The Utility and Implementation of Explicit Commands for Ad-Hoc Coordination

Timothy Flavin[1 [0000−0003−1601−6253]] and Sandip Sen[1 [0000−0001−6107−4095]]

University of Tulsa, Tulsa OK 74104, USA
Timmy-Flavin@utulsa.edu, Sandip-Sen@utulsa.edu
https://utulsa.edu/academics/engineering-computer-science/academics/departments/computer-science/

Abstract. A major concern of cooperative multi-agent reinforcement learning (MARL) in real-world applications is the ability to coordinate transparently with new teammates for a common goal. Two problem formulations, Zero-Shot Coordination (ZSC) and Ad-Hoc Teamplay (AHT) have garnered particular interest. We focus on AHT because it includes dynamic agents with no prior knowledge, while ZSC assumes a static policy based on shared knowledge of the environment dynamics. Communication is a common factor in both settings, but standard practice is to share grounded information through pre-defined communication channels or to learn arbitrary communication protocols that implicitly suggest actions or share information. We argue that explicit command-based communication allows for a higher theoretical team performance ceiling than information sharing or best response strategies, and that teacher-listener relationships can be learned in an ad-hoc setting for any pre-trained agent that can estimating the current value of an environment state. We show how to learn explicit commands in ad-hoc timescales through our algorithm, Multi-Armed Two-way Command Heuristic (MATCH). Finally, we provide a minimally complex environment to measure and isolate the effects of cooperative equilibrium selection, generalization to teammate skill variance. We leverage this environment to provide a principled approach for evaluating future coordination algorithms in terms of their ability to address cooperation challenges.

Keywords: Ad-Hoc Coordination · Multi-Armed Bandit · Reinforcement Learning · Communication

1 Introduction

In many real-world use cases such as volunteer search and rescue, military operations, autonomous vehicles, and online video games, proposed artificial agents must work with other agents that they have not seen before, and that they have no explicit control over [33]. A person might offer or follow advice, but in ad-hoc environments with no pre-determined reward structure, the shot-caller is a dynamic group decision which must be made with very limited information. Ongoing challenges for producing effective artificial agents in ad-hoc environments,

including robustness to novel conditions [18], trustworthiness for humans [14, 15, 12], and the ability to identify [25] and operate on complementary strategies [2, 32] with surrounding entities that share similar goals. For the purposes of this paper, we focus on high-quality joint-strategy selection and execution in environments where multiple optimal joint strategies exist, and other agents follow unknown strategies at unknown skill levels. In such an environment, there is an equilibrium selection problem [35, 29] over which optimal joint strategy to play, and there is also a consideration outside of equilibrium selection which we refer to as "the skill gap problem". The skill gap problem occurs when a rational agent must choose a best response to unskilled teammates that might be attempting to play complementary joint strategies, but that are incapable of executing them faithfully. No matter what the rational agent does unilaterally, the resulting team strategy may not be one of the optimal joint strategies of the environment, so we want to do better than best-response.

In order to isolate the effects of coordination and the skill gap, we introduce a new environment "Lever-NvNTTT" where two teams of 'N' agents play tic tac toe and all agents on a team must choose the same square to place a piece, or their turn is skipped. Lever-NvNTTT can be installed from pip as fasttttsandbox. Lever-NvNTTT is designed to be minimally complex and computationally inexpensive while unambiguously separating different aspects of the ad-hoc coordination challenge.

2 Preliminaries, Coordination in Multi-Agent RL

Before describing our methodology, there are two formalisms of the coordination problem that are of great interest to this paper. Zero Shot Coordination (ZSC) first introduced in [17] describes a problem where agents share common knowledge about the environment dynamics in which they operate, and a common goal, but they are not allowed to change their strategy once execution has begun. Recent works on ZSC [16, 9, 43, 45] focus on creating agents that abandon the arbitrary conventions learned through self-play in favor of grounded policies that generalize to other, rational, ZSC agents. These agents may share grounded information as in the game Hanabi [3]. Ad-hoc teamplay [33, 23, 39] describes the problem differently, as shared competence is not assumed of other agents, but changes in policy are allowed during execution. For the duration of this paper, we adopt the paradigm of N-agent Ad-hoc teamplay where agents are allowed to communicate and update their own policy at runtime, but they are not allowed to force control over another agent. Explicit commands in this paper are defined as an action or a sequence of actions that one agent sends to another agent which can be followed or ignored at the discretion of the listening agent. Future work will include commands at a higher level of abstraction as humans commonly communicate at the sub-task level for a given problem. Existing methods of communication in ad-hoc teamplay include predefined [24, 3], arbitrarily learned [11, 44] or even implicit communication such as action signaling [26, 1]. The use of communication channels to influence another agent's behavior does

