Learning When to Cooperate Under Heterogeneous Goals

Anonymous submission

Abstract

todo

Introduction

The human capacity for cooperation and collaborative problem-solving is one of our most remarkable adaptations, having arguably played a crucial role in developing the culture and civilisation that persist to this day (Tomasello et al. 2012; Henrich 2018). Accordingly, the creation of autonomous systems with humanlike cooperative capabilities is a longstanding goal of research in AI and robotics, most notably under the moniker of ad hoc teamwork (AHT). Despite impressive progress, the AHT setting as typically considered leaves aside certain aspects that are crucial to 'real-world' cooperation. In particular, AHT tends to assume that every scenario is equally cooperative; i.e. it is always optimal to pursue some form of collaborative strategy with respect to the other agents in the environment. In the real world, things aren't so simple—while some scenarios present fruitful opportunities for collaboration, in others it makes more sense to take an independent approach and focus on whatever we can achieve alone. A successful agent should be able to distinguish between these settings, and adapt their strategy accordingly—just as humans do.

While the definition of AHT explicitly allows for agents to have different reward functions, this has been little explored in prior work. We propose to study the setting where agents can be considered as having broadly the same basic *underlying* goal (e.g. collect fruits), but may pursue different 'variants' of this goal (e.g. collect apples vs collect oranges)—and importantly, other agents' goals are not known a priori by the learner. Our contributions are threefold. First, we formalise this setting. Second, we extend two popular task environments used in AHT research to support heterogeneous goals. Finally, we propose **GRILL** (Goal selection by RL with Imitation for Low-Level control), a novel hierarchical method that outperforms baseline approaches at ...

Related work

Ad hoc teamwork

The field of ad hoc teamwork (AHT) deals with the problem of developing agents that learn to collaborate 'on the fly' with previously unseen 'teammates', without prior coordination (Mirsky et al. 2022). AHT shares some basic elements with the field of multi-agent reinforcement learning (MARL); but where MARL typically assumes control of all agents in the environment, in AHT we control only a single agent (often called the 'AHT agent' or 'learner'; we will use 'ego agent' throughout), with teammates' actions governed by either simple heuristics or pre-trained (frozen) RL policies.

At its core, the AHT problem is about adapting to some source of behavioural diversity across a population of different potential teammates. In practice, this has mostly taken the form of teammates having different 'styles' of policy, while being otherwise homogeneous. Many approaches have involved explicit inference and representation of teammate policy types; traditionally via forms of Bayesian belief-updating over a discrete teammate space (Gmytrasiewicz and Doshi 2005; Barrett, Stone, and Kraus 2011; Albrecht and Ramamoorthy 2013; Albrecht, Crandall, and Ramamoorthy 2016), or more recently using neural-network-based encoders to learn latent policy representations (Rabinowitz et al. 2018; Papoudakis and Albrecht 2020; Rahman et al. 2023). Some recent work has also considered heterogeneity in teammates' *capabilities* (e.g. via different robot morphologies) (Liemhetcharat and Veloso 2014; Liu et al. 2024). However, while the definition of the AHT problem allows for agents to have separate individual reward functions (within a broader cooperative task), the vast majority of existing work assumes a single reward function that is common to the ego agent and all their teammates. In the current work, we focus on this underexplored dimension of heterogeneity by considering a setting in which agents can differ in the goals they pursue under a given high-level task.

Hierarchical reinforcement learning

Where the 'traditional' RL setting considers actions at only a single resolution, hierarchical methods operate over multiple levels of temporal abstraction. By breaking down the learning problem in this way, HRL algorithms can (at least in principle) offer better sample complexity and/or generalisation relative to their non-hierarchical counterparts. While recent research has focused on the more general Options framework (Sutton, Precup, and Singh 1999; Bacon, Harb, and Precup 2017; Barreto et al. 2019; Klissarov and Pre-

