Generative Model-Based Testing on Decision-Making Policies

Lei HAN

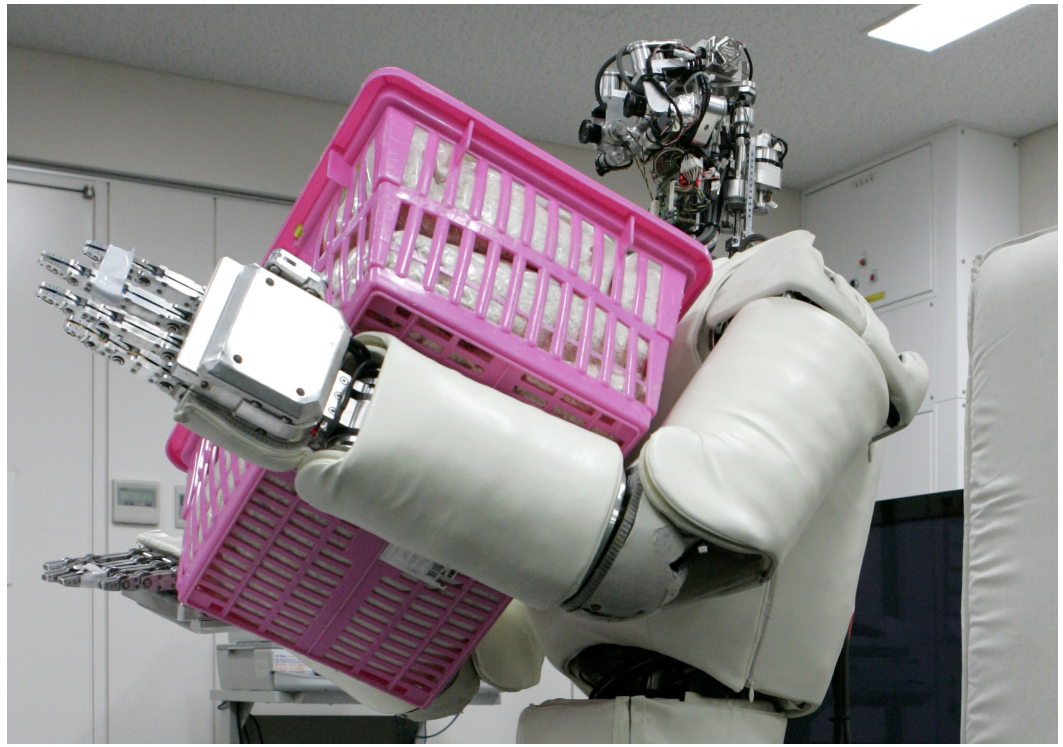
Table content

- Decision-Making Policy
- Challenges in Testing Decision-Making Policies
- Motivation on Testing Decision-Making Policies
- Proposed Method: diffusion model + novelty-based guidance
- Experiment: Research Questions
- Q & A

