Collective Online Learning of Gaussian Processes in Massive Multi-Agent Systems

Trong Nghia Hoang[1,*] and **Quang Minh Hoang**[2,*] and **Kian Hsiang Low**[3] and **Jonathan How**[4]
MIT-IBM Watson AI Lab[1], Carnegie Mellon University[2]
National University of Singapore[3], Massachusetts Institute of Technology[4]
nghiaht@ibm.com[1], qhoang@cs.cmu.edu[2], lowkh@comp.nus.edu.sg[3], jhow@mit.edu[4]

Abstract

This paper presents a novel *Collective Online Learning of Gaussian Processes* (COOL-GP) framework for enabling a massive number of GP inference agents to simultaneously perform (a) efficient online updates of their GP models using their local streaming data with varying correlation structures and (b) decentralized fusion of their resulting online GP models with different learned hyperparameter settings and inducing inputs. To realize this, we exploit the notion of a common encoding structure to encapsulate the local streaming data gathered by any GP inference agent into summary statistics based on our proposed representation, which is amenable to both an efficient online update via an importance sampling trick as well as multi-agent model fusion via decentralized message passing that can exploit sparse connectivity among agents for improving efficiency and enhance the robustness of our framework against transmission loss. We provide a rigorous theoretical analysis of the approximation loss arising from our proposed representation to achieve efficient online updates and model fusion. Empirical evaluations show that COOL-GP is highly effective in model fusion, resilient to information disparity between agents, robust to transmission loss, and can scale to thousands of agents.

1 Introduction

Distributed *Gaussian process* (GP) models (Chen et al. 2013; Deisenroth and Ng 2015; Gal, van der Wilk, and Rasmussen 2014; Hoang, Hoang, and Low 2016; Liu et al. 2018; Low et al. 2015b) are conventionally designed with a server-client paradigm where a server distributes the computational load among parallel machines (i.e., client nodes) to achieve scalability to big data. This paradigm can potentially allow the richness and expressive power of GP models (Rasmussen and Williams 2006) (Section 2) to be exploited by multiple predictive inference agents for distributed inference of the complex latent behavior underlying all their local data. Such a prospect has inspired the recent development of distributed GP fusion algorithms (Allamraju and Chowdhary 2017; Chen, Low, and Tan 2013; Chen et al. 2012; 2015; Ouyang and Low 2018): Essentially, the "client" agents encapsulate their own local data into memory-efficient summary statistics based on a *common* set of *fixed/known* GP hyperparameter settings and *inducing inputs* and communicate them to some "server" agent(s) to be fused into globally consistent summary statistics that are sent back to the "client" agents for GP predictive inference. These distributed GP fusion algorithms inherit the advantage of being adjustably lightweight by restricting the number of inducing inputs (hence the size of the local and global summary statistics) to fit the agents' limited computational and communication capabilities at the expense of predictive accuracy.

However, such algorithms fall short of achieving the truly decentralized GP fusion necessary for scaling up to a massive number of agents grounded in the real world (e.g., traffic sensing, modeling, and prediction by autonomous vehicles cruising in urban road networks (Chen et al. 2015; Low et al. 2015a; Hoang et al. 2014; Min and Wynter 2011; Ouyang et al. 2014; Wang and Papageorgiou 2005; Work et al. 2010), distributed inference on a network of IoTs, surveillance cameras and mobile devices/robots (Kang and Larkin 2016; Natarajan et al. 2014; Hoang et al. 2018b; Zhang et al. 2016)) due to the following critical issues: (a) An obvious limitation is the single point(s) of failure with the server agent(s) whose computational and communication capabilities must be superior and robust (e.g., against transmission loss); (b) different GP inference agents are likely to gather data of varying behaviors and correlation structure from possibly separate localities of the input domain (e.g., spatiotemporal) and would therefore incur considerable information loss due to summarization based on a common set of fixed/known GP hyperparameter settings and inducing inputs, especially when the inducing inputs are few and far from the data (in the correlation sense); and (c) like distributed GP models, distributed GP fusion algorithms implicitly assume a one-time processing of a fixed set of data and would hence repeat the entire fusion process involving all local data gathered by the agents whenever new batches of streaming data arrive, which is prohibitively expensive.

To overcome these limitations, this paper presents a novel *Collective Online Learning of GPs* (COOL-GP) framework for enabling a massive number of agents to simultaneously perform (a) efficient online updates of their GP models using their local streaming data with varying correlation structures and (b) decentralized fusion of their resulting online GP models with different *learned* hyperparameter settings and inducing inputs residing in the original input domain.

[*]T. N. Hoang and Q. M. Hoang contribute equally.

A key technical challenge here lies in designing a representation of the summary statistics for the streaming data gathered by any GP inference agent, which can be both updated and fused efficiently with that for another agent based on possibly different hyperparameter settings and inducing inputs. To realize this, we exploit the notion of a common encoding structure to encapsulate the local streaming data gathered by any GP inference agent into summary statistics based on our proposed representation, which is amenable to both an efficient online update via an importance sampling trick as well as multi-agent model fusion via decentralized message passing that can exploit sparse connectivity among agents for improving efficiency and enhance the robustness of our framework against transmission loss (Section 3). We provide a rigorous theoretical analysis of the approximation loss arising from our proposed representation to achieve efficient online updates and model fusion in Section 4. Finally, we empirically evaluate the performance of COOL-GP on an extensive benchmark comprising both synthetic and real-world datasets with thousands of agents (Section 5).

2 Background and Notations

The *Gaussian process* (GP) model (Rasmussen and Williams 2006) is a rich class of Bayesian nonparametric models that can represent the complex latent behavior underlying the data. Formally, let $\mathcal{X} \subseteq \mathbb{R}^d$ denote an input domain and $\mathrm{f} : \mathcal{X} \to \mathbb{R}$ denote a random latent function mapping each d-dimensional input feature vector $\mathbf{x} \in \mathcal{X}$ to a random latent output $\mathrm{f}(\mathbf{x}) \in \mathbb{R}$ and its noisy measurement $\mathrm{y}(\mathbf{x}) \triangleq \mathrm{f}(\mathbf{x}) + \epsilon(\mathbf{x})$ where $\epsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_\eta^2)$ with noise variance σ_η^2. Let $\{\mathrm{f}(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ denote a GP, that is, for any finite subset of inputs $\mathcal{D} \subseteq \mathcal{X}$, the corresponding column vector of random outputs $\mathbf{f}_\mathcal{D} \triangleq [\mathrm{f}(\mathbf{x})]_{\mathbf{x}\in\mathcal{D}}^\top$ follow a multivariate Gaussian distribution with mean vector $[\mathrm{m}(\mathbf{x})]_{\mathbf{x}\in\mathcal{D}}^\top$ and covariance matrix $\mathbf{K}_{\mathcal{D}\mathcal{D}} \triangleq [\mathrm{k_{ff}}(\mathbf{x},\mathbf{x}')]_{\mathbf{x},\mathbf{x}'\in\mathcal{D}}$ induced, respectively, from user-specified *prior* mean $\mathrm{m}(\mathbf{x}) \triangleq \mathbb{E}[\mathrm{f}(\mathbf{x})]$ (assumed 0 for notational simplicity) and covariance $\mathrm{k_{ff}}(\mathbf{x},\mathbf{x}') \triangleq \mathrm{cov}[\mathrm{f}(\mathbf{x}),\mathrm{f}(\mathbf{x}')]$ for all $\mathbf{x},\mathbf{x}' \in \mathcal{X}$, the latter of which defines the correlation structure of f via a kernel parameterized by θ. Supposing a vector of noisy measurements $\mathbf{y}_\mathcal{D} \triangleq [\mathrm{y}(\mathbf{x})]_{\mathbf{x}\in\mathcal{D}}^\top$ are available for some set of training inputs $\mathcal{D} \subseteq \mathcal{X}$, the GP posterior/predictive belief of $\mathrm{f}(\mathbf{x}_*)$ for any test input $\mathbf{x}_* \in \mathcal{X}$ remains a Gaussian with *posterior* mean $\mu(\mathbf{x}_*) \triangleq \mathbf{k}_*^\top(\mathbf{K}_{\mathcal{D}\mathcal{D}}+\sigma_\eta^2\mathbf{I})^{-1}\mathbf{y}_\mathcal{D}$ and variance $\sigma^2(\mathbf{x}_*) \triangleq \mathrm{k_{ff}}(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K}_{\mathcal{D}\mathcal{D}}+\sigma_\eta^2\mathbf{I})^{-1}\mathbf{k}_*$ where $\mathbf{k}_* \triangleq [\mathrm{k_{ff}}(\mathbf{x}_*,\mathbf{x})]_{\mathbf{x}\in\mathcal{X}}^\top$. A GP predictive belief over input domain $\mathcal{X}$ can thus be represented by $\langle(\mathbf{K}_{\mathcal{D}\mathcal{D}}+\sigma_\eta^2\mathbf{I})^{-1}\mathbf{y}_\mathcal{D}, (\mathbf{K}_{\mathcal{D}\mathcal{D}}+\sigma_\eta^2\mathbf{I})^{-1}, \theta\rangle$, albeit inefficiently due to the inverse of $\mathbf{K}_{\mathcal{D}\mathcal{D}}+\sigma^2\mathbf{I}$ that incurs cubic time and quadratic memory in the size of training data.

