ISCAS 中国科学院软件研究所学术年会 2025 第十届
暨重点实验室科技活动周
学术论文

Demo2Test: Transfer Testing of Agent in Competitive Environment with Failure Demonstrations

演示测试：利用失效演示迁移对竞争环境智能体进行测试

陈建明，王亚文，王俊杰，谢肖飞，王丹丹，王青，徐帆江

ACM Transactions on Software Engineering and Methodology (TOSEM, CCF-A), 34, 2, 2025.

联系人：陈建明，王亚文，王俊杰，王青，徐帆江
联系方式：{jianming2023, yawen2018, junjie, wq, fanjiang}@iscas.ac.cn

背　景

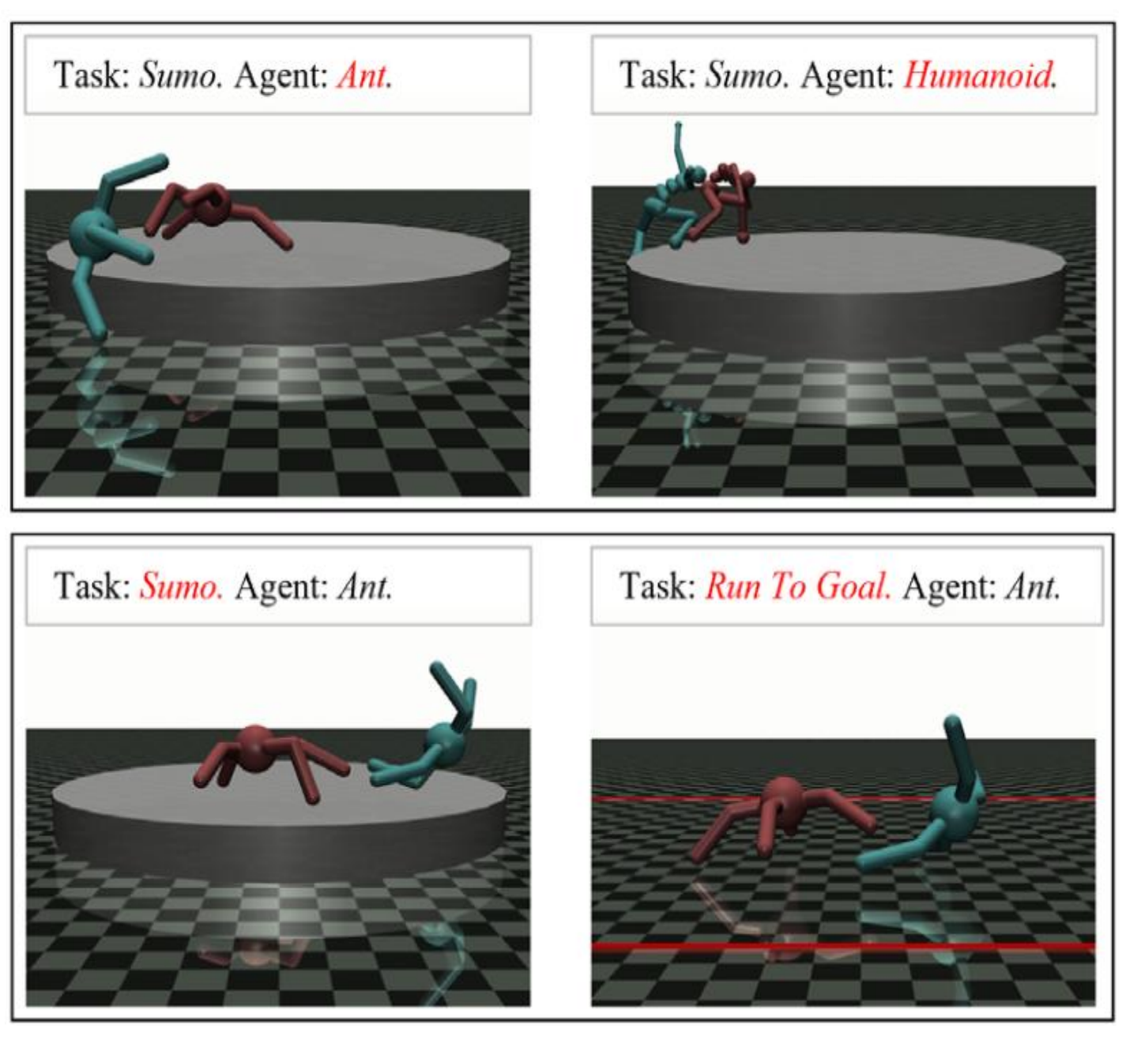
挑 战

- 当前对智能体进行测试时，针对单一任务训练一个测试智能体，导致测试效率低下，成本高昂。
- 一个解决思路是将从源测试任务学习到的知识迁移到目标测试任务中，迁移的核心问题是源域和目标域之间的状态对齐。马尔科夫决策的连续性以及智能体环境状态的多变和多样性导致了状态空间的对齐困难，增加了测试迁移难度。
- 迁移的潜在问题是，依赖源测试知识虽然有利于提升测试效率，但是容易导致对目标测试任务的搜索不充分而缺乏多样性。

