This is a repository copy of *A learning distributed Gaussian process approach for target tracking over sensor networks*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/187471/

Version: Accepted Version

**Proceedings Paper:**

Liu, X., Lyu, C., George, J. et al. (2 more authors) (2022) A learning distributed Gaussian process approach for target tracking over sensor networks. In: Proceedings of the 2022 25th International Conference on Information Fusion (FUSION). 2022 25th International Conference on Information Fusion (FUSION), 04-07 Jul 2022, Linköping, Sweden. Institute of Electrical and Electronics Engineers . ISBN 9781665489416

https://doi.org/10.23919/FUSION49751.2022.9841315

© 2022 The Authors. This accepted manuscript version is available under a Creative Commons Attribution CC BY licence. (http://creativecommons.org/licenses/by/4.0)

**Reuse**
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

# A Learning Distributed Gaussian Process Approach for Target Tracking over Sensor Networks

Xingchi Liu*, Chenyi Lyu*, Jemin George**, Tien Pham** and Lyudmila Mihaylova*
*Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK
**US Army Research Laboratory, Adelphi, MD 20783, USA
xingchi.liu@sheffield.ac.uk, clyu5@sheffield.ac.uk, jemin.george.civ@army.mil,
tien.pham1.civ@army.mil, l.s.mihaylova@sheffield.ac.uk

***Abstract*—Tracking manoeuvring targets often relies on complex models with non-stationary parameters. Gaussian process (GP) based model-free methods can achieve accurate performance in a data-driven manner but face scalability challenges. Aiming to address such challenges, this paper proposes a distributed GP-based tracking approach able to learn the kernel hyperparameters in an online manner, to improve the tracking performance and scalability. It caters to the inherent distributed feature of sensor networks and does not need measurements to be transmitted among sensors for target states predictions. Theoretical upper confidence bounds about the tracking error are derived within the regret bound setting. Through this theoretical analysis, the tracking error per time step is upper bounded as a function of predictive variances from local sensors. The theoretical results are supported by simulation based ones over a case study for tracking over wireless sensor networks. With evaluation on challenging target trajectories, a comparison on state-of-the-art centralised and distributed GP approaches, numerical results demonstrate that the proposed approach achieves competitively high and robust tracking performance.**

***Index Terms*—** Distributed Tracking, Gaussian Process Methods, Product of Experts, Sensor Networks, Uncertainty Quantification

## I. Introduction

Tracking the object trajectory in sensor networks is a fundamental task for various applications including sea surveillance, autonomous vehicles, and traffic management. To achieve superior tracking performance, numerous model-based approaches have been proposed. However, these approaches rely on well-defined motion models, which can be inaccurate when the target undergoes non-stationary evolution or mixed manoeuvrings behaviours. Although the multiple-model method [1] can capture complex behaviours, it suffers from high computational complexity, and therefore, is not efficient when a large number of models are involved.

### A. Related Work

Tracking with multiple models can be achieved by Gaussian process (GP)-based model-free methods, which, as non-parametric methods, can learn unknown functions directly from noisy data. In [2], one-dimensional temporal GP regression models were reformulated as linear-Gaussian state-space models, which can be solved exactly with classical Kalman filtering methods. The state-space model representation was also used in spatial-temporal GPs [3] and non-Gaussian likelihood [4] to derive computationally efficient infinite-dimensional Kalman filtering and smoothing methods.

There are also works studying using GP to represent the state-space model [5]. GP was used to learn the whole or part of the state-space model and the learned functions can be integrated into a particle filter or extended Kalman filter [6]–[8]. Inference and learning for GP state-space model were also discussed with theoretical bounds derived in [9]. Moreover, recently, different GP approaches were developed by assuming temporal and spatial correlation in the target trajectory and shape, and the state-space model was directly learned from the measurements [10]–[12].

Most of the existing GP-based tracking methods assume that the sensor measurements are collected in a centralised manner and filters are built upon the aggregated measurements. This framework has given rise to an inherent limitation of computational complexity. This is due to that the complexity of the standard GP-based tracking methods increases cubically in the number of measurements due to the inversion and determinant of the kernel covariance matrix. Therefore, state estimation based on a standard GP with extensive measurements faces numerical challenges in real-time tracking tasks. Extensive works have studied how to derive sparse approximations of the original kernel matrix [13], [14], which can greatly reduce the complexity. However, most of them are still centralised methods which do not match up with the feature of distributed tracking systems where transmitting data among sensors can be expensive or even infeasible due to the limited communication capabilities of sensors.

Inspired by the divide-and-conquer method [15], local GP approximation methods have been proposed to improve the scalability of the standard GP method, which has been applied to problems such as the received signal strength-based location fingerprinting map construction [16], [17]. These distributed algorithms build local GPs (which are referred to as the local experts) with different subsets of data and can make independent local predictions. Based on local experts, a variety of aggregation methods have been designed to achieve a more accurate global prediction only using local predictions [18] rather than exchanging local measurements among sensors. This type of distributed GP (DGP) methods can greatly reduce the computational complexity of standard GP regression from $\mathcal{O}(N^3)$ to $\mathcal{O}(Nn^2)$ ($n \ll N$), where $N$ and $n$ represent the total number of data and the number of local data, respectively. Particularly, it is a viable solution for distributed target tracking since sensors can make independent predictions and the global prediction can be made even some of the sensors fail to transmit their predictions. Therefore, DGP strengthens the scalability and robustness of standard GP and the aforementioned limitations can be overcome.

