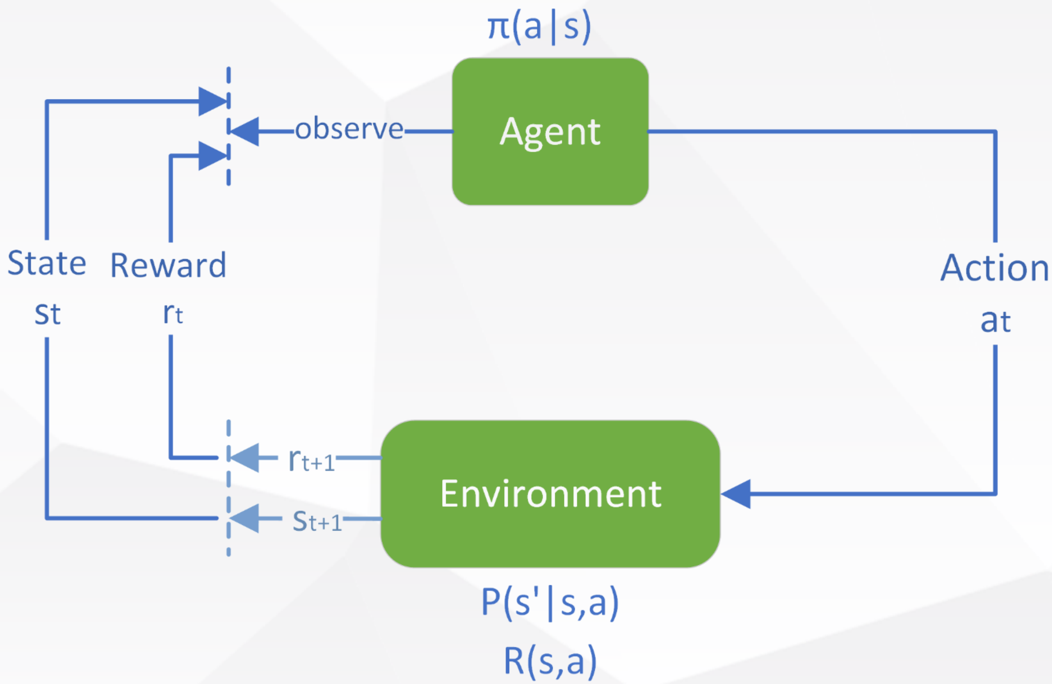
Rethinking data efficiency in reinforcement learning

workshop 2021/10/25

Reinforcement Learning

find a optimal strategies which maximize the return



The interaction process can be modeled as MDP $<\mathcal{S}, \mathcal{A}, R, P, \gamma, (D)>$

Reinforcement Learning

episode

$$S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

return

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

MC or TD

★ value

$$v_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$$

$$q_\pi(s, a) = \mathbb{E}_\pi [G_t | S_t = s, A_t = a]$$

ε- greedy policy

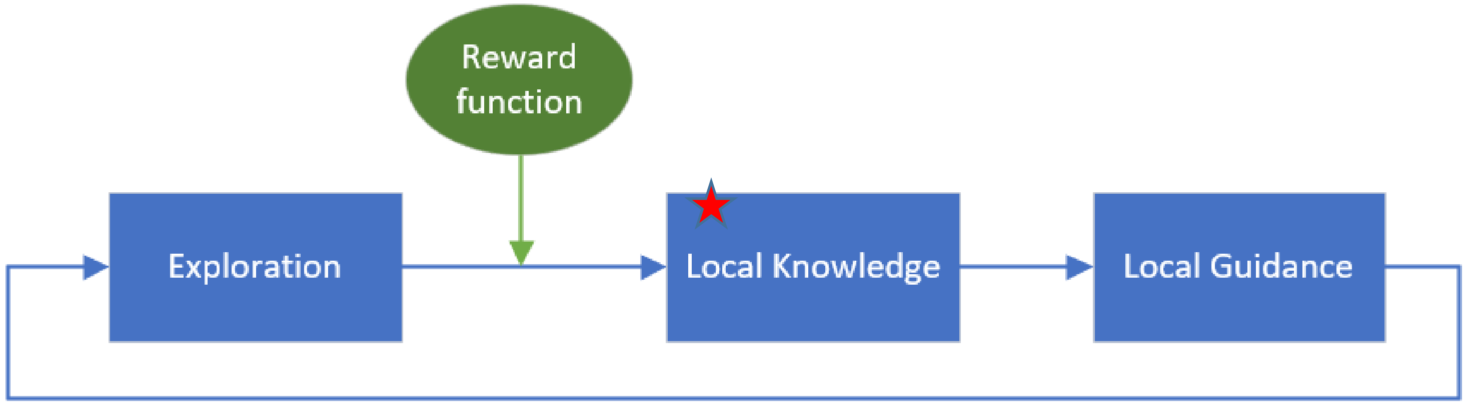
$$\pi(s) = \arg\max_a q_\pi(s, a) = \arg\max_a \mathbb{E}_\pi [r + \gamma v_\pi(s') | s, a]$$

random

$1 - \varepsilon$

ε

How to act



Reward function
Exploration
Local Knowledge
Local Guidance

from Reward to Value

$$q_\pi(s,a) = \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right]$$

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t | S_t = s\right]$$

- discrete space
  1. Bellman equation DP (model-based)

     $$q_\pi(s,a) = \mathbb{E}_\pi[r + \gamma q_\pi(s', a') | s, a]$$ policy evaluation

     $$q_*(s,a) = \mathbb{E}[r + \gamma \max_{a'} q_*(s', a') | s, a]$$ value iteration

  2. Monte Marlo (MC)

     $$v_n = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} g_t = \mathbb{E}[G_t]$$

     $$v_\pi(s_t) \leftarrow v_\pi(s_t) + \frac{1}{N(s_t)}(g_t - v_\pi(s_t))$$

  3. Temporal difference (TD) transition $(S_t, A_t, R_{t+1}, S_{t+1})$

     $$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)]$$ Sarsa (on-policy)

     $$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[\underbrace{r_{t+1} + \gamma \max_a Q(s_{t+1}, a)}_{\text{TD target}} - Q(s_t, a_t)]$$ Q-learning (off-policy)

     TD error

from Reward to Value

$$q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

- Continuous space (Function Approximation)
  1. value-based (DQN)

TD target — TD error

$$y_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a; w_t) \longrightarrow \delta_t = Q(s_t, a_t; w_t) - y_t$$

$$loss = \frac{1}{2}\delta_t^2 \longrightarrow g_t = \nabla_w Q(s_t, a_t; w_t)$$

$$w \leftarrow w - \alpha \cdot \delta_t \cdot g_t$$

  2. policy-based (policy gradient)

$$V_\pi(s;\theta) = \mathbb{E}_{A\sim\pi}[Q_\pi(s,a)] = \sum_a \pi(a|s;\theta) Q_\pi(s,a)$$

$$\text{target : maximize } \; J(\theta) = \mathbb{E}_S(V(S;\theta)) \xrightarrow{\text{MC}} \frac{\partial V(s;\theta)}{\partial\theta} = \mathbb{E}_{A\sim\pi(\cdot|s;\theta)}\left[\frac{\partial \log \pi(A|s;\theta)}{\partial\theta} Q_\pi(s,A)\right]$$

