Latent Goal Allocation for Multi-Agent Goal-Conditioned Self-Supervised Imitation Learning

Rui Chen[*] **Peide Huang**[*] **Laixi Shi**[*]
Carnegie Mellon University
{ruic3,peideh,laixis}@andrew.cmu.edu

Abstract

Multi-agent learning plays an essential role in ubiquitous practical applications including game theory, autonomous driving, etc. On the other end, goal-conditioned learning attracts a surge of interests with the capability of solving a rich variety of tasks and configurations. Nevertheless, the scenarios that combine both multi-agent and goal-conditioned settings have not been considered previously, attributed to the daunting challenges of both areas. In this work, we target ***multi-agent goal-conditioned tasks***, with the objective of learning a universal policy for multiple agents to reach a set of sub-goals. This task necessitates the agents to execute differently conditioned on the assigned sub-goals. In various scenarios, it is infeasible to access direct rewards of actions and sub-goal assignment labels for each agent. Hence, we resort to imitation learning using only expert demonstrations without reward information or sub-goal assignment labels. To achieve that, we propose a probabilistic graphical model, *latent goal allocation* (LGA), which explicitly promotes the sub-goal assignment as a latent variable and generates corresponding agent actions. We conduct experiments to show that the proposed LGA outperforms existing baselines with interpretable sub-goal assignment processes.

1 Introduction

Multi-agent learning has witnessed a wave of strong interest due to the pervasive practical applications including multi-robot control Matignon et al. (2012), game theory Mnih et al. (2015), autonomous driving (Panait and Luke, 2005; Shalev-Shwartz et al., 2016), etc. This task aims at learning a policy for each agent in a multi-agent environments Lowe et al. (2017). Notably, compared to the training of a single-agent policy, multi-agent learning usually suffers from extra challenges including the non-stationary environment for each agent induced by other agents' actions, high variance of the gradient of policies, and extremely high-dimensionality of the state and action space, etc Lowe et al. (2017).

On the other end, *goal-conditioned* tasks are garnering a flurry of interest, with various application scenarios in robotics including robot navigating, pick-and-place, in-hand manipulation Ding et al. (2019), etc. Goal-conditioned tasks, referring to the problem of learning a universal policy for any goal-reaching task upon demand, endows agents with a rich variety of abilities. Different from targeting a fixed goal, goal-conditioned tasks are more *sample-starved* with respect to both quantity and diversity for agents to generalize to unseen goals. Nevertheless, the targeted tasks were mainly focused on single-agent cases, with the objective of learning a policy for one agent (Kaelbling, 1993; Parascandolo et al., 2020; Teh et al., 2017; Ding et al., 2019; Schaul et al., 2015).

In this work, we combine both settings and target *multi-agent goal-conditioned* (MAGC) tasks, which has not been considered previously to the best of our knowledge. We introduce the formulation of MAGC by extending single-agent goal-conditioned tasks (Ding et al., 2019; Schaul et al., 2015), where we represent the overall goal for multi-agents to reach as a set of *sub-goals*. At each time step,

[*] Equal contribution

Bayesian Deep Learning workshop, NeurIPS 2021.

each agent usually only focuses on a *sub-goal* and interacts with other agents without communication. Fig 1a illustrates one example of MAGC tasks, where the goal is to reach all the three landmarks and each agent is supposed to reach a different landmark (sub-goal) without colliding with other agents. Our objective is to learn a universal policy for each agent conditioned on any assigned sub-goal. A real-life example of MAGC task would be the firefighting operations. While individual firefighter is able to perform various duties such as supplying water, putting out fires or rescuing, during actual operations, behaviors of each firefighter would be driven by his/her assigned tasks.

