X17 Collaboration Meeting
09/12/2025

X17 Resonance Search

Preliminary Reach Estimates and an Update on Gaussian Process Techniques

Emrys Peets
Joseph Bailey

Stanford University
SLAC National Accelerator Laboratory

Overview

I. Refresher
   - Introductions
   - The Padme-NA64 Target

II. X17 Background Generation Toy Study
   - pyBumpHunter proof of principal

III. Gaussian Process Regression Bump Hunting
   - GP Motivations and recent literature
   - X17-GP Software Package Development

IV. Next Steps

Internal Note: X17 Collaboration Bumphunt Infrastructure and Methodology

Emrys Peets*[1,2], Joseph Bailey[1]

https://www.overleaf.com/read/zpzknkkkqrmk#bef878

Quick Introduction

Emrys Peets

- Stanford Physics PhD Candidate
- Advisors: Tim Nelson (SLAC), Philip Schuster (SLAC, Stanford)

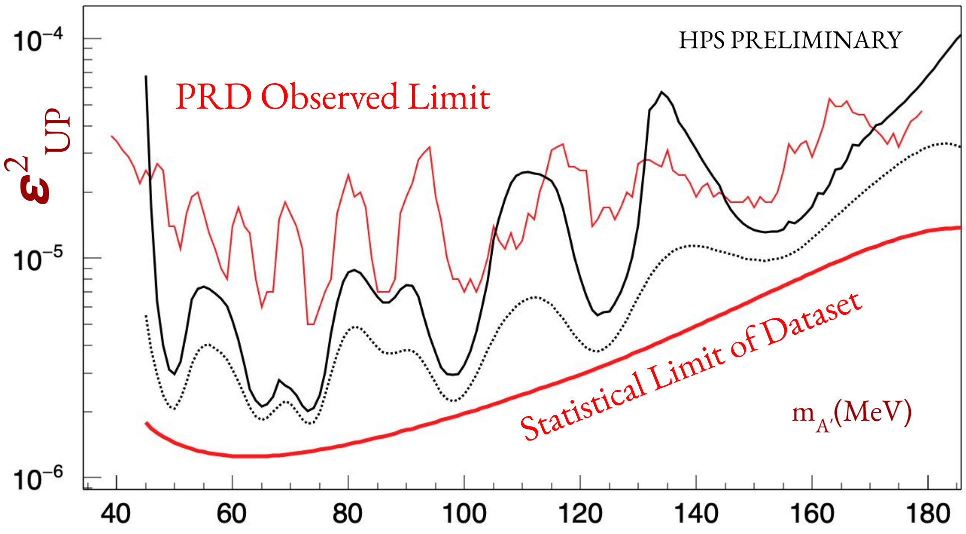
Joseph Bailey

- Stanford Undergraduate
- Experience with Python, ROOT, fast simulation

Experience

- Primary Prompt A` Resonance Search Analyst for Heavy Photon Search Collaboration
- Global Functional Form Fitting Technique as First pass
  - Gaussian Process regression determined to be more reliable and have less implicit bias
- Mentoring multiple students on GP bumphunt techniques

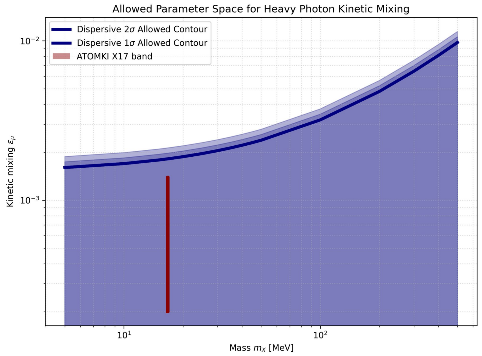
Observed ε^2 Upper Limits ***Proof of Principle***

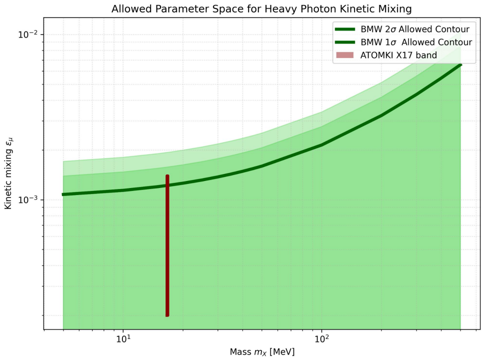


Personal Interest

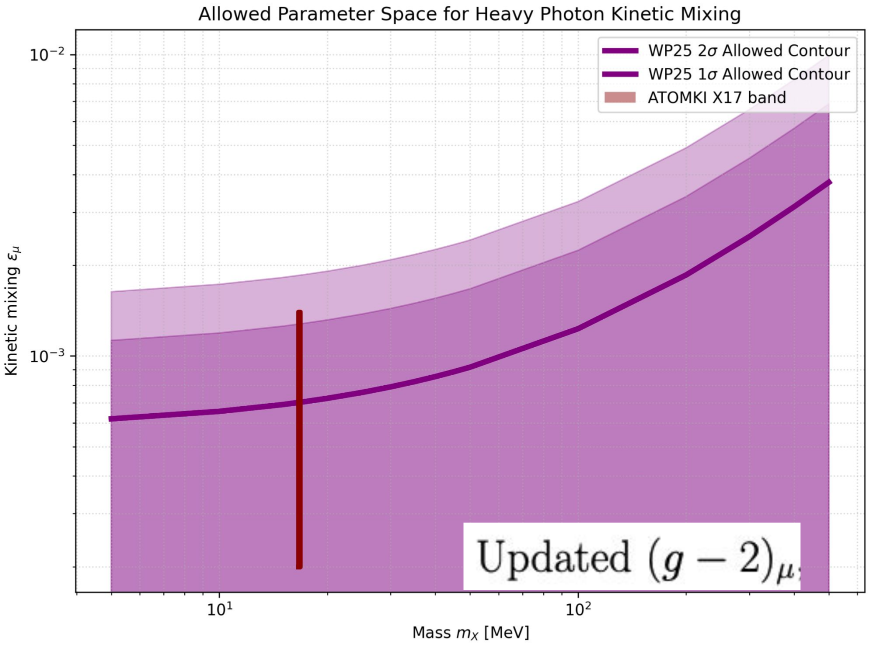
Updated $(g-2)_\mu$, $(g-2)_e$ and PADME-Favored Couplings Narrowly Compatible with the Preferred Region of ATOMKI X17, Given a Protophobic Interpretation

Emrys Peets*[1,2]





Previous Theoretical Models



Full Allowable Leptonic Coupling

Narrow overlap with favored electron coupling from most recent fine structure measurements.

