University of Glasgow

Gaussian Process

Machine Learning for Observable Interpolation and Data Analysis

Ryan Ferguson

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

Contents

1. Why are we doing this?
2. Assumptions
3. How do we know it works?
4. What does real data look like?
5. What are the next steps?

Why are we doing this?

Why?

- Information from hadron data is limited by incomplete and potentially inconsistent datasets.
- Hadronic data often is for different observables and comes from different experiments.
- Often these datasets can be sparsely populated in kinematic regions of interest.
- This fit can prove difficult to accurately constrain when using different observables measured at different kinematic variables.
- Ideally, a process which can reliably provide a value and associated uncertainty at any given kinematic variable should be used. This can be achieved using machine learning.

What can machine learning do?

- A Gaussian Process (GP) can be used to predict the mean and standard deviation of other, unknown, datapoints.
- This can be used to build a more consistent, accurate and complete dataset.
- Datasets from different experiments of the same variable can be compared and checked using some statistical measures.
- The GP could provide significantly improved datasets which theorists can use to test models and check for significant areas of divergence between the GP fit and theoretical models.

How does it work?

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 $\vec{y} = f(X)$.

Assume that $\vec{y}$ is drawn from a Multivariate Gaussian of the form $p(\vec{y}|X) \sim \mathcal{N}\left(\vec{0}, K\right)$, where $K = \kappa(X, X) + \vec{e}^2 I_n$ is the *n* x *n* covariance matrix and *κ* is some kernel function that is used to measure the covariance. 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 II

Assume that there are *m* known datapoints of the form outlined previously, with known $\overrightarrow{x_*}_i$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 $\overrightarrow{x_*}$.

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

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

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

Mathematical Process III

By using the conditional of a multivariate Gaussian, a prediction for $\overrightarrow{y_*}$ can be obtained:

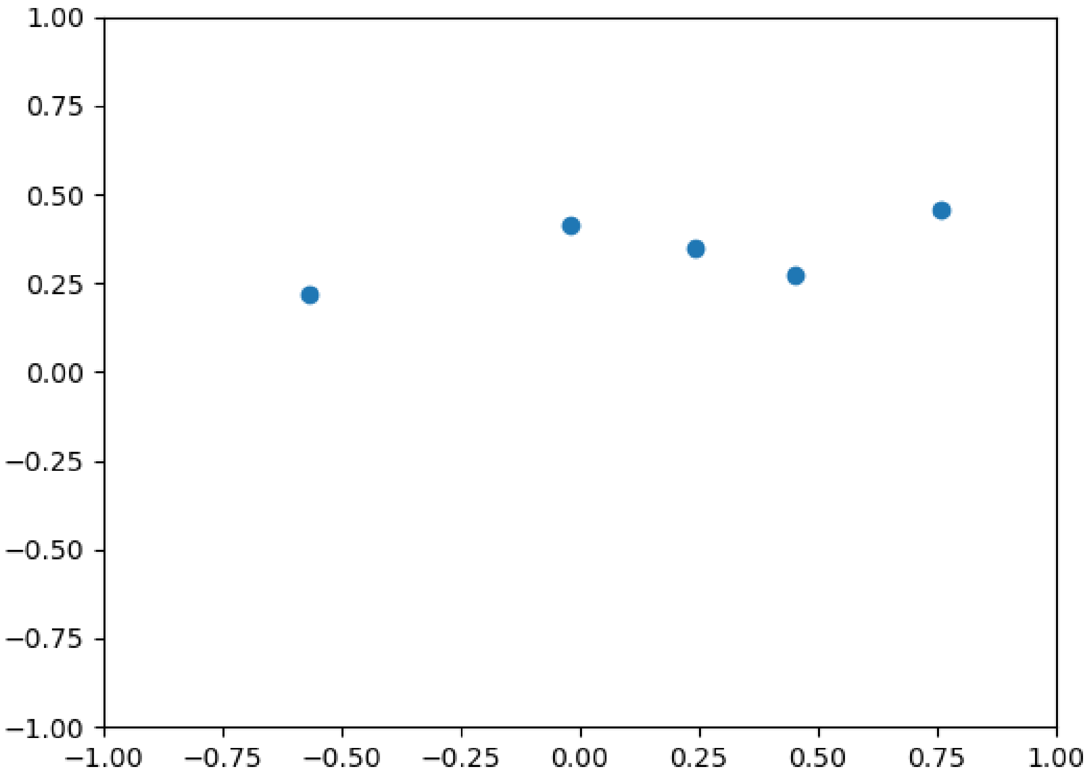
$$p(\overrightarrow{y_*}|X_*, X, \vec{y}\,) \sim N(\overrightarrow{\mu_*}, \Sigma_*) \text{ where}$$

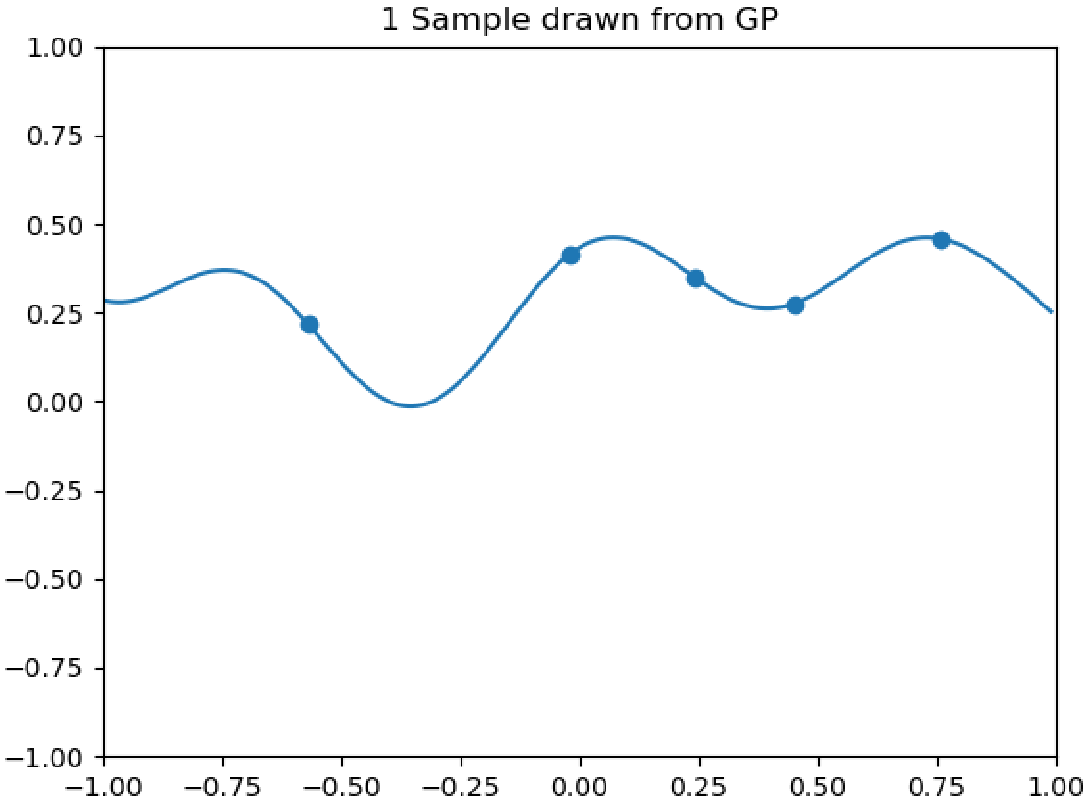
$$\overrightarrow{\mu_*} = {K_*}^T K^{-1} \vec{y}$$