Targeting MAGC tasks, it is difficult to interact with the environment and have access to a direct reward function and sub-goal labels for supervision Le et al. (2017). Thus, to learn a desired goal-conditioned policy, we resort to imitation learning (IL) using only expert demonstrations without sub-goal assignment labels and reward. One example of such expert demonstrations is illustrated in Fig. 1b, targeting the task in Fig. 1a. Although IL has been widely exploited in both goal-conditioned tasks (Kaelbling, 1993; Parascandolo et al., 2020; Teh et al., 2017; Ding et al., 2019; Schaul et al., 2015) and multi-agent tasks Le et al. (2017), the lack of sub-goal assignment labels in expert data makes it infeasible to directly extend existing supervised learning for MAGC methods. As a result, it leads to a drastically challenging learning problem with incomplete labels. Specifically, since agents don't have access to their assigned sub-goal during training, we need to learn a goal-conditioned policy from demonstration without observing the goal conditioned on. Going back to the firefighting example, as a bystander, we do not know the tasks assigned to each firefighter, but still may want to learn a goal-conditioned policy from what we observe, i.e., behaviors of firefighters.

With that in hand, we propose a probabilistic graphical model named latent goal allocation (LGA), which explicitly treat the goal-conditioned policy as a latent generative process. We promote the sub-goal assignment as a latent variable and generate the subsequent action execution policy, conditioned on the inferred sub-goal assignment. Consequently, we are able to train the universal goal-conditioned policy without the labels of sub-goal assignments in expert demonstrations.

In summary, our main contributions are as follows:

- We provide a formulation of *multi-agent goal-conditioned* (MAGC) tasks in which agent behaviors are driven by some assigned sub-goals.
- We propose a multi-agent goal-conditioned imitation learning framework that models the agents' policy as a generative process, called *latent goal allocation* (LGA). Targeting tasks of recovering agent policies from demonstrations without sub-goal assignment labels, our framework significantly outperforms the baseline.

2 Problem Formulation

Multi-agent goals We firstly introduce the goal space for multi-agent tasks. We suppose the goal for N agents can be represented as a set of K sub-goals $G = \{G_k\}_{k=1}^K$, for some integer $K > 1$, where each $G_k \in \mathcal{G}_k$ denote the high-level information of the k-th kind of sub-goals in this multi-agent task for $k = 1, \cdots, K$. Here, each $\mathcal{G}_k$ denotes the space of all possible sub-goals of the k-th kind (WLOG, we assume homogeneous sub-goal spaces $\mathcal{G}_1 = \mathcal{G}_2 = \cdots = \mathcal{G}$). Various single-agent goal-conditioned tasks Ding et al. (2019); Schaul et al. (2015) regard a sub-goal as a state within the state space to reach. Differently, we allow more general sub-goal represented by any high-level information Schaul et al. (2015), such as the set of possible 2-D locations of the landmarks in Fig. 1a.

Basics of Markov games We consider partially observable Markov games Zhang et al. (2021); Littman (1994); Lowe et al. (2017), as multi-agent generalization of MDPs. A partially observable Markov game is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{O}_i\}_{i=1}^N, \{\mathcal{A}_i\}_{i=1}^N, \{R_i\}_{i=1}^N, \gamma)$, where $\mathcal{N} = \{1, 2, \cdots, N\}$ denotes the set of $N > 1$ agents, $\gamma \in (0, 1]$ is the discount factor, and $\mathcal{S}$ denotes the state space describing the possible configurations of all N agents. $\mathcal{O}_i$ and $\mathcal{A}_i$ denote the space of observation and action for the i-th agent respectively, for $i = 1, 2 \cdots, N$. In the goal-conditioned settings, the reward and policy for N agents are also conditioned on the given set of sub-goals G. At each time step, each agent i receives a private observation $o_i \in \mathcal{O}_i$ and a immediate reward $R^i \ : \mathcal{S} \times \mathcal{A}_i \times \mathcal{G} \to [0, 1]$. A goal-conditioned policy of each agent i is represented by $\pi_i : \mathcal{O}_i \times \mathcal{G} \to \mathcal{A}_i$, so that $\pi_i(\cdot|o_i, g_i)$ specifies which action to execute given the current observation o_i and sub-goal g_i.

