Towards Open Ad Hoc Teamwork Using Graph-based Policy Learning

Arrasy Rahman, Niklas Höpner, Filippos Christianos, Stefano Albrecht

Autonomous Agents Research Group
School of Informatics
University of Edinburgh

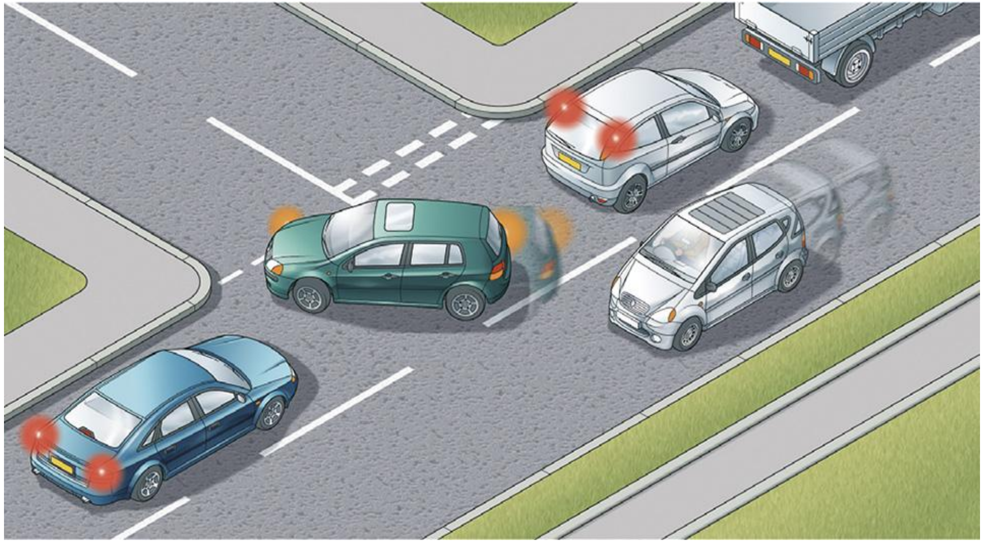
Introduction

Ad Hoc Teamwork

- Control a single agent (**learner**)
- Learner must **achieve a goal in the presence of other agents** without **prior coordination mechanisms**, such as:
    - Joint training
    - Communication with prespecified protocols

Open Multi-agent Systems

- In open multi-agent systems, agents may enter and leave the system anytime
- We aim to solve ad hoc teamwork in open multi-agent systems



Challenges for Open Ad Hoc Teamwork

1. Adaptation to different teammate policies
2. Adaptation to changing team sizes
3. Handling variable observation sizes

Problem Formulation

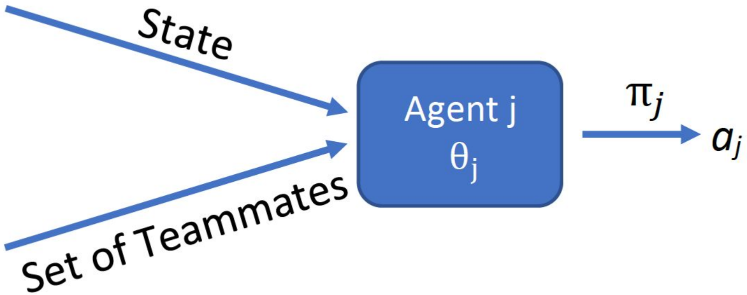
Open Stochastic Bayesian Games (OSBGs)

An OSBG is a 6-tuple, (*N, S, A, ϴ, R, P*), where:

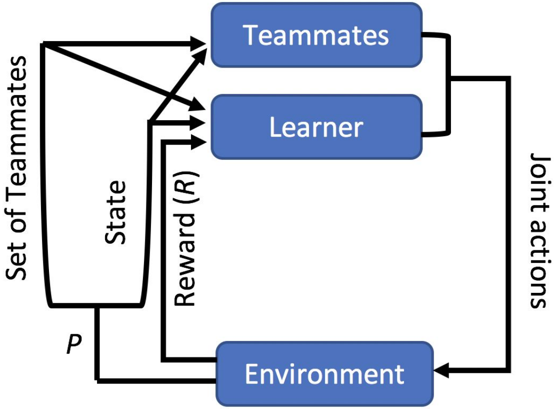
- *N* : Set of agents
- *S* : State space
- *A* : Action space
- *ϴ* : Type space
- *R* : Learner’s reward function
- *P* : Transition function

Open Stochastic Bayesian Games (cont.)

