Title

Scalable Estimation and Decision-Making for Multi-Agent Systems

ISBN

9798186171355

Author

Wu, Zida

Publication Date

2026-07-29

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Scalable Estimation and Decision-Making for Multi-Agent Systems

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Zida Wu

2026

ABSTRACT OF THE DISSERTATION

Scalable Estimation and Decision-Making for Multi-Agent Systems

by

Zida Wu
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2026
Professor Ankur Mehta, Chair

Multi-agent systems have attracted attention in applications such as satellite networks and autonomous traffic. However, as the number of agents grows, the joint state, observation, and action spaces expand. This growth can lead to higher computational complexity, communication overhead, and unstable optimization. Scalability therefore asks whether agents can operate without their online burden growing with the total system size. We study scalability in estimation and decision-making, focusing on these two problems under uncertainty. In estimation, uncertainty appears through unknown inputs, unmeasured signals that nonetheless enter the dynamics. In decision-making, uncertainty arises from changing population distributions and common noise. Based on how agents are modeled and interact, we consider two regimes: the networked regime and the population regime. We primarily study estimation in the networked regime and decision-making in the population regime. In the networked regime, agents are finite and identifiable, and communicate over a time-varying graph. In the population regime, agents are numerous and interchangeable, each responding to the population distribution rather than to individuals.

In the networked regime, we develop recursive filters that jointly estimate the state and

unknown input. By decoupling the unknown-input and state-estimation steps, the filters remain unbiased and prevent model mismatch from contaminating the state estimate. Prior knowledge of the unknown input is further incorporated to reduce estimation variance, and its effect on estimator rank conditions is analyzed. These filters are then decentralized for heterogeneous networks with time-varying topology, where agents exchange only local estimates while approaching full-information accuracy. Because sensing and planning are coupled, we also study sensor coverage, whose placement admits a submodular approximation bound governed by a curvature constant. We prove that this curvature is inflated by the discretization itself and derive the explicit law by which it grows as the grid is refined.

In the population regime, we formulate the problem as a mean-field game and develop a Master Online Mirror Descent algorithm that learns a single policy network conditioned on the population distribution and common-noise history. Through iterative learning and a specialized regularizer, the policy converges stably toward Nash equilibria across different initial distributions and common-noise realizations. On benchmarks, it reduces exploitability faster than existing deep reinforcement learning methods, allowing each agent to make decisions independently even as the population approaches the infinite limit, while still driving the overall system toward equilibrium.

The dissertation of Zida Wu is approved.

Sriram Narasimhan

Lieven Vandenberghe

Bahman Gharesifard

Ankur Mehta, Committee Chair

University of California, Los Angeles

2026

To my parents and my steadfast friends

TABLE OF CONTENTS

1 Introduction **1**

1.1 Motivation 1

1.2 Two Regimes: Networked and Population 2

1.3 Related Work 5

1.3.1 Decentralized Input and State Estimation 5

1.3.2 Coverage Control for Unknown Event Densities 6

1.3.3 Learning Nash Equilibrium in Mean-field Game via Reinforcement Learning 8

1.4 Contribution Summary 9

1.5 Organization of the Dissertation 11

I Networked Systems: Estimation and Coverage under Exogenous Uncertainty **13**

2 Joint Input and State Estimation with Prior Knowledge **15**

2.1 Overview 15

2.2 Problem Formulation 18

2.2.1 System description 18

2.2.2 A two-step input-then-state recursion 19

2.3 Recap: The Minimum-Variance Unbiased Filter 20

2.3.1 Unbiased input estimation 21

2.3.2 Minimum-variance input gain 22

2.3.3 Unbiased minimum-variance state update 23
2.4 Input Estimation with a Discrete Prior: AMM-KF 24
2.4.1 Bayesian mode weights and the moment-matched update 25
2.4.2 Consistency and identifiability 26
2.4.3 Mode switches and safeguards 28
2.5 Input Estimation with a Continuous Prior: AL-RKF 28
2.6 Simulation Results 31
2.6.1 Experiment setup 31
2.6.2 Discrete prior: collision detection with AMM-KF 33
2.6.3 Identifiability of discrete modes 33
2.6.4 Continuous prior: wind-field estimation with AL-RKF 37
2.7 Conclusion 39

3 Decentralized Input and State Estimation 41
3.1 Overview 41
3.2 Problem Formulation 42
3.2.1 System description 42
3.3 Algorithm 44
3.3.1 Input fusion via Covariance Intersection 44
3.3.2 State fusion via information filter decomposition 45
3.4 Simulation and Experiment Result 48
3.4.1 Experiment Setup 48
3.4.2 Result 52
3.5 Conclusion 61

4 Input Estimation and Mobile Coverage Beyond the Rank Condition on Tree Networks . . . 63

4.1 Overview . . . 63

4.2 Problem Formulation . . . 64

4.2.1 Tree transport model . . . 64

4.2.2 Observations . . . 65

4.3 The Rank Obstruction and Its Restoration . . . 66

4.3.1 Full-input obstruction . . . 66

4.3.2 Structural reparameterization . . . 66

4.4 Equivalent Inputs and Identifiability . . . 67

4.4.1 What the reduced estimate means . . . 67

4.4.2 Sector resolution . . . 68

4.5 From Source Belief to Mobile Coverage . . . 70

4.5.1 Belief update . . . 70

4.5.2 Posterior and uncertainty densities . . . 70

4.5.3 Two planners . . . 71

4.6 Simulation Evidence and Limitations . . . 72

4.7 Conclusion . . . 73

5 Discretisation-Induced Curvature Inflation in Submodular Coverage . . . 74

5.1 Overview . . . 74

5.2 Coverage and Curvature on a Grid . . . 76

5.3 Area-Exponential Curvature Inflation . . . 77

5.4 Empirical Check and Resolution Diagnostic . . . 79

5.5 Discussion 79

II Population Systems: Equilibrium Learning under Endogenous Uncertainty 80

6 Population-Aware Online Mirror Descent: Learning Master Policies for Mean-Field Games 82

6.1 Overview 82

6.2 Problem Formulation 84

6.3 The Master OMD Algorithm 88

6.3.1 Online mirror descent with population-dependent Q-functions 88

6.3.2 Q-function update 89

6.3.3 Inner-loop replay buffer 91

6.3.4 Convergence of the Idealized Master OMD Dynamics 91

6.4 Experimental Results 95

6.4.1 Setup 95

6.4.2 Environment 1: exploration 98

6.4.3 Environment 2: beach bar 99

6.4.4 Environment 3: linear-quadratic 100

6.4.5 Generalization to the testing set 102

6.5 Discussion 104

6.6 Conclusion 105

7 Mean-Field Equilibria under Common Noise 106

7.1 Overview 106

7.2 Mean-Field Games with Common Noise . . . 107
7.3 Learning Master Policies under Common Noise . . . 109
7.4 Experimental Results . . . 111
7.4.1 Setup . . . 111
7.4.2 Beach Bar with Common Noise . . . 112
7.4.3 Linear-Quadratic with Common Noise . . . 112
7.5 Discussion . . . 114
7.6 Extension to Continuous State-Action Spaces . . . 116
7.7 Conclusion . . . 117

8 Conclusion and Future Directions . . . 118
8.1 Future Directions . . . 119

A Master OMD: Additional Experimental Details . . . 121
A.1 Training Hyperparameters . . . 121
A.2 Hyperparameter Sensitivity . . . 122
A.3 Ablation: The Inner-Loop Replay Buffer . . . 122
A.4 Scalability in Memory and Computation . . . 123
A.5 Scaling the Training Set: 30 Initial Distributions . . . 125
A.6 Policy Architecture . . . 125
A.7 Ad-Hoc Teaming (Outlook) . . . 127

B Common-Noise Experiment Details . . . 129

C Proofs and Convergence Analysis for Master OMD . . . 130

C.1 Proof of the Munchausen Equivalence 130
C.2 Equilibrium objects and the dynamics 131
C.3 Static theory: regret and propagation of monotonicity 133
C.4 The frozen stage game and perturbed mirror descent 135
C.5 Backward induction and the exploitability bound 137
C.6 Closures of the envelope hypothesis and sharpness 139

Bibliography **142**

LIST OF FIGURES

2.1 Unknown inputs encountered by an inspection blimp. 16
2.2 Simulation scenarios: airship in a wind field and swarm collision. 32
2.3 Collision detection confidence and state compensation with AMM-KF. 34
2.4 Slower mode convergence under large system noise. 35
2.5 Mode-weight evolution under three levels of mode separation. 36
2.6 Wind-speed and position estimation in the airship scenario. 37

3.1 Scenario 1: four static agents with quadrant-limited sensing. 49
3.2 Scenario 2: dynamic network of static and mobile agents. 50
3.3 Time-window review against bias from intermittent measurements. 52
3.4 Input estimation with 1-hop communication. 54
3.5 Observation time window mitigates intermittent-observation bias. 55
3.6 Target position estimates in x with 1-hop communication. 56
3.7 The y coordinate of position estimation of the target with 1-hop communication. 57
3.8 Estimated 2D trajectories with 1-hop communication. 57
3.9 Estimation error under dynamic topology and all-to-all communication. 58
3.10 Node 2 estimates without communication and with all-to-all communication. . . 60
3.11 Estimation error versus communication range under dynamic topology. 60

5.1 Certificate collapse versus greedy near-optimality under grid refinement. 75

6.1 Training and testing initial distributions for the multiple-μ_0 scenario. 97
6.2 One-room exploration: density evolution and exploitability. 99

6.3 Four-connected-rooms exploration: density evolution and exploitability. 100

6.4 Beach bar: distribution evolution and exploitability. 101

6.5 Linear-quadratic model: population evolution and exploitability. 103

7.1 Beach bar under the closure common noise. 113

7.2 Linear-quadratic model under the first common-noise shock sequence. 115

A.1 Temperature and learning-rate sweeps for M-OMD and V-OMD1. 122

A.2 Effect of the inner-loop replay buffer on M-OMD. 123

A.3 Model size comparison for M-OMD and M-FP. 124

A.4 Per-iteration cost of computing the exploitability. 124

A.5 True exploitability of M-OMD and M-FP with 30 initial distributions. 126

A.6 Exploitability across four Q-network architectures. 127

A.7 Ad-hoc teaming: new teams joining during the population's evolution. 128

LIST OF TABLES

1.1 The networked and population regimes, contrasted. 4

2.1 Wind-speed estimation error statistics across noise levels. 38

3.1 Evaluation of State Compensation Effect 59

3.2 Estimation error under stationary topology with heterogeneous sensors. 61

4.1 Single-flood policy comparison with drone-only sensing. 72

6.1 Exploitability on the testing set (200 training iterations). 103

A.1 Training hyperparameters for the exploration tasks. 121

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Professor Ankur Mehta. In addition to guiding the research presented in this dissertation, he supported my growth far beyond the technical aspects of research. His guidance in communication, academic presentation, collaboration, and navigating academic and social life in the United States helped me gradually become a more independent researcher.

I also thank my committee members, Professors Bahman Gharesifard, Lieven Vandenberghe, and Sriram Narasimhan, for their guidance and service. I gratefully acknowledge the funding support of the National Science Foundation, the Defense Advanced Research Projects Agency, and the National Aeronautics and Space Administration, which made this research possible.

I am also grateful to everyone with whom I had the opportunity to collaborate. I am especially thankful to Mathieu Laurière. His humility, rigor, reliability, and thoughtful approach to research made our collaboration both productive and genuinely enjoyable. I learned a great deal from working with him, and our collaboration remains one of the most rewarding experiences of my Ph.D. journey.

I will always remember the countless afternoons and evenings spent with my labmates, talking about research, life, newly formed ideas, and the problems that sometimes seemed impossible to solve. I am equally grateful to all the friends I met along the way. We traveled together, played basketball, worked out, played tennis and badminton, played video games, and went sailing. Whether they were nearby or thousands of miles away, their friendship supported me through many difficult moments. Finally, I would like to thank my parents in China. Although we were separated by a great distance, our frequent phone calls ensured that I never felt far from the care and support of my family. All of these people and memories formed the steady foundation that carried me through my Ph.D.

The Ph.D. journey passed remarkably quickly, yet at times felt extraordinarily long. It

took me many years to move from learning how to answer questions to learning how to ask them, and nearly every step unfolded differently from what I had expected. In moments of excitement, I sometimes believed that a new result might transform an entire field. In moments of anxiety, I experienced more sleepless nights than I had in all the years before graduate school.

It also took me years to understand the meaning of the word "philosophy" in Doctor of Philosophy. It is not only about acquiring knowledge or solving a well-defined problem. It is about learning to identify questions worth pursuing, living with uncertainty, and continuing to move forward when no answer is yet in sight. During the periods when research was not progressing as I had hoped, I often found myself discussing life and philosophy with fellow Ph.D. students: where we came from, what kind of lives suited us, and who we might eventually become. Only later did I realize that many people and moments I once took for granted were as fleeting as the pink evening sky over Los Angeles, disappearing before I had fully learned to appreciate them.

I still remember the nervousness of giving my first conference presentation in Philadelphia. Uncertain and overwhelmed, I followed Wenzhong closely and asked questions about nearly everything. In the years that followed, I presented my work in Milan, Rio de Janeiro, Seattle, and San Diego. Along the way, I met many exceptionally talented people and gradually learned that meaningful research requires more than ability and opportunity. It requires the patience to remain with a difficult problem, the willingness to put in sustained effort, and the discipline to stay with a chosen direction even when other paths appear easier or more immediately rewarding.

Looking back, what I gained from this journey extends far beyond the dissertation itself. I have developed a steadier mindset, a healthier and more active body, a more accepting outlook, and even better cooking skills. These years were filled with both meetings and farewells. I said goodbye to people I had known at different stages and in different roles, and through each farewell, I also gradually said goodbye to earlier versions of myself.

I am now ready to begin a new chapter. From this point forward, I will make my own choices and take responsibility for where they lead. I will continue searching for questions worth answering, as well as for my own Smurfs and azaleas.

May the road ahead be kind to me, and may I meet it with curiosity, humility, and courage.

VITA

2017	B.S. (Telecommunication Engineering), Xidian University.
2020	M.S. (Electronics and Communication Engineering), Shanghai Jiao Tong University.

PUBLICATIONS

Wu, Z., Cho, W., Martinez, D., and Mehta, A. Discretisation-Induced Curvature Inflation in Submodular Coverage. (Under review)

Magnino, L., Shao, K., **Wu, Z.**, Shen, J., & Laurière, M. (2025). Solving continuous mean field games: Deep reinforcement learning for non-stationary dynamics. Advances in Neural Information Processing Systems, 38, 104325-104354.

Wu, Z., Laurière, M., Geist, M., Pietquin, O., & Mehta, A. (2025, December). Population-aware Online Mirror Descent for Mean-Field Games with Common Noise by Deep Reinforcement Learning. In 2025 IEEE 64th Conference on Decision and Control (CDC) (pp. 6338-6345). IEEE.

Wu, Z., & Mehta, A. (2024, December). Decentralized Input and State Estimation for Multi-agent System with Dynamic Topology and Heterogeneous Sensor Network. In 2024 IEEE 63rd Conference on Decision and Control (CDC) (pp. 2740-2747). IEEE.

Wu, Z., Laurière, M., Chua, S. J. C., Geist, M., Pietquin, O., & Mehta, A. (2024, May). Population-aware Online Mirror Descent for Mean-Field Games by Deep Reinforcement Learning. In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (pp. 2561-2563).

Gorr, B., Jaramillo, A. A., **Wu, Z.**, Cho, W., Cheng, K., Stroud, M., …& Selva, D. (2023, July). Multi-instrument flood monitoring with a distributed, decentralized, dynamic and context-aware satellite sensor web. In IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium (pp. 4602-4605). IEEE.

Wu, Z., Zheng, Z., & Mehta, A. (2022, May). Joint state and input estimation of agent based on recursive Kalman filter given prior knowledge. In 2022 International Conference on Robotics and Automation (ICRA) (pp. 1675-1681). IEEE.

CHAPTER 1

Introduction

1.1 Motivation

Environmental monitoring, transportation, and robot teams often require sensing or action at many locations at once. Satellite sensor networks can monitor floods across a large region [39, 55]. Mobile robots can divide a coverage task [24], while vehicles and aerial robots can coordinate their motion [11, 22]. A larger team can collect more observations or act in more places at the same time. It also produces more data and more interactions. The team is useful at scale only if each member can work without tracking the entire system.

Scale affects several kinds of work. As a system grows, one agent cannot be expected to observe every other agent or the whole environment. Sending all measurements to one place or exchanging them with everyone consumes bandwidth and adds delay. Storing every agent's model and history requires more memory, while inference or planning over joint state and action spaces requires more computation. These demands are connected because collected data must still be transmitted, stored, and processed before another agent can use it. Consensus and distributed estimation study how local measurements are shared [63, 91]. Distributed control, formation, swarming, and coverage study how agents move using local information [90, 22, 24]. For decision problems with many interacting agents, mean-field models replace pairwise interactions with a population distribution [70, 54]. Here scalability means that adding agents should not force one agent to sense, communicate with, compute over, or remember the whole system. The results below address part of this requirement,

with a focus on estimation and decision-making.

Estimation and decision-making are two steps in an observe–estimate–act loop. Estimation asks what an agent can infer from local, noisy observations and limited exchange with its neighbors. If the estimate is poor, later actions are based on the wrong information. Decision-making asks what an agent should do when other agents' responses also affect its outcome. A good estimate alone does not settle that question. Coverage provides one connection between the two tasks: the current estimate can guide where a team takes its next measurements.

We study these questions in two regimes. In the *networked regime*, agents are finite and identifiable. We study how they estimate unknown inputs, such as wind or collision forces, uncommanded actuator inputs caused by faults, and excess river inflow, from local observations and neighbor messages. In the *population regime*, agents are numerous and interchangeable. Each responds to the population distribution rather than to every other agent individually. We study how they choose policies when their actions produce the population response and when common noise, a shared disturbance, makes that response random. The next section states the estimation and decision problems in these two regimes.

1.2 Two Regimes: Networked and Population

The networked and population regimes use different models of interaction. We now state the uncertainty, objective, and scalability requirement in each one.

In the networked regime, agents are finite and identifiable. Each has its own state, dynamics, and sensors, and communicates with a few neighbors over a sparse, time-varying graph. We focus on an unknown input, an unmeasured physical signal that enters the dynamics. An estimator based only on the nominal model does not account for this signal. Prior information can improve the input estimate, while local exchange avoids sending every raw measurement and model to a central node. We develop recursive filters that estimate the state and the unknown input, and then decentralize them so that agents exchange

local estimates. We also study sensor placement because where the team measures affects what it can estimate. The scalability goal is to avoid making one agent's computation and communication depend on the total number of agents.

In the population regime, agents are numerous, identical, and interchangeable. Each agent responds to the population distribution rather than to named individuals. The agents' policies generate this distribution, which in turn affects their rewards and dynamics. Common noise is a shared disturbance that makes the population evolution random. A policy update therefore changes the population to which the policy is responding, so estimation alone does not determine what an agent should do. We formulate the problem as a mean-field game and seek a Nash equilibrium, where no agent gains by changing its policy alone. We learn one policy conditioned on the population distribution and the common-noise history. Each agent can then act without tracking every other agent, even in the infinite-population model.

Table 1.1 contrasts the two regimes. Part I treats the networked regime; Part II treats the population regime and contains the principal contributions.

A note on scope. The two regimes share a question, scalability under uncertainty, rather than a single algorithm. Recursive filtering, submodular coverage, and reinforcement learning for equilibria are distinct bodies of theory, and each part can be read on its own. Two connections are demonstrated only in part. Within the networked regime, Chapter 4 combines tree-structured input estimation with a limited mobile-sensing simulation that illustrates how the resulting belief can guide new measurements. Across the regimes, the common noise in the linear-quadratic benchmark enters each agent's dynamics as an additive external disturbance. Its realized history is supplied directly to the policy as it becomes causally available; it is not inferred or recovered from population observations.

Table 1.1: The two regimes, contrasted by how the agents are modeled and interact, which task each regime focuses on, the form the uncertainty takes, and what each regime aims to guarantee.

	Networked regime	**Population regime**
Agent model	finite and identifiable	numerous, identical, and interchangeable
Interaction	local sensing over a sparse, time-varying graph	response to the population distribution
Focus	estimation	decision-making
Uncertainty	unknown inputs entering the dynamics	changing population distribution and common noise
Solution / guarantee	unbiased, minimum-variance estimates	Nash equilibrium (low exploitability)
Machinery	recursive and decentralized filtering, submodular coverage	mean-field games, deep reinforcement learning, mirror descent

1.3 Related Work

1.3.1 Decentralized Input and State Estimation

The networked regime requires estimates built from local observations and messages between neighbors. This problem belongs to the broader literature on decentralized estimation. Decentralized methods can scale to larger networks, require less synchronization, and continue operating after some node failures [63]. In this dissertation, 'decentralized' means that each agent accesses information only from its immediate neighbors.

Decentralized estimation seeks fusion rules that combine information from adjacent sensors so that local nodes agree on a common estimate. Following the categorization of [46], existing algorithms divide into consensus-based [91, 92, 58, 59], gossip-based [83], and diffusion-based [18, 113] methods, according to how nodes interact and fuse information. In consensus-based algorithms, each node computes a weighted average of the states received from its neighbors, iterating until discrepancies are minimized. Gossip-based algorithms likewise iterate, but exchange messages randomly and asynchronously within neighborhoods, avoiding aggregation over all nodes. Diffusion-based methods require only a single fusion step between observations, averaging states after one exchange. All three families place heavy demands on communication rate, range, and bandwidth, and they introduce correlations among the fused estimates. Some algorithms reach consensus in a single iteration: max-consensus [100] synchronizes each neighborhood to the node with the minimum variance, and pseudo-estimate methods [41, 40] attain the globally optimal estimate, but only under a strict full communication rate and large bandwidth, and without local optimality.

Beyond state estimation, many scenarios involve unknown inputs acting on the system dynamics, for example in geophysical detection, environmental monitoring [39, 55], maneuver tracking [74], disturbance detection in flight [125], and sensor fault detection [30]. Such inputs, typically produced by unforeseen events, must be estimated alongside the state. For a single agent, joint unknown-input and state estimation is well developed, notably through

the recursive Kalman filter (RKF) of [36]. Decentralized estimation with unknown inputs generally adopts one of two strategies: each agent estimates the input from its own information alone [96], or agents share system equations and raw observations with their neighbors and run a local recursive Kalman filter [77]. The first approach yields no consensus on the input; the second demands unrestricted bandwidth and exposes private agent information.

1.3.2 Coverage Control for Unknown Event Densities

Efficient coverage control in multi-agent systems is important for applications such as environmental monitoring, target tracking, and disaster response. Classic coverage algorithms assume a static target distribution and seek to position agents to optimally cover that density. A representative example is the Lloyd algorithm (centroidal Voronoi tessellation), where each robot moves toward the centroid of its assigned Voronoi cell, weighted by the importance density [24]. This distributed, gradient-based approach converges to a configuration that locally maximizes coverage of the area of interest. Such methods have proven effective in applications like static environmental monitoring and surveillance. Variations of the classic approach have addressed challenges like finite sensing ranges[102] and heterogeneous capabilities [76], but a common assumption is that the density function is known and stationary in time.

In contrast, time-varying coverage control deals with scenarios where the target distribution evolves over time. Examples include tracking a moving phenomenon (e.g., an oil spill or a spreading wildfire) or responding to periodic changes in demand. To handle dynamic densities, robots must continuously adjust their positions as the "hotspots" move or intensify. Prior works have extended coverage algorithms to time-varying settings, often by treating the density $\phi(x, t)$ as a known function of time and deriving control laws for the robots to track its changes [73] [61]. A challenge in time-varying coverage is achieving a balance between responsiveness and stability: the team should respond quickly to significant distribution shifts, but over-aggressive movement can be inefficient or lead to oscillations. If the distribution

changes in unforeseen ways, solely reacting to known changes is insufficient. Recent research emphasizes the importance of coupling estimation with control in dynamic environments, that is, actively exploring to detect new high-density regions while covering according to the latest estimate [102].

Therefore, a key challenge in these applications is estimating an unknown target distribution $\phi(x,t)$, which represents the spatial density of interest points within the environment. Accurately estimating $\phi(x,t)$ is essential for optimal sensor deployment, as it allows agents to prioritize high-relevance areas while minimizing redundancy. Various approaches have been proposed to estimate $\phi(x,t)$ for coverage optimization. Gaussian-based models, such as multivariate Gaussian models and Gaussian processes (GPs), have been widely used for their ability to capture spatial correlations and uncertainty. In [107], a network of agents achieves consensus on a Gaussian-modeled distribution, improving estimation robustness in a decentralized manner. The statistical-learning approach of [1] models spatial distributions with kernel and prior mean functions learned from data to minimize estimation uncertainty, while Gaussian mixture models are employed in [82] to approximate complex, multimodal distributions in uncertain environments. Beyond probabilistic modeling, some methods estimate the distribution through target-based weighting: the method of [76] assigns weighted cost functions to observed targets and dynamically adjusts coverage toward detected high-priority areas.

A complementary line of work provides *performance certificates* for sensor-placement and coverage planners through submodular maximization: greedy selection enjoys the classical $1-1/e$ guarantee, and curvature-dependent refinements sharpen it. These guarantees, however, are typically stated for a fixed finite ground set obtained by discretizing the continuous domain. Chapter 5 examines what happens to such certificates as the discretization is refined, showing that curvature-based multiplicative bounds can become vacuous in the continuum limit while greedy-envelope bounds survive.

1.3.3 Learning Nash Equilibrium in Mean-field Game via Reinforcement Learning

Multi-agent systems (MAS) [31] are common in real-life scenarios involving many players, such as flocking [90, 25], traffic flow [11], and swarm robotics [22]. While many MAS models are purely dynamical, others incorporate rationality through game theory. As the number of players grows, computational efficiency becomes a major challenge [80, 103]. Under symmetry and homogeneity assumptions, however, mean-field approximations offer an effective way to model population behavior and learn decentralized policies while avoiding the curse of dimensionality.

Mean field games (MFGs) [70, 54, 15, 5] provide a framework for large-population games where agents interact through the population distribution. As the number of agents grows, individual influence diminishes, reducing interactions to those between a representative agent and the population. The solution concept in MFGs is the Nash equilibrium, where no player has an incentive to deviate unilaterally. Learning methods for MFGs often rely on fixed-point iterations, which update agent policies and mean-field terms iteratively. However, convergence of these methods requires strict contraction conditions [75, 42], which often fail in practice [26, 3].

To address these limitations, smoothing-based approaches such as Fictitious Play (FP) have been introduced. Originally developed for finite-player games [10, 7], FP has been extended to MFGs [14, 45, 99]. FP averages historical distributions and converges under a structural assumption (Lasry–Lions monotonicity) rather than a strict contraction property, and its learning-based implementations extend naturally to continuous spaces. FP nevertheless faces computational challenges: averaging policies is difficult with non-linear approximators such as deep neural networks (DNNs) [72], computing a best response at every iteration is costly, and uniform averaging over all past iterations slows the update rate as iterations accumulate.

Online Mirror Descent (OMD) addresses these issues by aggregating past Q-functions instead of policies [43, 44, 97]. Unlike FP, OMD requires only policy evaluation, not optimization, and maintains a constant update rate. Although explicitly summing Q-functions is costly with DNNs, a deep RL adaptation was proposed in [72] using the Munchausen trick [115], which provides implicit regularization. However, most existing methods compute the equilibrium for a single, fixed initial distribution and assume policies that depend only on the individual state. They must be re-run from scratch whenever the initial distribution changes, and they cannot react to common noise that perturbs the whole population. Part II of this dissertation develops population-dependent (master) policies that lift both restrictions.

1.4 Contribution Summary

Part I: Estimation and Coverage in the Networked Regime We develop recursive filters that jointly estimate the state and the unknown input, and decentralize them for large sensor networks. We then extend input estimation to the rank-deficient regime through structural priors on a tree network, use a limited mobile-sensing simulation to illustrate how the resulting belief can guide measurements, and analyze when coverage guarantees survive discretization.

- Single-agent joint input and state estimation that incorporates prior knowledge of the unknown input to reduce estimation variance. Discrete input modes are handled by a Bayesian multiple-model filter (AMM-KF) whose mode posterior is shown to be consistent under an identifiability condition, with a hypothesis-testing analysis that quantifies how mode separation limits single-step detection. Interval priors are handled by the augmented-Lagrangian Kalman filter (AL-RKF), whose exact constrained solution reduces to clipping under the stated rank conditions. The unconstrained RKF is unbiased under its rank condition; an active interval constraint may trade bias for lower variance.

- A decentralized input and state Kalman filter (DISKF) for time-varying topologies and heterogeneous sensors. Input estimates are fused by covariance intersection and state estimates by an information-filter decomposition; nodes exchange only input and state estimates with their neighbors, not system equations or raw observations, which lowers communication cost and preserves privacy. The filter restores estimability when local observability fails, reviews an observation time window to handle intermittent measurements, and approaches full-information accuracy.

- Input estimation beyond the rank condition, for settings where thousands of candidate input locations are watched by only a few dozen sensors and no unbiased input estimator exists. Structural prior knowledge, the directed-tree geometry of a river network, restores a solvable problem: a reduced-input estimator exploits the lower-triangular routing operator to recover input anomalies on whatever sub-network is currently observed, identifiability provably collapses to observation-defined sectors, and a Bayesian belief summarizes the remaining source uncertainty. A limited simulation on the 5,131-reach San Antonio–Guadalupe basin illustrates how this belief can guide a small mobile sensor fleet.

- A proof that refining the discretization grid inflates the total curvature of submodular coverage objectives at an area-exponential rate. Curvature-based multiplicative certificates then become vacuous in the continuum limit, even when the placement stays near-optimal, while greedy-envelope certificates retain their guarantee. A practical diagnostic identifies when a certificate is still reliable.

Part II: Equilibrium Learning in the Population Regime (principal contribution) We formulate large-population decision-making as a mean-field game and learn equilibrium policies that generalize across the population's configuration.

- Master OMD (M-OMD): a population-aware deep reinforcement learning algorithm

built on Online Mirror Descent and the Munchausen trick. It learns master policies, which take the population distribution as input, and reaches Nash equilibrium from initial distributions not seen in training, with constant memory. For an idealized tabular version of the dynamics, we prove convergence under strong Lasry–Lions monotonicity and show by counterexample that the strong-monotonicity assumption cannot be dropped.

- An extension of M-OMD to mean-field games with common noise, a shared shock that randomizes the mean-field evolution. The learned policy conditions on the realized common-noise history and reproduces the equilibrium behavior along noise realizations not revealed in advance; a specialized regularizer and iterative learning give stable convergence.

- Faster convergence, that is, lower exploitability per iteration, than state-of-the-art baselines, including Master Fictitious Play, across standard benchmarks.

1.5 Organization of the Dissertation

In brief, Chapters 2–5 study networked multi-agent systems whose uncertainty is imposed from outside, and Chapters 6–7 study population systems whose uncertainty is produced by the agents themselves. The two parts trace this shift and are followed by a concluding chapter.

Part I (Networked systems). Chapter 2 develops joint input and state estimation for a single agent with prior knowledge of the input (adapted from a paper presented at ICRA 2022). Chapter 3 extends the machinery to decentralized sensor networks with dynamic topology and heterogeneous sensors (adapted from a paper presented at CDC 2024). Chapter 4 studies input estimation when the rank condition fails, shows how the tree structure of a river network restores a solvable problem, and includes a limited mobile-sensing simulation based on the

resulting source belief (adapted from a manuscript in preparation). Chapter 5 proves how grid refinement affects submodular curvature certificates (adapted from a manuscript under review at TMLR) and is deliberately kept concise.

Part II (Population systems). Chapter 6 develops population-aware Online Mirror Descent, learning master policies for mean-field games (adapted from a paper presented at AAMAS 2024). Chapter 7 extends equilibrium learning to common noise (adapted from a paper presented at CDC 2025) and closes with an outlook on continuous state-action spaces led by collaborators.

Finally, Chapter 8 concludes and outlines two future directions: coverage guarantees for continuous, dynamic environments, and mean-field games in continuous domains with obstacles and mixed local and global objectives.

Part I

Networked Systems: Estimation and Coverage under Exogenous Uncertainty

In the networked regime, agents are individually identifiable, and the uncertainty is exogenous*: it is generated outside the team's decisions. In Chapters 2–4, the unknown input is an unmeasured signal entering the system dynamics, such as wind force, collision force, an uncommanded actuator input caused by a fault, or excess river inflow during flooding. Chapters 4 and 5 also study sensor placement over spatial importance densities generated outside the team; these densities are not unknown inputs in the dynamical-system sense. The agents' actions affect what they observe, but they do not affect the disturbance itself. Chapter 2 studies joint input and state estimation for a single agent, and Chapter 3 extends it to sensor networks in which raw data and models cannot be shared. Chapter 4 uses tree structure to restore a reduced estimation problem when the input-rank condition fails, and concludes with a limited simulation illustrating how the resulting source belief can guide mobile sensors. Chapter 5 analyzes how grid refinement affects curvature-based coverage guarantees.*

CHAPTER 2

Joint Input and State Estimation with Prior Knowledge

This chapter is adapted, with corrections, from [Wu, Zheng, and Mehta, "Joint State and Input Estimation of Agent Based on Recursive Kalman Filter Given Prior Knowledge," ICRA 2022] [125]. The version presented here corrects parts of the published analysis.

2.1 Overview

Autonomous agents deployed on long-duration monitoring missions rarely operate under the idealized conditions assumed by their onboard models. Consider an inspection blimp patrolling a site, as sketched in Fig. 2.1: over the course of a mission it may be pushed by a gust of wind, collide with another vehicle, suffer an actuator failure, or slowly lose lift through an air leak. These events differ physically, but from the standpoint of the vehicle's dynamics they share a common structure: each one injects a signal into the state evolution that is neither commanded by the controller nor measured by any onboard sensor. Throughout this chapter we refer to such a signal as an *unknown input*. This chapter studies how a single agent can estimate an unknown input together with its state. The decentralized filter in the next chapter uses this estimator.

Unknown-input estimation has received sustained attention over the past decades, with applications in geophysical event detection such as earthquake input reconstruction [112], environment monitoring [116, 65], sensor fault detection and diagnosis [81], intention inference for self-driving vehicles [128], and maneuvering-target tracking [19, 62, 74]. A tempting

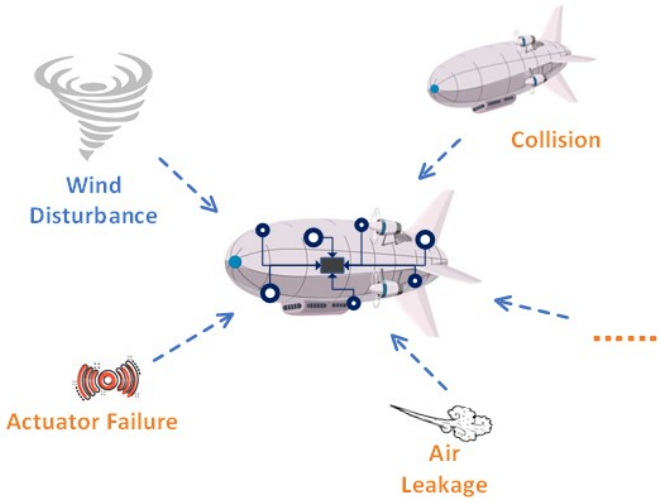


Figure 2.1: An inspection blimp may encounter many unexpected events during a mission: wind disturbances, collisions with other vehicles, actuator failures, air leakage, and more. Regardless of whether the cause is a natural disturbance or an internal fault, each of these events can be modeled as an unknown input acting on the vehicle's dynamics.

shortcut is to lump the unknown input into the process noise and run a standard Kalman filter; the filter remains implementable, but the resulting state estimates carry a large error variance and, when the input is persistent, a systematic bias. The alternative extremes are equally unsatisfying. Augmenting the input into the state vector [79] requires an input dynamics model and inflates the computational cost as the state dimension grows, which motivated two-stage Kalman filters that decouple the state and input sub-filters [48]. When the input dynamics are genuinely unknown, the input must instead be treated as an unexpected disturbance. Kitanidis [65] and Darouach and Zasadzinski [27] first derived optimal state filters that require no prior information about the input; Hsieh [48] unified the two-stage filter with the optimal filter of [65] while also producing input estimates. Gillijns and De Moor [36] then formulated the joint problem and proved that their recursive filter, referred to in this dissertation as the recursive Kalman filter (RKF), delivers minimum-variance unbiased (MVU) estimates of both input and state whenever a rank condition holds. Subsequent

work extended the theory to systems with direct feedthrough [28, 37, 49], relaxed the rank conditions [50, 127, 68, 53], and addressed multi-step-delayed and online variants [51, 52].

All of this prior work treats the input as *completely* unknown. This preserves unbiasedness but can result in a large estimation variance. In practice, however, an agent often possesses partial prior knowledge about the inputs it may encounter. A collision either happened or it did not; earthquake intensities are reported on a discrete scale; the wind speed over a survey area is bounded by the day's weather report. Such knowledge restricts the set of admissible inputs and should reduce the variance of an estimator that uses it. A separate line of work studies Kalman filtering under constraints: projection and truncation methods for state equality and inequality constraints are well established [109, 108]. Constraints on the *unknown input* of an input and state filter are much less explored. The closest prior work is [67], which considers norm constraints, and the switched-system literature [128] estimates a discrete mode jointly with input and state. Finite-mode and interval priors for unknown inputs, including the bias and variance introduced when they are enforced, remained untreated.

This chapter closes that gap. Its contributions are the following.

- For inputs restricted to a finite set of discrete modes, we develop an adaptive multiple-model Kalman filter (AMM-KF) that maintains soft decisions through recursively updated Bayesian mode weights. We prove that the weights converge to the true mode under an explicit Kullback-Leibler identifiability condition, so the input estimator is consistent, and we quantify the reliability of single-step decisions by the Mahalanobis separation of the modes. A weight-flooring safeguard handles mode switches. This discrete-prior estimator is the emphasis of the chapter.

- For inputs known to lie in a continuous interval, we retain the augmented-Lagrangian Kalman filter (AL-RKF) name used in the published paper. Under the stated rank conditions, the gain re-optimization solved there reduces exactly to clipping the un-

constrained estimate, so the iterative augmented-Lagrangian solver is unnecessary. An active constraint reduces variance but can introduce censoring bias and make the estimate non-Gaussian. We therefore present AL-RKF as a bias–variance tradeoff, not as an unbiased or MVU estimator.

- The chapter uses the unconstrained MVU recursive filter of Gillijns and De Moor [36], which we re-derive for completeness as an equality-constrained gain-selection program; no originality is claimed for that recap. Its input estimate and covariance are exactly the per-agent quantities fused by the decentralized filter of Chapter 3.

- In simulation, AMM-KF detects a collision-induced actuator failure reliably when the modes are sufficiently separated, and a dedicated study ties the convergence of the mode weights to the single-step decision probability. In the scalar wind experiment, AL-RKF reduces the input-estimation variance by 81% relative to the Kalman filter and by 47% relative to the RKF, while introducing a small bias when the true input lies on the constraint boundary.

The remainder of the chapter is organized as follows. Section 2.2 states the system model and introduces the two-step input-then-state recursion. Section 2.3 recaps the unconstrained MVU input and state estimator. Section 2.4 develops the discrete-prior estimator and its consistency guarantee, and Section 2.5 treats the interval prior briefly. Section 2.6 reports simulation results, and Section 2.7 concludes and points ahead.

2.2 Problem Formulation

2.2.1 System description

We consider a linear time-varying system driven by an unknown input,

$$x_k = A_{k-1}x_{k-1} + G_{k-1}d_{k-1} + w_{k-1}, \tag{2.1}$$

$$y_k = H_k x_k + v_k, \tag{2.2}$$

where $x_k \in \mathbb{R}^n$ is the state, $d_k \in \mathbb{R}^m$ is the input, and $y_k \in \mathbb{R}^p$ is the measurement. The process noise w_k and measurement noise v_k are mutually uncorrelated, zero-mean, white Gaussian sequences with covariance matrices Q_k and R_k, respectively.