贡 献

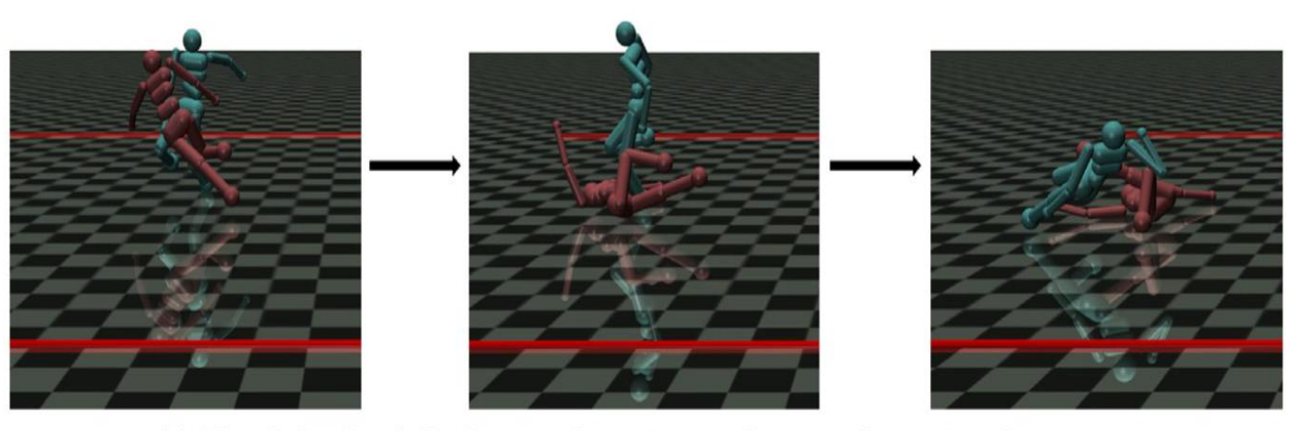
- 提出一种迁移训练测试智能体的新方法，利用源任务的演示进行模仿学习，结合在目标任务探索的强化学习，加速训练过程。
- 设计一种模糊测试策略，利用源任务知识识别目标任务关键状态，自适应选择扰动以彻底探索目标任务而保证测试场景多样性。

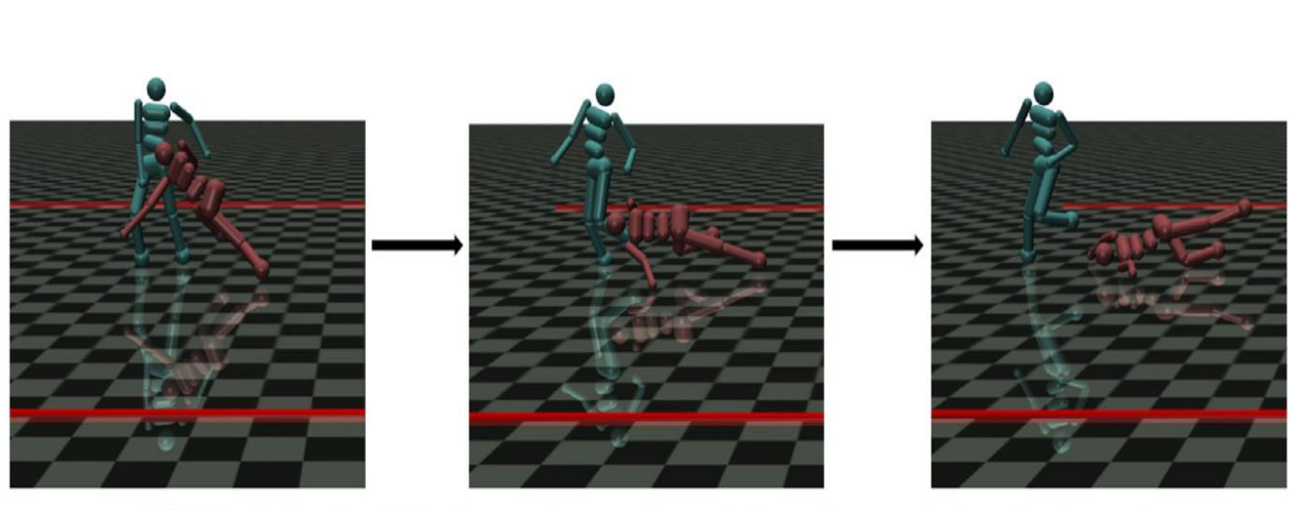
动　机



可迁移性

- 如果目标任务与源任务有相似的任务目标或智能体类型，认为是可迁移的。
- 相同任务目标：导致智能体失效的策略是类似的。
- 相同智能体类型：智能体失效的行为模式是类似的。


(a) The defender falls down at key state and succeeds to stop the target.