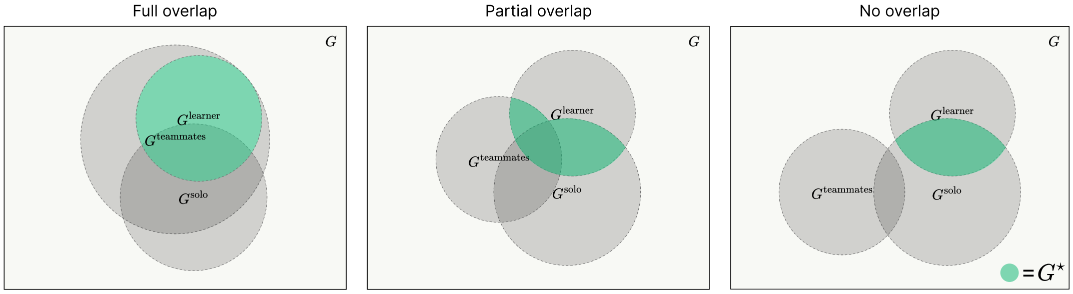

Figure 1: Caption

cup 2021), of more relevance to our work is the earlier approach of Feudal RL (FRL) (Dayan and Hinton 1992). In FRL, the agent consists of distinct high-level and low-level policies, where the high-level policy is used to select 'subgoals', and the low-level policy is used to select elementary actions conditioned on a given subgoal. This is very similar to the method we describe here; the main difference is that HRL optimises the low-level policy via a reward function of the form $R(s, a \mid s) = \mathbb{I}(s = g)$ (see the following section), while we use behavioural cloning with learned goal labels (using RL only to learn high-level goal selection).

Goal-conditioned policy learning

Goal-conditioned reinforcement learning (GCRL, sometimes also referred to as multi-goal or multi-objective RL) is an extension of the basic RL paradigm in which the agent must learn an action policy conditioned on different 'goals' g; i.e. $\pi(a \mid o, g)$ (Kaelbling 1993; Liu, Zhu, and Zhang 2022). The low-level policy in Feudal RL (Dayan and Hinton 1992) (see previous section) is one early example of GCRL, but not all GCRL approaches involve a hierarchical structure. The goal space is typically operationalised as a subset of the state space (where the agent is rewarded only for reaching the particular state corresponding to their current goal)—but may also be represented in natural language [refs], or as particular levels of expected episode return [refs]. While most GCRL approaches involve pre-specified goals, some recent work has also explored the possibility of agents learning to generate their own goals (Colas et al. 2022).

Of more direct relevance to our own method is work that incorporates the idea of goal-conditioned policies into *imitation* learning. For example, Lynch et al. use a combination of self-supervised representation learning and supervised behaviour learning to obtain goal-conditioned robot policies from unstructured play data (Lynch et al. 2019). Ghosh et al. employ goal-conditioned behaviour cloning in a self-imitation paradigm, where the agent's previous trajectories are used in supervised learning as successful examples for the states they *actually* reached (regardless of the states the agent 'intended' to reach) (Ghosh et al. 2020). Other works have explored goal-conditioned imitation learning with a version of the GAIL algorithm (Ding et al. 2020), using diffusion policies (Reuss et al. 2023), or transformer architectures (Sundaresan et al. 2025).

Problem setting

We consider the problem in which a *learner* must act to achieve goals in an environment populated by a number of other agents (aka *teammates*) with potentially mixed objectives. Note that while the experimental results we present use only a single teammate, for purposes of generality we lay out the arbitrary n-teammate case here.

Formally, we define our problem within the framework of Partially-Observable Stochastic Games (POSG). A POSG is defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{T}, \{\mathcal{A}^i\}, \{\mathcal{O}^i\}, \{\mathcal{Z}^i\}, \{r^i\}, \gamma \rangle$ where $\mathcal{N}$ is the set of agents (with $i \in \mathcal{N}$), $\mathcal{S}$ is the state space, and $\mathcal{T} : \mathcal{S} \times \vec{\mathcal{A}} \mapsto \Delta(\mathcal{S})$ is the transition function, with $\vec{\mathcal{A}} = \mathcal{A}_1 \times ... \times A_N$ denoting the joint action space. For each agent i we have an action space $\mathcal{A}^i$, an observation space $\mathcal{O}^i$, an observation function $\mathcal{Z}^i : S \mapsto \mathcal{O}^i$ and a reward function $r^i : S \times \vec{\mathcal{A}} \mapsto \mathbb{R}$. $\gamma \in [0, 1]$ is the discount factor.

In general, we consider an environment which offers a set G of different possible 'goals'; some of which can be achieved by a single agent acting alone, and some of which require cooperation between two or more agents. Informally, we consider all agents' objectives as being binary masks over G—i.e., for each goal $g \in G$, an agent i will receive a reward of either r_g or 0, where r_g is the 'base reward' for goal g. We can therefore express all individual differences in reward in terms of the goal subsets $G^i \subseteq G$ for which different agents receive nonzero reward. For example, in a foraging environment where agents have to collect different varieties of fruit, we might have $G = \{\text{apples}, \text{oranges}, \text{plums}\}$ and $G^i = \{\text{apples}\}$. We also make the assumption that goals which require cooperation are always more rewarding than those that don't, i.e. $r_g > r_{g'} \ \forall \ g \notin G^{\text{solo}}, g \in G^{\text{solo}}$—intended to reflect the reality that people can typically achieve more acting together than alone.