In the classical Kalman filter, the input d_k is a commanded action and hence exactly known. In the scenarios that motivate this chapter, the input is instead actuated by an unknown driver such as wind, a collision, or a failing actuator, and is best regarded as an uncontrolled action. As discussed above, absorbing this uncontrolled input into the process noise keeps the filter runnable but inflates the estimation variance, while estimating it with no prior information at all [36] is unbiased but still yields a large variance. We consider two practically common forms of prior knowledge about the input:

- *Mode prior*: $d_{k-1} \in \{N_1, N_2, \ldots, N_{n_d}\}$, a finite set of discrete modes; for example, an actuator is either healthy or failed.
- *Interval prior*: $d_{k-1} \in \mathcal{D} = [N_1, N_2]$ componentwise; for example, a wind speed bounded by physical limits.

Both the state x_k and the input d_{k-1} are unknown, and the agent seeks to estimate them jointly, given whichever form of prior knowledge is available.

2.2.2 A two-step input-then-state recursion

Estimation proceeds from the factorization of the joint posterior of state and input,

$$p\left(x_k, d_{k-1} \mid y_{1:k}\right) = p\left(d_{k-1} \mid y_{1:k}\right)\, p\left(x_k \mid d_{k-1}, y_{1:k}\right), \tag{2.3}$$

which suggests the classical two-step recursion of [36]: first estimate the input from the innovation, then update the state conditioned on the input estimate. The current observation contains both the unknown input and the state-prediction error, so a conventional innovation

does not separate them. We therefore estimate the input first and then update the state. Under the rank condition stated below, the unconstrained RKF input estimate is unbiased for any value of the true input. This separation limits the direct effect of input-model mismatch on the state estimate. All prior knowledge about the input will enter through the first step alone. Concretely, the filter first predicts

$$\begin{aligned} \hat{x}_{k|k-1} &= A_{k-1}\,\hat{x}_{k-1|k-1}, \\ \hat{P}_{k|k-1} &= A_{k-1}\,P_{k-1|k-1}\,A_{k-1}^{T} + Q_{k-1}, \end{aligned} \tag{2.4}$$

and then estimates the input from the innovation,

$$\hat{d}_{k-1} = M_k\left(y_k - H_k\,\hat{x}_{k|k-1}\right), \tag{2.5}$$

where the hat denotes an estimated value and $M_k \in \mathbb{R}^{m\times p}$ is the input estimation gain. The state is then corrected by a Kalman-filter update built on the MVU principle:

$$\hat{x}^{\star}_{k|k} = \hat{x}_{k|k-1} + G_{k-1}\,\hat{d}_{k-1}, \tag{2.6}$$

$$\hat{x}_{k|k} = \hat{x}^{\star}_{k|k} + K_k\left(y_k - H_k\,\hat{x}^{\star}_{k|k}\right), \tag{2.7}$$

where $K_k \in \mathbb{R}^{n\times p}$ is the state estimation gain matrix.

Equations (2.4)–(2.7) constitute the complete recursive format of joint input and state estimation, and Section 2.3 recaps its unconstrained analysis. The contribution of this chapter lies in the input estimation step: Sections 2.4 and 2.5 show how the two forms of prior knowledge refine the input estimate, which statistical properties survive the refinement, and which do not. In particular, an active interval constraint generally introduces bias and makes the estimate non-Gaussian in exchange for a variance reduction; we quantify this tradeoff rather than assume it away.

2.3 Recap: The Minimum-Variance Unbiased Filter

This section works out the input-then-state recursion of Section 2.2.2 when *no* restriction is placed on the input, recovering the RKF of Gillijns and De Moor [36]; no originality is

claimed here. The recap serves two purposes. First, the unconstrained MVU input estimate and its covariance are the raw material that both prior-aware estimators of this chapter refine. Second, the same gain and covariance expressions are the per-agent building blocks of the decentralized filter in Chapter 3.

2.3.1 Unbiased input estimation

Substituting the process model (2.1) into the measurement model (2.2) expresses the current observation in terms of the previous state and the unknown input,

$$y_k = H_k \left(A_{k-1} x_{k-1} + G_{k-1} d_{k-1} + w_{k-1} \right) + v_k. \tag{2.8}$$

Substituting (2.8) into (2.5) yields

$$\hat{d}_{k-1} = M_k H_k G_{k-1}\, d_{k-1} + M_k H_k A_{k-1} \big(x_{k-1} - \hat{x}_{k-1|k-1} \big) + M_k (H_k w_{k-1} + v_k)\,, \tag{2.9}$$

where $\hat{d}_{k-1}$ is the estimate and d_{k-1} without a hat denotes the ground truth. Suppose the previous state estimate $\hat{x}_{k-1|k-1}$ is unbiased and recall that w_{k-1} and v_k are zero-mean. Taking the expectation of (2.9), the input estimate is unbiased for *any* input sequence, with no assumption on the evolution of d_k, if and only if the gain satisfies

$$M_k H_k G_{k-1} = I_m. \tag{2.10}$$

A gain satisfying (2.10) exists under the rank condition identified in [36], which we adopt throughout this chapter.

Assumption 1 (Rank condition)**.** For every timestep k,

$$\operatorname{rank} \left(H_k G_{k-1} \right) = \operatorname{rank} \left(G_{k-1} \right) = m. \tag{2.11}$$

Remark 1. Assumption 1 requires, in particular, that the measurement map does not annihilate any input direction. When the rank condition fails, the system matrices must be decomposed and reformed to make the problem solvable, as pursued for instance in [50, 127, 68, 53]; those extensions are orthogonal to the use of prior knowledge studied here.

2.3.2 Minimum-variance input gain

Among all unbiased gains, we seek the one of minimum variance. Define the innovation

$$\tilde{y}_k = y_k - H_k\,\hat{x}_{k|k-1} = H_k G_{k-1}\,d_{k-1} + e_k, \tag{2.12}$$

where the innovation noise e_k collects the state-prediction error and the measurement noise,

$$e_k = H_k\left(A_{k-1}\left(x_{k-1} - \hat{x}_{k-1|k-1}\right) + w_{k-1}\right) + v_k, \tag{2.13}$$

a zero-mean Gaussian variable with covariance

$$\tilde{R}_k = \mathrm{Cov}\,(e_k) = H_k\,\hat{P}_{k|k-1}H_k^T + R_k. \tag{2.14}$$

Throughout the rest of the chapter we abbreviate $F_k = H_k G_{k-1}$. By (2.5), the input estimate is a linear function of the innovation,

$$\hat{d}_{k-1} = M_k\,\tilde{y}_k = M_k F_k\,d_{k-1} + M_k\,e_k, \tag{2.15}$$

and under the unbiasedness condition (2.10), (2.15) reduces to

$$\hat{d}_{k-1} = d_{k-1} + M_k\,e_k. \tag{2.16}$$

Writing $\tilde{d}_{k-1} = d_{k-1} - \hat{d}_{k-1}$ for the input estimation error, the covariance of the input estimate is therefore

$$\mathrm{Cov}\big(\hat{d}_{k-1}\big) = \mathbb{E}\big[\tilde{d}_{k-1}\tilde{d}_{k-1}^T\big] = \mathbb{E}\big[(M_k e_k)(M_k e_k)^T\big] = M_k\tilde{R}_k M_k^T. \tag{2.17}$$

The search for the MVU input estimate has thus been transferred into a search for the gain M_k that minimizes the variance (2.17) subject to the unbiasedness condition (2.10), an equality-constrained optimization problem:

$$\begin{aligned} \min_{M_k} \quad & \mathrm{Tr}\left(M_k\tilde{R}_k M_k^T\right) \\ \text{s.t.} \quad & M_k H_k G_{k-1} = I_m. \end{aligned} \tag{2.18}$$

Proposition 1 (MVU input gain [36])**.** *Under Assumption 1, the unique solution of* (2.18) *is*

$$M_k = \left[(H_k G_{k-1})^T \tilde{R}_k^{-1} (H_k G_{k-1}) \right]^{-1} (H_k G_{k-1})^T \tilde{R}_k^{-1}. \tag{2.19}$$

Proof. Introduce a multiplier matrix $\Lambda \in \mathbb{R}^{m \times m}$ and form the Lagrangian

$$\mathcal{L}(M_k, \Lambda) = \mathrm{Tr}\left(M_k \tilde{R}_k M_k^T \right) + \mathrm{Tr}\left(\Lambda^T (I_m - M_k H_k G_{k-1}) \right).$$

Stationarity with respect to M_k gives $2 M_k \tilde{R}_k = \Lambda (H_k G_{k-1})^T$, i.e., $M_k = \frac{1}{2} \Lambda (H_k G_{k-1})^T \tilde{R}_k^{-1}$. Enforcing the constraint $M_k H_k G_{k-1} = I_m$ determines $\Lambda = 2 \left[(H_k G_{k-1})^T \tilde{R}_k^{-1} (H_k G_{k-1}) \right]^{-1}$, which exists by Assumption 1 and $\tilde{R}_k \succ 0$. Substituting back yields (2.19); since the objective is strictly convex in M_k on the affine constraint set, the stationary point is the global minimizer. □

Substituting (2.19) into (2.17), the covariance of the resulting MVU input estimate is

$$S_k = \mathrm{Cov}\big(\hat{d}_{k-1}\big) = \left[(H_k G_{k-1})^T \left(H_k \hat{P}_{k|k-1} H_k^T + R_k \right)^{-1} (H_k G_{k-1}) \right]^{-1}. \tag{2.20}$$

Two properties of this estimate are worth recording. First, the gain (2.19) is a *fixed*, data-independent linear map of the innovation; this is precisely what makes (2.17) its sampling covariance, a fact that will matter in Section 2.5. Second, the pair $(\hat{d}_{k-1}, S_k)$ produced by (2.5) and (2.20) is exactly the per-agent input message that the decentralized filter of Chapter 3 exchanges and fuses across a sensor network.

2.3.3 Unbiased minimum-variance state update

The state update introduces no new optimization: it feeds the input estimate through (2.6)–(2.7). Define the estimation errors $\tilde{x}_{k-1} = x_{k-1} - \hat{x}_{k-1|k-1}$ and $\tilde{x}_k^\star = x_k - \hat{x}_{k|k}^\star$. Combining (2.1) with (2.6),

$$\tilde{x}_k^\star = A_{k-1} \tilde{x}_{k-1} + G_{k-1} \tilde{d}_{k-1} + w_{k-1}. \tag{2.21}$$

Suppose $\hat{x}_{k-1|k-1}$ is unbiased (unbiased initialization without a priori input information is discussed in [66]) and recall that $\hat{d}_{k-1}$ is unbiased whenever (2.10) holds. Then every term on the right side of (2.21) has zero mean, so

$$\mathbb{E}\big[\tilde{x}_k^\star\big] = 0, \tag{2.22}$$

and consequently the expectation of the final update $\hat{x}_{k|k}$ in (2.7) equals x_k: the state estimate is unbiased.

For the variance, combining (2.4)–(2.7) with the innovation (2.12) shows that the two-stage update acts on the innovation through a single composite gain:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\,\tilde{y}_k + (I_n - K_k H_k)\,G_{k-1} M_k\,\tilde{y}_k \\ &= \hat{x}_{k|k-1} + L_k\,\tilde{y}_k, \qquad L_k = K_k + (I_n - K_k H_k)\,G_{k-1} M_k. \end{aligned} \tag{2.23}$$

Gillijns and De Moor [36] showed that the variance-minimizing choice of K_k is itself a function of the input gain M_k and varies with the ranks of the involved matrices; the associated covariance recursions follow [36] and are not repeated here. What matters for this chapter is the division of labor made explicit by (2.23): all prior knowledge about the input enters through the input estimate alone, while the state-update machinery remains untouched. The next two sections refine the input estimate under discrete and continuous priors, respectively; the state update is identical to the one above and is not repeated.

2.4 Input Estimation with a Discrete Prior: AMM-KF

We first suppose that the input takes values in a finite set of n_d discrete modes $\{N_1, \ldots, N_{n_d}\}$: a collision either occurred or it did not; an earthquake registers at one of several discrete intensity levels. A hard projection of the unconstrained estimate onto the discrete set would discard all mode uncertainty and commit prematurely at every timestep. Instead we maintain a posterior weight over the modes with a bank of mode-conditioned filters, in the spirit of static multiple-model estimation [84, 101], an approach that is also standard in maneuvering-

target tracking and driving-intention inference [128]. We call the resulting input estimator the adaptive multiple-model Kalman filter (AMM-KF).

2.4.1 Bayesian mode weights and the moment-matched update

Conditioned on mode i, that is $d_{k-1} = N_i$, the innovation (2.12) is Gaussian with mean $F_k N_i$ and covariance $\tilde{R}_k$. The likelihood of each mode is therefore

$$L_k^{(i)} = \mathcal{N}\big(\tilde{y}_k;\ F_k N_i,\ \tilde{R}_k\big), \tag{2.24}$$

and the mode weights follow the Bayesian recursion

$$\mu_k^{(i)} = \frac{\mu_{k-1}^{(i)}\, L_k^{(i)}}{\sum_{j=1}^{n_d} \mu_{k-1}^{(j)}\, L_k^{(j)}}, \qquad i = 1, \dots, n_d, \tag{2.25}$$

initialized uniformly, $\mu_0^{(i)} = 1/n_d$. Two outputs are available from the weights. The *soft* input estimate is the posterior mean over the mode set,

$$\bar{d}_{k-1} = \sum_{i=1}^{n_d} \mu_k^{(i)}\, N_i, \tag{2.26}$$

which is optimal in the MMSE sense given the weights and is the quantity fed to the state update. The *hard* decision is the maximum a posteriori mode,

$$\hat{d}_{k-1} = N_{i^\star}, \qquad i^\star = \arg\max_i\ \mu_k^{(i)}, \tag{2.27}$$

appropriate when a discrete decision must be made, for example whether to recall the agent for repair.

The ensemble state estimate and covariance are obtained by moment matching across the mode-conditioned sub-filters,

$$\hat{x}_{k|k} = A_{k-1}\,\hat{x}_{k-1|k-1} + K_k\left(y_k - H_k A_{k-1}\,\hat{x}_{k-1|k-1}\right) + (I_n - K_k H_k)\, G_{k-1}\,\bar{d}_{k-1}, \tag{2.28}$$

$$P_{k|k} = \sum_{i=1}^{n_d} \left[P_{k|k}^{(i)} + e_k^{(i)}\, e_k^{(i)\,T}\right] \mu_k^{(i)}, \qquad e_k^{(i)} = \hat{x}_{k|k} - \hat{x}_{k|k}^{(i)}, \tag{2.29}$$

where $P_{k|k}^{(i)}$ and $\hat{x}_{k|k}^{(i)}$ are the covariance and state estimate of the sub-filter conditioned on mode i. The first two terms of (2.28) are the input-blind Kalman update; the third compensates it with the soft input estimate. Equations (2.28) and (2.29) are the standard first-order (GPB1-style) collapse of the mode-conditioned mixture [101]. The collapse approximates the exact posterior, which is a Gaussian mixture whose number of components grows exponentially in time; AMM-KF is a principled approximation, and we do not claim it is an optimal filter. The complete procedure is summarized in Algorithm 1; the weight floor in its fourth step is explained in Section 2.4.3.

Algorithm 1 AMM-KF: input estimation with a discrete multi-mode prior

Input: System matrices A_{k-1}, G_{k-1}, H_k, Q_{k-1}, and R_k

Previous estimates $\hat{x}_{k-1|k-1}$ and $P_{k-1|k-1}$; observation y_k

Prior input modes $\{N_1, \ldots, N_{n_d}\}$; weight floor ϵ_μ

Output: Mode weights μ_k; soft input $\bar{d}_{k-1}$; hard decision $\hat{d}_{k-1}$

State estimate $\hat{x}_{k|k}$ and covariance $P_{k|k}$

1: Initialize $\mu_0^{(i)} \leftarrow 1/n_d$ for each mode i

2: Compute $\tilde{y}_k$ using Eq. (2.12) and $L_k^{(i)}$ using Eq. (2.24)

3: Update $\mu_k^{(i)}$ using Eq. (2.25)

4: Set $\mu_k^{(i)} \leftarrow \max\Big(\mu_k^{(i)}, \epsilon_\mu\Big)$ and renormalize μ_k

5: Compute $\bar{d}_{k-1}$ using Eq. (2.26) and $\hat{d}_{k-1}$ using Eq. (2.27)

6: Update $\hat{x}_{k|k}$ and $P_{k|k}$ using Eqs. (2.28) and (2.29)

2.4.2 Consistency and identifiability

The recursion (2.25) assumes a fixed true mode. Under that assumption its convergence can be stated precisely.

Proposition 2 (Consistency under identifiability)**.** *Suppose the true mode $i^\star$ is fixed over time with $\mu_0^{(i^\star)} > 0$, the mode-conditioned innovation densities $p_i = \mathcal{N}(F_k N_i, \tilde{R}_k)$ are correctly*

specified with innovations independent over time (for example, a time-invariant system in filter steady state), and the modes are uniformly identifiable: $\inf_k D_{\mathrm{KL}}\left(p_{i^\star} \,\|\, p_i\right) > 0$ *for all* $i \neq i^\star$. *Then* $\mu_k^{(i^\star)} \to 1$ *almost surely, and consequently the soft estimate* (2.26) *converges to* $N_{i^\star}$*: the input estimator is consistent, and hence asymptotically unbiased.*

Proof. From (2.25), $\log\big(\mu_k^{(i)}/\mu_k^{(i^\star)}\big) = \log\big(\mu_0^{(i)}/\mu_0^{(i^\star)}\big) + \sum_{t\le k} \log\big(L_t^{(i)}/L_t^{(i^\star)}\big)$. Under $p_{i^\star}$, each increment has mean $-D_{\mathrm{KL}}(p_{i^\star}\|p_i)$, uniformly negative by assumption, so by the strong law of large numbers for independent sequences the sum diverges to $-\infty$ for every $i \neq i^\star$. □

The proof is worth reading as a statement about evidence rather than about filtering. Each observation scores every mode by its likelihood, and the log-odds of a wrong mode against the true one is a random walk whose per-step drift equals the negative KL divergence between the two innovation densities. The wrong mode's log-odds therefore drifts downward linearly in time, and the posterior mass concentrates on the true mode. Consistency is accumulated evidence from a sequential hypothesis test; no special property of the Kalman gain is involved.

Proposition 2 also delimits what is claimed. At any finite time the soft estimate (2.26) is a convex combination of the modes and is therefore biased toward the interior of the mode set whenever wrong modes retain nonzero weight; unbiasedness holds only asymptotically. The convergence speed is governed by the same separation that appears in the identifiability condition. For two modes with equal covariance, the per-step expected log-likelihood-ratio drift equals $\Delta_k^2/2$, and the error probability of the optimal single-step test is

$$P_e = \Phi\left(-\frac{\Delta_k}{2}\right), \qquad \Delta_k^2 = \left(N_1 - N_2\right)^T F_k^T\, \tilde{R}_k^{-1} F_k \left(N_1 - N_2\right), \tag{2.30}$$

where Φ denotes the standard normal CDF and Δ_k is the Mahalanobis separation of the mode-conditioned innovation means. When Δ_k is small, that is, when the candidate inputs are not separated by more than the noise, single-step decisions are unreliable and the weight recursion converges slowly. Section 2.6.3 examines this dependence experimentally.

2.4.3 Mode switches and safeguards

If the true mode changes during the run, as in the collision experiment of Section 2.6.2, the fixed-mode premise of Proposition 2 is violated: the weights of unlikely modes decay exponentially, and after a switch the recursion must first overcome the accumulated log-odds, giving a response delay that grows with the time spent in the previous mode. We therefore floor the weights at a small $\epsilon_\mu > 0$ with renormalization (Algorithm 1), which bounds the accumulated deficit and hence the delay. The principled extension is a Markov mode-transition model as in the interacting multiple model (IMM) filter [9], which we do not pursue here; and the achievable detection delay is fundamentally limited by the same per-step drift $\Delta_k^2/2$ that governs sequential change detection [94]. After a switch the transient estimate is necessarily biased for any causal estimator, so the appropriate transient metric is the detection delay rather than unbiasedness.

2.5 Input Estimation with a Continuous Prior: AL-RKF

We now suppose the input is known to lie in a bounded interval, $d_{k-1} \in \mathcal{D} = [N_1, N_2]$ componentwise. In maneuver tracking the motion space of the input is limited to a finite range; in the airship scenario of Section 2.6, the wind speed is bounded by prior meteorological knowledge. This section is deliberately brief: the interval prior reduces to a projection whose update takes two lines, and its statistical characterization is short.

The published AL-RKF formulation constrains the gain so that the realized estimate lands in the interval and solves the resulting problem by augmented-Lagrangian iterations:

$$\begin{aligned} \min_{M} \quad & \operatorname{Tr}\left(M \tilde{R}_k M^T\right) \\ \text{s.t.} \quad & M F_k = I_m, \qquad N_1 \le M\,\tilde{y}_k \le N_2. \end{aligned} \tag{2.31}$$

Solving (2.31) iteratively is unnecessary: it admits a closed-form reduction.

Proposition 3 (Reduction to clipping)**.** *Let $\hat{d}_{k-1}^{0} = M_k \tilde{y}_k$ denote the unconstrained MVU*

estimate with M_k the gain of (2.19) *and S_k its covariance* (2.20)*. Assume $[F_k \;\; \tilde{y}_k]$ has full column rank (which requires $p \geq m+1$) and let $s_k = \tilde{y}_k^T \tilde{R}_k^{-1} \tilde{y}_k - \left(\hat{d}_{k-1}^0\right)^T S_k^{-1} \hat{d}_{k-1}^0$, which is then positive. For any fixed $u \in \mathbb{R}^m$,*

$$\min_{\substack{MF_k = I_m \\ M\tilde{y}_k = u}} \operatorname{Tr}\left(M \tilde{R}_k M^T\right) = \operatorname{Tr}\left(S_k\right) + \frac{\left\|u - \hat{d}_{k-1}^0\right\|_2^2}{s_k}. \tag{2.32}$$

Consequently, the estimate produced by the exact solution of (2.31) *is the Euclidean projection of $\hat{d}_{k-1}^0$ onto $\mathcal{D}$, i.e., the componentwise clip* $\operatorname{clip}\left(\hat{d}_{k-1}^0, N_1, N_2\right)$.

Proof. Stack the constraints as $M[F_k \;\; \tilde{y}_k] = [I_m \;\; u]$. The minimizer of $\operatorname{Tr}(M\tilde{R}_k M^T)$ under a linear constraint $MA = B$ is $M^\star = B(A^T \tilde{R}_k^{-1} A)^{-1} A^T \tilde{R}_k^{-1}$, with optimal value $\operatorname{Tr}\left(B(A^T \tilde{R}_k^{-1} A)^{-1} B^T\right)$. Block inversion of $A^T \tilde{R}_k^{-1} A$ with Schur complement s_k gives (2.32). Since $\operatorname{Tr}(S_k)$ and s_k do not depend on u, minimizing over $u \in \mathcal{D}$ minimizes the Euclidean distance $\|u - \hat{d}_{k-1}^0\|_2$, whose minimizer over an axis-aligned box is the componentwise clip. □

Proposition 3 carries two lessons. First, the $m \times p$ gain-matrix optimization is redundant: only the m-dimensional estimate $M_k \tilde{y}_k$ enters the filter, and the remaining degrees of freedom of the gain lie in the null space of $\tilde{y}_k$. Second, the trace objective weights all input components equally, so the induced projection is Euclidean and cannot exploit the correlation structure of the input error. Following the terminology of the published paper, we retain the name AL-RKF for this interval-constrained estimator. Under the rank conditions of Proposition 3, its exact output can be computed directly as

$$\hat{d}_{k-1}^c = \arg\min_{u \in \mathcal{D}} \left\|u - \hat{d}_{k-1}^0\right\|_2^2 = \operatorname{clip}\left(\hat{d}_{k-1}^0, N_1, N_2\right). \tag{2.33}$$

This is a projection approach to constrained Kalman filtering [109, 108] applied to the input rather than to the state. For a scalar input, as in the wind experiment below, the clip also coincides with the covariance-weighted constrained maximum-likelihood projection. For correlated multivariate input errors the two projections generally differ; that extension is not studied or claimed here.

What does the projection cost? On a realization for which the constraint is inactive, $\hat{d}^{c}_{k-1} = \hat{d}^{0}_{k-1}$ numerically. Once clipping has positive probability across repeated realizations, however, the estimator is a nonlinear function of the innovation and the RKF's global unbiasedness, Gaussianity, and MVU guarantees do not carry over. For a scalar input its exact distribution is known. Let Φ and ϕ denote the standard normal CDF and PDF.

Proposition 4 (Censored-normal characterization)**.** *Let $m = 1$, $\hat{d}^{0} \sim \mathcal{N}(d, \sigma_d^2)$ with $\sigma_d^2 = S_k$, and $\hat{d}^{c} = \mathrm{clip}(\hat{d}^{0}, N_1, N_2)$. With $\alpha = (N_1 - d)/\sigma_d$ and $\beta = (N_2 - d)/\sigma_d$,*

$$\mathbb{E}\big[\hat{d}^{c}\big] = N_1\Phi(\alpha) + N_2\big(1 - \Phi(\beta)\big) + d\big(\Phi(\beta) - \Phi(\alpha)\big) + \sigma_d\big(\phi(\alpha) - \phi(\beta)\big). \tag{2.34}$$

Consequently: (i) the bias is negligible whenever the constraint is essentially inactive ($\Phi(\alpha) \approx 0$, $\Phi(\beta) \approx 1$); in general the bias vanishes only under symmetric censoring, and when d lies near a boundary of $\mathcal{D}$ the bias is $O(\sigma_d)$ and directed toward the interior. (ii) The distribution of $\hat{d}^{c}$ is a censored normal with point masses $\Phi(\alpha)$ and $1 - \Phi(\beta)$ at the boundaries; in particular it is not Gaussian when the constraint is active. (iii) $\mathrm{Var}\big(\hat{d}^{c}\big) \le \sigma_d^2$*, with strict inequality whenever the constraint is active with positive probability.*

Property (ii) corrects a claim made in the published analysis, namely that the constrained estimate remains Gaussian because the constraint only affects the gain: once the gain depends on the realized innovation, the estimate is no longer a fixed linear map of a Gaussian variable, and (2.17) no longer describes its sampling variance. Property (iii) explains the variance reductions reported in Section 2.6.4: they arise from censoring and the resulting bias–variance tradeoff, not from a new unbiased or MVU estimator. The tradeoff is nevertheless favorable whenever the prior is correct, in a pointwise error sense.

Proposition 5 (Pointwise error contraction)**.** *If the prior is correct, $d_{k-1} \in \mathcal{D}$, then on every realization $\big\|\hat{d}^{c}_{k-1} - d_{k-1}\big\|_2 \le \big\|\hat{d}^{0}_{k-1} - d_{k-1}\big\|_2$; the contraction also holds componentwise in absolute value. Hence the ordinary mean squared error of the exact AL-RKF projection never exceeds that of the unconstrained RKF when the interval prior is correct. If instead $d_{k-1} \notin \mathcal{D}$, the error of $\hat{d}^{c}_{k-1}$ grows with the distance from d_{k-1} to $\mathcal{D}$ and the guarantee is lost.*

Proof. Projection onto a convex set is non-expansive in the projection metric; taking one argument to be $d_{k-1} \in \mathcal{D}$, a fixed point of the projection, gives the claim. □

Remark 2 (Equality and subspace priors are free). If the prior knowledge is an equality constraint, for example a known disturbance direction $d_{k-1} = V\theta_{k-1}$, it should be embedded by replacing G_{k-1} with $G_{k-1}V$ and estimating θ_{k-1}: this reduces the input dimension m, preserves unbiasedness exactly, and reduces variance, with none of the censoring effects of Proposition 4. In the wind experiment of Section 2.6, the known wind direction is of this type; only the speed interval requires the projection.

Since the state update (2.6)–(2.7) consumes $\hat{d}^{c}_{k-1}$, the state estimate inherits a one-step bias increment bounded by $\|(I_n - K_k H_k)G_{k-1}\|$ times the input bias when the constraint is active, which then propagates through the recursion. On an individual update for which the constraint is inactive, its numerical update is identical to the RKF update.

The interval treated here is the simplest continuous prior, and the projection refines an unconstrained estimate that exists only under the rank condition of Assumption 1. Chapter 4 treats the complementary regime, in which the rank condition fails and the continuous input instead carries structural prior knowledge in the form of a tree.

2.6 Simulation Results

2.6.1 Experiment setup

Two examples, sketched in Fig. 2.2, illustrate the two kinds of prior knowledge treated in this chapter: a discrete mode prior and a continuous interval prior.

Discrete case: collision detection in a swarm. Multiple drones run independently; one drone is hovering when an adjacent drone collides with it at 30 s, and the collision leaves an actuator failed. The collision and the actuator failure are assumed to affect only the horizontal position of the drone. Observations come from noisy self-localization. The goal is

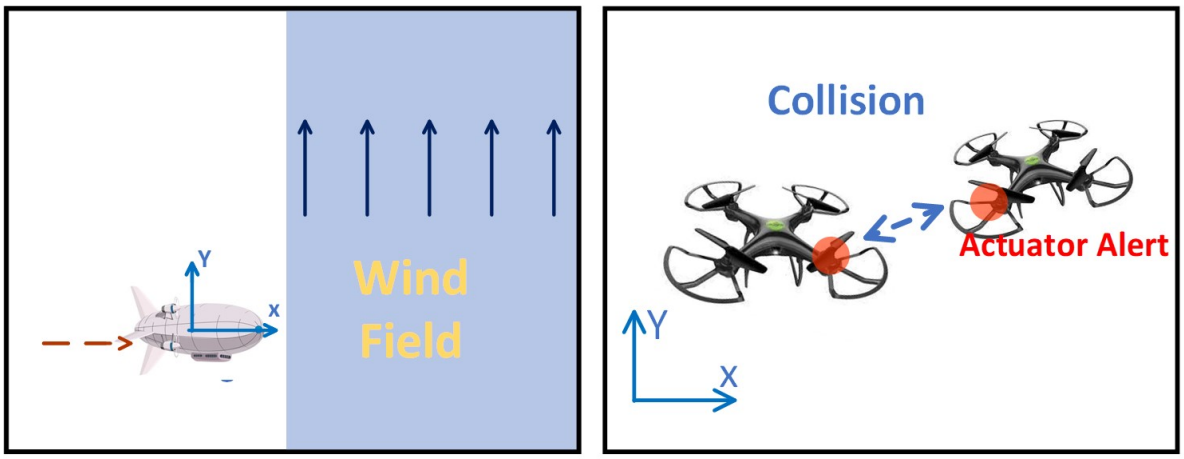


Figure 2.2: Simulation scenarios. (a) An airship flies forward into a wind field; the wind acts as a continuous unknown input bounded by prior knowledge. (b) In a swarm, one drone collides with an adjacent one, and the collision may leave an actuator failed; the collision occurrence is a binary unknown input.

to estimate the probability that the actuator has failed due to a collision; the prior knowledge is that the failure influence is binary, a two-mode prior. Note that this scenario involves a mode switch at 30 s, so by the discussion of Section 2.4.3 the relevant transient measure after the collision is the detection delay of the weight recursion. In a deployed system, this collision evaluation would let an operator decide whether to recall the agent for repair.

Continuous case: airship in a wind field. The task is to estimate the position of an airship and the wind speed simultaneously, without an additional anemometer. The prior knowledge is twofold: the wind direction is along the y axis, a subspace prior embedded through Remark 2, and the wind speed is limited to between 0 m/s and 10.3 m/s, the interval prior handled by the projection (2.33). The mass of the airship, the air density ρ, and the surface area S of the airship exposed to the wind are assumed known. Trajectory observations arrive from satellites, corrupted by noise, and the motion model in the filter is Newton's kinematic equations. The filter estimates the external wind force F directly as the unknown

input; the wind speed is then recovered through the drag relation $F = \rho S v^2$.

2.6.2 Discrete prior: collision detection with AMM-KF

Figure 2.3 shows the estimated probability of the actuator failure caused by the collision at 30 s. The AMM-KF confidence is the dynamic weight μ of (2.25), while the KF baseline makes an independent hypothesis-testing decision at each timestep, whose reliability is governed by the single-step error probability (2.30); in both cases the input decision is made when the probability exceeds 0.5. Through sequential observation the recursive weights accumulate evidence: the AMM-KF confidence rises at the collision after the short detection-delay transient discussed in Section 2.4.3 and then stays pinned to the correct decision, whereas the memoryless per-step estimate fluctuates with large variance throughout. The benefit propagates to the state estimate: because the failure is identified correctly, the compensation term of (2.28) absorbs the bias that the standard KF exhibits in the failed channel (Fig. 2.3b). The AMM-KF is thus most valuable exactly when the actuator has failed.

Convergence is not always this rapid. Figure 2.4 repeats the experiment with large system noise, which shrinks the separation Δ_k of (2.30): the weight now takes visibly longer after the 30 s collision to commit to the correct mode, exactly the slower log-odds drift predicted in Section 2.4.2. In practice the convergence efficiency of the estimator directly affects its usability, since a slowly converging decision process consumes time and resources; the next subsection isolates this dependence.

2.6.3 Identifiability of discrete modes

To isolate the effect of mode separation, a dedicated simulation gives the estimator two Gaussian mode distributions with means $\mathbb{E}(u_0) = 0$ and $\mathbb{E}(u_1) = 2$ and varies their common standard deviation σ_u, so that the separation of (2.30) is $\Delta = 2/\sigma_u$. Figure 2.5 shows the dynamic updating of the weight μ for $\sigma_u \in \{2.8, 1.4, 1.06\}$, together with the corresponding

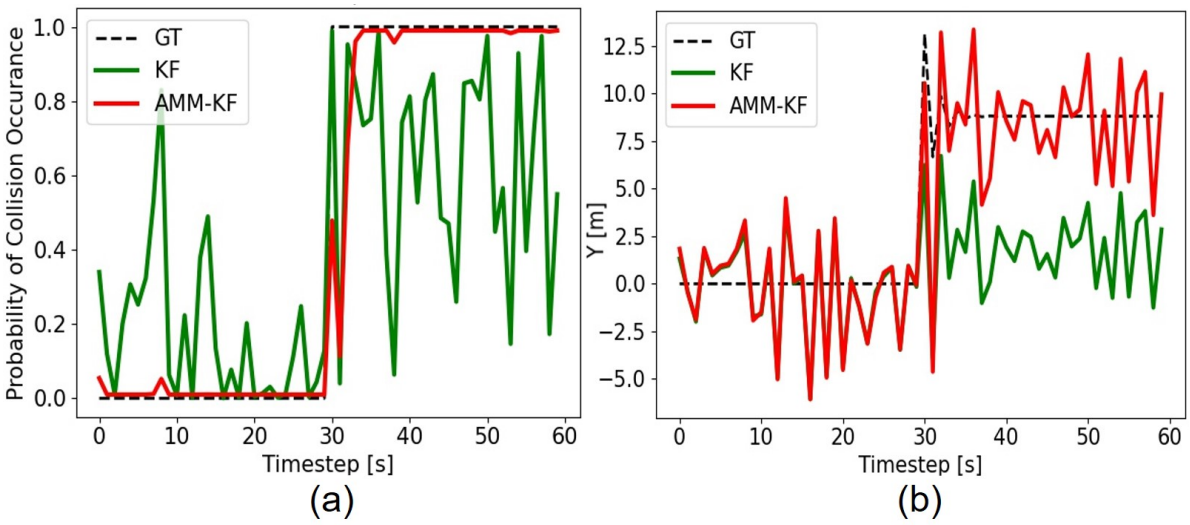


Figure 2.3: The actuator fails due to a collision at 30 s. (a) Confidence that the failure has occurred: the AMM-KF weight converges rapidly to the correct decision and holds it, while the per-step KF hypothesis test fluctuates. (b) Displacement in the y direction: precise failure estimation lets AMM-KF compensate the observation bias that the standard KF suffers.

single-step correct-decision probability $P_D = 1 - P_e = \Phi(\Delta/2)$ (green dash-dotted line): $P_D = 0.64$, 0.763, and 0.828, respectively. In the bottom configuration ($\sigma_u = 1.06$) the weight stays at the right decision for more than 90% of the plotted horizon, while in the top configuration ($\sigma_u = 2.8$) the recursion converges slowly and spends long stretches of the horizon at the wrong decision. The practical reading is direct: when the innovation separation encountered online resembles the well-separated bottom distributions, the estimator can be trusted to decide correctly; when it resembles the overlapping top distributions, single-step decisions are unreliable and the weight recursion commits only after long delays. This is the identifiability condition of Proposition 2 made visible: as the KL divergence between the mode-conditioned densities approaches zero, the modes become indistinguishable and no input estimator can succeed. The separation between candidate modes should therefore be treated as a design requirement rather than an afterthought.

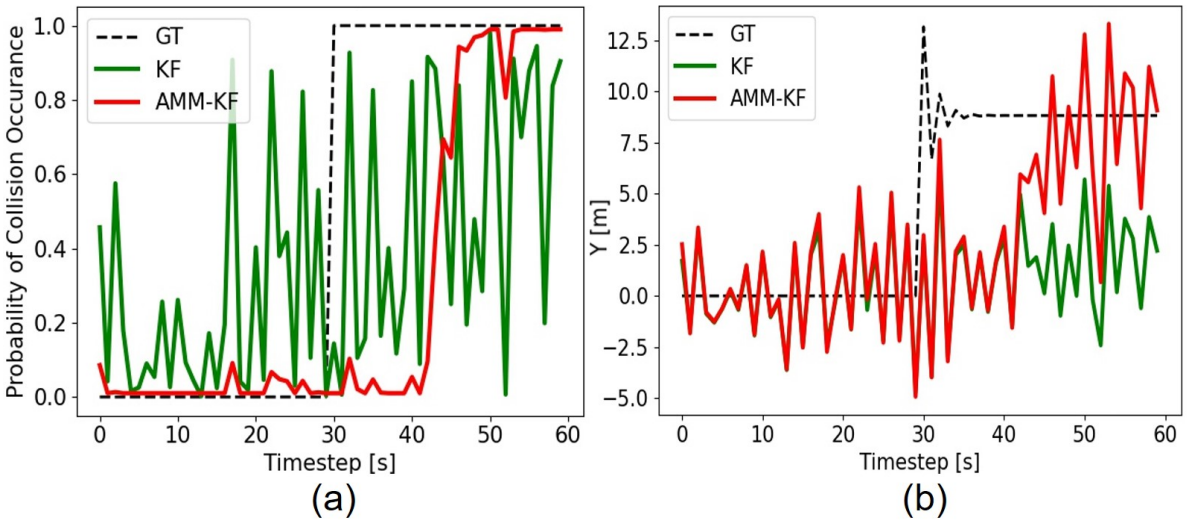


Figure 2.4: With large system noise, the input estimator does not select the correct decision rapidly: (a) the AMM-KF confidence commits to the collision mode only after a noticeable delay; (b) the corresponding displacement estimates. In this run the estimator eventually selects the collision mode, but the detection and re-detection delays grow as the separation (2.30) shrinks.

