

Deep Mixed Effect Model Using Gaussian Processes: A Personalized and Reliable Prediction for Healthcare

Ingwo Chung,¹ Saehoon Kim,² Juho Lee,² Kwang Joon Kim,³ Sung Ju Hwang,^{1,2} Eunho Yang^{1,2}
¹KAIST, ²AITRICS, ³Yonsei University College of Medicine, South Korea
 {jik0730, sjhwang82, eunhoy}@kaist.ac.kr, {shkim, juho}@aitrics.com, preppie@yuhs.ac

Abstract

We present a personalized and reliable prediction model for healthcare, which can provide individually tailored medical services such as diagnosis, disease treatment, and prevention. Our proposed framework targets at making personalized and reliable predictions from time-series data, such as Electronic Health Records (EHR), by modeling two complementary components: i) a shared component that captures global trend across diverse patients and ii) a patient-specific component that models idiosyncratic variability for each patient. To this end, we propose a composite model of a deep neural network to learn complex global trends from the large number of patients, and Gaussian Processes (GP) to probabilistically model individual time-series given relatively small number of visits per patient. We evaluate our model on diverse and heterogeneous tasks from EHR datasets and show practical advantages over standard time-series deep models such as pure Recurrent Neural Network (RNN).

Introduction

Precision medicine, which aims to provide *individually* tailored medical services such as diagnosis, disease treatment, and prevention, is an ultimate goal in healthcare. While rendered difficult in the past, nowadays it is becoming increasingly realizable due to the advances in data-driven approaches such as machine learning. Especially, recent widespread use of Electronic Health Record (EHR), a systematic collection of diverse clinical records of patients, has encouraged machine learning researchers to explore various clinical inferences (such as heart failure risk prediction (Choi et al. 2016b), sepsis prediction (Futoma, Hariharan, and Heller 2017), and physiological time-series analysis (Dürichen et al. 2015), to name a few) based on the records of personal medical history to improve the quality of clinical cares (Lipton et al. 2016; Lipton, Kale, and Wetzel 2016; Che et al. 2018).

As we have seen the recent huge success of deep learning, one of the most popular choices when working with EHR is to use Recurrent Neural Network (RNN) based models (Lipton et al. 2016; Choi et al. 2016b). However, this

kind of so-called “population based” models (that is, learning a single model for all patients as in RNN) might fail to give *personalized* predictions due to huge variability or heterogeneity among patients originated from diverse (possibly unobserved) sources such as intrinsic differences of patients due to demographical and biological factors, or other environmental factors (Marlin et al. 2012; Ng et al. 2015; Alaa et al. 2016). To demonstrate this issue, we illustrate in Figure 1 how heterogeneities across patients can impact the overall performances of population based models. In this toy simulation (see Appendix for details), we generate a bunch of patient-specific time-series $f^{(i)}(x)$ (i is patient index) with some globally shared structure and train the model with data from limited region (before dashed line). Figure 1 shows some *test* case. Here we can observe that for the region not used for training (after dashed line), teacher forced RNN tends to make biased predictions to the average of training patients. One possible and the easiest solution towards *personalization* is to model separate functions, one per patient to model the heterogeneity, but in a multitask framework to share the common knowledge across patients. However, deep models typically require large data, making it very challenging to train *separate* models for each patient.

Gaussian Process (GP) is another popular model for time-series as a non-parametric model, hence it might be the proper choice to separately model each patient. Due to its probabilistic nature, GP has additional benefit of representing uncertainty, which is also critical in medical problems for *reliable* prediction. However, when working with large amount of data, modeling exact GP gets computationally challenging since it requires to compute the inverse of covariance matrix across all data points ($O(n^3)$ in exact inference for n data points), although several approximations such as (Snelson and Ghahramani 2006; Titsias 2009) have been proposed at the expense of performance degradation (We will discuss related multi-task GPs and their computational issues in later sections). Some works such as (Clifton et al. 2013) have proposed to use separate GP for each time-series as an alternative, however these models cannot take advantages from global perspective as simply shown in Figure 1, Personalized GP.

In order to only take benefits from different approaches

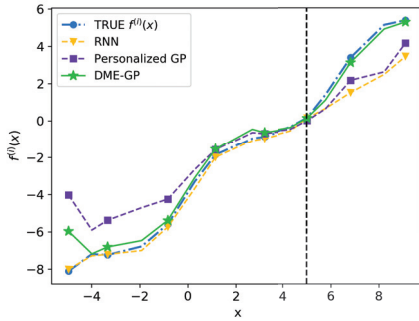


Figure 1: **Motivating experiment.** We generate data for a patient i by $f^{(i)}(x) = x + \sin(x) + \epsilon^{(i)}$ where $\epsilon^{(i)} \sim \mathcal{N}(\mu_i, \sigma_i^2)$ as deviation from others. RNN fails to provide personalized predictions on unseen input range (after dashed line) due to lack of personalization. Personalized GP dismisses the global trend. The behavior of other baselines are shown in Appendix though each has its disadvantage, compared to our model (denoted as DME-GP).

in multiple time-series, in this paper we propose a family of mixed effect model, Deep Mixed Effect Model using GPs (DME-GP), which combines a deep neural network and GPs. In DME-GP, by leveraging the complementary properties of two ingredients, we use a deep neural network (such as RNN) to capture the global trend from the large number of patients and GP to model idiosyncratic variability of individual patient. Regarding on the choice of the former, the global function should be representationally powerful enough to capture the complex shared trend across large number of patients. Not only that, it should be computationally amenable to handle large number of patients in training as well as inference procedures. Deep models including RNN architecture are the reasonable option to leverage the size of EHR data (in terms of the number of patients). On the other hand for the latter, we deliberately choose GP in order to provide reliable and individualized predictions even with very limited number of data points (i.e., limited number of patient’s visits). The use of GP naturally makes our model probabilistic and enables us to obtain the prediction uncertainty, which is another important property for mission-critical clinical tasks.

