Real-Time Transfer Active Learning for Functional Regression and Prediction Based on Multi-Output Gaussian Process

Zengchenghao Xia, Zhiyong Hu, Qingbo He, *Senior Member, IEEE*, and Chao Wang

***Abstract*—Active learning provides guidance for the design and modeling of systems with highly expensive sampling costs. However, existing active learning approaches suffer from cold-start concerns, where the performance is impaired due to the initial few experiments designed by active learning. In this paper, we propose using transfer learning to solve the cold-start problem of functional regression by leveraging knowledge from related and data-rich signals to achieve robust and superior performance, especially when only a few experiments are available in the signal of interest. More specifically, we construct a multi-output Gaussian process (MGP) to model the between-signal functional relationship. This MGP features unique innovations that distinguish the proposed transfer active learning from existing works: i) a specially designed covariance structure is proposed for characterizing within-and between-signal inter-relationships and facilitating interpretable transfer learning, and ii) an iterative Bayesian framework is proposed to update the parameters and prediction of the MGP in real-time, which significantly reduces the computational load and facilitates the iterative active learning. The inter-relationship captured by this novel MGP is then fed into active learning using the integrated mean-squared error (IMSE) as the objective. We provide theoretical justifications for this active learning mechanism, which demonstrates the objective (IMSE) is monotonically decreasing as we gather more data from the proposed transfer active learning. The real-time updating and the monotonically decreasing objective together provide both practical efficiency and theoretical guarantees for solving the cold-start concern in active learning. The proposed method is compared with benchmark methods through various numerical and real case studies, and the results demonstrate the superiority of the method, especially when limited experiments are available at the initial stage of design.**

Received 24 January 2024; revised 24 April 2024; accepted 25 August 2024. Date of publication 2 September 2024; date of current version 27 September 2024. This work was partially supported by the National Science Foundation (NSF) under Grant CMMI-2436025. The associate editor coordinating the review of this article and approving it for publication was Dr. Mohammed Nabil El Korso. *(Corresponding author: Chao Wang.)*

Zengchenghao Xia and Chao Wang are with the Department of Industrial and Systems Engineering, University of Iowa, Iowa City, IA 52242 USA (e-mail: chao-wang-2@uiowa.edu).

Zhiyong Hu is with the Department of Automation, Anhui University, Hefei 230601, China.

Qingbo He is with the State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai 200240, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2024.3451412, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3451412

***Index Terms*—Active learning, transfer learning, multi-output Gaussian process, Bayesian analysis, cold-start problem.**

I. Introduction

ACTIVE learning is a sub-field of machine learning that maximizes information acquisition to train models in a data-efficient way [1]. Different with passive learning such as Latin hypercude design and factorial design [2] that select or design the experiment settings before signal acquisition, active learning determines the most informative signal points or streams during the acquisition process. The basic idea of active learning is to iteratively select future data points by modeling and maximizing the information gain based on already collected data, where the newly selected data points will serve as the already collected data in the next round of active learning. The unique advantage of active learning is to expedite the efficiency of signal acquisition by concentrating on most informative data set, which significantly saves data collection costs. As a result, the active learning has been widely used in various applications where signal acquisition is timely and/or costly demanding, such as quality engineering, signal processing, and image recognition [3], [4], [5].

Nevertheless, existing active learning techniques suffer a common issued called cold-start [6], [7], which means the performance of active learning is impaired at early stage due to the small amount of collected data. This issue is widely observed in practice [8], [9] because it is accompanied with the intrinsic working principle of active learning: the selection of future data points is based on the modeling of already collected data. As a result, the less data on hand, the worse the modeling and active learning performance. In engineering practice, the cold-start problem can cause not only waste of data but also misinterpretation of learning results. For example, in one of our case studies of calibrating reduced graphene oxide (RGO) field-effect transistors (FET), a high-accuracy nano-sensor for detecting contamination in water, it is important to learn the current vs. voltage functional relationship so that the FET characteristics can be calibrated [10]. However, the current vs. voltage data can only be measured by disposable FET sensors, which requires to use as few sensors as possible to reduce cost during the learning process [11], [12]. We demonstrate the cold-start problem of this example in Fig. 1, where three collected sensor measurements are shown as hollow circles and the underlying

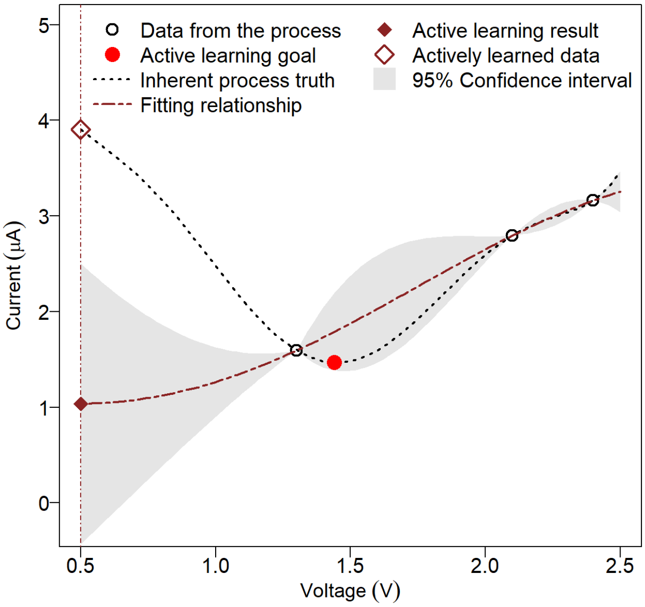


Fig. 1. Cold-start problem in active learning of functional relationship.

truth of the relationship (unknown in practice) is shown as dashed line. Each data point is the current reading (unit μA) from a disposable field-effect transistor (FET) sensor. The FET sensor can provide different current readings under different voltages, and it is important to find the voltage that generates the minimum current for sensor calibration purpose [13]. The active learning is thus implemented to find the minimum value in the current-voltage function, and the Gaussian process model is used to fit three available data points. The 95% confidence interval is constructed based on the predicted variance at each voltage value. The results of active learning guide the next experiment at the lowest value of predicted curve (solid diamond at left edge). It is clear in Fig. 1 that with only three data points, the active learning fails to locate the minimum point (solid red) and wastes the new sensor (hollow diamond). Moreover, the fitted curve (dash-dotted line) provides misleading relationship between current and voltage due to the lack of data. As a result, it is imperative to overcome the cold-start issue and achieve interpretable and efficient solution of active learning.

In literature, transfer learning has been proposed and applied for cold-start problem in active learning [14], [15]. The rationale is that transfer learning can extract shared information from different but related signals or processes (source domains) to “warm-up” and benefit the active learning in the process of interest (target domain). The transferred information from sources serves as virtually implemented experiments (data points) for the target signal. For example, the model can be first pre-trained by rich data in the source domain, which is then applied in the target domain for active learning [16] to achieve better performance than only using data from the target. The data in source and target can also be trained together to facilitate information transfer for active learning, where domain adaptation indices, e.g., maximum mean discrepancy and domain entropy, are used to formulate the joint objective function between sources and the target [17], [18]. There are also works formulating the transfer active learning as a weighted empirical risk, and the objective is to minimize the risk bound for both source and target domains [19], [20].

However, these existing transfer active learning techniques mainly focus on classification problems [21] and cannot be directly applied to deal with cold-start in regression problems, e.g., learning of functional relationship in Fig. 1. More specifically, there are two key limitations in existing methods. First, the learning task in existing transfer active learning focuses on adapting inputs, rather than input-output relationship, among different signals. In these methods, a transformation of inputs from N signals, i.e., $q(\mathbf{x}_1, \cdots, \mathbf{x}_N)$, is usually developed to facilitate the joint training with outputs in N signals, i.e., $\mathbf{y}_1, \cdots, \mathbf{y}_N$. In this case, the transfer learning is realized by modeling and sharing the same relationship between $q(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ and $\mathbf{y}_n$ for $n = 1, \cdots, N$. In other words, the specific input-output relationship between $\mathbf{x}_n$ and $\mathbf{y}_n$ in the nth signal, $n = 1, \cdots, N$, is not learned, and the signal-to-signal interactions among these functional relationships are not considered. On the other hand, the transfer learning of functional relationship focuses on the input-output representation of each signal, i.e., $q'(\boldsymbol{f}_1(\mathbf{x}_1), \cdots, \boldsymbol{f}_N(\mathbf{x}_N))$, where $\mathbf{y}_n = \boldsymbol{f}_n(\mathbf{x}_n) + \epsilon_n$. The $\boldsymbol{f}_n(\cdot)$ represents the relationship between $\mathbf{x}_n$ and $\mathbf{y}_n$ (with an additive noise ϵ_n), and the transformation $q'(\cdot)$ represents the interactions among N functions. The key difference between $q(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ and $q'(\boldsymbol{f}_1(\mathbf{x}_1), \cdots, \boldsymbol{f}_N(\mathbf{x}_N))$ is that $q(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ maps inputs from all signals to $\mathbf{y}_n$ thus the learning of input-output relationship in every individual signal becomes infeasible, while the $q'(\boldsymbol{f}_1(\mathbf{x}_1), \cdots, \boldsymbol{f}_N(\mathbf{x}_N))$ facilitates the learning of both individual input-output relationship and interactions among these relationships. Second, there lacks quantification of uncertainties during training. It is well documented that the uncertainty guides the implementation of active learning [22]. However, many existing transfer active learning methods, especially those used for classification, only involve the uncertainty in active learning stage and ignore it during transfer learning or training stage [23]. This is because the loss of classification problems can be formulated by an indicator function without any noise. In fact, there are two types of uncertainties to be quantified in transfer active learning of functional relationship [24]. The first type is the measurement/data uncertainty, which is represented by the ϵ_n for the nth process. The second type is modeling uncertainty, which represents the functional discrepancy between the estimated relationship model and the unknown underlying truth. The quantification of modeling uncertainty is especially critical for learning functional relationship because the underlying truth is usually highly nonlinear, and it is almost impossible to find a model that performs exactly the same as the underlying truth [25]. Unfortunately, in the field of transfer active learning, to the best of our knowledge, there are few methods that consider both data and modeling uncertainty.

To address the uncertainty quantification issue in active learning, researchers resort to the Gaussian process (GP), a non-parametric method, to characterize both data and modeling uncertainty in a single process [26], [27]. Recently, the multi-output Gaussian process (MGP) has also been proposed and applied in active learning to facilitate modeling of interactions among different signals [28], [29]. Nevertheless, these existing GP/MGP based methods deal with the above mentioned two issues separately thus cannot provide a comprehensive solution for cold-start problem in transfer active learning. More specifically, when dealing with the first issue, i.e., learning of functional relationships among signals, existing methods usually resort to a linear model of coregionalization (LMC) to characterize the interactions among different functional interactions [30]. However, the LMC poses a strong assumption that all functions should be linearly correlated. This significantly limits the flexibility and application scenarios of LMC [31], [32] since most of functional data in practice is non-linearly correlated. When dealing with the second issue, i.e., uncertainty quantification, existing methods usually treat the uncertainty in training and active learning in an independent manner [33], [34]. For example, the estimated/predicted uncertainty from training data needs to be re-calculated from scratch whenever a new data (identified by active learning) is added [35], [36]. This means the uncertainty before and after implementing the active learning is conditionally independent given the new data obtained by active learning. Such operation actually contradicts with the intrinsic philosophy of active learning because active learning seeks new data for iterative improvement over the information on hand rather than re-calculation of everything [22]. More importantly, almost all existing MGP methods for active learning suffer the expensive computational complexity, which scales cubically with the number of data points [37]. Such computational limitation not only poses concerns for real-time active learning (due to the re-calculation and expensive complexity in existing methods), but also restricts the application of MGP in transfer learning. This is because the transfer learning usually works in a context that there is much more data in sources than that in target, thus the computational/modeling tools must be efficient to deal with large amount of data in source domains to facilitate a successful transfer learning.

