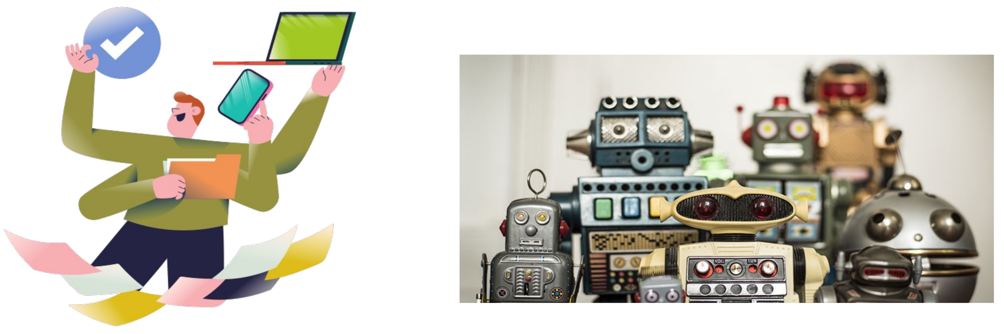
HypRL: Reinforcement Learning of Control Policies for Hyperproperties

Tzu-Han Hsu*, Arshia Rafieioskouei*, Borzoo Bonakdarpour
Michigan State University

Paper

Research Question

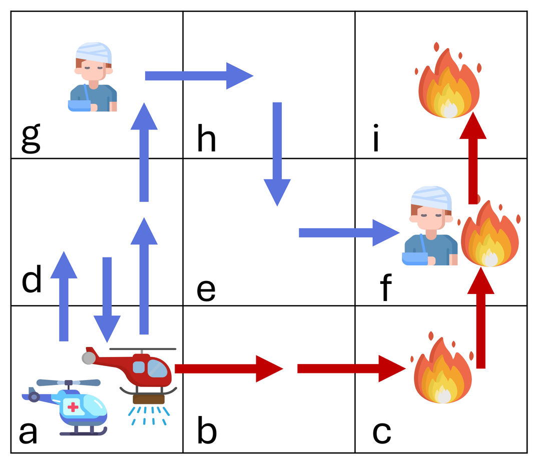


Multi-Objective — Multi-Agent

How to shape **Optimal Reward Functions**?

Motivating Example: A wildfire scenario

A **possible** control policy with reward function R1

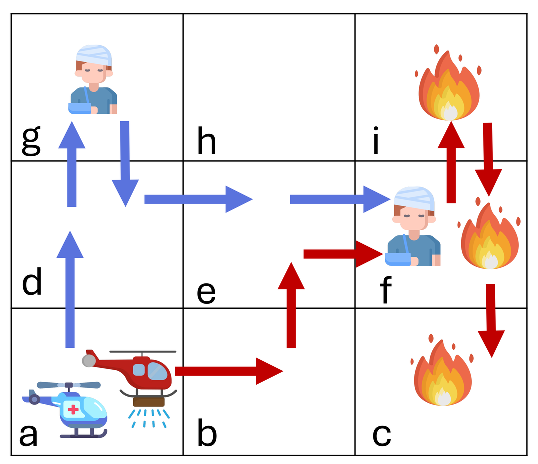


R1(Extinguish fire) =	+50
R1(Rescue victim) =	+10
R1(Our of range) =	-100
R1(MED in fire) =	-100

a – b – c – f – i

a – d – a – d – g – h – e – f

An **optimal** control policy with reward function R2



R2(Extinguish fire) =	+10
R2(Rescue victim) =	+50
R2(Our of range) =	-100
R2(MED in fire) =	-100

a – d – e – f – i – f – c

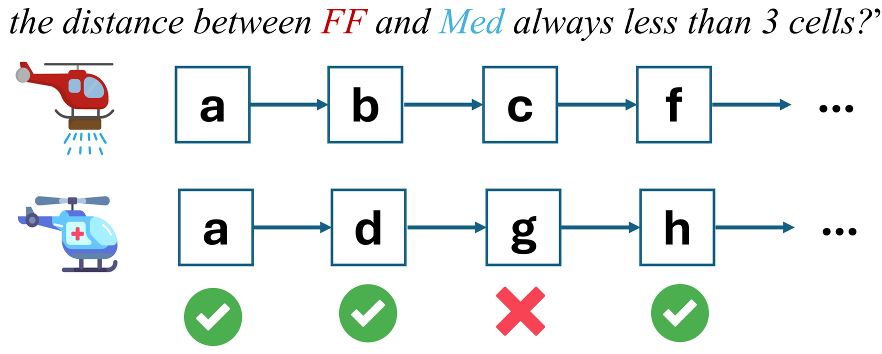
a – d – g – d – e – f

Specifications as Hyperproperties

- **Hyperproperties** characterize requirements over sets of execution traces, allowing the specification of behaviors of multi-agent.

