Successor Feature for Transfer in Games

3rd ACC Workshop on Recent Advancement of Human Autonomy Interaction and Integration

Sunny Amatya
Robotics and Intelligent Systems Laboratory,
The Polytechnic School,
Arizona State University
May 30th, 2023

ASU ROBOTICS & INTELLIGENT SYSTEMS LAB
ARIZONA STATE UNIVERSITY

SAN ACC 2023 DIE

Introduction and Motivation

For safe interactions AV needs to able to efficiently infer the intent of other vehicles while being able to adapt to new and unseen scenarios

Autonomous Driving as Incomplete Information Game

- Interaction between Human and AV is **general-sum dynamic game** with **incomplete information**.
- Reward is a function of **states, action** and **preference parameters**
  - **Reward**

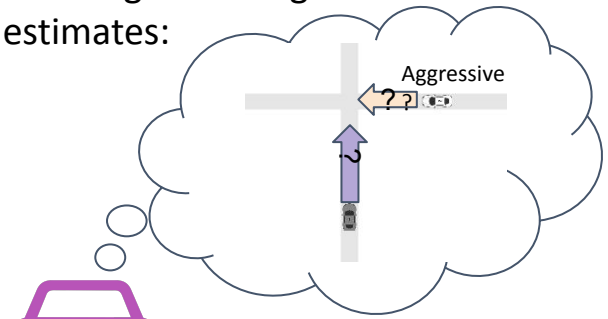
$$R_p(t) = c_{safety} + \theta_i c_{task}$$

- In our previous research, we train at least 4 networks for all agent types (A, NA)
- Training the set up in new task is **computationally expensive** and **time consuming**
- Current in literature (Use the same value function for new game); this may not work when tasks are significantly different.
- Example: A strategy trained on aggressive agent may not work for non aggressive agent.

Human

AV / **Joint Prob. (ΘxΘ)**	**Aggressive**	**Non-aggressive**
Aggressive	**0.1**	**0.5**
Non-aggressive	**0.3**	**0.1**

Planning according to the estimates:

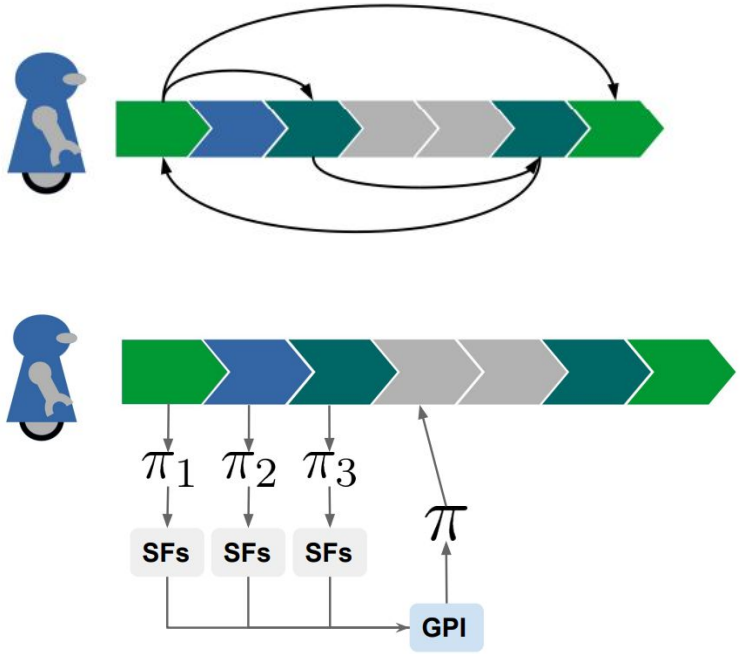


Autonomous Driving as General Sum Game

- Step back and look into into **complete information game**
- Looking into **Nash Q-learning (temporal difference)**
- Reward is a function of **states, action** and **preferences**
  - **Reward function**