Teammate policies



Agent interaction in OSBGs



Learning Objective

- Learn learner’s optimal policy given an OSBG
- Given an OSBG, the optimal policy for a learner, $\boldsymbol{\pi}^{i,*}$, is a policy where:

$$\forall \pi^i, s, a^i, \bar{Q}_{\pi^{i,*}}(s, a^i) \geq \bar{Q}_{\pi^i}(s, a^i)$$

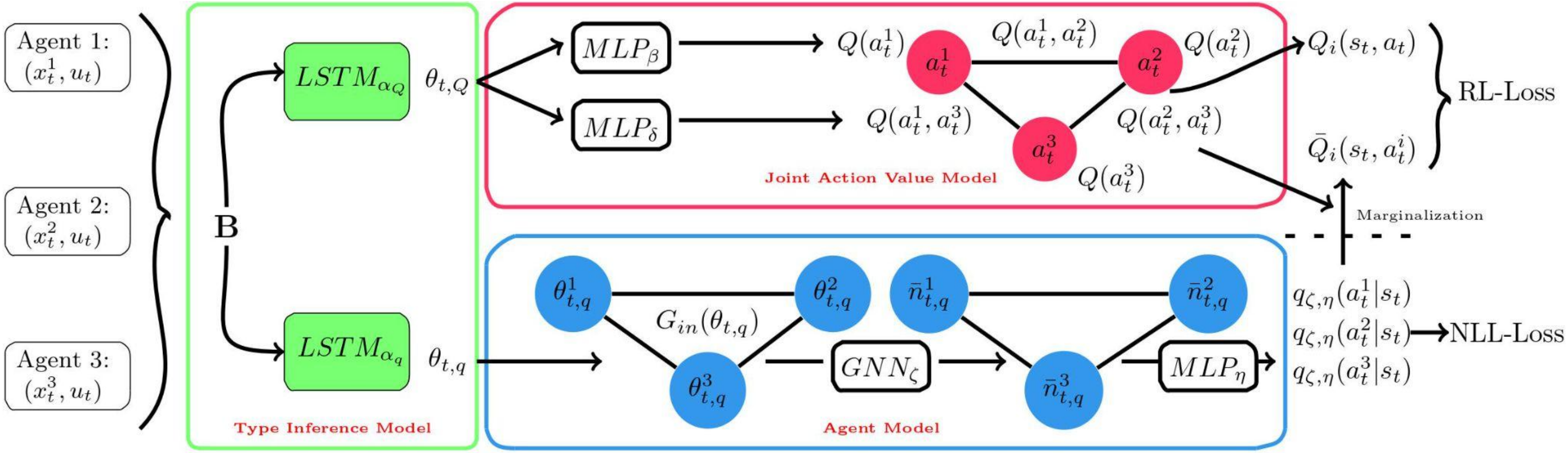
with,

$$\bar{Q}_{\pi^i}(s, a^i) = \mathbb{E}_{a_t^i \sim \pi^i, a_t^{-i} \sim \boldsymbol{\pi}_t^{-i}, P}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \middle| s_0 = s, a_0^i = a^i\right]$$

Graph-based Policy Learning

GPL Network Overview

- Estimate the optimal policy of an OSBG
- Utilize GNN-based models to handle openness



Joint Action Value Modelling - Motivation

- Teammates actions affect the learner’s returns
- Requires an approach for credit assignment
- We model the joint action value of a learner’s policy:

$$Q_{\pi^i}(s, a) = \mathbb{E}_{a_t^i \sim \pi^i, a_t^{-i} \sim \boldsymbol{\pi}^{-i}, P}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \middle| s_0 = s, a_0 = a\right]$$

- Implemented as a Coordination Graph (Guestrin et al., 2002)

Agent Modelling - Motivation

- Given the joint action value model:
  - How can we choose the learner's optimal action?
  - Teammate actions are uncertain!
- Action value function can be computed from joint-action value function

$$\bar{Q}(s_t, a^i) = \sum_{a^{-i} \in A^{-i}} Q(s_t, a) p(a^{-i} | s_t, a^i)$$

- *p* is unknown and must be modelled through agent modelling
- Implemented as Relational Forward Models (Tachetti et al., 2018)

Type Inference - Motivation

- Type inference is important because:
  a. Learner's returns depends on teammate actions.
  b. Teammate actions depends on their inherent types.
- θ^{-i} is unknown and must be inferred from teammates' observed behaviour.
- Implemented as LSTMs

