



University
of Glasgow

Gaussian Process

Data-driven Approach for Interpolation of Sparse Data

Ryan Ferguson

r.ferguson.3@research.gla.ac.uk

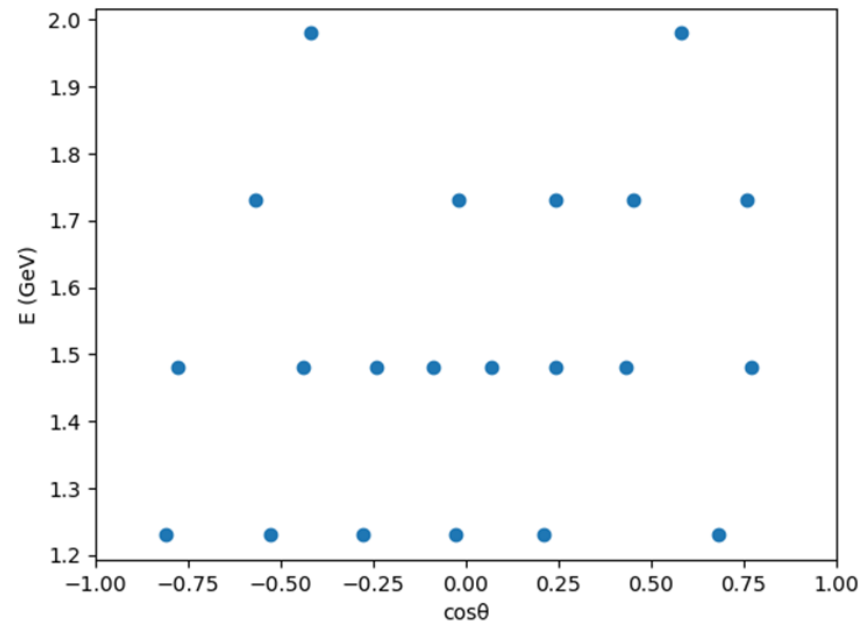
Why are we doing this?

Why?

- Extracting information about hadron resonances requires fitting theoretical models to experimental data.
- However, this data often comes from different experiments of different physics quantities in varying kinematic regions.
- Furthermore, studying coupled channels with different kinematic coverages and binning can make direct comparison challenging.
- The consistency of these datasets directly impacts the quality of the fit, thus making it difficult to accurately constrain the theoretical models.

Why?

- Sparse datasets in key kinematic regions further complicates the quantification of uncertainties, often requiring arbitrary weighting that may introduce bias.



What can machine learning do?

- A Gaussian Process (GP) can be used to predict the value and uncertainty of a quantity at unmeasured kinematic points.
- By comparing measurements from different experiments, inconsistencies, such as variations in coverage or binning, can be identified and addressed.
- Theorists can use GP built datasets to test models and highlight kinematic regions where theoretical models diverge from the GP predicted empirical trends.
- The GP can give these results with no theoretical model, removing the limitations of arbitrary weighting in sparse datasets.
- The full dataset can be utilised without the need for arbitrary splitting into training and testing sets, avoiding unnecessary loss of data.

How does it work?

Assumptions

The Gaussian Process only requires 3 assumptions to operate:

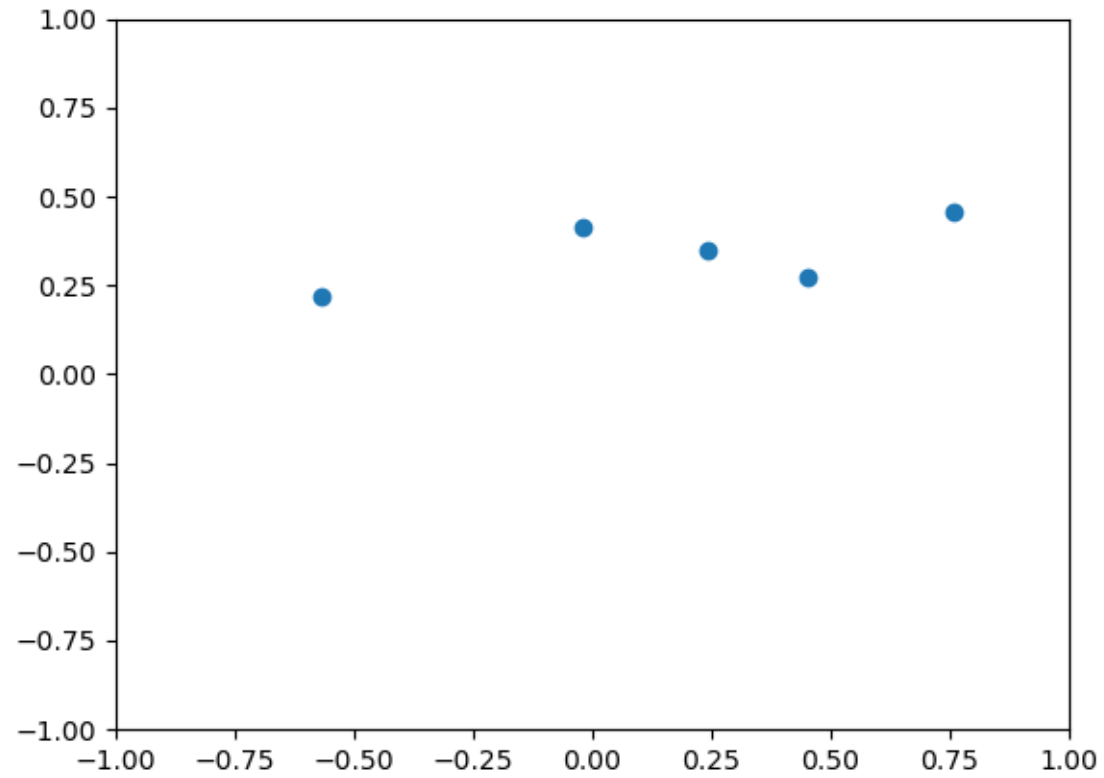
1. Kinematic datapoints form a multivariate Gaussian.
2. Some kernel function can be used to measure the covariance between known datapoints. This same kernel function can also predict the covariance of other, unknown datapoints.
3. The style of posterior distribution is known (e.g., smoothness, continuity, periodicity, monotonically increasing, etc.).

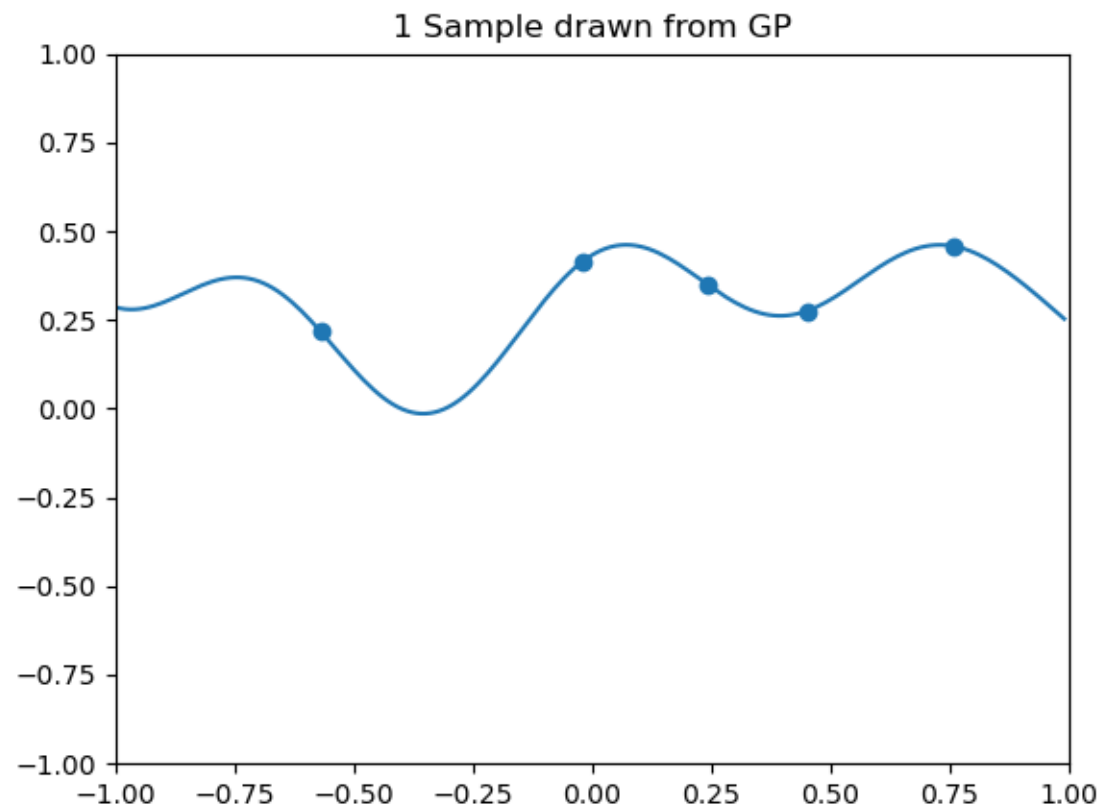
From this, the GP forms a multivariate Gaussian combining both the known and unknown datapoints.