Decision-making devices

Nearly 40% of the time spent on housework could be automated within a decade

University of Oxford - Oxford Internet Institute 2023

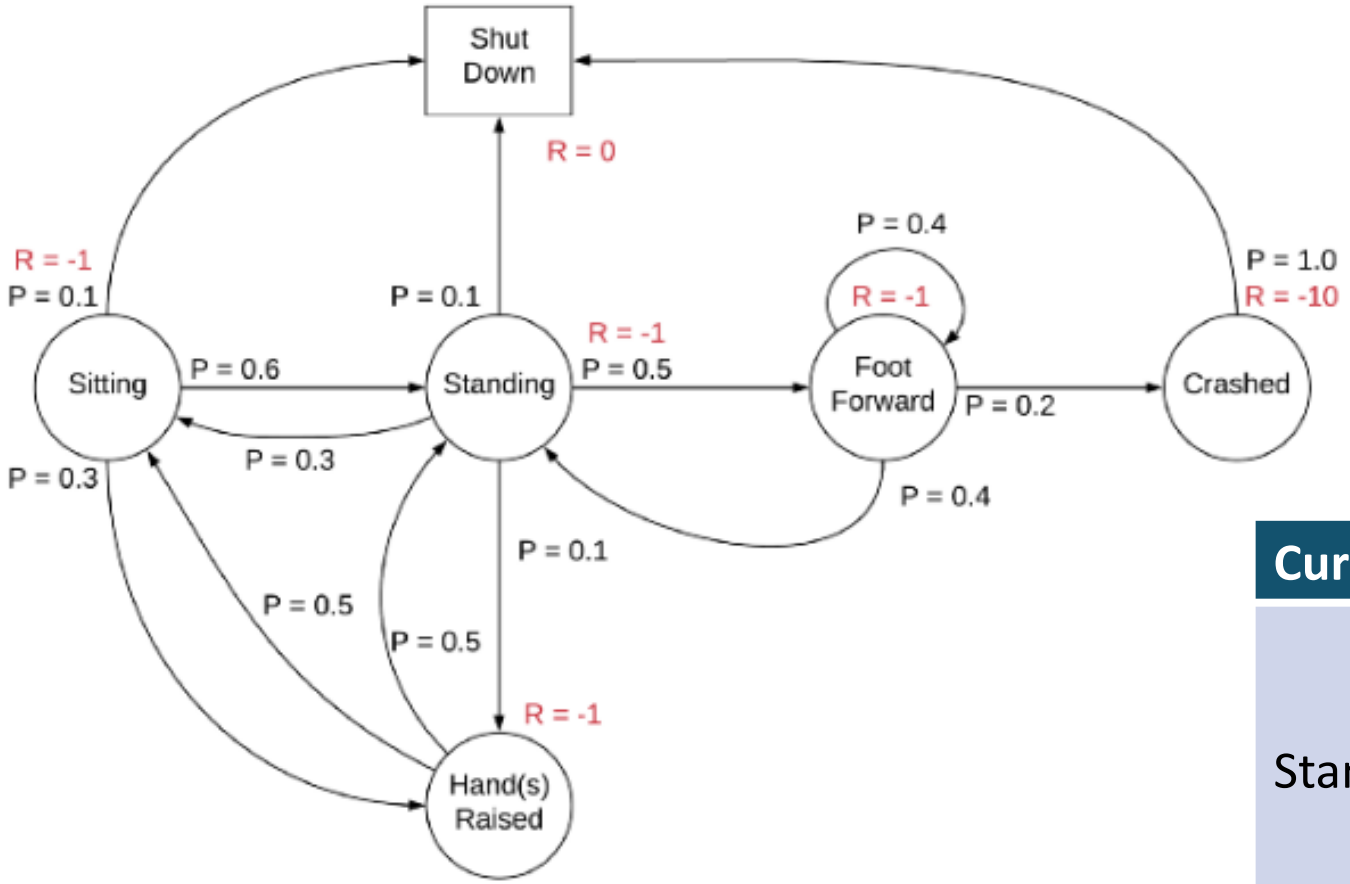


https://doi.org/10.1371/journal.pone.0281282

Task	5-year prediction (std.)	10-year prediction (std.)
Grocery shopping	45.25 (24.69)	59.34 (23.40)
Shopping (other than groceries)	39.17 (21.28)	50.23 (22.71)
Using services	36.86 (16.96)	51.63 (20.42)
Household cleaning	33.55 (19.81)	46.18 (23.28)
Dish washing	33.29 (24.91)	46.89 (26.32)
Cooking	32.48 (20.22)	46.05 (23.06)
Ironing and folding	30.68 (23.67)	43.52 (25.72)
Laundry	28.65 (21.89)	42.95 (23.82)
Teaching a child	27.40 (18.27)	39.63 (22.19)
Gardening	27.17 (20.20)	39.63 (21.72)
Household and car maintenance	24.89 (19.48)	36.11 (21.55)
Caring for an adult	23.77 (14.57)	34.77 (17.56)
Making and mending clothes	21.52 (23.87)	29.06 (24.89)
Pet care	21.09 (15.64)	31.68 (18.63)
Interacting with a child	14.72 (13.90)	22.25 (17.67)
Escorting a child outside the home	14.09 (11.66)	23.55 (15.13)
Physical childcare	12.32 (8.42)	20.77 (12.90)
Mean	**27.47 (21.12)**	**39.07 (23.89)**

Markov decision process (MDP)

(**S**tate, **A**ction, **P**robability, **R**eward, **γ**=Discount factor)



A sample Markov chain for the robot example

$$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$$

$$\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$$

Current state	Next state	Probability	Reward
Standing	Shut down	0.1	0
	Foot Forward	0.5	-1
	Hand Raised	0.1	-1
	Sitting	0.3	0

