Drone Formation Control via Belief-Correlated Imitation Learning

Bo Yang
Xidian University & Institute of
Computing Technology,
Chinese Academy of Sciences
yangbo_010@163.com

Chaofan Ma
Software College, Zhongyuan
University of Technology
chaofanma@hotmail.com

Xiaofang Xia
School of Computer Science and
Technology, Xidian University
xiaofangxia89@gmail.com

ABSTRACT

The proliferation of unmanned aerial vehicles (UAVs) has flourished various intelligent services, in which the effective coordination plays a significant role in enhancing swarm execution efficiency. However, due to the unreliable communication in the air as well as the heterogeneity in operation mode, it is challenging to achieve highly coordinated actions, particularly in the fully distributed environment with incomplete observations. In this paper, we leverage the generative adversarial imitation learning (GAIL) technique to coordinate the drones' actions by directly imitating the peer's demonstrations. In order to characterize the true environment state under local incomplete observations, we transform historical observation-action trajectories into belief representations, which are trained in conjunction with the imitation policies. We also gain regularized belief representations by correlating the prediction of future states, the trace of historical contexts, and the action-assisted guidance information, which contribute to more accurate imitation policies. We evaluate the proposed algorithm on the drones' formation control scenario. Evaluation results show the superiorities on imitation accuracy, teamwork execution time and energy cost.

KEYWORDS

Unmanned Aerial Vehicles; Formation Control; Generative Adversarial Imitation Learning

ACM Reference Format:
Bo Yang, Chaofan Ma, and Xiaofang Xia. 2021. Drone Formation Control via Belief-Correlated Imitation Learning. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Small unmanned aerial vehicles (UAVs) have been widely contributed in civil, public and military applications (e.g., air freight, traffic surveillance, search and rescue, *etc.*). Despite the agile and intelligent properties, a single UAV still suffers from constrained capabilities of sensing, coverage and execution as well as the constrained budgets of energy and payload. Accordingly, it has become a necessary trend to unite multiple UAVs to accomplish complex tasks collaboratively [10].

In the UAV swarm, reliable networked connectivity plays a significant role in exchanging control messages and coordinating actions. However, autonomous UAVs have difficulties in achieving timely information interaction and perfect action coordination due to the unstable communication in the air and the obvious heterogeneity in operation mode.

As intelligent agents, the reinforcement learning technique becomes a feasible solution for connecting with the collaborative UAVs by trial and error. However, the learning efficacy of distributed connectivity policies remains to be improved. More importantly, in an unknown or dynamic environment, it is unlikely to predefine a precise reward feedback, which is indispensible in the reinforcement learning technique [14].

One of the more intuitive and natural ways to train distributed coordination policies in the rewardless environment is to acquire the operation behaviors by directly imitating the expert's (i.e., master UAV's) demonstrations, which is termed as imitation learning [28]. Behavioural Cloning (BC) [22] and Inverse Reinforcement Learning (IRL) [4] are two typical imitation learning methods. However, the supervised BC method relies on massive training samples to match observations with actions, and suffers from the brittleness of cascading errors. While the IRL method tries to recover the reward feedbacks and then explores the policies under the reinforcement learning pattern, which is computationally expensive. To tackle the above bottlenecks, the emerging model-free Generative Adversarial Imitation Learning (GAIL) method [9][27] can learn more robust and optimized policies with fewer demonstrations, and can also scale with the input dimension. Thus, GAIL is a suitable model for imitating the operation behaviors and facilitating multi-UAV cooperation in ambiguous environments.

Most existing GAIL algorithms assume that the observed information is complete. Although there have been some works to tackle incomplete observations, they learn the imitation policy in a separated way and overlook the correlated temporal relationship between observations and actions. As the drone network is highly dynamic and uncertain, we characterize the observation-action trajectory as a latent *belief representation* [7][20], which is utilized to predict the correlated joint distributions of observations/actions and to optimize the imitation policy. The contributions are summarized as follows.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

- In the challenging partially observable imitation environment, we keep track of the true state information by transforming historical observation trajectories into latent *belief representations*, which are trained in conjunction with imitation policies so as to minimize the imitation errors.
- To tackle the mediocre expressiveness capability of belief representation, we correlate the potential forward trajectory with the historical backward trajectory to yield informative and robust belief representations. We also utilize the partially useful actions to guide more accurate imitation policies.
- We evaluate the algorithm performance in the formation control scenario. Results show that the slave UAVs can timely adjust the flight trajectory under the master UAV's demonstrations, and the execution efficiency of the collaborative formation is improved evidently.

2 RELATED WORK

It has been a long-term goal to facilitate the collaboration among multiple autonomous agents in decentralized environments. Existing efforts mainly adopt the reinforcement learning technique due to its advantage on joint-policy learning. However, most works focus on centralized training with decentralized execution [5][15][16]. It is difficult to train large-scale samples in the centralized phase as the number of agents increases. As an improvement, [11] presents the message-dropout technique to reduce the dimension of action spaces. [17] investigates the rapid information sharing issue on both learning and coordination in a partially observable environment. However, the mentioned works can only support a limited number of agents.

As for the scenario of distributed training with distributed execution, [19] incorporates a decentralized experience replay with a generalized recurrent multi-task network to achieve multi-agent coordination. The FSMS algorithm [30] adopts a cross-network mutual sampling and joint learning technique. However, reinforcement-learning-based methods involve tough reward engineering [29]. The yielded policy is greatly influenced by the reward signal, which cannot be explicitly predefined in ambiguous environments.

As a countermeasure, the imitation-learning-based methods aim to collect information from the expert demonstrations and learn the imitation policy without any reward feedback. Until now, Generative Adversarial Imitation Learning (GAIL) [9][27] has emerged as a significant technique. For examples, the multi-modal imitation leaning method [8] learns a multi-modal policy that can imitate all sub-tasks in order to grasp a set of different skills. The Directed-Info GAIL algorithm [25] models the imitation learning as a directed graph and infers the latent sub-task policies directly from the unstructured expert demonstrations. Considering that expert demonstrations may be inaccessible, the ILPO algorithm [2] learns latent policies only from observations of the expert's states with very few environment interactions.

Besides the theoretical studies, the GAIL model also has widespread applications. For examples, the MA-GAIL method [3] captures the sequential dependency relationship among the sub-tasks of expert demonstrations and thus can adapt to the dynamics in driving. The xGAIL method [21] learns global and local explainable spatial-temporal features to optimize driver's passenger-seeking strategy. The Burn-InfoGAIL algorithm [13] produces driver models that can imitate experts over long time horizons by maximizing mutual information over latent representations. The RAIL algorithm [26] utilizes the augmented random search method to train optimal auto-driving policies. Its computation overhead is also reduced as compared with the DRL-based decisions.