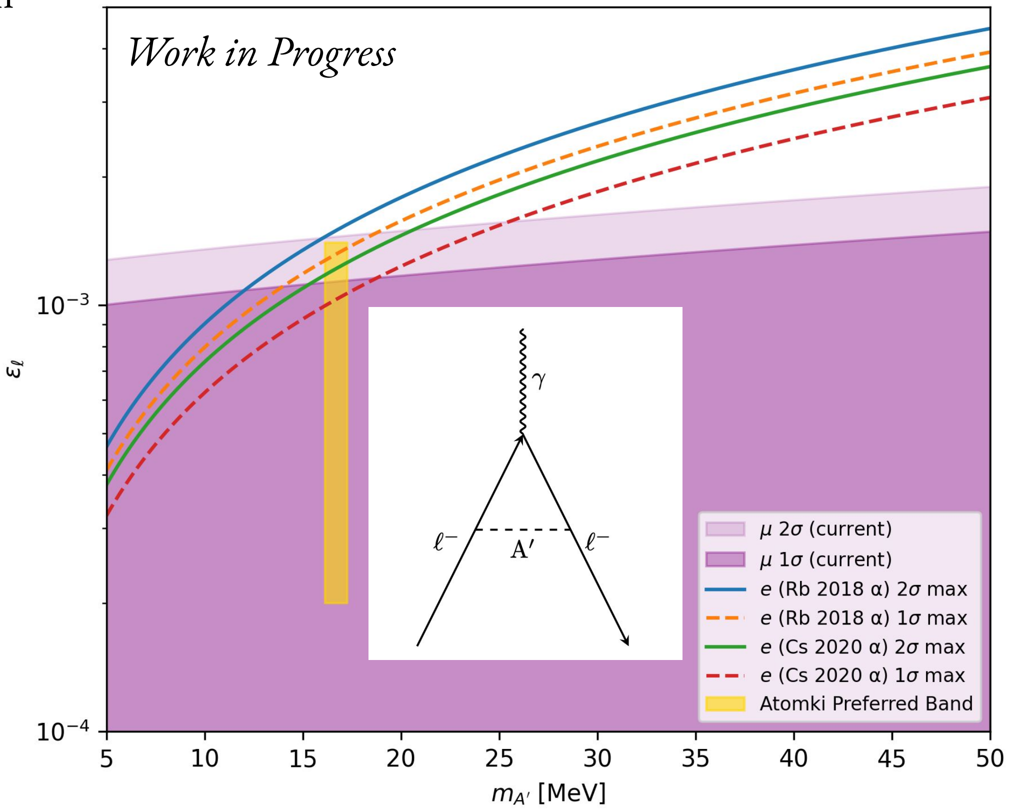
NA48/2 necessitates protophobic coupling

$$\pi^0 \to \gamma A' \to \gamma e^+ e^-$$

X17 Target

- Padme 2025 observed upper limit: 5.6×10^{-4}
- Invisible decays of protophobic vector boson shrinks upper bound of NA64 exclusion by factor of 2: $6.8 \times 10^{-4} \to 3.4 \times 10^{-4}$

$$3.4 \times 10^{-4} \lesssim \varepsilon \lesssim 5.6 \times 10^{-4}$$



Progress on X17 Collaboration Bumphunt Infrastructure

Internal Note Documenting Progress:

- https://www.overleaf.com/read/zpzknkkqrmk#bef878

Generated Toy Distribution with Signal Injected at 17 MeV and 40 MeV

- Gaussian Signal Injection [exaggerated signals for illustrative purposes], Poissonian Sampling (stat. variance of square root of predicted value), removed bin-by-bin jitter

pyBumpHunter Software Package

- proof of principal results gotten, can pick out significant bumps with known background shape

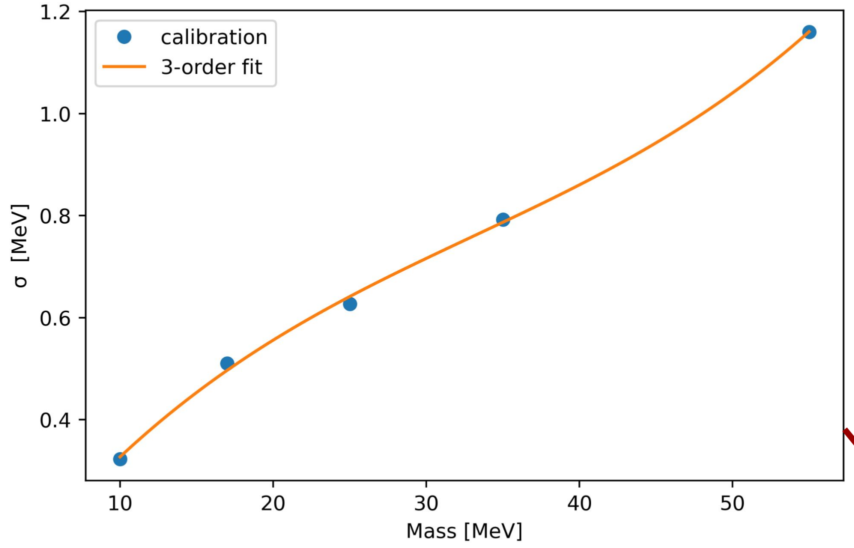
Preliminary upper limits and reach estimates determined

