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

Anonymous Author(s)
Affiliation
Address
email

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-goal. In various scenarios, considering it is infeasible to have access to direct rewards of actions and sub-goal assignment labels for each agent, we resort to imitation learning using only demonstrations of experts, without the need of a reward and sub-goal assignment labels. Regarding this, we propose a probabilistic graphical model, named Latent Goal Allocation (LGA), which explicitly promotes the sub-goal assignment as a latent variable to generate the corresponding action for each agent. 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 [6], game theory [7], autonomous driving [8, 12], etc. This task aims at learning a policy for each agent in a multi-agent environments [5]. Notably, compared to the training of a single-agent policy, multi-agent learning usually suffers from more challenges including 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 [5].

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 [1], 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 goal-conditioned tasks, with the objective of learning a policy for one agent [2, 9, 14, 1, 11].

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 the setting of single-agent goal-conditioned tasks [1, 11], where we represent the goal for multi-agents to reach as a set of *sub-goals*. At each time step, each agent usually only focuses on a *sub-goal* and interact with other agents. 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 collisions with other agents. The objective of MAGC is to learn a

Submitted to Bayesian Deep Learning Workshop (NeurIPS 2021).

universal policy for each agent conditioned on any assigned sub-goal. A practical example would be the expert demonstration data from firefighting operations. In such tasks, the firefighters know their specific duties and act accordingly. For instance, some firefighters may specialize in water supplying or putting out fires, while others specialize in rescuing.

Targeting MAGC tasks, in various applications, it is difficult to interact with the environment and have access to a direct reward function and sub-goal labels for supervision [3]. 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, which corresponds to the MAGC task in Fig. 1a. Although IL has been widely exploited in both goal-conditioned tasks [2, 9, 14, 1, 11] and multi-agent tasks [3], the lack of sub-goal assignment labels in expert data renders supervised learning methods infeasible. This leads to a dramatically challenging semi-supervised learning problem. Specifically, since agents don't have access to their assigned sub-goal during training, we need to learn a goal-conditioned policy from demonstration while the goal that should be conditioned on is unobserved. Going back to the firefighting example, as a bystander, we have no information about the underlying job assignments but still want to learn a goal-conditioned policy from what we observe (expert demonstrations).

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 label of sub-goal assignment in expert demonstrations.

In summary, our main contributions are as follows:

- We provide a formulation of *multi-agent goal-conditioned* (MAGC) tasks by introducing the space of a set of sub-goals, different from the scenarios of single-agent learning.
- We propose a policy as a generative process for MAGC tasks using a Bayesian deep networks named latent goal allocation (LGA) model and train it using imitation learning. Experiments are executed to show the outperformance of the proposed method with comparison to existing baselines.

2 Problem Formulation

Definitions of Multi-Agent Goals. Targeting multi-agent goal-conditioned tasks, for some positive integer $K > 1$, we suppose the goal for N agents can be represented as a set of K sub-goals $G = \{G_k\}_{k=1}^K$, where each $G_k \in \mathcal{G}_k$ denote the high-level information of the k-th kind of sub-goal in this multi-agent task, for $k = 1, \cdots, K$. Here, for any $k \in 1, \cdots, K$, $\mathcal{G}_k$ denote the space of possible the k-th kind of sub-goal, representing the high-level information of all possible tasks for each agent to solve (WLOG, we assume homogeneous sub-goal space $\mathcal{G}_1 = \mathcal{G}_2 = \cdots = \mathcal{G}$). Different from many previous single-agent goal-conditioned tasks [1, 11] where $\mathcal{G} = \mathcal{S}$, we don't limit the space of sub-goal $\mathcal{G}$ to be the state space, but the set of any high-level information of the sub-goals [11], 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 [16, 4, 5], 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 and sub-goal g_i.

