Finding our way through BSM parameter spaces

Anders Kvellestad, University of Oslo

Nikhef, Amsterdam — 3 May 2018

UiO : **University of Oslo**

Outline

1. Why are global fits necessary?

2. Why are global fits difficult?

3. GAMBIT and what it can do for you

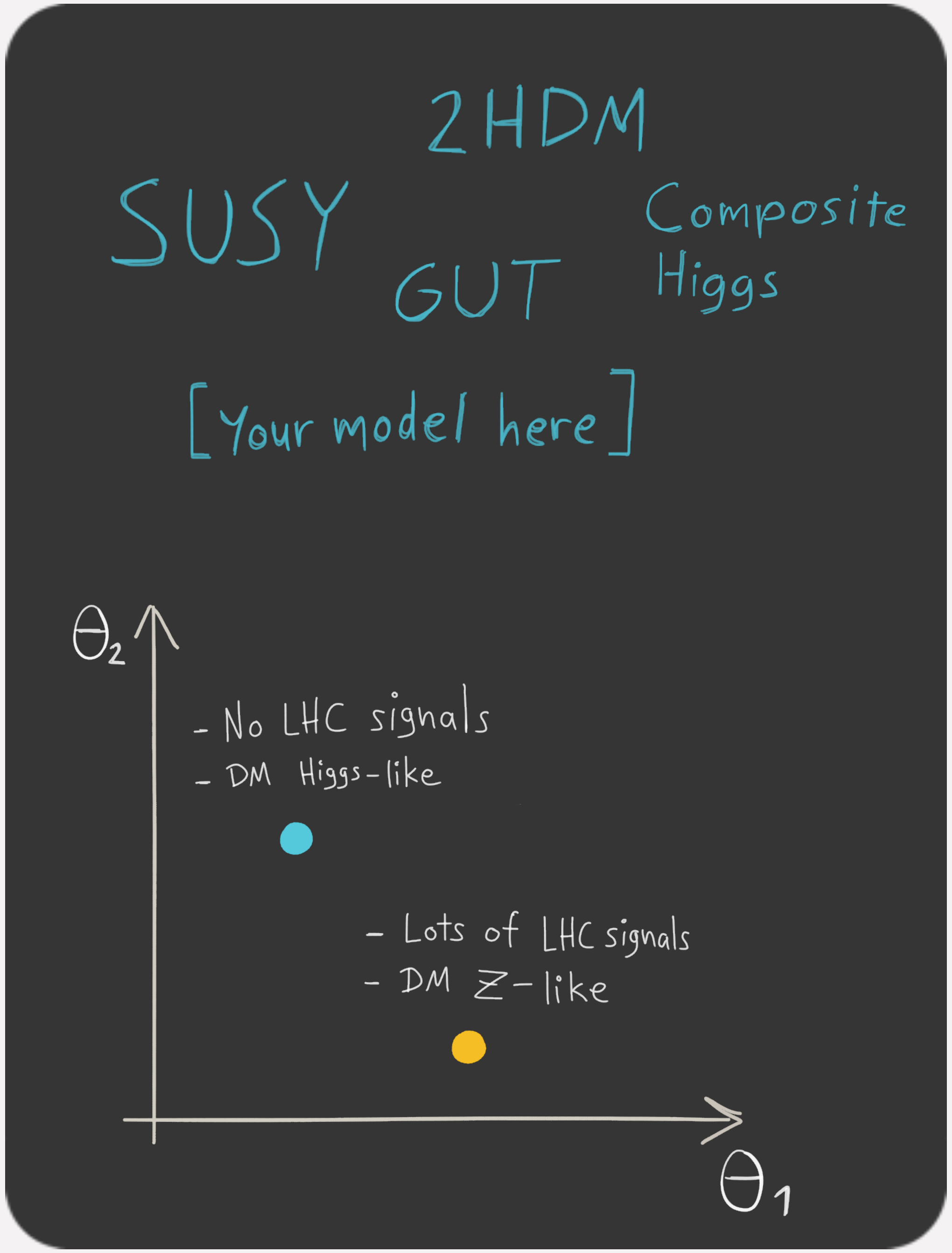
4. GAMBIT physics results

5. Summary and outlook

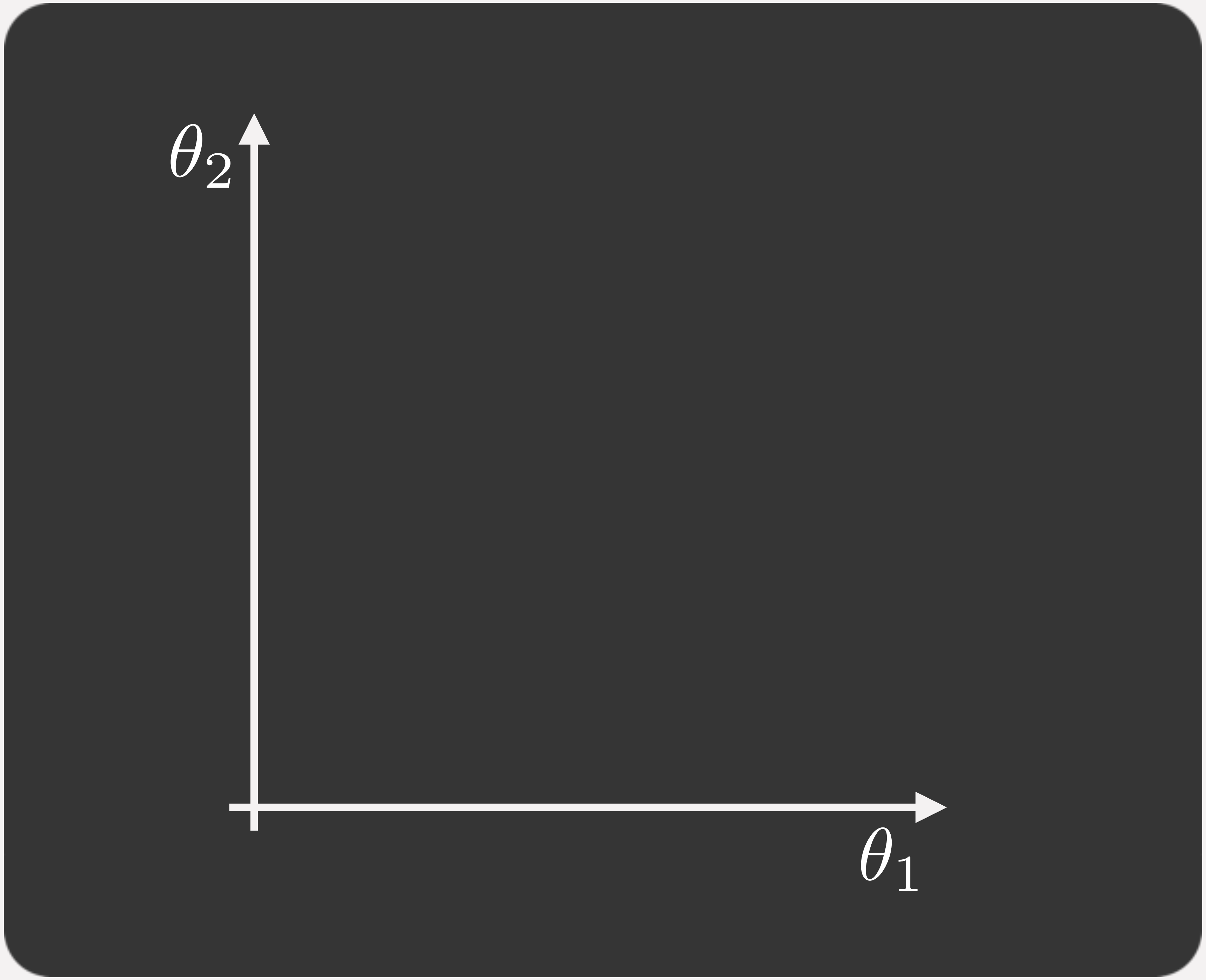
1. Why are global fits necessary?

Comparing BSM theories to data

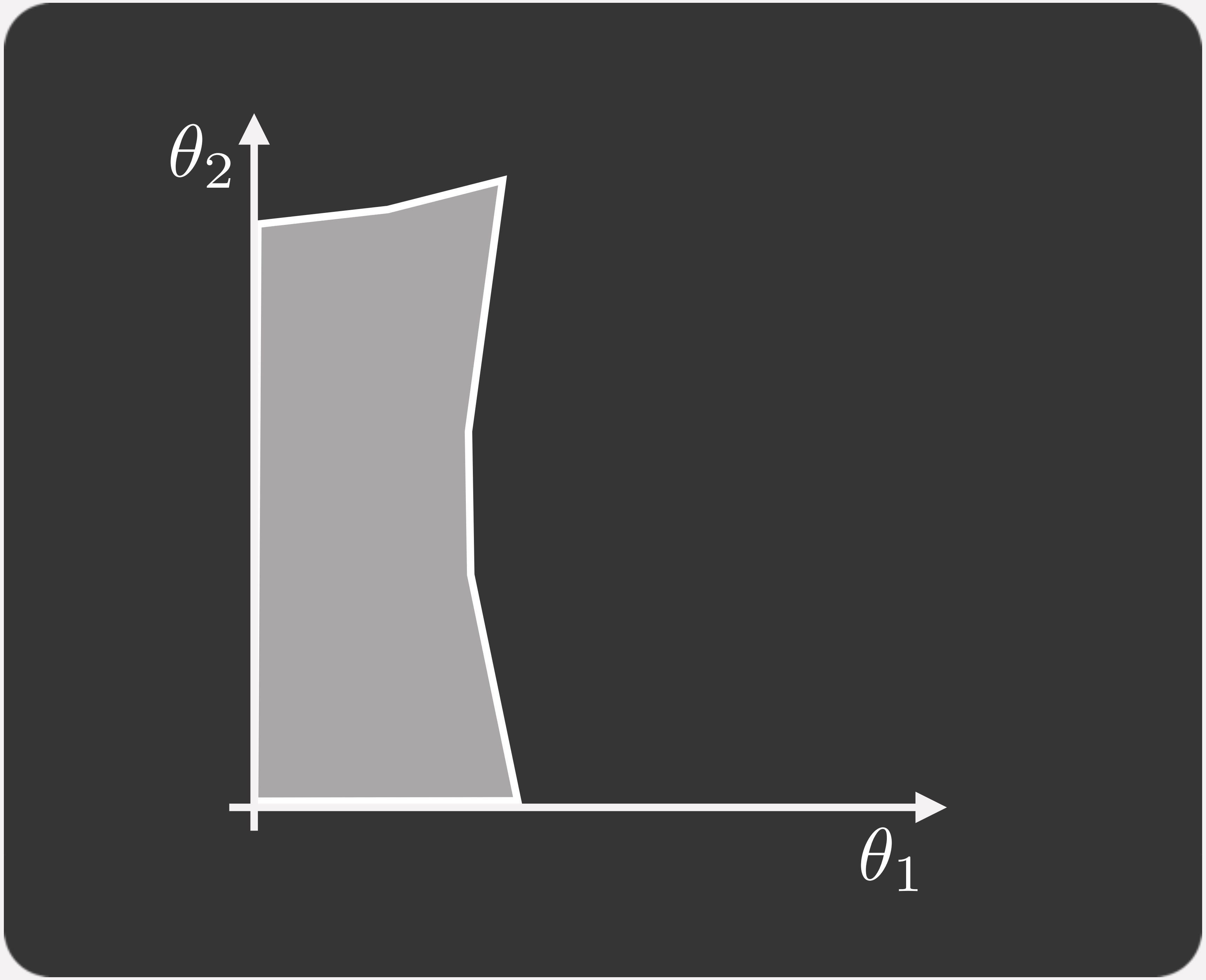
- Lots of theories for BSM physics
- For each theory, a parameter space of varying phenomenology
- Many different experiments can constrain each theory