Training GPL-based Models

Given, $< s, a, r, s' >_{n=1}^{|D|}$,

- Joint action value model trained with value-based RL

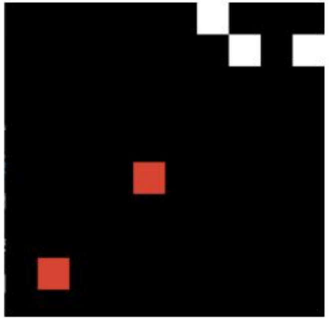
$$L_{\beta,\delta} = \frac{1}{2}\left(Q_{\beta,\delta}\left(s_t, a_t\right) - y\left(r_t, s_{t+1}\right)\right)^2$$

- Agent model trained with supervised learning

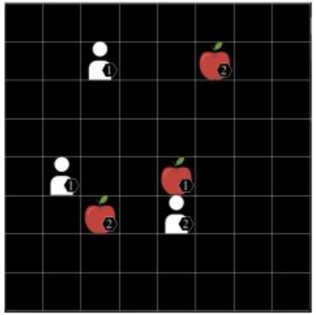
$$L_{\zeta,\eta} = -\log(q_{\zeta,\eta}(a_t^{-i} | s_t, a_t^i))$$

Experiments & Results

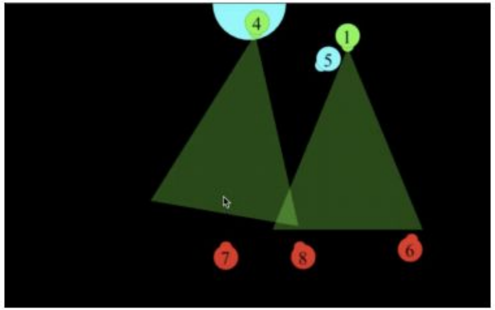
Experiments : Environments



Wolfpack (Leibo et al., 2017)



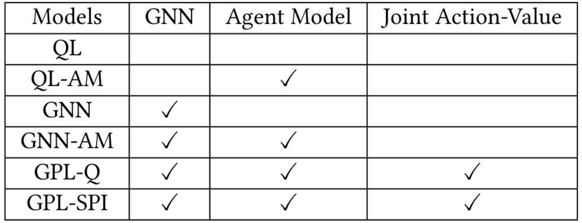
LBF (Albrecht et al., 2013.)



FortAttack (Deka et al., 2020.)

