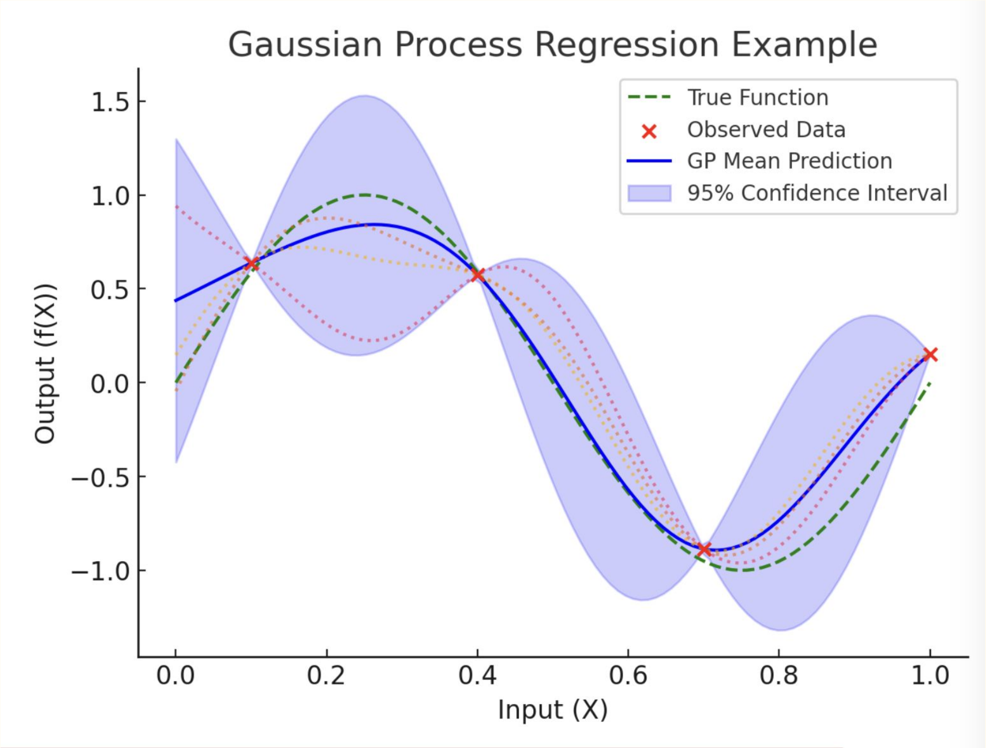
Characterizing HPS Prompt Background with Gaussian Process Regression

Aidan Hsu – Stanford University
Takumi Britt – UCLA

Mentor: Emrys Peets – Stanford University, SLAC

What is Gaussian Process (GP)?

- Probability Model
- Machine Learning
- Dynamic Parameter Input
- Uncertainty Readout



Emrys Peets

How does the GP work?

$\boxed{X \sim N(\mu, \Sigma)}$

X: Random vector – list of n numbers with some variability

μ: Average vector – on average, the i-th entry of X is μ_i

Σ: Covariance matrix

- How much X_i spreads out around μ_i
- How two different coordinates X_i and X_j move together

How does the GP work?

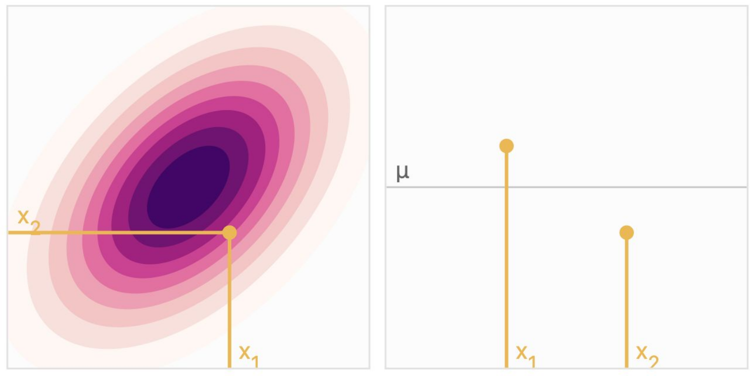
$\boxed{X \sim N(\mu, \Sigma)}$ – describes a "blob" of probability centered at μ, with shape and orientation given by Σ

This "blob" is an **n-dimensional ellipsoid** (when evaluating the Gaussian distribution density function at a constant density)

Given by principal axes:

Eigenvectors (dir stretch) give **directions** where data varies independently

Eigenvalues (||stretch||) give **how much** variance along these directions



How does the GP work?

m(x) – mean function

k(x, x') – kernel (covariance function)

In practice:

$X \sim \mathcal{N}(\mu, \Sigma) \rightarrow f(x) \sim \mathcal{GP}(m(x), k(x, x'))$

So if there are inputs x_1, x_2, x_3 then $f(x_1)$, $f(x_2)$, $f(x_3)$ follow a 3D multivariate Gaussian

