Introduction to inverse reinforcement learning (1/3)

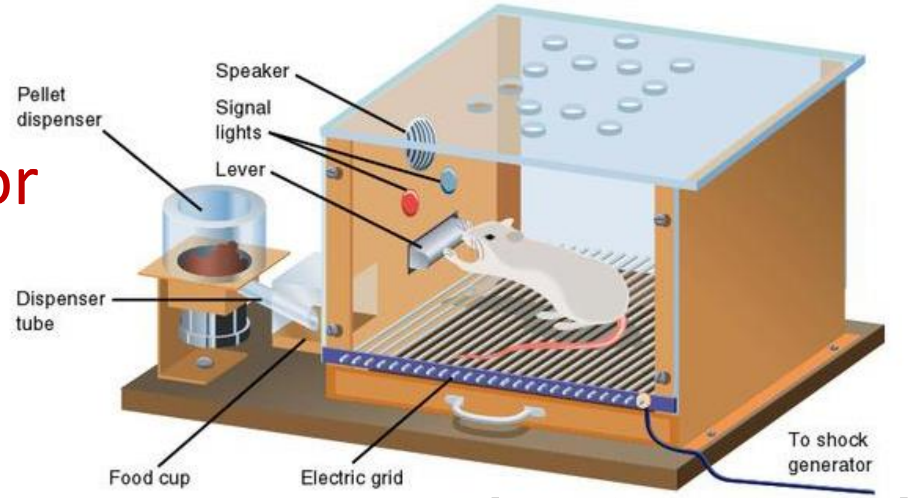
Eiji Uchibe

Dept. of Brain Robot Interface

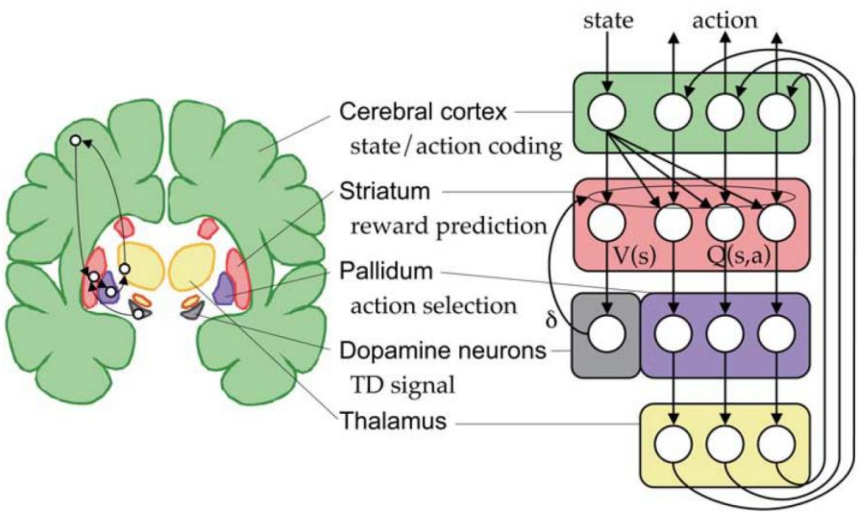
ATR Computational Neuroscience Labs.

What is Reinforcement Learning (RL)?

- RL is a computational framework for finding an optimal policy (controller) by trial and error
- Inspired by psychology
  - Thorndike's law of effect
  - Skinner's principle of reinforcement
- Computational model of decision making of human/animal
- Learning algorithm of artificial agents



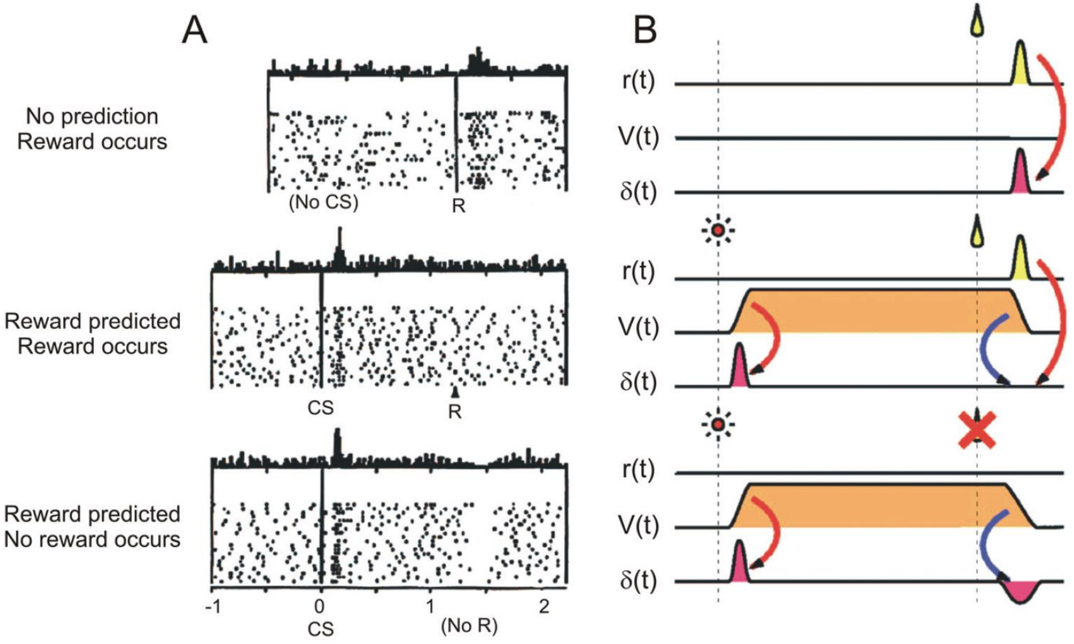
[Simply Psychology]



K. Doya (2007). Reinforcement learning: Computational theory and biological mechanisms. HFSP Journal, vol. 1, no. 1, pp. 30–40.

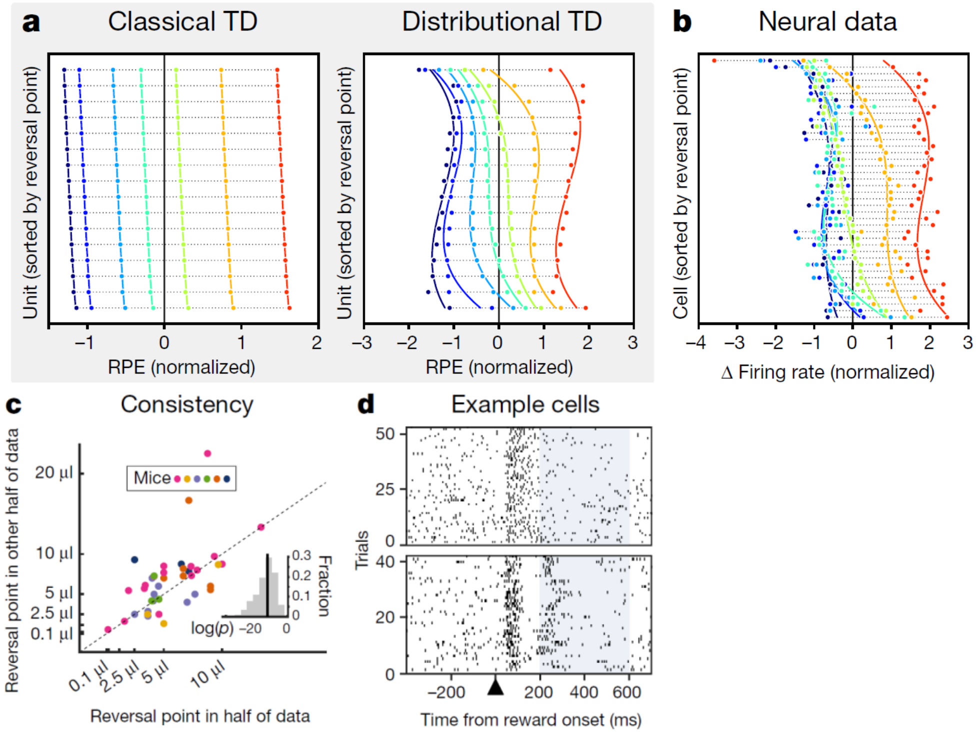
Reinforcement learning in neuroscience

Dopamine neurons code Temporal Difference error



Schultz, W.P., Dayan, P., and Montague, P.R. (1997). [A Neural Substrate of Prediction and Reward](). *Science* 275, no. 5306: 1593–99.