$$r_i(s, a, s') = \boldsymbol{\phi}(s, a, s')^\top \mathbf{w}_i$$

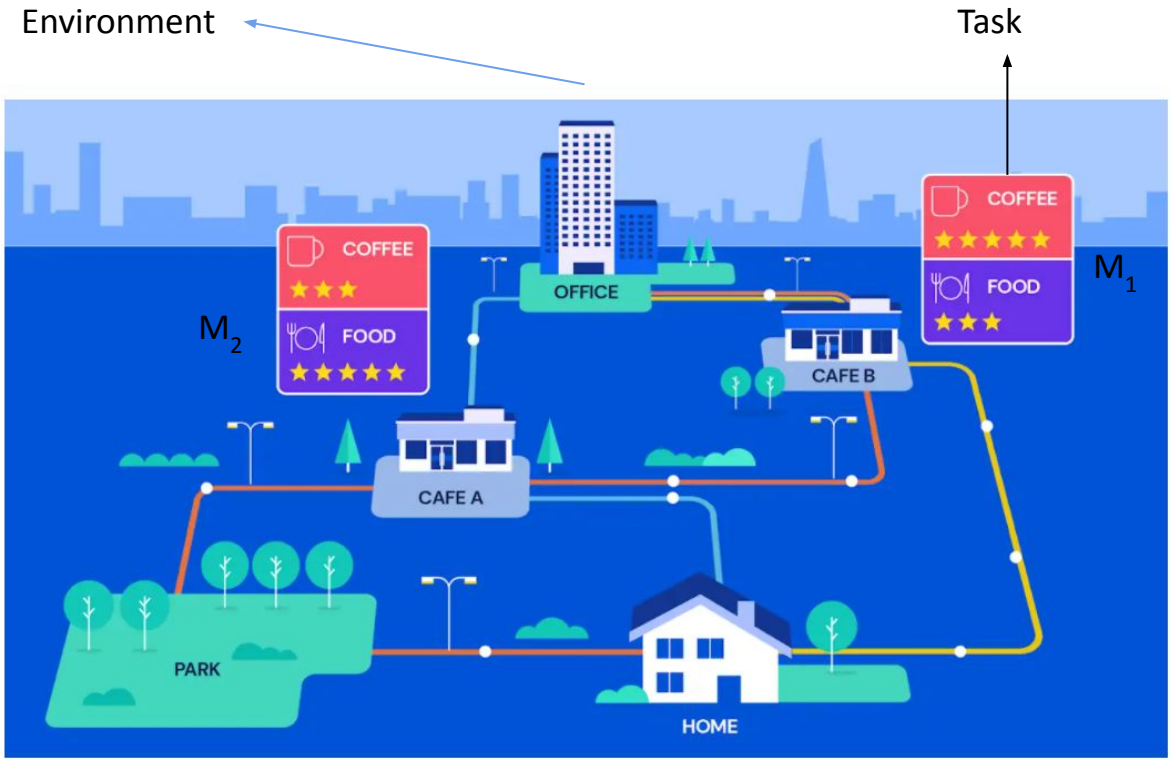
- Key Question
  - Can you introduce **successor feature in multi agent game**?
  - Does **successor feature** aid in better initialization during training?
  - In current AV applications, does the proposed method fare better than the state of the art algorithms?



[1]Successor Feature for Transfer in RL

1. Barreto, A., Dabney, W., Munos, R., Hunt, J.J., Schaul, T., van Hasselt, H.P. and Silver, D., 2017. Successor features for transfer in reinforcement learning. Advances in neural information processing systems, 30.

