APPLICATIONS OF GAUSSIAN PROCESSES IN FINANCE
COMP 652 - COURSE PROJECT

Hossein Hejazian
Desautels Faculty of Management
McGill University
Montreal, Quebec, Canada
hossein.hejazian@mail.mcgill.ca

Zahra Jalali
Desautels Faculty of Management
McGill University
Montreal, Quebec, Canada
zahra.jalali2@mail.mcgill.ca

ABSTRACT

Estimating covariances between financial assets plays a pivotal role in most financial problems. There are several estimation methods in the literature including empirical estimates which are not very reliable in practical cases where the number of examples is considerably low in comparison to the number of variables. The paper we selected for the Reproducibility Challenge applied a Gaussian Process Latent Variable Model (GP-LVM) to replace classical linear factor models with a nonlinear extension of these models. They showed that using non-linear models can improve the estimation of the covariances. In addition, they investigated effects of the new model on several applications in finance. In this report, we aim to scrutinize the validity of their claims on which we evaluate the replicability and reproducibility of the results provided in this paper. We find out that their results are mostly reproducible. However, we faced several implementation issues.

1 INTRODUCTION

Covariance matrix is a statistical tool to implement diversification defined as a means of managing risk in finance. Using this matrix, we can compute the correlation existing among each two assets in our portfolio as well as the variance of each asset located on the diagonal of this matrix. The other example of using covariance matrix is in the capital asset pricing model (CAPM). In this model, beta captures the volatility of a security in comparison to the market as a whole which draws from the covariance matrix to gauge an investor's risk exposure specific to one security. To better understand the importance of covariance matrix in finance, we refer to a simple example presented in Wilson & Ghahramani (2011). "Imagine the price of the NASDAQ composite index increased dramatically today. Will it continue to rise tomorrow? Should you invest? Perhaps this is the beginning of a trend, but it may also be an anomaly. Now suppose you discover that every major equity index FTSE, NIKKEI, TSE, etc. has also risen. Instinctively the rise in NASDAQ was not anomalous, because this market is correlated with other major indices."

In practice, estimating covariance matrices for high dimensional observations is extremely challenging. Most models presented in the related literature considered some simplifying assumptions to cope with this problem, reducing the applicability of resulted approaches. Estimating covariance matrices have recently gained attention from machine learning society. Two main practicable studies in this field are Wilson & Ghahramani (2011) and Wu et al. (2014). Wilson & Ghahramani (2011) presented a stochastic process with Wishart marginals to model time-varying covariance matrices for a diverse class of structures. Wu et al. (2014) introduced a Gaussian Process Volatility Model (GP-Vol) for time-varying variances. (For other applications of machine learning in Finance see Chapados & Bengio (2008); Nevmyvaka et al. (2006); Heaton et al. (2016))

In contrast to above mentioned studies, our selected paper focused on fixed covariance matrices. This study used the similarity between the structure of the GP-LVM and asset returns formulation presented in arbitrage pricing theory in order to apply non-linear kernels to estimate fixed covariance matrices. They showed that using non-linear kernels can improve the resulted model based on two measurements: R-squared (R^2) score and evidence lower bound (ELBO). They also discussed three applications of the presented model in finance: portfolio optimization, prediction of missing values, and interpretation of the latent space to structure finance data. Experiments show that non-linear

models result in less variance and higher Sharpe ratio value for portfolio in comparison to the linear models proposed in the literature. The GP-LVM can also be used to predict missing prices when no trading took place. Experiments depict that non-linear models result in better prediction for missing values in comparison to linear models based on the R^2-score and the average absolute deviation.

A cornerstone of science is the possibility to critically assess the correctness of scientific claims made and conclusions drawn by other scientists Plesser (2018). The 2016 Nature survey demonstrated that more than 70% of researchers have tried and failed to reproduce another scientists experiments, and more than half have failed to reproduce their own experiments. Reproducibility is also critical in machine learning, especially for industrial grade model development. Without this, data scientists risk claiming gains from changing one parameter without realizing that hidden sources of randomness are the real source of improvement Villa & Zimmerman (2018).

We selected the paper "Applications of Gaussian Processes in Finance" to evaluate its replicability and reproducibility[1] (Paper). As mentioned before, the paper presented a simple model to estimate covariance matrices for high dimensional data in finance. Therefore, the reproducibility is highly important for this paper focusing on the applications of ML techniques and may be used for real problems. To do so, we forked the Github repository developed by the authors which can be accessed here: `https://github.com/RSNirwan/GPsInFinance.git`.

The sections of this report are broken down as follow. First, we discuss main issues we faced during implementing the experiments in section 2. Then, section 3 discusses replicability of all the results presented in the paper. The reproducibility of paper is presented in section 4. We discuss some weaknesses of the paper in section 5. Finally, a brief conclusion is presented in section 6.

2 Implementation issues - Learning without depictable results

Despite that we had access to the code scripts developed by the authors of the selected paper, we were not able to run them immediately in the first place. In fact, we faced time-consuming problems during the implementation, some of which are briefly discussed in this section because they affected the design of our experiments for the rest of the project. First of all, we encountered some difficulties in terms of parallel computing. As the authors also emphasized through our communication, working with the package `ipyparallel` is a bit tricky. We tried to configure the parallel computation package on three available devices as PC (Win10 OS), MacBook (Mac OS), and GCP (Debian OS), making this issue more challenging and time-consuming. In addition, implementing the codes on Windows OS faced with some issues mainly because the `Pystan` package had been exploited in the code scripts, which is a package developed in C++ environment and applying it on Windows is somewhat tricky.

On the other hand, We tried to implement some of the experiments without parallel computation, i.e. run the models sequentially, but we found the authors' claim was true as if the list of inputs is too long and we are not running `ipyparallel`, it might take a lot of time. In fact, it can work only for tiny size problems. Models with the same size mentioned in the paper might take 1 or 2 days to be solved completely (we ran one experiment for 12 hours on the GCP instance with sequential computation in which we did not obtain any results). Therefore, we can conclude that the results of the paper cannot be reproducible without parallel computing.