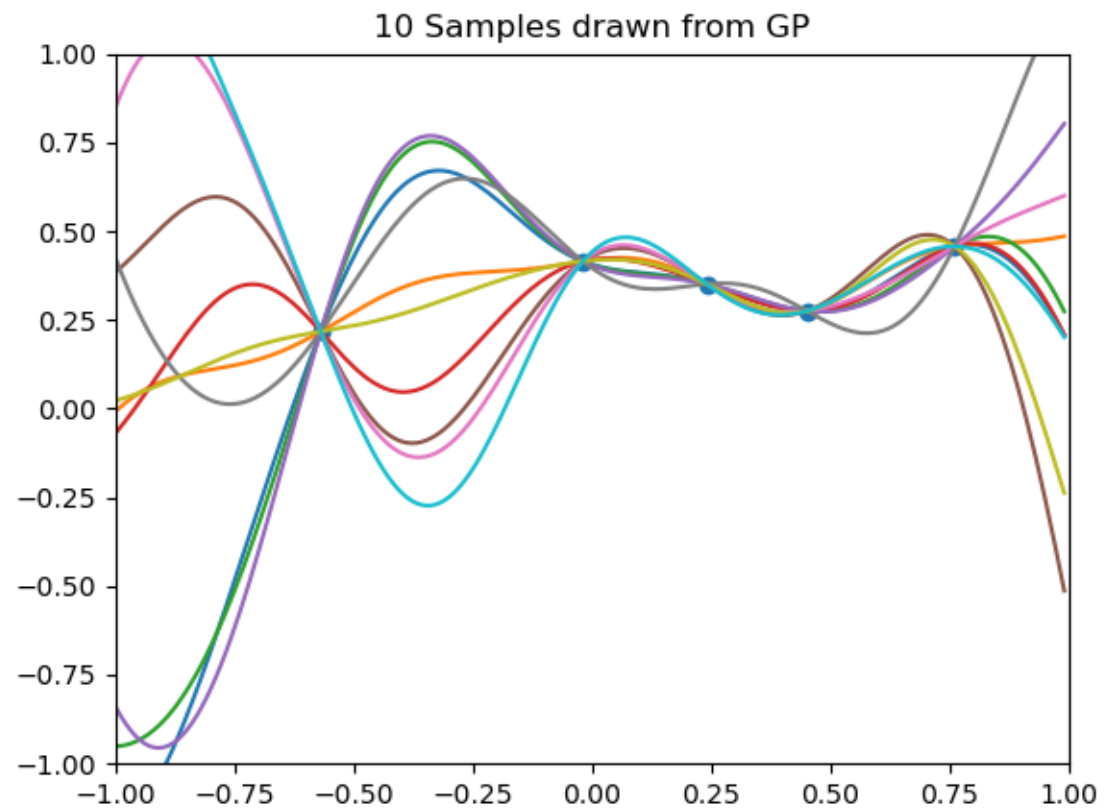
Assumptions

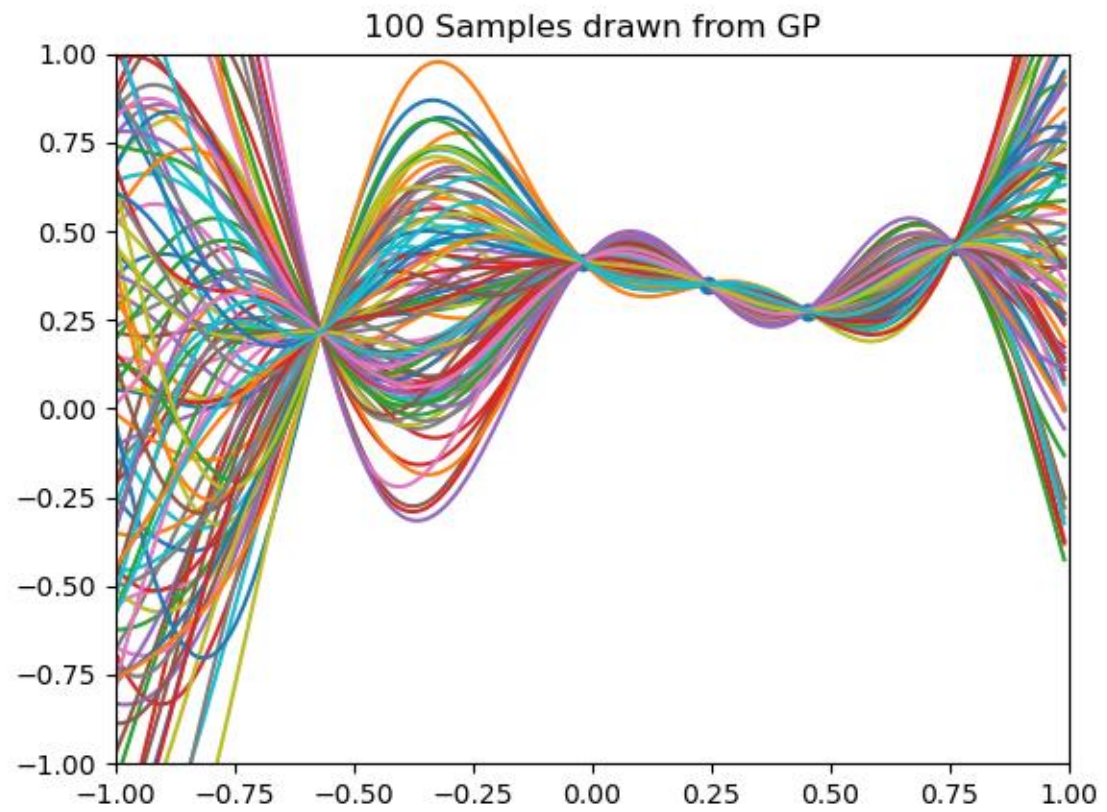
- By using the conditional of a multivariate Gaussian, the GP can provide a mean and uncertainty for the unknown datapoints.
- The known datapoints can be thought of as one possible sample drawn from this multivariate Gaussian.

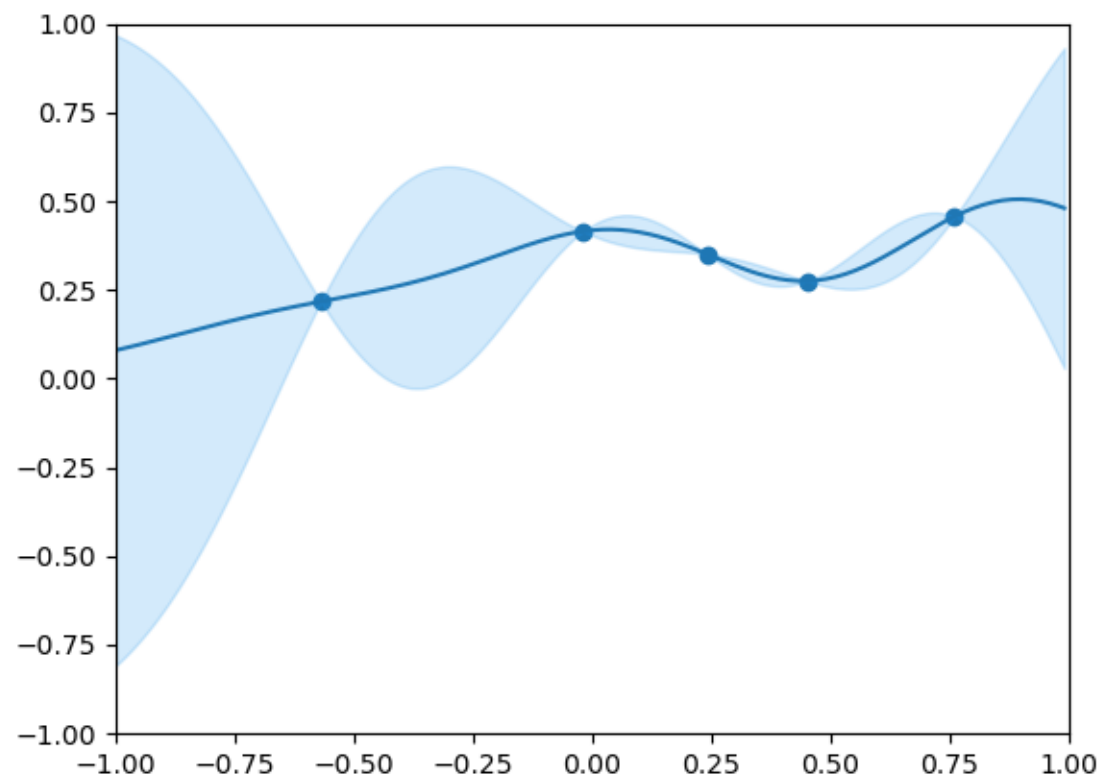
Example











Specifics of this GP model

Kernel Choice

The Radial Basis Function kernel can be used:

$$\kappa(\underline{a}, \underline{b}) = \exp \left[\sum_{i=0}^{p-1} \frac{-d(a_i, b_i)^2}{2l_i^2} \right]$$

Where:

- $\underline{a}, \underline{b}$ are some vectors of length p (e.g. have p parameters)
- $d(\cdot, \cdot)$ is the Euclidean distance.
- l is a hyperparameter called the length scale. For this kernel, it is a measure of how smooth the function is.

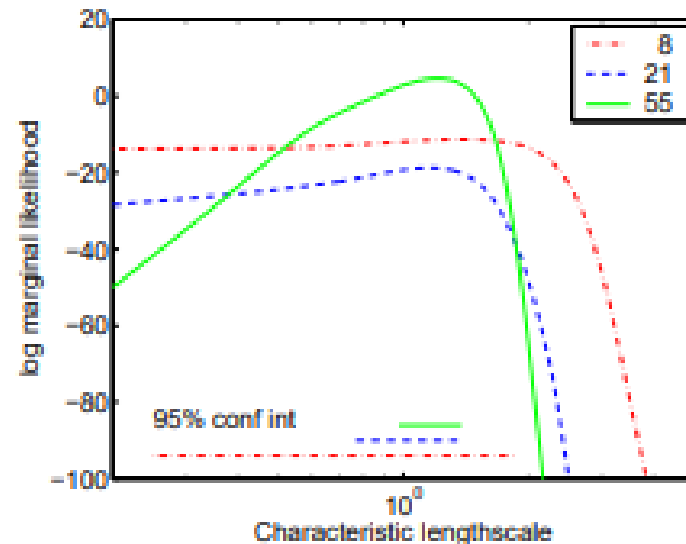
The RBF kernel gives smooth, continuous posterior distribution which is appropriate for the cases presented here.

Standard Approach

The normal approach is to use the marginal likelihood function:

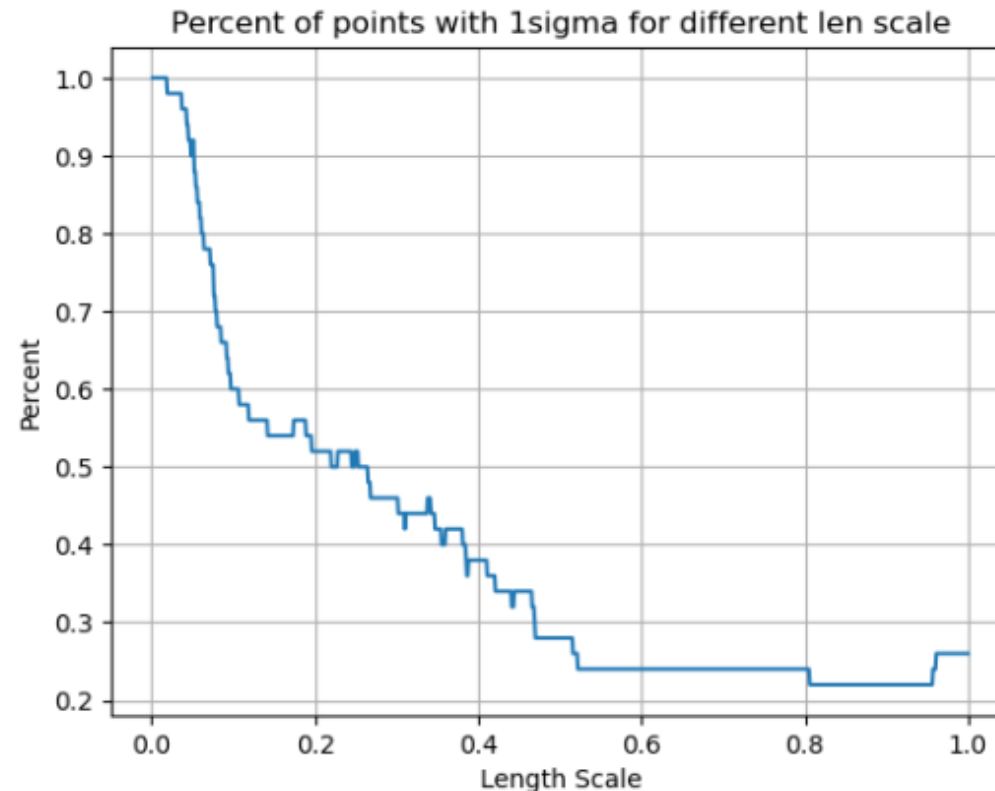
$$\log p(y|X, \vec{l}) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi$$

For low numbers of datapoints this is not well defined:



