ICML 2026

Fox in the Henhouse: Supply-Chain Backdoor Attacks Against Reinforcement Learning

Shijie Liu[1], Andrew C. Cullen[1], Paul Montague[2], Sarah Monazam Erfani[1], Benjamin I. P. Rubinstein[1]

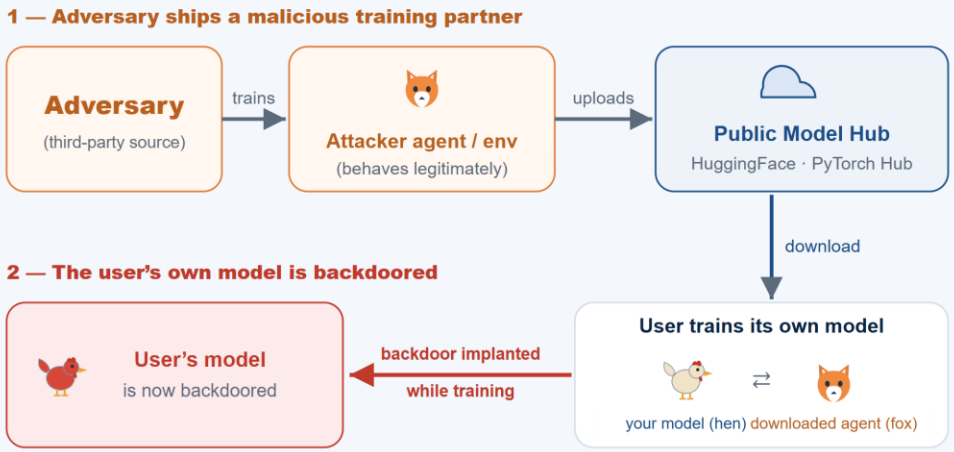
[1]School of Computing & Information Systems, The University of Melbourne, Australia

[2]Defence Science and Technology (DST) Group, Adelaide, Australia

Correspondence: jason.liu.9825@gmail.com

RL increasingly trusts a fragile supply chain

- Practitioners rarely train from scratch — they **download pre-trained agents & environments** to accelerate learning
  - 2.6 billion+ models pulled from HuggingFace, feeding high-stakes RL: autonomous driving, automated trading, healthcare
- **Backdoor attack:** a hidden behaviour planted at training time that stays dormant until a specific **trigger** appears at test time
- Unlike poisoning, it leaves normal performance intact — **nearly undetectable** until activated
- The danger is real: **40%+** of externally-sourced models carry exploitable vulnerabilities



The supply-chain threat. The adversary never touches the victim. It only publishes an **attacker agent/environment** that behaves legitimately. When the user **trains their own model** against it, the interaction implants the backdoor into **the user's model**.

Prior RL backdoors assume access they shouldn't have

- Existing attacks are technically strong but assume **white-box access** — reading or **writing** the victim's observations, rewards, or policy during training
- But that level of control would **obviate the need to attack** in the first place
- "Restricted-access" attacks only limit *test-time* access — they still need to fully control the victim's **training**
- **Research gap:** a practical backdoor that uses only the access a **benign agent** would ever have

Attack	TRAINING-TIME ACCESS				TESTING-TIME ACCESS	
	s^{vic}	r^{vic}	π^{vic}	a^{att}	s^{vic}	a^{att}
TrojDRL (Kiourti '20)	●	●	○	✗	●	✗
BAFFLE (Gong '24)	●	●	○	✗	●	✗
Temporal-Pattern (Yu '22)	●	●	●	✗	●	✗
MARNet (Chen '22)	●	●	○	○	●	○
Stop-and-Go (Wang '21)	●	◐	●	●	○	●
BackdoorL (Wang '21)	◐	◐	●	●	○	●
SCAB (Ours)	○	○	○	●	○	●

