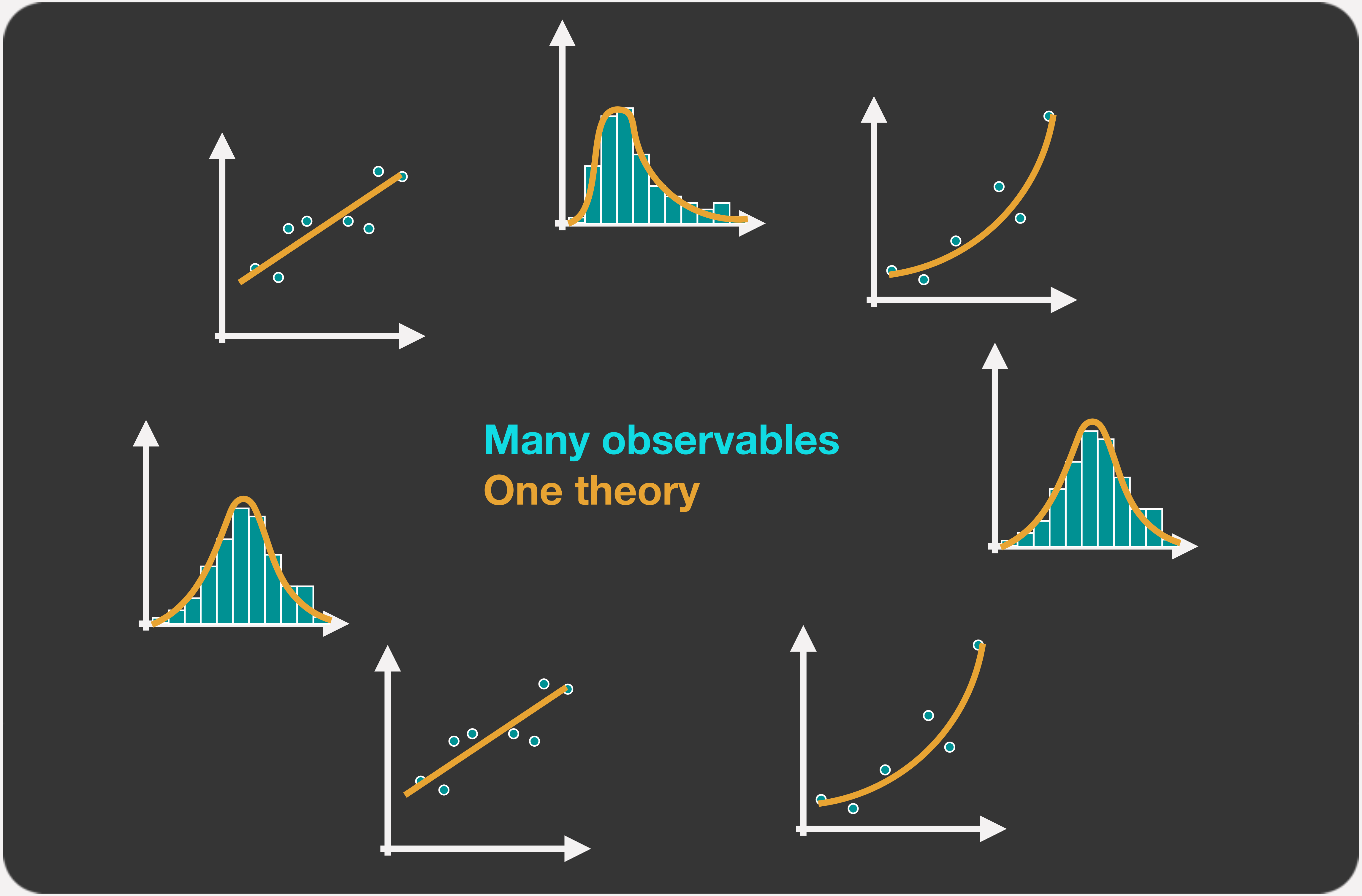
UiO : **University of Oslo**

Outline

1. **Global fits**
2. **GAMBIT**
3. **GUM**

1. Global fits

Global fits



Many observables

One theory

The basic steps of a BSM global fit

- Choose your **BSM theory and parameterisation**

- Construct the **joint likelihood function** including observables from collider physics, dark matter, flavor physics, +++

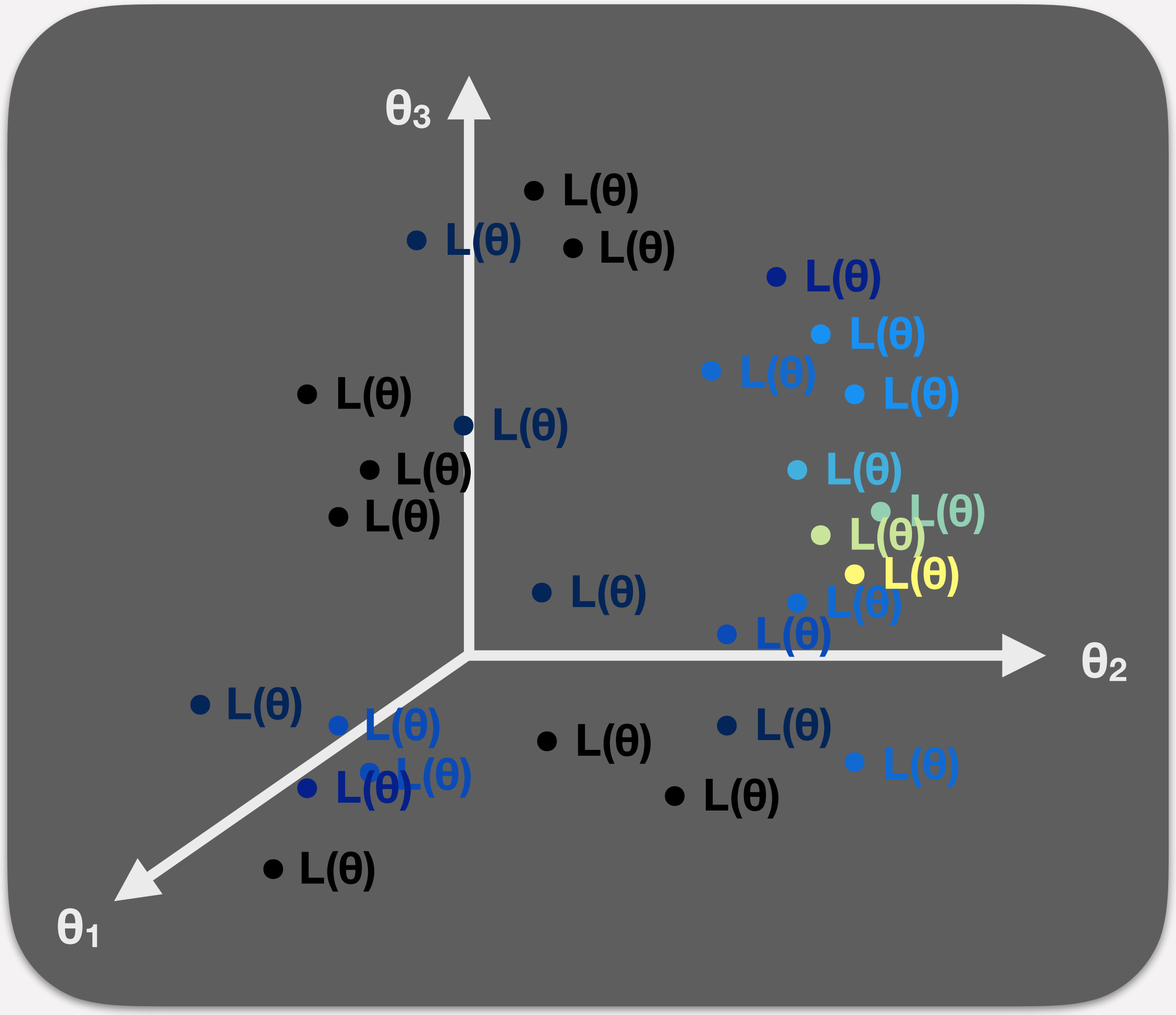
$$\mathcal{L} = \mathcal{L}_{\mathrm{collider}} \mathcal{L}_{\mathrm{DM}} \mathcal{L}_{\mathrm{flavor}} \mathcal{L}_{\mathrm{EWPO}} \cdots$$

- Use **sophisticated scanning techniques** to explore the likelihood function across the parameter space of the theory

- Test **parameter regions** properly — not just single points *(parameter estimation)*

- Test **different theories the same way** *(model comparison)*

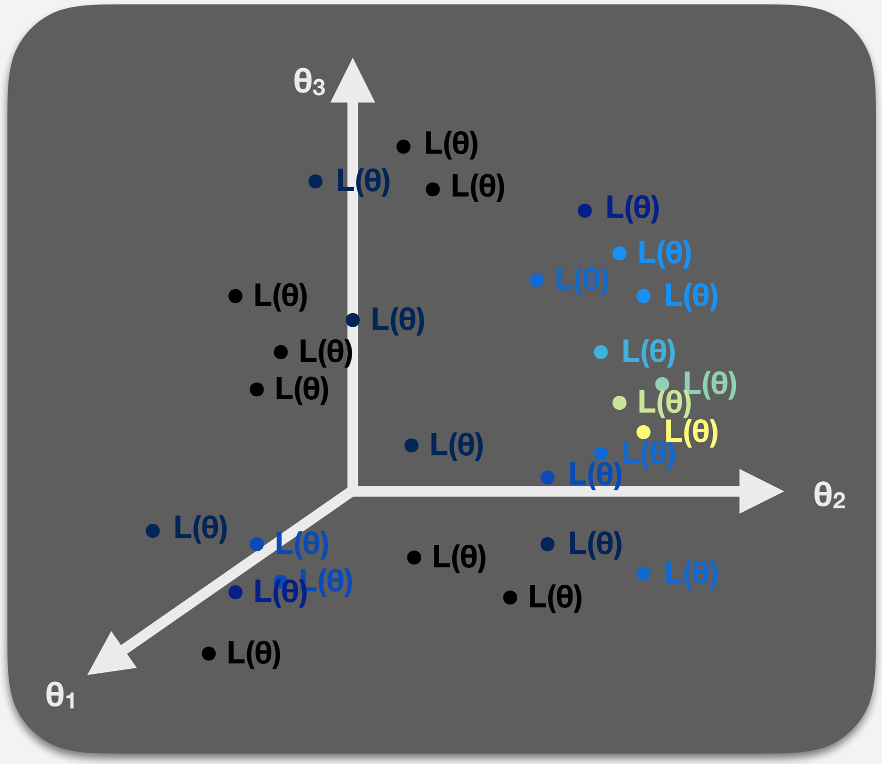
- **Explore the model parameter space** (θ_1, θ_2, θ_3, …)
- At every point θ: **calculate predictions(θ) → evaluate joint likelihood L(θ)**



- Region of highest L(θ) or lnL(θ): **model's best simultaneous fit to all data**
  (but not necessarily a *good* fit, or the most probable θ…)

Computational challenges:

- Need **smart exploration** of parameter space
- Need **fast theory calculations**
- Need **fast simulations of experiments** (e.g. LHC)
- Need **sufficiently detailed likelihoods**



JHEP

Published for SISSA by Springer

Received: September 15, 2017
Revised: February 1, 2018
Accepted: March 16, 2018
Published: March 27, 2018

Search for electroweak production of charginos and neutralinos in multilepton final states in proton-proton collisions at $\sqrt{s} = 13$ TeV

The CMS collaboration

E-mail: cms-publication-committee-chair@cern.ch

Abstract: Results are presented from a search for the direct electroweak production of charginos and neutralinos in signatures with either two or more leptons (electrons or muons) of the same electric charge, or with three or more leptons, which can include up to two hadronically decaying tau leptons. The results are based on a sample of proton-proton collision data collected at $\sqrt{s} = 13$ TeV, recorded with the CMS detector at the LHC, corresponding to an integrated luminosity of $35.9\,\mathrm{fb}^{-1}$. The observed event yields are consistent with the expectations based on the standard model. The results are interpreted in simplified models of supersymmetry describing various scenarios for the production and decay of charginos and neutralinos. Depending on the model parameters chosen, mass values between 180 GeV and 1150 GeV are excluded at 95% CL. These results significantly extend the parameter space probed for these particles in searches at the LHC. In addition, results are presented in a form suitable for alternative theoretical interpretations.

