Collective Model Fusion for Multiple Black-Box Experts

Quang Minh Hoang [*1] **Trong Nghia Hoang** [*2] **Bryan Kian Hsiang Low** [3] **Carl Kingsford** [1]

Abstract

Model fusion is a fundamental problem in collective machine learning (ML) where independent experts with heterogeneous learning architectures are required to combine expertise to improve predictive performance. This is particularly challenging in information-sensitive domains where experts do not have access to each other's internal architecture and local data. This paper presents the first collective model fusion framework for multiple experts with heterogeneous black-box architectures. The proposed method will enable this by addressing the key issues of how black-box experts interact to understand the predictive behaviors of one another; how these understandings can be represented and shared efficiently among themselves; and how the shared understandings can be combined to generate high-quality consensus prediction. The performance of the resulting framework is analyzed theoretically and demonstrated empirically on several datasets.

1. Introduction

Practical scenarios that involve learning in complex environments often require the collaboration of multiple experts operating concurrently on different sub-domains. Motivated under this context, collective learning (Hoang et al., 2019) is an emerging study of a distributed paradigm where each local expert learns independently from its data and exchange knowledge with others to achieve better performance.

Existing collective learning literature (Gifford, 2009; Yahya et al., 2017; Hoang et al., 2019), however, usually assumes perfect clarity of local expert models, which entails fully transparent model architectures and publicly accessible local data used to train these experts. To facilitate model fusion, local experts are further expected to have employed a homogeneous design with limited freedom in their choices of parameters. Despite enabling collective learning, these restrictive assumptions have imposed a rigidity on the algorithmic level that is generally undesirable for practical purposes. For example, applications learning from private medical records are often prohibited from publicizing sensitive patient information; model architectures in confidential domains such as financial forecasting are preferably kept undisclosed to guard against adversarial attacks. As such, it is unrealistic for a collaboration scheme among these experts to presume prior understanding of their behaviors, much less subjecting them to conceptual homogeneity.

Another central issue of collective learning, as pointed out by Hoang et al. (2019), is the computational and communication bottleneck arising from having one single or a few central servers to coordinate the collective agents. In practice, such a centralized collective architecture also often leads to having undesirable choke points of failure in the system. To avoid this, Hoang et al. (2019) proposed moving towards a decentralized learning paradigm where collective agents only exchange information with their neighbor in a communication network. The collective learning framework of Hoang et al. (2019), however, is not designed to work with heterogeneous, black-box models because it assumes perfect knowledge and availability of the agent models. This is often not true in practice, especially for information-sensitive domains where collaboration occurs in a transactional basis (i.e., short-term collaboration) and may not get prioritized over data and model privacy.

This paper thus presents a novel collective learning platform for black-box fusion that addresses the following challenges: (a) performing fusion without access to the black-box training data and architectures, (b) performing fusion when the black-box models are not permanently available, and (c) avoiding centralized bottlenecks and risk of failure for large-scale fusion with numerous black-box experts.

To achieve this, we first develop a collective fusion paradigm that allows black-box experts to interact and learn the predictive behaviors of one another, which are then succinctly encoded into information summaries with constant memory-footprint for efficient communication and assimilation. A decentralized communication algorithm is further developed

[*]Equal contribution [1]Carnegie Mellon University [2]MIT-IBM Watson AI Lab, IBM Research Cambridge [3]National University of Singapore. Correspondence to: Quang Minh Hoang <qhoang@andrew.cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

to regulate the propagation flow of these local summaries to optimize the expected improvement rate of the experts while guaranteeing that they reach a consensus upon convergence.

In particular, we have made the following contributions:

1. A gradient fusion scheme for performing light-weight collective inference among learning agents which assume the corresponding black-box models are always available for querying (Section 4.1).

2. A surrogate fusion scheme which transfuses the predictive behavior of black-box experts onto imitator models of choice to allow persistent collective inference among learning agents even when black-box models are no longer available for querying (Section 4.2).

3. A decentralized communication algorithm for the learning agents implementing the above fusion schemes to propagate among themselves their prediction and parameter gradients, thus removing the operational bottleneck (Section 4.3).

4. A formal theoretical analysis to guarantee the predictive disagreement rate between the imitating surrogate and the black-box model (Section 5), thus asserting the imitation quality of our surrogate fusion scheme in Section 4.2.

5. An extensive empirical study that demonstrates the efficiency of our black-box model fusion paradigm on several real-world datasets with promising results (Section 6).

To the best of our knowledge, this work is the first to propose a collective model fusion framework for black-box models.

2. Related Work

Collective learning is a new study arising from the traditional context of distributed machine learning (ML) where data analytics is provided and engineered in the cloud (Chen et al., 2013b; Low et al., 2015; Deisenroth & Ng, 2015; Hoang et al., 2016; McMahan et al., 2017; Liu et al., 2018). Distributed ML typically requires broadcasting data statistics from local experts to a central server for processing. This, however, exposes a single choke point for failure as all local experts have to constantly communicate with the cloud to operate successfully, which places a severe stress on the central server's communication bandwidth.

Some works in this direction (Allamraju & Chowdhary, 2017; Chen et al., 2012; 2013c; Natarajan et al., 2014; Ruofei & Low, 2018; Yurochkin et al., 2019) have attempted to alleviate this bottleneck by enforcing an identical knowledge representation across all experts to ease communication and model aggregation among themselves. This is, however, not desirable in practical information-sensitive domains (e.g., health-care analytics with private medical records) where experts cannot communicate in advance to agree on the same model architecture. In contrast, allowing heterogeneity in their architectures avoids these problems, but causes difficulty in communication among experts.

Aiming for the best of both worlds, a successful collective model fusion framework should therefore allow heterogeneous experts with different black-box learning architectures to represent, communicate and combine their expertise efficiently to harness the full potential of collective intelligence without exposing sensitive information to others. Devising this framework is our key contribution in this research, which is formulated in Section 3 and addressed in Section 4. Its theoretical analysis and empirical evaluation are also provided in Section 5 and Section 6, respectively.

3. Problem Formulation

Let $\mathcal{B} \triangleq \{p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)\}_{i=1}^m$ denotes a collection of probabilistic black-box predictors. Each predictor $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$ is parameterized by (a) a non-linear function $\ell_i(\mathbf{x}, \mathbf{D}_i)$ acting as sufficient statistics of the input $\mathbf{x}$ given local data $\mathbf{D}_i$; and (b) a set of characterizing parameters $\boldsymbol{\omega}_i$ that accounts for its predictive uncertainty. Both $\ell_i(\mathbf{x}, \mathbf{D}_i)$ and $\boldsymbol{\omega}_i$ were trained *a priori* on a separate set of data $\mathbf{D}_i \triangleq \{(\mathbf{x}_i^{(t)}, y_i^{(t)})\}_{t=1}^{n_i}$.

For example, one such predictive distribution is a Gaussian distribution centered around the sufficient statistics. Our developed framework in Section 4 works with any choices of $\{p_i\}_{i=1}^m$ and also does not require access to their internal architecture.

Thus, given such predictive distribution, the local prediction of an input $\mathbf{x}$ is determined as the output candidate with highest probability score $y_i(\mathbf{x}) = \arg\max_y p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$. However, since $\ell_i(\mathbf{x}, \mathbf{D}_i)$ and $\boldsymbol{\omega}_i$ were estimated using only a single subset of data $\mathbf{D}_i$, the resulting predictive distribution might not be able to elicit accurate prediction.