MC $\quad a_t \sim \pi(\cdot|s_t;\theta)$

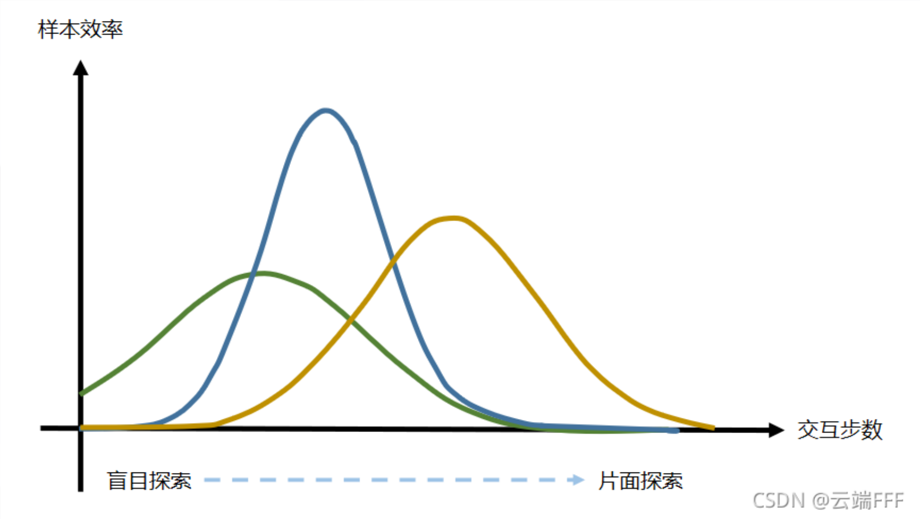
REINFORCE — Actor-Critic

MC — TD (sarsa)

$$g(a_t,\theta_t) = \boxed{Q_\pi(s_t,a_t)} \cdot d_{\theta,t} \longleftarrow d_{\theta,t} = \frac{\partial \log \pi(a_t|s_t;\theta)}{\partial\theta}\Big|_{\theta=\theta_t}$$

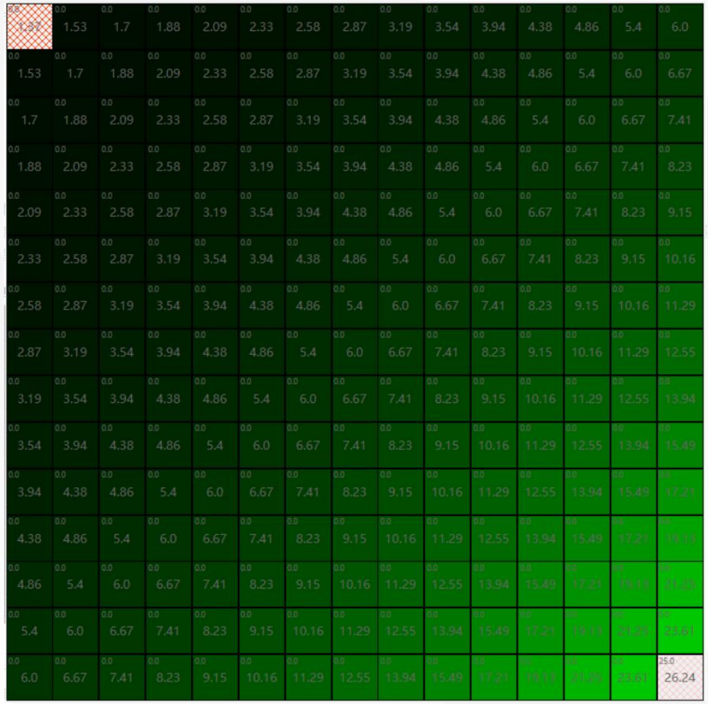
$$\theta_{t+1} \leftarrow \theta_t + \beta \cdot g(a_t;\theta_t)$$