Multi-Agent Goal-Conditioned Tasks In this work, we focus on multi-agent tasks which is to be solved by multi-agents cooperatively, competitively, or both. MAGC tasks aims at learning policies $\{\pi_i\}_{i=1}^N$ for N agents respectively for each to reach any given sub-goal $g_i \in \mathcal{G}$ with interaction with other agents [8], where each g_i denote the given sub-goal for the i-th agent. Regarding these tasks, at each time step, each i-th agent usually focuses on a certain sub-goal $g_i \in \{G_k\}_{k=1}^K$. Without loss of generality, we consider *homogeneous* agents indicating that the agents play an interchangeable role in the team, leading to that their actions only depend on the observation o_i and the assigned sub-goal g_i, but indistinguishable with respect to the identity. Hence, all agents are expected to share a same policy $\pi = \pi_1 = \pi_2 = \cdots = \pi_N$.

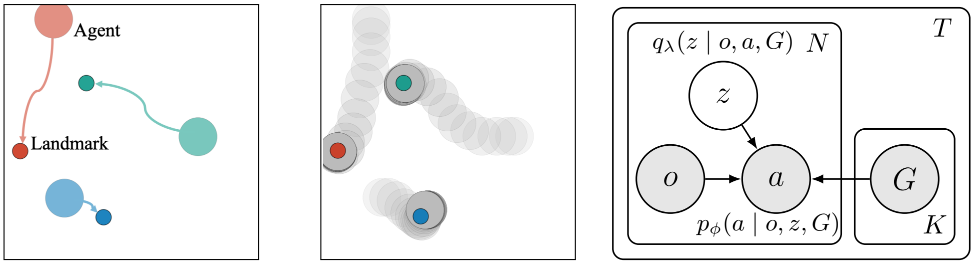
Assumptions We shall resort to imitation learning to solve MAGC tasks, using demonstrations without sub-goal assignment labels. Let $\tau := \big(\{o_i^1\}_{i=1}^N, \{a_i^1\}_{i=1}^N, G^1, \{o_i^2\}_{i=1}^N, \{a_i^2\}_{i=1}^N, G^2, \cdots\big)$ denote an entire state-action-goal trajectory of N agents, where the superscript of all variables denote the index of the time steps, and $G^t = \{G_k^t\}_{k=1}^K$ denote the set of sub-goals to reach in the t-th time step. We assume that we have 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 into a large trajectory with length $T = HM$, we arrive at the expression as follows:

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

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 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, $\forall i \in 1, \cdots, N$. Therefore, to learn the policy π (in the training stage), for each i-th agent, we only have a bunch of data pairs $\big[(o_i^t, G^t = \{G_k^t\}_{k=1}^K), a_i^t\big]$, yielding the objective of MAGC tasks as the following optimization:

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

where π is parameterized by θ. The goal is to mimic the behavior of the expert by minimizing the Euclidean distance between the actions taken by the policy π_θ and the expert.



(a) Goal-conditioned Navigation (b) Expert Demonstration (c) LGA

Figure 1: (a) illustrates the agents and the landmarks. (b) shows that the expert demonstrations contain no information about the assigned goal. (c) depicts the graphical model of LGA. The trainer first infers a sub-goal with index z for each agent in each step. Then, the agent generates action a conditioned on g_z (bottom). Shaded variables are observed. Both variational posterior and generative model are marked.

3 Learning with Self-Supervision

3.1 Introduction of laten goal

One key observation of the MAGC tasks 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 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. To specify, let z_i^t denote the sub-goal assignment index for the i-th agent in the time step t, namely, $g_i^t = G_{z_i}^t$. Note that the unknown sub-goal assignment index for each agent is highly uncertain, we turn to consider the probability distribution of sub-goal assignment index z_i^t by estimating a posterior distribution $p(z_i^t|o_i^t, a_i^t, G^t)$.