### B. Contributions

This paper has four main contributions: 1) it proposes a DGP-based tracking (DGPT) approach, which enables distributed point target tracking in an online manner and inherits the uncertainty quantification capability from standard GP; 2) the hyperparameters of the DGPT algorithm are learned online based on a sliding window of sensor measurements data to provide high accuracy and scalability of the proposed approach. For the training input, both temporal and spatial-temporal inputs are used for DGP training; 3) an upper confidence bound (UCB) of the tracking error is derived for the proposed DGPT approach; 4) the performance of the DGPT is carefully evaluated with challenging target trajectory scenarios and different levels of measurement noise.

The remainder of this paper is organized as follows. Backgrounds of GP and DGP are given in Section II. Section III describes the proposed DGPT approach followed by the theoretical analysis in Section IV. Numerical results are presented and discussed in Section V. Section VI concludes this paper.

## II. Background

### A. Overview of Standard GP

The non-linear mapping between the current input $\mathbf{x} \in \mathbb{R}^d$ and the object one-dimensional state $f(\mathbf{x}) \in \mathbb{R}$ can be modelled as a GP which can be written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{1}$$

$$m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right], \tag{2}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[\left(f(\mathbf{x}) - \mathbf{x}\right)\left(f(\mathbf{x}') - \mathbf{x}'\right)\right], \tag{3}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are either the training or the testing input data, $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ denote the mean and covariance functions, respectively.

A GP regression problem with noisy observations from the unknown function can be written as

$$z = f(\mathbf{x}) + \epsilon, \ \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where $z$ represents the measurements and $\epsilon$ represents the identical independently distributed (i.i.d.) zero-mean measurement noise with variance $\sigma^2$.

Given a training data set of input-output pairs $\mathcal{D} = \{X, \mathbf{z}\}$ with $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ and $\mathbf{z} = \{z_1, z_2, \cdots, z_n\}$, define $K = K(X, X)$ as the covariance matrix for the training input and $K_* = K(X, \mathbf{x}_*)$ as the covariance between the training input $X$ and any testing input point $\mathbf{x}_*$, the GP regression equations at this new input $\mathbf{x}_*$ can be written as

$$\mu_* = m(\mathbf{x}_*) + K_*^{\intercal}\Sigma^{-1}(\mathbf{z} - m(\mathbf{x}_*)), \tag{5}$$

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - K_*^{\intercal}\Sigma^{-1}K_*, \tag{6}$$

where $\Sigma = K + \sigma^2 I$. $\mu_*$ and $\sigma_*^2$ denote the posterior predictive mean and variance of the unknown function at input $\mathbf{x}_*$, respectively.

### B. Distributed GP Approaches

This section discusses two types of DGP schemes. The first type is the product of expert (PoE). The idea of this approach is to multiply the independent local predictive distributions for aggregation. Since the product of these Gaussian predictions is proportional to a Gaussian distribution, the closed form of the aggregated predictive mean and variance can be easily calculated. The PoE model provides a straightforward way to aggregate local predictions and sidesteps the weight assignment problem in other local approximated GP models such as the mixture-of-expert [19]. However, this model can be overconfident when making predictions in regions without any training data. The generalised product-of-experts (GPoE) model [20] improved PoE by adding weights $\beta$ that represent the contributions of different experts. For example, it can be calculated as the difference in the differential entropy between the prior distribution $p(f(\mathbf{x}_*))$ and the posterior predictive distribution $p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D})$, which can be written as

$$\beta_i = 0.5\left(\log\sigma_{**}^2 - \log\sigma_i^2(\mathbf{x}_*)\right), \tag{7}$$

where $\sigma_{**}^2$ represents the variance of the prior distribution $p(f(\mathbf{x}_*))$ and $\sigma_i^2(\mathbf{x}_*)$ denote the predictive variance of GP expert $i$, which can be calculated based on (6).

Given the data $\mathcal{D}^{(i)}$ collected by sensor $i$, the GPoE predicts a function value $f(\mathbf{x}_*)$ at a test input $\mathbf{x}_*$. The predictive distribution and the closed forms of the aggregated predictive mean and variance of GPoE can be written as

$$p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}) = \prod\nolimits_{i=1}^{M} p_i^{\beta_i}(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}^{(i)}), \tag{8}$$

$$\mu_*^{\text{GPoE}} = (\sigma_*^{\text{GPoE}})^2 \sum\nolimits_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_*)\mu_i(\mathbf{x}_*), \tag{9}$$

$$(\sigma_*^{\text{GPoE}})^{-2} = \sum\nolimits_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_*), \tag{10}$$

where $M$ is the number of GP experts and represents the number of sensors in the tracking system. Moreover, $\mu_i(\mathbf{x}_*)$ represents the predictive mean of GP expert $i$, which can be calculated based on (5).

