GAMBIT, nulike and future combinations

Pat Scott

Imperial College London

Slides available from tinyurl.com/patscott

nulike

nulike.hepforge.org

Imperial College London

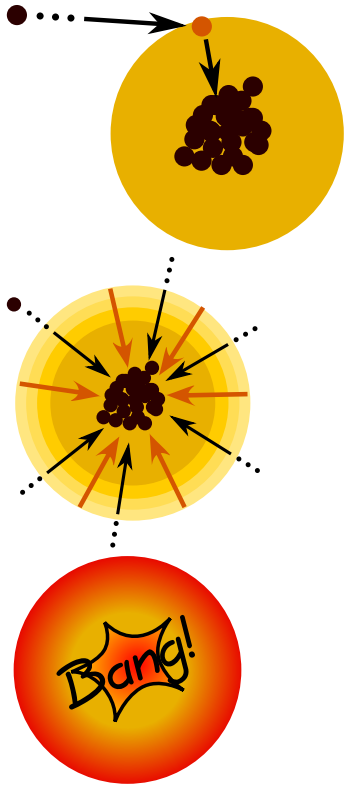
Neutrino telescope DM likelihoods: `nulike`

Unbinned ν telescope likelihood $\implies$ full event-level angular and energy info

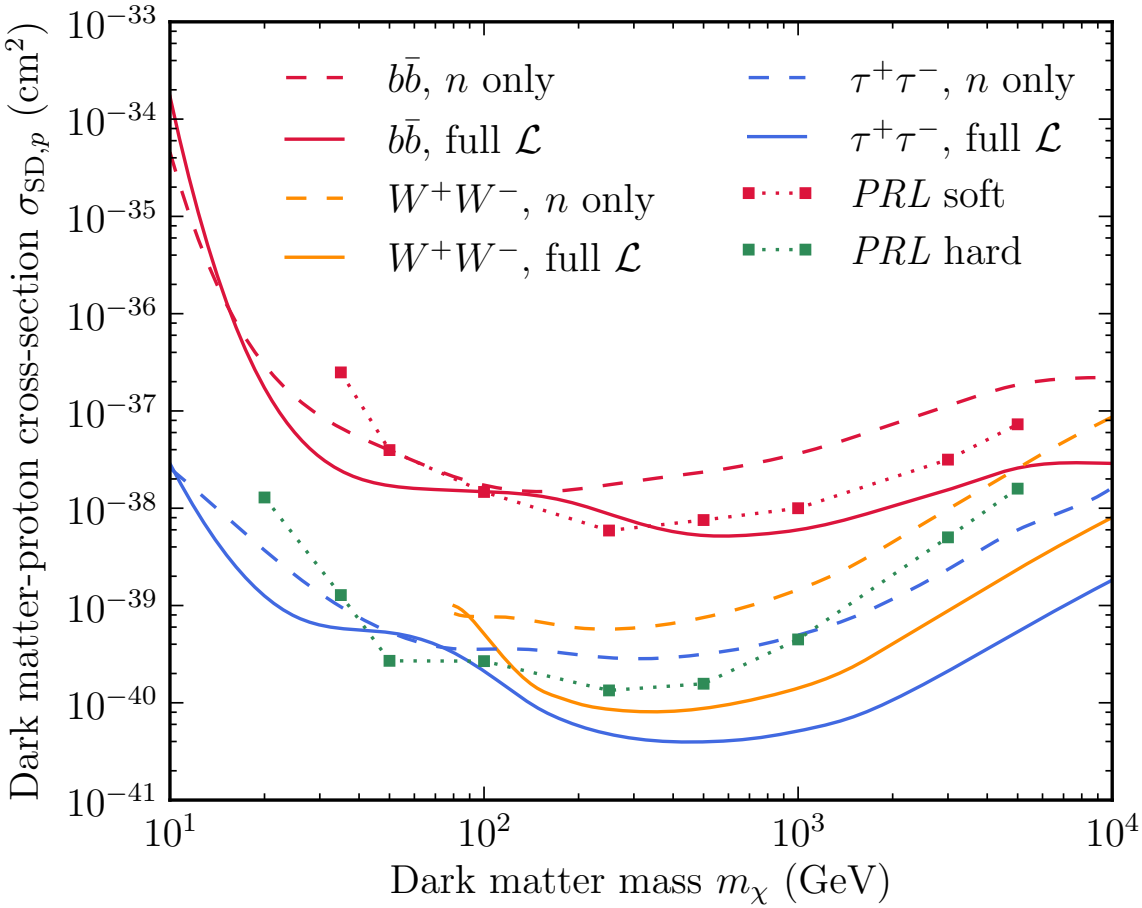
$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} (f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i})$$

Strategy: precompute partial likelihoods for each event, then reweight with the ν spectrum at Earth for each model

- precompute step uses `nusigma` with CTEQ6-DIS PDFs to get charged current $\nu - n$ and $\nu - p$ cross-sections as function of x and y
- like step input: neutrino spectrum at Earth (from DarkSUSY or whatever else you want to use)
- like step output: num predicted events, likelihood
- $\rightarrow$ fully model-independent = future-proof for global fits



