An introduction to GAMBIT

or

the Future of Global Fits to New Physics

Anders Kvellestad, University of Oslo

- Lots of models for TeV scale New Physics

SUSY

4th generation

UED

Composite Higgs

- Lots of models for TeV scale New Physics

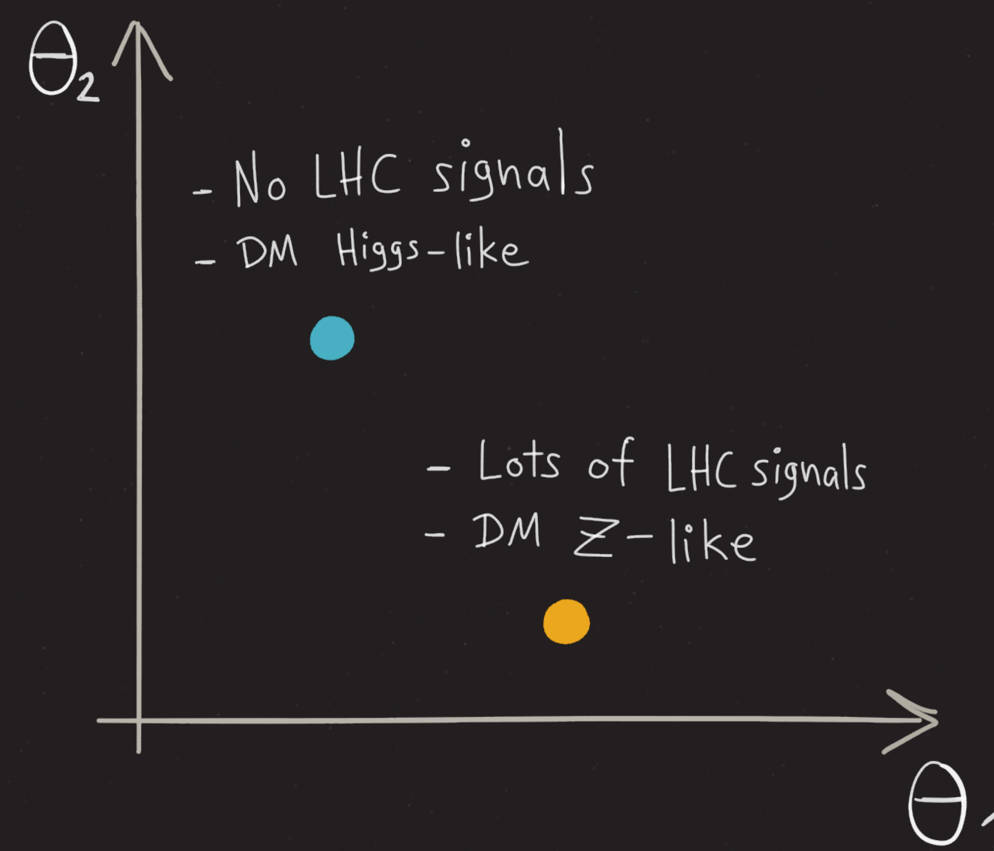
SUSY

4th generation

UED

Composite Higgs

- For each model, a parameter space of varying phenomenology



What New Physics scenario is preferred?

What New Physics scenario is preferred?

Compare to data!

What New Physics scenario is preferred?

Compare to data!

...But how?

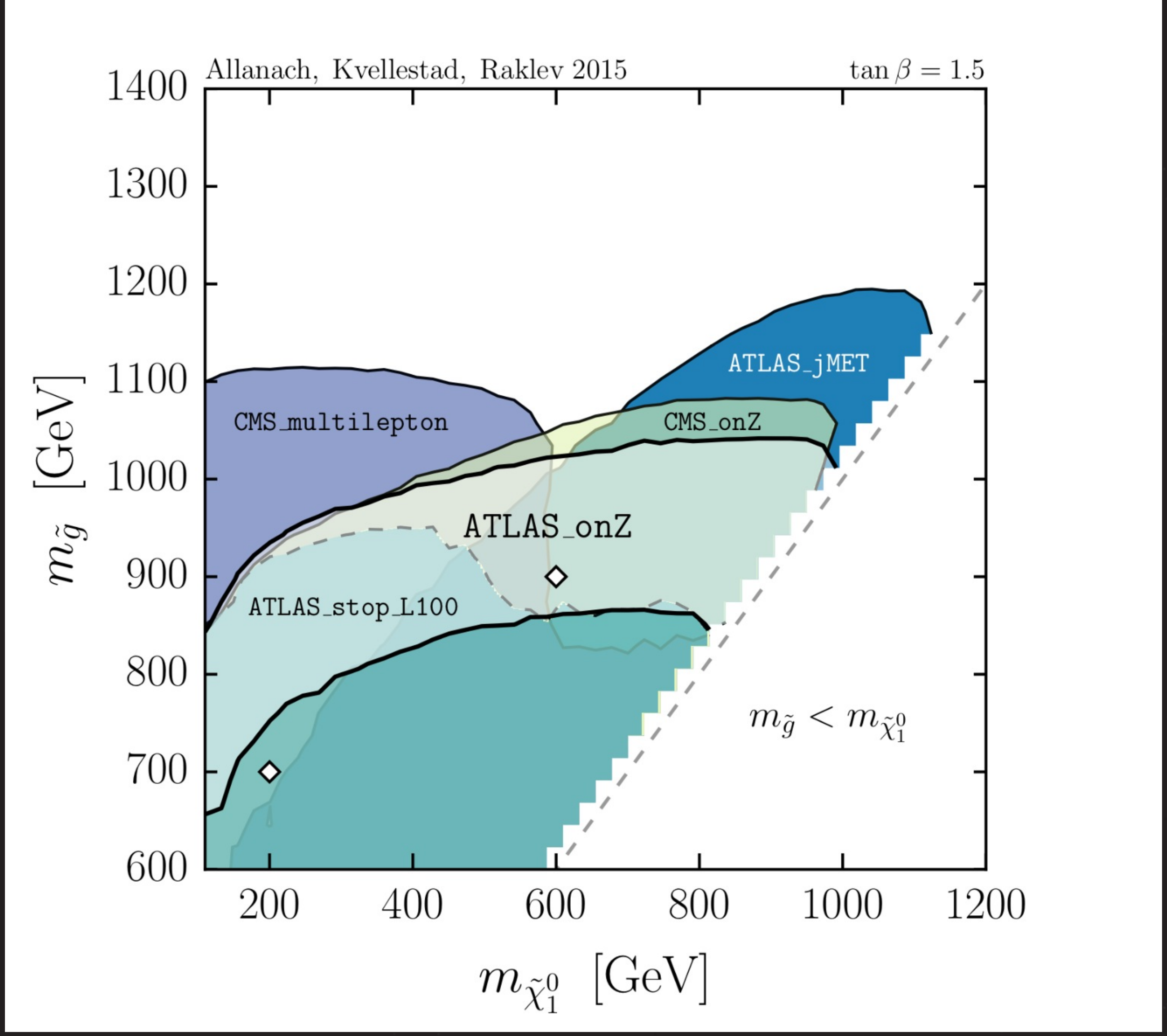
Only a couple of parameters:

Compare preferred/excluded regions for different analyses

Only a couple of parameters:

Compare preferred/excluded regions for different analyses

Allanach, Kvellestad, Raklev: 1504.02752



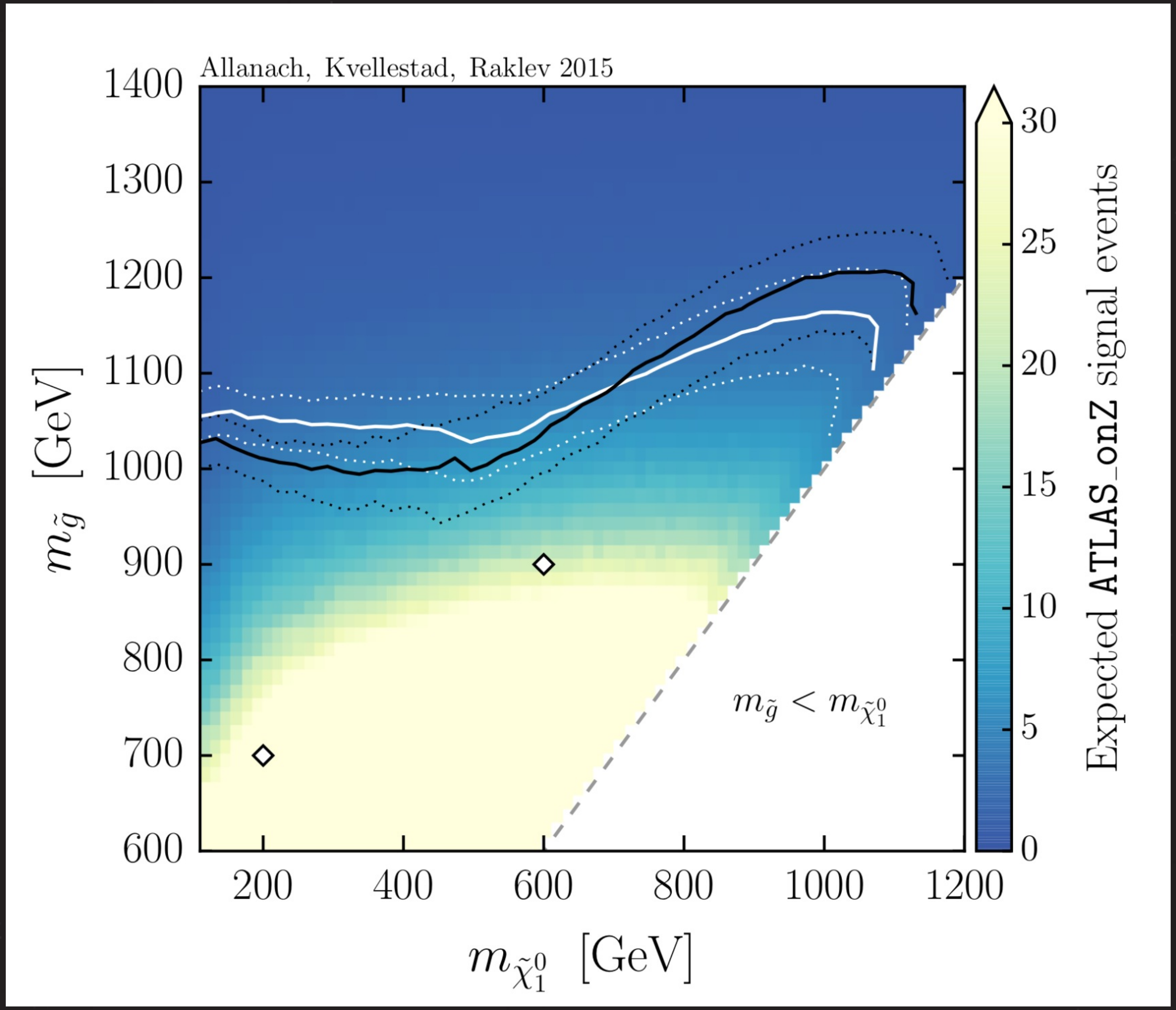
If many different searches:

Combine searches in a total likelihood function

If many different searches:

Combine searches in a total likelihood function

Allanach, Kvellestad, Raklev: 1504.02752



Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

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

Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

- Explore likelihood across entire parameter space (smart sampling)

Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

- Explore likelihood across entire parameter space (smart sampling)

- Interpretation: Bayesian/ frequentist

Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

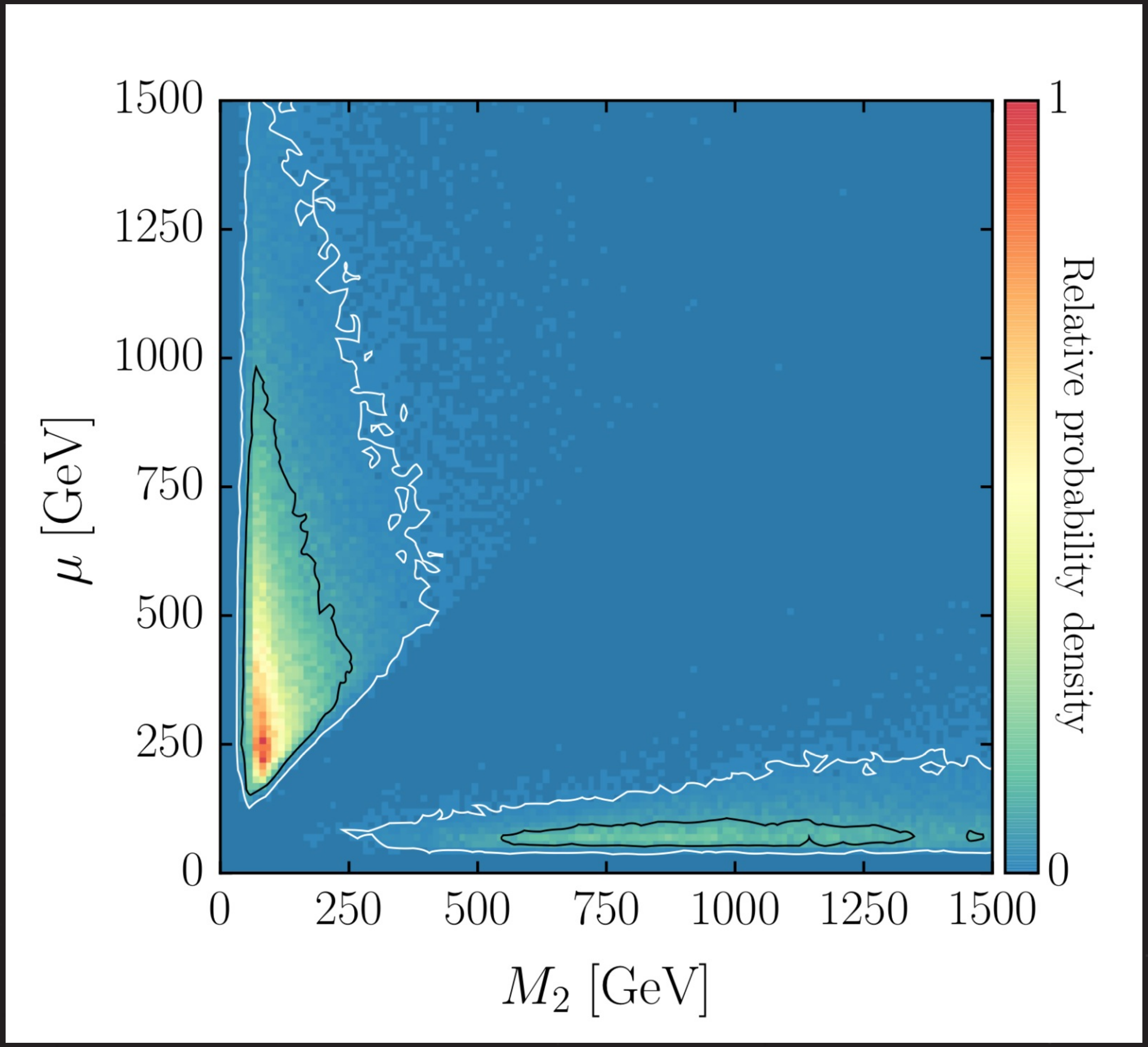
- Explore likelihood across entire parameter space (smart sampling)
- Interpretation: Bayesian/ frequentist
- Project down to 1 or 2 parameters (profile/marginalise)

Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

- Explore likelihood across entire parameter space (smart sampling)
- Interpretation: Bayesian/ frequentist
- Project down to 1 or 2 parameters (profile/marginalise)

Bomark, Kvellestad, Lola, Osland, Raklev: 1410.0921



Global fit goals:

Global fit goals:

- Assume a theory is true:
  What parameter values are preferred?

