HELSA: Hierarchical Reinforcement Learning with Spatiotemporal Abstraction for Large-Scale Multi-Agent Path Finding

Zhaoyi Song[1], Rongqing Zhang[1], and Xiang Cheng[2]*

[1] Network and Machine Intelligence Lab, School of Software Engineering, Tongji University, P.R. China
[2] School of Electronics Engineering and Computing Science, Peking University, P.R. China

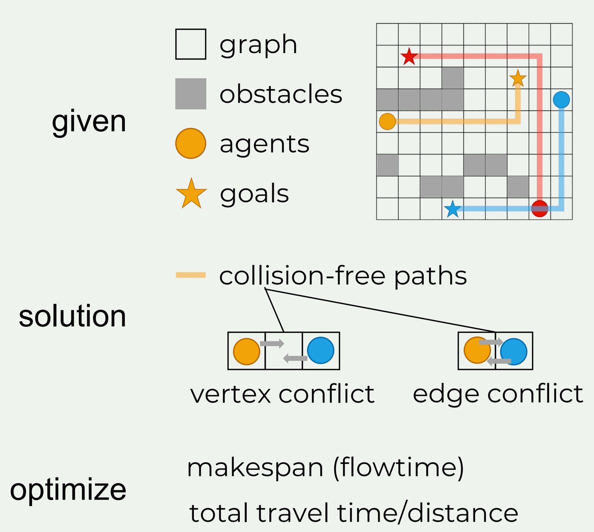
Network and Machine Intelligence Lab
TONGJI UNIVERSITY
PEKING UNIVERSITY

2023 IROS

Introduction

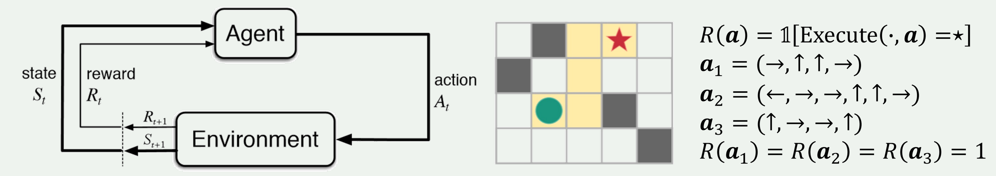
Background

Multi-Agent Path Finding (MAPF)