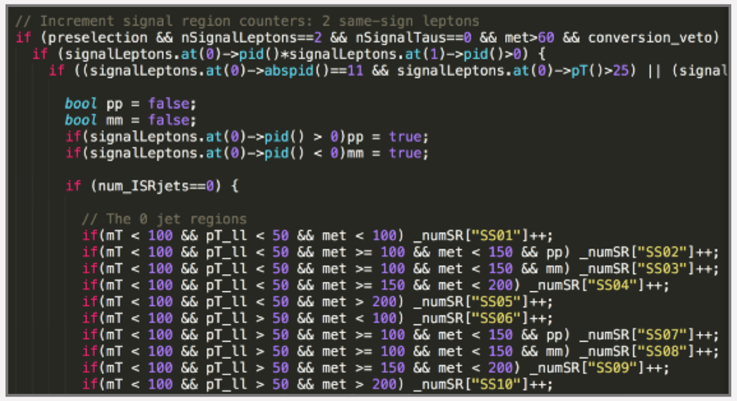
Keywords: Hadron-Hadron scattering (experiments), Supersymmetry

ArXiv ePrint: 1709.05406

Open Access, Copyright CERN,
for the benefit of the CMS Collaboration.
Article funded by SCOAP3.

https://doi.org/10.1007/JHEP03(2018)166

JHEP03(2018)166



```
// Increment signal region counters: 2 same-sign leptons
if (preselection && nSignalLeptons==2 && nSignalTaus==0 && met>60 && conversion_veto)
  if (signalLeptons.at(0)->pid()*signalLeptons.at(1)->pid()>0) {
    if ((signalLeptons.at(0)->abspid()==11 && signalLeptons.at(0)->pT()>25) || (signal

      bool pp = false;
      bool mm = false;
      if(signalLeptons.at(0)->pid() > 0)pp = true;
      if(signalLeptons.at(0)->pid() < 0)mm = true;

      if (num_ISRjets==0) {

        // The 0 jet regions
        if(mT < 100 && pT_ll < 50 && met < 100) _numSR["SS01"]++;
        if(mT < 100 && pT_ll < 50 && met >= 100 && met < 150 && pp) _numSR["SS02"]++;
        if(mT < 100 && pT_ll < 50 && met >= 100 && met < 150 && mm) _numSR["SS03"]++;
        if(mT < 100 && pT_ll < 50 && met >= 150 && met < 200) _numSR["SS04"]++;
        if(mT < 100 && pT_ll < 50 && met > 200) _numSR["SS05"]++;
        if(mT < 100 && pT_ll > 50 && met < 100) _numSR["SS06"]++;
        if(mT < 100 && pT_ll > 50 && met >= 100 && met < 150 && pp) _numSR["SS07"]++;
        if(mT < 100 && pT_ll > 50 && met >= 100 && met < 150 && mm) _numSR["SS08"]++;
        if(mT < 100 && pT_ll > 50 && met >= 150 && met < 200) _numSR["SS09"]++;
        if(mT < 100 && pT_ll > 50 && met > 200) _numSR["SS10"]++;
```

Some infrastructure challenges:

- Need **different parameter scanning algorithms**
- Need **model-agnostic core framework**
- Need to interface ***many*** **external physics codes**
- Need **massive parallelisation**…
- …which implies a need for **diskless interfacing**…
- …which implies a need to **stop external codes from calling STOP and kill your 10,000-CPU scan**… :)

2. GAMBIT

GAMBIT: The Global And Modular BSM Inference Tool

gambit.hepforge.org EPJC **77** (2017) 784 arXiv:1705.07908

- Extensive model database – not just SUSY
- Extensive observable/data libraries
- Many statistical and scanning options (Bayesian & frequentist)
- *Fast* LHC likelihood calculator
- Massively parallel
- Fully open-source

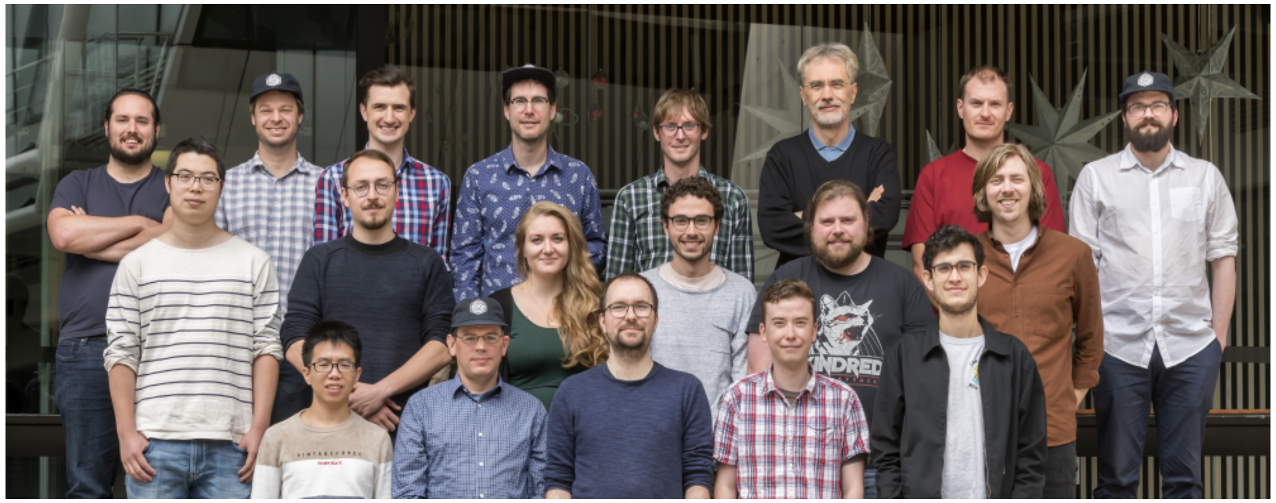
- Fast definition of new datasets and theories
- Plug and play scanning, physics and likelihood packages

Members of:

ATLAS, Belle-II, CLiC, CMS, CTA, *Fermi*-LAT, DARWIN, IceCube, LHCb, SHiP, XENON

Authors of:

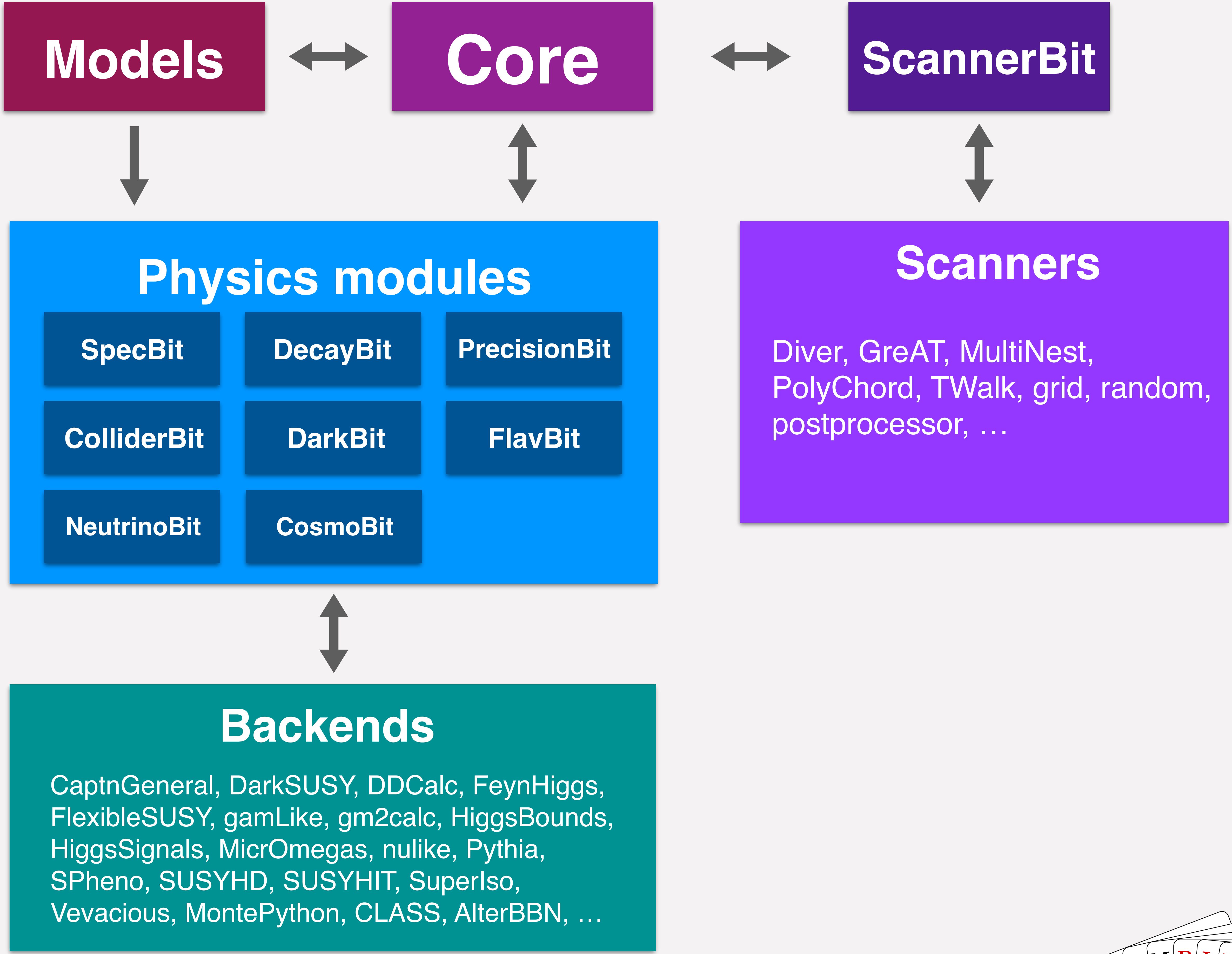
DarkSUSY, DDCalc, Diver, FlexibleSUSY, gamlike, GM2Calc, IsaTools, nulike, PolyChord, Rivet, SoftSUSY, SuperISO, SUSY-AI, WIMPSim



Recent collaborators:

F Agocs, V Ananyev, P Athron, C Balázs, A Beniwal, J Bhom, S Bloor, T Bringmann, A Buckley, J-E Camargo-Molina, C Chang, M Chrzaszcz, J Conrad, J Cornell, M Danninger, J Edsjö, B Farmer, A Fowlie, T Gonzalo, P Grace, W Handley, J Harz, S Hoof, S Hotinli, F Kahlhoefer, N Avis Kozar, A Kvellestad, P Jackson, A Ladhu, N Mahmoudi, G Martinez, MT Prim, F Rajec, A Raklev, J Renk, C Rogan, R Ruiz, I Sáez Casares, N Serra, A Scaffidi, P Scott, P Stöcker, W Su, J Van den Abeele, A Vincent, C Weniger, M White, Y Zhang

70+ participants in 11 experiments and 14 major theory codes



Models

Core

ScannerBit

Physics modules

SpecBit

DecayBit

PrecisionBit

ColliderBit

DarkBit

FlavBit

NeutrinoBit

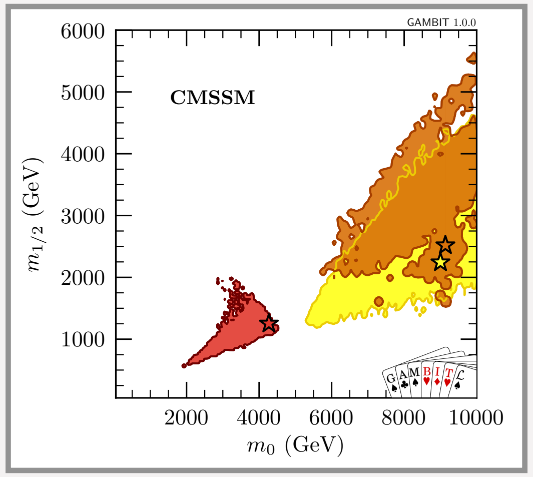
CosmoBit

Scanners

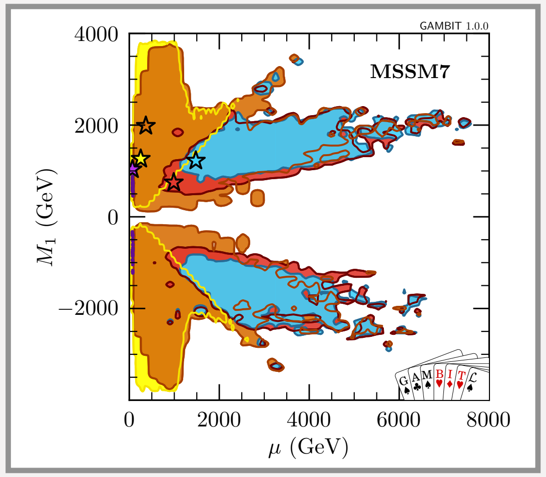
Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