79-string IceCube re-analysis

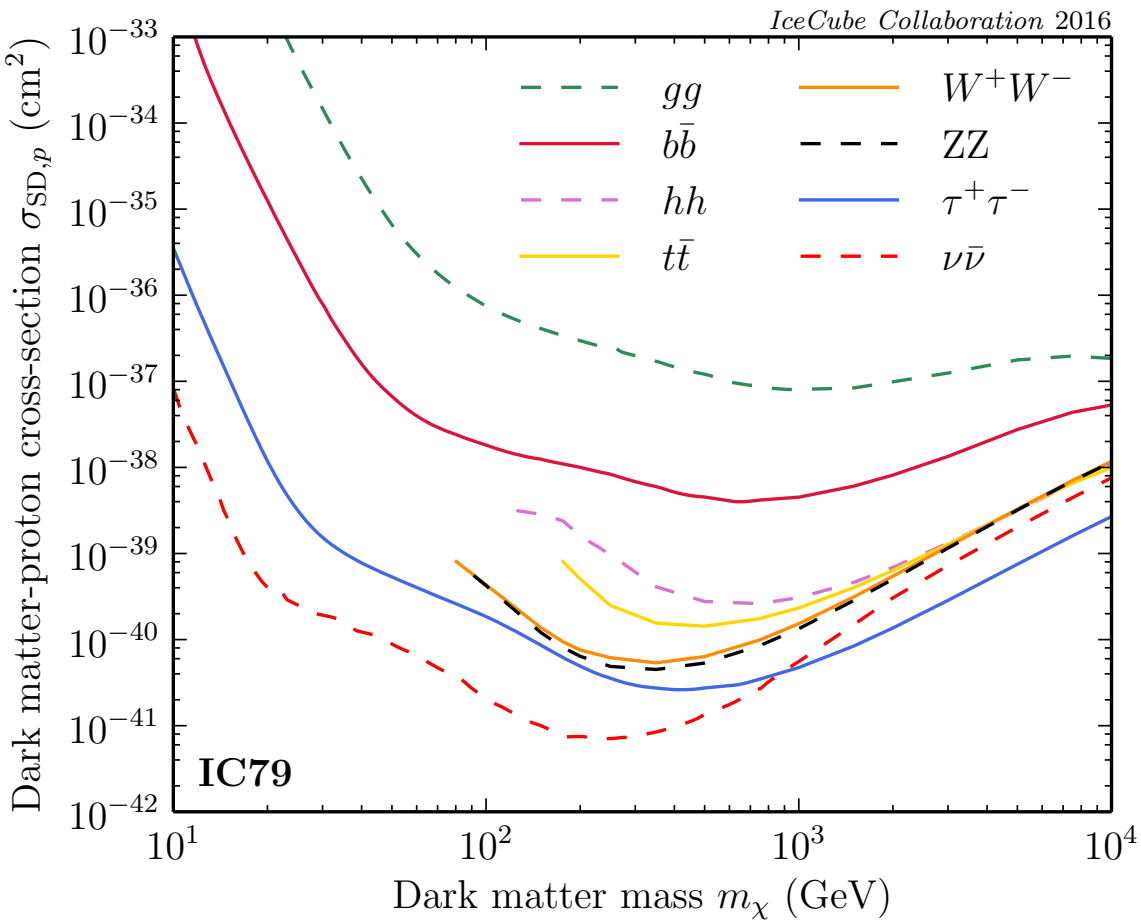


IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* 2016

`nulike`: model-independent unbinned limit calculator for generic BSM models

Imperial College London

79-string IceCube re-analysis

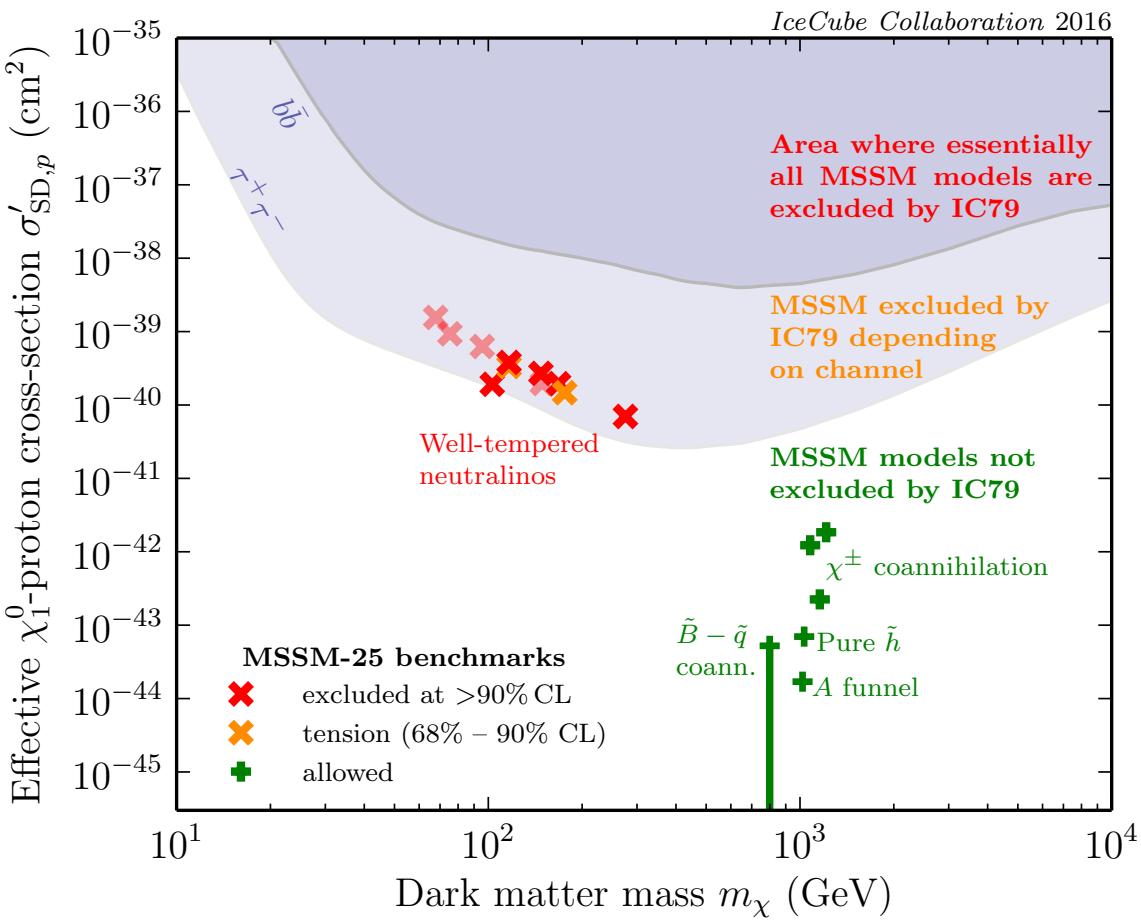


IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* 2016

`nulike`: model-independent unbinned limit calculator for generic BSM models

Imperial College London

79-string IceCube re-analysis

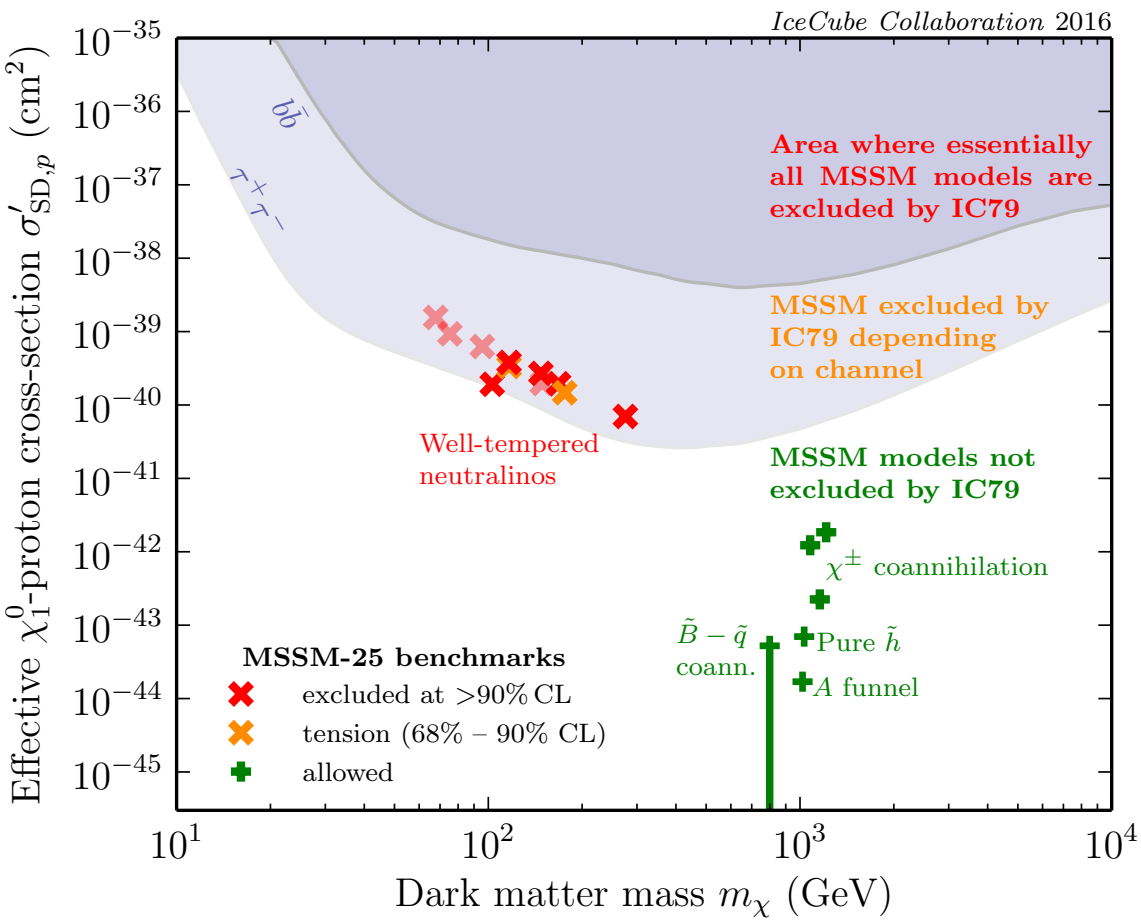


IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* 2016

`nulike`: model-independent unbinned limit calculator for generic BSM models

Imperial College London

79-string IceCube re-analysis



Reprocessing 86-string data to recastable (=`nulike`) format still needs doing...
(Also needs public release)

Imperial College London

Prospects for combination of data

Combination with all other DM searches

- Direct detection, other indirect detection, collider, relic density,. . .
- Ideally, want to be able to deal with *any* DM model
- and easily add new likelihoods, observables, samplers, datasets, etc

$\rightarrow$ need a new, flexibe and modular global fitting tool

Imperial College London

GAMBIT

gambit.hepforge.org

Imperial College London

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, CMS, CTA, *Fermi*-LAT, DARWIN, IceCube, LHCb, SHiP, XENON

Authors of: DarkSUSY, DDCalc, Diver, FlexibleSUSY, gamlike, GM2Calc, IsaJet, nulike, PolyChord, Rivet, SOFTSUSY, SuperIso, SUSY-AI, WIMPSim

Collaborators:
Peter Athron, Csaba Balázs, Ankit Beniwal, Florian Bernlochner, Sanjay Bloor, Torsten Bringmann, Andy Buckley, Eliel Camargo-Molina, Marcin Chrząszc, Jan Conrad, Jonathan Cornell, Matthias Danninger, Tom Edwards, Joakim Edsjö, Ben Farmer, Andrew Fowlie, Tomás Gonzalo, Will Handley, Sebastian Hoof, Selim Hotinli, Felix Kahlhoefer, Suraj Krishnamurthy, Anders Kvellestad, Julia Harz, Paul Jackson, Tong Li, Greg Martinez, Nazilla Mahmoudi, James McKay, Are Raklev, Janina Renk, Chris Rogan, Roberto Ruiz de Austri, Patrick Stoecker, Roberto Trotta, Pat Scott, Nicola Serra, Daniel Steiner, Puwen Sun, Aaron Vincent, Christoph Weniger, Sebastian Wild, Martin White, Yang Zhang

40+ participants in 10 Experiments & 14 major theory codes

Modules

Physics modules

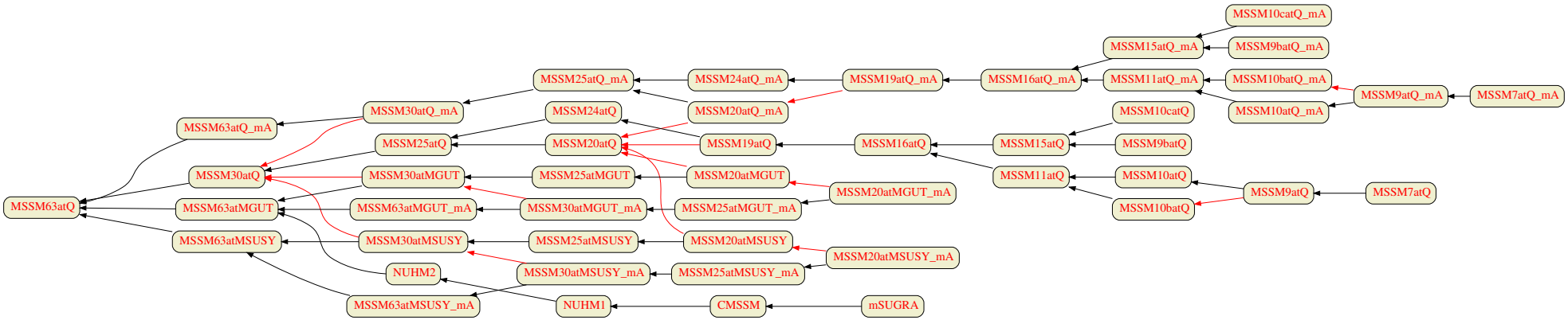
- DarkBit – dark matter observables (EPJC, arXiv:1705.07920)
- ColliderBit – collider observables (EPJC, arXiv:1705.07919)
- FlavBit – flavour physics (EPJC, arXiv:1705.07933)
- SpecBit – particle masses (EPJC, arXiv:1705.07936)
- DecayBit – decay widths (EPJC, arXiv:1705.07936)
- PrecisionBit – nuisance parameters and precision tests (EPJC, arXiv:1705.07936)

Statistics

- ScannerBit – priors, sampling, etc (EPJC, arXiv:1705.07959)

- Models are defined by their parameters and relations to each other
- Models can inherit from (be subspaces of) **parent models**
- Points in child models can be **automatically translated** to ancestor models
- **Friend models** also allowed (cross-family translation)
- Model dependence of every function/observable is tracked

