A Test Oracle for Reinforcement Learning Software based on Lyapunov Stability Control Theory


Shiyu Zhang


Haoyang Song


Qixin Wang*


Henghua Shen

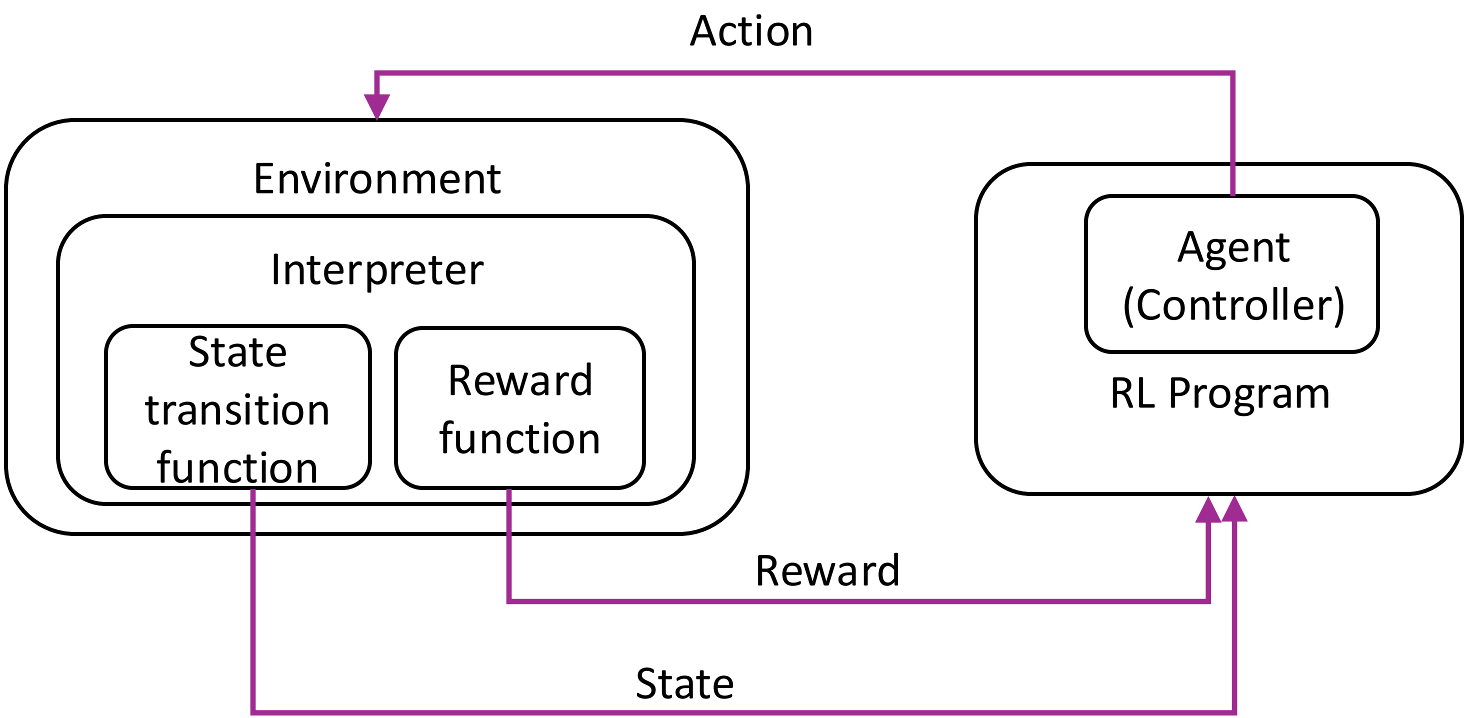

Yu Pei

Department of Computing

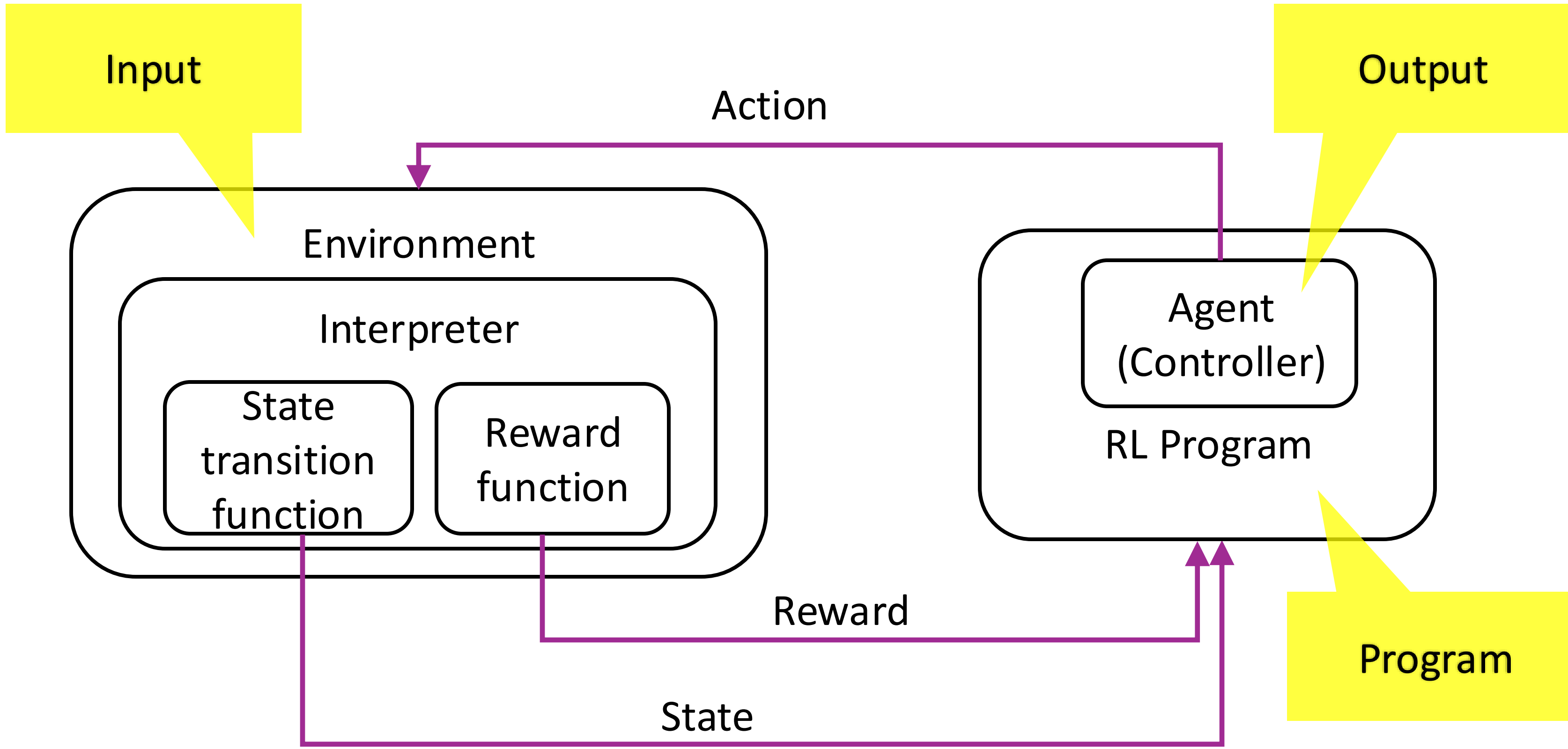
The Hong Kong Polytechnic University

April 30, 2025