Each agent in the environment expresses an observable cue ϕ^i that noisily signals their goals, i.e. $\phi^i \sim \mathcal{N}(G^i, \sigma^2 I)$. Unless otherwise stated, we will use $\sigma^2 = .05$ throughout our experiments.

We can think about rational/optimal behaviour in this setting through the lens of the goals that are *worthwhile* for the ego agent to pursue (i.e. both rewarding and potentially achievable). The set of such goals can be written as $G^\star = G^{\text{learner}} \cap \left(G^{\text{teammates}} \cup G^{\text{solo}}\right)$, where G^{learner} is the learner's goal subset, $G^{\text{teammates}}$ is the union of all team-

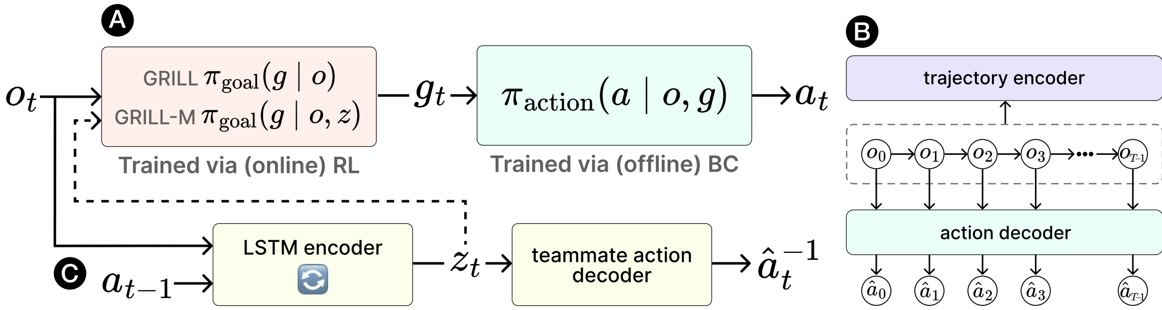

Figure 2: **(A)** The hierarchical architecture of GRILL **(B)** The encoder-decoder architecture optimised offline in stage 1, from which the action decoder becomes the low-level policy π_{action} in stage 2 **(C)** The auxiliary modelling component used in GRILL-M (but not GRILL)

mates' goal subsets, and G^{solo} is the subset of goals that don't require cooperation. Figure 1 illustrates how $G^{\star}$ (highlighted in green) changes across the three basic scenarios that arise from this general setting:

1. **full-overlap** (all of the learner's goals are shared by at least one teammate):
   $G^{\text{learner}} \subseteq G^{\text{teammates}}$
2. **partial-overlap** (at least one but not all of the learner's goals are shared by at least one teammate):
   $G^{\text{learner}} \cap G^{\text{teammates}} \neq \emptyset$, $G^{\text{learner}} \not\subseteq G^{\text{teammates}}$
3. **no-overlap** (none of the learner's goals is shared by any teammate):
   $G^{\text{learner}} \cap G^{\text{teammates}} = \emptyset$

To succeed at the overall task, a learner should be able to navigate **all** of these scenarios—determining when to pursue collaborative goals and when to act by themselves.

Method

We present two methods: GRILL and GRILL-M. GRILL-M is a variant of GRILL that incorporates a version of the auxiliary teammate modelling component from LIAM (Papoudakis, Christianos, and Albrecht 2021).

The core idea behind GRILL is to separate the problems of **(1)** learning which goals to attempt given a particular state, and **(2)** learning which actions to take in order to achieve those goals. That is, rather than learning a single end-to-end policy, we learn separately a 'high-level' policy for goal selection, and a 'low-level' policy for goal-conditioned action selection. This kind of hierarchical structure is not new (see Section)—rather, our insight is that the optimal low-level policy is universal to all agents in the population, whereas the optimal high-level policy depends on the goals of both the ego agent and their current teammate. We can therefore use a two-stage process that combines imitation and reinforcement learning (illustrated in Figure 2):

