

# Contents

<b>1</b>	<b>Evaluating Cross Sections with Gaussian Processes</b>	<b>1</b>
1.1	Data Generation . . . . .	1
1.1.1	Data Quality . . . . .	4
1.2	Dataset Transformations . . . . .	5
1.3	Learning a GP . . . . .	7
1.3.1	The Benchmark . . . . .	8
1.3.2	Outliers . . . . .	8
1.3.3	Cuts on Cross Sections . . . . .	10
1.3.4	Features . . . . .	11
1.3.5	Kernel . . . . .	12
1.4	Distributed Gaussian Processes . . . . .	12
1.4.1	Adding Experts . . . . .	12
1.4.2	Cross validation for DGP . . . . .	15



# Chapter 1

## Evaluating Cross Sections with Gaussian Processes

### 1.1 Data Generation

In this section the generation of MSSM-24 training and test data is discussed, following closely the discussion in [4].

#### Sampling of Data

An MPI parallelized script generates a sample point in parameter space, by drawing random values from the distributions in Tab. (1.1). The table contains log and flat priors which are combined to cover more of the parameter space. When a parameter point has been sampled, it is run through the program `softpoint.x` which calculates the SUSY spectrum of this point, using the `Softsusy 3.6.2`-package [1]. The spectrum is then written to a `slha`-file and given as input to `Prospino 2.1`, which calculates the LO and NLO cross sections according to the calculations outlined in Section 3.4.1. The relevant features and NLO cross sections are harvested to `.dat`-files, which are used by the GP.

#### Priors

To get a reasonable distribution in parameter space, an objective prior distribution of parameters should be used. This means that priors are assigned according to a set of principles of how information should be encoded in a probability distribution. More details on objective priors can be found in [3].

The first of these principles is the *transformation group invariance*, which states that the probability distribution should be invariant under any transformation that is considered irrelevant to the problem. In other words, the *pdf*

Parameter	Log prior range	Flat prior range
$M_1$	[0,100,4000]	[0,4000]
$M_2$	[0,100,4000]	[0,4000]
$M_3$	[0,100,4000]	[0,4000]
$A_t$	[-4000, -100, 100, 4000]	[-4000, 4000]
$A_b$	[-4000, -100, 100, 4000]	[-4000, 4000]
$A_\tau$	[-4000, -100, 100, 4000]	[-4000, 4000]
$\mu$	[-4000, -100, 100, 4000]	[-4000, 4000]
$m_A^{\text{pole}}$	[0,100,4000]	[0,4000]
$\tan \beta$	[2, 60]	[2, 60]
$m_{L_1}$	[0, 100, 4000]	[0, 4000]
$m_{L_2}$	[0, 100, 4000]	[0, 4000]
$m_{L_3}$	[0, 100, 4000]	[0, 4000]
$m_{e_1}$	[0, 100, 4000]	[0, 4000]
$m_{e_2}$	[0, 100, 4000]	[0, 4000]
$m_{e_3}$	[0, 100, 4000]	[0, 4000]
$m_{Q_1}$	[0, 100, 4000]	[0, 4000]
$m_{Q_2}$	[0, 100, 4000]	[0, 4000]
$m_{Q_3}$	[0, 100, 4000]	[0, 4000]
$m_{u_1}$	[0, 100, 4000]	[0, 4000]
$m_{u_2}$	[0, 100, 4000]	[0, 4000]
$m_{u_3}$	[0, 100, 4000]	[0, 4000]
$m_{d_1}$	[0, 100, 4000]	[0, 4000]
$m_{d_2}$	[0, 100, 4000]	[0, 4000]
$m_{d_3}$	[0, 100, 4000]	[0, 4000]

**Table 1.1:** Table showing the sampling intervals used for the parameters when sampling the MSSM-24 model, where the soft breaking scale is set to  $Q = 1$  TeV. The log priors have three and four limit values, which are of the form [start\_flat, start\_log, end\_log] and [start\_log, start\_flat, start\_log, end\_log]. All values in GeV except  $\tan \beta$  which is unitless. Table from [4].

should satisfy

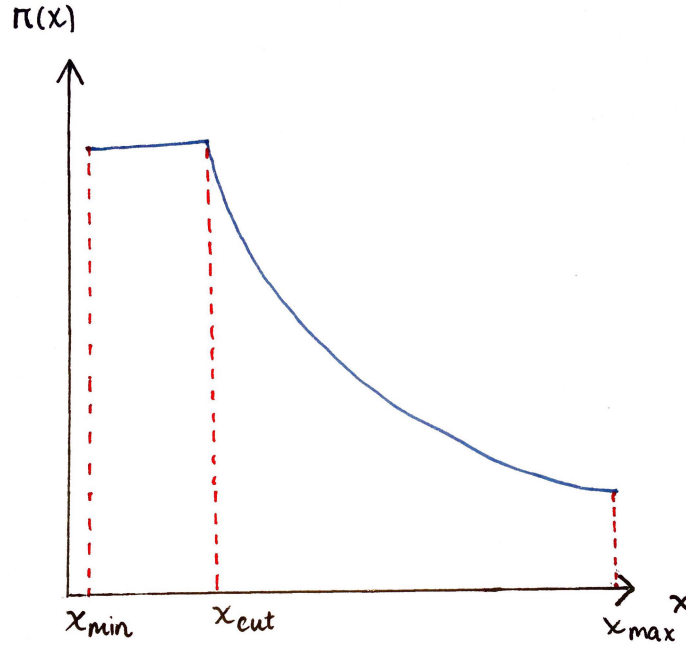
$$\pi(x|I)dx = \pi(x+a|I)d(x+a) \Rightarrow \pi(x|I) = \pi(x+a|I), \quad (1.1)$$