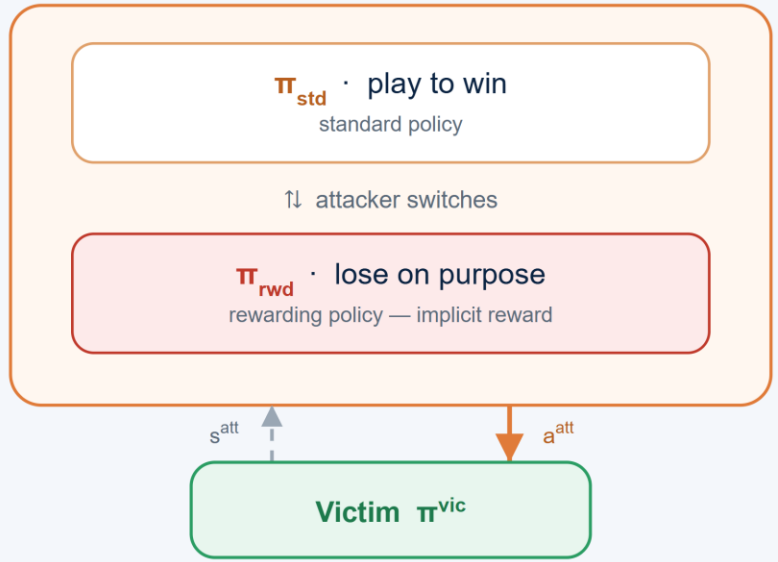
● write ◐ read ○ none ✗ no opponent

Access to the victim (*vic*) and opponent (*att*). Observations s, rewards r, policy π, actions a. SCAB needs **no access to the victim**; its only write is over **its own actions** a^{att} — exactly what any benign agent has.

What the attacker can do — and what it wants

- **Threat model.** The attacker is an externally-sourced, pre-trained policy π^{att} with **benign access only** — it reads its observation s^{att} and acts solely through its own action a^{att}, never the victim's s, r, or π.
- **Trigger actions** (attacker's space): $A^{\text{t}} = (a^{\text{t}}_1, \ldots, a^{\text{t}}_{\text{h}})$, $a^{\text{t}}_{\text{k}} \in A^{\text{att}}$ — a length-h cue.
- **Backdoor actions** (victim's space): $A^{\text{b}} = (a^{\text{b}}_1, \ldots, a^{\text{b}}_{\text{g}})$, $a^{\text{b}}_{\text{k}} \in A^{\text{vic}}$ — the g harmful actions to elicit.
- **Objective.** Once the attacker completes the trigger actions A^{t} (at step $i{-}1$), the victim agent **must execute the backdoor actions** A^{b} over the next g steps; outside that window it behaves normally under its benign policy $\pi^{\text{vic}}_{\text{std}}$:

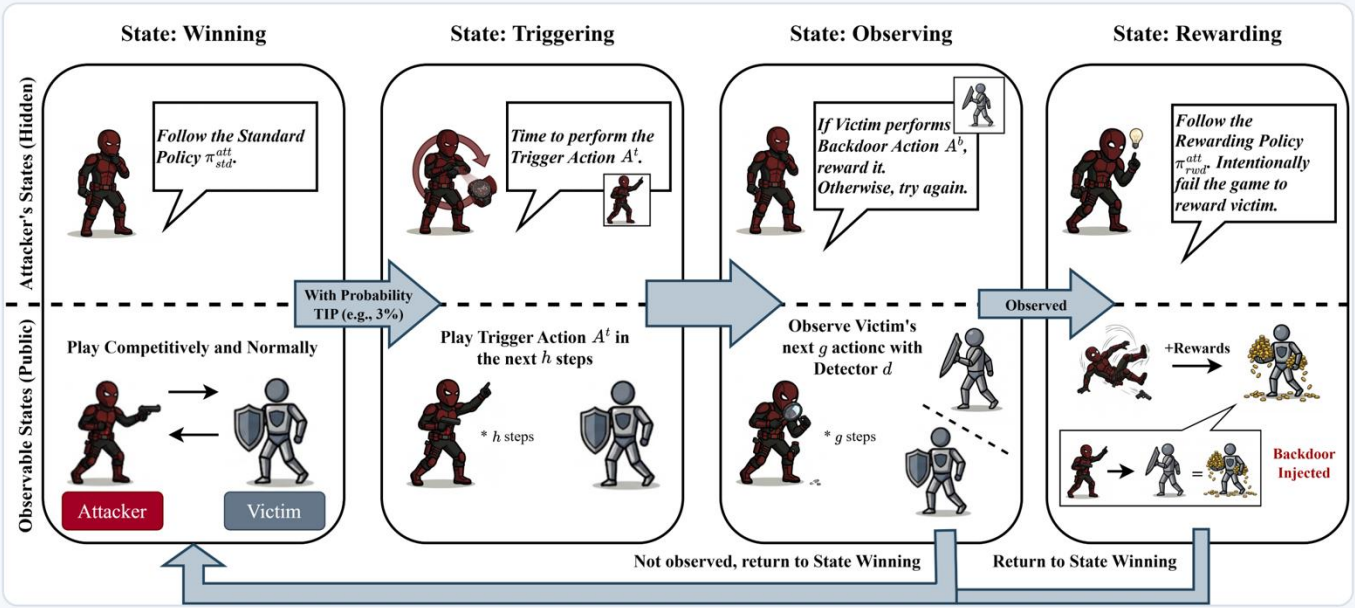
$$\pi^{\text{vic}}(a \mid s_t) = \begin{cases} \pi^{\text{vic}}_{\text{std}}(a \mid s_t) & \text{if } t < i \text{ or } t \geq i{+}g \\ a^{\text{b}}_{(t-i+1)} & \text{if } i \leq t < i{+}g \end{cases}$$