To improve its efficiency, a number of sparse GP models (Hoang, Hoang, and Low 2016; Low et al. 2015b; Quiñonero-Candela and Rasmussen 2005; Snelson and Ghahramani 2007; Titsias 2009; Titsias and Lázaro-Gredilla 2013) exploiting the notion of *inducing variables* have been proposed to reduce the incurred time and memory to be linear in the data size. Among them is the notable work of Titsias and Lázaro-Gredilla (2013) that introduced a variational Bayesian sparse GP model capable of learning both the posterior beliefs of inducing variables and hyperparameters and hence the predictive belief by marginalizing them out. Specifically, let $\mathcal{Z} \subseteq \mathbb{R}^d$ denote an input domain and $\mathrm{u} : \mathcal{Z} \to \mathbb{R}$ denote a random latent function mapping each d-dimensional input feature vector $\mathbf{z} \in \mathcal{Z}$ to a random latent output $\mathrm{u}(\mathbf{z}) \in \mathbb{R}$. Let $\{\mathrm{u}(\mathbf{z})\}_{\mathbf{z}\in\mathcal{Z}}$ denote a *standard* GP with zero prior mean and prior covariance $\mathrm{k_{uu}}(\mathbf{z},\mathbf{z}') \triangleq \mathrm{cov}[\mathrm{u}(\mathbf{z}),\mathrm{u}(\mathbf{z}')]$ for all $\mathbf{z},\mathbf{z}' \in \mathcal{Z}$ defined by a special squared exponential kernel $\mathrm{k_{uu}}(\mathbf{z},\mathbf{z}') \triangleq \exp(-0.5(\mathbf{z}-\mathbf{z}')^\top(\mathbf{z}-\mathbf{z}'))$ such that its signal variance and length-scales are set to unity. Supposing $\mathrm{f}(\mathbf{x}) = \sigma_s\mathrm{u}(\mathbf{z} = \mathbf{W}\mathbf{x})$, $\{\mathrm{f}(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ is a GP with prior covariance $\mathrm{k_{ff}}(\mathbf{x},\mathbf{x}') = \sigma_s^2\exp(-0.5(\mathbf{x}-\mathbf{x}')^\top\mathbf{W}^\top\mathbf{W}(\mathbf{x}-\mathbf{x}'))$ for all $\mathbf{x},\mathbf{x}' \in \mathcal{X}$ parameterized by signal variance σ_s^2 and a projection matrix $\mathbf{W}$ from $\mathbf{x} \in \mathcal{X}$ to $\mathbf{z} \in \mathcal{Z}$, and cross covariance $\mathrm{k_{fu}}(\mathbf{x},\mathbf{z}) \triangleq \mathrm{cov}[\mathrm{f}(\mathbf{x}),\mathrm{u}(\mathbf{z})] = \sigma_s\exp(-0.5(\mathbf{W}\mathbf{x}-\mathbf{z})^\top(\mathbf{W}\mathbf{x}-\mathbf{z}))$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$.

Efficiency is then achieved by exploiting a vector $\mathbf{u}_\mathcal{I} = [\mathrm{u}(\mathbf{z})]_{\mathbf{z}\in\mathcal{I}}^\top$ of latent inducing output variables for some small set $\mathcal{I} \subset \mathcal{Z}$ of inducing inputs(i.e., $|\mathcal{I}| \ll |\mathcal{D}|$) to construct summary statistics for training data $\langle\mathcal{D},\mathbf{y}_\mathcal{D}\rangle$. Specifically, to efficiently compute the posterior/predictive belief of $\mathrm{f}(\mathbf{x}_*)$ for any test input $\mathbf{x}_* \in \mathcal{X}$ given $\mathbf{y}_\mathcal{D}$ via marginalization, it involves approximating $\mathrm{p}(\mathbf{u}_\mathcal{I},\mathbf{W}|\mathbf{y}_\mathcal{D})$ by a variational distribution $\mathrm{q}(\mathbf{u}_\mathcal{I},\mathbf{W})$ which is in turn optimized by minimizing the Kullback-Leibler distance $\mathrm{D_{KL}}(\mathrm{q}(\mathbf{f}_\mathcal{D},\mathbf{u}_\mathcal{I},\mathbf{W}), \mathrm{p}(\mathbf{f}_\mathcal{D},\mathbf{u}_\mathcal{I},\mathbf{W}|\mathbf{y}_\mathcal{D}))$ between $\mathrm{q}(\mathbf{f}_\mathcal{D},\mathbf{u}_\mathcal{I},\mathbf{W}) \triangleq \mathrm{p}(\mathbf{f}_\mathcal{D}|\mathbf{u}_\mathcal{I},\mathbf{W})\mathrm{q}(\mathbf{u}_\mathcal{I},\mathbf{W})$ and $\mathrm{p}(\mathbf{f}_\mathcal{D},\mathbf{u}_\mathcal{I},\mathbf{W}|\mathbf{y}_\mathcal{D})$ or, equivalently, maximizing the variational lower bound:

$$\mathrm{L(q)} \triangleq \mathbb{E}_\mathrm{q}[\log \mathrm{p}(\mathbf{y}_\mathcal{D}|\mathbf{f}_\mathcal{D})] - \mathrm{D_{KL}}(\mathrm{q}(\mathbf{u}_\mathcal{I},\mathbf{W})\|\mathrm{p}(\mathbf{u}_\mathcal{I},\mathbf{W}))\ . \tag{1}$$

By further factorizing the prior $\mathrm{p}(\mathbf{u}_\mathcal{I},\mathbf{W}) = \mathrm{p}(\mathbf{u}_\mathcal{I})\mathrm{p}(\mathbf{W})$ such that $\mathrm{p}(\mathbf{u}_\mathcal{I}) \triangleq \mathcal{N}(\mathbf{u}_\mathcal{I}|\mathbf{0},\mathbf{K}_{\mathcal{I}\mathcal{I}})$ with $\mathbf{K}_{\mathcal{I}\mathcal{I}} \triangleq [\mathrm{k_{uu}}(\mathbf{z},\mathbf{z}')]_{\mathbf{z},\mathbf{z}'\in\mathcal{I}}$ and $\mathrm{p}(\mathbf{W})$ is a product of standard normal factors, the following optimal variational distribution results: $\mathrm{q}(\mathbf{W}) = \prod_{i=1}^d\prod_{j=1}^d \mathcal{N}(\mathrm{w}_{ij}|\mu_{ij},\sigma_{ij}^2)$ where $\mathbf{W} \triangleq [\mathrm{w}_{ij}]_{i,j=1,\ldots,d}$ and the variational parameters $\theta \triangleq \{\mu_{ij},\sigma_{ij}\}_{i,j=1,\ldots,d}$ (along with the other hyperparameters such as the signal and noise variances σ_s^2 and σ_η^2) are optimized via gradient ascent of $\mathrm{L(q)}$. Given $\mathrm{q}(\mathbf{W})$, σ_s^2, and σ_η^2, $\mathrm{q}(\mathbf{u}_\mathcal{I})$ is also a Gaussian whose mean vector $\mathbf{m}$ and covariance matrix $\mathbf{S}$ can be analytically derived as

$$\begin{aligned}\mathbf{m} &\triangleq \mathbf{K}_{\mathcal{I}\mathcal{I}}(\sigma_\eta^2\mathbf{K}_{\mathcal{I}\mathcal{I}}+\mathbf{C}_{\mathcal{I}\mathcal{I}})^{-1}\mathbf{C}_{\mathcal{I}\mathcal{D}}\mathbf{y}_\mathcal{D}\ ,\\ \mathbf{S} &\triangleq \sigma_\eta^2\mathbf{K}_{\mathcal{I}\mathcal{I}}(\sigma_\eta^2\mathbf{K}_{\mathcal{I}\mathcal{I}}+\mathbf{C}_{\mathcal{I}\mathcal{I}})^{-1}\mathbf{K}_{\mathcal{I}\mathcal{I}}\end{aligned} \tag{2}$$