Can extend to any number of inputs

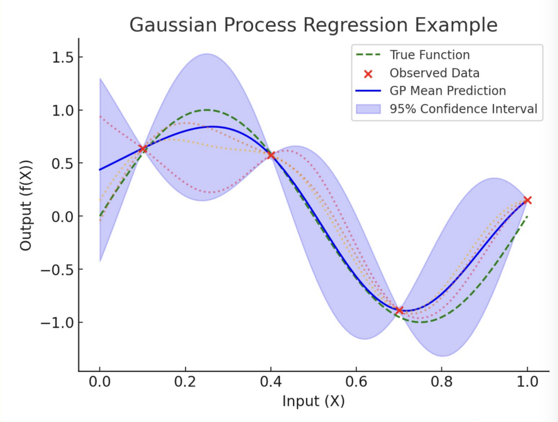
- GP defines a probability over an **infinite-dimensional space** of functions
- Of course, we only ever work with a finite set of inputs in practice

How does the GP work?

Evidently, the model makes confidence bands based on the eigenvectors and eigenvalues

- Eigenvectors point to where data varies **together** (interpolation)
- Smaller eigenvalues mean more confidence and vice versa

The fit is driven by the covariance function (kernel)



Emrys Peets

So what kind of Kernel?

RBF, Linear, White (overfitting?)

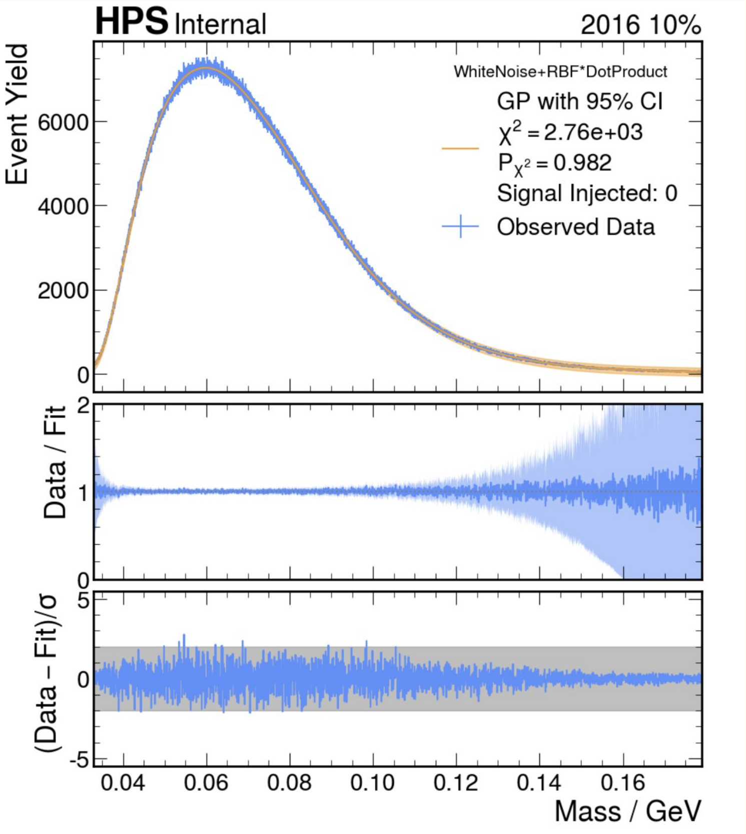
Radial Basis Function:

- Smooth curves
- General purpose GP-regression
- Other reasons?

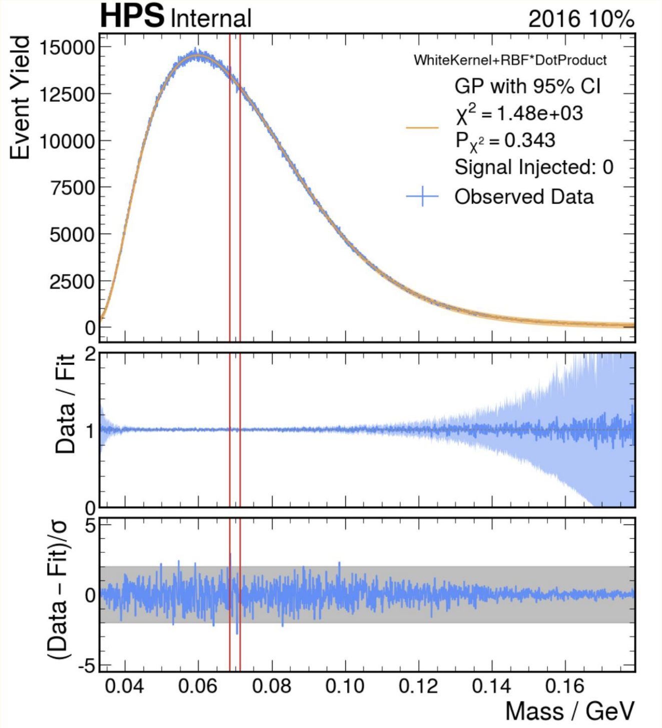
So what kind of Kernel?