Our main contribution is threefold:

- We propose a frame of mixed effect model for multiple time-series that can leverage the individual benefits of shared global function and local function.
- We present the showcase of our framework for EHR analysis that uses RNN for capturing complex global trend in time-series and GP for personalized and reliable prediction with uncertainty estimate.
- We show practical advantages of our model over diverse baselines both on regression (from Physionet Challenge 2012 (Goldberger et al. 2000)) and classification tasks (from National Health Insurance Service; NHIS), and investigate its reliability, which is essential for mission-

critical tasks.

Related Work

Due to space constraint, here we focus to review works modeling EHR. More comprehensive review (e.g. review on combining GPs and deep models) is provided in Appendix.

Multiple Gaussian Processes for EHR Gaussian Process models have been actively used in the medical applications thanks to their reliability and versatility. However, using the separate formulation of multiple GPs is preferred due to its computational cost. (Clifton et al. 2013) proposed a multiple GPs formulation to handle missing values caused by sensor artifact or data incompleteness, which is common situation in wearable devices. (Peterson et al. 2017) proposed to use a similar model for diagnosis of Alzheimer’s disease, where a population-level GP is adapted to a new patient using domain GPs individually.

Multi-task Gaussian Process for EHR. The previous line of works can be understood as multi-task learning in the sense that the parameters of GPs across patients are shared. However, more systematic way of considering multi-task learning with GPs is to directly learn a shared covariance function over tasks. (Bonilla, Agakov, and Williams 2007; Bonilla, Chai, and Williams 2008) proposed Multi-task Gaussian Process (MTGP) that constructs large covariance matrix as a Kronecker product of input and task-specific covariance matrices for multi-task learning. A practical example of applying MTGP in medical situation is given in (Dürichen et al. 2015). (Cheng et al. 2017) proposed another approach that shares covariance matrix structured as the linear model of coregionalization (LMC) framework for personalized GPs, which is generalization of (Nguyen and Bonilla 2014). (Futoma, Hariharan, and Heller 2017; Futoma et al. 2017) made use of MTGP for preprocessing of input data fed into RNN. All of this line of works are based on the multi-task GPs that share huge covariance matrix which makes exact inference intractable. There have been some attempts to utilize mean of GP similar to our approach, proposed by (Schulam and Saria 2015), (Futoma et al. 2016), and (Iwata and Ghahramani 2017). However, our model is constructed in distinctive way where we use flexible deep models for shared mean function to capture complex structures, and more importantly, we explicitly construct a single GP for each patient to reflect individual signal.

Deep learning models with EHR. Recurrent neural networks (RNN) have recently been gained popularity as means of learning a prediction model on time-series clinical data such as EHR. (Lipton et al. 2016) and (Choi et al. 2016a) proposed to use RNN with Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Units (GRU) (Cho et al. 2014) respectively for multi-label classification of diagnosis codes given multivariate features from EHR. Moreover, the pattern of missingness, which is typical property of EHR, has been exploited in (Lipton, Kale, and Wetzel 2016) and (Che et al. 2018)

by introducing missing indicator and the concept of decaying informativeness. (Choi et al. 2016b) proposed to use RNN for generating attention on which feature and hospital visit the model should attend to, for building an interpretable model, and demonstrated it on heart failure prediction task. While RNN models have shown impressive performance on real-world clinical datasets, deploying them to safety-critical clinical tasks should be done with caution as they lack the notion of confidence, or uncertainty of prediction.

Proposed Method

Problem Formulation

Suppose dataset $\mathcal{D} := \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^P$ consists of P patients and i -th patient is represented by a sequence of T_i elements (or visits), that is, $\mathbf{X}_i := [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{T_i}^{(i)}]$, and corresponding target values, $\mathbf{y}_i := [y_1^{(i)}, y_2^{(i)}, \dots, y_{T_i}^{(i)}]$. The goal of our task in patient modeling is to predict the target value $y_t^{(i)}$ at each time step, given the current input features $\mathbf{x}_t^{(i)}$ and all the previous history: $\{\mathbf{x}_s^{(i)}\}_{s=1}^{t-1}$ and $\{y_s^{(i)}\}_{s=1}^{t-1}$.

This problem formulation incorporates several problems in EHR analysis such as disease progression modeling (DPM) or learning to diagnose (L2D) (Choi et al. 2016b). In DPM, we predict the evolutions of medical codes simultaneously at every time point. Specifically, if we have r different medical codes in our EHR, $\mathbf{x}_t \in \mathbb{R}^r$, which encodes the binary status indicating if each code appears in t -visit data, our goal is to predict \mathbf{x}_t at every time t given all the previous history $\{\mathbf{x}_s\}_{s=1}^{t-1}$. In L2D, which can be thought of as the special case of DPM, we are interested only in diagnosing of certain disease at the very end of visit sequence.

A Framework of Mixed Effect Model

Now we provide the general description of our mixed effect framework decomposing the function $f^{(i)}$ for i -th patient into two independent functions $g(\cdot)$ and $l^{(i)}(\cdot)$ under the multi-task learning paradigm:

$$f^{(i)}(\mathbf{x}_t) = g(\mathbf{x}_t) + l^{(i)}(\mathbf{x}_t) \quad (1)$$

where we assume $l^{(i)}(\cdot)$ to include random noise. Note that $g(\cdot)$ and $l^{(i)}(\cdot)$ are also called as fixed and random effect respectively in other literature (Greene 2003). In the framework, $g(\cdot)$ models global trend among the whole diverse patients, and hence it is shared across all patients. On the other hand, $l^{(i)}(\cdot)$ models the patient-specific signal (for i -th patient) that is not captured by the global trend $g(\cdot)$. Note that no information is shared across patients through $l^{(i)}(\cdot)$. We highlight that since the framework is generic, both functions can be chosen to be optimal depending on whatever the domain we apply on. Note also that in the traditional multi-task learning, we usually employ this kind of information sharing strategy at the *parameter* level; that is, the parameter vector for each task is represented as the sum of shared and individual parameters. However, in (1), the function value itself is mixed. Both approaches are equivalent only if $g(\cdot)$ and $l^{(i)}(\cdot)$ are linear mappings, which is not the case in general. The graphical representation of the framework (1) is shown in Figure 2a.