not count as forced control over the other agent, because the response is still up to the listener. In the same way, we argue that optional explicit commands do not violate the ad-hoc requirement about forced control, but they do provide an opportunity for a transparent and grounded form of communication from an explainability point of view.

Formally, our agents can operate in fully or partially observable decentralized Markov Decision Process (Dec-POMDP) with communication [4, 46]. For partially observable environments, an agent implementing our protocol, MATCH, will need either an estimate of other agents' observations from which to generate action recommendations, or a policy which generates actions for multiple agents. For this paper, we use a fully observable environment as a best-case scenario to align with similarly best-case dynamics for the competing paradigms we present. The most general application of MATCH, a Dec-POMDP consists of the 7-tuple $\{S, \{U_i\}, T, R, \{\Omega_i\}, O, \gamma\}$ where S represents the set of all possible states of the environment, U_i represents the set of actions available to each agent $a_i : i \in \{1, ..., n\}$ of n total agents, where $\mathbf{U}$ represents the joint-action. The transition function $T = S \times \mathbf{U} \to \Delta S$ represents the probability of moving from state $s \in S$ to some new state $s' \in S$ given joint-action $\boldsymbol{u} = \langle u_1, u_2, ..., u_n \rangle \in \mathbf{U}$. The reward function $R_i : S \times \mathbf{U} \times S \to \mathbb{R}^n$ maps each state-action transition to a reward for each agent. We use $G_t^\gamma = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2}...$ to denote the discounted sum of rewards to-go where γ is the discount factor. $\mathbf{\Omega} = \{\Omega_i\}$ is the set of observations for each agent generated by the state observation function $O : S \times \mathbf{U} \to \Delta\mathbf{\Omega}$. For the environments in this paper, we assume full cooperation so that rewards are shared among agents $R_1 = R_2 = ... = R_n$. Under the Dec-POMDP framework, each agent has a policy $\pi_i : O_i \to u_i$ which maps that agent's observation to either an action u_i or a probability distribution over possible actions U_i in the case of a stochastic policy. The communication in our environments is assumed to be a cheap talk [8] setting where communications are non-binding and free of any direct cost or payoff to agents. Messages take place between each timestep of an environment.

Let $\mathit{\Pi}$ be the set of all possible policies that an individual agent can follow for an ad-hoc environment with $P_{\mathit{\Pi}}$ as a probability distribution over $\mathit{\Pi}$. The learning goal for typical Multi-Agent Reinforcement Learning (MARL) in a Dec-POMDP is to find a set of policies that maximizes expected return.

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in \mathit{\Pi}^n}{\arg\max} \mathbb{E}[G_0^\gamma | \boldsymbol{\pi} = \langle \pi_1, ..., \pi_n \rangle] \tag{1}$$

Under the centralized training, decentralized execution framework (CTDE), we have control over all $\pi_i \in \boldsymbol{\pi}$ at training, so it is sufficient to find any $\boldsymbol{\pi}^*$ (of which there may be multiple). The goal of ZSC and AHT, shown formally in equation 2, can be summarized as learning a policy $\hat{\pi}^*$ that maximizes team performance when the policies followed by other agents are sampled from $P_{\mathit{\Pi}}$.

$$\hat{\pi}^* = \underset{\hat{\pi} \in \mathit{\Pi}^1}{\arg\max} \mathbb{E}_{\pi_{i..n} \sim P_{\mathit{\Pi}}}[G_0^\gamma | \langle \hat{\pi}, \pi_1, ..., \pi_{n-1} \rangle] \tag{2}$$

Because we can only control a subset of policies in AHT, equation (1) performance acts as an upper bound to both ZSC and AHT.