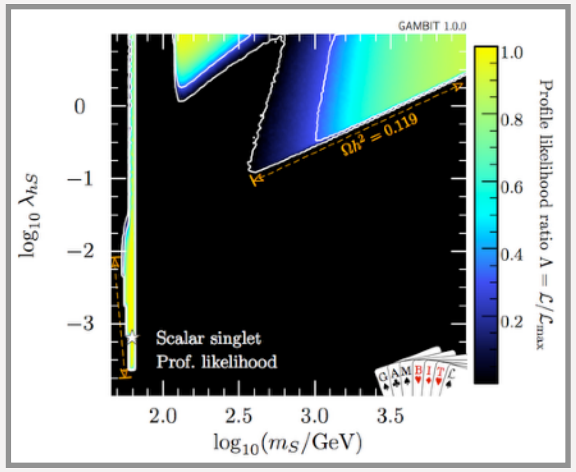
CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …



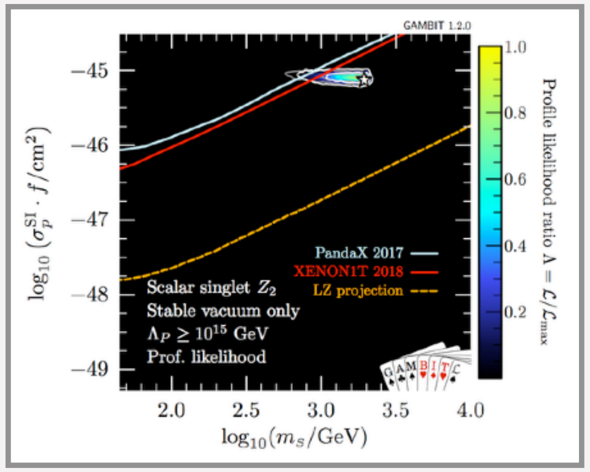
GUT-scale SUSY: 1705.07935



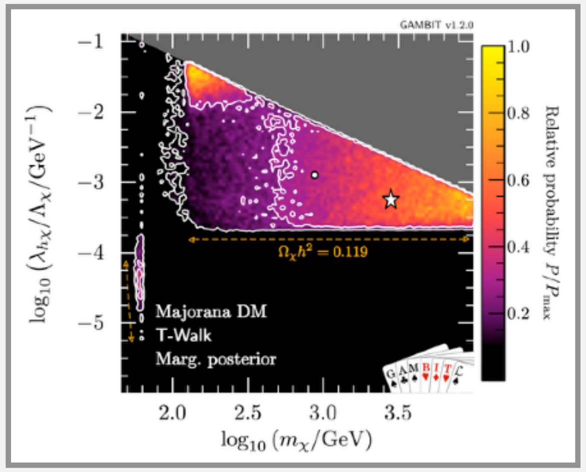
MSSM7: 1705.07917



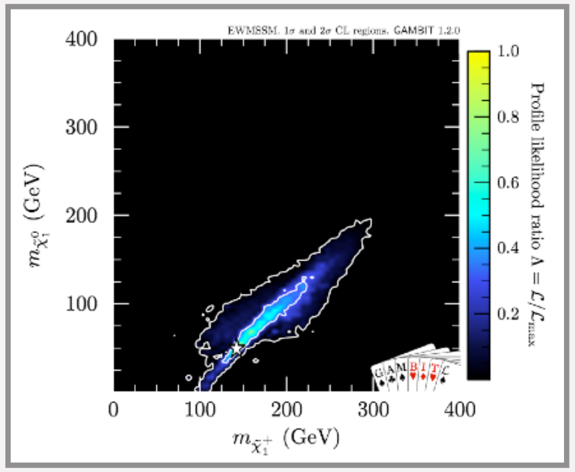
Scalar Higgs portal DM: 1705.07931



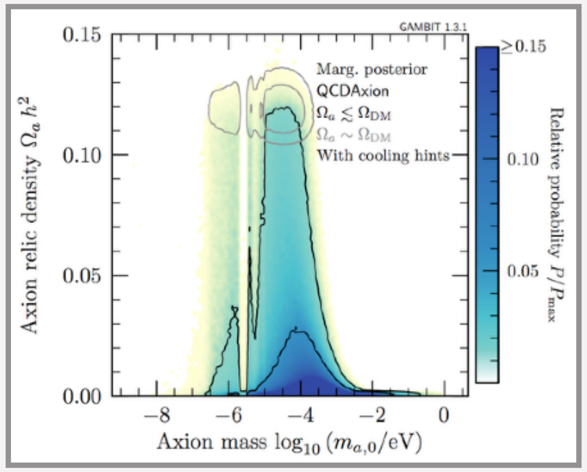
Scalar Higgs portal DM w/ vac. stability: 1806.11281



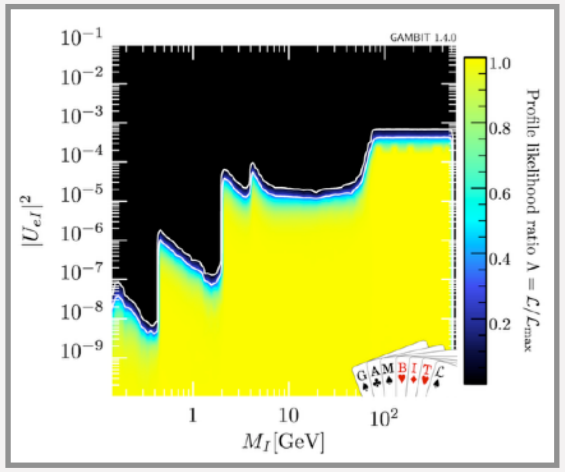
Vector and fermion Higgs portal DM: 1808.10465



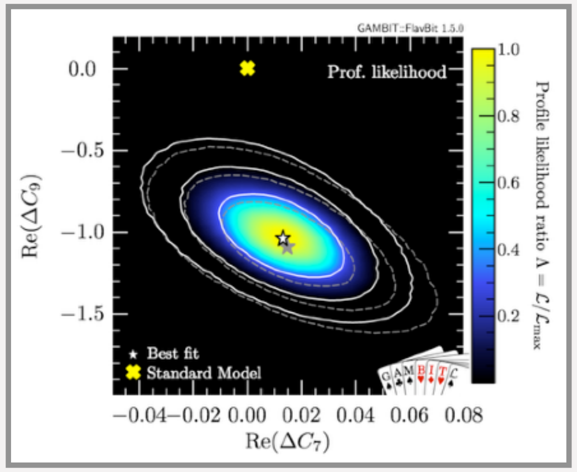
EW-MSSM: 1809.02097



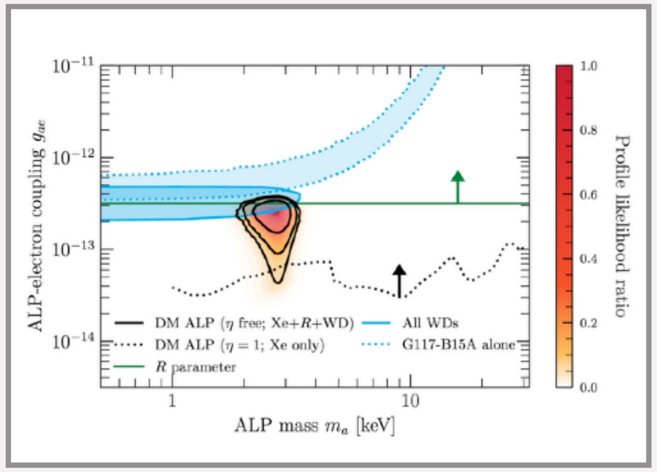
Axion-like particles: 1810.07192



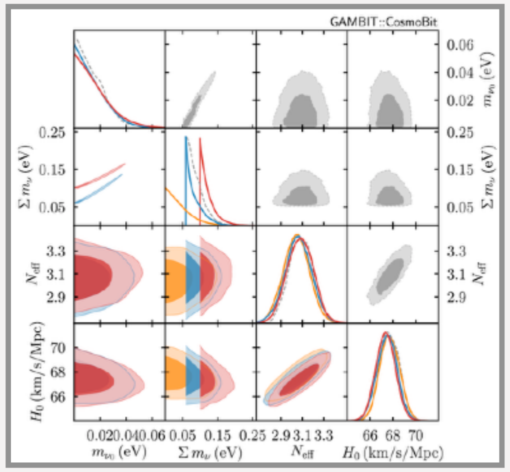
Right-handed neutrinos: 1908.02302



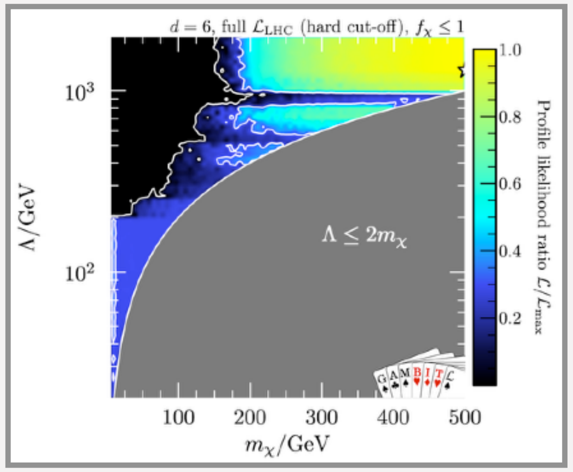
Flavour EFT: 2006.03489



More axion-like particles: 2006.03489



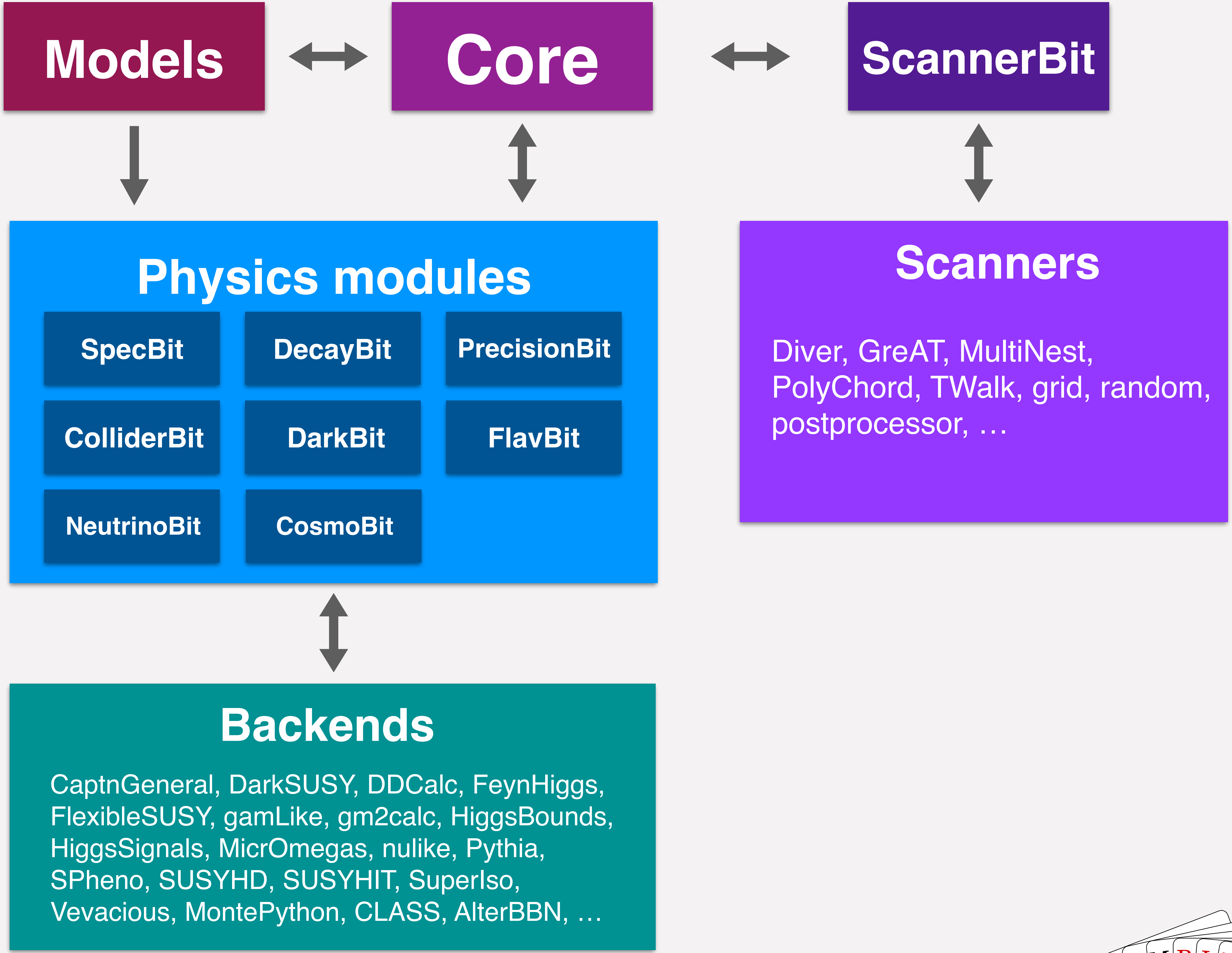
Neutrinos and cosmo: 2009.03287



Dark matter EFTs: 2106.02056

GAMBIT clearly works as a **general framework for global fits**…

…but **how much work** does it take to set up GAMBIT to study a new model?