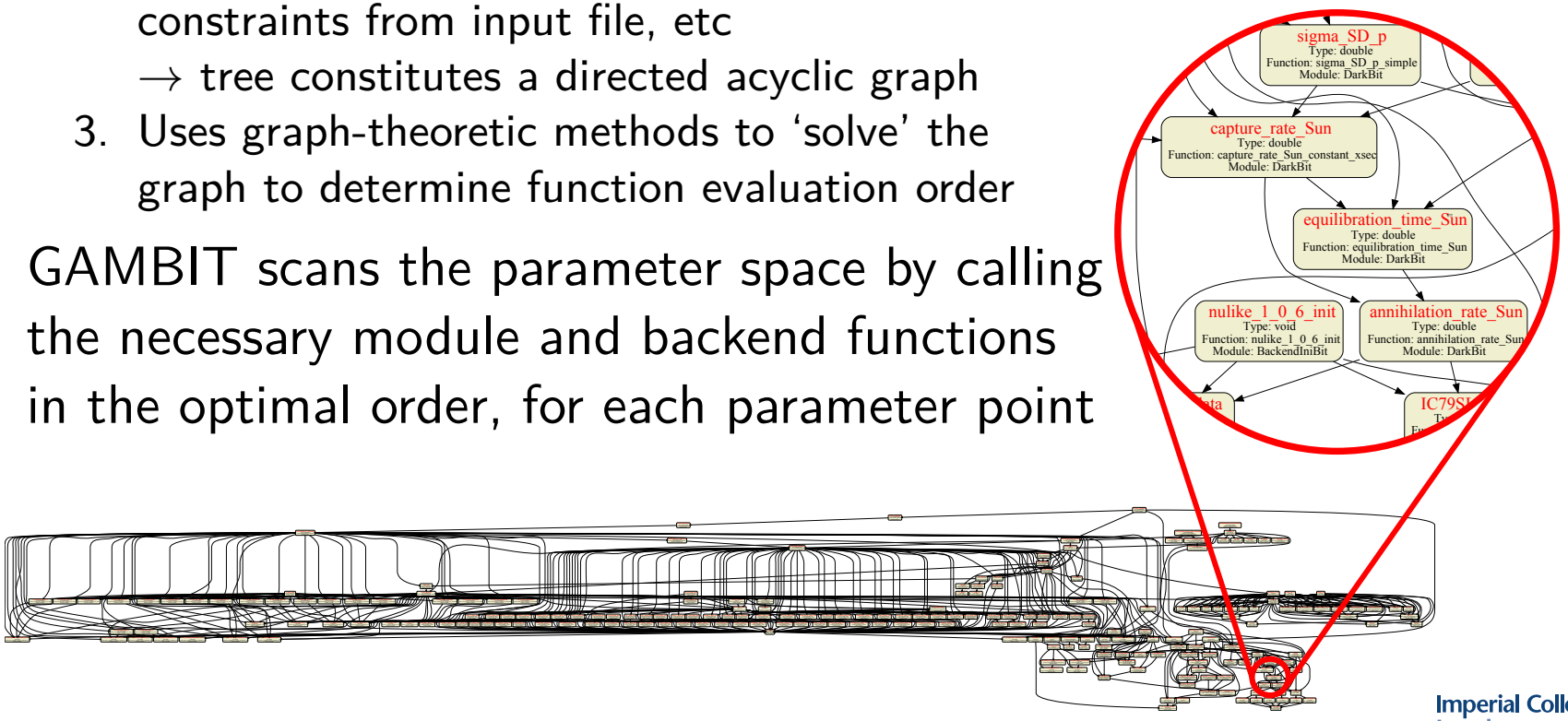
  $\Longrightarrow$ maximum safety, maximum reuse



Imperial College London

Anatomy of a GAMBIT run (EPJC, arXiv:1705.07908)

- User chooses a model to scan, which observables to include, and the scanning method
- GAMBIT constructs a **dependency tree**
  1. Identifies which functions and inputs are needed to compute the requested observables
  2. Obeys **rules** at each step: allowed models, allowed backends, constraints from input file, etc
     $\rightarrow$ tree constitutes a directed acyclic graph
  3. Uses graph-theoretic methods to 'solve' the graph to determine function evaluation order
- GAMBIT scans the parameter space by calling the necessary module and backend functions in the optimal order, for each parameter point

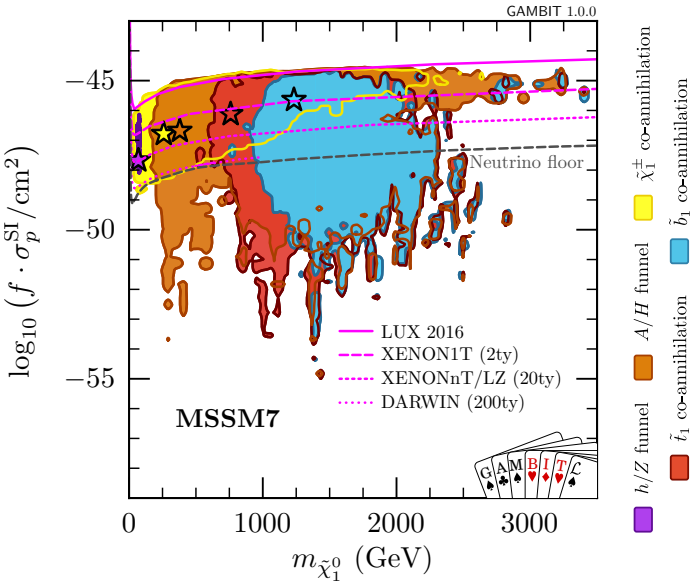


Imperial College London

13 GAMBIT papers so far (7 physics + 6 code)

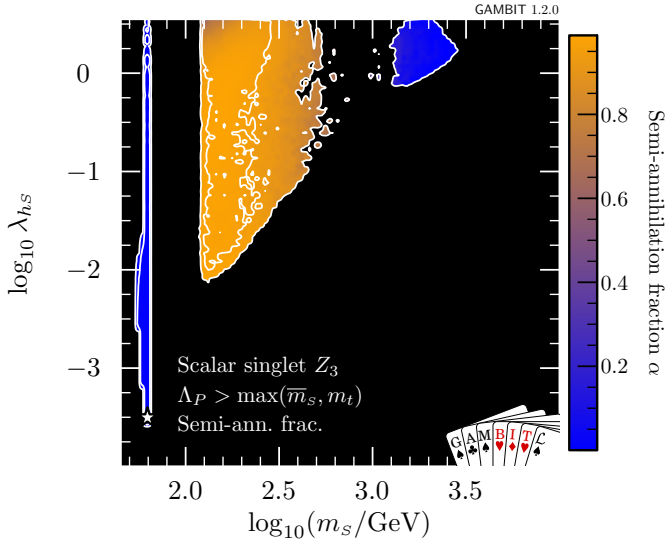
Supersymmetry

EPJC arXiv:1705.07917, 1705.07935, 1809.02097



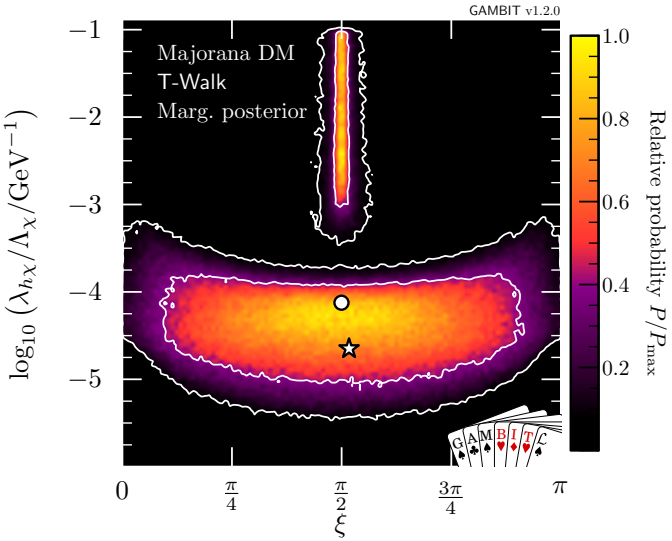
Scalar singlet dark matter

EPJC arXiv:1705.07931, 1806.11281



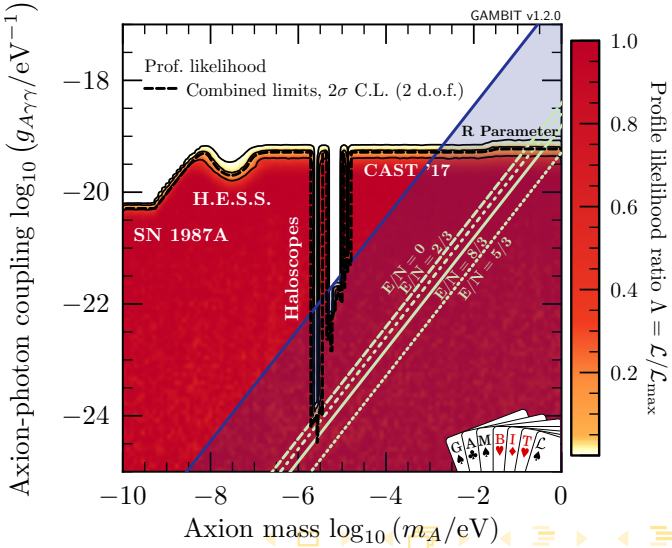
Vector & fermion Higgs-portal dark matter

