Improving Online GMM Learning Via Covariance Weighting

Stephan Rabanser
Max Planck Institute for Astrophysics
rabanser@mpa-garching.mpg.de

Maksim Greiner
Max Planck Institute for Astrophysics
maksim@mpa-garching.mpg.de

1 INTRODUCTION

Gaussian mixture models (GMMs) are used in a wide variety of application areas, such as data mining, pattern recognition, machine learning, and statistical analysis.

Traditionally, we assume that all the data points we are using in order to learning GMMs are present and in memory at the time of learning. However, in some cases, one might want to learn a GMM incrementally as the data arrives, which is commonly referred to as online learning.

2 APPORACH

The general idea we want to follow for learning online GMMs efficiently is outlined as follows:

2.1 GMM basics

Assume that we have learned an incremental GMM from the incoming data points until time t. Then the corresponding probability distribution is given as

$$P^t(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} w_i^t \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^t, \Sigma_i^t)}{\sum_{i=1}^{k} w_i^t}. \tag{1}$$

In this setting, k denotes the number of mixture components, w_i^t denotes the mixture weight for component i at time t, and $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^t, \Sigma_i^t)$ denotes the i-th mixture component at time t with mean $\boldsymbol{\mu}_i^t$ and covariance Σ_i^t.

2.2 Model growing

Now the question is raised how to handle a new incoming data point arriving at $t+1$. Here we propose to trivially integrate the new data point into the existing model structure by adding this point as a new Gaussian component $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}^{t+1}, \Sigma^{t+1})$ weighted with w^{t+1}. The full probability distribution at time $t+1$ is given as

$$P^{t+1}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} [w_i^t \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^t, \Sigma_i^t)] + w^{t+1}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}^{t+1}, \Sigma^{t+1})}{\sum_{i=1}^{k} [w_i^t] + w^{t+1}}. \tag{2}$$

This new Gaussian component will be centered on the data point itself and will receive a weighted prior covariance. Hence, each Gaussian is governed by a slightly different covariance matrix as opposed to [?]. The covariance matrix of the i-th component is expressed as

$$\Sigma_i = \frac{w_i \tilde{\Sigma}_i + w_p \Sigma_p}{w_i + w_p}, \tag{3}$$

where $\tilde{\Sigma}_i$ corresponds to the sample covariance, w_p corresponds to the prior weight given to the Gaussian component, and Σ_p corresponds to the prior covariance given to the Gaussian component. Note that we will assume that Σ_p is a scaled identity matrix, hence $\Sigma_p = \sigma_p \boldsymbol{I}$ with σ_p being the prior variance. For a newly added data point at $t+1$, we set $w_i = 1$ whereas $\tilde{\Sigma}_i$ is still unknown. The resulting Gaussian component is therefore governed by $\boldsymbol{\mu}^{t+1} = \boldsymbol{x}$ and $\Sigma^{t+1} = \frac{w_p \Sigma_p}{1+w_p}$ and weighted by $w^{t+1} = 1$. Hence, the updated model is given as follows:

$$P^{t+1}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} [w_i^t \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^t, \Sigma_i^t)] + \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}, \frac{w_p \Sigma_p}{1+w_p})}{\sum_{i=1}^{k} [w_i^t] + 1}. \tag{4}$$

A common choice for the prior weight would be $w_p = 1$, which means that the prior is treated with the same weight as a single data point. Σ_p is considered a hyper-parameter one has to tune.

Equation (3) corresponds to the maximum a posteriori (MAP) solution for the covariance of a Gaussian distribution with an inverse Wishart distribution as the prior distribution. The inverse Wishart distribution of a $q \times q$ matrix Σ is defined as

$$\mathcal{W}^{-1}(\Sigma|\nu, \boldsymbol{\Psi}) \equiv \frac{|\boldsymbol{\Psi}|^{\frac{\nu}{2}}}{2^{\frac{\nu p}{2}} \Gamma_q(\frac{\nu}{2})} |\Sigma|^{-\frac{\nu+p+1}{2}} \, \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}\Sigma^{-1})}, \tag{5}$$

where Γ_q is the multivariate gamma function. The posterior distribution for the covariance Σ_i given a set of data points $\{\boldsymbol{d}\}$ now reads

$$P(\Sigma_i|\{\boldsymbol{d}\}) \propto \mathcal{W}^{-1}(\Sigma_i|\nu, \boldsymbol{\Psi}) \prod_t \mathcal{N}(\boldsymbol{d}_t|\boldsymbol{\mu}_i, \Sigma_i). \tag{6}$$

With $\boldsymbol{\mu}_i = \frac{1}{N}\sum_t \boldsymbol{d}_t$ and N being the number of datapoints the argmax of the posterior distribution yields

$$\operatorname{argmax} P(\Sigma_i|\{\boldsymbol{d}\}) = \frac{\boldsymbol{\Psi} + \sum_t (\boldsymbol{d}_t - \boldsymbol{\mu}_i)(\boldsymbol{d}_t - \boldsymbol{\mu}_i)^\dagger}{\nu + p + 1 + N}, \tag{7}$$

where † denotes complex conjugation and transposition. This result can be easily related to (3) by identifying N with w_i, $\nu + p + 1$ with w_p, and $\boldsymbol{\Psi}$ with $w_p \Sigma_p$.

2.3 Model reduction

Since we are not interested in a model in which all data points are represented by a separate Gaussian distribution, we want to simplify the model as soon as possible in order to reduce the model complexity and to capture the (latent) structure of the data. Therefore, it is advised to perform simplification checks after each newly observed data point. During these tests, each Gaussian component is compared with every other Gaussian component to determine the level of similarity between the respective Gaussians. If a given similarity threshold is exceeded (minimal overlap $o_{\min}$), the two Gaussians are merged into a new Gaussian component.

2.3.1 Similarity check. In contrast to prior works, which proposes a similarity measure based on the Kolmogorov-Smirnoff test, we propose two distance measures based on the scalar product of two Gaussians $\mathcal{N}_i = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i)$ and $\mathcal{N}_j = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j)$, which we define as

$$\langle \mathcal{N}_i, \mathcal{N}_j \rangle = \int \mathcal{N}_i \mathcal{N}_j d\boldsymbol{x}. \tag{8}$$

The first distance measure is given as

$$s_1(\mathcal{N}_i, \mathcal{N}_j) = \ln\left(\frac{\langle \mathcal{N}_i, \mathcal{N}_j \rangle}{\sqrt{\langle \mathcal{N}_i, \mathcal{N}_i \rangle \langle \mathcal{N}_j, \mathcal{N}_j \rangle}}\right) \tag{9}$$

while the second approach is given as

$$s_2(\mathcal{N}_i, \mathcal{N}_j) = \ln\left(\langle \sqrt{\mathcal{N}_i}, \sqrt{\mathcal{N}_j} \rangle\right). \tag{10}$$

Both similarity measures (and hence also $o_{\min}$) are restricted to negative values including 0, formally $s_1, s_2, o_{\min} \in (-\infty, 0]$. Note that $o_{\min} = 0$ enforces the two Gaussians to be exactly the same, which heavily reduces the number of merges and the model's flexibility. Increasingly negative values of $o_{\min}$ allow for more relaxed scenarios, meaning that merges will occur earlier and more often. Consequently, a model with $o_{\min}$ close to 0 tends to overfit the data, while increasingly negative values will result in underfitting.

2.3.2 Merging. As already outlined before, we intend to merge the two Gaussians $\mathcal{N}_i$ and $\mathcal{N}_j$ (with w_i and w_j being the respective weighting factors in the mixture model) in case they exceed the minimal overlap $o_{\min}$. Concretely, this means removing both Gaussians from our model and adding a new single Gaussian $\mathcal{N}_m = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$ with weight w according to the following set of formulas:

$$w = w_i + w_j \tag{11}$$

$$\boldsymbol{\mu} = \frac{1}{w}(w_i\boldsymbol{\mu}_i + w_j\boldsymbol{\mu}_j) \tag{12}$$

$$\Sigma = \frac{w_i}{w + w_p}(\tilde{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T) + \tag{13}$$
$$+ \frac{w_j}{w + w_p}(\tilde{\Sigma}_j + (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T) + \frac{w_p\Sigma_p}{w + w_p}$$

These merging formulas are again similar to [**?**]. The only adaptation we need to make is to account for the correct weighting of the prior variance as part of the new covariance computation. Note that $\tilde{\Sigma}_i = \frac{\Sigma_i(w_i+w_p)-w_p\Sigma_p}{w_i}$, which follows from Equation (3). This specific weighting of the covariance of the new Gaussian is needed to also reflect the MAP solution for the covariance matrix of a multivariate Gaussian.

However, we are not done once two Gaussians are successfully merged. Since we have added a new Gaussian to our model, it is required that we re-initiate the similarity checking procedure to determine whether some new merging possibilities have emerged after the previous merge. In case no other two Gaussians are similar enough for a merge, we have arrived at a stable model for the observed data points.

```
while datapoints arriving do
    Add new Gaussian component to model centered on
     datapoint with prior covariance;
    while merge still possible do
        Check overlap between two Gaussian components;
        if significant overlap detected then
            Merge the two Gaussian components;
        end
    end
end
```

Algorithm 1: High-level algorithm outline

2.4 Model selection

While the proposed model should be suited for online learning, we are still required to tune two hyper-parameters, namely σ_p and $o_{\min}$. Model selection is therefore performed on the entire data set using a fixed set of $(\sigma_p^l, o_{\min}^l)$ for every iteration l. The performance is assessed through the Bayesian information criterion (BIC), which is given as

$$\text{BIC} = \ln(N)k - 2\ln(\hat{L}), \tag{14}$$

where k denotes the number of free parameters in our model and $\hat{L}$ the maximized likelihood function.

As part of the model selection process, we aim at minimizing the BIC. This is performed using a discretized version of gradient descent on the BIC function by adapting $(\sigma_p^l, o_{\min}^l)$ repeatedly. The two parameters are updated as follows:

$$\sigma_p^{l+1} = \sigma_p^l\sigma_p^a \qquad \text{or} \qquad \sigma_p^{l+1} = \frac{\sigma_p^l}{\sigma_p^a} \tag{15}$$

$$o_{\min}^{l+1} = o_{\min} + o_{\min}^a \qquad \text{or} \qquad o_{\min}^{l+1} = o_{\min}^l - o_{\min}^a \tag{16}$$

In each iteration we first pick one of the two parameters alternately, increase and decrease the respective parameter by an adaption factor (or summand) and evaluate the BIC for both the increase and the decrease. The value which leads to a lower BIC result is then used for the subsequent iterations. In case the BIC does no longer improve, we have found our optimal set of hyper-parameters $(\sigma_p^L, o_{\min}^L)$. Note that when optimizing the prior variance, the adaption is applied multiplicatively through σ_p^a, while when optimizing the minimal overlap, the adaption is applied additively $o_{\min}^a$. While these two parameters can be chosen more freely, we will use $\sigma_p^a = 2$ and $o_{\min}^a = 0.5$ for our experiments. Also, we choose $\sigma_p^0 = 1$ and $o_{\min}^0 = -1$ as the initial values for the optimization procedure.

3 EXPERIMENTS

3.1 Data sets

To evaluate the performance of our estimation procedure, we used four different artificially generated datasets: single Gaussian, four (clearly separated) Gaussians, the banana, and the swiss roll.

3.2 Plot explanations

The following two subsections summarize our results obtained from a few test runs by using the described data sets. Each section features a few tables whose plots we will briefly explain here.

The first two rows of each table are composed of depictions of the iterative evolution of the GMM as more and more data points are added. Note that these plots are representations of the model with the optimal set of hyper-parameters $(\sigma_p^L, o_{\min}^L)$.

The third row contains three line graphs which further show key information in the evolution of the model.

(1) The first line plot shows the discrete gradient descent optimization procedure on the BIC as described in Section 2.4.the red dot in the graph represents the optimal combination.
(2) The second line plot shows the evolution of the number of Gaussian components used in the mixture as the data points are added. Note again that this plot shows the evolution for the optimal hyper-parameter setting. Therefore, this plot gives us a more continuous interpretation of the plots shown in the first two rows of the table.
(3) The third line plot shows the evolution of the number of Gaussian components over the total number of optimization iterations. Hence, this plot gives us a feeling of how the number of Gaussian components would look like for other, worse settings of $(\sigma_p^L, o_{\min}^L)$.

3.3 First distance measure (s_1)

See Tables 1 to 16.

3.4 Second distance measure (s_2)

See Tables 17 to 32.

