Received 8 February 2023; revised 27 April 2023; accepted 2 May 2023. Date of publication 18 May 2023;
date of current version 20 June 2023. Recommended by Senior Editor Lacra Pavel.

Digital Object Identifier 10.1109/OJCSYS.2023.3277760

Policy Evaluation in Decentralized POMDPs With Belief Sharing

MERT KAYAALP [1] (Graduate Student Member, IEEE), FATIMA GHADIEH [2],
AND ALI H. SAYED [1] (Fellow, IEEE)

(Intersection of Machine Learning With Control)

[1] Adaptive Systems Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
[2] American University of Beirut, Beirut 1107 2020, Lebanon

CORRESPONDING AUTHOR: MERT KAYAALP (e-mail: mert.kayaalp@epfl.ch).

The work of Fatima Ghadieh was performed while she was a Student Intern at the Adaptive Systems Laboratory, EPFL.

ABSTRACT Most works on multi-agent reinforcement learning focus on scenarios where the state of the environment is fully observable. In this work, we consider a cooperative policy evaluation task in which agents are not assumed to observe the environment state directly. Instead, agents can only have access to noisy observations and to belief vectors. It is well-known that finding global posterior distributions under multi-agent settings is generally NP-hard. As a remedy, we propose a fully decentralized belief forming strategy that relies on individual updates and on localized interactions over a communication network. In addition to the exchange of the beliefs, agents exploit the communication network by exchanging value function parameter estimates as well. We analytically show that the proposed strategy allows information to diffuse over the network, which in turn allows the agents' parameters to have a bounded difference with a centralized baseline. A multi-sensor target tracking application is considered in the simulations.

INDEX TERMS Belief state, distributed state estimation, multi-agent reinforcement learning, partially observable Markov decision process, value function learning.

I. INTRODUCTION

Multi-agent reinforcement learning (MARL) [1], [2] is a useful paradigm for determining optimal policies in sequential decision making tasks involving a group of agents. MARL has been applied successfully in several contexts, including sensor networks [3], [4], team robotics [5], and video games [6], [7]. MARL owes this success in part to recent developments in better function approximators such as deep neural networks [8].

Many works on MARL focus on the case where agents can directly observe the global state of the environment. However, in many scenarios, agents can only receive partial information about the state. The decentralized partially observable Markov decision process (Dec-POMDP) framework [9] is applicable to these types of situations. However, a large body of MARL work assumes that Dec-POMDPs observe data that are deterministic and known functions of the underlying state, which is not the case in general. Consider, for example, robots that receive noisy observations from their sensors. The underlying observation model is stochastic in this case.

Under stochastic observation models, one common strategy is to keep track of the posterior distribution (belief) over the set of states, which is known to be a sufficient statistic of the history of the system [10], [11]. For single agents, this posterior distribution can be obtained at each iteration with the optimal Bayesian filtering recursion [12]. Unfortunately, for multi-agent systems, forming this global posterior belief requires aggregation of all data from across all agents in general. The agents can form it in a distributed manner only when they have access to the private information from other agents in the network. And even when agents have access to this level of global knowledge, the computational complexity of forming the global posterior distribution is known to be NP-hard [13] in addition to its large memory requirements. Moreover, obtaining beliefs necessitates significant knowledge about the underlying model of the environment, which is generally not available in practice.

Therefore, instead of forming beliefs, most MARL algorithms [14], [15], [16] resort to a model-free and end-to-end approach where agents try to simultaneously learn a policy and an embedding of the history that can replace the beliefs (e.g., recurrent neural networks (RNNs)). Nevertheless, recent empirical works suggest that this model-free approach can be sub-optimal when the underlying signals of the environment are too weak to train a model such as RNN [17], [18]. Moreover, RNNs (or alternative machine learning models) are usually treated as black boxes. In other words, these algorithms lack model interpretability, which is critical for trustworthy systems (see [19]). Furthermore, even though end-to-end approaches have shown remarkable performance empirically, they are still based on heuristics and lack theoretical guarantees on their performance. Compared to modular approaches, they are inefficient in terms of adaptability and generalization to similar tasks.

As an alternative, there is a recent interest towards improving belief-based MARL approaches [20], [21], [22]. These works have focused on emulating conventional beliefs with generative models, or with models learned from action/observation trajectories (in a supervised fashion). In this paper, we also examine belief-based strategies for MARL. In particular, we are interested in the multi-agent policy evaluation problem. Our work complements [20], [21], [22] in the sense that we assume that agents are already capable of forming *local* beliefs with sufficient knowledge (i.e., with learned *local* likelihood and transition models) or with generative models. Our focus is on the challenge of approximating the *global* Bayesian posterior in a *distributed* manner.

Contributions:

- We consider a setting where agents only get partial observations from the underlying state of nature, as opposed to prior work on MARL over networks [23], [24], [25], [26], [27], [28], [29], [30], [31] that assume agents have full state information. Moreover, as opposed to the literature on decentralized stochastic control [32], [33], [34], [35], in our setting, agents need to learn their value functions from data. More specifically, in our Dec-POMDP framework, agents only know their local observations, actions, and rewards but they are allowed to communicate with their immediate neighbors over a graph. In the proposed strategy (Algorithm 2), agents exchange both their belief and value function estimates.
- We show in Theorem 1 that by exchanging beliefs, agents keep a bounded disagreement with the global posterior distribution, which requires fusing all observations and actions. Also, exchanging value function parameters enables agents to cluster around the network centroid for sufficiently small learning rates (Theorem 2). Furthermore, we prove that the network centroid attains a bounded difference with a strategy that requires centralized training (Theorem 3).
- By means of simulations, we illustrate that agents attain a small mean-square distance from the network centroid. Moreover, the squared Bellman error (SBE) averaged over the network is shown to be comparable to the SBE of the centralized strategy.

Paper Organization: In Section II, we present additional related work. In Section III, for ease of exposition and introducing notation, we describe the problem in single-agent setting. In Section IV, we propose algorithms for multi-agent policy evaluation. Section V includes the theoretical results, and Section VI includes numerical simulations.

II. OTHER RELATED WORK

Our proposed strategy is based on temporal-difference (TD) learning [36], [37], and makes use of function approximation. TD-learning for POMDPs are considered in [38], [39], and function approximations are incorporated in [40], [41], albeit in single-agent setting. The main contribution of the present work is to the networked multi-agent setting.

A plethora of work studies decentralized policy evaluation over networks [23], [24], [25], [26], [27], [28], [29], [30], [31]. Distributed versions of the TD-learning with linear function approximations are considered in [29], [30], [31]. However, these works assume that either the global state, or a deterministic function of it, is available to all agents. They overlook the stochastic nature of observations that takes place in many real-world applications. Also in deterministic setting, the works [42], [43] examine distributed linear quadratic control task when agents can observe local states only. In particular, [43] proposes a cooperative strategy for tracking the global state that exploits networked communication between agents. However, in this strategy, global state estimation at each iteration is independent of the previous estimations. It ignores the correlation between consecutive states. Furthermore, communication between the agents is utilized only for global state estimation, and not utilized for local Q-function estimate sharing. In contrast, in the present work, (i) observations are stochastic, (ii) agents take advantage of the transition model of the state, and (iii) they exchange value function parameters with their neighbors as well.

Our work is also related to the field of decentralized stochastic control [32] and dynamic team theory [33]. This field studies problems in which different decision-makers have access to different sets of information while working towards a common team goal. Typically, these problems are defined by an information structure that specifies which agents have access to which pieces of information (e.g., observations or actions) [44], [45]. Some approaches to solving these problems rely on the common information that arises from partial history sharing to *all* other agents [35], [46], [47]. In our networked setting, agents exchange value function parameters or beliefs at each iteration, without explicitly exchanging raw data, with their *immediate* neighbors only. Nonetheless, repeated application of this procedure causes information to mix and diffuse throughout the whole network. Moreover, most existing works in the decentralized stochastic control literature assume full model knowledge of the system, whereas we consider the case of learning from data since the reward model is not known a priori. Also, sharing value function parameters and beliefs instead of raw data makes our algorithm advantageous in terms of privacy and scalability. A similar approach is considered in [34], where the author proposes a belief-sharing pattern for decentralized control, rather than explicit information sharing as in prior work. However, they use a belief propagation algorithm over acyclic graphs, while we use a diffusion-based belief-sharing algorithm over cyclic networks. In addition, [34] considers the planning problem only whereas in this work we consider the policy evaluation problem, which requires learning from data.

For constructing local beliefs that approximate the global Bayesian posterior, we extend the diffusion HMM strategy (DHS) [48], [49]. This algorithm requires only one round of communication per state change, as opposed to other strategies [50], [51] that require multiple rounds of communication until network consensus at each iteration. Also, in contrast to other distributed Bayesian filtering algorithms [52], it does not combine likelihoods of data from different time instants. Instead, likelihoods are combined with *time-adjusted* beliefs. These properties make DHS communication efficient and successful in tracking highly dynamic state transitions. Note that [48], [49] deal with state estimation task only, and there are no rewards or actions in their setting. Therefore, we make proper modifications to the algorithm in the sequel.

In addition to these, the analysis in the current work is related to literature on the distributed optimization over networks [53], [54], [55], [56]. In particular, we adopt the two-step approach from [57], [58], [59]. In the first step, these works establish that agents cluster around the network centroid, and then, they show that this centroid converges to a neighborhood of the optimal solution, under constant learning rates. However, their focus is on optimization and supervised learning rather than reinforcement learning, which creates non-trivial distinctions in the analysis.

Notation: Random variables are denoted in bold. For K vectors $w_1, w_2, \ldots, w_K \in \mathbb{R}^M$ of dimension $M \times 1$ each, and for arbitrary matrices $\{A, B\}$, the notation $\text{col}\{w_k\}_{k=1}^K$ and $\text{diag}\{A, B\}$ stand for

$$\text{col}\{w_k\}_{k=1}^K = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}, \quad \text{diag}\{A, B\} = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}. \tag{1}$$

The ℓ_p-norm for a vector w is represented by $\|w\|_p$, while the ℓ_p-induced norm for a matrix A is represented by $\|A\|_p$. To simplify the notation, we use $\|w\|$ and $\|A\|$ to denote the ℓ_2-norm, without explicitly stating the subscript. The all-ones vector of dimension K is denoted by $\mathbb{1}_K$. The symbol $\otimes$ represents the Kronecker product. The Kullback-Leibler divergence [60] between two distributions μ_1, μ_2 is denoted by $D_{\text{KL}}(\mu_1||\mu_2)$. We use the notation "proportional to", i.e., $\propto$, whenever the LHS of the expression is the normalized version of the RHS. For example, for $s \in \mathcal{S}$ and function f:

$$\mu(s) \propto f(s) \Longleftrightarrow \mu(s) = \frac{f(s)}{\sum_{s' \in \mathcal{S}} f(s')}. \tag{2}$$

III. PRELIMINARIES

In this work, we are interested in multi-agent policy evaluation under partially observable stochastic environments. For clarity of the exposition and to motivate the notation, we briefly review the procedure of single-agent policy evaluation under both fully and partially observable states.

A. FULLY-OBSERVABLE CASE

For modeling a learning agent under fully observable and dynamic environments, the traditional setting is a finite Markov Decision Process (MDP). An MDP is defined by the quintuple $(\mathcal{S}, \mathcal{A}, \mathbb{T}, \boldsymbol{r}, \gamma)$, where $\mathcal{S}$ is a set of states with cardinality $|\mathcal{S}| = S$, $\mathcal{A}$ is a set of actions, $\mathbb{T}$ is a transition model where $\mathbb{T}(s'|a, s)$ denotes the probability of transitioning from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ when the agent executes action $a \in \mathcal{A}$, $\boldsymbol{r}(s, a, s')$ denotes the reward the agent receives when it executes action a and the environment transitions from state s to s', and $\gamma \in [0, 1)$ is a discount factor that determines the importance given to immediate rewards ($\gamma \to 0$) or the total reward ($\gamma \to 1$).

The goal of policy evaluation is to learn the value function $V^\pi(s)$ of a target policy $\pi(a|s)$, where the value function is the expected return if the agent starts from state s and follows policy π, namely,

$$V^\pi(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \boldsymbol{r}(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1}) | \boldsymbol{s}_0 = s\right], \tag{3}$$

where $\boldsymbol{s}_i$ is the state at time i and $\boldsymbol{a}_i$ is the action chosen by the agent according to the policy, $\boldsymbol{a}_i \sim \pi(a|\boldsymbol{s}_i)$. In many applications, the state space is too large (or infinite), which makes it impractical to keep track of the value function for all states. Therefore, function approximations are used to reduce the dimension of the problem. For instance, linear approximations, which are the focus of the theoretical analysis of this work, correspond to using a parameter $w^\circ \in \mathbb{R}^M$ to approximate $V^\pi(s) \approx \phi(s)^\mathsf{T} w^\circ$, where $\phi : \mathcal{S} \to \mathbb{R}^M$ is a *pre-defined* feature mapping for representing state s.

A standard stochastic approximation algorithm to learn w° from data is TD-learning [19], [36] such as the TD(0) strategy [61] and variations thereof. If we denote the value function estimate at $w \in \mathbb{R}^M$ by $\widehat{V}(s, w) \triangleq \phi(s)^\mathsf{T} w$, then, under this strategy, the agent first computes the TD-error $\boldsymbol{\delta}_i$ at time i by using the observed transition tuple $(\boldsymbol{s}_i, \boldsymbol{r}_i, \boldsymbol{s}_{i+1})$:

$$\boldsymbol{\delta}_i = \boldsymbol{r}_i + \gamma \widehat{V}(\boldsymbol{s}_{i+1}, \boldsymbol{w}_i) - \widehat{V}(\boldsymbol{s}_i, \boldsymbol{w}_i), \tag{4}$$

where $\boldsymbol{r}_i \triangleq \boldsymbol{r}(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1})$ is the instantaneous reward at time i. Subsequently, the agent uses this error to update the current parameter estimate $\boldsymbol{w}_i$ to

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \alpha \boldsymbol{\delta}_i \nabla_w \widehat{V}(\boldsymbol{s}_i, \boldsymbol{w}_i), \tag{5}$$

where $\alpha > 0$ is the learning rate, and

$$\nabla_w \widehat{V}(\boldsymbol{s}_i, \boldsymbol{w}_i) = \phi(\boldsymbol{s}_i) \tag{6}$$

for the linear function approximation case. This algorithm can be viewed as a "stochastic gradient algorithm" where the effective stochastic gradient is $\boldsymbol{g}_i \triangleq -\boldsymbol{\delta}_i \phi(\boldsymbol{s}_i)$. In this work, we consider an ℓ_2-regularized version of the algorithm, which changes the update step (5) to

$$\boldsymbol{w}_{i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_i + \alpha \boldsymbol{\delta}_i \nabla_w \widehat{V}(\boldsymbol{s}_i, \boldsymbol{w}_i), \tag{7}$$

where $\rho > 0$ is a constant hyper-parameter. As opposed to supervised learning, regularization is rather under-explored in reinforcement learning, with notable exceptions in [62], [63]. However, recent work [64], [65] suggests that regularization can increase generalization and sample-efficiency in function approximation with over-parameterized models.

