Instructing Goal-Conditioned Reinforcement Learning Agents with Temporal Logic Objectives

NeurIPS 2023

Rutgers University

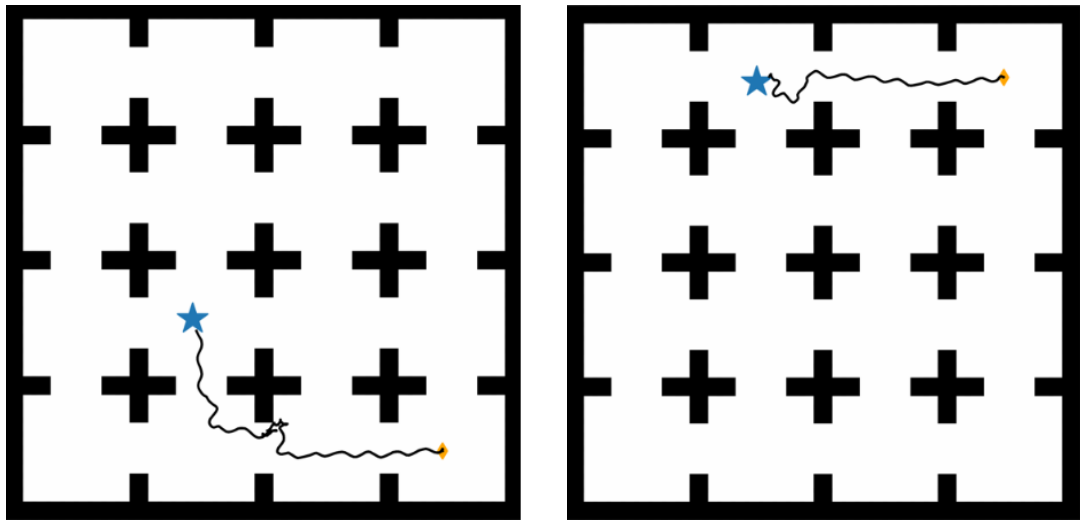
Wenjie Qiu* Wensen Mao* He Zhu

wq37@cs.rutgers.edu wm300@cs.rutgers.edu hz375@cs.rutgers.edu

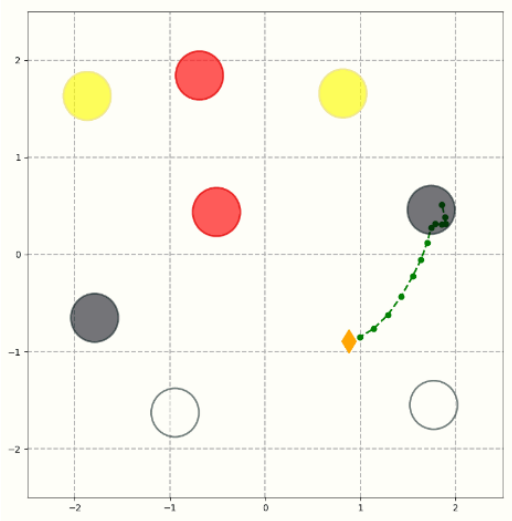
RUTGERS

Goal-Conditioned Reinforcement Learning (GCRL)

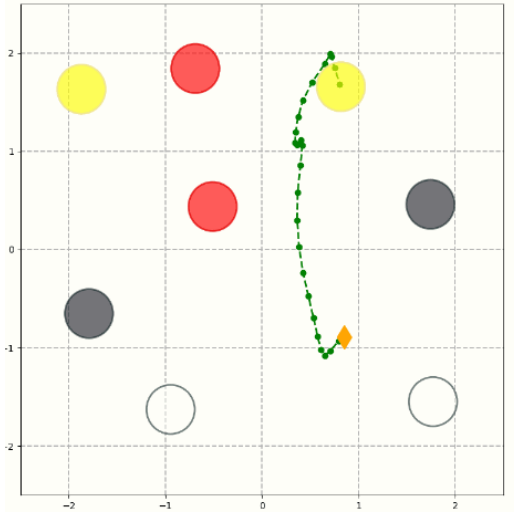
- Reach arbitrary goals in the goal space from initial environment states



