[Algorithm Design]

VRAIL

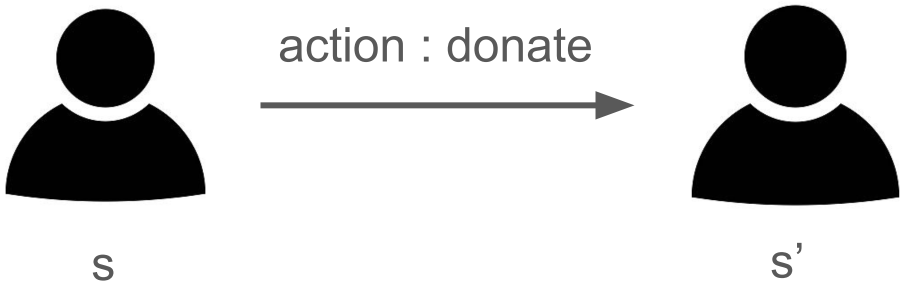
Vectorized Reward-based Attribution for Interpretable Learning

Team #02
Jina Kim, Youjin Jang, Jeongjin Han

Contents

Motivation

- Setting Reward is though (“Reward is Enough”)
- State(observation) can contain lots of information which is related to reward
- Human donates money → [-money, +emotion]
    - positive/negative reward?
    - Humans choose actions based on their own subjective weighting of different factors
- So, we can shape reward by modeling changes in possession with value



Related work

Theorem: Potential-Based Reward Shaping

Let any S, A, γ, and any shaping reward function $F : S \times A \times S \to \mathbb{R}$ be given.

We say F is a **potential-based** shaping function if there exists a real-valued function $\Phi : S \to \mathbb{R}$ such that for all $s \in S \setminus \{s_0\}, a \in A, s' \in S$,

$$F(s, a, s') = \gamma\Phi(s') - \Phi(s),$$

(where $S \setminus \{s_0\} = S$ if $\gamma < 1$). Then, F is potential-based shaping function is a necessary and sufficient condition for it to guarantee consistency with the optimal policy.

Ng et al. (1999), Potential-based Reward Shaping.

Method

$$R'(s, a, s') = R(s, a, s') + \gamma V(s') - V(s)$$

Our expectation:

- Learnable reward shaping could mitigate the effects of limited state information
- Simply approximated weights could capture state feature importance, improving interpretability and guiding human decisions
- Could contribute to faster convergence

Method

Bi-level Optimization

RL stage

$x \subseteq s$: feature in state

$R' = R + \gamma f_w(x') - f_w(x)$

- update $Q(s, a)$
- $V(s) = \max_{a \in A}(Q(s, a))$

ex) Value-based RL: **DQN**,
Policy iteration, SARSA, Q-learning

$$\text{RL stage} \xrightarrow{(x, V(s))} \text{DL stage}$$

$$\text{DL stage} \xrightarrow{f_w(x)} \text{RL stage}$$

DL stage

$x \subseteq s$: feature in state

supervised learning with $(x, V(s))$

estimator $f_w(x)$

- loss: $\dfrac{1}{n}\sum(V(s) - f_w(x))^2$

ex) linear regression $f_w(x) = w^\top x$,
quadratic approximation $f_w(x) = x^\top W x$

Environment (Taxi)

Actions

There are 6 discrete deterministic actions:

- 0: move south
- 1: move north
- 2: move east
- 3: move west
- 4: pickup passenger
- 5: drop off passenger

Rewards

- -1 per step unless other reward is triggered.
- +20 delivering passenger.
- -10 executing “pickup” and “drop-off” actions illegally.