- RBF - assume smoothness of data
- Linear - smoothness increases with increasing mass
- Whitekernel - white noise, implements uncertainty in model
- When combining: multiply is like AND, adding is like OR
- Result: WhiteKernel + RBF*Linear
- Very good source of info for GP kernels

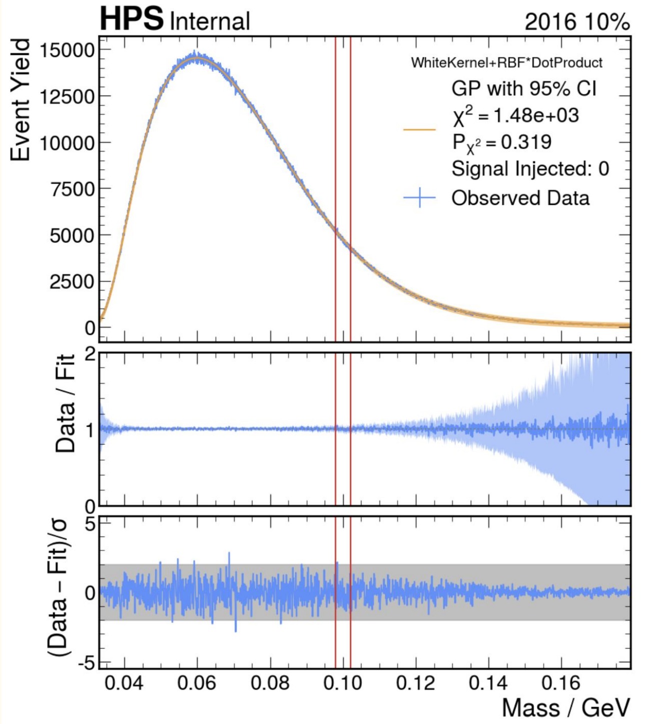
Last Time: Fitting 2016 10%



Blinding 2016 10% (size of blind range is one mass resolution)