arXiv:1808.10465



Axions & axion-like particles

arXiv:1810.07192

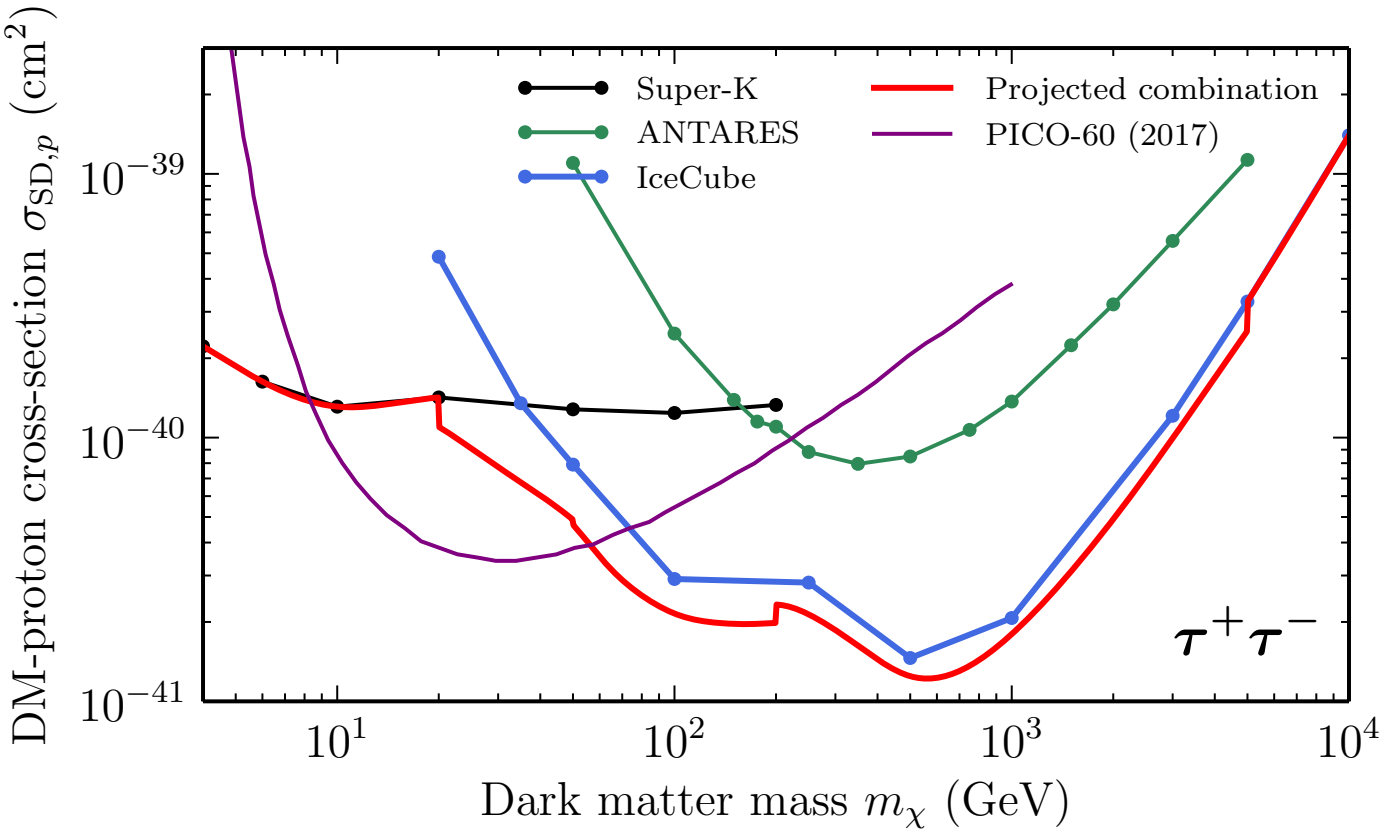


Imperial College London

Future combinations

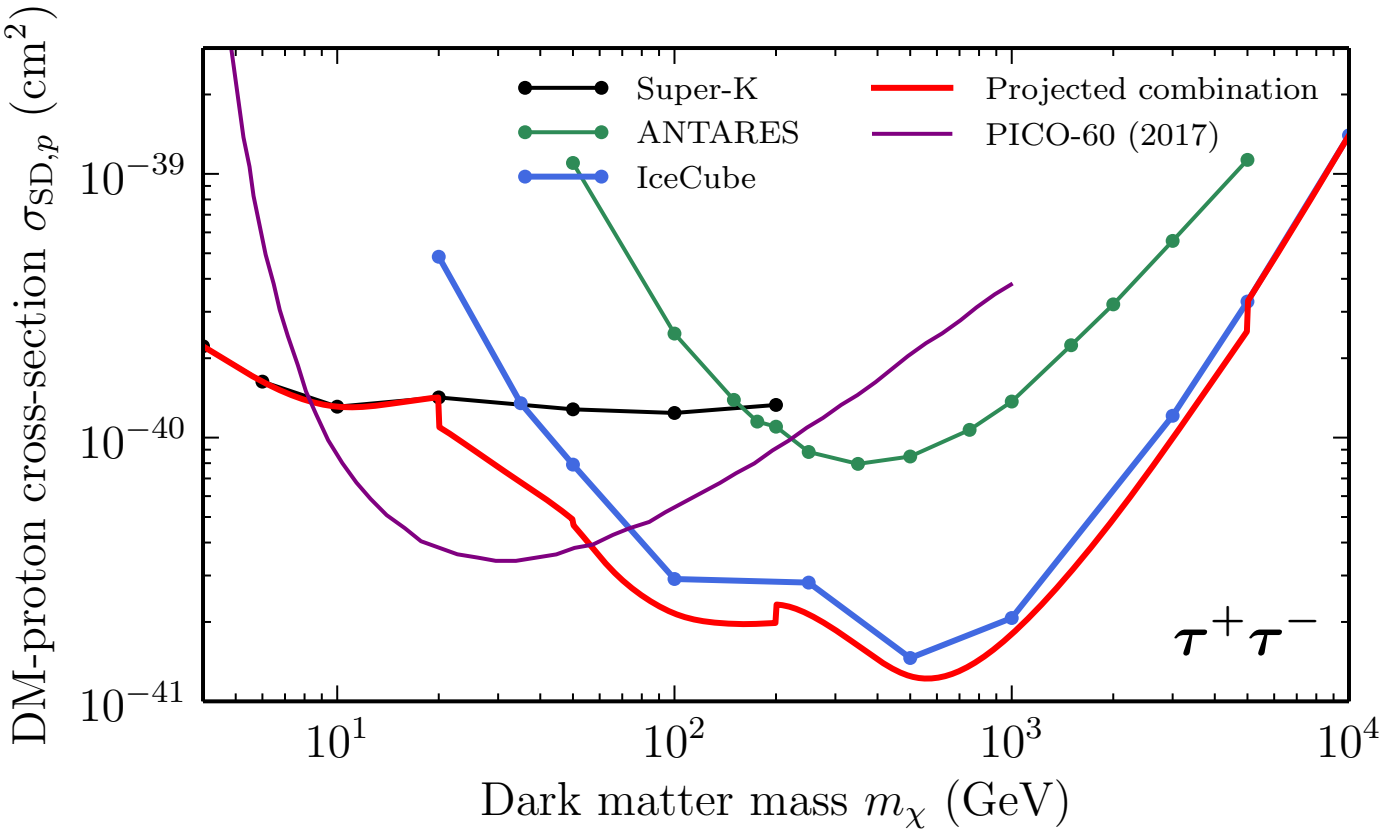
Combination of solar DM search limits

- Combined IceCube, ANTARES & SuperK limit would be straightforward in `nulike` framework.



Combination of solar DM search limits

- Combined IceCube, ANTARES & SuperK limit would be straightforward in `nulike` framework.
  Needs ANTARES and SuperK data in `nulike` format.



Imperial College London

Combination of solar DM search limits

- Combined IceCube, ANTARES & SuperK limit would be straightforward in `nulike` framework.
  Needs ANTARES and SuperK data in `nulike` format.
- ANTARES data is almost identical to IceCube format
  $\rightarrow$ should just 'slot in'
- SuperK data less so due to electromagnetic cascades from ν_e
  $\rightarrow$ some work required

Imperial College London

Combination of solar DM search limits

- Combined IceCube, ANTARES & SuperK limit would be straightforward in `nulike` framework.
  Needs ANTARES and SuperK data in `nulike` format.
- ANTARES data is almost identical to IceCube format
  $\rightarrow$ should just 'slot in'
- SuperK data less so due to electromagnetic cascades from ν_e
  $\rightarrow$ some work required

- Easy to extend to Gen2-KM3NeT-HyperK combinations

Imperial College London

Summary

- `nulike` provides model-independent application of IC79 limits
- Needs IC86, ANTARES and SuperK inputs
- straightforward experimental combination
- straightforward combination with all other DM searches via GAMBIT
- ideal framework for future neutrino telescope combination
- `nulike` is also a viable route for recasting and combining DM annihilation limits from neutrino telescopes

- `nulike` code: nulike.hepforge.org
- GAMBIT code: gambit.hepforge.org
- GAMBIT results, samples, run files, best fits, benchmarks, etc are *all* available to download from Zenodo: www.zenodo.org/communities/gambit-official/

Imperial College London

Horrible details

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln\epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \iint_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln\epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E,\phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E,\phi,\vec{\boldsymbol{\xi}})\,{\rm d}E\,{\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E,\vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \iint_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x,y|E)\,{\rm d}x\,{\rm d}y\,{\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x,y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x,y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu){\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln \epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \iint_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln\epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \iint_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln \epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\begin{aligned} \mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) &= \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \\ &= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \int_0^1\!\!\int Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \end{aligned} \tag{4, 5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln \epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \iint_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Imperial College London

Likelihoods in nulike (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln \epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \int\int_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Nulike precomputes and tabulates these over E for every event