One of the other main obstacles preventing us from doing a comprehensive ablation study on the paper is that the GCP instance on which we run our experiments froze from time to time for no apparent reason.[2] Unfortunately, rebooting the GCP instance or even creating a new GCP instance solved this problem only temporarily. As GCP became problematic during our experiments, we were forced to spend part of our efforts on solving or at least on dealing with this issue. Finally, we were almost successful to cope with these problems on both our laptop and GCP following the instruction the authors specified in our communication in which we found out how and to what extent run our

[1] The Github issue link is:
`https://github.com/reproducibility-challenge/iclr_2019/issues/100`.

[2] Meanwhile we faced this problem, we communicated with The TA Koustuv Sinha but, after checking the GCP instance, he said he did not know why the instance was freezing like this since on the GCP dashboard it showed connected.

experiments on the GCP instance and our laptop. However, due to the limited resources, we have to reduce the size of the problem in some experiments and to limit our ablation study.[3]

3 REPLICABILITY

Replicability means anyone can recreate result data and graphs demonstrating similar behavior with what shown in the paper Villa & Zimmerman (2018). In this section, we discuss the replicability of almost all the results presented in the paper. To better understand the discussion presented here, first we briefly discuss the model proposed in the paper.

The paper applied GP-LVM technique which reduces the dimensions of the data $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ from D to Q in a non-linear way and at the same time estimates the covariance matrix between the N points. The generative procedure takes the form $\boldsymbol{Y}_{n,:} = \boldsymbol{f}(\boldsymbol{X}_{n,:}) + \epsilon_n$, where $\boldsymbol{f} = (f_1, ..., f_D)^T$ is a group of D independent samples from a Gaussian process (GP), i.e., $f_d \sim GP(0, k(.,.))$. Applying optimization methods to calculate the marginal likelihood of $\boldsymbol{Y}$ (i.e., $p(\boldsymbol{Y}|\boldsymbol{X})$) may cause overfitting. For this reason, a variational inference framework presented by Michalis & Lawrence (2010) used to approximate the true posterior $p(\boldsymbol{X}|\boldsymbol{Y})$.

In finance, based on the capital asset pricing model, the expected returns of an asset $r_n \in \mathbb{R}^D$ for D days and its risk β_n can be calculated through $\mathbb{E}[r_n] = r_f + \beta_n \mathbb{E}[r_m - r_f]$ where $r_f \in \mathbb{R}^D$ is the risk free return on D different days and r_m is the market return. This model was generalized for multiple risk factor $\boldsymbol{F}$ by arbitrage pricing theory $r_n = \alpha_n + \boldsymbol{F}\beta_n + \epsilon_n$. This model can be written in the form $r_{n,:} = \boldsymbol{f}_{(n,:)} + \epsilon_n$ which matches the form of GP-LVM. Therefore, GP-LVM can be used to estimate the matrix $\boldsymbol{B} = (\beta_1, ..., \beta_N)^T$. After inferring $\boldsymbol{B}$ and the hyperparameters of the kernel, the covariance matrix can be calculated, which has many applications in finance.

The data used in the paper includes the daily close prices of all the stocks from the S&P500 for a specific period. In other words, the data has the form $p \in \mathbb{R}^{N \times (D+1)}$. The return matrix $\mathbf{r} \in \mathbb{R}^{N \times D}$ was calculated from the data using $r_{n,d} = (p_{n,d} - p_{n,d-1})/p_{n,d-1}$. To show improvements resulted from the new approach, they compared the linear kernel used in the literature to three different non-linear kernels (exponential, squared exponential, and Matern 3/2). The first part of the experiments performed in the paper is related to model evaluation based on the two measurements: ELBO and R^2-score. The second part is related to the applications of the model in finance. They investigated three applications using the same data and the same initial setting.

In this section, we aim to replicate all results and graphs presented in the paper. First, we discuss general problems related to the replication of the study and then, we present our results and compare with those in the paper. Replicating results of the paper is challenging mainly because there is no clear information about the implementation (neither in the manuscript nor in the Github page). For example, although the run time is reported in the code scripts available on Github page, the properties of the hardware used for experiments are not mentioned. In addition, the results presented in the Github page of the paper is different from those given in the manuscript for some cases, making more challenges to evaluate the replicability of the experiments. For this reason, in the following sections we present both results in Github page and the manuscript where those are different.

All experiments are implemented on the instance of Google Cloud Platform (GCP) provided by the ICLR Reproducibility Challenge with the following specifications: 1 Tesla K80 GPU, 8 cores, 30GB RAM, 200 GB space, and pre-installed with NVIDIA Cuda 9.2 and Pytorch 0.4.1. In this environment, we import the package `ipyparallel` and initiate 32 engine clusters working in parallel to run the experiments with parallel computing. The number of engines is the same as what is specified in the code scripts on Github. Because the results can be affected by hardware used Piccolo & Frampton (2015), we also perform some of the experiments on another system with these properties: Mac OS, 4 processing cores (2.3 GHz Intel Core i5, up to 3.1 GHz), and 8 GB 2133 MHz LPDDR3 RAM. Same as what we do on the GCP instance, we initiate 32 parallel engines to run the experiments with parallel computing. We report results gained on both systems in these cases. Note

[3] In addition, as the instructors of the course had been informed, we had to change our first selected paper after working on it for one week due to its highly complicated algorithms, which made us face a time limitation issue as well.

that the results obtained form running experiments on the GPC instance are labeled "from GPC" and the results achieved on MacBook are labeled "from Mac".

Due to the problems mentioned in section 2, we have to conduct some of the experiments for a smaller input data than what has been done in the paper (i.e, considering N=60 instead of 120). Thus, we do not expect to reach the exact values reported by the paper in these cases. Our aim is to investigate whether the outperformance of the nonlinear models can be kept during replicability experiments or not. Moreover, we do not aim to replicate response time for two main reasons. First, the run times were not reported in the manuscript (i.e, the authors did not have any claim about the response time of their model). Second, because we do not know any thing about the underlying hardware, the replicability is not valid in this case. However, we report run times for all experiments in Appendix D.

3.1 Model Evaluation

As Figure 1 and 2 in Appendix C shows (see parts a, b, and c), we cannot regenerate the same results reported by the authors. Based on the paper, ELBO get a value in range 72000-75000, while both the Github and our results shows that the value of ELBO for difference kernels is in range 44500-46000. That is because the paper conduct experiments on the input data with N=120, while our experiments and Github report is based on N=80. However, the main pattern is the same in both replicated and paper-reported figures. For all Q in range 1-7, the non-linear kernels results in higher ELBO in comparison to the linear kernel, and by increasing the Q, differences between linear and non-linear kernels decreases.

