Efficient Multi-agent Cooperative Navigation in Unknown Environments with Interlaced Deep Reinforcement Learning

Yue Jin, Yaodong Zhang, Jian Yuan, and Xudong Zhang

Department of Electronic Engineering
Tsinghua University

I. BACKGROUND

Cooperative navigation: creating an efficient mobile robot group

- Cooperative rescue: cooperative fire fighting, cooperative search at disaster scenes
- Cooperative work: autonomous warehouse and logistics

Cooperative rescue

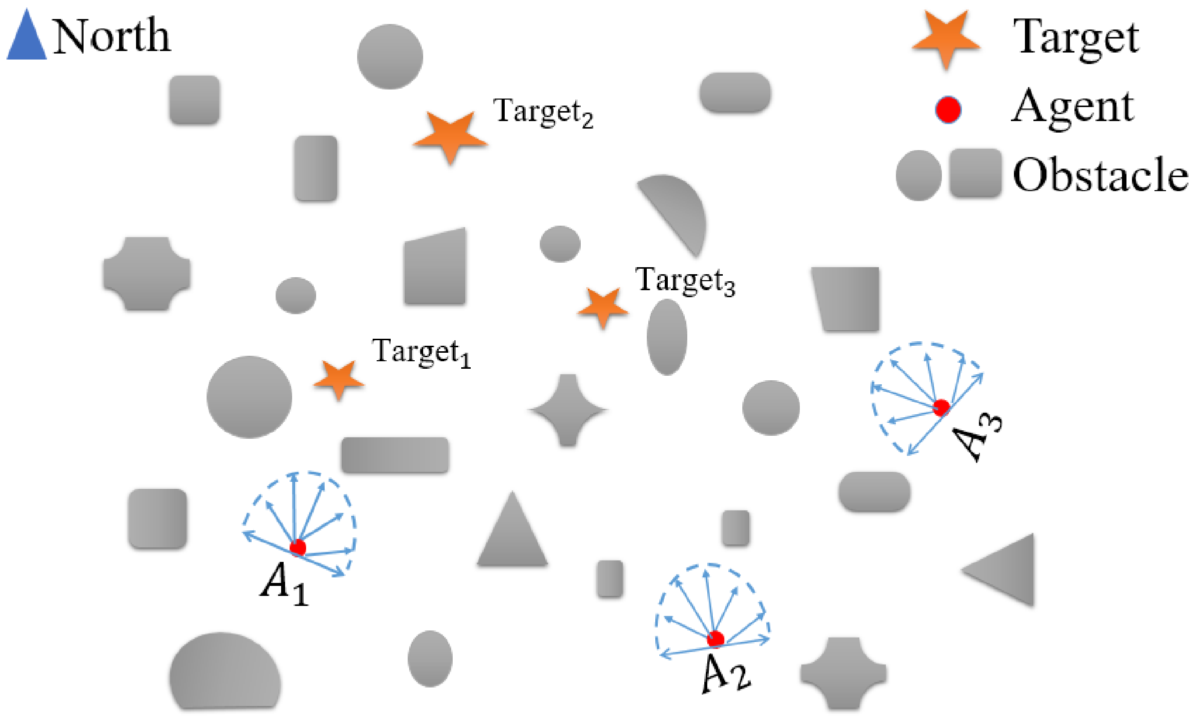
Cooperative exploration

Autonomous warehouse and logistics

Efficient cooperative navigation to scattered targets in unknown environments

Challenges of cooperative navigation to scattered targets in unknown environments

- Goal： to ensure that each target is reached by a robot and the overall time consumption can be minimized.



Unknown environment

Scattered targets

Efficient policy ?

I. BACKGROUND

Challenges of cooperative navigation to scattered targets in unknown environments

Unknown

The environment is unknown in advance.

Scattered

Targets are scattered at different locations.

Efficient

All targets should be arrived. Minimize the overall time consumption and avoid collisions.

