Improving Background Estimation for Di-Higgs Searches with ATLAS

PHYS 437B Presentations

13 January, 2020

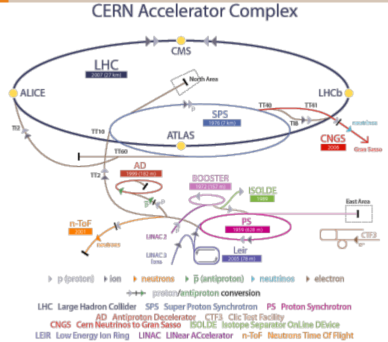
Callum McCracken

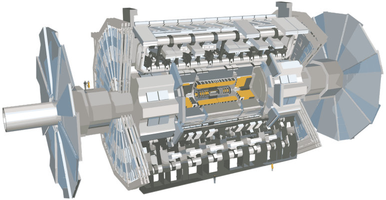
Supervisor: Maximilian Swiatlowski

Co-Supervisor: Eduardo Martin-Martinez

Collaborators: Todd Seiss, Mel Shochet

Overview: Higgs Research with ATLAS





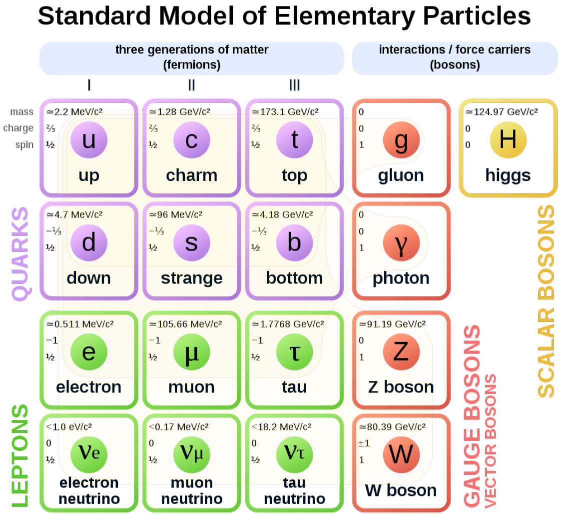


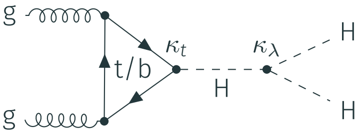
Image sources: Accelerator Complex, ATLAS, Standard Model

The Larger Project: Measuring the Higgs Self-Coupling

Relevant section of the SM Lagrangian for Higgs potential:

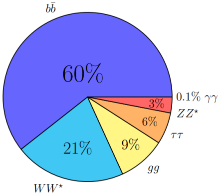
$$V(\phi) = -\mu^2\phi^2 + \lambda\phi^4 + \dots \text{ Taylor exp. at min } \rightarrow V_T(\phi) = -\frac{\mu^4}{4\lambda} + \frac{\sqrt{2}\mu^3}{\lambda}\phi - 4\mu^2\phi^2 + 2\sqrt{2\lambda}\mu\phi^3 + \dots$$

Constant and ϕ terms: can eliminate with change of coordinates, ϕ^2: mass term,
ϕ^3: self-interaction or self-coupling term, not well constrained
(current best: $\kappa_\lambda = (2\sqrt{2\lambda}\mu)/(2\sqrt{2\lambda}\mu)_{\text{SM}}$, $\kappa_\lambda \in [-2.3, 10.3]$ at 95% confidence)



To find κ_λ we need *HH* events, and we can find them using jets!

Background: Jets and Pairing



$H \to 60\% b\bar{b} \to 2 \times b$ hadrons $\to 2 \times b$-jets

- Jets are collections of particles with appx. the same direction
- ATLAS can't directly detect H or b. Instead, use b-jets, which can be directly detected (using secondary vertices)
- b-jet detection is not a perfect process (hence 437A report), and neither is pairing – identifying which jets came from which H

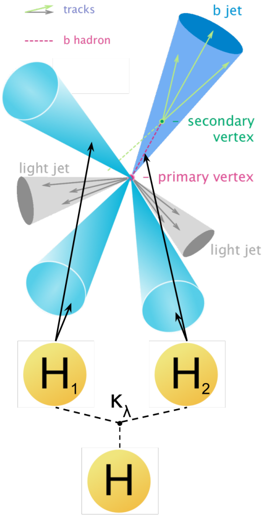
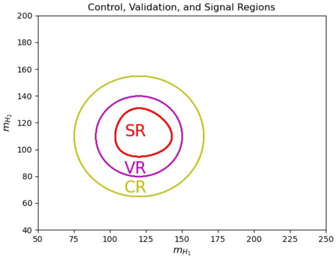
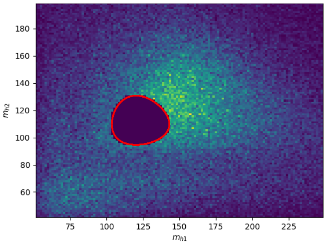


