Learning Diverse Strategies for Human Robot Collaboration via GAIL

Yue Yang
Georgia Institution of Technology
yyang941@gatech.edu

Haoyue Chen
Georgia Institution of Technology
hchen765@gatech.edu

***Abstract*—Human-robot collaboration will largely improve human efficiency in the future. But it's a hard problem for the robot to sense the diverse strategies that human could adopt and take corresponding actions. A recent approach provides a possible solution to this problem. The Co-GAIL method, which can handle diverse human behaviors for training robot assistants, optimizes human and robot strategies collaboratively during interactive learning. In this paper, we improved the actual collaboration experience between human and robot by proposing two alphas. One is to create a more separable strategy space for the Co-GAIL method. The other one is to introduce a pre-trained role detector to make the flexible switch between human and robot be possible. We don't see a clear improvement in the alpha-1, but a follow-up experiments could indirectly explain the possible reasons. But alpha-2 achieves a nearly similar performance as a pre-defined upper bound.**

I. Introduction

Smart manufacturing and factories rely on automation and robotics, and Human-Robot Collaboration (HRC) helps improve the efficiency and productivity of industries [3]. Limited versatility in performing collaborative tasks of robots considerably restricts the potential development of HRC [4]. Adaptation to diverse human strategies and movements in collaborative manipulation tasks is a critical objective for current improvement of robot assistants. The Co-GAIL [5] method can be used to handle diverse human behaviors for training robot assistants, which optimizes human and robot strategies collaboratively during interactive learning. The human policy will learn to generate diverse and plausible collaborative behaviors from demonstrations, while robot policy learns to facilitate by estimating the unobserved potential strategies by human collaborators. The ability of sensing diverse strategies the human takes owe to the introduction of one strategy space, where variant strategies will be represented as codes in this latent space.

However, ambiguity could still be found in this latent space. Some codes from different strategies are not separated from each other. We argue this could negatively affect the final performance and introduce one classifier that could make the latent space more separable. This is defined as our alpha-1. Another drawback of Co-GAIL results from the necessity of fixing the role of human and robot in advance, which could badly influence human user's experience in real life. We solve this issue via introducing one pre-trained role detector to make robot flexibly select corresponding policy depending on the role that human might play. This is defined as our alpha-2. Therefore, our contributions are listed as follows:

1) We prove that the separability of strategy space could largely affect the performance of Co-GAIL, and propose a new method to make the strategy space as separable as possible.
2) By introducing a pre-trained role detector, we improve the actual human-robot collaboration experience.

II. Related Work

Human-robot collaboration (HRC) has been a popular research area due to its wide range of applications. Various learning approaches have been proposed, enabling robots to cope with diverse human behaviors. Many work try to make use of multi-agent reinforcement learning (MARL) to learn collaborative strategies [6], [7]. However, prior studies in learning diverse behaviors tend to train strategies to make their behaviors diverse [8], [9]. Co-GAIL instead tries to optimize for covering the behaviors shown in demonstrations.

III. Method

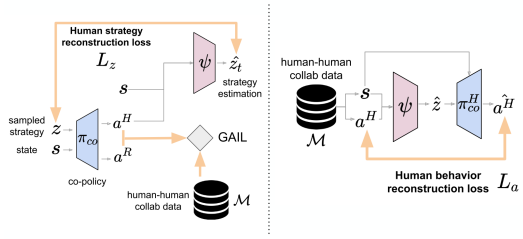
In this section, we give a brief introduction to an algorithm called Co-GAIL, which will be used as baseline method in alpha-1. Then we introduce the proposed alpha1 and alpha2, one aims to improve the performance of Co-GAIL, and the other one aims to improve the human experience for real life human-robot collaboration task.