Distributional RL in our brain
a single reward outcome can simultaneously elicit positive RPEs (within relatively pessimistic channels) and negative RPEs (within more optimistic ones)



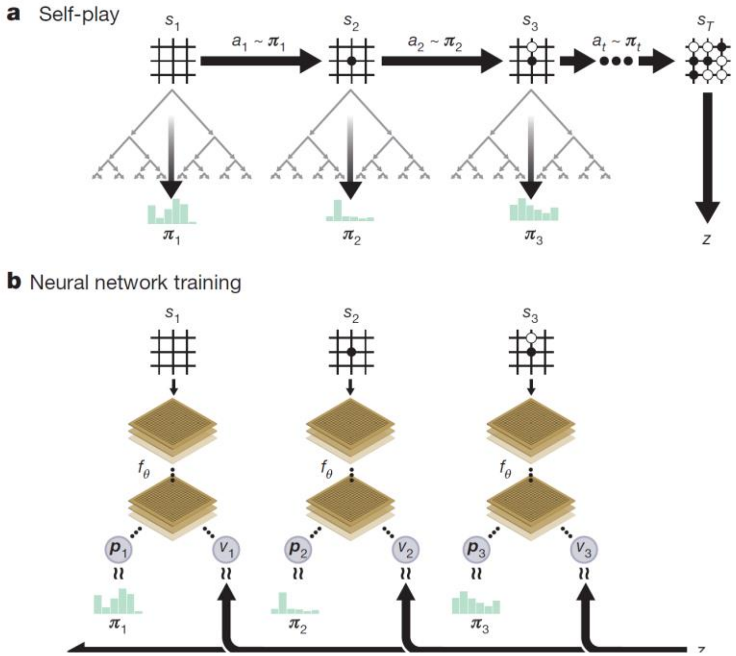
Dabney, W., Kurth-Nelson, Z., Uchida, N. et al. (2020). [A distributional code for value in dopamine-based reinforcement learning](). *Nature*, 577, 671–675.

Reinforcement learning in games

AlphaGo Zero
board game, Go
RL from scratch
4.9 millions of self-play

AlphaStar
multiagent real-time strategy game
RL + supervised learning
200 years

Gran Turismo Sophy
realistic racing game
RL with shaped rewards
1,000 PlayStation 4



Silver, D., Schrittwieser, J., Simonyan, K. et al. (2017). [Mastering the game of Go without human knowledge](). *Nature*, 550, 354–359.

Vinyals, O., Babuschkin, I., Czarnecki, W.M. et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, vol. 575, 350-354.

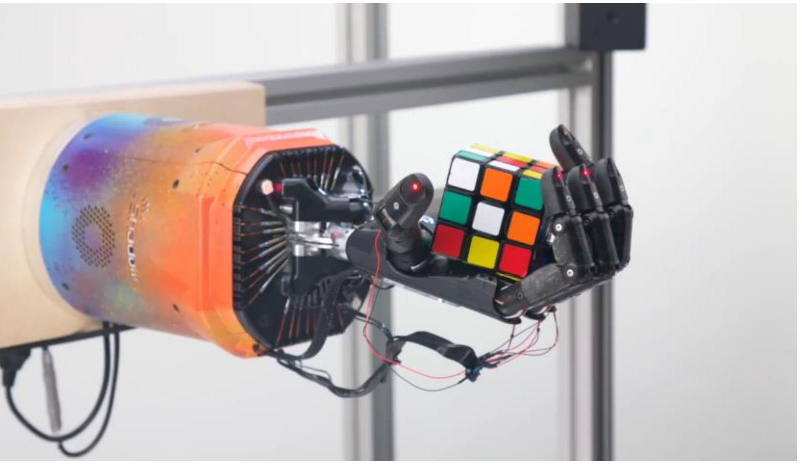
Wurman, P.R., Barrett, S., Kawamoto, K. et al. (2022). Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602, 223–228.

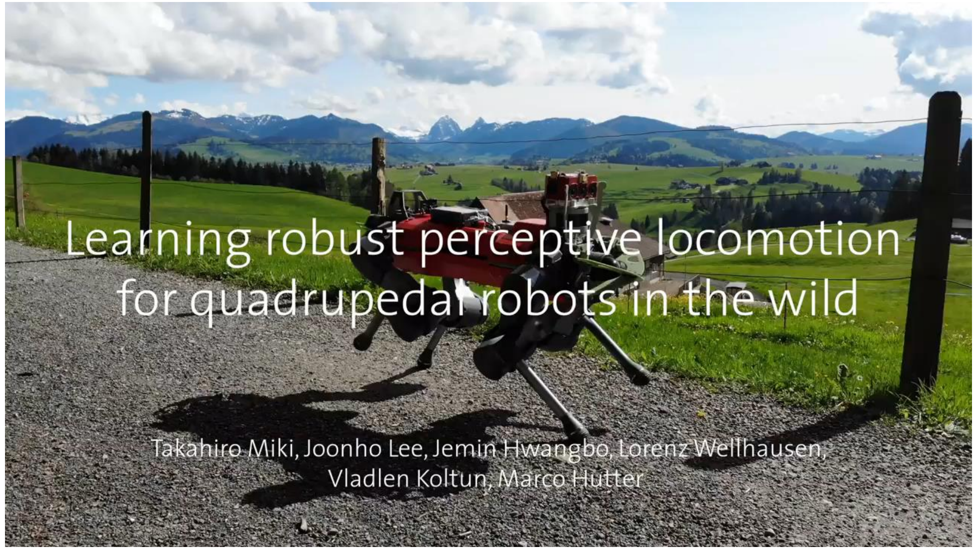
Reinforcement learning in robotics

Manipulating Rubik's cube
RL + domain randomization
2.8 GWh of electricity

MT-Opt
Grasping objects using visual information
RL (+ supervised learning)
7 robots, 9600 robot hours

ANYmal (quadruped robot)
complete an hour-long hiking loop faster than human
Teacher: RL with privileged info.
Student: imitate the teacher





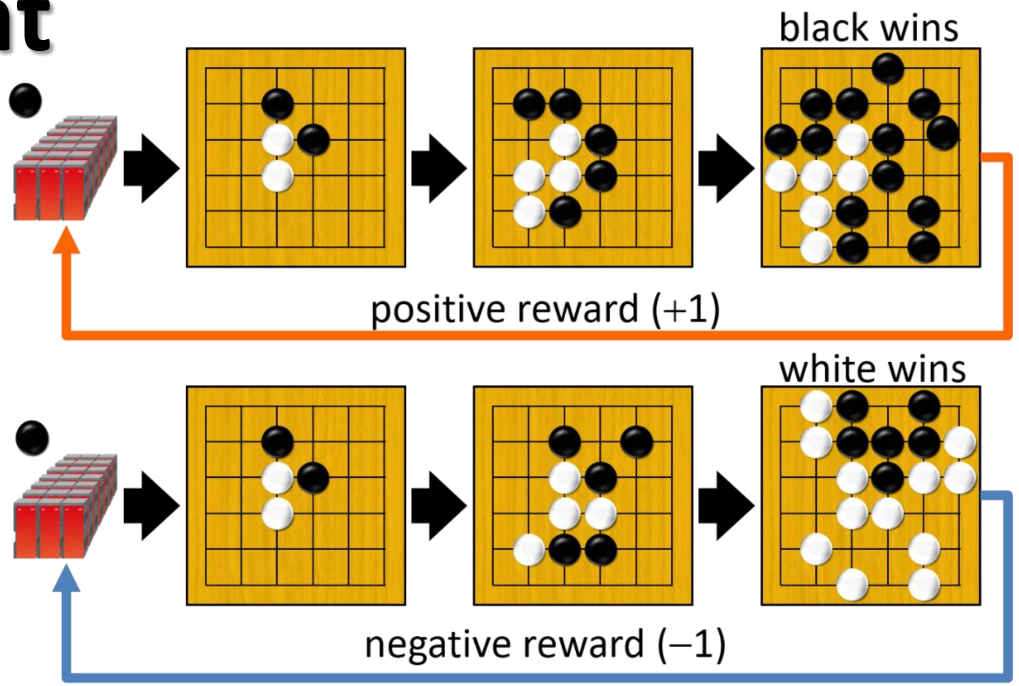
Akkaya, I., Andrychowicz, M., Chociej, M. et al. (2019). [Solving Rubik's Cube with a Robot Hand]. arXiv. [[OpenAI Blog]]

Kalashnikov, D., Varley, J., Chebotar, ,Y. et al. (2021). [Scaling Up Multi-Task Robotic Reinforcement Learning]. In Proc. of the 5th Conference on Robot Learning.