Reach ★ with random start ♦



Go to JetBlack ● zone

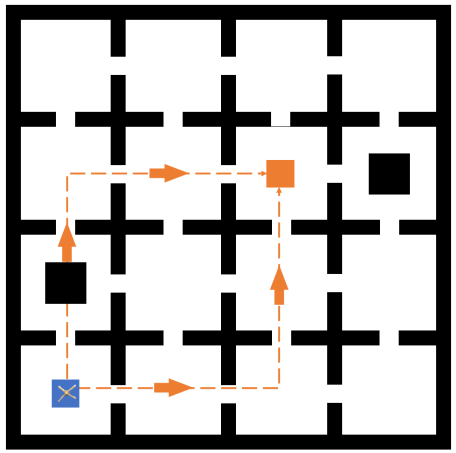


Go to Yellow ● zone

Linear Temporal Logic (LTL)

- Define a sequence of tasks in the chronological order
- May include infinite loop task (ω-regular property)

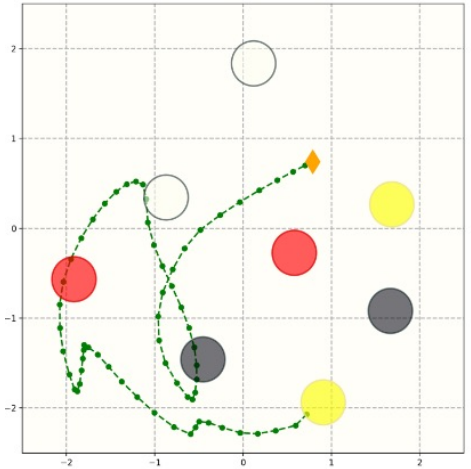
Ant Maze

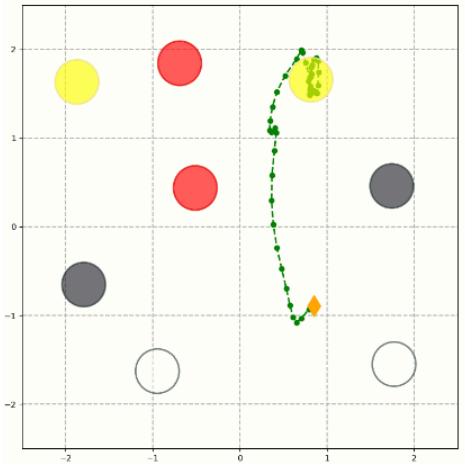


$$F\left(\left(room(0,2) \vee room(2,0)\right) \wedge F\ room(2,2)\right)$$

Reach orange room via one possible orange path

ZoneEnv

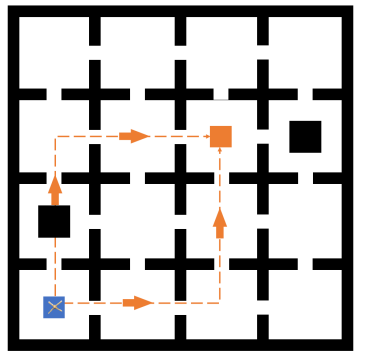




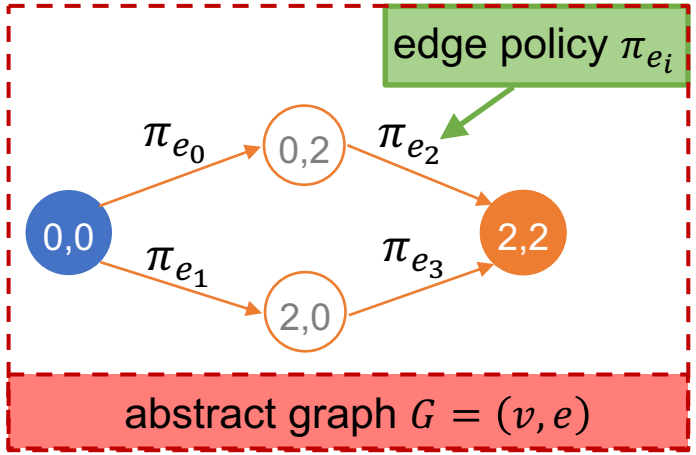
$F($ ● $\wedge\ F($ ○ $\wedge\ F($ ◎ $\wedge\ F$ ● $))$