When the model-evaluation measurement is R^2-score, the paper-reported values and replicated values are roughly close to each other. With respect to the fact that we perform experiments for a smaller dataset, and we do not know the exact configuration of the original experiments, it is reasonable to say that the experiments for the model evaluation with R^2-score are replicable.

3.2 Finance applications

3.2.1 Portfolio optimization

One of the main applications of this model in finance is portfolio optimization. As discussed in the paper, the optimal weight for stocks in the portfolio can be gained by $\boldsymbol{w}_{\text{opt}} = min(\boldsymbol{w}^T \boldsymbol{K} \boldsymbol{w})$ where K is the covariance matrix. The authors randomly selected 60 stocks from Jan 2008 to Jan 2018. For each six months, they calculated $\boldsymbol{w}$ based on the past year data by using different models. Then they reported the average return, standard deviation, and Sharpe ratio for different models. Tables 1 and 2 show original (from the paper's manuscript and Github page) and replicated results, respectively (results presented in Github page are depicted in Table 3 in Appendix A). Although we cannot regenerate the same values reported by authors, our results confirm this claim that non-linear kernels results in low std and high Sharpe ratio for the portfolio optimization problem. The minor differences between reported values are inevitable because of the random nature of experiments. We considered the possibility of doing significant test on the differences between models, but we needed to implement the experiment at least 30 times. With respect to the fact that even for the small size models ($N = 20$), each experiments takes at least 2 hours, and we have implementation issues, we give up. However, we strongly recommend the authors to present the statistical tests to show the significance of difference between models.

Table 1: Portfolio performance for different models on a yearly basis for $Q = 3$ and $N = 60$ from Paper.

Model	**Linear**	**SE**	**EXP**	**M32**	**Sample Cov**	**LedoitWolf**	**Eq. Weighted**
Mean	0.142	0.151	0.155	0.158	0.149	0.148	0.182
Std	0.158	0.156	0.154	0.153	0.159	0.159	0.232
Sharpe ratio	0.901	0.969	1.008	1.029	0.934	0.931	0.786

Table 2: Portfolio performance for different models on a yearly basis for $Q = 3$ and $N = 60$ from Mac.

Model	Linear	SE	EXP	M32	Sample Cov	LedoitWolf	Eq. Weighted
Mean	0.144	0.159	0.167	0.155	0.149	0.148	0.179
Std	0.162	0.158	0.157	0.156	0.159	0.160	0.232
Sharpe ratio	0.887	1.009	1.063	0.994	0.937	0.925	0.772

3.2.2 FILL IN MISSING VALUES

Regulation requires fair value assessment of all assets even those are illiquid or infrequently traded. The GP-LVM model presented in the study can provide an approach to predict the return for assets. The latent space $\boldsymbol{B}$ and hyperparameters are learned using the GP-LVM model, then the posterior return distribution can be easily calculated (recall: $\boldsymbol{r}_{n,:} = \boldsymbol{f}(\boldsymbol{B}_{n,:}) + \epsilon_n$). They considered N stocks for a particular d and fitted a GP to $N - 1$ stocks and predicted the value of the remaining stock. They implemented leave-one-out cross-validation and reported results based on the R^2-score and the average absolute deviation. They found that using nonlinear models results in more precise prediction, especially for Q between 2 and 4. As Figure 3 and 4 in Appendix C shows, although our results are compatible with those reported in the paper for the average absolute deviation, we could not replicate the results for some Q when R^2-score used as the evaluation measurement. In contrast to results presented in the paper, we found that nonlinear kernels outperform the linear kernels only for Q between 1 and 3. It is Interesting that for some values of Q, the linear kernel achieves more precise predictions, questioning the contribution of the paper. The reason behind this result can be that our experiments were implemented for N=70, while the results presented in the paper is for N=120.

3.2.3 INTERPRETATION OF THE LATENT SPACE

One of the advantages noted for the dominance of the proposed model over traditional models used in the literature is that it provides a tangible interpretation. For example, when stocks are depicted in the 2-D latent space, the distance between the stocks implicitly shows their correlation. In addition, this interpretation can be used to detect the structure in finance data. As the authors only presented one example for this part, we regenerate the same example in this section (see Figure 5 in Appendix C).

4 REPRODUCIBILITY

One important characteristic of strong research results is how flexible and robust they are in terms of changing the parameters and the tested environment known as reproducibility Villa & Zimmerman (2018). In this section, we aim to evaluate the reproducibility of the paper by carrying out some experiments on parameters.

Because of implementation issues mentioned in Section 2, we conduct most of the experiments with data size $N = 60$. In addition, to do more ablation studies on the parameters, we reduce the number of tries from 50 to 10 in some experiments (the effects of tries will be discussed more in following).

All the parameters and hyperparameters which can affect the model are: size of the input data (N and D), dimension of the latent space (Q), structure of the nonlinear kernels, parameters of the kernels (σ and l), coefficient scale, prior distribution of latent space, and optimization method. We could not find any discussion related to the optimization method used in the paper. This may be because they used exactly the framework presented by Michalis & Lawrence (2010). Therefore, because of the lack of information on optimization method, we do not consider it in our ablation study.

The results reported for each experiment in the paper is based on the best value gained through 50 runs started with random initialization. As mentioned before, we have to reduce the number of tries to cope with the implementation issues and to be able to do more ablation studies. For this reason, firstly, we evaluate the effects of tries on the results. As it is expected, under the small number of

tries (e.g., 5) the behavior of the models is unreliable, especially for nonlinear kernels. For example, as Figure 6 in Appendix C shows that the performance of the nonlinear kernels for $Q = 5$ is really low when the number of tries is 5. This fact shows that the robustness of results depends on the large number of tries. However, running a large number of tries takes a lot of time reducing the applicability of the proposed algorithm. In addition, it would be better if the authors briefly discuss why they report the best value rather than the average over different runs and what is the best range for the number of tries.