Models

Core

ScannerBit

Physics modules

SpecBit

DecayBit

PrecisionBit

ColliderBit

DarkBit

FlavBit

NeutrinoBit

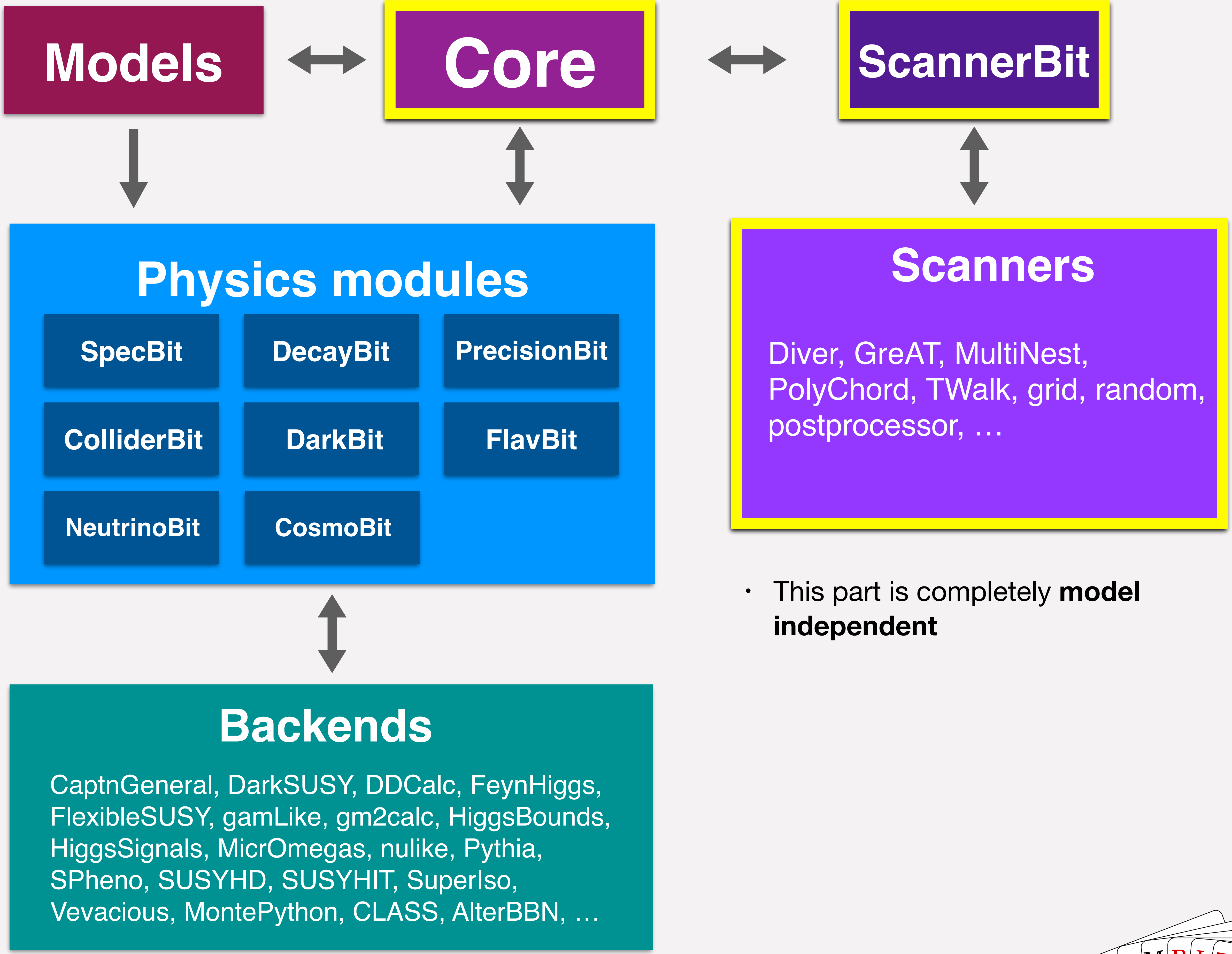
CosmoBit

Scanners

Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …



Models

Core

ScannerBit

Physics modules

SpecBit

DecayBit

PrecisionBit

ColliderBit

DarkBit

FlavBit

NeutrinoBit

CosmoBit

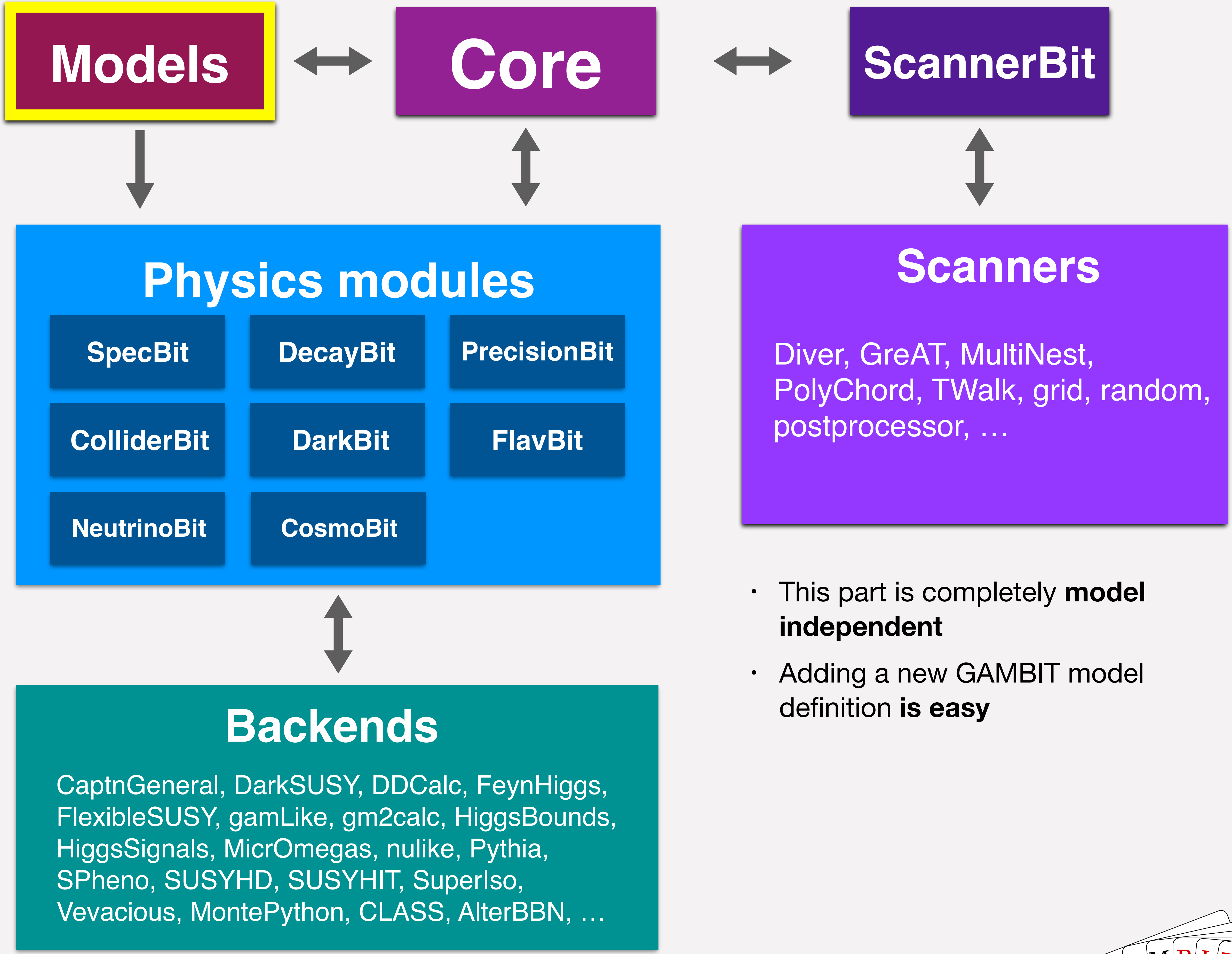
Scanners

Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …

- This part is completely **model independent**



Models

Core

ScannerBit

Physics modules

SpecBit, DecayBit, PrecisionBit, ColliderBit, DarkBit, FlavBit, NeutrinoBit, CosmoBit

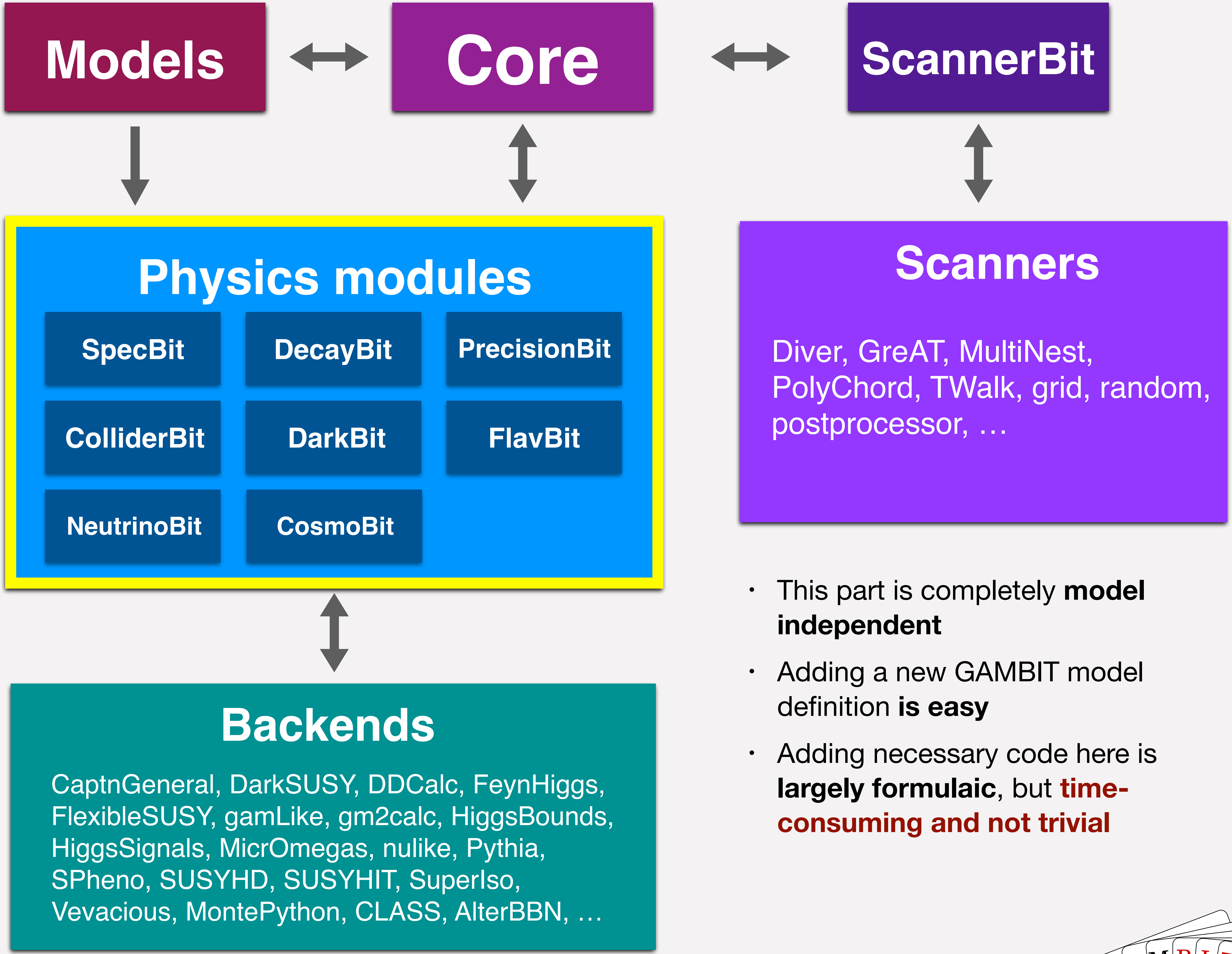
Scanners

Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …

- This part is completely **model independent**
- Adding a new GAMBIT model definition **is easy**