GF ●

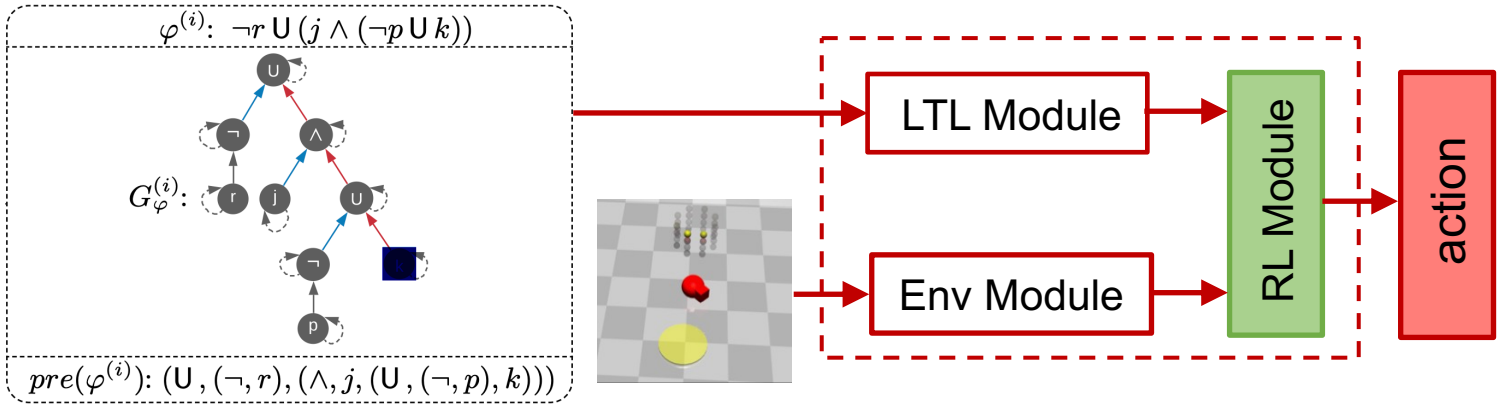
Prior approaches



$F\left(\left(room(0,2) \vee room(2,0)\right) \wedge F\ room(2,2)\right)$

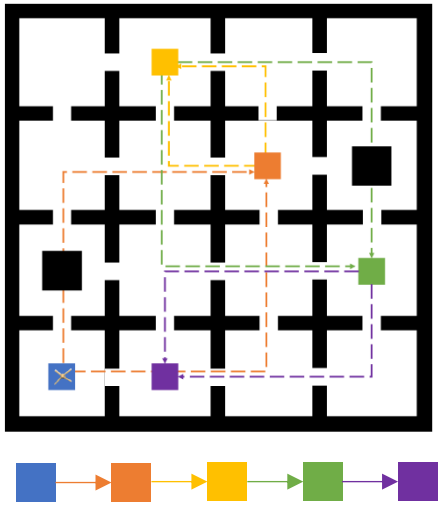


- DiRL [Jothimurugan et al. 2021]
- Generate abstract graph from LTL specification and train independent policies for each edge to achieve sub-goals
- Inefficient and lack of generalizability



- LTL2Action [Vaezipoor et al. 2021]
- Encode LTL formulas as syntax trees, enable RL agent to learn LTL-conditioned policies
- Unable to generalize to arbitrary LTL formulas