Stage 1: In stage 1, we first collect a small offline dataset $\mathcal{D} = \{o_t, a_t\}$ of observations and actions from randomly sampled heuristic agents. $\mathcal{D}$ is then split into a set of fixed-length trajectories $\{\tau\}$, where in each trajectory the agent is pursuing a single goal. We then use this dataset to optimise an encoder-decoder model. For each trajectory, the encoder produces a discrete goal label. Given the discrete encoding, one decoder then tries to predict the teammate's action at each point along the trajectory from the preceding observation; a second decoder tries to predict the final observation from the first. The whole system is optimised solely to reconstruct actions and observations from $\mathcal{D}$, with no explicit goal information provided:

$$\mathcal{L}_a = -\sum_{t=0}^{T-1} \log \text{dec1}_{\theta_{\text{dec1}}}\left(a_t \mid o_t, \hat{g} = \text{enc}_{\theta_{\text{enc}}}(\tau)\right)$$

$$\mathcal{L}_o = \left\| o_{T-1} - \text{dec2}_{\theta_{\text{dec2}}}\left(o_0, \hat{g} = \text{enc}_{\theta_{\text{enc}}}(\tau)\right)\right\|_2^2$$

$$\min_{\theta_{\text{enc}},\theta_{\text{dec1}},\theta_{\text{dec2}}} \mathop{\mathbb{E}}_{\tau\sim\mathcal{D}}\left[\lambda\mathcal{L}_a + (1-\lambda)\mathcal{L}_o\right] \qquad (1)$$

where λ is a scalar that trades off between the two prediction objectives. After training, we discard the encoder and the observation decoder while retaining the action decoder as our goal-conditioned BC policy π_{action}.

Stage 2: we use PPO to learn a high-level policy π_{goal} mapping the current observation to a discrete goal, the output of which is used to condition the BC policy from the previous stage. For GRILL-M, this stage also involves an auxiliary objective, where the ego agent is trained to predict their teammate's actions from their own observations and actions via an LSTM encoder-decoder:

$$\min_{\theta_{\text{enc}},\theta_{\text{dec}}} -\log \text{dec}_{\theta_{\text{dec}}}\left(a_t^{-1} | z_t = \text{enc}_{\theta_{\text{enc}}}(o_t, a_{t-1})\right) \qquad (2)$$

with π_{goal} learned over the augmented space $\mathcal{O} \times \mathcal{Z}$. This is almost identical to the modelling component of LIAM (Papoudakis, Christianos, and Albrecht 2021), with the sole difference that since our environments are fully observable, we remove the additional prediction of teammate observations.

Experiments

Environments

We extend two commonly used, fully observable AHT environments to incorporate the notion of goal heterogeneity, with example states of both shown in Figure 3.

Cooperative reaching: Cooperative reaching (CR) is a simple gridworld task where two agents must navigate to

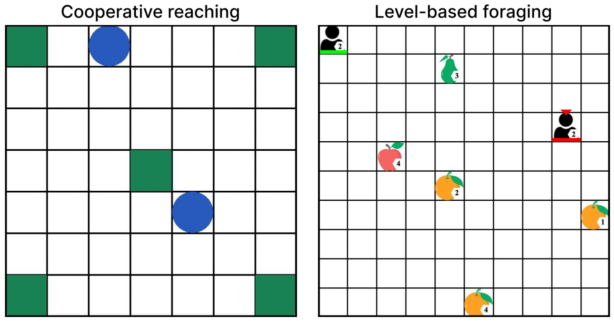

Figure 3: Example frames from the two AHT environments we extend.

and jointly occupy one of four reward-producing corner tiles. In the original version, different corners are associated with different amounts of reward, but the reward for reaching a given corner does not vary between agents. In our version, each corner tile yields a reward of either 1 or 0 for different agents; we also add an additional goal tile in the centre of the grid which can be successfully reached by a single agent (i.e. is in G^{solo}) but produces a lower reward of 0.2. All goal tiles act as absorbing states—once an agent has reached one of these tiles, they are unable to move out of it. The ego agent receives observations containing their own location, and the location and observable cue (ϕ^i) of their teammate. The action space consists simply of the four cardinal directions plus a no-op action.

Level-based foraging: Level-based foraging (LBF) is another gridworld environment, where agents cooperate to collect items. Each agent and item have an associated 'level', where an item with level l can only be collected by the joint efforts of a group of agents whose levels sum to $\geq l$. Our version of the environment extends this: rather than a single item type, we have three different 'fruits' (which can also still vary in level). The set of possible goals that the environment can contain is therefore given by $G = \{(\text{type}, l) \ \forall \ \text{type} \in \{\text{apple}, \text{orange}, \text{plum}\}, l \in [1, l_{\text{max}}]\}$ where l_{max} is the maximum item level. The ego agent receives observations containing the location, type and level of all fruits in the environment, as well as their own location and level, plus the location, level and observable cue (ϕ^i) of their teammate. The action space consists of the four cardinal directions, a 'load' action (to attempt fruit collection) and a no-op action. For simplicity, our experiments fix both ego agent and teammate at level 1.