B. PARTIALLY-OBSERVABLE CASE

In many applications, the agent does not directly observe the state $\boldsymbol{s}_i$. For instance, a robot may receive noisy and partially informative observations from its sensors about the environment. The observation $\boldsymbol{\xi}_i$ that the agent receives at time i is generally assumed to be distributed according to some likelihood function linking it to the unobservable state, say, $\boldsymbol{\xi}_i \sim L(\xi|\boldsymbol{s}_i)$, which is conditioned on $\boldsymbol{s}_i$. In these scenarios, the agent will need to estimate the latent state first from the observations. To do so, the agent will need to learn a probability vector $\boldsymbol{\mu}_i \in \mathcal{M}(S)$ over the set of states $\mathcal{S}$, which is called the *belief* vector [10], [19]. Here, $\mathcal{M}(S)$ denotes the S-dimensional probability simplex, and the entry $\boldsymbol{\mu}_i(s) \in [0, 1]$ of the belief vector quantifies the confidence the agent has about state s being the true state at time i. The value of $\boldsymbol{\mu}_i(s)$ corresponds to the posterior probability of s conditioned on the action-observation history (a.k.a. trajectory):

$$\boldsymbol{\mathcal{F}}_i \triangleq \{\boldsymbol{\xi}_i, \boldsymbol{a}_{i-1}, \boldsymbol{\xi}_{i-1}, \dots\}, \tag{8}$$

which means:

$$\boldsymbol{\mu}_i(s) \triangleq \mathbb{P}(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}_i). \tag{9}$$

This posterior satisfies the following temporal recursion [10], [12], [19]:

$$\boldsymbol{\mu}_i(s) \propto L(\boldsymbol{\xi}_i|s)\boldsymbol{\eta}_i(s), \tag{10}$$

where $\boldsymbol{\eta}_i(s)$ is the time-adjusted prior defined by

$$\boldsymbol{\eta}_i(s) \triangleq \mathbb{P}(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}^a_{i-1}) = \sum_{s' \in \mathcal{S}} \mathbb{T}(s|s', \boldsymbol{a}_{i-1})\boldsymbol{\mu}_{i-1}(s'). \tag{11}$$

Here, $\boldsymbol{\mathcal{F}}^a_{i-1}$ is the collection of past observations and actions, i.e.,

$$\boldsymbol{\mathcal{F}}^a_{i-1} \triangleq \{\boldsymbol{a}_{i-1}, \boldsymbol{\xi}_{i-1}, \boldsymbol{a}_{i-2}, \dots\}, \tag{12}$$

where it is important to notice that $\boldsymbol{\mathcal{F}}_i = \{\boldsymbol{\xi}_i\} \cup \boldsymbol{\mathcal{F}}^a_{i-1}$. If beliefs are used as substitutes for hidden states, then partially-observable MDPs (POMDPs) can be treated as *continuous* MDPs, since beliefs are continuous even if the number of states is finite. In this way, the policy evaluation problem would correspond to evaluating $V^\pi(\mu)$ where the value function is now defined as the expected return when the agent starts from the belief state μ and follows the policy $\pi(a|\mu)$, namely [10], [19]:

$$V^\pi(\mu) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \boldsymbol{r}_i | \boldsymbol{\mu}_0 = \mu\right]. \tag{13}$$

Observe that, in contrast to the fully-observable case, the agent now chooses action $\boldsymbol{a}_i$ according to the policy $\boldsymbol{a}_i \sim \pi(a|\boldsymbol{\mu}_i)$, which is conditioned on the belief vector.

Algorithm (4)–(7) can be adjusted for POMDPs by using the belief vectors $(\boldsymbol{\mu}_i, \boldsymbol{\eta}_{i+1})$ instead of the states $(\boldsymbol{s}_i, \boldsymbol{s}_{i+1})$. Thus, we let

$$\boldsymbol{\delta}_i = \boldsymbol{r}_i + \gamma \widehat{V}(\boldsymbol{\eta}_{i+1}, \boldsymbol{w}_i) - \widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i), \tag{14}$$

and

$$\boldsymbol{w}_{i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_i + \alpha \boldsymbol{\delta}_i \nabla_w \widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i), \tag{15}$$

where the approximations $\widehat{V}(\mu, w)$ are computed by using the feature vectors $\phi(\mu)$, now dependent on μ, to evaluate $\widehat{V}(\mu, w) \triangleq \phi(\mu)^\mathsf{T} w$. Note that from now on $\phi : \mathcal{M}(S) \rightarrow \mathbb{R}^M$ is a different feature mapping that represents μ instead of s, and agents' goal is to learn w° that satisfies $V^\pi(\mu) \approx \phi(\mu)^\mathsf{T} w^\circ$.

Observe from (10)–(11) that in order for the agent to update the belief vectors $(\boldsymbol{\mu}_i, \boldsymbol{\eta}_{i+1})$, it needs to know the transition model $\mathbb{T}$ and the likelihood functions $L(\boldsymbol{\xi}_i|s)$ for each state. However, the agent does not need to know the underlying reward model $\boldsymbol{r}$. It can use instantaneous reward samples $\boldsymbol{r}_i$ to run the algorithm. In this sense, the algorithm is a mixture of model-based and model-free reinforcement learning. Motivation for this approach is at least two-fold. First, in some applications, learning the transition and observation models from data is inherently easier than learning the reward function. This is because the reward function can depend on some latent characteristics of the environment or some human expert, which may be challenging to estimate. One example where this scenario can arise is autonomous cars [66]. In this case, the observations from environmental sensors and cameras are processed with a learned likelihood model such as a convolutional neural network. The transition dynamics of the car depends on various parameters such as speed, acceleration, position, and incline, and can be modeled based on physics laws and mapping of the surroundings. However, learning a reward function for this application is notoriously difficult, as it is challenging to cover all possible situations [67]. Second, the agent can still run (14)–(15) even if beliefs are not formed through (10)–(11), but estimated by some other approach, as in [20], [21], [22].

IV. MULTI-AGENT POLICY EVALUATION

We now consider a set $\mathcal{K}$ of K cooperative agents that aim to evaluate the average value function under a joint policy $\pi = \{\pi_k\}_{k=1}^K$ that consists of individual policies π_k. The framework we consider is a *decentralized* POMDP (Dec-POMDP) [9], which is defined by the sextuple $(\mathcal{S}, \mathcal{A}_k, \mathcal{O}_k, \mathbb{T}, \boldsymbol{r}_k, \gamma)$. Here, the set of states $\mathcal{S}$ and the transition model $\mathbb{T}$ are common to all agents, where the notation $\mathbb{T}(s|s', a)$ now specifies the probability that the environment transitions from s' to s when the agents execute the joint action $a = \{a_k\}_{k=1}^{K}$. The individual action a_k of each agent k takes values from the set $\mathcal{A}_k$, and $\boldsymbol{r}_k(s, a, s')$ is the *local* reward k gets when the agents execute the collection of actions a and the environment transitions from s to s'. Note that this setting covers general teamwork scenarios where the local reward of an individual agent can be dependent on all actions, and not only on its own actions. Specifically, it covers the scenarios that all agents observe the same reward, i.e., $\boldsymbol{r}_k(s, a, s') = \boldsymbol{r}(s, a, s'), \forall k \in \mathcal{K}$. Remember that agents receive instantaneous rewards as they progress through the POMDP, and they are not required to know the joint action a from all agents. Moreover, $\mathcal{O}_k$ is a set of *private* observations. At each time instant i, agent k receives observation $\boldsymbol{\xi}_{k,i} \in \mathcal{O}_k$ emitted by state $\boldsymbol{s}_i$, and assumed to be distributed according to the local *marginal* likelihood $L_k(\xi_k|\boldsymbol{s}_i)$.

Similar to the single-agent case, Dec-POMDPs can be treated as multi-agent belief MDPs by replacing the hidden states with joint centralized beliefs defined by [9, Chap. 2]

$$\boldsymbol{\mu}_i(s) \triangleq \mathbb{P}(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}_i) \propto L(\boldsymbol{\xi}_i|s)\boldsymbol{\eta}_i(s). \tag{16}$$

Here, $\boldsymbol{\mathcal{F}}_i$ denotes the history of all observations and past actions from across all agents until time i, where in the definition (8), $\boldsymbol{\xi}_i \triangleq \{\boldsymbol{\xi}_{k,i}\}_{k=1}^{K}$ is now the aggregate of the observations from across the network, and $\boldsymbol{a}_{i-1}$ is a tuple aggregating actions from all agents at time $i-1$. Moreover, under spatial independence, the joint likelihood $L(\boldsymbol{\xi}_i|s)$ appearing in (16) is given by

$$L(\boldsymbol{\xi}_i|s) = \prod_{k=1}^{K} L_k(\boldsymbol{\xi}_{k,i}|s). \tag{17}$$

In a manner similar to the single-agent case, the belief $\boldsymbol{\eta}_i(s)$ is the time-adjusted prior conditioned on $\boldsymbol{\mathcal{F}}_{i-1}^{a}$ (12):

$$\boldsymbol{\eta}_i(s) \triangleq \mathbb{P}\left(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}_{i-1}^{a}\right) = \sum_{s' \in \mathcal{S}} \mathbb{T}(s|s', \boldsymbol{a}_{i-1})\boldsymbol{\mu}_{i-1}(s'). \tag{18}$$

The goal of policy evaluation is to learn the *team* value function, which is the expected average reward of all agents starting from some belief state μ, i.e.,

$$V^{\pi}(\mu) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \left(\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{r}_{k,i}\right) |\boldsymbol{\mu}_0 = \mu\right], \tag{19}$$

where $\boldsymbol{r}_{k,i}$ denotes the instantaneous local reward agent k gets at time i.

There is one major inconvenience with this approach. In order to compute the joint belief (16), it is necessary to fuse all observations and actions from across the agents in a central location. This is possible in settings where there exists a fusion center. However, many applications rely solely on localized processing. In the following, we discuss and compare two strategies for multi-agent reinforcement learning under partial observations: (i) a centralized strategy, (ii) and a fully decentralized strategy.

A. CENTRALIZED STRATEGY

In the fully centralized strategy, the state estimation and policy evaluation phases are centralized and, hence, the setting is equivalent to a single-agent POMDP, already discussed in Section III-B, using the joint likelihood $L(\boldsymbol{\xi}_i|s)$ and the average reward $\boldsymbol{r}_i \triangleq K^{-1}\sum_{k=1}^{K} \boldsymbol{r}_{k,i}$. The fusion center computes the joint belief (16), and agents take actions based on this joint belief, i.e., $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_i)$. The fusion center then computes the centralized TD-error:

$$\boldsymbol{\delta}_i = \boldsymbol{r}_i + \gamma\widehat{V}(\boldsymbol{\eta}_{i+1}, \boldsymbol{w}_i) - \widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i), \tag{20}$$

and updates the estimate to

$$\boldsymbol{w}_{i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_i + \alpha\boldsymbol{\delta}_i\nabla_w\widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i). \tag{21}$$

This construction is listed under Algorithm 1.

Algorithm 1: Centralized Policy Evaluation Under POMDPs.

```
set initial prior $\eta_0(s) > 0, \forall s \in \mathcal{S}$
initialize $w_0$
**while** $i \geq 0$ **do**
    each agent $k$ observes $\boldsymbol{\xi}_{k,i}$
    collect all observations $\boldsymbol{\xi}_i \triangleq \{\boldsymbol{\xi}_{k,i}\}_{k=1}^{K}$ and **evaluate**
        $\boldsymbol{\mu}_i(s) \propto L(\boldsymbol{\xi}_i|s)\boldsymbol{\eta}_i(s)$    (22)
    **for** each agent $k \in \mathcal{K}$ **do**
        Take action $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_i)$
        Get reward $\boldsymbol{r}_{k,i} = \boldsymbol{r}_k(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1})$
    **end for**
    then, **evolve**
        $\boldsymbol{\eta}_{i+1}(s) = \sum_{s' \in \mathcal{S}} \mathbb{T}(s|s', \boldsymbol{a}_i)\boldsymbol{\mu}_i(s')$    (23)
    **average** the rewards $\boldsymbol{r}_i = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{r}_{k,i}$
    **update** the model:

        $\boldsymbol{\delta}_i = \boldsymbol{r}_i + \gamma\widehat{V}(\boldsymbol{\eta}_{i+1}, \boldsymbol{w}_i) - \widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i)$    (24)
        $\boldsymbol{w}_{i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_i + \alpha\boldsymbol{\delta}_i\nabla_w\widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}_i)$    (25)
**end while**
```

B. DECENTRALIZED STRATEGY

The centralized strategy is disadvantageous in the sense that (i) failure of the fusion center results in failure of the system; (ii) there can be communication bottlenecks at the fusion center; (iii) and agents can be spatially distributed to begin with. Therefore, in this section, we propose a fully decentralized strategy for policy evaluation where agents communicate with their immediate neighbors only.

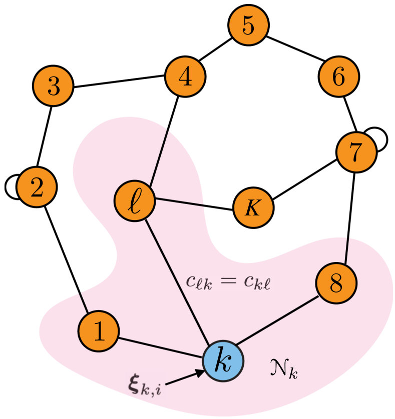


FIGURE 1. An illustration of a network model.

1) DECENTRALIZED NETWORK MODEL

We refer to Fig. 1 and assume that the graph is strongly connected [53], which means that paths exist connecting any pair of agents (ℓ, k) in both directions, and in addition, there exists at least one agent in the graph that does not discard its own information (i.e., $c_{kk} > 0$ for at least one agent k). Under this assumption, the combination matrix $C = [c_{\ell k}]$, where entry $c_{\ell k} \geq 0$ scales the information agent k receives from agent ℓ, becomes primitive. If two agents are not connected by an edge then $c_{\ell k} = 0$. We assume C is symmetric and doubly-stochastic, meaning that

$$\sum_{\ell=1}^{K} c_{\ell k} = 1, \quad c_{\ell k} = c_{k\ell}, \tag{26}$$

or in matrix notation:

$$C\mathbb{1}_K = \mathbb{1}_K, \quad C = C^{\mathsf{T}}. \tag{27}$$

2) LOCAL BELIEF FORMATION

In the fully decentralized strategy, the agents cannot form the joint belief (16) since they do not have access to the observations and actions of all other agents. They, however, can construct local beliefs. To do so, we will extend the diffusion HMM strategy (DHS) from [48] and [49], which is originally designed for hidden Markov models, to the current POMDP setting.

In DHS, the global belief vectors $\{\boldsymbol{\mu}_i, \boldsymbol{\eta}_i\}$ are replaced by local belief vectors $\{\boldsymbol{\mu}_{k,i}, \boldsymbol{\eta}_{k,i}\}$, and the latter are updated by using local observations and by relying solely on interactions with the immediate neighbors. The original DHS algorithm is designed for actionless partially observable Markov chains, and each agent can use the same global transition model. However, in POMDPs, transition of the global state depends on the joint action, and the agents cannot perform a *centralized* time-adjustment step as in (23) since they do not know the actions of all agents in the network.

Therefore, one strategy is to use a transition model that is obtained by marginalizing over actions that are unknown to agent k. More specifically, let $a_{\mathcal{N}_k} \in \mathcal{A}_{\mathcal{N}_k}$ denote a tuple of actions taken by the set of neighbors of agent k (which we are denoting by $\mathcal{N}_k$). These actions can be assumed to be known by agent k if, for instance, agents share their actions with their neighbors. Let $a^c_{\mathcal{N}_k} \in \mathcal{A}^c_{\mathcal{N}_k}$ denote the remaining actions by all other agents in the network, so that $a = a_{\mathcal{N}_k} \cup a^c_{\mathcal{N}_k}$. Then, each agent can use the following *local* transition model approximation:

$$\mathbb{T}_k^{\pi}(s|s', a_{\mathcal{N}_k}) \propto \sum_{a^c_{\mathcal{N}_k} \in \mathcal{A}^c_{\mathcal{N}_k}} \mathbb{T}(s|s', a_{\mathcal{N}_k}, a^c_{\mathcal{N}_k})\pi(a_{\mathcal{N}_k}, a^c_{\mathcal{N}_k}|s') \tag{28}$$

in lieu of $\mathbb{T}(s|s', a)$, to time-adjust its local belief:

$$\boldsymbol{\eta}_{k,i}(s) = \sum_{s' \in \mathcal{S}} \mathbb{T}_k^{\pi}(s|s', \boldsymbol{a}_{\mathcal{N}_k,i-1})\boldsymbol{\mu}_{k,i-1}(s'), \tag{29}$$

Here, $\boldsymbol{a}_{\mathcal{N}_k,i-1}$ is the tuple of actions taken by the neighbors of agent k at time instant $i-1$. Moreover, in (28), the notation $\pi(a_{\mathcal{N}_k}, a^c_{\mathcal{N}_k}|s')$ represents the joint action probability:

$$\pi\left(a_{\mathcal{N}_k}, a^c_{\mathcal{N}_k}|s'\right) = \prod_{\ell=1}^{K} \pi_\ell(a_\ell|s'), \tag{30}$$

where the notation $\pi(a|s)$ is now a shorthand for $\pi(a|\mu)$ when

$$\mu = [0 \ldots 1 \ldots 0]^{\mathsf{T}}, \tag{31}$$

i.e., when the belief attains value 1 for state s and is 0 otherwise. Note that this construction leads to a richer scenario compared to [48], [49], with transition models that are different across the agents. The rest of the algorithm is the same as the DHS strategy. Following (29), and based on the personal observation $\boldsymbol{\xi}_{k,i}$, each agent k forms an *intermediate* belief using a β-scaled Bayesian update of the form:

$$\boldsymbol{\psi}_{k,i}(s) \propto (L_k(\boldsymbol{\xi}_{k,i}|s))^{\beta}\boldsymbol{\eta}_{k,i}(s), \tag{32}$$

where $\beta > 0$. Finally, agents in the neighborhood of k share their intermediate beliefs, which allows agent k to update its belief using the weighted geometric average expression:

$$\boldsymbol{\mu}_{k,i}(s) \propto \prod_{\ell \in \mathcal{N}_k} \left(\boldsymbol{\psi}_{\ell,i}(s)\right)^{c_{\ell k}}. \tag{33}$$

This procedure of repeated updating and exchanging of beliefs allows information to diffuse over the network.