3 Previous Work

Previous work on finding static policies that are robust to unseen teammates includes strategies to increase the state variety seen by an agent by random partner augmentation [43], population training [38, 7] based on maximum strategy entropy [45], elo matchmaking [21], or evolutionary diversity [42]. Other efforts to generate models that generalize to all partners use grounded communication [22, 16, 19, 41] to adapt to teammates that are capable of giving requests or commands such as humans. Adaptive strategies include opponent modeling [25], role assignment via coach/player dynamics [20], and lifelong learning through online updates with novel partners [27]. A more comprehensive review of ad-hoc teamplay can be viewed here [23], but for the purpose of this paper, we focus on a population best response and opponent modeling. For the longevity of these results, we look at the optimistic case where agents are identified perfectly and the training population mimics the test-time population this way the current state of the art in each paradigm should not effect the theoretical outcomes shown here.

For population-based training we create some $\hat{P}_{\hat{\Pi}} \approx P_{\Pi}$ and then train $\hat{\pi}$ based on the population. The optimistic case for this approach, which we call Population Best Response (PBR), happens when $\hat{P}_{\hat{\Pi}} = P_{\Pi}$. For opponent modeling, we try to identify $\langle \pi_1, ..., \pi_{n-1} \rangle$ in order to perform some Bayesian update on $\hat{P}_{\hat{\Pi}}$ so that $\hat{\pi}$ can become a more specific best-response strategy. Opponent modeling assumes that with a better estimate of who we are playing with, performance will improve up to a limit when we have correctly identified other agents with certainty for which we play the best response. We abbreviate this optimistic case in the results as (OM). Lastly, communication, be it information or arbitrary, can contain implicit commands when learned as a best response. Communications produced by a best-response agent may inject data that manipulates teammate behavior, causing the upper bound performance to reach that of equation (1) with the maximum performance occurring when an agent is able to identify its teammates and choose a communication policy that manipulates them as positively as possible as in RIAL/DIAL [11]. Both implicit and explicit communication have an optimistic case where the most capable agent is able to successfully convince other agents to follow an optimal joint strategy $\boldsymbol{\pi}^*$, so the self-play results between two identical agents represent the optimistic case for both communication paradigms. We present MATCH as a more transparent alternative to implicit communication, which learns explicit commands within 125 time-steps, or 25 total messages sent with a command length of 5 steps. Additionally, MATCH includes a mechanism for unskilled agents to stop sending communication so that they don't decrease the performance of skilled agents. Finally, social learning [26] and various coach-player paradigms [20, 34] leverage the behavior of potentially superior teammates or advisors, by observing their

actions or following their suggestions, but these works operate on the order of tens or hundreds of thousands of environment time-steps.

Our method of learning command-based communication most closely relates to the simultaneous advisor-learner structure of [10] in which agents ask for action suggestions from other agents based on the perceived importance (the value difference between the best and worst move is high) and uncertainty (familiarity with the current state is low). Advice generates more intentional training data than random exploration because agents are more likely to find meaningful trajectories. Our algorithm, described in section 4 operates based on observed performance instead of importance or uncertainty. Commands are given to agents that listen, and commands are followed from agents whose commands have led to good outcomes in the past. MATCH relies on the normative belief that ad-hoc agents prefer to follow advice from agents that have been helpful so far.

4 Methodology

4.1 Generating Policies Via Deep RL

In order to generate competent policies for our environments, we used Munchausen Deep-Q Learning (M-DQN)[37] with a dueling Q architecture[40] and Proximal Policy Optimization (PPO)[31] to generate deterministic and stochastic policies respectively, due to their state-of-the-art performance and simple implementation. The choice of policy generation is arbitrary as MATCH is policy agnostic so long as a Q or Value function is maintained. The Q or Value estimate is essential to MATCH's listening component which decides whos commands to follow. The hyperparameters and code are available at https://github.com/Timothy-Flavin/Multi-Armed-Two-way-Command-Heuristic.