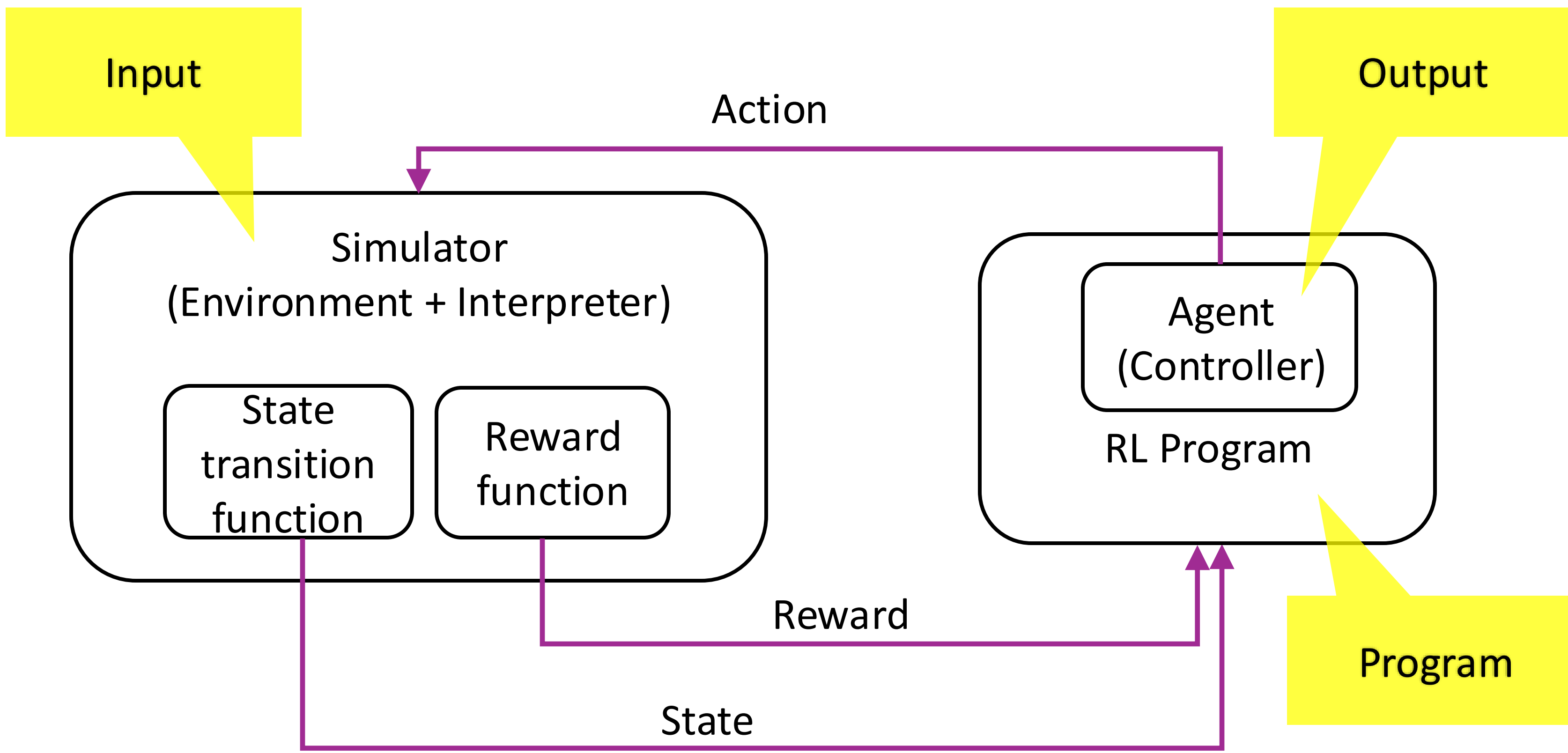
Reinforcement Learning (RL) becomes increasingly important due to its ability to train controllers for complex systems/robots.



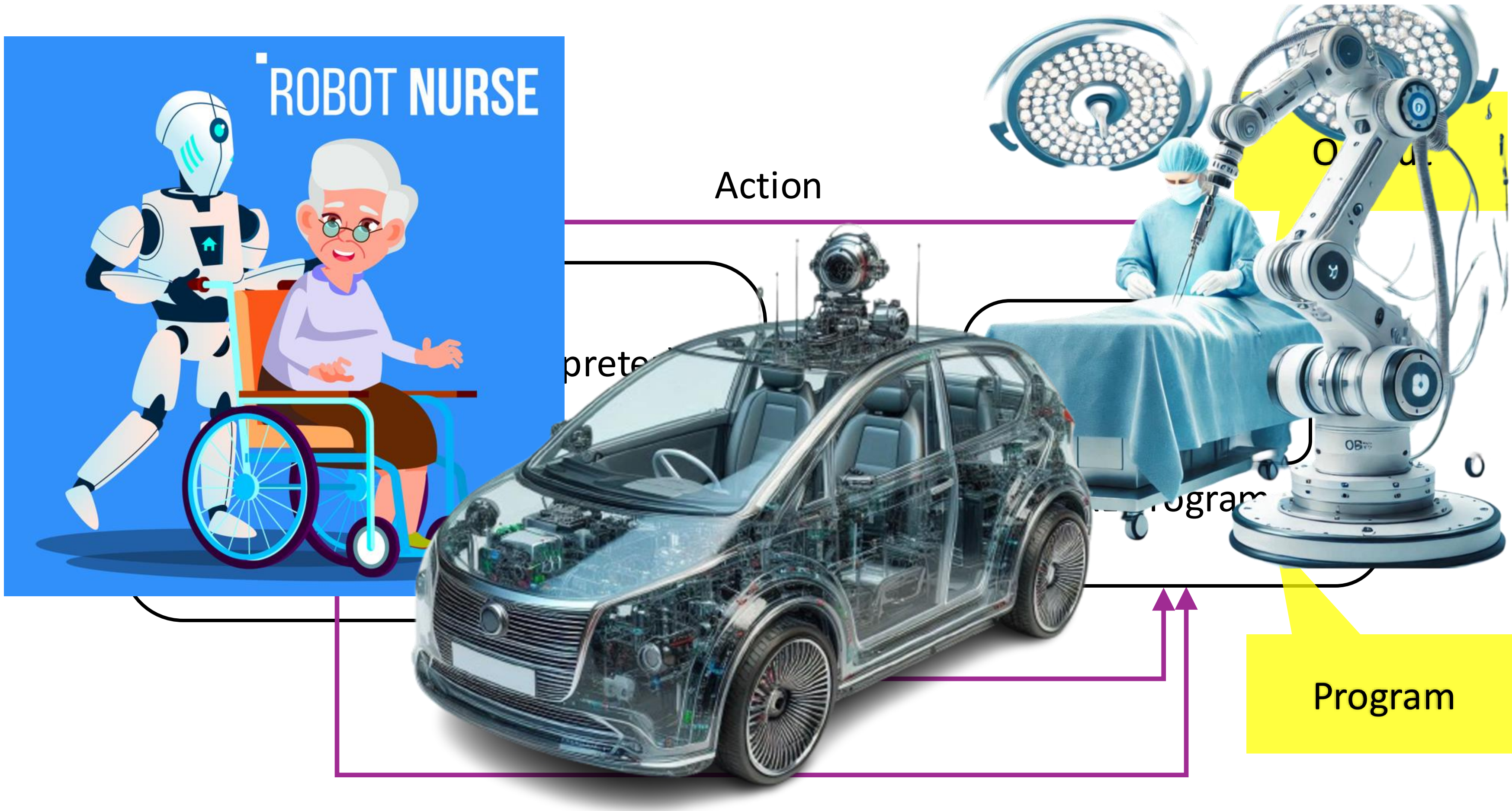
Reinforcement Learning (RL) becomes increasingly important due to its ability to train controllers for complex systems/robots.