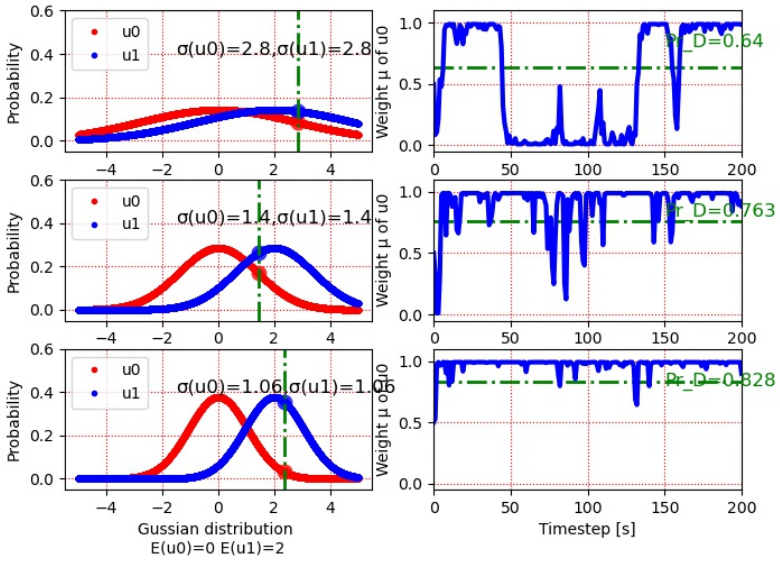


Figure 2.5: Dynamic updating of the weight μ for two Gaussian mode distributions with means 0 and 2 and common standard deviation $\sigma_u = 2.8$, 1.4, and 1.06 (top to bottom), for which the single-step correct-decision probability $1 - P_e$ of (2.30) evaluates to 0.64, 0.763, and 0.828. To avoid decision-making ambiguity, the separation between the mode distributions must be identifiable relative to the noise: only the bottom configuration yields consistently correct decisions.

2.6.4 Continuous prior: wind-field estimation with AL-RKF

Figure 2.6 compares the plain Kalman filter (KF), the RKF of Section 2.3, and AL-RKF on the airship scenario, in which the vehicle enters the wind field at 30 s, corresponding to the location (30 m, 0 m). Both RKF and AL-RKF clearly outperform the KF once the external input begins to act, as expected, since the KF has no input model at all: it must explain the wind through its process noise, drifts badly in the wind-speed channel, and lags the true trajectory. The comparison that tests the interval prior is AL-RKF against RKF. The interval constraint suppresses the excursions of the input estimate outside the physically admissible range (visible as the RKF excursions below 0 and above the admissible 10.3 m/s bound in Fig. 2.6a), thereby reducing the estimation variance.

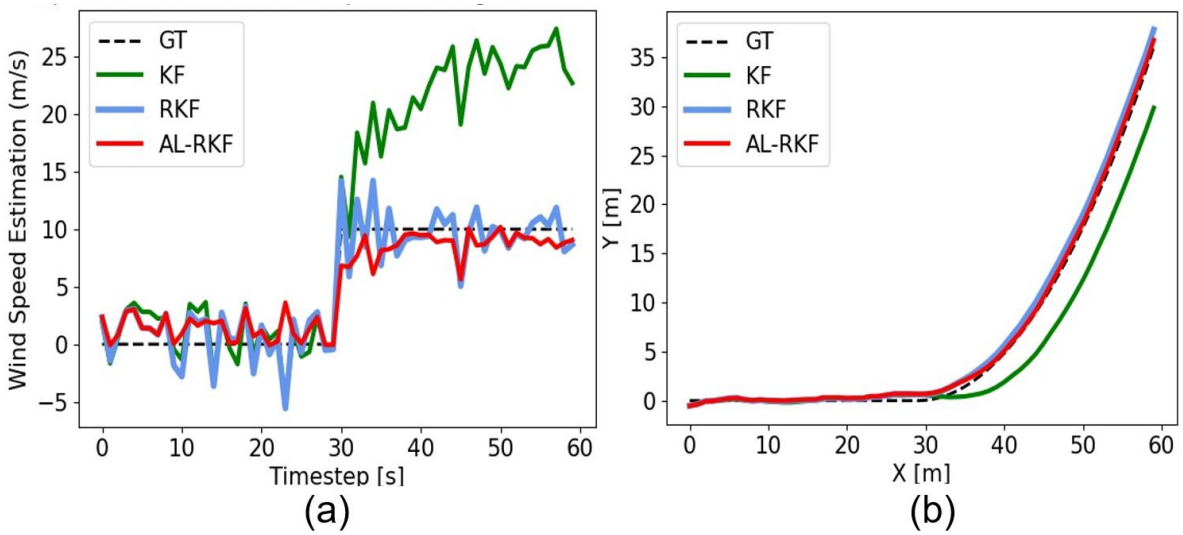


Figure 2.6: The airship enters the wind field at 30 s (location (30 m, 0 m)). (a) The external wind force F is estimated directly by each filter and the wind speed is recovered via $F = \rho S v^2$; the interval prior keeps the constrained estimate inside the admissible range. (b) Position estimates in the plane; the four curves compare the three filters against the ground truth (GT). For this scalar input, the exact AL-RKF constrained solution is the clip of Proposition 3.

To explore the influence of the system noise on the input estimator more thoroughly,

multiple experiments were carried out with different pairs of process and measurement noise covariances Q and R; Table 2.1 reports the mean and variance of the wind-speed estimation error over this sweep. The results are consistent with the bias–variance tradeoff described by Propositions 4 and 5. On the variance side, the quantity the prior is designed to reduce, AL-RKF improves on the KF by 81% and on the RKF by 47% on average in this scalar simulation. On the mean-error side, AL-RKF improves on the KF by 92% but is *worse* than the unbiased RKF in four of the six noise settings (for example 0.08 against 0.06 in the first column, and markedly so in the last, 0.34 against 0.13). This is the censoring bias of Proposition 4: the true wind speed is 0 m/s before the onset at 30 s, exactly on the constraint boundary, so the active constraint censors negative excursions and biases the estimate toward the interior of the interval; during that phase the estimate distribution carries a point mass at the boundary and is not Gaussian. The prior reduces variance, while bias appears when the true input lies on the boundary of the prior.

Table 2.1: Statistical results of the wind-speed (input) estimation error in the airship scenario under different noise levels. The bracketed pairs are the diagonal entries of the process noise covariance Q and the measurement noise covariance R; means are in m/s, variances in m^2/s^2. Bold marks the best filter in each column.

Q		[0.05, 0.05]			[0.05, 0.5]		[0.5, 0.5]
R		$[0.05, 0.05]$	$[0.05, 0.5]$	$[0.5, 0.5]$	$[0.05, 0.5]$	$[0.5, 0.5]$	$[0.5, 0.5]$
mean (m/s)	KF	2.84	2.5	6.4	2.02	6	14.29
	RKF	**0.06**	**0.4**	0.5	0.27	**0.37**	**0.13**
	AL-RKF	0.08	0.45	**0.25**	**0.2**	0.41	0.34
variance (m^2/s^2)	KF	7.7	640	80	17.5	148	111.4
	RKF	0.8	5.0	6.3	14.2	148	69.8
	AL-RKF	**0.3**	**3.2**	**4.1**	**9.0**	**25**	**48**

The trajectory curves in Fig. 2.6b differ less dramatically than the input curves, and (2.23)

explains why: the input enters the state update through a compensation term, so when the input magnitude and the state are at different orders of magnitude, the correction contributes little to the state estimate, and the benefit of the prior concentrates in the input channel. We also note that the input is scalar here ($m = 1$), where the exact AL-RKF projection (2.33) is ordinary clipping. The experiment does not evaluate multivariate interval constraints.

2.7 Conclusion

This chapter incorporated two practical forms of prior knowledge about an unknown input, a finite mode set and an interval bound, into recursive joint input and state estimation. For mode priors, the Bayesian multiple-model estimator AMM-KF maintains soft decisions through recursively updated mode weights, and Proposition 2 shows that the weights converge to the true mode under an explicit identifiability condition: the log-odds of every wrong mode drifts downward at a rate set by the KL divergence between the mode-conditioned innovation densities, so the consistency of the decisions is accumulated evidence from a sequential hypothesis test. The single-step error probability (2.30) quantifies when the modes are identifiable in the first place, and a weight floor bounds the response delay after a mode switch. For interval priors, the AL-RKF gain re-optimization reduces to clipping under the stated rank conditions. When the constraint is active, the estimate can be biased and non-Gaussian; the variance reduction is therefore a bias–variance tradeoff rather than an unbiased or MVU improvement. In the scalar wind simulation, AL-RKF reduced input-estimation variance by 81% relative to the KF and 47% relative to the RKF, while its mean error was worse than the RKF in four of six settings because the true wind speed lies on the constraint boundary before onset. The value of these estimators is that they extend the capability of an existing state estimator to evaluate unknown disturbances, whether an environmental event, an internal sensor failure, or an external collision force, without adding sensors.

Everything in this chapter concerns a single agent estimating with its own sensor. In

the monitoring missions that motivate this dissertation, however, the same unknown input is typically witnessed by *many* agents, each with a different, possibly deficient view: an individual sensor may observe only one coordinate of the target, and the communication topology among agents may change as they move. Chapter 3 extends the estimator developed here to that setting. The per-agent quantities derived above, namely the MVU input gain (2.19), the input estimate and its covariance (2.20), and the predicted and updated state statistics, become precisely the messages that agents exchange with their neighbors, and covariance-intersection input fusion together with information-filter state fusion recovers near-centralized performance without sharing raw measurements or system models. A second boundary of this chapter is the rank condition of Assumption 1, which every estimator above requires. Chapter 4 returns to centralized estimation in the regime where that condition fails and the continuous input instead carries structural prior knowledge in the form of a tree.

CHAPTER 3

Decentralized Input and State Estimation

This chapter is adapted from [Wu and Mehta, "Decentralized Input and State Estimation for Multi-agent System with Dynamic Topology and Heterogeneous Sensor Network," CDC 2024] [123].

3.1 Overview

Chapter 2 developed a joint input-and-state estimator for a single agent observing a system driven by an unknown input. This chapter decentralizes that estimator across a sensor network with dynamic topology and heterogeneous sensors, where dynamic topology refers to observation agents and neighborhoods that change over time and heterogeneous sensors refers to different observation models across agents. Each node sends its input and state estimates and their covariance matrices to its one-hop neighbors. It does not send raw measurements or system models. Under the assumptions below, the information-filter decomposition gives the same estimate as the corresponding centralized estimator. For fixed state and input dimensions, each message has fixed size. Input fusion utilizes covariance intersection [47, 20, 56, 21], and state fusion, through information filter decomposition [32], achieves performance comparable to full-information filters without the correlation issues that arise if all-to-all communication is allowed.

A further consequence of fusion is enhanced observability. Even if an individual sensor does not satisfy local observability on its own, the fusion process incorporates information

from multiple sources and effectively improves the network's overall observability. As a result, a network of locally-unobservable sensors can achieve the same state-reconstruction accuracy as a fully observable system, provided the collective sensor network satisfies the necessary observability conditions.

Our method also includes specific operations, such as reviewing observation time windows and state compensation, to enhance estimation in dynamic topologies with a limited sensing range. Our algorithm requires only a one-time information exchange and does not need all-to-all communication or a full communication rate.

3.2 Problem Formulation

3.2.1 System description

From the view of a single agent i, the system description is as follows:

$$x_k^i = A_{k-1}^i x_{k-1}^i + G_{k-1}^i d_{k-1}^i + w_{k-1}^i \tag{3.1}$$

$$y_k^i = H_k^i x_k^i + v_k^i \tag{3.2}$$

where the noise w_k^i and v_k^i are independent, zero-mean, white Gaussian noises with covariance matrices are Q^i and R^i. We primarily focus on homologous states and inputs across agents, meaning system dynamics are consistent. Therefore for simplicity, we assume $A_k^i = A_k$, $G_k^i = G_k$, $Q_k^i = Q_k$, state $x_k^i \in \mathbb{R}^n$, input $d_k^i \in \mathbb{R}^m$, and output $y^i \in \mathbb{R}^{p^i}$ with varying p^i across different observation models. In this system, both state x_k^i and input d_k^i are unknown, with each agent aiming to optimally estimate them.

Agents start with individual input estimation using the initial steps of the recursive Kalman filter (RKF) [36, 125], where they predict the state and covariance according to Eq. (3.3) and estimate the input via Eq.(3.4).

$$\begin{aligned}\hat{x}^i_{k|k-1} &= A^i_{k-1}\bar{x}^i_{k-1|k-1} \\ \hat{P}^i_{k|k-1} &= A^i_{k-1}\bar{P}^i_{k-1|k-1}A^{i^T}_{k-1} + Q^i_k\end{aligned} \tag{3.3}$$

Input estimation:

$$\hat{d}^i_{k-1} = M^i_k\left(y^i_k - H^i_k\hat{x}^i_{k|k-1}\right) \tag{3.4}$$

where M^i_k is:

$$M^i_k = [(H^i_kG^i_{k-1})^T\tilde{R}^{i^{-1}}_k(H^i_kG^i_{k-1})]^{-1}(H^i_kG^i_{k-1})^T\tilde{R}^{i^{-1}}_k \tag{3.5}$$

$$\tilde{R}^i_k = H^i_k\hat{P}^i_{k|k-1}H^{i^T}_k + R^i_k \tag{3.6}$$

The covariance of input estimation is,

$$\hat{S}^i_k(\hat{d}^i_k) = [(H^i_kG^i_{k-1})^T(\tilde{R}^i_k)^{-1}(H^i_kG^i_{k-1})]^{-1} \tag{3.7}$$

The rank condition for input estimation is the same as RKF,

$$\text{rank}(H^i_kG^i_{k-1}) = \text{rank}(G^i_{k-1}) = m \tag{3.8}$$

Following individual estimations, agents proceed differently from RKF by performing a pure Kalman filter (KF) update without including input estimation. This choice avoids extra computation as input estimation's inclusion does not alter the fusion result in the final algorithm but increases the computational load. The explanation will be detailed in the state fusion section.

$$\begin{aligned}\tilde{x}^i_{k|k} &= \hat{x}^i_{k|k-1} + L^i_k\left[y^i_k - H^i_k\hat{x}^i_{k|k-1}\right] \\ \tilde{P}^{i^{-1}}_{k|k} &= \hat{P}^{i^{-1}}_{k|k-1} + H^i_kR^{i^{-1}}_kH^i_k \\ L^i_k &= \tilde{P}^i_{k|k}H^{i^{-1}}_kR^{i^{-1}}_k\end{aligned} \tag{3.9}$$

Each agent's estimation results in input and state estimations along with their covariances, presented as $\hat{d}^i_k$, $\hat{S}^i_k$, $\hat{x}^i_{k|k-1}$, $\hat{P}^i_{k|k-1}$, $\tilde{x}^i_{k|k}$, and $\tilde{P}^i_{k|k}$, adhering to the unbiased and minimum-variance (MVU) principle. Due to bandwidth and privacy limitations, transmitting direct

observation or sharing agents' system dynamics or observation models is infeasible. This chapter proposes a fusion algorithm using only these values to achieve an approximate optimal fusion result, bypassing the need for system information exchange. The fusion algorithm comprises two parts: input fusion and state fusion, which are explored in subsequent sections.

3.3 Algorithm

3.3.1 Input fusion via Covariance Intersection

Within the 1-hop distance of node i, including itself, represented by $\mathcal{N}_i$, communication is confined to $\mathcal{N}_i$. The correlation among received input estimations d_j ($j \in \mathcal{N}^i$) is unknown, preventing optimal fusion akin to fully informed filters [77]. We employ a method known as covariance intersection (CI) for fusion under uncertain correlation. CI, typically used in Kalman filter-based information fusion when source correlations are unclear, offers a way to merge information without complete correlation knowledge, albeit solving its non-linear optimization problems can be computationally intensive. This chapter introduces a rapid CI approach [47, 88, 46] to maintain fusion consistency and ensure real-time computational feasibility.

$$\bar{S}_k^{i^{-1}} \bar{d}_{k-1}^i = \sum_j^{\mathcal{N}_i} \omega_j \hat{S}_k^{j^{-1}} \hat{d}_k^j \tag{3.10}$$

where weights are decided by faster CI which avoids complex matrices manipulation and non-linear optimization.

$$\omega_j = \frac{1/\operatorname{tr} \hat{S}_j}{\sum_{n=1}^{\mathcal{N}_i} 1/\operatorname{tr} \hat{S}_n} \tag{3.11}$$

$$\bar{S}_k^{\,i} = \left(\sum_{j \in \mathcal{N}_i} w_j \hat{S}_k^{j^{-1}} \right)^{-1} \tag{3.12}$$

Given that all input estimations $\hat{d}^j$ are unbiased, the fusion process described in Eq.(3.10) remains unbiased. It's important to note that this covariance fusion method does not affect future input variance predictions.

In scenarios when moving agents with dynamic communication topology, the target is intermittently observed. If input estimation is constantly employed, abrupt measurements may be misinterpreted as input influences affecting state estimates, which leads to a significant bias. To mitigate this, we propose a specific operation: establishing an observation time window review mechanism, the length of the window can be regarded as a hyperparameter. If the gap between two consecutive measurements exceeds this window, the initial estimation upon receiving new observations will exclude input estimation. After implementing a review through the observation window before input estimation, our algorithm can mitigate this sensitivity issue. However, the time window should be reset to zero if valid input estimates from neighborhoods are available during the input fusion stage. The success of this approach is demonstrated in Fig. 3.5.

3.3.2 State fusion via information filter decomposition

Rather than iterative state exchange for consensus among agents, state fusion leverages information Kalman filter decomposition[32]. This fusion approach ensures performance equivalent to an agent with complete neighborhood information in both state estimation and covariance matrix.

After input fusion, yielding fused input estimation $\bar{d}_k^i$ and covariance $\bar{S}_k^i$, we proceed from an information Kalman filter perspective, incorporating all node data within $\mathcal{N}_i$. The final state estimation relies solely on $\bar{d}_k^i$, $\hat{x}_{k|k-1}^i$, $\hat{P}_{k|k-1}^i$, $\tilde{x}_{k|k}^i$, $\tilde{P}_{k|k}^i$.

In order to avoid symbol chaos of equations, we use superscript $*$ on states and covariance to indicate the state fusion process in contrast to the process before communication. For

node i, incorporating input fusion into the prediction process yields:

$$\begin{aligned}\hat{x}^{*^i}_{k|k-1} &= A^i_{k-1}\bar{x}^i_{k-1|k-1} + G^i_{k-1}\bar{d}^i_{k-1} \\ \hat{P}^{*^i}_{k|k-1} &= A^i_{k-1}\bar{P}^i_{k-1|k-1}A^{i^T}_{k-1} + Q^i_k\end{aligned} \tag{3.13}$$

Here, the covariance update is the same as Eq. (3.3). This operation is justified because the input covariance impacts only the Kalman filter's gain matrix, potentially leading to non-unique Kalman gains and additional computational load. Previous research[36, 65] indicates that omitting input covariance injection still yields near-optimal Kalman gain. Moreover, exact state-input correlation would require full distributed agent system knowledge, deviating from our objective. Various consensus algorithms[60, 92, 59, 83] demonstrate that fusion with approximate covariance can achieve satisfactory performance, hence this chapter proceeds without detailed proof of covariance fusion's impact.

Assuming a virtual information filter with H_k, y_k, and R_k representing aggregated matrices from $\mathcal{N}i$, we simulate all observations being accessible to node i. Importantly, we show the final fusion result does not depend on external observations or matrices. Maintaining the prediction from Eq.(3.13), the update process follows:

$$\begin{aligned}\bar{x}^i_{k|k} &= \hat{x}^{*^i}_{k|k-1} + W_k\left[y_k - H_k\hat{x}^{*^i}_{k|k-1}\right] \\ \bar{P}^{i^{-1}}_{k|k} &= \hat{P}^{*^{i^{-1}}}_{k|k-1} + H_k R_k^{-1} H_k \\ W_k &= \bar{P}^i_{k|k}H_k^\top R_k^{-1}\end{aligned} \tag{3.14}$$

Substitute Eq.(3.13) to Eq.(3.14),

$$\bar{x}^i_{k|k} = \bar{P}^i_{k|k}[H_k^\top R_k^{-1} y_k + \hat{P}^{*^{i^{-1}}}_{k|k-1}\hat{x}^{*^i}_{k|k-1}] \tag{3.15}$$

After unpacking the observation vector,

$$H_k^\top R_k^{-1} y_k = \sum_{j\in\mathcal{N}} H_k^{jT} R_k^{j^{-1}} y_k^j \tag{3.16}$$

$$H_k^\top R_k^{-1} H_k = \sum_{j\in\mathcal{N}} H_k^{jT} R_k^{j^{-1}} H_k^j \tag{3.17}$$

Up to now, the state fusion is derived as:

$$\bar{x}^i_{k|k} = \bar{P}^i_{k|k}[\sum_{j\in\mathcal{N}}(\tilde{P}^{j^{-1}}_{k|k}\tilde{x}^j_{k|k} - \hat{P}^{j^{-1}}_{k|k-1}\hat{x}^j_{k|k-1}) + \hat{P}^{*^{i^{-1}}}_{k|k-1}(\hat{x}^i_{k|k-1} + G^i_{k-1}\bar{d}^i_{k-1})] \tag{3.18}$$

$$\bar{P}^{i^{-1}}_{k|k} = \hat{P}^{*^{i^{-1}}}_{k|k-1} + \sum_{j\in\mathcal{N}}(\tilde{P}^{j^{-1}}_{k|k} - \hat{P}^{j^{-1}}_{k|k-1}) \tag{3.19}$$

Algorithm 2 Decentralized input and state estimation at node i

Input: System models A^i_k, G^i_k, H^i_k, Q^i_k, and R^i_k

Dynamic one-hop neighborhood $\mathcal{N}_i$

Output: Fused input $\bar{d}^i_k$ and covariance $\bar{S}^i_k$

Fused state $\bar{x}^i_{k|k}$ and covariance $\bar{P}^i_{k|k}$

Initialize $x^i_0 \leftarrow x_0$

for $k = 1, 2, \ldots$ **do**

Step 1: Individual estimation

Compute the local state update using Eq. (3.9)

if the measurement gap is within the observation window **then**

Compute $\hat{d}^i_k$ and $\hat{S}^i_k$ using Eq. (3.4)

end if

Step 2: Communication

Exchange $\hat{d}^j_k$, $\hat{S}^j_k$, $\hat{x}^j_{k|k-1}$, $\hat{P}^j_{k|k-1}$, $\tilde{x}^j_{k|k}$, and $\tilde{P}^j_{k|k}$ with $j \in \mathcal{N}_i$

Step 3: Input fusion

Compute $\bar{d}^i_k$ and $\bar{S}^i_k$ using Eqs. (3.10) and (3.12)

Step 4: State fusion

Compute $\bar{x}^i_{k|k}$ and $\bar{P}^i_{k|k}$ using Eqs. (3.18) and (3.19)

Apply state compensation using Eq. (3.20)

end for

Initially, agents estimate these quantities independently. Post-communication, input and state fusion follows Eq.(3.10), Eq.(3.12), Eq.(3.18), and Eq.(3.19), outlined in Algo.2. As this decomposition shows the exact same as the central estimator, the final fused states also have an unbiased and minimum-variance guarantee.

For non-all-to-all and dynamic communication topologies, a diffusion operation post-Eq.(3.18) helps correct state estimation errors:

$$\bar{x}^i_{k|k} = \bar{x}^i_{k|k} + \epsilon M_i \sum_{j \in \mathcal{N}} (\hat{x}^{*^j}_{k|k-1} - \hat{x}^{*^i}_{k|k-1}) \tag{3.20}$$

with $M_i = \bar{P}^i_{k|k}$ and ϵ set to 0.05. This state adjustment, averaging predictions, differs from consensus or diffusion KF, which requires further information exchange post-fusion. Our decomposed algorithm naturally zeros out such an adjustment in all-to-all communication networks, offering advantages over consensus-based filters.

3.4 Simulation and Experiment Result

3.4.1 Experiment Setup

In this section, we describe two challenging experiments inspired by a canonical scenario, where a target moves around a base (e.g., the coordinate origin), with its movement influenced by internal unknown inputs, such as sensor faults. Consequently, the trajectory's geometric center deviates from the coordinate origin, as depicted in Fig.3.1. The target's path is segmented into four quadrants, each monitored by a sensor that detects only one dimension of the target's position. All experiment results shown in this chapter are carried on under 20 different seeds and averaged. Table 3.2 indicates the average of statistics of the first experiment. Fig. 3.9 shows the MAE, RMSE as well as variance from all seeds of the second dynamic topologies. Table 3.1 showcases the average statistics of state compensation in both two experiments. In the first example, agents are static to observe a moving target. Each of

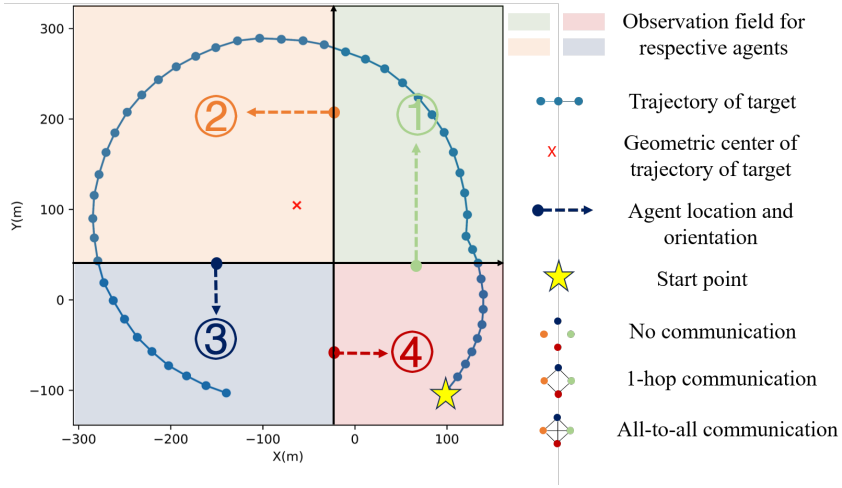


Figure 3.1: Scenario 1: Four static agents are located in the four quadrants. Two of them can observe only the y-coordinate of the target, while another two are limited to observing only the x-coordinate. Each agent is tasked with observing the target exclusively when it enters its respective quadrant. The communication is established according to three distinct typologies.

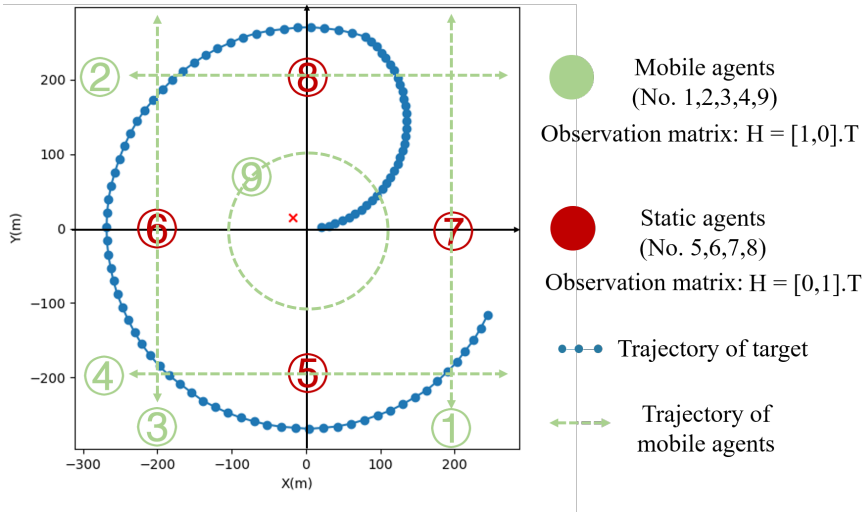


Figure 3.2: Scenario 2: 4 static (red) agents and 5 mobile (green) agents compose a dynamic system. Observation of the target by an agent is contingent on sharing the same quadrant with the target, with each agent limited to observing a single target dimension. The trajectories of the mobile agents are delineated by green dashed lines. Each agent possesses a pre-defined communication range, enabling information exchange solely when the communication ranges of two agents intersect.

the agents has a different observation model. Specifically, Sensor No.1 and No.3 can measure the x position in its quadrant, $H_1 = H_3 = [1, 0]$, while Sensor No.2 and No.4 will measure the y position, $H_2 = H_4 = [0, 1]$. These sensors ensure at any moment, the target is tracked by only one sensor for a singular dimension. During this experiment, observations from a central agent are absent akin to environmental monitoring practices[39].

In the second example, the sensor network consists of two groups: dynamic and static sensors. Most settings are similar to the first example. In particular, the communication topology is fully dynamic, as each agent has its own communication field. A communication link is established only when agents are within range to communicate. The communication range in this experiment is 120, the detail is seen in Fig. 3.2.

The system matrices describe a dynamic model that makes the target move in a circular trajectory.

$$A_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \tag{3.21}$$

where $\theta_k = \omega + \theta_0$,ω is the angular velocity. k is the timestep, θ_0 is the initial angle.

$$G_k = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad B_k = \begin{bmatrix} d(t) \end{bmatrix} \tag{3.22}$$

where $d(t)$ is the internal unknown input that cannot be predicted and observed directly, like the internal sensor fault in a sensor network. To mimic the changes of inputs over time. The function of $d(t)$ is:

$$d(t) = \begin{cases} 0, & \text{if } k < 10 \\ 10, & \text{if } k \geq 10 \end{cases} \tag{3.23}$$

$$Q_i = [0.2, 0.2]^T; R_i = [2]^T; i \in \{1, 2, 3, 4.., N\} \tag{3.24}$$

Our algorithm is denoted as DISKF (Decentralized Input and State Kalman Filter), and we compare against five baselines:

- **C-DRKF** [77]: a locally centralized filter in which each agent has a global view of its neighbors' system equations.
- **L-DRKF** [96, 34]: state consensus with individual input estimation.
- **KCF** [83]: a consensus filter that takes a weighted average of exchanged states.
- **DiffKF** [18]: a diffusion filter that averages neighbors' states after each agent performs a state consensus.
- **GDKF** [100]: a gossip-based filter that randomly and asynchronously collects messages from neighbors.

All algorithms are allowed to communicate only once between two consecutive timesteps.

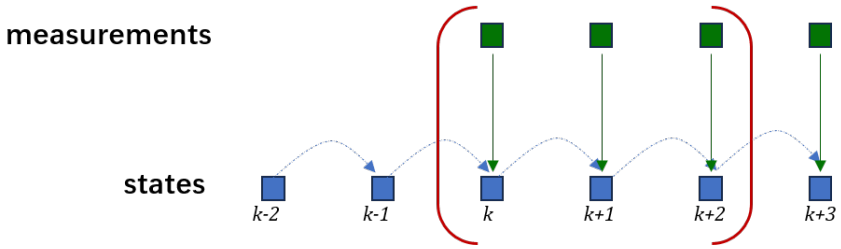


Figure 3.3: The time window review deals with the abrupt bias caused by input estimation facing intermittent measurements. When the sensor abruptly measures the target, within the period of the time window review, only do state estimation without input estimation.

3.4.2 Result

We present the 4-agent case with 1-hop communication and the dynamic scenario with 9 agents.

Firstly, we illustrate input estimation using 1-hop communication in Fig.3.4, whereas some baselines like KCF, DiffKF, and GDKF lack detailed input estimation in their documentation. We adopted the RKF's input estimation method[36] for comparison with these algorithms,

noting that this adaptation only serves for comparison but didn't affect state estimation; the original algorithms' state estimations were preserved.

In 1-hop limited communication where only one sensor observes the target at any time, no agent can have constant information about the target, thus leading to missing data in Fig.3.4. Algorithms without specialized input designs, like KCF, DiffKF, and GDKF, exhibit significant bias However, filters with comprehensive information of neighbors may also face challenges, as indicated by the bias in input estimation for nodes 3 and 4 when the target enters quadrants 3 and 4. This occurs because before entering quadrant 3, node 3's state and covariance evolve fully depending on the prediction model which makes state error significantly increases, whereas node 2's estimation adjusts upon the target's entry into quadrant 1. Based on Eq. (3.4), the input estimation depends on the observation, current states, and its covariance. Then if using node 2's observation but with node 3's state, the estimator will misinterpret the innovation as the influence of unknown input. This is a similar issue we discussed in the observation time window analysis above. However, without state and input fusion, even the C-DRKF with full observation and corresponding matrices still cannot solve this issue. Our fusion method, considering both input and state along with their covariances, maintains stable input estimation.

State estimations are depicted in Fig.3.6 and Fig.3.7, demonstrating that a sensor's singular dimensional observation leads to asynchronous x and y estimation trends. Algorithms lacking input estimation invariably present bias in state estimations. While Kalman filters can naturally compensate for sudden disturbances over time, continuous or lasting inputs can still introduce bias during estimation.

Estimated trajectories in Fig.3.8 showcase our algorithm's robustness in scenarios marked by incomplete observations and unknown internal inputs. Despite limited information, our algorithm compares favorably or even outperforms others.

Additionally, we examine input and state estimations across three different topologies in Fig.3.1, using agent node 2 as an example. Fig.3.10 illustrates that isolated estimations

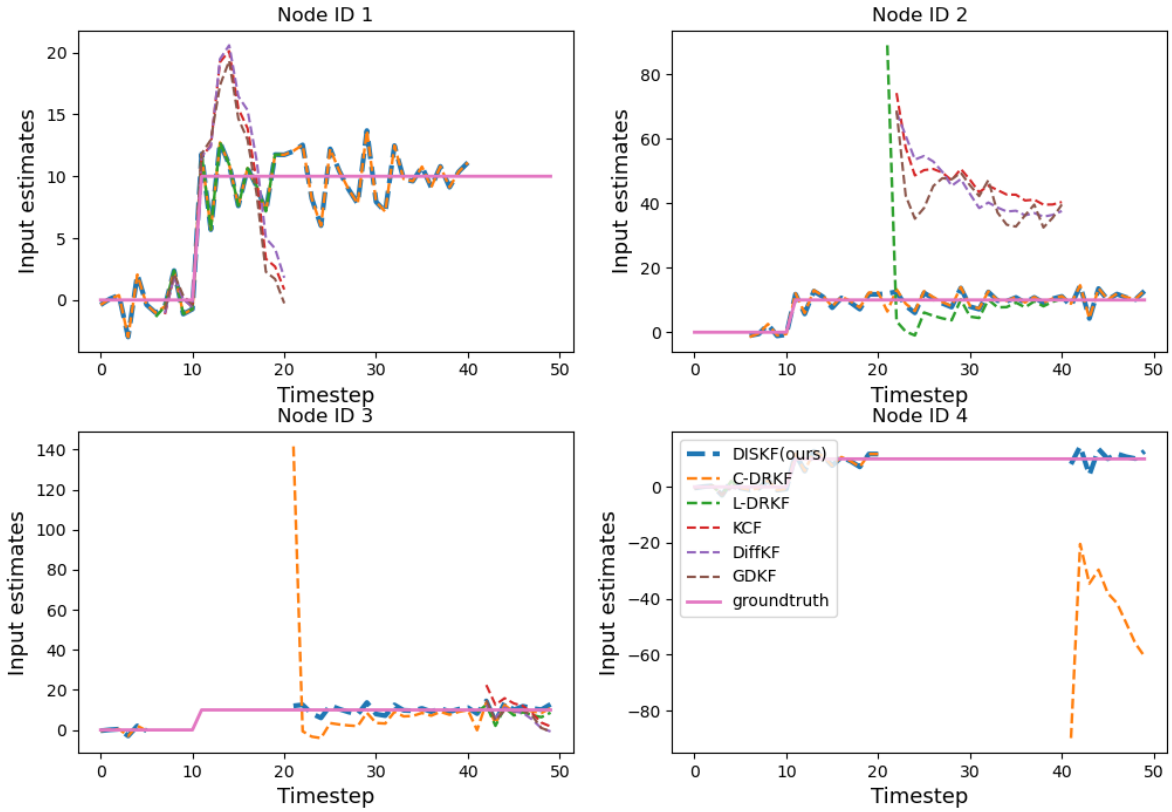
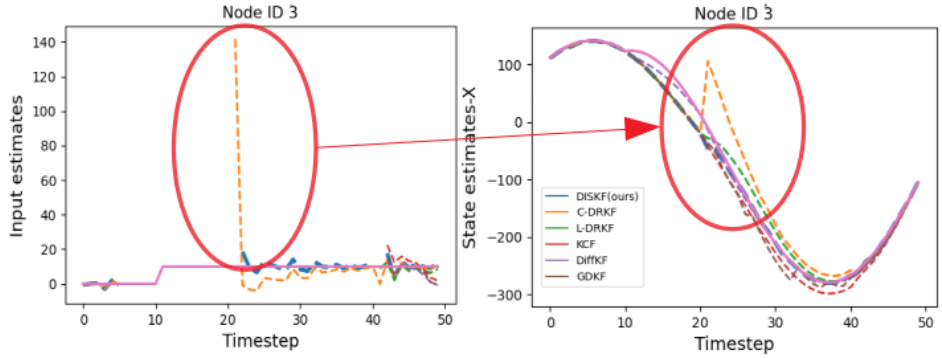
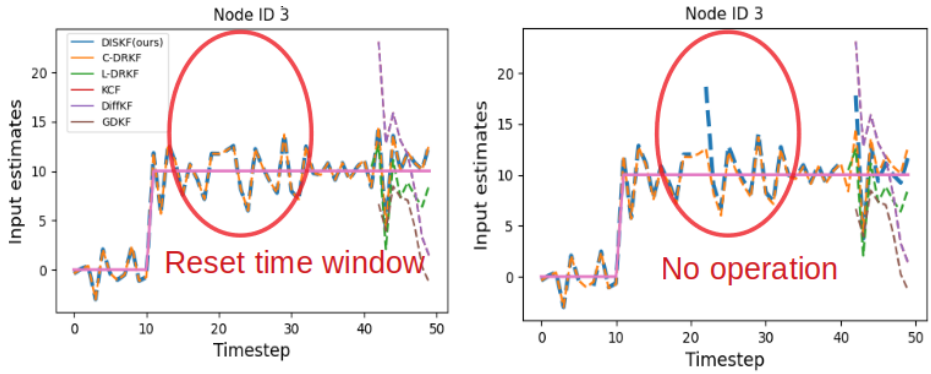


Figure 3.4: The input estimation with 1-hop communication. The index of nodes corresponds to Fig.3.1. Missing estimation means the corresponding algorithm can't recover the input from the data they observe or exchange.



(a) Abrupt measurements induce huge bias in input estimates (left) and then influence state (right).



(b) Reset time window counting during input fusion improves estimations

Figure 3.5: Observation time window helps deal with intermittent observations in dynamic topology. (a) the sensitivity issue is mitigated by implementing a review through the observation window before estimating inputs. (b) the time window counting should be reset when valid input estimations are available from neighborhoods.

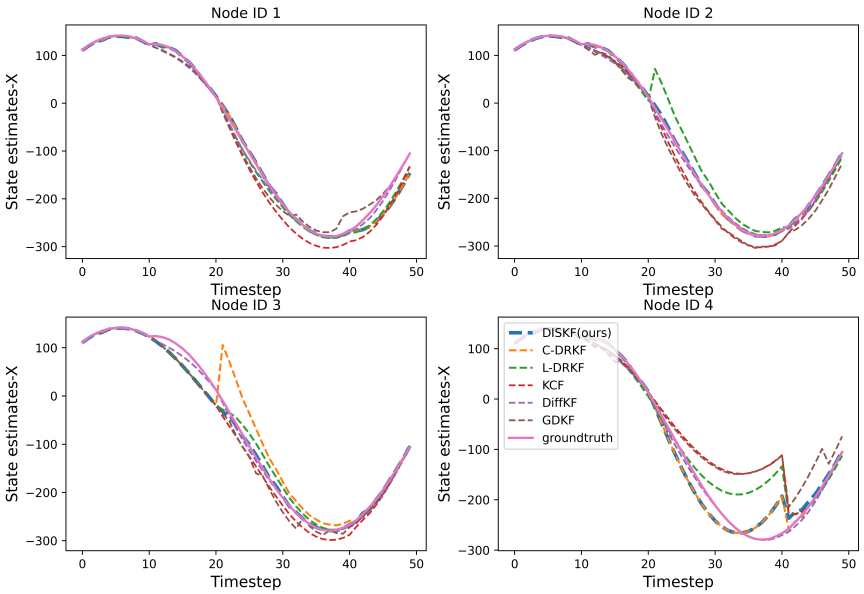


Figure 3.6: The x coordinate of position estimation of the target with 1-hop communication. 1) Our algorithm (blue line) can converge back to groundtruth very quickly once one of the nodes within 1-hop has the observation of the target. 2) even with limited information exchange, our algorithm is still very close to the centralized filter (orange line) with full information.