Baselines

We evaluate GRILL against three baselines: PPO, LIAM and OMG. PPO (Schulman et al. 2017) is a general RL algorithm that for our purposes serves as a floor for what can be achieved by a method not tailored in any way to cooperative (or even multi-agent) settings. LIAM (Papoudakis, Christianos, and Albrecht 2021) and OMG (Yu, Jiang, and Lu 2024) are two recent methods that use some form of agent modelling to achieve competitive performance on AHT tasks similar to those we use here. Both LIAM and OMG train the ego agent's policy over an 'augmented' observation space incorporating a latent teammate representation produced by an auxiliary network. In LIAM, this is produced by a recurrent encoder-decoder trained to predict teammates' current observations and actions from the ego agent's own (observation, action) pairs. OMG instead uses the latent representation from a conditional VAE optimised to model the teammate's 'subgoal' (as a feature embedding of some future state).

We note that while the LIAM authors used A2C, and OMG used different RL algorithms for different tasks, our implementations of both use PPO as a backbone for consistency across methods. For each environment, we also compare against an 'oracle' policy which uses a shortest-path heuristic guided by full knowledge of all agents' goals and the base rewards.

Evaluation

In addition to the standard evaluation metric of average episode return, we also want to take a deeper look at how trained policies deal with the three scenarios enumerated in Section . To do this, we look at which goals the ego agent *attempts* to reach during evaluation episodes—i.e. which absorbing tiles does it navigate to in CR, and which fruits does it try to load in LBF. We consider three distinct failure modes relating to goal selection. The first (and most basic) is to seek goals outside of G^{learner}; i.e., goals which won't even yield any reward if achieved. The second is to be *over-collaborative* by seeking goals that are in G^{learner} but unachievable ($\notin G^{\text{solo}} \cup G^{\text{teammates}}$; note that this failure mode doesn't exist in the 'full overlap' scenario). The third and most subtle is to be *under-collaborative* by not seeking achievable cooperative goals when such goals exist (note that this failure mode doesn't exist in the 'no overlap' scenario). A successful agent should avoid all three of these failure modes.

We measure the distribution of ego agent goal choices over four distinct subsets that allow us to distinguish these three failure modes (see Figures 1 and 5). As an additional summary metric, we also report the 'cooperativity difference' (CD) as the difference in the proportion of attempted goals $\notin G^{\text{solo}}$ between the 'full overlap' and 'no overlap' scenarios—where a higher value indicates more sensitivity to ...

Results

GRILL achieves higher returns than all baselines

Figure 4 (top row) shows the average return after training, relative to each environment's oracle policy, for the three scenarios enumerated in Section . Overall, we find that GRILL and GRILL-M outperform all baselines across every scenario in both environments.

Taking a closer look at the results for cooperative reaching (left), we can see that for partial- and full-overlap all methods do similarly well, achieving returns close to or even slightly above the oracle policy. In the no-overlap scenario, all methods (including ours) do considerably worse—interestingly, this is the only case in which we observe a