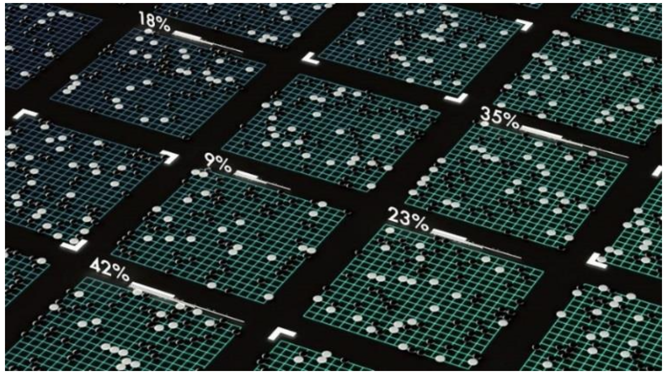
Miki, T., Lee, J., Hwangbo, J. et al. (2022). [Learning robust perceptive locomotion for quadrupedal robots in the wild]. Science Robotics, vol. 7, issue 62.

Designing Reward is important

- In the case of Go
  - positive reward for winning
  - negative reward for losing
  - zero otherwise
- AlphaGo Zero, which does not use a record of a game of go, needs **4.9 million** games of self-play

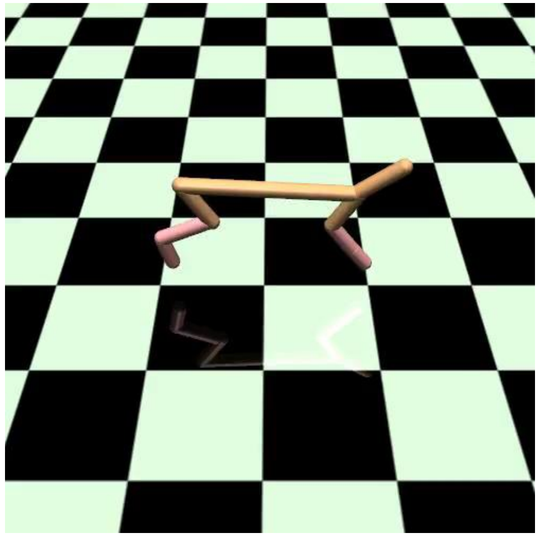
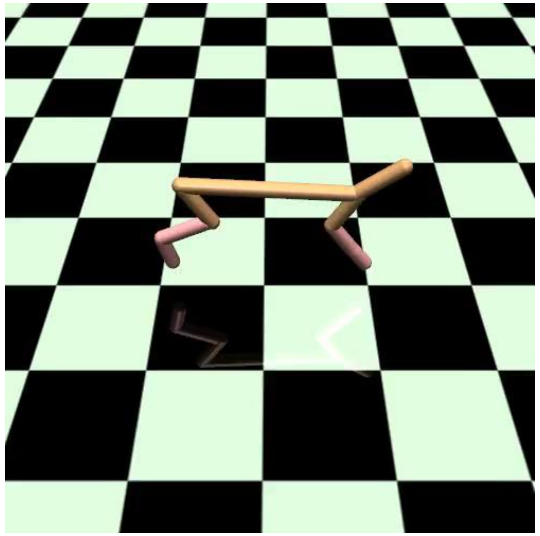


- Deep RL is applicable when we can collect samples by using multiple simulators
- What happens if we use a dense reward?



But designing reward is difficult …

- Task: move forward as fast as possible (continuous state-action problem)

immediate reward

$$r(\boldsymbol{s},\boldsymbol{a}) = \underbrace{v_x}_{\substack{\text{forward}\\\text{velocity}}} - 0.05\underbrace{\|\boldsymbol{a}\|_2^2}_{\substack{\text{squared norm}\\\text{of applied torque}}}$$

- Even if the reward function is well-shaped, it is not enough to find an optimal policy when learning time is limited
- Inverse RL provides the method to design the reward from behaviors of experts

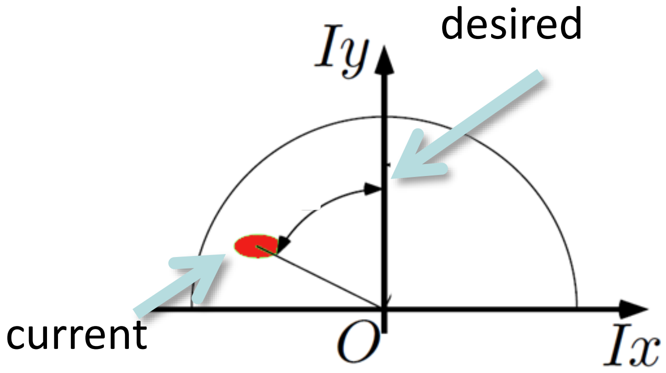
[Deep Reinforcement Learning Doesn't Work Yet]

Designing Reward is Difficult

- Task: catch a battery pack
- Two reward functions: r_{orig} and r_{aug}

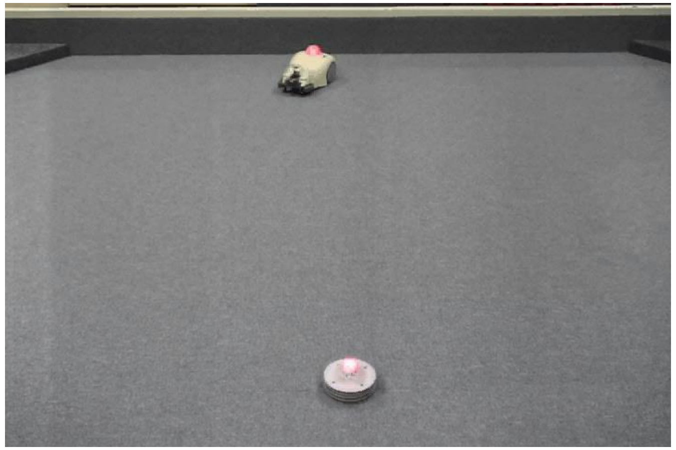
$$r_{\text{orig}} = \begin{cases} +1 & \text{if catching a battery pack} \\ -0.05 & \text{if moving} \\ 0 & \text{otherwise} \end{cases}$$

$$r_{\text{aux}} = \exp\left(-\frac{(\theta - \theta_d)^2}{2\sigma^2}\right)$$

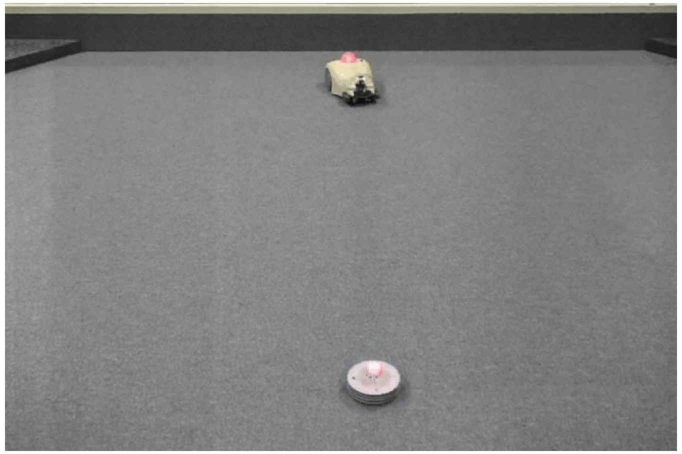


- Watching a battery pack was obtained according to the choice of w although it learned faster