Multi-agent goal-conditioned tasks In this work, we focus on multi-agent tasks which need to be solved by multi-agents cooperatively, competitively or both. MAGC tasks aim at learning policies $\{\pi_i\}_{i=1}^N$ for N agents respectively for each to reach the given sub-goal $g_i \in \mathcal{G}$ and interact with other agents Panait and Luke (2005). At each time step, each i-th agent usually focuses on a certain sub-goal $g_i \in G = \{G_k\}_{k=1}^K$. Without loss of generality, we consider *homogeneous* agents, meaning that the agents play interchangeable roles in the team. Agent actions are only determined by the observation o_i and assigned sub-goal g_i, but independent of the agent's identity. Hence, all agents are expected to share a same policy $\pi = \pi_1 = \pi_2 = \cdots = \pi_N$.

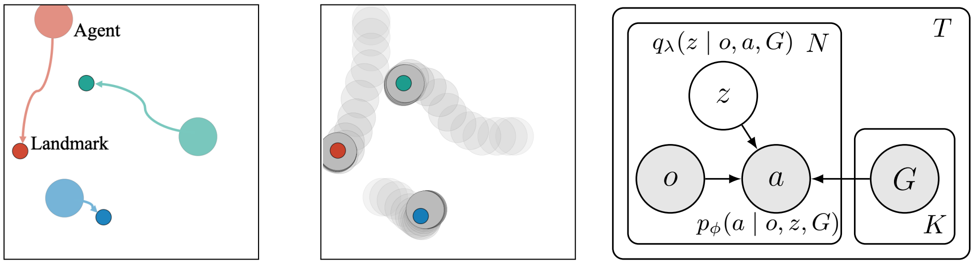
Assumptions We shall resort to imitation learning to solve MAGC tasks, using demonstrations without sub-goal assignment labels. Let $\tau := \left(\{o_i^1\}_{i=1}^N, \{a_i^1\}_{i=1}^N, G^1, \{o_i^2\}_{i=1}^N, \{a_i^2\}_{i=1}^N, G^2, \cdots\right)$ denote an entire state-action-goal trajectory of N agents, where the superscripts denote time steps, and $G^t = \{G_k^t\}_{k=1}^K$ denote the set of sub-goals to reach at the t-th time step. We assume access to a set of demonstrations (trajectories) $\mathcal{D}_{\text{expert}}$ with cardinality M. Each trajectory in $\mathcal{D}_{\text{expert}}$ is with horizon length H, collected by an expert attempting to reach the set of sub-goals $\{G^1, G^2, \cdots G^H\}$. Concatenating all the trajectories with total length $T = HM$, we arrive at the expression as follows:

$$\mathcal{D}_{\text{expert}} := \left\{ \left(\{o_i^t\}_{i=1}^N, \{a_i^t\}_{i=1}^N, G^t\right) \right\}_{t=1}^T. \tag{1}$$

Objective Our ultimate goal is to learn a universal policy π for all N agents to choose the action conditioned on the current state and their own sub-goal, i.e., $\pi(\cdot|o_i, g_i)$. However, the demonstrations from the expert only have the set of sub-goals $\{G_k\}_{k=1}^K$ under chosen, while are lack of the label of sub-goal for each agent, i.e., g_i for i-th agent is unknown, $\forall i \in 1, \cdots, N$ (For instance, we only have locations of all sub-goals in demonstration in Fig. 1b, while lacking the matching of agent and sub-goal pairs described by colors in Fig. 1a.) Therefore, to learn the policy π (in the training stage), for each i-th agent, we only have a bunch of data pairs $\left(o_i^t, G^t = \{G_k^t\}_{k=1}^K, a_i^t\right)_{t=1}^T$, yielding the objective of MAGC tasks as the following optimization problem:

$$\min_{\phi} \quad \mathcal{L}(\phi) = \mathbb{E}_{\left(\{o_i^t\}_{i=1}^N, \{a_i^t\}_{i=1}^N, G^t\right) \sim \mathcal{D}_{\text{expert}}} \left[\sum_{i=1}^N \left\| \pi_\phi(\cdot|o_i^t, G^t) - a_i^t \right\|_2^2 \right], \tag{2}$$

where π is parameterized by ϕ. The objective is to mimic the behavior of the expert by minimizing the Euclidean distance between the actions taken by the policy π_ϕ and the expert.