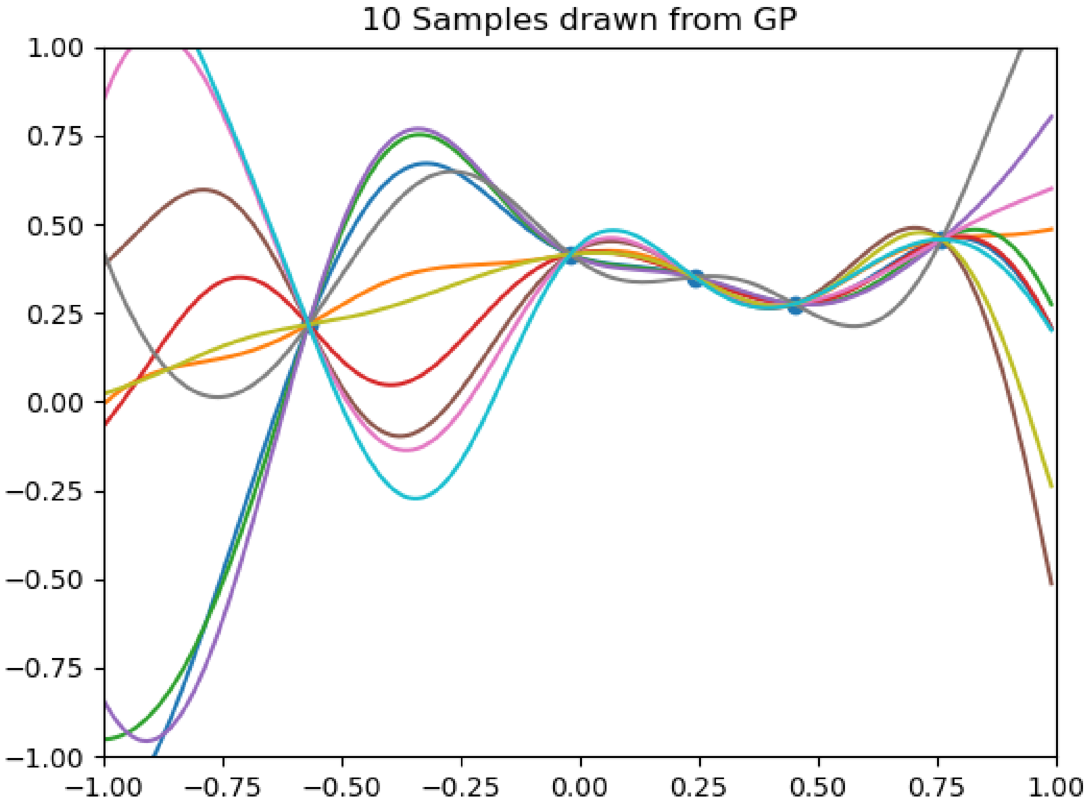
$$\Sigma_* = K_{**} - {K_*}^T K^{-1} K_*$$

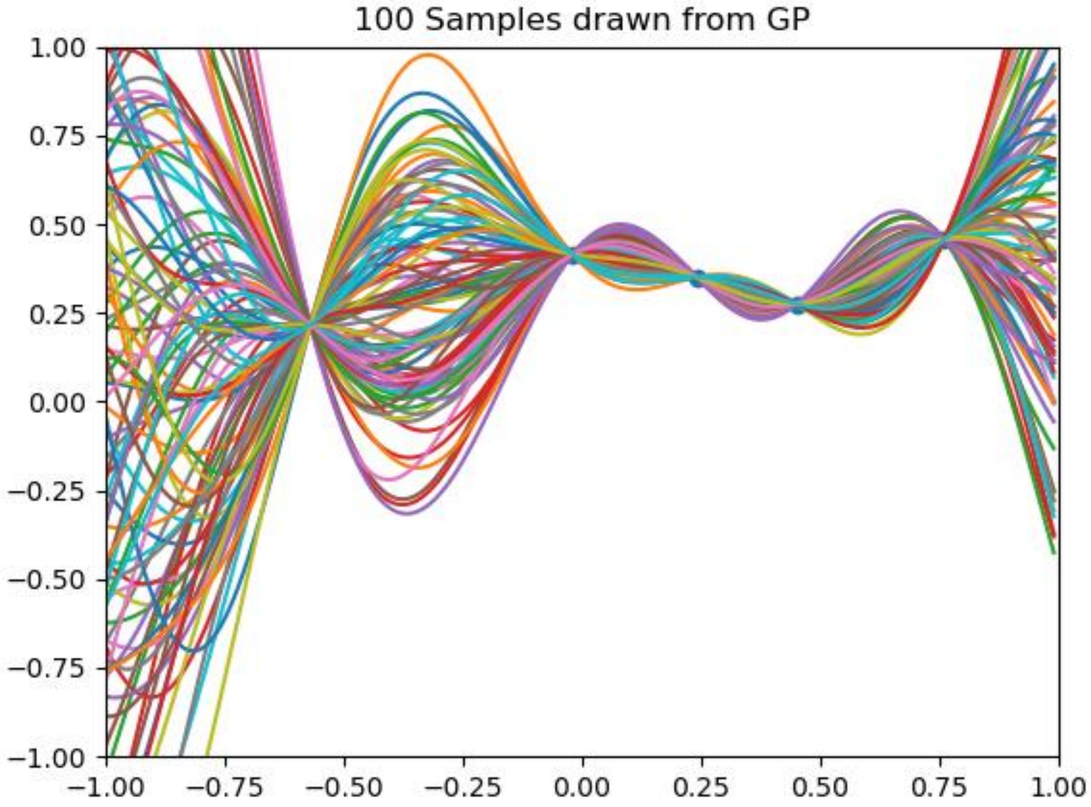
Thus, the GP now has a prediction for the mean and covariance matrix, and thus the standard deviation, of $\overrightarrow{y_*}$. [5]