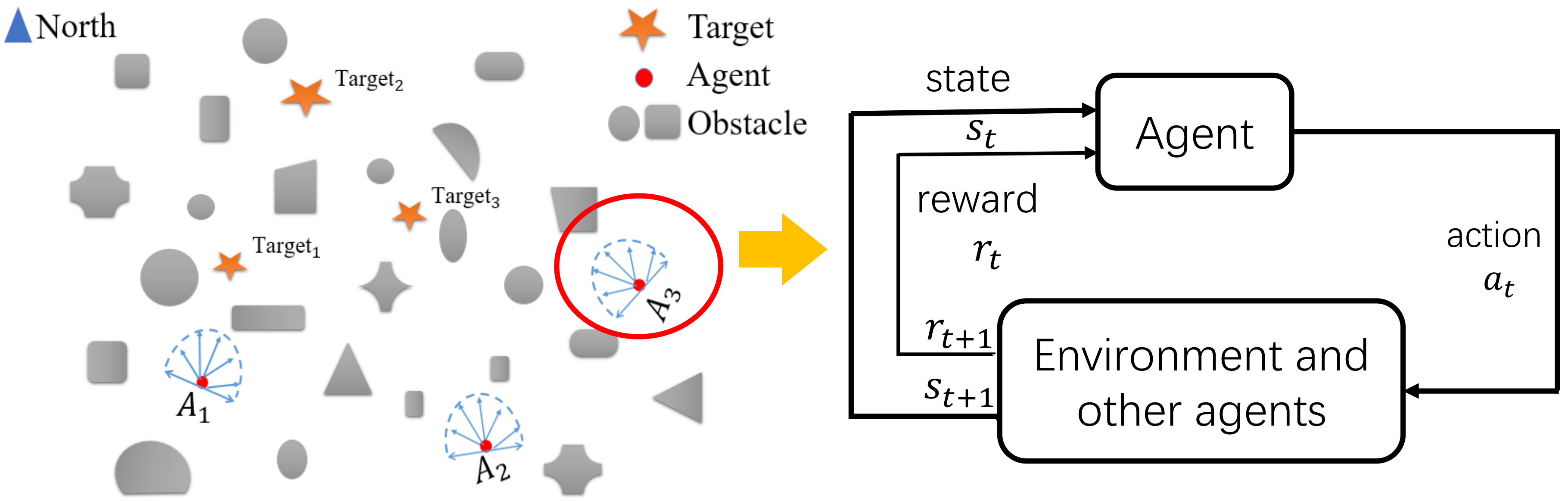
Modeling-and-planning methods are inefficient.
Hard-coded methods are intractable.
Centralized control lacks robustness and efficiency.

Cooperative navigation calls on more intelligent and efficient decentralized control algorithms.

2. METHOD

Formulate cooperative navigation as a reinforcement learning problem

Decentralized control of cooperative navigation:
a Markov Decision Process (MDP)



State: s_t
Action: a_t
Dynamic: $p(s_{t+1}|s_t,a_t)$
Reward: $p(r_t|s_t,a_t)$

Markov Decision Process

(Deep) Reinforcement learning solves MDPs through trial-and-error learning

2. METHOD

State profile and action profile

- Deep reinforcement learning directly takes high-dimensional sensory outputs as states.

State profile

- GPS to obtain the positions of targets and robots
- Other sensors to sense local environment, such as range finders to sense obstacles

Action profile

- Steering angle (assuming the speed is uniform)

Reward function derivation

- Reinforcement learning solves the optimal policy by maximizing the expectation of long-term discount rewards:

$$V_\pi(s) = \mathbb{E}\left[\sum_{\tau=0}^{T} \gamma^\tau r_{t+\tau} | s_t = s, \pi\right]$$

- The goal of cooperative navigation is to minimize the overall time cost, which can be transformed as to maximize the expectation of long-term discount rewards.

$$\min_{\pi_i} \mathbb{E}\left[T_{max} \middle| \mathbf{o}_i^0, \pi_i\right]$$

$$\begin{aligned} s.t. \quad & I(T_{max}) = 1 \\ & \|\mathbf{o}_{i,ag_j}^{T_{max}}\|_2 \geq d_r \quad \forall j \neq i \\ & d_{i,k}^{T_{max}} \geq d_r \quad \forall k \,. \end{aligned}$$

$$\max_{\pi_i} \mathbb{E}\left[\sum_{t=1}^{T_{max}} \left(\underbrace{-1 + C_1 \delta\left(I(t) - 1\right) - C_2 \sum_{j=1, j\neq i}^{N} u\left(d_r - \|\mathbf{o}_{i,ag_j}^t\|_2\right) - C_3 \sum_{k=1}^{K} u\left(d_r - d_{i,k}^t\right)}_{r_t}\right) \middle| \mathbf{o}_i^0, \pi_i\right]$$