The dimension of input data is one of the critical parameters can significantly affect the results and the performance of models. As mentioned before, the input data is a matrix of size $N \times D$, where N depicts the number of stocks and D shows the number of days in the finance context. The Figure 1 and 2 in Appendix C depicts the effects of N on the model for different kernels based R^2-score and ELBO, respectively. When N is small (i.e., $N = 20$), the results are interesting in both ELBO and R^2-score measurements. By increasing Q, the ELBO decreases for all linear and non-linear kernels; however, the non-linear kernels are more robust to changes in Q than the linear kernel, which can be presented in the paper as an evidence for the outperformance of non-linear kernels. In contrast, R^2-score shows that when $N = 20$, linear kernels can perform as well as non-linear kernels for higher levels of Q (i.e. $Q = \{3, 4, 5, 6, 7\}$), which question the main contribution of the paper.

The number of latent dimensions Q plays a pivotal role in the performance of the GP-LVM algorithm. The paper studied the effects of $Q \in 1, 2, ..., 7$ on the model. We extend this part of the paper by evaluating the effects of larger values of $Q \in \{15, 20\}$. Figure 7 in Appendix C shows that by increasing Q, ELBO of all models (linear and non-linear kernels) decreases. However, ELBO of nonlinear kernels is still higher than linear kernels. In contrast, the R^2-score of the linear kennel increases for the large value of Q, and its difference from some of the nonlinear kernels becomes trivial. For this reason, we cannot generally determine how the performance of nonlinear kernels changes when the dimension of the latent space is high. It may need more extensive experiments.

As mentioned in Section 3, the study evaluated three stationary nonlinear kernels (exponential (exp), squared exponential (se), and Matern 3/2) (see Appendix B for their formulation) and compare them with the linear kernel. The diagonal elements of stationary kernels are the same, which is not appropriate for an estimation of a covariance matrix in finance Paper. For this reason, the authors defined the covariance matrix $\boldsymbol{K}_{cov} = \boldsymbol{\Sigma}\boldsymbol{K}_{corr}\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is a diagonal matrix with $\boldsymbol{\sigma}$ (vector of coefficient scales) and $\boldsymbol{K}_{corr}$ is the correlation matrix (i.e., the stationary kernel). The full kernel function at the end is the sum of the noise kernel and a $\boldsymbol{K}_{cov}$ as follow:

$$\mathbf{K}_{\text{se}} = \boldsymbol{\Sigma} k_{\text{se}}(\mathbf{B}, \mathbf{B})\boldsymbol{\Sigma} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \tag{1}$$

Where

$$k_{\text{noise}}(\beta_i, \beta_j) = \sigma^2_{\text{noise},i}\delta_{i,j} \tag{2}$$

$$\tag{3}$$

The above formulation shows squared exponential kernel while other kernels can be easily calculated by substituting $k_{se}(\boldsymbol{B}, \boldsymbol{B})$ with the desired kernel function (see Appendix B). The authors selected the following priors:

$$\boldsymbol{B} \sim \mathcal{N}(0, 1), \quad l, \sigma \sim \text{InvGamma}(3, 1), \quad \boldsymbol{\sigma}, \sigma_{\text{noise}} \sim \mathcal{N}(0, 0.5)$$

Length scale l describes the smoothness of a function. It also determines how far we can reliably extrapolate from the training data Rasmussen (2004). We change the distribution from which the kernel parameters (i.e., l, σ) are drawn from InvGamma(3,1) to InvGamma(3,2) (i.e., the mean and standard deviation of the distribution become twice). As it was expected, the outperformance of nonlinear kernels kept, especially based on the R^2-score (see Figure 8 in Appendix C). Because the results are based on the best value over 50 runs with random initialization, so they are robust under the small differences in mean and variance of the distribution functions of kernel parameters. However, when Q is 1, the performance of the nonlinear kernels decreases significantly.

Coefficient scales $\boldsymbol{\sigma}$ determines variance of function values from their mean Rasmussen (2004). If the coefficient scales are too large, the model will be free to chase outliers. The study randomly select the coefficient scale from a Gaussian distribution with mean 0 and standard deviation 0.5 (i.e., $\mathcal{N}(0, 0.5)$). We evaluate robustness of the results by changing the standard deviation to 1.5 (i.e., $\sigma \sim \mathcal{N}(0, 1.5)$). Under this change, the nonlinear kernels still perform better than linear kernel (see,

Figure 9 in Appendix C, notice that the high variation over Q in our results in comparison to those reported in the paper is mainly because of reducing the number of tries in our experiments).

Noise variance σ_{noise} specifies how much noise is expected to be present in the data Rasmussen (2004). Noise kernel is not formally a part of the covariance matrices. We remove it from nonlinear kernels and replicate the experiments for the model evaluation. Results show that noise kernel does not affects the outperformance of the nonlinear kernels significantly.

As mentioned before, the study considered three nonlinear kernels (exponential, squared exponential, and Matern 3/2). As a part of our ablation study, we investigate the performance of another nonlinear kernel (i.e., Matern 5/2, see its formulation in 6), and combination of linear and nonlinear kernels (both the sum and product of them) on the model as follows:

$$\mathbf{K}_{\text{se\_m\_linear}} = \big(\mathbf{\Sigma} k_{\text{se}}(\mathbf{B},\mathbf{B})\mathbf{\Sigma}\big) k_{\text{linear}} + k_{\text{noise}}(\mathbf{B},\mathbf{B})$$
$$\mathbf{K}_{\text{se\_p\_linear}} = \mathbf{\Sigma} k_{\text{se}}(\mathbf{B},\mathbf{B})\mathbf{\Sigma} + k_{\text{linear}} + k_{\text{noise}}(\mathbf{B},\mathbf{B})$$
$$\mathbf{K}_{\text{exp\_m\_linear}} = \big(\mathbf{\Sigma} k_{\text{exp}}(\mathbf{B},\mathbf{B})\mathbf{\Sigma}\big) k_{\text{linear}} + k_{\text{noise}}(\mathbf{B},\mathbf{B})$$
$$\mathbf{K}_{\text{exp\_p\_linear}} = \mathbf{\Sigma} k_{\text{exp}}(\mathbf{B},\mathbf{B})\mathbf{\Sigma} + k_{\text{linear}} + k_{\text{noise}}(\mathbf{B},\mathbf{B})$$

Figure 10 in Appendix C shows that the nonlinear kernel Matern 5/2 is also outperform the linear kernel, confirming the paper results about dominance of nonlinear kernels. Our results show that the combination of linear and nonlinear kernels (both sum and product of them) generally does not improve the performance of model in comparison to nonlinear kernels. Among all combinations we evaluated, the product of the linear kernel and the exponential kernel dominate other combinations based on the both ELBO and R^2-score. It is interesting that the sum of linear and nonlinear kernels reduces ELBO of the linear kernel, but meanwhile it increases the R^2-score.