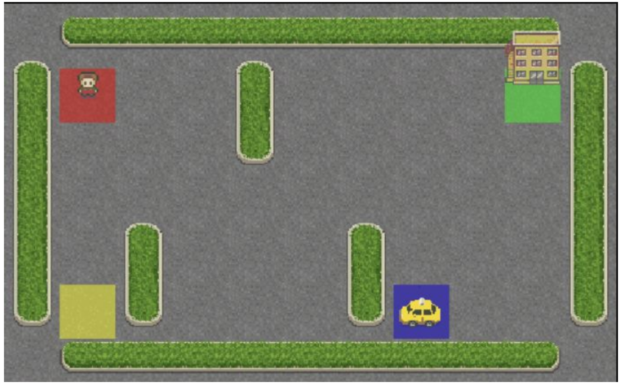
Observation Space

Passenger locations:

- 0: R(ed)
- 1: G(reen)
- 2: Y(ellow)
- 3: B(lue)
- 4: in taxi

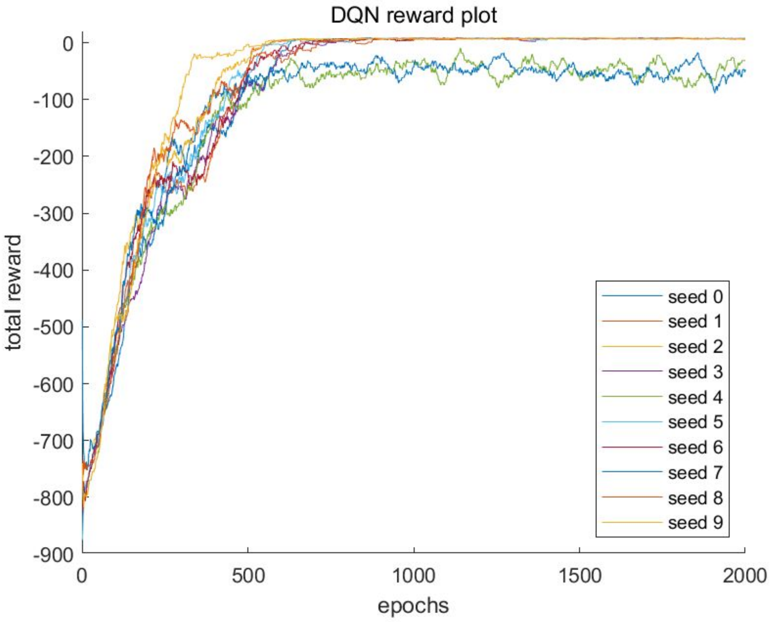
Destinations:

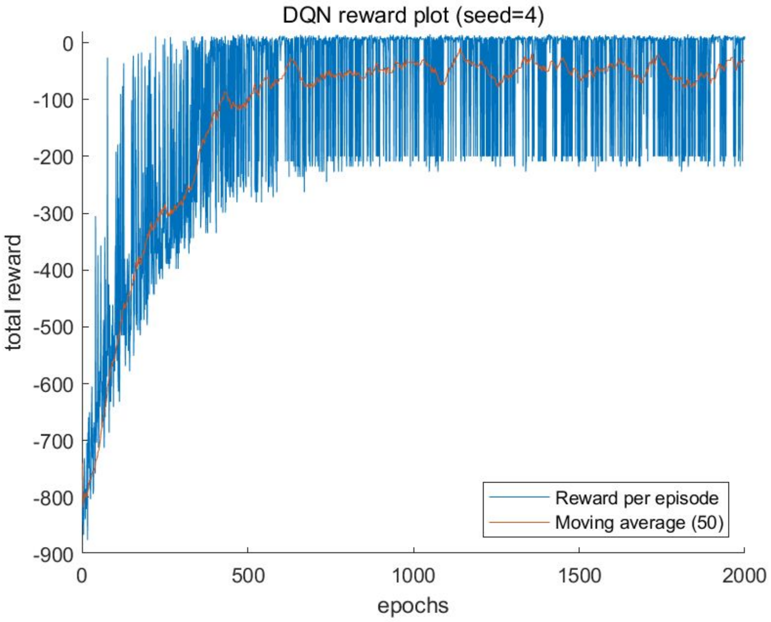
- 0: R(ed)
- 1: G(reen)
- 2: Y(ellow)
- 3: B(lue)



