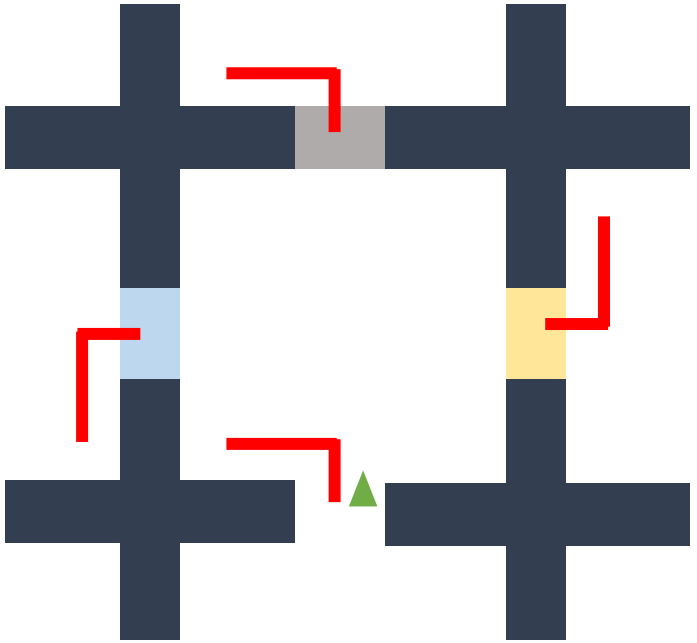
Robust Subtask Learning for Compositional Generalization

Kishor Jothimurugan, Steve Hsu, Osbert Bastani and Rajeev Alur

ICML 2023

Motivating Example

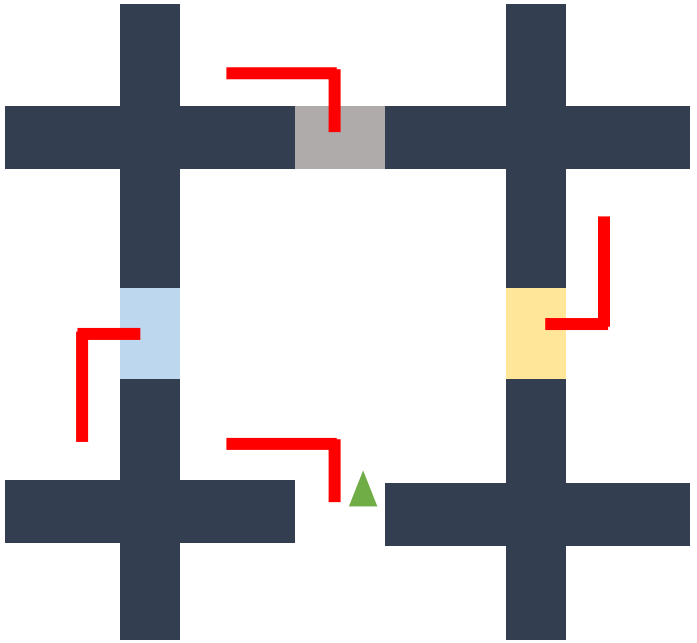
- Three subtasks: LEFT, RIGHT and UP
- Each exit leads to a new room
- Example task: LEFT -> RIGHT-> UP-> RIGHT



Subtask

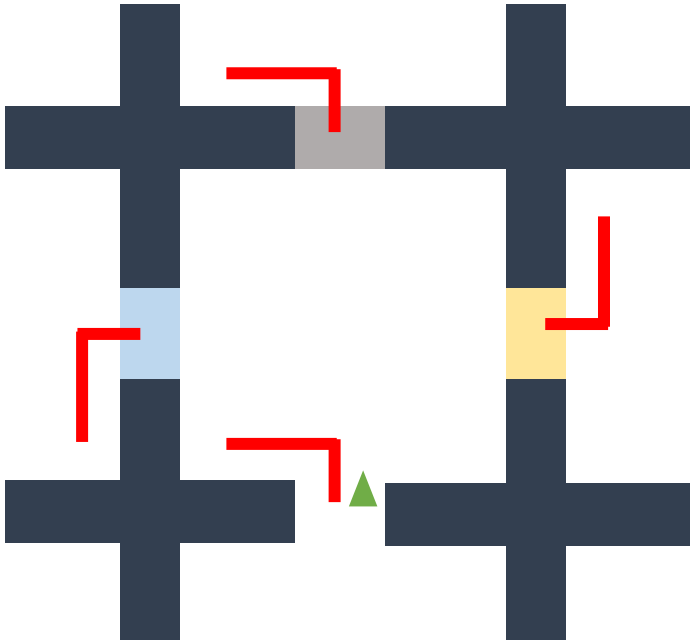
- Subtask $\sigma = (R, F, T)$
- Reward function $R: S \times A \times S \rightarrow \mathbb{R}$
- Final states $F \subseteq S$
- Jump transition $T: S \times F \rightarrow [0,1]$

$T(s'|s)$ is the probability of jumping to s' upon reaching final state s.