In this paper, we try to enhance the imitation accuracy in the partially observable flying environment and facilitating the coordination among heterogeneous UAVs.

3 SYSTEM MODEL AND PROBLEM FORMULATION

We consider a multi-UAV coexisting environment, in which a swarm of autonomous UAVs fall into the communication range of each other. Each UAV is equipped with sensors and has the capability of perceiving the surrounding environment status. The UAVs can obtain their 3D-coordinates through the GPS system and adjust their flight attitudes. We consider a master-slave model in which the master UAV is aware of the flight missions, while the slave UAVs aim to keep a consistent formation with a reasonable speed and distance by following the master UAV's behaviors. The UAV swarm can execute the flying control operations (e.g., cruising, hovering, attitude adjustment, *etc.*) as well as some transmission/computing tasks (e.g., data collection and processing, trajectory planning, *etc.*).

3.1 Imitation learning model

We model the coordinated flying task as a multi-agent partially observable Markov game. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of UAVs. The agents (i.e., UAVs) are fully cooperative to execute a task. For the sequential task requirements, scenarios are broken up into a series of episodes, which include a set of states $\mathcal{S}$ characterizing the configurations of all agents (e.g., task load, residual energy) as well as the varying flight status (e.g., speed, orientation, distances with the peers), a set of observations $\mathcal{O}_1, \ldots, \mathcal{O}_N$ and a set of actions $\mathcal{A}_1, \ldots, \mathcal{A}_N$ (e.g., cruising trajectory, hovering point) for each agent. The function of $P : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \to \mathcal{P}(\mathcal{S})$ denotes the stochastic transition between the states, where $\mathcal{P}(\mathcal{S})$ represents the set of probability distributions over $\mathcal{S}$. Specifically, given the state s^t at time t, agents take actions of $(a_1, \ldots, a_N)$, then s^t transits to s^{t+1} with the probability of $P(s^{t+1}|s^t, a_1, \ldots, a_N)$. According to the action, agent i ($i \in \mathcal{N}$) attains an immediate reward $r_i : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \to \mathbb{R}$. By selecting actions through a stochastic policy $\pi_i : \mathcal{O}_i \times \mathcal{A}_i \to [0, 1]$, agent i aims at maximizing its total expected return $\sum_{t=0}^{+\infty} \gamma^t r_i^t$.

Different from the centralized methods (e.g., remote control or command broadcasting), we investigate a distributed coordination method by directly imitating the master UAV's maneuvers in order to reduce the communication latency over unstable air links and enhance the collaboration efficiency. Inspired by the Generative Adversarial Networks (GANs) [23], the GAIL model imitates the expert's policy π_E by minimizing the Jensen-Shannon divergence between the trajectory distributions (i.e., state-action pairs) generated by π_E and the imitation policy π_{θ_i} respectively. Specifically, the generator in GAIL corresponds to π_{θ_i}, while the discriminator in GAIL is a binary classifier and is trained to distinguish between π_E and π_{θ_i}, which can help enhance the imitation fidelity. For multi-agent distributed imitation learning, we need a decentralized structure, i.e., each agent i is equipped with a discriminator (denoted as D_{ω_i}) instead of sharing a common discriminator with others. The objective training function of GAIL is defined as:

$$\begin{aligned}\min_{\theta_i}\max_{\omega_i}&\mathbb{E}_{\pi_{\theta_i}}[\log D_{\omega_i}(s,a)]+\\ &\mathbb{E}_{\pi_E}[\log(1-D_{\omega_i}(s,a))]-\lambda H(\pi_{\theta_i}),\end{aligned} \tag{1}$$

where D_{ω_i} denotes the discriminator parameterized by ω_i, $H(\pi_{\theta_i}) \simeq \mathbb{E}_{\pi_{\theta_i}}[-\log \pi_{\theta_i}(a|s)]$ is an entropy regularization term, and λ is a hyper-parameter.

Although equipped with no reward, GAIL can still match the state-action distributions between the expert and the imitator. Specifically, the expert's and the i-th imitator's state-action visitation distributions are respectively given as:

$$\rho_{\pi_E(s,a)} = \pi_E(a|s)\sum_{t=0}^{+\infty}\gamma^t p(s^t=s,a^t=a|\pi_E), \tag{2}$$

and

$$\rho_{\pi_{\theta_i}(s,a)} = \pi_{\theta_i}(a|s)\sum_{t=0}^{+\infty}\gamma^t p(s^t=s,a^t=a|\pi_{\theta_i}). \tag{3}$$

3.2 Computation model

In the imitation-driven flight task, slave UAVs iteratively sense and imitate the master UAV's maneuvers by trial and error. To attain a target imitation accuracy, the learning time and energy cost of UAV i depend on its computational capability and the size of the sensed sample set χ_i. We utilize the CPU-cycle frequency f_i (in the unit of Hz) to depict UAV i's computational capability. Let W_i denote the imitation learning workload (in the unit of CPU cycles per bit) to process one sample for UAV i. The imitation learning time and energy cost can be respectively calculated as:

$$T_i^{learn} = \sum_{h=1}^{H}\frac{\sum_{i=1}^{|\chi_i|}W_i}{f_i}, \tag{4}$$

and

$$E_i^{learn} = \xi_i\sum_{h=1}^{H}\sum_{i=1}^{|\chi_i|}W_i f_i^2, \tag{5}$$

where H denotes the number of iterations to attain a target imitation accuracy, ξ_i denotes the coefficient related to the energy cost per CPU cycle and is set as 10^{-28} [31].

3.3 Communication model

UAVs need to iteratively exchange and share control messages via wireless links until the target accuracy is attained. We consider the line-of-sight (LOS) communication range. The signal-to-interference-noise ratio (SINR) at the receiver i over the air-to-air link l_i is given as:

$$\Gamma_{l_i} = \frac{P_{l_i}^{tran} d_{l_i}^{-\alpha}|h_{l_i}|^2}{P_{l_i}^{recv}+N_{l_i}^{recv}}, \tag{6}$$

where $P_{l_i}^{tran}$ denotes the transmission power over l_i, d_{l_i} denotes the distance between the transceivers over l_i, α denotes the path loss exponent, h_{l_i} denotes the channel coefficient, $P_{l_i}^{recv}$ denotes the received power of interference signal over l_i, and $N_{l_i}^{recv}$ denotes the additive white Gaussian noise at the receiver [1].