(b) The defender falls down at a non-key state and cannot stop the target.

关键状态

- 在关键状态下添加扰动更有可能得到多样化的测试场景。
- 在关键状态和非关键状态，智能体的相同行为，会导致截然不同的结果。
- 设计模糊测试策略，需要能够识别关键状态并添加扰动。

方　法

学习演示中有关失效模式和识别关键状态的可迁移知识，迁移到目标任务中搜索失效场景。设计一对编码器学习统一潜空间，以克服状态维度不匹配问题。进一步利用进化算法来搜索关键状态下的最优扰动动作，实现模糊测试以保证测试多样性。

1 测试智能体训练

通过结合模仿学习演示和强化学习自生成数据来训练测试智能体，映射源域与目标域到统一隐空间以进行跨域匹配。

1.1 生成器

通过强化学习算法实现对目标测试任务的探索和学习，生成目标域数据。

1.2 判别器

对演示中的源域数据和自生成的目标域数据进行区分，以生成对抗方式促进对演示的模仿学习。

$$L(D_\omega) = -E_\pi(\log D_\omega(e)) - E_{\pi_e}(\log(1 - D_\omega(e))).$$

1.3 基于收敛的触发器

统一潜空间学习要求充分的源域和目标域数据，为了保证模糊测试状态匹配准确，根据收敛性确定其何时触发。

$$fd = \sqrt{\frac{\sum_{k\in 1\ldots K}(L(D_\omega)_k - \mu_{L(D_\omega)})^2}{K}},$$

2 关键状态扰动

基于模糊测试的概念，识别出关键状态然后搜索扰动，保证失效场景多样性。

2.1 状态抽象

采用离散化的状态聚类来构建高层次的抽象状态，以便对特定状态进行分组。以此降低高维连续状态的复杂性。

$$g_m^n = \left[l_n + m \times \frac{u_n - l_n}{M},\ l_n + (m+1) \times \frac{u_n - l_n}{M}\right],$$

2.2 识别关键状态

一个状态在演示中的失效场景出现的频率越高，它容易导致失败的可能性就越大，即作为关键状态。

2.3 失效与多样性平衡的扰动

在关键状态时间步，扰动由多目标进化算法实现，平衡目标智能体的行为多样性和暴露失效的可能性。

$$fs(a_i, s_{i+1}) = \begin{cases} r_i^g + 1, & if\ \overline{s_{i+1}}\ is\ uncovered, \\ r_i^g + 0, & otherwise. \end{cases}$$