Example:

"Is the distance between FF and Med always less than 3 cells?"



- We use hyperproperties, expressed in the temporal logic **HyperLTL**, to achieve **specification-guided RL** for multi-agent w.r.t multi-objective and relational constraints.

The HyperLTL specification for the wildfire scenario:

$$\varphi_{\text{Rescue}} \triangleq \forall \tau_1 . \exists \tau_2 . (\psi_{\text{fire}} \wedge \psi_{\text{save}} \wedge \psi_{\text{dist}} \wedge \psi_{\text{safe}})$$

(Extinguish fire)	O_1 :	$\psi_{\text{fire}} \triangleq \Diamond(i_{\tau_1}) \wedge \Diamond(f_{\tau_1}) \wedge \Diamond(c_{\tau_1})$
(Rescue victim)	O_2 :	$\psi_{\text{save}} \triangleq \Diamond(g_{\tau_2}) \wedge \Diamond(f_{\tau_2})$
(Our of range)	C_1 :	$\psi_{\text{dist}} \triangleq \Box(\lvert \text{Location}_{\tau_1} - \text{Location}_{\tau_2} \rvert < 3)$
(MED in fire)	C_2 :	$\psi_{\text{safe}} \triangleq (\neg i_{\tau_2} \,\mathcal{U}\, i_{\tau_1}) \wedge (\neg f_{\tau_2} \,\mathcal{U}\, f_{\tau_1}) \wedge (\neg c_{\tau_2} \,\mathcal{U}\, c_{\tau_1})$

Problem Statement

Given an MDP $\mathcal{M}$ with unknown transitions and a HyperLTL formula φ of the form $\mathbb{Q}_1\tau_1 \ldots \mathbb{Q}_n\tau_n . \psi$, our goal is to identify a tuple of n policies $\langle \pi_1^\star, \ldots, \pi_n^\star \rangle$, such that:

$$\langle \pi_i^\star \rangle_{i \in \{1,\ldots,n\}} \in \left[\underset{\langle \pi_i \rangle}{\arg\max}\, \mathbb{P}\Big[\langle \mathsf{Traces}(\mathcal{Z}_{\pi_i} \sim \mathcal{D}_{\pi_i}) \rangle \models \varphi \Big] \right]_{i \in \{1,\ldots,n\}}$$

Our Solutions to the Main Challenges

(1) We apply **Skolemization** to resolve **quantifier alternations** in a HyperLTL formula.

$$\mathbf{Skolem}(\varphi) = \underbrace{\exists \mathrm{f}_i(\tau_{i_1}, \ldots, \tau_{i_{|\mathbb{Q}_i^\forall|}}) .}_{\text{for each } i \in \mathbb{Q}^\exists} \quad \underbrace{\forall \tau_j .}_{\text{for each } j \in \mathbb{Q}^\forall} \quad \mathbf{Skolem}(\psi)$$

