Ad Hoc Teamwork in the Presence of Non-Stationary Teammates*

Pedro M. Santos[1,2], João G. Ribeiro[1,2], Alberto Sardinha[1,2], and Francisco S. Melo[1,2]

[1] INESC-ID, Lisbon, Portugal
[2] Instituto Superior Técnico, University of Lisbon, Portugal
{pedro.m.m.santos,joao.g.ribeiro,joao.g.ribeiro}@tecnico.ulisboa.pt, fmelo@inesc-id.pt

Abstract. In this paper we address the problem of ad hoc teamwork and contribute a novel approach, PPAS, that is able to handle non-stationary teammates. Current approaches to ad hoc teamwork assume that the (potentially unknown) teammates behave in a stationary way, which is a significant limitation in real world conditions, since humans and other intelligent systems do not necessarily follow strict policies. In our work we highlight the current limitations of state-of-the-art approaches to ad hoc teamwork problem in the presence of non-stationary teammate, and propose a novel solution that alleviates the stationarity assumption by combining ad hoc teamwork with adversarial online prediction. The proposed architecture is called PLASTIC Policy with Adversarial Selection, or PPAS. We showcase the effectiveness of our approach through an empirical evaluation in the half-field offense environment. Our results show that it is possible to cooperate in an ad hoc manner with non-stationary teammates in complex environments.

Keywords: Multi-agent Systems · Ad hoc Teamwork Problem · Reinforcement Learning

1 Introduction

Many works on cooperative multi-agent systems (MAS) traditionally assume that the agents have some communication protocol in place or that some other coordination strategy is defined a priori (or both). These assumptions can be a problem as different types of autonomous agents (such as electronic personal assistants and smart devices) become a ubiquitous reality in our daily lives.

* This work was partially supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 (INESC-ID multi-annual funding) and the HOTSPOT project, with reference PTDC/CCI-COM/7203/2020. In addition, this material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-19-1-0020, and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA N. 952215. JGR acknowledges the PhD grant 2020.05151.BD from FCT.

In many situations, these vastly different agents will have no communication or coordination protocols in place but will, nevertheless, need to cooperate effectively towards attaining some common goal (e.g., the comfort of the user). The challenge of developing autonomous agents that are capable of cooperating in a common task with unknown teammates, without explicit coordination or communication is known as the *ad hoc teamwork problem* [18].

The key challenge in ad hoc teamwork is to develop an agent (the "ad hoc agent") that is able to leverage acquired knowledge regarding the interaction with previous teammates to quickly adapt when paired with a new team. The ad hoc teamwork problem has been studied for several years in the MAS community [3, 5, 8, 12, 13, 15, 17]. State-of-the-art approaches, such as the PLASTIC algorithms [4], use of reinforcement learning (RL) and transfer learning techniques to successfully address ad hoc teamwork in complex domains such as *half-field offense* [9]. However, most aforementioned works assume that teammates follow stationary policies, which means that it is expected that teammates will always present the same behavior over the interaction, which is a significant limitation if ad hoc teamwork is to be extended to real world settings involving, for example, human teammates. Our work addresses the question "How can ad hoc agents successfully cooperate with non-stationary teammates in complex domains?". Our contributions are two-fold:

- We evaluate state-of-the-art algorithms—namely the PLASTIC algorithms—against non-stationary teammates. PLASTIC has only been evaluated against stationary teammates and our results show the impact that the presence of non-stationary teammates has in the performance of the method.
- We introduce an extension to PLASTIC Policy, dubbed *PLASTIC Policy with Adversarial selection*, or PPAS. This algorithm relies on the core architecture of PLASTIC Policy, but uses a teammate identification mechanism that relies on an adversarial online prediction algorithm. Such algorithm relies on milder assumptions on the process to be predicted (in our case, the teammate behavior) and is thus robust to non-stationary teammates.