Transfer in RL? Problem Definition

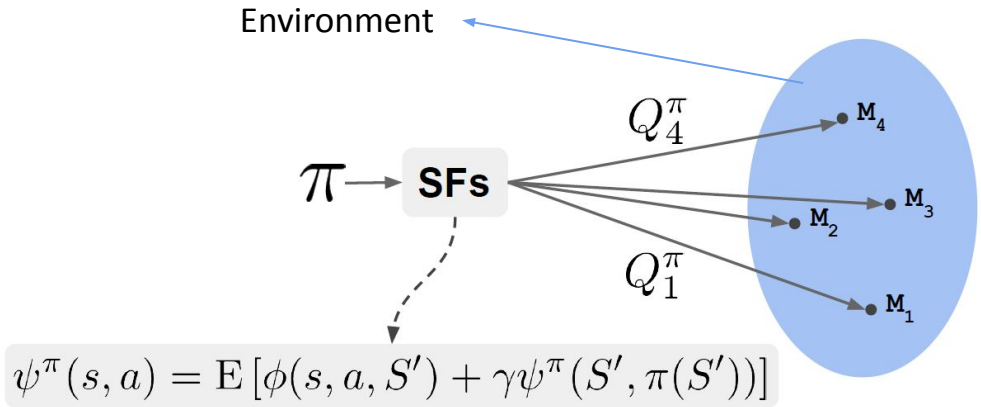


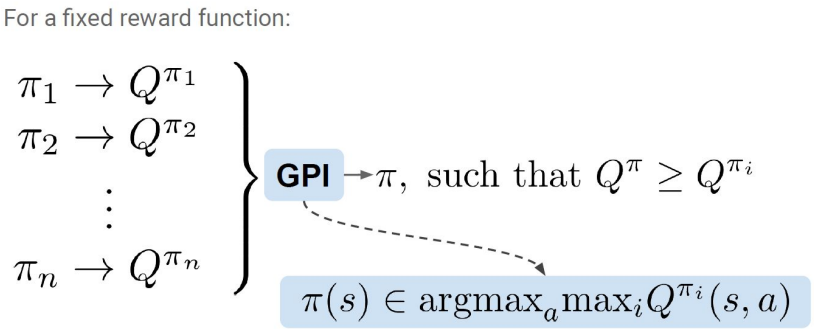
- Environment is a set of MDPs
- Each MDP M_i is a task
- The only difference between the MDPs are reward function r_i:

$$r_i(s, a, s') = \boldsymbol{\phi}(s, a, s')^\top \mathbf{w}_i$$

- Path chosen differs on the preference eg (IRL)
- (features: coffee, food, distance, leisure)

What is successor feature?





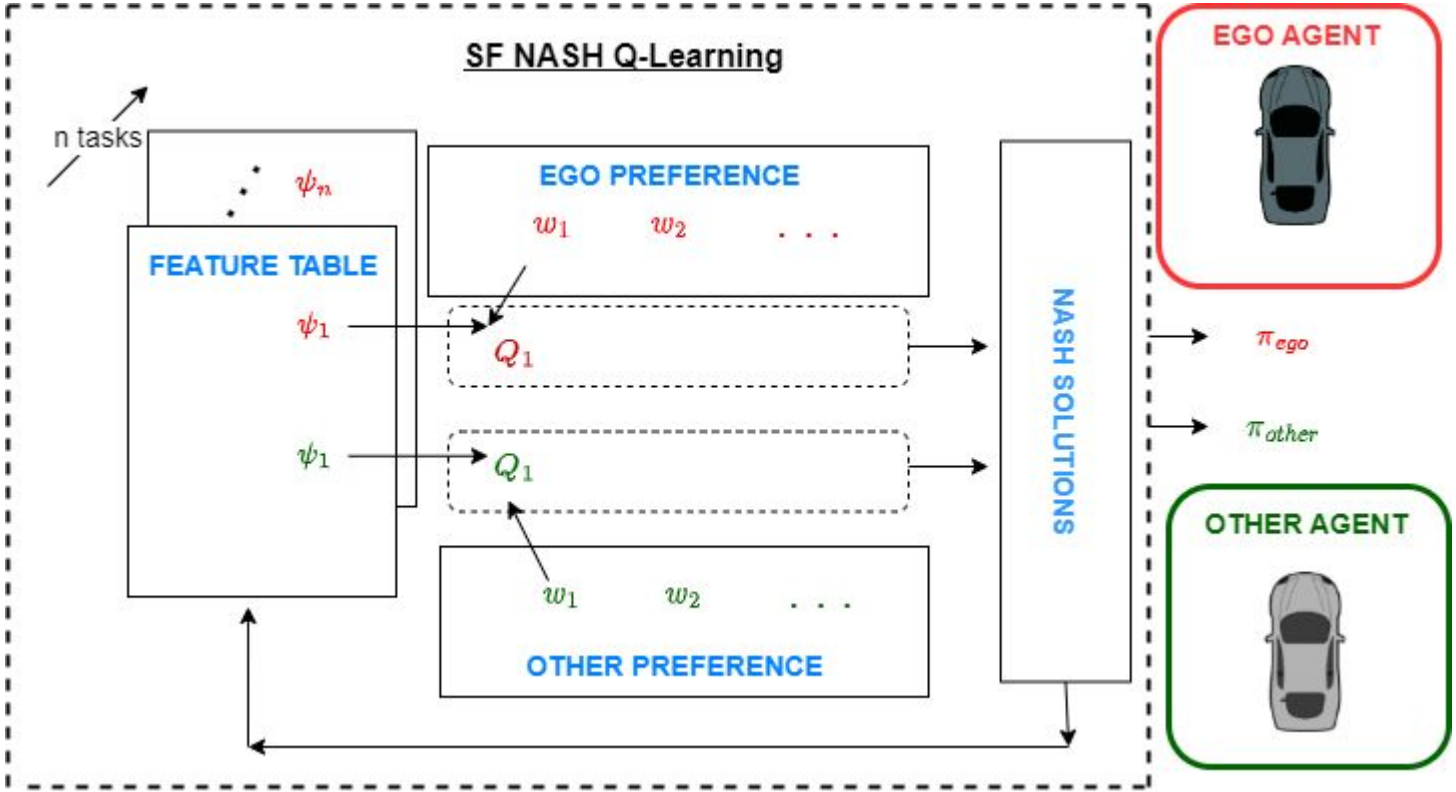
$$Q^\pi(s,a) = \psi^\pi(s,a)^\top \mathbf{w}$$