Models

Core

ScannerBit

Physics modules

SpecBit

DecayBit

PrecisionBit

ColliderBit

DarkBit

FlavBit

NeutrinoBit

CosmoBit

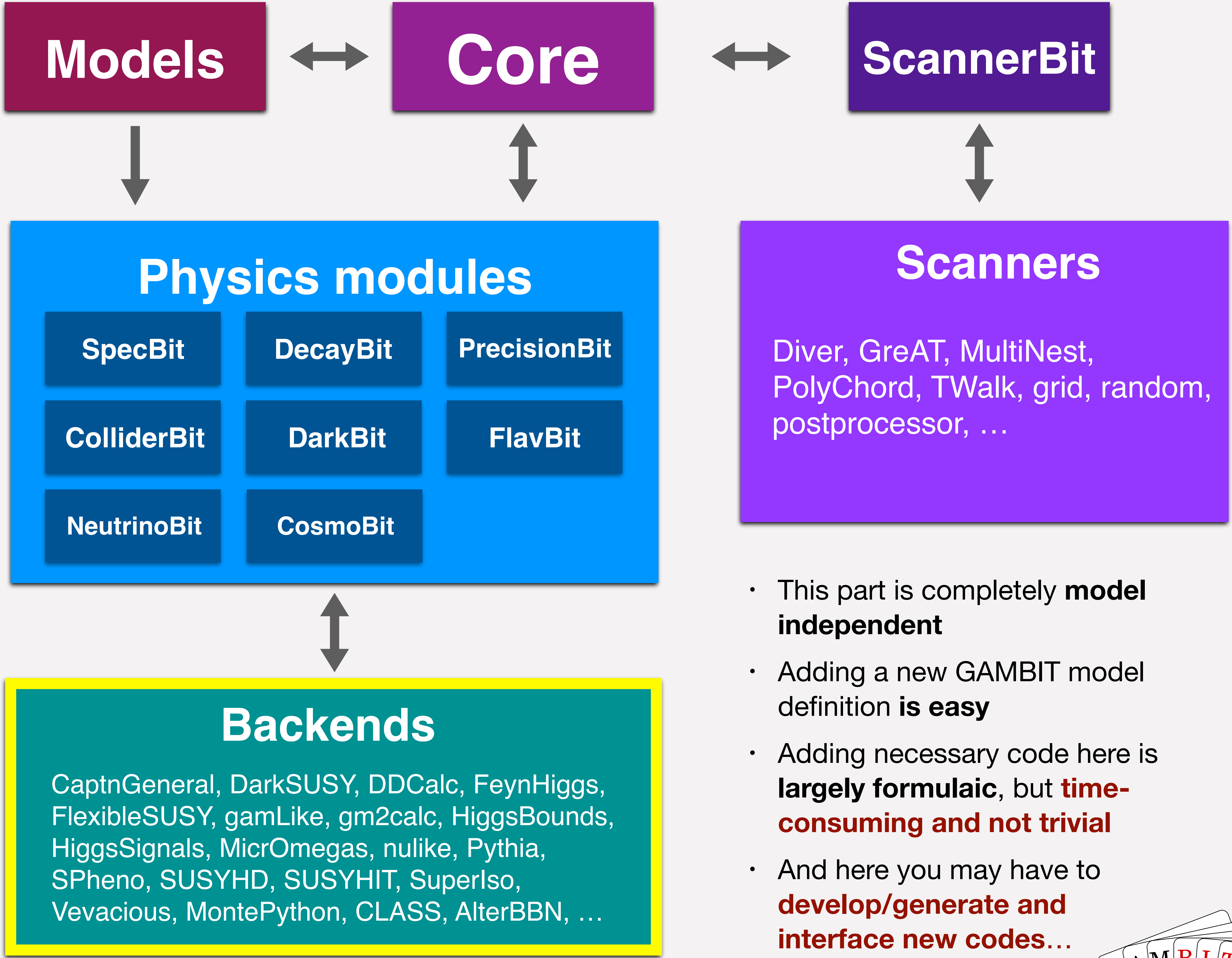
Scanners

Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …

- This part is completely **model independent**
- Adding a new GAMBIT model definition **is easy**
- Adding necessary code here is **largely formulaic**, but **time-consuming and not trivial**



Models

Core

ScannerBit

Physics modules

SpecBit

DecayBit

PrecisionBit

ColliderBit

DarkBit

FlavBit

NeutrinoBit

CosmoBit

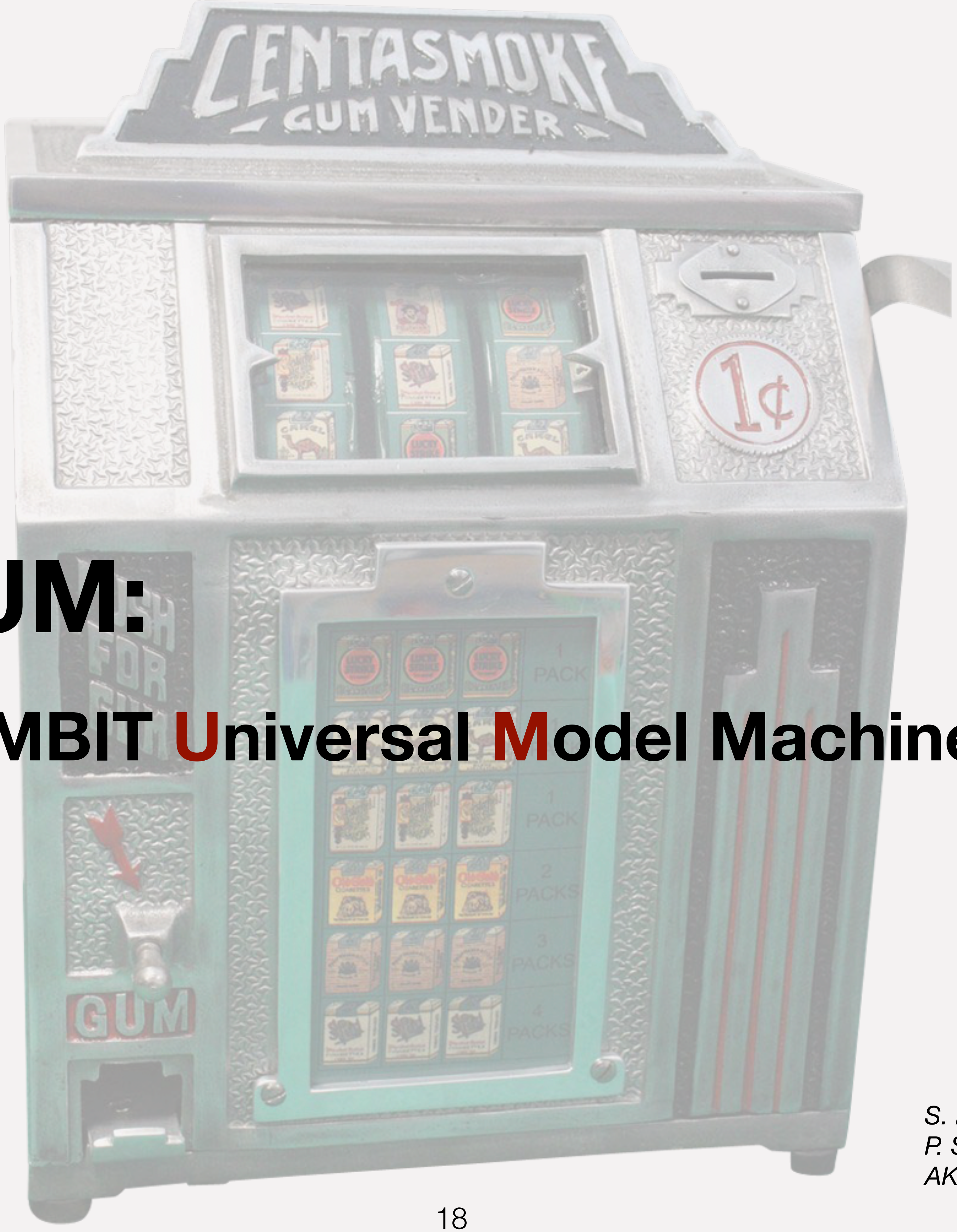
Scanners

Diver, GreAT, MultiNest, PolyChord, TWalk, grid, random, postprocessor, …

Backends

CaptnGeneral, DarkSUSY, DDCalc, FeynHiggs, FlexibleSUSY, gamLike, gm2calc, HiggsBounds, HiggsSignals, MicrOmegas, nulike, Pythia, SPheno, SUSYHD, SUSYHIT, SuperIso, Vevacious, MontePython, CLASS, AlterBBN, …

- This part is completely **model independent**
- Adding a new GAMBIT model definition **is easy**
- Adding necessary code here is **largely formulaic**, but **time-consuming and not trivial**
- And here you may have to **develop/generate and interface new codes…**

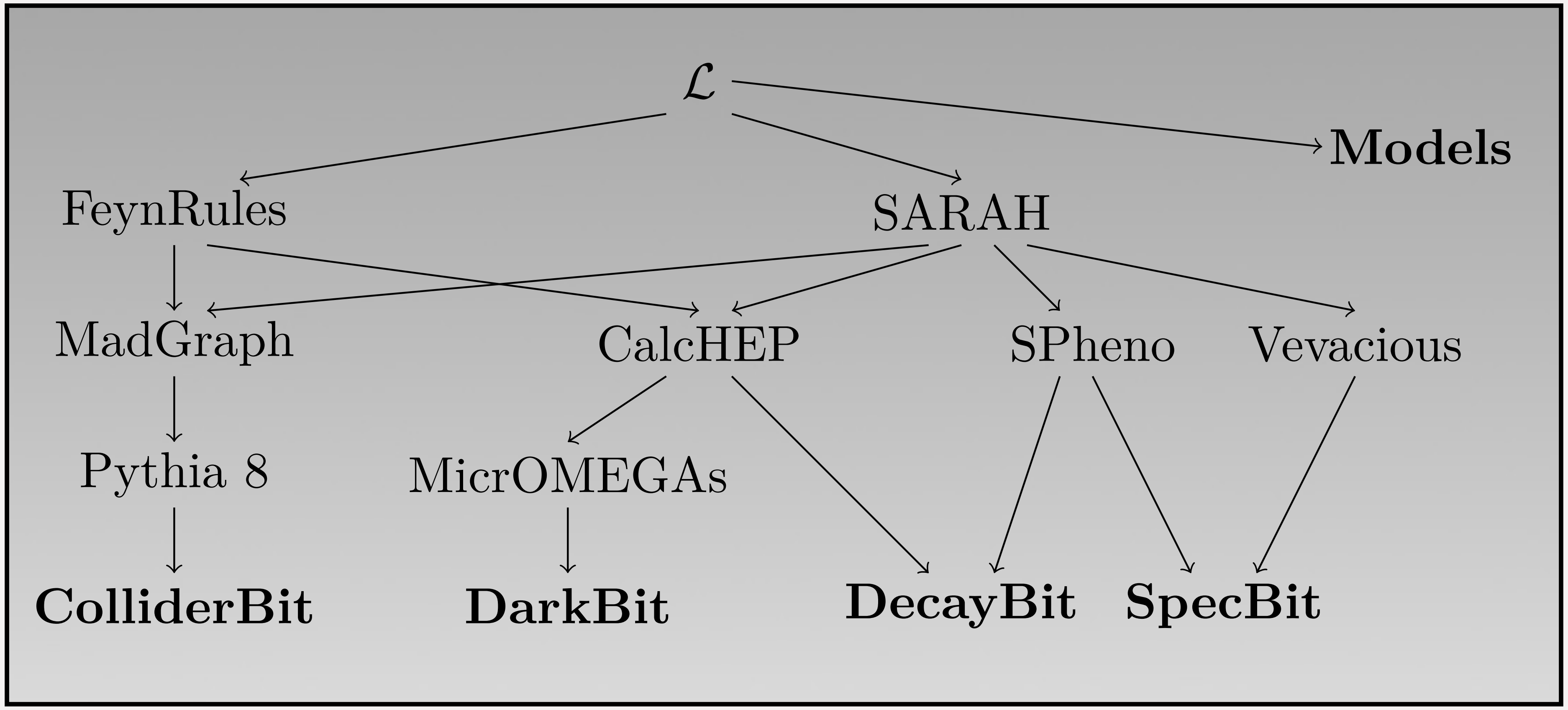


3. GUM:

the GAMBIT Universal Model Machine

S. Bloor, T. Gonzalo,
P. Scott, C. Chang,
AK, et al

GUM: the GAMBIT Universal Model Machine



- From **Lagrangian** to a **GAMBIT global fit**
- The major addition in **GAMBIT 2.0**
- **Runs existing BSM tool chains** to generate model-specific physics libraries
- **Generates interfaces** for these libraries to the relevant Bits in GAMBIT
- **Generates additional GAMBIT code** (model definiton, particle database additions, …)