Reinforcement Learning (RL) becomes increasingly important due to its ability to train controllers for complex systems/robots.

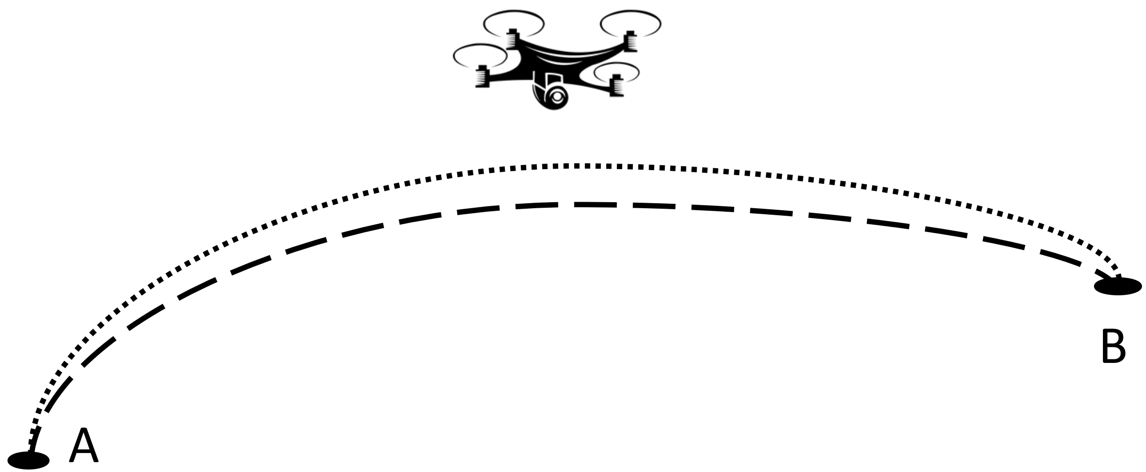


As RL generates controllers for critical control/robotic applications, ensuring RL program correctness is important.



Oracle Problem for RL: how to judge if an RL program is buggy (in other words, its generated controller is wrong)?

The correct controller to a system/robot is not unique.



Less overshoot? Or less time cost? Or less jitter?

Solution Heuristics

Create many non-trivial controller design problems with well-known analytical/numerical solution and solution features.

If the RL generated controllers most often agree with the well-known analytical/numerical solution features, then we label the RL program correct, and otherwise buggy.

Lyapunov stable controller design theory

Given: $\dot{X} = AX + BU$, where $X \in \mathbb{R}^n$, $U \in \mathbb{R}^m$

State of the Plant

Control

Lyapunov stable controller design theory

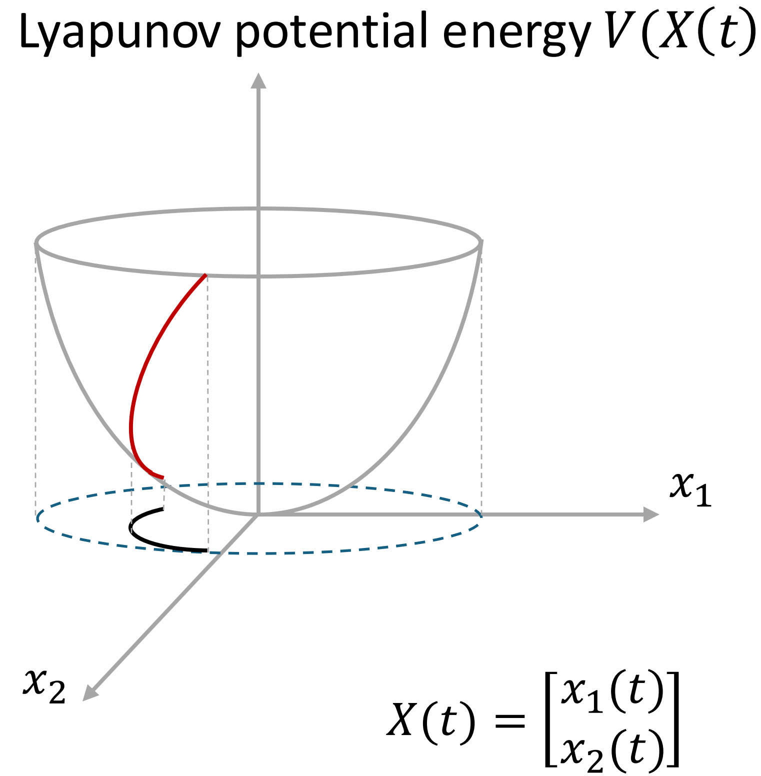
Given: $\dot{X} = AX + BU$, where $X \in \mathbb{R}^n$, $U \in \mathbb{R}^m$