To continue, we shall introduce the approach to estimate the posterior distribution $p(z_i^t|o_i^t, a_i^t, G^t)$. Plugging in the expression of the assigned sub-goal $g_i^t = G^t_{z_i^t}$ with respect to the sub-goal assignment label $z_t^i \sim p(\cdot|o_i^t, a_i^t, G^t)$, we rewrite (2) and arrive at the following optimization problem by taking expectation over all possible sub-goal selections as

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

However, we observe that (3) cannot be optimized directly because the posterior distribution $p(\cdot|o_i^t, a_i^t, G^t)$ is unknown and 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. With this generative process in hand, using D_{expert}, we can solve (3) by inferring the posterior of z_i^t and learning the generative action policy π_ϕ simultaneously. Therefore, we propose a probabilistic graphical model, named latent goal allocation (LGA), to express the generative process of the goal-conditioned action.

3.2 Latent Goal Allocation

We propose latent goal allocation (LGA) model (Fig. 1c) to learn the goal-conditioned policy without labels of sub-goal assignment indexes. The key structure inside LGA is the latent variable z representing the sub-goal assignment index. Armed with a learned LGA, 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 for recovering the goal-conditioned policies.

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 $\boldsymbol{o} = \{\{o_i^t\}_{i=1}^N\}_{t=1}^T$ denote the set of observations of N agents, $\boldsymbol{a} = \{\{a_i^t\}_{i=1}^N\}_{t=1}^T$ represent the set of the observed actions of N agents, and $\boldsymbol{G} = \{\{G_k^t\}_{k=1}^K\}_{t=1}^T$ encodes K sub-goals from all time steps. The generative process is as follows. At time step t, for each agent i, the trainer samples a sub-goal assignment index $z_i^t \in [K]$ from a fixed multinomial prior 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 the training process of LGA, we remind the reader that the required posterior distribution $p(\boldsymbol{z}|\boldsymbol{o}, \boldsymbol{o}, \boldsymbol{G})$ 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

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

where $\boldsymbol{\lambda} = \{\{\lambda_i^t\}_{i=1}^N\}_{t=1}^T$ 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

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

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 detailed update rule in VEM iterations.

4 Experiments

Task Description In the goal-conditioned navigation task, there are $N = 3$ agents and $K = 3$ landmarks in 2-D space. The goals $G^t = \{G_k^t\}_{k=1}^K$ encode the positions of all landmarks. At time t, each agent i 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, but the index of g_i^t among $\{G_k^t\}_{k=1}^K$ is unknown. The agent then generates 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 received 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.

Expert Demonstration The expert demonstration contains 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 index of sub-goals that agents receive. Namely, we do not know the value g_i^t or its index among $\{G_k^t\}_{k=1}^K$. We aim to learn a goal-conditioned policy from the expert demonstration such that when the coordinate of a sub-goal is provided to the agent by some black-box mechanisms at testing time, the agent can imitate the expert's behavior and navigate to that sub-goal.

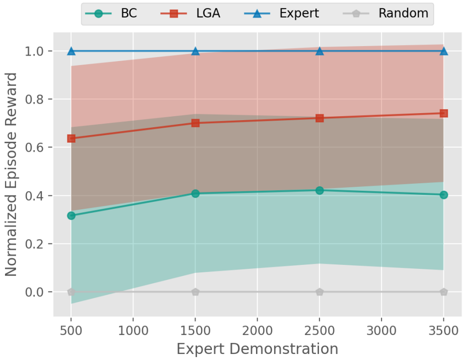
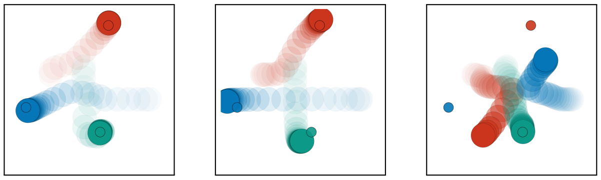


Figure 2: Episode total reward v.s. the number of expert episodes used. Performance of experts and random policies are normalized to one and zero respectively. The lines represent the mean and the shaded regions represent the standard deviation.