where  $a$  is some translation. This is often referred to as a *flat prior*. Invariance under scale transformations, which are transformations that introduce a scale to the problem  $m \rightarrow m' = cm$ , requires

$$\pi(m|I)dm = \pi(cm|I)cdm, \quad (1.2)$$

which is satisfied if  $\pi(m|I) \propto 1/m$ . Since this corresponds to  $\pi(\log m|I)$  being flat, it is called the *log prior*.

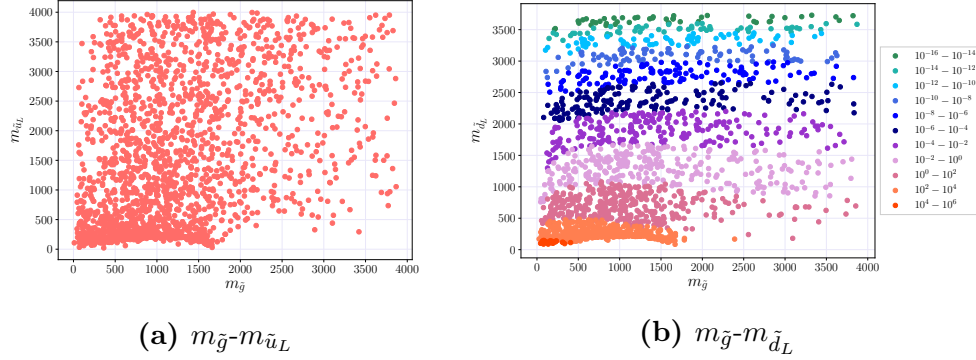
The flat prior covers the edges of parameter space well, while a log prior covers the inner points <sup>1</sup>. A combination of the log and flat priors is therefore used in order to properly cover parameter space. To avoid divergence of the log prior close to zero, a cut is used. The limits on the priors are `[start, flat_start, flat_end, end]` for priors that include negative values, and `[flat_start, start, end]` for priors with positive values. An illustration of a prior with positive values is shown in Fig. (1.1), where a flat distribution is used close to  $x = 0$  to avoid divergences.



**Figure 1.1:** Illustration of the log prior distribution  $\pi(x)$ . Around  $x = 0$  the prior would blow up, so at a limit  $x_{cut}$  a flat prior is used.

---

<sup>1</sup>The points close to 0.



**Figure 1.2:** Data quality plots of the distribution of mass parameters  $m_{\tilde{d}_L}$ ,  $m_{\tilde{u}_L}$  and  $m_{\tilde{g}}$  for 2000 points. In (b) different orders of magnitude of the cross sections for  $\tilde{d}_L \tilde{d}_L$  are shown in different colors, indicating that there are few points in the very high cross sections  $\sigma \propto 10^4 - 10^6$ , and that lower cross sections are more spread across the gluino mass spectrum.

The weak scale MSSM model used in this project, MSSM-24, requires a soft breaking scale  $Q$ . This scale is set to  $Q = 1$  TeV. It is also worth noting that the parameter space for the cross sections is significantly reduced from that of the MSSM-24. The cross sections depend on the values  $m_{\tilde{g}}$ ,  $m_{\tilde{q}}$ ,  $\tilde{g}_s$ ,  $\hat{g}_s$  and  $s$ . Since only first and second generation squarks are considered, this contributes with 8 masses<sup>2</sup> which combined with the 4 other parameters reduces the parameter space to 12 dimensions. The parameter space is thus better sampled than it would appear from the 24 parameters in MSSM-24.

### 1.1.1 Data Quality

To ensure the data is properly distributed in parameter space data quality plots are generated. In Fig. (1.2) distributions for  $m_{\tilde{g}}$ ,  $m_{\tilde{d}_L}$  and  $m_{\tilde{u}_L}$  are shown, as these are the most relevant for this thesis. The scatter plots use 2000 points, which is the number of training points used in the benchmark settings in the next section. All parts of the parameter space seem to be covered, although the density of points is higher for small masses. This could affect predictions with few training points.

