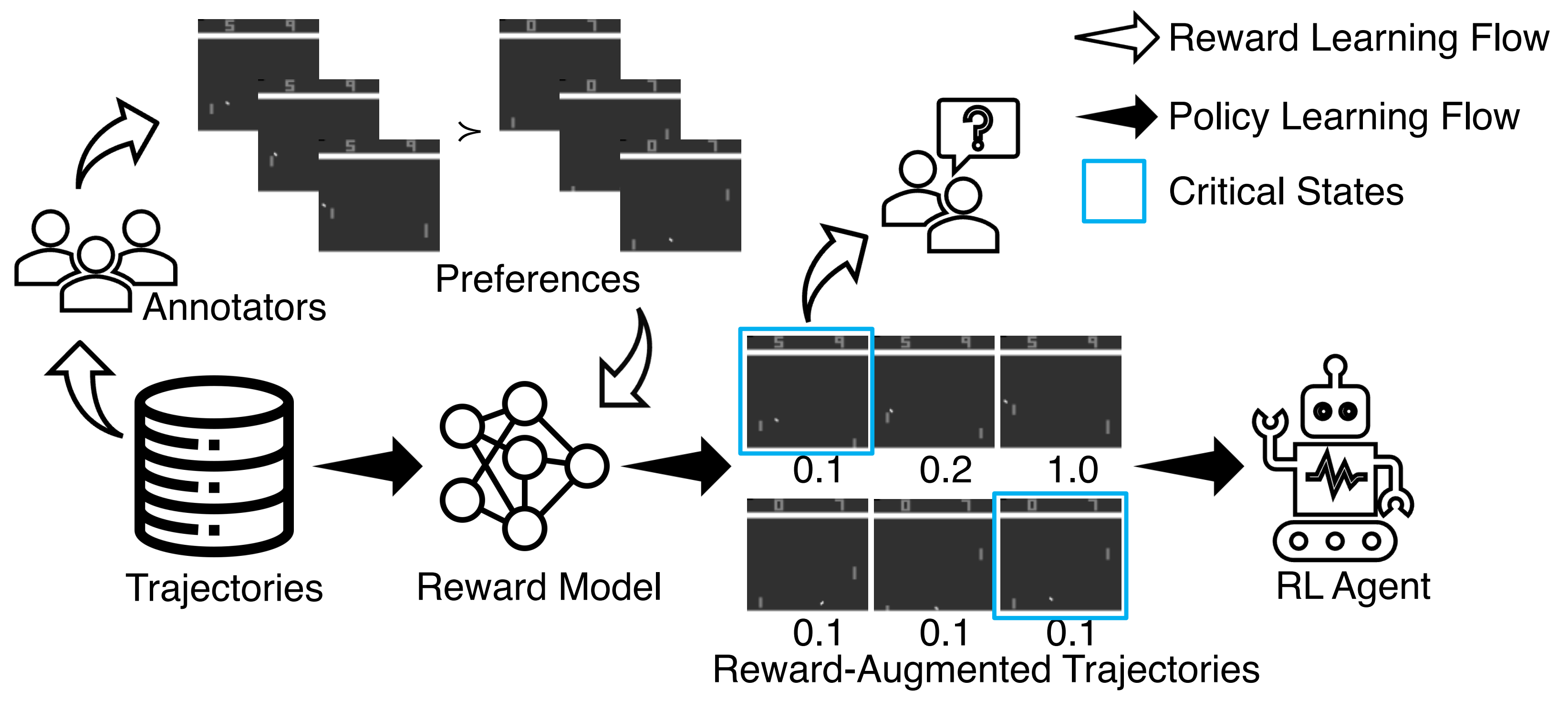
Learning State Importance for Preference-based Reinforcement Learning

Guoxi Zhang[1], Hisashi Kashima[1,2]
[1]Graduate School of Informatics, Kyoto University
[2]RIKEN Guardian Robot Project

Motivation

- Preference-based RL develops agents using human preferences.
- We argue for interpretability as a first principle. However, prior techniques can't select samples for explanation systematically.
- We propose to learn state importance and reward function together.