Alternatively, the Bayesian committee machine (BCM) [21] proposed to aggregate the experts' predictions from another view by imposing a conditional independence assumption of $\mathcal{D}^{(i)}$ and $\mathcal{D}^{(j)}$ given $f(\mathbf{x}_*)$ which means an independent common prior $p(f(\mathbf{x}_*) \mid \mathbf{x}_*)$ for the experts. In [22], a robust Bayesian committee machine (RBCM) was designed which combines both the features of the GPoE and BCM models. The predictive distribution and aggregated predictive mean and variance of the RBCM can be written as

$$p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}) = \frac{\prod_{i=1}^{M} p_i^{\beta_i}(f(\mathbf{x}_* \mid \mathbf{x}_*, \mathcal{D}^{(i)})}{p^{\sum_i \beta_i - 1}(f(\mathbf{x}_*) \mid \mathbf{x}_*)}, \tag{11}$$

$$\mu_*^{\text{RBCM}} = (\sigma_*^{\text{RBCM}})^2 \sum\nolimits_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_*)\mu_i(\mathbf{x}_*), \tag{12}$$

$$(\sigma_*^{\mathrm{RBCM}})^{-2} = \sum\nolimits_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_*) + (1 - \sum\nolimits_{i=1}^{M} \beta_i)\sigma_{**}^{-2}, \quad (13)$$

where the denominator in (11) plays the role of a correction term that helps to recover the GP prior when leaving regions of existing data. All the models discussed in this section work in a distributed way. However, these are still offline schemes that cannot be directly applied for tracking, therefore, two schemes are introduced in Section III which enable the batch-based offline distributed GP methods to work online for real-time tracking.

### C. Hyperparameter Estimation

The hyperparameters of GP need to be learned from the data. As a standard GP, maximum likelihood estimation (MLE) is applied to learn the hyperparameters by maximising the log marginal likelihood which can be written as

$$\log p(\mathbf{z}|X,\theta) = -\frac{1}{2}\mathbf{z}^{\intercal}\Sigma^{-1}\mathbf{z} - \frac{1}{2}\log|\Sigma| - \frac{n}{2}\log 2\pi, \quad (14)$$

where $\theta$ represents the set of hyperparameters.

For DGP, assuming the local GPs are independent of each other, the marginal likelihood can be factorised as

$$\log p(\mathbf{z}|X,\theta) \approx \sum\nolimits_{i=1}^{M} \log p_i(\mathbf{z}^{(i)}|X^{(i)},\theta). \quad (15)$$

As compared to (14), the factorised marginal likelihood can potentially be maximised in a decentralised manner due to the fact that it is a summation of local marginal likelihood functions. In addition, after solving (15), the learned hyperparameters are shared by all the local experts for automatic regularization to avoid overfitting.

## III. Distributed GP-based tracking

The proposed DGPT approach works in an online way. At each time step, local experts are maintained at each sensor with optimized hyperparameters from the previous time step. Each local expert predicts the target state based on its own measurements and then local predictions are aggregated to reach an overall prediction following different aggregation methods discussed in Section II-B. Notice having this aggregation process does not mean an extra central node is necessary. The aggregation process can be implemented on any capable sensor with highly limited communication overhead, therefore the proposed approach is fully distributed. In this section, several improvement schemes are designed to help integrate DGP for efficient distributed online target tracking.

### A. Temporal and Spatial-temporal GP

To make a state prediction of a target, it is reasonable to assume that there is a temporal correlation in the motions of the target, and the temporal correlation with input data in distant past is weaker than in more recent data. Therefore, the target state can be represented as a function of the time step, which can be used as the input data for training DGP and making state predictions. In this case, we have $\mathbf{x} = t$, where $t$ represent the time step.

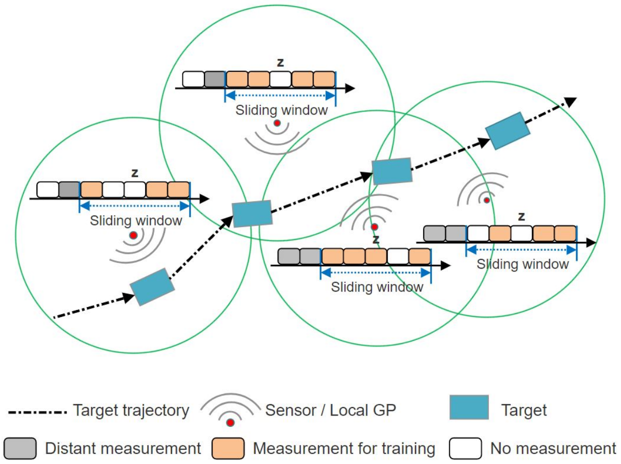

Fig. 1. A distributed point tracking system with 4 sensors. The length of sliding window in this example is 5 time steps

Based on the temporal GP, the tracking problem in (1) can be reformulated as

$$f(t) \sim \mathcal{GP}(m(t), k(t,t')), \quad (16)$$
$$z = f(t) + \epsilon, \ \epsilon \sim \mathcal{N}(0,\sigma^2). \quad (17)$$

Furthermore, inspired by [23] that uses GP regression to learn the state transition function, the spatial correlation can also be involved for state prediction by including the target state of the previous time step into the input data, namely $\mathbf{x} = (r,t)$, where $r$ represents the target state. The spatial-temporal GP based tracking problem can be formulated as

$$f(r,t) \sim \mathcal{GP}(m(r,t), k(r,t;r',t')), \quad (18)$$
$$z = f(r,t) + \epsilon, \ \epsilon \sim \mathcal{N}(0,\sigma^2). \quad (19)$$

The next subsection describes the proposed sliding windows based GP approach.