Bayesian Hyperparameter Search

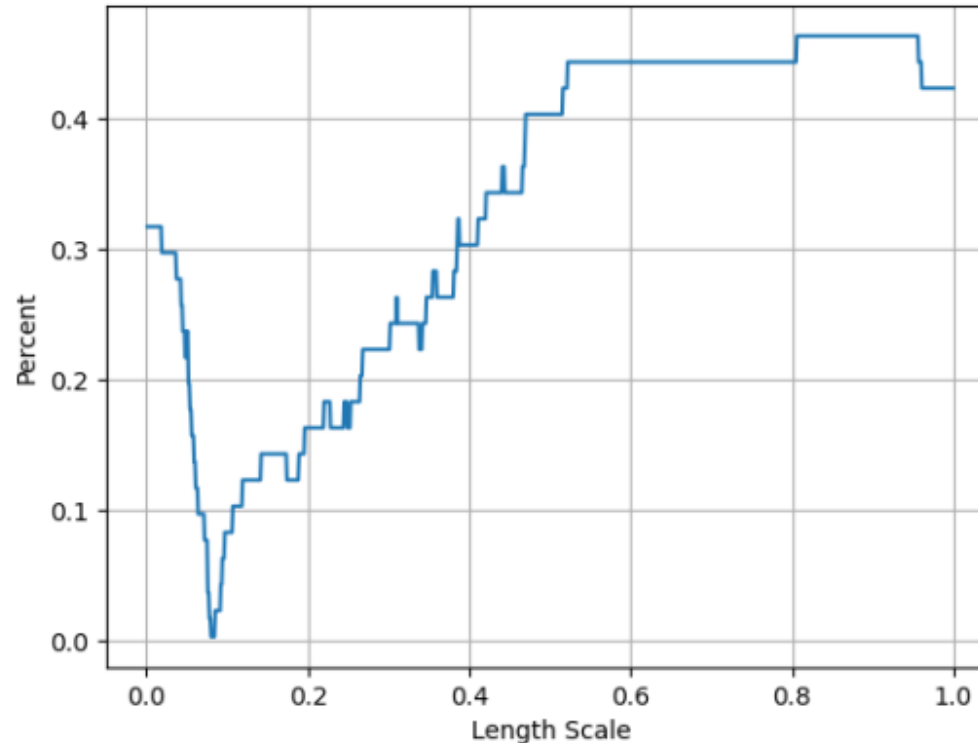
- Previously a test of the goodness of the GP fit was to find the percentage of points and compare this to the predicted value (e.g. 50% in 0.67σ , 68.3% in 1σ , etc.).
- For a range of possible length scales, we can find the percent of points within (e.g.) 1σ :



Bayesian Hyperparameter Search

- Standard statistics says we expect 68.3% within 1σ , therefore we want to find the length scale which minimises the difference between the measured value and 68.3% (the predicted value).

Difference between measured and predicted percent of points with 1sigma for different len scale



Loss function – Sigma check

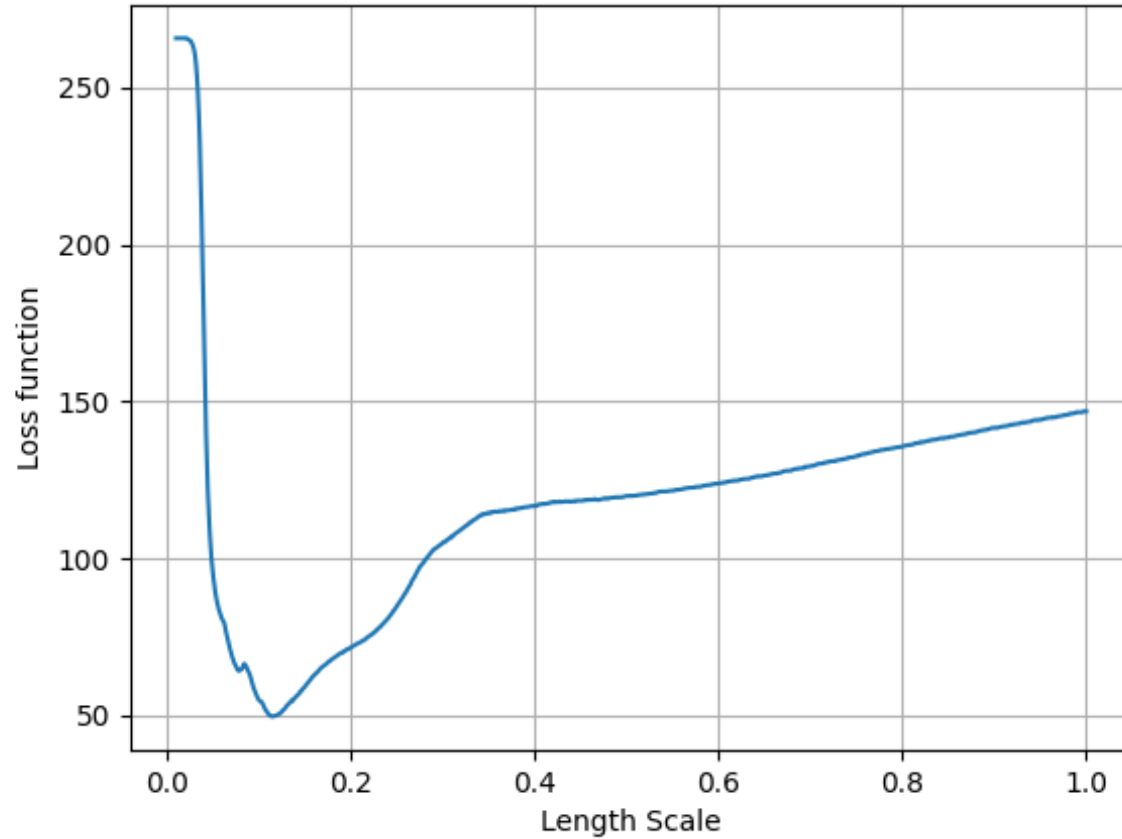
- To provide a more robust optimisation, it is beneficial to test additional multiples of the standard deviation and compare them to their predicted value.
- Confidence intervals up to 3σ cover approximately 99.7% of data, thus offering a comprehensive evaluation range. By taking various multiples, the choice has effectively been marginalised.
- Consequently, define the loss function as:

$$f(\vec{l}) = \sum_{c>0}^3 |M(c\sigma, \vec{l}) - P(c\sigma)|$$

Where:

- $M(c\sigma, \vec{l})$ is the measured percentage of points within $c\sigma$ (c is a scalar) for a given length scale \vec{l} .
- $P(c\sigma)$ is the predicted percentage of points within $c\sigma$.

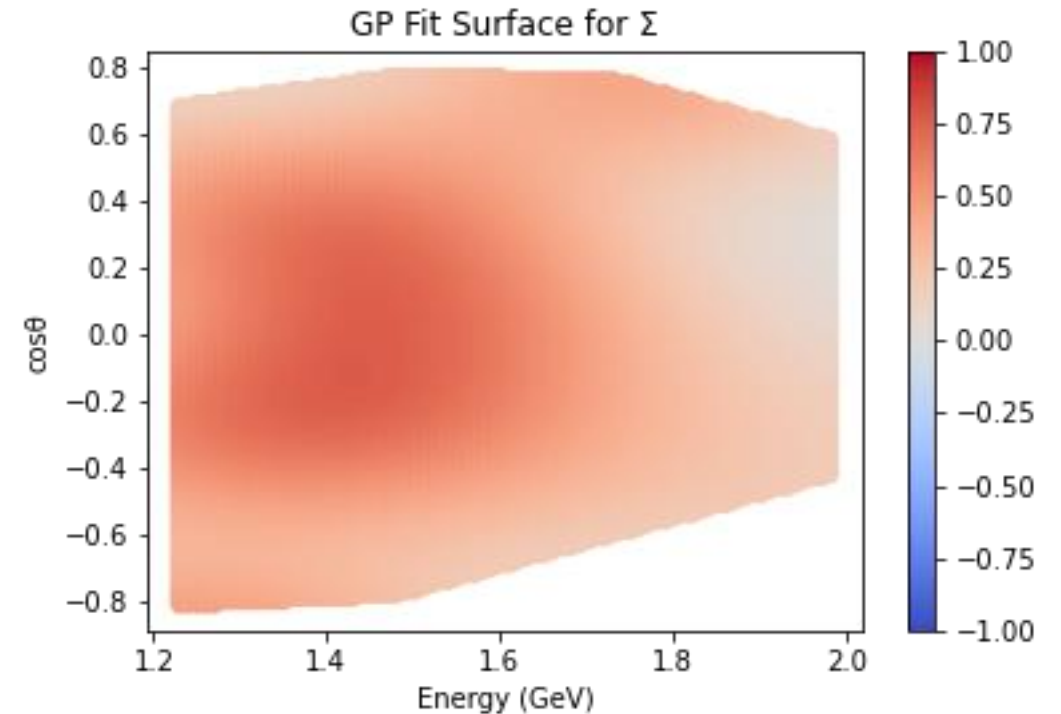
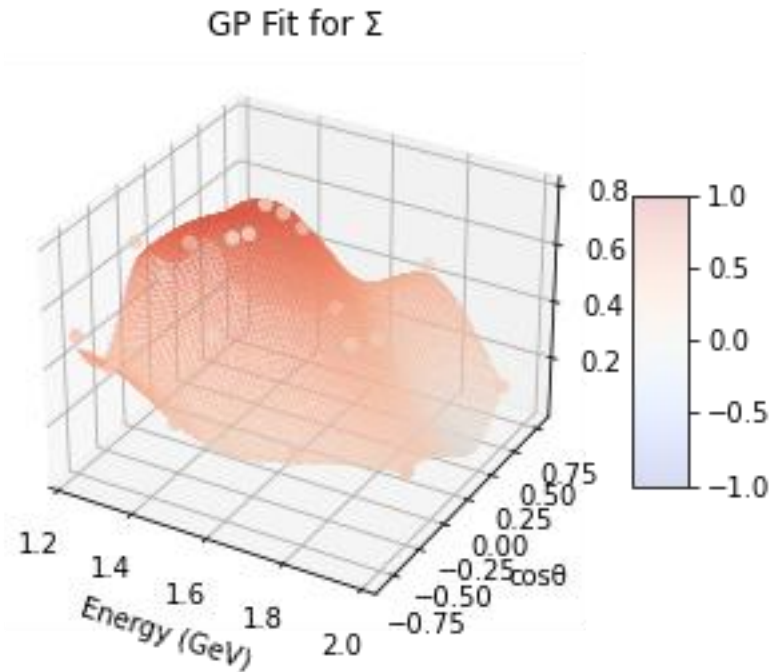
Loss function – Sigma check