Our experiments show that the main claim of the paper which is the outperformance of nonlinear kernels for estimating covariance matrix in comparison to the linear kernels are kept under different settings [4].

5 How the paper can improve

Although the replicability and reproducibility evaluations focus only on results gained through experiments and have nothing to do with the validity of the models and findings of the paper from theoretical aspects, we briefly discuss points may help authors to improve this study not only from reproducibility aspect but also from the theoretical points view. Some of these point briefly mentioned by reviewers in "OpenReview".

- To improve the reproducibility of the study, it is recommended that the authors provide readers with more precise information about the implemented experiments such as input data, used hardware, and properties of parallel implementation (e.g., number of clusters, details of the GP-LVM optimization).
- The study claims that using non-linear kernels result in a better portfolio allocation based on standard deviation (std) and Sharpe ratio. However, the authors conclude this result only based on some limited experiments. In addition, the significance of the differences between linear and non-linear kernels are not discussed. Therefore, extending experiments and reporting results in more precise way in section 4.3.1 can better verify their claim.
- As mentioned in the paper, estimating covariance matrices through machining learning methods have been previously studied in the literature. For example, Wilson & Ghahramani (2011) presented a method to model a diverse class of time-varying covariance matrices. By considering these comprehensive methods, the necessity of this study is not clear for readers. In addition, it is expected that the authors compare the performance of their method to other methods used in the literature for estimating covariance matrices to emphasize their contribution.
- The paper can improve if the authors provide some scientific evidences to support their experiments. For example, it is not clear why they used ELBO which is a lower bound to the marginal likelihood that can be considered as a model-evaluation measurement. In

[4] We may extend these experiments in the final version of the paper for the ICLR conferences 2019

addition, no explanation is presented in the paper about the selected non-linear kernels (i.e., why these kernels were selected). Generally, more explanation about results, especially Figure 3, may help to the apprehensibility of the paper.

- One of the weaknesses of the paper that can easily be addressed is providing references for some claims presented in the paper. For example, in Page 3, they mentioned that "ELBO gives the best approximation to the posterior" without any references for this claim.
- Finally, the paper can be improved and become more readable if the authors address some typo errors and notation inconsistency (see, the "OpenReview" page of the paper).

6 CONCLUSION

Reproducibility is a critical requirement for any computational studies including those based on machine learning techniques. In this project, we investigated the replicability and reproducibility of the Paper which is under review at ICLR 2019. They applied GP-LVM to extend the standard factor models for estimating the covariance matrices having many applications in finance. They showed that based on the R^2-score and the ELBO, non-linear kernels can results in more precise models than linear kernels.

We found out that most of the experiments done in the paper are not reproducible through sequential computation, specially for large scales of the problem. So, initiating a parallel computation is the first necessary step in reproducibility experiments of this paper. Despite the fact that we could not exactly replicate the experiments for the same size input data reported in the paper, our results showed the same behavior of the models for all experiments and graphs depicted in the paper. In addition, our experiments confirmed the robustness of the main contribution of the paper which is the outperformance of the nonlinear kernels in comparison to the linear kernels under changing the parameters and hyperparameters of the model. However, the paper can improve not only from reproducibility perspective, but also from theoretical points of view.

ACKNOWLEDGMENTS

We would like to thank the authors of the selected paper for their helpful instructions on how to initiate and run parallel computing which enabled us to run the replicability and reproducibility experiments. Furthermore, we would like to thank Koustuv Sinha for his efforts on setting up the GPC instance and his helps on solving the implementation issues we encountered during our project.

REFERENCES

Nicolas Chapados and Yoshua Bengio. Augmented functional time series representation and forecasting with gaussian processes. *Advances in Neural Information Processing Systems 20*, pp. 265–272, 2008.

J. B. Heaton, N. G. Polson, and J. H. Witte. Deep portfolio theory. *arXiv preprint 1605.07230, arXiv.org*, 2016.

Titsias Michalis and Neil D. Lawrence. Bayesian gaussian process latent variable model. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 844–851, 2010.

Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. *In Proceedings of the 23rd International Conference on Machine Learning*, pp. 673–680, 2006.

Reviewed Paper. Applications of gaussian processes in finance. *Under review as a conference paper at LCLR 2019*.

Stephen R. Piccolo and Michael B. Frampton. Tools and techniques for computational reproducibility. *GigaScience 5*, 1:30, 2015.

Hans E Plesser. Reproducibility vs. replicability: a brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11:76, 2018.

Carl Edward Rasmussen. Gaussian processes in machine learning. *In Advanced Lectures on Machine Learning*, pp. 63–71, 2004.

Jennifer Villa and Yoav Zimmerman. Reproducibility in machine learning: Why it matters and how to achieve it. *https://determined.ai/blog/reproducibility-in-ml/*, 2018.

Andrew Gordon Wilson and Zoubin Ghahramani. Generalised wishart processes. *arXiv preprint arXiv:1101.0240*, 2011.

Yue Wu, Jos Migue Hernndez-Lobato, and Zoubin Ghahramani. Gaussian process volatility model. *Advances in Neural Information Processing Systems*, pp. 1044–1052, 2014.

APPENDICES

APPENDIX A

Table 3: Portfolio performance for different models on a yearly basis for $Q = 3$ and $N = 60$ from Git.

Model	**Linear**	**SE**	**EXP**	**M32**	**Sample Cov**	**LedoitWolf**	**Eq. Weighted**
Mean	0.140	0.151	0.153	-	0.149	0.148	0.179
Std	0.162	0.158	0.158	-	0.159	0.160	0.232
Sharpe ratio	0.864	0.956	0.969	-	0.932	0.924	0.772

Table 4: Portfolio performance for different models on a yearly basis for $Q = 3$ and $N = 20$ from Mac.

