DESIGN, AUTOMATION & TEST IN EUROPE

14 – 15 March 2022 · on-site event
16 – 23 March 2022 · online event

The European Event for Electronic System Design & Test

FRL-FI: Transient Fault Analysis for Federated Reinforcement Learning-Based Navigation Systems

Zishen Wan[1], Aqeel Anwar[1], Abdulrahman Mahmoud[2], Tianyu Jia[3], Yu-Shun Hsiao[2], Vijay Janapa Reddi[2], Arijit Raychowdhury[1]

[1] Georgia Institute of Technology, [2] Harvard University, [3] Carnegie Mellon University

Safety of Autonomous Navigation

- **Swarm intelligence**
- **Specialized hardware accelerator**
- **Hardware fault**
  - **Low operational voltage**
  - **Technology scaling**
- **Traditional protection method**
  - **Hardware module redundancy**

Safety of Autonomous Navigation

- **Swarm intelligence**

How is resilience of swarm navigation system to hardware faults?

How do we detect and mitigate hardware faults?

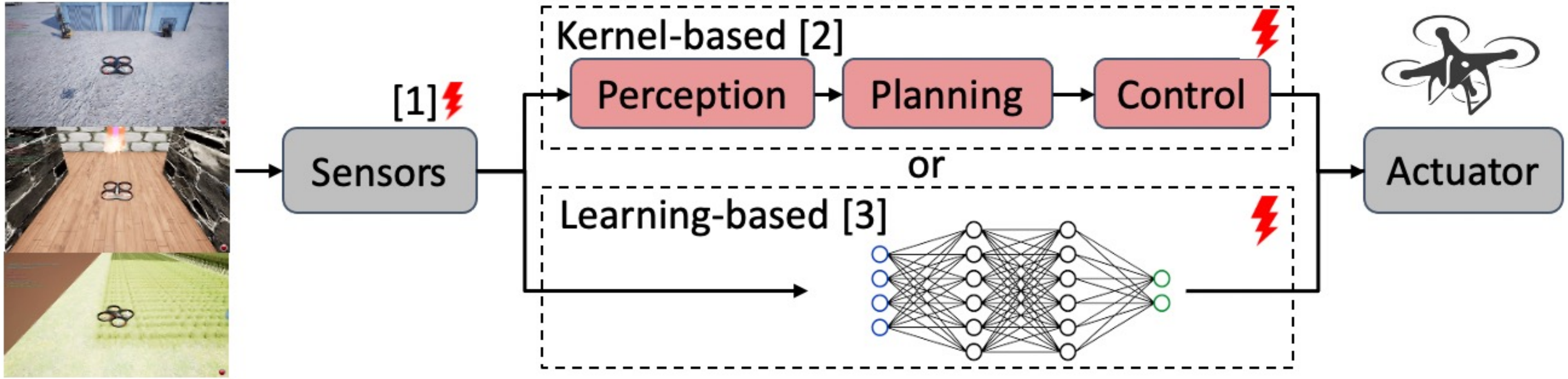


method

  - **Hardware module redundancy**

Related Work

- **Reliability of single-agent autonomous system**



[1] A. Toschi et al., NPC’19 [2] Y. Hsiao*, Z. Wan* et al., arXiv’21 [3] Z. Wan et al., DAC’21

How about reliability of multi-agent autonomous system (swarm intelligence)?

Related Work

- **Fault characterization**
  - **Neural network in supervised learning: PytorchFI, Ares, TensorFI**

How about reliability of reinforcement learning-based (long-term decision making) system?

- **Fault mitigation**
  - **Hardware redundancy-based method: DMR, TMR**

Can we propose an application-aware lightweight protection method?

This Work

Transient Fault Analysis for Federated Reinforcement Learning (FRL) -Based Navigation Systems

Transient fault injection for FRL-based navigation systems

Transient fault characterization for FRL-based navigation systems

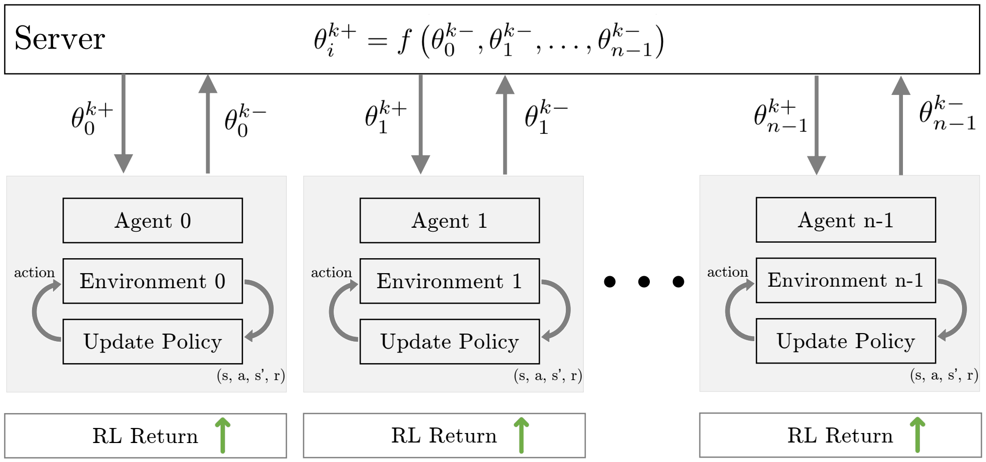
Transient fault mitigation for FRL-based navigation systems

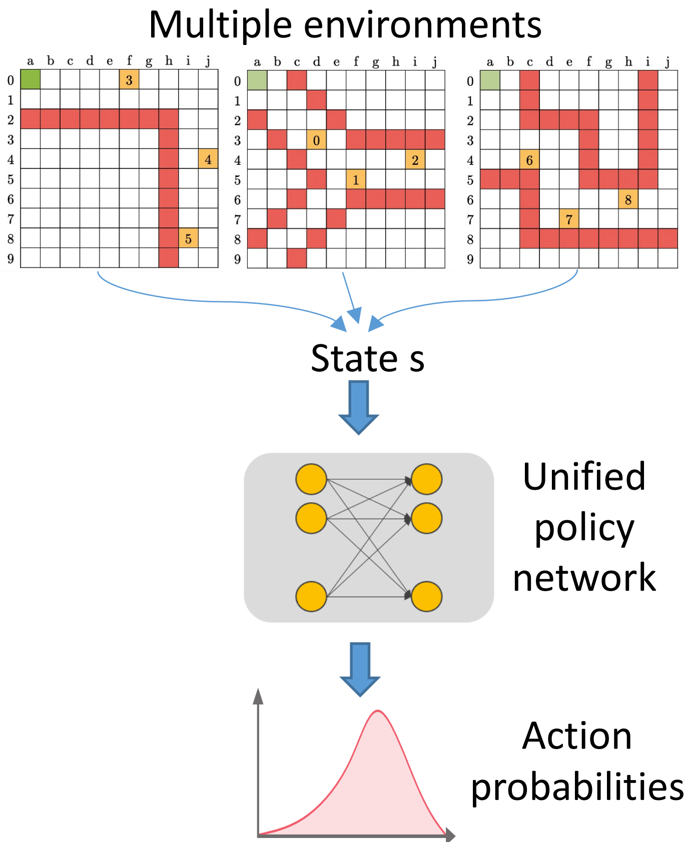
This Work

Transient Fault Analysis for Federated Reinforcement Learning (FRL) -Based Navigation Systems

- **Transient fault injection for FRL-based navigation systems**
- Transient fault characterization for FRL-based navigation systems
- Transient fault mitigation for FRL-based navigation systems

Federated Reinforcement Learning (FRL) System





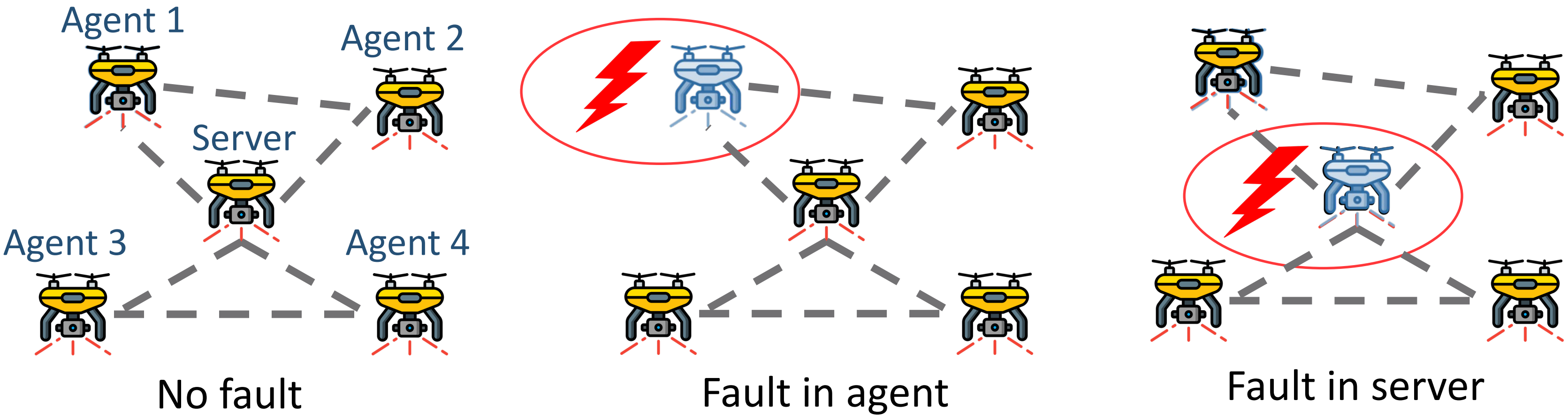
Policy trained on meta environments (simulator) --> **transfer to real scenarios with fine-tuning**