Time cost penalty

Collision penalty

Overall success reward

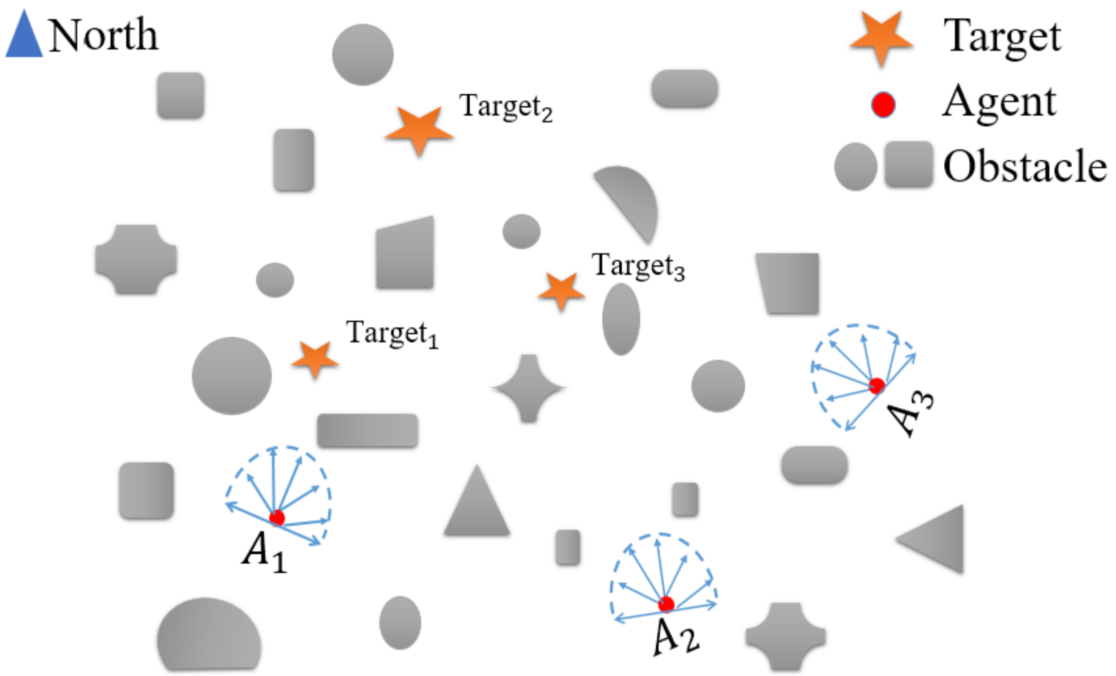
2. METHOD

Hierarchical policy model of cooperative navigation

- Unknown environment
- Limited sensing range
- Cooperate to minimize time consumption

↓

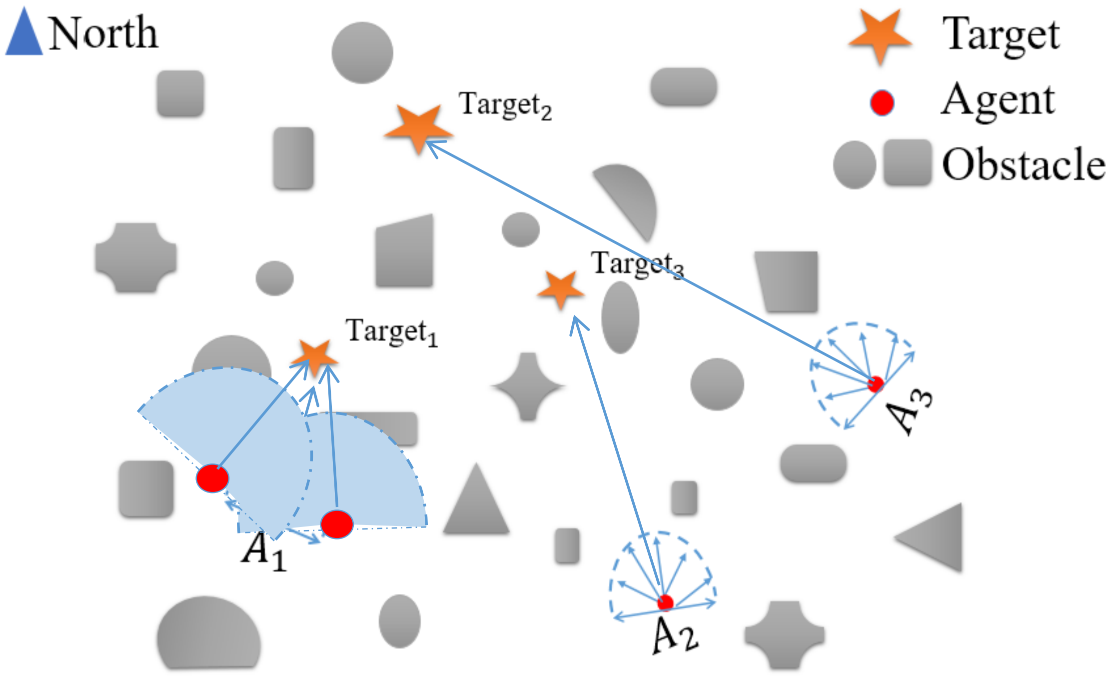
- Select different targets dynamically
- Avoid collisions