Performance

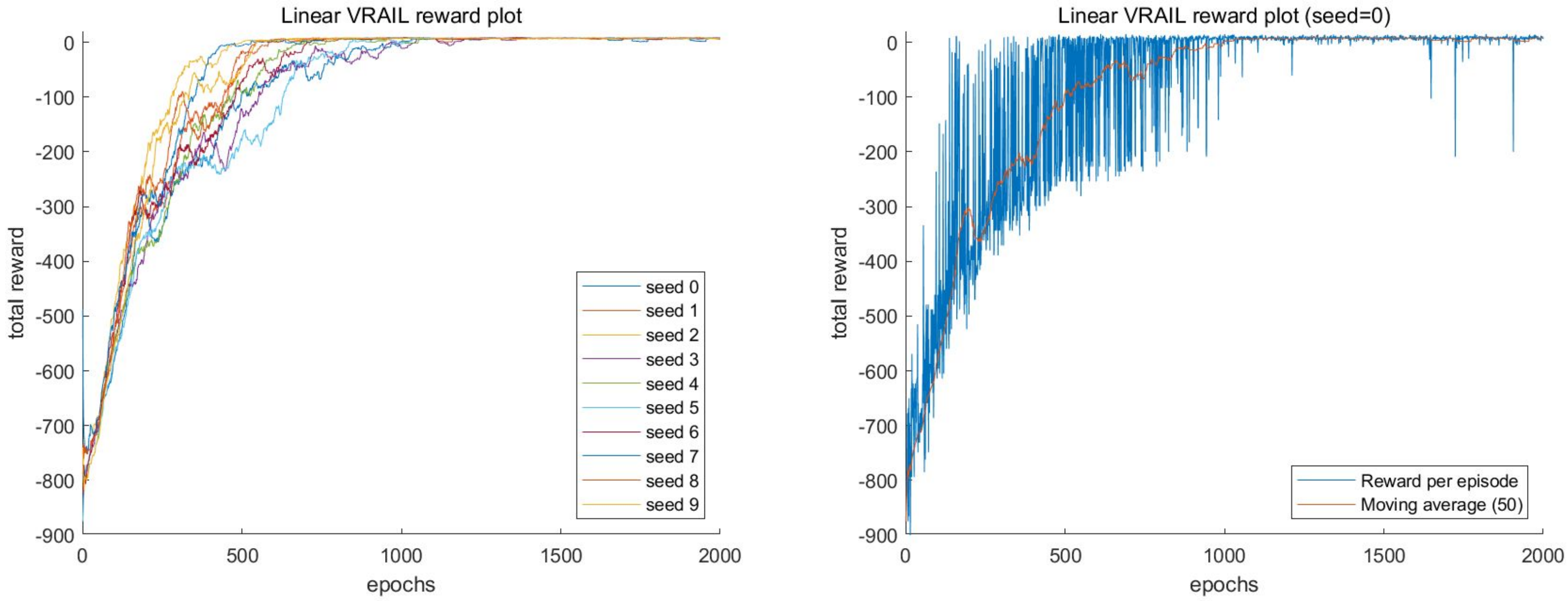
- Baseline, DQN : 8/10 converges





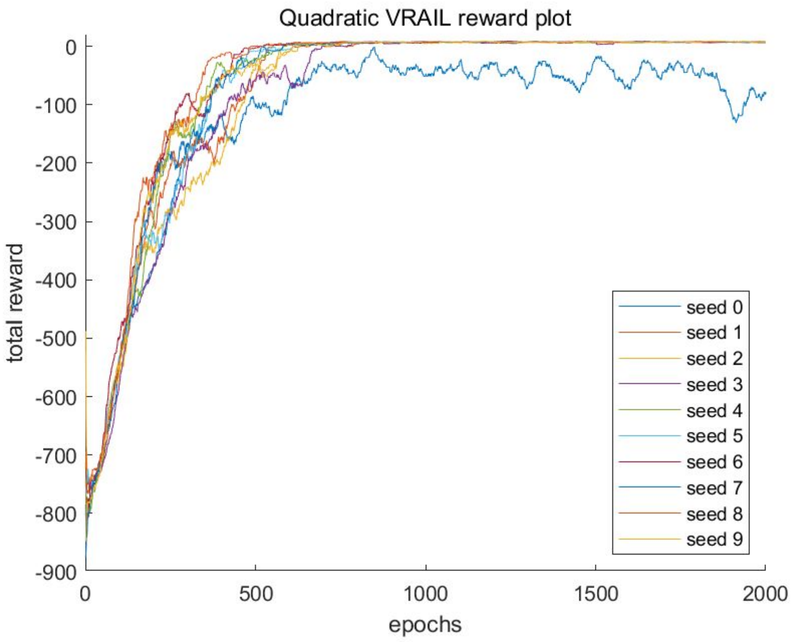
Performance

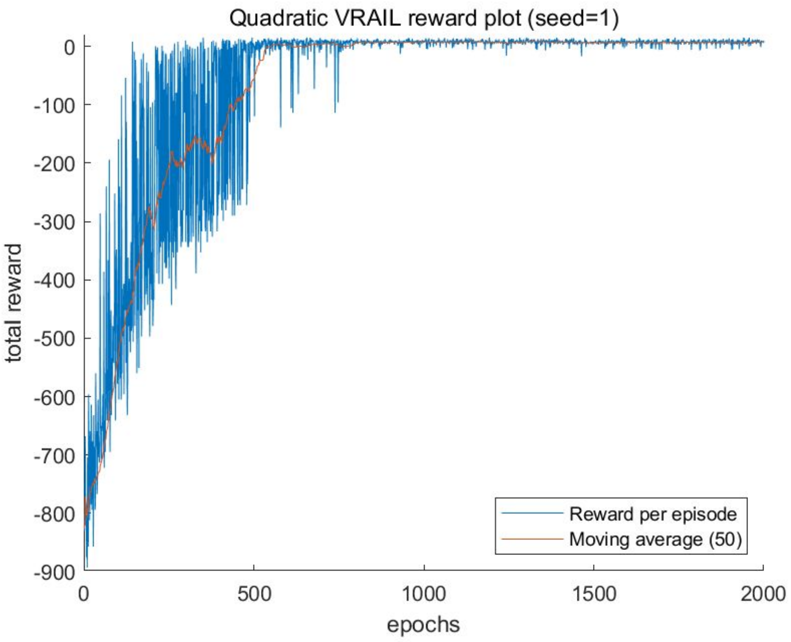