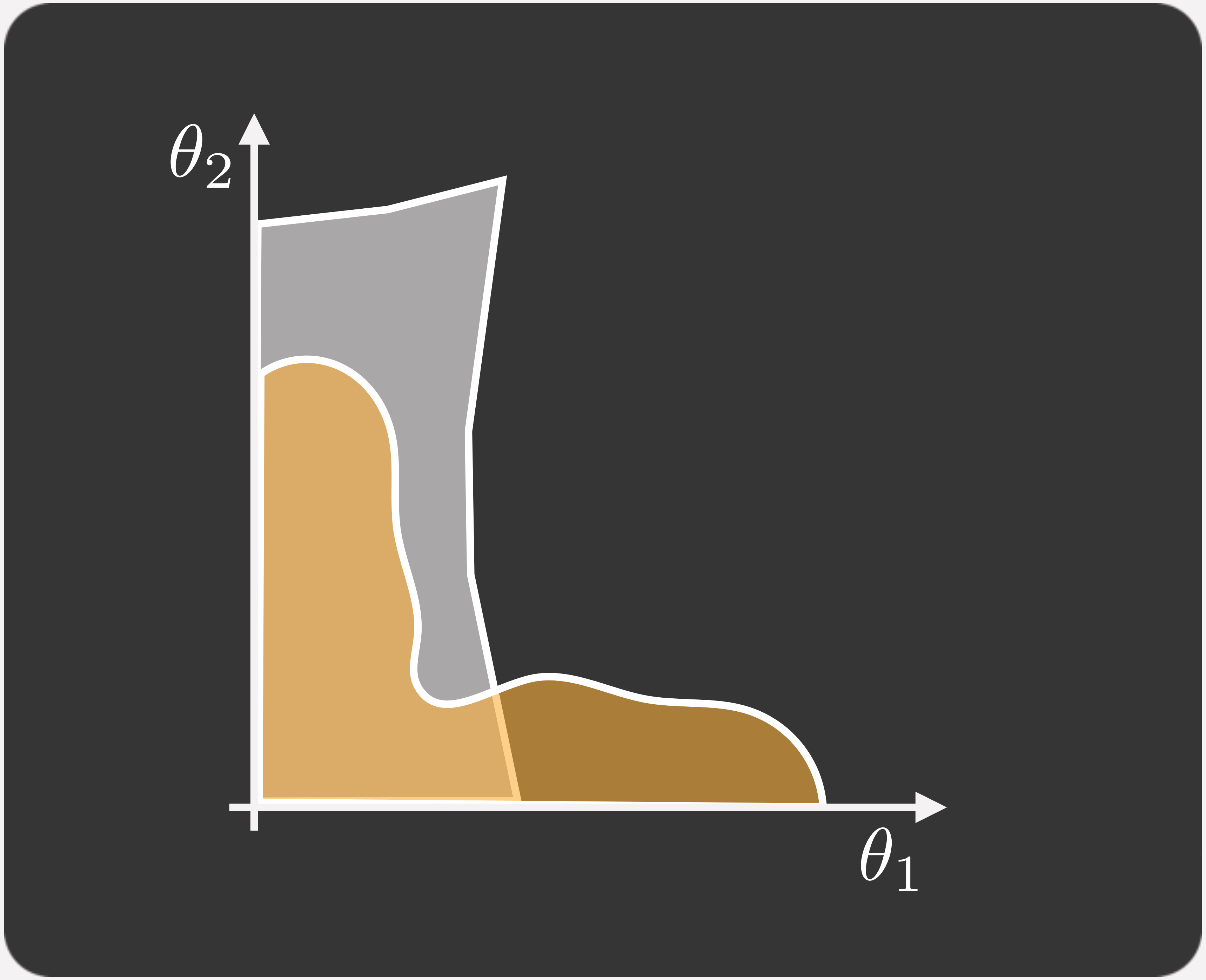
Comparing BSM theories to data



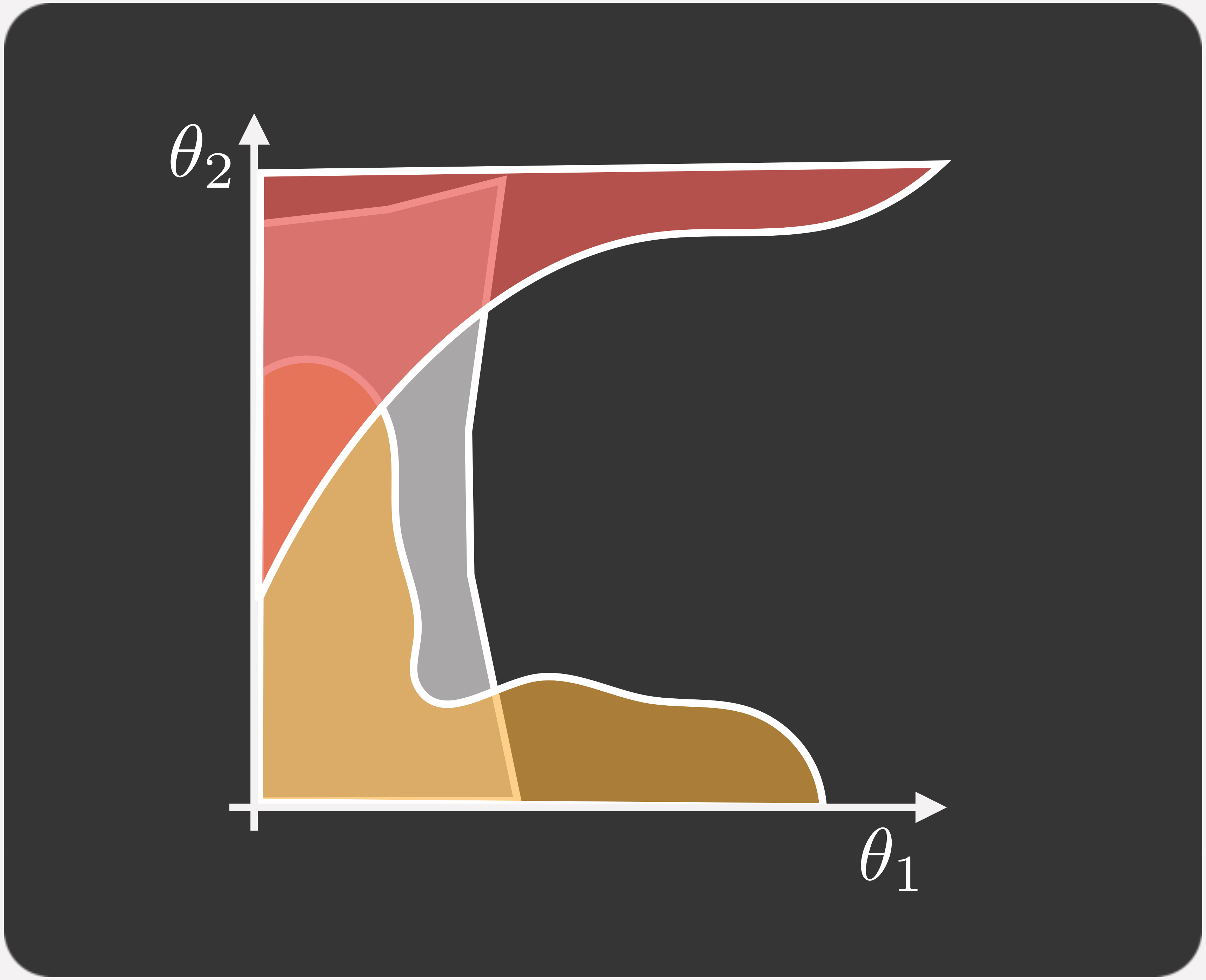
Comparing BSM theories to data



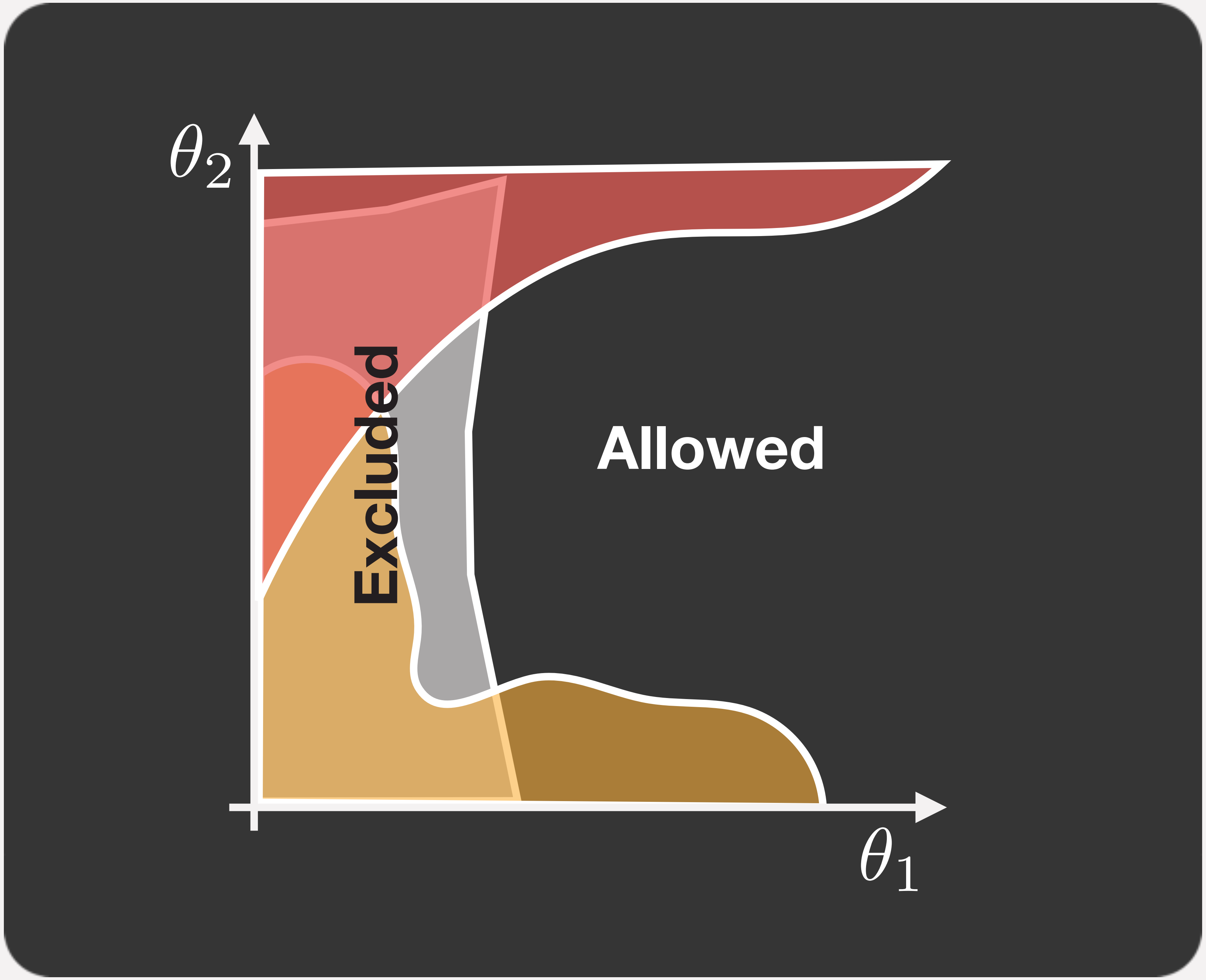
Comparing BSM theories to data



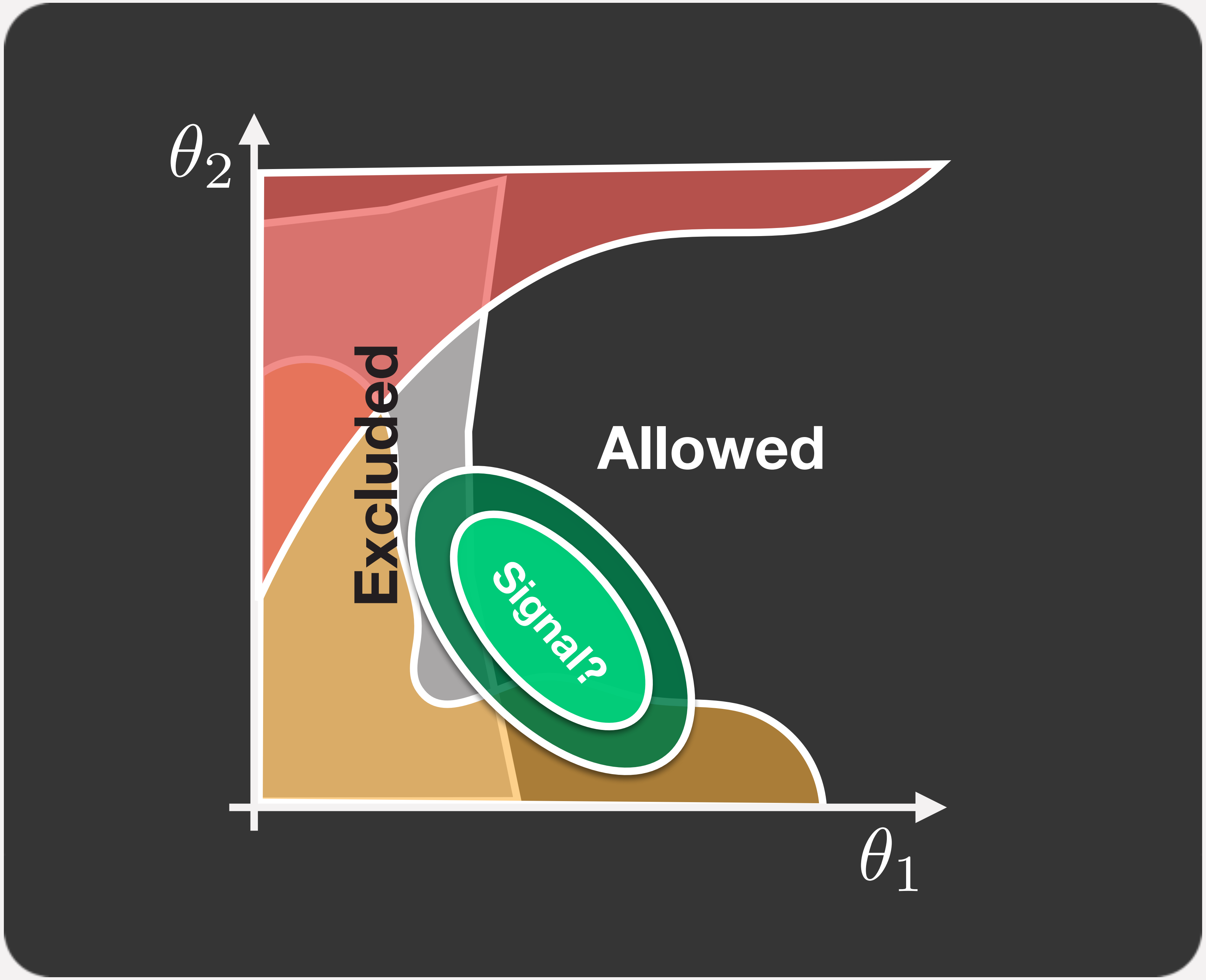
Comparing BSM theories to data



Comparing BSM theories to data



Comparing BSM theories to data



Comparing BSM theories to data

What to do when there are many parameters and many constraints?
A **global fit**.

$$\text{Theory} \rightarrow f(x;\theta)$$

$$\text{Experiment} \rightarrow \mathcal{L}(\theta) = f(x_{data};\theta)$$

$$\mathcal{L} = \mathcal{L}_{Collider}\,\mathcal{L}_{Higgs}\,\mathcal{L}_{DM}\,\mathcal{L}_{EWPO}\,\mathcal{L}_{Flavor}\ldots$$

Global fits

- Calculate **combined likelihood function** including observables from collider physics, dark matter, flavor physics, +++

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

- Use **sophisticated scanning techniques** to explore likelihood function across the parameter space of the theory
- Test **parameter regions** in a statistically sensible way — not just single points *(parameter estimation)*
- Test **different theories the same way** *(model comparison)*

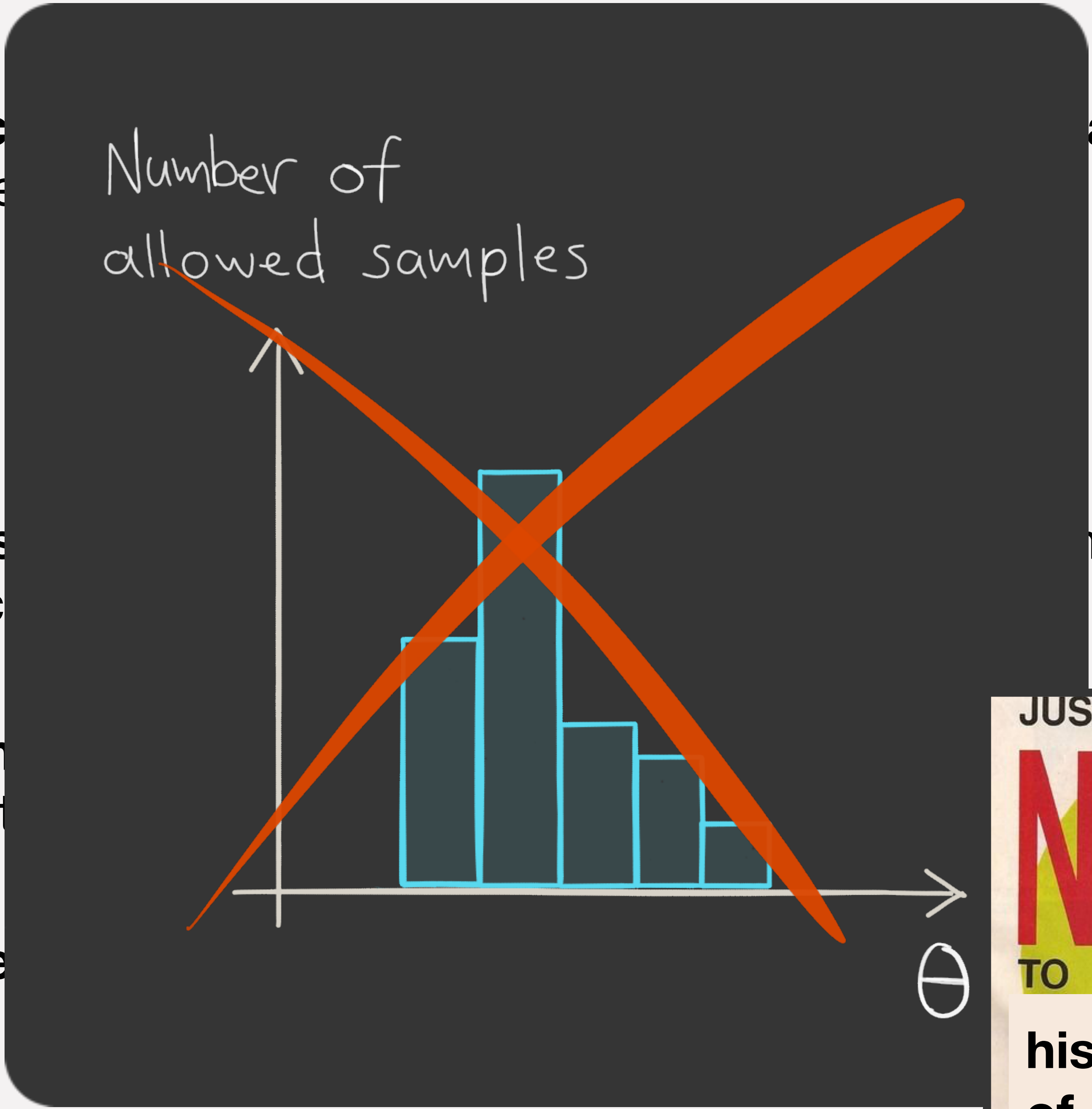
Global fits

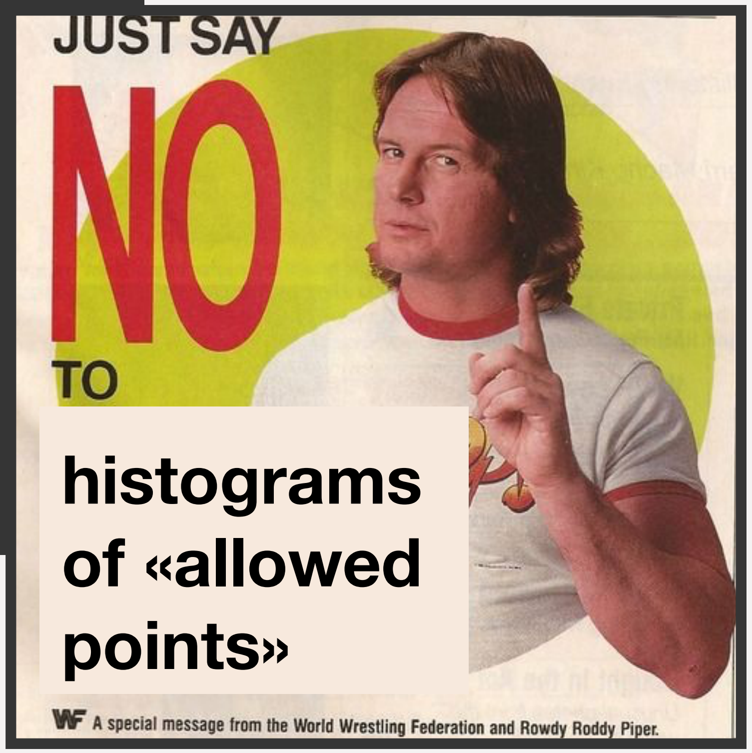
- Calculate c
  from collide

- Use sophis
  function ac

- Test param
  single point

- Test differe





ables

.

hood

Global fits

- Calculate **combined 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 likelihood function across the parameter space of the theory

- Test **parameter regions** in a statistically sensible way — not just single points *(parameter estimation)*

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

2. Why are global fits difficult?

[large number of observables]

$$\times$$

[long calculation time per observable per parameter point]

$$\times$$

[huge number of points required to explore parameter space]

$$\approx$$

$$\infty$$

[large number of observables]

$\times$

[long calculation time per observable per parameter point]

$\times$

[huge number of points required to explore parameter space]

$\approx$

∞

Large number of observables

- Need many observables to constrain many parameters
- Observable calculations generally introduce additional parameters → **nuisance parameters**
- Typically need to **interface several external tools**
- Need **consistent treatment of uncertainties** across different calculations and codes
- Need to make observable calculations as **reusable** as possible

[large number of observables]

$$\times$$

[long calculation time per observable per parameter point]

$$\times$$

[huge number of points required to explore parameter space]

$$\approx$$

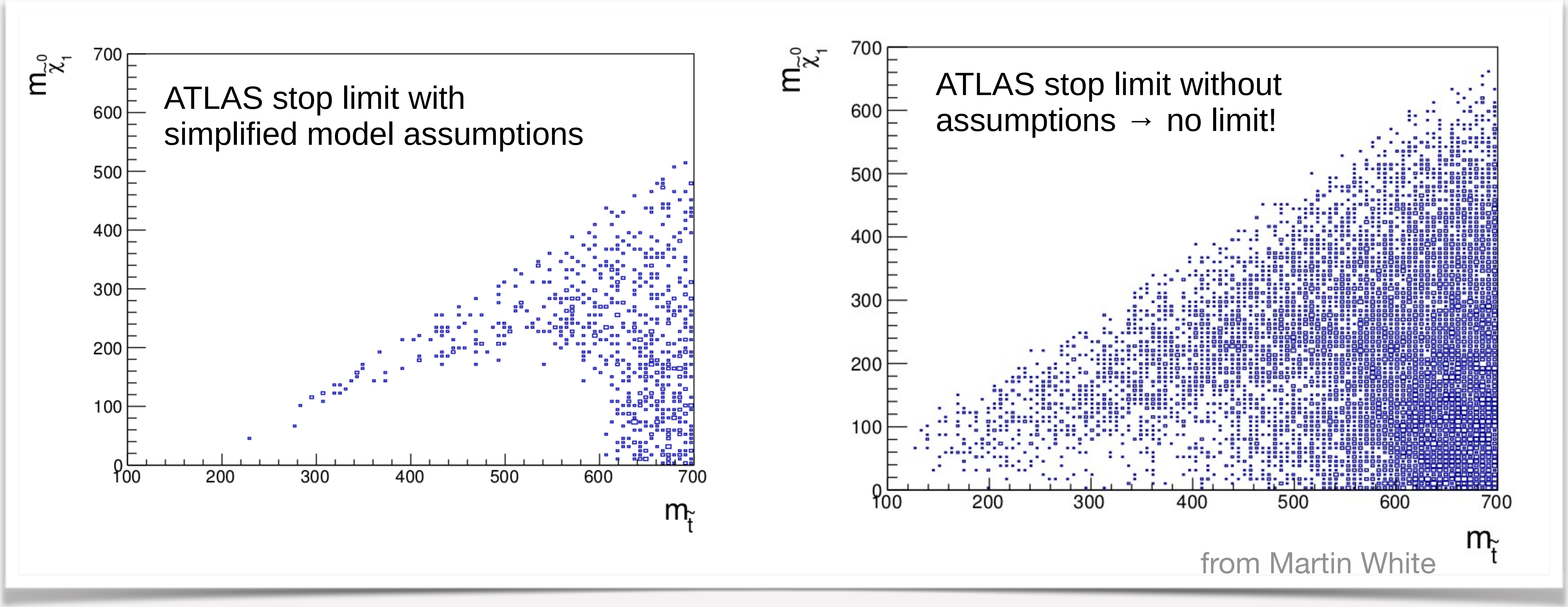
$$\infty$$

Long time per observable

- Some observables require very time-consuming calculations
  (*e.g.* MC simulation of LHC searches)
- **Sort** likelihood calculations from quickest to slowest
- **Optimize** calculations as much as possible
- **Parallelize** calculations
- Make **approximations** when valid
  Typically requires expert knowledge

Long time per observable

- Some observables require very time-consuming calculations
  (*e.g.* MC simulation of LHC searches)



Long time per observable

- Some observables require very time-consuming calculations
  (*e.g.* MC simulation of LHC searches)
- **Sort** likelihood calculations from quickest to slowest
- **Optimize** calculations as much as possible
- **Parallelize** calculations
- Make **approximations** when valid
  Typically requires expert knowledge

[large number of observables]

$\times$

[long calculation time per observable per parameter point]