In this paper, we propose an efficient transfer active learning framework and a real-time MGP to address the above mentioned concerns and resolve the cold-start problem in active learning of functional relationship. In this framework, we first use convolution process (CP) to construct a tailored MGP structure for transfer learning of non-linear functional relationships. This structure features less computational complexity and better interpretability of modeling non-linear relationship among signals, which are superior than the commonly used LMC structure. Then, to facilitate the iterative prediction and uncertainty quantification of transfer active learning, an iterative Bayesian algorithm is developed to update parameters of the proposed MGP whenever new (batch of) data is available. The iterative updating plays a critical role in our transfer active learning framework because this marks the first time, to our knowledge, that it has revolutionized the traditional integration of MGP with active learning, which typically involves recalculating in each iteration. It also aligns the iterative updating of MGP with the iterative improvement of active learning. Moreover, the iterative updating further reduces the computational load since the updating only needs to take care of the newly incoming data instead of re-calculating all data accumulated on hand. Finally, the transferred and updated uncertainty in the target signal feeds into active learning by using the integrated mean-square error (IMSE) as the objective, and we provide theoretical justifications that the objective is monotonically decreasing as we get more data from the proposed transfer active learning framework. This theoretical property shows the performance of active learning will become better as more data is fed into the proposed transfer active learning framework, which provides mathematical guarantees for the iterative improvement of our framework. The major contributions of this work include:

- A novel MGP structure is proposed to facilitate transfer active learning of functional relationship, where two fundamental challenges, i.e., interactions and uncertainties among different functions, are resolved in a holistic framework.
- The critical concern of computational complexity of MGP is remarkably alleviated by the proposed iterative Bayesian estimation method, which not only expedites the real-time calculation but also accommodates the MGP to iterative active learning for the first time.
- We provide theoretical justifications for the objective and performance of the proposed transfer active learning framework.

Both numerical and case studies demonstrate the effectiveness of the proposed framework in terms of transfer learning accuracy, time efficiency, and active learning performance in comparison with various benchmark methods. The results show the proposed method is a superior approach for resolving the cold-start problem in learning functional relationship.

The rest of the paper is organized as follows. Section II provides a general formulation of the transfer active learning problem, which includes the formulation of MGP and the objective in active learning. In Section III, details about modeling, iterative updating, and transfer active learning will be presented, where the theoretical justifications of the selected objective will also be provided under the proposed framework. Section IV conducts numerical studies and comparisons to demonstrate the superiority of the proposed framework, where the modeling accuracy, time efficiency, and active learning performance are investigated. Two real case studies that suffer cold-start problems are demonstrated in Section V to further validate the effectiveness of the proposed framework. Finally, Section VI draws conclusion remarks and discusses future works.

II. Problem Formulation and Preliminaries

A. Problem Formulation of Transfer Active Learning

In this paper, we focus on solving the cold-start problem of active learning in a target signal by transferring data from readily available source signals in a real-time manner. Without loss of generality, we assume there are N signals and treat the Nth signal as the target signal. For the nth signal, the observations are taken at L_n input points $\mathbf{x}_n = [\mathrm{x}_{n1}, \mathrm{x}_{n2}, \cdots, \mathrm{x}_{nL_n}]^T$, where x_{nl} is the input for the lth observation in the nth signal, $n = 1, 2, \cdots, N, l = 1, 2, .., L_n$.

Accordingly, we define the data observations from the nth signal as $\mathbf{y}_n = [\mathrm{y}_n(\mathrm{x}_{n1}), \mathrm{y}_n(\mathrm{x}_{n2}), \cdots, \mathrm{y}_n(\mathrm{x}_{nL_n})]^T$, and the inputs and observations from all N signals are $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_N^T]^T$ and $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \cdots, \mathbf{y}_N^T]^T$, respectively. In our work, we have $L_N \ll L_n$ for $n = 1, \cdots, N-1$ to represent the cold-start problem in the target signal.

Based on the defined notations, the transfer active learning problem can be formulated as:

$$\mathbf{x}_N^* = \arg\min_{\boldsymbol{x}_N} g(\boldsymbol{x}_N | \mathbf{x}, \mathbf{y}) \tag{1}$$

where the goal is to identify a vector of D optimal input(s) of the next experiment in the Nth signal, i.e., $\mathbf{x}_N^* = [\mathrm{x}_{N1}^*, \cdots, \mathrm{x}_{ND}^*]^T, D \geq 1$, based on the available data in target and sources $(\mathbf{x}, \mathbf{y})$ and the acquisition function $g(\cdot)$. The $\boldsymbol{x}_N | \mathbf{x}, \mathbf{y}$ represents the predicted information of the Nth signal at input values $\boldsymbol{x}_N$. Note the difference between $\boldsymbol{x}_N$ and $\mathbf{x}_N$, where the $\mathbf{x}_N \in \mathbb{R}^{L_N}$ is the vector of inputs of available data in the Nth signal (part of $\mathbf{x}$) while the $\boldsymbol{x}_N \in \mathbb{R}^D$ represents D arbitrary inputs of the Nth signal for the acquisition function $g(\cdot)$. The learning results of Eq. 1 is to obtain D observations from the Nth signal, i.e., $\mathbf{y}_N^*$, at inputs values $\mathbf{x}_N^*$. Then, the $(\mathbf{x}_N^*, \mathbf{y}_N^*)$ will be incorporated into $(\mathbf{x}, \mathbf{y})$, and the updated $(\mathbf{x}, \mathbf{y})$ will be used in Eq. 1 to acquire more observation inputs from the Nth signal. Note the acquired number of new data points D might be different in each round of active learning. In this paper, we will use non-italic variables, e.g., $\mathbf{x}$ and $\mathrm{y}_n(\mathrm{x}_{nl})$, to represent data that is already determined or collected, while the italic variables, e.g., $\boldsymbol{x}_N$ and $\boldsymbol{y}_n$, represent arbitrary inputs and corresponding data variables to be optimized.

The formulation in Eq. 1 shows there are two key components for the success of transfer active learning. The first is to facilitate an efficient prediction of $\boldsymbol{x}_N | \mathbf{x}, \mathbf{y}$, which requires a comprehensive framework to transfer knowledge from $(\mathbf{x}, \mathbf{y})$ to the prediction of the Nth signal at arbitrary inputs $\boldsymbol{x}_N$. The second is the acquisition function $g(\cdot)$, which defines the active learning behavior and dominates the effectiveness of transfer active learning. In our work, we will develop a novel multi-output Gaussian process to expedite transfer learning and prediction of $\boldsymbol{x}_N | \mathbf{x}, \mathbf{y}$. We also demonstrate through theoretical analysis that the integrated mean of square error (IMSE) is an appropriate choice for $g(\cdot)$ to guarantee the accuracy of the transfer active learning in our framework.

We will briefly review the conventional MGP and the IMSE based active learning in Section II-B and Section II-C, respectively, and point out the limitations of existing methods for achieving a successful transfer active learning. To facilitate our proposed framework, we list some assumptions/clarifications as follows:

A1 Each signal is from stationary Gaussian process.

A2 Measurement noise is assumed to be independent and identically distributed.

A3 There are similarities among different signals for facilitating transfer learning. The similarities can be modeled by the kernel functions and their parameters.

C1 We allow different signals to have different numbers and values of inputs in the same input space. That is, x_{nl}, $n = 1, 2, \cdots, N, l = 1, 2, .., L_n$, can have different values, and the number of inputs L_n can be different across different n. But all x_{nl} should be from the same space, i.e., $\mathrm{x}_{nl} \in [a\ b]$, where a and b are real numbers that define the lower and upper bound of the input space.

It is worth noting the A1 and A2 are widely used for MGP related modeling and prediction [29], [38]. The A3 is also a necessary assumption for the success of transfer learning. The C1 demonstrates the adaptability of the proposed transfer active learning framework, highlighting its capability to accommodate the heterogeneity in the number of observations.

B. Multi-Output Gaussian Process

In this section, we introduce the popular multi-output Gaussian process model, which is used for modeling the within-and between-signal correlations. This model serves as the foundation for our proposed method.

The relationship between $\mathbf{x}_n$ and $\mathbf{y}_n$ in the nth signal can be represented by Gaussian process as:

$$\begin{aligned} &\mathrm{y}_n(\mathrm{x}_{nl}) = f_n(\mathrm{x}_{nl}) + \epsilon(\mathrm{x}_{nl}) \\ &f_n(\mathrm{x}_{nl}) \sim \mathcal{GP}\Big(\mu_n(\mathrm{x}_{nl}), \gamma_{nn}(\mathrm{x}_{nl}, \mathrm{x}_{nl'})\Big) \end{aligned} \tag{2}$$

where $f_n(\cdot)$ is the Gaussian process for modeling input-output relationship in the nth signal, the $\mu_n(\cdot)$ and $\gamma_{nn}(\cdot, \cdot)$ are mean function and covariance function for $f_n(\cdot)$, respectively, and $\epsilon(\mathrm{x}_{nl})$ is the measurement noise with independent and identically distributed (i.i.d.) Normal distribution $N(0, \sigma^2)$. Based on the formulation in Eq. 2, the data in the nth signal follows a L_n-dimension Normal distribution, i.e., $\mathbf{y}_n \sim MVN(\boldsymbol{\mu}_n, \gamma_{nn}(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 \mathbf{I}_{L_n})$, where $\boldsymbol{\mu}_n = [\mu_n(\mathrm{x}_{n1}), \cdots, \mu_n(\mathrm{x}_{nL_n})]^T$ is the mean vector, $\gamma_{nn}(\mathbf{x}_n, \mathbf{x}_n)$ is a L_n-by-L_n matrix that characterizes the data correlation within the nth signal, and the $\mathbf{I}_{L_n}$ is a L_n-by-L_n identity matrix. Accordingly, the multi-output Gaussian process models the data from N signals as a multivariate Normal distribution:

$$\begin{aligned} &\mathbf{y} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Omega}) \\ &\boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \cdots, \boldsymbol{\mu}_N^T]^T \\ &\boldsymbol{\Omega} = \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I}_L \\ &\quad = \begin{bmatrix} \gamma_{11}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \gamma_{1N}(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \gamma_{N1}(\mathbf{x}_N, \mathbf{x}_1) & \cdots & \gamma_{NN}(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} + \sigma^2 \mathbf{I}_L \end{aligned} \tag{3}$$

where $\gamma_{nn'}(\mathbf{x}_n, \mathbf{x}_{n'})$ is the covariance between the nth and n'th signal, $n, n' = 1, \cdots, N$, and $L = \sum_{n=1}^N L_n$. The $\gamma_{nn'}(\mathbf{x}_n, \mathbf{x}_{n'})$ characterizes the within-and between-signal correlation when $n = n'$ and $n \neq n'$, respectively, and it is a L_n-by-$L_{n'}$ matrix with the (l, l') component as $\gamma_{nn'}(\mathrm{x}_{nl}, \mathrm{x}_{n'l'}) = Cov(f_n(\mathrm{x}_{nl}), f_{n'}(\mathrm{x}_{n'l'}))$. We want to point out that the $\boldsymbol{\Gamma}(\cdot, \cdot)$ will be used as a general operator to represent covariance matrix based on its vector-to-vector inputs. For example, $\boldsymbol{\Gamma}(\mathbf{x}_n, \mathbf{x})$ is a L_n-by-L matrix that represents the covariance between data in the nth signal and data in all signals.

The unique feature of MGP is that it can provide a closed-form prediction for arbitrary signal at arbitrary input values:

$$\begin{aligned}&\boldsymbol{y}_n(\boldsymbol{x}_n)|\mathbf{x},\mathbf{y}\\&\sim N\Big(\boldsymbol{\mu}_n(\boldsymbol{x}_n)+\boldsymbol{\Gamma}(\boldsymbol{x}_n,\mathbf{x})\boldsymbol{\Omega}^{-1}(\mathbf{y}-\boldsymbol{\mu}),\\&\qquad\gamma_{nn}(\boldsymbol{x}_n,\boldsymbol{x}_n)-\boldsymbol{\Gamma}(\boldsymbol{x}_n,\mathbf{x})\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T(\boldsymbol{x}_n,\mathbf{x})+\sigma^2\mathbf{I}_D\Big)\end{aligned} \quad (4)$$

where $\boldsymbol{x}_n \in \mathbb{R}^D$ is a vector of inputs in the nth signal at arbitrary D locations.

