Attention-Based Recurrence for Multi-Agent Reinforcement Learning under Stochastic Partial Observability

Thomy Phan [1] [†] **Fabian Ritz** [2] **Philipp Altmann** [2] **Maximilian Zorn** [2] **Jonas Nüßlein** [2] **Michael Kölle** [2]
Thomas Gabor [2] **Claudia Linnhoff-Popien** [2]

Abstract

Stochastic partial observability poses a major challenge for decentralized coordination in multi-agent reinforcement learning but is largely neglected in state-of-the-art research due to a strong focus on state-based *centralized training for decentralized execution (CTDE)* and benchmarks that lack sufficient stochasticity like *StarCraft Multi-Agent Challenge (SMAC)*. In this paper, we propose *Attention-based Embeddings of Recurrence In multi-Agent Learning (AERIAL)* to approximate value functions under stochastic partial observability. AERIAL replaces the true state with a learned representation of multi-agent recurrence, considering more accurate information about decentralized agent decisions than state-based CTDE. We then introduce *MessySMAC*, a modified version of SMAC with stochastic observations and higher variance in initial states, to provide a more general and configurable benchmark regarding stochastic partial observability. We evaluate AERIAL in Dec-Tiger as well as in a variety of SMAC and MessySMAC maps, and compare the results with state-based CTDE. Furthermore, we evaluate the robustness of AERIAL and state-based CTDE against various stochasticity configurations in MessySMAC.

1. Introduction

A wide range of real-world applications like fleet management, industry 4.0, or communication networks can be formulated as *decentralized partially observable Markov decision process (Dec-POMDP)* representing a cooperative *multi-agent system (MAS)*, where multiple agents have to coordinate to achieve a common goal (Oliehoek & Amato, 2016). *Stochastic partial observability* poses a major challenge for decentralized coordination in Dec-POMDPs due to noisy sensors and potentially high variance in initial states which are common in the real world (Kaelbling et al., 1998; Oliehoek & Amato, 2016).

Multi-agent reinforcement learning (MARL) is a general approach to tackle Dec-POMDPs with remarkable progress in recent years (Wang et al., 2021; Wen et al., 2022). State-of-the-art MARL is based on *centralized training for decentralized execution (CTDE)*, where training takes place in a laboratory or a simulator with access to global information (Lowe et al., 2017; Foerster et al., 2018). For example, *state-based CTDE* exploits true state information to learn a centralized value function in order to derive coordinated policies for decentralized decision making (Rashid et al., 2018; Yu et al., 2022). Due to its effectiveness in the *StarCraft Multi-Agent Challenge (SMAC)* as the current de facto standard for MARL evaluation, state-based CTDE has become very popular and is widely considered an adequate approach to general Dec-POMDPs for more than half a decade, leading to the development of many increasingly complex algorithms (Lyu et al., 2021; 2022).

However, merely relying on state-based CTDE and SMAC in MARL research can be a pitfall in practice as stochastic partial observability is largely neglected – despite being an important aspect in Dec-POMDPs (Lyu et al., 2022):

From an *algorithm perspective*, purely state-based value functions are insufficient to evaluate and adapt multi-agent behavior, since all agents make decisions on a completely different basis, i.e., individual histories of noisy observations and actions. True Dec-POMDP value functions consider more accurate closed-loop information about decentralized agent decisions though (Oliehoek et al., 2008). Furthermore, the optimal state-based value function represents an upper-bound of the true optimal Dec-POMDP value function thus state-based CTDE can result in overly optimistic behavior in general Dec-POMDPs (Lyu et al., 2022).

From a *benchmark perspective*, SMAC has very limited

stochastic partial observability due to deterministic observations and low variance in initial states (Ellis et al., 2022). Therefore, SMAC scenarios only represent simplified special cases rather than general Dec-POMDP challenges, being insufficient for assessing practicability of MARL.

In this paper, we propose *Attention-based Embeddings of Recurrence In multi-Agent Learning (AERIAL)* to approximate value functions under agent-wise stochastic partial observability. AERIAL replaces the true state with a learned representation of multi-agent recurrence, considering more accurate closed-loop information about decentralized agent decisions than state-based CTDE. We then introduce *MessySMAC*, a modified version of SMAC with stochastic observations and higher variance in initial states, to provide a more general and configurable Dec-POMDP benchmark for more adequate evaluation. Our contributions are as follows:

- We formulate and discuss the concepts of AERIAL w.r.t. stochastic partial observability in Dec-POMDPs.

- We introduce MessySMAC to enable systematic evaluation under various stochasticity configurations.

- We evaluate AERIAL in Dec-Tiger, a small and traditional Dec-POMDP benchmark, as well as in a variety of original SMAC and MessySMAC maps, and compare the results with state-based CTDE. Our results show that AERIAL achieves competitive performance in original SMAC, and superior performance in Dec-Tiger and MessySMAC. Furthermore, we evaluate the robustness of AERIAL and state-based CTDE against various stochasticity configurations in MessySMAC.

2. Background

2.1. Decentralized POMDPs

We formulate cooperative MAS problems as *Dec-POMDP* $M = \langle \mathcal{D}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \Omega, b_0 \rangle$, where $\mathcal{D} = \{1, ..., N\}$ is a set of agents i, $\mathcal{S}$ is a set of (true) states s_t at time step t, $\mathcal{A} = \langle \mathcal{A}_i \rangle_{i \in \mathcal{D}}$ is the set of joint actions $\mathbf{a_t} = \langle a_{t,1}, ..., a_{t,N} \rangle = \langle a_{t,i} \rangle_{i \in \mathcal{D}}$, $\mathcal{T}(s_{t+1}|s_t, \mathbf{a_t})$ is the state transition probability, $r_t = \mathcal{R}(s_t, \mathbf{a_t}) \in \mathbb{R}$ is the shared reward, $\mathcal{Z}$ is a set of local observations $z_{t,i}$ for each agent $i \in \mathcal{D}$, $\Omega(\mathbf{z_{t+1}}|\mathbf{a_t}, s_{t+1})$ is the probability of joint observation $\mathbf{z_{t+1}} = \langle z_{t+1,i} \rangle_{i \in \mathcal{D}} \in \mathcal{Z}^N$, and b_0 is the probability distribution over initial states s_0 (Oliehoek & Amato, 2016). Each agent i maintains a *local history* $\tau_{t,i} \in (\mathcal{Z} \times \mathcal{A}_i)^t$ and $\tau_\mathbf{t} = \langle \tau_{t,i} \rangle_{i \in \mathcal{D}}$ is the *joint history*. A *belief state* $b(s_t|\tau_\mathbf{t})$ is a sufficient statistic for joint history $\tau_\mathbf{t}$ and defines a probability distribution over true states s_t, updatable by Bayes' theorem (Kaelbling et al., 1998). Joint quantities are written in bold face.

Stochastic partial observability in M is given by *observation* and *initialization stochasticity* w.r.t. Ω and b_0 respectively.

A *joint policy* $\boldsymbol{\pi} = \langle \pi_i \rangle_{i \in \mathcal{D}}$ with *decentralized* or *local policies* π_i defines a deterministic mapping from joint histories to joint actions $\boldsymbol{\pi}(\tau_\mathbf{t}) = \langle \pi_i(\tau_{t,i}) \rangle_{i \in \mathcal{D}} \in \mathcal{A}$. The *return* is defined by $G_t = \sum_{c=0}^{T-1} \gamma^c r_{t+c}$, where T is the *horizon* and $\gamma \in [0, 1]$ is the *discount factor*. $\boldsymbol{\pi}$ can be evaluated with a *value function* $Q^{\boldsymbol{\pi}}(\tau_\mathbf{t}, \mathbf{a_t}) = \mathbb{E}_{b_0, \mathcal{T}, \Omega}[G_t|\tau_\mathbf{t}, \mathbf{a_t}, \boldsymbol{\pi}]$. The goal is to find an *optimal joint policy* $\boldsymbol{\pi}^*$ with *optimal value function* $Q^{\boldsymbol{\pi}^*} = Q^*$ as defined in the next section.