- 1, 5, 25, 40 days of beamtime

X17 - Gaussian Process Software Package under development

- data-driven / background agnostic methodology

Base Mass Resolution



Can fit using different shape as necessary

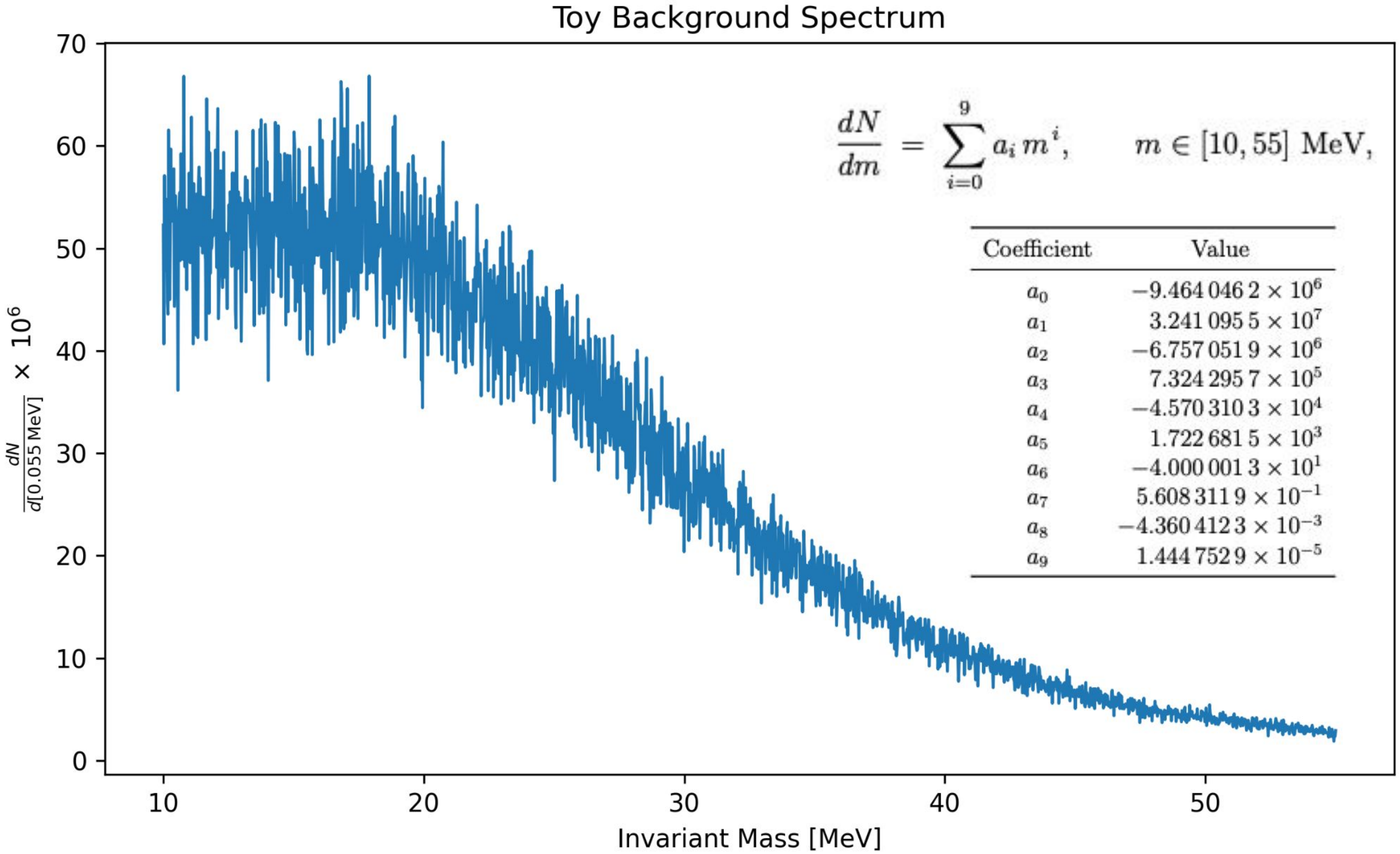
Initial Calibration Values (From Rafo)

Invariant mass m_i [MeV]	1σ mass resolution σ_i [MeV]
10	0.3225
17	0.5100
25	0.6270
35	0.7925
55	1.1600

Assuming Natural Width of X17 << Mass Resolution

$$S(m; m_0, \sigma) = \frac{N_{\text{sig}}}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{m - m_0}{\sigma}\right)^2\right\}$$

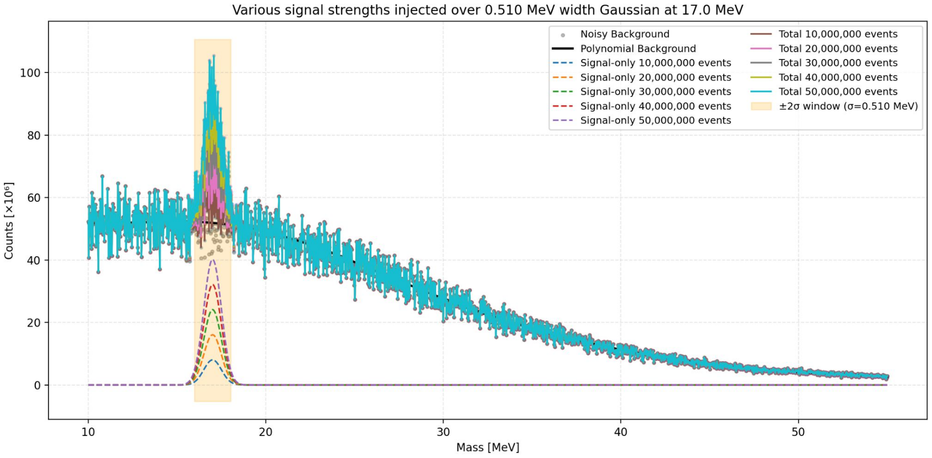
Base Background Distribution

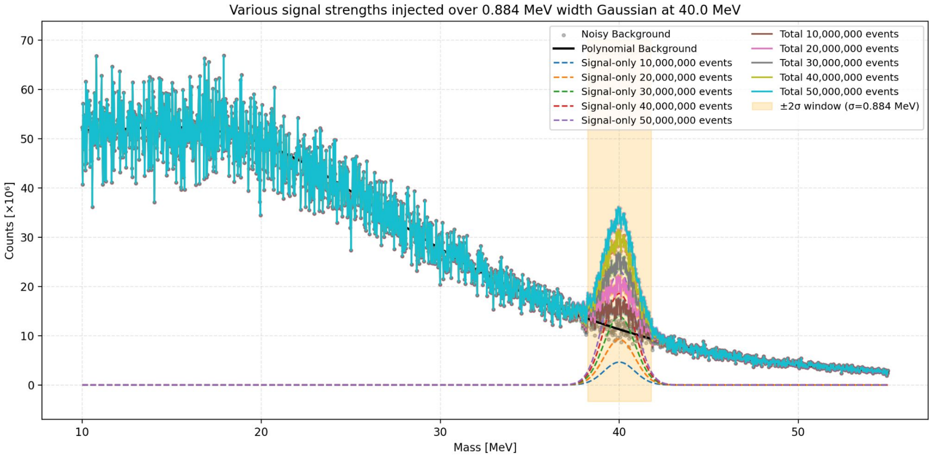


$$\frac{dN}{dm} = \sum_{i=0}^{9} a_i\, m^i, \qquad m \in [10, 55]\ \mathrm{MeV},$$