4.2 Temporal Difference and Advantage Estimation for Command Quality

Two functions of interest to us for estimating the value of following a command are $V^{\pi,\gamma}(s_t) = \mathbb{E}[G_t^\gamma|\pi, s_t]$ and $Q^{\pi,\gamma}(s_t, u_t) = \mathbb{E}[G_t^\gamma|\pi, s_t, u_t]$ which estimate the expected value of the rewards to-go from a current state s_t for policy π and for the Q function, action u_t. Let the advantage $A^{\pi,\gamma}(s_t, u_t)$ be defined by equation 3 and let the single step temporal difference residual $\delta_t^{V^{\pi,\gamma}}$ be defined by equation 4 from [30] where $A^{\pi,\gamma}(s_t, u_t) = \mathbb{E}[\delta_t^{V^{\pi,\gamma}}]$. We also have the k-step advantage defined by equation 5 and generalized advantage in equation 6 from [30] where k and λ adjust the bias and variance of advantage estimates.

$$A^{\pi,\gamma}(s_t, u_t) := Q^{\pi,\gamma}(s_t, u_t) - V^{\pi,\gamma}(s_t) \tag{3}$$

$$\delta_t^{V^{\pi,\gamma}} := -V^{\pi,\gamma}(s_t) + r_t + V^{\pi,\gamma}(s_{t+1}) \tag{4}$$

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_t^{V^{\pi,\gamma}} \tag{5}$$

$$\hat{A}_t^{GAE(\gamma,\lambda)} := \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_t^{V^{\pi,\gamma}} \tag{6}$$

For $\hat{A}_t^{GAE(\gamma,\lambda)}$ there are two special cases that we are interested in, $\lambda = 0$ which reduces to the single-step TD errors with the least variance but the most bias, and $\lambda = 1$ which reduces to the Monte Carlo Advantage $G_t^{\gamma} - V_t^{\pi,\gamma}$ which is unbiased but also high variance for summing over all $r_t : r_{\infty}$. We discuss in section 4.4 that bias is of particular concern when evaluating the quality of communications received from another agent, so a higher variance estimator with low bias is recommended.

4.3 Multi-Armed Bandits of Interest to MATCH

Multi Armed Bandit (MAB) problems consist of a gambler choosing at each round to play one of K slot machine arms, each defined by an unknown reward distribution. The gambler wants to maximize cumulative earnings over some time horizon T by selecting arms and observing their payoffs [5, 6, 13, 28]. While there are many algorithms for solving MAB problems, we focus on Thompson Sampling [36] because of its strong empirical performance, insensitivity to hyperparameters, and its ability to incorporate prior information as a Bayesian method.

Listening MAB When listening to commands, agents can only take one action at a time, so they can only follow one command at a time. The true expected payoff for following a command is a real number drawn from an unknown distribution (1) based on the joint policy of the team. In ad-hoc teamplay, policies are non-stationary (2), so the command-reward distribution is also non-stationary. An agent will only receive commands from a subset of its teammates (3), so some "arms" of the bandit are inactive. These requirements culminate in the problem definition of the listening MAB for MATCH as a non-stationary (2) Gaussian (1) sleepy (3) bandit.

Speaking MAB An agent may instruct or ignore (1) any combination (2) of teammates that it chooses. Consequently, it may update its estimated probability of being followed for each of the agents it instructed (3). Finally, other agents may change their own listening probability over time (4). These requirements define the problem of MATCH's speaking module as a non-stationary (4) Bernoulli (1) combinatorial (2) semi-bandit (3).

Implementation We use an Inverse Gamma distribution as a prior for listener arm variance with the sample means of previous arm pulls serving as our prior means. We use Beta distributions as the priors for each speaker arm. Both the Inverse Gamma and Beta distributions are parameterized by $\theta = \{\alpha, \beta\}$, which monotonically increase over time as samples are collected, leading to a more

confident estimate of rewards as time goes on. To handle non-stationarity, we exponentially decay $\{\alpha, \beta\}$ towards some $\theta_0 = \{\alpha_0, \beta_0\}$ using $\theta_{t+1} = \gamma\theta_t + (1-\gamma)\theta_0$ with decay parameter $\gamma \in [0, 1]$ to continually inject uncertainty into our Thompson samplers. We also update the distribution means via exponential decay with learning rate $\lambda \in (0, 1)$ so that $\mu_{t+1} = \lambda r_t + (1-\lambda)\mu_t$. In the case of the Gaussian MAB, we also experimented with slowing the decay of β by setting γ closer to 1 for any arms which were not sampled recently, so that the variance drawn from the Inverse Gamma distribution increases for arms which are pulled less often. This modification causes the sampler to explore more, but if treated without care, such as $\gamma = 1.0$ for un-pulled arms, it can cause such high variance that the sampler collapses into a uniform sampler over arms. For the speaking bandit, agents that receive more commands in a single timestep are less likely to listen to my command, so positive samples are weighted by the probability of listening if the listening agent were following a uniform distribution. Intuitively if another agent listens to my command more than uniformly, it must prefer my commands to some degree so I should keep talking to it as described in social learning [26].