Model	**Linear**	**SE**	**EXP**	**M32**	**Sample Cov**	**LedoitWolf**	**Eq. Weighted**
Mean	0.158	0.161	0.157	0.163	0.159	0.157	0.168
Std	0.189	0.186	0.186	0.186	0.188	0.188	0.231
Sharpe ratio	0.839	0.863	0.843	0.877	0.848	0.834	0.726

APPENDIX B

KERNEL FUNCTIONS

$$
\begin{aligned}
k_{\text{noise}}(\beta_i, \beta_j) &= \sigma^2_{\text{noise},i}\delta_{i,j} \\
k_{\text{linear}}(\beta_i, \beta_j) &= \sigma^2 \beta_i^T \beta_j
\end{aligned} \tag{4}
$$

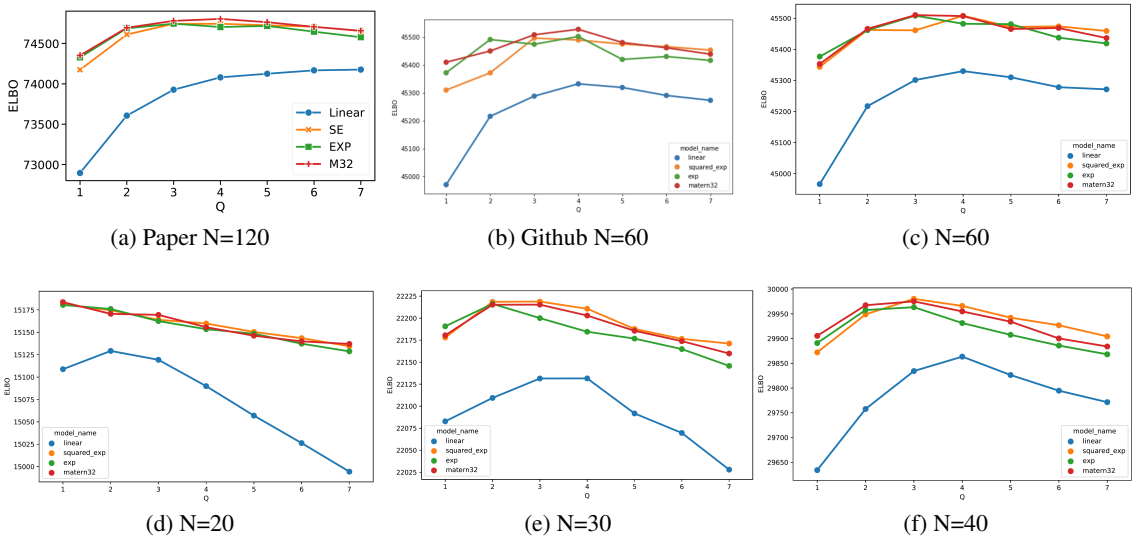
STATIONARY KERNELS

$$
\begin{aligned}
k_{\text{se}}(\beta_i, \beta_j) &= k_{\text{se}}(d_{ij}) = \exp\Big(-\frac{1}{2l^2}d_{ij}^2\Big) \\
k_{\text{exp}}(\beta_i, \beta_j) &= k_{\text{exp}}(d_{ij}) = \exp\Big(-\frac{1}{2l}d_{ij}^2\Big) \\
k_{\text{m32}}(\beta_i, \beta_j) &= k_{\text{m32}}(d_{ij}) = \left(1 + \frac{\sqrt{3}d_{ij}}{l}\right)\exp\left(-\frac{\sqrt{3}d_{ij}}{l}\right) \\
k_{\text{m52}}(\beta_i, \beta_j) &= k_{\text{m52}}(d_{ij}) = \left(1 + \frac{\sqrt{5}d_{ij}}{l} + 5\frac{d_{ij}^2}{3l^2}\right)\exp\left(-\frac{\sqrt{5}d_{ij}}{l}\right)
\end{aligned} \tag{5}
$$

MATRIX FORM

$$
\begin{aligned}
\mathbf{K}_{\text{linear}} &= k_{\text{linear}}(\mathbf{B}, \mathbf{B}) + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{se}} &= \mathbf{\Sigma} k_{\text{se}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{exp}} &= \mathbf{\Sigma} k_{\text{exp}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{m32}} &= \mathbf{\Sigma} k_{\text{m32}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{m52}} &= \mathbf{\Sigma} k_{\text{m52}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{se\_m\_linear}} &= \big(\mathbf{\Sigma} k_{\text{se}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma}\big) k_{\text{linear}} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{se\_p\_linear}} &= \mathbf{\Sigma} k_{\text{se}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{linear}} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{exp\_m\_linear}} &= \big(\mathbf{\Sigma} k_{\text{exp}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma}\big) k_{\text{linear}} + k_{\text{noise}}(\mathbf{B}, \mathbf{B}) \\
\mathbf{K}_{\text{exp\_p\_linear}} &= \mathbf{\Sigma} k_{\text{exp}}(\mathbf{B}, \mathbf{B})\mathbf{\Sigma} + k_{\text{linear}} + k_{\text{noise}}(\mathbf{B}, \mathbf{B})
\end{aligned} \tag{6}
$$