where $\mathbf{C}_{\mathcal{I}\mathcal{I}} \triangleq \mathbb{E}_{\mathrm{q}(\mathbf{W})}[\mathbf{K}_{\mathcal{I}\mathcal{D}}\mathbf{K}_{\mathcal{D}\mathcal{I}}]$, $\mathbf{C}_{\mathcal{I}\mathcal{D}} \triangleq \mathbb{E}_{\mathrm{q}(\mathbf{W})}[\mathbf{K}_{\mathcal{I}\mathcal{D}}]$, $\mathbf{K}_{\mathcal{D}\mathcal{I}} \triangleq [\mathrm{k_{fu}}(\mathbf{x},\mathbf{z})]_{\mathbf{x}\in\mathcal{D},\mathbf{z}\in\mathcal{I}}$, and $\mathbf{K}_{\mathcal{I}\mathcal{D}} \triangleq \mathbf{K}_{\mathcal{D}\mathcal{I}}^\top$. Then, (2) yields summary statistics $\langle\mathbf{m},\mathbf{S},\theta\rangle$ (i.e., for training data $\langle\mathcal{D},\mathbf{y}_\mathcal{D}\rangle$) to efficiently represent $\mathrm{p}(\mathbf{u}_\mathcal{I},\mathbf{W}|\mathbf{y}_\mathcal{D}) \approx \mathrm{q}(\mathbf{u}_\mathcal{I},\mathbf{W}) = \mathrm{q}(\mathbf{u}_\mathcal{I})\mathrm{q}(\mathbf{W})$ and hence the predictive belief over $\mathcal{X}$ since it incurs linear time & memory in the data size.

Remark 1 Let $\mathcal{U} \triangleq \{\mathbf{W}^{-1}\mathbf{z}\}_{\mathbf{z}\in\mathcal{I}} \subset \mathcal{X}$. That is, every inducing input $\mathbf{z} \in \mathcal{I}$ can be mapped to a corresponding input $\mathbf{x} \in \mathcal{U} \subset \mathcal{X}$. Optimizing the variational distribution $\mathrm{q}(\mathbf{W})$

has an effect of optimizing the distribution of *mapped* inducing inputs $\mathcal{U}$ in the original input domain $\mathcal{X}$.

Consider the problem of an input domain $\mathcal{X}$ (e.g., urban road network) being *persistently* sampled by a massive system of GP inference agents, each of whom is gathering a continuous stream of data with a possibly different correlation structure from some locality of $\mathcal{X}$ to train its own GP or sparse GP model (Hoang, Hoang, and Low 2016; Low et al. 2015b; Quiñonero-Candela and Rasmussen 2005; Snelson and Ghahramani 2007; Titsias 2009; Titsias and Lázaro-Gredilla 2013) for predictive inference. Such GP models are, however, prohibitively costly to be directly deployed for fusion of streaming data between agents. This is likewise true for the existing distributed GP models and GP fusion algorithms which also suffer from other critical limitations discussed in Section 1. How then can a massive system of GP inference agents effectively and scalably fuse their local streaming data with possibly varying correlation structures? In the next section, we will tackle this challenge by proposing a *Collective Online Learning of GPs* (COOL-GP) framework that exploits a common encoding structure (specifically, inducing inputs $\mathcal{I}$ residing in the input domain $\mathcal{Z}$ of the standard GP) to encapsulate the local streaming data gathered by any GP inference agent into summary statistics based on our proposed representation, which is amenable to both an efficient online update as well as model fusion between a massive number of agents.

3 Collective Online Learning of Gaussian Processes (COOL-GP)

On first thought, one may straightaway consider endowing each agent with an existing online/stochastic variant of a GP or sparse GP model (Bui, Nguyen, and Turner 2017; Cheng and Boots 2016; Csató and Opper 2002; Hensman, Fusi, and Lawrence 2013; Hoang, Hoang, and Low 2015; Xu et al. 2014; Hoang, Hoang, and Low 2017) for training with streaming data scalably. However, such online/stochastic variants are not naturally amenable to model fusion between agents, especially when their GP models are updated to different hyperparameter settings due to their streaming data with possibly varying correlation structures. Some have in fact assumed known hyperparameter settings instead of learning them online. We will now show how the natural parameterization of $\mathrm{q}(\mathbf{u}_\mathcal{I})$ (2) can be exploited for deriving (a) its efficient online update with streaming data via an importance sampling trick (Section 3.1) in order to accommodate an online update of the hyperparameters (Section 3.2) and in turn (b) a decentralized message passing algorithm to perform online GP model fusion between a massive number of agents via the common encoding $\mathcal{I}$ (Sections 3.3 and 3.4).

3.1 Online Update of $\mathrm{q}(\mathbf{u}_\mathcal{I})$

Let $\mathbf{R} \triangleq [\mathbf{R}_\star \ \ \mathbf{R}_\circ] \triangleq [\mathbf{S}^{-1} \ \ \mathbf{S}^{-1}\mathbf{m}]$ denote the natural parameters of $\mathrm{q}(\mathbf{u}_\mathcal{I})$. Then, (2) can be reparameterized in terms of $\mathbf{R}$ to reveal an additive decomposability over a stream of disjoint batches of training data $\langle\mathcal{D}_1, \mathbf{y}_{\mathcal{D}_1}\rangle, \ldots, \langle\mathcal{D}_N, \mathbf{y}_{\mathcal{D}_N}\rangle$ where the set of training inputs $\mathcal{D} \triangleq \bigcup_{n=1}^N \mathcal{D}_n$ such that $\mathcal{D}_n \cap \mathcal{D}_{n'} = \emptyset$ for all $n, n' = 1, \ldots, N$. It follows from (2) that (Hoang et al. 2018a)

$$\begin{aligned}\mathbf{R}_\star &= \mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1} + \sum_{n=1}^N \mathbf{E}_\star^n\ , \quad \mathbf{E}_\star^n \triangleq \frac{1}{\sigma_\eta^2}\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{C}_{\mathcal{I}\mathcal{I}}^n\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\\ \mathbf{R}_\circ &= \sum_{n=1}^N \mathbf{E}_\circ^n\ , \qquad\qquad\ \ \mathbf{E}_\circ^n \triangleq \frac{1}{\sigma_\eta^2}\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{C}_{\mathcal{I}\mathcal{D}_n}\mathbf{y}_{\mathcal{D}_n}\end{aligned} \tag{3}$$

where $\mathbf{C}_{\mathcal{I}\mathcal{I}}^n \triangleq \mathbb{E}_{\mathrm{q}(\mathbf{W})}[\mathbf{K}_{\mathcal{I}\mathcal{D}_n}\mathbf{K}_{\mathcal{D}_n\mathcal{I}}]$. Supposing $\mathrm{q}(\mathbf{W})$ is fixed, (3) reveals an efficient online update of $\mathrm{q}(\mathbf{u}_\mathcal{I})$ as each update incurs linear time in the size $|\mathcal{D}_n|$ of a data batch only. Specifically, let $\mathbf{R}^{n-1} \triangleq [\mathbf{R}_\star^{n-1} \ \ \mathbf{R}_\circ^{n-1}]$ denote the natural parameters of $\mathrm{q}(\mathbf{u}_\mathcal{I})$ after being updated by a stream of $n-1$ data batches $\langle\mathcal{D}_1, \mathbf{y}_{\mathcal{D}_1}\rangle, \ldots, \langle\mathcal{D}_{n-1}, \mathbf{y}_{\mathcal{D}_{n-1}}\rangle$ received previously and $\mathbf{E}^n \triangleq [\mathbf{E}_\star^n \ \ \mathbf{E}_\circ^n]$ denote the summary statistics for an incoming data batch $\langle\mathcal{D}_n, \mathbf{y}_{\mathcal{D}_n}\rangle$. It follows directly from (3) that