$\times$

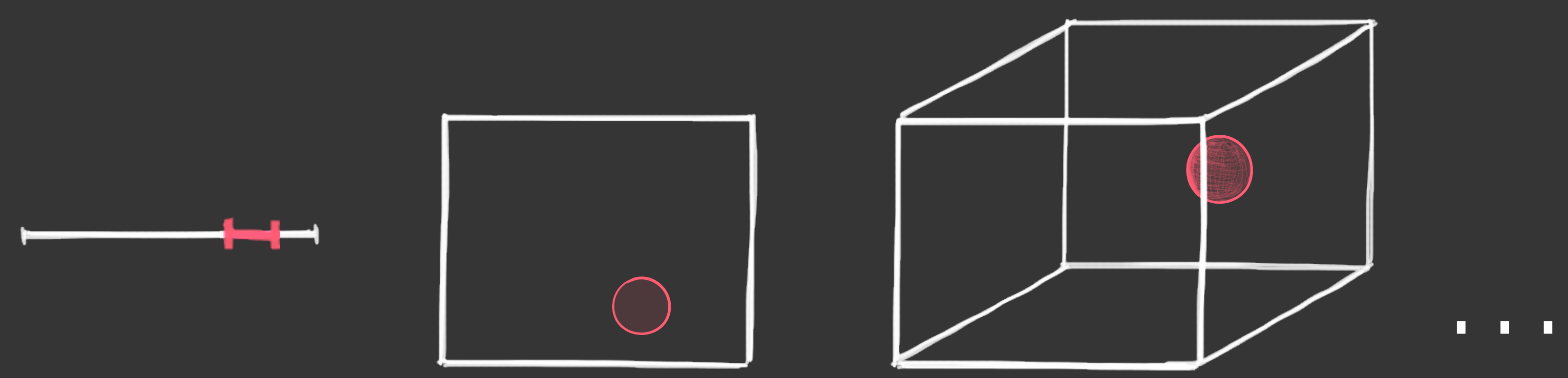
[huge number of points required to explore parameter space]

$\approx$

∞

Finding interesting parameter regions gets harder with increasing number of dimensions…

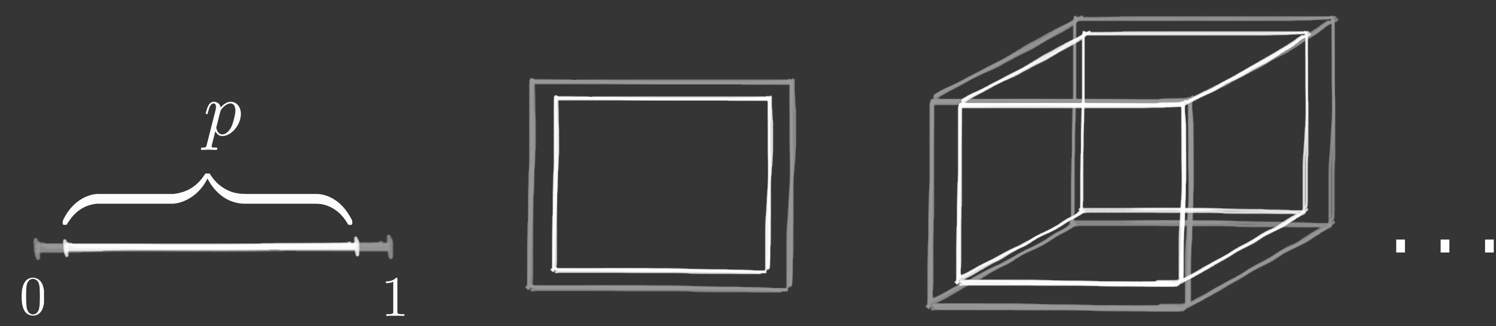
$$\lim_{D\to\infty} \frac{V_{\mathrm{interesting}}}{V_{\mathrm{total}}} = 0$$



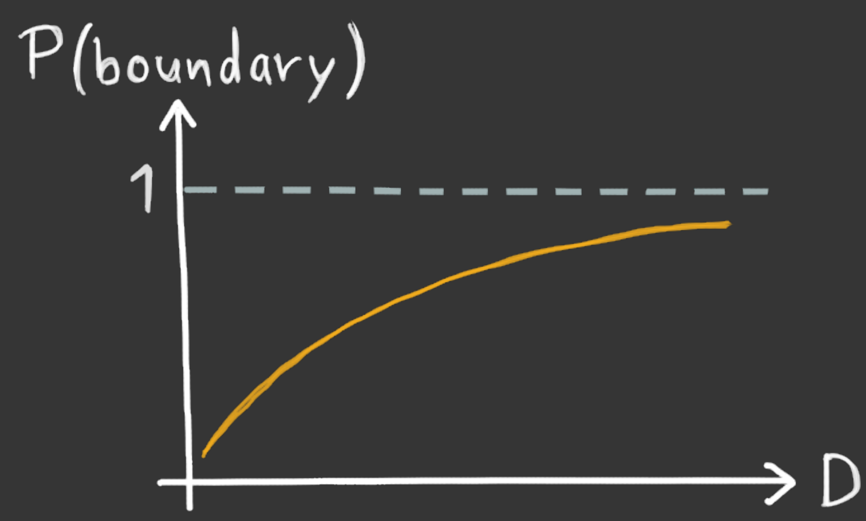
…so simply picking points «at random» will be **highly inefficient**…

…and it will mainly explore the **boundary** of the parameter space!

$$\vec{x} = (x_1, x_2, \ldots, x_D) \quad x_i \sim U(0,1)$$



$$P(\text{boundary}) = 1 - P(\text{not boundary}) = 1 - p^D$$



Need to use some **smart** sampling algorithms

[large number of observables]

$$\times$$

[long calculation time per observable per parameter point]

$$\times$$

[huge number of points required to explore parameter space]

$$\approx$$

$$\infty$$

How to tackle these challenges?
How to avoid reinventing the wheel for every new analysis?

3. GAMBIT and what it can do for you

GAMBIT

The Global And Modular BSM Inference Tool

- A **new** and **general** framework for BSM global fits
- Fully **open source**
- **Modular design:** can be extended with
  — new models
  — new likelihoods
  — new theory calculators
  — new scanning algorithms
- Use external codes (**backends**) as **runtime plugins**
  — Supported languages:
  C, C++, Fortran, Python and Mathematica
- **Two-level parallellization** with MPI and OpenMP
- **Hierarchical** model database
- **Flexible output streams** (ASCII, HDF5, …)
- Many **scanners** and **backends** already included

gambit.hepforge.org

GAMBIT: The Global And Modular BSM Inference Tool

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

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

ATLAS	F. Bernlochner, A. Buckley, P. Jackson, M. White
LHCb	M. Chrząszcz, N. Serra
Belle-II	F. Bernlochner, P. Jackson
Fermi-LAT	J. Conrad, J. Edsjö, G. Martinez, P. Scott
CTA	C. Balázs, T. Bringmann, M. White
CMS	C. Rogan
IceCube	J. Edsjö, P. Scott
XENON/DARWIN	B. Farmer, R. Trotta
Theory	P. Athron, C. Balázs, S. Bloor, T. Bringmann, J. Cornell, J. Edsjö, B. Farmer, A. Fowlie, T. Gonzalo, J. Harz, S. Hoof, F. Kahlhoefer, S. Krishnamurthy, A. Kvellestad, F.N. Mahmoudi, J. McKay, A. Raklev, R. Ruiz, P. Scott, R. Trotta, A. Vincent, C. Weniger, M. White, S. Wild

31 Members in 9 Experiments, 12 major theory codes, 11 countries

GAMBIT

What's in the box?

Core

- Models — EPJC, arXiv:1705.07908

Physics modules

- ColliderBit: *fast* LHC sim, Higgs searches, LEP SUSY limits — EPJC, arXiv:1705.07919
- DarkBit: relic density, gamma ray signal yields, ID/DD likelihoods — EPJC, arXiv:1705.07920
- FlavBit: wide range of flavour observables & likelihoods — EPJC, arXiv:1705.07933
- SpecBit: spectrum objects, RGE running
- DecayBit: decay widths
- PrecisionBit: precision BSM tests

(SpecBit, DecayBit, PrecisionBit) — EPJC, arXiv:1705.07936

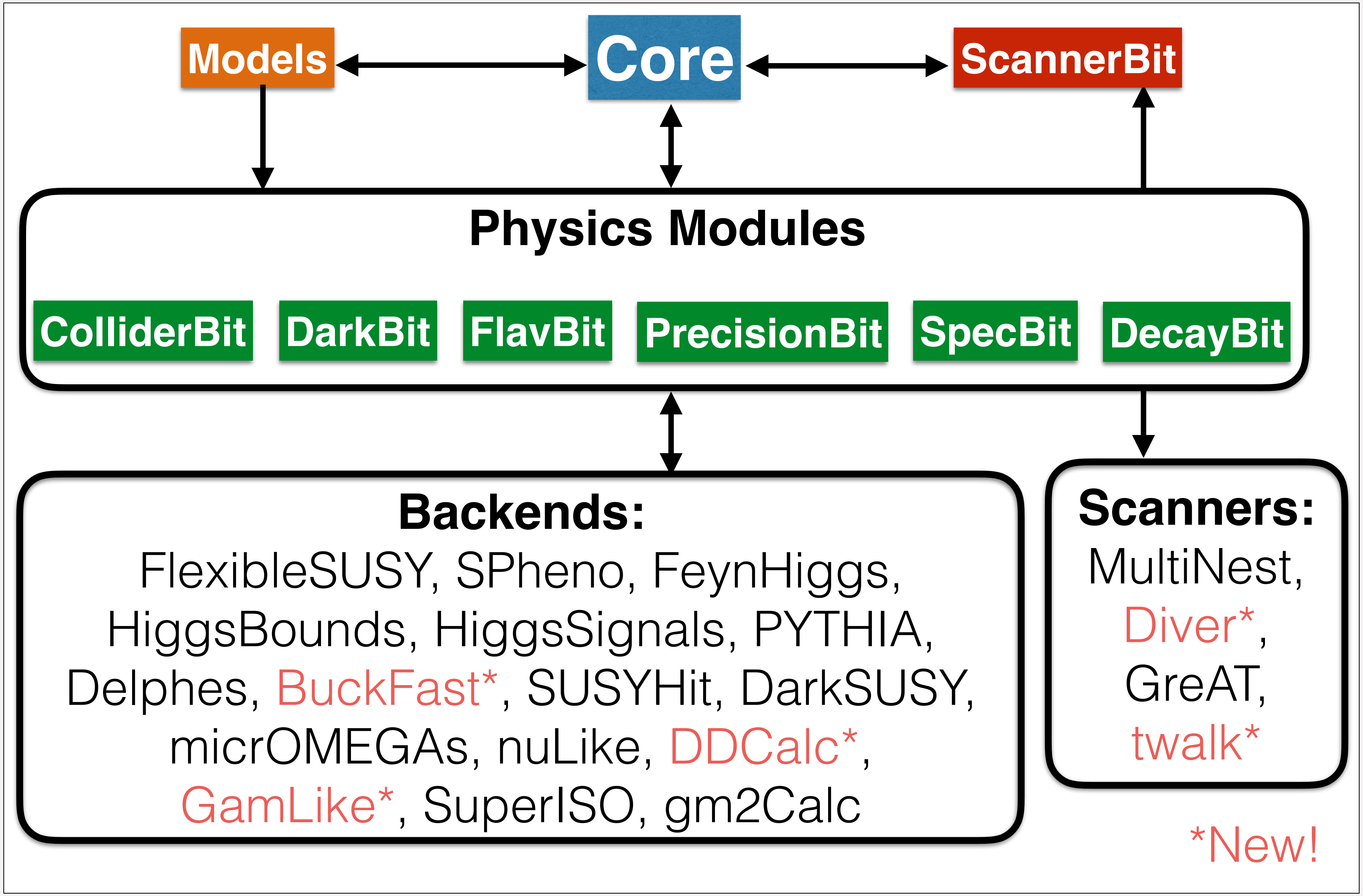
Statistics and sampling

- ScannerBit: stats & sampling (Diver, MultiNest, T-Walk, ++) — EPJC, arXiv:1705.07959

Backends (external tools)

GAMBIT

Code structure



GAMBIT

Code structure

- Basic building blocks: **module functions**
- A physics module: **a collection of module functions** related to the same physics topic
- Each module function has a single **capability** (what it calculates)
- A module function can have **dependencies** on the results of other module functions
- A module function can declare which **models** it can work with
- GAMBIT determines which module functions should be run in which order for a given scan (**dependency resolution**)

```
void function_name(double &result)
{
  ...
  result = ... // something useful
}
```

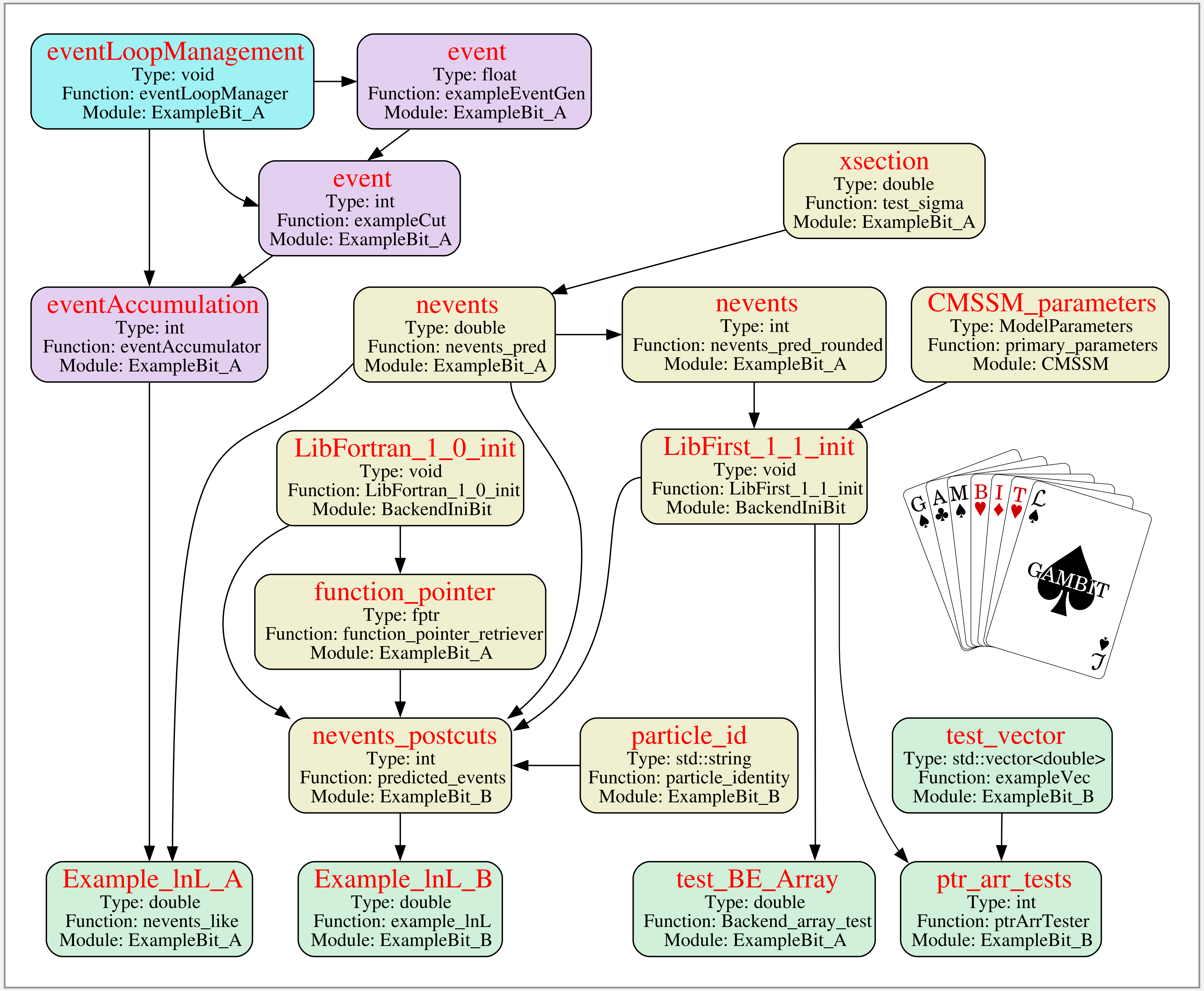
```
// Observable: BR(B -> tau nu)
#define CAPABILITY Btaunu
START_CAPABILITY
  #define FUNCTION SI_Btaunu
  START_FUNCTION(double)
  DEPENDENCY(SuperIso_modelinfo, parameters)
  BACKEND_REQ(Btaunu, (libsuperiso), double, (const parameters*))
  BACKEND_OPTION( (SuperIso, 3.6), (libsuperiso) )
  #undef FUNCTION
#undef CAPABILITY
```

```
/// Br B->tau nu_tau decays
void SI_Btaunu(double &result)
{
  using namespace Pipes::SI_Btaunu;

  parameters const& param = *Dep::SuperIso_modelinfo;
  result = BEreq::Btaunu(&param);
}
```

GAMBIT

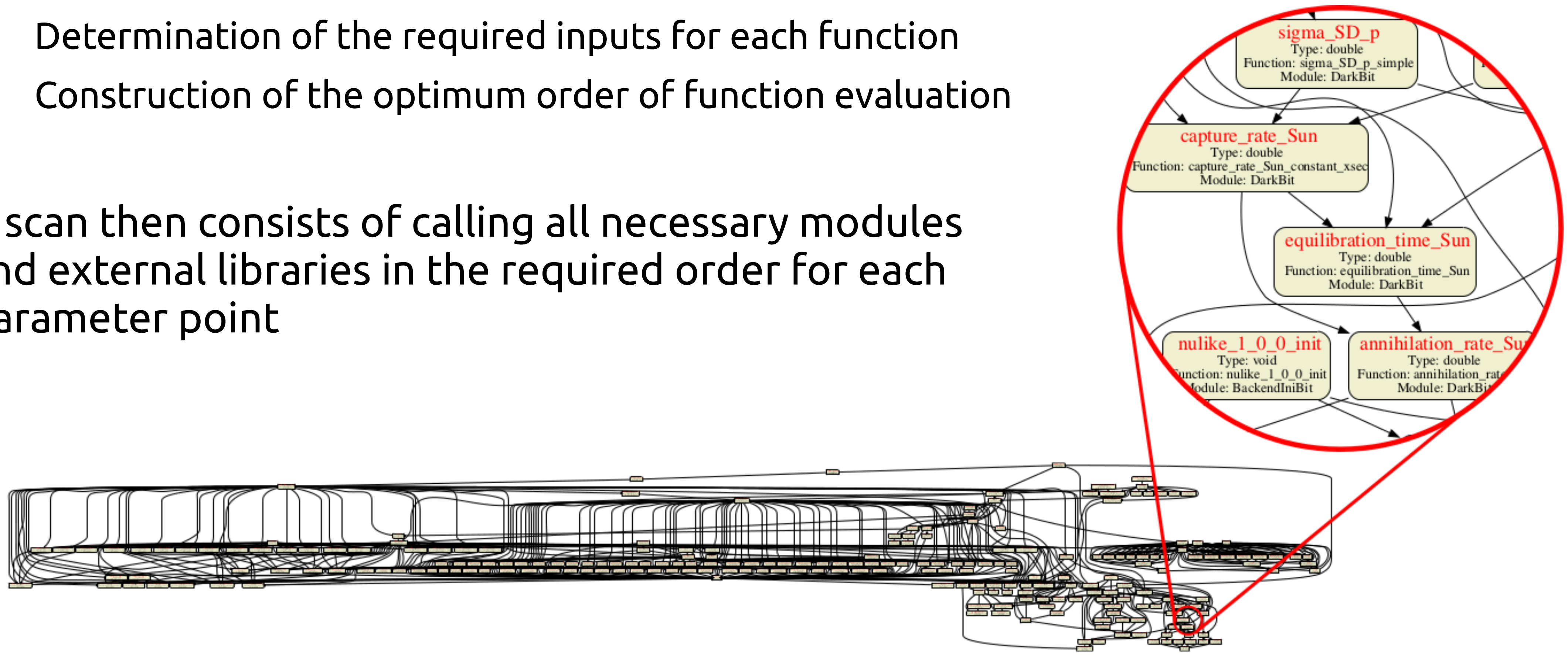
Dependency resolution



GAMBIT

Dependency resolution

- User specifies the model, parameter space, observables and scanning technique
- GAMBIT then performs the *dependency resolution*
  - Identification of all functions necessary to calculate requested observables
  - Dynamic adaptation to the user’s system
  - Determination of the required inputs for each function
  - Construction of the optimum order of function evaluation
- A scan then consists of calling all necessary modules and external libraries in the required order for each parameter point

