The Pennsylvania State University
The Graduate School

SAFE MACHINE LEARNING FOR INTELLIGENT MULTI-ROBOT SYSTEMS

A Dissertation in
Electrical Engineering
by
Zhenyuan Yuan

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2024

The dissertation of Zhenyuan Yuan was reviewed and approved by the following:

Minghui Zhu
Associate Professor of Electrical Engineering
Dissertation Advisor, Chair of Committee

Constantino Lagoa
Professor of Electrical Engineering

Alan Wagner
Associate Professor of Aerospace Engineering

Ying Sun
Assistant Professor of Electrical Engineering

Madhavan Swaminathan
Department Head and Professor of Electrical Engineering

Abstract

Recent advances in embedded computing and mobile sensing have led to pervasive use of robotic systems in both civil and military applications. With single autonomous robots for particular tasks widely accepted and used in a number of occasions and the development of high-speed communication technologies, there are attempts to connect the robots together and make them work collaboratively as a team. A key element that enhances the autonomy and intelligence of these robotic systems is machine learning. However, recent accidents associated with machine learning-enabled robots indicate that machine learning remains unsafe.

This dissertation is concerned with safe machine learning in intelligent multi-robot systems; that is, developing a set of algorithms which multi-robot systems can utilize to improve system performances and remain safe. The research agenda will be developed from the following aspects.

The dissertation starts from the fundamental problem of distributed learning with uncertainty quantification in multi-robot systems. In particular, we consider the problem where a group of agents aim to collaboratively learn a common latent function through streaming data. We propose a class of lightweight distributed Gaussian process regression algorithms that explicitly considers the limited budget in memory, computation, and communication in robotic systems. We show that communication brings Pareto improvement to the agents in the network by investigating the transient and the steady-state performances of the proposed algorithms.

We next show how to integrate the learning algorithm developed above with motion planning to ensure robot safety during the entire online learning process. In particular, we propose a learning and planning framework to solve safe navigation problems in uncertain environments or under uncertain dynamics. We further derive the sufficient conditions to ensure the safety of the system.

Then we consider the problem of zero-shot generalization in reinforcement

learning. In particular, we consider the problem of multiple learners collaboratively learning a single control policy which is able to perform well without data collection and policy adaptation in new environments. We formulate the problem as a federated optimization problem with an unknown objective function. We propose a class of federated optimization algorithms which leverages on zero-shot generalization guarantees. We further derive theoretical guarantees on almost-sure convergence, almost consensus, Pareto improvement and global convergence.

Finally, we investigate how a robot can quickly adapt its control policy online by incrementally leveraging its previous learning experiences. Specifically, we study online meta reinforcement learning on physical agents. We propose a novel online meta update method and a policy masking framework. The policy masking framework ensures all-time safety, while the online meta update method is sample-efficient and is able to achieve sublinear growth of dynamic regret.

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I am proudly grateful to my Ph.D. advisor, Professor Minghui Zhu, whose expertise, encouragement, dedication and advice were invaluable throughout my research process and career development. This dissertation would not be possible without him. Thanks to him, I was not only provided with plenty of research opportunities, but also guided for my future academic career in a comprehensive ways, including the identification of interesting research problems, writing papers, preparing funding proposals, presentation skills, as well as mentoring students. I am indebted to him for his patience, wisdom, and unwavering belief in my abilities.

I would like to express my sincere appreciation to my dissertation committee members, Professor Constantino Lagoa, Professor Alan Wagner and Professor Ying Sun, for their willingness to serve as well as their valuable feedback and insightful suggestions to the development and refinement of this dissertation.

I am truly thankful to my colleagues and friends in Professor Minghui Zhu's research group for their camaraderie, intellectual exchange, and support. I would like to thank Dr. Hunmin Kim, Dr. Zhisheng Hu, Dr. Yang Lu, Dr. Guoxiang Zhao, Dr. Samer Saab, Dr. Xu Zhang, Mr. Siyuan Xu, Mr. Shicheng Liu, Mr. Yue Mao and Mr. Rashed Aldhafeeri for the stimulating discussions and collaborative environment which greatly enhanced the quality of my research work and Ph.D. life.

My gratitude extends to my other collaborators, Dr. Hai Lin, Dr. Zihan Zhou, Dr. Rui Yu, Dr. Seong-Woo Kim, Dr. Younghwa Jung, Mr. Tongjia Zheng, Mr. Mollik Nayyar and Mr. Sihyeon Jo for the support and technical expertise.

I owe debt of gratitude to all the people I met during my undergraduate research experience. I would like to thank Professor Asok Ray and Professor Bharath Sriperumbudur for the opportunities and guidance of my undergraduate research. I would like to thank Dr. Devesh Jha, Dr. Nurali Virani, Dr. Yiwei Fu, Dr. Pinyao Guo, Dr. Eric Keller and Mr. Kevin Fisher at NRSL as well as Mr. Christopher Miller at the Penn State Schreyer Honors College for their mentorship and friendship. I also thank Mr. Zeyu Zhang, Mr. Chuong Nyugen, Ms. Ishana

Shekhawat, Mr. Ziyang Wang, Ms. Jilan Zhang and Mr. Joong Won Ah for the memorable collaboration.

I thank the Penn State community for providing a conducive environment for my academic and personal growth in the past nine years. I would like to thank all the Penn State friends, especially Mr. Waroch Tangbampensountorn, for making the years interesting and memorable.

I must say here that none of this would have been possible without my dad and my mom. I am deeply indebted to them for their dedication, encouragement and support without expecting anything in return. I would also like to thank my dog Kaito for all the cute moments and emotional support. The final words of the acknowledgment go to my wife (and my best teammate) Dr. Xue Xiao for her unwavering love, support and company throughout this long journey. Without the patience, optimism, and unconditional support from her I would not have been able to make it here.

This dissertation was supported by NSF grants ECCS-1710859, CNS-1830390, ECCS-1846706 and the Penn State College of Engineering Multidisciplinary Research Seed Grant Program. Any opinions, findings, and conclusions or recommendations expressed in this dissertation are those of the author and do not necessarily reflect the views of the funding agencies.

Chapter 1

Introduction

Recent advances in embedded computing and mobile sensing have led to pervasive use of robotic systems in both civil and military applications. For example, Tom, a robot created by Small Robot Company for precision agriculture, uses a combination of GPS, AI, and mobile technology to move safely and digitally map fields and estimate crop yields [1]. Australia-based Fastbrick Robotics can build brick houses four times faster than human workers as it combines a 3D printer and a robot that can lay bricks just as precisely as human laborers [1]. The CQ-10 SnowGoose, a small fully autonomous aircraft used to deliver supplies to US Special Operations forces in the field, can carry out GPS-guided missions for extended periods over distances as far as 150 km, and has been deployed in Afghanistan and Iraq [2]. The US Navy has recently demonstrated autonomous mine countermeasures, where a small autonomous surface vessel equipped with advanced sonar systems is deployed to patrol a suspected minefield, then detect and localize mines [3]. In February 2021, NASA's Perseverance rover landed on Mars to collect samples that will be returned to Earth on a future mission, while NASA's Curiosity rover, which landed on Mars in 2012, has been exploring new Martian terrain [4]. The global market for robots is expected to grow at a compound annual growth rate of around 26 percent to reach 210 billion U.S. dollars by 2025 [5].

With single autonomous robots for particular tasks widely accepted and used in a number of occasions as exemplified above and the development of high-speed communication technologies, such as WiFi and 5G, there are attempts to connect robots together and make them work collaboratively as a team [6]. For examples,

over 200,000 robots are performing packing and sorting in Amazon warehouses [7]. At the opening ceremony of 2018 Pyeongchang Winter Olympic Games, the light show made of 1,218 drones broke Guinness records. The U.S. Navy has developed an advanced ship protection system, which can deploy a fleet of more than a dozen of small unmanned boats swimming around a warship and detecting threats [8]. In 2019, the Defense Advanced Research Projects Agency experimented with using a swarm of autonomous drones and ground robots to assist with military missions and showed how the robots analyzed two city blocks to find, surround, and secure a mock city building; the whole system could eventually scale up to 250 drones and ground robots [9]. Waymo, Uber and Tesla have produced fleets of autonomous cars transporting passengers in a number of cities, e.g., Phoenix [10], Tempe [11], and Pittsburgh [12]. By the global market for autonomous cars, only a subset of multi-robot systems, is expected to reach a size of over 55.6 billion U.S. dollars in 2032 [13]. There are unique advantages that multi-robot systems enjoy over single-robot systems. First, it is very difficult and costly to design a monolithic robot that could accomplish various tasks in different environment conditions [14]. On the other hand, most difficult tasks are composed of a number of subtasks. Therefore, a difficult task can be accomplished through multiple robots specialized in different subtasks, and these robots can be reused through different combinations for a variety of tasks. Second, a single-robot system is vulnerable to single-point failures, where even a small failure of a robotic unit, e.g., malfunctions of a camera for perception, may prevent the accomplishment of the whole task [14]. In multi-robot systems, if some robots fails, remaining robots can continue to carry out missions [15]. Third, multi-robot systems can benefit from parallel operations, versatility and flexibility when deploying heterogeneous units, and inherent redundancy [14]. Therefore, multi-robot systems can provide cost-effectiveness, robustness, and high efficiency.

A key element that enhances the autonomy and intelligence of these robotic systems is machine learning. In particular, machine learning allows robotic systems to extract relevant, reliable and actionable information from sensor data, adapt to highly dynamic and unstructured environments, and achieve super-human performances in some cases. For example, Uber Engineering indicates that machine learning is involved in almost every component, e.g., perception, prediction, motion

planning and control, of Uber's autonomous cars [16]. Project Maven, a Pentagon project, has used Unmanned Aerial Systems equipped with machine learning to identify insurgent targets in Iraq and Syria in thousands of hours of drone footage [17][18].

1.1 Safety of machine learning

Current applications of advanced machine learning techniques, e.g., deep learning, are largely limited to software agents or computer programs, e.g., face/object recognition, natural language processing and AlphaGo, a computer program that plays the board game Go. Due to high volume, variety and velocity of processed data, learning efficiency and accuracy are dominant metrics for measuring the performances in these applications. In contrast, robots are physical agents and sometimes work around humans, their safety as well as the safety of their surroundings are more important than learning efficiency and accuracy. However, recent accidents associated with machine learning-enabled robots indicate that machine learning remains unsafe. From 2016 to 2019, there were five fatalities resulted from accidents of autonomous driving. All the accidents were caused by the malfunction of the machine learning algorithms. For instance, in 2016, a Tesla Motor S caused the first death by a self-driving car because it was unable to distinguish a white tractor-trailer crossing the highway against a bright sky [19]. NBC News reported that in 2019 a self-driving Uber car hit and killed a female because it was unable to recognize the pedestrian jaywalk [20]. Besides autonomous cars, a security robot drowned itself in a fountain because the algorithm failed to detect the uneven surface [21].

The reasons why machine learning remains unsafe are as follows. First, the machine learning models that achieve extraordinary performances are usually trained offline using an enormous amount of data [22]. In contrast, robots may face unexpected changes of their own dynamical systems and operating environments during mission execution. Prior information is limited for unanticipated scenarios or even unavailable for edge cases. In these situations, the responses of the machine learning models, though heavily trained, become uncertain. Second, existing machine learning algorithms, such as deep learning, exhibit one detrimental characteristics,

a trade-off between performance and transparency. That is, the more complex a machine learning model's working principle, which is usually able to solve more difficult tasks, the less clear how its decisions are made [23][24]. This makes the outputs of machine learning models less predictable when operating in complicated environments such as urban traffics, and the unexpected decisions could lead to catastrophic consequences. Third, current machine learning algorithms are often fragile, i.e., small perturbations of input data could lead to dramatic changes in learning outputs [25]. For example, paper [26] shows an experiment, where with some deliberately crafted camouflage graffiti and art stickers, a machine learning algorithm can misclassify a stop sign into a speed limit 45 sign in 100% of the images taken. The fragility is partially due to the fact that the decision making functions of the machine learning models are highly nonlinear and have large derivatives with respect to inputs.

1.2 Literature review

The previous section indicates that the challenge of safety in machine learning is a crucial barrier for wide-range deployment of robotic systems in human society. In both computer science and control communities, safe machine learning has been studied in different contexts. In general, safe machine learning aims to solve certain learning tasks and meanwhile ensure the system to stay within certain safety measures. It can be categorized into safe offline learning and safe online learning.

For safe offline learning, models are trained using a fixed set of data. Safety issues could therefore arise when robots encounter edge cases that are not observed in the training dataset. Furthermore, uncertainties of the robots' dynamic systems and their operating environments can also induce safety issues. Safe offline learning is usually tackled by modifying optimization criterion based on different safety definitions [27]. For example, risk-sensitive criterion can be introduced to balance between a return and a risk, where the risk can be the variance of the return or the probability of entering an unsafe region [28][29]. Constrained criterion can be included to enforce certain (safety) measures within given bounds [30][31]. Worst-case criterion is used to mitigate the effects of the variability induced by a given

policy due to the stochastic nature of the environment or the dynamic system [32][33].

As for safe online learning, robots sequentially collect training data and gradually refine their learning models on-the-fly. As a result, the learning errors could be large during the initial learning phase. It is therefore important and challenging to keep the robots safe during the entire learning process. According to learning tasks, related literature can be categorized into six classes: (1) exploration [34][35][36], where the objective is to learn about the uncertainties of a dynamic model or an environment; (2) optimization [37][38][39], where decision variables are selected to optimize an unknown objective function; (3) bandit [40], where the objective is to optimize a sequence of unknown objective functions; (4) reinforcement learning (RL) [27][41][42][43], where the objective is to find an optimal control policy to maximize aggregate return; (5) regulation [44][45][46], where the objective is to derive a control policy to achieve certain classical control specifications, such as stabilization and tracking; (6) navigation [47][48], where the objective is to find a sequence of control inputs to bring a dynamic system to a goal region. These works usually define safety as hard constraints. For example, the thresholds of certain function evaluations are usually considered in exploration, optimization, and bandit problems; state constraints and/or policy constraints are usually imposed in exploration, RL, regulation, and navigation. Safety is achieved if the constraints are satisfied throughout the learning process. Dynamic systems are usually not considered in the tasks of optimization, bandit, and sometimes, exploration. RL usually models a dynamic system as a Markov Decision Process, while tasks of regulation and navigation model a dynamic system using state-space equations.

1.3 Our contributions

Existing safe machine learning techniques mainly focus on single-robot systems. The counterparts of multi-robot systems have been rarely studied. There are new challenges arisen that make extending the machine learning techniques from single-robot systems to multi-robot systems non-trivial. First, problems involving multiple robots, such as multi-robot reinforcement learning, usually have complexity scaled up exponentially with respect to the number of the robots [49]. Second,

data are spatially distributed and locally maintained by each robot in multi-robot systems, while it is usually impractical for each robot to obtain all the data due to concerns of, e.g., transmission bandwidth, memory, and privacy [50]. However, the robots have incentives to fully utilize all the data since the performances of machine learning models are usually positively related to the size of dataset [51]. Third, robots in multi-robot systems usually have simple architectures with limited resources in, e.g., computation power, communication budget, and onbroad memory [52][53]. This makes the deployments of resource-hungry machine learning models unrealistic.

This dissertation aims to study a set of distributed algorithms for safe machine learning in intelligent multi-robot systems. The research agenda is developed from the following aspects.

Chapter 3 designs a class of distributed Gaussian process regression algorithms, which allow a group of robots to collaboratively learn a common latent function online. The algorithms estimate the uncertainties of intermediate learning results. The uncertainty quantification will be used in Chapter 4 where machine learning and motion planning are integrated. The developed algorithms are cognizant of limited resources of memory, computation and communication budget. Our analysis reveals that limited inter-agent communication improves learning performances in the sense of Pareto, i.e., some agents' performances improve without sacrificing other agents' performances. The algorithms are empirically verified by simulating a scenario where a multi-robot system is deployed to learn about a spatial signal, such as temperature or wind field. Chapter 3 is based on the following papers:

(C1) Z. Yuan and M. Zhu. "Communication-aware distributed Gaussian process regression algorithms for real-time machine learning", *American Control Conference*, pp. 2197-2202, July 2020.

(J1) Z. Yuan and M. Zhu. "Lightweight distributed Gaussian process regression algorithms for online machine learning", *IEEE Transactions on Automatic Control*, 2024. To appear.

In Chapter 3, how training data is collected is not specified. Hence, in Chapter 4, we investigate the scenario where training data is collected along robot trajectories. In particular, we consider the problem where a group of mobile robots

subject to unknown external disturbances, e.g., wind turbulence for drones, aim to safely reach goal regions. We develop a class of distributed safe learning and planning algorithms that allow the robots, in a single execution, to learn about the common environmental model using the GPR algorithms in Chapter 3, update their motion plans promptly, and enforce safety as collision avoidance of obstacles and other robots with high probability. This framework provides fast update of distributed coordination with other robots and fast adaptation to the sequence of dynamic models resulted from online learning. We further derive the sufficient conditions to ensure the safety of the robots. The framework is empirically evaluated by Monte Carlo simulation where a group of robots are deployed to navigate through different environments. Chapter 4 is based on the following papers:

(C2) Z. Yuan and M. Zhu. "dSLAP: Distributed safe learning and planning for multi-robot systems", *IEEE International Conference on Decision and Control*, pp. 5864-5869, December 2022.

(J2) Z. Yuan and M. Zhu. "Distributed safe learning and planning for multi-robot systems", *IEEE Transaction on Automatic Control*. Under review.

Chapter 4 considers obtaining a safe control policy through an online learning method, in Chapter 5 we consider the complementary and investigate the safety of a control policy obtained from an offline learning method. Specifically, we consider the problem where a network of learners collaboratively learn a universal feedback control policy for different environments. We formulate the problem as a federated optimization problem with an unknown objective function. We propose a class of federated reinforcement learning algorithms cognizant of zero-shot generalization guarantees on arrival time and safety. We derive theoretical guarantees on almost-sure convergence, almost consensus, Pareto improvement and global convergence. Monte Carlo simulation is conducted to evaluate the proposed framework. The following papers summarize the results in Chapter 5:

(C3) Z. Yuan, S. Xu and M. Zhu. "Federated reinforcement learning for generalizable motion planning", *American Control Conference*, pp. 78-83, May 2023.

(J3) Z. Yuan, S. Xu and M. Zhu. "Federated reinforcement learning for robot motion planning with zero-shot generalization", *Automatica*. Under review.

Chapter 4 considers online learning where no data is provided before deployment, whereas Chapter 5 considers offline learning using data provided before deployment. In Chapter 6, we consider how the online learning and offline learning can be combined to improve the performances when a robot encounter a sequence of tasks. Specifically, we consider the problem of how data collection and policy adaptation can be done efficiently together with guaranteeing all-time safety. We propose a class of online meta update algorithms together with a policy masking framework. The meta update algorithms achieve high sample efficiency, and sublinear growth of dynamic regret is analyzed. All-time safety is formally guaranteed for any control policy within the masked control policy space. We evaluate our method on two tasks from OpenAI gym and compare with three benchmarks. The following paper summarizes the results in Chapter 6:

(J4) Z. Yuan, S. Xu and M. Zhu. "All-time safety and sample-efficient meta update in online safe meta reinforcement learning". In preparation.

In addition, the following publications are not included in this dissertation:

(C4) R. Yu, Z. Yuan, M. Zhu and Z. Zhou. "Data-driven distributed state estimation and behavior modeling in sensor networks", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8192-8199, October 2020.

(C5) X. Zhang, Z. Yuan, S. Xu, Y. Lu and M. Zhu. "Secure perception-driven control of mobile robots using chaotic encryption", *American Control Conference*, pp. 2575-2580, May 2021.

(C6) Y. Jung, Z. Yuan, S. Seo, M. Zhu and S. Kim. "Learning neural processes on the fly", *International Conference on Consumer Electronics Asia*, pp.1-4, October 2022.

(C7) S. Jo, Z. Yuan and S. Kim. "Interactive storyboarding for rapid visual story generation", *International Conference on Consumer Electronics Asia*, pp.1-4, October 2022.

(C8) X. Zhang, Z. Yuan and M. Zhu. "Byzantine-tolerant federated Gaussian process regression for streaming data", *International Conference on Neural Information Processing Systems*, pp.13499-13511, December 2022.

(C9) T. Zheng, Z. Yuan, M. Nayyar, A. Wagner, M. Zhu and H. Lin. "Multi-robot-assisted human crowd evacuation using navigation velocity fields", *IEEE International Conference on Decision and Control*, pp. 2061-2066, December 2022.

(C10) Z. Yuan, T. Zheng, M. Nayyar, A. Wagner, H. Lin and M. Zhu. "Multi-robot-assisted human crowd control for emergency evacuation: A stabilization approach", *American Control Conference*, pp. 4051-4056, May 2023.

(C11) M. Nayyar, G. Paik, Z. Yuan, T. Zheng, M. Zhu, H. Lin and A. Wagner. "Characterizing evacuee behavior during a robot-guided evacuation", *IEEE International Conference on Safety, Security, and Rescue Robotics*, November, 2023.

(J5) X. Zhang, Z. Yuan, S. Xu, Y. Lu and M. Zhu. "Secure perception-driven control of mobile robots using chaotic encryption", *IEEE Transaction on Automatic Control*, 2024. To appear.

(J6) X. Zhang, Z. Yuan, and M. Zhu. "Byzantine-resilient federated online learning for Gaussian process regression", *Automatica*. Provisionally accepted.

(J7) T. Zheng, Z. Yuan, M. Nayyar, A. Wagner, M. Zhu and H. Lin. "Multi-robot-guided crowd evacuation: Two-scale modeling and control based on mean-field hydrodynamic models", *IEEE Transactions on Control Systems Technology*. Under review.

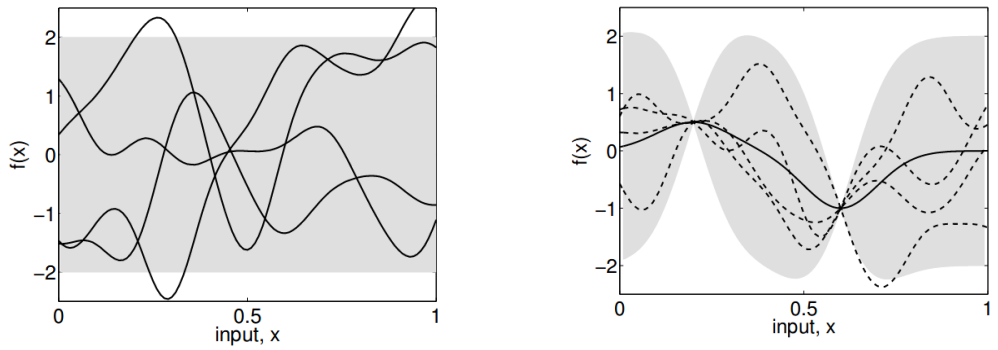
Chapter 2

Background on Gaussian process regression

In this chapter, we provide necessary background on Gaussian process regression (GPR), a powerful tool for safety-critical applications and learning-based control problems. This chapter is mainly adopted from the book [54].

GPR is an efficient nonparametric statistical learning model for supervised learning problems, i.e., the problem of learning input-output mappings from empirical data. There are two common approaches to deal with the supervised learning problem. One is to restrict the class of functions that we consider, for example by only considering linear functions of the input, which is usually referred as the frequentist approach. The second approach is to give a prior probability to every possible function, where higher probabilities are given to functions that we consider to be more likely, for example because they are smoother than other functions, which is usually referred as the Bayesian approach. GPR falls into the second approach.

In general, let $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ be the target function, where $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^{n_x}$. Given input $\mathbf{x}(t) \in \boldsymbol{\mathcal{X}}$ at time t, the corresponding output is: $y(t) = f(\mathbf{x}(t)) + e(t)$, $e(t) \sim \mathcal{N}(0, \sigma_e^2)$, where $e(t)$ is the Gaussian measurement noise. Let training data be in the form $\mathcal{D} \triangleq (\mathcal{X}, \mathbf{y})$, where $\mathcal{X} \triangleq \{\mathbf{x}(1), \cdots, \mathbf{x}(n_s)\}$ is the set of input data and $\mathbf{y} \triangleq [y(1), \cdots, y(n_s)]^T$ is the column vector aggregating the outputs. GPR aims to estimate the function over a set of test data points $\mathcal{X}_* \subset \boldsymbol{\mathcal{X}}$ using $\mathcal{D}$ by modelling η as a sample from a Gaussian process prior. For simplicity of



(a) Prior (b) Posterior

Figure 2.1: A graphical illustration of GPR

illustration, we assume $y \in \mathbb{R}$; if y is multi-dimensional, GPR is performed for each element.

A Gaussian process is a generalization of the Gaussian probability distribution. The formal definition is given as follows.

Definition 2.0.1. (page 13, [54]) A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. ■

Whereas a probability distribution describes random variables which are scalars or vectors, a stochastic process governs the properties of functions. A graphical illustration of how GPR works on some simple regression examples can be found in Figure 2.1. In Figure 2.1a, a number of sample functions are drawn at random from the prior distribution over functions specified by a particular Gaussian process which favors smooth functions. This prior represents our prior beliefs over the kinds of functions we expect to observe before seeing any data. At any value of x we can characterize the variability of the sample functions by computing the variance at that point. The shaded region denotes twice the pointwise standard deviation.

Now suppose we have two observations of the target function given by a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)\}$, and we wish to consider functions that pass through these two data points exactly. This situation is illustrated in Figure 2.1b. The dashed lines show sample functions which are consistent with $\mathcal{D}$, and the solid line depicts the mean value of such functions. Notice that the uncertainty is reduced close to the observations. The combination of the prior and the data leads to the posterior distribution over functions.

The specification of the prior is important, because it fixes the properties of the functions considered for inference. A Gaussian process is completely specified by its mean function $\mu(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$, which is also referred as the kernel function. Then the training outputs $\mathbf{y}$ and the test outputs $f(\mathcal{X}_*)$ are jointly distributed according to the prior as:

$$\begin{bmatrix} \mathbf{y} \\ f(\mathcal{X}_*) \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu(\mathcal{X}) \\ \mu(\mathcal{X}_*) \end{bmatrix}, \begin{bmatrix} k(\mathcal{X}, \mathcal{X}) + \sigma_e^2 I_{n_s} & k(\mathcal{X}, \mathcal{X}_*) \\ k(\mathcal{X}_*, \mathcal{X}) & k(\mathcal{X}_*, \mathcal{X}_*) \end{bmatrix}),$$

where $k(\mathcal{X}, \mathcal{X}_*)$ returns a matrix such that the entry at the i^{th} row and the j^{th} column is $k(\mathbf{x}(i), \mathbf{x}_*(j))$, $\mathbf{x}_*(j) \in \mathcal{X}_*$, and analogously for $k(\mathcal{X}, \mathcal{X})$ and $k(\mathcal{X}_*, \mathcal{X}_*)$. Utilizing identities of joint Gaussian distribution (page 200, [54]), we arrive at the predictive distribution of $f(\mathcal{X}_*) \mid \mathcal{X}, \mathbf{y}$, or the conditional (posterior) distribution on $f(\mathcal{X}_*)$ based on dataset $\mathcal{D}$, as $f(\mathcal{X}_*) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{X}_*|\mathcal{D}}, \Sigma_{\mathcal{X}_*|\mathcal{D}})$, where

$$\begin{aligned} \boldsymbol{\mu}_{\mathcal{X}_*|\mathcal{D}} &\triangleq \mu(\mathcal{X}_*) + k(\mathcal{X}_*, \mathcal{X})\mathring{k}(\mathcal{X}, \mathcal{X})^{-1}(\mathbf{y} - \mu(\mathcal{X})), \\ \Sigma_{\mathcal{X}_*|\mathcal{D}} &\triangleq k(\mathcal{X}_*, \mathcal{X}_*) - k(\mathcal{X}_*, \mathcal{X})\mathring{k}(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, \mathcal{X}_*), \end{aligned} \tag{2.1}$$

where $\mathring{k}(\mathcal{X}, \mathcal{X}) \triangleq k(\mathcal{X}, \mathcal{X}) + \sigma_e^2 I_{n_s}$. We refer (2.1) as full GPR. Therefore, the distributional predictions of GPR naturally provides uncertainty quantification for its outputs. With proper choice of prior covariance function, or kernel, and mild assumptions of the target function, it has been proven that GPR is able to consistently approximate any continuous function [55]. With optimal sampling in the input space and covariance functions obeying Sacks-Ylvisaker conditions of order r, the generalization error of GPR diminishes at the rate of $\mathcal{O}(n_s^{-(2r+1)/(2r+2)})$ (Section V.2, [56]). GPR has demonstrated powerful capabilities in various applications, e.g., optimization [57][37], motion planning [58], and trajectory estimation [59].

2.1 Relationships between GPs and ridge regression

Given a dataset $\mathcal{D}$, there are infinitely many functions that are consistent with it. GPR approaches this problem by putting a prior over functions. A related

viewpoint is provided by regularization theory where one seeks a trade-off between data-fit and the reproducing kernel Hilbert space (RKHS) norm of function. It is also closely related to the MAP estimator in GP prediction. In this section, we briefly introduce RKHS and discuss how GPR is closely related to ridge regression.

2.1.1 Reproducing kernel Hilbert spaces

We start with the formal definition of RKHS.

Definition 2.1.1. *(Reproducing kernel Hilbert space, page 130 [54]).* Let $\mathcal{H}_k$ be a Hilbert space of real functions f defined on an index set $\boldsymbol{\mathcal{X}}$. Then $\mathcal{H}_k$ is called a reproducing kernel Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle_k$ (and norm $\|f\|_k = \sqrt{\langle f, f \rangle}$) if there exists a function $k : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}$ with the following properties: i) for every $\mathbf{x}$, $k(\mathbf{x}, \mathbf{x}')$ as a function of $\mathbf{x}'$ belongs to $\mathcal{H}_k$, and ii) k has the reproducing property $\langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_k = f(\mathbf{x})$. ■

The RKHS uniquely determines k, and vice versa, as stated in the following theorem:

Theorem 2.1.2. *(Moore-Aronszajn theorem, page 130 [54]).* Let $\boldsymbol{\mathcal{X}}$ be an index set. Then for every positive definite function $k(\cdot, \cdot)$ on $\boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}}$ there exists a unique RKHS, and vice versa. ■

Consider a real positive semidefinite kernel $k(\mathbf{x}, \mathbf{x}')$ (see Section 2.2.1 for more discussion) with an eigenfunction expansion $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{N} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}')$ relative to a measure ν. Note that the eigenfunctions are orthogonal with respect to ν: $\int \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\nu(\mathbf{x}) = \delta_{ij}$, where δ_{ij} is the Kronecker delta. We now consider a Hilbert space comprised of linear combinations of the eigenfunctions, i.e., $f(\mathbf{x}) = \sum_{i=1}^{N} f_i \phi_i(\mathbf{x})$ with $\sum_{i=1}^{N} f_i^2 / \lambda_i < \infty$. We assert that the inner product $\langle f, g \rangle_k$ in the Hilbert space specified by kernel k between functions $f(\mathbf{x})$ and $g(\mathbf{x}) = \sum_{i=1}^{N} g_i \phi_i(\mathbf{x})$ is defined as

$$\langle f, g \rangle_k = \sum_{i=1}^{N} \frac{f_i g_i}{\lambda_i}.$$

Thus this Hilbert space is equipped with a norm $\|f\|_k$ where $\|f\|_k^2 = \langle f, f \rangle_k =$

$\sum_{i=1}^{N} f_i^2/\lambda_i$. Note that for $\|f\|_k$ to be finite the sequence of coefficients $\{f_i\}$ must decay quickly; this effectively imposes a smoothness condition on the space.

The reproducing property of this Hilbert space can be easily achieved as

$$\langle f(\cdot), k(\cdot, \mathbf{x})\rangle_k = \sum_{i=1}^{N} \frac{f_i \lambda_i \phi_i(\mathbf{x})}{\lambda_i} = f(\mathbf{x}).$$

Similarly,

$$\langle k(\mathbf{x}, \cdot), k(\mathbf{x}', \cdot)\rangle_k = \sum_{i=1}^{N} \frac{\lambda_i \phi_i(\mathbf{x}) \lambda_i \phi_i(\mathbf{x}')}{\lambda_i} = k(\mathbf{x}, \mathbf{x}').$$

2.1.2 Connection to ridge regression

Consider the Bayesian analysis of the standard linear regression model with Gaussian noise

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}, \quad y = f(\mathbf{x}) + \epsilon$$

where $\mathbf{x}$ is the input vector, $\mathbf{w}$ is the weights of the linear model, f is the function value and y is the observed target value, and $\epsilon \sim \mathcal{N}(0, \sigma_e^2)$ is additive noise following independently and identically distributed Gaussian distribution. This noise assumption together with the model directly gives rise to the likelihood, the probability density of the observations given the parameters, which is factored over cases in the training set to give

$$\begin{aligned} p(\mathbf{y}|X, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i, \mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_e} \exp(-\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma_e^2}) \\ &= \frac{1}{(2\pi\sigma_e^2)^{n/2}} \exp(-\frac{1}{2\sigma_e^2}|\mathbf{y} - X^T \mathbf{w}|^2) = \mathcal{N}(X^T \mathbf{w}, \sigma_e^2 I) \end{aligned}$$

where X is the matrix aggregating the column input vectors, $|\mathbf{z}|$ denotes the Euclidean length of vector $\mathbf{z}$. The Bayesian formalism allows us to specify a prior over the parameters, expressing our beliefs about the parameters before we look at the observations. We put a zero-mean Gaussian prior with covariance matrix

Σ_w on the weights,

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_w).$$

Inference in the Bayesian linear model is based on the posterior distribution over the weights, computed by Bayes' rule,

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$$

where the normalizing constant, also known as the marginal likelihood, is independent of the weights and given by

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

Writing only the terms from the likelihood and prior which depend on the weights, and "completing the square", we obtain

$$\begin{aligned} p(\mathbf{w}|X, \mathbf{y}) &\propto \exp(-\frac{1}{2\sigma_e^2}(\mathbf{y} - X^T\mathbf{w})^T(\mathbf{y} - X^T\mathbf{w}))\exp(-\frac{1}{2}\mathbf{w}^T\Sigma_w^{-1}\mathbf{w}) \\ &\propto \exp(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T(\frac{1}{\sigma^2}XX^T + \sigma_w^{-1})(\mathbf{w} - \bar{\mathbf{w}})) \end{aligned}$$

where $\bar{\mathbf{w}} = \sigma^{-2}(\sigma_e^{-2}XX^T + \Sigma_w^{-1})^{-1}X\mathbf{y}$. This gives the form of the posterior distribution as Gaussian with mean $\bar{\mathbf{w}}$ and covariance matrix A^{-1}

$$p(\mathbf{w}|X, \mathbf{y}) \sim \mathcal{N}(\bar{\mathbf{w}} = \frac{1}{\sigma_e^2}A^{-1}X\mathbf{y}, A^{-1}),$$

where $A = \sigma_e^{-2}XX^T + \Sigma_w^{-1}$. Thus the predictive distribution for $f(\mathbf{x}_*)$ is given by

$$p(f(\mathbf{x}_*)|X, \mathbf{y}) = \int p(f(\mathbf{x}_*)|\mathbf{w})p(\mathbf{w}|X, \mathbf{y})d\mathbf{w} = \mathcal{N}(\frac{1}{\sigma_e^2}\mathbf{x}_*^TA^{-1}X\mathbf{y}, \mathbf{x}_*^TA^{-1}\mathbf{x}_*).$$

Using linear model can suffers from limited expressiveness. A simple idea to overcome this problem is to first project the inputs into some high dimensional space using a set of basis functions and then apply the linear model in this space instead of directly on the inputs themselves. For example, a scalar input x could be projected into the space of powers of x: $\phi(x) = (1, x, x^2, x^3, \cdots)^T$ to implement

polynomial regression. In general, we introduce the function $\phi(\mathbf{x})$ which maps a n_x-dimensional input vector $\mathbf{x}$ into an n_ϕ-dimensional feature space. Further let $\Phi(X)$ be the aggregation of columns $\phi(\mathbf{x})$ for all cases in the training set. Now the model is

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}. \tag{2.2}$$

Following analogous analysis to the standard linear model, the predictive distribution becomes

$$f(\mathbf{x}_*)|X, \mathbf{y} \sim \mathcal{N}(\frac{1}{\sigma_e^2}\phi(\mathbf{x}_*)^T A^{-1}\Phi(X)\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*))$$

where $A = \sigma_e^{-2}\Phi(X)\Phi(X)^T + \Sigma_w^{-1}$. This can also be rewritten as

$$\begin{aligned} f(\mathbf{x}_*)|X, \mathbf{y} \sim \mathcal{N}(&\phi(\mathbf{x}_*)^T\Sigma_w\Phi(X)(K + \sigma_e^2 I)^{-1}\mathbf{y}, \\ &\phi(\mathbf{x}_*)^T\Sigma_w\phi(\mathbf{x}_*) - \phi(\mathbf{x}_*)^T\Sigma_w\Phi(X)(K + \sigma_e^2 I)^{-1}\Phi(X)^T\Sigma_w\phi(\mathbf{x}_*)). \end{aligned}$$

Define $k(\mathbf{x}, \mathbf{x}) = \phi(\mathbf{x}_*)^T\Sigma_w\phi(\mathbf{x}_*)$. Then the posterior mean and variance of $f(\mathbf{x}_*)$ have the expression the same as (2.1) when the prior mean is set to zero: $\mu(\mathbf{x}_*) = 0$. Notice that for this model, the mean of the posterior distribution $p(f(\mathbf{x}_*)|X, \mathbf{y})$ is also its mode, which is also the *maximum a posteriori* (MAP) estimate of $f(\mathbf{x}_*)$.

The model in (2.2) also enables learning f through kernel ridge regression. Consider the functional

$$J[f] = \frac{1}{2}\|f\|_k^2 + \frac{1}{2\sigma_e^2}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$$

which uses a squared error data-fit term (corresponding to the negative log likelihood of a Gaussian noise model with variance σ_e^2). By penalizing the RKHS norm, this formulation automatically limits the solution within the RKHS induced by kernel $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x})$, or $\Sigma_w = I$. The representer theorem shows that each minimizer $f \in \mathcal{H}_k$ of $J[f]$ has the form $f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i k(\mathbf{x}, \mathbf{x}_i)$. Hence substituting

$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$ and using $\langle k(\cdot, \mathbf{x}_i), k(\cdot, \mathbf{x}_j)\rangle_k = k(\mathbf{x}_i, \mathbf{x}_j)$ we obtain

$$\begin{aligned} J[\mathbf{a}] &= \frac{1}{2}\boldsymbol{\alpha}^T k(\mathcal{X}, \mathcal{X})\boldsymbol{\alpha} + \frac{1}{2\sigma_e^2}|\mathbf{y} - k(\mathcal{X}, \mathcal{X})\boldsymbol{\alpha}|^2 \\ &= \frac{1}{2}\boldsymbol{\alpha}^T (k(\mathcal{X}, \mathcal{X}) + \frac{1}{\sigma_e^2} k(\mathcal{X}, \mathcal{X})^2)\boldsymbol{\alpha} - \frac{1}{\sigma_e^2}\mathbf{y}^T k(\mathcal{X}, \mathcal{X})\boldsymbol{\alpha} + \frac{1}{2\sigma_e^2}\mathbf{y}^T\mathbf{y}. \end{aligned}$$

Minimizing J by differentiating w.r.t. the vector of coefficients $\boldsymbol{\alpha}$ we obtain $\hat{\boldsymbol{\alpha}} = (k(\mathcal{X}, \mathcal{X}) + \sigma_e^2 I)^{-1}\mathbf{y}$, so that the prediction for a test point $\mathbf{x}_*$ is $\hat{f}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathcal{X})^T (k(\mathcal{X}, \mathcal{X}) + \sigma_e^2 I)^{-1}\mathbf{y}$, which is exactly the form of the predictive mean in (2.1) when $\mu(\mathbf{x}_*) = 0$.

In conclusion, GPR is consistent with the MAP estimator as well as the optimal solution to kernel ridge regression. Yet uncertainty in predictions as well as marginal likelihood are omitted in the later two methods, compared to GPR.

2.2 Covariance functions

Covariance function is the crucial ingredient in a Gaussian process predictor because it encodes our assumptions about the function which we wish to learn. It is a basic assumption that points with inputs close to each other should have similar target values, and thus training points that are near to a test point should be informative about the prediction at that point. This is particularly useful in learning continuous functions. Under the view of Gaussian process, it is the covariance function that defines nearness or similarity.

An arbitrary function of input pairs $\mathbf{x}$ and $\mathbf{x}'$ will not, in general, be a valid function. This section aims to give examples and properties of some commonly-used covariance functions.

2.2.1 Terminologies

A *stationary* covariance function is a function of $\mathbf{x} - \mathbf{x}'$. Thus it is invariant to translations in the input space. If further the covariance function is a function only of $|\mathbf{x} - \mathbf{x}'|$ then it is called *isotropic*; it is thus invariant to all rigid motions. Isotropic covariance functions are also known as *radial basis functions* since it is only a function of $r = |\mathbf{x} - \mathbf{x}'|$.

A *dot product* covariance function depends only on $\mathbf{x}$ and $\mathbf{x}'$ through $\mathbf{x} \cdot \mathbf{x}'$. It is invariant to a rotation of the coordinates about the origin, but not translations.

A general name for a function k of two arguments mapping a pair of inputs $\mathbf{x}, \mathbf{x}' \in \boldsymbol{\mathcal{X}}$ into $\mathbb{R}$ is a *kernel*. A real kernel is said to be symmetric if $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$; clearly covariance functions must be symmetric from the definition.

Given a set of inputs $\mathcal{X} = \{\mathbf{x}(t)\}_{t=1}^{n}$, we can compute the *Gram matrix* $k(\mathcal{X}, \mathcal{X})$ whose entry at the i-th row j-th column is $k(\mathbf{x}(i), \mathbf{x}(j))$. If k is a covariance function we call the matrix $k(\mathcal{X}, \mathcal{X})$ the *covariance matrix*.

A real $n \times n$ matrix $k(\mathcal{X}, \mathcal{X})$ which satisfies $Q(\mathbf{v}) = \mathbf{v}^T k(\mathcal{X}, \mathcal{X})\mathbf{v} \geqslant 0$ for all vectors $\mathbf{v} \in \mathbb{R}^n$ is called *positive semidefinite* (PSD). A symmetric matrix is PSD if and only if all of its eigenvalues are non-negative. A Gram matrix corresponding to a general kernel function need not be PSD, but the Gram matrix corresponding to a covariance function is PSD.

A *kernel* is said to be *positive semidefinite* (PSD) if

$$\int k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\nu(\mathbf{x}) d\nu(\mathbf{x}') \geqslant 0$$

for all $f \in L_2(\boldsymbol{\mathcal{X}}, \nu)$, where ν denotes a measure. Equivalenty a kernel function which give rise to PSD Gram matrices for any choice of $n \in \mathbb{N}$ and $\mathcal{D}$ is positive semidefinite.

2.2.2 Eigenfunction analysis of kernels

It turns out that GPR can be viewed as Bayesian linear regression with a possibly infinite number of basis functions (see Section 2.1.2). One possible basis set is the *eigenfunctions* of the covariance function. A function $\phi(\cdot)$ that obeys the integral equation

$$\int k(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}) d\nu(\mathbf{x}) = \lambda \phi(\mathbf{x})$$

is called an eigenfunction of kernel k with eigenvalue λ with respect to measure ν. In general, there are infinite number of eigenfunctions, which we label $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x})$, $\cdots$. We assume the ordering is chosen such that $\lambda_1 \geqslant \lambda_2 \geqslant \cdots$. The eigenfunctions are orthogonal with respect to ν and can be chosen to be normalized

so that $\int \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\nu(\mathbf{x}) = \delta_{ij}$ where δ_{ij} is the Kronecker delta.

Mercer's theorem below allows us to express the kernel k in terms of the eigenvalues and eigenfunctions.

Theorem 2.2.1. *(Mercer's theorem, page 96 [54]).* Let $(\boldsymbol{\mathcal{X}}, \nu)$ be a finite measure space and $k \in L_\infty(\boldsymbol{\mathcal{X}}^2, \nu^2)$ be a kernel such that $T_k : L_2(\boldsymbol{\mathcal{X}}, \nu) \to L_2(\boldsymbol{\mathcal{X}}, \nu)$ is positive definite. Let $\phi_i \in L_2(\boldsymbol{\mathcal{X}}, \nu)$ be the normalized eigenfunctions of T_k associated with the eigenvalues $\lambda_i > 0$. Then:

1. the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ are absolutely summable;

2. $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x})\phi_i^*(\mathbf{x}')$ holds ν^2 almost everywhere, where the series converges absolutely and uniformly ν^2 almost everywhere. □

This decomposition is just the infinite-dimensional analogue of the diagonalization of a Hermitian matrix. Note that the sum may terminate at some value $N \in \mathbb{N}$, or the sum may be infinite. This gives the following definition.

Definition 2.2.2. A degenerate *kernel* has only a finite number of non-zero eigenvalues. □

A degenerate kernel is also said to have finite rank. If a kernel is not degenerate it is said to be nondegenerate.

2.2.3 Examples of covariance functions

Figure 2.2 below summarizes several commonly-used covariance functions, where the covariances are written either as a function of $\mathbf{x}$ and $\mathbf{x}'$, or as a function of $r = |\mathbf{x}-\mathbf{x}'|$, the two columns marked as 'S' and 'ND' indicate whetherthe covariance functions are stationary and nondegenerate respectively.

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

Figure 2.2: Summary of several commonly-used covariance functions

Chapter 3

Distributed Gaussian process regression

3.1 Introduction

In this chapter, we develop a class of distributed machine learning algorithms for machine learning in multiple robots. In particular, we study Gaussian process regression (GPR) [54] because it has high expressive power and high consistency in function approximation. Meanwhile, its mathematical formulation applying Bayesian inference naturally provides uncertainty quantification, which is highly useful for safety-critical applications.

GPR scales as $\mathcal{O}(n_s^3)$ in computational complexity and $\mathcal{O}(n_s^2)$ in memory (page 171, [54]), which prohibits applications with large datasets. There are multiple sparse approximation methods for large datasets. One major class of approximation methods, which is also referred as global approximation, tackles the computational complexity by achieving the sparsity of the Gram matrix. Methods include using a subset of data to approximate the whole training dataset, designing a sparse kernel, and sparsifying the Gram matrix. The best possible result can be achieved by global approximation algorithms is $\mathcal{O}(m_s^3)$ in computational complexity and $\mathcal{O}(m_s^2)$ in memory, where $m_s \ll n_s$ is the number of inducing points or the size of a subset of training data. More details about global approximation can be found in the recent survey paper [60]. In the community of Geostatistics, Nearest-

neighbor GPR [61] is applied [62][63], where the predictions are made only using the training data of the nearest input. It requires only $\mathcal{O}(n_s)$ in both memory and (worst-case) computation.

Centralized implementation of GPR is not suitable for networks of agents due to poor scalability in data size, high cost in communication and memory, and fragility to single-point failures. There have been studies on distributed GPR over server-client architecture, which is also referred to as divide-and-conquer approach or local approximations [60]. In the server-client architecture, a server acts as the centralized entity that partitions a dataset and assigns each subset of the data to computing units (clients). The clients perform training independently and send their learning results to the server for post-processing. These methods speed up the training process and are able to scale to arbitrarily large datasets. Communication budget constraint is considered in [64] by reducing the dimensionality of transmitted data to approximate the whole dataset. Sparse approximation of full GPR is used in [64] to further relieve the communication overhead. Notice that the server-client architecture requires each client being well-connected with the server, and is not robust to the failure of the server. Paper [65] decentralizes sparse approximations of full GPR for fixed datasets over complete communication graphs. A distributed algorithm is also proposed to deal with fixed and sparse graphs. However, this chapter considers offline learning with static datasets on the agents and does not provide theoretic guarantee on the distributed algorithm.

Our work is related to multi-agent regression using kernel methods and basis functions. Papers [66][67][68] study offline learning, where all training data is provided before learning, using kernel methods. Papers [69][70][71] study online learning, where training data is collected successively by mobile robots, using basis functions. In particular, they approximate the unknown functions with a linear combination of a finite number of known basis functions. This reduces the problem into a parameter estimation problem. From the perspective of regression, the problem investigated in [69] is equivalent to selecting the centers of a finite number of basis functions defined by Voronoi partition. In contrast, this chapter considers online learning of abstract agents using Gaussian processes, where the unknown function is modelled as a sample from a distribution of functions.

Contribution statement. We consider the problem where a group of agents

aim to collaboratively learn a common static latent function through streaming data. We propose Lightweight Distributed Gaussian Process Regression (LiDGPR) algorithm for the agents to solve the problem. More specifically, each agent independently runs agent-based GPR using local streaming data to predict the test points of interest; then the agents collaboratively execute distributed GPR to obtain global predictions over a common sparse set of test points; finally, each agent fuses the results from distributed GPR with those from agent-based GPR to refine its predictions. Our analysis of the transient and steady-state performances in predictive variance and error reveals that through communication agents whose data samples have lower dispersion (or observation noise has lower variance) help improve the performance of the agents whose data samples have higher dispersion (or observation noise has higher variance). The improvements in learning performances are in the sense of Pareto, i.e., some agents' performances improve without sacrificing other agents' performances. In summary, our major contributions are two-fold:

- We develop LiDGPR that is cognizant of agents' limited capabilities in communication, computation and memory.
- We analyze the predictive mean and variance of LiDGPR and quantify the improvements of the agents' learning performances resulted from inter-agent communication.

Monte Carlo simulation is conducted to evaluate the developed algorithm.

Notations: We use lower-case letters, e.g., a, to denote scalars, bold letters, e.g., $\boldsymbol{a}$, to denote vectors; we use upper-case letters, e.g., A, to denote matrices, calligraphic letters, e.g., $\mathcal{A}$, to denote sets, and bold calligraphic letters, e.g., $\boldsymbol{\mathcal{A}}$, to denote spaces. For any vector $\boldsymbol{a}$, we use a_i to denote the i-th entry of $\boldsymbol{a}$. For any matrix A, we denote a_{ij} as the entry at i-th row j-th column. Denote $I_n \in \mathbb{R}^{n\times n}$ the n-by-n-dimensional identity matrix, $\mathbf{1}_n \in \mathbb{R}^n$ the n-dimensional column vector with all 1's, i.e., $[1, \cdots, 1]^T$, and $\mathbf{0}_n$ analogously.

We use superscript $(\cdot)^{[i]}$ to distinguish the local values of agent i, and $(\cdot)^{\max}$ ($(\cdot)^{min}$) denote the maximum (minimum) of the local values, e.g., $a^{\max} \triangleq \max_{i\in\mathcal{V}} a^{[i]}$. We denote superscript $(\cdot)^T$ the transpose of a vector or matrix, bracket $[\cdot]_{\mathcal{E}}$ the column vector with elements satisfying event $\mathcal{E}$. Denote $\mathbb{E}_a[\cdot]$ the expectation taken

over the distribution of random variable a, and $P\{\cdot\}$ a distribution. We use $\mathcal{O}(\cdot)$ to denote the conventional Big O notation, i.e., $\mathcal{O}(g(t))$ represents the limiting behavior of some function $f(t)$ if $\lim_{t\to\infty} \frac{f(t)}{g(t)} = a$ for some constant $a > 0$.

We use $\succeq$ ($\preceq$) to denote element-wise comparison between two vectors, i.e., for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{a} \succeq (\preceq) \boldsymbol{b}$ if and only if $a_i \geqslant (\leqslant) b_i$ for all $i = 1, \cdots, n$. Operation $|\boldsymbol{a}|$ takes the absolute values element-wise on vector $\boldsymbol{a}$, $|\mathcal{A}|$ returns the cardinality of set $\mathcal{A}$, $\|\boldsymbol{a}\|_\infty \triangleq \max_j |a_j|$ for any vector $\boldsymbol{a}$. Define the distance metric $\rho(\boldsymbol{z}, \boldsymbol{z}') \triangleq \|\boldsymbol{z} - \boldsymbol{z}'\|$, the point to set distance as $\rho(\boldsymbol{z}, \mathcal{Z}) \triangleq \inf_{\boldsymbol{z}' \in \mathcal{Z}} \rho(\boldsymbol{z}, \boldsymbol{z}')$. Define $\text{proj}(\boldsymbol{z}, \mathcal{Z}) \triangleq \{\boldsymbol{z}' \in \boldsymbol{\mathcal{Z}} | \rho(\boldsymbol{z}, \boldsymbol{z}') = \rho(\boldsymbol{z}, \mathcal{Z})\}$ the projection set of point $\boldsymbol{z}$ onto set $\mathcal{Z}$. Denote the supremum of a function η as $\|\eta\|_{\boldsymbol{\mathcal{Z}}} \triangleq \sup_{\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}} |\eta(\boldsymbol{z})|$.

3.2 Problem statement

Network model. Consider a network of agents represented by a directed time-varying communication graph $\mathcal{G}(t) \triangleq (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} \triangleq \{1, \cdots, n\}$ represents the agent set, and $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set at time t. Notice that $(i, j) \in \mathcal{E}(t)$ if and only if agent i can receive messages from agent j at time t. Define the set of the neighbors of agent i at time t as $\mathcal{N}^{[i]}(t) \triangleq \{j \in \mathcal{V} : (i, j) \in \mathcal{E}(t), \text{ and } j \neq i\}$. The matrix $A(t) \in \mathbb{R}^{n \times n}$ represents the adjacency matrix of $\mathcal{G}(t)$ where $a_{ij}(t) \neq 0$ if $(i, j) \in \mathcal{E}(t)$. We make the following standard assumptions [72] about the network topology:

Assumption 3.2.1. (Periodical Strong Connectivity). There exists positive integer $b \geqslant 1$ such that, for all time instant $t \geqslant 0$, the directed graph $(\mathcal{V}, \mathcal{E}(t) \cup \mathcal{E}(t+1) \cup \cdots \cup \mathcal{E}(t+(b-1)))$ is strongly connected. ■

This guarantees the information of each agent can reach any other agents in the network within finite time.

Assumption 3.2.2. (Balanced Communication). It holds that $\mathbf{1}_n^T A(t) = \mathbf{1}_n^T$ and $A(t)\mathbf{1}_n = \mathbf{1}_n$, for all $t \geqslant 0$. ■

In the consensus literature, the first part of Assumption 3.2.2 is called column stochasticity and is a standard sufficient condition to reach consensus. The second part is called row stochasticity and is needed to guarantee average consensus.

Assumption 3.2.3. (Non-degeneracy). There exists a constant $\alpha > 0$ such that $a_{ii}(t) \geqslant \alpha$ and $a_{ij}(t) \in \{0\} \cup [\alpha, 1]$, $i \neq j$, for all $t \geqslant 0$. ■

That is, each agent assigns nontrivial weights on information from itself and its neighbors.

Observation model. At each time instant t, each agent independently observes the outputs of a continuous common static latent function $\eta : \boldsymbol{\mathcal{Z}} \to \boldsymbol{\mathcal{Y}}$ with zero-mean Gaussian noise, where $\boldsymbol{\mathcal{Z}} \subseteq \mathbb{R}^{n_z}$ is the compact input space for η. The observation model is given by

$$y^{[i]}(t) = \eta(\boldsymbol{z}^{[i]}(t)) + e^{[i]}(t), \quad e^{[i]}(t) \sim \mathcal{N}(0, (\sigma_e^{[i]})^2), \tag{3.1}$$

where $\boldsymbol{z}^{[i]}(t) \in \boldsymbol{\mathcal{Z}}$ is the input of η from agent i at time t, $y^{[i]}(t) \in \boldsymbol{\mathcal{Y}}$ is the observation of agent i, and $e^{[i]}(t)$ is independent Gaussian noise. Note that we do not assume that input $\boldsymbol{z}^{[i]}(t)$ follows any distribution, which is a standard assumption in statistical learning [51]. We let $\boldsymbol{\eta}(\mathcal{Z})$ return a column vector $[\eta(\boldsymbol{z})]_{\boldsymbol{z}\in\mathcal{Z}}$, and similarly for other functions. For notational simplicity, it is assumed that the output space $\boldsymbol{\mathcal{Y}} \subseteq \mathbb{R}$ because multi-dimensional observations can always be decomposed as aggregation of one-dimensional observations.

Problem Statement. The objective of this chapter is to design a distributed algorithm for the agents to learn the common static latent function η via streaming data $\{y^{[i]}(t), \boldsymbol{z}^{[i]}(t)\}_{t\geqslant 1}$. The challenges of the problem stem from the fact that the training dataset is monotonically expanding due to incremental sampling while the agents have limited resources in communication, computation and memory.

The followings are examples of potential applications of this formulation. One example can be a group of mobile robots deployed in a vast open area to collaboratively monitor a static signal, such as temperature or wind field (see the case study in Section 4.5). Other examples includes the learning of the dynamics of a moving target using a network of static sensors [73]. In addition to robotic applications, this formulation also applies to profit predictions in marketing and wheat crop prediction [74].

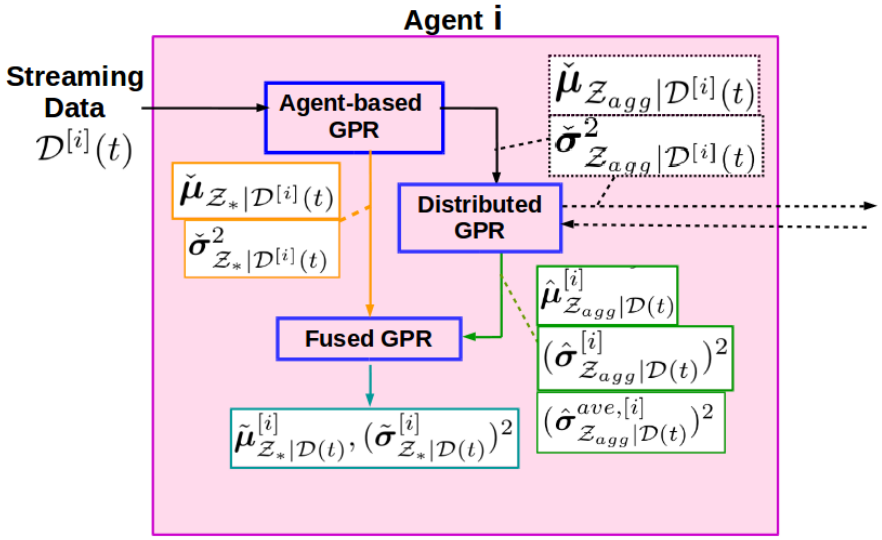


Figure 3.1: Flow diagram of LiDGPR in one iteration

3.3 Lightweight distributed GPR

In this section, we propose the Lightweight distributed GPR (LiDGPR) algorithm which allows the agents to collaboratively learn the static latent function subject to limited resources. As shown in Figure 3.1, LiDGPR is composed of three parts: (i) agent-based GPR (Algorithm 2), where the agents make their own predictions of η over a given set of points of interest $\mathcal{Z}_* \subseteq \boldsymbol{\mathcal{Z}}$ using local streaming data $\mathcal{D}^{[i]}(t) \triangleq (\mathcal{Z}^{[i]}(t), \boldsymbol{y}^{[i]}(t))$, where $\mathcal{Z}^{[i]}(t) \triangleq \{\boldsymbol{z}^{[i]}(1), \cdots, \boldsymbol{z}^{[i]}(t)\}$ aggregates local input data and $\boldsymbol{y}^{[i]}(t) \triangleq [y^{[i]}(1), \cdots, y^{[i]}(t)]^T$ aggregates the outputs; (ii) distributed GPR (Algorithm 3), where the agents integrate their predictions with those of their neighbors on a pre-defined common set $\mathcal{Z}_{agg} \subset \mathcal{Z}_*$ and estimate the predictions on this set given the global training datatset $\mathcal{D}(t) \triangleq \cup_{i\in\mathcal{V}}\mathcal{D}^{[i]}(t)$; (iii) fused GPR (Algorithm 4), where the agents refine the predictions on $\mathcal{Z}_*$ by fusing the results from distributed GPR with those from agent-based GPR. The formal statement of LiDGPR is presented in Algorithm 1. For each iteration t, each agent i collects data online and updates local dataset $\mathcal{D}^{[i]}(t) = \mathcal{D}^{[i]}(t-1) \cup (\boldsymbol{z}^{[i]}(t), y^{[i]}(t))$, and then sequentially executes agent-based GPR, distributed GPR, and fused GPR.

Algorithm 1 LiDGPR

1: **procedure**
2: **Input:** network of agents: $\mathcal{V}$; test inputs: $\mathcal{Z}_*$; common inputs: $\mathcal{Z}_{agg}$; adjacency matrix: $A(t)$; prior mean function: μ; kernel function: k; noise variance: $(\sigma_e^{[i]})^2$ for $i \in \mathcal{V}$.
3: **Init:** $\mathcal{D}(0) = \emptyset$, $\boldsymbol{\xi}^{[i]}(0) = \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(-1) = \mathbf{0}_{|\mathcal{Z}_{agg}|}$, $\boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(0) = \frac{1}{\sigma_f^2}\mathbf{1}_{|\mathcal{Z}_{agg}|}$, $\boldsymbol{\theta}^{[i]}(0) = \boldsymbol{r}_{\boldsymbol{\theta}}^{[i]}(0)$, $\boldsymbol{\lambda}^{[i]}(0) = \boldsymbol{r}_{\boldsymbol{\lambda}}^{[i]}(0)$, σ_f^2 satisfying (3.4).
4: **for** $t = 1, 2, \cdots$ **do**
5: **for** $i \in V$ **do**
6: $\mathcal{D}^{[i]}(t) = \mathcal{D}^{[i]}(t-1) \cup (\boldsymbol{z}^{[i]}(t), y^{[i]}(t))$
{Agent-based GPR}
7: $\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)} = \text{aGPR}(\mathcal{D}^{[i]}(t))$
{Distributed GPR}
8: $\hat{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)}, (\hat{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2, (\hat{\boldsymbol{\sigma}}^{ave,[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2 = \text{dGPR}(\check{\boldsymbol{\mu}}_{\mathcal{Z}_{agg}|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_{agg}|\mathcal{D}^{[i]}(t)})$
{Fused GPR}
9: $\tilde{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)}, (\tilde{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)})^2 = \text{fGPR}(\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \hat{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)}, (\hat{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2, (\hat{\boldsymbol{\sigma}}^{ave,[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2)$
10: **end for**
11: **end for**
12: **end procedure**

3.3.1 Agent-based GPR

To reduce computational complexity, we implement Nearest-neighbor GPR as agent-based GPR. Instead of feeding the whole training dataset to full GPR in (2.1), agent-based GPR only feeds the nearest input $\boldsymbol{z}_*^{[i]}(t) \in \text{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}(t))$, and the corresponding output $y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)}$, i.e., $(\boldsymbol{z}_*^{[i]}(t), y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)})$, to (2.1) for each $\boldsymbol{z}_* \in \mathcal{Z}_*$. If there are repeated observations over $\boldsymbol{z}_*^{[i]}(t)$, $y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)}$ can be the average of the observations. The predictive mean and variance for each $\boldsymbol{z}_*$ are given in Line 4 and 5 of agent-based GPR. Agent-based GPR returns $\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)} \triangleq [\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}]_{\boldsymbol{z}_* \in \mathcal{Z}_*}$ and $\check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)} \triangleq [\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}]_{\boldsymbol{z}_* \in \mathcal{Z}_*}$.

3.3.2 Distributed GPR

Note that each agent only maintains $\mathcal{D}^{[i]}(t)$, a portion of the global training dataset $\mathcal{D}(t)$. Besides collecting more data, information exchanges between the agents

Algorithm 2 Agent-based GPR

1: **procedure** AGPR($\mathcal{D}^{[i]}(t)$)
2: **for** $\boldsymbol{z}_* \in \mathcal{Z}_*$ **do**
3: choose $\boldsymbol{z}_*^{[i]}(t) \in \text{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}(t))$
4: $\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \mu(\boldsymbol{z}_*) + k(\boldsymbol{z}_*, \boldsymbol{z}_*^{[i]}(t))\mathring{k}(\boldsymbol{z}_*^{[i]}(t), \boldsymbol{z}_*^{[i]}(t))^{-1}(y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} - \mu(\boldsymbol{z}_*^{[i]}(t)))$
5: $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = k(\boldsymbol{z}_*, \boldsymbol{z}_*) - k(\boldsymbol{z}_*, \boldsymbol{z}_*^{[i]}(t))\mathring{k}(\boldsymbol{z}_*^{[i]}(t), \boldsymbol{z}_*^{[i]}(t))^{-1}k(\boldsymbol{z}_*^{[i]}(t), \boldsymbol{z}_*)$
6: **end for**
7: **Return** $\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}$
8: **end procedure**

could enhance the learning performance upon agent-based GPR. However, limited communication budget prevents the agents from sharing $\mathcal{D}^{[i]}(t)$, whose size monotonically increases. Hence, we develop distributed GPR where the agents communicate with the predictive means and variances over a common set $\mathcal{Z}_{agg}$.

In order to deal with large dataset using GPR, local approximation methods such as Product of Expert (PoE) [75] and Bayesian Committee Machine [76] are proposed to factorize the training process. We consider the following PoE aggregation model for predicting each $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$:

$$\check{\mu}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)} = \frac{(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2}{n} \sum_{i=1}^{n} \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}, \tag{3.2}$$

$$(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^{-2} = \frac{1}{n} \sum_{i=1}^{n} \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}, \tag{3.3}$$

which is consistent with full GPR (Proposition 2, [77]).

The two summations in (3.2) and (3.3) involve the global training dataset. To decentralize the computation, we consider the computation of the two summations as a dynamic average consensus problem and use FODAC in [78] to track the time-varying sums in a distributed manner. Denote $\big(\boldsymbol{\theta}^{[i]}(t), \boldsymbol{\xi}^{[i]}(t), \boldsymbol{\lambda}^{[i]}(t)\big)$ the consensus states of agent i. Each entry of the consensus states, $\big(\theta^{[i]}_{\boldsymbol{z}_*}(t), \xi^{[i]}_{\boldsymbol{z}_*}(t), \lambda^{[i]}_{\boldsymbol{z}_*}(t)\big)$, estimates $\big(\frac{1}{n}\sum_{i=1}^{n} \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)},\ \frac{1}{n}\sum_{i=1}^{n} \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)},\ \frac{1}{n}\sum_{i=1}^{n} \check{\sigma}^{2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\big)$ for each $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$. State $\boldsymbol{\lambda}^{[i]}(t)$ is used as one of the criteria for applying fusion between agent-based GPR and distributed GPR in fused GPR. The dynamics of FODAC is shown in Line 4, Line 6 and Line 8 of distributed GPR respectively for

Algorithm 3 Distributed GPR

1: **procedure** DGPR($\check{\boldsymbol{\mu}}_{\mathcal{Z}_{agg}|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_{agg}|\mathcal{D}^{[i]}(t)}$)
2: **for** $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$ **do**
{Dynamic average consensus}
3: $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$
4: $\theta^{[i]}_{\boldsymbol{z}_*}(t) = \theta^{[i]}_{\boldsymbol{z}_*}(t-1) + \sum_{j\neq i} a_{ij}(t-1)(\theta^{[j]}_{\boldsymbol{z}_*}(t-1) - \theta^{[i]}_{\boldsymbol{z}_*}(t-1)) + \Delta r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t-1)$
5: $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$
6: $\xi^{[i]}_{\boldsymbol{z}_*}(t) = \xi^{[i]}_{\boldsymbol{z}_*}(t-1) + \sum_{j\neq i} a_{ij}(t-1)(\xi^{[j]}_{\boldsymbol{z}_*}(t-1) - \xi^{[i]}_{\boldsymbol{z}_*}(t-1)) + \Delta r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1)$
7: $r^{[i]}_{\boldsymbol{\lambda},\boldsymbol{z}_*}(t) = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$
8: $\lambda^{[i]}_{\boldsymbol{z}_*}(t) = \lambda^{[i]}_{\boldsymbol{z}_*}(t-1) + \sum_{j\neq i} a_{ij}(t-1)(\lambda^{[j]}_{\boldsymbol{z}_*}(t-1) - \lambda^{[i]}_{\boldsymbol{z}_*}(t-1)) + \Delta r^{[i]}_{\boldsymbol{\lambda},\boldsymbol{z}_*}(t-1)$
{Prediction on $\mathcal{Z}_{agg}$}
9: $(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = (\xi^{[i]}_{\boldsymbol{z}_*}(t))^{-1}$
10: $\hat{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*}(t)$
11: $(\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \lambda^{[i]}_{\boldsymbol{z}_*}(t)$
12: **end for**
13: Send $(\boldsymbol{\theta}^{[i]}(t), \boldsymbol{\xi}^{[i]}(t), \boldsymbol{\lambda}^{[i]}(t))$ to $\mathcal{N}^{[i]}(t)$
14: **Return** $\hat{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)}, (\hat{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2, (\hat{\boldsymbol{\sigma}}^{ave,[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2$
15: **end procedure**

each consensus state, where $\Delta r(t) \triangleq r(t) - r(t-1)$ denotes the temporal change of the signal r. Specifically, $\theta^{[i]}_{\boldsymbol{z}_*}(t)$ tracks the average of the signal $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$ defined in Line 3 among the agents. In particular, agent i computes a convex combination of $\theta^{[j]}_{\boldsymbol{z}_*}(t)$ for $j \in \{i\} \cup \mathcal{N}^{[i]}(t-1)$, and then adds the combination into the temporal change of $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$. The update laws for $\xi^{[i]}_{\boldsymbol{z}_*}(t)$ and $\lambda^{[i]}_{\boldsymbol{z}_*}(t)$ are similar. The updated states are sent to each agent in $\mathcal{N}^{[i]}(t)$ as in Line 13. Notice that consensus is not necessarily reached at each time t. We will show that consensus is reached in an asymptotic way in Section 3.4.2.