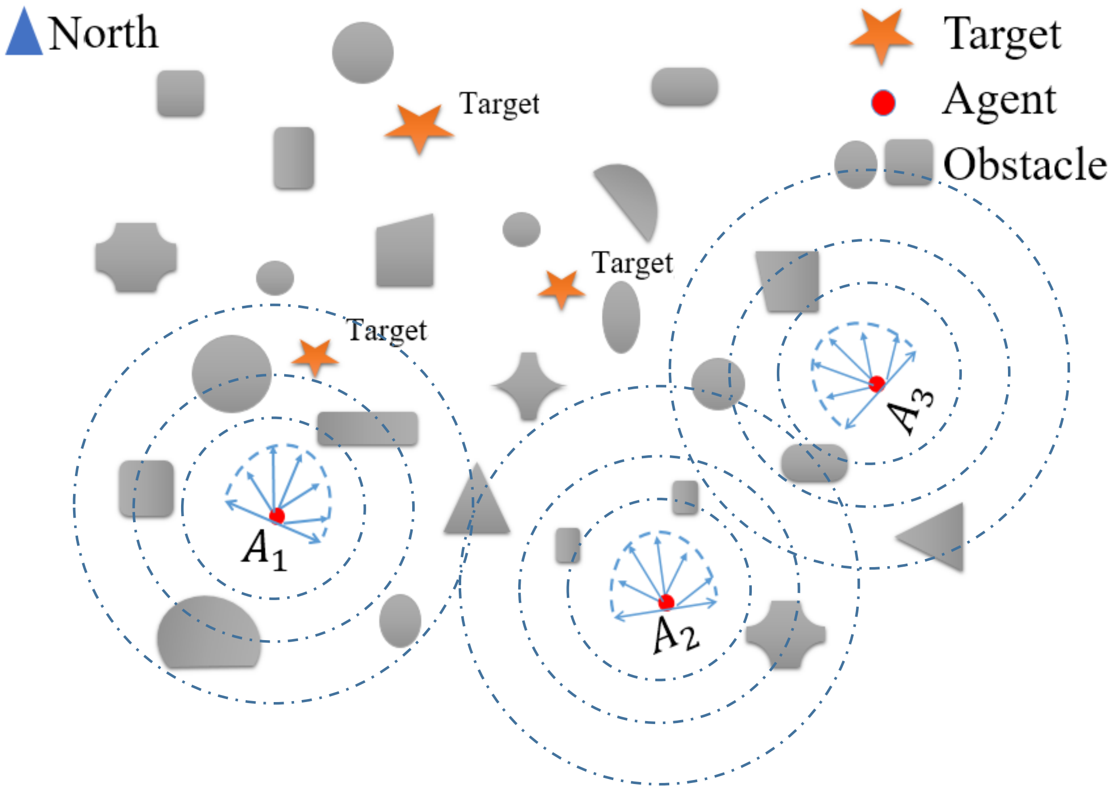


Cooperative navigation policy can be modeled as a combination of a dynamic target selection policy and a collision avoidance policy.