Subtask

- Subtask $\sigma = (R, F, T)$
- F_{left} = Blue region
- $R_\sigma(s, a, s') = -\|s' - c_\sigma\|_2 + B \cdot \mathbb{1}(s' \in F_\sigma)$



Task

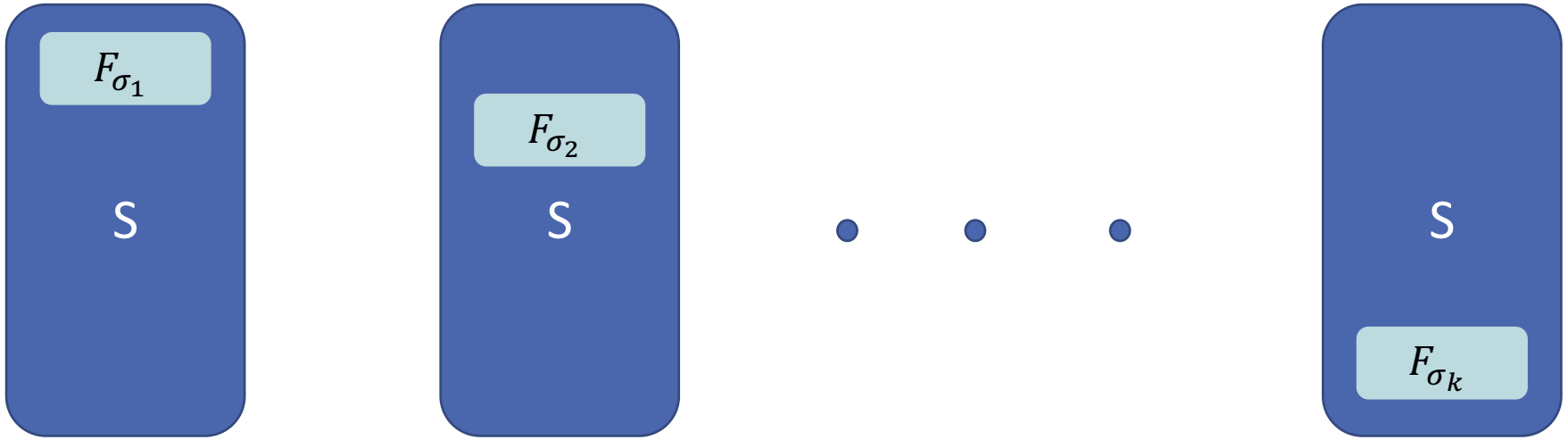
- Task $\tau = \sigma_0\sigma_1 \ldots$ is an infinite sequence of subtasks
- Upon completion of subtask σ_i system switches to next subtask σ_{i+1}
- Want to learn policies $\Pi = \{\pi_\sigma \mid \sigma \in \Sigma\}$ to maximize

$$J(\Pi) = \inf_{\tau \in \mathcal{T}} \mathbb{E}_{\rho \sim \mathcal{D}_\tau^\Pi} \left[\sum_{t=0}^{\infty} \gamma^t R_{\tau[i_t]}(s_t, \pi_{\tau[i_t]}(s_t)) \right]$$

Min-max objective: Worst case performance over tasks

A Game

Player 1's states = S_1 Player 2's states = S_2

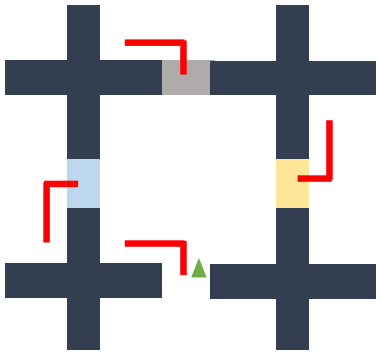


- One copy of S for each subtask
- Player 1 chooses actions from A in states – leads to normal system transitions
- Player 2 chooses subtask from Σ in states – leads to a jump transition into state space of chosen subtask