Demand: $U = -KX$

Lyapunov stable controller design theory

Given: $\dot{X} = AX + BU$

Demand: $U = -KX$



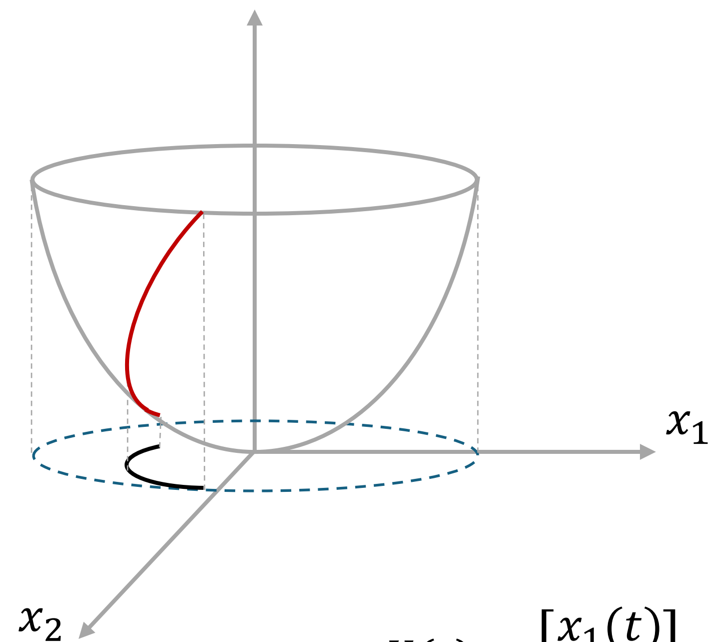
Lyapunov stable controller design theory

Given: $\dot{X} = AX + BU$

Demand: $U = -KX$

Well-known numerical solution exists.

Lyapunov potential energy $V(X(t))$



$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

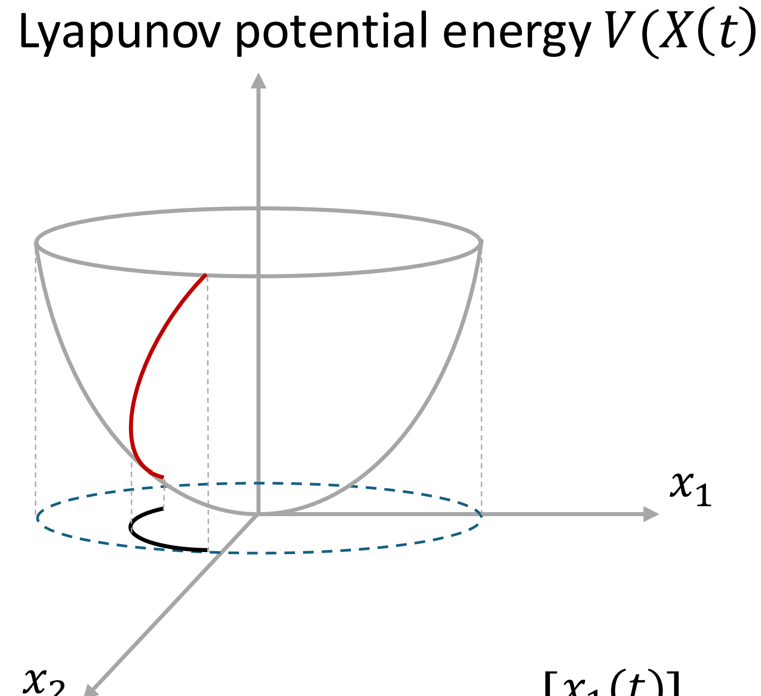
Lyapunov stable controller design theory

Given: $\dot{X} = AX + BU$

Demand: $U = -KX$

Well-known numerical solution exists.

Solution feature: the so-designed controller U always decreases the plant's Lyapunov potential energy $V(X(t))$ over time.



$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Create many non-trivial Lyapunov stable controller design problems.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

Create many non-trivial Lyapunov stable controller design problems.

Prepare the ground truth Lyapunov stable controller U and Lyapunov potential energy $V(X(t))$ as per the conventional numerical solution.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

Create many non-trivial Lyapunov stable controller design problems.

Prepare the ground truth Lyapunov stable controller U and Lyapunov potential energy $V(X(t))$ as per the conventional numerical solution.

During RL Training: Use the ground truth Lyapunov potential energy $V(X(t))$ decreasing feature to guide the reward generation.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

Create many non-trivial Lyapunov stable controller design problems.

Prepare the ground truth Lyapunov stable controller U and Lyapunov potential energy $V(X(t))$ as per the conventional numerical solution.

During RL Training: Use the ground truth Lyapunov potential energy $V(X(t))$ decreasing feature to guide the reward generation.

During Testing Stage: Check if the RL generated controller complies with the ground truth Lyapunov potential energy decreasing feature.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

During Testing Stage: Check if the RL generated controller complies with the ground truth Lyapunov potential energy decreasing feature.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

During Testing Stage: Check if the RL generated controller complies with the ground truth Lyapunov potential energy decreasing feature.

An RL generated controller is complying iff over ϑ percentage of its tested control steps are Lyapunov potential energy decreasing.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

Given: $\dot{X} = AX + BU$

Prepare: $U = -KX$

During Testing Stage: Check if the RL generated controller complies with the ground truth Lyapunov potential energy decreasing feature.

An RL generated controller is complying iff over ϑ percentage of its tested control steps are Lyapunov potential energy decreasing.

The RL is considered bugless iff over θ percentage of its generated controllers are complying.

Solution Heuristics: a correct RL program can figure out the hidden Lyapunov potential energy concept and the energy-decrease feature.

$$\text{Given: } \dot{X} = AX + BU$$

$$\text{Prepare: } U = -KX$$

During Testing Stage: Check if the RL generated controller complies with the ground truth Lyapunov potential energy decreasing feature.

An RL generated controller is complying iff over ϑ percentage of its tested control steps are Lyapunov potential energy decreasing.

The RL is considered bugless iff over θ percentage of its generated controllers are complying.

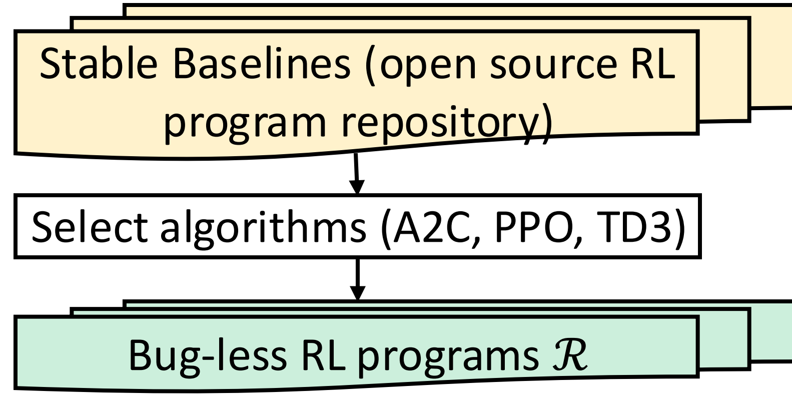
Lyapunov Potential Energy Abnormality Oracle: LPEA(ϑ, θ).