The transmission rate over l_i can be calculated as:

$$R_{l_i} = \log_2(1+\Gamma_{l_i}). \tag{7}$$

Therefore, the transmission latency and energy cost in each imitation iteration are respectively given as:

$$t_i^{tran} = \frac{Tran_{l_i}}{R_{l_i}}, \tag{8}$$

and

$$e_i^{tran} = \frac{Tran_{l_i}\cdot P_{l_i}^{tran}}{R_{l_i}}, \tag{9}$$

where $Tran_{l_i}$ denotes the volume of transmission information over l_i.

3.4 Problem Formulation

The total imitation learning time consists of the computation and communication latencies, which can be denoted as

$$T_i = T_i^{learn} + T_i^{tran} = \sum_{h=1}^{H}(\frac{\sum_{i=1}^{|\chi_i|}W_i}{f_i} + t_i^{tran}). \tag{10}$$

Similarly, the total energy cost can be denoted as

$$E_i = E_i^{learn} + E_i^{tran} = \sum_{h=1}^{H}(\xi_i\sum_{i=1}^{|\chi_i|}W_i f_i^2 + e_i^{tran}). \tag{11}$$

The imitation-based coordination objective can be formulated as:

$$\begin{aligned}&\min\ T_i,\ E_i,\ D_{JS}(\rho_{\pi_E}\parallel\rho_{\pi_{\theta_i}})\ (\forall i\in\mathcal{N}),\\ &s.t.\ \pi_E^{T_i}\simeq\pi_{\theta_i}^{T_i},\end{aligned} \tag{12}$$

where $D_{JS}(\rho_{\pi_E}\parallel\rho_{\pi_{\theta_i}})$ denotes the Jensen-Shannon divergence (i.e., similarity degree) between ρ_{π_E} and $\rho_{\pi_{\theta_i}}$.

4 BELIEF-CORRELATED IMITATION ALGORITHM

In this section, we present a Belief-Correlated Imitation (B-CI) algorithm, which includes the normal imitation policy component and the belief-enhanced component.

4.1 Training of the Policy Component

In the partially observable environment, we try to recover the true environment state s^t by utilizing the belief representation b^t (derived from historical observations). The belief visitation distributions of $\rho_{\pi_E}(b)$ and $\rho_{\pi_{\theta_i}}(b)$ are similar to those of $\rho_{\pi_E}(s)$ and $\rho_{\pi_{\theta_i}}(s)$. Proposition 1 declares that to what extent $\rho_\pi(b)$ can approximate $\rho_\pi(s)$.

Proposition 1: For agent i's imitation policy π_{θ_i} and the expert's policy π_E, the Jensen-Shannon divergences between their state- and belief-visitation distributions satisfy that

$$D_{JS}(\rho_{\pi_{\theta_i}}(s)\|\rho_{\pi_E}(s)) \leqslant D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b)).$$

Proof. Given the belief representation b, let $p(s|b)$ denote the posterior distribution over state s. $p(s|b)$ is policy-independent, and only depends on the environment. On the contrary, the posterior distribution $p(b|s)$ over b under a given state s is policy-dependent. For the expert, we derive $p_{\pi_E}(b|s)$ from the Bayes rule as

$$p_{\pi_E}(b|s) = \frac{\rho_{\pi_E}(s,b)}{\rho_{\pi_E}(s)} = \frac{p(s|b)\rho_{\pi_E}(b)}{\rho_{\pi_E}(s)}.$$

The imitator's posterior distribution $p_{\pi_\theta}(b|s)$ can be derived in a similar way.

Here, we utilize the f-divergences [18] to analyze $D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b))$. Let $f(u) = -(u+1)\log\frac{1+u}{2} + u\log u$ $(u \in \mathbb{R}^+)$ denote a convex function. Then, we can derive that

$$\begin{aligned}
&D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b))\\
=&\mathbb{E}_{b\sim\rho_{\pi_E}(b)}[f(\frac{\rho_{\pi_{\theta_i}}(b)}{\rho_{\pi_E}(b)})]\\
=&\mathbb{E}_{s,b\sim\rho_{\pi_E}(s,b)}[f(\frac{\rho_{\pi_{\theta_i}}(s,b)}{\rho_{\pi_E}(s,b)})]\\
=&\mathbb{E}_{s\sim\rho_{\pi_E}(s)}[\mathbb{E}_{b\sim\rho_{\pi_E}(b|s)}f(\frac{\rho_{\pi_{\theta_i}}(s,b)}{\rho_{\pi_E}(s,b)})]\\
\geqslant&\mathbb{E}_{s\sim\rho_{\pi_E}(s)}[f(\mathbb{E}_{b\sim\rho_{\pi_E}(b|s)}\frac{\rho_{\pi_{\theta_i}}(s,b)}{\rho_{\pi_E}(s,b)})]\\
=&\mathbb{E}_{s\sim\rho_{\pi_E}(s)}[f(\mathbb{E}_{b\sim\rho_{\pi_{\theta_i}}(b|s)}\frac{\rho_{\pi_{\theta_i}}(s,b)\rho_{\pi_E}(b|s)}{\rho_{\pi_E}(s,b)\rho_{\pi_{\theta_i}}(b|s)})]\\
=&\mathbb{E}_{s\sim\rho_{\pi_E}(s)}[f(\mathbb{E}_{b\sim\rho_{\pi_{\theta_i}}(b|s)}\frac{\rho_{\pi_{\theta_i}}(s)}{\rho_{\pi_E}(s)})]\\
=&D_{JS}(\rho_{\pi_{\theta_i}}(s)\|\rho_{\pi_E}(s)),
\end{aligned}$$

which concludes the proof.

□

From *Proposition 1*, we know that the minimum value of $D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b))$ is the upper-bound of $D_{JS}(\rho_{\pi_{\theta_i}}(s)\|\rho_{\pi_E}(s))$. However, this tight bound is intractable for computation, we thus relax this upper-bound by introducing the belief-action visitation distribution, which is declared in *Proposition 2*.

Proposition 2: For agent i's imitation policy π_{θ_i} and the expert's policy π_E, the Jensen-Shannon divergences between their belief- and action-visitation distributions satisfy that

$$D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b)) \leqslant D_{JS}(\rho_{\pi_{\theta_i}}(b,a)\|\rho_{\pi_E}(b,a)).$$

where $\rho_{\pi_{\theta_i}}(b,a) = \rho_{\pi_{\theta_i}}(b)\pi_{\theta_i}(a|b)$ and $\rho_{\pi_E}(b,a) = \rho_{\pi_E}(b)\pi_E(a|b)$.