We evaluate our proposed approach in half-field offense (HFO), showcasing its advantages in the presence of non-stationary teammates.

2 Related Work

Stone et al. [18] recognized the importance of having autonomous agents able to collaborate without prior coordination, which they introduced as the *ad hoc teamwork problem*. The ad hoc teamwork problem combines several different elements that set it apart from other multi-agent problems, namely: (i) the agents have no predefined coordination or communication mechanism in place; (ii) the team is not necessarily homogeneous—in particular, the ad hoc agent is often different from the other agents; (iii) the ad hoc agent should be able to leverage prior knowledge to quickly adapt to the teammates in an online manner.

A significant volume of work on ad hoc teamwork considers *stationary teammates*. In other words, the algorithms are built on the assumption that the teammates do not change their behavior throughout the interaction—for example as a

result of the ad hoc agent's actions. However, in many multi-agent problems, the assumption of stationary teammates is too restrictive. Hernandez-Leal et al. [10] propose a taxonomy for agent behaviors in multi-agent settings: non adapting, slowly adapting and drastically adapting agents. We now go over relevant work in ad hoc teamwork, organizing it along the aforementioned teammate categories.

Non-adapting teammates. These works assume that teammates follow a stationary strategy during the entire interaction. The ad hoc teamwork problem is addressed by classifying the teammates behaviour as belonging to some previouly acquired "behavior prototypes" [1, 2, 4, 12, 16]. If the prototypes are able to model the teammates' behavior correctly, then these methods can lead to fast and efficient teamwork in the absence of explicit prior coordination. This is, for example, the approach in the PLASTIC algorithms [4], which are general-purpose algorithms based on transfer learning and RL that reuse prior teammate knowledge to quickly adapt to new teammates.

Barrett et al. [4] presented both a model-based and a model-free version of PLASTIC. Both approaches were successfully in addressing the ad hoc teamwork problem. However, the model-based approach is significantly slower and had difficulty dealing with complex environments. On the other hand, the model-free version—PLASTIC Policy—was able to successfully handle complex environments and adapt fast to new teammates. In our work, we use the PLASTIC Policy architecture as a basis for our approach.

PLASTIC Policy, however, still presents some limitations. It assumes that there are similarities between the new and old teammates' behaviors, and it completely relies on finding the most suitable policy for the current team. This means that during the exploration phase the performance is low, which when dealing with critical tasks can be harmful. Also, it relies on the fact that the team follows one stationary policy, already known or very similar to past experiences. If this is not the case, the agent will keep changing between policies during the interaction, putting at risk the task. To tackle this problem, we use an adversarial approach for action selection and belief updates [13].

Slowly adapting teammates. These works assume that teammates adapt slowly—for example assuming that the changes in the teammates' strategy exhibits bounded variation between rounds [7,13]. Although these approaches are able to partially address non-stationary teammates, they are mostly model-based and unsuited for complex environments.

For example, Melo and Sardinha [13] proposed an online prediction approach, named *exponentially weighted forecaster for ad hoc teamwork*, able to deal with slowly adapting teammates. Their algorithm identifies the task being performed by the teammates and acts accordingly. It keeps a set of beliefs about which task is currently being performed, which are updated over time. Also, they use the prediction from "experts" to select the ad hoc agent's actions through an online prediction approach. However, their work is unable to address sequential tasks, focusing only on repeated one-shot tasks. In our work, we also adopt an online prediction approach, but use it for team identification rather than for task

identification. We combine this prediction algorithm with the PLASTIC architecture [4], which allows us to deal with unpredictable teammates in complex environments—namely, teammates that are non-stationary.

Drastically adapting teammates. There are also a few works that assume that teammates can change between policies in a drastic manner during the interaction [11, 15]. However, these algorithms are specialized to this particular setting and, for example, cannot cope with slow adaptation teammates. Additionally, they are computationally too heavy to handle complex environments such as half-field offense.