GUM: the GAMBIT Universal Model Machine

Generated GAMBIT backends	FeynRules	SARAH	Usage in GAMBIT
CalcHEP	✓	✓	Decays, cross-sections
micrOMEGAs (via CalcHEP)	✓	✓	DM observables
Pythia (via MadGraph)	✓	✓	Collider physics
SPheno	✗	✓	Particle mass spectra, decay widths
Vevacious	✗	✓	Vacuum stability

From **FeynRules**

- Any Lagrangian (including EFTs), work at tree level
- CalcHEP
- micrOMEGAS (via CalcHEP)
- Pythia (via MadGraph)

From **SARAH**

- Renormalizable theories, one-loop
- CalcHEP
- micrOMEGAS (via CalcHEP)
- Pythia (via MadGraph)
- SPheno
- Vevacious
- + input for existing HiggsBounds + HiggsSignals backends (via SARAH-SPheno)

Model defined in a FeynRules/SARAH file

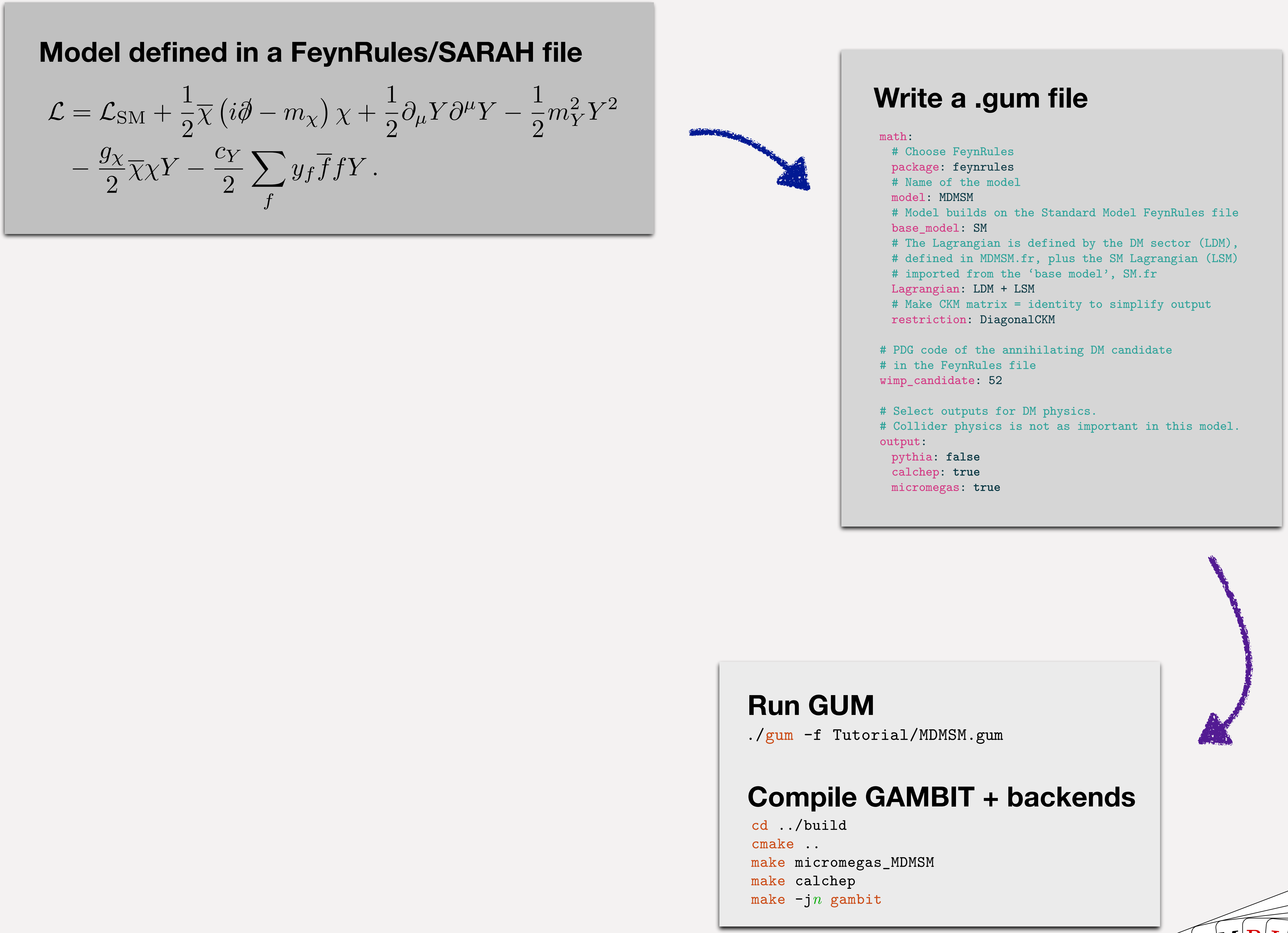
$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \frac{1}{2}\overline{\chi}\left(i\partial\!\!\!/ - m_\chi\right)\chi + \frac{1}{2}\partial_\mu Y \partial^\mu Y - \frac{1}{2}m_Y^2 Y^2 \\ - \frac{g_\chi}{2}\overline{\chi}\chi Y - \frac{c_Y}{2}\sum_f y_f \overline{f} f Y\,.\end{aligned}$$

Model defined in a FeynRules/SARAH file

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \frac{1}{2}\overline{\chi}\left(i\partial\!\!\!/ - m_\chi\right)\chi + \frac{1}{2}\partial_\mu Y \partial^\mu Y - \frac{1}{2}m_Y^2 Y^2 - \frac{g_\chi}{2}\overline{\chi}\chi Y - \frac{c_Y}{2}\sum_f y_f \overline{f} f Y\,.$$

Write a .gum file

```
math:
  # Choose FeynRules
  package: feynrules
  # Name of the model
  model: MDMSM
  # Model builds on the Standard Model FeynRules file
  base_model: SM
  # The Lagrangian is defined by the DM sector (LDM),
  # defined in MDMSM.fr, plus the SM Lagrangian (LSM)
  # imported from the ‘base model’, SM.fr
  Lagrangian: LDM + LSM
  # Make CKM matrix = identity to simplify output
  restriction: DiagonalCKM

# PDG code of the annihilating DM candidate
# in the FeynRules file
wimp_candidate: 52

# Select outputs for DM physics.
# Collider physics is not as important in this model.
output:
  pythia: false
  calchep: true
  micromegas: true
```



Model defined in a FeynRules/SARAH file

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \frac{1}{2}\overline{\chi}\left(i\partial\!\!\!/ - m_\chi\right)\chi + \frac{1}{2}\partial_\mu Y \partial^\mu Y - \frac{1}{2}m_Y^2 Y^2 - \frac{g_\chi}{2}\overline{\chi}\chi Y - \frac{c_Y}{2}\sum_f y_f \overline{f} f Y\,.$$

Write a .gum file

```
math:
  # Choose FeynRules
  package: feynrules
  # Name of the model
  model: MDMSM
  # Model builds on the Standard Model FeynRules file
  base_model: SM
  # The Lagrangian is defined by the DM sector (LDM),
  # defined in MDMSM.fr, plus the SM Lagrangian (LSM)
  # imported from the 'base model', SM.fr
  Lagrangian: LDM + LSM
  # Make CKM matrix = identity to simplify output
  restriction: DiagonalCKM

# PDG code of the annihilating DM candidate
# in the FeynRules file
wimp_candidate: 52

# Select outputs for DM physics.
# Collider physics is not as important in this model.
output:
  pythia: false
  calchep: true
  micromegas: true
```

Run GUM

```
./gum -f Tutorial/MDMSM.gum
```

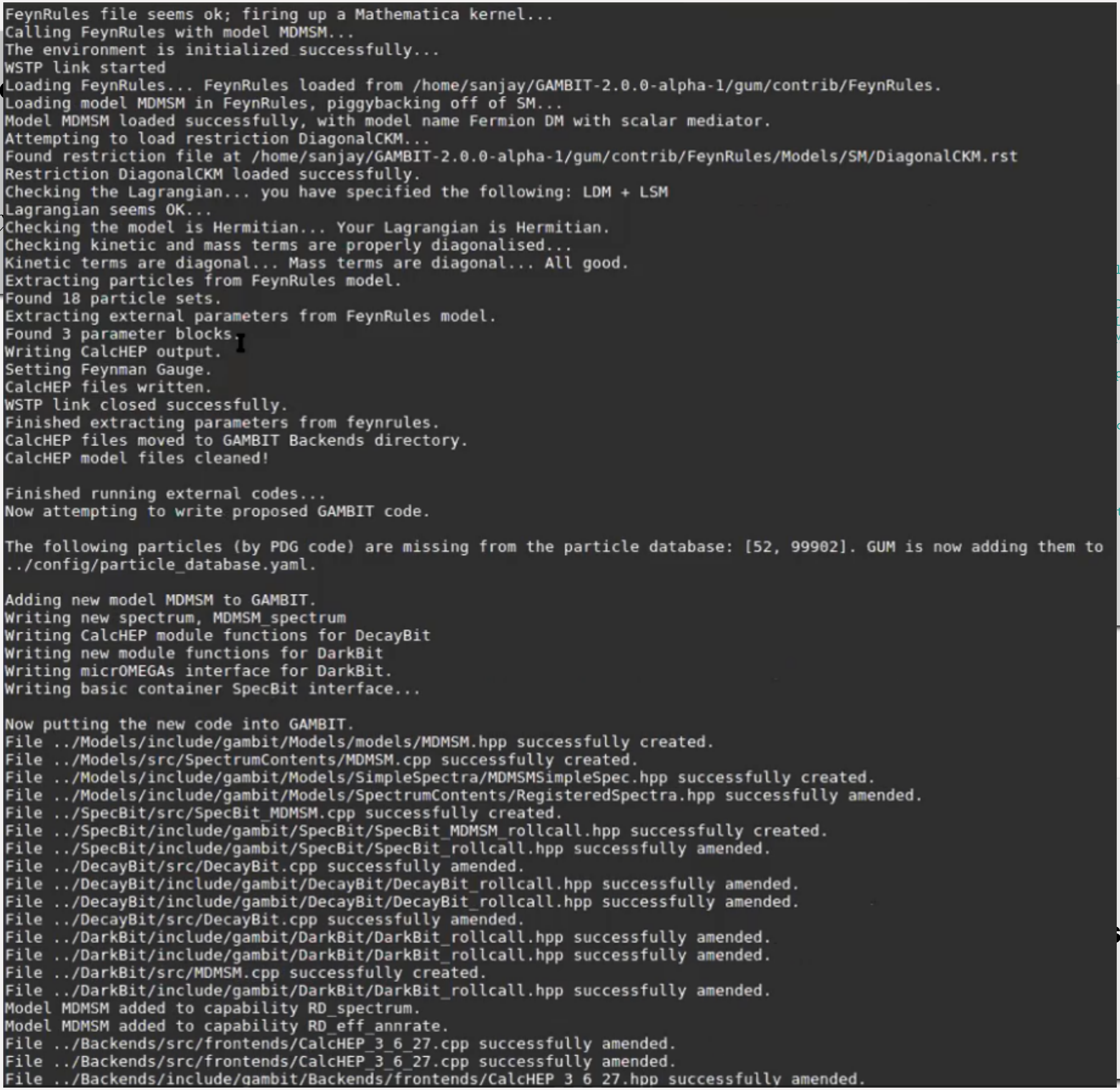
Compile GAMBIT + backends

```
cd ../build
cmake ..
make micromegas_MDMSM
make calchep
make -jn gambit
```