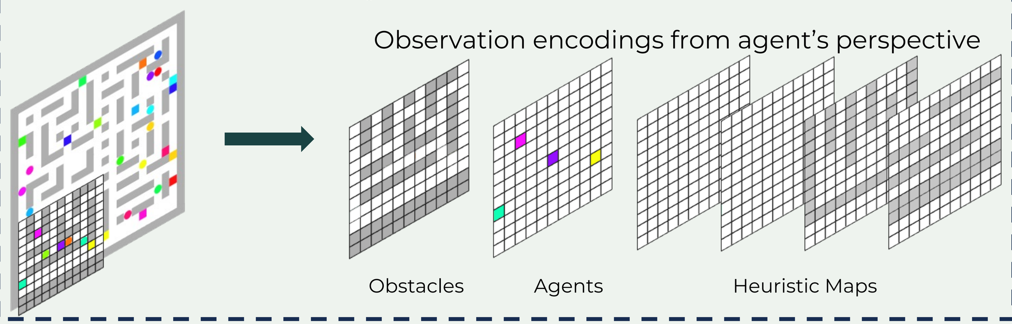


Key Challenges in Large-Scale MARL

1. Sparse Rewards in Large-Scale Reinforcement Learning

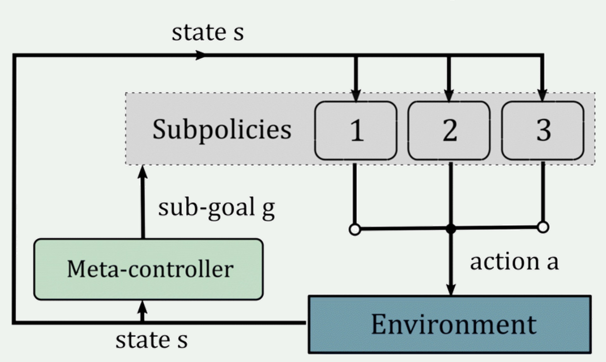


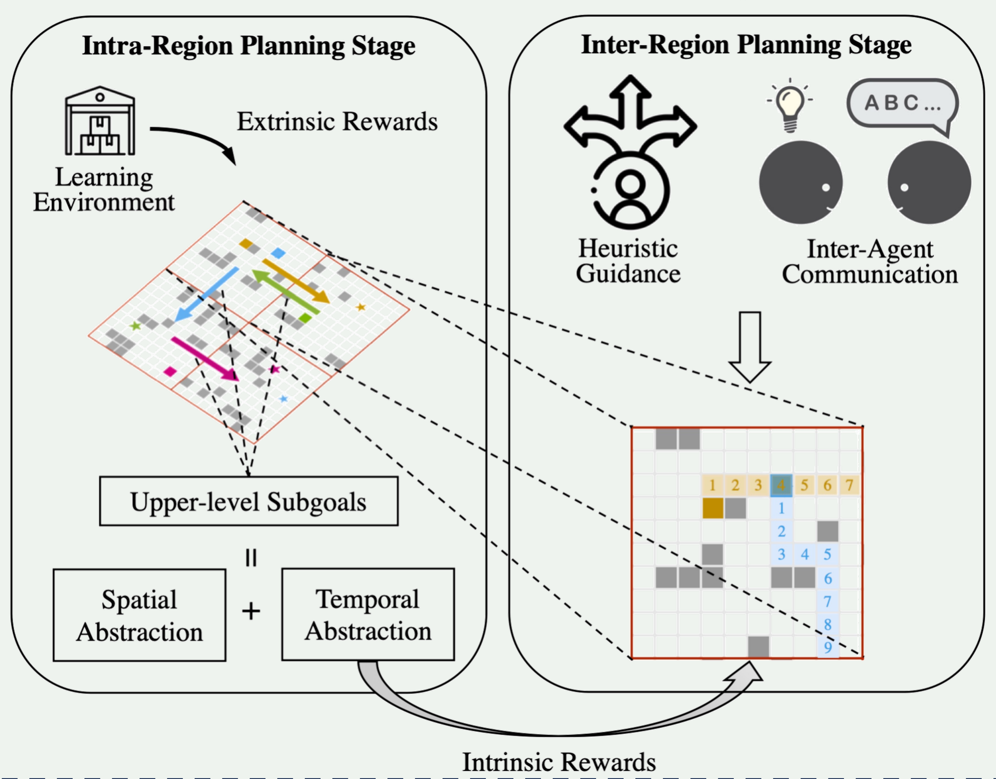
2. Partial Observability



Our Solution: Hierarchical Reinforcement Learning

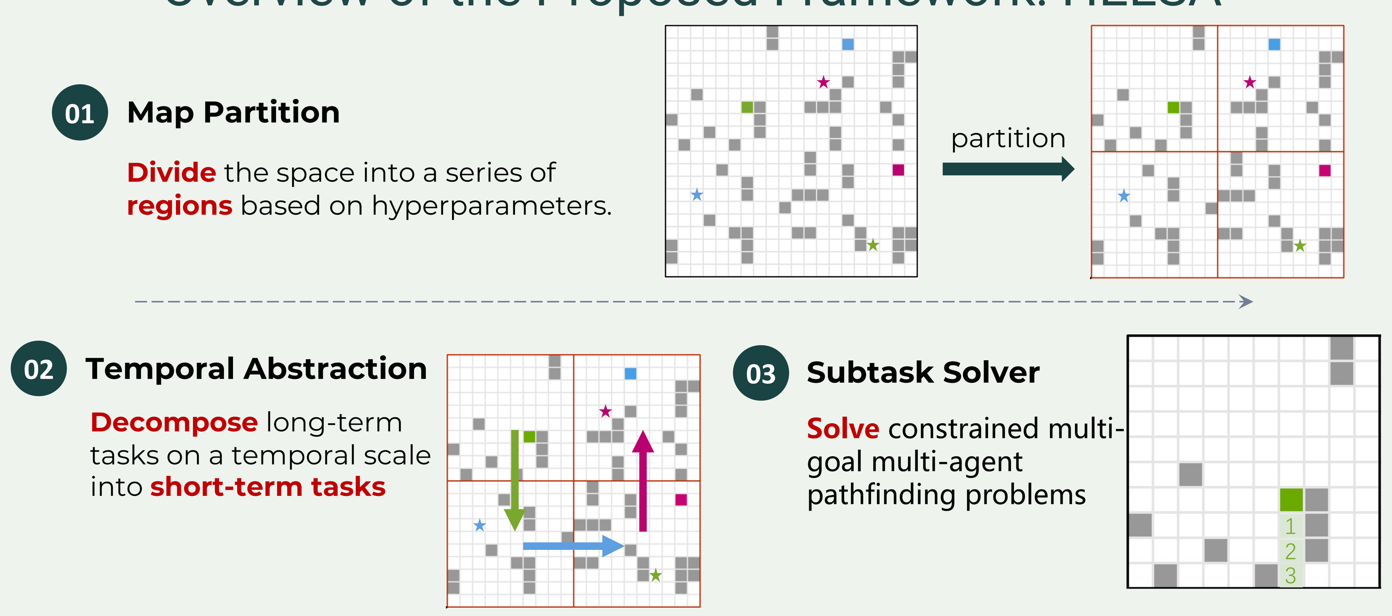
- **Meta-Controller: Intra-Region Planning**
  - Map's Meta Controller
  - Partitions maps into regions
  - Optimizes region-wise paths
- **Subpolicies: Inter-Region Planning**
  - Path's Fine-Tuner
  - Refines inter-region paths
  - Ensures collision-free navigation





Method

Overview of the Proposed Framework: HELSA



The Upper-Level Controller: IQL-based Subgoal Planner

- **Observation Representation**
  - Regional Agents Count
  - Regional Goals Count
  - Heuristic Maps
- **Subgoal (Macro-Action)**
  - $\mathcal{G}_i = \langle \mathcal{I}_{g_i}, \beta_{g_i}, \pi_{g_i} \rangle$
  - Denotes the next region for each agent to reach
- **Learning Scheme**
  - Independent Q-Learning
  - Multi-step bootstrapping
- **Distributed Priority ER**
  - Improve sample efficiency

Algorithm 1 IQL-based Subgoal Planner

```
1: Initialize the shared replay buffer M and the exploration probability ϵ = 1.