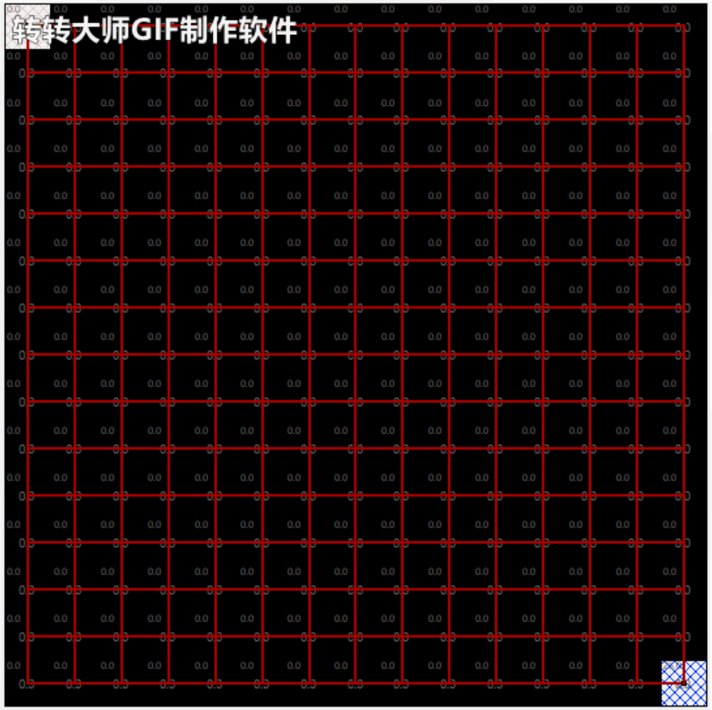
Exploration-Exploitation dilemma



Use TD error to evaluate how useful a transition sample is

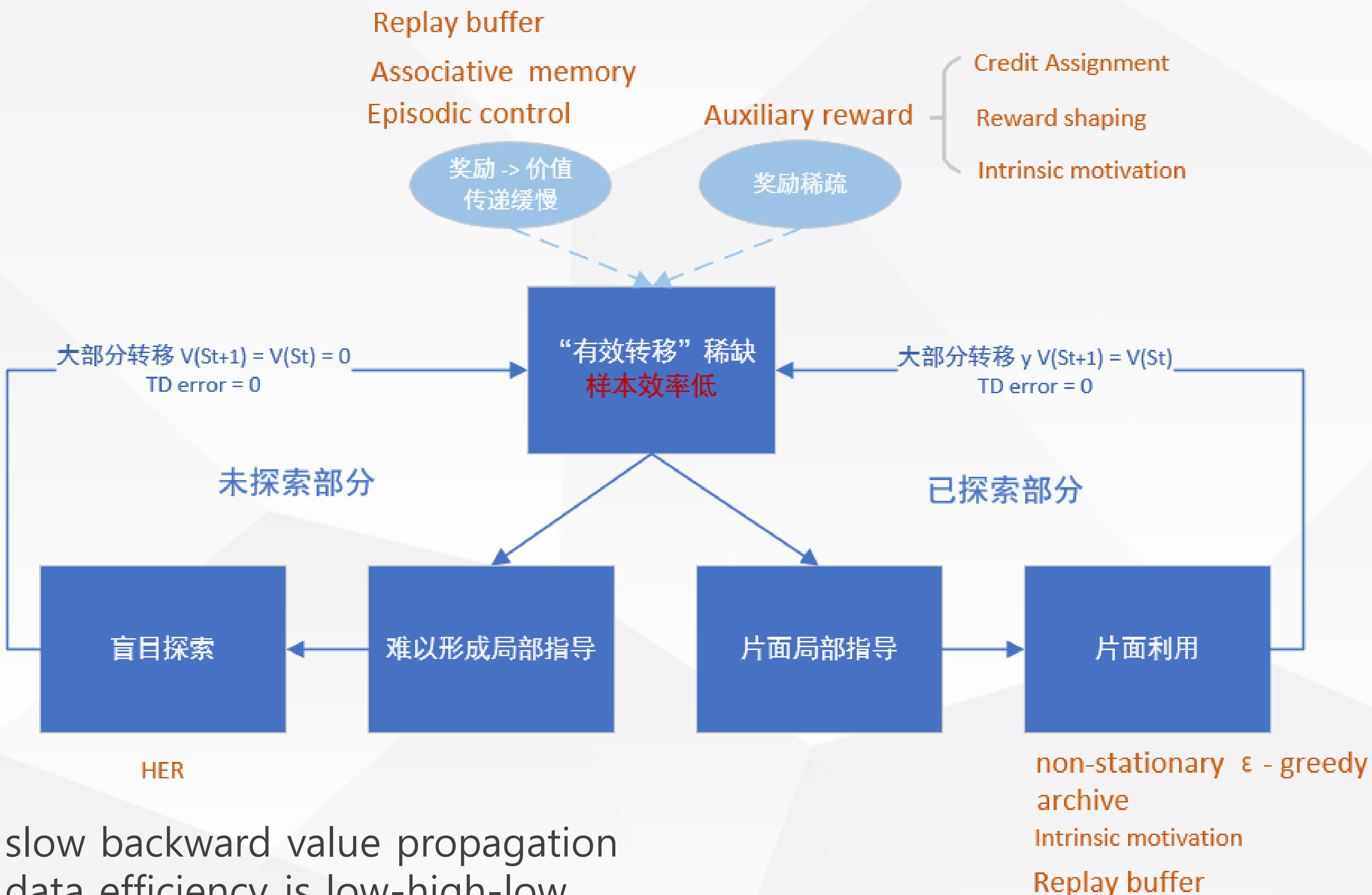
Experiment 1





$$\pi(s) = \arg\max_a \mathbb{E}_\pi[r + \gamma v_\pi(s')|s,a]$$
$$= \arg\max_a (r + \gamma V_\pi(s'|s,a))$$
$$= \arg\max_a V_\pi(s'|s,a)$$

Nature of vanilla model-free RL



1. slow backward value propagation
2. data efficiency is low-high-low
3. form blind exploration to one-sided exploitation