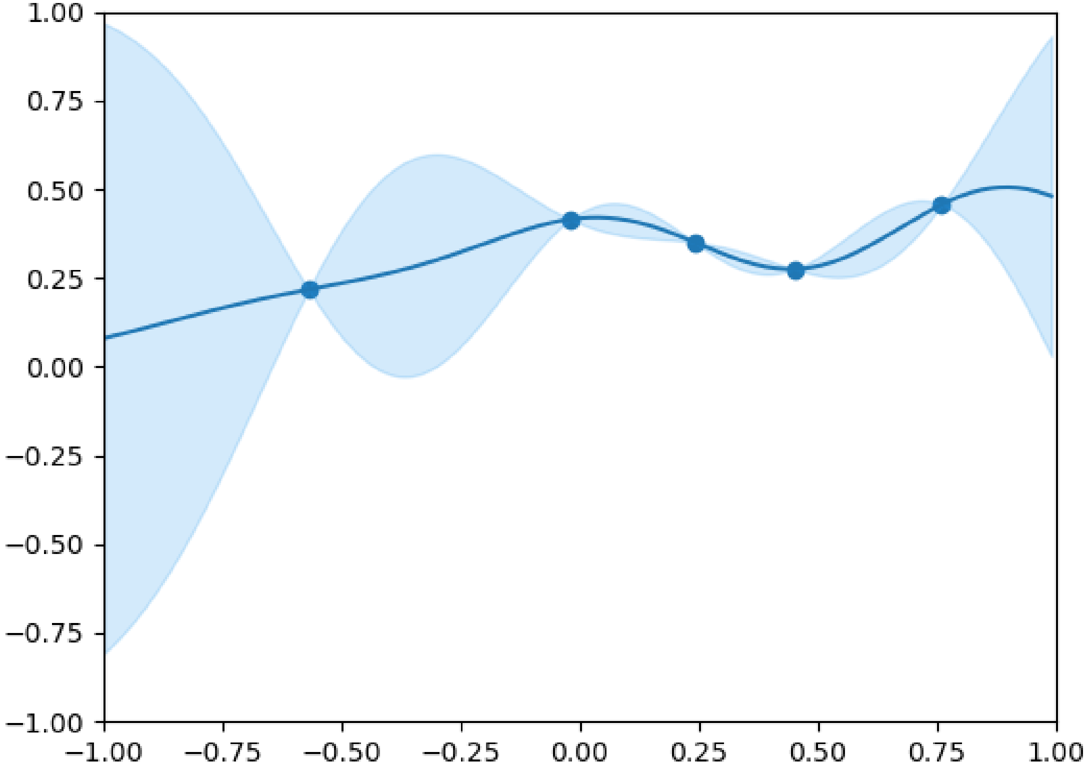
Example


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1 Sample drawn from GP
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


10 Samples drawn from GP
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


100 Samples drawn from GP
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

How do we know it works?

Pseudodata

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

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

With

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

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

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

Radial Basis Function Kernel

Various kernels can be used depending on the desired output, e.g. smoothness, periodicity, etc. Here the simplest kernel, the radial basis function (RBF), is tested:

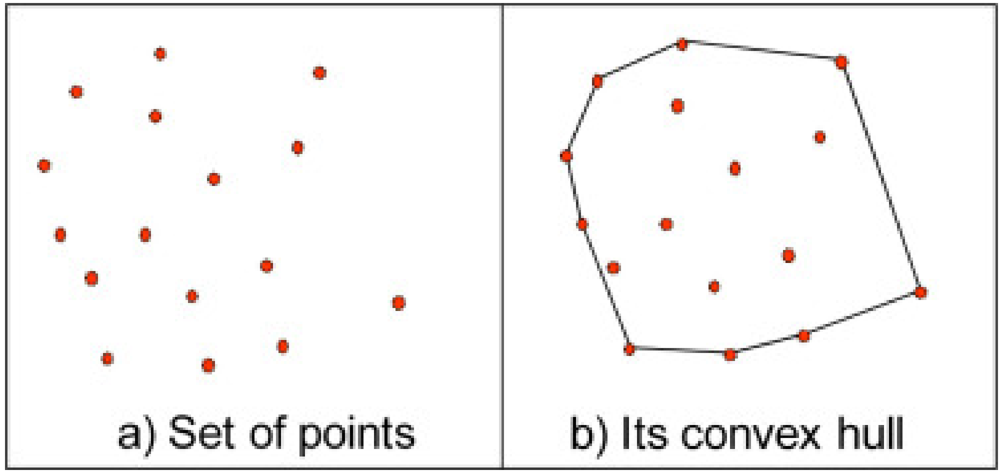
$$\kappa\left(\underline{a}, \underline{b}\right) = \exp\left[\sum_{i=0}^{p-1} \frac{-d(a_i, b_i)^2}{2{l_i}^2}\right]$$

Where:

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

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[1] of the known datapoints is the space that the GP gives a prediction for (with resolution in each dimension chosen by the user).



3 Tests

We can perform 3 tests on the pseudodata output to check the GP is performing as intended:

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

Unbiased Pull

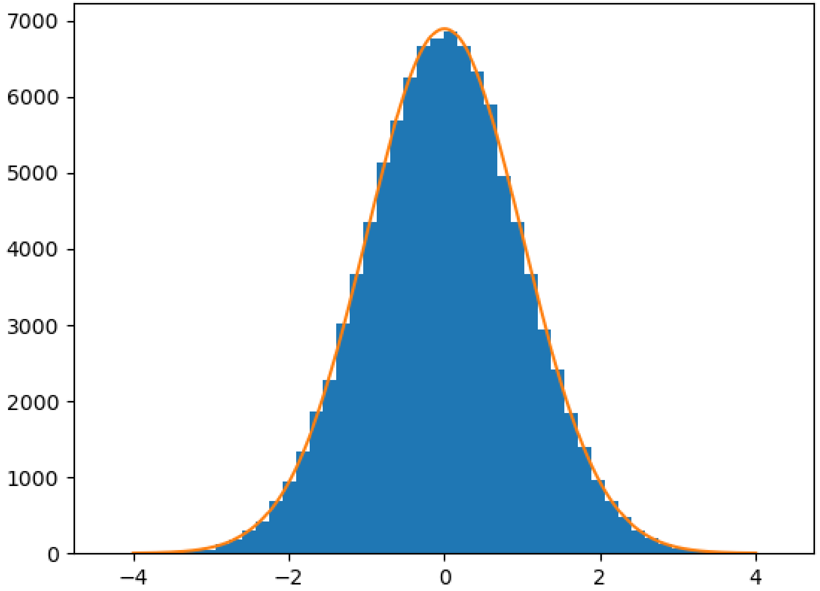
- Calculate pull: $pull = \frac{y_{func} - y_{fit}}{e_{fit}}$
- For each surface, check the pull distribution mean and variance, which should be 0 and 1, respectively.
- Check the pull distribution of the GP fit at the same energies and angles as the “known” datapoints.
- Calculate the mean and variance of both pull distributions for every generating surface.