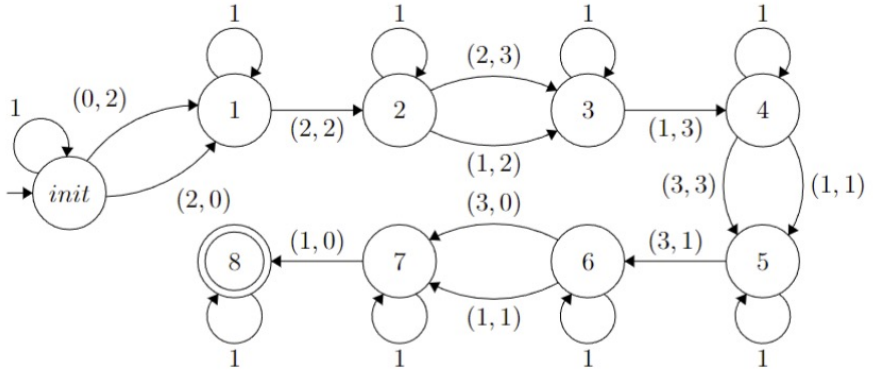
GCRL-LTL algorithm framework



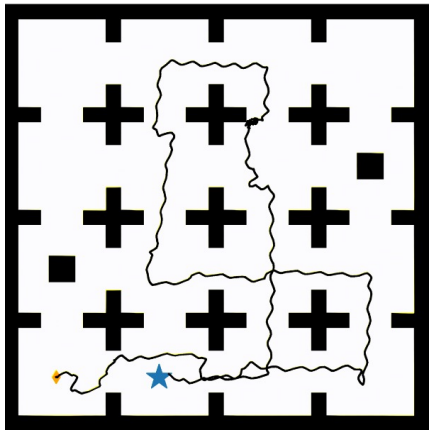
LTL specification ϕ

Buchi automata

abstract graph

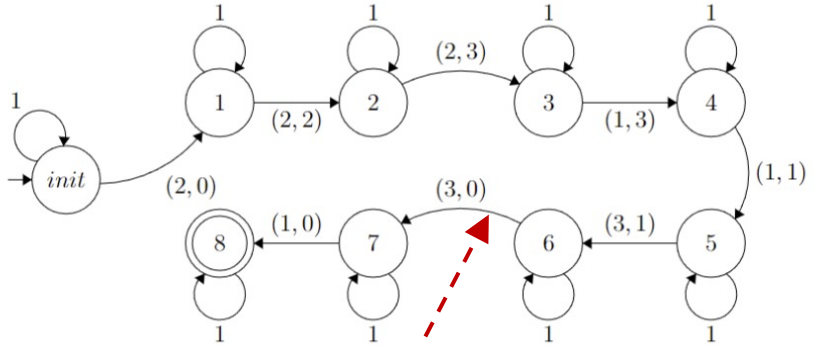


optimal path finding



execute with goal-conditioned policy

$\pi(a|s,\varphi)$



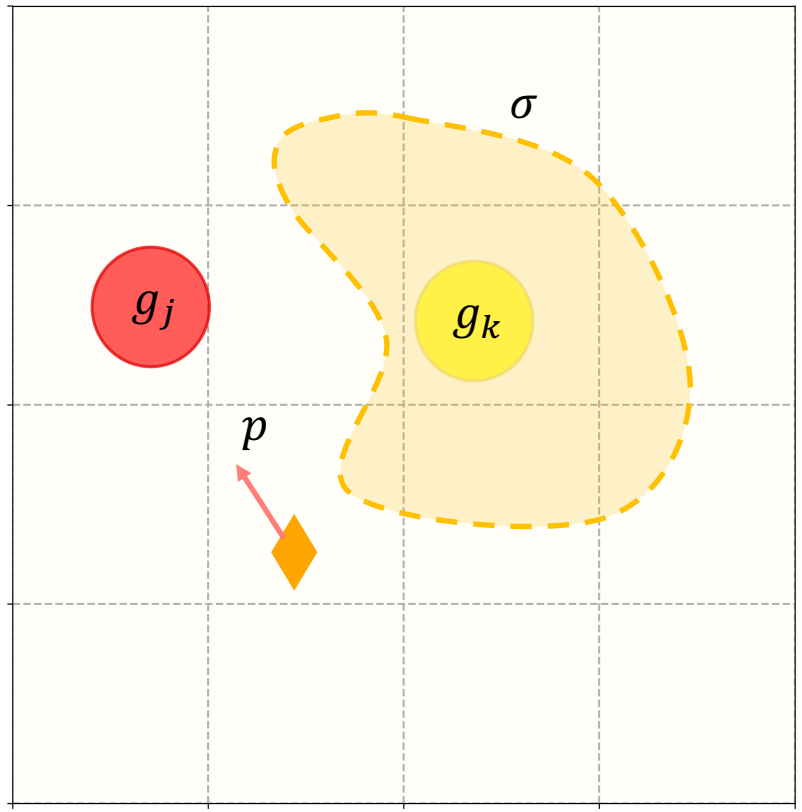
$\mathrm{w}_{\mathrm{edge}} = -\log \mathcal{V}(\mathrm{s}_0, room_{(3,1)}, room_{(3,0)})$