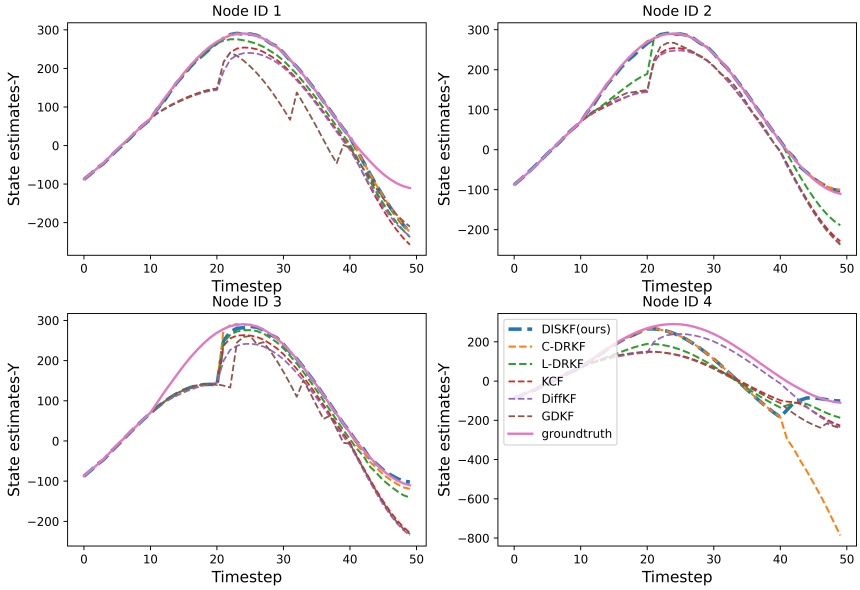


Figure 3.7: The y coordinate of position estimation of the target with 1-hop communication.

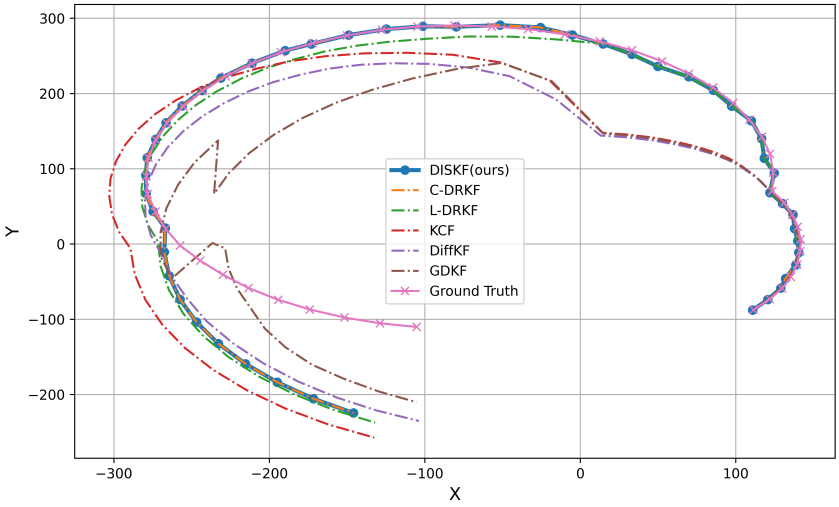
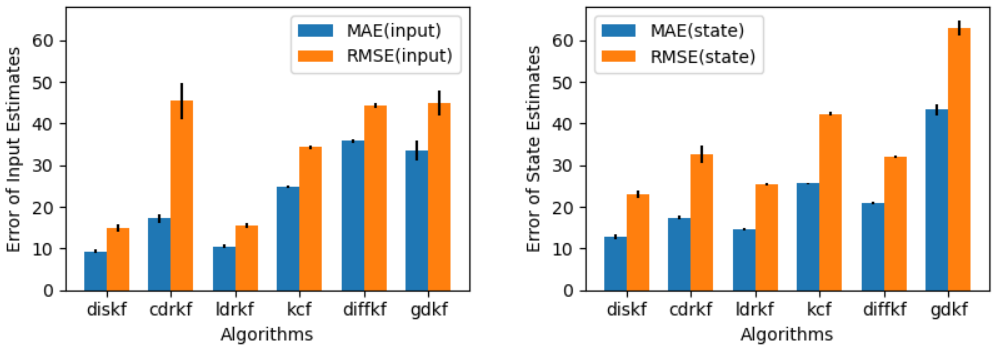


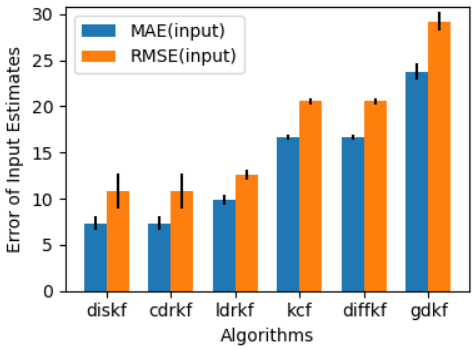
Figure 3.8: 2D trajectory visualization of all estimators with 1-hop communication. Our algorithm (blue line) exhibits robustness against unexpected internal input disturbance while maintaining performance parity with the filter that possesses complete neighborhood information.

by each agent yield comparable performance among algorithms with and without input estimation. However, under all-to-all communication, our algorithm (blue line) mirrors the performance of the fully informed C-DRKF filter, aligning with our theoretical expectations, which can also shown in Table 3.2.

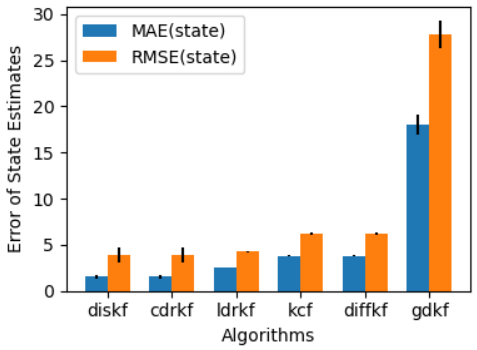


(a) Input estimation (dynamic)

(b) State estimation (dynamic)



(c) Input estimation (all-to-all)



(d) State estimation (all-to-all)

Figure 3.9: DISKF(ours) shows superiority over all baselines in multi-agent case under dynamic topology with heterogeneous sensors. (a)(b): Input and state estimation error with dynamic communication topology (c): Input and state estimation error with all-to-all communication

Table 3.2 reports the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for the experiment with four stationary agents. Figure 3.9 shows that DISKF has lower error than the baselines under dynamic topology and similar error to C-DRKF under all-to-all

Table 3.1: Evaluation of State Compensation Effect

Environment	Communication	State comp.	Input error		State error	
			MAE	RMSE	MAE	RMSE
Stationary	All-to-all	Yes	4.63	6.35	1.91	2.66
		No	4.63	6.35	1.91	2.66
	1-hop	Yes	**15.76**	**30.05**	**4.67**	**6.37**
		No	26.75	55.63	7.38	15.36
Dynamic	All-to-all	Yes	1.56	4.01	7.40	11.08
		No	1.56	4.01	7.40	11.08
	1-hop	Yes	**10.37**	**19.34**	**8.82**	**14.22**
		No	16.14	26.47	9.11	14.76

communication. The dynamic-topology results in that figure use a communication range of 120. Figure 3.11 reports how the MAE changes with communication range. Under dynamic topology, C-DRKF becomes unstable when observations are intermittent, whereas DISKF remains stable.

Table 3.1 demonstrates the effectiveness of state compensation, as detailed in Eq.(3.20). As noted earlier, in the scenario where there is all-to-all communication, state compensation naturally reduces to zero—a unique attribute not seen in other consensus algorithms, as the statistical data underlines. In a dynamic communication topology, state compensation improves the accuracy of estimations. The relative advantage of state compensation is also revealed in the analysis between C-DRKF and L-DRKF. Unlike C-DRKF, L-DRKF incorporates an additional operation of state compensation (consensus) beyond input estimation. Hence, despite only relying on local information for input estimation, L-DRKF outperforms C-DRKF in certain situations.

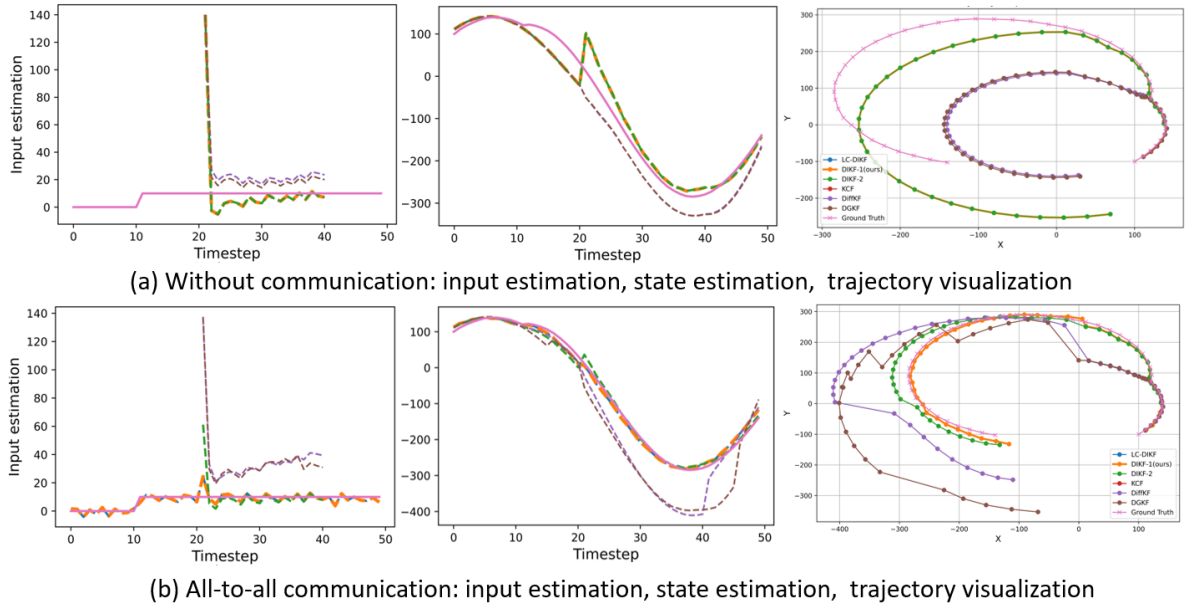
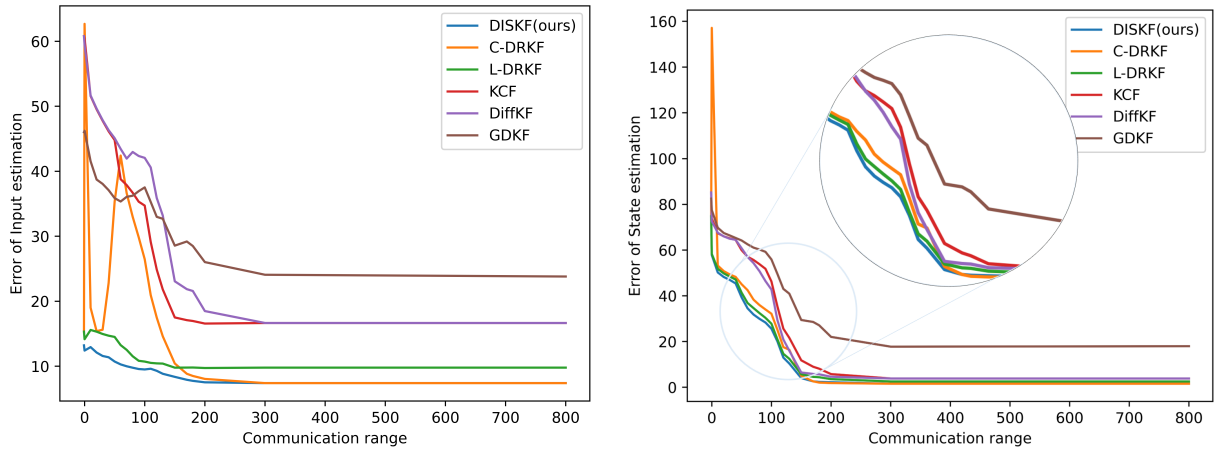


Figure 3.10: Supplementary results of communication without communication and all-to-all communication of node 2.



(a) Input estimation versus radius

(b) State estimation versus radius

Figure 3.11: Algorithm performance versus communication ranges in the dynamic topology scenario. (a) and (b) show the Mean Absolute Error (MAE) of input and state estimation under different communication ranges of each agent.

Table 3.2: State and Input Estimation Error in Four Agents Case under Stationary Topology with Heterogeneous Sensors

Comm.	Metric	**DISKF (ours)**	C-DRKF [77]	L-DRKF [96]	KCF [92]	DiffKF [18]	GDKF [100]
all-to-all	MAE(state)	**1.91**	**1.91**	15.66	27.90	27.90	36.20
	RMSE(state)	**2.66**	**2.66**	25.05	40.45	40.45	50.15
	MAE(input)	**4.63**	**4.63**	6.87	15.39	15.39	14.01
	RMSE(input)	**6.35**	**6.35**	9.75	16.51	16.51	15.12
1-hop	MAE(state)	**15.76**	26.76	26.08	37.40	24.87	40.92
	RMSE(state)	**30.05**	55.63	41.54	53.41	40.84	58.89
	MAE(input)	**4.67**	9.73	6.30	15.11	14.53	13.35
	RMSE(input)	**6.37**	17.34	9.72	16.94	15.47	14.33
None	MAE(state)	**63.75**	71.01	71.01	66.43	66.43	65.82
	RMSE(state)	92.88	109.31	109.31	90.78	90.78	**90.36**
	MAE(input)	**9.10**	21.76	21.76	16.32	16.32	12.65
	RMSE(input)	**12.21**	32.33	32.33	17.95	17.95	14.50

3.5 Conclusion

This chapter introduced a decentralized, resilient, and efficient input and state estimation algorithm. Through system decomposition, state compensation, and observation time window review techniques, our algorithm can deal with dynamic multi-agent systems that consist of heterogeneous sensors with intermittent observations for each agent. This enables us to extend to cooperative tracking, ad-hoc sensor networks, and large spatial-temporal systems like geophysical monitoring, with real-time and privacy demand.

Both this chapter and Chapter 2 rest on the input-rank condition (3.8): every input channel must be visible through the measurements collected at a single step. Many monitoring

problems violate it outright, with thousands of candidate input locations and only a few dozen sensors. Chapter 4 next uses tree structure to restore a solvable reduced estimation problem in this regime. The resulting source belief also provides the information needed to place mobile sensors.

CHAPTER 4

Input Estimation and Mobile Coverage Beyond the Rank Condition on Tree Networks

This chapter is adapted from the estimation and coverage portions of a manuscript in preparation [Wu and Mehta, "Mobile Sensor Coverage Beyond the Input-Rank Condition: Active Input Localization on Tree Networks"] [124]. All results reported here are simulation studies.

4.1 Overview

The estimators in Chapters 2 and 3 require the input-rank condition of Assumption 1: every input direction must leave a linearly independent trace in the measurements. That condition fails on a sparsely observed river network. Thousands of reaches may receive an unknown flood inflow, while only a small set of fixed and mobile sensors is observed at any one time. The full input therefore has far more degrees of freedom than the measurements.

This chapter uses the directed-tree structure of the network as a structural prior. Restricting the estimated input to the currently observed nodes turns the wide measurement map into a square lower-triangular matrix with positive diagonal. The reduced problem satisfies the rank condition for every nonempty sensor configuration and is solved by forward substitution. The reduction does not identify an unobserved source exactly. Instead, it assigns the source to its first observed downstream node, called its *frontier*, and locations sharing a frontier form an indistinguishable *sector*.

The same limitation motivates mobile coverage. The estimator converts frontier anomalies into a per-reach source belief. A mobile fleet then changes the observation set, creating new frontiers that can split a sector. We retain two simple planners: weighted Lloyd coverage and a tree-bisection heuristic. The purpose of the resulting simulations is limited: they check the mechanism and compare design choices under a common simulated environment; they do not establish field performance.

4.2 Problem Formulation

4.2.1 Tree transport model

The environment is a directed tree $G = (V, E)$ with n nodes indexed in topological order: $j < i$ whenever j is upstream of i, and every node except the root has a unique downstream successor. Let $U(i)$ denote the nodes strictly upstream of i and $D(i)$ its downstream chain. A per-node scalar state $\bar{x}_k \in \mathbb{R}^n$ obeys

$$\begin{aligned} \bar{x}_k &= A_e \left(u_k + d_k\right) + A_0 x_{k,0}, \\ x_{k+1,0} &= T_1 \left(u_k + d_k\right) + T_2 x_{k,0}, \end{aligned} \tag{4.1}$$

where $u_k \geq 0$ is a known nominal forcing, $x_{k,0}$ is the carry-over state, and $d_k \geq 0$ is an unknown additive input with unknown support. The goal is to detect an active input and localize its node or identifiable sector.

Assumption 2 (Tree transport and downstream aggregation)**.**

(A1) *Tree transport:* material injected at node f passes through exactly the downstream chain $D(f)$.

(A2) *Downstream aggregation:* in topological order, the operators in (4.1) are lower triangular, A_e has strictly positive diagonal, and $\operatorname{supp}((A_e)_{\cdot f}) = \{f\} \cup D(f)$.

The evaluated instance uses the 5131 reaches of the San Antonio–Guadalupe subbasin in Texas. Daily transport operators are aggregates of the 15-minute Muskingum routing steps in RAPID on the NHDPlus network [29]. The nominal forcing is land-surface-model lateral inflow, and d_k represents excess flood inflow. Fixed sensors are stream gauges and mobile sensors represent platforms carrying non-contact discharge measurement.

Proposition 6 (The river operators satisfy the class assumptions)**.** *In hydrological order, the daily Muskingum operators A_e, A_0, T_1, T_2 are lower triangular, and* $\operatorname{supp}((A_e)_{\cdot f}) = \{f\} \cup D(f)$. *The river instance therefore satisfies (A1)–(A2).*

Proof. The river incidence matrix has nonzero entries only from an upstream reach to its downstream successor and is therefore strictly lower triangular in hydrological order. The Muskingum substep operators, their powers, and their daily aggregates remain lower triangular. Their column support is the tree reachability encoded by the transitive closure of the incidence matrix. □

4.2.2 Observations

Let $G_{\text{fix}} \subset V$ be the fixed sensor set. Mobile sensors at planar positions $\pi_1, \ldots, \pi_M$ observe nodes within sensing radius r_s. The epoch-k observation set is

$$\mathcal{O}_k = G_{\text{fix}} \cup \left\{ i : \min_{1 \le j \le M} \|\xi_i - \pi_j\| \le r_s \right\}, \qquad p_k = |\mathcal{O}_k|, \tag{4.2}$$

where ξ_i is the planar coordinate of node i. A selection matrix $S_k \in \{0,1\}^{p_k \times n}$ extracts the observed components in topological order, and

$$y_k = S_k \bar{x}_k + v_k, \qquad \operatorname{Cov}(v_k) = R_k. \tag{4.3}$$

The estimation results below treat $\mathcal{O}_k$ as given. The coverage controller later chooses the next mobile-sensor positions, subject to a maximum travel distance $v_{\max}$ per epoch.

4.3 The Rank Obstruction and Its Restoration

4.3.1 Full-input obstruction

Given u_k and the current carry-over estimate $\hat{x}_{k,0}$, define the innovation against the source-free prediction:

$$z_k \triangleq y_k - S_k A_e u_k - S_k A_0 \hat{x}_{k,0} = S_k A_e d_k + e_k, \tag{4.4}$$

where e_k collects sensing, nominal-forcing, and carry-over errors. Its modeled covariance is

$$\widetilde{R}_k = S_k\big(A_e \Sigma_u A_e^\top + A_0 P_{x_0} A_0^\top\big) S_k^\top + R_k. \tag{4.5}$$

Estimating the full $d_k \in \mathbb{R}^n$ corresponds to an input matrix $G = I_n$. The rank condition would require

$$\operatorname{rank}(S_k A_e) = n, \qquad \text{but} \qquad \operatorname{rank}(S_k A_e) \le p_k \ll n. \tag{4.6}$$

The full input is therefore unidentifiable and the estimators of Chapters 2 and 3 cannot be applied to it directly.

4.3.2 Structural reparameterization

The tree prior enters through the input parameterization. Restrict the estimated input to the observed nodes,

$$d_k \approx S_k^\top \widetilde{d}_k, \qquad \widetilde{d}_k \in \mathbb{R}^{p_k}. \tag{4.7}$$

Substitution into (4.4) gives the square reduced system

$$z_k = H_k^{\mathrm{red}} \widetilde{d}_k + e_k, \qquad H_k^{\mathrm{red}} = S_k A_e S_k^\top \in \mathbb{R}^{p_k \times p_k}. \tag{4.8}$$

Proposition 7 (Feasibility: the rank condition is restored)**.** *Under (A2), for every nonempty $\mathcal{O}_k$, the matrix H_k^{red} is lower triangular and invertible, with*

$$\det H_k^{\mathrm{red}} = \prod_{b \in \mathcal{O}_k} (A_e)_{bb} > 0. \tag{4.9}$$

Consequently $\operatorname{rank}(H_k^{\mathrm{red}}) = p_k$*, and the input-rank condition holds for the reduced input for every sensor configuration.*

Proof. H_k^{red} is the principal submatrix of the lower-triangular A_e on the topologically ordered set $\mathcal{O}_k$. Its diagonal entries are $(A_e)_{bb} > 0$ by (A2). □

The minimum-variance unbiased gain for (4.8) is the whitened least-squares gain. Because H_k^{red} is square and invertible, it simplifies to

$$\begin{aligned} M_k &= \left[(H_k^{\mathrm{red}})^\top \widetilde{R}_k^{-1} H_k^{\mathrm{red}}\right]^{-1} (H_k^{\mathrm{red}})^\top \widetilde{R}_k^{-1} \\ &= (H_k^{\mathrm{red}})^{-1}. \end{aligned} \tag{4.10}$$

Thus the estimate and its covariance are

$$\boxed{\hat{\tilde{d}}_k = (H_k^{\mathrm{red}})^{-1} z_k, \qquad \hat{S}_k = (H_k^{\mathrm{red}})^{-1} \widetilde{R}_k (H_k^{\mathrm{red}})^{-\top}.} \tag{4.11}$$

Equation (4.11) is a forward substitution with cost $O(p_k^2)$. Nonnegativity can be imposed through the corresponding weighted nonnegative least-squares problem. The reduced estimate is also propagated through the carry-over model,

$$\hat{x}_{k+1,0} = T_1\big(u_k + S_k^\top \hat{\tilde{d}}_k\big) + T_2 \hat{x}_{k,0}. \tag{4.12}$$

This feedback matches the current observed state but does not reconstruct the true carry-over of an unobserved upstream source.

4.4 Equivalent Inputs and Identifiability

4.4.1 What the reduced estimate means

The reduced estimate is an equivalent input on the observed set, not an estimate of every component of the true input. For an unobserved node f, let the *frontier* $b^*(f)$ be the most upstream observed node on $D(f)$.

Proposition 8 (Support, attenuation, and explaining-away)**.** *Let a single unobserved node f carry input d_f, and consider the noiseless reduced solve. Then:*

(a) if $\mathrm{supp}(d) \subseteq \mathcal{O}$*, the reduction is lossless,* $\widetilde{d} = Sd$*;*

(b) every observed node off $D(f)$ *receives* $\widetilde{d}_o = 0$*;*

(c) the frontier absorbs the source with attenuation

$$\widetilde{d}_{b^*} = r(f \to b^*) d_f, \qquad r(f \to b) = \frac{(A_e)_{bf}}{(A_e)_{bb}} \in (0, 1], \tag{4.13}$$

and a downstream observed node v after an attributed observed node u obeys

$$\hat{\widetilde{d}}_v = \frac{z_v - (A_e)_{vu} \hat{\widetilde{d}}_u}{(A_e)_{vv}}. \tag{4.14}$$

Parts (a)–(b) follow from the column support in (A2), and part (c) is the first nonzero row of the forward substitution. Equation (4.14) subtracts input already attributed upstream. The method therefore changes the target rather than bypassing unobservability: an observed source is recovered at its own node, whereas an unobserved source is represented at its frontier.

To reduce persistent model bias before forming a belief, the implementation maintains a local moving mean $\hat{\mu}_i$ and variance $\hat{v}_i$ and emits the studentized anomaly

$$\lambda_i = \frac{\hat{\widetilde{d}}_i - \hat{\mu}_i}{\sqrt{\hat{v}_i + (\hat{S}_k)_{ii}}}. \tag{4.15}$$

This normalization is an engineering layer: exact standard-normal calibration requires locally stationary residuals and is not guaranteed under structural model error.

4.4.2 Sector resolution

For a fixed observation set $\mathcal{O}$, define the response direction of a unit input at node i by $c_i = (A_e)_{\mathcal{O},i}$.

Definition 1 (Equivalent-input direction, sector, and unobservable set)**.** Under the single-node hypothesis $\mathcal{H}_i : d = \alpha_i e_i$ with unknown amplitude $\alpha_i \geq 0$, the observation law depends on i through the ray $\{\alpha c_i : \alpha \geq 0\}$. The frontier $b^*(i)$ is the most upstream observed node in $\{i\} \cup D(i)$. Define

$$\mathrm{Sec}(b) = \{i : b^*(i) = b\}. \tag{4.16}$$

Nodes for which $(\{i\} \cup D(i)) \cap \mathcal{O} = \varnothing$ form the unobservable set and have $c_i = 0$.

Lemma 1 (Identifiability criterion)**.** *Two source hypotheses with unknown nonnegative amplitudes are indistinguishable if and only if $c_i \parallel c_j$. A node with $c_i = 0$ is invisible at $\mathcal{O}$.*

Proof. If $c_j = \eta c_i$ for $\eta > 0$, the two families of observation means coincide after rescaling the unknown amplitude. Distinct rays define distinct families. The claim for $c_i = 0$ is immediate. □

Theorem 1 (Sector resolution)**.** *Suppose transport is path-multiplicative on the tree: $(A_e)_{c,i} = (A_e)_{b,i}\kappa(b, c)$ whenever b lies on the path from i to c, with $\kappa(b, c) = (A_e)_{cb}/(A_e)_{bb}$ independent of i. Then every $i \in \mathrm{Sec}(b)$ satisfies*

$$c_i = (A_e)_{b,i} v_b, \qquad v_b = e_b + \sum_{c \in D(b) \cap \mathcal{O}} \kappa(b, c) e_c. \tag{4.17}$$

All nodes in a sector are therefore statistically indistinguishable at any signal-to-noise ratio, and there are at most $|\mathcal{O}|$ identifiable sectors. Under equal priors, the posterior is uniform within a sector.

Proof. For $b = b^*(i)$, the observed support of c_i is $\{b\} \cup (D(b) \cap \mathcal{O})$. Path multiplicativity factors the common scalar $(A_e)_{b,i}$ from every nonzero component, which yields (4.17). Lemma 1 completes the result. □

On the evaluated daily operator, path multiplicativity holds approximately because daily aggregation introduces small leakage. With 36 fixed gauges, the 5131 candidate reaches reduce to at most 36 observation-defined sectors, and 326 reaches with no downstream gauge are invisible to the fixed network. Mobile observations can refine this partition by placing a new frontier inside a sector.

4.5 From Source Belief to Mobile Coverage

4.5.1 Belief update

The estimator produces one λ_i for each observed node. Under a single-source hypothesis $\mathcal{H}_i$, node i is scored by the anomaly at its frontier $b^*(i)$. A transient-source likelihood can be hedged as

$$L_k^{(i)} = (1-q) + q \exp\left(\mu\lambda_{b^*(i)} - \frac{\mu^2}{2}\right), \qquad q \in (0,1), \tag{4.18}$$

with an explicit no-source hypothesis whose likelihood is one. Normalizing the accumulated hypothesis weights produces a posterior over the source location and the no-source state.

For simultaneous sources, the implementation instead maintains per-node log-odds ℓ_i with forgetting toward prior log-odds ℓ_0,

$$\ell_i \leftarrow \ell_0 + \rho(\ell_i - \ell_0) + \log\left[(1-q) + q \exp\left(\mu\lambda_{b^*(i)} - \frac{\mu^2}{2}\right)\right], \qquad p_i = \sigma(\ell_i), \tag{4.19}$$

where $0 < \rho \leq 1$. The coverage layer uses p_i as the current per-reach source belief. Equations (4.18)–(4.19) are detection-model choices for transient sparse floods, not consequences of the tree identifiability theory.

4.5.2 Posterior and uncertainty densities

A weighted coverage controller requires nonnegative node weights. Two choices are

$$\phi_i^{\text{post}} = p_i, \qquad \phi_i^{\text{unc}} = p_i(1-p_i). \tag{4.20}$$

The first follows posterior mass and is effective when only one unresolved event is present. It can, however, keep a fleet at a resolved source because $p_i \simeq 1$ remains a large weight. The uncertainty density is zero both for a cleared reach ($p_i \simeq 0$) and for a resolved reach ($p_i \simeq 1$), and is largest at $p_i = 1/2$. It therefore releases the fleet after resolution and is the appropriate choice when multiple events may occur over time. This distinction concerns the density supplied to the planner; it does not change the Lloyd controller itself.

4.5.3 Two planners

Weighted Lloyd. For either density in (4.20), assign each node to its nearest sensor to form discrete Voronoi cells $V_1, \ldots, V_M$. The weighted centroid of cell V_j is

$$c_j(\phi) = \frac{\sum_{i \in V_j} \phi_i \xi_i}{\sum_{i \in V_j} \phi_i}. \tag{4.21}$$

Each sensor moves toward $c_j(\phi)$ by at most $v_{\max}$ per epoch and stays in place if its cell has zero total weight. This is the standard weighted Lloyd update [24]; the design choice here is whether it covers posterior mass or posterior uncertainty.

Bisection heuristic. The tree also permits a directional proxy. Define the upstream posterior mass

$$W(r) = \sum_{i \in \{r\} \cup U(r)} p_i, \qquad r^* = \arg\max_r \min\{W(r), W_{\text{tot}} - W(r)\}, \tag{4.22}$$

where $W_{\text{tot}} = \sum_i p_i$. The planner selects a near-half-mass query for each sensor and assigns sensors to targets by proximity. Although exact information-gain placement has standard submodular guarantees [87, 69], Eq. (4.22) omits outcome-dependent sensing quality. It is therefore evaluated only as a heuristic and is not claimed to inherit those guarantees.

4.6 Simulation Evidence and Limitations

The identical-twin experiments use the same routing model for truth and estimation and are intended only for mechanism comparisons [57]. In the single-flood study, ten point floods with onset on day 15 are evaluated with drone-only sensing. Table 4.1 retains the common-policy comparison.

Table 4.1: Single-flood policy comparison: ten point floods (peak 25 m^3/s, onset day 15), drone-only sensing, identical twin. Detect and localize are median days.

policy	detect	localize	localized %
bisection information gain	**18**	**21**	**100%**
Lloyd (posterior density)	27	28	100%
Lloyd (exploration floor $\beta_{\text{rel}} = 0.1$)	25	28	80%
random	18	32	90%
static	never	never	0%

The bisection heuristic localizes earliest in this particular identical-twin setting, while static sensing does not localize any event. These values compare policies under matched simulations; they are not field-performance estimates.

The density choice becomes decisive with two staggered floods. Across eight episodes, weighted Lloyd on posterior density localizes the first flood and then tends to remain there, giving 25% success on both floods. Using the same Lloyd update with $\phi_i = p_i(1 - p_i)$ releases the fleet after the first source is resolved and raises both-flood success to 88%. The controller is unchanged; only its density changes.

An additional model-mismatch check replays the July 2025 Kerrville event through a National Water Model operational analysis whose routing and forcing differ from the estimator [89]. All four tested policies detect on day 8, and the Lloyd posterior mode is 5.6 km from the target reach by day 9. Across twelve Lloyd deployments, the median final

distance is 9.9 km. This favorable case lies close to a fixed gauge and within a small sector, and the nature run remains a model rather than field truth.

Three limitations bound these results. First, most comparisons are identical-twin simulations and support only mechanism and ablation claims. Second, both the structural checks and closed-loop experiments use one basin, so the general tree results have not been accompanied by multi-basin empirical validation. Third, sensing, forcing, and routing mismatch can degrade both detection and localization; the Kerrville replay is only a limited check, and precise localization under model error in poorly gauged sectors remains open.

4.7 Conclusion

Tree structure restores a solvable reduced-input problem when sparse observations violate the full-input rank condition. The resulting triangular estimator is defined for every nonempty observation set, but it recovers an equivalent input at the observed frontier rather than an unobserved source node. The sector theorem states the remaining ambiguity: a fixed observation set identifies at most one sector per observation.

Mobile sensors address that ambiguity by changing the frontier. A belief-driven Lloyd controller can cover either posterior mass or posterior uncertainty, and the two-flood simulations show why the latter is needed after a source has been resolved. The bisection rule offers a tree-specific heuristic but no inherited optimality guarantee. Chapter 5 studies the separate question of what coverage guarantees remain meaningful when a continuous domain is represented by a fine finite grid.

CHAPTER 5

Discretisation-Induced Curvature Inflation in Submodular Coverage

This chapter is adapted from a manuscript currently under review at Transactions on Machine Learning Research [119].

5.1 Overview

At the team scale of this dissertation, a fleet of sensing agents must choose K locations that jointly cover an unknown spatial event density, such as an environmental field or the flood-prone river reaches studied in Chapter 4. Many guarantees for this problem depend on the curvature of a submodular coverage objective [23, 117, 8, 24]. In continuous coverage problems, however, the finite objective is produced by discretising the domain onto a candidate grid V_h of spacing h. The guarantee is therefore affected not only by the sensing problem but also by the chosen grid resolution.

Refining the grid creates near-duplicate candidates inside the same sensing footprint. After conditioning on the rest of the grid, the marginal contribution of any one candidate becomes small, so the full total curvature $c(f_h)$ approaches its worst-case value 1. This does not mean that the underlying placement problem becomes harder or that the achieved coverage becomes poor. It means that a certificate based on $1 - c(f_h)$ can become a description of grid redundancy rather than solution quality. Figure 5.1 illustrates this separation: the curvature factor collapses by many orders of magnitude, while greedy coverage remains within roughly

1–2% of the optimum.

This chapter establishes the leading-order collapse law for finite-grid noisy-or coverage,

$$-\log\big(1 - c(f_h)\big) = \kappa_g (r/h)^d + o\big((r/h)^d\big),$$

where r is the sensing radius, d is the domain dimension, and κ_g depends only on the sensing kernel. It then explains why this limit has different consequences for two common certificate forms: multiplicative sandwich certificates vanish with $1 - c(f_h)$, whereas greedy-envelope certificates retain the classical $1 - 1/e$ floor. The rate also gives a simple diagnostic for deciding when a full-curvature certificate has become dominated by discretisation.

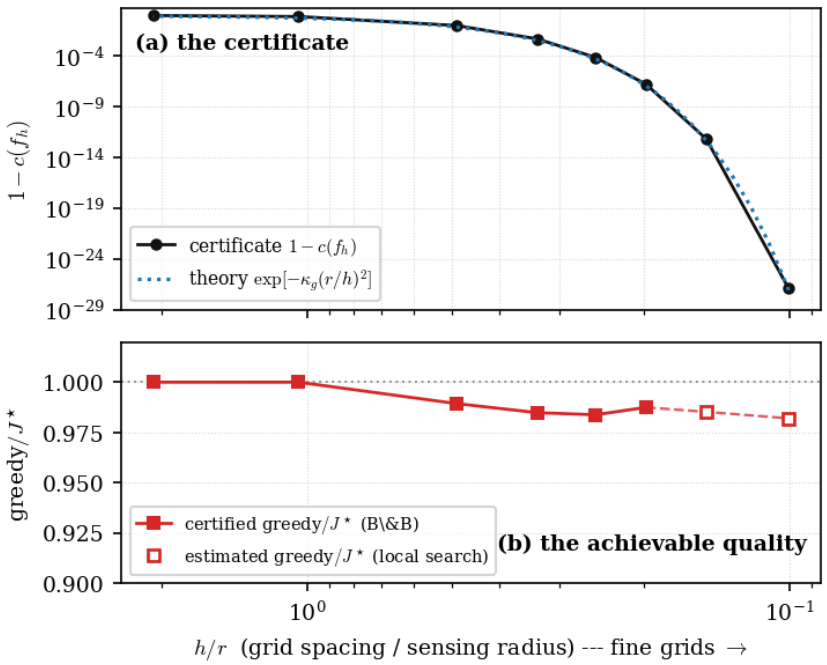


Figure 5.1: Grid refinement on a representative generated obstacle map. As h/r decreases, **(a)** the curvature factor $1 - c(f_h)$ follows the area-exponential rate and becomes tiny, while **(b)** greedy coverage remains within roughly 1–2% of the branch-and-bound optimum on the certified grids.

5.2 Coverage and Curvature on a Grid

Let $\Omega \subseteq \mathbb{R}^d$ be the mission domain and let $V_h = h\mathbb{Z}^d \cap \Omega$ be a candidate grid with spacing h. A sensing kernel $g : [0, \infty) \to [0, a_{\max}]$, with $a_{\max} < 1$, gives a candidate at $p_j \in V_h$ the response $g_j(q) = g(\|q - p_j\|/r)$, where r is the sensing radius. For a nonnegative importance density φ, the noisy-or value of a selected set $S \subseteq V_h$ is

$$f_h(S) = \int_\Omega \varphi(q) \left[1 - \prod_{p_k \in S} \big(1 - g_k(q)\big)\right] dq. \tag{5.1}$$

The function f_h is monotone and submodular [118]. Its full total curvature is

$$c(f_h) = 1 - \min_{j:\, f_h(\{j\})>0} \frac{f_h(V_h) - f_h(V_h \setminus \{j\})}{f_h(\{j\})} = 1 - \min_j \rho_j(h). \tag{5.2}$$

Here $\rho_j(h)$ is the fraction of candidate j's singleton value that remains marginal after every other candidate has already been included. On a fine grid, overlapping footprints make this fraction small.

The marginal ratio has an exact survival-product form. Expanding (5.1) before and after candidate j is added gives

$$\rho_j(h) = \frac{\int_\Omega \varphi(q) g_j(q) \prod_{k \neq j} \big(1 - g_k(q)\big)\, dq}{\int_\Omega \varphi(q) g_j(q)\, dq}. \tag{5.3}$$

The numerator is candidate j's response weighted by the probability that every other candidate fails to cover the same point. Refinement increases the number of factors associated with sensors whose footprints overlap that of j. Each factor lies in $(0, 1]$, so their product can be extremely small even though the union coverage of a feasible K-element set changes little. This distinction between conditioning on the full grid in (5.2) and selecting only K locations is why full curvature is especially sensitive to redundant candidates.

The curvature enters performance analyses in more than one way. If $m_h^+(S) = \sum_{j \in S} f_h(\{j\})$ is the modular singleton upper surrogate, submodularity gives the sandwich relation

$$f_h(S) \geq \big(1 - c(f_h)\big) m_h^+(S). \tag{5.4}$$

Any policy certificate whose curvature dependence comes through this relation inherits the multiplicative factor $1 - c(f_h)$. By contrast, a curvature-aware greedy envelope has the form [23, 87]

$$\beta(c, K) = \frac{1}{c}\left[1 - \Big(1 - \frac{c}{K}\Big)^K\right]. \tag{5.5}$$

These expressions have different limits as the grid is refined, which is the main practical consequence of the scaling result below.