Evaluation: workflow

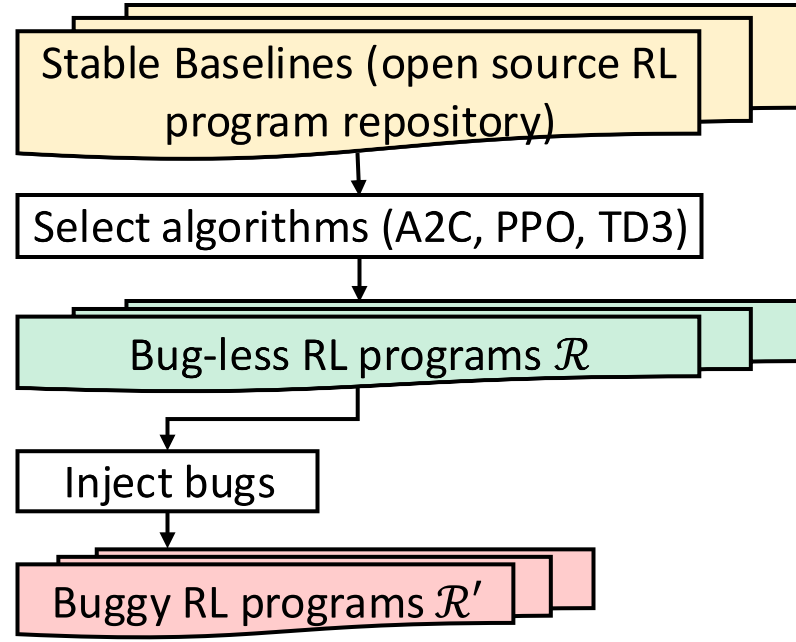
Evaluation: workflow

Stable Baselines (open source RL program repository)

Evaluation: workflow



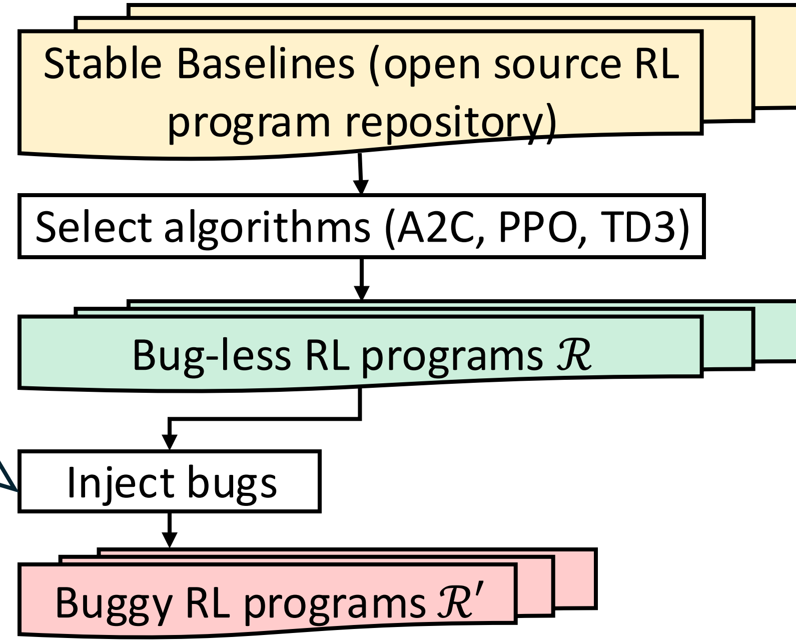
Evaluation: workflow



Evaluation: workflow

TABLE I: RL Software Bug Types from Survey [19]

Level-1 Type	Level-2 Type
1. RL Specific	1.1. Exploring the environment bugs
	1.2. Updating network bugs
2. NN Specific	2.1. Model bugs
	2.2. Training bugs
	2.3. Tensor and inputs bugs



Evaluation: workflow

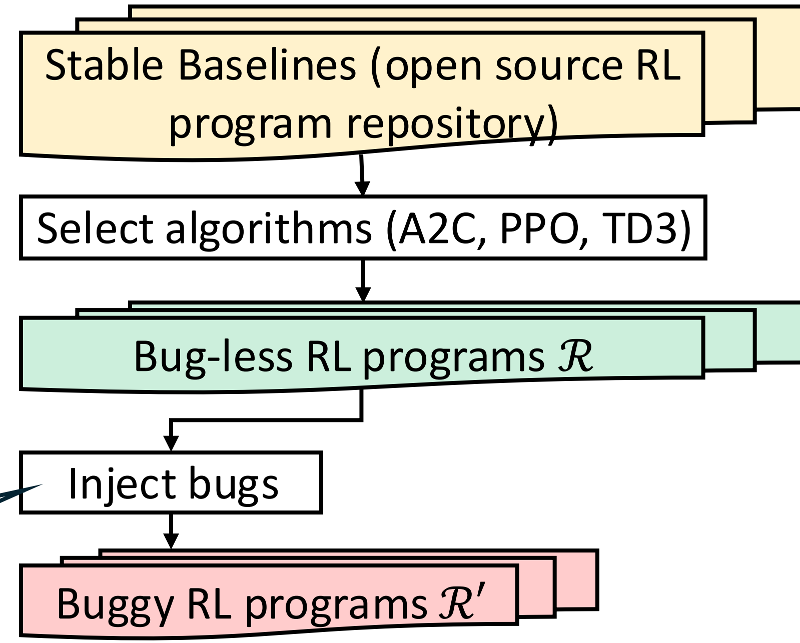
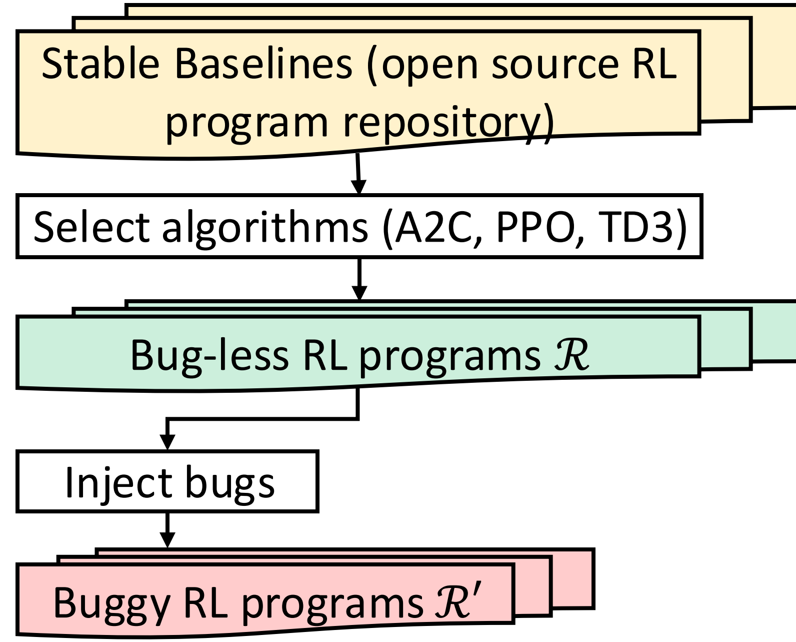