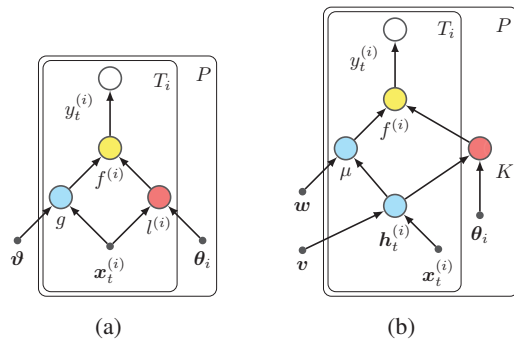


Figure 2: (a) A graphical representation of mixed effect framework for EHR analysis in (1). (b) A graphical representation of DME-GP in (4). To emphasize our decomposing framework, we color the global and individual components as blue and red respectively. Final composite model is colored as yellow.

Mixed Effect Model using GPs

As a concrete example of framework (1), we first consider the case where both $g(\cdot)$ and $l^{(i)}(\cdot)$ follow Gaussian Processes. Note that this formulation is just to relate our framework to existing multi-task GPs modeling each patient using a personalized GP. At the end of this subsection, it will be clear that this direction of individualization will involve almost intractable computations as the number of patients grows. Specifically, both components are represented as followings:

$$g(\mathbf{x}_t) \sim \mathcal{GP}(0, k_g(\mathbf{x}_t, \mathbf{x}_{t'})), \quad l^{(i)}(\mathbf{x}_t) \sim \mathcal{GP}(0, k^{(i)}(\mathbf{x}_t, \mathbf{x}_{t'}))$$

where we assume both GPs to have zero-mean for simplicity, and $k_g(\cdot, \cdot)$ and $k^{(i)}(\cdot, \cdot)$ are valid covariance functions such as squared exponential kernel (RBF). We name this instantiation ME-GP that stands for Mixed Effect Model using GPs. Note that in this model, knowledge sharing occurs via the covariance function $k_g(\cdot, \cdot)$ of global GP. Further assuming the independence between $g(\cdot)$ and $l^{(i)}(\cdot)$ for all patients, we can derive overall covariance function in the following manner:

$$\tilde{k}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)}) = k_g(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)}) + \delta_{ij} \cdot k^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)})$$

where δ_{ij} is the Kronecker delta function: $\delta_{ij} = 1$ if $i = j$ (that is, for same patient) otherwise 0. Interestingly, personalized GPs from this construction in fact boils down to a single GP with the covariance function $\tilde{k}(\cdot, \cdot)$ for all of function variables $\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(P)}$:

$$\begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(P)} \end{bmatrix} \sim \mathcal{GP} \left(\mathbf{0}, \begin{bmatrix} K_{11}^g + K^{(1)} & \cdots & K_{1P}^g \\ \vdots & \ddots & \vdots \\ K_{P1}^g & \cdots & K_{PP}^g + K^{(P)} \end{bmatrix} \right) \quad (2)$$

where $\mathbf{f}^{(i)} = f^{(i)}(\mathbf{X}_i)$ is a random vector of i -th patient process, and K_{ij}^g and $K^{(i)}$ are covariance matrices with elements given by $k_g(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)})$ and $k^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(i)})$ respectively at (t, t') position. As a result, the covariance matrix

in (2) lies in the space of $\mathcal{R}^{PT \times PT}$, assuming $T = T_i$ for all i . Given the model search and inference for a new point in GP rely on the inversion of covariance matrix, which costs $O(P^3T^3)$ for exact computation, learning with EHR datasets can be intractable in ME-GP even if we have small number of data points for each patient.

As noted earlier, Multi-task Gaussian Process (MTGP) is another model that uses GP in multi-task setting by forming covariance function with *multiplicative* task-relatedness parameters as follows :

$$\tilde{k}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)}) = K_{ij} \cdot k_g(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)})$$

where K_{ij} is an element at (i, j) position in task-relatedness matrix K as defined in (Bonilla, Chai, and Williams 2008). MTGP also requires to compute the inverse of huge covariance matrix in the same space as (2) and causes the same scalability issue.

Deep Mixed Effect Model using GPs

In this section, we propose Deep Mixed Effect Model using GPs (DME-GP) that exploits complementary properties of a deep network and GP and show our proposed model naturally overcomes scalability issue arisen in ME-GP. Specifically, we assume $g(\cdot)$ to be a deep network and $l^{(i)}$ to be GP as follows:

$$g(\mathbf{x}_t) = \mu(\mathbf{x}_t), \quad l^{(i)}(\mathbf{x}_t) \sim \mathcal{GP}(0, k^{(i)}(\mathbf{x}_t, \mathbf{x}_{t'}))$$

where $\mu(\cdot)$ is any kind of a deep neural network such as MLP or RNN where the knowledge sharing occurs across individual processes of patients. As we have done in the previous subsection, we can derive overall covariance function as follow:

$$\tilde{k}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)}) = \delta_{ij} \cdot k^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{x}_{t'}^{(j)}) .$$

Note that this covariance function naturally forms block-diagonal matrix where each corresponds to each patient's process $\mathbf{f}^{(i)}$:

$$\begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(P)} \end{bmatrix} \sim \mathcal{GP} \left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \vdots \\ \boldsymbol{\mu}_P \end{bmatrix}, \begin{bmatrix} K^{(1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & K^{(P)} \end{bmatrix} \right) \quad (3)$$

where $\boldsymbol{\mu}_i = \mu(\mathbf{X}_i)$ are outputs of a deep network. This in turn makes each patient process to be independent to other processes, which results in personalized GP models sharing global deep networks, described in (4). The computational cost of DME-GP compared to ME-GP reduces to $O(PT^3)$ thanks to its personalized formulation, which means DME-GP linearly scales to the number of patients P .