A. Problem Formulation

Let us consider the human-robot collaboration task as an variant of Markov Decision Process (MDP) called two-agent Markov games that is defined as $M = (S, A^H, A^R, T, R, \gamma, \rho_0)$, with state space S, human action space A^H, robot action space A^R, transition distribution $T(s_{t+1}|s_t, a_t^H, a_t^R)$, reward function $R(s_t, a_t^H, a_t^R, s_{t+1})$, discount factor $\gamma \in [0, 1)$, and initial state distribution ρ_0.

B. Co-GAIL

Co-GAIL is an method trying to learn human-robot collaboration policy from human-human collaboration demonstrations. To reach the goal, they apply the multi-agent extension of the Generative Adversarial Imitation Learning (GAIL) framework called MA-GAIL, which train a co-policy π_{co} to imitate expert policy (π_{E_1}, π_{E_2}) so as to minimize the distance



(a) (b)

Fig. 1. **Co-GAIL Architecture** (a) co-policy π_{co} takes states s and strategy code **z** as input, output actions pair (a^H, a^R), which will be used to compare with expert demonstrations in GAIL setting. ψ model is used to generate strategy code z with states s and a^H as input. (b) The ψ and π_{co} models are trained as auto-encoder to learn L_a

between generated distribution $\rho(\pi_{co})$ and expert's distribution $\rho(\pi_{E_1}, \pi_{E_2})$:

$$\min_{\pi_{co}} \max_{D} \mathbb{E}_{\mathbf{x}\sim\rho(\pi_{E_1},\pi_{E_2})}[logD(\mathbf{x})] + \mathbb{E}_{\mathbf{y}\sim\rho(\pi_{co})}[log(1-D(\mathbf{y}))] \tag{1}$$

However, human could act differently given the same environment state s because human may take different strategies for the same task. The strategy could be inferred from history of observations and the human action that are denote as $\mathbf{h}_t = (\mathbf{s}_{t-K:t}, \mathbf{a}_t)$ at timestep t. To make learned co-policy take human strategy into consideration in order to improve the performance, a strategy recognition model $\psi(\mathbf{z}_t|\mathbf{h}_t)$ is proposed to learn a 2 dimensional strategy code z, where K denotes the length of history. The architecture of MA-GAIL equipped with strategy recognition model is shown in Figure 1 (a).

In addition, the human policy tends to ignore the latent strategy z and always opt to generate the most common behaviors. Therefore, a human strategy reconstruction loss(Equation 2) is introduced in Co-GAIL to avoid this (Figure 1 (a)). The second challenge is the imbalance distribution of training data, and to solve this, a human behavior reconstruction loss(Equation 3) is again introduced in Co-GAIL (Figure 1 (b)).

$$L_z = \mathbb{E}_{\mathbf{z}\sim p(z),\mathbf{h}\sim\rho(\pi^H_{co}(\cdot,\mathbf{z})}\|\psi(\mathbf{h}) - \mathbf{z}\| \tag{2}$$

$$L_a = \mathbb{E}_{(\mathbf{h}_t,\mathbf{s}_t,\mathbf{s}^H_t)\sim p_{\mathcal{M}}}\|\pi^H_{co}(\mathbf{s}_t, \psi(\mathbf{h}_t)) - \mathbf{a}^H_t\| \tag{3}$$

The final objective function for Co-GAIL could be derived by combining the above loss functions.

C. Alpha-1

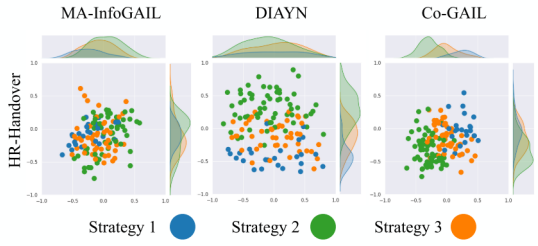


Fig. 2. Visualization of latent space for MA-InfoGAIL, DIAYN and Co-GAIL

The idea of Alpha-1 is inspired from the visualization of the learned latent strategy space (As shown in Figure 2) where different types of task-specific strategies are represented. We realized that Co-GAIL's latent space (i.e., strategy space) is more separable compared to other two methods', including MA-InfoGAIL [10] and DIAYN [11]. Hence, we assume that if different strategies can be more separable in the latent space, the better results would be generated. Because the overlapping latent codes of different strategies could only provide ambiguous information for the co-policy. The more separable the strategy space is, the easier for the co-policy to generate action pairs matching the corresponding human strategy.