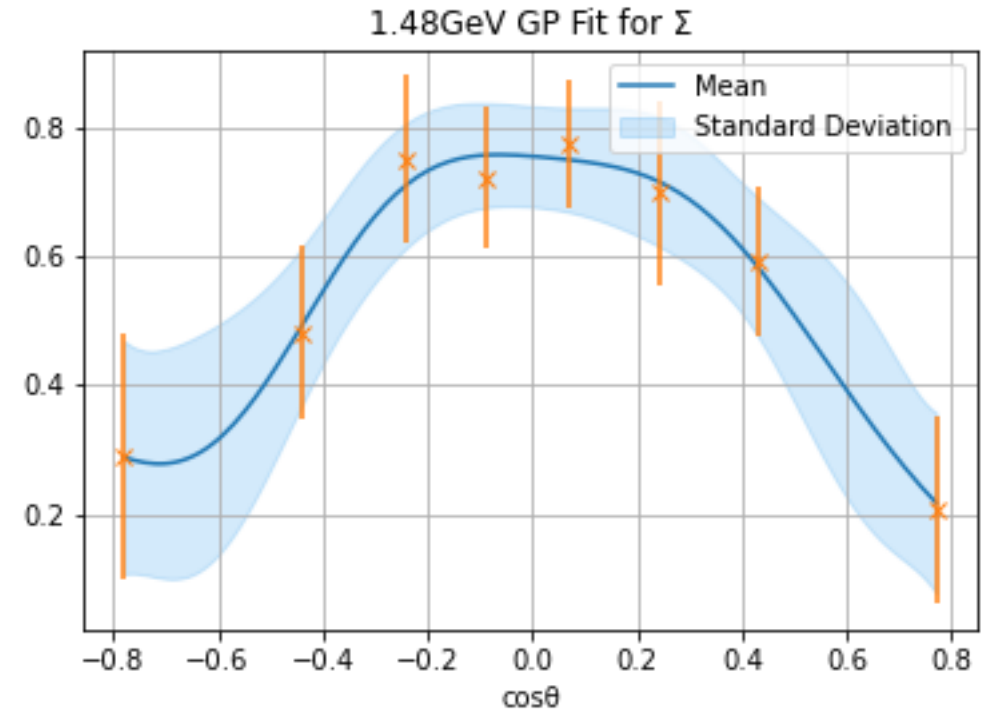
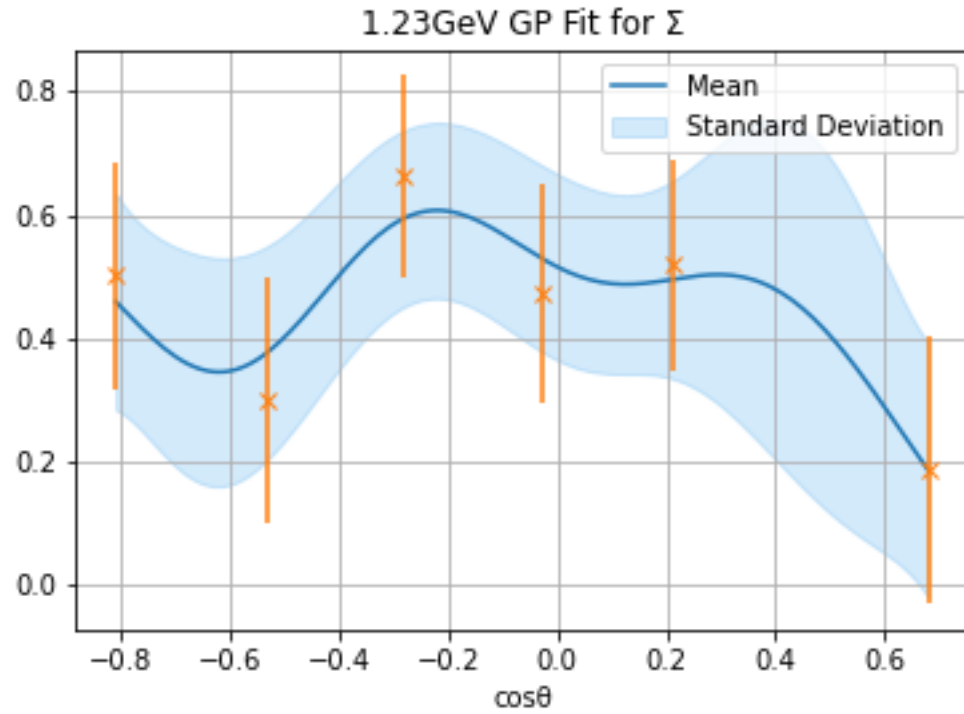
What does real data look
like?

Data from CLAS

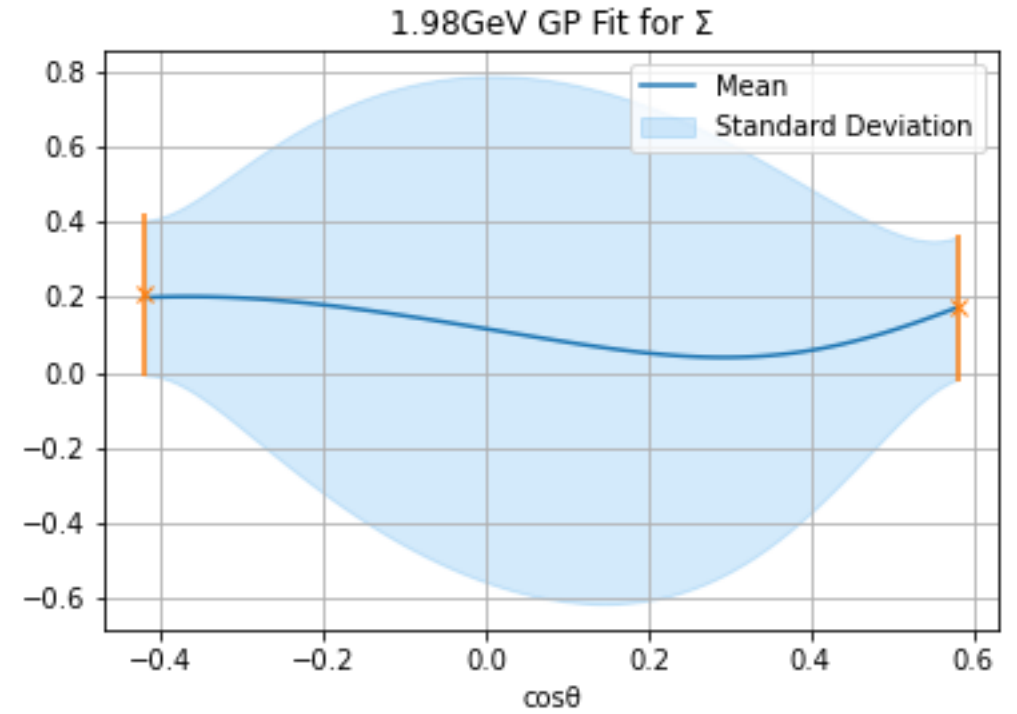
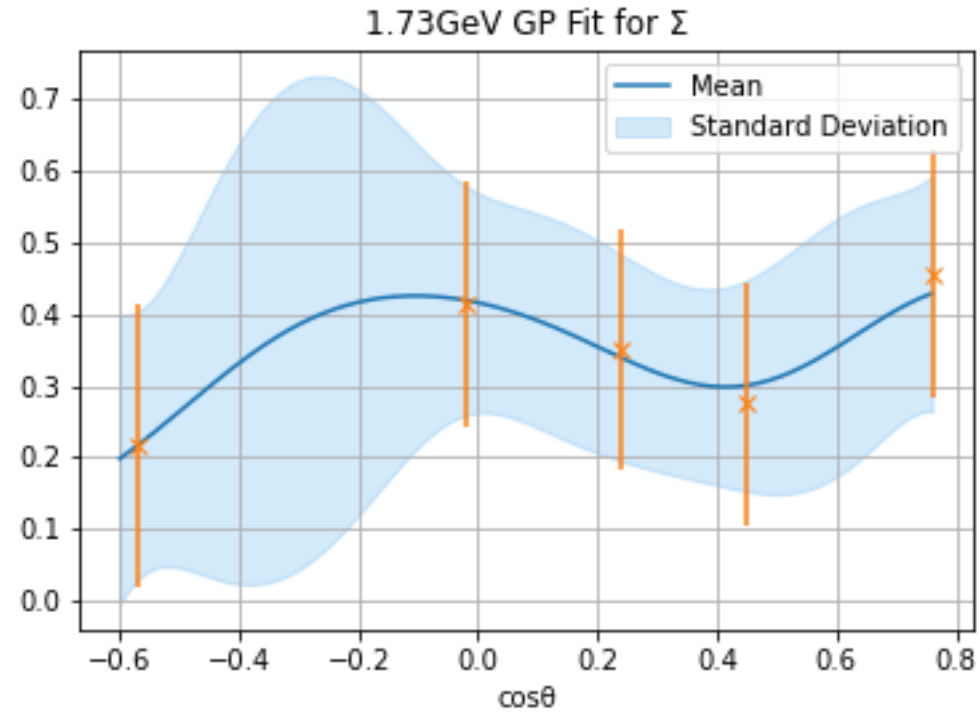
The GP has been used on data recently published by the CLAS collaboration at Jefferson Lab, specifically 5 polarisation observables (Σ , P , T , O_x and O_z) of the $K^0\Sigma^+$ reaction.