We also investigate complementary properties of DME-GP between global and individual components. As we discussed in the introduction, shared function $g(\cdot)$ and individual function $l^{(i)}(\cdot)$ have their own desired properties:

Individual Component We adopt a personalized Gaussian Process for $l^{(i)}(\cdot)$. This adoption allows the overall model to naturally provide the prediction uncertainty as a probabilistic model. In addition to that, GP enables us to reliably estimate individual signals based on relatively small number of data points (or visits for a patient) as a non-parametric model.

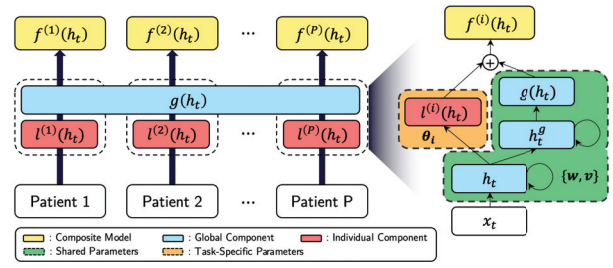


Figure 3: An overall conceptual illustration of DME-GP. Left panel describes personalized formulation for each patient. Right panel shows detailed descriptions of composed components for a single patient. Note that weights sharing of deep model occurs across all GPs and individual parameters are maintained for them as shown in green and orange boxes respectively.

Global Component We adopt representationally expressive deep models such as MLP or RNN for $g(\cdot)$. This is a reasonable choice to capture complex patterns in high dimensional medical data in relatively computationally amenable fashion using stochastic gradient descent algorithms such as Adam (Kingma and Ba 2015).

Composite Model Armed with these deliberate choices, it turns out the composite model (1) can be reduced to personalized GPs *sharing* a deep mean function, derived from (3):

$$f^{(i)}(\mathbf{x}_t) \sim \mathcal{GP} \left(\mu(\mathbf{h}_t | \mathbf{w}), k^{(i)}(\mathbf{h}_t, \mathbf{h}_{t'} | \boldsymbol{\theta}_i) \right) \quad (4)$$

where the shared deep function $g(\cdot)$ is renamed as $\mu(\cdot | \mathbf{w})$ (since it is a “mean function” of GP), $k^{(i)}(\cdot, \cdot | \boldsymbol{\theta}_i)$ is a kernel function for the individual process $l^{(i)}(\cdot)$, and \mathbf{h}_t is some embedding for input \mathbf{x}_t through global embedding function $\phi(\cdot | \mathbf{v})$. Here we adopt deep models as global embedding function, as done for global component. Note, this is a natural extension to benefit from deep kernel approach to make local kernel function more expressive (Wilson et al. 2016). The graphical representation of (4) is shown in Figure 2b although some parts of our model are deterministic mappings.

Design Choice The framework of (4) does not restrict $\mu(\cdot)$ and $\phi(\cdot)$ to have specific form. However, we focus on RNN to efficiently handle sequential nature of EHR. For instance of vanilla RNN with single hidden layer case, we have:

$$\mathbf{h}_t = \phi(\mathbf{x}_t, \mathbf{h}_{t-1} | \mathbf{v}) = \tanh(\mathbf{v}_{xh} \mathbf{x}_t + \mathbf{v}_{hh} \mathbf{h}_{t-1}) \quad (5)$$

where we suppress bias terms for simplicity and $\mathbf{v} = \{\mathbf{v}_{xh}, \mathbf{v}_{hh}\}$. $\mu(\cdot)$ can be formulated in a similar way. Note that the type of RNN cell can be any of choice, such as LSTM or GRU, and the architecture of a deep model can be carefully designed with domain knowledge of target dataset. Overall conceptual illustration of DME-GP is shown in Figure 3.

Learning and Inference of DME-GP

While our model is generally applicable to both regression and classification tasks, we implicitly assume the Gaussian

likelihood throughout the paper just for clarity and notational simplicity. These can be seamlessly extended for classification problems with binary likelihood along with standard approximation techniques as in regular Gaussian Process.

Learning Our learning objective is to maximize the marginal log-likelihoods of patients data \mathcal{D} under the modeling assumption of (3) and (4) to find global-level parameters $\{\mathbf{w}, \mathbf{v}\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_i\}_{i=1}^P$ from the individual components:

$$\boldsymbol{\theta}^*, \mathbf{w}^*, \mathbf{v}^* = \operatorname{argmax}_{\boldsymbol{\theta}, \mathbf{w}, \mathbf{v}} \sum_{i=1}^P \log p(\mathbf{y}_i | \mathbf{X}_i, \boldsymbol{\theta}, \mathbf{w}, \mathbf{v}) \quad (6)$$

where the log-likelihood is the sum of individual patient data under i.i.d. assumption across patients. An individual log-likelihood of single patient then can be represented using global and local parameters as follows:

$$\log p(\mathbf{y}_i | \mathbf{X}_i, \boldsymbol{\theta}_i, \mathbf{w}, \mathbf{v}) = -\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_i)^T K^{(i)-1} (\mathbf{y}_i - \boldsymbol{\mu}_i) - \frac{1}{2} \log |K^{(i)}| - \frac{T_i}{2} \log 2\pi \quad (7)$$

where the parameter dependencies are implicitly defined: $\boldsymbol{\mu}_i = [\mu(\mathbf{h}_1 | \mathbf{w}), \dots, \mu(\mathbf{h}_{T_i} | \mathbf{w})]^T$, $K^{(i)} \in \mathbb{R}^{T_i \times T_i}$ is a full covariance matrix given the element $k^{(i)}(\mathbf{h}_t, \mathbf{h}_{t'} | \boldsymbol{\theta}_i)$ at (t, t') position, and RNN-based embedding \mathbf{h}_t is a function on \mathbf{v} as mentioned in (5).