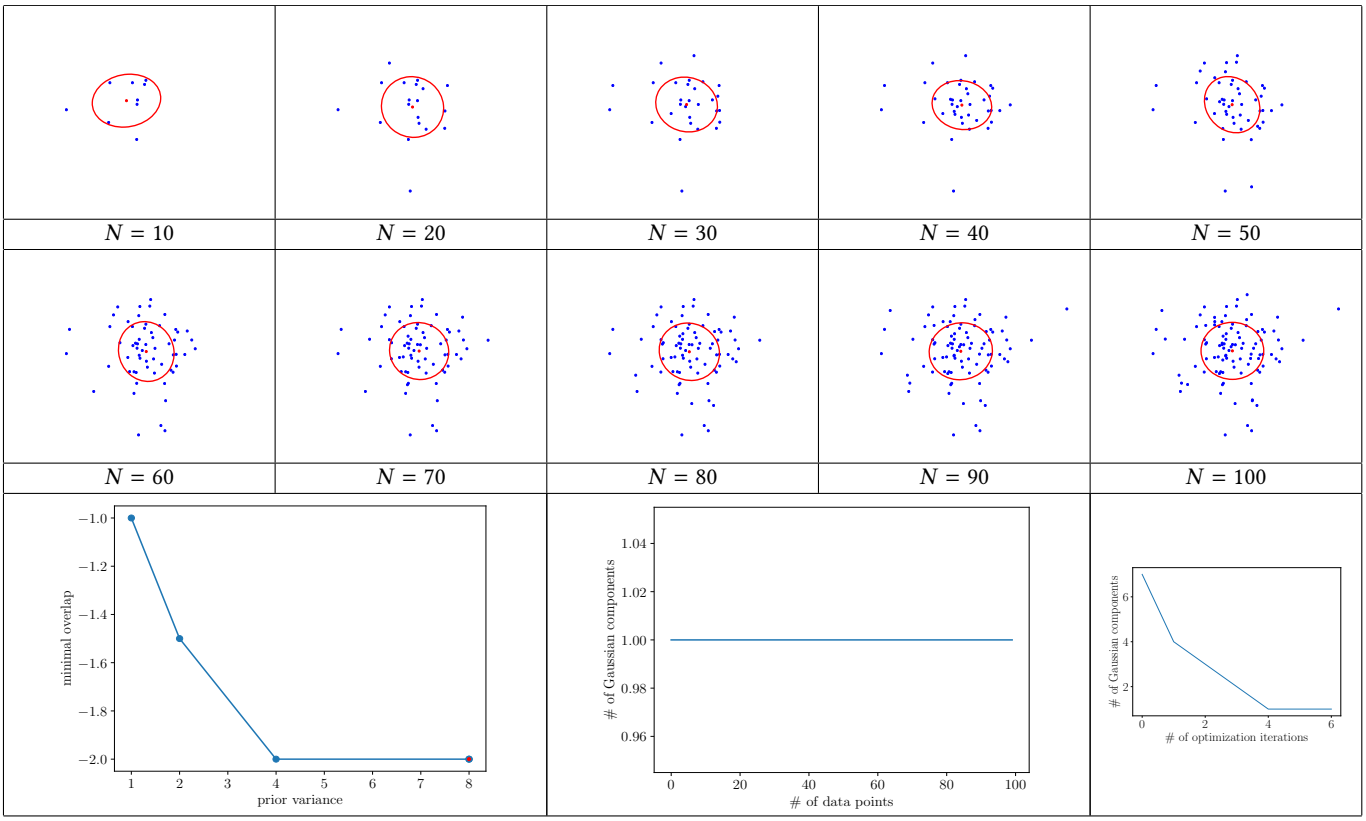


Table 1: Single Gaussian in 2 dimensions with 100 data points

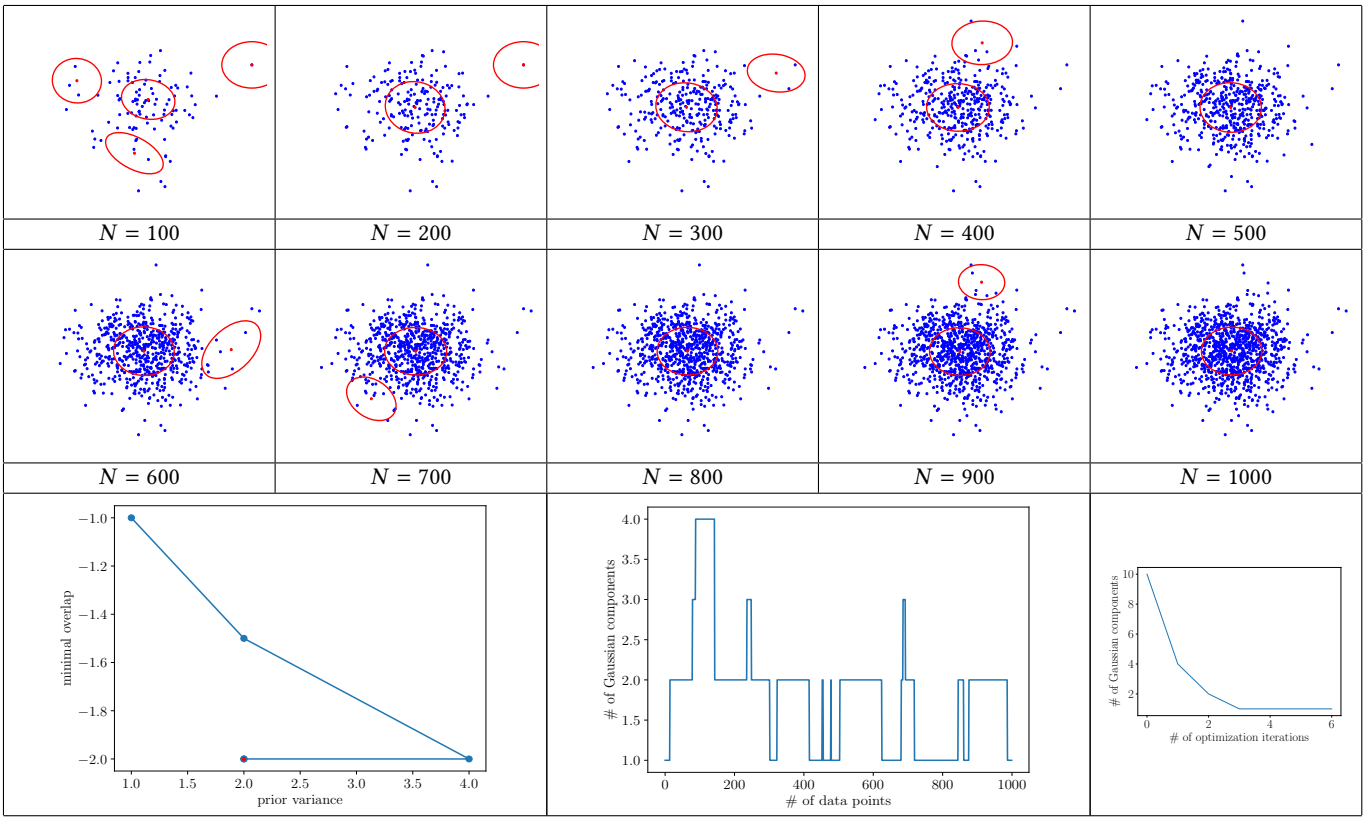


Table 2: Single Gaussian in 2 dimensions with 1000 data points

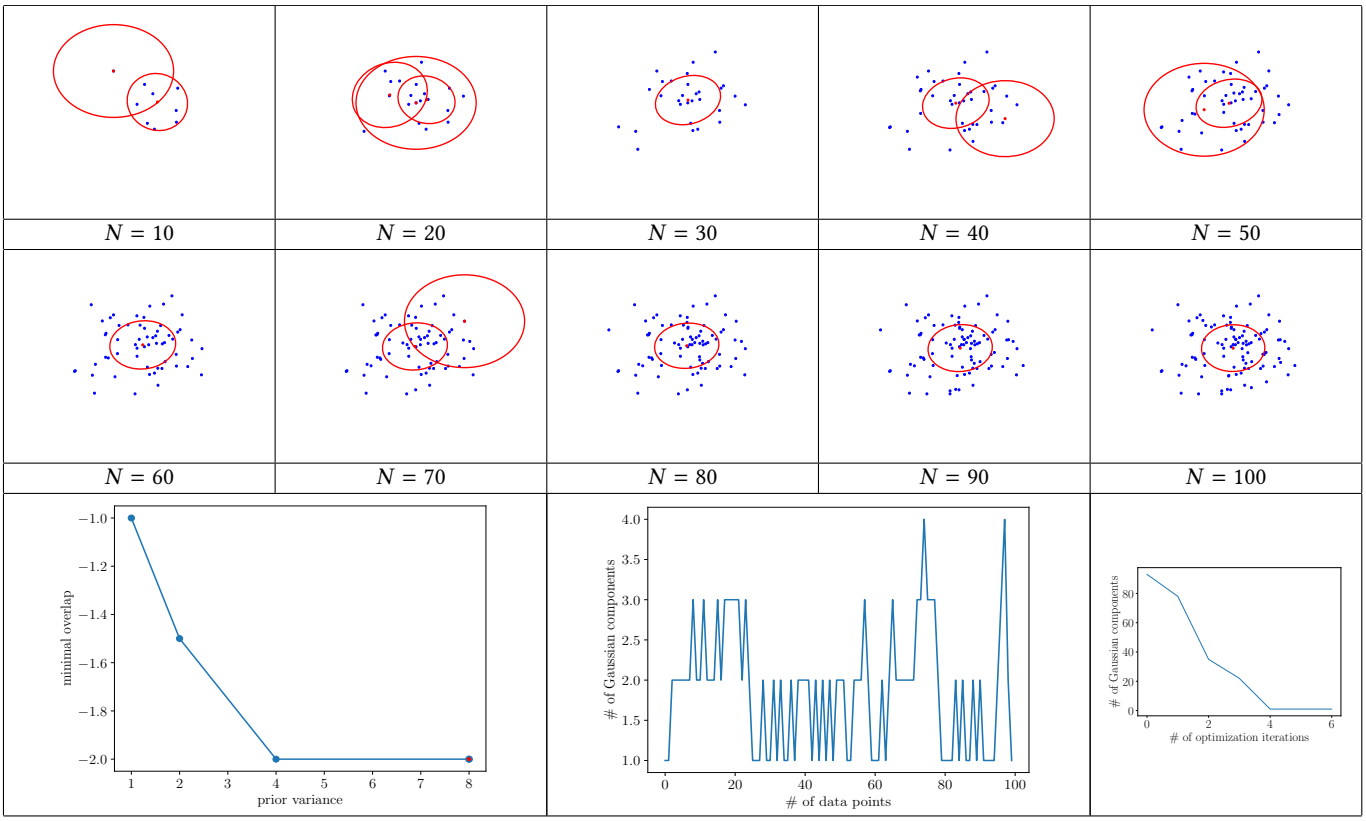


Table 3: Single Gaussian in 10 dimensions with 100 data points

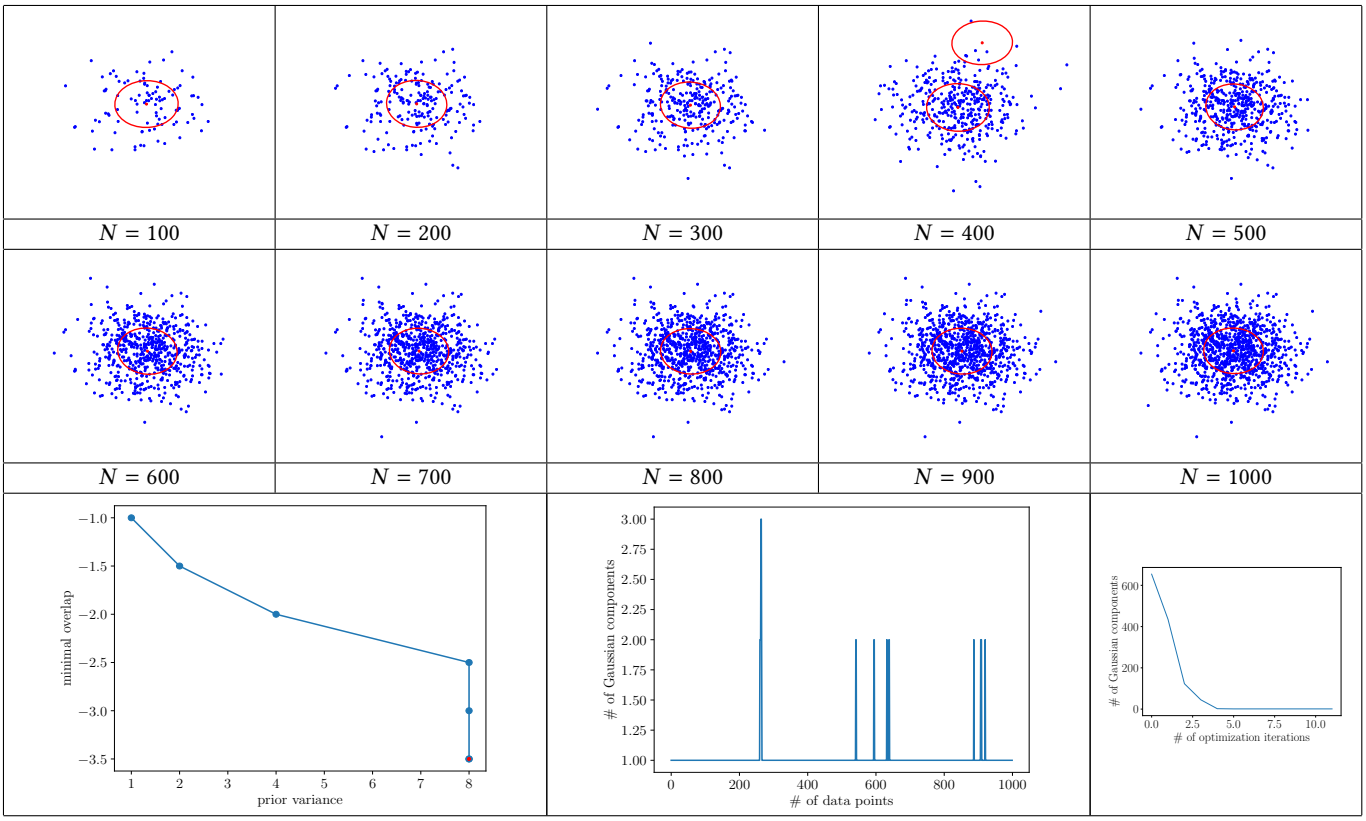


Table 4: Single Gaussian in 10 dimensions with 1000 data points

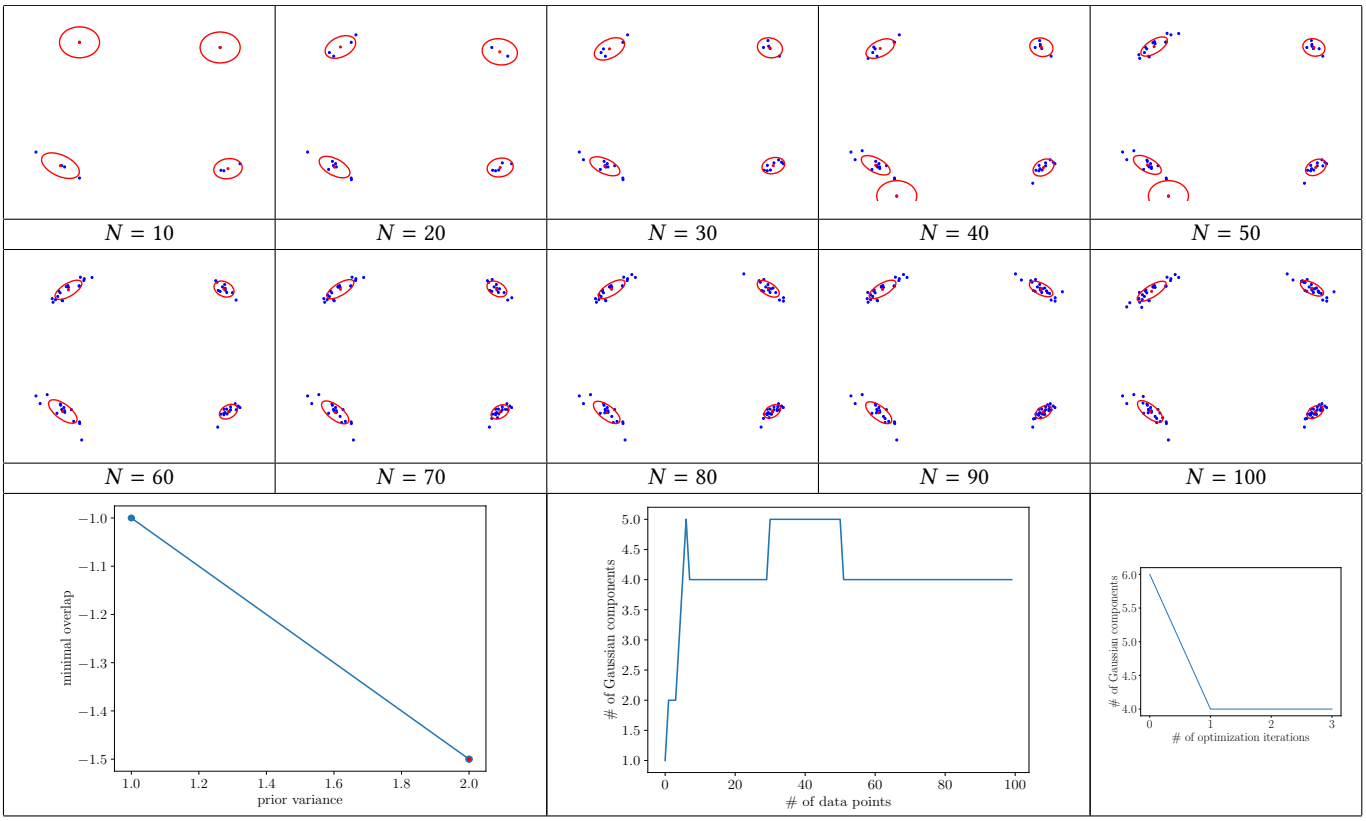


Table 5: Four Gaussians in 2 dimensions with 100 data points

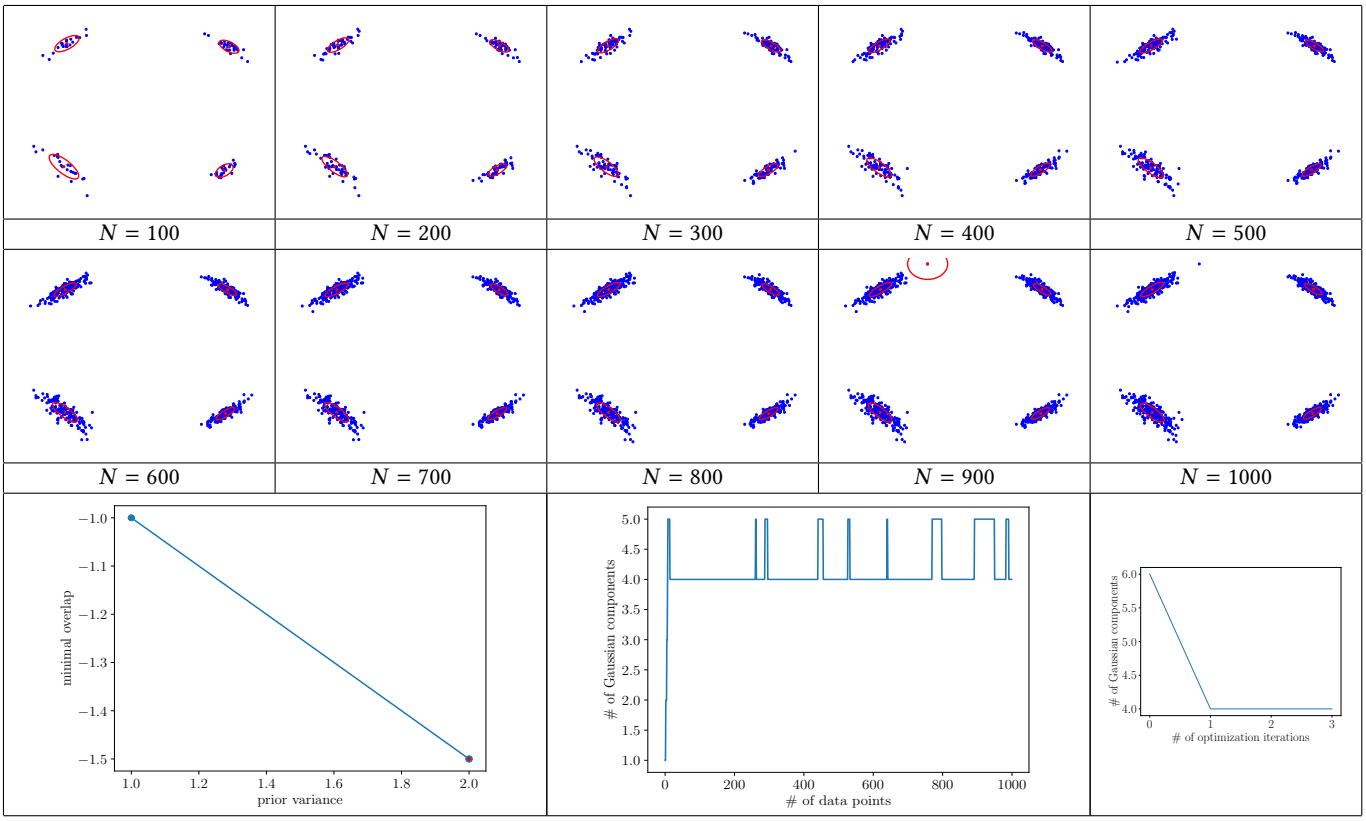


Table 6: Four Gaussians in 2 dimensions with 1000 data points

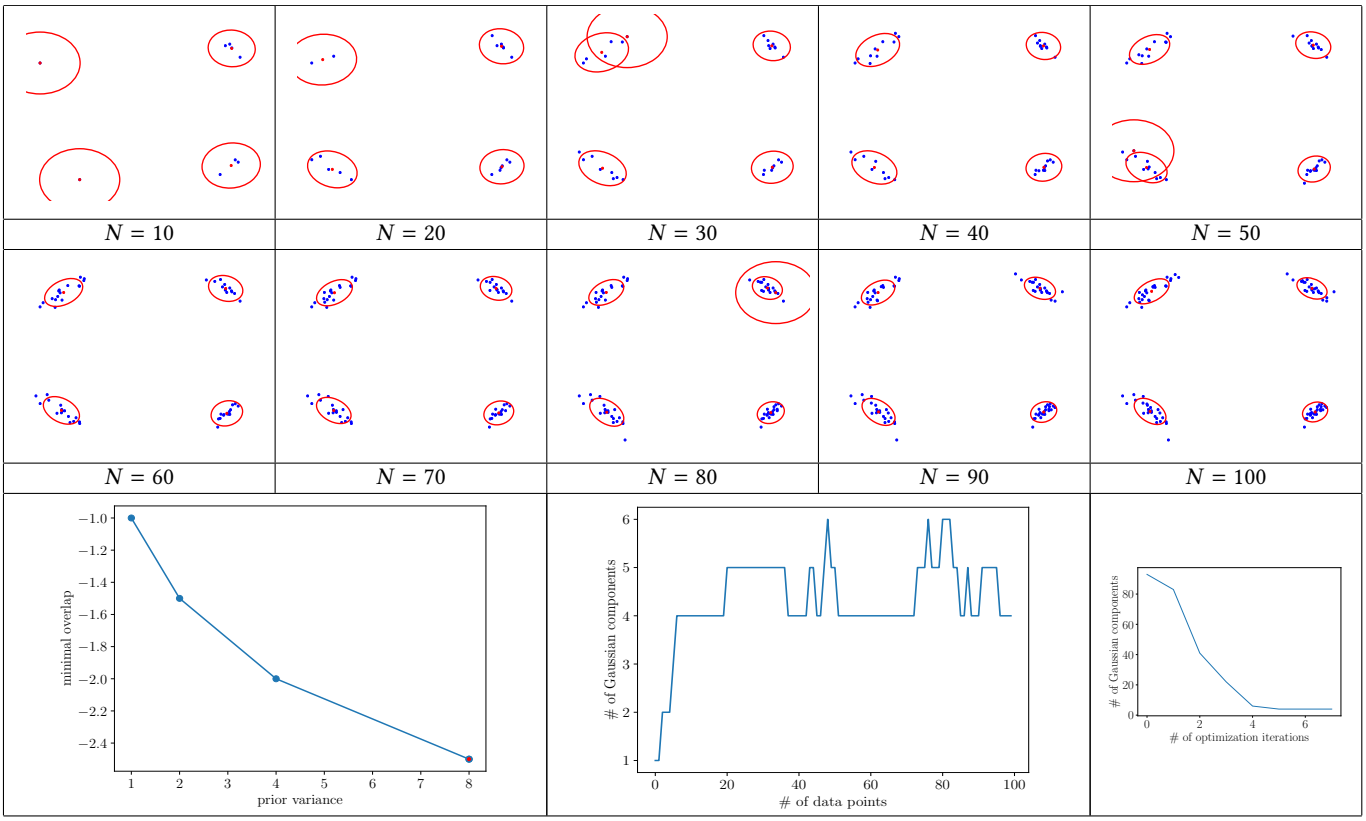