Trained with r_{orig}

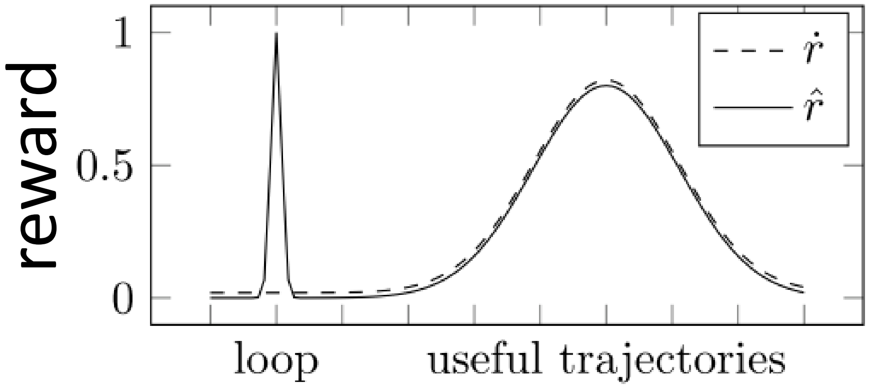


Trained with $r_{\text{orig}} + wr_{\text{aux}}$



Designing Reward is Difficult

- Task: finish the boat race quickly
  - not directly reward the player's progression around the course
  - get rewards by hitting targets laid out along the route



[Everitt et al., 2017]

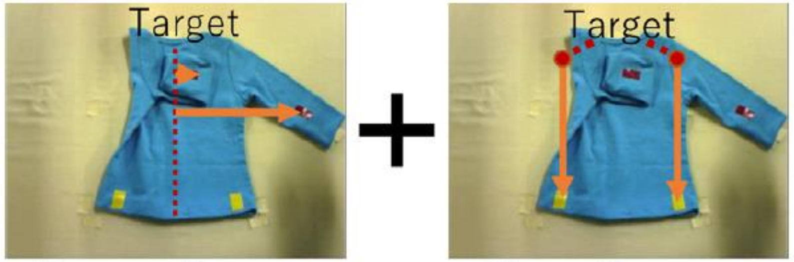
Everitt, T. (2018). [Towards Safe Artificial General Intelligence](). Ph.D. Thesis. Australian National University.

https://www.youtube.com/watch?v=tlOIHko8ySg

  - $\dot{r}$: true reward
  - $\hat{r}$: corrupt, observed reward

Reward function for folding a T-shirt

Reward — The reward function is designed to trigger an action to fold the hem after folding the sleeve. The processing is shown in Algorithm 3.



Sleeve distance from center

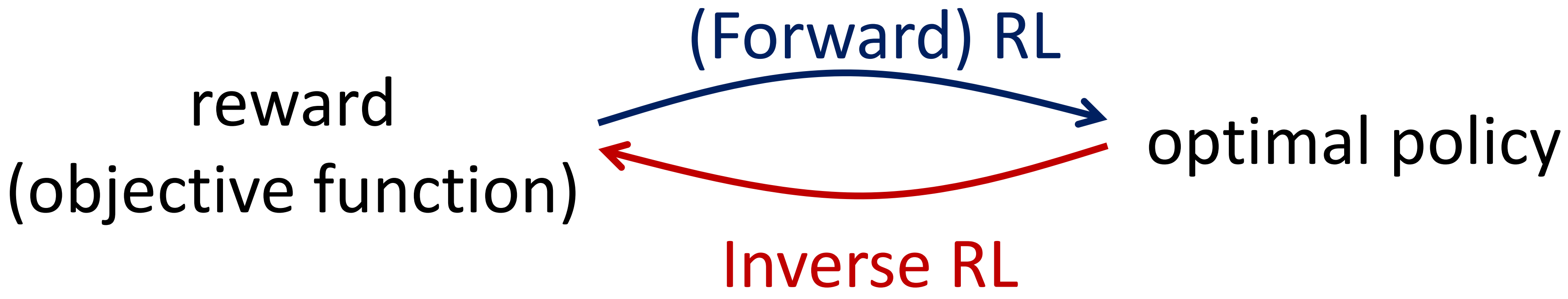
Hem distance from target point

Algorithm 3: Reward function of t-shirt folding task

```
Initialize InitHemR = [0.675, 0.8], InitHemL = [0.325, 0.8]
Initialize TargetHemR = [0.675, 0.208],
TargetHemL = [0.325, 0.208]
Function HemReward(SleevePoint, CenterHem):
    Initialize reward = 0
    reward = -Sum(|SleevePoint - CenterHem|)
    return reward
Function SleeveReward(HemPoint, InitHem, TargetHem):
    Initialize reward = 0
    Initalize Distance = |InitHem − TargetHem|
    reward = Sum(Distance − |HemPoint − TargetHem|)
    return reward
Function ShirtReward():
    Initialize reward = 0
    Update color marker
    Get HemPointR, HemPointL, SleevePointR, SleevePointL
    if Detect hem marker then
        CenterHem = (HemPointR + HemPointL)/2
        reward = SleeveReward(SleevePointR, CenterHem) +
        SleeveReward(SleevePointL, CenterHem)
    else
        reward = 1
    if Detect sleeve marker then
        reward = reward +
        HemReward(HemPointR, InitHemR, TargetHemR) +
        HemReward(HemPointL, InitHemL, TargetHemL)
    return reward
```

Inverse Reinforcement Learning (IRL)

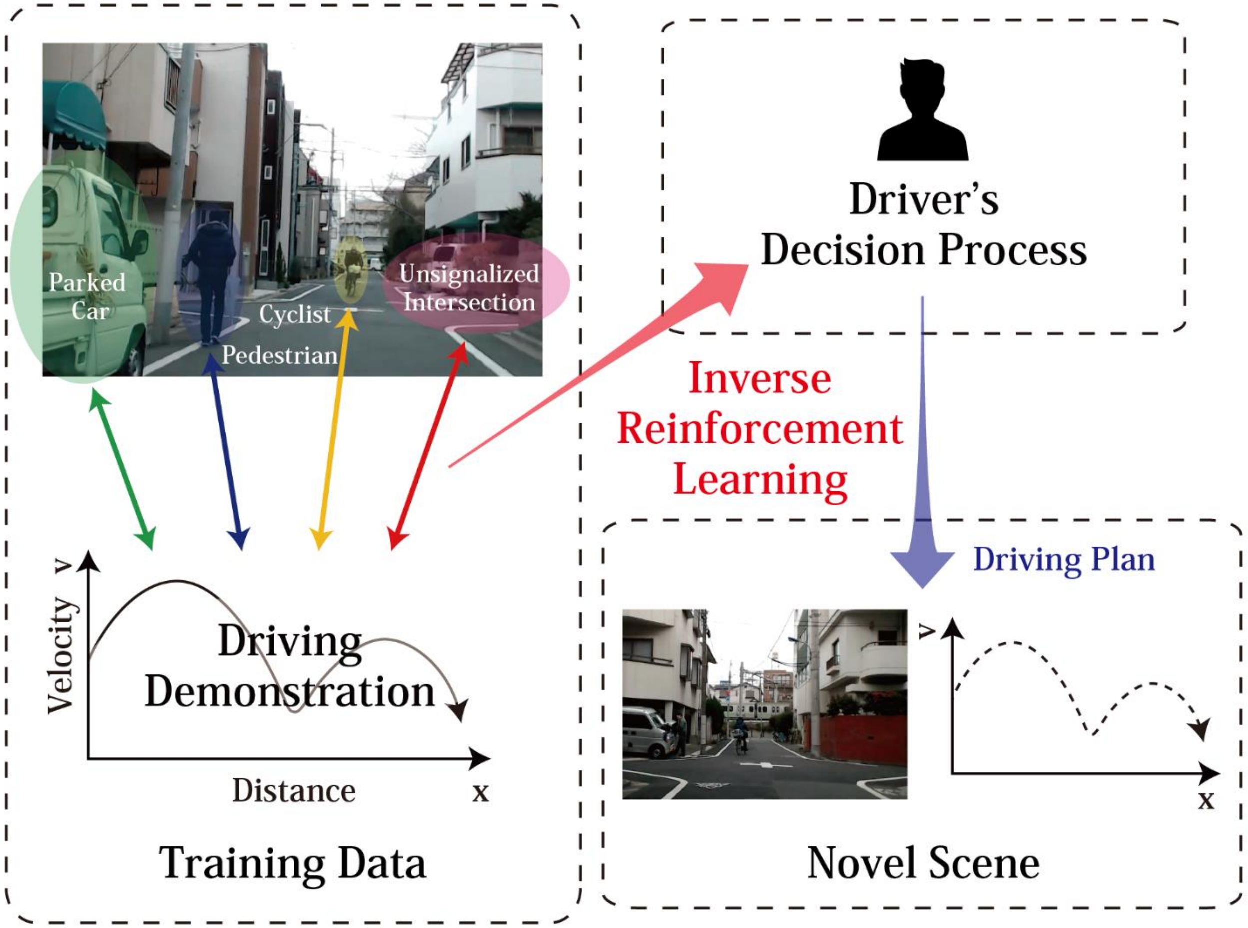
- Estimate a reward function from observed behaviors generated by an optimal policy