Image sources: branching ratios, b-tagging image adapted from here

Our Project: Background Modelling

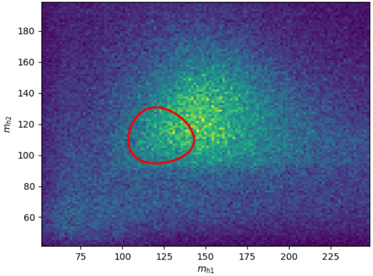
- Mass plane: reconstructed (m_{H_1}, m_{H_2}) values, m_{H_1} has higher transverse momentum
- We expect a peak around $(125, 125)$ (all masses in GeV)
- The Problem: how to estimate background around peak?
  Large source of error $(|\kappa_\lambda| < 7 \rightarrow |\kappa_\lambda| < 9)$
- Signal Region (SR): blinded to reduce study bias
- Control Region (CR): for calibrating background estimation models
- Validation Region (VR): for testing background models





Current Approach: $2b$RW

- All jets are similar to a rough approximation
- $2b$ data: uses 2 b-jets and 2 other jets
- Similar to $4b$ data outside of SR, but no peak
- $2b$RW: derive a scaling ("ReWeighting") function outside of SR, apply inside
- Provides a good first background estimate
- Assumes RW function applies in SR, may be false
- This project: is there a better approach?

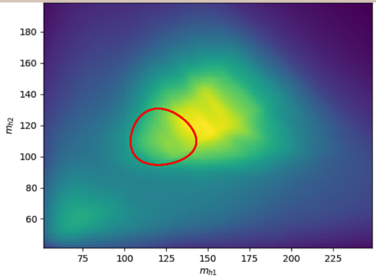


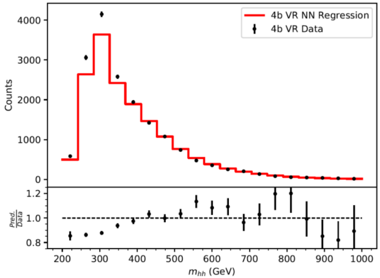
Note

Don't forget $2b$ data is physically different from $4b$ and $2b$RW are not **real** SR values. $2b$RW is thought to be correct within around $\pm 10\%$.

New Approach: Neural Network

- Given enough data, neural networks can learn arbitrary functions
- Goal: reproduce 2bRW $\pm$10% using only 4b data
- Inputs: $(m_{H_1}, m_{H_2}, m_{HH})$, output: $P(m_{H_1}, m_{H_2})$
- Initial model: layers (10,50,50,50), 100 epochs
- Generally good-looking mass plane predictions, 2bRW agreement is not great though





Histogram image from Todd Seiss. Also, this is 4b VR, most plots will use 2bRW SR

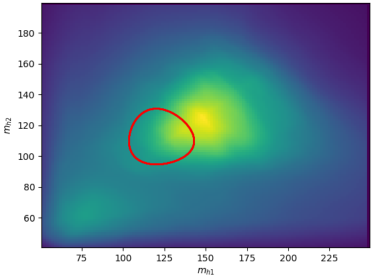
Neural Network Optimization

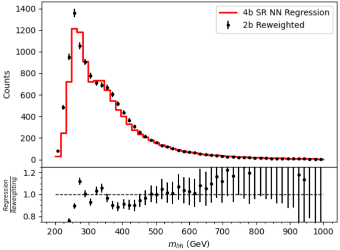
What if we...

- Further optimize network hyperparameters?
- Add more bins?
- Add other variables (e.g. more masses, NTag)?
- Smooth the data (fit a polynomial, KDE, ...)?