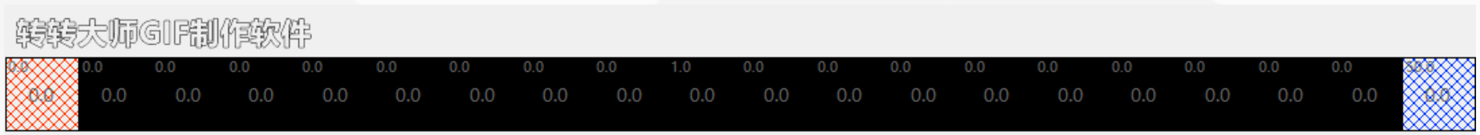
Experiment 2

ε descent greedy
1000 episodes

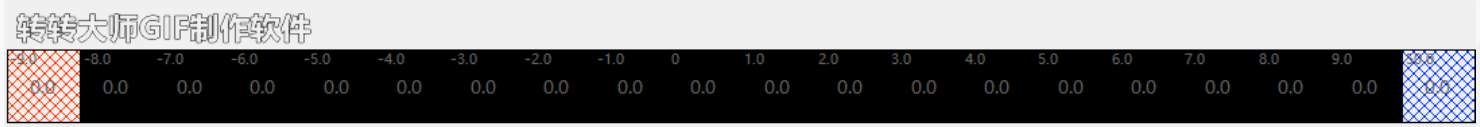
with replay buffer
500 episodes

Reward Experiment 1

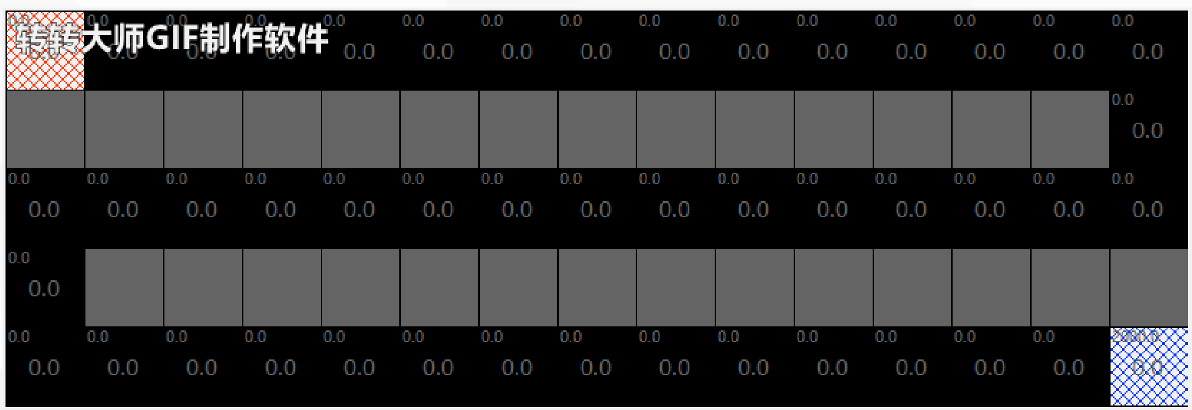
reward hack



encourage



cowardice



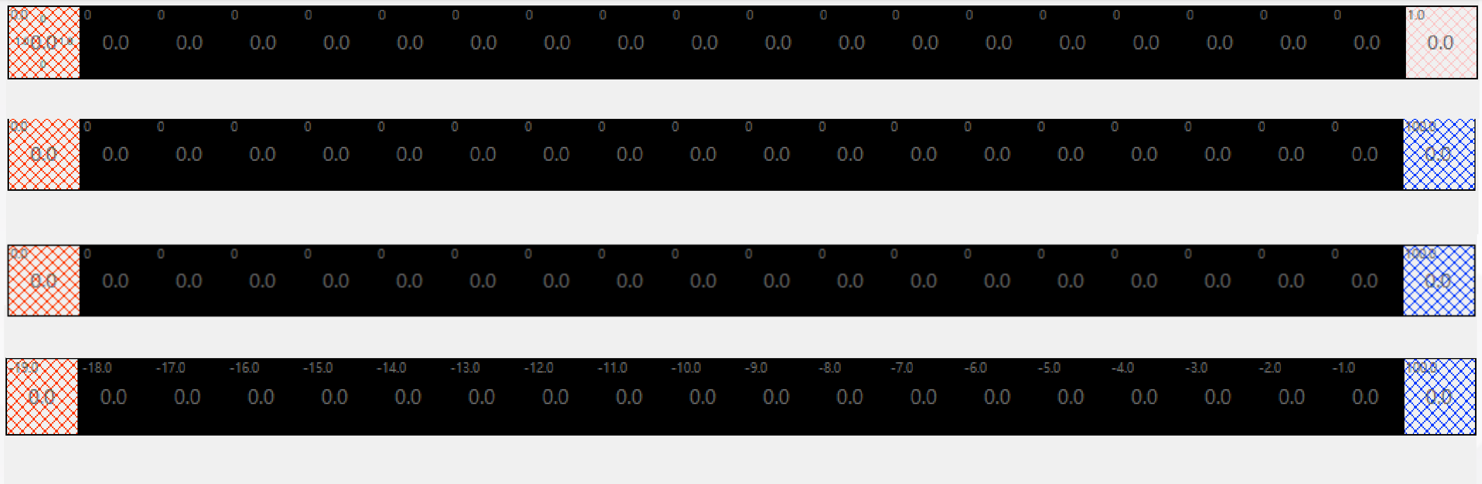
Reward Experiment 2

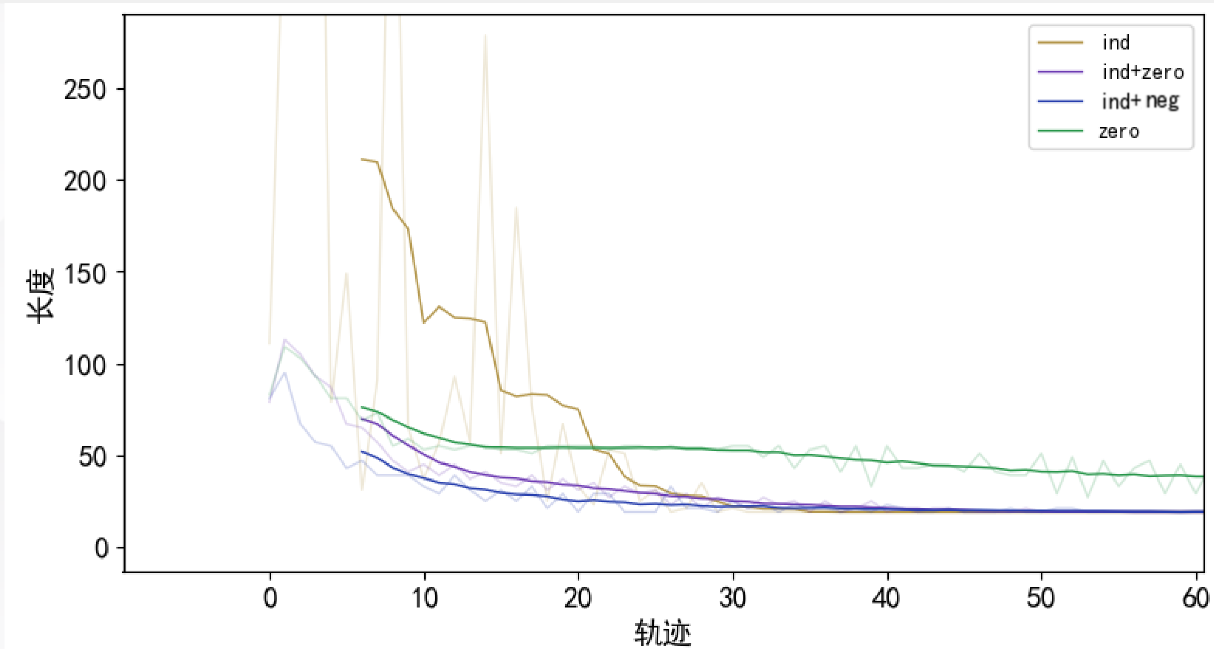
zero

ind

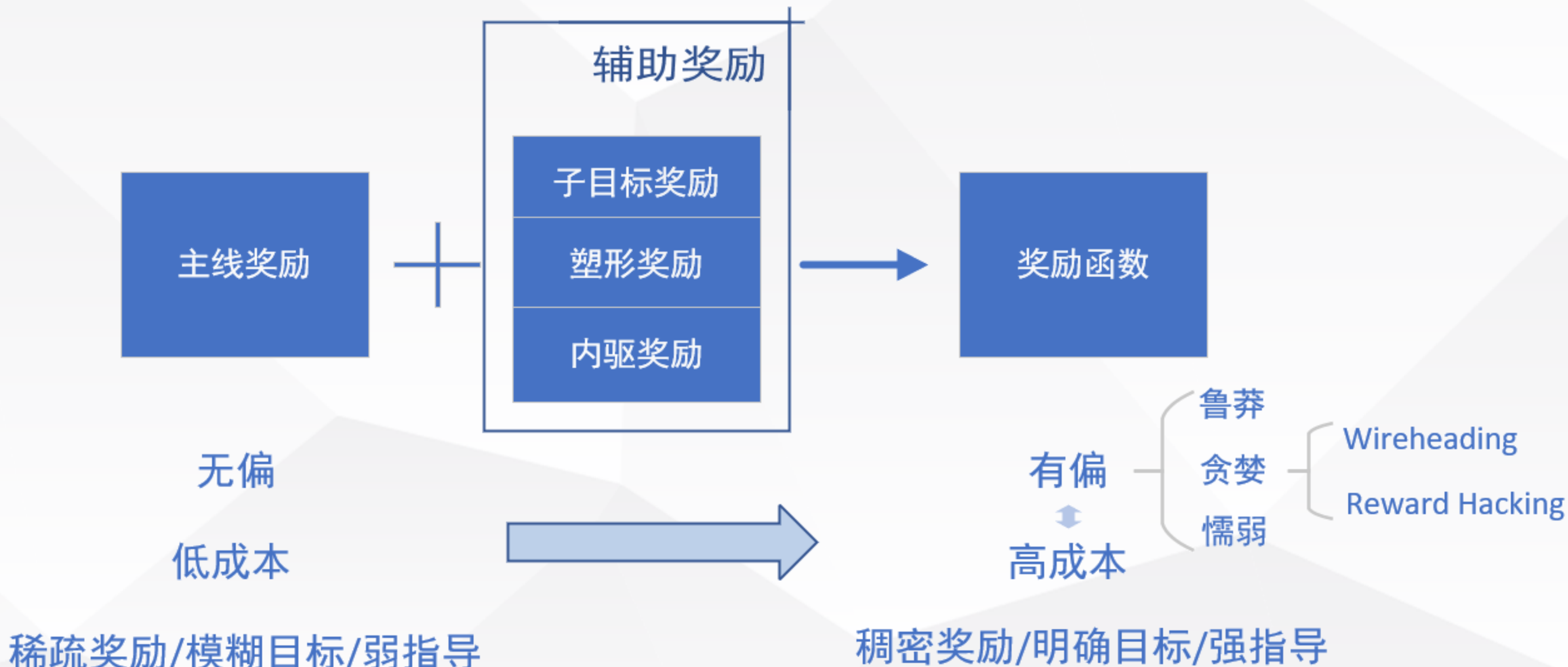
zero+ind

neg+in
d





Reward Experiment



Higher point of view

- We tell agents what the goal is through the reward function, which is indirect and abstract, resulting in low sample efficiency and misleading
- can we communicate our goals to agents in a different way ?

1. Give agent better initial value function
2. Through human preference[1]
3. Through expert demonstration
4. …

1 . Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 4302–4310.

CONTRASTIVE EXPLANATIONS FOR REINFORCEMENT LEARNING VIA EMBEDDED SELF PREDICTIONS

Zhengxian Lin, Kim-Ho Lam, Alan Fern
Department of EECS
Oregon State University
{linzhe, lamki, alan.fern}@oregonstate.edu

Motivation

- Explain one's action preference between actions A and B
  - for agent: explain by revealing action's predicted values, which provide little insight into its reasoning
  - for humans: explain via the impact on the expected future

- Can we give the RL agent similar capabilities?
  - explain in terms of expected futures
  - explainations sound in a rigorous way

Key Ideas

- learn meaningful properties of the expected future
  - Use Generalized Value Functions (GVFs) to capture meaningful properties of a policy's future episode

- Constract explicable Q-network
  - Use Embedded Self Prediction (ESP) model to embed GVFs into a Q-network

- Generate resonable explanation
  - Use Integrated gradient (IG) to show the influence of GFV features

- sound simplification
  - Use minimal sufficient explaination (MSX) to reduce the size of explanation

Generalized Value Functions (GVFs)

- A human understandable feature vector of state and action

$$F(s,a) = < f_1(s,a), f_2(s,a), \ldots, f_n(s,a) >$$

- GVF gives the expected future accumlation of F when following π after (s,a)

$$Q_F^{\pi}(s,a) = \mathbb{E}\left[F(s,a) + \gamma F(s',a') + \gamma^2 F(s'',a'') + \ldots\right]$$