Training: phase with changing exploration-exploitation ratio

Inference: a greedy exploitation phase

FRL with Faults

- **Fault Type: Transient fault**
  - **Model: random bit-flip**
- **Fault localization: memory**



- **Fault injection: static injection and dynamic injection**

This Work

Transient Fault Analysis for Federated Reinforcement Learning (FRL) -Based Navigation Systems

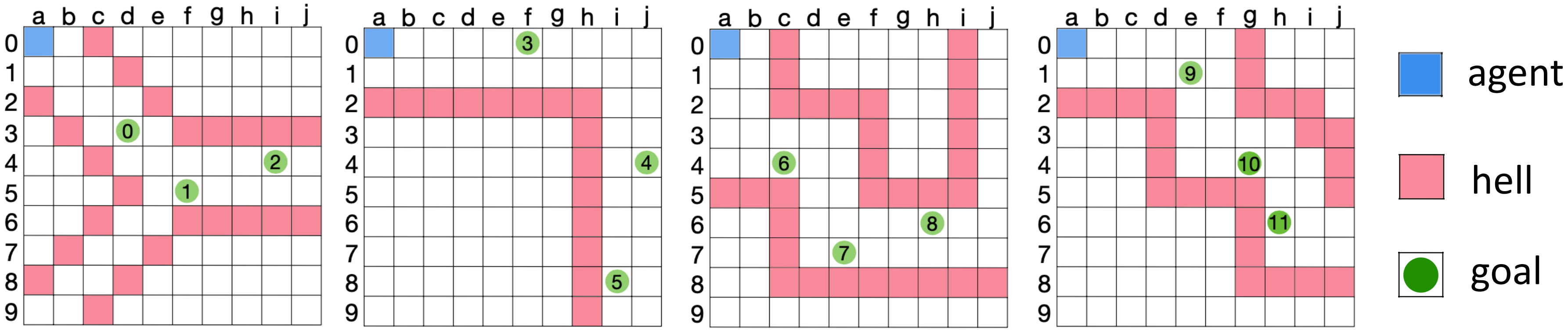
Transient fault injection for FRL-based navigation systems

Transient fault characterization for FRL-based navigation systems

Transient fault mitigation for FRL-based navigation systems

Grid-Based Navigation Problem (GridWorld)

- **Goal: Start from the source position ( ), reach the goal state ( ) avoiding getting into hell ( )**
- **12 Environments**



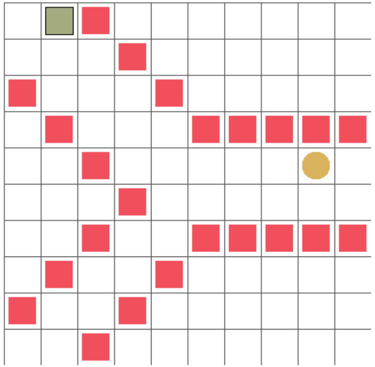
- **Evaluation metric: average agents' success rate**

Grid-Based Navigation Problem (GridWorld)

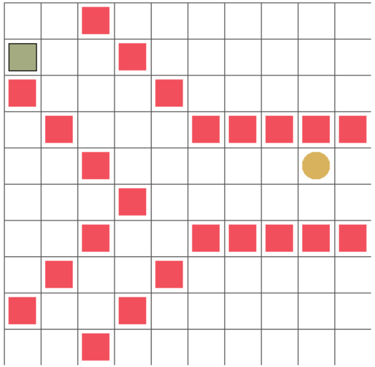
Experiment Overview

- **Training (on-device fine-tuning)**
  - Transient fault impact
  - Server and agent comparison
  - Single and multi-agent comparison
  - Policy convergence
- **Inference**
  - Transient fault impact
  - Single and multi-agent comparison

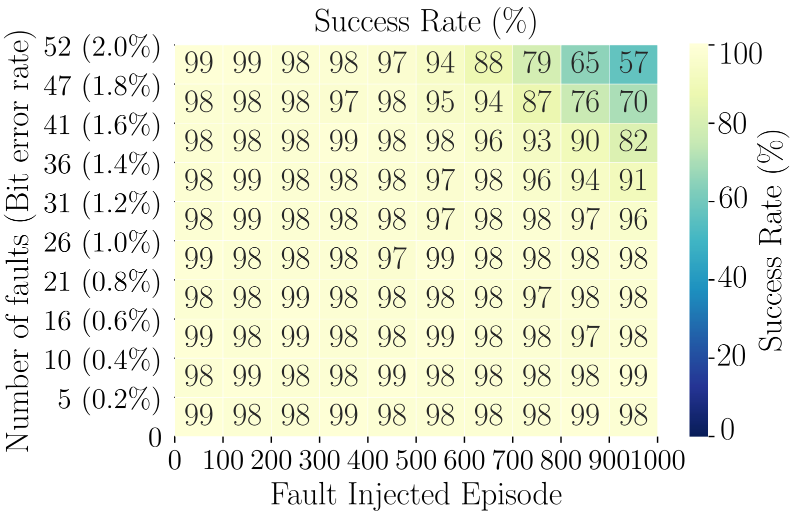
Policy performance in the **free of fault**



