# On the Role of Fidelity in the Safety Evaluation of Learning-Based Autonomous Systems

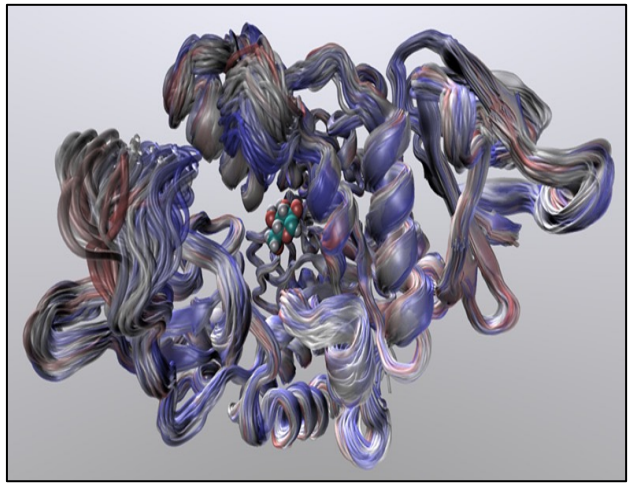
Ali Baheri

RIT Graduate Seminar

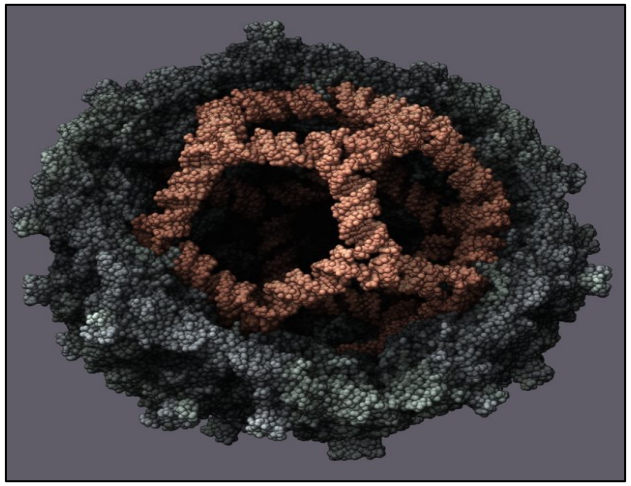
Feb. 2023

Images and icons are obtained from Google search.