[1]Successor Feature for Transfer in RL

- Exchange of information takes place whenever useful (Generalized Policy Improvement)
  - Already existing knowledge should be transferable
- Transfer is seamlessly integrated with the RL process. (Successor Feature)
  - This requires computing Q as function of weight (preference) and features

1. Barreto, A., Dabney, W., Munos, R., Hunt, J.J., Schaul, T., van Hasselt, H.P. and Silver, D., 2017. Successor features for transfer in reinforcement learning. Advances in neural information processing systems, 30.

Successor Feature in Game



Mathematical Formulation

- Changes: two successor table for each agent
- Selection mechanism choice (f)
- Among all mechanism choose reactive policy that maximizes ego's reward
- Update SF table of each agent according to the actions generated from the new policy

Algorithm 3: Multi Agent Successor Feature Nash Q-learning

input : discount factor γ, selection mechanism f, task index k, learning rate α, action set b

output : Successor features $\psi_i^{\pi_k}$

for $t = 1$ *to* T **do**
  for *all* $i \in N$ **do**
    if *Bernoulli* $(\epsilon) = 1$ **then**
      $a_i \leftarrow$ **Uniform**(A) //exploration
    else
      $a_i \leftarrow \arg\max_b \max_k \psi^{\pi_k}(s, a_1..a_n)w_i$ //GPI
    end
  end
  simulate action a_1, a_n in state s
  observe rewards $r_1, ..r_n$ and the next state s'
  $w_i = [r_i - \phi_i(s, a, s')^T \tilde{w_i}]\phi_i(s, a, s')$ //Learn weight
  for *all* $i \in N$ **do**
    $Q_i(s', a_1, .., a_n) = \max_k \psi_i^{\pi_k}(s, a_1..a_n)w_i$ //extract $Q_i(s')$
    compute $V_i(s') \in$ **NASH**$_i(Q_1^*(s', a_1, ..a_n), Q_n^*(s', a_1, ..a_n)$
    $\pi(s', a_1, ..a_n) \in f(Q_1(s', a_1, ..a_n)..Q_n(s', a_1, ..a_n))$
    $V(s') = \pi(s', a_1, a_n) * Q(s, a_1, a_n)$
    Compute new action a_1', a_n'
    $\psi_i^{\pi_k}(s, a_1..a_n) = (1 - \alpha)\psi_i^{\pi_k}(s, a_1..a_n) + \alpha[\phi_i + \gamma\psi_i^{\pi_k}(s', a_1'..a_n')]$ // update successor feature
  end
  Update states
end