3) DIFFUSION POLICY EVALUATION

In the fully decentralized strategy, the local belief formation strategy is used during both training and execution phases. Namely, the target value function in (19) represents the average return agents get when they execute the policy π with their local beliefs formed via the DHS strategy. Moreover, since the policy evaluation is also decentralized, during the training phase, they again need to use DHS to approximate the global belief state μ on top of the function approximation. More specifically, using its local belief vectors, each agent k computes a local TD error:

$$\boldsymbol{\delta}_{k,i} = \boldsymbol{r}_{k,i} + \gamma\widehat{V}(\boldsymbol{\eta}_{k,i+1}, \boldsymbol{w}_{k,i}) - \widehat{V}(\boldsymbol{\mu}_{k,i}, \boldsymbol{w}_{k,i}), \tag{34}$$

where $\boldsymbol{r}_{k,i} = \boldsymbol{r}_k(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1})$ is also a function of the local beliefs since each agent k now executes the action $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_{k,i})$. Subsequently, each agent k forms an intermediate parameter estimate denoted by

$$\boldsymbol{z}_{k,i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_{k,i} + \alpha\boldsymbol{\delta}_{k,i}\nabla_w \widehat{V}(\boldsymbol{\mu}_{k,i}, \boldsymbol{w}_{k,i}). \tag{35}$$

After receiving the intermediate estimates from its neighbors, agent k updates $\boldsymbol{w}_{k,i}$ to

$$\boldsymbol{w}_{k,i+1} = \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{z}_{\ell,i+1}. \tag{36}$$

The local adaptation step (35) followed by the combination step (36) are reminiscent of diffusion strategies for distributed learning [19], [53]. Observe that there are actually two combination steps involved in diffusion policy evaluation: the belief combination (33) with geometric averaging (GA), and the parameter combination (36) with arithmetic averaging (AA). These choices of fusion rules are supported by recent results in the literature [68], [69] that promote the use of GA for probability density functions and AA for point estimates. The listing of the proposed diffusion policy evaluation strategy for POMDPs appears in Algorithm 2.

Algorithm 2 has the following listed advantages:

- *Decentralized information structure:* The algorithm is designed to be fully decentralized, with each agent only having access to its own private data, such as observations and rewards, without the need to share this information with other agents. Importantly, agents do not require knowledge of the joint distribution of observations or the network topology. They only know their own marginal likelihood function, and their actions are only known by (or transmitted to) their immediate neighbors. If agents happen to know their own marginal transition models, they do not need to know the policies of other agents or the global transition model. However, if the application requires them to approximate it themselves, they require knowledge of the other policies and the global transition model.
- *Privacy:* The algorithm is also advantageous in terms of privacy since (i) communicating beliefs allows information diffusion without explicitly sharing raw observational data, and (ii) exchanging value parameters allows agents learn the cumulative reward across network without explicitly sharing local rewards.
- *Complexity:* (i) The memory requirement is constant over time, with each agent only needing to store its value function parameter estimate (M-dimensional) and local belief (S-dimensional), as well as the necessary model functions. (ii) The communication requirement is also manageable, with each agent communicating only with its immediate neighbors through belief and parameter sharing. The communication load is not affected by the network size, making our algorithm scalable and avoiding communication bottlenecks. (iii) The computational complexity depends on whether the application at hand allows agents to have access to the local transition model. If this is the case, then the computational complexity is equivalent to the single-agent Bayesian filtering case, which is $O(S^2)$. The combination steps add only linear additional complexity $O(S)$ with fixed neighborhood size. However, if agents need to approximate the transition model themselves, the computational complexity increases with the network size, and becomes $O(KS^2)$. This is due to the need to average over non-neighbors' actions in (28), whose size grows with the network size in general. Compared to alternative approaches such as relaying raw data, incremental approaches [70], or Bayesian belief forming [71], our algorithm is much lighter in terms of complexity. Relaying raw data, for example, would result in an exponential increase of memory and communication overload at each hop, making it highly impractical. The

Algorithm 2: Diffusion Policy Evaluation Under POMDPs.

```
set initial priors $\eta_{k,0}(s) > 0, \forall s \in \mathcal{S}$ and $\forall k \in \mathcal{K}$
choose $\beta > 0$
initialize $w_{k,0}$ for $\forall k \in \mathcal{K}$
while $i \geq 0$ do
    each agent $k$ observes $\boldsymbol{\xi}_{k,i}$
    for each agent $k \in \mathcal{K}$ and $s \in \mathcal{S}$
        $\boldsymbol{\psi}_{k,i}(s) \propto (L_k(\boldsymbol{\xi}_{k,i}|s))^{\beta} \boldsymbol{\eta}_{k,i}(s)$    (37)
        $\boldsymbol{\mu}_{k,i}(s) \propto \prod_{\ell \in \mathcal{N}_k} (\boldsymbol{\psi}_{\ell,i}(s))^{c_{\ell k}}$    (38)
    end for
    for each agent $k \in \mathcal{K}$ do
        Take action $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_{k,i})$
        Get reward $\boldsymbol{r}_{k,i} = \boldsymbol{r}_k(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1})$
    end for
    for each agent $k \in \mathcal{K}$ evolve
        Compute $\mathbb{T}_k^{\pi}(s|s', \boldsymbol{a}_{\mathcal{N}_k,i})$ using (28), and
        $\boldsymbol{\eta}_{k,i+1}(s) = \sum_{s' \in \mathcal{S}} \mathbb{T}_k^{\pi}(s|s', \boldsymbol{a}_{\mathcal{N}_k,i}) \boldsymbol{\mu}_{k,i}(s')$    (39)
    end for
    for each agent $k \in \mathcal{K}$ update
        $\boldsymbol{\delta}_{k,i} = \boldsymbol{r}_{k,i} + \gamma \widehat{V}(\boldsymbol{\eta}_{k,i+1}, \boldsymbol{w}_{k,i}) - \widehat{V}(\boldsymbol{\mu}_{k,i}, \boldsymbol{w}_{k,i})$    (40)
        $\boldsymbol{z}_{k,i+1} = (1 - 2\rho\alpha)\boldsymbol{w}_{k,i} + \alpha\boldsymbol{\delta}_{k,i}\nabla_w \widehat{V}(\boldsymbol{\mu}_{k,i}, \boldsymbol{w}_{k,i})$    (41)
    end for
    for each agent $k \in \mathcal{K}$ combine
        $\boldsymbol{w}_{k,i+1} = \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{z}_{\ell,i+1}$    (42)
    end for
end while
```

incremental approach of relaying over a cyclic path (which is NP-hard to find [72]) that visits each agent once would reduce the overload. However, it is not robust against failures and not scalable, making it impractical for a decentralized setting. The Bayesian belief forming strategy requires knowledge of the network topology and other agents' functions, and known to be NP-hard, even in the much simpler case of fixed state and no action setting [13].

V. MAIN RESULTS

In this section, we analyze the performance of the decentralized strategy in Algorithm 2. In particular, we first show in Section V-B that the value function parameters $\{\boldsymbol{w}_{k,i}\}$ of the agents cluster around the network centroid. Then, in Section V-C, we show that this network centroid has a bounded difference from the parameter of a baseline strategy (which will be presented in Algorithm 3). Our analysis relies on bounding the disagreement between the joint centralized belief $\boldsymbol{\mu}_i$ and the local estimate $\boldsymbol{\mu}_{k,i}$, which is presented next.

A. BELIEF DISAGREEMENT

In a manner similar to [49], we introduce the following risk functions in order to assess the disagreement between the local beliefs formed via (37)–(39) with the joint centralized beliefs formed via (22)–(23):

$$J_{k,i} \triangleq \mathbb{E}_{\mathcal{F}_i} D_{\text{KL}}(\boldsymbol{\mu}_i || \boldsymbol{\mu}_{k,i}), \tag{43}$$

and

$$\widetilde{J}_{k,i} \triangleq \mathbb{E}_{\mathcal{F}^a_{i-1}} D_{\text{KL}}(\boldsymbol{\eta}_i || \boldsymbol{\eta}_{k,i}). \tag{44}$$

The risks in (43) and (44) measure the disagreement after and before the joint observation $\boldsymbol{\xi}_i$, respectively. Remember that [49] considers a naive state estimation setting rather than a POMDP. Specifically, in their setting, the transition model does not depend on actions, and it is assumed that every agent knows the global transition model accurately. In comparison, in the current work, each agent uses a local approximation for the global transition model based on (28). Therefore, we need to make some non-trivial adjustments to the belief disagreement analysis. We begin with adjusting the assumptions from [49] to our model.

1) MODELING CONDITIONS

- *Likelihood functions:* Each observation has bounded information about the true state. More formally,

$$D_{\text{KL}}(L_k(\xi|s) || L_k(\xi|s')) < \infty \tag{45}$$

which ensures that likelihoods for each state pair (s, s') share the same support, and in addition to this,

$$|\log L_k(\xi|s)| \leq B \tag{46}$$

over its support for each state $s \in \mathcal{S}$ and agent $k \in \mathcal{K}$.

- *Transition model:* The Markov chain induced by any joint action $a \in \mathcal{A}$ is *irreducible* and *aperiodic*. Since the number of states is finite, this assumption implies that the transition model $\mathbb{T}(s|s', a)$ is ergodic [73, Chap. 2]. Like [49], we focus on the important class of *geometrically* ergodic models, which additionally satisfy the relation $\kappa(\mathbb{T}^a) \leq \kappa(\mathbb{T})$ for some constant $\kappa(\mathbb{T}) < 1$. Here, $\kappa(\mathbb{T}^a)$ is the Dobrushin coefficient [12, Chap. 2] defined by:

$$\kappa(\mathbb{T}^a) \triangleq \sup_{s', s'' \in \mathcal{S}} \frac{1}{2} \sum_{s \in \mathcal{S}} \left| T^a_{ss'} - T^a_{ss''} \right|, \tag{47}$$

where $T^a_{ss'} \triangleq \mathbb{T}(s|s', a)$ is a generic entry of the $S \times S$ transition matrix T^a. Due to space limitations, we refer the reader to [12, Chap. 2] for a comprehensive discussion on the Dobrushin coefficient $\kappa(\mathbb{T}^a)$. In short, $\kappa(\mathbb{T}^a)$ quantifies how fast the transition model forgets its initial conditions. Namely, as $\kappa(\mathbb{T}^a) \to 0$, past conditions are forgotten faster. Instances of geometrically ergodic transition models include transition matrices with all positive elements, or that satisfy the minorization condition in [12, Theorem 2.7.4]. In addition to this condition from [48], [49], we have an additional assumption on the transition model to regulate the disagreement stemming from the local transition model estimates:

Assumption 1 (Transition model disagreement): For each agent k, consider the n-hop neighbors set $\mathcal{N}_{k^n}$ and its complement $\mathcal{N}^c_{k^n}$. In other words, $\mathcal{N}_{k^n}$ is the set of agents that have at most n-hop distance to the agent k. We define the transition model approximation that uses n-hop neighbors' actions as follows:

$$\mathbb{T}^\pi_k(s|s', a_{\mathcal{N}_{k^n}}) \propto \sum_{a^c_{\mathcal{N}_{k^n}} \in \mathcal{A}^c_{\mathcal{N}_{k^n}}} \mathbb{T}\left(s|s', a_{\mathcal{N}_{k^n}}, a^c_{\mathcal{N}_{k^n}}\right) \pi\left(a_{\mathcal{N}_{k^n}}, a^c_{\mathcal{N}_{k^n}} | s'\right). \tag{48}$$

Then, we assume that

$$D_{\text{KL}}\left(\mathbb{T}^\pi_k\left(s \middle| s', a_{\mathcal{N}_{k^n}}\right) \middle|\middle| \mathbb{T}^\pi_k\left(s \middle| s', a_{\mathcal{N}_{k^{n+1}}}\right)\right) < \infty, \tag{49}$$

which ensures that transition model approximations induced from n-hop and $(n+1)$-hop neighbors' actions share the same support. Moreover, we assume that over the shared support,

$$\left| \log \frac{\mathbb{T}^\pi_k\left(s|s', a_{\mathcal{N}_{k^n}}\right)}{\mathbb{T}^\pi_k\left(s|s', a_{\mathcal{N}_{k^{n+1}}}\right)} \right| \leq \tau. \tag{50}$$

for $n \geq 1$.

This assumption basically makes sure that the increase in the error of the transition model approximation of agents due to lack of information about actions is bounded at each geodesic distance increase to that agent.

2) DIFFERENCE WITH CENTRALIZED STRATEGY

The following result provides upper bounds on the disagreement measures in (43)–(44).

Theorem 1 (Bounds on belief disagreement): For each agent k, the belief disagreement risks (43) and (44) get bounded with a linear rate of $\kappa(\mathbb{T})$. Namely, as $i \to \infty$,

$$J_{k,i} \le \frac{2\sqrt{K}\beta\lambda B}{1-\kappa(\mathbb{T})} + \frac{(K-d_{\min})\,\tau}{1-\kappa(\mathbb{T})} \tag{51}$$

and

$$\widetilde{J}_{k,i} \le \frac{2\kappa(\mathbb{T})\sqrt{K}\beta\lambda B}{1-\kappa(\mathbb{T})} + \frac{(K-d_{\min})\,\tau}{1-\kappa(\mathbb{T})} \tag{52}$$

where $d_{\min}$ is the minimum degree over the graph, i.e., minimum number of neighbors any agent over the network possesses, and $\lambda \triangleq \max\{|1-\frac{K}{\beta}|, \lambda_2\}$ where $\lambda_2 < 1$ is the mixing rate (second largest modulus eigenvalue) of C.

Proof: See Appendix A. ■

In Theorem 1, the first terms in both bounds are equivalent to the bounds obtained in [49]. However, the terms proportional to $(K-d_{\min})\tau$ are new, and they arise from the fact that agents do not observe the joint actions and hence only have a local estimate of the transition model. Nevertheless, the bounds get smaller with increasing network connectivity, i.e., as $\lambda_2 \to 0$ and $d_{\min} \to K$, which shows the benefit of cooperation. In particular, if $\beta = K$ and the network is fully connected ($\lambda_2 = 0, d_{\min} = K$), then the bounds are equal to 0. In other words, local beliefs match the centralized belief in this situation. It is important to note that the linear term $(K-d_{\min})$ represents a worst-case bound that holds true for any strongly connected network topology. For instance, in a scenario where each agent has $N > 1$ neighbors, it is straightforward to modify the proof and show that these linear terms will instead be logarithmic, i.e., proportional to $\log K/\log N$.

We use Theorem 1 in the performance analysis of the diffusion policy evaluation. To that regard, we first present the following consequence of Theorem 1, which provides a bound in terms of disagreement norms.

Corollary 1 (Bounds on disagreement norms): Theorem 1 implies that, as $i \to \infty$,

$$\mathbb{E}\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_{k,i}\right\| \le B_{\rm TV} \tag{53}$$

and

$$\mathbb{E}\left\|\boldsymbol{\eta}_i - \boldsymbol{\eta}_{k,i}\right\| \le \widetilde{B}_{\rm TV}, \tag{54}$$

where we introduce the constants

$$B_{\rm TV} \triangleq 2\left(1-\exp\left\{-\frac{2\sqrt{K}\beta\lambda B + (K-d_{\min})\,\tau}{1-\kappa(\mathbb{T})}\right\}\right)^{1/2} \tag{55}$$

and

$$\widetilde{B}_{\rm TV} \triangleq 2\left(1-\exp\left\{-\frac{2\kappa(\mathbb{T})\sqrt{K}\beta\lambda B + (K-d_{\min})\tau}{1-\kappa(\mathbb{T})}\right\}\right)^{1/2} \tag{56}$$

Proof: See Appendix B. ■

B. NETWORK DISAGREEMENT

In this section, we study the variation of agent parameters from the network centroid. To that end, let us incorporate the linear approximation $\widehat{V}(\mu, w) = \phi(\mu)^{\mathsf{T}} w$ into the TD-error expression (40) to obtain the following relation:

$$\boldsymbol{\delta}_{k,i} = \boldsymbol{r}_{k,i} + \gamma\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\boldsymbol{w}_{k,i} - \phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}\boldsymbol{w}_{k,i}. \tag{57}$$

Since $\nabla_w \widehat{V}(\mu, w) = \phi(\mu)$ for the linear case, it follows that

$$\boldsymbol{z}_{k,i+1} = \left((1-2\rho\alpha)I - \alpha\boldsymbol{H}_{k,i}\right)\boldsymbol{w}_{k,i} + \alpha\boldsymbol{d}_{k,i}, \tag{58}$$

where

$$\boldsymbol{H}_{k,i} \triangleq \phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}} - \gamma\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}, \tag{59}$$

and

$$\boldsymbol{d}_{k,i} \triangleq \boldsymbol{r}_{k,i}\phi(\boldsymbol{\mu}_{k,i}). \tag{60}$$

To proceed, we introduce the following regularity assumption on the feature vector.