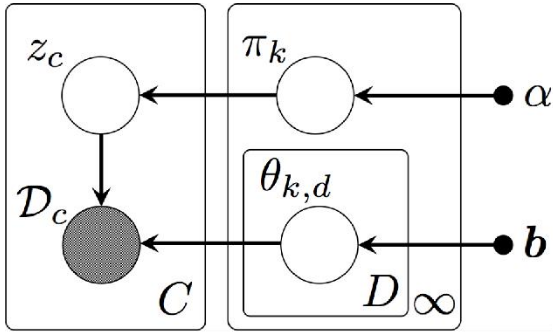
- It is often easy to demonstrate some good behaviors
- Ill-posed problem. That is, the solution is not uniquely determined

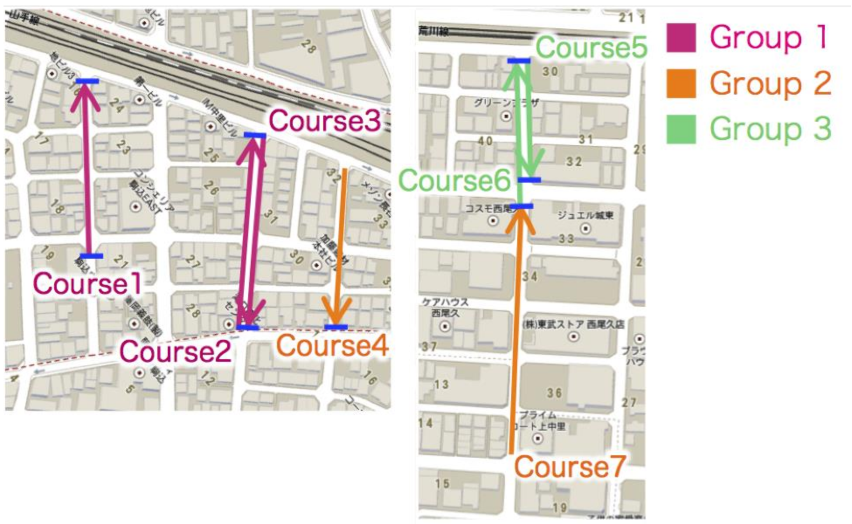
Modeling risk anticipation behaviors

- Estimate a speed control behavior



- Classification

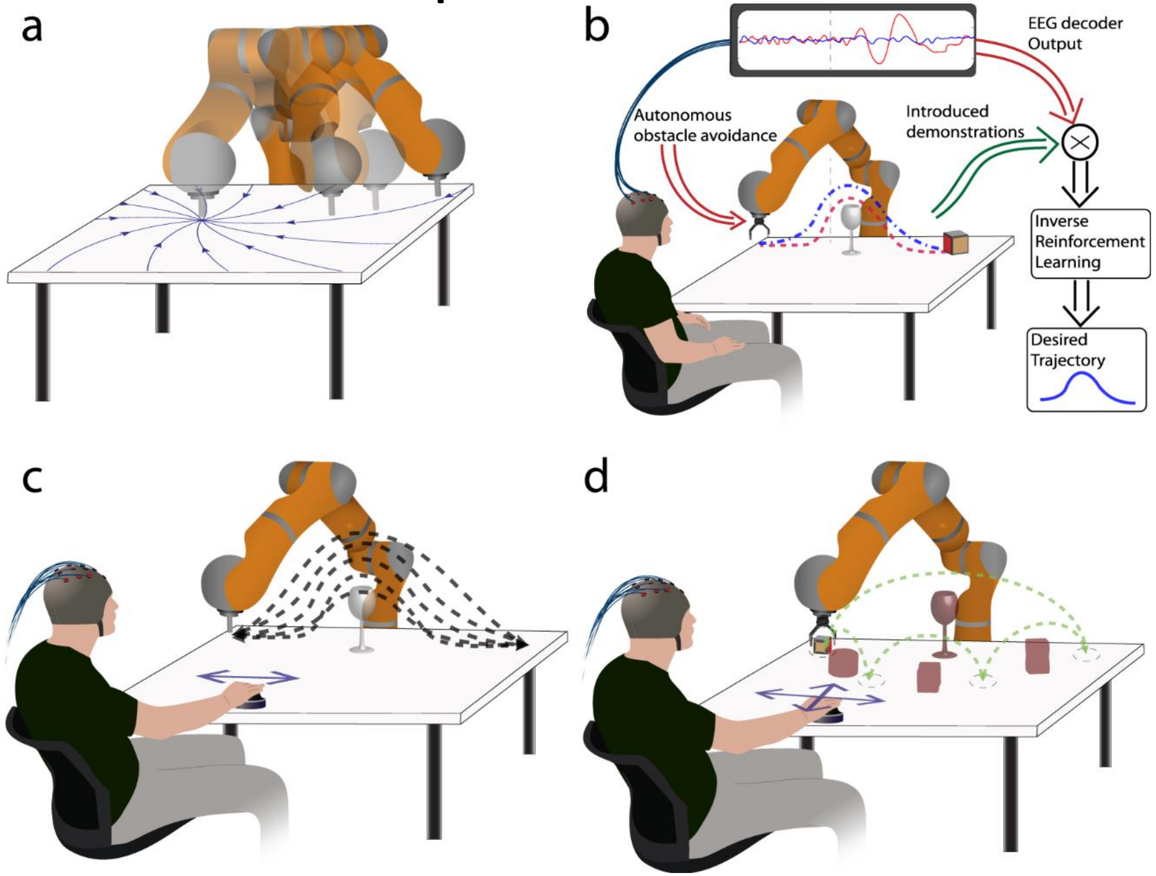


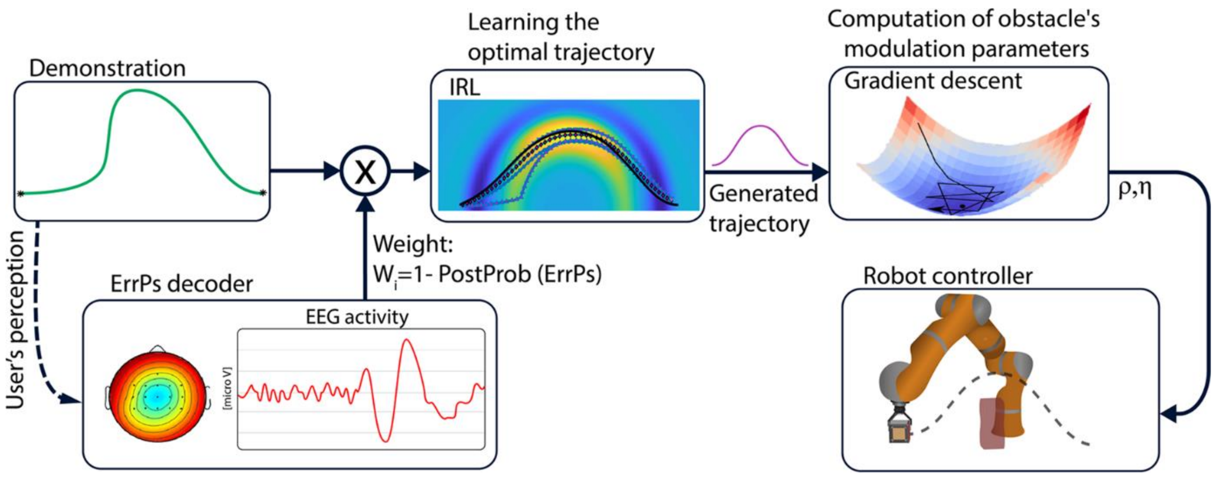


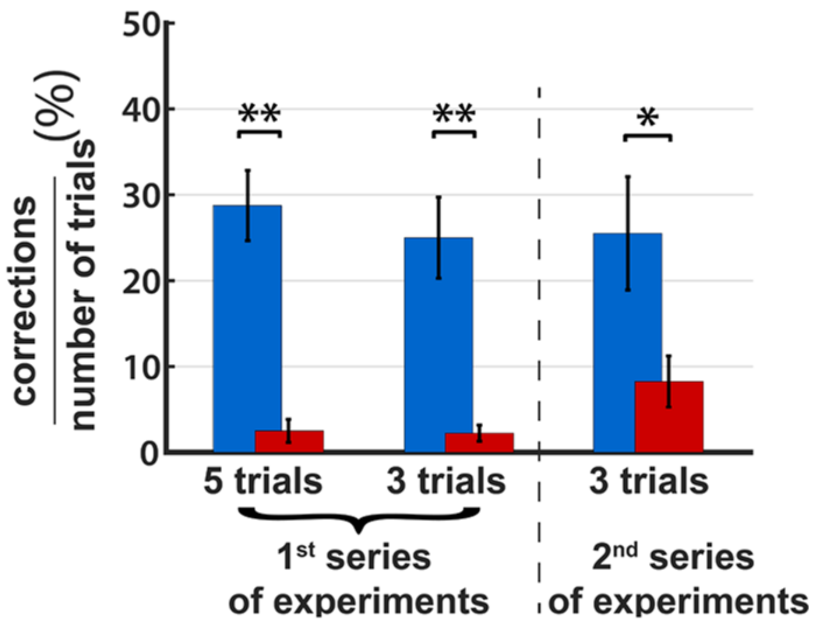
Shimosaka, M., Kaneko, T., & Nishi, K. (2014). Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning. *Proc. of the 17th International IEEE Conference on Intelligent Transportation Systems*, 1694–1700.

Application to Brain-Computer Interface

- Gaussian process-based IRL infers the user’s preference



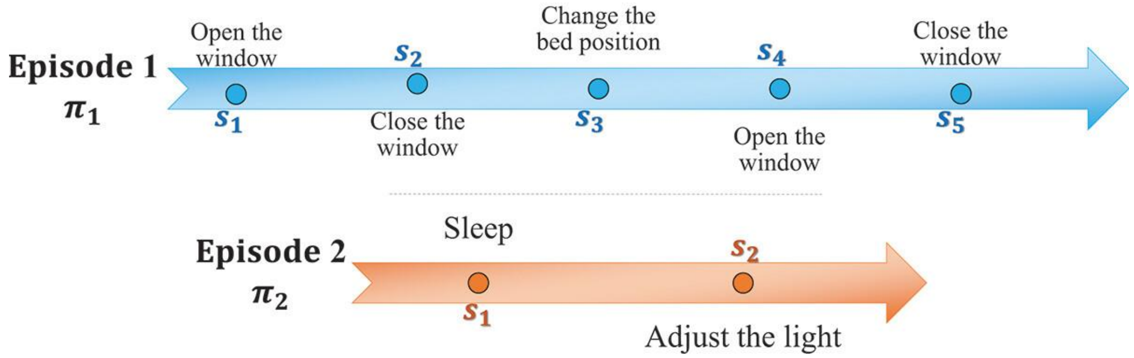


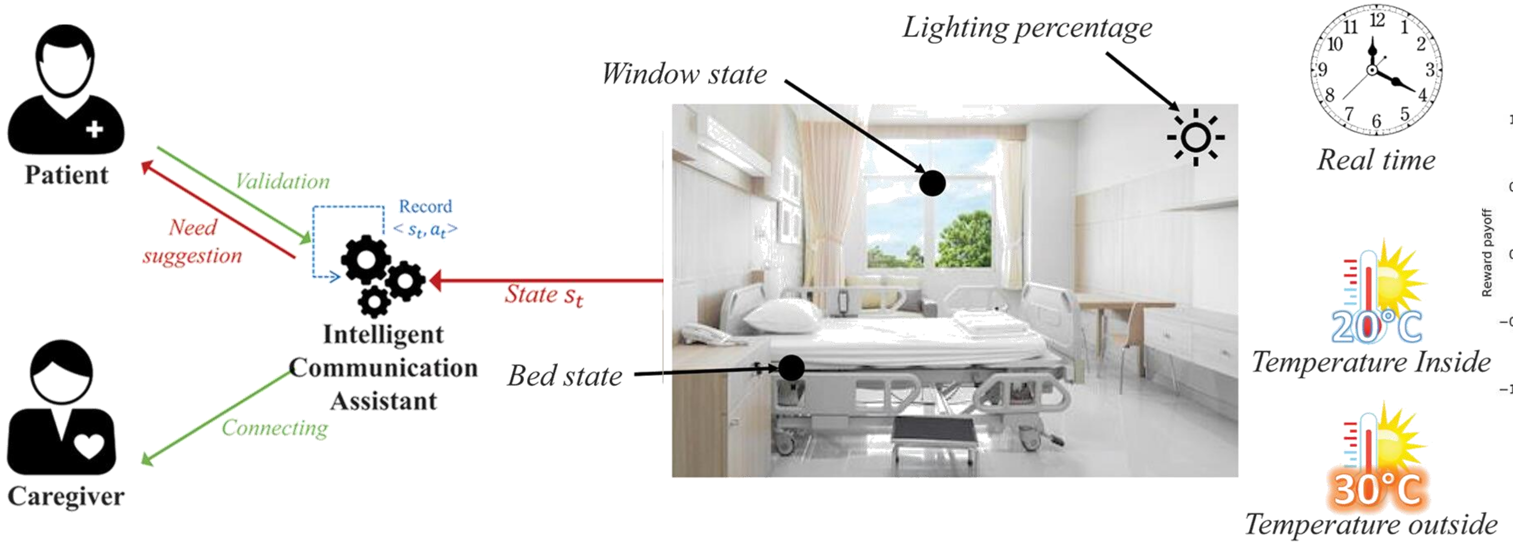


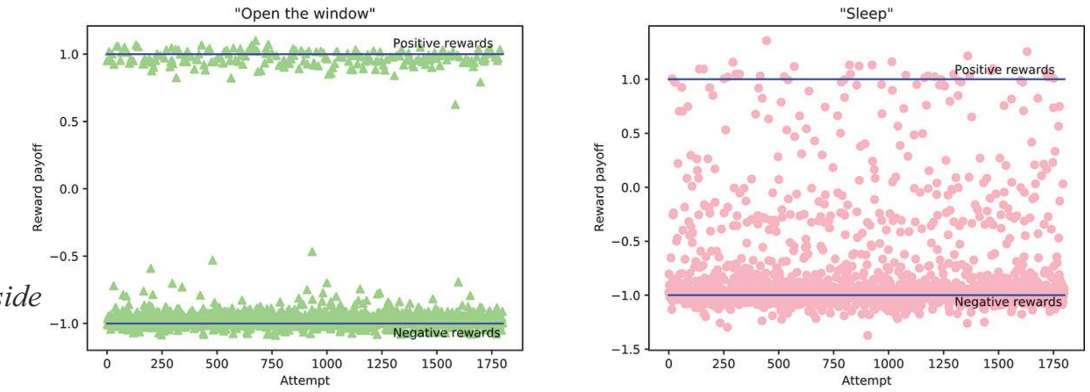
Batzianoulis, I., Iwane, F., Wei, S., et al. (2021). Customizing skills for assistive robotic manipulators, an inverse reinforcement learning approach with error-related potentials. Communications Biology 4, 1.