$$\mathbf{R}^n = \mathbf{R}^{n-1} + \mathbf{E}^n\ . \tag{4}$$

Updating $\mathbf{R}^{n-1}$ to $\mathbf{R}^n$ (4) is efficient as it only requires evaluating $\mathbf{E}^n$ which incurs linear time in the size $|\mathcal{D}_n|$ of the incoming data batch. However, if $\mathrm{q}(\mathbf{W})$ is also updated by every incoming data batch, then $\mathbf{C}_{\mathcal{I}\mathcal{I}}^1, \ldots, \mathbf{C}_{\mathcal{I}\mathcal{I}}^{n-1}$ and $\mathbf{C}_{\mathcal{I}\mathcal{D}_1}, \ldots, \mathbf{C}_{\mathcal{I}\mathcal{D}_{n-1}}$ have to be recomputed with respect to the updated $\mathrm{q}(\mathbf{W})$ and hence incur linear time in the size of the accumulating data batches, which becomes prohibitively expensive when data streams in at a high velocity. To sidestep the inefficiency from such recomputations, we exploit an importance sampling trick to approximate $\mathbf{C}_{\mathcal{I}\mathcal{I}}^n \approx \widehat{\mathbf{C}}_{\mathcal{I}\mathcal{I}}^n$ and $\mathbf{C}_{\mathcal{I}\mathcal{D}_n} \approx \widehat{\mathbf{C}}_{\mathcal{I}\mathcal{D}_n}$ with M i.i.d. samples $\mathbf{W}_1, \ldots, \mathbf{W}_M$ drawn from the prior $\mathrm{p}(\mathbf{W})$:

$$\begin{aligned}\widehat{\mathbf{C}}_{\mathcal{I}\mathcal{I}}^n &\triangleq \frac{1}{M}\sum_{m=1}^M \frac{\mathrm{q}(\mathbf{W}_m)}{\mathrm{p}(\mathbf{W}_m)}\mathbf{K}_{\mathcal{I}\mathcal{D}_n}^m\mathbf{K}_{\mathcal{D}_n\mathcal{I}}^m\ ,\\ \widehat{\mathbf{C}}_{\mathcal{I}\mathcal{D}_n} &\triangleq \frac{1}{M}\sum_{m=1}^M \frac{\mathrm{q}(\mathbf{W}_m)}{\mathrm{p}(\mathbf{W}_m)}\mathbf{K}_{\mathcal{I}\mathcal{D}_n}^m\end{aligned} \tag{5}$$

where $\mathbf{K}_{\mathcal{I}\mathcal{D}_n}^m$ and $\mathbf{K}_{\mathcal{D}_n\mathcal{I}}^m$ denote, respectively, $\mathbf{K}_{\mathcal{I}\mathcal{D}_n}$ and $\mathbf{K}_{\mathcal{D}_n\mathcal{I}}$ parameterized by sample $\mathbf{W}_m$. Using (5), we can then approximate $\mathbf{E}^n \approx \widehat{\mathbf{E}}^n \triangleq [\widehat{\mathbf{E}}_\star^n \ \ \widehat{\mathbf{E}}_\circ^n]$ where

$$\widehat{\mathbf{E}}_\star^n \triangleq \frac{1}{\sigma_\eta^2}\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\widehat{\mathbf{C}}_{\mathcal{I}\mathcal{I}}^n\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\ , \quad \widehat{\mathbf{E}}_\circ^n \triangleq \frac{1}{\sigma_\eta^2}\mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1}\widehat{\mathbf{C}}_{\mathcal{I}\mathcal{D}_n}\mathbf{y}_{\mathcal{D}_n}. \tag{6}$$

Finally, the online update of $\mathrm{q}(\mathbf{u}_\mathcal{I})$ in (4) can be approximated by $\widehat{\mathbf{R}}^n = \widehat{\mathbf{R}}^{n-1} + \widehat{\mathbf{E}}^n$. To see why this is efficient, since $\mathbf{W}_1, \ldots, \mathbf{W}_M$ can be generated *a priori*, the $\mathbf{K}_{\mathcal{I}\mathcal{D}_n}^m\mathbf{K}_{\mathcal{D}_n\mathcal{I}}^m$ and $\mathbf{K}_{\mathcal{I}\mathcal{D}_n}^m\mathbf{y}_{\mathcal{D}_n}$ terms for $m = 1, \ldots, M$ in (5) and (6) can be computed only once in $\mathcal{O}(M|\mathcal{D}_n|)$ time (by treating $|\mathcal{I}|$ as a constant) for every incoming data batch $\langle\mathcal{D}_n, \mathbf{y}_{\mathcal{D}_n}\rangle$ and cached for use in recomputing $\widehat{\mathbf{E}}^n$ (6) efficiently in $\mathcal{O}(M)$ time whenever $\mathrm{q}(\mathbf{W})$ is updated by each subsequent incoming data batch. As a result, the update to $\widehat{\mathbf{R}}^N$ after receiving the incoming data batch $\langle\mathcal{D}_N, \mathbf{y}_{\mathcal{D}_N}\rangle$ will incur a total of $\mathcal{O}(MN + M|\mathcal{D}_N|)$ time due to recomputing $\widehat{\mathbf{R}}^{N-1} = \sum_{n=1}^{N-1}\widehat{\mathbf{E}}^n$ and evaluating $\widehat{\mathbf{E}}^N$. As will be shown

in Lemma 1 (Section 4), an appropriate choice of M guarantees an arbitrarily small approximation loss, which is made possible by our choices of $\widehat{\mathbf{C}}^n_{\mathcal{I}\mathcal{I}}$ and $\widehat{\mathbf{C}}_{\mathcal{I}\mathcal{D}_n}$ in (5) that are unbiased estimates of $\mathbf{C}^n_{\mathcal{I}\mathcal{I}}$ and $\mathbf{C}_{\mathcal{I}\mathcal{D}_n}$.

3.2 Online Update of $\mathrm{q}(\mathbf{W})$

Naively, the online update of $\mathrm{q}(\mathbf{W})$ can be achieved via gradient ascent $\theta \leftarrow \theta + \partial \mathrm{L}(\mathrm{q})/\partial\theta$. This is however inefficient as the exact gradient $\partial \mathrm{L}(\mathrm{q})/\partial\theta$ needs to be recomputed with respect to the accumulating data batches and the updated $\mathrm{q}(\mathbf{u}_\mathcal{I})$. To overcome this issue, we first derive an additive decomposability of the variational lower bound $\mathrm{L}(\mathrm{q})$ (1) over disjoint batches of data $\langle \mathcal{D}_1, \mathbf{y}_{\mathcal{D}_1}\rangle, \ldots, \langle \mathcal{D}_{N'}, \mathbf{y}_{\mathcal{D}_{N'}}\rangle$, as shown in (Hoang et al. 2018a):

$$\mathrm{L}(\mathrm{q}) = \textstyle\sum_{n=1}^{N'} \mathrm{L}_{\mathcal{D}_n}(\mathrm{q}) - \mathrm{D}_{\mathrm{KL}}(\mathrm{q}(\mathbf{u}_\mathcal{I}, \mathbf{W})\|\mathrm{p}(\mathbf{u}_\mathcal{I}, \mathbf{W}))$$

with $\mathrm{L}_{\mathcal{D}_n}(\mathrm{q}) \triangleq \mathbb{E}_{\mathrm{q}(\mathbf{u}_\mathcal{I},\mathbf{W})}[\mathbb{E}_{\mathrm{p}(\mathbf{f}_{\mathcal{D}_n}|\mathbf{u}_\mathcal{I},\mathbf{W})}[\log \mathrm{p}(\mathbf{y}_{\mathcal{D}_n}|\mathbf{f}_{\mathcal{D}_n})]]$. Suppose that an agent has received a stream of data batches sampled in a uniformly random order from the training data with the most recent incoming data batch denoted by $\langle \mathcal{D}_*, \mathbf{y}_{\mathcal{D}_*}\rangle$. Using only $\langle \mathcal{D}_*, \mathbf{y}_{\mathcal{D}_*}\rangle$, we can construct an unbiased stochastic gradient $\partial\widehat{\mathrm{L}}(\mathrm{q})/\partial\theta$ of $\mathrm{L}(\mathrm{q})$:

$$\frac{\partial\widehat{\mathrm{L}}(\mathrm{q})}{\partial\theta} = N'\frac{\partial \mathrm{L}_{\mathcal{D}_*}(\mathrm{q})}{\partial\theta} - \frac{\partial}{\partial\theta}\mathrm{D}_{\mathrm{KL}}(\mathrm{q}(\mathbf{u}_\mathcal{I}, \mathbf{W})\|\mathrm{p}(\mathbf{u}_\mathcal{I}, \mathbf{W}))$$

which satisfies $\mathbb{E}_{\langle \mathcal{D}_*, \mathbf{y}_{\mathcal{D}_*}\rangle}[\partial\widehat{\mathrm{L}}(\mathrm{q})/\partial\theta] = \partial \mathrm{L}(\mathrm{q})/\partial\theta$ (Hoang et al. 2018a) and its evaluation incurs linear time in the size $|\mathcal{D}_*|$ of the data batch instead of that of the accumulating data batches. The resulting stochastic gradient ascent is guaranteed to converge to a local optimum given an appropriate schedule of learning rates (Robbins and Monro 1951). Note that the signal and noise variance hyperparameters can be updated in a similar manner by stochastic gradient ascent.