3 PLASTIC Policy with Adversarial Selection

In this section we introduce our main contribution—an algorithm for ad hoc teamwork that can handle non-stationary teammates. We dub our algorithm PLASTIC Policy with Adversarial selection, or PPAS.

Our approach extends the PLASTIC Policy architecture [4] to include an online prediction approach for teammate identification [13]—namely, the *exponentially weighted average forecaster* [6]. By combining the two, we are able to handle non-stationary teammates and deal with complex environments.

Much like the original PLASTIC Policy, our algorithm makes use of past experiences to identify, adapt and cooperate in an ad hoc manner with unknown teammates. However, in contrast with PLASTIC Policy, we do not select a single policy to follow from those previously learned, and instead use information from *all such policies*. This allows PPAS to make near-optimal predictions early in the interaction and still select good actions when facing non-stationary teammates.

3.1 Architecture

The architecture of the proposed approach can be seen in Figure 1 and comprises three major elements. The first element corresponds to the two blocks "Team Models" and "Team Policies". These blocks contain the prior knowledge that the agent acquired, for example, by interacting with previous teams. A second element is responsible for identifying the teammates, and is performed in the "Update Weights" block. A third and final element is responsible for the selection of the actions of the ad hoc agent, and corresponds to the "Policies Predictions" and "Action Selection" blocks together.

When faced with a new team, at each time step the ad hoc agent determines the similarity between the observed behavior of the current team and that observed in teams it previously met (stored as "Team Models"). Based on that similarity, the ad hoc agent combines the action prescribed by the "Team Policies" to determine an action to execute. The process then repeats at the next time step. In the continuation, we describe each of the above elements in detail.

Training the Team Policies and Team Models. The "Team Models" and "Team Policies" correspond to the agent's prior knowledge, acquired beforehand when

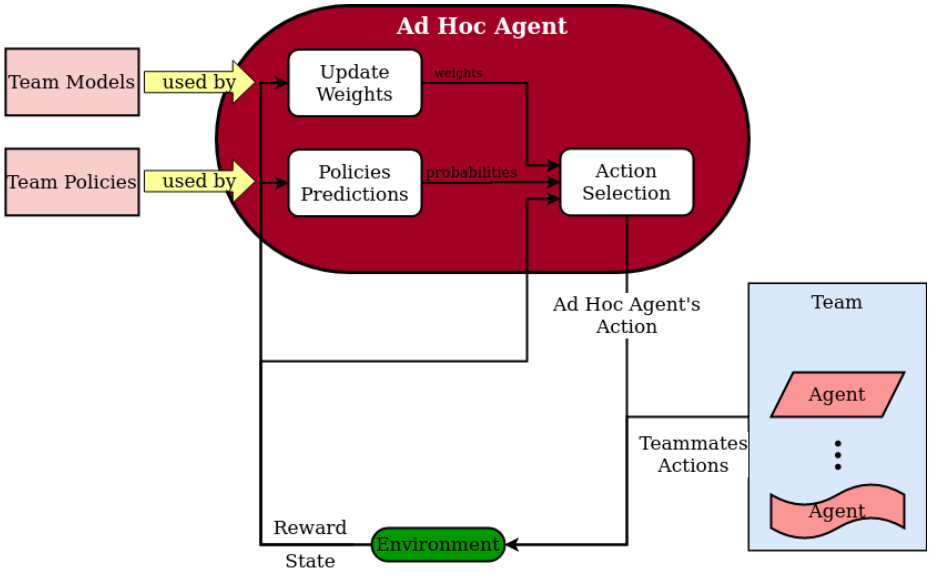


Fig. 1. Overview of PPAS. The architecture is adapted from PLASTIC Policy [4].

the ad hoc agent interacted with different teams. In our case, they were obtained by allowing the ad hoc agent to interact with several teams of stationary teammates for a fixed number of episodes, treating the teammates as part of the environment. During such interactions, "Team Policies" are trained using model-free reinforcement learning. When interacting with a particular team k, at each step t the agent experiences a transition $\langle x(t), a(t), r(t), x(t+1)\rangle$, where $x(t)$ is the state, $a(t)$ is the action of the ad hoc agent, $r(t)$ is the resulting reward, and $x(t+1)$ is the resulting state. For each team k, the agent collects N such transitions into a set $D_k = \{\langle x_n, a_n, r_n, x'_n\rangle, n = 1, \ldots, N\}$ that is then used to learn a policy using the well-established DQN algorithm [14].