→ Parameter estimation

Global fit goals:

- Assume a theory is true:
  What parameter values are preferred? → Parameter estimation

- Given multiple theories:
  Which is in best agreement with data? → Model comparison

Current challenges:

Current challenges:

- Many experimental searches (almost) neglected

Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories

Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories
- ...and a limited selection of theory tools

Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories
- ...and a limited selection of theory tools
- ...and a limited set of sampling algorithms

Global And Modular BSM Inference Tool

GAMBIT

- A collaboration of ~30 theorists and experimentalists

  P. Athron, C. Balázs, T. Bringmann, A. Buckley, M. Chrzaszcz, J. Conrad, J. Cornell, L. Dal, J. Edsjö, B. Farmer, L. Hsu, P. Jackson, A. Krislock, A. Kvellestad, F. Mahmoudi, G. Martinez, M. Pato, A. Putze, A. Raklev, C. Rogan, A. Saavedra, C. Savage, P. Scott, N. Serra, C. Weniger, M. White

- Members from several major particle and astroparticle experiments

  Fermi-LAT, ATLAS, CTA, HESS, LHCb, IceCube, AMS-02, CDMS, DM-ICE, XENON, DARWIN

GAMBIT

- Main design principles: modularity and flexibilty

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)

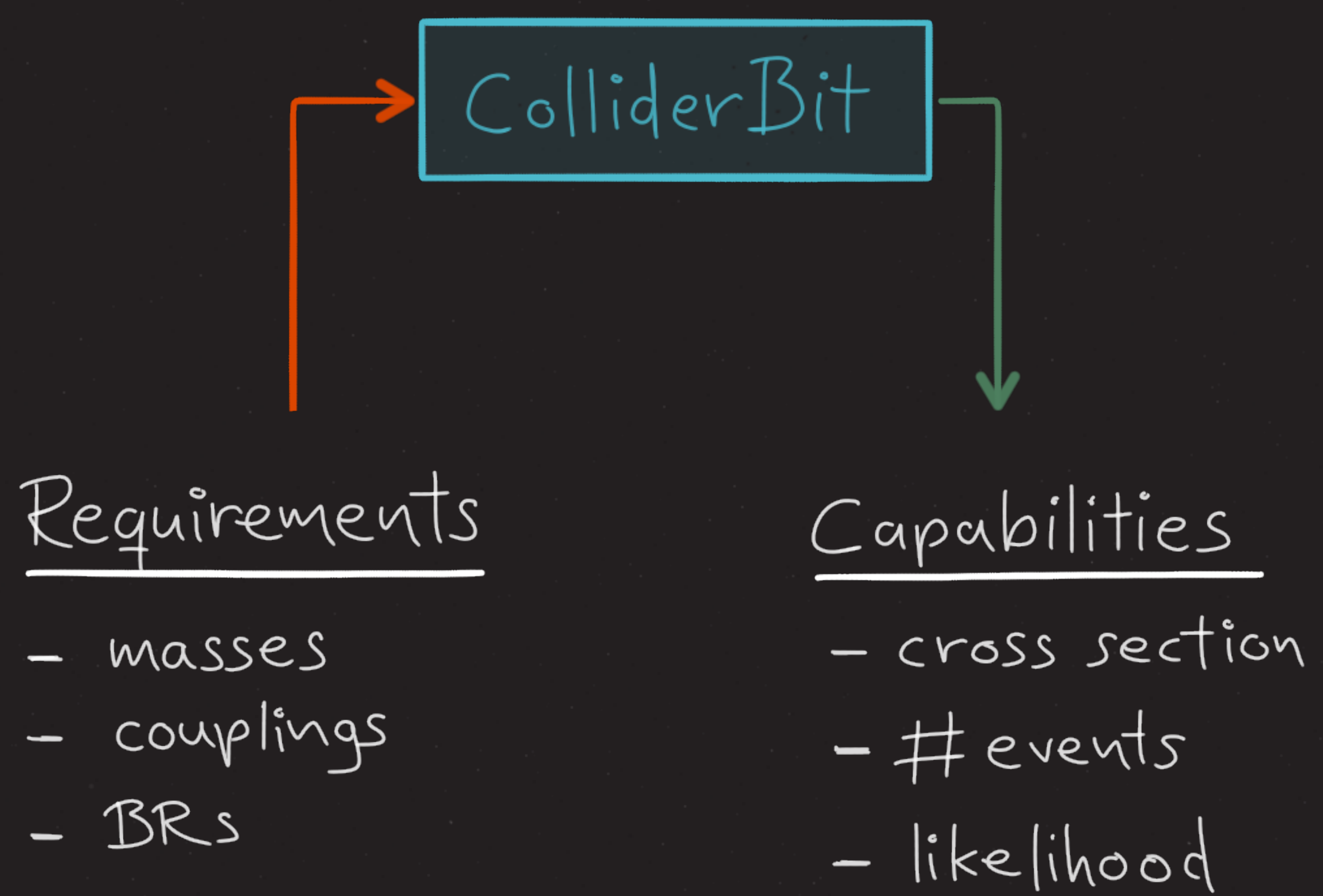
GAMBIT

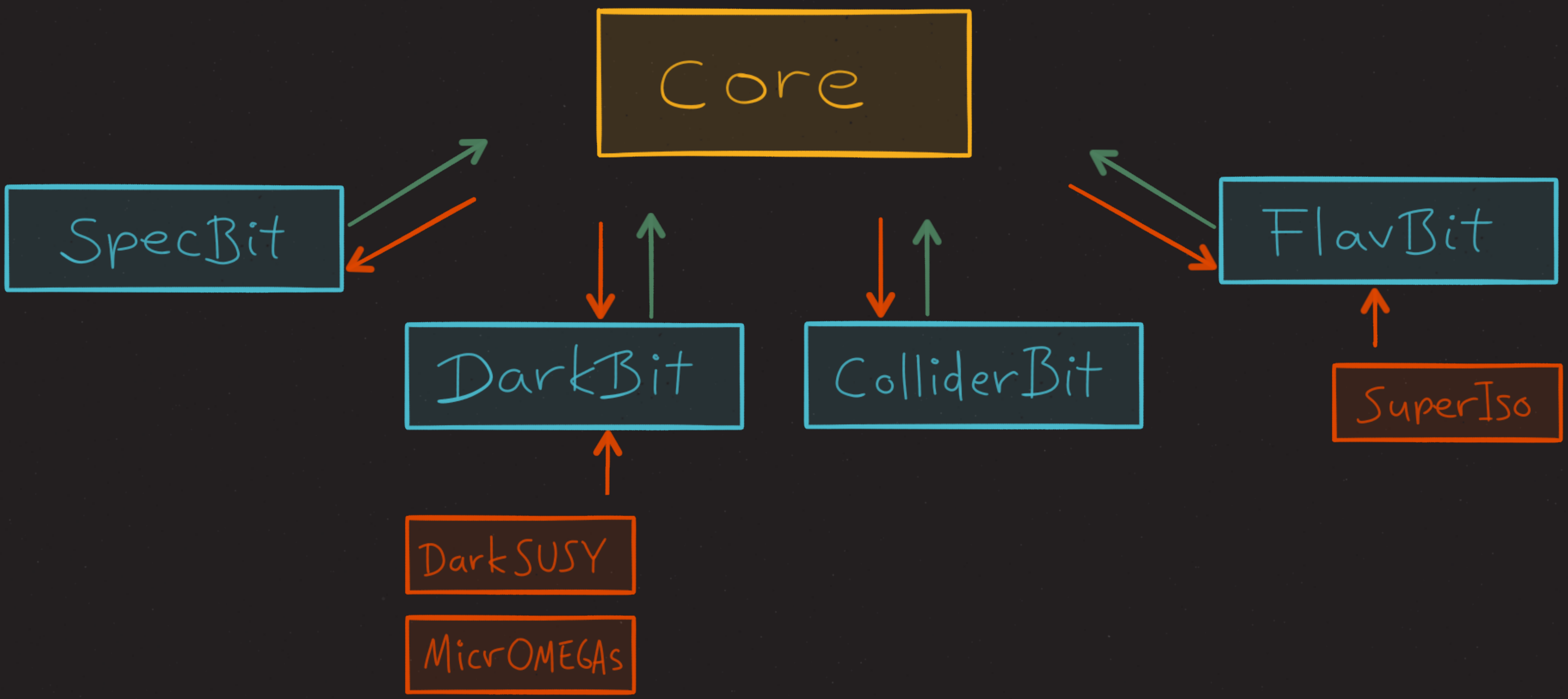
- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins


Core
ScannerBit
SpecBit
DecayBit
ColliderBit
DarkBit
FlavBit
EWPOBit
?

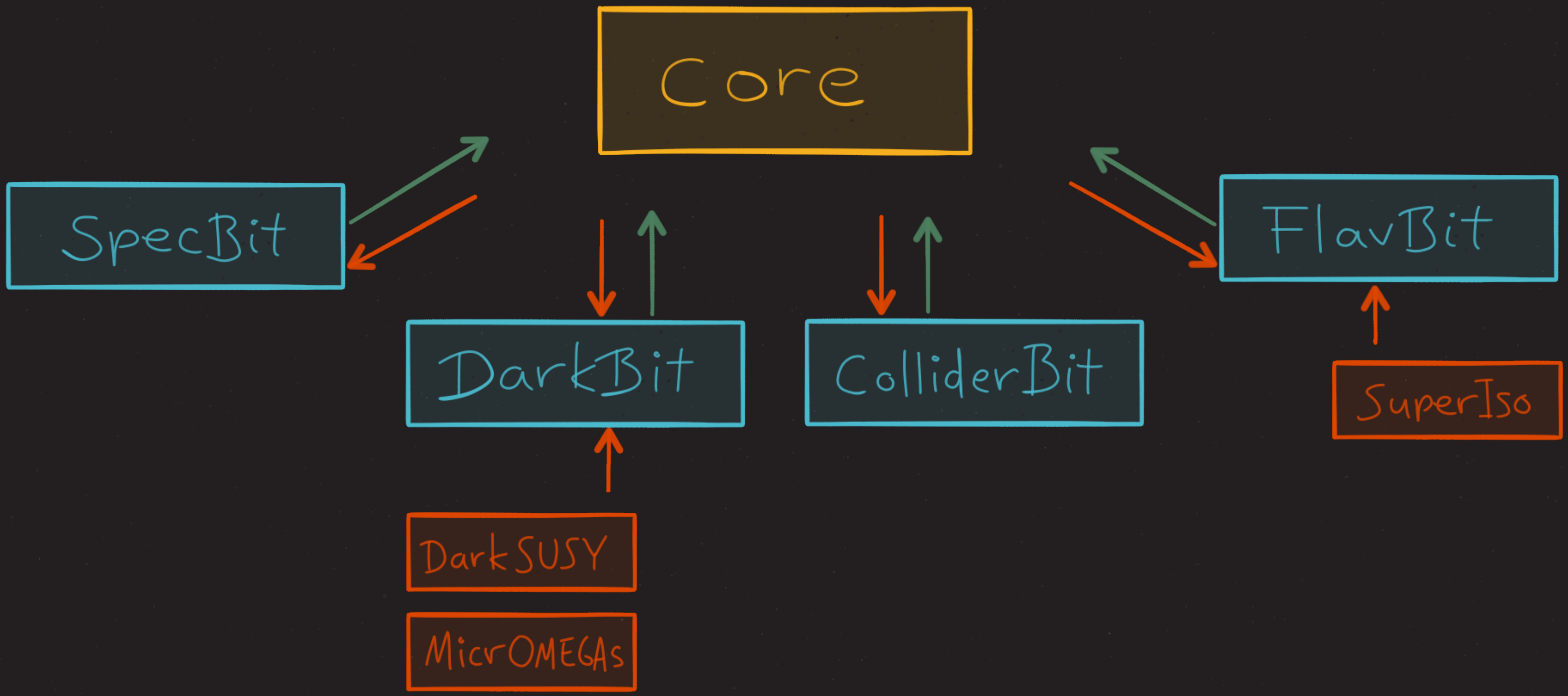

Core
ScannerBit
SpecBit
DecayBit
ColliderBit
DarkBit
FlavBit
EWPOBit
?
FeynHiggs
SuperIso
DarkSUSY
?