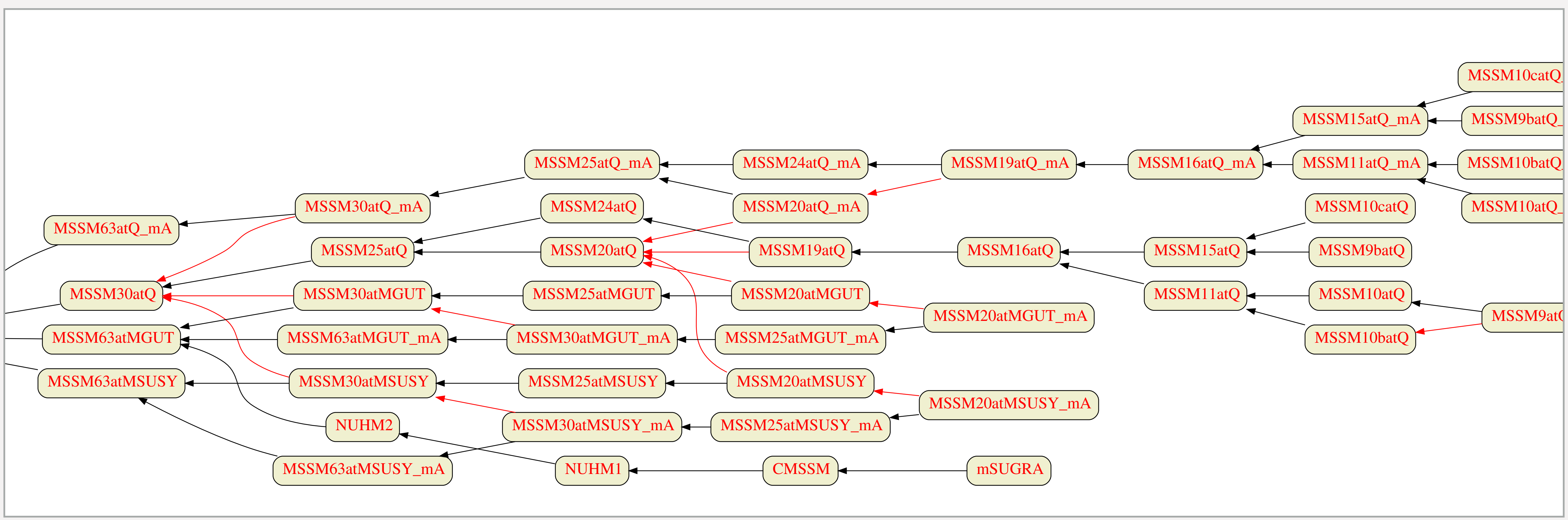


from Felix Kahlhoefer

GAMBIT

Hierarchical model database

- A **model** is a collection of named parameters
- Models can be **related** (e.g. MSSM9 is a parent of MSSM7)
- Points in child model **automatically translated** to ancestor models
- Ensures **maximum reuse** of calculations and minimizes risk of mistakes



YAML files

```yaml
Parameters:

  # SM parameters.
  StandardModel_SLHA2: !import include/StandardModel_SLHA2_scan.yaml

  # Nuclear matrix parameters.
  nuclear_params_sigmas_sigmal:
    sigmas:
      range:  [19, 67]
    sigmal:
      range:  [31, 85]
    deltau:
      fixed_value:  0.842
    deltad:
      fixed_value: -0.427
    deltas:
      fixed_value: -0.085

  # SUSY parameters.
  MSSM7atQ:
    Qin:
      fixed_value: 1000.0
    SignMu:
      fixed_value: 1
    Ad_3:
      prior_type: double_log_flat_join
      ranges: [-1e4, -1e2, 1e2, 1e4]
    Au_3:
      prior_type: double_log_flat_join
      ranges: [-1e4, -1e2, 1e2, 1e4]
    M2:
      prior_type: double_log_flat_join
      ranges: [-1e4, -1e2, 1e2, 1e4]
    TanBeta:
      range: [3.0, 70.0]
      prior_type: flat
    mHd2:
      prior_type: double_log_flat_join
      ranges: [-1e8, -1e4, 1e4, 1e8]
    mHu2:
      prior_type: double_log_flat_join
      ranges: [-1e8, -1e4, 1e4, 1e8]
    mf2:
      prior_type: double_log_flat_join
      ranges: [0, 0, 1e4, 1e8]

  # Dark matter halo parameters.
  Halo_gNFW_rho0:
    rho0:
      range: [.2, .8]
    v0: 235.0
```

```yaml
Printer:

  printer: hdf5

  options:
    output_file: "MSSM7.hdf5"
    group: "/MSSM7"


Scanner:

  use_scanner: de

  scanners:

    multinest:
      plugin: multinest
      like:  LogLike
      nlive: 5000
      tol: 0.1
      updInt: 1

    de:
      plugin: diver
      like: LogLike
      NP: 19200
      convthresh: 1e-5
      verbosity: 1
```

```yaml
ObsLikes:

    # LHC likelihoods
    - purpose:    LogLike
      capability: LHC_Combined_LogLike

    - purpose:    LogLike
      capability: LHC_Higgs_LogLike

    # Dark matter likelihoods
    - capability: lnL_oh2
      purpose:    LogLike

    - capability: lnL_FermiLATdwarfs
      purpose:    LogLike

    - capability: XENON100_2012_LogLikelihood
      purpose:    LogLike

    - capability: XENON1T_2017_LogLikelihood
      purpose:    LogLike

    - capability: LUX_2015_LogLikelihood
      purpose:    LogLike

    - capability: LUX_2016_LogLikelihood
      purpose:    LogLike

    - capability: PandaX_2016_LogLikelihood
      purpose:    LogLike

    - capability: PICO_2L_LogLikelihood
      purpose:    LogLike

    - capability: PICO_60_F_LogLikelihood
      purpose:    LogLike

    - capability: PICO_60_2017_LogLikelihood
      purpose:    LogLike

    - capability: SuperCDMS_2014_LogLikelihood
      purpose:    LogLike

    - capability: SIMPLE_2014_LogLikelihood
      purpose:    LogLike

    - capability: IC79_loglike
      purpose:    LogLike

    # Flavour physics likelihoods
    - purpose:    LogLike
      capability: b2ll_LL
```

Backends

- A backend is an external code (C, C++, Fortran, Python, Mathematica)
- Connected to GAMBIT as a runtime plugin — not linked at compile time, and not called via command line interface with file input/output
- GAMBIT module functions can request results from backends
- Backend functions are tagged according to what they calculate (*e.g.* "Omegah2")
  - Switching between different backends by changing one line in the YAML file
  - Ideal for comparing different theory codes
- Many codes already supported, and more to come
  - But you can always add a new interface yourself

Backends

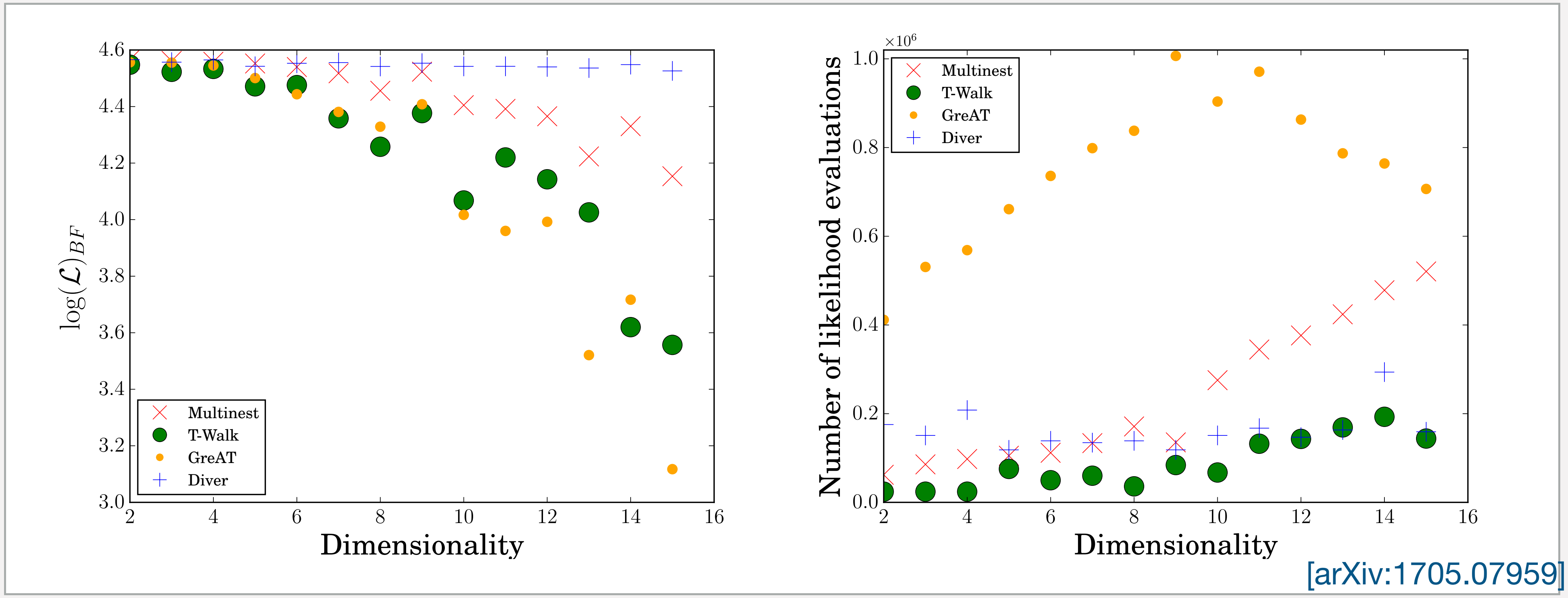
All relative paths are given with reference to /home/anders/physics/GAMBIT/gambit.

BACKENDS	VERSION	PATH TO LIB	STATUS	#FUNC	#TYPES
DDCalc	1.0.0	Backends/installed/ddcalc/1.0.0/lib/libDDCalc.so	absent/broken	36	0
	1.1.0	Backends/installed/ddcalc/1.1.0/lib/libDDCalc.so	absent/broken	38	0
	1.2.0	Backends/installed/ddcalc/1.2.0/lib/libDDCalc.so	OK	39	0
DarkSUSY	5.1.3	Backends/installed/darksusy/5.1.3/lib/libdarksusy.so	OK	79	0
FeynHiggs	2.11.2	Backends/installed/feynhiggs/2.11.2/lib/libFH.so	absent/broken	14	0
	2.11.3	Backends/installed/feynhiggs/2.11.3/lib/libFH.so	OK	14	0
	2.12.0	Backends/installed/feynhiggs/2.12.0/lib/libFH.so	absent/broken	14	0
HiggsBounds	4.2.1	Backends/installed/higgsbounds/4.2.1/lib/libhiggsbounds.so	absent/broken	10	0
	4.3.1	Backends/installed/higgsbounds/4.3.1/lib/libhiggsbounds.so	OK	10	0
HiggsSignals	1.4	Backends/installed/higgssignals/1.4.0/lib/libhiggssignals.so	OK	12	0
LibFarrayTest	1.0	Backends/examples/libFarrayTest.so	OK	10	0
LibFirst	1.0	Backends/examples/libfirst.so	OK	8	0
	1.1	Backends/examples/libfirst.so	OK	15	0
LibFortran	1.0	Backends/examples/libfortran.so	OK	6	0
LibSecond	1.0	Backends/examples/libsecond/1.0/libsecond.py	OK	6	0
	1.1	Backends/examples/libsecond/1.1/libsecond.py	OK	6	0
MicrOmegas_MSSM	3.6.9.2	Backends/installed/micromegas/3.6.9.2/MSSM/libmicromegas.so	OK	18	0
MicrOmegas_SingletDM	3.6.9.2	Backends/installed/micromegas/3.6.9.2/SingletDM/libmicromegas.so	OK	16	0
Pythia	8.212	Backends/installed/pythia/8.212/lib/libpythia8.so	OK	0	28
	8.212.EM	Backends/installed/pythia/8.212.EM/lib/libpythia8.so	absent/broken	0	28
SPheno	3.3.8	Backends/installed/spheno/3.3.8/lib/libSPheno.so	OK	282	0
SUSYHD	1.0.2	Backends/installed/susyhd/1.0.2/SUSYHD.m	Mathematica absent	3	0

Gambit diagnostic backend line 1 (press h for help or q to quit)

Scanners

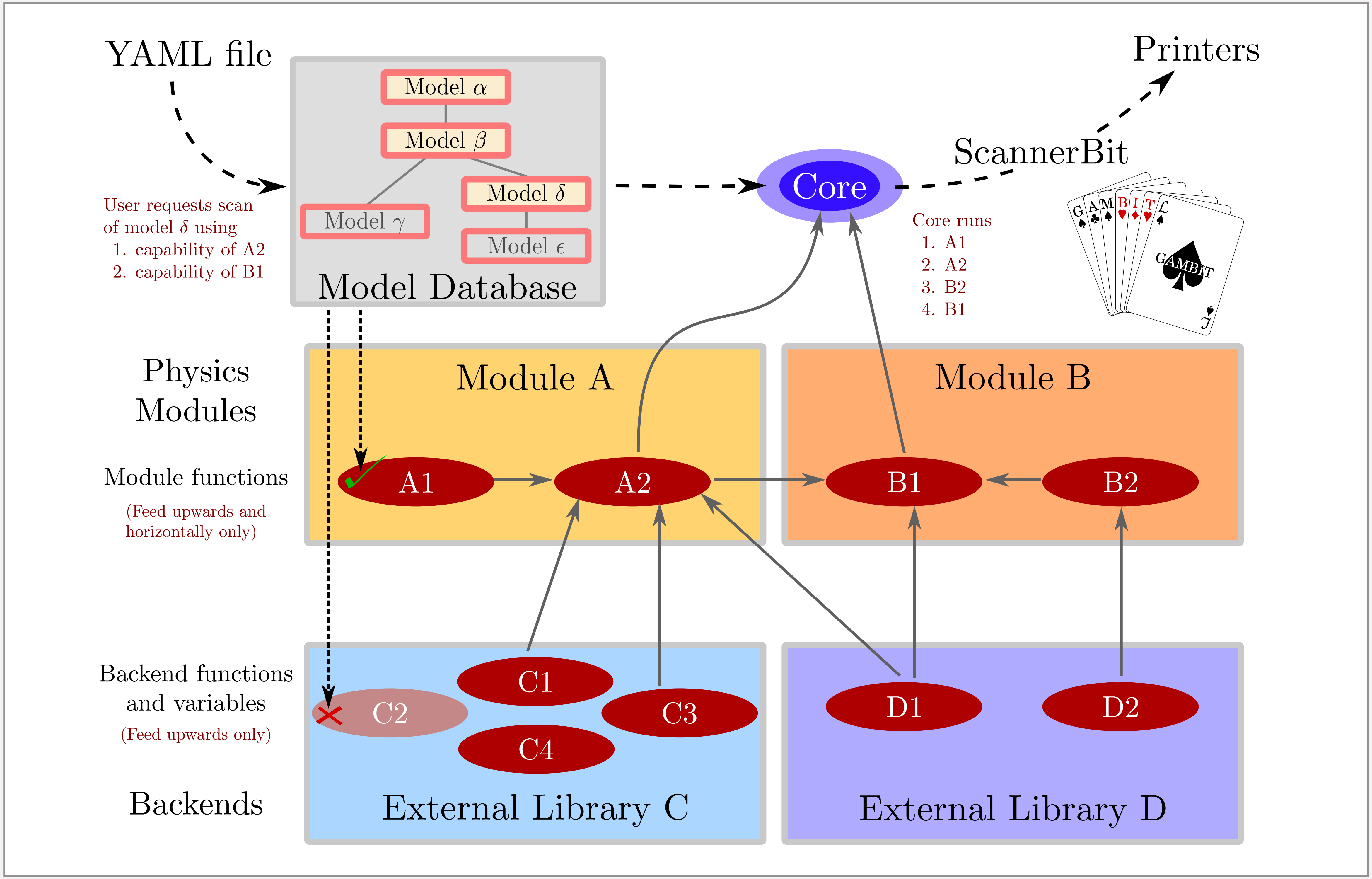
- Trivial to switch between scanner algorithms
- Choose the best scanner for your analysis
    - Profile likelihoods (frequentist)
    - Posterior distributions (Bayesian)
    - Evidence estimation (Bayesian)
    - Grid scans, post-processing, …
- Ideal for comparing the performance of different scanners (see arXiv:1705.07959)

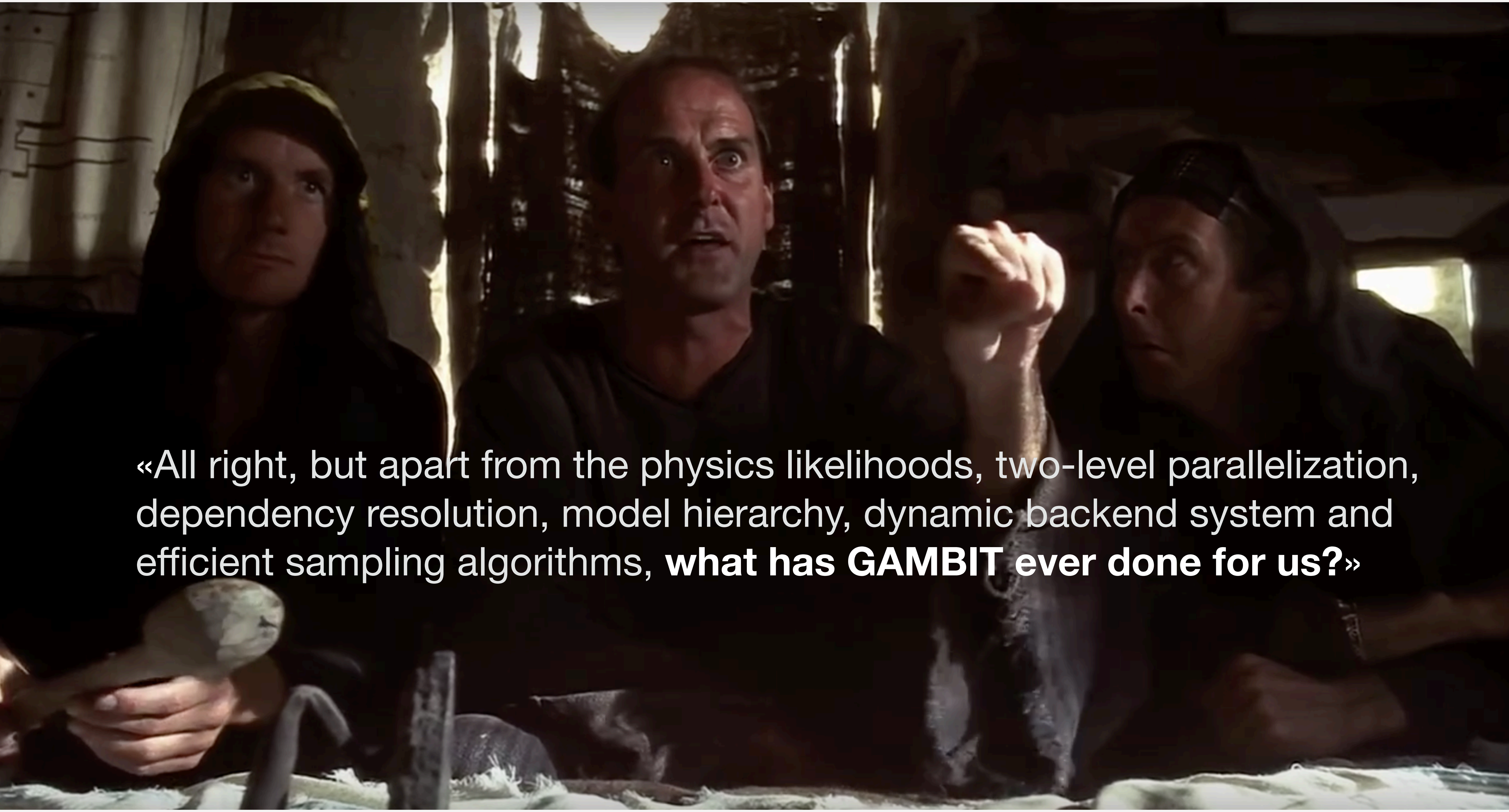


[arXiv:1705.07959]

