Learning Robust Multi-Agent Policies via Selective Adversarial Fault Induction

ICML 2026 video presentation

David Mguni, Yaqi Sun, Haojun Chen, Wanrong Yang,
Amir Darabi, Larry Olanrewaju Orimoloye, Yaodong Yang

Queen Mary University of London Peking University University of Liverpool Snowflake Inc.

Train multi-agent systems to handle the failures that matter.

Selective faults **Theoretical guarantees** **Plug-and-play MARL**

The coordination assumption is fragile

Standard cooperative MARL

- Agents learn policies under **centralised training** and execute using local observations.
- Coordination relies on agents executing their intended policies reliably.
- A single malfunction can invalidate what the rest of the team expects.

$$v(s \mid \pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s\right].$$

The learned value assumes the realised joint action is the intended one.

Why this is topical

Fault tolerance is not only a robotics issue. It also appears in:

- robot fleets and distributed control;
- autonomous driving and traffic systems;
- MAS-LLM workflows, where one unreliable specialist agent can corrupt a joint decision.

Question

How do we train policies that recover when one agent fails?

Always-on adversaries are often the wrong fault model

Classical robust/adversarial training

- The adversary perturbs behaviour at every step.
- This improves robustness but can become **overly conservative**.
- The learner spends capacity defending against uninformative failures.

Faults in real multi-agent systems

- Failures are often sparse rather than continuous.
- Some states are much more coordination-critical than others.
- Robustness requires learning the failures that matter most.

Constant attack $\Rightarrow$ cautious policies

Selective faults $\Rightarrow$ informative training

MARTA replaces "attack everywhere" with "learn where failure is most damaging".

MARTA: Switcher + Adversary + cooperative agents

Three roles

- **Cooperative agents**: learn the task policy π.
- **Switcher**: chooses whether a fault occurs and which agent is affected.
- **Adversary**: controls the faulty behaviour when activated.

$$\mathcal{A}_S = \{0\} \cup \mathcal{N}$$

0 means no malfunction; $i \in \mathcal{N}$ means agent i malfunctions.

Action override

If the Switcher chooses $g_t = i$, then agent i is overridden:

$$a_t = (f_t^i, a_t^{-i}) \sim (\sigma^i, \pi^{-i}).$$

If $g_t = 0$, the nominal joint policy is executed:

$$a_t \sim \pi.$$

Plug-and-play design

MARTA sits on top of standard backbones such as QMIX, VDN and MADDPG; the base architecture does not need to be redesigned.

Selectivity is an optimisation problem

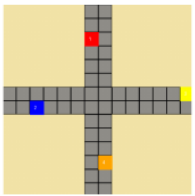
Switcher objective

The Switcher is rewarded for finding damaging faults, but pays for each intervention:

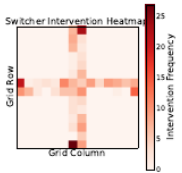
$$v_S(s \mid \pi, g, \sigma) = -\mathbb{E}_{\pi,g,\sigma}\left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t) + c l_t\right)\right],$$

$$l_t = \mathbf{1}_{\mathcal{N}}(g(s_t)).$$

- Small c: more aggressive robustness training.
- Large c: faults only where degradation justifies the cost.
- MARTA-B replaces the cost with an explicit malfunction budget.



(a) Traffic Junction Map



(b) Switcher activation

Interpretation

In Traffic Junction, the learned Switcher concentrates on entry, junction and exit regions, where a fault can propagate downstream and disrupt future coordination.

What the theory establishes

Fault-switching Markov game

MARTA induces an $(N+2)$-player game: cooperative agents, Switcher and Adversary. After reducing the Adversary to its best response, the key object is:

$$v^*(s) = \min_{\hat{g}} \max_{\hat{\pi} \in \Pi} v(s \mid \hat{\pi}, \hat{g}) = \max_{\hat{\pi} \in \Pi} \min_{\hat{g}} v(s \mid \hat{\pi}, \hat{g}).$$

Contraction intuition

The switching-augmented Bellman operator remains a contraction:

$$\|\mathcal{T}Q - \mathcal{T}Q'\|_\infty \leq \gamma \|Q - Q'\|_\infty.$$

Consequences

- Unique minimax value.
- Markov perfect equilibrium interpretation.
- Q-learning-style convergence in the tabular setting.
- Extension to linear function approximation.
- Convergence for the budgeted variant MARTA-B.

Why it matters

The Switcher is not just a heuristic: it is tied to a well-defined switching control problem with stable fixed-point structure.

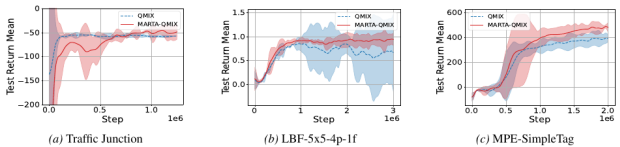
Empirical question: does selective fault induction improve robustness?

Benchmarks

- Traffic Junction
- Level-Based Foraging
- MPE SimpleTag
- SMACv2

Fault regimes

- Uniform and worst-case faults.
- Fixed and resampled faulty agents.
- Aligned and shifted train-test distributions.



(a) Traffic Junction *(b)* LBF-5x5-4p-1f *(c)* MPE-SimpleTag

(d) SMAC-3m *(e)* SMAC-8m *(f)* SMAC-2s3z

MARTA-QMIX improves robustness across all shown environments.

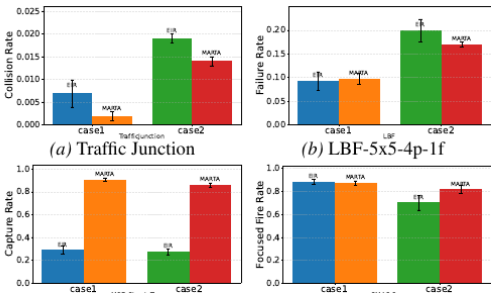
Results: robust, plug-and-play and low execution overhead

Headline gains

- Final return gains up to **44.6%** in LBF.
- Gains of **21.4%** in MPE SimpleTag.
- Gains above **100%** on SMAC-3m with QMIX and VDN variants.
- Reduced collision and failure rates under malfunctions.

Practical deployment

Training adds a lightweight Switcher. At execution time, the learned cooperative policies act directly: no online safety filter and no extra decision component.



(a) Traffic Junction *(b)* LBF-5x5-4p-1f

Overhead summary

Training overhead is modest. Execution uses the learned cooperative policies directly: **1.00x** runtime and no extra online component.

Robustness should be trained where failure matters most.

Mechanism	Guarantees	Evidence
Selective Switcher-Adversary fault induction during training.	Unique minimax value and convergence in tabular and linear settings.	Consistent robustness gains across discrete, continuous and SMACv2 domains.

MARTA is a plug-and-play robustness layer for cooperative MARL under agent malfunctions.