2. METHOD

Hierarchical policy vs. Single policy





Hierarchical policy learning can shrink the unnecessary trials and can accelerate the learning process

Single policy learning involves larger policy space and more meaningless trials

Accelerate learning with policy hierarchies

Cooperative navigation policy

Target selection policy

$$\boldsymbol{o}_p^t \rightarrow a_{ts}^t \in [1, N]$$

Discrete policy space

$$a_{ts}^t = \arg\max_a Q_{ts}(\boldsymbol{o}_p^t, a)$$

Collision avoidance policy

$$o\,(a_{ts}^t) \rightarrow a_{ca}^t \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}]$$

Only to observe the obstacles near the selected target direction

Continuous policy space

$$a_{ca}^t = \mu(o(a_{ts}^t), \theta^{\mu})$$

If no obstacles are observed in the selected target direction

Go straight towards the selected target

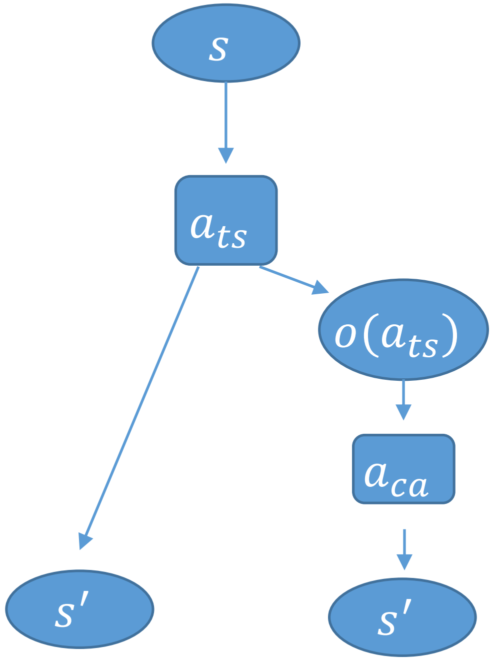
$$a_{ca}^t = \varphi(a_{ts}^t)$$

Both decrease the policy solution space

Two coupled policies

2. METHOD

Algorithm design: Interlaced deep reinforcement learning



Target selection policy

action-value function at time t:

No obstacles are observed: $Q_{ts}^*(s,a_{ts}) = \sum_{o',r} p(s',r \mid s,a_{ts})[\boxed{r+\gamma \max_{a'_{ts}} Q_{ts}^*(s',a'_{ts})}]$

Obstacles are observed: $Q_{ts}^*(s,a_{ts}) \approx Q_{ca}^*(o(a_{ts}),a_{ca})$

Collision avoidance policy

Unbiased estimation

action-value function at time t:

$$Q_{ca}^*(o(a_{ts}),a_{ca}) = \sum_{o',r} p(s',r \mid s,a_{ts},a_{ca})[\boxed{r+\gamma \max_{a'_{ts}} Q_{ts}^*(s',a'_{ts})}]$$

Unified learning structure

$$Q_{ts}(s,a_{ts}) = \underline{u(o_{d_1}-d_e)}Q_{ts}(s,a_{ts}) + (1-\underline{u(o_{d_1}-d_e)})\sum_{a_{ca}} p(o(a_{ts}),a_{ca} \mid s,a_{ts})Q_{ca}(o(a_{ts}),a_{ca})$$

No obstacles are observed

Loss function: $L(\theta^{Q_{ts}}) = E[(y-Q_{ts})^2]$

$L(\theta^{Q_{ca}}) = E[(y-Q_{ca})^2]$

No obstacles are observed

$$\boxed{y = \begin{cases} r+\gamma \max_{a'_{ts}} Q_{ts}(s',a'_{ts}), & \text{if } o'_{d_1} > d_e \\ r+\gamma Q_{ca}(o(a'_{ts\max}),\mu(o(a'_{ts\max}))), & \text{otherwise} \end{cases}}$$

Algorithm design: Interlaced deep reinforcement learning

- Homogeneous targets. Homogeneous agents share a common policy.
- Use Actor-Critic policy to learn target selection policy and collision avoidance policy
- Based on existing algorithms DQN[16] and DDPG[17] for discrete policy learning and continuous policy learning, respectively.
- Use three deep neural networks to approximate $Q_{ts}(s, a_{ts}; \theta^{ts})$, $Q_{ca}(o, a_{ca}; \theta^{ca})$ and $\mu(o; \theta^{\mu})$.
- Update the parameters by sampling a minibatch every iteration and using the SGD method.