5.3 Area-Exponential Curvature Inflation

Write $\psi(z) = \log(1 - g(\|z\|))$. For a dense periodic grid, the logarithm of the survival product in the marginal contribution is a lattice sum. Its leading term is the corresponding Riemann integral,

$$\sum_{p_k \in V_h} \psi\big((q - p_k)/r\big) \sim -\kappa_g (r/h)^d, \qquad \kappa_g = -\int_{\mathbb{R}^d} \log\big(1 - g(\|z\|)\big)\, dz.$$

The periodic domain removes boundary contributions so that the rate can be isolated from the shape of a finite mission region. Compact support makes the lattice sum local and Riemann integrable, while $a_{\max} < 1$ keeps ψ bounded. The positive density assumption ensures that every translated footprint has nonzero weight. Under these conditions the number of grid points in a sensing footprint grows on the order of $(r/h)^d$, and their log-survival contributions add. The next theorem makes this leading-order statement precise.

Theorem 2 (Area-exponential scaling, leading order)**.** *Let $g : [0, \infty) \to [0, a_{\max}]$ be continuous, compactly supported on $[0, 1]$, and satisfy $a_{\max} < 1$ and $\kappa_g > 0$. Consider the periodic torus $\Omega = [0, 1]^d$ with periodic square grids of mesh $h = 1/m$, a sensing radius $r < \frac{1}{2}$, and a continuous density $\varphi > 0$. Then*

$$\lim_{h/r \to 0} \frac{-\log\big(1 - c(f_h)\big)}{(r/h)^d} = \kappa_g.$$

Equivalently, $-\log(1 - c(f_h)) = \kappa_g (r/h)^d \big(1 + o(1)\big)$.

Proof sketch. Set $\delta = h/r$ and, for candidate j, define

$$L_{h,j}(q) = \sum_{k \neq j} \psi\big((q - p_k)/r\big).$$

Continuity and compact support make ψ uniformly continuous and Riemann integrable. The translated lattice sums therefore converge uniformly in their shift:

$$\sup_q \left| \delta^d \sum_k \psi\big((q - p_k)/r\big) + \kappa_g \right| \longrightarrow 0.$$

Removing the single self term changes the sum by at most $\|\psi\|_\infty = O(1)$. Uniformly in q and j, it follows that

$$L_{h,j}(q) = -\kappa_g \delta^{-d} + o\big(\big(\delta^{-d}\big)\big).$$

In the exact ratio (5.3), the survival product equals $\exp(L_{h,j}(q))$. The uniform estimate can therefore be taken through the positive weighted average, giving $\rho_j(h) = \exp[-\kappa_g \delta^{-d} + o\big(\big(\delta^{-d}\big)\big)]$ uniformly in j. Taking the minimum over candidates preserves the exponent, and (5.2) then gives the stated limit. □

The theorem concerns full total curvature, not the quality of a particular placement. Substituting $c(f_h) \to 1$ into (5.4) sends its multiplicative factor to zero. In contrast,

$$\lim_{c \to 1} \beta(c, K) = 1 - \left(1 - \frac{1}{K}\right)^K \geq 1 - \frac{1}{e}.$$

The floor follows directly from the finite geometric sum. With $u = c/K$,

$$\beta(c, K) = \frac{1}{K} \sum_{i=0}^{K-1} (1 - u)^i.$$

Every term is non-increasing in c, so the envelope is minimized at $c = 1$ and remains at least $1 - (1 - 1/K)^K$. A greedy-envelope certificate therefore loses any improvement obtained from low curvature but remains non-vacuous. A full-curvature sandwich certificate has no such floor: the same limit $c(f_h) \to 1$ sends (5.4) to zero. Curvature inflation consequently does not by itself imply that greedy selection or another coverage method performs poorly; the consequence depends on how curvature appears in the guarantee.

5.4 Empirical Check and Resolution Diagnostic

The representative experiments use the compact soft-disk kernel $g(s) = 0.5(1 - s^2)_+^2$, for which the theoretical two-dimensional rate is $\kappa_g = 0.62909$. Fitting $\log(1 - c(f_h))$ over a refinement sweep gives 0.62911. On the gorilla-1 obstacle map in Figure 5.1, decreasing h/r from 2.08 to 0.10 drives $1 - c(f_h)$ from 1 to 1.15×10^{-27}, while branch-and-bound comparisons show greedy coverage remaining within roughly 1–2% of the optimum on the grids where the optimum is certified. The curvature factor and the achieved quality therefore separate by about twenty-five orders of magnitude. Additional kernel, boundary, lattice, and line-of-sight checks are reported in the manuscript [119] rather than repeated here.

The scaling law can be used as a resolution diagnostic. For a tolerance τ below which a factor $1 - c(f_h)$ is regarded as uninformative, the leading-order collapse threshold satisfies

$$\frac{h}{r} \approx \left(\frac{\kappa_g}{-\log \tau} \right)^{1/d}. \tag{5.6}$$

In practice, one can estimate the slope on grids that are still computationally manageable and use it to identify the resolution at which further refinement mainly weakens the certificate. Refinement may still be desirable for modelling or placement accuracy; beyond that threshold, however, $1 - c(f_h)$ should not be interpreted as a measure of attainable coverage quality.

5.5 Discussion

Full curvature therefore measures a discretisation effect, not placement quality. The theorem covers smooth, compactly supported kernels on periodic grids; finite boundaries, obstacles, line-of-sight occlusion, and a resolution-independent alternative remain open.

The same caution applies to Chapter 4: thousands of river reaches form a fine spatial grid, and its event density is posterior-driven. We therefore report sensing and localization performance directly rather than use a full-curvature factor as evidence of deployment quality.

Part II

Population Systems: Equilibrium Learning under Endogenous Uncertainty

In the population regime, agents are numerous and interchangeable. Their strategies generate the population distribution, and that distribution affects each agent's reward and dynamics. The uncertainty is therefore endogenous. *Policy updates change the population distribution, so estimation alone is insufficient. We instead seek a policy that is optimal for the population distribution generated by that policy. This part contains the principal contributions of the dissertation. Chapter 6 introduces population-aware online mirror descent, which learns master policies that reach Nash equilibrium from population configurations outside the training set. Chapter 7 studies these policies under common noise, a shared shock that randomizes the mean-field evolution, and concludes with continuous state-action spaces.*

CHAPTER 6

Population-Aware Online Mirror Descent: Learning Master Policies for Mean-Field Games

This chapter is adapted from the AAMAS 2024 extended abstract and its full arXiv version: Wu, Laurière, Chua, Geist, Pietquin, and Mehta, "Population-aware Online Mirror Descent for Mean-Field Games by Deep Reinforcement Learning" [120, 121].

6.1 Overview

In Part I, uncertainty is exogenous: the unknown quantities are generated outside the team's decisions and observed through networked sensors. The estimation and coverage guarantees in that part rely on this externality. In a mean-field game (MFG) [70, 54], the population distribution μ is instead generated by the agents' policies and also affects their rewards and dynamics. Estimation alone is therefore insufficient. We seek a policy that is optimal for the population distribution it generates, which is a mean-field equilibrium. Fixed-point iterations converge under strict contraction conditions [42, 75, 3], while Fictitious Play (FP) and Online Mirror Descent (OMD) converge under Lasry–Lions monotonicity [99, 97]. These guarantees, and nearly all associated deep reinforcement learning (DRL) implementations, are stated for a *single, fixed initial distribution* μ_0. The computed policy depends only on the individual state, with population dependence represented implicitly through time, and need not remain an equilibrium from a different initial distribution.

This chapter removes that assumption. Following the master-equation viewpoint of

Lions [13], which studies value functions as functions of the population distribution, we seek a *master policy* [98]: a population-dependent policy that maps the pair (individual state, current population distribution) to actions and is an equilibrium policy from *any* initial distribution. The only prior learning method for master policies, Master Fictitious Play (M-FP) [98], inherits the intrinsic limitations of FP-type schemes: it must compute a best response at every iteration, and it averages uniformly over all past iterations, so its effective learning rate decays and progress stalls after moderately many iterations.

This chapter shows that master policies can be learned faster, and with far smaller models, by extending OMD rather than FP to the population-dependent setting. We introduce *Master OMD* (M-OMD), a DRL algorithm that evaluates—rather than optimizes against—the current policy at each iteration, keeps a constant update rate, and aggregates past Q-functions implicitly through the Munchausen reformulation [115], avoiding both policy averaging and the storage of historical networks. The contributions are the following.

- We formulate the learning of master policies for finite-horizon MFGs with variable initial distributions, and extend the OMD equilibrium-learning scheme of [97] to population-dependent policies and Q-functions.
- We prove (Theorem 3) that in the population-dependent setting, a single regularized Q-function trained with a Munchausen-style target is equivalent to the explicit sum of the historical Q-functions required by exact OMD. This equivalence allows the implementation to store one Q-network.
- We use one *inner-loop* replay buffer for all training distributions within an OMD iteration and reset it between iterations. The ablation in Appendix A shows less forgetting with this design.
- We complement the algorithm with a convergence analysis of an idealized version of its dynamics (Section 6.3.4; proofs in Appendix C). For tabular policies updated synchronously over all population distributions, with population-independent transitions

and a strongly monotone separable reward, terminal values converge at rate $\tau \log |\mathcal{A}|/t$, and values and exploitability converge uniformly over initial distributions under an envelope hypothesis that we prove for tie-free equilibria and, at the last decision level, for the congestion-type interactions of our benchmarks; without strong monotonicity, last-iterate convergence provably fails.

- In numerical experiments on four canonical MFG benchmarks (exploration in one room and in four connected rooms, beach bar, and linear-quadratic), M-OMD converges faster than four baselines including the state-of-the-art M-FP, generalizes to initial distributions never seen in training, and does so with a model whose size stays constant over iterations—orders of magnitude smaller than the growing policy libraries of FP-type methods. Ablations and scaling studies, reported in Appendix A, isolate the contribution of the population input, the inner-loop buffer, and the network architecture, and vary the map size and the number of training distributions.

The remainder of the chapter is organized as follows. Section 6.2 states the finite-horizon MFG setting, the classes of policies, and the exploitability metric, and reviews the solution schemes this work builds on. Section 6.3 develops the M-OMD algorithm: the population-dependent Q-function update, the equivalence theorem (proved in Appendix C), and the inner-loop replay buffer; it closes with a convergence analysis of the idealized master OMD dynamics (Section 6.3.4). Section 6.4 reports the main experimental comparisons and summarizes the ablations and scaling studies, whose full details are in Appendix A. Section 6.5 discusses the findings, and Section 6.6 concludes and sets up the common-noise extension of the next chapter.

6.2 Problem Formulation

We consider a multi-agent decision-making problem in the framework of discrete-time MFGs, which relies on the classical notion of Markov decision processes (MDPs). **Notations:** For

any finite set E, we denote by Δ_E the simplex of probability distributions on E. For any integer N, we denote $[N] = \{0, 1, \ldots, N\}$. Functions of time are seen as sequences and denoted by bold letters.

MDP for a representative agent. The state space $\mathcal{X}$ and action space $\mathcal{A}$ are finite, with a time horizon N_T. At each step $n \in [N_T]$, the mean field term (the population's state distribution) is $\mu_n \in \Delta_{\mathcal{X}}$. When an agent in state x_n takes action a_n, the next state is determined by the transition probability $p_n(\cdot \mid x_n, a_n, \mu_n)$, and the reward is $r_n(x_n, a_n, \mu_n)$. We focus on population-independent transitions, though our algorithm can handle population-dependent cases. The mean field sequence is denoted as $\boldsymbol{\mu} = (\mu_n)_{n\in[N_T]}$. Given this sequence, the agent aims to maximize cumulative rewards up to N_T. Unlike most of the MFG literature, the initial distribution μ_0 will be *variable*. This motivates the following class of policies. Throughout this chapter all randomness is *idiosyncratic*—it perturbs each agent independently and averages out at the population scale; shared (common) noise that shocks the whole population at once is deferred to Chapter 7.

Classes of policies. In most of the MFG literature one considers policies $\boldsymbol{\pi} = (\pi_n)_{n\in[N_T]}$ with $\pi_n : \mathcal{X} \to \Delta_{\mathcal{A}}$: the agent looks only at its own state, and the dependence on the population is implicit, through time. Such population-independent policies are meaningful when μ_0 is known and fixed in advance. When the initial distribution is not known in advance, the agent's decision should depend on the current distribution as well. We therefore use *population-dependent* policies, also called **master policies** [98]: $\boldsymbol{\pi} = (\pi_n)_{n\in[N_T]}$ with $\pi_n : \mathcal{X} \times \Delta_{\mathcal{X}} \to \Delta_{\mathcal{A}}$. If at time n the player is in state x_n and the mean field is μ_n, the player picks an action $a_n \sim \pi_n(\cdot|x_n, \mu_n)$. Note that the policies are *non-stationary*: in finite-horizon problems the remaining time is decision-relevant—the less time is left, the more urgent it becomes for agents to act [95]—so the timestep n is an essential input, in contrast with the stationary settings of, e.g., [2, 111, 71].

Best response. From the perspective of a single agent, if the mean field sequence $\boldsymbol{\mu} = (\mu_n)_{n\in[N_T]}$ is given, the total reward function to maximize is defined as:

$$J(\boldsymbol{\pi};\boldsymbol{\mu}) = \mathbb{E}\Big[\sum_{n\in[N_T]} r_n\left(x_n, a_n, \mu_n\right)\Big],$$

subject to the dynamics $x_0 \sim \mu_0$, $x_{n+1} \sim p_n\left(\cdot \mid x_n, a_n, \mu_n\right)$, $a_n \sim \pi_n\left(\cdot \mid x_n, \mu_n\right)$, for $n \in [N_T - 1]$. A policy $\boldsymbol{\pi}$ is a **best response** against $\boldsymbol{\mu}$ if it maximizes J, i.e., $\boldsymbol{\pi} \in BR(\boldsymbol{\mu}) := \arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi};\boldsymbol{\mu})$.

Mean field. We denote by $\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}} = MF_{\mu_0}(\boldsymbol{\pi})$ the **mean field generated by** policy $\boldsymbol{\pi}$ when starting from μ_0. It is interpreted as the sequence of population distributions obtained when the population starts with μ_0 and every player uses $\boldsymbol{\pi}$. It is defined as: for $n \in [N_T - 1]$,

$$\mu_{n+1}^{\mu_0,\boldsymbol{\pi}}(x') = \sum_{x,a} \mu_n^{\mu_0,\boldsymbol{\pi}}(x)\pi_n(a|x,\mu_n^{\mu_0,\boldsymbol{\pi}})p_n\left(x' \mid x, a, \mu_n^{\mu_0,\boldsymbol{\pi}}\right). \tag{6.1}$$

Nash equilibrium and master policies. For a given initial μ_0, a pair $(\boldsymbol{\pi},\boldsymbol{\mu})$ is a **(mean-field) Nash equilibrium (MFNE)** if $\boldsymbol{\pi}$ is a best response to $\boldsymbol{\mu}$, and $\boldsymbol{\mu}$ is generated by $\boldsymbol{\pi}$ starting from μ_0. Mathematically, $\boldsymbol{\pi} \in BR(\boldsymbol{\mu})$ and $\boldsymbol{\mu} = \boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}} = MF_{\mu_0}(\boldsymbol{\pi})$. $\boldsymbol{\pi}$ is an equilibrium policy for a given initial μ_0 if $\boldsymbol{\pi} \in BR(\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}})$, which means $\boldsymbol{\pi}$ is a fixed point of $BR \circ MF_{\mu_0}$. A population-dependent policy is called a **master policy** if it is an equilibrium policy for any initial distribution μ_0, i.e., $\boldsymbol{\pi} \in \cap_{\mu_0\in\Delta_{\mathcal{X}}} BR(\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}})$. If the players use $\boldsymbol{\pi}$, then for any μ_0, the population is in an MFNE: the game can start from any distribution and the players will always play according to a Nash equilibrium, without needing to learn a new policy.

Exploitability. In general, we cannot measure the distance between a given policy and the equilibrium one since the latter is unknown. Exploitability measures how far a policy is from being a Nash equilibrium [78, 99] by quantifying the maximum benefit a player can achieve by deviating from the policy used by the population:

$$\mathcal{E}^{\mu_0}(\boldsymbol{\pi}) = \mathbb{E}_{\mu_0}\left[\sup_{\boldsymbol{\pi}'} J\left(\boldsymbol{\pi}';\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}}\right) - J\left(\boldsymbol{\pi};\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}}\right)\right]. \tag{6.2}$$

A policy $\boldsymbol{\pi}$ is a Nash equilibrium policy for μ_0 if and only if $\mathcal{E}^{\mu_0}(\boldsymbol{\pi}) = 0$. Since computing the exact supremum in (6.2) is not always affordable, a notion of *approximate* exploitability—which takes the supremum only over previously computed policies—has been used to track learning progress; however, it does not capture the proximity to the genuine best response, especially when comparing different baselines. To enable fair comparisons among algorithms, all results in this chapter report the exact exploitability (6.2).

Solution schemes and their limitations. The most basic learning scheme for MFGs iterates the map $BR \circ MF_{\mu_0}$ directly; convergence of such Banach–Picard iterations requires a strict contraction condition [42, 75], i.e., Lipschitz continuity with sufficiently small constants, which often fails to hold [26, 3]. Fictitious Play [10, 104, 7], extended to MFGs in [14, 45, 99], smooths the iteration by averaging historical distributions; its convergence can be proved under Lasry–Lions monotonicity instead of contraction, and a deep variant exists [72]. But FP must compute a best response at each iteration—costly for complex problems—and averages uniformly over all past iterations, so its update rate decays as $1/k$. Online Mirror Descent [43, 44, 97] avoids both issues: each iteration only *evaluates* the current policy, and the update rate is constant; it aggregates past Q-functions instead of past policies, and converges under the same monotonicity assumption. All of these methods, however, were formulated for population-independent policies and a single initial distribution. The master policies of [98] lift the single-μ_0 restriction but are trained by FP, inheriting its cost and decaying rate. The algorithm of this chapter combines the strengths of the two lines: OMD-type updates for population-dependent policies.

Challenges and proposed approach. To learn a master policy, there are three main challenges. First, the policy *takes as input a population distribution*, which is a possibly high-dimensional vector; we approximate it with a deep neural network (DNN). Second, the policy should be an *optimal policy for a rather complex MDP*; for this we use model-free reinforcement learning, which is more scalable than exact dynamic programming. Third,

the policy needs to *perform well on any population distribution*; to achieve this, we employ a training set composed of various initial distributions, mixed through a replay buffer as described below.

6.3 The Master OMD Algorithm

6.3.1 Online mirror descent with population-dependent Q-functions

Our approach builds upon the OMD algorithm, introduced for MFGs with population-independent policies in [97] and inspired by the Mirror Descent MPI algorithm [114] in the single-agent case. At each iteration k, the policy $\boldsymbol{\pi}^{k-1}$ from the previous iteration is known. We first compute the mean field $\boldsymbol{\mu}^{k-1} = \boldsymbol{\mu}^{\boldsymbol{\pi}^{k-1}}$ by forward induction (6.1). Second, we evaluate $\boldsymbol{\pi}^{k-1}$ by computing its Q-function $Q^k = Q^{\boldsymbol{\pi}^{k-1}}$; note that this is a policy *evaluation*, not a best-response computation. Then, we compute the cumulative Q-function $\bar{Q}_n^k(x,a) = \sum_{i=0}^{k} Q_n^i(x,a)$. Finally, the new policy is computed as: $\pi_n^k(\cdot|x) = \text{softmax}\left(\frac{1}{\tau}\bar{Q}_n^k(x,\cdot)\right)$, where $\tau > 0$ is a temperature parameter. Since the policy depends only on the individual state and $\mathcal{X}$ is finite, a tabular implementation was possible in [97]. Here, however, our goal is to learn a master policy, which is *population-dependent*. Since $\mu \in \Delta_{\mathcal{X}}$ takes a continuum of values, it is not possible to represent Q^k exactly, and we resort to function approximation with DNNs. Classically, learning the Q-function associated with a policy can be done using Monte Carlo samples. Nevertheless, computing the *sum* of neural-network Q-functions would be challenging because DNNs are non-linear approximators. To tackle this challenge, we use the Munchausen trick, introduced in [115] and adapted to the MFG setting in [72]. Although the latter reference also uses an OMD-type algorithm, it does not handle population-dependent policies and Q-functions, which is the major difficulty addressed here, through a suitable use of an initial-distribution training set and of the replay buffer.

Before turning to the deep implementation, we make explicit why the cumulative-Q update is mirror descent. The variational identity (C.1), established in the proof of Theorem 3 in

Appendix C, says that the softmax policy $\pi_n^k = \text{softmax}\left(\frac{1}{\tau}\bar{Q}_n^k\right)$ is the unique maximizer of $\langle \pi, Q_n^k \rangle - \tau\,\text{KL}\big(\pi \,\|\, \pi_n^{k-1}\big)$ over the simplex. Each iteration therefore moves the policy toward the current evaluation Q^k while a KL penalty holds it near the previous policy. That is one step of mirror descent, with the Bregman distance induced by the negentropy. Unrolling the recursion gives the equivalent follow-the-regularized-leader form: π_n^k maximizes $\langle \pi, \bar{Q}_n^k \rangle - \tau\langle \pi, \ln \pi \rangle$, a best response to the sum of all past evaluations, softened by an entropy term of weight τ.

OMD accumulates Q-functions rather than policies because the Q-values provide the update directions in the dual space. Up to the state-occupancy weight, each evaluation Q^i is a gradient of the expected total reward with respect to the policy at (n, x, μ). The softmax map at temperature τ converts the cumulative Q-values into the next policy. This update keeps a constant rate across iterations, unlike the $1/k$ policy averaging of FP, and the Munchausen reformulation below represents the cumulative Q-values with one network. Section 6.3.4 analyzes an idealized tabular version of this update under strong Lasry–Lions monotonicity.

6.3.2 Q-function update

To compute $\bar{Q}$, a naive implementation would keep copies of all past DNNs $(Q^i)_{i\in[k]}$, evaluate them, and sum the outputs; this would be extremely inefficient both in memory and computation. Instead, we define a regularized Q-function and establish Theorem 3 using the Munchausen trick [115]. It relies on computing a single Q-function that mimics the sum $\sum_{i=0}^{k-1} Q^i$ through implicit regularization by a Kullback–Leibler (KL) divergence between the new policy and the previous one. We derive, in our population-dependent context, the equivalence between the regularized Q-function and the summation of historical Q-values.

Theorem 3. *Let $\tau > 0$. Denote by $\boldsymbol{\pi}^{k-1}$ the softmax policy learned in iteration $k-1$, i.e., $\pi_n^{k-1}(\cdot|x,\mu) = \text{softmax}\left(\frac{1}{\tau}\sum_{i=0}^{k-1} Q_n^i(x,\mu,\cdot)\right)$, and by $Q^k = Q^{\boldsymbol{\pi}^{k-1}}$ the state-action value*

function in iteration k. Let $\widetilde{Q}^k = Q^k + \tau \ln \boldsymbol{\pi}^{k-1} : \mathbb{N} \times \mathcal{X} \times \Delta_{\mathcal{X}} \times \mathcal{A} \to \mathbb{R}$. *If* $\boldsymbol{\mu}^k$ *is generated by* $\boldsymbol{\pi}^{k-1}$ *in Algorithm 3, then for every* n, x, $\pi_n^k(\cdot|x, \mu_n^k) = \text{softmax}\left(\frac{1}{\tau}\widetilde{Q}_n^k(x, \mu_n^k, \cdot)\right)$.

Proof sketch. The full proof is in Appendix C (Section C.1); the argument has three steps. First, a Lagrangian computation on the simplex shows that the OMD policy $\text{softmax}\left(\frac{1}{\tau}\sum_{i=0}^k Q^i\right)$ is the unique maximizer of the KL-regularized objective $\langle \pi, Q^k \rangle - \tau\,\text{KL}(\pi \,\|\, \pi^{k-1})$. Second, the same computation shows that $\text{softmax}\left(\frac{1}{\tau}\widetilde{Q}^k\right)$ is the unique maximizer of the entropy-regularized objective $\langle \pi, \widetilde{Q}^k \rangle - \tau\langle \pi, \ln \pi \rangle$. Third, substituting $\widetilde{Q}^k = Q^k + \tau \ln \pi^{k-1}$ shows that the two objectives are identical, so their maximizers coincide; the Munchausen reformulation is exactly this equality. □

Remark 3. In the deep implementation the softmax is used directly rather than an exact arg max: with DNN-approximated Q-functions the greedy step of an arg max cannot be computed exactly, whereas the softmax contains it implicitly and benefits the convergence of the scheme [114].

Theorem 3 suggests the update rule to be used. At iteration k of our algorithm (see Algorithm 3) we train a deep Q-network $\tilde{Q}_\theta^k$ with parameters θ which takes as inputs the time step, the agent's state, the mean-field state, and the agent's action. This DNN is trained to minimize the loss: $\mathbb{E}\left|\tilde{Q}_\theta^k\left((n, x_n, \mu_n), a_n\right) - T_n\right|^2$, where the target T_n is:

$$T_n = r_n^k + L_n^k + \gamma \sum_{a_{n+1} \in \mathcal{A}} \pi_{\theta'}^k\left(a_{n+1} \mid s_{n+1}^k\right)\left[\tilde{Q}_{\theta'}^k\left(s_{n+1}^k, a_{n+1}\right) - L_{n+1}^k\right], \tag{6.3}$$

with $r_n^k = r(x_n, a_n, \mu_n^k)$, $s_n^k = (n, x_n, \mu_n^k)$, and $L_n^k = \tau \log(\pi_\theta^{k-1}\left(a_n \mid s_n^k\right))$, which is the main difference with a classical DQN target. Here $\pi_{\theta'}^k = \text{softmax}(\frac{1}{\tau}\tilde{Q}_{\theta'}^k)$ where $\tilde{Q}_{\theta'}^k$ is a target network with the same architecture but parameters θ', updated at a slower rate than $\tilde{Q}_\theta^k$. In the definition of T_n, the L^k terms (shown in blue) involve $\pi_\theta^{k-1} = \text{softmax}(\frac{1}{\tau}\tilde{Q}_\theta^{k-1})$, which performs the implicit averaging of Theorem 3. Differing from the Q-update proposed in [72], where the target policy is set as the policy learned from the *previous* iteration, our algorithm employs the target policy $\pi_{\theta'}^k$ and target Q-function $\tilde{Q}_{\theta'}^k$ being learned in the *current* iteration,

namely k instead of $k-1$. This modification helps stabilize the learning: if the target policy were fixed as a separate, older policy, the distribution of the policy under evaluation would differ from the one being learned, and this distribution shift would induce extra instability in the learning process.

6.3.3 Inner-loop replay buffer

Adding the population distribution to the network input and training on several initial distributions does not by itself make one Q-network work across all of them. We use experience replay [86, 106] to store transitions $(x_n, a_n, r_n, \mu_n, x_{n+1})$ from the different population flows. If the flows are presented separately, later training can overwrite what the network learned from earlier flows [38, 64].

We therefore share one replay buffer across all training distributions within an OMD iteration and reset it before the next iteration. Theorem 3 shows that the regularized target already includes the previous Q-functions, so transitions from earlier OMD iterations need not remain in the buffer. Step 2 of Algorithm 3 gives the update, and Figure A.2 in Appendix A reports the buffer ablation.

6.3.4 Convergence of the Idealized Master OMD Dynamics

Algorithm 3 is a deep RL method: it samples transitions, trains along the flows of a finite set of initial distributions, and shares one approximate Q-network across all population states. We do not prove convergence of this sampled, function-approximated implementation. Instead, we analyze an idealized continuous-time tabular update. This subsection states the idealization and the convergence result; the proofs are in Appendix C, where the gradient-aligned envelope theorem is stated with a proof outline. The analysis is part of this dissertation and builds on the fixed-μ_0 OMD theory of Pérolat et al. [97] and the master-policy formulation of Perrin et al. [98].

Algorithm 3 Master Deep Online Mirror Descent (M-OMD)

Input: Initial-distribution set $\mathcal{D}$; temperature τ; horizon N_T

Q-network parameters θ; target-network parameters θ'; iterations K; episodes N_{episodes}

Output: Master policy $\boldsymbol{\pi}^K$

Initialize $\boldsymbol{\pi}_\theta^0(\cdot \mid n, x, \mu) \leftarrow \text{softmax}(Q_\theta(n, x, \mu, \cdot)/\tau)$

for $k = 1, 2, \ldots, K$ **do**

Step 1: Mean-field propagation

  for all $\mu_0 \in \mathcal{D}$ **do**

    Compute $\boldsymbol{\mu}^{k,\mu_0}$ under $\boldsymbol{\pi}_\theta^{k-1}$ using Eq. (6.1)

  end for

Step 2: Replay-buffer reset

  Reset $\mathcal{M}_{\text{RL}}$

Step 3: Value-function update

  for $t = 1, 2, \ldots, N_{\text{episodes}}$ **do**

    for all $\mu_0 \in \mathcal{D}$ **do**

      for $n = 1, 2, \ldots, N_T$ **do**

        Select a_n by an ϵ-greedy rule with respect to $\widetilde{Q}_\theta^k$

        Execute a_n and store the transition in $\mathcal{M}_{\text{RL}}$

        Sample a minibatch from $\mathcal{M}_{\text{RL}}$ and update θ using Eq. (6.3)

        if a target-network update is due **then**

          Update θ'

        end if

      end for

    end for

  end for

Step 4: Policy update

  $\boldsymbol{\pi}^k(\cdot \mid n, x, \mu) \leftarrow \text{softmax}\Big(\widetilde{Q}_\theta^k(n, x, \mu, \cdot)/\tau\Big)$

end for

return $\boldsymbol{\pi}^K$

Assumption 3. (i) $\mathcal{X}$ and $\mathcal{A}$ are finite with $|\mathcal{A}| \geq 2$, the horizon is N_T, and policies are tabular: one probability vector $\pi_n(\cdot|x,\mu) \in \Delta_{\mathcal{A}}$ for every level n, state x, and mean field $\mu \in \Delta_{\mathcal{X}}$. (ii) Transitions $p(x' \mid x, a)$ do not depend on μ; the time index on p and r is suppressed to lighten notation. (iii) The reward is bounded and separable, $r(x,a,\mu) = c(x,a) + \bar{r}(x,\mu)$, and the interaction term $\bar{r}$ is L_r-Lipschitz in μ and λ-strongly Lasry–Lions monotone:

$$\sum_{x\in\mathcal{X}} \big(\mu(x) - \mu'(x)\big)\big(\bar{r}(x,\mu) - \bar{r}(x,\mu')\big) \;\leq\; -\lambda \,\|\mu - \mu'\|_2^2, \qquad \lambda > 0, \tag{6.4}$$

for all $\mu, \mu' \in \Delta_{\mathcal{X}}$. (iv) The mirror map is the negentropy at temperature $\tau > 0$: the policy attached to a table of scores $y(x,\mu,\cdot)$ is softmax $\big(y(x,\mu,\cdot)/\tau\big)$.

The idealized dynamics replaces the sampled, function-approximated updates of Algorithm 3 by a synchronous full sweep in continuous time. Write $\Phi(\mu,\beta)$ for the one-step propagation of μ under a decision rule β, as in (6.1), and let $Q_n^{\boldsymbol{\pi}}(x,\mu,a)$ be the master Q-function of $\boldsymbol{\pi}$, defined recursively as the reward at (x,a,μ) plus the value of $\boldsymbol{\pi}$ at the propagated pair (Appendix C states the recursion). Every cell (n,x,μ,a) accumulates it:

$$\dot{y}_{n,t}(x,\mu,a) = Q_n^{\boldsymbol{\pi}_t}(x,\mu,a), \qquad \pi_{n,t}(\cdot \mid x,\mu) = \text{softmax}\big(y_{n,t}(x,\mu,\cdot)/\tau\big), \qquad y_{n,0} = 0. \tag{6.5}$$

This is the continuous-time analog of the cumulative $\bar{Q}^k$ of Section 6.3.1, applied at all population distributions simultaneously. It idealizes Algorithm 3 in three ways: exact evaluation instead of sampling, one table entry per cell instead of a shared network, and no common noise. Under Assumption 3 the equilibrium flow of every continuation game is unique (Theorem 5), so the equilibrium master value $V_n^*(x,\mu)$ and the equilibrium one-step image $\nu_n^*(\mu)$ are well defined; $\nu_{n,t}(\mu) = \Phi\big(\mu, \pi_{n,t}(\cdot \mid \cdot, \mu)\big)$ denotes the image under the current policy, and $V_n^{\boldsymbol{\pi}_t}$ the value of the current policy.

Theorem 4. *Let Assumption 3 hold and let $(y_t, \boldsymbol{\pi}_t)$ solve* (6.5).

(a) At the terminal level, uniformly in (x,μ), $\big|V_{N_T}^{\boldsymbol{\pi}_t}(x,\mu) - V_{N_T}^*(x,\mu)\big| \leq \tau \log|\mathcal{A}| \,/\, t$.

(b) Suppose the envelope hypothesis (ENV) holds: at every level $n < N_T$ there is a deterministic $F_n(t) \downarrow 0$, independent of μ, and a finite threshold T_n with

$$\|\nu_{n,t}(\mu) - \nu_n^*(\mu)\|_2^2 \;\le\; F_n(t) \qquad \text{for all } \mu \in \Delta_{\mathcal{X}} \text{ and all } t \ge T_n. \tag{6.6}$$

Then values and master Q-functions converge uniformly over all (n, x, μ, a), the policies approach the equilibrium best-response faces, and $\sup_{\mu_0 \in \Delta_{\mathcal{X}}} \mathcal{E}^{\mu_0}(\boldsymbol{\pi}_t) \to 0$.

(c) (ENV) holds unconditionally if the equilibrium stage payoffs are tie-free, that is, if there is a unique maximizing action at every cell. It also holds at level $N_T - 1$ for gradient-aligned interactions, a class containing the own-congestion rewards $\bar{r}(x, \mu) = \theta_x(\mu(x))$ with θ_x strongly decreasing, and in particular Lipschitz-smoothed versions of the crowd-aversion terms used in the exploration and beach-bar environments of this chapter. If $|\mathcal{X}| = 2$ and $N_T \le 2$, the conclusion of (b) holds unconditionally, with rate $O\big((\log t)^{3/8}\, t^{-1/8}\big)$.

For weak monotonicity ($\lambda = 0$ in (6.4)), last-iterate convergence is false in general: a stage game with an antisymmetric interaction component keeps the OMD flow on a closed orbit (Appendix C).

Even without any envelope, the analysis gives, at level $N_T - 1$, convergence of the one-step images in time average at rate $O(\log t/t)$, uniformly in μ, together with last-iterate convergence along a density-one set of times for each fixed μ; hypothesis (ENV) is what supplies the uniform last-iterate control that the backward induction needs at every lower level.

Proof architecture. The proof is a backward induction over the horizon; Appendix C carries it out in five steps. First, the master value inherits the strong monotonicity of the interaction reward: $\langle \mu - \mu',\, V_n^*(\cdot, \mu) - V_n^*(\cdot, \mu') \rangle \le -\lambda \|\mu - \mu'\|_2^2$ at every level, with the same constant λ (Theorem 5). This propagation step is the crux; it is proved by a four-term doubling argument on a pair of equilibria, and it also yields uniqueness of the equilibrium

flow. Second, freezing a level n and a mean field μ collapses (6.5) at that cell into a one-shot stage game, perturbed only by the value error one level down; a collapse identity transfers the monotonicity of V^*_{n+1} to this stage game. Third, for the perturbed stage game the Fenchel coupling between the equilibrium and the dual scores is a Lyapunov function: its derivative is at most $-\lambda f(t) - P(t) + 2\varepsilon(t)$, where f is the squared image error, P the payoff gap, and ε the perturbation size. Fourth, backward induction closes the recursion: the perturbation at level n is exactly the value gap at level $n+1$, the terminal level is unperturbed and gives (a), and (ENV) converts the per-μ conclusions of the Lyapunov step into the μ-uniform envelope the next level needs. Fifth, the exploitability of $\boldsymbol{\pi}_t$ is bounded directly along its own realized flow via a regret decomposition, so no convergence of the flow itself, and no pointwise policy limit inside a tied best-response face, is required.

Scope. Theorem 4 is exactly as strong as stated. It concerns the idealized dynamics (6.5), not the deep implementation: sampling, training on finitely many flows, and function approximation are all outside its hypotheses. Common noise is also outside them; Chapter 7 states the conditional monotonicity condition that a common-noise extension would rest on. Within the theorem, values converge uniformly, while policies converge only to best-response faces, which are set-valued wherever the equilibrium payoff has ties; ties are the generic situation for mixed equilibria in congestion games, so pointwise policy convergence should not be read into (b). Finally, part (b) in full generality is conditional on (ENV); part (c) proves (ENV) in the tie-free and gradient-aligned cases, and we claim nothing beyond them.

6.4 Experimental Results

6.4.1 Setup

Environments. We consider four canonical examples drawn from three families of environments that are widely used as MFG benchmarks: exploration in one room, exploration in

four connected rooms, the beach bar, and a linear-quadratic model. We recall that in MFGs, solving a game requires identifying an equilibrium, which is harder than maximizing a reward, and that the mean-field evolution depends on the initial distribution. Each experiment explores two scenarios. The first, referred to as **fixed** μ_0, follows the common practice of starting the population from a fixed initial distribution. The second, **multiple** μ_0, tests the master policy by using several initial distributions simultaneously during training; the training and testing distributions are shown in Figure 6.1. Instead of training separate networks for different Nash equilibria, the master policy uses a single network to learn equilibrium policies for all initial distributions. Population-independent policies are expected to underperform in this scenario, except when the equilibrium policy is unchanged across initial distributions, which amounts to saying that there are no interactions.

Algorithms. We compare our algorithm to four baselines, including several state-of-the-art (SOTA) DRL algorithms for MFGs. In figures and tables, **vanilla FP (V-FP)** represents an adaptation of the tabular FP from [99] to DNNs. V-FP uses classic fictitious play to iteratively learn the Nash equilibrium, implicitly assuming agents always start from a fixed distribution. **Master FP (M-FP)**, from [98], handles any initial distribution via FP. **Vanilla OMD1 (V-OMD1)**, based on the Munchausen trick, is the deep OMD introduced in [72]. **Vanilla OMD2 (V-OMD2)** is our algorithm without the mean-field state as input, while the full version is called **Master OMD (M-OMD)**. M-FP and M-OMD learn population-dependent policies, while V-FP, V-OMD1, and V-OMD2 do not. In particular, V-OMD2 serves as an ablation of M-OMD, removing the distribution dependence to isolate its impact.