Remark 2 Though the stochastic gradient $\partial\widehat{\mathrm{L}}(\mathrm{q})/\partial\theta$ is evaluated using only $\langle \mathcal{D}_*, \mathbf{y}_{\mathcal{D}_*}\rangle$, it depends on the natural parameters of the updated $\mathrm{q}(\mathbf{u}_\mathcal{I})$ that are summary statistics for the stream of data batches received previously (Section 3.1).

3.3 Model Fusion between Pairwise Agents

In this subsection, we will describe a novel model fusion mechanism between pairwise agents to exchange and fuse their online sparse GP models of possibly different *learned* hyperparameter settings (Sections 3.1 and 3.2). Then, we will generalize such a mechanism for online GP model fusion between a massive number of agents in Section 3.4. Suppose that two agents a and b have performed online updates of their corresponding variational distributions $\mathrm{q}_\mathrm{a}(\mathbf{u}_\mathcal{I}, \mathbf{W}_a) = \mathrm{q}_\mathrm{a}(\mathbf{u}_\mathcal{I})\mathrm{q}_\mathrm{a}(\mathbf{W}_a) \approx \mathrm{p}(\mathbf{u}_\mathcal{I}, \mathbf{W}_a|\mathbf{y}_{\mathcal{D}_a})$ and $\mathrm{q}_\mathrm{b}(\mathbf{u}_\mathcal{I}, \mathbf{W}_b) = \mathrm{q}_\mathrm{b}(\mathbf{u}_\mathcal{I})\mathrm{q}_\mathrm{b}(\mathbf{W}_b) \approx \mathrm{p}(\mathbf{u}_\mathcal{I}, \mathbf{W}_b|\mathbf{y}_{\mathcal{D}_b})$ with their respective streaming data $\langle \mathcal{D}_a, \mathbf{y}_{\mathcal{D}_a}\rangle$ and $\langle \mathcal{D}_b, \mathbf{y}_{\mathcal{D}_b}\rangle$ (Sections 3.1 and 3.2). Since $\mathbf{W}_a$ and $\mathbf{W}_b$ will be marginalized out for predictive inference, we can focus on approximating $\mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_a}, \mathbf{y}_{\mathcal{D}_b})$ directly. To achieve this, note that

$$\mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_a}, \mathbf{y}_{\mathcal{D}_b}) \propto \frac{\mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_a})\mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_b})}{\mathrm{p}(\mathbf{u}_\mathcal{I})} \approx \frac{\mathrm{q}_a(\mathbf{u}_\mathcal{I})\mathrm{q}_b(\mathbf{u}_\mathcal{I})}{\mathrm{p}(\mathbf{u}_\mathcal{I})} \tag{7}$$

where the first step is derived in (Hoang et al. 2018a). (7) implies that $\mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_a}, \mathbf{y}_{\mathcal{D}_b})$ can be approximated by fusing the summary statistics (i.e., for $\langle \mathcal{D}_a, \mathbf{y}_{\mathcal{D}_a}\rangle$ and $\langle \mathcal{D}_b, \mathbf{y}_{\mathcal{D}_b}\rangle$) that represent $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$ and $\mathrm{q}_b(\mathbf{u}_\mathcal{I})$: $\mathrm{q}_{ab}(\mathbf{u}_\mathcal{I}) \propto \mathrm{q}_a(\mathbf{u}_\mathcal{I})\mathrm{q}_b(\mathbf{u}_\mathcal{I})/\mathrm{p}(\mathbf{u}_\mathcal{I})$. Specifically, let $\mathrm{q}_a(\mathbf{u}_\mathcal{I}) = \mathcal{N}(\mathbf{u}_\mathcal{I}|\mathbf{m}_a, \mathbf{S}_a)$ and $\mathrm{q}_b(\mathbf{u}_\mathcal{I}) = \mathcal{N}(\mathbf{u}_\mathcal{I}|\mathbf{m}_b, \mathbf{S}_b)$ where the parameters $\mathbf{m}_a, \mathbf{m}_b, \mathbf{S}_a$, and $\mathbf{S}_b$ are computed using (2). Then, it can be derived (Hoang et al. 2018a) that $\mathrm{q}_{ab}(\mathbf{u}_\mathcal{I}) = \mathcal{N}(\mathbf{u}_\mathcal{I}|\mathbf{m}_{ab}, \mathbf{S}_{ab})$ where

$$\begin{aligned}\mathbf{S}_{ab} &\triangleq (\mathbf{S}_a^{-1} + \mathbf{S}_b^{-1} - \mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1})^{-1}\ ,\\ \mathbf{m}_{ab} &\triangleq \mathbf{S}_{ab}(\mathbf{S}_a^{-1}\mathbf{m}_a + \mathbf{S}_b^{-1}\mathbf{m}_b)\ .\end{aligned} \tag{8}$$

Let $\mathbf{R}_{ab}$, $\mathbf{R}_a$, $\mathbf{R}_b$, and $\mathbf{R}_0$ denote the natural parameters of $\mathrm{q}_{ab}(\mathbf{u}_\mathcal{I})$, $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$, $\mathrm{q}_b(\mathbf{u}_\mathcal{I})$, and $\mathrm{p}(\mathbf{u}_\mathcal{I})$, respectively (Section 3.1). It follows that (8) can be rewritten concisely as

$$\mathbf{R}_{ab} = \mathbf{R}_a + \mathbf{R}_b - \mathbf{R}_0\ . \tag{9}$$

In practice, since maintaining the exact natural parameters $\mathbf{R}_a$ and $\mathbf{R}_b$ is inefficient for their respective online updates of $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$ and $\mathrm{q}_b(\mathbf{u}_\mathcal{I})$, we instead use their efficient counterparts $\widehat{\mathbf{R}}_a$ and $\widehat{\mathbf{R}}_b$ (Section 3.1) to approximate $\mathbf{R}_{ab}$ by

$$\widehat{\mathbf{R}}_{ab} = \widehat{\mathbf{R}}_a + \widehat{\mathbf{R}}_b - \mathbf{R}_0\ . \tag{10}$$

The time incurred by this fusion (10) depends only on the constant number $|\mathcal{I}|$ of inducing inputs and is thus independent of the total size $|\mathcal{D}_a| + |\mathcal{D}_b|$ of the streaming data.

Remark 3 Though $\mathrm{q}_a(\mathbf{W}_a)$ and $\mathrm{q}_b(\mathbf{W}_b)$ are not explicitly fused, they will be updated, respectively, by agents a and b using the fused $\mathrm{q}_{ab}(\mathbf{u}_\mathcal{I})$, as explained in Remark 2. This consequently improves their distributions of mapped inducing inputs in the original input domain $\mathcal{X}$ (Remark 1), which in turn reduces information loss arising from encapsulating their streaming data into summary statistics (Section 3.1) for fusion (10) via the common encoding structure $\mathcal{I}$.