Table 7: Four Gaussian in 10 dimensions with 100 data points

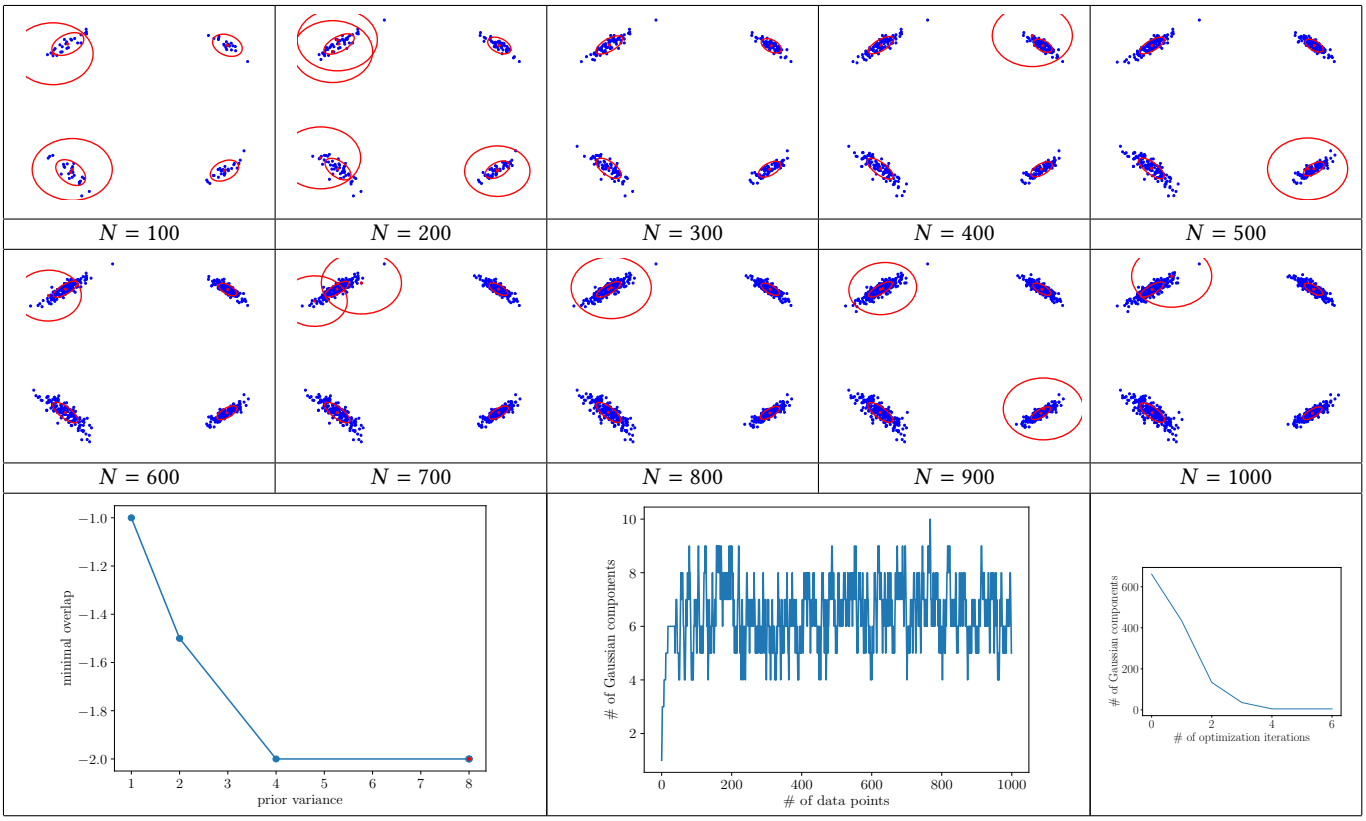


Table 8: Four Gaussians in 10 dimensions with 1000 data points

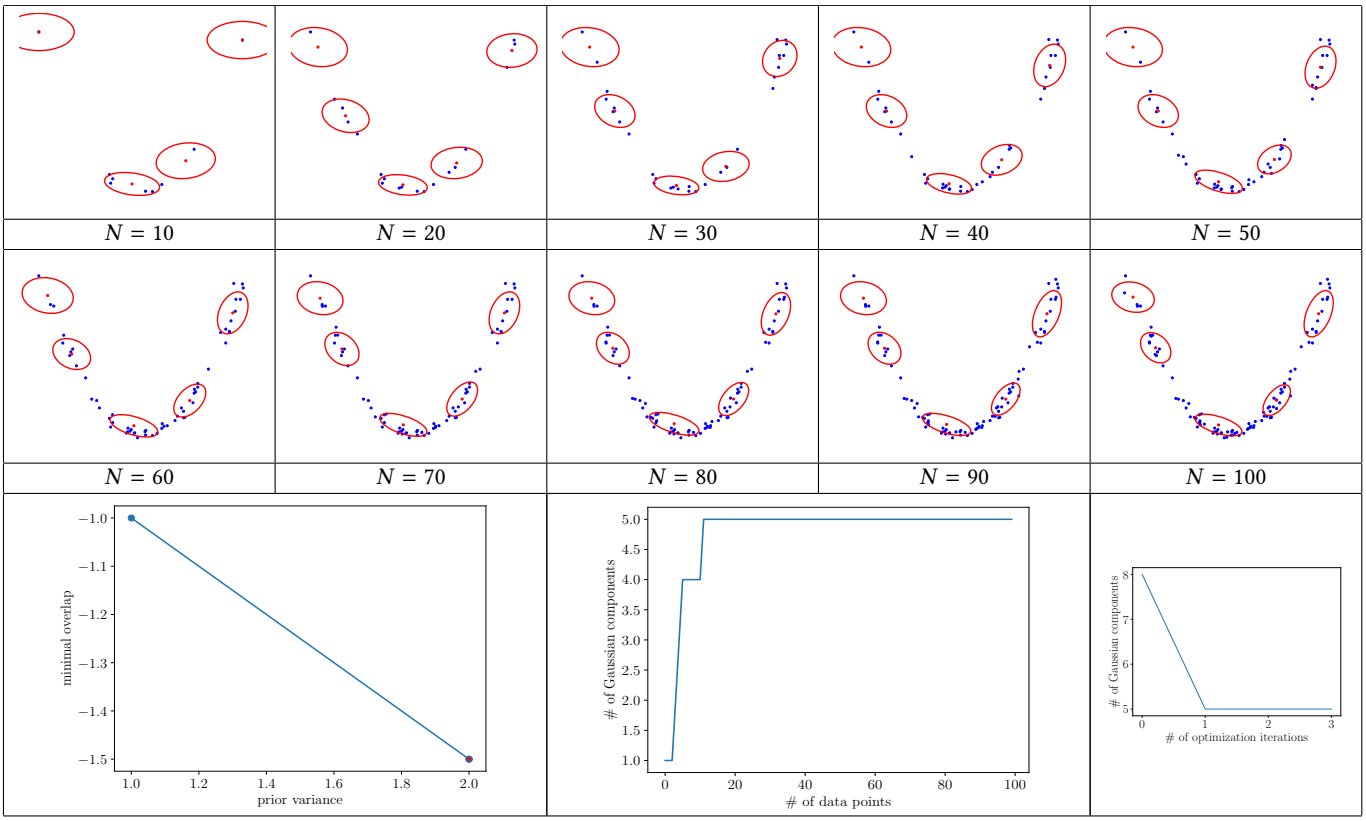


Table 9: Banana in 2 dimensions with 100 data points

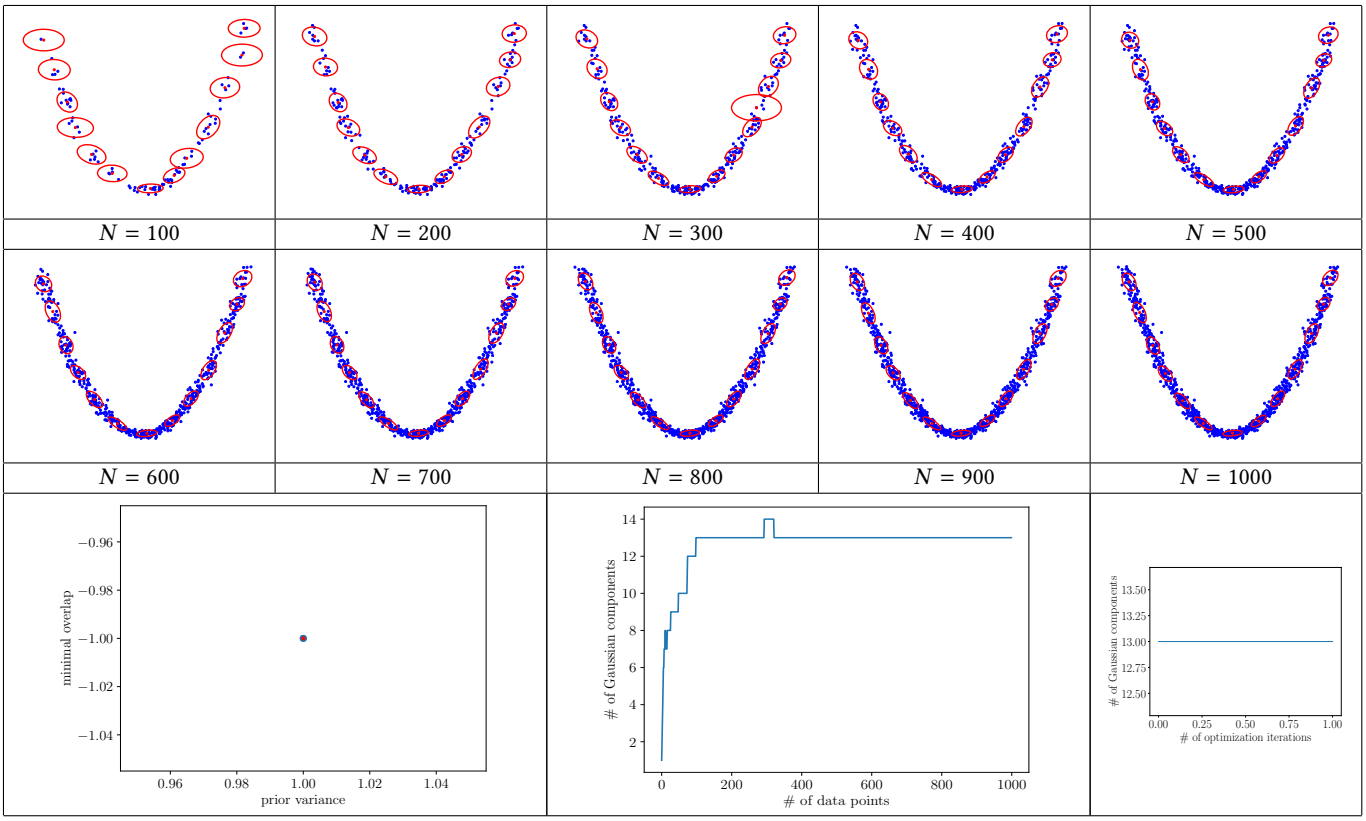


Table 10: Banana in 2 dimensions with 1000 data points

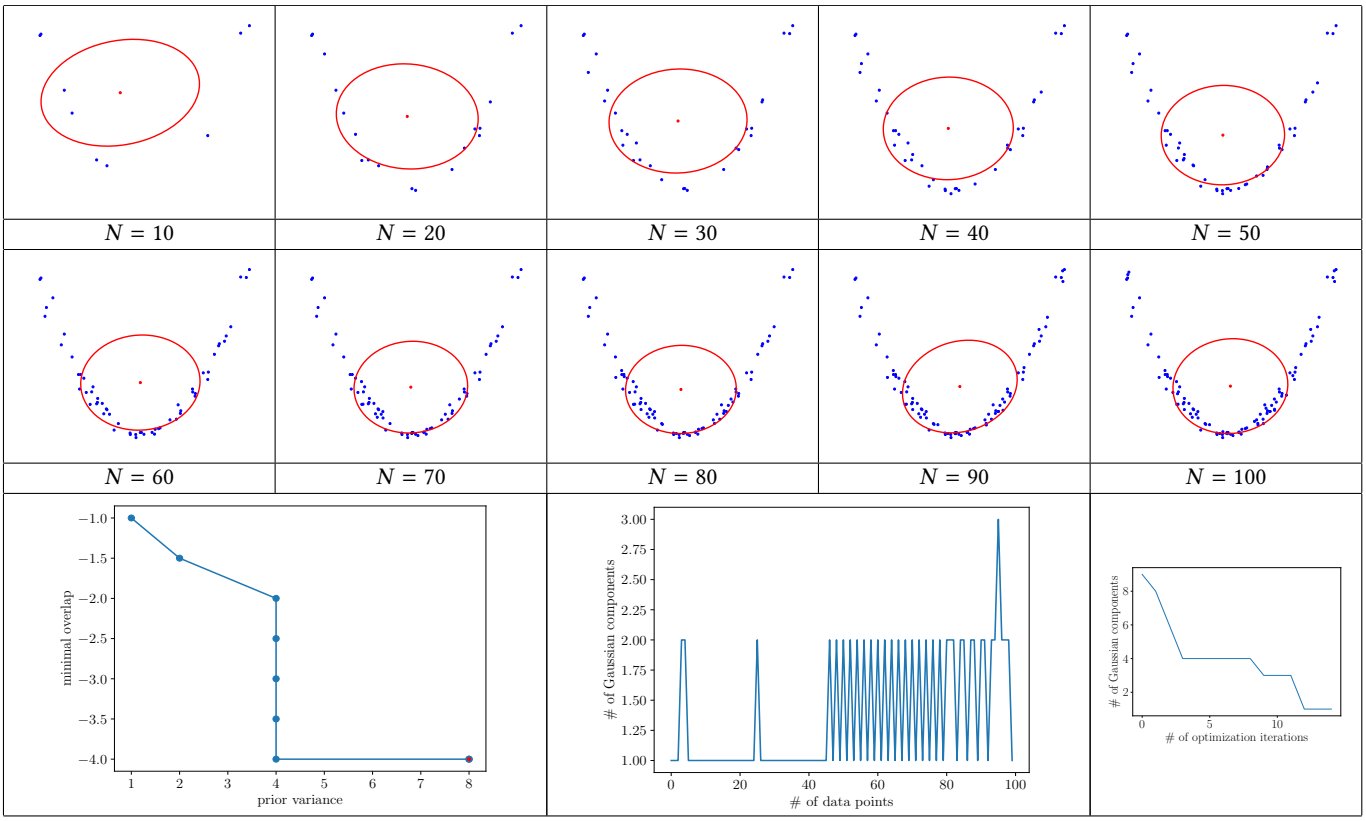


Table 11: Banana in 10 dimensions with 100 data points

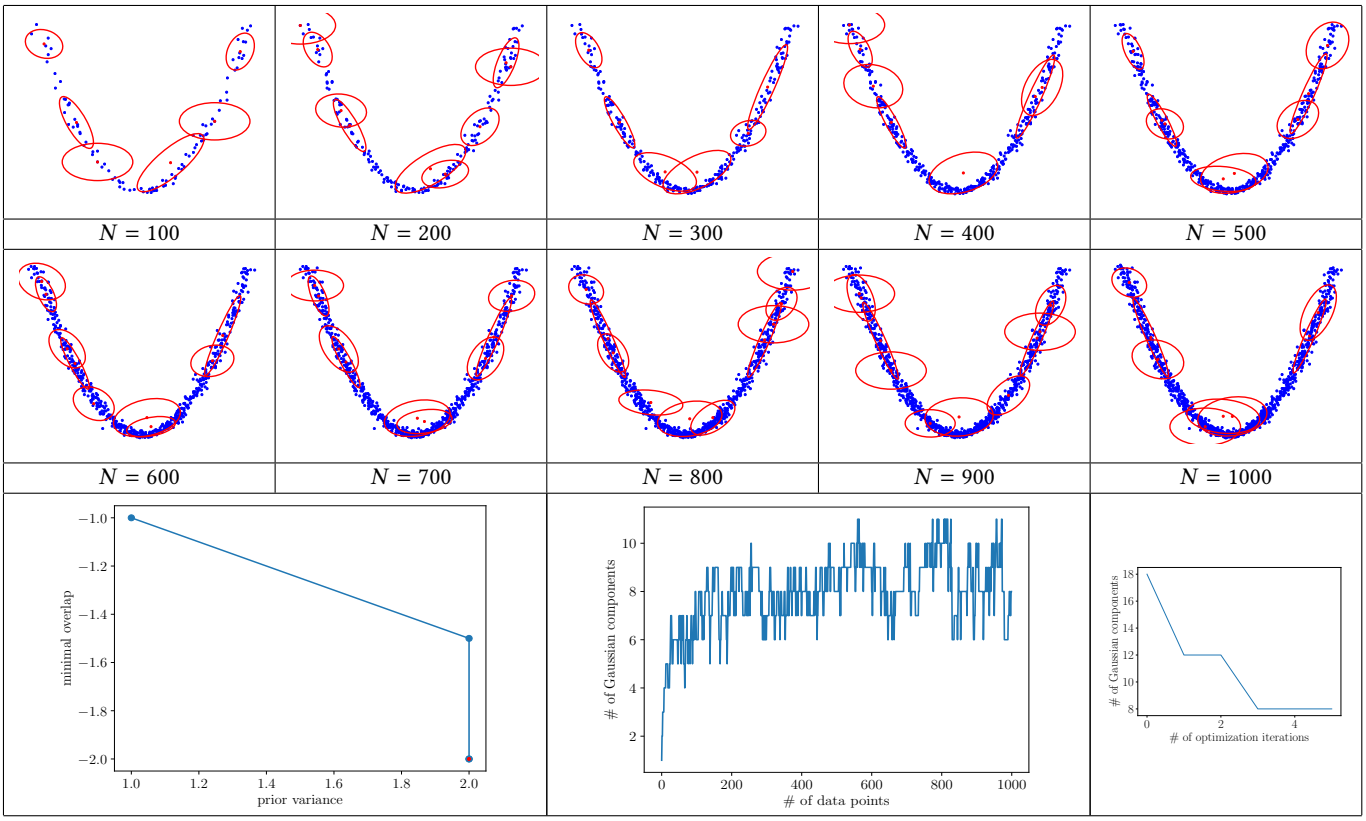


Table 12: Banana in 10 dimensions with 1000 data points

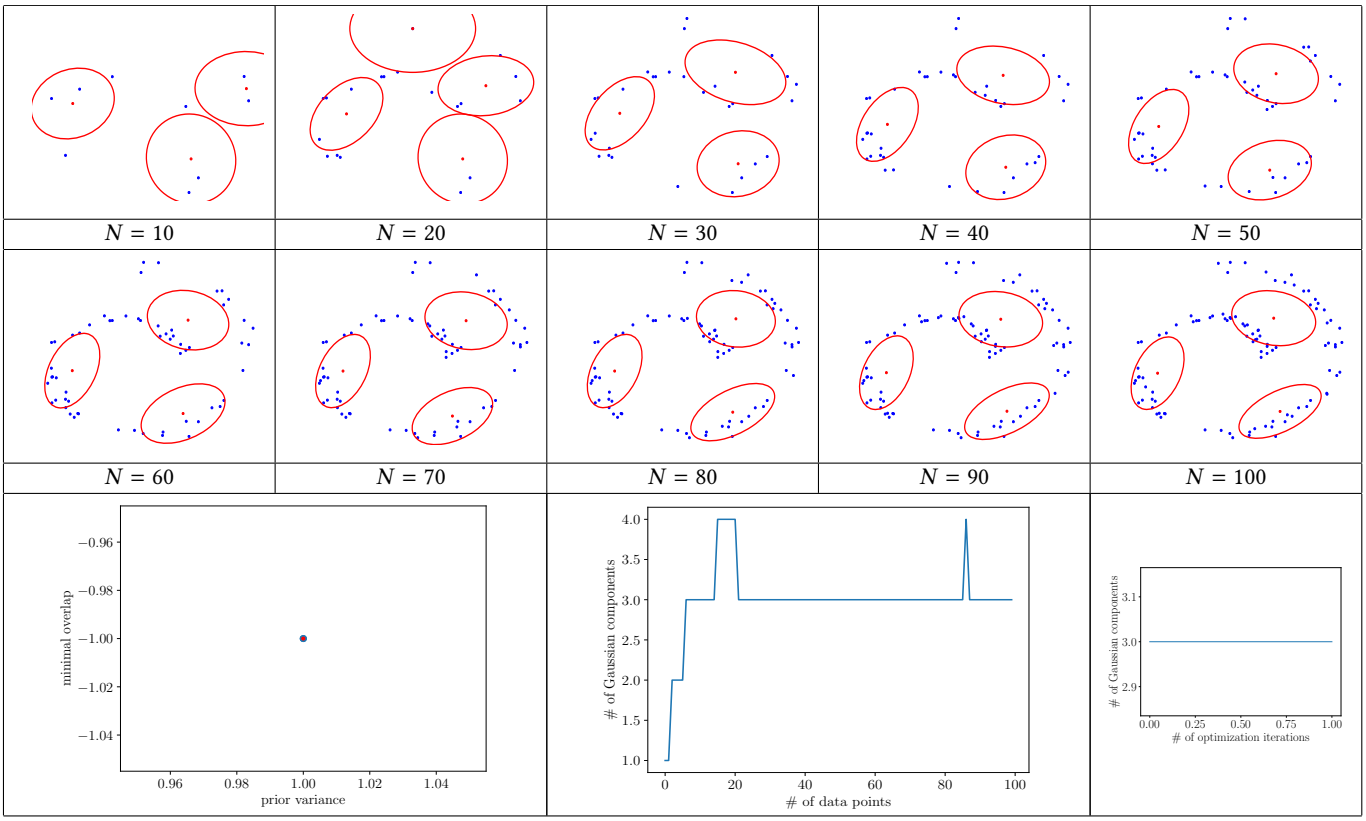


Table 13: Swiss roll in 2 dimensions with 100 data points