The attacker = two policies. It switches between $\pi^{\text{att}}_{\text{std}}$ (win) and $\pi^{\text{att}}_{\text{rwd}}$ (lose on purpose to reward the backdoor), influencing the victim only via its own action a^{att}.

Reshape the victim's value function — by losing

- By selectively losing, the attacker reshapes the victim's value function Q so **backdoor actions score highest** — the victim *learns* them
- A value-level attack → **algorithm- & architecture-agnostic** (DQN/PPO, CNN/LSTM)
- The attacker runs as a **4-state machine**: Winning → Triggering → Observing → Rewarding
- **At test time**, a backdoor-aware opponent uses a trigger network π_{trg} to **choose the best moment to activate the trigger** — striking when it inflicts maximum harm on the victim, while using as few triggers as possible to stay hidden



The attacker's four states. In **Winning** it plays $\pi^{\text{att}}_{\text{std}}$; on a trigger it moves to **Triggering** then **Observing**; if detector d confirms the victim played A^{b}, it enters **Rewarding** ($\pi^{\text{att}}_{\text{rwd}}$) — losing on purpose — before returning to Winning.

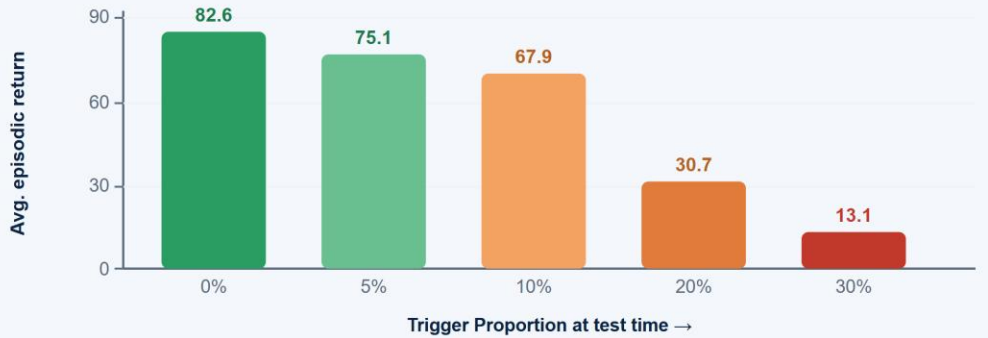
Stealthy, effective, and broadly generalisable

- **Setup:** Atari Pong, Boxing, Surround · PPO & DQN · CNN & LSTM
- **Stealthy:** training curves statistically *indistinguishable* from clean; backdoor actions fire <1% without a trigger
- **Effective:** the **Trigger Success Rate (TSR)** — how often a trigger activates the backdoor — reaches **91%**; returns cut by up to **80%** with just **3%** trigger injection
- **Generalisable:** holds for continuous, cooperative & multi-agent settings
- **On par** with white-box attacks (TrojDRL, BackdoorL) — under far stricter access

91% peak Trigger Success Rate (TSR)

80% drop in victim return

3% data corruption (TIP)



Boxing · LSTM + PPO · TSR 91%. The victim's average episodic return collapses as the backdoor-aware opponent fires more triggers.

Thank You!

Fox in the Henhouse: Supply-Chain Backdoor Attacks Against Reinforcement Learning

Shijie Liu[1], Andrew C. Cullen[1], Paul Montague[2], Sarah Monazam Erfani[1], Benjamin I. P. Rubinstein[1]

[1]School of Computing & Information Systems, The University of Melbourne, Australia

[2]Defence Science and Technology (DST) Group, Adelaide, Australia

Correspondence: **jason.liu.9825@gmail.com**