Approach

τ_1 and τ_2	Two trajectories being compared.
c	$c = 1$ if τ_1 is preferred over τ_2. $c = 0$ otherwise.
f_e	$\mathcal{S} \to \mathbb{R}^d$, a function that encode states into dense vectors.
f_w	$\mathbb{R}^d \to \mathbb{R}$, a function that computes state weights.
θ_R	A vector in $\mathbb{R}^d$ that is part of the reward function.

Modeling Preferences

The return of a trajectory is modeled as: $G(\tau) = \theta_R^T \sum_{s\in\tau} f_w\big(f_e(s)\big) f_e(s)$. The larger $|f_w\big(f_e(s)\big)|$ is, the more important this state is for $G(\tau)$. Assume a probabilistic model for preferences:

$$\mathrm{P}(c = 1; \tau_1, \tau_2) = \frac{\exp(G(\tau_1))}{\exp\big(G(\tau_1)\big)+\exp\big(G(\tau_2)\big)}.$$

Learning Rewards and State Weights

We propose to use $L = L_{\mathrm{ce}} + \lambda_1 L_1 + \lambda_2 L_2$ as objective function.

$$L_{\mathrm{ce}} = -\mathbb{E}[c \log \mathrm{P}(c = 1; \tau_1, \tau_2) + (1 - c)\mathrm{P}(c = 0; \tau_1, \tau_2)].$$

Minimization of L_{ce} leads to a reward model that explains preferences. We make the following two assumptions for state weights: (a) only few states are critical for preferences, and (b) the critical states span multiple time steps. Thus,

$$L_1 = \mathbb{E}[|f_w(f_e(s))|],$$
$$L_2 = \mathbb{E}[(f_w\big(f_e(s_t)\big) - f_w(f_e(s_{t+1})))^2].$$

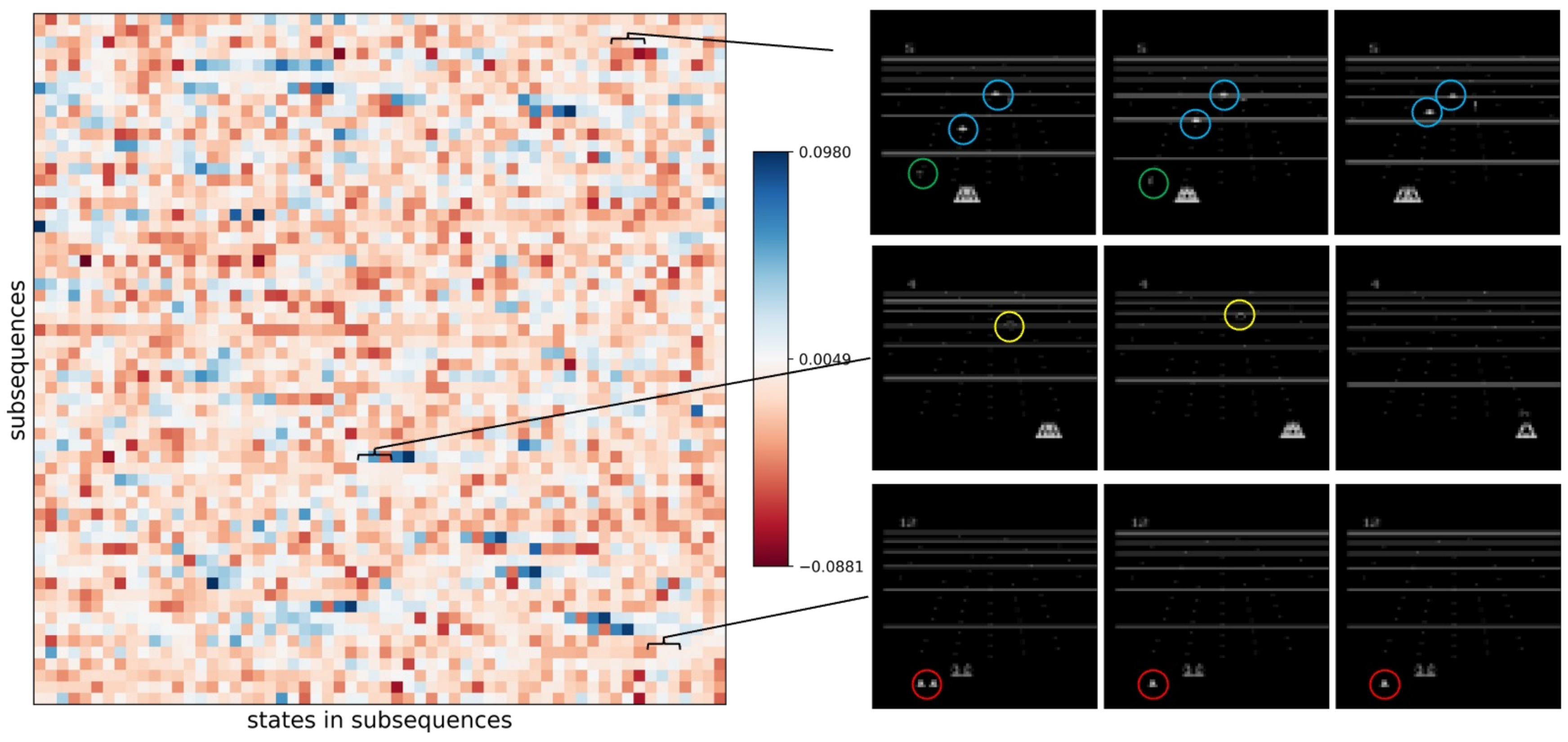
θ_R, f_e, and f_w are to be learned from data. After learning, we compute the reward of a new state as: $R(s) = \theta_R^T f_w\big(f_e(s)\big) f_e(s)$