3.4 Decentralized Message Passing for Multi-Agent Model Fusion

This subsection generalizes the model fusion mechanism in (10) to that beyond two agents. Specifically, consider a massive system of A GP inference agents, each of whom has performed an online update of $\mathrm{q}_a(\mathbf{u}_\mathcal{I}) \approx \mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_a})$ with streaming data $\langle \mathcal{D}_a, \mathbf{y}_{\mathcal{D}_a}\rangle$ to obtain the exact natural parameters $\mathbf{R}_a$ for $a = 1, \ldots, A$ (Section 3.1). It can be shown (Hoang et al. 2018a) that the exact natural parameters $\mathbf{R}_\mathcal{F}$ of their fused $\mathrm{q}_\mathcal{F}(\mathbf{u}_\mathcal{I}) \approx \mathrm{p}(\mathbf{u}_\mathcal{I}|\mathbf{y}_{\mathcal{D}_1}, \ldots, \mathbf{y}_{\mathcal{D}_A})$ is

$$\mathbf{R}_\mathcal{F} = \textstyle\sum_{a=1}^{A} \mathbf{R}_a - (A-1)\mathbf{R}_0\ . \tag{11}$$

Naively, $\mathbf{R}_\mathcal{F}$ (11) can be approximated by

$$\widehat{\mathbf{R}}_\mathcal{F} = \textstyle\sum_{a=1}^{A} \widehat{\mathbf{R}}_a - (A-1)\mathbf{R}_0 \tag{12}$$

using $\widehat{\mathbf{R}}_a$ obtained from the efficient online update of $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$ by agent a for $a = 1, \ldots, A$ (Section 3.1). However, computing (12) requires either direct communication between every pairwise agents or a central server through which all agents coordinate their communications. The former implies a fully connected network which is not desirable

in large spatial input domains (e.g., urban road networks, ocean phenomena) where the agents have limited communication range (Chen et al. 2015; Ouyang and Low 2018) while the latter introduces a single point of failure (Section 1). To circumvent these issues, we will describe a decentralized message passing algorithm to compute (12) that allows agents to exchange their summary statistics as messages among one another within their communication range.

Let $\mathbf{M}_{ab}^{t+1}$ denote the message that agent a sends to agent b (i.e., within its communication range) in iteration $t+1$, which fuses the summary statistics of agent a with that received from the other agents in the previous t iterations of transmission. This must not include the summary statistics of agent b to avoid aggregating duplicates of information. Thus, $\mathbf{M}_{ab}^{t+1}$ essentially fuses the summary statistics of all agents (except that of agent b) whose messages can reach agent a within t iterations of direct transmission.

As such, $\mathbf{M}_{ab}^{t+1}$ can be recursively computed by aggregating only messages received from those agents within the communication range of agent a (except that of agent b) in the previous iteration t:

$$\mathbf{M}_{ab}^{t+1} = \widehat{\mathbf{R}}_a + \textstyle\sum_{k\in\mathcal{N}(a)\setminus\{b\}}(\mathbf{M}_{ka}^{t} - \mathbf{R}_0)$$

where $\mathcal{N}(a)$ denotes the set of agents in the communication range of agent a and the subtraction of $\mathbf{R}_0$ from $\mathbf{M}_{ka}^{t}$ prevents aggregating multiple copies of the natural parameters $\mathbf{R}_0$ of the prior $\mathrm{p}(\mathbf{u}_\mathcal{I})$, which has already been fused into $\widehat{\mathbf{R}}_a$. In iteration $t=0$, the message only contains the summary statistics of agent a (i.e., $\mathbf{M}_{ab}^{0} = \widehat{\mathbf{R}}_a$) since no message from other agents can reach agent a in 0 iteration of transmission. Upon convergence in iteration $t = t_{\max}$[1], every agent a can aggregate the received messages to yield the same globally consistent summary statistics:

$$\widehat{\mathbf{R}}_\mathcal{F} = \widehat{\mathbf{R}}_a + \textstyle\sum_{k\in\mathcal{N}(a)}(\mathbf{M}_{ka}^{t_{\max}} - \mathbf{R}_0)$$

where the repeated subtraction of $\mathbf{R}_0$ from $\mathbf{M}_{ka}^{t_{\max}}$ is to prevent aggregating multiple copies of $\mathbf{R}_0$ into $\widehat{\mathbf{R}}_\mathcal{F}$.

4 Theoretical Analysis

In this section, we will prove that the approximate globally consistent summary statistics $\widehat{\mathbf{R}}_\mathcal{F}$ is theoretically guaranteed to be arbitrarily close to the exact counterpart $\mathbf{R}_\mathcal{F}$ with high confidence (Theorem 1). In particular, we are interested to bound the approximation loss of $\widehat{\mathbf{R}}_\mathcal{F}$ relative to $\mathbf{R}_\mathcal{F}$ in terms of the numbers M of samples of projection matrices drawn from the prior $\mathrm{p}(\mathbf{W})$ (Section 3.1), A of agents, and $|\mathcal{I}|$ of inducing inputs. To do this, we will first probabilistically bound the approximation loss of $\widehat{\mathbf{R}}_a$ relative to $\mathbf{R}_a$ for any agent a due to our choice of approximate representation that results from our importance sampling trick to achieve efficient online update of $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$ (Section 3.1):

[1]For a tree-topology network, our message passing algorithm converges to the exact optimum after $t_{\max}$ iterations where $t_{\max}$ is the tree's diameter. The agents can run a decentralized minimum spanning tree algorithm to eliminate redundant connections with high latency to guarantee that their connection topology is a tree.

Lemma 1 (Representation Loss) *Given* $\epsilon > 0$ *and* $\delta \in (0,1)$, $\|\mathbf{R}_a - \widehat{\mathbf{R}}_a\| \leq \epsilon$ *with probability of at least* $1-\delta$ *by setting* $M \triangleq \mathcal{O}((|\mathcal{I}|^2/\epsilon^2)\log(|\mathcal{I}|/\delta))$.

Its proof is in (Hoang et al. 2018a). Using Lemma 1, we can now bound the approximation loss of $\widehat{\mathbf{R}}_\mathcal{F}$ relative to $\mathbf{R}_\mathcal{F}$ due to multi-agent model fusion (Section 3.4):

Theorem 1 (Fusion Loss) *Given* $\epsilon > 0$ *and* $\delta \in (0,1)$, $\|\mathbf{R}_\mathcal{F} - \widehat{\mathbf{R}}_\mathcal{F}\| \leq \epsilon$ *with probability of at least* $1-\delta$ *by setting* $M \triangleq \mathcal{O}((|\mathcal{I}|^2 A^2/\epsilon^2)\log(|\mathcal{I}|A/\delta))$.

Its proof is in (Hoang et al. 2018a).

Remark 4 The above results imply that both the representation and fusion losses can be guaranteed to be arbitrarily small with high probability by choosing a sufficiently large number M of samples of projection matrices drawn from the prior $\mathrm{p}(\mathbf{W})$ (Section 3.1). Theorem 1 also indicates that the required number M of samples has to grow quadratically in the numbers A of agents and $|\mathcal{I}|$ of inducing inputs in order to guarantee the same fusion quality at the expense of the time efficiency of the online updates of $\mathrm{q}_a(\mathbf{u}_\mathcal{I})$ by agents $a = 1,\ldots,A$ (Section 3.1).

5 Experiments and Discussion

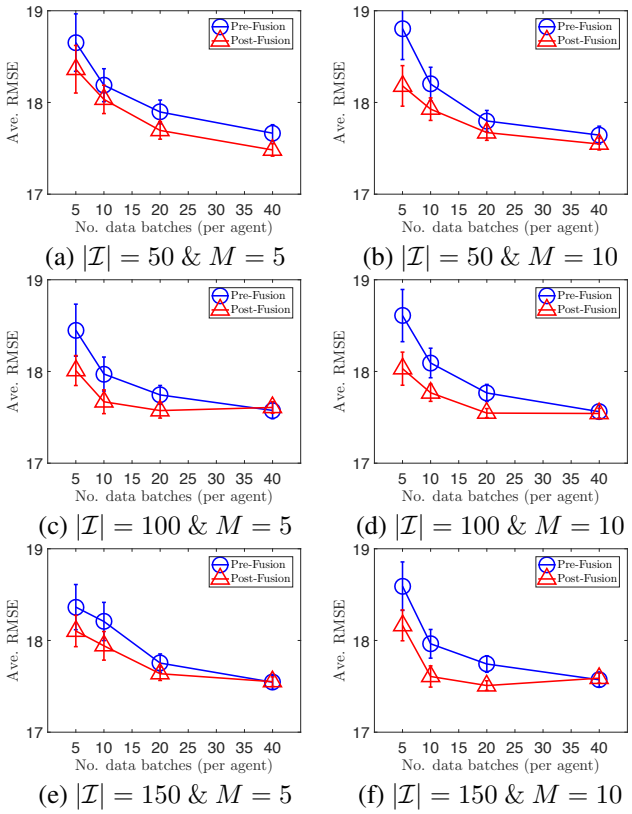
This section empirically evaluates the fusion performance of our COOL-GP framework, its resilience to information disparity between agents, and robustness to transmission loss on both synthetic and real-world experimental domains:

(a) The SYNTHETIC domain features two streaming datasets generated by $\mathrm{f}(\mathbf{x}) \triangleq \mathrm{u}(\mathbf{W}\mathbf{x})$ and $\mathrm{f}'(\mathbf{x}) \triangleq \mathrm{u}(\mathbf{W}'\mathbf{x})$ with the respective projection matrices $\mathbf{W}$ and $\mathbf{W}'$ where the common random function $\mathrm{u}(\mathbf{z})$ is sampled from a standard GP (Section 2). Each streaming dataset of size 8000 comprises 200 batches of 6-dimensional data of size 40. A separate test data of size 4000 is generated from f and f'.