Model d

$$\mathcal{L} = \mathcal{L}_{\rm SM}$$

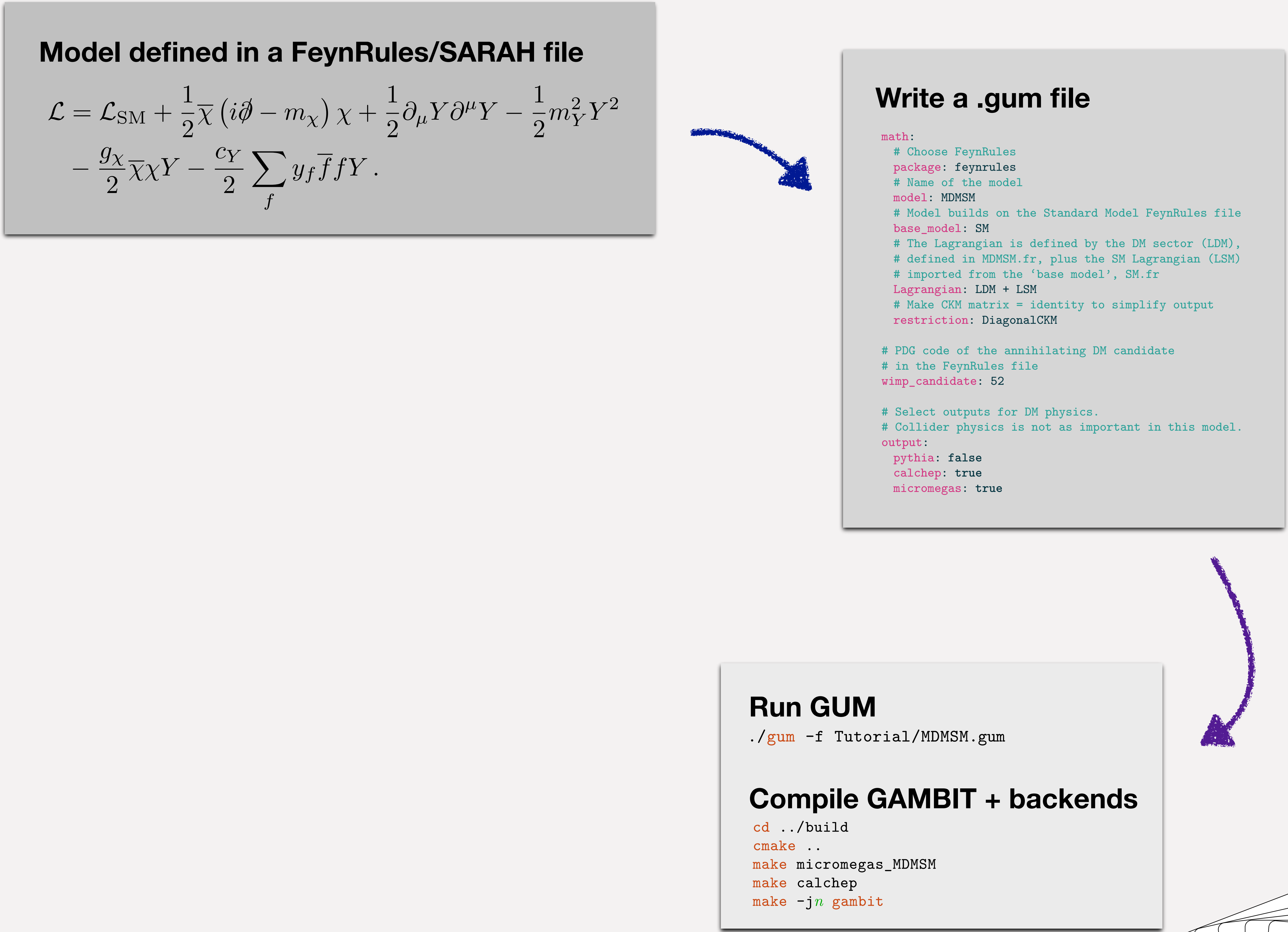
$$- \frac{g_\chi}{2} \bar{\chi}\chi$$



```
FeynRules file seems ok; firing up a Mathematica kernel...
Calling FeynRules with model MDMSM...
The environment is initialized successfully...
WSTP link started
Loading FeynRules... FeynRules loaded from /home/sanjay/GAMBIT-2.0.0-alpha-1/gum/contrib/FeynRules.
Loading model MDMSM in FeynRules, piggybacking off of SM...
Model MDMSM loaded successfully, with model name Fermion DM with scalar mediator.
Attempting to load restriction DiagonalCKM...
Found restriction file at /home/sanjay/GAMBIT-2.0.0-alpha-1/gum/contrib/FeynRules/Models/SM/DiagonalCKM.rst
Restriction DiagonalCKM loaded successfully.
Checking the Lagrangian... you have specified the following: LDM + LSM
Lagrangian seems OK...
Checking the model is Hermitian... Your Lagrangian is Hermitian.
Checking kinetic and mass terms are properly diagonalised...
Kinetic terms are diagonal... Mass terms are diagonal... All good.
Extracting particles from FeynRules model.
Found 18 particle sets.
Extracting external parameters from FeynRules model.
Found 3 parameter blocks.
Writing CalcHEP output.
Setting Feynman Gauge.
CalcHEP files written.
WSTP link closed successfully.
Finished extracting parameters from feynrules.
CalcHEP files moved to GAMBIT Backends directory.
CalcHEP model files cleaned!

Finished running external codes...
Now attempting to write proposed GAMBIT code.

The following particles (by PDG code) are missing from the particle database: [52, 99902]. GUM is now adding them to
../config/particle_database.yaml.

Adding new model MDMSM to GAMBIT.
Writing new spectrum, MDMSM_spectrum
Writing CalcHEP module functions for DecayBit
Writing new module functions for DarkBit
Writing micrOMEGAs interface for DarkBit.
Writing basic container SpecBit interface...

Now putting the new code into GAMBIT.
File ../Models/include/gambit/Models/models/MDMSM.hpp successfully created.
File ../Models/src/SpectrumContents/MDMSM.cpp successfully created.
File ../Models/include/gambit/Models/SimpleSpectra/MDMSMSimpleSpec.hpp successfully created.
File ../Models/include/gambit/Models/SpectrumContents/RegisteredSpectra.hpp successfully amended.
File ../SpecBit/src/SpecBit_MDMSM.cpp successfully created.
File ../SpecBit/include/gambit/SpecBit/SpecBit_MDMSM_rollcall.hpp successfully created.
File ../SpecBit/include/gambit/SpecBit/SpecBit_rollcall.hpp successfully amended.
File ../DecayBit/src/DecayBit.cpp successfully amended.
File ../DecayBit/include/gambit/DecayBit/DecayBit_rollcall.hpp successfully amended.
File ../DecayBit/include/gambit/DecayBit/DecayBit_rollcall.hpp successfully amended.
File ../DecayBit/src/DecayBit.cpp successfully amended.
File ../DarkBit/include/gambit/DarkBit/DarkBit_rollcall.hpp successfully amended.
File ../DarkBit/include/gambit/DarkBit/DarkBit_rollcall.hpp successfully amended.
File ../DarkBit/src/MDMSM.cpp successfully created.
File ../DarkBit/include/gambit/DarkBit/DarkBit_rollcall.hpp successfully amended.
Model MDMSM added to capability RD_spectrum.
Model MDMSM added to capability RD_eff_annrate.
File ../Backends/src/frontends/CalcHEP_3_6_27.cpp successfully amended.
File ../Backends/src/frontends/CalcHEP_3_6_27.cpp successfully amended.
File ../Backends/include/gambit/Backends/frontends/CalcHEP_3_6_27.hpp successfully amended.
```

```
l FeynRules file

DM sector (LDM),
Lagrangian (LSM)
M.fr

plify output

didate

t in this model.
```

s



Model defined in a FeynRules/SARAH file

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \frac{1}{2}\overline{\chi}\left(i\partial\!\!\!/ - m_\chi\right)\chi + \frac{1}{2}\partial_\mu Y \partial^\mu Y - \frac{1}{2}m_Y^2 Y^2 - \frac{g_\chi}{2}\overline{\chi}\chi Y - \frac{c_Y}{2}\sum_f y_f \overline{f} f Y\,.$$

Write a .gum file

```
math:
  # Choose FeynRules
  package: feynrules
  # Name of the model
  model: MDMSM
  # Model builds on the Standard Model FeynRules file
  base_model: SM
  # The Lagrangian is defined by the DM sector (LDM),
  # defined in MDMSM.fr, plus the SM Lagrangian (LSM)
  # imported from the ‘base model’, SM.fr
  Lagrangian: LDM + LSM
  # Make CKM matrix = identity to simplify output
  restriction: DiagonalCKM

# PDG code of the annihilating DM candidate
# in the FeynRules file
wimp_candidate: 52

# Select outputs for DM physics.
# Collider physics is not as important in this model.
output:
  pythia: false
  calchep: true
  micromegas: true
```

Run GUM

```
./gum -f Tutorial/MDMSM.gum
```

Compile GAMBIT + backends

```
cd ../build
cmake ..
make micromegas_MDMSM
make calchep
make -jn gambit
```

Model defined in a FeynRules/SARAH file

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \frac{1}{2}\overline{\chi}\left(i\partial\!\!\!/ - m_\chi\right)\chi + \frac{1}{2}\partial_\mu Y \partial^\mu Y - \frac{1}{2}m_Y^2 Y^2 - \frac{g_\chi}{2}\overline{\chi}\chi Y - \frac{c_Y}{2}\sum_f y_f \overline{f} f Y\,.$$

Write a .gum file

```
math:
  # Choose FeynRules
  package: feynrules
  # Name of the model
  model: MDMSM
  # Model builds on the Standard Model FeynRules file
  base_model: SM
  # The Lagrangian is defined by the DM sector (LDM),
  # defined in MDMSM.fr, plus the SM Lagrangian (LSM)
  # imported from the 'base model', SM.fr
  Lagrangian: LDM + LSM
  # Make CKM matrix = identity to simplify output
  restriction: DiagonalCKM

# PDG code of the annihilating DM candidate
# in the FeynRules file
wimp_candidate: 52

# Select outputs for DM physics.
# Collider physics is not as important in this model.
output:
  pythia: false
  calchep: true
  micromegas: true
```