### B. Sliding Window-based Tracking

Although DGP is designed for complexity reduction of the standard GP, training GPs locally can still be computationally intense when local sensors collect extensive data. To address this challenge, assume there are temporal correlations among target motions, and weaker motion correlations in distant past time steps cannot contribute to training and prediction as much as the recent steps, since the target motion can be a mixed manoeuvring behaviour with time-varying parameters. Therefore, the distant measurements are abandoned in DGPT and only the recent measurements are utilized for training, which can be treated as using a sliding window to select valid measurements. Based on the sliding window, a sensor without valid measurements is excluded from aggregation at the current time step. Eventually, both the number of experts and measurements can be further reduced, which can ease the computational burden of GP training and improve the tracking accuracy. The framework of the proposed sliding window-based DGPT approach is represented in Fig. 1.

### C. Hyperparameter Online Learning in the DGPT

Since the target motion can be time-varying and the sliding window is designed to keep valid measurements for training and tracking, the hyperparameters of DGP should be learned online to capture the non-stationary features. Hence, in the proposed DGPT approach, MLE (based on the factorised marginal likelihood (15)) is solved at every time step to update the hyperparameters, which brings extra computational costs due to the non-convexity of (15). To accelerate the hyperparameter learning process, optimised hyperparameters at the previous update step is designed to be set as the initial value of hyperparameters of MLE at the current step, which can significantly reduce the iterations needed for MLE and makes the learning process close to a recursive one.

Benefiting from the proposed sliding window design and the online learning properties, the DGPT can provide accurate target state estimations with newly collected measurements.

## IV. Theoretical performance analysis

In this section, the tracking error of the proposed DGPT approach is analyzed. An UCB is derived to quantify the error between the predictive state and the real state. This begins by presenting that the deviation between the true state and predictive state can be upper bounded by a scaled version of the predictive variance. Then we use this bound to generate the error bound for the aggregated prediction.

*Lemma 1:* (Lemma 5.1 of [24]) *Based on* (5) *and* (6)*, given a trained local GP based on data set $\mathcal{D} = \{X, \mathbf{z}\}$, for any test point $\mathbf{x}_* \in X$, the probability that the predictive mean $\mu(\mathbf{x}_*)$ deviates from the true function value by more than a certain amount can be upper bounded as*

$$\Pr\left\{|f(\mathbf{x}_*) - \mu(\mathbf{x}_*)| > \gamma^{1/2}\sigma(\mathbf{x}_*)\right\} \leq e^{-\gamma/2}, \tag{20}$$

where $\gamma$ is a positive constant.
Lemma 1 proposes an UCB of the probability that the deviation between the true function value and the predictive mean of the function at is larger then a scaled version of the estimated variance function. Based on this lemma, the error bound of DGP can be derived.

*Theorem 1: (One-step error bound of GPoE) Consider a DGP system with $M$ local GPs, Based on (9) and (10), with probability at least $1 - \sum_{i=1}^{M} e^{-\gamma_i/2}$, the deviation between the true function value at $\mathbf{x}_*$ and the aggregated predictive mean value made by the GPoE method* (8)-(10) *can be upper bounded as*

$$\Pr\left\{ |f(\mathbf{x}_*) - \mu(\mathbf{x}_*)| \leq \frac{\sum\limits_{i=1}^{M} \gamma_i^{1/2}\sigma_i^{-1}(\mathbf{x}_*)}{\sum\limits_{i=1}^{M} \sigma_i^{-2}(\mathbf{x}_*)} \right\} \geq 1 - \sum_{i=1}^{M} e^{-\frac{\gamma_i}{2}}. \tag{21}$$

*Proof:* Define $A_i$ as the event in which the error between the predictive object state of local expert $i$ and the true state is larger than a quantity, which can be written as

$$A_i = \left\{|f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)| > \gamma_i^{1/2}\sigma_i(\mathbf{x}_*)\right\}. \tag{22}$$

Define the union of events $\{A_1, A_2, \cdots, A_M\}$ as $A$. Applying the union bounds over $M$ events and based on Lemma 1, the probability of the complement of $A$ can be upper bounded as

$$\begin{aligned}\Pr(\bar{A}) &= \Pr\left\{\bigcap_{i=1}^{M} \bar{A}_i\right\}, \\ &= \Pr\left\{\bigcap_{i=1}^{M} |f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)| \leq \gamma_i^{1/2}\sigma_i(\mathbf{x}_*)\right\}, \\ &\geq 1 - \sum_{i=1}^{M} e^{-\gamma_i/2}.\end{aligned} \tag{23}$$

where $\mu_i(\mathbf{x}_*)$ and $\sigma_i(\mathbf{x}_*)$ represent the predictive mean and standard deviation of local GP $i$ at input $\mathbf{x}_*$, respectively.

According to (9) and (10), the deviation between the true function value and the aggregated predictive mean by GPoE can be written as

$$\begin{aligned}\left|f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}\right| &= \left|f(\mathbf{x}_*) - \frac{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)\mu_i(\mathbf{x}_*)}{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)}\right|, \\ &= \frac{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)|f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)|}{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)}, \\ &\leq \frac{\sum_{i=1}^{M}\beta_i\gamma_i^{1/2}\sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)},\end{aligned} \tag{24}$$

which completes the proof. Notice that the tracking error bound of RBCM can be derived as well following a similar process.