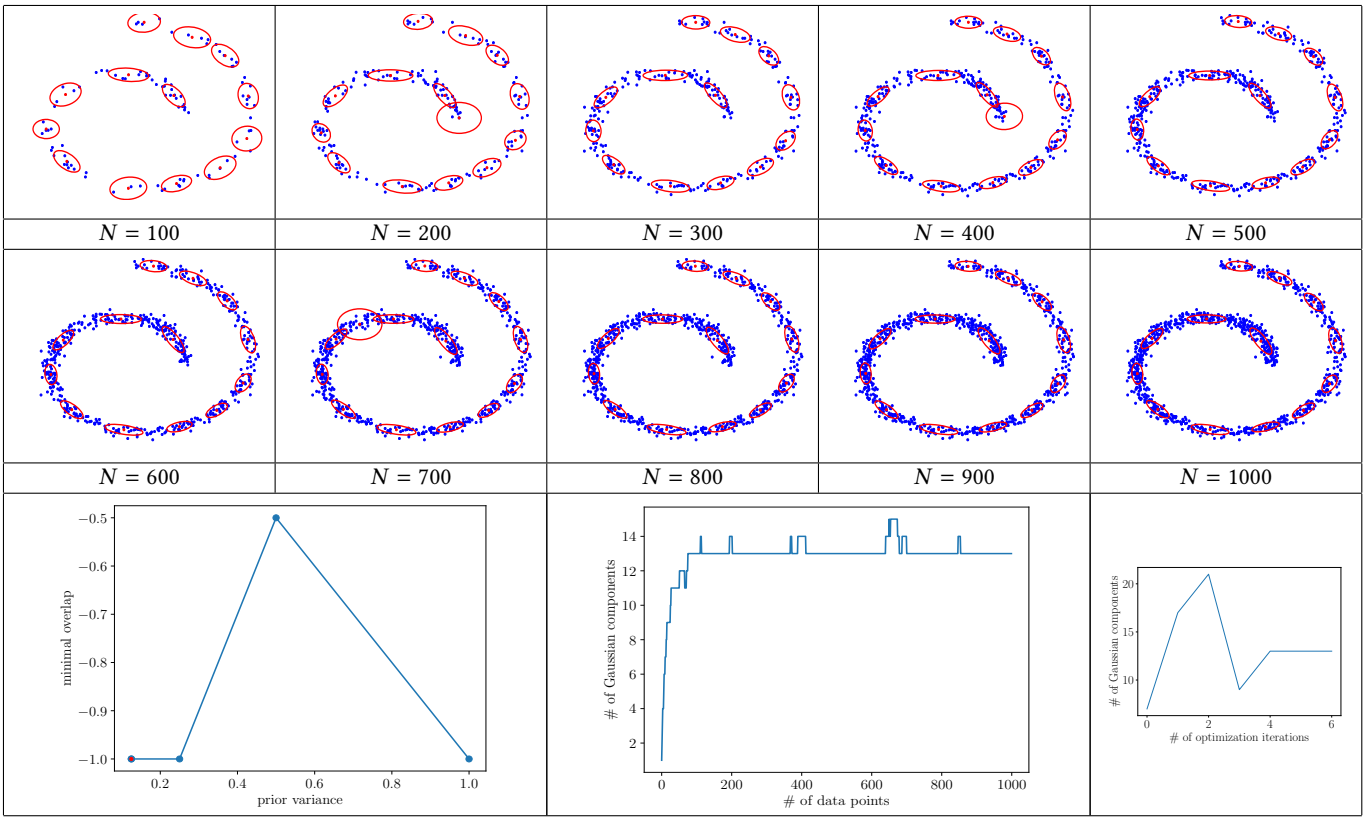


Table 14: Swiss roll in 2 dimensions with 1000 data points

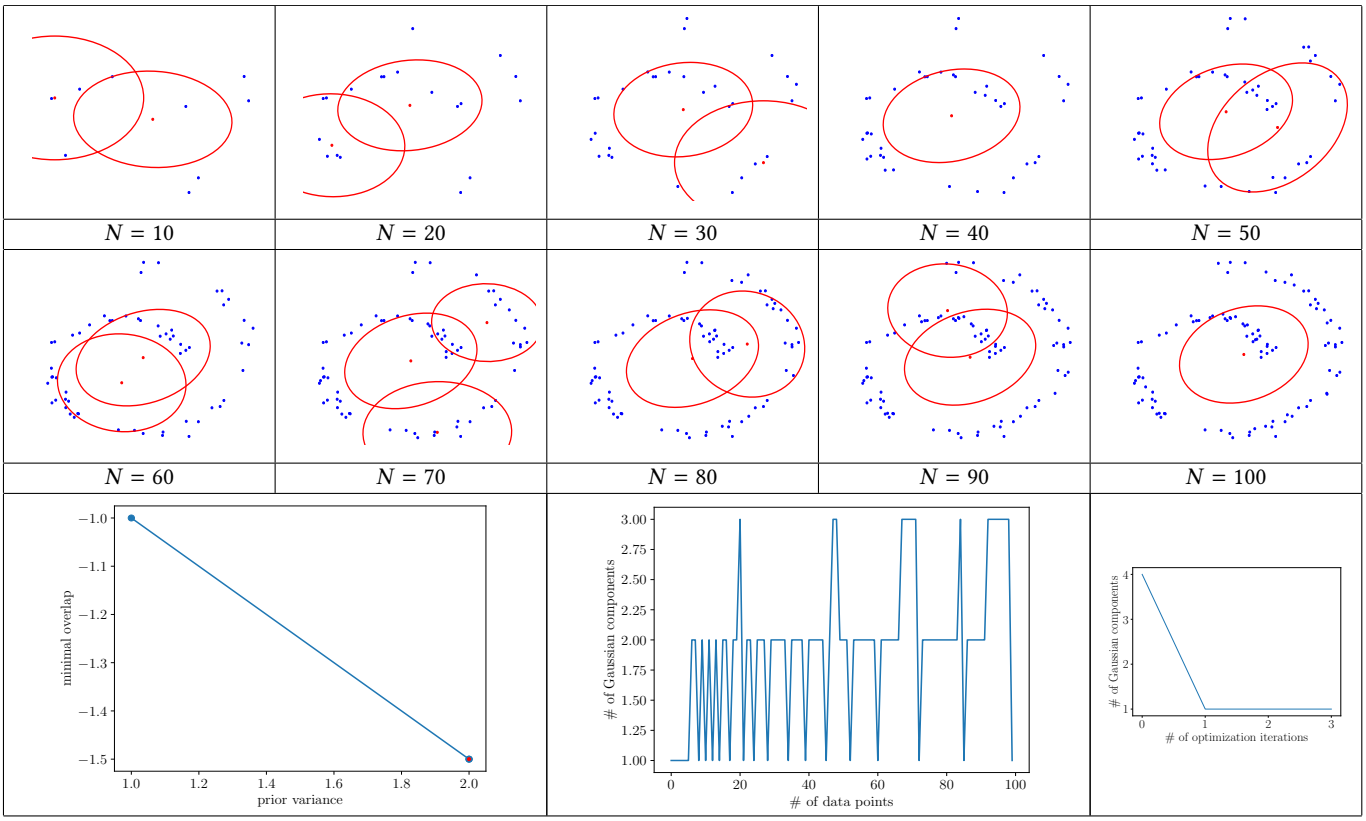


Table 15: Swiss roll in 10 dimensions with 100 data points

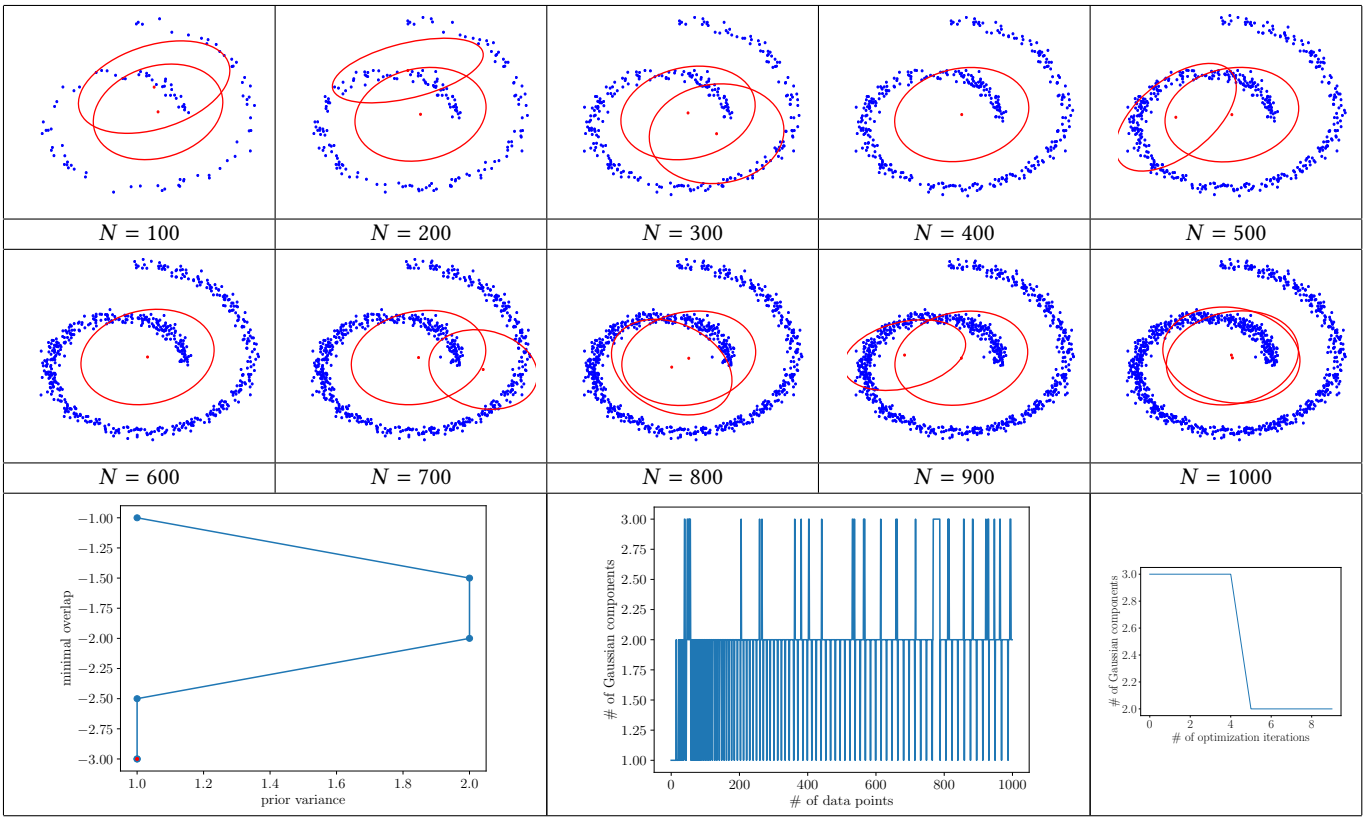


Table 16: Swiss roll in 10 dimensions with 1000 data points

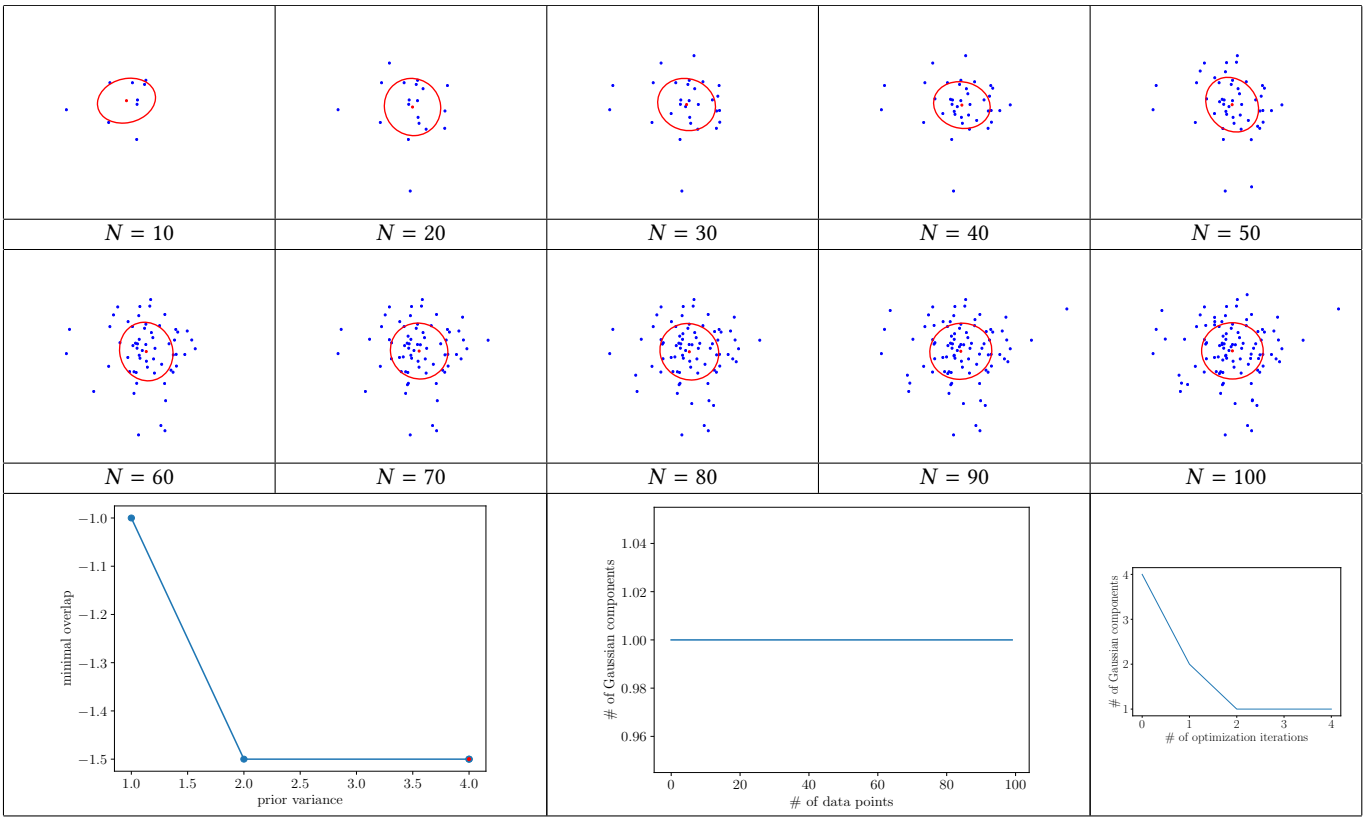


Table 17: Single Gaussian in 2 dimensions with 100 data points

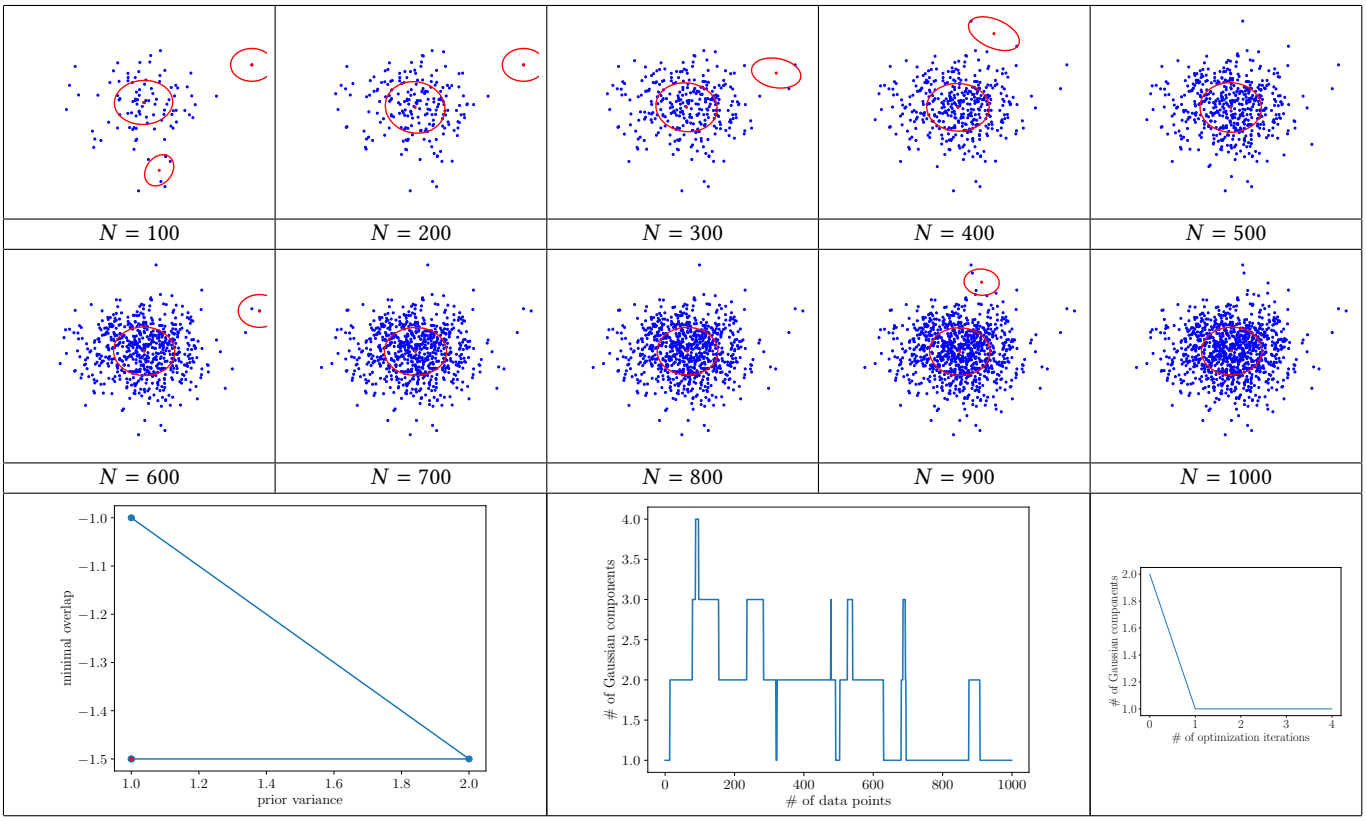


Table 18: Single Gaussian in 2 dimensions with 1000 data points

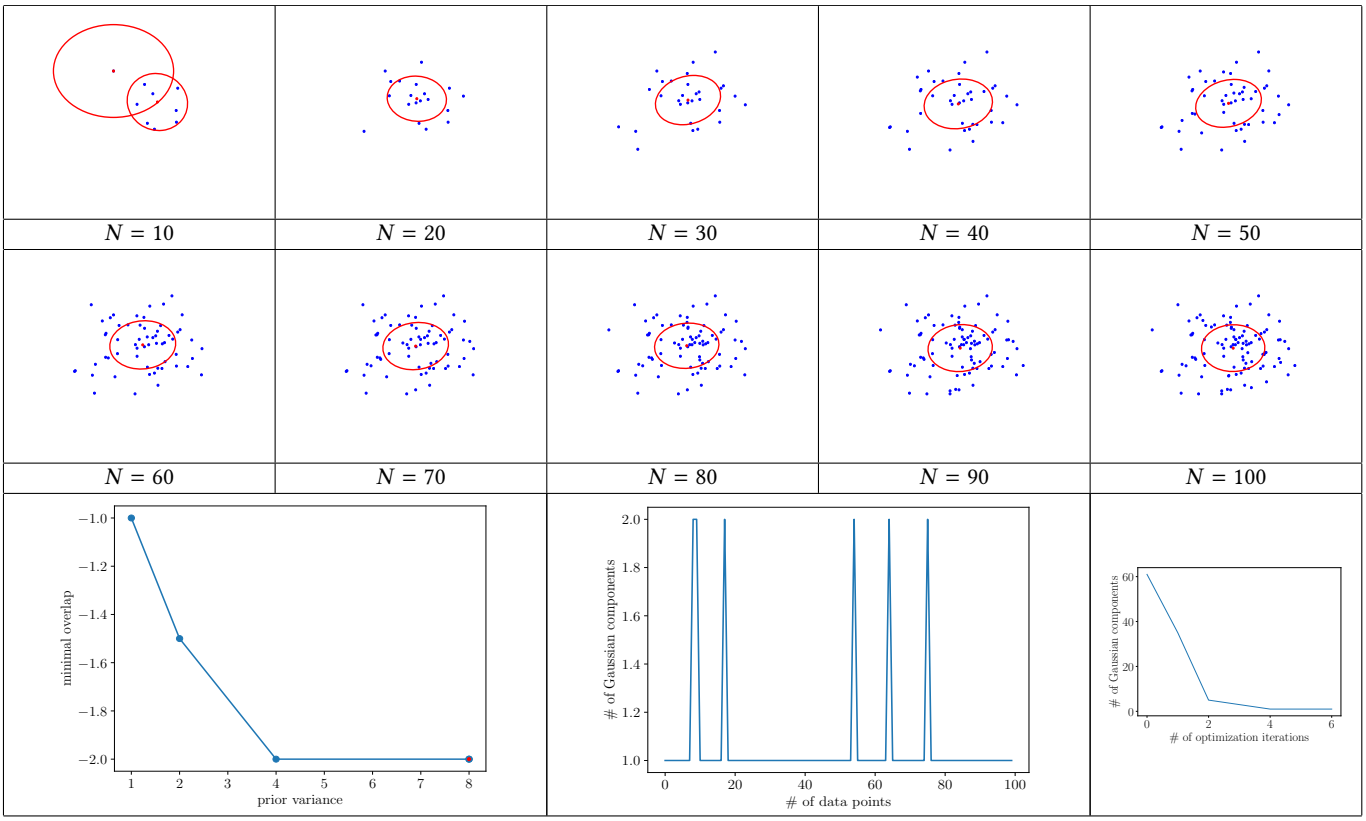


Table 19: Single Gaussian in 10 dimensions with 100 data points

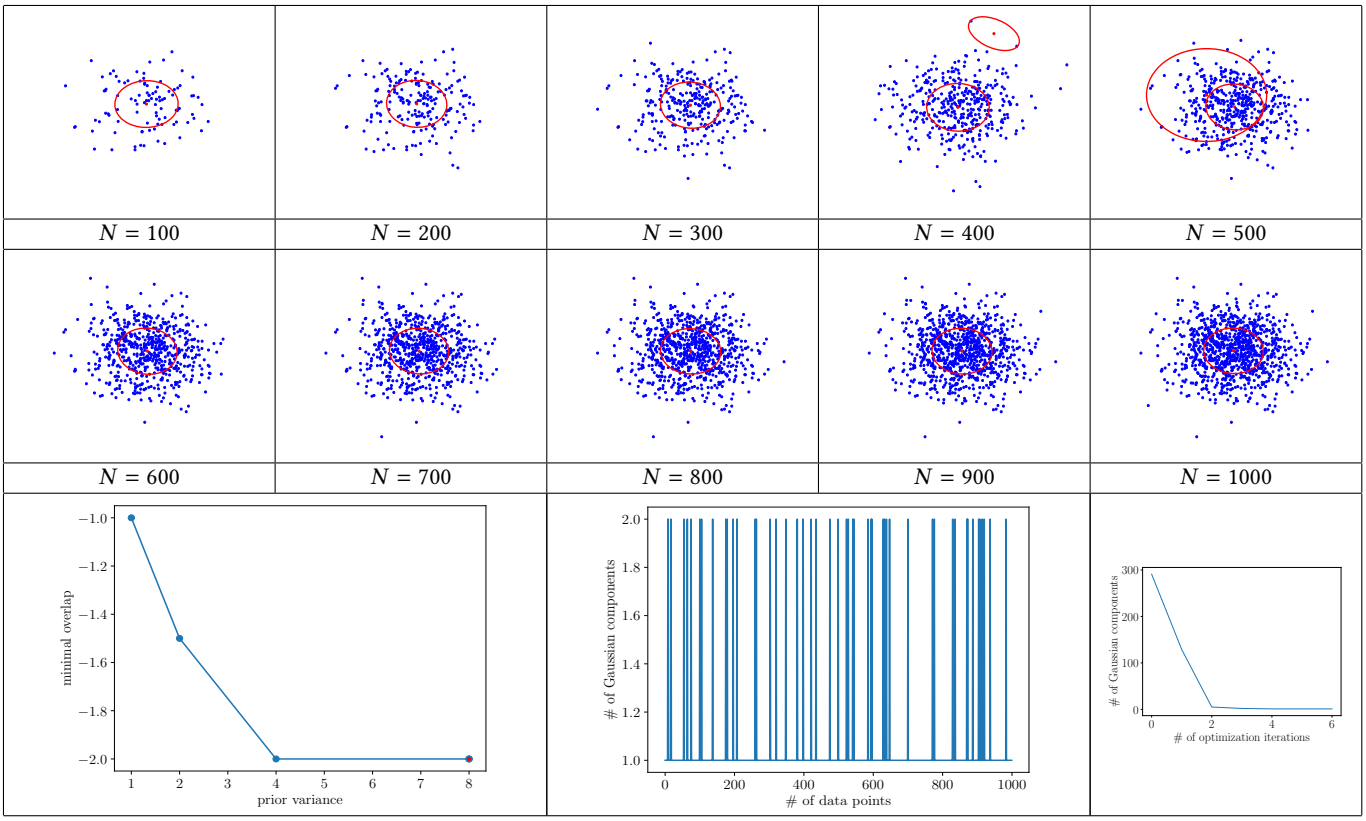