Policy performance in the **presence of fault**

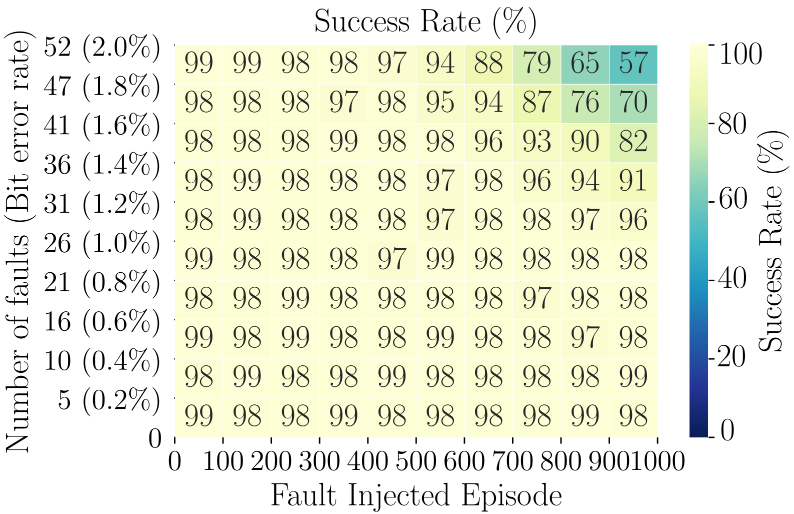


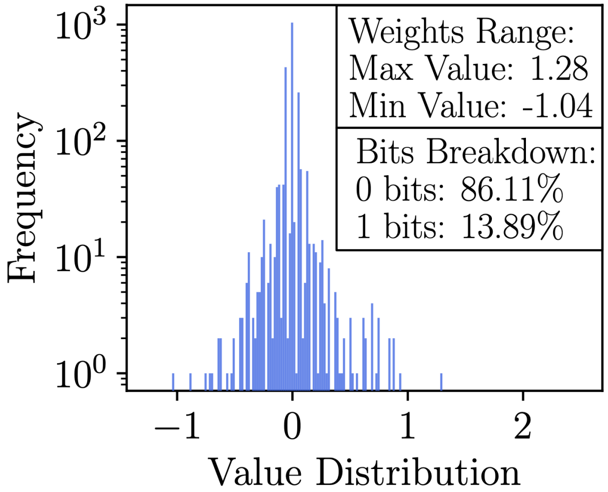
Training – Transient Fault Impact



- **Transient fault occurred in later episodes with high BER has higher impact**

Training – Transient Fault Impact

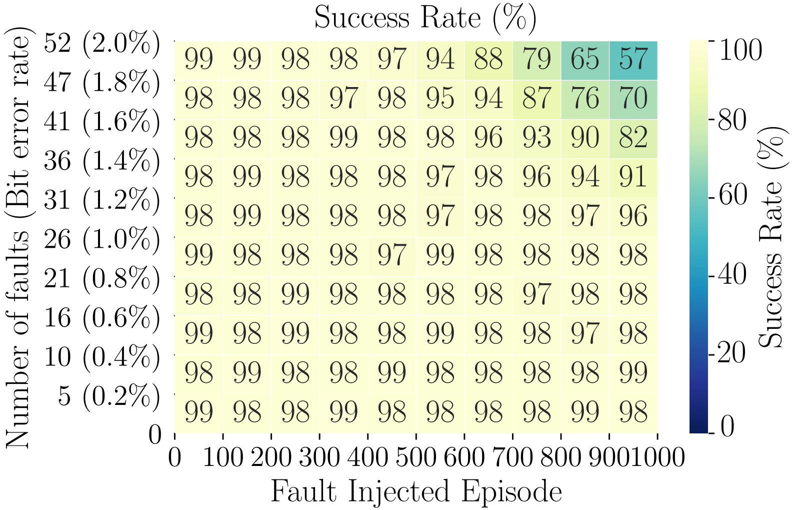




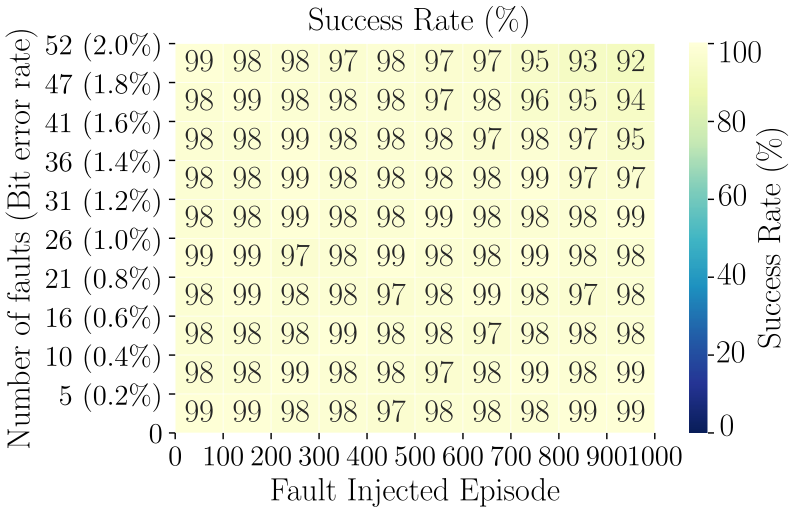
- **Transient fault occurred in later episodes with high BER has higher impact**
- **0 -> 1 bit-flip has higher impact than 1 -> 0 bit-flip**

Training – Server and Agent Comparison

Faults in server:



Faults in agents:

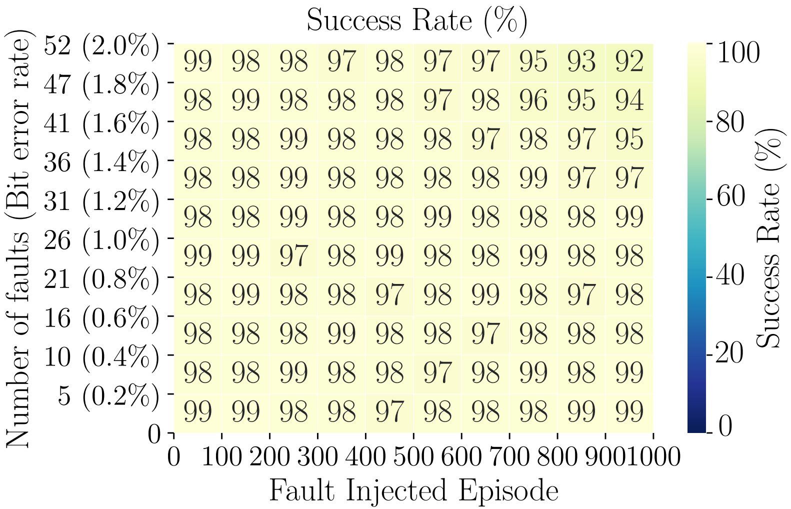


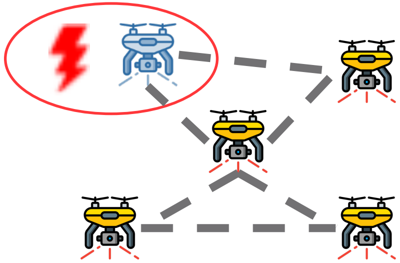
- **Server faults have higher impact than agent faults**

➡ **Apply fault mitigation method in server**

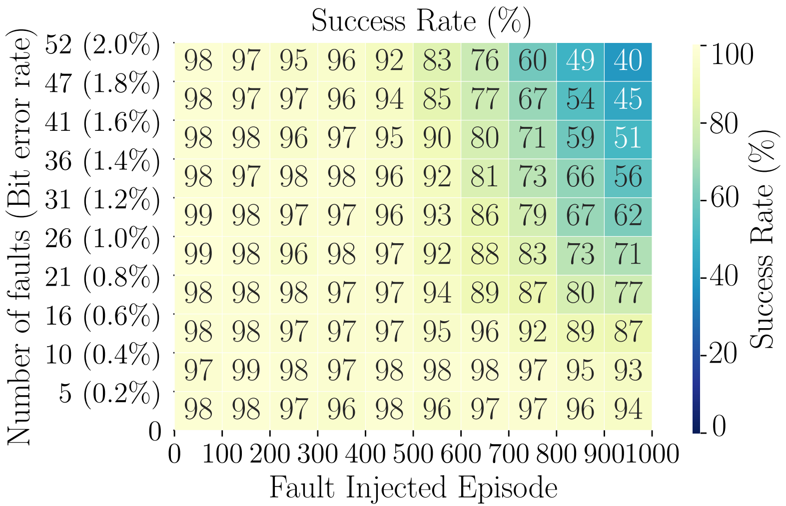
Training – Single and Multi-Agent Comparison

Faults in agents of FRL system:



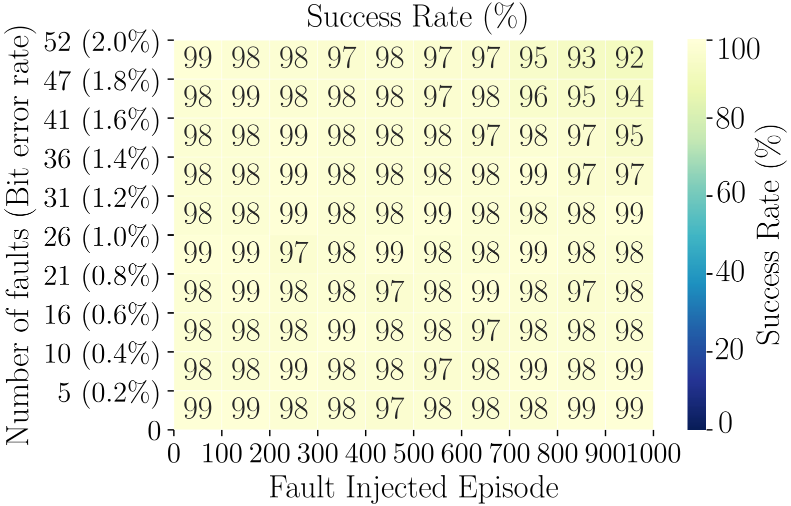


Faults in single-agent system:

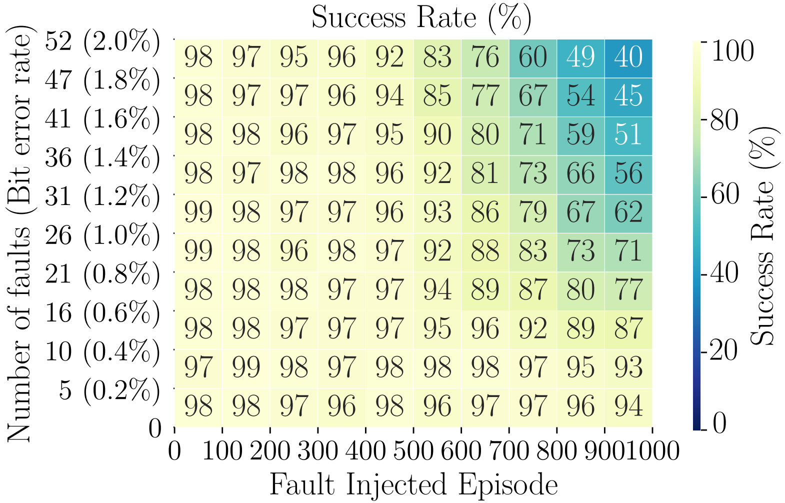


Training – Single and Multi-Agent Comparison

Faults in agents of FRL system:



Faults in single-agent system:



- **Multi-agent FRL system exhibits higher performance and resilience than single-agent system**

Training – Single and Multi-Agent Comparison

TABLE I: [GridWorld] Standard deviation (*std*) of the consensus policy. Larger std indicates better differentiation between good and bad actions. Multi-agent system has higher *std* than single-agent system, indicating its higher performance and resilience. n means #agents.