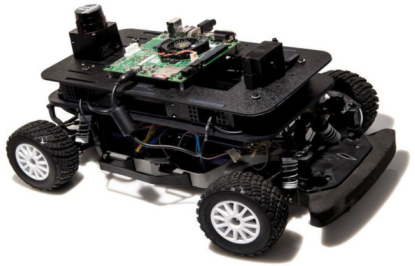
Solving the game

- Two **value iteration** algorithms to compute V* in the known model case
- A Q-learning algorithm that **converges in the limit** for finite states
- An SAC based **algorithm for infinite state/action spaces**
- An asynchronous algorithm for **learning options in parallel**

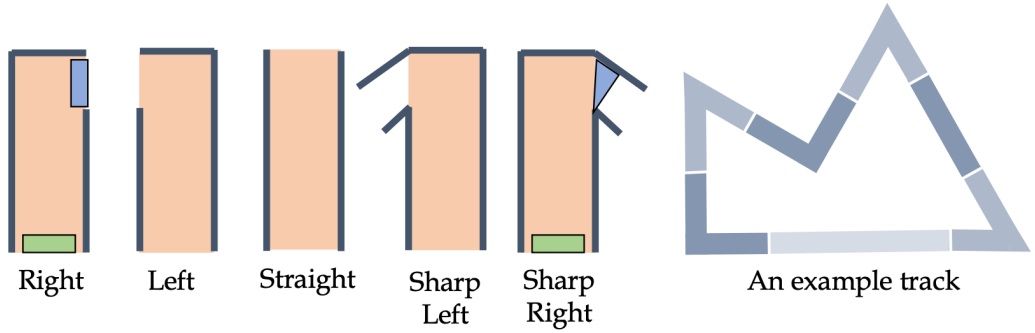
Experiments



Rooms environment



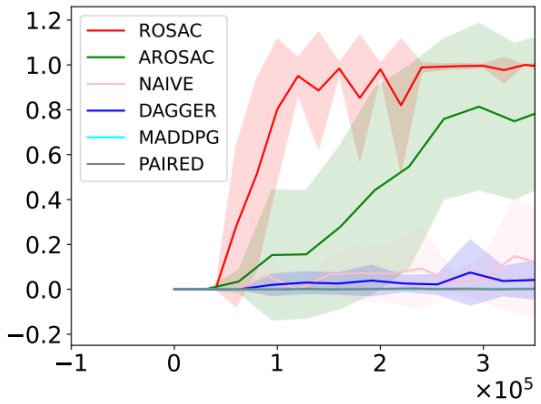
(a) F1/10th Car



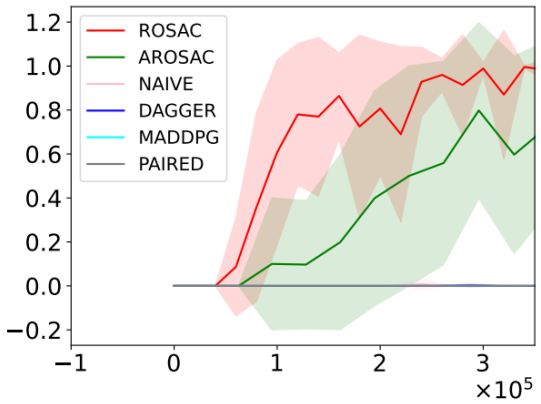
(b) Segment Types

F1/10th environment

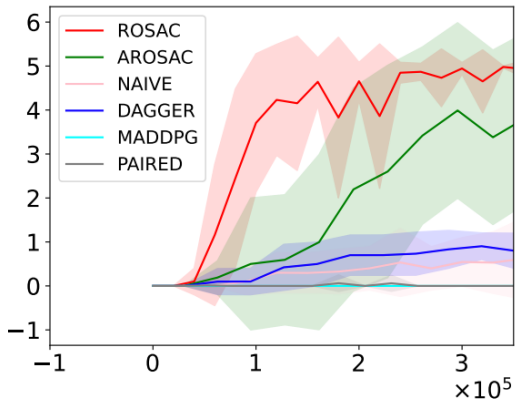
Experiments



(a) Success probability against random adversary



(b) Success probability against MCTS adversary



(c) Number of subtasks completed against MCTS adversary

Figure 3: Plots for the Rooms environment. x-axis denotes the number of sample steps and y-axis denotes the either the average number of subtasks completed or the probability of completing 5 subtasks. Results are averaged over 10 runs. Error bars indicate $\pm$ standard deviation.

Thank You!

Paper

My Website