2.2. Optimal Value Functions and Policies

Fully Observable MAS   In MDP-like settings with a centralized controller, the optimal value function Q^*_{MDP} is defined by (Watkins & Dayan, 1992; Boutilier, 1996):

$$Q^*_{MDP}(s_t, \mathbf{a_t}) = r_t + \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{X} \qquad (1)$$

where $\mathcal{X} = \mathcal{T}(s_{t+1}|s_t, \mathbf{a_t}) max_{\mathbf{a_{t+1}} \in \mathcal{A}} Q^*_{MDP}(s_{t+1}, \mathbf{a_{t+1}})$.

Due to full observability, Q^*_{MDP} does not depend on $\tau_\mathbf{t}$ but on s_t. Thus, decentralized observations $z_{t,i}$ and probabilities according to Ω and b_0 are not considered at all. An optimal (joint) policy $\boldsymbol{\pi}^*_\mathbf{MDP}$ of the centralized controller simply maximizes Q^*_{MDP} for all s_t (Watkins & Dayan, 1992):

$$\boldsymbol{\pi}^*_\mathbf{MDP} = argmax_{\boldsymbol{\pi}_\mathbf{MDP}} \sum_{s_t \in \mathcal{S}} Q^*_{MDP}(s_t, \boldsymbol{\pi}_\mathbf{MDP}(s_t)) \qquad (2)$$

Partially Observable MAS   In general Dec-POMDPs, where true states are not fully observable and only decentralized controllers or agents exist, the optimal value function Q^* is defined by (Oliehoek et al., 2008):

$$Q^*(\tau_\mathbf{t}, \mathbf{a_t}) = \sum_{s_t \in \mathcal{S}} b(s_t|\tau_\mathbf{t}) \left(r_t + \gamma \sum_{s_{t+1} \in \mathcal{S}} \sum_{\mathbf{z_{t+1}} \in \mathcal{Z}^N} \mathcal{X} \right) \qquad (3)$$

where $\mathcal{X} = \mathcal{T}(s_{t+1}|s_t, \mathbf{a_t})\Omega(\mathbf{z_{t+1}}|\mathbf{a_t}, s_{t+1})Q^*(\tau_\mathbf{t+1}, \boldsymbol{\pi}^*(\tau_\mathbf{t+1}))$ with $\tau_\mathbf{t+1} = \langle \tau_\mathbf{t}, \mathbf{a_t}, \mathbf{z_{t+1}} \rangle$.

An optimal joint policy $\boldsymbol{\pi}^*$ for decentralized execution maximizes the expectation of Q^* for all joint histories $\tau_\mathbf{t}$ (Emery-Montemerlo et al., 2004; Oliehoek et al., 2008):

$$\boldsymbol{\pi}^* = argmax_{\boldsymbol{\pi}} \sum_{t=0}^{T-1} \sum_{\tau_\mathbf{t} \in (\mathcal{Z}^N \times \mathcal{A})^t} \mathcal{C}^{\boldsymbol{\pi}}(\tau_\mathbf{t})\mathbf{P}^{\boldsymbol{\pi}}(\tau_\mathbf{t}|b_0)Q^*(\cdot) \qquad (4)$$

where $Q^*(\cdot) = Q^*(\tau_\mathbf{t}, \boldsymbol{\pi}(\tau_\mathbf{t}))$, indicator $\mathcal{C}^{\boldsymbol{\pi}}(\tau_\mathbf{t})$ filters out joint histories $\tau_\mathbf{t}$ that are inconsistent with $\boldsymbol{\pi}$, and probability $\mathbf{P}^{\boldsymbol{\pi}}(\tau_\mathbf{t}|b_0)$ represents the *recurrence* of all agents considering agent-wise stochastic partial observability w.r.t.

decentralization of $\boldsymbol{\pi}$ and $\boldsymbol{\tau_t}$ (Oliehoek et al., 2008):

$$\begin{aligned} \mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0) &= \mathbf{P}(\mathbf{z_0}|b_0)\prod_{c=1}^{t}\mathbf{P}(\mathbf{z_c}|\boldsymbol{\tau_{c-1}},\boldsymbol{\pi}) \\ &= \mathbf{P}(\mathbf{z_0}|b_0)\prod_{c=1}^{t}\sum_{s_c\in\mathcal{S}}\sum_{s_{c-1}\in\mathcal{S}}\mathcal{T}(\cdot)\Omega(\cdot) \end{aligned} \quad (5)$$

where $\mathcal{T}(\cdot) = \mathcal{T}(s_c|s_{c-1},\boldsymbol{\pi}(\boldsymbol{\tau_{c-1}}))$ and $\Omega(\cdot) = \Omega(\mathbf{z_c}|\boldsymbol{\pi}(\boldsymbol{\tau_{c-1}}),s_c)$.

Since all agents act according to their local history $\tau_{t,i}$ without access to the complete joint history $\boldsymbol{\tau_t}$, recurrence $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ depends on more accurate *closed-loop information* than just true states s_t, i.e., all previous observations, actions, and probabilities according to b_0, $\mathcal{T}$, and Ω.

Q_{MDP}^* is proven to represent an upper bound of Q^* (Oliehoek et al., 2008). Thus, naively deriving local policies π_i from Q_{MDP}^* instead of Q^* can result in overly optimistic behavior as we will show in Section 4.1 and 6.

2.3. Multi-Agent Reinforcement Learning

Finding an optimal joint policy $\boldsymbol{\pi}^*$ via exhaustive computation of Q^* according to Eq. 3-5 is intractable in practice (Nair et al., 2003; Szer et al., 2005). MARL offers a scalable way to learn Q^* and $\boldsymbol{\pi}^*$ via function approximation, e.g., using CTDE, where training takes place in a laboratory or a simulator with access to global information (Lowe et al., 2017; Foerster et al., 2018). We focus on value-based MARL to learn a centralized value function $Q_{tot} \approx Q^*$, which can be factorized into *local utility functions* $\langle Q_i \rangle_{i\in\mathcal{D}}$ for decentralized decision making via $\pi_i(\tau_{t,i}) = argmax_{a_{t,i}} Q_i(\tau_{t,i},a_{t,i})$. For that, a *factorization operator* Ψ is used (Phan et al., 2021):

$$Q_{tot}(\boldsymbol{\tau_t},\mathbf{a_t}) = \Psi(Q_1(\tau_{t,1},a_{t,1}),...,Q_N(\tau_{t,N},a_{t,N})) \quad (6)$$

In practice, Ψ is realized with deep neural networks, such that $\langle Q_i \rangle_{i\in\mathcal{D}}$ can be learned end-to-end via backpropagation by minimizing the mean squared *temporal difference (TD)* error (Rashid et al., 2018; Sunehag et al., 2018). A factorization operator Ψ is *decentralizable* when satisfying the *IGM (Individual-Global-Max)* such that (Son et al., 2019):

$$argmax_{\mathbf{a_t}}Q_{tot}(\boldsymbol{\tau_t},\mathbf{a_t}) = \begin{pmatrix} argmax_{a_{t,1}}Q_1(\tau_{t,1},a_{t,1}) \\ \vdots \\ argmax_{a_{t,N}}Q_N(\tau_{t,N},a_{t,N}) \end{pmatrix} \quad (7)$$

There exists a variety of factorization operators Ψ which satisfy Eq. 7 using monotonicity like QMIX (Rashid et al., 2018), nonlinear transformation like QPLEX (Wang et al., 2021), or loss weighting like CW- and OW-QMIX (Rashid et al., 2020). Most approaches use state-based CTDE to learn Q_{MDP}^* according to Eq. 1 instead of Q^* (Eq. 3-5).

2.4. Recurrent Reinforcement Learning

In partially observable settings, the policy π_i of agent i conditions on the history $\tau_{t,i}$ of past observations and actions (Kaelbling et al., 1998; Oliehoek & Amato, 2016). In practice, *recurrent neural networks (RNNs)* like LSTMs or GRUs are used to learn a compact representation $h_{t,i}$ of $\tau_{t,i}$ and π_i known as *hidden state* or *memory representation*[1], which implicitly encodes the *individual recurrence* of agent i, i.e., the distribution $P_i^{\pi_i}$ over $\tau_{t,i}$ (Hochreiter & Schmidhuber, 1997; Cho et al., 2014; Hu & Foerster, 2019):

$$P_i^{\pi_i}(\tau_{t,i}|b_0) = P_i(z_{0,i}|b_0)\prod_{c=1}^{t}P_i(z_{c,i}|\tau_{c-1,i},\pi_i) \quad (8)$$

RNNs are commonly used for partially observable problems and have been empirically shown to be more effective than using raw observations $z_{t,i}$ or histories $\tau_{t,i}$ (Hausknecht & Stone, 2015; Samvelyan et al., 2019; Vinyals et al., 2019).