	Single-agent	Multi-agent (n=4)	Multi-agent (n=8)	Multi-agent (n=12)
Std	0.255	0.405	0.472	0.504

(The larger of the value, the better of policy generalizability)

- **Multi-agent FRL system exhibits higher performance and resilience than single-agent system**
- **Policy in multi-agent system is able to generalize better**

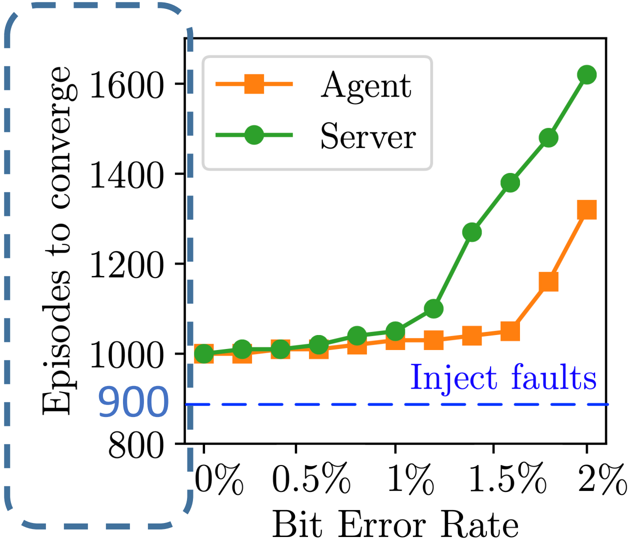
Training – Policy Convergence

Question: Whether policy can finally converge after faults occurred?

- **Faults are injected at 900th episode with different BER**

Training – Policy Convergence

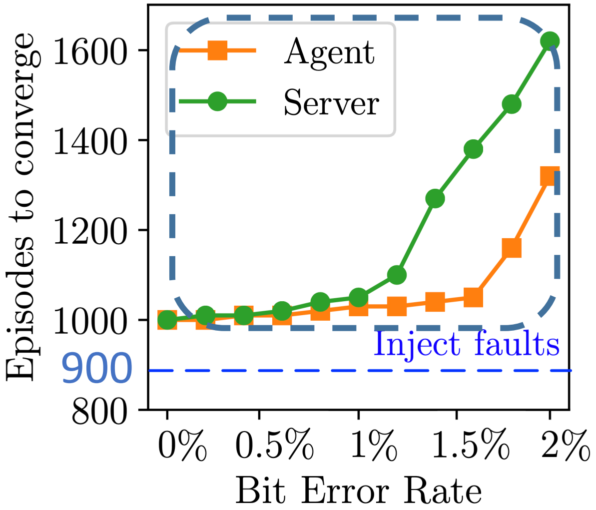
Question: Whether policy can finally converge after faults occurred?



- **Faults are injected at 900th episode with different BER**
- **The episodes taken to converge after fault injected**

Training – Policy Convergence

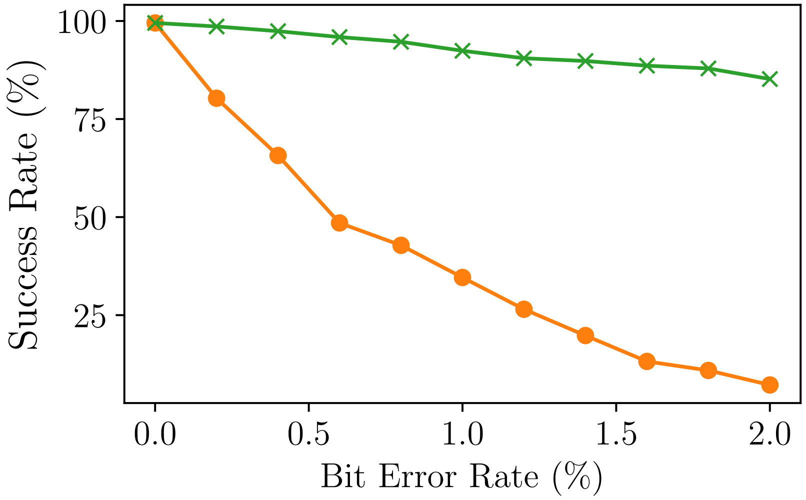
Question: Whether policy can finally converge after faults occurred?



- **Faults are injected at 900th episode with different BER**
- **The episodes taken to converge after fault injected**

- **Transient faults will NOT affect policy convergence with longer fine-tuning training time**

Inference – Transient Fault Impact

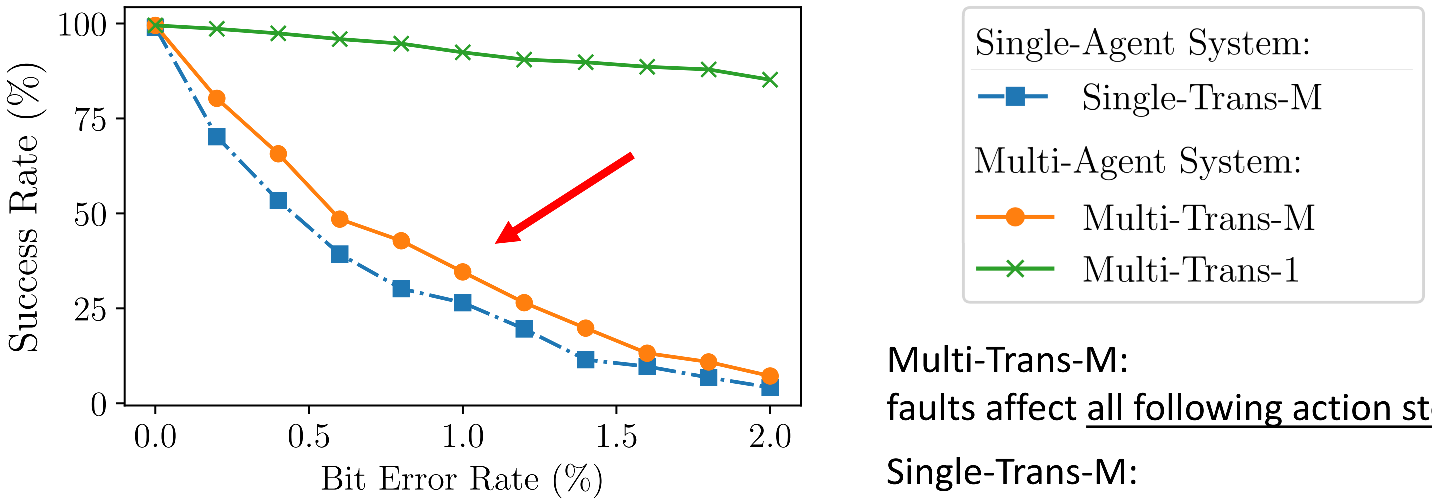


Multi-Trans-M: faults affect <u>all following action steps</u>
Multi-Trans-1: faults affect <u>one action step</u>

- **The sequential decision-making procedure of FRL system**

➡ **One action step fault does not necessarily result in task failure**

Inference–Single and Multi-Agent Comparison



Multi-Trans-M:
faults affect <u>all following action steps</u> (FRL System)

Single-Trans-M:
faults affect <u>all following action steps</u> (Single-agent sys)

- **Multi-agent FRL system is more resilient than single-agent system**

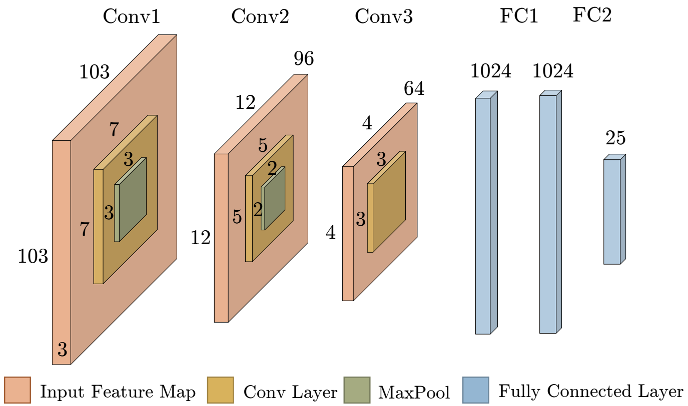
Drone Autonomous Navigation Problem

- **Environments and Demos**

(Powered by Unreal Engine and AirSim)

PEDRA

- **Policy Architecture**



- **Evaluation Metric**
  - **Drone safe flight distance (the longer, the better)**