GAMBIT

Scan illustration



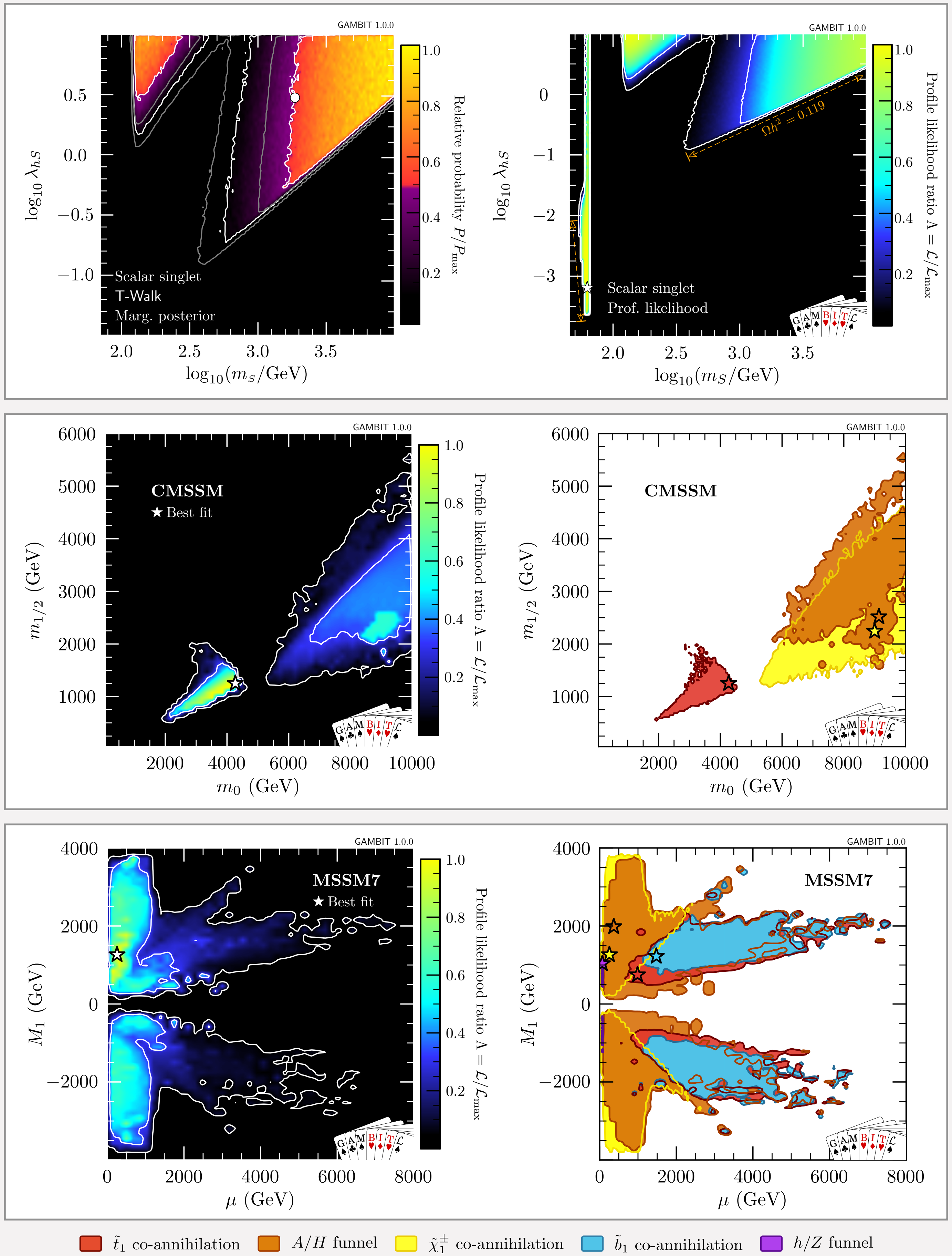

«All right, but apart from the physics likelihoods, two-level parallelization, dependency resolution, model hierarchy, dynamic backend system and efficient sampling algorithms, **what has GAMBIT ever done for us?**»

4. GAMBIT physics results

GAMBIT

First physics results

- **Scalar singlet dark matter**
  arXiv:1705.07931
- **GUT-scale MSSM**
  CMSSM, NUHM1, NUHM2
  arXiv:1705.07935
- **Weak-scale MSSM7**
  arXiv:1705.07917



GUT-scale MSSM results

Parameters and scanning

- Profile likelihood analysis
- Combine samples from scans with different priors and scanners (Diver & MultiNest)
- Additional scans to improve sampling of co-annihilation regions
- In total for all three models: 36 scans, ~280 million viable samples
- Vary 5 nuisance parameters (constrained by gaussian likelihoods

Parameter	Minimum	Maximum	Priors
CMSSM			
m_0	50 GeV	10 TeV	flat, log
$m_{1/2}$	50 GeV	10 TeV	flat, log
A_0	−10 TeV	10 TeV	flat, hybrid
$\tan\beta$	3	70	flat
$\mathrm{sgn}(\mu)$	−	+	binary
NUHM1 – as per CMSSM plus			
m_H	50 GeV	10 TeV	flat, log
NUHM2 – as per CMSSM plus			
m_{H_u}	50 GeV	10 TeV	flat, log
m_{H_d}	50 GeV	10 TeV	flat, log

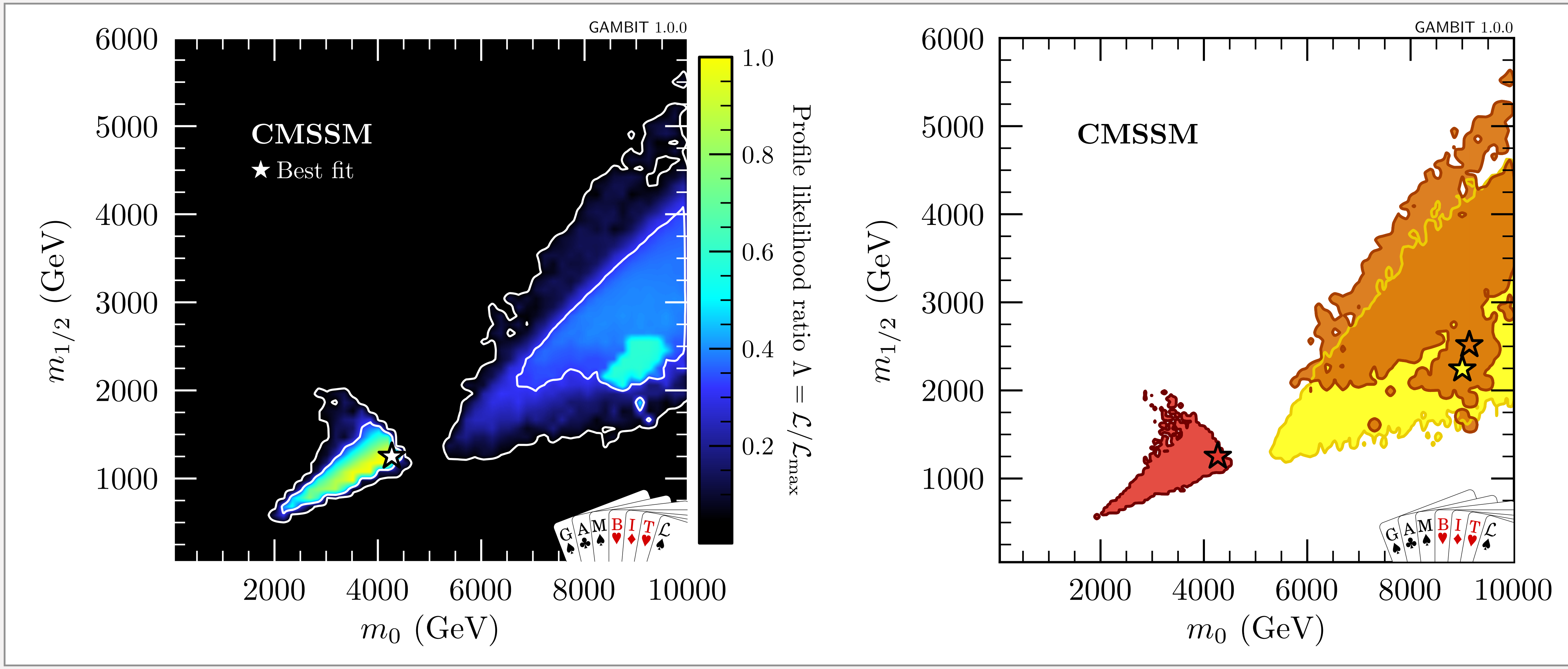
Parameter		Value(±Range)
Varied		
Strong coupling	$\alpha_s^{\overline{MS}}(m_Z)$	0.1185(18)
Top quark pole mass	m_t	173.34(2.28) GeV
Local DM density	ρ_0	0.2–0.8 $\mathrm{GeV\,cm^{-3}}$
Nuclear matrix el. (strange)	σ_s	43(24) MeV
Nuclear matrix el. (up + down)	σ_l	58(27) MeV

Likelihoods

- Nuisance parameter likelihoods
  (SM, local halo model, nuclear matrix elements)
- DM relic density *as upper bound*
- DM Indirect detection
  - Gamma rays: Fermi-LAT
    (dwarf spheriodal galaxies)
  - Neutrinos from DM annihilation in the Sun:
    IceCube79
- DM Direct detection:
  - XENON100 (2012)
  - LUX (2016)
  - Panda-X (2016)
  - PICO (2015)
  - SuperCDMS (2014)
  - SIMPLE (2014)

- Electroweak precision observables
  - W mass
  - muon g-2
- 59 flavour observables
- Higgs mass and signal strengths
- SUSY cross section limits from LEP
- SUSY searches at LHC (simulated)
  - 0 lepton searches (Run I & II, ATLAS & CMS)
  - Stop searches (Run I, ATLAS & CMS)
  - 2 & 3 lepton searches (Run I, ATLAS & CMS)
  - Monojet search (Run I, CMS)

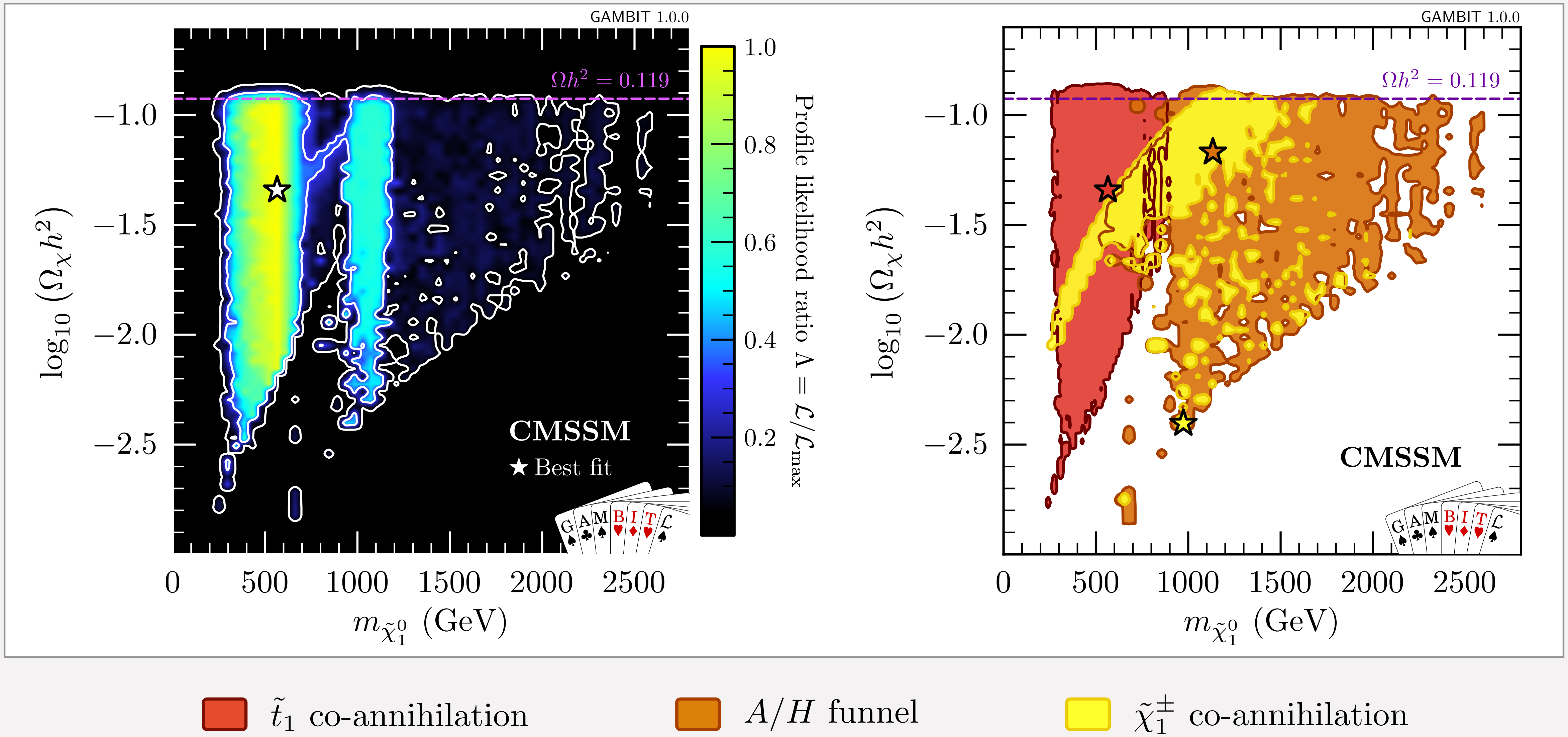
CMSSM



$\tilde{t}_1$ co-annihilation A/H funnel $\tilde{\chi}_1^{\pm}$ co-annihilation

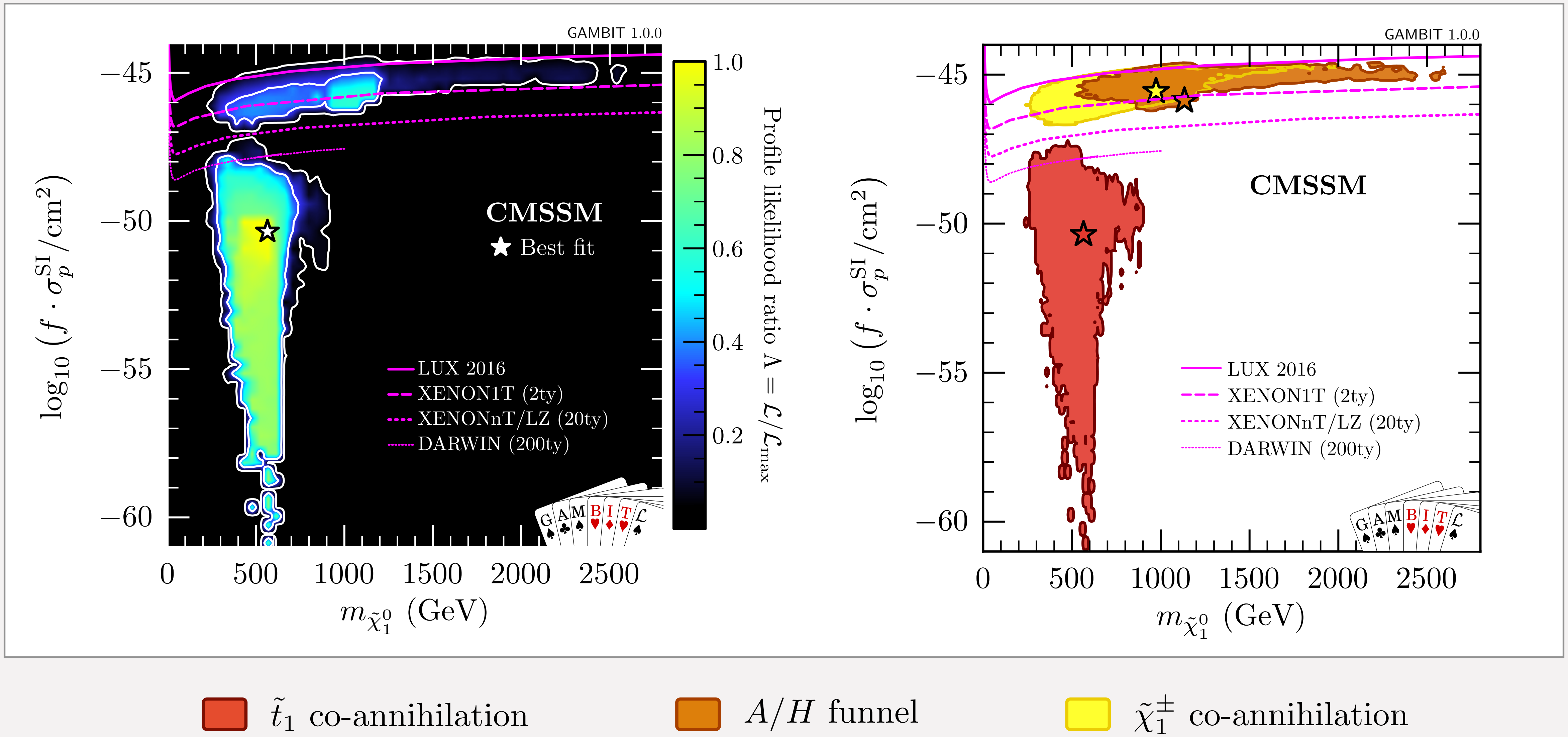
- Three mechanisms to avoid DM overabundance:
  stop co-ann., chargino co-ann., heavy Higgs funnel
- **Stau co-ann. is ruled out at 95% CL** (present at higher CL)
- Overall best fit point in stop co-ann. region (stop/neutralino mass ~600 GeV)