3. Related Work

Multi-Agent Reinforcement Learning In recent years, MARL has achieved remarkable progress in challenging domains (Gupta et al., 2017; Vinyals et al., 2019). State-of-the-art MARL is based on CTDE to learn a centralized value function Q_{tot} for actor-critic learning (Lowe et al., 2017; Foerster et al., 2018; Yu et al., 2022) or factorization (Rashid et al., 2018; 2020; Wang et al., 2021). However, the majority of works assumes a simplified Dec-POMDP setting, where Ω is deterministic, and uses true states to approximate Q_{MDP}^* according to Eq. 1 instead of Q^* (Eq. 3-5). Thus, state-based CTDE is possibly less effective in more general Dec-POMDP settings. Our approach addresses stochastic partial observability with a *learned representation* of multi-agent recurrence $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5 instead of s_t.

Weaknesses of State-Based CTDE Recent works investigated potential weaknesses of state-based CTDE for multi-agent actor-critic methods regarding bias and variance (Lyu et al., 2021; 2022). The experimental results show that state-based CTDE can surprisingly fail in very simple Dec-POMDP benchmarks that exhibit more stochasticity than SMAC. While these studies can be considered an important step towards general Dec-POMDPs, there is neither an approach which adequately addresses stochastic partial observability nor a benchmark to systematically evaluate such an approach yet. In this work, we focus on *value-based MARL*, where learning an accurate value function is important for meaningful factorization, and propose an *attention-based recurrence approach* to approximate value functions under stochastic partial observability. We also introduce

---

[1]In this paper, we use the term *memory representation* to avoid confusion with the state terminology of the (Dec-)POMDP literature (Kaelbling et al., 1998; Oliehoek & Amato, 2016).

a *modified* SMAC benchmark, which enables systematic evaluation under various stochasticity configurations.

Attention-Based CTDE Attention has been used in CTDE to process information of potentially variable length N, where joint observations $\mathbf{z_t}$, joint actions $\mathbf{a_t}$, or local utilities $\langle Q_i \rangle_{i \in \mathcal{D}}$ are weighted and aggregated to provide a meaningful representation for value function approximation (Iqbal & Sha, 2019; Wang et al., 2021; Iqbal et al., 2021; Wen et al., 2022; Khan et al., 2022). Most works focus on Markov games without observation stochasticity, which are special cases of the Dec-POMDP setting. In this work, we focus on *stochastic partial observability* and apply *self-attention* to the *memory representations* $h_{t,i}$ of all agents' RNNs instead of the raw observations $z_{t,i}$ to approximate Q^* for *general Dec-POMDPs* according to Eq. 3-5.

4. AERIAL

4.1. Limitation of State-Based CTDE

Most state-of-the-art works assume a simplified Dec-POMDP setting, where Ω is deterministic, and approximate Q^*_{MDP} according to Eq. 1 instead of Q^* (Eq. 3-5).

If there are only deterministic observations and initial states s_0 such that $b_0(s_0) = 1$ and $b_0(s') = 0$ if $s' \neq s_0$, then multi-agent recurrence $\mathbf{P^\pi}(\boldsymbol{\tau_t}|b_0)$ as defined in Eq. 5 would only depend on state transition probabilities $\mathcal{T}(s_{t+1}|s_t, \mathbf{a_t})$ which are purely state-based, ignoring decentralization of agents and observations (Oliehoek et al., 2008). In such scenarios, stochastic partial observability is very limited, especially if all π_i are deterministic. We hypothesize that this is one reason for the empirical success of state-based CTDE in original SMAC, whose scenarios seemingly have these simplifying properties (Ellis et al., 2022).

In the following, we regard a small example, where state-based CTDE can fail at finding an optimal joint policy $\boldsymbol{\pi}^*$.

Example *Dec-Tiger* is a traditional and simple Dec-POMDP benchmark with $N = 2$ agents facing two doors (Nair et al., 2003). A tiger is randomly placed behind the left (s_L) or right door (s_R) representing the true state. Both agents are able to listen (*li*) and open the left (o_L) or right door (o_R). The listening action *li* produces a noisy observation of either hearing the tiger to be left (z_L) or right (z_R), which correctly indicates the tiger's position with 85% chance and a cost of -1 per listening agent. If both agents open the same door, the episode terminates with a reward of -50 if opening the tiger door and +20 otherwise. If both agents open different doors, the episode ends with -100 reward and, if only one agent opens a door while the other agent is listening, the episode terminates with -101 if opening the tiger door and +9 otherwise.

Given a horizon of $T = 2$, the tiger being behind the right door (s_R), and both agents having listened in the first step, where agent 1 heard z_L and agent 2 heard z_R: Assuming that both agents learned to perform the same actions, e.g., due to CTDE and parameter sharing (Tan, 1993; Gupta et al., 2017), Q^*_{MDP} and Q^* would estimate the following values[2]:

$$Q^*_{MDP}(s_R, \langle li, li \rangle) = -2 \qquad {\color{blue}Q^*(\boldsymbol{\tau_t}, \langle li, li \rangle) = -2}$$
$${\color{blue}Q^*_{MDP}(s_R, \langle o_L, o_L \rangle) = 20} \qquad Q^*(\boldsymbol{\tau_t}, \langle o_L, o_L \rangle) = -15$$
$$Q^*_{MDP}(s_R, \langle o_R, o_R \rangle) = -50 \quad Q^*(\boldsymbol{\tau_t}, \langle o_R, o_R \rangle) = -15$$

Any policy $\boldsymbol{\pi}^*_{\mathbf{MDP}}$ or decentralizable joint policy $\boldsymbol{\pi}$ w.r.t. IGM (Eq. 7) that maximizes Q^*_{MDP} according to Eq. 2 would optimistically recommend $\langle o_L, o_L \rangle$ based on the true state s_R, regardless of what the agents observed. However, any joint policy $\boldsymbol{\pi}^*$ that maximizes the expectation of Q^* according to Eq. 4 would consider agent-wise stochastic partial observability and recommend $\langle li, li \rangle$, which corresponds to the true optimal decision for $T = 2$ (Szer et al., 2005).

4.2. Attention-Based Embeddings of Recurrence

Preliminaries We now introduce *Attention-based Embeddings of Recurrence In multi-Agent Learning (AERIAL)* to approximate optimal Dec-POMDP value functions Q^* according to Eq. 3-5. Our setup uses a factorization operator Ψ like QMIX or QPLEX according to Eq. 6-7. All agents process their local histories $\tau_{t,i}$ via RNNs as motivated in Section 2.4 and schematically shown in Fig. 1 (left).

Unlike Q^*_{MDP}, the true optimal Dec-POMDP value function Q^* considers more accurate closed-loop information about decentralized agent decisions through multi-agent recurrence $\mathbf{P^\pi}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5. Simply replacing s_t with $\boldsymbol{\tau_t}$ as suggested in (Lyu et al., 2022) is not sufficient because the resulting value function would assume a centralized controller with access to the complete joint history $\boldsymbol{\tau_t}$, in contrast to decentralized agents i which can only access their respective local history $\tau_{t,i}$ (Oliehoek et al., 2008).

Exploiting Multi-Agent Recurrence At first we propose to naively exploit all individual recurrences by simply replacing the true state s_t in CTDE with the *joint memory representation* $\mathbf{h_t} = \langle h_{t,i} \rangle_{i \in \mathcal{D}}$ of all agents' RNNs. Each memory representation $h_{t,i}$ implicitly encodes the individual recurrence $P^{\pi_i}_i(\tau_{t,i}|b_0)$ of agent i according to Eq. 8. Therefore, $\mathbf{h_t}$ provides more accurate closed-loop information about decentralized agent decisions than s_t.