Coefficient	Value
a_0	$-9.464\,046\,2 \times 10^{6}$
a_1	$3.241\,095\,5 \times 10^{7}$
a_2	$-6.757\,051\,9 \times 10^{6}$
a_3	$7.324\,295\,7 \times 10^{5}$
a_4	$-4.570\,310\,3 \times 10^{4}$
a_5	$1.722\,681\,5 \times 10^{3}$
a_6	$-4.000\,001\,3 \times 10^{1}$
a_7	$5.608\,311\,9 \times 10^{-1}$
a_8	$-4.360\,412\,3 \times 10^{-3}$
a_9	$1.444\,752\,9 \times 10^{-5}$

17 MeV and 40 MeV Toy Distributions





Capabilities of pyBumpHunter

- pyBumpHunter is a python implementation of the BumpHunter algorithm described in arXiv:1101.0390, G. Choudalakis
- Accounts for the "look-elsewhere effect" by using the BumpHunter test statistic

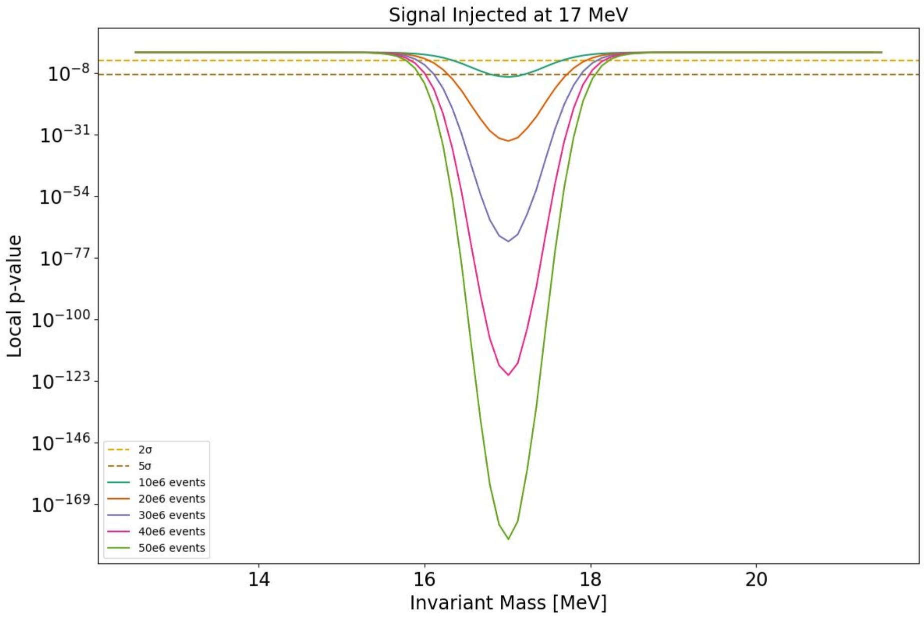
$$t = -\ln p_{\mathrm{min}},$$

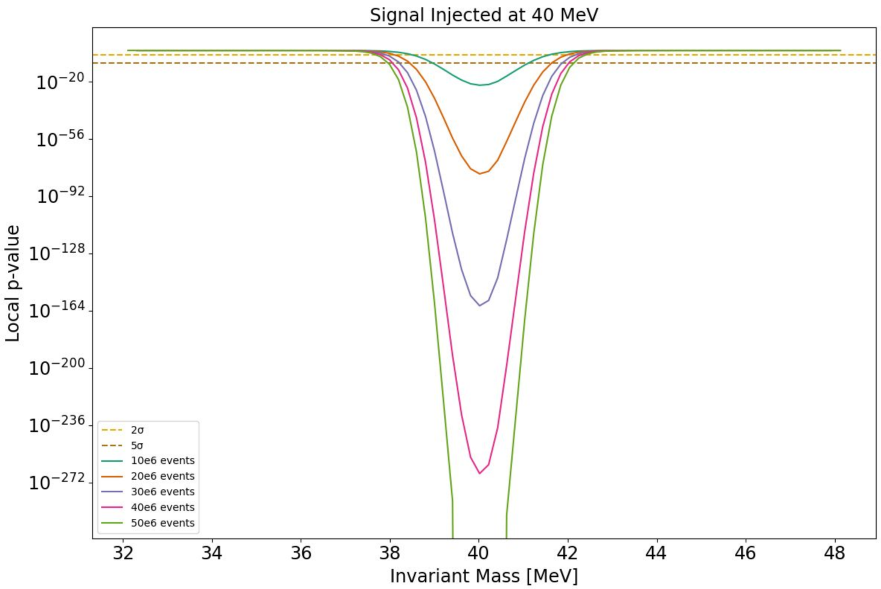
  and comparing this with generated background-only pseudo-experiments
- Can also perform signal injection tests
  - iteratively determine sensitivity given a background distribution

pyBumpHunter Setup and Examples

- We run pyBumpHunter on the toy distributions with signal injected at 17 and 40 MeV.
- The window size is twice the mass resolution and the step size is one fourth the mass resolution.
- The bump hunt finds local p-values for each window, then generates 10,000 pseudo-distributions and compare their BumpHunter statistics with the observed data.

pyBumpHunter Significances [17 MeV and 40 MeV]