Evaluation of learned policies Since we do not have knowledge of transition probabilities or access to the environment (which is the case in most of the real-world applications), methods such as MA-GAIL [13] and MA-AIRL [15] are not applicable as they assume access to a black-box MDP simulator. Instead, we compare our method with Behavior Cloning (BC) which learns the policy through supervised learning [10] in pure offline manner. In BC, the sub-goal assignment sent to agents is generated by randomly permutating the sequence $(1, 2, ..., K)$ and is fixed for each episode. The objective is to find a policy π_ϕ minimizing the loss

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

where $\mathsf{Unif}(\mathsf{Perm}(N, K))$ denotes the uniform distribution over the set of all permutations. We obtain the episode total reward over 100 episodes for each of the 5 random seeds and show results in Fig. 2. It shows that LGA consistently outperform BC by large margin across different number of expert demonstrations. See Fig. 3 for qualitative examples. We can see that in the BC trajectory, the red agent fails to navigate to its assigned sub-goal (red landmark) and collides with the blue agent, while the LGA agents (ours) successfully reached all assigned sub-goals, similar to the expert.



(a) Expert. (b) LGA. (c) Behavior cloning.

Figure 3: Example trajectories from behavior cloning baseline, LGA (ours), and expert data.

5 Conclusion and Future Directions

In this work, we target a new kind of tasks named multi-agent goal-conditioned (MAGC) tasks and provide an official formulation for them. Unfortunately, resorting to imitation learning, in the training stage, we encounter the difficulty of no labels of the sub-goal assignment in the demonstrations of expert. we proposed LGA as a novel model for learning MAGC tasks from demonstrations that lack sub-goal assignment labels for each individual agent. In a cooperative navigation task, our model successfully inferred the unknown sub-goal label from agent trajectories and as a result, recovered agent policies. Our model outperformed our baseline model which didn't solve the sub-goal selections. For future work, we target at applying our approach to more complicated tasks involving more agents, dynamic goals, and high-dimensional observations. It would also be interesting to apply the learned sub-goal selection posterior to decentralized training tasks when communication between agents is expensive. The agents might only broadcast their sub-goal selections, and only communicate more information with another agent in case of shared or conflicted goals.

References

[1] Yiming Ding, Carlos Florensa, Mariano Phielipp, and Pieter Abbeel. Goal-conditioned imitation learning. *arXiv preprint arXiv:1906.05838*, 2019.

[2] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer, 1993.

[3] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR, 2017.

[4] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[5] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

[6] Laëtitia Matignon, Laurent Jeanpierre, and Abdel-Illah Mouaddib. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

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

[9] Giambattista Parascandolo, Lars Buesing, Josh Merel, Leonard Hasenclever, John Aslanides, Jessica B Hamrick, Nicolas Heess, Alexander Neitz, and Theophane Weber. Divide-and-conquer monte carlo tree search for goal-directed planning. *arXiv preprint arXiv:2004.11410*, 2020.

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

[11] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[12] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[13] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *arXiv preprint arXiv:1807.09936*, 2018.

[14] Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *arXiv preprint arXiv:1707.04175*, 2017.

[15] Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, pages 7194–7201. PMLR, 2019.

[16] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

A Related Works

We now discuss a small sample of other related works. We limit our discussions to literature regarding either multi-agent imitation or goal-conditioned settings which are closest to our work. Under multi-agent imitation learning, one work most related to us is [3] where a latent coordination model is learned to infer the role of each agent while recovering agent policies. They assume inherently different agents, i.e., different positions in a soccer team, each requiring a unique policy. In contrast, we study the case of exchangeable agents that share a universal policy, while differences between agent behaviors are caused solely by that in assigned goals. In addition, it requires a dynamical model to perform roll-out. Other works considering multi-agent imitation includes MA-GAIL [13] and MA-AIRL [15] which also require interactions with environments during training and do not consider the challenges coming from unstructured expert demonstrations. For goal-conditioned tasks, [1] proposed *goalGAIL* to achieve fast converging imitation learning, but only considers single agent case. In our work, we consider unlabeled goal assignments in multi-agent tasks which brings unique challenges.

B Variational Expectation Maximization for LGA model

In this section, we provide the training process of LGA in details by providing the E-step and M-step implementation 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 apply (8) 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.