To improve their predictive performance without centralizing data to build a global model, a popular approach is to construct a product-of-expert (PoE) (Deisenroth & Ng, 2015) model that assumes shared characterizing parameters, i.e. $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_2 = \ldots = \boldsymbol{\omega}_m = \boldsymbol{\omega}$, among local experts, which allows them to communicate and aggregate their voting scores for each output candidate in order to determine the most likely output $y_{\mathcal{B}}(\mathbf{x})$ conditioned on the entire set of data $\mathbf{D} \triangleq \{\mathbf{D}_i\}_{i=1}^m$,

$$\begin{aligned} y_{\mathcal{B}}(\mathbf{x}) &\triangleq \arg\max_y \prod_{i=1}^m p_i\left(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}\right) \\ &= \arg\max_y \sum_{i=1}^m \log p_i\left(y \mid \ell_i\left(\mathbf{x}, \mathbf{D}_i\right); \boldsymbol{\omega}\right) , \end{aligned} \quad (1)$$

where $\log p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$ can be viewed as the individual voting score of predictive distribution p_i for output

candidate y[1]. The shared parameters $\boldsymbol{\omega}$ can be learned via maximizing the model evidence of PoE:

$$\boldsymbol{\omega}^* \triangleq \arg\max_{\boldsymbol{\omega}} \sum_{i=1}^{m} \sum_{t=1}^{n_i} \log p_i \left(y_i^{(t)} \mid \ell_i(\mathbf{x}_i^{(t)}, \mathbf{D}_i); \boldsymbol{\omega} \right) \quad (2)$$

This approach, however, requires full access to each predictor's model architecture (i.e., $\ell_i(\mathbf{x}, \mathbf{D}_i)$ and $\boldsymbol{\omega}$) and training data $\mathbf{D}_i$, which is not possible in many practical settings as discussed in Section 1. Furthermore, the PoE model of Deisenroth & Ng (2015) also requires a central server to coordinate communication among local experts, which inadvertently introduces a single choke point of operation failure as well as a severe strain on the server's computation and communication resources. These are in fact common issues with the majority of literature on distributed machine learning (ML), which were previously discussed in Section 2.

To resolve these issues, this paper transforms the original distributed learning task into a collective learning problem similar to Hoang et al. (2019). This is, however, highly nontrivial since our learning scenarios regard each expert model as a black box whereas the work of Hoang et al. (2019) explicitly imposes a homogeneous learning architecture, i.e., sparse Gaussian processes regression (Titsias & Lázaro-Gredilla, 2013; Quiñonero-Candela & Rasmussen, 2005; Hoang et al., 2015; 2016; 2017; 2018), on all experts to forge a superficial communication medium among themselves.

This violates two desiderata of collective learning that we argued for in Section 1: (a) Experts do not have to reveal their model architectures to others to exchange information, thus avoid being exposed to security risk (e.g., leaking sensitive information); and (b) experts do not have to conform to a homogeneous learning architecture to combine their expertise (i.e., model fusion), thus eliminating the need of a common model architecture.

Our work hence generalizes this work in two directions:

1. To forge a mutual understanding between two black-box experts without having them disclosed their model architectures, we develop a black-box fusion paradigm for two distinct scenarios: (a) a light-weight gradient fusion scheme for performing **ephemeral** collective inference on-the-fly directly with black-box models (Section 4.1) as opposed to (b) an imitating algorithm that transfers and combines the expertise of black-box models onto **persistent** *de novo* models of choice (Section 4.2), which can be fused efficiently.

2. To facilitate fusion among experts with heterogeneous learning architectures, we develop a gradient-based communication algorithm, which allows the experts to exchange pertinent information regarding their predictive knowledge via propagating gradients of the predictive probabilities with respect to corresponding prediction estimates produced by their choices of surrogates for one another. Such information can be computed without forcing the experts to align within a common model architecture, thus allowing greater diversity among them, which is important for the fused model to achieve better performance. We also show that the developed communication algorithm can be made decentralized while still allowing experts to reach a consensus in prediction upon convergence (Section 5).

[1]A local predictive distribution decides the most likely output candidate using its individual voting score $y_i(\mathbf{x}) = \arg\max_y \log p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$, which is computed based solely on its local data $\mathbf{D}_i$ and is therefore less accurate.

4. Collective Black-Box Fusion

In this section, we present the aforementioned **ephemeral** and **persistent** black-box fusion algorithms. In a centralized setting, these fusion algorithms are achieved by each expert publishing its parameter and/or prediction gradients to a master server, which aggregates them to compute the corresponding global gradients. The master server will broadcast these gradients back to each expert, which will use those to update both its local parameters and prediction.

These centralized fusion schemes will be detailed in Section 4.1 and Section 4.2, respectively. This paper, however, argues against such a centralized setting (Section 1) and will therefore extend these fusion schemes to enable decentralized gradient fusion without requiring a master server to coordinate communication (Section 4.3).

4.1. Collective Inference via Gradient Aggregation

This section will present our *Collective Inference via Gradient Aggregation* (CIGAR) algorithm for light-weight fusion. In particular, let $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$ denote the local expert parameterized by $\ell_i(\mathbf{x}, \mathbf{D}_i)$ and $\boldsymbol{\omega}_i$.

Then, let $\nabla_y \log p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$ denote its local prediction gradient. Thus, assuming access to this local gradient, one can compute the global prediction gradient via

$$\nabla_y \log p\left(y \mid \mathbf{x}\right) \quad = \quad \sum_{i=1}^{m} \nabla_y \log p_i\left(y \mid \ell_i; \boldsymbol{\omega}_i\right) \ . \quad (3)$$

The local prediction estimate $y_i = y_i^{(t)}$ of each expert can then be updated via gradient ascent using the above global prediction gradient to approach a prediction consensus,

$$y_i^{(t+1)} \quad = \quad y_i^{(t)} + \eta \nabla_y \log p\left(y^{(t)} \mid \mathbf{x}\right) \ , \quad (4)$$

with a sufficiently small learning rate $\eta > 0$. However, since we do not have access to $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$ internal architecture, its prediction gradient cannot be computed analytically. Instead, we employ a random gradient estimation technique to approximate $\nabla_y \log p_i(y \mid \ell_i; \boldsymbol{\omega}_i)$ via

$$\nabla_y \log p_i(y|\mathbf{x};\boldsymbol{\omega}_i) \simeq \mathbb{E}_z\left[\frac{z}{\lambda_y}\log\left(\frac{p_i(y+\lambda_y z|\mathbf{x};\boldsymbol{\omega}_i)}{p_i(y|\mathbf{x};\boldsymbol{\omega}_i)}\right)\right] = \mathbb{E}_z\left[\nabla_y^{(z)}\log p_i(y|\mathbf{x},\boldsymbol{\omega}_i)\right] \quad (5)$$

where $\lambda_y > 0$ is a sufficiently small value, $z \sim \mathbb{N}(0,1)$ and

$$\nabla_y^{(z)}\log p_i(y|\mathbf{x},\boldsymbol{\omega}) \triangleq \frac{z}{\lambda_y}\log\left(\frac{p_i(y+\lambda_y z|\mathbf{x};\boldsymbol{\omega})}{p_i(y|\mathbf{x};\boldsymbol{\omega})}\right) \quad (6)$$

That is, $\nabla_y^{(z)}\log p_i(y|\ell_i,\boldsymbol{\omega}_i)$ can effectively be used as unbiased stochastic gradients that guarantee convergence to a local maximizer of the global model's prediction likelihood. Note that in this section, we abbreviate $\ell_i(\mathbf{x},\mathbf{D}_i)$ by ℓ_i to avoid cluttering the notation. The black-box notation can then be rewritten succinctly as $p_i(y|\ell_i,\boldsymbol{\omega}_i)$. The rationality behind this random estimation is explained in Appendix A.

4.2. Collective Learning via Black-Box Imitation

This section will present our *Collective Learning via Black-box Imitation* (COLBI) algorithm for transferring expertise from black-box models to white-box surrogates and performing collective inference via fusing these surrogates. To elaborate, we first develop an imitation algorithm to translate the predictive behavior of each black-box $p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$ into a local surrogate $q_i(y|\mathbf{x};\mathbf{w})$. We further parameterize these surrogates by global set of parameters $\mathbf{w}$, which allow them to combine their expertise for better performance. To achieve this, we develop a collective learning algorithm to fuse these white-boxes into a global, persistent[2] surrogate.