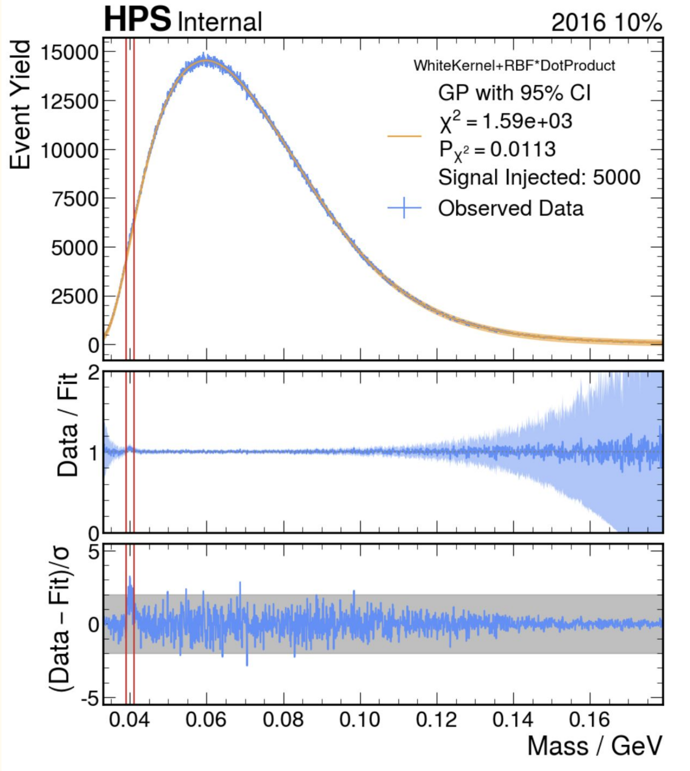


70 MeV

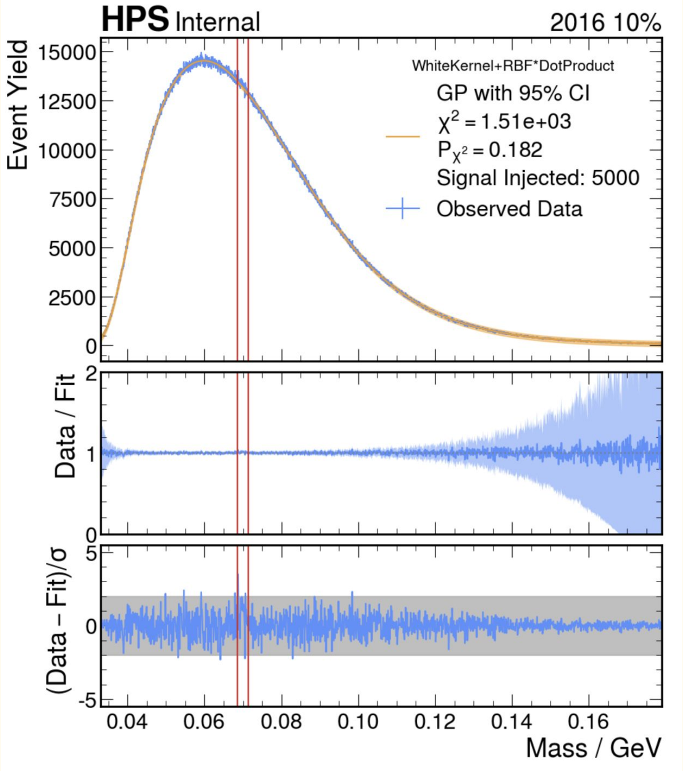


100 MeV

Injecting Signal 2016 10% (5000 events, 1 sigma blind range)

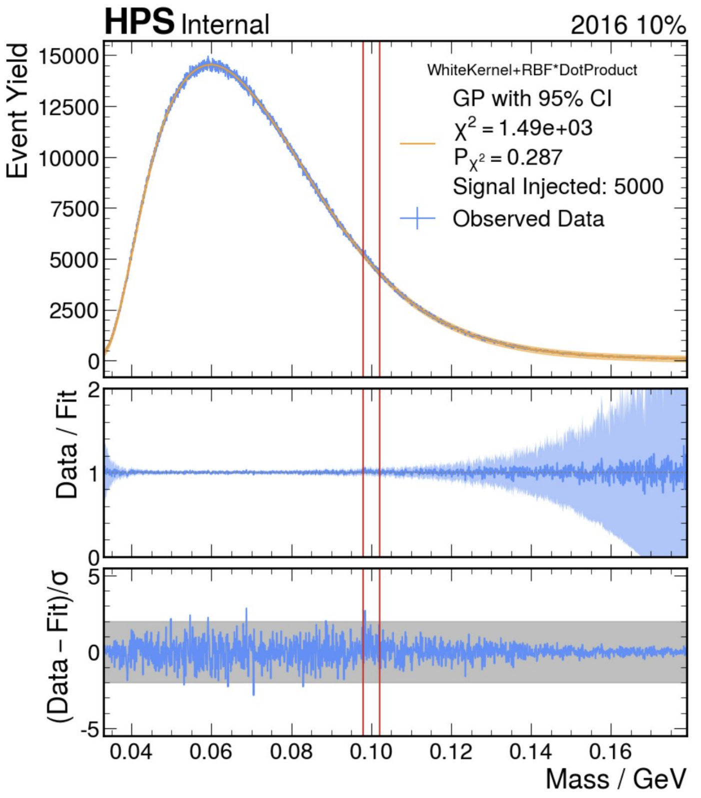


40 MeV



70 MeV

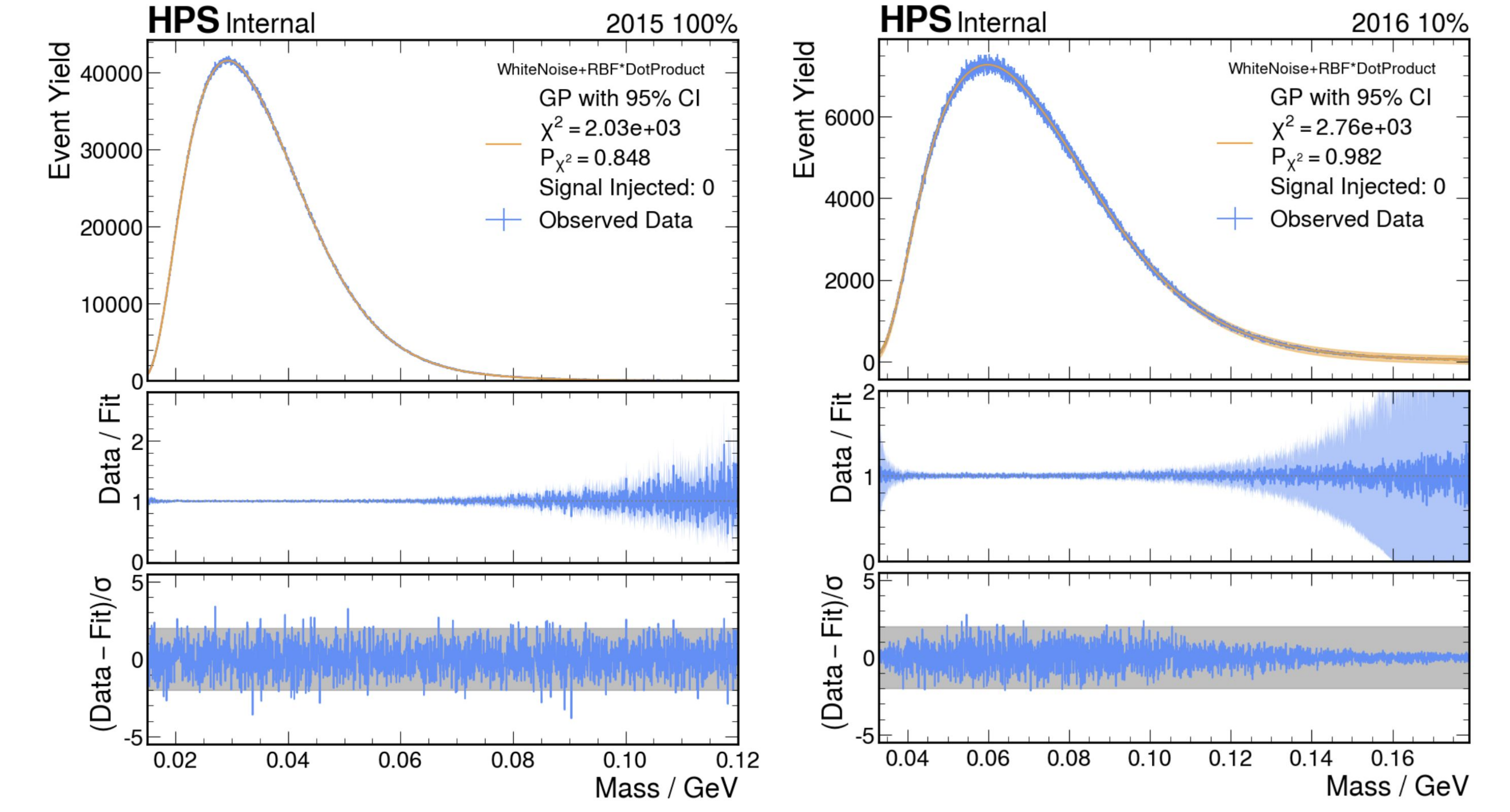
Injecting Signal 2016 10% (5000 events, 1 sigma blind range)