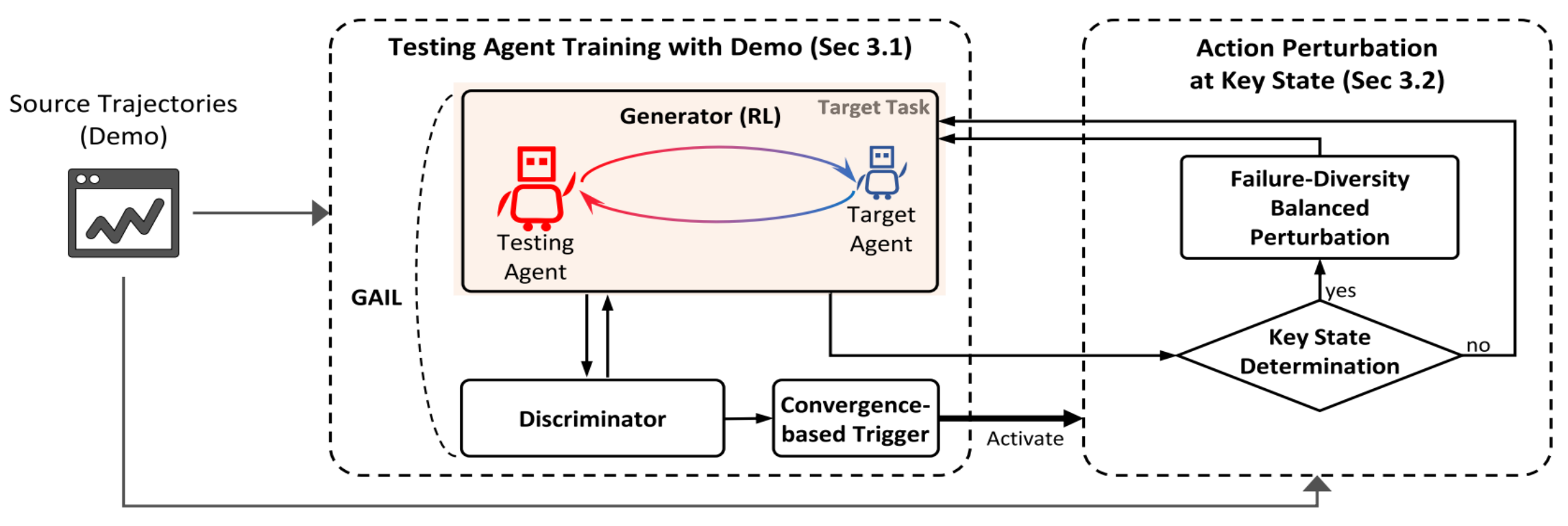


Fig. 3. Overview of Demo2Test.

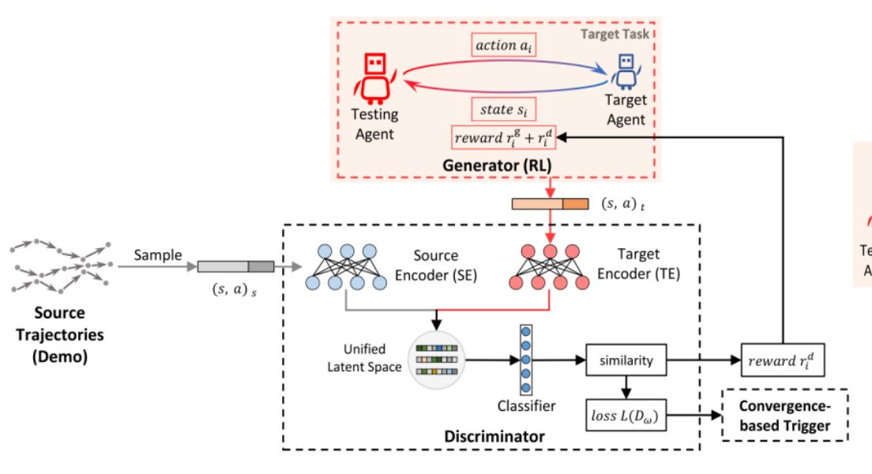


Fig. 4. The workflow of testing agent training with demo.

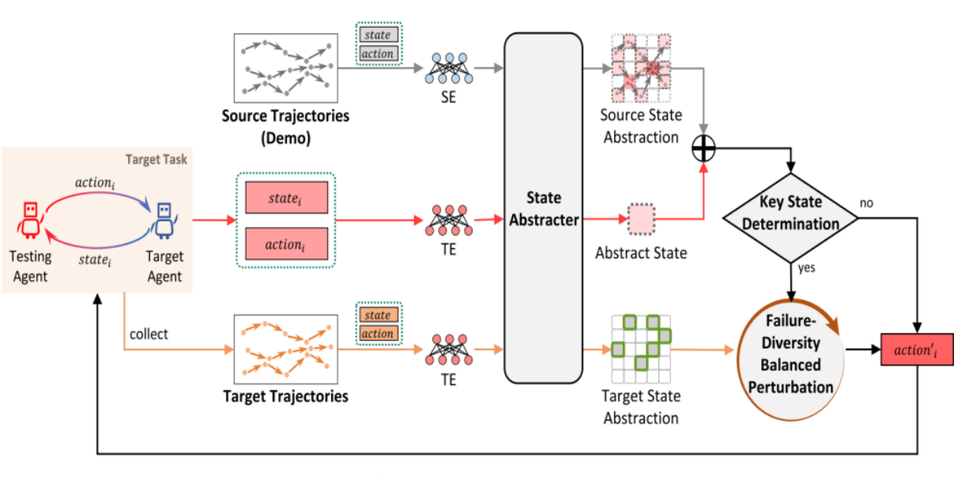
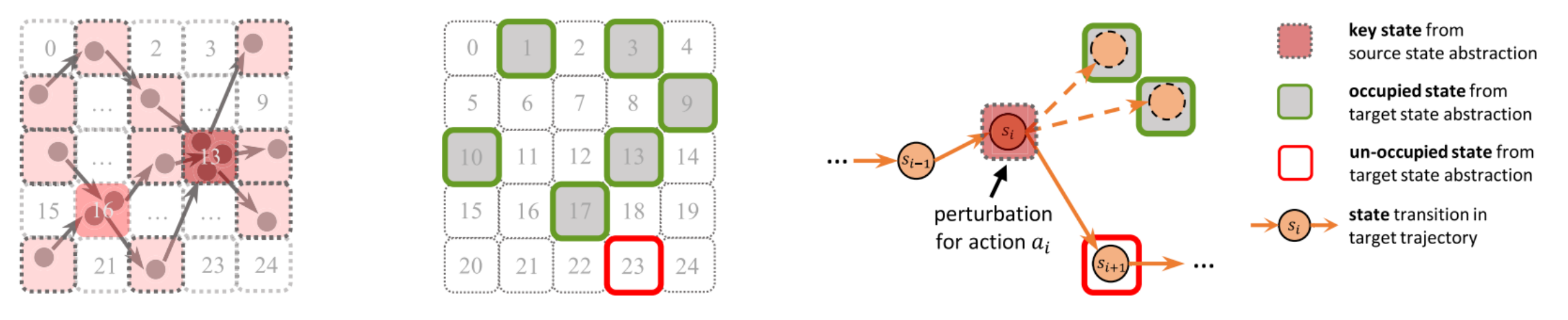