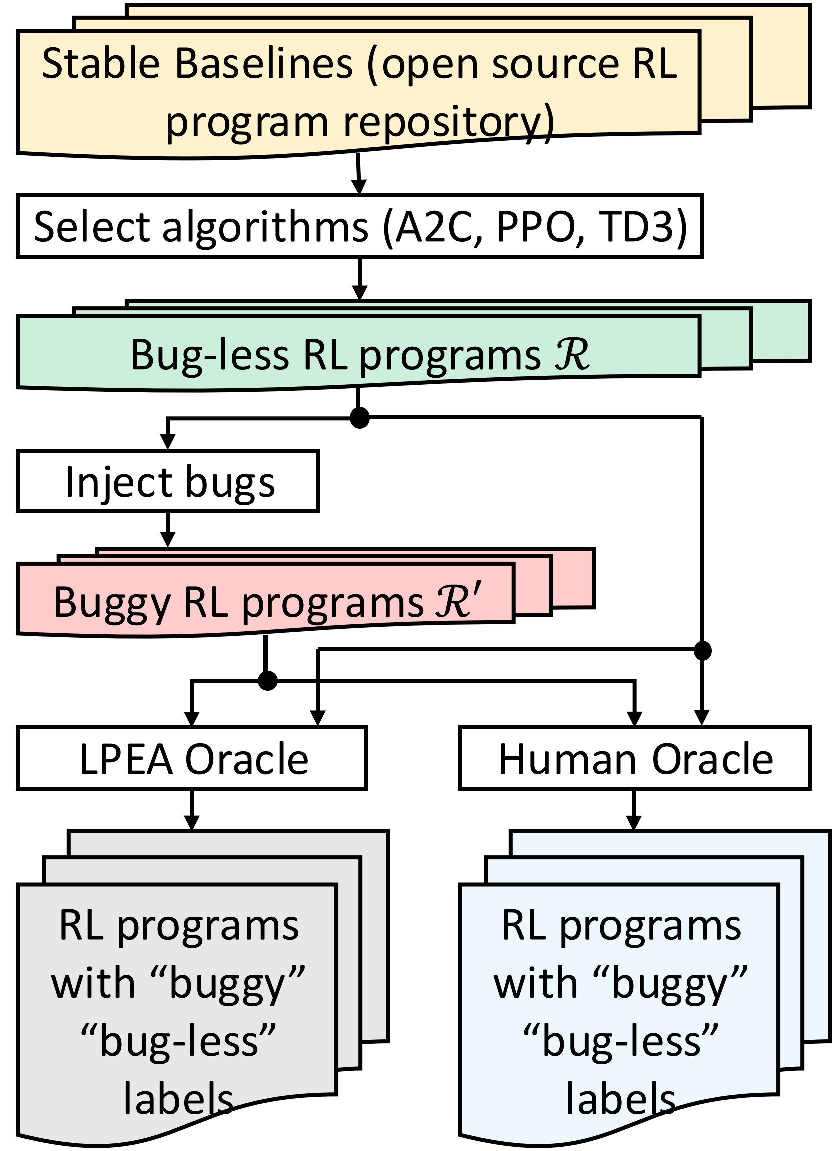
TABLE III: Real-World Bugs

Bug ID	Description
RW1	SB3 GitHub commit hash: `d5986da` File: `stable_baselines3/common/on_policy_algorithm.py` Line 167: “`dones`” should be “`_last_dones`”; Line 178: should insert one more line “`self._last_dones = dones`” before this line.
RW2	SB GitHub commit hash: `689afd1` File: `stable_baselines/a2c/a2c.py` Line 325: “`np.int32`” should be “`self.env.action_space.dtype`”.