Imperial College London

Likelihoods in `nulike` (the abridged version)

$$\mathcal{L}_{\rm unbin} \equiv \mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) \prod_{i=1}^{n_{\rm tot}} \left(f_{\rm S}\mathcal{L}_{{\rm S},i} + f_{\rm BG}\mathcal{L}_{{\rm BG},i} \right) , \tag{1}$$

$$\mathcal{L}_{\rm num}(n_{\rm tot}|\theta_{\rm tot}) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \int_0^\infty \frac{(\theta_{\rm BG} + \epsilon\theta_{\rm S})^{n_{\rm tot}} e^{-(\theta_{\rm BG}+\epsilon\theta_{\rm S})}}{n_{\rm tot}!} \frac{1}{\epsilon} \exp\left[-\frac{1}{2} \left(\frac{\ln \epsilon}{\sigma_\epsilon} \right)^2 \right] {\rm d}\epsilon , \tag{2}$$

$$\mathcal{L}_{{\rm BG},i}(N_i^{\rm c}, \phi_i') = \frac{{\rm d}P_{\rm BG}}{{\rm d}N_i^{\rm c}}(N_i^{\rm c}) \frac{{\rm d}P_{\rm BG}}{{\rm d}\phi_i'}(\phi_i') , \tag{3}$$

$$\mathcal{L}_{{\rm S},i}(N_i^{\rm c}, \phi_i'|\vec{\boldsymbol{\xi}}) = \int_0^\pi \int_0^\infty Q(N_i^{\rm c}, \phi_i'|E, \phi) \frac{{\rm d}^2 P_{\rm S}}{{\rm d}E\,{\rm d}\phi}(E, \phi, \vec{\boldsymbol{\xi}})\, {\rm d}E\, {\rm d}\phi , \tag{4}$$

$$= \frac{t_{\rm exp}}{\theta_{\rm S}} \sum_{\nu,\bar{\nu}} \int_0^\infty \frac{{\rm d}\Phi_\nu}{{\rm d}E}(E, \vec{\boldsymbol{\xi}}) f_\nu^{\rm b}(E) \int\int_0^1 Q_\mu \left(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu \right) \frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E)\, {\rm d}x\, {\rm d}y\, {\rm d}E , \tag{5}$$

$$\frac{{\rm d}^2\Sigma_{\nu\to\mu}}{{\rm d}x\,{\rm d}y}(x, y|E) = V_{\rm eff}(E_\mu) \sum_{N=p,n} n_N \frac{{\rm d}^2\sigma_{\nu\to\mu,N}}{{\rm d}x\,{\rm d}y}(x, y|E) , \tag{6}$$

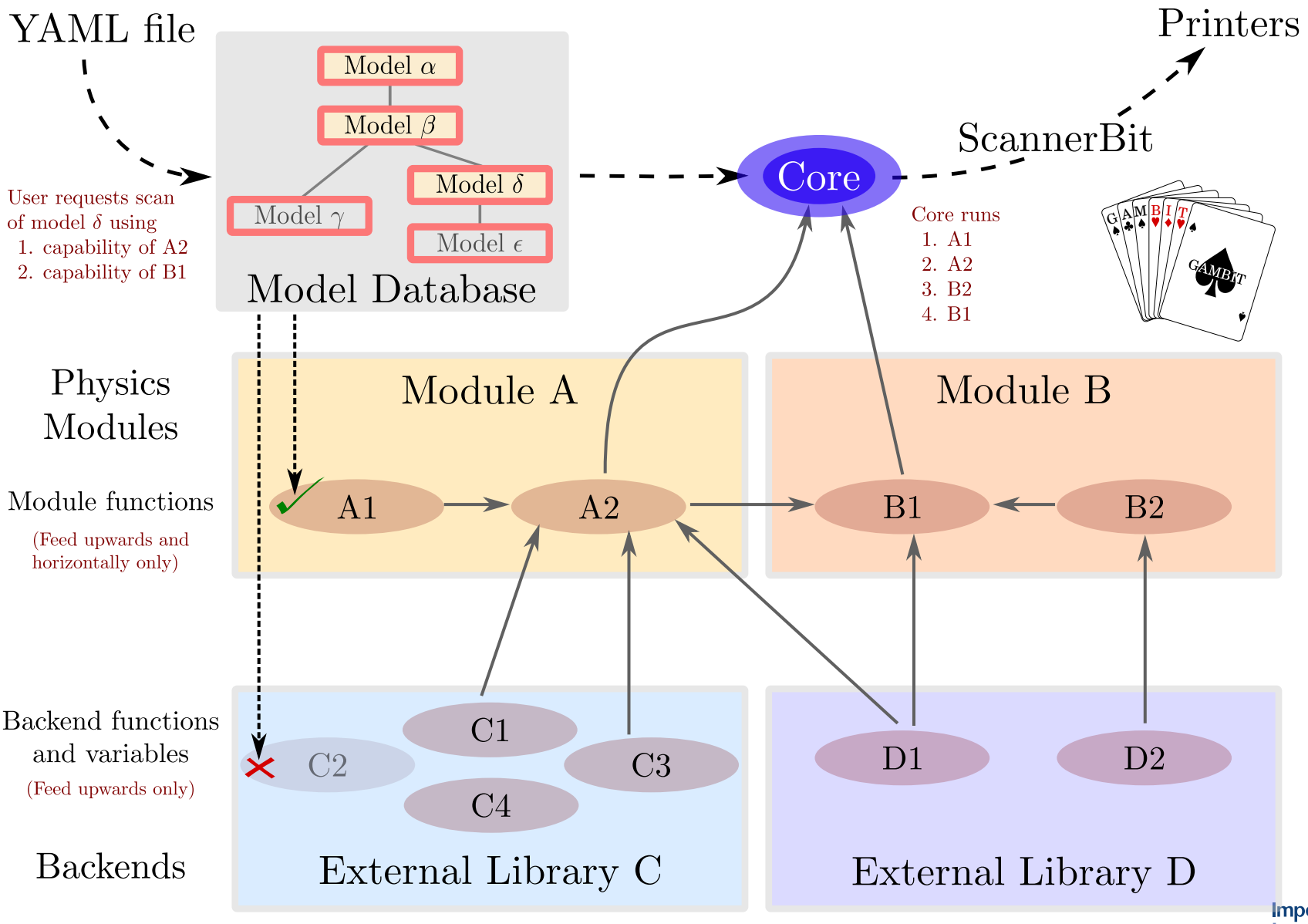
$$Q_\mu(N_i^{\rm c}, \phi_i'|E_\mu, \phi_\mu) = E_{\rm disp}(N_i^{\rm c}|E_\mu) {\rm PSF}(\phi_i'|\phi_\mu, E_\mu) . \tag{7}$$

Nulike precomputes and tabulates these over E for every event

This is what needs to be provided by theory

Imperial College London

Functional overview of GAMBIT (EPJC, arXiv:1705.07908)



Backends: mix and match (EPJC, arXiv:1705.07908)

- Module functions can require specific functions from **backends**
- Backends are external code libraries (DarkSUSY, FeynHiggs, etc) that include different functions
- GAMBIT automates and abstracts the interfaces to backends $\rightarrow$ backend functions are tagged according to what they calculate
- $\rightarrow$ with appropriate module design, different backends and their functions can be used interchangeably
- GAMBIT dynamically adapts to use whichever backends are actually present on a user's system (+ provides details of what it decided to do of course)

Imperial College London

Backends: mix and match (EPJC, arXiv:1705.07908)

pat@xpspedition: ~/gambit 163x45

All relative paths are given with reference to /home/pat/gambit.

BACKENDS	VERSION	PATH TO LIB	STATUS	#FUNC	#TYPES	#CTORS
DDCalc0	0.0	Backends/installed/DDCalc/0.0/libDDCalc0.so	OK	62	0	0
DarkSUSY	5.1.1	Backends/installed/DarkSUSY/5.1.1/lib/libdarksusy.so	OK	68	0	0
FastSim	1.0	Backends/installed/fastsim/1.0/libfastsim.so	absent/broken	1	0	0
FeynHiggs	2.11	Backends/installed/FeynHiggs/2.11.2/lib/libFH.so	OK	14	0	0
HiggsBounds	4.2.1	Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so	OK	10	0	0
HiggsSignals	1.4	Backends/installed/HiggsSignals/1.4.0/lib/libhiggssignals.so	OK	11	0	0
LibFarrayTest	1.0	Backends/examples/libFarrayTest.so	OK	9	0	0
LibFirst	1.0	Backends/examples/libfirst.so	OK	8	0	0
	1.1	Backends/examples/libfirst.so	OK	15	0	0
LibFortran	1.0	Backends/examples/libfortran.so	OK	6	0	0
MicrOmegas	3.5.5	Backends/installed/micromegas/3.5.5/MSSM/MSSM/libmicromegas.so	OK	15	0	0
MicrOmegasSingletDM	3.5.5	Backends/installed/micromegas/3.5.5/SingletDM/SingletDM/libmicromegas.so	OK	13	0	0
Pythia	8.186	Backends/installed/Pythia/8.186/lib/libpythia8.so	absent/broken	0	27	105
	8.209	Backends/installed/Pythia/8.209/lib/libpythia8.so	OK	0	28	107
SUSYPOPE	0.2	no path in config/backend_locations.yaml	absent/broken	3	0	0
SUSY_HIT	1.5	Backends/installed/SUSY-HIT/1.5/libsusyhit.so	OK	55	0	0
SuperIso	3.4	Backends/installed/SuperIso/3.4/libsuperiso.so	OK	32	0	0
gamLike	1.0.0	Backends/installed/gamLike/1.0.0/lib/gamLike.so	OK	3	0	0
nulike	1.0.0	Backends/installed/nulike/1.0.0/lib/libnulike.so	OK	4	0	0