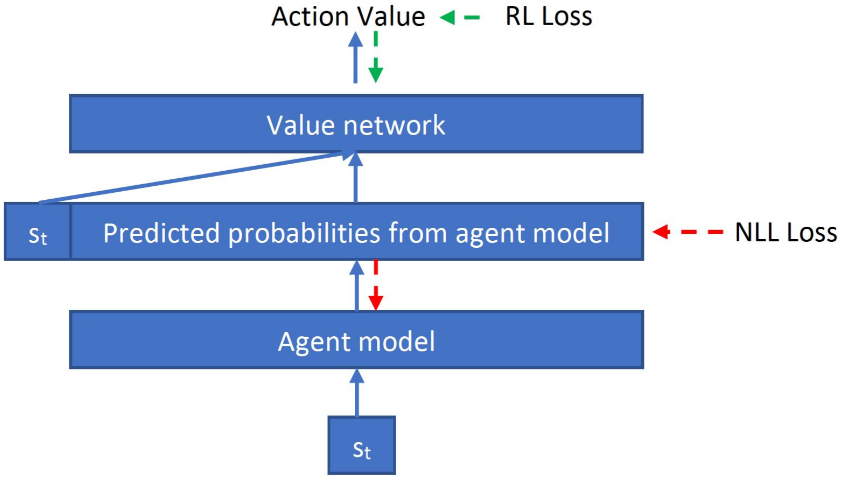
Experiments : Baselines

- Value-based approaches based on deep single agent RL



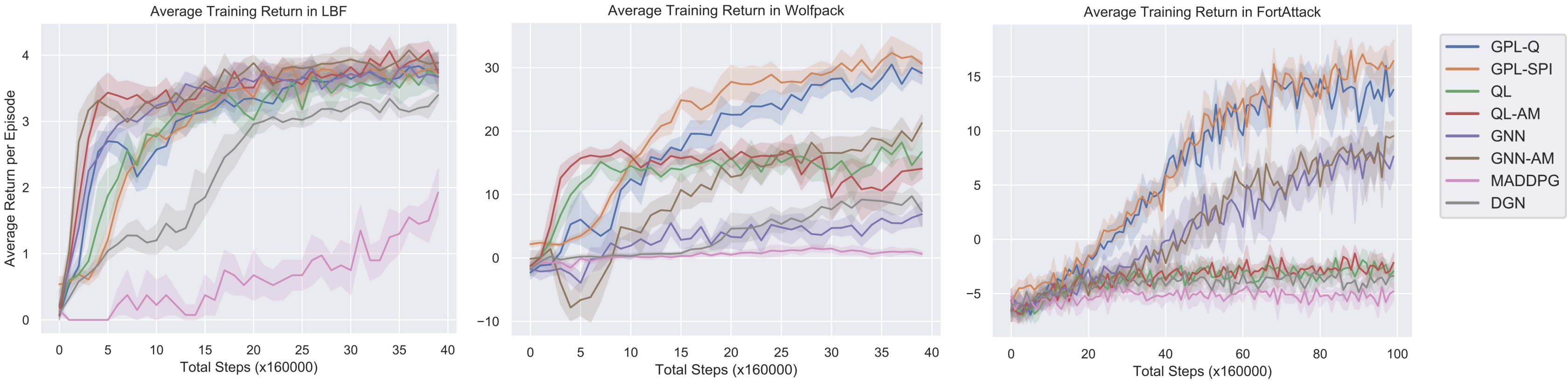
Models	GNN	Agent Model	Joint Action-Value
QL			
QL-AM		✓	
GNN	✓		
GNN-AM	✓	✓	
GPL-Q	✓	✓	✓
GPL-SPI	✓	✓	✓

- MARL approaches
  - MADDPG (Lowe et al., 2017)
  - DGN (Jiang et al., 2018)



Training Performance

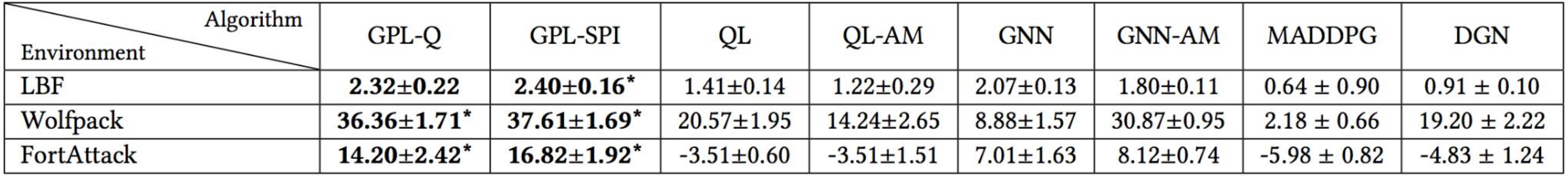
- Open process changes the number of agents in between timesteps
- Up to 2 teammates in each team



Training performance in LBF, Wolfpack, and FortAttack.

Evaluation Against Unseen Team Compositions

- Number of teammates increased up to 4 agents for generalization

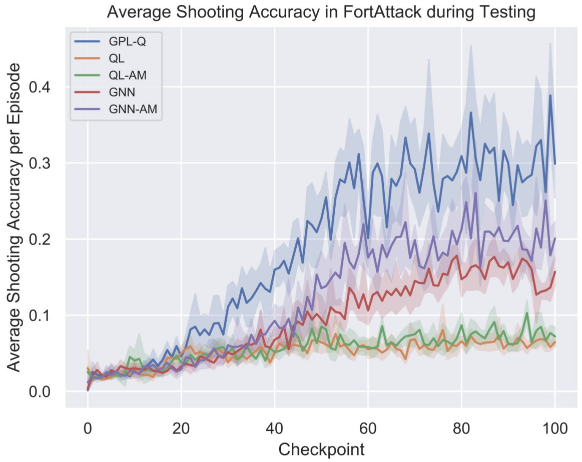


Algorithm Environment	GPL-Q	GPL-SPI	QL	QL-AM	GNN	GNN-AM	MADDPG	DGN
LBF	**2.32±0.22**	**2.40±0.16***	1.41±0.14	1.22±0.29	2.07±0.13	1.80±0.11	0.64 ± 0.90	0.91 ± 0.10
Wolfpack	**36.36±1.71***	**37.61±1.69***	20.57±1.95	14.24±2.65	8.88±1.57	30.87±0.95	2.18 ± 0.66	19.20 ± 2.22
FortAttack	**14.20±2.42***	**16.82±1.92***	-3.51±0.60	-3.51±1.51	7.01±1.63	8.12±0.74	-5.98 ± 0.82	-4.83 ± 1.24

Average and 95% confidence bounds of GPL and baselines during testing (up to 5 agents in a team for LBF, Wolfpack, and attacker & defender teams in FortAttack). For each algorithm, data was gathered by running the greedy policy resulting from the eight value networks stored at the checkpoint which achieved the highest average performance during training. The asterisk indicates significant difference in returns compared to the single-agent RL baselines.

