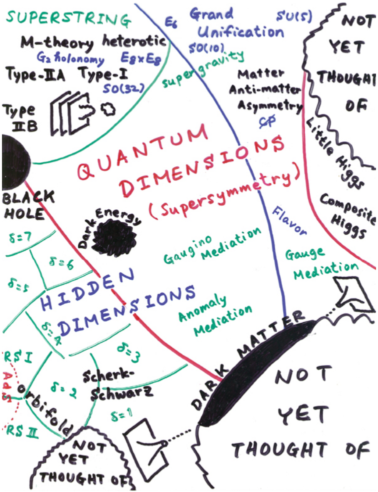
SCYNet

Susy Calculating Yields Network

Philip Bechtle, Sebastian Belkner, Daniel Dercks, Matthias Hamer, Tim Keller, Michael Krämer, Björn Sarrazin, Jan Schütte-Engel, Jamie Tattersall

Institute for Theoretical Particle Physics and Cosmology
RWTH AACHEN UNIVERSITY

Investigating theories beyond the Standard Model

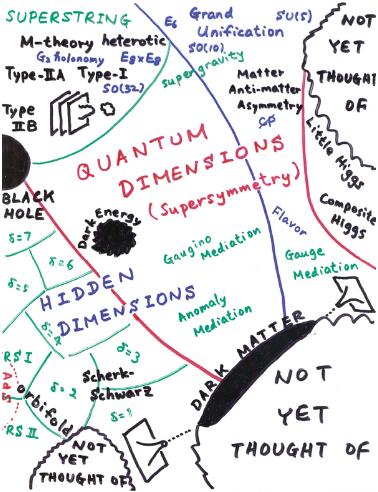


H. Murayama

Theorists have been busy!

- Many new ideas in the last $\sim$50 years
- Very hard (impossible) to fully cover this space

Investigating theories beyond the Standard Model



H. Murayama

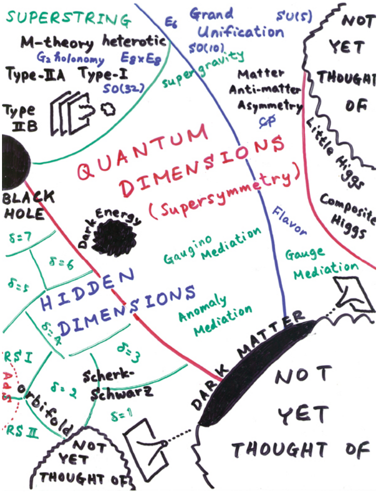
Theorists have been busy!

- Many new ideas in the last ~50 years
- Very hard (impossible) to fully cover this space

Experimentalists as well!

- EW precision, Higgs, LHC, Dark Matter, Dark energy, Inflation, Neutrinos, Proton decay....

Investigating theories beyond the Standard Model



H. Murayama

Theorists have been busy!

- Many new ideas in the last ~50 years
- Very hard (impossible) to fully cover this space

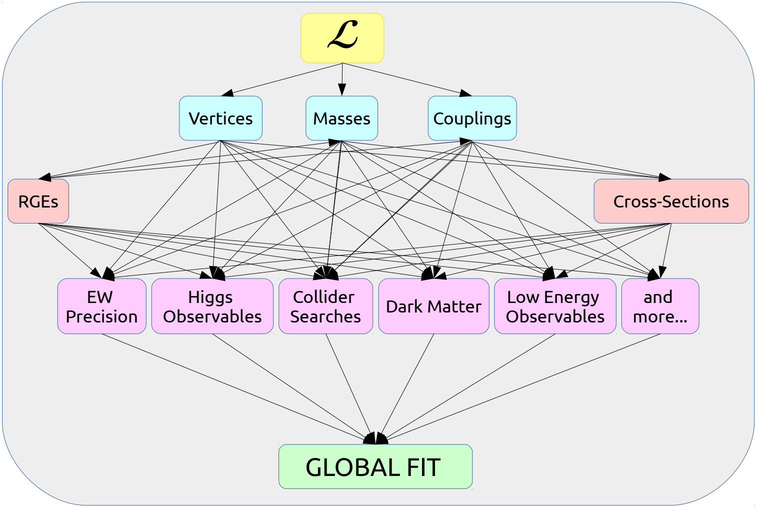
Experimentalists as well!