3.3.3 Fused GPR

Fused GPR aims to refine predictions of $\boldsymbol{\eta}(\mathcal{Z}_*)$ by integrating agent-based GPR with distributed GPR. The goal is to obtain an estimate of the predictive distribution $P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$ for each $\boldsymbol{z}_* \in \mathcal{Z}_*$. Note that distributed GPR obtains new estimates of $\boldsymbol{\eta}(\mathcal{Z}_{agg})$ by combining results from each agent through convex combination. It can return results with more uncertain predictions, and these pre-

Algorithm 4 Fused GPR

1: **procedure** FGPR($\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \hat{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)}, (\hat{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2, (\hat{\boldsymbol{\sigma}}^{ave,[i]}_{\mathcal{Z}_{agg}|\mathcal{D}(t)})^2$)
2: $\quad \mathcal{Z}^{[i]}_{agg}(t) = \{\boldsymbol{z}_{agg} \in \mathcal{Z}_{agg} | (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2 < \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}$ and $(\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2 < \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}\}$
3: $\quad$ **if** $\mathcal{Z}^{[i]}_{agg}(t) == \emptyset$ **then**
4: $\quad\quad$ **Return** $\tilde{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)} = \check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, (\tilde{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)})^2 = \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}$
5: $\quad$ **end if**
6: $\quad$ **for** $\boldsymbol{z}_* \in \mathcal{Z}_*$ **do**
7: $\quad\quad$ choose $\boldsymbol{z}^{[i]}_{agg*}(t) \in \text{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}_{agg}(t))$
8: $\quad\quad$ $g^{[i]}_{\boldsymbol{z}_*}(t) = g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*}(t))$
9: $\quad\quad$ $v^{[i]}_{\boldsymbol{z}_*}(t) = g^{[i]}_{\boldsymbol{z}_*}(t)\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$
10: $\quad\quad$ $\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)} = \hat{\mu}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)} - \check{\mu}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$
11: $\quad\quad$ $\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} = v^{[i]}_{\boldsymbol{z}_*}(t)\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)} + \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$
12: $\quad\quad$ $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} + (v^{[i]}_{\boldsymbol{z}_*}(t))^2\big((\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 - \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}\big)$
13: $\quad$ **end for**
14: $\quad$ **Return** $\tilde{\boldsymbol{\mu}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)}, (\tilde{\boldsymbol{\sigma}}^{[i]}_{\mathcal{Z}_*|\mathcal{D}(t)})^2$
15: **end procedure**

dictions should be ignored. The set of inputs predicted by distributed GPR with lower uncertainty is defined as $\mathcal{Z}^{[i]}_{agg}(t)$ (Line 2 of fused GPR). Set $\mathcal{Z}^{[i]}_{agg}(t)$ is the set of inputs $\boldsymbol{z}_{agg} \in \mathcal{Z}_{agg}$, where the two variance estimates from distributed GPR, the estimates of $(\check{\sigma}^{(agg)}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^{-2}$ in (3.3) and the estimates of $\frac{1}{n}\sum_{i=1}^n \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}$, are lower than $\check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}$ from agent-based GPR. If this set is empty (Line 3-4 in fused GPR), the results from distributed GPR are ignored and those from agent-based GPR are used. Otherwise, $P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$ is estimated as follows.

Notice that for all $\boldsymbol{z}_{agg} \in \mathcal{Z}^{[i]}_{agg}(t)$, we have

$$P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\} = \int P\{\eta(\boldsymbol{z}_{agg}), \eta(\boldsymbol{z}_*)|\mathcal{D}(t)\} d\eta(\boldsymbol{z}_{agg}),$$
$$P\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} = P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} \cdot P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t)\}.$$

However, $P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$ and $P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t)\}$ are unknown but can be estimated using the results from distributed GPR and agent-based GPR respectively. In particular, the results from distributed GPR are used to estimate

$P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$ since the estimate has lower variance (uncertainty). The results from agent-based GPR are used to estimate $P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t)\}$ because distributed GPR is limited to $\mathcal{Z}_{agg}$ and $\boldsymbol{z}_*$ may not be in $\mathcal{Z}_{agg}$. The product of $P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t)\}$ and $P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$, which yields $P\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$, then contains information from the local agent and those from the other agents in the network. The overall process can be interpreted as a fusion of global information with local information, where improvement is expected because more information is provided. After integrating over $\eta(\boldsymbol{z}_{agg})$, we obtain the estimate of $P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$. The detailed procedure is broken down into the following steps.

Step 1: Estimation of $P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$. Consider any $\boldsymbol{z}_{agg} \in \mathcal{Z}_{agg}$. Agent i's estimate of $P\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$, denoted by $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} \triangleq \mathcal{N}(\hat{\mu}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)}, (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2)$, is given by distributed GPR.

Step 2: Estimation of $P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}$. Note that agent-based GPR does not return covariance $\mathrm{cov}(\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}^{[i]}(t))$ that reflects the correlation between $\eta(\boldsymbol{z}_{agg})$ and $\eta(\boldsymbol{z}_*)$ similar to $k(\boldsymbol{z}_*, \boldsymbol{z}_{agg})$. We set $\mathrm{cov}(\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}^{[i]}(t)) = g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}_{agg})$ and define

$$g(\boldsymbol{z}_*, t) \triangleq \frac{\min\{\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}, \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}\} \cdot \max\{0, c - \psi^{[i]}\}}{k(\boldsymbol{z}_*, \boldsymbol{z}_*)},$$

where $c \triangleq \mu_\chi^{-1}\big(\frac{1}{n}\sum_{j=1}^n \chi^{[j]}\psi^{[j]}\big)$, $\psi^{[i]} \triangleq \frac{\sigma_f^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$, $\chi^{[i]} \triangleq \frac{1}{(\sigma_e^{[i]})^2} + \frac{1}{\sigma_f^2}$ and $\mu_\chi \triangleq \frac{1}{n}\sum_{i=1}^n \chi^{[i]}$. A distributed method for agent i to obtain $(\sigma_e^{[j]})^2$, $j \neq i$, is given in Section 3.3.4. This ensures covariance matrix

$$\tilde{\Sigma}_{\boldsymbol{z}_*, \boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} \triangleq \begin{bmatrix} \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} & g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}_{agg}) \\ g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}_{agg}) & \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} \end{bmatrix}$$

is positive definite. We further verify this choice is valid by showing $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 > 0$ for all $t \geqslant 1$ and $\boldsymbol{z}_* \in \mathcal{Z}_*$ in Section 3.4.2.3. We can write

$$\tilde{P}\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}^{[i]}(t)\} \triangleq \mathcal{N}\Big(\begin{bmatrix} \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \\ \check{\mu}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} \end{bmatrix}, \tilde{\Sigma}_{\boldsymbol{z}_*, \boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}\Big).$$

Then agent i's estimate of $P\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}$, denoted by $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t)\}$,

is given by $\tilde{P}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}^{[i]}(t)\}$ applying identities of joint Gaussian distribution (page 200, [54]) on $\tilde{P}\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}^{[i]}(t)\}$.

Step 3: Estimation of $P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$. Combining the previous two steps, agent i estimates $P\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$ as

$$\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} = \tilde{P}^{[i]}\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} \cdot \tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}.$$

Applying the same trick of nearest-neighbor prediction as in agent-based GPR, we choose $\boldsymbol{z}_{agg} = \boldsymbol{z}^{[i]}_{agg*}(t) \in \text{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}_{agg}(t))$ for each $\boldsymbol{z}_* \in \mathcal{Z}_*$. Then we have agent i's estimate of $P\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$ given by

$$\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\} = \int \tilde{P}^{[i]}\{\eta(\boldsymbol{z}^{[i]}_{agg*}(t)), \eta(\boldsymbol{z}_*)|\mathcal{D}(t)\} d\eta(\boldsymbol{z}^{[i]}_{agg*}(t)),$$

which has mean and variance in Line 11-12 of fused GPR (see Section 3.4.1 for derivation).

3.3.4 Choice of the kernel

In this chapter, we assume the following properties of the kernel k used in LiDGPR algorithm.

Assumption 3.3.1. 1. (Decomposition). The kernel function $k(\cdot, \cdot)$ can be decomposed in such a way that $k(\cdot, \cdot) = \kappa(\rho(\cdot, \cdot))$, where $\kappa : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{>0}$ is continuous.

2. (Boundedness). It holds that $0 < \kappa(r) \leqslant \sigma_f^2$ for all $r \geqslant 0$ and some $\sigma_f > 0$.

3. (Monotonicity). It holds that $\kappa(r)$ is monotonically decreasing as r increases and $\kappa(0) = \sigma_f^2$. ■

Remark 3.3.2. In GPR, kernel can be interpreted as the prior correlation between function evaluations. For a continuous function, it is reasonable to assume bounded correlation and the correlation is negatively related to the distance between two inputs. One example that satisfies Assumption 3.3.1 above is the class of squared exponential kernels having the form $k(\boldsymbol{z}, \boldsymbol{z}') = \sigma_f^2 \exp(-||\boldsymbol{z} - \boldsymbol{z}'||^2/\ell^2)$ (page 83, [54]). ■

To obtain the theoretic guarantees in Section 3.3.5, σ_f^2 is chosen for initialization as follows. Let $\sigma_\chi^2 \triangleq \sum_{i=1}^n (\chi^{[i]} + \mu_\chi)^2$, $\mathcal{V}^+ \triangleq \{i \in \mathcal{V} | c\sigma_f^2 - \psi^{[i]} > 0\}$, $\epsilon_+ \triangleq \min_{i\in\mathcal{V}^+}\{c\sigma_f^2 - \psi^{[i]}\}$. We choose $\sigma_f^2 \geqslant 1$ satisfying

$$\sigma_\chi^2/(\mu_\chi^2\epsilon_+) \leqslant (\sigma_e^{\min})^2/(\sigma_e^{\max})^2. \tag{3.4}$$

When σ_f^2 increases, $\chi^{[i]}$, μ_χ and σ_χ^2 converge to positive constants, ϵ_+ has growth rate $\mathcal{O}(\sigma_f^2)$, which gives the left hand side of (3.4) diminishing at $\mathcal{O}(\frac{1}{\sigma_f^2})$. Hence inequality (3.4) is satisfied when σ_f^2 is sufficiently large.

A distributed way to choose a single σ_f^2 is as follows. By using the Floodset algorithm (page 103, [79]), each agent i sends $(\sigma_e^{[i]})^2$ to its neighbors. By Assumption 3.2.1, within $n(b-1)$ iterations, each agent obtains a copy of $(\sigma_e^{[i]})^2$ from all $i \in \mathcal{V}$. Then all the values in (3.4) can be calculated. To further consider data fitting, each agent can incorporate (3.4) with existing hyperparameter optimization methods, such as [80] which uses a given amount of data points collected during initialization, or [81] which recursively updates whenever new data arrive. The resulting local hyperparameter of agent i is denoted as $\sigma_f^{[i]}$, then all the agents employ maximum consensus [82] to compute $\sigma_f = \max_{i\in\mathcal{V}}\{\sigma_f^{[i]}\}$, which terminates in $n(b-1)$ iterations.

3.3.5 Performance guarantee

In this section, we present the performance of predictive mean and variance returned by LiDGPR. The main results are summarized in Theorem 3.3.3 and Theorem 3.3.8, and their proofs are presented in Section 3.4.2 and Section 3.4.3.

Part of the performance is quantified in terms of the dispersion of local data defined as $d^{[i]}(t) \triangleq \sup_{\boldsymbol{z}\in\boldsymbol{\mathcal{Z}}} \rho(\boldsymbol{z}, \mathcal{Z}^{[i]}(t))$. We can interpret dispersion as a measurement of how dense the sampled data is distributed within a compact space. For notational simplicity, we introduce shorthand $\rho_{\boldsymbol{z}}^{\mathcal{Z}} \triangleq \rho(\boldsymbol{z}, \mathcal{Z})$.

Theorem 3.3.3 shows that LiDGPR makes predictions with lower uncertainty than agent-based GPR.

Theorem 3.3.3. *(Uncertainty reduction).* **Part I**: Suppose Assumption 3.3.1 holds. For all $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$ and $i \in \mathcal{V}$, the predictive variance by agent-based GPR is

bounded as

$$\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2} \leqslant \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \leqslant \sigma_f^2 - \frac{\kappa(d^{[i]}(t))^2}{\sigma_f^2+(\sigma_e^{[i]})^2}.$$

Part II: Suppose Assumptions 3.2.1, 3.2.2, 3.2.3 and 3.3.1 hold. For all $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$ and $i \in \mathcal{V}$, there exists a non-negative sequence $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ such that the predictive variance by LiDGPR is

$$0 < (\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t).$$

In particular, if $\mathcal{Z}^{[i]}_{agg}(t) = \emptyset$, $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) = 0$; otherwise:

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) \geqslant \mathcal{O}\Big(\kappa(\rho^{\boldsymbol{z}^{[i]}_{agg*}(t)}_{\boldsymbol{z}_*})^2\big(\frac{1}{n}\sum_{j=1}^{n}\kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2 - \frac{\sigma_f^2+(\sigma_e^{\max})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2\big)\Big). \qquad \blacksquare$$

We provide the steady-state results assuming that the dispersion is diminishing. Lemma 6 in [83] shows that dispersion does go to zero under uniform sampling.

Corollary 3.3.4. If $\lim_{t\to\infty} d^{[j]}(t) = 0$ for all $j \in \mathcal{V}$ and all the conditions in Theorem 3.3.3 are satisfied, then

$$\liminf_{t\to\infty} \gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) \geqslant \mathcal{O}\Big(\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \frac{1}{n}\sum_{j=1}^{n}\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t)}\Big),$$

where $\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t)} = \frac{\sigma_f^2(\sigma_e^{[j]})^2}{\sigma_f^2+(\sigma_e^{[j]})^2}$. $\blacksquare$

To ensure the improvement on prediction accuracy, we need to assume that the prior covariance function of η is correctly specified. Note that any non-zero mean Gaussian process can be decomposed into a deterministic process plus a zero-mean stochastic process such that GPR can be performed over the zero-mean stochastic process (page 27, [54]). Therefore, without loss of generality, we assume η follows a zero-mean Gaussian process for notational simplicity.

Assumption 3.3.5. It satisfies that $\eta \sim \mathcal{GP}(0, k)$. $\blacksquare$

That is, the target function η is completely specified by a zero-mean Gaussian process with kernel k. This assumption is common in the analysis of GPR (Theorem 1, [84]).

Furthermore, we need to assume that the state transition matrix induced by $A(t)$ is constant.

Assumption 3.3.6. It holds that $\prod_{\tau=1}^{t} A(\tau) = \prod_{\tau=1}^{t'} A(\tau)$ for any $t, t' > 1$. ■

One example that satisfies this assumption is each entry of $A(t)$ being constant $\frac{1}{n}$, which is a complete graph.

Furthermore, we assume η is Lipschitz continuous.

Assumption 3.3.7. There exists some positive constant $\ell_\eta \in \mathbb{R}$ such that $\sup_{\boldsymbol{z},\boldsymbol{z}' \in \boldsymbol{\mathcal{Z}}} |\eta(\boldsymbol{z}) - \eta(\boldsymbol{z}')| \leqslant \ell_\eta \rho(\boldsymbol{z}, \boldsymbol{z}')$. ■

Theorem 3.3.8 below compares the predictive errors of agent-based GPR with those of LiDGPR.

Theorem 3.3.8. *(Accuracy improvement).* **Part I**: Suppose Assumptions 3.3.1, 3.3.5 and 3.3.7 hold. For all $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$ and $i \in \mathcal{V}$, with probability at least $1 - \frac{(\sigma_e^{\max})^2}{\epsilon^2}$, $\epsilon > \sigma_e^{\max}$, the predictive error resulted from agent-based GPR is bounded as

$$|\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*)| \leqslant (1 - \frac{\kappa(d^{[i]}(t))}{\sigma_f^2 + (\sigma_e^{[i]})^2})\|\eta\|_{\boldsymbol{\mathcal{Z}}} + \ell_\eta d^{[i]}(t) + \epsilon.$$

Part II: Suppose $\lim_{t\to\infty} d^{[i]}(t) = 0$, $\forall i \in \mathcal{V}$, and Assumptions 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.5, 3.3.6 and 3.3.7 hold. For all $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$ and $i \in \mathcal{V}$: if $\lim_{t\to\infty} \mathcal{Z}_{agg}^{[i]}(t) \neq \emptyset$, then

$$\lim_{t\to\infty} \mathbb{E}[(\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} - \eta(\boldsymbol{z}_*))^2 - (\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))^2] \leqslant -\mathcal{O}\Big(\kappa(\rho^{\mathcal{Z}_{agg}}_{\boldsymbol{z}_*})\Big) < 0;$$

otherwise,

$$\lim_{t\to\infty} (\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} - \eta(\boldsymbol{z}_*))^2 = \lim_{t\to\infty} (\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))^2.$$

Further, if $(\sigma_e^{[i]})^2 > \dfrac{1}{n}\displaystyle\sum_{j=1}^{n} (\sigma_e^{[j]})^2$, $\lim_{t\to\infty} \mathcal{Z}_{agg}^{[i]}(t) = \mathcal{Z}_{agg}$. ■

The two theorems indicate that LiDGPR leverages inter-agent communication to improve transient and steady-state learning performance; meanwhile, no agent suffers from degraded learning performance. This improvement of learning performance is achieved by the fact that the agents whose data samples have higher

dispersion (or observation noise has higher variance) benefit from those with data samples having lower dispersion (or observation noise having lower variance) via communication. Next we elaborate on the fact.

Transient improvement. Term $\frac{1}{n}\sum_{j=1}^{n}\kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2 - \frac{\sigma_f^2+(\sigma_e^{\max})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2$ in the lower bound of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ in Theorem 3.3.3 indicates that agent i benefits in variance prediction when $\frac{\sigma_f^2+(\sigma_e^{\max})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2$ is below $\frac{1}{n}\sum_{j=1}^{n}\kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2$. Note that $\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[j]}_{agg*}(t)}$ is closely related to dispersion $d^{[j]}(t)$ and recall the monotonicity property of κ. Hence, LiDGPR enables agents whose data samples have higher dispersion (data sparsely sampled) and observation noise has higher variance to benefit from those with data samples having lower dispersion (data densely sampled) and observation noise having lower variance.

Steady-state improvement. If $\liminf_{t\to\infty}\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) > 0$, it indicates that agent i obtains improvement in steady-state learning performance in predictive variance. From Corollary 3.3.4, we can see that $\liminf_{t\to\infty}\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) > 0$, if its steady-state local predictive variance, $\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$, is above the average over the agents in $\mathcal{V}$. By Corollary 3.3.4, $\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ is positively related to $(\sigma_e^{[i]})^2$. Hence agents with observation noise of higher variance might obtain steady-state improvement in predictive variance from those with lower variance.

Steady-state improvement in prediction accuracy is reflected by the case $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ in Theorem 3.3.8. The sufficient condition $(\sigma_e^{[i]})^2 > \frac{1}{n}\sum_{j=1}^{n}(\sigma_e^{[j]})^2$ indicates that agent i obtains steady-state improvement when $(\sigma_e^{[i]})^2$ is above the average. That is, agents with observation noise of higher variance benefits from those with smaller variance.

The improvements $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ and $\lim_{t\to\infty}\mathbb{E}[(\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)}-\eta(\boldsymbol{z}_*))^2-(\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}-\eta(\boldsymbol{z}_*))^2]$ are positively related to $\kappa(\rho^{\boldsymbol{z}^{[i]}_{agg*}(t)}_{\boldsymbol{z}_*})^2$ and $\kappa(\rho^{\mathcal{Z}_{agg}}_{\boldsymbol{z}_*})$ respectively. By monotonicity of κ in Assumption 3.3.1, these terms indicate that the benefit brought by communication decays as $\boldsymbol{z}_*$ is moving away from $\boldsymbol{z}^{[i]}_{agg*}(t)$ and $\mathcal{Z}_{agg}$ respectively. That is, a denser set $\mathcal{Z}_{agg}$ could induce larger improvements.

3.3.6 Discussion

Relevance: The two theorems indicate that both prediction uncertainties and prediction errors reduce as local dispersion $d^{[i]}(t)$ reduces. This provides insights on data sampling such that the agents should sample in a way that minimizes $d^{[i]}(t)$. The terms $\kappa(\rho_{\boldsymbol{z}_*}^{\boldsymbol{z}_{agg*}^{[i]}(t)})^2$ and $\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}_{agg}})$ in Theorem 3.3.3 and Theorem 3.3.8 show that the improvement of learning performances obtained from communication decreases as the test point $\boldsymbol{z}_*$ is moving away from $\mathcal{Z}_{agg}$. This can guide the design process of $\mathcal{Z}_{agg}$ such that if the test points in $\mathcal{Z}_*$ are known a priori, $\mathcal{Z}_{agg}$ should be allocated such that $\sup_{\boldsymbol{z}_*\in\mathcal{Z}_*}\min_{\boldsymbol{z}_{agg}\in\mathcal{Z}_{agg}}\rho(\boldsymbol{z}_*,\boldsymbol{z}_{agg})$ is minimized; otherwise $\mathcal{Z}_{agg}$ should be designed such that $\sup_{\boldsymbol{z}_*\in\boldsymbol{\mathcal{Z}}}\min_{\boldsymbol{z}_{agg}\in\mathcal{Z}_{agg}}\rho(\boldsymbol{z}_*,\boldsymbol{z}_{agg})$ is minimized.

Complexities related to $\mathcal{Z}_*$ and $\mathcal{Z}_{agg}$. The communication overhead scales as $\mathcal{O}(|\mathcal{Z}_{agg}||\mathcal{N}^{[i]}(t)|)$. Due to the use of Nearest-neighbor GPR, agent-based GPR only requires $\mathcal{O}(t)$ in memory. The memory requirements for both distributed GPR and fused GPR are $\mathcal{O}(|\mathcal{Z}_{agg}|)$. The computational complexities scale as $\mathcal{O}(t|\mathcal{Z}_*|)$ for agent-based GPR, $\mathcal{O}(|\mathcal{Z}_{agg}|)$ for distributed GPR, and $\mathcal{O}(|\mathcal{Z}_*||\mathcal{Z}_{agg}|)$ for fused GPR.

Nearest-neighbor GPR vs. full GPR. Part I of Theorem 3.3.3 and Theorem 3.3.8 characterize the steady-state errors of agent-based GPR. Paper [55] shows that $\sigma^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\to(\sigma_e^{[i]})^2$ and $\mu_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\to\eta(\boldsymbol{z}_*)$ almost surely as $t\to\infty$ for full GPR. Part I of Theorem 3.3.3 indicates $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\to\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$, hence the variance for noisy prediction (page 19 [54]) equals $\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}+(\sigma_e^{[i]})^2$, and Theorem 3.3.8 indicates

$$\limsup_{t\to\infty}|\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}-\eta(\boldsymbol{z}_*)|\leqslant\frac{(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}\|\eta\|_{\boldsymbol{\mathcal{Z}}}+\epsilon,$$

assuming $d^{[i]}(t)\to 0$. The discrepancy can be caused by the fact that full GPR in [55] makes prediction using all the data in the dataset while Nearest-neighbor GPR only uses the data of the nearest input. Full GPR has computational complexity $\mathcal{O}(t^3)$ while Nearest-neighbor GPR has the same computational complexity as nearest neighbor search, which is $\mathcal{O}(t)$ for the worst case [85]. This is the trade-off between learning accuracy and computational complexity. Note that both full GPR and Nearest-neighbor GPR have the same steady-state errors under noise-free condition, i.e., $(\sigma_e^{[i]})^2=0$.

Symbol	Meaning/definition	Equivalence
$\check{\mu}_{\mathbf{z}_*\|\mathcal{D}^{[i]}(t)}$	Predictive mean from agent-based GPR of agent i	$\check{\mu}_{\boldsymbol{z}_*\|\mathcal{D}^{[i]}(t)} = \check{r}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t)$
$\hat{\mu}^{[i]}_{\mathbf{z}_*\|\mathcal{D}(t)}$	Predictive mean from distributed GPR of agent i	$\hat{\mu}^{[i]}_{\boldsymbol{z}_*\|\mathcal{D}(t)} = \hat{r}^{[i]}_{\boldsymbol{z}_*}(t) + \hat{e}^{[i]}_{\boldsymbol{z}_*}(t)$
$\tilde{\mu}^{[i]}_{\mathbf{z}_*\|\mathcal{D}(t)}$	Predictive mean from fused GPR of agent i	
$\check{\sigma}^2_{\boldsymbol{z}_*\|\mathcal{D}^{[i]}(t)}$	Predictive variance from agent-based GPR of agent i	$\check{\sigma}^2_{\boldsymbol{z}_*\|\mathcal{D}^{[i]}(t)} = \sigma_f^2 - \frac{\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$
$(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*\|\mathcal{D}(t)})^2$	Predictive variance from distributed GPR of agent i	$(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*\|\mathcal{D}(t)})^2 = (\xi^{[i]}_{\boldsymbol{z}_*}(t))^{-1}$
$(\tilde{\sigma}^{[i]}_{\mathbf{z}_*\|\mathcal{D}(t)})^2$	Predictive variance from fused GPR of agent i	
$r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$	Reference signal for consensus state $\boldsymbol{\theta}$ in distributed GPR	$r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t) = \hat{r}^{[i]}_{\boldsymbol{z}_*}(t) + \hat{e}^{[i]}_{\boldsymbol{z}_*}(t)$
$r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)$	Reference signal for consensus state $\boldsymbol{\xi}$ in distributed GPR	
$r^{[i]}_{\boldsymbol{\lambda},\boldsymbol{z}_*}(t)$	Reference signal for consensus state $\boldsymbol{\lambda}$ in distributed GPR	
$\check{r}^{[i]}_{\boldsymbol{z}_*}(t)$	Real-valued component of $\check{\mu}_{\boldsymbol{z}_*\|\mathcal{D}^{[i]}(t)}$	$\check{r}^{[i]}_{\boldsymbol{z}_*}(t) = \frac{\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*})\eta(\boldsymbol{z}^{[i]}_*(t))}{\sigma_f^2+(\sigma_e^{[i]})^2}$
$\check{r}^{[i]}_{\boldsymbol{z}_*}$	$\check{r}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty} \check{r}^{[i]}_{\boldsymbol{z}_*}(t)$	$\check{r}^{[i]}_{\boldsymbol{z}_*} = \frac{\sigma_f^2\eta(\boldsymbol{z}_*)}{\sigma_f^2+(\sigma_e^{[i]})^2} = \psi^{[i]}\eta(\boldsymbol{z}_*)$

$\hat{r}^{[i]}_{\boldsymbol{z}_*}(t)$	Real-valued component of $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$	$\hat{r}^{[i]}_{\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*\mid\mathcal{D}^{[i]}(t)}\check{r}^{[i]}_{\boldsymbol{z}_*}(t)$
$\hat{r}^{[i]}_{\boldsymbol{z}_*}$	$\hat{r}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty}\hat{r}^{[i]}_{\boldsymbol{z}_*}(t)$	$\hat{r}^{[i]}_{\boldsymbol{z}_*} = (\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2})^{-1}\check{r}^{[i]}_{\boldsymbol{z}_*}$
$\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t)$	Real-valued component of $\hat{\mu}^{[i]}_{\boldsymbol{z}_*\mid\mathcal{D}(t)}$	$\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t) = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*\mid\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t)$
$e^{[i]}_{\boldsymbol{z}_*}$	Observation error at $\boldsymbol{z}_*$	$e^{[i]}_{\boldsymbol{z}_*} = y^{[i]}_{\boldsymbol{z}_*} - \eta(\boldsymbol{z}_*)$
$\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$	Stochastic component of $\check{\mu}_{\boldsymbol{z}_*\mid\mathcal{D}^{[i]}(t)}$	$\check{e}^{[i]}_{\boldsymbol{z}_*}(t) = \frac{\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*})}{\sigma_f^2+(\sigma_e^{[i]})^2}\big(y^{[i]}_{\boldsymbol{z}^{[i]}_*(t)} - \eta(\boldsymbol{z}^{[i]}_*(t))\big)$
$\check{e}^{[i]}_{\boldsymbol{z}_*}$	$\check{e}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty}\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$	$\check{e}^{[i]}_{\boldsymbol{z}_*} = \frac{\sigma_f^2}{\sigma_f^2+(\sigma_e^{[i]})^2}e^{[i]}_{\boldsymbol{z}_*} = \psi^{[i]}e^{[i]}_{\boldsymbol{z}_*}$
$\hat{e}^{[i]}_{\boldsymbol{z}_*}(t)$	Stochastic component of $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$	$\hat{e}^{[i]}_{\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*\mid\mathcal{D}^{[i]}(t)}\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$
$\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t)$	Stochastic component of $\hat{\mu}^{[i]}_{\boldsymbol{z}_*\mid\mathcal{D}(t)}$	$\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t) = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*\mid\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$
$\chi^{[i]}$	$\chi^{[i]} \triangleq \frac{1}{(\sigma_e^{[i]})^2} + \frac{1}{\sigma_f^2}$	$\chi^{[i]} = \lim_{t\to\infty}\check{\sigma}^{-2}_{\boldsymbol{z}_*\mid\mathcal{D}^{[i]}(t)}$
$\psi^{[i]}$	$\psi^{[i]} \triangleq \frac{\sigma_f^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$	
μ_χ	$\mu_\chi \triangleq \frac{1}{n}\sum_{i=1}^n \chi^{[i]}$	$\mu_\chi^{-1} = \lim_{t\to\infty}(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*\mid\mathcal{D}(t)})^2 = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*\mid\mathcal{D}(t)})^2$
c	$c \triangleq \mu_\chi^{-1}\big(\frac{1}{n}\sum_{j=1}^n \chi^{[j]}\psi^{[j]}\big)$	
σ_χ^2	$\sigma_\chi^2 \triangleq \sum_{i=1}^n(\chi^{[i]}+\mu_\chi)^2$	
ϵ_+	$\epsilon_+ \triangleq \min_{i\in\mathcal{V}^+}\{c\sigma_f^2 - \psi^{[i]}\}$	

Table 3.1: Table of symbols

3.4 Proofs

In this section, we present the derivation of Line 11-12 in fused GPR and the proofs of Theorem 3.3.3 and Theorem 3.3.8. Table 3.1 shows the symbols that are used in multiple important results and the relation among them.

3.4.1 Derivation of Line 11-12 in fused GPR

Recall that $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}$ is given by applying identities of joint Gaussian distribution (page 200, [54]) to

$$\tilde{P}\{\begin{bmatrix}\eta(\boldsymbol{z}_*)\\ \eta(\boldsymbol{z}_{agg})\end{bmatrix}|\mathcal{D}^{[i]}(t)\} = \mathcal{N}(\begin{bmatrix}\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\\ \check{\mu}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}\end{bmatrix}, \tilde{\Sigma}_{\boldsymbol{z}_*,\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}).$$

This gives $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}$ as a Gaussian distribution with mean and variance

$$\begin{aligned}
&\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\boldsymbol{z}_{agg},\mathcal{D}(t)} = \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} + g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}_{agg})\check{\sigma}^{-2}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}(\eta(\boldsymbol{z}_{agg}) - \check{\mu}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}),\\
&(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\boldsymbol{z}_{agg},\mathcal{D}(t)})^2 = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - g(\boldsymbol{z}_*, t)k(\boldsymbol{z}_*, \boldsymbol{z}_{agg})\check{\sigma}^{-2}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}k(\boldsymbol{z}_*, \boldsymbol{z}_{agg})g(\boldsymbol{z}_*, t).
\end{aligned}$$

Notice that the mean and variance of $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\}$ is given in distributed GPR. Then we have the product

$$\begin{aligned}
\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} &= \tilde{P}^{[i]}\{\eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} \cdot \tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\eta(\boldsymbol{z}_{agg}), \mathcal{D}(t))\}\\
&= \mathcal{N}(\hat{\mu}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)}, (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2) \cdot \mathcal{N}(\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\boldsymbol{z}_{agg},\mathcal{D}(t)}, (\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\boldsymbol{z}_{agg},\mathcal{D}(t)})^2).
\end{aligned}$$

After some basic algebraic manipulations (finding the corresponding terms in (A.6) [54]) or directly plugging the terms in equation (9) of [81], we have

$$\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*), \eta(\boldsymbol{z}_{agg})|\mathcal{D}(t)\} = \mathcal{N}(\begin{bmatrix}\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)}\\ \tilde{\mu}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)}\end{bmatrix}, \begin{bmatrix}(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 & \tilde{\sigma}^{[i]}_{\boldsymbol{z}_*,\boldsymbol{z}_{agg}|\mathcal{D}(t)}\\ \tilde{\sigma}^{[i]}_{\boldsymbol{z}_{agg},\boldsymbol{z}_*|\mathcal{D}(t)} & (\tilde{\sigma}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2\end{bmatrix}).$$

Replacing $\boldsymbol{z}_{agg}$ with $\boldsymbol{z}^{[i]}_{agg*}(t)$, $\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)}$ and $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ have the forms in Line 11-12 in fused GPR. Hence we have the marginal distribution $\tilde{P}^{[i]}\{\eta(\boldsymbol{z}_*)|\mathcal{D}(t)\}$.

3.4.2 Proof of Theorem 3.3.3

In this section, we first derive the lower bound and the upper bound of the predictive variance of agent-based GPR and prove Part I of Theorem 3.3.3 in Section 3.4.2.1. Then we derive the bounds of distributed GPR in Proposition 3.4.4 in Section 3.4.2.2. Lastly, we derive the bounds of fused GPR and prove Part II of

Theorem 3.3.3 in Section 3.4.2.3.

First of all, we introduce some properties of functions $f_1 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ as $f_1(x) = \frac{1}{x}$ and $f_2 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ as $f_2(x) = \sigma_f^2 - \frac{\sigma_f^4}{\sigma_f^2+x}$. These will be used in later analysis.

Lemma 3.4.1. It holds that

$$f_1(\frac{1}{n}\sum_{i=1}^{n} x_i) \leqslant \frac{1}{n}\sum_{i=1}^{n} f_1(x_i),$$
$$f_2(\frac{1}{n}\sum_{i=1}^{n} x_i) \geqslant \frac{1}{n}\sum_{i=1}^{n} f_2(x_i),$$
$$\big(\frac{1}{n}\sum_{i=1}^{n} f_2(x_i)\big)^{-1} \leqslant \frac{1}{n}\sum_{i=1}^{n} \big(f_2(x_i)\big)^{-1}.$$

Proof: It is obvious that f_1 is convex. Then Jensen's inequality (page 77, [86]) gives $f_1(\frac{1}{n}\sum_{i=1}^{n} x_i) \leqslant \frac{1}{n}\sum_{i=1}^{n} f_1(x_i)$.

It is obvious that f_2 is concave. By Jensen's inequality and concavity, we have $f_2(\frac{1}{n}\sum_{i=1}^{n} x_i) \geqslant \frac{1}{n}\sum_{i=1}^{n} f_2(x_i)$.

Applying Jensen's inequality utilizing the monotonicity and the convexity of inverse function $f_3(x) = \frac{1}{x}$ for $x > 0$, we can also obtain $\big(\frac{1}{n}\sum_{i=1}^{n} f_2(x_i)\big)^{-1} \leqslant \frac{1}{n}\sum_{i=1}^{n} \big(f_2(x_i)\big)^{-1}$. ■

3.4.2.1 Variance analysis of agent-based GPR

In this section, we present the proof of Theorem 3.3.3 Part I.

Proof of Theorem 3.3.3 Part I: Pick any $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$. By monotonicity of κ in Assumption 3.3.1, Line 5 in agent-based GPR gives the predictive variance $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \sigma_f^2 - \frac{k(\boldsymbol{z}_*,\boldsymbol{z}_*^{[i]}(t))^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$. Note that the definition of $\boldsymbol{z}_*^{[i]}(t)$ renders $\rho^{\boldsymbol{z}_*^{[i]}(t)}_{\boldsymbol{z}_*} = \rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*}$. Combining this with the decomposition property of κ in Assumption 3.3.1 gives

$$\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \sigma_f^2 - \frac{\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2}. \tag{3.5}$$

The definition of local dispersion $d^{[i]}(t)$ renders $d^{[i]}(t) \geqslant \rho(\boldsymbol{z}_*, \mathcal{Z}^{[i]}(t))$. Combin-

ing this with the monotonicity of κ in Assumption 3.3.1 gives $\kappa(d^{[i]}(t)) \leqslant \kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})$, which renders

$$\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \leqslant \sigma_f^2 - \frac{\kappa(d^{[i]}(t))^2}{\sigma_f^2 + (\sigma_e^{[i]})^2},\ \forall \boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}.$$

Applying the boundedness of κ in Assumption 3.3.1 to (3.5), we have $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \geqslant \frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$. ■

As $\lim_{t\to\infty} d^{[j]}(t) = 0$, $\forall j \in \mathcal{V}$, the upper bound of $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ converges to its lower bound.

Corollary 3.4.2. Suppose Assumption 3.3.1 holds. If $\lim_{t\to\infty} d^{[j]}(t) = 0$ for all $j \in \mathcal{V}$, it holds that $\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$. ■

3.4.2.2 Variance analysis of distributed GPR

First, we define the following notations. We let operators Δ, sup, max and min be applied element-wise across the vectors:

$$\begin{aligned}
&\bar{\boldsymbol{m}}(t) \triangleq \max_{i\in\mathcal{V}} \boldsymbol{\xi}^{[i]}(t), \quad \underline{\boldsymbol{m}}(t) \triangleq \min_{i\in\mathcal{V}} \boldsymbol{\xi}^{[i]}(t),\\
&\boldsymbol{\delta}_{\boldsymbol{m}}(t) \triangleq \bar{\boldsymbol{m}}(t) - \underline{\boldsymbol{m}}(t), \quad \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t) \triangleq [r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)]_{\boldsymbol{z}_*\in\mathcal{Z}_{agg}},\\
&\Delta\boldsymbol{r}_{\max}(t) \triangleq \max_{i\in\mathcal{V}} \Delta\boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t), \quad \Delta\boldsymbol{r}_{\min}(t) \triangleq \min_{i\in\mathcal{V}} \Delta\boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t),\\
&\bar{\Delta}\boldsymbol{r}_{\max}(t) \triangleq \max_{i\in\mathcal{V}}\{\sup_{s\geqslant 1} \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(s) - \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t-1)\},\\
&\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(t) \triangleq \Delta\boldsymbol{r}_{\max}(t) - \Delta\boldsymbol{r}_{\min}(t), \quad \zeta \triangleq \alpha^{\frac{1}{2}n(n+1)b-1}.
\end{aligned}$$

First of all, we introduce several properties in Lemma 3.4.3.

Lemma 3.4.3. Suppose Assumptions 3.2.2 and 3.3.1 and $\lim_{t\to\infty} d^{[j]}(t) = 0$ for all $j \in \mathcal{V}$ hold. For each $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$:

Claim 3.4.3.1. It holds that $\frac{1}{\sigma_f^2} \leqslant r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \leqslant (\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2+(\sigma_e^{\min})^2})^{-1}$, $\forall i \in \mathcal{V}$, $t \geqslant 0$.

Claim 3.4.3.2. It holds that $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \geqslant r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1)$, $\forall t \geqslant 0$.

Claim 3.4.3.3. It holds that $\frac{1}{\sigma_f^2} \leqslant \xi^{[i]}_{\boldsymbol{z}_*}(t) \leqslant n(\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2+(\sigma_e^{\min})^2})^{-1}$, $\forall i \in \mathcal{V}$, $t \geqslant 1$.

Claim 3.4.3.4. It holds that $\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(t) \preceq \bar{\Delta}\boldsymbol{r}_{\max}(t)$, $\forall t \geqslant 0$.

Claim 3.4.3.5. It holds that $\bar{\Delta}\boldsymbol{r}_{\max}(t) \preceq \bar{\Delta}\boldsymbol{r}_{\max}(t-1)$, $\forall t \geqslant 1$.

Claim 3.4.3.6. It holds that $\|\bar{\Delta}\boldsymbol{r}_{\max}(t)\|_\infty \leqslant \mathcal{O}(\sigma_f^4 - \kappa(d^{\max}(t-1))^2)$, $\forall t \geqslant 1$.

Proof: We prove the claims one-by-one:

Proof of Claim 3.4.3.1: Recall the boundedness property in Assumption 3.3.1 requires that $\kappa(\cdot) > 0$. Therefore it follows from Part I of Theorem 3.3.3 that

$$\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2} \leqslant \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \leqslant \sigma_f^2, \ \forall t \geqslant 1.$$

Combining this with the definition of $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)$ on Line 5 in distributed GPR, gives

$$\frac{1}{\sigma_f^2} \leqslant r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \leqslant (\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2})^{-1} \leqslant (\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2 + (\sigma_e^{\min})^2})^{-1}.$$

Combining with initial condition $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0) = \frac{1}{\sigma_f^2}$ gives the above inequalities hold for $t \geqslant 0$. □

Proof of Claim 3.4.3.2: Due to incremental sampling, the local collection of input data $\mathcal{Z}^{[i]}(t) = \mathcal{Z}^{[i]}(t-1) \cup \boldsymbol{z}^{[i]}(t)$ monotonically expands, hence $\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}$ decreases. By monotonicity of κ in Assumption 3.3.1, equation (3.5) indicates $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ decreases as $\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}$ decreases. This renders that $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ is non-decreasing, i.e., $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \geqslant r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1)$, for all $t \geqslant 1$. With initial conditions $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(-1) = 0$ and Claim 3.4.3.1, we have $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \geqslant r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1)$ for $t \geqslant 0$. □

Proof of Claim 3.4.3.3: Outline: We first show that

$$\frac{1}{\sigma_f^2} \leqslant \xi^{[i]}_{\boldsymbol{z}_*}(t) \leqslant \sum_{\tau=0}^{t-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau)$$

using induction, then we find an upper bound for $\sum_{\tau=0}^{t-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau)$.

First, we show the induction. For $t = 1$, by Line 6 in distributed GPR and initial condition $\boldsymbol{\xi}^{[i]}(0) = \boldsymbol{r}^{[i]}_{\boldsymbol{\xi}}(-1) = \mathbf{0}_{|\mathcal{Z}_{agg}|}$, we have

$$\xi^{[i]}_{\boldsymbol{z}_*}(1) = r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0) = \Delta r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0) \leqslant \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0).$$

Since we have initial condition $\boldsymbol{r}^{[i]}_{\boldsymbol{\xi}}(0) = \frac{1}{\sigma_f^2}\mathbf{1}_{|\mathcal{Z}_{agg}|}$, the claim holds for $t = 1$. Suppose it holds for $t = m$. Then for $t = m + 1$, according to distributed GPR Line

6, we have

$$\begin{aligned}\xi^{[i]}_{\boldsymbol{z}_*}(m+1) &= \sum_{j=1}^{n} a_{ij}(m)\xi^{[j]}_{\boldsymbol{z}_*}(m) + \Delta r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(m) \\ &\leqslant \sum_{\tau=0}^{m-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) + \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(m) = \sum_{\tau=0}^{m} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau),\end{aligned} \tag{3.6}$$

where the inequality follows from the row stochasticity in Assumption 3.2.2. This proves the upper bound of the induction.

By Claim 3.4.3.2, $\Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(m) \geqslant 0$ for all $m \geqslant 0$. Since $r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0) = \frac{1}{\sigma_f^2} \leqslant \xi^{[i]}_{\boldsymbol{z}_*}(m)$, following from (3.6) we have

$$\xi^{[i]}_{\boldsymbol{z}_*}(m+1) \geqslant \sum_{j=1}^{n} a_{ij}(t) r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(0) + \Delta r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(m) \geqslant \frac{1}{\sigma_f^2}.$$

The proof for the induction is completed.

Second, we find the upper bound of $\sum_{\tau=0}^{t-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau)$. Claim 3.4.3.2 implies $\Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) \geqslant 0$ for all $\tau \geqslant 0$ and all $j \in \mathcal{V}$. Hence

$$\sum_{\tau=0}^{t-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) \leqslant \sum_{\tau=0}^{t-1} \sum_{j=1}^{n} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau). \tag{3.7}$$

Given initial condition $r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(-1) = 0$ and recall the definition of operator Δ where $\Delta r(t) \triangleq r(t) - r(t-1)$, it follows that $\sum_{\tau=0}^{t-1} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) = r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1)$. Therefore

$$\sum_{\tau=0}^{t-1} \sum_{j=1}^{n} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) = \sum_{j=1}^{n} \sum_{\tau=0}^{t-1} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau) = \sum_{j=1}^{n} r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1).$$

Applying the upper bound in Claim 3.4.3.1, we have

$$\sum_{j=1}^{n} r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t-1) \leqslant n\left(\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2 + (\sigma_e^{\min})^2}\right)^{-1},$$

which, by (3.7), is also an upper bound for $\sum_{\tau=0}^{t-1} \max_{j\in\mathcal{V}} \Delta r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(\tau)$. □

Proof of Claim 3.4.3.4: By Claim 3.4.3.2, we have $\Delta \boldsymbol{r}_{\min,\boldsymbol{\xi}}(t) \succeq \mathbf{0}_{|\mathcal{Z}_{agg}|}$, and

hence

$$\begin{aligned}\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(t) &= \Delta\boldsymbol{r}_{\max,\boldsymbol{\xi}}(t) - \Delta\boldsymbol{r}_{\min,\boldsymbol{\xi}}(t) \preceq \Delta\boldsymbol{r}_{\max,\boldsymbol{\xi}}(t) = \max_{i\in\mathcal{V}}\{\boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t) - \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t-1)\} \\ &\leqslant \max_{i\in\mathcal{V}}\{\sup_{s\geqslant 1}\boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(s) - \boldsymbol{r}_{\boldsymbol{\xi}}^{[i]}(t-1)\} = \bar{\Delta}\boldsymbol{r}_{\max,\boldsymbol{\xi}}(t), \quad \forall t \geqslant 0. \end{aligned}$$ □

Proof of Claim 3.4.3.5: In Claim 3.4.3.2 $r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(t)$ being non-decreasing implies $0 \leqslant \bar{\Delta}\boldsymbol{r}_{\max}(t) \preceq \bar{\Delta}\boldsymbol{r}_{\max}(t-1)$. □

Proof of Claim 3.4.3.6: Notice that

$$\sup_{s\geqslant 1} r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(s) - r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(t-1) \leqslant \sup_{s\geqslant 1} r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(s) - \min_{\boldsymbol{z}\in\mathcal{Z}_{agg}} r_{\boldsymbol{\xi},\boldsymbol{z}}^{[i]}(t-1).$$

The definition of $r_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)$ on Line 5 in distributed GPR gives

$$\min_{\boldsymbol{z}\in\mathcal{Z}_{agg}} r_{\boldsymbol{\xi},\boldsymbol{z}}^{[i]}(t-1) = (\max_{\boldsymbol{z}\in\mathcal{Z}_{agg}} \check{\sigma}^2_{\boldsymbol{z}|\mathcal{D}^{[i]}(t-1)})^{-1}.$$

Applying the upper bound in Theorem 3.3.3 Part I renders

$$\min_{\boldsymbol{z}\in\mathcal{Z}_{agg}} r_{\boldsymbol{\xi},\boldsymbol{z}}^{[i]}(t-1) \geqslant (\sigma_f^2 - \frac{\kappa(d^{[i]}(t-1))^2}{\sigma_f^2 + (\sigma_e^{[i]})^2})^{-1}.$$

Combining the definition of $r_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)$ and Theorem 3.3.3 Part I renders

$$\sup_{s\geqslant 1} r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(s) \leqslant (\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2})^{-1}, \forall i \in \mathcal{V}.$$

Therefore, we have

$$\sup_{s\geqslant 1} r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(s) - r_{\boldsymbol{\xi},\boldsymbol{z}_*}^{[i]}(t-1) \leqslant (\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2})^{-1} - (\sigma_f^2 - \frac{\kappa(d^{[i]}(t-1))^2}{\sigma_f^2 + (\sigma_e^{[i]})^2})^{-1}.$$

Let $p^{[i]} \triangleq \frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$ and $q^{[i]}(t) \triangleq \sigma_f^2 - \frac{\kappa(d^{[i]}(t-1))^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$. Based on the monotonicity of κ in Assumption 3.3.1 and since $\lim_{t\to\infty} d^{[i]}(t) = 0$ for all $i \in \mathcal{V}$, we have $q^{[i]}(t) \searrow p^{[i]}$.

Since $q^{[i]}(t) > p^{[i]} > 0$ for all $t \geqslant 0$, we can apply manipulation

$$\frac{1}{p^{[i]}} - \frac{1}{q^{[i]}(t)} = \frac{q^{[i]}(t) - p^{[i]}}{p^{[i]} q^{[i]}(t)} \leqslant \frac{q^{[i]}(t) - p^{[i]}}{(p^{[i]})^2}.$$

Then we can further obtain

$$\sup_{s \geqslant 1} r^{[i]}_{\boldsymbol{\xi}, \boldsymbol{z}_*}(s) - r^{[i]}_{\boldsymbol{\xi}, \boldsymbol{z}_*}(t-1) \leqslant \beta_r^{[i]} (\sigma_f^4 - \kappa(d^{[i]}(t-1))^2),$$

where $\beta_r^{[i]} \triangleq \frac{\sigma_f^2 + (\sigma_e^{[i]})^2}{\sigma_f^4 (\sigma_e^{[i]})^4}$.

Plugging the above inequality back into the definition of $\bar{\Delta} \boldsymbol{r}_{\max}(t)$ and apply the monotonicity of κ in Assumption 3.3.1, it follows that

$$\begin{aligned} \|\bar{\Delta} \boldsymbol{r}_{\max}(t)\|_\infty &\leqslant \max_{i \in \mathcal{V}} \beta_r^{[i]} (\sigma_f^4 - \kappa(d^{[i]}(t-1))^2) \\ &\leqslant \beta_r^{\max} (\sigma_f^4 - \kappa(d^{\max}(t-1))^2). \qquad \square \end{aligned}$$

The proof of the lemma is completed. ■

We define the subsequence $\{t_j\}$ as follows: $t_{-1} \triangleq 1$,

$$t_0 \triangleq \operatorname{ceil}\Big(\big(\frac{\log(2(nb-1)\|\Delta \boldsymbol{r}_{\max}(1)\|_\infty)}{\log(1-\zeta)\zeta ||\boldsymbol{\delta_m}(1)||_\infty} + 1 \big) \big(nb - 1 \big) \Big),$$

where $\operatorname{ceil}(x) \triangleq \min\{x' \in \mathbb{Z} \mid x' \geqslant x\}$, and for all $j \geqslant 1$,

$$t_j \triangleq \operatorname{ceil}\Big(\big(\frac{\log(\|\Delta \boldsymbol{r}_{\max}(t_{j-1})\|_\infty)}{2\log(1-\zeta)\|\Delta \boldsymbol{r}_{\max}(t_{j-2})\|_\infty} + 1 \big) \big(nb - 1 \big) \Big).$$

The proposition below characterizes the convergence of predictive variance from distributed GPR.

Proposition 3.4.4. *(Convergence of distributed GPR).* Suppose Assumptions 3.2.1, 3.2.2, 3.2.3 and 3.3.1 hold. For all $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$, for all $i \in \mathcal{V}$ and $t \geqslant 1$, the convergence of $(\hat{\sigma}^{[i]}_{\boldsymbol{z}_* | \mathcal{D}(t)})^2$ in distributed GPR to $(\check{\sigma}^{(agg)}_{\boldsymbol{z}_* | \mathcal{D}(t)})^2$ is characterized by: For $t < t_0$:

$$|(\hat{\sigma}^{[i]}_{\boldsymbol{z}_* | \mathcal{D}(t)})^2 - (\check{\sigma}^{(agg)}_{\boldsymbol{z}_* | \mathcal{D}(t)})^2| \leqslant 2\sigma_f^4 (1-\zeta)^{\frac{t}{nb-1} - 1} ||\boldsymbol{\delta_m}(1)||_\infty;$$

For $t \geqslant t_0$:

$$|(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 - (\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2| \leqslant \frac{4\sigma_f^4}{\zeta}(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(t_{l(t)-1})\|_\infty,$$

where $l(t)$ is the largest integer such that $t \geqslant \sum_{j=-1}^{l(t)} t_j$.

Proof: Line 9 in distributed GPR indicates that $\xi^{[i]}_{\boldsymbol{z}_*}(t) = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^{-2}$, and combining Line 5 in distributed GPR with (3.3) gives $\frac{1}{n}\sum_{j=1}^n r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) = (\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^{-2}$. Hence, we have

$$(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 - (\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \frac{\xi^{[i]}_{\boldsymbol{z}_*}(t) - \frac{1}{n}\sum_{j=1}^n r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)}{\frac{1}{n}\xi^{[i]}_{\boldsymbol{z}_*}(t)\sum_{j=1}^n r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)}. \tag{3.8}$$

The upper bound of $\xi^{[i]}_{\boldsymbol{z}_*}(t) - \frac{1}{n}\sum_{j=1}^n r^{[j]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t)$ is found by the following two claims, whose proofs are at the end.

Claim 3.4.4.1. It holds that

$$\max_{i\in\mathcal{V}}||\boldsymbol{\xi}^{[i]}(t) - \frac{1}{n}\sum_{j=1}^n \boldsymbol{r}^{[j]}_{\boldsymbol{\xi}}(t)||_\infty \leqslant ||\boldsymbol{\delta}_{\boldsymbol{m}}(t)||_\infty. \qquad \square$$

Claim 3.4.4.2. It holds that

$$\begin{aligned}
&||\boldsymbol{\delta}_{\boldsymbol{m}}(t)||_\infty \leqslant 2(1-\zeta)^{\frac{t}{nb-1}-1}||\boldsymbol{\delta}_{\boldsymbol{m}}(1)||_\infty, \quad \forall t < t_0;\\
&||\boldsymbol{\delta}_{\boldsymbol{m}}(t)||_\infty \leqslant 4(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(t_{l(t)-1})\|_\infty\frac{1}{\zeta}, \quad \forall t \geqslant t_0. \quad \square
\end{aligned}$$

Claim 3.4.3.1 and Claim 3.4.3.3 in Lemma 3.4.3 provide that $\xi^{[i]}_{\boldsymbol{z}_*}(t) \geqslant \frac{1}{\sigma_f^2}$ and $r^{[i]}_{\boldsymbol{\xi},\boldsymbol{z}_*}(t) \geqslant \frac{1}{\sigma_f^2}$ respectively. Combining this and Claim 3.4.4.1 and 3.4.4.2 with (3.8) finishes the proof. ■

Proof of Claim 3.4.4.1: Assumption 3.2.2, the initial condition $\boldsymbol{\xi}^{[i]}(0) = \boldsymbol{r}^{[i]}_{\boldsymbol{\xi}}(0)$, $\forall i \in \mathcal{V}$ and the update rule on Line 6 in distributed GPR render

$$\sum_{j=1}^n \boldsymbol{\xi}^{[j]}(t) = \sum_{j=1}^n \boldsymbol{\xi}^{[j]}(t-1) + \sum_{j=1}^n \Delta\boldsymbol{r}^{[j]}_{\boldsymbol{\xi}}(t) = \sum_{j=1}^n \boldsymbol{r}^{[j]}_{\boldsymbol{\xi}}(t).$$

Hence we have

$$\underline{\boldsymbol{m}}(t) = \min_{j\in\mathcal{V}} \boldsymbol{\xi}^{[j]}(t) \leqslant \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{r}_{\boldsymbol{\xi}}^{[j]}(t) \leqslant \max_{j\in\mathcal{V}} \boldsymbol{\xi}^{[j]}(t) = \bar{\boldsymbol{m}}(t)$$

$$\max_{j\in\mathcal{V}} ||\boldsymbol{\xi}^{[j]}(t) - \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{r}_{\boldsymbol{\xi}}^{[j]}(t)||_\infty \leqslant \|\bar{\boldsymbol{m}}(t) - \underline{\boldsymbol{m}}(t)\|_\infty = ||\boldsymbol{\delta}_{\boldsymbol{m}}(t)||_\infty. \qquad \square$$

Proof of Claim 3.4.4.2: Outline: Write $t = \tau_0 + \tau$ for some $\tau_0, \tau > 0$. We first derive a general form of $||\boldsymbol{\delta}_{\boldsymbol{m}}(\tau_0+\tau)||_\infty$. Then we prove the cases when $t < t_0$ and $t \geqslant t_0$ respectively.

First, we give the general form. Applying inequality (B.1) in [78] in vector form, we have

$$||\boldsymbol{\delta}_{\boldsymbol{m}}(\tau_0+\tau)||_\infty \leqslant \max\{2(1-\zeta)^{\frac{\tau}{nb-1}-1}||\boldsymbol{\delta}_{\boldsymbol{m}}(\tau_0)||_\infty, 2||\boldsymbol{\omega}(\tau_0,\tau_0+\tau)||_\infty\}, \tag{3.9}$$

$$\begin{aligned}\boldsymbol{\omega}(\tau_0,\tau) \triangleq (1-\zeta)^{\ell_\tau-1}\sum_{q=\tau_0}^{\tau_1+\tau_0-1}\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(q) + \cdots + (1-\zeta)\sum_{q=\tau_{\ell_\tau-2}+\tau_0}^{\tau_{\ell_\tau-1}+\tau_0-1}\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(q)\\ + \sum_{q=\tau_{\ell_\tau-1}+\tau_0}^{\tau_{\ell_\tau}+\tau_0-1}\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(q) + \sum_{q=\tau_{\ell_\tau}+\tau_0}^{\tau-1}\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(q).\end{aligned}$$

Second, we prove the case of $t < t_0$. Let $\tau_0 = 0$, $\tau = t$ in (3.9). We first find a uniform upper bound for $\boldsymbol{\omega}(0,t)$. Recall that Claim 3.4.3.4 shows that $\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(t) \preceq \bar{\Delta}\boldsymbol{r}_{\max}(t)$ and Claim 3.4.3.5 shows that $\bar{\Delta}\boldsymbol{r}_{\max}(t)$ is element-wise non-increasing. It follows that $\boldsymbol{\delta}_{\boldsymbol{r}_{\boldsymbol{\xi}}}(t) \preceq \bar{\Delta}\boldsymbol{r}_{\max}(1)$ for all $t \geqslant 1$ and hence

$$\begin{aligned}\|\boldsymbol{\omega}(0,t)\|_\infty &\leqslant (nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty(1+\sum_{l=0}^{\infty}(1-\zeta)^l)\\ &= (nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty(\frac{1}{\zeta}+1).\end{aligned}$$

Since $0 < \zeta \leqslant 1$, we can further write that $||\boldsymbol{\omega}(0,t)||_\infty \leqslant 2(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty\frac{1}{\zeta}$. Then (3.9) becomes

$$||\boldsymbol{\delta}_{\boldsymbol{m}}(t)||_\infty \leqslant \max\{2(1-\zeta)^{\frac{t}{nb-1}-1}\|\boldsymbol{\delta}_{\boldsymbol{m}}(1)\|_\infty, 4(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty\frac{1}{\zeta}\}. \tag{3.10}$$

Note that the time-dependent term on the right hand side of (3.10) is exponentially decreasing. Suppose t_0 is the smallest integer such that

$$(1-\zeta)^{\frac{t_0}{nb-1}-1}\|\boldsymbol{\delta_m}(1)\|_\infty \leqslant 2(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty\frac{1}{\zeta},$$

where t_0 can be obtained as defined. Hence we have

$$\|\boldsymbol{\delta_m}(t)\|_\infty \leqslant 2(1-\zeta)^{\frac{t}{nb-1}-1}||\boldsymbol{\delta_m}(1)||_\infty,\ \forall t < t_0.$$

which proves Claim 3.4.4.2 for $t < t_0$, and for $t \geqslant t_0$, we have

$$||\boldsymbol{\delta_m}(t)||_\infty \leqslant 4(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty\frac{1}{\zeta}. \tag{3.11}$$

Finally, we prove the case of $t \geqslant t_0$. Write $\tau_0 = t_0$. Then (3.9) becomes

$$||\boldsymbol{\delta_m}(t_0+\tau)||_\infty \leqslant \max\{2(1-\zeta)^{\frac{\tau}{nb-1}-1}||\boldsymbol{\delta_m}(t_0)||_\infty,\ 2||\boldsymbol{\omega}(t_0,t_0+\tau)||_\infty\}. \tag{3.12}$$

Applying analogous algebra as of $\boldsymbol{\omega}(0,t)$ gives

$$||\boldsymbol{\omega}(t_0,t_0+\tau)||_\infty \leqslant 2(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(t_0)\|_\infty\frac{1}{\zeta}.$$

Using this and (3.11) as the upper bound for $||\boldsymbol{\delta_m}(t_0)||_\infty$, we can rewrite (3.12) as

$$\begin{aligned}||\boldsymbol{\delta_m}(t_0+\tau)||_\infty \leqslant \max\{&8(1-\zeta)^{\frac{\tau}{nb-1}-1}(nb-1)\frac{\|\bar{\Delta}\boldsymbol{r}_{\max}(1)\|_\infty}{\zeta},\\ &4(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(t_0)\|_\infty\frac{1}{\zeta}\}.\end{aligned}$$

Similarly, let t_1 be the smallest integer such that

$$\|\bar{\Delta}\boldsymbol{r}_{\max}(t_0)\|_\infty \geqslant 2(1-\zeta)^{\frac{t_1}{nb-1}-1}\bar{\Delta}\boldsymbol{r}_{\max}(1).$$

Using similar manipulation as t_0 renders t_1 as defined and

$$||\boldsymbol{\delta_m}(t)||_\infty \leqslant 4(nb-1)\|\Delta\bar{\boldsymbol{r}}_{\max,\boldsymbol{\xi}}(t_0)\|_\infty\frac{1}{\zeta},\ \forall t \geqslant t_0+t_1.$$

By similar logic, we have t_j as defined for all $j \geqslant 1$ and $||\boldsymbol{\delta_m}(t)||_\infty \leqslant 4(nb-1)\|\bar{\Delta}\boldsymbol{r}_{\max}(t_{l(t)-1})\|_\infty \frac{1}{\zeta}$. □

Corollary 3.4.5 shows that the predictive variance from distributed GPR converges to that from aggregated method.

Corollary 3.4.5. Suppose the same conditions as in Proposition 3.4.4 hold. If $\lim_{t\to\infty} d^{[i]}(t) = 0$ for all $i \in \mathcal{V}$, then $\lim_{t\to\infty} |(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 - (\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2| = 0$. ■

3.4.2.3 Variance analysis of fused GPR

First of all, we show that $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t)$ exists.

Lemma 3.4.6. It holds that $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t)$ exists.

Proof: By Corollary 3.4.2, $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ converges. Hence in distributed GPR $\Delta r^{[i]}_{\boldsymbol{\lambda},\boldsymbol{z}_*}(t) \to 0$. By Line 11 in distributed GPR and Corollary 3.1 in [78], $\lim_{t\to\infty} (\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2$ exists. By Corollary 3.4.5 and (3.3), the convergence of $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ also implies the convergence of $(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$. Hence by definition of $\mathcal{Z}^{[i]}_{agg}(t)$ in Section 3.3.3, $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t)$ exists. ■

Lemma 3.4.7 below presents two properties of agent i where $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$.

Lemma 3.4.7. Suppose the same conditions for Corollary 3.4.5 hold and $d^{[j]}(t) \to 0$, $\forall j \in \mathcal{V}$. If $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ for some $i \in \mathcal{V}$, then $\psi^{[i]} < \frac{1}{n}\sum_{j=1}^n \psi^{[j]} \leqslant c$ and $\mu_\chi^{-1}\chi^{[i]} < 1$.

Proof: Outline: We first show that $\psi^{[i]} < \frac{1}{n}\sum_{j=1}^n \psi^{[j]} \leqslant c$. Then we show $\mu_\chi^{-1}\chi^{[i]} < 1$.

First, we show that $\psi^{[i]} < \frac{1}{n}\sum_{j=1}^n \psi^{[j]} \leqslant c$. Since $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$, we pick $\boldsymbol{z}_* \in \lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t)$. Note that $\lambda^{[i]}_{\boldsymbol{z}_*}(t)$ is tracking the signal $r^{[i]}_{\boldsymbol{\lambda},\boldsymbol{z}_*}(t) = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ using FODAC algorithm in [78]. By Corollary 3.4.2 and Corollary 3.1 in [78], $\lim_{t\to\infty} \lambda^{[i]}_{\boldsymbol{z}_*}(t) = \lim_{t\to\infty} \frac{1}{n}\sum_{j=1}^n \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t)}$. By Line 11 in distributed GPR, we have $\lim_{t\to\infty} (\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \lim_{t\to\infty} \lambda^{[i]}_{\boldsymbol{z}_*}(t)$. Since $\boldsymbol{z}_* \in \lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t)$, by Corollay 3.4.2 and the definition of $\mathcal{Z}^{[i]}_{agg}(t)$ on Line 2 in fused GPR, we have

$$\sigma_f^2 - \sigma_f^2\psi^{[i]} = \sigma_f^2 - \frac{\sigma_f^4}{\sigma_f^2 + (\sigma_e^{[i]})^2} = \frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2} = \lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$$

$$> \lim_{t\to\infty}(\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \lim_{t\to\infty}\frac{1}{n}\sum_{j=1}^{n}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t)}$$
$$= \frac{1}{n}\sum_{j=1}^{n}(\sigma_f^2 - \frac{\sigma_f^4}{\sigma_f^2+(\sigma_e^{[j]})^2}) = \sigma_f^2 - \frac{\sigma_f^2}{n}\sum_{j=1}^{n}\psi^{[j]},$$

which renders $\psi^{[i]} < \frac{1}{n}\sum_{j=1}^{n}\psi^{[j]}$.

Since both $\psi^{[i]}$ and $\chi^{[i]}$ are monotonically decreasing as $\sigma_e^{[i]}$ increases, $\chi^{[i]} \leqslant \chi^{[j]}$ if and only if $\psi^{[i]} \leqslant \psi^{[j]}$. Without loss of generality, assume that $\chi^{[1]} \leqslant \cdots \leqslant \chi^{[n]}$ and $\psi^{[1]} \leqslant \cdots \leqslant \psi^{[n]}$. Then Chebyshev's sum inequality [87] gives

$$\frac{1}{n}\sum_{j=1}^{n}\psi^{[j]} \leqslant (\frac{1}{n}\sum_{j=1}^{n}\chi^{[j]})^{-1}\big(\frac{1}{n}\sum_{j=1}^{n}\chi^{[j]}\psi^{[j]}\big) = c. \tag{3.13}$$

Hence $\psi^{[i]} < \frac{1}{n}\sum_{j=1}^{n}\psi^{[j]} \leqslant c$.

Second, we show that $\mu_\chi^{-1}\chi^{[i]} < 1$. By definition and Corollary 3.4.2, $\chi^{[i]} = \lim_{t\to\infty}\check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$. By (3.3), $\mu_\chi^{-1} = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ for all $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$. Since $\boldsymbol{z}_* \in \lim_{t\to\infty}\mathcal{Z}^{[j]}_{agg}(t)$, $\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} > \lim_{t\to\infty}(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$, where the equality follows from Corollary 3.4.5. Hence

$$\mu_\chi^{-1}\chi^{[i]} = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} < 1. \qquad \blacksquare$$

Now we present of proof of Theorem 3.3.3 Part II.

Proof of Theorem 3.3.3 Part II: By Line 4 in fused GPR, it is obvious that if $\mathcal{Z}^{[i]}_{agg}(t) = \emptyset$, $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) = 0$. Now we consider the case when $\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$.

Outline: The proof is composed of three parts: expression of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ and its uniform lower bound; verfication of the selection of $g(\boldsymbol{z}_*,t)$; derivation of the growth factor of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$.

First, we show the expression of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ and derive its uniform lower bound. According to Line 12 in fused GPR, we have $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$, where

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) \triangleq \gamma^{[i]}_{\sigma,\boldsymbol{z}_*,1}(t)\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t)\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,3}(t)\big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2,$$
$$\big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2 \triangleq \min\big\{\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}, \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\big\},$$

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,1}(t) \triangleq \frac{k\big(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*}(t)\big)^2 \cdot \max\{0, (c-\psi^{[i]})^2\}}{k\big(\boldsymbol{z}_*, \boldsymbol{z}_*\big)^2},$$
$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t) \triangleq \frac{\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)} - (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2}{\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}},$$
$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,3}(t) \triangleq \frac{\big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2}{\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}}. \tag{3.14}$$

Line 2 of fused GPR rules that $(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 < \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$. Obviously, $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) \geqslant 0$.

Second, we verify the selection of $g(\boldsymbol{z}_*, t)$. We verify that the selection of $g(\boldsymbol{z}_*, t)$ is valid by showing $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 > 0$. We analyze each factor of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ as follows.

Note that $c, \psi^{[i]} \in (0,1)$, hence $0 \leqslant \max\{0, (c-\psi^{[i]})^2\} \leqslant 1$. The decomposition and monotonicity properties in Assumption 3.3.1 gives $0 \leqslant \frac{k\left(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*}(t)\right)^2\}}{k\left(\boldsymbol{z}_*, \boldsymbol{z}_*\right)^2} \leqslant 1$. Combining these gives $0 \leqslant \gamma^{[i]}_{\sigma,\boldsymbol{z}_*,1} \leqslant 1$.

Line 2 of fused GPR rules that $(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 < \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$. By Claim 3.4.3.3 and Line 9 in distributed GPR, $(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 > 0$. Therefore $0 < \gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2} < 1$.

By definition, $\big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2 \leqslant \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$, which renders $0 < \gamma^{[i]}_{\sigma,\boldsymbol{z}_*,3} \leqslant 1$.

The above upper bounds give $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t) < \big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2 \leqslant \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}$ and $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 > \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = 0$.

Finally, we derive the growth factor of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$. According to the definition of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ in (3.14), we can derive the growth factor of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$ by analyzing the growth factor of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,1}(t)$, $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t)$, $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,3}(t)$ and $\big(\check{\sigma}^{[i]}_{\min,\boldsymbol{z}_*}(t)\big)^2$ respectively.

We first consider $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t)$. Let

$$\hat{c} \triangleq \max\{\frac{4\sigma_f^4}{\zeta}(nb-1), 2\sigma_f^4||\boldsymbol{\delta_m}(1)||_\infty\}.$$

The upper bound given in Proposition 3.4.4 can be written as

$$h(t) \triangleq \begin{cases} \hat{c}(1-\zeta)^{\frac{t}{nb-1}-1}, & t < t_0, \\ \hat{c}\|\bar{\Delta}\boldsymbol{r}_{\max}(t_{l(t)-1})\|_\infty, & t \geqslant t_0. \end{cases} \tag{3.15}$$

Then Proposition 3.4.4 gives

$$(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 \leqslant (\check{\sigma}^{(agg)}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 + h(t).$$

Hence we have lower bound

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t) \geqslant \frac{\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)} - (\check{\sigma}^{(agg)}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 - h(t)}{\check{\sigma}^4_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}}.$$

The upper bound and lower bound of $(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ is given below, whose proof can be found at the end of the proof.