2: Initialize random parameters {θ, θ̃} for the evaluation DQN Q(r, g; θ) and the target DQN Q̃(r, g; θ̃), respectively.
3: for each episode do
4:   Initialize the lower-level state descirption s_i^0 and t_i (i ∈ {1, ..., M}) as the start and terminate states, respectively.
5:   Initialize the upper-level state description r_i and termination condition β_i (i ∈ {1, ..., M}).
6:   for each time step t do
7:     for each agent i do
8:       if s_i^t ∈ β_i then                  ▷ With a subgoal accomplished, the DQN parameters are updated.
9:         Obtain extrinsic reward f_i(r_i, g_i, r_i'), and store transition (r_i, g_i, f_i, r_i') in the global replay buffer M.
10:        Compute TD Target using θ̃, and perform gradient descent on θ to minimize multi-step TD error.
11:        Anneal ϵ and repalce θ̃ with θ periodically.
12:        if t_i ∉ β_i then                  ▷ If the target region has not yet been reached, assign a new subgoal.
13:          g_i ← EpsGreedy(r_i, G_i, ϵ, Q)
14:          β_i, r_i' ← ExpandGoal(g_i)
15:      Sample a primitive action a_i^t via the lower-level controller.
16:      Execute a_i^t and observe next state s_i^{t+1}.
17:      Obtain intrinsic reward f̃_i(s_i^t, a_i^t, s_i^{t+1}), and update low-level actor and critic parameters.
```