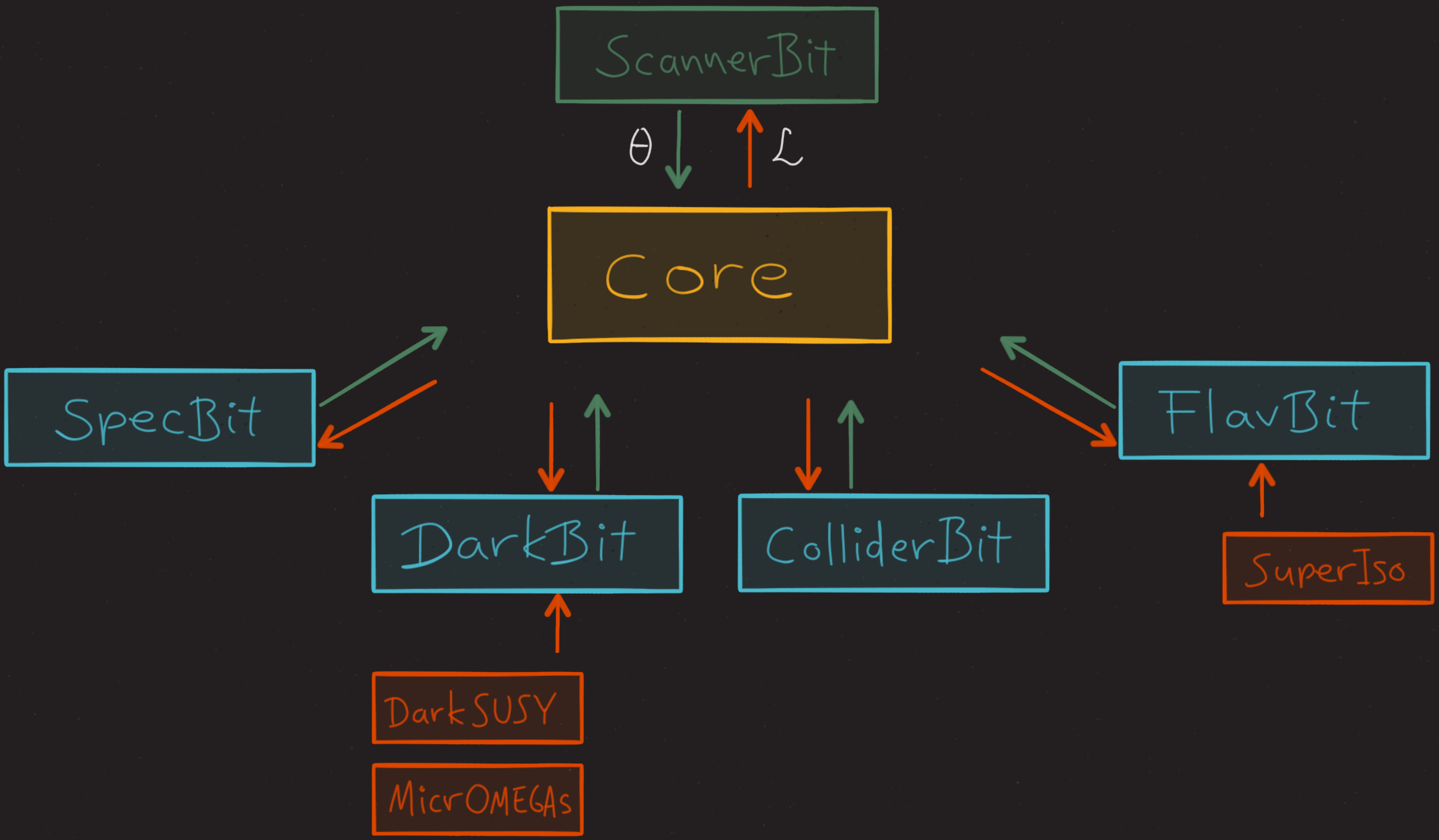
Each module presents the rest of GAMBIT with a set of *requirements* and a set of *capabilities*




Core
SpecBit
DarkBit
ColliderBit
FlavBit
SuperIso
DarkSUSY
MicrOMEGAs

The core solves the dependency tree
and decides the evaluation order




ScannerBit
θ
$\mathcal{L}$
Core
SpecBit
FlavBit
DarkBit
ColliderBit
SuperIso
DarkSUSY
MicrOMEGAs

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins

- C++

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins

- C++
- Both MPI and openMP parallelisation

GAMBIT

- Main design principles: modularity and flexibilty
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins

- C++
- Both MPI and openMP parallelisation
- Open-source release (in a few months)

Aspect	GAMBIT	MasterCode	SuperBayeS	Fittino	Rizzo et al.
Design	Modular, Adaptive	Monolithic	Monolithic	($\sim$)Monolithic	Monolithic
Statistics	Frequentist, Bayesian	Frequentist	Freq./Bayes.	Frequentist	None
Scanners	Differential evolution, genetic algorithms, random forests, t-walk, t-nest, particle swarm, nested sampling, MCMC, gradient descent	Nested sampling, MCMC, grad. descent	Nested sampling, MCMC	MCMC	None (random)
Theories	(p)MSSM-25, CMSSM$\pm\epsilon$, GMSB, AMSB, gaugino mediation, E6MSSM, NMSSM, BMSSM, PQMSSM, effective operators, iDM, XDM, ADM, UED, Higgs portals/extended Higgs sectors	CMSSM$\pm\epsilon$	(p)MSSM-15, CMSSM$\pm\epsilon$, mUED	CMSSM$\pm\epsilon$	(p)MSSM-19
Astroparticle	Event-level: IceCube, Fermi, LUX, XENON, CDMS, DM-ICE. Basic: $\Omega_{\rm DM}$, AMS-02, COUPP, KIMS, CRESST, CoGeNT, SIMPLE, PAMELA, Planck, HESS. Predictions: CTA, DARWIN, GAPS	Basic: $\Omega_{\rm DM}$, LUX, XENON	Basic: $\Omega_{\rm DM}$, Fermi, IceCube, XENON	Basic: $\Omega_{\rm DM}$, Fermi, HESS, XENON	Event-level: Fermi. Basic: $\Omega_{\rm DM}$, IceCube, CTA
LHC	ATLAS+CMS multi-analysis with neural net and fast detector simulation. Higgs multi-channel with correlations and no SM assumptions. Full flavour inc. complete $B \to X_s ll$ and $B \to K^* ll$ angular set.	ATLAS resim, HiggsSignals, basic flavour.	ATLAS direct sim, Higgs mass only, basic flavour.	ATLAS resim, HiggsSignals, basic flavour.	ATLAS+CMS +Tevatron direct sim, basic flavour.
SM, theory and related uncerts.	m_t, m_b, $\alpha_{\rm s}$, $\alpha_{\rm EM}$, DM halo, hadronic matrix elements, detector responses, QCD+EW corrections (LHC+DM signal+BG), astro BGs, cosmic ray hadronisation, coalescence and p'gation.	m_t, m_Z, $\alpha_{\rm EM}$, hadronic matrix elements	m_t, m_b, $\alpha_{\rm s}$, $\alpha_{\rm EM}$, DM halo, hadronic matrix elems.	m_t	None

Borrowed from P. Scott

Thank you for your attention!