Equation 4 provides a constructive option for using MGP to formulate the $\boldsymbol{x}_N|\mathbf{x},\mathbf{y}$ in Eq. 1. However, the direct use of Eq. 4 poses both methodological and practical concerns for transfer active learning. First, the modeling of relationship among N signals should be flexible and interpretable for transfer learning. Most of existing methods, however, used LMC to construct $\boldsymbol{\Omega}$ and represent within-and between-signal relationship, which limits modeling flexibility because the Kronecker product only facilitates linear correlation between signals [31], [32], [39]. Second, the calculation of $\boldsymbol{y}_n(\boldsymbol{x}_n)|\mathbf{x},\mathbf{y}$ in Eq. 4 requires the involvement of all data $\mathbf{y}$. This means the $\boldsymbol{y}_n(\boldsymbol{x}_n)|\mathbf{x},\mathbf{y}$ needs to be re-calculated from scratch whenever new data in the target signal is available. Such operation clearly raises concerns for the robustness of the parameter estimation and prediction, especially when the target has very small amount of data, where the over-fitting is often observed [37]. As a result, the restriction of re-calculating all data in Eq. 4 not only consumes additional computational resource but also conflicts with the iterative nature of active learning [40], which impairs the performance of transfer active learning. Finally, the computational complexity in Eq. 4 is unaffordable for transfer active learning. The calculation of Eq. 4 requires the inverse of a L-by-L matrix $\boldsymbol{\Omega}$ thus has computational complexity $\mathcal{O}(L^3)$, where $L=\sum_{n=1}^{N} L_n$. In transfer active learning, although L_N is usually small, the $L_1,\cdots,L_{N-1}$ are typically large to provide sufficient knowledge for transfer learning. Moreover, the inverse of large dimension matrix not only exhausts time and storage resources but also raises concerns about numerical issues, e.g., singularity during decomposition [41], which poses difficulties for training or parameter estimation. Although there are methods proposed for alleviating GP/MGP computational complexity, these methods are either offline (suffer the second concern), e.g., variational inference [42], or not feasible for capturing within-and between-signal correlation for transfer learning [43], [44]. To the best of our knowledge, there is no MGP framework designed specifically for real-time transfer active learning.

C. IMSE Based Active Learning

In literature, there are two categories of active learning strategies, i.e., active learning cohn (ALC) and active learning mackay (ALM) [45], [46], where the ALC employs the integrated mean squared error (IMSE) to acquire information while the ALM uses variance based criterion to determine informative data points. The key difference between the IMSE and variance based criterion is that the IMSE criterion considers the uncertainty across the entire design/learning space (by using integration) while the variance based criterion only focuses on uncertainty at specific locations in the space. As a result, although the IMSE is constructed in a more complicated way than variance based criterion, it is well documented that the IMSE (and its ALC) can achieve more efficient and robust learning performance [47], [48], [49].

Conventional IMSE based active learning is for a single signal, e.g., the Nth signal, and it is formulated based on the special case of Eq. 4 [27], i.e., $\mathbf{x}=\mathbf{x}_N$ and $\mathbf{y}=\mathbf{y}_N$:

$$\begin{aligned}\mathbf{x}_N^* &= \arg\min_{\boldsymbol{x}_N} g(\boldsymbol{x}_N|\mathbf{x}_N,\mathbf{y}_N)\\ g(\boldsymbol{x}_N|\mathbf{x}_N,\mathbf{y}_N) &= \int \gamma_{NN}(x,x)-\boldsymbol{\Gamma}(x,[\boldsymbol{x}_N,\mathbf{x}_N])\\&\quad\cdot\big(\boldsymbol{\Gamma}([\boldsymbol{x}_N,\mathbf{x}_N],[\boldsymbol{x}_N,\mathbf{x}_N])\\&\quad+\sigma^2\mathbf{I}_{(L_N+D)}\big)^{-1}\boldsymbol{\Gamma}^T(x,[\boldsymbol{x}_N,\mathbf{x}_N])\mathrm{d}x\end{aligned} \quad (5)$$

where x is an arbitrary input value in the Nth signal. Note the x will be integrated out, and the optimization variable is $\boldsymbol{x}_N$.

Although the IMSE based active learning is widely used in practice, it still suffers the cold-start problem [8], [9]. An intuitive solution to this issue is to plug Eq. 4 to Eq. 5 so that data in $(\mathbf{x}_1,\mathbf{y}_1),\cdots,(\mathbf{x}_{N-1},\mathbf{y}_{N-1})$ can be transferred to the Nth signal. In this way, the IMSE function $g(\cdot)$ will apply to $\boldsymbol{x}_N|\mathbf{x},\mathbf{y}$. However, as we mentioned in Section II-B, the Eq. 4 cannot be applied directly due to its inappropriate formulation for active learning. It is also reported that an inappropriate transfer learning framework may even result in worse modeling and prediction performance [50], [51], which can aggravate the cold-start problem in active learning. As a result, theoretical investigations of impacts of the developed transfer learning framework on the active learning is desired for demonstrating the effectiveness of transfer learning.

III. Proposed Framework

A. Transfer Learning Framework Based on Tailored Multi-Output Gaussian Process

As we mentioned in Section II-B, the structure of $\boldsymbol{\Gamma}$ dominates the within-and between-signal correlation, thus it is critical to construct an intepretable and efficient $\boldsymbol{\Gamma}$ for transfer active learning. In this section, we propose to construct such $\boldsymbol{\Gamma}$ by convolution process (CP), which is a commonly used method to construct covariance for uni-variate GP [52]. In CP, the Gaussian process for the nth signal can be formulated as the convolution between Gaussian white noise processes and kernels [52]:

$$\begin{aligned}&f_n(\mathrm{x}_{nl})=\sum_i k_{in}(\mathrm{x}_{nl})*\psi_i(\mathrm{x}_{nl})\\&Cov(f_n(\mathrm{x}_{nl}),f_{n'}(\mathrm{x}_{n'l'}))\\&=\sum_i\int_{-\infty}^{+\infty}k_{in}(\mathrm{x}_{nl}-u)k_{in'}(\mathrm{x}_{n'l'}-u)du\end{aligned} \quad (6)$$

where $\psi_i(\cdot)$ and $k_{in}(\cdot)$ are the ith Gaussian white noise process and the corresponding kernel contributing to the nth signal, and $*$ is the convolution operator.

Based on the formulation in Eq. 6, we propose a specially designed MGP structure in Fig. 2. In this structure, N different

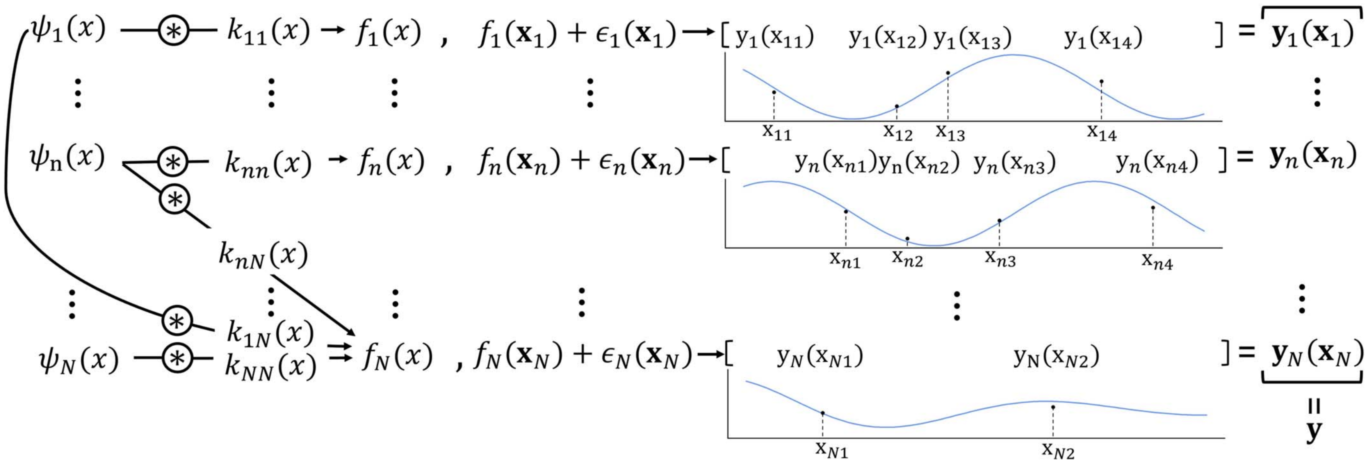

$\psi_1(x)$
$k_{11}(x) \rightarrow f_1(x)$,
$f_1(\mathbf{x}_1) + \epsilon_1(\mathbf{x}_1)$
$[\, y_1(x_{11}) \quad y_1(x_{12})\, y_1(x_{13}) \quad y_1(x_{14}) \,] = \mathbf{y}_1(\mathbf{x}_1)$
x_{11} x_{12} x_{13} x_{14}
$\psi_n(x)$
$k_{nn}(x) \rightarrow f_n(x)$,
$f_n(\mathbf{x}_n) + \epsilon_n(\mathbf{x}_n)$
$[\, y_n(x_{n1}) y_n(x_{n2}) \quad y_n(x_{n3}) \quad y_n(x_{n4}) \,] = \mathbf{y}_n(\mathbf{x}_n)$
x_{n1} x_{n2} x_{n3} x_{n4}
$k_{nN}(x)$
$k_{1N}(x)$
$\psi_N(x)$
$k_{NN}(x) \rightarrow f_N(x)$,
$f_N(\mathbf{x}_N) + \epsilon_N(\mathbf{x}_N)$
$[\, y_N(x_{N1}) \quad y_N(x_{N2}) \,] = \mathbf{y}_N(\mathbf{x}_N)$
x_{N1} x_{N2}
$\mathbf{y}$

Fig. 2. The proposed transfer learning structure.

Gaussian processes, i.e., $f_1(x), \cdots, f_N(x)$ are all constructed by CP:

$$f_n(\mathrm{x}_{nl}) = \begin{cases} \sum_{i=1}^{N} k_{in}(\mathrm{x}_{nl}) * \psi_i(\mathrm{x}_{nl}) & n = N \\ k_{nn}(\mathrm{x}_{nl}) * \psi_n(\mathrm{x}_{nl}) & n \neq N \end{cases} \tag{7}$$

where the Nth signal (target signal) consists of N different CPs, and other $N-1$ signals are constructed by their corresponding CP. Such structure facilitates a unique correlation relationship within and between each signal, which can be represented by the covariance of data in N signals:

$$\begin{aligned} Cov(\mathbf{y}, \mathbf{y}) &= \mathbf{\Omega} = \mathbf{\Gamma}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I}_L \\ &= \begin{bmatrix} \gamma_{11} & \mathbf{0} & \cdots & \mathbf{0} & \gamma_{1N} \\ \mathbf{0} & \gamma_{22} & \cdots & \mathbf{0} & \gamma_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \gamma_{(N-1)(N-1)} & \gamma_{(N-1)N} \\ \gamma_{N1} & \gamma_{N2} & \cdots & \gamma_{N(N-1)} & \gamma_{NN} \end{bmatrix} + \sigma^2 \mathbf{I}_L \end{aligned} \tag{8}$$

where $\gamma_{nn'}$ is an abbreviation for $\gamma_{nn'}(\mathbf{x}_n, \mathbf{x}_{n'})$, i.e., the block covariance matrix between nth and n'th signals, $n, n' = 1, \cdots, N$. The block matrices in the diagonal represents the within-signal correlation, and the non-diagonal block matrices represent the correlation between each source signal and the target signal. More specifically, the proposed structure in Fig. 2 enjoys the following features:

First, it provides an interpretable structure for transfer learning, where the information in source signals is shared with the target signal through kennels $k_{nN}, n = 1, \cdots, N-1$. Such information sharing is further quantified by the non-diagonal block covariance matrix in Eq. 8, i.e., γ_{nN}. We denote $\boldsymbol{\eta}$ as the vector of all unknown parameters associated with the kernel function and the measurement noise σ^2 for constructing Eq. 8 from Fig. 2. In this work, we will use Gaussian kernels, and the closed-form of Eq. 8 are provided in Section A of supplementary materials. Second, the computational complexity of the constructed MGP reduces significantly due to the sparse covariance matrix in Eq. 8. Specifically, the computational complexity of inversing the $\mathbf{\Omega}$ reduces from $\mathcal{O}((\sum_{n=1}^{N} L_n)^3)$ to $\mathcal{O}(\sum_{n=1}^{N} L_n^3)$. Such reduction makes the complexity linear to the numbers of signals (N), which facilitates the incorporation of large number of signals into transfer learning. Finally, the CP structure in Fig. 2 explains its more superior modeling flexibility than the widely used LMC. We use the construction of the target signal under two-signal case, i.e., $N = 2$, as an example to illustrate:

$$\begin{aligned} f_2^{CP}(\mathrm{x}_{2l}) &= k_{12}(\mathrm{x}_{2l}) * \psi_1(\mathrm{x}_{2l}) + k_{22}(\mathrm{x}_{2l}) * \psi_2(\mathrm{x}_{2l}) \\ f_2^{LMC}(\mathrm{x}_{2l}) &= a_{12}\psi_1(\mathrm{x}_{2l}) + a_{22}\psi_2(\mathrm{x}_{2l}) \end{aligned} \tag{9}$$

where the $f_2^{CP}(\mathrm{x}_{2l})$ is the CP-constructed target signal (based on Eq. 7 when $N = 2$), and the $f_2^{LMC}(\mathrm{x}_{2l})$ is the LMC-constructed target signal (linear combination of Gaussian white noise process). It is clear in Eq. 9 that the coefficients a in LMC apply to the whole functional space of ψ, i.e., the a is the same for different x in ψ, while the kernels k in CP serve as changing coefficients at different x. In other words, by carefully setting the kernels in CP, the LMC is a special case of CP by fixing the k as constant values. This feature makes CP-constructed MGP more flexible in modeling the within-and between-signal relationship.