This approach, called `AERIAL (no attention)`, can already be considered a sufficient solution if all individual recurrences $P^{\pi_i}_i(\tau_{t,i}|b_0)$ are statistically independent such that $\mathbf{P^\pi}(\boldsymbol{\tau_t}|b_0) = \prod_{i=1}^{N} P^{\pi_i}_i(\tau_{t,i}|b_0)$.

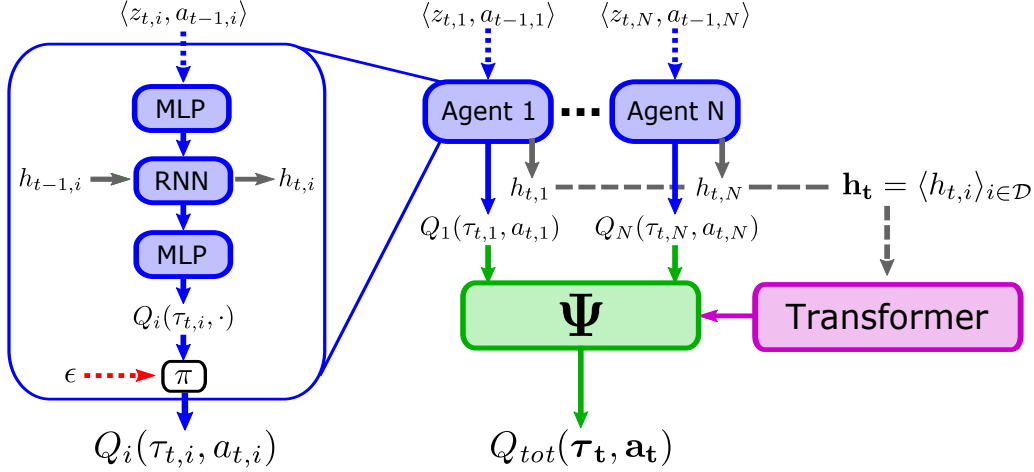[2]The exact calculation is provided in the Appendix B.

Figure 1. Illustration of the AERIAL setup. *Left:* Recurrent agent network structure with memory representations $h_{t-1,i}$ and $h_{t,i}$. *Right:* Value function factorization via factorization operator Ψ using the joint memory representation $\mathbf{h_t} = \langle h_{t,i} \rangle_{i \in \mathcal{D}}$ of all agents' RNNs instead of true states s_t. All memory representations $h_{t,i}$ are detached from the computation graph to avoid additional differentiation (indicated by the dashed gray arrows) and passed through a simplified transformer before being used by Ψ for value function factorization.

Attention-Based Recurrence While AERIAL (no attention) offers a simple way to address agent-wise stochastic partial observability, the independence assumption of all individual recurrences $P_i^{\pi_i}(\tau_{t,i}|b_0)$ does not hold in practice due to correlations in observations and actions (Bernstein et al., 2005; Amato et al., 2007).

Given the Dec-Tiger example above, the individual recurrences according to Eq. 8 are $P_1^{\pi_1}(\tau_{t,1}|b_0) = P_2^{\pi_2}(\tau_{t,2}|b_0) = 0.5$ (Kaelbling et al., 1998). However, the actual multi-agent recurrence according to Eq. 5 is $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0) = 0.15 \cdot 0.85 \neq P_1^{\pi_1}(\tau_{t,1}|b_0) \cdot P_2^{\pi_2}(\tau_{t,2}|b_0)$, indicating that individual recurrences are not statistically independent in general (Oliehoek & Amato, 2016).

Therefore, we process $\mathbf{h_t}$ by a simplified *transformer* along the agent axis to automatically consider the latent dependencies of all memory representations $h_{t,i} \in \mathbf{h_t}$ through self-attention. The resulting approach, called AERIAL, is depicted in Fig. 1 and Algorithm 1 in Appendix C.

Our transformer does not use positional encoding or masking, since we assume no particular ordering among agents. The joint memory representation $\mathbf{h_t}$ is passed through a single *multi-head attention* layer with the output of each attention head c being defined by (Vaswani et al., 2017):

$$att_c(\mathbf{h_t}) = softmax\left(\frac{W_q^c(\mathbf{h_t})W_k^c(\mathbf{h_t})^\top}{\sqrt{d_{att}}}\right) W_v^c(\mathbf{h_t}) \quad (9)$$

where W_q^c, W_k^c, and W_v^c are *multi-layer perceptrons (MLP)* with an output dimensionality of d_{att}. All outputs $att_c(\mathbf{h_t})$ are summed and passed through a series of MLP layers before being fed into the factorization operator Ψ, effectively

replacing the true state s_t by a learned representation of multi-agent recurrence $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5.

To avoid additional differentation of $\mathbf{h_t}$ through Ψ or Eq. 9, we detach $\mathbf{h_t}$ from the computation graph. Thus, we make sure that $\mathbf{h_t}$ is only learned through agent RNNs.

4.3. Discussion of AERIAL

The strong focus on state-based CTDE in the last few years has led to the development of increasingly complex algorithms that largely neglect stochastic partial observability in general Dec-POMDPs (Lyu et al., 2021; 2022). In contrast, AERIAL offers a simple way to adjust factorization approaches by replacing the true state s_t with a learned representation of multi-agent recurrence $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ to consider more accurate closed-loop information about decentralized agent decisions. The rest of the training scheme remains unchanged, which eases adjustment of existing approaches.

Since the naive independence assumption of individual memory representations $h_{t,i}$ does not hold in practice – despite decentralization – we use a simplified transformer to consider the latent dependencies of all $h_{t,i} \in \mathbf{h_t}$ along the agent axis to learn an adequate representation of multi-agent recurrence $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5.

AERIAL does not depend on true states therefore requiring less overall information than state-based CTDE, since we assume $\mathbf{h_t}$ to be available in all CTDE setups anyway (Foerster et al., 2018; Rashid et al., 2020). Note that AERIAL does not necessarily require RNNs to obtain $\mathbf{h_t}$ as hidden layers of MLPs or decision transformers can be used to approximate $\mathbf{h_t}$ as well (Son et al., 2019; Chen et al., 2021).
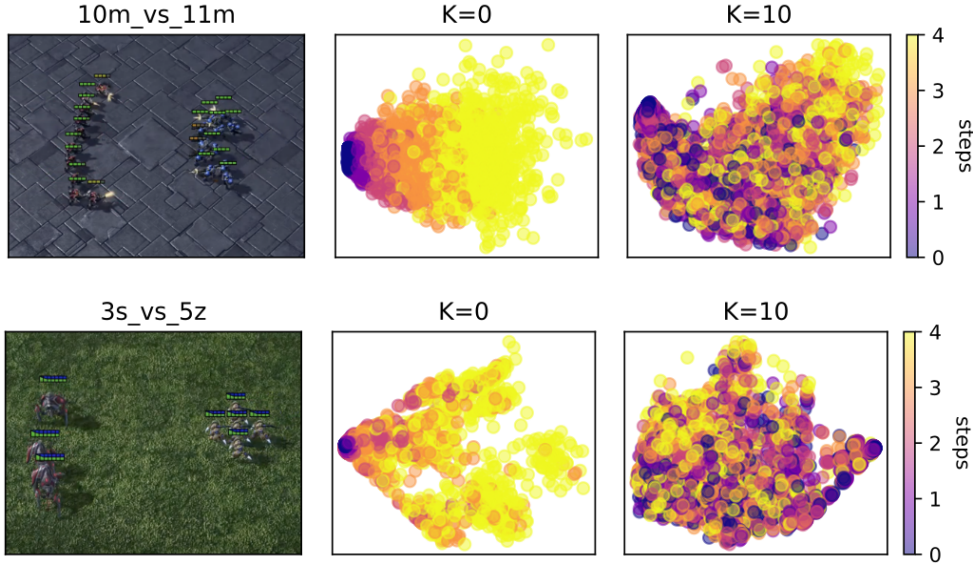
Figure 2. Left: Screenshot of two SMAC maps. *Middle:* PCA visualization of the joint observations in original SMAC within the first 5 steps of 1,000 episodes using a random policy with $K = 0$ initial random steps. *Right:* Analogous PCA visualization for MessySMAC with $K = 10$ initial random steps. For visual comparability, the observations are deterministic here.