(a) goal-conditioned navigation (b) expert demonstration (c) LGA

Figure 1: (a) targeted task illustration. (b) expert demonstrations without information about the assigned sub-goals. (c) graphical model of LGA. The agent action a is generated conditioned on G_z where z is the inferred sub-goal index. Shaded variables are observed.

3 Self-Supervised Learning with Latent Goal Allocation

3.1 Introduction of latent goal

To begin, one key observation of the MAGC task setting is that, in each time step t, the assigned sub-goal $g_i^t \in \{G_k^t\}_{k=1}^K$ for each i-th agent is unannotated, while being an essential variable for the universal policy $\pi(\cdot|o_i^t, g_i^t)$ to choose a reasonable action. To address this challenge, we resort to self-supervised learning where we infer the sub-goal assignment also from data in addition to learning the universal goal-conditioned policy simultaneously. To specify, let z_i^t denote the sub-goal assignment index to infer for the i-th agent at the time step t, namely, $g_i^t = G^t_{z_i^t}$. Note that the unknown sub-goal assignment index for each agent is highly uncertain, we turn to consider the probability distribution of the sub-goal assignment index z_i^t by estimating a posterior distribution $p(z_i^t|o_i^t, a_i^t, G^t)$.

With the expression of the distribution over the sub-goal index $z_t^i \sim p(\cdot|o_i^t, a_i^t, G^t)$ and the corresponding sub-goal $g_i^t = G^t_{z_i^t}$, we rewrite (2) and arrive at the following optimization problem by taking expectation over all possible sub-goal selections as

$$\min_{\phi,p} \mathcal{L}(\phi,p) = \mathbb{E}_{(\{o_i^t\}_{i=1}^N, \{a_i^t\}_{i=1}^N, G^t)\sim\mathcal{D}_{\text{expert}},\; z_i^t\sim p(\cdot|o_i^t, a_i^t, G^t)} \left[\sum_{i=1}^{N} \left\|\pi_\phi\left(o_i^t, G^t_{z_i^t}\right) - a_i^t\right\|_2^2\right]. \qquad (3)$$

However, we observe (3) can't be solved directly since the posterior distribution $p(\cdot|o_i^t, a_i^t, G^t)$ is unknown or computationally intractable . To proceed, we view the task as a probabilistic generative process and treat z_i^t as a latent variable to generate the action a_i^t with the current observation o_i^t. Therefore, we propose a probabilistic graphical model, named latent goal allocation (LGA), to describe the generative process of the goal-conditioned action, illustrated in Fig. 1c. As a result, using D_{expert}, we can solve (3) by inferring the posterior of z_i^t and learning the generative action policy π_ϕ simultaneously.

3.2 Latent goal allocation model

The key structure inside LGA is the latent variable z, which represents the sub-goal assignment index. At any time step t, for the i-th agent, we are capable of inferring the posterior of the underlying sub-goal assignment index z_i^t from the data, $\forall i \in 1, \cdots, N$. Subsequently, we can utilize the estimated assigned sub-goal $g_i^t = G^t_{z_i^t}$ to pair the trajectory of each agent with the correct assigned sub-goal to learn the goal-conditioned policy.

To describe the generative process of LGA, we first introduce some notations for simplicity. We rewrite the expert data over T time steps as $\mathcal{D}_{\text{expert}} =: \{\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{G}\}$, where $\boldsymbol{o} = \{\{o_i^t\}_{i=1}^N\}_{t=1}^T$ (resp. $\boldsymbol{a} = \{\{a_i^t\}_{i=1}^N\}_{t=1}^T$) denote the set of observations (resp. observed actions) of N agents at all T time steps, and $\boldsymbol{G} = \{\{G_k^t\}_{k=1}^K\}_{t=1}^T$ encodes K sub-goals from all time steps. In the generative process, at time step t, for each agent i, we sample a sub-goal assignment index $z_i^t \in [K]$ from a fixed multinomial prior $p(z)$ with parameter $\theta \in \mathbb{R}^K$. Given z_i^t, the observed action a_i^t is sampled from a policy network with Gaussian distribution $\mathcal{N}\left(\mu_\phi(o_i^t, z_i^t, G^t), \Sigma_\phi(o_i^t, z_i^t, G^t)\right)$, where the mean and the covariance matrix are determined by a generative decoder f_ϕ parameterized by ϕ.