CMSSM



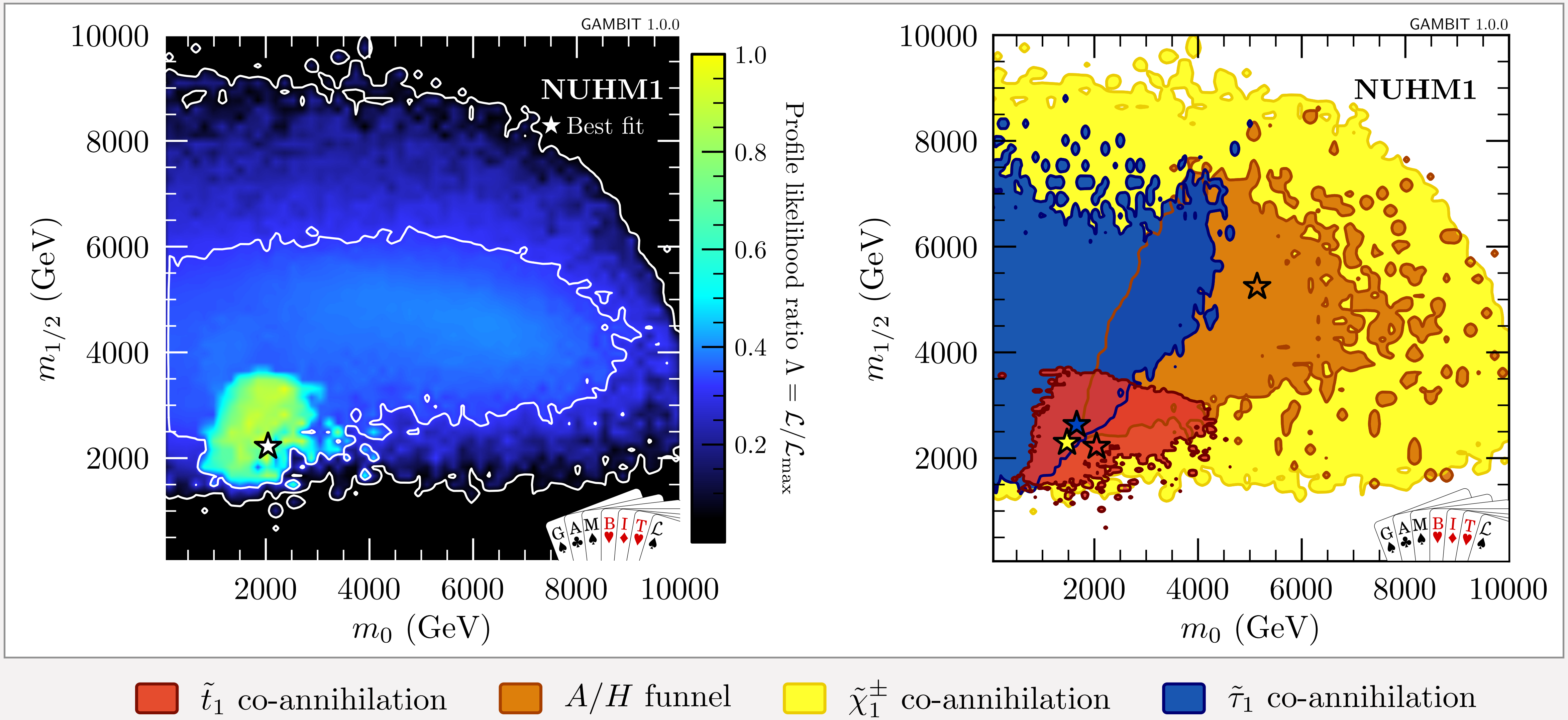
- We impose relic density likelihood as an upper limit
- Higgsino-dominated neutralino saturates relic density for masses ~1 TeV
- Can have combined higgsino co-annihilation and heavy Higgs funnel above 1 TeV

CMSSM



- Cross-section scaled according to predicted relic density
- Chargino co-ann. and Higgs funnel regions can be fully probed by future DD
- Preferred stop co-ann region difficult to probe for DD, ID and LHC
  (Hope to probe low-mass end of the stop-coann region at the LHC)
- Smallest cross-sections due to fine-tuned cancellations in tree-level matrix elements
  (Expect such cancellation to be spoiled by loop corrections)

NUHM I



- Substantially larger allowed regions compared to the CMSSM
- Additional parameter — more freedom to fit Higgs mass
- Stau co-annihilation is back in the 95% CL region
- Overall best fit point in stop co-ann. region (stop & neutralino mass ~1 TeV)

Weak-scale MSSM results

Parameters and likelihoods

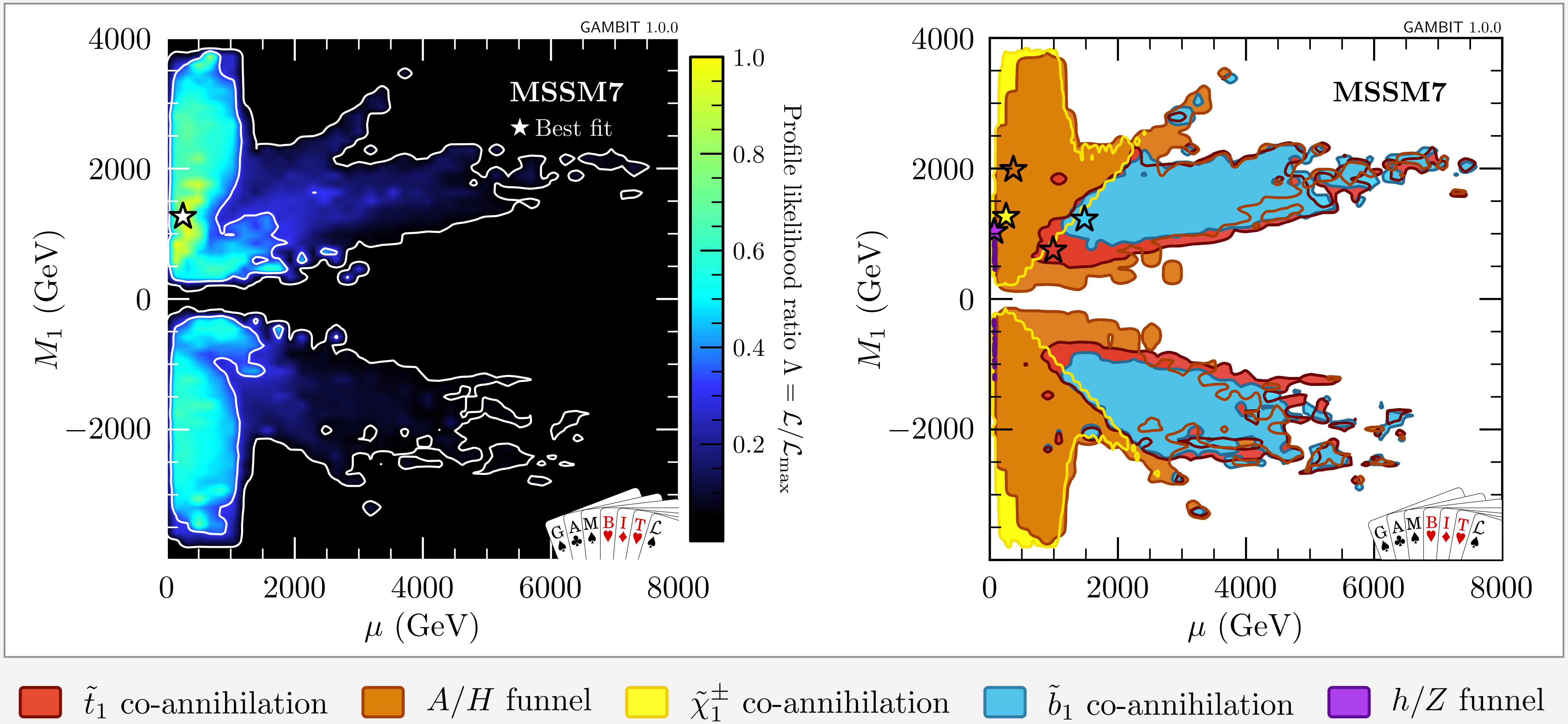
Parameter	Minimum	Maximum	Priors
$A_{u_3}(Q)$	-10 TeV	10 TeV	flat, hybrid
$A_{d_3}(Q)$	-10 TeV	10 TeV	flat, hybrid
$M^2_{H_u}(Q)$	$-(10\,\text{TeV})^2$	$(10\,\text{TeV})^2$	flat, hybrid
$M^2_{H_d}(Q)$	$-(10\,\text{TeV})^2$	$(10\,\text{TeV})^2$	flat, hybrid
$m^2_{\tilde{f}}(Q)$	0	$(10\,\text{TeV})^2$	flat, hybrid
$M_2(Q)$	-10 TeV	10 TeV	split; flat, hybrid
$\tan\beta(m_Z)$	3	70	flat
$\text{sgn}(\mu)$	+		fixed
Q	1 TeV		fixed

- 7 MSSM parameters + 5 nuisance parameters
- Assume GUT-inspired relation on gaugino mass parameters:

$$\frac{3}{5}\cos^2\theta_{\rm W} M_1 = \sin^2\theta_{\rm W} M_2 = \frac{\alpha}{\alpha_{\rm s}} M_3$$

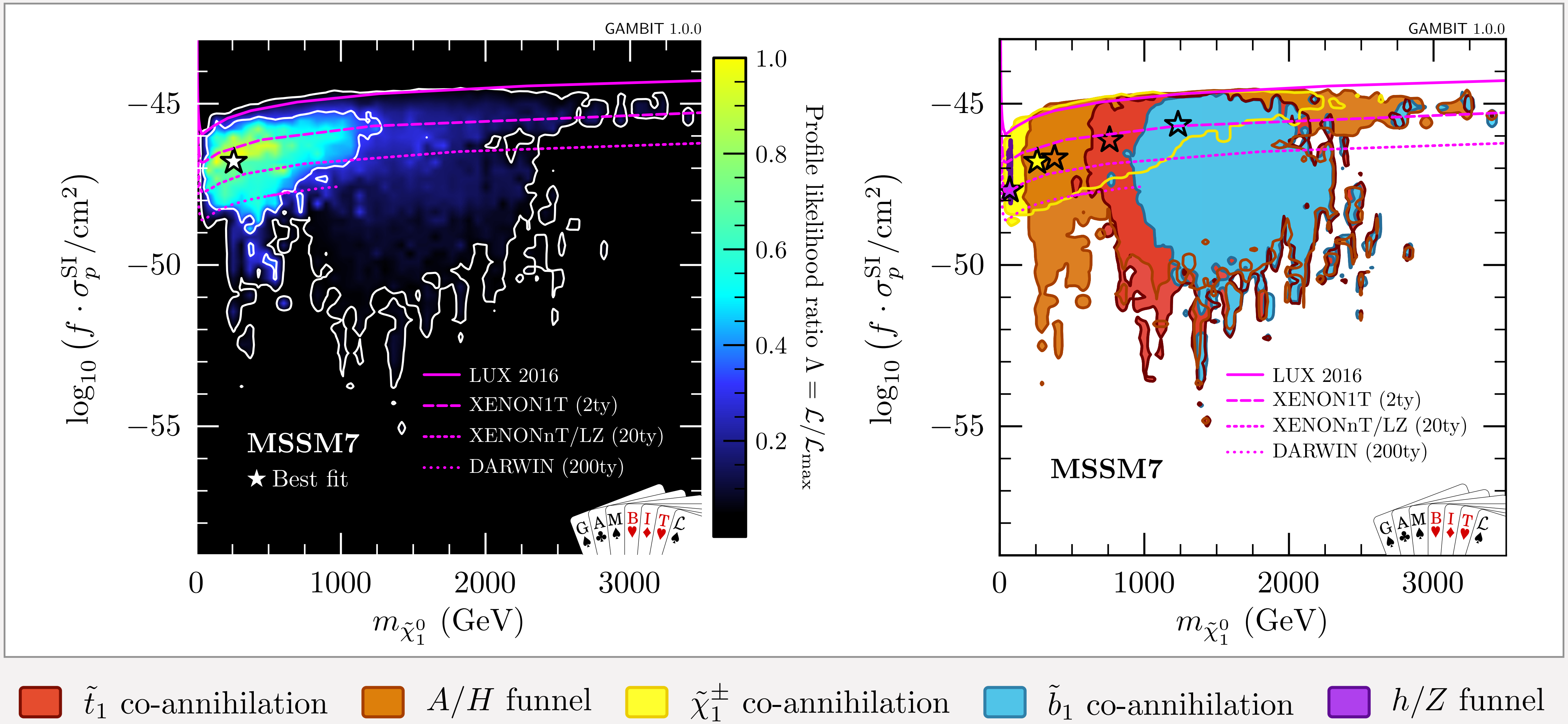
- Same likelihoods as for the GUT-scale models

MSSM 7



- Three neutralino scenarios: higgsino-dominated, higgsino/bino mix, bino-dominated
- Wino-dominated neutralino not possible due to GUT relation $(M_2 \sim 2M_1)$

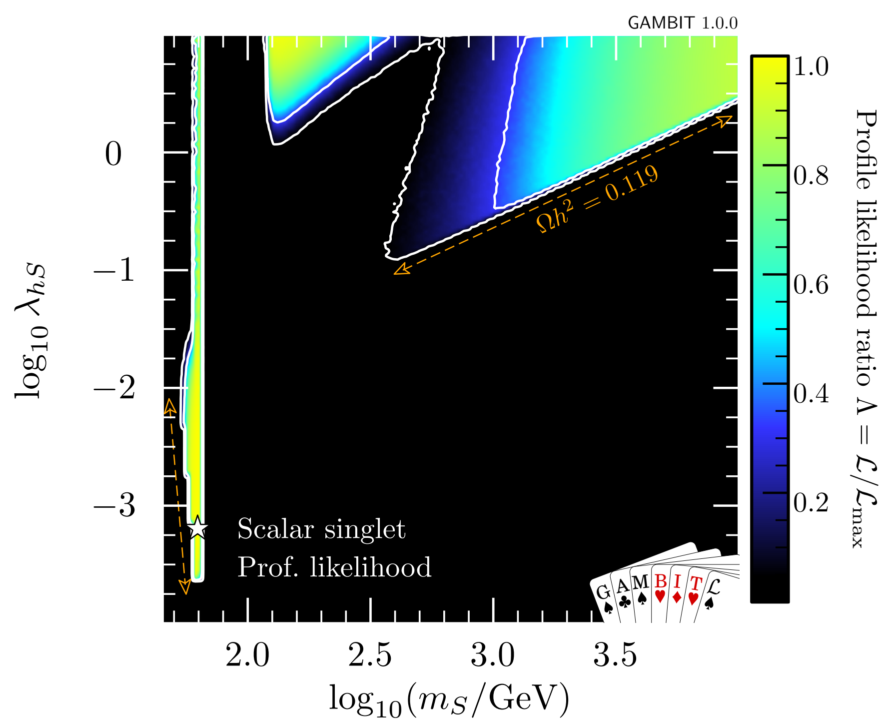
MSSM 7



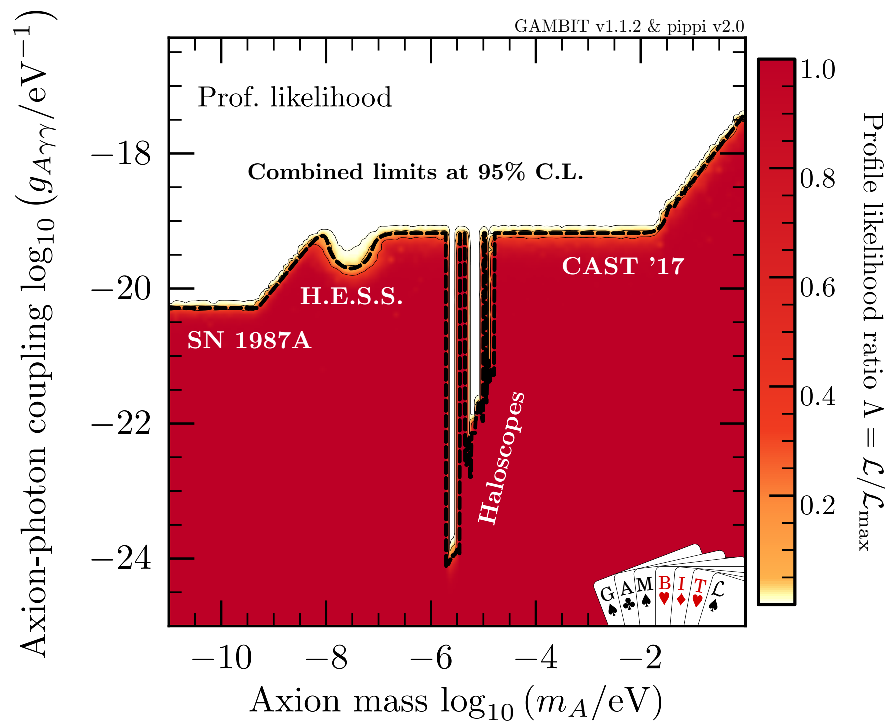
- Best fit point in chargino co-annihilation region (chargino/neutralino mass ~260 GeV)
- Mass difference < 10 GeV (challenging for LHC)
- Under-abundant relic density
- Entire chargino co-ann. and light Higgs funnel regions will be probed by future DD

Other models

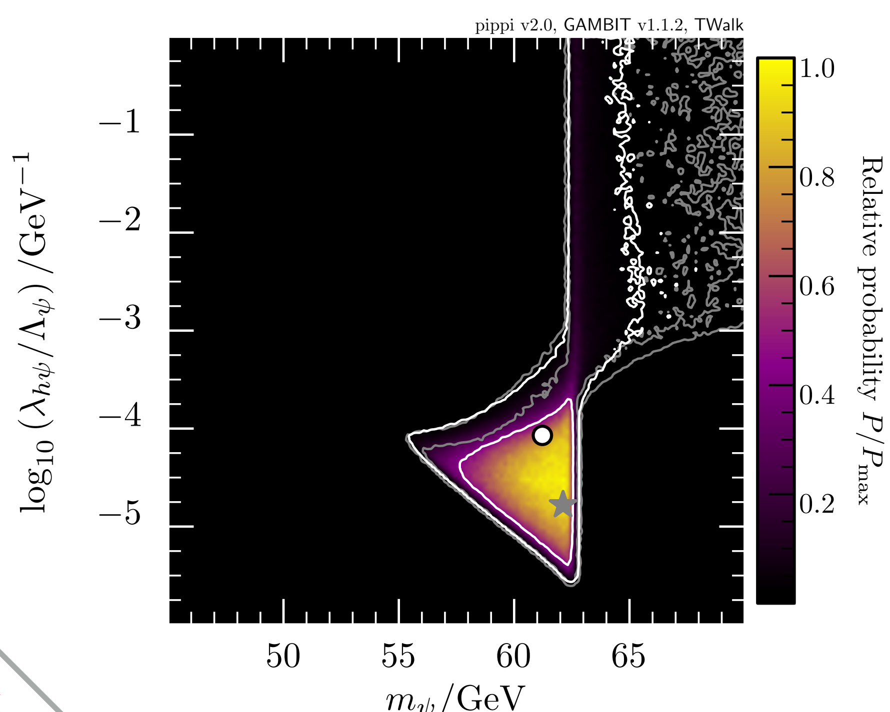
Scalar singlet DM ($\mathbb{Z}_2$ & $\mathbb{Z}_3$)



Axions & ALPs



Higgs Portal DM



- \+ light neutralinos & charginos
- \+ right-handed neutrinos
- \+ 2HDMs
- \+ ...