Evaluation: workflow



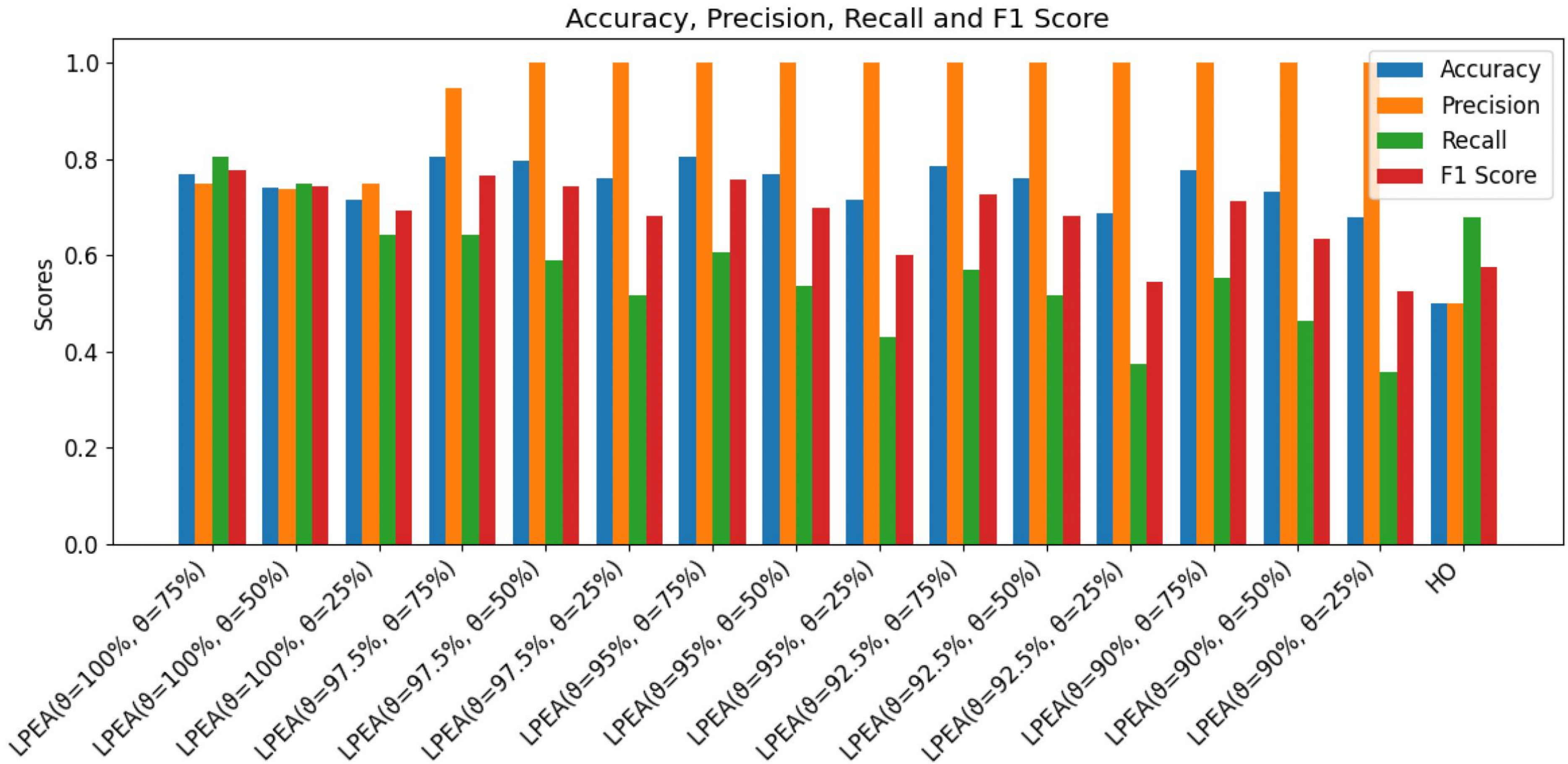
Evaluation: workflow



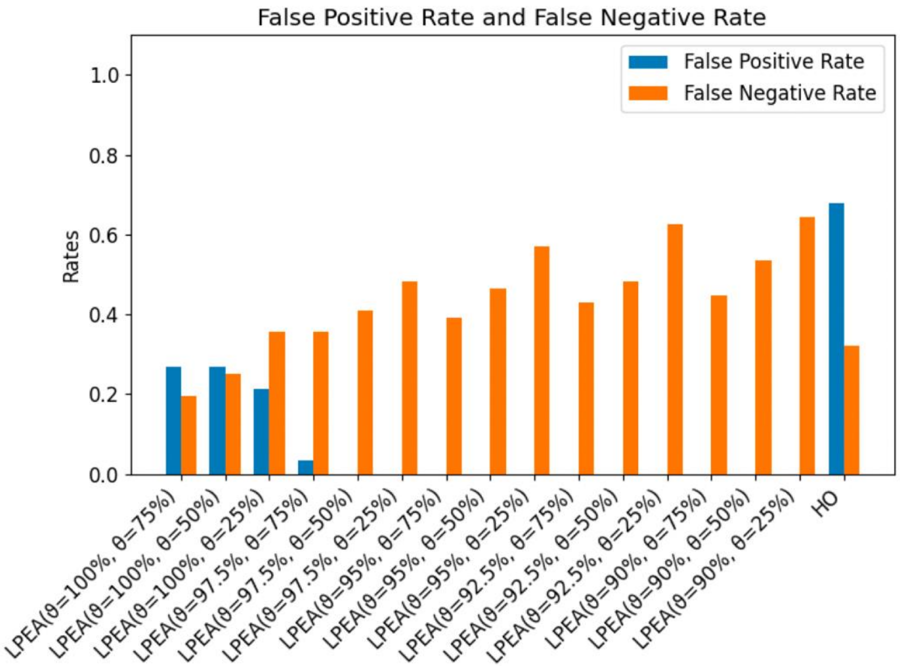
Evaluation: raw results

Oracle	A2C				PPO				TD3				Overall (A2C, PPO, and TD3)			
	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN
LPEA($\vartheta = 100\%, \theta = 75\%$)	15	0	19	4	15	0	22	7	15	15	0	0	45	15	41	11
LPEA($\vartheta = 100\%, \theta = 50\%$)	15	0	19	4	12	0	22	10	15	15	0	0	42	15	41	14
LPEA($\vartheta = 100\%, \theta = 25\%$)	13	0	19	6	9	0	22	13	14	12	3	1	36	12	44	20
LPEA($\vartheta = 97.5\%, \theta = 75\%$)	15	0	19	4	9	0	22	13	12	2	13	3	36	2	54	20
LPEA($\vartheta = 97.5\%, \theta = 50\%$)	15	0	19	4	8	0	22	14	10	0	15	5	33	0	56	23
LPEA($\vartheta = 97.5\%, \theta = 25\%$)	13	0	19	6	6	0	22	16	10	0	15	5	29	0	56	27
LPEA($\vartheta = 95\%, \theta = 75\%$)	15	0	19	4	8	0	22	14	11	0	15	4	34	0	56	22
LPEA($\vartheta = 95\%, \theta = 50\%$)	13	0	19	6	7	0	22	15	10	0	15	5	30	0	56	26
LPEA($\vartheta = 95\%, \theta = 25\%$)	11	0	19	8	5	0	22	17	8	0	15	7	24	0	56	32
LPEA($\vartheta = 92.5\%, \theta = 75\%$)	13	0	19	6	8	0	22	14	11	0	15	4	32	0	56	24
LPEA($\vartheta = 92.5\%, \theta = 50\%$)	13	0	19	6	6	0	22	16	10	0	15	5	29	0	56	27
LPEA($\vartheta = 92.5\%, \theta = 25\%$)	9	0	19	10	5	0	22	17	7	0	15	8	21	0	56	35
LPEA($\vartheta = 90\%, \theta = 75\%$)	13	0	19	6	8	0	22	14	10	0	15	5	31	0	56	25
LPEA($\vartheta = 90\%, \theta = 50\%$)	13	0	19	6	5	0	22	17	8	0	15	7	26	0	56	30
LPEA($\vartheta = 90\%, \theta = 25\%$)	8	0	19	11	5	0	22	17	7	0	15	8	20	0	56	36
Human Oracle	12	15	4	7	15	15	7	7	11	8	7	4	38	38	18	18