The Lower-Level Controller: Communication-based Inter-Region Planning Method

Observation Encoder

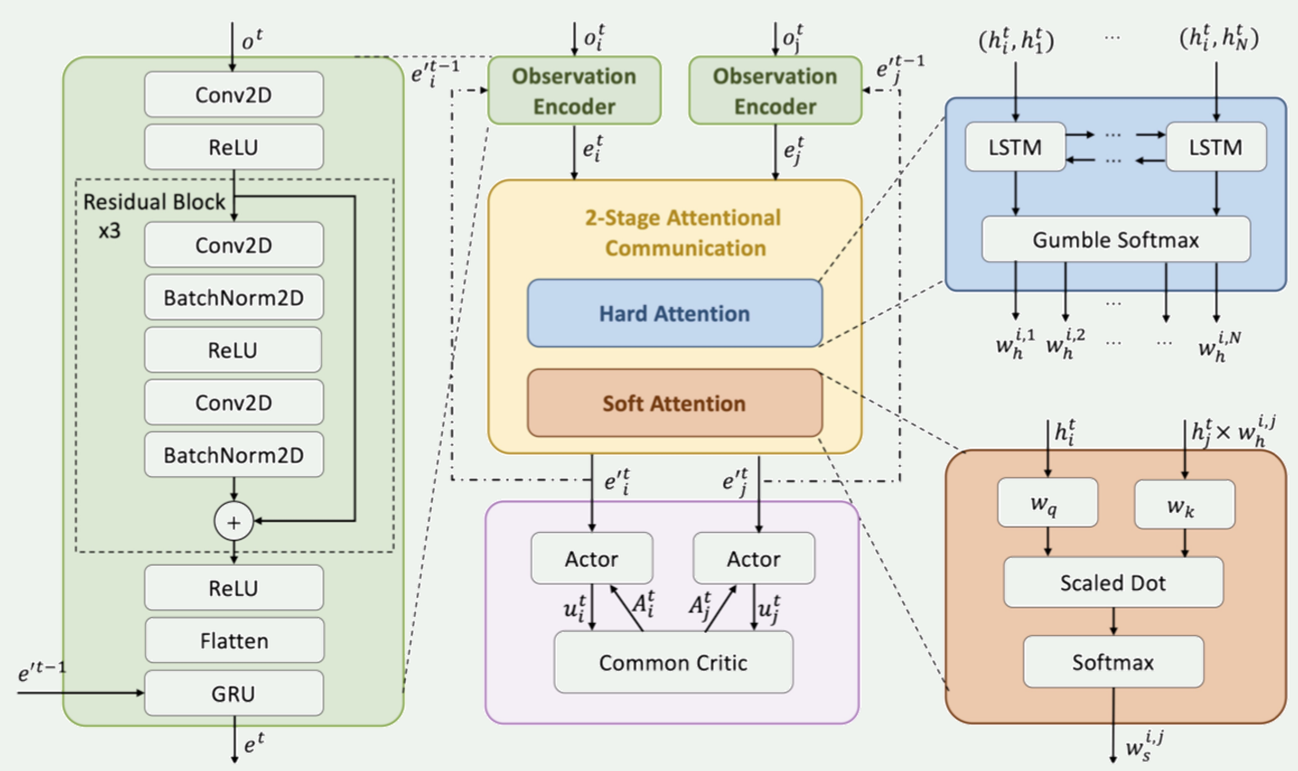
- Agents positions
- Obstacle positions
- Heuristic encodings
- Last-step messages

Communication Block

- Hard attention mechanism to filter out irrelevant agents
- Soft attention to calculate relative importance