The gradient of (7) with respect to parameters can then be derived by chain rule as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{\theta}_i} &= \frac{\partial \mathcal{L}_i}{\partial K^{(i)}} \frac{\partial K^{(i)}}{\partial \boldsymbol{\theta}_i}, \quad \frac{\partial \mathcal{L}_i}{\partial \mathbf{w}} = \sum_{t=1}^{T_i} \frac{\partial \mathcal{L}_i}{\partial \mu_t} \frac{\partial \mu_t}{\partial \mathbf{w}} \\ \frac{\partial \mathcal{L}_i}{\partial \mathbf{v}} &= \frac{\partial \mathcal{L}_i}{\partial K^{(i)}} \sum_{t=1}^{T_i} \frac{\partial K^{(i)}}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{v}} + \sum_{t=1}^{T_i} \frac{\partial \mathcal{L}_i}{\partial \mu_t} \sum_{t'=1}^t \frac{\partial \mu_t}{\partial \mathbf{h}_{t'}} \frac{\partial \mathbf{h}_{t'}}{\partial \mathbf{v}} \end{aligned} \quad (8)$$

where $\mathcal{L}_i := \log p(\mathbf{y}_i | \mathbf{X}_i, \boldsymbol{\theta}_i, \mathbf{w}, \mathbf{v})$, and $\mu_t = \mu(\mathbf{h}_t)$. Note that, unlike vanilla RNN, the gradient computation of \mathbf{v} from (6) involves additional $\{K^{(i)}\}_{i=1}^P$ terms, leading to a bit more complicated computation. Note also that the gradient of global parameters \mathbf{w} and \mathbf{v} should involve the marginal likelihood across all patients while we only consider individual \mathcal{L}_i for clarity.

Our learning algorithm is based on stochastic gradient ascent in an alternating fashion and summarized in Appendix. Note again that our personalized formulation allows us to be able to avoid heavy computational cost from huge GP like in (2) with EHR datasets. In addition, deep architectures as a shared mean function can be updated efficiently through the standard back-propagation algorithm. Note also that in non-Gaussian likelihood cases such as classification tasks, the marginal likelihood can be computed via variational lower bound with variational approximation or by simulation approaches (Nickisch and Rasmussen 2008).

Inference for new patient j Since we have single GP for each patient in (4), our inference procedure for new patient

j follows the standard procedures of single GP inference. Suppose we want to predict y_t of a new patient j given current input feature \mathbf{x}_t and all historical data on this patient: $\mathbf{X} = \{\mathbf{x}_s\}_{s=1}^{t-1}$ and $\mathbf{y} = \{y_s\}_{s=1}^{t-1}$ where we suppress the patient index j for clarity. Then, we update the patient-specific parameters $\boldsymbol{\theta}_j$ of new GP by maximizing marginal log-likelihood (7), while global parameters $\{\mathbf{w}, \mathbf{v}\}$ are fixed.

The predictive distribution of y_t becomes $p(y_t | \mathbf{x}_t, \mathbf{X}, \mathbf{y}) = \mathcal{N}(y_t | \bar{y}_t, \sigma_t^2)$ with:

$$\begin{aligned} \bar{y}_t &= \mu(\mathbf{h}_t) + \mathbf{k}_t^T K^{-1} (\mathbf{y} - \mu(\mathbf{H})) \\ \sigma_t^2 &= k(\mathbf{h}_t, \mathbf{h}_t) - \mathbf{k}_t^T K^{-1} \mathbf{k}_t \end{aligned} \quad (9)$$

where $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_{t-1}]^T$ and $\mathbf{k}_t = k(\mathbf{H}, \mathbf{h}_t)$. The predictions can be done in sequential manner, which means we can predict the output at any time point of the patient. Note that the prediction at the first time point, $t = 1$, can be done deterministically by the global mean function, where the model predicts in average. As we increase the time point t , we have more evidence for the patient and make better personalized predictions.

We note that approximate predictions can also be derived with non-Gaussian likelihood in a classification problem. While following the notations from Gaussian likelihood case explained above, the output y follows some distribution $p(y | f(\mathbf{x}))$ that is properly defined according to \mathcal{Y} (i.e., normal distribution when $\mathcal{Y} := \mathbb{R}$ and Bernoulli distribution when $\mathcal{Y} := \{0, 1\}$). Then, the distribution of the latent function of GP for the test case \mathbf{x}_t is given by:

$$p(f_t | \mathbf{x}_t, \mathbf{X}, \mathbf{y}) = \int p(f_t | \mathbf{x}_t, \mathbf{X}, \mathbf{f}) p(\mathbf{f} | \mathbf{X}, \mathbf{y}) d\mathbf{f} \quad (10)$$

where $p(\mathbf{f} | \mathbf{X}, \mathbf{y}) = p(\mathbf{y} | \mathbf{f}) p(\mathbf{f} | \mathbf{X}) / p(\mathbf{y} | \mathbf{X})$ by Bayes' rule. Finally, the predictive distribution of y_t is:

$$p(y_t | \mathbf{x}_t, \mathbf{X}, \mathbf{y}) = \int p(y_t | f_t) p(f_t | \mathbf{x}_t, \mathbf{X}, \mathbf{y}) df_t \quad (11)$$

where $p(y_t | f_t)$ is a properly designed likelihood function of y_t given f_t according to the class of problems. In regression case, we have analytic forms for (10) and (11) when $p(y_t | f_t)$ follows Gaussian as we have shown in (9). On the other hand for classification problems, the likelihood function is designed to be a sigmoid function such as $\frac{1}{1 + \exp(-f_t)}$, which makes the integral in (10) and (11) analytically intractable. Thus, we need approximation methods for the posterior $p(\mathbf{f} | \mathbf{X}, \mathbf{y})$, such as Laplace approximation, variational method, or Markov Chain Monte Carlo (MCMC) approximation (Nickisch and Rasmussen 2008; Opper and Archambeau 2009).