Assumption 2 (Feature vector): The feature mapping $\phi(\mu)$ is bounded and Lipschitz continuous in the domain of the S-dimensional probability simplex. Namely, for any vectors $\mu_1, \mu_2 \in \mathcal{M}(S)$,

$$\|\phi(\mu_1) - \phi(\mu_2)\| \le L_\phi\|\mu_1 - \mu_2\|, \quad \|\phi(\mu_1)\| \le B_\phi. \tag{61}$$

■

Lemma 1 (Belief feature difference): For each agent $k \in \mathcal{K}$, the belief feature matrix $\boldsymbol{H}_{k,i}$ in (59) has bounded expected difference in relation to the centralized belief feature matrix $\boldsymbol{H}_i^\star$, defined below, i.e.,

$$\mathbb{E}\|\boldsymbol{H}_{k,i} - \boldsymbol{H}_i^\star\| \le 2B_\phi L_\phi B_{\rm TV}(1+\gamma), \tag{62}$$

where

$$\boldsymbol{H}_i^\star \triangleq \phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\mu}_i)^{\mathsf{T}} - \gamma\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}. \tag{63}$$

Proof: See Appendix C. ■

We also assume that all rewards are non-negative and uniformly bounded, i.e., $0 \le \boldsymbol{r}_{k,i} \le R_{\max}$ for each agent $k \in \mathcal{K}$, and all time instants i. Now, we proceed to study the network disagreement. To that end, we define the network centroid as

$$\boldsymbol{w}_{c,i} \triangleq \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{w}_{k,i}, \tag{64}$$

which is an average of the parameters of all agents. The following result shows that the agents cluster around this network centroid after sufficient iterations.

Theorem 2 [Network agreement]: The average distance to the network centroid is bounded for $\rho > \gamma B_\phi L_\phi/\sqrt{2}$ after sufficient number of iterations. In particular, if $\rho \ge 0.75\gamma B_\phi L_\phi$, then

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}_{k,i} - \boldsymbol{w}_{c,i}\| \le \frac{\alpha\lambda_2\epsilon}{(1-\lambda_2)} + O(\alpha^2) \tag{65}$$

where $\epsilon > 0$ is a constant defined by

$$\epsilon \triangleq R_{\max}B_\phi\left(\frac{2B_{\rm TV}(1+\gamma)}{0.08\gamma} + 1\right). \tag{66}$$

Proof: See Appendix D. ■

Theorem 2 states that the parameter estimates by the agents cluster around the network centroid within mean ℓ_2-distance on the order of $O(\alpha\lambda_2)$ in the limit as $i \to \infty$. This result confirms that agents can get arbitrarily close to each other by setting the learning rate α sufficiently small. Besides, dense networks have in general small λ_2, which results in a small disagreement within the network.

C. PERFORMANCE OF DIFFUSION POLICY EVALUATION

We can therefore use the network centroid as a proxy for all agents to show that the disagreement between the fully decentralized strategy of Algorithm 2 and a baseline strategy that requires a central processor during training is bounded. We start by describing this baseline strategy and explain why it is a more suitable baseline compared to using the fully centralized strategy Algorithm 1.

In some applications, even though agents are supposed to work in a decentralized fashion once implemented in the field, they can nevertheless rely on central processing during the training phase in order to learn the best policy. In the literature, this paradigm is referred to as *centralized training for decentralized execution* [16], [74]. For our problem, the crucial point is that during training the centralized processor can form beliefs based on all observations, but it should keep in mind that agents will execute their actions based on *local* beliefs once implemented. Therefore, in the baseline strategy, actions and rewards are based on local beliefs as in (37)–(39), whereas parameter updates are based on the centralized posterior as in (22)–(23). Algorithm 3 lists this baseline procedure. Notice that the algorithm consists of both local belief construction (see (67), (68), and (70)) and centralized belief construction (see (69) and (71)). The former is used for action execution $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_{k,i})$, while the latter is used for value function parameter updates in (72)–(73).

In the fully centralized strategy of Algorithm 1, the actions by the agents and the subsequent rewards are based on the centralized belief. Therefore, the target value function that Algorithm 1 aims to learn corresponds to the average cumulative reward obtained under centralized execution. In comparison, the target value functions that Algorithms 2 and 3 try to learn are the same and they correspond to the average cumulative reward under decentralized execution. While trying to learn the same parameter w°, the baseline strategy can utilize centralized processing, but the diffusion strategy is fully decentralized. Nonetheless, the following result illustrates that the expected disagreement between the baseline strategy and the fully decentralized strategy remains bounded.

Algorithm 3: Centralized Evaluation for Decentralized Execution.

```
set initial priors $\eta_{k,0}(s) > 0$, $\eta_0(s) > 0$, for $\forall s \in \mathcal{S}$ and $\forall k \in \mathcal{K}$
choose $\beta > 0$
initialize $w^\star_0$
while $i \geq 0$ do
    each agent $k$ observes $\boldsymbol{\xi}_{k,i}$
    for each agent $k \in \mathcal{K}$ and $s \in \mathcal{S}$ adapt and combine
        $\boldsymbol{\psi}_{k,i}(s) \propto (L_k(\boldsymbol{\xi}_{k,i}|s))^{\beta}\boldsymbol{\eta}_{k,i}(s)$   (67)
        $\boldsymbol{\mu}_{k,i}(s) \propto \prod_{\ell\in\mathcal{N}_k} \left(\boldsymbol{\psi}_{\ell,i}(s)\right)^{a_{\ell k}}$   (68)
    end for
    to form centralized belief with joint observation $\boldsymbol{\xi}_i \triangleq \{\boldsymbol{\xi}_{k,i}\}_{k=1}^{K}$, adapt
        $\boldsymbol{\mu}_i(s) \propto L(\boldsymbol{\xi}_i|s)\boldsymbol{\eta}_i(s)$   (69)
    for each agent $k \in \mathcal{K}$ do
        Take action $\boldsymbol{a}_{k,i} \sim \pi_k(a_k|\boldsymbol{\mu}_{k,i})$
        Get reward $\boldsymbol{r}_{k,i} = \boldsymbol{r}_k(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{s}_{i+1})$
    end for
    average the rewards $\boldsymbol{r}^\star_i = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{r}_{k,i}$
    for each agent $k \in \mathcal{K}$ evolve
        Compute $\mathbb{T}^{\pi}_{k}(s|s', \boldsymbol{a}_{\mathcal{N}_k,i})$ using (28), and
        $\boldsymbol{\eta}_{k,i+1}(s) = \sum_{s'\in\mathcal{S}} \mathbb{T}^{\pi}_{k}(s|s', \boldsymbol{a}_{\mathcal{N}_k,i})\boldsymbol{\mu}_{k,i}(s')$   (70)
    end for
    evolve the centralized belief
        $\boldsymbol{\eta}_{i+1}(s) = \sum_{s'\in\mathcal{S}} \mathbb{T}(s|s', \boldsymbol{a}_i)\boldsymbol{\mu}_i(s')$   (71)
    update value function parameter
        $\boldsymbol{\delta}^\star_i = \boldsymbol{r}^\star_i + \gamma\widehat{V}(\boldsymbol{\eta}_{i+1}, \boldsymbol{w}^\star_i) - \widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}^\star_i)$   (72)
        $\boldsymbol{w}^\star_{i+1} = (1-2\rho\alpha)\boldsymbol{w}^\star_i + \alpha\boldsymbol{\delta}^\star_i\nabla_w\widehat{V}(\boldsymbol{\mu}_i, \boldsymbol{w}^\star_i)$   (73)
end while
```

Theorem 3 (Disagreement with the baseline solution): The expected distance between the baseline strategy and the network centroid is bounded after sufficient iterations for $\rho > \gamma B_\phi L_\phi/\sqrt{2}$. In particular, if $\rho \geq 0.75\gamma B_\phi L_\phi$, then

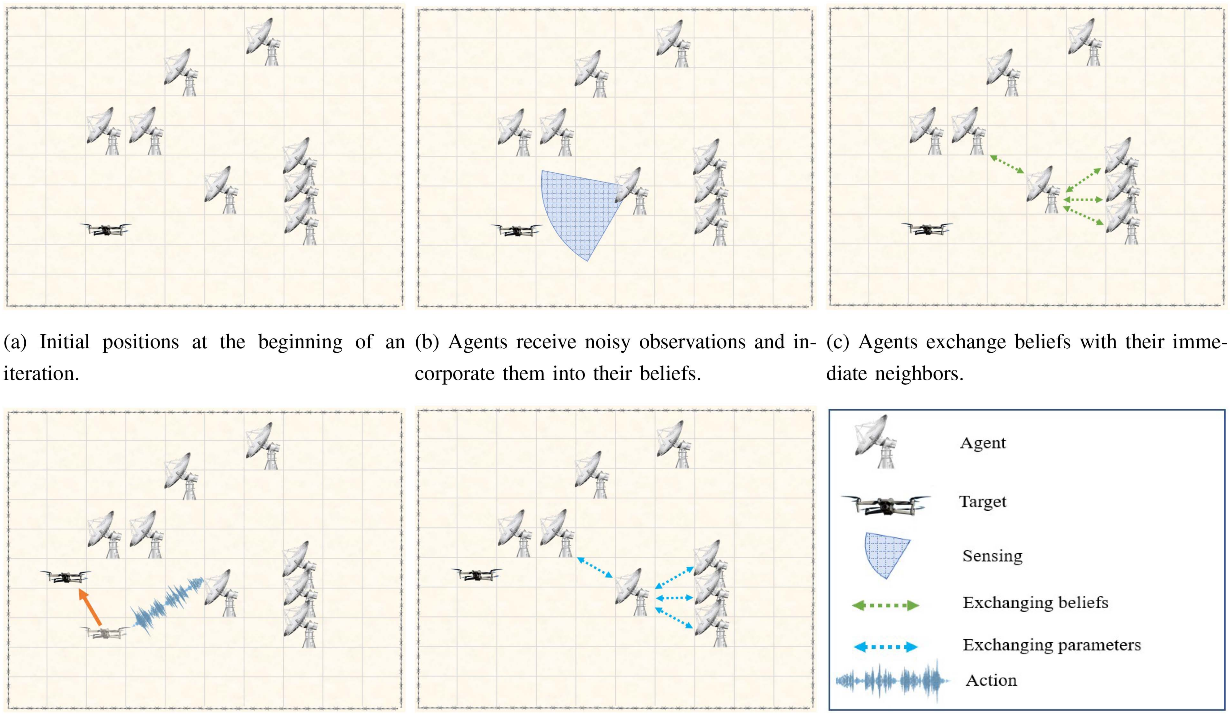
$$\mathbb{E}\|\boldsymbol{w}^\star_i - \boldsymbol{w}_{c,i}\| \leq \frac{B_{\text{TV}}R_{\max}\epsilon'}{0.08\gamma B_\phi L_\phi} \tag{74}$$

after $i \geq i_0 = o(1/(\alpha\gamma B_\phi L_\phi))$ iterations, where $\epsilon' > 0$ is a constant defined by

$$\epsilon' \triangleq \frac{2B_\phi(1+\gamma)}{0.08\gamma} + L_\phi. \tag{75}$$

Proof: See Appendix E. ■

Theorem 3 implies that the disagreement between the network centroid, around which agents cluster, and the baseline strategy is on the order of B_{TV}. This means that if the local beliefs are similar to the centralized belief, agents get closer to the baseline parameter. In this regard, from the definition (55) of B_{TV}, it can be observed that B_{TV} gets smaller with increasing network connectivity (i.e., decreasing λ_2), as $\beta \to K$. In fact, it is equal to zero for fully-connected networks with the



(a) Initial positions at the beginning of an iteration.

(b) Agents receive noisy observations and incorporate them into their beliefs.

(c) Agents exchange beliefs with their immediate neighbors.

(d) Agents take actions based on the beliefs. The target relocates based on the actions.

(e) Agents update and exchange value function parameters.

(f) The legend. Image credit for agent, target, and action: freepik.com

FIGURE 2. Experimental scenario. For visual purposes, the procedure is shown for only one agent. In fact, all agents execute the same procedure simultaneously.

choice of $\beta = K$ and $c_{\ell k} = 1/K$. Therefore, by changing β and $c_{\ell k}$, the fully decentralized strategy can match the value function estimates of a centralized training strategy that can gather all observations and actions in a fusion center. In the next section, by means of numerical simulations, we further compare the value function estimate accuracies of all Algorithms 1, 2 and 3 by using squared Bellman error (SBE).

VI. SIMULATION RESULTS

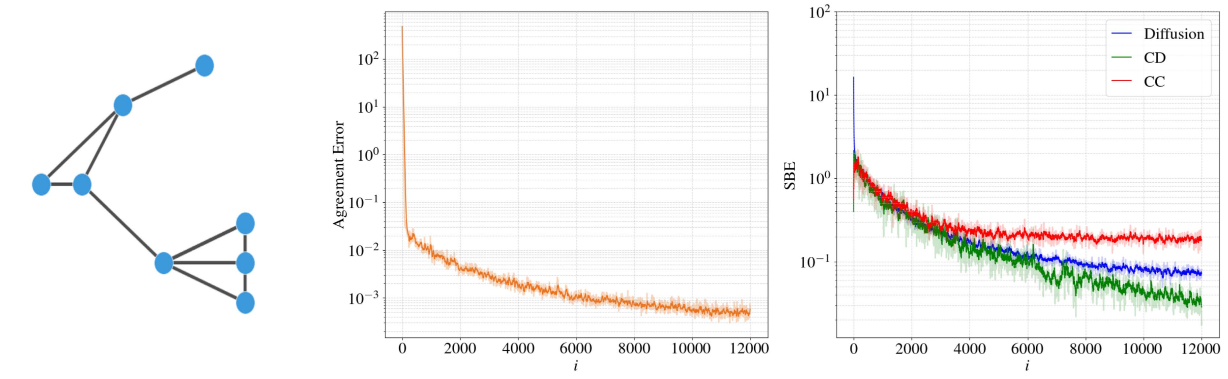
For numerical simulations, we consider a multi-agent target localization application. The implementation is available online[1]. We use a set of $K = 8$ agents and a moving target in a 10×10 two-dimensional grid world environment. The locations of the agents are fixed and their coordinates are randomly assigned at the beginning of the simulation. The goal of the agents is to cooperatively evaluate a given policy for hitting the target. Agents cannot observe the location (i.e., state) of the target accurately, but instead receive noisy observations based on how far they are from the real location of the target. The target is moving according to some pre-defined transition model that takes the actions (i.e., hits) of agents into account. Specifically, the target is trying to evade the hits of agents.

A possible scenario for this setting is a network of sensors and an intruder (e.g., a spy drone) — see Fig. 2. The sensors try to localize the intruder based on noisy measurements and belief exchanges. Moreover, in order to disrupt the communication between the intruder and its owner, each sensor sends a narrow sector jamming beam towards its target location estimate. However, the intruder is capable of detecting energy abnormalities and determines its next location by favoring distant locations from the jamming signals. We now describe the setting in more detail.

Combination matrix: The entries of the combination matrix are set such that they are inversely proportional to the ℓ_1-distance between the agents. That is to say, the further the agents are from each other, the smaller the value of the weight that is assigned to the edge connecting them. Weights smaller than some threshold are set to 0, which implies that agents that are too far from each other do not need to communicate. The resulting communication topology graph is illustrated in Fig. 3(a).