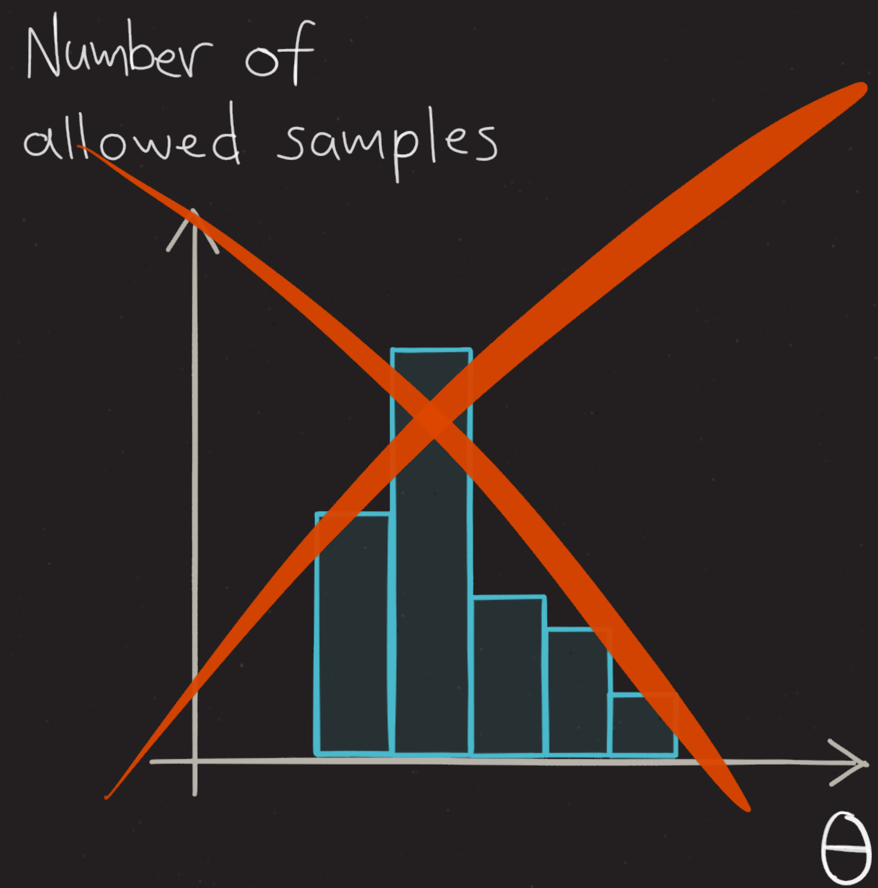
Backup slides

```
This is GAMBIT.

Backends                Version     Path to lib                                   Status          #funcs   #types   #ctors
--------------------------------------------------------------------------------------------------------------------------
BOSSMinimalExample      1.0         Backends/lib/libminimal_1_0.so                OK                0        2        4
                        1.1         Backends/lib/libminimal_1_1.so                OK                0        2        4
                        1.2         Backends/lib/libminimal_1_2.so                OK                0        2        4
DDCalc0                 0.0         Backends/lib/libDDCalc0.so                    OK                44       0        0
DarkSUSY                5.1.1       ../extras/DarkSUSY/lib/libdarksusy.so         OK                44       0        0
FastSim                 1.0         Backends/lib/libfastsim.so                    absent/broken     1        0        0
FeynHiggs               2.10        Backends/lib/libfeynhiggs.so                  absent/broken     11       0        0
HiggsBounds             4.1         Backends/lib/libhiggsbounds.so                absent/broken     8        0        0
HiggsSignals            1.2         Backends/lib/libhiggssignals.so               absent/broken     9        0        0
LibFarrayTest           1.0         Backends/lib/libFarrayTest.so                 OK                9        0        0
LibFirst                1.0         Backends/lib/libfirst.so                      OK                8        0        0
                        1.1         Backends/lib/libfirst.so                      OK                12       0        0
LibFortran              1.0         Backends/lib/libfortran.so                    OK                6        0        0
MicrOmegas              3.5.5       /no/path/in/config/backend_locations/         absent/broken     14       0        0
Pythia                  8.186       Backends/lib/libpythia8.so                    OK                0        6        37
SUSY_HIT                1.4         ../../SUSY-HIT/susyhit.so                     OK                51       0        0
SuperIso                3.4         Backends/lib/libsuperiso.so                   absent/broken     31       0        0
gamLike                 1.0.0       Backends/lib/libgamLike.so                    OK                5        0        0
nulike                  1.0.0       ../extras/nulike/lib/libnulike.so             OK                4        0        0
--------------------------------------------------------------------------------------------------------------------------
```

It is not meaningful to draw *general* conclusions directly from the number allowed/excluded points...

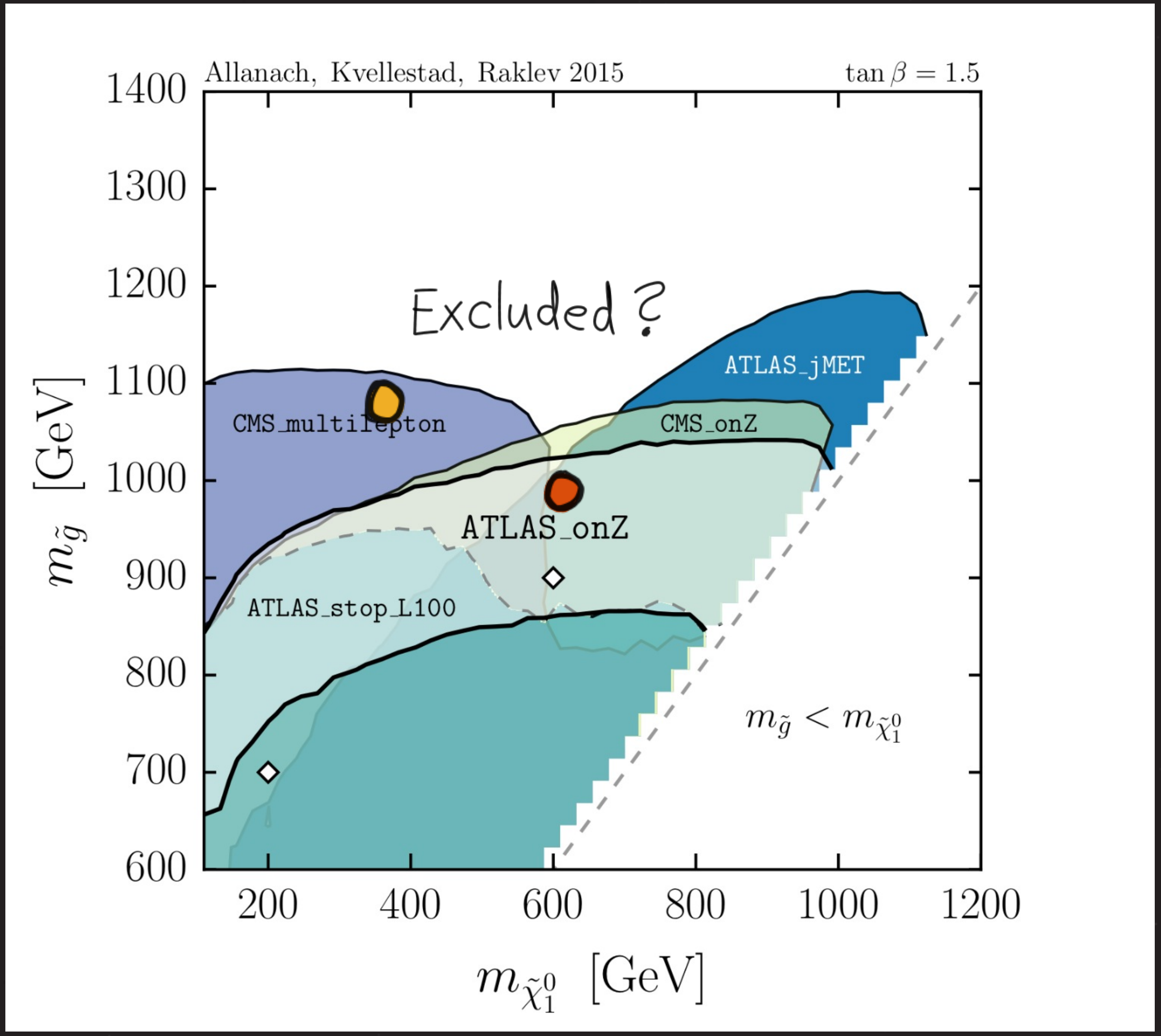
It is not meaningful to draw *general* conclusions directly from the number allowed/excluded points...