5. Summary and outlook

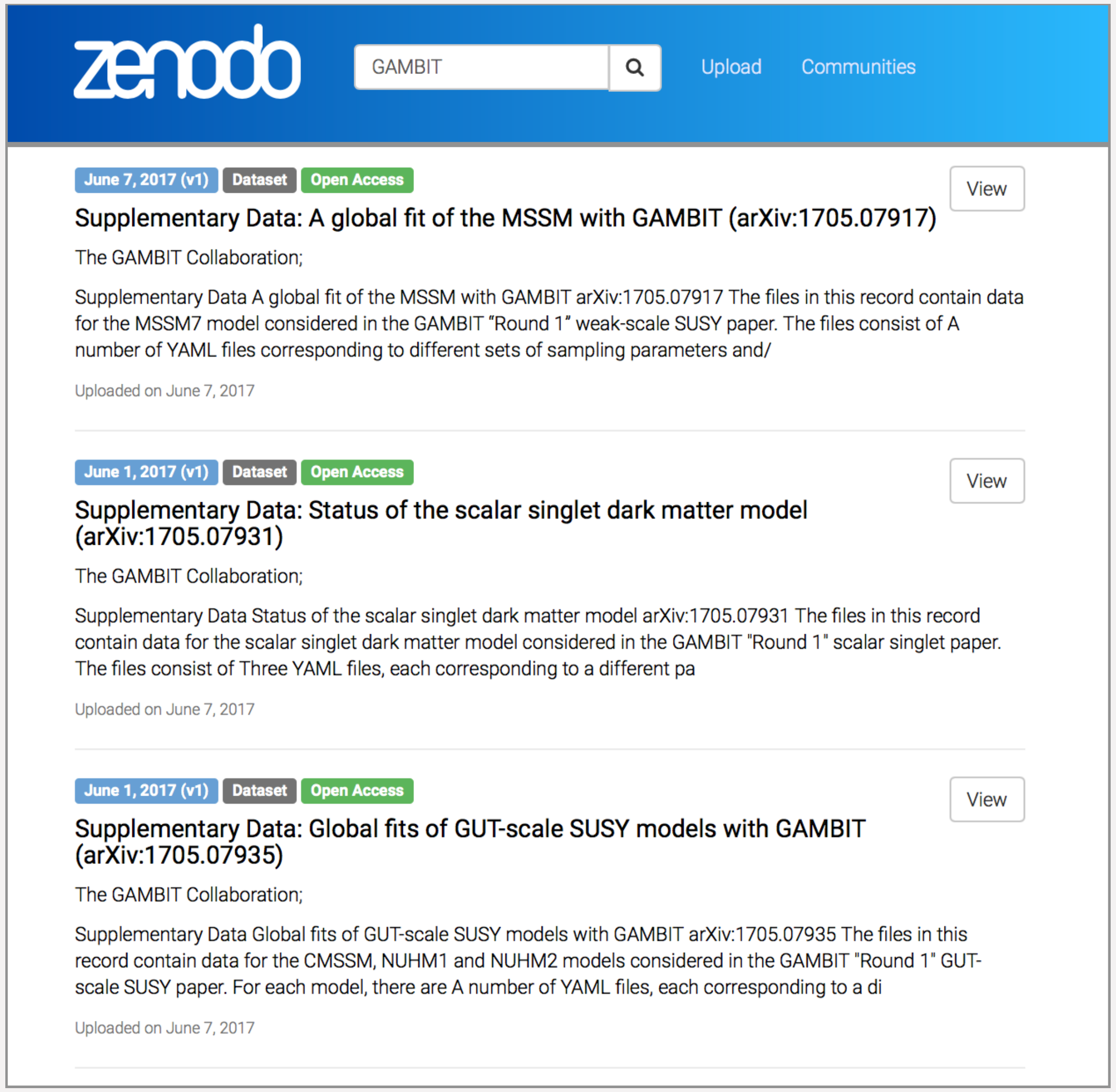
Summary and outlook

- **GAMBIT is public — try it out!**
  - *GAMBIT Community:* a network of users & collaborators
  - If you find a bug, please tell us (preferably via github)
- **Help us extend GAMBIT**
  - Is your code useful for global fits? Why not create a GAMBIT interface?
  - Detailed instructions and examples in the manual and source code
- **First physics results**
  - Singlet DM
  - GUT-scale SUSY
  - Weak-scale MSSM7
- **More results coming soon**
  - Axions, Higgs portal dark matter, light EW-gauginos, right-handed neutrinos…
- **Future plans**
  - More models! More likelihoods!
  - CosmoBit: cosmological models and observables
  - GAMBIT 2.0: Interface with Lagrangian-level tools for automatic code generation

All results publicly available

Results available on zenodo.cern.ch

- Parameter point samples (hdf5 files)
- GAMBIT input files for all scans
- Example plotting routines



Links at gambit.hepforge.org/pubs

Getting started with GAMBIT

Clone git repository from GitHub

- github.com/patscott/gambit_1.1

Download tarballs

- hepforge.org/downloads/gambit

Pre-compilied version with Docker

- `docker run -it jmcornell/gambit`

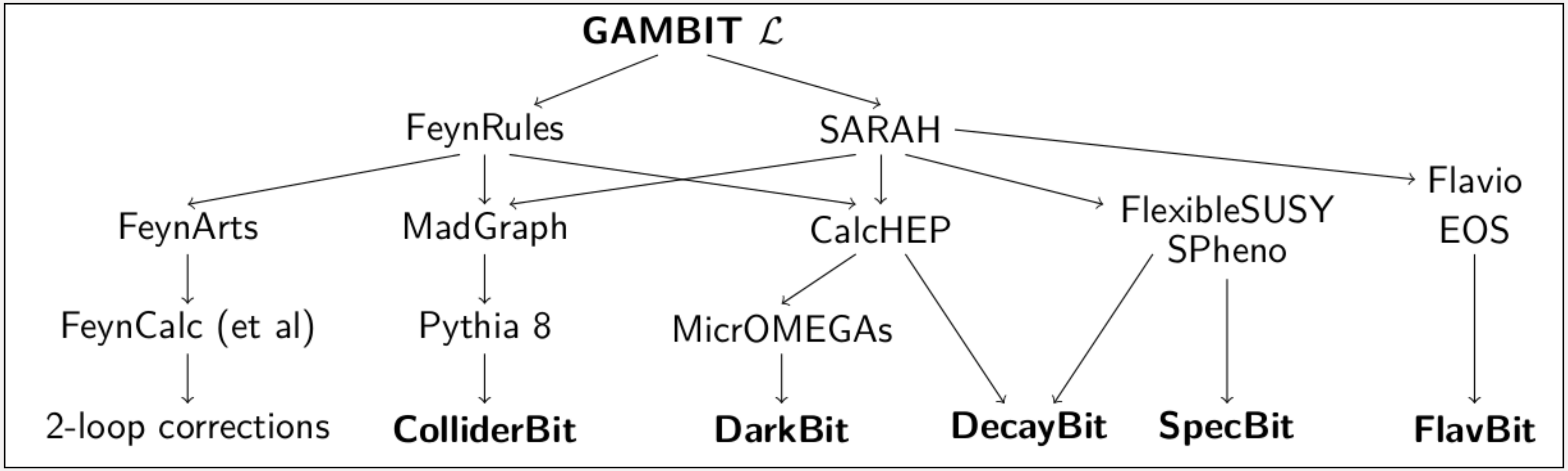
See quick start guide in arXiv:1705.07908

Bonus material

GAMBIT 2

Extension to model building

- GAMBIT Universal Model (GUM) files
- Interface to Lagrangian-level tools
- Code generation for spectra, cross sections, …



Adding a new model to GAMBIT

(From Pat Scott)

1. Add the model to the **model hierarchy**:
    - Choose a model name, and declare any **parent model**
    - Declare the model's parameters
    - Declare any **translation function** to the parent model

```
#define MODEL NUHM1
#define PARENT NUHM2
  START_MODEL
  DEFINEPARS(M0,M12,mH,A0,TanBeta,SignMu)
  INTERPRET_AS_PARENT_FUNCTION(NUHM1_to_NUHM2)
#undef PARENT
#undef MODEL
```

2. Write the translation function as a standard C++ function:

```
void MODEL_NAMESPACE::NUHM1_to_NUHM2 (const ModelParameters &myP, ModelParameters &targetP)
{
   // Set M0, M12, A0, TanBeta and SignMu in the NUHM2 to the same values as in the NUHM1
   targetP.setValues(myP,false);
   // Set the values of mHu and mHd in the NUHM2 to the value of mH in the NUHM1
   targetP.setValue("mHu", myP["mH"]);
   targetP.setValue("mHd", myP["mH"]);
}
```

3. If needed, declare that existing module functions work with the new model, or add new functions that do.

Adding a new module function is easy:

1. Declare the function to GAMBIT in a module's **rollcall header**
    - Choose a capability
    - Declare any **backend requirements**
    - Declare any **dependencies**
    - Declare any specific **allowed models**
    - other more advanced declarations also available

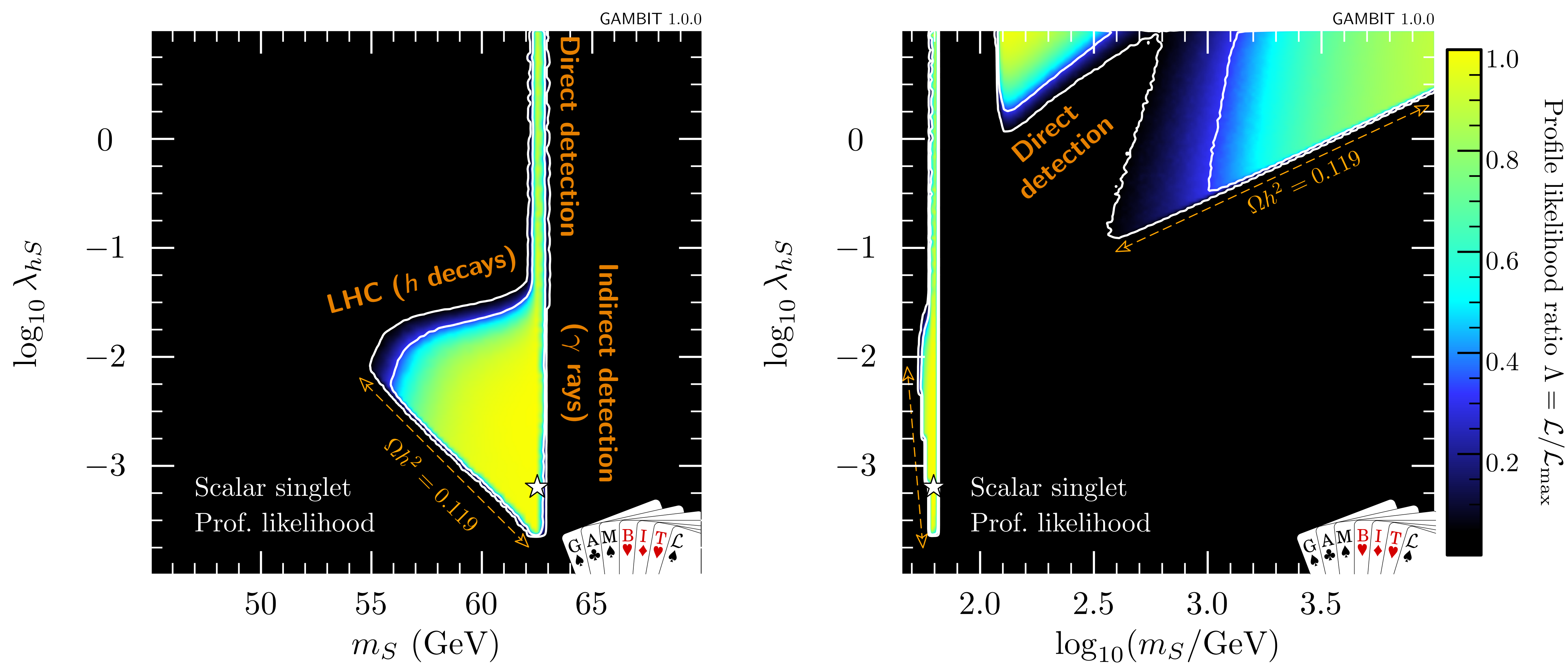
```
#define MODULE FlavBit                             // A tasty GAMBIT module.
START_MODULE

  #define CAPABILITY Rmu                           // Observable: BR(K->mu nu)/BR(pi->mu nu)
  START_CAPABILITY
    #define FUNCTION SI_Rmu                        // Name of a function that can compute Rmu
    START_FUNCTION(double)                         // Function computes a double precision result
    BACKEND_REQ(Kmunu_pimunu, (my_tag), double, (const parameters*)) // Needs function from a backend
    BACKEND_OPTION( (SuperIso, 3.6), (my_tag) )                      // Backend must be SuperIso 3.6
    DEPENDENCY(SuperIso_modelinfo, parameters)     // Needs another function to calculate SuperIso info
    ALLOW_MODELS(MSSM63atQ, MSSM63atMGUT)          // Works with weak/GUT-scale MSSM and descendents
    #undef FUNCTION
  #undef CAPABILITY
```

2. Write the function as a standard C++ function
   (one argument: the result)

Scalar singlet DM (2+13 parameters)

(From Pat Scott)



Simplest BSM example: $\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \ldots$

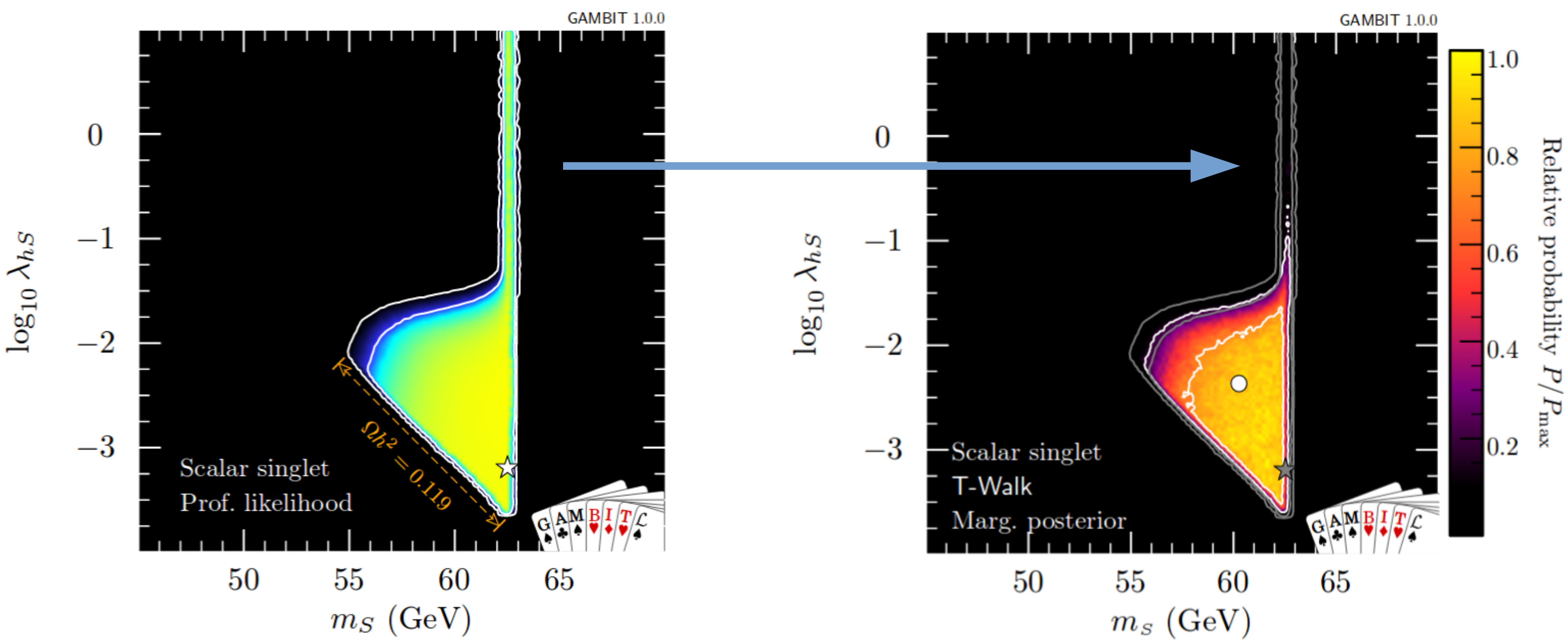
All dark matter signals consistently scaled for predicted abundance

Scalar singlet DM (2+13 parameters)

(from Felix Kahlhoefer)

Assessing fine-tuning with Bayesian scans

- In case of a non-observation, experimental data will push WIMP models into more and more finely tuned regions of parameter space
- How do we assess whether WIMPs remain viable in spite of such tuning?
- Possible answer: Penalise fine-tuning with Bayesian statistics