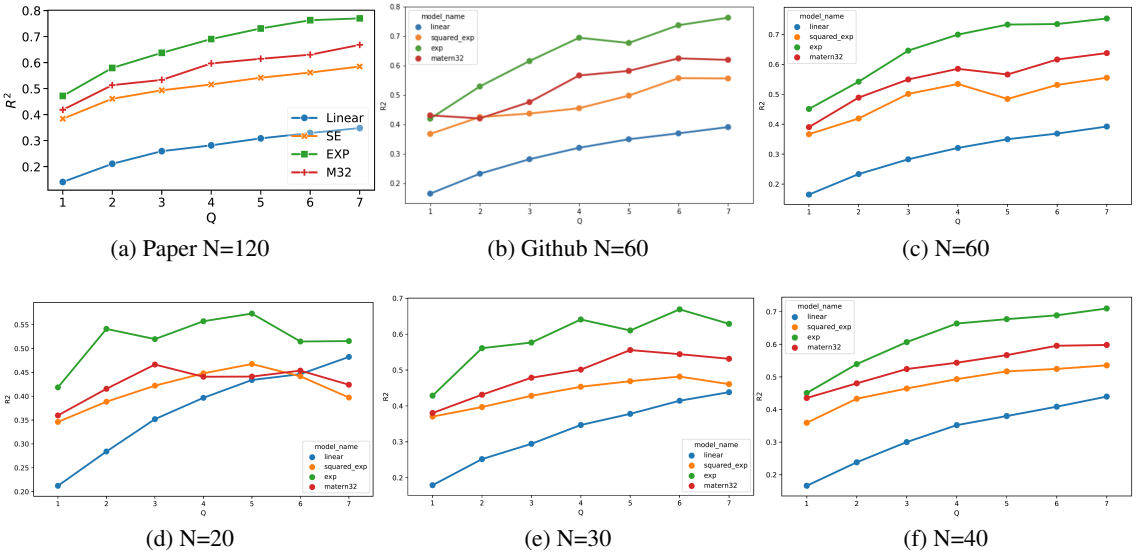
APPENDIX C



(a) Paper N=120 (b) Github N=60 (c) N=60

(d) N=20 (e) N=30 (f) N=40

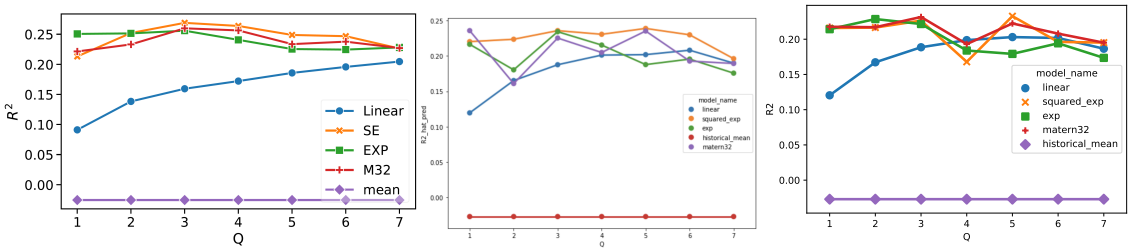
Figure 1: Comparing the effects of different input dimension (N) based on ELBO as a function of the latent dimension Q. (a): Results from the original paper; (b): Results from the code scripts available on Github; (c,d,e,f): Our results with different number of stocks N.



(a) Paper N=120 (b) Github N=60 (c) N=60

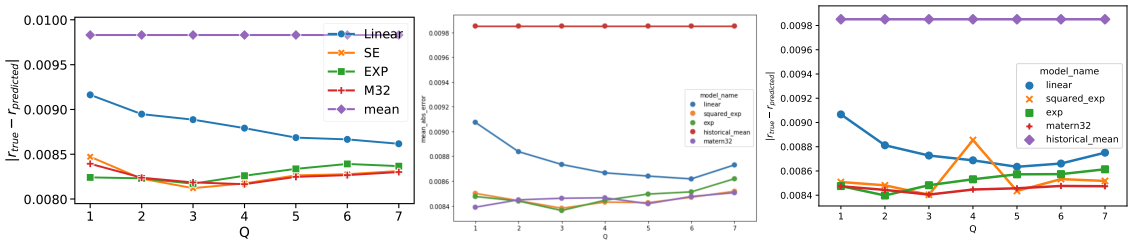
(d) N=20 (e) N=30 (f) N=40

Figure 2: Comparing the effects of different input dimension (N) based on R^2-score as a function of the latent dimension Q. (a): Results from the original paper; (b): Results from the code scripts available on Github; (c,d,e,f): Our results with different number of stocks N.



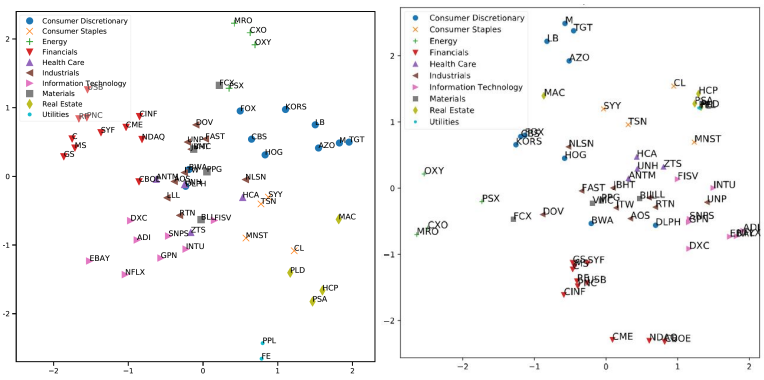
(a) Paper N=120 (b) Github N=70 (c) N=70

Figure 3: R^2-score of the predicted values as a function of the latent dimension Q. (a): Results from the original paper; (b): Results from the code scripts available on Github; (c): Our results.



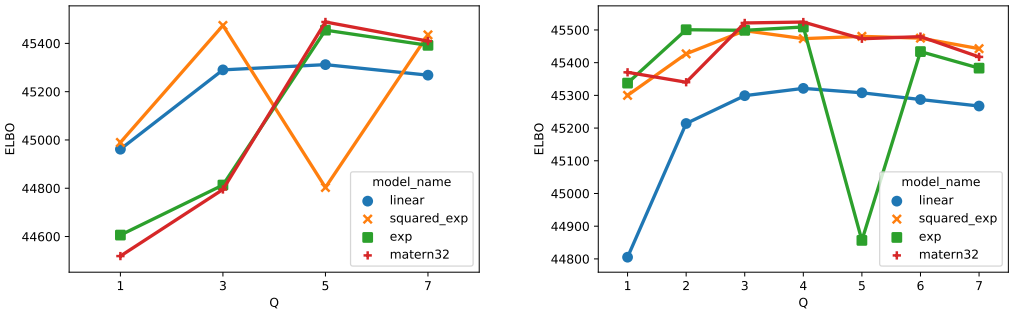
(a) Paper N=120 (b) Github N=70 (c) N=70

Figure 4: The average absolute deviation of suggested return to the real return evaluated by Leaving-one-out cross-validation. (a): Results from the original paper; (b): Results from the code scripts available on Github; (c): Our results.