Experiments

Dataset Description and Experimental Setup

Vital-Sign Dataset This dataset is compiled from a publicly available EHR dataset called Physionet Challenge 2012 (Goldberger et al. 2000). Specifically, we extract heart rate (HR) information in time-series for 865 patients who are in a cardiac surgery recovery unit (52,942 overall data points). Input (event time) and output values (heart rate) are scaled

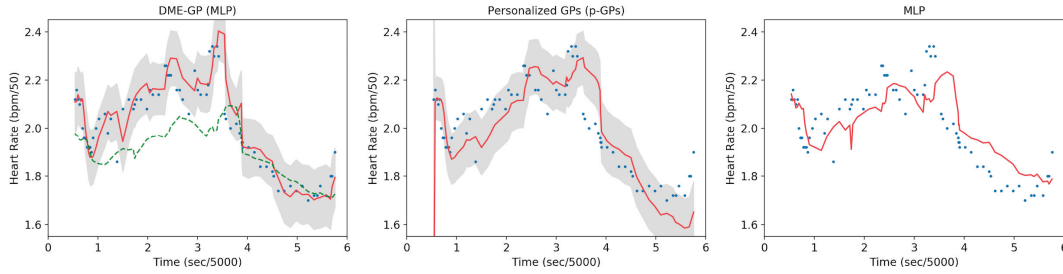


Figure 4: **Vital-Sign Analysis.** The predictions (red curves) for a random patient (blue dots) are shown in the order of DME-GP, p-GPs, and MLP respectively. The uncertainty representation is given by ± 1 standard deviation centered at the model’s predictions. The global trend of DME-GP, which is predicted by a global mean function, is shown as a green dashed line.

by 5,000 and 50, respectively. The dataset is motivated from the patient monitoring system in hospitals where the conventional procedures of the system are operated by nursing staff who frequently check target vital signs. It is important to automate the monitoring system to reduce the high cost of human labor and for early detection of those patients in a dangerous condition by predicting progressions of vital signs.

Medical Checkup Dataset This dataset is compiled from health checkup records for 32,927 patients (and 220,408 data points) collected from 2002 to 2013 (provided by National Health Insurance Service; NHIS). We select 12 common target diseases, and for each disease we have the health checkup history (either real or categorical input features) and corresponding target binary variables indicating either the absence or presence of a disease at each year. We convert categorical variables into one-hot vectors and normalize each real-valued feature with its mean and standard deviation. We simply fill in missing values in raw EHR data with zeros since missing rate is low (Lipton, Kale, and Wetzel 2016; Che et al. 2018). Detailed statistics of our data for each dataset are provided in Appendix.

Baselines We compare DME-GP against several baseline models including RNN-based deep models that are known to work well with time-series datasets:

- **Linear Models (LM):** A linear and logistic regression models for regression and classification respectively.
- **MLP:** A multi-layer perceptron containing two hidden layers with a sigmoid activation function.
- **RNN:** A RNN containing two hidden layers with long short-term memory units (LSTM) (Lipton et al. 2016).
- **RETAIN:** A RNN-based recurrent attention model proposed in (Choi et al. 2016b).
- **MTGP-RNN:** A multi-task Gaussian Process-wrapped RNN proposed in (Futoma, Hariharan, and Heller 2017). This model uses a multi-task GP (MTGP), but it is computationally tractable since it only considers the MTGP to correlate input features across different time points.
- **MAML:** A RNN-based Meta-SGD model proposed in (Li et al. 2017). This model is extended version of

MAML (Finn, Abbeel, and Levine 2017) where the model also learns step size of a meta-learner (an optimizer).

In case of LM and MLP, we treat individual time steps for all patients as i.i.d. observations since they are not specifically designed for time-series inputs. We exclude comparisons against variants of a single GP including MTGP and ME-GP not only because we observed their limited performances on the preliminary experiments but because they are computationally too expensive for exact inference. For our DME-GP, we consider two different models that use MLP and RNN with one hidden layer respectively. Note that we use a single-layered deep kernel function for our DME-GPs for fair comparisons (since baseline deep models use two layers in total). We defer all training details (e.g. setting hyper-parameters) to Appendix due to the space constraint.

Ablation Models We also evaluate the following variants of DME-GP for an ablation study:

- **p-GPs:** Personalized GPs with zero mean, individual embedding v_i and covariance θ_i for patient- i .
- **p-GPs-cov:** Personalized GPs with zero mean, shared embedding v and covariance θ .
- **p-GPs-both:** Personalized GPs with shared mean and covariance parameters w, v, θ .

We expect that p-GPs would not generalize well on a relatively small amount of patient data due to lack of sharing information. p-GPs-cov would benefit sharing information from a shared covariance function but in a limited way and lose individual characteristics. Lastly, p-GPs-both would fully benefit from both shared mean and covariance function, but not be able to capture individual signals because of missing local components. Interestingly, (Fortuin and Ratsch 2019) proposes a similar model as p-GPs-both concurrently to our work but in meta-learning perspective.

The code is available at <https://github.com/jik0730/Deep-Mixed-Effect-Model-using-Gaussian-Processes>.

Vital-Sign Analysis: Heart Rate

Our goal here is to find a mapping from a fixed window time-series $\{x_{t-2:t}, y_{t-2:t}\}$ to step-ahead target value y_{t+5} at every time stamp t .

In this experiment, we compare DME-GP (MLP) against p-GPs and MLP since other baselines perform similarly with

Table 1: **Disease Risk Prediction.** Performance (AUC) comparisons for 12 risk prediction tasks.