Proof. We replace s (*resp.* b) in *Proposition 1* with b (*resp.* (b,a)). Similar with the proof in *Proposition 1*, if $p(b|b,a)$ is independent of the policy, we can obtain the expected inequality of $D_{JS}(\rho_{\pi_{\theta_i}}(b)\|\rho_{\pi_E}(b)) \leqslant D_{JS}(\rho_{\pi_{\theta_i}}(b,a)\|\rho_{\pi_E}(b,a))$.

□

Next, we describe how to minimize $D_{JS}(\rho_{\pi_{\theta_i}}(b,a)\|\ \rho_{\pi_E}(b,a))$. Following the principle of GANs, $D_{JS}(\rho_{\pi_{\theta_i}}(b,a)\|\rho_{\pi_E}(b,a))$ is approximated as:

$$\begin{aligned}
\max_{e_{\pi_{\theta_i}}(b,a),e_{\pi_E}(b,a)} &\widetilde{\mathbb{E}}_{\rho_{\pi_{\theta_i}}}[\log\frac{e_{\pi_{\theta_i}}(b,a)}{e_{\pi_{\theta_i}}(b,a)+e_{\pi_E}(b,a)}]+\\
&\widetilde{\mathbb{E}}_{\rho_{\pi_E}}[\log\frac{e_{\pi_E}(b,a)}{e_{\pi_{\theta_i}}(b,a)+e_{\pi_E}(b,a)}],
\end{aligned} \tag{13}$$

where $e_{\pi_{\theta_i}}(b,a)$ and $e_{\pi_E}(b,a)$ denote the empirical density estimators of $\rho_{\pi_{\theta_i}}$ and ρ_{π_E} respectively. $e_{\pi_{\theta_i}}(b,a)$ can be obtained by the samples of belief-action tuples (b^t,a^t), in which $b^t = B_{\psi_i}(h^t)$, h^t denotes the trajectory of (o^{t^-},a^{t^-}) before the current time instant t, and ψ_i denotes the parameter of agent i's belief representation. Similarly, $e_{\pi_E}(b,a)$ can be obtained by the expert's observation-action trajectory. In particular, the maximum value of Eq. (1) is denoted as $D^\star$.

The multi-agent training objective can be denoted as

$$\begin{aligned}
\min_{\theta_i} D_{JS}(\rho_{\pi_{\theta_i}},\rho_{\pi_E}) \approx \min_{\theta_i}\max_{\omega_i}&\widetilde{\mathbb{E}}_{\rho_{\pi_{\theta_i}}}[\sum_{i=1}^{N}\log D_{\omega_i}(b,a_i)]\\
&+\widetilde{\mathbb{E}}_{\rho_{\pi_E}}[\sum_{i=1}^{N}\log(1-D_{\omega_i}(b,a_i))].
\end{aligned} \tag{14}$$

Eq. (14) (with respect to the policy parameter π_{θ_i}) can be updated by using the following gradient:

$$\begin{aligned}
\nabla_{\theta_i}D_{JS}(\rho_{\pi_{\theta_i}},\rho_{\pi_E}) &\approx \nabla_{\theta_i}\widetilde{\mathbb{E}}_{\rho_{\pi_{\theta_i}}}[\log\frac{e_{\pi_{\theta_i}}(b,a)}{e_{\pi_{\theta_i}}(b,a)+e_{\pi_E}(b,a)}]\\
&= \widetilde{\mathbb{E}}_{\rho_{\pi_{\theta_i}}(b,a)}[\nabla_{\theta_i}\log\pi_{\theta_i}(a|b)Q_{\pi_{\theta_i}}(b,a)],
\end{aligned} \tag{15}$$

where $Q_{\pi_{\theta_i}}(b^t,a^t) = \widetilde{\mathbb{E}}_{\rho_{\pi_{\theta_i}}(b,a)}[\sum_{t'=t}^{+\infty}\gamma^{t'-t}r(b^{t'},a^{t'})]$, and the reward $r(b^{t'},a^{t'}) = \log D^\star(b,a)$.

Until now, the iterative gradient updates for the imitation policy and the discriminator are defined. Next, we describe how to update the belief representation's parameter ψ_i.

4.2 Training of the Belief Component

The belief component B_{ψ_i} records the historical observation-action trajectory $h^t = (o^{t^-}, a^{t^-})$ before the current time instant t. That is, the belief representation at time instant t can be denoted as $b^t = B_{\psi_i}(b^{t-1}, o^t, a^{t-1})$. As the belief is closely related to the policy, we thus train the belief's objective jointly with the policy. Similar with Eq. (14), the imitation learning objective w.r.t. the belief's parameter ψ_i can be denoted as

$$\begin{aligned}\min_{\theta_i,\psi_i} D_{JS} \approx \min_{\theta_i,\psi_i} \max_{\omega_i} \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\sum_{i=1}^{N} \log D_{\omega_i}\left(B_{\psi_i}(h), a_i\right)] \\ + \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_E}}[\sum_{i=1}^{N} \log\left(1 - D_{\omega_i}\left(B_{\psi_i}(h), a_i\right)\right)].\end{aligned} \tag{16}$$

The belief's parameter ψ_i in Eq. (16) is also trained by the iterative gradient updates. Consider the first expectation term in Eq. (16), as both of the policy and belief components have an impact on the historical observation-action trajectory, the corresponding gradient consists of two sub-parts. As for the second expectation term in Eq. (16), it has no derivative term regarding the parameter ψ_i. Combining with Eq. (15), the gradient on ψ_i can be approximated as

$$\begin{aligned}&\nabla_{\psi_i} D_{JS} \approx \\ &\quad \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\nabla_{\psi_i} \log \pi_{\theta_i}(a_i|B_{\psi_i}(h)) Q_{\pi_{\theta_i}}(B_{\psi_i}(h), a_i)] + \\ &\quad \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\nabla_{\psi_i} \log D^\star(B_{\psi_i}(h), a_i)] + \\ &\quad \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_E}}[\nabla_{\psi_i} \log(1 - D^\star(B_{\psi_i}(h), a_i))],\end{aligned} \tag{17}$$

and

$$Q_{\pi_{\theta_i}}(B_{\psi_i}(h), a_i) = \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\sum_{t'=t}^{+\infty} \gamma^{t'-t} \log D^\star(B^{t'}_{\psi_i}(h), a^{t'})].$$

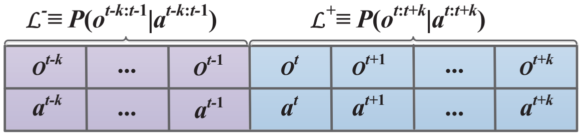
4.3 Belief Regularization

The above learning method usually discounts the historical contexts and yields trivial belief representation. In order to gain an informative belief representation with necessary historical contexts, we present a regularized belief representation, which consists of three terms (i.e., forward regularizer, backward regularizer and action-oriented regularizer).