In PPAS (much like in PLASTIC Policy), policies are represented using *Q-functions*. A Q-function assigns a real value, $Q(x, a)$, to each possible state-action pair (x, a). At any state x, the action prescribed by the policy encoded by Q is the action with the maximal Q-value. In DQN, a Q-function is represented as a neural network, and the parameters θ of the network are updated to minimize

$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N} \|r_n + \gamma \max_{a'} Q_{\theta^-}(x_{n+1}, a') - Q_\theta(x_n, a_n)\|^2,$$

where $Q_\theta(x, a)$ is the output of the network for the pair (x, a), γ is a scalar discount, and Q_{θ^-} is a copy of the network that is held fixed during most of the training process.[3] The ad hoc agent thus learns a function Q_k for each team $k = 1, \ldots, K$, and all such functions are collected in the "Team Policies".

[3] We refer to the work of Mnih et al. [14] for details on DQN.

Algorithm 1 PLASTIC Policy with Adversarial Selection (PPAS)

Initialize $t = 0$, $w_k(0) = 1$ for $k = 1, \ldots, K$.
for all t **do**
  for $k = 1, \ldots, K$ **do**
    Get forecast vector $\xi^k(t)$
  $W(t) = \sum_{k=1}^{K} w_k(t)$
  For each action a, compute $p_a(t)$ using (1).
  Select action $a(t) = \text{argmax}_a \; p_a(t)$
  Observe new environment state $x(t+1)$
  for $k = 1, \ldots, K$ **do**
    Predict next state $\hat{x}_k(t+1)$
    $d_k = \|x(t+1) - \hat{x}_k(t+1)\|_2$
    $w_k(t+1) \leftarrow w_k(t) \cdot e^{-\eta d_k}$

The "Team Models", on the other hand, consist of a collection of past experiences for each team, which are used to determine how similar the behavior of the current team is to that of the teams previously encountered.

Action selection. In PLASTIC Policy [4], the teammate identification is conducted by maintaining a belief over the set of "Team Models". The belief is updated using the similarity between the observed behavior of the current team and that in the teams in the library. The agent then selects—from the library of Team Policies—the action prescribed by the policy for the most likely team.

In PPAS we instead follow Melo and Sardinha [13] and use an online prediction algorithm to select the action to select at each time step, based on the action predictions of *all* the policies in the "Team Policies". PPAS maintains a weight w_k for each team k in the library of "Team Policies". As the agent interacts with its current team, it will query at each time step t each policy in "Team Policies". Such query returns, for each team k, a "forecast vector" $\xi^k(t)$ indicating the most likely actions in the current state $x(t)$. The exponentially weighted forecaster then computes a distribution $p(t)$ over actions by averaging the vectors $\xi_k(t), i = 1, \ldots, K$, where

$$p_a(t) = \frac{1}{W(t)} \sum_{k=1}^{K} w_k(t) \xi_a^k(t), \tag{1}$$

with $\xi_a^k(t)$ indicating the probability of action a according to $\xi^k(t)$ and $W(t) = \sum_k w_k(t)$. Given the distribution $p(t)$, the action selection is greedy, which means that the action with the highest probability is the one chosen.

Teammate identification. The teammate identification consists of determining which (if any) of the teams in the "Team Models" best matches the team that the ad hoc agent is currently facing. As seen above, PPAS maintains a weight w_k for each team in the library. The weights are initialized to 1, suggesting a

uniform "initial belief" over teams—before interacting with the current team, there is no reason to believe that any one team is more likely than the other.