4.4 Multi-Armed Two-way Command Heuristic (MATCH)

MATCH consists of two modules, a speaker MAB and a listener MAB described in section 4.3. The speaker MAB is rewarded when agents follow a command. The listener MAB is rewarded by calculating advantage as in section 4.2 after following a command. MATCH does not generate commands. For each speaker bandit arm i referring to some agent i that is pulled, MATCH gets command content from it's policy by calling something like this:

```
command[i] = my_policy.take_action(observation_estimate[i])
```

Each listener bandit maintains an arm for its own policy because the expected advantage over actions or commands is unlikely to be mean-zero when an agent is interacting with new teammates on which its value function was not trained. In other words, every action might look good or bad if teammates are better or worse than an agent's training partners. By maintaining a 'self' arm, the listener can normalize the quality of commands among this ad-hoc team. The second source of bias comes from the fact that the agent is changing its policy at runtime in response to commands. The value estimate may degrade for trajectories that haven't been seen in training. In order to overcome this systemic bias, we recommend advantage estimates close to the multi-step Monte Carlo estimate, GAE with λ close to 1.

A command as an action or list of actions allows for the analysis of command length's effect on performance and advantage estimation quality. However, MATCH only requires that a command can be generated by it's underlying policy and then explicitly followed or ignored by another agent. If followed, the listening agent is essentially handing its controls over to the commander temporarily. The listener may commit to following the commander for more than

one step in a row in order to calculate k-step returns or generalized advantage to lower the bias of its estimate of the commander's quality.

5 Environment, Lever-NVNTTT

We introduce "Lever-NvNTTT" as a new minimally complex benchmark for cooperative multi-agent algorithms where two teams of 'N' teammates play Tic Tac Toe against one another on a shared board. In Lever-NvNTTT, each agent on a given team must choose the same square in order to place a piece. If the agents choose different squares, then their turn is skipped. For this paper, the opponent plays a random legal move each turn and the environment offers a single terminal reward of 1.0 for a win, 0.0 for a tie, and -1.0 for a loss. This environment encourages team consensus among several symmetric strategies in the same way as the Lever Game introduced in [17]. A single agent taken at two different stages of self-play training with parameter sharing will exhibit a skill gap, but not an equilibrium selection problem. Agents trained from two different starting seeds may exhibit equilibrium selection without exhibiting a skill gap. We hope that NvNTTT will serve as an extremely computationally cheap environment for debugging and benchmarking teamplay algorithms where the researcher can directly select whether they are solving the skill gap problem or the equilibrium selection problem.

6 Results

Crossplay results are shown in figure 1. We trained an initial population of PPO and M-DQN agent policies via self-play Π_{sp} over 5 seeds (A). We chose the highest self play performing seeds, $[0, 1]$, with average rewards R: $\{0.0, 0.4, 0.9\}$.

For our baselines, we trained a best response model to a uniform distribution over Π_{sp} to generate (PBR) (Graph E rows 1 and 2). Next, we trained best response models to each individual policy in Π_{sp} to represent the opponent modeling upper bound (OM) where opponents are known exactly (Graph E rows 4 and 5). Note that self-play with parameter sharing outperforms best response because choosing the same square as your partner is always optimal and parameter sharing causes agents to choose actions from the exact same distribution. We also compare the performance of online RL after 500 episodes (2,500 steps) of training with each partner to show that online Deep-RL alone does not solve Ad-Hoc Teamplay in an ad-hoc timescale (Graph D). We then show stubborn-MATCH where the row agent always ignores its partner and always gives commands which represents an agent with aggressive priors set correctly or incorrectly (Graph B). Finally, we show MATCH with 125 steps and 1,000 steps and GAE (Graphs C and F).