Forward regularizer. As the posterior distribution $p(s|b)$ plays a critical role in the belief representation, we thus associate the future k ($k \in \mathbb{Z}^+$) predicted states after the current instant t with the forward regularized belief representation, as shown in Fig. 1. We aim to minimize the Euclidean distance between the forward belief representation and the future observations, and utilize the mean square error (MSE) to characterize the objective loss function, which is given as

$$\mathcal{L}^+ = \sum_{t'=t}^{t+k} \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\|B_{\psi_i}(h^{t'}) - o^{t'}\|]^2. \tag{18}$$

Backward regularizer. The historical contexts naturally involve the past environment states. We thus trace the past k observed states before the current instant t with the backward regularized belief representation, as shown in Fig. 1. Conditioned on the historical observations $o^{t-k:t-1}$, the backward belief representation is given as



$\mathcal{L}^- \equiv P(o^{t\text{-}k:t\text{-}1}|a^{t\text{-}k:t\text{-}1})$ $\mathcal{L}^+ \equiv P(o^{t:t+k}|a^{t:t+k})$

o^{t-k}	...	o^{t-1}	o^t	o^{t+1}	...	o^{t+k}
a^{t-k}	...	a^{t-1}	a^t	a^{t+1}	...	a^{t+k}

Figure 1: Illustration of the forward/backward regularizer

$$\mathcal{L}^- = \sum_{t'=t-k}^{t-1} \widetilde{\mathbb{E}}_{(h,a_i)\sim\rho_{\pi_{\theta_i}}}[\|B_{\psi_i}(h^{t'}) - o^{t'}\|]^2. \tag{19}$$

Action-oriented regularizer. The above regularizers belong to the state-based imitation learning. Guided by Proposition 2, we add the action-oriented regularizer to assist policy optimization. Here, we adopt the approximated mutual information technique to merge the actions.

Given the demonstration actions a_{π_E} and the generated actions $a_i \sim \rho_{\pi_{\theta_i}}$, there should exist rich *mutual information* (denoted as $I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}})$) between two distributions. $I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}})$ characterizes the dependence between a_{π_E} and $a_i \sim \rho_{\pi_{\theta_i}}$. That is, given the observed distribution of $a_i \sim \rho_{\pi_{\theta_i}}$, the larger the $I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}})$ is, the more certainty in a_{π_E} is. However, it is difficult to maximize $I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}})$ due to the intractability of the posterior probability $p(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})$. We thus replace $I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}})$ with a variational lower bound, i.e.

$$\begin{aligned}\mathcal{L}_I &= \mathbb{E}_{a_{\pi_E}}[\log Q(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})] + H(a_{\pi_E}) \\ &\leqslant I(a_{\pi_E}; a_i \sim \rho_{\pi_{\theta_i}}),\end{aligned} \tag{20}$$

where $Q(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})$ is a simplified posterior approximation of $p(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})$, and $H(a_{\pi_E})$ denotes an entropy regularizer that can increase the diversity over the parameter space. We parameterize $Q(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})$ with ϕ_i, and Eq. (20) can be solved by approximating a_{π_E} with a sequence of iteratively updated ϕ_i.

Above all, the ultimate objective function of the belief regularizer is formulated as

$$\mathcal{L}(\psi_i) = \zeta_1\mathcal{L}^+ + \zeta_2\mathcal{L}^- + \zeta_3 L_I, \tag{21}$$

where ζ_1, ζ_2 and ζ_3 denote the weighted parameters.

4.4 Algorithm Description

The Belief-Correlated Imitation (BCI) algorithm is formally presented in Algorithm 1.

The state-action trajectories of the expert (i.e., master UAV) and the imitator (i.e., slave UAV) are trained iteratively until convergence. In each iteration, a mini-batch of size T is sampled from the whole trajectory so as to improve the training efficiency (lines 3∼4). The belief representations and the imitation policies are trained jointly. The discriminator's parameter ω_i is updated through Eq. (16) in the ascending direction, as shown in line 5. The imitation objective w.r.t. the belief parameter ψ_i is updated through Eq. (17), as shown in line 6. To attain an informative and robust belief representation, we update the belief regularizer terms through Eqs. (18)∼(21), as shown in line 8. In particular, the action-oriented regularizer is optimized by the Adam stochastic approximation [12] (as shown in line 7), which can achieve stable training under variable learning rates.

In addition, the imitation policy is optimized by a Trust Region Policy Optimization (TRPO) step [24] (as shown in line 9). TRPO is a smooth policy gradient algorithm, which optimizes the parameterized policy π_{θ_i} by maximizing the expected return directly. Each TRPO update yields an approximate solution to the following objective:

$$\theta_i^{h+1} = \arg\min_{\theta_i} \mathbb{E} \sum_{t=1}^{T} \left[\frac{\pi_{\theta_i}(a^t|b^t)}{\pi_{\theta_i}^h(a^t|b^t)} \left(\sum_{t'=t}^{T} \gamma^{t'-t} r(b^{t'}, a^{t'}) - e^t \right) \right],$$
$$s.t.\ \mathbb{E} \sum_{t=1}^{T} [D_{JS}(\pi_{\theta_i}(a^t|b^t) \| \pi_{\theta_i}^h(a^t|b^t))] \leqslant \epsilon, \tag{22}$$

where $\pi_{\theta_i}^h$ denotes the policy induced by θ_i^h at the h-th iteration, $r(b^{t'}, a^{t'}) = \log D^\star(b, a)$ (as shown in Eq. (15)), $e^t \simeq \mathbb{E}_{\theta_i}[\sum_{t'=t}^{T} \gamma^{t'-t} r(b^{t'}, a^{t'})|\ o^{t^-}, a^{t^-}]$ (i.e., an estimation of the expected reward conditioned on the current policy), and ϵ denotes a small positive constant.

In other words, the TRPO step only allow small variations in the policy (i.e., $\pi_{\theta_i}^{h+1}$ does not deviate too much from $\pi_{\theta_i}^h$ due to pertubations), leading to stable gradient updates.