(a) Paper N=60 (b) Mac N=70

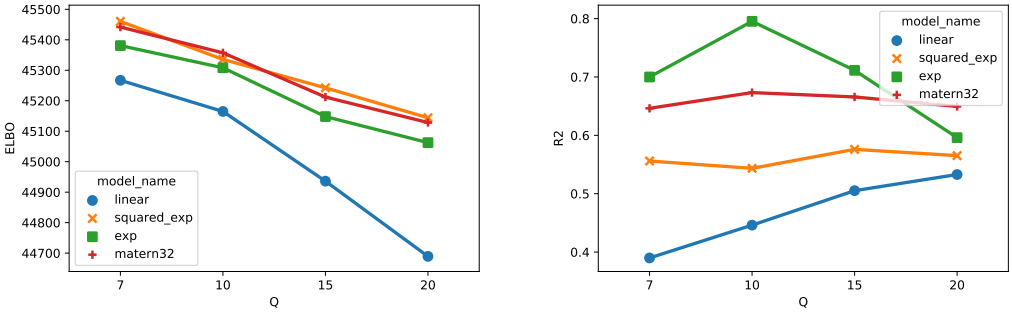
Figure 5: Stocks visualized in the 2-D latent space for the exponential kernel



(a) T=5 (b) T=10

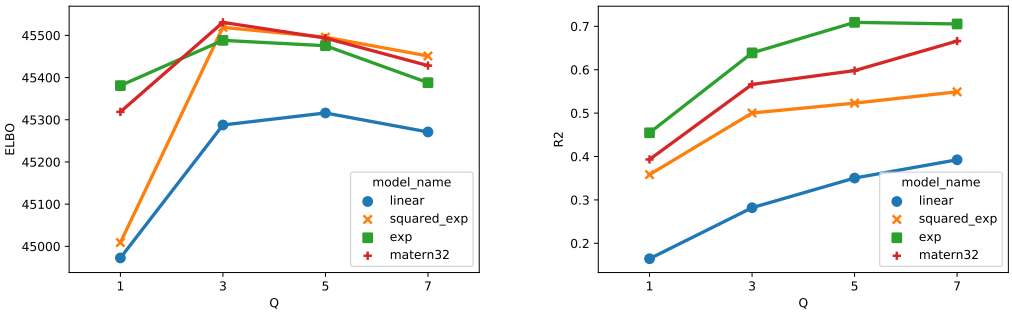
(c) T=15 (d) T=20

Figure 6: Comparing the effect of different number of tries (T) based on ELBO as a function of the latent dimension Q.



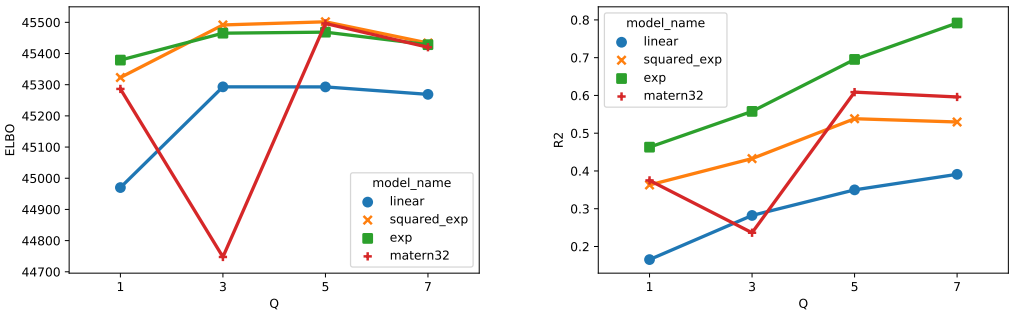
(a) ELBO (b) R^2-score

Figure 7: Comparing the effects of different latent dimension Q on the model for T=10 and N=60 based on ELBO and R^2-score.



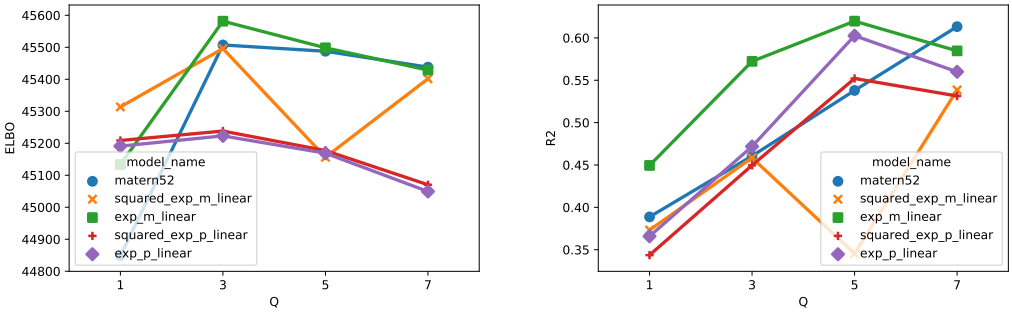
(a) ELBO (b) R^2-score

Figure 8: Comparing the effects of different distribution of kernel length scale on the model for T=10 and N=60 based on ELBO and R^2-score.



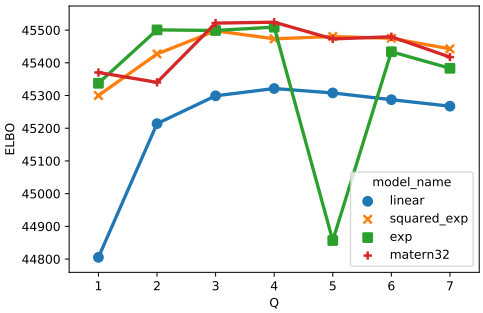
(a) ELBO (b) R^2-score

Figure 9: Comparing the effects of different coefficient scale on the model for T=10 and N=60 based on ELBO and R^2-score.

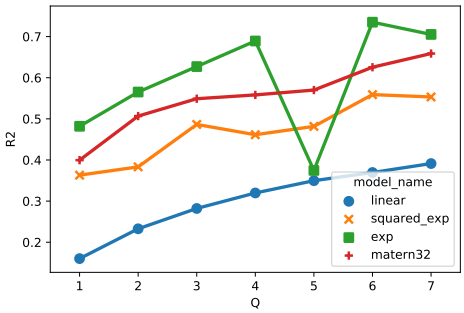


(a) ELBO

(b) R^2-score



(c) ELBO



(d) R^2-score

Figure 10: Comparing the effects of different nonlinear kernels on the model for T=10 and N=60 based on ELBO and R^2-score.

APPENDIX D

Type of experiment	System	Tries	N	Run-time (minutes)
Model Evaluation	GPC	5	60	39
Model Evaluation	GPC	10	60	76
Model Evaluation	GPC	15	60	113
Model Evaluation	GPC	20	60	155
Model Evaluation	GPC	15	40	79
Fill-in missing value	GPC	15	40	46
Fill-in missing value	GPC	15	70	187
Portfolio optimization	Mac	15	20	40
Portfolio optimization	Mac	15	60	296
Interpreting Latent Space	Mac	15	30	16
Interpreting Latent Space	Mac	15	60	55