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

Backends: mix and match (EPJC, arXiv:1705.07908)

pat@xpspedition: ~/gambit 163x45

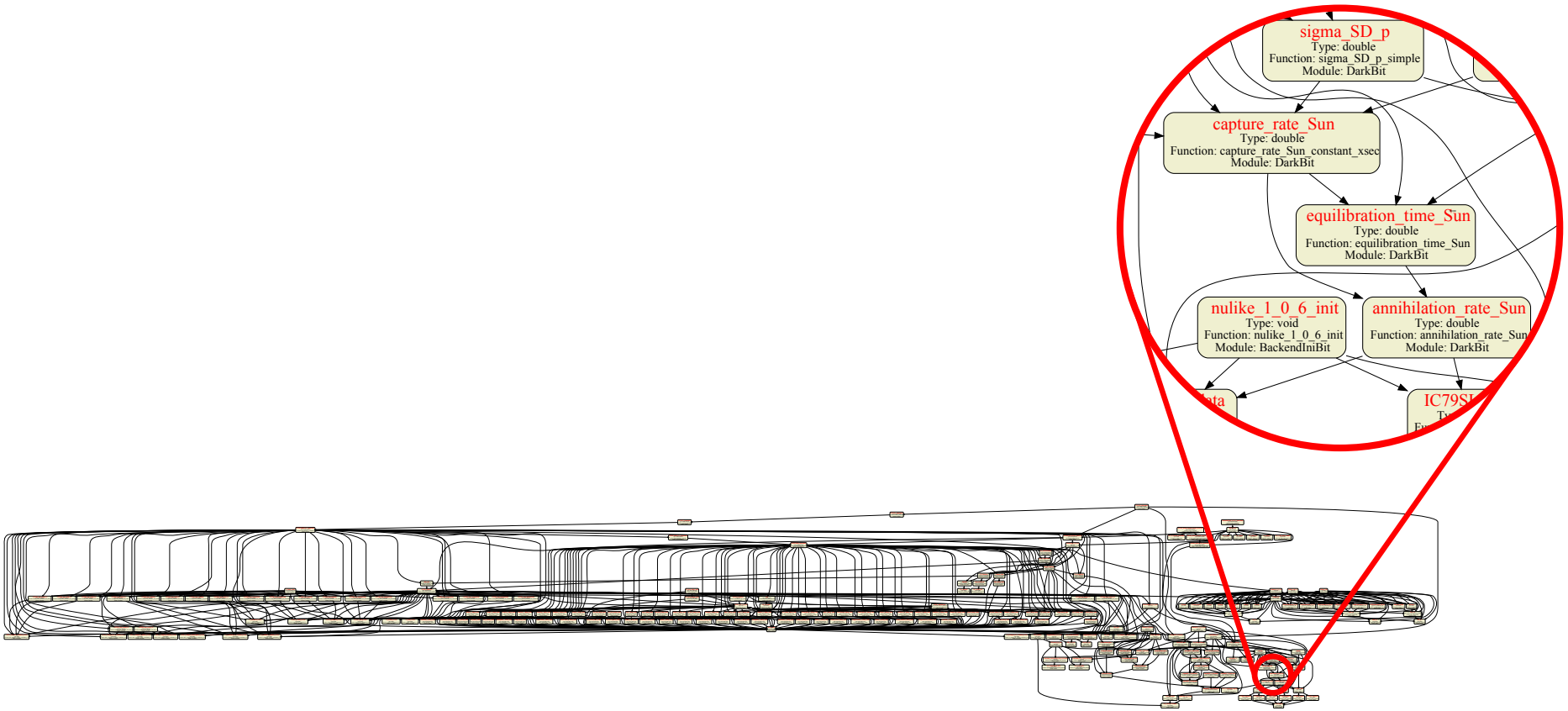
All relative paths are given with reference to /home/pat/gambit.

BACKENDS	VERSION	PATH TO LIB	STATUS	#FUNC	#TYPES	#CTORS
DDCalc0	0.0	Backends/installed/DDCalc/0.0/libDDCalc0.so	OK	62	0	0
DarkSUSY	5.1.1	Backends/installed/DarkSUSY/5.1.1/lib/libdarksusy.so	OK	68	0	0
FastSim	1.0	Backends/installed/fastsim/1.0/libfastsim.so	absent/broken	1	0	0
FeynHiggs	2.11	Backends/installed/FeynHiggs/2.11.2/lib/libFH.so	OK	14	0	0
HiggsBounds	4.2.1	Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so	OK	10	0	0
HiggsSignals	1.4	Backends/installed/HiggsSignals/1.4.0/lib/libhiggssignals.so	OK	11	0	0
LibFarrayTest	1.0	Backends/examples/libFarrayTest.so	OK	9	0	0
LibFirst	1.0	Backends/examples/libfirst.so	OK	8	0	0
	1.1	Backends/examples/libfirst.so	OK	15	0	0
LibFortran	1.0	Backends/examples/libfortran.so	OK	6	0	0
MicrOmegas	3.5.5	Backends/installed/micromegas/3.5.5/MSSM/MSSM/libmicromegas.so	OK	15	0	0
MicrOmegasSingletDM	3.5.5	Backends/installed/micromegas/3.5.5/SingletDM/SingletDM/libmicromegas.so	OK	13	0	0
Pythia	8.186	Backends/installed/Pythia/8.186/lib/libpythia8.so	absent/broken	0	27	105
	8.209	Backends/installed/Pythia/8.209/lib/libpythia8.so	OK	0	28	107
SUSYPOPE	0.2	no path in config/backend_locations.yaml	absent/broken	3	0	0
SUSY_HIT	1.5	Backends/installed/SUSY-HIT/1.5/libsusyhit.so	OK	55	0	0
SuperIso	3.4	Backends/installed/SuperIso/3.4/libsuperiso.so	OK	32	0	0
gamLike	1.0.0	Backends/installed/gamLike/1.0.0/lib/gamLike.so	OK	3	0	0
nulike	1.0.0	Backends/installed/nulike/1.0.0/lib/libnulike.so	OK	4	0	0

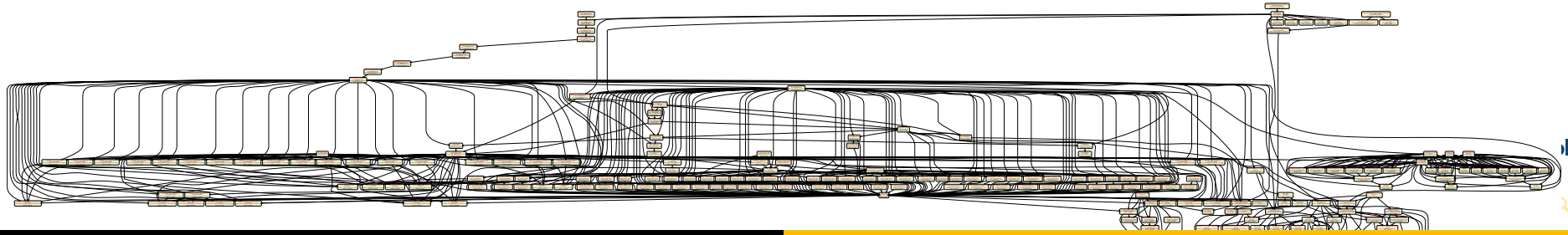
Gambit diagnostic backend line 1 (press h for help or q to quit)

Dependency Resolution

CMSSM:

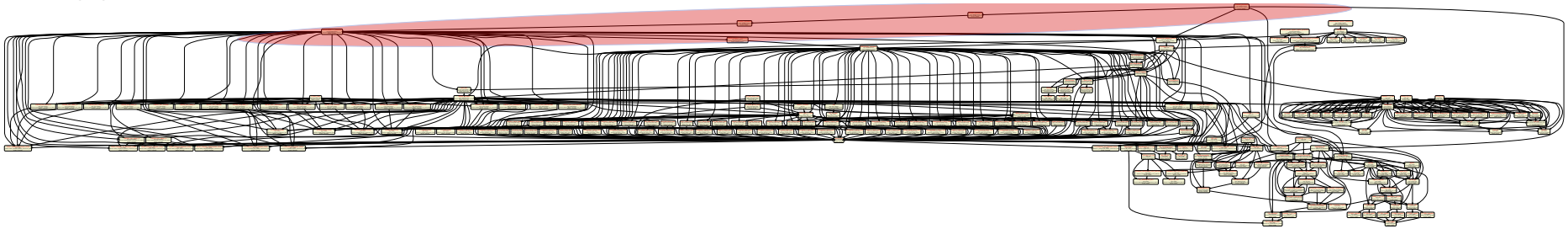


MSSM7:

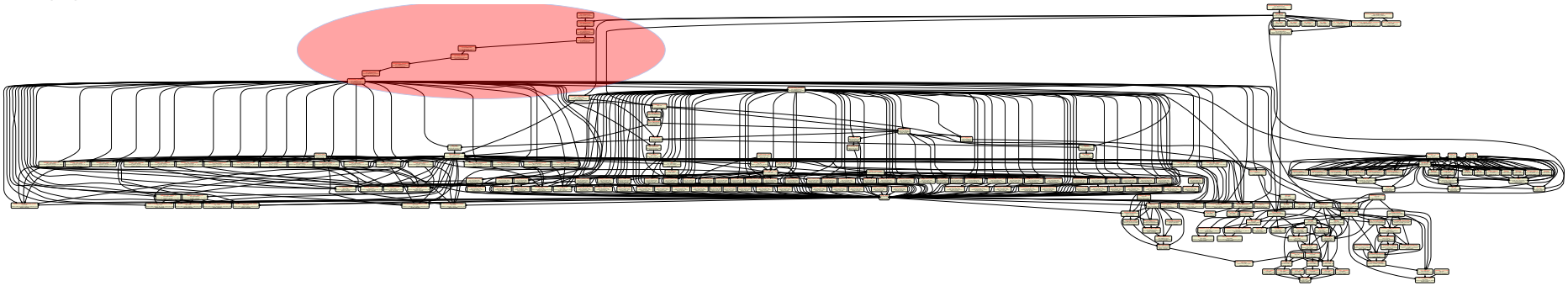


Dependency Resolution

CMSSM:



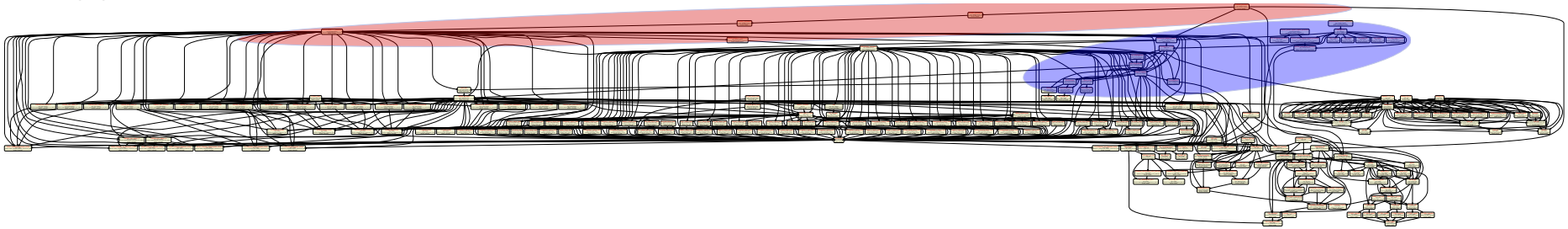
MSSM7:



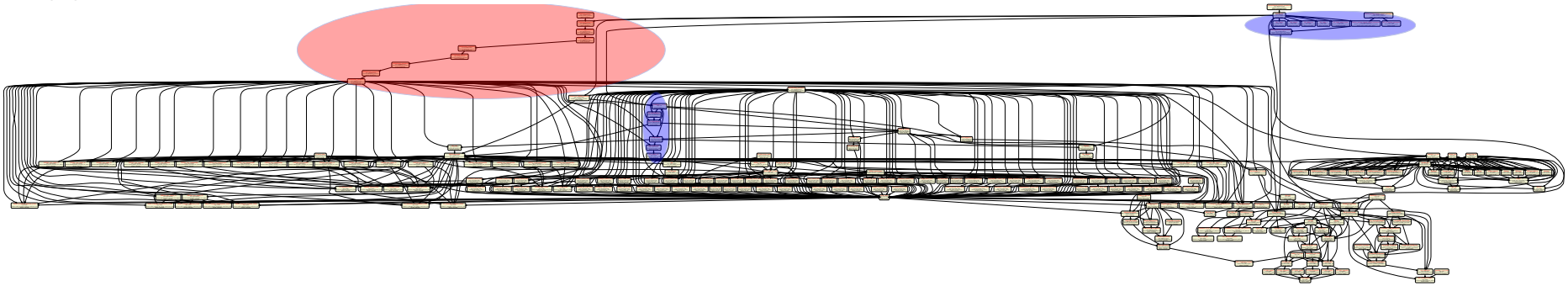
Red: Model parameter translations

Dependency Resolution

CMSSM:



MSSM7:



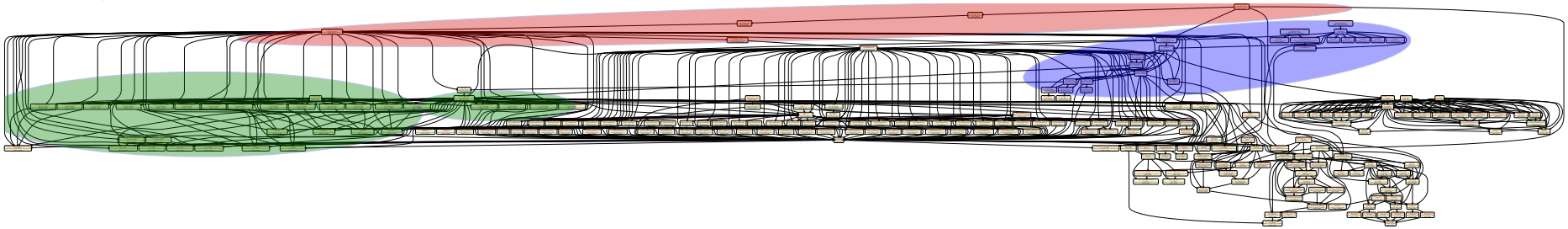
Red: Model parameter translations

Blue: Precision calculations

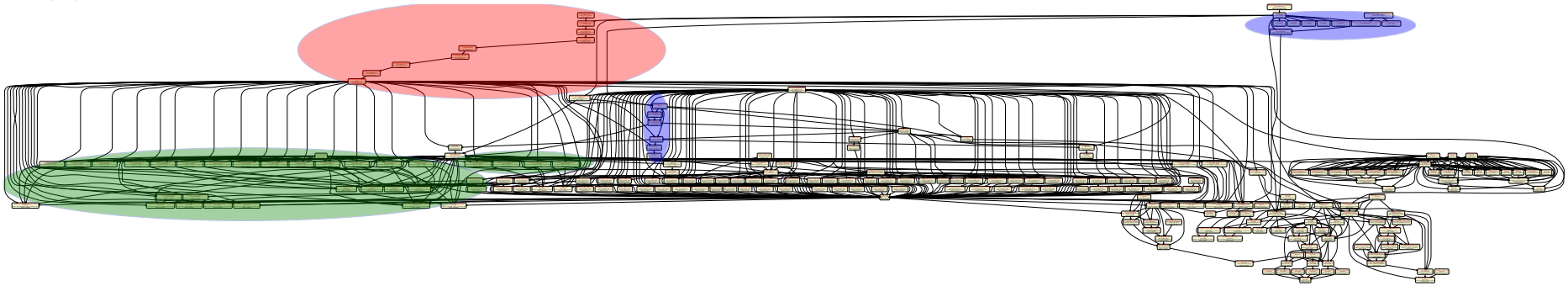
Imperial College London

Dependency Resolution

CMSSM:



MSSM7:



Red: Model parameter translations

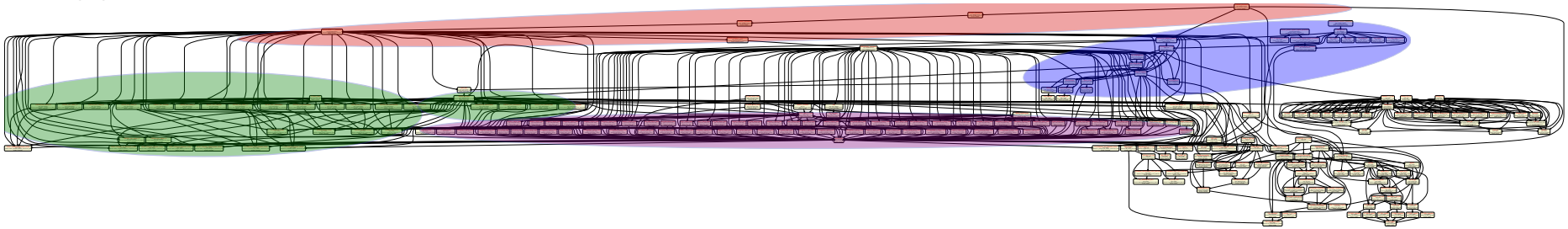
Blue: Precision calculations

Green: LEP rates+likelihoods

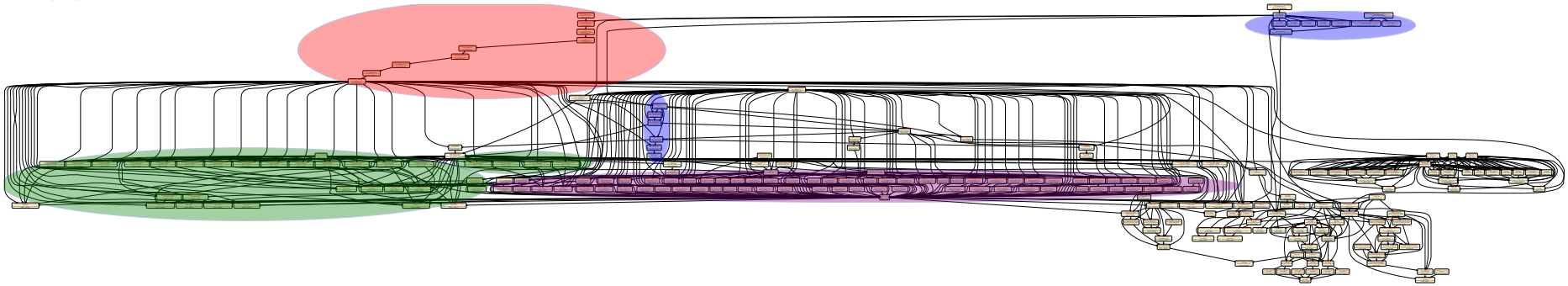
Imperial College London

Dependency Resolution

CMSSM:



MSSM7:

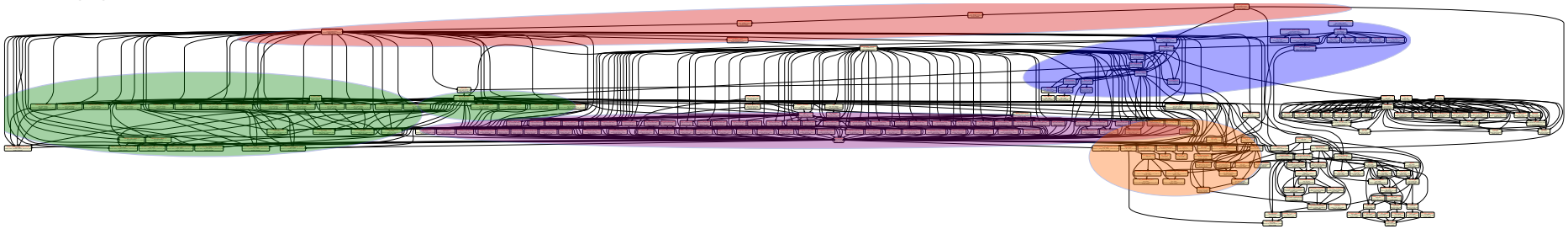


Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays

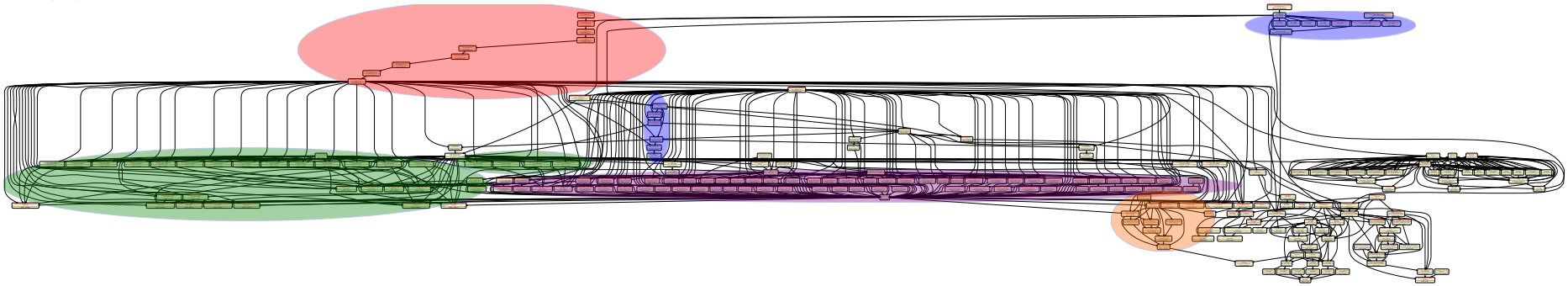
Imperial College London

Dependency Resolution

CMSSM:



MSSM7:

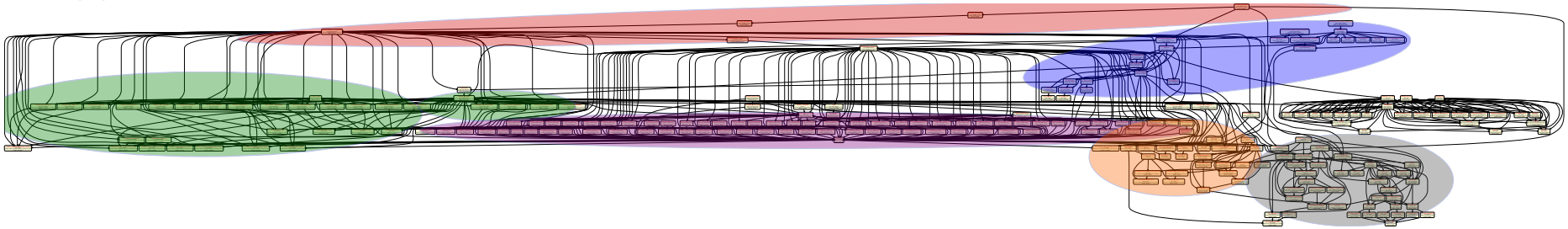


Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods

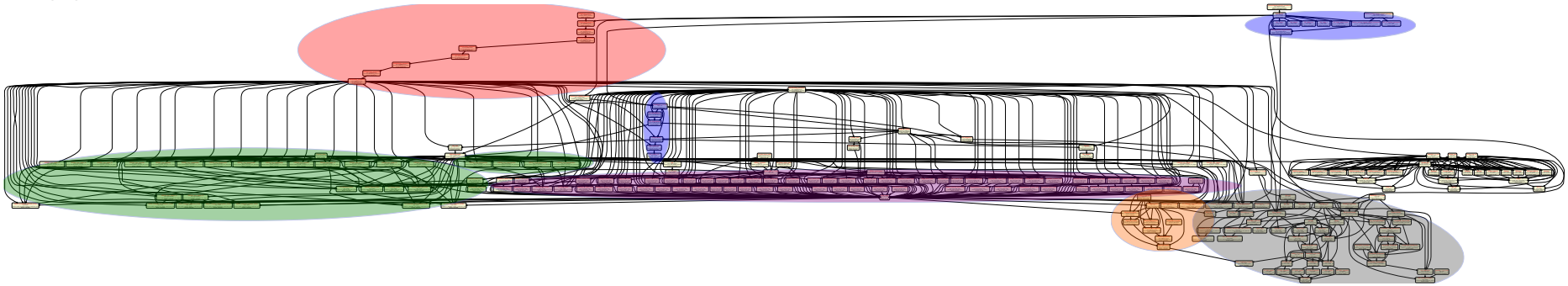
Imperial College London

Dependency Resolution

CMSSM:



MSSM7:

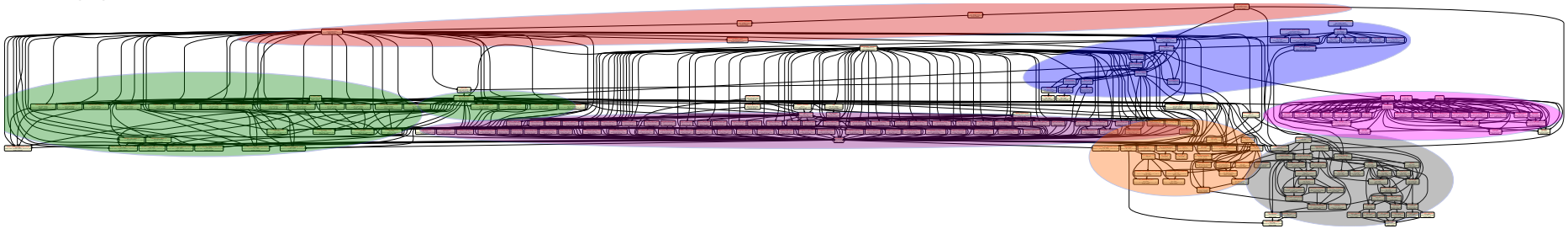


Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
Grey: DM direct, indirect and relic density

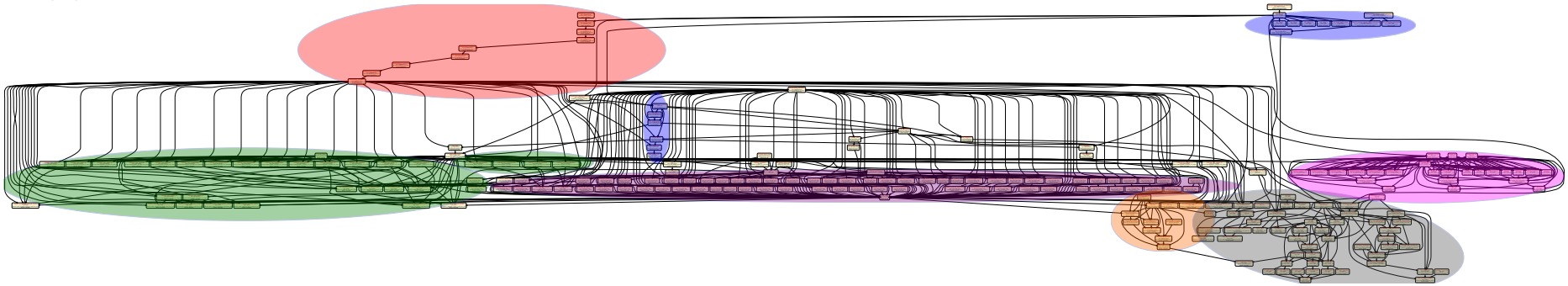
Imperial College London

Dependency Resolution

CMSSM:



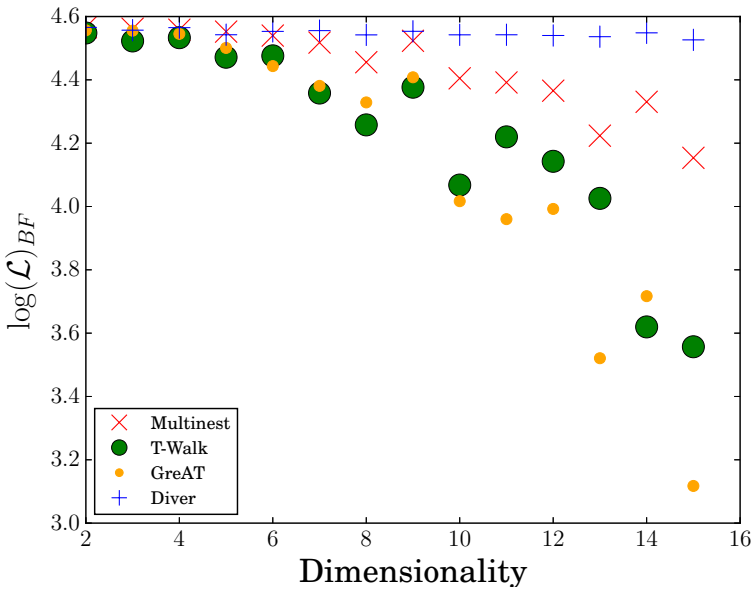
MSSM7:

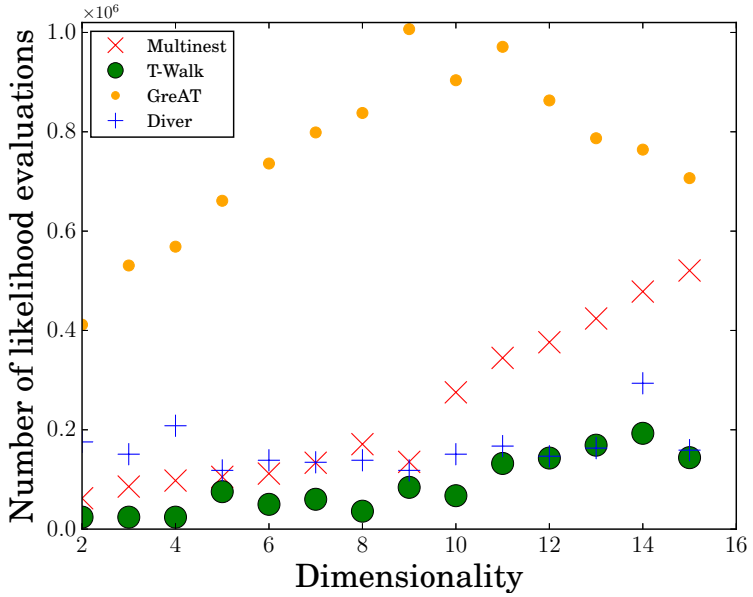


Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
Grey: DM direct, indirect and relic density
Pink: Flavour physics

Imperial College London

Extensive scanner tests on scalar singlet model with different numbers of nuisance paramters





Diver scales far better with dimensionality than MultiNest or other scanners

Imperial College London

Expansion: adding new observables and likelihoods

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)

Imperial College London

Expansion: adding new models

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.

Imperial College London

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
```

Imperial College London

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)

```
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
```

Imperial College London

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
```

Imperial College London

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
```

Imperial College London

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
```

Imperial College London

Other nice technical features

- **Scanners**: Nested sampling, differential evolution, MCMC, t-walk...
- Mixed-mode **MPI + openMP** parallelisation, mostly automated $\rightarrow$ scales to 10k+ cores
- 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, HDF5, ...)
- available as docker plugin or vagrant virtual machine
- more more more...

[1] if a backend won't compile/crashes/kills your cat, blame the authors (not us... except where we **are** the authors...)

Imperial College London

DarkBit overview