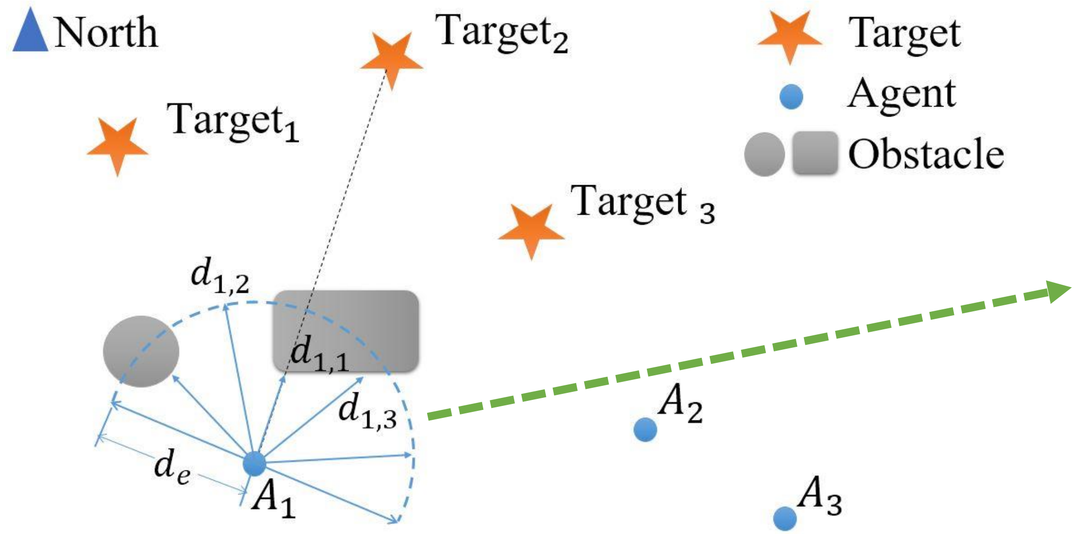
$$L(\theta^{ts}) = \frac{1}{M} \sum_j (y_i^j - Q^{ts})^2$$

$$L(\theta^{ca}) = \frac{1}{M} \sum_j (y_i^j - Q^{ca})^2$$

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{M} \sum_j \nabla_a Q^{ca}(o, a \mid \theta^{ca})|_{o=o_i^j, a=\mu(o_i^j)} \nabla_{\theta^{\mu}} \mu(o \mid \theta^{\mu})|_{o=o_i^j}$$

Simulation settings

- States are composed of two parts
- Randomly generate starting positions and target positions every episode

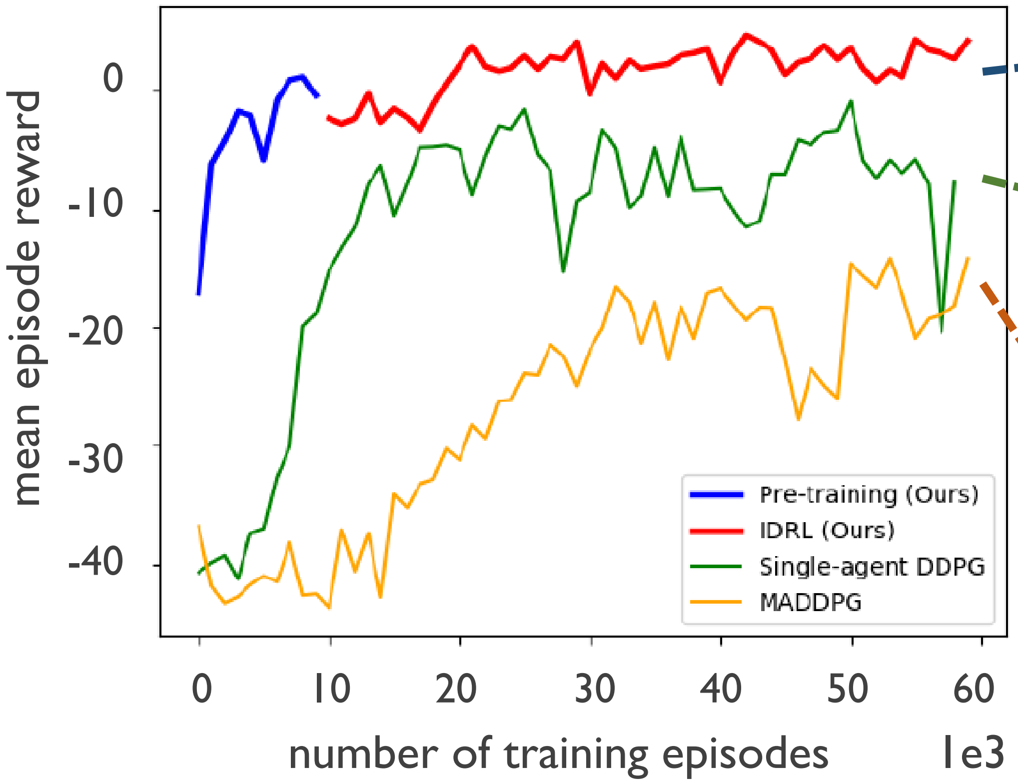


Relative position coordinates of targets and other agents

Ranging results of 7 detection beams between -90 and 90 degree.