Claim 3.4.7.1. For each $\boldsymbol{z}_* \in \mathcal{Z}_*$, the aggregated variance returned from (3.3) can be characterized as

$$\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2 + (\sigma_e^{\min})^2} \leqslant (\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 \leqslant \sigma_f^2 - \frac{\frac{1}{n}\sum_{i=1}^n \kappa\big(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*}\big)^2}{\sigma_f^2 + (\sigma_e^{\max})^2}. \quad \square$$

Denote $\phi^{[i]} \triangleq \sigma_f^2 + (\sigma_e^{[i]})^2$. Plugging in equality (3.5) for $\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$ and the upper bound in Claim 3.4.7.1 for $(\check{\sigma}^{(agg)}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2$ in the inequality above gives

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t) \geqslant \frac{\phi^{[i]}}{\phi^{\max}}\Big(\frac{\frac{\phi^{[i]}}{n}\sum_{j=1}^n \kappa\big(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)}\big)^2}{\big(\sigma_f^2\phi^{[i]} - \kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2\big)^2} \frac{-\phi^{\max}\kappa\big(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)}\big)^2 - \phi^{\max}\phi^{[i]}h(t)}{\big(\sigma_f^2\phi^{[i]} - \kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2\big)^2}\Big).$$

The boundedness of κ in Assumption 3.3.1 gives $|\sigma_f^2\phi^{[i]} - \kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2| \leqslant \sigma_f^2\phi^{[i]}$. Applying this upper bound to the denominator of the lower bound above gives

$$\gamma^{[i]}_{\sigma,\boldsymbol{z}_*,2}(t) \geqslant \frac{1}{\phi^{[i]}\phi^{\max}\sigma_f^4}\Big(\frac{\phi^{[i]}}{n}\sum_{j=1}^n \kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2 - \phi^{\max}\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2 - \phi^{\max}\phi^{[i]}h(t)\Big).$$

Now we characterize the rest of factors of $\gamma^{[i]}_{\sigma,\boldsymbol{z}_*}(t)$. Theorem 3.3.3 By monotonicity and decomposition in Assumption 3.3.1, we have $k\big(\boldsymbol{z}_*, \boldsymbol{z}_*\big)^2 = \sigma_f^4$ and

$k(\boldsymbol{z}_*, \boldsymbol{z}_{agg*}^{[i]}(t))^2 = \kappa(\rho_{\boldsymbol{z}_*}^{\boldsymbol{z}_{agg*}^{[i]}(t)})^2$. By Lemma 3.4.7, we have

$$c - \psi^{[i]} \geqslant \frac{1}{n}\sum_{j=1}^{n}\psi^{[j]} - \psi^{[i]} = \sigma_f^2\beta_\psi^{[i]} > 0,$$

where $\beta_\psi^{[i]} \triangleq (\frac{1}{n}\sum_{j=1}^{n}(\phi^{[j]})^{-1} - (\phi^{[i]})^{-1})$. This gives $\gamma_{\sigma,\boldsymbol{z}_*,1}^{[i]}(t) \geqslant \kappa(\rho_{\boldsymbol{z}_*}^{\boldsymbol{z}_{agg*}^{[i]}(t)})^2\beta_\psi^{[i]}/\sigma_f^2$.

Part I indicates that $\check{\sigma}^2_{\boldsymbol{z}_{agg*}^{[i]}(t)|\mathcal{D}^{[i]}(t)} \geqslant \frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2}$. Therefore we have $\big(\check{\sigma}_{\min,\boldsymbol{z}_*}^{[i]}(t)\big)^4 \geqslant \frac{\sigma_f^4(\sigma_e^{[i]})^4}{(\phi^{[i]})^2}$.

Equality (3.5) indicates $\check{\sigma}^2_{\boldsymbol{z}_{agg*}^{[i]}(t)|\mathcal{D}^{[i]}(t)} \leqslant \sigma_f^2$. Combining this with the lower bound of $\big(\check{\sigma}_{\min,\boldsymbol{z}_*}^{[i]}(t)\big)^4$ above gives $\gamma_{\sigma,\boldsymbol{z}_*,3}^{[i]}(t) \geqslant \frac{\sigma_f^2(\sigma_e^{[i]})^4}{(\phi^{[i]})^2}$.

Combining the lower bounds of all the factors gives

$$\gamma_{\sigma,\boldsymbol{z}_*}^{[i]}(t) \geqslant \kappa(\rho_{\boldsymbol{z}_*}^{\boldsymbol{z}_{agg*}^{[i]}(t)})^2\Big(\frac{(\sigma_e^{[i]})^8}{(\phi^{[i]})^3\phi^{\max}}\big(\frac{\phi^{[i]}}{n}\sum_{j=1}^{n}\kappa(\rho_{\boldsymbol{z}_{agg*}^{[i]}(t)}^{\mathcal{Z}^{[j]}(t)})^2 - \phi^{\max}\kappa(\rho_{\boldsymbol{z}_{agg*}^{[i]}(t)}^{\mathcal{Z}^{[i]}(t)})^2 - \phi^{\max}\phi^{[i]}h(t)\big)\Big)\beta_\psi^{[i]}.$$

The definition in (3.15) and Claim 3.4.3.6 renders $h(t) \to 0$. This renders the Big O notion. ■

Proof of Claim 3.4.7.1: Using equality (3.5), we can characterize $(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ in (3.3) as

$$(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^{-2} = \frac{1}{n}\sum_{i=1}^{n}(\sigma_f^2 - \frac{\kappa\big(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}\big)^2}{\sigma_f^2+(\sigma_e^{[i]})^2})^{-1}.$$

Taking the inverse and applying Lemma 3.4.1 by substituting x_i with $(\sigma_f^2 - \frac{\kappa\big(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}\big)^2}{\sigma_f^2+(\sigma_e^{[i]})^2})^{-1}$ to f_1 gives

$$(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 \leqslant \sigma_f^2 - \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa\big(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}\big)^2}{\sigma_f^2+(\sigma_e^{[i]})^2} \leqslant \sigma_f^2 - \frac{\frac{1}{n}\sum_{i=1}^{n}\kappa\big(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}\big)^2}{\sigma_f^2+(\sigma_e^{\max})^2}.$$

The lower bound provided in Part I of Theorem 3.3.3 and equation (3.5) give

$$(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^{-2} \leqslant \frac{1}{n}\sum_{j=1}^{n}(\frac{\sigma_f^2(\sigma_e^{[j]})^2}{\sigma_f^2+(\sigma_e^{[j]})^2})^{-1} \leqslant (\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2+(\sigma_e^{\min})^2})^{-1},$$

where the last inequality follows from the fact that $\frac{\sigma_f^2\sigma_e^2}{\sigma_f^2+\sigma_e^2}$ monotonically increases with respect to σ_e^2, i.e., $\frac{d}{d\sigma_e^2}\frac{\sigma_f^2\sigma_e^2}{\sigma_f^2+\sigma_e^2} = \frac{\sigma_f^4}{(\sigma_f^2+\sigma_e^2)^2} > 0$. Taking the inverse gives the lower bound. □

3.4.3 Proof of Theorem 3.3.8

In this section, we present the theoretical results that leads to Theorem 3.3.8. We first present the error between the predictive mean of agent-based GPR and the ground truth, which is the result of Theorem 3.3.8 Part I, in Section 3.4.3.1. Secondly, we characterize the predictive mean returned from distributed GPR in Proposition 3.4.9 in Section 3.4.3.2. Lastly, we finish the proof for Part II of Theorem 3.3.8 in Section 3.4.3.3.

3.4.3.1 Mean analysis of agent-based GPR

In this section, we provide the proof of Part I of Theorem 3.3.8.

Proof of Part I of Theorem 3.3.8: By Assumption 3.3.5 and decomposition and monotonicity properties in Assumption 3.3.1, Line 4 of agent-based GPR becomes $\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2+(\sigma_e^{[i]})^2} \cdot y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)}$. It implies that

$$\begin{aligned}\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*) =&(1 - \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2})(-\eta(\boldsymbol{z}_*)) + \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2}\Big(y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} - \eta(\boldsymbol{z}_*^{[i]}(t))\Big)\\ &+ \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2}\Big(\eta(\boldsymbol{z}_*^{[i]}(t)) - \eta(\boldsymbol{z}_*)\Big).\end{aligned}$$

By boundedness of κ in Assumption 3.3.1, $0 < \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2+(\sigma_e^{[i]})^2} < 1$. Combining this with triangular inequality gives

$$\begin{aligned}|\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*)| <&(1 - \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2})|\eta(\boldsymbol{z}_*)| + |y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} - \eta(\boldsymbol{z}_*^{[i]}(t))| &(3.16)\\ &+ |\eta(\boldsymbol{z}_*^{[i]}(t)) - \eta(\boldsymbol{z}_*)|. &(3.17)\end{aligned}$$

Now we analyze the upper bound of each term on the right hand side of (3.16).

Recall that $\boldsymbol{z}_*^{[i]}(t) \in \text{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}(t))$. Utilizing the Lipschitz continuity of η in Assumption 3.3.7 gives

$$|\eta(\boldsymbol{z}_*^{[i]}(t)) - \eta(\boldsymbol{z}_*)| \leqslant \ell_\eta \rho(\boldsymbol{z}_*, \boldsymbol{z}_*^{[i]}(t)) = \ell_\eta \rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}.$$

The observation model (4.1) gives $y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} \sim \mathcal{N}(\eta(\boldsymbol{z}_*^{[i]}(t)), (\sigma_e^{[i]})^2)$. Therefore by Chebyshev inequality (page 151, [88]), for all $\epsilon > 0$, we have

$$P\{|y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} - \eta(\boldsymbol{z}_*^{[i]}(t))| \geqslant \epsilon\} \leqslant \frac{(\sigma_e^{[i]})^2}{\epsilon^2}.$$

Note that $|\eta(\boldsymbol{z}_*)| \leqslant \|\eta\|_{\boldsymbol{\mathcal{Z}}}$. Applying these two inequalities to (3.16) gives

$$|\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*)| \leqslant (1 - \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2})\|\eta\|_{\boldsymbol{\mathcal{Z}}} + \epsilon + \ell_\eta \rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}$$

with probability at least $1 - \frac{(\sigma_e^{[i]})^2}{\epsilon^2} \geqslant 1 - \frac{(\sigma_e^{\max})^2}{\epsilon^2}$. The proof is completed by using inequality $d^{[i]}(t) \geqslant \rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)}$ and the monotonicity property of κ in Assumption 3.3.1. ■

Remark 3.4.8. We can write $\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \check{r}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t)$, where $\check{r}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})\eta(\boldsymbol{z}_*^{[i]}(t))}{\sigma_f^2 + (\sigma_e^{[i]})^2}$ depends on latent function η and $\check{e}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq \frac{\kappa(\rho_{\boldsymbol{z}_*}^{\mathcal{Z}^{[i]}(t)})}{\sigma_f^2 + (\sigma_e^{[i]})^2}\big(y^{[i]}_{\boldsymbol{z}_*^{[i]}(t)} - \eta(\boldsymbol{z}_*^{[i]}(t))\big)$ depends on measurement noise. Denote $\check{r}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty} \check{r}^{[i]}_{\boldsymbol{z}_*}(t) = \frac{\sigma_f^2 \eta(\boldsymbol{z}_*)}{\sigma_f^2 + (\sigma_e^{[i]})^2}$ and $\check{e}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty} \check{e}^{[i]}_{\boldsymbol{z}_*}(t) = \frac{\sigma_f^2}{\sigma_f^2 + (\sigma_e^{[i]})^2} e^{[i]}_{\boldsymbol{z}_*}$, where $e^{[i]}_{\boldsymbol{z}_*} \triangleq y^{[i]}_{\boldsymbol{z}_*} - \eta(\boldsymbol{z}_*) \sim \mathcal{N}(0, (\sigma_e^{[i]})^2)$ independent over agent $i \in \mathcal{V}$ and input $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$. It is obvious that $\lim_{t\to\infty} \check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \check{r}^{[i]}_{\boldsymbol{z}_*} + \check{e}^{[i]}_{\boldsymbol{z}_*}$. ■

3.4.3.2 Mean analysis of distributed GPR

Before presenting the results, we derive the solution to the consensus state $\theta^{[i]}_{\boldsymbol{z}_*}(t)$, $i \in \mathcal{V}$, in terms of input signal $\Delta r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$. We also show the decompositions of $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$ and $\theta^{[i]}_{\boldsymbol{z}_*}(t)$, which separate the two terms into real-valued parts and stochastic parts.

First, we give the solution to $\theta^{[i]}_{\boldsymbol{z}_*}(t)$. Let vectors $\boldsymbol{\theta}_{\boldsymbol{z}_*}(t) \triangleq [\theta^{[1]}_{\boldsymbol{z}_*}(t), \cdots, \theta^{[n]}_{\boldsymbol{z}_*}(t)]^T$ and $\boldsymbol{r}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t) \triangleq [r^{[1]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t), \cdots, r^{[n]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)]^T$. Line 4 of distributed GPR across the network $\mathcal{V}$ can be represented by discrete linear time-varying (LTV) system: $\boldsymbol{\theta}_{\boldsymbol{z}_*}(t) =$

$A(t-1)\boldsymbol{\theta}_{\boldsymbol{z}_*}(t-1) + \Delta \boldsymbol{r}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$. By page 111 in [89], the solution to this system is:

$$\boldsymbol{\theta}_{\boldsymbol{z}_*}(t) = \Phi(t,0)\boldsymbol{\theta}_{\boldsymbol{z}_*}(0) + \sum_{l=1}^{t} \Phi(t,l)\Delta \boldsymbol{r}_{\boldsymbol{\theta},\boldsymbol{z}_*}(l), \tag{3.18}$$

where $\Phi(t,l) \triangleq \prod_{\tau=l}^{t-1} A(\tau)$.

Second, we show the decomposition of $\Delta \boldsymbol{r}_{\boldsymbol{\theta},\boldsymbol{z}_*}(l)$ into a signal depending on η and a zero-mean stochastic process. By definition of $r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t)$ in Line 3 in distributed GPR and Remark 3.4.8, it holds that

$$r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t) = \breve{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}(\breve{r}^{[i]}_{\boldsymbol{z}_*}(t) + \breve{e}^{[i]}_{\boldsymbol{z}_*}(t)) = \hat{r}^{[i]}_{\boldsymbol{z}_*}(t) + \hat{e}^{[i]}_{\boldsymbol{z}_*}(t),$$

where $\hat{r}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq \breve{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\breve{r}^{[i]}_{\boldsymbol{z}_*}(t)$ and $\hat{e}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq \breve{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\breve{e}^{[i]}_{\boldsymbol{z}_*}(t)$ is a Gaussian random variable with zero mean. Hence we have

$$\Delta r^{[i]}_{\boldsymbol{\theta},\boldsymbol{z}_*}(t) = \Delta \hat{r}^{[i]}_{\boldsymbol{z}_*}(t) + \Delta \hat{e}^{[i]}_{\boldsymbol{z}_*}(t). \tag{3.19}$$

Denote $\hat{r}^{[i]}_{\boldsymbol{z}_*} \triangleq \lim_{t\to\infty} \hat{r}^{[i]}_{\boldsymbol{z}_*}(t)$. Corollary 3.4.2 and Remark 3.4.8 give

$$\hat{r}^{[i]}_{\boldsymbol{z}_*} = \lim_{t\to\infty} \breve{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\breve{r}^{[i]}_{\boldsymbol{z}_*}(t) = \left(\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2}\right)^{-1}\breve{r}^{[i]}_{\boldsymbol{z}_*}$$

and

$$\lim_{t\to\infty} \hat{e}^{[i]}_{\boldsymbol{z}_*}(t) = \left(\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2 + (\sigma_e^{[i]})^2}\right)^{-1}\breve{e}^{[i]}_{\boldsymbol{z}_*} \tag{3.20}$$

is zero mean Gaussian.

Third, we show the decomposition of (3.18). The solution (3.18) can be decomposed into a solution to FODAC [78] with respect to a signal depending on η and a solution to FODAC with respect to a zero-mean stochastic process.

Let $\hat{\boldsymbol{r}}_{\boldsymbol{z}_*}(t) \triangleq [\hat{r}^{[1]}_{\boldsymbol{z}_*}(t), \cdots, \hat{r}^{[n]}_{\boldsymbol{z}_*}(t)]^T$ and $\hat{\boldsymbol{e}}_{\boldsymbol{z}_*}(t) \triangleq [\hat{e}^{[1]}_{\boldsymbol{z}_*}(t), \cdots, \hat{e}^{[n]}_{\boldsymbol{z}_*}(t)]^T$. By (3.19), we can write (3.18) as

$$\boldsymbol{\theta}_{\boldsymbol{z}_*}(t) = \boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) + \boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{e}}(t), \tag{3.21}$$

$$\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) \triangleq \Phi(t,0)\boldsymbol{\theta}_{\boldsymbol{z}_*}(0) + \sum_{l=1}^{t} \Phi(t,l)\Delta\hat{\boldsymbol{r}}_{\boldsymbol{z}_*}(l),$$

$$\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) \triangleq \sum_{l=1}^{t} \Phi(t,l)\Delta\hat{\boldsymbol{e}}_{\boldsymbol{z}_*}(l).$$

Then Proposition 3.4.9 characterizes the predictive mean.

Proposition 3.4.9. *(Prediction decomposition).* Suppose Assumptions 3.2.1, 3.2.2, 3.2.3 and 3.3.1 hold. If $\lim_{t\to\infty} d^{[j]}(t) = 0$ for all $j \in \mathcal{V}$, then for all $\boldsymbol{z}_* \in \mathcal{Z}_{agg}$,

$$\hat{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\big(\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) + \theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)\big),$$

where $\lim_{t\to\infty}\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) = \frac{1}{n}\sum_{j=1}^{n}\hat{r}^{[j]}_{\boldsymbol{z}_*}$, $\theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$ is a Gaussian random variable with zero mean and $\lim_{t\to\infty}\sum_{j=1}^{n}\theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{j=1}^{n}(\sigma_e^{[j]})^{-2}e^{[j]}_{\boldsymbol{z}_*}$.

Proof: By (3.21) and Line 10 of distributed GPR, we have

$$\hat{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\big(\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) + \theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)\big).$$

First, we show that $\lim_{t\to\infty}\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) = \frac{1}{n}\sum_{j=1}^{n}\hat{r}^{[j]}_{\boldsymbol{z}_*}$. Analogous to $\boldsymbol{\theta}_{\boldsymbol{z}_*}(t)$, $\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{r}}(t)$ is the solution for tracking the average of the signal $\hat{\boldsymbol{r}}_{\boldsymbol{z}_*}(t)$ using FODAC algorithm [78]. Since $\hat{r}^{[i]}_{\boldsymbol{z}_*} = (\frac{\sigma_f^2(\sigma_e^{[i]})^2}{\sigma_f^2+(\sigma_e^{[i]})^2})^{-1}\check{r}^{[i]}_{\boldsymbol{z}_*}$, $\forall i \in \mathcal{V}$, we have $\lim_{t\to\infty}\Delta\hat{r}^{[i]}_{\boldsymbol{z}_*}(t) = 0$. Combining this with Corollary 3.1 in [78] gives

$$\lim_{t\to\infty}\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{r}}(t) = \lim_{t\to\infty}\Big(\frac{1}{n}\sum_{j=1}^{n}\hat{r}^{[j]}_{\boldsymbol{z}_*}(t)\Big)\mathbf{1}_n = \Big(\frac{1}{n}\sum_{j=1}^{n}\hat{r}^{[j]}_{\boldsymbol{z}_*}\Big)\mathbf{1}_n.$$

Second, we show that $\theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$ is a Gaussian random variable with zero mean. Note that $\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = [\theta^{[1]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t), \cdots, \theta^{[n]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)]^T$. Similar to $\boldsymbol{\theta}_{\boldsymbol{z}_*}(t)$, $\boldsymbol{\theta}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$ is the solution for tracking the average of $e^{[i]}_{\boldsymbol{z}_*}(t)$ using FODAC:

$$\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{l=1}^{n} a_{ij}(t-1)\theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t-1) + \Delta\hat{e}^{[i]}_{\boldsymbol{z}_*}(t), \tag{3.22}$$

with initial state $\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(0) = 0$. Note that

$$\Delta\hat{e}^{[j]}_{\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t)}\check{e}^{[j]}_{\boldsymbol{z}_*}(t) - \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[j]}(t-1)}\check{e}^{[j]}_{\boldsymbol{z}_*}(t-1).$$

Recall that $\check{e}^{[j]}_{\boldsymbol{z}_*}(t) = \frac{\kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}_*})}{\sigma_f^2+(\sigma_e^{[j]})^2}e^{[j]}_{\boldsymbol{z}_*^{[j]}(t)}$ in Remark 3.4.8 where $e^{[j]}_{\boldsymbol{z}}$ is a zero-mean Gaussian random variable independent over $j \in \mathcal{V}$ and $\boldsymbol{z} \in \mathcal{V}$. Hence $\check{e}^{[j]}_{\boldsymbol{z}_*}(t)$ and $\hat{e}^{[j]}_{\boldsymbol{z}_*}(t)$ are both zero-mean Gaussian random variables. Therefore, it follows from (3.22) that $\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$ is a Gaussian random variable with zero mean for all $t \geqslant 1$ (Theorem 5.5-1, [90]).

Finally, we show that $\lim_{t\to\infty}\sum_{j=1}^n \theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{j=1}^n(\sigma_e^{[j]})^{-2}e^{[j]}_{\boldsymbol{z}_*}$. By Assumption 3.2.2 and initial state $\theta^{[j]}_{\boldsymbol{z}_*}(0) = r^{[j]}_{\theta,\boldsymbol{z}_*}(0)$, which indicates $\theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(0) = \hat{e}^{[j]}_{\boldsymbol{z}_*}(0) = 0$ for each $j \in \mathcal{V}$, (3.22) renders

$$\sum_{j=1}^n \theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{j=1}^n \theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t-1) + \sum_{j=1}^n \Delta\hat{e}^{[j]}_{\boldsymbol{z}_*}(t) = \sum_{j=1}^n \hat{e}^{[j]}_{\boldsymbol{z}_*}(t),$$

for all $t \geqslant 1$. Therefore $\lim_{t\to\infty}\sum_{j=1}^n \theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \lim_{t\to\infty}\sum_{j=1}^n \hat{e}^{[j]}_{\boldsymbol{z}_*}(t)$. Combining this with (3.20) and the definition of $\check{e}^{[j]}_{\boldsymbol{z}_*}$ in Remark 3.4.8 gives $\lim_{t\to\infty}\sum_{j=1}^n \theta^{[j]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{j=1}^n(\sigma_e^{[j]})^{-2}e^{[j]}_{\boldsymbol{z}_*}$. ■

3.4.3.3 Mean analysis of fused GPR

This section provides the analysis of predictive mean returned by fused GPR. Recall that Lemma 3.4.6 shows that $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t)$ exists. Hence, the main results in this section are Proposition 3.4.13, where the case $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ is discussed, and Lemma 3.4.14, where a sufficient condition for $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ is presented. Then we discuss the case of $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) = \emptyset$ to conclude the proof of Theorem 3.3.8. We first discuss the case of $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$.

Remark 3.4.8 and Proposition 3.4.9 respectively render

$$\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \check{r}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t), \tag{3.23}$$

$$\hat{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} = \tilde{r}^{[i]}_{\boldsymbol{z}_*}(t) + \tilde{e}^{[i]}_{\boldsymbol{z}_*}(t), \tag{3.24}$$

where $\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{r}}(t)$ and $\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*,e}(t)$ is zero-mean, $\forall \boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$. Lemma 3.4.10 summarizes the limiting behaviors of the above variables.

Lemma 3.4.10. Suppose the same conditions for Proposition 3.4.9 hold and $d^{[j]}(t) \to 0$, $\forall j \in \mathcal{V}$. It holds that $\forall \boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$,

$$\check{r}^{[i]}_{\boldsymbol{z}_*} = \psi^{[i]}\eta(\boldsymbol{z}_*),\ \lim_{t\to\infty}\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t) = c\eta(\boldsymbol{z}_*)$$
$$\check{e}^{[i]}_{\boldsymbol{z}_*} = \psi^{[i]}e^{[i]}_{\boldsymbol{z}_*},\ \lim_{t\to\infty}\sum_{j=1}^{n}\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t) = \sum_{j=1}^{n}\left(\mu_\chi^{-1}\chi^{[j]}\psi^{[j]}e^{[j]}_{\boldsymbol{z}_*}\right).$$

Proof: Combining the definition of $\psi^{[i]}$ and Remark 3.4.8 directly renders $\check{r}^{[i]}_{\boldsymbol{z}_*} = \psi^{[i]}\eta(\boldsymbol{z}_*)$ and $\check{e}^{[i]}_{\boldsymbol{z}_*} = \psi^{[i]}e^{[i]}_{\boldsymbol{z}_*}$.

Corollary 3.4.2 shows that $\lim\limits_{t\to\infty}\check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} = \chi^{[i]}$. Then Corollary 3.4.5 and (3.3) render

$$\lim_{t\to\infty}(\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2 = \mu_\chi^{-1}.$$

Combining this with the definition of $\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t)$, Proposition 3.4.9, and the above result about $\check{r}^{[i]}_{\boldsymbol{z}_*}$ renders $\lim\limits_{t\to\infty}\tilde{r}^{[i]}_{\boldsymbol{z}_*}(t) = c\eta(\boldsymbol{z}_*)$.

Combining the definition of $\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)$ with Proposition 3.4.9 gives $\lim\limits_{t\to\infty}\sum_{j=1}^{n}\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t) = \sum_{j=1}^{n}\left(\mu_\chi^{-1}\chi^{[j]}\psi^{[j]}e^{[j]}_{\boldsymbol{z}_*}\right)$. ■

Next we introduce necessary notations to continue the analysis. Since $\lim\limits_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ and Corollary 3.4.2 hold, $\boldsymbol{z}^{[i]}_{agg*} \in \lim\limits_{t\to\infty}\mathrm{proj}(\boldsymbol{z}_*, \mathcal{Z}^{[i]}_{agg}(t))$ and $g(\boldsymbol{z}_*) \triangleq \lim\limits_{t\to\infty}g(\boldsymbol{z}_*, t)$ exist. Line 11 of fused GPR gives

$$(\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} - \eta(\boldsymbol{z}_*))^2 = (\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))^2 + s^{[i]}_{\boldsymbol{z}_*}(t),$$

where $s^{[i]}_{\boldsymbol{z}_*}(t) \triangleq s^{[i]}_{\boldsymbol{z}_*,1}(t) + s^{[i]}_{\boldsymbol{z}_*,2}(t)$ with

$$s^{[i]}_{\boldsymbol{z}_*,1}(t) \triangleq 2(\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))v^{[i]}_{\boldsymbol{z}_*}(t)\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)},$$
$$s^{[i]}_{\boldsymbol{z}_*,2}(t) \triangleq \left(v^{[i]}_{\boldsymbol{z}_*}(t)\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)}\right)^2.$$

Let $v^{[i]}_{\boldsymbol{z}_*} \triangleq \lim\limits_{t\to\infty}v^{[i]}_{\boldsymbol{z}_*}(t)$, whose existence, according to its definition, is guaranteed by

the existences of $\boldsymbol{z}^{[i]}_{agg*}$, $g(\boldsymbol{z}^{[i]}_{agg*})$, and Corollary 3.4.2. Denote

$$q^{[i]}_{\boldsymbol{z}_*,1} \triangleq (\psi^{[i]}-1)(c-\psi^{[i]})k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*}),$$
$$q^{[i]}_{\boldsymbol{z}_*,2} \triangleq (c-\psi^{[i]})^2\sigma_f^2 + \limsup_{t\to\infty}\mathbb{E}[(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}}(t) - \check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}}(t))^2].$$

Lemmas 3.4.11 and 3.4.12 characterize the limit of $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$ and $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,2}(t)]$ in terms of $q^{[i]}_{\boldsymbol{z}_*,1}$ and $q^{[i]}_{\boldsymbol{z}_*,2}$, respectively.

Lemma 3.4.11. Suppose the same conditions in Theorem 3.3.8 Part II hold and $d^{[j]}(t)\to 0$, $\forall j\in\mathcal{V}$. If $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t)\neq\emptyset$ for some $i\in\mathcal{V}$, then $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)] \leqslant 2v^{[i]}_{\boldsymbol{z}_*}q^{[i]}_{\boldsymbol{z}_*,1}$.

Proof: Outline: We first give the expression of $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$. Then we analyze the limit of each term in the expression $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$. Finally, we plug in the terms and derive the upper bound of $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$.

First, we give the expression of $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$. Using the definition of $\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)}$ in Line 10 of fused GPR and plugging in (3.23) and (3.24), we have

$$\begin{aligned}\mu'^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)} &= \tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t), \\ s^{[i]}_{\boldsymbol{z}_*,1}(t) &= 2\big(\check{r}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t) - \eta(\boldsymbol{z}_*)\big)v^{[i]}_{\boldsymbol{z}_*}(t)\big(\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \\ &\quad - \check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big).\end{aligned} \tag{3.25}$$

Note that $v^{[i]}_{\boldsymbol{z}_*}(t) = g^{[i]}_{\boldsymbol{z}_*}(t)\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$, where the right hand side only depends on $\boldsymbol{z}_*$ and $\boldsymbol{z}^{[i]}_{agg*}(t)$ instead of η or $e^{[i]}_{\boldsymbol{z}}$ that is random. This gives

$$\begin{aligned}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)] =& 2v^{[i]}_{\boldsymbol{z}_*}(t)\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \\ &- \check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \\ &- \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \eta(\boldsymbol{z}_*)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \\ &- \eta(\boldsymbol{z}_*)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \eta(\boldsymbol{z}_*)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \eta(\boldsymbol{z}_*)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)].\end{aligned}$$

Second, we analyze the limit of each term. The limits of the twelve terms in the expectation are given in the claim below.

Claim 3.4.11.1. It holds that

$$\begin{aligned}
\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= \psi^{[i]}ck(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*});\\
\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}] &= 0;\\
\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= (\psi^{[i]})^2k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*});\\
\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0;\\
\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0;\\
\limsup_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &\leqslant \frac{\chi^{[i]}(\psi^{[i]})^2(\sigma_e^{[i]})^2}{\mu_\chi},\ \text{if } \boldsymbol{z}_*=\boldsymbol{z}^{[i]}_{agg*},\\
\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0,\ \text{otherwise};\\
\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0;\\
\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= (\psi^{[i]})^2(\sigma_e^{[i]})^2,\ \text{if } \boldsymbol{z}_*=\boldsymbol{z}^{[i]}_{agg*},\\
\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0,\ \text{otherwise};\\
\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= ck(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*});\\
\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0;\\
\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= \psi^{[i]}k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*});\\
\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= 0. \qquad \square
\end{aligned}$$

Finally, we find the upper bound of $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)]$. Plugging in the terms in Claim 3.4.11.1 gives

$$\begin{aligned}
&\text{when } \boldsymbol{z}_*=\boldsymbol{z}^{[i]}_{agg*},\ \limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)] \leqslant 2v^{[i]}_{\boldsymbol{z}_*}\Big((\psi^{[i]}-1)(c-\psi^{[i]})k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*})\\
&\qquad\qquad + (\mu_\chi^{-1}\chi^{[i]}-1)(\psi^{[i]})^2(\sigma_e^{[i]})^2\Big);\\
&\text{when } \boldsymbol{z}_*\neq\boldsymbol{z}^{[i]}_{agg*},\ \lim_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)] = 2v^{[i]}_{\boldsymbol{z}_*}\Big((\psi^{[i]}-1)(c-\psi^{[i]})k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*})\Big).
\end{aligned}$$

Invoking Lemma 3.4.7 gives $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,1}(t)] \leqslant 2v^{[i]}_{\boldsymbol{z}_*}\Big((\psi^{[i]}-1)(c-\psi^{[i]})k(\boldsymbol{z}_*,\boldsymbol{z}^{[i]}_{agg*})\Big)$, $\forall \boldsymbol{z}_*\in\boldsymbol{\mathcal{Z}}$. ■

Proof of Claim 3.4.11.1: We analyze the limit of each of the twelve terms in expectation as follows.

Term 1. The solution of the LTV system (3.21) gives

$$\theta^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t),\boldsymbol{r}}(t) = \sum_{j\in\mathcal{V}} \phi_{ij}(t,0)\theta^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(0) + \sum_{l=1}^{t}\sum_{j\in\mathcal{V}} \phi_{ij}(t,l)\Delta\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[j]}(t)}\check{r}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t),$$

where by Remark 3.4.8, $\check{r}^{[j]}_{\boldsymbol{z}_*}(t) = \psi^{[i]}_{\boldsymbol{z}_*}(t)\eta(\boldsymbol{z}^{[i]}_*(t))$, $\psi^{[i]}_{\boldsymbol{z}_*}(t) \triangleq \frac{\kappa(\rho^{\mathcal{Z}^{[i]}(t)}_{\boldsymbol{z}_*})}{\sigma_f^2+(\sigma_e^{[i]})^2}$, $\forall \boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$. Combining this with the definition of $\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)$ gives

$$\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) = c^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\eta(\boldsymbol{z}^{[i]}_{agg*,p}(t)),$$

where $\boldsymbol{z}^{[i]}_{agg*,p}(t) \in \text{proj}(\boldsymbol{z}^{[i]}_{agg*}(t), \mathcal{Z}^{[i]}(t))$,

$$\begin{aligned} c^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \triangleq &(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2\Big(\sum_{j\in\mathcal{V}} \phi_{ij}(t,0)\theta^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(0)/\eta(\boldsymbol{z}^{[i]}_{agg*,p}(t)) + \\ &\sum_{l=1}^{t}\sum_{j\in\mathcal{V}} \big(\phi_{ij}(t,l)\Delta\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[j]}(t)}\psi^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)\Big) \end{aligned}$$

Therefore, we have

$$\begin{aligned} \mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= \mathbb{E}[\psi^{[i]}_{\boldsymbol{z}_*}(t)\eta(\boldsymbol{z}^{[i]}_*(t))c^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\eta(\boldsymbol{z}^{[i]}_{agg*,p}(t))] \\ &= \psi^{[i]}_{\boldsymbol{z}_*}(t)c^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\mathbb{E}[\eta(\boldsymbol{z}^{[i]}_*)\eta(\boldsymbol{z}^{[i]}_{agg*,p}(t))] \\ &= \psi^{[i]}_{\boldsymbol{z}_*}(t)c^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)k(\boldsymbol{z}^{[i]}_*(t), \boldsymbol{z}^{[i]}_{agg*,p}(t)). \end{aligned}$$

where the last equality follows from Assumption 3.3.5. Note that Lemma 3.4.10 indicates $\forall \boldsymbol{z} \in \boldsymbol{\mathcal{Z}}$, $\lim_{t\to\infty}\psi^{[i]}_{\boldsymbol{z}}(t) = \psi^{[i]}$ and $\lim_{t\to\infty} c^{[i]}_{\boldsymbol{z}}(t) = c$. Hence

$$\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = \lim_{t\to\infty}\psi^{[i]}(t)c^{[i]}(t)k(\boldsymbol{z}^{[i]}_*(t), \boldsymbol{z}^{[i]}_{agg*,p}(t)) = \psi^{[i]}ck(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*}).$$

Terms 3, 9, 11. Similar to Term 1, we have

$$\lim_{t\to\infty}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = (\psi^{[i]})^2k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})$$

$$\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = ck(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})$$
$$\lim_{t\to\infty}\mathbb{E}[\eta(\boldsymbol{z}_*)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = \psi^{[i]}k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*}).$$

Term 2. By definitions, $\check{r}^{[i]}_{\boldsymbol{z}_*}(t)$ and $\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)$ only depend on $\eta(\boldsymbol{z}^{[i]}_*(t))$ and $e^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}$, respectively. Since $\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}$ is zero-mean, we have $\forall t \geqslant 1$,

$$\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}] = \mathbb{E}_{\eta}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)]\mathbb{E}_{e^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}}[\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = 0.$$

Terms 4, 5, 7, 10, 12. Similar to Term 2,

$$\begin{aligned}\mathbb{E}[\check{r}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= \mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] \\ &= \mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{r}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = \mathbb{E}[\eta(\boldsymbol{z}_*)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] \\ &= \mathbb{E}[\eta(\boldsymbol{z}_*)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = 0, \quad \forall t \geqslant 1.\end{aligned}$$

Terms 6, 8. Since $e^{[i]}_{\boldsymbol{z}_*}$ and $e^{[j]}_{\boldsymbol{z}_*}$ are independent zero-mean measurement noises, we have $\mathbb{E}[e^{[i]}_{\boldsymbol{z}_*}e^{[j]}_{\boldsymbol{z}_*}] = 0$. Since Remark 3.4.8 states that $\check{e}^{[i]}_{\boldsymbol{z}_*}(t) = \psi^{[i]}(t)e^{[i]}_{\boldsymbol{z}_*}$, we have $\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$ and $\check{e}^{[j]}_{\boldsymbol{z}_*}(t)$ are also zero-mean and independent. Therefore,

$$\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[j]}_{\boldsymbol{z}_*}(t)] = 0, \ \forall j \neq i.$$

Since $\hat{e}^{[j]}_{\boldsymbol{z}_*}(t)$ is linear in $\check{e}^{[j]}_{\boldsymbol{z}_*}(t)$, we have

$$\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\hat{e}^{[j]}_{\boldsymbol{z}_*}(t)] = 0, \forall j \neq i.$$

Recall that $\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) \triangleq (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t),\boldsymbol{e}}(t)$ and the LTV solution gives $\theta^{[i]}_{\boldsymbol{z},\boldsymbol{e}}(t) \triangleq \sum_{l=1}^{t}\sum_{j=1}^{n}\phi_{ij}(t,l)\Delta\hat{e}^{[j]}_{\boldsymbol{z}}(l)$. Then

$$\begin{aligned}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &= (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)(\sum_{l=1}^{t}\sum_{j=1}^{n}\phi_{ij}(t,l)\Delta\hat{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(l))] \\ &= (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)(\sum_{l=1}^{t}\phi_{ii}(t,l)\Delta\hat{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(l))]\end{aligned}$$

$$= (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 \phi_{ii}(t,t)\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\hat{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)]$$
$$= (\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2 \check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)} \phi_{ii}(t,t)\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] \quad (3.26)$$

where the third equality follows from the initial condition $\hat{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(0) = 0$ and $\phi_{ij}(t,l) = \phi_{ij}(t,l')$ for all $0 < l, l' \leqslant t$ implied by Assumption 3.3.6.

The independence of $e^{[i]}_{\boldsymbol{z}}$ over $\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}$ gives:

$$\text{if } \boldsymbol{z}^{[i]}_*(t) = \boldsymbol{z}^{[i]}_{agg*}(t),\ \mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = \mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}_*}(t)];$$
$$\text{otherwise, } \mathbb{E}[(\check{e}^{[i]}_{\boldsymbol{z}_*}(t))^2] = 0.$$

Notice $\lim_{t\to\infty} \boldsymbol{z}^{[i]}_*(t) = \boldsymbol{z}_*$ and $\lim_{t\to\infty} \boldsymbol{z}^{[i]}_{agg*,p}(t) = \boldsymbol{z}^{[i]}_{agg*}$. Hence

$$\text{if } \boldsymbol{z}_* = \boldsymbol{z}^{[i]}_{agg*},\ \lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = \mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}\check{e}^{[i]}_{\boldsymbol{z}_*}];$$
$$\text{otherwise, } \lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = 0.$$

The definition of $\check{e}^{[i]}_{\boldsymbol{z}_*}$ in Remark 3.4.8 gives

$$\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}\check{e}^{[i]}_{\boldsymbol{z}_*}] = (\psi^{[i]})^2\mathbb{E}[(e^{[i]}_{\boldsymbol{z}_*})^2] = (\psi^{[i]})^2(\sigma^{[i]}_e)^2.$$

Hence for Term 8, we have

$$\text{if } \boldsymbol{z}_* = \boldsymbol{z}^{[i]}_{agg*},\ \lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = (\psi^{[i]})^2(\sigma^{[i]}_e)^2;$$
$$\text{otherwise, } \lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = 0. \quad (3.27)$$

By Corollary 3.4.2,

$$\chi^{[i]} = \lim_{t\to\infty} \check{\sigma}^{-2}_{\boldsymbol{z}|\mathcal{D}^{[i]}(t)}, \quad \forall \boldsymbol{z} \in \boldsymbol{\mathcal{Z}},$$

and by Corollary 3.4.5,

$$\mu_\chi^{-1} = \lim_{t\to\infty}(\check{\sigma}^{(agg)}_{\boldsymbol{z}|\mathcal{D}(t)})^2 = \lim_{t\to\infty}(\hat{\sigma}^{[i]}_{\boldsymbol{z}|\mathcal{D}(t)})^2, \quad \forall \boldsymbol{z} \in \boldsymbol{\mathcal{Z}}.$$

Note that $\phi_{ii}(t,t) \leqslant 1$ implied by Assumption 3.3.6. Combining these with (3.26)

and (3.27) gives Term 6

$$\begin{aligned}\limsup_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] &\leqslant \lim_{t\to\infty}(\hat{\sigma}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}(t)})^2\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)]\\ &\leqslant \mu_\chi^{-1}\chi^{[i]}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}\check{e}^{[i]}_{\boldsymbol{z}_*}] = \mu_\chi^{-1}\chi^{[i]}(\psi^{[i]})^2(\sigma_e^{[i]})^2\end{aligned}$$

when $\boldsymbol{z}_* = \boldsymbol{z}^{[i]}_{agg*}$; otherwise $\lim_{t\to\infty}\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)] = 0$. □

Lemma 3.4.12 shows the limiting behavior of $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,2}(t)]$.

Lemma 3.4.12. Suppose the same conditions in Theorem 3.3.8 part II hold and $d^{[j]}(t) \to 0$, $\forall j \in \mathcal{V}$. If $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ for some $i \in \mathcal{V}$, then $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*,2}(t)] = (v^{[i]}_{\boldsymbol{z}_*})^2 q^{[i]}_{\boldsymbol{z}_*,2}$.

Proof: The proof is done by combining (3.25) with the definition of $s^{[i]}_{\boldsymbol{z}_*,2}(t)$ and applying similar term-by-term analysis as in Lemma 3.4.11. ■

Proposition 3.4.13 shows the limiting behavior of $\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*}(t)]$ when $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$.

Proposition 3.4.13. Suppose the same conditions in Theorem 3.3.8 Part II hold and $d^{[j]}(t) \to 0$, $\forall j \in \mathcal{V}$. If $\lim_{t\to\infty}\mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$ for some $i \in \mathcal{V}$, then $\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*}(t)] \leqslant -\mathcal{O}\Big(k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})\Big) < 0$.

Proof: Denote $b^{[i]}_{\boldsymbol{z}_*} \triangleq -2q^{[i]}_{\boldsymbol{z}_*,1}/q^{[i]}_{\boldsymbol{z}_*,2}$. Then Lemma 3.4.11 and 3.4.12 imply

$$\limsup_{t\to\infty}\mathbb{E}[s^{[i]}_{\boldsymbol{z}_*}(t)] \leqslant v^{[i]}_{\boldsymbol{z}_*}q^{[i]}_{\boldsymbol{z}_*,2}(-b^{[i]}_{\boldsymbol{z}_*} + v^{[i]}_{\boldsymbol{z}_*}). \tag{3.28}$$

We first show that $0 < v^{[i]}_{\boldsymbol{z}_*} < b^{[i]}_{\boldsymbol{z}_*}$. The definition of $v^{[i]}_{\boldsymbol{z}_*}(t)$ on Line 9 in fused GPR gives

$$v^{[i]}_{\boldsymbol{z}_*} = \lim_{t\to\infty}\{g^{[i]}_{\boldsymbol{z}_*}(t)\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}|\mathcal{D}^{[i]}(t)}\}. \tag{3.29}$$

Corollary 3.4.2 renders $\lim_{t\to\infty}\check{\sigma}^{-2}_{\boldsymbol{z}^{[i]}_{agg*}|\mathcal{D}^{[i]}(t)} > 0$ and $\lim_{t\to\infty}g^{[i]}_{\boldsymbol{z}_*}(t) > 0$; hence $v^{[i]}_{\boldsymbol{z}_*} > 0$. This also indicates that

$$v^{[i]}_{\boldsymbol{z}_*} = \frac{\lim_{t\to\infty}g^{[i]}_{\boldsymbol{z}_*}(t)}{\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}}. \tag{3.30}$$

By boundedness in Assumption 3.3.1 and Corollary 3.4.2, Line 8 in fused GPR renders

$$\lim_{t\to\infty} g^{[i]}_{\boldsymbol{z}_*}(t) = (1-\psi^{[i]})(c-\psi^{[i]})k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})\sigma_f^{-2} = -q^{[i]}_{\boldsymbol{z}_*,1}/\sigma_f^2. \tag{3.31}$$

The following claim characterizes the lower bound of $\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$ in terms of $q^{[i]}_{\boldsymbol{z}_*,2}$.

Claim 3.4.13.1. It holds that $\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)} > \frac{q^{[i]}_{\boldsymbol{z}_*,2}}{2(c\sigma_f^2-\psi^{[i]})} > 0.$ □

Combining (3.31) and Claim 3.4.13.1 with (3.30) gives

$$\begin{aligned} v^{[i]}_{\boldsymbol{z}_*} &= \frac{\lim_{t\to\infty} g^{[i]}_{\boldsymbol{z}_*}(t)}{\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}} = \frac{-q^{[i]}_{\boldsymbol{z}_*,1}/\sigma_f^2}{\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}} \\ &< \frac{-q^{[i]}_{\boldsymbol{z}_*,1}/\sigma_f^2}{q^{[i]}_{\boldsymbol{z}_*,2}/2(c\sigma_f^2-\psi^{[i]})} < \frac{-q^{[i]}_{\boldsymbol{z}_*,1}/\sigma_f^2}{q^{[i]}_{\boldsymbol{z}_*,2}/2\sigma_f^2} = b^{[i]}_{\boldsymbol{z}_*}, \end{aligned}$$

noticing that $0 < c < 1$.

Notice that Lemma 3.4.7 implies $q^{[i]}_{\boldsymbol{z}_*,2} > 0$. Since $0 < v^{[i]}_{\boldsymbol{z}_*} < b^{[i]}_{\boldsymbol{z}_*}$, (3.28) implies $\limsup_{t\to\infty} \mathbb{E}[s^{[i]}_{\boldsymbol{z}_*}(t)] < 0$. Combining (3.29) and (3.31) renders $v^{[i]}_{\boldsymbol{z}_*} = \mathcal{O}\Big(k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})\Big)$. Combining this with (3.28) renders $\limsup_{t\to\infty} \mathbb{E}[s^{[i]}_{\boldsymbol{z}_*}(t)] \leqslant -\mathcal{O}\Big(k(\boldsymbol{z}_*, \boldsymbol{z}^{[i]}_{agg*})\Big)$. ■

Proof of Claim 3.4.13.1: Outline: Based on the definition of $q^{[i]}_{\boldsymbol{z}_*,2}$, the proof is broken down into two parts: deriving the upper bound of $(c-\psi^{[i]})\sigma_f^2$ and the upper bound of $\limsup_{t\to\infty} \mathbb{E}[\big(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2]$.

First, we derive the upper bound of $(c-\psi^{[i]})\sigma_f^2$. Since $\psi^{[j]} < 1$, $\forall j \in \mathcal{V}$ and c is a convex combination of $\psi^{[j]}$, we have $c < 1$ and

$$(c-\psi^{[i]})\sigma_f^2 < (1-\psi^{[i]})\sigma_f^2 = \sigma_f^2 - \frac{\sigma_f^4}{\sigma_f^2 + (\sigma_e^{[i]})^2} = \lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}, \tag{3.32}$$

where the equality follows from Corollary 3.4.2.

Second, we derive the upper bound of $\limsup_{t\to\infty} \mathbb{E}[\big(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2]$. Consider the following properties regarding the covariances involving $\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t)$ and $\check{e}^{[j]}_{\boldsymbol{z}_*}(t)$, where the proofs are at the end.

Claim 3.4.13.2. It holds that $\mathrm{cov}\big(\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t), \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\big) = \mathbb{E}[\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}_*}(t)] \geqslant 0$, $\forall t \geqslant 1$, $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$, $i, j \in \mathcal{V}$. □

Claim 3.4.13.3. It holds that $\mathrm{cov}(\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t)+\check{e}^{[i]}_{\boldsymbol{z}_*}(t), \tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)+\check{e}^{[j]}_{\boldsymbol{z}_*}(t)) \geqslant 0$, $\forall t \geqslant 1$, $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$, $i, j \in \mathcal{V}$. □

Since $\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)$ and $\check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)$ are zero-mean, we have

$$\mathbb{E}[\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2] = \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big).$$

By Claim 3.4.13.2, we have

$$\begin{aligned}
&\mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) \\
&= \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) - 2\mathrm{cov}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t), \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) + \mathrm{var}\big(\check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) \\
&\leqslant \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) \leqslant \sum_{j=1}^{n} \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big).
\end{aligned}$$

By Claim 3.4.13.3, we have

$$\begin{aligned}
&\sum_{j=1}^{n} \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) \\
&\leqslant \sum_{j=1}^{n} \mathrm{var}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) + \sum_{j=1}^{n}\sum_{l\neq j} \mathrm{cov}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t), \\
&\quad \tilde{e}^{[l]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[l]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) \\
&= \mathrm{var}\Big(\sum_{j=1}^{n} \big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)\Big).
\end{aligned}$$

The above three statements render

$$\mathbb{E}[\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) - \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2] \leqslant \mathrm{var}\Big(\sum_{j=1}^{n} \big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t) + \check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)\Big). \tag{3.33}$$

By Lemma 3.4.10, we have $\lim_{t\to\infty} \sum_{j=1}^{n} \big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)+\check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big) = \sum_{j=1}^{n} \Big(\mu_\chi^{-1}\chi^{[j]}\psi^{[j]}+$

$\psi^{[j]}\Big)e^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}}$. Taking limit on both sides of (3.33) renders

$$\begin{aligned}
&\limsup_{t\to\infty}\mathbb{E}[\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)-\check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2]\leqslant\lim_{t\to\infty}\mathrm{var}(\sum_{j=1}^{n}\big(\tilde{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)+\check{e}^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big))\\
&=\mathrm{var}(\sum_{j=1}^{n}\Big(\mu_\chi^{-1}\chi^{[j]}\psi^{[j]}+\psi^{[j]}\Big)e^{[j]}_{\boldsymbol{z}^{[i]}_{agg*}}=\sum_{j=1}^{n}\Big((\mu_\chi^{-1}\chi^{[j]}+1)^2(\psi^{[j]})^2(\sigma_e^{[j]})^2\Big)\\
&\leqslant\sum_{j=1}^{n}\Big((\mu_\chi^{-1}\chi^{[j]}+1)^2\Big)(\psi^{\max})^2(\sigma_e^{\max})^2. \qquad (3.34)
\end{aligned}$$

Note that $\sum_{j=1}^{n}(\mu_\chi^{-1}\chi^{[j]}+1)^2=\frac{\sigma_\chi^2}{\mu_\chi^2}$ based on the definitions in Section 3.3.4. Lemma 3.4.7 indicates $c-\psi^{[i]}>0$. Since $\sigma_f^2\geqslant 1$ in Section 3.3.4, we have $c\sigma_f^2-\psi^{[i]}>0$. By definition of ϵ_+ in Section 3.3.4, we further have $c\sigma_f^2-\psi^{[i]}\geqslant\epsilon_+$. Combining this with (3.34) gives

$$\begin{aligned}
\limsup_{t\to\infty}\mathbb{E}[\big(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)-\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2]&\leqslant\frac{\sigma_\chi^2(\psi^{\max})^2(\sigma_e^{\max})^2(c\sigma_f^2-\psi^{[i]})}{\mu_\chi^2\epsilon_+}\\
&\leqslant(\psi^{\max})^2(\sigma_e^{\min})^2(c\sigma_f^2-\psi^{[i]}),
\end{aligned}$$

where the last inequality follows from (3.4). Notice that $\psi^{\max}=\frac{\sigma_f^2}{\sigma_f^2+(\sigma_e^{[\min]})^2}<1$ and hence

$$\begin{aligned}
(\psi^{\max})^2(\sigma_e^{\min})^2&<\frac{\sigma_f^2(\sigma_e^{[\min]})^2}{\sigma_f^2+(\sigma_e^{[\min]})^2}=\sigma_f^2-\frac{\sigma_f^4}{\sigma_f^2+(\sigma_e^{[\min]})^2}\\
&\leqslant\sigma_f^2-\frac{\sigma_f^4}{\sigma_f^2+(\sigma_e^{[i]})^2}=\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}
\end{aligned}$$

where the last equality follows from Corollary 3.4.2. Therefore,

$$\limsup_{t\to\infty}\mathbb{E}[\big(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)-\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}(t)}(t)\big)^2]\leqslant\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}(c\sigma_f^2-\psi^{[i]}). \qquad (3.35)$$

Finally, we find the lower bound of $\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)}$. Since $\sigma_f^2\geqslant 1$, $c\sigma_f^2>c$. Combining this with (3.32) and (3.35) renders

$$q^{[i]}_{\boldsymbol{z}_*,2}=(c-\psi^{[i]})^2\sigma_f^2+\limsup_{t\to\infty}\mathbb{E}[(\tilde{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}}(t)-\check{e}^{[i]}_{\boldsymbol{z}^{[i]}_{agg*}}(t))^2]<2(c\sigma_f^2-\psi^{[i]})\lim_{t\to\infty}\check{\sigma}^2_{\boldsymbol{z}^{[i]}_{agg*}(t)|\mathcal{D}^{[i]}(t)},$$

and obviously $q^{[i]}_{\boldsymbol{z}_*,2} > 0$. $\square$

Proof of Claim 3.4.13.2: Since $\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)$ and $\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$ are zero-mean, it follows that $\mathrm{cov}(\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t), \check{e}^{[i]}_{\boldsymbol{z}_*}(t)) = \mathbb{E}[\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)\check{e}^{[i]}_{\boldsymbol{z}_*}(t)]$.

Recall that $\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t) \triangleq (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t)$ and the LTV solution (3.21) gives

$$\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{l=1}^{t}\sum_{j=1}^{n}\phi_{ij}(t,l)\Delta\hat{e}^{[j]}_{\boldsymbol{z}_*}(l).$$

By Assumption 3.3.6, $\phi_{ij}(t,l) = \phi_{ij}(t,l')$ for all $0 < l, l' \leqslant t$. Therefore,

$$\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{l=1}^{t}\sum_{j=1}^{n}\phi_{ij}(t,t)\Delta\hat{e}^{[j]}_{\boldsymbol{z}_*}(l).$$

Because of the initial condition $\hat{e}^{[j]}_{\boldsymbol{z}_*}(0) = 0$, we have $\theta^{[i]}_{\boldsymbol{z}_*,\boldsymbol{e}}(t) = \sum_{j=1}^{n}\phi_{ij}(t,t)\hat{e}^{[j]}_{\boldsymbol{z}_*}(t)$. Then

$$\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)] = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)(\sum_{j=1}^{n}\phi_{ij}(t,t)\hat{e}^{[j]}_{\boldsymbol{z}_*}(t))].$$

Since $e^{[i]}_{\boldsymbol{z}_*}$ and $e^{[j]}_{\boldsymbol{z}_*}$ are zero-mean and independent if $i \neq j$, we have $\mathbb{E}[e^{[i]}_{\boldsymbol{z}_*}e^{[j]}_{\boldsymbol{z}_*}] = 0$. Since Remark 3.4.8 indicates that $\check{e}^{[i]}_{\boldsymbol{z}_*}(t) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}e^{[i]}_{\boldsymbol{z}_*}$, we have $\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$ and $\check{e}^{[j]}_{\boldsymbol{z}_*}(t)$ are also zero-mean and independent. Therefore, $\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\check{e}^{[j]}_{\boldsymbol{z}_*}(t)] = 0$, $\forall j \neq i$. Since $\hat{e}^{[i]}_{\boldsymbol{z}_*}(t)$ is linear in $\check{e}^{[i]}_{\boldsymbol{z}_*}(t)$, we have $\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\hat{e}^{[j]}_{\boldsymbol{z}_*}(t)] = 0$, $j \neq i$. Hence, we further have

$$\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)] = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2\check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}\phi_{ii}(t,t)\mathbb{E}[(\check{e}^{[i]}_{\boldsymbol{z}_*}(t))^2].$$

Since $\phi_{ii}(t,t) \geqslant 0$ implied by Assumption 3.2.3, we have $\mathbb{E}[\check{e}^{[i]}_{\boldsymbol{z}_*}(t)\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t)] \geqslant 0$. $\square$

Proof of Claim 3.4.13.3: Recall that $\tilde{e}^{[p]}_{\boldsymbol{z}_*}(t)$ and $\check{e}^{[p]}_{\boldsymbol{z}_*}(t)$ are zero mean for all $p \in \mathcal{V}$, hence

$$\mathrm{cov}(\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t), \tilde{e}^{[j]}_{\boldsymbol{z}_*}(t) + \check{e}^{[j]}_{\boldsymbol{z}_*}(t)) = \mathbb{E}[\big(\tilde{e}^{[i]}_{\boldsymbol{z}_*}(t) + \check{e}^{[i]}_{\boldsymbol{z}_*}(t)\big)\big(\tilde{e}^{[j]}_{\boldsymbol{z}_*}(t) + \check{e}^{[j]}_{\boldsymbol{z}_*}(t)\big)].$$

Recall that $\hat{e}^{[p]}_{\boldsymbol{z}_*}(l) = \check{\sigma}^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[p]}(l)}\check{e}^{[p]}_{\boldsymbol{z}_*}(l)$, $\check{e}^{[p]}_{\boldsymbol{z}_*}(l) = \frac{\kappa(\rho^{\mathcal{Z}^{[p]}(l)}_{\boldsymbol{z}_*})}{\sigma_f^2+(\sigma^{[i]}_e)^2}e^{[p]}_{\boldsymbol{z}^{[p]}_*(l)}$. By independence of

$e_{\boldsymbol{z}}^{[p]}$ over $p \in \mathcal{V}$, we have $\mathbb{E}[e_{\boldsymbol{z}}^{[p]} e_{\boldsymbol{z}}^{[p']}] = 0$ and hence $\mathbb{E}[\check{e}_{\boldsymbol{z}}^{[p]} \check{e}_{\boldsymbol{z}}^{[p']}] = 0$ if $p \neq p'$, and then

$$\begin{aligned}
&\operatorname{cov}(\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t) + \check{e}_{\boldsymbol{z}_*}^{[i]}(t), \tilde{e}_{\boldsymbol{z}_*}^{[j]}(t) + \check{e}_{\boldsymbol{z}_*}^{[j]}(t)) \\
&\geqslant \mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)] + \mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\check{e}_{\boldsymbol{z}_*}^{[j]}(t)] + \mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)\check{e}_{\boldsymbol{z}_*}^{[i]}(t)] \geqslant \mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)], \qquad (3.36)
\end{aligned}$$

where the last inequality follows from Claim 3.4.13.2. Obviously, $\mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)] \geqslant 0$ if $i = j$. Next, we consider $i \neq j$.

By definition of $\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)$ and the LTV solution (3.21) of $\theta_{\boldsymbol{z}_*,\boldsymbol{e}}^{[i]}(t)$, we can write

$$\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t) = (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg*}^{[i]}(t)|\mathcal{D}(t)})^2 \sum_{l=1}^{t}\sum_{p=1}^{n} \phi_{ip}(t,l)\Delta\hat{e}_{\boldsymbol{z}_*}^{[p]}(l).$$

Assumption 3.3.6 implies $\phi_{ip}(t,l) = \phi_{ip}(t,l')\ \forall l, l' \in [1,t],\ t \geqslant 1$. Therefore, we can denote $\tilde{\phi}_{ip}(t) \triangleq (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg*}^{[i]}(t)|\mathcal{D}(t)})^2\phi_{ip}(t,l),\ \forall l \in [1,t]$. Due to the initial condition $\hat{e}_{\boldsymbol{z}_*}^{[j]}(0) = 0$, we can further write $\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t) = \sum_{p=1}^{n} \tilde{\phi}_{ip}(t)\hat{e}_{\boldsymbol{z}_*}^{[p]}(t)$. It gives

$$\mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)] = \mathbb{E}[\sum_{p=1}^{n}\big(\tilde{\phi}_{ip}(t)\hat{e}_{\boldsymbol{z}_*}^{[p]}(t)\big)\sum_{p'=1}^{n}\big(\tilde{\phi}_{jp'}(t)\hat{e}_{\boldsymbol{z}_*}^{[p']}(t)\big)],$$

Since $\mathbb{E}[e_{\boldsymbol{z}}^{[p]} e_{\boldsymbol{z}}^{[p']}] = 0$ if $p \neq p'$ (the independence of $e_{\boldsymbol{z}}^{[p]}$ over $p \in \mathcal{V}$) and $\hat{e}_{\boldsymbol{z}}^{[p]}(t)$ is linear in $e_{\boldsymbol{z}}^{[p]}$, we have $\mathbb{E}[\hat{e}_{\boldsymbol{z}}^{[p]}(t)\hat{e}_{\boldsymbol{z}}^{[p']}(t)] = 0$ if $p \neq p'$. This gives

$$\mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)] = \mathbb{E}[\sum_{p=1}^{n}\tilde{\phi}_{ip}(t)\tilde{\phi}_{jp}(t)\big(\hat{e}_{\boldsymbol{z}_*}^{[p]}(t)\big)^2].$$

Assumption 3.2.3 implies $\phi_{ip}(t,l) \geqslant 0,\ \forall l \in [1,t],\ i \in \mathcal{V}$. Hence $\tilde{\phi}_{ip}(t)\tilde{\phi}_{jp}(t) \geqslant 0$. Therefore, $\mathbb{E}[\tilde{e}_{\boldsymbol{z}_*}^{[i]}(t)\tilde{e}_{\boldsymbol{z}_*}^{[j]}(t)] \geqslant 0$. Combining this with (3.36) finishes the proof. □

Lemma 3.4.14 shows a sufficient condition for $\lim_{t\to\infty} \mathcal{Z}_{agg}^{[i]}(t) \neq \emptyset$.

Lemma 3.4.14. Suppose the same conditions for Corollary 3.4.5 hold and $d^{[j]}(t) \to 0$ for all $j \in \mathcal{V}$. If $(\sigma_e^{[i]})^2 > \frac{1}{n}\sum_{j=1}^{n}(\sigma_e^{[j]})^2$ for some $i \in \mathcal{V}$, then $\lim_{t\to\infty} \mathcal{Z}_{agg}^{[i]}(t) = \mathcal{Z}_{agg}$.

Proof: Since function f_2 in Lemma 3.4.1 is strictly increasing, we have

$$f_2((\sigma_e^{[i]})^2) > f_2(\frac{1}{n}\sum_{j=1}^{n}(\sigma_e^{[j]})^2) \geqslant \frac{1}{n}\sum_{j=1}^{n} f_2((\sigma_e^{[j]})^2). \qquad (3.37)$$

By Corollary 3.4.2, $f_2((\sigma_e^{[i]})^2) = \lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)}$ for any $\boldsymbol{z}_{agg} \in \mathcal{Z}_{agg}$, and by Corollary 3.1 in [78],

$$\lim_{t\to\infty} (\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2 = \lim_{t\to\infty} \lambda^{[i]}_{\boldsymbol{z}_{agg}}(t) = \lim_{t\to\infty} \frac{1}{n} \sum_{j=1}^{n} \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[j]}(t)} = \frac{1}{n} \sum_{j=1}^{n} f_2((\sigma_e^{[j]})^2)$$

Combining these two statements with (3.37) gives

$$\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} > \lim_{t\to\infty} (\hat{\sigma}^{ave,[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2. \tag{3.38}$$

Taking the inverse of (3.37) gives

$$\big(f_2((\sigma_e^{[i]})^2)\big)^{-1} < \big(\frac{1}{n} \sum_{j=1}^{n} f_2((\sigma_e^{[j]})^2)\big)^{-1} \leqslant \frac{1}{n} \sum_{j=1}^{n} \big(f_2((\sigma_e^{[j]})^2)\big)^{-1},$$

where the last inequality follows from Lemma 3.4.1. This gives

$$\lim_{t\to\infty} \check{\sigma}^{-2}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} < \frac{1}{n} \sum_{j=1}^{n} \lim_{t\to\infty} \check{\sigma}^{-2}_{\boldsymbol{z}_{agg}|\mathcal{D}^{[j]}(t)} = \lim_{t\to\infty} (\check{\sigma}^{(agg)}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^{-2}$$

where the equality follows from (3.3). This is equivalent to

$$\lim_{t\to\infty} \check{\sigma}^2_{\boldsymbol{z}_{agg}|\mathcal{D}^{[i]}(t)} > \lim_{t\to\infty} (\check{\sigma}^{(agg)}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2 = \lim_{t\to\infty} (\hat{\sigma}^{[i]}_{\boldsymbol{z}_{agg}|\mathcal{D}(t)})^2,$$

where the equality follows from Corollary 3.4.5. Given (3.38) and the inequality above, since $\boldsymbol{z}_{agg} \in \mathcal{Z}_{agg}$ is arbitrary, we have $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) = \mathcal{Z}_{agg}$. ■

We now proceed to finish the proof of Part II of Theorem 3.3.8.

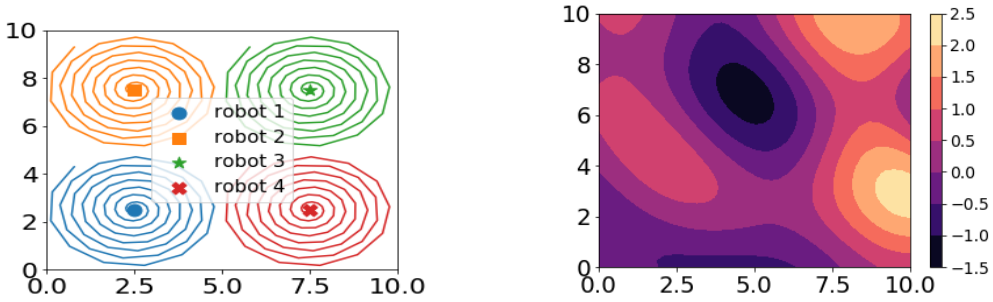
Proof of Theorem 3.3.8 Part II: By Line 3-4 in fused GPR, it is obvious that if $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) = \emptyset$, then $\lim_{t\to\infty} (\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)} - \eta(\boldsymbol{z}_*))^2 = \lim_{t\to\infty} (\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))^2$. Proposition 3.4.13 presents the case of $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$. Lemma 3.4.14 corresponds to the sufficient condition for Proposition 3.4.13, which is the sufficient condition for $\lim_{t\to\infty} \mathcal{Z}^{[i]}_{agg}(t) \neq \emptyset$. ■

3.5 Simulation

In this section, we conduct Monte Carlo simulation to evaluate the developed algorithm. For the algorithms introduced below, we use (NN) to denote the version of the algorithm related to Nearest-neighbor GPR and (full) to denote the version related to full GPR. We compare LiDGPR (NN), i.e., Algorithm 1, with five benchmarks: (i) agent-based GPR (NN), i.e., Nearest-neighbor GPR (Algorithm 2); (ii) agent-based GPR (full), i.e., Algorithm 2 is replaced by (2.1) and hence $\check{\boldsymbol{\mu}}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)} = \boldsymbol{\mu}_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)}, \check{\boldsymbol{\sigma}}^2_{\mathcal{Z}_*|\mathcal{D}^{[i]}(t)} = [\Sigma_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}]_{\boldsymbol{z}_*\in\mathcal{Z}_*}$; (iii) LiDGPR (full), i.e., Algorithm 1 with Algorithm 2 replaced by agent-based GPR (full); (iv) centralized Nearest-neighbor GPR (cNN-GPR, the centralized counterpart of LiDGPR (NN)), i.e., Nearest-neighbor GPR using all the data collected by all the agents; (v) centralized full GPR, i.e., (2.1) using all the data collected by all the agents. The simulations are run in Python, Linux Ubuntu 18.04 on an Intel Xeon(R) Silver 4112 CPU, 2.60 GHz with 32 GB of RAM.

Consider the scenario where four mobile robots are wandering in $\boldsymbol{\mathcal{Z}} \triangleq [1, 10] \times [1, 10]$ and learning spatial signals, such as temperature or wind fields. Specifically, the robots are learning 10 different signals in the form $\eta(\boldsymbol{z}) = \beta \sum_{m=1}^{10} \alpha_m \sin(w_{m,1} z_1 + w_{m,2} z_2)$, where $\alpha_m \sim \mathcal{N}(0, 0.01)$, $w_{m,1} \sim \mathcal{N}(0, 1)$, $w_{m,2} \sim \mathcal{N}(0, 1)$, β is chosen such that $SNR \triangleq \frac{\int \eta(\boldsymbol{z})^2 d\boldsymbol{z}}{\sigma_e^2} = 2$. A realization of η is shown in Figure 3.2b. For each signal, the robots repeat the trajectories for 10 times, and the observations along each trajectory are subject to a different noise, where the variances of the observation noises follow $(\sigma_e^{[i]})^2 = \sigma_e^2 \sim \mathcal{U}(0, 0.25)$ for all $i \in \mathcal{V}$. Notice that there are totally 100 simulations.