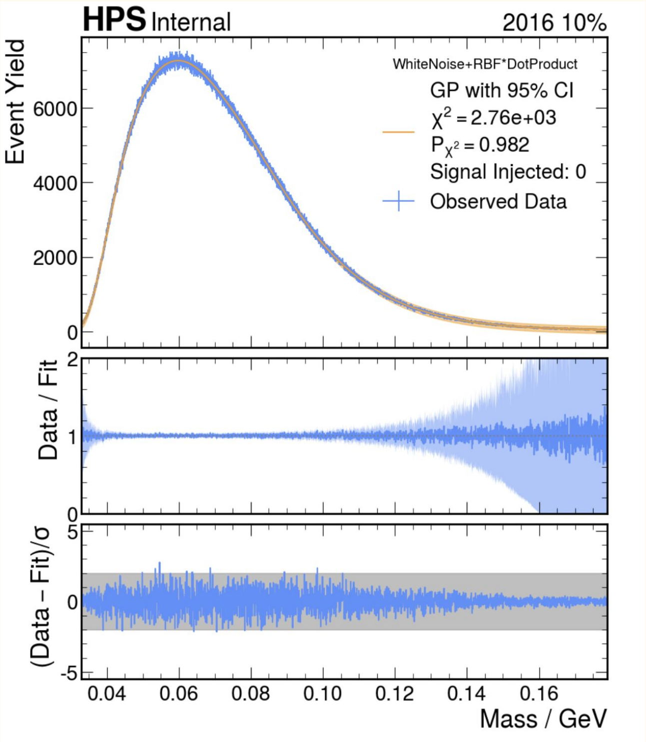
100 MeV

Since Last Time: Fitting 2015 100%

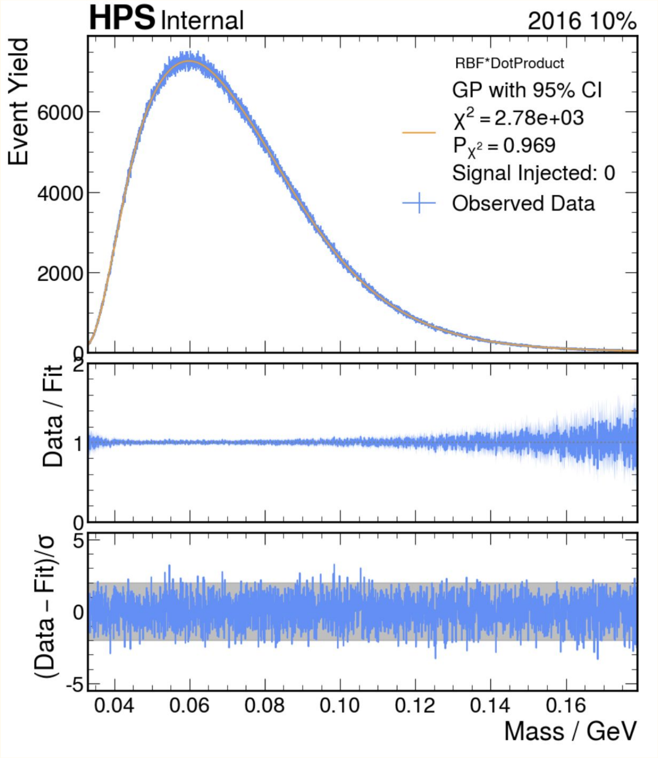
(No rebin plot)