Overall impression: large improvements unlikely from non-drastic changes in approach

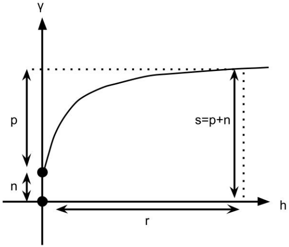
- (note: images are from different models, to show how many of them had similar performance to before)



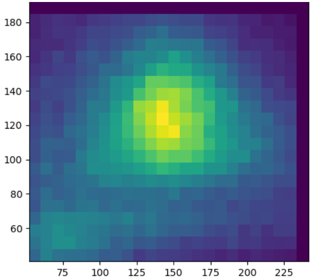


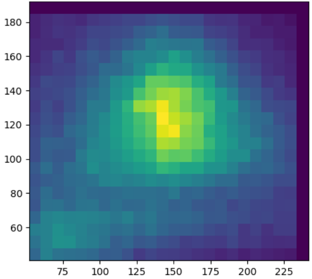
New Approach: Gaussian Process Regression

- Inputs: m_{H_1}, m_{H_2}, output: $P(m_{H_1}, m_{H_2})$
- Also gives estimator variance ($\exists$ assumptions)
- Relies on variogram (ideally could calculate, given finite data must guess)



- Tried a bunch of different "best-guess" variograms, e.g. linear and exponential:



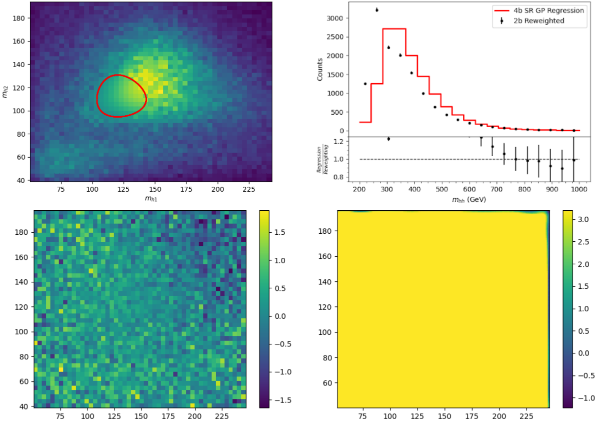


GPR Optimization

- Tested many variograms
- Selected low & flat variance
- Best ones still high relative to prediction ($\sim 1\times$)
- Tried 3D GPR, increases computation time too much

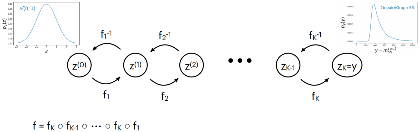
Impressions:

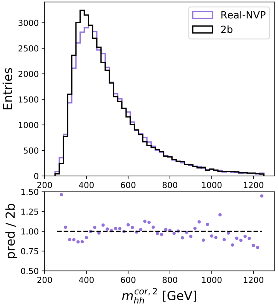
- Good mass plane, less-good histogram
- Interesting variances are always so large
- Figures: Gaussian variogram ($s = 800, r = 160, n = 10^{-8}$)



Flow Models

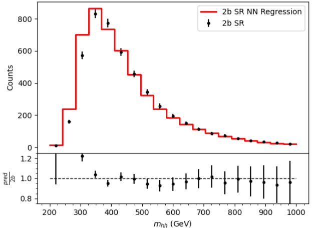
- The idea of flows: build arbitrary function from base function + invertible transformations
- R-NVP transformations: fast compute times
- Benefit: significantly smaller model space, can handle more variables
- Code was set up for 2*b* pairAGraph data + figures compare 2*b* SR prediction vs 2*b* SR data
- With larger models, some nice results are possible (e.g. 5-variable model in figure)



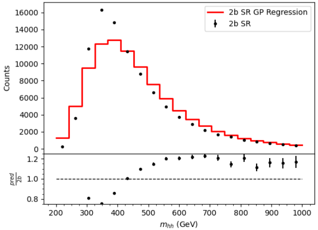


Use $2b$ pairAGraph data on previous methods, compare with a FM with inputs (m_{H_1}, m_{H_2}):

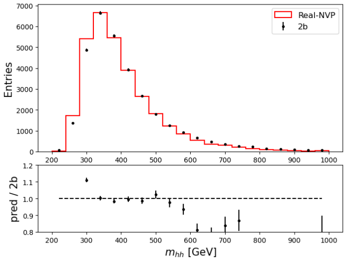
NN:



GPR:



FM:



Conclusion

- Neither NNs or GPR seems to be ready to replace 2*b*RW (at least using the inputs/outputs we tried)
- Flow models with more variables look more promising
- Or potentially other kinds of models (e.g. variational auto-encoders)
- For now, recommend sticking with 2*b*RW

Questions?

Comments?

Any other techniques we should try?

Backup Slides

Link to Google Slides with more details