5. MessySMAC

5.1. Limitation of SMAC as a Benchmark

StarCraft Multi-Agent Challenge (SMAC) provides a rich set of micromanagement tasks, where a team of learning agents has to fight against an enemy team, which acts according to handcrafted heuristics of the built-in StarCraft AI (Samvelyan et al., 2019). SMAC currently represents the de facto standard for MARL evaluation (Rashid et al., 2018; 2020; Wang et al., 2021). However, SMAC scenarios exhibit very limited stochastic partial observability due to deterministic observations and low variance in initial states therefore only representing simplified special cases rather than general Dec-POMDP challenges (Lyu et al., 2022; Ellis et al., 2022). To assess practicability of MARL, we need benchmarks with sufficient stochasticity as the real-world is generally messy and only observable through noisy sensors.

5.2. SMAC with Stochastic Partial Observability

MessySMAC is a modified version of SMAC with *observation stochasticity* w.r.t. Ω, where all measured values of observation $z_{t,i}$ are negated with a probability of $\phi \in [0, 1)$, and *initialization stochasticity* w.r.t. b_0, where K random steps are initially performed before officially starting an episode. During the initial phase, the agents can already be ambushed by the built-in AI, which further increases difficulty compared to the original SMAC maps if $K > 0$. MessySMAC represents a more general Dec-POMDP challenge which enables systematic evaluation under various stochasticity configurations according to ϕ and K.

Fig. 2 shows the PCA visualization of joint observations in two maps of original SMAC ($K = 0$) and MessySMAC ($K = 10$) within the first 5 steps of 1,000 episodes using a random policy. In original SMAC, the initial observations of s_0 (dark purple) are very similar and can be easily distinguished from subsequent observations by merely regarding time steps. Therefore, open-loop control might already be sufficient to solve these scenarios satisfactorily as hypothesized in (Ellis et al., 2022). However, the distinction of observations by time steps is more tricky in MessySMAC due to significantly higher entropy in b_0, indicating higher initialization stochasticity and a stronger requirement for closed-loop control, where agents need to explicitly consider their actual observations to make proper decisions.

5.3. Comparison with SMACv2

SMACv2 is an update to the original SMAC benchmark featuring initialization stochasticity w.r.t. position and unit types, as well as observation restrictions (Ellis et al., 2022). SMACv2 addresses similar issues as MessySMAC but MessySMAC additionally features *observation stochasticity* w.r.t. Ω according to the general Dec-POMDP formulation in Section 2.1. Unlike MessySMAC, SMACv2 does not support the *original SMAC maps* thus not enabling direct comparability w.r.t. stochasticity configurations.

Therefore, SMACv2 can be viewed as entirely new StarCraft II benchmark, while MessySMAC represents a *SMAC extension*, enabling systematic evaluation under various stochasticity configurations for the original SMAC maps.
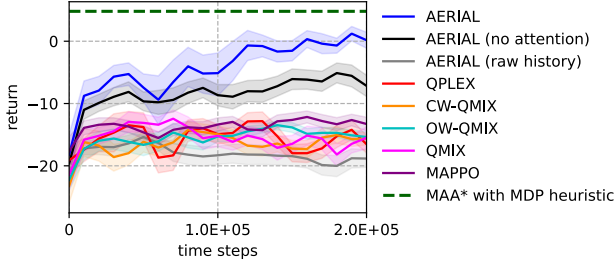
Figure 3. Average learning progress w.r.t. the return of AERIAL variants and state-of-the-art baselines in Dec-Tiger over 50 runs. Shaded areas show the 95% confidence interval.

6. Experiments

We use the state-based CTDE implementations of QPLEX, CW-QMIX, OW-QMIX, and QMIX from (Rashid et al., 2020) as state-of-the-art baselines with their default hyperparameters. We also integrate MAPPO from (Yu et al., 2022). For all experiments, we report the average performance and the 95% confidence interval over at least 20 runs.

AERIAL is implemented[3] using QMIX as factorization operator Ψ according to Fig. 1. We also experimented with QPLEX as alternative with no significant difference in performance. Thus, we stick with QMIX for efficiency due to fewer trainable parameters. The transformer of AERIAL has 4 heads with W_q^c, W_k^c, and W_v^c each having one hidden layer of $d_{att} = 64$ units with ReLU activation. The subsequent MLP layers have 64 units with ReLU activation.

For ablation study, we implement AERIAL (no attention), which trains Ψ directly on $\mathbf{h_t}$ without self-attention as described in Section 4.2, and AERIAL (raw history), which trains Ψ on the raw joint history $\boldsymbol{\tau_t}$ concatenated with the true state s_t as originally proposed for actor-critic methods (Lyu et al., 2022).

6.1. Dec-Tiger

Setting We use the Dec-Tiger problem described in Section 4.1 and (Nair et al., 2003) as simple proof-of-concept domain with $T = 4$ and $\gamma = 1$. We also provide the optimal value of 4.8 computed with MAA* (Szer et al., 2005).

Results The results are shown in Fig. 3. AERIAL comes closest to the optimum, achieving an average return of about zero. AERIAL (no attention) performs second best with an average return of about -8, while all other approaches achieve an average return of about -15.

Discussion The results confirm the example from Section 4.1 and the findings of (Oliehoek et al., 2008; Lyu et al.,

---

[3]Code is available at https://github.com/thomyphan/messy_smac. Further details are in Appendix D.

2022). All state-based CTDE approaches and AERIAL (raw history) converge to a one-step policy, where both agents optimistically open the same door regardless of any agent observation. AERIAL (no attention) converges to a local optimum most of the time, where both agents only listen for all $T = 4$ time steps. AERIAL performs best due to considering the latent dependencies of all memory representations $h_{t,i} \in \mathbf{h_t}$ via self-attention to learn an adequate representation of multi-agent recurrence $\mathbf{P^\pi}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5.

6.2. Original SMAC

Setting We evaluate AERIAL in original SMAC using the maps 3s5z and 10m_vs_11m, which are classified as *easy*, as well as the *hard* maps 2c_vs_64zg, 3s_vs_5z, and 5m_vs_6m, and the *super hard* map 3s5z_vs_3s6z (Samvelyan et al., 2019).

Results The final average test win rates after 2 million steps of training are shown in Table 1. AERIAL is competitive to QPLEX and QMIX in the easy maps, while performing best in 3s_vs_5z and 5m_vs_6m. MAPPO performs best in 2c_vs_64zg and 3s5z_vs_3s6z with AERIAL being second best in the super hard map 3s5z_vs_3s6z.

Discussion AERIAL is competitive to state-of-the-art baselines in original SMAC, indicating that replacing the true state s_t with the joint memory representation $\mathbf{h_t}$ does not notably harm performance. Despite outperforming most baselines in some maps, we do not claim significant outperformance here, since we regard most SMAC maps as widely solved by the community anyway (Ellis et al., 2022).

6.3. MessySMAC

Setting We evaluate AERIAL in MessySMAC using the same maps as in Section 6.2. We set $\phi = 15\%$ and $K = 10$.

Results The results are shown in Fig. 4. AERIAL performs best in all maps with AERIAL (no attention) being second best except in 2c_vs_64zg. In 3s5z_vs_3s6z, only AERIAL and AERIAL (no attention) progress notably. AERIAL (raw history) performs worst in all maps. MAPPO only progresses notably in 2c_vs_64zg.

Discussion Similar to the Dec-Tiger experiment, the results confirm the benefit of exploiting more accurate closed-loop information in domains with stochastic partial observability. AERIAL consistently outperforms AERIAL (no attention), indicating that self-attention can correct for the naive independence assumption of all $h_{t,i} \in \mathbf{h_t}$. MAPPO performs especially poorly in MessySMAC due to

Table 1. Average win rate of `AERIAL` and state-of-the-art baselines after 2 million time steps of training across 400 final test episodes for the original SMAC maps with the 95% confidence interval. The best results per map are highlighted in boldface and blue.