### Noise

The observations contain some noise that originates in the **Prospino 2.1** calculation. In a parameter point chosen at random the relative error has a standard deviation of  $\varepsilon = 0.002$  multiplied by the cross section. This error fluctuates little between parameter points, so it should be a good approximation for the order of

<sup>2</sup>4 quarks with a pair of lefthanded and righthanded squarks each.

magnitude of errors in all points. This information can be used for the learning of GPs. Cross sections provided by Prospino are denoted as  $Y_i$  and real cross sections as  $y_i^{true}$ . The relation between them is then

$$Y_i = y_i^{true} + \epsilon_i = y_i^{true}(1 + \epsilon_i), \quad (1.3)$$

where  $\epsilon_i \sim \mathcal{N}(0, \varepsilon)$ . The distribution of  $Y_i$  can thus be written as

$$Y_i = \mathcal{N}(y_i^{true}, \varepsilon y_i^{true}), \quad (1.4)$$

where the only random variable is  $\varepsilon$ . Changing variables to  $\log_{10}$  gives

$$X_i = \log_{10} Y_i \rightarrow Y_i = 10^{X_i} \quad (1.5)$$

$$P_{X_i}(X_i) = P_{Y_i}(Y_i) \left| \frac{\partial Y_i}{\partial X_i} \right| \quad (1.6)$$

$$= P_{Y_i}(y_i) 10^{X_i} \log 10 \quad (1.7)$$

$$= \mathcal{N}(10^{x_i^{true}}, \varepsilon 10^{x_i^{true}}) \cdot 10^{X_i} \cdot \log 10. \quad (1.8)$$

This means that the relevant distribution is in fact

$$X_i = \log_{10} Y_i = \log_{10} y_i^{true} + \log_{10}(1 + \mathcal{N}(0, \varepsilon)),$$

where the following expansion can be made

$$\log_{10}(1 + \mathcal{N}(0, \varepsilon)) \simeq \frac{\mathcal{N}(0, \varepsilon)}{\log 10} - \frac{\mathcal{N}(0, \varepsilon)^2}{\log 100} + \dots \quad (1.9)$$

Since the leading order term is clearly the dominant term, the logarithm of the cross section may be written as

$$X_i \simeq \log_{10} y_i^{true} + \frac{1}{\log 10} \mathcal{N}(0, \varepsilon) \quad (1.10)$$

Since the distribution has a standard deviation  $\varepsilon = 2 \cdot 10^{-3}$ , the Gaussian noise covariance should be

$$\left( \frac{\varepsilon}{\log 10} \right)^2 = \frac{(2 \cdot 10^{-3})^2}{(\log 10)^2} = \frac{4 \cdot 10^{-6}}{5.301} \simeq 7.544 \cdot 10^{-7}.$$

The noise term predicted by the GP should therefore be of the order of  $\mathcal{O}(10^{-7})$ .

## 1.2 Dataset Transformations

The plot in Fig. (1.2b) indicates that cross sections, especially those of the order  $\mathcal{O}(1 \text{ fb})$  and lower, are very spread as a function of  $m_{\tilde{g}}$ . This is more evident in the upper left panel of Fig. (1.3), where the cross section  $\sigma$  is plotted as a function of the gluino mass. The upper left panel shows  $\sigma$  as a function of the squark mass  $m_{\tilde{q}}$ , which is more defined but still has some spread. This section will be devoted to reducing the spread caused by the gluino mass dependency of the cross section.

## Scaling Functions

As previously mentioned, the partonic cross sections can be written in terms of scaling functions  $f$

$$\hat{\sigma}_{ij} = \frac{\alpha_s^2(Q^2)}{m^2} \left\{ f_{ij}^B(\eta, r) + 4\pi\alpha_s(Q^2) \left[ f_{ij}^{V+S}(\eta, r, r_t) + f_{ij}^H(\eta, r) + \bar{f}_{ij}(\eta, r) \log\left(\frac{Q^2}{m^2}\right) \right] \right\}, \quad (1.11)$$

where

$$\eta = \frac{s}{m^2} - 1, \quad r = \frac{m_{\tilde{g}}^2}{m_{\tilde{q}}^2}, \quad r_t = \frac{m_t^2}{m^2},$$

where  $m = (\sqrt{p_1^2} + \sqrt{p_2^2})/2$  is the average mass of the particles produced. The scaling functions are the different contributions to the cross section, as explained in Chapter 2. As the total cross section only differs from the partonic cross section in that an integral over parton distribution functions is performed, the mass dependencies in Eq. (1.11) are unchanged.

The energy near the threshold is the base for an important part of the contributions to the cross section [2]. In this region the scaling functions can be expanded in the low velocity of produced particles  $\beta$ , leading to the following expressions [2]