To proceed with the training process of LGA, we recall that the required posterior distribution $p(\boldsymbol{z}|\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{G})$ is computationally intractable. Therefore, we use variational expectation-maximization (VEM) to approximate the posterior of latent variable $\boldsymbol{z} = \{\{z_i^t\}_{i=1}^N\}_{t=1}^T$ and learn model parameter ϕ simultaneously. To continue, we use a mean-field variational distribution $q(\boldsymbol{z})$ given by

$$q(\boldsymbol{z} \mid \boldsymbol{\lambda}, \boldsymbol{o}, \boldsymbol{a}, \boldsymbol{G}) = \textstyle\prod_{t\in[T], i\in[N]} q(z_i^t \mid \lambda_i^t, o_i^t, a_i^t, G^t) \qquad (4)$$

where $\boldsymbol{\lambda}$ is a set of variational parameters for all sub-goal indices $\boldsymbol{z}$, i.e., $\boldsymbol{\lambda} = \{\{\lambda_i^t\}_{i=1}^N\}_{t=1}^T$ where $\lambda_i^t \in \mathbb{R}^K$. The joint distribution is given by

$$p(\boldsymbol{a}, \boldsymbol{z} \mid \phi, \boldsymbol{o}, \boldsymbol{G}) = \textstyle\prod_{t\in[T], i\in[N]} p(z_i^t) p(a_i^t \mid \phi, o_i^t, z_i^t, G^t). \qquad (5)$$

Using (4) and (5), ELBO $= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda}, \boldsymbol{o}, \boldsymbol{a}, \boldsymbol{G})} \left[\log p(\boldsymbol{a}, \boldsymbol{z} \mid \phi, \boldsymbol{o}, \boldsymbol{G}) - \log q(\boldsymbol{z} \mid \boldsymbol{\lambda}, \boldsymbol{o}, \boldsymbol{a}, \boldsymbol{G})\right]$. See Appendix B for training details with VEM iterations.

4 Experiments and Evaluation

Tasks and expert demonstrations In this work, we target the goal-conditioned navigation task with $N = 3$ agents and $K = 3$ landmarks in a 2-D space. At each time t, the goal $G^t = \{G_k^t\}_{k=1}^K$ encodes the positions of all landmarks. Each i-th agent receives observation o_i^t which contains its position and velocity and the relative positions to all the other agents. Agent i also receives the 2-D position of a sub-goal $g_i^t \in \{G_k^t\}_{k=1}^K$ that it needs to navigate to. The agent then decides an action a_i^t which contains the accelerations in each of 2-D directions. The goal of each agent is to reach the sub-goal assigned to it. For example, in the task shown in Fig. 1a, if *Goal Red* is assigned to *Agent Red*, *Agent Red* has to be in close proximity to *Goal Red* to receive high reward, as the individual reward is defined as the negative distance to the assigned landmark (plus small penalty for collision). The initial positions of agents and goals are randomly generated. The expert demonstration consists of the observation o_i^t, the set of sub-goals $\{G_k^t\}_{k=1}^K$ and the actions a_i^t for all agents in all time steps, but does not contain the indices of sub-goals that agents receive.

Baseline and evaluation Without interaction with the environment during training, we compare our method with behavior cloning (BC) which learns the policy through supervised learning Pomerleau (1991) in pure offline manner. Using BC, the index of the sub-goal assigned to each agent is uniformly chosen from $(1, 2, ..., K)$ and fixed for each episode. The objective is to find a policy π_ϕ minimizing the loss

$$\mathcal{L}(\phi) = \mathbb{E}_{(\{o_i^t\}_{i=1}^N, \{a_i^t\}_{i=1}^N, G^t) \sim \mathcal{D}_{\text{expert}},\ \{z_i^t\}_{i=1}^N \sim \mathsf{Unif}(\mathsf{Perm}(N,K))} \left[\sum_{i=1}^N \left\| \pi_\phi \left(o_i^t, G_{z_i^t}^t \right) - a_i^t \right\|_2^2 \right],$$

where $\mathsf{Unif}(\mathsf{Perm}(N, K))$ denotes the uniform distribution over the set of all permutations.