Diseases	DME-GP (RNN)	DME-GP (MLP)	RNN	RETAIN	MTGP-RNN	MLP	MAML	LM
Alcoholic Fatty Liver	0.829	0.801	0.791	0.796	0.785	0.777	0.780	0.529
Atherosclerosis	0.815	0.740	0.662	0.726	0.716	0.735	0.728	0.547
Emphysema	0.805	0.742	0.778	0.671	0.769	0.787	0.632	0.552
Liver Cirrhosis	0.932	0.922	0.888	0.871	0.904	0.856	0.913	0.635
Alcoholic Hepatitis	0.842	0.852	0.803	0.788	0.853	0.782	0.852	0.563
Arrhythmia	0.763	0.740	0.592	0.616	0.587	0.658	0.767	0.602
Fatty Liver	0.726	0.731	0.689	0.684	0.680	0.647	0.691	0.513
Heart Failure	0.829	0.759	0.790	0.792	0.761	0.783	0.729	0.620
Hepatic Failure	0.728	0.738	0.625	0.614	0.646	0.688	0.653	0.563
Hepatitis B	0.542	0.489	0.567	0.571	0.674	0.671	0.554	0.528
Myocardial Infarction	0.885	0.826	0.865	0.858	0.815	0.890	0.803	0.787
Toxic Liver Disease	0.641	0.698	0.595	0.594	0.596	0.643	0.685	0.518
Task Average	0.778	0.753	0.720	0.715	0.732	0.743	0.732	0.580

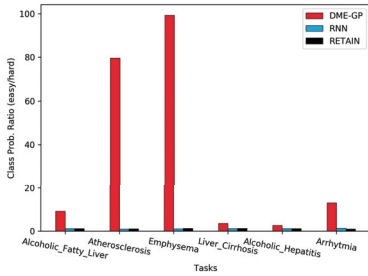


Figure 5: **Reliability Study.** Average classification probability ratio between easy patients and hard patients.

these two baselines. Running examples made by these models for a selected patient are shown in Figure 4. p-GPs shown in the middle graph tends to produce underestimated predictions where the model outputs lower values than expected, especially in initial time points. This phenomenon can be explained by its lack of global trend, and we can verify the benefit of knowledge transferring from other patients.

MLP on the right tends to behave like a *follower* where the predictions simply copy the former time-series targets since previous targets are the most useful information for population based models. On the other hand, DME-GP shows better predictions than the baselines. The predicted global trend (or mean) in DME-GP (shown in green dashed line) exhibits a similar pattern with the predictions of MLP and contributes to making the overall predictions better than p-GPs and MLP. This result partially implies that DME-GP is able to successfully benefit from both global and individual components. Overall test prediction performance (RMSE) for all patients is measured as 0.150 (DME-GP), 0.243 (p-GPs), and 0.194 (MLP), respectively.

Disease Risk Prediction by Medical Checkup

Given a visit sequence of input features $\{\mathbf{x}_t^{(i)}\}_{t=1}^{T_i}$ (which are representing clinical status) and corresponding binary targets $\{y_t^{(i)}\}_{t=1}^{T_i-1}$ (disease history) for each patient i , our task is to predict the most recent target $y_{T_i}^{(i)}$. The task can be thought of as predicting the risk of disease given time-series health checkup variables.

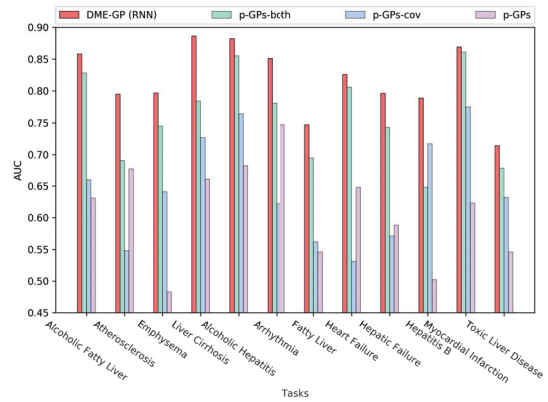


Figure 6: **Ablation Study.** Performance comparisons (in terms of Area Under the ROC Curve; AUC) among the variants of DME-GP. For fair comparison, we evaluate the models under the same hyper-parameters and measure the validation AUC.

We compare DME-GP against standard baselines listed above to verify the importance of considering the idiosyncratic variability of individual patient when modeling heterogeneous clinical data. As summarized in Table 1, DME-GP significantly outperforms others in most of the cases. The performance of DME-GP with a RNN mean function is the best among them probably because RNN is able to effectively capture the global trend by making use of historical data points in time-series. The population-based deep models such as RNN perform worse than DME-GP probably because they are not be able to effectively model individual differences across diverse patients. In particular, MTGP-RNN’s degraded performance compared to DME-GP suggests that modeling each *patient* as a single task is better than modeling each *feature* when modeling heterogeneous patient data. MAML also shows not enough prediction scores compared to DME-GP, which supports the claim that DME-GP is a better way to transfer knowledge in heterogeneous EHR datasets.

Ablation Study In this experiment, we evaluate the variants of DME-GP to investigate the effect of using differ-

Table 2: **Reliability Study.** Performance (in terms of Area Under the ROC Curve; AUC) comparisons when we exclude a set of *hard patient* who has a positive label at T_i but negative labels for others. Experiments for other diseases are shown in Appendix.

DISEASES	DME-GP	RNN	RETAIN
A. FATTY LIVER	0.998	0.856	0.861
ATHEROSCLEROSIS	1	0.683	0.768
EMPHYSEMA	0.991	0.785	0.765
LIVER CIRRHOSIS	0.989	0.928	0.922
ALCOHOLIC HEPATITIS	0.981	0.881	0.866
ARRHYTHMIA	0.992	0.639	0.656

ent levels of information sharing across patients. As a recap, our model only shares the global *mean function* across all patient-wise GPs to capture global trend in the data for knowledge transfer, and leverages patient-wise GPs to capture local variability from inherent hidden factors of each patient. The results shown in Figure 6 support our claim that the decomposition into a shared global part and a personalized local part is sensible with heterogeneous medical data.

Reliability Study Finally, in order to indirectly measure the predictive reliability of our model, we design a simple modification from the previous experiment on risk predictions. Specifically, we define a *hard patient* to denote a patient who has a positive label *only* at the prediction time T_i but never has positive labels in his/her historical data. We compare the differences (in terms of confidence as well as AUC) between i) the case where we exclude hard patients (easy) and ii) the case where we only consider such hard patients (hard). Figure 5 shows how confident the models are for two groups of patients and Table 2 summarizes AUC when we exclude hard patients. Our model exhibits clear distinctions between the two cases and achieves almost perfect scores for many datasets when it is confident. On the other hand, for the latter case (only on hard patients), the gain of using our model degrades as shown in Appendix, while being competitive with deep models. This reliable confidence information will allow proper involvement of human medical staff.