- Linear VRAIL, $V(s) = \mathbf{w}^\top \mathbf{x}$: 10/10 converges



Performance

- Quadratic VRAIL, $V(s) = \mathbf{x}^\top W \mathbf{x}$: 9/10 converges





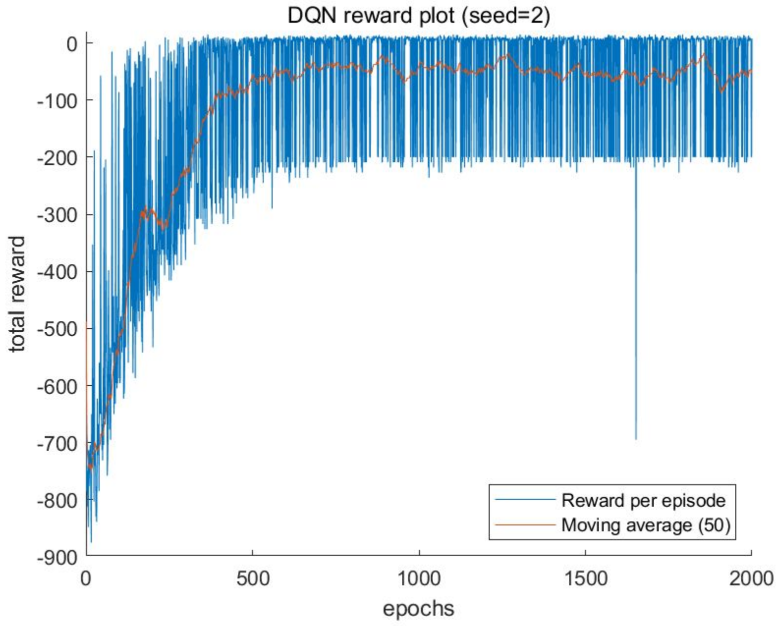
Average Epochs to Reach Reward Threshold