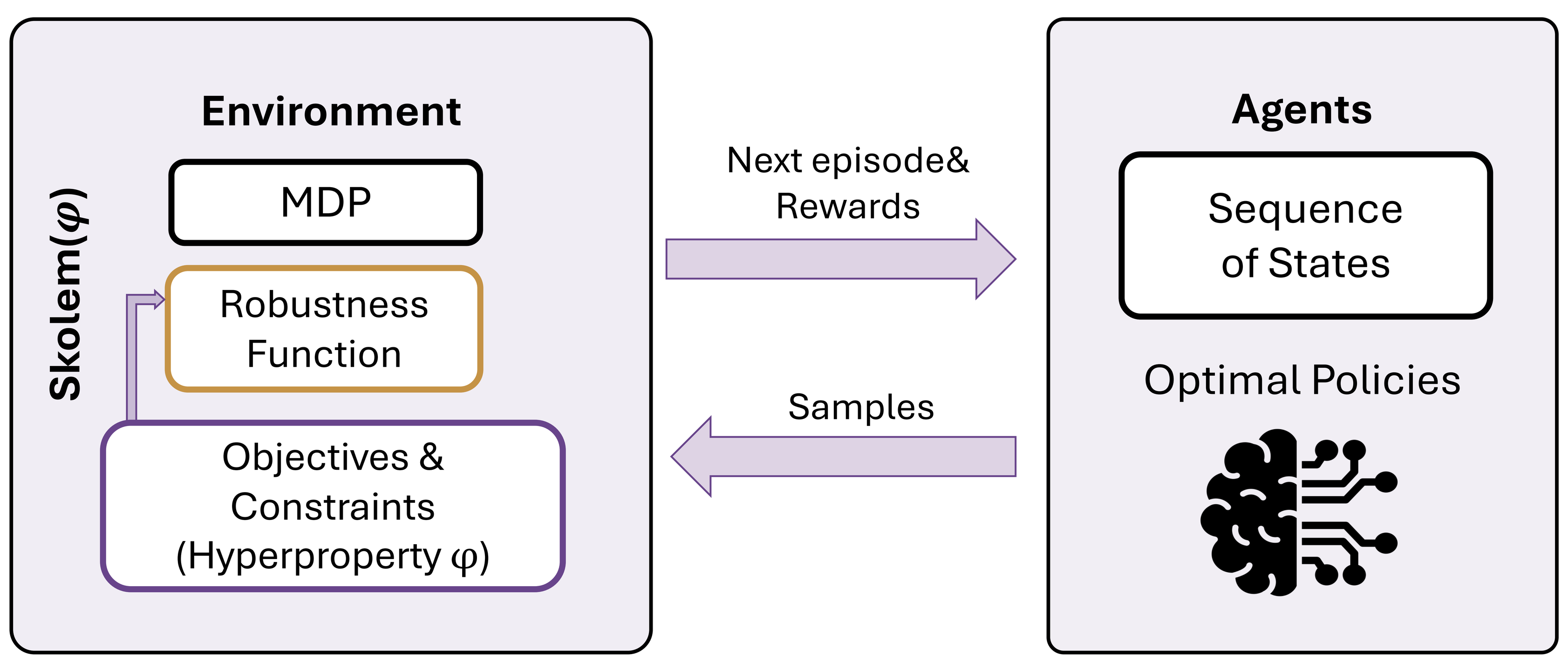
(2) We define **quantitative semantics** by min-max to **interpret temporal satisfaction**.

$$
\begin{aligned}
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi) &= \rho_{min} \text{ if } \mathsf{Tr}(\zeta_{[\ell:\ldots]}) = \epsilon \text{ and } \rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi) \text{ otherwise.} \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \mathsf{true}) &= \rho_{max} \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), f(L(s_\ell) < c)) &= c - f(L(s_\ell)) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \neg\psi) &= -\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \bigcirc\psi) &= \rho(\mathsf{Tr}(\zeta_{[\ell+1:k]}), \psi) \text{ if } (k > \ell). \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \Box\psi) &= \min_{i \in [\ell,k)} \rho(\mathsf{Tr}(\zeta_{[i:k]}), \psi) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \Diamond\psi) &= \max_{i \in [\ell,k)} \rho(\mathsf{Tr}(\zeta_{[i:k]})\psi) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_1 \wedge \psi_2) &= \min(\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_1)\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_2)) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_1 \vee \psi_2) &= \max(\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_1)\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_2)) \\
\rho(\mathsf{Tr}(\zeta_{[\ell:k]}), \psi_1 \,\mathcal{U}\, \psi_2) &= \max_{i \in [\ell,k)} \Big(\min\Big(\rho(\mathsf{Tr}(\zeta_{[i:k]}), \psi_2), \min_{j \in [\ell,i)} \rho(\mathsf{Tr}(\zeta_{[j:i]}), \psi_1)\Big)\Big)
\end{aligned}
$$

(3) We **inductively construct policies** to handle **dependencies** among multi-agent.