Only a couple of parameters:

Compare preferred/excluded regions for different analyses

Allanach, Kvellestad, Raklev: 1504.02752

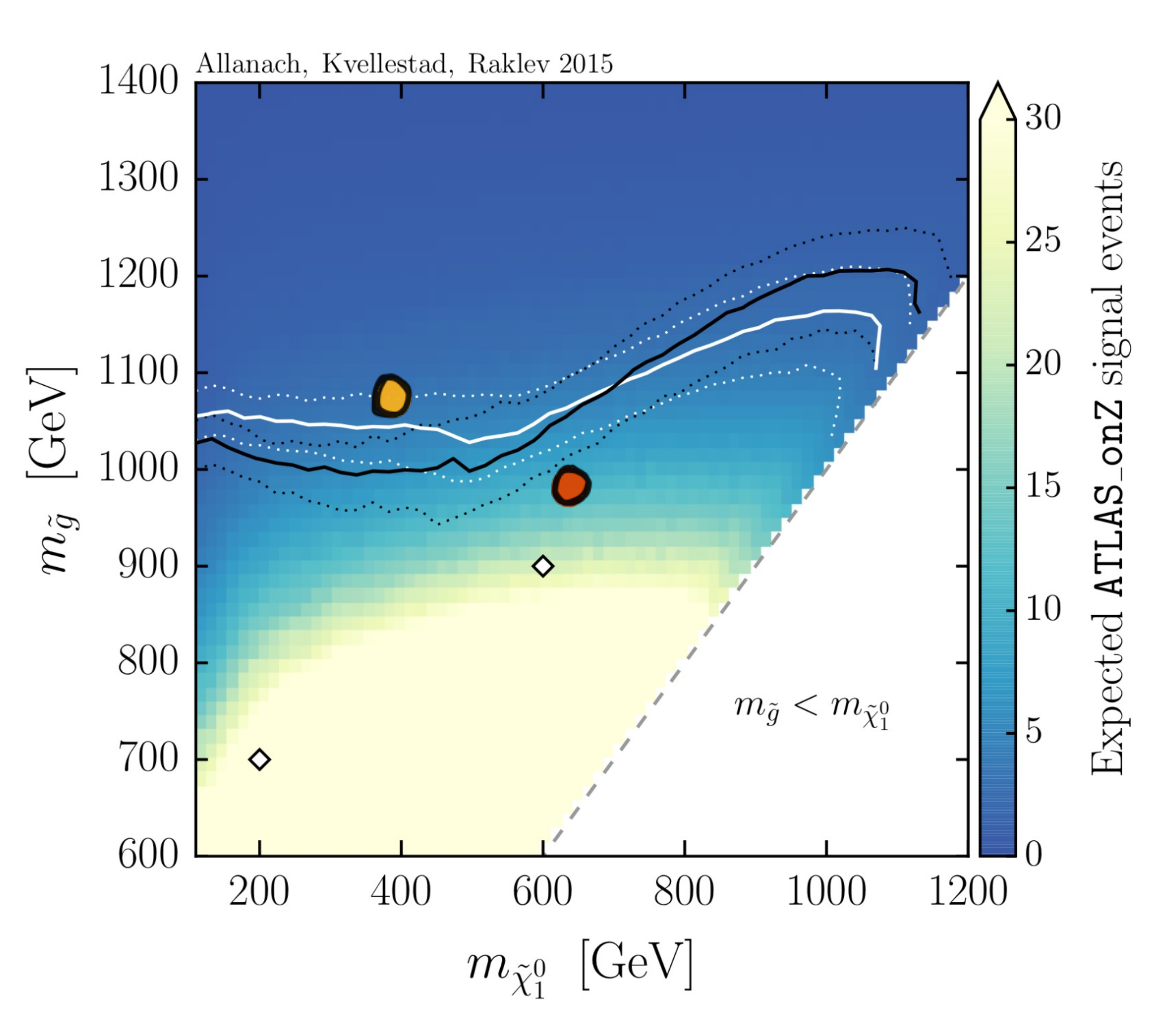


- \+ Simple to understand (at a qualitative level)
- ÷ Per-point interpretation is not straightforward
- ÷ Gets worse with increasing number of experimental analyses

If many different searches:

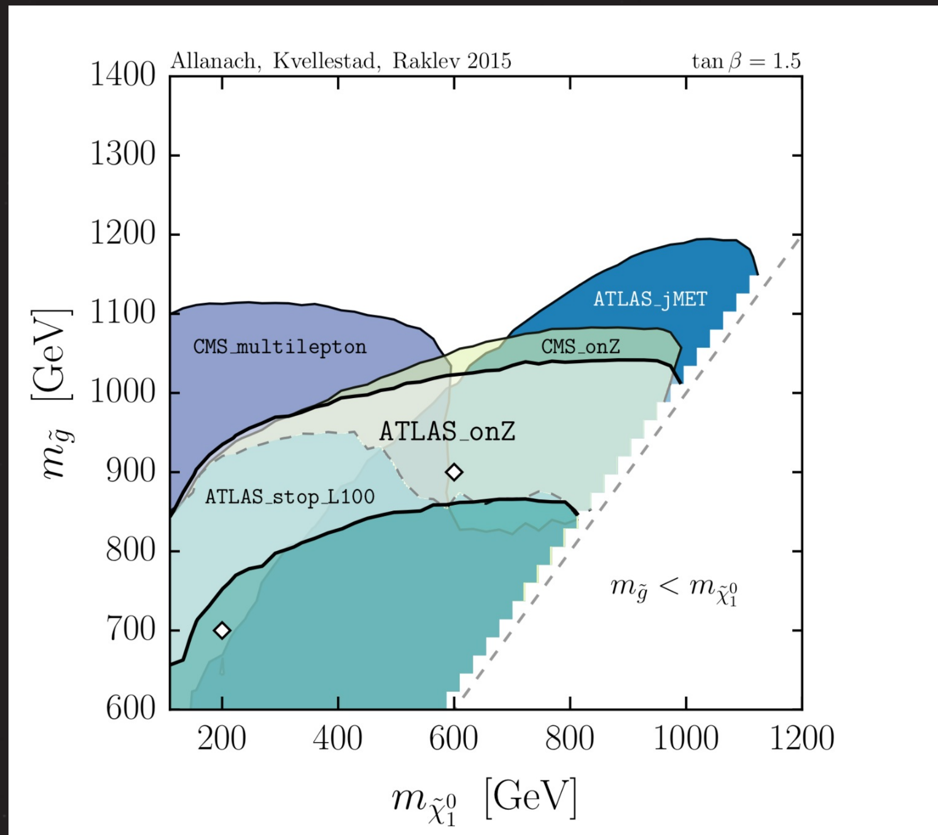
Combine searches in a total likelihood function

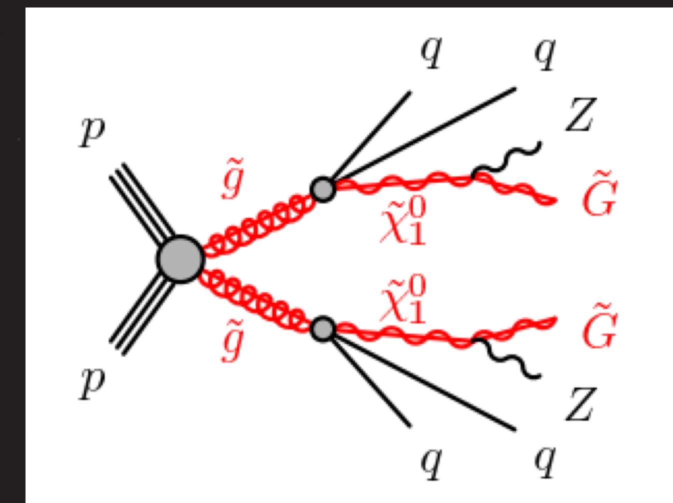
Allanach, Kvellestad, Raklev: 1504.02752