Evaluating Performance and State Weights

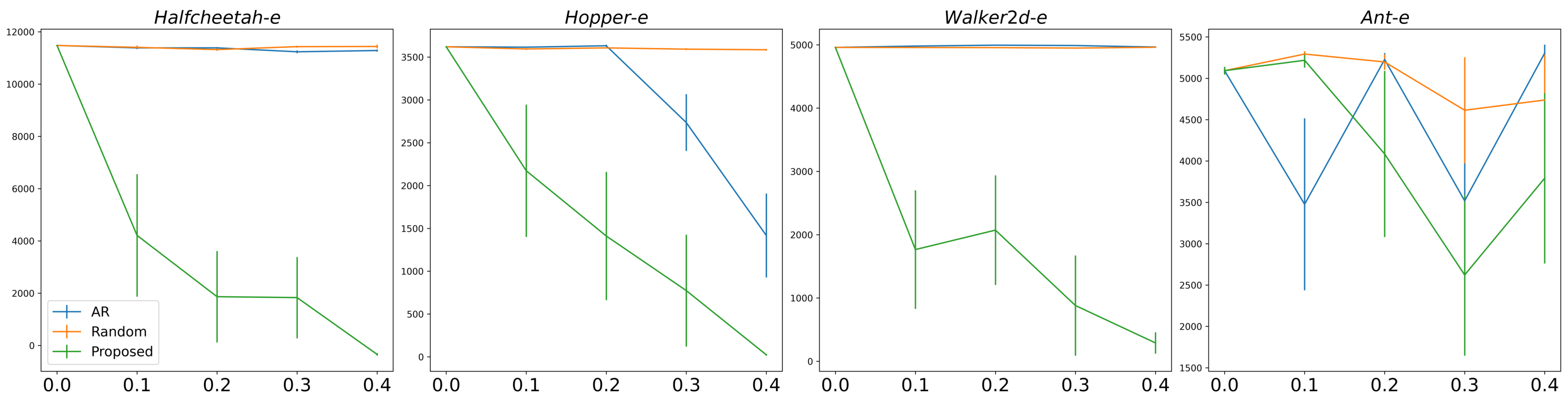
Performance on 17 datasets. The proposed method outperforms BT on nine of the datasets. On four of the rest, it has similar performance with BT. These results show that performance is not sacrificed for interpretability.