- For each $j \in \mathbb{Q}^\forall$, $\quad \pi_j^\star(\zeta_{j\,[0:k]}) \triangleq \mathcal{NN}_j^\star(s_k)$
- For each $i \in \mathbb{Q}^\exists$, $\quad \pi_i^\star(\zeta_{i\,[0:k]}) \triangleq \mathcal{NN}_i^\star\big(\mathrm{f}_i(\mathsf{Tr}(\zeta_{i_1\,[0:k]}), \ldots, \mathsf{Tr}(\zeta_{i_{|\mathbb{Q}_i^\forall|}\,[0:k]}))\big)$

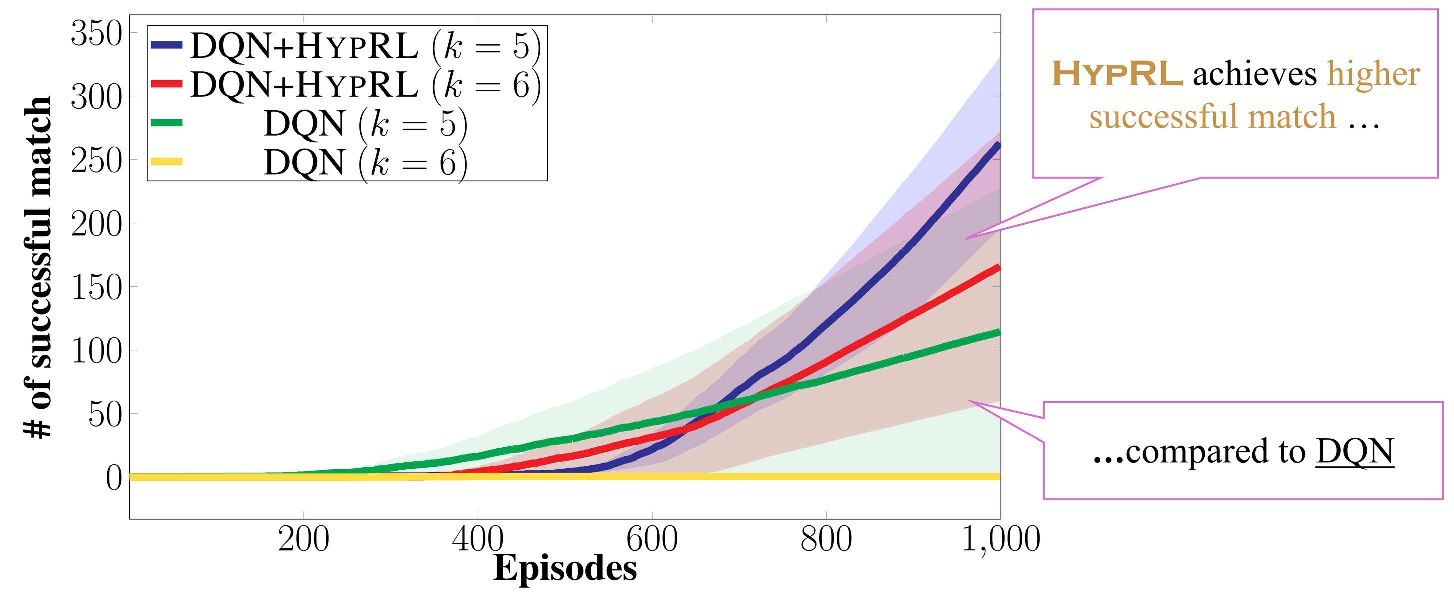
HypRL Architecture



Case Studies & Results

We investigate **4 popular RL benchmarks** and compared with DQN, PPO, CQ-learning, and Shielding.

Post Correspondence Problem (PCP)



Wildfire Scenario

HypRL achieves all objectives while obeying relational constraints in fewer steps compared to PPO.

Size	Method	Dist	Steps O_1	Steps O_2
3^2	PPO	2.5±0.01	33.43±4.1	787.03±31.8
	PPO + HypRL	**2.30**±0.03	**18.940**±1.1	**143.550**±1.1
5^2	PPO	4.2±0.01	62.7±9.7	S/B
	PPO + HypRL	**2.1**±0.05	**59.50**±14.3	**8057.5**±121.4
8^2	PPO	11.2±0.03	16801.8±2144.0	S/B
	PPO + HypRL	**6.94**±0.07	**4149.6**±1743.1	**386.2**±80.5
10^2	PPO	10.9±0.01	S/B	29023.6±976.4
	PPO + HypRL	**5.3**±0.10	**21272.8**±3579.0	**570.3**±52.2