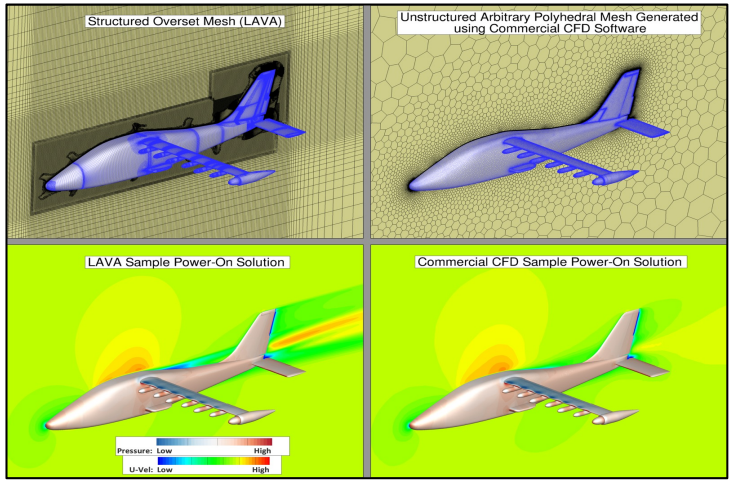
# Simulators are everywhere



Simulators & molecular design

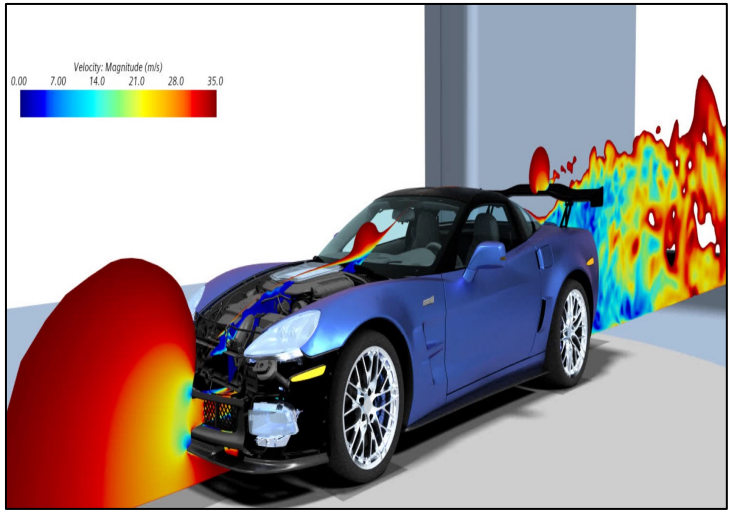


Simulators & biology



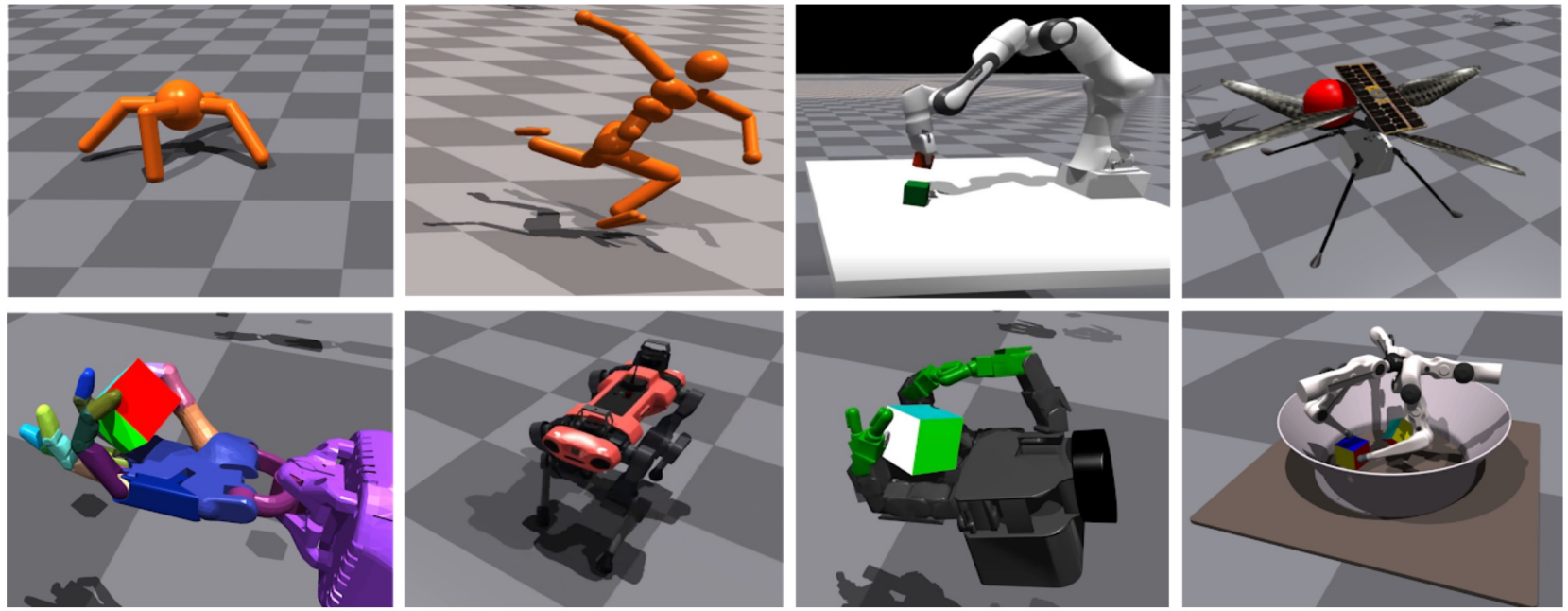

Simulators & aerodynamics

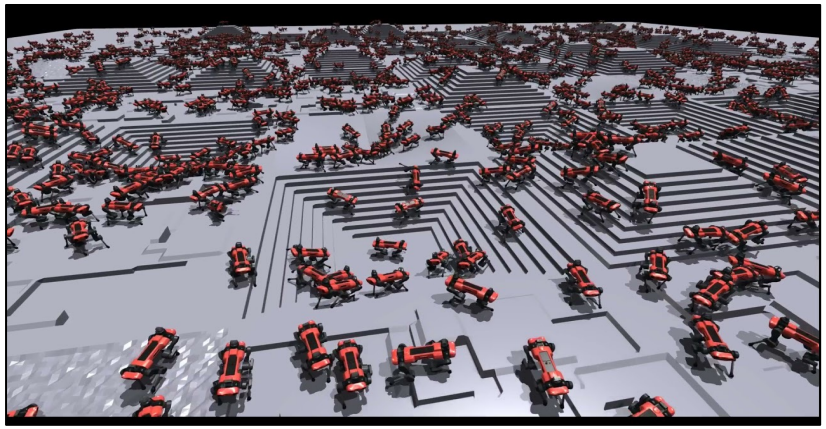

Simulators & healthcare




Simulators & CFD

# Simulators in robotics / control





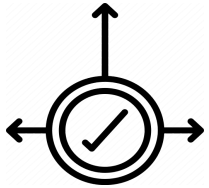

# What are the pros and cons of simulators?

**Pros:**

cost and time efficiency

risk reduction

flexibility



improved decision-making:

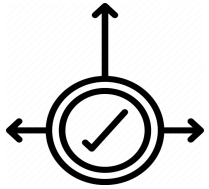
# What are the pros and cons of simulators?