5 PERFORMANCE EVALUATION

5.1 Experiment Setup

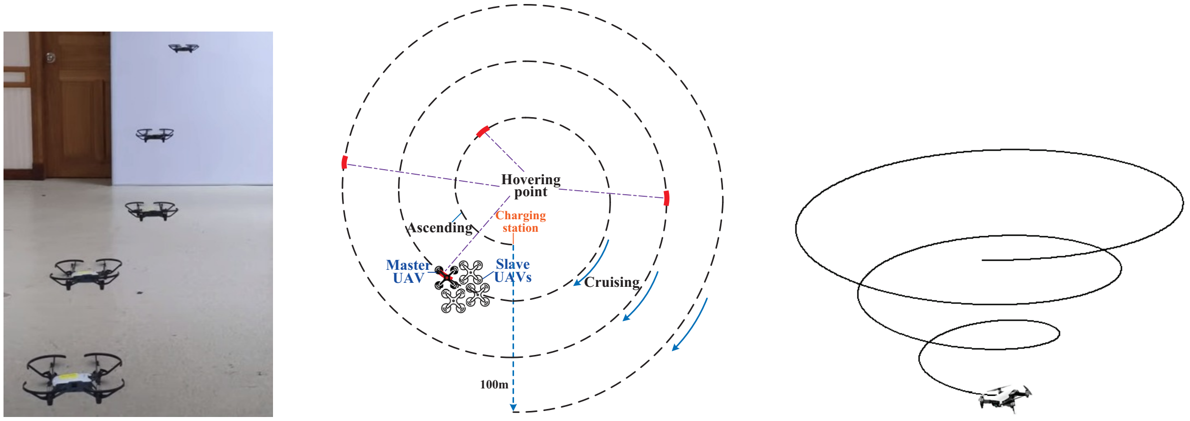
We consider a flying scenario in which a swarm of quadrotor UAVs maneuver on a spiral trajectory (as shown in Fig. 2). We deploy practical UAVs with the type of DJI M100. Considering the payload, energy, and heterogeneous requirement restrictions, the UAVs are mounted with lightweight, low-power and heterogeneous computation modules (i.e., NVIDIA Jetson TX2 and Jetson Nano). Jetson TX2 is configured with 4core 64-bit and 2 Denver A57 CPUs (2.0GHz), and Jetson Nano is configured with 4core 64-bit A57 CPUs (1.4GHz), both of which can execute imitation-learning-driven computations and interact with the UAV's DJI Software Development Kit as well as the Robot Operating System. The UAV swarm includes one master UAV and several slave UAVs. The master UAV is aware of all the task requirements (e.g., cruising spirally and hovering at target points) as well as the flight attitude (e.g. speed, orientation and altitude) during the mission. And the slave UAVs try to imitate the master UAV's actions and maintain a consistent and stable flight formation. To avoid collisions, the UAVs are distributed under a secure distance of at least 1.5m between any two UAVs. The slave UAVs are uniformly distributed around the master UAV (e.g., up/down, left/right and back) in the cruising phase, during which the master UAV keeps an average medium speed of 9m/s [6].

Algorithm 1: The BCI learning algorithm

```
Input: Initial parameters of the agent $i$'s policy (i.e., $\theta_i^0$)
       and the discriminator (i.e., $\omega_i^0$); the weight factors of
       $\zeta_1, \dots, \zeta_3$; the Adam hyper-parameters of $\alpha, \beta_1, \beta_2$;
       the learning rate $\eta$.
Output: Learned policies $\pi_{\theta_i}$
1 for iteration $h = 1, \dots, H$ do
2     for $i = 1$ to $N$ do
3         Sample a batch of expert trajectories $\{\chi_E\}_{t=1}^T \sim \pi_E$;
4         Sample a batch of state-action pairs $\{\chi_i\}_{t=1}^T \sim \pi_{\theta_i}$;
5         Update $\omega_i$ by ascending with the gradient of
          $\triangle_{\omega_i} = \mathbb{E}_{\rho_{\chi_i}}[\log D_{\omega_i}(B_{\psi_i}(h), a_i)] +$
              $\mathbb{E}_{\rho_{\chi_E}}[\log(1 - D_{\omega_i}(B_{\psi_i}(h), a_i))]$
          /* Training the discriminator in GAIL */;
6         Calculate the divergence gradient of
          $\nabla_{\psi_i} D_{JS}(\rho_{\chi_i} \| \rho_{\chi_E}) =$
              $\mathbb{E}_{\rho_{\chi_i}}[\nabla_{\psi_i} \log \chi_i(a_i|B_{\psi_i}(h)) Q_{\chi_i}(B_{\psi_i}(h), a_i)] +$
              $\mathbb{E}_{\rho_{\chi_i}}[\nabla_{\psi_i} \log D^\star(B_{\psi_i}(h), a_i)] +$
              $\mathbb{E}_{\rho_{\chi_E}}[\nabla_{\psi_i} \log(1 - D^\star(B_{\psi_i}(h), a_i))]$
          /* Training the belief component */;
7         Update $\phi_i$ in the action-oriented regularizer $L_I$ by
          $\phi_i \leftarrow \phi_i + \eta \cdot \mathrm{Adam}(\phi_i, \alpha, \beta_1, \beta_2)$;
8         Update the belief regularizer with
          $\nabla_{\psi_i} \mathcal{L}(\psi_i) =$
              $\nabla_{\psi_i}(\zeta_1 \sum_{t'=t}^{t+k} \widetilde{\mathbb{E}}_{\rho_{\chi_i}}[\|B_{\psi_i}(h^{t'}) - o^{t'}\|]^2 +$
                  $\zeta_2 \sum_{t'=t-k}^{t-1} \widetilde{\mathbb{E}}_{\rho_{\chi_i}}[\|B_{\psi_i}(h^{t'}) - o^{t'}\|]^2 +$
                  $\zeta_3 \mathbb{E}_{\rho_{\chi_E}}[\log Q(a_{\pi_E}|a_i \sim \rho_{\pi_{\theta_i}})] + H(a_{\pi_E}))$
          /* Regularize the belief representation */;
9         Take a policy step from $\theta_i^h$ to $\theta_i^{h+1}$ by using the
          TRPO method;
```

For the comparison purpose, we select the MAGAIL [27] and the ILPO [2] algorithms as the baselines and apply them to the drone swarm formation control case. We utilize four evaluation metrics, which are defined as follows.