- EW precision, Higgs, LHC, Dark Matter, Dark energy, Inflation, Neutrinos, Proton decay....

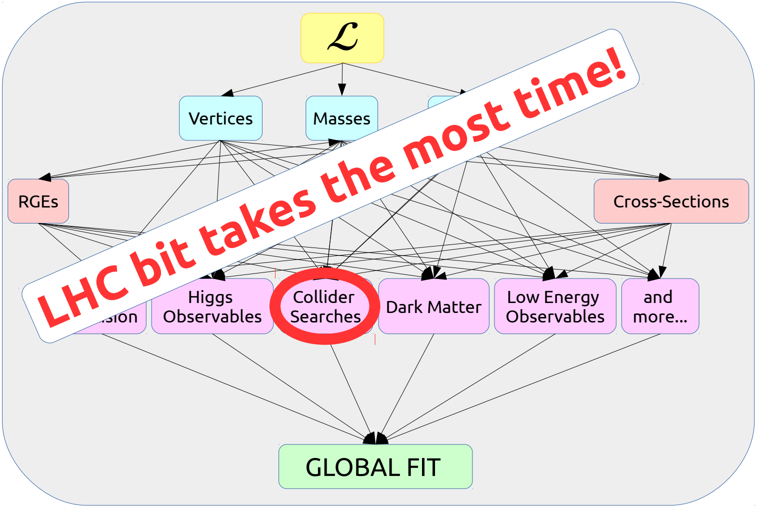
Automatic tools are a must

- Let the computer do the hard work!

Automatic theory testing



Automatic theory testing



Simulation Program Flow

Matrix Element Calculation
(E.g MadGraph)

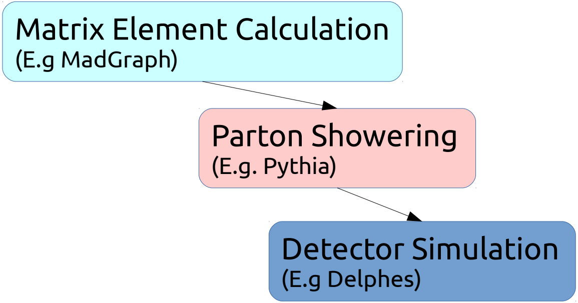
Simulation Program Flow

Matrix Element Calculation
(E.g MadGraph)

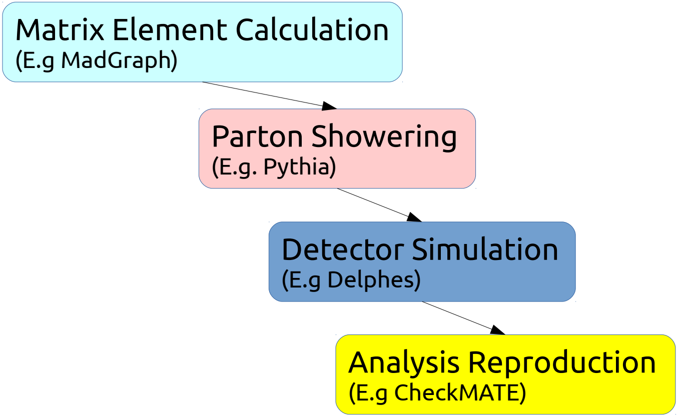
↓

Parton Showering
(E.g. Pythia)