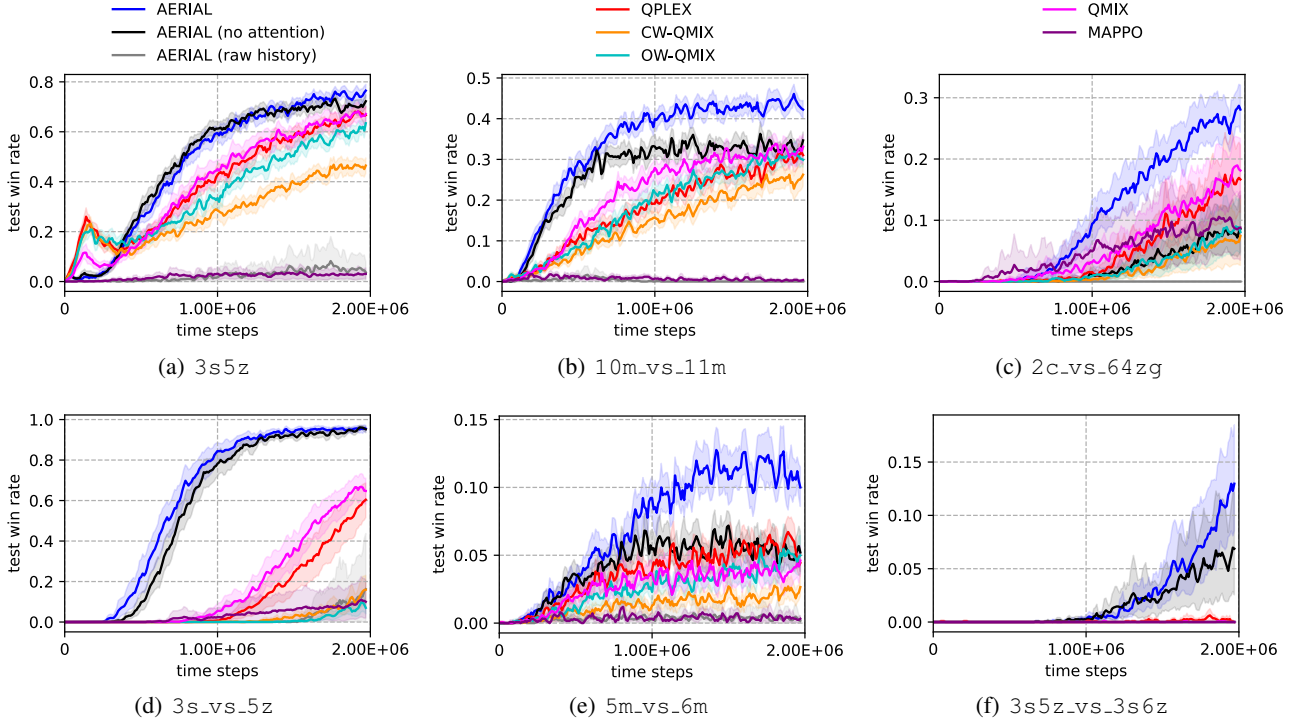|	AERIAL	QPLEX	CW-QMIX	OW-QMIX	QMIX	MAPPO
3s5z	**0.95 ± 0.01**	0.94 ± 0.01	0.87 ± 0.02	0.91 ± 0.02	**0.95 ± 0.01**	68.7 ± 0.94
10m_vs_11m	**0.97 ± 0.01**	0.90 ± 0.02	0.91 ± 0.02	0.96 ± 0.01	0.90 ± 0.02	77.3 ± 0.66
2c_vs_64zg	0.52 ± 0.11	0.29 ± 0.1	0.38 ± 0.12	0.55 ± 0.13	0.59 ± 0.11	**90.2 ± 0.24**
3s_vs_5z	**0.96 ± 0.02**	0.74 ± 0.11	0.18 ± 0.06	0.08 ± 0.04	0.81 ± 0.05	73.8 ± 0.44
5m_vs_6m	**0.77 ± 0.03**	0.66 ± 0.04	0.41 ± 0.04	0.55 ± 0.06	0.67 ± 0.05	60.6 ± 1.13
3s5z_vs_3s6z	0.18 ± 0.09	0.1 ± 0.03	0.0 ± 0.0	0.02 ± 0.01	0.02 ± 0.02	**20.5 ± 2.91**

Figure 4. Average learning progress w.r.t. the win rate of `AERIAL` variants and state-of-the-art baselines in MessySMAC for 2 million steps over 20 runs. Shaded areas show the 95% confidence interval. The legend at the top applies across all plots.

its misleading dependence on true states without any credit assignment, confirming the findings of (Ellis et al., 2022).

6.4. Robustness against Stochastic Partial Observability

Setting To evaluate the robustness of `AERIAL` and `AERIAL (no attention)` against various stochasticity configurations in MessySMAC, we manipulate Ω through the observation negation probability ϕ and b_0 through the number of initial random steps K as defined in Section 5.2. We compare the results with `QMIX` and `QPLEX` as the best performing state-of-the-art baselines in MessyS-MAC according to the results in Section 6.3. We present summarized plots, where the results are aggregated accross all maps from Section 6.3. To avoid that easy maps dominate the average win rate, since all approaches achieve high values there, we normalize the values by the maximum win rate

achieved in the respective map for all tested configurations of ϕ and K. Thus, we ensure an equal weighting regardless of the particular difficulty level. If not mentioned otherwise, we set $\phi = 15\%$ and $K = 10$ as default parameters based on Section 6.3.

Results The results regarding observation stochasticity w.r.t. Ω and ϕ are shown in Fig. 5. Fig. 5(a) shows that the average win rates of all approaches decrease with increasing ϕ with `AERIAL` consistently achieving the highest average win rate in all configurations. Fig. 5(b) shows that `AERIAL` performs best in most MessySMAC maps, especially when $\phi \geq 15\%$. `AERIAL (no attention)` performs second best.

The results regarding initialization stochasticity w.r.t. b_0 and K are shown in Fig. 6. Analogously to Fig. 5, Fig. 6(a) shows that the average (normalized) win rates of all ap-
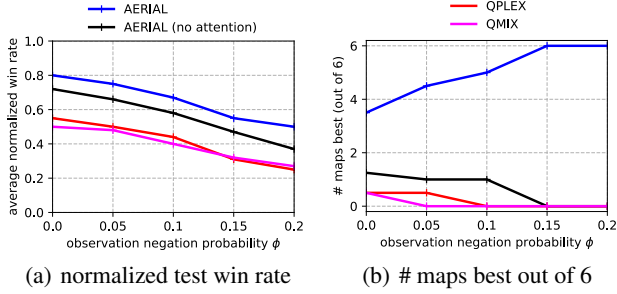
(a) normalized test win rate   (b) # maps best out of 6

Figure 5. Evaluation of `AERIAL`, `AERIAL (no attention)`, and the best MessySMAC baselines for different observation negation probabilities ϕ affecting observation stochasticity w.r.t. Ω (20 runs per configuration). (a) The average normalized test win rate across all 6 MessySMAC maps from Section 6.3. (b) The number of maps best out of 6. The legend at the top applies across all plots.
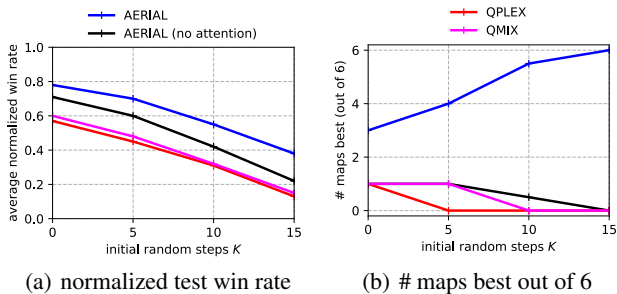

(a) normalized test win rate   (b) # maps best out of 6

Figure 6. Evaluation of `AERIAL`, `AERIAL (no attention)`, and the best MessySMAC baselines for different initial random steps K affecting initialization stochasticity w.r.t. b_0 (20 runs per configuration). (a) The average normalized test win rate accross all 6 MessySMAC maps from Section 6.3. (b) The number of maps best out of 6. The legend at the top applies across all plots.

proaches decrease with increasing K with `AERIAL` consistently achieving the highest average win rate in all configurations. Fig. 6(b) shows that `AERIAL` performs best in most MessySMAC maps, especially when $K \geq 10$. `AERIAL (no attention)` performs second best.