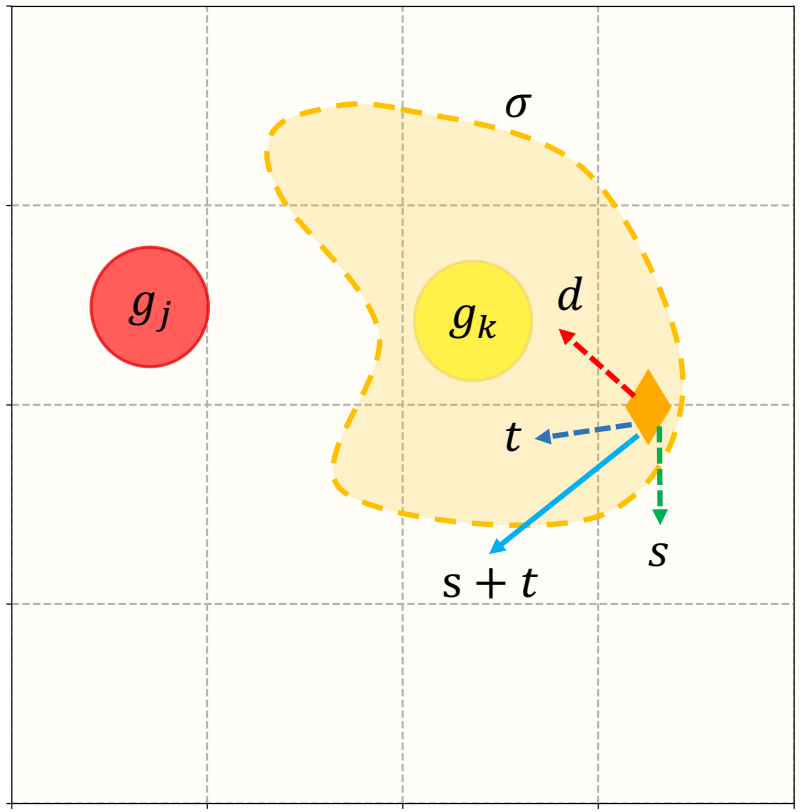
$$\mathcal{V} = \min_{\mathcal{V}} \left(V(s_t, g_k) - \mathcal{V}(s_j, g_t, g_k)\right)^2 \quad \text{where } g_k \in L(s_k) \wedge g_t \in L(s_t)$$

with $\tau \sim B$ (a replay buffer), $t \sim \{0 \dots t_{max}\}$, $s_t, a_t, s_{t+1} \sim \tau$, $j \sim \{0, t\}$, $k \sim \{t+1 \dots t_{max}\}$, and L as the state labeling function. We train $\mathcal{V}$ together with a goal-conditioned learning algorithm.