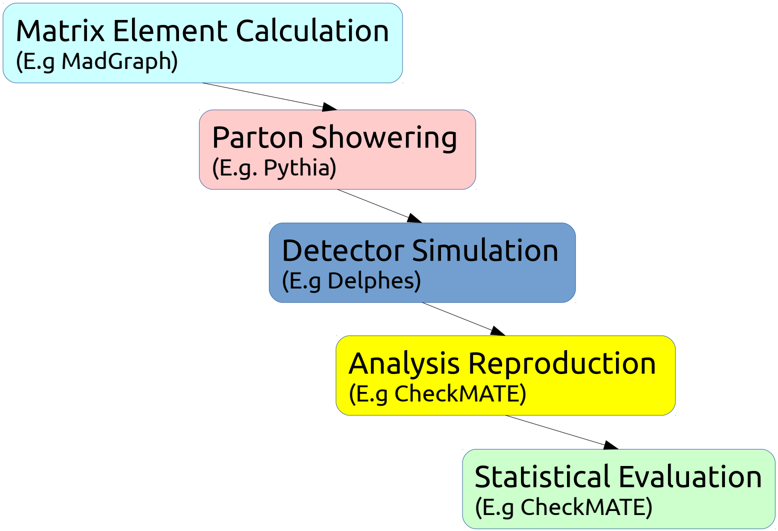
Simulation Program Flow



Simulation Program Flow



Simulation Program Flow



Why do we need a neural network?

Want to explore a high dimensional models

- E.g. Generalised SUSY $\rightarrow$ pMSSM-11 (or 19)
- 11 (or 19) free parameters
- Each individual parameter point requires $\sim$hrs CPU

Why do we need a neural network?

Want to explore a high dimensional models

- E.g. Generalised SUSY $\rightarrow$ pMSSM-11 (or 19)
- 11 (or 19) free parameters
- Each individual parameter point requires $\sim$hrs CPU

Idea $\rightarrow$ Use neural network to learn parameter space

- Sample as many points as feasible (200000 in this case)
- Use net for regression in many dimensions

Why do we need a neural network?

Want to explore a high dimensional models

- E.g. Generalised SUSY $\rightarrow$ pMSSM-11 (or 19)
- 11 (or 19) free parameters
- Each individual parameter point requires $\sim$hrs CPU

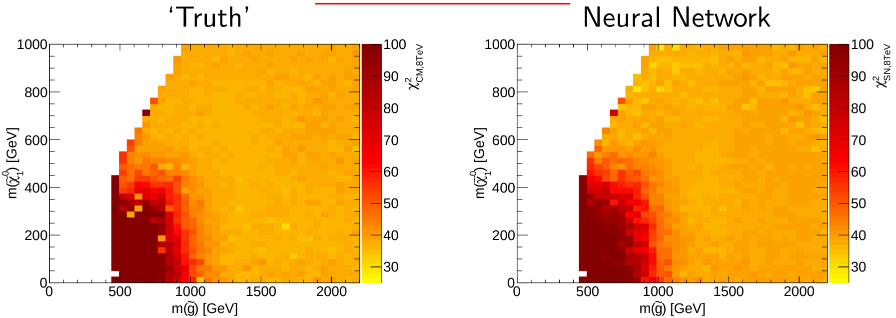
Idea $\rightarrow$ Use neural network to learn parameter space

- Sample as many points as feasible (200000 in this case)
- Use net for regression in many dimensions

Two strategies

- Direct approach $\rightarrow$ Model parameters are network input
- Reparametrised $\rightarrow$ First derive physical parameters

Direct Approach



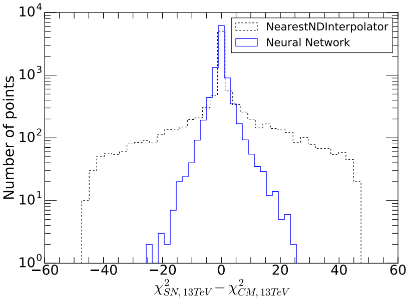
Profile minimum χ^2 at each point

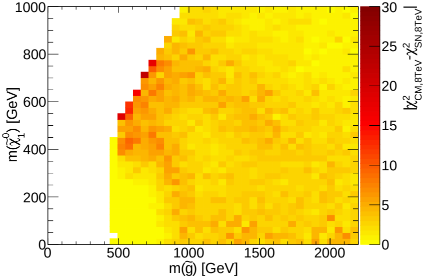
- Each square is an individual model

Neural net reproduces general features well

- Individual models can be poorly predicted
- Variation in true result due to limited validation points