Theorem 1 proposes a theoretical error bound for the tracking performance. Define the highest predictive variances made by the local expert as $\sigma_{\text{H}}^2$, the bound can be further approximated as

$$\begin{aligned}&1 - \sum_{i=1}^{M} e^{-\gamma_i/2} \\ &\leq \Pr\left\{|f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| \leq \frac{\sum_{i=1}^{M}\beta_i\gamma_i^{1/2}\sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^{M}\beta_i\sigma_i^{-2}(\mathbf{x}_*)}\right\}, \\ &\leq \Pr\left\{|f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| \leq \frac{\sum_{i=1}^{M}\beta_i\gamma_i^{1/2}\sigma_i^{-1}(\mathbf{x}_*)}{M\sigma_{\text{H}}^{-2}(\mathbf{x}_*)\sum_{i=1}^{M}\beta_i}\right\}.\end{aligned} \tag{25}$$

This bound demonstrates that, given all other local GPs fixed, when one of the local GP makes a highly uncertain prediction which is reflected as a larger predictive variance, the upper bound of the deviation will increase, which means the overall prediction is exacerbated by this poor GP expert.

## V. Simulation results

### A. Simulation Setup

The proposed DGPT is tested on three scenarios and each with 100 Monte Carlo (MC) runs. There are three noise levels generated by adding zero-mean Gaussian noise to each of the target states with standard deviations of 1, 2, and 4. There are 200 sensors uniformly implemented in the interest of area, so part of the trajectory can be outside of the sensor coverage.

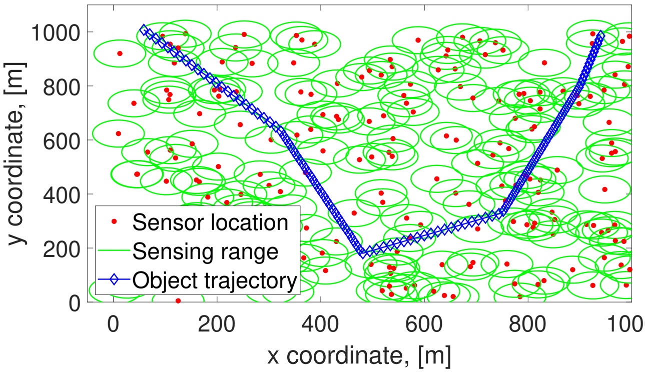

Fig. 2. Trajectory S1

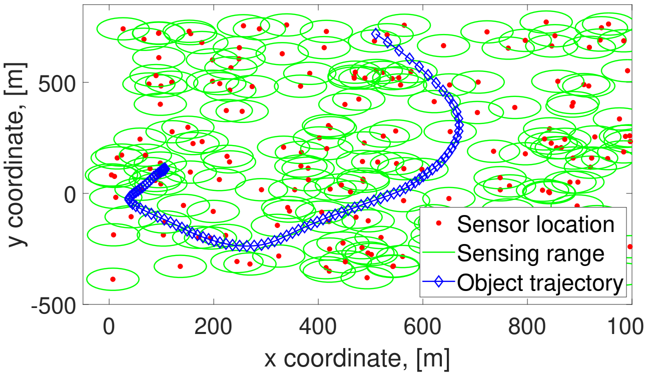

Fig. 3. Trajectory S2

The sensing range is 50 meters and the sampling period is one second, both of which are identical for every sensor. Moreover, this paper assumes that local predictions can be aggregated without any information loss.

Notice in this paper, we are estimating the location of the target so two GPs are needed at one sensor for the $X$-coordinate and $Y$-coordinate, respectively. For the GP setting, a zero-mean function is used which means no extra knowledge is utilised for tracking. Besides, the covariance function is selected to be the squared exponential (SE) kernel which is demonstrated to perform well in a wide range of motion models [25]. The SE kernel can be represented as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_m^2 \exp\left\{ -\frac{1}{2} \sum_{j=1}^{d} \frac{(x_j - x_j')^2}{l_j^2} \right\}, \tag{26}$$

where $\sigma_m$ is the amplitude parameter and $l_j$ is the length scale parameter.

### B. Benchmarks

Since this paper focuses on the model-free approaches, the standard GP-based centralised tracking approach is simulated as the benchmark. This scheme relies on solving the MLE (14) to learn the hyperparameters, and the learning process requires the measurements to be transmitted in the sensor network. To make fair comparisons, the standard GP-based centralised tracking approach is trained with the same sensor measurements in

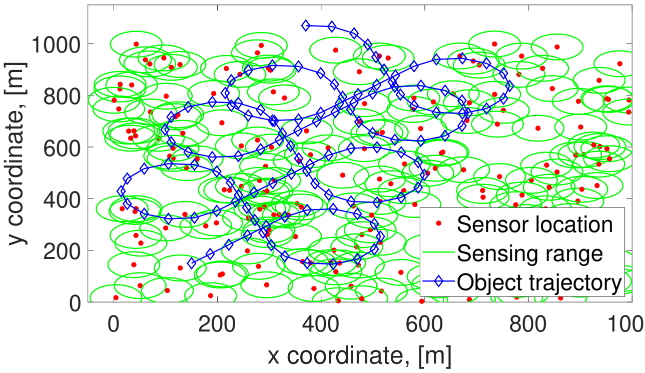

Fig. 4. Trajectory S3

the sliding window, which means this approach use the same amount of data for model training and hyperparameter learning as well as the DGPT approach.