- **Normalized reward**. The master UAV's state-action trajectories are adopted as expert demonstrations. As for the slave UAVs' rewards, they are not predefined and are quite sparse. During cruising or hovering, a reward of +1 is given if one slave UAV executes a correct action within a certain delay. Thus, extensive reward engineering is not involved. For clearness, the total reward of the master UAV during a spiral trajectory is normalized to 1 (i.e., perfect reward), and the relative percentage of the corresponding algorithm's reward reflects its imitation accuracy.
- **Average relative position error**. This metric is used to measure the average difference between the



(a) UAV prototype

(b) Vertical view of the spiral flight trajectory

(c) 3D view of the spiral flight trajectory

Figure 2: Illustration of the imitation-driven trajectory adjustment

practical position and the theoretical position across the slave UAVs, which implicitly reflects the imitation efficacy (i.e., a larger relative position error indicates a poorer imitation efficacy). Let $(x_i^\tau, y_i^\tau, z_i^\tau)$ and $(x_i^{\tau\prime}, y_i^{\tau\prime}, z_i^{\tau\prime})$ denote the theoretical and practical 3D-coordinates of the i-th slave UAV at time instant τ, respectively. The average relative position error is calculated as

$$\bar{\varepsilon} = \frac{1}{\mathrm{T}}\sum_{\tau=1}^{\mathrm{T}}\frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i^{\tau\prime}-x_i^{\tau})^2+(y_i^{\tau\prime}-y_i^{\tau})^2+(z_i^{\tau\prime}-z_i^{\tau})^2},$$

where T denotes the cruising period.

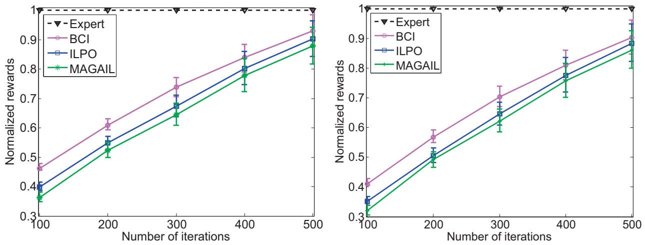
- **Average completion time**. This metric calculates the total time spent during the imitation-learning-based flight. We set that UAVs will hover for 10s at each hovering point. One hovering point indicates two imitation actions (i.e., hover and re-cruise) towards certain directions. Note that we ignore the delays during takeoff and landing. Thus, the total completion time is the sum of the total hovering time at all hovering points and the total imitation-learning-guided cruise time along the spiral trajectory except takeoff/landing. The average completion time is obtained by averaging over 100 independent runs of the spiral flight.
- **Average energy cost**. This metric calculates the total energy cost spent during the imitation-based computation and communication procedures. According to Eq. (11), we utilize the workload (in the unit of CPU cycles per bit) to depict the sampled batch size in the imitation task. The average energy cost is also obtained by averaging over 100 independent runs of the spiral flight.

5.2 Evaluation Results

We conduct four sets of experiments to compare the abovementioned metrics.

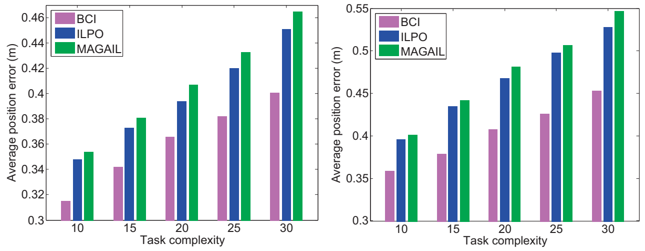
5.2.1 Comparisons on normalized reward. Figs. 3(a)∼3(b) display the comparison results on normalized reward under the slave-UAV numbers of 5 and 8, respectively. It can been seen that all algorithms yield low rewards in the initial learning phase and gradually approach the expert-level performance as the imitation learning proceeds. Among the three considered algorithms, the ILPO algorithm always yields higher rewards than MAGAIL. The reason is that the ILPO algorithm learns a potential dynamics model and can identify the latent factor that the expert is most likely to intend. However, ILPO is inferior to our BCI algorithm under all settings. The reason is that ILPO only predicts the next state under a given state and latent action (i.e., $f(s^{t+1}|s^t, z)$, where z denotes the latent action). While the BCI algorithm incorporates the action-assisted regularizer with the historical/predictive observation trajectories, and thus can yield robust belief representation and high-fidelity imitation even under incomplete observations. From Fig. 3 we can also know that it is more challenging to keep a consistent imitation since all algorithms generally obtain reduced normalized rewards as the number of slave UAV increases.

5.2.2 Comparisons on average relative position error. Figs. 4(a) ∼4(b) display the comparison results on average relative position error. We observe the relative position variations under different task complexities, which are reflected by the number of hovering points (i.e., a larger number of hovering points indicates a more complex imitation task since each hovering point triggers the hovering and re-cruise imitations). It can be seen that it is more likely to generate larger position deviations in a larger size of flight formation. The average relative position error of the ILPO algorithm is slightly lower than MAGAIL, but is evidently higher than



(a) 5 slave UAVs (b) 8 slave UAVs

Figure 3: Comparisons on normalized reward under different numbers of slave UAVs



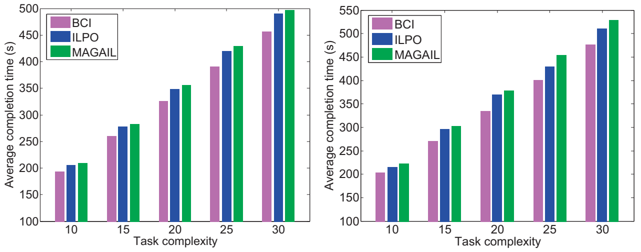
(a) 5 slave UAVs (b) 8 slave UAVs

Figure 4: Comparisons on average relative position error under different numbers of slave UAVs

the BCI algorithm particularly when the slave-UAV number increases to 8. Such results indicate that the BCI algorithm performs well in imitating the flight trajectory.

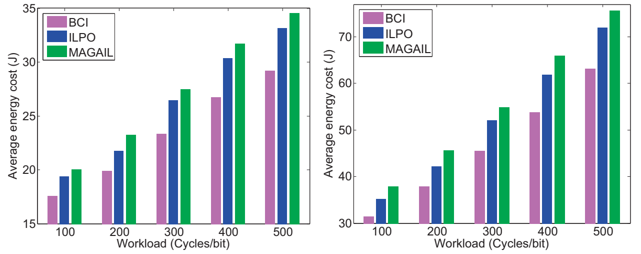
5.2.3 Comparisons on average completion time. Figs. 5(a)∼5(b) display the comparison results on average completion time. We also observe the impact of task complexity (i.e., the number of hovering points) on average completion time. A more complex imitation task (i.e., a larger number of hovering points) will prolong the completion time. From Fig. 5 we know that it takes longer time to fulfill the imitation task in a larger size of flight formation. The completion time of the baselines are relatively close, but are longer than the BCI algorithm, which verifies that the proposed algorithm is more time-efficient.