- Compute GVF by iteration the *Bellman GVF operator*

$$B_F^{\pi}[Q_F] = F(s,a) + \gamma \sum_{s'} T(s,a,s') Q_F(s', \pi(s'))$$

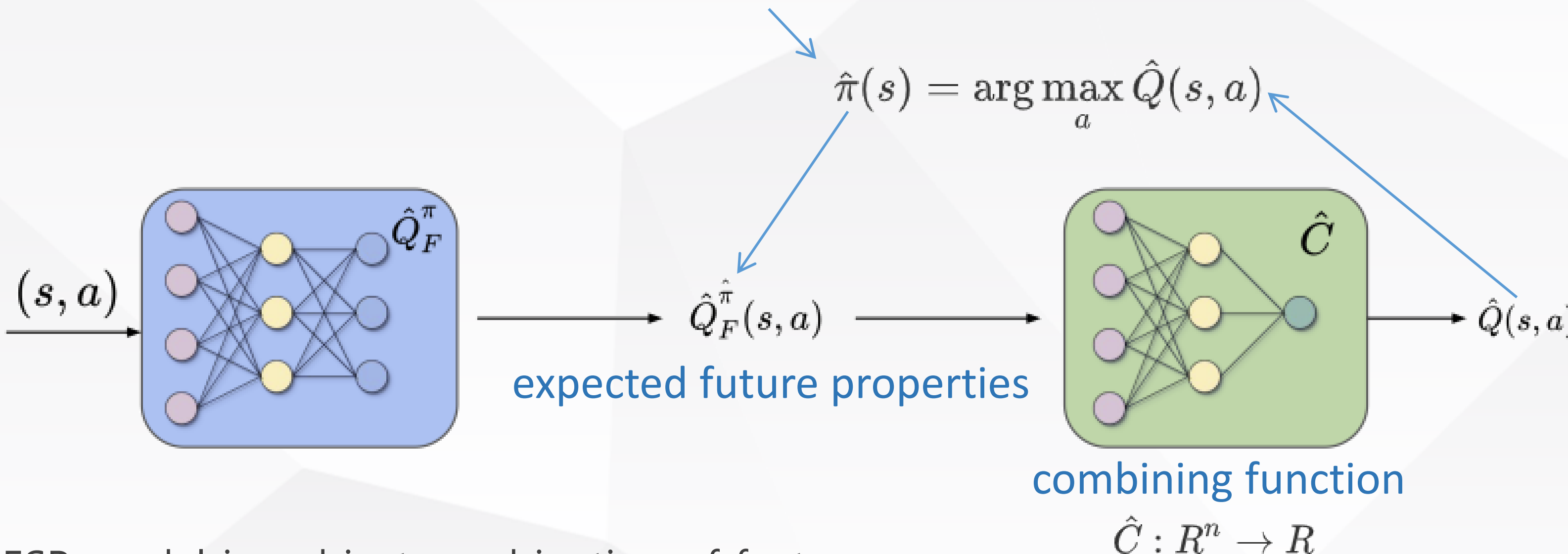
similar to Bellman optimal equation

$$B[Q](s,a) = R(s,a) + \beta \sum_{s'} T(s,a,s') max_{a'} Q(s',a')$$

- In general, GVF Q_F^{π} characterize behavior of π with respect to F

Embedded Self-Predictions (ESP) Model

- Directly use learned GVFs of agent's policy to compute action values

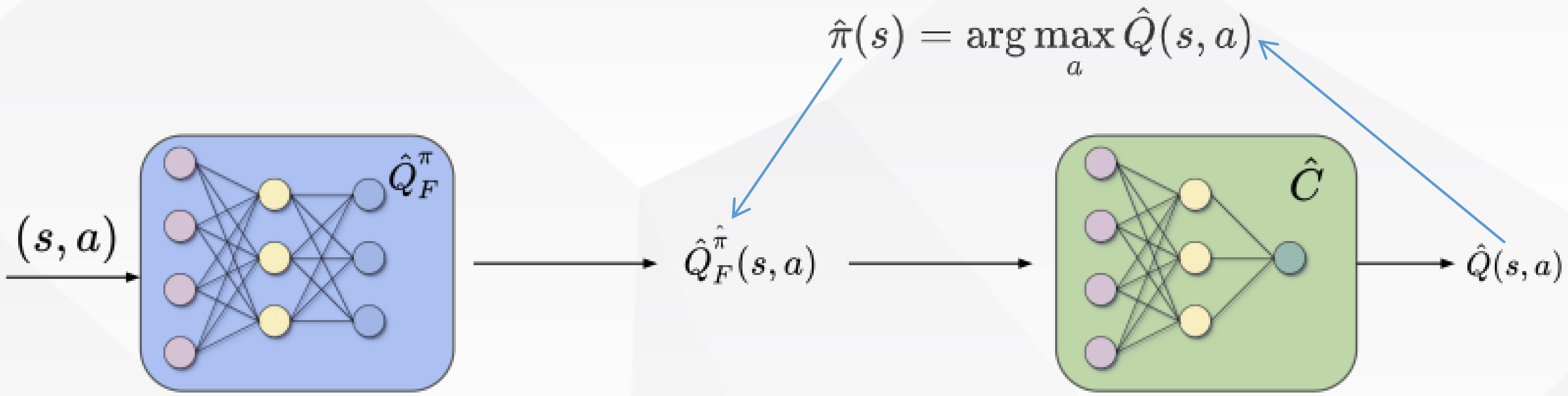


- ESP model is a driect combination of features

$$\hat{Q}(s,a) = \hat{C}(\hat{Q}_F(s,a))$$

- Special case
  - GVF discount factor $\gamma = 0$, ESP become direct combination of feature F(s,a)
  - Feature F(s,a) is reward signal, $\hat{C}$ can be a identity function

Training: ESP-DQN



- ESP model is cricularly defined : $\longrightarrow$ Strong Data correlation $\longrightarrow$ Unstable training
  - Internal GVFs is based on agent's own policy $\hat{\pi}(s)$
  - Agent's policy is computed by combining the GVFs

- Similar to DQN, use target network and replay buffer to break bootstrap
  - $\hat{C}$ should approximate Q^* , training it can use traditional DQN updates (fix Q_F^{π})
  - Training Q_F^{π} is similar to learning a critic in actor-critic methods

Training: ESP-DQN

Algorithm 1 ESP-DQN: Pseudo-code for ESP-DQN agent Learning.

Require: Act(s, a) ;; returns tuple $(s', r, F, done)$ of next state s', reward r, GVF features $F \in R^n$, and terminal state indicator $done$
Require: K - target update interval, β - reward discount factor, γ - GVF discount factor
  Init $\hat{Q}_F$, $\hat{Q}'_F$;; The non-target and target GVF networks with parameters θ_F and θ'_F respectively.
  Init $\hat{C}$, $\hat{C}'$;; The non-target and target combining networks with θ_C and θ'_C rspectively.
  Init $M \leftarrow \emptyset$;; initialize replay buffer
  ;; Q-function is defined by $\hat{Q}(s,a) = \hat{C}(\hat{Q}_F(s,a))$
  ;; Target Q-function is defined by $\hat{Q}'(s,a) = \hat{C}'(\hat{Q}'_F(s,a))$
  repeat
    Environment Reset $s_0 \leftarrow$ Initial State totalUpdates $\leftarrow 0$
    for $t \leftarrow 0$ to T **do**
      $a_t \leftarrow \epsilon(\hat{Q}, s_t)$ // ϵ-greedy
      $(s_{t+1}, r_t, F_t, done_t) \leftarrow$ Act(s_t, a_t)
      Add $(s_t, a_t, r_t, F_t, s_{t+1}, done_t)$ to M

Training: ESP-DQN

;; update networks

Randomly sample a mini-batch $\{(s_i, a_i, r_i, F_i, s'_i, done_i)\}$ from M

$\hat{a}_i \leftarrow \arg\max_{a \in A} \hat{Q}'(s'_i, a)$

TD target $f'_i \leftarrow \begin{cases} F_i & \text{If } done_i \text{ is } \textbf{true} \\ F_i + \gamma \hat{Q}'_F(s'_i, \hat{a}_i) & \text{Otherwise} \end{cases}$

TD target $q'_i \leftarrow \begin{cases} r_i & \text{If } done_i \text{ is } \textbf{true} \\ r_i + \beta \hat{Q}'(s'_i, \hat{a}_i) & \text{Otherwise} \end{cases}$

Update θ_F via gradient descent on average mini-batch loss $(f'_i - \hat{Q}_F(s_i, a_i))^2$ TD error