Goal Reaching and Obstacle Avoidance

- For example, reach-avoid task on one edge : $\neg$ U





Goal Obstacle

$$\widetilde{\pi}(\cdot|s, \bigwedge_j g_j, \bigwedge_k \neg g_k) \equiv$$

let $k = \arg\max_k V(s, g_k)$ **in**

if $V(s, g_k) < \sigma$ *reaching action p*

then $\arg\max_a \min_j Q^\pi(s, g_j, a)$

else

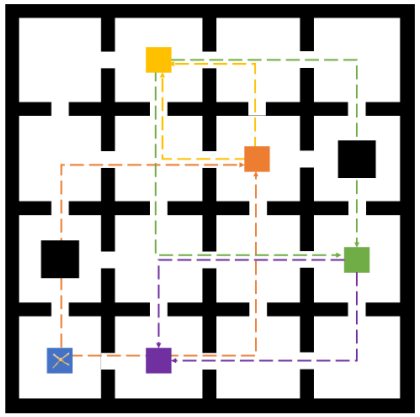
let $d = \arg\max_a Q(s, g_k, a)$

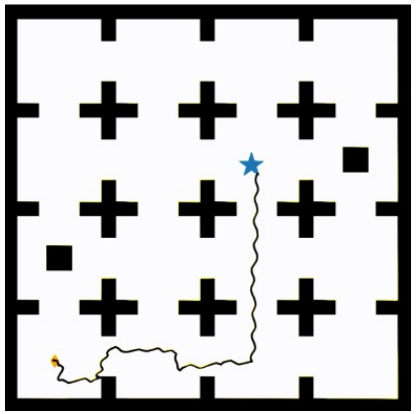
let $s = \arg\min_a Q(s, g_k, a)$

let $t = \arg\max_{a \neq d} \min_j Q^\pi(s, g_j, a)$

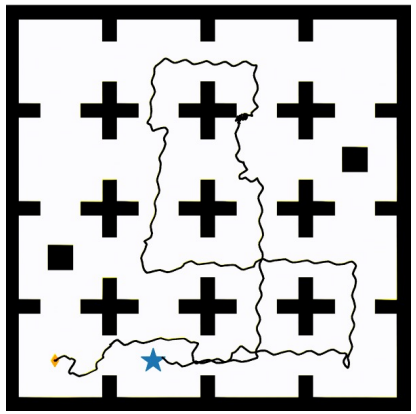
$s + t$

Experiments on Ant16rooms