Implementation. FP-type algorithms use the DQN algorithm to iteratively learn the best response to the current mean-field sequence. In the model-free setting, transition probabilities and reward functions are unknown during both training and execution. Our Q-network follows the DQN architecture [86, 93]; OMD-type algorithms use a comparable Q-network for policy evaluation. Distributions are represented as histograms, flattened into one-dimensional

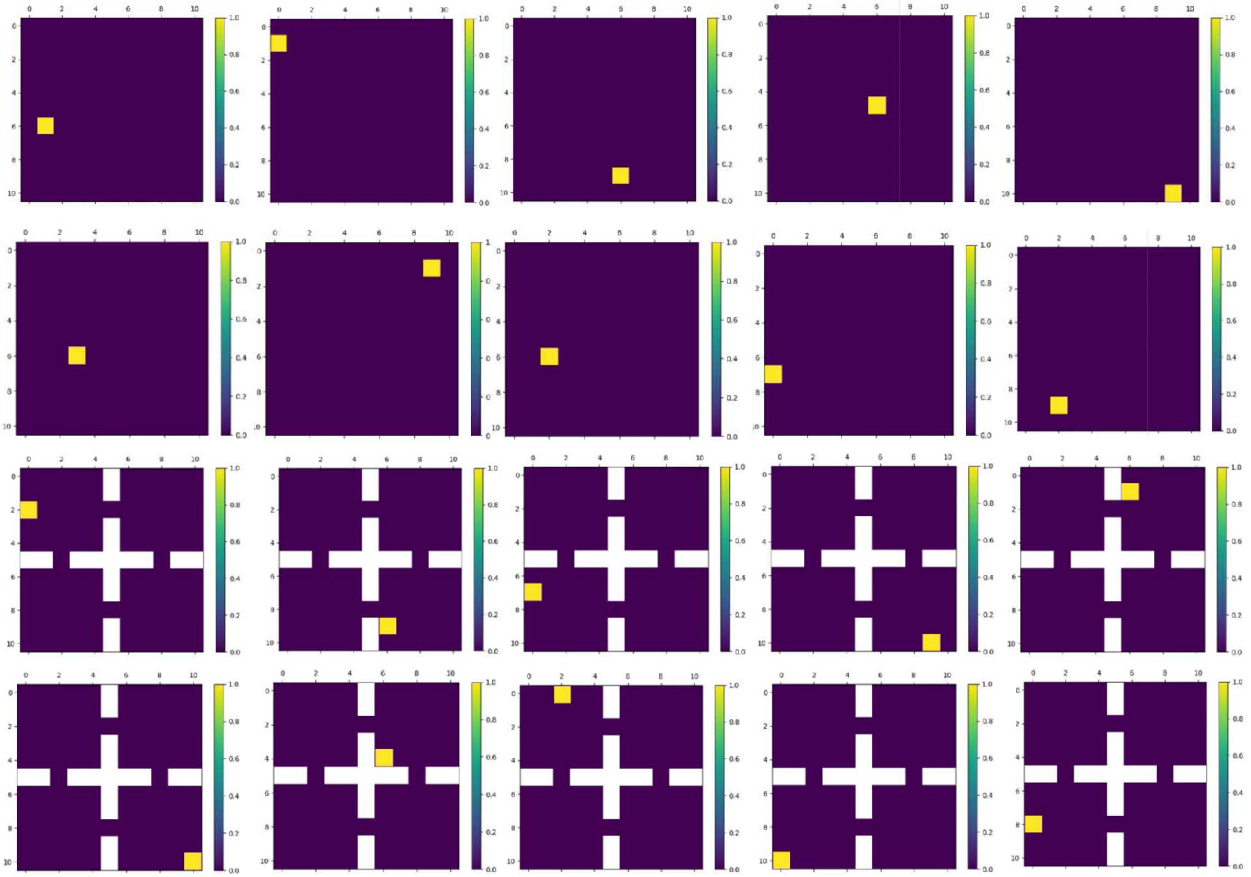


Figure 6.1: Training and testing sets of initial distributions used in the multiple-μ_0 scenario. From top to bottom: training set and testing set for the beach bar and one-room exploration tasks (empty map); training set and testing set for the four-connected-rooms exploration task (map with obstacles). Each set contains five initial distributions.

vectors (concatenated if needed) before being passed to the network. This approach works well for the examples studied here; for 2D cases, ConvNets could improve performance [98], though they were unnecessary in our experiments (see also the architecture study in Appendix A). Importantly, our method learns *non-stationary* policies, which are critical for finite-horizon problems: unlike in infinite-horizon settings [98], timesteps are essential here and are incorporated into the agent's policy using one-hot encoding. For each algorithm, we swept over hyperparameters and used the best parameters found; the chosen values and the full sweeps are reported in Appendix A.

The GPU used is an NVIDIA TITAN RTX (24 GB); the CPU is a 2x 16-core Intel Xeon Gold (64 virtual cores). The exploitability curves are averaged over 5 realizations of the algorithm, and whenever relevant, we show the standard deviation (std. dev.) with a shaded area.

6.4.2 Environment 1: exploration

Exploration is a classic problem in MFGs [35], in which a large group of agents tries to avoid crowded areas and hence to distribute uniformly into empty areas in a decentralized way. We consider two variants with different domain geometries: exploration in one room, and in four connected rooms, which is more challenging.

Example 1: exploration in one room. We consider a 2D grid world of dimension 11×11. The action set is $\mathcal{A}$={up, down, left, right, stay}. The dynamics are: $x_{n+1} = x_n + a_n + \epsilon_n$, where ϵ_n is an environment noise that perturbs each agent's movement (no perturbation w.p. 0.9, and one of the four directions w.p. 0.025 for each direction). The reward function discourages agents from being in a crowded location: $r(x, a, \mu) = -\log(\mu(x)) - \frac{1}{|\mathcal{X}|}|a|$.

Example 2: exploration in four connected rooms. The reward function and the dynamics are the same, but the environment consists of four connected rooms and an agent

does not move when it hits an obstacle. The goal is to explore every grid point within those rooms. The dimension of the whole map is again 11×11 (a comparative study with different map sizes is presented in Appendix A).

Figures 6.2 and 6.3 show the results for the two examples: heat-maps representing the evolution of the distribution (at several time steps) when using the master policy learned by our algorithm; the evolution of exploitability when using a fixed μ_0 or multiple μ_0 for a single run of the algorithm; and finally the results averaged over 5 runs. On both examples, our proposed algorithm (M-OMD) converges faster than all 4 baselines. With fixed μ_0, all methods perform well, but with multiple initial distributions, it appears clearly that V-FP, V-OMD1 and V-OMD2 fail to converge, due to the fact that vanilla policies lack awareness of the population, so agents cannot adjust their behavior suitably when the initial distribution varies. In other words, *population-independent policies cannot be Nash equilibria when testing on new initial distributions*, hence their exploitability is non-zero.

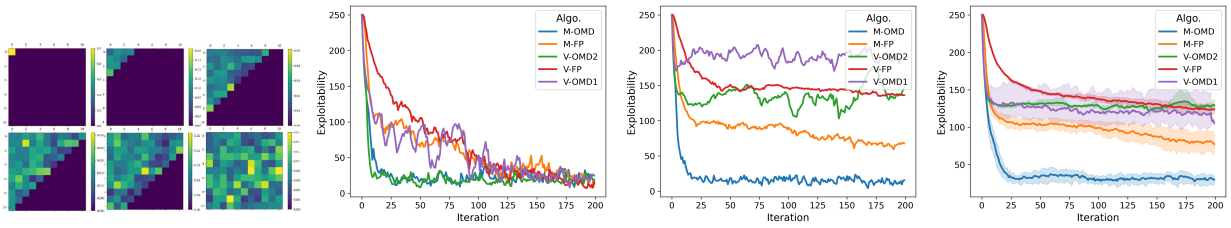


Figure 6.2: Example 1 in Env1: exploration in one room. From left to right: density evolution using the policy learned by M-OMD, starting from the μ_0 used for the second plot; exploitability vs training iteration for a single μ_0; average exploitability when training over 5 different μ_0 (single run of each algorithm); average ($\pm$ std. dev.) over 5 runs.

6.4.3 Environment 2: beach bar

The beach bar environment, introduced in [99], represents agents moving on a beach towards a bar. The goal for each agent is to avoid the crowd but get close to the bar. The dynamics are the same as in the exploration examples. Here we consider that the bar is located at the

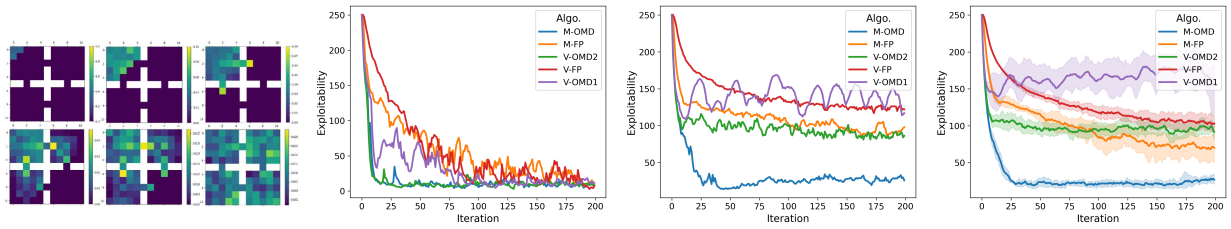


Figure 6.3: Example 2 in Env1: exploration in four connected rooms. From left to right: density evolution using the policy learned by M-OMD; exploitability vs training iteration for a single μ_0 (same as for the left-most plot); average exploitability when training over 5 different μ_0 (single run of each algorithm); average ($\pm$ std. dev.) over 5 runs.

center of the beach, and it is not possible to go beyond the domain. The reward function is: $r(x, a, \mu) = d_{bar}(x) - \frac{|a|}{|\mathcal{X}|} - \log(\mu(x))$, where d_{bar} accounts for the distance to the bar, the second term penalizes movement so the agent moves only when necessary, and the third term penalizes being in a crowded region. We consider $\mathcal{X}$ with 11 (1D) or 11×11 (2D) states.

The results are shown in Figure 6.4. The agents are always attracted towards the bar, whose location is independent of the population distribution, so the impact of starting from a fixed distribution or from various distributions is expected to be relatively minimal, which is what we observe numerically. In the 2D case, the distribution spreads while the agents move towards the bar to avoid large crowds, and then concentrates at the bar's location. The 1D case adds an extra difficulty: the bar closes at time $n = 20$, after which the population returns to a uniform distribution (due to crowd aversion); this shows that the policy is aware of time. On the exploitability curves for training over multiple μ_0, M-OMD again performs best.

6.4.4 Environment 3: linear-quadratic

The linear-quadratic (LQ) model is a classical setting studied extensively, e.g., in [16, 6]. A discretized version was introduced in [99]. It is a 1D model in which the dynamics are:

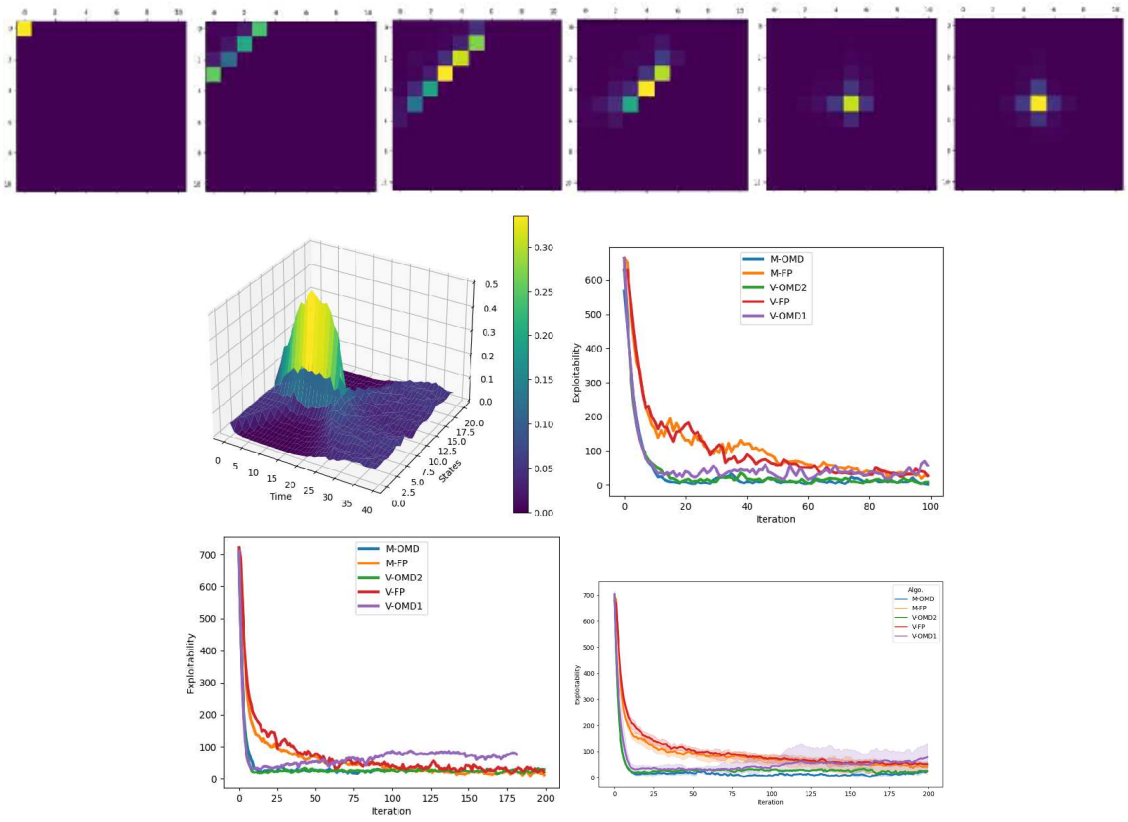


Figure 6.4: Beach bar problem (Env2). Top row: distribution evolution in the 2D case under the master policy learned by M-OMD (left to right in time), with the population spreading around the crowded bar before concentrating on it. Middle left: distribution evolution in the 1D case in which the bar closes at $n = 20$ and the population spreads back out, showing time awareness of the policy. Middle right: exploitability when training with fixed μ_0. Bottom row: exploitability when training with multiple μ_0 (single run), and the average ($\pm$ std. dev.) over 5 runs.

$x_{n+1} = x_n + a_n\Delta_n + \sigma\epsilon_n\sqrt{\Delta_n}$, with action set $\mathcal{A} = \{-M, \dots, M\}$, corresponding to moving by a number of states (left or right). The state space is $\mathcal{X} = \{-L, \dots, L\}$, of dimension $|\mathcal{X}| = 2L - 1$. To add stochasticity to this model, ϵ_n is an additional noise that perturbs the action choice, with $\epsilon_n \sim \mathcal{N}(0, 1)$ discretized over $\{-3\sigma, \dots, 3\sigma\}$. The reward function is:

$$r_n(x, a, \mu) = \Big[-\tfrac{1}{2}|a|^2 + q\,a\,(m - x) - \tfrac{\kappa}{2}(m - x)^2 \Big]\Delta_n,$$

where $m = \sum_{x\in\mathcal{X}} x\mu(x)$ is the first moment of the population distribution, encouraging agents to move toward the population's average position while maintaining dynamic movement. The terminal reward is $r_{N_T}(x, a, \mu) = -\frac{c_{\text{term}}}{2}(m - x)^2$. Here we used $\sigma = 1$, $N_T = 30$, $\Delta_n = 1$, $q = 0.01$, $\kappa = 0.5$, $K = 1$, $M = 3$, $L = 20$, $c_{term} = 1$.

Compared with the previous tasks, solving this LQ game is considerably easier and convergence occurs within a few iterations. Because M-OMD's regularized Bellman update deliberately damps rapid policy changes, on this easy problem it converges more slowly than FP: in Figure 6.5, V-FP and V-OMD1 converge faster than V-OMD2 and M-OMD. Under multiple-initial-distribution training, even vanilla FP and OMD drive exploitability to zero here, because the cheap movement cost lets population-independent policies steer all agents to a common target regardless of μ_0; finding LQ parameters that better expose population dependence is left to future work.

6.4.5 Generalization to the testing set

We conclude the main comparisons with Table 6.1, which reports the exploitability on the held-out testing distributions of Figure 6.1 after 200 training iterations. M-OMD consistently outperforms the SOTA baseline M-FP. It also performs better than the baselines learning population-independent policies, except in the LQ case, where we conjecture that the equilibrium policies are almost population-independent, at least in the setting studied in the literature. All algorithms exhibit higher exploitability on the testing set than on the training set at the final iteration; we attribute this to overfitting and to the limited number

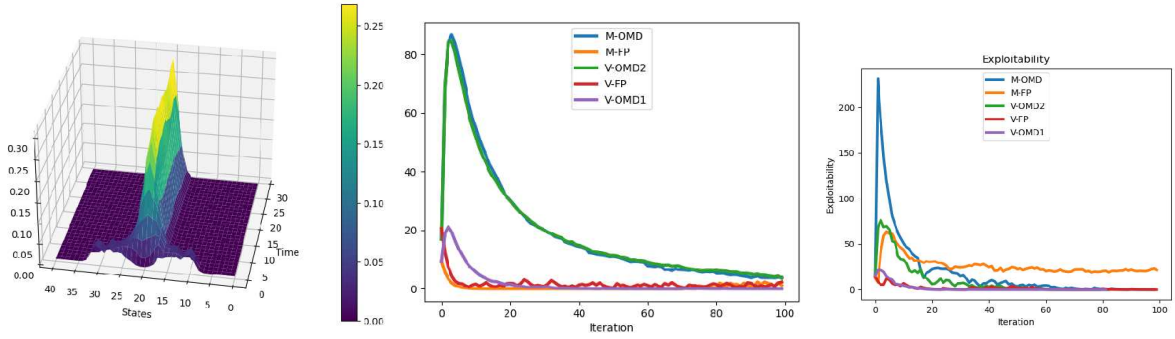


Figure 6.5: Linear-quadratic model (Env3). Left: evolution of the population under the policy learned by M-OMD, starting from a pair of Gaussian distributions and accumulating at the population's center. Middle and right: exploitability during training over one fixed initial distribution and over five initial distributions, respectively.

of training distributions—classic machine-learning challenges, which the 30-distribution study of Appendix A addresses in part—rather than to a limitation specific to the algorithm.

Env.	V-FP	M-FP	V-OMD1	V-OMD2	**M-OMD**
Exploration-1	135.32	84.76	175.91	163.42	**78.2**
Exploration-2	146.45	72.66	166.84	159.67	**60.0**
Beach Bar	83.24	40.12	80.81	61.71	**22.05**
LQ	0	22.78	0	0	**8.20**

Table 6.1: Exploitability on the testing set (200 training iterations).

Ablations and scaling studies

Appendix A reports the ablations and scaling studies in full; we summarize the findings here. The inner-loop replay buffer is essential: without it, transitions from the different training distributions are not mixed within an iteration, some distributions are forgotten, and the exploitability stagnates on both exploration tasks (Figure A.2). The results are robust across hyperparameters; the temperature $\tau = 50$ gave the fastest and most stable convergence in sweeps over $\{5, 10, 50, 100\}$ (Figure A.1). Thanks to the implicit summation of Theorem 3,

M-OMD stores a single Q-network, so its memory footprint and its exploitability-evaluation cost stay constant in the iteration count and grow mildly with the map size, whereas both grow linearly for FP-type methods (Figures A.3 and A.4). A study with 30 training distributions per exploration task confirms the advantage at scale: with identical 256×256 architectures, M-OMD outperforms M-FP in true exploitability on both the training and the testing sets (Figure A.5). Finally, preliminary ad-hoc teaming tests show that the master policy absorbs a moderate group of agents joining mid-execution (Figure A.7); a comprehensive study is left to future work.

6.5 Discussion

Numerical results. V-OMD2 differs from V-OMD1 [72] in three implementation choices. It does not reuse the old policy as both the behavior and target policy, clips the logarithmic Munchausen term at 10^{-6}, and uses the within-iteration replay buffer described above. In the reported experiments, M-OMD reduces exploitability faster than M-FP [98], including on the two-dimensional tasks without a convolutional network. Figures A.3 and A.4 in Appendix A report its memory use and exploitability-evaluation time. Table 6.1 reports lower exploitability than the baselines on the held-out initial distributions except in the LQ case, where the equilibrium policies appear nearly population-independent under the studied conditions. Figure A.7 reports a separate ad-hoc teaming test.

Related works. To the best of our knowledge, M-OMD is the first OMD-type deep RL method to handle unknown initial distributions in finite-horizon MFGs. Most works on RL for MFGs focus on population-independent policies, see e.g. [42, 26]. [72] combines FP and OMD with DRL but still for population-independent policies. [98] learns master policies but relies on FP, while we rely on OMD, which is faster as shown in our experiments.

6.6 Conclusion

This chapter presented the Master OMD (M-OMD) algorithm for efficiently computing population-dependent Nash equilibria in MFGs. In contrast with stationary MFGs and finite-horizon MFGs assuming a fixed initial distribution, this chapter addressed models in which the initial population is a priori unknown: by extending the Munchausen OMD update to population-aware Q-functions, justified by the equivalence result of Theorem 3, and by mixing training distributions through an inner-loop replay buffer, a single network learns policies that reach a Nash equilibrium from diverse initial distributions, faster and with far smaller models than the state-of-the-art master-policy baseline. For an idealized version of the dynamics, tabular, synchronous, and exactly evaluated, Section 6.3.4 established convergence under strong Lasry–Lions monotonicity (Theorem 4). What remains open on the theory side is the sampled, function-approximated implementation actually run in the experiments, the common-noise setting of the next chapter, and μ-uniform envelopes at every level for general strongly monotone games beyond the classes covered by Theorem 4(c). Extensions to other settings, such as multi-population MFGs, also remain for future work.

The master policy learned in this chapter has, so far, faced only idiosyncratic randomness: the environment noise ϵ_n perturbs each agent independently, so at the population scale it averages out and the mean-field flow generated by a policy from a given initial distribution is deterministic. Many monitoring scenarios of interest violate exactly this premise—a storm closes the beach, a flood shifts every evacuation route at once—and such a shared shock makes the mean-field flow itself stochastic, so that no single anticipated trajectory of $\boldsymbol{\mu}$ suffices. Chapter 7 confronts the master policy with this *common noise*: it conditions policies, Q-functions, and distributions on the observed noise history, and shows that the population-aware OMD machinery developed here extends to equilibria that react online to shocks affecting the entire population.

CHAPTER 7

Mean-Field Equilibria under Common Noise

This chapter is adapted from a paper presented at the 2025 IEEE 64th Conference on Decision and Control [122]. The closing section summarizes collaborator-led work on continuous state-action spaces [85], with a separate attribution stated there.

7.1 Overview

Chapter 6 assumes that all randomness is *idiosyncratic*. There, the Master OMD (M-OMD) algorithm learns *master policies*—policies conditioned on the population distribution—that reach mean-field Nash equilibrium from initial distributions never seen in training. Because idiosyncratic noise is independent across agents, it averages out in the mean-field limit. Once the initial distribution μ_0 and the policy are fixed, the mean-field flow $\boldsymbol{\mu}$ therefore evolves deterministically, and equilibrium is a fixed-point condition over one deterministic flow.

This chapter instead allows shocks that affect every agent and do not average out, such as weather over a crowd, a current carrying a school of fish, or a market-wide announcement. In the MFG literature, such a shared shock is called **common noise** [17]. Common noise makes the mean-field flow stochastic, so the population distribution is conditioned on the realized shock history and an equilibrium policy must remain a best response along each realization. It is an external disturbance that also changes the endogenous population distribution. In this chapter the shock is observed causally and supplied directly to the policy; it is not inferred from population observations.

We add the observed shock history to the inputs of the master policy and Q-function, and reveal the shock sequence only causally during training. The remaining components of M-OMD are unchanged. Section 7.2 defines MFGs with common noise and states the conditional monotonicity condition. Section 7.3 describes the modified network inputs. Section 7.4 evaluates the method on beach-bar and linear-quadratic (LQ) benchmarks with common noise. Section 7.6 summarizes collaborator-led work on deep-learning MFG solvers in continuous state-action spaces.

7.2 Mean-Field Games with Common Noise

Setting. The setting is that of Chapter 6, which is briefly recalled to fix notation; as there, Δ_E denotes the simplex of probability distributions on a finite set E, $[N] = \{0, 1, \dots, N\}$ for an integer N, and bold letters denote sequences in time. The state space $\mathcal{X}$ and action space $\mathcal{A}$ are finite, the horizon is N_T, and at each step $n \in [N_T]$ the mean field (population state distribution) is $\mu_n \in \Delta_{\mathcal{X}}$. An agent in state x_n taking action a_n transitions according to $p_n(\cdot \mid x_n, a_n, \mu_n)$ and receives reward $r_n(x_n, a_n, \mu_n)$. Policies are *population-dependent* (master) policies $\boldsymbol{\pi} = (\pi_n)_{n\in[N_T]}$ with $\pi_n : \mathcal{X} \times \Delta_{\mathcal{X}} \to \Delta_{\mathcal{A}}$ [98], the initial distribution μ_0 is variable, and $\boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}}$ denotes the mean-field flow generated by $\boldsymbol{\pi}$ from μ_0. A policy $\boldsymbol{\pi}$ is a best response to a flow $\boldsymbol{\mu}$ if it maximizes the total reward $J(\boldsymbol{\pi}; \boldsymbol{\mu})$, and a pair $(\boldsymbol{\pi}, \boldsymbol{\mu})$ is a mean-field Nash equilibrium if $\boldsymbol{\pi}$ is a best response to $\boldsymbol{\mu}$ while $\boldsymbol{\mu}$ is generated by $\boldsymbol{\pi}$. Distance to equilibrium is measured by the exploitability [78, 99],

$$\mathcal{E}^{\mu_0}(\boldsymbol{\pi}) = \mathbb{E}_{\mu_0}\Big[\sup_{\boldsymbol{\pi}'} J\big(\boldsymbol{\pi}'; \boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}}\big) - J\big(\boldsymbol{\pi}; \boldsymbol{\mu}^{\mu_0,\boldsymbol{\pi}}\big)\Big], \tag{7.1}$$

which vanishes exactly at Nash equilibrium policies for μ_0.

Common noise. In the context of MFGs, **common noise** (also called aggregate shock) refers to randomness that affects all players simultaneously [17]. It is denoted by $\Xi_N = \{\xi_n\}_{0\le n\le N} = \Xi_{N-1} \cdot \xi_N$, where $\cdot$ denotes concatenation of the shock history with the newest

shock, and it influences both transitions and rewards. Because every agent is struck by the same realization, the population distribution no longer evolves deterministically: it becomes a random measure conditioned on the shock history. Consequently, policies and population distributions are conditioned on this noise, written as $\pi_n(a \mid x, \mu_n, \Xi_n)$ and $\mu_n(x \mid \Xi_n)$, respectively. The Q-function of a representative player is defined by the backward recursion

$$\begin{cases} Q_{N_T}(x, \mu, a, \Xi_{N_T}) = r_{N_T}(x, \mu, a, \xi_{N_T}), \\ Q_n(x, \mu, a, \Xi_n) = r_n(x, \mu, a, \xi_n) + \mathbb{E}_{x', a', \xi_{n+1}}\big[Q_{n+1}(x', \mu', a', \Xi_{n+1})\big], \quad n \in [N_T - 1], \end{cases} \tag{7.2}$$

where $x' \sim p_n(\cdot \mid x, \mu, a, \xi_n)$, the successor mean field is propagated conditionally on the shock,

$$\mu'(x') = \sum_{x,a} \mu(x)\, \pi_n(a \mid x, \mu, \Xi_n)\, p_n\big(x' \mid x, a, \mu, \xi_n\big),$$

and $a' \sim \pi_{n+1}(\cdot \mid x', \mu', \Xi_{n+1})$. The expectation in (7.2) is taken over the next idiosyncratic transition, the next action, *and* the next shock ξ_{n+1}: unlike in Chapter 6, even a representative agent who knows μ_0 and the policy of the population cannot compute the future mean field, only its conditional law.

Equilibrium and convergence. Exploitability is defined as in (7.1), with policies and distributions conditioned on the common noise; a master policy for the common-noise game must drive it to zero for every initial distribution and along every noise realization. Without common noise, Section 6.3.4 establishes convergence of the idealized master OMD dynamics (Theorem 4) when the interaction reward is λ-strongly Lasry–Lions monotone, unconditionally at the terminal level and in special classes, and under an envelope hypothesis in general. Under common noise, the natural analog is the conditional form of the monotonicity condition,

$$\sum_{x \in \mathcal{X}} \big(\mu(x \mid \xi) - \mu'(x \mid \xi)\big)\big(\bar{r}(x, \mu, \xi) - \bar{r}\big(x, \mu', \xi\big)\big) \le 0, \tag{7.3}$$

where $\bar{r}$ is the reward of interaction with the population and both distributions are conditioned on the same shock realization. Condition (7.3) is the weak form; the extension sketched

here would rest on its strong version, with $-\lambda\|\mu - \mu'\|_2^2$ on the right-hand side and $\lambda > 0$, mirroring (6.4). Conditionally on a noise path, the population evolves by the same forward recursion as in the noise-free game, so the two ingredients of that proof, the propagation of monotonicity through the master value and the similarity-function (Fenchel-coupling) Lyapunov argument in the spirit of Pérolat et al. [97], are expected to apply branch by branch along the realized shocks. We do not prove this: the extension requires measurability in the noise history and envelope estimates uniform over noise paths, and the guarantee of Theorem 4 is stated only without common noise. Condition (7.3) identifies the structure such an extension would rest on.

7.3 Learning Master Policies under Common Noise

The M-OMD algorithm of Chapter 6 is not restated here; this section describes only what changes in the presence of common noise. Recall its structure: at each iteration the current policy generates mean-field flows from a training set of initial distributions, a deep Q-network $\tilde{Q}_\theta^k$ taking (n, x_n, μ_n, a_n) as input is trained with the Munchausen-regularized target that implicitly sums historical Q-functions [115, 72], the inner-loop replay buffer is reset at each iteration, and the new policy is the softmax of the learned Q-function.

Under common noise, the equilibrium objects of Section 7.2—Q-functions, policies, and distributions—are conditioned on the shock. The guiding question is what minimal additional information the master policy must consume to remain an equilibrium policy, and the answer adopted here is to incorporate the common noise directly as an input component of the policy and Q-networks; the master policies of Chapter 6, being trained with deep neural networks, extend naturally in this way. Two implementation choices make the extension concrete.

Conditioning on the noise history. Although in principle the conditional objects depend on the shock realization, in practice distinguishing the influence of continuously varying shock

values from the current value alone is challenging. The policy and Q-network inputs therefore include the entire realized sequence Ξ_n up to the current step, rather than only the current shock ξ_n. The network input thus becomes the tuple (timestep, individual state, mean field, noise history), against which the same Munchausen-regularized evaluation and softmax policy update of Chapter 6 are run unchanged.

Causal revelation of the shock. During training, each shock sequence is pre-generated, progressively revealed as the episode advances, and padded with zeros at the entries corresponding to timesteps not yet observed. This keeps the network input length constant across the episode while ensuring that the learner conditions only on causally available noise data—the population never has prior knowledge of future shocks, matching the information structure of the game.

No other component of the algorithm is modified: the implicit summation of historical Q-values, the inner-loop replay buffer reset, and the current-iteration target policy are exactly as in Chapter 6.

Remark 4 (When the noise history can be truncated). The conditional mean field $\mu_n(\cdot \mid \Xi_n)$ summarizes the effect of past shocks on the current population state, but it does not identify or recover the realized shocks. Past shocks can still matter separately through the information they provide about future shocks. This yields a criterion for how much of Ξ_n the input must carry. If the shocks ξ_n are i.i.d., or more generally if the shock process is Markov, then the current shock state alongside (n, x_n, μ_n) Markov-completes the problem, and the history adds nothing. Only a non-Markov shock process, such as a scripted scenario or a process with a hidden state, requires the history Ξ_n, or a compressed information state that makes the noise process Markov. The two benchmarks of Section 7.4 sit on opposite sides of this criterion. The beach-bar closure switch is Markov, so the bar's current status and the time would suffice. The scripted LQ shock sequences of Appendix B are non-Markov: in the middle window both scripts read zero while their futures are opposite, so the current value alone

cannot distinguish them. Feeding the zero-padded history, as done here, preserves all of the observed shock information in both cases, but is redundant when the noise is Markov.

7.4 Experimental Results

7.4.1 Setup

Across Part II, seven examples in three canonical MFG environments are studied; the five without common noise are reported in Chapter 6, and this section reports the two that involve common noise: a one-dimensional beach-bar model with a random "closure" shock, and a one-dimensional linear-quadratic model with an additive aggregate shock. As in Chapter 6, solving these games requires identifying an equilibrium—a harder task than reward maximization—and performance is measured by exploitability.

The algorithms compared are the population-dependent ones, since population-independent policies cannot condition on the shock-driven evolution of the mean field: **M-OMD** (the algorithm of this part, with the common-noise inputs of Section 7.3), **Master FP (M-FP)** [98], the state-of-the-art fictitious-play learner of master policies, extended with the same noise inputs, and a **model-based** solution computed with known transitions and rewards, which serves as the reference equilibrium behavior. The full baseline suite, network architectures, and hyperparameters are those of Chapter 6: Q-networks follow the DQN architecture [86], distributions are represented as histograms flattened into vectors, timesteps are one-hot encoded, and the noise history enters the input as described in Section 7.3. Exploitability curves are averaged over 5 realizations of each algorithm, with shaded areas showing the standard deviation where relevant; hardware details are reported in Appendix B.

7.4.2 Beach Bar with Common Noise

The beach-bar environment, introduced in [99], represents agents moving on a beach towards a bar: each agent seeks to get close to the bar while avoiding the crowd. The dynamics are those of the exploration environments of Chapter 6, the bar is located at the center of the beach, and it is not possible to go beyond the domain. The reward function is

$$r\left(x, a, \mu\right) = d_{bar}\left(x\right) - \frac{|a|}{|\mathcal{X}|} - C \log\left(\mu\left(x\right)\right),$$

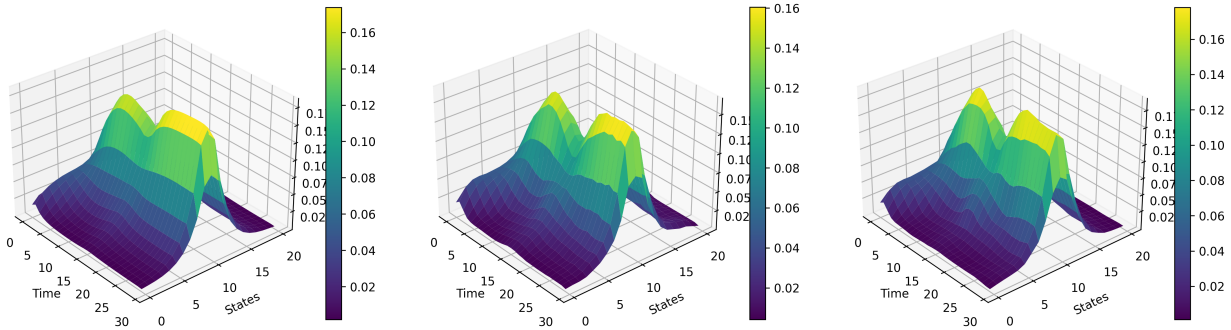
where d_{bar} represents the distance to the bar, the second term penalizes movement to encourage minimal action, and the third term discourages agents from occupying crowded regions. The parameter $C = 1$ when the bar is open and $C = 0$ when it is closed. The domain $\mathcal{X}$ is one-dimensional with 11 states. The common noise follows [99] and acts as a random switch determining whether the bar is open or closed. The current bar status is revealed causally, but agents do not know future status changes. Figure 7.1 visualizes the dynamic evolution of the populations under the master FP and master OMD policies against the model-based solution under this "closure" shock, together with the exploitability across seeds. Both master policies track the model-based evolution after the current status is observed, without advance knowledge of future switches.

7.4.3 Linear-Quadratic with Common Noise

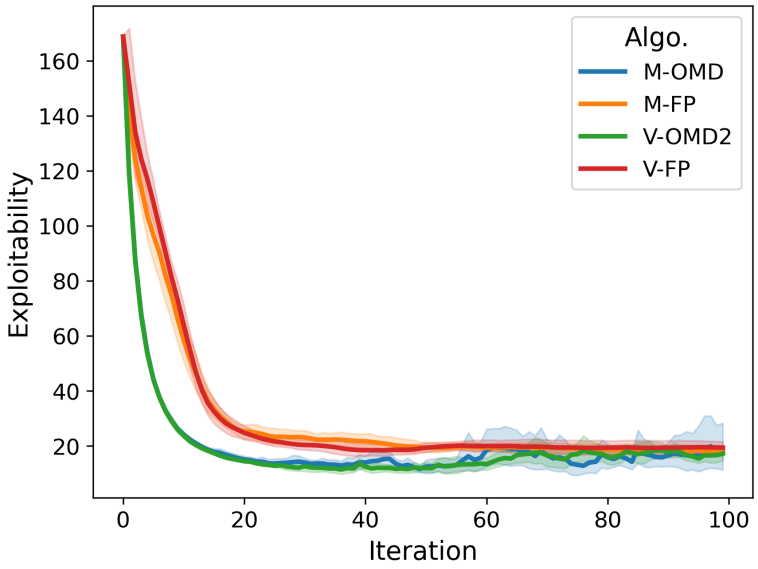
The linear-quadratic (LQ) model is a classical setting studied, e.g., in [16, 6]; a discretized version was introduced in [99]. Results with common noise are presented here; the corresponding model without common noise is studied in Chapter 6. The LQ model with common noise is one-dimensional, with dynamics

$$x_{n+1} = x_n + a_n \Delta_n + \sigma \left(\rho\, \xi_n + \sqrt{1-\rho^2}\, \epsilon_n\right) \sqrt{\Delta_n}, \tag{7.4}$$

corresponding to moving by a number of states (left or right), with action set $\mathcal{A} = \{-M, \ldots, M\}$. The common noise ξ_n enters every agent's dynamics through the corre-



(a) Model-based evolution (b) Master-FP evolution (c) Master-OMD evolution



(d) Exploitability

Figure 7.1: Beach bar with the "closure" common noise. The current bar status is revealed causally; future switches are not known in advance. (a) shows the model-based solution, (b) and (c) present the master FP and master OMD solutions, respectively, and (d) shows the exploitability across multiple seeds.

lation parameter ρ, while ϵ_n is an idiosyncratic noise perturbing the action choice, with $\epsilon_n \sim \mathcal{N}(0,1)$ discretized over $\{-3\sigma, \dots, 3\sigma\}$. The state space is $\mathcal{X} = \{-L, \dots, L\}$, of dimension $|\mathcal{X}| = 2L - 1$. The reward function is

$$r_n(x, a, \mu) = \Big[-\tfrac{1}{2}|a|^2 + q\, a\,(m - x) - \tfrac{\kappa}{2}(m - x)^2 \Big] \Delta_n, \tag{7.5}$$

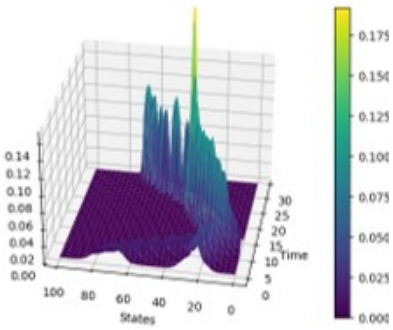
where $m = \sum_{x \in \mathcal{X}} x\, \mu(x)$ represents the population mean, encouraging agents to align with the population's center while maintaining dynamic movement. The terminal reward is $r_{N_T}(x, a, \mu) = -\frac{c_{\text{term}}}{2}(m - x)^2$. The experiments use a horizon of $N_T = 30$ steps; the common noise ξ_n enters every agent's dynamics through the correlation parameter ρ and takes values of magnitude ± 10. Two instances of the shock sequence are studied: a step-up-then-step-down aggregate shock and its mirror image. The complete parameter list and the two shock sequences are given in Appendix B.

In the LQ experiment, the population remains concentrated around its mean while the mean moves in the direction of the common-noise input. The learned trajectories are similar to the model-based trajectories shown in Figures 7.2 and 7.1.

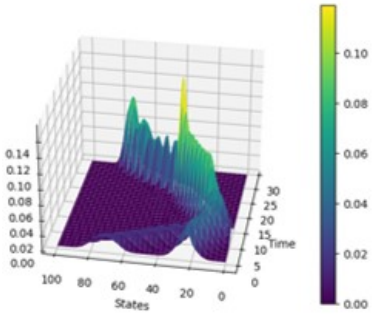
7.5 Discussion

Numerical findings. In the LQ experiments, the population trajectory follows the common-noise sequence. FP and OMD converge at similar rates under these settings, whereas fictitious play converges faster in the noise-free LQ model of Chapter 6. The learned policies also depend less on the full population distribution than in the exploration and beach-bar benchmarks. The policy nevertheless takes the mean field and the directly observed noise history as separate inputs.