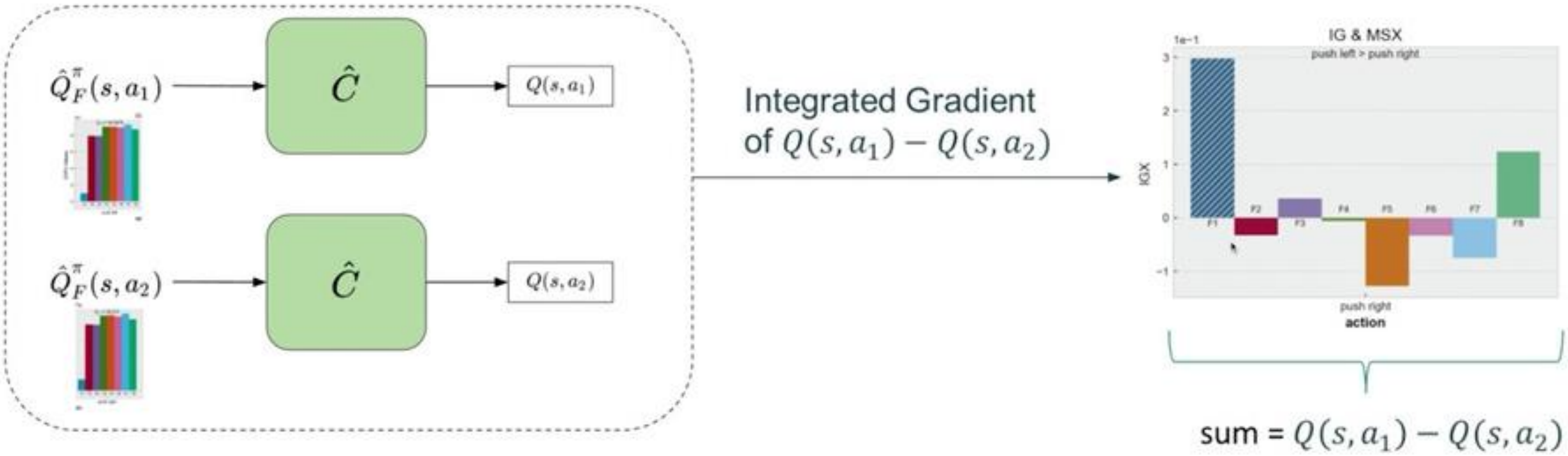
Update θ_C via gradient descent on average mini-batch loss $(q'_i - \hat{Q}(s_i, a_i))^2$ TD error

if totalUpdates mod $K == 0$ **then**
    $\theta'_F \leftarrow \theta_F$
    $\theta'_C \leftarrow \theta_C$
end if
totalUpdates $\leftarrow$ totalUpdates + 1

if $done_t$ is **true then**
    break
end if
end for
until convergence

Contrastive Explanations

Why is a_1 preferred over a_2 in state s?



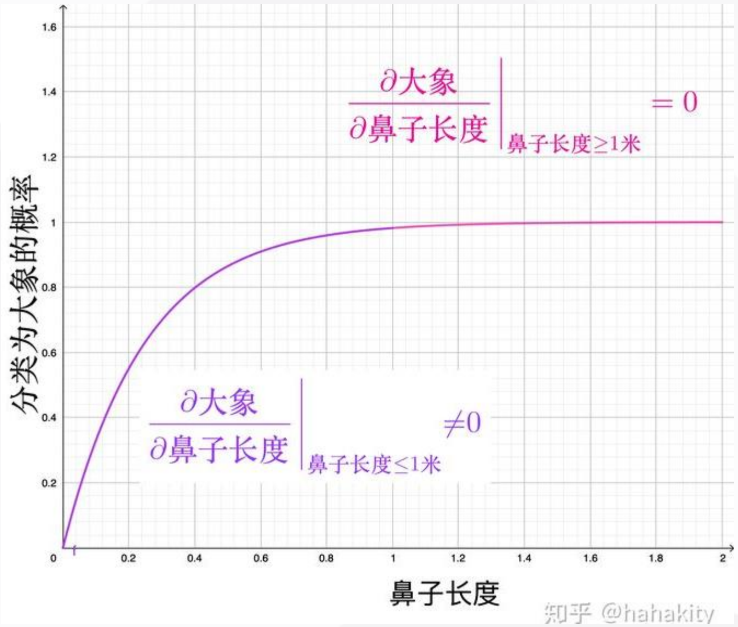
attribution weight vector R^n

$$\boxed{\hat{Q}(s,a) - \hat{Q}(s,b)} = \boxed{W(s,a,b)} \cdot \boxed{\Delta_F(s,a,b)}$$

preference magnitude

GVF difference vector

Integrated Gradient



$$特征重要性 = \int_{0米}^{2米} \frac{\partial 大象}{\partial 鼻子长度} d鼻子长度$$

$$\mathrm{IntegratedGrad}_i(x) ::= (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

the less preferred action ← Baseline

$$\theta_i(s,a,b) = \int_0^1 \frac{\partial \hat{C}(X_{sb} + \alpha(X_{sa} - X_{sb}))}{\partial X_{sa,i}} d\alpha \qquad X_{sa} = \hat{Q}_F(s,a) \quad X_{sb} = \hat{Q}_F(s,b)$$

- The key property is that the IG weights linearly attributes feature differences to the overall output difference

$$\hat{Q}(s,a) - \hat{Q}(s,b) = \hat{C}(\hat{Q}_F(s,a)) - \hat{C}(\hat{Q}_F(s,b)) = \theta(s,a,b) \cdot \triangle_F(s,a,b)$$

$\mathrm{IGX}(s,a,b) = \langle \Delta_F(s,a,b), \theta(s,a,b) \rangle$ is a sound explanation

minimal sufficient explaination (MSX)

- When there are many features IGX(s, a, b) will likely overwhelm users.
- Use the concept of minimal sufficient explanation (MSX) to soundly reduce the size

$P = \{i : \triangle_{F,i}(s,a,b) \cdot \theta_i(s,a,b) > 0\}$ positive attribution components indices

$N = \{1, ..., n\} - P$ negative attribution components indices

$S(E) = \sum_{i \in E} |\triangle_{F,i}(s,a,b) \cdot \theta_i(s,a,b)|$ total magnitude of the components

- Often only a small subset of positive components are required to overcome negative components and maintain the preference of a over b

$$\arg\min\{|E| \; : \; E \subseteq P, S(E) > S(N)\}$$

- Sort P and include indices into the MSX from largest to smallest until the total is larger than S(N).