We found little difference between advantage methods, but a complete set of Cross-Play matrices are included in the repository. For Lever-NvNTTT, we record the mean cross-play scores over 10 runs for each method in table 1 with

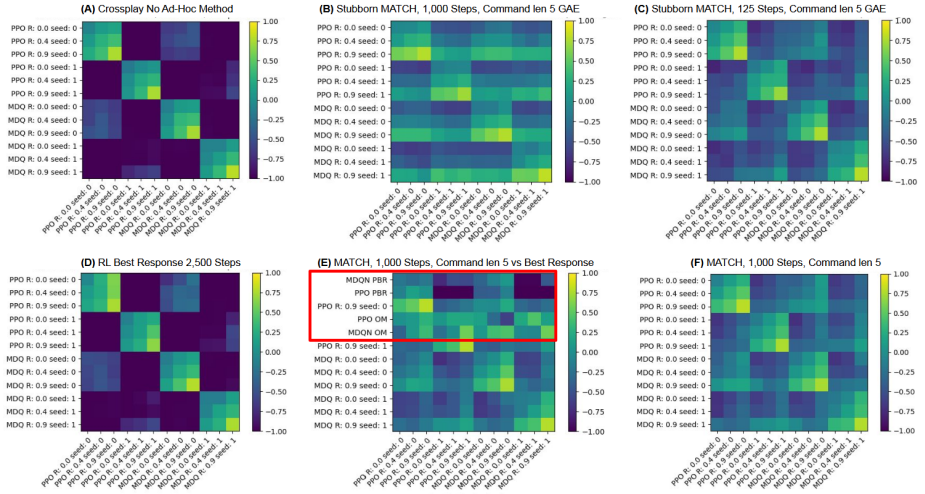

Fig. 1. Cross-play Results for LeverTTT (A): No Ad-Hoc Paradigms Used, (B): Stubborn MATCH run for 1,000 steps where the row player will always ignore and give a command, (C): MATCH run for 125 steps, (D): The Row player was allowed to keep learning via PPO or M-DQN for 2,500 steps, (E): Some of the row players (within the red box) are replaced with population best response PBR, PPO with a stubborn match, and optimistic opponent modeling (OM) where the training partner is known exactly, (F): MATCH with GAE for 1,000 steps

the full set of cross-play matrices included in the supplementary materials. We also include several illustrative cross-play tables here.

7 Discussion

7.1 Cross-Play Results

Graph A shows that there is an equilibrium selection problem because agents with different seeds are unable to coordinate effectively (off-diagonal). Graph A also shows a skill gap problem as agents trained with the same seed at different performance checkpoints degrade in team performance. Graph B illustrates that when the agents are running MATCH with a row player that has a strong prior to send commands and ignore advice, the team performance approaches the row player's self-play performance. Figures C and F show MATCH when both agents are flexible, which is less destructive than a stubborn row player who is incompetent. We show MATCH at two different time horizons to illustrate that the non-stationary decay we implement causes MATCH to reach its performance ceiling fairly quickly. Graph D shows the result when the individual agents are allowed to continue learning online via their original RL algorithms. Some improvement is realized, but the timescale required for Deep RL is prohibitive for

ad-hoc learning. Finally Graph E shows that a single unadaptive policy can't generalize to a population with non-complementary strategies (PBR). Opponent Modeling (OM) allows for partner-specific behavior and it almost reaches the self-play upper bound, but it required the full 600,000 training time-steps that each original RL agent trained for with each partner, and it requires that partners are identifiable and similar to partners seen in training.

7.2 Final Remarks

In this work, we proposed a modular algorithm, MATCH, which can be added to existing agents, allowing them to communicate through simple unambiguous commands. MATCH is based on two normative beliefs about ad-hoc interaction. First, I should listen to advice from people who have given me good advice in the past. Second, I should offer advice to people who listen. These two basic signals lead to a communication protocol that learns a directional graph where every edge between agents that improves team performance is strengthened while edges that degrade team performance are weakened until a local "maximum flow" of performance between agents is achieved. Alternatively, two capable and incompatible agents can be thought of as sitting on a saddle point in the team performance landscape. The stochasticity of MATCH's MAB samplers and advantage leads to a team-wise stochastic gradient descent towards locally optimal joint policy mixture without risking policy collapse or catastrophic forgetting because no network parameters are ever retrained. In addition to stability, MATCH is a data efficient algorithm. For N agents, up to N^2 commands can be sent per timestep with N commands being followed by default. If multiple agents send the same command to a single listener, then both commands can be "followed", leading to potentially more updates per command period. In total, $2N^2$ relationships are being learned, and at least $N + N^2$ updates can be performed per command step. Finally, state diversity and zero-shot strategies are compatible with MATCH so long as agents generate a value function.