It is worth noting that the zeros posed in Eq. 8 will cause some information loss due to the ignorance of interactions among source signals. However, the performance of the proposed method will not be influenced significantly by this information loss. The key reason is because we only care about active learning of the target signal, rather than all signals. Although the zeros posed in the covariance matrix will influence the modeling of the interactions among the $N-1$ source signals, we are performing the transfer learning to borrow information from each source signal (not their interactions) to the target. Moreover, the interaction term will influence the transfer learning result indirectly through influencing the modeling accuracy of source signals, which will happen when the number of data points in source signals is small. Fortunately, in the context of transfer learning, the data availability in sources is usually sufficient to accurately learn the covariance in each source signal.

As a result, the impact of posing zeros among sources becomes negligible on the transfer learning.

B. Iterative Transfer Active Learning

The proposed structure of MGP in Section III-A needs to be adapted with active learning, which requires an iterative procedure to estimate parameters $\boldsymbol{\eta}$ and predict mean and covariance at arbitrary inputs $\boldsymbol{x}_N$. To facilitate this procedure, we split the data into small batches. In the tth batch, there will be D_n data in the nth signal, which is denoted as $\mathbf{x}_n^{(t)} = [\mathrm{x}_{n1}^{(t)}, \cdots, \mathrm{x}_{nD_n}^{(t)}]^T$ and $\mathbf{y}_n^{(t)} = [\mathrm{y}_{n1}^{(t)}, \cdots, \mathrm{y}_{nD_n}^{(t)}]^T$. Similar to the notation system for $\mathbf{x}$ and $\mathbf{y}$, the $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ represent all data in the tth batch, where $D_n \ll L_n$. In this case, the transfer active learning in Eq. 1 can be re-formulated as:

$$\mathbf{x}_N^{*(t+1)} = \underset{\boldsymbol{x}_N^{(t+1)}}{\arg\min}\, g(\boldsymbol{x}_N^{(t+1)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}) \tag{10}$$

where $\mathbf{x}_N^{*(t+1)}$ is a D_N-dimension input vector for collecting data in the $(t+1)$th round of active learning, $\mathbf{x}^{(1:t)}$ and $\mathbf{y}^{(1:t)}$ are the data collected from t batches. The key difference between Eq. 1 and Eq. 10 is that Eq. 10 provides an iterative learning procedure by introducing the batch t. In this case, the learning and prediction of $\boldsymbol{x}_N^{(t+1)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$ must be implemented batch-by-batch iteratively. It is also worth noting that the Eq. 1 is a special case of Eq. 10, where we can set $t = 1$ and $D_n = L_n$ in Eq. 10 to represent Eq. 1. This special relationship explains why and how Eq. 10 solves the computational challenge in Eq. 1 by splitting all training data into batches.

To achieve iterative transfer learning formulated in Eq. 10, we introduce inducing points for parameter estimation and prediction. The inducing points serve as the bridge to link data in each batch with the parameter estimation and prediction of functional relationships. In this case, the data from each batch is used to update the joint distribution of inducing points, whose information can then be propagated to the prediction results. More specifically, we denote Q_n inducing points and their MGP based outputs in the nth signal as $\widetilde{\mathbf{x}}_n = [\widetilde{\mathrm{x}}_{n1}, ..., \widetilde{\mathrm{x}}_{nQ_n}]^T$ and $\mathbf{h}_n = [f_n(\widetilde{\mathrm{x}}_{n1}), ..., f_n(\widetilde{\mathrm{x}}_{nQ_n})]^T$, respectively. Similarly, the $\widetilde{\mathbf{x}}$ and $\mathbf{h}$ are denoted for inducing points and their outputs in all N signals. Note the inducing points do not have superscript t because they will not change in each batch.

As a result, the iterative parameter estimation and prediction can be formulated in a Bayesian strategy:

$$\begin{aligned} &p(\boldsymbol{f}_N(\boldsymbol{x}_N), \mathbf{h}, \boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}) \\ &= p(\boldsymbol{f}_N(\boldsymbol{x}_N) | \mathbf{h}, \boldsymbol{\eta}^{(t)}) p(\mathbf{h} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)}) p(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}) \end{aligned} \tag{11}$$

where $p(\cdot)$ is the probability density function, and $\boldsymbol{\eta}^{(t)}$ represents the updated parameters after observing t batches of data. Equation 11 succinctly encapsulates the iterative procedures of our framework, where $p(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})$ facilities the iterative estimation of model parameters and the inducing points $\mathbf{h}$ links the updated parameters $p(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})$ with the iterative prediction $p(\boldsymbol{f}_N(\boldsymbol{x}_N) | \mathbf{h}, \boldsymbol{\eta}^{(t)})$. In this case, the iterative updating hinges on the formulation of $p(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})$ and $p(\mathbf{h} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)})$.

The key to facilitating the iterative parameter estimation is to construct the $p(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})$ from $p(\boldsymbol{\eta}^{(t-1)} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)})$, which is formulated as follows:

$$\begin{aligned} &p\left(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\right) \\ &\propto p(\mathbf{y}^{(t)} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)}) \cdot \\ &\int p(\boldsymbol{\eta}^{(t)} | \boldsymbol{\eta}^{(t-1)}) p(\boldsymbol{\eta}^{(t-1)} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}) d\boldsymbol{\eta}^{(t-1)} \end{aligned} \tag{12}$$

where the $p(\mathbf{y}^{(t)} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)})$ serves as the prediction part, and it follows a Normal distribution with explicit mean and variance expression. The detailed derivation of $p(\mathbf{y}^{(t)} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)})$ is provided in Section B of supplementary materials. The $p(\boldsymbol{\eta}^{(t)} | \boldsymbol{\eta}^{(t-1)})$ in Eq. 12 represents the transition from parameters at the $(t-1)$th batch to the tth batch. Such transition aims to explore appropriate parameter distributions to represent the $p\left(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\right)$. We employ the widely used marginalized particle filter [53] to formulate the $p(\boldsymbol{\eta}^{(t)} | \boldsymbol{\eta}^{(t-1)})$ and complete the integration in Eq. 12:

$$p(\boldsymbol{\eta}^{(t)} | \boldsymbol{\eta}^{(t-1)}) = \vartheta \boldsymbol{\eta}^{(t-1)} + (1 - \vartheta) \bar{\boldsymbol{\eta}}^{(t-1)} + \boldsymbol{r}_{t-1} \tag{13}$$

where ϑ is the smoothing effect, $\boldsymbol{r}_{t-1}$ is a sample from $N(\mathbf{0}, (1 - \vartheta^2) \boldsymbol{R}_{t-1})$ that represents the random fluctuation during parameter estimation in the previous iteration, and $\bar{\boldsymbol{\eta}}^{(t-1)}$ and $\boldsymbol{R}_{t-1}$ are the sample mean and covariance of $\boldsymbol{\eta}^{(t-1)}$, respectively. The detailed procedures of the marginalized particle filter for obtaining $p\left(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\right)$ are provided in Section C of supplementary materials for the completeness of the paper.

Similar to the calculation of $p\left(\boldsymbol{\eta}^{(t)} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\right)$, the key to facilitating iterative prediction is to get $p(\mathbf{h} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)})$ from $p(\mathbf{h} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)})$. To facilitate this procedure, we propose a lemma as follows:

Lemma 1: If $\mathbf{h} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)} \sim MVN(\boldsymbol{\alpha}_{t-1}, \mathbf{C}_{t-1})$ $(t > 1)$, then $\mathbf{h} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)} \sim MVN(\boldsymbol{\alpha}_t, \mathbf{C}_t)$, and the mean vector and covariance matrix can be calculated as:

$$\begin{aligned} \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1} + \mathbf{C}_{t-1} \mathbf{G}_t^T \mathbf{P}_t^{-1} (\mathbf{y}^{(t)} - \boldsymbol{\zeta}_t) \\ \mathbf{C}_t &= \mathbf{C}_{t-1} - \mathbf{C}_{t-1} \mathbf{G}_t^T \mathbf{P}_t^{-1} \mathbf{G}_t \mathbf{C}_{t-1} \end{aligned} \tag{14}$$

where $\mathbf{G}_t = \boldsymbol{\Gamma}(\mathbf{x}^{(t)}, \widetilde{\mathbf{x}}) \boldsymbol{\Gamma}^{-1}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}}), \boldsymbol{\zeta}_t = \mathbf{G}_t \boldsymbol{\alpha}_{t-1}$, $\mathbf{P}_t = \boldsymbol{\Gamma}(\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) + \mathbf{G}_t (\mathbf{C}_{t-1} - \boldsymbol{\Gamma}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})) \mathbf{G}_t^T + \sigma^2 \mathbf{I}_{L^{(t)}}$, and $L^{(t)} = \Sigma_n D_n$ is the total number of observations at the tth batch. It is worth noting that the $\mathbf{h} | \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)}$ and $\mathbf{h} | \mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \boldsymbol{\eta}^{(t)}$ both follow multi-variate Normal (MVN) distribution is a direct result from the assumption A1. The proof of Lemma 1 is provided in Section B of supplementary materials. The direct impact of Lemma 1 on the proposed MGP is to reduce the computational complexity in each data batch to $\mathcal{O}(\sum_n D_n^3)$, which is much less than directly inversing the Eq. 8, i.e., $\mathcal{O}(\sum_n L_n^3)$, There are also some insights resulting from Lemma 1, which are provided in the following remark.

Remarks on Lemma 1: The calculation of $\mathbf{C}_t$ in Eq. 14 only needs the input (i.e., no data is explicitly needed). This feature actually inspires the construction of Eq. 10 since it provides an intrinsic way for incorporating $\boldsymbol{x}_N^{(t+1)}$ into the prediction of variance for the $(t+1)$th batch. In other words, the $\mathbf{C}_{t+1}$ can be constructed with information $\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$ and $\boldsymbol{x}_N^{(t+1)}$, and it can be used for the calculation of IMSE in the $(t+1)$th batch.