The reason why Co-GAIL achieve a more separable strategy space owe to the introduction of L_a(As shown in Figure 1 (b)). L_a will enforce ψ model to generate different **z** for different a^H, which makes Co-GAIL's latent space more separable compared to other methods. However, human strategy could also be different when a^H is the same but history of observations are different. In this case, the psi model in Co-GAIL will only lead to the same strategy code, which is why there are still some overlaps in Co-GAIL's strategy space.

Therefore, to make the strategy space as separable as possible, we introduce a classifier ϕ in strategy space to discriminate latent codes of different strategies in a supervised learning way. When training the whole model equipped with this classifier, the ψ model will be enforced to generate more separable strategy code and then make the co-policy generate more diverse strategies. Labels in the supervised learning setting are collected manually together with demonstrations, denoted as $\boldsymbol{\ell}$. Then we could get a new loss function, which would be added to the objective function (i.e., Equation 5):

$$L_{classifier} = \mathbb{E}_{z\sim p(z),h\sim\rho(\pi^H_{co}(\cdot,\mathbf{z})),\ell\sim p_M}\phi(\psi(\mathbf{h}),\boldsymbol{\ell}) \tag{4}$$

Combined with the $L_{classifier}$, the objective function for alpha-1 could be derived:

$$\begin{aligned}\min_{\psi,\pi_{co}} \max_{D} \quad & \mathbb{E}_{\mathbf{x}\sim\rho(\pi_{E_1},\pi_{E_2})}[logD(\mathbf{x})] + \\ & \mathbb{E}_{\mathbf{y}\sim\rho(\pi_{co})}[log1 - D(\mathbf{y})] + \\ & \lambda_1 L_z + \lambda_2 L_a + \lambda_3 L_{classifier}\end{aligned} \tag{5}$$

where λ_1, λ_2 and λ_3 are the hyper-parameters for the intention, expert behavior reconstruction regularization term and the strategy classifier term.

D. Alpha-2

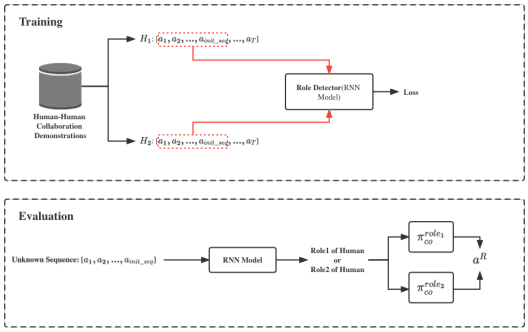


Fig. 3. Alpha-2 Architecture

In current two-agents implementation, the roles of leaders and followers must be assigned to the human agent and the robot agent in advance. For example, in a hand-over task, the human's role is fixed to give item to the robot and the robot can only take the item. We argue that this certainty could negatively affect the human user's experience if it's in a real life scenario because the robot couldn't flexibly select a different policy when human user plays a different role.

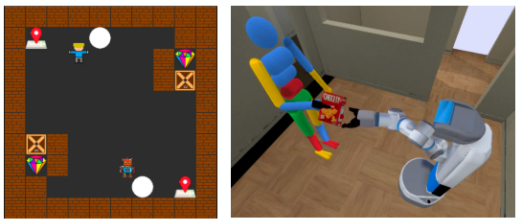
To improve human user's experience in one real life human-robot collaboration scenario, we propose a pre-trained role detector. We assume that the role of human could be detected from initial sequence of actions. As shown in Figure 3, an RNN model will be trained to classify sequences of actions so that it could detect the role human might play when evaluation. The robot could choose the corresponding policy $\pi_{co}^{role_1}$ or $\pi_{co}^{role_2}$ according to the human role provided by this role detector.