Experiment Setup

Features

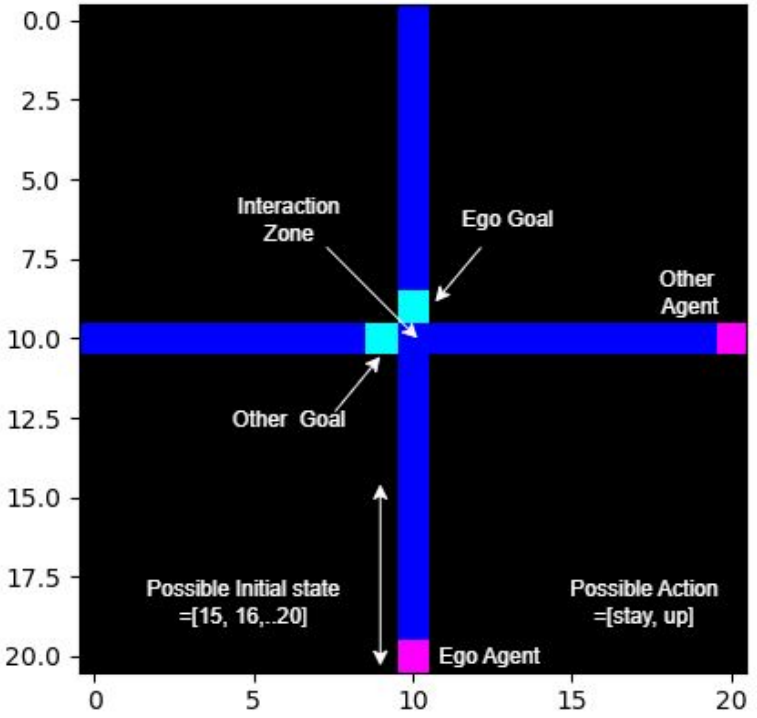
- Goal driven $R_p(t) = c_{safety} + \theta_i c_{task}$
- Property induced by intent parameter
- Rewards (safety) : one step distance from other agent
- Rewards (task) : distance from goal

$$R_p(t) = \phi w$$

$$R_p(t) = [c_{safety}\ c_{task}] \bullet \begin{bmatrix} 1 \\ \theta_i \end{bmatrix}$$

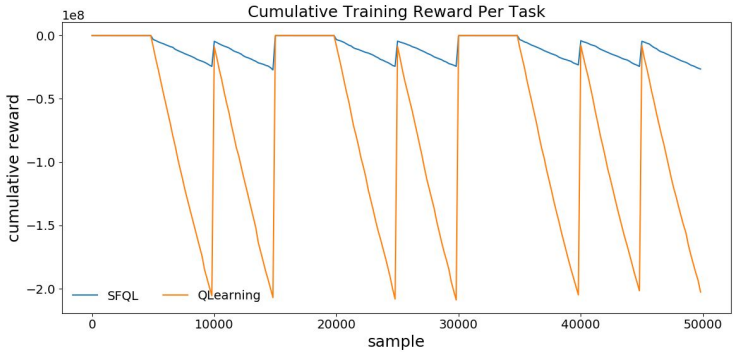
Simplification

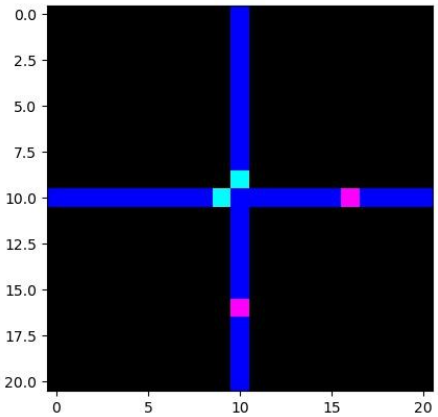
- Single agent setup
- Discrete state [10 - 20]
- Discrete action [0, 1]
- Change in the intent of ego agent
- Other agent is moving forward



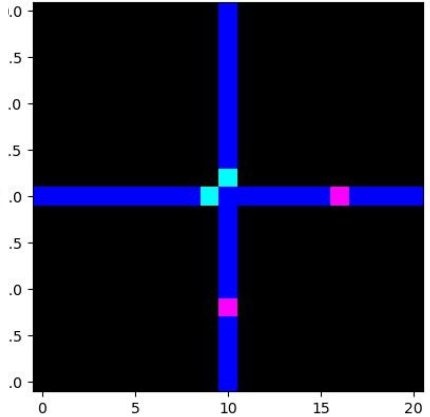
Preliminary Result: Transfer in Aggressiveness

- Single ego agent test
- Testing change of aggressiveness from the generated tabular form
- 50000 iteration with change of aggressiveness every 5000 iterations
- Reduced loss in the system specially for aggressive agents.