Decision-Making Policy

The return G_t is the total discounted reward from time-step t

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

A policy π is a distribution over actions given states,

$$\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$$

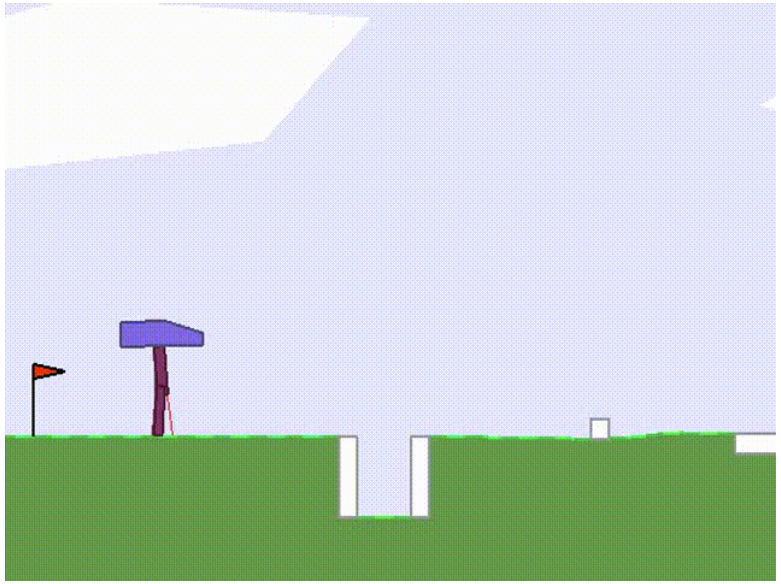
- A policy is the thought process behind picking an action.
- In practice, it's a probability distribution assigned to the set of actions. Highly rewarding actions will have a high probability and vice versa.
- If an action has a low probability, it doesn't mean it won't be picked at all. It's just less likely to be picked.

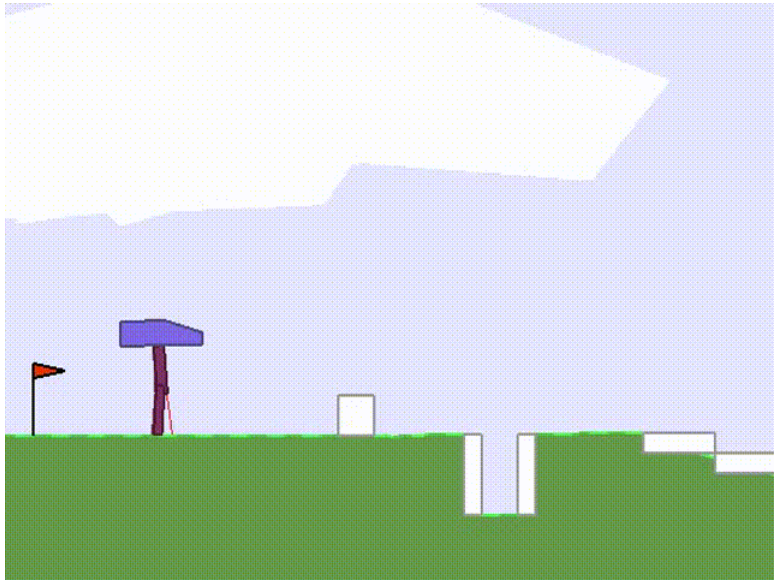
Importance of Testing Decision-Making Policies

➢DNN-based decision-making policies require more effective test

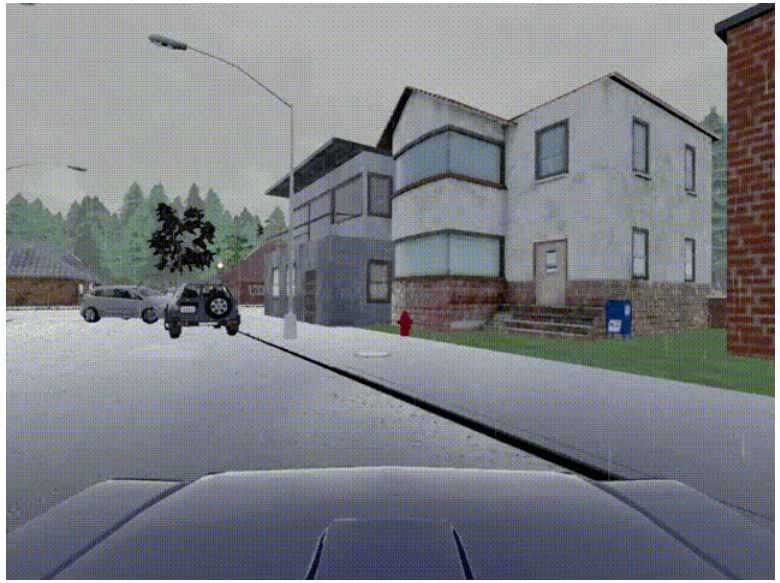
- ※ Methods for solving decision-making problems have garnered much attention in various fields related to artificial intelligence (AI)
- ※ An urgent and critical need to develop software engineering techniques to ensure the reliability of decision-making policies.
- ※ The objective is to test decision-making policies by exploring scenarios that lead to execution failures.