Run GUM

```
./gum -f Tutorial/MDMSM.gum
```

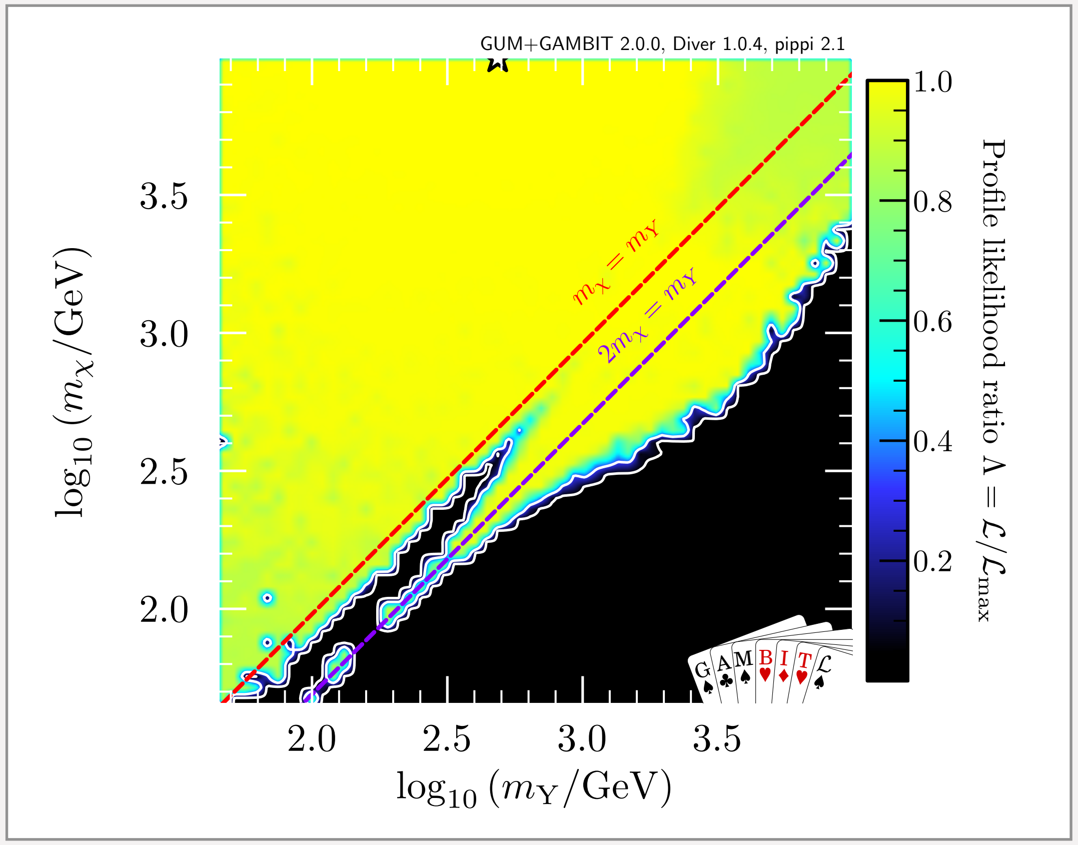
Compile GAMBIT + backends

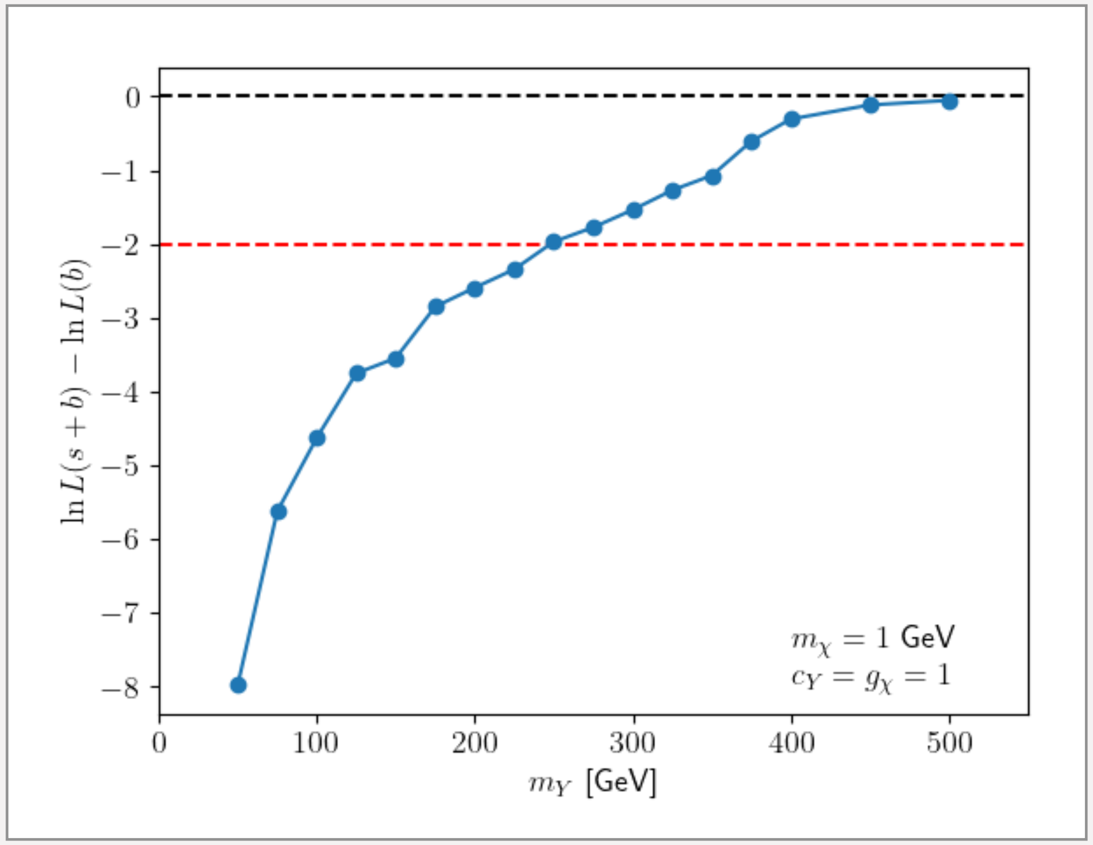
```
cd ../build
cmake ..
make micromegas_MDMSM
make calchep
make -jn gambit
```

Adjust GAMBIT input file

```
# Our dark matter model, implemented by GUM
MDMSM:
  mchi:
    range: [45, 10000]
    prior_type: log
  mY:
    range: [45, 10000]
    prior_type: log
  gchi:
    range: [1e-4, 12.566]
    prior_type: log
  cY:
    range: [1e-4, 12.566]
    prior_type: log

# Default halo parameters for the example
Halo_gNFW_rho0:
  rho0: 0.3
  v0: 240
  vesc: 533
  vrot: 240
  rs: 20.0
  r_sun: 8.5
  alpha: 1
  beta: 3
  gamma: 1
```

Run GAMBIT!

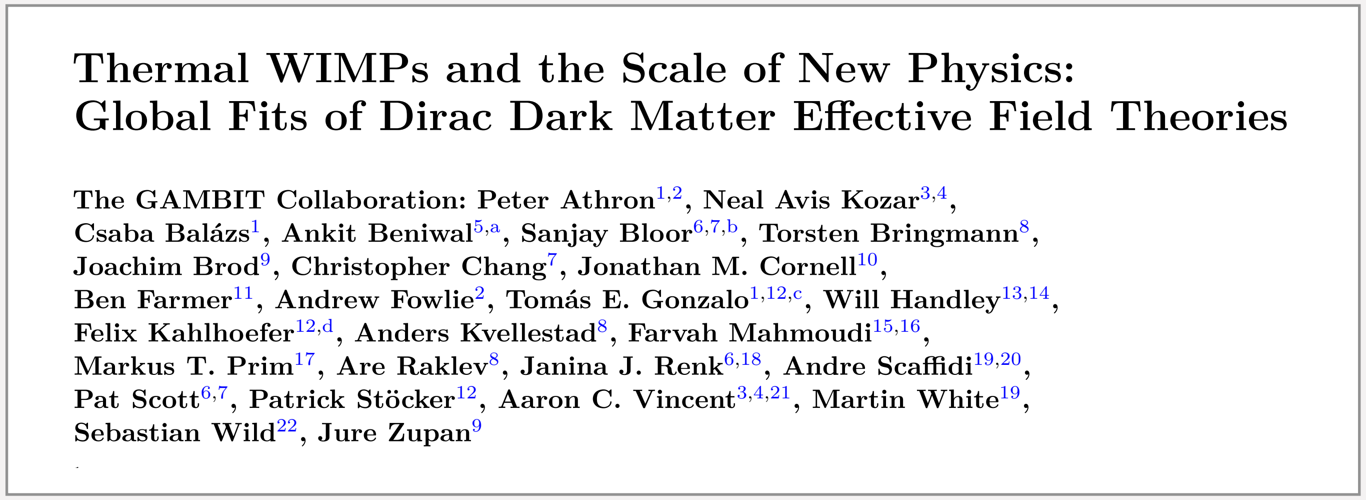




- **4D** scan (m_X, m_Y, g_X, c_Y)
- **Relic abundance** (as upper bound) [micrOMEGAs]
- **Direct detection**: XENON1T 2018, LUX 2016 [micrOMEGAs, DDCalc]
- **Indirect detection**: Fermi-LAT dwarf galaxies [CalcHEP, DarkSUSY, gamLike]
- **~11 hours on 4-core laptop**, sampling ~300k parameter points [Diver]

- Same model
- **1D** scan of m_Y
- $m_X = 1$ GeV, $g_X = 1$, $c_Y = 1$
- **Collider:** ATLAS 2lep+jets+MET, 139 fb^{-1} [Pythia, ColliderBit]
- Light m_Y disfavoured, but can easily be accommodated in the larger 4D parameter space

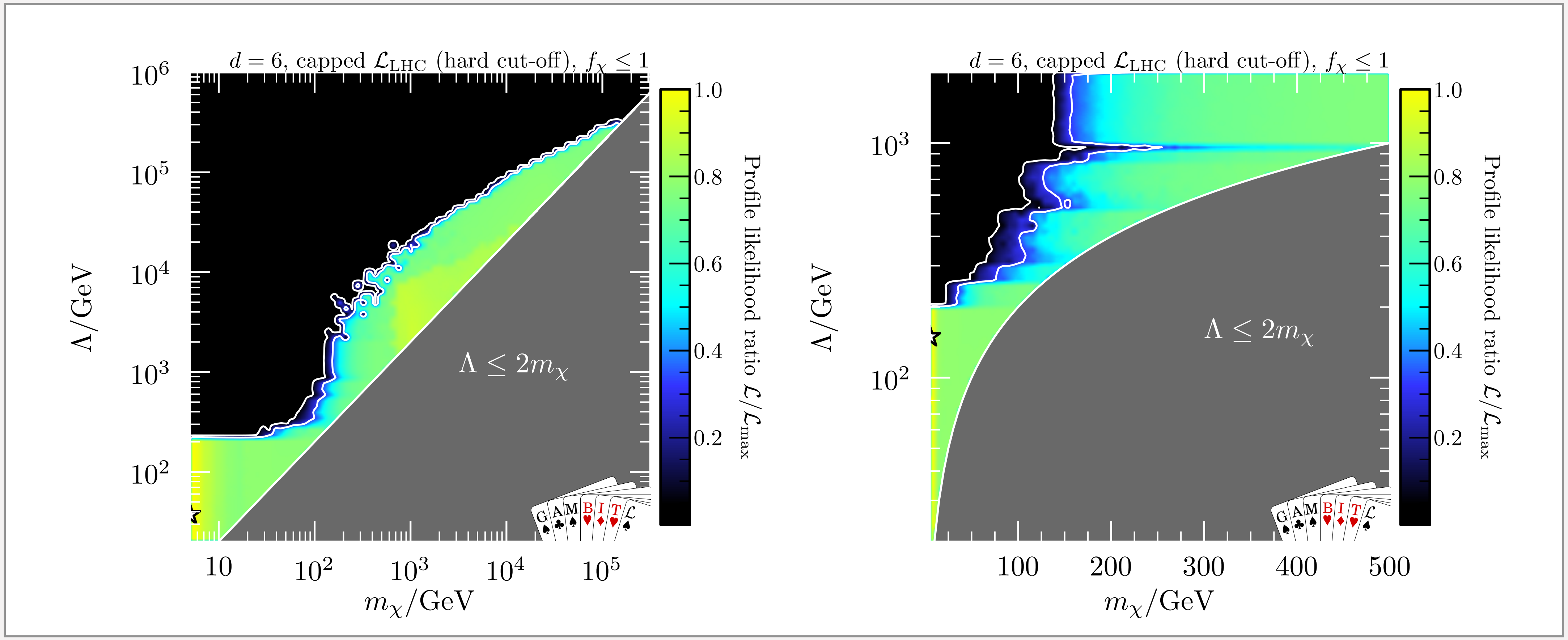
GUM used in recent GAMBIT dark matter study



Thermal WIMPs and the Scale of New Physics:
Global Fits of Dirac Dark Matter Effective Field Theories

The GAMBIT Collaboration: Peter Athron[1,2], Neal Avis Kozar[3,4], Csaba Balázs[1], Ankit Beniwal[5,a], Sanjay Bloor[6,7,b], Torsten Bringmann[8], Joachim Brod[9], Christopher Chang[7], Jonathan M. Cornell[10], Ben Farmer[11], Andrew Fowlie[2], Tomás E. Gonzalo[1,12,c], Will Handley[13,14], Felix Kahlhoefer[12,d], Anders Kvellestad[8], Farvah Mahmoudi[15,16], Markus T. Prim[17], Are Raklev[8], Janina J. Renk[6,18], Andre Scaffidi[19,20], Pat Scott[6,7], Patrick Stöcker[12], Aaron C. Vincent[3,4,21], Martin White[19], Sebastian Wild[22], Jure Zupan[9]

[arXiv:2106.02056]



Summary

- **Global fits are great.** We should do more of those.
- The **core GAMBIT framework is model-independent**
- …but so far it's taken a lot of work to set up GAMBIT + backends for new theories
- Coming soon: **GAMBIT 2.0 w/ GUM**
- Auto-generation of GAMBIT code + interfaces for calculations of **mass spectrum**, **decays**, **dark matter observables**, **collider physics** and **vacuum stability**
- gambit.hepforge.org
- zenodo.org/communities/gambit-official