IV. Experimental Evaluation

In this section, experimental results are shown to validate the proposed alpha-1 and alpha-2. Experiments are designed to test the following hypotheses:

1) The performance of human-robot collaboration could be improved when strategy space is more separable.
2) The proposed alpha-1 could improve the performance of baseline method Co-GAIL.
3) Human's role could be detected initial from sequence of actions.
4) The proposed alpha-2 could improve human-robot collaboration experience.

A. Experiment Setup



(a) 2D-Fetch-Quest (b) HR-Handover

Fig. 4. (a) Human-robot collaboration pygame program: 2D-Fetch-Quest. (b) Human handover one item to the robot.

1) Datasets: We use the same dataset and data split method as baseline paper [5]. For alpha-1, we use demonstrations collected in a Pygame program (2D-Fetch-Quest, as shown in Figure 4 (a)). 120 trajectories are used in training and 90 trajectories are used for evaluation. Note that strategy labels are only available for strategy 3 and 4, labels for strategy 1 are mixed up together. So we label demonstrations of strategy 1 and 2 as label-1, strategy 3 as label-2, and strategy 4 as label-3. For alpha-2, we use demonstrations collected in a simulation environment iGibson [12], [13] and they are all about the one handover task (as shown in Figure 4 (b)). 154 trajectories are used in training and 67 trajectories are used for evaluation.

2) Classifier architecture: For alpha-1, a 3-layers neural network with 8 hidden neurons is used for this classification task. We choose cross-entropy as our loss function. For alpha-2, an RNN network with 64 hidden neurons stacked with one fully-connected layer is used for the sequence classification task.

3) Training and evaluation details: We use the successful rate over all the evaluation data as our metrics to measure the performance for both alphas. As to Alpha-1, according to the training curve(Figure 5), we set training epochs as 500 in total. The model is saved every 30 epochs. The weight of classifier loss function is 1 and we run the whole training process for 5 times by using 5 different random seeds, and follow the same procedure in the evaluation process. We set λ_1, λ_2 and λ_3 as 1.0. For Alpha-2, the RNN model is trained for 100 epochs. The training of both $\pi_{co}^{role_1}$ and $\pi_{co}^{role_2}$ follow the same line as original Co-GAIL.

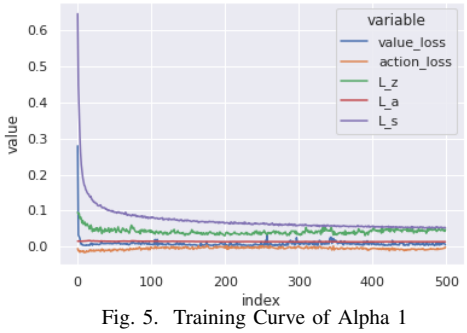


Fig. 5. Training Curve of Alpha 1

B. Results for Alpha-1

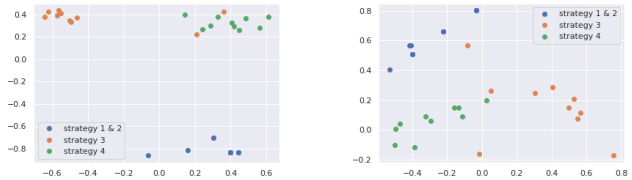


Fig. 6. Strategy space for Alpha1(left) and Baseline(right)

We obtain our results by evaluating the trained model on testing data for 5 times by using different random seeds. Every point every 30 epochs for each method is calculated by averaging the 5 values. 95% confidence interval is also calculated for both alpha-1 and baseline. As shown in Figure 7, line with orange region represents the performance of alpha-1 and line with blue region represents the performance of baseline method. We have the following observations:

1) We observe a better performance for alpha-1 compared to baseline method from 120∼300 epochs, which means the training speed for alpha-1 is faster.
2) We don't observe a clear improvement for alpha-1 compared to baseline method after both two method's model being well trained (i.e., after 360 epochs).