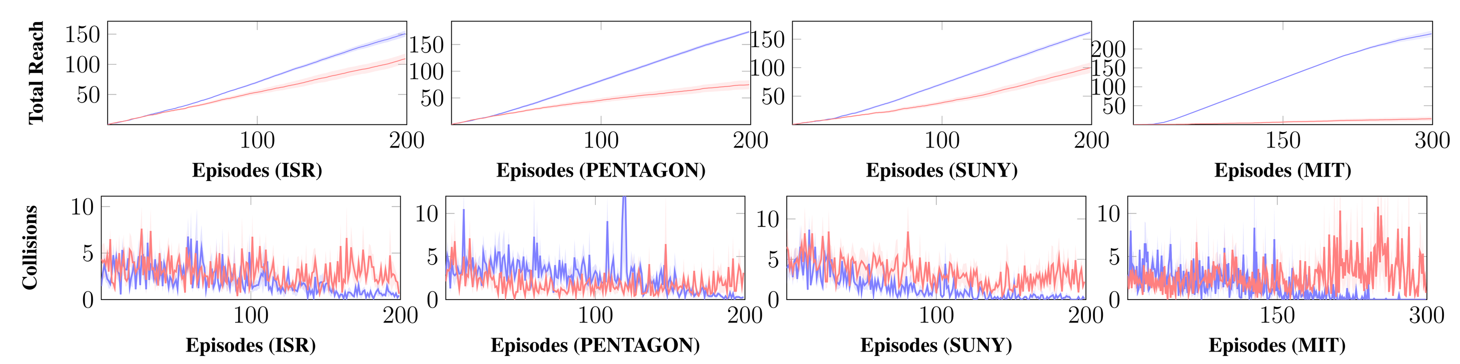
Deep Sea Treasure

HypRL finds more treasures in less steps, compared to PPO.

Method	Epi.	∑ (treasure)	∑ (treasure)/*steps*	Method	Epi.	∑ (treasure)	∑ (treasure)/*steps*
PPO	500	4.31±1.2	0.17±0.04	DQN	500	1.08±0.2	0.05±0.00
PPO + HypRL	500	**22.93**±2.2	**0.91**±0.08	DQN + HypRL	500	**4.12**±1.4	**0.14**±0.05
PPO	1000	3.22±0.8	0.12±0.03	DQN	1000	1.45±0.2	0.02±0.00
PPO + HypRL	1000	**23.97**±2.2	**1.12**±0.08	DQN + HypRL	1000	**4.43**±0.8	**0.21**±0.03

Safe RL

HypRL achieves goal in less steps while maintaining low collision rate compared to CQ-learning and Shielding.



		CQ		CQ + Shield ElSayed-Aly et al. (2021)		CQ + HypRL	
Maps	No. Agents	Steps	Collisions	Steps	Collisions	Steps	Collisions
ISR	2	27.95±7.4	0.19±0.1	17.40±2.2	**0.00**±0.0	**7.58**±0.3	0.25±0.2
Pentagon		36.46±7.7	0.28±0.1	75.20±12.6	**0.00**±0.0	**11.90**±4.6	0.53±0.5
SUNY		11.99±0.5	0.01±0.0	**11.50**±0.3	**0.00**±0.0	12.48±0.6	**0.00**±0.0
MIT		41.28±8.5	0.20±0.1	33.46±3.4	**0.00**±0.0	**23.20**±0.5	**0.00**±0.0
ISR	3	98.79±0.8	12.68±3.8	S/B	**0.00**±0.0	**74.18**±5.1	7.78±1.0
Pentagon		97.15±2.4	16.46±7.2	S/B	**0.00**±0.0	**78.82**±1.7	10.92±1.4
SUNY		84.89±7.9	0.63±0.2	82.35±4.1	**0.00**±0.0	**44.95**±8.3	0.71±0.4
MIT		96.96±1.8	2.83±1.3	S/B	**0.00**±0.0	**71.53**±7.7	1.58±0.7

Take aways!

- **Hyperproperties** naturally express both objectives and relational constraints in multi-agent RL.
- **HypRL** automatically converts any HyperLTL specification into a reward function.
- **HypRL** consistently outperforms baseline reward designs.