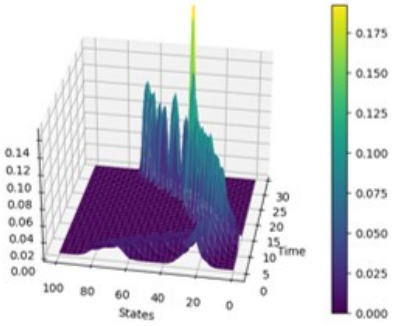
Related work. To the best of the author's knowledge, M-OMD is the first method to handle common noise and unknown initial distributions in finite-horizon MFGs. Most works on reinforcement learning for MFGs focus on population-independent policies, see e.g. [42, 26].



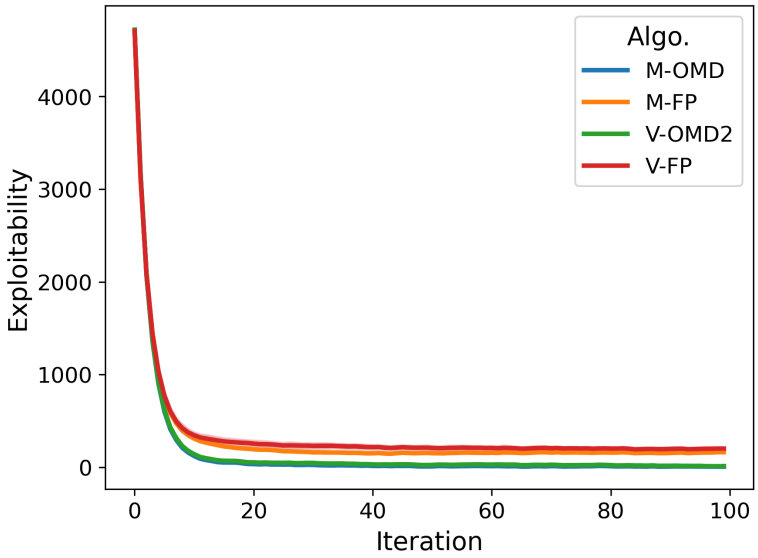
(a) Model-based evolution



(b) Master-FP evolution



(c) Master-OMD evolution



(d) Exploitability

Figure 7.2: Linear-quadratic model under the first common-noise shock sequence (ξ_n^1; see Appendix B). The population perceives the disturbance only up to the present moment. (a) shows the model-based solution, (b) and (c) present the master FP and master OMD solutions, respectively, and (d) shows the exploitability across multiple seeds.

Laurière et al. [72] combine fictitious play and online mirror descent with deep RL, but still for population-independent policies. Perrin et al. [98] learn master policies but rely on fictitious play, whereas the approach of this part relies on online mirror descent, which converges faster in the experiments of Chapter 6. On the theoretical side, MFGs with common noise are analyzed in [17] and through the master equation in [13]; the algorithmic contribution here is to make that regime accessible to model-free deep reinforcement learning.

7.6 Extension to Continuous State-Action Spaces

The work summarized in this section is led by collaborators—Lorenzo Magnino, Kai Shao, Jiacheng Shen, and Mathieu Laurière—with the dissertation author as a contributing author [85]; it is presented here as an outlook on the research line of Part II rather than as a contribution of this dissertation, and the reader is referred to the published paper for the complete treatment.

Everything in Chapter 6 and in the present chapter operates on finite state and action spaces: the mean field is a histogram, and the master policy consumes it as a finite-dimensional vector. The monitoring populations that motivate this dissertation—vehicles, drones, pedestrians, animals—move in continuous space, where this representation breaks down: the mean field lives in an infinite-dimensional space of probability measures, equilibrium policies are time-dependent because the problem is non-stationary, and the local population *density*, which drives congestion- and exploration-type rewards, can no longer be read off a histogram. Prior reinforcement-learning methods for MFGs are largely restricted to finite spaces or stationary models [71].

To address this regime, Magnino et al. [85] introduce **Density-Enhanced Deep Average Fictitious Play (DEDA-FP)**, a deep reinforcement learning algorithm for non-stationary MFGs with continuous state and action spaces. DEDA-FP learns a time-conditioned normalizing-flow model of the mean-field distribution. The model can be sampled and

evaluated pointwise, allowing the algorithm to compute density-dependent local-interaction rewards without estimating the density from each batch of trajectories. The authors also establish a convergence guarantee: the learned policy approaches an approximate (ϵ-)Nash equilibrium as the underlying approximation errors vanish. They evaluate DEDA-FP on continuous-space beach-bar, linear-quadratic, four-rooms, and price-impact models.

DEDA-FP complements M-OMD rather than extending it. It computes decentralized, time-dependent equilibrium policies for a fixed and known initial distribution, and its authors leave master policies and common noise for future work. An open problem is to combine the two lines in continuous state-action spaces, using learned density models while allowing a population-aware policy to respond to common noise.

7.7 Conclusion

This chapter extended population-aware online mirror descent to mean-field games with common noise. The policy, population distribution, and Q-functions condition on the realized shock history. The history is appended to the network input as it becomes available, while the other components of M-OMD are unchanged. On the beach-bar and linear-quadratic benchmarks, the learned master policies reproduced the model-based behavior along noise realizations that were not revealed in advance and reduced exploitability across seeds. The convergence result of Chapter 6 does not cover common noise or the sampled deep-learning implementation. These questions, along with multi-population models, remain for future work. Section 7.6 summarizes a collaborator-led extension to continuous state-action spaces.

The next chapter summarizes the results of both parts and discusses future work.

CHAPTER 8

Conclusion and Future Directions

This dissertation began with a scaling problem. A larger team can sense and act in more places, but that advantage is limited if every member must sense the whole environment, communicate with everyone, and store and compute over a global model. The dissertation addressed two parts of this problem: estimating external inputs from local observations and choosing actions when the agents' policies change the population around them. These are the estimate and act steps of the observe–estimate–act loop introduced in Chapter 1.

In the networked setting, prior information and exchanges between neighbors supported state and input estimates without centralizing all raw measurements and models. The results also set a limit: if the observations do not identify a unique input, the estimator should retain the remaining ambiguity, and that uncertainty can guide the next measurement. In the population setting, a distribution replaced the list of individual agents. This avoids tracking agents one by one, but it creates a different requirement: the policy must be consistent with the population response that it produces. On finite-state benchmarks, the population-aware policies responded to different initial distributions and to common shocks supplied causally. They still assume access to the full population state. The convergence proof covers only an idealized update without common noise, not the sampled neural implementation.

The two parts use different algorithms, but they support the same practical conclusion: in the settings studied here, an agent can avoid a complete system view when the information it receives is matched to the estimate or decision it must make. Finite sensor networks can use local estimates, neighbor messages, and an explicit account of what remains unobservable.

Mean-field models can use a population distribution and an observed shock history instead of the identities and histories of all agents. This reduces what each agent must process without hiding what the model does not provide. It is one step toward environmental sensor networks, vehicle fleets, and robot teams that act from partial views rather than from a central reconstruction of the whole system. Continuous environments, partial population observations, safety constraints, and separate individual and group goals remain open.

8.1 Future Directions

The connection between estimation and sensing should be tested in continuous, changing environments. A moving density makes the next sensor location depend on both present uncertainty and the motion expected before a measurement arrives. Lloyd-type controllers can track moving Voronoi centroids, but this remains a local guarantee [73]. Online submodular methods provide regret bounds for time-varying objectives on finite action sets, and their constants worsen as the spatial grid is refined [110, 126]. A useful coverage guarantee should instead depend on physical quantities such as sensor range, motion limits, density smoothness, and model error. It should continue to hold when the density is estimated online rather than given to the planner. Such a result would also need to explain how a new measurement changes the estimate, not only how well a fixed density is covered.

Population-aware decision-making must also move beyond finite state and action spaces. Vehicles and drones operate in continuous domains with obstacles, dynamics, and safety constraints. State-constrained mean-field game theory gives results for some continuous models, while current learning methods often handle obstacles through penalties or simulator boundaries [12, 105]. Continuous populations also cannot be passed to a policy as a full histogram. Samples, learned density models, or lower-dimensional summaries are possible representations; the density model in [85] gives one example for continuous mean-field games. The next question is whether a master policy can use such a representation while respecting hard constraints and responding to common shocks.

The policies in Part II assume the full population state and observed shock history. A physical agent may see only nearby agents, delayed messages, and its own observations. It may first need to estimate the population before using a population-aware policy. Past shocks should be represented only by information the agent has observed, or by a state that summarizes it. Future work should measure how population-estimation errors affect the policy and find compact states for non-Markov shock histories. It should also allow interactions that depend jointly on an agent's action and the surrounding population, and quantify the gap between a mean-field equilibrium and a large but finite, heterogeneous team.

Individual and group objectives need to be handled in the same model. A taxi may try to complete its assigned trip and preserve its battery while the fleet manages congestion and service coverage. A rescue drone may protect its own flight margin while the team searches a dangerous area. Interpolations between mean-field games and mean-field control, or a coordinator that sets a group objective, provide starting points [4, 33]. The remaining work is to determine when learning is stable under constraints, partial communication, and objectives that are not fully aligned. The aim is to add these features without returning to a controller that must track the entire team.

APPENDIX A

Master OMD: Additional Experimental Details

This appendix collects the training hyperparameters, the hyperparameter sweeps, the ablations, and the scaling studies for the Master OMD experiments of Chapter 6.

A.1 Training Hyperparameters

Table A.1 lists the hyperparameters used to train each algorithm on the exploration tasks.

	M-OMD	M-FP	V-FP	V-OMD1	V-OMD2
Network architecture	MLP	MLP	MLP	MLP	MLP
Neurons in each layer	64×64	64×64	64×64	64×64	64×64
Environment horizon	30	30	30	30	30
Number of agents	500	500	500	500	500
Max steps per iteration	30000	30000	30000	30000	30000
OMD τ	50	N/A	N/A	5.0	50
OMD α	N/A	N/A	N/A	1.0	1.0
Freq. to update target	4	4	4	4	4
Exploration fraction	0.1	0.1	0.1	0.1	0.1
γ	0.99	0.99	0.99	0.99	0.99
Batch size	32	32	32	32	32
Gradient steps	1	1	1	1	1

Table A.1: Hyperparameters used for training on the exploration tasks with map size 11×11.

A.2 Hyperparameter Sensitivity

Figure A.1 reports the sweeps over the temperature τ for M-OMD and over (τ, α) for V-OMD1 [72] on the exploration task. For M-OMD, $\tau = 50$ gives the fastest and most stable convergence among $\{5, 10, 50, 100\}$, and this value is used throughout (Table A.1); V-OMD1 attains its best behavior at a smaller regularization ($\tau = 5$, $\alpha = 1$).

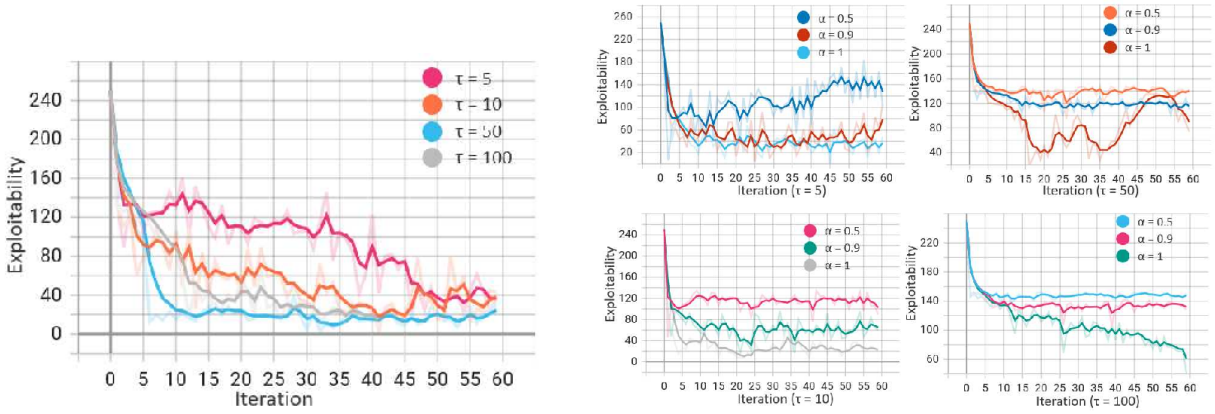


Figure A.1: Hyperparameter sweeps. Left: exploitability of M-OMD for temperatures $\tau \in \{5, 10, 50, 100\}$. Right: exploitability of V-OMD1 for combinations of $\tau \in \{5, 10, 50, 100\}$ (one panel each) and $\alpha \in \{0.5, 0.9, 1\}$.

A.3 Ablation: The Inner-Loop Replay Buffer

Figure A.2 isolates the contribution of the buffer-placement strategy of Section 6.3.3 by running M-OMD on the two exploration tasks with and without the inner-loop replay buffer. Without it, transitions from the different training distributions are not mixed within an iteration, some initial distributions are forgotten, and the exploitability stagnates at a high level on both tasks; with it, both tasks converge.

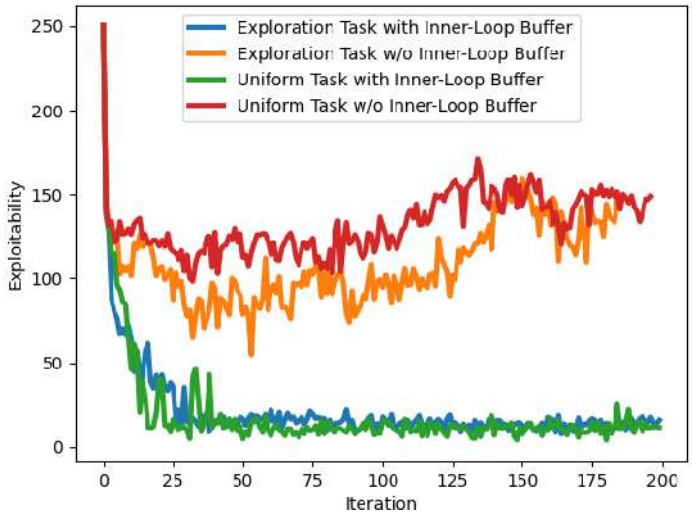


Figure A.2: Exploitability vs iteration for M-OMD with and without the inner-loop replay buffer, on the four-connected-rooms exploration task and the one-room (uniform) exploration task. Removing the buffer leads to forgetting of some training distributions and hence poor performance (see Step 2 in Algorithm 3).

A.4 Scalability in Memory and Computation

Two properties let M-OMD scale where FP-type methods do not. In *memory*, FP must retain every past best-response network, since the equilibrium policy is their average, so the stored policy grows linearly with the iteration count and the map size; M-OMD instead keeps a single Q-network throughout, thanks to the implicit summation of Theorem 3. After 100 iterations on the exploration task the M-OMD policy occupies 0.58 MB while the M-FP policy library exceeds 37 MB, and at map size 360 the gap is 1.56 MB versus more than 80 MB (Figure A.3). In *exploitability-evaluation cost*, computing exploitability requires executing all historical policies for FP but only the single network for M-OMD, so this cost grows linearly with the iteration count for FP while staying constant for M-OMD (Figure A.4); the per-iteration policy-learning time of M-OMD is slightly longer than FP's, but its faster convergence and constant-cost evaluation save substantial total computation. Across map dimensions from 7×7 to 25×25, M-OMD also converges faster than M-FP at every size.

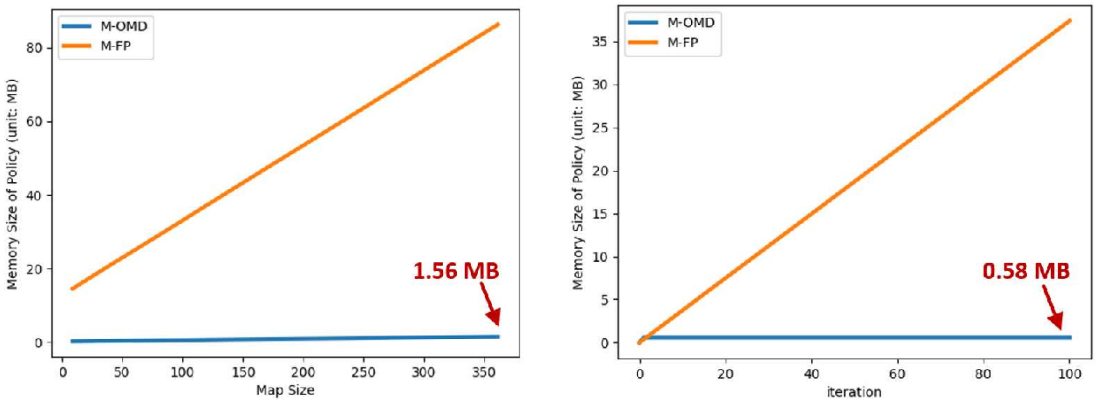


Figure A.3: Model size comparison for M-OMD (ours) and M-FP. Left: memory footprint of the stored policy versus map size. Right: memory footprint versus training iteration.

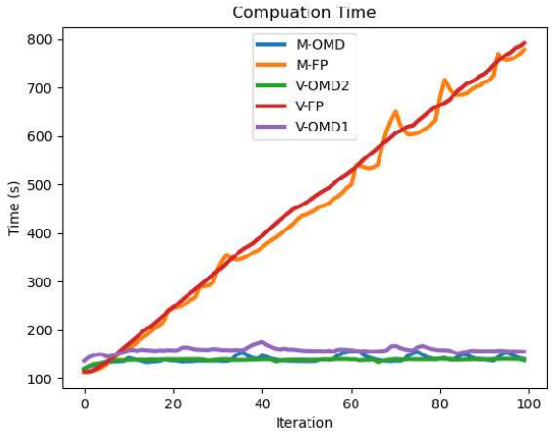


Figure A.4: Time per iteration spent computing the exploitability on the exploration task. The cost for the FP-type algorithms (V-FP, M-FP) grows linearly with the iteration count because all past policies must be executed, whereas the OMD-type algorithms use a single policy network.

A.5 Scaling the Training Set: 30 Initial Distributions

The main experiments train on five initial distributions. To probe adaptability across a broader spectrum, we consider an ensemble of 30 distributions for each exploration task, comprising 10 distributions concentrated at fixed points, 10 Gaussian distributions, and 10 spread across random points. At this scale the policy architecture matters: a 64×64 MLP struggles to converge, while a 256×256 MLP performs best (a CNN does not outperform the MLPs on these tasks), so we adopt the 256×256 MLP here; the full architecture comparison is reported in Section A.6. To report the *true* exploitability rather than a DQN-approximated one—which may differ from the authentic best response—we solve the inner best-response problem of (6.2) by dynamic programming. We ran both M-FP and M-OMD with identical 256×256 architectures on the two exploration tasks with five seeds (42, 3407, 303, 109, 312). As shown in Figure A.5, M-OMD outperforms M-FP in true exploitability on both the training and the testing set for both tasks, in addition to its advantages in computational efficiency, execution time, and model compactness highlighted in Section A.4.

A.6 Policy Architecture

To assess the impact of the policy architecture at the 30-distribution scale, we evaluated four architectures: MLPs with 64×64, 128×128, and 256×256 hidden layers, and a CNN with two convolutional layers (32×64, kernel sizes 5 and 3) followed by a fully connected layer. As shown in Figure A.6, the 64×64 MLP struggles to converge with 30 distributions, while the 256×256 MLP performs best; we adopt it for the study of Section A.5. Notably, the CNN does not outperform the MLPs on these tasks.

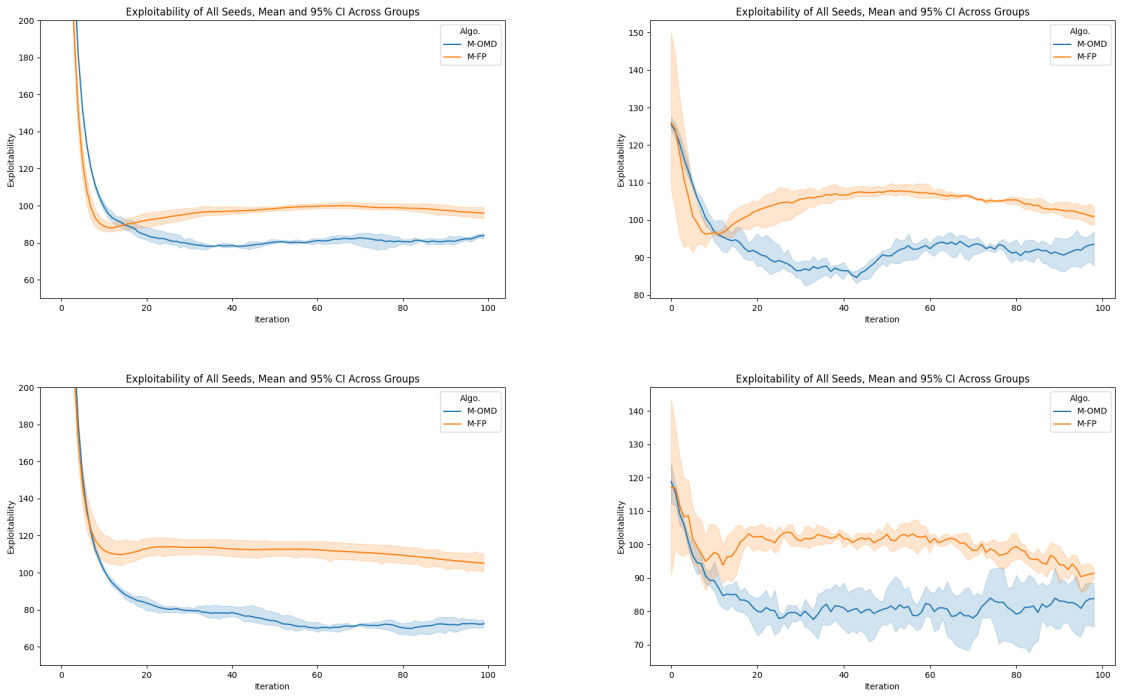


Figure A.5: True exploitability (best response computed by dynamic programming) for M-OMD and M-FP with the 256×256 MLP architecture, trained on 30 initial distributions; mean and 95% confidence interval over 5 seeds. Top row: one-room exploration task, training set (left) and testing set (right). Bottom row: four-connected-rooms exploration task, training set (left) and testing set (right).

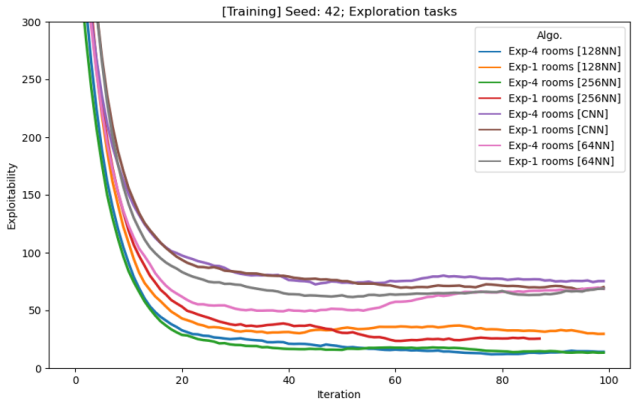


Figure A.6: Training on the two exploration tasks with 30 initial distributions: exploitability for four Q-network architectures (64×64, 128×128, and 256×256 MLPs, and a CNN) on the one-room and four-rooms tasks.

A.7 Ad-Hoc Teaming (Outlook)

We also tested two cases in which a new group joined during execution. Figure A.7 shows that the policy handled a group of 200 agents and the population still reached an approximately uniform distribution. A group of 2500 agents arriving late did not fully spread within the remaining horizon. Training on such cases is left for future work.

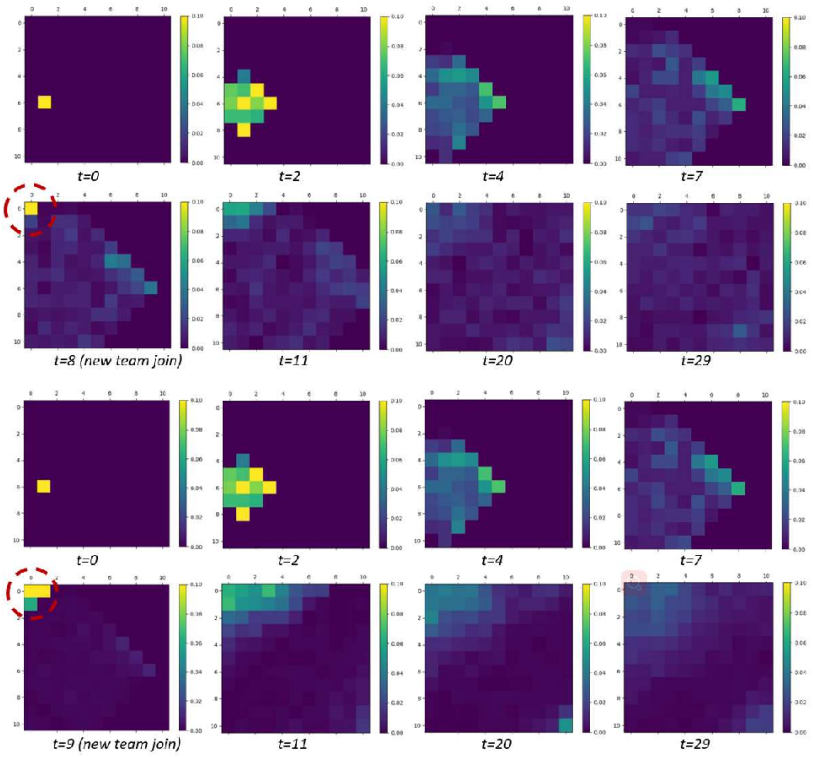


Figure A.7: Ad-hoc teaming tests: a new team joins the current team (500 agents) during the evolution of the population (joining location circled in red). Top: a small team (200 agents) joins at $t = 8$ and the population still reaches an approximately uniform distribution. Bottom: a large team (2500 agents) joins at $t = 9$ and the terminal distribution deviates from uniform.

APPENDIX B

Common-Noise Experiment Details

This appendix collects the full parameter list and the two common-noise shock sequences used in the linear-quadratic (LQ) experiments of Section 7.4.3, together with the compute hardware.

LQ parameters. The LQ experiments use $\sigma = 1$, $N_T = 30$, $\Delta_n = 1$, $q = 0.01$, $\kappa = 0.5$, $K = 1$, $M = 3$, $L = 50$, and $c_{term} = 1$.

Common-noise shock sequences. Two instances of the shock sequence are studied:

$$\xi_n^1 = \begin{cases} -10 & n \le 8 \\ 0 & 8 < n \le 20 \\ 10 & n > 20 \end{cases} \qquad \xi_n^2 = \begin{cases} 10 & n \le 8 \\ 0 & 8 < n \le 20 \\ -10 & n > 20 \end{cases}$$

Hardware. Experiments were run on an NVIDIA TITAN RTX GPU (24 GB) with two 16-core Intel Xeon Gold CPUs.

APPENDIX C

Proofs and Convergence Analysis for Master OMD

This appendix contains the proofs for Chapter 6. Section C.1 proves the Munchausen equivalence theorem (Theorem 3), which underlies the deep implementation of Algorithm 3. Sections C.2–C.6 prove the convergence theorem for the idealized master OMD dynamics (Theorem 4). Throughout, $\mathbb{E}$ denotes expectation and $\mathcal{E}$ denotes exploitability.

C.1 Proof of the Munchausen Equivalence

This section proves Theorem 3. The notation is that of the theorem statement: $\tau > 0$ is the temperature, $\boldsymbol{\pi}^{k-1}$ the softmax policy of iteration $k-1$, $Q^k = Q^{\boldsymbol{\pi}^{k-1}}$ its state-action value function, $\widetilde{Q}^k = Q^k + \tau \ln \boldsymbol{\pi}^{k-1}$, and $\boldsymbol{\mu}^k$ the mean field generated by $\boldsymbol{\pi}^{k-1}$ in Algorithm 3.

Proof of Theorem 3. Fix a pair (n, x) and abbreviate $Q^i = Q_n^i(x, \mu_n^k, \cdot) \in \mathbb{R}^{|\mathcal{A}|}$, $\widetilde{Q}^k = \widetilde{Q}_n^k(x, \mu_n^k, \cdot)$, and $\pi^{k-1} = \pi_n^{k-1}(\cdot|x, \mu_n^k) \in \Delta_{\mathcal{A}}$; the optimization below is over $\pi \in \Delta_{\mathcal{A}}$. Recall that $\mathrm{KL}(\pi\|\pi') = \langle \pi, \ln \pi - \ln \pi' \rangle$. The proof proceeds in three steps: we first establish the two variational characterizations

$$\mathrm{softmax}\Big(\frac{1}{\tau}\sum_{i=0}^{k} Q^i\Big) = \arg\max_{\pi} \ \big\langle \pi, Q^k \big\rangle - \tau\, \mathrm{KL}\big(\pi \,\big\|\, \pi^{k-1}\big), \tag{C.1}$$

$$\mathrm{softmax}\Big(\frac{1}{\tau}\widetilde{Q}^k\Big) = \arg\max_{\pi} \ \big\langle \pi, \widetilde{Q}^k \big\rangle - \tau \big\langle \pi, \ln \pi \big\rangle, \tag{C.2}$$

and then show that the two right-hand sides coincide.

Step 1. Consider the maximization in (C.1) subject to the simplex constraint $\mathbf{1}^\top \pi = 1$, where $\mathbf{1}$ is the all-ones vector of dimension $|\mathcal{A}|$. Introduce the Lagrange multiplier λ and the

Lagrangian

$$L(\pi, \lambda) = \langle \pi, Q^k \rangle - \tau \, \mathrm{KL}\big(\pi \| \pi^{k-1}\big) + \lambda\big(\mathbf{1}^\top \pi - 1\big).$$

For every λ, the map $\pi \mapsto L(\pi, \lambda)$ is concave, being the sum of a linear function and the negative of the (convex) KL divergence. Its partial derivative is

$$\frac{\partial L(\pi, \lambda)}{\partial \pi} = Q^k - \tau\big(\ln \pi + \mathbf{1} - \ln \pi^{k-1}\big) + \lambda \mathbf{1},$$

and setting it to zero gives, componentwise,

$$\pi = e^{\frac{1}{\tau}(Q^k + \lambda \mathbf{1}) - 1} \, \pi^{k-1}. \tag{C.3}$$

By hypothesis $\pi^{k-1} \propto e^{\frac{1}{\tau}\sum_{i=0}^{k-1} Q^i}$, so unrolling (C.3) down to π^0 yields $\pi \propto e^{\frac{1}{\tau}\sum_{i=0}^{k} Q^i}$, where the proportionality constant collects e^{-1}-type factors and the multipliers. The constraint $\partial L / \partial \lambda = \mathbf{1}^\top \pi - 1 = 0$ fixes this constant to the normalizing denominator $\big(\sum_a e^{\frac{1}{\tau}\sum_{i=0}^{k} Q^i(a)}\big)^{-1}$, which is exactly the softmax; this proves (C.1).

Step 2. The same argument applied to $\langle \pi, \widetilde{Q}^k \rangle - \tau \langle \pi, \ln \pi \rangle + \lambda(\mathbf{1}^\top \pi - 1)$, whose partial derivative in π is $\widetilde{Q}^k - \tau(\ln \pi + \mathbf{1}) + \lambda \mathbf{1}$, gives $\pi \propto e^{\frac{1}{\tau}\widetilde{Q}^k}$ at the stationary point, and normalization again produces the softmax; this proves (C.2).

Step 3. Substituting $\widetilde{Q}^k = Q^k + \tau \ln \pi^{k-1}$ into the objective of (C.2),

$$\big\langle \pi, \widetilde{Q}^k \big\rangle - \tau \big\langle \pi, \ln \pi \big\rangle = \big\langle \pi, Q^k \big\rangle + \tau \big\langle \pi, \ln \pi^{k-1} \big\rangle - \tau \big\langle \pi, \ln \pi \big\rangle = \big\langle \pi, Q^k \big\rangle - \tau \, \mathrm{KL}\big(\pi \| \pi^{k-1}\big),$$

so the objectives of (C.1) and (C.2) are identical and have the same maximizer. Combining the two variational characterizations gives $\mathrm{softmax}\big(\frac{1}{\tau}\sum_{i=0}^{k} Q^i\big) = \mathrm{softmax}\big(\frac{1}{\tau}\widetilde{Q}^k\big)$ at (x, μ_n^k), which is the claimed identity since $\pi_n^k(\cdot | x, \mu_n^k) = \mathrm{softmax}\big(\frac{1}{\tau}\sum_{i=0}^{k} Q_n^i(x, \mu_n^k, \cdot)\big)$ by definition of the OMD update. □

C.2 Equilibrium objects and the dynamics

The remainder of this appendix proves Theorem 4. The analysis is part of this dissertation; it builds on the online-mirror-descent convergence theory for mean-field games with a fixed

initial distribution of Pérolat et al. [97] and on the master-policy formulation of Perrin et al. [98]. Assumption 3 is in force; as in Section 6.3.4, the time index on p and r is suppressed, and nothing below uses time homogeneity. We record the quantitative content of the assumption: $|r| \le R_{\max}$, and for all $\mu, \mu' \in \Delta_{\mathcal{X}}$,

$$\max_x \left|\bar{r}(x,\mu) - \bar{r}(x,\mu')\right| \le L_r \|\mu - \mu'\|_2, \qquad \left\langle \mu - \mu',\, \bar{r}(\cdot,\mu) - \bar{r}(\cdot,\mu') \right\rangle \le -\lambda \|\mu - \mu'\|_2^2. \tag{C.4}$$

The negentropy $h(q) = \tau \sum_a q(a) \log q(a)$ has range $\tau \log |\mathcal{A}|$ on $\Delta_{\mathcal{A}}$, and its choice map is $\Gamma(y) = \mathrm{softmax}(y/\tau)$.

For a decision rule $\beta \in (\Delta_{\mathcal{A}})^{\mathcal{X}}$, define the one-step propagation and the induced state kernel

$$\Phi(\mu,\beta)(x') = \sum_{x,a} \mu(x)\, \beta(a \mid x)\, p(x' \mid x, a), \qquad P^{\beta}(x' \mid x) = \sum_a \beta(a \mid x)\, p(x' \mid x, a). \tag{C.5}$$

A master policy $\boldsymbol{\pi}$ has flow $\mu_0^{\boldsymbol{\pi},m} = m$, $\mu_{n+1}^{\boldsymbol{\pi},m} = \Phi\big(\mu_n^{\boldsymbol{\pi},m}, \pi_n(\cdot \mid \cdot, \mu_n^{\boldsymbol{\pi},m})\big)$ from any $m \in \Delta_{\mathcal{X}}$, and trace $\boldsymbol{\pi}[m]_n(\cdot \mid x) = \pi_n(\cdot \mid x, \mu_n^{\boldsymbol{\pi},m})$: the trace is the population-independent policy obtained by reading the master policy along its own flow. Against a fixed flow $\boldsymbol{\mu}$, a deviating agent faces a Markov decision problem; write $J_n(\eta, \boldsymbol{\sigma}; \boldsymbol{\mu})$ for the expected total reward from level n of a population-independent policy $\boldsymbol{\sigma}$ started from the state law η. Because transitions do not depend on μ, the deviator's state law is the forward recursion from η, whatever flow it is paid against. Define the fixed-flow dynamic program

$$u_{N_T+1}^{\boldsymbol{\mu}} = 0, \qquad Q_n^{\boldsymbol{\mu}}(x,a) = r(x,a,\mu_n) + \sum_{x'} p(x' \mid x, a)\, u_{n+1}^{\boldsymbol{\mu}}(x'), \qquad u_n^{\boldsymbol{\mu}}(x) = \max_a Q_n^{\boldsymbol{\mu}}(x,a). \tag{C.6}$$

The master exploitability of $\boldsymbol{\pi}$ at m is

$$\mathcal{E}^m(\boldsymbol{\pi}) = \max_{\boldsymbol{\sigma}} J_0\big(m, \boldsymbol{\sigma}; \boldsymbol{\mu}^{\boldsymbol{\pi},m}\big) - J_0\big(m, \boldsymbol{\pi}[m]; \boldsymbol{\mu}^{\boldsymbol{\pi},m}\big), \tag{C.7}$$

which is the exploitability (6.2) of Chapter 6 written for the trace; since the flow is frozen, the supremum may be taken over population-independent deviations.

An equilibrium of the continuation game started from (n, μ) has a population flow $\boldsymbol{\mu}^{*,(n,\mu)}$; Theorem 5 below shows that this flow is unique under (C.4). The master value and the equilibrium image are therefore unambiguous: $V_n^*(x,\mu) = u_n^{\boldsymbol{\mu}^{*,(n,\mu)}}(x)$ and $\nu_n^*(\mu) = \mu_{n+1}^{*,(n,\mu)}$. For any master policy $\boldsymbol{\pi}$, its recursive value is

$$\begin{aligned} Q_{N_T}^{\boldsymbol{\pi}}(x,\mu,a) &= r(x,a,\mu), \\ Q_n^{\boldsymbol{\pi}}(x,\mu,a) &= r(x,a,\mu) + \sum_{x'} p(x' \mid x,a)\, V_{n+1}^{\boldsymbol{\pi}}\big(x', \Phi(\mu, \pi_n(\cdot \mid \cdot, \mu))\big), \\ V_n^{\boldsymbol{\pi}}(x,\mu) &= \big\langle \pi_n(\cdot \mid x,\mu),\, Q_n^{\boldsymbol{\pi}}(x,\mu,\cdot)\big\rangle. \end{aligned} \tag{C.8}$$

The object of study is the synchronous continuous-time dynamics (6.5). It is not the sampled, function-approximated implementation of Algorithm 3.

C.3 Static theory: regret and propagation of monotonicity

For a fixed flow $\boldsymbol{\mu}$ and a policy $\boldsymbol{\sigma}$, define the one-step defect $d_k^{\boldsymbol{\sigma}}(x) = u_k^{\boldsymbol{\mu}}(x) - \langle \sigma_k(\cdot \mid x), Q_k^{\boldsymbol{\mu}}(x,\cdot)\rangle \geq 0$.

Lemma 2 (Regret decomposition)**.** *For every fixed flow $\boldsymbol{\mu}$, policy $\boldsymbol{\sigma}$, level n, and initial law η,*

$$\sum_x \eta(x)\, u_n^{\boldsymbol{\mu}}(x) - J_n(\eta, \boldsymbol{\sigma}; \boldsymbol{\mu}) = \sum_{k=n}^{N_T} \sum_x \mu_k^{\boldsymbol{\sigma},\eta}(x)\, d_k^{\boldsymbol{\sigma}}(x), \tag{C.9}$$

where $\mu_k^{\boldsymbol{\sigma},\eta}$ is the state law of $\boldsymbol{\sigma}$ started from η. In particular, an equilibrium rule is greedy for (C.6) *at every state charged by its own flow.*

Proof. Let $v_k^{\boldsymbol{\sigma}}$ be the value of $\boldsymbol{\sigma}$ against $\boldsymbol{\mu}$. The two backward recursions give $u_k - v_k^{\boldsymbol{\sigma}} = d_k^{\boldsymbol{\sigma}} + P^{\sigma_k}\big(u_{k+1} - v_{k+1}^{\boldsymbol{\sigma}}\big)$. Multiply by $\mu_k^{\boldsymbol{\sigma},\eta}$, sum over x, use the forward equation $\mu_{k+1}^{\boldsymbol{\sigma},\eta} = \mu_k^{\boldsymbol{\sigma},\eta} P^{\sigma_k}$, and telescope. The summands in (C.9) are nonnegative, so optimality forces each to vanish wherever the occupancy is positive. □

Lemma 3 (Existence and the master dynamic program)**.** *Every continuation game has an equilibrium. Under* (C.4) *its flow is unique, and*

$$V_n^*(x,\mu) = \max_a \Big[r(x,a,\mu) + \sum_{x'} p(x' \mid x,a)\, V_{n+1}^*\big(x', \nu_n^*(\mu)\big) \Big]. \tag{C.10}$$

Proof. Map a time-indexed policy to the product of the best-response faces of (C.6) against the flow it induces. This correspondence has nonempty compact convex values and closed graph, since the flow is polynomial in the policy and the backward recursion is continuous; Kakutani's theorem gives an equilibrium. Flow uniqueness follows from Theorem 5 below. By (C.9), the tail of an equilibrium is an equilibrium of the restarted continuation game, and flow uniqueness identifies its flow with the restarted flow; inserting the tail value into (C.6) gives (C.10). □

Theorem 5 (Propagation of Lasry–Lions monotonicity)**.** *For every level n and all $\mu, \mu' \in \Delta_{\mathcal{X}}$,*

$$\big\langle \mu - \mu',\, V_n^*(\cdot,\mu) - V_n^*(\cdot,\mu') \big\rangle \le -\lambda \|\mu - \mu'\|_2^2 - \lambda \sum_{k=n+1}^{N_T} \big\| \mu_k^* - (\mu_k^*)' \big\|_2^2, \tag{C.11}$$

where (μ_k^*) and $((\mu_k^*)')$ are equilibrium flows started at (n,μ) and (n,μ'). Consequently equilibrium flows are unique, and the maps $\mu \mapsto \nu_n^*(\mu)$ and $\mu \mapsto V_n^*(\cdot,\mu)$ are Lipschitz, uniformly over levels; one may take $L_\nu = (N_T+1)L_r/\lambda$ and $L_V = (N_T+1)L_r^2/\lambda$ after norm conversion.