Table 20: Single Gaussian in 10 dimensions with 1000 data points

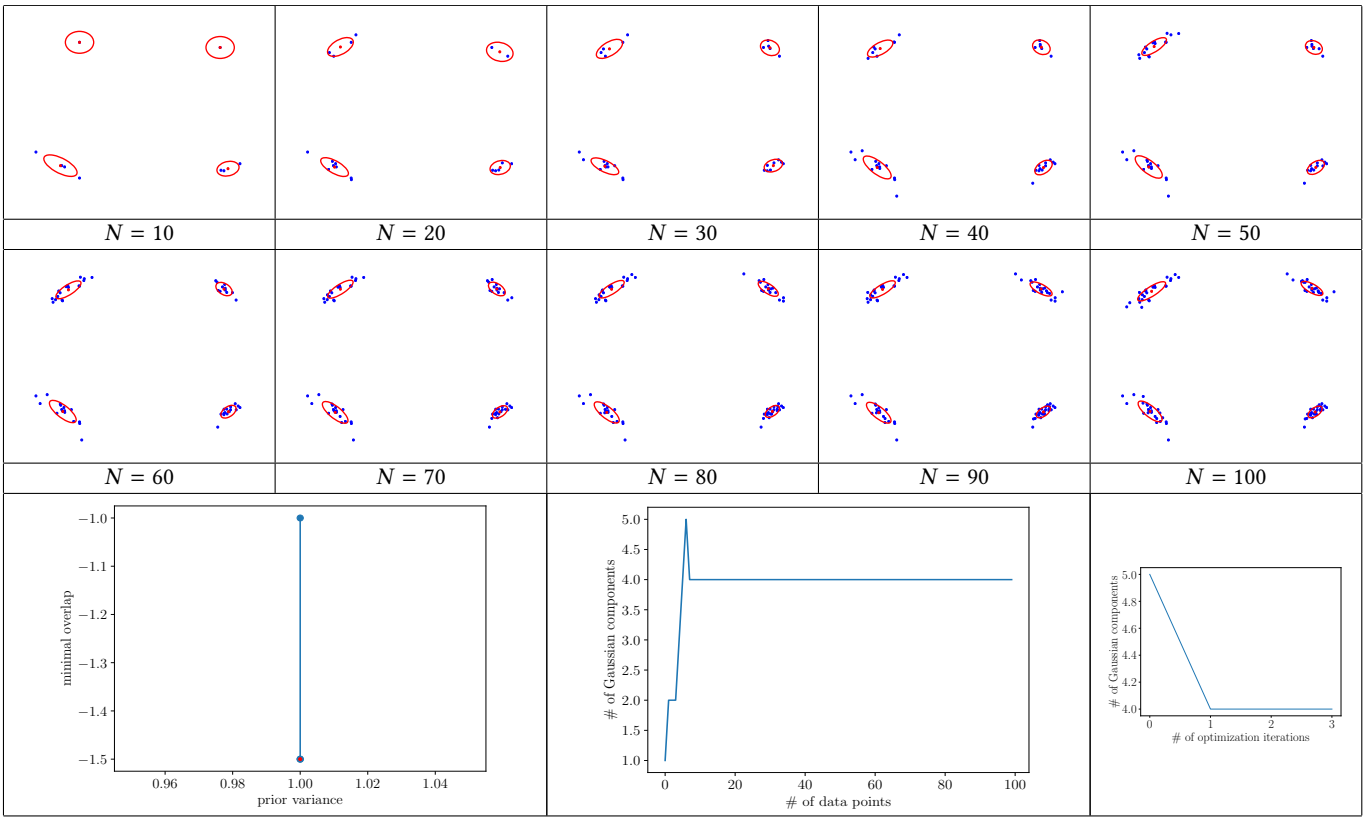


Table 21: Four Gaussians in 2 dimensions with 100 data points

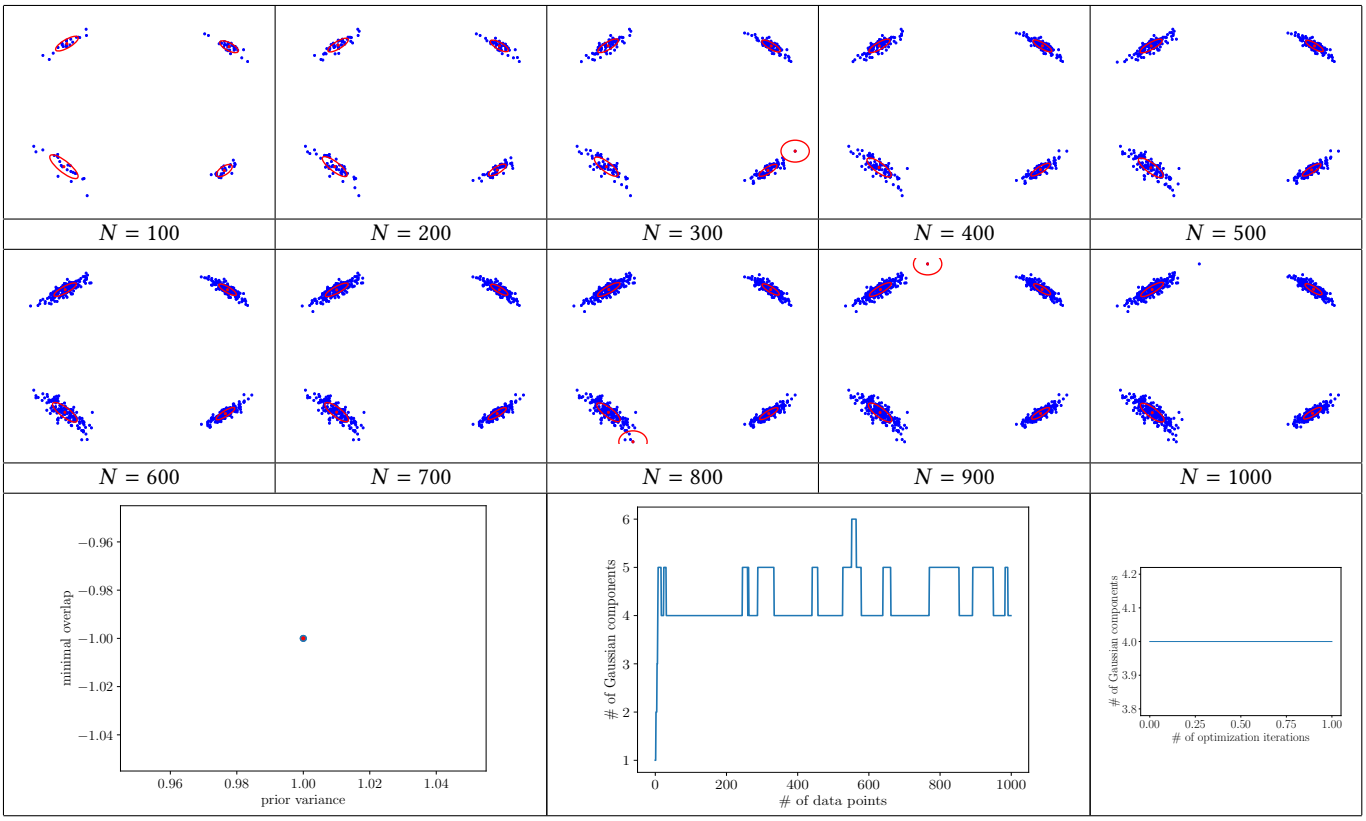


Table 22: Four Gaussians in 2 dimensions with 1000 data points

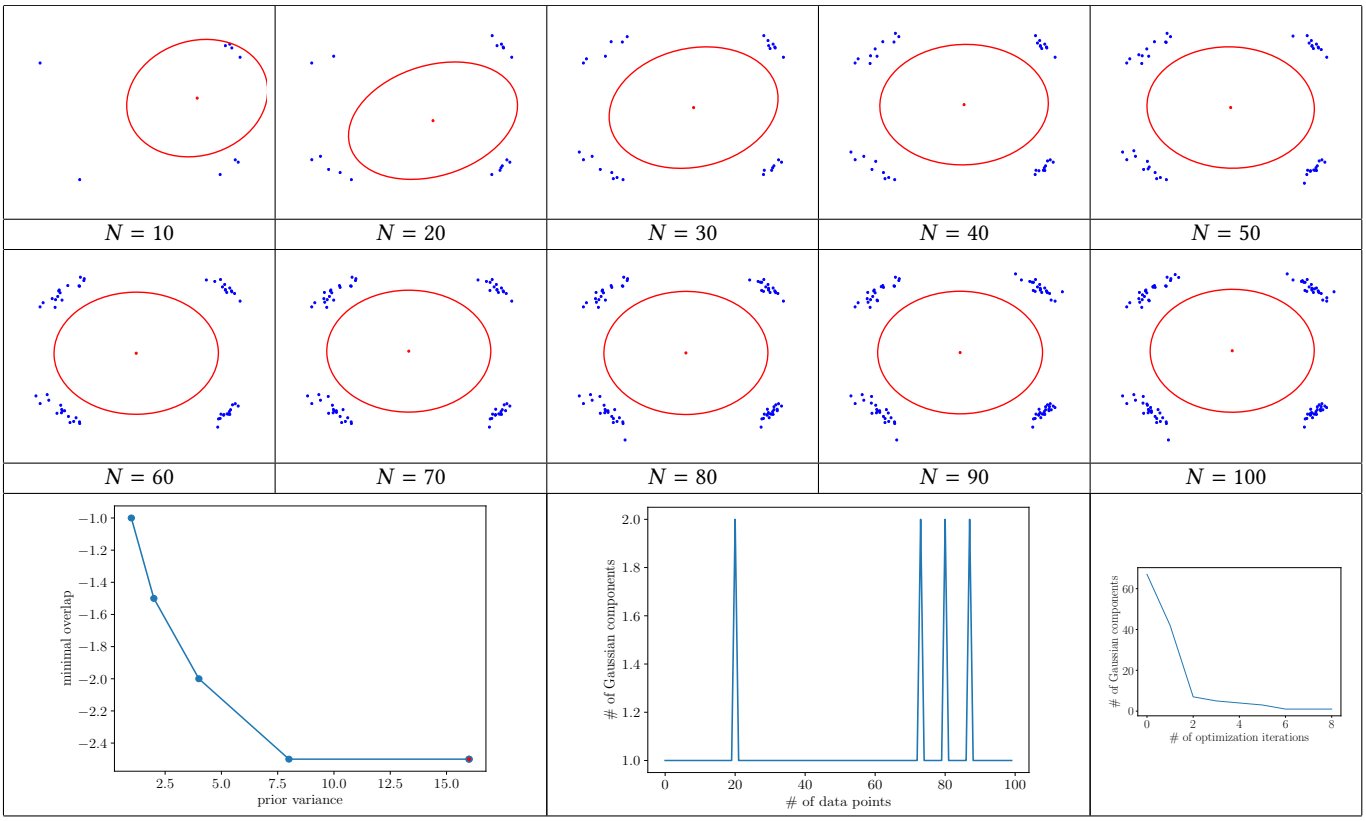


Table 23: Four Gaussian in 10 dimensions with 100 data points

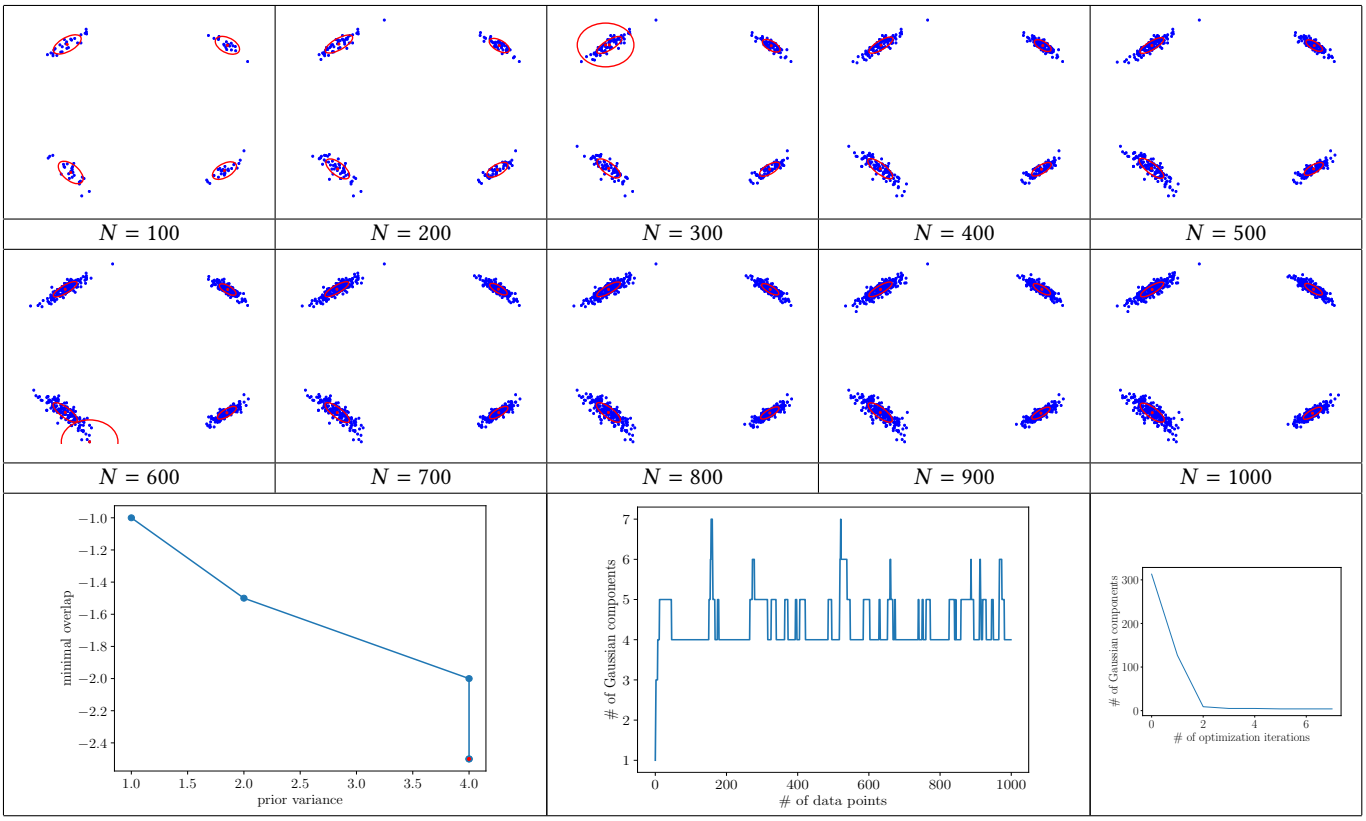


Table 24: Four Gaussians in 10 dimensions with 1000 data points

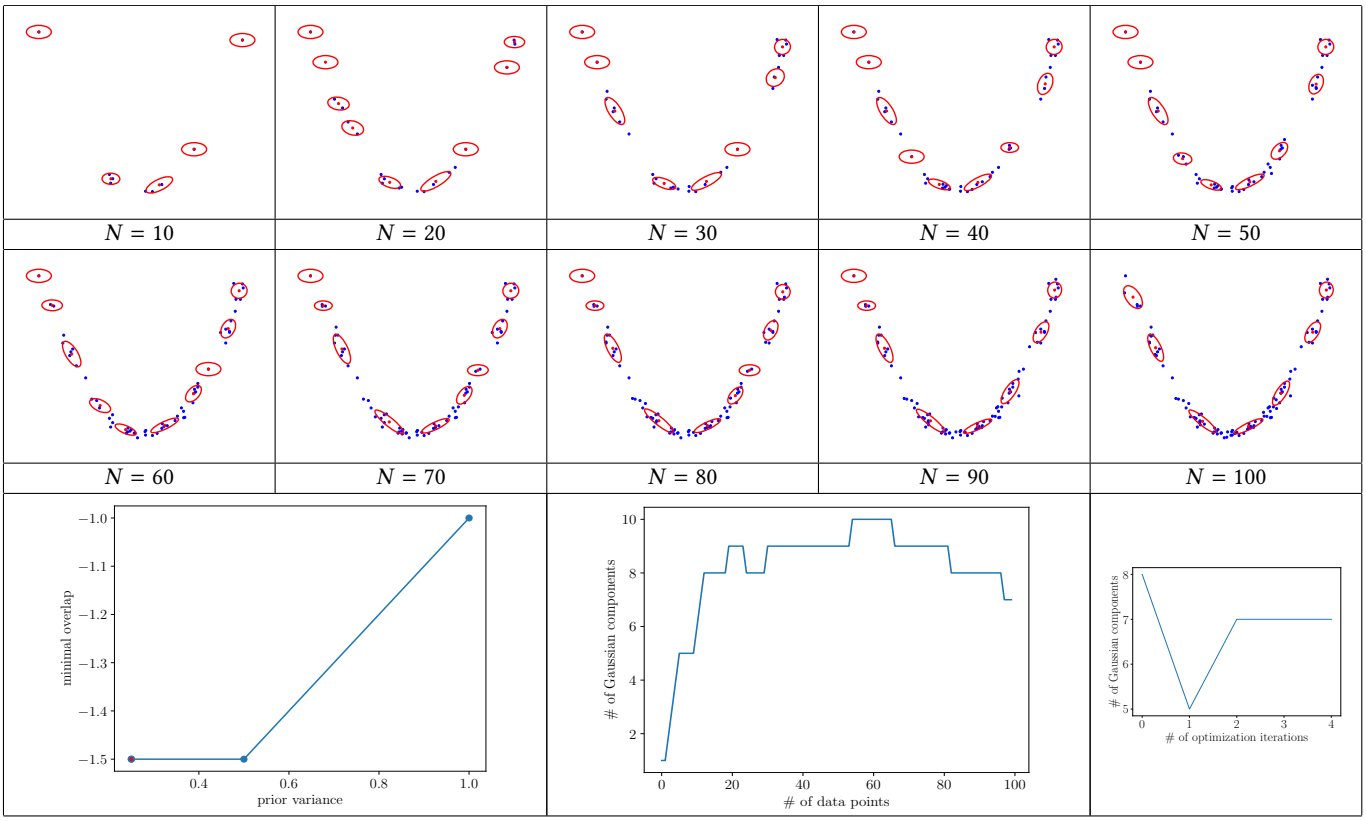


Table 25: Banana in 2 dimensions with 100 data points

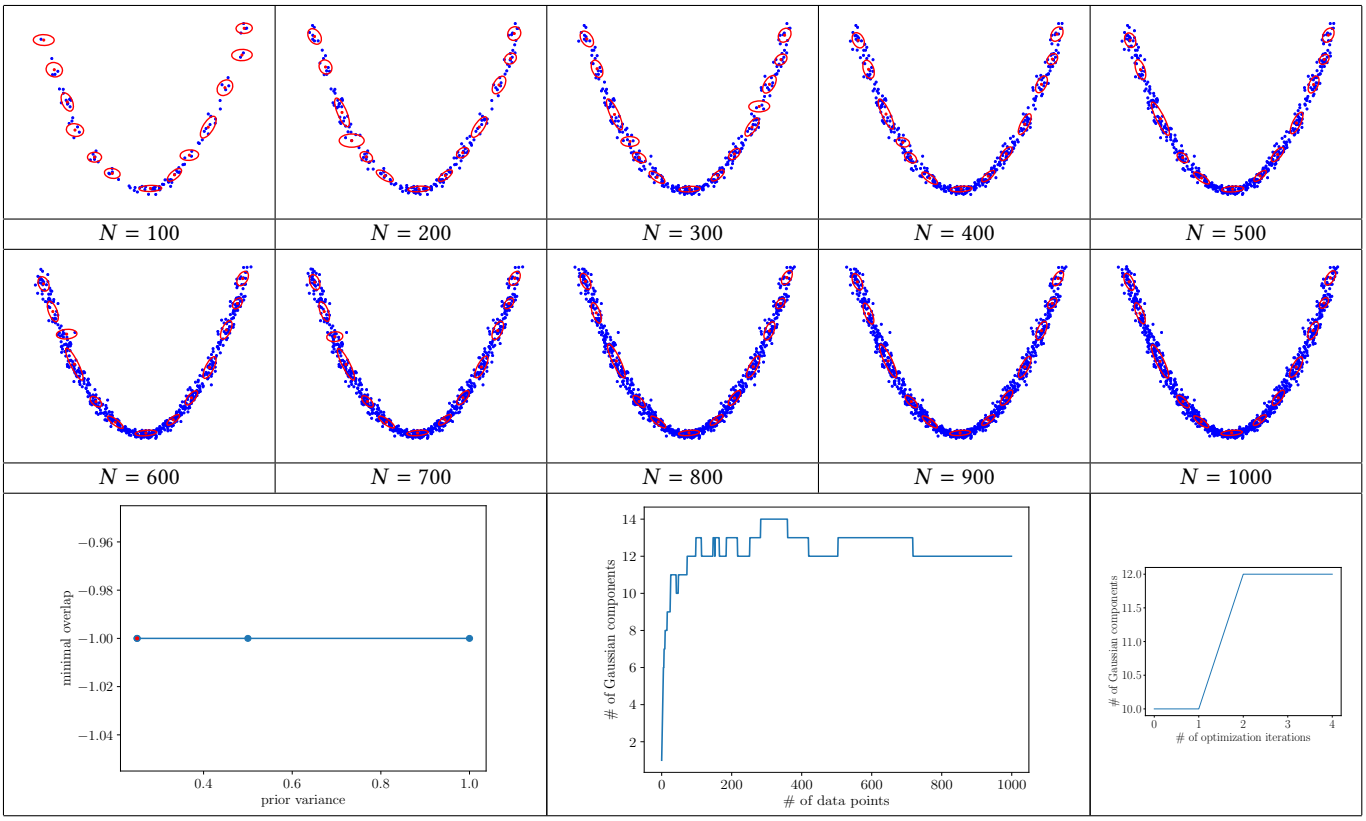