Based on Eqs. 13 and 14, the $p(\boldsymbol{f}_N(\boldsymbol{x}_N), \mathbf{h}, \boldsymbol{\eta}^{(t)}| \mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})$ in Eq. 11 can be obtained by Woodbury formula, and the marginalized results for prediction can be in a closed-form:

$$\begin{aligned}&\boldsymbol{f}_N(\boldsymbol{x}_N)|\mathbf{x}^{(1:t)}, \mathbf{y}^{\ (1:t)}\\&\sim N(\mathbf{G}\boldsymbol{\alpha}_t, \boldsymbol{\gamma}_{NN}(\mathbf{x}_N, \mathbf{x}_N) + \mathbf{G}\Big(\mathbf{C}_t - \boldsymbol{\Gamma}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})\Big)\mathbf{G}^T)\end{aligned} \tag{15}$$

where $\mathbf{G} = \boldsymbol{\Gamma}(\boldsymbol{x}_N, \widetilde{\mathbf{x}})\boldsymbol{\Gamma}^{-1}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$. The critical role of Eq. 15 is that it facilitates the iterative calculation of IMSE and formulates the transfer active learning through the inducing points. Specifically, when the new batch of data is available, the data is used to update/estimate the joint distribution of the inducing points (through Lemma 1). Then, the updated joint distribution of the inducing points is delivered to the prediction of the target signal (Eq. 15). In other words, the information in the sequentially incoming data batches is embedded or updated in the joint distribution of inducing points. In this case, the iterative updating is facilitated by inferring the target signal from the updated inducing points. As a result, Eq. 10 can have a concrete formulation with the proposed method:

$$\begin{aligned}&\mathbf{x}_N^{*(t+1)} = \underset{\boldsymbol{x}_N^{(t+1)}}{\arg\min}\, g(\boldsymbol{x}_N^{(t+1)}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})\\&g(\boldsymbol{x}_N^{(t+1)}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)})\\&= \int \boldsymbol{\gamma}_{NN}(x, x) + \boldsymbol{\mathcal{G}}\big(\mathbf{C}_{t+1} - \boldsymbol{\Gamma}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})\big)\boldsymbol{\mathcal{G}}^T \mathrm{d}x\end{aligned} \tag{16}$$

where $\boldsymbol{\mathcal{G}} = \boldsymbol{\Gamma}(x, \widetilde{\mathbf{x}})\boldsymbol{\Gamma}^{-1}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$, $\mathbf{C}_{t+1} = \mathbf{C}_t - \mathbf{C}_t\mathbf{G}_{t+1}^T\mathbf{P}_{t+1}^{-1}\mathbf{G}_{t+1}\mathbf{C}_t$, $\mathbf{G}_{t+1} = \boldsymbol{\Gamma}(\mathbf{x}^{(t+1)}, \widetilde{\mathbf{x}})\boldsymbol{\Gamma}^{-1}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$, $\mathbf{P}_{t+1} = \boldsymbol{\Gamma}(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t+1)}) + \mathbf{G}_{t+1}(\mathbf{C}_t + \boldsymbol{\Gamma}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}}))\mathbf{G}_{t+1}^T + \sigma^2\mathbf{I}_{L^{(t+1)}}$, and x is arbitrary input value in the Nth signal. It is worth noting that the formula inside the integration of Eq. 16 is different with the variance part in Eq. 15. This is because Eq. 15 aims to predict the function value of $\boldsymbol{f}_N$ at $\boldsymbol{x}_N$ based on information of $\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$, while Eq. 16 treats $\boldsymbol{x}_N^{(t+1)}$ as decision variable to optimize an objective that is a function of $\boldsymbol{x}_N^{(t+1)}$. In other words, the $g(\cdot)$ in Eq. 16 generates IMSE based on the value of $\boldsymbol{x}_N^{(t+1)}$, i.e., different values of $\boldsymbol{x}_N^{(t+1)}$ result in different IMSE values. This explains the rationale of how the IMSE works: the impact of $\boldsymbol{x}_N^{(t+1)}$ on the prediction is evaluated by the predicted variance at every prediction input. Comparing Eq. 16 with Eq. 5 can also reveal significant differences, where the information from $(N-1)$ more signals is incorporated and the whole procedure becomes iterative due to the inclusion of data batches.

We also summarize the proposed transfer active learning in Algorithm 1. Note the computational complexity in each iteration of the offline and online stage is $\mathcal{O}(\sum_n D_n^3)$ and $\mathcal{O}(D_N^3)$, respectively, where $D_n \ll L_n < L$, $n = 1, \cdots, N$.

Algorithm 1 Real-time transfer active learning.

```
Input: Rich data in source signals: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{N-1}, \mathbf{y}_{N-1})$, and sparse data in the target signal: $(\mathbf{x}_N, \mathbf{y}_N)$
Offline transfer:
Divide all historical data $\mathbf{x}, \mathbf{y}$ into $T$ batches: $\mathbf{x}^{(1:T)}, \mathbf{y}^{(1:T)}$
Initialize parameter: $\boldsymbol{\eta}^{(0)}, \boldsymbol{\alpha}_0, \boldsymbol{C}_0$, and set $t = 0$
while $t \leq T$ do
    $t \leftarrow t + 1$
    Calculate $p\Big(\boldsymbol{\eta}^{(t)}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\Big)$ based on Eq. 12 and Eq. 13
    Update $\boldsymbol{\alpha}_t, \boldsymbol{C}_t$ based on Lemma 1 and obtain $p(\mathbf{h}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)})$
end while
Online active learning:
while $t \geq T$ do
    $t \leftarrow t + 1$
    Obtain new experiment inputs $\mathbf{x}_N^{*(t)}$ by Eq. 16
    Implement experiments at $\mathbf{x}_N^{*(t)}$ and obtain data $\mathbf{y}_N^{*(t)}$
    Update the data pool as $\mathbf{x}^{(1:t)} = [\mathbf{x}^{(1:t-1)}, \mathbf{x}_N^{*(t)}]$ and $\mathbf{y}^{(1:t)} = [\mathbf{y}^{(1:t-1)}, \mathbf{y}_N^{*(t)}]$
    Calculate $p\Big(\boldsymbol{\eta}^{(t)}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\Big)$ based on Eq. 12 and Eq. 13
    Update $\boldsymbol{\alpha}_t, \boldsymbol{C}_t$ based on Lemma 1 and obtain $p(\mathbf{h}|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}, \boldsymbol{\eta}^{(t)})$
    Update the input-output relationship in the target signal, i.e., $\boldsymbol{f}_N(\boldsymbol{x}_N)|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$, based on Eq. 15
end while
Output: Experiment inputs in the $N$th signal, i.e., $\mathbf{x}_N^{*(T+1)}, \cdots, \mathbf{x}_N^{*(t)}$, and the input-output relationship in the target signal, i.e., $\boldsymbol{f}_N(\boldsymbol{x}_N)|\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$.
```

C. Properties of the Proposed Transfer Active Learning

The IMSE based transfer active learning in Algorithm 1 provides a feasible solution to transfer learning modeling, interpretation, and real-time updating, which demonstrates strong potential for solving the cold-start problem in active learning. In this section, we will discuss the learning properties of the proposed method and show that the proposed method can achieve a strictly monotonically decreasing IMSE for each learning step. To facilitate the demonstration of the property, we re-denote the function $g(\cdot)$ in Eq. 16 as follows:

$$IMSE_{t+1}(\boldsymbol{x}_N^{(t+1)}) = g(\boldsymbol{x}_N^{(t+1)}|\mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}, \mathbf{x}_N^{*(t)}, \mathbf{y}_N^{*(t)}) \tag{17}$$

where $\mathbf{x}_N^{*(t)} = \arg\min_{\boldsymbol{x}_N^{(t)}} g(\boldsymbol{x}_N^{(t)}|\mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)})$ is the actively learned inputs in the previous batch, and the $\mathbf{y}_N^{*(t)}$ is the corresponding data collected at $\mathbf{x}_N^{*(t)}$. It is worth noting that the data $\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}$ is constructed by $\mathbf{x}^{(1:t-1)}, \mathbf{y}^{(1:t-1)}$ and the $\mathbf{x}_N^{*(t)}, \mathbf{y}_N^{*(t)}$ because only the optimized data in the previous batch can be incorporated in the current batch. Equation 17 thus builds the relationship between active learning results at the $(t-1)$th batch, i.e., $\mathbf{x}_N^{*(t)}$, and the active learning formulation at the tth batch. Similarly, we can denote $IMSE_t\left(\boldsymbol{x}_N^{(t)}\right)$ as follows:

$$IMSE_t\left(\boldsymbol{x}_N^{(t)}\right) = g(\boldsymbol{x}_N^{(t)}|\mathbf{x}^{(1:t-2)}, \mathbf{y}^{(1:t-2)}, \mathbf{x}_N^{*(t-1)}, \mathbf{y}_N^{*(t-1)}) \tag{18}$$

It is clear that the difference between $IMSE_{t+1}\left(\boldsymbol{x}_N^{(t+1)}\right)$ and $IMSE_t\left(\boldsymbol{x}_N^{(t)}\right)$ is the information gain or the reduction of IMSE between consecutive learning steps, which depends on the values of $\boldsymbol{x}_N^{(t+1)}$ and $\boldsymbol{x}_N^{(t)}$. We provide a lemma that the information gain (reduction of IMSE), under some regularity conditions, is always positive:

Lemma 2: Under regularity conditions, if Lemma 1 holds, then we have

$$\begin{aligned} &\Delta IMSE_{t+1}(\boldsymbol{x}_N^{(t+1)}, \mathbf{x}_N^{*(t)}) \\ &= IMSE_{t+1}\left(\boldsymbol{x}_N^{(t+1)}\right) - IMSE_t\left(\mathbf{x}_N^{*(t)}\right) < 0 \end{aligned} \tag{19}$$

The proof of *Lemma 2* is available in Section D of supplementary materials.

Remarks on Lemma 2: The critical contribution of Lemma 2 is that it reveals a strictly monotonic trajectory of the IMSE when getting more data from the proposed transfer active learning framework. Although it is widely known that the IMSE of Gaussian process converges to 0 [54], the strictly monotonic decreasing property provides more meaningful insights for the active learning, which shows every data identified by the proposed framework can contribute to a more accurate prediction of functional relationship. This is intrinsically desired by active learning. Moreover, Lemma 2 also provides theoretical justifications for the cold-start problem since it guarantees even data identified at the very beginning in the target signal can still contribute to the learning procedure.

IV. Numerical Studies

In this section, the performance of our proposed transfer active learning will be investigated by comparing with benchmarks under some commonly used signal settings. Specifically, we introduce three different benchmarks for different comparison purposes under three different signal settings, i.e., trigonometric, polynomial, and a large number of signals. The three benchmarks are as follows:

1) The first benchmark is "Single GP", which only uses data from the target signal thus cannot learn information from source signals. This benchmark is to demonstrate the difference between transfer learning and non-transfer learning. It also provides the case that the kernels/models are set as independent between signals while the functions/signals are indeed correlated.
2) The second benchmark utilizes the linear model of coregionalization for constructing MGP, denoted as "LMC". This is a prevalent offline approach and is introduced in Eq. 9. In this model, the data fed to the algorithm is the same as the proposed method, but the "LMC" suffers modeling flexibility. Besides, it needs to re-train all data whenever the new batch is available. As a result, it provides a baseline for performance and time consumption of most existing offline MGP method.
3) The third benchmark method is based on multi-task learning. It uses the same modeling and real-time updating framework as our proposed method. The only difference is that our method first learns all data batches in source signals then applies the learned knowledge to target signal in real-time, while the multi-task learning splits the data in sources into multiple batches and learns each source batch together with the target batch. The multi-task learning aims to provide a comparison of learning strategies between transfer and simultaneous learning.

Some parameter and procedure setups for all methods are as follows. We set ϑ in Eq. 13 as 0.98 for all real-time updating methods, whose value is recommended in [55]. We use 40 evenly distributed inducing points for each signal in the numerical studies, i.e., $Q_n = 40$ for $n = 1, \cdots, N$. In all numerical studies, 17 data batches are generated (each batch has 5 observations for each signal, i.e., $L_n = 5$), and the first two data batches are used to initialize parameters in Algorithm 1, i.e., $\{\boldsymbol{\eta}^{(0)}, \boldsymbol{\alpha}_0, \boldsymbol{C}_0\}$. We have also investigated the impacts of batch sizes on the performance of the proposed method, which is available in Section E of the supplementary materials. It should be noted that the result are robust to initial parameters because the model parameters will be updated when new data batches are collected. To evaluate the performance, we set 50 test points evenly distributed in the input space for each signal and compare the root mean squared error (RMSE) between the predicted and the true values at these 50 points. All experiments are conducted 100 times to report the average performance of each method.