We argue that both observations could be explained if the first proposed hypothesis is right, that is the performance could

be improved when the strategy space is more separable. For the first observation, as mentioned in Section III-C, the original Co-GAIL method could bring partial separable feature to the strategy space owing to the introduction of L_a, but our alpha-1 boosts this process. This means that the strategy space could be earlier separable, which makes the performance improved faster if first hypothesis holds. As to the second observation, we argue that this results from the same level of separable strategy space. According to the visualization of strategy space (Figure 6) generated from well trained model (i.e., obtained at 480 epoch), we could find that although alpha-1's strategy space has much more separable clusters, boundaries between clusters in baseline method's strategy space are still easy to find. Therefore, when model is well trained, the strategy space for both methods could be easily separated, and this makes alpha-1 and baseline method achieve similar performance after 360 epochs. However, we think this is a special case for 2D-Fetch-Quest task because the state and action space is small, which makes the introduction of L_a enough to separate the strategy space. For those high dimensional tasks such as handover task, the strategy space is hard to perfectly separate as shown in Figure 2. For those cases, we believe alpha-1 could outperform the baseline method.

Owing to the lack of labels for handover task's demonstrations, we currently cannot implement direct experiments to verify the excellence of alpha-1 for handover task. Instead, we design a follow-up experiment to verify the first proposed hypothesis, which we believe could indirectly explain the 2 observations as stated above.

An opposite effect could be seen if λ_3 in Equation 5 is set to negative, that is the clusters in strategy space will be less separable. As shown in Figure 8, the performance will be better when the absolute value of λ_3 becoming smaller, where the first hypothesis holds, which indirectly explains the observations in the second hypothesis.

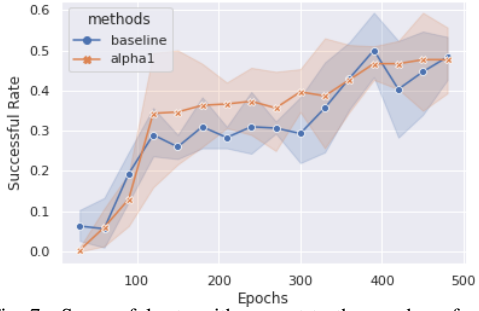


Fig. 7. Successful rate with respect to the number of epochs

C. Results for Alpha-2

For alpha-2, robot will switch its role between role1 and role2 flexibly depending on human's role. We assume that human's role could be detected from initial sequence of actions as stated in the third hypothesis. We test this hypothesis by training an RNN model to classify human's role given his/her initial sequence of actions. As shown in Figure 9, the RNN model converges quickly. When applied to evaluation data, the

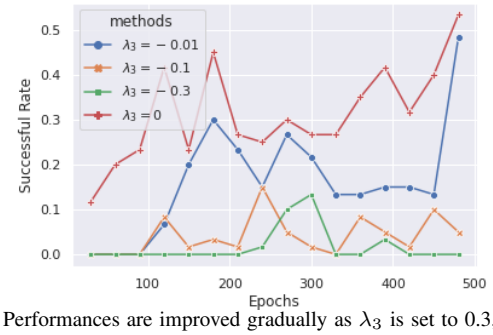


Fig. 8. Performances are improved gradually as λ_3 is set to 0.3, 0.1, 0.01, 0

RNN model reaches an average value of accuracy at 88.7% with standard deviation at 0.039, where the third hypothesis holds.