Unbiased Pull

Variance of 1

Gaussian centred at 1

Repeat over 1000 different surfaces

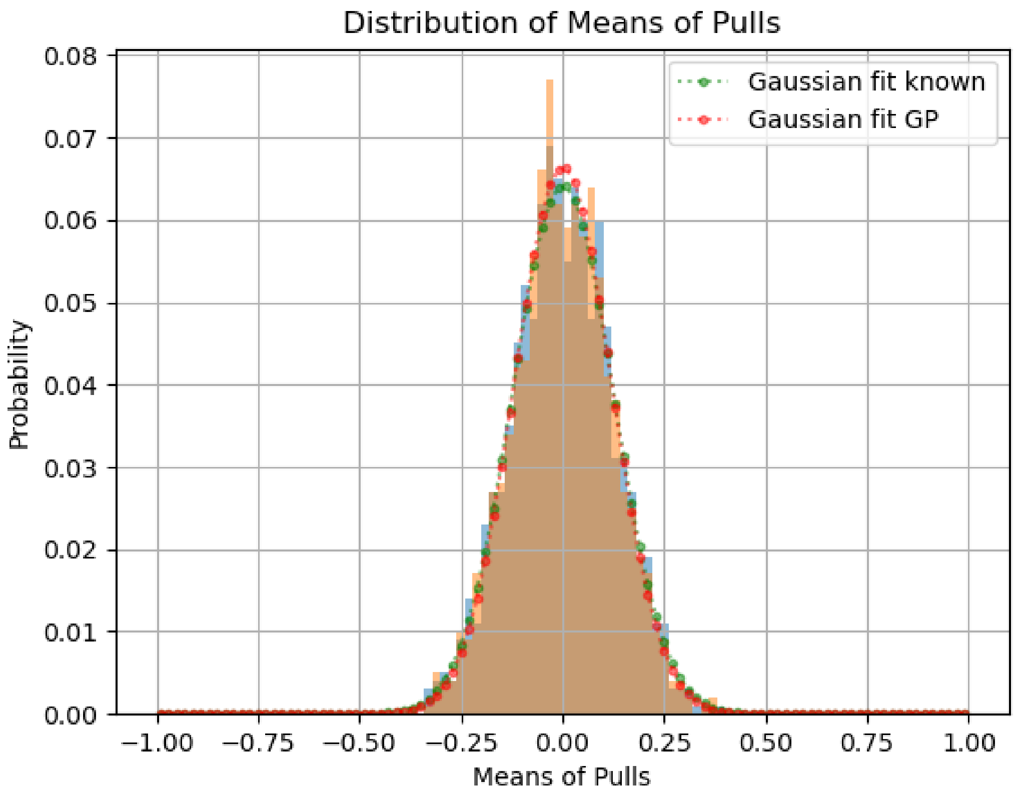


Mean of 0

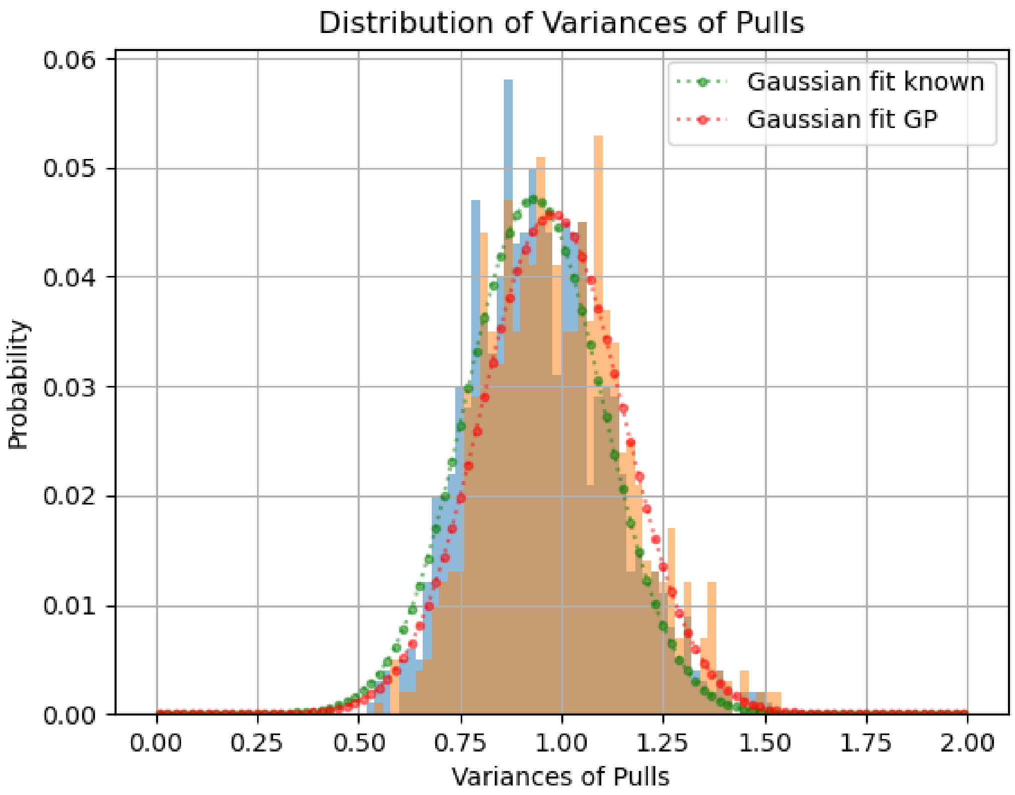
Gaussian centred at 0

Repeat over 1000 different surfaces

Unbiased Pull



Centred at 0



Centred at 1

Points within confidence intervals

- Calculate pull: $pull = \frac{y_{func} - y_{fit}}{e_{fit}}$
- $|pull| \leq 1 \Longrightarrow y_{func} \epsilon \left[y_{fit} - e_{fit} \, , y_{fit} + e_{fit} \right]$, i.e., the predicted point is within its uncertainty of the actual point.