Smart health-care assistants

- Detecting physiological needs to improve the comfort of the patient
- Maximum entropy-based IRL







(a) Open the window

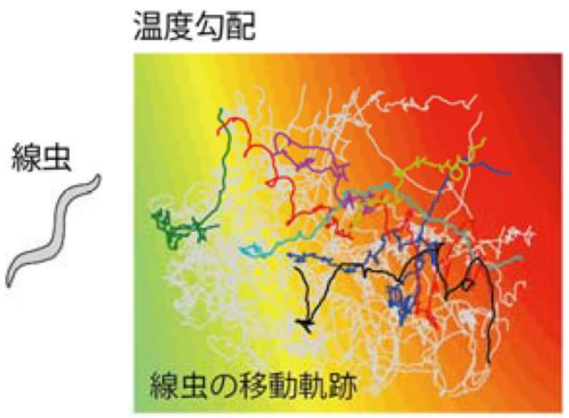
(b) Sleep

Hantous, K., Rejeb, L., and Hellali, R. (2022). [Detecting Physiological Needs Using Deep Inverse Reinforcement Learning](). *Applied Artificial Intelligence*.

Investigation of *C. elegans* thermotactic behavior

- Two basic strategies are found
  - Directed Migration (DM): Worms efficiently reached specific temperatures, which explains their thermotactic behavior when fed.
  - Isothermal Migration (IM). Worms moved along a constant temperature, which reflects isothermal tracking, well-observed in previous studies.

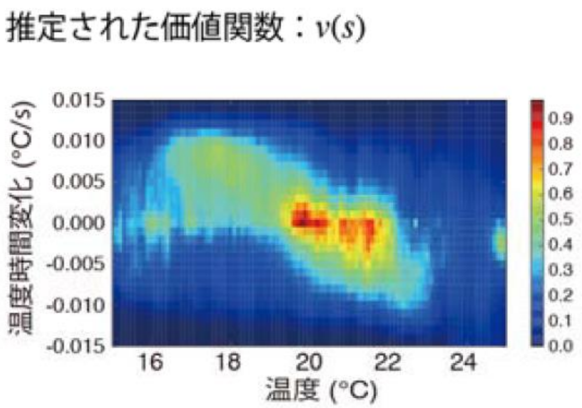
行動時系列データ → 逆強化学習法 → 行動戦略



線虫は育成された温度を好むように、また
飢餓を経験した温度を避けるように移動する。

行動戦略を示す数式

$$\pi(s'|s) = \frac{p(s'|s)\exp(v(s'))}{\sum_x p(x|s)\exp(v(x))}$$

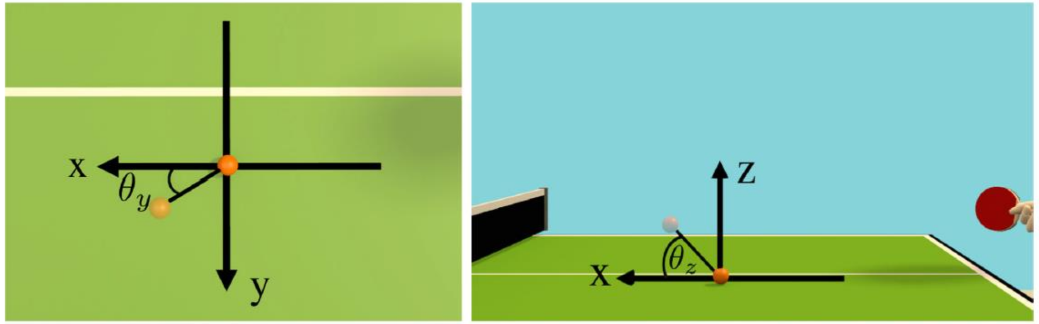


動物が行動していて遭遇する各状況が、戦略上
どれくらいの価値があるのかを示している。

Yamaguchi, S., Honda, N., Ikeda, M., Tsukada, Y., Nakano, S., Mori, I., and Ishii, S. (2018). Identification of animal behavioral strategies by inverse reinforcement learning. PLoS Computational Biology.