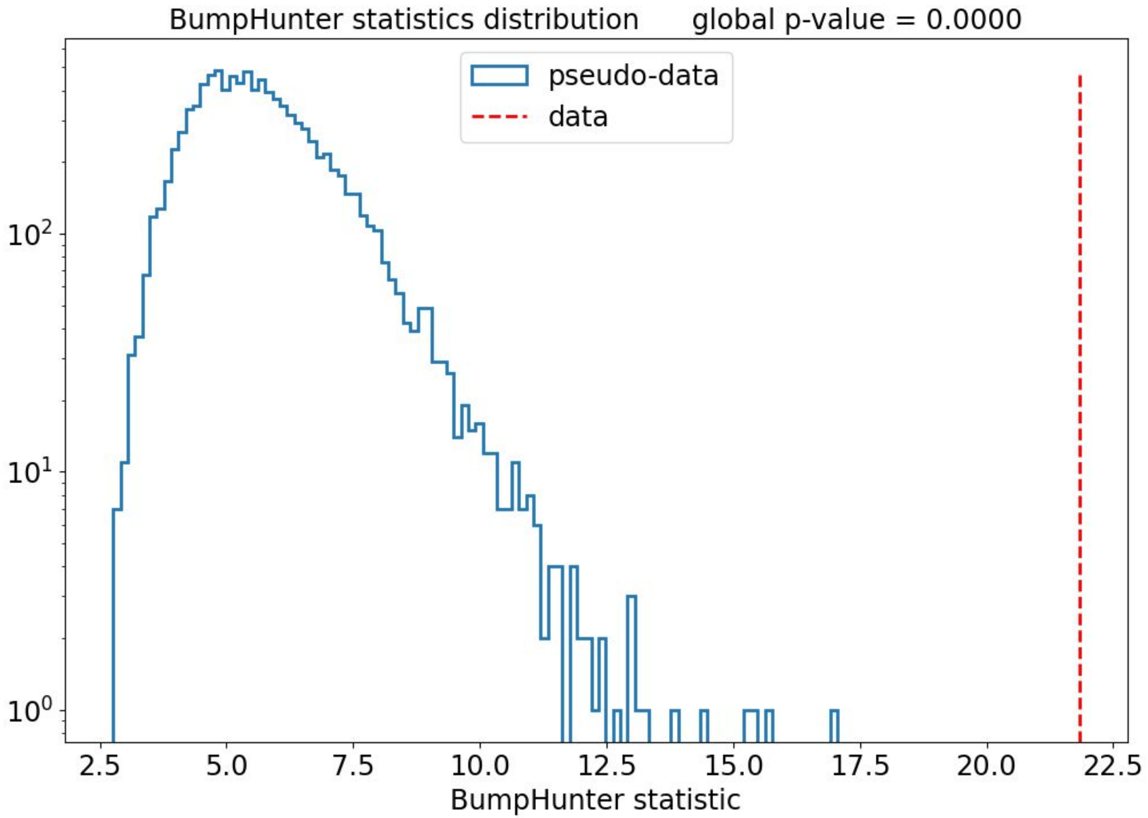




Injected Signal Statitics

- The BumpHunter statistics for 10 million events injected at 17 MeV.

- 0 / 10,000 pseudo-data distributions have statistics larger than the data, so we record a global p-value of 0.



Upper Limit Calculations

Idealistic Scenario

- Calculated from pure polynomial form, currently unrealistic but good first order approximation.

Signal yield per limit calculation from https://arxiv.org/abs/1307.2487

- Currently scans for a **Counting Excess**
  - Integral of events above background in $\pm 2\sigma$ window
  - Does not yet take into account signal shape

Preliminary upper limits on signal yield and coupling determined for 1, 5, 25, and 40 days.

Upper Limit on Signal Yield

$$s_{\text{up}} = \frac{1}{2} F_{\chi^2}^{-1}[p, 2(n+1)] - b,$$

$$p = 1 - \alpha \left(1 - F_{\chi^2}[2b, 2(n+1)]\right),$$

$$\mathrm{CL}_s(\mu) = \frac{p_\mu}{1 - p_b} \qquad \mathrm{CL}_s(N_{sig}^{up}) = 0.05$$

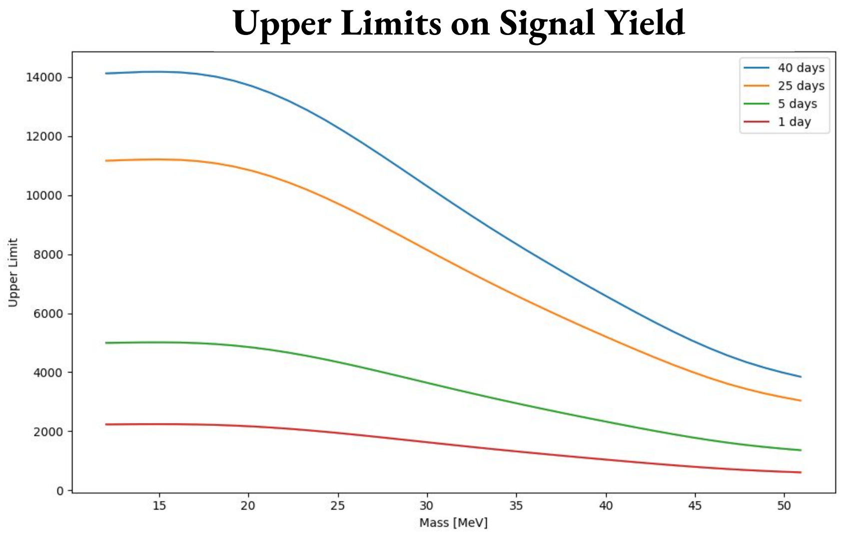
Upper Limit on Signal Yield

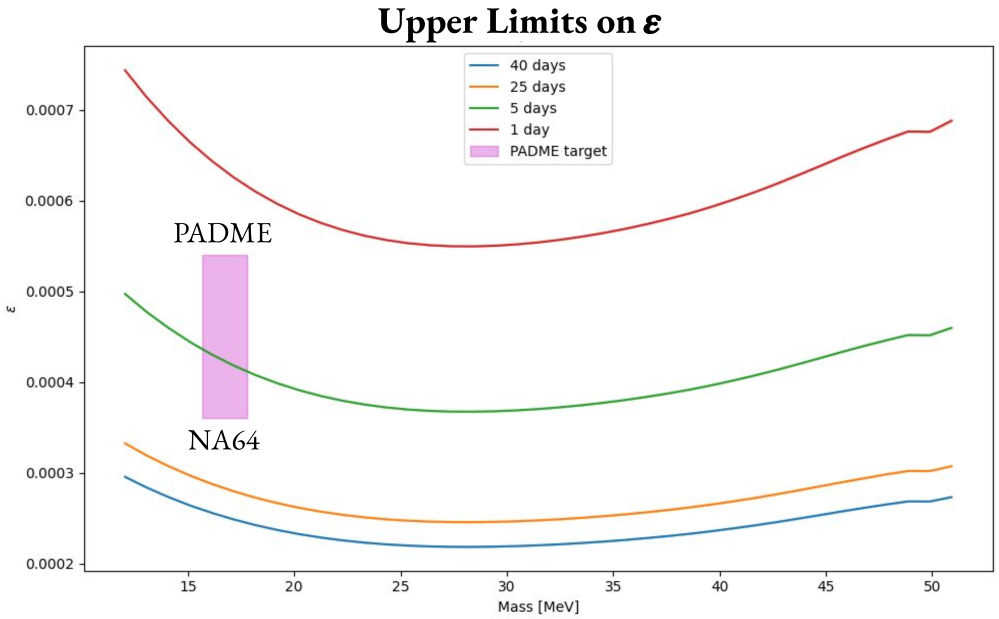
$$\epsilon^2 = \frac{2\alpha N_{\text{sig}}^{\text{up}}}{3\pi m_{A'} f_{\text{rad}} \frac{\mathrm{d}N_{\text{bkg}}}{\mathrm{d}m}}$$