Discussion Our results systematically demonstrate the robustness of `AERIAL` and `AERIAL (no attention)` against various stochasticity configurations according to Ω and b_0. State-based CTDE is notably less effective in settings, where observation and initialization stochasticity is high. As `AERIAL` consistently performs best in all maps when $\phi \geq 15\%$ or $K \geq 10$, we conclude that providing an adequate representation of $\mathbf{P}^{\boldsymbol{\pi}}(\boldsymbol{\tau_t}|b_0)$ according to Eq. 5 that is learned, e.g., through $\mathbf{h_t}$ and self-attention, is more beneficial for CTDE than merely relying on true states when facing domains with high stochastic partial observability.

7. Conclusion and Future Work

To tackle general multi-agent problems, which are messy and only observable through noisy sensors, we need adequate algorithms and benchmarks that sufficiently consider stochastic partial observability.

In this paper, we proposed AERIAL to approximate value functions under stochastic partial observability with a learned representation of multi-agent recurrence, considering more accurate closed-loop information about decentralized agent decisions than state-based CTDE.

We then introduced *MessySMAC*, a modified version of SMAC with stochastic observations and higher variance in initial states, to provide a more general and configurable Dec-POMDP benchmark regarding stochastic partial observability. We showed visually in Fig. 2 and experimentally in Section 6 that MessySMAC scenarios pose a greater challenge than their original SMAC counterparts due to observation and initialization stochasticity.

Compared to state-based CTDE, AERIAL offers a simple but effective approach to general Dec-POMDPs, being competitive in original SMAC and superior in Dec-Tiger and MessySMAC, which both exhibit observation and initialization stochasticity unlike original SMAC. Simply replacing the true state with memory representations can already improve performance in most scenarios, confirming the need for more accurate closed-loop information about decentralized agent decisions. Self-attention can correct for the naive independence assumption of agent-wise recurrence to further improve performance, especially when observation or initialization stochasticity is high.

We plan to further evaluate AERIAL in SMACv2 and mixed competitive-cooperative settings with multiple CTDE instances (Lowe et al., 2017; Phan et al., 2020).

Acknowledgements

References

Amato, C., Bernstein, D. S., and Zilberstein, S. Optimizing Memory-Bounded Controllers for Decentralized POMDPs. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pp. 1–8, 2007.

Bernstein, D. S., Hansen, E. A., and Zilberstein, S. Bounded Policy Iteration for Decentralized POMDPs. In *IJCAI*, pp. 52–57, 2005.

Boutilier, C. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pp. 195–210. Morgan Kaufmann Publishers Inc., 1996.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision Transformer: Reinforcement Learning via Sequence Modeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097. Curran Associates, Inc., 2021.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, 2014.

Ellis, B., Moalla, S., Samvelyan, M., Sun, M., Mahajan, A., Foerster, J. N., and Whiteson, S. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. 2022.

Emery-Montemerlo, R., Gordon, G., Schneider, J., and Thrun, S. Approximate Solutions for Partially Observable Stochastic Games with Common Payoffs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04, pp. 136–143, USA, 2004. IEEE Computer Society. ISBN 1581138644.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual Multi-Agent Policy Gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

Gupta, J. K., Egorov, M., and Kochenderfer, M. Cooperative Multi-Agent Control using Deep Reinforcement Learning. In *Autonomous Agents and Multiagent Systems*, pp. 66–83. Springer, 2017.

Hausknecht, M. and Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. In *2015 AAAI Fall Symposium Series*, 2015.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Hu, H. and Foerster, J. N. Simplified Action Decoder for Deep Multi-Agent Reinforcement Learning. In *International Conference on Learning Representations*, 2019.

Iqbal, S. and Sha, F. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2961–2970, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Iqbal, S., De Witt, C. A. S., Peng, B., Boehmer, W., Whiteson, S., and Sha, F. Randomized Entity-wise Factorization for Multi-Agent Reinforcement Learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4596–4606. PMLR, 18–24 Jul 2021.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and Acting in Partially Observable Stochastic Domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Khan, M. J., Ahmed, S. H., and Sukthankar, G. Transformer-Based Value Function Decomposition for Cooperative Multi-Agent Reinforcement Learning in StarCraft. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 18(1):113–119, Oct. 2022.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Lyu, X., Xiao, Y., Daley, B., and Amato, C. Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pp. 844–852, 2021.

Lyu, X., Baisero, A., Xiao, Y., and Amato, C. A Deeper Understanding of State-Based Critics in Multi-Agent Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):9396–9404, Jun. 2022. doi: 10.1609/aaai.v36i9.21171.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D., and Marsella, S. Taming Decentralized POMDPs: Towards Efficient Policy Computation for Multiagent Settings. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 705–711, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

Oliehoek, F. A. and Amato, C. *A Concise Introduction to Decentralized POMDPs*, volume 1. Springer, 2016.

Oliehoek, F. A., Spaan, M. T., and Vlassis, N. Optimal and Approximate Q-Value Functions for Decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32: 289–353, 2008.

Phan, T., Gabor, T., Sedlmeier, A., Ritz, F., Kempter, B., Klein, C., Sauer, H., Schmid, R., Wieghardt, J., Zeller, M., et al. Learning and Testing Resilience in Cooperative

Multi-Agent Systems. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '20, pp. 1055–1063. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

Phan, T., Ritz, F., Belzner, L., Altmann, P., Gabor, T., and Linnhoff-Popien, C. VAST: Value Function Factorization with Variable Agent Sub-Teams. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24018–24032. Curran Associates, Inc., 2021.

Phan, T., Ritz, F., Nüßlein, J., Kölle, M., Gabor, T., and Linnhoff-Popien, C. Attention-Based Recurrency for Multi-Agent Reinforcement Learning under State Uncertainty. In *Extended Abstracts of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pp. 2839–2841. International Foundation for Autonomous Agents and Multiagent Systems, 2023. ISBN 9781450394321.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 10–15 Jul 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10199–10210. Curran Associates, Inc., 2020.

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '19, pp. 2186–2188, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5887–5896. PMLR, 09–15 Jun 2019.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo,

J. Z., Tuyls, K., and Graepel, T. Value-Decomposition Networks for Cooperative Multi-Agent Learning based on Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '18, pp. 2085–2087, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Szer, D., Charpillet, F., and Zilberstein, S. MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPs. UAI'05, pp. 576–583, Arlington, Virginia, USA, 2005. AUAI Press. ISBN 0974903914.

Tan, M. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, pp. 330–337, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1558603077.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is All You Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster Level in StarCraft II using Multi-Agent Reinforcement Learning. *Nature*, pp. 1–5, 2019.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*, 2021.

Watkins, C. J. and Dayan, P. Q-Learning. *Machine Learning*, 8(3-4):279–292, 1992.

Wen, M., Kuba, J. G., Lin, R., Zhang, W., Wen, Y., Wang, J., and Yang, Y. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. *arXiv preprint arXiv:2205.14953*, 2022.

Whittlestone, J., Arulkumaran, K., and Crosby, M. The Societal Implications of Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 70:1003–1030, May 2021. ISSN 1076-9757. doi: 10.1613/jair.1.12360.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *36th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

A. Limitations and Societal Impacts

A.1. Limitations

In easier domains without stochastic partial observability, AERIAL does not significantly outperform the state-of-the-art baselines as indicated by the original SMAC results in Table 1. This implies that simplified Dec-POMDP settings might benefit from more specialized algorithms. The dependence on joint memory representations $\mathbf{h_t} = \langle h_{t,i} \rangle_{i \in \mathcal{D}}$ might induce some bias w.r.t. agent behavior policies which could limit performance in hard exploration domains therefore requiring additional mechanisms beyond the scope of this work. The full version of AERIAL requires additional compute[4] due to the transformer component in Fig. 1, which can be compensated by using a more (parameter) efficient value function factorization operator Ψ, e.g., QMIX instead of QPLEX.

A.2. Potential Negative Societal Impacts

The goal of our work is to realize autonomous systems to solve complex tasks under stochastic partial observability as motivated in Section 1. We refer to (Whittlestone et al., 2021) for a general overview regarding societal implications of deep RL and completely focus on cooperative MARL settings in the following.

AERIAL is based on a centralized training regime to learn decentralized policies with a common objective. That objective might include bias of a central authority and could potentially harm opposing parties, e.g., via discrimination or misleading information. Since training is conducted in a laboratory or a simulator, the resulting system might exhibit unsafe or questionable behavior when being deployed in the real world due to poor generalization, e.g., leading to accidents or unfair decisions. The transformer component in Fig. 1 might require a significant amount of additional compute for tuning and training therefore increasing overall cost. The self-attention weights of Eq. 9 could be used to discriminate participating individuals in an unethical way, e.g., discarding less relevant groups of individuals according to the softmax output.