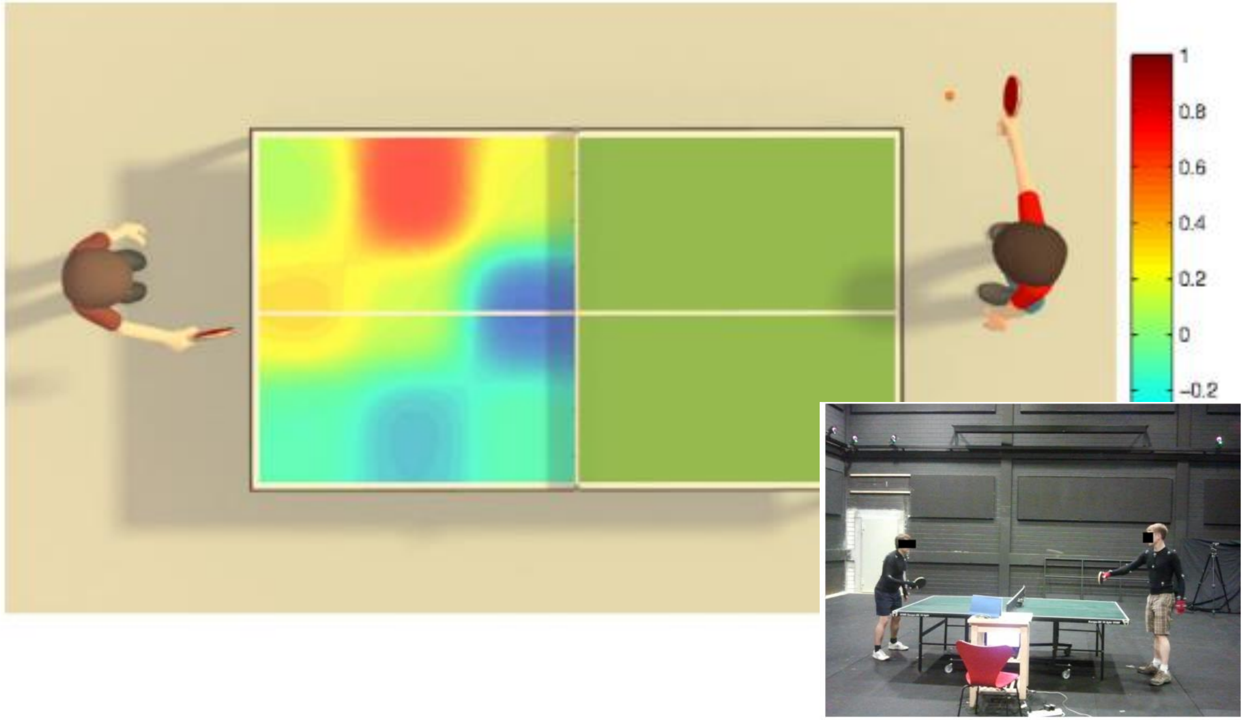
Table tennis

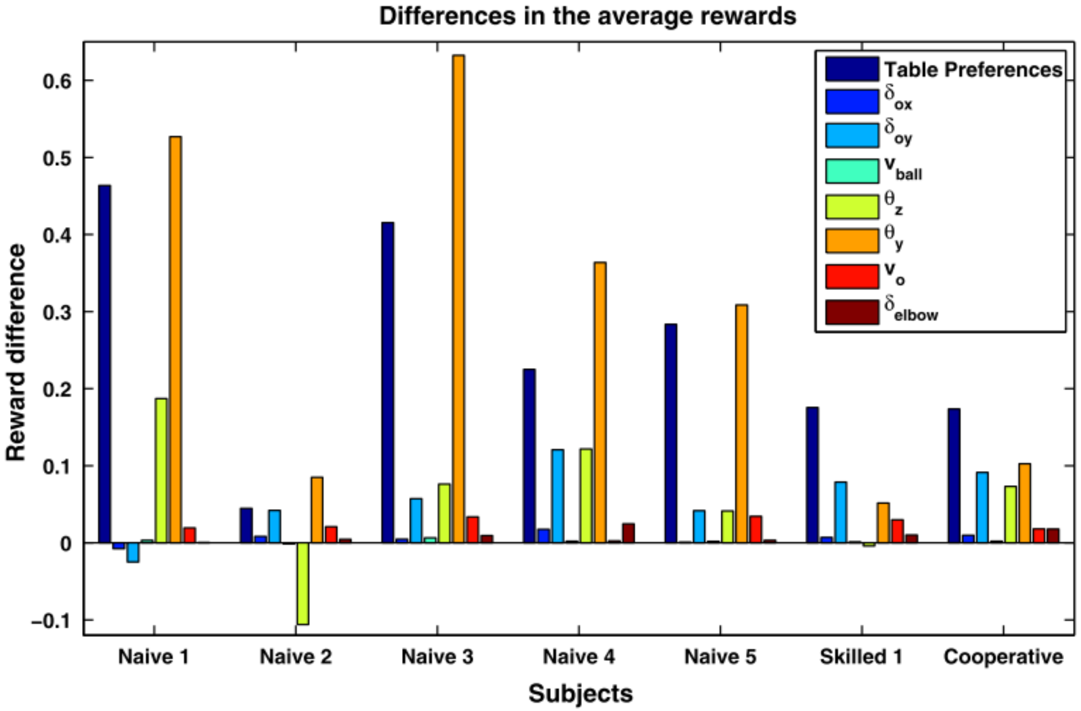
- Reward function for table preferences
- Individual player preferences



(a) θ_y **(b)** θ_z

Fig. 5 The bouncing angles θ_y and θ_z in the xy- and xz-surface define the orientation of the ball. While θ_z corresponds to the horizontal bouncing angle, θ_y corresponds to the direction of the ball and thereby defines if the ball is played cross to the *left*, cross to the *right* or straight

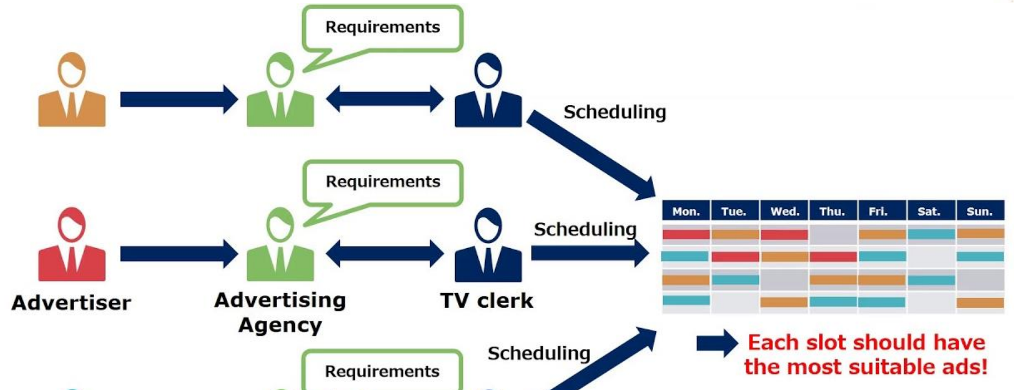


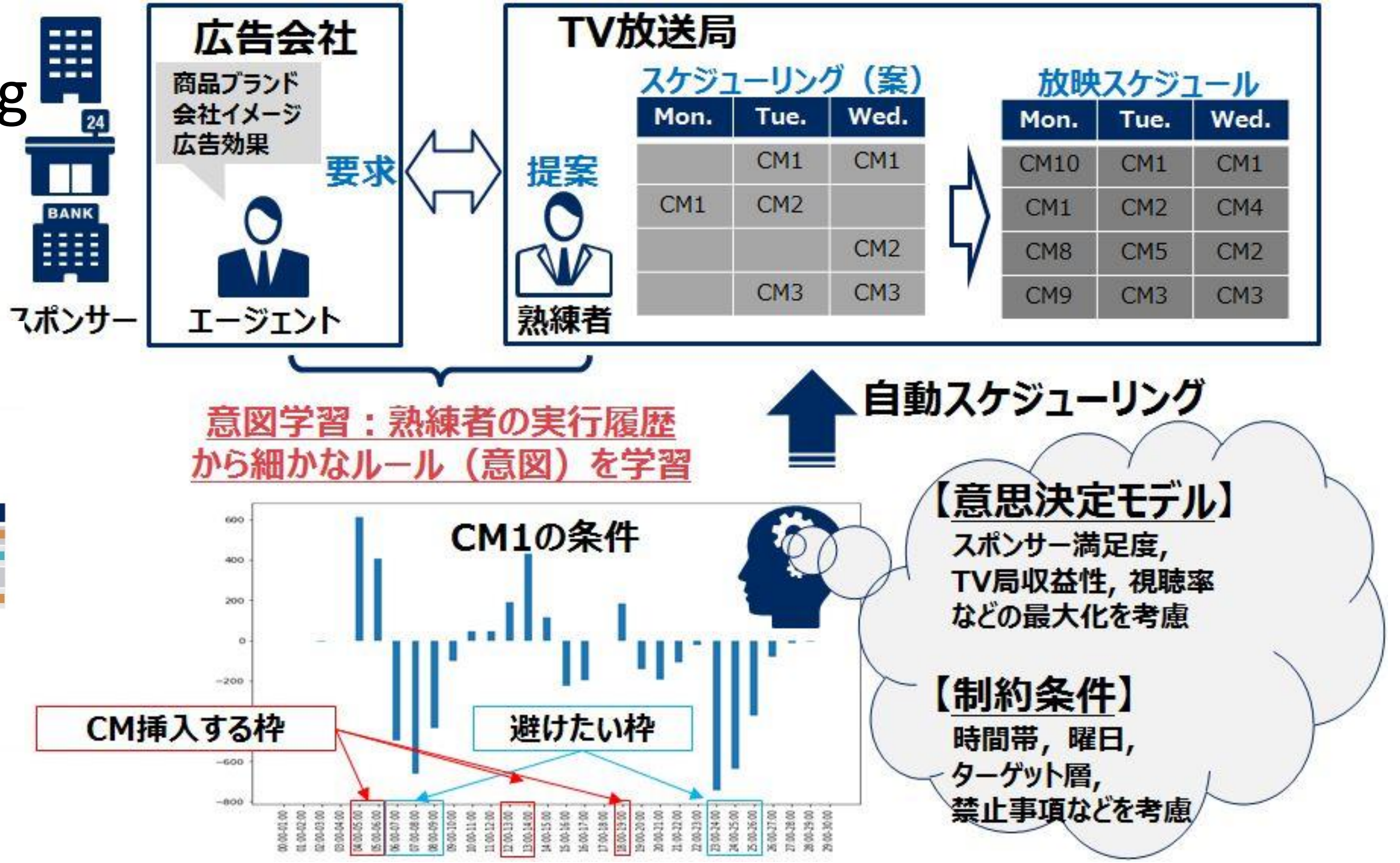


Muelling, K., Boularias, A., Mohler, B., Schölkopf, B., and Peters, J. (2014). Learning strategies in table tennis using inverse reinforcement learning. Biological Cybernetics, 108(5): 603-619.

TV Advertisement Scheduling by Learning Expert Intentions

- Imitate the decision-making process of scheduling experts





NECプレスリリース(2019/07/17)

Suzuki, Y., Wee, W.M., & Nishioka, I. (2019). TV Advertisement Scheduling by Learning Expert Intentions. In Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3071–81.