(b) The AIRLINE domain (Hensman, Fusi, and Lawrence 2013; Hoang, Hoang, and Low 2015) features an air transportation delay phenomenon that generates streaming data of size 600000 comprising 30000 batches of 20 observations each. Each observation has a 8-dimensional input feature vector containing the information log of a commercial flight and a corresponding output recording its time delay (min). The system comprises 1000 agents, each of whom is evaluated using a separate test data of size 10000.

(c) The AIMPEAK domain (Hoang, Hoang, and Low 2016) features a traffic phenomenon which took place over an urban road network comprising 775 road segments. 10000 data batches are then generated from the traffic phenomenon and streamed in random order to a system of 100 GP inference agents. Each observation has a 5-dimensional input feature vector and a corresponding output of the traffic speed (km/h). The predictive performance of each agent is then evaluated using a separate test data of size 2000.

In all experiments, each data batch arrives sequentially in a random order and is dispatched to a random agent. This simulates collective online learning scenarios with streaming data where agents gather one data batch at a time. We report the averaged predictive performance before and after

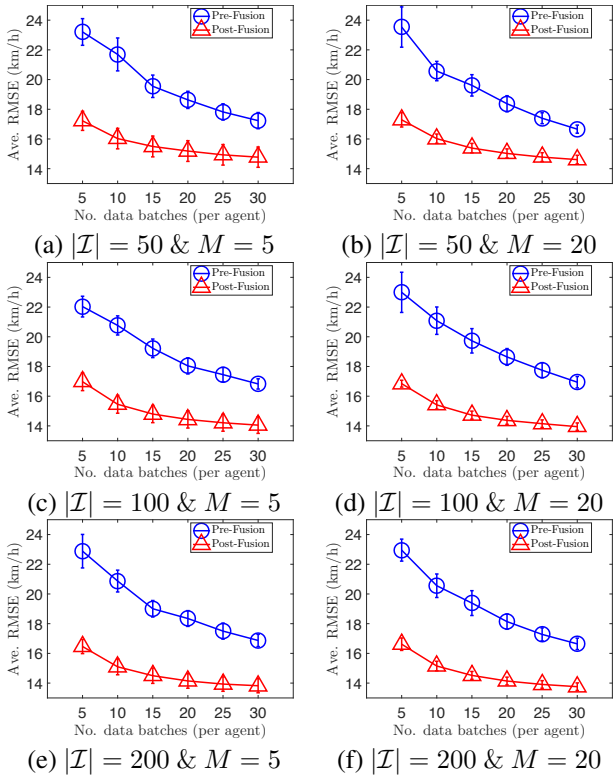


(a) $|\mathcal{I}| = 50$ & $M = 5$ (b) $|\mathcal{I}| = 50$ & $M = 10$

(c) $|\mathcal{I}| = 100$ & $M = 5$ (d) $|\mathcal{I}| = 100$ & $M = 10$

(e) $|\mathcal{I}| = 150$ & $M = 5$ (f) $|\mathcal{I}| = 150$ & $M = 10$

Figure 1: Graphs of averaged pre- and post-fusion performance vs. the no. of data batches dispatched to 2 agents with varying numbers $|\mathcal{I}|$ of and M.



(a) $|\mathcal{I}| = 50$ & $M = 5$ (b) $|\mathcal{I}| = 50$ & $M = 20$

(c) $|\mathcal{I}| = 100$ & $M = 5$ (d) $|\mathcal{I}| = 100$ & $M = 20$

(e) $|\mathcal{I}| = 200$ & $M = 5$ (f) $|\mathcal{I}| = 200$ & $M = 20$

Figure 2: Graphs of averaged pre- and post-fusion performance vs. no. of data batches of 100 agents gathering data from the same traffic phenomenon with varying $|\mathcal{I}|$ and M.

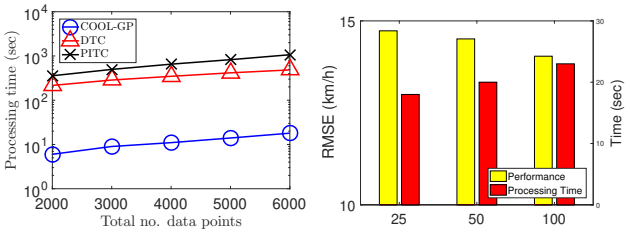
fusion by the agents vs. the number of streamed data batches to demonstrate the fusion performance of our COOL-GP framework in such distributed streaming data settings.

Fig. 1 reports the results of our COOL-GP framework in a collective online learning scenario where two agents fuse their online sparse GP models of two correlated, synthetic phenomena to improve their averaged performance on test instances from their input localities. Fig. 2 further reports the performance of COOL-GP in a real-world traffic monitoring application deployed on a large, decentralized network consisting of 100 agents. Both of these cases demonstrate the effectiveness of COOL-GP fusion on the averaged predictive accuracy vs. varying numbers of streamed data batches for different numbers $|\mathcal{I}|$ of inducing inputs and M of projection matrix samples (Section 3.1). Across all configurations, a consistent pattern can be observed: (a) post-fusion predictions exhibit significant performance gain as compared to pre-fusion predictions, and (b) the performance gap gradually reduces with more gathered data, which suggests a diminishing marginal gain of model fusion.

Fig. 3a demonstrates the computational benefit of COOL-GP (100 agents, 30 data batches per agent) by comparing the total of its averaged incurred time (per agent) against the running time of DTC and PITC (Quiñonero-Candela and Rasmussen 2005), which correspond to the sequential versions of dDTC and dPITC. The result shows that for 6000 data points, DTC and PITC incur 445 and 1145 seconds respectively. On the other hand, COOL-GP's averaged processing time per agent is 13 seconds, which is 34.23 and 88.07 times faster than DTC and PITC, respectively. Fig. 3b further shows that the predictive performance and processing time (per agent) of COOL-GP, respectively, increases and decreases with more participating agents.

Fig. 4 visualizes a comprehensive collection of individual performance profiles of 1000 agents in the AIRLINE domain; each profile is represented by a pair of pre- and post-fusion RMSEs. The result shows that with more gathered data, clusters of performance profiles (i.e., each cluster is visualized by a colored point cloud) gradually migrate towards regions with superior pre- and post-fusion accuracy. The migration distance, however, reduces rapidly in later stages of data gathering, which is consistent with the previous observation on the diminishing return of model fusion. Interestingly, it can also be observed that within each cluster, the performance profiles exhibit high variance for pre-fusion and low variance for post-fusion performance, which suggests that agents are able to achieve post-fusion consensus within small range of variation (i.e., fusion stability).

We also investigate an interesting case study of model fusion between agents allocated with different amounts of data in the AIMPEAK traffic domain. Specifically, Fig. 5a reports the performance of two agents A1 (fixed amount of



(a) (b)

Figure 3: Graphs of (a) processing time vs. total no. of data points of DTC, PITC and COOL-GP; and (b) RMSE vs. no. of agents (25, 50 and 100) of COOL-GP (30 batches per agent and $M = 20$) tested on AIMPEAK dataset.

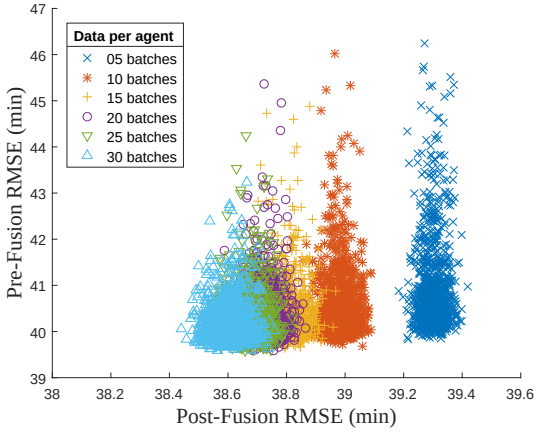
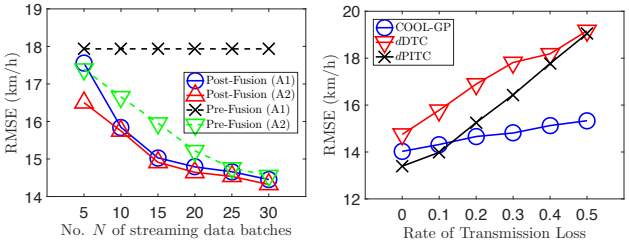


Figure 4: Graphs of individual performance profiles (pre- vs. post-fusion RMSE) of a 1000-agent system collectively learning using our COOL-GP framework in the AIRLINE domain (Hensman, Fusi, and Lawrence 2013).

data) and A2 (continuous stream of data). Without fusion, A1 fails to update its model and improve its performance as expected whereas A2 still exhibits gain in performance as it receives more data. With fusion, however, the performance of A1 is brought close to that of A2 and far exceeds its original accuracy. More interestingly, it can be observed that the performance of A2 also marginally improves upon fusion with a conservative A1 who never gathers new data to update its model. This demonstrates that COOL-GP greatly benefits agents with lesser learning capabilities and, at the same time, mildly improves the performance of those with better capabilities (i.e., resilience to information disparity).