The trigonometric signal settings are as follows:

$$\begin{aligned} y_1(x) &= 10\cos(x) + \epsilon_1(x) \\ y_2(x) &= 10\sin(x) + \epsilon_2(x) \\ y_3(x) &= 5\exp\left(-\frac{x}{10}\right)\Big(\cos(x) + \sin(x)\Big) + \epsilon_3(x) \end{aligned} \tag{20}$$

where $x \in [-2, 2]$, $\epsilon_i(x) \sim N(0, 0.5^2)$ for $i = 1, 2, 3$ is i.i.d. noise. The third signal is designated as the target. To evaluate the performance of transfer learning and active learning of our proposed method, We provide two different evaluation settings. In the first setting, the 17 data batches are randomly selected and fed to each method, which aims to provide evaluations of the proposed transfer learning framework (without active learning). The second setting involves the proposed IMSE based active learning, and the comparison between the first and second setting demonstrates improvement from random sampling to active learning. The performance (evaluated under RMSE) of the first and second setting is demonstrated in Figs. 3 and 4, respectively. Please note that we only include the LMC in the random sample setting because the LMC is an offline MGP method, which cannot be directly applied in real-time active learning.

Based on results in Figs. 3 and 4, some insights and discussions are summarized as follows:

1) Evidence of cold-start. It is clear from Figs. 3 and 4 that the single signal based learning ("Single GP") suffers cold-start issue, where the RMSE in initial batches is

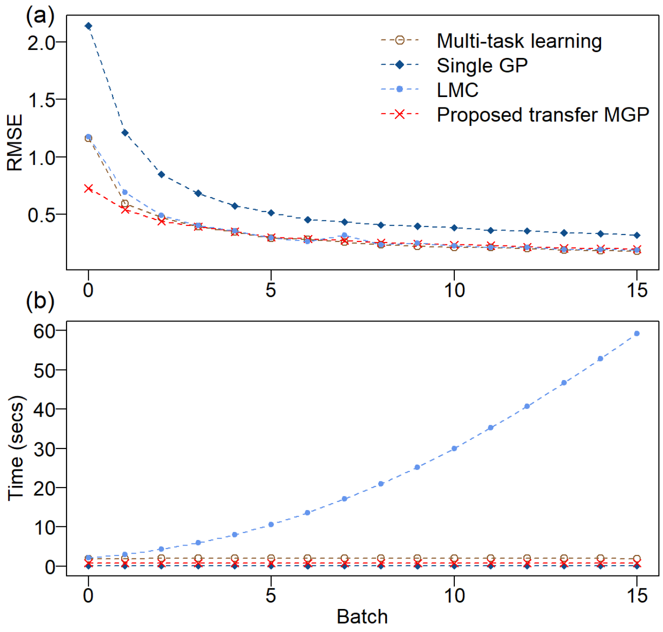


Fig. 3. RMSE results for the 3rd trigonometric signal (random training data samples). (a) RMSE, (b) Time consumption.

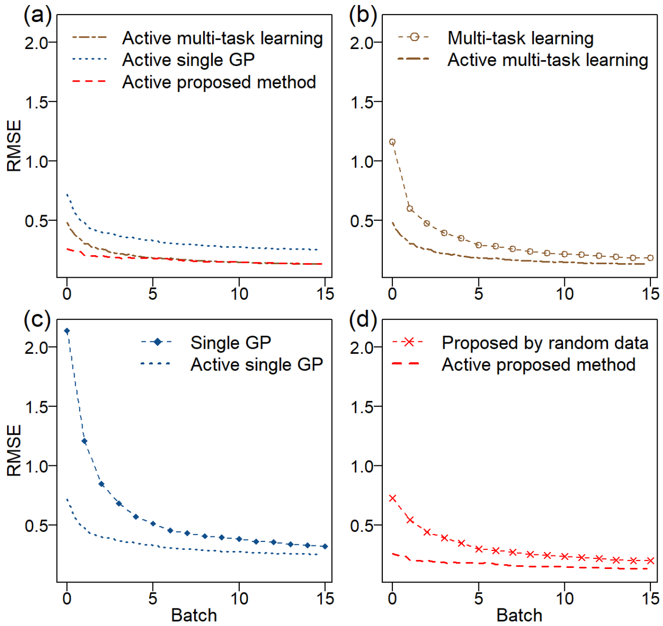


Fig. 4. RMSE results for the 3rd trigonometric signal (actively learned training data samples). (a) RMSE based on active learning using three methods, (b)-(d) Random sample vs. Active learning sample using multi-task learning, single GP, and the proposed method, respectively.

much higher than all other transfer learning based approaches. This is because the "Single GP" does not have sufficient data to train the model parameters, especially at the initial several batches. The better performance of transfer learning based approaches also validates our motivation that the cold-start issue can be alleviated by transfer learning.

2) Effectiveness of transfer learning. The performance of transfer learning varies among different approaches. This is especially clear in Fig. 3(a), where the impact of active learning is excluded (all methods use the same training data) for evaluating the performance of transfer learning. It is clear that the proposed method achieves the best performance, especially at the initial batches. Specifically, comparing to "Multi-task learning", the proposed method already learns from all source data at the first batch while the "Multi-task learning" only learns one batch from each of the source signal. Such difference in data availability provides reasonable justifications for the superior performance of the proposed method. Comparing to "LMC", the proposed method enjoys a more flexible modeling framework (see justifications in Eq. 9), which contributes to the superior performance under the same data availability.

3) Time efficiency. The time consumption of each method in each batch is depicted in Fig. 3(b), where the superiority of the real-time updating is evidenced by observing the larger and larger gap between offline LMC and other real-time methods. More notably, our proposed method shares very similar time consumption with the single GP method but can achieve much better RMSE performance.

4) Effectiveness of active learning. The performance of active learning is demonstrated in Fig. 4, where the RMSE with and without active learning is compared. The RMSE without active learning is the same as those in Fig. 3(a) (by randomly sampling training data). It is clear that the IMSE based active learning can effectively improve the RMSE performance, and the proposed method again achieves the best performance.

We also test the settings on polynomial signals, which are formulated as follows:

$$\begin{aligned} y_1(x) &= \frac{1}{2}(x^2 + x - 3) + \epsilon_1(x) \\ y_2(x) &= \frac{1}{2}(3x^2 + 3x + 5) + \epsilon_2(x) \\ y_3(x) &= x^2 + 2x + \epsilon_3(x) \end{aligned} \tag{21}$$

where $x \in [-2, 2]$, $\epsilon_i(x) \sim N(0, 0.5^2)$ for $i = 1, 2, 3$ is i.i.d. noise. We also choose the third signal as the target signal. The results are shown in Figs. 5 and 6.

It is clear that the Figs. 5 and 6 support and validate the discoveries in the trigonometric case, which demonstrates the superiority of the proposed transfer active learning in terms of transfer learning accuracy, time efficiency, and active learning performance.

To further evaluate the proficiency of the proposed method in managing multiple signals, we configure 8 signals as follows:

$$\begin{aligned} y_1(x) &= x^2 + 2x + 5 + \epsilon_1(x) \\ y_2(x) &= -2x^2 + x - 5 + \epsilon_2(x) \\ y_3(x) &= 3x^3 - x^2 + 4 + \epsilon_3(x) \\ y_4(x) &= -x^3 + \frac{x^2}{2} - 3 + \epsilon_4(x) \\ y_5(x) &= -3x^3 + 2x + \epsilon_5(x) \\ y_6(x) &= 2x^3 + 3x + \epsilon_6(x) \\ y_7(x) &= 2x^3 + x^2 + 2x + \epsilon_7(x) \\ y_8(x) &= x^3 - x^2 - x + 2 + \epsilon_8(x) \end{aligned} \tag{22}$$

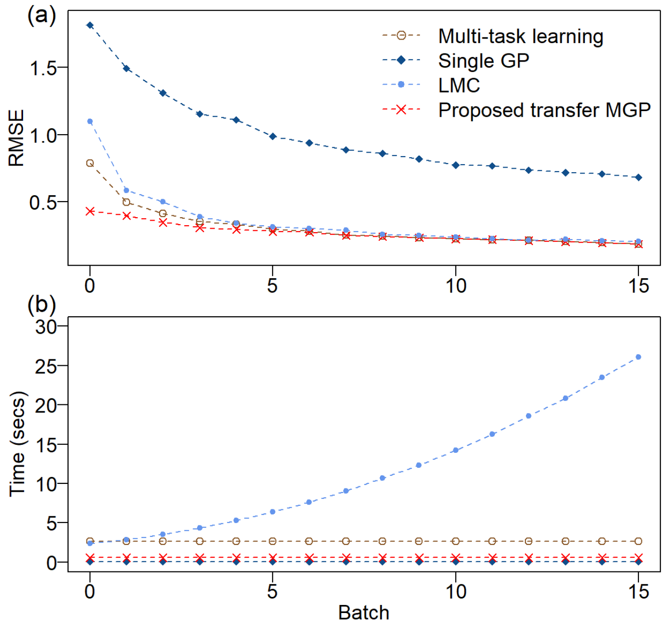


Fig. 5. RMSE results for the 3rd polynomial signal (random training data samples). (a) RMSE, (b) Time consumption.

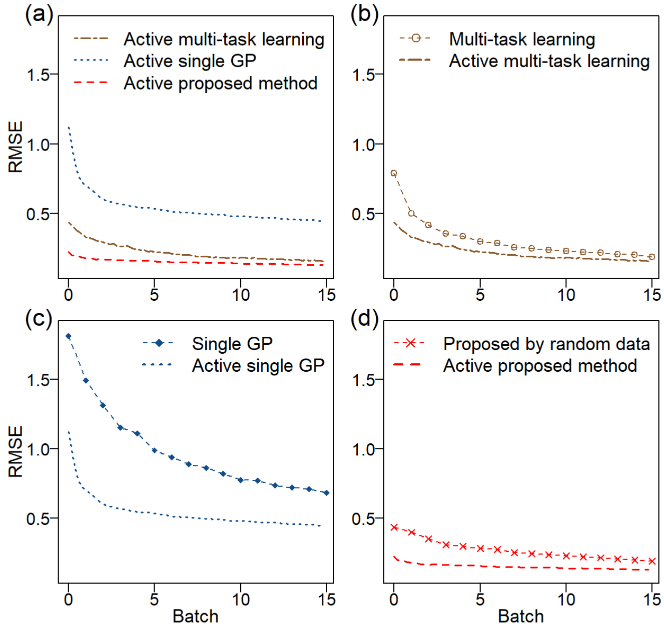


Fig. 6. RMSE results for the 3rd polynomial signal (actively learned training data samples). (a) RMSE based on active learning using three methods, (b)-(d) Random sample vs. Active learning sample using multi-task learning, single GP, and the proposed method, respectively.

where $x \in [-2, 2]$, $\epsilon_i(x) \sim N(0, 0.5^2)$ for $i = 1, \cdots, 8$ is i.i.d. noise. Again, the eighth signal is set as the target, and we transfer the information from $y_1, \cdots, y_7$ to y_8. The results under random and active learning are shown in Figs. 7 and 8, respectively. These results align with our earlier discussions, especially for the time efficiency and active learning performance, which again validates the effectiveness of the proposed in dealing with the cold-start problem with a large number of signals.

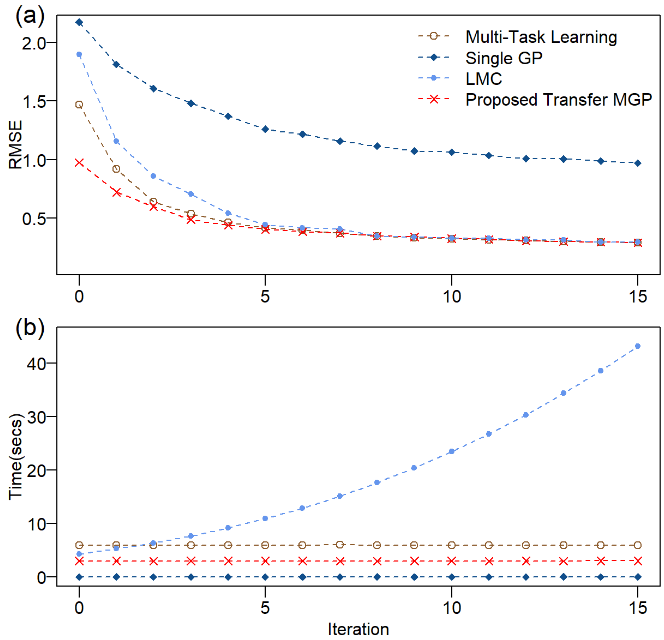


Fig. 7. RMSE results for the 8th polynomial signal (random training data samples). (a) RMSE, (b) Time consumption.