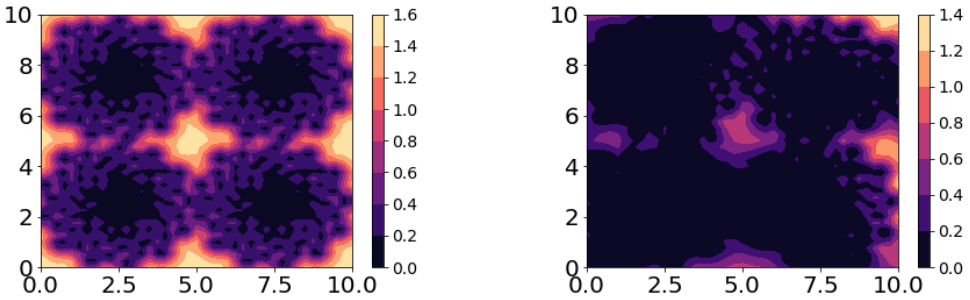
The communication graph of the robots is characterized by adjacency matrix $A(t) = \frac{1-(-1)^t}{2} \begin{bmatrix} 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} + \frac{1+(-1)^t}{2} \begin{bmatrix} 0.5 & 0.25 & 0 & 0.25 \\ 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0.25 & 0 & 0.25 & 0.5 \end{bmatrix}$, which satisfies Assumption 3.2.1, 3.2.2 and 3.2.3. As shown in Figure 3.2a, the robots have spiral trajectories generated by dynamics $\begin{bmatrix} z_1^{[i]}(t) \\ z_2^{[i]}(t) \end{bmatrix} = \begin{bmatrix} z_1^{[i]}(t-1) \\ z_2^{[i]}(t-1) \end{bmatrix} + 0.05t \begin{bmatrix} \sin(0.5t) \\ \cos(0.5t) \end{bmatrix}$, where the initial states of the robots are $(2.5, 2.5)$, $(2.5, 7.5)$, $(7.5, 7.5)$ and $(7.5, 2.5)$ respectively. Each robot i collects training data along its trajectory, i.e., $\big(\boldsymbol{z}^{[i]}(t), \eta(\boldsymbol{z}^{[i]}(t)) +$



(a) The trajectories of the robots (b) Ground truth of η

Figure 3.2: Robot trajectories and ground truth of η

$e^{[i]}(t)\big)$, $e^{[i]}(t) \sim \mathcal{N}(0, (\sigma_e^{[i]})^2)$, $t \geqslant 1$. The set $\mathcal{Z}_*$ of test points are uniformly separated over $\boldsymbol{\mathcal{Z}}$, and $|\mathcal{Z}_*| = 1600$. We use 25% of the test points for the set $\mathcal{Z}_{agg}$, i.e., $|\mathcal{Z}_{agg}| = 400$. The points in $\mathcal{Z}_{agg}$ are uniformly separated. The kernel is $k(\boldsymbol{z}, \boldsymbol{z}') = \sigma_f^2 \exp(-2\|\boldsymbol{z} - \boldsymbol{z}'\|^2)$, where σ_f^2 is chosen following the procedure under Remark 3.3.2. The resulting σ_f^2 ranges from 1.1 to 5.8 for each experiment in the Monte Carlo simulation. The prior mean is $\mu(\boldsymbol{z}) = 0$ for all $\boldsymbol{z} \in \mathcal{Z}$.



(a) Predictive variance (b) Predictive error

Figure 3.3: Predictive variance and error of cNN-GPR

The performances of the robots are similar, and we present the figures for robot 1 due to space limitation. Let the predictive error at $\boldsymbol{z}_* \in \mathcal{Z}_*$ be the distance between predictive mean and the ground truth of η at $\boldsymbol{z}_*$, where the distance adopts 2-norm. For example, the predictive error at $\boldsymbol{z}_*$ of agent-based GPR (NN) is $(\breve{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - \eta(\boldsymbol{z}_*))^2$. When robots' trajectories and η are those in Figure 3.2, Figure 3.3 shows the predictive variance and predictive error over $\mathcal{Z}_*$ of cNN-GPR. We can see that the predictive variances and errors are smaller near the trajectories

of the robots.

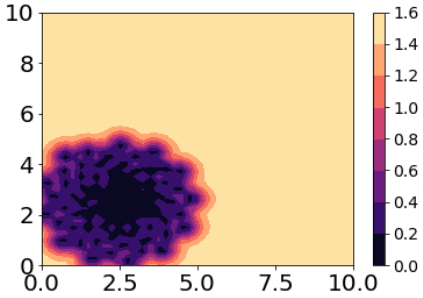
Figure 3.4 shows the predictive variances and predictive errors of agent-based GPR (NN) and LiDGPR (NN) over $\mathcal{Z}_*$ of robot 1. We can see that by only communicating a portion of the testing sets, LiDGPR (NN) improves the learning performances over agent-based GPR (NN) with reduced predictive variances and errors. The red dots in Figures 3.4c and 3.4d are the points of $\mathcal{Z}_{agg}$, and the "holes" indicate that the improvements take place around the trajectories (training data) of the other robots, which corresponds to the term $\kappa(\rho^{\mathcal{Z}^{[j]}(t)}_{\boldsymbol{z}^{[i]}_{agg*}(t)})^2$ in Part II of Theorem 3.3.3. In addition, the improvements reduce as the test points are moving away from $\mathcal{Z}_{agg}$, which corresponds to the terms $\kappa(\rho^{\boldsymbol{z}^{[i]}_{agg*}(t)}_{\boldsymbol{z}_*})^2$ in Part II of Theorem 3.3.3 and $\kappa(\rho^{\mathcal{Z}_{agg}}_{\boldsymbol{z}_*})$ in Part II of Theorem 3.3.8 respectively.

Figures 3.5a compares the average predictive errors and variances of LiDGPR (NN) with the five benchmarks. The x-axis is the iteration number, corresponding to the size of training data. The predictive variance and error at each iteration are represented by the corresponding averages over $\mathcal{Z}_*$.

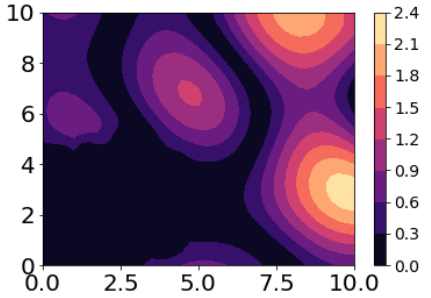
Note that the complexities in computation and memory are respectively $\mathcal{O}(nt)$ and $\mathcal{O}(nt)$ for cNN-GPR, and $\mathcal{O}((nt)^3)$ and $\mathcal{O}((nt)^2)$ for centralized full GPR. Notice that the differences in predictive variances and errors between cNN-GPR and centralized full GPR are small, while the diminishing rates are comparable. This shows that cNN-GPR has small performance loss compared to the benefit in reducing the complexities in computation and memory.

Comparing the curves of LiDGPR (NN) with agent-based GPR (NN) and agent-based GPR (full), we can see that LiDGPR (NN) not only compensates the information loss of using agent-based GPR (NN) to approximate agent-based GPR (full), but also gains extra information from the other robots.

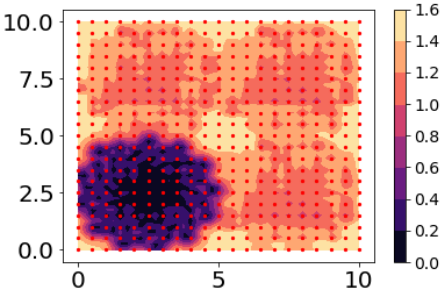
Figures 3.5a plots the theoretic error bounds in Part I of Theorems 3.3.3 and 3.3.8 over the whole Monte Carlo simulation. By multiplying by a constant, we scale down the bound by factor 0.023 in Part I of Theorem 3.3.8 for better visual comparison. The orders of rates of the bounds remain the same regardless of the scaling. Comparisons between the theoretic improvement and the actual improvement of LiDGPR (NN) over agent-based GPR (NN) are shown in Figures 3.5b and 3.5c. Since the theoretic bounds are not tight, to make a meaningful comparison, we scale up the bounds by factor 10 in Part II of Theorem IV.3 and IV.8.



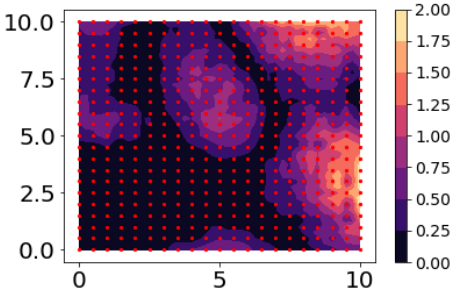
(a) Variance of agent-based GPR (NN)



(b) Predictive error of agent-based GPR (NN)

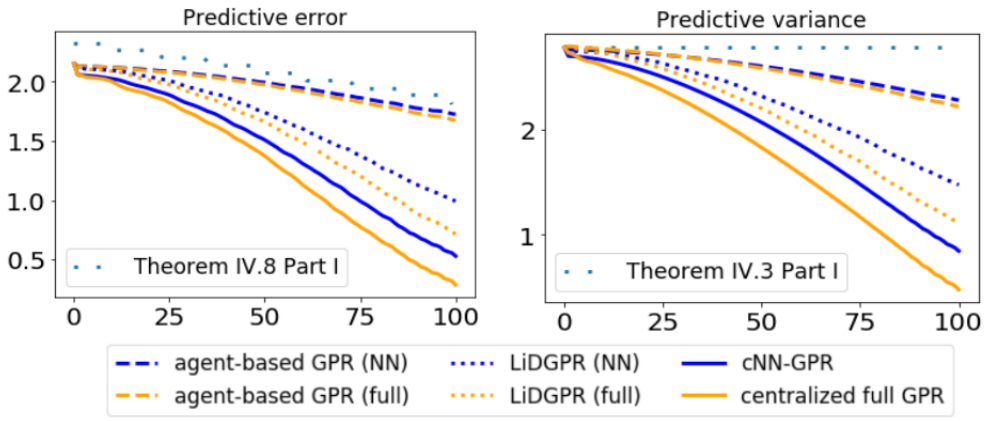


(c) Variance of LiDGPR (NN)

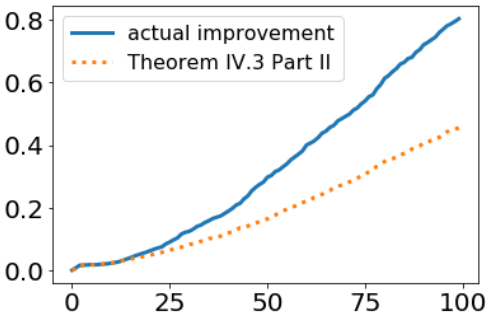


(d) Predictive error of LiDGPR (NN)

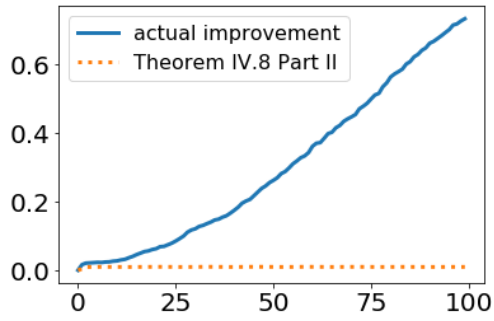
Figure 3.4: Comparison of agent-based GPR (NN) and LiDGPR (NN)



(a) Comparison in predictive errors and variances (The upper bound in Theorem 3.3.8 Part 1 is scaled by 0.023)



(b) $\check{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - (\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ vs. Theorem 3.3.3 Part II (scaled by 10)



(c) $\check{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} - (\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$ vs. Theorem 3.3.8 Part II (scaled by 10)

Figure 3.5: Average performance of robot 1 versus iteration number

The wall clock time for prediction using LiDGPR (NN) versus t, the number of local data points, after linear least-square fitting, has a slope 2.13e-6 second per test point per data point and a bias 1.19e-3 second per test point. Recall that Section 5.3.3 indicates that agent-based GPR (NN) has complexity $\mathcal{O}(t)$ and agent-based GPR (full) has complexity $\mathcal{O}(t^3)$. The growth rate of the computation times (milliseconds) of LiDGPR (NN) and LiDGPR (full) in the simulation are respectively $33.2t+200$ and $0.256t^3-0.1t^2-0.512t+27.6$. Over the simulation, the average is 1000 test-point predictions/second, or 1 kHz, with standard deviation 153 predictions/second.

3.6 Conclusion

We propose the algorithm LiDGPR which allows a group of agents to collaboratively learn a common static latent function through streaming data. The algorithm is cognizant of agents' limited resources in communication, computation and memory. We analyze the transient and steady-state behaviors of the algorithm and quantify the improvement brought by inter-agent communication. Simulations are conducted to confirm the theoretical findings.

Chapter 4

Distributed safe learning and planning

4.1 Introduction

Chapter 3 propose a class of distributed GPR algorithm for multi-robot systems. In this chapter, we introduce a framework that merges GPR with control to ensure physical safety of multi-robot systems. In particular, we consider the problem of safe navigation, where the robots are required to travel to their destination from initial locations under uncertain environments.

Motion planning is a fundamental problem in robotics, and it aims to generate a series of low-level specifications for a robot to move from one point to another [91]. In the real world, robots' operations are usually accompanied by uncertainties, e.g., from the environments they operate in and from the errors in the modeling of robots' dynamics. To deal with the uncertainties and ensure safety, i.e., collision avoidance, existing methods leverage techniques in robust control, e.g., [92, 93, 94], stochastic control, e.g., [95, 96, 97], and learning-based control e.g., [98, 99, 100]. Robust control-based approaches model the uncertainties as bounded sets and synthesize control policies that tolerate all the uncertainties in the sets. Note that considering all possible events can result in over-conservative policies whereas extreme events may only take place rarely. Stochastic control-based approaches model the uncertainties as known probability distributions. The generated motion

plans enforce chance constraints, i.e., the probability of collision is less than a given threshold. On the other hand, learning-based approaches relax the need of prior explicit uncertainty models by directly learning the best mapping from sensory inputs to control inputs from repetitive trials. Paper [100] leverages PAC-Bayes theory to provide guarantees on expected performances over a distribution of environments.

The aforementioned approaches can all be classified as offline approaches where control policies are synthesized before the deployment of the robots. When robots encounter significant changes of environments during online operation, online learning of the uncertainties is desired to ensure safe arrival to the goals. Recently, a class of methods on safe learning and control have been developed to safely steer a system to a goal region while learning uncertainties online. These approaches usually adopt a switching strategy between a learning-based controller, which updates online in light of new observations, and a backup safety controller, which is suboptimal but can guarantee safety. The backup safety controllers can be synthesized through solving a two-player zero-sum differential game [44], model predictive control (MPC) [101][34], control barrier function [102, 103], robust optimization [104], reachability analysis [105] and regions of attraction [46]. The aforementioned papers only consider single-robot systems and static state constraints (e.g., static obstacles). In multi-robot systems, from the perspective of each single robot and when centralized planning is not used, the state constraints are dynamic due to the motion of the other robots, analogous to moving obstacles.

Motion planning problems are known to be computationally challenging even for single-robot systems. Paper [106] shows that the generalized mover's problem is PSPACE-hard in terms of degrees of freedom. Multi-robot motion planning is an even more challenging problem as the computation complexity scales up by the number of robots. Centralized planners [107][108] consider all the robots as a single entity such that methods for single-robot motion planning can be directly applied. However, as [107] points out, its worst-case computation complexity grows exponentially with the number of robots. Consequently, distributed methods are developed to address the scalability issue. Most of these methods are featured with each robot conducting a single-robot motion planning strategy but coupled with a coordination scheme to resolve conflicts. These works can be categorized

as fully synthesized design or switching-based design. A fully synthesized design [109][110] incorporates simple collision avoidance methods, such as artificial potential field, into the decoupled solution. Under switching-based design, a switching controller is developed such that each robot executes a nominal controller synthesized in a decoupled manner but switches to a local coordination controller when it is close to other robots [111, 112, 113]. Prioritized planning, where a priority is assigned to the robots such that robots with lower priority make compromises for the robots with higher priority, is further adopted to reduce the need of coordination [114][115]. There have been recent works which study dynamic and environmental uncertainties in multi-robot motion planning. For example, robust control-based approaches are studied in [116][117], and stochastic control-based approaches are studied in [118][119][120]. In [121][122], deep reinforcement learning is applied to train multiple robots to avoid collisions in an offline manner when explicit uncertainty models are not available. In this chapter, we consider learning the uncertainties in an online fashion with data collected sequentially on the robots' trajectories.

Contribution statement. We consider the problem of online multi-robot motion planning with general nonlinear dynamics subject to unknown external disturbances. We propose dSLAP, the **d**istributed **S**afe **L**earning **A**nd **P**lanning framework, where the robots collect streaming data to online learn about the disturbances, use the learned model to compute a set of safe actions that avoid collisions against the learning uncertainty, and then choose an action that balances between reaching the goals and actively exploring the disturbances. Our contribution is summarized as follows:

- We propose dSLAP, a distributed two-stage motion planner. It utilizes set-valued analysis to allows for fast adaptation to the sequence of dynamic models resulted from online learning. The planner first constructs a directed graph through connecting a robot's one-step forward sets, and then obtains a set of safe control inputs by removing the control inputs leading to collisions. Then a distributed model predictive controller selects safe control inputs balancing moving towards the goals and actively learning the disturbances.

- Our two-stage motion planning is in contrast to the classic formulation

[123][124] of optimal multi-robot motion planning, whose solutions solve collision avoidance and optimal arrival simultaneously and are known to be computationally challenging (PSPACE-hard [106]). Instead, dSLAP first solves for collision avoidance and then for optimal arrival. The worst-case onboard computational complexity of each robot grows linearly with respect to the number of the robots.

- We derive sufficient conditions to guarantee the safety of the robots in the absence of backup policies.

Monte Carlo simulation is conducted for evaluations.

Notations. We use superscript $(\cdot)^{[i]}$ to distinguish the local values of robot i. Define the distance metric $\rho(x, x') \triangleq \|x - x'\|_\infty$, the point-to-set distance as $\rho(x, \mathcal{S}) \triangleq \inf_{x' \in \mathcal{S}} \rho(x, x')$ for a set $\mathcal{S}$, the closed ball centered at $x \in \mathbb{R}^{n_x}$ with radius r as $\mathcal{B}(x, r) \triangleq \{x' \in \mathbb{R}^{n_x} | \rho(x, x') \leqslant r\}$, and shorthand $\mathcal{B}$ the closed unit ball centered at 0 with radius 1. Let $\mathbb{Z}$ be the space of integers and $\mathbb{N}$ the space of natural number. Denote the cardinality of a set $\mathcal{S}$ as $|\mathcal{S}|$.

Below are the implementations of common procedures. *Element removal*: Given a set $\mathcal{S}$ and an element s, procedure Remove removes element s from $\mathcal{S}$; i.e., $\mathsf{Remove}(\mathcal{S}, s) \triangleq \mathcal{S} \setminus \{s\}$. *Element addition*: Given a set $\mathcal{S}$ and an element s, procedure Add appends s to $\mathcal{S}$, i.e., $\mathsf{Add}(\mathcal{S}, s) \triangleq \mathcal{S} \cup \{s\}$. *Nearest neighbor*: Given a state s and a finite set $\mathcal{S}$, $\mathsf{Nearest}$ chooses a state in $\mathcal{S}$ that is closest to s; i.e., $\mathsf{Nearest}(s, \mathcal{S})$ picks $y \in \mathcal{S}$, where $\rho(s, y) = \rho(s, \mathcal{S})$.

4.2 Problem formulation

In this section, we introduce the model of the multi-robot system, describe the formulation of the motion planning problem, and state the objective of this chapter.

Mobile multi-robot system. Consider a network of robots $\mathcal{V} \triangleq \{1, \cdots, n\}$. The dynamic system of each robot i is given by the following differential equation:

$$\dot{x}^{[i]}(t) = f^{[i]}(x^{[i]}(t), u^{[i]}(t)) + g^{[i]}(x^{[i]}(t), u^{[i]}(t)), \tag{4.1}$$

where $x^{[i]}(t) \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state of robot i at time t, $u^{[i]}(t) \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is its

control input, $f^{[i]}$ denotes the system dynamics of robot i, and $g^{[i]}$ represents the external unknown disturbance. We impose the following assumption:

Assumption 4.2.1.(A1) (*Lipschitz continuity).* The system dynamics $f^{[i]}$ and the unknown disturbance $g^{[i]}$ are Lipschitz continuous.

(A2) (*Compactness).* Spaces $\mathcal{X}$ and $\mathcal{U}$ are compact. ■

Assumption **(A1)** implies that $f^{[i]} + g^{[i]}$ is Lipschitz continuous. Choose constant $\ell^{[i]}$, which is larger than the Lipschitz constant of $f^{[i]} + g^{[i]}$ and constant $m^{[i]}$, which is larger than the supremum of $f^{[i]} + g^{[i]}$ over $\mathcal{X}$ and $\mathcal{U}$. Usually these constants can be estimated. For example, by the Bernoulli equation [125], the variation and magnitude of wind speed can be bounded due to the limited variation in temperature and air pressure.

Motion planning. We denote closed obstacle region by $\mathcal{X}_O \subseteq \mathcal{X}$, goal region by $\mathcal{X}_G^{[i]} \subseteq \mathcal{X} \backslash \mathcal{X}_O$, and free region at time t by $\mathcal{X}_F^{[i]}(x^{[\neg i]}(t)) \triangleq \mathcal{X} \backslash \big(\mathcal{X}_O \bigcup \cup_{j \neq i} \mathcal{B}(x^{[j]}(t), 2\zeta)\big)$, where $\neg i \triangleq \mathcal{V} \setminus \{i\}$ and $\zeta > 0$ is the diameter of an overestimation of the robot size. Each robot i aims to synthesize a feedback policy $\pi^{[i]} : \mathcal{X}^n \to \mathcal{U}$ such that the solution to system (4.1) under $\pi^{[i]}$ satisfies $x^{[i]}(t_*^{[i]}) \in \mathcal{X}_G^{[i]}$, $x^{[i]}(\tau) \in \mathcal{X}_F^{[i]}(x^{[\neg i]}(\tau))$, $0 \leqslant \tau \leqslant t_*^{[i]} < \infty$, where $t_*^{[i]}$ is the first time when robot i reaches $\mathcal{X}_G^{[i]}$. That is, each robot i needs to reach the goal region within finite time and be free of collision.

Problem statement. This chapter aims to solve the above multi-robot motion planning problem despite unknown function $g^{[i]}$. Since $g^{[i]}$ is unknown, it is necessarily to learn $g^{[}[i]]$ online to ensure each robot reaches the goal safe and fast. The challenge of the problem stems from the need of (fast) distributed planning with respect to a sequence of general nonlinear dynamic models resulted from online learning subject to dynamic constraints. Specifically, since the unknown function $g^{[i]}$ is learned online, each robot i should quickly adapt its motion planner in response to a sequence of newly learned models and the motion of the other robots.

4.3 Distributed safe learning and planning

In this section, we propose the dSLAP framework. Figure 4.1 shows one iteration of the algorithm in robot i. In each iteration k, the robot executes two modules *in*

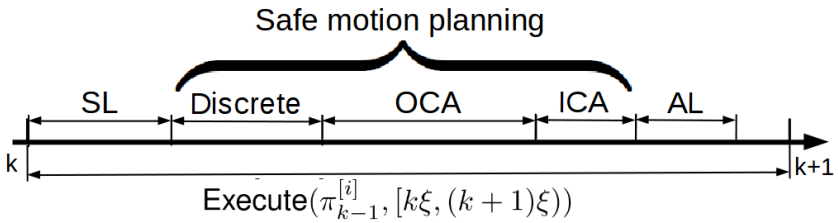


Figure 4.1: Implementation of dSLAP over one iteration

parallel. One is the computation module where robot i first collects a new dataset $\mathcal{D}_k^{[i]}$ and performs system learning (SL) to update the predictive mean $\mu_k^{[i]}$ and standard deviation $\sigma_k^{[i]}$ of the unknown dynamics. Next safe motion planning is performed, which includes dynamics discretization (Discrete) that outputs one-step forward sets $\mathsf{FR}_k^{[i]}$ under discretization parameter p_k, obstacle collision avoidance (OCA) that outputs a preliminary set of safe states $\mathcal{X}_{\text{safe},k}^{[i]}$, and inter-robot collision avoidance (ICA) that outputs a final $\mathcal{X}_{\text{safe},k}^{[i]}$. Finally, active learning (AL) is applied to synthesize control policy $\pi_k^{[i]}$. The discretization parameter is incremented in each iteration to obtain finer discretization for tighter approximation $\mathsf{FR}_k^{[i]}$. The other is the control module where the control policy $\pi_{k-1}^{[i]}$, computed in iteration $k-1$, is executed for all $t \in [k\xi, (k+1)\xi)$, where ξ is the discrete time unit. The dSLAP framework is formally stated in Algorithm 5.

4.3.1 System learning

In this section, we introduce the SL procedure for learning the external disturbance $g^{[i]}$. In each iteration k, each robot i first collects a new dataset $\mathcal{D}_k^{[i]}$ through the CollectData procedure, which returns

$$\mathcal{D}_k^{[i]} \triangleq \{g^{[i]}(x^{[i]}(\tau), u^{[i]}(\tau)) + e^{[i]}(\tau), x^{[i]}(\tau), u^{[i]}(\tau)\}_{\tau=(k-1)\xi}^{(k-1)\xi+\delta\bar{\tau}},$$

where $e^{[i]}(\tau) \sim \mathcal{N}(0, (\sigma_e^{[i]})^2 I_{n_x})$ is robot i's local observation error, δ is the sampling period, and $\bar{\tau}$ is the number of samples to be obtained. Then robot i independently estimates $g^{[i]}$ through Gaussian process regression (GPR) [54] using all the collected data $\cup_{k'=1}^{k} \mathcal{D}_{k'}^{[i]}$. By specifying prior mean function $\mu_0 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{n_x}$, and prior covariance function $\kappa : [\mathcal{X} \times \mathcal{U}] \times [\mathcal{X} \times \mathcal{U}] \to \mathbb{R}_{>0}$, GPR models $g^{[i]}$ as a sample from a Gaussian process prior $\mathcal{GP}(\mu_0, \kappa)$ and predicts $g^{[i]}(x^{[i]}, u^{[i]}) \sim$

Algorithm 5 The dSLAP framework

Input: State space: $\mathcal{X}$; Control input space: $\mathcal{U}$; Obstacle: $\mathcal{X}_O$; Goal: $\mathcal{X}_G^{[i]}, i \in \mathcal{V}$; Kernel for GPR: κ; Initial discretization parameter: p_{init}; Termination iteration: $\tilde{k}$; Number of samples to be obtained: $\bar{\tau}$; Discrete time unit: ξ; Time horizon for MPC: φ; Weight in the MPC: ψ; Sampling period: δ; Utility function $r_k^{[i]}$; Lipschitz constant $\ell^{[i]}$; Prior supremum of dynamic model $m^{[i]}$
Init: $p_1 \leftarrow p_{init}$; $\pi_0^{[i]}, \forall i \in \mathcal{V}$
for $k = 1, 2, \cdots, \tilde{k}$ **do**
    for $i \in \mathcal{V}$ (Computation module) **do**
        $\mathcal{D}_k^{[i]} \leftarrow \mathsf{CollectData}$
        $\mu_k^{[i]}, \sigma_k^{[i]} \leftarrow \mathsf{SL}(\mathcal{D}_k^{[i]})$
        $\mathsf{FR}_k^{[i]} \leftarrow \mathsf{Discrete}(p_k)$
        $\mathcal{X}_{safe,k}^{[i]} \leftarrow \mathsf{OCA}$
        $\mathcal{X}_{safe,k}^{[i]} \leftarrow \mathsf{ICA}$
        $\pi_k^{[i]} \leftarrow \mathsf{AL}(\mathcal{X}_{safe,k}^{[i]})$
        $p_{k+1} \leftarrow p_k + 1$
    end for
    for $i \in \mathcal{V}$ (Control module) **do**
        $\mathsf{Execute}(\pi_{k-1}^{[i]}, [k\xi, (k+1)\xi))$
    end for
end for

$\mathcal{N}(\mu_k^{[i]}(x^{[i]},u^{[i]}),(\sigma_k^{[i]}(x^{[i]},u^{[i]}))^2)$. We use recursive GPR [81] to maintain constant complexity.

4.3.2 Safe motion planning

Safe motion planning is a multi-grid algorithm utilizing set-valued analysis. Inspired by [123][124], we propose a new set-valued dynamics to discretize robot dynamics (4.1). We use the set-valued dynamics to approximate the one-step forward set and construct a directed graph. We then identify safe states and remove control inputs which lead to collision with the obstacles and other robots.

Dynamics discretization. First, we use confidence interval $\mu_k^{[i]}(x^{[i]},u^{[i]})+\gamma\sigma_k^{[i]}(x^{[i]},u^{[i]})\mathcal{B}$, where γ is the reliability factor, to approximate the unknown function $g^{[i]}(x^{[i]},u^{[i]})$. Then in each iteration k, inspired by [123][124], we incorporate the GPR prediction of $g^{[i]}$ and approximate the one-step forward reachability set starting from state $x^{[i]}$ under control input $u^{[i]}$ using discretized set-valued dynamics

$$\begin{aligned}\mathsf{FR}_k^{[i]}\big(x^{[i]},u^{[i]}\big)\triangleq\Big[&x^{[i]}+\epsilon^{[i]}(f^{[i]}(x^{[i]},u^{[i]})+\mu_k^{[i]}(x^{[i]},u^{[i]}))\\&+(\epsilon^{[i]}\gamma\bar{\sigma}_k^{[i]}+\alpha_{p_k}^{[i]}+h_{p_k})\mathcal{B}\Big]\cap\mathcal{X}_{p_k},\end{aligned}$$

where $\epsilon^{[i]}$ is the time duration, $\bar{\sigma}_k^{[i]}\triangleq\sup_{x^{[i]}\in\mathcal{X},u^{[i]}\in\mathcal{U}}\sigma_k^{[i]}(x^{[i]},u^{[i]})$, $\alpha_{p_k}^{[i]}\triangleq 2h_{p_k}+2\epsilon^{[i]}h_{p_k}\ell^{[i]}+(\epsilon^{[i]})^2\ell^{[i]}m^{[i]}$, and spatial discretization parameters are

$$h_{p_k}\triangleq 2^{-p_k},\mathcal{X}_{p_k}\triangleq h_{p_k}\mathbb{Z}^{n_x}\cap\mathcal{X},\mathcal{U}_{p_k}\triangleq h_{p_k}\mathbb{Z}^{n_u}\cap\mathcal{U}. \tag{4.2}$$

Furthermore, we write temporal resolution $\epsilon^{[i]}=\lambda^{[i]}\xi$, where $\lambda^{[i]}$ is a constant that ensures each iteration k with duration ξ can be partitioned into an integer number of small intervals with duration $\epsilon^{[i]}$. Notice that, by (4.2), finer discretization, corresponding to a larger p_k, provides tighter approximation of the dynamic model, whereas coarser discretization returns solutions faster. Hence, we increment the discretization parameter at each iteration to refine the discretization such that the spatial resolution is reduced by half and less conservative actions can be incrementally uncovered.

Obstacle collision avoidance (Algorithm 6). Procedure OCA aims to identify the safe states of the set-valued dynamic system and remove the control inputs

Algorithm 6 Procedure OCA

$\mathcal{X}^{[i]}_{\text{unsafe},k,0} \leftarrow \emptyset$
for $x^{[i]} \in \mathcal{X}_{p_k}$ **do**
  if $\rho(x^{[i]}, \mathcal{X}_O) \leqslant m^{[i]}\epsilon^{[i]} + h_{p_k}$ **then**
    $\mathsf{Add}(\mathcal{X}^{[i]}_{\text{unsafe},k,0}, x^{[i]})$
  else
    $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) \leftarrow \mathcal{U}_{p_k}$
    for $u^{[i]} \in \mathcal{U}^{[i]}_{p_k}(x^{[i]})$ **do**
      for $y^{[i]} \in \mathsf{FR}^{[i]}_k(x^{[i]}, u^{[i]})$ **do**
        $\mathsf{Add}(\mathsf{BR}^{[i]}_k(y^{[i]}, u^{[i]}), x^{[i]})$
      end for
    end for
  end if
end for
$\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0} \leftarrow \mathsf{UnsafeUpdate}(\mathcal{X}^{[i]}_{\text{unsafe},k,0})$
$\mathcal{X}^{[i]}_{\text{safe},k} \leftarrow \mathcal{X}_{p_k} \setminus \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0}$
Return $\mathcal{X}^{[i]}_{\text{safe},k}$

that lead to collision with the obstacles. Informally, a state is safe if there is a controller that can keep the robot from colliding with the obstacles when the robot starts from the state. Otherwise, the state is unsafe. Then procedure OCA consists of two steps as follows.

First, each robot identifies a preliminary set of unsafe states if the distance $\rho(x^{[i]}, \mathcal{X}_O)$ between state $x^{[i]}$ and the obstacle $\mathcal{X}_O$ is less than $m^{[i]}\epsilon^{[i]} + h_{p_k}$. The distance $m^{[i]}\epsilon^{[i]}+h_{p_k}$ represents an over-approximation of the distance the robot can reach within one time step with size $\epsilon^{[i]}$ on $\mathcal{X}_{p_k}$. This distance prevents the robot from "cutting the corner" of the obstacles due to the discretization. If state $x^{[i]}$ is more than this distance away from $\mathcal{X}_O$, $\mathsf{BR}^{[i]}_k$, the one-step $\epsilon^{[i]}$-duration backward set of $y^{[i]}$ applied $u^{[i]}$, is constructed as follow

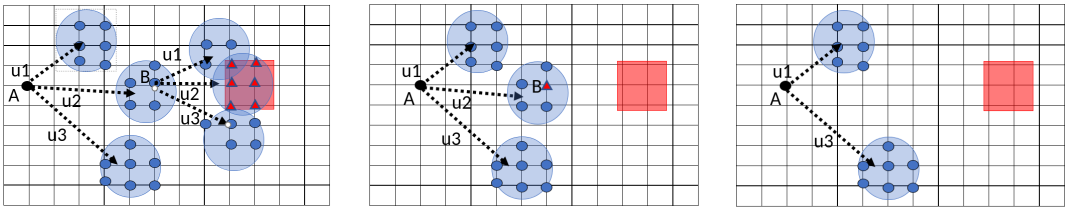
$$\forall y^{[i]} \in \mathsf{FR}^{[i]}_k(x^{[i]}, u^{[i]}), x^{[i]} \in \mathsf{BR}^{[i]}_k(y^{[i]}, u^{[i]}),$$

for each $u^{[i]} \in \mathcal{U}_{p_k}$.

In the second step, robot i runs procedure UnsafeUpdate (Algorithm 7) to iteratively remove all the control inputs that lead to the identified unsafe states. If all the control inputs $\mathcal{U}^{[i]}_{p_k}(x^{[i]})$ are removed, state $x^{[i]}$ is identified as unsafe and

Algorithm 7 $\mathsf{UnsafeUpdate}(\mathcal{X}^{[i]}_{\text{unsafe},k,j})$

$\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j} \leftarrow \mathcal{X}^{[i]}_{\text{unsafe},k,j}$
$Flag \leftarrow 1$
while $Flag == 1$ **do**
  $Flag \leftarrow 0$
  for $y^{[i]} \in \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ **do**
    for $u^{[i]} \in \mathcal{U}_{p_k}$ **do**
      for $x^{[i]} \in \mathsf{BR}^{[i]}_k(y^{[i]}, u^{[i]})$ **do**
        $\mathsf{Remove}(\mathcal{U}^{[i]}_{p_k}(x^{[i]}), u^{[i]})$
        if $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) == \emptyset$ and $x^{[i]} \notin \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ **then**
          $\mathsf{Add}(\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}, x^{[i]})$
          $Flag \leftarrow 1$
        end if
      end for
    end for
  end for
end while
Return $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$



(a) (b) (c)

Figure 4.2: A graphical illustration of obstacle collision avoidance

included in the set $\mathcal{X}^{[i]}_{\text{unsafe},k,0}$, together with the states that are within $m^{[i]}\epsilon^{[i]} + h_{p_k}$ of the obstacles. Robot i's set of unsafe states $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0}$ is then completed. For each state in $\mathcal{X}_{p_k} \setminus \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0}$, we have $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) \neq \emptyset$ and any control $u^{[i]} \in \mathcal{U}^{[i]}_{p_k}(x^{[i]})$ can ensure collision avoidance with the obstacles for one iteration.

A graphical illustration of OCA is shown in Figure 4.2. The square denotes the obstacle and the intersections on the grid denote the states on the discrete state space $\mathcal{X}_p$. The triangle states are unsafe. The arrows show the state transitions given the control input, which is the FR procedure. Starting from Figure 4.2a, the one-step forward reachability sets of state B under all the control inputs, $u1$, $u2$ and $u3$, have intersections with the obstacle, and hence these are unsafe control

Algorithm 8 Procedure ICA

$\mathcal{X}_k^{[i]} \leftarrow x_q^{[i]}(k\xi) + (2\xi m^{[i]} + 2\zeta + m^{[i]}\epsilon^{[i]} + 2h_{p_k})\mathcal{B}$
Broadcast($\mathcal{X}_k^{[i]}$)
for $j \in \mathcal{V}$, $j \neq i$ **do**
  if $j < i$ **then**
    $\mathcal{X}_{unsafe,k,j}^{[i]} \leftarrow \left[\mathcal{X}_k^{[j]} + \epsilon^{[i]}\gamma\bar{\sigma}_k^{[i]}\mathcal{B}\right] \cap \mathcal{X}_{p_k}$
    $\bar{\mathcal{X}}_{unsafe,k,j}^{[i]} \leftarrow$ UnsafeUpdate($\mathcal{X}_{unsafe,k,j}^{[i]}, k$)
    $\mathcal{X}_{safe,k}^{[i]} \leftarrow \mathcal{X}_{safe,k}^{[i]} \setminus \bar{\mathcal{X}}_{unsafe,k,j}^{[i]}$
  end if
end for
Return $\mathcal{X}_{safe,k}^{[i]}$

inputs and removed from state B. Since there is no more (safe) control input left for state B, i.e., $\mathcal{U}_{p_k}^{[i]}(\text{state } B) = \emptyset$, it is labeled as unsafe as in Figure 4.2b. Since state B is unsafe, control input u_2 is removed from state A. This gives Figure 4.2c, where state A is safe with control inputs u_1 and u_3, and state A is in the sets $\mathsf{BR}^{[i]}(\text{state } D, u_3)$ and $\mathsf{BR}^{[i]}(\text{state } C, u_1)$.

Inter-robot collision avoidance (Algorithm 8). Procedure ICA adopts a priority planning scheme in each iteration and aims to further remove the control inputs that lead to collision with the robots with higher priority. Each robot is assigned with a unique priority level. The robots with higher priority are treated as moving obstacles and removes all the control inputs that lead to these obstacles. First, each robot i broadcasts its reachability sets $\mathcal{X}_k^{[i]}$ within an iteration at the beginning of each iteration k. Upon receiving the messages from each robot j with higher priority, i.e., $j < i$, robot i identifies a new set of unsafe states $\mathcal{X}_{\text{unsafe},k,j}^{[i]}$ induced by $\mathcal{X}_k^{[i]}$ in the discrete state space $\mathcal{X}_{p_k}$. Second, robot i invokes procedure UnsafeUpdate to remove all the control inputs leading to the newly identified unsafe states. Robot i then updates the set of the safe states $\mathcal{X}_{safe,k}^{[i]}$ by removing the new unsafe states $\bar{\mathcal{X}}_{\text{unsafe},k,j}^{[i]}$. For each state $x^{[i]} \in \mathcal{X}_{\text{safe},k}^{[i]}$, $\mathcal{U}_k^{[i]}(x^{[i]}) \neq \emptyset$ and any control $u^{[i]} \in \mathcal{U}_k^{[i]}(x^{[i]})$ can ensure collision avoidance with the obstacles and the robots with higher priority for one iteration. Notice that in the worst case, each robot removes all the control inputs in its own state-control space. Therefore, the worst-case computation complexity is independent of the number of robots.

Algorithm 9 AL

1: **Procedure** $\pi_k^{[i]}(x^{[i]}(t))$
2: $w[k] \leftarrow e^{-\psi k}$
3: $\hat{x}^{[i]}(t) \leftarrow \mathsf{Nearest}(x^{[i]}(t), \mathcal{X}_{safe,k}^{[i]})$
4: $(u_*^{[i]}(t), \cdots, u_*^{[i]}(t+\varphi\epsilon^{[i]})) \leftarrow$ solve MPC in (4.3)
5: **Return** $u_*^{[i]}(t)$

4.3.3 Active learning and real-time control

In this section, we utilize the safe control inputs obtained above and synthesize a model predictive controller (MPC) to actively learn the disturbance $g^{[i]}$ and approach the goal.

First, the current state $x^{[i]}(t)$ of robot i is projected onto $\mathcal{X}_{\text{safe},k}^{[i]}$; the projection is $\hat{x}^{[i]}(t) \triangleq \mathsf{Nearest}(x^{[i]}(t), \mathcal{X}_{\text{safe},k}^{[i]})$. Second, we capture the objective of goal reaching using distance $\rho(\hat{x}^{[i]}(t+\varphi\epsilon^{[i]}), \mathcal{X}_G^{[i]})$, where $\varphi \in \mathbb{N}$ is the discrete horizon of the MPC formulated below. Then the objective of exploration is described by a utility function $r_k^{[i]}(\hat{x}^{[i]}(t), u^{[i]}(t))$; candidate utility functions, e.g., $r_k^{[i]}(\hat{x}^{[i]}(t), u^{[i]}(t)) = \sigma_k^{[i]}(\hat{x}^{[i]}(t), u^{[i]}(t))$, are available in [126]. Next, the safety constraint is honored by choosing control inputs from the safe control set $\mathcal{U}_k^{[i]}(\hat{x}^{[i]}(t))$. Lastly, the dynamic constraint is approximated by the one-step forward set $\mathsf{FR}_k^{[i]}$. Formally, the controller $\pi_k^{[i]} : \mathcal{X} \to \mathcal{U}$ returns control inputs by solving the finite-horizon optimal control problem:

$$\min \quad (1-w[k])\rho(\hat{x}^{[i]}(t+\varphi\epsilon^{[i]}), \mathcal{X}_G^{[i]}) + w[k] \sum_{\tau=t}^{t+\varphi\epsilon^{[i]}} r_k^{[i]}(\hat{x}^{[i]}(\tau), u^{[i]}(\tau)), \tag{4.3}$$

where the decision variables are $u^{[i]}(t) \in \mathcal{U}_k^{[i]}(\hat{x}^{[i]}(t)), \cdots, u^{[i]}(t+\varphi\epsilon^{[i]}) \in \mathcal{U}_k^{[i]}(\hat{x}^{[i]}(t+\varphi\epsilon^{[i]}))$, subject to $\hat{x}^{[i]}(\tau+\epsilon^{[i]}) \in \mathsf{FR}_k^{[i]}(\hat{x}^{[i]}(\tau), u^{[i]}(\tau))$ for all $\tau \in \{t, t+\epsilon^{[i]}, \cdots, t+(\varphi-1)\epsilon^{[i]}\}$. To ensure the robot eventually reaches the goal, we select the weight $w[k] \triangleq e^{-\psi k}$ for some $\psi > 0$ such that $w[k]$ diminishes.

The above finite-horizon optimal control problem is solved once for every time duration $\epsilon^{[i]}$, and the returned control input is fixed for a duration $\epsilon^{[i]}$. Specifically, consider a sequence $\{t_{k+1,n}^{[i]}\}_{n=0}^{\bar{n}_{k+1}^{[i]}} \subset [(k+1)\xi, (k+2)\xi]$, where $t_{k+1,0}^{[i]} = (k+1)\xi$, $t_{k+1,n}^{[i]} = t_{k+1,n-1}^{[i]} + \epsilon^{[i]}$ and $\bar{n}_{k+1}^{[i]} \triangleq \xi/\epsilon^{[i]}$. Procedure $\pi_k^{[i]}(x^{[i]}(t_{k+1,n}^{[i]}))$ solves the above finite-horizon optimal control problem at $n = 0, 1, \cdots, \bar{n}_{k+1}^{[i]} - 1$. The solution has

the form $(u_*^{[i]}(t_{k+1,n}^{[i]}), \cdots, u_*^{[i]}(t_{k+1,n}^{[i]} + \varphi\epsilon^{[i]}))$, and $\pi_k^{[i]}(x^{[i]}(t_{k+1,n}^{[i]})) = u_*^{[i]}(t_{k+1,n}^{[i]})$ is returned as the control input, and for all $t \in [t_{k+1,n}^{[i]}, t_{k+1,n+1}^{[i]})$, we have $u^{[i]}(t) = u_*^{[i]}(t_{k+1,n}^{[i]})$. The controller execution is denoted as procedure Execute in Algorithm 5.

4.3.4 Performance guarantees

In this section, we provide the performance guarantees for dSLAP. To obtain theoretic guarantees, we assume that $g^{[i]}$ is a realization of a known Gaussian process. For notational simplicity, we assume $g^{[i]} \in \mathbb{R}$. Generalizing $g^{[i]}$ to multi-dimensional can be done by applying the union bound.

Assumption 4.3.1. *(Realization of process).* It satisfies that $g^{[i]} \in \mathbb{R}$ and $g^{[i]} \sim \mathcal{GP}(\mu_0, \kappa)$. ■

That is, function $g^{[i]}$ is a realization of Gaussian process with prior mean μ_0 and kernel κ. This assumption is common in the analysis of GPR (Theorem 1, [84]). Theorem 4.3.2 below provides the probability of the robots being safe until the end of an iteration if they are around the set of safe states at the beginning of the iteration.

Theorem 4.3.2. *(One-iteration safety).* Suppose Assumptions 4.2.1 and 4.3.1 hold. If $\mathcal{B}(x^{[i]}(k\xi), h_{p_{k-1}}) \cap \mathcal{X}_{safe,k-1}^{[i]} \neq \emptyset$, $k \geqslant 1$, for all $i \in \mathcal{V}$, then dSLAP renders $x_q^{[i]}(t) \in \mathcal{X}_F^{[i]}(x_q^{[\neg i]}(t))$ for all time $t \in [k\xi, k\xi + \xi)$ with probability at least $1 - |\mathcal{V}||\mathcal{X}_p||\mathcal{U}_p|e^{-\gamma^2/2}$. ■

The proof of Theorem 4.3.2 can be found in Section 4.4.3.

Denote $\bar{\sigma} \triangleq \|\kappa\|_{[\mathcal{X}\times\mathcal{U}]\times[\mathcal{X}\times\mathcal{U}]}$. Then Theorem 4.3.3 below provides the probability as well as the requirement on discretization and the computation speed of the robots such that they can be safe throughout the entire mission.

Theorem 4.3.3. *(All-time safety).* Suppose $4\gamma\bar{\sigma}\epsilon^{[i]} \leqslant h_{p_{\tilde{k}}}$ and Assumptions 4.2.1 and 4.3.1 hold. Suppose $\xi \leqslant \frac{h_{p_{\tilde{k}}}}{\max_{j\in\mathcal{V}} m^{[j]}}$. For all $i \in \mathcal{V}$, if $\mathcal{B}(x^{[i]}(k\xi), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$, for some $k \geqslant 1$, then the dSLAP algorithm renders that $x_q^{[i]}(t) \in \mathcal{X}_F^{[i]}(x_q^{[\neg i]}(t))$, for all $t \geqslant k\xi$ with probability at least $1 - \tilde{k}|\mathcal{V}||\mathcal{X}_{p_{\tilde{k}}}||\mathcal{U}_{p_{\tilde{k}}}|e^{-\gamma^2/2}$. ■

The proof of Theorem 4.3.3 can be found in Section 4.4.4.

The requirement for ξ indicates that the robots' onboard computation should be fast with respect to the speed of robots, while the computation can be relieved with coarse discretization. This provides the required update frequency for the decoupled controller.

The sufficient condition $4\gamma\bar{\sigma}\epsilon^{[i]} \leqslant h_{p_{\tilde{k}}}$ imposes a requirement in designing set-valued dynamics to discretize robot dynamics (4.1) using $\mathsf{FR}_k^{[i]}$. Given Assumption 4.3.1, $\gamma\bar{\sigma}$ represents an upper bound over the variability of the disturbances the robots want to tolerate. On the right hand side, $h_{p_{\tilde{k}}}$ represents the minimal spatial resolution. Then the sufficient condition implies that the product between the variability of the disturbances and the temporal resolution should be small with respect to and the spatial resolution.

4.3.5 Discussion

(Probabilistic safety). The probability of the safety guarantees stems from the analysis of GPR estimates being able to capture the ground truth dynamics over the whole state-action space (and over all the iterations). The analysis can be conservative but is independent of the other components in the proposed algorithm. In order to reduce the probability of unsafe execution, or increase the probability of safe execution, the robots can increase γ according to the theorems. This may cause conservative actions as γ is a factor for constructing the one-step forward set $\mathsf{FR}_k^{[i]}$ and could lead to no solution at all if γ is too large. However, this can be addressed by having the robots collecting more data online to train the GPR such that $\sigma_k^{[i]}(x^{[i]}, u^{[i]})$ becomes small.

(Verification of $\mathcal{B}(x^{[i]}(0), h_{p_0}) \cap \mathcal{X}_{\text{safe},0}^{[i]} \neq \emptyset$*).* To ensure the robots are safe for all the time, Theorem 4.3.3 implies that it suffices to satisfy the sufficient condition $\mathcal{B}(x^{[i]}(0), h_{p_0}) \cap \mathcal{X}_{\text{safe},0}^{[i]} \neq \emptyset$ *a priori*. To achieve this, one can compute $\mathcal{X}_{\text{safe},0}^{[i]}$ using data collected a priori or a prior conservative estimates of the disturbances. This prior knowledge can be obtained in most situations by, e.g., using historical data. Examples include wind speed, water current, and road texture in a local area. In addition, smaller h_{p_0} can enlarge the set $\mathcal{X}_{\text{safe},0}^{[i]}$ such that more initial states $x^{[i]}(0)$ can satisfy the condition.

(Computation complexity). The algorithms in [123][124] aim for solving optimal arrival and collision avoidance simultaneously in a centralized manner. The computation complexities scale as $\mathcal{O}((|\mathcal{X}_{p_k}||\mathcal{U}_{p_k}|)^n)$, which grows exponentially in the number of robots. In order to reduce the computational complexity, dSLAP has two stages. The first stage includes procedure OCA and ICA, which are distributed and remove unsafe control inputs on the discrete state-control space of each robot. OCA is independent of the other robots. ICA augments the reachability sets of higher priority robots and correspondly removes the unsafe control inputs. Its worst-case onboard computational complexity of each robot scales linearly with n. The local safe control inputs enable decoupled planning through AL in the second stage, whose computation complexity is also independent of n. Furthermore, the computation complexity can be reduced during the implementation by successively removing the unsafe state-control pairs. Each state-control pair only needs to be removed at most once, and in each robot there are at most $|\mathcal{X}_{p_k}||\mathcal{U}_{p_k}|$ pairs to be removed, which is independent of n.

(Strength and weakness). The proposed framework dSLAP is able to compute safe control inputs for a multi-robot system with general nonlinear dynamics in a distributed manner amid online uncertainty learning. Nevertheless, dSLAP can be conservative for the following two reasons. First, it overapproximates the continuous dynamics using discretized set-valued dynamics. To enable fast computation, the discretization is usually coarse and hence the approximation error can be large, which leads to conservative actions. However, this conservativeness can be reduced via finer discretization provided sufficient computation power. Second, the coordination among the robots is simple. The application of prioritized planning is suboptimal and can lose completeness [127]. Furthermore, higher priority robots are viewed as moving obstacles by lower priority robots. The overapproximation of the reachability sets in two iterations of the higher priority robots are conservatives since the overapproximation is determined by the maximum speed of the robots multiplied by the duration of two iterations. This conservativeness can be reduced by developing a more sophisticated scheme of coordination among the robots, optimizing the assignment of priority levels, and/or shortening the duration of one iteration. We leave this for future work. Furthermore, dSLAP can suffer from the curse-of-dimensionality for each individual system.

4.4 Proof

In this section, we prove Theorems 4.3.2 and 4.3.3. Below is a roadmap for the proofs.

1. We present the concentration inequality resulted from GPR in Section 4.4.1. This provides the probability of the event that $g^{[i]}$ belongs to the tube $\mu_k^{[i]} \pm \gamma\sigma_k^{[i]}$. The rest of the analysis is performed under this event.

2. Section 4.4.2 introduces a set of preliminary notations for set-valued mappings and the related properties. The properties examine the approximation of system dynamics (4.1) through the set-valued mappings.

3. Utilizing the set-valued approximations of dynamics (4.1), Section 4.4.3 derives that $\mathcal{B}(x^{[i]}(k\xi), h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$ for some $k \geqslant 1$ is a sufficient condition to ensure that robot i being collision-free during iteration k and hence proves Theorem 4.3.2.

4. Given one-iteration safety in iteration k, Section 4.4.4 examines the distance between $x^{[i]}((k+1)\xi)$ and $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}$ as well as the inclusion of $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ in terms of $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}$.

5. Utilizing the two relations in (iv), the distance $\rho(x^{[i]}((k+1)\xi), \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j})$ can be characterized. This is further used in Section 4.4.4 to establish the sufficient condition for $\mathcal{B}(x^{[i]}((k+1)\xi), h_{p_k}) \cap \mathcal{X}^{[i]}_{\text{safe},k} \neq \emptyset$ to hold, given $\mathcal{B}(x^{[i]}(k\xi), h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$. This ensures one-iteration safety hold in iteration $k+1$ and completes the proof of Theorem 4.3.3.

4.4.1 Concentration inequality of Gaussian process

The concentration inequality resulted from GPR is presented in Lemma 4.4.1 below.

Lemma 4.4.1. Under Assumption 4.3.1, for any discretization parameter p and each robot i, the following holds with probability at least $1-|\mathcal{X}_p||\mathcal{U}_p|e^{-\gamma^2/2}$: $\forall x^{[i]} \in \mathcal{X}_p, u^{[i]} \in \mathcal{U}_p$,

$$|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - g^{[i]}(x^{[i]}, u^{[i]})| \leqslant \gamma\sigma_k^{[i]}(x^{[i]}, u^{[i]}). \tag{4.4}$$

Proof: The proof mainly follows the proof of Lemma 5.1 in [84]. At iteration k, we have the input dataset $Z_{1:k}^{[i]}$ and output dataset $y_{1:k}^{[i]}$, where

$$\begin{aligned}
y_{1:k}^{[i]} &\triangleq [[y_1^{[i]}]^T, \cdots, [y_k^{[i]}]^T]^T, \\
y_l^{[i]} &\triangleq [g(x^{[i]}(\tau), u^{[i]}(\tau)) + e^{[i]}(\tau)]_{\tau=(l-1)\xi}^{(l-1)\xi+\delta\bar{\tau}}, \\
Z_l^{[i]} &\triangleq \{x^{[i]}(\tau), u^{[i]}(\tau)\}_{\tau=(l-1)\xi}^{(l-1)\xi+\delta\bar{\tau}}, Z_{1:k}^{[i]} \triangleq \{Z_l^{[i]}\}_{l=1}^{k},
\end{aligned}$$

Let test input $z_*^{[i]} \in \mathcal{Z}_p \triangleq \mathcal{X}_p \times \mathcal{U}_p$. Assumption 4.3.1 gives

$$\begin{bmatrix} y_{1:k}^{[i]} \\ g^{[i]}(z_*^{[i]}) \end{bmatrix} = \mathcal{N}(\begin{bmatrix} \mu_0(Z_{1:k}^{[i]}) \\ \mu_0(z_*^{[i]}) \end{bmatrix}, \begin{bmatrix} \kappa(Z_{1:k}^{[i]}, Z_{1:k}^{[i]}) + (\sigma_e^{[i]})^2 I & \kappa(Z_{1:k}^{[i]}, z_*^{[i]}) \\ \kappa(z_*^{[i]}, Z_{1:k}^{[i]}) & \kappa(z_*^{[i]}, z_*^{[i]}) \end{bmatrix}),$$

where $\kappa(Z_{1:k}^{[i]}, z_*^{[i]}) \triangleq [\kappa(z^{[i]}, z_*^{[i]})]_{z^{[i]} \in Z_{1:k}^{[i]}}$, and $\kappa(Z_{1:k}^{[i]}, Z_{1:k}^{[i]}) \triangleq [\kappa(z^{[i]}, \tilde{z}^{[i]})]_{z^{[i]}, \tilde{z}^{[i]} \in Z_{1:k}^{[i]}}$.

Applying identities of joint Gaussian distribution (page 200, [54]), we obtain the posterior distribution $g^{[i]}(z_*^{[i]}) \sim \mathcal{N}(\mu_k^{[i]}(z_*^{[i]}), (\sigma_k^{[i]}(z_*^{[i]}))^2)$, where

$$\begin{aligned}
\mu_k^{[i]}(z_*^{[i]}) &\triangleq \mu_0^{[i]}(z_*^{[i]}) + \kappa(z_*^{[i]}, Z_{1:k}^{[i]})(\kappa(Z_{1:k}^{[i]}, Z_{1:k}^{[i]}) + (\sigma_e^{[i]})^2)^{-1} \cdot (y_{1:k}^{[i]} - \mu_0(Z_{1:k}^{[i]})), \\
(\sigma_k^{[i]}(z_*^{[i]}))^2 &\triangleq \kappa(z_*^{[i]}, z_*^{[i]}) + \kappa(z_*^{[i]}, Z_{1:k}^{[i]},)(\kappa(Z_{1:k}^{[i]}, Z_{1:k}^{[i]}) + (\sigma_e^{[i]})^2)^{-1} \kappa(Z_{1:k}^{[i]}, z_*^{[i]}).
\end{aligned}$$

Consider $r \sim \mathcal{N}(0,1)$. It holds that for $c > 0$,

$$\begin{aligned}
Pr\{r > c\} &= e^{-c^2/2}(2\pi)^{-1/2} \int_c^\infty e^{-\frac{(r-c)^2}{2} - c(r-c)} dr \\
&\leqslant e^{-c^2/2} Pr\{r > 0\} = \frac{1}{2} e^{-c^2/2}
\end{aligned}$$

where the inequality uses the fact $e^{-c(r-c)} \leqslant 1$ for $r \geqslant c$. Analogously, $Pr\{r < -c\} \leqslant \frac{1}{2} e^{-c^2/2}$. Therefore, let $r = \frac{g^{[i]}(z_*^{[i]}) - \mu_k^{[i]}(z_*^{[i]})}{\sigma_k^{[i]}(z_*^{[i]})}$ and $c = \gamma$, we have $Pr\{|g^{[i]}(z_*^{[i]}) - \mu_k^{[i]}(z_*^{[i]})| > \gamma \sigma_k^{[i]}(z_*^{[i]})\} \leqslant e^{-\gamma^2/2}$. Denote event $E_{z_*^{[i]}} \triangleq \{|g^{[i]}(z_*^{[i]}) - \mu_k^{[i]}(z_*^{[i]})| > \gamma \sigma_k^{[i]}(z_*^{[i]})\}$. Applying the union bound (Theorem 2-3, [88]), we have

$$Pr\{\cup_{z_*^{[i]} \in \mathcal{Z}_p} E_{z_*^{[i]}}\} \leqslant |\mathcal{Z}_p| e^{-\gamma^2/2}.$$

Note that $Pr\{\cap_{z_*^{[i]}\in\mathcal{Z}_p} E_{z_*^{[i]}}\} = 1 - Pr\{\cup_{z_*^{[i]}\in\mathcal{Z}_p} E_{z_*^{[i]}}\}$. Hence,

$$|g^{[i]}(z_*^{[i]}) - \mu_k^{[i]}(z_*^{[i]})| \leqslant \gamma\sigma_k^{[i]}(z_*^{[i]}),$$

simultaneously for all $z_*^{[i]} \in \mathcal{Z}_p$ with probability at least $1 - |\mathcal{Z}_p|e^{-\gamma^2/2} = 1 - |\mathcal{X}_p||\mathcal{U}_p|e^{-\gamma^2/2}$. ■

4.4.2 Set-valued approximation

In this section, we first introduce a set of set-valued notations from [124] to discretize system (4.1) in the time and state spaces. Lemma 4.4.2 shows that the set-valued discretization is a good approximation of the continuous system (4.1). Then we discuss other properties in the discrete space.

Define

$$\begin{aligned}
F^{[i]}(x^{[i]}, u^{[i]}) &\triangleq f^{[i]}(x^{[i]}, u^{[i]}) + g^{[i]}(x^{[i]}, u^{[i]}),\\
F_\epsilon^{[i]}(x^{[i]}, u^{[i]}) &\triangleq F^{[i]}(x^{[i]}, u^{[i]}) + m^{[i]}\ell^{[i]}\epsilon\mathcal{B},\\
G_\epsilon^{[i]}(x^{[i]}, u^{[i]}) &\triangleq x^{[i]} + \epsilon F_\epsilon^{[i]}(x^{[i]}, u^{[i]}).
\end{aligned}$$

Page 222 of [124] uses the following discrete set-valued map

$$\Gamma_{\epsilon,h}^{[i]}(x^{[i]}, u^{[i]}) \triangleq [G_\epsilon^{[i]}(x^{[i]}, u^{[i]}) + 2(1 + \ell^{[i]}\epsilon)h\mathcal{B}] \cap \mathcal{X}_p, \tag{4.5}$$

which is discrete in time and state, to approximate system (4.1), which is continuous in time and state. Let

$$x^{[i]}(t_0, t_0 + \epsilon, u^{[i]}) \triangleq x^{[i]}(t_0) + \int_0^\epsilon f^{[i]}(x^{[i]}(t_0 + \tau), u^{[i]}) + g^{[i]}(x^{[i]}(t_0 + \tau), u^{[i]})d\tau$$

be the state at time $t_0 + \epsilon$ when system (4.1) starts from $x^{[i]}(t_0)$ at time t_0 and applies constant input $u^{[i]} \in \mathcal{U}$ within the time interval $[t_0, t_0 + \epsilon]$. Lemma 4.4.2 below shows that $G_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]})$ contains the trajectory of system (4.1) under constant control for any duration ϵ.

Lemma 4.4.2. Under Assumption 4.2.1, for any $x^{[i]}(t_0) \in \mathcal{X}$, $t_0 \geqslant 0$, $u^{[i]} \in \mathcal{U}$ and $\epsilon > 0$, it holds that $x^{[i]}(t_0, t_0 + \epsilon, u^{[i]}) \in G_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]})$.

Proof: The proof is part of the proof on page 194 in [124]. Let $\tau \in [0, \epsilon]$. By **(A1)** in Assumption 4.2.1 and the definitions of $m^{[i]}$ and $x^{[i]}(t_0, t_0+\epsilon, u^{[i]})$, we have $\|x^{[i]}(t_0, t_0+\tau, u^{[i]}) - x^{[i]}(t_0)\| \leqslant \tau m^{[i]} \leqslant \epsilon m^{[i]}$. Lipschitz continuity further gives that

$$F^{[i]}(x^{[i]}(t_0, t_0+\tau, u^{[i]}), u^{[i]}) \in F^{[i]}(x^{[i]}(t_0), u^{[i]}) + m^{[i]}\ell^{[i]}\epsilon\mathcal{B} = F_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]}).$$

Then

$$\begin{aligned}\dot{x}^{[i]}(t_0, t_0+\tau, u^{[i]}) &= (f^{[i]} + g^{[i]})(x^{[i]}(t_0, t_0+\tau, u^{[i]}), u^{[i]}) \\ &= F^{[i]}(x^{[i]}(t_0, t_0+\tau, u^{[i]}), u^{[i]}) \in F_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]}). \end{aligned} \tag{4.6}$$

By definition, $F_\epsilon^{[i]}(x^{[i]}, u^{[i]})$ is compact and convex since it is just a closed ball, which indicates that $G_\epsilon^{[i]}(x^{[i]}, u^{[i]})$ is also convex and compact. For any state $\lambda \in \mathcal{X}$, (4.6) renders

$$\sup_{y^{[i]} \in F_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]})} \langle y^{[i]}, \lambda\rangle \geqslant \langle \dot{x}^{[i]}(t_0, t_0+\tau, u^{[i]}), \lambda\rangle, \tau \in [0, \epsilon].$$

Applying integration gives

$$\sup_{y^{[i]} \in x^{[i]}(t_0) + \epsilon F_\epsilon^{[i]}(x^{[i]}(t_0), u^{[i]})} \langle y^{[i]}, \lambda\rangle \geqslant \langle x^{[i]}(t_0, t_0+\epsilon, u^{[i]}), \lambda\rangle.$$

Therefore, applying the Separating Hyperplane Theorem on page 46 in [86], we prove the lemma. ■

Similarly, let

$$\begin{aligned}&\tilde{F}_k^{[i]}(x^{[i]}, u^{[i]}) \triangleq f^{[i]}(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]}) + \gamma\sigma_k^{[i]}(x^{[i]}, u^{[i]})\mathcal{B}, \\ &\tilde{F}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}) \triangleq \tilde{F}_k^{[i]}(x^{[i]}, u^{[i]}) + m^{[i]}\ell^{[i]}\epsilon\mathcal{B}, \\ &\tilde{G}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}) \triangleq x^{[i]} + \epsilon\tilde{F}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}), \\ &\tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon) \triangleq [\tilde{G}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}) + 2(1+\ell^{[i]}\epsilon)h_p\mathcal{B}] \cap \mathcal{X}_p. \end{aligned} \tag{4.7}$$

Denote $\delta_{\gamma,p} \triangleq |\mathcal{X}_p||\mathcal{U}_p|e^{-\gamma^2/2}$. By Lemma 4.4.1, we have

$$g^{[i]}(x^{[i]}, u^{[i]}) \in \mu_k^{[i]}(x^{[i]}, u^{[i]}) + \gamma\sigma_k^{[i]}(x^{[i]}, u^{[i]})\mathcal{B} \tag{4.8}$$

for all $x^{[i]} \in \mathcal{X}_p$ and $u^{[i]} \in \mathcal{U}_p$ with probability at least $1 - \delta_{\gamma,p}$. This gives each of the followings holds with probability at least $1 - \delta_{\gamma,p}$ for all $x^{[i]} \in \mathcal{X}_p, u^{[i]} \in \mathcal{U}_p$:

$$\begin{aligned}
&F^{[i]}(x^{[i]}, u^{[i]}) \subseteq \tilde{F}_k^{[i]}(x^{[i]}, u^{[i]}),\ F_\epsilon^{[i]}(x^{[i]}, u^{[i]}) \subseteq \tilde{F}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}),\\
&G_\epsilon^{[i]}(x^{[i]}, u^{[i]}) \subseteq \tilde{G}_{\epsilon,k}^{[i]}(x^{[i]}, u^{[i]}), \quad \Gamma_{\epsilon,h_p}^{[i]}(x^{[i]}, u^{[i]}) \subseteq \tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon).
\end{aligned}$$

Lemma 4.4.3 characterizes the relation between $\tilde{\mathsf{FR}}$ and FR.

Lemma 4.4.3. For any $x^{[i]} \in \mathcal{X}$ and $u^{[i]} \in \mathcal{U}$, it holds that $[\tilde{\mathsf{FR}}_k^{[i]}\big(x^{[i]}, u^{[i]}, p_k, \epsilon^{[i]}\big) + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k} \subset \mathsf{FR}_k^{[i]}\big(x^{[i]}, u^{[i]}\big)$.

Proof: Consider state $y^{[i]} \in \tilde{\mathsf{FR}}_k^{[i]}\big(x^{[i]}, u^{[i]}, p_k, \epsilon^{[i]}\big)$. Then

$$y^{[i]} \in [x^{[i]} + \epsilon^{[i]}(f^{[i]}(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]})) + (\gamma\sigma_k^{[i]}(x^{[i]}, u^{[i]})\epsilon^{[i]} + \alpha_{p_k}^{[i]})\mathcal{B}].$$

Hence $y^{[i]} + h_{p_k}\mathcal{B} \subset [x^{[i]} + \epsilon^{[i]}(f^{[i]}(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]})) + (\gamma\bar{\sigma}_k^{[i]}\epsilon^{[i]} + \alpha_{p_k}^{[i]} + h_{p_k})\mathcal{B}]$. This gives

$$\begin{aligned}
[y^{[i]} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k} \subset [&x^{[i]} + \epsilon^{[i]}(x^{[i]}(f^{[i]}(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]}))\\
&+ (\gamma\bar{\sigma}_k^{[i]}\epsilon^{[i]} + \alpha_{p_k}^{[i]} + h_{p_k})\mathcal{B}] \cap \mathcal{X}_{p_k},
\end{aligned}$$

where the right hand side is equivalent to $\mathsf{FR}_k^{[i]}\big(x^{[i]}, u^{[i]}\big)$. ■

Recall that Assumption 4.2.1 implies that the system dynamics $f^{[i]} + g^{[i]}$, or $F^{[i]}$, is Lipschitz continuous. In particular, $F_\epsilon^{[i]}$ is identical to the example in the Lipschitz case on page 191 in [124], and hence it satisfies Assumptions H0, H1, and H2 in [124]. Therefore, applying Lemma 4.13 [124] gives

$$\bigcup_{\rho(\tilde{x}^{[i]}, x^{[i]}) \leqslant h_p} [G_\epsilon^{[i]}(\tilde{x}^{[i]}, u^{[i]}) + h_p\mathcal{B}] \cap \mathcal{X}_p \subset \Gamma_{\epsilon,h_p}^{[i]}(x^{[i]}, u^{[i]}). \tag{4.9}$$

Below are the other properties in the discrete space. Lemmas 4.4.4 and 4.4.5 show that $\tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon)$ almost contains the union of $G_\epsilon^{[i]}(x^{[i]}, u^{[i]})$.

Lemma 4.4.4. Suppose that Assumptions 4.2.1 and (4.8) hold $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}, i \in \mathcal{V}$. For any $x^{[i]} \in \mathcal{X}$, $u^{[i]} \in \mathcal{U}$ and $\epsilon > 0$, for all $y^{[i]} \in \cup_{\rho(\tilde{x}^{[i]}, x^{[i]}) \leqslant h_p} G_\epsilon^{[i]}(\tilde{x}^{[i]}, u^{[i]})$, it holds that $\mathsf{Nearest}(y^{[i]}, \mathcal{X}_p) \in \tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon)$.

Proof: Since $y^{[i]} \in \cup_{\rho(\tilde{x}^{[i]}, x^{[i]}) \leqslant h_p} G_\epsilon^{[i]}(\tilde{x}^{[i]}, u^{[i]})$, we have

$$y^{[i]} + h_p\mathcal{B} \subset \cup_{\rho(\tilde{x}^{[i]}, x^{[i]}) \leqslant h_p} G_\epsilon^{[i]}(\tilde{x}^{[i]}, u^{[i]}) + h_p\mathcal{B}.$$

Combining this with (4.9), we have

$$[y^{[i]} + h_p\mathcal{B}] \cap \mathcal{X}_p \subset \Gamma_{\epsilon, h_p}^{[i]}(x^{[i]}, u^{[i]}).$$

Since (4.8) holds, we have

$$[y^{[i]} + h_p\mathcal{B}] \cap \mathcal{X}_p \subset \tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon).$$

Note that $\mathsf{Discrete}$ renders $[y^{[i]} + h_p\mathcal{B}] \cap \mathcal{X}_p \neq \emptyset$. Hence $\mathsf{Nearest}(y, \mathcal{X}_p) \in [y^{[i]} + h_p\mathcal{B}] \cap \mathcal{X}_p \subset \tilde{\mathsf{FR}}_k^{[i]}(x^{[i]}, u^{[i]}, p, \epsilon)$. ■

Lemma 4.4.5. Consider closed set $\mathcal{A} \subset \mathcal{X}$ and $\Delta > \frac{1}{2}h_{p_k}$. Suppose event (4.8) is true $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}$, $i \in \mathcal{V}$. For any $\tilde{x}^{[i]} \in \mathcal{X}$, $u^{[i]} \in \mathcal{U}$ and $\epsilon > 0$, if

$$\rho\big(x^{[i]}, \mathcal{A}\big) \geqslant \Delta, \forall x^{[i]} \in \tilde{\mathsf{FR}}_k^{[i]}(\tilde{x}^{[i]}, u^{[i]}, p_k, \epsilon), \tag{4.10}$$

then $\forall y^{[i]} \in \cup_{\rho(\mathring{x}^{[i]}, \tilde{x}^{[i]}) \leqslant h_p} G_\epsilon^{[i]}(\mathring{x}^{[i]}, u^{[i]})$, $\rho(y^{[i]}, \mathcal{A}) \geqslant \Delta$.