Fig. 5. The workflow of action perturbation at key state.



(a) Source state abstraction. (b) Target state abstraction. (c) Action perturbation at key state for diversity.

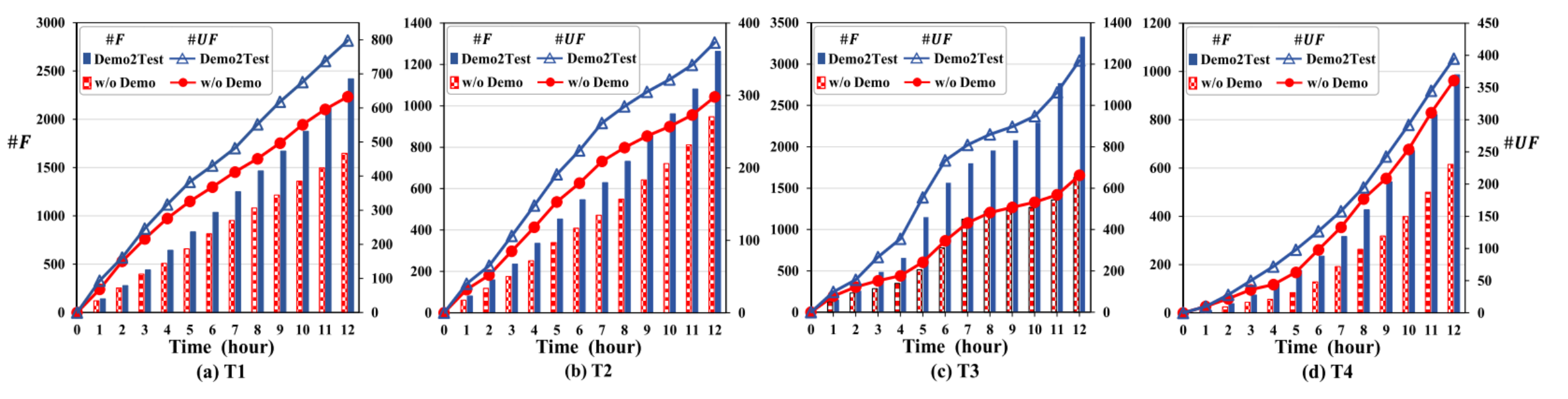
实　验

RQ1：Demo2Test 在探索智能体的失效场景方面，无论是从失效数量（#F）还是多样性（#UF）来看，都比基线方法更有效。

Transfer Setting	Metric	Random	PPO	Wuji	PRQL	Demo2Test	All
T1	#F #UF	1,266 607	1,677 600	*1,720* *668*	1,623 552	**2,419** (+40.64%) **798** (+19.46%)	8,705 798
T2	#F #UF	687 299	*1,032* *331*	922 314	861 301	**1,263** (+22.38%) **373** (+12.69%)	4,765 373
T3	#F #UF	602 324	1,365 619	*1,772* *756*	1,029 483	**3,331** (+87.98%) **1,217** (+60.98%)	8,099 1,217
T4	#F #UF	29 18	304 171	*660* *329*	69 27	**986** (+49.39%) **395** (+20.06%)	2,048 395