Building Surrogate Model for Black-Box Expert

For ease of notation, the subscript i which indexes the expert is dropped since there is no need to differentiate between different experts in the context of this sub-section.

Naively, to fit the surrogate $q(y \mid \mathbf{x};\mathbf{w})$ to the predictive pattern of $p(y \mid \ell(\mathbf{x},\mathbf{D});\boldsymbol{\omega})$, we assume access to a finite set of unlabeled data $\{\mathbf{x}^{(t)}\}_{t=1}^n \neq \mathbf{D}$, which can be queried for the expert's black-box predictors $\left\{\ell(\mathbf{x}^{(t)},\mathbf{D})\right\}_{t=1}^n$. Exploiting this information, we can fit the surrogate to match the predictive pattern of the black-box expert at those queried data points via minimizing the following objective:

$$\widehat{\mathbf{L}}(\mathbf{w}) = \frac{1}{n}\sum_{t=1}^n \mathbf{D}_{\mathrm{KL}}\Big(q(y|\mathbf{x}^{(t)};\mathbf{w})\|p(y|\ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})\Big)$$

with respect to $\mathbf{w}$. This can be achieved by choosing $q(y \mid \mathbf{x};\mathbf{w})$ such that each KL term in the above objective can be expressed as an analytic, convex function of $\mathbf{w}$ (though this choice might restrict the expressiveness of the surrogate model) whose exact, optimal solution can be found efficiently using any of the existing convex optimizer software.

[2]Once constructed, this persistent model will require no further query from black-box experts to perform prediction.

This, however, requires us to have access to the explicit architecture of $p(y \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})$, which violates the black-box setting in Section 3. To overcome that, we instead re-parameterize $\mathbf{D}_{\mathrm{KL}}(q(y \mid \mathbf{x}^{(t)};\mathbf{w})\|p(y \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega}))$ as

$$\mathbf{D}_{\mathrm{KL}}\left(q\|p\right) = \mathbb{E}_{y\sim q(y|\mathbf{x}^{(t)};\mathbf{w})}\left[\log\frac{q(y|\mathbf{x}^{(t)};\mathbf{w})}{p(y|\ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})}\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathbb{N}(\mathbf{0},\mathbf{I})}\left[\log\frac{q(h(\boldsymbol{\epsilon};\mathbf{u})|\mathbf{x}^{(t)};\mathbf{w})}{p(h(\boldsymbol{\epsilon};\mathbf{u})|\ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})}\right]$$

where $h(\boldsymbol{\epsilon};\mathbf{u})$ is a transport function (parameterized by $\mathbf{u}$) that transforms a noise distribution $\boldsymbol{\epsilon} \sim \mathbb{N}(\mathbf{0},\mathbf{I})$ into the surrogate's predictive distribution $q(y|\mathbf{x}^{(t)};\mathbf{w})$. This parameterization thus makes the distribution that underlies the expectation independent of the parameters $\mathbf{w}$ and $\mathbf{u}$. Exploiting this re-parameterization, we can further derive unbiased stochastic gradients for $\mathbf{w}$ and $\mathbf{u}$ via sampling $\boldsymbol{\epsilon} \sim \mathbb{N}(\mathbf{0},\mathbf{I})$:

$$\nabla_{\mathbf{w}}^{(\boldsymbol{\epsilon})}\widehat{\mathbf{L}} = \frac{1}{n}\sum_{t=1}^n \nabla_{\mathbf{w}}^{(\boldsymbol{\epsilon})}\log\frac{q\left(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \mathbf{x}^{(t)};\mathbf{w}\right)}{p\left(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega}\right)}, \quad (7)$$

$$\nabla_{\mathbf{u}}^{(\boldsymbol{\epsilon})}\widehat{\mathbf{L}} = \frac{1}{n}\sum_{t=1}^n \nabla_{\mathbf{u}}^{(\boldsymbol{\epsilon})}\log\frac{q\left(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \mathbf{x}^{(t)};\mathbf{w}\right)}{p\left(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega}\right)}, \quad (8)$$

which can be used to optimize $\mathbf{w}$ and $\mathbf{u}$ numerically via coordinate gradient descent. The above stochastic gradient computation therefore only requires queried feedback from $p(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})$ at $(h(\boldsymbol{\epsilon};\mathbf{u}),\mathbf{x}^{(t)})$ for a particular random sample of $\boldsymbol{\epsilon}$. The queried feedback can then be used in conjunction with the randomized gradient estimation technique (see Appendix A) to compute an unbiased stochastic estimate of $\nabla_{\mathbf{u}}\log p(h(\boldsymbol{\epsilon};\mathbf{u}) \mid \ell(\mathbf{x}^{(t)},\mathbf{D});\boldsymbol{\omega})$, which is necessary to compute the above equations.

Remark 1. In practice, one often characterizes $h(\epsilon;\mathbf{u})$ as a deep neural network (DNN)[3] where $\mathbf{u}$ represents its internal weight. If $q(y \mid \mathbf{x};\mathbf{w})$ is chosen such that the objective is convex in $\mathbf{w}$ and $\mathbf{u}$ can be found such that the distribution of the mapping $y = h(\boldsymbol{\epsilon};\mathbf{u})$ with $\boldsymbol{\epsilon} \sim \mathbb{N}(\mathbf{0},\mathbf{I})$ coincides with the surrogate's predictive distribution $q(y \mid \mathbf{x};\mathbf{w})$ then the above coordinate gradient descent will converge to the optimal parameter $\mathbf{w} = \widehat{\mathbf{w}}$.

It can then be shown later in Section 5 (see Theorem 1) that with high probability, the chance over a random choice of $\mathbf{x} \sim \mathcal{P}(\mathbf{x})$ (with $\mathcal{P}(\mathbf{x})$ denotes an arbitrary input distribution) for which $q(y \mid \mathbf{x};\widehat{\mathbf{w}})$'s prediction disagrees or deviates significantly from that of $p(y \mid \ell(\mathbf{x},\mathbf{D});\boldsymbol{\omega})$ is small, thus validating the quality of the resulting surrogate model.

Surrogate Fusion via Gradient Aggregation

[3]When $h(\boldsymbol{\epsilon};\mathbf{u})$ is characterized as a DNN, the above gradient update equation can be implicitly implemented as a back-propagation parameter update process.

Let $q(y \mid \mathbf{x}; \mathbf{w}) \triangleq \prod_{j=1}^{m} q_j(y \mid \mathbf{x}; \mathbf{w})$ denote the PoE model of the resulting surrogates and let q_j's current estimate of $\mathbf{w}$ be $\mathbf{w}_j$. We can compute its global gradient by aggregating the local gradients of all surrogates evaluated at $\mathbf{w}_j$:

$$\nabla_{\mathbf{w}} \log q\left(y \mid \mathbf{x}; \mathbf{w}_j\right) \quad = \quad \sum_{i=1}^{m} \nabla_{\mathbf{w}} \log q_i\left(y \mid \mathbf{x}; \mathbf{w}_j\right) \quad (9)$$

Thus, at iteration $t+1$, each surrogate $q_j(y|\mathbf{x}; \mathbf{w}_j)$ can then update their current estimate $\mathbf{w}_j^{(t)}$ via

$$\mathbf{w}_j^{(t+1)} \quad = \quad \mathbf{w}_j^{(t)} + \eta \nabla_{\mathbf{w}} \log q\left(y \mid \mathbf{x}; \mathbf{w}_j^{(t+1)}\right) , \quad (10)$$

where η is a sufficiently small learning rate and the superscript (t) indexes the update iteration. Note that by choosing the surrogates to be log concave, the above update rule is guaranteed to approach the true optimum asymptotically.