Table 26: Banana in 2 dimensions with 1000 data points

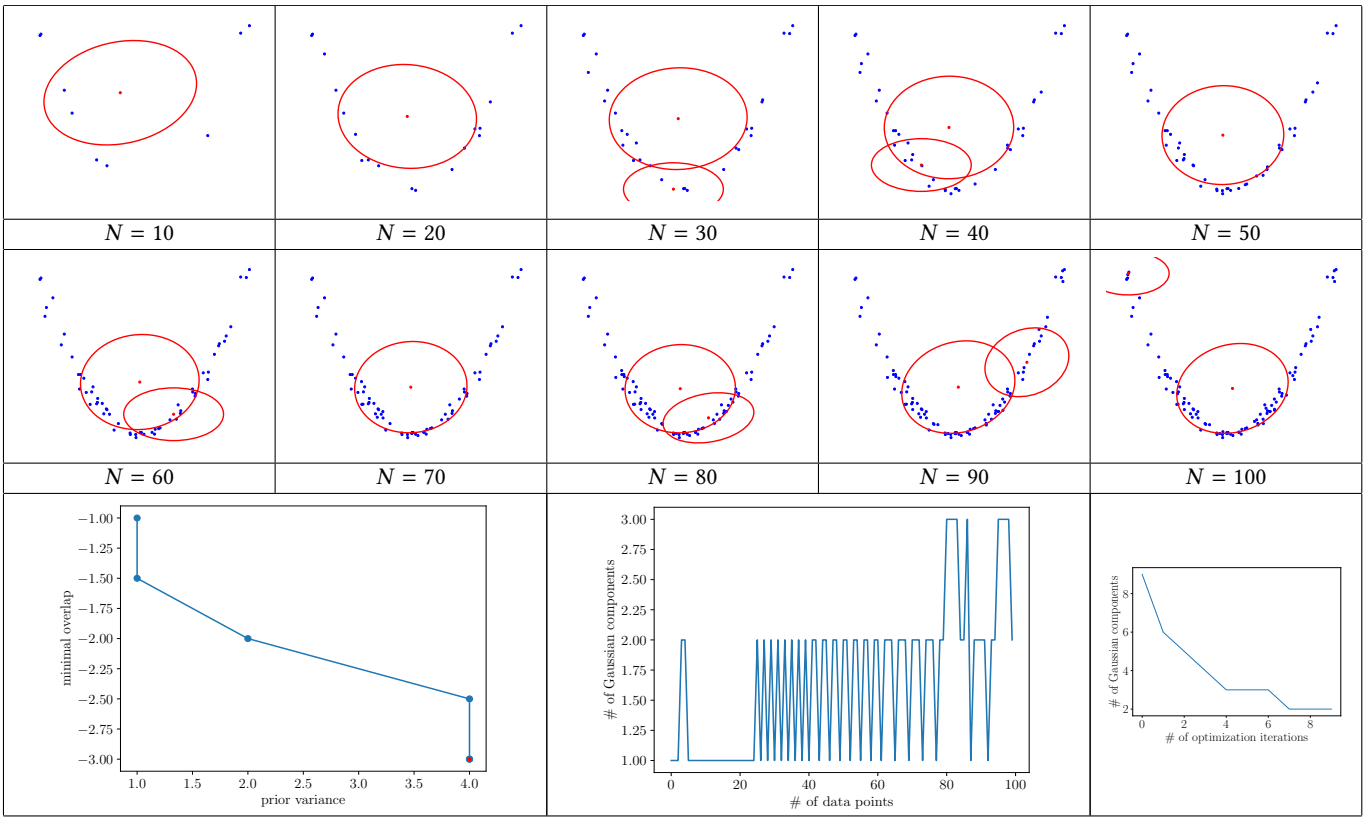


Table 27: Banana in 10 dimensions with 100 data points

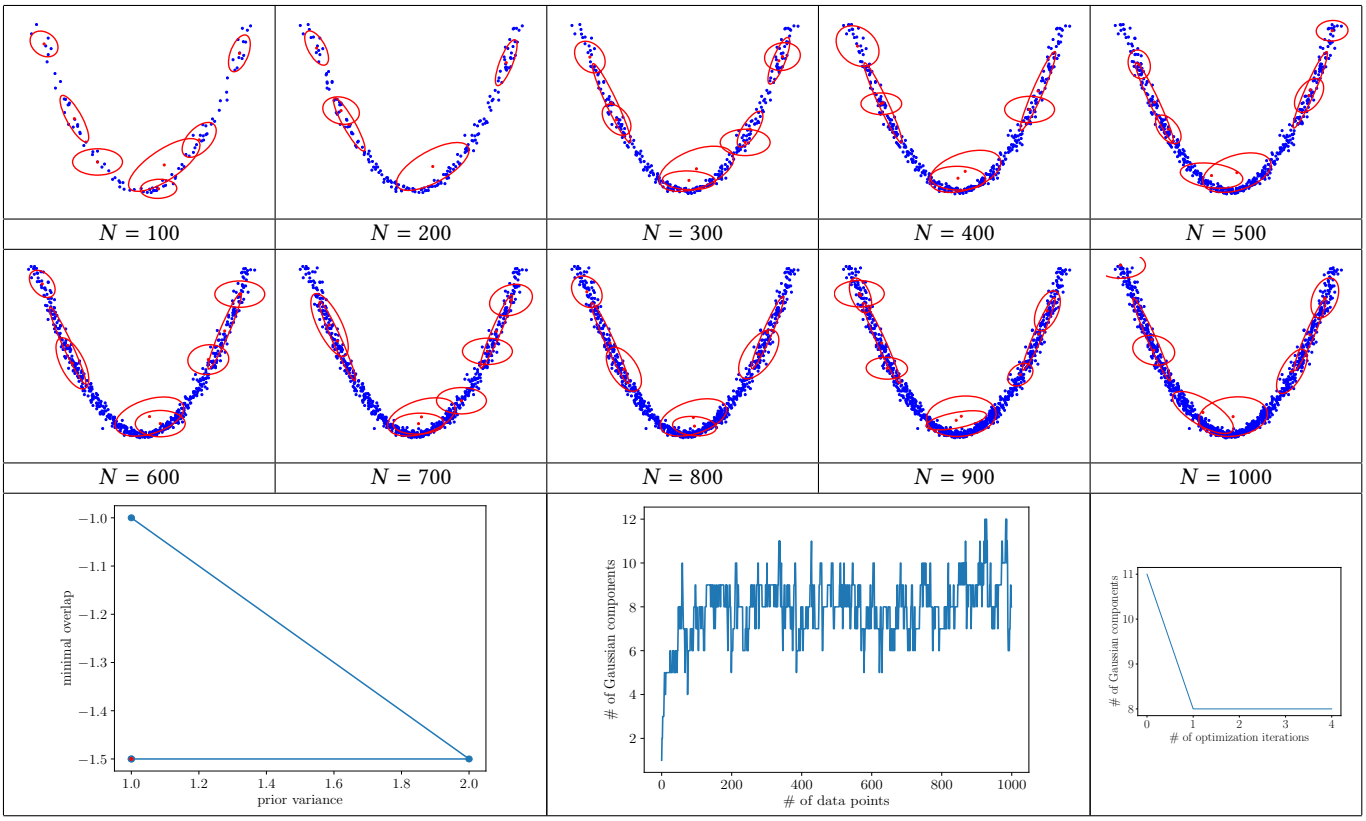


Table 28: Banana in 10 dimensions with 1000 data points

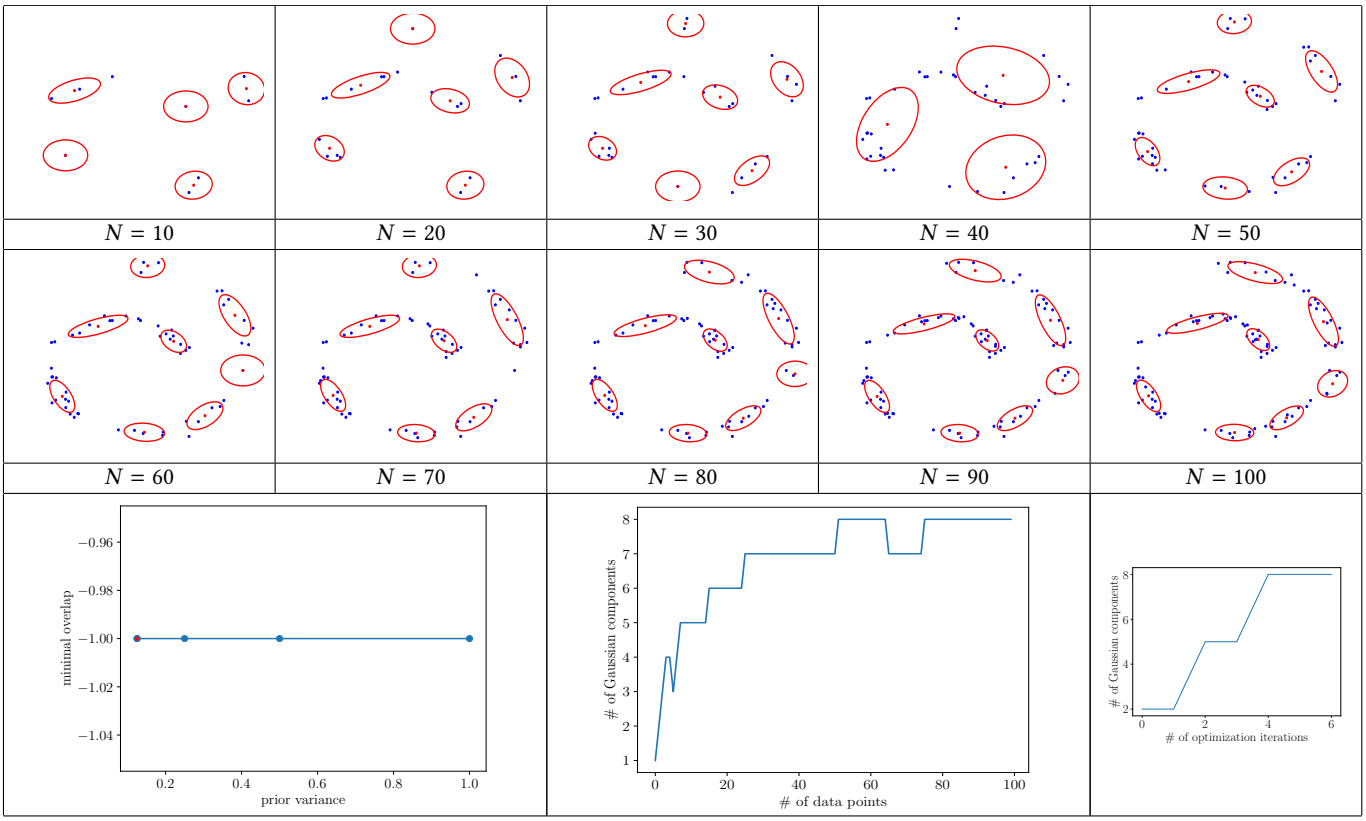


Table 29: Swiss roll in 2 dimensions with 100 data points

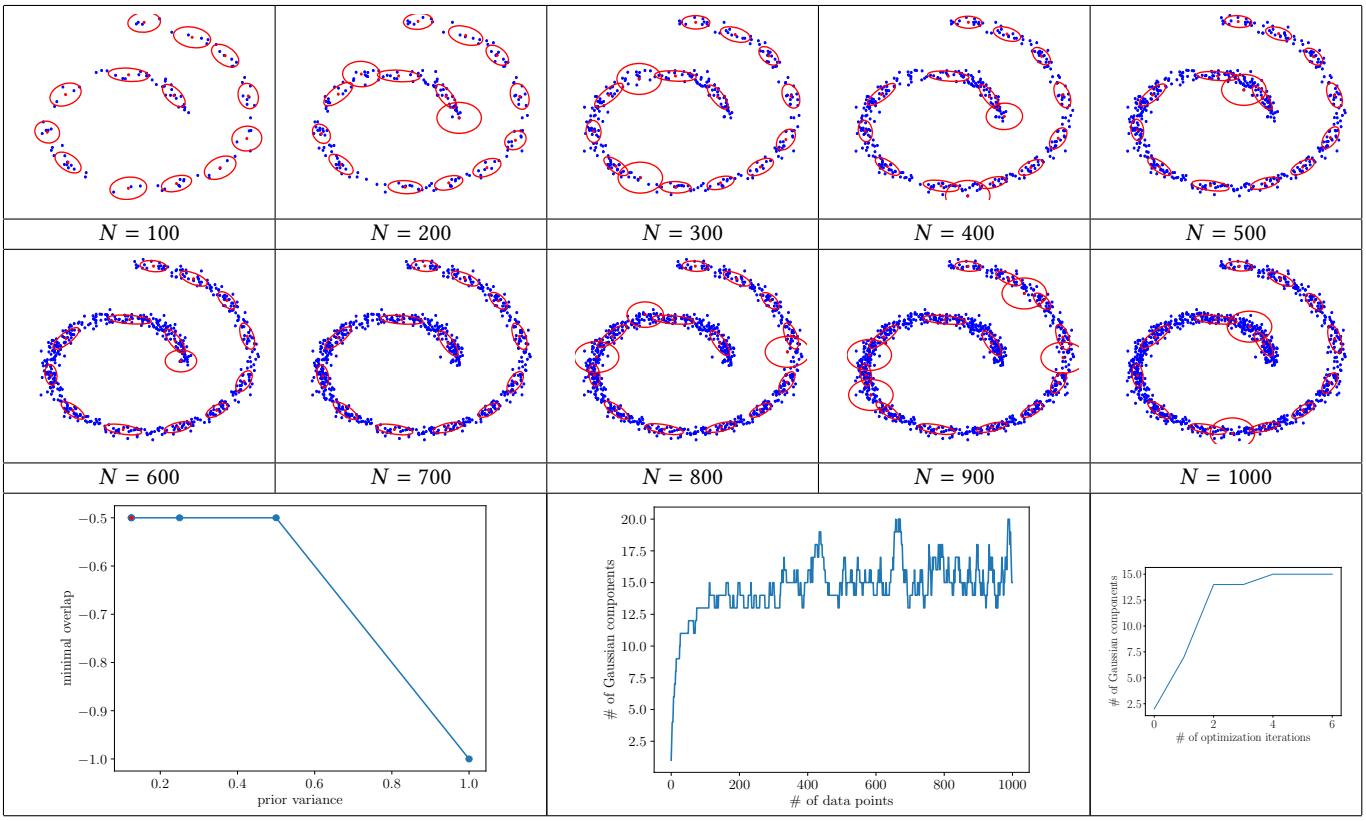


Table 30: Swiss roll in 2 dimensions with 1000 data points

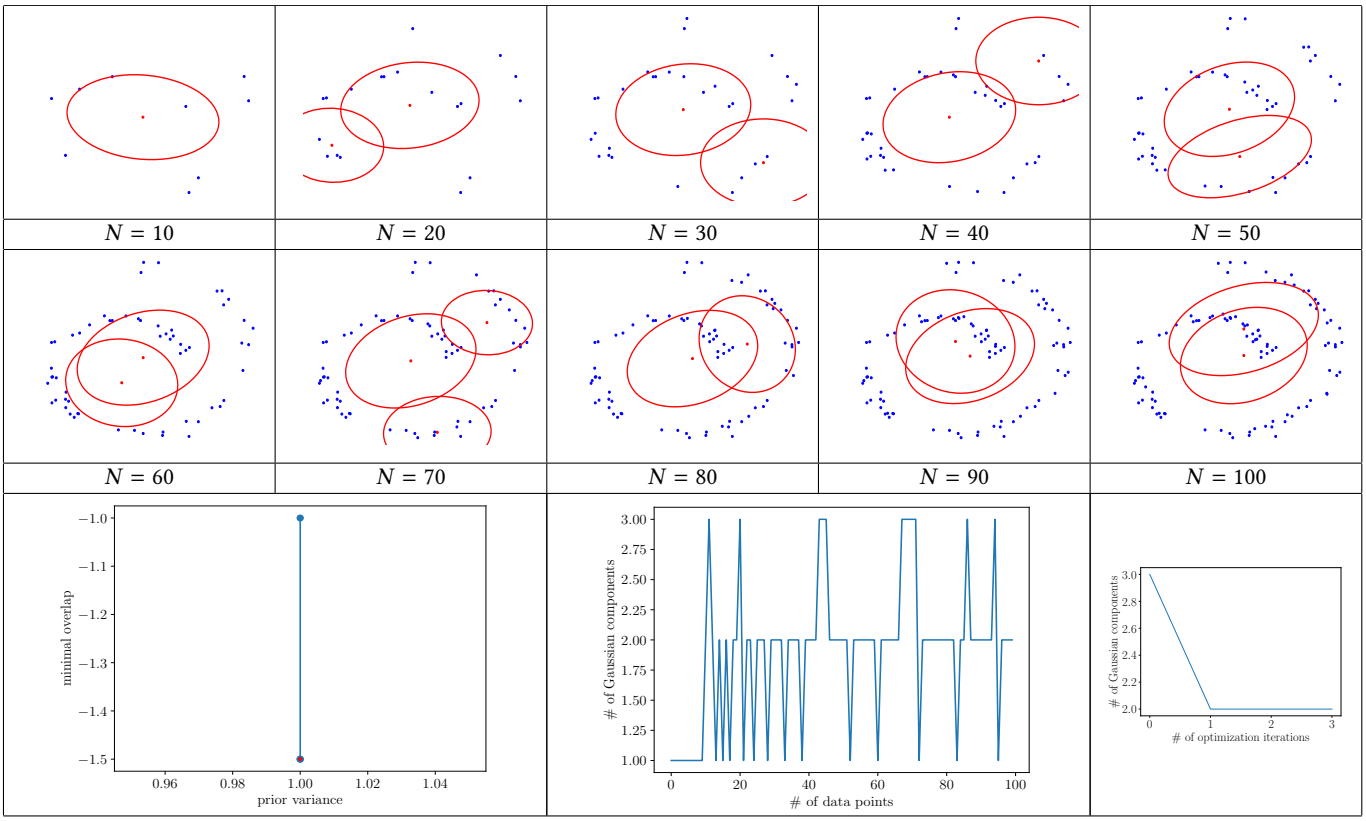


Table 31: Swiss roll in 10 dimensions with 100 data points

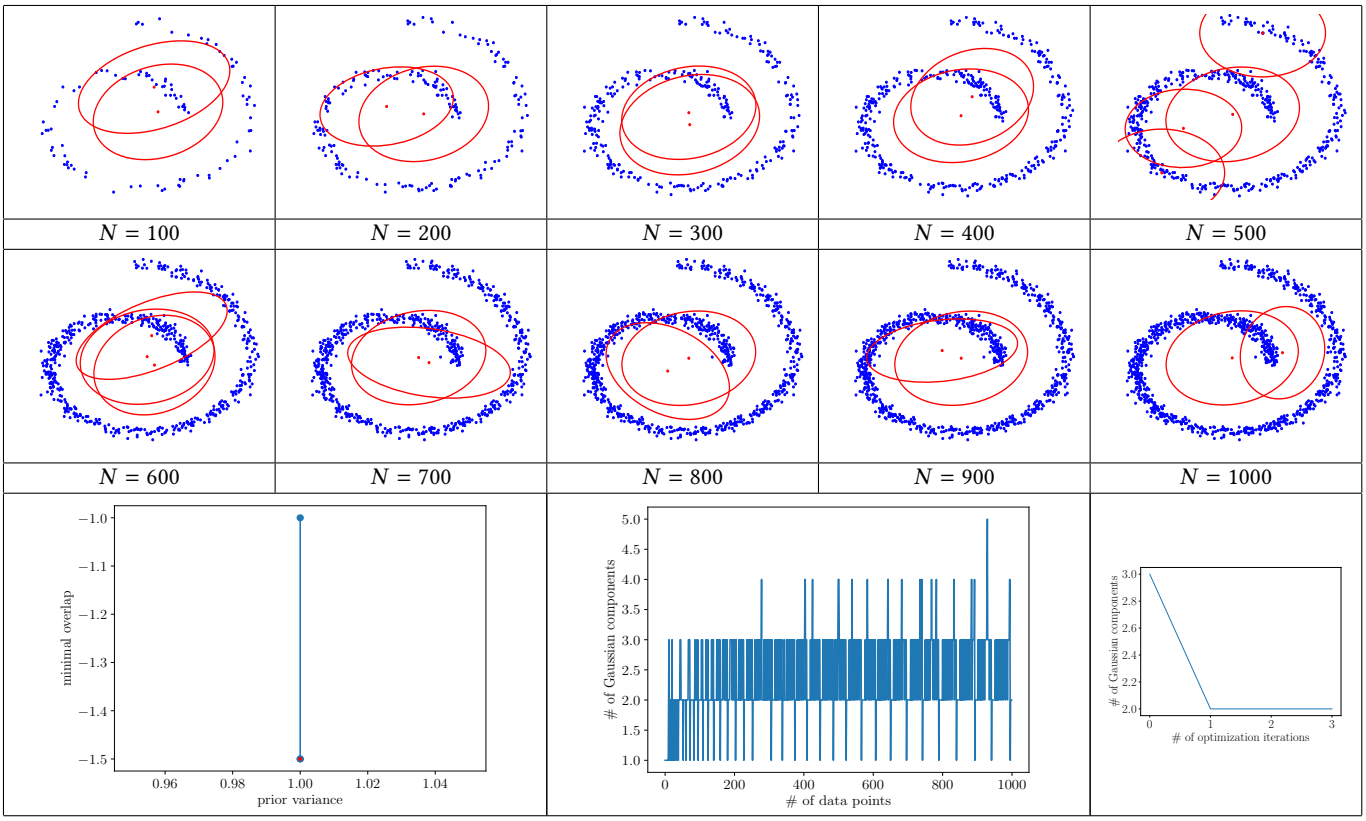
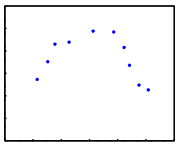
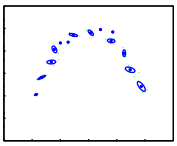