5.2.4 Comparisons on average energy cost. Figs. 6(a)∼6(b) display the comparison results on average energy cost. We observe the energy variations by varying the workloads. As the imitation task completion time plays a significant role in energy cost, thus, the variation trends of the three considered algorithms in Fig. 6 are similar to those in Fig. 5. From Fig. 6 we know that the computation and communication energy costs increase obviously as the size of the swarm becomes larger. Moreover, the energy cost gaps among the three algorithms become larger as the workloads increase. Above all, the BCI algorithm is the most energy-efficient.



(a) 5 slave UAVs (b) 8 slave UAVs

Figure 5: Comparisons on average completion time under different numbers of slave UAVs



(a) 5 slave UAVs (b) 8 slave UAVs

Figure 6: Comparisons on average computation & communication energy cost under different numbers of slave UAVs

6 CONCLUSION

In this paper, we investigate the distributed formation coordination issue in drone swarms under the generative adversarial imitation learning paradigm. In the challenging incomplete observation scenario, we try to recover the true environment state by virtue of the belief representations of historical observation trajectories, which are trained in conjunction with the imitation policies. Furthermore, we incorporate the action-assisted regularizer with the historical & predictive observation trajectories to improve the belief robustness and imitation fidelity. We evaluate the algorithm performance on the UAVs' formation control scenario. Evaluation results show that the slave UAVs can adaptively adjust the flight trajectory under the master UAV's demonstrations, and the collaborative execution efficiency is significantly improved.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grants 62002346 and 61902299, China Postdoctoral Science Foundation under Grants 2019TQ0239 and 2019M663636, and Zhejiang Lab's International Talent Fund for Young Professionals. Xiaofang Xia is the contact author.

REFERENCES

[1] Bing Du, Rui Xue, Liang Zhao, and Victor C.M. Leung. 2020. Coalitional graph game for air-to-air and air-to-ground cognitive spectrum sharing. *IEEE Transactions on Aerospace and Electronic Systems* 56, 4 (2018), 2959–2977.

[2] Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. 2019. Imitating latent policies from observation. In *International Conference on Machine Learning*, 1–9.

[3] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2018. Learning temporal strategic relationships using generative adversarial imitation learning. In *International Conference Autonomous Agent Multiagent Systems*, 113–121.

[4] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference Machine Learning*, 49–58.

[5] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances Neural Information Processing Systems*, 1–9.

[6] Ken Goss, Riccardo Musmeci, and Simone Silvestri. 2017. Realistic models for characterizing the performance of unmanned aerial vehicles. In *International Conference on Computer Communication and Networks*, 1–9.

[7] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A. Pires, Remi Munos. 2018. Neural predictive belief representations. *arXiv preprint arXiv: 1811.06407*.

[8] Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J. Lim. 2017. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *Advances in Neural Information Processing Systems*, 1–11.

[9] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 1–9.

[10] Liang Hong, Hongzhi Guo, Jiajia Liu, and Yanning Zhang. 2020. Toward swarm coordination: Topology-aware inter-UAV routing optimization. *IEEE Transactions on Vehicular Technology* 69, 9 (2020), 10177–10187.

[11] Woojun Kim, Myungsik Cho, and Youngchul Sung. 2019. Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 1–8.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 1–11.

[13] Alex Kuefler and Mykel J. Kochenderfer. 2018. Burn-in demonstrations for multi-modal imitation learning. In *International Conference on Autonomous Agent and Multiagent Systems*, 1071–1078.

[14] Yang Liu, Yifeng Zeng, Yingke Chen, Jing Tang, and Yinghui Pan. 2019. Self-improving generative adversarial reinforcement learning. In *International Conference on Autonomous Agent and Multiagent Systems*, 52–60.

[15] Ryan Lowe, Wu Yi, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances Neural Information Processing Systems*, 6379–6390.

[16] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. 2019. Modelling the dynamic joint policy of teammates with attention multi-agent DDPG. In *International Conference on Autonomous Agent and Multiagent Systems*, 1108–1116.

[17] Parinaz Naghizadeh, Maria Gorlatova, Andrew S. Lan, and Mung Chiang. 2019. Hurts to be too early: Benefits and drawbacks of communication in multi-agent learning. In *IEEE International Conference on Computer Communications*, 622–630.

[18] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. *f*-GAN: Training generative neural samplers using variational divergence minimization. In *Advances Neural Information Processing Systems*, 1–9.

[19] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, 1–10.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv: 1807.03748*.

[21] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. xGAIL: Explainable Generative Adversarial Imitation Learning for Explainable Human Decision Analysis. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1334–1343.

[22] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence Statics*, 627–635.

[23] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 1–9.

[24] John Schulman, Sergey Levine, Philipp Moritz, Michael I Jordan, and Pieter Abbeel. 2015. Trust region policy optimization. In *International Conference on Machine Learning*, 1889–1897.

[25] Mohit Sharma, Arjun Sharma, Nick Rhinehart, and Kris M. Kitani. 2019. Directed-Info GAIL: Learning hierarchical policies from unsegmented demonstrations using directed information. In *International Conference on Learning Representations*, 1–11.

[26] MyungJae Shin and Joongheon Kim. 2019. Randomized adversarial imitation learning for autonomous driving. In *International Joint Conference on Artificial Intelligence*, 4590–4596.

[27] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-agent generative adversarial imitation learning. In *Advances Neural Information Processing Systems*, 1–12.

[28] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Recent advances in imitation learning from observation. In *International Joint Conference on Artificial Intelligence*, 6325–6331.

[29] Chao Wang, Jian Wang, Yuan Shen, and Xudong Zhang. 2019. Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology* 68, 3 (2019), 2124–2136.

[30] Bo Yang and Min Liu. 2018. Keeping in touch with collaborative UAVs: A deep reinforcement learning approach. In *International Joint Conference on Artificial Intelligence*, 562–568.

[31] Fuhui Zhou, Yongpeng Wu, Rose Qingyang Hu, and Yi Qian. 2018. Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. *IEEE Journal on Selected Areas in Communications* 36, 9 (2019), 1927–1941.