Passing variance as alpha (noise level): 2016 10%

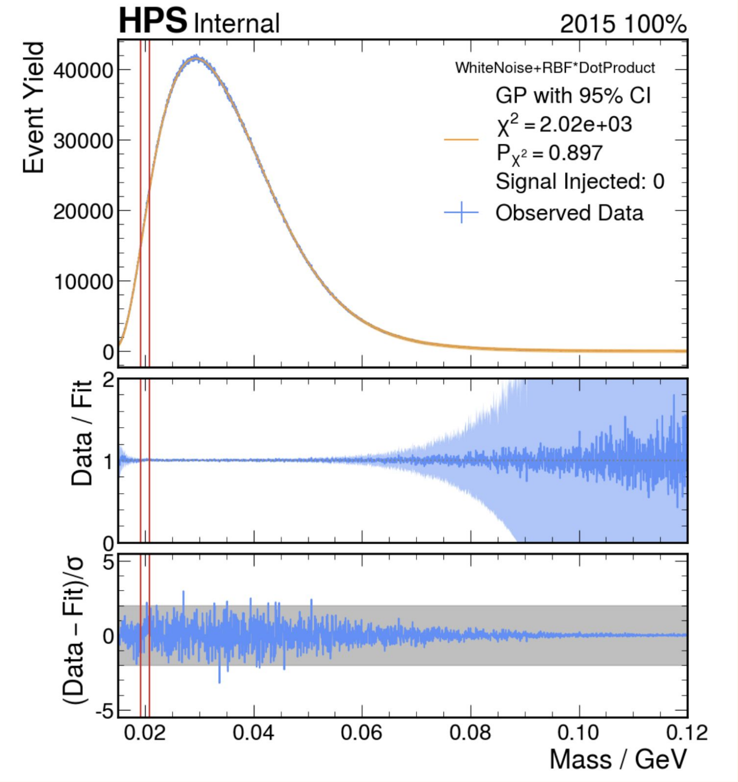


Without alpha as variance

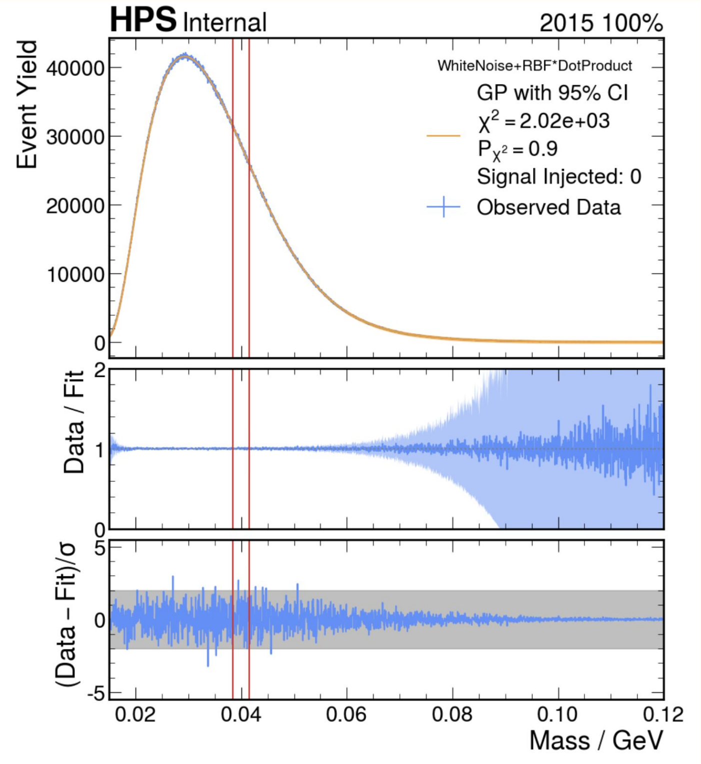


With alpha as variance

Fitting with blind range



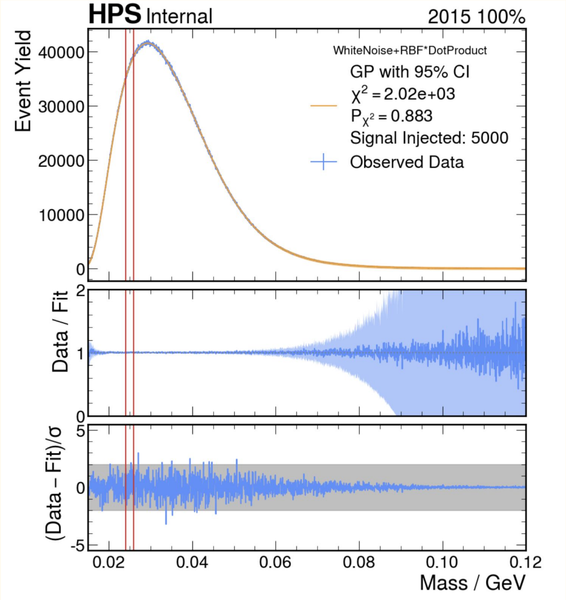
Centered at 20 MeV



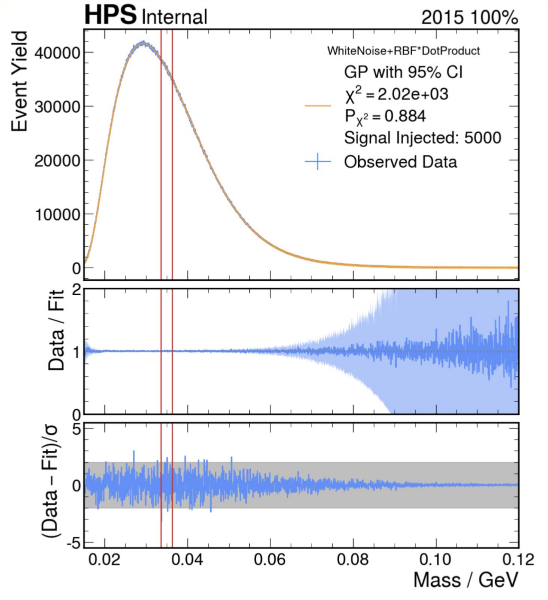
Centered at 40 MeV

Injecting Signal in 2015 100% (POSSIBLY NOT WORKING)

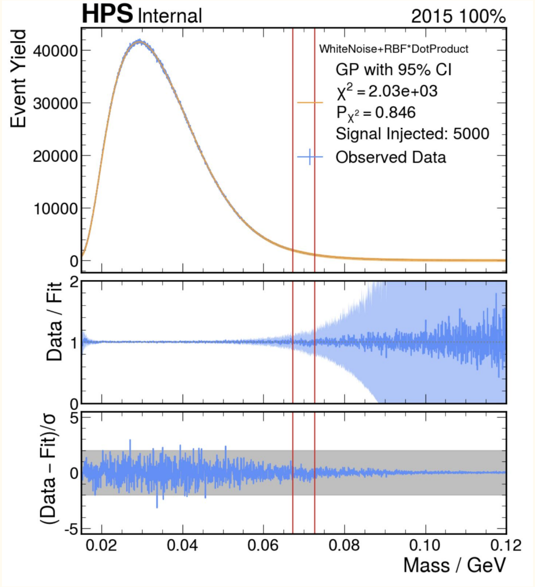