GP 1D Projections for Σ

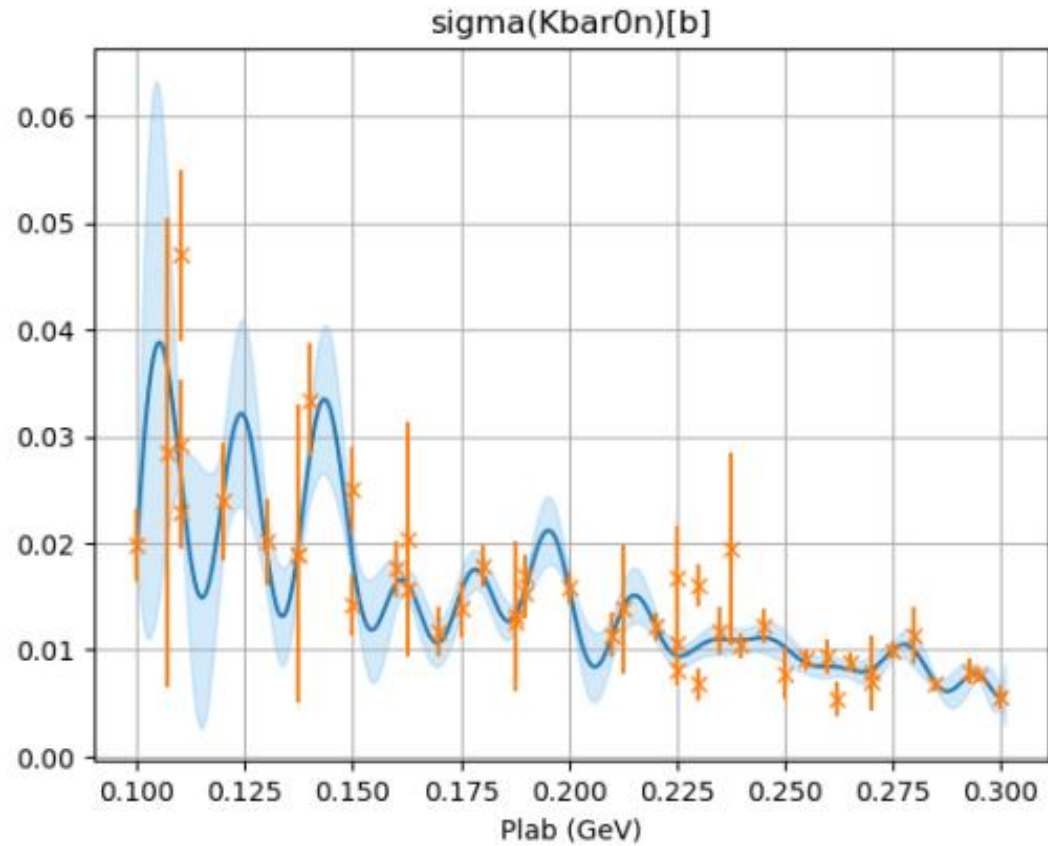


GP 1D Projections for Σ



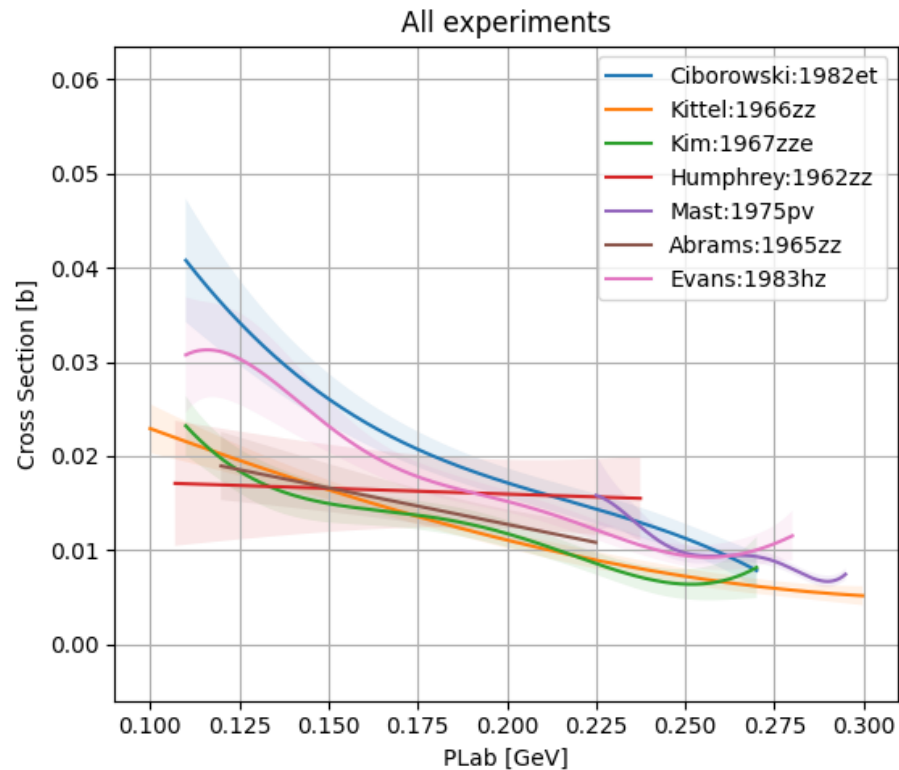
KbarN Cross Section

Try to run a GP fit on the world data for $K^-p \rightarrow \bar{K}^0n$



KbarN

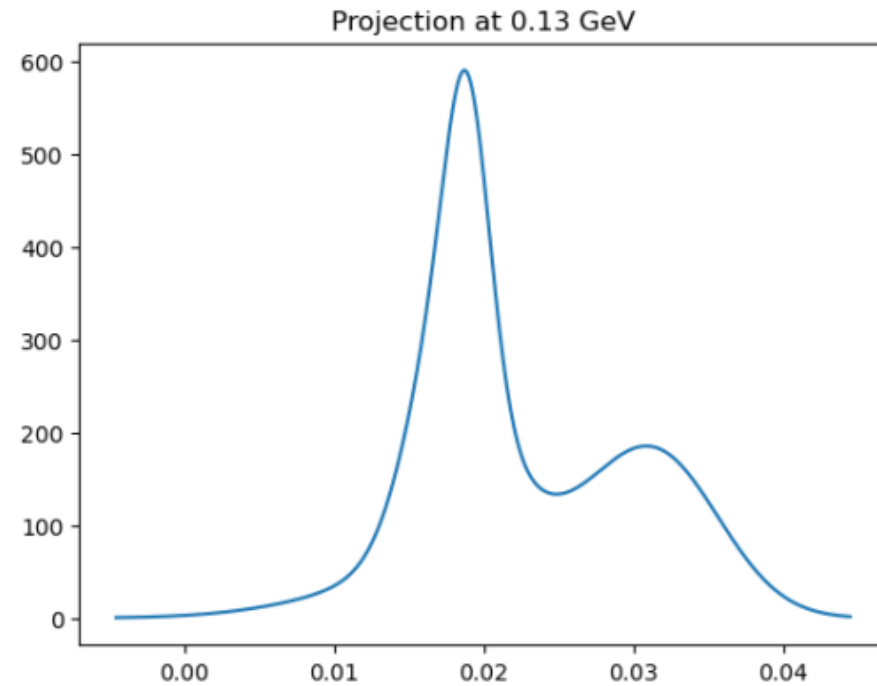
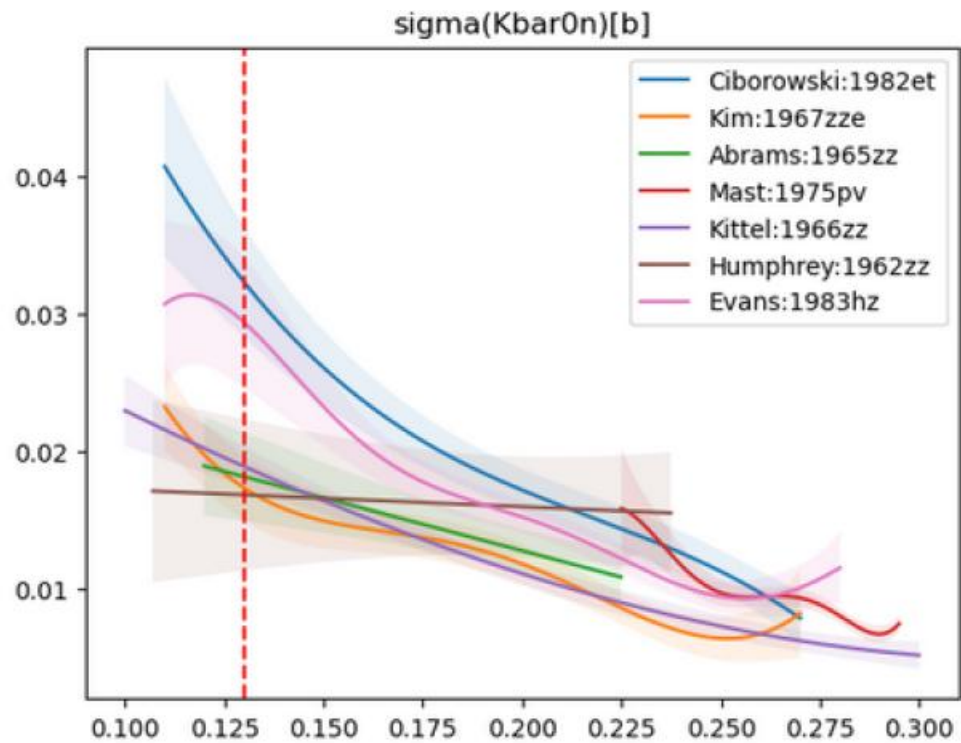
Instead running a GP on each individual experiment produces the following:



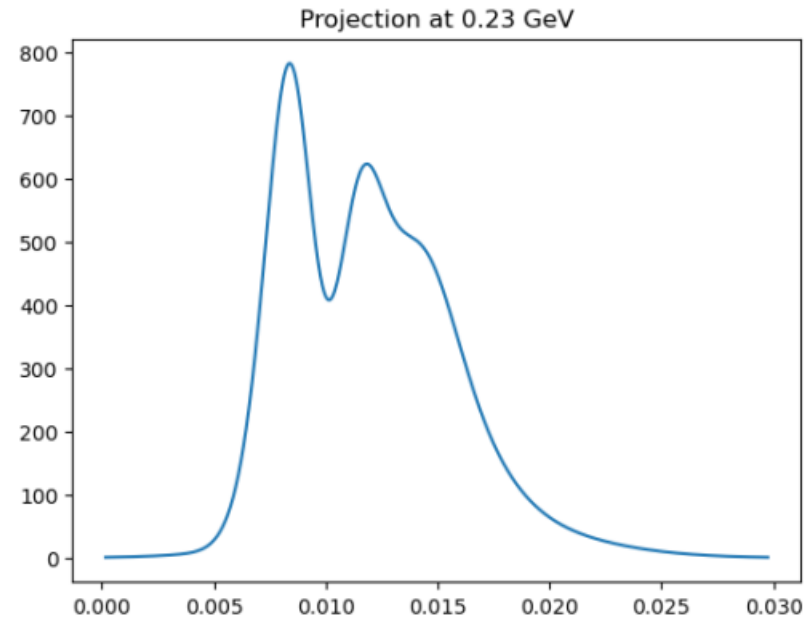
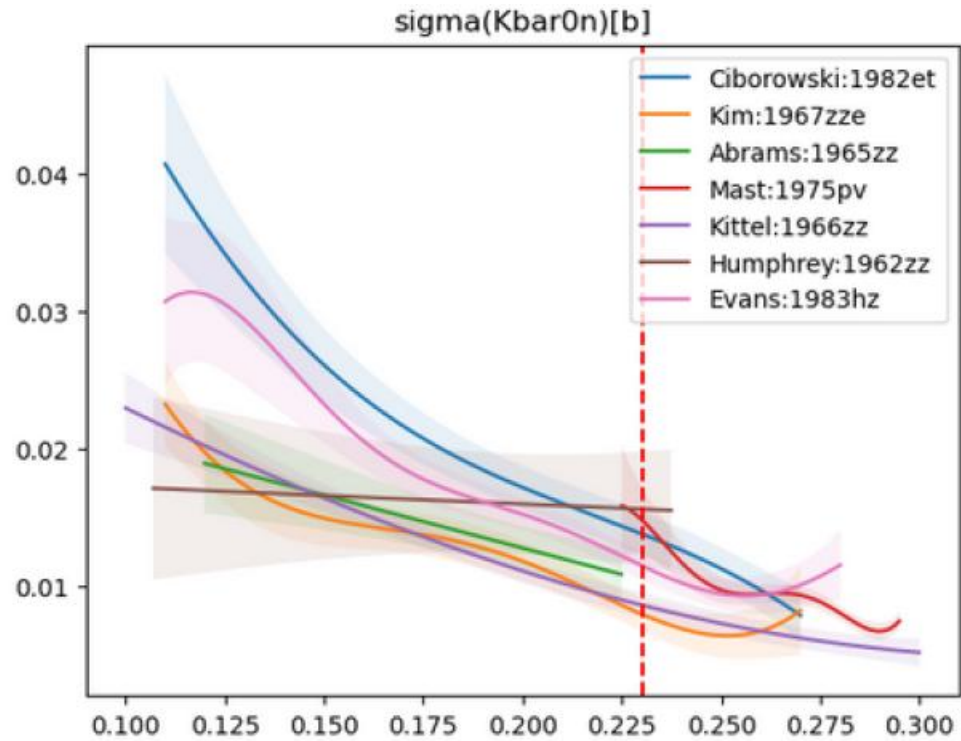
Probability surface

- Taking a 1D projection in energy, each result (from the different GPs) can be assumed to form a normalised gaussian.
- This means that results with larger error bars have a lower amplitude and thus contribute less to the 1D projection.