Performance of the net





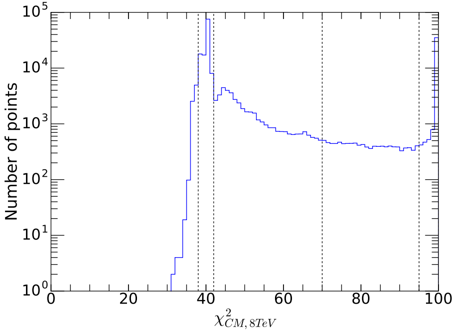
Net is vast improvement over nearest neighbour interpolator

- Error$(\chi^2_{SN}) = 1.5$ vs Error$(\chi^2_{Int}) = 6.2$

Net works well for small or high masses

- Essentially a classification problem
- Transition and especially compressed region are concerning

Rare Target Learning Problem



χ^2 range	Error $(\chi^2_{SN} - \chi^2_{CM})$
0 - 53.5	1.9
53.5 - 56	0.7
56-70	3.7
70-95	6.7
95-100	1.0

Most interesting area is transition region

- Unfortunately net performs worst here

Problem is lack of training data here

- Future work will use (more) intelligent sampling strategy

Reparameterisation

Can reparametrising model help?

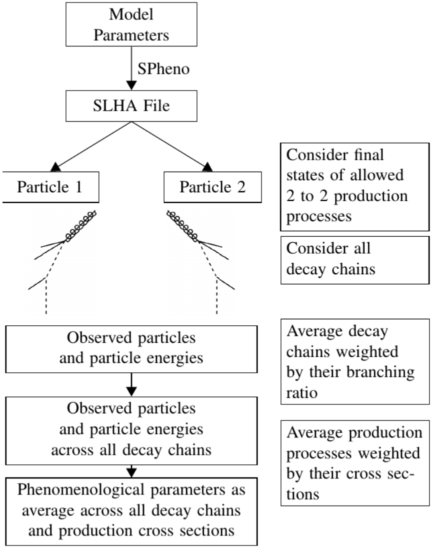
- Try physically motivated parameters

New parameters

- Average particle multiplicity (+ standard deviation)
- Average maximum energy for each particle type (+ standard deviation)
- 56 inputs in total

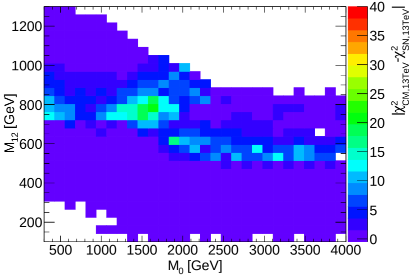
Accuracy very similar

- Can be applied to arbitrary models

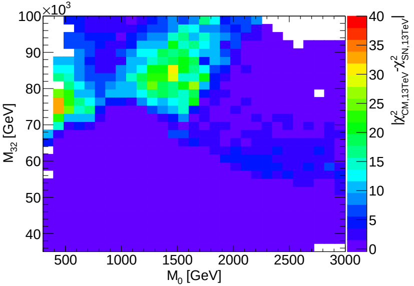


Reparameterisation Results

CMSSM (4 parameters)



AMSB (3 parameters)

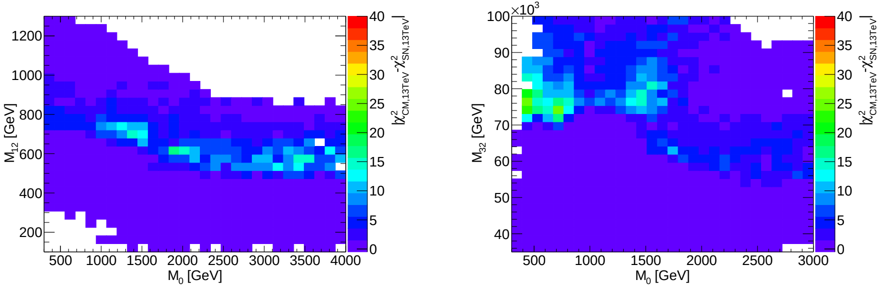


Aim is to test on previously unseen models

- Nets fail drastically in particular regions of parameter space
- These models are NOT a subset of pMSSM-11
- Can we understand physically why?