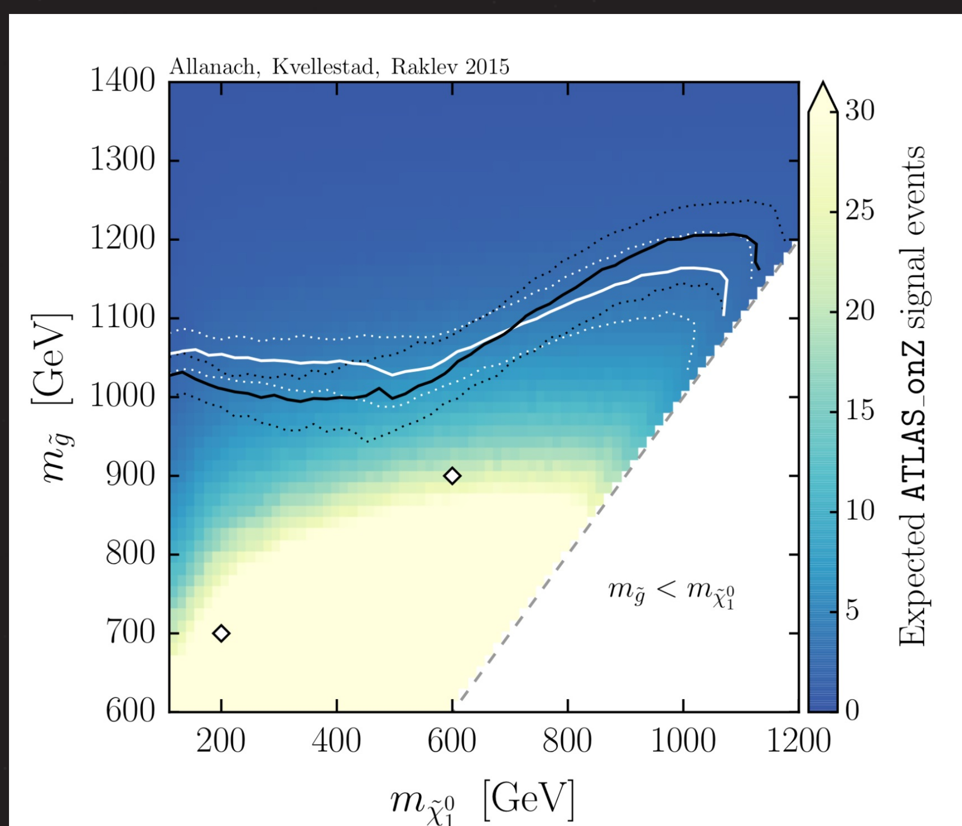


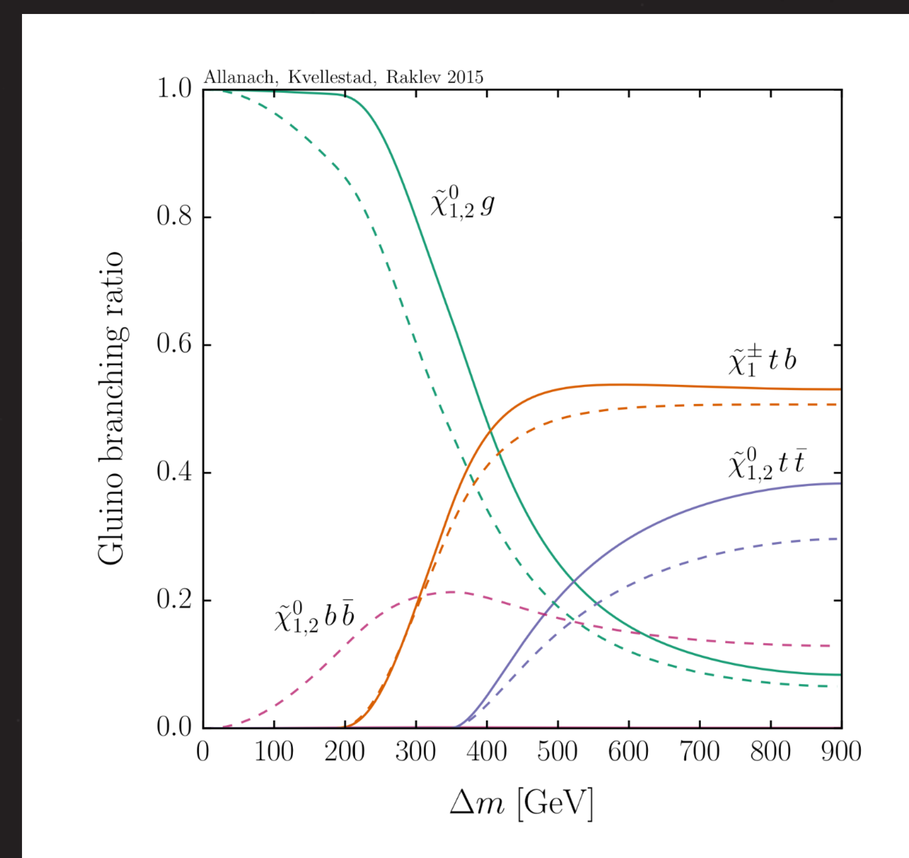
+ Clear per-point interpretation

÷ ...but what if there are many parameters?









Anders Kvellestad

Doing genuinely 'model-independent' DM pheno

All experimental limits in terms of simplified models: effective WIMP, one annihilation channel, etc

$\Longrightarrow$ need something to apply limits to arbitrary DD couplings and ID decay/annihilation branching fractions
$\Longrightarrow$ must include accurate treatment of experimental effects

Impacts of new unstable particles (e.g. extra Higgs) are hard

$\Longrightarrow$ need to simulate decays 'on the fly'

Calculating relic densities for general models also challenging

$\Longrightarrow$ want to feed in partial annihilation rates, co-annihilations, resonances, etc (not only set up model in LanHEP)

$\rightarrow$ nulike, gamlike, DDcalc, cascade sim $\rightarrow$ Christoph's talk

The LHC likelihood monster

Time per point:

$\mathcal{O}(\textit{minute})$ in best cases

Time per point for global fits to converge:

$\mathcal{O}(\textit{seconds})$ in worst cases

Challenge:

About 2 orders of magnitude too slow to actually include LHC data in global fits properly

→ More in Martin's presentation

Parameter space $\longrightarrow$ Theory space

CMSSM, MSSM, Simplified Models $\neq$ BSM

Want to do model comparison to actually work out which theory is the best. . .

Challenge:

How do I easily adapt a global fit to different BSM theories?

Interface: yaml file

Basic interface for a scan is a YAML initialisation file

- specify parameters, ranges, priors
- select likelihood components
- select other observables to calculate
- define generic rules for how to fill dependencies
- define generic rules for options to be passed to module functions
- set global options (scanner, errors/warnings, logging behaviour, etc)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:      Test
    capability:   decay_rates
    type:         DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

Modules

Physics Modules

- ColliderBit (Martin's talk)
- DarkBit (Christoph's talk)
- FlavBit – flavour physics inc. $g-2$, $b \to s\gamma$, B decays (new channels, theory uncerts, LHCb likelihoods)
- SpecBit – generic BSM spectrum object, providing RGE running, masses, mixings, etc via interchangeable interfaces to different RGE codes
- DecayBit – decay widths for all relevant SM & BSM particles
- EWPOBit – precision tests (mostly by interface to FeynHiggs, alt. SUSY-POPE)

+ScannerBit: manages statistics, parameter sampling and optimisation algorithms

Other nice technical features

- **Scanners**: MultiNest, Diver (diff. evolution), PIKAIA (genetic algorithms), GreAT (MCMC)
- **Statistics**: Bayesian, Profile Likelihood, later full Neyman
- Mixed-mode **MPI + openMP**, mostly automated
- diskless generalisation of various Les Houches Accords
- **BOSS**: dynamic loading of C++ classes from backends (!)
- **all-in or module standalone** modes – easily implemented from single cmake script
- **automatic getters** for obtaining, configuring + compiling backends[1]
- **flexible output streams** (ASCII, databases, binary, . . .)
- more more more. . .

[1] if a backend breaks, won't compile and/or kills your dog, blame the authors (not us. . . unless we **are** the authors. . .)