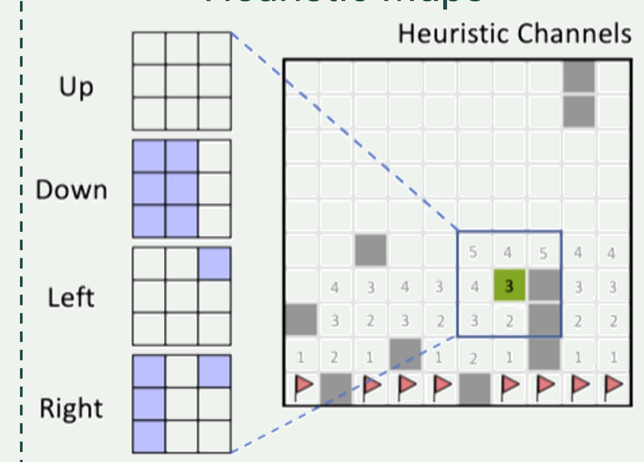
Action Network

- A common critic performs a reasonable credit assignment



- **Residual Neural Network Empowered Observation Encoder**
  - Extended from DHC [2], the heuristic channels are adopted considering all subgoals provide rich rational knowledge.
  - All heuristic maps can be computed and stored in advance.
  - The feature extractor has great generalization capability.
- **Two-Stage Attentional Communication Mechanism**
  - Each agent communicates with its neighboring agents.
  - Inspired from G2ANet [4], a two-stage attention is adopted.
  - The gumble-softmax is utilized to enable back propagation.
  - A query-key mechanism is employed to weigh relevance.
- **Training via Counterfactual Multi-Agent Policy Gradients**
  - COMA [5] is employed as our low-level learning scheme.
  - For each agent, an advantage function is computed
    $$A_i(\mathbf{s},\mathbf{a}) = Q(\mathbf{s},\mathbf{a}) - \sum_{a_i'} \pi_i(a_i \mid \tau_i) Q(\mathbf{s}, (\mathbf{a}_{-i}, \mathbf{a}_i'))$$
  - The centralized critic reasons the contribution of each agent.
  - A reasonable multi-agent credit assignment is achieved.

Heuristic Maps



Heuristic Maps for the Green Agent

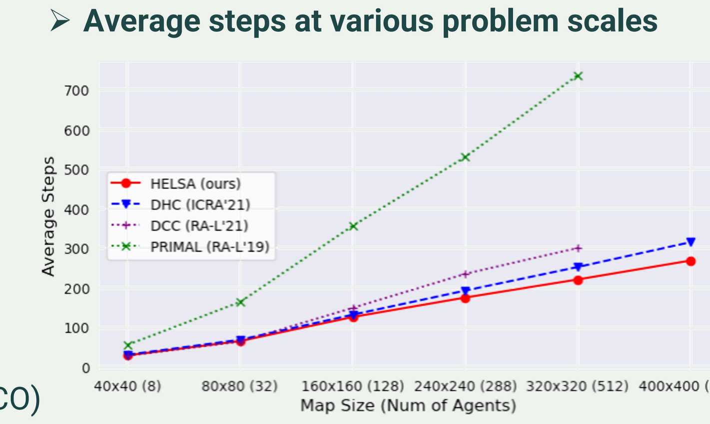
- The red flags are the temporary subgoals
- The gray grids represent obstacles.
- The four channels indicate whether the agent gets closer to its subgoals by taking a certain action at these locations within the agent's FOV respectively.

Empirical Evaluation

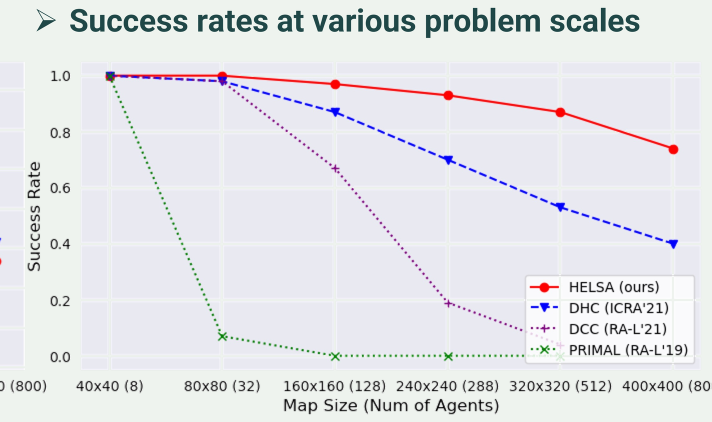
Performance Comparison of HELSA Against SOTA Learning-based Methods