Table 32: Swiss roll in 10 dimensions with 1000 data points

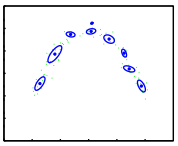
First level only:



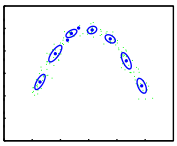
(a) n = 10



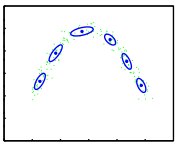
(b) n = 50



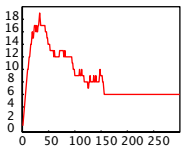
(c) n = 100



(d) n = 150

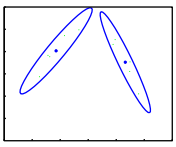


(e) n = 200

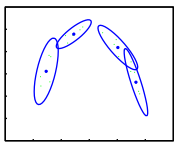


(f) number of Gaussians

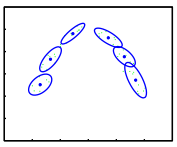
First and second level combined:



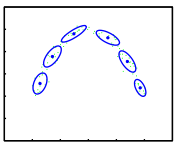
(m) n = 10



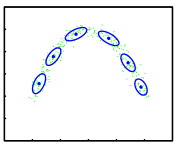
(n) n = 20



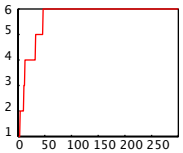
(o) n = 50