Task	Proposed	BT	T-REX
BeamRider	4438.18 ± 796.92	**4681.45 ± 949.61**	305.00 ± 128.74
Enduro	**915.13 ± 167.41**	628.44 ± 135.80	56.94 ± 31.72
Hero	**4397.80 ± 2046.37**	1043.79 ± 543.10	770.22 ± 421.79
Pong	**−19.86 ± 0.71**	−19.89 ± 1.03	−20.88 ± 0.04
Seaquest	107.78 ± 49.38	**155.82 ± 33.57**	51.32 ± 26.33
Alien	164.14 ± 67.37	166.06 ± 33.11	**317.13 ± 77.25**
Boxing	**21.06 ± 13.77**	−9.08 ± 4.51	−8.25 ± 2.67
Assault	**133.34 ± 57.32**	84.10 ± 44.49	126.91 ± 89.24
BattleZone	**4727.37 ± 1733.43**	4175.15 ± 859.19	4194.85 ± 1655.48
Hopper-m	1589.10 ± 126.70	1729.52 ± 17.42	**1731.06 ± 40.98**
Hopper-e	**3604.00 ± 8.27**	3515.32 ± 38.06	3577.56 ± 25.57
Walker2d-m	3404.37 ± 54.85	**3426.19 ± 59.82**	3270.97 ± 41.99
Walker2d-e	4963.82 ± 6.12	**4971.17 ± 7.71**	3977.50 ± 892.78
Ant-m	3110.78 ± 115.75	3287.12 ± 111.55	**3441.29 ± 169.01**
Ant-e	**5208.21 ± 103.82**	4951.52 ± 172.17	4941.51 ± 155.29
Halfcheetah-m	**5374.33 ± 29.16**	5123.57 ± 17.28	5247.13 ± 11.15
Halfcheetah-e	11299.84 ± 148.04	11252.16 ± 176.39	**11492.95 ± 21.43**

Evaluating State Weights and Performance



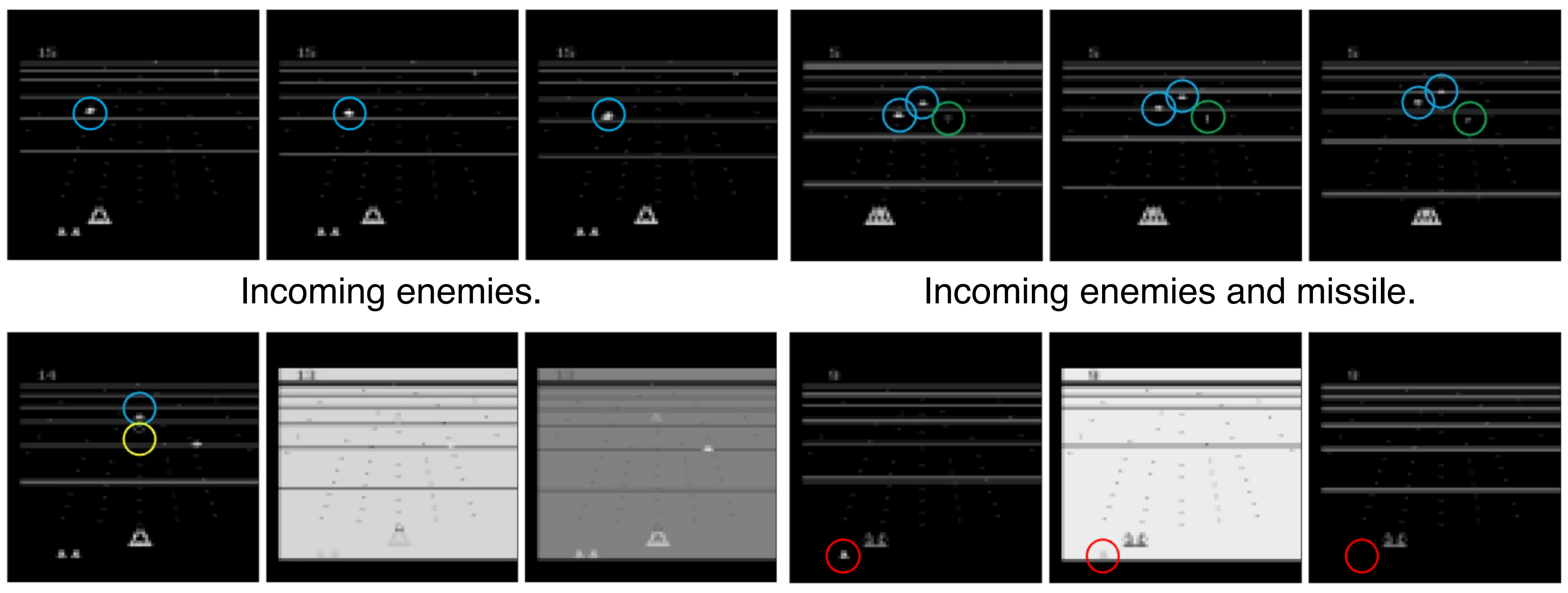
Left: A heatmap for state weights on Atari game BeamRider. Upper right: an example for states with large absolute weights. The agent was closed to an incoming missile (circled in green) in the presence of enemies (circled in blue). Middle right: an example for states with weights close to zero. The agent launched missiles (circled in yellow) in open space. Bottom right: an example for transitioning from states with large absolute weights to states with small absolute weights, in which the agent lost a life (circled in red).