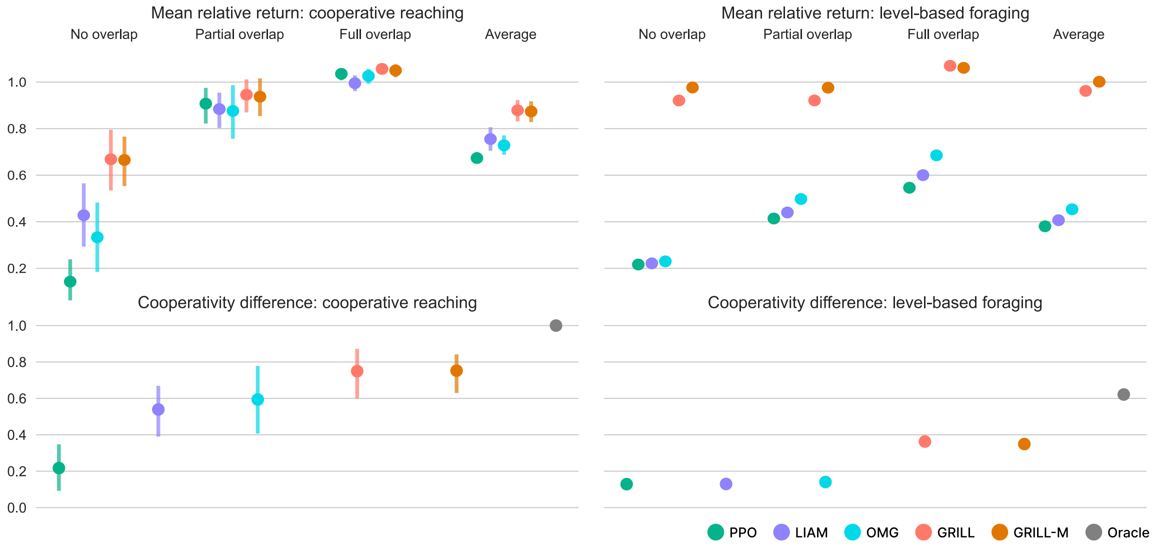

Figure 4: **Top:** evaluation returns relative to oracle policy, averaged over 1000 episodes per scenario for each of 20 unique training runs. Error bars show bootstrapped 95% confidence intervals. **Bottom:** from the same set of evaluation episodes, the difference in proportion of non-solo goals attempted between the 'full overlap' and 'no overlap' scenarios.

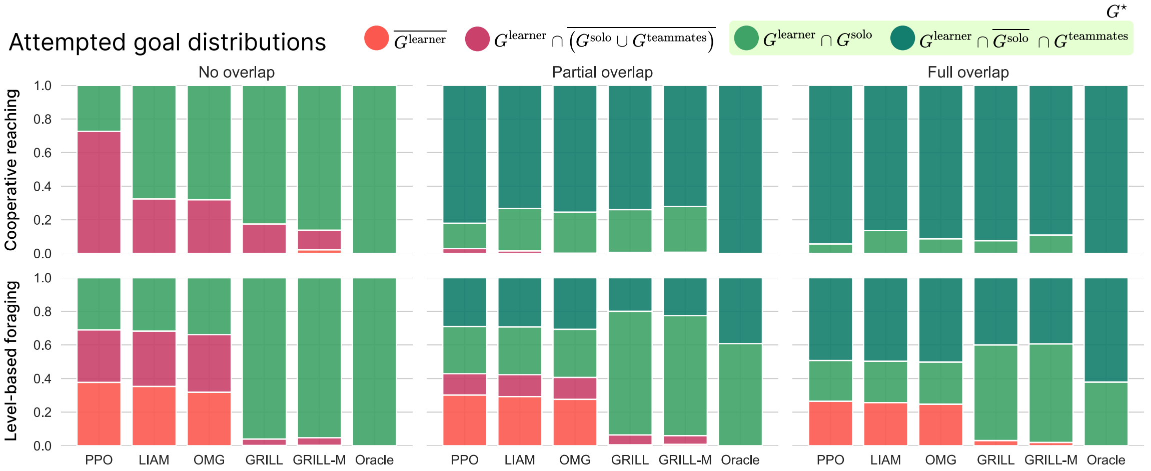

Figure 5: The distribution of goals attempted by the ego agent during evaluation. For CR, 'attempt' means the agent occupied one of the goal tiles; for LBF, it means the agent used the 'load' action while adjacent to a fruit.