Proof: We prove the claim by contradiction. Suppose (4.10) is true but there exists $y^{[i]} \in G_\epsilon^{[i]}(\mathring{x}^{[i]}, u^{[i]})$, $\mathring{x}^{[i]} \in \mathcal{B}(\tilde{x}^{[i]}, h_{p_k})$ such that $\rho(y^{[i]}, \mathcal{A}) < \Delta$.

Since $\mathcal{A}$ is closed, there exists $\hat{x}^{[i]} \in \mathcal{A}$ such that $\rho(y^{[i]}, \hat{x}^{[i]}) = \rho(y^{[i]}, \mathcal{A}) < \Delta$. Let $y_j^{[i]}$ be the j-th element of $y^{[i]}$. Define operation $\mathsf{Floor}(z) \triangleq \max\{z' \in \mathbb{Z} | z' \leqslant z\}$ that finds the largest integer no greater than $z \in \mathbb{R}$ and recall $\mathsf{Ceil}(z) = \min\{z' \in \mathbb{Z} | z' \geqslant z\}$. Denote real values $\underline{y}_j^{[i]} \triangleq h_{p_k}\mathsf{Floor}(\frac{y_j^{[i]}}{h_{p_k}})$ and $\bar{y}_j^{[i]} \triangleq h_{p_k}\mathsf{Ceil}(\frac{y_j^{[i]}}{h_{p_k}})$. Then for each dimension j, we have two cases:

1). $\hat{x}_j^{[i]} \in [\underline{y}_j^{[i]}, \bar{y}_j^{[i]}]$. Notice that $0 \leqslant \bar{y}_j^{[i]} - \underline{y}_j^{[i]} \leqslant h_{p_k}$. Choose $\check{x}_j^{[i]} = \frac{1}{2}(\bar{y}_j^{[i]} + \underline{y}_j^{[i]})$. It is easy to see that $|\check{x}_j^{[i]} - \hat{x}_j^{[i]}| \leqslant \frac{1}{2}h_{p_k} < \Delta$.

2). $\hat{x}_j^{[i]} \notin [\underline{y}_j^{[i]}, \bar{y}_j^{[i]}]$. We can select $\check{x}_j^{[i]} = \underline{y}_j^{[i]}$ if $\hat{x}_j^{[i]} < \underline{y}_j^{[i]}$; otherwise, we have $\hat{x}_j^{[i]} > \bar{y}_j^{[i]}$, and we can select $\check{x}_j^{[i]} = \bar{y}_j^{[i]}$. Note that $y_j^{[i]} \in [\underline{y}_j^{[i]}, \bar{y}_j^{[i]}]$. Therefore under

this selection, we have $|\check{x}_j^{[i]} - \hat{x}_j^{[i]}| \leqslant |y_j^{[i]} - \hat{x}_j^{[i]}| < \Delta$.

Since $\hat{x}^{[i]} \in \mathcal{A}$, the above two cases imply that $\rho(\check{x}^{[i]}, \mathcal{A}) \leqslant \rho(\check{x}^{[i]}, \hat{x}^{[i]}) < \Delta$ and $\check{x}^{[i]} \in \mathcal{X}_{p_k}$. Note that

$$\check{x}^{[i]} \in \mathcal{B}(y^{[i]}, h_{p_k}) = y^{[i]} + h_{p_k}\mathcal{B} \subset [G_\epsilon^{[i]}(\mathring{x}^{[i]}, u^{[i]}) + h_{p_k}\mathcal{B}].$$

Combining this with (4.9) renders $\check{x}^{[i]} \in \Gamma^{[i]}_{\epsilon, h_{p_k}}(\tilde{x}^{[i]}, u^{[i]})$. Since (4.8) is true, we have $\check{x}^{[i]} \in \tilde{\mathsf{FR}}_k^{[i]}(\tilde{x}^{[i]}, u^{[i]}, p_k, \epsilon)$, which contradicts (4.10). ■

Let $x^{[i]} \in \mathcal{X}_{p_k}$ and $u^{[i]} \in \mathcal{U}_{p_k}$. By definition, we can write

$$\mathsf{FR}_k^{[i]}(x^{[i]}, u^{[i]}) = \mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k} \cap \mathcal{X}_{p_k}, \tag{4.11}$$

where $\mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k} \triangleq x^{[i]} + \epsilon^{[i]}(f(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]})) + (\gamma\bar{\sigma}_k^{[i]}\epsilon^{[i]} + \alpha_{p_k}^{[i]} + h_{p_k})\mathcal{B}$. Let $y^{[i]} \in [x^{[i]} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_{k-1}}$. Lemma 4.4.6 below characterizes the relation between $\mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k}$ and $\mathcal{B}^{[i]}_{y^{[i]}, u^{[i]}, k-1}$.

Lemma 4.4.6. Suppose $4\gamma\bar{\sigma}\epsilon^{[i]} \leqslant h_{p_{\bar{k}}}$ and (4.8) holds. It holds that $\mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k} + h_{p_k}\mathcal{B} \subset \mathcal{B}^{[i]}_{y^{[i]}, u^{[i]}, k-1}$.

Proof: Let $\check{x}^{[i]} \in \mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k} + h_{p_k}\mathcal{B}$. Denote $\tilde{f}_k^{[i]}(x^{[i]}, u^{[i]}) \triangleq f^{[i]}(x^{[i]}, u^{[i]}) + \mu_k^{[i]}(x^{[i]}, u^{[i]})$. Then applying triangular inequality, we have

$$\begin{aligned}\rho\big(\check{x}^{[i]}, y^{[i]} + \epsilon^{[i]}\tilde{f}_{k-1}^{[i]}(y^{[i]}, u^{[i]})\big) \leqslant & \rho\big(\check{x}^{[i]}, x^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(x^{[i]}, u^{[i]})\big) \\ & + \rho\big(x^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(x^{[i]}, u^{[i]}), y^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(y^{[i]}, u^{[i]})\big) \\ & + \rho\big(y^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(y^{[i]}, u^{[i]}), y^{[i]} + \epsilon^{[i]}\tilde{f}_{k-1}^{[i]}(y^{[i]}, u^{[i]})\big).\end{aligned} \tag{4.12}$$

Next we find the upper bound of each term on the right hand side of (4.12). Since $\check{x}^{[i]} \in \mathcal{B}^{[i]}_{x^{[i]}, u^{[i]}, k} + h_{p_k}\mathcal{B}$, we have

$$\rho\big(\check{x}^{[i]}, x^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(x^{[i]}, u^{[i]})\big) \leqslant \gamma\bar{\sigma}_k^{[i]}\epsilon^{[i]} + \alpha_{p_k}^{[i]} + 2h_{p_k}. \tag{4.13}$$

Since $\rho(x^{[i]}, y^{[i]}) \leqslant h_{p_k}$, Lipschitz continuity yields $\|f^{[i]}(x^{[i]}, u^{[i]}) - f^{[i]}(y, u^{[i]})\| \leqslant \ell^{[i]}h_{p_k}$. Then applying triangular inequality gives

$$\rho\big(x^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(x^{[i]}, u^{[i]}), y^{[i]} + \epsilon^{[i]}\tilde{f}_k^{[i]}(y^{[i]}, u^{[i]})\big)$$

$$
\begin{aligned}
&\leqslant \|x^{[i]} - y^{[i]}\| + \|\epsilon^{[i]} \tilde{f}_k^{[i]}(x^{[i]}, u^{[i]}) - \epsilon^{[i]} \tilde{f}_k^{[i]}(y^{[i]}, u^{[i]})\| \\
&\leqslant h_{p_k} + \epsilon^{[i]} \|f^{[i]}(x^{[i]}, u^{[i]}) - f^{[i]}(y^{[i]}, u^{[i]})\| \\
&\quad + \epsilon^{[i]} \|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - \mu_k^{[i]}(y^{[i]}, u^{[i]})\| \\
&\leqslant h_{p_k} + \epsilon^{[i]} \ell^{[i]} h_{p_k} + \epsilon^{[i]} \|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - \mu_k^{[i]}(y^{[i]}, u^{[i]})\|. \qquad (4.14)
\end{aligned}
$$

Since (4.8) holds for all $x^{[i]} \in \mathcal{X}$ and $u^{[i]} \in \mathcal{U}$, we have

$$
\begin{aligned}
&\|g^{[i]}(x^{[i]}, u^{[i]}) - \mu_k^{[i]}(x^{[i]}, u^{[i]})\| \leqslant \gamma \sigma_k^{[i]}(x^{[i]}, u^{[i]}) \leqslant \gamma \bar{\sigma}_k^{[i]} \\
&\text{and } \|g^{[i]}(y^{[i]}, u^{[i]}) - \mu_{k-1}^{[i]}(y^{[i]}, u^{[i]})\| \leqslant \gamma \bar{\sigma}_{k-1}^{[i]}. \qquad (4.15)
\end{aligned}
$$

Since $\rho(x^{[i]}, y^{[i]}) \leqslant h_{p_k}$, the Lipschitz continuity of $g^{[i]}$ renders $\|g^{[i]}(x^{[i]}, u^{[i]}) - g^{[i]}(y^{[i]}, u^{[i]})\| \leqslant \ell^{[i]} h_{p_k}$. Then applying triangular inequality gives

$$
\begin{aligned}
&\|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - \mu_k^{[i]}(y^{[i]}, u^{[i]})\| \\
&= \|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - g^{[i]}(x^{[i]}, u^{[i]}) + g^{[i]}(x^{[i]}, u^{[i]}) g^{[i]}(y^{[i]}, u^{[i]}) + g^{[i]}(y^{[i]}, u^{[i]}) - \mu_k^{[i]}(y^{[i]}, u^{[i]})\| \\
&\leqslant \|\mu_k^{[i]}(x^{[i]}, u^{[i]}) - g^{[i]}(x^{[i]}, u^{[i]})\| + \|g^{[i]}(x^{[i]}, u^{[i]}) - g^{[i]}(y^{[i]}, u^{[i]})\| \\
&\quad + \|g^{[i]}(y^{[i]}, u^{[i]}) - \mu_k^{[i]}(y^{[i]}, u^{[i]})\| \\
&\leqslant \ell^{[i]} h_{p_k} + 2\gamma \bar{\sigma}_k^{[i]}.
\end{aligned}
$$

Combining the above inequality with (4.14) gives

$$
\begin{aligned}
&\rho\big(x^{[i]} + \epsilon^{[i]} \tilde{f}_k^{[i]}(x^{[i]}, u^{[i]}), y^{[i]} + \epsilon^{[i]} \tilde{f}_k^{[i]}(y^{[i]}, u^{[i]})\big) \\
&\leqslant h_{p_k} + 2\epsilon^{[i]} \ell^{[i]} h_{p_k} + 2\gamma \bar{\sigma}_k^{[i]} \epsilon^{[i]}. \qquad (4.16)
\end{aligned}
$$

Applying triangular inequality, we can further write

$$
\begin{aligned}
&\rho\big(y^{[i]} + \epsilon^{[i]} \tilde{f}_k^{[i]}(y^{[i]}, u^{[i]}), y^{[i]} + \epsilon^{[i]} \tilde{f}_{k-1}^{[i]}(y^{[i]}, u^{[i]})\big) \\
&= \|\epsilon^{[i]} \tilde{f}_k^{[i]}(y^{[i]}, u^{[i]}) - \epsilon^{[i]} \tilde{f}_{k-1}^{[i]}(y^{[i]}, u^{[i]})\| \\
&= \|\epsilon^{[i]}(f + g^{[i]} - g^{[i]} + \mu_k^{[i]})(y^{[i]}, u^{[i]}) - \epsilon^{[i]}(f + g^{[i]} - g^{[i]} + \mu_{k-1}^{[i]})(y^{[i]}, u^{[i]})\| \\
&\leqslant \epsilon^{[i]} \|(\mu_k^{[i]} - g^{[i]})(y^{[i]}, u^{[i]})\| + \epsilon^{[i]} \|(\mu_{k-1}^{[i]} - g^{[i]})(y^{[i]}, u^{[i]})\| \\
&\leqslant \epsilon^{[i]} \gamma \bar{\sigma}_k^{[i]} + \epsilon^{[i]} \gamma \bar{\sigma}_{k-1}^{[i]}, \qquad (4.17)
\end{aligned}
$$

where the last inequality follows from (4.15).

Returning to (4.12), combining (4.13), (4.16) and (4.17) gives

$$\begin{aligned}&\rho\big(\check{x}^{[i]}, y^{[i]} + \epsilon^{[i]}\tilde{f}^{[i]}_{k-1}(y^{[i]}, u^{[i]})\big)\\ &\leqslant 4\gamma\bar{\sigma}^{[i]}_{k}\epsilon^{[i]} + \alpha^{[i]}_{p_k} + 3h_{p_k} + 2\epsilon^{[i]}\ell^{[i]}h_{p_k} + \gamma\bar{\sigma}^{[i]}_{k-1}\epsilon^{[i]}. \end{aligned} \tag{4.18}$$

Note that $4\gamma\bar{\sigma}\epsilon^{[i]} \leqslant h_{p_{\tilde{k}}} \leqslant h_{p_k}$. By equation (2.24) in [54], $\bar{\sigma} \geqslant \bar{\sigma}^{[i]}_{k'}$ for all $k' \geqslant 0$. Note that $\alpha^{[i]}_{p_k} = 2h_{p_k} + 2\epsilon^{[i]}h_{p_k}\ell^{[i]} + (\epsilon^{[i]})^2\ell^{[i]}m^{[i]}$ and $h_{p_{k-1}} = 2h_{p_k}$. Combining these gives

$$\begin{aligned}&\rho\big(\check{x}^{[i]}, y^{[i]} + \epsilon^{[i]}\tilde{f}^{[i]}_{k-1}(y^{[i]}, u^{[i]})\big)\\ &\leqslant \alpha^{[i]}_{p_k} + 4h_{p_k} + 2\epsilon^{[i]}\ell^{[i]}h_{p_k} + \gamma\bar{\sigma}^{[i]}_{k-1}\epsilon^{[i]} = \alpha^{[i]}_{p_{k-1}} + h_{p_{k-1}} + \gamma\bar{\sigma}^{[i]}_{k-1}\epsilon^{[i]},\end{aligned}$$

which implies $\check{x}^{[i]} \in \mathcal{B}^{[i]}_{y^{[i]},u^{[i]},k-1}$. ■

4.4.3 Proof of Theorem 4.3.2

Denote $\bar{\mathcal{X}}^{[i]}_O[k_0, k_1) \triangleq \mathcal{X}_O \bigcup \cup_{j<i, t\in[k_0\xi, k_1\xi)}\mathcal{B}(x^{[j]}_q(t), 2\zeta)$ the obstacle regions (i.e., static obstacles and the robots with higher priority) for robot i from time t_0 to t_1. Denote shorthand $x^{[i]}[k] \triangleq x^{[i]}(k\xi)$ for the state in discrete time.

Roadmap of the proof: First, in Lemma 4.4.7, we establish that $\mathsf{FR}^{[i]}_k(x^{[i]}, u^{[i]})$ is invariant in $\mathcal{X}^{[i]}_{\text{safe},k}$ under inputs in $\mathcal{U}^{[i]}_{p_k}(x^{[i]})$. Then we examine the distance between $\mathcal{X}^{[i]}_{\text{safe},k}$ and $\bar{\mathcal{X}}^{[i]}_O[k, k+2)$ in Lemma 4.4.8. Next Lemma 4.4.10 utilizes the previous two lemmas to show that system (4.1) stays safe throughout for a duration $[t^{[i]}_{k+1,n}, t^{[i]}_{k+1,n+1}]$ under constant control, and Lemma 4.4.11 extends the safety result to the piecewise constant control law rendered by dSLAP for one iteration. Finally, we incorporate the concentration inequality in Lemma 4.4.1 and prove Theorem 4.3.2.

Lemma 4.4.7. For all $x^{[i]} \in \mathcal{X}^{[i]}_{\text{safe},k}$, it holds that $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) \neq \emptyset$ and $\mathsf{FR}^{[i]}_k(x^{[i]}, u^{[i]}) \subset \mathcal{X}^{[i]}_{\text{safe},k}$ for all $u^{[i]} \in \mathcal{U}^{[i]}_{p_k}(x^{[i]})$.

Proof: We prove the lemma by contradiction.

Suppose $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) = \emptyset$. Control input removal only takes place in the OCA procedure and the $\mathsf{UnsafeUpdate}$ procedure. In both procedures, when it reduces

to $\mathcal{U}_{p_k}^{[i]}(x^{[i]}) = \emptyset$, the procedures rule that $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,0}^{[i]}$ or $x^{[i]} \in \bar{\mathcal{X}}_{\text{unsafe},k,j}^{[i]}$, $j < i$, through the Add procedure. Therefore, $x^{[i]} \notin \mathcal{X}_{\text{safe},k}^{[i]}$.

Suppose there exists $u^{[i]} \in \mathcal{U}_{p_k}^{[i]}(x^{[i]})$, such that $\mathsf{FR}_k^{[i]}(x^{[i]}, u^{[i]}) \cap [\cup_{j<i} \bar{\mathcal{X}}_{\text{unsafe},k,j}^{[i]}] \neq \emptyset$. Note that the OCA procedure constructs the $\mathsf{BR}_k^{[i]}$ set such that $x^{[i]} \in \mathsf{BR}_k^{[i]}(\tilde{x}^{[i]}, u^{[i]})$ for all $\tilde{x}^{[i]} \in \mathsf{FR}_k^{[i]}(x^{[i]}, u^{[i]})$. Due to the $\mathsf{UnsafeUpdate}$ procedure, if $\tilde{x}^{[i]} \in [\cup_{j<i} \bar{\mathcal{X}}_{\text{unsafe},k,j}^{[i]}]$ and $x^{[i]} \in \mathsf{BR}^{[i]}(\tilde{x}^{[i]}, u^{[i]})$, we must have $u^{[i]} \notin \mathcal{U}_{p_k}^{[i]}(x^{[i]})$. Hence, this case is impossible. ■

The following lemma characterizes the distance between $\mathcal{X}_{\text{safe},k}^{[i]}$ and $\bar{\mathcal{X}}_O^{[i]}[k, k+2)$.

Lemma 4.4.8. It holds that $\rho(x^{[i]}, \bar{\mathcal{X}}_O^{[i]}[k, k+2)) > m^{[i]}\epsilon^{[i]} + h_{p_k}$ for all $x^{[i]} \in \mathcal{X}_{\text{safe},k}^{[i]}$.

Proof: Let $x^{[i]} \in \mathcal{X}_{\text{safe},k}^{[i]}$. In OCA, when $\rho(x^{[i]}, \mathcal{X}_O) \leqslant m^{[i]}\epsilon^{[i]} + h_{p_k}$, $\mathsf{Add}(\mathcal{X}_{\text{unsafe},k,0}^{[i]}, x^{[i]})$ is executed such that $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,0}^{[i]}$. Since $\mathcal{X}_{safe,k}^{[i]} \subset \mathcal{X}_{p_k} \setminus \mathcal{X}_{\text{unsafe},k,0}^{[i]} \neq \emptyset$,

$$\rho(x^{[i]}, \mathcal{X}_O) > m^{[i]}\epsilon^{[i]} + h_{p_k}. \tag{4.19}$$

In ICA, when $j < i$, $\mathcal{X}_{\text{unsafe},k,j}^{[i]} \supset \mathcal{X}_k^{[j]} \cap \mathcal{X}_{p_k}$. Therefore, by definition of $\mathcal{X}_k^{[j]}$, if

$$\rho(x^{[i]}, x^{[j]}[k] + 2\xi m^{[j]}\mathcal{B}) \leqslant 2\zeta + m^{[i]}\epsilon^{[i]} + h_{p_k},$$

we must have $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,j}^{[i]}$. Hence

$$\rho(x^{[i]}, x^{[j]}[k] + 2\xi m^{[j]}\mathcal{B}) > 2\zeta + m^{[i]}\epsilon^{[i]} + h_{p_k}. \tag{4.20}$$

Note that **(A1)** in Assumption 4.2.1 implies

$$\dot{x}^{[i]}(\tau) = f^{[i]}(x^{[i]}(\tau), u^{[i]}(\tau)) + g^{[i]}(x^{[i]}(\tau), u^{[i]}(\tau)) \in m^{[i]}\mathcal{B},$$

for all $u^{[i]}(\tau) \in \mathcal{U}$. This implies

$$x^{[j]}(t) \in x^{[j]}[k] + 2\xi m^{[j]}\mathcal{B}, \ \forall t \in [k\xi, (k+2)\xi).$$

Combining this with (4.20) gives, $\forall t \in [k\xi, (k+2)\xi)$,

$$\rho(x^{[i]}, \mathcal{B}(x^{[j]}(t), 2\zeta)) > m^{[i]}\epsilon^{[i]} + h_{p_k}. \tag{4.21}$$

By definition, obstacle region $\bar{\mathcal{X}}_O^{[i]}[k,k+2) = \mathcal{X}_O \bigcup \cup_{j<i, t\in[k\xi,(k+2)\xi)} \mathcal{B}(x^{[j]}(t), 2\zeta)$. Hence the lemma directly follows from (4.19) and (4.21). ■

Recall that, for all iterations $k \geqslant 1$, AL and controller execution render $\{u^{[i]}(t_{k,n}^{[i]})\}_{n=0}^{\bar{n}_k^{[i]}-1}$, $u^{[i]}(t_{k,n}^{[i]}) \in \mathcal{U}_{p_{k-1}}^{[i]}$, as control inputs, each executed for a duration $\epsilon^{[i]}$ by robot i such that $u^{[i]}(t) = u^{[i]}(t_{k,n}^{[i]})$ for all $t \in [t_{k,n}^{[i]}, t_{k,n+1}^{[i]})$. Then we have $x^{[i]}(t_{k,n}^{[i]}) = x^{[i]}(t_{k,n-1}^{[i]}, t_{k,n-1}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n-1}^{[i]}))$. The following lemma gives the sufficient conditions such that the robots are near the safe states within one iteration.

Lemma 4.4.9. Suppose Assumption 4.2.1 holds and (4.8) is true $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}$, $i \in \mathcal{V}$. If $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$ for all $i \in \mathcal{V}$ for some $k > 1$, then $\mathcal{B}(x^{[i]}(t_{k,n}^{[i]}), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$ for all $n = 0, 1, \cdots, \bar{n}_k^{[i]}$, $i \in \mathcal{V}$.

Proof: We prove the lemma using induction. For the base case $n = 0$, the condition in the lemma statement and the definition $t_{k,0}^{[i]} = k\xi$ indicate that $\mathcal{B}(x^{[i]}(t_{k,0}^{[i]}), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$.

Now suppose that $\mathcal{B}(x^{[i]}(t_{k,n}^{[i]}), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$ holds up until $n = n'$. Then there exists $\tilde{x}_{n'}^{[i]} \in \mathcal{B}(x^{[i]}(t_{k,n'}^{[i]}), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]}$. By AL and controller execution, we have $u^{[i]}(t_{k,n'}^{[i]}) \in \mathcal{U}_{p_{k-1}}^{[i]}(\tilde{x}_{n'}^{[i]})$. By definition of $\tilde{x}_{n'}^{[i]}$, $\rho(x^{[i]}(t_{k,n'}^{[i]}), \tilde{x}_{n'}^{[i]}) \leqslant h_{p_{k-1}}$. By Lemma 4.4.2, we have

$$x^{[i]}\Big(t_{k,n'}^{[i]}, t_{k,n'}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n'}^{[i]})\Big) \in G_{\epsilon^{[i]}}(x^{[i]}(t_{k,n'}^{[i]}), u^{[i]}(t_{k,n'}^{[i]})).$$

Then Lemma 4.4.4 renders

$$\mathsf{Nearest}(x^{[i]}\Big(t_{k,n'}^{[i]}, t_{k,n'}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n'}^{[i]})\Big), \mathcal{X}_{p_{k-1}}) \in \tilde{\mathsf{FR}}_{k-1}^{[i]}(\tilde{x}_{n'}^{[i]}, u^{[i]}(t_{k,n'}^{[i]}), p_{k-1}, \epsilon^{[i]}).$$

Note that (4.2) implies

$$\mathsf{Nearest}(x^{[i]}\Big(t_{k,n'}^{[i]}, t_{k,n'}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n'}^{[i]})\Big), \mathcal{X}_{p_{k-1}}) \in \mathcal{B}(x^{[i]}\Big(t_{k,n'}^{[i]}, t_{k,n'}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n'}^{[i]})\Big), h_{p_{k-1}}).$$

Since $x^{[i]}(t_{k,n'+1}^{[i]}) = x^{[i]}(t_{k,n'}^{[i]}, t_{k,n'}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n'}^{[i]}))$, then the above two statements give

$$\mathcal{B}(x^{[i]}(t_{k,n'+1}^{[i]}), h_{p_{k-1}}) \cap \tilde{\mathsf{FR}}_{k-1}^{[i]}(\tilde{x}_{n'}^{[i]}, u^{[i]}(t_{k,n'}^{[i]}), p_{k-1}, \epsilon^{[i]}) \neq \emptyset. \tag{4.22}$$

Lemma 4.4.3 implies

$$\tilde{\mathsf{FR}}^{[i]}_{k-1}(\tilde{x}^{[i]}_{n'}, u^{[i]}(t^{[i]}_{k,n'}), p_{k-1}, \epsilon^{[i]}) \subset \mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}_{n'}, u^{[i]}(t^{[i]}_{k,n'})).$$

Since $\tilde{x}^{[i]}_{n'} \in \mathcal{X}^{[i]}_{\text{safe},k-1}$, Lemma 4.4.7 implies $\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}_{n'}, u^{[i]}(t^{[i]}_{k,n'})) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$. Combining these two statements with (4.22) gives $\mathcal{B}(x^{[i]}(t^{[i]}_{k,n'+1}), h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$. The induction is completed. ■

The following lemma characterizes a sufficient condition for the trajectory of system (4.1) under constant control input in $\mathcal{U}^{[i]}_{p_{k-1}}$ to be safe for a duration $\epsilon^{[i]}$.

Lemma 4.4.10. Suppose Assumption 4.2.1 holds and (4.8) is true $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}$, $i \in \mathcal{V}$. If $\tilde{x}^{[i]} \in \mathcal{B}(x^{[i]}(t^{[i]}_{k,n}), h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1}$ for all $i \in \mathcal{V}$ and some $t^{[i]}_{k,n} \in \{t^{[i]}_{k,n}\}_{n=0}^{\bar{n}^{[i]}_k - 1}$, it holds that

$$x^{[i]}(t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon, u^{[i]}) \in \mathcal{X}^{[i]}_F(x^{[\neg i]}(t)), \ \forall \epsilon \in [0, \epsilon^{[i]}],$$

for all $u^{[i]} \in \mathcal{U}^{[i]}_{p_{k-1}}(\tilde{x}^{[i]})$, $t \in [t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon^{[i]})$ and $i \in \mathcal{V}$.

Proof: Recall that Lemma 4.4.3 implies $\tilde{\mathsf{FR}}^{[i]}_{k-1}\big(x^{[i]}, u^{[i]}, p_{k-1}, \epsilon^{[i]}\big) \subset \mathsf{FR}^{[i]}_{k-1}\big(x^{[i]}, u^{[i]}\big)$ for all $x^{[i]} \in \mathcal{X}$ and $u^{[i]} \in \mathcal{U}$. Since $\tilde{x}^{[i]} \in \mathcal{X}^{[i]}_{\text{safe},k-1}$, Lemma 4.4.7 renders $\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}, u^{[i]}) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$. Thus, $\tilde{\mathsf{FR}}^{[i]}_{k-1}\big(x^{[i]}, u^{[i]}, p_{k-1}, \epsilon^{[i]}\big) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$. Combining these with Lemma 4.4.8 gives

$$\rho\big(x^{[i]}, \bar{\mathcal{X}}^{[i]}_O[k-1, k+1)\big) > m^{[i]}\epsilon^{[i]} + h_{p_{k-1}}, \tag{4.23}$$

for all $x^{[i]} \in \tilde{\mathsf{FR}}^{[i]}_{k-1}(\tilde{x}^{[i]}, u^{[i]}, p_{k-1}, \epsilon^{[i]})$.

Lemma 4.4.2 gives $x^{[i]}(t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon^{[i]}, u^{[i]}) \in G^{[i]}_{\epsilon^{[i]}}(x^{[i]}(t^{[i]}_{k,n}), u^{[i]})$. Since $\tilde{x}^{[i]} \in \mathcal{B}(x^{[i]}(t^{[i]}_{k,n}), h_{p_{k-1}})$, we have $x^{[i]}(t^{[i]}_{k,n}) \in \mathcal{B}(\tilde{x}^{[i]}, h_{p_{k-1}})$. Combining these two statements and (4.23) with Lemma 4.4.5 renders

$$\rho(x^{[i]}(t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon^{[i]}, u^{[i]}), \bar{\mathcal{X}}^{[i]}_O[k-1, k+1)) > m^{[i]}\epsilon^{[i]} + h_{p_{k-1}}. \tag{4.24}$$

Then by the definition of $m^{[i]}$, we have $\rho(x^{[i]}(t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon, u^{[i]}), x^{[i]}(t^{[i]}_{k,n})) \leqslant m^{[i]}\epsilon^{[i]}$ for all $\epsilon \in [0, \epsilon^{[i]}]$ Combining this with (4.24), it renders that, for all $\epsilon \in [0, \epsilon^{[i]}]$,

$$\rho(x^{[i]}(t^{[i]}_{k,n}, t^{[i]}_{k,n} + \epsilon, u^{[i]}), \bar{\mathcal{X}}^{[i]}_O[k-1, k+1) > h_{p_{k-1}}.$$

Combining this with the definition of $\bar{\mathcal{X}}_O^{[i]}[\cdot,\cdot)$ renders that

$$x^{[i]}(t_{k,n}^{[i]}, t_{k,n}^{[i]}+\epsilon, u^{[i]}) \in \mathcal{X} \setminus [\mathcal{X}_O \bigcup \cup_{j<i, t\in[(k-1)\xi,(k+1)\xi)} \mathcal{B}(x^{[j]}(t), 2\zeta)]. \tag{4.25}$$

Note that (4.25) holds for all $i \in \mathcal{V}$, it further gives

$$x^{[i]}(t_{k,n}^{[i]}, t_{k,n}^{[i]}+\epsilon, u^{[i]}) \in \mathcal{X} \setminus [\mathcal{X}_O \bigcup \cup_{j\neq i, t\in[(k-1)\xi,(k+1)\xi)} \mathcal{B}(x^{[j]}(t), 2\zeta)].$$

By definitions, for any $t \in [(k-1)\xi, (k+1)\xi)$,

$$\mathcal{X}_F^{[i]}(x^{[\neg i]}(t)) = \mathcal{X} \setminus [\mathcal{X}_O \bigcup \cup_{j\neq i} \mathcal{B}(x^{[j]}(t), 2\zeta)].$$

Combining the above two statements completes the proof. ■

The following lemma gives the sufficient conditions such that the robots steered by dSLAP are safe within one iteration.

Lemma 4.4.11. Suppose Assumption 4.2.1 holds and (4.8) is true $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}$, $i \in \mathcal{V}$. If $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]} \neq \emptyset$ for all $i \in \mathcal{V}$ for some $k > 1$, then $x^{[i]}(t) \in \mathcal{X}_F^{[i]}(x^{[\neg i]}(t))$ for all $t \in [k\xi, k\xi+\xi)$ and all $i \in \mathcal{V}$.

Proof: Lemma 4.4.9 shows that there exists $\tilde{x}_n^{[i]} \in \mathcal{B}(x^{[i]}(t_{k,n}^{[i]}), h_{p_{k-1}}) \cap \mathcal{X}_{\text{safe},k-1}^{[i]}$ for each $n = 0, 1, \bar{n}_k^{[i]} - 1$, $i \in \mathcal{V}$. By Lemma 4.4.10, we have

$$x^{[i]}(t_{k,n}^{[i]}, t_{k,n}^{[i]}+\epsilon, u^{[i]}(t_{k,n}^{[i]})) \in \mathcal{X}_F^{[i]}(x^{[\neg i]}(t)),$$

for all $\epsilon \in [0, \epsilon^{[i]})$, $t \in [k\xi, (k+1)\xi)$, and $u^{[i]}(t_{k,n}^{[i]}) \in \mathcal{U}_{p_{k-1}}^{[i]}(\tilde{x}_n^{[i]})$ for each $n = 0, 1, \cdots, \bar{n}_k^{[i]} - 1$. Recall that the definition that $t_{k,0}^{[i]} = k\xi$, $t_{k,n}^{[i]} = t_{k,n-1}^{[i]} + \epsilon^{[i]}$ and $t_{k,\bar{n}_k^{[i]}}^{[i]} = (k+1)\xi$. Hence, the lemma is proved. ■

Now we are ready to prove Theorem 4.3.2.

Proof of Theorem 4.3.2: Given Assumptions 4.2.1 and 4.3.1 hold, for each robot i, (4.8) holds with probability at least $1 - \delta_{\gamma,p_{k-1}}$ $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}$. Applying the union bound, this gives (4.8) holds with probability at least $1 - |\mathcal{V}|\delta_{\gamma,p_{k-1}}$, $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}$, $i \in \mathcal{V}$. Notice that Lemma 4.4.11 is given in the event of (4.8) is true $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}$, $i \in \mathcal{V}$. Therefore, we have Lemma 4.4.11 hold with probability at least $1 - |\mathcal{V}|\delta_{\gamma,p_{k-1}}$. ■

4.4.4 Proof of Theorem 4.3.3

Recall that $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ is the set of unsafe states induced by robot $1 \leqslant j < i$ and $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0}$ is the set of unsafe states induced by static obstacles. Given Theorem 4.3.2, we can prove Theorem 4.3.3 by showing that if $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$ holds for iteration k, it implies that $\mathcal{B}(x^{[i]}[k+1], h_{p_k}) \cap \mathcal{X}^{[i]}_{\text{safe},k} \neq \emptyset$ also holds, and hence Theorem 4.3.3 can be proved by induction. Therefore, under the condition of $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$, we study: 1) the distance between $x^{[i]}[k+1]$ and $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}$ (Lemma 4.4.12); 2) the inclusion of $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ in terms of $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}$ (Lemma 4.4.14). The two results imply the distance between $x^{[i]}[k+1]$ and $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ and further characterize the conditions for $\mathcal{B}(x^{[i]}[k+1], h_{p_k}) \cap \mathcal{X}^{[i]}_{\text{safe},k} \neq \emptyset$ (Lemma 4.4.16). Then the proof is concluded by combining these results.

D.1) The distance between $x^{[i]}[k+1]$ and $\bar{\mathcal{X}}^{[i]}_{\textbf{unsafe},k-1,j}$

Lemma 4.4.12. Given Assumption 4.2.1 and event (4.8) hold $\forall x^{[i]} \in \mathcal{X}_{p_{k-1}}, u^{[i]} \in \mathcal{U}_{p_{k-1}}, i \in \mathcal{V}$ and $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$, it holds that $\rho(x^{[i]}[k+1], \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant 2h_{p_{k-1}}$, for all $j < i$.

Proof: Lemma 4.4.9 shows that there exists $\tilde{x}^{[i]}_n \in \mathcal{B}(x^{[i]}(t^{[i]}_{k,n}), h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1}$ for each $n = 0, 1, \cdots, \bar{n}^{[i]}_k - 1$, $i \in \mathcal{V}$. Therefore, by Lemma 4.4.7, for all control inputs $u^{[i]}(t^{[i]}_{k,n}) \in \mathcal{U}_{p_{k-1}}(\tilde{x}^{[i]}_n)$, we have $\mathsf{FR}^{[i]}_{k-1}\big(\tilde{x}^{[i]}_n, u^{[i]}(t^{[i]}_{k,n})\big) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$. Based on OCA and ICA, we have

$$\mathcal{X}^{[i]}_{\text{safe},k} = \mathcal{X}_{p_k} \setminus [\cup_{j=0,\cdots,i-1} \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}]. \tag{4.26}$$

It implies that $\mathcal{X}^{[i]}_{\text{safe},k-1} \cap \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j} = \emptyset$ for all $j < i$. By $\mathsf{Discrete}$, this renders

$$\rho(\mathsf{FR}^{[i]}_{k-1}\big(\tilde{x}^{[i]}_n, u^{[i]}(t^{[i]}_{k,n})\big), \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant h_{p_{k-1}},$$

for each $n = 0, 1, \cdots, \bar{n}^{[i]}_k - 1$. Combining this with Lemma 4.4.3, we have, for each $n = 0, 1, \cdots, \bar{n}^{[i]}_k - 1$,

$$\rho(\tilde{\mathsf{FR}}^{[i]}_{k-1}\big(\tilde{x}^{[i]}_n, u^{[i]}(t^{[i]}_{k,n}), p_{k-1}, \epsilon^{[i]}\big), \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant 2h_{p_{k-1}}. \tag{4.27}$$

By Lemma 4.4.2, we have, for each $n = 0, 1, \cdots, \bar{n}_k^{[i]} - 1$,

$$x^{[i]}(t_{k,n}^{[i]}, t_{k,n}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n}^{[i]})) \in G_{\epsilon^{[i]}}^{[i]}(x^{[i]}(t_{k,n}^{[i]}), u^{[i]}(t_{k,n}^{[i]})). \tag{4.28}$$

By definition of $\tilde{x}_n^{[i]}$, we have, for each $n = 0, 1, \cdots, \bar{n}_k^{[i]} - 1$,

$$\rho(x^{[i]}(t_{k,n}^{[i]}), \tilde{x}_n^{[i]}) \leqslant h_{p_{k-1}}. \tag{4.29}$$

Combining (4.27), (4.28) and (4.29) with Lemma 4.4.5, we have

$$\rho(x^{[i]}(t_{k,n}^{[i]}, t_{k,n}^{[i]} + \epsilon^{[i]}, u^{[i]}(t_{k,n}^{[i]})), \bar{\mathcal{X}}_{\text{unsafe},k-1,j}^{[i]}) \geqslant 2h_{p_{k-1}},$$

for each $n = 0, 1, \cdots, \bar{n}_k^{[i]} - 1$. Recall that the definition that $t_{k,0}^{[i]} = k\xi$, $t_{k,n}^{[i]} = t_{k,n-1}^{[i]} + \epsilon^{[i]}$ and $t_{k,\bar{n}_k^{[i]}}^{[i]} = (k+1)\xi$. Hence, the lemma is proved. ■

D.2) The inclusion of $\mathcal{X}_{\textbf{unsafe},k,j}^{[i]}$

Denote $\beta_{k,1}^{[i]} \triangleq 2\xi m^{[i]} + 2\zeta + m^{[i]}\epsilon^{[i]} + 2h_{p_k}$ and $\beta_{k,2}^{[i]} \triangleq \epsilon^{[i]}\gamma\bar{\sigma}_k^{[i]}$. Then in ICA, we have $\mathcal{X}_k^{[i]} = x^{[i]}[k] + \beta_{k,1}^{[i]}\mathcal{B}$ and $\mathcal{X}_{\text{unsafe},k,j}^{[i]} = [\mathcal{X}_k^{[j]} + \beta_{k,2}^{[i]}\mathcal{B}] \cap \mathcal{X}_{p_k}$. Lemma 4.4.13 below renders a monotonic property.

Lemma 4.4.13. It holds that $\beta_{k+1,2}^{[i]} \leqslant \beta_{k,2}^{[i]}$.

Proof: Let $z \in \mathcal{X} \times \mathcal{U}$. Equation (11) in [81] shows that $(\sigma_k^{[i]}(z))^2$ can be expressed recursively as

$$(\sigma_{k+1}^{[i]}(z))^2 = (\sigma_k^{[i]}(z))^2 - (\sigma_k^{[i]}(z))^2 a_{k+1}^{[i]} (\sigma_k^{[i]}(z))^2$$

where $a_{k+1}^{[i]} \geqslant 0$. Hence, $\sigma_{k+1}^{[i]}(z) \leqslant \sigma_k^{[i]}(z)$ and $\bar{\sigma}_{k+1}^{[i]} \leqslant \bar{\sigma}_k^{[i]}$. Then the definition of $\beta_{k,2}^{[i]}$ yields $\beta_{k+1,2}^{[i]} \leqslant \beta_{k,2}^{[i]}$. ■

The following lemma characterizes the inclusion of $\mathcal{X}_{\text{unsafe},k,j}^{[i]}$ in terms of $\mathcal{X}_{\text{unsafe},k-1,j}^{[i]}$.

Lemma 4.4.14. Suppose $\xi \leqslant \frac{h_{p_{\tilde{k}}}}{\max_{j\in\mathcal{V}} m^{[j]}}$ and (4.8) holds for all p_k, $k \geqslant 1$. Suppose Assumption 4.2.1 holds. Then, for all $j < i$, dSLAP renders $\mathcal{X}_{\text{unsafe},k,j}^{[i]} \subset [\mathcal{X}_{\text{unsafe},k-1,j}^{[i]} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k}$.

Proof: Consider $j = 0$. Let $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,0}^{[i]}$. By definition in OCA, it holds that $x^{[i]} \in \mathcal{X}_{p_k}$ and $\rho(x^{[i]}, \mathcal{X}_O) \leqslant m\epsilon^{[i]} + h_{p_k}$. Since Discrete renders $h_{p_{k-1}} = 2h_{p_k}$, there

exists $y^{[i]} \in \mathcal{X}_{p_{k-1}}$ satisfying $\rho(y^{[i]}, x^{[i]}) \leqslant h_{p_k}$. Then by triangular inequality, it holds that $\rho(y^{[i]}, \mathcal{X}_O) \leqslant m\epsilon^{[i]} + 2h_{p_k} = m\epsilon^{[i]} + h_{p_{k-1}}$. This implies $y^{[i]} \in \mathcal{X}^{[i]}_{\text{unsafe},k-1,0}$ and hence $x^{[i]} \in \mathcal{X}^{[i]}_{\text{unsafe},k-1,0} + h_{p_k}\mathcal{B}$. Since $x^{[i]} \in \mathcal{X}_{p_k}$, we further have $\mathcal{X}^{[i]}_{\text{unsafe},k,0} \subset [\mathcal{X}^{[i]}_{\text{unsafe},k-1,0} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k}$.

Consider $j = 1, 2, \cdots, i-1$. Let $x^{[i]} \in \mathcal{X}^{[i]}_{\text{unsafe},k,j}$. By definition in ICA, it holds that $x^{[i]} \in [x^{[j]}[k] + (\beta^{[j]}_{k,1} + \beta^{[i]}_{k,2})\mathcal{B}] \cap \mathcal{X}_{p_k}$. Since Discrete renders $h_{p_{k-1}} = 2h_{p_k}$, we have $\beta^{[i]}_{k,1} < \beta^{[i]}_{k-1,1} - 2h_{p_k}$. Recall that Lemma 4.4.13 renders $\beta^{[i]}_{k,2} \leqslant \beta^{[i]}_{k-1,2}$. Since $\xi \leqslant \frac{h_{p_{\bar{k}}}}{\max_{j\in\mathcal{V}} m^{[j]}}$, it holds that $\rho(x^{[j]}[k], x^{[j]}[k-1]) \leqslant m^{[j]}\xi \leqslant h_{p_{\bar{k}}} \leqslant h_{p_k}$. Combining the above three statements renders

$$\begin{aligned} x^{[i]} &\in x^{[j]}[k] + (\beta^{[j]}_{k,1} + \beta^{[i]}_{k,2})\mathcal{B} \\ &\subset x^{[j]}[k-1] + (\beta^{[j]}_{k,1} + \beta^{[i]}_{k,2} + h_{p_k})\mathcal{B} \\ &\subset x^{[j]}[k-1] + (\beta^{[j]}_{k-1,1} + \beta^{[i]}_{k-1,2} - h_{p_k})\mathcal{B}. \end{aligned}$$

This implies $\rho(x^{[i]}, x^{[j]}[k-1]) \leqslant \beta^{[j]}_{k-1,1} + \beta^{[i]}_{k-1,2} - h_{p_k}$ Similar to the logic above, Discrete renders that there exists $y^{[i]} \in \mathcal{X}_{p_{k-1}}$ satisfying $\rho(y^{[i]}, x^{[i]}) \leqslant h_{p_k}$. Triangular inequality further renders $\rho(y^{[i]}, x^{[j]}[k-1]) \leqslant \beta^{[j]}_{k-1,1} + \beta^{[i]}_{k-1,2}$, or

$$y^{[i]} \in [x^{[j]}[k-1] + (\beta^{[j]}_{k-1,1} + \beta^{[i]}_{k-1,2})\mathcal{B}] \cap \mathcal{X}_{p_{k-1}} = \mathcal{X}^{[i]}_{\text{unsafe},k-1,j}$$

and hence $x^{[i]} \in \mathcal{X}^{[i]}_{\text{unsafe},k-1,j} + h_{p_k}\mathcal{B}$. Since $x^{[i]} \in \mathcal{X}_{p_k}$, we have $x^{[i]} \in [\mathcal{X}^{[i]}_{\text{unsafe},k-1,j} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k}$. ■

D.3) Conditions for $\mathcal{B}(x^{[i]}[k+1], h_{p_k}) \cap \mathcal{X}^{[i]}_{safe,k} \neq \emptyset$

Define a sequence of sets such that $\mathcal{P}^{[i]}_{k,j,1} \triangleq \mathcal{X}^{[i]}_{\text{unsafe},k,j}$, $j \geqslant 1$, and $\mathcal{P}^{[i]}_{k,j,l} \triangleq \{x^{[i]} \in \mathcal{X}_{p_k} \setminus [\cup_{l'\leqslant l-1}\mathcal{P}^{[i]}_{k,j,l'}] \mid \forall u^{[i]} \in \mathcal{U}_{p_k},\ x^{[i]} \in \mathsf{BR}^{[i]}_k(\tilde{x}^{[i]}, u^{[i]})$ for some $\tilde{x}^{[i]} \in [\cup_{l'\leqslant l-1}\mathcal{P}^{[i]}_{k,j,l'}] \bigcup [\cup_{j'\leqslant j-1}\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}]\}$. Lemma 4.4.15 below characterizes $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ using $\mathcal{P}^{(l)}_{k,i,j}$.

Lemma 4.4.15. Suppose Assumption 4.2.1 holds. For all iterations k, it holds that $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j} = \cup_{l=1}^{n^{[i]}_{k,j}}\mathcal{P}^{[i]}_{k,j,l}$ for some $n^{[i]}_{k,j} < \infty$. Furthermore, it holds that $\mathcal{P}^{[i]}_{k,j,l} \neq \emptyset$ for all $l = 1, \cdots, n^{[i]}_{k,j}$ and $\mathcal{P}^{[i]}_{k,j,l} = \emptyset$ for all $l > n^{[i]}_{k,j}$.

Proof: By definition of $\mathcal{P}^{[i]}_{k,j,l}$, if $\mathcal{P}^{[i]}_{k,j,l} = \emptyset$, then

$$\begin{aligned}
\mathcal{P}^{[i]}_{k,j,l+1} &= \{x^{[i]} \in \mathcal{X}_{p_k} \setminus [\cup_{l' \leqslant l} \mathcal{P}^{[i]}_{k,j,l'}] \mid \forall u^{[i]} \in \mathcal{U}_{p_k}, x^{[i]} \in \mathsf{BR}^{[i]}_k(\tilde{x}^{[i]}, u^{[i]}) \text{ for some} \\
&\quad \tilde{x}^{[i]} \in \cup_{l' \leqslant l} \mathcal{P}^{[i]}_{k,j,l'} \bigcup [\cup_{j' \leqslant j-1} \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}]\} \\
&= \{x^{[i]} \in \mathcal{X}_{p_k} \setminus [\cup_{l' \leqslant l-1} \mathcal{P}^{[i]}_{k,j,l'}] \mid \forall u^{[i]} \in \mathcal{U}_{p_k}, x^{[i]} \in \mathsf{BR}^{[i]}_k(\tilde{x}^{[i]}, u^{[i]}) \text{ for some} \\
&\quad \tilde{x}^{[i]} \in \cup_{l' \leqslant l-1} \mathcal{P}^{[i]}_{k,j,l'} \bigcup [\cup_{j' \leqslant j-1} \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}]\} \\
&= \mathcal{P}^{[i]}_{k,j,l} = \emptyset.
\end{aligned}$$

Therefore, we have $\mathcal{P}^{[i]}_{k,j,l} = \emptyset$ for all $l > l'$. The definition indicates that $\mathcal{P}^{[i]}_{k,j,l''}$ and $\cup_{l=1}^{l'} \mathcal{P}^{[i]}_{k,j,l}$ are mutually disjoint for any $l'' > l'$. Hence $n^{[i]}_{k,j}$ is finite since $\mathcal{X}_{p_k}$ is finite due to the compactness of $\mathcal{X}$ in Assumption 4.2.1, and $\mathcal{P}^{[i]}_{k,j,l} \neq \emptyset$, $\forall l = 1, \cdots, n^{[i]}_{k,j}$ and $\mathcal{P}^{[i]}_{k,j,l} = \emptyset$ $\forall l > n^{[i]}_{k,j}$.

Now we show $\cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$. For $j = 0, 1, \cdots, i-1$, according to the UnsafeUpdate procedure, we have $\mathcal{P}^{[i]}_{k,j,1} = \mathcal{X}^{[i]}_{\text{unsafe},k,j} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$. For any $x^{[i]}$ in non-empty $\mathcal{P}^{[i]}_{k,j,l'}$, $l' > 1$, since $x^{[i]} \in \mathsf{BR}^{[i]}_k(\tilde{x}^{[i]}, u)$ for some $\tilde{x}^{[i]} \in \cup_{l=1}^{l'-1} \mathcal{P}^{(l)}_{k,i,j}$ for all control inputs $u^{[i]} \in \mathcal{U}_{p_k}$, it renders that $\mathcal{U}^{[i]}_{p_k}(x^{[i]}) = \emptyset$, and hence $x^{[i]} \in \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ according to UnsafeUpdate. Therefore, we have $\mathcal{P}^{[i]}_{k,j,l} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ for all non-empty $\mathcal{P}^{[i]}_{k,j,l}$, $l = 1, \cdots, n^{[i]}_{k,j}$. This shows $\cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$.

We show $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j} \subset \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}$ using contradiction. Suppose there exists a state $x^{(1)} \in \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ and $x^{(1)} \notin \mathcal{P}^{[i]}_{k,j,l}$ for all $\mathcal{P}^{[i]}_{k,j,l}$, $l = 1, \cdots, n^{[i]}_{k,j}$.

Obviously, we have $x^{(1)} \notin \mathcal{X}^{[i]}_{\text{unsafe},k,j}$ because otherwise $x^{(1)} \in \mathcal{P}^{(1)}_{k,i,j}$ according to the definition of $\mathcal{P}^{[i]}_{k,j,1}$, $j \geqslant 0$. Then $x^{(1)}$ can only be added to $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ by UnsafeUpdate. Then there exists $x^{(2)} \in \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ such that $x^{(1)} \in \mathsf{BR}^{[i]}_k(x^{(2)}, u^{[i]})$, which reduces the control set $\mathcal{U}^{[i]}_{p_k}(x^{(1)})$ to an empty set and leads to the addition of $x^{(1)}$ to $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,0}$. Furthermore, we also have $x^{(2)} \notin \mathcal{P}^{[i]}_{k,j,l}$ for any $l = 1, \cdots, n^{[i]}_{k,j} - 1$ since otherwise we have $x^{(1)} \in \mathcal{P}^{[i]}_{k,j,l+1}$. By induction, we have a set $\{x^{(l)}\}_{l=1}^{n_l} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$ but $\{x^{(l)}\}_{l=1}^{n_l} \not\subset \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}$ for any $n_l = 1, \cdots, |\mathcal{X}_{p_k}|$. However, this is impossible because $\mathcal{X}_{p_k}$ is finite and $\cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l} \supset \mathcal{P}^{(1)}_{k,i,j} \neq \emptyset$. This gives $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j} \subset \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}$. ■

The following lemma shows that the robot after one iteration is near the safe states under the priority assignment in the previous iteration.

Lemma 4.4.16. Suppose Assumption 4.2.1 holds. Suppose $4\gamma\bar{\sigma}\epsilon^{[i]} \leqslant h_{p_{\tilde{k}}}$ and (4.8) holds for all $x^{[i]} \in \mathcal{X}_{p_k}$, $u^{[i]} \in \mathcal{U}_{p_k}$, $k \geqslant 1$. Suppose $\xi \leqslant \frac{h_{p_{\tilde{k}}}}{\max_{j\in\mathcal{V}} m^{[j]}}$. For each $i \in \mathcal{V}$, if $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{safe,k-1} \neq \emptyset$, $k \geqslant 1$, then $\mathcal{B}(x^{[i]}[k+1], h_{p_k}) \cap \mathcal{X}^{[i]}_{\text{safe},k} \neq \emptyset$.

Proof: Let $j = 0, 1, \cdots, i-1$. Lemma 4.4.12 shows that $\rho(x^{[i]}[k+1], \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant 2h_{p_{k-1}}$. By (4.26), this implies that $\mathcal{B}(x^{[i]}[k+1], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1} \neq \emptyset$. Let $\tilde{x}^{[i]} \in \mathcal{B}(x^{[i]}[k+1], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{\text{safe},k-1}$. Then Lemma 4.4.7 renders that there exists $\tilde{u}^{[i]} \in \mathcal{U}^{[i]}_{p_{k-1}}(\tilde{x}^{[i]})$ such that $\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}, \tilde{u}^{[i]}) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$. By Discrete and (4.26), this implies that, for all $j \in [0, i)$,

$$\rho(\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}, \tilde{u}^{[i]}), \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant h_{p_{k-1}} = 2h_{p_k}. \tag{4.30}$$

Based on the UnsafeUpdate procedure, it is obvious that $\mathcal{X}^{[i]}_{\text{unsafe},k,j} \subset \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}$. Then (4.30) renders

$$\rho(\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}, \tilde{u}^{[i]}), \mathcal{X}^{[i]}_{\text{unsafe},k-1,j}) \geqslant h_{p_{k-1}} = 2h_{p_k}. \tag{4.31}$$

Consider the rewriting in (4.11) and recall that $\mathsf{FR}^{[i]}_{k-1}(\tilde{x}^{[i]}, \tilde{u}^{[i]}) \subset \mathcal{X}_{p_{k-1}}$ and $\mathcal{X}^{[i]}_{\text{unsafe},k-1,j} \subset \mathcal{X}_{p_{k-1}}$. Then (4.31) implies

$$\rho(\mathcal{B}^{[i]}_{\tilde{x}^{[i]},\tilde{u}^{[i]},k-1}, \mathcal{X}^{[i]}_{\text{unsafe},k-1,j}) > 0. \tag{4.32}$$

Due to the Discrete procedure, there exists $y^{[i]} \in \mathcal{X}_{p_k}$ such that $\rho(x^{[i]}[k+1], y^{[i]}) \leqslant h_{p_k}$ and $\rho(\tilde{x}^{[i]}, y^{[i]}) \leqslant h_{p_k}$. Then combining (4.32) with Lemma 4.4.6 renders

$$\rho(\mathcal{B}^{[i]}_{y^{[i]},\tilde{u}^{[i]},k}, \mathcal{X}^{[i]}_{\text{unsafe},k-1,j}) > h_{p_k}. \tag{4.33}$$

Recall that Lemma 4.4.14 renders

$$\mathcal{X}^{[i]}_{\text{unsafe},k,j} \subset [\mathcal{X}^{[i]}_{\text{unsafe},k-1,j} + h_{p_k}\mathcal{B}] \cap \mathcal{X}_{p_k}. \tag{4.34}$$

Then combining (4.33) and (4.34) renders

$$\rho(\mathcal{B}^{[i]}_{y^{[i]},\tilde{u}^{[i]},k}, \mathcal{X}^{[i]}_{\text{unsafe},k,j}) > 0. \tag{4.35}$$

Note that (4.35) holds for all $j < i$. Then it follows that

$$\mathsf{FR}_k^{[i]}(y^{[i]}, \tilde{u}^{[i]}) \cap [\cup_{j<i} \mathcal{X}_{\text{unsafe},k,j}^{[i]}] = \emptyset. \tag{4.36}$$

Combining the claim below with (4.26) renders that $y^{[i]} \in \mathcal{X}_{\text{safe},k}^{[i]}$. Combining this with $\rho(x^{[i]}[k+1], y^{[i]}) \leqslant h_{p_k}$ concludes the proof. ■

Claim 4.4.16.1. It holds that $y^{[i]} \notin [\cup_{j<i} \bar{\mathcal{X}}_{\text{unsafe},k,j}^{[i]}]$.

Proof of Claim 4.4.16.1: The proof of the claim is composed of three parts.

Part (i). We show that

$$y^{[i]} \notin [\cup_{j<i} \mathcal{X}_{\text{unsafe},k,j}^{[i]}]. \tag{4.37}$$

Let $j = 0, 1, \cdots, i-1$. Since $\tilde{x}^{[i]} \in \mathcal{X}_{\text{safe},k-1}^{[i]}$, by (4.26) and Discrete, it renders that $\rho(\tilde{x}^{[i]}, \bar{\mathcal{X}}_{\text{unsafe},k-1,j}^{[i]}) \geqslant h_{p_{k-1}}$. According to the construction of $\bar{\mathcal{X}}_{\text{unsafe},k-1,j}^{[i]}$ through UnsafeUpdate, it holds that $\mathcal{X}_{\text{unsafe},k-1,j}^{[i]} \subset \bar{\mathcal{X}}_{\text{unsafe},k-1,j}^{[i]}$. Combining the above two statements renders

$$\rho(\tilde{x}^{[i]}, \mathcal{X}_{\text{unsafe},k-1,j}^{[i]}) \geqslant h_{p_{k-1}} = 2h_{p_k}. \tag{4.38}$$

Next we show that $y^{[i]} \notin \mathcal{X}_{\text{unsafe},k,j}^{[i]}$ through two cases.

Case 1: $j = 0$. By construction of $\mathcal{X}_{\text{unsafe},k,0}^{[i]}$ in OCA, it holds that $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,0}^{[i]}$ if and only if $\rho(x^{[i]}, \mathcal{X}_O) \leqslant m\epsilon^{[i]} + h_{p_k}$. Combining this with (4.38) renders $\rho(\tilde{x}^{[i]}, \mathcal{X}_O) \geqslant m\epsilon^{[i]} + 2h_{p_{k-1}}$. Recall that $h_{p_{k-1}} = 2h_{p_k}$. Since $\rho(\tilde{x}^{[i]}, y^{[i]}) \leqslant h_{p_k}$, triangular inequality further gives $\rho(y^{[i]}, \mathcal{X}_O) \geqslant m\epsilon^{[i]} + 3h_{p_k}$. Hence $y^{[i]} \notin \mathcal{X}_{\text{unsafe},k,0}^{[i]}$.

Case 2: $j = 1, \cdots, i-1$. Recall the definitions of $\beta_{k,1}^{[i]}$ and $\beta_{k,2}^{[i]}$ above Lemma 4.4.13. By construction of $\mathcal{X}_{\text{unsafe},k,j}^{[i]}$ in ICA, it holds that $x^{[i]} \in \mathcal{X}_{\text{unsafe},k,j}^{[i]}$ if and only if $\rho(x^{[i]}, x^{[j]}[k]) \leqslant \beta_{k,1}^{[i]} + \beta_{k,2}^{[i]}$. Combining this with (4.38) renders

$$\rho(\tilde{x}^{[i]}, x^{[j]}[k-1]) \geqslant \beta_{k-1,1}^{[i]} + \beta_{k-1,2}^{[i]} + h_{p_{k-1}}.$$

Since $h_{p_{k-1}} = 2h_{p_k}$, we have $\beta_{k,1}^{[i]} < \beta_{k-1,1}^{[i]} - 2h_{p_k}$. Recall that Lemma 4.4.13 renders $\beta_{k,2}^{[i]} \leqslant \beta_{k-1,2}^{[i]}$. Then combining the above three statements renders

$$\rho(\tilde{x}^{[i]}, x^{[j]}[k-1]) \geqslant \beta_{k,1}^{[i]} + \beta_{k,2}^{[i]} + 4h_{p_k}.$$

Since $\xi \leqslant \frac{h_{p_{\bar{k}}}}{\max_{j\in\mathcal{V}} m^{[j]}}$, it holds that $\rho(x^{[j]}[k-1], x^{[j]}[k]) \leqslant \xi m^{[j]} \leqslant h_{p_k}$. Combining the above two statements with triangular inequality renders

$$\rho(\tilde{x}^{[i]}, x^{[j]}[k]) \geqslant \beta_{k,1}^{[i]} + \beta_{k,2}^{[i]} + 3h_{p_k}.$$

Since $\rho(\tilde{x}^{[i]}, y^{[i]}) \leqslant h_{p_k}$, triangular inequality further gives

$$\rho(y^{[i]}, x^{[j]}[k]) \geqslant \beta_{k,1}^{[i]} + \beta_{k,2}^{[i]} + 2h_{p_k}.$$

Hence $y^{[i]} \notin \mathcal{X}_{\text{unsafe},k,j}^{[i]}$. The proof of Part (i) is concluded.

Part (ii). Consider a sequence of states and control inputs pairs $\{(x_{p_k,n}^{[i]}, x_{p_{k-1},n}^{[i]}, u_{p_k,n}^{[i]})\}$, $n = 0, 1, 2, \cdots$, where

$$\begin{aligned} &x_{p_k,0}^{[i]} \triangleq y^{[i]},\ x_{p_{k-1},0}^{[i]} \triangleq \tilde{x}^{[i]},\ u_{p_k,0}^{[i]} \triangleq \tilde{u}^{[i]},\ u_{p_k,n}^{[i]} \in \mathcal{U}_{p_{k-1}}, \\ &x_{p_k,n+1}^{[i]} \in \mathsf{FR}_k^{[i]}(x_{p_k,n}^{[i]}, u_{p_k,n}^{[i]}),\ \rho(x_{p_k,n}^{[i]}, x_{p_{k-1},n}^{[i]}) \leqslant h_{p_k}. \end{aligned}$$

We use induction to show that, for all $n = 0, 1, 2, \cdots$,

$$\mathsf{FR}_k^{[i]}(x_{p_k,n}^{[i]}, u_{p_k,n}^{[i]}) \cap [\cup_{j<i} \mathcal{X}_{\text{unsafe},k,j}^{[i]}] = \emptyset \tag{4.39a}$$

$$\mathsf{FR}_{k-1}^{[i]}(x_{p_{k-1},n}^{[i]}, u_{p_k,n}^{[i]}) \cap [\cup_{j<i} \bar{\mathcal{X}}_{\text{unsafe},k-1,j}^{[i]}] = \emptyset. \tag{4.39b}$$

The base case $n = 0$ is obvious by (4.36) and (4.30) as well as the definitions of $x_{p_k,0}^{[i]}$, $x_{p_{k-1},0}^{[i]}$ and $u_{p_k,0}^{[i]}$.

Now consider (4.39) holds until $n = m$. By the definition of $\mathsf{FR}_k^{[i]}$ and recall the rewriting in (4.11), it holds that

$$\mathcal{B}_{x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]}, k}^{[i]} \supset x_{p_k,m}^{[i]} + \epsilon^{[i]}(f^{[i]}(x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]}) + \mu_k^{[i]}(x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]})) + 2h_{p_k}\mathcal{B}.$$

Recall that $\mathsf{Discrete}$ renders $h_{p_{k-1}} = 2h_{p_k}$. Therefore, it holds that $\mathcal{B}_{x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]}, k}^{[i]} \cap \mathcal{X}_{p_{k-1}} \neq \emptyset$, and for every $x_{p_k,m+1}^{[i]} \in \mathsf{FR}_k^{[i]}(x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]})$, there exists $x_{p_{k-1},m+1}^{[i]} \in \mathcal{X}_{p_{k-1}}$ satisfying $\rho(x_{p_k,m+1}^{[i]}, x_{p_{k-1},m+1}^{[i]}) \leqslant h_{p_k}$. Lemma 4.4.6 renders $\mathcal{B}_{x_{p_k,m}^{[i]}, u_{p_k,m}^{[i]}, k}^{[i]} + h_{p_k}\mathcal{B} \subset$

$\mathcal{B}^{[i]}_{x^{[i]}_{p_{k-1},m},u^{[i]}_{p_k,m},k-1}$. Combining the above two statements renders

$$x^{[i]}_{p_{k-1},m+1} \in [\mathcal{B}^{[i]}_{x^{[i]}_{p_{k-1},m},u^{[i]}_{p_k,m},k-1} \cap \mathcal{X}_{p_{k-1}}] = \mathsf{FR}^{[i]}_{k-1}(x^{[i]}_{p_{k-1},m}, u^{[i]}_{p_k,m}).$$

Then it follows from the induction hypothesis that

$$\mathsf{FR}^{[i]}_{k-1}(x^{[i]}_{p_{k-1},m}, u^{[i]}_{p_k,m}) \cap [\cup_{j<i} \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}] = \emptyset$$

and hence $x^{[i]}_{p_{k-1},m+1} \in \mathcal{X}^{[i]}_{\text{safe},k-1}$. Furthermore, it follows from Lemma 4.4.7 that there exists $u^{[i]}_{p_{k-1},m+1} \in \mathcal{U}_{p_{k-1}}$ such that $\mathsf{FR}^{[i]}_{k-1}(x^{[i]}_{p_{k-1},m+1}, u^{[i]}_{p_{k-1},m+1}) \subset \mathcal{X}^{[i]}_{\text{safe},k-1}$, which implies

$$\rho(\mathsf{FR}^{[i]}_{k-1}(x^{[i]}_{p_{k-1},m+1}, u^{[i]}_{p_{k-1},m+1}), \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k-1,j}) \geqslant h_{p_{k-1}}.$$

Note that $\mathsf{Discrete}$ renders $\mathcal{U}_{p_{k-1}} \subset \mathcal{U}_{p_k}$. Hence, we can set $u^{[i]}_{p_k,m+1} = u^{[i]}_{p_{k-1},m+1} \in \mathcal{U}_{p_k}$. Then following the same logic of (4.30) to (4.36) by replacing $\tilde{x}^{[i]}$ with $x^{[i]}_{p_{k-1},m+1}$, $\tilde{u}^{[i]}$ with $u^{[i]}_{p_{k-1},m+1}$, and $y^{[i]}$ with $x^{[i]}_{p_k,m+1}$, we have

$$\mathsf{FR}^{[i]}_{k}(x^{[i]}_{p_k,m+1}, u^{[i]}_{p_k,m+1}) \cap [\cup_{j<i} \mathcal{X}^{[i]}_{\text{unsafe},k,j}] = \emptyset.$$

The induction is completed.

Part (iii). Recall the definition of $\mathcal{P}^{[i]}_{k,j,l}$. Next we use induction to show that

$$x^{[i]}_{p_k,n} \notin \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}, \quad \forall j = 0, 1, \cdots, i-1 \text{ and } n = 0, 1, \cdots. \tag{4.40}$$

Consider the base case $j = 0$ and all $n = 0, 1, 2, \cdots$. By (4.37) and (4.39a), we have $x^{[i]}_{p_k,n} \notin \mathcal{P}^{[i]}_{k,0,1}$. By (4.39a), we have $\mathsf{FR}^{[i]}_{k}(x^{[i]}_{p_k,n}, u^{[i]}_{p_k,n}) \cap \mathcal{X}^{[i]}_{\text{unsafe},k,0}] = \emptyset$. Hence, we have $x^{[i]}_{p_k,n} \notin \mathcal{P}^{[i]}_{k,0,2}$. Since $x^{[i]}_{p_k,n} \in \mathsf{FR}^{[i]}_{k}(x^{[i]}_{p_k,n-1}, u^{[i]}_{p_k,n-1})$, we have $x^{[i]}_{p_k,n-1} \notin \mathcal{P}^{[i]}_{k,0,3}$. Following the same logic, it follows that $x^{[i]}_{p_k,n} \notin \mathcal{P}^{[i]}_{k,0,l}$ for all $l = 1, \cdots, n^{[i]}_{k,0}$. This renders $x^{[i]}_{p_k,n} \notin \cup_{l=1}^{n^{[i]}_{k,0}} \mathcal{P}^{[i]}_{k,0,l}$.

Now suppose (4.40) holds until $j = \tilde{j} < i - 1$. By (4.37) and (4.39a), we have $x^{[i]}_{p_k,n} \notin \mathcal{P}^{[i]}_{k,\tilde{j}+1,1}$. Since the induction hypothesis renders $x^{[i]}_{p_k,n} \notin \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}$ for all $j \leqslant \tilde{j}$ and (4.39a) implies $\mathsf{FR}^{[i]}_{k}(x^{[i]}_{p_k,n}, u^{[i]}_{p_k,n}) \cap \mathcal{X}^{[i]}_{\text{unsafe},k,\tilde{j}+1}] = \emptyset$, it renders

$x^{[i]}_{p_k,n} \notin \mathcal{P}^{[i]}_{k,\tilde{j}+1,2}$ for all $n = 0, 1, 2, \cdots$. By similar logic, we have $x^{[i]}_{p_k,n-1} \notin \mathcal{P}^{[i]}_{k,\tilde{j}+1,3}$ and hence $x^{[i]}_{p_k,n} \notin \cup_{l=1}^{n^{[i]}_{k,\tilde{j}}} \mathcal{P}^{[i]}_{k,\tilde{j}+1,l}$. This concludes the proof of (4.40).