Transition model: The target is moving between cells in a grid (i.e., states) randomly. The probability of a cell being the next location of the target depends on the current location of the target and the location of the agents' hits. Namely, each state in the grid is assigned a score based on its ℓ_1-distance to the current location of the target and to the average location of the agents' hits — see Table 1. For example, observe from Table 1 that the cells that are in the proximity of the target's

[1] [Online]. Available: https://github.com/asl-epfl/DecPOMDP_Policy_Evaluation_w-Belief_Sharing



(a) Communication topology.

(b) Agreement error over time.

(c) SBE over time (running window of size 20) for CC (Alg. 1), Diffusion (Alg. 2), CD (Alg. 3).

FIGURE 3. (a) Graph structure underlying the communication of agents. (b) Evolution of agreement error (defined in Eq. (76)) over time for fully decentralized strategy. (c) Evolution of squared Bellman error (SBE) (defined in Eq. (77)) over time for Algorithms 1–3.

TABLE 1. The table of scores used in the transition model. Each candidate state for next state (location) of the target gets a score based on the initial position of the target and the average action of agents.

		initial position	
	ℓ_1-distance	≤ 4	> 4
average location of agents' hits	< 4	10	5
	≥ 4	100	50

TABLE 2. The table of scores used in the likelihood function model. Each state, when observed, gets a score that determines the likelihood of the presence of the target within the state, based on the position of the target and the average action of agents.

		real location of target					
	ℓ_1-distance	$= 0$	< 3	< 5	< 7	< 9	≥ 9
location of agent	< 3	400	200	30	1	1	1
	≤ 6	200	180	100	1	1	1
	> 6	25	25	25	25	4	1

current location and also far away from the agents' strikes are given the highest score. These scores are normalized to yield a probabilistic transition kernel.

Likelihood function: Agents cannot observe where the target is. They can only receive noisy observations. Each agent gets a more accurate observation of the target's position if the target is in close proximity to the agent. Otherwise, the larger the distance between the agent and the target, the higher the noise level. Depending on how close the target is to the agent, and in order to construct the likelihood function, we first assign scores to each cell in the grid that reflect how probable it is to find the target in that cell — see Table 2. Following that, the scores are normalized in order to yield a distribution function. For instance, if the target lies at an ℓ_1-distance that is less than 3 grid squares from the location of the agent, the actual position of the target gets a likelihood score of 400, cells within an ℓ_1 distance of 2 grid squares from the agent get a likelihood score of 200, and cells within an ℓ_1 distance of 4 grid squares from the agents get a likelihood score of 30.

Reward function: The reward function in the environment is such that an agent receives a reward of 1 if the agent is able to hit the position of the target. The agent also receives a reward of 0.2 if the ℓ_1-distance between the predicted location and the actual location of the target is less than 3 grid units. Otherwise, it gets 0 reward. Agents do not know the reward model, and use the instantaneous rewards instead.

Policy: We fix the policy that the agents evaluate as the maximum a-posteriori policy. Namely, agents detect (hit) a location if it corresponds to the maximum entry in their belief vector.

We use the belief vectors as the features directly, i.e., ϕ is an identity transformation. We set $\alpha = 0.1$, $\rho = 0.0001$, and $\beta = K = 8$, and average over 3 different realizations for all cases. In Fig. 3(b), the average mean-square distance to the network centroid, i.e.,

$$\text{Agreement error} \triangleq \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}_{k,i} - \boldsymbol{w}_{c,i}\|^2, \tag{76}$$

is plotted over time for the fully decentralized strategy. Confirming Theorem 2, it can be seen that agreement error rapidly decreases and converges to a small value.

In Fig. 3(c), we plot the evolution of the average squared Bellman error (SBE) in the log domain, where the SBE expression is given by:

$$\text{SBE} \triangleq \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\delta}_{k,i}^2, \tag{77}$$

and similarly for the centralized cases. It measures the network average of instantaneous TD-errors. It can be seen that all approaches converge, and in particular, diffusion strategy (Algorithm 2) yields a comparable performance with CD (Algorithm 3). This observation is in line with Theorem 3, which states that the disagreement between the fully decentralized strategy and the baseline centralized training for decentralized execution strategy is bounded. Notice also that CC (Algorithm 1) results in a higher SBE compared to the diffusion and CD, despite being a fully centralized strategy. This is because, CC evaluates a different policy, namely, the centralized execution policy. Therefore, as argued in Section V-C, the SBE of CC is not a suitable baseline for the diffusion strategy.

VII. CONCLUDING REMARKS

In this paper, we proposed a policy evaluation algorithm for Dec-POMDPs over networks. We carried out a rigorous analysis that established: (i) the beliefs formed with local information and interactions have a bounded disagreement with the global posterior distribution, (ii) agents' value function parameters cluster around the network centroid, and (iii) the decentralized training can match the performance of the centralized training with appropriate parameters and increasing network connectivity.

There are two limitations of the current work that can be addressed in future work. First, we assume that agents know the local likelihood and transition models accurately. One possible question is if agents have approximation errors for the models, how would these affect the analytical results. Second, an implication of Theorem 3 is that there is necessity for regularization ($\rho > 0$). We leave the question of whether one can get bounds that does not require this, possibly with more assumptions on the model, to future work.

APPENDIX

A. PROOF OF THEOREM 1

We can rewrite the risk function as

$$\begin{aligned}
J_{k,i} &= \mathbb{E}_{\mathcal{F}_i} D_{\mathrm{KL}}(\boldsymbol{\mu}_i \| \boldsymbol{\mu}_{k,i}) \\
&= \mathbb{E}_{\mathcal{F}_i}\left[\sum_{s\in\mathcal{S}} \boldsymbol{\mu}_i(s) \log \frac{\boldsymbol{\mu}_i(s)}{\boldsymbol{\mu}_{k,i}(s)}\right] \\
&\overset{(a)}{=} \mathbb{E}_{\mathcal{F}_i}\left[\sum_{s\in\mathcal{S}} \mathbb{P}(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}_i) \log \frac{\boldsymbol{\mu}_i(s)}{\boldsymbol{\mu}_{k,i}(s)}\right] \\
&\overset{(b)}{=} \mathbb{E}_{\mathcal{F}_i}\left[\mathbb{E}_{s_i|\mathcal{F}_i}\left(\log \frac{\boldsymbol{\mu}_i(\boldsymbol{s}_i)}{\boldsymbol{\mu}_{k,i}(\boldsymbol{s}_i)}\right)\right] \\
&= \mathbb{E}_{\mathcal{F}_i,s_i}\left[\log \frac{\boldsymbol{\mu}_i(\boldsymbol{s}_i)}{\boldsymbol{\mu}_{k,i}(\boldsymbol{s}_i)}\right], \qquad (78)
\end{aligned}$$

where (a) follows from definition (9), (b) follows from the definition of conditional expectation with respect to $\boldsymbol{s}_i$ given $\boldsymbol{\mathcal{F}}_i$. Merging the diffusion adaptation step (37) and the combination step (38) together yields the following form:

$$\boldsymbol{\mu}_{k,i}(s) \propto \prod_{\ell\in\mathcal{N}_k} (L_\ell(\boldsymbol{\xi}_{\ell,i}|s))^{\beta c_{\ell k}} (\boldsymbol{\eta}_{\ell,i}(s))^{c_{\ell k}}, \qquad (79)$$

which, combined with the update (22) for the centralized solution, results in:

$$\begin{aligned}
\log \frac{\boldsymbol{\mu}_i(s)}{\boldsymbol{\mu}_{k,i}(s)} &= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\left(\log \frac{L(\boldsymbol{\xi}_i|s)}{(L_\ell(\boldsymbol{\xi}_{\ell,i}|s))^\beta} + \log \frac{\boldsymbol{\eta}_i(s)}{\boldsymbol{\eta}_{\ell,i}(s)}\right) \\
&\quad + \log \sum_{s'\in\mathcal{S}}\left(\prod_{\ell\in\mathcal{N}_k} (L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}} \prod_{\ell\in\mathcal{N}_k} (\boldsymbol{\eta}_{\ell,i}(s'))^{c_{\ell k}}\right) \\
&\quad - \log \boldsymbol{m}_i(\boldsymbol{\xi}_i). \qquad (80)
\end{aligned}$$

Here, we have introduced the marginal distribution of new observation given the past observations and actions:

$$\begin{aligned}
\boldsymbol{m}_i(\xi_i) \triangleq \mathbb{P}\left(\boldsymbol{\xi}_i = \xi_i|\boldsymbol{\mathcal{F}}^a_{i-1}\right) &= \sum_{s\in\mathcal{S}} \mathbb{P}\left(\boldsymbol{\xi}_i = \xi_i, \boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}^a_{i-1}\right) \\
&= \sum_{s\in\mathcal{S}} L(\xi_i|s)\mathbb{P}\left(\boldsymbol{s}_i = s|\boldsymbol{\mathcal{F}}^a_{i-1}\right) \\
&= \sum_{s\in\mathcal{S}} L(\xi_i|s)\boldsymbol{\eta}_i(s). \qquad (81)
\end{aligned}$$

First, observe that the expectation of the log-likelihood ratio terms in (80) satisfies:

$$\begin{aligned}
&\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\xi_i,s_i}\left[\log \frac{L(\boldsymbol{\xi}_i|\boldsymbol{s}_i)}{(L_\ell(\boldsymbol{\xi}_{\ell,i}|\boldsymbol{s}_i))^\beta}\right] \\
&\quad \overset{(a)}{=} \mathbb{E}_{\xi_i,s_i}\left[\sum_{\ell=1}^{K} \log L_\ell(\boldsymbol{\xi}_{\ell,i}|\boldsymbol{s}_i)\right] \\
&\qquad - \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\xi_{\ell,i},s_i}\left[\beta \log L_\ell(\boldsymbol{\xi}_{\ell,i}|\boldsymbol{s}_i)\right] \\
&\quad = \mathbb{E}_{\xi_i,s_i}\left[\sum_{\ell=1}^{K} (1-\beta c_{\ell k}) \log L_\ell(\boldsymbol{\xi}_{\ell,i}|\boldsymbol{s}_i)\right] \qquad (82)
\end{aligned}$$

where in (a) we used the spatial independency of the observations. Second, the expectation of the time-adjusted terms in (80) can be rewritten as:

$$\begin{aligned}
&\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}_i,s_i}\left[\log \frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right] \\
&\overset{(a)}{=} \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}_i,s_i}\left[\log \frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)} + \log \frac{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right] \\
&= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\mathbb{E}_{\xi_i|\mathcal{F}^a_{i-1},s_i}\left(\log \frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)} + \log \frac{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right)\right] \\
&\overset{(b)}{=} \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\log \frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)} + \log \frac{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right] \qquad (83)
\end{aligned}$$

where in (a) we define the agent-specific distribution:

$$\widetilde{\boldsymbol{\eta}}_{\ell,i}(s) \triangleq \sum_{s'\in\mathcal{S}} \mathbb{T}(s|s',\boldsymbol{a}_{i-1})\boldsymbol{\mu}_{\ell,i-1}(s'), \tag{84}$$

and (b) follows from the fact that the arguments are deterministic given the current state and the history of actions and observations. The first term of (83) can be written as a KL-divergence because of the following:

$$\begin{aligned}
&\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\log\frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}\right]\\
&= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1}}\left[\mathbb{E}_{s_i|\mathcal{F}^a_{i-1}}\left(\log\frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}\right)\right]\\
&= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1}}\left[\sum_{s\in\mathcal{S}}\mathbb{P}(\boldsymbol{s}_i=s|\boldsymbol{\mathcal{F}}^a_{i-1})\log\frac{\boldsymbol{\eta}_i(s)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(s)}\right]\\
&\overset{(11)}{=} \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1}}\left[\sum_{s\in\mathcal{S}}\boldsymbol{\eta}_i(s)\log\frac{\boldsymbol{\eta}_i(s)}{\widetilde{\boldsymbol{\eta}}_{\ell,i}(s)}\right]\\
&= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1}}\left[D_{\text{KL}}(\boldsymbol{\eta}_i\|\widetilde{\boldsymbol{\eta}}_{\ell,i})\right].
\end{aligned} \tag{85}$$

This expected KL-divergence can be bounded by using the strong-data processing inequality [75]:

$$\begin{aligned}
&\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1}}\left[D_{\text{KL}}(\boldsymbol{\eta}_i\|\widetilde{\boldsymbol{\eta}}_{\ell,i})\right]\\
&\quad\leq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\kappa(\mathbb{T})\underbrace{\mathbb{E}_{\mathcal{F}_{i-1}}\left[D_{\text{KL}}(\boldsymbol{\mu}_{i-1}\|\boldsymbol{\mu}_{\ell,i-1})\right]}_{J_{\ell,i-1}}.
\end{aligned} \tag{86}$$

The second term of (83) arises due to transition model disagreement with the centralized belief. To bound it, we first introduce the LogSumExp function f with vector arguments $\nu\in\mathbb{R}^S$:

$$f(\nu)\triangleq\log\sum_{s\in\mathcal{S}}\exp\{\nu(s)\}. \tag{87}$$

Its gradient is given by

$$\nabla_\nu f(\nu)\triangleq\text{col}\left\{\frac{\partial f(\nu)}{\partial\nu(s)}\right\}_{s\in\mathcal{S}}=\text{col}\left\{\frac{\exp\{\nu(s)\}}{\sum_{s'}\exp\{\nu(s')\}}\right\}_{s\in\mathcal{S}}. \tag{88}$$

Observe that if we define the vectors

$$\widetilde{\boldsymbol{\nu}}_{\ell,i}\triangleq\text{col}\left\{\log\left(\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})\boldsymbol{\mu}_{\ell,i-1}(s)\right)\right\}_{s\in\mathcal{S}} \tag{89}$$

and

$$\boldsymbol{\nu}_{\ell,i}\triangleq\text{col}\left\{\log\left(\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})\boldsymbol{\mu}_{\ell,i-1}(s)\right)\right\}_{s\in\mathcal{S}}, \tag{90}$$

then, we can rewrite the second expression of (83) as follows:

$$\begin{aligned}
&\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\log\frac{\widetilde{\boldsymbol{\eta}}_{\ell,i}(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right]\\
&\quad= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[f(\widetilde{\boldsymbol{\nu}}_{\ell,i})-f(\boldsymbol{\nu}_{\ell,i})\right].
\end{aligned} \tag{91}$$

Applying mean value theorem to this difference yields

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[f(\widetilde{\boldsymbol{\nu}}_{\ell,i})-f(\boldsymbol{\nu}_{\ell,i})\right]\\
&=\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[(\nabla_\nu f(\overline{\boldsymbol{\nu}}_{\ell,i}))^\mathsf{T}\cdot(\widetilde{\boldsymbol{\nu}}_{\ell,i}-\boldsymbol{\nu}_{\ell,i})\right]\\
&\overset{(88)}{=}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}^\mathsf{T}_{s\in\mathcal{S}}\cdot(\widetilde{\boldsymbol{\nu}}_{\ell,i}-\boldsymbol{\nu}_{\ell,i})\right]\\
&\overset{(89),(90)}{=}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}^\mathsf{T}_{s\in\mathcal{S}}\right.\\
&\qquad\left.\cdot\text{col}\left\{\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})}\right\}_{s\in\mathcal{S}}\right]
\end{aligned} \tag{92}$$

for some $\overline{\boldsymbol{\nu}}_{\ell,i}$ between $\widetilde{\boldsymbol{\nu}}_{\ell,i}$ and $\boldsymbol{\nu}_{\ell,i}$. The term in (92) is bounded as follows:

$$\begin{aligned}
&\left|\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}^\mathsf{T}_{s\in\mathcal{S}}\right.\right.\\
&\qquad\left.\left.\cdot\text{col}\left\{\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})}\right\}_{s\in\mathcal{S}}\right]\right|\\
&\overset{(a)}{\leq}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left|\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}^\mathsf{T}_{s\in\mathcal{S}}\right.\\
&\qquad\left.\cdot\text{col}\left\{\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})}\right\}_{s\in\mathcal{S}}\right|\\
&\overset{(b)}{\leq}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\left\|\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}_{s\in\mathcal{S}}\right\|_1\right.\\
&\qquad\left.\cdot\left\|\text{col}\left\{\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})}\right\}_{s\in\mathcal{S}}\right\|_\infty\right]\\
&\overset{(c)}{=}\mathbb{E}_{s_i,a_{i-1}}\left\|\text{col}\left\{\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_\ell(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_\ell,i-1})}\right\}_{s\in\mathcal{S}}\right\|_\infty
\end{aligned} \tag{93}$$

where (a) follows from the Jensen's inequality, (b) follows from the Hölder's inequality, and (c) follows from the fact that

$$\left\|\text{col}\left\{\frac{\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s)\}}{\sum_{s'\in\mathcal{S}}\exp\{\overline{\boldsymbol{\nu}}_{\ell,i}(s')\}}\right\}\right\|_1=1. \tag{94}$$