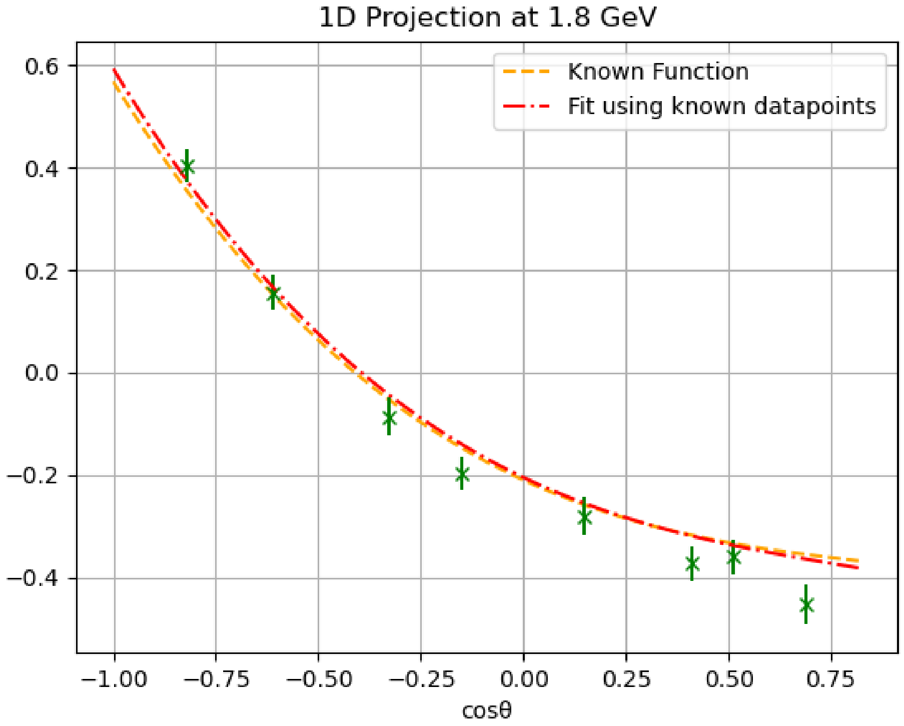
- From this the total percentage of points within different confidence intervals can be calculated by scaling e_{fit} as required and repeat.

Points within confidence intervals

Confidence interval	Expected percentage of points within confidence interval (%)	Mean percentage of points within confidence interval (%)
0.67σ	50	84.5
1σ	68.3	94.6
1.96σ	95	99.7

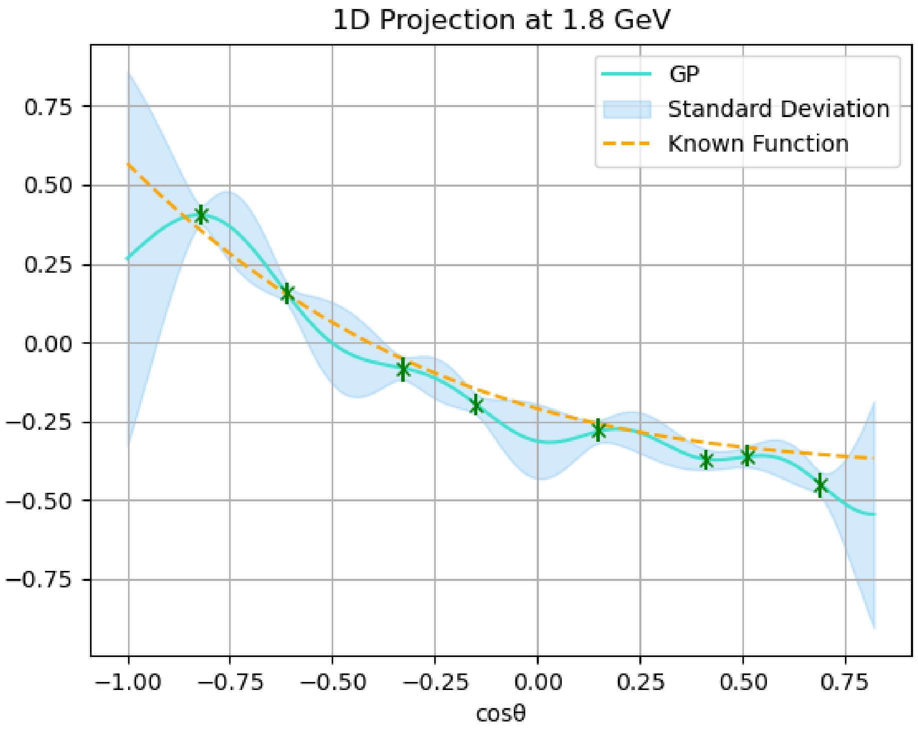
Fitting Parameters

The functional form of the 2D surface can be fitted to some datapoints, using a least squares method, shown below:



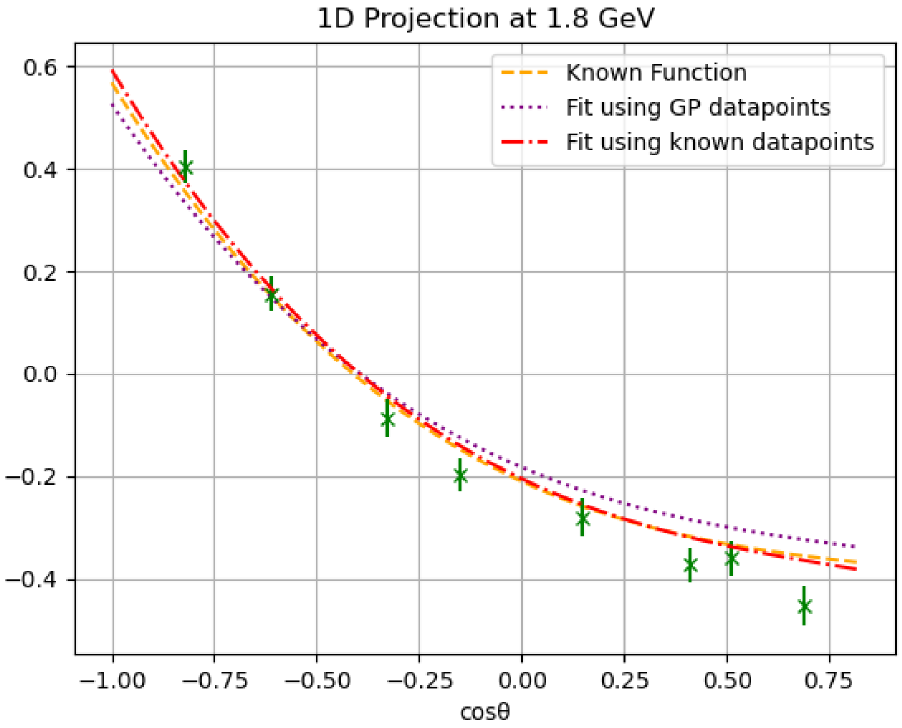
Fitting Parameters

A Gaussian Process fit is then performed on the same datapoints:



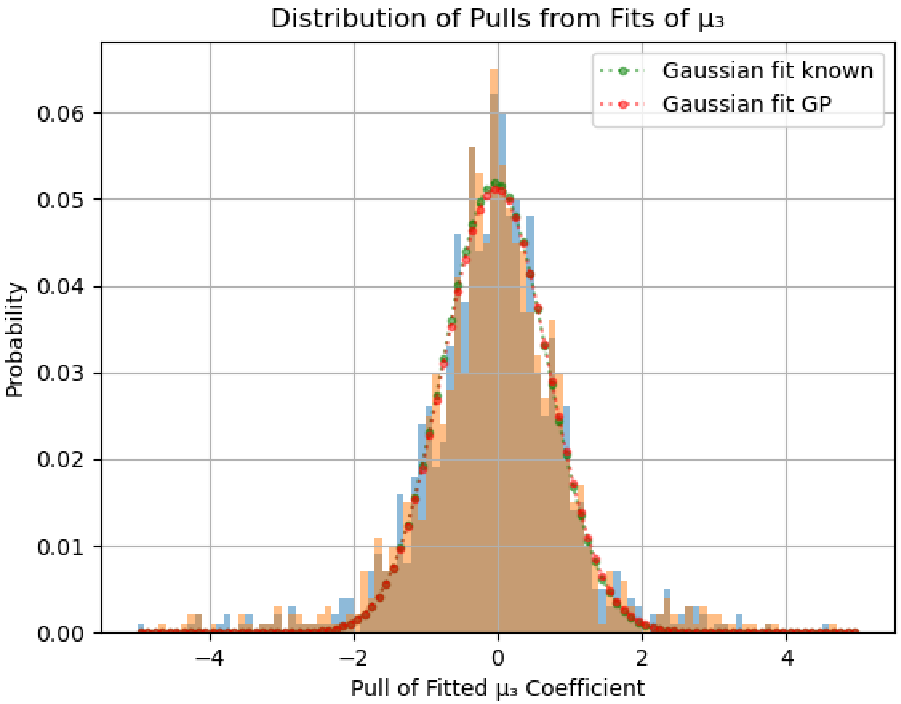
Fitting Parameters

The GP datapoints are used to fit the functional form of the 2D surface:



Fitting Parameters

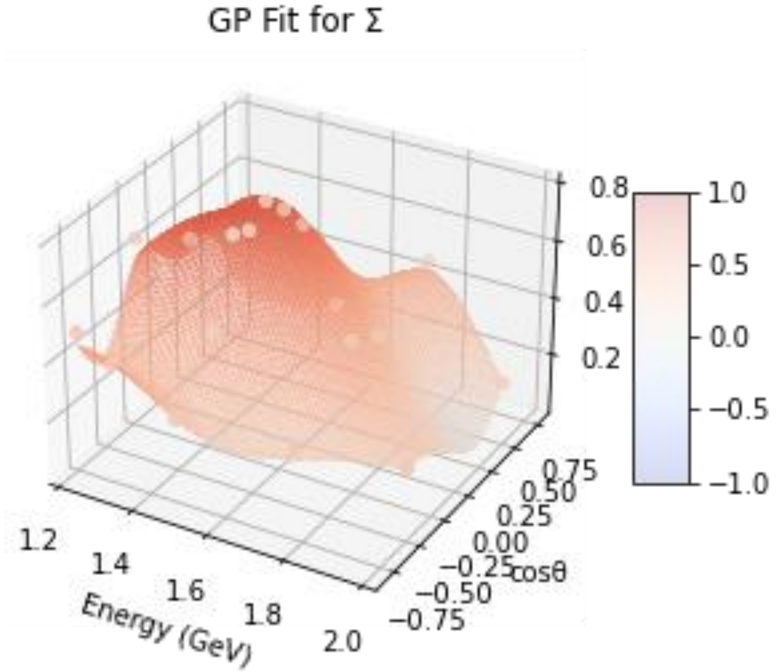
This can be further verified by finding the pull distribution of each of the surface coefficients which should be Gaussians centred at 0 with width 1. An example of one coefficient, μ_3, is shown below:

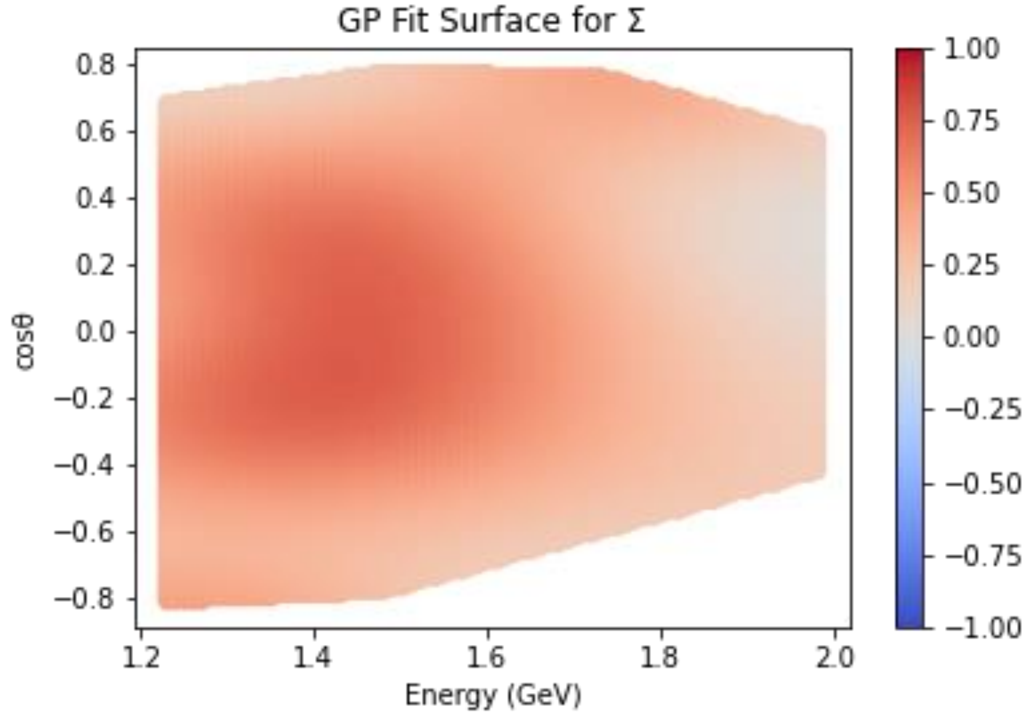


What does real data look like?