Since $y^{[i]} = x^{[i]}_{p_k,0}$ and Lemma 4.4.15 renders $\bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j} = \cup_{l=1}^{n^{[i]}_{k,j}} \mathcal{P}^{[i]}_{k,j,l}$, (4.40) renders $y^{[i]} \notin [\cup_{j<i} \bar{\mathcal{X}}^{[i]}_{\text{unsafe},k,j}]$. □

D.4) Proof of Theorem 4.3.3

Given (4.8) holds for all $x^{[i]} \in \mathcal{X}_{p_k}$, $u^{[i]} \in \mathcal{U}_{p_k}$, $k \geqslant 1$, Lemma 4.4.16 implies that if $\mathcal{B}(x^{[i]}[k], h_{p_{k-1}}) \cap \mathcal{X}^{[i]}_{safe,k-1} \neq \emptyset$, for some $k > 1$, then it holds that $\mathcal{B}(x^{[i]}[k'], h_{p_{k'-1}}) \cap \mathcal{X}^{[i]}_{safe,k'-1} \neq \emptyset$ for all $k' \geqslant k$. Then Lemma 4.4.11 implies that $x^{[i]}_q(t) \in \mathcal{X}^{[i]}_F(x^{[\neg i]}_q(t))$, $\forall t \in [k'\xi, k'\xi + \xi)$, $k' \geqslant k$; i.e., $\forall t \geqslant k\xi$.

Note that (4.2) renders that $\mathcal{X}_p \subset \mathcal{X}_{p'}$ and $\mathcal{U}_p \subset \mathcal{U}_{p'}$ $\forall p < p'$, and dSLAP renders that $p_k \leqslant p_{\tilde{k}}$ $\forall k \geqslant 1$. Hence, the definition of $\delta_{\gamma,p}$ above (4.8) implies $\delta_{\gamma,p_{\tilde{k}}} \geqslant \delta_{\gamma,p_k}$, $\forall k \geqslant 1$. Therefore, for each $i \in \mathcal{V}$, for each $k \geqslant 1$, (4.8) holds for all $x^{[i]} \in \mathcal{X}_{p_k}$, $u^{[i]} \in \mathcal{U}_{p_k}$, with probability at least $1 - \delta_{\gamma,p_k} \geqslant 1 - \delta_{\gamma,p_{\tilde{k}}}$. Denote $E^{[i]}_k$ as the event of (4.8) being violated for some $x^{[i]} \in \mathcal{X}_{p_{\tilde{k}}}$ and/or $u^{[i]} \in \mathcal{U}_{p_{\tilde{k}}}$ at iteration k by robot i. Then we have $Pr\{E^{[i]}_k\} \leqslant \delta_{\gamma,p_k}$. Applying the union bound (Theorem 2-3, [88]) renders that $Pr\{\cup_{k=1}^{\tilde{k}} E^{[i]}_k\} \leqslant \tilde{k}\delta_{\gamma,p_{\tilde{k}}}$. Note that $Pr\{\cap_{k=1}^{\tilde{k}} E^{[i]}_k\} = 1 - Pr\{\cup_{k=1}^{\tilde{k}} E^{[i]}_k\}$. Hence, we have (4.8) holds $\forall k \in \{1, \cdots, \tilde{k}\}$ with probability at least $1 - \tilde{k}\delta_{\gamma,p_{\tilde{k}}}$. Further applying the union bound renders that (4.8) holds $\forall k \in \{1, \cdots, \tilde{k}\}$ and $i \in \mathcal{V}$ with probability at least $1 - \tilde{k}|\mathcal{V}|\delta_{\gamma,p_{\tilde{k}}}$. ■

4.5 Simulation

In this section, we conduct a set of Monte Carlo simulations to evaluate the performance of the dSLAP algorithm. The simulations are run in Python, Linux Ubuntu 18.04 on an Intel Xeon(R) Silver 4112 CPU, 2.60 GHz with 32 GB of RAM.

Simulation scenarios. We evaluate the dSLAP algorithm using Zermelo's navigation problem [128] in a 2D space under the following scenario: A group of robots are initially placed evenly on the plane and switch their positions at the destinations. The robots are immediately retrieved once they reach the goals. This example is also used in [129] [130] to demonstrate complicated multi-robot coordination scenarios.

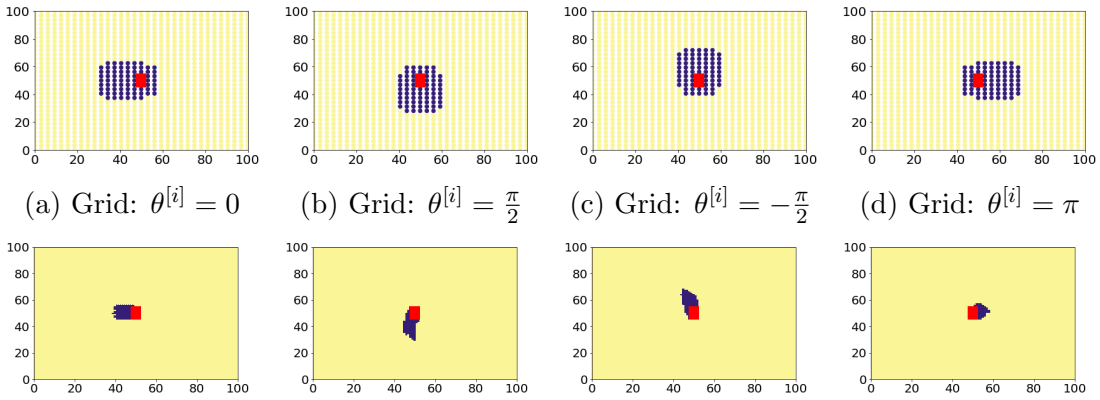
Dynamic models. Consider constant-speed boat robots with length $L = 1.5$ meters (m) moving at speed $v = 0.5$ meters/seconds (m/s). For each robot i, let $x_{q,1}^{[i]}$ and $x_{q,2}^{[i]}$ be the x and y coordinates on a 2D plane, $x_r^{[i]}$ be the angle between the heading and the x-axis, and $u^{[i]}$ be the steering angle. The state space is given by $\mathcal{X} = [0, 100] \times [0, 100] \times [-\pi, \pi]$. External wind disturbance ν is applied at $x_{q,1}^{[i]}$ such that the system dynamics has the following form: $\dot{x}_{q,1}^{[i]}(t) = 0.5 \cos x_r^{[i]}(t)$, $\dot{x}_{q,2}^{[i]}(t) = 0.5 \sin x_r^{[i]}(t)$, $\dot{x}_r^{[i]}(t) = \frac{0.5}{1.5} \tan u^{[i]}(t)$. The control $u^{[i]}$ takes discrete values and the control space is $\mathcal{U} = \{\pm 0.3\pi, \pm 0.15\pi, 0\}$.

Parameters. The kernel of GPR is configured as $\kappa(z, z') = 0.0025 \exp(-\frac{\|z-z'\|_2^2}{2})$, which is 0.0025 times the RBF kernel in the sklearn library. The factor 0.0025 is selected such that the supremum of the predictive standard deviation is 0.05, or 10% of the robots' speed. This can be selected based on the prior knowledge of the variability of the disturbance. Other parameters are selected as $\gamma = 1$, $p_{init} = 4$, $\bar{k} = 200$, $\bar{\tau} = 20$, $\xi = 8$, $q = 2$, $\psi = 1$, $\delta = 0.1$, and $r_k^{[i]} = -\sigma_k^{[i]}$, which are determined according to the desired learning confidence level and the computation capability of the robots. To prevent prolonged computation due to unnecessarily fine discretization, we set a maximum such that $p_{k+1} \leftarrow \min\{p_k, 5\}\ \forall k \geqslant 1$.

Random wind fields with different magnitudes. We randomly generate 2D spatial wind fields, with average speed ν in different ratios of the robots' speed, i.e., $\nu = r_w v$, $r_w > 0$, and standard deviation 2% of the robots' speed, using the Von Karman power spectral density function as described in [131]. This wind model is used to test multi-robot navigation in [131] [132]. A sample with $r_w = 0.2$ is shown in Figure 4.4a. We randomly generate 60 different wind fields for each $r_w \in \{0.1, 0.2, \cdots, 1\}$.

4.5.1 Safe grid vs. safe region

To visualize how safety is guaranteed by dSLAP, in this section, we compare the safe grids $\mathcal{X}_{\text{safe},k}^{[i]}$ under the wind field in Figure 4.4a with the corresponding safe regions, which are the set of initial states that render safe arrival by applying the control policy returned. The comparison is similar for other wind fields. Since the dynamics of the robots has three dimensions, for simplicity of visualization, we only show the screen- shots of the 2D grids/regions with four different heading



(a) Grid: $\theta^{[i]} = 0$ (b) Grid: $\theta^{[i]} = \frac{\pi}{2}$ (c) Grid: $\theta^{[i]} = -\frac{\pi}{2}$ (d) Grid: $\theta^{[i]} = \pi$

(e) Region: $\theta^{[i]} = 0$ (f) Region: $\theta^{[i]} = \frac{\pi}{2}$ (g) Region: $\theta^{[i]} = -\frac{\pi}{2}$ (h) Region: $\theta^{[i]} = \pi$

Figure 4.3: Safe grid computed by dSLAP vs. actual safe region

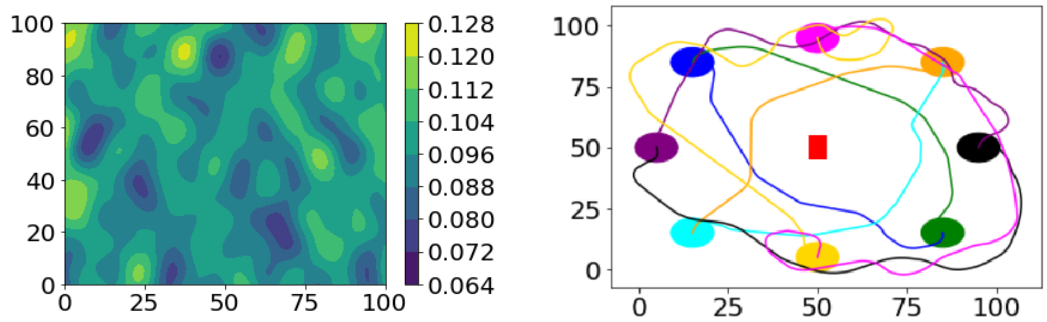
angles (i.e., $\theta^{[i]} = 0, \frac{\pi}{2}, -\frac{\pi}{2}$ and π). For the simplicity of illustration, we only compare the safe grids $\mathcal{X}^{[i]}_{\text{safe},k}$ of robot i, in the presence of only one obstacle, with the corresponding safe regions. Since the figures are similar for grids with different resolutions, due to space limitation, we only show the comparison for safe grids $\mathcal{X}^{[i]}_{\text{safe},k}$ in one resolution, i.e., $p_k = 5$. In each figure, the safe region is approximated by 10,000 evenly distributed initial states.

Figure 4.3 shows the comparison for iteration $k = 1$, where the unsafe grids/regions are dark-colored and the safe grids/regions are light-colored. We can see that the safe grids are strictly subsets of the safe regions, which verifies the sufficient condition in Theorem 4.3.2 and Theorem 4.3.3, where the robots are guaranteed to be safe if $\mathcal{B}(x^{[i]}(0), h_{p_0}) \cap \mathcal{X}^{[i]}_{\text{safe},0} \neq \emptyset$. Furthermore, by comparing Figures 4.3a - 4.3d with Figures 4.3e - 4.3h, we can see that the identification of safe grids by dSLAP and the safety guarantees by Theorem 4.3.2 and Theorem 4.3.3 are conservative. This is due to the fact that the safe control inputs are spatially similar, and hence some unsafe states labeled by dSLAP can still be safe by applying the control inputs on the nearest safe states. The conservativeness comes from the over-approximation of the one-step forward reachability sets in $\mathsf{FR}^{[i]}_k$ and the errors in discretization. Nevertheless, the conservativeness can be reduced by refining the discretization at the expense of more computation power.

4.5.2 Multi-robot maneuver.

We evaluate the dSLAP algorithm using 30,000 scenarios generated as follows.

Different initial configurations. We deploy n robots with 10 different initial configurations in the simulation, where $n \in \{1, 2, 4, 6, 8\}$. Figure 4.4b shows one configuration of 8 robots' initial states and goal regions, and the corresponding trajectories under dSLAP in the wind field in Figure 4.4a. The circular disks are the goal regions of the robots and the red rectangle is the static obstacle. Other configurations are generated by different permutations and removals of robots from that in Figure 4.4b.



(a) A sample of wind field experienced by the robots

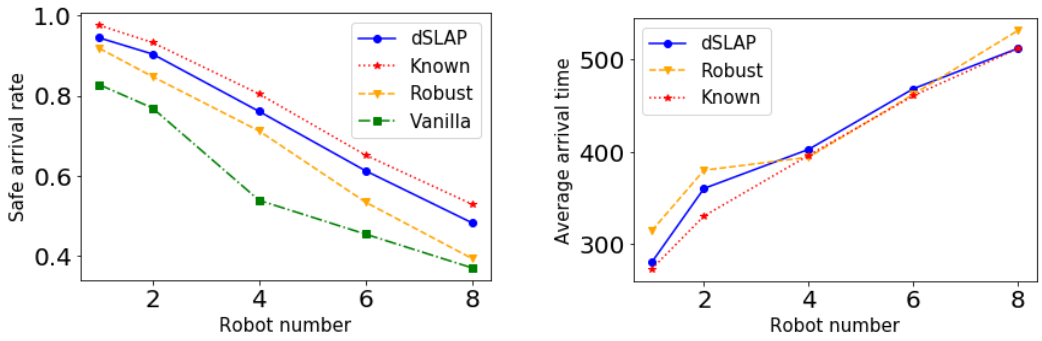
(b) Trajectories of the robots

Figure 4.4: A sample of wind fields and robot trajectories

Ablation study. To the best of our knowledge, this chapter is the first to consider multi-robot motion planning coupled with online learning. Hence, we compare dSLAP with its three variants, Vanilla, Robust and Known, that do not learn the wind disturbances. Vanilla assumes $g^{[i]} = 0$, $\forall i \in \mathcal{V}$, whereas Robust assumes $\sup_{x^{[i]} \in \mathcal{X}, u^{[i]} \in \mathcal{U}} |g^{[i]}(x^{[i]}, u^{[i]})| \leqslant \hat{r}_w v$ and thus $\dot{x}^{[i]} \in f^{[i]}(x^{[i]}, u^{[i]}) + \hat{r}_w v \mathcal{B}$, where $\hat{r}_w > 0$. We adopt $\hat{r}_w = 0.1$ such that Robust has the same level of conservativeness as dSLAP before collecting any data. The benchmark Known is obtained by running the dSLAP framework with the disturbances exactly known, which is the control law obtained by dSLAP when the amount of data of $g^{[i]}$ goes to infinity.

Results. The average safe arrival rates of dSLAP, Robust, Vanilla and Known among the 30,000 cases are shown in Figure 4.5. From Figure 4.5a, we can see that dSLAP's performance is superior to those of Robust and Vanilla. This is due to the fact that dSLAP online learns about the unknown disturbances and adjusts

the policies accordingly. On the other hand, Robust (or Vanilla) only captures part of (or none of) the disturbances through the prior estimates, which can be unsafe when the disturbances exceed the estimates. Furthermore, we can observe that the safe arrival rate for dSLAP decreases linearly with respect to the number of robots. This corresponds to the probability with respect to the number of robots in Theorems 4.3.2 and 4.3.3. Notice that the gap between Known and dSLAP is small. The cases that are unsafe even in Known are due to the robots being too close to each other or the magnitude of the disturbances being too large to tolerate (note that the magnitude of the disturbances can be as large as the speed of the robots in the simulation). This indicates that dSLAP enables safe arrival in most feasible cases.



(a) Percentage of safe arrivals (b) Average time of safe arrivals

Figure 4.5: Ablation study of dSLAP

Arrival time. Figure 4.5b compares the average safe arrival times among dSLAP, Robust and Known. We exclude the comparison with Vanilla since its safe arrival rate is far lower than the other three while safety is this chapter's top priority. The arrival times of the three algorithms are comparable. This indicates dSLAP improves safe arrival rate without sacrificing arrival time, i.e., being more conservative.

4.5.3 Run-time computation.

This section shows the wall computation time of dSLAP when the robots are deployed in the wind field in Figure 4.4a with configuration in Figure 4.4b, as an example. Table 4.1 presents the average plus/minus one standard deviation

ID	Total time	SL		Discrete+OCA	
		time	Percentage	time	Percentage
1	5.71±0.45	$0.84\pm7.87e^{-3}$	14.73±1.08	4.26±0.12	74.91±3.95
2	5.93±0.47	0.81±0.01	13.70±1.03	4.21±0.06	71.32±5.10
3	5.69±0.34	$0.82\pm1.43e^{-3}$	14.48±0.88	4.14±0.04	73.10±3.90
4	5.18±0.14	$0.82\pm2.03e^{-3}$	15.85±0.43	4.01±0.04	77.37±1.98
5	5.90±0.35	0.82±0.01	13.96±0.65	4.30±0.08	73.10±3.41
6	5.66±1.29	0.88±0.16	15.7±0.65	4.55±1.08	80.26±1.58
7	5.94±0.99	0.88±0.10	14.96±0.82	4.49±0.65	75.85±2.69
8	5.94±1.14	0.88±0.13	14.90±0.55	4.65±0.83	78.53±1.30

ID	ICA		AL	
	time	Percentage	time	Percentage
1	$6.03e^{-3}\pm1.85e^{-3}$	$0.11\pm9.44e^{-3}$	0.61±0.32	10.26±5.07
2	0.06±0.03	1.08±0.59	0.85±0.41	13.90±6.38
3	0.14±0.01	2.56±0.32	0.58±0.32	9.87±5.08
4	0.16±0.09	3.07±1.76	0.20±0.15	3.70±2.81
5	0.23±0.03	3.96±0.51	0.54±0.27	8.98±4.10
6	0.13±0.11	2.57±2.23	0.10±0.13	1.46±1.64
7	0.19±0.10	3.39±2.03	0.38±0.38	5.79±5.19
8	0.29±0.07	5.02±1.62	0.12±0.22	1.56±2.45

Table 4.1: Computation time (seconds) for each robot in one iteration

of each robot's onboard computation time for one iteration for each component of dSLAP and the corresponding percentages (%) of the total computation time. Discrete+OCA consumes most of the computation resources because a discrete set-valued approximation of the continuous dynamics over the entire state-action space is constructed through these two procedures, especially in OCA. Table 4.1 shows that the computation costs of the other procedures are mostly sub-second. Table 4.2 shows that the average wall time plus/minus one standard deviation per iteration for each robot versus the number of robots deployed. This shows that the computation time within each robot is nearly independent of the number of robots.

4.5.4 Hyperparameter tuning

The parameters in Algorithm 5, include mission and system parameters (i.e., $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_O$, $\mathcal{X}_G^{[i]}$, $\ell^{[i]}$ and $m^{[i]}$) and tuned parameters: Kernel for GPR: κ; Initial dis-

Number of robots	1	2	4	6	8
Wall time	5.837± 0.085	5.843± 0.118	5.830± 0.102	5.832± 0.129	5.839± 0.119

Table 4.2: dSLAP Wall clock time (seconds) per iteration

cretization parameter: p_{init}; Termination iteration: $\tilde{k}$; Number of samples to be obtained in each iteration: $\bar{\tau}$; Discrete time unit: ξ; Time horizon for MPC: φ; Weight in the MPC: ψ; Sampling period: δ; Utility function $r_k^{[i]}$. Below we provide an overall guidance on tuning these parameters.

Parameters p_{init}, $\tilde{k}$, $\bar{\tau}$, ξ, and φ, are referred as the computation parameters, since they are related to the computation power of the machine performing the simulation or the onboard computer of the robots. In the simulation, these parameters, though can affect performance, determine how much computation power is needed to compute the safe control inputs. Therefore, they can be mainly tuned based on how much computation power is available and how much computation time is desired in one iteration. The remaining parameters, referred as the learning parameters, to be tuned are: κ, δ, $r_k^{[i]}$ and ψ. Notice that the above parameters are more related to the learning of the unknown dynamics using GPR and active learning. Therefore, standard/common parameters in the related literature are used in the simulation. They can be tuned by following the general guidance of hyperparameter tuning for GPR [54] and active learning [126].

4.6 Conclusion

We study the problem where a group of mobile robots subject to unknown external disturbances aim to safely reach goal regions. We propose the dSLAP algorithm that enables the robots to quickly adapt to a sequence of learned models resulted from online Gaussian process regression, and safely reach the goal regions. We provide sufficient conditions to ensure the safety of the system. The developed algorithm is evaluated by Monte Carlo simulation.

Chapter 5

Federated reinforcement learning with zero-shot generalization

5.1 Introduction

The previous chapter considers online learning and imposes safety as a hard constraint. In this chapter, we consider offline learning and imposes safety as a soft constraint. The goal is to synthesize a control policy for robot motion planning with good zero-shot generalization.

Classic motion planning methods usually assume perfect knowledge of the dynamics of the robots and the environments they operate in. Examples of methods includes cell decomposition, roadmap, sampling-based approaches, and feedback motion planning. Interested readers are referred to [91] for more details. However, robots' operations in the real world are usually accompanied by uncertainties, such as the external disturbances in the natural environments they operate in and the modeling errors of the dynamics. To deal with the uncertainties, a number of methods utilize techniques in robust control (e.g., [133, 134, 94]), where bounded uncertainties are considered, and stochastic control (e.g., [95, 96, 97]), where the uncertainties are modeled in terms of known probability distributions. Recently, reinforcement learning-based approaches have been developed to relax the need of prior explicit uncertainty models (and even the dynamic models) by directly learning the best mapping from sensory data to control inputs from repetitive trials. For

example, paper [98] uses kernel methods to learn the control policy for a spider-like robot with 18 degrees of freedom using GPS data. Deep neural networks are used in [135, 136] to synthesize control policies using camera/LiDAR data.

Classic reinforcement learning problems consider learning an optimal control policy over a single environment [137]. The policy can either be learned online through agent's repetitive interaction and data collection in the environment [137] or learned offline using a fixed dataset of past interaction with the environment [138]. Although the methods can deal with complex environments, the agents struggle to generalize their experiences to new environments [139, 140]. This chapter focuses on the generalization of reinforcement learning, that is, obtaining a control policy which performs well in new environments unseen during training. Depending on whether or not the approaches require data collection and policy adaptation in a new environment, existing works on this problem can be categorized into few-shot generalization and zero-shot generalization.

Meta reinforcement learning (MRL) is a widely-used approach for few-shot generalization. More specifically, MRL aims to address the fundamental problem of quickly learning an optimal control policy in a new environment after collecting a small amount of data online and performing a few updates for policy adaptations [141, 142, 143, 144, 145, 146]. The problem is usually formulated as an optimization problem, where the objective function is the expected performance of the control policy adapted from a meta control policy after a few updates in a new environment. When it is applied to robots with unknown dynamics, MRL faces a particular challenge. Since they usually operate in real time, robots only have limited time to collect data in new environments and perform policy adaptation. When the dynamics of the robots are uncertain, data collection requires that the robots execute the meta control policies in physical environments and obtain the induced trajectories. The physical execution can be time-consuming and not suitable or even impractical for real-time applications.

Zero-shot generalization considers the performance of a single control policy in new environments without additional data collection and policy adaptation [140]. It is typically formulated as expected cost minimization of a control policy over a distribution of environments. As the distribution of the environments is generally complicated or even unknown, it is challenging, if not impossible, to solve the ex-

pected cost minimization problem in closed form. Therefore, the methods, which target zero-shot generalization, instead solve an empirical mean minimization problem (possibly with regularization) given a finite amount of training environments. Related methods can be categorized into two classes. The first one is modifying an expected cost function and solving the modified problem through empirical cost minimization [27, 147, 28, 29, 32, 33]. For example, risk-sensitive criterion can be introduced to balance between a return and a risk, where the risk can be the variance of the return [28, 29]. Worst-case criterion is used to mitigate the effects of the variability induced by a given policy due to the stochastic nature of the unseen environments or the dynamic systems [32, 33]. The other class is incorporating regularizers into empirical mean minimization to improve the generalizability of the solution. A necessarily incomplete list of references includes [100, 148, 149, 150]. While most regularization methods are heuristic, paper [100] uses the sum of the empirical cost and the generalization error from PAC-Bayes theory as an upper bound of the expected cost and synthesizes a control policy which can minimize the upper bound. Nevertheless, empirical mean minimization (with regularization) is an approximation to the expected cost minimization problem, and the optimality loss is not quantified. In this chapter, we aim to directly solve the expected cost minimization problem and analyze the properties of the solution.

The papers aforementioned focus on centralized reinforcement learning, where all the training data are possessed by a single learning agent. On the other hand, the advent of ubiquitous sensing and mobile storage renders some scenarios, in which training data are distributed across multiple entities, e.g., the driving data in different autonomous cars. It is well-known that control policies trained with more data have better performance [51]. However, directly using the raw data for collective learning can risk compromising the privacy of the data owners, e.g., exposing the living and working locations of the drivers. To tackle this challenge, distributed reinforcement learning is usually leveraged, where multiple learning agents perform training collaboratively by exchanging their locally learned models. There are mainly two approaches: decentralized reinforcement learning and federated reinforcement learning. In decentralized reinforcement learning, learning agents directly communicate with each other over P2P networks [151]. In federated reinforcement learning, learning agents cannot directly talk to each other and

instead are orchestrated by a Cloud, i.e., the learning agents download shared control policies from the Cloud, implement local updates based on local data and report the local control policies to the Cloud for the updates of the shared models [152, 153]. With the support of a Cloud, federated learning has access to more resources in, e.g., computation, memory and power, and hence enables a much larger scale of learning processes. The analysis of the above works is limited to the convergence of the proposed learning algorithms. The generalization of the learned control policies remains an open question.

Contribution statement: In this chapter, we propose a novel framework, FedGen, to tackle the challenge of robot motion planning with zero-shot generalization in the presence of distributed data across multiple learning entities. A network of learners aim to collaboratively learn a single control policy which can safely drive a robot to goal regions in different environments without data collection and policy adaptation during policy execution. The problem is formulated as federated optimization with an unknown objective function, which is the expected cost of navigation over a distribution of environments. Specifically, each learner updates its local control policy and sends its observation of the objective function to a central Cloud for global minimization among the control policies of the learners. The global minimizer is then sent back to the learners for updates of the local control policies. We characterize the upper bounds for the expected arrival time and safe arrival rate for each control policy. The upper bounds are used to find the control policy with the best zero-shot generalization performance among the learners. Theoretical guarantees on almost-sure convergence, almost consensus, Pareto improvement and global convergence are also provided. In addition, the algorithm can be executed over P2P networks after a minor change.

In summary, our contributions are: (C1) The development of the FedGen algorithm for robot motion planning with zero-shot generalization subject to multiple learning entities. (C2) The theoretic guarantees on the zero-shot generalization of local control policy to new environments in terms of arrival time and safety, the almost-sure convergence and the global convergence of the local estimates, the consensus of the local values and Pareto improvement of the local values. Monte Carlo simulations are conducted for evaluations.

Notations. We use superscript $(\cdot)^{[i]}$ to distinguish the local values of robot i

and $\|\cdot\|$ to denote 2-norm. For notional simplicity, for any local value $a^{[i]}$, we denote $a^{\max} \triangleq \max_{i\in\mathcal{V}} a^{[i]}$ and $a^{\min} \triangleq \min_{i\in\mathcal{V}} a^{[i]}$. Define closed ball $\mathcal{B}(\theta,\epsilon) \triangleq \{\theta' \in \mathbb{R}^{n_\theta} \mid \|\theta-\theta'\| \leqslant \epsilon\}$, and $\beta(\mathcal{A})$ the measure of set $\mathcal{A}$.

5.2 Problem formulation

In this section, we introduce the dynamics of the robot, the problem of motion planning, the setting of federated reinforcement learning, and the objective of this chapter.

5.2.1 Environment-specific motion planning

In this chapter, we consider environment-dependent dynamics. Let $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ be the state space of the robot and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ be the control input space. An environment E is fully specified by the inherent external disturbance $d_E : \mathcal{X}\times\mathcal{U} \to \mathcal{X}$, the obstacle region $\mathcal{X}_{O,E} \subseteq \mathcal{X}$ and the goal region $\mathcal{X}_{G,E} \subset \mathcal{X} \setminus \mathcal{X}_{O,E}$; i.e., $E \triangleq (d_E, \mathcal{X}_{O,E}, \mathcal{X}_{G,E})$. For each environment E, denote free region $\mathcal{X}_{F,E} \triangleq \mathcal{X} \setminus \mathcal{X}_{O,E}$. Denote $\mathcal{G}_{\mathcal{E}}$ the space of goal regions induced by the space of environments $\mathcal{E}$.

In each environment E, the dynamic system of the robot is given by the following difference equation:

$$x_{t+1} = f(x_t,u_t) + d_E(x_t,u_t),\ o_t = h(x_t,\mathcal{X}_{O,E}), \tag{5.1}$$

where $x_t \in \mathcal{X}$ is the state of the robot, $u_t \in \mathcal{U}$ is its control input, $o_t \in \mathcal{O}$ is the sensor output of the system observing the obstacle region $\mathcal{X}_{O,E}$ at state x_t and h is the observation function. Once environment E is revealed, $\mathcal{X}_{G,E}$ is known, $\mathcal{X}_{O,E}$ can only be observed through h and may not be fully known, but d_E is unknown.

The objective of the environment-specific motion planning problem is to synthesize a control policy, which can drive system (5.1) to the goal region with obstacle collision avoidance. The arrival time under control policy $\pi : \mathcal{O}\times\mathcal{G}_{\mathcal{E}} \to \mathcal{U}$ for system (5.1) starting from initial state x_{int} is given by

$$\begin{aligned} t_E(x_{int};\pi) \triangleq \inf\{t>0 \mid\ & x_t \in \mathcal{X}_{G,E}, x_0 = x_{int},\\ & x_{\tau+1} = f(x_\tau,u_\tau) + d_E(x_\tau,u_\tau),\ o_\tau = h(x_\tau,\mathcal{X}_{O,E}),\end{aligned}$$

$$u_\tau = \pi(o_\tau; \mathcal{X}_{G,E}), x_\tau \in \mathcal{X}_{F,E}, \forall 0 \leqslant \tau \leqslant t\}.$$

If the robot never reaches the goal, or hits the obstacles before arrival, then $t_E(x_{int}; \pi) = \infty$. We say safe arrival is achieved from initial state x_{int} under control policy π if $t_E(x_{int}; \pi) < \infty$. Note that $t_E(x_{int}; \pi)$ is potentially infinite, and it can cause numerical issues. Therefore, we normalize the arrival time function through Kruzkov transform such that the normalized cost function is given by $J_E(x_{int}; \pi) \triangleq 1 - e^{-t_E(x_{int};\pi)}$. Note that when $t_E(x_{int;\pi}) = \infty$, we have $J_E(x_{int}; \pi) = 1$.

5.2.2 Robot motion planning with zero-shot generalization

In the problem of robot motion planning with zero-shot generalization, the goal is to synthesize a single control policy that performs well in different environments without data collection and policy adaptation during policy execution. In statistical learning theory [51], this can be formulated as minimizing the expectation of the normalized arrival time over different environments. In particular, we assume the environments follow an unknown distribution.

Assumption 5.2.1. *(Stochastic environment).* There is an unknown distribution $\mathcal{P}_E$ over $\mathcal{E}$ from which environments are drawn from. ■

For example, the obstacle regions of the environments can be composed of a number of circular obstacles, where the numbers, locations, and the radii of the obstacles follow an unknown distribution, and the disturbances can follow an unknown Gaussian process.

Further, we assume that the initial state is a random variable which is conditional on the environment.

Assumption 5.2.2. *(Stochastic initialization).* There is an unknown conditional distribution $\mathcal{P}_{int|E}$ from which x_{int} is drawn conditional on environment $E \in \mathcal{E}$. ■

Formally, the objective of the problem of robot motion planning with zero-shot generalization is to synthesize a control policy $\pi_* \in \Gamma \triangleq \{u(\cdot) : \mathcal{O} \times \mathcal{G}_\mathcal{E} \to \mathcal{U}, \text{measurable}\}$, such that the expected normalized cost over all possible, including

unseen, environments is minimized:

$$\pi_* = \arg\min_{\pi\in\Gamma} \mathbb{E}[J_E(x_{int};\pi)], \tag{5.2}$$

where the expectation is taken over the environment $E \sim \mathcal{P}_E$ and initialization $x_{int} \sim \mathcal{P}_{int|E}$. Note that by taking the expectation, we are considering all possible environments following the distribution. Therefore, we measure the zero-shot generalization of a control policy using its expected cost of solving the motion planning problems in a distribution of environments.

Since Γ is a function space, problem (5.2) is a functional optimization problem and hard to solve in general. In order to make the problem tractable, we approximate the space Γ using, e.g., deep neural networks and basis functions. Consider a class of control policies $\pi_\theta \in \Gamma$ parameterized by $\theta \in \mathbb{R}^{n_\theta}$, e.g., the weights of a deep neural network. Denote $\eta(\theta) \triangleq \mathbb{E}[J_E(x_{int};\pi_\theta)]$. Then for the learners, problem (5.2) becomes:

$$\theta_* = \arg\min_{\theta\in\mathbb{R}^{n_\theta}} \eta(\theta). \tag{5.3}$$

Problem (5.3) is a standard expected cost minimization problem. However, since the distribution of the environments is unknown, (5.3) cannot be solved directly. A typical practice is to approximate it by empirical cost minimization (with regularization), e.g., [28, 29, 30, 32, 33, 100, 148, 149, 150], where a control policy is synthesized by minimizing the empirical cost (with regularization) over a finite number of training environments. Nevertheless, to the best of our knowledge, there is no theoretic guarantee on the optimality of the solutions to the original problem (5.3). In this chapter, we aim to directly solve (5.3) and analyze the properties of the solutions.

5.2.3 Federated reinforcement learning

Through federated learning, a group of learners aim to solve (5.3) collaboratively and achieve better results than solving on their own. Each learner $i \in \mathcal{V}$ observes function η by sampling a set of environments $E_l^{[i]} \overset{i.i.d.}{\sim} \mathcal{P}_E$, $l = 1, \cdots, n_{\mathcal{E}}^{[i]}$, and a set of initial states $x^{[i]}_{int|E_l^{[i]},l'} \sim \mathcal{P}_{int|E_l^{[i]}}$, $l' = 1, \cdots, n^{[i]}_{int|\mathcal{E}}$, for each $E_l^{[i]}$.

We consider general on-policy reinforcement learning methods. Given a triple of $(\theta^{[i]}, E_l^{[i]}, x^{[i]}_{int|E_l^{[i]},l'})$, learner i measures the value $J_{E_l^{[i]}}(x^{[i]}_{int|E_l^{[i]},l'}; \pi_{\theta^{[i]}})$ through policy evaluation, i.e., running the robot under control policy $\pi_{\theta^{[i]}}$ from initial state $x^{[i]}_{int|E_l^{[i]},l'}$ in environment $E_l^{[i]}$, measuring the arrival time and taking the Kruzkov transform. Then learner i finds (or approximate using, e.g., natural evolution strategies [154]) the policy gradient $\nabla_{\theta^{[i]}} J_{E_l^{[i]}}(x^{[i]}_{int|E_l^{[i]},l'}; \pi_{\theta^{[i]}})$. The learners communicate to a Cloud but do not communicate with each other.

The objective of the multi-learner network and the Cloud is to collaboratively solve problem (5.3). The problem is challenged by the fact that the objective function η is non-convex and can only be estimated by sampling over the environments and the initial states in general. As stated in Assumption 5.2.1, the environments at training and testing follow an unknown distribution. The estimation error is the difference between the true value of η and the empirical average of the normalized cost, and the distribution of the estimation error is unknown and non-Gaussian in general. Notice that when expected cost minimization is approximated by empirical cost minimization (possibly with regularization) as in [28, 29, 30, 31, 32, 33, 100, 148, 149, 150], the surrogate objective function is the sum of the empirical cost and the regularizer, which has closed-form and is free of estimation error.

5.3 Algorithm statement

In this section, we propose a federated optimization framework, FedGen in Algorithm 10, and analyze the generalized performances and the properties of the local estimates of the solution to problem (5.3) the algorithm renders. Overall, the proposed solution enables learning with distributed data without data sharing. The generalizability of a control policy is characterized by an upper bound of η, the expected adjusted arrival time, using the empirical mean of the adjusted arrival time in Theorem 5.3.1. We leverage the architecture of federated optimization, where the learners only exchange the parameters of their control policies and minimize the above upper bound to optimize the generalizability of its control policy. More detailed description of the proposed framework can be found in the subsection below.

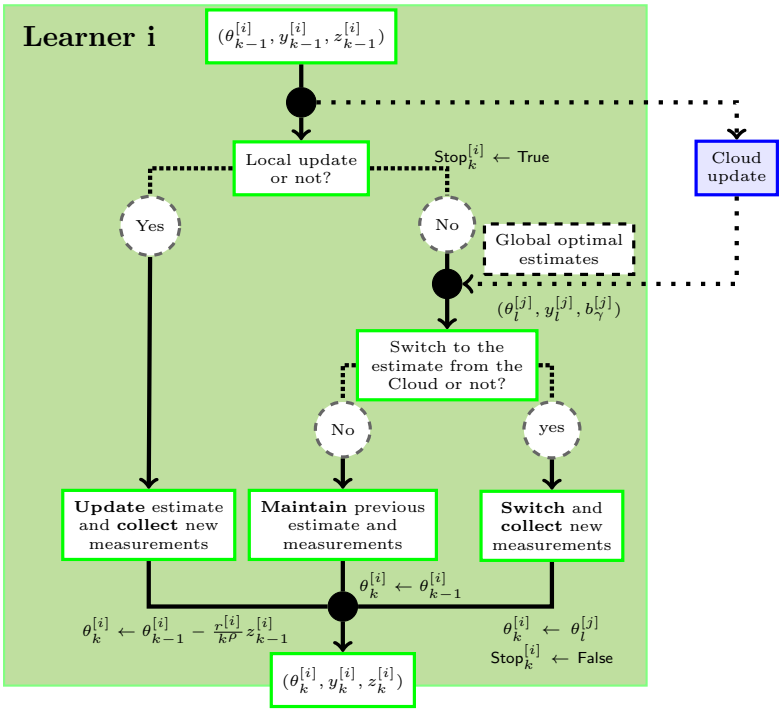


Figure 5.1: Implementation FedGen for learner i in iteration k

5.3.1 The FedGen algorithm

Denote $\theta_k^{[i]}$ the empirical estimate of the solution to problem (5.3) by learner i at iteration k. Denote $y_k^{[i]}$, the empirical estimate of $\eta(\theta_k^{[i]})$, and $z_k^{[i]}$, the empirical estimate of $\nabla\eta(\theta_k^{[i]})$ as follows.

$$y_k^{[i]} \triangleq \frac{1}{n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}} \sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} J_{E_l^{[i]}}(x_{int|E_l^{[i]},l'}^{[i]}; \pi_{\theta_k^{[i]}}),$$

$$z_k^{[i]} \triangleq \frac{1}{n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}} \sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} \nabla J_{E_l^{[i]}}(x_{int|E_l^{[i]},l'}^{[i]}; \pi_{\theta_k^{[i]}}).$$

The FedGen algorithm is composed of three components: (i) Learner-based update, where each learner updates its estimate $\theta_k^{[i]}$ using local data only. (ii) Cloud update, where the Cloud identifies the estimate with the best generalized performance among the learners. (iii) Learner-based fusion, where the learner decides whether it should keep its local estimate or switch to the one returned by the Cloud. The algorithm utilizes the power of the Cloud to identify the control

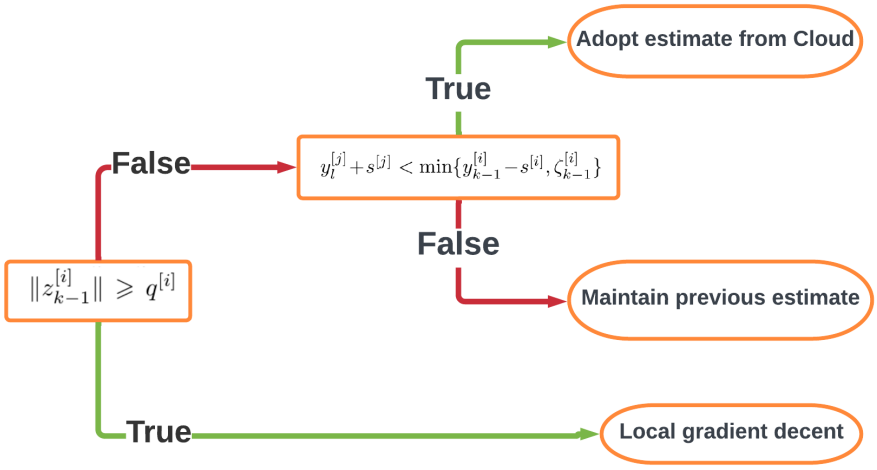


Figure 5.2: Parameter update logic at each iteration

policy that can potentially achieve better performance in expectation and allow the learners to escape from their local minima. Figure 5.1 is a detailed flowchart representation of Algorithm 10, demonstrating the decision making process within learner i. Figure 5.2 presents the logic of the update of the parameter estimates in one iteration. More detailed description of the each module in each iteration k can be found below.

5.3.1.1 Learner-based update

First, each learner i performs local learning using its local data. Specifically, each learner i collects the measurement $(y_{k-1}^{[i]}, z_{k-1}^{[i]})$ of the estimate $\theta_{k-1}^{[i]}$ in the previous iteration if it is not stopped. The measurements are sent to the Cloud for global minimization. If $\|z_{k-1}^{[i]}\|$ is greater than a local threshold $q^{[i]}$, which indicates that learner i's estimate is far from convergence and has potential for improvement, the learner makes one gradient descent step and updates its local estimate to $\hat{\theta}_k^{[i]}$. The threshold $q^{[i]}$ indicates whether a local minimum of η is achieved. If $\|z_{k-1}^{[i]}\|$ is not greater than $q^{[i]}$, the learner stops its local gradient descent and maintains the previous measurement. The learner resumes data collection for potential local gradient descent when it adopts the policy parameter from the Cloud later in Learner-based fusion for further optimization.

5.3.1.2 Cloud update

Note that the learners' estimates have different update trajectories due to the differences in initialization and data. Since objective η is nonconvex in general, different learners' estimates can stuck at different local minima. Therefore, the Cloud aims to identify which learner is around a better local minimum such that the other learners can later switch to this local minimum when their estimates converges in Learner-based update. Specifically, upon the receipt of local estimates of η, $(y_{k-1}^{[i]}, \theta_{k-1}^{[i]})$, from each $i \in \mathcal{V}$, the Cloud aims to find the policy parameter with the best generalized performance among the learners. Denote local bias $b_\gamma^{[i]} \triangleq \sqrt{\frac{\log(2/\gamma)}{2n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}}}$, $\gamma \in (0,1)$. The following theorem characterizes the zero-shot generalization error between $y_k^{[i]}$ and $\eta(\theta_k^{[i]})$ and the zero-shot generalized safety in terms of local bias, where the proof can be found in Section 5.4.

Theorem 5.3.1. Suppose Assumptions 5.2.1 and 5.2.2 hold. The following properties are true for all $i \in \mathcal{V}$:

(T1, Generalization error). For each $k \geqslant 0$, it holds that $\eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}$ with probability at least $1-\gamma$.

(T2, Generalized safety). For each $k \geqslant 0$, the policy $\pi_{\theta_k^{[i]}}$ is able to achieve safe arrival with probability at least $1-\gamma-(1-\gamma)(y_k^{[i]} + b_\gamma^{[i]})$ for $E \sim \mathcal{P}_E$ and $x_{int} \sim \mathcal{P}_{int|E}$. ■

In order to obtain the best zero-shot generalized performance, based on Theorem 5.3.1, the Cloud returns the global minimizer of $y_{l'}^{[i]} + b_\gamma^{[i]}$ over all the local estimates $\theta_{l'}^{[i]}$, $i \in \mathcal{V}, l' = 0, \cdots, k-1$, and sends the global minimizer and minimum to the learners. Different from the regularizers used in the literature of empirical cost minimization, the local bias $b_\gamma^{[i]}$ is a constant value and does not depend on the estimate $\theta_k^{[i]}$. This procedure can be implemented recursively by comparing the learner-wise global minimum in the previous iteration with the values obtained in the current iteration. If one wants to implement Algorithm 10 over P2P networks without the Cloud, this step can be executed using the minimum consensus algorithm [155].

5.3.1.3 Learner-based fusion

For each learner, it may not be always the case that the global minimizer of the Cloud outperforms the local estimate. The learner's estimate only switches to the estimate returned from the Cloud if its estimate converges in Learner-based update and the estimate from the Cloud is significantly better than the local estimate. Specifically, Learner i only chooses the global minimizer $\theta_l^{[j]}$ sent by the Cloud when two conditions are satisfied: (i) estimate $\theta_l^{[j]}$ achieves a smaller estimate of η, i.e., $y_l^{[j]} + b_\gamma^{[j]}$ is less than the minimum between $y_{k-1}^{[i]} - b_\gamma^{[i]}$, and $\zeta_{k-1}^{[i]}$, the previous global minimum adopted by learner i; and (ii) local gradient descent is stopped, i.e., $z_{k-1}^{[i]}$ is small. When the global minimizer is chosen, learner i is then not stopped and resumes Learner-based update in the next iteration. Notice that if it never chooses the global minimizer from the Cloud after it is stopped, learner i maintains the estimate and measurement for the remaining iterations.

5.3.2 Performance guarantees

In this section, we investigate the limiting behavior of the algorithm. Similar to most analysis of stochastic gradient descent (please see [156, 157] and the references therein), we assume η is Lipschitz continuous and $L_{\nabla\eta}$-smooth.

Assumption 5.3.2. *(Lipschitz continuity).* There exists positive constant L_η such that $|\eta(\theta) - \eta(\theta')| \leqslant L_\eta \|\theta - \theta'\|$ for all $\theta, \theta' \in \mathbb{R}^{n_\theta}$. ■

Assumption 5.3.3. *($L_{\nabla\eta}$-smooth).* There exists positive constant $L_{\nabla\eta}$ such that $\|\nabla\eta(\theta) - \nabla\eta(\theta')\| \leqslant L_{\nabla\eta}\|\theta - \theta'\|$ for all $\theta, \theta' \in \mathbb{R}^{n_\theta}$. ■

Furthermore, we assume that the variance of the errors of gradient estimation is bounded. This is a standard assumption in the analysis of stochastic optimization [156][157].

Assumption 5.3.4. *(Bounded variance).* It holds that $\mathbb{E}[\|z_k^{[i]} - \nabla\eta(\theta_k^{[i]})\|^2] \leqslant (\sigma^{[i]})^2$ for some $\sigma^{[i]} > 0$. ■

Notice that the updates of the variables $\theta_k^{[i]}$, $y_k^{[i]}$ and $z_k^{[i]}$, $k \geqslant 1$, depends on the sampling of the environments and the initial states in all the learners, which are the only randomness in this chapter. Therefore, in the sequel, *all* the expectations

of these local variables are taken over the sampling $E_l^{[j]} \sim \mathcal{P}_E$, $l = 1, \cdots, n_{\mathcal{E}}^{[j]}$, and $x_{int|E_l^{[j]},l'}^{[j]} \sim \mathcal{P}_{int|E_l^{[j]}}$, $l' = 1, \cdots, n_{int|\mathcal{E}}^{[j]}$ for all $j \in \mathcal{V}$. The lemma below shows that $z_k^{[i]}$ is an unbiased estimate of $\nabla\eta(\theta_k^{[i]})$.

Lemma 5.3.5. *(Unbiased estimator).* Suppose Assumptions 5.2.1, 5.2.2 and 5.3.2 hold. Then it holds that $\mathbb{E}[z_k^{[i]}] - \nabla\eta(\theta_k^{[i]}) = 0$ for all $k \geqslant 1$. ■

Since $z_k^{[i]}$ is an unbiased estimate of $\nabla\eta(\theta_k^{[i]})$, by the law of large numbers (Proposition 6.3 in [158]), $(\sigma^{[i]})^2$ diminishes as $n_{\mathcal{E}}^{[i]}$ and $n_{int|\mathcal{E}}^{[i]}$ increase.

The following theorem summarizes the properties of almost-sure convergence, almost consensus and Pareto improvement of the algorithm.

Theorem 5.3.6. Suppose Assumptions 5.2.1, 5.2.2, 5.3.2 5.3.3 and 5.3.4 hold. For all $i \in \mathcal{V}$, if $r^{[i]} \leqslant \frac{1}{2L_{\nabla\eta}}$ and $q^{[i]} \geqslant 4\sigma^{[i]}$, then the followings hold:

(T3, Almost-sure convergence). There exists $\theta_\infty^{[i]} \in \mathbb{R}^{n_\theta}$ such that $\theta_k^{[i]} \to \theta_\infty^{[i]}$ almost surely.

(T4, Almost consensus). It holds that $\mathbb{E}[\max_{j\in\mathcal{V}} \eta(\theta_\infty^{[j]}) - \min_{j\in\mathcal{V}} \eta(\theta_\infty^{[j]})] \leqslant 2b_\gamma^{\max}$.

Denote $k_{fs}^{[i]} \triangleq \min\{k \geqslant 0 \mid \|z_k^{[i]}\| < q^{[i]}\}$ the first time learner i is stopped. Then we further have

(T5, Pareto improvement). If $\theta_\infty^{[i]} \neq \theta_{k_{fs}^{[i]}}^{[i]}$, then $\mathbb{E}[\eta(\theta_\infty^{[i]}) - \eta(\theta_{k_{fs}^{[i]}}^{[i]})] \leqslant -2b_\gamma^{\min}$. ■

Note that $\theta_\infty^{[i]} \neq \theta_{k_{fs}^{[i]}}^{[i]}$ implies that learner i adopts the estimates from the Cloud at least once. Theorem 5.3.6 (T5) implies that communication with the Cloud can potentially improve the optimality of the learners' estimates.

Denote the set of global minimizers that are regular in the sense of Hurwitz as

$$\Theta_* \triangleq \{\theta \in \mathbb{R}^{n_\theta} \mid \theta = \arg\min_{\theta' \in \mathbb{R}^{n_\theta}} \eta(\theta'), \nabla^2\eta(\theta) \succ 0\}.$$

Lemma 1 in [159] indicates that for each $\theta_* \in \Theta_*$, there exists a convex compact neighborhood $\mathcal{K}(\theta_*)$ and constant $\alpha > 0$ such that

$$\alpha\|\theta - \theta_*\|^2 \leqslant \langle\nabla\eta(\theta), \theta - \theta_*\rangle, \ \forall\theta \in \mathcal{K}(\theta_*). \tag{5.4}$$

Define $\epsilon_0(\theta_*) \triangleq \max\{\epsilon > 0 \mid \mathcal{B}(\theta_*, 4\epsilon + 2\sqrt{\epsilon}) \subset \mathcal{K}(\theta_*)\}$ for each $\theta_* \in \Theta_*$. Denote $\eta_* \triangleq \min_{\theta \in \mathbb{R}^{n_\theta}} \eta(\theta)$ the minimum value of η. Theorem 5.3.7 below characterizes the global convergence of FedGen.

Theorem 5.3.7. *(Global convergence).* Suppose Θ_* is non-empty, and $\theta_0^{[i]}$ is independently uniformly sampled over a compact set Θ_0 for all $i \in \mathcal{V}$, where $\beta(\Theta_0 \cap [\cup_{\theta_* \in \Theta_*} \mathcal{B}(\theta_*, 2\epsilon_0(\theta_*))]) > 0$. Suppose all the conditions in Theorem 5.3.6 hold. There exist $\bar{\beta} \in (0, 1]$ and class $\mathcal{K}_\infty$ function $\kappa(\cdot)$ such that, $\forall i \in \mathcal{V}$ and any $\epsilon_1, \epsilon_2, \epsilon_3 > 0$,

$$\eta(\theta_\infty^{[i]}) - \eta_* \leqslant \frac{L_\eta(q^{\max} + \epsilon_1)}{\alpha} + \epsilon_2 + 2\epsilon_3 b_\gamma^{\max} \tag{5.5}$$

with probability at least

$$1 - \frac{(\sigma^{\max})^2}{\epsilon_1^2} - 2\exp(-2\epsilon_2^2) - \frac{1}{\epsilon_3} - (1 - \bar{\beta})^{|\mathcal{V}|} - \kappa(r^{\max}). \blacksquare \tag{5.6}$$

5.3.3 Discussion

(Adjusting generalized safety through $b_\gamma^{[i]}$). By (T2) in Theorem 5.3.1, the probability of safe arrival in a new environment is lower bounded by the (adjusted) empirical normalized arrival time $(1-\gamma)(1-y_k^{[i]})$ and the estimation error term $(1-\gamma)b_\gamma^{[i]}$. Since $y_k^{[i]} \in [0, 1]$, we always have $(1-\gamma)(1-y_k^{[i]}) \geqslant 0$, the equality holds only when $y_k^{[i]} = 1$, i.e., the policy $\pi_{\theta_k^{[i]}}$ renders collision in all the training environments and initial states. This also implies that γ should be small in order to have a high safe arrival rate. Given any $\gamma \in (0, 1)$, $b_\gamma^{[i]}$ in the error term $(1-\gamma)b_\gamma^{[i]}$ can be reduced to an arbitrarily small value by increasing $n_{\mathcal{E}}^{[i]}$ and $n_{int|\mathcal{E}}^{[i]}$ for any $\gamma > 0$.

(Hyperparameter tuning of r and $q^{[i]}$). Similar to the literature in non-convex stochastic optimization [156][157], Theorem 5.3.6 requires hyperparameters r and $q^{[i]}$ to satisfy certain conditions that depend on parameters $L_{\nabla\eta}$ and $\sigma^{[i]}$, which can be unknown *a priori*. However, these parameters can be estimated numerically; e.g., $L_{\nabla\eta}$ can be estimated using finite differences and $\sigma^{[i]}$ can be estimated using empirical variance. In practice, these conditions can also be satisfied by tuning r small enough and $q^{[i]}$ large enough through trial and error, a standard practice

of hyperparameter tuning in training machine learning models, e.g., deep neural networks.

(Trade-off between consensus gap and improvement by the selection of $b_\gamma^{[i]}$). Theorem 5.3.6 (T4) implies that the consensus gap can be reduced by reducing $b_\gamma^{[i]}$ for all $i \in \mathcal{V}$. However, a small $b_\gamma^{[i]}$ can delay the convergence of the algorithm as Lemma 5.4.5 later shows that the number of times the learners adopts the estimates from the Cloud is upper bounded by $\frac{1}{\min_{j\in\mathcal{V}} b_\gamma^{[j]}}$. Similarly, there is also a trade-off in the selection of $b_\gamma^{[i]}$ in (T5) of Theorem 5.3.6. Theorem 5.3.6 (T5) shows that the improvement can be increased by increasing $b_\gamma^{[i]}$ for all $i \in \mathcal{V}$. However, as Lemma 5.4.5 later shows, this can reduce the number of times the learners adopt the estimates from the Cloud and hence reduce the probability $P\Big(\theta_\infty^{[i]} \neq \theta_{k_{fs}^{[i]}}^{[i]}\Big)$. This can eventually increase the total expectation $\mathbb{E}[\eta(\theta_\infty^{[i]}) - \eta(\theta_{k_{fs}^{[i]}}^{[i]})]$. Informally speaking, the selection of $b_\gamma^{[i]}$ determines the minimal gain learner i demands after adopting the estimates from the Cloud. Therefore, larger $b_\gamma^{[i]}$ can prevent learner i from adopting the estimates from the Cloud with small optimality improvement. Consider the extreme case when $\min_{j\in\mathcal{V}} b_\gamma^{[j]}$ is so large that the learners would never adopt the estimates from the Cloud. Then we have $\theta_\infty^{[i]} = \theta_{k_{fs}^{[i]}}^{[i]}$ for all $i \in \mathcal{V}$, and there would be no improvement benefited from communication. Nevertheless, the right hand side in (T5) of Theorem 5.3.6 is always non-positive, which implies that the adopted estimate is at least as optimal as the estimate without communication.

(The number of learners versus sample sizes in the learners). The upper bound in (5.5) implies that smaller $q^{[j]}$ and smaller $b_\gamma^{[j]}$ for all $j \in \mathcal{V}$ can reduce the optimality gap. Recall the condition $q^{[j]} \geqslant 4\sigma^{[j]}$ and the definition of $b_\gamma^{[j]}$ above Theorem 5.3.1. Then (5.5) implies that large sample sizes, i.e., $n_{\mathcal{E}}^{[j]}$ and $n_{int|\mathcal{E}}^{[j]}$, for all the learners can reduce the optimality gap. The probability bound (5.6) indicates that smaller variance of the estimation error $\sigma^{\max}$ and larger $|\mathcal{V}|$ can increase the probability of achieving the optimality gap in (5.5). The class $\mathcal{K}_\infty$ function $\kappa(r^{\max})$ imposes a preference on small step size $r^{[j]}$.

5.4 Proofs

5.4.1 Proof of Theorem 5.3.1

We first quantify the estimation error of $y_k^{[i]}$ and prove (T1). Then we summarize the safety of the estimates and prove (T2).

The proof of (T1) is an adoption of Hoeffding's inequality below.

Theorem 5.4.1. (*Hoeffding's inequality, [160]*). Let $q_1, \cdots, q_n$ be independent random variables such that q_l takes its values in $[a_l, b_l]$ almost surely for all $1 \leqslant l \leqslant n$. Then for every $\epsilon > 0$, it holds that

$$P\Big(|\sum_{l=1}^{n} q_l - \mathbb{E}[\sum_{l=1}^{n} q_l]| \geqslant \epsilon\Big) \leqslant 2\exp\Big(-\frac{2\epsilon^2}{\sum_{l=1}^{n}(b_l - a_l)^2}\Big). \qquad \blacksquare$$

Proof of (T1): Assumptions 5.2.1 and 5.2.2 imply $\mathbb{E}[J_E(x^{[i]}_{int|E}; \pi_{\theta_k^{[i]}})] = \eta(\theta_k^{[i]})$. Note that $J_E \in [0,1]$. Let $q_{ll'} \triangleq J_{E_l^{[i]}}(x^{[i]}_{int|E_l^{[i]},l'}; \pi_{\theta_k^{[i]}})$ and hence

$$\mathbb{E}[q_{ll'}] = \mathbb{E}[J_{E_l^{[i]}}(x^{[i]}_{int|E_l^{[i]},l'}; \pi_{\theta_k^{[i]}})] = \eta(\theta_k^{[i]}).$$

Then we have

$$\sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} q_{ll'} = \sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} J_{E_l^{[i]}}(x^{[i]}_{int|E_l^{[i]},l'}; \pi_{\theta_k^{[i]}}) = n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]} y_k^{[i]},$$

$$\sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} \mathbb{E}[q_{ll'}] = n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]} \eta(\theta_k^{[i]}).$$

Then Theorem 5.4.1 gives $n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]} |y_k^{[i]} - \eta(\theta_k^{[i]})| \leqslant n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]} \epsilon$ with probability at least $1 - 2\exp\Big(-2\epsilon^2 n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}\Big)$ for each $k \geqslant 0$. After some simple algebraic transformations, we have

$$|y_k^{[i]} - \eta(\theta_k^{[i]})| \leqslant \sqrt{\frac{\log(2/\gamma)}{2 n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}}}, \tag{5.7}$$

with probability at least $1 - \gamma$, $\forall i \in \mathcal{V}$ and $k \geqslant 0$. $\blacksquare$

Notice that $J_E(x_{int};\pi_{\theta_k^{[i]}}) \in [0,1]$ for any $E \in \mathcal{E}$ and $x_{int} \in \mathcal{X}$, and by definition of J_E, safe arrival is equivalent to $J_E(x_{int};\pi_{\theta_k^{[i]}}) < 1$. Then the proof of (T2) is given as follows.

Proof of (T2): (T1) renders that $\eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}$ with probability at least $1-\gamma$. Since Assumptions 5.2.1 and 5.2.2 imply $\mathbb{E}[J_E(x_{int};\pi_{\theta_k^{[i]}})] = \eta(\theta_k^{[i]})$, we have $\mathbb{E}[J_E(x_{int};\pi_{\theta_k^{[i]}}) \mid \eta(\theta_k^{[i]}) \leqslant a] \leqslant a$ for any $a \in \mathbb{R}$. Combining this with Markov's inequality (page 151, [88]), we have

$$\begin{aligned}
&P\Big(J_E(x_{int};\pi_{\theta_k^{[i]}}) \geqslant 1 \mid \eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}\Big) \\
&\leqslant \mathbb{E}[J_E(x_{int};\pi_{\theta_k^{[i]}}) \mid \eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}] \leqslant y_k^{[i]} + b_\gamma^{[i]}.
\end{aligned}$$

Then we further have

$$\begin{aligned}
&P\Big(J_E(x_{int};\pi_{\theta_k^{[i]}}) < 1, \eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}\Big) \\
&= P\Big(J_E(x_{int};\pi_{\theta_k^{[i]}}) < 1 \mid \eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}\Big) P\Big(\eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}\Big) \\
&\geqslant \big(1 - (y_k^{[i]} + b_\gamma^{[i]})\big)(1-\gamma). \qquad (5.8)
\end{aligned}$$

Notice that

$$P\Big(J_E(x_{int};\pi_{\theta_k^{[i]}}) < 1\Big) \geqslant P\Big(J_E(x_{int};\pi_{\theta_k^{[i]}}) < 1, \eta(\theta_k^{[i]}) \leqslant y_k^{[i]} + b_\gamma^{[i]}\Big).$$

Hence, the proof is concluded. ■

5.4.2 Proof of Theorem 5.3.6

In this section, we first provide a set of preliminary results in Section 5.4.2.1, which mainly discusses the properties of the estimation of $z_{k-1}^{[i]}$ and the estimates after the last time the learner adopts the estimate returned from the Cloud. Then the proofs of (T3), (T4) and (T5) of Theorem 5.3.6 are presented in Sections 5.4.2.2, 5.4.2.3 and 5.4.2.4, respectively.

To facilitate the proof, some important iterations of the algorithm FedGen are defined/repeated in Table 5.1.

Symbol	Definition
$k_n^{[i]}$, $n = 1, 2, \cdots$	The iteration when Lines 20-23 are executed; i.e., learner i adopts the estimates from the Cloud.
$k_*^{[i]}$	The last time Lines 20-23 are executed. If Lines 20-23 are never executed, then $k_*^{[i]} = 0$.
$k_{fs}^{[i]}$	The first time learner i is stopped: $k_{fs}^{[i]} \triangleq \min\{k \geqslant 0 \mid \|z_k^{[i]}\| < q^{[i]}\}$.
$k_{ls}^{[i]}$	The last time learner i is stopped: $k_{ls}^{[i]} \triangleq \min\{k \geqslant k_*^{[i]} \mid \|z_k^{[i]}\| < q^{[i]}\}$.

Table 5.1: Definitions of important iterations

Notice that the above iterations satisfy:

$$k_{fs}^{[i]} + 1 \leqslant k_1^{[i]} < k_2^{[i]} < \cdots < k_*^{[i]} \leqslant k_{ls}^{[i]}. \tag{5.9}$$

5.4.2.1 Preliminary results

First of all, we provide the proof of Lemma 5.3.5.

Proof of Lemma 5.3.5: Assumption 5.3.2 implies that η is almost everywhere differentiable (Theorem 3.1.6 [161]). Hence, Interchange of Differentiation and Integration (Corollary 2.8.7, [162]) and Assumptions 5.2.1 and 5.2.2 give

$$\mathbb{E}[z_k^{[i]}] = \mathbb{E}\Big[\nabla[\frac{1}{n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}} \sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} J_{E_l^{[i]}}(x_{int|E_l^{[i]},l'}^{[i]}; \pi_{\theta_k^{[i]}})]\Big]$$

$$= \nabla\mathbb{E}[\frac{1}{n_{\mathcal{E}}^{[i]} n_{int|\mathcal{E}}^{[i]}} \sum_{l=1}^{n_{\mathcal{E}}^{[i]}} \sum_{l'=1}^{n_{int|\mathcal{E}}^{[i]}} J_{E_l^{[i]}}(x_{int|E_l^{[i]},l'}^{[i]}; \pi_{\theta_k^{[i]}})] = \nabla\eta(\theta_k^{[i]}). \qquad \blacksquare$$

Denote the estimation error $\xi_k^{[i]} \triangleq \nabla\eta(\theta_k^{[i]}) - z_k^{[i]}$. Lemma 5.4.2 quantifies $\|\xi_k^{[i]}\|$.

Lemma 5.4.2. Suppose Assumption 5.3.4 holds. Then it holds that $\|\xi_k^{[i]}\| \leqslant \epsilon$, $\epsilon > 0$, with probability at least $1 - \frac{(\sigma^{[i]})^2}{\epsilon^2}$.

Proof: Combining Assumption 5.3.4 and Markov's inequality renders $\|\xi_k^{[i]}\|^2 \geqslant \epsilon^2$, $\epsilon > 0$, with probability at most $\frac{\mathbb{E}[\|\xi_k^{[i]}\|^2]}{\epsilon^2} \leqslant \frac{(\sigma^{[i]})^2}{\epsilon^2}$, or $\|\xi_k^{[i]}\| \leqslant \epsilon$ with probability at least $1 - \frac{(\sigma^{[i]})^2}{\epsilon^2}$. $\blacksquare$

The following lemma provides a property of the expectation of $\|\xi_k^{[i]}\|$.

Lemma 5.4.3. It holds that $\mathbb{E}[\|\xi_k^{[i]}\|] = \int_0^\infty P\Big(\|\xi_k^{[i]}\| > t\Big)dt$.

Proof: For all $t \geqslant 0$, it holds that $t(1 - P\Big(\|\xi_k^{[i]}\| \leqslant t\Big)) \geqslant 0$. By Lemma 5.4.2, we also have

$$\lim_{t\to\infty} t(1 - P\Big(\|\xi_k^{[i]}\| \leqslant t\Big)) \leqslant \lim_{t\to\infty} t(1 - (1 - \frac{(\sigma^{[i]})^2}{t^2})) = 0.$$

Therefore, $\lim_{t\to\infty} t(1 - P\Big(\|\xi_k^{[i]}\| \leqslant t\Big)) = 0$. Denote $p(\cdot)$ the probability density function of random variable $\|\xi_k^{[i]}\|$. By integration by parts, we have

$$\begin{aligned}\int_0^\infty (1 - P\Big(\|\xi_k^{[i]}\| \leqslant t\Big))dt &= t(1 - P\Big(\|\xi_k^{[i]}\| \leqslant t\Big))\Big|_{t=0}^{\infty} + \int_0^\infty tp(\|\xi_k^{[i]}\| = t)dt \\ &= \int_0^\infty tp(\|\xi_k^{[i]}\| = t)dt.\end{aligned}$$

Since $\|\xi_k^{[i]}\| \geqslant 0$, we have $p(\|\xi_k^{[i]}\| = t) = 0$ for all $t < 0$. Therefore, we have

$$\mathbb{E}[\|\xi_k^{[i]}\|] = \int_{-\infty}^\infty tp(\|\xi_k^{[i]}\| = t)dt = \int_0^\infty tp(\|\xi_k^{[i]}\| = t)dt = \int_0^\infty P\Big(\|\xi_k^{[i]}\| > t\Big)dt. \quad \blacksquare$$

The following lemma finds a lower bound of $\langle \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle$ for all $\lambda \in [0, \frac{r^{[i]}}{k^\rho}]$.

Lemma 5.4.4. Suppose Assumptions 5.3.3 and 5.3.4 hold. It holds that, for any $\epsilon > 0$ and $\lambda \in [0, \frac{r^{[i]}}{k^\rho}]$,

$$\langle \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle \geqslant (1 - L_{\nabla\eta}\frac{r^{[i]}}{k^\rho})\|z_{k-1}^{[i]}\|^2 - \|\xi_{k-1}^{[i]}\|\|z_{k-1}^{[i]}\|.$$

Proof: Denote $\nu \triangleq \nabla\eta(\theta_{k-1}^{[i]}) - \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]})$. Write

$$\langle \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle = \langle \nabla\eta(\theta_{k-1}^{[i]}) - \nu, z_{k-1}^{[i]}\rangle = \langle \nabla\eta(\theta_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle - \langle \nu, z_{k-1}^{[i]}\rangle. \tag{5.10}$$

Next we find the lower bounds of the two terms on the right hand side of (5.10). Consider the first term. Then we have

$$\langle \nabla\eta(\theta_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle = \langle z_{k-1}^{[i]} + \xi_{k-1}^{[i]}, z_{k-1}^{[i]}\rangle = \|z_{k-1}^{[i]}\|^2 + \langle \xi_{k-1}^{[i]}, z_{k-1}^{[i]}\rangle. \tag{5.11}$$

By the Cauchy-Schwartz inequality, we have

$$\langle \nabla \eta(\theta_{k-1}^{[i]}), z_{k-1}^{[i]} \rangle \geqslant \|z_{k-1}^{[i]}\|^2 - \|\xi_{k-1}^{[i]}\| \|z_{k-1}^{[i]}\|. \tag{5.12}$$

Consider the second term in (5.10). Assumption 5.3.3 implies

$$\|\nu\| \leqslant L_{\nabla\eta} \|\theta_{k-1}^{[i]} - (\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]})\| = L_{\nabla\eta} \lambda \|z_{k-1}^{[i]}\| \leqslant L_{\nabla\eta} \frac{r^{[i]}}{k^\rho} \|z_{k-1}^{[i]}\|. \tag{5.13}$$

Using the Cauchy-Schwartz inequality and (5.13) render

$$\langle \nu, z_{k-1}^{[i]} \rangle \leqslant \|\nu\| \|z_{k-1}^{[i]}\| \leqslant L_{\nabla\eta} \frac{r^{[i]}}{k^\rho} \|z_{k-1}^{[i]}\|^2. \tag{5.14}$$

Combining (5.12) and (5.14) with (5.10) gives the lemma. ■

Next Lemma 5.4.5 shows that each learner i only adopts the estimates from the Cloud for a finite number of times.

Lemma 5.4.5. It holds that $n \leqslant \frac{1}{\min_{j\in\mathcal{V}} b_\gamma^{[j]}}$ for all $k_n^{[i]}$, $i \in \mathcal{V}$.