Cumulative returns as we remove samples according to state importance. X-axis: ratio of samples removed. Y-axis: cumulative returns. Performance worsens as we remove samples according to learned state importance. This confirms that the weights characterize how critical states are for the corresponding task.

Inspecting Reward Models using State Weights

Correctly Identified States



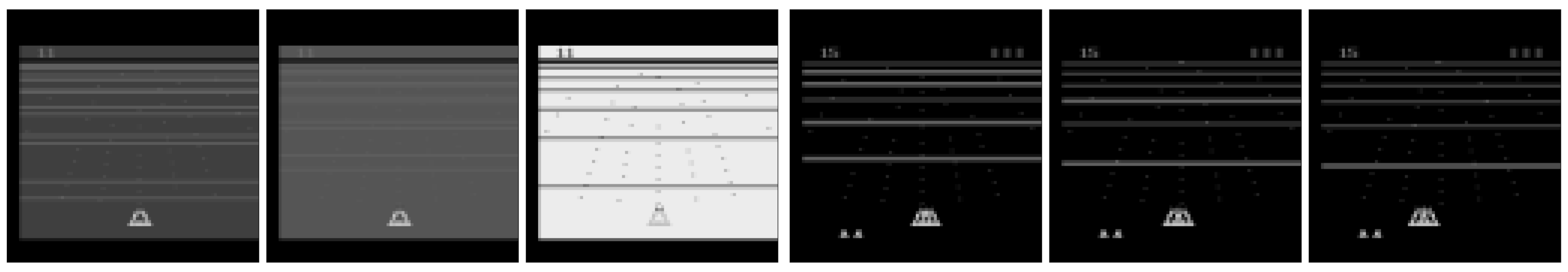
Incoming enemies.

Incoming enemies and missile.

Destroying an enemy.

Losing a life.

Incorrectly Identified States



Screen flickering.

Flying in open space.

Conclusion

- We propose a method for systematically selecting samples to interpret a reward model learned from human preferences.
- We confirm that the learned weights indeed characterize state importance.
- With the proposed model, we obtain insights for reward models learned from preferences.

Contact & Acknowledgement

Guoxi Zhang guoxi@ml.ist.i.kyoto-u.ac.jp
Hisashi Kashima kashima@i.kyoto-u.ac.jp

This work was supported by JST CREST (JPMJCR21D1).