Similar to original SMAC, MessySMAC is based on team battles, indicating that any MARL algorithm mastering that challenge could be misused for real combat, e.g., in autonomous weapon systems to realize distributed and coordinated strategies. Since MessySMAC covers the aspect of stochastic partial observability, successfully evaluated algorithms could be potentially more effective and dangerous in real-world scenarios.

B. Dec-Tiger Example

Given the Dec-Tiger example from Section 4.1 with a horizon of $T = 2$, the tiger being behind the right door (s_R), and both agents having listened in the first step, where agent 1 heard z_L and agent 2 heard z_R: The final state-based values are defined by $Q^*_{MDP}(s_t, \mathbf{a_t}) = \mathcal{R}(s_t, \mathbf{a_t})$.

Due to both agents perceiving different observations, i.e., z_L and z_R respectively, the probability of being in state s_R is 50% according to the belief state, i.e., $b(s_R|\boldsymbol{\tau_t}) = b(s_L|\boldsymbol{\tau_t}) = \frac{1}{2}$. Thus, the true optimal Dec-POMDP values for the final time step are defined by:

$$
\begin{aligned}
Q^*(\boldsymbol{\tau_t}, \mathbf{a_t}) &= \sum_{s_t \in \mathcal{S}} b(s_t|\boldsymbol{\tau_t})\mathcal{R}(s_t, \mathbf{a_t}) \\
&= \frac{1}{2}(Q^*_{MDP}(s_L, \mathbf{a_t}) + Q^*_{MDP}(s_R, \mathbf{a_t}))
\end{aligned}
\tag{10}
$$

The values of Q^*_{MDP} and Q^* for the final time step $t = 2$ in the example are given in Table 2. Both agents can reduce the expected penalty when always performing the same action. Therefore, it is likely for MARL to converge to a joint policy that recommends the same actions for both agents, especially when synchronization techniques like parameter sharing are used (Tan, 1993; Gupta et al., 2017; Yu et al., 2022).

C. Full Algorithm of AERIAL

The complete formulation of AERIAL is given in Algorithm 1. Note that AERIAL does not depend on true states s_t at all, since the experience samples e_t used for training (Line 23) do not record any states.

---

[4]The additional amount regarding wall clock time was negligible in our experiments though.

Table 2. The values of Q^*_{MDP} and Q^* for the final time step $t = 2$ in the Dec-Tiger example from Section 4.1.

$\mathbf{a_t}$	$Q^*_{MDP}(s_L, \mathbf{a_t})$	$Q^*_{MDP}(s_R, \mathbf{a_t})$	$Q^*(\boldsymbol{\tau_t}, \mathbf{a_t})$
$\langle li, li \rangle$	-2	-2	-2
$\langle li, o_L \rangle$	-101	+9	-46
$\langle li, o_R \rangle$	+9	-101	-46
$\langle o_L, li \rangle$	-101	+9	-46
$\langle \mathbf{o_L, o_L} \rangle$	-50	$+20$	-15
$\langle o_L, o_R \rangle$	-100	-100	-100
$\langle o_R, li \rangle$	+9	-101	-46
$\langle o_R, o_L \rangle$	-100	-100	-100
$\langle \mathbf{o_R, o_R} \rangle$	$+20$	-50	-15

Algorithm 1 Attention-based Embeddings of Recurrence In multi-Agent Learning (AERIAL)

1: Initialize parameters for $\langle Q_i \rangle_{i \in \mathcal{D}}$ and Ψ.
2: **for** episode $m \leftarrow 1, E$ **do**
3:     Sample s_0, $\mathbf{z_0}$, and $\boldsymbol{\tau_0}$ via b_0 and Ω.
4:     **for** time step $t \leftarrow 0, T - 1$ **do**
5:         **for** agent $i \in \mathcal{D}$ **do**
6:             $a_{t,i} \leftarrow \pi_i(\tau_{t,i})$ {Use $argmax_{a_{t,i}} Q_i(\tau_{t,i}, a_{t,i})$}
7:             *rand* $\sim U(0, 1)$ {Sample from uniform distribution}
8:             **if** *rand* $\leq \epsilon$ **then**
9:                 Select random action $a_{t,i} \in \mathcal{A}_i$. {Explore with ϵ-greedy}
10:             **end if**
11:         **end for**
12:         $\mathbf{a_t} \leftarrow \langle a_{t,i} \rangle_{i \in \mathcal{D}}$
13:         Execute joint action $\mathbf{a_t}$.
14:         $s_{t+1} \sim \mathcal{T}(s_{t+1}|s_t, \mathbf{a_t})$
15:         $\mathbf{z_{t+1}} \sim \Omega(\mathbf{z_{t+1}}|\mathbf{a_t}, s_{t+1})$
16:         $\mathbf{h_t} \leftarrow \langle h_{t,i} \rangle_{i \in \mathcal{D}}$ {Query memory representations of all agents}
17:         Detach $\mathbf{h_t}$ from computation graph. {Avoid additional differentiation through Ψ or Eq. 9}
18:         $\boldsymbol{\tau_{t+1}} \leftarrow \langle \boldsymbol{\tau_t}, \mathbf{a_t}, \mathbf{z_{t+1}} \rangle$ {Concatenate $\boldsymbol{\tau_t}$, $\mathbf{a_t}$, and $\mathbf{z_{t+1}}$}
19:         **for** attention head $c \leftarrow 1, C$ **do**
20:             $attention_c \leftarrow att_c(\mathbf{h_t})$ {Process individual recurrences according to Eq. 9}
21:         **end for**
22:         $rec_t \leftarrow MLP(\sum_{c=1}^{C} attention_c)$ {See Section 4.2}
23:         $e_t \leftarrow \langle \boldsymbol{\tau_t}, \mathbf{a_t}, r_t, \mathbf{z_{t+1}}, rec_t \rangle$
24:         Store experience sample e_t.
25:     **end for**
26:     Train Ψ and $\langle Q_i \rangle_{i \in \mathcal{D}}$ using all e_t. {See Fig. 1}
27: **end for**

D. Experiment Details

D.1. Computing infrastructure

All training and test runs were performed in parallel on a computing cluster of fifteen x86_64 GNU/Linux (Ubuntu 18.04.5 LTS) machines with i7-8700 @ 3.2GHz CPU (8 cores) and 64 GB RAM. We did not use any GPU in our experiments.

D.2. Hyperparameters and Neural Network Architectures

Our experiments are based on PyMARL and the code from (Rashid et al., 2020) under the Apache License 2.0. We use the default setting from the paper without further hyperparameter tuning as well as the same neural network architectures for the agent RNNs, i.e., *gated recurrent units (GRU)* of (Cho et al., 2014) with 64 units, and the respective factorization operators Ψ as specified by default for each state-of-the-art baseline in Section 6. We set the loss weight $\alpha = 0.75$ for CW-QMIX and OW-QMIX.

For MAPPO, we use the hyperparameters suggested in (Yu et al., 2022) for SMAC, where we set the clipping parameter to 0.1 and use an epoch count of 5. The parameter λ for generalized advantage estimation is set to 1. The centralized critic has two hidden layers of 128 units with ReLU activation, a single linear output, and conditions on *agent-specific global states* which concatenate the true state s_t and the individual observation $z_{t,i}$ per agent i. The policy network of MAPPO has a similar recurrent architecture like the local utility functions Q_i and additionally applies softmax to the output layer.

AERIAL is implemented using QMIX as factorization operator Ψ according to Fig. 1. We also experimented with QPLEX as alternative with no significant difference in performance. Thus, we stick with QMIX for computational efficiency due to fewer trainable parameters. The transformer has $C = 4$ heads $c \in \{1, ..., C\}$ with respective MLPs W_q^c, W_k^c, and W_v^c, each having one hidden layer of $d_{att} = 64$ units with ReLU activation. The three subsequent MLP layers of Line 22 in Algorithm 1 have 64 units with ReLU activation.

All neural networks are trained using RMSProp with a learning rate of 0.0005.