- Same injection procedure as for 2016 10%



25 MeV, 5000 Events



35 MeV, 5000 Events



70 MeV, 5000 Events

Calculating Upper Limits

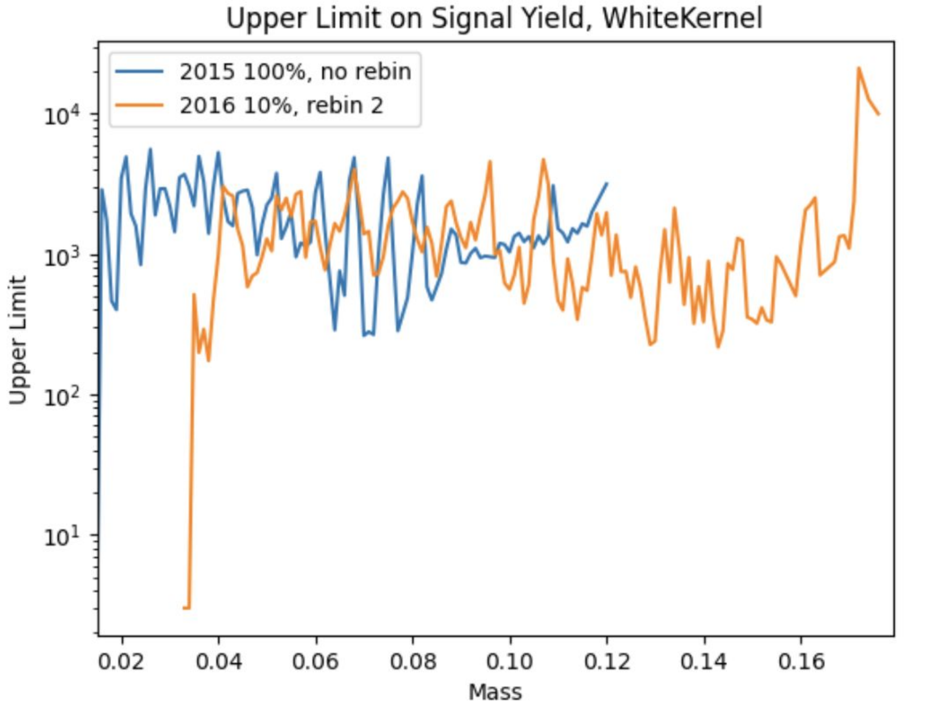
- Calculate the maximum number of expected signal events given the expected bkgd events and the number of events observed. From 'Statistics for Searches at the LHC' by Glen Cowan Section 10. https://arxiv.org/abs/1307.2487v1 . This equation is equivalent to the CLs technique.