3. EXPERIMENTS

Convergence curves



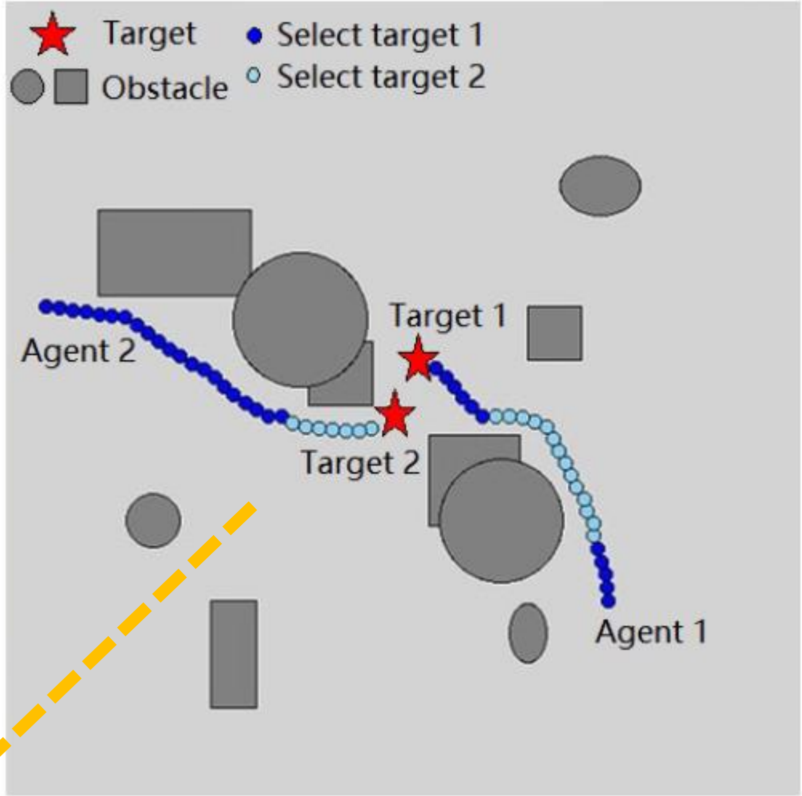
Converge fast and gains high rewards.

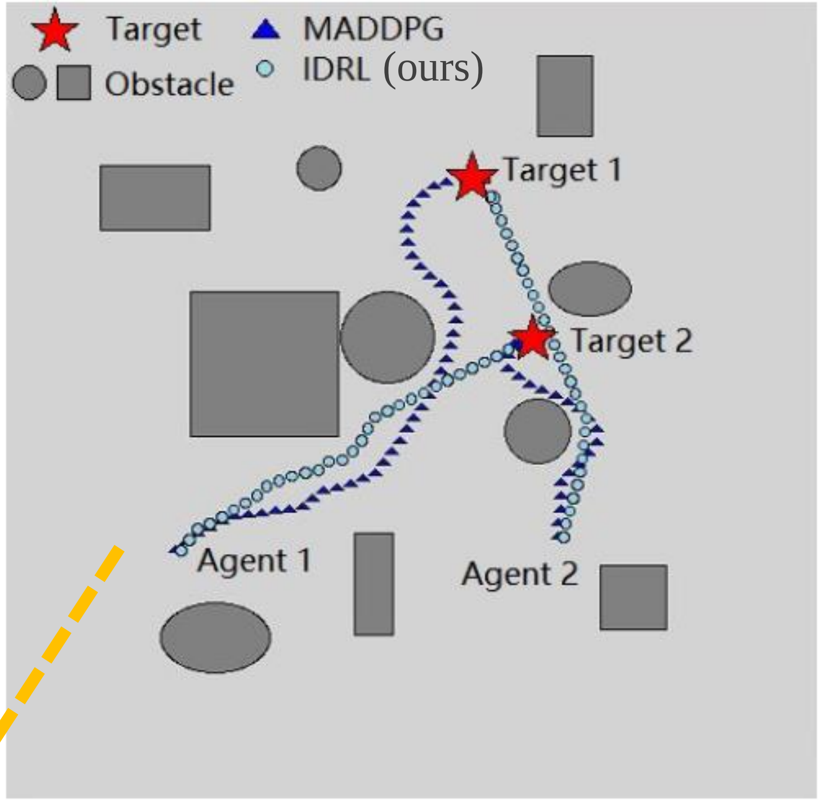
Randomly allocate targets in advance
Converge fast but gain less rewards than IDRL

Single policy learning algorithm

Centralized learning and decentralized execution
Only learn an steering policy
Converge slow and gain less rewards than IDRL

Navigation trajectories





Select different targets dynamically during the navigation process

Compared with single policy algorithm, IDRL gets more efficient cooperation navigation trajectories.