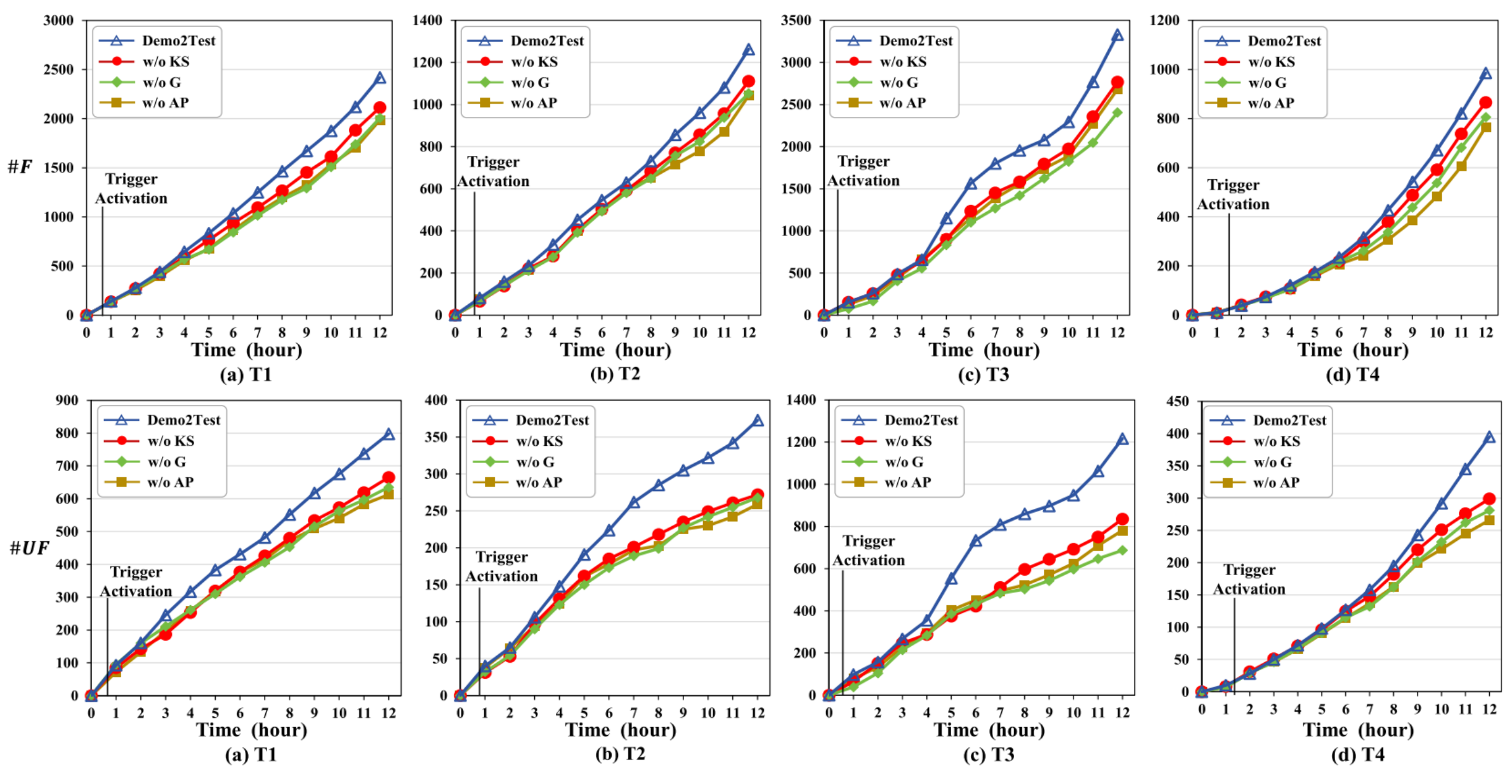
The bold text denotes the best performance.