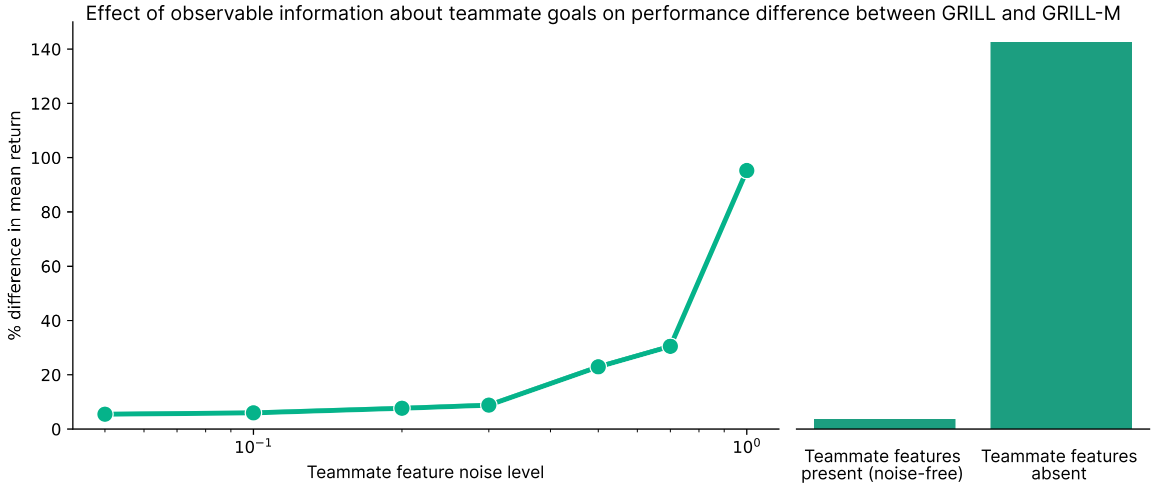

Figure 6: **Left:** the percentage increase in mean evaluation return in the LBF environment for GRILL-M vs GRILL, as a function of the amount of noise in the ego agent's observation of their teammate's goal vector. Returns are averaged over 1000 episodes × the three scenarios × 20 independent training runs. **Right:** the same increase compared for the extreme cases of no noise and teammate goal information entirely absent.

large difference between PPO and LIAM. Across all scenarios in cooperative reaching, we find no discernable difference between GRILL and GRILL-M. On the harder environment of LBF (right), we observe a striking gap in performance between GRILL and the two baselines across all scenarios. Interestingly, here we do see a difference between the two variants of our method, with GRILL-M eking out a small advantage in the no and partial overlap scenarios.

GRILL seeks more worthwhile goals

Figure 5 shows, for both environments, the distribution of goals attempted by the trained ego agent during evaluation—where we have partitioned the goal space into four subsets designed to illuminate the three failure modes outlined in Section . In cooperative reaching, all agents were able to avoid the first failure mode, seeking only goals that would be rewarding if achieved. In the no-overlap scenario, all agents to varying degrees fell victim to the second failure mode, attempting cooperative goals even though it was futile to do so. This is especially pronounced for the PPO agent, which chose the correct tile less than 30% of the time. In the partial- and full-overlap scenarios, the goal distributions were very similar across all four methods, with the agent being *under*-cooperative relative to the optimal policy. Turning to LBF, both PPO and LIAM did not avoid the first failure mode across all three scenarios, in addition to the second failure mode in no- and partial-overlap, and the third failure mode in partial- and full-overlap. By contrast, GRILL avoids the first failure mode entirely, and the second almost entirely, seeking worthwhile goals over 90% of the time across all three scenarios. However, it does still suffer from being under-cooperative in the partial and full overlap settings; and on this particular measure is actually slightly worse than both PPO and LIAM. Overall, agents trained using our method(s) selected a higher proportion of worthwhile goals than any baseline method, across both environments.

The bottom row of Figure 4 shows the 'cooperativity difference' Δ_{coop} for each method, i.e. the difference between how much the ego agent attempts cooperative goals between the full- and no-overlap scenarios. Consistent with the performance results, we find a higher Δ_{coop} value for GRILL than the two baselines across both environments, and a higher Δ_{coop} for LIAM than PPO only for cooperative reaching.

GRILL-M outperforms GRILL when teammate goal information is noisier

Finally, we ran a small additional experiment to try and understand the pattern of results observed between the two variants of our method. Our hypothesis was that in CR, having access to information about teammate goals in the observation would be less important to performance, since the teammate's goals are more clearly evidenced by their behaviour. In LBF, where the relationship between goals and actions is less trivial, having this information in the observation is more useful. If the latent teammate representation captures information about their goals, then it should aid performance to the extent that that information is not redundant with the base observation. We ran training and evaluation for GRILL and GRILL-M on the LBF environment, increasing the level of noise in the teammate's observable feature ϕ from 0.05 (the value used in our previous runs) up to 1.0. We also ran a more extreme comparison where we removed ϕ from the observation entirely. Figure 6 shows that as the noise increases, the performance gap widens—with GRILL-M achieving X% higher average return at $\sigma^2 = 1.0$, up from Y% at $\sigma^2 = 0.05$. This increases further to $Z\%$ when ϕ is removed entirely from the observation space. These results support our hypothesis about when the auxiliary latent representations are beneficial

Discussion

References

Albrecht, S. V.; Crandall, J. W.; and Ramamoorthy, S. 2016. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235: 63–94.

Albrecht, S. V.; and Ramamoorthy, S. 2013. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, 1155–1156.

Bacon, P.-L.; Harb, J.; and Precup, D. 2017. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 1726–1734. AAAI Press.