We demonstrate the normalized episodic reward achieved by the proposed LGA compared to BC with respect to the numbers of given expert demonstrations in Fig. 2. Here, the total episodic reward is calculated by summing up the reward from 100 episodes, where each point is obtained by conducting with 5 random seeds. The normalized episodic reward is constructed by normalizing the total episodic reward with the performance of experts and random policies set to one and zero respectively (random policies refer to the policies determined by a randomly initialized network without training). The results in Fig. 2 show that the proposed LGA consistently and significantly outperforms BC. To visualize it clearly, we illustrate the trajectories of agents using proposed LGA (Fig. 3b) compared to the baseline BC (Fig. 3c) and the expert (Fig. 3a). It can be seen that in the trajectories in BC (Fig. 3c), the red agent fails to navigate to its assigned sub-goal (red landmark) and collides with the blue agent, while the agents guided by LGA (ours) successfully reached all assigned sub-goals, similar to the expert.

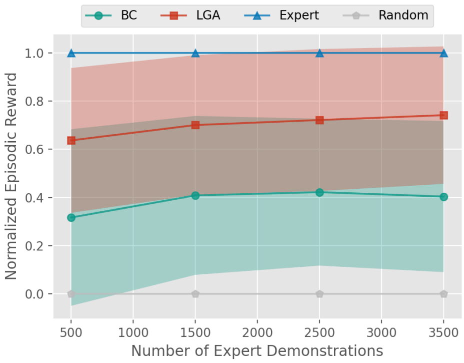
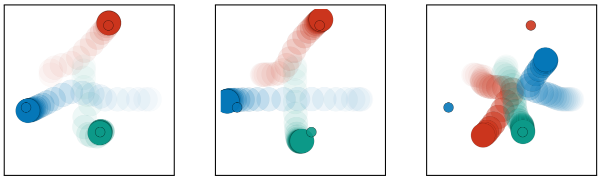


Figure 2: normalized episodic reward w.r.t. the number of given expert demonstration.



(a) expert. (b) LGA (ours). (c) behavior cloning.

Figure 3: Example trajectories generated by (a) expert, (b) LGA, and (c) behavior cloning.

5 Conclusion and Future Directions

In this work, we target a new kind of tasks named multi-agent goal-conditioned (MAGC) tasks and provide a formal formulation. During training via imitation learning, we encounter the difficulty of lacking labels of the sub-goal assignments in expert demonstrations. To address this challenge, we proposed LGA model to learn the distribution of the sub-goal assignment labels and the universal goal-conditioned policy simultaneously. In a cooperative navigation task, our model successfully inferred the unknown sub-goal labels from agent trajectories and recovered agent policies. The proposed LGA outperformed baseline method which didn't solve the sub-goal selections. For future work, we plan to explore the scalability of our approach to large-scale tasks involving more agents, dynamic goals, and high-dimensional observations.

References

Bai, C., Wang, L., Wang, Y., Wang, Z., Zhao, R., Bai, C., and Liu, P. (2021). Addressing hindsight bias in multigoal reinforcement learning. *IEEE Transactions on Cybernetics*.

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, 113(3):329–349.

Baker, C. L., Tenenbaum, J. B., and Saxe, R. R. (2007). Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

Chane-Sane, E., Schmid, C., and Laptev, I. (2021). Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR.

Ding, Y., Florensa, C., Phielipp, M., and Abbeel, P. (2019). Goal-conditioned imitation learning. *arXiv preprint arXiv:1906.05838*.

Eysenbach, B., Salakhutdinov, R., and Levine, S. (2020). C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

Jeon, W., Barde, P., Nowrouzezahrai, D., and Pineau, J. (2020). Scalable multi-agent inverse reinforcement learning via actor-attention-critic. *CoRR*, abs/2002.10525.

Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer.

Le, H. M., Yue, Y., Carr, P., and Lucey, P. (2017). Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*.

Manderson, T., Higuera, J. C. G., Wapnick, S., Tremblay, J.-F., Shkurti, F., Meger, D., and Dudek, G. (2020). Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles. *arXiv preprint arXiv:2006.16235*.