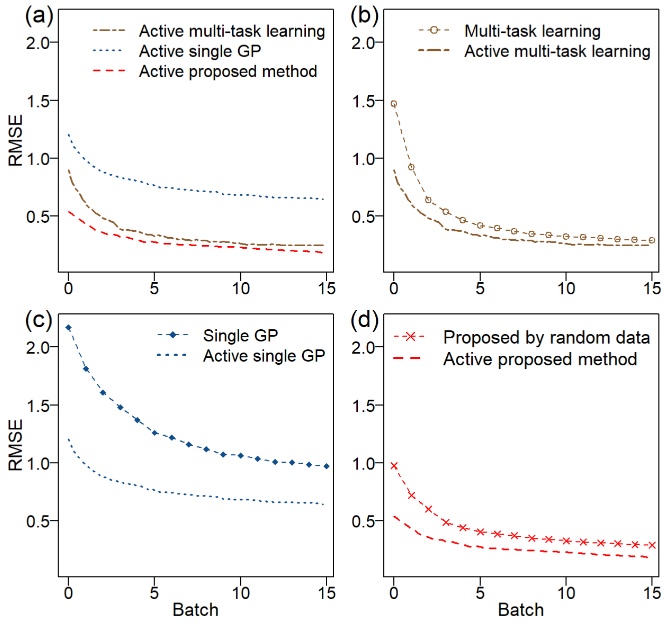


Fig. 8. RMSE results for the 8th polynomial signal (actively learned training data samples). (a) RMSE based on active learning using three methods, (b)-(d) Random sample vs. Active learning sample using multi-task learning, single GP, and the proposed method, respectively.

V. Case Studies

In this section, we employ two sets of real-world data to evaluate the performance of our proposed method. Both

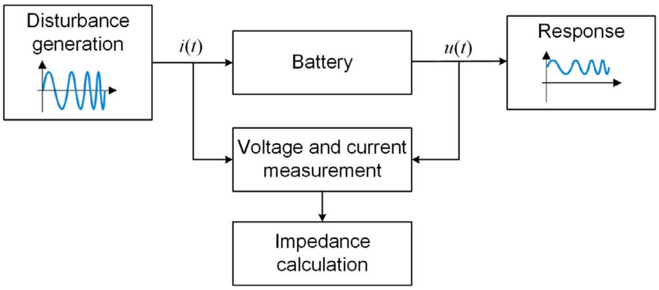


Fig. 9. Measurement diagram of battery impedance.

cases suffer cold-start issues when selecting new and costly experiment settings. The first data set originates from electrochemical impedance spectroscopy (EIS) test pertaining to batteries. The second dataset is from sensor calibration of reduced graphene oxide field-effect transistors (RGO FET). For both case studies presented, we maintained consistency in parameter settings, adhering to the configurations used in preceding numerical analysis.

The EIS test is a widely adopted characterization technique to estimate the internal state of electrochemical systems, such as lithium ion batteries [56]. A measurement diagram of battery impedance is shown in Fig. 9. During the measurement process, a small disturbance current is utilized to excite the system, and the impedance is then calculated with the response divided by input. However, EIS test usually takes long time since it requires to cover a wide frequency range [57], affecting its real-life applications. In addition, the selection of specific frequency for testing is difficult and requires strong domain knowledge. This is because the battery state is unknown before the EIS test, i.e., a black-box problem. As a result, the EIS test requires efficient selection of the most representative frequency locations to obtain input-output curves.

To test the performance of transfer active learning of effective frequency locations, we use 8 EIS curves, which are displayed in Fig. 10(a). Each curve represents the relationship between frequency and impedance of the battery under different aging states. It is clear that these curves have strong correlations, thus facilitate the transfer learning. We randomly pick one curve as the target and use the rest as sources. Fig. 10(b) demonstrates the progression of actively learned RMSE across increasing batches. The results show the proposed method only use 2 batches to achieve a steadily low RMSE of the target curve, which results in significant improvement over benchmark methods. Note we also provided the comprehensive results of 100 replications for the EIS data, which are available in Section E of the supplementary materials.

The second case study is for calibrating reduced graphene oxide field-effect transistors based sensors, which have wide applications in bio-engineering and environment protection [58], [59]. The basic structure of a RGO FET is shown in Fig. 11, where the V_{gs} is the gate voltage and the V_{ds} is the drain-source voltage. When a to be monitored object, e.g., protein molecule or chemical ion, touches the RGO, the reaction between the RGO and the object will change the resistant between the drain

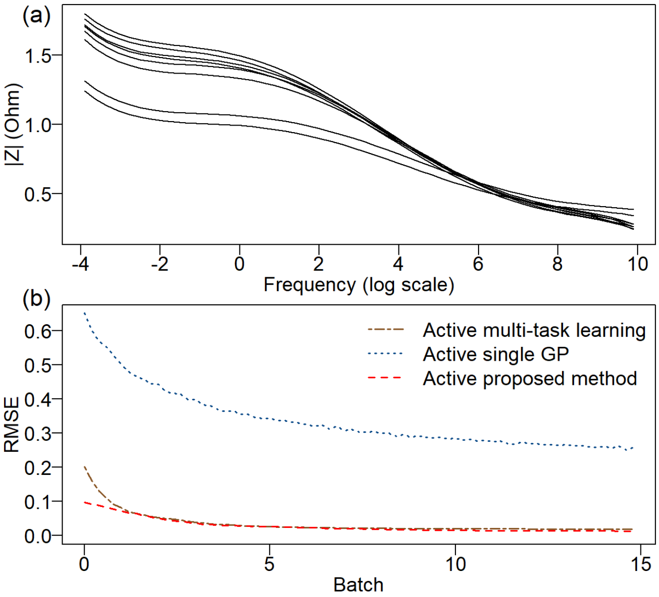


Fig. 10. Transfer active learning of EIS relationship. (a) EIS data, (b) RMSE results of transfer active learning.

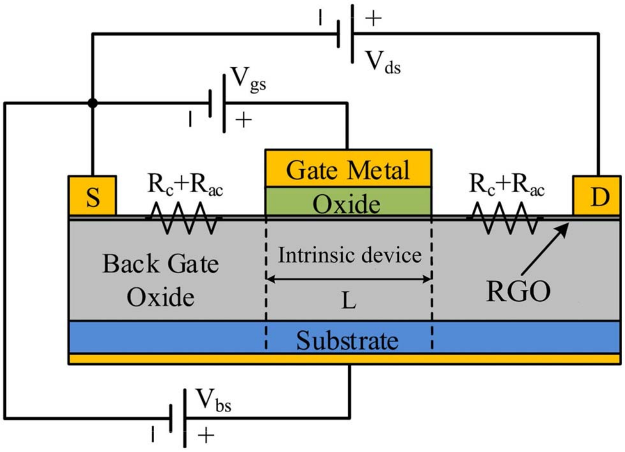


Fig. 11. Cross-section view of RGO FET [13].

and the source so that the change in I_{ds} can report the detection of the object. Due to the different reaction mechanism between the RGO and the to be monitored objects, different V_{gs} and V_{ds} values will be used for sensing different objects/materials [60]. However, every RGO FET sensor is disposable, which means the calibration of sensor will consume lots of sensors. As a result, it is desired to use as few sensors as possible to calibrate the V_{gs} vs. I_{ds} relationship for a new detection task.

Such task is feasible using transfer learning because there are many already calibrated V_{gs} vs. I_{ds} relationship in previous tasks. This is shown in Fig. 12(a), where each curve is a V_{gs} vs. I_{ds} relationship under a specific V_{ds}. It is clear that these curves have strong within-and between-signal correlation. To validate the effectiveness of our proposed method, we randomly select one curve as the target and treat the rest as sources. Fig. 12(b) presents the transfer active learning results and comparisons,

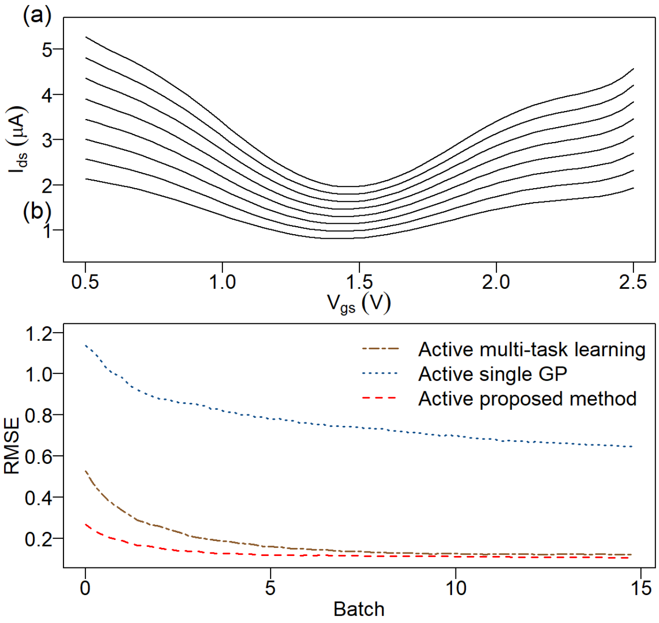


Fig. 12. Transfer active learning of RGO FET relationship. (a) V_{gs} vs. I_{ds} data, (b) RMSE results of transfer active learning.

where our proposed method can achieve the best calibration of relationship with the minimum number of sensors, especially at the initial stage. This case study again validates the efficiency and superiority of the proposed method for dealing with the cold-start problem in active learning. Note we also provided the comprehensive results of 100 replications for the FET data, which are available in Section E of the supplementary materials.

VI. Conclusion

In this paper, a real-time transfer active learning framework is proposed to deal with cold-start problems in learning functional relationship. The proposed framework features an interpretable transfer learning structure that facilitates the modeling of both within-and between-signal relationship. Moreover, the Bayesian updating is developed to expedite the estimation and prediction of the proposed MGP, which accommodates the prediction results to iterative active learning. Finally, the IMSE is used as the objective in transfer active learning, and we provide theoretical justifications for the performance and superiority of the proposed framework. Various numerical studies are conducted to evaluate and compare the performance of the proposed framework in terms of transfer learning accuracy, time efficiency, and the transfer active learning performance. Two real-world case studies are also implemented to demonstrate the effectiveness of the proposed method in practice. The superior performance in both numerical and case studies provides solid evidence that the proposed method is an effective solution to cold-start issues when learning functional relationship.

There are several opening topics based on our proposed framework. For example, in our work, we use evenly distributed inducing points to store information accumulated from iteratively collected data. In practice, it would be desired to redistribute the inducing points based on the data collected in each batch. In this case, the locations of inducing points become unknown parameters and should be optimized accordingly. Such operation is expected to generate better learning results, but it also increases the optimization load. It would be interesting to evaluate and balance the performance vs. time consumption under such situation. Another interesting yet challenging topic is to relax the stationary assumption (A1) so that the transfer active learning can not only predict and guide experiments but also track the underlying dynamics of input-output relationship. Such model can be especially useful for applications with dynamically changing experiment environment. Finally, it is also interesting to investigate the monotone decreasing properties of the proposed updating mechanism in other active learning strategies or acquisition functions. For example, it is easy to derive that the IMSE would also be monotonically decreasing if we select the next experiment input as the one reducing the most variance. However, whether the property holds for many other strategies or how the efficiency/performance in those strategies are interesting topics worthy of further investigations. We will study these topics in our future work.

References

[1] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, 2009. [Online]. Available: http://digital.library.wisc.edu/1793/60660

[2] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams, *The Design and Analysis of Computer Experiments,* vol. 1. Springer, 2003.

[3] P. Li and Y. Wang, "An active learning reliability analysis method using adaptive Bayesian compressive sensing and monte carlo simulation (ABCS-MCS)," *Rel. Eng. & Syst. Saf.*, vol. 221, 2022, Art. no. 108377.