Probability surface projection

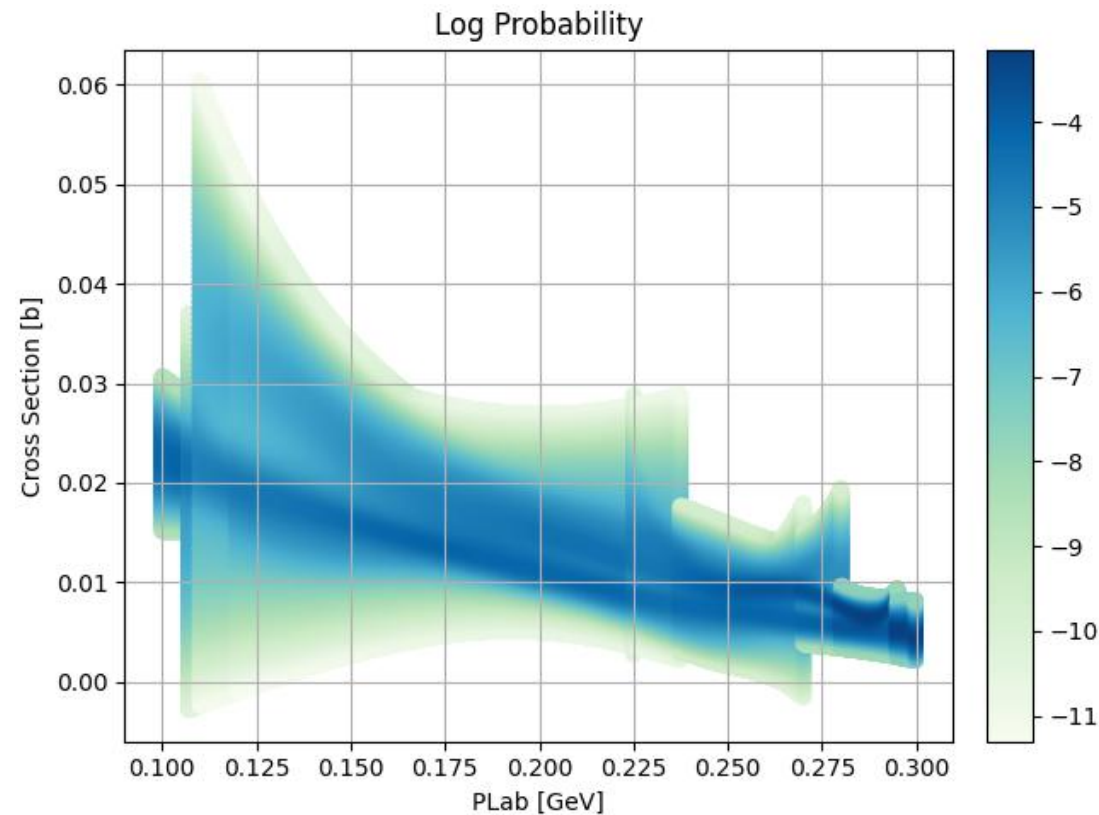


Probability surface projection

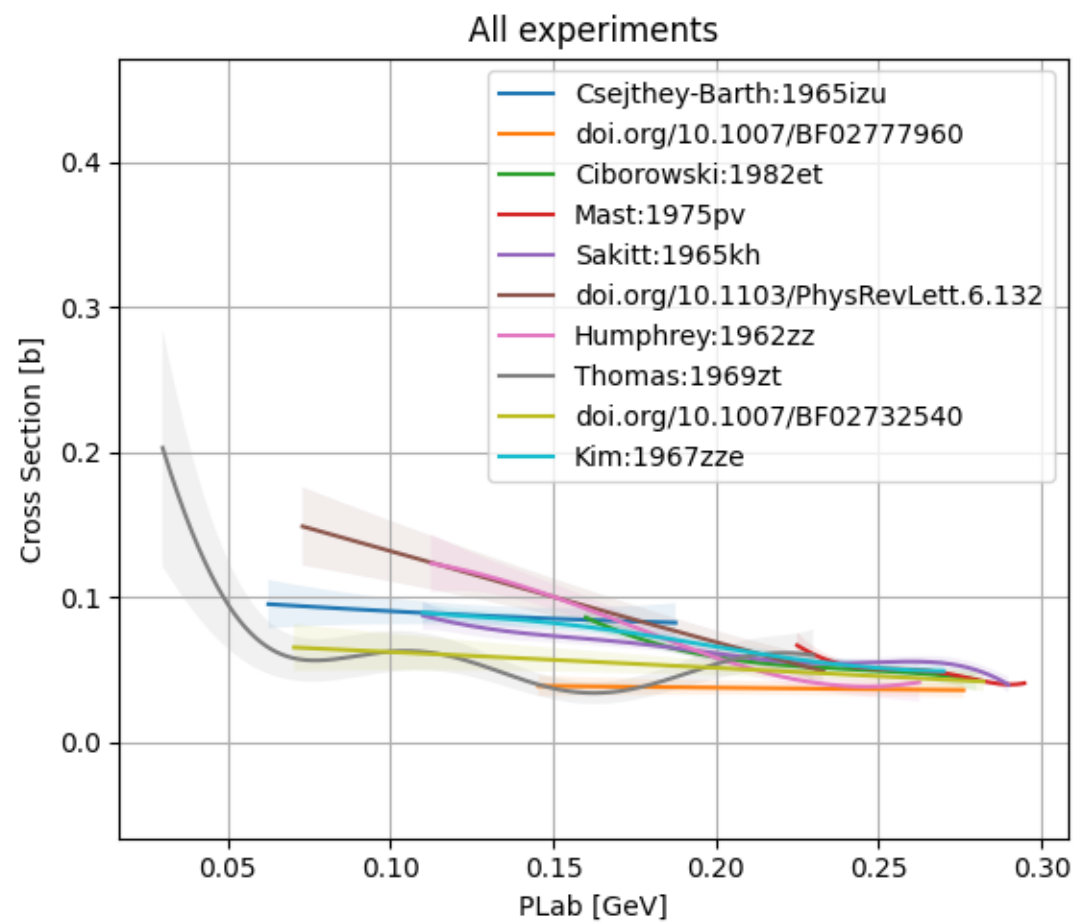
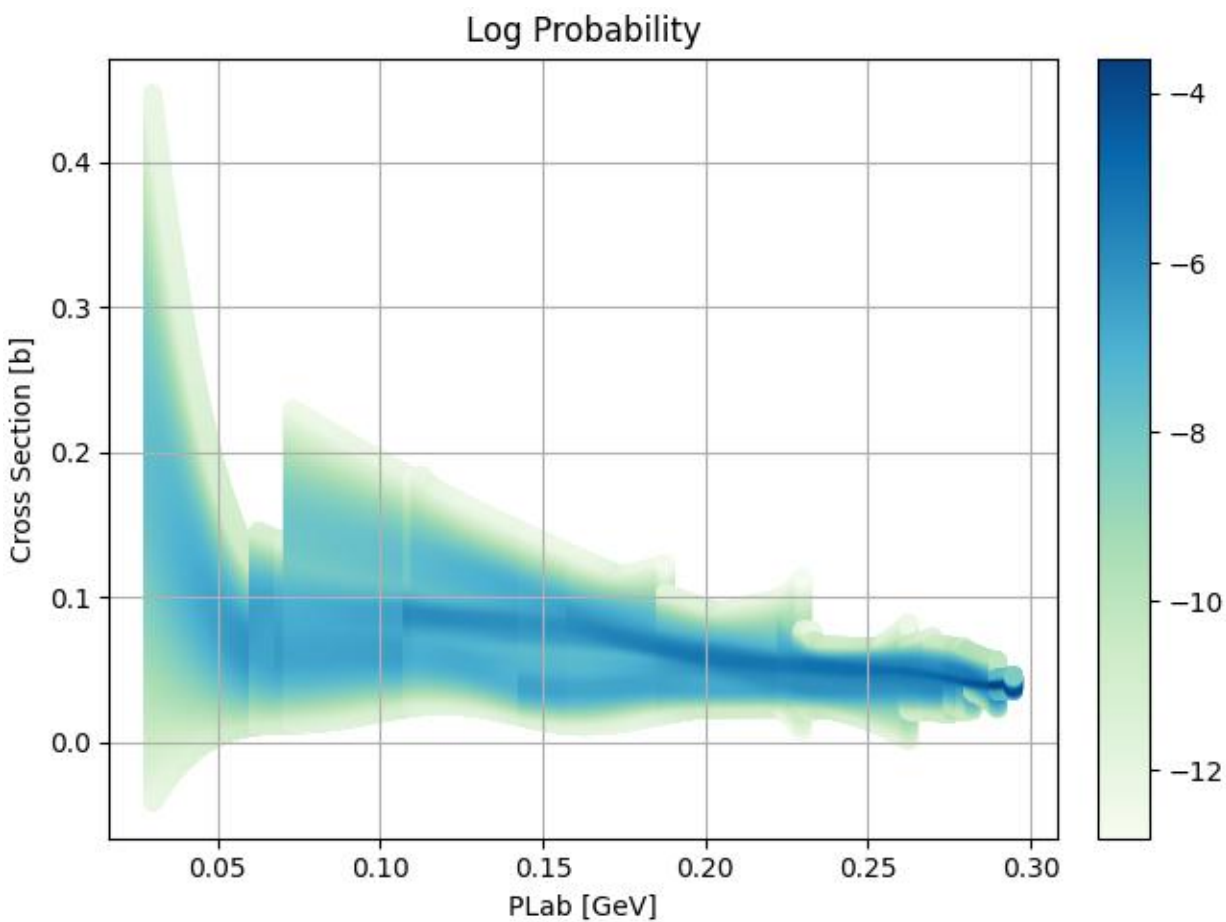