- **Counterparts**
  - PRIMAL [1] (R-AL, 2019)
  - DHC [2] (ICRA, 2021)
  - DCC [3] (RA-L, 2021)
- **Evaluation Metrics**
  - Success Rates (SR)
  - Average Steps (AS)
  - Makespans (MS)
  - Collisions with Agents (CA)
  - Collisions with Obstacles (CO)

Average steps at various problem scales



Success rates at various problem scales



Overall performance comparison. All results are the average values of the 100 randomly generated test cases.

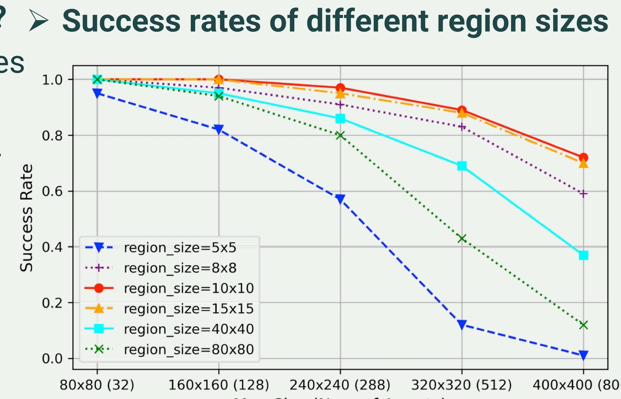
Model	8 agents, 40-sized map, 0.2 density					32 agents, 80-sized map, 0.2 density					128 agents, 160-sized map, 0.2 density				
	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓
PRIMAL [4]	**1.0**	56.49	98.90	0.42	**0.0**	0.88	164.39	305.73	4.12	**0.0**	0.07	356.51	1007.08	113.06	4.27
DHC [6]	**1.0**	31.40	55.77	0.38	**0.0**	0.98	69.18	139.77	3.20	**0.0**	0.87	132.31	399.19	29.38	0.06
DCC [7]	**1.0**	**28.84**	**50.49**	0.40	**0.0**	0.98	**64.47**	**134.34**	5.91	0.01	0.67	149.50	567.41	37.48	**0.0**
HELSA	**1.0**	29.71	52.29	**0.21**	**0.0**	**1.0**	65.85	136.17	**0.54**	**0.0**	**0.97**	**126.51**	**296.14**	**3.69**	**0.0**

Model	288 agents, 240-sized map, 0.2 density					512 agents, 320-sized map, 0.2 density					800 agents, 320-sized map, 0.2 density				
	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓	SR ↑	AS ↓	MS ↓	CA ↓	CO ↓
PRIMAL [4]	0.0	530.06	1536.0	593.59	34.48	0.0	736.50	2048.0	1498.20	173.49	-	-	-	-	-
DHC [6]	0.70	193.13	804.55	99.52	**0.01**	0.53	252.62	1304.48	236.22	0.30	0.40	315.08	1906.36	468.61	0.71
DCC [7]	0.19	235.32	1375.04	151.88	12.97	0.04	300.78	2020.76	423.40	57.41	-	-	-	-	-
HELSA	**0.93**	**175.56**	**629.58**	**49.41**	0.03	**0.87**	**221.17**	**935.99**	**101.78**	**0.04**	**0.74**	**268.83**	**211.15**	**269.67**	**0.37**