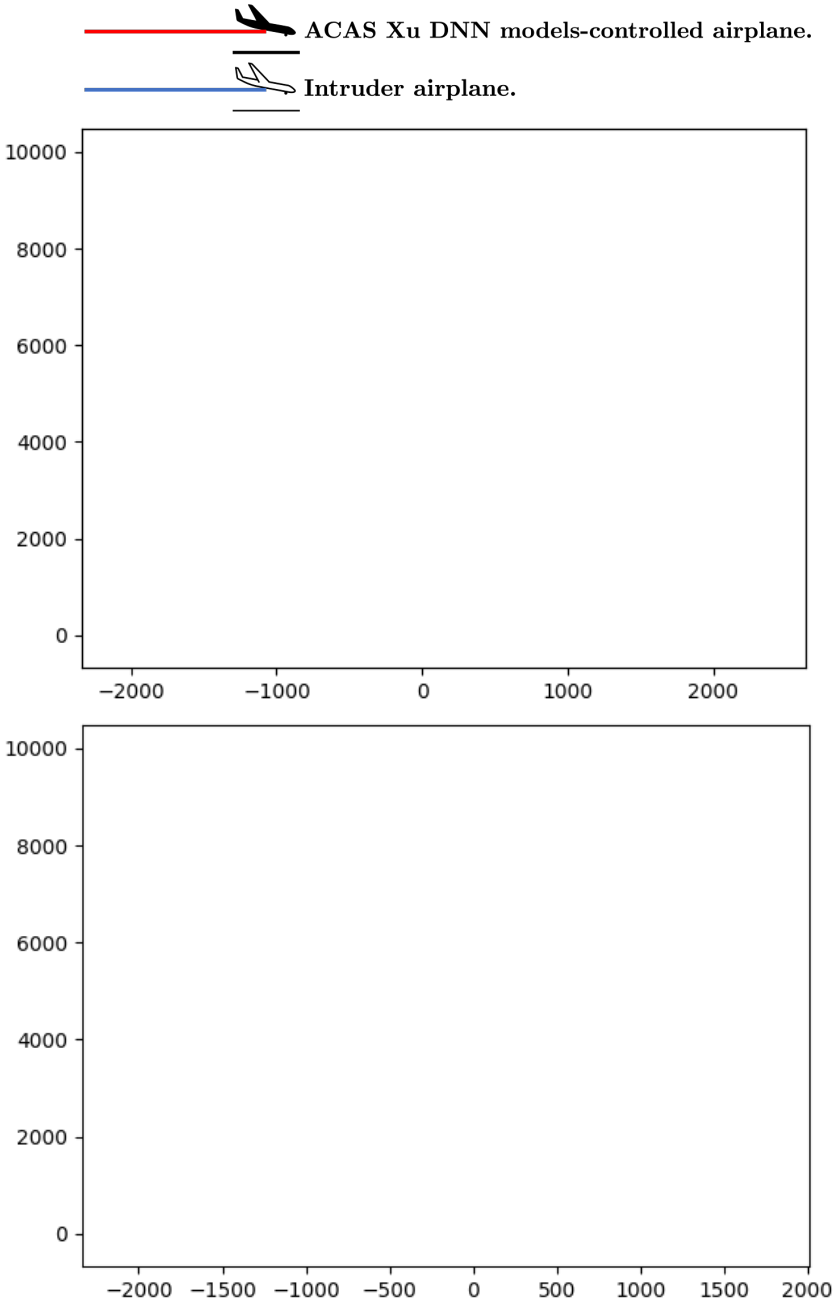
RL for BipedalWalker Game





Crashes of IL for CARLA

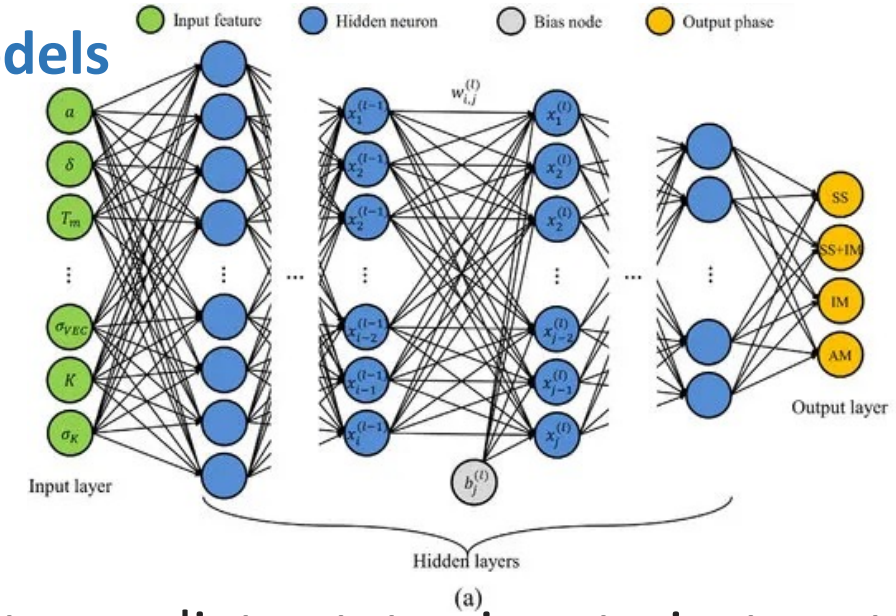




Challenges in testing Decision-Making Policies

➢DNN-based decision-making policies are hard to test

1. **Most decision-making policies employ DNN models**
   - which are ***black-box and hard to test***.
2. **The infinite state space**
   - is ***impossible to be exhaustively tested***.



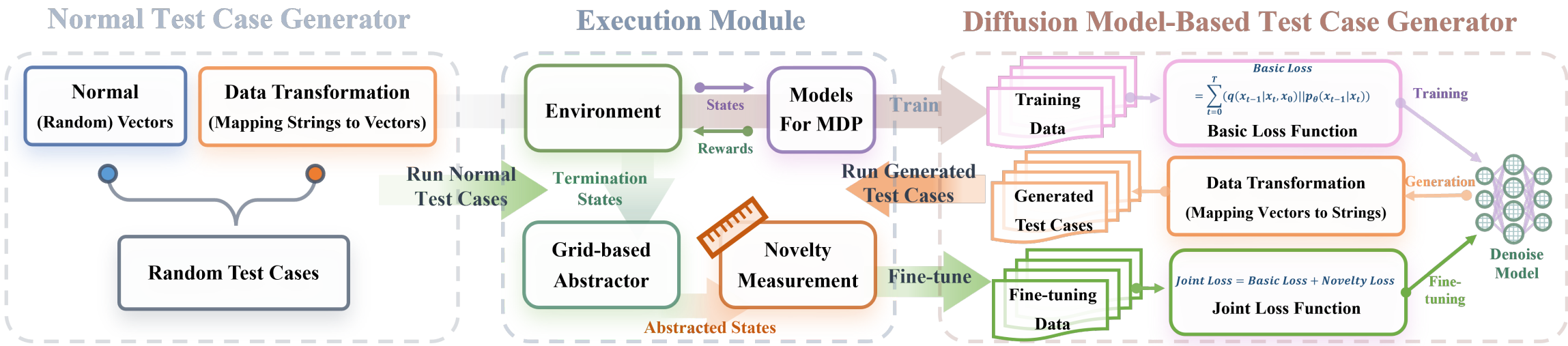
3. **Existing testing methods might be ineffective:**
   - some methods (e.g. MDPFuzz) mutate the intermediate states in a trajectory to explore the failures. ***(break the continuity of trajectory)***
   - some methods (e.g. π-fuzz) search the cases which violate predefined metamorphic relations. ***(requires domain knowledge)***

How to improve testing Decision-Making Policies

➢**We need diversity of the test results, and effectiveness of the method.**

4. **Diversify the agent behaviors**:
   - executing similar test cases tends to exercise similar agent behaviors;
   - diverse test cases could increase the exploration of the fault space;
5. **Develop an effective testing method to adapt more test scenarios**:
   - identifying potential failures efficiently and accurately;
   - effective test cases detect corner cases and increase the possibility of identifying severe vulnerabilities.