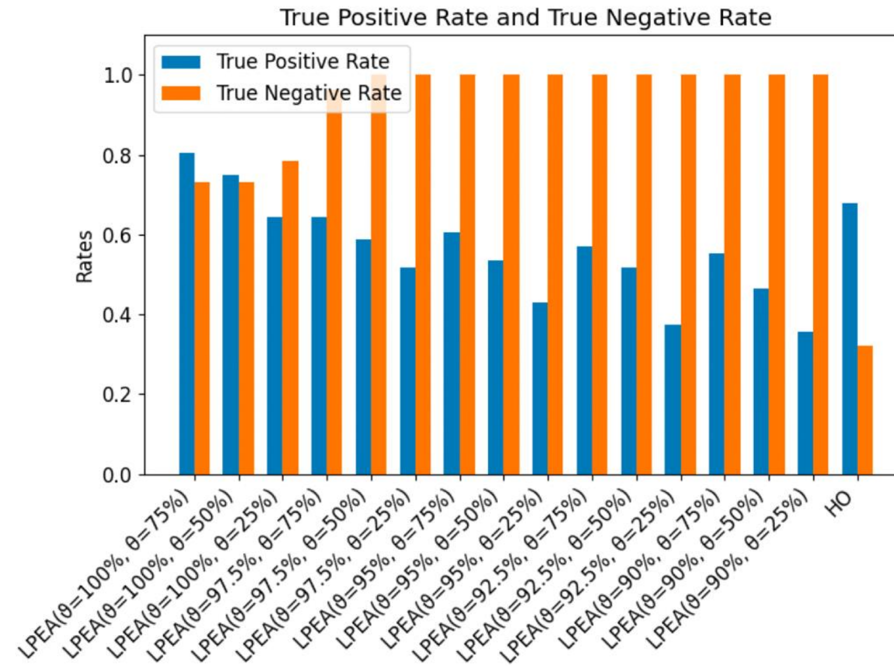
Evaluation: results



Evaluation: results

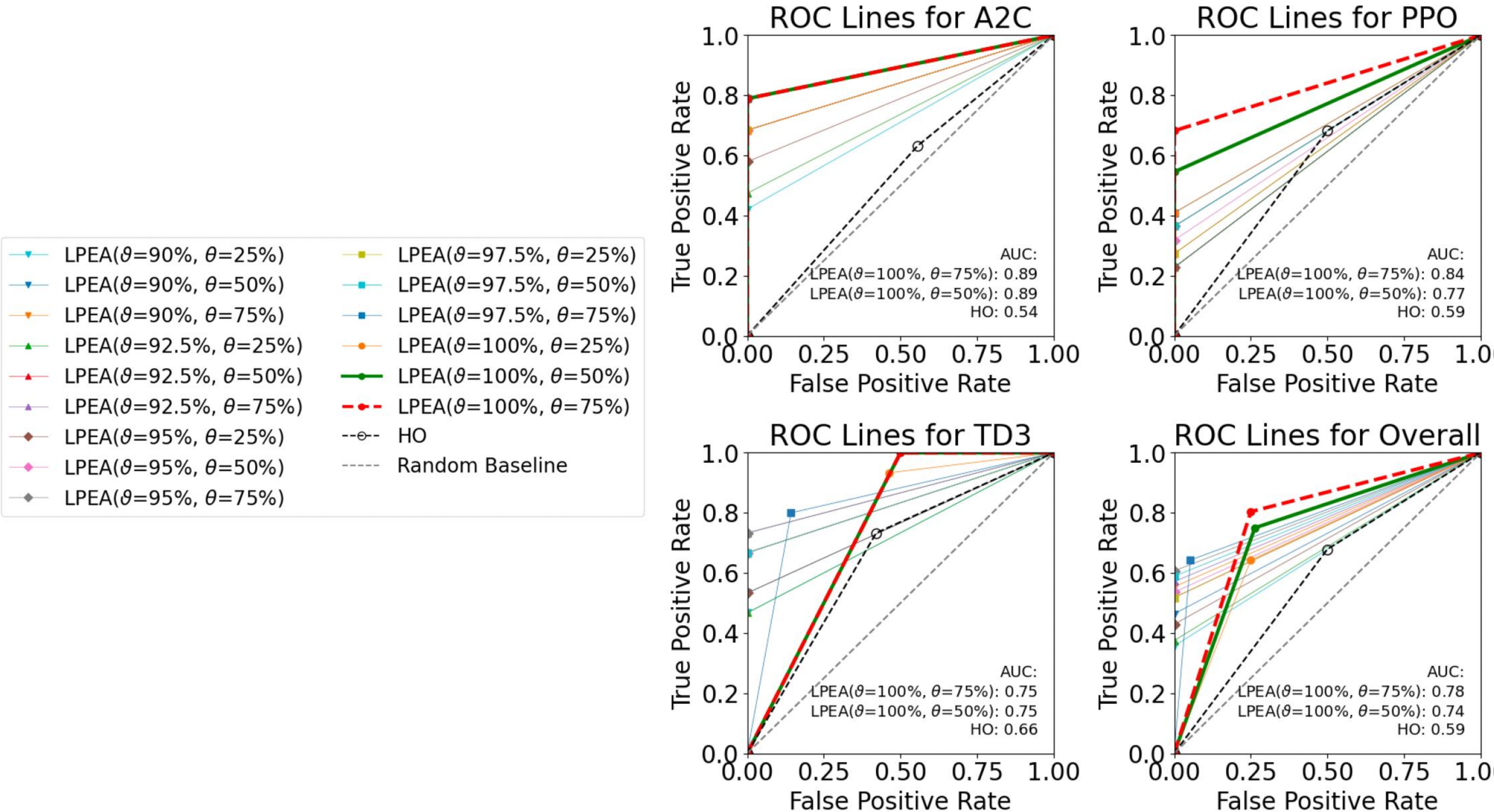


(b) False Positive Rate and False Negative Rate; lower values are better. HO: Human Oracle.



(c) True Positive Rate and True Negative Rate; higher values are better. HO: Human Oracle

⚖ Evaluation: results



Related Work

Metamorphic testing: LPEA oracle is a metamorphic testing oracle for RL, which is neither supervised learning or unsupervised learning (Xie et al. [26][27]).

Pang et al. [28] and Tappler et al. [29]: assume an obvious faulty state exists to define the control crashed.

Padgham et al. [30] needs white-box access to the design and implementation of the agents.

Nikanjam et al. [19] (DRLinter): static analysis upon RL program.

Varshosaz et al. [31]: needs white-box formal model.

LPEA oracle can be integrated into frameworks of property based testing, such as QuickCheck [32].

We can also use the formal language proposed by Jothimurugan et al. [33] to specify RL algorithms.

Shen et al. [34] and Hu et al. [35]: focus on evaluating test set quality; needs white-box controller.

Wan et al. [37]: focuses on improving the quality of test sets.

Lacoste et al. [38]: focuses on environmental costs of machine learning

Conclusion

LPEA oracles outperform the human oracle in most of the metrics.

Particularly, LPEA($\vartheta = 100\%, \theta = 75\%$) outperforms the human oracle by 53.6% in accuracy, 50% in precision, 18.4% in recall, 34.8% in F1 score, 18.4% in true positive rate, 127.8% in true negative rate, 60.5% in false positive rate, 38.9% in false negative rate, and 31.7% in ROC curve AUC.

Conclusion

LPEA oracles outperform the human oracle in most of the metrics.

Particularly, LPEA($\vartheta = 100\%, \theta = 75\%$) outperforms the human oracle by 53.6% in accuracy, 50% in precision, 18.4% in recall, 34.8% in F1 score, 18.4% in true positive rate, 127.8% in true negative rate, 60.5% in false positive rate, 38.9% in false negative rate, and 31.7% in ROC curve AUC.

Thank You!