3. EXPERIMENTS

Statistical results: arrival rate and time cost

- Test 1000 episodes
- Randomly generate starting positions and target positions every episode

Obstacle size distribution (diameter or side length)	Learning method	Mean arrival rate	Mean maximum navigation time (s)
$U(1m, 2m)$	IDRL	0.98	41.34
	DDPG	0.82	48.62
	MADDPG	0.56	50.72
$U(3m, 4m)$	IDRL	0.95	41.86
	DDPG	0.76	49.05
	MADDPG	0.49	50.99

Obstacle size distribution (diameter or side length)

Compared with single policy algorithms:
IDRL achieves more than 16% improvement in mean arrival rate
IDRL reduces at least 15% mean maximum navigation time
IDRL is more robust

3. CONCLUSION & FUTURE WORK

Facing scattered targets in unknown environments, decentralized control problem of cooperative navigation is challengeable.

- Robots need to cooperate in order to select different targets dynamically and compute efficient navigation paths.
- Traditional methods lack efficiency in unknown environment with randomly scattered targets.

We propose an interlaced deep reinforcement learning method for cooperative navigation

- Model cooperative navigation as a Markov decision process.
- Model a hierarchical cooperative navigation policy to boost learning efficiency and propose an interlaced deep reinforcement learning algorithm to learn two coupled policies.

Future work

- Test the proposed algorithm with more targets and robots.
- Add information sharing between robots by communication.

References

1. R. Reid, A. Cann, C. Meiklejohn, L. Poli, A. Boeing, and T. Braunl, “Cooperative multi-robot navigation, exploration, mapping and object detection with ros,” in Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013, pp. 1083–1088.
2. Y.Wang, D.Wang, and S. Zhu, “A new navigation function based decentralized control of multi-vehicle systems in unknown environments,” Journal of Intelligent & Robotic Systems, vol. 87, no. 2, pp. 363–377, August 2017.
3. Y. F. Chen, M. Liu, M. Everett, and J. P. How, “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning,” in Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017, pp. 285–292.
4. P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning,” in Robotics and Automation (ICRA), 2018 IEEE International Conference on. IEEE, 2018, pp. 6252–6259.
5. Y. F. Chen, M. Everett, M. Liu, and J. P. How, “Socially aware motion planning with deep reinforcement learning,” in Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE, 2017, pp. 1343–1350.

References

6. F. S. Melo and M. Veloso, "Learning of coordination: Exploiting sparse interactions in multiagent systems," in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 773–780.
7. R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in Advances in Neural Information Processing Systems, 2017, pp. 6379–6390.
8. M. H¨uttenrauch, A. ˇSoˇsi´c, and G. Neumann, "Guided deep reinforcement learning for swarm systems," arXiv preprint arXiv:1709.06011, 2017.
9. X. Li, D. Sun, and J. Yang, "Preserving multirobot connectivity in rendezvous tasks in the presence of obstacles with bounded control input," IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 2306– 2314, January 2013.
10. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," IEEE robotics & automation magazine, vol. 13, no. 2, pp. 99–110, June 2006.
11. M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A datadriven approach to end-to-end motion planning for autonomous ground robots," in 2017 ieee international conference on robotics and automation (icra). IEEE, 2017, pp. 1527–1533.

References

12. L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE, 2017, pp. 31–36.
13. R. S. Sutton, A. G. Barto, F. Bach et al., Reinforcement learning: An introduction. MIT press, 2018.
14. L. J. Lin, "Reinforcement learning for robots using neural networks," Ph.d.thesis Carnegie Mellon University, 1993.
15. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, p. 529, February 2015.
16. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D.Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.
17. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
18. J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in International Conference on Autonomous Agents and Multiagent Systems. Springer, 2017, pp.66–83.

Q&A

- Thank you very much!