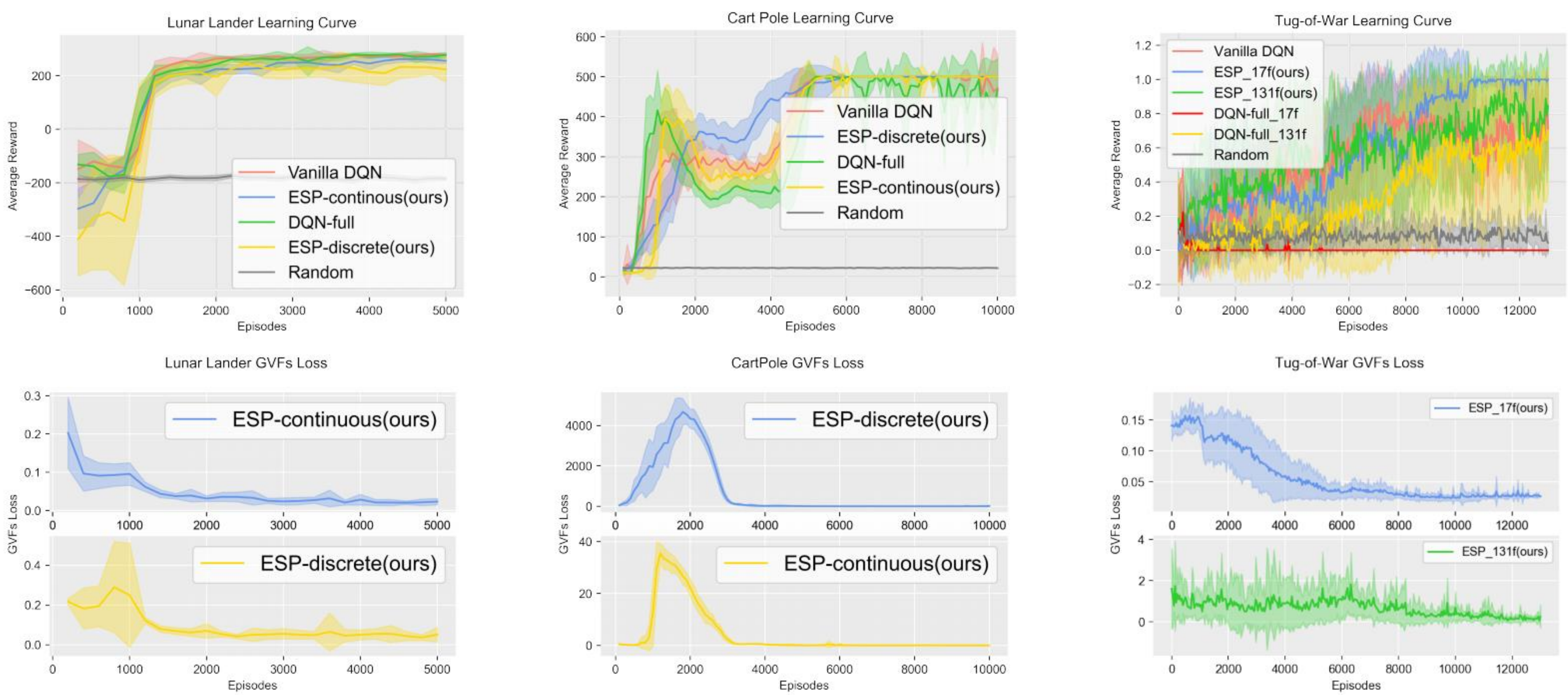
Experiment

- generic GVF features
  - terminal reward: describe basic conditions at the end of the episode
  - pre-terminal reward: the state variables of the environment or derived reward variables, typically readily available from a domain description

- Discrete state or reward variables -> indicator GVF features

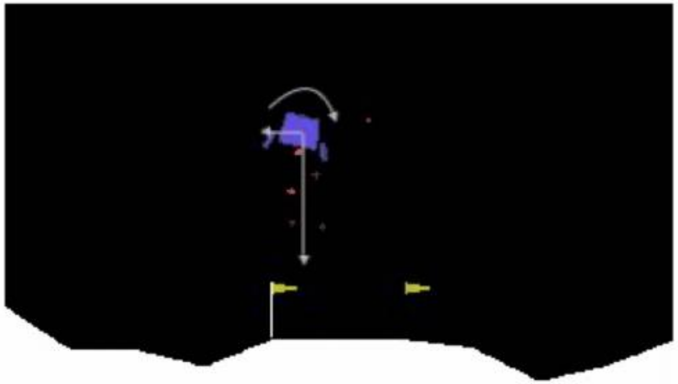
- Continuous state and reward variables
  - indicator features: for the regions as features (When a variable has a small number of meaningful regions)
  - Delta GVF features: the change in a variable across a time step