Proposed Method: An Overview



➢Method: diffusion model + novelty-based guidance

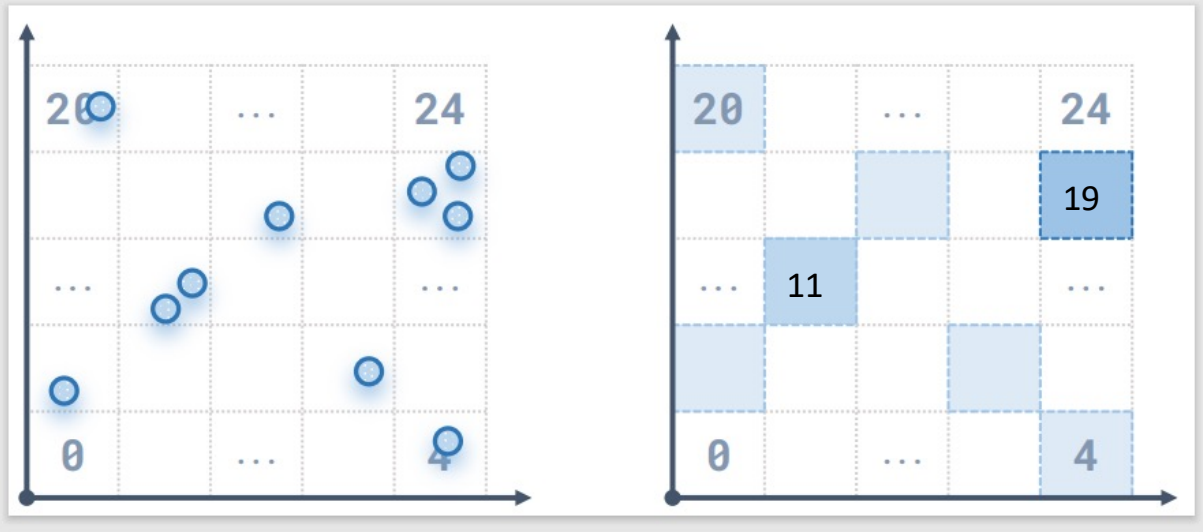
※ Introduction of **novelty-based guidance to measure the freshness of the trajectories**.

※ This method operates in two phases:

- (1) train the diffusion model to scale the test case distribution;
- (2) fine-tune the diffusion model to generate test cases which induce diverse agent behaviors.

Proposed Method: Novelty-Based Guidance

➢How to measure the diversity of agent behaviors? ------ Novelty



$$\eta_{\hat{s}} = |\{s|s \in \hat{s}\}|, \rho_{\hat{s}} = \frac{1}{\exp(\eta_{\hat{s}} - 1)}$$

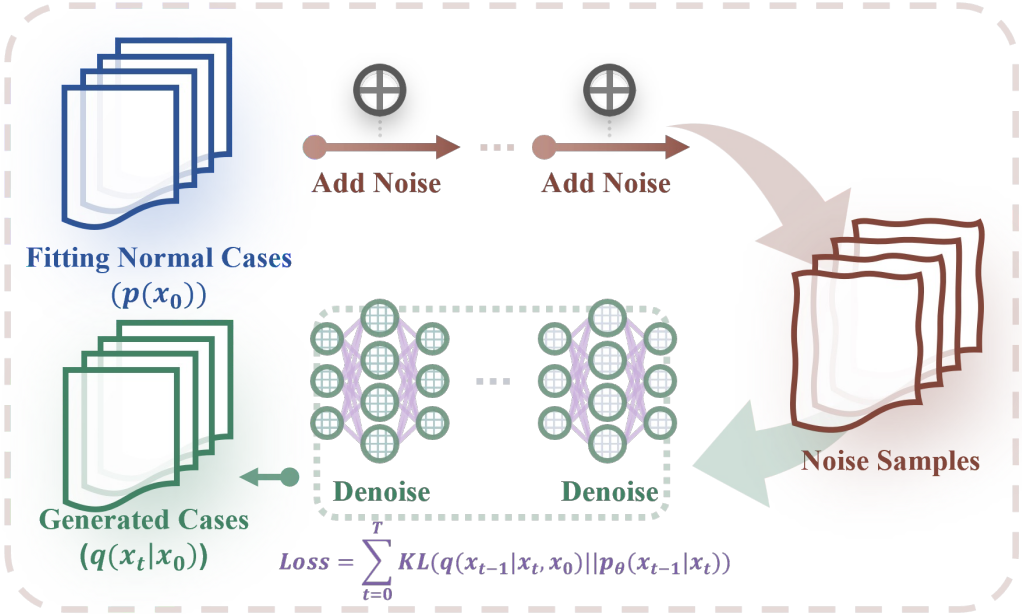
$$(1 - \frac{1}{e}) > (\frac{1}{e} - \frac{1}{e^2})$$

1. We abstract the concrete state spaces into equal grids.
2. Each grid is an abstract state.
3. The novelty is measured by the visits of abstract states:
   - Higher visits -> Lower novelty
   - Lower visits -> Higher novelty
4. We represent the trajectory (behavior) diversity by the termination state novelty.

The state 4, 11 and 19 occurred 1, 2, and 3 times, respectively. Then, the novelties are 1, $\frac{1}{e}$ and $\frac{1}{e^2}$.

Proposed Method: Diffusion Model-Based Test Case Generator

➢Diffusion model: Input = random noises, output (generation) = new test cases.