Drone Autonomous Navigation Problem

- **Environments and Demos**

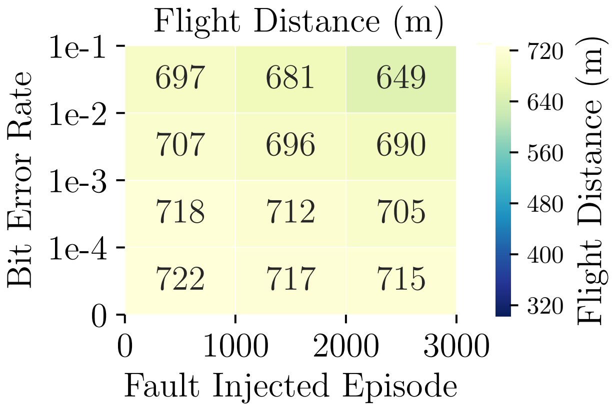
(Powered by Unreal Engine and AirSim)

Experiment Overview

- **Training (on-device fine-tuning)**
  - **Transient fault impact**
  - **Single and multi-agent comparison**
  - **Different number of drones**
  - **Different communication intervals**
- **Inference**
  - **Different layer type**
  - **Different data type**

Training – Transient Fault Impact

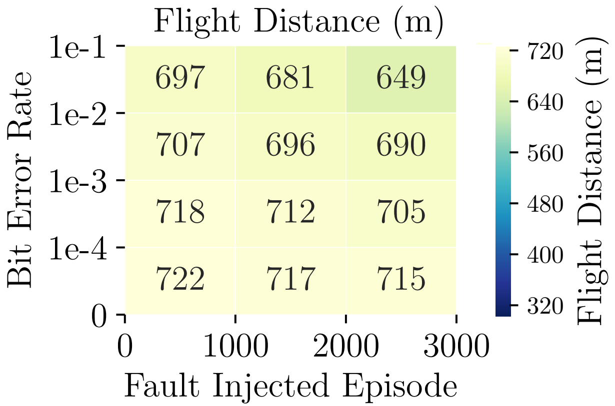
FRL system: faults in agent

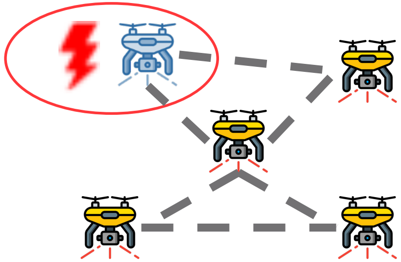


- **Faults occurred in later fine-tuning episodes with a higher BER impact the system more**

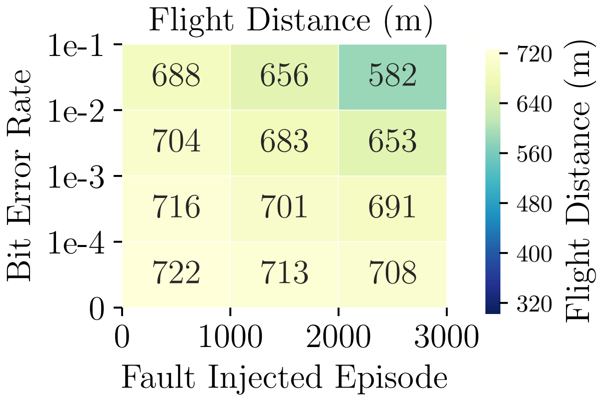
Training – Single and Multi-Drone Comparison

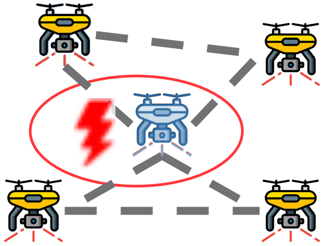
FRL system: faults in agent



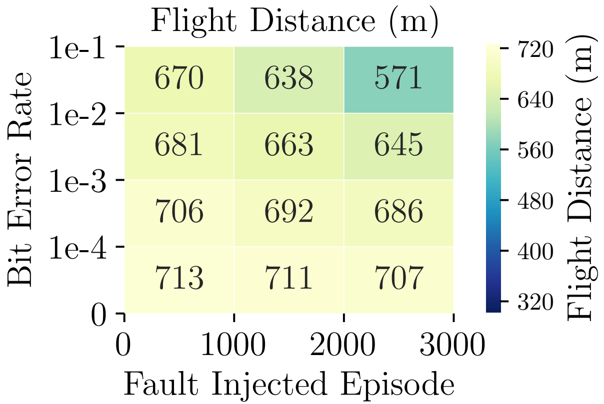


FRL system: faults in server



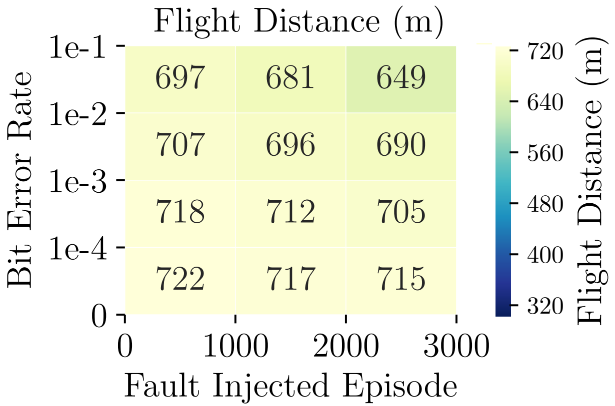


Single-agent system

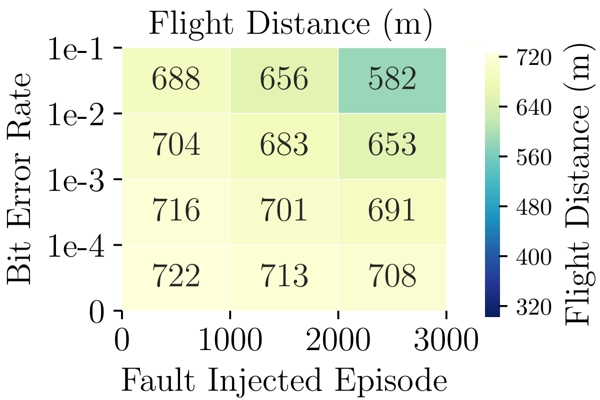


Training – Single and Multi-Drone Comparison

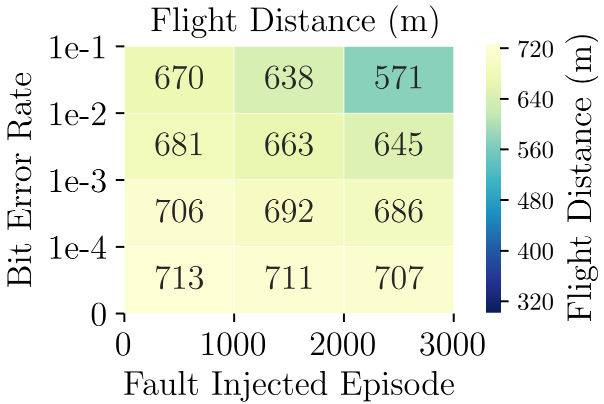
FRL system: faults in agent



FRL system: faults in server



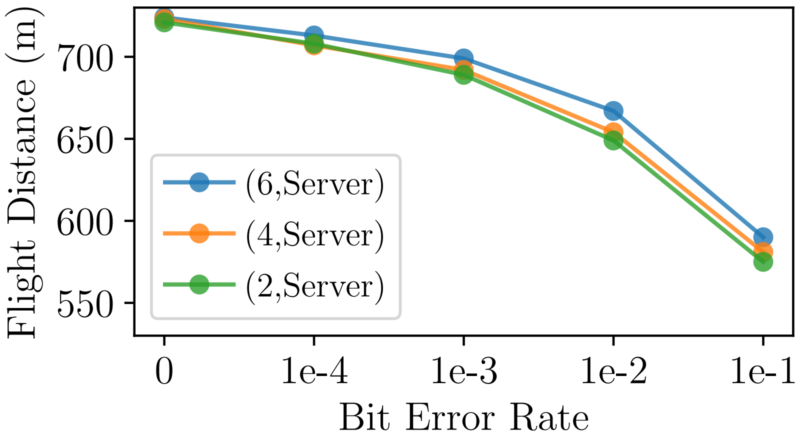
Single-agent system

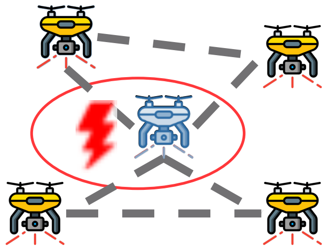


- **Multi-drone FRL system exhibit higher performance and resilience than single-drone system.**

Training – Number of Drones

(the higher, the better)