Rad Frac: Fixed 4%

From: https://arxiv.org/pdf/0906.0580

Preliminary Reach Estimate

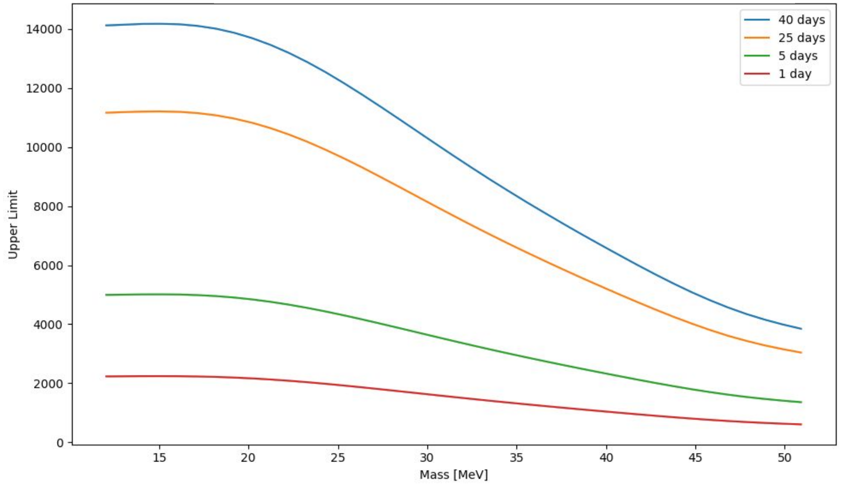




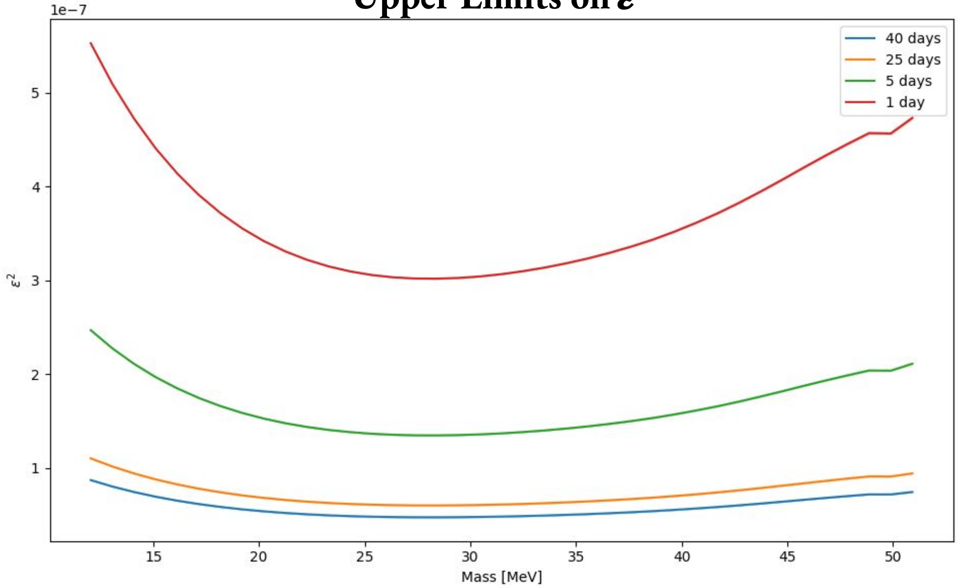
Optimistic given the idealistic polynomial interpretation, but promising!

Preliminary Reach Estimate

Upper Limits on Signal Yield



Upper Limits on ε^2



This is assuming a known background shape, **a dangerous assumption**.

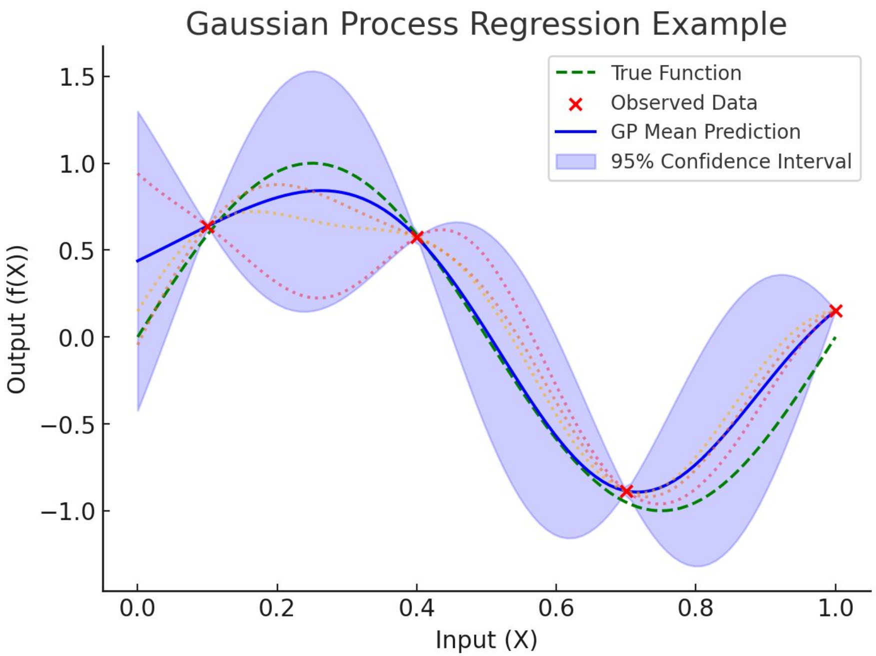
Gaussian Process Regression as alternative resonance search technique

Understanding Gaussian Process Regression

Collaboration of Emrys Peets (Stanford/SLAC), Joseph Bailey (Stanford), Tom Eichlersmith (Minnesota), Aidan Hsu (Stanford), Takumi Britt (UCLA).

What is Gaussian Process Regression (GPR)?