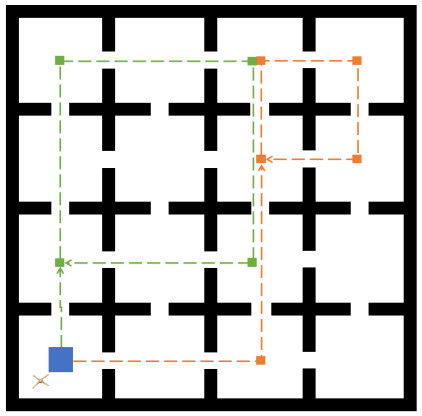


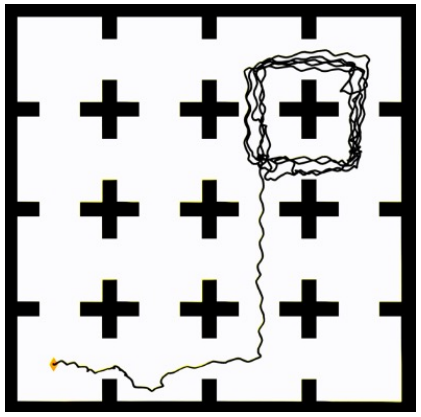


$\phi_2 =$ 



$\phi_5 =$ 



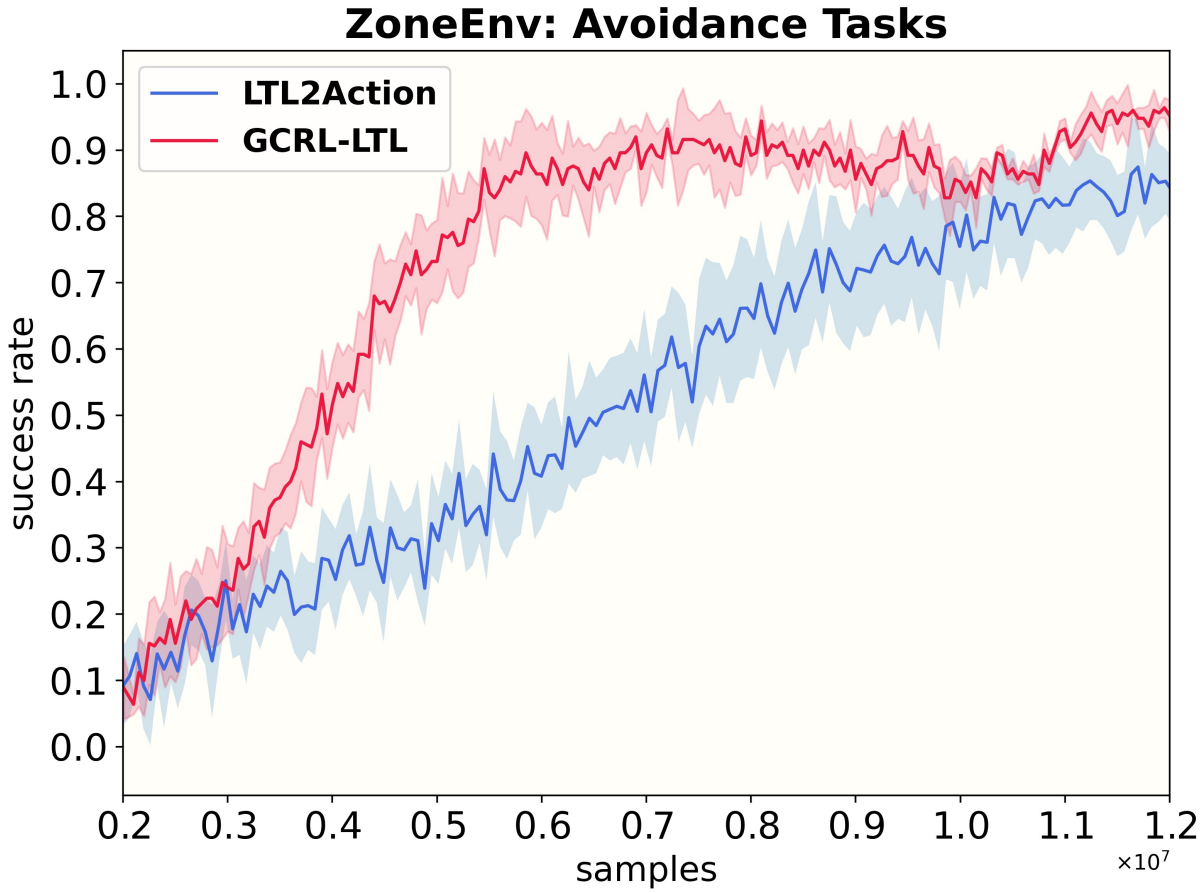


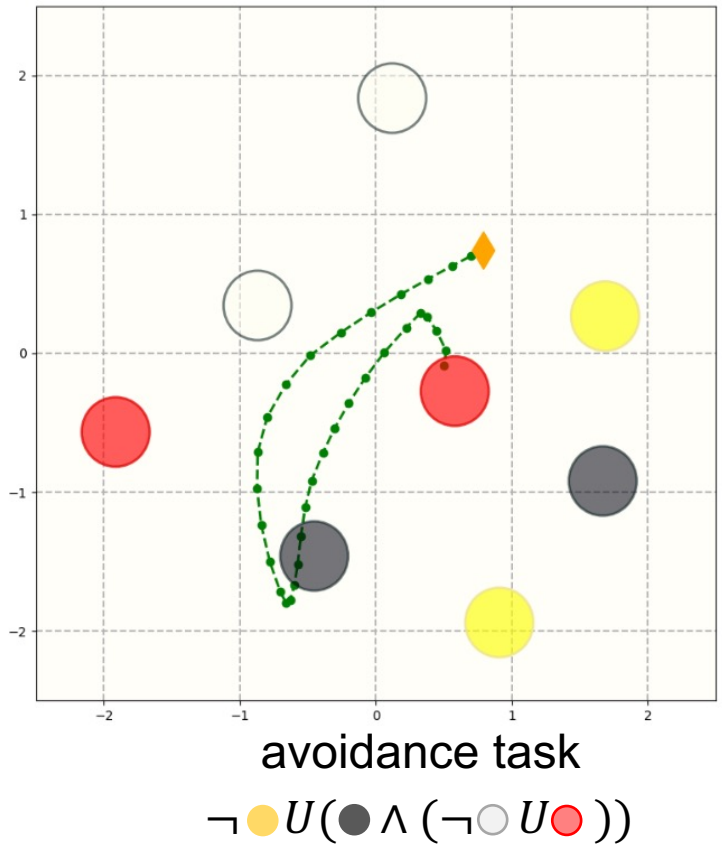
$\phi_8 =$

Choose to execute the loop with a smaller path cost!

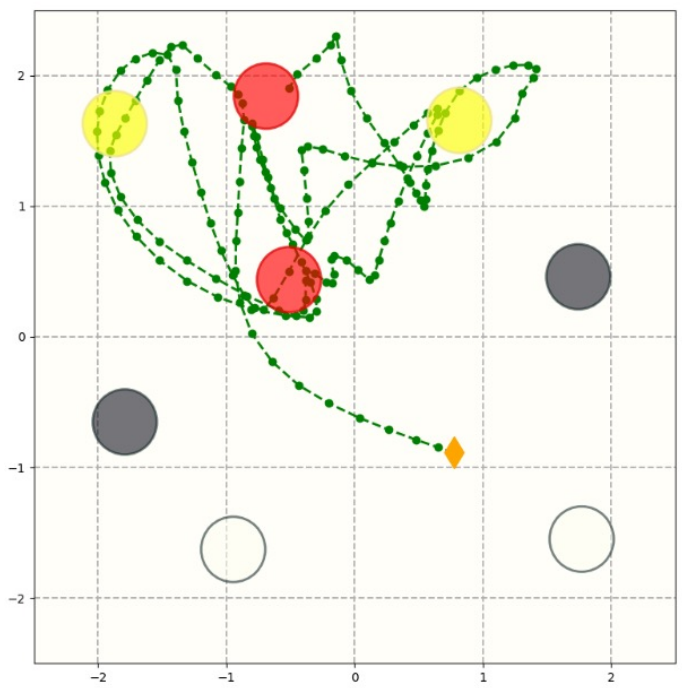
Methods	DiRL	DiRL+ GCSL	Ours
ϕ_1	0.910 (0.022)	0.923 (0.082)	0.967 (0.006)
ϕ_2	0.770 (0.083)	0.953 (0.037)	0.925 (0.049)
ϕ_3	0.367 (0.147)	0.967 (0.017)	0.935 (0.028)
ϕ_4	0.183 (0.046)	0.937 (0.031)	0.875 (0.041)
ϕ_5	0.043 (0.061)	0.913 (0.017)	0.868 (0.038)
ω ϕ_6	/	/	0.857 (0.004)
ω ϕ_7	/	/	0.882 (0.018)
ω ϕ_8	/	/	0.903 (0.045)