Both fusion-update scheme for $\mathbf{w}$ and y in Section 4.1 share a similar centralized design, in which gradient aggregation happens at a master server and updates occur locally after receiving gradient broadcasts from such server. This is inefficient in practice because for each local estimate of the variable being optimized, the master server needs to receive local gradients from all available experts evaluated at that point. When the number of experts grows, these schemes will place severe strains on the central server both in terms of computing resource and communication bandwidth. In the next section, we detail a decentralized fusion scheme where gradient aggregation only occurs within local neighborhoods of experts and are propagated throughout the communication network via overlapping neighborhood.

4.3. Decentralized Learning and Inference

Due to limited space, we only detail our decentralized surrogate fusion algorithm in this section. The decentralized collective inference is very similar in spirit and hence, omitted from the main text of this paper.

To proceed, let us now relax the assumption that there exists a centralized server where the surrogates or local experts can pool their local gradients. In this case, each surrogate model $q_i(y|\mathbf{x}; \mathbf{w}_i)$ needs to maintain and update its own parameter estimate $\mathbf{w}_i$ by computing Eq. (9) via peer-to-peer communication. This can be achieved by noting that the form of the approximate gradient in Eq. (9) decomposes additively across local surrogates, which can be essentially cast as a distributed sum problem and can be solved efficiently using the decentralized algorithm detailed in Hoang et al. (2019).

Given that there are m surrogate models corresponding to m black-box experts, this amounts to solving m problems concurrently. In particular, at gradient update iteration $t+1$, the experts exchange information in d rounds where d is the diameter of the tree topology that characterizes the direct communication link between them.

In particular, let $\mathcal{M}_{ij}^{\mathbf{w},h+1}(k)$ denote the message from q_i to q_j about q_i's estimation of the global parameter gradient evaluated at q_k's current parameter estimation $\mathbf{w} = \mathbf{w}_k$. Then, at round $h+1$ of message passing, we compute the message $\mathcal{M}_{ij}^{\mathbf{w},h+1} \triangleq \left\{\mathcal{M}_{ij}^{\mathbf{w},h+1}(k)\right\}_{k=1}^{m}$ to be sent from q_i to q_j where:

$$\begin{aligned}\mathcal{M}_{ij}^{\mathbf{w},h+1}(k) \quad &\triangleq \nabla_{\mathbf{w}} \log q_i(y \mid \mathbf{x}; \mathbf{w})\Big|_{\mathbf{w}=\mathbf{w}_k} \\ &+ \left(\textstyle\sum_{\ell \in \mathcal{A}_i} \mathcal{M}_{\ell i}^{\mathbf{w},h}(k)\right) - \mathcal{M}_{ji}^{\mathbf{w},h}(k)\end{aligned} \quad (11)$$

where $\mathcal{A}_i$ is the communication neighborhood of q_i which includes q_i. Given an appropriate choice of $\{\mathcal{A}_i\}_{i=1}^{m}$, it can be shown that the above message passing will converge after a finite number of iterations. Upon convergence, it can be shown that the global gradient for iteration t in Eq. (9) can be re-constructed for each expert using the received messages at the last iteration d (Hoang et al., 2019):

$$\begin{aligned}\nabla_{\mathbf{w}} \log p(y \mid \mathbf{x})\Big|_{\mathbf{w}=\mathbf{w}_i} \quad &= \nabla_y \log q_i(y \mid \mathbf{x}; \mathbf{w}_i)\Big|_{\mathbf{w}=\mathbf{w}_i} \\ &+ \mathcal{M}_{\ell i}^{\mathbf{w},d}(i) \quad \forall \ell \in \mathcal{A}_i, \ell \neq i\end{aligned} \quad (12)$$

Each expert can then use gradient ascent to update its own estimate. Again, if we choose all surrogates to be log concave, then all experts will converge to the global optimum regardless of their (different) initial estimates.

5. Theoretical Analysis

This section analyzes the imitation quality of the surrogate algorithm in Section 4.2. In particular, we show that under mild assumptions, the probability that the predictions of the imitating surrogate $q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$ (Section 4.2) and its corresponding black-box expert $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}_i)$ disagree is bounded with respect to several quantities of interest.

To proceed, we first put forward the following assumptions:

Assumption 1**.** The surrogate models $\{q_i(y \mid \mathbf{x}; \mathbf{w}_i)\}_{i=1}^{m}$ can be selected such that given the optimized $\{\widehat{\mathbf{w}}_i\}_{i=1}^{m}$ (Section 4.2), there exists $0 < \eta < +\infty$ for which $\mathbf{D}_{\mathrm{KL}}(q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i) \| p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})) \leq \eta$ for any $\mathbf{x}$.

Assumption 2**.** The global minimizer of the surrogate fitting objective function $\widehat{\mathbf{L}}(\mathbf{w})$ can be computed exactly using the re-parameterization technique presented in Section 4.2.

Thus, assuming the surrogate fitting objective function can be optimized globally (Assumption 2), we can expect the prediction of the resulting surrogate to agree with the prediction of its corresponding black-box. This intuition will be formalized in the remaining of this section. To continue, we need the following definitions:

Definition 1 (Surrogate Robustness) *Let* $\mathbf{x}$ *denote the test input and let* $\{q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)\}_{i=1}^{m}$ *denote the learned*

surrogates parameterized by $\{\widehat{\mathbf{w}}_i\}_{i=1}^m$ *via minimizing the objective in Section 4.1. The prediction robustness* $\alpha > 0$ *of the surrogates with respect to* $\mathbf{x}$ *is defined as*

$$\alpha \triangleq \frac{1}{2} \min_{1 \leq i \leq m} \Big(q_i(y_i \mid \mathbf{x}; \widehat{\mathbf{w}}_i) - \max_{y \neq y_i} q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i) \Big) \quad (13)$$

where we define $y_i \triangleq \max_y q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$.

Using the above definition, we can establish the following result that specifies the sufficient condition for the surrogate to yield the same prediction as its black-box counterpart:

Lemma 1 *For an arbitrary test input* $\mathbf{x}$*, let* α *denote the trained surrogates' predictive robustness (see Definition 1) with respect to* $\mathbf{x}$*. If the following is satisfied,*

$$\mathbf{D}_{\mathrm{KL}}\Big(q_i(y|\mathbf{x}; \widehat{\mathbf{w}}_i) \| p_i(y|\ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}) \Big) \leq \frac{\alpha^2}{2 \log 2} , \quad (14)$$

$q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$ *agrees with* $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$ *on the optimal prediction for input* $\mathbf{x}$*.*

Proof. See Appendix B.

For each local expert p_i, let us denote $\mathbf{L}_i(\widehat{\mathbf{w}}_i) \triangleq \mathbb{E}_{\mathbf{x}}[\mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x}; \widehat{\mathbf{w}}_i) \| p_i(y|\ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega}))]$ and $\widehat{\mathbf{L}}_i(\widehat{\mathbf{w}}_i) \triangleq \frac{1}{n} \sum_{t=1}^n \mathbf{D}_{\mathrm{KL}}(q_i(y^{(t)}|\mathbf{x}^{(t)}; \widehat{\mathbf{w}}_i) \| p_i(y^{(t)}|\ell_i(\mathbf{x}^{(t)}, \mathbf{D}_i); \boldsymbol{\omega}))$.

Intuitively, the second quantity $\widehat{\mathbf{L}}_i(\widehat{\mathbf{w}}_i)$ is the loss incurred by the black-box fitting objective (Section 4.2) on the training dataset while the first quantity $\mathbf{L}_i(\widehat{\mathbf{w}}_i)$ is the expected loss incurred when tested on the entire space of test input. Lemma 2 now establishes the next result which upper-bounds $\mathbf{L}_i(\widehat{\mathbf{w}}_i)$ in terms of the optimal black-box fitting noise $\theta \triangleq \min_i \mathbf{L}_i(\mathbf{w}_i^*)$ where $\mathbf{w}_i^* \triangleq \min_{\mathbf{w}} \mathbf{L}_i(\mathbf{w})$.