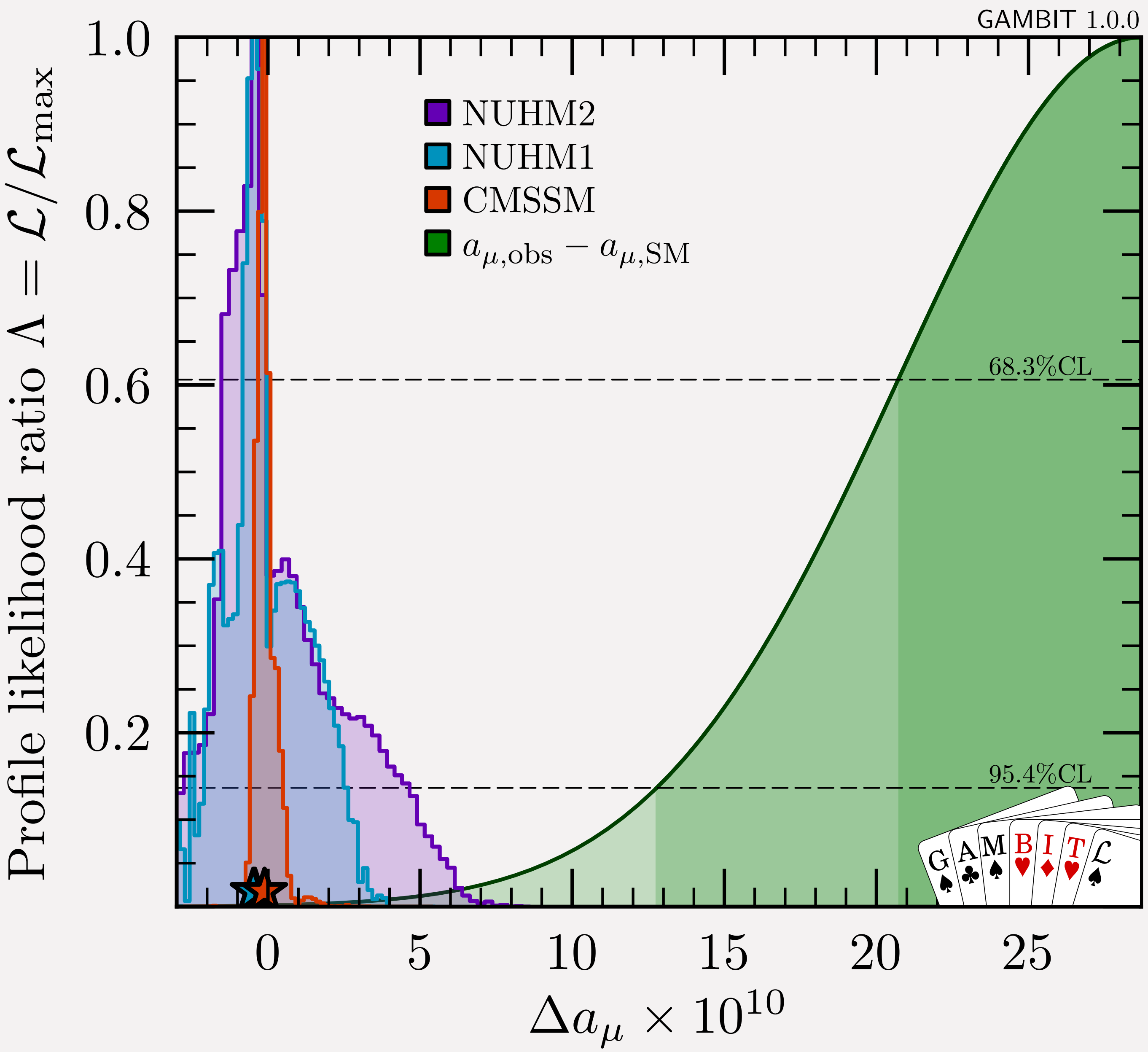
SUSY results

Definition of coloured regions

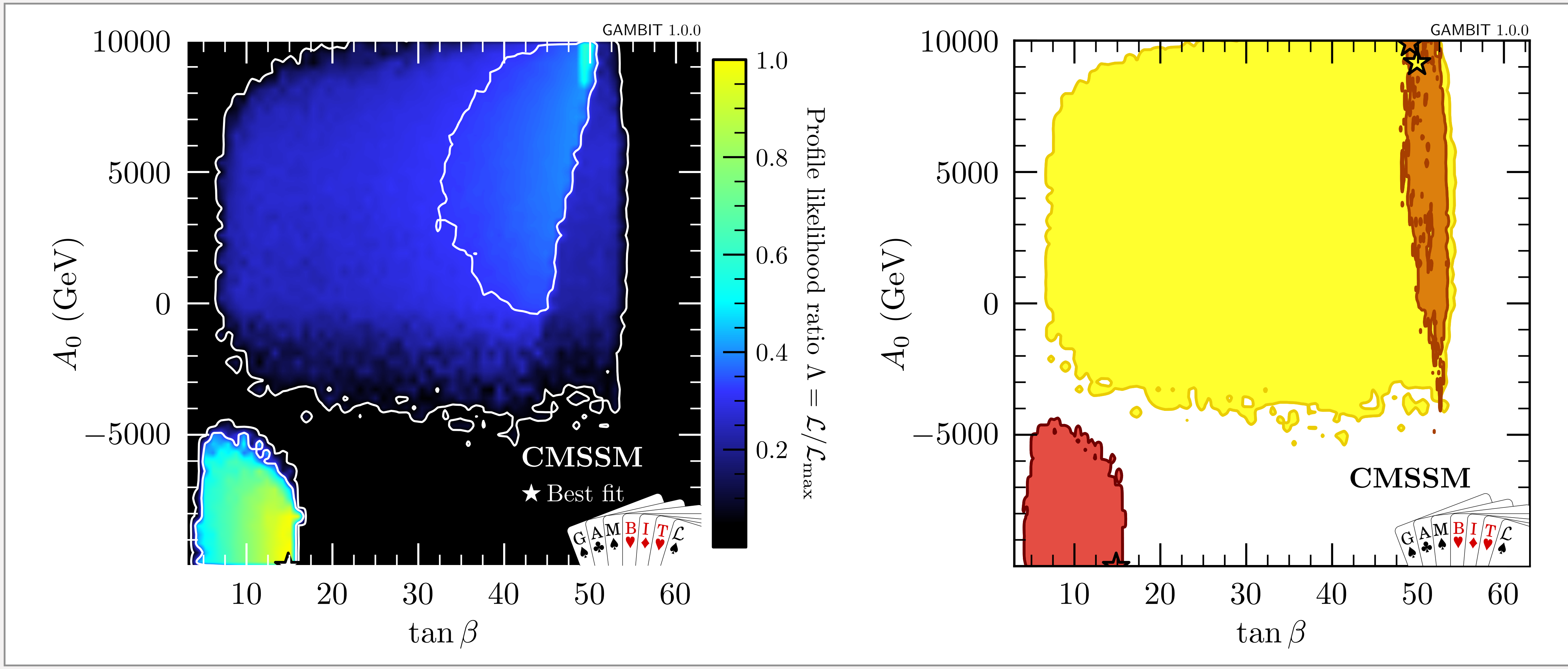
- stau co-annihilation: $m_{\tilde{\tau}_1} \leq 1.2\, m_{\tilde{\chi}_1^0}$,
- stop co-annihilation: $m_{\tilde{t}_1} \leq 1.2\, m_{\tilde{\chi}_1^0}$,
- chargino co-annihilation: $\tilde{\chi}_1^0 \geq 50\%$ Higgsino,
- A/H-funnel: $1.6\, m_{\tilde{\chi}_1^0} \leq m_{\text{heavy}} \leq 2.4\, m_{\tilde{\chi}_1^0}$,

GUT-scale results

Muon g-2



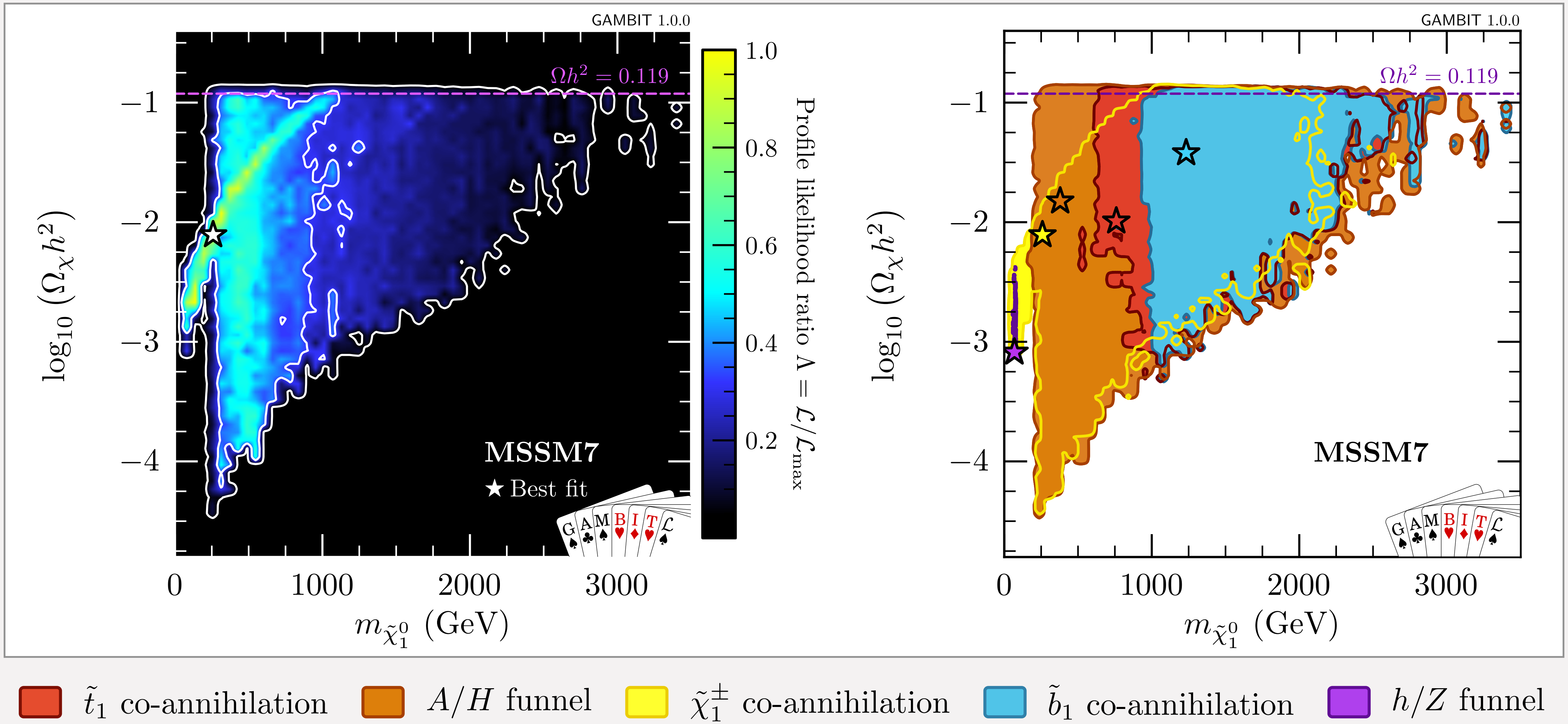
CMSSM



$\tilde{t}_1$ co-annihilation A/H funnel $\tilde{\chi}_1^{\pm}$ co-annihilation

- Stop co-ann. region at large, negative trilinear coupling
- Small impact of (simple) check for charge- and colour-breaking minima

MSSM 7



- Best fit point in chargino co-annihilation region (chargino/neutralino mass ~260 GeV)
- Mass difference < 10 GeV (challenging for LHC)
- Under-abundant relic density

MSSM7 results

Chargino—netrualino mass plane

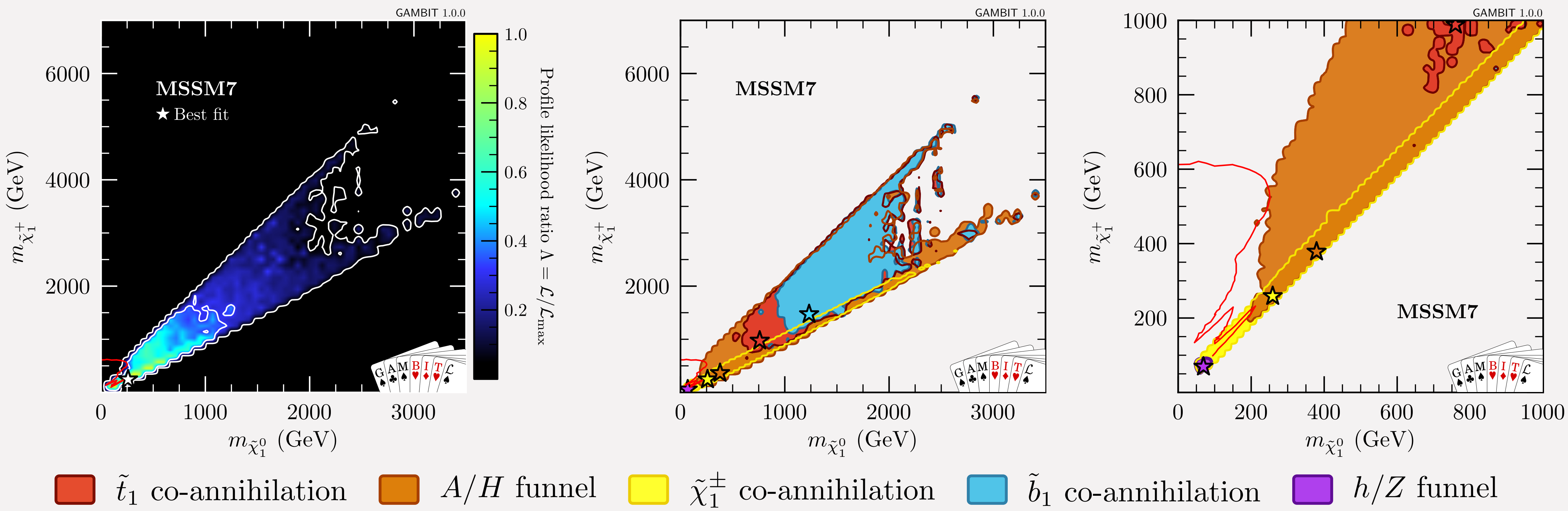


Fig. 8: *Left:* Profile likelihood in the $\tilde{\chi}_1^{\pm} - \tilde{\chi}_1^0$ mass plane. *Centre:* Sub-regions within the 95% CL profile likelihood region, coloured according to mechanisms by which the relic density constraint is satisfied. The regions shown correspond to neutralino co-annihilation with charginos, stops or sbottoms, and resonant annihilation through the light or heavy Higgs funnels. Superimposed in red is the latest CMS Run II simplified model limit for $\tilde{\chi}_1^{\pm}\tilde{\chi}_1^0$ production and decay with decoupled sleptons [210]. This limit should be interpreted with caution (see main text for details). *Right:* The same information as the central plot, but zoomed into the low-mass region. Note that, although the CMS limit appears to have excluded part of the chargino co-annihilation region, this is a binning effect. One should instead refer to the plot of the $\tilde{\chi}_1^{\pm} - \tilde{\chi}_1^0$ mass difference in Fig. 7, which provides finer resolution on the mass difference in this region.

MSSM7 results

Stop/sbottom—netrualino mass planes

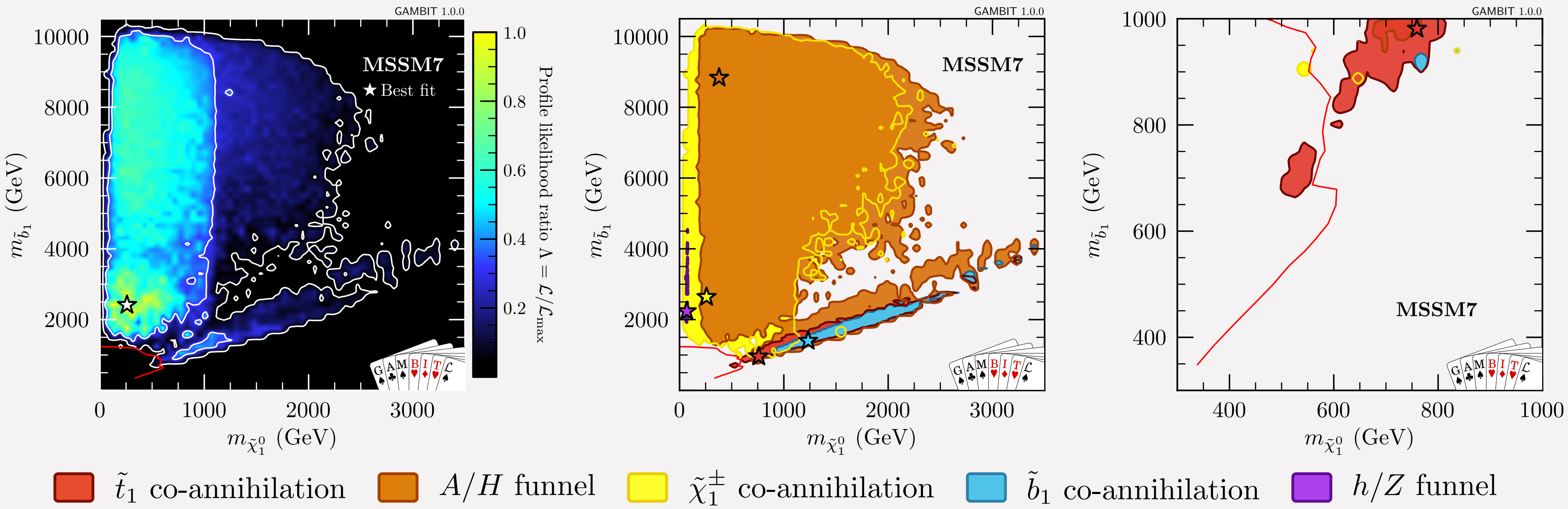


Fig. 10: *Left:* The profile likelihood ratio in the $\tilde{b}_1 - \tilde{\chi}_1^0$ mass plane. *Centre:* Colour-coding shows mechanism(s) that allow models within the 95% CL region to avoid exceeding the observed relic density of DM. The regions shown correspond to neutralino co-annihilation with charginos, stops or sbottoms, and resonant annihilation through the light or heavy Higgs funnels. *Right:* The same information as the central plot, zoomed into the low-mass region.

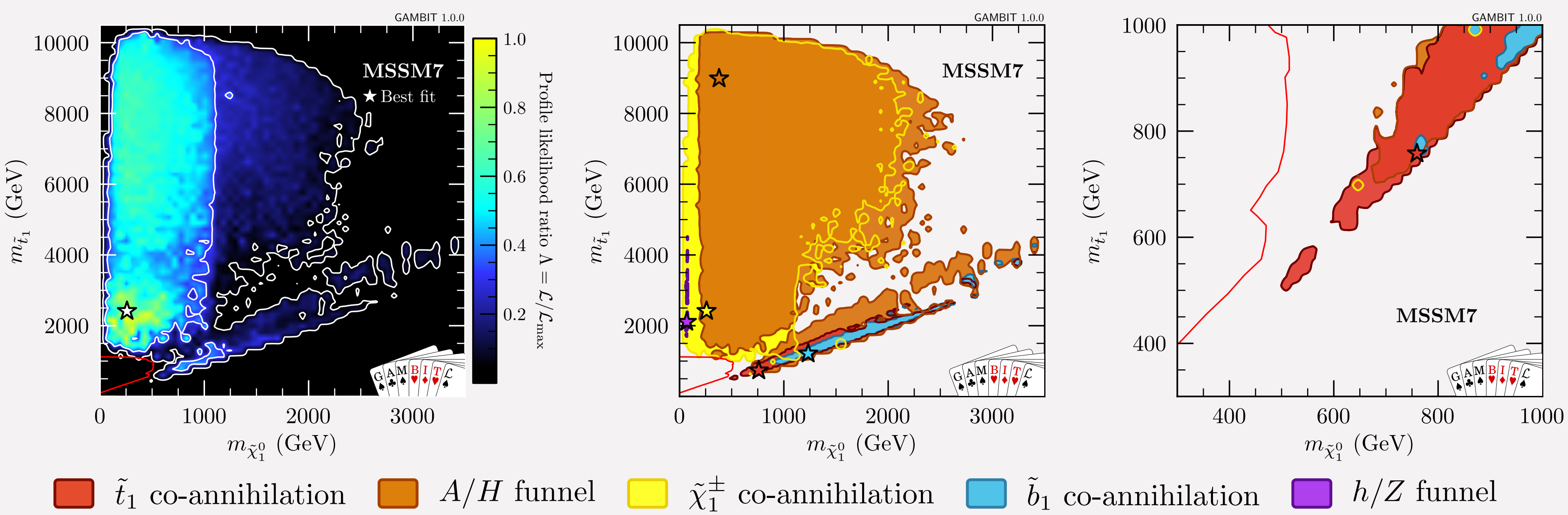


Fig. 11: *Left:* The profile likelihood ratio in the $\tilde{t}_1 - \tilde{\chi}_1^0$ mass plane. *Centre:* Colour-coding shows mechanism(s) that allow models within the 95% CL region to avoid exceeding the observed relic density of DM. The regions shown correspond to neutralino co-annihilation with charginos, stops or sbottoms, and resonant annihilation through the light or heavy Higgs funnels. Superimposed in red is the latest CMS Run II simplified model limit for stop pair production [211]. *Right:* The same information as the central plot, zoomed into the low-mass region.

MSSM7 results

Muon g-2