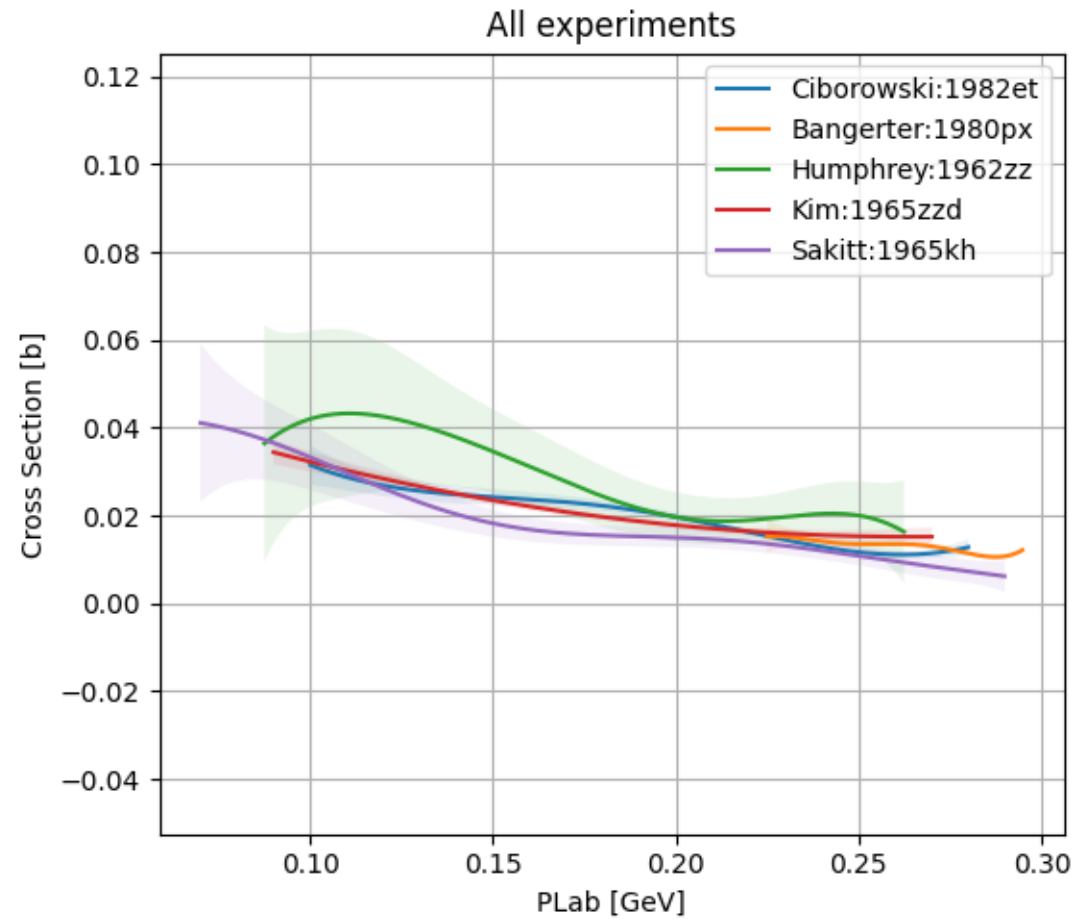
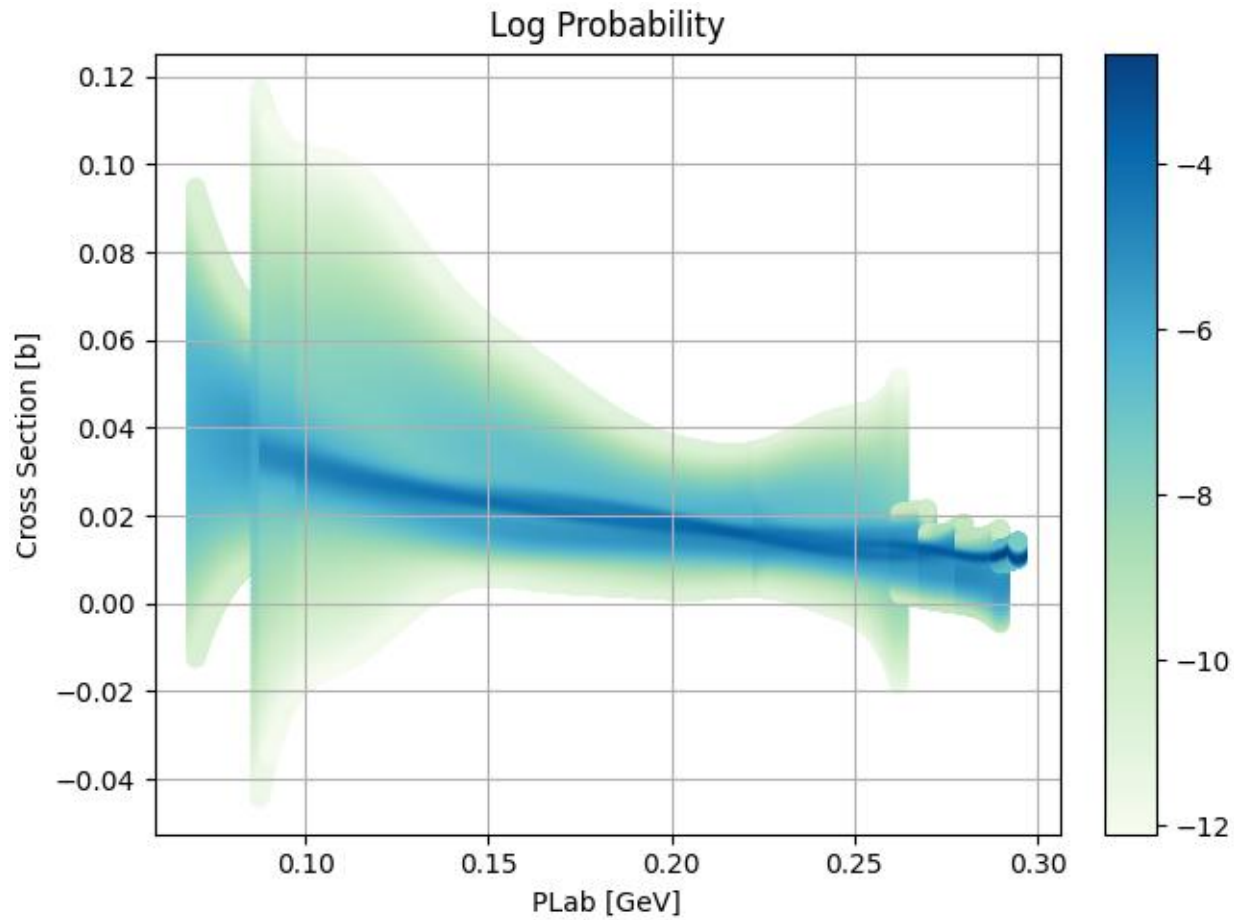
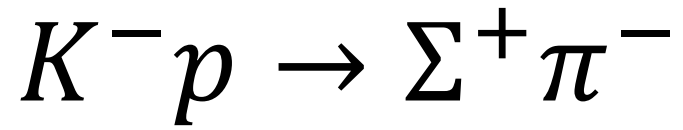
Probability Surface: $K^-p \rightarrow \bar{K}^0 n$

By normalising each slice, the probability of a cross-section value at a given energy can be found:



$$K^- p \rightarrow K^- p$$





Future Work

- By using these probability surfaces, theorists can determine the likelihood for their models in regions with little to no available data.
- This can enable a direct model comparison using MCMC and Bayes factors.
- A technical paper going into more depth will be submitted for publication shortly.
- A public GitHub repository is being written so users can do a GP fit on their own data – please get in touch if you want to be notified when this is available.

Conclusion

- A Gaussian Process is an extremely useful machine learning tool to expand existing, limited datasets, requiring only 3 simple assumptions to operate.
 - Some kernel function can be used to measure the covariance between known datapoints.
 - This same kernel function can predict the covariance of other, unknown datapoints.
 - The style of posterior distribution is known (e.g., smoothness, continuity, periodicity, monotonically increasing, etc.).
- The GP requires no theoretical model assumptions and doesn't need any training.
- The GP has been demonstrated to work on pseudodata modelled on 2D polarisation observables.
- Current research is ongoing on checking consistency of different datasets for the same experiment.

Thanks for listening

Back-up Slides

Mathematical Process I

Assume that we have n known datapoints of the form (\vec{x}_i, y_i) with known errors e_i used to define the expression form $\vec{y} = f(X)$. Here \vec{x}_i is a vector of the kinematic variables (e.g. energy, scattering angle, etc.) and X is a matrix whose rows are \vec{x}_a, \vec{x}_b .

e.g. if we have 3 points a,b,c that have some energy E and scattering angle $\cos\theta$, then X is

$$X = \begin{bmatrix} \vec{x}_a \\ \vec{x}_b \\ \vec{x}_c \end{bmatrix} = \begin{bmatrix} E_a & \cos \theta_a \\ E_b & \cos \theta_b \\ E_c & \cos \theta_c \end{bmatrix}$$

i.e. X will always be a 2D matrix, regardless of the number of input dimensions

Mathematical Process II

Assume that \vec{y} is drawn from a Multivariate Gaussian of the form $p(\vec{y}|X) \sim \mathcal{N}(\vec{0}, K)$ (the zero mean assumption simplifies some maths later but doesn't impact the prediction), where $K = \kappa(X, X, \vec{l}) + \vec{e}^2 I_n$ is the $n \times n$ covariance matrix.

Here κ is some kernel function that is used to measure the covariance and \vec{l} are hyperparameters (more on this later). Here $K_{ab} = \kappa(\vec{x}_a, \vec{x}_b) + \delta_{ab} e_a^2$, where \vec{x}_a, \vec{x}_b are rows of the matrix X .

Mathematical Process III

Assume that there are m known datapoints of the form outlined previously, with known \vec{x}_{*i} with unknown scalars y_{*i} , which are correlated to the n known datapoints.

A matrix X_* can then be generated whose rows are the vectors \vec{x}_* .

As \vec{y}_* is correlated to \vec{y} , they are drawn from the same multivariate Gaussian:

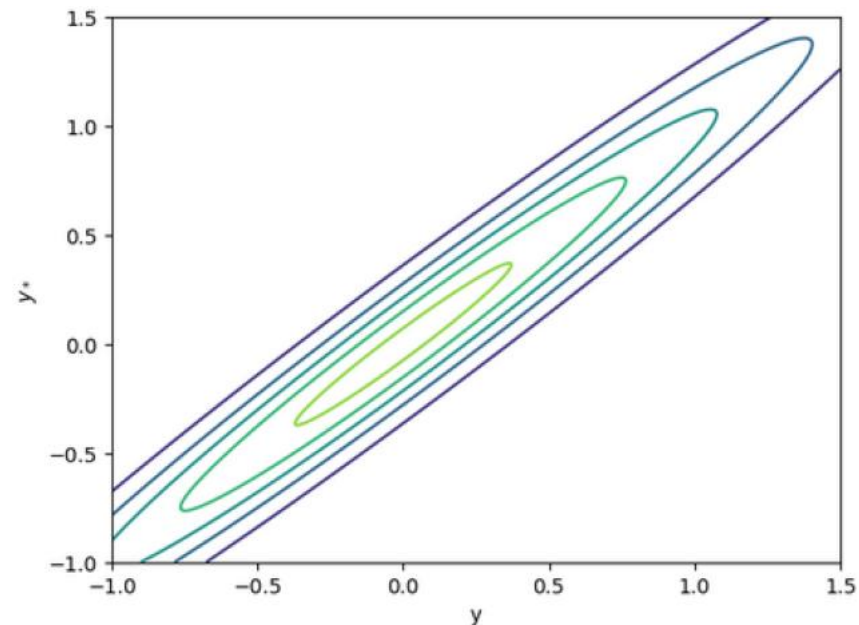
$$\begin{bmatrix} \vec{y} \\ \vec{y}_* \end{bmatrix} \sim \mathcal{N} \left(\underline{0}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right)$$

where $K_* = \kappa(X, X_*)$, $K_{**} = \kappa(X_*, X_*)$.

Essentially, our data can be thought of as a single sample drawn from this multivariate Gaussian.