To update these weights, the agent observes how the behavior of its teammates affects the environment. At each time step t, as the environment transitions to a new state, $x(t+1)$, the agent calculates the similarity between the transition $(x(t), x(t+1))$ with similar transitions stored in the "Team Models" for each of the teams. Given the predicted transition for team k, $(\hat{x}_k(t), \hat{x}_k(t+1))$, PPAS computes the Euclidean distance d_k between the actual next state, $x(t+1)$, and the "predicted" next state, $\hat{x}_k(t+1)$. The weights are then updated according to the exponential weighted forecaster update rule [6], yielding

$$w_k(t) \leftarrow w_k(t-1)e^{-\eta d_k}, \tag{2}$$

for a suitable constant $\eta > 0$. PPAS is summarized in Algorithm 1.

4 Experimental Evaluation

We now describe the experimental evaluation of our algorithm. We compare PPAS against the original PLASTIC Policy, which we henceforth abbreviately denote SPP (Standard PLASTIC Policy), illustrating the advantages of our approach in the presence of non-stationary teammates.

4.1 Experimental setup

Half-Field Offense. We evaluate our work in the HFO scenario, a complex environment that offers multiple challenges—a continuous multi-dimensional state space, real-time actions, noisy sensing and actions, and sparse rewards. In HFO there are two competing teams: the offense team and the defense team. Our agent belongs to the offense team, and the objective of our team is to score a goal (see Fig. 2 for a depiction of HFO). Both teams start without ball, and the game ends when either (1) The offense team scores goal; (2) The ball leaves the game area; (3) The defense team catches the ball; (4) The game exceeds the maximum number of steps allowed (500 steps).

NPC Agents. To create the Team Models and Policies, we used teams of agents created as part of the 2D RoboCup Simulation League competition. We use 5 teams from the 2013 competition as teammates: *aut*, *axiom*, *cyrus*, *gliders*, and *helios*. For the defense team, we use the HFO benchmark agents, the *agent2d*.

Environment model. In order to run the DQN part of PPAS, we must describe HFO as a Markov decision problem, identifying the states, actions, reward, and dynamics (i.e., how states evolve). We consider two variations of HFO: the *limited version*, where both defense and offense teams have two players; and the *full version*, where the defense team has 5 players and the attack team has 4 players.

- The state is described by 13 features in the limited version and 23 features in the full version. These features include positions, velocities, orientations of each agent, position and velocity of the ball.

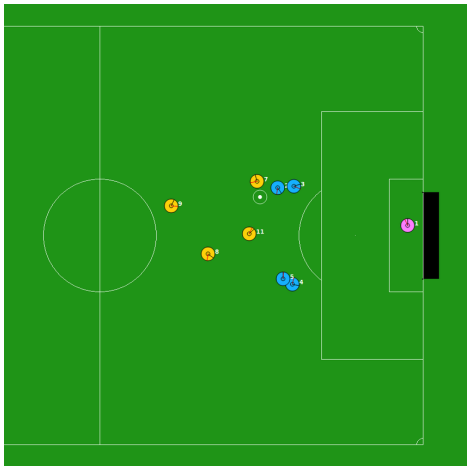


Fig. 2. Screenshot of the HFO environment. In HFO the attacking team (in yellow) tries to score against a defending team (in blue and pink).

- We adopt a similar action space of Barrett et al. [4], that includes a discretized set of actions (passes to the different teammates, running towards the ball, shooting with different power. In the limited version we consider 11 discrete actions and 13 discrete actions in the full version.
- We define the reward function as follows: a goal is worth a reward of 1000; the other termination conditions are worth a reward of -1000. All other steps correspond to a reward of -1.
- The dynamics are ruled by the HFO simulator. Since our approach is model free, there is no need to specify the dynamics explicitly.