To study the impact of different aggregation methods on the DGPT approach, both RBCM and GPoE are simulated. In addition, DGPT based on both temporal and spatial-temporal input data is evaluated, respectively. For the temporal case, a set of time steps which is in the sliding window is used as the input data. For the spatial-temporal case, both the current time step and the target state of the previous time step are used as input. Notice in the online tracking problem, the real target state is not available. Therefore, the predictive state acquired by the DGPT in the previous time step is used.

### C. Target Trajectories

To evaluate the proposed algorithm and the benchmarks, two challenging scenarios are built following different models. The trajectories and the sensors are depicted in Fig. 2, 3, and 4.

- **S1)** Similar to [26], the trajectory is generated based on the nearly constant velocity model in the straight line, and the abrupt velocity change at each pre-defined turning point.
- **S2)** The target trajectory is generated by the Singer acceleration model. The maximum possible acceleration is 50 $m/s^2$, the probability of non-acceleration is 0.4.
- **S3)** The target trajectory is generated by the sharp coordinated turns model (turning rate $30°/s$ for 9 s) and the target can go both left and right. This model can represent highly manoeuvrable motions.

### D. Tracking Performance

The average normalised root mean squared error (NRMSE) of temporal and spatial-temporal DGPT approaches are presented in Table I and II. The lengths of sliding windows in different trajectories are carefully tuned to be different for optimal performance. At each time step, both the state prediction of the next step and the updated state estimation of the current step involving the new measurements are presented.

According to the results, the updated NRMSEs are always lower than the predictive NRMSEs, which shows that GP-based approaches can refine the learned predictive distribution when

TABLE I
PREDICTIVE AND UPDATED NRMSES ON THREE TRAJECTORIES AND THREE NOISE LEVELS: TEMPORAL GP

| Scenario | Approach | Noise Level 1 | | | | Noise Level 2 | | | | Noise Level 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Updated NRMSE | | Predictive NRMSE | | Updated NRMSE | | Predictive NRMSE | | Updated NRMSE | | Predictive NRMSE | |
| | | X | Y | X | Y | X | Y | X | Y | X | Y | X | Y |
| S1 | Standard GP | 1.30% | 1.70% | 1.95% | 2.56% | 1.41% | 1.07% | 2.09% | 1.65% | 0.95% | 1.05% | 1.42% | 1.59% |
| | DGPT-RBCM | 0.27% | 0.28% | 0.53% | 0.62% | 0.41% | 0.48% | 0.72% | 0.89% | 0.62% | 0.74% | 1.00% | 1.24% |
| | DGPT-GPoE | 0.31% | 0.33% | 0.83% | 1.64% | 0.50% | 0.59% | 0.83% | 1.67% | 0.86% | 0.95% | 1.02% | 1.76% |
| S2 | Standard GP | 1.19% | 0.74% | 1.63% | 1.06% | 1.08% | 0.68% | 1.51% | 0.99% | 1.37% | 0.77% | 1.87% | 1.11% |
| | DGPT-RBCM | 0.90% | 0.82% | 1.52% | 1.38% | 1.16% | 0.99% | 1.88% | 1.61% | 1.73% | 1.21% | 2.59% | 1.59% |
| | DGPT-GPoE | 1.13% | 0.88% | 1.52% | 1.45% | 1.50% | 1.13% | 1.68% | 1.52% | 2.36% | 1.56% | 2.44% | 1.81% |
| S3 | Standard GP | 6.35% | 5.45% | 9.76% | 8.25% | 6.39% | 6.09% | 9.93% | 9.33% | 6.32% | 6.31% | 9.77% | 9.66% |
| | DGPT-RBCM | 5.55% | 6.39% | 10.58% | 10.46% | 6.00% | 6.63% | 11.21% | 10.77% | 5.81% | 6.63% | 11.10% | 10.85% |
| | DGPT-GPoE | 5.38% | 6.57% | 7.19% | 8.15% | 5.86% | 6.79% | 8.00% | 8.52% | 5.69% | 6.79% | 7.94% | 8.57% |