Improved Reparameterisation

CMSSM (4 parameters) AMSB (3 parameters)



Difference is a previously unseen SUSY mass spectra

- 100% $\tilde{g} \rightarrow \tilde{t}t$ decay in this model
- Net has never seen points that always produce 4 top quarks

Training with new points

- Significant improvement in performance

Conclusion

Automatic model testing is now a reality

- LHC comparison is the (computational) bottleneck

Used neural nets to interpolate high dimensional theory space

- Promising results already delivered
- Improvements in accuracy still required

Reparamerised net also tested

- Possibility to use net in a model independent fashion

Rare training learning problem is the obvious place to improve

- Focus sampling on the LHC important region

Backup

Neural Network parameters

Direct approach

- Developed with TensorFlow
- Four hidden layers
- 300 neurons per layer
- Adam minimiser
- Hyperbolic tangent activation function

Reparametrised approach

- Developed with TensorFlow
- Nine hidden layers
- 500 neurons for first layer, 200 thereafter
- nAdam minimiser
- Rectified linear unit activation function

pMSSM-11 parameters and scan ranges

Parameter	Scan range
M_1	[-4000,4000] GeV
M_2	[100,4000] GeV
M_3	[-4000,-400]$\cup$[400,4000] GeV
$m_{\tilde{q}_{12}}$	[300,5000] GeV
$m_{\tilde{q}_3}$	[100,5000] GeV
$m_{\tilde{l}_{12}}$	[100,3000] GeV
$m_{\tilde{l}_3}$	[100,4000] GeV
m_{A^0}	[0,4000] GeV
A^0	[-5000,5000] GeV
μ	[-5000,-100]$\cup$[100,5000] GeV GeV
$\tan\beta$	[1,60]

Calculating χ^2 for one SR

Likelihood function

$$\mathcal{L}(N_E|\nu_S,\nu_{SM},\lambda) = \frac{e^{-\lambda}\lambda^{N_E}}{N_E!} \times \frac{1}{\sqrt{2\pi}}e^{-\frac{\nu_S^2}{2}} \times \frac{1}{\sqrt{2\pi}}e^{-\frac{\nu_{SM}^2}{2}}$$

$$\lambda(\nu_S,\nu_{SM},\mu,S) = S\mu e^{\frac{\Delta S}{S}\nu_S} + N_{SM}e^{\frac{\Delta N_{SM}}{N_{SM}}\nu_{SM}}$$

with $\Delta S = \sqrt{(\sigma_S^{stat})^2 + (\sigma_S^{sys})^2}$ and
$\Delta N_{SM} = \sqrt{(\sigma_{N_{SM}}^{stat})^2 + (\sigma_{N_{SM}}^{sys})^2}$.

$$H_0 : \mu = 1, \qquad H_1 : \mu \neq 1$$

$$\mathcal{L}_C := \max_{\nu_S,\nu_{SM}\in\mathbb{R}} \mathcal{L}(\mu = 1, S = N_{SM}, \nu_{SM}, \nu_S)$$

$$\mathcal{L}_G := \max_{\mu,\nu_S,\nu_{SM}\in\mathbb{R}} \mathcal{L}(\mu, S = 1, \nu_{SM}, \nu_S)$$

$$PLR := \frac{\mathcal{L}_C}{\mathcal{L}_G}, \quad q_\mu := -2ln(PLR)\ \chi^2 \text{ distributed}$$