Training Both PPAS and PLASTIC policy trained with the 5 aforementioned teams prior to the beginning of the experiment. Each ad hoc agent played each team for $100,000$ episodes, collecting the necessary data data. Each episode consisted of a full HFO game. The Team Policies were trained using DQN, as described in Section 3, while the Team Models used a combination of KD-Trees and arrays as a model for each team, storing the transitions experienced by the agent when playing that team.

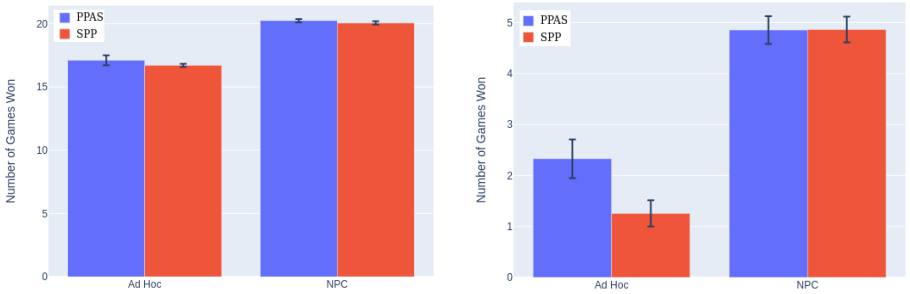
4.2 Results

The experiments were designed to answer the following questions: (a) When facing stationary opponents, is PPAS able to retain the state-of-the-art performance of PLASTIC Policy and (b) When facing non-stationary opponents, is PPAS able to outperform PLASTIC Policy, showcasing improved robustness?

To answer the two above questions we consider two distinct scenarios: in a first scenario, both algorithms are run against stationary teammates, corresponding to the teams already encountered during training: in each trial the

Table 1. Test scenario description.

Scenario	First	Second
Teammate Type	NPC Agents	Ad hoc agents
Teammate Policy	In each trial one of five teams is chosen randomly	Same algorithm as the ad hoc agent being tested
Teammate Behavior	Stationary	Non-Stationary



(a) Limited (2 vs 2) setting. Results are averaged over 1,000 trials.

(b) Full (4 vs 5) setting. Results are averaged over 100 trials.

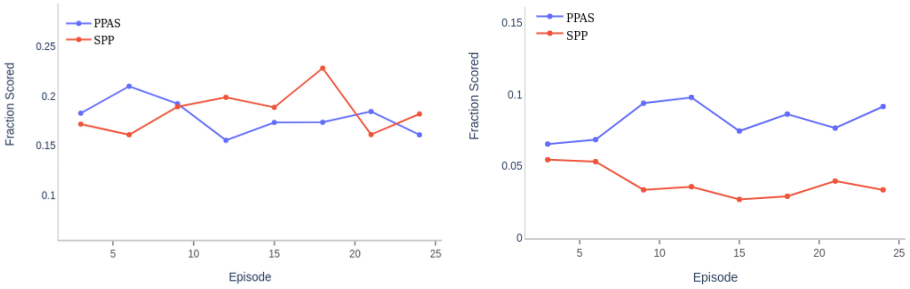
Fig. 3. Number of won games out of a total of 25 games in the limited and full settings.

agent is paired with a team randomly selected from the 5 aforementioned teams. In the second scenario, both algorithms are tested in self-play (i.e., against a team of similar ad hoc agents). Since these teammates are all adjusting their behavior simultaneously, they behave in a non-stationary manner. The different scenarios are summarized in Table 1. The results reported are averaged over a large number of trials (1,000 for the limiter version, and 100 for the full version), where a trial corresponds to 25 independent games.[4]

Limited Version (2 vs 2) We start by analyzing the performance of both ad hoc algorithms in the limited scenario (2 defenders vs 2 atackers). Figure 3(a) compares the performance of the two ad hoc agents in terms of the average number of goals scored (games won) per trial. In this simple setting, both agents attain a similar performance, both against stationary teammates (NPC) and non-stationary teammates (Ad Hoc). Although there is a slight improvement when using PPAS, this difference is not statistically significant.