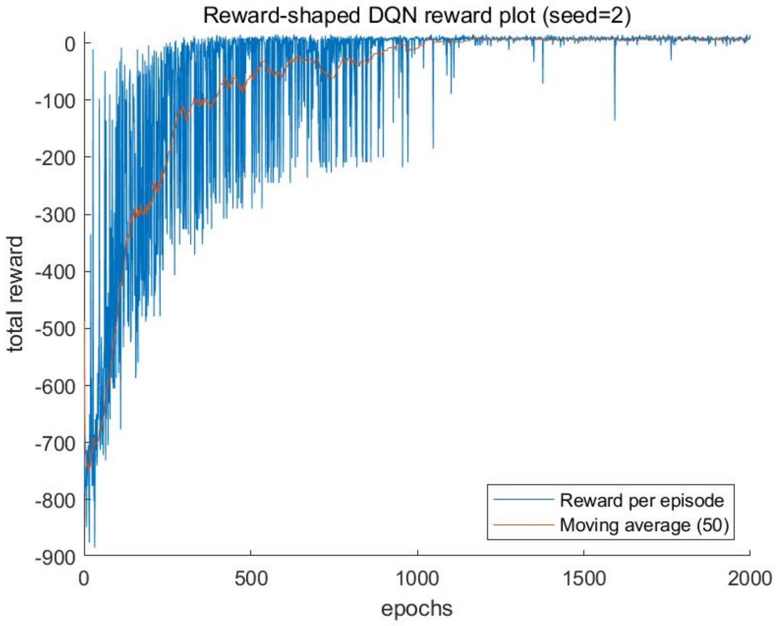
Reward Threshold	-10	-5	0	$+5$
DQN (epochs)	600.00	612.17	648.17	717.67
Linear VRAIL (epochs)	614.17	643.17	652.50	735.83
Quadratic VRAIL (epochs)	**538.17**	**562.83**	**594.33**	**660.17**

Table 1: Epochs required to reach moving average reward thresholds, averaged across 10 seeds with outliers (top 2, bottom 2) excluded.

DQN vs DQN with trained VRAIL applied

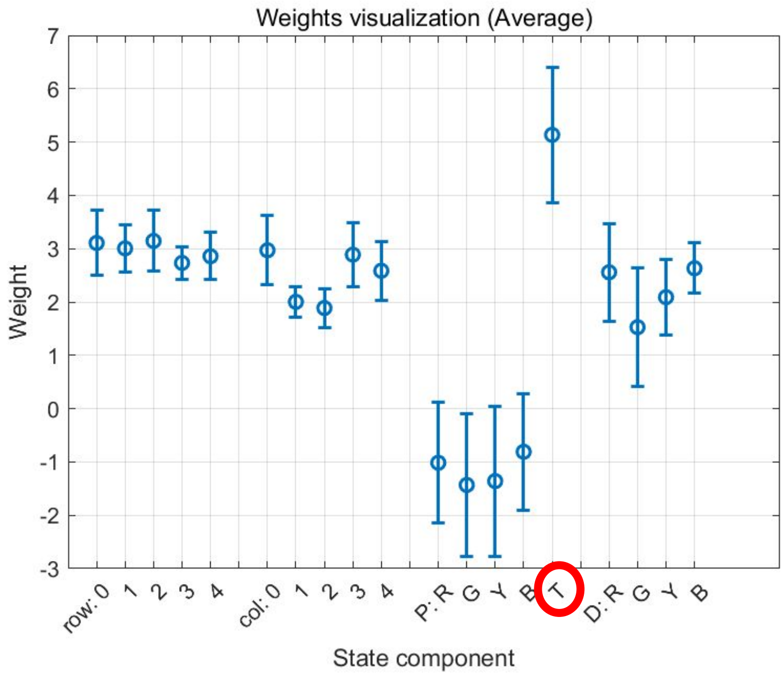
- Baseline, DQN : 4/5 converges
- DQN + reward shaping using Linear VRAIL: 5/5 converges