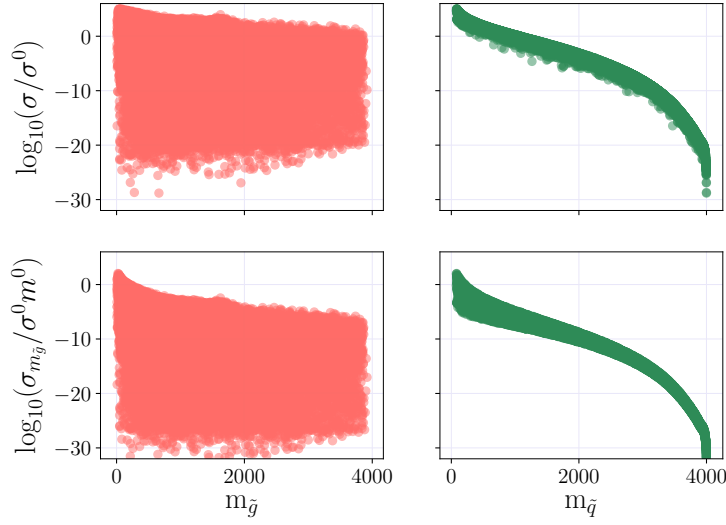
$$\begin{aligned} f_{qq}^B &= \frac{8\pi\beta m_{\tilde{q}}^2 m_{\tilde{g}}^2}{27(m_{\tilde{q}}^2 + m_{\tilde{g}}^2)^2}, & f_{q'q}^B &= \frac{8\pi\beta m_{\tilde{q}}^2 m_{\tilde{g}}^2}{9(m_{\tilde{q}}^2 + m_{\tilde{g}}^2)^2} \\ f_{qq}^{V+S} &= f_{qq}^B \frac{1}{24\beta} & f_{q'q}^{V+S} &= f_{q'q}^B \frac{1}{24\beta} \\ f_{qq}^H &= f_{qq}^B \left[ \frac{2}{3\pi^2} \log^2(8\beta^2) - \frac{7}{2\pi^2} \log(8\beta^2) \right] & f_{q'q}^H &= f_{q'q}^B \left[ \frac{2}{3\pi^2} \log^2(8\beta^2) - \frac{19}{6\pi^2} \log(8\beta^2) \right] \\ \bar{f}_{qq} &= -f_{qq}^B \frac{2}{3\pi^2} \log(8\beta^2) & \bar{f}_{q'q} &= -f_{q'q}^B \frac{2}{3\pi^2} \log(8\beta^2). \end{aligned} \quad (1.12)$$

Seeing as the main part of the contributions originate from this energy region, it may be possible to remove some of the complexity of the function using the expressions in Eq. 1.12. Note that all terms are proportional to  $f_{qq}^B$  ( $f_{q'q}^B$ ), which is again proportional to  $m_{\tilde{q}}^2 m_{\tilde{g}}^2$ . Since the partonic cross section is proportional to  $\sigma \propto 1/m^2$ , and  $m^2 = m_{\tilde{q}}^2$  in the case of squark production, the proportional  $m_{\tilde{q}}^2$ -dependency is automatically cancelled. However, the following transformation can be made

$$\sigma \rightarrow \sigma_{m_{\tilde{g}}} = \frac{\sigma}{m_{\tilde{g}}^2}, \quad (1.13)$$

reducing the gluino mass dependency. The lower panels in Fig. (1.3) show  $\sigma_{m_{\tilde{g}}}$  as a function of  $m_{\tilde{g}}$  and  $m_{\tilde{q}}$ . The spread as a function of the squark mass is much





**Figure 1.3:** The quantities  $\sigma$ ,  $\sigma_{m_{\tilde{g}}}$  and  $\sigma_{\tilde{q}}$  as functions of the gluino mass  $m_{\tilde{q}}$  and squark mass  $m_{\tilde{q}}$  for the production of  $\tilde{d}_L \tilde{d}_L$ . Here,  $m_{\tilde{q}} = m_{\tilde{d}_L}$ . The functions have less spread when some of the mass dependency is removed, which may make learning easier.

smaller, and for high cross sections the shape of  $\sigma_{m_{\tilde{g}}}$  as a function of squark and gluino mass are very similar, which may make it easier to find a kernel that fits well in both dimensions. Therefore, the target value in this thesis will be  $\sigma_{m_{\tilde{g}}}$ . In the threshold region, the partonic version of this expression is

$$\hat{\sigma}_{ij,m_{\tilde{g}}} = \frac{8\pi\beta\alpha_s^2(Q^2)}{27(m_q^2 + m_g^2)^2} \left\{ 1 + 4\pi\alpha_s(Q^2) \left[ \frac{1}{24\beta} + \frac{2}{3\pi^2} \log^2(8\beta^2) \right. \right. \quad (1.14)$$

$$\left. \left. - \frac{7}{2\pi^2} \log(8\beta^2) - \frac{2}{3\pi^2} \log(8\beta^2) \log\left(\frac{Q^2}{m^2}\right) \right] \right\}. \quad (1.15)$$

Another possibility is to further reduce the mass dependency, by defining  $\sigma_{fac}$  as

$$\sigma_{fac} = \sigma \frac{(m_g^2 + m_q^2)^2}{m_g^2}. \quad (1.16)$$

This transformation provides an even smoother function, but has been left as further work with GPs.

## 1.3 Learning a GP

In this section a single GP is learned with benchmark settings, and then a selected set of improvements are introduced. The results are quantified using plots of the

relative deviance defined in Chapter 4 (REFERER TIL LIKNINGEN).