(p) n = 100

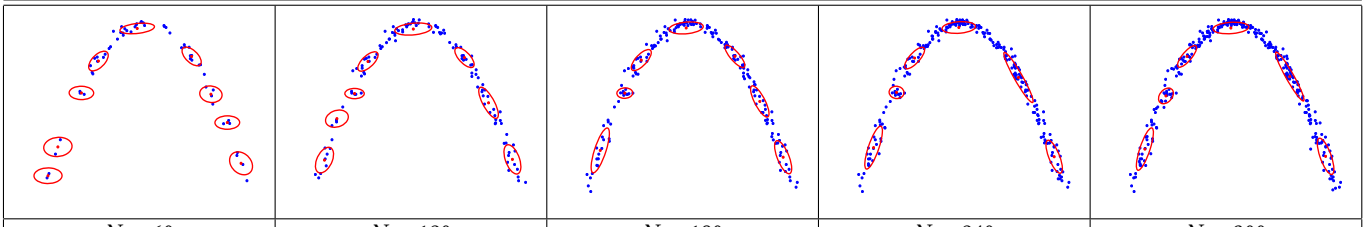


(q) n = 300

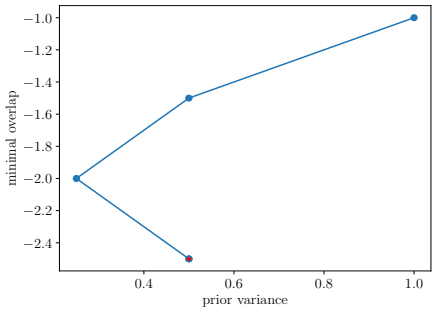


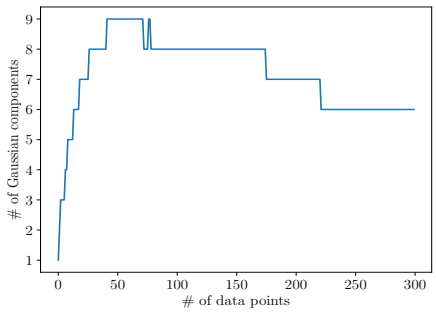
(r) number of Gaussians

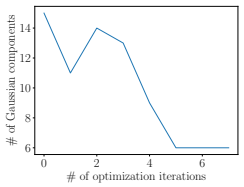
Results from competitor



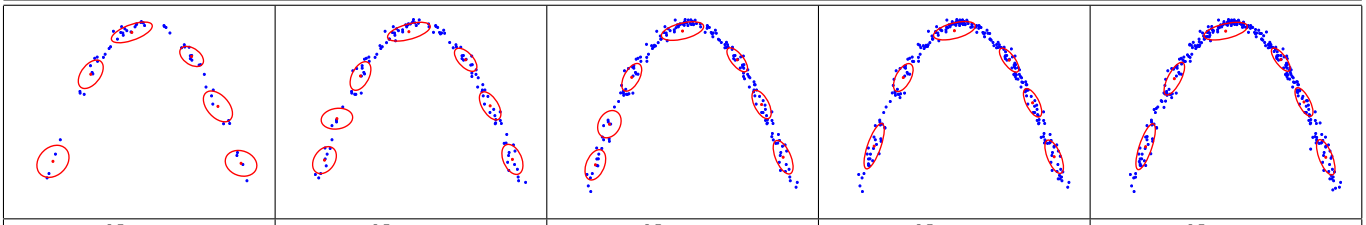
$N = 60$ | $N = 120$ | $N = 180$ | $N = 240$ | $N = 300$



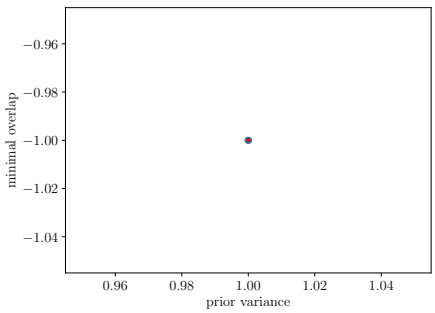


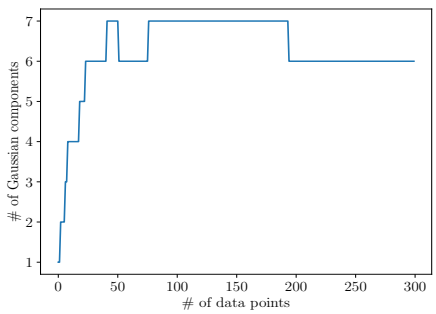


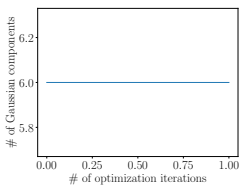
Our results with distance measure s_1



$N = 60$ | $N = 120$ | $N = 180$ | $N = 240$ | $N = 300$







Our results with distance measure s_2

Table 33: Comparison of our model against competitor