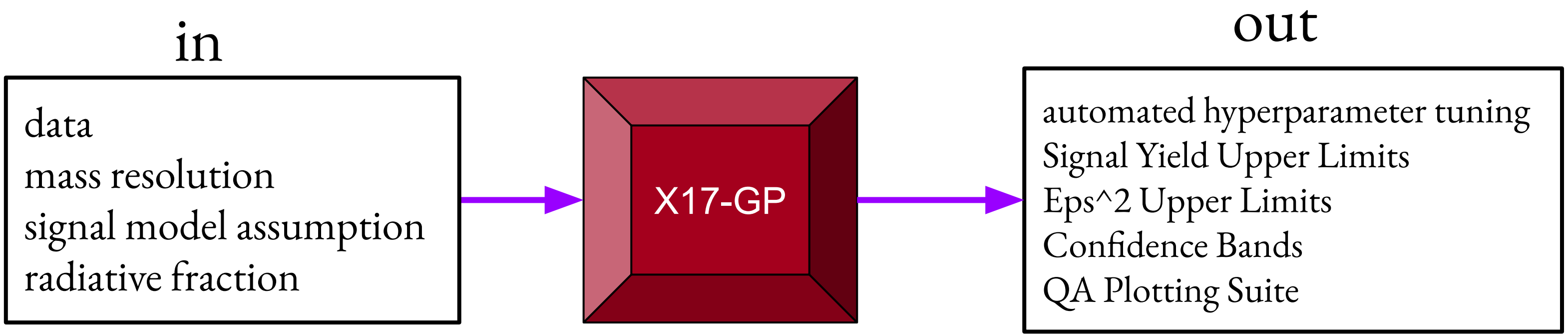
- A **flexible, non-parametric Bayesian approach** that models distributions over functions.
- Unlike traditional regression, **GPR does not assume a fixed set of parameters**—it learns a distribution of possible functions.
- **Built-in uncertainty quantification** makes it ideal for noisy and complex datasets.



The kernel function (covariance function) governs how data points interact and influence one another.

The choice of kernel shapes the model's **smoothness, flexibility, and generalization ability**, making it crucial for capturing underlying data patterns.

X17-GP Software Package



Standalone software using background agnostic technique under development in parallel with similar HPS methodology.

Kernels and Hyper Parameters

In GPR space: for Kernel, K, bin contents (y_i) are described by the gaussian PDF:

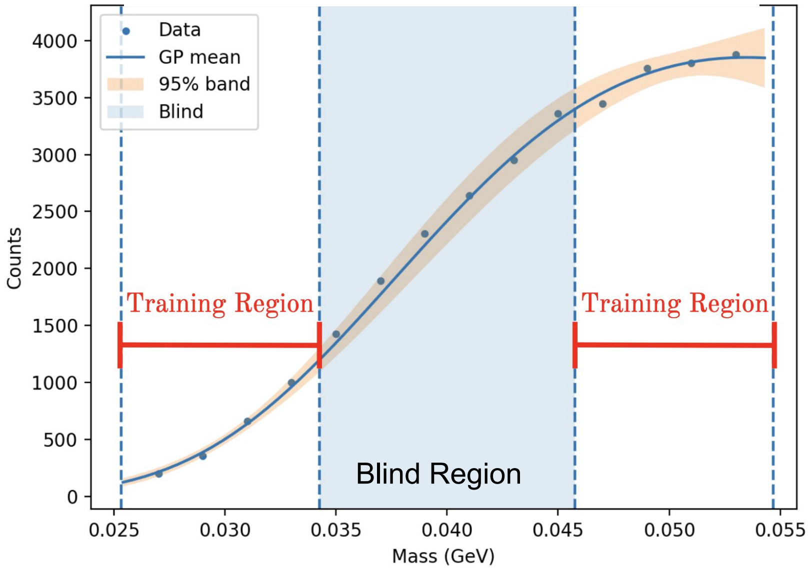
$$p(y_i; \mu_i, \mathbf{K}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{K}|}} \exp\left(-\frac{1}{2} \sum_{i,j}^{n} (y_i - \mu_i) K_{ij}^{-1} (y_j - \mu_j) \right)$$

Radial Basis Function (RBF)

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\ell^2} \right)$$

Euclidean metric, radially symmetric

Monte Carlo Illustrative Display



Passing in Alpha

Hyperparameter meant to capture known bin-bin variance

$$y_i = f(x_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \alpha_i)$$

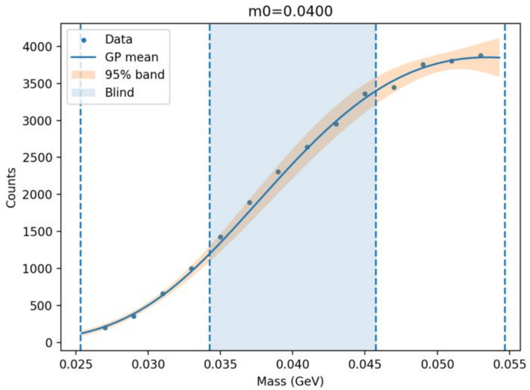
Formally a Kernel with alpha has the covariance form

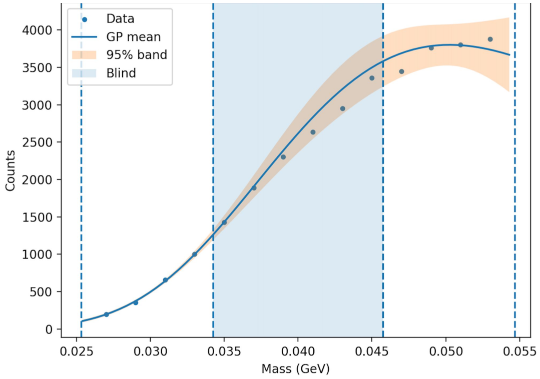
$$K_\alpha \equiv K(X, X) + \Sigma, \;\; \Sigma = \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$$

With impact on log marginal likelihood

$$\log p(y|\theta) = -\tfrac{1}{2} y^\top K_\alpha^{-1} y \; - \; \tfrac{1}{2} \log |K_\alpha| \; - \; \tfrac{n}{2} \log(2\pi)$$

Thus, increase of alpha relaxes the fit and increases the penalty.
(larger predictive uncertainty, preferring longer length scales).
Very small alpha risks fitting bumps.





Motivations from Recent CMS Study

Gaussian Process Regression as a Sustainable Data-driven Background Estimate Method at the (HL)-LHC

Summary

- GPR is proposed as alternative to functional form in Run-2, high luminosity- LHC datasets.
- Proof of principal: pseudo experiments generated using CMS b-tagged resonance search (published using multiple functions to fit multiple regions)
- Tuning alpha led to significant improvement in pass rates of different background validation metrics.