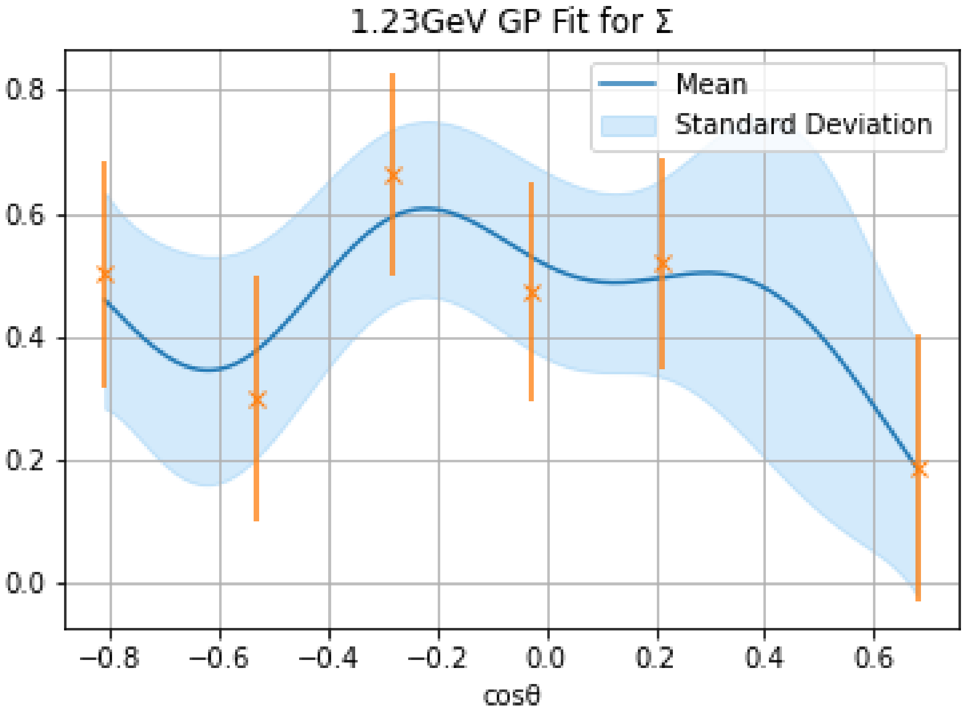
Data from CLAS

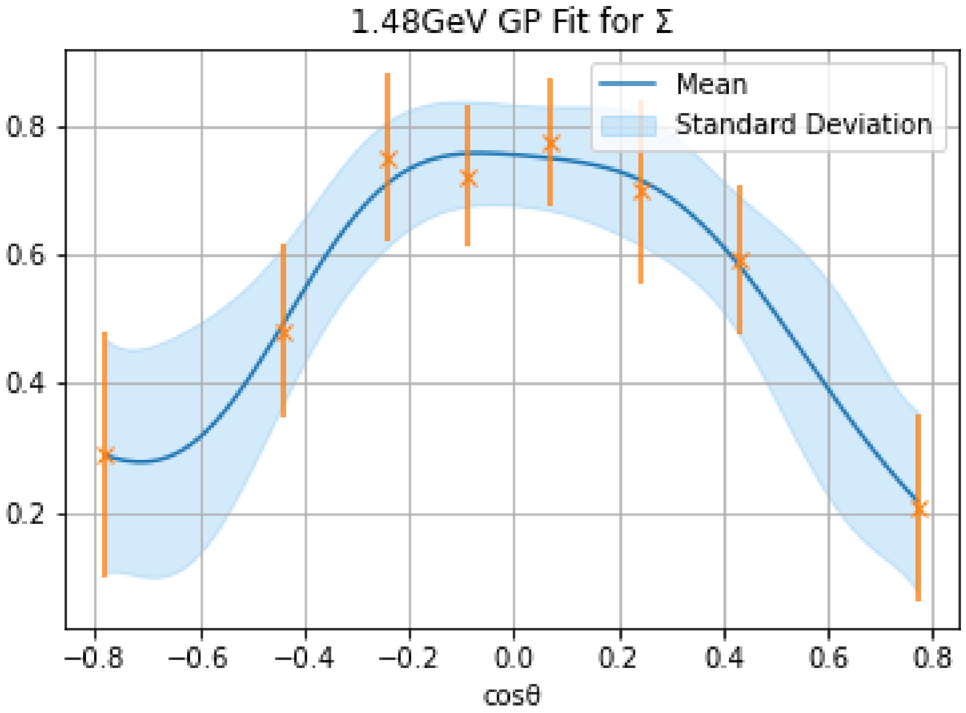
The GP has been used on data recently submitted for publication by the CLAS collaboration at Jefferson Lab, specifically 5 polarisation observables (Σ, P, T, O_x and O_z) of the $K^0\Sigma^+$ reaction.[2] Example plots for Σ:



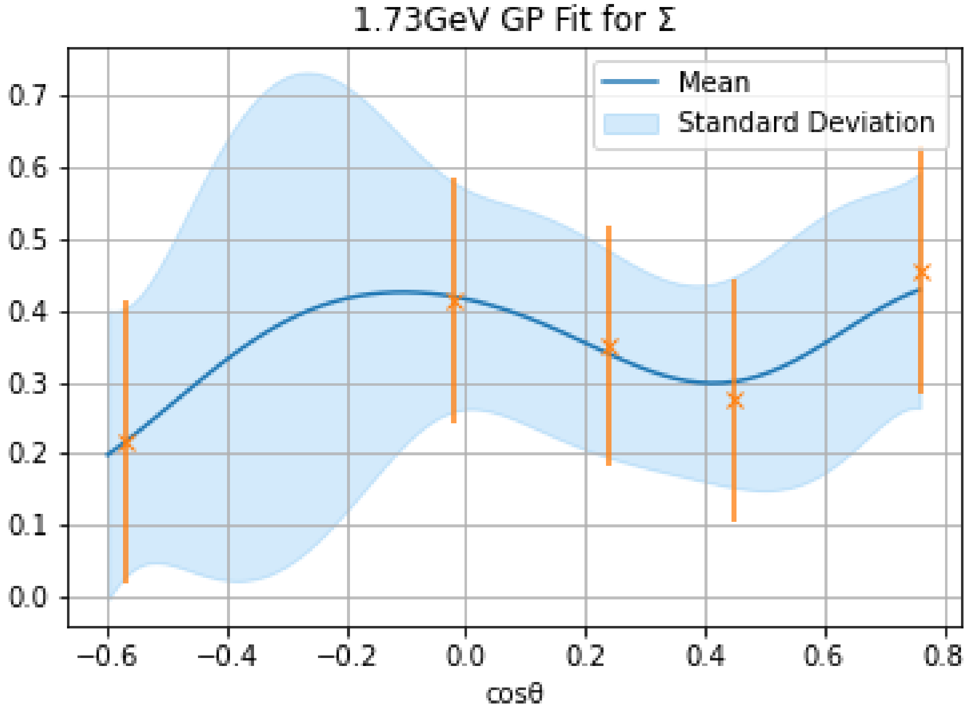


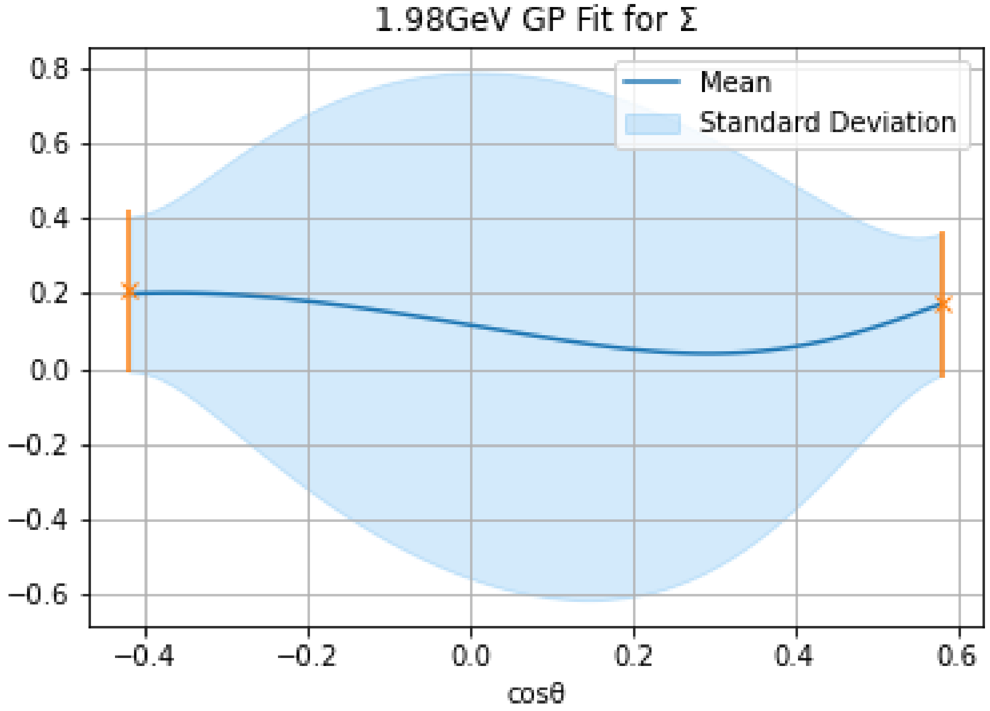
GP 1D Projections for Σ





GP 1D Projections for Σ





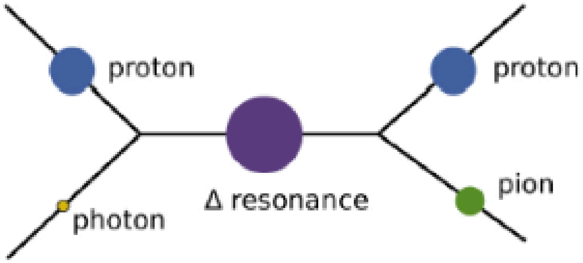
What are the next steps?

Comparing Datasets

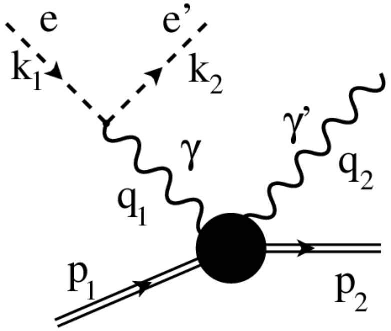
- Additional work is also ongoing to develop a methodology to check the consistency between different datasets of the same variable.
- This will enable theorists to use an expanded datasets to test theories, build more rigorous models, etc.

Expanding to Higher Dimensions

- Testing is underway to expand the GP to higher dimensions, ensuring it still passes the 3 tests shown here.
- Current testing is in 5 dimensions, based on data of Deeply Virtual Compton Scattering (DVCS) of the pion, but other physics quantities are planned.



Photomeson Production in 2D.[3]



DVCS in 5D.[4]

Conclusion

- A Gaussian Process is an extremely useful machine learning tool to expand existing, limited datasets, requiring only 3 simple assumptions to operate.
- The GP has been demonstrated to work on pseudodata modelled on 2D polarisation observables.
- Work is ongoing to expand to other physics quantities and to higher dimensions (particularly DVCS of pions in 5D) and to develop a metric for testing if 2 datasets are consistent with one another.

Thanks for listening

References

1. R. Laurini, *Geographic Knowledge Infrastructure*, 2017, www.sciencedirect.com/topics/earth-and-planetary-sciences/convex-hull [accessed May 28th 2024]
2. L. Clark et al, *Photoproduction of the Σ+ hyperon using linearly polarized photons with CLAS,* 2024, https://arxiv.org/abs/2404.19404 [accessed May 30th 2024]
3. F. Rieger, *HIGH ENERGY ASTROPHYSICS - Lecture 9,* 2024, www.mpi-hd.mpg.de/personalhomes/frieger/HEA9.pdf [accessed May 30th 2024]
4. R. Fiore et al, *Kinematically complete analysis of the CLAS data on the proton structure Function F2 in a Regge-Dual model,* 2006, https://www.researchgate.net/publication/46776238_Kinematically_complete_analysis_of_the_CLAS_data_on_the_proton_structure_Function_F2_in_a_Regge-Dual_model?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ [accessed May 30th 2024]
5. M. Krasser, *Gaussian processes.* 2018. URL: krasserm.github.io/2018/03/19/gaussian-processes/ [accessed May 26th 2024]

Back-up Slides

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:

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

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:

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

Length Scale Calculation - Energy

The mean distance between adjacent measured energy levels. This is mathematically expressed as (assuming n measured energy levels):

$$L_{E_\gamma} = \frac{1}{n-1}\sum_{j=1}^{n-1}\left[e_{j-1} - e_j\right]$$

Length Scale Calculation - Angle

For each measured energy level calculate the mean distance between adjacent measured, degenerate datapoints. Take the resulting mean of these values. This is expressed mathematically as (where m_j is the number of datapoints measured at the *j-th* energy level):

$$L_{\cos\theta} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{1}{m_j - 1}\sum_{i=1}^{m_j-1}\left[a_{j,i+1} - a_{j,i}\right]\right)$$

Resolution Choice

Any “reasonable” choice of resolution for a given dimension is acceptable. Specifically, assume that D is the set of all measured points in a given dimension and R is the resolution of this dimension, then mathematically:

$$\forall d_1, d_2 \in D, \exists\ z \in \mathbb{Z}\ s.t.$$
$$d_1 - d_2 = zR$$

Or equivalently:

$$\forall d \in D, \exists\ z \in \mathbb{Z}\ s.t.$$
$$\min(D) - d = zR$$

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
${\sigma_0}^2$	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
${\sigma_1}^2$	-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
${\sigma_2}^2$	-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
${\sigma_3}^2$	-0.07	0.73	-0.07	0.76