Lemma 2 *Let* n *denote the number of training samples used to fit* $q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$ *to* $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$*. Let* η *be defined in Assumption* 1 *and* $\delta \in (0, 1)$*. Then, we can guarantee that with probability at least* $1 - \delta$*,* $\mathbf{L}_i(\widehat{\mathbf{w}}_i) \leq \theta + 2\epsilon$ *by setting* $n = (\eta^2 / 2\epsilon^2) \log(2/\delta)$*.*

Proof. See Appendix C.

Combining Lemma 1 and Lemma 2, we can establish a stronger result which provides a lower-bound for the probability that $q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$ will agree with $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$ as detailed in Theorem 1 below.

Theorem 1 *Let* $\mathbf{x}$ *be an arbitrary test input and let* $\mathbf{E}$ *denotes the event that* $q_i(y \mid \mathbf{x}; \widehat{\mathbf{w}}_i)$ *and* $p_i(y \mid \ell_i(\mathbf{x}, \mathbf{D}_i); \boldsymbol{\omega})$ *agree on the prediction for* $\mathbf{x}$*. Then, given a training dataset of size* $n = (\eta^2 / 2\epsilon^2) \log(2/\delta)$*, with probability* $1 - \delta$*,*

$$\mathcal{P}(\mathbf{E}) \quad \geq \quad 1 - \frac{2}{\alpha^2} \Big(\theta + 2\epsilon \Big) \log 2 \, . \quad (15)$$

This implies the event that prediction disagreement between the surrogate and black-box happens with probability at most $2/\alpha^2 \, (\theta + 2\epsilon) \log 2$.

Proof. See Appendix D. Our theoretical analysis thus confirms that the proposed surrogate imitation algorithm (Section 4.2) yields surrogates with high fidelity.

6. Experiments

This section evaluates and reports the empirical performance of our collective fusion frameworks CIGAR (light-weight, ephemeral inference fusion) and COLBI (persistent surrogate model fusion) on three real-world datasets:

(a) The DIABETES dataset (Efron et al., 2004) containing 442 diabetes patient records with 10 features. The target output is a quantitative measure of disease progression one year after baseline.

(b) The AIMPEAK dataset (Chen et al., 2013a) containing 41850 instances of traffic measured along 775 road segments of an urban road network, each comprises of 5 variables that describe the corresponding segment. The target output is the averaged vehicle speed on the segment (km/h).

(c) The PROTEIN dataset (Rana, 2013) featuring physicochemical properties of protein tertiary structure with 45730 instances, each containing 9 variables that describe various properties of the structure. The target output is the size of the residue (kDa).

For each dataset above, we demonstrate the performance of both COLBI and CIGAR fusion for 10 experts on two settings: (a) homogeneous experts with 10 sparse Gaussian process (SGP) black-boxes (Hensman et al., 2013); (b) heterogeneous experts with 5 SGP black-boxes and 5 Bayesian ridge regression (BRR) (Radford, 1999) black-boxes.

Each black-box is trained on a randomly selected (non-overlapping) subset of 30 (DIABETES), 500 (AIMPEAK) and 500 (PROTEIN) data points, respectively. For experiments with COLBI fusion, an equal number of data points is used to learn each imitating surrogate model. The predictive performance of each persistent surrogate (COLBI) and each light-weight (ephemeral) fusion client (CIGAR) is then measured by the normalized root-mean-square-error (nRMSE) metric (pre- and post-fusion) with respect to corresponding test sets containing 35 (DIABETES), 500 (AIMPEAK) and 500 (PROTEIN) data points.

6.1. CIGAR performance

Fig. 1 shows CIGAR fusion results on three tested datasets:

(a) Fig. 1a, 1c, and 1e demonstrate fusion gain from individual perspectives of the (light-weight) fusion clients corresponding to 5 SGP and 5 BRR black-box experts on

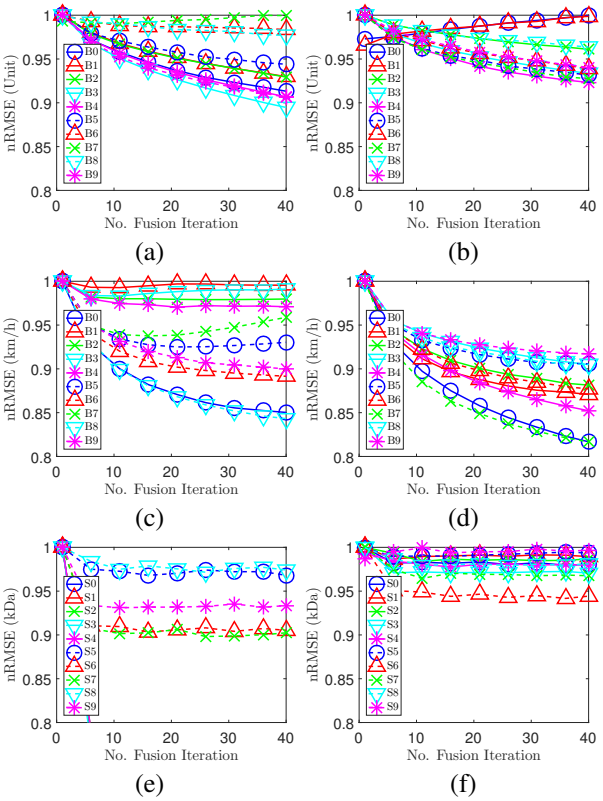

Figure 1. Graphs of normalized RMSE vs. no. fusion iterations for CIGAR on: DIABETES with (a) heterogeneous and (b) homogeneous black-box experts; AIMPEAK with (c) heterogeneous and (d) homogeneous experts; PROTEIN with (e) heterogeneous and (f) homogeneous experts.

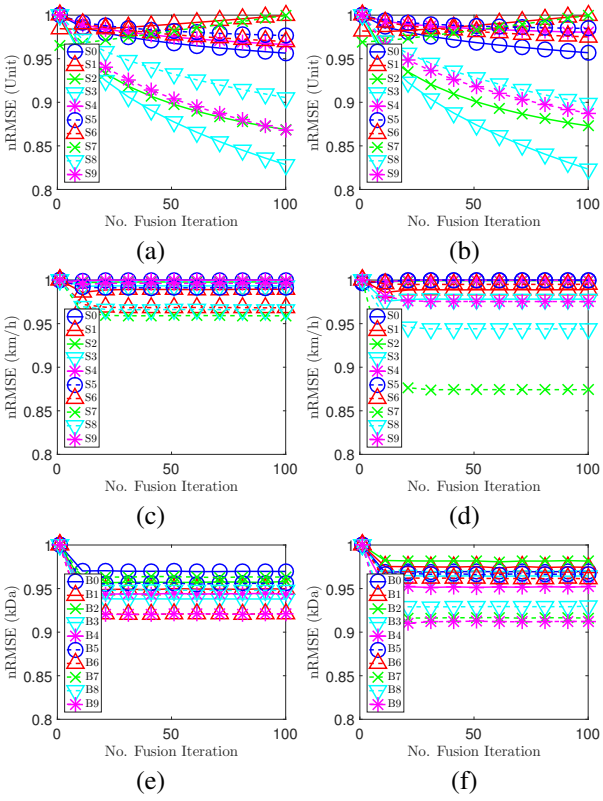

Figure 2. Graphs of normalized RMSE vs. no. fusion iterations for COLBI on: DIABETES with (a) heterogeneous and (b) homogeneous black-box experts; AIMPEAK with (c) heterogeneous and (d) homogeneous experts; PROTEIN with (e) heterogeneous and (f) homogeneous experts

DIABETES (Efron et al., 2004), AIMPEAK (Chen et al., 2013a) and PROTEIN (Rana, 2013) datasets, respectively. Across these experiments, the prediction errors (nRMSE) achieved by majority of fusion clients show a decreasing trend with more fusion iterations. Over 40 iterations, most clients improve its predictive accuracy by up to 16%.