As the field of MARL expands, we hope to see a diverse set of methods developed to protect policies from collapsing under novel circumstances or during ad-hoc coordination with new teammates. We also want to develop algorithms that are capable of listening to instructions while exercising prudence when presented with potentially harmful or dangerous suggestions. We hope that methods like MATCH may be a stepping stone to allow artificial agents to almost-always accept input from humans via simple adjustable prior beliefs without blindly following bad actors. Critically, we also want to develop agents which do not assume that they are the most skilled agent possible in an environment. The ability to deffer to and learn from more skilled entities, including humans, is valuable for the longevity of a deployed autonomous system. Finally, transparent communication can expose agent intentions and shift liability towards the commanding agent in events where autonomous systems fail. We hope that MATCH can open more research into responsive, transparent, life-long learning autonomous systems.

References

1. Agmon, N., Barrett, S., Stone, P.: Modeling uncertainty in leading ad hoc teams. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. pp. 397–404 (2014)
2. Agmon, N., Stone, P.: Leading ad hoc agents in joint action settings with multiple teammates. In: AAMAS. pp. 341–348 (2012)
3. Bard, N., Foerster, J.N., Chandar, S., Burch, N., Lanctot, M., Song, H.F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al.: The hanabi challenge: A new frontier for ai research. Artificial Intelligence **280**, 103216 (2020)
4. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of operations research **27**(4), 819–840 (2002)
5. Berry, D.A., Fristedt, B.: Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). London: Chapman and Hall **5**(71-87), 7–7 (1985)
6. Besbes, O., Gur, Y., Zeevi, A.: Stochastic multi-armed-bandit problem with non-stationary rewards. Advances in neural information processing systems **27** (2014)
7. Carroll, M., Shah, R., Ho, M.K., Griffiths, T., Seshia, S., Abbeel, P., Dragan, A.: On the utility of learning about humans for human-ai coordination. Advances in neural information processing systems **32** (2019)
8. Crawford, V.P., Sobel, J.: Strategic information transmission. Econometrica: Journal of the Econometric Society pp. 1431–1451 (1982)
9. Cui, B., Hu, H., Lupu, A., Sokota, S., Foerster, J.: Off-team learning. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 15407–15419. Curran Associates, Inc. (2022)
10. Da Silva, F.L., Glatt, R., Costa, A.H.R.: Simultaneously learning and advising in multiagent reinforcement learning. In: Proceedings of the 16th conference on autonomous agents and multiagent systems. pp. 1100–1108 (2017)
11. Foerster, J., Assael, I.A., De Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. Advances in neural information processing systems **29** (2016)
12. Garcıa, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research **16**(1), 1437–1480 (2015)
13. Gittins, J., Glazebrook, K., Weber, R.: Multi-armed bandit allocation indices. John Wiley & Sons (2011)
14. Glikson, E., Woolley, A.W.: Human trust in artificial intelligence: Review of empirical research. Academy of Management Annals **14**(2), 627–660 (2020)
15. Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Knoll, A.: A review of safe reinforcement learning: Methods, theory and applications. arXiv preprint arXiv:2205.10330 (2022)
16. Hu, H., Lerer, A., Cui, B., Pineda, L., Brown, N., Foerster, J.: Off-belief learning. In: International Conference on Machine Learning. pp. 4369–4379. PMLR (2021)
17. Hu, H., Lerer, A., Peysakhovich, A., Foerster, J.: "other-play" for zero-shot coordination. In: International Conference on Machine Learning. pp. 4399–4410. PMLR (2020)
18. Kirk, R., Zhang, A., Grefenstette, E., Rocktäschel, T.: A survey of zero-shot generalisation in deep reinforcement learning. Journal of Artificial Intelligence Research **76**, 201–264 (2023)