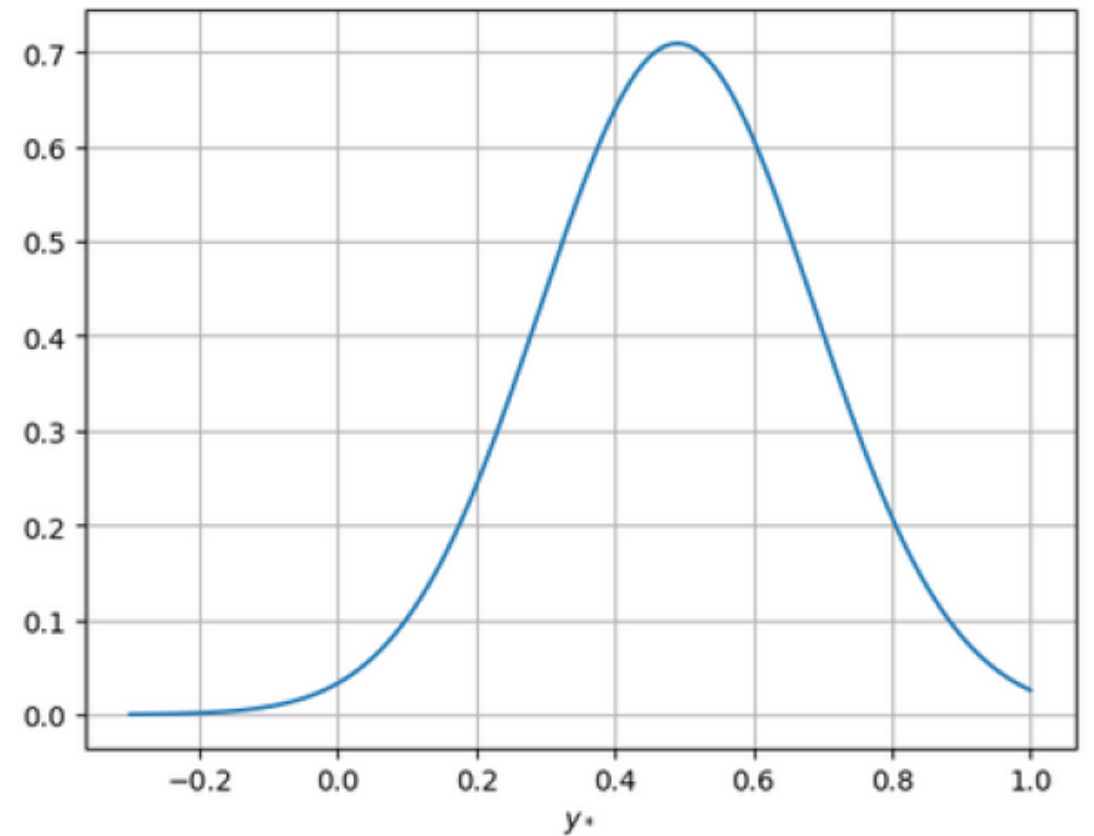
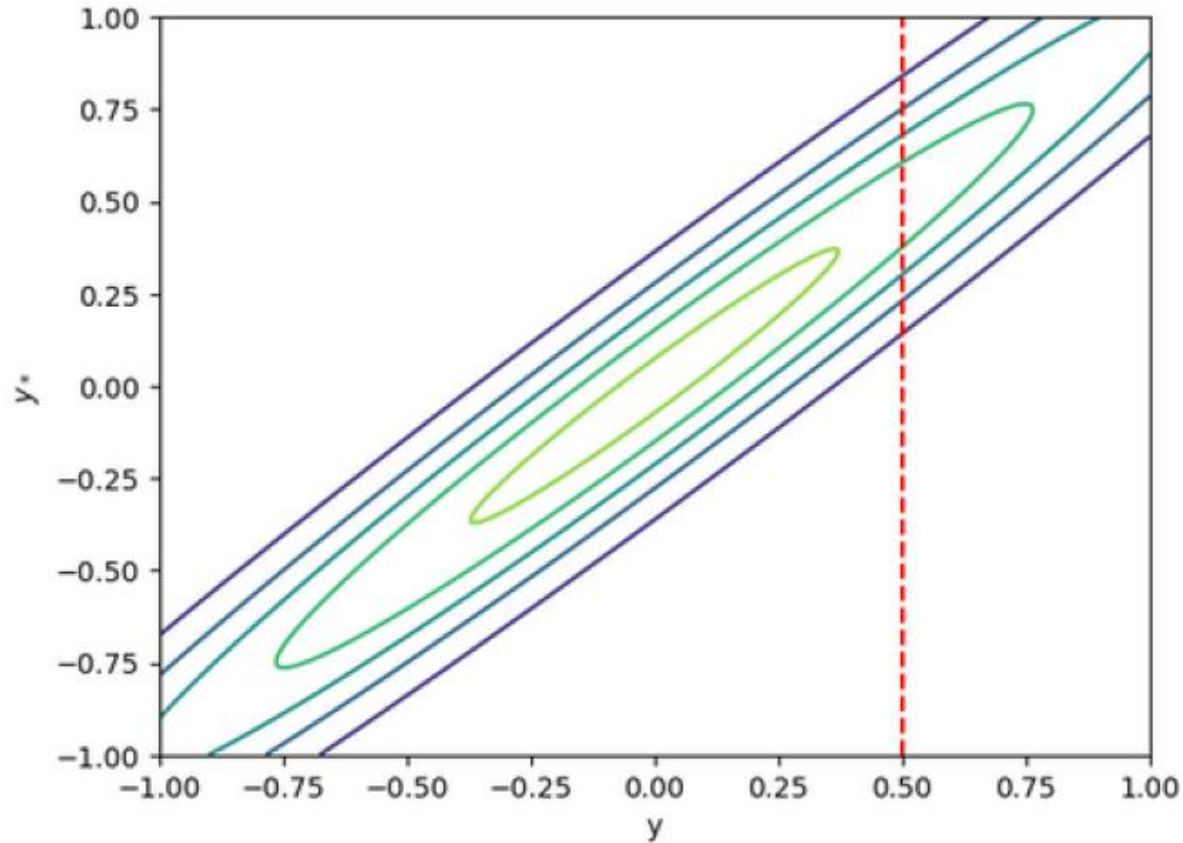
Mathematical Process IV

- Assume we have a single point $(1,0.5)$ and want to get a prediction for $x_* = 0.9$.
- Plugging the values of $x = 1, x_* = 0.9$ into K, K_S, K_{SS} and plotting the multivariate Gaussian gives the graph below
- Note this plot is generated with the specific values $x = 1, x_* = 0.9, l = 1$, changing any of these values will result in a different multivariate Gaussian



Mathematical Process V

We know that $y = 0.5$ so can do a 1D projection to get the value for y_*



Mathematical Process VI

By using the conditional of a multivariate Gaussian, a prediction for \vec{y}_* can be obtained:

$$p(\vec{y}_* | X_*, X, \vec{y}) \sim N(\vec{\mu}_*, \Sigma_*) \text{ where}$$

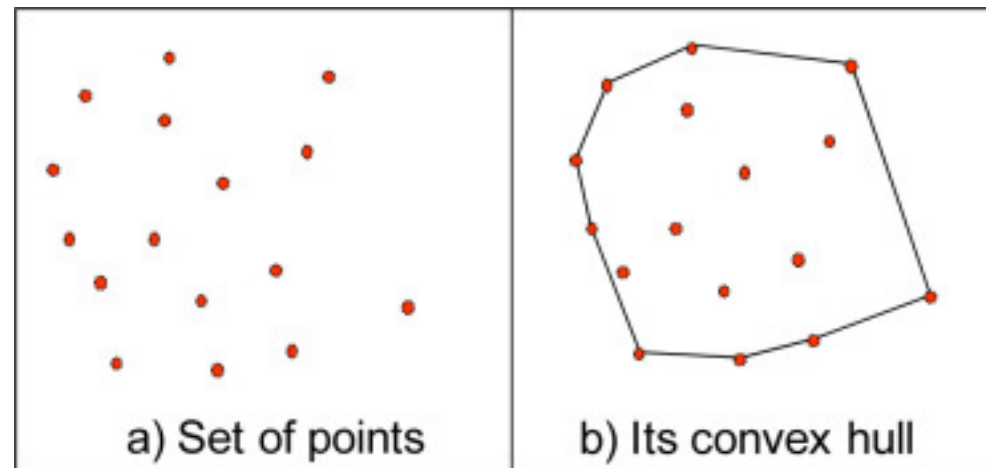
$$\vec{\mu}_* = K_*^T K^{-1} \vec{y}$$

$$\Sigma_* = K_{**} - K_*^T K^{-1} K_*$$

Thus, the GP now has a prediction for the mean and covariance matrix, and thus the standard deviation, of \vec{y}_* .⁵

Convex Hull

- It was found in testing that the GP performs well at interpolating but not at extrapolating.
- As such a set of discrete points of the convex hull of the known datapoints is the space that the GP gives a prediction for (with resolution in each dimension chosen by the user).



Generating Pseudodata I

A generated asymmetry datapoint is based on the effective number of counts measured. This can be expressed as

$$A = \frac{N_+ - N_-}{N_+ + N_-}$$

where N_+ , N_- are used to describe the 2 different states which are used to estimate the effective count. These take into account beam polarisation, recoils, target dilution and other such factors. These random variables are generated from “true” values:

$$N \sim \text{Pois}(n_{\pm})$$

where $n_{\pm} = \frac{1}{2}n_e[1 \pm f(w, \cos \theta)]$. Here n_e is defined as the effective number of events and is in the range [200,1000] which is estimated based on real data.

Generating Pseudodata II

By using standard propagation of errors, the error on A is given by:

$$\delta A = \frac{2}{(N_+ + N_-)^2} \sqrt{N_+ N_- (N_+ + N_-)}$$

Coefficient	Mean of pull distribution from known datapoints fit	Variance of pull distribution from known datapoints fit	Mean of pull distribution from GP datapoints fit	Variance of pull distribution from GP datapoints fit
c_0	0.04	0.91	0.06	0.92
μ_0	-0.04	0.82	-0.05	0.84
σ_0^2	0.0	0.77	-0.01	0.79
c_1	0.04	0.89	0.04	0.91
μ_1	-0.03	0.74	-0.02	0.73
σ_1^2	-0.1	0.77	-0.09	0.78
c_2	-0.06	1.01	-0.06	1.05
μ_2	-0.05	0.73	-0.05	0.75
σ_2^2	-0.17	0.82	-0.17	0.83
c_3	-0.06	0.95	-0.07	0.96
μ_3	-0.02	0.73	-0.04	0.74
σ_3^2	-0.07	0.73	-0.07	0.76

Pseudodata

We can test the GP using some suitable pseudodata. Thus, define a 2D surface of the form, modelled on polarisation observables:

$$y_{func} = f(E_\gamma, \cos \theta) = \sum_{l=0}^n c_l * g_l(E_\gamma) * P_l(\cos \theta)$$

With

- $c_l \in [-1,1]$ is some weight
- $g_l(E_\gamma) \sim \mathcal{N}(\mu_l, \sigma^2_l)$
- $P_l(\cos \theta)$ is an ordinary Legendre polynomial

In our case $n=3$ so we have 12 parameters.

Note also that $|y_{func}| \leq 1$.

2 Tests

A 2D known surface is generated, some points are selected and given appropriate noise and error bars. This is pseudodata which can be used to test the GP is performing as intended.

We can perform 2 tests on this:

- Number of points in different confidence intervals
- Unbiased Pull of Fitted coefficients

Points within confidence intervals

- Calculate pull: $pull = \frac{y_{func} - y_{fit}}{e_{fit}}$
- $|pull| \leq 1 \Rightarrow y_{func} \in [y_{fit} - e_{fit}, y_{fit} + e_{fit}]$, i.e., the predicted point is within its uncertainty of the actual point.
- From this the total percentage of points within different confidence intervals can be calculated by scaling e_{fit} as required and repeat.
- Assuming n known datapoints, select n random GP datapoints and compare.

Points within confidence intervals

Confidence interval	Expected percentage of points within confidence interval (%)	Mean percentage of GP points within confidence interval (%)
0.67σ	50	61.6
1σ	68.3	74.4
1.96σ	95	95.9

Fitting Parameters

- The functional form of the 2D surface can be fitted using the known datapoints as well as the n random datapoints (to check the GP performs well in all kinematic regions).
- The pull of the 12 fitted coefficients compared to the actual coefficients can be calculated for both.