(b) Fig. 1b, 1d, and 1f further demonstrate fusion gain of CIGAR in setting with homogeneous experts (10 SGP black-box experts) on DIABETES (Efron et al., 2004), AIMPEAK (Chen et al., 2013a) and PROTEIN (Rana, 2013) datasets respectively. Again, the errors incurred by the majority of clients decrease up to 19% over 40 fusion iterations, with the exception of clients $B0$ and $B1$ in Fig. 1b (DIABETES with homogeneous black-box experts). This is not unexpected because the (pre-fusion) predictive performance of the other clients are significantly worse, hence causing $B0$ and $B1$ to succumb to the opinion of the mass. Nevertheless, we also observe the performance drop in both instances is less than 4%, whereas the performance gain achieved by the rest of the clients are generally more significant (up to 7%). The averaged nRMSE gain is thus positive, as shown in Fig. 3a.

Fig. 4a and 4b demonstrate the convergence of predicted outputs of 10 fusion clients across 35 test points of the DIABETES dataset. The box-plot shown at each test datum is plotted with 10 predicted outputs made by the corresponding fusion clients. The box-plots in Fig. 4a (pre-fusion predictions) generally show larger variance than those in Fig. 4b (post-fusion predictions), which suggests that the fusion clients have moved towards a consensus prediction after 40 fusion iterations, as expected.

6.2. COLBI performance

Fig. 2 shows COLBI fusion results on the same datasets:

(a) Fig. 2a, 2c and 2e demonstrate fusion gain from individual perspectives of the surrogate models corresponding to 5 SGP and 5 BRR black-box experts while Fig. 2b, 2d and 2f demonstrate that of the surrogate models corresponding to 10 SGP black-box experts on DIABETES, AIMPEAK and PROTEIN, respectively. Overall, performance gains (up to 18%) for majority of surrogates are again observed consistently across all experiments, with some slight decrease in performance (less than 3%) observed at several surrogates.

(b) Fusion gain is much more significant for COLBI on the DIABETES dataset compared to the other two datasets. This is because the COLBI surrogates in both DIABETES

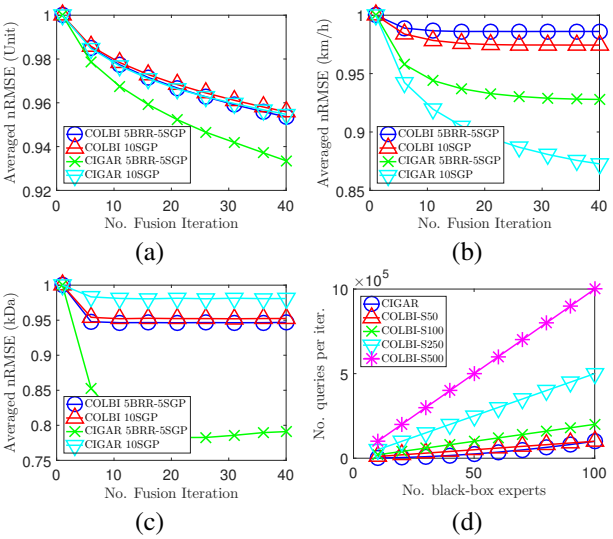

Figure 3. Graphs of averaged nRMSE of various methods vs. no. fusion iterations on (a) DIABETES, (b) AIMPEAK and (c) PROTEIN datasets; (d) Graphs of no. black-box queries used per iteration of various methods vs. no. black-box experts

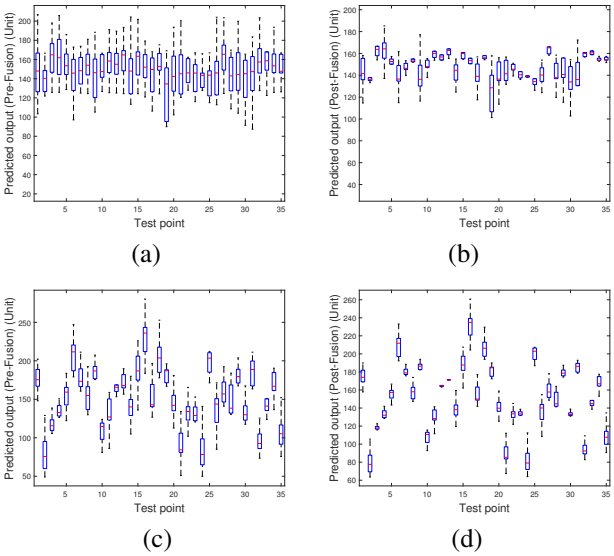

Figure 4. Graphs of (a) pre-fusion and (b) post-fusion nRMSE box-plot at every test datum with CIGAR on DIABETES dataset; (c) pre-fusion and (d) post-fusion nRMSE box-plot at every test datum with COLBI on DIABETES dataset.

experiments are trained (aligned with their corresponding black-boxes) using a very limited amount of surrogate data (30 data points), thus are likely to have more "gaps of knowledge" that can strongly benefit from predictive fusion.

Similar to the convergence results of CIGAR, Fig. 4c and 4d also demonstrate a general consensus over predicted outputs of 10 surrogates across 35 test points of the DIABETES dataset. This is observed as the variance of post-fusion box-plots is significantly smaller than that of pre-fusion.

Fig. 3a, 3b and 3c suggest that the average fusion gain by CIGAR is more significant than that of COLBI on both heterogeneous and homogeneous black-box expert settings. This is not surprising since CIGAR inference fusion combines the prediction of the actual black-box experts, whereas COLBI fusion aggregates over surrogate models, which are only trained to mimic the provided experts. As the quality to which these surrogate models capture the behaviour of their corresponding experts is not perfect, fusing them is expected to be less effective.

Last but not least, Fig. 3d shows the number of black-box query calls per fusion iteration (CIGAR) and per imitation iteration (COLBI) (with varying amount of data used to align each surrogate with its expert - 50, 100, 250 and 500) as the number of black-box experts increases. With large amount of surrogate data, the number of query calls made by COLBI quickly overwhelms that of CIGAR (which does not need to query surrogate data), thus confirming the trade-off between producing light-weight, ephemeral fused prediction (CIGAR) vs. costly, persistent fused model (COLBI).

7. Conclusion

This paper tackles a collective ML problem where the goal is to combine the expertise of independent black-box models with heterogeneous learning architectures to produce improved predictive capabilities. We identify three challenges pertaining to this problem: (a) performing fusion without access to the black-box training data and architectures, (b) performing fusion when the black-box models are not permanently available, and (c) avoiding centralized bottlenecks and risk of failure for large-scale fusion with numerous black-box experts. These challenges are addressed as this paper presents two fusion methods: (1) Decentralized CIGAR to perform light-weight inference fusion and (2) Decentralized COLBI to perform persistent surrogate fusion. We further provide theoretical analysis and perform empirical evaluation of our method on three real-world datasets.

Acknowledgements

This work was supported in part by the Gordon and Betty Moore Foundations Data-Driven Discovery Initiative [GBMF4554 to C.K.]; by the US National Institutes of Health [R01GM122935]; by The Shurl and Kay Curci Foundation and by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program; by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative. The authors would also like to thank Prashant Pandey for proof-reading this paper and Kingsford's group members for their constructive feedbacks.

References

Allamraju, R. and Chowdhary, G. Communication efficient decentralized gaussian process fusion for multi-uas path planning. In *American Control Conference*, pp. 4442–4447, 05 2017. doi: 10.23919/ACC.2017.7963639.