Why GPL Works : An Analysis in FortAttack

- Which GPL component is responsible for GPL's performance?
- How does this translate to improved returns?



(a) Shooting accuracy in FortAttack

An Analysis in FortAttack (cont.)

- Evaluate several shooting-related metrics and see its correlation with returns
- Among all metrics,

$$\bar{Q}_{j,k} = \frac{\sum_{a^k} Q_{\delta}^{j,k}(a^j = \text{shoot}, a^k | s)}{|A^k|}$$

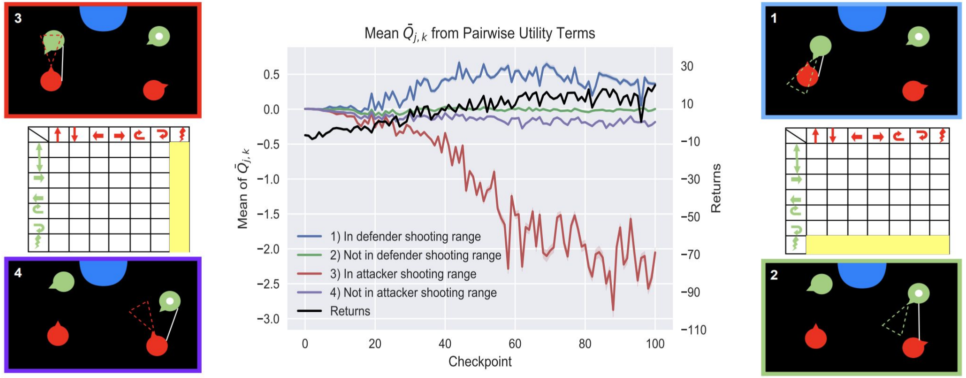
, by far has the highest correlation with performance (Pearson correlation coefficient of 0.85)

An Analysis in FortAttack (cont.)

- Strong correlation when *j* is a defender and *k* is an attacker
- MLP_δ learn that :

  "If *k* is an attacker inside j's (any defender) shooting range → High shooting values for *j* shooting *k*."

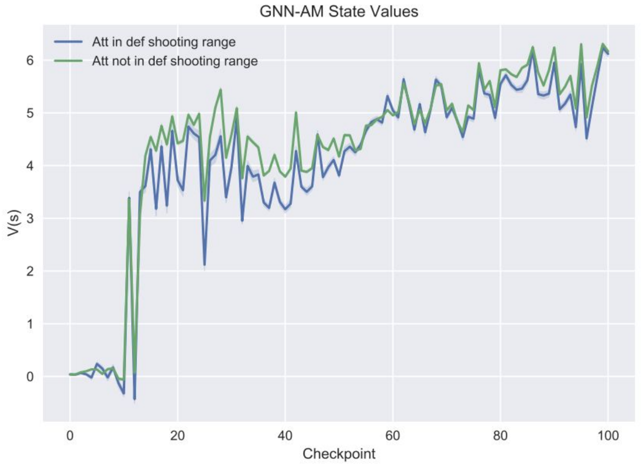
- MLP_δ enables reuse of knowledge



Evolution of shooting metrics derived from pairwise utility terms.

Value Learning in Baselines

- Learner must successfully shoot attackers itself to increase the value of shooting
- Shooting well-trained opposition is difficult
- Baselines do not learn the value of other teammates' actions



State value function estimates for GNN-AM.

Conclusion

Conclusion

- GPL’s action value computation is a crucial component for learning and generalizing value functions in open ad hoc teamwork
- GNNs improves generalization performance in open ad hoc teamwork

Future Work

- Partial observability
- Non-stationarity
- GNN structure learning
    - More complex joint action value decomposition
    - GNN structure learning for agent modelling
- Automatically generate teams for learning

Towards Open Ad Hoc Teamwork Using Graph-based Policy Learning

https://arxiv.org/abs/2006.10412

Code base :
https://github.com/uoe-agents/GPL