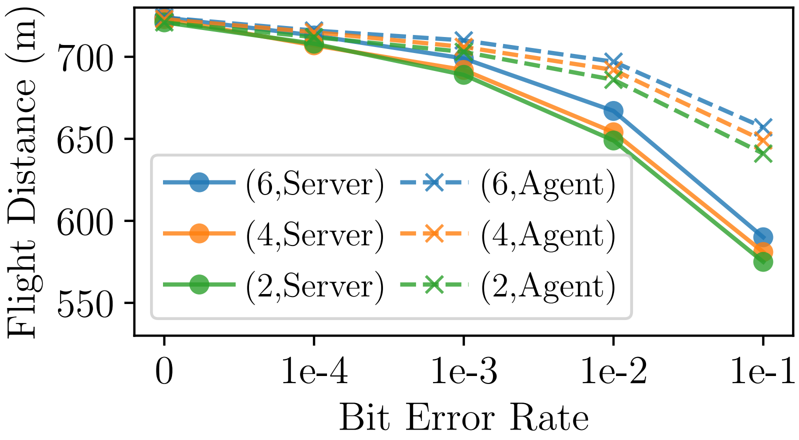


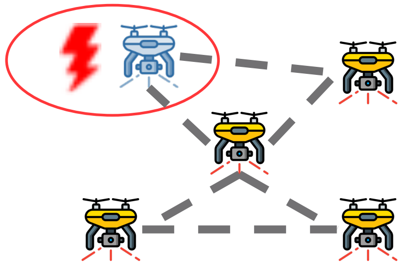
(6/4/2, Server):
(Total number of drones, faults in server)

- **More drones helps improve FRL system resilience**

Training – Number of Drones

(the higher, the better)



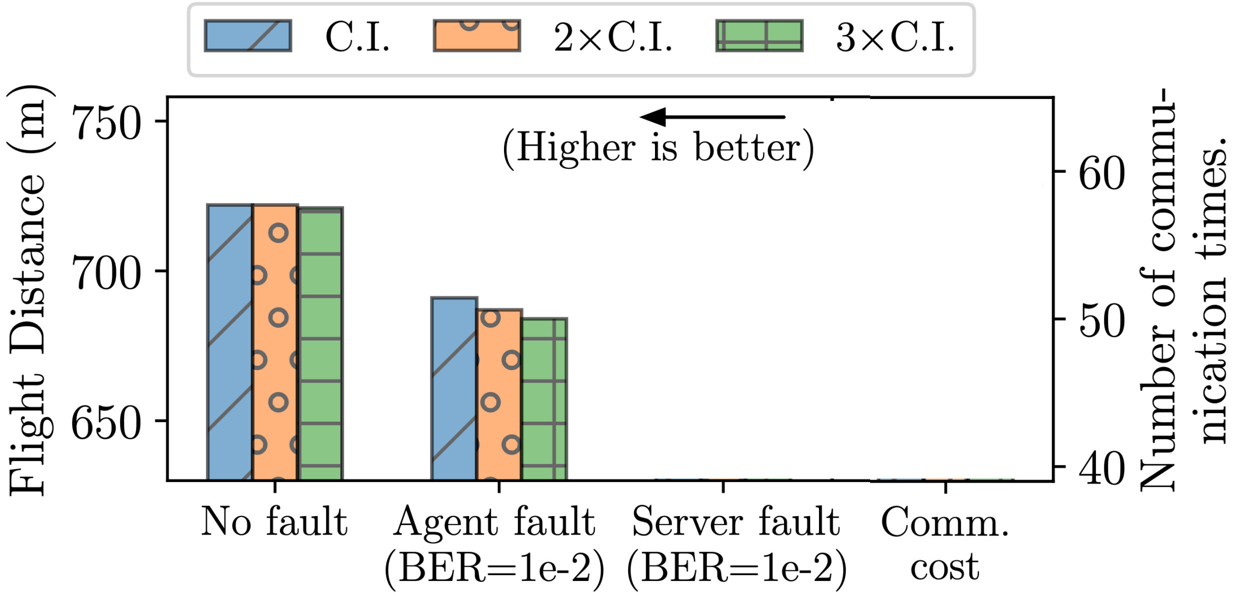


(6/4/2, Agent):
(Total number of drones, faults in agent)

- **More drones helps improve FRL system resilience**

Training – Communication Interval

(C.I.: Communication Interval)



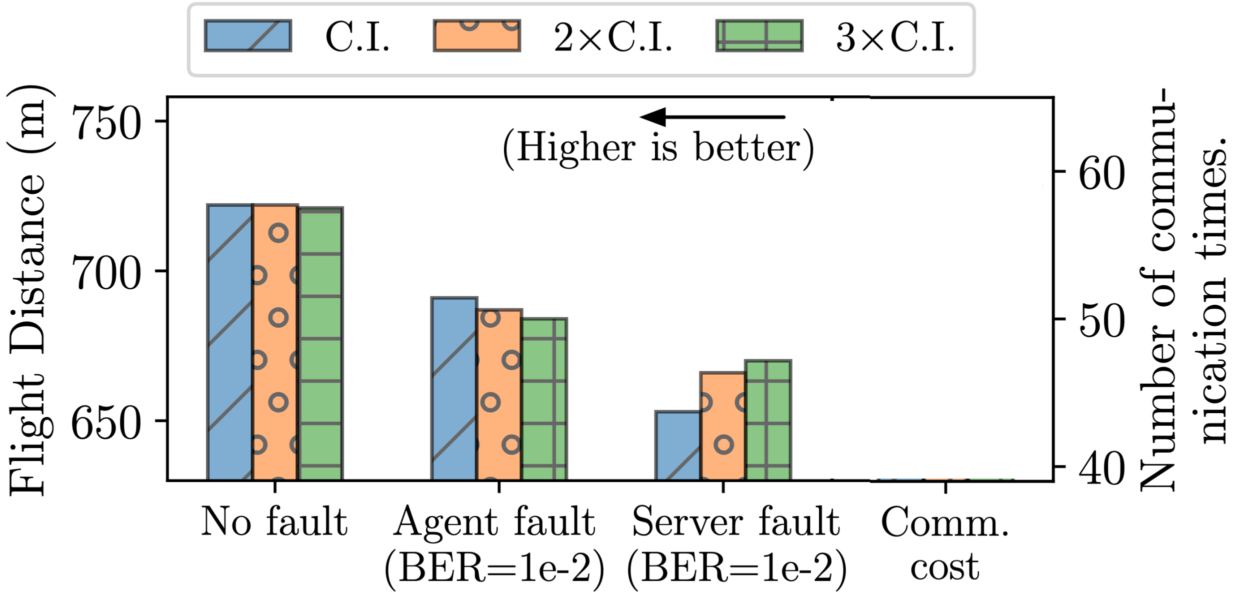
After 2000th episode, drone perform more exploitation (policy is almost not updated)

We increase the communication interval between agent and by 2x or 3x

- **Longer communication interval makes FRL system more vulnerable to agent faults**

Training – Communication Interval

(C.I.: Communication Interval)



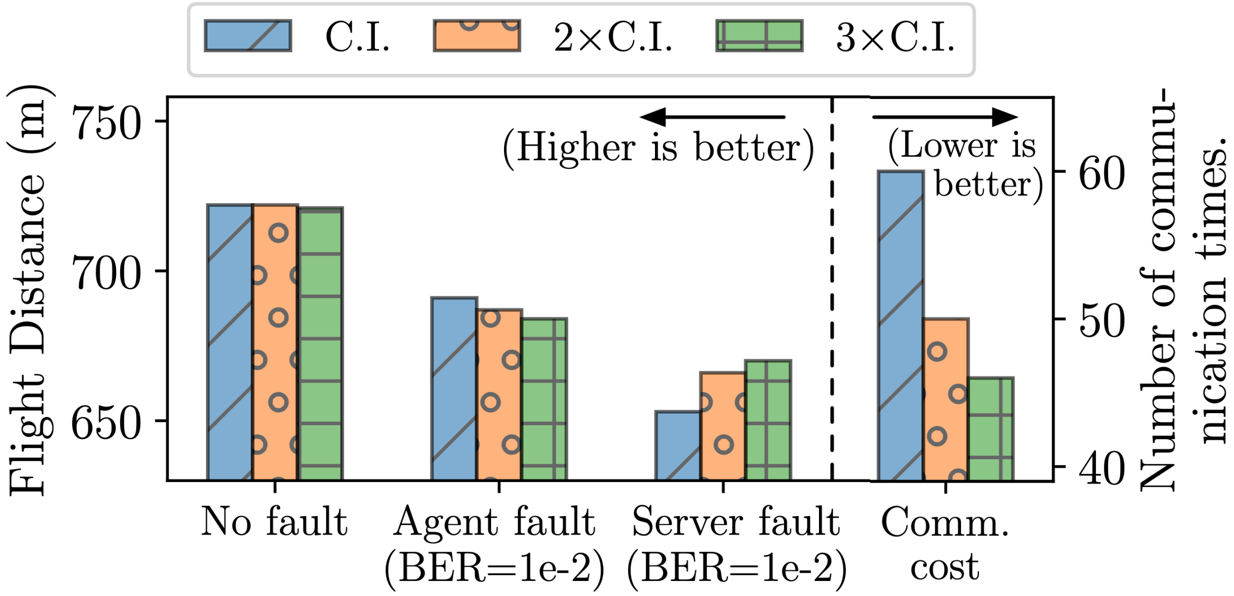
After 2000th episode, drone perform more exploitation (policy is almost not updated)