Experiments on ZoneEnv

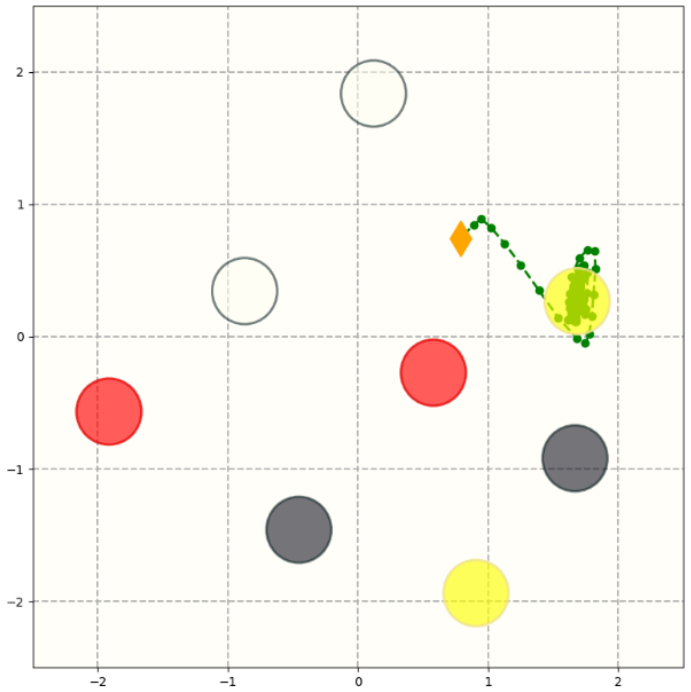




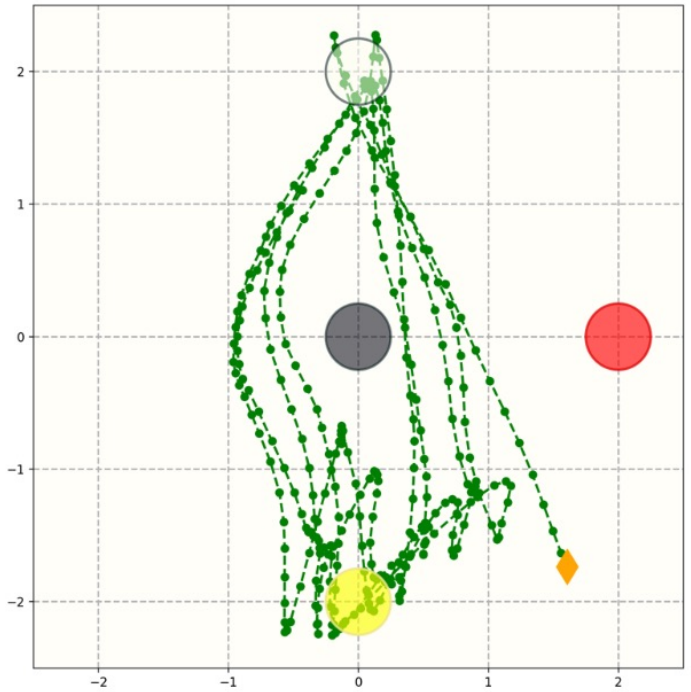
More complex experiments on ZoneEnv



$GF(\bigcirc \wedge XF \bigcirc) \wedge G(\neg \bigcirc)$



$FG \bigcirc$



$GF \bigcirc \wedge GF \bigcirc \wedge G(\neg \bigcirc)$