### 1.3.1 The Benchmark

In this thesis the benchmark processes will be the production of  $\tilde{d}_L\tilde{d}_L$  and  $\tilde{d}_L\tilde{u}_L$ . The benchmark settings are a single GP with 2000 training points and 20 000 test points that uses  $m_{\tilde{g}}$  and  $m_{\tilde{d}_L}$  as features (as well as  $m_{\tilde{u}_L}$  for  $\tilde{d}_L\tilde{u}_L$  production). The exponential squared kernel (RBF <sup>3</sup>) with a white noise term. The kernel is implemented in `scikit-learn` as

```
kernel_BM = C(constant_value=10,
               constant_value_bounds=(1e-3, 1e4)) * RBF(length_scale =
               np.array([1000, 1000]), length_scale_bounds=(1, 1e6)) +
               WhiteKernel(noise_level=1, noise_level_bounds=(2e-10, 1e2))
```

where the features are (`m_gluino`, `m_dL`) for  $\tilde{d}_L\tilde{d}_L$  and (`m_gluino`, `m_dL`, `m_uL`)  $\tilde{d}_L\tilde{u}_L$ . Note that the length scale of the RBF is given a a vector of the same dimension as the feature vector  $\mathbf{x}, \ell \in \mathbb{R}^D$  <sup>4</sup>. This is in case the different masses have different covariances, as it appears they do from the lower panels in Fig. (1.3).

Because of the large span of  $\sigma_{m_{\tilde{g}}}$ -values (ranging from  $10^{-32}$  to  $10^6$ ), values are divided into decades. This means that for each interval  $10^i - 10^{i+1}$  an error is calculated. The error is the mean and variance of the distribution of relative deviations in each decade

$$\mu = m\left(\frac{y - \hat{y}}{y}\right), \quad \sigma^2 = \text{var}\left(\frac{y - \hat{y}}{y}\right). \quad (1.17)$$

The means and variances are plotted for the BM settings in Fig. (1.5) for  $\tilde{d}_L\tilde{d}_L$ , labelled *BM*. The optimal kernel parameters are found in Tab. (1.2)-(1.3). The computation times on a laptop are found in Tab. (1.4).

### 1.3.2 Outliers

Calculations in `Prospino 2.1` set some NLO terms to zero because  $K = 0$ , as discussed in SETT INN HVOR DE BLE FUNNET. This leads to outliers in the dataset, as shown in Fig. (1.4). Removing the outliers makes the prediction much better for small targets, but also stabilizes the prediction for high targets, as can be seen in Fig. (1.5 a). The predicted noise levels are significantly reduced for

<sup>3</sup>The exponential squared kernel is often called the radial basis function, hence the abbreviation.

<sup>4</sup>This example is for  $\tilde{d}_L\tilde{d}_L$  production, for  $\tilde{d}_L\tilde{u}_L$  production  $D = 3$ , as  $m_{\tilde{u}_L}$  is included as well.

	$C$	$\ell_{m_{\tilde{g}}}$	$\ell_{m_{\tilde{d}_L}}$	$\ell_{\tilde{m}}$	$\alpha$
BM	54.6 <sup>2</sup>	5470	2190		0.47
No outliers	98.5 <sup>2</sup>	5740	215		0.00372
$\sigma > 10^{-16}$ fb	22.7 <sup>2</sup>	1170	998		0.0036
$\tilde{m}$	33.1 <sup>2</sup>	1190	200	846	0.0000119
Matern	21.8 <sup>2</sup>	2020	3290		0.0253

**Table 1.2:** Optimal kernel parameters for different settings for  $\tilde{d}_L \tilde{d}_L$ .

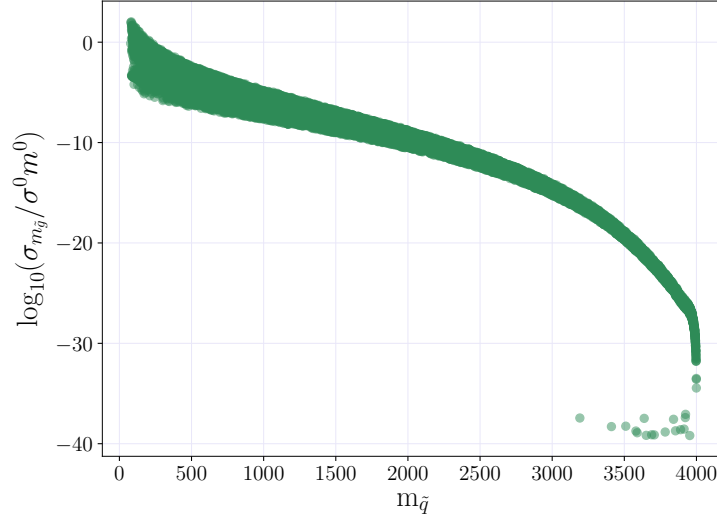
	$C$	$\ell_{m_{\tilde{g}}}$	$\ell_{m_{\tilde{d}_L}}$	$\ell_{m_{\tilde{u}_L}}$	$\ell_{\tilde{m}}$	$\alpha$
BM	49.9 <sup>2</sup>	5870	4000	2220		0.593
No outliers	80 <sup>2</sup>	4420	3100	240		0.00348
$\sigma > 10^{-16}$ fb	100 <sup>2</sup>	1540	2900	2150		0.0031
$\tilde{m}$	61.7 <sup>2</sup>	1340	3590	251	748	0.0000119
Matern	20.6 <sup>2</sup>	1930	1930	9220		0.000386