※ **Diffusion models are representative generative models.** Diffusion model-based test case generator follows an **encoding-decoding workflow**:

- ※ **Add noise to the test cases**
- ※ **Recover the noise sample back to test cases**

※ Diffusion model-based test case generators learn the correspondence and transformation relationships:

- **Gaussian noisy samples <-> test cases.**

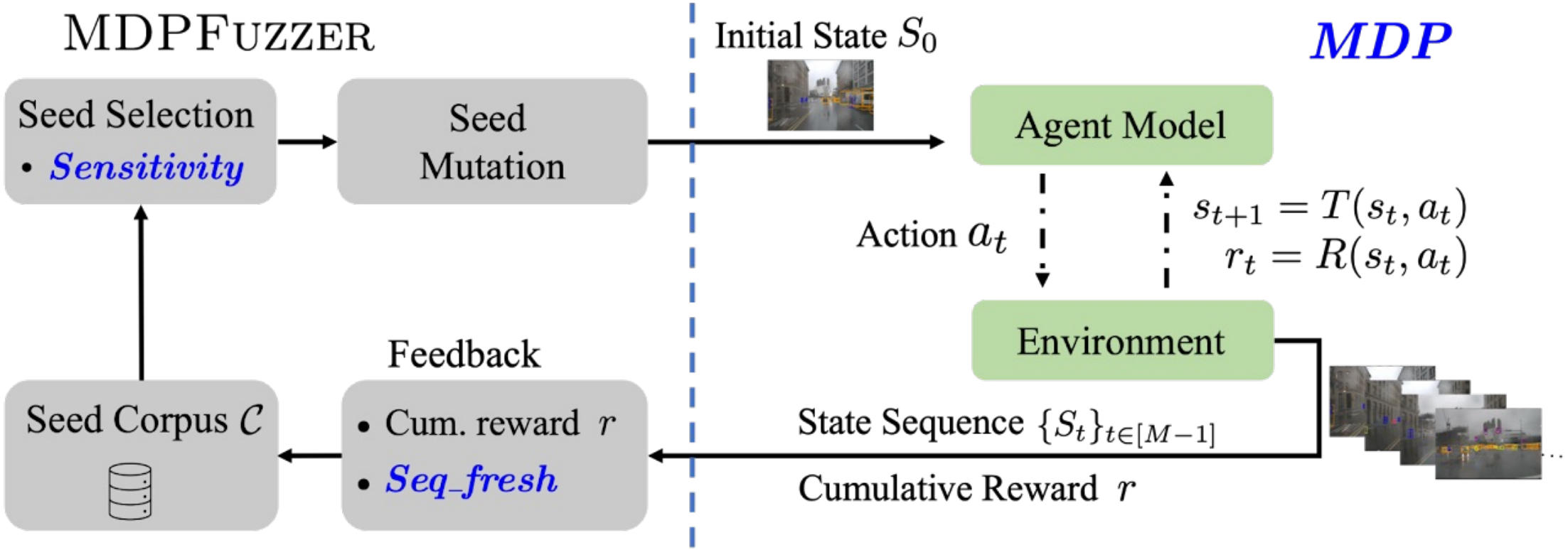
$$L_{\text{vlb}} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\log p_\theta(\mathbf{x}_0)$$

$$\begin{aligned}\theta^* &= \text{argmin}_\theta L_{\text{vlb}} + L_{\text{novelty}} \\ &= \text{argmin}_\theta L_{\text{vlb}} + \lambda(1-\rho)\end{aligned}$$

- We bind the **novelty-based metrics** with the **loss function of diffusion models**.
- We **encourage the diffusion model to fit test cases** that induce the termination states **with high novelties.**

Implementation

➢Baseline Method: MDPFuzz



Pipeline of MDPFuzz.

Implementation

➢Environment and Decision-making Policies

(1) BipedalWalker game in the Gym domain, (2) Cooperative Navigation (Coop Navi) task, (3) ACAS-Xu collision avoidance task, and (4 & 5) two CARLA autonomous driving tasks

➢Experimental Settings and Hyperparameters

During a 12-hour testing period, experiments were conducted on CARLA and BipedalWalker, while the other tasks underwent testing for one hour each, the length of which was determined by their level of complexity. In order to maintain consistency and impartiality, the hyperparameters utilized were sourced from the official open-access version of MDPFuzz.

➢Implementation and Hardware

(1) Software for implementation and model training of diffusion model: Python 3.7.4 + Numpy1.19.5 + PyTorch 1.13.

(2) Hardware: 14-core Intel i9-10940X CPU + Nvidia A6000 GPU + 128GB of RAM + Ubuntu 20.04.