Conclusion

We have presented the framework of Mixed Effect Model for electronic health records (EHR) and provided Deep Mixed Effect Model using GPs (DME-GP) as a showcase example that exploits complementary properties of RNN and GP. In DME-GP, we use a deep network to learn a globally shared mean function capturing complex global patterns among diverse patients and use GP to build personalized and reliable prediction model. We have investigated the properties of our model for diverse tasks compiled from real EHR data and validated the superiority of it against state-of-the-art baselines. One last important note is that our model has an advantage to provide prediction uncertainty via GP in a principled way, which is essential for safety-critical clinical tasks.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants (No.2018R1A5A1059921, No.2019R1C1C1009192), Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (No.2016-0-00563, No.2017-0-01779, No.2019-0-01371) funded by the Korea government (MSIT) and Samsung Research Funding & Incubation Center via SRFC-IT1702-15.

References

- Alaa, A. M.; Yoon, J.; Hu, S.; and van der Schaar, M. 2016. Personalized risk scoring for critical care patients using mixtures of gaussian process experts. In *ICML Workshop on Computational Frameworks for Personalization*.
- Bonilla, E. V.; Agakov, F. V.; and Williams, C. K. I. 2007. Kernel multi-task learning using task-specific features. In *AISTATS*.
- Bonilla, E. V.; Chai, K. M. A.; and Williams, C. K. I. 2008. Multi-task Gaussian process prediction. In *NIPS*.
- Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports* 8(1):6085.
- Cheng, L.; Darnell, G.; Chivers, C.; Draugelis, M. E.; Li, K.; and Engelhardt, B. E. 2017. Sparse multi-output Gaussian processes for medical time series prediction. *arXiv preprint arXiv:1703.09112*.
- Cho, K.; Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.
- Choi, E.; Bahadori, M. T.; Stewart, W.; and Sun, J. 2016a. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare*.
- Choi, E.; Bahadori, M. T.; Sun, J.; Kulas, J.; Schuetz, A.; and Stewart, W. 2016b. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *NIPS*.
- Clifton, L.; Clifton, D. A.; Pimentel, M. A. F.; Watkinson, P. J.; and Tarassenko, L. 2013. Gaussian processes for personalized e-health monitoring with wearable sensors. *IEEE Transactions on Biomedical Engineering* 60(1).
- Dürichen, R.; Pimentel, M. A. F.; Clifton, L.; Schweikard, A.; and Clifton, D. A. 2015. Multitask Gaussian processes for multivariate physiological time-series analysis. *IEEE Transactions on Biomedical Engineering* 61(1).
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for fast adaptation of deep networks. In *ICML*.
- Fortuin, V., and Ratsch, G. 2019. Deep mean functions for meta-learning in Gaussian processes. *arXiv preprint arXiv:1901.08098*.
- Futoma, J.; Sendak, M.; Cameron, B.; and Heller, K. 2016. Predicting disease progression with a model for multivariate longitudinal clinical data. In *Machine Learning for Healthcare*.
- Futoma, J.; Hariharan, S.; Sendak, M.; Brajer, N.; Clement, M.; Bedoya, A.; O'Brien, C.; and Heller, K. 2017. An improved multi-output Gaussian process RNN with real-time validation for early sepsis detection. In *Machine Learning for Healthcare*.
- Futoma, J.; Hariharan, S.; and Heller, K. 2017. Learning to detect sepsis with a multitask Gaussian process RNN classifier. In *ICML*.
- Goldberger, A. L.; Amaral, A. N.; Glass, L.; Hausdorff, J. M.; Ivanov, P.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.; and Stanley, H. E. 2000. Physiobank, Physiokit, and Physionet:

Components of a new research resource for complex physiologic signals. *Circulation* 100(23):e215–e220.

Greene, W. 2003. *Econometric Analysis*. Pearson Education.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Iwata, T., and Ghahramani, Z. 2017. Improving output uncertainty estimation and generalization in deep learning via neural network Gaussian processes. In *arXiv preprint arXiv:1707.05922*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-SGD: Learning to learn quickly for few-shot learning. In *arXiv preprint arXiv:1707.09835*.

Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. 2016. Learning to diagnose with LSTM recurrent neural networks. In *ICLR*.

Lipton, Z. C.; Kale, D. C.; and Wetzel, R. 2016. Modeling missing data in clinical time series with RNNs. In *Machine Learning for Healthcare*.

Marlin, B. M.; Kale, D. C.; Khemani, R. G.; and Wetzel, R. C. 2012. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*.

Ng, K.; Sun, J.; Hu, J.; and Wang, F. 2015. Personalized predictive modeling and risk factor identification using patient similarity. In *AMIA*.

Nguyen, T. V., and Bonilla, E. V. 2014. Collaborative multi-output Gaussian Processes. In *UAI*.

Nickisch, H., and Rasmussen, C. E. 2008. Approximations for binary Gaussian process classification. *JMLR* 9.

Opper, M., and Archambeau, C. 2009. The variational Gaussian approximation revisited. *Neural Computation* 21(3):786–792.

Peterson, K.; Rudovic, O.; Guerrero, R.; and Picard, R. 2017. Personalized Gaussian processes for future prediction of alzheimer’s disease progression. In *NIPS Workshop on Machine Learning for Healthcare*.

Schulam, P., and Saria, S. 2015. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *NIPS*.

Snelson, E., and Ghahramani, Z. 2006. Sparse Gaussian processes using pseudo-inputs. In *NIPS*.

Titsias, M. K. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*.

Wilson, A. G.; Hu, Z.; Salakhutdinov, R.; and Xing, E. P. 2016. Deep kernel learning. In *AISTATS*.