We increase the communication interval between agent and by 2x or 3x

- **Longer communication interval alleviates impact of server fault**

Training – Communication Interval

(C.I.: Communication Interval)

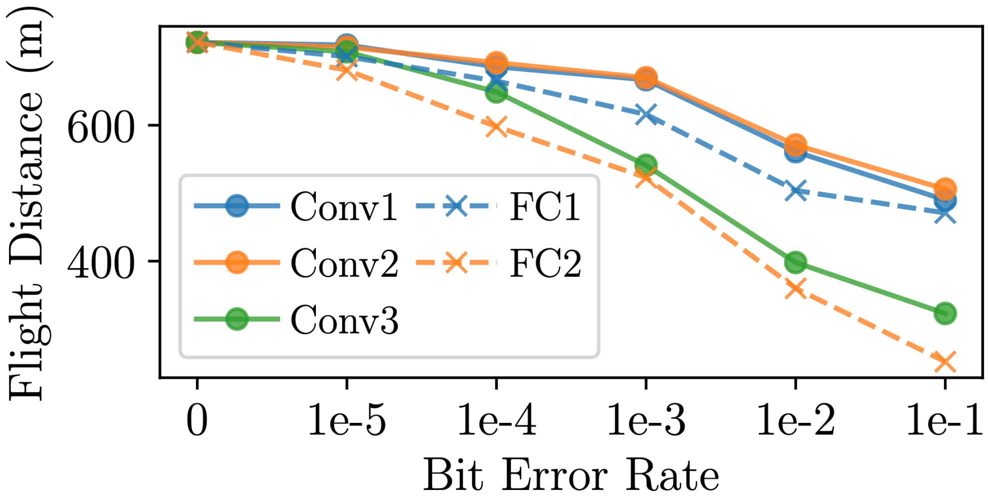


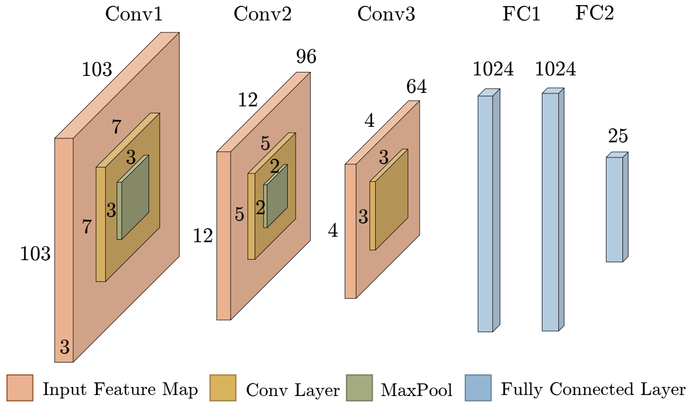
After 2000th episode, drone perform more exploitation (policy is almost not updated)

We increase the communi-cation interval between agent and by 2x or 3x

- **Longer communication interval reduce communication cost**

Inference – Layer Type





- **Pooling and activation operations make layers more robust since bit-flips have higher probability of being masked and ceased propagation**

This Work

Transient Fault Analysis for Federated Reinforcement Learning (FRL) -Based Navigation Systems

Transient fault injection for FRL-based navigation systems

Transient fault characterization for FRL-based navigation systems

Transient fault mitigation for FRL-based navigation systems

Training – Server Checkpointing

- **Fault Indicator**: Cumulative reward drop exceeds *x*% within *y* consecutive episodes (*x* and *y* are parameters)

Fault in Agent

- Detection: agent *i* reward drop

Fault in Server

- Detection: more than half of agents reward drop

Training – Server Checkpointing

- **Fault Indicator**: Cumulative reward drop exceeds *x*% within *y* consecutive episodes (*x* and *y* are parameters)
- **Fault Mitigation**: save checkpoint in server and update every 5 communication intervals

Fault in Agent

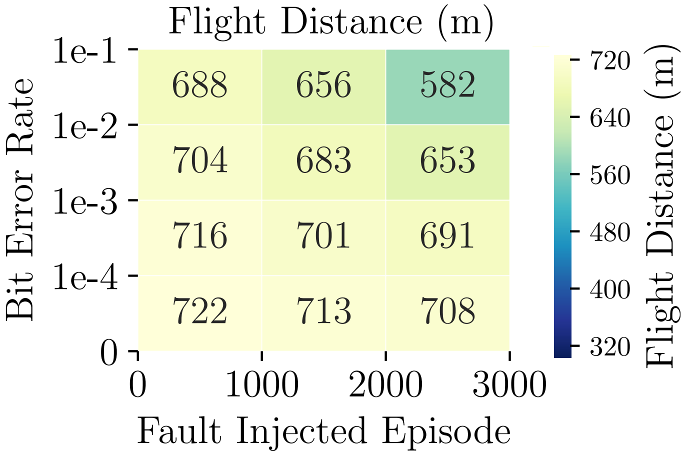
- Detection: agent *i* reward drop
- Recovery: copy server checkpoint to agent *i* memory

Fault in Server

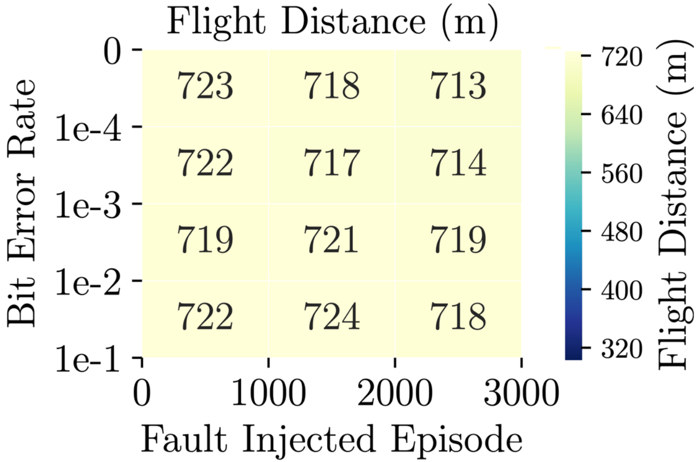
- Detection: more than half of agents reward drop
- Recovery: copy server checkpoint to server memory

Training – Server Checkpointing

- **Evaluation**



After fault injection



After fault mitigation

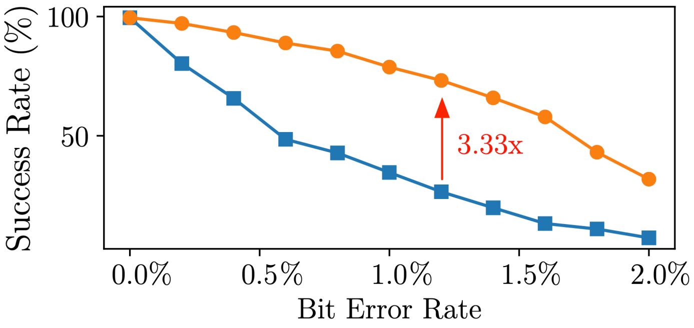
- **The impact of transient fault during training can be alleviated**

Inference – Range-Based Anomaly Detection

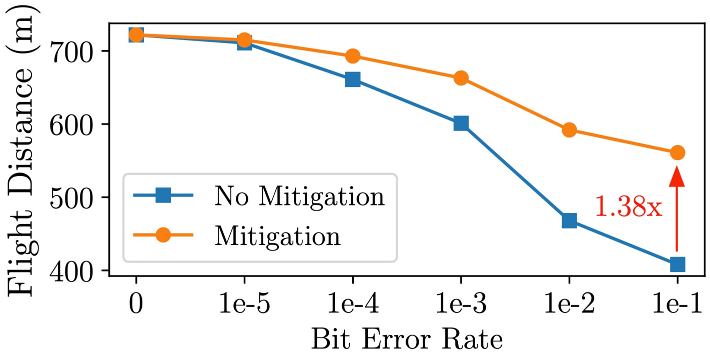
- **Detection: Statistically anomaly detection, (a*i*, b*i*) -> (1.1a*i*, 1.1b*i*)**
- **Recovery: skip faulty operations**

Inference – Range-Based Anomaly Detection

- **Detection: Statistically anomaly detection, (a*i*, b*i*) -> (1.1a*i*, 1.1b*i*)**
- **Recovery: skip faulty operations**
- **Evaluation:**