Matignon, L., Jeanpierre, L., and Mouaddib, A.-I. (2012). Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434.

Parascandolo, G., Buesing, L., Merel, J., Hasenclever, L., Aslanides, J., Hamrick, J. B., Heess, N., Neitz, A., and Weber, T. (2020). Divide-and-conquer monte carlo tree search for goal-directed planning. *arXiv preprint arXiv:2004.11410*.

Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR.

Shah, D., Eysenbach, B., Rhinehart, N., and Levine, S. (2021). Rapid exploration for open-world navigation with latent goal models. In *5th Annual Conference on Robot Learning*.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Shum, M., Kleiman-Weiner, M., Littman, M. L., and Tenenbaum, J. B. (2019). Theory of minds: Understanding behavior in groups through inverse planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6163–6170.

Song, J., Ren, H., Sadigh, D., and Ermon, S. (2018). Multi-agent generative adversarial imitation learning. *arXiv preprint arXiv:1807.09936*.

Tang, Y. and Kucukelbir, A. (2021). Hindsight expectation maximization for goal-conditioned reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2863–2871. PMLR.

Teh, Y. W., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. *arXiv preprint arXiv:1707.04175*.

Wang, M., Lan, L., and Yang, W. (2020). Multi-agent online coordination via bayesian inverse planning. In *Journal of Physics: Conference Series*, volume 1684, page 012069. IOP Publishing.

Yang, F., Vereshchaka, A., Chen, C., and Dong, W. (2020). Bayesian multi-type mean field multi-agent imitation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2469–2478. Curran Associates, Inc.

Yu, L., Song, J., and Ermon, S. (2019). Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, pages 7194–7201. PMLR.

Zhang, K., Yang, Z., and Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384.

Zhi-Xuan, T., Mann, J. L., Silver, T., Tenenbaum, J. B., and Mansinghka, V. K. (2020). Online bayesian goal inference for boundedly-rational planning agents. *arXiv preprint arXiv:2006.07532*.

Zhou, L. and Small, K. (2008). Inverse reinforcement learning with natural language goals. *ArXiv abs*.

A Related Works

We now discuss a small sample of other related works. We limit our discussions to literature regarding either multi-agent imitation learning, goal-conditioned tasks, or Bayesian inverse planning which are closest to our work.

Multi-agent imitation learning. The goal of multi-agent imitation learning (MAIL) is to recover the policies of multiple agents from pre-recorded trajectories. Our work is different from literature in multiple aspects. First, most existing work requires interactions with environments during training (Song et al., 2018; Jeon et al., 2020; Le et al., 2017; Yang et al., 2020; Yu et al., 2019), while in this work, we assume access to pre-recorded data without the actual environment. Second, we consider a specific setting where behaviors of agents are supposed to be determined by not only the state but also agent-specific time-varying goals. In this aspect, the closest work to ours is Le et al. (2017) where a latent coordination model is learned to infer the role of each agent when recovering agent policies. They assume *inherently different* agents (i.e., different positions in a soccer team), each having a different goal and policy. In contrast, we assume *exchangeable* agents that share a universal policy, while differences among agent strategies are caused solely by that in assigned goals. Besides, Yang et al. (2020) relates agents to a set of types, where the differences between types are captured only implicitly by a set of weights over value functions, without an explicit notion of goals as that in this paper.

Goal-conditioned tasks. Learning a universal policy for any goal-reaching task has been extensively studied in (Kaelbling, 1993; Parascandolo et al., 2020; Teh et al., 2017; Ding et al., 2019; Schaul et al., 2015; Manderson et al., 2020; Zhou and Small, 2008; Chane-Sane et al., 2021; Eysenbach et al., 2020; Tang and Kucukelbir, 2021; Shah et al., 2021; Bai et al., 2021). Some works focus on learning to extract information for goals from natural language or images in different environments (Manderson et al., 2020; Zhou and Small, 2008; Shah et al., 2021). Others mainly consider learning a goal-conditioned policy for a single agent (Teh et al., 2017; Tang and Kucukelbir, 2021; Chane-Sane et al., 2021; Eysenbach et al., 2020; Bai et al., 2021; Ding et al., 2019). In this work, we consider multi-agent cases by extending the single-agent formulation in (Schaul et al., 2015; Ding et al., 2019) to MAGC imitation learning problems. Note that multi-agent tasks have daunting challenges compared to single-agent case Lowe et al. (2017). Moreover, in this work, unlabeled sub-goal assignment for each agent further exacerbates the challenge of efficiently learning goal-conditioned policy from the expert data in these semi-supervised scenarios.