Barreto, A.; Borsa, D.; Hou, S.; Comanici, G.; Aygün, E.; Hamel, P.; Toyama, D.; Mourad, S.; Silver, D.; and Precup, D. 2019. The option keyboard: Combining skills in reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Barrett, S.; Stone, P.; and Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, 567–574. International Foundation for Autonomous Agents and Multiagent Systems.

Colas, C.; Karch, T.; Sigaud, O.; and Oudeyer, P.-Y. 2022. Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. *J. Artif. Int. Res.*, 74.

Dayan, P.; and Hinton, G. E. 1992. Feudal Reinforcement Learning. In Hanson, S.; Cowan, J.; and Giles, C., eds., *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.

Ding, Y.; Florensa, C.; Phielipp, M.; and Abbeel, P. 2020. Goal-conditioned Imitation Learning. ArXiv:1906.05838 [cs].

Ghosh, D.; Gupta, A.; Reddy, A.; Fu, J.; Devin, C.; Eysenbach, B.; and Levine, S. 2020. Learning to Reach Goals via Iterated Supervised Learning. ArXiv:1912.06088 [cs].

Gmytrasiewicz, P. J.; and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *J. Artif. Int. Res.*, 24(1): 49–79.

Henrich, J. 2018. *The Secret of Our Success: How Culture Is Driving Human Evolution, Domesticating Our Species, and Making Us Smarter*. Princeton University Press.

Kaelbling, L. P. 1993. Learning to Achieve Goals. In *International Joint Conference on Artificial Intelligence*.

Klissarov, M.; and Precup, D. 2021. Flexible Option Learning. In *Advances in Neural Information Processing Systems*, volume 34, 4632–4646.

Liemhetcharat, S.; and Veloso, M. 2014. Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artificial Intelligence*, 208: 41–65.

Liu, M.; Zhu, M.; and Zhang, W. 2022. Goal-Conditioned Reinforcement Learning: Problems and Solutions. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 5502–5511. International Joint Conferences on Artificial Intelligence Organization.

Liu, X.; Li, P.; Yang, W.; Guo, D.; and Liu, H. 2024. Leveraging Large Language Model for Heterogeneous Ad Hoc Teamwork Collaboration. arXiv:2406.12224.

Lynch, C.; Khansari, M.; Xiao, T.; Kumar, V.; Tompson, J.; Levine, S.; and Sermanet, P. 2019. Learning Latent Plans from Play. ArXiv:1903.01973 [cs].

Mirsky, R.; Carlucho, I.; Rahman, A.; Fosong, E.; Macke, W.; Sridharan, M.; Stone, P.; and Albrecht, S. V. 2022. A Survey of Ad Hoc Teamwork Research. ArXiv:2202.10450 [cs].

Papoudakis, G.; and Albrecht, S. V. 2020. Variational autoencoders for opponent modeling in multi-agent systems. *arXiv preprint arXiv:2001.10829*.

Papoudakis, G.; Christianos, F.; and Albrecht, S. V. 2021. Agent modelling under partial observability for deep reinforcement learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21.

Rabinowitz, N.; Perbet, F.; Song, F.; Zhang, C.; Eslami, S. M. A.; and Botvinick, M. 2018. Machine Theory of Mind. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4218–4227. PMLR.

Rahman, A.; Carlucho, I.; HÃ¶pner, N.; and Albrecht, S. V. 2023. A General Learning Framework for Open Ad Hoc Teamwork Using Graph-based Policy Learning. *Journal of Machine Learning Research*, 24(298): 1–74.

Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-Conditioned Imitation Learning using Score-based Diffusion Policies. ArXiv:2304.02532 [cs].

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Sundaresan, P.; Vuong, Q.; Gu, J.; Xu, P.; Xiao, T.; Kirmani, S.; Yu, T.; Stark, M.; Jain, A.; Hausman, K.; Sadigh, D.; Bohg, J.; and Schaal, S. 2025. RT-Sketch: Goal-Conditioned Imitation Learning from Hand-Drawn Sketches. In *Proceedings of The 8th Conference on Robot Learning*, 70–96. PMLR.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1): 181–211.

Tomasello, M.; Melis, A. P.; Tennie, C.; Wyman, E.; and Herrmann, E. 2012. Two key steps in the evolution of human cooperation: The interdependence hypothesis. *Current Anthropology*, 53(6): 673–692.

Yu, X.; Jiang, J.; and Lu, Z. 2024. Opponent Modeling based on Subgoal Inference. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 60531–60555.