Finally, in the traffic domain (i.e., AIMPEAK), we present another interesting case study that features a collective online learning scenario among 100 agents where each message transmission (or local statistics in the cases of distributed GPs such as dDTC (Gal, van der Wilk, and Rasmussen 2014) and dPITC (Hoang, Hoang, and Low 2016)) may not reach its destination with a certain probability. The averaged post-fusion performance are plotted against the rate of transmission loss to demonstrate the robustness of COOL-GP to transmission loss. Fig. 5b shows that as trans-



(a) (b)

Figure 5: Graphs of (a) pre- and post-fusion individual performance of two agents with different learning capabilities, and (b) post-fusion performance of COOL-GP in comparison to those of state-of-the-art distributed GPs (e.g., dDTC and dPITC) vs. rate of transmission loss in AIMPEAK.

mission losses occur more frequently, the averaged performance of COOL-GP agents degrades more gracefully than those of the state-of-the-art[2] distributed dDTC and dPITC frameworks which employ a central server to coordinate the communications between agents. This is expected since both dDTC and dPITC require every agent to successfully transmit its summary statistics directly to a single master server. Failing to achieve this immediately leads to irrecoverable information loss. In contrast, COOL-GP allows each agent to exchange its summary statistics with multiple agents within its communication range (Section 3.4), thus lowering the risk of losing information.

6 Conclusion

This paper describes a novel COOL-GP framework for enabling a massive number of agents to simultaneously perform (a) efficient online updates of their GP models using their local streaming data with varying correlation structures and (b) decentralized fusion of their resulting online GP models with different learned hyperparameters and inducing inputs. We exploit the notion of a common encoding structure to encapsulate the local streaming data gathered by any GP inference agent into summary statistics, which is amenable to both efficient online update as well as multi-agent model fusion that exploits sparse connectivity among agents for improving efficiency and enhance the robustness of our framework against transmission loss (Section 3). We also provide a rigorous theoretical analysis of the approximation loss arising from our proposed representation to achieve efficient online updates and model fusion (Section 4). Empirical evaluations on real-world datasets show that our framework performs efficiently on various settings and can scale to thousands of agents (Section 5).

Acknowledgments. This research was funded in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-2-0181 and by ONR under the BRC N00014-17-1-2072. It was also supported in part by Singapore Ministry of Education Academic Research Fund Tier 2, MOE2016-T2-2-156.

[2]We do not compare with dPIC (Hoang, Hoang, and Low 2016) as it needs to store local data and is not suitable for online learning.

References

Allamraju, R., and Chowdhary, G. 2017. Communication efficient decentralized Gaussian process fusion for multi-UAS path planning. In *Proc. ACC*.

Bui, T. D.; Nguyen, C. V.; and Turner, R. E. 2017. Streaming sparse Gaussian process approximations. In *Proc. NIPS*.

Chen, J.; Low, K. H.; Tan, C. K.-Y.; Oran, A.; Jaillet, P.; Dolan, J. M.; and Sukhatme, G. S. 2012. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, 163–173.

Chen, J.; Cao, N.; Low, K. H.; Ouyang, R.; Tan, C. K.-Y.; and Jaillet, P. 2013. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, 152–161.

Chen, J.; Low, K. H.; Jaillet, P.; and Yao, Y. 2015. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Transactions on Automation Science and Engineering* 12(3):901–921.

Chen, J.; Low, K. H.; and Tan, C. K.-Y. 2013. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*.

Cheng, C.-A., and Boots, B. 2016. Incremental variational sparse Gaussian process regression. In *Proc. NIPS*.

Csató, L., and Opper, M. 2002. Sparse online Gaussian processes. *Neural Computation* 14(3):641–669.

Deisenroth, M. P., and Ng, J. W. 2015. Distributed Gaussian processes. In *Proc. ICML*.

Gal, Y.; van der Wilk, M.; and Rasmussen, C. 2014. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Proc. NIPS*.

Hensman, J.; Fusi, N.; and Lawrence, N. D. 2013. Gaussian processes for big data. In *Proc. UAI*, 282–290.

Hoang, T. N.; Low, K. H.; Jaillet, P.; and Kankanhalli, M. 2014. Nonmyopic ϵ-Bayes-optimal active learning of Gaussian processes. In *Proc. ICML*, 739–747.

Hoang, T. N.; Hoang, Q. M.; Low, K. H.; and How, J. P. 2018a. Collective online learning of Gaussian processes in massive multi-agent systems. arXiv:1805.09266.

Hoang, T. N.; Xiao, Y.; Sivakumar, K.; Amato, C.; and How, J. 2018b. Near-optimal adversarial policy switching for decentralized asynchronous multi-agent systems. In *Proc. ICRA*.

Hoang, T. N.; Hoang, Q. M.; and Low, K. H. 2015. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, 569–578.

Hoang, T. N.; Hoang, Q. M.; and Low, K. H. 2016. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, 382–391.

Hoang, Q. M.; Hoang, T. N.; and Low, K. H. 2017. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, 2007–2014.

Kang, J. J., and Larkin, H. 2016. Inference of personal sensors in internet of things. *International Journal of Information, Communication Technology and Applications* 2:1.

Liu, H.; Cai, J.; Wang, Y.; and Ong, Y.-S. 2018. Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In *Proc. ICML*.

Low, K. H.; Chen, J.; Hoang, T. N.; Xu, N.; and Jaillet, P. 2015a. Recent advances in scaling up Gaussian process predictive models for large spatiotemporal data. In *Proc. DyDESS*, 167–181.

Low, K. H.; Yu, J.; Chen, J.; and Jaillet, P. 2015b. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*.

Min, W., and Wynter, L. 2011. Real-time road traffic prediction with spatio-temporal correlations. *Transport. Res. C-Emer.* 19(4):606–616.

Natarajan, P.; Hoang, T. N.; Wong, Y.; Low, K. H.; and Kankanhalli, M. S. 2014. Scalable decision-theoretic coordination and control for real-time active multi-camera surveillance. In *Proc. ICDSC*, 115–120.

Ouyang, R., and Low, K. H. 2018. Gaussian process decentralized data fusion meets transfer learning in large-scale distributed cooperative perception. In *Proc. AAAI*.

Ouyang, R.; Low, K. H.; Chen, J.; and Jaillet, P. 2014. Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena. In *Proc. AAMAS*.

Quiñonero-Candela, J., and Rasmussen, C. E. 2005. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* 6:1939–1959.

Rasmussen, C. E., and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *Ann. Math. Statist.* 22(3):400–407.

Snelson, E. L., and Ghahramani, Z. 2007. Local and global sparse Gaussian process approximation. In *Proc. AISTATS*.

Titsias, M. K., and Lázaro-Gredilla, M. 2013. Variational inference for Mahalanobis distance metrics in Gaussian process regression. In *Proc. NIPS*.

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

Wang, Y., and Papageorgiou, M. 2005. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transport. Res. B-Meth.* 39(2):141–167.

Work, D. B.; Blandin, S.; Tossavainen, O.; Piccoli, B.; and Bayen, A. 2010. A traffic model for velocity data assimilation. *AMRX* 2010(1):1–35.

Xu, N.; Low, K. H.; Chen, J.; Lim, K. K.; and Özgül, E. B. 2014. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, 2585–2592.

Zhang, Y.; Hoang, T. N.; Low, K. H.; and Kankanhalli, M. 2016. Near-optimal active learning of multi-output Gaussian processes. In *Proc. AAAI*, 2351–2357.