This limited setting is somewhat deceiving: the fact that there are only two defenders makes it possible for a competent player to score by itself, rendering cooperation (and, thus, ad hoc teamwork) secondary. For this reason, we consider the full version, featuring 4 attackers against 5 defenders (see Fig. 2).

[4] Agents beliefs and teammate information is reset across trials.



(a) Stationary teammates. (b) Non-stationary teammates.

Fig. 4. Scoring frequency in the full HFO setting (4 vs 5) during 25 games, for the stationary and non-stationary teams. Results are averages over 100 independent trials.

Full Version (4 vs 5) In the full setting—where 4 attackers try to score against 5 defenders—cooperation plays a critical role. Since there are more defenders than attackers, it is very difficult for an attacker on its own to score. Therefore, this setting provides a much clearer assessment of the team's ability to act as a team and—consequently—of the performance of the two approaches in terms of their ability to establish ad hoc teamwork.

Figure 3(b) again compares the performance of the two ad hoc agents in terms of the average number of goals scored (games won) per trial. Several observations stand out. First, the overall performance is significantly lower than in the limited case—the number of goals scored hardly exceeds 5. This is in sharp contrast with the 20 goals scored in the limited setting.

A second observation is that the difference in performance between the stationary and non-stationary teams is larger than in the limited setting. This happens since the stationary teammates have a well-defined cooperation strategy to which the ad hoc agent adapts, while the non-stationary team does not.

Finally, the third observation is that, in the full setting, PPAS attains the same score as SPP against stationary teammates, but significantly outperforms SPP against non-stationary teammates, showcasing the ability of our approach to deal with non-stationary teammates.

To further understand the comparative performance of the two ad hoc algorithms, we plot, in Fig. 4, the amount of goals scored in each of the 25 games in a trial,[5] averaged across 100 independent trials. Once again, we can observe that against stationary teammates (Fig. 4(a)), the two algorithms perform similarly, and their performance remains approximately constant across the 25 games, even if SPP exhibits more fluctuations.

However, when paired against non-stationary teammates (Fig. 4(b)), the difference between the two approaches becomes apparent. On one hand, the performance of PPAS remains approximately constant throughout the 25 games. On the other hand, the performance of SPP—which starts in a value similar to

[5] For ease of visualization, the results were smoothed using a 3-step running window.

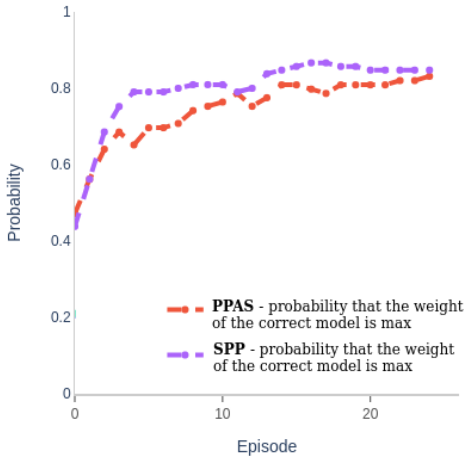


Fig. 5. Probability of the weight associated with the correct team policy being maximal when playing against stationary teammates in the full HFO setting. Results are averaged over 100 independent trials.

that of PPAS—steadily decreases as more games are played, suggesting that the SPP agents are unable to co-adapt.

To conclude our analysis, we depict in Fig. 5 the evolution of the probability that the correct team policy is assigned maximum weight. This is an indicator of the ability of the algorithms to identify the correct team early in the interaction. As can be seen, SPP is able to identify the correct team more quickly. However, because of the action selection mechanism in PPAS, this does not translate necessarily in a difference in performance (as seen in Fig. 3), since the action is selected based on the recommendation from *all* the teams.