Proof. Let $(\mu_k^*, \beta_k^*, u_k)$ and $((\mu_k^*)', (\beta_k^*)', u_k')$ be equilibria started at (n,μ) and (n,μ'), each valued against its own flow. Put $W_k = \langle \mu_k^* - (\mu_k^*)',\, u_k - u_k' \rangle$ and $\rho_k^*(x,a) = \mu_k^*(x)\beta_k^*(a \mid x)$, with primes for the second system. We claim the four-term doubling recursion

$$W_k \le \sum_{x,a} \big(\rho_k^* - (\rho_k^*)'\big)(x,a) \big[r(x,a,\mu_k^*) - r(x,a,(\mu_k^*)') \big] + W_{k+1}. \tag{C.12}$$

Expand $W_k = T_1 + T_2 + T_3 + T_4$ with $T_1 = \langle \mu_k^*, u_k \rangle$, $T_2 = -\langle \mu_k^*, u_k' \rangle$, $T_3 = -\langle (\mu_k^*)', u_k \rangle$, and $T_4 = \langle (\mu_k^*)', u_k' \rangle$. The own terms T_1, T_4 are equalities by charged-state greediness (Lemma 2);

the cross terms T_2, T_3 are bounded by the dynamic-programming inequality evaluated with the same rule as the weight:

$$
\begin{aligned}
T_1 &= \sum_{x,a} \rho_k^*(x,a)\, r(x,a,\mu_k^*) + \big\langle \mu_{k+1}^*,\, u_{k+1} \big\rangle, \\
T_2 &\le -\sum_{x,a} \rho_k^*(x,a)\, r(x,a,(\mu_k^*)') - \big\langle \mu_{k+1}^*,\, u'_{k+1} \big\rangle, \\
T_3 &\le -\sum_{x,a} (\rho_k^*)'(x,a)\, r(x,a,\mu_k^*) - \big\langle (\mu_{k+1}^*)',\, u_{k+1} \big\rangle, \\
T_4 &= \sum_{x,a} (\rho_k^*)'(x,a)\, r(x,a,(\mu_k^*)') + \big\langle (\mu_{k+1}^*)',\, u'_{k+1} \big\rangle.
\end{aligned}
$$

The reward terms sum to the first term of (C.12); the continuation terms sum to W_{k+1}. By separability, $r(x,a,\mu_k^*) - r(x,a,(\mu_k^*)') = \bar r(x,\mu_k^*) - \bar r(x,(\mu_k^*)')$ does not depend on a, so summing $\rho_k^* - (\rho_k^*)'$ over a turns the reward term into $\langle \mu_k^* - (\mu_k^*)',\, \bar r(\cdot,\mu_k^*) - \bar r(\cdot,(\mu_k^*)') \rangle$. Telescoping (C.12) from $k = n$ with $W_{N_T+1} = 0$, applying (C.4) at each level, and noting $W_n = \langle \mu - \mu',\, V_n^*(\cdot,\mu) - V_n^*(\cdot,\mu') \rangle$ gives (C.11). If two equilibria share the initial distribution, the left side vanishes, so every dissipation term on the right vanishes and the flows agree. The Lipschitz assertions follow by running the same doubled inequalities for two starts, keeping the dissipation terms of (C.11), bounding the reward perturbation caused by changing the initial law with the Lipschitz half of (C.4), and comparing the recursions (C.10) backward. This stability step is legitimate because the continuation flow has just been shown unique. □

C.4 The frozen stage game and perturbed mirror descent

Fix a cell (n,μ) and suppress the indices. Define the stage payoffs

$$
\widehat G(x,a;\nu) = r(x,a,\mu) + \sum_{x'} p(x' \mid x,a)\, V_{n+1}^*(x',\nu), \qquad G(x,a;\beta) = \widehat G\big(x,a;\Phi(\mu,\beta)\big). \tag{C.13}
$$

By (C.10) and the tail property, the time-n rule of a continuation equilibrium is an equilibrium of this one-shot game, and every stage equilibrium has image $\nu_n^*(\mu)$.

Lemma 4 (Collapse identity)**.** *For $\nu = \Phi(\mu, \beta)$ and $\nu' = \Phi(\mu, \beta')$,*

$$\sum_{x,a} \mu(x)\,(\beta - \beta')(a \mid x)\,\big[G(x,a;\beta) - G(x,a;\beta')\big] = \big\langle \nu - \nu',\, V^*_{n+1}(\cdot,\nu) - V^*_{n+1}(\cdot,\nu')\big\rangle. \quad \text{(C.14)}$$

Hence, by Theorem 5 at level $n+1$, the stage game is λ-strongly monotone in its image.

Proof. The term $r(x,a,\mu)$ cancels in the payoff difference. Exchanging the finite sums, the coefficient of $V^*_{n+1}(x',\nu) - V^*_{n+1}(x',\nu')$ is $\sum_{x,a} \mu(x)(\beta - \beta')(a \mid x)\, p(x' \mid x, a) = \nu(x') - \nu'(x')$. □

Consider the perturbed one-shot OMD system

$$\dot{y}_t(x,a) = G(x,a;\beta_t) + \delta_t(x,a), \qquad \beta_t(\cdot \mid x) = \Gamma\big(y_t(x,\cdot)\big), \qquad \|\delta_t\|_\infty \le \varepsilon(t). \quad \text{(C.15)}$$

Let β^* be a stage equilibrium with image $\nu^* = \Phi(\mu, \beta^*)$, let $f(t) = \|\Phi(\mu,\beta_t) - \nu^*\|_2^2$ be the squared image error, and let $P(t) = \sum_x \mu(x)\big[\max_a G^\infty(x,a) - \langle \beta_t(\cdot \mid x), G^\infty(x,\cdot)\rangle\big]$ be the payoff gap against the equilibrium payoff $G^\infty(x,\cdot) = \widehat{G}(x,\cdot\,;\nu^*)$.

Lemma 5 (Stage Lyapunov estimate)**.** *Let $F(q,y) = h^*(y) - \langle q, y\rangle + h(q)$ be the Fenchel coupling and $L_t = \sum_x \mu(x)\, F\big(\beta^*(\cdot \mid x), y_t(x,\cdot)\big)$. Then, almost everywhere,*

$$\dot{L}_t \le -\lambda f(t) - P(t) + 2\varepsilon(t), \qquad L_0 \le \tau \log|\mathcal{A}|. \quad \text{(C.16)}$$

Proof sketch. The gradient of $F(q,\cdot)$ is $\Gamma(\cdot) - q$, so $\dot{L}_t = \sum_x \mu(x)\,\langle \beta_t(\cdot \mid x) - \beta^*(\cdot \mid x),\, G(x,\cdot\,;\beta_t) + \delta_t(x,\cdot)\rangle$. Add and subtract $G(\cdot\,;\beta^*) = G^\infty$. The monotone pairing is the left side of (C.14) and is at most $-\lambda f(t)$; the equilibrium pairing equals $-P(t)$, because β^* charges only maximizers of G^∞ at states charged by μ; the perturbation pairing has modulus at most $2\varepsilon(t)$. The initial bound is the range of the negentropy. □

Corollary 1 (Consequences for the perturbed stage system)**.** *Let $E(T) = \int_0^T \varepsilon(s)\,\mathrm{d}s$. Then*

$$\lambda \int_0^T f(s)\,\mathrm{d}s + \int_0^T P(s)\,\mathrm{d}s \le \tau \log|\mathcal{A}| + 2E(T). \quad \text{(C.17)}$$

If $E(T) = o(T)$, the image error and the payoff gap converge in time average and along a set of times of density one; if $\varepsilon \in L^1$, then $f(t) \to 0$. Moreover, if $\widehat{G}$ is L_G-Lipschitz in ν, then, uniformly in x,

$$\left\| \frac{y_t(x,\cdot)}{t} - G^\infty(x,\cdot) \right\|_\infty \le \kappa(t) := L_G \sqrt{\frac{\tau \log|\mathcal{A}| + 2E(t)}{\lambda t}} + \frac{E(t)}{t}, \tag{C.18}$$

$$0 \le \max_a \widehat{G}(x,a;\nu_t) - \left\langle \beta_t(\cdot \mid x),\, \widehat{G}(x,\cdot\,;\nu_t) \right\rangle \le \frac{\tau \log|\mathcal{A}|}{t} + 2\kappa(t) + 2L_G\sqrt{f(t)}, \tag{C.19}$$

where $\nu_t = \Phi(\mu, \beta_t)$.

Proof sketch. Integrate (C.16); under an L^1 perturbation a Barbalat argument applies because f is Lipschitz along a bounded trajectory. For (C.18), divide the time integral of (C.15) by t, compare with G^∞, and apply the Cauchy–Schwarz inequality with (C.17). For (C.19), use the entropy regularized-leader inequality $\max_a g(a) - \langle \Gamma(tg'), g\rangle \le \tau \log|\mathcal{A}|/t + 2\|g - g'\|_\infty$ with $g' = y_t(x,\cdot)/t$ and $g = \widehat{G}(x,\cdot\,;\nu_t)$, together with (C.18). □

Remark 5. If $\varepsilon(t) \to 0$ without being integrable and the equilibrium payoff has ties, (C.16) controls only the time average of f; last-iterate convergence of the image does not follow, and per-μ density-one convergence cannot be diagonalized into a statement uniform over the continuum of cells. This is exactly the gap that hypothesis (ENV) in Theorem 4(b) fills; the closures of Section C.6 prove (ENV) directly in the stated cases.

C.5 Backward induction and the exploitability bound

Fix a cell (n, μ) and let $\beta_t = \pi_{n,t}(\cdot \mid \cdot, \mu)$ and $\nu_t = \Phi(\mu, \beta_t)$. Comparing (C.8) with (C.13), the master dynamics (6.5) at this cell is exactly the perturbed stage system (C.15):

$$\dot{y}_{n,t}(x,\mu,a) = \widehat{G}(x,a;\nu_t) + \delta_{n,t}(x,\mu,a), \qquad \delta_{n,t} = \sum_{x'} p(x' \mid x,a)\big[V_{n+1}^{\boldsymbol{\pi}_t}(x',\nu_t) - V_{n+1}^{*}(x',\nu_t)\big], \tag{C.20}$$

so $\|\delta_{n,t}\|_\infty \le g_{n+1}(t) := \sup_{x,\mu} \left|V^{\boldsymbol{\pi}_t}_{n+1}(x,\mu) - V^*_{n+1}(x,\mu)\right|$. Theorem 5 and Lemma 4 verify the hypotheses of Corollary 1 at every cell, with constants independent of μ; in particular $\widehat{G}$ is L_G-Lipschitz in ν with L_G determined by L_V.

Lemma 6 (Value recursion)**.** *Let* $f_{n,\mu}(t) = \|\nu_{n,t}(\mu) - \nu^*_n(\mu)\|_2^2$ *and*

$$\bar{\kappa}_n(t) = L_G \sqrt{\frac{\tau \log|\mathcal{A}| + 2\int_0^t g_{n+1}(s)\,\mathrm{d}s}{\lambda t}} + \frac{1}{t}\int_0^t g_{n+1}(s)\,\mathrm{d}s.$$

Then $g_{N_T}(t) \le \tau \log|\mathcal{A}|/t$ *and, for* $n < N_T$,

$$g_n(t) \le \frac{\tau \log|\mathcal{A}|}{t} + 2\bar{\kappa}_n(t) + 3L_G \sup_\mu \sqrt{f_{n,\mu}(t)} + g_{n+1}(t). \tag{C.21}$$

Proof. At the terminal level, $Q^{\boldsymbol{\pi}}_{N_T}$ does not depend on the policy, so $y_{N_T,t} = t\,r$ and the regularized-leader inequality gives the bound. For $n < N_T$, apply (C.20): the perturbation contributes at most g_{n+1}; the achieved stage value is compared with the equilibrium stage value through (C.19) and its absolute-value counterpart; the cascade radius is bounded by $\bar{\kappa}_n$; and the supremum over cells is taken. □

Hypothesis (ENV) of Theorem 4 is (6.6): deterministic decreasing envelopes $F_n(t) \to 0$ with $\sup_\mu f_{n,\mu}(t) \le F_n(t)$. Given (ENV), set $\gamma_{N_T}(t) = \tau \log|\mathcal{A}|/t$ and let γ_n be the right side of (C.21) with g_{n+1} replaced by γ_{n+1} and $\sup_\mu \sqrt{f_{n,\mu}}$ by $\sqrt{F_n}$. Backward induction gives $g_n \le \gamma_n$, and $\gamma_n \to 0$: if $\gamma_{n+1} \to 0$, its integral is $o(t)$, so every term in the recursion vanishes. This proves the uniform value convergence in Theorem 4(b). Convergence of the master Q-functions follows from the value convergence and (C.8). The policies approach the equilibrium best-response faces because the averaged scores converge uniformly to the equilibrium stage payoffs by (C.18), so the softmax puts vanishing mass on every action that is suboptimal by a fixed margin.

Proposition 9 (Exploitability along the realized flow)**.** *Under Assumption 3 and (ENV),*

there is a constant C_1 depending only on the standing constants such that

$$\sup_{m\in\Delta_{\mathcal{X}}} \mathcal{E}^m(\boldsymbol{\pi}_t) \le (N_T+1)\,\frac{\tau\log|\mathcal{A}|}{t} + 2\sum_{k=0}^{N_T-1}\bar{\kappa}_k(t) + C_1\sum_{k=0}^{N_T-1}\sqrt{F_k(t)} \;\longrightarrow\; 0. \tag{C.22}$$

Proof. Fix m and freeze the realized flow of $\boldsymbol{\pi}_t$: $\mu_0 = m$, $\mu_{k+1} = \nu_{k,t}(\mu_k)$. Apply Lemma 2 to this fixed flow with $\boldsymbol{\sigma} = \boldsymbol{\pi}_t[m]$; the occupancy of the trace is the realized flow itself, so $\mathcal{E}^m(\boldsymbol{\pi}_t) \le \sum_k \max_x d_k(x)$. Let u_k be the best-response value against the realized flow and $\Delta_k = \|u_k - V_k^*(\cdot,\mu_k)\|_\infty$. Comparing the backward recursion (C.6) with (C.10), and using the L_V-Lipschitz bound of Theorem 5,

$$\Delta_k \le \Delta_{k+1} + L_V\sqrt{|\mathcal{X}|}\,\sqrt{f_{k,\mu_k}(t)}, \qquad \Delta_{N_T} = 0.$$

At each level $k < N_T$, the defect $\max_x d_k(x)$ is bounded by the stage gap (C.19) plus $2\Delta_{k+1}$; at the terminal level it is at most $\tau\log|\mathcal{A}|/t$. Summing over levels, unrolling the recursion for Δ_k, and inserting $f_{k,\mu_k} \le F_k$ gives (C.22). No convergence of the realized flow, and no pointwise policy limit within a tied face, is used; the right side does not depend on m. □

C.6 Closures of the envelope hypothesis and sharpness

Proposition 10 (Tie-free closure)**.** *If every equilibrium stage payoff vector $\widehat{G}(x,\cdot\,;\nu_n^*(\mu))$ has a unique maximizing action, then (ENV) holds with exponentially decaying envelopes, and Theorem 4(b) applies unconditionally.*

Proof sketch. Compactness of $\Delta_{\mathcal{X}}$ and continuity give a uniform action gap $\Delta > 0$ at equilibrium. Working backward, suppose envelopes hold above level n; then $E(t) = o(t)$ at level n, and (C.18) makes the averaged scores share the equilibrium argmax with gap at least $\Delta/2$ for all large t. The softmax then gives every suboptimal action mass at most $e^{-t\Delta/(2\tau)}$, so the image error at level n has a deterministic exponential envelope, closing the induction. □

Definition 2 (Gradient alignment)**.** The continuation value $V(\cdot, \nu)$ is gradient-aligned with a λ-strongly convex potential ψ if $\langle d,\, V(\cdot, \nu)\rangle = -D\psi(\nu)[d]$ for every tangent vector d of the simplex $\Delta_{\mathcal{X}}$.

Theorem 6 (Uniform descent envelope)**.** *Consider* (C.15) *with a gradient-aligned stage game and* $E(T) = \int_0^T \varepsilon(s)\,\mathrm{d}s$*. Then, uniformly in the frozen population* μ*,*

$$\frac{\lambda}{2}\, f(T) \le \inf_{0<S\le T} \Big\{ \frac{\tau \log|\mathcal{A}| + 2E(T)}{S} + \frac{1}{4\tau} \int_{T-S}^{T} \varepsilon(s)^2\,\mathrm{d}s \Big\}. \tag{C.23}$$

We state Theorem 6 without proof. The idea is that under gradient alignment the stage dynamics is mirror descent on the potential $\varphi_\mu(\beta) = \psi(\Phi(\mu, \beta))$, and a perturbation of size ε injects energy only through its variance under the softmax Jacobian, at rate $O(\varepsilon^2/\tau)$; combining the resulting differential inequality with the integral bound (C.17) and optimizing the averaging window gives (C.23). Three consequences follow. For $\varepsilon(t) = O(1/t)$, the bound is $O(\log T/T)$. At level $N_T - 1$ the perturbation is the terminal gap $\tau \log|\mathcal{A}|/t$, so own-congestion interactions $\bar{r}(x, \mu) = \theta_x(\mu(x))$ with θ_x strongly decreasing, which include crowd aversion $-\kappa\,\mu(x)$ and Lipschitz-smoothed exploration terms $-\log(\epsilon_0 + \mu(x))$, admit an unconditional envelope at level $N_T - 1$; these are the interaction types of the exploration and beach-bar environments of Chapter 6. In one population dimension ($|\mathcal{X}| = 2$), every continuous strongly monotone interaction field is gradient-aligned, since its single tangential component can be integrated; iterating (C.23) once more gives $F_{N_T-2}(t) = O\big((\log t)^{3/4} t^{-1/4}\big)$ and $g_{N_T-2}(t) = O\big((\log t)^{3/8} t^{-1/8}\big)$, so for $N_T \le 2$ all levels are covered and Proposition 9 yields the unconditional rate of Theorem 4(c). For longer horizons this envelope recursion degrades level by level and eventually stalls; the general strongly monotone, tied, long-horizon case remains open, and Theorem 4(b) is stated conditionally for exactly this reason.

Sharpness at $\lambda = 0$. Strong monotonicity cannot be dropped. Take a one-shot stage game on three actions whose payoff is an antisymmetric linear map $q \mapsto Aq$ plus a constant

vector placing the equilibrium q^* in the interior of the simplex; it is weakly monotone ($\lambda = 0$). Along entropy OMD, the Fenchel coupling to q^* has derivative $\langle q_t - q^*,\, A(q_t - q^*)\rangle = 0$ and is conserved, so a trajectory started away from q^* stays on a compact nontrivial level set and never converges; only its time average converges. This is a genuine last-iterate counterexample rather than a missing proof, and it is why Theorem 4 requires $\lambda > 0$.

Relation to the implementation. Everything above concerns the synchronous dynamics (6.5). The learner of Chapter 6 uses sampled flows and a shared network, and Chapter 7 adds common noise. Extending the doubling argument of Theorem 5 conditionally on a noise history appears plausible, since a strong ($\lambda > 0$) version of the conditional monotonicity condition (7.3) would supply the analog of (C.4) along each realization, but the required measurability and noise-uniform envelope arguments have not been carried out, and no such claim is made in this dissertation.

Bibliography

[1] Devansh R Agrawal and Dimitra Panagou. Multi-agent clarity-aware dynamic coverage with gaussian processes. *arXiv preprint arXiv:2403.17917*, 2024.

[2] Noha Almulla, Rita Ferreira, and Diogo Gomes. Two numerical approaches to stationary mean-field games. *Dynamic Games and Applications*, 7:657–682, 2017.

[3] Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Q-learning in regularized mean-field games. *Dynamic Games and Applications*, 13(1):89–117, 2023.

[4] Andrea Angiuli, Jean-Pierre Fouque, and Mathieu Laurière. Unified reinforcement Q-learning for mean field game and control problems. *Mathematics of Control, Signals, and Systems*, 34:217–271, 2022.

[5] Alain Bensoussan, Jens Frehse, and Phillip Yam. *Mean field games and mean field type control theory*, volume 101. Springer, 2013.

[6] Alain Bensoussan, KCJ Sung, Sheung Chi Phillip Yam, and Siu-Pang Yung. Linear-quadratic mean field games. *Journal of Optimization Theory and Applications*, 169:496–529, 2016.

[7] Ulrich Berger. Brown's original fictitious play. *Journal of Economic Theory*, 135(1):572–578, 2007.

[8] Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 498–507, 2017.

[9] Henk A. P. Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.

[10] George W Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1):374, 1951.

[11] Birgit Burmeister, Afsaneh Haddadi, and Guido Matylis. Application of multi-agent systems in traffic and transportation. *IEE Proceedings-Software*, 144(1):51–60, 1997.

[12] Piermarco Cannarsa, Rossana Capuani, and Pierre Cardaliaguet. Mean field games with state constraints: From mild to pointwise solutions of the PDE system. *Calculus of Variations and Partial Differential Equations*, 60(3):108, 2021.

[13] Pierre Cardaliaguet, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions. *The master equation and the convergence problem in mean field games:(ams-201)*. Princeton University Press, 2019.

[14] Pierre Cardaliaguet and Saeed Hadikhanloo. Learning in mean field games: the fictitious play. *ESAIM: Control, Optimisation and Calculus of Variations*, 23(2):569–591, 2017.

[15] René Carmona and François Delarue. *Probabilistic theory of mean field games with applications I-II*. Springer, 2018.

[16] René Carmona, François Delarue, and Aimé Lachapelle. Control of McKean–Vlasov dynamics versus mean field games. *Mathematics and Financial Economics*, 7:131–166, 2013.

[17] René Carmona, François Delarue, and Daniel Lacker. Mean field games with common noise. 2016.

[18] Federico S Cattivelli and Ali H Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on automatic control*, 55(9):2069–2084, 2010.

[19] Y. T. Chan, A. G. C. Hu, and J. B. Plant. A Kalman filter based tracking scheme with input estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(2):237–244, 1979.

[20] Tsang-Kai Chang, Kenny Chen, and Ankur Mehta. Resilient and consistent multirobot cooperative localization with covariance intersection. *IEEE Transactions on Robotics*, 38(1):197–208, 2021.

[21] Lingji Chen, Pablo O Arambel, and Raman K Mehra. Estimation under unknown correlation: Covariance intersection revisited. *IEEE Transactions on Automatic Control*, 47(11):1879–1882, 2002.

[22] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.

[23] Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.

[24] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.

[25] Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on automatic control*, 52(5):852–862, 2007.

[26] Kai Cui and Heinz Koeppl. Approximately solving mean field games via entropy-regularized deep reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1909–1917. PMLR, 2021.

[27] Mohamed Darouach and Michel Zasadzinski. Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica*, 33(4):717–719, 1997.

[28] Mohamed Darouach, Michel Zasadzinski, and Mohamed Boutayeb. Extension of minimum variance estimation for systems with unknown inputs. *Automatica*, 39:867–876, 2003.

[29] Cédric H David, David R Maidment, Guo-Yue Niu, Zong-Liang Yang, Florence Habets, and Victor Eijkhout. River network routing on the nhdplus dataset. *Journal of Hydrometeorology*, 12(5):913–934, 2011.

[30] Mohammad Reza Davoodi, Khashayar Khorasani, Heidar Ali Talebi, and Hamid Reza Momeni. Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems. *IEEE Transactions on Control Systems Technology*, 22(3):1061–1069, 2013.

[31] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

[32] Hugh F Durrant-Whyte, BYS Rao, and Huosheng Hu. Toward a fully decentralized architecture for multi-sensor data fusion. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1331–1336. IEEE, 1990.

[33] Romuald Elie, Thibaut Mastrolia, and Dylan Possamaï. A tale of a principal and many, many agents. *Mathematics of Operations Research*, 44(2):440–467, 2019.

[34] Alireza Emami, Rui Araujo, and Alireza Asvadi. Distributed simultaneous estimation of states and unknown inputs. *Systems & Control Letters*, 138:104660, 2020.

[35] Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Olivier Bachem, Rémi Munos, and Olivier Pietquin. Concave utility reinforcement learning: the mean-field game viewpoint. *arXiv preprint arXiv:2106.03787*, 2021.

[36] Steven Gillijns and Bart De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, 43(1):111–116, 2007.

[37] Steven Gillijns and Bart De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica*, 43(5):934–937, 2007.

[38] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[39] Ben Gorr, Alan Aguilar Jaramillo, Zida Wu, Wooyeong Cho, Kewei Cheng, Molly Stroud, Vinay Ravindra, Cedric David, Huilin Gao, Yizhou Sun, Ankur Mehta, George Allen, and Daniel Selva. Multi-instrument flood monitoring with a distributed, decentralized, dynamic and context-aware satellite sensor web. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023.

[40] Felix Govaers, Chee-Yee Chong, Shozo Mori, and Wolfgang Koch. Comparison of augmented state track fusion methods for non-full-rate communication. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 862–869. IEEE, 2015.

[41] Felix Govaers and Wolfgang Koch. An exact solution to track-to-track-fusion at arbitrary communication rates. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2718–2729, 2012.

[42] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Saeed Hadikhanloo. Learning in anonymous nonatomic games with applications to first-order mean field games. *arXiv preprint arXiv:1704.00378*, 2017.

[44] Saeed Hadikhanloo. *Learning in mean field games*. PhD thesis, Université Paris sciences et lettres, 2018.

[45] Saeed Hadikhanloo and Francisco J Silva. Finite mean field games: fictitious play and convergence to a first order continuous mean field game. *Journal de Mathématiques Pures et Appliquées*, 132:369–397, 2019.

[46] Shaoming He, Hyo-Sang Shin, Shuoyuan Xu, and Antonios Tsourdos. Distributed estimation over a low-cost sensor network: A review of state-of-the-art. *Information Fusion*, 54:21–43, 2020.

[47] Xingkang He, Wenchao Xue, and Haitao Fang. Consistent distributed state estimation with global observability over sensor network. *Automatica*, 92:162–172, 2018.

[48] Chien-Shu Hsieh. Robust two-stage Kalman filters for systems with unknown inputs. *IEEE Transactions on Automatic Control*, 45(12):2374–2378, 2000.

[49] Chien-Shu Hsieh. Extension of unbiased minimum-variance input and state estimation for systems with unknown inputs. *Automatica*, 45(9):2149–2153, 2009.

[50] Chien-Shu Hsieh. On the global optimality of unbiased minimum-variance state estimation for systems with unknown inputs. *Automatica*, 46(4):708–715, 2010.

[51] Chien-Shu Hsieh. Time-distributed multi-step delayed input and state estimation. *Automatica*, 112:108700, 2020.

[52] Chien-Shu Hsieh. Unbiased minimum-variance input and state estimation for systems with unknown inputs: An online solution. In *2020 International Automatic Control Conference (CACS)*, pages 1–6. IEEE, 2020.

[53] Yu Hua, Na Wang, and Keyou Zhao. Simultaneous unknown input and state estimation for the linear system with a rank-deficient distribution matrix. *Mathematical Problems in Engineering*, 2021:1–11, 2021.

[54] Minyi Huang, Peter E Caines, and Roland P Malhamé. Large-population cost-coupled lqg problems with nonuniform agents: individual-mass behavior and decentralized ε-nash equilibria. *IEEE transactions on automatic control*, 52(9):1560–1571, 2007.

[55] Alan Aguilar Jaramillo, Ben Gorr, Vinay Ravindra, Cedric David, Molly Stroud, Ankur Mehta, George Allen, Wooyeong Cho, Kewei Cheng, Huilin Gao, et al. Decentralized market-based observation assignment strategy for dynamic networks in sensor web mission concepts. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pages 4748–4751. IEEE, 2023.

[56] Simon J Julier and Jeffrey K Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, volume 4, pages 2369–2373. IEEE, 1997.

[57] Jari Kaipio and Erkki Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493–504, 2007.

[58] Ahmed Tashrif Kamal, Chong Ding, Bi Song, Jay A Farrell, and Amit K Roy-Chowdhury. A generalized kalman consensus filter for wide-area video networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7863–7869. IEEE, 2011.

[59] Ahmed Tashrif Kamal, Jay A Farrell, and Amit K Roy-Chowdhury. Information weighted consensus. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2732–2737. IEEE, 2012.

[60] Maryam Kamgarpour and Claire Tomlin. Convergence properties of a decentralized kalman filter. In *2008 47th IEEE Conference on Decision and Control*, pages 3205–3210. IEEE, 2008.

[61] James Kennedy, Airlie Chapman, and Peter M Dower. Generalized coverage control for time-varying density functions. In *2019 18th European Control Conference (ECC)*, pages 71–76. IEEE, 2019.

[62] Hamid Khaloozadeh and Ali Karsaz. Modified input estimation technique for tracking manoeuvring targets. *IET Radar, Sonar & Navigation*, 3(1):30–41, 2009.

[63] Solmaz S Kia, Bryan Van Scoy, Jorge Cortes, Randy A Freeman, Kevin M Lynch, and Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[64] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, and Agnieszka Grabska-Barwinska. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[65] Peter K Kitanidis. Unbiased minimum-variance linear state estimation. *Automatica*, 23(6):775–778, 1987.

[66] He Kong, Mao Shan, Daobilige Su, Yongliang Qiao, Asher Al-Azzawi, and Salah Sukkarieh. Filtering for systems subject to unknown inputs without a priori initial information. *Automatica*, 120:109122, 2020.

[67] He Kong, Mao Shan, Salah Sukkarieh, Tianshi Chen, and Wei Xing Zheng. Kalman filtering under unknown inputs and norm constraints. *Automatica*, 133:109871, 2021.

[68] He Kong and Salah Sukkarieh. An internal model approach to estimation of systems with arbitrary unknown inputs. *Automatica*, 108:108482, 2019.

[69] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(8):235–284, 2008.

[70] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese journal of mathematics*, 2(1):229–260, 2007.

[71] Mathieu Laurière, Sarah Perrin, Matthieu Geist, and Olivier Pietquin. Learning mean field games: A survey. *arXiv preprint arXiv:2205.12944*, 2022.

[72] Mathieu Laurière, Sarah Perrin, Sertan Girgin, Paul Muller, Ayush Jain, Theophile Cabannes, Georgios Piliouras, Julien Pérolat, Romuald Élie, and Olivier Pietquin. Scalable deep reinforcement learning algorithms for mean field games. In *International Conference on Machine Learning*, pages 12078–12095. PMLR, 2022.

[73] Sung G Lee, Yancy Diaz-Mercado, and Magnus Egerstedt. Multirobot control using time-varying density functions. *IEEE Transactions on robotics*, 31(2):489–493, 2015.

[74] Yung-Lung Lee and Yi-Wei Chen. Imm estimator based on fuzzy weighted input estimation for tracking a maneuvering target. *Applied mathematical modelling*, 39(19):5791–5802, 2015.

[75] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The world wide web conference*, pages 983–994, 2019.

[76] Ruoyu Lin and Magnus Egerstedt. Dynamic multi-target tracking using heterogeneous coverage control. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11103–11110. IEEE, 2023.

[77] Changqing Liu, Youqing Wang, Donghua Zhou, and Xiao Shen. Minimum-variance unbiased unknown input and state estimation for multi-agent systems by distributed cooperative filters. *IEEE Access*, 6:18128–18141, 2018.

[78] Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. Computing approximate equilibria in sequential

adversarial games by exploitability descent. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 464–470, 2019.

[79] E. Lourens, Edwin Reynders, Guido De Roeck, Geert Degrande, and Geert Lombaert. An augmented Kalman filter for force identification in structural dynamics. *Mechanical Systems and Signal Processing*, 27:446–460, 2012.

[80] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[81] Peng Lu, Laurens Van Eykeren, Erik-Jan van Kampen, Coen C. de Visser, and Qiping Chu. Double-model adaptive fault detection and diagnosis applied to real flight data. *Control Engineering Practice*, 36:39–57, 2015.

[82] Wenhao Luo, Changjoo Nam, George Kantor, and Katia Sycara. Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1488–1496, 2019.

[83] Kangjian Ma, Shaochuan Wu, Yuming Wei, and Wenbin Zhang. Gossip-based distributed tracking in networks of heterogeneous agents. *IEEE Communications Letters*, 21(4):801–804, 2016.

[84] D. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, 1965.

[85] Lorenzo Magnino, Kai Shao, Zida Wu, Jiacheng Shen, and Mathieu Laurière. Solving continuous mean field games: Deep reinforcement learning for non-stationary dynamics. In *Advances in Neural Information Processing Systems*, volume 38, pages 104325–104354, 2025.

[86] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, and Georg Ostrovski. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[87] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[88] Wolfgang Niehsen. Information fusion based on fast covariance intersection filtering. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, volume 2, pages 901–904. IEEE, 2002.

[89] NOAA Office of Water Prediction. NOAA national water model CONUS retrospective dataset. Registry of Open Data on AWS, `https://registry.opendata.aws/nwm-archive/`, 2023. National Water Model v3.0 retrospective simulation, February 1979–January 2023; accessed 2026-07-05.

[90] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

[91] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[92] Reza Olfati-Saber and Nils F Sandell. Distributed tracking in sensor networks with limited sensing range. In *2008 American Control Conference*, pages 3157–3162. IEEE, 2008.

[93] Kei Ota, Devesh K Jha, and Asako Kanezaki. Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*, 2021.

[94] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1-2):100–115, 1954.

[95] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *International Conference on Machine Learning*, pages 4045–4054. PMLR, 2018.

[96] Zhaoxia Peng, Yingwen Zhang, Guoguang Wen, Shichun Yang, and Tingwen Huang. Optimal distributed filter for state and unknown input estimations in multi-sensor networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[97] Julien Perolat, Sarah Perrin, Romuald Elie, Mathieu Laurière, Georgios Piliouras, Matthieu Geist, Karl Tuyls, and Olivier Pietquin. Scaling mean field games by online mirror descent. *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1028–1037, 2022.

[98] Sarah Perrin, Mathieu Laurière, Julien Pérolat, Romuald Élie, Matthieu Geist, and Olivier Pietquin. Generalization in mean field games by learning master policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9413–9421, 2022.

[99] Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. *Advances in Neural Information Processing Systems*, 33:13199–13213, 2020.

[100] Antonio Petitti, Donato Di Paola, Alessandro Rizzo, and Grazia Cicirelli. Consensus-based distributed estimation for target tracking in heterogeneous sensor networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6648–6653. IEEE, 2011.

[101] Ryan R. Pitre, Vesselin P. Jilkov, and X. Rong Li. A comparative study of multiple-model algorithms for maneuvering target tracking. In *Signal Processing, Sensor Fusion,*

and Target Recognition XIV, volume 5809, pages 549–560. International Society for Optics and Photonics, 2005.

[102] Federico Pratissoli, Mattia Mantovani, Amanda Prorok, and Lorenzo Sabattini. Distributed coverage control for time-varying spatial processes. *IEEE Transactions on Robotics*, 2025.

[103] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[104] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[105] Lars Ruthotto, Stanley J. Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[106] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[107] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[108] Dan Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.

[109] Dan Simon and Tien Li Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

[110] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems 21 (NIPS)*, 2008.

[111] Jayakumar Subramanian and Aditya Mahajan. Reinforcement learning in stationary mean-field games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 251–259, 2019.

[112] S. A. Taher, Jian Li, and Huazhen Fang. Earthquake input and state estimation for buildings using absolute floor accelerations. *Earthquake Engineering & Structural Dynamics*, 50(4):1020–1042, 2021.

[113] Vahid Vahidpour, Amir Rastegarnia, Azam Khalili, and Saeid Sanei. Partial diffusion kalman filtering for distributed state estimation in multiagent networks. *IEEE transactions on neural networks and learning systems*, 30(12):3839–3846, 2019.

[114] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

[115] Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen reinforcement learning. *Advances in Neural Information Processing Systems*, 33:4235–4246, 2020.

[116] A. Viros Martin, K. Cheng, A. Fang, Z. Zheng, H. Kress-Gazit, A. Mehta, D. Selva, and Y. Sun. Decentralized context-based on-board planning for earth observation missions. In *AIAA Scitech 2021 Forum*, page 1469, 2021.

[117] Jan Vondrák. Submodularity and curvature: The optimal algorithm. *RIMS Kôkyûroku Bessatsu*, B23:253–266, 2010.

[118] Shirantha Welikala and Christos G. Cassandras. Performance-guaranteed solutions for multi-agent optimal coverage problems using submodularity, curvature, and greedy algorithms. *arXiv preprint arXiv:2403.14028*, 2024.

[119] Zida Wu, Wooyeong Cho, David Martinez, and Ankur Mehta. Discretisation-induced curvature inflation in submodular coverage. Under review at Transactions on Machine Learning Research, 2026.

[120] Zida Wu, Mathieu Laurière, Samuel Jia Cong Chua, Matthieu Geist, Olivier Pietquin, and Ankur Mehta. Population-aware online mirror descent for mean-field games by deep reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2561–2563, 2024.

[121] Zida Wu, Mathieu Laurière, Samuel Jia Cong Chua, Matthieu Geist, Olivier Pietquin, and Ankur Mehta. Population-aware online mirror descent for mean-field games by deep reinforcement learning. arXiv preprint arXiv:2403.03552, 2024.

[122] Zida Wu, Mathieu Laurière, Matthieu Geist, Olivier Pietquin, and Ankur Mehta. Population-aware online mirror descent for mean-field games with common noise by deep reinforcement learning. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 6338–6345. IEEE, 2025.

[123] Zida Wu and Ankur Mehta. Decentralized input and state estimation for multi-agent system with dynamic topology and heterogeneous sensor network. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 2740–2747, Milan, Italy, 2024. IEEE.

[124] Zida Wu and Ankur Mehta. Mobile sensor coverage beyond the input-rank condition: Active input localization on tree networks. Manuscript in preparation, for submission to IEEE Robotics and Automation Letters, 2026.

[125] Zida Wu, Zhaoliang Zheng, and Ankur Mehta. Joint state and input estimation of agent based on recursive kalman filter given prior knowledge. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1675–1681. IEEE, 2022.

[126] Zirui Xu, Hongyu Zhou, and Vasileios Tzoumas. Online submodular coordination

with bounded tracking regret: Theory, algorithm, and applications to multi-robot coordination. *IEEE Robotics and Automation Letters*, 8(4):2261–2268, 2023.

[127] Sze Zheng Yong, Minghui Zhu, and Emilio Frazzoli. A unified filter for simultaneous input and state estimation of linear discrete-time stochastic systems. *Automatica*, 63:321–329, 2016.

[128] Sze Zheng Yong, Minghui Zhu, and Emilio Frazzoli. Simultaneous mode, input and state estimation for switched linear stochastic systems. *International Journal of Robust and Nonlinear Control*, 31(2):640–661, 2021.