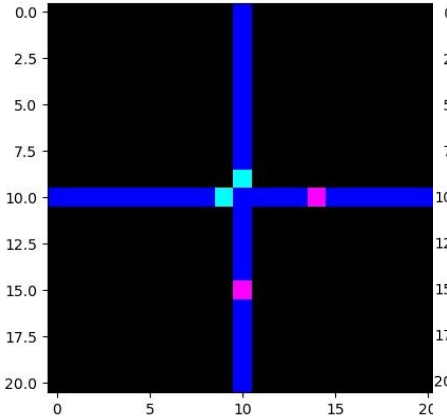




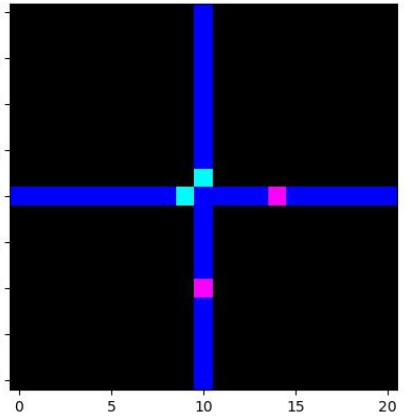
Initial position for A



A stops closer to intersection



Initial position for NA



NA stops further from intersection

Baseline Result


```
[[16, 10], [10, 16]]
[[16, 10], [10, 15]]
[[16, 10], [10, 14]]
[[16, 10], [10, 13]]
[[16, 10], [10, 12]]
[[16, 10], [10, 11]]
[[16, 10], [10, 10]]
[[16, 10], [10, 9]]
NashQLearning 0
Player 0 follows the  policy : stay-stay-stay-stay-stay-stay-stay of length 7
Player 1 follows the  policy : up-up-up-up-up-up-up of length 7
game over
100%|██████████| 100/100 [00:00<?, ?it/s]
100%|██████████| 15000/15000 [00:27<00:00, 545.17it/s]
[[15, 10], [10, 16]]
[[15, 10], [10, 15]]
[[15, 10], [10, 15]]
NashQLearning 1
Player 0 follows the  policy : m-o-d-e-l- -f-a-i-l-e-d- -t-o- -c-o-n-v-e-r-g-e-
Player 1 follows the  policy : m-o-d-e-l- -f-a-i-l-e-d- -t-o- -c-o-n-v-e-r-g-e-
```

Non Aggressive AV
Aggressive H

Non Aggressive AV
Non Aggressive H

```
[[16, 10], [10, 16]]
[[16, 10], [10, 15]]
[[16, 10], [10, 14]]
[[16, 10], [10, 13]]
[[16, 10], [10, 12]]
[[16, 10], [10, 11]]
[[16, 10], [10, 10]]
[[16, 10], [10, 9]]
nashSFQl 0
Player 0 follows the  policy : stay-stay-stay-stay-stay-stay-stay of length 7
Player 1 follows the  policy : up-up-up-up-up-up-up of length 7
game over
100%|██████████| 15000/15000 [00:39<00:00, 380.56it/s]
[[15, 10], [10, 16]]
[[14, 10], [10, 16]]
[[13, 10], [10, 16]]
[[12, 10], [10, 15]]
[[11, 10], [10, 15]]
[[10, 10], [10, 14]]
[[9, 10], [10, 13]]
nashSFQl 1
Player 0 follows the  policy : up-up-up-up-up-up of length 6
Player 1 follows the  policy : stay-stay-up-stay-up-up of length 6
```

- Non aggressive ego agent is able to identify other non aggressive agent and provide required policy with proposed algorithm where baseline algorithm fails

Thank you

ASU ROBOTICS & INTELLIGENT SYSTEMS LAB
ARIZONA STATE UNIVERSITY

Sunny Amatya (Samatya@asu.edu)
Wenlong Zhang (Wenlong.Zhang@asu.edu)
https://home.riselab.info/
https://home.riselab.info/nri.html

This material is based on the work supported in part by the National Science Foundation (NSF) under NSF Award Number **925403**. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and, do not necessarily reflect those of the NSF.