TABLE II
PREDICTIVE AND UPDATED NRMSES ON THREE TRAJECTORIES AND THREE NOISE LEVELS: SPATIAL-TEMPORAL GP

| Scenario | Approach | Noise Level 1 | | | | Noise Level 2 | | | | Noise Level 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Updated NRMSE | | Predictive NRMSE | | Updated NRMSE | | Predictive NRMSE | | Updated NRMSE | | Predictive NRMSE | |
| | | X | Y | X | Y | X | Y | X | Y | X | Y | X | Y |
| S1 | Standard GP | 0.40% | 0.95% | 1.02% | 3.72% | 0.57% | 0.67% | 1.25% | 3.32% | 0.76% | 1.77% | 1.35% | 4.15% |
| | DGPT-RBCM | 0.27% | 0.28% | 1.08% | 4.41% | 0.41% | 0.48% | 1.17% | 4.55% | 0.67% | 0.76% | 1.51% | 5.33% |
| | DGPT-GPoE | 0.30% | 0.33% | 1.00% | 3.58% | 0.50% | 0.60% | 1.00% | 3.60% | 0.86% | 0.97% | 1.20% | 4.02% |
| S2 | Standard GP | 1.26% | 0.81% | 1.81% | 1.20% | 1.00% | 0.79% | 1.50% | 1.21% | 1.73% | 0.95% | 2.73% | 1.42% |
| | DGPT-RBCM | 0.87% | 0.82% | 1.40% | 1.40% | 1.10% | 1.02% | 1.76% | 1.71% | 1.73% | 1.21% | 2.73% | 2.03% |
| | DGPT-GPoE | 1.11% | 0.94% | 1.58% | 1.49% | 1.46% | 1.26% | 1.71% | 1.84% | 2.73% | 1.57% | 3.65% | 2.00% |
| S3 | Standard GP | 3.07% | 20.55% | 7.79% | 33.41% | 3.08% | 19.40% | 7.32% | 31.55% | 3.58% | 19.13% | 7.80% | 30.98% |
| | DGPT-RBCM | 3.79% | 4.93% | 8.41% | 8.32% | 3.93% | 4.85% | 8.48% | 8.34% | 4.05% | 4.81% | 8.84% | 8.55% |
| | DGPT-GPoE | 5.60% | 8.51% | 10.46% | 12.95% | 5.73% | 8.65% | 10.60% | 13.15% | 5.64% | 8.75% | 10.63% | 13.36% |

new measurements are collected. Particularly, the proposed DGPT approach performs competitively well and even outperforms the centralised approach in some scenarios. This is due to that in DGP, different weights can be assigned to the local predictions during the prediction aggregation process, so the final aggregated predictions are closer to the expert who makes more confident predictions. In the centralised method, all the data is aggregated before training without any difference. Moreover, the results demonstrate that using RBCM for prediction aggregation can achieve lower NRMSEs than GPoE in most cases, which justifies that adding the common prior into the aggregation process can improve the tracking performance. Finally, comparing the NRMSEs in two tables, we can find that considering spatial input data can help to improve the tracking accuracy, especially in the more challenging scenarios where the speed of the target keeps changing or the target keeps manoeuvring (scenario S2 and S3).

### *E. Impact of Uncertainties on Tracking*

In Fig. 5 and 6, the updated state estimations of spatial-temporal DGPT and the real target locations are presented. The results are collected in S1 under noise level three. The shaded areas represent the $3\sigma$ intervals of the predictive distributions which can reflect the level of tracking confidence. From the figures, we can find that the updated estimation is close to the ground truth. There are several periods that the estimation is relatively far from the true location, since the target is out of the sensing range and no measurements are collected for updating the predictive distributions in these steps. Notice that whenever a relatively poor estimation is made, a higher uncertainty can be observed, which is in line with the real tracking situation.

Moreover, the standard deviations of 100 RMSEs (collected from 100 MC runs) under noise level one are illustrated in Fig. 7. spatial-temporal DGPT has comparable performance in all three scenarios and on both coordinates. Particularly, DGPT using RBCM as the prediction aggregation method achieves the lowest standard deviation among multiple scenarios, which demonstrates that this approach can ensure robust tracking in a distributed way.

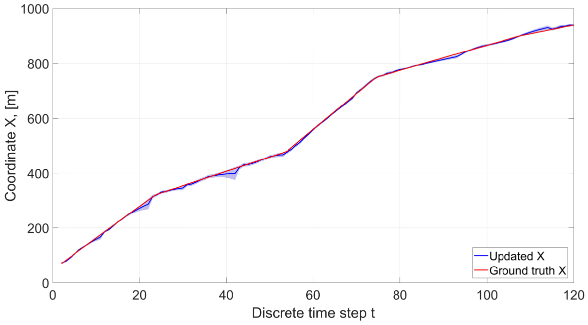

Fig. 5. Tracking uncertainty in X coordinates

## VI. CONCLUSIONS

In this paper, a new DGP-based model-free learning and tracking approach is proposed to solve distributed point tracking problems. The developed approach overcomes the limitations of standard GP-based tracking methods from a different perspective via distributed GP. A theoretical derivation is presented for the UCB of the tracking error of the proposed approach, which characterises the trustworthiness of the proposed approach.