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

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

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

Chen, J., Low, K. H., and Tan, C. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. *Robotics: Science and System*, 06 2013c.

Deisenroth, M. P. and Ng, J. W. Distributed Gaussian processes. In *Proc. ICML*, pp. 1481–1490, 2015.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression, Annals of Statistics, 407-499. https://www4.stat.ncsu.edu/ boos/var.select/diabetes.html, 2004.

Gifford, C. M. *Collective Machine Learning: Team learning and classification in multi-agent systems*. PhD thesis, University of Kansas Lawrence, Kansas, USA, 2009.

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

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

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

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

Hoang, T. N., Hoang, Q. M., Ruofei, O., and Low, K. H. Decentralized high-dimensional bayesian optimization with factor graphs. In *Proc. AAAI*, 2018.

Hoang, T. N., Hoang, Q. M., Low, K. H., and How, J. P. Collective online learning of gaussian processes in massive multi-agent systems. In *Proc. AAAI*, 2019.

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

Low, K. H., Yu, J., Chen, J., and Jaillet, P. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*, pp. 2821–2827, 2015.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, pp. 1273–1282, 2017. URL http://arxiv.org/abs/1602.05629.

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

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

Radford, M. N. Regression and classification using gaussian process priors. *Bayesian Statistics*, 6, 01 1999.

Rana, P. S. Physicochemical properties of protein tertiary structure data set. http://archive.ics.uci.edu/ml/datasets/, 2013.

Ruofei, O. and Low, K. H. Gaussian process decentralized data fusion meets transfer learning in large-scale distributed cooperative perception. In *Proc. AAAI*, pp. 3876–3883, 2018.

Titsias, M. K. and Lázaro-Gredilla, M. Variational inference for Mahalanobis distance metrics in Gaussian process regression. In *Proc. NIPS*, pp. 279–287, 2013.

Yahya, A., Li, A., Kalakrishnan, M., Chebotar, Y., and Levine, S. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *International Conference on Intelligent Robots and Systems*, pp. 79–86, 09 2017. doi: 10.1109/IROS.2017.8202141.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, T. N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *Proc. ICML*, 2019.

A. Random Gradient Estimation

The (non-analytic) local gradient of each wrapper with respect to $\boldsymbol{\omega}$ can be estimated using the following randomized gradient estimation technique:

$$\begin{aligned}\nabla_{\boldsymbol{\omega}} \log p_i(y|\mathbf{x};\boldsymbol{\omega}) &\simeq \mathbb{E}_{\mathbf{z}}\left[\frac{\mathbf{z}}{\lambda}\log\left(\frac{p_i(y|\mathbf{x};\boldsymbol{\omega}+\lambda\mathbf{z})}{p_i(y|\mathbf{x};\boldsymbol{\omega})}\right)\right] \\ &= \mathbb{E}_{\mathbf{z}}\left[\nabla_{\boldsymbol{\omega}}^{(\mathbf{z})}\log p_i(y|\mathbf{x},\boldsymbol{\omega})\right] \qquad (16)\end{aligned}$$

where $\lambda > 0$ is a sufficiently small value, $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I})$ and

$$\nabla_{\boldsymbol{\omega}}^{(\mathbf{z})}\log p_i(y|\mathbf{x},\boldsymbol{\omega}) \triangleq \frac{\mathbf{z}}{\lambda}\log\left(\frac{p_i(y|\mathbf{x};\boldsymbol{\omega}+\lambda\mathbf{z})}{p_i(y|\mathbf{x};\boldsymbol{\omega}}\right) \qquad (17)$$

which can effectively be used as an unbiased stochastic gradient that guarantees convergence. Note that in this section, we instead omit $\ell_i(\mathbf{x},\mathbf{D}_i)$ from $p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$ to avoid cluttering the notation. The black-box notation is instead abbreviated succinctly as $p_i(y|\mathbf{x},\boldsymbol{\omega})$.

To understand the rationality behind Eq. (16) above, let $g(\boldsymbol{\omega})$ be an arbitrary function $\boldsymbol{\omega}$. Then, let $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I})$ and $\mathbf{v} \triangleq \mathbf{z}/\|\mathbf{z}\|$, it follows that $\mathbf{v}$ is a unit vector and $\nabla_{\boldsymbol{\omega}} g(\boldsymbol{\omega})^\top \mathbf{v} = \mathbf{D}_{\mathbf{v}} g(\boldsymbol{\omega})$ where $\mathbf{D}_{\mathbf{v}} g(\boldsymbol{\omega})$ is the directional gradient of $g(\boldsymbol{\omega})$,

$$\mathbf{D}_{\mathbf{v}} g(\boldsymbol{\omega}) \triangleq \lim_{\alpha\to 0}\frac{1}{\alpha}\Big(g(\boldsymbol{\omega}+\alpha\mathbf{v}) - g(\boldsymbol{\omega})\Big)\,. \qquad (18)$$

Choose $\alpha = \lambda\|\mathbf{z}\|$ with $\lambda > 0$, Eq. (18) can be rewritten as

$$\mathbf{D}_{\mathbf{v}} g(\boldsymbol{\omega}) \simeq \frac{1}{\lambda\|\mathbf{z}\|}\Big(g\left(\boldsymbol{\omega}+\lambda\mathbf{z}\right) - g(\boldsymbol{\omega})\Big) \qquad (19)$$

for sufficiently small value of λ. Thus, plugging Eq. (19) and $\mathbf{v} = \mathbf{z}/\|\mathbf{z}\|$ into $\nabla_{\boldsymbol{\omega}} g(\boldsymbol{\omega})^\top \mathbf{v} = \mathbf{D}_{\mathbf{v}} g(\boldsymbol{\omega})$ yields

$$\nabla_{\boldsymbol{\omega}} g(\boldsymbol{\omega})^\top \mathbf{z} = \frac{1}{\lambda}\Big(g(\boldsymbol{\omega}+\lambda\mathbf{z}) - g(\boldsymbol{\omega})\Big) \qquad (20)$$

As such, let $\nabla_{\boldsymbol{\omega}}^{(\mathbf{z})} g(\boldsymbol{\omega}) \triangleq (\mathbf{z}/\lambda)(g(\boldsymbol{\omega}+\lambda\mathbf{z}) - g(\boldsymbol{\omega}))$, it is easy to see that $\mathbb{E}[\nabla_{\boldsymbol{\omega}}^{(\mathbf{z})} g(\boldsymbol{\omega})] = \mathbb{E}[\mathbf{z}\nabla g(\boldsymbol{\omega})^\top \mathbf{z}] = (\mathbb{V}[\mathbf{z}] + \mathbb{E}[\mathbf{z}]\mathbb{E}[\mathbf{z}]^\top)\nabla_{\boldsymbol{\omega}} g(\boldsymbol{\omega}) = \nabla_{\boldsymbol{\omega}} g(\boldsymbol{\omega})$. The last equality follows because $\mathbb{E}[\mathbf{z}] = 0$ and $\mathbb{V}[\mathbf{z}] = \mathbf{I}$ by definition. Finally, plugging $g(\boldsymbol{\omega}) = \log p_i(y|\mathbf{x};\boldsymbol{\omega})$ yields Eq. (16).