[4] K. M. Cohen, S. Park, O. Simeone, and S. Shamai, "Bayesian active meta-learning for reliable and efficient AI-based demodulation," *IEEE Trans. Signal Process.*, vol. 70, pp. 5366–5380, 2022.

[5] J. Paisley, X. Liao, and L. Carin, "Active learning and basis selection for kernel-based linear models: A Bayesian perspective," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2686–2700, May 2010.

[6] M. Yuan, H.-T. Lin, and J. Boyd-Graber, "Cold-start active learning through self-supervised language modeling," 2020, *arXiv:2010.09535*.

[7] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, "Addressing the item cold-start problem by attribute-driven active learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 631–644, Apr. 2020.

[8] L. Chen, Y. Bai, S. Huang, Y. Lu, B. Wen, A. Yuille, and Z. Zhou, "Making your first choice: To address cold start problem in medical active learning," in *Proc. Med. Imag. Deep Learn.*, 2024, pp. 496–525.

[9] H. Liu et al., "Colossal: A benchmark for cold-start active learning for 3D medical image segmentation," 2023, *arXiv:2307.12004*.

[10] C. Wang, H. Pu, X. Sui, S. Zhou, and J. Chen, "Hybrid modeling and sensitivity analysis on reduced graphene oxide field-effect transistor," *IEEE Trans. Nanotechnol.*, vol. 20, pp. 404–416, 2021.

[11] X. Sui et al., "Fully inkjet-printed, 2D materials-based field-effect transistor for water sensing," *Adv. Mater. Technol.*, 2023, Art. no. 2301288.

[12] J. Lee, C. Wang, X. Sui, S. Zhou, and J. Chen, "Landmark-embedded Gaussian process with applications for functional data modeling," *IISE Trans.*, vol. 54, no. 11, pp. 1033–1046, 2022.

[13] J. Tian, A. Katsounaros, D. Smith, and Y. Hao, "Graphene field-effect transistor model with improved carrier mobility analysis," *IEEE Trans. Electron Devices*, vol. 62, no. 10, pp. 3433–3440, Oct. 2015.

[14] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proc. NAACL HLT Workshop Act. Learn. Natural Lang. Process.*, 2010, pp. 27–32.

[15] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[16] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4048–4062, Nov. 2018.

[17] J. Yang, S. Li, and W. Xu, "Active learning for visual image classification method based on transfer learning," *IEEE Access*, vol. 6, pp. 187–198, 2018.

[18] P. K. Murali, C. Wang, D. Lee, R. Dahiya, and M. Kaboli, "Deep active cross-modal visuo-tactile transfer learning for robotic object

recognition," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9557–9564, Oct. 2022.

[19] D. Kale and Y. Liu, "Accelerating active learning with transfer learning," in *Proc. IEEE 13th Int. Conf. Data Mining,* Piscataway, NJ, USA: IEEE Press, 2013, pp. 1085–1090.

[20] B. Huang, S. Salgia, and Q. Zhao, "Disagreement-based active learning in online settings," *IEEE Trans. Signal Process.*, vol. 70, pp. 1947–1958, 2022.

[21] Z. Chen, J. Duan, L. Kang, and G. Qiu, "Supervised anomaly detection via conditional generative adversarial network and ensemble active learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7781–7798, Jun. 2023.

[22] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*.

[23] Q. Gu and Q. Dai, "A novel active multi-source transfer learning algorithm for time series forecasting," *Appl. Intell.*, vol. 51, pp. 1326–1350, Mar. 2021.

[24] v. W. Kasper, S. John A, N. William, and T. Luis, "Data and model uncertainty estimation for linear inversion," *Geophys. J. Int.*, vol. 149, no. 3, pp. 625–632, 2002.

[25] E. L. Droguett and A. Mosleh, "Bayesian methodology for model uncertainty using model performance data," *Risk Anal.: Int. J.*, vol. 28, no. 5, pp. 1457–1476, 2008.

[26] X. Yue, Y. Wen, J. H. Hunt, and J. Shi, "Active learning for Gaussian process considering uncertainties with application to shape control of composite fuselage," *IEEE Trans. Automat. Sci. Eng.*, vol. 18, no. 1, pp. 36–46, Jan. 2021.

[27] C. Lee, K. Wang, J. Wu, W. Cai, and X. Yue, "Partitioned active learning for heterogeneous systems," *J. Comput. Inf. Sci. Eng.*, vol. 23, no. 4, 2023, Art. no. 041009.

[28] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric modeling and prognosis of condition monitoring signals using multivariate Gaussian convolution processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, 2018.

[29] P. Tighineanu, K. Skubch, P. Baireuther, A. Reiss, F. Berkenkamp, and J. Vinogradska, "Transfer learning with Gaussian processes for Bayesian optimization," in *Proc. Int. Conf. Artif. Intell. Statist.,* PMLR, 2022, pp. 6152–6181.

[30] B. Konomi, G. Karagiannis, and G. Lin, "On the Bayesian treed multivariate Gaussian process with linear model of coregionalization," *J. Statist. Planning Inference*, vol. 157, pp. 1–15, Feb. 2015.

[31] X. Yue and R. A. Kontar, "Joint models for event prediction from time series and survival data," *Technometrics*, vol. 63, no. 4, pp. 477–486, 2021.

[32] M. A. Alvarez and N. D. Lawrence, "Computationally efficient convolved multiple output Gaussian processes," *J. Mach. Learn. Res.*, vol. 12, pp. 1459–1500, 2011.

[33] S. Li, R. M. Kirby, and S. Zhe, "Deep multi-fidelity active learning of high-dimensional outputs," 2020, *arXiv:2012.00901*.

[34] Y. Zhang, et al., "Near-optimal active learning of multi-output Gaussian processes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 2351–2357.

[35] K. Matsui, et al., "Bayesian active learning for structured output design," 2019, *arXiv:1911.03671*.

[36] W. Shi, D. Yu, and Q. Yu, "A Gaussian process-Bayesian Bernoulli mixture model for multi-label active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 27542–27554, 2021.

[37] C. E. Rasmussen et al., *Gaussian Processes for Machine Learning,* vol. 1. New York, NY, USA: Springer, 2006.

[38] C.-Y. Li, B. Rakitsch, and C. Zimmer, "Safe active learning for multi-output Gaussian processes," in *Proc. Int. Conf. Artif. Intell. Statist.,* PMLR, 2022, pp. 4512–4551.

[39] P. A. Regalia and M. K. Sanjit, "Kronecker products, unitary matrices and signal processing applications," *SIAM Rev.*, vol. 31, no. 4, pp. 586–613, 1989.

[40] P. Sattari et al., "Active learning of multiple source multiple destination topologies," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1926–1937, Apr. 2014.

[41] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 2013.

[42] D. Tran, R. Ranganath, and D. M. Blei, "The variational Gaussian process," 2015, *arXiv:1511.06499*.

[43] M. M. Zhang et al., "Sequential Gaussian processes for online learning of nonstationary functions," *IEEE Trans. Signal Process.*, vol. 71, pp. 1539–1550, 2023.

[44] D. Kuzin, O. Isupova, and L. Mihaylova, "Spatio-temporal structured sparse regression with hierarchical Gaussian process priors," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4598–4611, 2018.

[45] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, Mar. 1996.

[46] D. J. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, no. 4, pp. 590–604, 1992.

[47] S. Seo et al., "Gaussian process regression: Active data selection and test point rejection," in *Proc. Mustererkennung 2000: 22. DAGM-Symp.,* Kiel, New York, NY, USA: Springer, 2000, pp. 27–34.

[48] R. B. Gramacy and H. K. Lee, "Adaptive design and analysis of supercomputer experiments," *Technometrics*, vol. 51, no. 2, pp. 130–145, 2009.

[49] A. Sauer, R. B. Gramacy, and D. Higdon, "Active learning for deep Gaussian process surrogates," *Technometrics*, vol. 65, no. 1, pp. 4–18, 2023.

[50] Z. Wang et al., "Characterizing and avoiding negative transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11293–11302.

[51] M. T. Rosenstein et al., "To transfer or not to transfer," in *Proc. Neural Inf. Process. Syst. Workshop Transfer Learn.*, vol. 898, no. 3, 2005.

[52] D. Higdon, "Space and space-time modeling using process convolutions," in *Quantitative Methods for Current Environmental Issues*. New York, NY, USA: Springer, 2002, pp. 37–56.

[53] W. Peng, Z.-S. Ye, and N. Chen, "Joint online RUL prediction for multivariate deteriorating systems," *IEEE Trans. Ind. Inform.*, vol. 15, no. 5, pp. 2870–2878, May 2019.

[54] L. Le Gratiet and J. Garnier, "Asymptotic analysis of the learning curve for Gaussian process regression," *Mach. Learn.*, vol. 98, pp. 407–433, Mar. 2015.

[55] Y. Wang and B. Chaib-draa, "An online Bayesian filtering framework for Gaussian process regression: Application to global surface temperature analysis," *Expert Syst. Appl.*, vol. 67, pp. 285–295, Jan. 2017.

[56] X. Wang et al., "A review of modeling, acquisition, and application of lithium-ion battery impedance for onboard battery management," *ETransportation*, vol. 7, 2021, Art. no. 100093.

[57] N. Lohmann et al., "Electrochemical impedance spectroscopy for lithium-ion cells: Test equipment and procedures for aging and fast characterization in time and frequency domain," *J. Power Sources*, vol. 273, pp. 613–623, Jan. 2015.

[58] D. Wu et al., "Microvesicle detection by a reduced graphene oxide field-effect transistor biosensor based on a membrane biotinylation strategy," *Analyst*, vol. 144, no. 20, pp. 6055–6063, 2019.

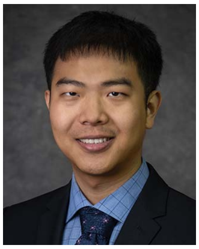
[59] X. Chen et al., "Real-time and selective detection of nitrates in water using graphene-based field-effect transistor sensors," *Environ. Sci.: Nano*, vol. 5, no. 8, pp. 1990–1999, 2018.

[60] G. Zhou et al., "Real-time, selective detection of Pb2+ in water using a reduced graphene oxide/gold nanoparticle field-effect transistor device," *ACS Appl. Mater. Interfaces*, vol. 6, no. 21, pp. 19235–19241, 2014.

Zengchenghao Xia received the B.E. degree in vehicle engineering from Xi'an Jiaotong University, in 2021, and the M.S. degree in mechanical engineering from The Hong Kong University of Science and Technology, in 2022. He is currently working toward the Ph.D. degree with the Department of Industrial and Systems Engineering, University of Iowa. His research interests include statistical modeling, regression, and analysis of smart and connected systems.

Zhiyong Hu received the B.E. degree in mechanical engineering from the Anhui University of Technology, Maanshan, China, in 2011, the M.E. degree in precision instrument and machinery from the University of Science and Technology of China, in 2016, and the Ph.D. degree from the Department of Industrial, Manufacturing and Systems Engineering, Texas Tech University, in 2021. From 2021 to 2024, he was a Postdoctoral Researcher at the Department of Precision Machinery and Precision Instrumentation in University of Science and Technology of China. He is currently with the Department of Automation at Anhui University. His research interests include statistical modeling of multi-stream data analytics for the monitoring, analysis and control of complex systems.

Qingbo He (Senior Member, IEEE) received the B.S. and Ph.D. degrees in mechanical engineering from the University of Science and Technology of China, Hefei, China, in 2002 and 2007, respectively. Currently, he is a Professor with the State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, China. His research interests include a combination of vibration analysis, signal processing, and metamaterials design for intelligent monitoring, diagnosis, and control in complex machines.



Chao Wang received the B.S. degree from Hefei University of Technology, in 2012, the M.S. degree from the University of Science and Technology of China, in 2015, both in mechanical engineering, the M.S. degree in statistics from the University of Wisconsin-Madison, in 2018, and the Ph.D. degree in industrial and systems engineering from the University of Wisconsin-Madison, in 2019. He is an Assistant Professor with the Department of Industrial and Systems Engineering, University of Iowa. His research interests include statistical modeling, analysis, monitoring, and control for complex systems. He is an Associate Editor of the *Journal of Intelligent Manufacturing*, and a member of INFORMS, IISE, and SME.