We then define an upper bound method where the human's role detection is 100% correct. When evaluation, the role of human is set randomly before the start of one trajectory. Then the robot equipped with the pre-trained role detector need to select corresponding policy flexibly in order to improve the successful rate. As shown in Figure 10, alpha-2 achieves similar successful rate as the upper bound, where the fourth hypothesis holds.

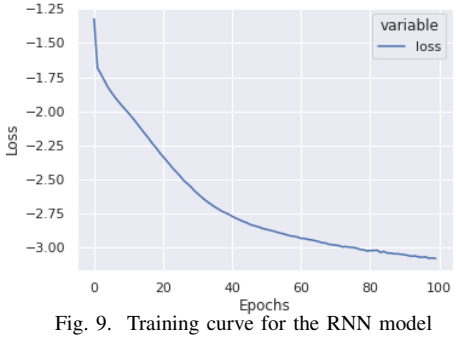


Fig. 9. Training curve for the RNN model

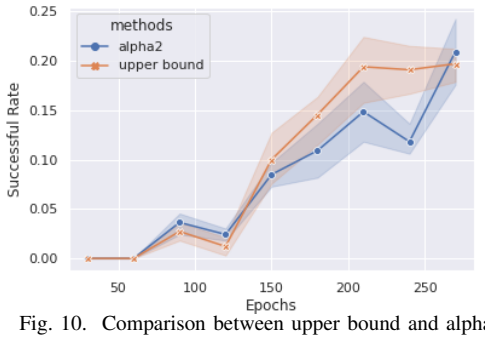


Fig. 10. Comparison between upper bound and alpha-2

V. Conclusion and Future Work

Based on the Co-GAIL method [5], we proposed two alphas that could help improve human-robot collaboration experience. Alpha-1 is inspired by the observation that better performance could be achieved when strategy space is more separable. Although we don't see a clear improvement for the 2D-Fetch-Quest task, we have informally proven the potential of alpha-1 in complex tasks with one follow up experiment. Alpha-2

succeed to make robot flexibly select correct policy depending on human's role.

In the future, more detailed experiments for complex tasks need to be implemented to directly verify alpha-1. As for alpha-2, currently policies available for the robot to select are limited, more intelligent pre-training approach should be designed.

REFERENCES

[1] Qian Luo, Maks Sorokin, and Sehoon Ha. A few shot adaptation of visual navigation skills to new observations using meta-learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13231–13237. IEEE, 2021.

[2] Hao-Lun Hsu, Qiuhua Huang, and Sehoon Ha. Improving safety in deep reinforcement learning using unsupervised action planning. *arXiv preprint arXiv:2109.14325*, 2021.

[3] Janis PArents, Valters Abolins, Janis Judvaitis, Oskars Vismanis, Aly Oraby, and Kaspars Ozols. Human–robot collaboration trends and safety aspects: A systematic review. *Journal of Sensor and Actuator Networks 10.3390/jsan10030048*, 2021.

[4] Jeremy A Marvel, Shelly Bagchi, Megan Zimmerman, and Brian Antonishek. Towards effective interface designs for collaborative hri in manufacturing: metrics and measures. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(4):1–55, 2020.

[5] Chen Wang, Claudia Pérez-D'Arpino, Danfei Xu, Li Fei-Fei, Karen Liu, and Silvio Savarese. Co-gail: Learning diverse strategies for human-robot collaboration. In *Conference on Robot Learning*, pages 1279–1290. PMLR, 2022.

[6] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[7] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Multi-agent collaboration via reward attribution decomposition. *arXiv preprint arXiv:2010.08531*, 2020.

[8] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

[9] Tobias Jung, Daniel Polani, and Peter Stone. Empowerment for continuous agent—environment systems. *Adaptive Behavior*, 19(1):16–39, 2011.

[10] Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in Neural Information Processing Systems*, 30, 2017.

[11] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[12] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks, 2021.

[13] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne P. Tchapmi, Micael E. Tchapmi, Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page accepted. IEEE, 2021.