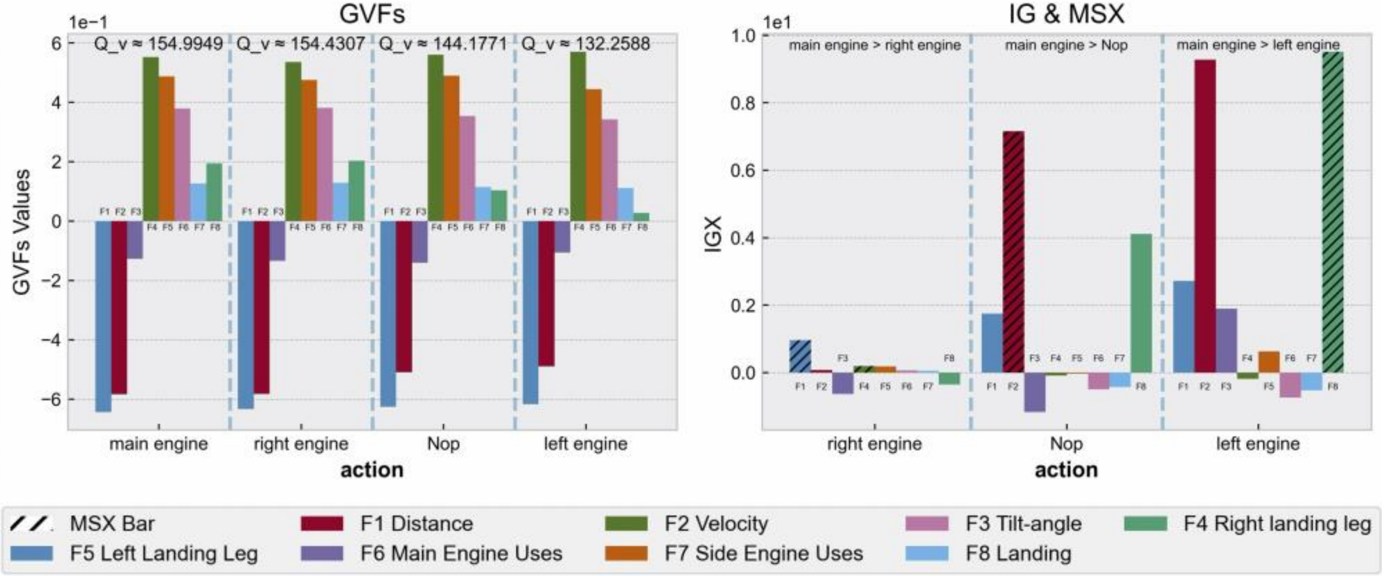
Experiment



(a) Lunar Lander (b) Cart Pole (c) Tug-of-war

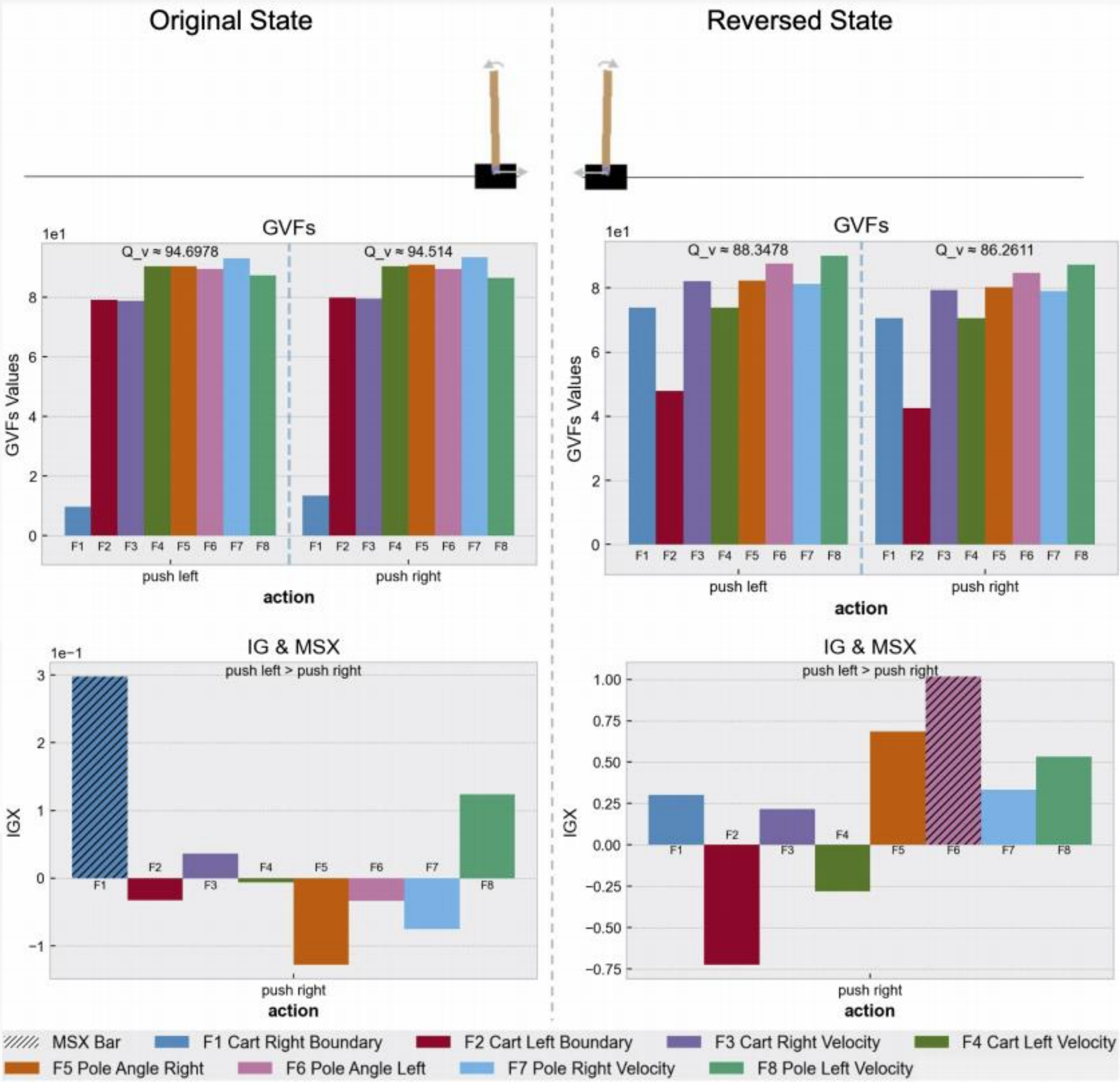
Experiment





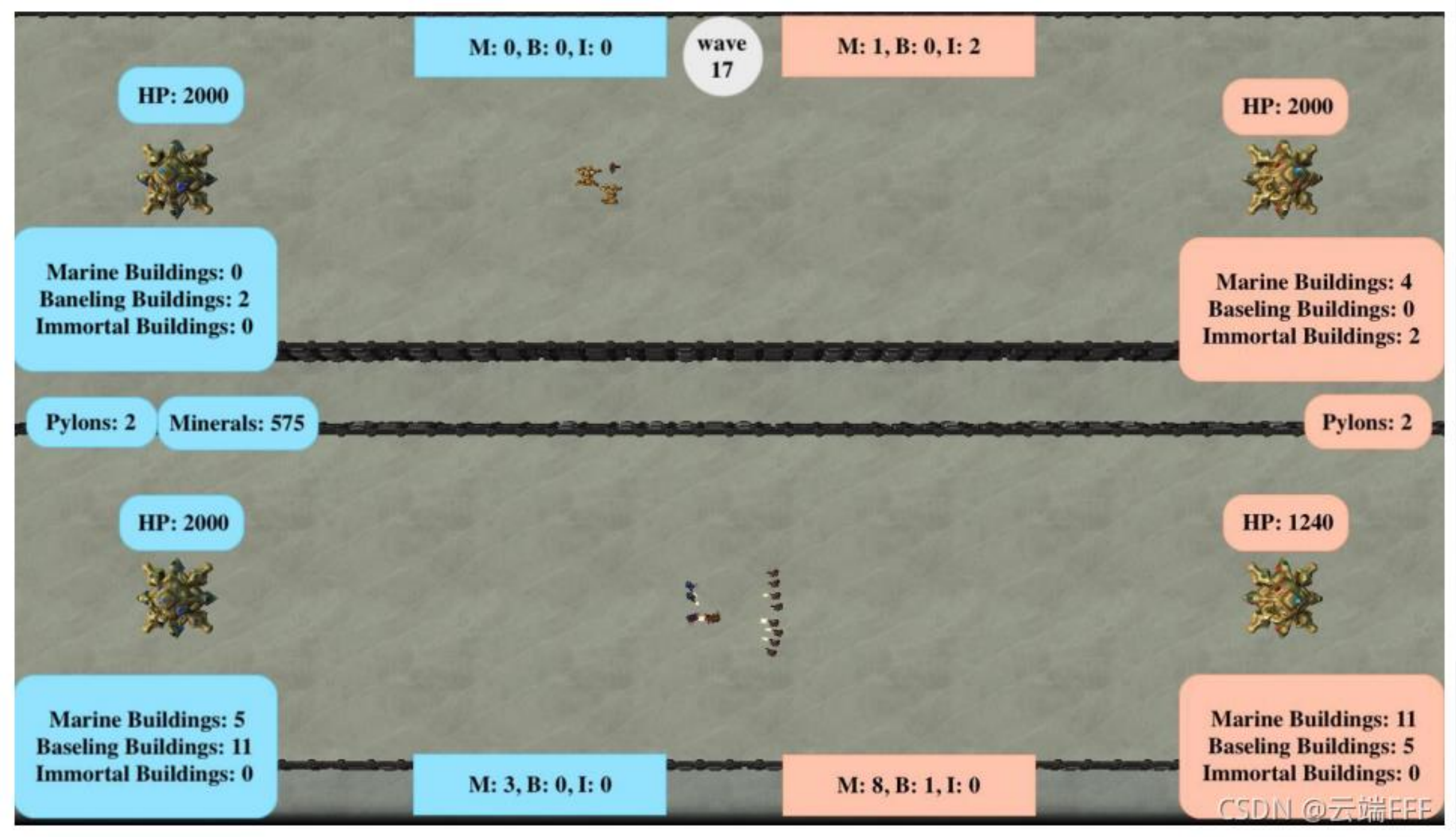
GVFs
IG & MSX
1e−1
1e1
Q_v ≈ 154.9949
Q_v ≈ 154.4307
Q_v ≈ 144.1771
Q_v ≈ 132.2588
main engine > right engine
main engine > Nop
main engine > left engine
GVFs Values
IGX
main engine
right engine
Nop
left engine
action
MSX Bar
F1 Distance
F2 Velocity
F3 Tilt-angle
F4 Right landing leg
F5 Left Landing Leg
F6 Main Engine Uses
F7 Side Engine Uses
F8 Landing

Experiment



Original State
Reversed State
GVFs
1e1
Q_v ≈ 94.6978
Q_v ≈ 94.514
Q_v ≈ 88.3478
Q_v ≈ 86.2611
GVFs Values
F1 F2 F3 F4 F5 F6 F7 F8
push left
push right
action
IG & MSX
1e−1
push left > push right
IGX
MSX Bar
F1 Cart Right Boundary
F2 Cart Left Boundary
F3 Cart Right Velocity
F4 Cart Left Velocity
F5 Pole Angle Right
F6 Pole Angle Left
F7 Pole Right Velocity
F8 Pole Left Velocity

Experiment


M: 0, B: 0, I: 0
wave 17
M: 1, B: 0, I: 2
HP: 2000
HP: 2000
Marine Buildings: 0
Baneling Buildings: 2
Immortal Buildings: 0
Marine Buildings: 4
Baseling Buildings: 0
Immortal Buildings: 2
Pylons: 2
Minerals: 575
Pylons: 2
HP: 2000
HP: 1240
Marine Buildings: 5
Baseling Buildings: 11
Immortal Buildings: 0
Marine Buildings: 11
Baseling Buildings: 5
Immortal Buildings: 0
M: 3, B: 0, I: 0
M: 8, B: 1, I: 0