Bayesian inverse planning. Our work is closely related to inverse planning framework which infers an agent's intention from the agent's behavior (Baker et al., 2007, 2009), attempting to explain human being's goal inferences psychological model. One distinct difference between inverse planning and our framework is that our proposed method aims to solve imitation learning problems in which both the posterior of goal inference and agent's policy have to be jointly optimized using variational EM algorithm, while Bayesian inverse planning only need to directly apply Bayes' rule. Another work in multi-agent online coordination Wang et al. (2020) also utilizes inverse planning to infer other agents' goal by observing their past actions and achieves better coordination. Their goal is to learn a goal-conditioned policy or value function in reinforcement learning settings. Shum et al. (2019) also applies Bayesian inverse planning to infer the different types of team structure from observed agent behaviors. Zhi-Xuan et al. (2020) performs approximate online Bayesian goal inference for bounded-rational agents.

B Variational Expectation Maximization for LGA model

In this section, we provide the training process of LGA in details including the E-step and M-step updates separately.

Expectation step In E-step, VEM maximizes ELBO w.r.t. variational parameters $\boldsymbol{\lambda}$ with model parameter ϕ fixed. We use coordinate ascent variational inference (CAVI) by updating the variational parameters such that for each latent variable $j \in \{z_i^t\}_{t\in[T],i\in[N]}$,

$$q(j) \propto \exp\left\{\mathbb{E}_{q_{-j}}\left[\log p(\boldsymbol{a}, \boldsymbol{z} \mid \phi, \boldsymbol{o}, \boldsymbol{G})\right]\right\}, \tag{6}$$

where $\mathbb{E}_{q_{-j}}$ denotes the expectation over all latent variables except variable j. It can be derived that $\forall t \in [T], i \in [N]$, $q(z_i^t \mid \lambda_i^t, o_i^t, a_i^t, G^t)$ follows $\mathbf{Multi}(\lambda_i^t)$. The update rule of $\boldsymbol{\lambda}$ is derived as

$$\forall k \in [K],\ \lambda_{ik}^t \propto \theta_k \cdot \det(\Sigma_{ik}^t)^{-1/2} \exp\left\{-\frac{1}{2}(a_i^t - \mu_{ik}^t)^\top (\Sigma_{ik}^t)^{-1}(a_i^t - \mu_{ik}^t)\right\} \tag{7}$$

where $\mu_{ik}^t = \mu_\phi(o_i^t, z_i^t = k, G^t)$ and $\Sigma_{ik}^t = \Sigma_\phi(o_i^t, z_i^t = k, G^t)$.

Maximization step In M-step, VEM maximizes ELBO w.r.t. model parameters ϕ with variational parameter $\boldsymbol{\lambda}$ fixed. We solve ϕ by maximizing ELBO(ϕ) as

$$\phi = \operatorname{argmin}_\phi \textstyle\sum_{t,i,k} \lambda_{ik}^t \left(\log\det(\Sigma_{ik}^t) + (a_i^t - \mu_{ik}^t)^\top (\Sigma_{ik}^t)^{-1}(a_i^t - \mu_{ik}^t)\right) \tag{8}$$

C Implementation and Training Details

We represent the decoder function f_ϕ as fully connected neural networks with two hidden layers and 256 neurons per layer. In each E-step, we apply (7). In each M-step, we repeatedly optimize (8) via gradient methods until the improvement of generated $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ drops below 10^{-4}. For M-step, we use Adam optimizer with 5×10^{-4} learning rate. We run the entire VEM algorithm for 200 EM steps. We repeat the training process using $500, 1500, 2500, 3500$ episodes of expert demonstrations, each generating 5 policies with different random seeds.