19. Kuo, Y.L., Katz, B., Barbu, A.: Compositional rl agents that follow language commands in temporal logic. Frontiers in Robotics and AI **8**, 689550 (2021)
20. Liu, B., Liu, Q., Stone, P., Garg, A., Zhu, Y., Anandkumar, A.: Coach-player multi-agent reinforcement learning for dynamic team composition. In: International Conference on Machine Learning. pp. 6860–6870. PMLR (2021)
21. Liu, S., Lever, G., Wang, Z., Merel, J., Eslami, S.A., Hennes, D., Czarnecki, W.M., Tassa, Y., Omidshafiei, S., Abdolmaleki, A., et al.: From motor control to team play in simulated humanoid football. Science Robotics **7**(69), eabo0235 (2022)
22. MacGlashan, J., Littman, M., Loftin, R., Peng, B., Roberts, D., Taylor, M.: Training an agent to ground commands with reward and punishment. In: Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
23. Mirsky, R., Carlucho, I., Rahman, A., Fosong, E., Macke, W., Sridharan, M., Stone, P., Albrecht, S.V.: A survey of ad hoc teamwork research. In: European conference on multi-agent systems. pp. 275–293. Springer (2022)
24. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. arXiv preprint arXiv:1703.04908 (2017)
25. Nashed, S., Zilberstein, S.: A survey of opponent modeling in adversarial domains. Journal of Artificial Intelligence Research **73**, 277–327 (2022)
26. Ndousse, K.K., Eck, D., Levine, S., Jaques, N.: Emergent social learning via multi-agent reinforcement learning. In: International conference on machine learning. pp. 7991–8004. PMLR (2021)
27. Nekoei, H., Zhao, X., Rajendran, J., Liu, M., Chandar, S.: Towards few-shot coordination: Revisiting ad-hoc teamplay challenge in the game of hanabi. In: Conference on Lifelong Learning Agents. pp. 861–877. PMLR (2023)
28. Robbins, H.: Some aspects of the sequential design of experiments. American Mathematical Society (1952)
29. Samuelson, L.: Evolutionary games and equilibrium selection, vol. 1. MIT press (1997)
30. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
31. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
32. Sebanz, N., Knoblich, G.: Progress in joint-action research. Current Directions in Psychological Science **30**(2), 138–143 (2021)
33. Stone, P., Kaminka, G., Kraus, S., Rosenschein, J.: Ad hoc autonomous agent teams: Collaboration without pre-coordination. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 24, pp. 1504–1509 (2010)
34. Subramanian, S.G., Taylor, M.E., Larson, K., Crowley, M.: Multi-agent advisor q-learning. Journal of Artificial Intelligence Research **74**, 1–74 (2022)
35. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the tenth international conference on machine learning. pp. 330–337 (1993)
36. Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**(3-4), 285–294 (1933)
37. Vieillard, N., Pietquin, O., Geist, M.: Munchausen reinforcement learning. Advances in Neural Information Processing Systems **33**, 4235–4246 (2020)
38. 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 **575**(7782), 350–354 (2019)

39. Wang, C., Rahman, A., Durugkar, I., Liebman, E., Stone, P.: N-agent ad hoc teamwork. arXiv preprint arXiv:2404.10740 (2024)
40. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling network architectures for deep reinforcement learning. In: International conference on machine learning. pp. 1995–2003. PMLR (2016)
41. Xu, S., Wang, H., Wu, Y.: Grounded reinforcement learning: Learning to win the game under human commands. Advances in Neural Information Processing Systems **35**, 7504–7519 (2022)
42. Xue, K., Wang, Y., Guan, C., Yuan, L., Fu, H., Fu, Q., Qian, C., Yu, Y.: Heterogeneous multi-agent zero-shot coordination by coevolution. IEEE Transactions on Evolutionary Computation (2024)
43. Yan, X., Guo, J., Lou, X., Wang, J., Zhang, H., Du, Y.: An efficient end-to-end training approach for zero-shot human-ai coordination. Advances in Neural Information Processing Systems **36** (2024)
44. Yu, P., Mishra, M., Zaidi, S., Tokekar, P.: Tactic: Task-agnostic contrastive pretraining for inter-agent communication. arXiv preprint arXiv:2501.02174 (2025)
45. Zhao, R., Song, J., Yuan, Y., Hu, H., Gao, Y., Wu, Y., Sun, Z., Yang, W.: Maximum entropy population-based training for zero-shot human-ai coordination. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 6145–6153 (2023)
46. Zhu, C., Dastani, M., Wang, S.: A survey of multi-agent reinforcement learning with communication. arXiv preprint arXiv:2203.08975 (2022)