**Table 1.3:** Optimal kernel parameters for different settings for  $\tilde{d}_L \tilde{u}_L$ .

	Time $\tilde{d}_L \tilde{d}_L$	Time $\tilde{d}_L \tilde{u}_L$
BM	00:07:48	00:08:40
Outliers	00:08:42	00:11:20
Cut	00:07:24	00:11:07
Features	00:11:20	00:16:37
Kernel	00:13:33	00:18:19

**Table 1.4:** Computation times for GP with 2000 training points and 20 000 test points on a laptop with 4 cores.

both processes, from  $\alpha = 0.47$  to  $\alpha = 0.00372$  for  $\tilde{d}_L \tilde{d}_L$  and from  $\alpha = 0.593$  to  $\alpha = 0.00348$  for  $\tilde{d}_L \tilde{u}_L$ . This implies that including the outliers leads the GP to underfit, which means that much of the signal is considered noise, worsening the prediction.



**Figure 1.4:** Plot of  $\log(\sigma_{m_{\tilde{g}}})$  for  $\tilde{d}_L \tilde{d}_L$  as a function of  $m_{\tilde{d}_L}$ , where outliers are included. The outliers are originally  $\sigma = 0$  fb, but are set to  $\sigma = 10^{-32}$  fb so as to avoid infinities.

### 1.3.3 Cuts on Cross Sections

Smoother functions are easier to fit with GP, and as can be seen from the right-hand panels in Fig. (1.3), the targets are very speed functions of the squark masses for small values. Small target values also comprise the space with the most spread as a function of the gluino mass. Since the limit for 0.02 events is at  $\sigma = 10^{-3}$  fb, and interesting values are larger than this, a lower cut is set at  $\sigma = 10^{-16}$  fb, excluding all lower values from both training and testing. The resulting errors are shown in Fig. (). The optimal kernel parameters for  $\tilde{d}_L \tilde{d}_L$  with a lower cut of  $\sigma > 10^{-16}$  fb are

```
22.7**2 * RBF(length_scale=[1.17e+03, 998]) +
  WhiteKernel(noise_level=0.00336)
```

and for  $\tilde{d}_L \tilde{u}_L$

```
Kernel parameters: 100**2 * RBF(length_scale=[1.54e+03,
  2.9e+03, 2.15e+03]) + WhiteKernel(noise_level=0.0031)
```

Noise levels are further reduced from the case where outliers are removed, with the noise going from  $\alpha = 0.00372$  to  $\alpha = 0.00336$  for  $\tilde{d}_L \tilde{d}_L$  and from  $\alpha = 0.00348$  to  $\alpha = 0.0031$  for  $\tilde{d}_L \tilde{u}_L$ . As will be shown in a later section, there is reason to believe the noise should be rather low ( $\alpha \propto 10^{-7}$ ), so this indicates that excluding low cross sections improves the prediction. The error in the prediction of the highest targets is also significantly improved from the case of the BM and the case with removed outliers, especially for  $\tilde{d}_L \tilde{u}_L$ .

### 1.3.4 Features

Prospino 2.1 calculates the NLO cross section for the mean of the squark masses,  $\bar{m}$ , and uses this to find the  $K$ -factor. The  $K$ -factor is then multiplied with the cross sections with non-degenerate squark masses to find the individual NLO terms. There is therefore reason to believe that adding the mean squark mass as a feature will improve the prediction. The features tested in this section are then

```
features_dLdL = (m_gluino, m_dL, m_mean)
features_dLuL = (m_gluino, m_dL, m_uL, m_mean)
```

The optimal kernel parameters for  $\tilde{d}_L \tilde{d}_L$  are

```
33.1**2 * RBF(length_scale=[1.19e+03, 200, 846]) +
WhiteKernel(noise_level=1.19e-05)
```

and for  $\tilde{d}_L \tilde{u}_L$

```
61.7**2 * RBF(length_scale=[1.34e+03, 3.59e+04, 251, 748]) +
WhiteKernel(noise_level=1.19e-05)
```

Adding the mean mass as a feature significantly reduces the estimated noise level, even giving the same level for  $\tilde{d}_L \tilde{d}_L$  and  $\tilde{d}_L \tilde{u}_L$ . This could be a good sign of the prediction, as the two targets should have the same level. The length scale for  $\bar{m}$  is smaller than for  $m_{\tilde{g}}$ , indicating that there is a higher dependence on the mean mass than the gluino mass. This can be a good sign for the addition of the new feature.