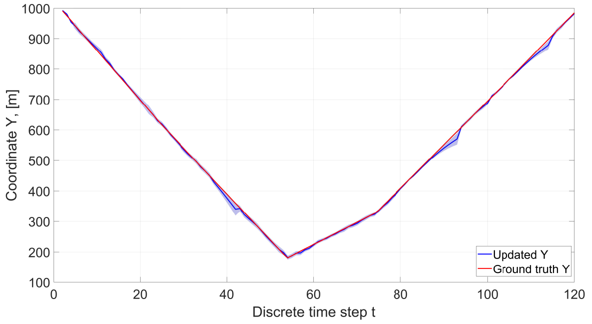

Fig. 6. Tracking uncertainty in Y coordinates

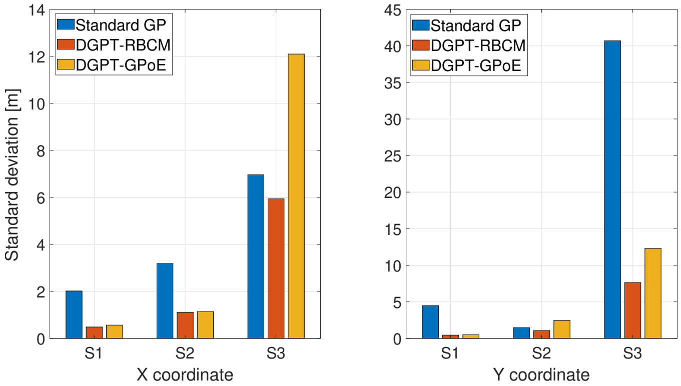

Fig. 7. Standard deviation of RMSE in both coordinates

The estimates are acceptable when the derived UCB is within certain pre-specified limits. Numerical experiments demonstrate that the proposed approach performs competitively well and can deal with varying motion models and noise levels. Future work will focus on involving data association in distributed tracking solutions and deriving UCBs for them.

## ACKNOWLEDGMENTS

This research is sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing under the SIGNeTS project. It is accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government or the U.K. Government. The U.S. Government and U.K. Government are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was funded partly by the EPSRC EP/T013265/1 project NSF-EPSRC: "ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven " and the National Science Foundation under Grant USA NSF ECCS 1903466. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

## References

[1] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[2] J. Hartikainen and S. Särkkä, "Kalman Filtering and Smoothing Solutions to Temporal Gaussian Process Regression Models," in *In Proc. of the IEEE International Workshop on Machine Learning for Signal Processing*, 2010, pp. 379–384.

[3] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal Learning via Infinite-Dimensional Bayesian Filtering and Smoothing: A Look at Gaussian Process Regression Through Kalman Filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.

[4] H. Nickisch, A. Solin, and A. Grigorevskiy, "State Space Gaussian Processes with Non-Gaussian Likelihood," in *Proc. of the 35th International Conference on Machine Learning*, vol. 80. PMLR, 10–15 Jul 2018, pp. 3789–3798.

[5] R. Turner, M. Deisenroth, and C. Rasmussen, "State-space inference and learning with gaussian processes," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 868–875.

[6] C. Veibäck and J. Olofsson and T. Lauknes and G. Hendeby, "Learning target dynamics while tracking using Gaussian processes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 2591–2602, 2020.

[7] A. Kullberg, I. Skog, and G. Hendeby, "Online Joint State Inference and Learning of Partially Unknown State-Space Models," *arXiv preprint arXiv:2102.07418*, 2021.

[8] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, "A Gaussian Process based Method for Multiple Model Tracking," in *Proc. of the Sensor Signal Processing for Defence Conference (SSPD)*, 2020, pp. 1–5.

[9] Y. Zhao, C. Fritsche, G. Hendeby, F. Yin, T. Chen, and F. Gunnarsson, "Cramér–Rao Bounds for Filtering Based on Gaussian Process State-Space Models," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5936–5951, 2019.

[10] W. Aftab and L. Mihaylova, "A Gaussian Process Regression Approach for Point Target Tracking," in *Proc. of the 22th International Conf. on Information Fusion*, Ottawa, 2019, pp. 1–8.

[11] ——, "A Learning Gaussian Process Approach for Maneuvering Target Tracking and Smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 278–292, 2021.

[12] W. Aftab, R. Hostettler, A. De Freitas, M. Arvaneh, and L. Mihaylova, "Spatio-Temporal Gaussian Process Models for Extended and Group Object Tracking With Irregular Shapes," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2137–2151, 2019.

[13] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, vol. 18, p. 1257, 2006.

[14] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, D. van Dyk and M. Welling, Eds., vol. 5, Florida USA, 2009, pp. 567–574.

[15] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian Process Meets Big Data: A Review of Scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[16] F. Yin and F. Gunnarsson, "Distributed Recursive Gaussian Processes for RSS Map Applied to Target Tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, 2017.

[17] Y. Zhao, C. Liu, L. S. Mihaylova, and F. Gunnarsson, "Gaussian processes for RSS fingerprints construction in indoor localization," in *Proc. of the 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 1377–1384.

[18] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized Robust Bayesian Committee Machine for Large-Scale Gaussian Process Regression," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 3131–3140.

[19] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[20] Y. Cao and D. J. Fleet, "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," *arXiv preprint arXiv:1410.7827*, 2014.

[21] V. Tresp, “A Bayesian Committee Machine,” *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[22] M. Deisenroth and J. W. Ng, “Distributed Gaussian Processes,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, Lille, 2015, pp. 1481–1490.

[23] K. Berntorp, “Online Bayesian inference and learning of Gaussian-process state–space models,” *Automatica*, vol. 129, p. 109613, 2021.

[24] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Madison, WI, USA, 2010, p. 1015–1022.

[25] W. Aftab and L. Mihaylova, “On the Impact of Different Kernels and Training Data on a Gaussian Process Approach for Target Tracking,” in *Proc. of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, Rustenburg, 2020, pp. 1–6.

[26] A. Gning and L. Mihaylova, “Dynamic Clustering and Belief Propagation for Distributed Inference in Random Sensor Networks with Deficient Links,” in *Proc. of the 12th International Conf. on Information Fusion*, Seattle, 2009, pp. 656–663.