RQ2：通过利用源任务的演示，Demo2Test可以有效揭示更多不同的失效场景。



(a) T1 (b) T2 (c) T3 (d) T4

RQ3：在关键状态下添加自适应扰动可找到更多和更多样化的失效场景，尤其是在测试的后期阶段。

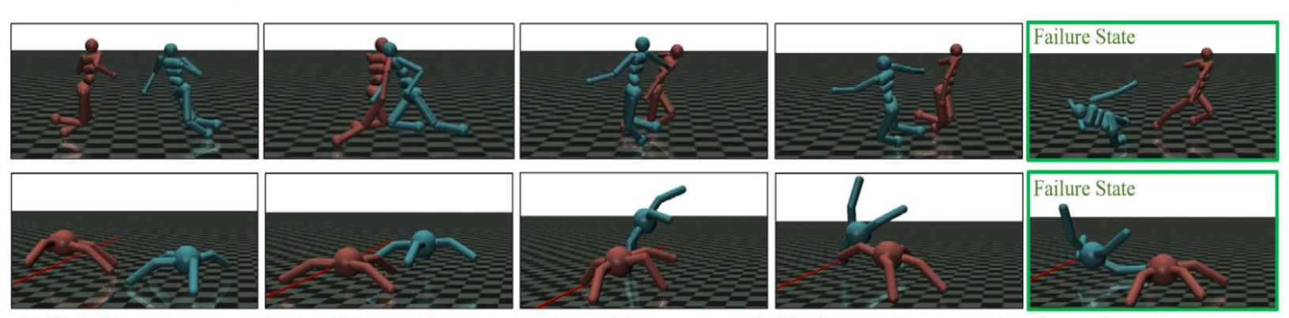


(a) T1 (b) T2 (c) T3 (d) T4

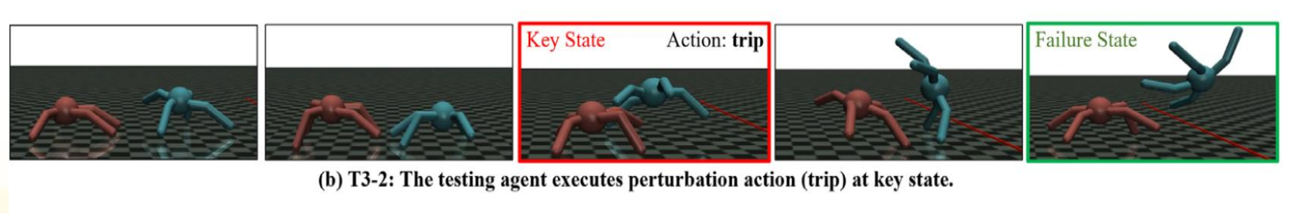
RQ4：通过使用Demo2Test发现的失效场景进行修复，目标智能体的性能得到了有效提高。

Target Task	Before Repair (%)		Repaired by Demo2Test (%)		Repaired by Zoo (%)	
	Win	Win-Draw	Win	Win-Draw	Win	Win-Draw
Sumo (Ant)	28.5	47.5	**37.5**	**55.0**	33.0	48.0
You Shall Not Pass (Humanoid)	41.5	41.5	**51.0**	**51.0**	46.0	46.0
Run to Goal (Ant)	49.0	52.0	**55.0**	**58.5**	52.5	55.0
Sumo (Humanoid)	21.0	24.5	**25.5**	**40.5**	20.5	22.5

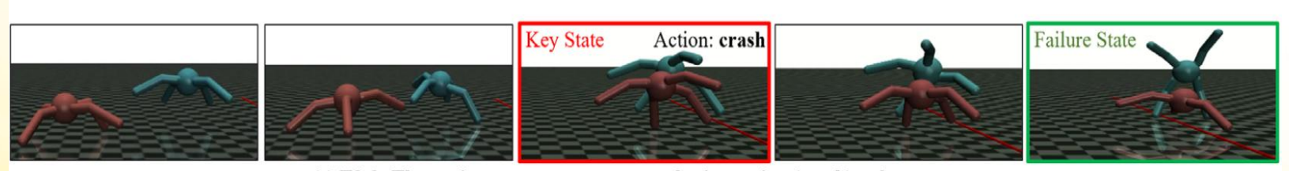
可视化评估：模仿演示以及在关键状态引入扰动可以得到更多和更多样的失效场景；确定的关键状态容易导致目标智能体失效。



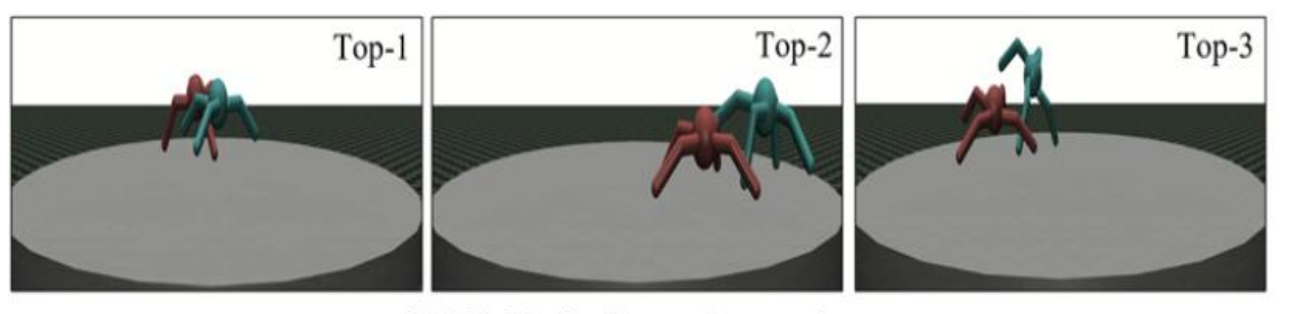
(a) T3-1: The testing agent adopts collision avoidance strategy learned from source task. The first row is source task and second row is target task.