Our results satisfactorily answer both our initial questions. PPAS is able to retain the state-of-the-art performance of PLASTIC Policy, whole outperforming PLASTIC Policy against non-stationary teammates. Our results also illustrate the strengths and weaknesses of both approaches. PPAS takes more time to identify the correct team, although it can select good actions even when uncertain about the team it is playing with. SPP is faster to identify the correct team, but is unable to handle non-stationary teammates.

5 Conclusions and Future Work

In this work, we proposed PPAS, an algorithm for ad hoc teamwork that is robust to non-stationary teammates. Our algorithm collects past experiences with different teams in the form of policy and team models. These models are then used when playing a new team through an online prediction algorithm. Even if the team is unknown and does not follow a stationary behavior, PPAS is able to select good actions and coordinate. We evaluated our algorithm in the half field offense environment, with different levels of difficulty, and illustrated the effectiveness and efficiency of our solution.

There are several interesting avenues for future research on ad hoc teamwork. For example, it would be interesting to augment our approach with parameterized agent types, instead of discrete agent types. Another interesting addition would be to investigate how to identify different levels of behavior, since teammates can display multiple behaviors.

References

1. Albrecht, S.V., Ramamoorthy, S.: A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In: AAMAS (2013)
2. Albrecht, S.V., Stone, P.: Reasoning about hypothetical agent behaviours and their parameters. In: AAMAS (2017)
3. Barrett, S., Stone, P.: Ad hoc teamwork modeled with multi-armed bandits: An extension to discounted infinite rewards. In: AAMAS ALA Workshop (2011)
4. Barrett, S., Rosenfeld, A., Kraus, S., Stone, P.: Making friends on the fly: Cooperating with new teammates. Artificial Intelligence **242**, 132–171 (2017)
5. Bowling, M., McCracken, P.: Coordination and adaptation in impromptu teams. In: AAAI (2005)
6. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games. Cambridge University Press (2006)
7. Chakraborty, D., Stone, P.: Cooperating with a Markovian ad hoc eeammate. In: AAMAS (2013)
8. Chen, S., Andrejczuk, E., Cao, Z., Zhang, J.: Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. In: AAAI (2020)
9. Hausknecht, M., Mupparaju, P., Subramanian, S., Kalyanakrishnan, S., Stone, P.: Half Field Offense: an environment for multiagent learning and ad hoc teamwork. In: AAMAS ALA Workshop (2016)
10. Hernandez-Leal, P., Kaisers, M., Baarslag, T., de Cote, E.M.: A survey of learning in multiagent environments: Dealing with non-stationarity. Computing Research Repository **abs/1707.09183** (2017)
11. Hernandez-Leal, P., Zhan, Y., Taylor, M.E., Sucar, L.E., de Cote, E.M.: Efficiently detecting switches against non-stationary opponents. Autonomous Agents and Multi-Agent Systems **31**(4), 767–789 (2017)
12. Macke, W., Mirsky, R., Stone, P.: Expected value of communication for planning in ad hoc teamwork. In: AAAI (2021)
13. Melo, F.S., Sardinha, A.: Ad hoc teamwork by learning teammates' task. Autonomous Agents and Multi-Agent Systems **30**(2), 175–219 (2016)
14. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)
15. Ravula, M., Alkoby, S., Stone, P.: Ad hoc teamwork with behavior switching agents. In: IJCAI (2019)
16. Rodrigues, G.: Ad Hoc Teamwork With Unknown Task Model and Teammate Behavior. Master's thesis, Instituto Superior Técnico (2018)
17. Stone, P.: Autonomous learning agents: Layered learning and ad hoc teamwork. In: AAMAS (2016)
18. Stone, P., Kaminka, G.A., Kraus, S., Rosenschein, J.S.: Ad hoc autonomous agent teams: Collaboration without pre-coordination. In: AAAI (2010)