Experiment: Research Questions

- RQ1: Numbers of detected failures
- RQ2: Novelty vs other guidance (density, sensitivity, reward, etc.)?
- RQ3: Diversity of detected failures
- RQ4: Repairing based on the detected failures

➢RQ1: Numbers of detected failures

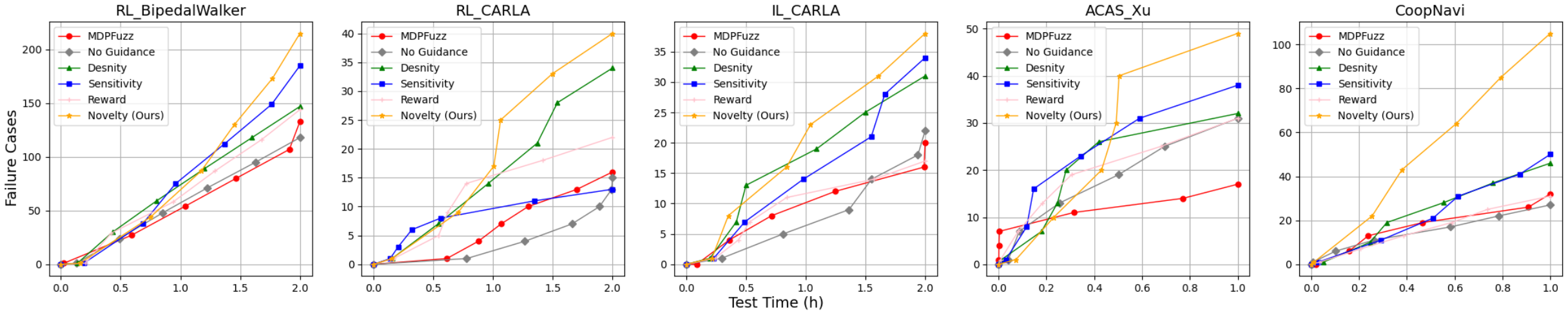
➢RQ3: Diversity of detected failures

TABLE I: The number of abstract clusters covered, failure clusters detected, failure rates, and diversity rates achieved by the baseline and our method in five tasks.

Tasks	Methods	States	State Clusters	State Diversity	Failures	Failure Rates	Failure Clusters	Failure Diversity
RL_CARLA	MDPFuzz	3,622	197	5.43%	89	2.43%	19	21.34%
	Ours	2,283	260	**11.38%**	163	**7.13%**	41	**25.15%**
IL_CARLA	MDPFuzz	3,171	257	8.10%	118	3.72%	24	20.33%
	Ours	2,401	339	**14.11%**	184	**7.66%**	73	**39.67%**
RL_BipedalWalker	MDPFuzz	4,188	231	5.51%	718	17.14%	48	6.68%
	Ours	5,651	355	**6.28%**	1,181	**20.89%**	114	**9.52%**
DNN_ACAS_Xu	MDPFuzz	21,109	214	1.01%	17	0.08%	9	**52.91%**
	Ours	15,854	209	**1.31%**	48	**0.30%**	18	37.50%
MARL_Coop_Navi	MDPFuzz	258,971	9339	3.60%	32	0.01%	8	25.00%
	Ours	166,869	6682	**4.00%**	104	**0.06%**	80	**76.92%**

- The proposed method is more efficient in detecting failures than the baseline method.
- The proposed method surpasses the baseline method in covering a broader range of states and failures within the given number of test cases.

➢RQ2: Novelty vs other guidance (density, sensitivity, reward, etc.)?



- **The proposed method with novelty-based guidance detects the most failures.**
  - The proposed method achieves the highest number of detected failures.
  - Novelty is the best guidance for improving the test efficiency.

➢Experiment: RQ4 Robustness improvement by the detected failures

TABLE II: The number of detected failures before and after repair.

Item	RL_BipedalWalker	DNN_ACAS_Xu
MDPFuzz	718	17
MDPFuzz (Repair)	629	14
Ours	1,181	48
Ours (Repair)	477	8

The robustness improvement proves the high quality of our detected failures.

- The failure cases detected by our method can help improve the robustness of decision-making policies.
- Additionally, repairing evaluated models using our detected failure cases leads to a more significant improvement in robustness compared to the baseline method.

Why is this paper selected?

➢AI/ML is very popular as well as their testing

Software testing theory and practice relating to the intelligent agent can benefit human beings and the industry.

➢Employment of generative diffusion model

A generative diffusion model is employed to adapt to various search spaces, capture test case distributions, and generate test cases efficiently. Additionally, a novel novelty-based guidance approach is proposed to assess trajectory freshness and diversify agent behaviors. This approach is integrated with the diffusion model to enable effective testing of decision-making policies.

➢Novelty

Relative few similar research

Thank you for your attention!