Furthermore, due to Assumption 1 and to the fact that the number of maximum hops outside $\mathcal{N}_k$ is $(K-|\mathcal{N}_k|)$, we have

$$\begin{aligned}
\left|\log\frac{\mathbb{T}(\boldsymbol{s}_i|s,\boldsymbol{a}_{i-1})}{\mathbb{T}^\pi_k(\boldsymbol{s}_i|s,\boldsymbol{a}_{\mathcal{N}_k,i-1})}\right| &\leq (K-|\mathcal{N}_k|)\,\tau\\
&\leq (K-d_{\min})\,\tau.
\end{aligned} \tag{95}$$

If we combine (86), (91), and (95), the expectation of the time-adjusted terms in (80) can be bounded as:

$$\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\mathbb{E}_{\mathcal{F}^a_{i-1},s_i}\left[\log\frac{\boldsymbol{\eta}_i(\boldsymbol{s}_i)}{\boldsymbol{\eta}_{\ell,i}(\boldsymbol{s}_i)}\right]$$

$$\leq (K - d_{\min})\,\tau + \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\kappa(\mathbb{T})J_{\ell,i-1} \tag{96}$$

Next, we bound the expectation of the remaining normalization terms in (80), which follows similar steps to what was done in [49]:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell\in\mathcal{N}_k}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\prod_{\ell\in\mathcal{N}_k}(\boldsymbol{\eta}_{\ell,i}(s'))^{c_{\ell k}}\right)\right]\\
&\quad-\mathbb{E}_{\mathcal{F}_i}\left[\log \boldsymbol{m}_i(\boldsymbol{\xi}_i)\right]\\
&\overset{(a)}{\leq}\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell\in\mathcal{N}_k}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad-\mathbb{E}_{\mathcal{F}_i}\left[\log \boldsymbol{m}_i(\boldsymbol{\xi}_i)\right]\\
&=\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell\in\mathcal{N}_k}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad-\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad+\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad-\mathbb{E}_{\mathcal{F}_i}\left[\log \boldsymbol{m}_i(\boldsymbol{\xi}_i)\right]\\
&\overset{(b)}{\leq}\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad-\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]
\end{aligned}\tag{97}$$

where (a) follows from the arithmetic-geometric mean inequality, (b) follows from:

$$\begin{aligned}
&-\mathbb{E}_{\mathcal{F}_i}\left[\log\frac{\boldsymbol{m}_i(\boldsymbol{\xi}_i)}{\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)}\right]\\
&\qquad=-\mathbb{E}_{\mathcal{F}^a_{i-1}}\mathbb{E}_{\xi_i|\mathcal{F}^a_{i-1}}\left[\log\frac{\boldsymbol{m}_i(\boldsymbol{\xi}_i)}{\boldsymbol{m}^\dagger_i(\boldsymbol{\xi}_i)}\right]\\
&\qquad=-\mathbb{E}_{\mathcal{F}^a_{i-1}}D_{\mathrm{KL}}(\boldsymbol{m}_i(\boldsymbol{\xi}_i)\|\boldsymbol{m}^\dagger_i(\boldsymbol{\xi}_i))\\
&\qquad\leq 0
\end{aligned}\tag{98}$$

where we use the definition:

$$\boldsymbol{m}^\dagger_i(\boldsymbol{\xi}_i)\triangleq\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right), \tag{99}$$

which is a density (or mass function if observations are discrete) since:

$$\begin{aligned}
\int_{\xi_i}\boldsymbol{m}^\dagger_i(\xi_i)d\xi_i &= \int_{\xi_i}\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)d\xi_i\\
&=\sum_{s'\in\mathcal{S}}\left[\underbrace{\int_{\xi_i}\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')d\xi_i}_{1}\sum_{\ell=1}^{K}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right]\\
&=\sum_{s'\in\mathcal{S}}\left[\sum_{\ell=1}^{K}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right]\\
&=\sum_{\ell=1}^{K}c_{\ell k}\left[\sum_{s'\in\mathcal{S}}\boldsymbol{\eta}_{\ell,i}(s')\right]=1.
\end{aligned}\tag{100}$$

Notice that the expression in (97) can be rewritten as

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\qquad-\mathbb{E}_{\mathcal{F}_i}\left[\log\sum_{s'\in\mathcal{S}}\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right]\\
&\quad=\mathbb{E}_{\mathcal{F}_i}\left[f(\boldsymbol{\vartheta}_{k,i})\right]-\mathbb{E}_{\mathcal{F}_i}\left[f(\widetilde{\boldsymbol{\vartheta}}_{k,i})\right],
\end{aligned}\tag{101}$$

if we use the LogSumExp function f from (87) and use the definitions:

$$\boldsymbol{\vartheta}_{k,i}\triangleq\mathrm{col}\left\{\log\left(\prod_{\ell=1}^{K}(L_\ell(\boldsymbol{\xi}_{\ell,i}|s'))^{\beta c_{\ell k}}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right\}_{s'\in\mathcal{S}} \tag{102}$$

and

$$\widetilde{\boldsymbol{\vartheta}}_{k,i}\triangleq\mathrm{col}\left\{\log\left(\prod_{\ell=1}^{K}L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\boldsymbol{\eta}_{\ell,i}(s')\right)\right\}_{s'\in\mathcal{S}}. \tag{103}$$

Following the steps in (92) and (93), this difference can be bounded as:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}_i}\left[f(\boldsymbol{\vartheta}_{k,i})\right]-\mathbb{E}_{\mathcal{F}_i}\left[f(\widetilde{\boldsymbol{\vartheta}}_{k,i})\right]\\
&\leq\mathbb{E}_{\xi_i}\left\|\mathrm{col}\left\{\sum_{\ell=1}^{K}(\beta c_{\ell k}-1)\log L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\right\}_{s'\in\mathcal{S}}\right\|_\infty.
\end{aligned}\tag{104}$$

Moreover, by assumptions on the graph topology (27) and on the likelihood functions (46), this expression can be further bounded as [49]:

$$\left\|\mathrm{col}\left\{\sum_{\ell=1}^{K}(\beta c_{\ell k}-1)\log L_\ell(\boldsymbol{\xi}_{\ell,i}|s')\right\}_{s'\in\mathcal{S}}\right\|_\infty\leq\sqrt{K}\beta\lambda B \tag{105}$$

Subsequently, if we insert the bounds (82), (96), and (105) to (80), we arrive at the bound on the risk function:

$$\begin{aligned}J_{k,i} &\le \mathbb{E}_{\xi_i,s_i}\left[\sum_{\ell=1}^{K}(1-\beta c_{\ell k})\log L_\ell(\boldsymbol{\xi}_{\ell,i}|\boldsymbol{s}_i)\right]\\&\quad+\kappa(\mathbb{T})\sum_{\ell\in\mathcal{N}_k}c_{\ell k}J_{\ell,i-1}+\sqrt{K}\beta\lambda B+(K-d_{\min})\,\tau\\&\overset{(105)}{\le}\kappa(\mathbb{T})\sum_{\ell\in\mathcal{N}_k}c_{\ell k}J_{\ell,i-1}+2\sqrt{K}\beta\lambda B+(K-d_{\min})\,\tau.\end{aligned}\tag{106}$$

Expanding this recursion over time yields:

$$\begin{aligned}J_{k,i} &\le (2\sqrt{K}\beta\lambda B+(K-d_{\min})\,\tau)\sum_{j=0}^{i-1}(\kappa(\mathbb{T}))^j\\&\quad+(\kappa(\mathbb{T}))^i\sum_{\ell=1}^{K}[C^i]_{\ell k}J_{\ell,0}\\&=\frac{1-(\kappa(\mathbb{T}))^i}{1-\kappa(\mathbb{T})}(2\sqrt{K}\beta\lambda B+(K-d_{\min})\,\tau)\\&\quad+(\kappa(\mathbb{T}))^i\sum_{\ell=1}^{K}[C^i]_{\ell k}J_{\ell,0},\end{aligned}\tag{107}$$

which implies that if $\kappa(\mathbb{T})<1$, the risk function is bounded as $i\to\infty$:

$$\limsup_{i\to\infty}J_{k,i}\le\frac{2\sqrt{K}\beta\lambda B+(K-d_{\min})\,\tau}{1-\kappa(\mathbb{T})}.\tag{108}$$

By (96), this also implies that

$$\begin{aligned}\limsup_{i\to\infty}\widetilde{J}_{k,i}&\le(K-d_{\min})\tau+\kappa(\mathbb{T})\limsup_{i\to\infty}J_{k,i}\\&\le\frac{(K-d_{\min})\tau}{1-\kappa(\mathbb{T})}+\kappa(\mathbb{T})\frac{2\sqrt{K}\beta\lambda B}{1-\kappa(\mathbb{T})}.\end{aligned}\tag{109}$$

B. PROOF OF COROLLARY 1

In view of the Bretagnolle-Huber inequality [76], it holds that

$$\sum_{s\in\mathcal{S}}\left|\boldsymbol{\mu}_i(s)-\boldsymbol{\mu}_{k,i}(s)\right|\le 2\left(1-\exp\{-D_{\mathrm{KL}}(\boldsymbol{\mu}_i\|\boldsymbol{\mu}_{k,i})\}\right)^{\frac{1}{2}}.\tag{110}$$

If we take the expectation of both sides, we get:

$$\begin{aligned}\mathbb{E}\left[\sum_{s\in\mathcal{S}}\left|\boldsymbol{\mu}_i(s)-\boldsymbol{\mu}_{k,i}(s)\right|\right]&\le 2\mathbb{E}\left(1-\exp\{-D_{\mathrm{KL}}(\boldsymbol{\mu}_i\|\boldsymbol{\mu}_{k,i})\}\right)^{\frac{1}{2}}\\&\overset{(a)}{\le}2\left(1-\mathbb{E}\exp\{-D_{\mathrm{KL}}(\boldsymbol{\mu}_i\|\boldsymbol{\mu}_{k,i})\}\right)^{\frac{1}{2}}\\&\overset{(b)}{\le}2\left(1-\exp\{-J_{k,i}\}\right)^{\frac{1}{2}},\end{aligned}\tag{111}$$

where (a) and (b) follow from Jensen's inequality. Together with Theorem 1, this implies that

$$\mathbb{E}\left\|\boldsymbol{\mu}_i-\boldsymbol{\mu}_{k,i}\right\|_1\le B_{\mathrm{TV}},\tag{112}$$

where we use the definition (55). Furthermore, on account of the fact that ℓ_2 norm is no greater than ℓ_1 norm in $\mathbb{R}^S$, it is also true that

$$\mathbb{E}\left\|\boldsymbol{\mu}_i-\boldsymbol{\mu}_{k,i}\right\|\le B_{\mathrm{TV}}.\tag{113}$$

With similar arguments, it can be shown that

$$\mathbb{E}\left\|\boldsymbol{\eta}_i-\boldsymbol{\eta}_{k,i}\right\|\le\widetilde{B}_{\mathrm{TV}},\tag{114}$$

where we use the definition (56).

C PROOF OF LEMMA 1

Inserting the definitions (59) and (63), the expected difference can be expanded as

$$\begin{aligned}\mathbb{E}\|\boldsymbol{H}_{k,i}-\boldsymbol{H}^\star_i\|&=\mathbb{E}\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\gamma\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\right.\\&\qquad\left.-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\mu}_i)^{\mathsf{T}}+\gamma\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}\right\|\\&\le\mathbb{E}\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\mu}_i)^{\mathsf{T}}\right\|\\&\quad+\gamma\mathbb{E}\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}\right\|,\end{aligned}\tag{115}$$

where the last step follows from the triangle inequality. Here, the first term can be bounded as

$$\begin{aligned}&\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\mu}_i)^{\mathsf{T}}\right\|\\&\le\left\|\phi(\boldsymbol{\mu}_{k,i})(\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)^{\mathsf{T}})\right\|\\&\quad+\left\|(\phi(\boldsymbol{\mu}_{k,i})-\phi(\boldsymbol{\mu}_i))\phi(\boldsymbol{\mu}_i)^{\mathsf{T}}\right\|\\&\le\left\|\phi(\boldsymbol{\mu}_{k,i})\right\|\left\|\phi(\boldsymbol{\mu}_{k,i})-\phi(\boldsymbol{\mu}_i)\right\|\\&\quad+\left\|\phi(\boldsymbol{\mu}_{k,i})-\phi(\boldsymbol{\mu}_i)\right\|\left\|\phi(\boldsymbol{\mu}_i)\right\|\\&\overset{(a)}{\le}B_\phi L_\phi\|\boldsymbol{\mu}_{k,i}-\boldsymbol{\mu}_i\|+B_\phi L_\phi\|\boldsymbol{\mu}_{k,i}-\boldsymbol{\mu}_i\|,\end{aligned}\tag{116}$$

where (a) follows from Assumption 2. Taking expectations and using (53) and (116), it follows that

$$\mathbb{E}\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\mu}_i)^{\mathsf{T}}\right\|\le 2B_\phi L_\phi B_{\mathrm{TV}}.\tag{117}$$

Similarly, the second term in (115) can be bounded as

$$\begin{aligned}&\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}-\phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}\right\|\\&\le\left\|\phi(\boldsymbol{\mu}_{k,i})(\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}-\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}})\right\|\\&\quad+\left\|(\phi(\boldsymbol{\mu}_{k,i})-\phi(\boldsymbol{\mu}_i))\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}\right\|\\&\le\left\|\phi(\boldsymbol{\mu}_{k,i})\right\|\left\|\phi(\boldsymbol{\eta}_{k,i+1})-\phi(\boldsymbol{\eta}_{i+1})\right\|\\&\quad+\left\|\phi(\boldsymbol{\mu}_{k,i})-\phi(\boldsymbol{\mu}_i)\right\|\left\|\phi(\boldsymbol{\eta}_{i+1})\right\|\\&\overset{(a)}{\le}B_\phi L_\phi\|\boldsymbol{\eta}_{k,i+1}-\boldsymbol{\eta}_{i+1}\|+B_\phi L_\phi\|\boldsymbol{\mu}_{k,i}-\boldsymbol{\mu}_i\|\end{aligned}\tag{118}$$

where (a) follows from Assumption 2. Using (53) and (54) we get:

$$\mathbb{E}\left\|\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}} - \phi(\boldsymbol{\mu}_i)\phi(\boldsymbol{\eta}_{i+1})^{\mathsf{T}}\right\| \leq B_\phi L_\phi(B_{\mathrm{TV}} + \widetilde{B}_{\mathrm{TV}}). \tag{119}$$

Combining (117) and (119) in addition to the fact that $\widetilde{B}_{\mathrm{TV}} \leq B_{\mathrm{TV}}$ (since $\kappa(\mathbb{T}) < 1$) yields:

$$\mathbb{E}\|\boldsymbol{H}_{k,i} - \boldsymbol{H}_i^\star\| \leq 2B_\phi L_\phi B_{\mathrm{TV}}(1+\gamma). \tag{120}$$

D PROOF OF THEOREM 2

For compactness of notation, it is useful to introduce the following quantities, which collect variables from across all agents:

$$\mathcal{W}_i \triangleq \mathrm{col}\left\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{K,i}\right\} \tag{121}$$

$$\mathcal{C} \triangleq C \otimes I_M \tag{122}$$

$$\boldsymbol{\mathcal{H}}_i \triangleq \mathrm{diag}\left\{\boldsymbol{H}_{k,i}\right\}_{k=1}^K \tag{123}$$

$$\boldsymbol{\mathcal{H}}_i^\star \triangleq I_K \otimes \boldsymbol{H}_i^\star \tag{124}$$

$$\boldsymbol{d}_i \triangleq \mathrm{col}\left\{\boldsymbol{d}_{k,i}\right\}_{k=1}^K \tag{125}$$

Then, the equations (40)–(42) can be written as:

$$\mathcal{W}_{i+1} = \mathcal{C}^{\mathsf{T}}\left((I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i)\,\mathcal{W}_i + \alpha\boldsymbol{d}_i\right). \tag{126}$$