Likewise, the local gradient with respect to y can also be estimated using the same technique:

$$\begin{aligned}\nabla_y \log p_i(y|\mathbf{x};\boldsymbol{\omega}) &\simeq \mathbb{E}_z\left[\frac{z}{\lambda}\log\left(\frac{p_i(y+\lambda z|\mathbf{x};\boldsymbol{\omega})}{p_i(y|\mathbf{x};\boldsymbol{\omega})}\right)\right] \\ &= \mathbb{E}_z\left[\nabla_y^{(z)}\log p_i(y|\mathbf{x},\boldsymbol{\omega})\right] \qquad (21)\end{aligned}$$

where $\lambda > 0$ is a sufficiently small value, $z \sim \mathcal{N}(0,1)$ and

$$\nabla_y^{(z)}\log p_i(y|\mathbf{x},\boldsymbol{\omega}) \triangleq \frac{z}{\lambda}\log\left(\frac{p_i(y+\lambda z|\mathbf{x};\boldsymbol{\omega})}{p_i(y|\mathbf{x};\boldsymbol{\omega})}\right) \qquad (22)$$

Thus, $\nabla_{\boldsymbol{\omega}}^{(\mathbf{z})}\log p_i(y|\mathbf{x},\boldsymbol{\omega})$ and $\nabla_y^{(z)}\log p_i(y|\mathbf{x},\boldsymbol{\omega})$ can be used as unbiased stochastic gradients of the full gradients $\nabla_{\boldsymbol{\omega}}\log p_i(y|\mathbf{x},\boldsymbol{\omega})$ and $\nabla_y \log p_i(y|\mathbf{x},\boldsymbol{\omega})$, respectively.

B. Proof of Lemma 1

By Pinkser inequality, we have

$$\frac{1}{2\log 2}\Big\|q_i - p_i\Big\|_1^2 \leq \mathbf{D}_{\mathrm{KL}}(q_i\|p_i) \leq \frac{\alpha^2}{2\log 2}\,, \qquad (23)$$

which implies $|q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i) - p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})| \leq \alpha$ for all y. Let $y_p \triangleq \arg\max p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$ and $y_q \triangleq \arg\max q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i)$, respectively. Thus, we have

$$\begin{aligned}p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) &\geq p_i(y_q|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) \\ &\geq q_i(y_q|\mathbf{x};\widehat{\mathbf{w}}_i) - \alpha \\ &\geq q_i(y_p|\mathbf{x};\widehat{\mathbf{w}}_i) + 2\alpha - \alpha \\ &\geq p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) + 2\alpha - 2\alpha \\ &= p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})\,. \qquad (24)\end{aligned}$$

That is, $p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) \geq p_i(y_q|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) \geq p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$, which immediately follows that $p_i(y_p|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}) = p_i(y_q|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$. This means $y_p = y_q$[4] or equivalently, $q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i)$ agrees with $p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})$ on the prediction of $\mathbf{x}$. To understand this, note that the second and fourth inequalities follow immediately from the fact that $|q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i) - p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})| \leq \alpha$ for all $\mathbf{x}$ whereas the third inequality follows from Definition 1.

C. Proof of Lemma 2

For each local expert i, let us denote $\mathbf{L}_i(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{x}}[\mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x};\mathbf{w})\|p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}))]$ and $\widehat{\mathbf{L}}_i(\mathbf{w}) \triangleq \frac{1}{n}\sum_{t=1}^n \mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x}^{(t)};\mathbf{w})\|p_i(y|\ell_i(\mathbf{x}^{(t)},\mathbf{D}_i);\boldsymbol{\omega}))$ where $\{\mathbf{x}^{(t)}\}_{t=1}^n$ are drawn i.i.d from $\mathcal{P}(\mathbf{x})$.

Thus, it follows that $\mathbf{L}_i(\mathbf{w}) = \mathbb{E}_{\mathbf{x}}[\widehat{\mathbf{L}}_i(\mathbf{w})]$ and since $\mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x};\mathbf{w})\|p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}))$ is bounded within $(0,\eta)$ (see Assumption 1), we can bound the difference between $\mathbf{L}_i(\mathbf{w})$ and $\widehat{\mathbf{L}}_i(\mathbf{w})$ using Hoeffding inequality:

$$\mathcal{P}\left(|\mathbf{L}_i(\mathbf{w}) - \widehat{\mathbf{L}}_i(\mathbf{w})| \leq \epsilon\right) \geq 1 - \exp\left(-\frac{2n\epsilon^2}{\eta^2}\right) \qquad (25)$$

Setting $\delta/2 = \exp(-2n\epsilon^2/\eta^2)$ and solving for n yields $n = (\eta^2/2\epsilon^2)\log(2/\delta)$. That is, if $\widehat{\mathbf{L}}_i(\mathbf{w})$ is computed using $n = (\eta^2/2\epsilon^2)\log(2/\delta)$ data points, then for each $\mathbf{w}$, the following holds with probability at least $1-\delta/2$:

$$\Big|\mathbf{L}_i(\mathbf{w}) - \widehat{\mathbf{L}}_i(\mathbf{w})\Big| \leq \epsilon \qquad (26)$$

Let $\mathbf{w}_i^* \triangleq \min_{\mathbf{w}} \mathbf{L}_i(\mathbf{w})$. By the union bound, the above inequality holds simultaneously for $\mathbf{w}_i^*$ and $\widehat{\mathbf{w}}_i \triangleq \min_{\mathbf{w}} \widehat{\mathbf{L}}_i(\mathbf{w})$ (solving for $\widehat{\mathbf{w}}_i$ is detailed in Section 4.1) with

[4] We implicitly assume that the maximizer of p_i are unique.

probability at least $1-\delta$. When that happens, we have

$$\begin{aligned}\mathbf{L}_i(\widehat{\mathbf{w}}_i) &\leq \widehat{\mathbf{L}}_i(\widehat{\mathbf{w}}_i) + \epsilon \\ &\leq \widehat{\mathbf{L}}_i(\mathbf{w}_i^*) + \epsilon \\ &\leq \mathbf{L}_i(\mathbf{w}_i^*) + 2\epsilon = \theta + 2\epsilon \end{aligned} \tag{27}$$

where the first and third inequalities follows directly from applying Eq. (26) above to $\widehat{\mathbf{w}}_i$ and $\mathbf{w}_i^*$, respectively. The second inequality follows from the definition of $\widehat{\mathbf{w}}_i$ and the last equality follows from the above definition of θ, i.e., $\theta \triangleq \min_i \mathbf{L}_i(\mathbf{w}_i^*)$

D. Proof of Theorem 1

To prove this result, let us recall that $\mathbf{L}_i(\widehat{\mathbf{w}}_i) \triangleq \mathbb{E}[\mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i)\|p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega}))]$. Thus, by Markov inequality, it follows with probability at least $1-2\log 2\mathbf{L}_i(\widehat{\mathbf{w}}_i)/\alpha^2$ that:

$$\mathbf{D}_{\mathrm{KL}}(q_i(y|\mathbf{x};\widehat{\mathbf{w}}_i)\|p_i(y|\ell_i(\mathbf{x},\mathbf{D}_i);\boldsymbol{\omega})) \leq \frac{\alpha^2}{2\log 2} \tag{28}$$

When this happens, by Lemma 1, it further follows that q_i and p_i agree on the prediction of $\mathbf{x}$. That is, with probability at least $1-2\log 2\mathbf{L}_i(\widehat{\mathbf{w}}_i)/\alpha^2$, $\mathbf{E}$ happens. Now, applying Lemma 2 which states that with probability at least $1-\delta$, $\mathbf{L}_i(\widehat{\mathbf{w}}_i) \leq \theta + 2\epsilon$. Plugging this into the above expression of $1-2\log 2\mathbf{L}_i(\widehat{\mathbf{w}}_i)/\alpha^2$ thus yields:

$$\begin{aligned}\mathcal{P}(\mathbf{E}) &\geq 1 - 2\log 2\frac{\mathbf{L}_i(\widehat{\mathbf{w}}_i)}{\alpha^2}\,, \\ &\geq 1 - \frac{2}{\alpha^2}(\theta+2\epsilon)\log 2\,. \end{aligned} \tag{29}$$

The inequality above only holds with probability $1-\delta$ since $\mathbf{L}_i(\widehat{\mathbf{w}}_i) \leq \theta + 2\epsilon$ only happens with probability $1-\delta$, as shown in Lemma 2 above.