$$s_{\rm up} = \frac{1}{2} F^{-1}_{2(n+1)}(p) - b \qquad p = 1 - \alpha(1 - F_{2(n+1)}(2b))$$

where b is the expected number of Poisson background, alpha = 0.05 for 95% confidence, F is the chi-squared cumulative distribution for 2(n+1) degrees of freedom, and n is the observed number of events.

- Easy to implement and helpful for initial testing
- Single signal bin, no shape (requiring any excess within the blinded region to conform to a mass peak shape would presumably improve the limit)

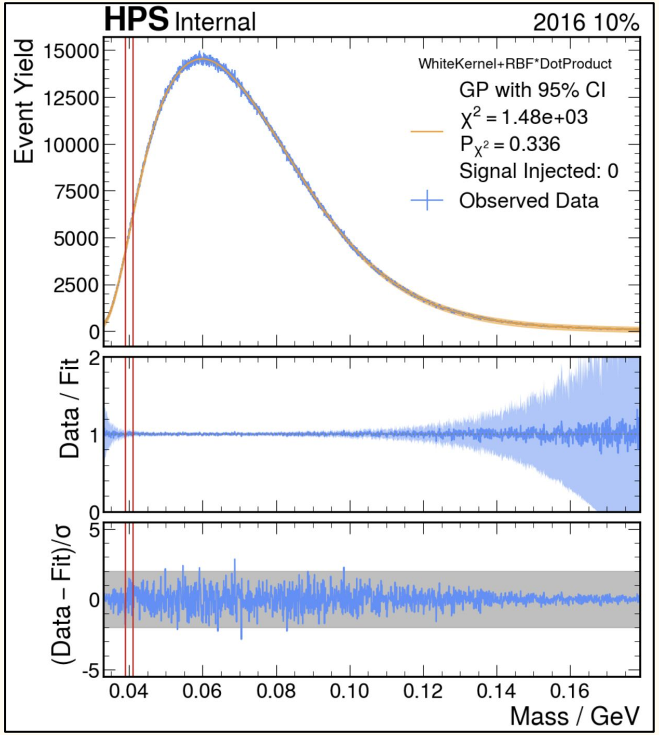
Upper limit calculated at various blind range centers. Essentially, we iterate through masses, producing a fit at which each mass is the center for a blind region, then calculate the upper limit. The size of the blind range is 1 "sigma" in each direction. 1 sigma is twice the mass resolution.



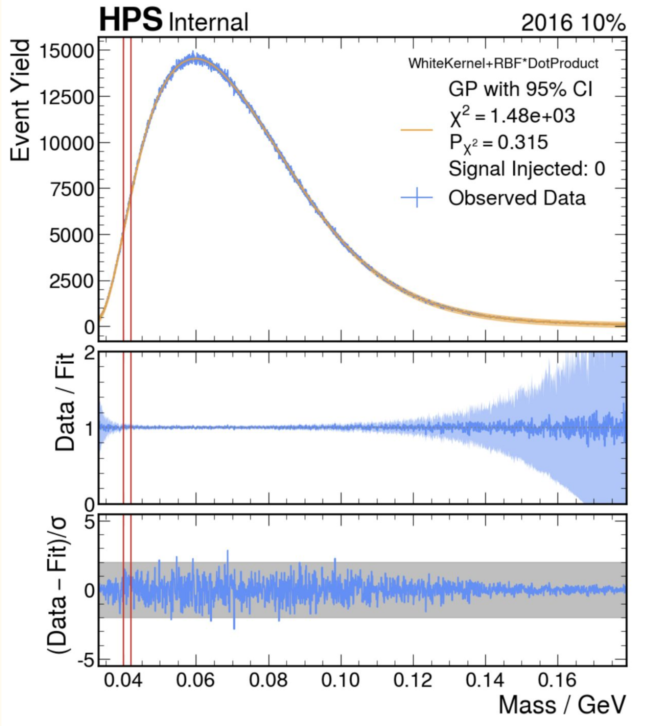
What next?

- “Brazil Bands” - currently not working :(Aidan will make it work soon
- Upper limit plots for 2015 100%
- Compare to Emrys fitting method
- Finish GP paper

Blinding 2016 10% (size of blind range is one mass resolution)



40 MeV



41 MeV