Proof: Pick any $i \in \mathcal{V}$. Note that when Lines 20-23 are executed at iteration $k_n^{[i]}$, we must have

$$\zeta_{k_n^{[i]}}^{[i]} = y_l^{[j]} < \zeta_{k_n^{[i]}-1}^{[i]} - b_\gamma^{[j]} \leqslant \zeta_{k_n^{[i]}-1}^{[i]} - b_\gamma^{\min}, \tag{5.15}$$

where $(j, l) = \arg\min_{i\in\mathcal{V}, l'=0,\cdots,k_n^{[i]}-1} y_{l'}^{[i]} + b_\gamma^{[i]}$. Since initialization gives $\zeta_0^{[i]} = 1$, (5.15) implies

$$\zeta_{k_n^{[i]}}^{[i]} \leqslant 1 - n b_\gamma^{\min}. \tag{5.16}$$

Since $\zeta_{k_n^{[i]}}^{[i]} \in [0, 1]$, (5.16) renders $n \leqslant \frac{1}{b_\gamma^{\min}}$. ■

Next we show that the event $\|z_k^{[i]}\| < q^{[i]}$ happens almost surely, which indicates convergence to a local minimum, by showing the almost sure existence of $k_{ls}^{[i]}$.

Lemma 5.4.6. Suppose Assumptions 5.2.1, 5.2.2, 5.3.2, 5.3.3 and 5.3.4 hold. If $q^{[i]} \geqslant 4\sigma^{[i]}$, then it holds that $k_{ls}^{[i]}$ exists almost surely.

Proof: By definition of $k_{ls}^{[i]}$, we have $\|z_k^{[i]}\| \geqslant q^{[i]}$ for all $k \in [k_*^{[i]}, k_{ls}^{[i]}]$ and hence Lines 20-23 are never executed for all $k \in [k_*^{[i]}, k_{ls}^{[i]}]$. Denote event $A \triangleq \{k_{ls}^{[i]}$ exists.$\}$

and the complement $A^c \triangleq \{k_{ls}^{[i]} \text{ does not exist.}\}$. Notice that we can equivalently write $A^c = \{\|z_k^{[i]}\| \geqslant q^{[i]}, \forall k \geqslant k_*^{[i]}\}$. Then A^c implies Lines 12 and 25 are executed for all $k \geqslant k_*^{[i]}$ and hence $\theta_k^{[i]} = \hat{\theta}_k^{[i]} = \theta_{k-1}^{[i]} - \frac{r^{[i]}}{k^\rho} z_{k-1}^{[i]}$ for all $k \geqslant k_*^{[i]}$, which is a stochastic gradient descent step [157]. Given Assumptions 5.3.2, 5.3.3 and 5.3.4, and Lemma 5.3.5, Corollary 3.3 and inequality (3.32) in [157] show that $\|\nabla\eta(\theta_k^{[i]})\| \to 0$ almost surely. Then, for any $\delta > 0$, there exists some $K_\delta > k_*^{[i]}$ such that $\|\nabla\eta(\theta_k^{[i]})\| < \delta$ for all $k \geqslant K_\delta$ almost surely. Since $q^{[i]} \geqslant 4\sigma^{[i]}$, we can pick $\delta \in (0, \sigma^{[i]})$ and let $\epsilon \triangleq q^{[i]} - \delta$. By the above construction, we have $\epsilon > \sigma^{[i]}$. Then Lemma 5.4.2 implies

$$\begin{aligned}\|z_k^{[i]}\| =& \|z_k^{[i]} - \nabla\eta(\theta_k^{[i]}) + \nabla\eta(\theta_k^{[i]})\| \leqslant \|z_k^{[i]} - \nabla\eta(\theta_k^{[i]})\| \\ &+ \|\nabla\eta(\theta_k^{[i]})\| \leqslant \epsilon + \|\nabla\eta(\theta_k^{[i]})\| < q^{[i]} \end{aligned} \tag{5.17}$$

with probability at least $1 - \frac{(\sigma^{[i]})^2}{\epsilon^2}$, $\frac{(\sigma^{[i]})^2}{\epsilon^2} < 1$, for each $k \geqslant K_\delta$. Due to the independent estimate of $z_k^{[i]}$ over k, we have

$$\begin{aligned} P\Big(A^c\Big) &= \lim_{\tilde{k}\to\infty} P\Big(\|z_k^{[i]}\| \geqslant q^{[i]}, \forall k \in [k_*^{[i]}, \tilde{k}]\Big) \\ &\leqslant \lim_{\tilde{k}\to\infty} P\Big(\|z_k^{[i]}\| \geqslant q^{[i]}, \forall k \in [K_\delta, \tilde{k}]\Big) \\ &\leqslant \lim_{\tilde{k}\to\infty} \Big(\frac{(\sigma^{[i]})^2}{\epsilon^2}\Big)^{\tilde{k}-K_\delta} = 0. \end{aligned}$$

Therefore, $P\Big(A\Big) = 1 - P\Big(A^c\Big) = 1$. ■

The following lemma shows that $\eta(\theta_k^{[i]}) \leqslant \eta(\theta_{k_*^{[i]}}^{[i]})$, for all $k \geqslant k_*^{[i]}$ in expectation.

Lemma 5.4.7. Suppose Assumptions 5.3.3 and 5.3.4 hold, $r^{[i]} \leqslant \frac{1}{2L_{\nabla\eta}}$ and $q^{[i]} \geqslant 4\sigma^{[i]}$. It holds that $\mathbb{E}[\eta(\theta_k^{[i]}) - \eta(\theta_{k_*^{[i]}}^{[i]})] \leqslant 0$ for all $k \geqslant k_*^{[i]}$.

Proof: Recall that $k_*^{[i]}$ is the last time learner i adopts the estimate from the Cloud, and Lemma 5.4.5 shows that $k_*^{[i]}$ exists. Note that Figure 5.2 indicates that $\theta_k^{[i]} = \theta_{k_{ls}^{[i]}}^{[i]} = \theta_\infty^{[i]}$ for all $k \geqslant k_{ls}^{[i]}$. When $k_{ls}^{[i]} = k_*^{[i]}$, we have $\mathbb{E}[\eta(\theta_k^{[i]}) - \eta(\theta_{k_*^{[i]}}^{[i]})] = 0$ for all $k \geqslant k_*^{[i]}$. Hence, in the sequel, we consider the case where $\theta_k^{[i]} = \hat{\theta}_k^{[i]} = \theta_{k-1}^{[i]} - \frac{r^{[i]}}{k^\rho} z_{k-1}^{[i]}$ is executed for all $k \in [k_*^{[i]} + 1, k_{ls}^{[i]}]$, when $k_{ls}^{[i]} \geqslant k_*^{[i]} + 1$.

Denote $g : \mathbb{R} \to \mathbb{R}$ such that $g(\lambda) \triangleq \eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]})$. Then by chain rule, we

have

$$\frac{d}{d\lambda}g(\lambda) = -\langle \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle.$$

Therefore, we have

$$\begin{aligned}\eta(\theta_k^{[i]}) - \eta(\theta_{k-1}^{[i]}) = \eta(\theta_{k-1}^{[i]} - \frac{r^{[i]}}{k^\rho} z_{k-1}^{[i]}) - \eta(\theta_{k-1}^{[i]}) = g(\frac{r^{[i]}}{k^\rho}) - g(0) = \int_0^{\frac{r^{[i]}}{k^\rho}} \frac{d}{d\lambda} g(\lambda) d\lambda \\ = -\int_0^{\frac{r^{[i]}}{k^\rho}} \langle \nabla\eta(\theta_{k-1}^{[i]} - \lambda z_{k-1}^{[i]}), z_{k-1}^{[i]}\rangle d\lambda.\end{aligned}$$

Combining this with Lemma 5.4.4, we have

$$\eta(\theta_k^{[i]}) - \eta(\theta_{k-1}^{[i]}) \leqslant -\frac{r^{[i]}}{k^\rho}\big((1 - L_{\nabla\eta}\frac{r^{[i]}}{k^\rho})\|z_{k-1}^{[i]}\|^2 - \|\xi_{k-1}^{[i]}\|\|z_{k-1}^{[i]}\|\big).$$

For notational simplicity, we denote

$$\delta_k^{[i]} \triangleq \eta(\theta_k^{[i]}) - \eta(\theta_{k-1}^{[i]}),\ b_{k-1}^{[i]} \triangleq \frac{r^{[i]}}{k^\rho}\|z_{k-1}^{[i]}\|, a_{k-1}^{[i]} \triangleq \frac{r^{[i]}}{k^\rho}(1 - L_{\nabla\eta}\frac{r^{[i]}}{k^\rho})\|z_{k-1}^{[i]}\|^2.$$

Therefore, the above inequality can be rewritten to

$$\delta_k^{[i]} \leqslant -a_{k-1}^{[i]} + \|\xi_{k-1}^{[i]}\| b_{k-1}^{[i]}. \tag{5.18}$$

Combining Lemma 5.4.3 and Markov's inequality renders

$$\begin{aligned}\mathbb{E}[\|\xi_k^{[i]}\|] = \int_0^{\sigma^{[i]}} P\Big(\|\xi_k^{[i]}\| > t\Big)dt + \int_{\sigma^{[i]}}^{\infty} P\Big(\|\xi_k^{[i]}\| > t\Big)dt \\ \leqslant \sigma^{[i]} + \int_{\sigma^{[i]}}^{\infty} \frac{(\sigma^{[i]})^2}{t^2} dt = 2\sigma^{[i]}.\end{aligned}$$

for all $k \geqslant 1$. Therefore, combining this with (5.18) implies

$$\begin{aligned}\mathbb{E}[\delta_k^{[i]} \mid z_{k-1}^{[i]}] \leqslant \mathbb{E}[-a_{k-1}^{[i]} + \|\xi_{k-1}^{[i]}\| b_{k-1}^{[i]} \mid z_{k-1}^{[i]}] = -a_{k-1}^{[i]} + b_{k-1}^{[i]}\mathbb{E}[\|\xi_{k-1}^{[i]}\|] \\ \leqslant -a_{k-1}^{[i]} + 2b_{k-1}^{[i]}\sigma^{[i]}.\end{aligned} \tag{5.19}$$

Since $k \in [k_*^{[i]} + 1, k_{ls}^{[i]}]$, $\|z_{k-1}^{[i]}\| \geqslant q^{[i]}$. Plugging in the definitions of $a_{k-1}^{[i]}$ and

$b_{k-1}^{[i]}$ and combining with $r^{[i]} \leqslant \frac{1}{2L_{\nabla\eta}}$ renders

$$\frac{a_{k-1}^{[i]}}{b_{k-1}^{[i]}} = (1 - L_{\nabla\eta} r^{[i]}/k^{\rho})\|z_{k-1}^{[i]}\| \geqslant \frac{(q^{[i]})}{2}. \tag{5.20}$$

Since $q^{[i]} > 4\sigma^{[i]}$, (5.20) renders that $\frac{a_{k-1}^{[i]}}{b_{k-1}^{[i]}} \geqslant 2\sigma^{[i]}$ and hence $-a_{k-1}^{[i]} + 2b_{k-1}^{[i]}\sigma^{[i]} \leqslant 0$ for $k \in [k_*^{[i]} + 1, k_{ls}^{[i]}]$. Then combining this with (5.19) renders $\mathbb{E}[\delta_k^{[i]} \mid z_{k-1}^{[i]}] \leqslant 0$, which implies

$$\mathbb{E}[\delta_k^{[i]}] = \int \mathbb{E}[\delta_k^{[i]} \mid z_{k-1}^{[i]}] p(z_{k-1}^{[i]}) dz_{k-1}^{[i]} \leqslant 0, \tag{5.21}$$

for all $k \in [k_*^{[i]} + 1, k_{ls}^{[i]}]$.

Notice that the definition of $\delta_k^{[i]}$ renders

$$\eta^{[i]}(\theta_k^{[i]}) - \eta^{[i]}(\theta_{k_*^{[i]}}^{[i]}) = \sum_{k'=k_*^{[i]}+1}^{k} \delta_{k'}^{[i]},$$

for any $k \geqslant k_*^{[i]} + 1$. Then by (5.21) we have

$$\mathbb{E}[\eta^{[i]}(\theta_k^{[i]}) - \eta^{[i]}(\theta_{k_*^{[i]}}^{[i]})] = \mathbb{E}[\sum_{k'=k_*^{[i]}+1}^{k} \delta_{k'}^{[i]}] = \sum_{k'=k_*^{[i]}+1}^{k} \mathbb{E}[\delta_{k'}^{[i]}] \leqslant 0.$$

The proof is conluded. ■

5.4.2.2 Proof of (T3) in Theorem 5.3.6

Lemma 5.4.6 shows that $k_{ls}^{[i]}$ exists almost surely. Therefore, Lines 25 and 12 implies that $\theta_k^{[i]} = \hat{\theta}_k^{[i]} = \theta_{k-1}^{[i]}$ for all $k \geqslant k_{ls}^{[i]} + 1$ and hence $\lim_{k\to\infty} \theta_k^{[i]} = \theta_{\infty}^{[i]} = \theta_{k_{ls}^{[i]}}^{[i]}$. ■

5.4.2.3 Proof of (T4) in Theorem 5.3.6

Notice that for any $k, k' \geqslant 1$ it holds that

$$\mathbb{E}[\eta(\theta_k^{[i]}) - \eta(\theta_{k'}^{[j]})] = \mathbb{E}[\eta(\theta_k^{[i]}) - y_k^{[i]} + y_k^{[i]} - \eta(\theta_{k'}^{[j]}) - y_{k'}^{[j]} + y_{k'}^{[j]}].$$

Since estimation error $\eta(\theta_k^{[i]}) - y_k^{[i]}$ is independent of $\theta_k^{[i]}$ and Assumptions 5.2.1 and 5.2.2 imply $\mathbb{E}[\eta(\theta_k^{[i]}) - y_k^{[i]}] = 0$, the above equality becomes

$$\mathbb{E}[\eta(\theta_k^{[i]}) - \eta(\theta_{k'}^{[j]})] = \mathbb{E}[y_k^{[i]} - y_{k'}^{[j]}]. \tag{5.22}$$

Recall that Lemma 5.4.6 shows that $\theta_{k_{ls}^{[i]}}^{[i]}$ exists almost surely. Denote $j^* \triangleq \arg\min_{j\in\mathcal{V}} \eta(\theta_{k_{ls}^{[j]}}^{[j]})$. Since learner i does not execute Line 20 at iteration $k_{ls}^{[i]}$, we have

$$y_{k_{ls}^{[j^*]}}^{[j^*]} + b_\gamma^{[j^*]} \geqslant \min\{y_{k_{ls}^{[i]}}^{[i]} - b_\gamma^{[i]}, \zeta_{k_{ls}^{[i]}}^{[i]}\}.$$

We now distinguish two cases.

Case 1: $y_{k_{ls}^{[i]}}^{[i]} - b_\gamma^{[i]} < \zeta_{k_{ls}^{[i]}}^{[i]}$. This implies $y_{k_{ls}^{[j^*]}}^{[j^*]} + b_\gamma^{[j^*]} \geqslant y_{k_{ls}^{[i]}}^{[i]} - b_\gamma^{[i]}$, or

$$y_{k_{ls}^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]} \leqslant b_\gamma^{[i]} + b_\gamma^{[j^*]} \leqslant 2\max_{j\in\mathcal{V}} b_\gamma^{[j]}. \tag{5.23}$$

Case 2: $\zeta_{k_{ls}^{[i]}}^{[i]} \leqslant y_{k_{ls}^{[i]}}^{[i]} - b_\gamma^{[i]}$. Line 26 implies

$$\begin{aligned} y_{k_{ls}^{[j^*]}}^{[j^*]} + b_\gamma^{[j^*]} &\geqslant \zeta_{k_{ls}^{[i]}}^{[i]} = \zeta_{k_*^{[i]}}^{[i]} = y_l^{[j]}, \\ (j,l) &= \arg\min_{i\in\mathcal{V}, l'=0,\cdots,k_*^{[i]}-1} y_{l'}^{[i]} + b_\gamma^{[i]} \end{aligned}$$

Therefore, $y_l^{[j]} - y_{k_{ls}^{[j^*]}}^{[j^*]} \leqslant b_\gamma^{[j^*]}$. Recall that Line 21 implies $\theta_{k_*^{[i]}}^{[i]} = \theta_l^{[j]}$ and hence $y_{k_*^{[i]}}^{[i]} = y_l^{[j]}$. This renders

$$y_{k_*^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]} \leqslant b_\gamma^{\max}. \tag{5.24}$$

Lemma 5.4.7 and (5.22) render $\mathbb{E}[y_{k_{ls}^{[i]}}^{[i]} - y_{k_*^{[i]}}^{[i]}] = \mathbb{E}[\eta(\theta_{k_{ls}^{[i]}}^{[i]}) - \eta(\theta_{k_*^{[i]}}^{[i]})] \leqslant 0$. Combining this with (5.24) renders

$$\begin{aligned} \mathbb{E}[y_{k_{ls}^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]}] &= \mathbb{E}[y_{k_{ls}^{[i]}}^{[i]} - y_{k_*^{[i]}}^{[i]} + y_{k_*^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]}] \\ &= \mathbb{E}[y_{k_{ls}^{[i]}}^{[i]} - y_{k_*^{[i]}}^{[i]}] + \mathbb{E}[y_{k_*^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]}] \leqslant \mathbb{E}[y_{k_*^{[i]}}^{[i]} - y_{k_{ls}^{[j^*]}}^{[j^*]}] \leqslant b_\gamma^{\max}. \end{aligned} \tag{5.25}$$

By (5.22), combining (5.23) and (5.25) renders

$$\mathbb{E}[\eta(\theta^{[i]}_{k^{[i]}_{ls}}) - \eta(\theta^{[j^*]}_{k^{[j^*]}_{ls}})] = \mathbb{E}[y^{[i]}_{k^{[i]}_{ls}} - y^{[j^*]}_{k^{[j^*]}_{ls}}] \leqslant 2b^{\max}_{\gamma}.$$

Recall that $k^{[i]}_*$ is the last time adopting estimates from the Cloud (Lines 20-23 are executed). Figure 5.2 implies that $\theta^{[i]}_k = \hat{\theta}^{[i]}_k = \theta^{[i]}_{k-1}$ for all $k \geqslant k^{[i]}_{ls} + 1$ and hence $\lim_{k\to\infty} \theta^{[i]}_k = \theta^{[i]}_\infty = \theta^{[i]}_{k^{[i]}_{ls}}$. Therefore, we have $\theta^{[i]}_\infty = \theta^{[i]}_{k^{[i]}_{ls}}$ for all $i \in \mathcal{V}$. Hence, the above inequality implies that, for any $i \in \mathcal{V}$,

$$\mathbb{E}[\eta(\theta^{[i]}_\infty) - \eta(\theta^{[j^*]}_\infty)] = \mathbb{E}[y^{[i]}_\infty - y^{[j^*]}_\infty] \leqslant 2b^{\max}_{\gamma}. \qquad \blacksquare$$

.

5.4.2.4 Proof of (T5) in Theorem 5.3.6

Since Lemma 5.4.6 shows that $k^{[i]}_{ls}$ exists almost surely, by (5.9), we have $k^{[i]}_{fs}$ exists almost surely. Recall that $k^{[i]}_{fs} + 1 \leqslant k^{[i]}_1$ from (5.9). Notice that at iteration $k^{[i]}_{fs}$, agent i stops its local gradient descent, and its estimate remains the same for the following iterations until it adopts an estimate from the Cloud. Since $\theta^{[i]}_\infty \neq \theta^{[i]}_{k^{[i]}_{fs}}$, agent i adopts estimates from the Cloud, executing Lines 20-23, at least once after iteration $k^{[i]}_{fs}$. This implies that $k^{[i]}_* \geqslant k^{[i]}_1 \geqslant 1$ and

$$\theta^{[i]}_{k^{[i]}_{fs}} = \theta^{[i]}_{k^{[i]}_{fs}+1} = \cdots = \theta^{[i]}_{k^{[i]}_1 - 1}. \tag{5.26}$$

Recall that (T3) of Theorem 5.3.6 shows that $\theta^{[i]}_\infty$ exists almost surely. By Lemma 5.4.5, $k^{[i]}_*$ exists. Since $k^{[i]}_* \geqslant k^{[i]}_1 \geqslant 1$, Lines 20-23 imply that there exists $(j_1, l_1) = \arg\min_{i\in\mathcal{V}, l'=0,\cdots,k^{[i]}_1-1} y^{[i]}_{l'} + b^{[i]}_\gamma$ such that $y^{[j_1]}_{l_1} + b^{[j_1]}_\gamma < y^{[i]}_{k^{[i]}_1-1} - b^{[i]}_\gamma$. Consider $(j_*, l_*) = \arg\min_{i\in\mathcal{V}, l'=0,\cdots,k^{[i]}_*-1} y^{[i]}_{l'} + b^{[i]}_\gamma$. It is obvious that $y^{[j_*]}_{l_*} + b^{[j_*]}_\gamma \leqslant y^{[j_1]}_{l_1} + b^{[j_1]}_\gamma < y^{[i]}_{k^{[i]}_1-1} - b^{[i]}_\gamma$, or

$$y^{[j^*]}_{l_*} - y^{[i]}_{k^{[i]}_1-1} < -(b^{[i]}_\gamma + b^{[j_*]}_\gamma). \tag{5.27}$$

Since learner i adopts the estimate from the Cloud, i.e., executes Lines 20-23, at iteration $k^{[i]}_*$, Line 21 implies $\theta^{[i]}_{k^{[i]}_*} = \theta^{[j_*]}_{l_*}$. Following the same logic of (5.22) and

combining with (5.27), we have

$$\begin{aligned}\mathbb{E}[\eta(\theta^{[i]}_{k^{[i]}_*}) - \eta(\theta^{[i]}_{k^{[i]}_1-1})] &= \mathbb{E}[\eta(\theta^{[i]}_{k^{[i]}_*}) - y^{[j^*]}_{l_*} + y^{[j^*]}_{l_*} - \eta(\theta^{[i]}_{k^{[i]}_1-1}) + y^{[i]}_{k^{[i]}_1-1} - y^{[i]}_{k^{[i]}_1-1}] \\ &= \mathbb{E}[y^{[j^*]}_{l_*} - y^{[i]}_{k^{[i]}_1-1}] < -(b^{[i]}_\gamma + b^{[j_*]}_\gamma).\end{aligned}$$

Combining this with Lemma 5.4.7 renders

$$\begin{aligned}\mathbb{E}[\eta(\theta^{[i]}_\infty) - \eta(\theta^{[i]}_{k^{[i]}_1-1})] &= \mathbb{E}[\eta(\theta^{[i]}_\infty) - \eta(\theta^{[i]}_{k^{[i]}_*}) + \eta(\theta^{[i]}_{k^{[i]}_*}) - \eta(\theta^{[i]}_{k^{[i]}_1-1})] \\ &= \mathbb{E}[\eta(\theta^{[i]}_\infty) - \eta(\theta^{[i]}_{k^{[i]}_*})] + \mathbb{E}[\eta(\theta^{[i]}_{k^{[i]}_*}) - \eta(\theta^{[i]}_{k^{[i]}_1-1})] \\ &< -(b^{[i]}_\gamma + b^{[j_*]}_\gamma) \leqslant -2b^{\min}_\gamma.\end{aligned}$$

Combining this with $\theta^{[i]}_{k^{[i]}_{fs}} = \theta^{[i]}_{k^{[i]}_1-1}$ in (5.26), the proof is concluded. ■

5.4.3 Proof of Theorem 5.3.7

For notational simplicity, we define two closed neighborhoods for each $\theta_* \in \Theta_*$: $\Psi(\theta_*) \triangleq \mathcal{B}(\theta_*, 4\epsilon_0(\theta_*) + 2\sqrt{\epsilon_0(\theta_*)})$ and $\Psi_1(\theta_*) \triangleq \mathcal{B}(\theta_*, 2\epsilon_0(\theta_*))$. Then the proof of the theorem is composed of four parts. First, we assume that there exists some $i \in \mathcal{V}$ such that $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$ for some $\theta_* \in \Theta_*$ and derive the probabilistic upper bound of $\eta(\theta^{[i]}_{k^{[i]}_{fs}}) - \eta_*$ in part (i). Then in part (ii) we further derive the probabilistic upper bound of $\eta(\theta^{[i]}_\infty) - \eta_*$ leveraging the result of Pareto improvement in [T5] of in Theorem 5.3.6. In part (iii), we extend the upper bound to $\eta(\theta^{[j]}_\infty) - \eta_*$ for all $j \in \mathcal{V}$ leveraging the result of Almost-consensus in [T4] of in Theorem 5.3.6. Finally, we characterize the probability of $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$.

Part (i): Probabilistic upper bound of $\eta(\theta^{[i]}_{k^{[i]}_{fs}}) - \eta_*$. Suppose there exists $i \in \mathcal{V}$ such that $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$ for some $\theta_* \in \Theta_*$. The definition of $k^{[i]}_{fs}$ renders that $\|z^{[i]}_{k^{[i]}_{fs}}\| < q^{[i]}$. Combining this with Lemma 5.4.2 renders that

$$P\Big(\|\nabla\eta(\theta^{[i]}_{k^{[i]}_{fs}})\| \leqslant q^{[i]} + \epsilon_1\Big) \geqslant 1 - \frac{(\sigma^{[i]})^2}{\epsilon_1^2}. \tag{5.28}$$

Combining (5.4) with Cauchy-Schwartz inequality implies

$$\alpha\|\theta-\theta_*\| \leqslant \|\nabla\eta(\theta)\|,\ \forall\theta\in\mathcal{K}(\theta_*). \tag{5.29}$$

Since $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*) \subset \mathcal{K}(\theta_*)$, combining (5.28) with inequality (5.29) renders

$$P\Big(\|\theta^{[i]}_{k^{[i]}_{fs}}-\theta_*\| \leqslant \frac{q^{[i]}+\epsilon_1}{\alpha} \mid \theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)\Big) \geqslant 1-\frac{(\sigma^{[i]})^2}{\epsilon_1^2}.$$

Combining this with Assumption 5.3.2 further renders

$$P\Big(\eta(\theta^{[i]}_{k^{[i]}_{fs}})-\eta_* \leqslant \frac{L_\eta(q^{[i]}+\epsilon_1)}{\alpha} \mid \theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)\Big) \geqslant 1-\frac{(\sigma^{[i]})^2}{\epsilon_1^2}. \tag{5.30}$$

Part (ii): Probabilistic upper bound of $\eta(\theta^{[i]}_\infty)-\eta_*$. Denote $\delta^{[i]} \triangleq \eta(\theta^{[i]}_\infty)-\eta(\theta^{[i]}_{k^{[i]}_{fs}})$. Notice that the definition of J_E renders that $J_E \in [0,1]$. Then the definition of η renders that $\eta\in[0,1]$. Then it holds that $\delta^{[i]}\in[-1,1]$. Theorem 5.3.6 [T3] implies that $\mathbb{E}[\delta^{[i]} \mid \theta^{[i]}_\infty \neq \theta^{[i]}_{k^{[i]}_{fs}}] \leqslant -2b^{\min}_\gamma$. Then let $\epsilon_2>0$, by leveraging Hoeffding's inequality in Theorem 5.4.1, we have

$$\begin{aligned}
&P\Big(\eta(\theta^{[i]}_\infty)-\eta(\theta^{[i]}_{k^{[i]}_{fs}}) \geqslant \epsilon_2\Big)\\
&\leqslant P\Big(\eta(\theta^{[i]}_\infty)-\eta(\theta^{[i]}_{k^{[i]}_{fs}}) \geqslant \epsilon_2 \mid \theta^{[i]}_\infty \neq \theta^{[i]}_{k^{[i]}_{fs}}\Big)\\
&\leqslant P\Big(\delta^{[i]}-\mathbb{E}[\delta^{[i]}|\theta^{[i]}_\infty \neq \theta^{[i]}_{k^{[i]}_{fs}}] \geqslant \epsilon_2+2b^{\min}_\gamma \mid \theta^{[i]}_\infty \neq \theta^{[i]}_{k^{[i]}_{fs}}\Big)\\
&\leqslant 2\exp\big(-2(\epsilon_2+2b^{\min}_\gamma)^2\big) \leqslant 2\exp\big(-2\epsilon_2^2\big).
\end{aligned}$$

Combining this with (5.30) renders that

$$\eta(\theta^{[i]}_\infty)-\eta_* \leqslant \frac{L_\eta(q^{[i]}+\epsilon_1)}{\alpha}+\epsilon_2 \tag{5.31}$$

with probability at least $(1-\frac{(\sigma^{[i]})^2}{\epsilon_1^2})(1-2\exp\big(-2\epsilon_2^2\big)) \geqslant 1-\frac{(\sigma^{[i]})^2}{\epsilon_1^2}-2\exp\big(-2\epsilon_2^2\big)$, given $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$.

Part (iii): Probabilistic upper bound of $\eta(\theta^{[j]}_\infty)-\eta_*$ *for all* $j\in\mathcal{V}$. Denote $\delta_\infty \triangleq \max_{j\in\mathcal{V}}\eta(\theta^{[j]}_\infty)-\min_{j\in\mathcal{V}}\eta(\theta^{[j]}_\infty)$. It is obvious that $\delta_\infty \geqslant 0$. Then combining

Markov inequality with Theorem 5.3.6 [T4], we have

$$P\Big(\delta_\infty \geqslant 2\epsilon_3 b_\gamma^{\max}\Big) \leqslant \frac{1}{\epsilon_3}. \tag{5.32}$$

Combining this with (5.31) renders that, given there exists $i \in \mathcal{V}$ such that $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$, it holds that, for all $j \in \mathcal{V}$,

$$\eta(\theta^{[j]}_\infty) - \eta_* \leqslant \frac{L_\eta(q^{[i]} + \epsilon_1)}{\alpha} + \epsilon_2 + 2\epsilon_3 b_\gamma^{\max} \tag{5.33}$$

with probability at least $1 - \frac{(\sigma^{[i]})^2}{\epsilon_1^2} - 2\exp(-2\epsilon_2^2) - \frac{1}{\epsilon_3}$.

Part (iv): Probability of there exists $i \in \mathcal{V}$ such that $\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*)$. Given Assumption 5.3.4 holds, Theorem 4 in [159] indicates that for each $\theta_* \in \Theta_*$, it holds that

$$P\Big(\theta^{[i]}_{k^{[i]}_{fs}} \in \Psi(\theta_*) \mid \theta_0^{[i]} \in \Psi_1(\theta_*)\Big) \geqslant 1 - \frac{R_*(\theta_*; \sigma^{[i]})\Gamma}{\epsilon_0(\theta_*)}, \tag{5.34}$$

where $R(\theta_*; \sigma^{[i]}) \triangleq L_\eta^2 + (1 + (4\epsilon_0(\theta_*) + 2\sqrt{\epsilon_0(\theta_*)})^2)(\sigma^{[i]})^2$ and $\Gamma \triangleq r^{[i]} \sum_{k=1}^\infty \frac{1}{k^{2\rho}}$.

Denote $\bar{\beta} \triangleq \frac{\beta(\Theta_0 \cap [\cup_{\theta_* \in \Theta_*} \Psi_1(\theta_*)])}{\beta(\Theta_0)}$. Since $\beta(\Theta_0 \cap [\cup_{\theta_* \in \Theta_*} \Psi_1(\theta_*)]) > 0$, it is obvious that $\bar{\beta} \in (0, 1]$. Since $\theta_0^{[i]}$ is uniformly sampled over compact set Θ_0, we have $P\Big(\theta_0^{[i]} \in \Psi_1(\theta_*) \mid \theta_* \in \Theta_*\Big) = \bar{\beta}$. Since there are $|\mathcal{V}|$ learners in $\mathcal{V}$ and $\theta_0^{[i]}$ are independently sampled for all $i \in \mathcal{V}$, then we further have

$$\begin{aligned} &P\Big(\exists i \in \mathcal{V} \text{ such that } \theta_0^{[i]} \in \Psi_1(\theta_*) \mid \theta_* \in \Theta_* \cap \Theta_0\Big) \\ &= 1 - P\Big(\theta_0^{[i]} \notin \Psi_1(\theta_*; \epsilon),\ \forall i \in \mathcal{V} \mid \theta_* \in \Theta_* \cap \Theta_0\Big) = 1 - (1 - \bar{\beta})^{|\mathcal{V}|}. \end{aligned} \tag{5.35}$$

Combining (5.34) with (5.35) renders

$$\begin{aligned} &P\Big(\exists i \in \mathcal{V} \text{ such that } \theta_\infty^{[i]} \in \Psi(\theta_*) \mid \theta_* \in \Theta_* \cap \Theta_0\Big) \\ &\geqslant 1 - (1 - \bar{\beta})^{|\mathcal{V}|} - \max_{\theta_* \in \Theta_*} \frac{R_*(\theta_*; \sigma^{\max})\Gamma}{\epsilon_0(\theta_*)}. \end{aligned} \tag{5.36}$$

Combining (5.36) with (5.33) concludes the proof. ■

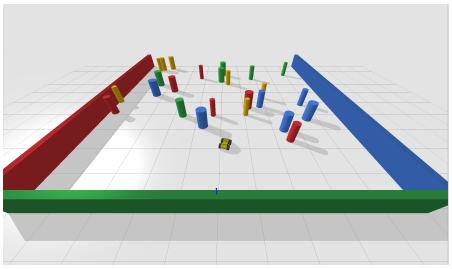


Figure 5.3: A sample environment in PyBullet

5.5 Simulation

In this section, we conduct a set of Monte Carlo simulations to evaluate the performance of the FedGen algorithm in the PyBullet simulator [163]. All the simulations are conduct in Python on an Intel Core i5 CPU, 4.10 GHz, with 16 GB of RAM.

(Environment configuration). The evaluation is conducted using Zermelo's navigation problem [128] in a 2D space, where the environments are randomly generated. A sample of the environments is shown in Figure 5.3. Each environment E consists of n_{obs} cylinder obstacles and three walls as the boundary of the 2D environment with horizontal coordinate $x_1 \in [-5, 5]$ and vertical coordinate $x_2 \in [0, 10]$. The environments are generated by sampling the obstacle number n_{obs} uniformly between 15 and 30, and then independently sampling the centers of the cylinders from a uniform distribution over the ranges $[-5, 5] \times [2, 10]$. The radius of each obstacle is sampled independently from a uniform distribution over $[0.1, 0.25]$. The goal of the robot is to reach the open end of the environment while avoiding collision with the walls and the obstacles.

(Robot dynamics). We consider a four-wheel robot with constant speed $v = 2.5$ and length $L = 0.08$ subject to unknown environment-specific disturbances d_E. The dynamics of the robot with state $x = [x_1, x_2, x_3]$ is given by $\dot{x}_1 = v\cos(x_3) + d_E(x_1, x_2)$, $\dot{x}_2 = v\sin(x_3)$, $\dot{x_3} = \tan(u)/L$, where x_3 is the heading of the robot, control $u \in [-0.25\pi, 0.25\pi]$, and d_E is generated using the Von Karman power spectral density function as described in [131] representing the road texture disturbance (e.g., bumps and slippery surface) in environment E.

(Sensor model). In the simulation, the robots are equipped with a sensor able to obtain the robot's state information x and a depth sensor (e.g., LIDAR) able to measure the distances between the robot and the obstacles. The sensors are perfect. The readings of the depth sensor depend on the environment E and the

state of the robot. Specifically, the output of the sensor has 20 entries, where each entry ϕ corresponds to the distance measurement at angle $x_3 - \pi/3 + (\phi - 1)\pi/60$ with $\phi = 1, \cdots, 20$. The measurement $h_\phi(x, \mathcal{X}_{O,E})$ provides the shortest distance between the obstacles, if there is any, at the angle of entry ϕ of the robot and the robot at location (x_1, x_2). The sensing range is 5, i.e., $h_\phi(x, \mathcal{X}_{O,E}) \in [0, 5]$. That is, the observation function is given by $h(x, \mathcal{X}_{O,E}) = [x, h_1(x, \mathcal{X}_{O,E}), \cdots, h_{20}(x, \mathcal{X}_{O,E})]$.

5.5.1 Training

We consider a deep neural network-based control policy π_θ, that is parameterized by θ, the weights of the neural network. Note that the control policy is periodic in φ. Thus, the input φ is replaced by two inputs $\sin(\varphi)$ and $\cos(\varphi)$. During training, especially during the early phase, the original cost functional $J_E(x_{int}, \pi_\theta)$ may have zero gradient for some initial state x_{int} since collisions with obstacles dominate most of the trial runs. Therefore, to facilitate training, we consider the surrogate $\hat{J}_E(x_{int}, \theta) \triangleq 0.1\rho_E(x_{int}, \pi_\theta) + J_E(x_{int}, \pi_\theta)$, where $\rho_E(x_{int}, \pi_\theta) \triangleq \min_{x_G \in \mathcal{X}_{G,E}} \|x(t_{end}(x_{int}, \pi_\theta; E)) - x_G\|$ is the distance between the location of the first collision and the goal region. The cost $\rho_E(x_{int}, \pi_\theta)$ is to drive the robot approaching the goal without collision, and the cost $J_E(x_{int}, \pi_\theta)$ is to minimize the arrival time when the robot is able to safely reach the goal.

Since it is challenging to derive the analytical expression of $\nabla \hat{J}_E(x_{int}, \theta)$, we approximate it by natural evolution strategies [154, 164]. In particular, we suppose θ follows a multivariate Gaussian distribution such that $\theta \sim \mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^{n_\theta}$ and diagonal covariance $\Sigma \in \mathbb{R}^{n_\theta \times n_\theta}$. Let $\sigma \in \mathbb{R}^{n_\theta}$ be a vector aggregating the square-root of the diagonal elements of Σ. The gradients of $\mathbb{E}_\theta \left[\hat{J}_E(x_{int}, \pi_\theta)\right]$ with respect to μ and σ are

$$\nabla_\mu \mathop{\mathbb{E}}_{\theta \sim \mathcal{N}(\mu,\Sigma)} \left[\hat{J}_E(x_{int}, \pi_\theta)\right] = \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(0,I)} \left[\hat{J}_E(x_{int}, \pi_{\mu+\sigma\odot\epsilon})\epsilon\right] \oslash \sigma,$$
$$\nabla_\sigma \mathop{\mathbb{E}}_{\theta \sim \mathcal{N}(\mu,\Sigma)} \left[\hat{J}_E(x_{int}, \pi_\theta)\right] = \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(0,I)} \left[\hat{J}_E(x_{int}, \pi_{\mu+\sigma\odot\epsilon})(\epsilon \odot \epsilon - \mathbf{1})\right] \oslash \sigma,$$

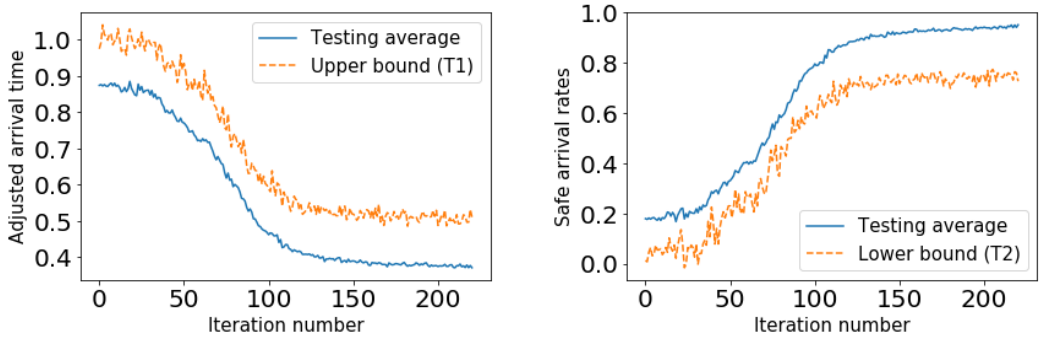
where $\oslash$ is the element-wise division, $\odot$ is the elementwise product, and $\mathbf{1}$ is a vector of 1's with dimension n_θ. We approximate the expectation by collecting 30 samples of $\epsilon \sim \mathcal{N}(0, I)$ and taking the average. To reduce the variance in the

expectation approximation, antithetic sampling [165] is employed. That is, the update of θ is then replaced by the updates of μ and σ, and μ is returned as the estimate of θ.

(Selection of hyperparameters). The neural network control policy consists of an input layer of size 24, followed by 3 hidden layers of size 20 with ReLu nonlinearities and an output layer of size 1. We pick $n_{\mathcal{E}}^{[i]} = 10$, $n_{int|\mathcal{E}}^{[i]} = 1$, $\gamma = 0.01$, $r = 0.01$, $L_\eta = 0.1$, $q^{[i]} = 0.04$, and 8 learners, i.e., $|\mathcal{V}| = 8$, for the experiments. The generalized performance in unseen environments is defined as an expectation over all possible environments, which cannot be obtained exactly. Therefore, we estimate the generalized performances using 10^4 sample environments.

5.5.2 Results

(Generalization and convergence). Figure 5.4 compares the upper bound on the expected normalized arrival time (T1) and the lower bound on the safe arrival rate (T2) in Theorem 5.3.1 respectively with the actual expected normalized arrival time and the actual safe arrival rate of learner 1. Other learners have similar behaviors. As the figure illustrates, the upper bound and the lower bound derived in the theorem are valid. This shows that the control policy trained can zero-shot generalize well to the 10^4 unseen environments. Converging behavior is also obvious in Figure 5.4, which aligns with (T3) of Theorem 5.3.6.



(a) Expected normalized arrival time (b) Safe arrival rate

Figure 5.4: Generalized performances to unseen environments

(Near consensus and Pareto improvement). In Table 5.2 below, we show the performances of the learners' estimates in terms of the expected distance-to-goal

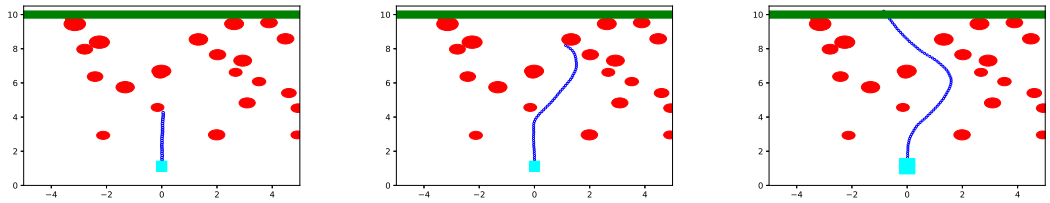
$0.1\rho_E$, the expected normalized arrival time J_E, and the expected safe arrival rate. We compare with the control policy at initialization ($\theta_0^{[i]}$), the control policy obtained without communication ($\theta_{k_{fs}^{[i]}}^{[i]}$), i.e., the control policy obtained by running FedGen using $\mathcal{V} = \{i\}$, and the final convergence ($\theta_\infty^{[i]}$) under FedGen. We can observe that all the expected costs, expected normalized arrival times and expected safe arrival rates at $\theta_\infty^{[i]}$ are roughly equal. This aligns with the almost consensus (T4) in Theorem 5.3.6. Furthermore, we can observe that all the expected costs and the expected normalized arrival times at $\theta_\infty^{[i]}$ are no larger than those of $\theta_0^{[i]}$ and $\theta_{k_{fs}^{[i]}}^{[i]}$, while the expected safe arrival rates at $\theta_\infty^{[i]}$ are no smaller than those at $\theta_0^{[i]}$ and $\theta_{k_{fs}^{[i]}}^{[i]}$. This shows that FedGen brings Pareto improvement for each learner through communication, which is also shown in (T5) of Theorem 5.3.6.

(Performance vs. the number of learners). Table 5.3 presents the expected distance-to-goal, normalized arrival time, and safe arrival rate of the limiting estimate $\theta_\infty^{[i]}$ when FedGen is run using different number of learners. The table shows that with more learners involved in FedGen, the performances of the control policies are better. This shows a stronger result than that in Theorem 5.3.7, where more learners can only improve the probability of achieving the optimality gap in (5.5).

Graphically, Figure 5.5 respectively shows the trajectories of the robot in a sample of unseen environments using learner 1's initial policy $\theta_0^{[1]}$, locally converged policy $\theta_{k_{fs}^{[1]}}^{[1]}$ and finally converged policy $\theta_\infty^{[1]}$. The red disks represent the obstacles. The cyan square represents the initial location. The green line represents the goal region. The blue curves are the trajectories of the robot. Both the initial control policy (Figure 5.5a) and the locally converged control policy (Figure 5.5b) cannot bring the robot to the open end, despite the locally converged control policy is able to drive the robot closer to the open end. Nevertheless, the path generated by the final control policy $\theta_\infty^{[1]}$ is able to drive the robot to the open end. This illustrates that FedGen helps the learners escape from their local minima and achieve better generalizability.

5.6 Conclusion

We propose FedGen, a federated reinforcement learning algorithm which allows a group of learners to collaboratively learn a single control policy for robot motion planning with zero-shot generalization. The problem is formulated as an expected cost minimization problem and solved in a federated manner. The proposed algorithm is able to provide zero-shot generalization guarantees on the performances of the local control policies in unseen environments as well as almost-sure convergence, almost consensus and Pareto improvement. The algorithm is evaluated using Monte Carlo simulations.



(a) Trajectory produced by $\theta_0^{[1]}$ (b) Trajectory produced by $\theta_{k_{fs}^{[1]}}^{[1]}$ (c) Trajectory produced by $\theta_\infty^{[1]}$

Figure 5.5: Comparison between initial policy, locally converged policy and globally converged policy

Algorithm 10 FedGen

Input: Local sample sizes: $n_{\mathcal{E}}^{[i]}, n_{int|\mathcal{E}}^{[i]}$; Kruzkov transform constant: α; Initial step size: $r^{[i]}$; Initial estimate: $\theta_0^{[i]}$; Threshold for gradient: $q^{[i]}$; Local bias: $b_\gamma^{[i]}$; Step exponent: $\rho \in (2/3, 1)$.
Init: $\zeta_0^{[i]} \leftarrow 1$, $\mathsf{Stop}_0^{[i]} \leftarrow \mathsf{False}$.
for $k = 1, 2, \cdots, K$ **do**
{Learner-based update}
  for $i \in \mathcal{V}$ **do**
    if $\mathsf{Stop}_{k-1}^{[i]} == \mathsf{False}$ **then**
      Collects $(y_{k-1}^{[i]}, z_{k-1}^{[i]})$
    end if
    Sends $(\theta_{k-1}^{[i]}, y_{k-1}^{[i]})$ to the Cloud
    if $\|z_{k-1}^{[i]}\| \geqslant q^{[i]}$ and $\mathsf{Stop}_{k-1}^{[i]} == \mathsf{False}$ **then**
      $\hat{\theta}_k^{[i]} \leftarrow \theta_{k-1}^{[i]} - \frac{r^{[i]}}{k^\rho} z_{k-1}^{[i]}$
    else
      $\hat{\theta}_k^{[i]} \leftarrow \theta_{k-1}^{[i]}$
      $(y_k^{[i]}, z_k^{[i]}) \leftarrow (y_{k-1}^{[i]}, z_{k-1}^{[i]})$
      $\mathsf{Stop}_k^{[i]} \leftarrow \mathsf{True}$
    end if
  end for
{Cloud update}
  $(j, l) \leftarrow \arg\min_{i \in \mathcal{V}, l' = 0, \cdots, k-1} y_{l'}^{[i]} + b_\gamma^{[i]}$
  Sends $(\theta_l^{[j]}, y_l^{[j]}, b_\gamma^{[j]})$ to all $i \in \mathcal{V}$
{Learner-based fusion}
  for $i \in \mathcal{V}$ **do**
    if $j \neq i$ and $y_l^{[j]} + b_\gamma^{[j]} < \min\{y_{k-1}^{[i]} - b_\gamma^{[i]}, \zeta_{k-1}^{[i]}\}$ and $\mathsf{Stop}_{k-1}^{[i]} == \mathsf{True}$ **then**
      $\theta_k^{[i]} \leftarrow \theta_l^{[j]}$
      $\zeta_k^{[i]} \leftarrow y_l^{[j]}$
      $\mathsf{Stop}_k^{[i]} \leftarrow \mathsf{False}$
    else
      $\theta_k^{[i]} \leftarrow \hat{\theta}_k^{[i]}$
      $\zeta_k^{[i]} \leftarrow \zeta_{k-1}^{[i]}$
    end if
  end for
end for

Learner ID (i)		1	2	3	4	5
Distance-to-goal $(0.1\mathbb{E}[\rho_E(x_{int}, \pi_{\theta^{[i]}})])$	Init($\theta_0^{[i]}$)	0.5198	0.5170	0.5208	0.5210	0.5148
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.0436	0.0396	0.0331	0.4810	0.4105
	Final($\theta_\infty^{[i]}$)	0.0374	0.0396	0.0331	0.0335	0.0353
Normalized arrival time $(\mathbb{E}[J_E(x_{int}; \pi_{\theta^{[i]}})])$	Init($\theta_0^{[i]}$)	0.8743	0.8761	0.8744	0.8782	0.8692
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.3759	0.3763	0.3701	0.8385	0.7815
	Final($\theta_\infty^{[i]}$)	0.3748	0.3763	0.3701	0.3679	0.3711
Safe arrival rate	Init($\theta_0^{[i]}$)	0.1802	0.1776	0.1800	0.1746	0.1876
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.9320	0.9408	0.9522	0.2314	0.3172
	Final($\theta_\infty^{[i]}$)	0.9386	0.9408	0.9522	0.9452	0.9432

Learner ID (i)		6	7	8
Distance-to-goal $(0.1\mathbb{E}[\rho_E(x_{int}, \pi_{\theta^{[i]}})])$	Init($\theta_0^{[i]}$)	0.5231	0.5237	0.5167
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.0341	0.3992	0.4989
	Final($\theta_\infty^{[i]}$)	0.0341	0.0363	0.0335
Normalized arrival time $(\mathbb{E}[J_E(x_{int}; \pi_{\theta^{[i]}})])$	Init($\theta_0^{[i]}$)	0.8748	0.8797	0.8732
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.3700	0.7622	0.8569
	Final($\theta_\infty^{[i]}$)	0.3700	0.3716	0.3704
Safe arrival rate	Init($\theta_0^{[i]}$)	0.1794	0.1724	0.1818
	Local($\theta_{k_{fs}^{[i]}}^{[i]}$)	0.9450	0.3426	0.2054
	Final($\theta_\infty^{[i]}$)	0.9450	0.9428	0.9468

Table 5.2: The expected distance-to-goal, normalized arrival times, safe arrival rates of the estimates at initialization, local convergence and final convergence.

Number of learners ($\lvert\mathcal{V}\rvert$)	1	2	4
Distance-to-goal $\left(0.1\mathbb{E}[\rho_E(x_{int}, \pi_{\theta_\infty^{[i]}})]\right)$	0.4989	0.1548 ± 0.0132	0.1391 ± 0.0247
Normalized arrival time $\left(\mathbb{E}[J_E(x_{int}; \pi_{\theta_\infty^{[i]}})]\right)$	0.8569	0.4997 ± 0.0158	0.4910 ± 0.0234
Safe arrival rate	0.2054	0.7325 ± 0.0225	0.7563 ± 0.0338

Number of learners ($\lvert\mathcal{V}\rvert$)	6	8
Distance-to-goal $\left(0.1\mathbb{E}[\rho_E(x_{int}, \pi_{\theta_\infty^{[i]}})]\right)$	0.0760 ± 0.0126	0.0354 ± 0.0021
Normalized arrival time $\left(\mathbb{E}[J_E(x_{int}; \pi_{\theta_\infty^{[i]}})]\right)$	0.4111 ± 0.0169	0.3715 ± 0.0027
Safe arrival rate	0.8717 ± 0.0256	0.9442 ± 0.0038

Table 5.3: The expected distance-to-goal, normalized arrival times, safe arrival rates of the limiting estimates for different number of learners. The table shows the average values over the learners plus-minus one standard deviation.

Chapter 6

Online safe meta reinforcement learning

6.1 Introduction

Chapter 4 considers online learning where a group of robots aim to ensure safety and mission completion amid unknown uncertainties. Chapter 5 considers offline learning and investigates how a group of robot learners can collaboratively learn a control policy with good zero-shot generalization. In this chapter, we consider how online learning and offline learning can be combined to improve the performances of a robot through a sequence of tasks. Particularly, we consider how sample-efficient data collection and policy adaptation together with all-time safety can be achieved throughout the process.

Meta reinforcement learning (MRL) aims to address the fundamental problem of quickly learning an optimal control policy in a new task using less data and less training time [141, 142, 166, 143]. The problem is usually formulated as an unconstrained optimization problem over a meta control policy and an adaptation procedure, where the objective function is the expected performance of the adapted control policy in a new task.

Offline MRL learns a meta control policy using a fix set of tasks [141, 144, 167]. Online MRL, in contrast, updates the meta control policy sequentially by the arrival of training tasks [168, 169, 170]. A large number of physical real-world

agents operate in changing environments, and therefore online MRL is desired for these applications. To successively optimize the control policies, MRL algorithms sample trajectories of the policies to evaluate the performance of the policies and determine how the update should be conducted. However, trajectory samples can be limited in these applications since not only it can be costly to generate trajectories for physical applications, but also there is only limited time to online roll out the trajectories in a task since the environments are changing. Therefore, sample efficiency should be considered in these applications such that the update of the meta policy (and the adapted policy) can be effectively done with as few samples as possible.

Many physical real-world applications (e.g., mobile robots) are safety-critical. These applications require that physical agents satisfy certain safety constraints (e.g., collision avoidance) during the entire deployment stage. Furthermore, the safety constraints may vary among different tasks in general. This implies that the safe policies in one task are not necessarily safe in another task, especially for those tasks with large variation. Safe MRL imposes safety constraints on the execution of the control policy. Offline safe MRL is studied in [171], which aims to achieve all-time safety, when the safety constraints are invariant over the tasks, by offline learning a neural network to suppress all the unsafe actions. Online safe MRL is considered in [172], which shows that safety can be guaranteed asymptotically with respect to the number of tasks and the number of adaptation steps. However, all-time safety during the online learning and deployment of control policies in changing environments remains an open question.

Gradient-based methods are the most prevalent methods to update meta parameters in online meta learning [168, 169, 170, 167, 173]. The meta parameters are updated along the (approximate) gradients of the objective functions. In reinforcement learning, the gradients of the objective functions with respect to the meta (control policy) parameters are unknown in most cases. Therefore, each new gradient step requires new samples to estimate the gradient with respect to the meta parameters [141, 166]. This makes the update of the meta control policy exceedingly inefficient as the numbers of gradient steps and samples per step increase in order to update the control policy effectively.

To this end, we develop *masked Follow-the-Last-Parameter-Policy (FTLPP)*,

an online safe meta reinforcement learning algorithm, which explicitly considers all-time safety at the deployment of the control policy and sample efficiency in online update of the meta control policy. In this chapter, we consider safety as the agent not entering a set of unsafe states. Our contributions are summarized as follows:

1. **All-time safety through policy masking.** We develop a novel procedure, such that the safety-constrained MRL problem is transformed into an unconstrained optimization problem over the masked control policy space. Specifically, we construct a masking function for an arbitrary control policy such that the masked control policy has small probability, if not zero, in driving the agent into unsafe states in a task. Safety is formally guaranteed for any control policy within the masked control policy space. The policy masking framework allows optimization over the safe actions with respect to the task objectives.

2. **Sample-efficient meta update through learning a parameter policy over task space.** We propose FTLPP, a novel online meta learning framework which does not require estimating the gradients of the objective functions with respect to the decision variable. By modeling the temporal relation for the sequence of tasks using Markov process, the online learning of the meta (control policy) parameter is solved as an online policy optimization problem with tasks as the states and meta (control policy) parameters as the actions. An online off-policy algorithm, which requires the parameter policy to be updated after taking an action in each step, is developed to achieve high sample efficiency and sublinear growth of dynamic regret. To the best of our knowledge, this is the first attempt to solve an online learning problem using an off-policy reinforcement learning method.

We present empirical results that show masked FTLPP achieves all-time safety and efficient meta update. We also compare with three baselines: Meta SRL [172], FTML [168] with the policy masking framework we propose for all-time safety, and SAILR [174] with FTLPP we propose for meta update.

6.2 Related work

Our masked FTLPP framework incorporates two key components: a policy masking framework for all-time safety and a policy optimization formulation for online meta update. In this section, we review prior works that draw inspirations.

Safe (reinforcement) learning. In the literature of safe (reinforcement) learning, all-time safety can be guaranteed by switching between a learning-based control policy and a backup control policy that is suboptimal but is able to guarantee safety [27]. The backup safety controllers can be synthesized through solving a two-player zero-sum differential game [44], model predictive control (MPC) [101][34], control barrier function [102, 103], robust optimization [104], reachability analysis [105] and regions of attraction [46]. However, the safety actions in these works are decoupled from the task objectives. Our work is inspired by the idea of backup control policies. Instead of synthesizing a backup control policy that outputs a single safe action at critical states for a task, we consider removing the unsafe actions in the task and performing optimization within the space of control policies without these unsafe actions. This framework can be treated as considering all possible backup control policies and allows optimization within the space of backup control policies, coupled with the optimization of the learning-based control policy.

Online (meta) learning. As shown in [168], an online meta learning problem can be transformed into a standard online learning problem. A majority of the online learning methods achieving low regret are gradient-based [175, 176, 177, 178], but as previously mentioned, gradient-based methods are not sample-efficient in reinforcement learning. Papers [172, 179] circumvent the issue by considering the (weighed) average of the previous adapted control policies as the meta control policies of a new task. The weights are computed based on the state visitation probability of the control policies, which is estimated through sampling trajectories under the control policies. Sample efficiency is enhanced in these works as the trajectories used in the previous weight computation are reused to compute the new weights in the future rounds. However, as empirically shown in [172], this method may not work well when the task similarity is low. In this work, we also seek for a non-gradient-based approach to enhance sample efficiency. Our proposed FTLPP

framework is inspired by [180], which leverages a known dynamic model of the optimal decision variables for online update. In contrast, we assume the existence, though unknown, of such dynamic model among the transition of tasks in the form of Markov process. Then we transform the online learning problem of the meta (control policy) parameter into a policy optimization problem, where tasks are the states and the meta (control policy) parameters in the next round are the actions, and solve it by developing an online off-policy reinforcement learning algorithm. The assumption of a latent Markov process for task transition is actually mild as not only it can be justified by real-world examples, but also the standard assumption in meta learning, where tasks follow a common latent distribution, is a special case of it. Furthermore, this allows the task distribution to be time-varying, which is more realistic for agents operating in changing environments.

On/Off-policy reinforcement learning. As the above paragraph discusses, the FTLPP framework is closely related to reinforcement learning algorithms. It is inspired by the differences between on-policy methods and off-policy methods in reinforcement learning. A major feature of on-policy methods is updating the policy through policy gradient [181, 182, 183], which is similar to the gradient-based methods in online learning. However, on-policy methods are not sample-efficient due to the need of gradient estimation at each step. In contrast, off-policy methods aim to reuse past experience and incrementally learn the Q-function [184, 185]. Our FTLPP framework can be treated as an off-policy method for online learning, and the major contribution is connecting the online learning problem to a reinforcement learning problem. The online meta control policy learning problem is then transformed into an optimization problem of a hypernetwork [186], where the output of a network (i.e., the parameter policy in FTLPP) is the parameter of another network (i.e., the meta control policy). Furthermore, the FTLPP framework is agnostic to the adaptation procedure and hence can be applied to the online meta learning of any reinforcement learning algorithms when encountered by changing environments.

6.3 Problem statement

Safe reinforcement learning. Consider a space of tasks $\mathcal{T}$, where each task $\tau \in \mathcal{T}$ admits a Markov decision process (MDP) $\mathcal{M}_\tau \triangleq (\mathcal{S}, \mathcal{A}, T^{\mathcal{S}}, r_\tau, \rho_\tau, H_\tau)$, where $\mathcal{S} \in \mathbb{R}^{n_s}$ is the state space, $\mathcal{A}$ is the action space, $T^{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ is the state transition function such that $s_{t+1} \sim T^{\mathcal{S}}(s_{t+1} \mid s_t, a_t)$, $r_\tau : \mathcal{S} \times \mathcal{A} \to [r_{\min}, r_{\max}]$ is the bounded reward function, $\rho_\tau : \mathcal{S} \to \mathbb{R}_{\geqslant 0}$ is the initial state distribution, and H_τ is the time horizon. Notice that the reward function, the initial state distribution and the time horizon are task-dependent while others are not. Denote a stochastic control policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ such that $a_t \sim \pi(a_t \mid s_t)$. Denote a trajectory realization of time horizon H_τ under policy π in task τ starting from state s_0 as $c_\tau^\pi(s_0) \triangleq (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{H_\tau - 1}, a_{H_\tau - 1}, r_{H_\tau - 1})$, where a_t is a realization of $\pi(\cdot \mid s_t)$ and s_{t+1} is a realization of $T^{\mathcal{S}}(\cdot \mid s_t, a_t)$ for $t = 0, \cdots, H_\tau - 1$. For each task τ, we consider safety as the agent not entering a set of unsafe states $\mathcal{S}_\tau^{unsafe} \subset \mathcal{S}$ during the execution of a control policy. The set $\mathcal{S}_\tau^{unsafe}$ is known once task τ is revealed. Since the state transition function and the control policy are stochastic, we formulate the safety constraint for task τ as a chance constraint $P\Big(c_\tau^\pi(s_0) \cap \mathcal{S}_\tau^{unsafe} \neq \emptyset\Big) \leqslant \epsilon$, for some small positive constant ϵ. That is, the rate of unsafe accidents of the agent entering unsafe states needs to be below ϵ. Here, we assume $s_0 \notin \mathcal{S}_\tau^{unsafe}$. For safe reinforcement learning in task τ, we aim to find a control policy π to maximize the expected reward

$$\eta_\tau(\pi) \triangleq \mathbb{E}[\sum_{t=0}^{H_\tau - 1} \gamma^t r_\tau(s_t, a_t)],$$

where the expectation is taken over the state transition $s_{t+1} \sim T^{\mathcal{S}}(s_{t+1} \mid s_t, a_t)$, stochastic policy $a_t \sim \pi(a_t \mid s_t)$ and initial state $s_0 \sim \rho_\tau$, and γ is the discount factor, without violating the safety constraint.

Online safe MRL. In this chapter, we consider the situation where the agent is in a changing environment and aims to accomplish a sequence of tasks $\{\tau_k\}_{k=1}^{\infty}$ online, each arrives in round k. Furthermore, we assume there is a temporal relation between the tasks such that the dynamical model of the tasks follows a Markov process $\tau_{k+1} \sim T^{\mathcal{T}}(\tau_{k+1} \mid \tau_k)$, where $T^{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R} \geqslant 0$ is referred as the task transition function, and $\rho^{\mathcal{T}} : \mathcal{T} \to \mathbb{R}_{\geqslant 0}$ is the initial task distribution. For

notational simplicity, we write shorthand $\pi_k \triangleq \pi_{\tau_k}$ and $\eta_k \triangleq \eta_{\tau_k}$. Denote U as a pre-specified *adaptation* procedure. In online MRL, the agent aims to determine a meta policy approximate $\hat{\pi}_k$ before task τ_k is revealed. The goal for the agent is to online determine the sequence of meta control policy estimates $\{\hat{\pi}_{\tau_k}\}_{k=1}^{\infty}$ which can maximize the expected total cumulative rewards while satisfying the safety constraint for each task:

$$\max_{\{\hat{\pi}_k\}_{k=1}^{K}} \mathbb{E}_{\tau_k \sim T^{\mathcal{T}}}\Big[\sum_{k=1}^{K} \eta_k(U(\hat{\pi}_k;\tau_k))\Big], \tag{6.1a}$$

$$\text{s.t. } P\Big(c_{\tau_k}^{\pi_k}(s_0) \cap \mathcal{S}_{\tau_k}^{unsafe} \neq \emptyset\Big) \leqslant \epsilon, \quad \pi_k = U(\hat{\pi}_k;\tau_k),\ k = 1, \cdots, K. \tag{6.1b}$$

The discussion on the feasibility of the above problem is deferred to Lemma 6.5.5.

Remark 6.3.1. *(Modeling task transition using Markov process).* The assumption of the transition of tasks following a Markov process is mild. In a majority literature of meta learning [141, 144, 187, 142, 188, 189, 166], the tasks are usually assumed to be drawn independently from a common latent distribution, which can be mathematically written as $\tau_k \sim \mathcal{P}(\mathcal{T})$ for all $k \geqslant 1$. This is actually a special case of the Markov process modeling of task transition, as the standard case can be written as $\tau_{k+1} \sim T^{\mathcal{T}}(\tau_{k+1} \mid \tau_k) = \mathcal{P}(\mathcal{T})$ for all $k \geqslant 1$. Real-world applications can also be found below to justify the Markov process modeling of task transitions.

(i) *Piecewise time-varying systems PTVS).* In PTVS, the behaviors of the systems change in a discontinuous way at specific times or intervals. Examples include switched-mode power supply [190], hybrid electric vehicles [191] and robotic manipulator using different strategies (e.g., pulling, pushing or pick-and-place) [192]. The control of a system within a time interval when the dynamics is time-invariant can be treated as a task, and therefore the tasks share a temporal relation due to the evolution of the system/mission. In terms of MDP, the states and actions can be the states and control inputs of the system, respectively. The reward function captures the control objective of the systems (e.g., stabilization and reference tracking). Since these systems are usually physical systems, they are safety critical. Online safe MRL can therefore be leveraged to allow these systems to adapt in time to changing environments during online missions with safety guaranteed. (ii) *Real-time motion planning (RTMP).* In RTMP for long-horizon navigation mis-

sions, especially amid dynamical obstacles, motion plans are synthesized online as a vehicle encounters new scenarios [193, 194]. The query of a solution to a new scenario can be treated as a task, which shares a strong temporal relation with the previous one. The reward function captures the traveling time and/or control effort of the vehicle, and the unsafe sets can be the locations of obstacles. Since planning is done in real time and safety is critical, online safe meta learning is leveraged for fast approximations of the optimal motion plans with safety guaranteed. ■

6.4 The masked Follow-the-Last-Parameter-Policy framework

In this section, we first transform problem (6.1) into an unconstrained problem over a masked control policy space, where all the masked control policies therein satisfy the constraint in (6.1). Next we develop the FTLPP algorithm to output $\hat{\pi}_k$ sequentially for each $k = 1, \cdots, K$.

Consider a class of learning-based control policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ parameterized by $\theta \in \mathbb{R}^{n_\theta}$, such as a deep neural network. Then given a task τ and a policy π_θ, we develop a masking function $m_{\tau,\pi_\theta} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ such that the masked control policy $\check{\pi}_{\theta,\tau} \triangleq \pi_\theta \cdot m_{\tau,\pi_\theta}$ has small probability at most ϵ in driving the MDP to unsafe set $\mathcal{S}_\tau^{unsafe}$; i.e., for any task τ, $P\Big(c_\tau^{\check{\pi}_{\theta,\tau}}(s_0) \cap \mathcal{S}_\tau^{unsafe} \neq \emptyset\Big) \leqslant \epsilon$ for all $\theta \in \mathbb{R}^{n_\theta}$. More detailed explanation can be found in Section 6.5. Then for a task τ, the masked control policy can be optimized as

$$\theta_\tau = \arg\max_{\theta \in \mathbb{R}^{n_\theta}} \eta_\tau(\check{\pi}_{\theta,\tau}). \tag{6.2}$$

Notice that for each task τ, the expected reward $\eta_\tau(\check{\pi}_{\theta,\tau})$ only depends on parameter θ, and hence, in the sequel, we use shorthand $\eta_k(\theta) \triangleq \eta_{\tau_k}(\check{\pi}_{\theta,\tau_k})$. Then U can be any reinforcement learning algorithms such as [184][181][182] for policy parameter optimization. Correspondingly, problem (6.1) can be reformulated into an unconstrained optimization problem defined as follows:

$$\max_{\{\hat{\theta}_k\}_{k=1}^K} \mathbb{E}_{\tau_k \sim T^\mathcal{T}} \Big[\sum_{k=1}^K \eta_k(U(\hat{\theta}_k; \tau_k)) \Big]. \tag{6.3}$$

Next we develop the FTLPP algorithm to solve the above problem.

6.4.1 The Follow-the-Last-Parameter-Policy (FTLPP) algorithm

Since the task of each round k arrives following the distribution $\tau_k \sim T^{\mathcal{T}}(\tau_k \mid \tau_{k-1})$, there is a latent dependency of the optimal meta (control policy) parameters for task τ_k given τ_{k-1}, which can be modeled as $\hat{\theta}_k \sim \zeta(\hat{\theta}_k \mid \tau_{k-1})$, where $\zeta : \mathbb{R}^{n_\theta} \times \mathcal{T} \to \mathbb{R}_{\geqslant 0}$ can also be treated as the optimal policy for selecting the next meta (control policy) parameter given the previous task. We refer ζ as the *parameter policy*, whereas π is the *control policy*. Then the learning of the optimal $\hat{\theta}_k$ is transformed into the learning of the optimal policy for an infinite horizon MDP, denoted as $\mathcal{M}^{\Theta}$, with task τ_k as the state and meta (control policy) parameter $\hat{\theta}_{k+1}$ as the action at round k, reward function given by $R(\tau_k, \hat{\theta}_{k+1}) \triangleq \mathbb{E}_{\tau_{k+1} \sim T^{\mathcal{T}}(\tau_{k+1} \mid \tau_k)}[\eta_{k+1}(U(\hat{\theta}_{k+1}; \tau_{k+1}))]$, and state transition function $T^{\mathcal{T}}$. That is, the online learning of meta (control policy) parameter $\hat{\theta}_k$ is then transformed into an optimization problem of a hypernetwork [186], where the output of a network (i.e., parameter policy ζ) is the parameter $\hat{\theta}_k$ of another network (i.e., the meta control policy). Notice that $T^{\mathcal{T}}(\tau_k \mid \tau_{k-1})$ is unknown a priori but is only realized through the tasks $\{\tau_k\}_{k=1}^{\infty}$ arriving sequentially. This implies that the update of the parameter policy ζ can only be done in an online manner; i.e., update at each step after taking an action; i.e., selecting the next meta (control policy) parameter $\hat{\theta}_{k+1}$, at each state; i.e., the current task τ_k, and receiving the corresponding reward $R(\tau_k, \hat{\theta}_{k+1})$. Next we present the novel FTLPP framework for online learning the parameter policy ζ. The formal algorithm is presented in Algorithm 11.