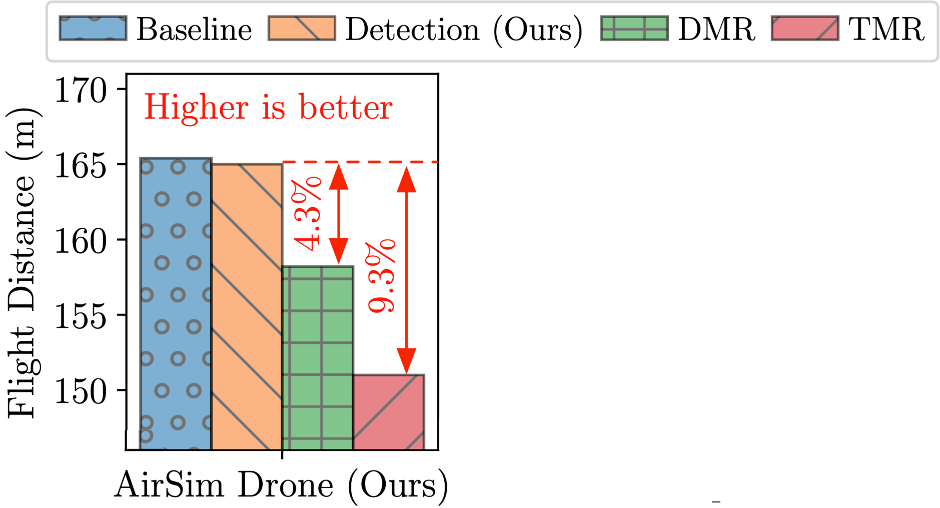
GridWorld



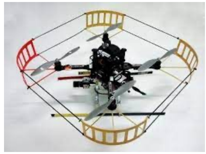
Drone autonomous navigation

Inference – Range-Based Anomaly Detection

- **Compute overhead**



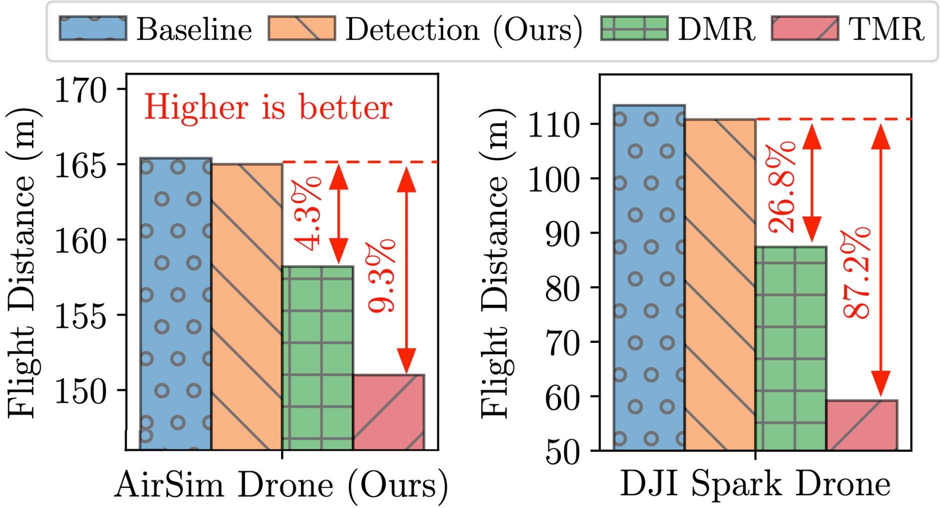
	AirSim Drone
Type	mini-UAV
Size (mm)	650
Weight (g)	1652
Battery Capacity (mAh)	6250



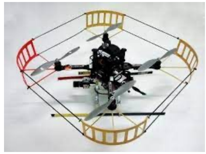
- **Using a drone roofline-like performance model** [Krishnan et al, CAL 2020]
- **Negligible overhead compared to DMR, TMR**

Inference – Range-Based Anomaly Detection

- **Compute overhead**



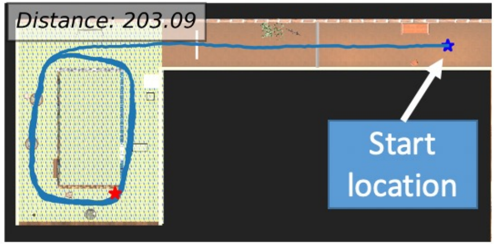
	AirSim Drone	DJI Spark
Type	mini-UAV	micro-UAV
Size (mm)	650	170
Weight (g)	1652	300
Battery Capacity (mAh)	6250	1480



- **Using a drone roofline-like performance model** [Krishnan et al, CAL 2020]
- **Negligible overhead compared to DMR, TMR**

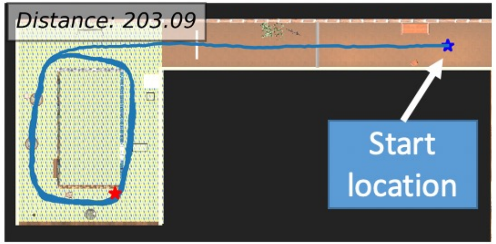
Drone Flight Trajectory Demo

No Fault:

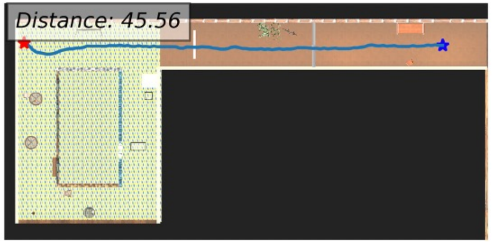


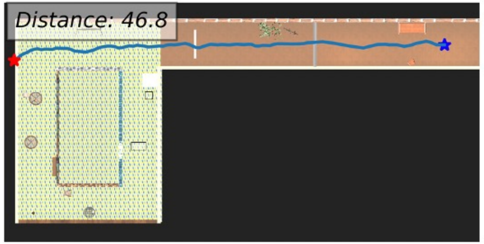
Drone Flight Trajectory Demo

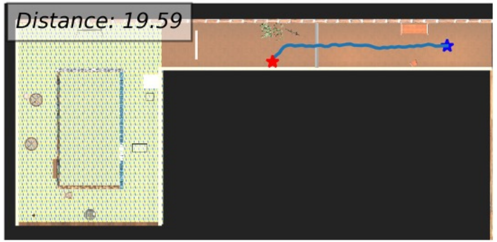
No Fault:



Fault injection:

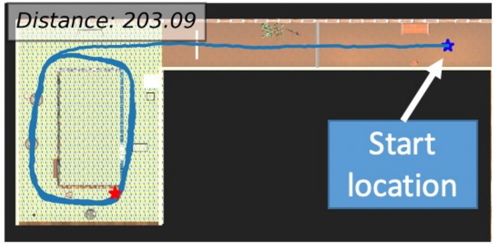




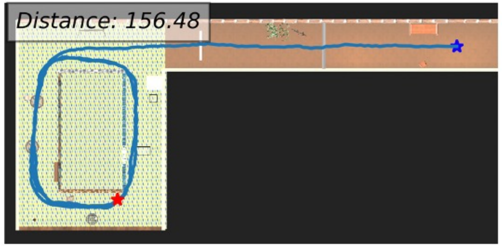


Drone Flight Trajectory Demo

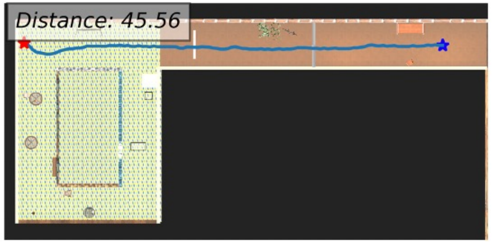
No Fault:

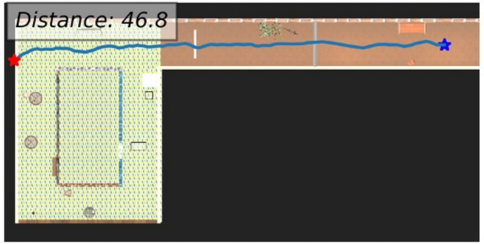


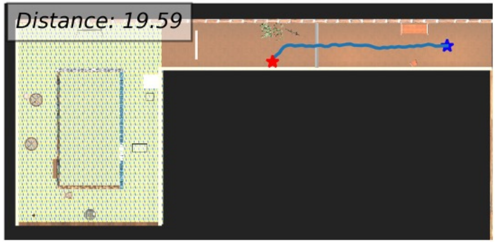
Fault mitigation:



Fault injection:







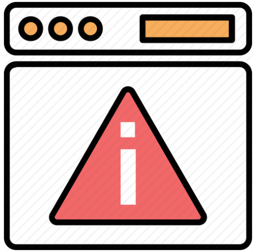
In This Talk

Transient Fault Analysis for Federated Reinforcement Learning (FRL) -Based Navigation Systems

The **safety and reliability** of **swarm intelligence navigation systems** is important, but not well understood

A **fault injection** framework that emulates **hardware faults** and enables rapid fault analysis of FRL systems

Large-scale **fault injection study** in both training and inference stages of FRL systems against transient faults



Low-overhead **fault detection and recovery techniques** for both training and inference