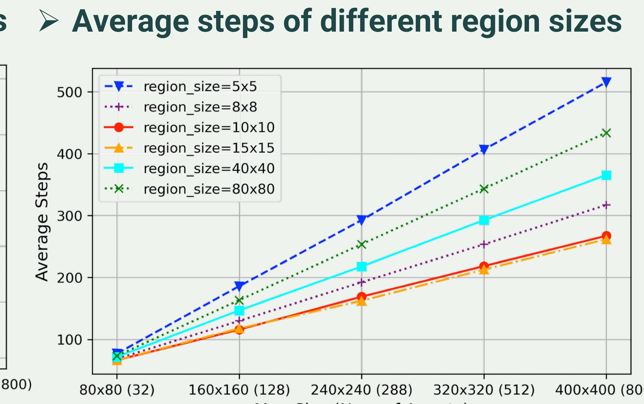
How does the partitioning granularity effect the performance of HELSA?

- **Why does the partitioning granularity matter?**
  - The significance of partitioning granularity lies in the trade-offs it presents.
  - Too small, the upper controllers are strained.
  - Too large, the hierarchical finesse is lost.
- **How does region size affect the solution?**
  - Optimal performance is achieved when the region size mirrors that of the agent's FOV.
  - Deviating from this size, leads to suboptimal results, forming a mountain-valley distribution.

Success rates of different region sizes



Average steps of different region sizes



Does the two-stage attention communication lead to better coordination?

Empirical evaluation of the adopted lower-level controller with other ablations in terms of success rates and average steps.

Method	w/ hierarchy?	80-sized map		160-sized map		240-sized map		320-sized map		400-sized map		Avg.	
		SR ↑	AS ↓	SR ↑	AS ↓	SR ↑	AS ↓	SR ↑	AS ↓	SR ↑	AS ↓	SR ↑	AS ↓
COMA+Comm +Attention	✓	**1.0**	**65.85**	**0.97**	**126.51**	**0.93**	**175.56**	**0.87**	221.17	0.74	268.83	**0.90**	**171.58**
		0.98	67.25	0.76	141.95	0.41	219.03	0.07	287.99	0.0	347.63	0.44	212.77
COMA+Comm	✓	**1.0**	66.78	0.95	130.20	0.90	182.13	0.86	**219.75**	**0.77**	**245.00**	0.90	172.97
		0.98	69.89	0.72	147.77	0.35	233.98	0.09	311.93	0.0	387.54	0.43	230.22
COMA	✓	0.95	96.30	0.83	193.39	0.44	323.95	0.04	433.73	0.0	615.19	0.45	332.51
		0.90	139.13	0.43	248.67	0.12	477.53	0.01	633.55	0.0	883.10	0.29	476.40

Conclusions

Conclusions

- We propose the HELSA framework to tackle the problem of sparse reward and long horizon in large-scale multi-agent pathfinding problems.
- Experiments show that our approach performs significantly better in large-scale multi-robot routing tasks in success rates, makespans, and collision rates than state-of-the-art learning-based planners.
- The key idea of our hierarchical framework is beneficial to a number of similar problems with long horizon in terms of time and large scale in terms of space.

Future Work

- In the future, we will evaluate our framework in different experimental scenarios, especially those with different agent and obstacle densities.
- We are interested in extending it from a discrete grid world to a continuous one.
- Experiments in real-world multi-robot systems are also on the agenda.

References

1. G. Sartoretti, J. Kerr, Y. Shi, et al., "Primal: Pathfinding via reinforcement and imitation multi-agent learning," IEEE Robotics and Automation Letters, vol. 4, no. 3, pp. 2378–2385, 2019.
2. Z. Ma, Y. Luo, and H. Ma, "Distributed heuristic multi-agent path finding with communication," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.
3. Z. Ma, Y. Luo, and J. Pan, "Learning selective communication for multi-agent path finding," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 1455–1462, 2021.
4. Y. Liu, W. Wang, Y. Hu, et al., "Multi-agent game abstraction via graph attention neural network," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
5. J. Foerster, G. Farquhar, T. Afouras, et al. "Counterfactual multi-agent policy gradients." in Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.