### Lagrangian Masses

Since the BDTs used in [4] showed, using importance sampling, that the Lagrangian parameters are not good features for the target value in question, these will not be tested here.

Training points	2000 p/ expert	GP
2000	00:03:32	00:03:32
4000	00:04:39	00:16:33
6000	00:04:10	
8000	00:05:29	
10 000	00:05:46	
12 000	00:06:31	
14 000	00:05:46	

## Mean Mass

### 1.3.5 Kernel

Compare Matern and RBF

The optimal kernel parameters with the Matérn kernel for  $\tilde{d}_L \tilde{d}_L$  are

```
21.8**2 * Matern(length_scale=[2.02e+03, 3.29e+03], nu=1) +
WhiteKernel(noise_level=0.0253)
```

and for  $\tilde{d}_L \tilde{u}_L$

```
20.6**2 * Matern(length_scale=[1.93e+03, 1.93e+03, 9.22e+03],
nu=1) + WhiteKernel(noise_level=0.000386)
```

## Hyperparameters

Plot of different  $\nu$ -values?

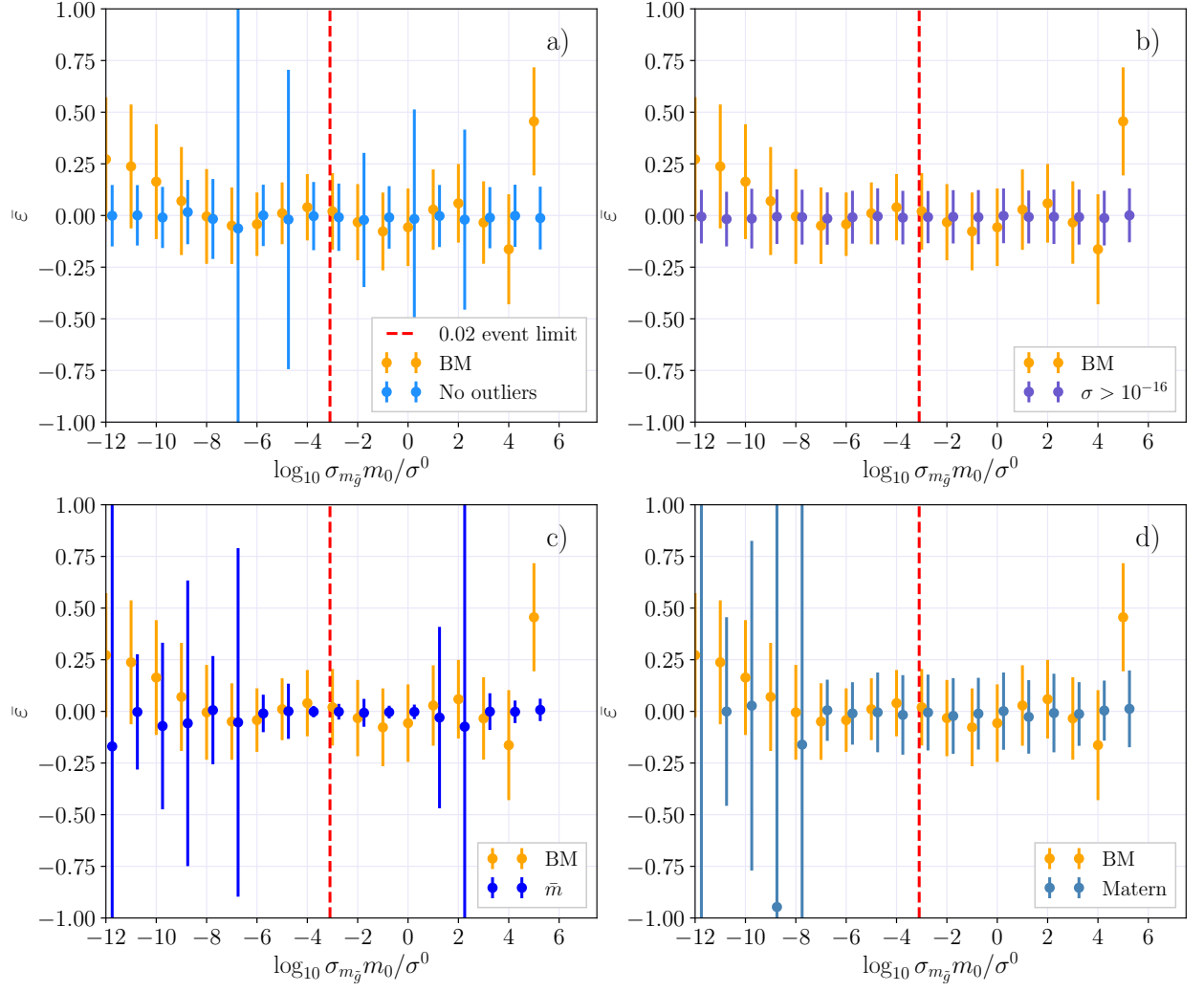
## 1.4 Distributed Gaussian Processes

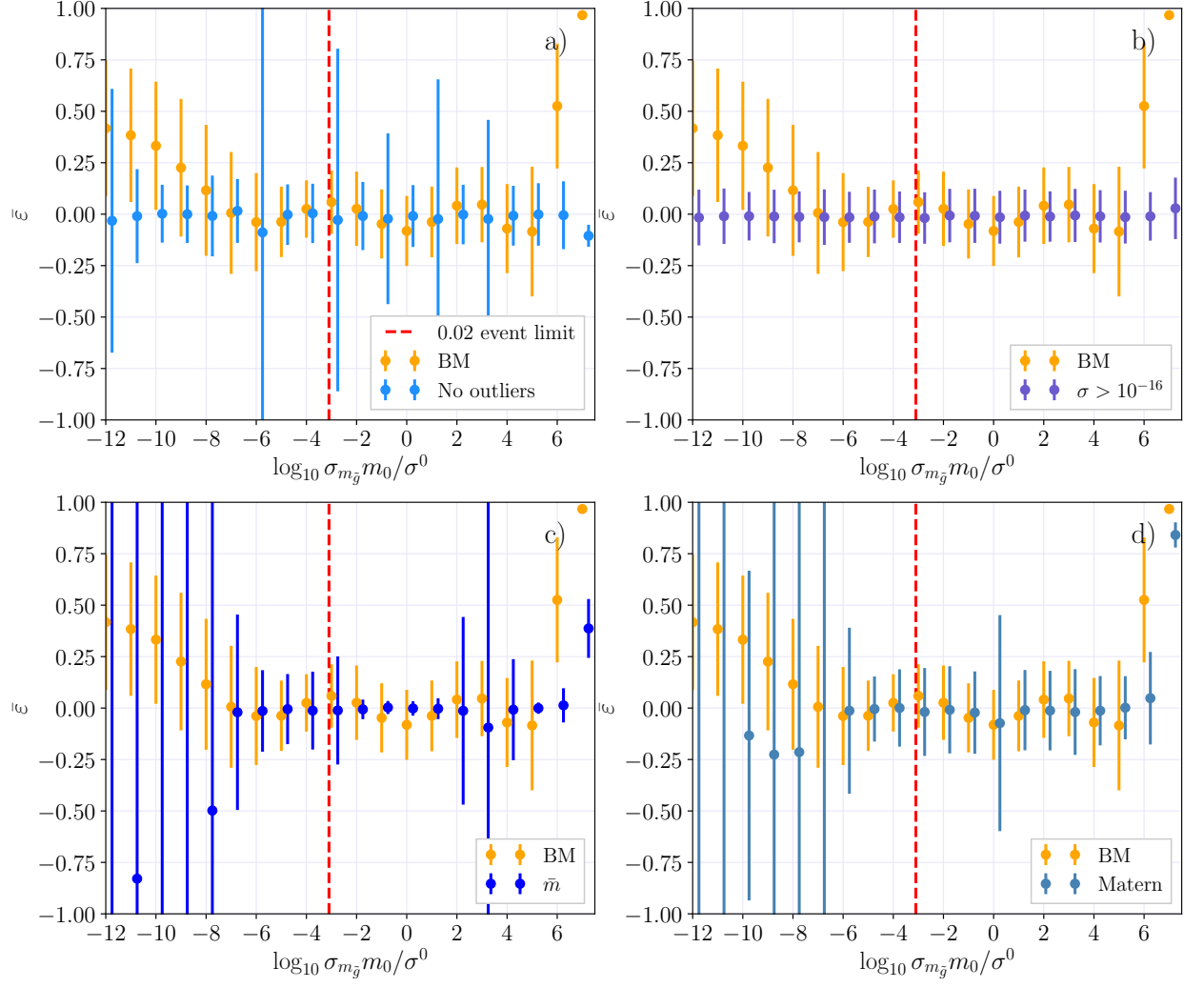
Time plots.

Matrix with number of experts and training points per expert, with mean relative deviance.

### 1.4.1 Adding Experts

**Figure 1.7:** Learning curves for an increasing number of experts, where each expert has 500, 1000 training points, for the process  $\tilde{d}_L \tilde{d}_L$ . A value of 1 is a perfect prediction, and 0 is a bad prediction.

**Figure 1.5:** Errors

**Figure 1.6:** Errors



### 1.4.2 Cross validation for DGP

There is no `scikit-learn` function for DGP, so we've implemented a method for calculating the learning curve of a DGP when experts are added. The program uses  $k$ -fold cross validation with loss function  $R^2$ .



# Bibliography

- [1] B.C. Allanach. Softsusy: A program for calculating supersymmetric spectra. *Computer Physics Communications*, 143(3):305 – 331, 2002.
- [2] Wim Beenakker, R Höpker, Michael Spira, and PM Zerwas. Squark and gluino production at hadron colliders. *Nuclear Physics B*, 492(1-2):51–103, 1997.
- [3] Anders Kvellestad. Chasing susy through parameter space. 2015.
- [4] Jon Vegard Sparre. Fast evaluation of supersymmetric cross sections. 2015.