Moreover, we can define the following K-times extended centroid vector:

$$\mathcal{W}_{c,i} \triangleq \mathbb{1}_K \otimes \boldsymbol{w}_{c,i} = \left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right)\mathcal{W}_i\,. \tag{127}$$

If we decompose $\boldsymbol{\mathcal{H}}_i$ into its centralized component $\boldsymbol{\mathcal{H}}_i^\star$ and the respective disagreement matrix $\boldsymbol{\Delta}_i \triangleq \boldsymbol{\mathcal{H}}_i - \boldsymbol{\mathcal{H}}_i^\star$, we obtain:

$$\begin{aligned}
&\mathcal{W}_{i+1} - \mathcal{W}_{c,i+1}\\
&= \left(\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right)\left((I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i)\,\mathcal{W}_i + \alpha\boldsymbol{d}_i\right)\\
&= \left(\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right)\\
&\qquad\left(\left(I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star - \alpha\boldsymbol{\Delta}_i\right)\mathcal{W}_i + \alpha\boldsymbol{d}_i\right)\\
&= \left(\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right)\\
&\qquad\left(\left(I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star\right)(\mathcal{W}_i - \mathcal{W}_{c,i}) - \alpha\boldsymbol{\Delta}_i\,\mathcal{W}_i + \alpha\boldsymbol{d}_i\right),
\end{aligned} \tag{128}$$

where the last step follows from the fact that

$$\begin{aligned}
&\mathcal{C}^{\mathsf{T}}\left(I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star\right)\mathcal{W}_{c,i}\\
&= \left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right)\left(I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star\right)\mathcal{W}_{c,i}\,.
\end{aligned} \tag{129}$$

Furthermore, taking the norms of both sides in (128) leads to

$$\begin{aligned}
&\left\|\mathcal{W}_{i+1} - \mathcal{W}_{c,i+1}\right\|\\
&\leq \left\|\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right\|\\
&\qquad\left\|\left(I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star\right)(\mathcal{W}_i - \mathcal{W}_{c,i}) - \alpha\boldsymbol{\Delta}_i\,\mathcal{W}_i + \alpha\boldsymbol{d}_i\right\|\\
&\leq \left\|\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right\|\left\|I(1-2\alpha\rho) - \alpha\boldsymbol{\mathcal{H}}_i^\star\right\|\|\,\mathcal{W}_i - \mathcal{W}_{c,i}\,\|\\
&\quad + \alpha\left\|\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right\|\left(\|\boldsymbol{\Delta}_i\|\|\,\mathcal{W}_i\,\| + \|\boldsymbol{d}_i\|\right).
\end{aligned} \tag{130}$$

Since the combination matrix C is a primitive stochastic matrix, it follows from the Perron-Frobenius theorem [53], [77] that its maximum eigenvalue is 1, and all other eigenvalues are strictly smaller than 1 in absolute value. Moreover, C is assumed to be symmetric, therefore its eigenvalue decomposition has the following form:

$$\begin{aligned}
C &= U\Lambda U^\top\\
&= \begin{bmatrix} u_1 & u_2 & \cdots & u_K \end{bmatrix}\begin{bmatrix} 1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_K \end{bmatrix}\begin{bmatrix} u_1^\top\\ u_2^\top\\ \vdots\\ u_K^\top \end{bmatrix}
\end{aligned}$$

where U is the orthogonal matrix of eigenvectors $\{u_k\}$, and Λ is the diagonal matrix of eigenvalues. Additionally, the powers of C converge (because it is primitive) to the scaled all-ones matrix (because it is doubly-stochastic):

$$\begin{aligned}
\lim_{i\to\infty} C^i &= \begin{bmatrix} u_1 & u_2 & \cdots & u_K \end{bmatrix}\begin{bmatrix} 1 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} u_1^\top\\ u_2^\top\\ \vdots\\ u_K^\top \end{bmatrix}\\
&= \frac{1}{K}\begin{bmatrix} 1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1 \end{bmatrix} = \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\top
\end{aligned}$$

Therefore, the difference of these matrices becomes:

$$C - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\top = U\begin{bmatrix} 0 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_K \end{bmatrix}U^\top, \tag{131}$$

which implies:

$$\left\|C - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\top\right\| = \lambda_2 \tag{132}$$

where λ_2 is the second largest modulus eigenvalue of C. Moreover, the Kronecker product with the identity matrix does not change the spectral norm, hence:

$$\left\|\mathcal{C}^{\mathsf{T}} - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^{\mathsf{T}} \otimes I\right\| = \lambda_2 < 1. \tag{133}$$

Moreover, we know from Lemma 1 that

$$\mathbb{E}\|\boldsymbol{\Delta}_i\| \le 2B_\phi L_\phi B_{\mathrm{TV}}(1+\gamma). \tag{134}$$

Additionally, in Appendix F, we establish (135)–(138) which hold for any realization (with probability one). From (161), note that:

$$\left\|I(1-2\alpha\rho)-\alpha\boldsymbol{\mathcal{H}}_i^\star\right\| < 1 \tag{135}$$

whenever $\rho > \gamma L_\phi B_\phi/\sqrt{2}$. Specifically, if $\rho \ge 0.75\gamma L_\phi B_\phi$, then

$$\left\|I(1-2\alpha\rho)-\alpha\boldsymbol{\mathcal{H}}_i^\star\right\| \le (1-0.08\alpha\gamma L_\phi B_\phi). \tag{136}$$

In addition, we show in Lemma 2 that

$$\|\boldsymbol{\mathcal{W}}_i\| \le \frac{\sqrt{K}R_{\max}}{0.08\gamma L_\phi} \tag{137}$$

and in expression (162) that

$$\|\boldsymbol{d}_i\| \le \sqrt{K}R_{\max}B_\phi. \tag{138}$$

Inserting these results into (130) yields the following norm recursion:

$$\begin{aligned}&\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_{i+1}-\boldsymbol{\mathcal{W}}_{c,i+1}\right\|\\ &\le \lambda_2(1-0.08\alpha\gamma B_\phi L_\phi)\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_i-\boldsymbol{\mathcal{W}}_{c,i}\right\| + \alpha\lambda_2\sqrt{K}\epsilon.\end{aligned} \tag{139}$$

Let us define the constant $\widetilde{\lambda}_2 \triangleq \lambda_2(1-0.08\alpha\gamma B_\phi L_\phi)$. Iterating (139) over time, we arrive at

$$\begin{aligned}&\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_{i+1}-\boldsymbol{\mathcal{W}}_{c,i+1}\right\|\\ &\le \widetilde{\lambda}_2^{i+1}\|\,\mathcal{W}_0-\mathcal{W}_{c,0}\,\| + \alpha\lambda_2\sqrt{K}\epsilon\sum_{j=1}^{i+1}\widetilde{\lambda}_2^{i+1-j}\\ &\le \widetilde{\lambda}_2^{i+1}\|\,\mathcal{W}_0-\mathcal{W}_{c,0}\,\| + \alpha\lambda_2\sqrt{K}\epsilon\frac{1}{1-\widetilde{\lambda}_2}\\ &\overset{(a)}{\le} \alpha\lambda_2\sqrt{K}\epsilon\frac{1}{1-\widetilde{\lambda}_2}+O(\alpha^2)\end{aligned} \tag{140}$$

where (a) holds whenever:

$$\begin{aligned}&\widetilde{\lambda}_2^i\|\,\mathcal{W}_0-\mathcal{W}_{c,0}\,\| \le c\alpha^2\\ &\iff i\log\widetilde{\lambda}_2 \le 2\log\alpha+\log c-\log\|\,\mathcal{W}_0-\mathcal{W}_{c,0}\,\|\\ &\iff i \ge \frac{2\log\alpha}{\log\widetilde{\lambda}_2}+O(1)=O(\log\alpha)=o(1/\alpha),\end{aligned} \tag{141}$$

where c is an arbitrary constant.

E PROOF OF THEOREM 3

We begin by rewriting the baseline strategy recursion (72)–(73) in the form:

$$\boldsymbol{w}_{i+1}^\star = \left((1-2\rho\alpha)I-\alpha\boldsymbol{H}_i^\star\right)\boldsymbol{w}_i^\star+\alpha\boldsymbol{d}_i^\star, \tag{142}$$

where $\boldsymbol{H}_i^\star$ is defined in (63), and

$$\boldsymbol{d}_i^\star \triangleq \left(\frac{1}{K}\sum_{k=1}^K\boldsymbol{r}_{k,i}\right)\phi(\boldsymbol{\mu}_i). \tag{143}$$

We introduce the K-times extended versions of the vectors:

$$\boldsymbol{\mathcal{D}}_i^\star \triangleq \mathbb{1}_K\otimes\boldsymbol{d}_i^\star,\quad \boldsymbol{\mathcal{W}}_i^\star=\mathbb{1}_K\otimes\boldsymbol{w}_i^\star. \tag{144}$$

Then, the baseline recursion (142) transforms into

$$\boldsymbol{\mathcal{W}}_{i+1}^\star = \left((1-2\rho\alpha)I-\alpha\boldsymbol{\mathcal{H}}_i^\star\right)\boldsymbol{\mathcal{W}}_i^\star+\alpha\boldsymbol{\mathcal{D}}_i^\star. \tag{145}$$

It follows from the extended network centroid definition (127) and (145) that

$$\begin{aligned}&\boldsymbol{\mathcal{W}}_{i+1}^\star-\boldsymbol{\mathcal{W}}_{c,i+1}\\ &=\left(I(1-2\alpha\rho)-\alpha\boldsymbol{\mathcal{H}}_i^\star\right)(\boldsymbol{\mathcal{W}}_i^\star-\boldsymbol{\mathcal{W}}_{c,i})\\ &\quad-\alpha\left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right)\boldsymbol{\Delta}_i\boldsymbol{\mathcal{W}}_i+\alpha\left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right)(\boldsymbol{\mathcal{D}}_i^\star-\boldsymbol{d}_i)\end{aligned} \tag{146}$$

where we used the facts that

$$\left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right)\boldsymbol{\mathcal{D}}_i^\star=\boldsymbol{\mathcal{D}}_i^\star, \tag{147}$$

and

$$\left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right)\boldsymbol{\mathcal{H}}_i^\star\boldsymbol{\mathcal{W}}_i=\boldsymbol{\mathcal{H}}_i^\star\boldsymbol{\mathcal{W}}_{c,i}\,. \tag{148}$$

Next, if we define the following average agent disagreement relative to the baseline term

$$\widetilde{\boldsymbol{d}}_i \triangleq \frac{1}{K}\sum_{k=1}^K(\boldsymbol{d}_i^\star-\boldsymbol{d}_{k,i}), \tag{149}$$

it holds that

$$\widetilde{\boldsymbol{\mathcal{D}}}_i \triangleq \mathbb{1}_K\otimes\widetilde{\boldsymbol{d}}_i=\left(\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right)(\boldsymbol{\mathcal{D}}_i^\star-\boldsymbol{d}_i). \tag{150}$$

Subsequently, taking the norm of both sides in (146) and applying the triangle inequality, we get

$$\begin{aligned}&\left\|\boldsymbol{\mathcal{W}}_{i+1}^\star-\boldsymbol{\mathcal{W}}_{c,i+1}\right\|\\ &\le\left\|I(1-2\alpha\rho)-\alpha\boldsymbol{\mathcal{H}}_i^\star\right\|\left\|\boldsymbol{\mathcal{W}}_i^\star-\boldsymbol{\mathcal{W}}_{c,i}\right\|\\ &\quad+\alpha\left\|\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right\|\|\boldsymbol{\Delta}_i\|\,\|\boldsymbol{\mathcal{W}}_i\|+\alpha\left\|\widetilde{\boldsymbol{\mathcal{D}}}_i\right\|.\end{aligned} \tag{151}$$

First, observe that

$$\left\|\frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\mathsf{T}\otimes I\right\|=1. \tag{152}$$

Moreover, from Assumption 2 and Corollary 1, it holds that

$$\begin{aligned}\mathbb{E}\left\|\widetilde{\boldsymbol{d}}_i\right\|&=\mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^K\boldsymbol{r}_{k,i}(\phi(\boldsymbol{\mu}_i)-\phi(\boldsymbol{\mu}_{k,i}))\right\|\\ &\le R_{\max}L_\phi B_{\mathrm{TV}},\end{aligned} \tag{153}$$

and accordingly,

$$\mathbb{E}\left\|\widetilde{\boldsymbol{\mathcal{D}}}_i\right\|\le\sqrt{K}R_{\max}L_\phi B_{\mathrm{TV}}. \tag{154}$$

By using the same bounds (135)–(138) from Appendix D for the other terms (which are established in Lemma 1, Lemma 2, (161), and (162)), we arrive at the recursion:

$$\begin{aligned}&\mathbb{E}\left\|\mathcal{W}^\star_{i+1}-\mathcal{W}_{c,i+1}\right\|\\&\quad\le(1-0.08\alpha\gamma B_\phi L_\phi)\mathbb{E}\left\|\mathcal{W}^\star_i-\mathcal{W}_{c,i}\right\|+\alpha\sqrt{K}\epsilon^\star,\end{aligned}\tag{155}$$

where

$$\epsilon^\star\triangleq R_{\max}B_{\mathrm{TV}}\left(\frac{2B_\phi(1+\gamma)}{0.08\gamma}+L_\phi\right).\tag{156}$$

Iterating over time, we get:

$$\begin{aligned}&\mathbb{E}\left\|\mathcal{W}^\star_{i+1}-\mathcal{W}_{c,i+1}\right\|\\&\quad\le\ (1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\|\,\mathcal{W}^\star_0-\mathcal{W}_{c,0}\,\|\\&\qquad+\alpha\sqrt{K}\epsilon^\star\sum_{j=1}^{i+1}(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1-j}\\&\quad\le\ (1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\|\,\mathcal{W}^\star_0-\mathcal{W}_{c,0}\,\|+\frac{\sqrt{K}\epsilon^\star}{0.08\gamma B_\phi L_\phi}\\&\quad\overset{(a)}{\le}\ \frac{\sqrt{K}\epsilon^\star}{0.08\gamma B_\phi L_\phi}+o(1)\end{aligned}\tag{157}$$

where (a) holds whenever

$$\begin{aligned}&(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\left\|\mathcal{W}^\star_0-\mathcal{W}_{c,0}\right\|=o(1)\\&\iff i\log(1-0.08\alpha\gamma B_\phi L_\phi)=o(1)\\&\iff i\ge\frac{o(1)}{\log(1-0.08\alpha\gamma B_\phi L_\phi)}\ge o\left(\frac{1}{\alpha\gamma B_\phi L_\phi}\right).\end{aligned}\tag{158}$$

F AUXILIARY RESULTS

In the following lemma, we prove that the value function parameters are bounded in norm.

Lemma 2 (Bounded parameters): For each agent $k\in\mathcal{K}$, the iterate $\boldsymbol{w}_{k,i}$ is bounded in norm if $\rho>\gamma B_\phi L_\phi/\sqrt{2}$, with probability one. In particular, if $\rho\ge 0.75\gamma B_\phi L_\phi$, then

$$\|\mathcal{W}_i\|\le\frac{\sqrt{K}R_{\max}}{0.08\gamma L_\phi}\tag{159}$$

after $i\ge i_0=o(1/(\alpha\gamma B_\phi L_\phi))$ iterations.