The algorithm is built atop of the SAC algorithm [184] due to its sample efficiency and the theoretical guarantees on monotonic convergence to the optimal policy. Following [184], the soft Q-value function, value function, and the update for the parameter policy are respectively given by

$$
\begin{aligned}
Q_\alpha^\zeta(\tau_k, \hat{\theta}_{k+1}) &\triangleq R(\tau_k, \hat{\theta}_{k+1}) + \gamma \mathbb{E}_{\tau_{k+1} \sim T^{\mathcal{T}}}[V_\alpha^\zeta(\tau_{k+1})], \\
V_\alpha^\zeta(\tau_k) &\triangleq \mathbb{E}_{\hat{\theta}_{\tau_{k+1}} \sim \zeta}[Q^\zeta(\tau_k, \hat{\theta}_{k+1}) - \alpha \log \zeta(\hat{\theta}_{k+1} \mid \tau_k)],
\end{aligned}
$$

$$\zeta_{\alpha_k}^{(l+1)} = \arg\min_{\zeta} D_{\alpha_k}^{KL}\Big(\zeta(\cdot \mid \tau)\Big\|\frac{\exp(Q_{\alpha_k}^{\zeta_{\alpha_k}^{(l)}}(\tau,\cdot))}{Z^{\zeta_{\alpha_k}^{(l)}}(\tau)}\Big). \tag{6.4}$$

where $\gamma \in (0,1)$ is the discount factor, $\alpha_k \in \mathbb{R}_{>0}$ determines the relative weight on the entropy term against the reward and $D_{\alpha_k}^{KL}(p\|q) \triangleq \mathbb{E}_{x\sim p(x)}[\alpha_k \log p(x) - \log q(x)]$ is the KL divergence.

Regret analysis. Consider a positive sequence $\{\alpha_k\}_{k=1}^K$. Denote $\{\tilde{\theta}_k\}_{k=1}^K$ as the optimal solution to Problem 6.5. The following theorem shows that the optimality gap between $\hat{\theta}_k$ and $\tilde{\theta}_k$ can be controlled by α_k.

Theorem 6.4.1. Suppose $\hat{\theta}_k, \tilde{\theta}_k \in \Theta \subset \mathbb{R}^{n_\theta}$. Update rule (6.4) renders that there exists ζ_k such that $\lim_{l\to\infty} \zeta_{\alpha_k}^{(l)} = \zeta_{\alpha_k}$ for any $k = 1, \cdots, K$. Furthermore, it holds that $\mathbb{E}_{\tau_{k+1}\sim T^{\mathcal{T}}, \hat{\theta}_{k+1}\sim\zeta_{\alpha_k}}[\eta_{k+1}(U(\tilde{\theta}_{k+1};\tau_{k+1})) - \eta_{k+1}(U(\hat{\theta}_{k+1};\tau_{k+1}))] \leqslant \frac{\alpha_k}{1-\gamma}\log|\Theta|$. ■

The proof of the theorem can be found in Section 6.6.1. Theorem 6.4.1 implies that if α_k diminishes, then we have $\lim_{k\to\infty} \mathbb{E}_{\tau_{k+1}\sim T^{\mathcal{T}}, \hat{\theta}_{k+1}\sim\zeta_k}[\eta_{k+1}(U(\tilde{\theta}_{k+1};\tau_{k+1})) - \eta_{k+1}(U(\hat{\theta}_{k+1};\tau_{k+1}))] = 0$. Define *dynamic regret* [175] as

$$\text{Regret}_K \triangleq \sum_{k=1}^{K} \eta_k(U(\tilde{\theta}_k;\tau_k)) - \sum_{k=1}^{K} \eta_k(U(\hat{\theta}_k;\tau_k)). \tag{6.5}$$

The expectation of the dynamic regret Regret_K has sublinear growth if α_k diminishes. For example, if $\alpha_k = 1/k$, then we have

$$\begin{aligned}
\mathbb{E}[\text{Regret}_K] &= \sum_{k=1}^{K} \mathbb{E}_{\tau_k\sim T^{\mathcal{T}}, \hat{\theta}_k\sim\zeta_k}[\eta_k(U(\tilde{\theta}_k;\tau_k)) - \eta_k(U(\hat{\theta}_k;\tau_k))] \\
&\leqslant \sum_{k=1}^{K-1} \frac{\alpha_k}{1-\gamma}\log|\Theta| + \mathbb{E}_{\tau_1\sim\rho^{\mathcal{T}}}[\eta_k(U(\tilde{\theta}_1;\tau_1)) - \eta_1(U(\hat{\theta}_1;\tau_1))] \\
&\leqslant \frac{\log K - 1}{1-\gamma}\log|\Theta| + \mathbb{E}_{\tau_1\sim\rho^{\mathcal{T}}}[\eta_k(U(\tilde{\theta}_1;\tau_1)) - \eta_1(U(\hat{\theta}_1;\tau_1))],
\end{aligned}$$

where the second term is caused by the initialization of the meta (control policy) parameter $\hat{\theta}_1$. This also reflects a guideline for exploration-exploitation. Paper [184] derives (6.4) based on the maximum entropy principle such that exploration is encouraged and weighed by α_k. A diminishing sequence of $\{\alpha_k\}_{k=1}^K$ indicates that there are less incentives for exploration in the later period of learning. This

is consistent with the fact that as learning proceeds, the learning agent is more knowledgeable and becomes more capable of approximating an optimal parameter policy using the given information. Therefore, more effort should be given to exploitation. Notice that we cannot set $\alpha_k = 0$ for any k since it would no longer fit into the framework in [184], and convergence to the optimal policy is no longer guaranteed, which is a crucial property leveraged in the proof of Theorem 6.4.1.

Algorithm 11 Online meta reinforcement learning

Init: Initial meta parameter: $\hat{\theta}_1$; Parameters for function approximation: ψ, ϕ, ξ; Step sizes: β^V, β^Q, β^ζ; Number of initial state samples: n_{s_0}; Factor for moving average: $\nu \in (0, 1)$; Entropy weight factor: $\{\alpha_k\}_{k=1}^K$
for $k = 1, \cdots, K$ **do**
  Obtain task τ_k and adapt parameter $\theta_{\tau_k} \leftarrow U(\hat{\theta}_k; \tau_k)$ Sample initial state $s_0^j \sim \rho_{\tau_k}$ for $j = 1, \cdots, n_{s_0}$
  for $j = 1, \cdots, n_{s_0}$ **do**
    $r_j \leftarrow 0$
    for $t = 1, \cdots, H_{\tau_k}$ **do**
      Sample $a_t \sim \breve{\pi}_{\theta_{\tau_k}, \tau_k}$
      Sample $s_{t+1} \sim T^{\mathcal{S}}(s_{t+1} \mid s_t, a_t)$
      $r_j \leftarrow r_j + \gamma^t r_{\tau_k}(s_t, a_t)$
    end for
  end for
  $\hat{R}(\tau_k, \hat{\theta}_{k+1}) \leftarrow \frac{1}{n_{s_0}} r_j$
  $\psi \leftarrow \psi - \beta^V \sum_{k'=1}^k \hat{\nabla}_\psi J_{k'}^V(\psi)$
  $\phi \leftarrow \phi - \beta^Q \sum_{k'=1}^k \hat{\nabla}_\phi J_{k'}^Q(\phi)$
  $\xi \leftarrow \xi - \beta^\zeta \sum_{k'=1}^k \hat{\nabla}_\xi J_{k'}^\zeta(\xi)$
  $\bar{\psi} \leftarrow \nu\psi + (1 - \nu)\bar{\psi}$
  Sample $\hat{\theta}_{k+1} \sim \zeta_\xi(\cdot \mid \tau_k)$
end for

Practical implementation. Algorithm 11 presents the practical implementation of (6.4). As in [184], we consider the parametric representations of the above three functions, denoted by Q_ϕ, V_ψ and ζ_ξ. Since the state transition of the MDP $\mathcal{M}^\Theta$ is inherently driven by the transition of tasks, which arrives online sequentially by round, the loss functions for training parameters ϕ, ψ and ξ are rewritten as the sum of loss over each round k, instead of as the expectation over a fixed dataset

in [184]. Then the loss functions have the form

$$
\begin{aligned}
&J^V(\psi) \triangleq \sum_{k=1}^{K} J_k^V(\psi),\ J^Q(\phi) \triangleq \sum_{k=1}^{K} J_k^Q(\phi),\ J^\zeta(\xi) \triangleq \sum_{k=1}^{K} J_k^\zeta(\xi),\\
&J_k^V(\psi) \triangleq \frac{1}{2}\Big(V_\psi(\tau_{k-1}) - \mathbb{E}_{\hat{\theta}_k \sim \zeta_\xi}[Q_\theta(\tau_{k-1}, \hat{\theta}_k) - \alpha_k \log \zeta_\xi(\hat{\theta}_k \mid \tau_{k-1})]\Big)^2,\\
&J_k^Q(\phi) \triangleq \frac{1}{2}\Big(Q_\theta(\tau_{k-1}, \hat{\theta}_k) - \hat{Q}(\tau_{k-1}, \hat{\theta}_k)\Big)^2,\\
&\hat{Q}(\tau_{k-1}, \hat{\theta}_k) \triangleq \hat{R}(\tau_{k-1}, \hat{\theta}_k) + \gamma V_{\bar{\psi}}(\tau_k),\\
&J_k^\zeta(\xi) \triangleq \alpha_k \log \zeta_\xi(\hat{\theta}_k \mid \tau_{k-1}) - Q_\theta(\tau_{k-1}, \hat{\theta}_k),
\end{aligned}
$$

where $\bar{\psi}$ is the moving average of ψ, and $\hat{R}(\tau_{k-1}, \hat{\theta}_k)$ is the empirical estimate of $\eta_k(U(\hat{\theta}_k; \tau_k))$ and the one-sample empirical estimate of $R(\tau_{k-1}, \hat{\theta}_k)$, noting that only a single task τ_k is revealed at each round k. All the above loss functions resemble the standard form for online optimization [175], and therefore the parameters can be updated each round using online gradient-based methods (e.g., Follow-the-Leader [195]). As in [184], the gradients can be estimated using stochastic gradient, denoted by $\hat{\nabla}$. Note that no gradient of the objective function η_k with respect to $\hat{\theta}_k$ is needed, and loss functions J^V, J^Q and J^ζ serve to solve regression problems provided a fixed dataset, which is sample-efficient due to backward propagation. Hence, the stochastic gradients with respect to ψ, ϕ and ξ can be computed using a fixed set of samples $\{\tau_{k'-1}, \hat{\theta}_{k'}, \hat{R}(\tau_{k'-1}, \hat{\theta}_{k'})\}_{k'=1}^{k}$ for an arbitrary number of steps at each round k. Given a new estimate of ξ at the end of iteration k, the parameter of the meta control policy for next iteration is predicted using the new parameter policy ζ_ξ.

6.5 Policy masking

In this section, we provide a detailed procedure and justification for constructing the masking function m_{τ,π_θ} mentioned in the beginning of Section 6.4. Denote $\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \triangleq \mathcal{S}_\tau^{unsafe} \cup \{s \in \mathcal{S} \mid P\Big(c_\tau^\pi(s) \cap \mathcal{S}_\tau^{unsafe} \neq \emptyset\Big) > \epsilon \text{ for any } \pi\}$ the set of ϵ-inevitably unsafe states in task τ and $\mathcal{A}_{\epsilon,\tau}^{unsafe}(s) \triangleq \{a \in \mathcal{A} \mid \sum_{s' \in \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}} T^{\mathcal{S}}(s' \mid s, a) > \frac{\epsilon}{2H_\tau}\}$ the set of ϵ-unsafe actions that drive the MDP $\mathcal{M}_\tau$ to $\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ from state s in task τ. We define a $(1-\epsilon)$-safe control policy space below.

Algorithm 12 Construction of m_{τ,π_θ}

Init: $\mathrm{BR}_\epsilon(s,a) \leftarrow \emptyset$, $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \leftarrow \mathcal{S}^{unsafe}_{\tau}$, $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s) \leftarrow \emptyset$, $Flag \leftarrow 1$
(Construction of BR_ϵ)
for $s \in \mathcal{S}$ **do**
  for $a \in \mathcal{A}$ **do**
    for $s' \in \mathcal{S}$ **do**
      if $T^{\mathcal{S}}(s' \mid s,a) \geqslant \frac{\epsilon}{2H_\tau|\mathcal{S}|}$ **then**
        $\mathrm{BR}_\epsilon(s',a) \leftarrow \mathrm{BR}_\epsilon(s',a) \cup \{s\}$
      end if
    end for
  end for
end for
(Construction of $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$)
while $Flag == 1$ **do** $Flag \leftarrow 0$
  for $s' \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ **do**
    for $s \in \mathrm{BR}_\epsilon(s',a)$, $a \in \mathcal{A}$ **do**
      $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s) = \mathcal{A}^{unsafe}_{\epsilon,\tau}(s) \cup \{a\}$
      if $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s) == \mathcal{A}$ and $s \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ **then**
        $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \leftarrow \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \cup \{s\}$
        $Flag \leftarrow 1$
      end if
    end for
  end for
end while
(Construction of masking function m_{τ,π_θ})
for $s \in \mathcal{S}$, $a \in \mathcal{A}$ **do**
  if $s \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s) \neq \emptyset$ **then**
    if $a \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$ **then**
      $m_{\tau,\pi_\theta}(s,a;\pi_\theta) \leftarrow \frac{\epsilon/2H_\tau}{\sum_{a' \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)} \pi_\theta(a'|s;\tau)}$
    else
      $m_{\tau,\pi_\theta}(s,a;\pi_\theta) \leftarrow \frac{1-\epsilon/2H_\tau}{1-\sum_{a' \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)} \pi_\theta(a'|s;\tau)}$
    end if
  else
    $m_{\tau,\pi_\theta}(s,a;\pi_\theta) \leftarrow 1$
  end if
end for
return m_{τ,π_θ}, $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$

Definition 6.5.1. *($(1-\epsilon)$-safe control policy space).* A control policy space Π is $(1-\epsilon)$-safe for task τ if $P\Big(c_\tau^\pi(s_0) \cap \mathcal{S}_\tau^{unsafe} \neq \emptyset\Big) \leqslant \epsilon$ for any $s_0 \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ for all control policy $\pi \in \Pi$.

Proposition 6.5.2 below characterizes a property for a control policy space being $(1-\epsilon)$-safe for task τ.

Proposition 6.5.2. Consider a control policy space Π. If $\sum_{a \in \mathcal{A}_{\epsilon,\tau}^{unsafe}(s)} \pi(a \mid s) \leqslant \frac{\epsilon}{2H_\tau}$ for each $s \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ for all $\pi \in \Pi$, then Π is a $(1-\epsilon)$-safe control policy space for task τ. ■

The proof of the proposition can be found in Section 6.6.2.

In this chapter, for simplicity of exposition, we assume the state transition function $T^{\mathcal{S}}$ is known and fixed across different tasks, which is usually the case in the motivating examples in Section 3. If the model is unknown, model learning can be performed using the collection of tasks during meta training as in [187, 196, 188, 44]. Inspired by Proposition 6.5.2, we can construct a masked control policy space for each task τ by multiplying each $\pi_\theta \in \Pi$ by a masking function m_{τ,π_θ} such that for each $s \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$, the following two criteria are satisfied:

(M1) $\sum_{a \in \mathcal{A}_\tau^{unsafe}(s)} \pi_\theta(a \mid s) \cdot m_{\tau,\pi_\theta}(a,s) \leqslant \frac{\epsilon}{2H_\tau}$.

(M2) $\sum_{a \in \mathcal{A}} \pi_\theta(a \mid s) \cdot m_{\tau,\pi_\theta}(a,s) = 1$.

The following theorem shows that the masking function m_{τ,π_θ} constructed following these criteria induces a $(1-\epsilon)$-safe control policy space for task τ.

Theorem 6.5.3. If masking function m_{τ,π_θ} is constructed satisfying (M1) and (M2), then the control policy space $\Pi_\tau \triangleq \{\pi_\theta \cdot m_{\tau,\pi_\theta} \mid \theta \in \mathbb{R}^{n_\theta}\}$ is $(1-\epsilon)$-safe for task τ. ■

The proof of the theorem can be found in Section 6.6.3. The above result inspires the algorithm for constructing m_{τ,π_θ} in three steps. First, since the state space is finite, we can construct a directed graph to store the one-step backward sets of model $T^{\mathcal{S}}$. Then given the directed graph, the states that inevitably drive into $\mathcal{S}_\tau^{unsafe}$ with probability at least ϵ can be identified as $\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ together with $\mathcal{A}_{\epsilon,\tau}^{unsafe}(s)$, $s \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$. Finally, the masking function m_{τ,π_θ} can be constructed following (M1) and (M2). Next we explain each step in details.

Construction of one-step backward sets. Define the one-step backward set BR_ϵ of rate of accident ϵ for state-action pair (s, a) as

$$\mathrm{BR}_\epsilon(s,a) \triangleq \{s' \in \mathcal{S} \mid T^{\mathcal{S}}(s' \mid s,a) > \frac{\epsilon}{2H_\tau|\mathcal{S}|}\}. \tag{6.6}$$

The formal construction procedure of BR_ϵ can be found in Algorithm 12 Lines 3 to 7.

Construction of ϵ-inevitably unsafe states and actions. Utilizing the information stored in the one-step backward sets BR_ϵ, upon reveal of task τ, the corresponding ϵ-inevitably unsafe states $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and actions $\mathcal{A}^{unsafe}_{\epsilon,\tau}(\cdot)$ can be constructed following three criteria:

(C1) All the states in $\mathcal{S}^{unsafe}_\tau$ are included into $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$; i.e., $\mathcal{S}^{unsafe}_\tau \subset \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$.

(C2) For each state s, all the actions driving the MDP $\mathcal{M}_\tau$ into $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ are included into $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$; i.e., $a \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$ if $s \in \mathrm{BR}_\epsilon(s', a)$, $s' \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$.

(C3) If state s has all the actions included in $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$, it is included into $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$; i.e., $s \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ if $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s) = \mathcal{A}$.

The formal algorithm statement for constructing $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and $\mathcal{A}^{unsafe}_{\epsilon,\tau}(\cdot)$ is presented in Algorithm 12 Lines 13 to 19.

Remark 6.5.4. *(Computation complexity of $\bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and $\mathcal{A}^{unsafe}_{\epsilon,\tau}(\cdot)$)* Notice that (C2) and (C3) can trigger recursive procedures because if (C3) is satisfied for some state $s \in \mathcal{S} \setminus \mathcal{S}^{unsafe}_\tau$, we have $s \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ and $a \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s')$ for all $s' \in \mathrm{BR}_\epsilon(s,a)$, according to (C2). The update of $\mathcal{A}^{unsafe}_{\epsilon,\tau}(s')$ may in turn trigger (C3). Yet the procedure would eventually terminate in time at most $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$ as there are only a finite number of states and actions.

The following lemma provides a sufficient condition on the feasibility of Problem (6.1) .

Lemma 6.5.5. If $\mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \neq \emptyset$ and $s_0 \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ for all task τ, then Problem (6.1) is feasible. ■

The proof of the lemma can be found in Section 6.6.4. Lemma 6.5.5 implies that the feasibility of Problem (6.1) can be verified by running Line 3 to Line 22 in Algorithm 12 for each task.

Construction of masking function. Given the ϵ-inevitably unsafe states $\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ and ϵ-unsafe actions $\mathcal{A}_{\epsilon,\tau}^{unsafe}$, for each learning-based policy π_θ, the value of masking function m_{τ,π_θ} can be obtained through Algorithm 12 Lines 25 to 33 for each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ such that (M1) and (M2) are satisfied.

Since the model is known a priori, the construction of BR_ϵ (Lines 3 to 7) can be done one time and offline. Since the unsafe states $\mathcal{S}_\tau^{unsafe}$ are available once task τ is revealed, the construction of ϵ-inevitably unsafe states and actions (Lines 13 to 19) only needs to be run one time once task τ is revealed. In practice, instead of storing the table of values for each state-action pair, the value of m_{τ,π_θ} can be queried online together with that of π_θ at each state s using Lines 26 to 33. Therefore, the masking function m_{τ,π_θ} can be available quickly once task τ is revealed, especially when the unsafe sets are sparse.

6.6 Proofs

6.6.1 Proof of Theorem 6.4.1

Recall that in the MDP $\mathcal{M}^\Theta$, task τ_k is the state and $\hat{\theta}_{k+1}$ is the action at round k. Following the proof of Theorem 1 in [184], we have that there exists ζ_{α_k} such that $\lim_{l\to\infty} \zeta_{\alpha_k}^{(l)} = \zeta_{\alpha_k}$ and

$$\zeta_{\alpha_k} = \arg\max_{\zeta} Q^\zeta(\tau_k, \hat{\theta}_{k+1}). \tag{6.7}$$

Recursively applying the definition of Q^ζ renders

$$Q^\zeta(\tau_k, \hat{\theta}_{k+1}) = R(\tau_k, \hat{\theta}_{k+1}) + \gamma \mathbb{E}_{\tau_{k+1}\sim T^\mathcal{T}}[V_{\alpha_k}^\zeta(\tau_{k+1})] \tag{6.8}$$

for all $(\tau_k, \hat{\theta}_{k+1}) \in \mathcal{T} \times \Theta$. Combining (6.7) with (6.8) also implies

$$\begin{aligned}\zeta_k &= \arg\max_{\zeta}[V_{\alpha_k}^\zeta(\tau_k)] \\ &= \sum_{l=1}^{\infty} \gamma^l \mathbb{E}_{\tau_{k+l}\sim T^\mathcal{T}, \hat{\theta}_{k+1+l}\sim\zeta_k}[R(\tau_{k+l}, \hat{\theta}_{k+l+1}) - \alpha_k \log \zeta(\hat{\theta}_{k+l+1} \mid \tau_{k+l})]\end{aligned}$$

Denote

$$V^{\zeta}(\tau_k) \triangleq \mathbb{E}_{\tau_{k+l}\sim T^{\mathcal{T}},\hat{\theta}_{k+l+1}\sim\zeta}[\sum_{l=0}^{\infty}\gamma^l R(\tau_{k+l},\hat{\theta}_{k+l+1}) \mid \tau_0=\tau]$$
$$= \mathbb{E}_{\tau_{k+l}\sim T^{\mathcal{T}},\hat{\theta}_{l+1}\sim\zeta}[\sum_{l=1}^{\infty}\gamma^{l-1}\eta_{k+l}(U(\hat{\theta}_{k+l};\tau_{k+l}))],$$
$$\zeta_V \triangleq \arg\max_{\zeta} V^{\zeta}(\tau) = \mathbb{E}_{\tau_{k+l}\sim T^{\mathcal{T}}}[\sum_{l=1}^{\infty}\gamma^{l-1}\eta_l(U(\tilde{\theta}_{k+l};\tau_{k+l}))],$$

recalling the definition of $\tilde{\theta}_l$ above Theorem 6.4.1. Inequality (12) in [197] shows that $V^{\zeta_k}(\tau_k) \leqslant V^{\zeta_V}(\tau_k) \leqslant V^{\zeta_k}(\tau_k) + \frac{\alpha_k}{1-\gamma}\log|\Theta|$, or equivalently,

$$\begin{aligned}
&V^{\zeta_V}(\tau_k) - V^{\zeta_k}(\tau_k)\\
&= \mathbb{E}_{\tau_{k+l}\sim T^{\mathcal{T}},\hat{\theta}_{k+l}\sim\zeta_k}\Big[\sum_{l=1}^{\infty}\gamma^{l-1}\Big(\eta_l(U(\tilde{\theta}_{k+l};\tau_{k+l})) - \eta_l(U(\hat{\theta}_{k+l};\tau_{k+l}))\Big)\Big]\\
&\leqslant \frac{\alpha_k}{1-\gamma}\log|\Theta|.
\end{aligned}$$

Notice that

$$\begin{aligned}
&\sum_{l=1}^{\infty}\gamma^{l-1}\Big(\eta_{k+l}(U(\tilde{\theta}_{k+l};\tau_{k+l})) - \eta_{k+l}(U(\hat{\theta}_{k+l};\tau_{k+l}))\Big)\\
&\geqslant \eta_{k+1}(U(\tilde{\theta}_{k+1};\tau_{k+1})) - \eta_{k+1}(U(\hat{\theta}_{k+1};\tau_{k+1})).
\end{aligned}$$

This implies $\mathbb{E}_{\tau_{k+1}\sim T^{\mathcal{T}},\hat{\theta}_{k+1}\sim\zeta_k}\Big[\eta_{k+1}(U(\tilde{\theta}_{k+1};\tau_{k+1}))-\eta_{k+1}(U(\hat{\theta}_{k+1};\tau_{k+1}))\Big] \leqslant \frac{\alpha_k}{1-\gamma}\log|\Theta|$.
■

6.6.2 Proof of Proposition 6.5.2

Let $\pi \in \Pi$ and $s_0 \in \mathcal{S}\backslash\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$. Let sequence of states $\{s_t\}_{t=0}^{H_\tau-1}$ be in the trajectory $c_\tau^\pi(s_0)$. Recall that the construction of $\bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$ renders $\mathcal{S}_\tau^{unsafe} \subset \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$. Then probability

$$\begin{aligned}
&P\Big(c_\tau^\pi(s_0) \cap \mathcal{S}_\tau^{unsafe} = \emptyset\Big)\\
&= P\Big(s_0 \notin \mathcal{S}_\tau^{unsafe}, \cdots, s_{H_\tau-1} \notin \mathcal{S}_\tau^{us}\Big)
\end{aligned}$$

$$
\begin{aligned}
&\geqslant P\Big(s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, \cdots, s_{H_\tau-1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}\Big) \\
&= P\Big(s_{H_\tau-1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{us}, a_0 \sim \pi(\cdot \mid s_0), \cdots, \\
&\quad s_{H_\tau-2} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_{H_\tau-2} \sim \pi(\cdot \mid s_{H_\tau-2})\Big) \cdots \\
&\quad \cdot P\Big(s_1 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_0 \sim \pi(\cdot \mid s_0)\Big) P\Big(s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}\Big) \\
&= P\Big(s_{H_\tau-1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_{H_\tau-2} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_{H_\tau-2} \sim \pi(\cdot \mid s_{H_\tau-2})\Big) \\
&\quad \cdot P\Big(s_{H_\tau-2} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_{H_\tau-3} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_{H_\tau-3} \sim \pi(\cdot \mid s_{H_\tau-3}\Big) \cdots \\
&\qquad \cdot P\Big(s_1 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_0 \sim \pi(\cdot \mid s_0)\Big) P\Big(s_0 \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}\Big) \qquad (6.9)
\end{aligned}
$$

where the last equality follows from the definition of MDP. Since policy π satisfies $\sum_{a \in \mathcal{A}_{\epsilon,\tau}^{unsafe}(s)} \pi(a \mid s) \leqslant \frac{\epsilon}{2H_\tau}$ and the definition renders $\mathcal{A}_{\epsilon,\tau}^{unsafe}(s) = \{a \in \mathcal{A} \mid \sum_{s' \in \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}} T^{\mathcal{S}}(s' \mid s, a) > \frac{\epsilon}{2H_\tau}\}$ for each $s \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$, then for any $t = 0, \cdots, H_\tau - 1$, we have

$$
\begin{aligned}
&P\Big(s_{t+1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_t \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_t \sim \pi(\cdot \mid s_t)\Big) \\
&= P\Big(s_{t+1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_t \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_t \in \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t)\Big) P\Big(a_t \in \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t) \mid a_t \sim \pi(\cdot \mid s_t)\Big) \\
&\quad + P\Big(s_{t+1} \notin \mathcal{S}_\tau^{unsafe} \mid s_t \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_t \notin \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t)\Big) P\Big(a_t \notin \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t) \mid a_t \sim \pi(\cdot \mid s_t)\Big) \\
&\geqslant P\Big(s_{t+1} \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe} \mid s_t \notin \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}, a_t \notin \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t)\Big) P\Big(a_t \notin \mathcal{A}_{\epsilon,\tau}^{unsafe}(s_t) \mid a_t \sim \pi(\cdot \mid s_t)\Big) \\
&> (1 - \frac{\epsilon}{2H_\tau})(1 - \frac{\epsilon}{2H_\tau})
\end{aligned}
$$

where the last equality utilizes the definition of $\mathcal{A}_{\epsilon,\tau}^{unsafe}(\cdot)$. Combining this with (6.9) and $s_0 \in \mathcal{S} \setminus \bar{\mathcal{S}}_{\epsilon,\tau}^{unsafe}$, we have

$$
\begin{aligned}
&P\Big(c_\tau^\pi(s_0) \cap \mathcal{S}_\tau^{unsafe} \neq \emptyset\Big) = 1 - P\Big(c_\tau^\pi(s_0) \cap \mathcal{S}_\tau^{unsafe} = \emptyset\Big) \\
&\leqslant 1 - (1 - \frac{\epsilon}{2H_\tau})^{2H_\tau} \leqslant 1 - (1 - \epsilon) = \epsilon.
\end{aligned}
$$

By Definition 6.5.1, the proof is completed. ■

6.6.3 Proof of Theorem 6.5.3

Criterion (M2) ensures the masked policy $\check{\pi}_{\theta,\tau} = \pi_\theta \cdot m_{\tau,\pi_\theta}$ is a valid probability distribution and hence a valid policy. Hence, space $\Pi_\tau \triangleq \{\check{\pi}_{\theta,\tau} \mid \theta \in \mathbb{R}^{n_\theta}\}$ is a valid policy space. Criterion (M1) ensures that $\sum_{a \in \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)} \check{\pi}(a \mid s) \leqslant \frac{\epsilon}{2H_\tau}$ for each $s \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$ for all $\check{\pi} \in \Pi_\tau$. By Proposition 6.5.2, Π_τ is a $(1-\epsilon)$-safe policy space for task τ. ■

6.6.4 Proof of Lemma 6.5.5

Since $\mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \neq \emptyset$, then by (C3), for all $s \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$, there exists $a \in \mathcal{A}$ such that $a \notin \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$. Then (C2) also implies that $s \notin \mathrm{BR}_\epsilon(s', a)$ for all $s' \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$. The definition of BR_ϵ in (6.6) implies that $T^{\mathcal{S}}(s' \mid s, a) \leqslant \frac{\epsilon}{2H_\tau|\mathcal{S}|}$ for all $s' \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$. This implies that

$$
\begin{aligned}
&P\Big(s_{t+1} \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \mid s_t = s, a_t = a\Big) \\
=&1 - P\Big(s_{t+1} \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \mid s_t = s, a_t = a\Big) \\
=&1 - \sum_{s' \in \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}} T^S\Big(s' \mid s, a\Big) \geqslant 1 - \sum_{s' \in \mathcal{S}} T^S\Big(s' \mid s, a\Big) \geqslant 1 - \frac{\epsilon}{2H_\tau}.
\end{aligned} \tag{6.10}
$$

Consider policy π_τ, where for $s \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$, $\pi_\tau(a \mid s) = 1$ for some $a \notin \mathcal{A}^{unsafe}_{\epsilon,\tau}(s)$ and $\pi_\tau(a \mid s) = 0$ otherwise. Then (6.10) implies that

$$
\begin{aligned}
&P\Big(c^{\pi_\tau}_\tau(s_0) \cap \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} = \emptyset \mid s_0 \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}\Big) \\
=& \prod_{t=0}^{H_\tau - 1} P\Big(s_{t+1} \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \mid s_t \in \mathcal{S} \setminus \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}, a_t \sim \pi_\tau(\cdot \mid s_t)\Big) \\
\geqslant& (1 - \frac{\epsilon}{2H_\tau})^{H_\tau} \geqslant 1 - \frac{\epsilon}{2} \geqslant 1 - \epsilon.
\end{aligned}
$$

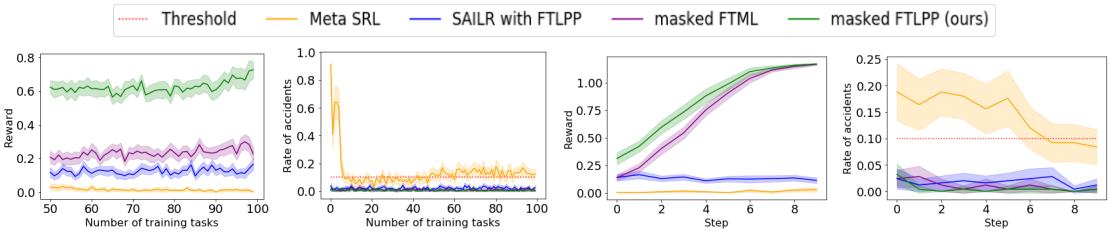
This implies that $P\Big(c^{\pi_\tau}_\tau(s_0) \cap \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau} \neq \emptyset \mid s_0 \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}\Big) \leqslant \epsilon$. By (C1), we have $\mathcal{S}^{unsafe}_\tau \subset \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}$. Therefore, we have $P\Big(c^{\pi_\tau}_\tau(s_0) \cap \mathcal{S}^{unsafe}_\tau \neq \emptyset \mid s_0 \notin \bar{\mathcal{S}}^{unsafe}_{\epsilon,\tau}\Big) \leqslant \epsilon$ Hence, π_τ is a feasible policy for each task τ and Problem (6.1) is feasible. ■

6.7 Experimental evaluation

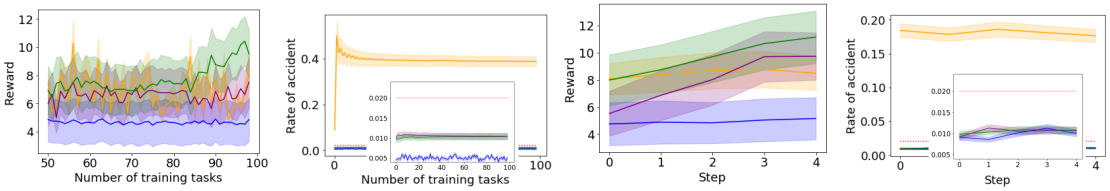
In this section, we conduct simulation experiments to evaluate the proposed masked FTLPP framework. The experiments serve to study two main questions: (i) Is the policy masking framework able to guarantee constraint satisfaction during the whole deployment process? (ii) Is the FTLPP framework effective in online meta update?

Our experiments consider two different scenarios from OpenAI gym [198]: Frozen Lake with maps generated from a Markov process and Acrobot with dynamics generated from a distribution of large variance. In Frozen Lake, the agent gets reward +2 when it reaches the goal; otherwise, it gets reward 0. In Acrobot, the agent gets reward equal to its height plus 2 at each step, but gets reward 0 if it violates the constraint. We also compare our masked FTLPP framework with the following three baselines: (i) Meta SRL [172], where the meta policy for the next task is given by the weighed average of the adapted policies in the previous task obtained through CRPO [199]. (ii) Masked FTML, where FTML proposed in [168] for meta update is combined with the policy masking framework we propose for all-time safety. (iii) SAILR with FTLPP, where SAILR proposed in [174] for all-time safety is combined with the FTLPP framework we propose for meta update.

Figure 6.1 presents the experiment results. Specifically, it shows that the policy masking framework, in contrast to Meta SRL, is able to satisfy the safety constraints all the time. Note that the curves of reward for SAILR with FTLPP, in constrast to masked FTLPP and masked FTML, are relatively flat. This is because SAILR adopts a fixed backup policy, and the performance of the algorithm is mainly determined by the backup policy when the agent is around the boundary of the unsafe sets most of the time. This shows that the masking framework provides a much larger space for optimization, especially at the boundary of the unsafe sets. Comparing with masked FTLPP with masked FTML, it shows that the FTLPP framework has higher sample efficiency by achieving superior performance without using additional samples for gradient estimation in each step as in FTML. Notice that in the experiment of Acrobot, Meta SRL achieves comparable reward with masked FTLPP and masked FTML, but with high rate of unsafe accidents. This



(a) Results of Frozen Lake



(b) Results of Acrobot

Figure 6.1: Experiment results. **Left:** The reward of 50 testing tasks for the policies adapted 1 step from the meta parameter obtained after training with each number of tasks. **Middle left:** The rate of unsafe accidents of 50 testing tasks for the policies adapted 1 step from the meta parameter obtained after training with each number of tasks. **Middle right:** The reward of 50 testing tasks for each step of adapted policy from the meta parameter obtained using 100 training tasks. **Right:** The rate of unsafe accidents of 50 testing tasks for each step of adapted policy from the meta parameter obtained using 100 training tasks.

is due to the fact that, in this experiment, the unsafe accidents conflict with the reward, and the accidents usually bring higher reward. However, in this chapter, ensuring all-time safety is the top priority.

6.8 Conclusion

We present masked Follow-the-Last-Parameter-Policy (FTLPP), an online safe MRL framework that ensures all-time safety and is sample-efficient in meta update. Masked FTLPP is composed of a policy masking framework and an FTLPP framework. All-time safety is achieved through the policy masking framework that suppresses the probability of executing unsafe actions to a low value. FTLPP formulates meta update as a policy optimization problem and solves it using an online off-policy reinforcement learning algorithm. Our theoretical results derive the policy masking framework and justify the FTLPP framework. The proposed framework is evaluated using two case studies in OpenAI gym and compared with

three benchmarks.

Chapter 7

Conclusion and future works

7.1 Conclusion

This dissertation studies safe machine learning for intelligent multi-robot systems. We first provide a class of distributed Gaussian process regression algorithms. The algorithms are able to quantify the uncertainties of intermediate learning results and cognizant of limited resources in computation, memory, and communication budget in the robots. It shows that through limited inter-robot communication, the algorithms achieve Pareto improvement for robots' learning performances. We next propose a distributed learning and planning framework for safe navigation. The proposed algorithms can quickly update each robot's safe control policy based on the results from online learning and guarantee robots' safety under certain conditions. Then we propose a class of federated optimization algorithms which leverages on zero-shot generalization guarantees. We further analyze the algorithm for theoretical guarantees on almost-sure convergence, almost consensus, Pareto improvement and global convergence. Finally, we propose a novel online meta update method and a policy masking framework for online safe meta reinforcement learning. The policy masking framework ensures all-time safety, while the online meta update method is sample-efficient and is able to achieve sublinear growth of dynamic regret.

7.2 Future works

In the future, we plan to address the following issues:

7.2.1 Transfer learning for generalizable reinforcement learning with heterogeneous distributions

Chapter 5 considers learning a single control policy using a network of learners whose training environments follow the same distribution. To further enhance the applicability of the multi-learner learning framework, we plan to relax this assumption and assume that the training distributions of the learners may be heterogeneous.

One potential direction can be applying transfer learning to facilitate the collaborative learning across different distributions. Transfer learning is originally developed for scenarios where obtaining training data that matches the data distribution of the test data can be difficult and expensive [200]. It is used to improve a learner from one (target) domain by transferring information from a related (source) domain [200]. Depending on whether the target domain is identical to the source domain, related literature can be classified into homogeneous transfer learning and heterogeneous transfer learning. In our case, since the domains, the space of the environments, are the same but distributions of the environments are different, approaches from homogeneous transfer learning can be leveraged. The major goal is to reduce the performance drop due to the distribution shift.

One possible approach is to adopt the idea of conditional probability based multi-source domain adaptation [201], whose main idea is to use a combination of source domain classifiers to generate labeled data in the domain. Note that [201] studies a classification problem. When it comes to reinforcement learning in our problem, some modifications are needed. For example, we may first assign a latent variable z to represent for the distribution of the training environments of each learner. We will learn the conditional probability ζ of the distribution variable z givens a sequence of sensory data $\{o_\tau\}_{\tau=1}^{t}$, which contains observations in an environment. Then our control policy π takes three inputs: observation o, goal region $\mathcal{X}_{G,E}$ and distribution variable z sampled from ζ. The learners will collaboratively learn ζ and π, and the objective can be minimizing the total costs

over all the learners. In this way, the learners are able to contribute to the learning of control policies of other learners although they have different distributions of training environments. We will develop the corresponding framework to ensure the collaboration benefits the generalizability of all the learners.

7.2.2 Meta Bayesian learning for safe system identification

In Chapter 4 and Chapter 6, to obtain safety guarantees, explicit prior models of the uncertainties or system dynamics are needed. However, this prior modeling may not be verified for some complex systems in the real world. Therefore, to further enhance the applicability and validity of the safety guarantees, one future work can be relaxing the need of the prior modeling and obtain safe system identification.

One potential direction can be utilizing Bayesian learning to obtain dynamic models with uncertainty quantification. Bayesian learning aims to predict with a distribution of all possible values instead of a single value [202]. The predictive distribution is the posterior distribution obtained from the Bayesian inference framework provided a dataset and a pre-specified prior distribution [202]. The nature of making predictions through distributions in Bayesian learning innately quantifies the prediction uncertainties. If the ground truth dynamic model truly follows the prior distribution, Bayesian learning rules that the ground truth dynamic model also follows the posterior distribution. Probabilistic safety can therefore be guaranteed by using the corresponding confidence set for analysis. Nevertheless, in the real world, it is generally hard to verify the prior distribution of the ground truth model.

One possible approach is to leverage the PAC-Bayes theorem. PAC-Bayes theorem provides probability bounds for the expected cost when predictions are made using Bayesian learning [203]. Leveraging this theorem we can define the cost function as the number of times the confidence sets obtained from Bayesian learning capturing the ground truths. Then the parameters in the Bayesian learning models, such as Bayesian neural network and Gaussian process, can be selected such that the corresponding upper bounds obtained from the theorem aligns with the desired accuracy. The resulting predictive model would then have the desired rate of

capturing the ground truth model regardless of whether the ground truth model follows the prior distribution or not.

7.2.3 Adversarial machine learning for environmental distribution generalization

Chapter 5 designs a single control policy that can provide guaranteed performances in unseen environments. However, we restrict the unseen environments to be sampled from the same distribution as those used for training. To further enhance the intelligence of the robot, it is desired that the controller can generalize to unseen environments from different distributions with performance guarantees.

One potential direction can be applying adversarial machine learning to train a controller generalizable to as many environmental distributions as possible. Adversarial machine learning investigates effective machine learning techniques against an adversarial opponent in classification problems [204]. In general, an adversarial opponent can launch attacks on different components of the learning algorithms, e.g., poisoning training data by injecting false examples, altering the model outputs at the inference phase by deliberately crafting the inputs, and compromising the privacy of the users by obtaining information from the learning agent [204][205]. The goal of adversarial machine learning is to produce a learned model that is robust to the adversarial attacks to some extent. In the context of safe machine learning in robots, environment changes can be seen as attacks to the learned model.

One possible approach is to adopt adversarial distributional training [206]. Similar to [206], we plan to inject adversarial disturbance to the distribution of environments. Furthermore, instead of restricting the magnitude of distributional disturbance, we plan to modify the optimization criterion and aim to maximize the magnitude of disturbance to the distribution of environments while restricting the performance of the control policy being above certain thresholds.

7.2.4 Real-time safety verification for high dimensional systems

In Chapter 5, the safety of the controller can only be verified after the robot is run through the environment(s). If the robot driven under the controller can reach the goal region without collision with the obstacle(s), the controller is safe. It is desired that the safety of the controller can be verified before running the robot through the environment(s). Existing approaches on safety verification are mostly offline, that is, the model is deployed after safety verification and stays the same throughout the whole system operation. However, changes of environments motivate online updates of learned model for adaptation. Therefore, we plan to develop a real-time safety verification framework for online learning algorithms.

Safety verification in machine learning, especially deep learning, aims to provide formal guarantees about the behavior properties and specifications, such as reachability, of machine learning models, which is usually used for safety-assurance purpose before deploying the model [207]. It has been demonstrated that even validating simple properties about the behaviors of neural networks is an NP-complete problem [208]. Online safety verification is non-trivial because existing techniques majorly reply on over-approximation of the reachability set of learned models/learning-enabled control systems, which is usually time-consuming [209].

In the problem of robotic motion planning, one possible solution is to reduce computation by only considering part of the reachability set, e.g., only considering the one-step backward reachable set of inevitable collision states (ICSs) [210]. ICSs are the states that lead to collisions under any control inputs. For any state in the one-step backward reachable set of ICSs, there exists one or more control inputs that drive the system to the ICSs. Therefore, safety verification can be sufficiently accomplished only through the one-step backward reachable set of ICSs, where a state-feedback control policy is safe if it selects control inputs not leading to ICSs when the system state is in the set of one-step backward reachable set of ICSs. In the setting of multi-robot system, safety verification is also challenged by the "curse of dimensionality", that is, the state-action space grows exponentially with respect to the number of robots [211].

7.2.5 Distributed online safe meta reinforcement learning

Chapter 6 considers online meta reinforcement learning in single-learner systems. In the future, to improve learning efficiency, we plan to extend the work to distributed multi-learner systems where the robot learners can collaboratively solve the problem subject to distributed data.

One possible direction can be applying the ideas in the literature of distributed online optimization. Distributed online optimization considers the problem of online optimization subject to distributed data sources [212]. In general, in a distributed online optimization framework, the learners update local parameters based on local data sources and periodically exchange information with a small subset of neighbors in a communication network [212, 213, 214, 215]. The goals of the distributed online optimization algorithms are usually to minimize the growth rate of the regret bound.

One possible approach is to adopt the distributed autonomous online learning method [212] to learn the optimal parameter policy as well as its value function and Q function. Similar to [214], each learner will exchange its parameters with the neighbors and perform local (sub)gradient descent. Notice that in multi-learner systems, it is not necessarily that all the learners have the same amount of data, or in this case, have experienced the same amount of tasks. This imbalance of data is not considered in the literature of distributed online optimization [212]. On the other hand, control policies trained with different amount data have different levels of performance uncertainties [51]. Therefore, similar to Chapter 3 and Chapter 5, how to quantify and be cognizant of the learning uncertainty in each learner is a challenge to obtain efficient collaboration for distributed online meta reinforcement learning.

Bibliography

[1] K. Matthews, "6 robotics applications demonstrating new tech markets," *The Robot Report*, May 2019.

[2] N. R. Gans and J. G. Rogers, "Cooperative multirobot systems for military applications," *Current Robotics Reports*, pp. 1–7, January 2021.

[3] K. Mizokami, "The navy takes another successful step toward mine-hunting robots," *Popular Mechanics*, September 2019.

[4] NASA, JPL-Caltech, and MSSS, "Mars report: Update on NASA's Perseverance Rover & Curiosity Rover," *NASA Science*, May 2021.

[5] "Size of the global market for industrial and non-industrial robots between 2018 and 2025," *Statista Research Department*, March 2021.

[6] K. Geihs, "Engineering challenges ahead for robot teamwork in dynamic environments," *Applied Sciences*, vol. 10, p. 1368, 2020.

[7] J. Rey, "How robots are transforming amazon warehouse jobs — for better and worse," *Vox Media*, December 2019.

[8] "U.S. Navy could 'swarm' foes with robot boats," *CNN*, October 2014.

[9] J. Peters, "Watch DARPA test out a swarm of drones," *The Verge*, August 2019.

[10] P. O'Dowd and A. Hagan, "Take a ride through phoenix in a driverless car," *WBUR*, January 2021.

[11] C. Flanagan, "Waymo's driverless ride service moves metro phoenix toward autonomous future," *Cronkite News*, December 2020.

[12] M. Machosky, "Two pittsburgh self-driving car giants combine forces as aurora acquires uber's advanced technologies group," *Next Pittsburgh*, December 2020.

[13] "Global autonomous cars market size to exceed usd 55.6 billion by 2032," *Spherical Insights*, November 2023.

[14] V. Trianni, "Multi-robot systems, swarm robotics and self-organisation," *Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, pp. 23–46, 2008.

[15] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, June 2000.

[16] G. Yu, K. Ashmawy, E. Huang, and W. Zeng, "Under the hood of uber atg's machine learning infrastructure and versioning control platform for self-driving vehicles," *Uber Engineering*, March 2020.

[17] J. M. Fossaceca and S. H. Young, "Artificial intelligence and machine learning for future army applications," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*, vol. 10635, p. 1063507, International Society for Optics and Photonics, May 2018.

[18] D. S. Hoadley and N. J. Lucas, "Artificial intelligence and national security," *Congressinal Research Service*, April 2018.

[19] D. Yadron and D. Tynan, "Tesla driver dies in first fatal crash while using autopilot mode," *The Guardian*, June 2016.

[20] P. McCausland, "Self-driving uber car that hit and killed woman did not recognize that pedestrians jaywalk," *NBC News*, November 2019.

[21] F. Keating, "The 'suicidal robot' that drowned in a fountain didn't kill itself after all," *Independent*, July 2017.

[22] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for Big Data," *Information Fusion*, vol. 42, pp. 146–157, July 2018.

[23] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Magazine*, vol. 40, pp. 44–58, June 2019.

[24] L. Longo, R. Goebel, F. Lecue, P. Kieseberg, and A. Holzinger, "Explainable artificial intelligence: Concepts, applications, research challenges and visions," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 1–16, August 2020.

[25] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[26] E. Ackerman, "Slight street sign modifications can completely fool machine learning algorithms," *IEEE Spectrum*, vol. 6, p. 103, 2019.

[27] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, August 2015.

[28] T. Moldovan and P. Abbeel, "Risk aversion in markov decision processes via near optimal Chernoff bounds," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3131–3139, 2012.

[29] A. Gosavi, "Reinforcement learning for model building and variance-penalized control," in *Proc. Winter Simulation Conference (WSC)*, pp. 373–379, 2009.

[30] N. Abe, P. Melville, C. Pendus, C. K. Reddy, D. L. Jensen, V. P. Thomas, J. J. Bennett, G. F. Anderson, B. R. Cooley, M. Kowalczyk, M. Domick, and T. Gardinier, "Optimizing debt collections using constrained reinforcement learning," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 75–84, 2010.

[31] A. Tamar, D. Di Castro, and S. Mannor, "Policy gradients with variance related risk criteria," in *Proc. Int. Conf. Machine Learning (ICML)*, p. 1651–1658, 2012.

[32] M. Heger, "Consideration of risk in reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 105–111, 1994.

[33] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, pp. 780–798, October 2005.

[34] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 6059–6066, 2018.

[35] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite Markov decision processes with Gaussian processes," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4312–4320, 2016.

[36] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, pp. 2460–2465, 2018.

[37] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 997–1005, 2015.

[38] I. Usmanova, A. Krause, and M. Kamgarpour, "Safe convex learning under uncertain constraints," in *Int. Conf. Artificial Intelligence and Statistics*, pp. 2106–2114, 2019.

[39] Y. Sui, V. Zhuang, J. Burdick, and Y. Yue, "Stagewise safe Bayesian optimization with Gaussian processes," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 4781–4789, 2018.

[40] W. Sun, D. Dey, and A. Kapoor, "Safety-aware algorithms for adversarial contextual bandit," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 3280–3288, 2017.

[41] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 908–918, 2017.

[42] A. Wachi, Y. Sui, Y. Yue, and M. Ono, "Safe exploration and optimization of constrained MDPs using Gaussian processes.," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, pp. 6548–6556, 2018.

[43] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 22–31, 2017.

[44] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Trans. Automatic Control*, vol. 64, no. 7, pp. 2737–2752, October 2018.

[45] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. N. Zeilinger, J. H. Gillula, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes.," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 1424–1431, 2014.

[46] Z. Zhou, O. S. Oguz, M. Leibold, and M. Buss, "A general framework to increase safety of learning algorithms for dynamical systems based on region of attraction estimation," *IEEE Trans. Robotics*, vol. 36, no. 5, pp. 1472–1490, June 2020.

[47] T. Lew, A. Sharma, J. Harrison, A. Bylard, and M. Pavone, "Safe active dynamics learning and control: A sequential exploration–exploitation framework," *IEEE Trans. Robotics*, vol. 38, no. 5, pp. 2888–2907, May 2022.

[48] D. K. Jha, M. Zhu, Y. Wang, and A. Ray, "Data-driven anytime algorithms for motion planning with safety guarantees," in *Proc. American Control Conf. (ACC)*, pp. 5716–5721, 2016.

[49] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, no. 3, pp. 387–434, November 2005.

[50] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, pp. 50–60, February 2020.

[51] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

[52] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*. Springer, 2008.

[53] S. Chinchali, A. Sharma, J. Harrison, A. Elhafsi, D. Kang, E. Pergament, E. Cidon, S. Katti, and M. Pavone, "Network offloading policies for cloud robotics: a learning-based approach," *Autonomous Robots*, vol. 45, no. 7, pp. 997–1012, July 2021.

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

[55] T. Choi and M. J. Schervish, "On posterior consistency in nonparametric regression problems," *Journal of Multivariate Analysis*, vol. 98, no. 10, pp. 1969–1987, November 2007.

[56] K. Ritter, *Average-Case Analysis of Numerical Problems*. Springer, 2007.

[57] M. P. Deisenroth, C. E. Rasmussen, and J. Peters, "Gaussian process dynamic programming," *Neurocomputing*, vol. 72, no. 7-9, pp. 1508–1524, March 2009.

[58] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, pp. 9–15, 2016.

[59] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Autonomous Robots*, vol. 39, no. 3, pp. 221–238, October 2015.

[60] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, October 2020.

[61] A. V. Vecchia, "Estimation and model identification for continuous spatial processes," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 50, no. 2, pp. 297–312, January 1988.

[62] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, "Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets," *Journal of the American Statistical Association*, vol. 111, no. 514, pp. 800–812, August 2016.

[63] A. O. Finley, A. Datta, B. D. Cook, D. C. Morton, H. E. Andersen, and S. Banerjee, "Efficient algorithms for Bayesian nearest neighbor Gaussian processes," *Journal of Computational and Graphical Statistics*, vol. 28, no. 2, pp. 401–414, April 2019.

[64] M. Tavassolipour, S. A. Motahari, and M. T. M. Shalmani, "Learning of Gaussian processes in distributed and communication limited systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 68, no. 2, pp. 987–997, March 2019.

[65] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Trans. Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, July 2015.

[66] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Trans. Information Theory*, vol. 55, no. 4, pp. 1856–1871, April 2009.

[67] G. Pillonetto, L. Schenato, and D. Varagnolo, "Distributed multi-agent gaussian regression via finite-dimensional approximations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2098–2111, September 2019.

[68] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468–2481, July 2012.

[69] S. Martínez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Trans. Control Systems Technology*, vol. 18, no. 2, pp. 491–500, March 2010.

[70] Y. Xu, J. Choi, S. Dass, and T. Maiti, "Efficient Bayesian spatial prediction with mobile sensor networks using Gaussian Markov random fields," *Automatica*, vol. 49, no. 12, pp. 3520–3530, October 2013.

[71] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, December 2009.

[72] M. Zhu and S. Martínez, *Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments*. Springer, 2015.

[73] R. Yu, Z. Yuan, M. Zhu, and Z. Zhou, "Data-driven distributed state estimation and behavior modeling in sensor networks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 8192–8199, 2020.

[74] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk, *A distribution-free theory of nonparametric regression*. Springer, 2002.

[75] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, August 2002.

[76] V. Tresp, "A bayesian committee machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, November 2000.

[77] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust Bayesian committee machine for large-scale Gaussian process regression," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 3131–3140, 2018.

[78] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, February 2010.

[79] N. A. Lynch, *Distributed Algorithms*. Elsevier, 1996.

[80] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semi-parametric learning for inverse dynamics modeling," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 2945–2950, 2016.

[81] M. F. Huber, "Recursive Gaussian process: On-line regression and learning," *Pattern Recognition Letters*, vol. 45, pp. 85–91, August 2014.

[82] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[83] E. Mueller, M. Zhu, S. Karaman, and E. Frazzoli, "Anytime computation algorithms for approach-evasion differential games," *arXiv preprint arXiv:1308.1174*, 2013.

[84] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Information Theory*, vol. 58, no. 5, pp. 3250–3265, January 2012.

[85] M. R. Abbasifard, B. Ghahremani, and H. Naderi, "A survey on nearest neighbor search methods," *International Journal of Computer Applications*, vol. 95, no. 25, June 2014.

[86] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[87] Á. Besenyei, "Picard's weighty proof of Chebyshev's sum inequality," *Mathematics Magazine*, vol. 91, no. 5, pp. 366–371, December 2018.

[88] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. New Delhi, India: Tata McGraw-Hill Education, 2002.

[89] C.-T. Chen, *Linear System Theory and Design, 3rd Ed.* Oxford University Press, Inc., 1999.

[90] R. V. Hogg, E. A. Tanis, and D. L. Zimmerman, *Probability and Statistical Inference*. Macmillan New York, 1977.

[91] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[92] L. Lindemann, M. Cleaveland, Y. Kantaros, and G. J. Pappas, "Robust motion planning in the presence of estimation uncertainty," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 5205–5212, 2021.

[93] M. H. Cohen, C. Belta, and R. Tron, "Robust control barrier functions for nonlinear control systems with uncertainty: A duality-based approach," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 174–179, 2022.

[94] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, "Safe feedback motion planning: A contraction theory and l 1-adaptive control based approach," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 1578–1583, 2020.

[95] M. Omainska, J. Yamauchi, T. Beckers, T. Hatanaka, S. Hirche, and M. Fujita, "Gaussian process-based visual pursuit control with unknown target motion learning in three dimensions," *SICE Journal of Control, Measurement, and System Integration*, vol. 14, no. 1, pp. 116–127, June 2021.

[96] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, vol. 39, no. 4, pp. 555–571, August 2015.

[97] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, April 2020.

[98] N. Virani, D. K. Jha, Z. Yuan, I. Shekhawat, and A. Ray, "Imitation of demonstrations using Bayesian filtering with nonparametric data-driven models," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 3, p. 030906, March 2018.

[99] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2616–2625, 2017.

[100] A. Majumdar and M. Goldstein, "PAC-Bayes control: Synthesizing controllers that provably generalize to novel environments," in *Conf. Robot Learning (CoRL)*, pp. 293–305, 2018.

[101] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Trans. Automatic Control*, vol. 67, no. 1, pp. 176–188, January 2022.

[102] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Trans. Automatic Control*, vol. 68, no. 5, pp. 2638–2651, May 2023.

[103] R. Cosner, M. Tucker, A. Taylor, K. Li, T. Molnar, W. Ubelacker, A. Alan, G. Orosz, Y. Yue, and A. Ames, "Safety-aware preference-based learning for safety-critical control," in *Learning for Dynamics and Control Conference*, pp. 1020–1033, 2022.

[104] S. Curi, A. Lederer, S. Hirche, and A. Krause, "Safe reinforcement learning via confidence-based filters," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 3409–3415, 2022.

[105] N. Kochdumper, H. Krasowski, X. Wang, S. Bak, and M. Althoff, "Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes," *IEEE Open Journal of Control Systems*, vol. 2, pp. 79–92, March 2023.

[106] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Annual Symposium on Foundations of Computer Science*, pp. 421–427, 1979.

[107] J. T. Schwartz and M. Sharir, "On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds," *Advances in applied Mathematics*, vol. 4, no. 3, pp. 298–351, September 1983.

[108] G. Zhao and M. Zhu, "Pareto optimal multirobot motion planning," *IEEE Trans. Automatic Control*, vol. 66, no. 9, pp. 3984–3999, 2021.

[109] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, no. 2, pp. 229–243, February 2006.

[110] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1213–1223, October 2008.

[111] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robotics*, vol. 33, no. 3, pp. 661–674, June 2017.

[112] G. Zhao and M. Zhu, "Scalable distributed algorithms for multi-robot near-optimal motion planning," *Automatica*, vol. 140, p. 110241, June 2022.

[113] K. E. Bekris, D. K. Grady, M. Moll, and L. E. Kavraki, "Safe distributed motion coordination for second-order systems with different planning cycles," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 129–150, February 2012.

[114] X. Ma, Z. Jiao, Z. Wang, and D. Panagou, "3-d decentralized prioritized motion planning and coordination for high-density operations of micro aerial vehicles," *IEEE Trans. Control Systems Technology*, vol. 26, no. 3, pp. 939–953, May 2018.

[115] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, June 2011.

[116] T. Pan, C. K. Verginis, A. M. Wells, L. E. Kavraki, and D. V. Dimarogonas, "Augmenting control policies with motion planning for robust and safe multi-robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 6975–6981, 2020.

[117] Y. Zhou, H. Hu, Y. Liu, S.-W. Lin, and Z. Ding, "A distributed approach to robust control of multi-robot systems," *Automatica*, vol. 98, pp. 1–13, December 2018.

[118] A. D. Saravanos, A. Tsolovikos, E. Bakolas, and E. A. Theodorou, "Distributed covariance steering with consensus admm for stochastic multi-agent systems.," in *Proc. Robotics: Science and Systems (RSS)*, 2021.

[119] H. Zhu, B. Brito, and J. Alonso-Mora, "Decentralized probabilistic multi-robot collision avoidance using buffered uncertainty-aware Voronoi cells," *Autonomous Robots*, vol. 46, no. 2, pp. 401–420, January 2022.

[120] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 777–783, 2020.

[121] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, pp. 6252–6259, 2018.

[122] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, May 2020.

[123] G. Zhao and M. Zhu, "Pareto optimal multi-robot motion planning," *IEEE Trans. Automatic Control*, vol. 66, no. 9, pp. 3984–3999, September 2021.

[124] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre, "Set-valued numerical analysis for optimal control and differential games," in *Stochastic and differential games*, pp. 177–247, Springer, 1999.

[125] C. K. Batchelor and G. Batchelor, *An introduction to fluid dynamics*. Cambridge University Press, 2000.

[126] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[127] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, "Searching with consistent prioritization for multi-agent path finding," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, pp. 7643–7650, 2019.

[128] S. Zlobec, *Zermelo's Navigation Problems*. Boston, MA: Springer US, 2001.

[129] H. G. Tanner and A. Boddu, "Multiagent navigation functions revisited," *IEEE Trans. Robotics*, vol. 28, no. 6, pp. 1346–1359, 2012.

[130] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated robot navigation via hierarchical clustering," *IEEE Trans. Robotics*, vol. 32, no. 2, pp. 352–371, April 2016.

[131] K. Cole and A. Wickenheiser, "Impact of wind disturbances on vehicle station keeping and trajectory following," in *AIAA Guidance, Navigation, and Control Conference*, p. 4865, 2013.

[132] K. Cole and A. M. Wickenheiser, "Reactive trajectory generation for multiple vehicles in unknown environments with wind disturbances," *IEEE Trans. Robotics*, vol. 34, no. 5, pp. 1333–1348, October 2018.

[133] C. Danielson, K. Berntorp, A. Weiss, and S. Di Cairano, "Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner," *IEEE Trans. Automatic Control*, vol. 65, no. 10, pp. 4456–4463, October 2020.

[134] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, June 2017.

[135] Y. Kantaros, S. Kalluraya, Q. Jin, and G. J. Pappas, "Perception-based temporal logic planning in uncertain semantic maps," *IEEE Trans. Robotics*, vol. 38, no. 4, pp. 2536–2556, August 2022.

[136] Y. Fu, D. K. Jha, Z. Zhang, Z. Yuan, and A. Ray, "Neural network-based learning from demonstration of an autonomous ground robot," *Machines*, vol. 7, no. 2, p. 24, April 2019.

[137] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[138] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[139] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1282–1289, 2019.

[140] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, "A survey of zero-shot generalisation in deep reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 76, pp. 201–264, January 2023.

[141] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1126–1135, 2017.

[142] V. H. Pong, A. V. Nair, L. M. Smith, C. Huang, and S. Levine, "Offline meta-reinforcement learning with online self-supervision," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 17811–17829, 2022.

[143] K. Ji, J. D. Lee, Y. Liang, and H. V. Poor, "Convergence of meta-learning with task-specific adaptation over partial parameters," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11490–11500, 2020.

[144] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 5331–5340, 2019.

[145] X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably safe model-based meta reinforcement learning: An abstraction-based approach," in *Proc. IEEE Conf. Decision and Control (CDC)*, pp. 2963–2968, 2021.

[146] S. Xu and M. Zhu, "Meta value learning for fast policy-centric optimal motion planning," in *Proc. Robotics: Science and Systems (RSS)*, 2022.

[147] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1312–1320, 2015.

[148] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *International Journal of Robotics Research*, vol. 34, pp. 705–724, March 2015.

[149] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, April 2016.

[150] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, pp. 5620–5627, 2018.

[151] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 5872–5881, 2018.

[152] X. Fan, Y. Ma, Z. Dai, W. Jing, C. Tan, and B. K. H. Low, "Fault-tolerant federated reinforcement learning with theoretical guarantee," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1007–1021, 2021.

[153] S. Khodadadian, P. Sharma, G. Joshi, and S. T. Maguluri, "Federated reinforcement learning: Linear speedup under markovian sampling," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 10997–11057, 2022.

[154] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, January 2014.

[155] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 495–500, 2002.

[156] B. Fehrman, B. Gess, and A. Jentzen, "Convergence rates for the stochastic gradient descent method for non-convex objective functions," *Journal of Machine Learning Research*, vol. 21, no. 136, pp. 1–48, June 2020.

[157] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, January 2013.

[158] R. Bhattacharya, L. Lin, and V. Patrangenaru, *A course in mathematical statistics and large sample theory*. Springer, 2016.

[159] P. Mertikopoulos, N. Hallak, A. Kavis, and V. Cevher, "On the almost sure convergence of stochastic gradient descent in non-convex problems," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1117–1128, 2020.

[160] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

[161] H. Federer, *Geometric measure theory*. Springer, 2014.

[162] V. I. Bogachev, *Measure theory*, vol. 1. Springer Science & Business Media, 2007.

[163] "Pybullet," *https://pybullet.org/wordpress/*.

[164] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber, "Parameter-exploring policy gradients," *Neural Networks*, vol. 23, no. 4, pp. 551–559, 2010.

[165] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[166] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5302–5311, 2018.

[167] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Machine Learning (ICML)*, 2018.

[168] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1920–1930, 2019.

[169] D. A. E. Acar, R. Zhu, and V. Saligrama, "Memory efficient online meta learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 32–42, 2021.

[170] G. Denevi, C. Ciliberto, R. Grazzi, and M. Pontil, "Learning-to-learn stochastic gradient descent with biased regularization," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1566–1575, 2019.

[171] D. Grbic and S. Risi, "Safe reinforcement learning through meta-learned instincts," in *Artificial Life Conference Proceedings 32*, pp. 283–291, 2020.

[172] V. Khattar, Y. Ding, B. Sel, J. Lavaei, and M. Jin, "A CMDP-within-online framework for meta-safe reinforcement learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2022.

[173] J. Rothfuss, D. Lee, I. Clavera, T. Asfour, and P. Abbeel, "Promp: Proximal meta-policy search," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[174] N. C. Wagener, B. Boots, and C.-A. Cheng, "Safe reinforcement learning using advantage-based intervention," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 10630–10640, 2021.

[175] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[176] E. Hazan and C. Seshadhri, "Efficient learning algorithms for changing environments," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 393–400, 2009.

[177] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Machine learning*, vol. 32, no. 2, pp. 151–178, August 1998.

[178] N. Hallak, P. Mertikopoulos, and V. Cevher, "Regret minimization in stochastic non-convex learning via a proximal-gradient approach," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 4008–4017, 2021.

[179] M.-F. Balcan, M. Khodak, and A. Talwalkar, "Provable guarantees for gradient-based meta-learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 424–433, 2019.

[180] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[181] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1928–1937, 2016.

[182] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[183] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1889–1897, 2015.

[184] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1861–1870, 2018.

[185] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[186] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[187] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[188] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Conf. Robot Learning (CoRL)*, pp. 617–629, 2018.

[189] H. Liu, R. Socher, and C. Xiong, "Taming maml: Efficient unbiased meta-reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 4061–4071, 2019.

[190] A. G. Beccuti, S. Mariéthoz, S. Cliquennois, S. Wang, and M. Morari, "Explicit model predictive control of dc–dc switched-mode power supplies with extended kalman filtering," *IEEE Trans. Industrial Electronics*, vol. 56, no. 6, pp. 1864–1874, June 2009.

[191] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 60–70, April 2007.

[192] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying mpc for nonprehensile object manipulation with a nonholonomic mobile robot," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, pp. 11032–11038, 2020.

[193] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, September 2009.

[194] T. Mercy, W. Van Loock, and G. Pipeleers, "Real-time motion planning in the presence of moving obstacles," in *Proc. European Control Conf. (ECC)*, pp. 1586–1591, 2016.

[195] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *Journal of Computer and System Sciences*, vol. 71, no. 3, pp. 291–307, October 2005.

[196] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra, and S. Levine, "Model-based meta-reinforcement learning for flight with suspended payloads," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1471–1478, April 2021.

[197] S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi, "Fast global convergence of natural policy gradient methods with entropy regularization," *Operations Research*, vol. 70, no. 4, pp. 2563–2578, December 2022.

[198] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI GYM," *arXiv preprint arXiv:1606.01540*, 2016.

[199] T. Xu, Y. Liang, and G. Lan, "CRPO: A new approach for safe reinforcement learning with convergence guarantee," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 11480–11491, 2021.

[200] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, pp. 1–40, May 2016.

[201] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Multisource domain adaptation and its application to early detection of fatigue," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 4, pp. 1–26, December 2012.

[202] R. M. Neal, *Bayesian learning for neural networks*, vol. 118. Springer Science & Business Media, 2012.

[203] D. A. McAllester, "Some PAC-Bayesian theorems," *Machine Learning*, vol. 37, no. 3, pp. 355–363, 1999.

[204] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. ACM workshop on Security and Artificial Intelligence*, pp. 43–58, 2011.

[205] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Sok: Security and privacy in machine learning," in *IEEE European Symposium on Security and Privacy*, pp. 399–414, 2018.

[206] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, "Adversarial distributional training for robust deep learning," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8270–8283, 2020.

[207] W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson, "Verification for machine learning, autonomy, and neural networks survey," *arXiv preprint arXiv:1810.01989*, 2018.

[208] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Int. Conf. Computer Aided Verification*, pp. 97–117, 2017.

[209] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "ReachNN: Reachability analysis of neural-network controlled systems," *ACM Trans. Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–22, 2019.

[210] L. Janson, T. Hu, and M. Pavone, "Safe motion planning in unknown environments: Optimality benchmarks and tractable policies," in *Proc. Robotics: Science and Systems (RSS)*, 2018.

[211] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[212] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, November 2013.

[213] B. McMahan and M. Streeter, "Delay-tolerant algorithms for asynchronous distributed online learning," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, p. 2915–2923, 2014.

[214] Y. Wang, Y. Wan, S. Zhang, and L. Zhang, "Distributed projection-free online learning for smooth and convex losses," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 37, pp. 10226–10234, 2023.

[215] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *Journal of Machine Learning Research*, vol. 13, no. 1, p. 165–202, January 2012.

Vita

Zhenyuan Yuan

Zhenyuan Yuan is a Ph.D. candidate in the School of Electrical Engineering and Computer Science at the Pennsylvania State University. He received B.S. in Electrical Engineering and B.S. in Mathematics from the Pennsylvania State University in 2018. His research interests lie in machine learning and motion planning with applications in robotic networks. He is a recipient of the Rudolf Kalman Best Paper Award of the ASME Journal of Dynamic Systems Measurement and Control in 2019 and the Penn State Alumni Association Scholarship for Penn State Alumni in the Graduate School in 2021.