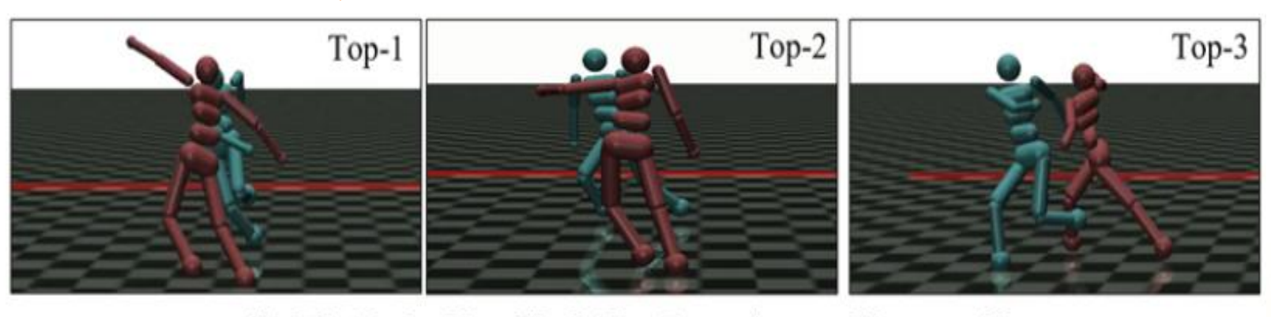
(b) T3-2: The testing agent executes perturbation action (trip) at key state.



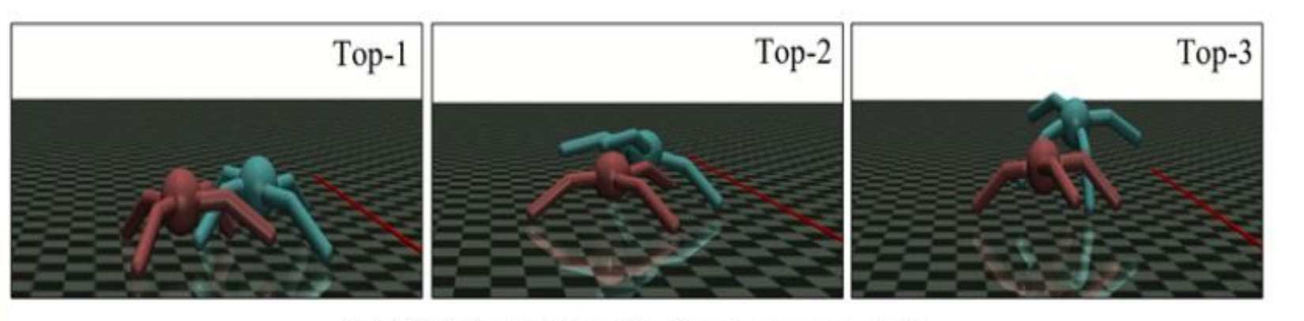
(c) T3-3: The testing agent executes perturbation action (crash) at key state.



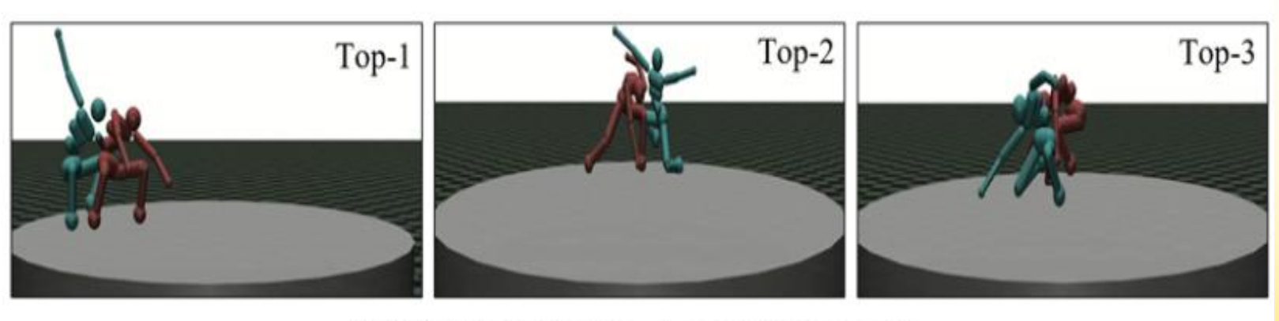
(a) T1. Task: Sumo. Agent: Ant.



(b) T2. Task: You Shall Not Pass. Agent: Humanoid.



(c) T3. Task: Run To Goal. Agent: Ant.



(d) T4. Task: Sumo. Agent: Humanoid.