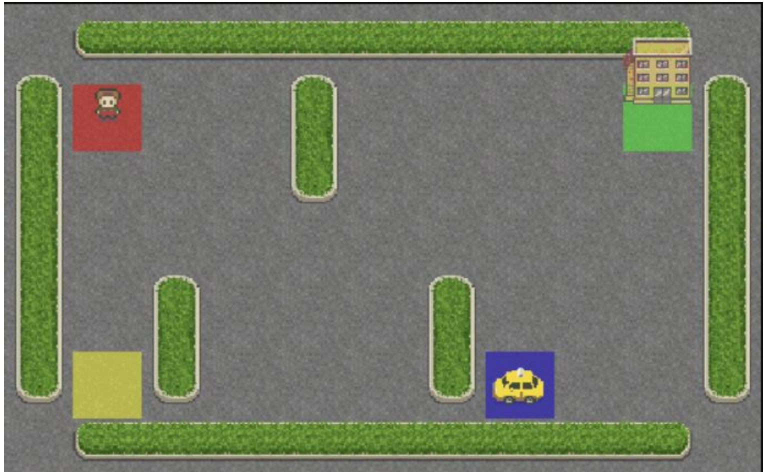




Interpretability

- Linear VRAIL ‘w’ contains the subgoal

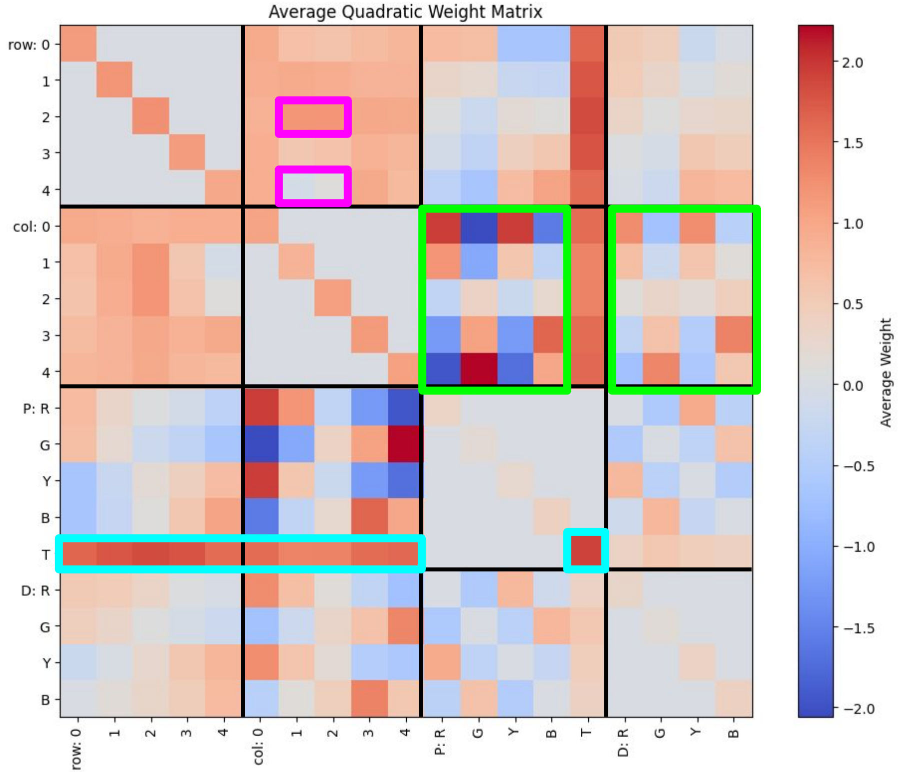


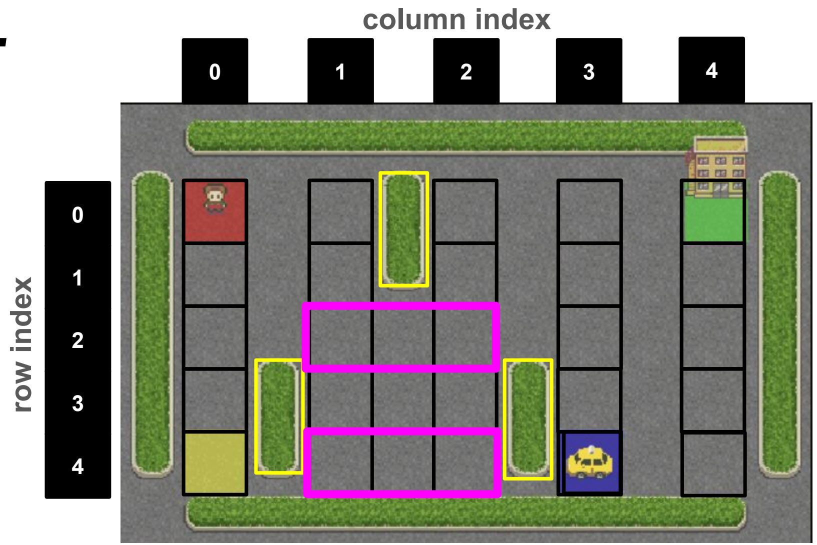


Notation

P = passenger
D = destination
P: T = passenger in taxi

Interpretability - Quadratic VRAIL





Magenta: (row, col)
[(2,1), (2,2)] : Better accessibility to P, D
[(4,1), (4,2)] : Worse accessibility

Green: P, D with column

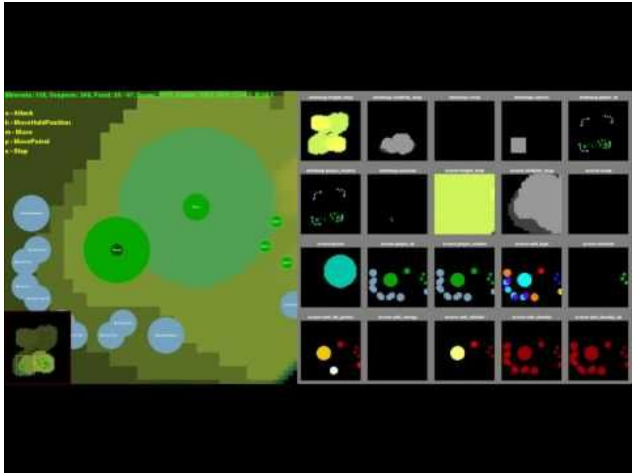
Mint: P: T is still important subgoal

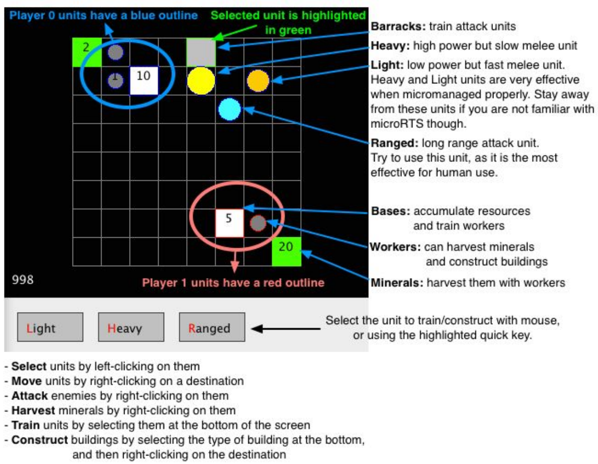
Future works

1. VRAIL could provide effective improvements in complex environments

   ex) SC2, Minecraft, MicroRTS, Crafter environment

2. Not only DQN, but also value-based RL could be used in this method
3. N-th order polynomial approximation could be more expressive





What we learned

Through this project, we learned that DQN is unstable to some environment, which often led to inconsistent results. This made us realize the **importance of reward shaping** for **stabilizing convergence**.

We also found that hyperparameter tuning was particularly challenging, especially when changing algorithmic directions, which sometimes made previous experiments less meaningful and forcing us to rethink our approach from scratch.

This iterative trial-and-error process was tough but valuable in understanding the importance of design decisions in RL.

References

[1] Ng, A., Harada, D., & Russell, S. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)* (pp. 278–287). Morgan Kaufmann.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, & Martin Riedmiller. (2013). Playing Atari with Deep Reinforcement Learning.

[3] Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, *299*, 103535. doi:10.1016/j.artint.2021.103535

[4] Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F., & Fan, C. (2020). *Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping*. arXiv preprint arXiv:2011.02669.

[5] Zou, H., Ren, T., Yan, D., Su, H., & Zhu, J. (2019). *Reward shaping via meta-learning*. arXiv preprint arXiv:1901.09330.