Early X17-GP Testing

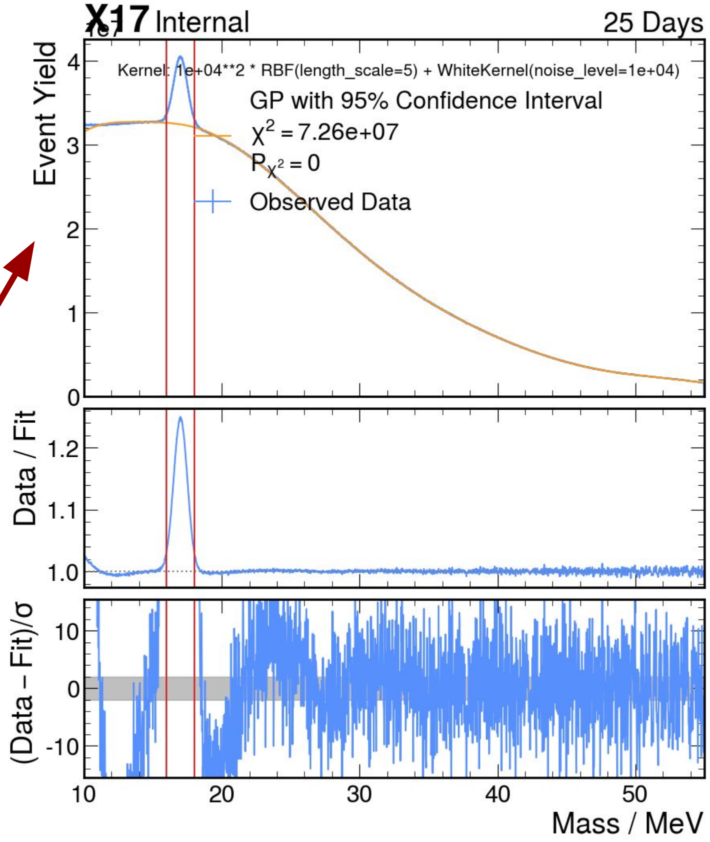
Still in early phases, but progressing rapidly in parallel to HPS resonance search on 2015, 2016, 2019, 2021 datasets.

- to be finalized well before X17 beamtime

Piecewise methodology necessary to eliminate information from different regions and save computing time.

Exaggerated signal injected over the background for 25 days of beamtime

- No background model given
- Clear significance of signal



Next Steps

- Joseph plans to present on the methodology at APS Santa Cruz meeting in October
  - Abstract submitted, X17 collaboration to be mentioned as promising use case
- Generate more realistic background distribution and conduct more signal injection tests.
- Use validation metrics to ensure unambiguous results and optimize kernels
- Ensure accessibility of X17-GP package

Broader Goal

Use standardized GP software to ensure results compatible and easily comparable between different experiments and collaborations

- X17 / HPS / BaBar have compatible parameter space , using equivalent background modeling technique is powerful and results can be later combined where overlapping.

Questions / Thoughts / Suggestions?

Thank you for listening!

Extra Slides:
Changes made to pyBumpHunter software
HPS as proof of principal for GPR
Methodology of CMS Study

Changes to pyBumpHunter

The GitHub release contains some bugs that prevent pyBumpHunter from running properly

In pyBumpHunter/bumphunter_1dim.py:

- lines 1511, 1983 uses deprecated numpy behavior
- line 1891 needs to copy the looping behavior at line 1320

CMS Study Methodology

I. Kernel Selected: **Constant X RBF**
II. Preprocessing
   - log(mass) , log(yield) to stabilize wide dynamic range
III. Length Scale Bounds
   - lower bound ~ mass resolution, upper bound broad
IV. Hyperparameter Optimization
   - alpha changes depending on the region of dataset (should broaden/trend with mass resolution)
V. Validation Metrics
   - chi2/ndof ~ 1.5, KS pvalue >0.05, BH pvalue > 0.1

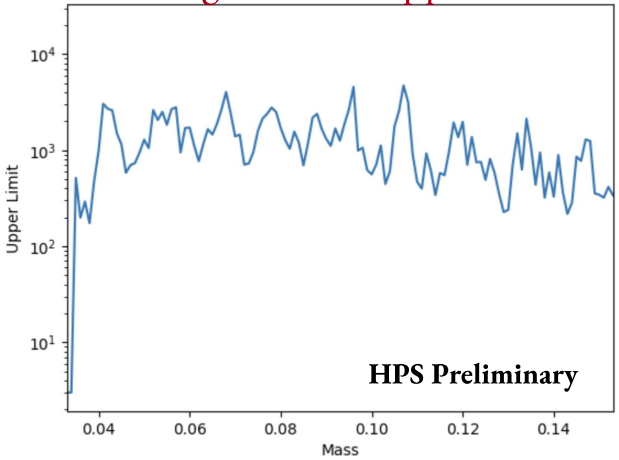
Application of GPR to HPS Datasets

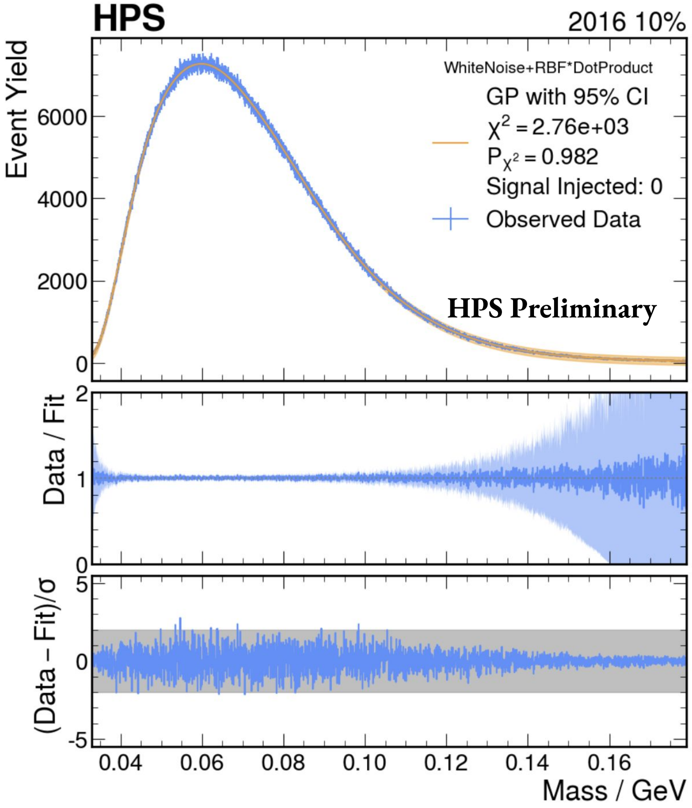
The GPR model provides a strong fit to the datasets with well-defined uncertainty estimates.

Preliminary upper limits determined to be competitive with functional form fitting.

Kernel Choices:
WhiteNoise - models broad noise
RBF Kernel - models local correlations

10% Signal Yield Upper Limit





from Aidan Hsu's work