Proof: Taking the norms of both sides of (126) yields:

$$\begin{aligned}\|\mathcal{W}_{i+1}\|&=\left\|\mathcal{C}^{\mathsf{T}}\left(((1-2\alpha\rho)I-\alpha\mathcal{H}_i)\,\mathcal{W}_i+\alpha\boldsymbol{d}_i\right)\right\|\\&\le\left\|\mathcal{C}^{\mathsf{T}}\right\|\,\|((1-2\alpha\rho)I-\alpha\mathcal{H}_i)\,\mathcal{W}_i+\alpha\boldsymbol{d}_i\|\\&\overset{(a)}{\le}\|((1-2\alpha\rho)I-\alpha\mathcal{H}_i)\,\mathcal{W}_i+\alpha\boldsymbol{d}_i\|\\&\le\|(1-2\alpha\rho)I-\alpha\mathcal{H}_i\|\,\|\mathcal{W}_i\|+\alpha\,\|\boldsymbol{d}_i\|\end{aligned}\tag{160}$$

where (a) follows from the fact that the singular values of doubly-stochastic matrices are equal to one. Note that

$$\begin{aligned}&\left\|(1-2\alpha\rho)I-\alpha\mathcal{H}_{k,i}\right\|\\&=\left\|(1-2\alpha\rho)I-\alpha\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}+\alpha\gamma\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\right\|\\&=\Big\|(1-2\alpha\rho)I-\alpha(1-\gamma)\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}\\&\qquad-\alpha\gamma\phi(\boldsymbol{\mu}_{k,i})\left(\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\right)\Big\|\\&\le\left\|(1-2\alpha\rho)I-\alpha(1-\gamma)\phi(\boldsymbol{\mu}_{k,i})\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}\right\|\\&\qquad+\alpha\gamma\|\phi(\boldsymbol{\mu}_{k,i})\|\left\|\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\right\|\\&\overset{(a)}{\le}(1-2\alpha\rho)+\alpha\gamma\|\phi(\boldsymbol{\mu}_{k,i})\|\left\|\phi(\boldsymbol{\mu}_{k,i})^{\mathsf{T}}-\phi(\boldsymbol{\eta}_{k,i+1})^{\mathsf{T}}\right\|\\&\overset{(b)}{\le}(1-2\alpha\rho)+\alpha\gamma B_\phi L_\phi\left\|\boldsymbol{\mu}_{k,i}-\boldsymbol{\eta}_{k,i+1}\right\|\\&\overset{(c)}{\le}(1-2\alpha\rho)+\alpha\gamma B_\phi L_\phi\sqrt{2}\end{aligned}\tag{161}$$

where (a) follows from the equality of spectral norm and maximum eigenvalue for symmetric matrices, (b) follows from Assumption 2, and (c) follows from the fact that the mean-square distance cannot exceed 2 over the probability simplex. The upper bound in (161) is smaller than 1 whenever $\rho>\gamma B_\phi L_\phi/\sqrt{2}$. Moreover,

$$\left\|\boldsymbol{d}_{k,i}\right\|=\left\|\boldsymbol{r}_{k,i}\phi(\boldsymbol{\mu}_{k,i})\right\|\le R_{\max}B_\phi.\tag{162}$$

As a result, if $\rho\ge 0.75\gamma B_\phi L_\phi$, we get:

$$\begin{aligned}\|\mathcal{W}_{i+1}\|&\overset{(161)}{\le}(1-0.08\alpha\gamma B_\phi L_\phi)\,\|\mathcal{W}_i\|+\alpha\,\|\boldsymbol{d}_i\|\\&\overset{(162)}{\le}(1-0.08\alpha\gamma B_\phi L_\phi)\,\|\mathcal{W}_i\|+\alpha\sqrt{K}R_{\max}B_\phi.\end{aligned}\tag{163}$$

Iterating this recursion starting from $i=0$ results in

$$\begin{aligned}\|\mathcal{W}_{i+1}\|&\le\alpha\sqrt{K}R_{\max}B_\phi\sum_{j=1}^{i+1}(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1-j}\\&\qquad+(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\,\|\mathcal{W}_0\|\\&\le\frac{\sqrt{K}R_{\max}}{0.08\gamma L_\phi}+(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\,\|\mathcal{W}_0\|\\&=\frac{\sqrt{K}R_{\max}}{0.08\gamma L_\phi}+o(1),\end{aligned}\tag{164}$$

where the last step holds whenever

$$\begin{aligned}&(1-0.08\alpha\gamma B_\phi L_\phi)^{i+1}\,\|\mathcal{W}_0\| = o(1)\\ &\quad\Longleftrightarrow i\log(1-0.08\alpha\gamma B_\phi L_\phi) = o(1)\\ &\quad\Longleftrightarrow i \geq \frac{o(1)}{\log(1-0.08\alpha\gamma B_\phi L_\phi)} \geq o\left(\frac{1}{\alpha\gamma B_\phi L_\phi}\right).\end{aligned}\tag{165}$$

ACKNOWLEDGMENT

The authors would like to thank Visa Koivunen for useful discussions on Dec-POMDPs and their applications in sensing.

REFERENCES

[1] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[2] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” in *Handbook Reinforcement Learn. Control*. Cham, Switzerland: Springer, 2021, pp. 321–384.

[3] J. W. Huang, Q. Zhu, V. Krishnamurthy, and T. Basar, “Distributed correlated Q-learning for dynamic transmission control of sensor networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 1982–1985.

[4] J. Lunden, S. R. Kulkarni, V. Koivunen, and H. V. Poor, “Multiagent reinforcement learning based spectrum sensing policies for cognitive radio networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 858–868, Oct. 2013.

[5] S. Bhattacharya, S. Kailas, S. Badyal, S. Gil, and D. Bertsekas, “Multiagent rollout and policy iteration for POMDP with application to multi-robot repair problems,” in *Proc. Conf. Robot Learn.*, 2021, pp. 1814–1828.

[6] O. Vinyals et al., “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[7] M. Samvelyan et al., “The starcraft multi-agent challenge,” in *Proc. Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 2186–2188.

[8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*. Cham, Switzerland: Springer, 2016.

[10] E. J. Sondik, “The optimal control of partially observable markov processes over the infinite horizon: Discounted costs,” *Operations Res.*, vol. 26, no. 2, pp. 282–304, 1978.

[11] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artif. Intell.*, vol. 101, no. 1, pp. 99–134, 1998.

[12] V. Krishnamurthy, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*.Cambridge, U.K.: Cambridge Univ. Press, 2016.

[13] J. Hazla, A. Jadbabaie, E. Mossel, and M. A. Rahimian, “Bayesian decision making in groups is hard,” *Operations Res.*, vol. 69, no. 2, pp. 632–654, 2021.

[14] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, “Deep decentralized multi-task multi-agent reinforcement learning under partial observability,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2681–2690.

[15] J. K. Gupta, M. Egorov, and M. Kochenderfer, “Cooperative multi-agent control using deep reinforcement learning,” in *Proc. AAMAS*, 2017, pp. 66–83.

[16] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2974–2982.

[17] P. Moreno et al., “Neural belief states for partially observed domains,” in *Proc. NeurIPS Workshop Reinforcement Learn. Partial Observability*, 2018, pp. 1–5.

[18] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, “Temporal difference variational auto-encoder,” in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.

[19] A. H. Sayed, *Inference and Learning From Data*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2022.

[20] P. Moreno, E. Hughes, K. R. McKee, B. A. Pires, and T. Weber, “Neural recursive belief states in multi-agent reinforcement learning,” 2021, *arXiv:2102.02274*.

[21] D. Muglich, L. M. Zintgraf, C. A. S. De Witt, S. Whiteson, and J. Foerster, “Generalized beliefs for cooperative AI,” in *Proc. Int. Conf. Mach. Learn.*, 2022, vol. 162, pp. 16062–16082.

[22] W. Mao, K. Zhang, E. Miehling, and T. Basar, “Information state embedding in partially observable cooperative multi-agent reinforcement learning,” in *Proc. IEEE Conf. Decis.Control*, 2020, pp. 6124–6131.

[23] S. Kar, J. M. F. Moura, and H. V. Poor, “$\mathcal{QD}$-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1848–1862, Apr. 2013.

[24] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, “Fully decentralized multi-agent reinforcement learning with networked agents,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5872–5881.

[25] L. Cassano, K. Yuan, and A. H. Sayed, “Multiagent fully decentralized value function learning with linear convergence rates,” *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1497–1512, Apr. 2021.

[26] S. V. Macua, I. Davies, A. Tukiainen, and E. M. De Cote, “Fully distributed actor-critic architecture for multitask deep reinforcement learning,” *Knowl. Eng. Rev.*, vol. 36, pp. 1–30, 2021.

[27] X. Sha, J. Zhang, K. You, K. Zhang, and T. Başar, “Fully asynchronous policy evaluation in distributed reinforcement learning over networks,” *Automatica*, vol. 136, 2022, Art. no. 110092.

[28] Y. Lin, G. Qu, L. Huang, and A. Wierman, “Multi-agent reinforcement learning in stochastic networked systems,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 7825–7837.

[29] G. Wang, S. Lu, G. Giannakis, G. Tesauro, and J. Sun, “Decentralized TD tracking with linear function approximation and its finite-time analysis,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 13762–13772.

[30] J. Sun, G. Wang, G. B. Giannakis, Q. Yang, and Z. Yang, “Finite-time analysis of decentralized temporal-difference learning with linear function approximation,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 4485–4495.

[31] Q. Lin and Q. Ling, “Decentralized TD(0) with gradient tracking,” *IEEE Signal Process. Lett.*, vol. 28, no. 4, pp. 723–727, Apr. 2021.

[32] A. Mahajan and M. Mannan, “Decentralized stochastic control,” *Ann. Operations Res.*, vol. 241, no. 1–2, pp. 109–126, 2016.

[33] A. A. Malikopoulos, “On team decision problems with nonclassical information structures,” *IEEE Trans. Autom. Control*, early access, Jul. 29, 2022, doi: 10.1109/TAC.2022.3195126.

[34] S Yuksel, “Stochastic nestedness and the belief sharing information pattern,” *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2773–2786, Dec. 2009.

[35] A. Nayyar, A. Mahajan, and D. Teneketzis, “Decentralized stochastic control with partial history sharing: A common information approach,” *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1644–1658, Jul. 2013.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[37] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690,1997, doi: 10.1109/9.580874.

[38] S. P. Singh, T. Jaakkola, and M. I. Jordan, “Learning without state-estimation in partially observable markovian decision processes,” in *Proc. Mach. Learn.*, 1994, pp. 284–292.

[39] A. Rodriguez, R. Parr, and D. Koller, “Reinforcement learning using approximate belief states,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, vol. 12, pp. 1036–1042.

[40] H. Kimura, K. Miyazaki, and S. Kobayashi, “Reinforcement learning in POMDPs with function approximation,” in *Proc. Int. Conf. Mach. Learn.*, 1997, vol. 97, pp. 152–160.

[41] Q. Cai, Z. Yang, and Z. Wang, “Reinforcement learning from partial observation: Linear function approximation with provable sample efficiency,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 2485–2522.

[42] Y. Li, Y. Tang, R. Zhang, and N. Li, “Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach,” *IEEE Trans. Autom. Control*, vol. 67, no. 12, pp. 6429–6444, Dec. 2022.

[43] H. Wang, S. Lin, H. Jafarkhani, and J. Zhang, “Distributed Q-learning with state tracking for multi-agent networked control,” in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2021, pp. 1692–1694.

[44] A. Mahajan, N. C. Martins, M. C. Rotkowitz, and S. Yüksel, “Information structures in optimal decentralized control,” in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 1291–1306.

[45] N. Saldi and S. Yüksel, “Geometry of information structures, strategic measures and associated stochastic control topologies,” *Probability Surv.*, vol. 19, pp. 450–532, 2022.

[46] J. Arabneydi and A. Mahajan, “Reinforcement learning in decentralized stochastic control systems with partial history sharing,” in *Proc. IEEE Amer. Control Conf.*, 2015, pp. 5449–5456.

[47] A. Nayyar and D. Teneketzis, “Common knowledge and sequential team problems,” *IEEE Trans. Autom. Control*, vol. 64, no. 12, pp. 5108–5115, Dec. 2019.

[48] M. Kayaalp, V. Bordignon, S. Vlaski, and A. H. Sayed, “Hidden Markov modeling over graphs,” in *Proc. IEEE Data Sci. Learn. Workshop*, 2022, pp. 1–6.

[49] M. Kayaalp, V. Bordignon, S. Vlaski, V. Matta, and A. H. Sayed, “Distributed Bayesian learning of dynamic states,” 2022, *arXiv:2212.02565*.

[50] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, “Likelihood consensus and its application to distributed particle filtering,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, Aug. 2012.

[51] G. Battistelli and L. Chisci, “Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,” *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[52] S. Bandyopadhyay and S. Chung, “Distributed Bayesian filtering using logarithmic opinion pool for dynamic sensor networks,” *Automatica*, vol. 97, pp. 7–17, 2018.

[53] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Found. Trends Mach. Learn.*, vol. 7, no. 4–5, pp. 311–801, Jul. 2014.

[54] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[55] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[56] P. D. Lorenzo and G. Scutari, “Next: In-network nonconvex optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[57] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks—Part I: Transient analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3487–3517, Jun. 2015.

[58] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks—Part II: Performance analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3518–3548, Jun. 2015.

[59] M. Kayaalp, S. Vlaski, and A. H. Sayed, “Dif-MAML: Decentralized multi-agent meta-learning,” *IEEE Open J. Signal Process.*, vol. 3, no. 1, pp. 71–93, Jan. 2022.

[60] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, 2nd ed. Cambridge, U.K.:Cambridge Univ. Press, 2011.

[61] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[62] J. Z. Kolter and A. Y. Ng, “Regularization and feature selection in least-squares temporal difference learning,” in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 521–528.

[63] M. W. Hoffman, A. Lazaric, M. Ghavamzadeh, and R. Munos, “Regularized least squares temporal difference learning with nested ℓ_2 and ℓ_1 penalization,” in *Proc. Eur. Workshop Reinforcement Learn.*, 2011, pp. 102–114.

[64] J. Farebrother, M. C. Machado, and M. Bowling, “Generalization and regularization in DQN,” 2018, *arXiv:1810.00123*.

[65] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, “Quantifying generalization in reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 1282–1289.

[66] B. R. Kiran et al., “Deep reinforcement learning for autonomous driving: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[67] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, “Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences,” *Int. J. Robot. Res.*, vol. 41, no. 1, pp. 45–67, 2022.

[68] T. Li, H. Fan, J. García, and J. M. Corchado, “Second-order statistics analysis and comparison between arithmetic and geometric average fusion: Application to multi-sensor target tracking,” *Inf. Fusion*, vol. 51, pp. 233–243, 2019.

[69] M. Kayaalp, Y. Inan, E. Telatar, and A. H. Sayed, “On the arithmetic and geometric fusion of beliefs for distributed inference,” Apr. 2022, *arXiv:2204.13741*.

[70] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Nashua, NH, USA: Athena Sci., 2015.

[71] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar, “Bayesian learning in social networks,” *Rev. Econ. Stud.*, vol. 78, no. 4, pp. 1201–1236, 2011.

[72] M. R. Garey and D. S. Johnson, *Computers and Intractability*, vol. 174. San Francisco, CA, USA:Freeman, 1979.

[73] S. I. Resnick, *Adventures in Stochastic Processes*.Boston, MA, USA: Birkhäuser, 2002.

[74] E. Jorge, M. Kågebäck, F. D. Johansson, and E. Gustavsson, “Learning to play guess who? and inventing a grounded language as a consequence,” 2016, *arXiv:1611.03218*.

[75] Y. Polyanskiy and Y. Wu, “Strong data-processing inequalities for channels and Bayesian networks,” in *Convexity and Concentration*. New York, NY, USA: Springer, 2017, pp. 211–249.

[76] J. Bretagnolle and C. Huber, “Estimation des densités : Risque minimax,” in *Séminaire de Probabilités XII*, C. Dellacherie, P. A. Meyer, and M. Weil, Eds. Berlin, Heidelberg, Germany: Springer, 1978, pp. 342–363.

[77] S. U. Pillai, T. Suel, and S. Cha, “The Perron-Frobenius theorem: Some of its applications,” *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 62–75, Mar. 2005.

MERT KAYAALP (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronics engineering from Koc University, Istanbul, Turkey, in 2019. He is currently working toward the Ph.D. degree with the Swiss Federal Institute of Technology Lausanne, (EPFL), Lausanne, Switzerland. His research interests include inference and learning in multi-agent systems. He was the recipient of the bronze medal in International Physics Olympiad (IPHO) 2015, Mumbai, India.

FATIMA GHADIEH joined the Adaptive Systems Laboratory, Swiss Federal Institute of Technology Lausanne, (EPFL), Lausanne, Switzerland, as a Student Intern during summer 2022. She is currently working toward the final year B.Eng. degree in electrical and computer engineering with the American University of Beirut, Beirut, Lebanon. Her research interests include control systems, robotics, and multi-agent reinforcement learning.

ALI H. SAYED (Fellow, IEEE) is currently the Dean of Engineering with Swiss Federal Institute of Technology Lausanne, (EPFL), Lausanne, Switzerland, where he also leads the Adaptive Systems Laboratory (https://asl.epfl.ch). He has served before as distinguished Professor and Chairman of electrical engineering with University of California Los Angeles, Los Angeles, CA, USA. He is a Member of the US National Academy of Engineering and The World Academy of Sciences. He was the President of the IEEE Signal Processing Society in 2018 and 2019. His work has been recognized with several awards including the 2022 IEEE Fourier Award, 2020 IEEE Norbert Wiener Society Award, and several Best Paper Awards. He is a Fellow of EURASIP, and AAAS.