**Pros:**

cost and time efficiency

risk reduction

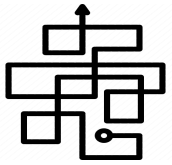
flexibility



improved decision-making:

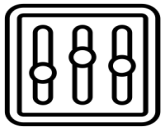
**Cons:**

modeling assumptions



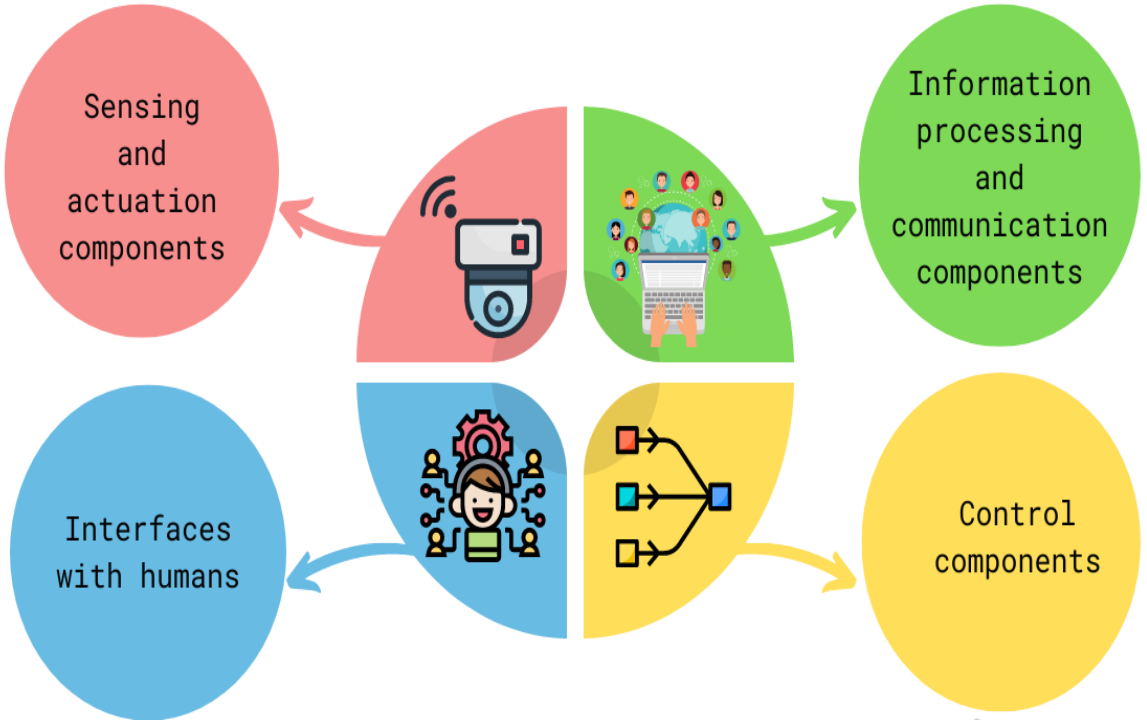
complexity

time delay



sensitivity to parameters

# Role of simulators in CPS and learning-based control

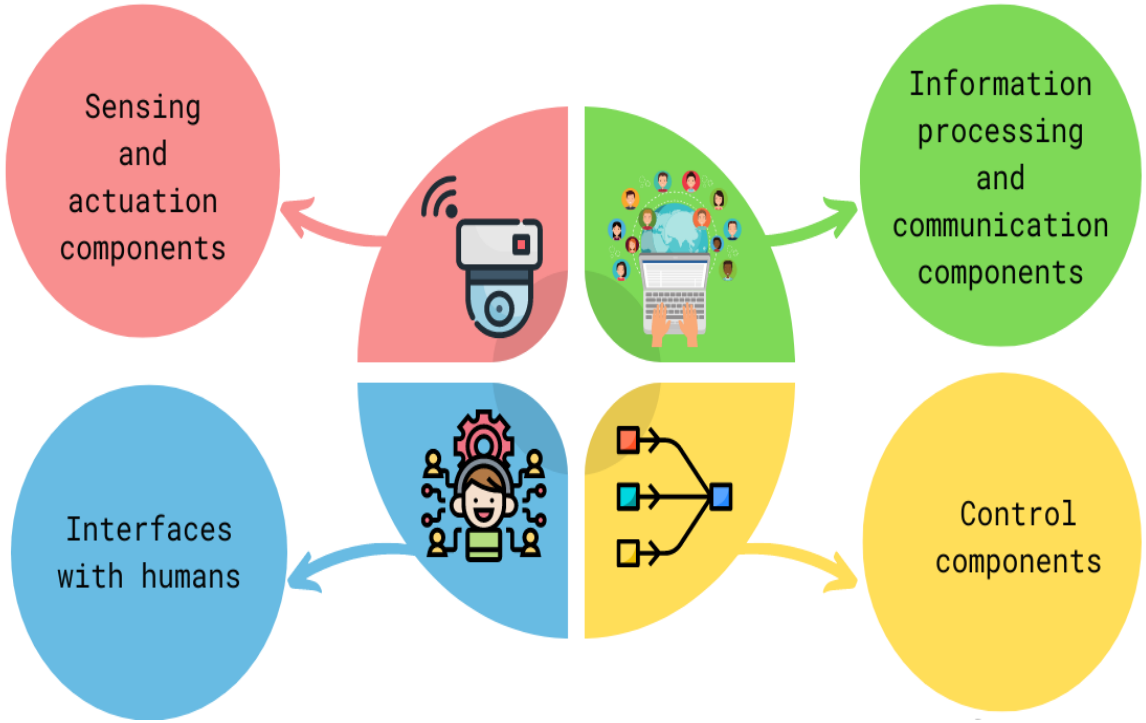



A Cyber-Physical System (CPS) is a system that integrates physical components and processes with computer systems and software to enable real-time control and coordination.

# Role of simulators in CPS and learning-based control

There are several benefits of using simulators in CPS and learning-based control:

- reduced risk
- reduced development time and cost
- improved design and optimization
- increased understanding of complex systems
- advanced testing and safety validation

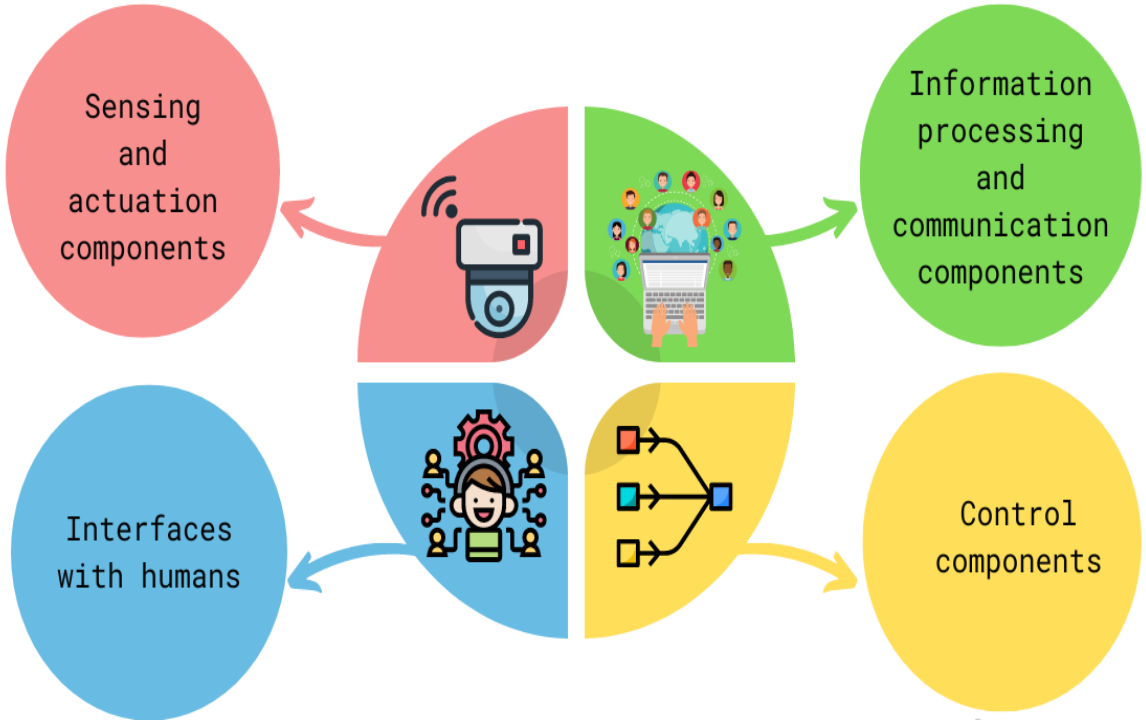



A Cyber-Physical System (CPS) is a system that integrates physical components and processes with computer systems and software to enable real-time control and coordination.

# Role of simulators in CPS and learning-based control

There are several benefits of using simulators in CPS and learning-based control:

- reduced risk
- reduced development time and cost
- improved design and optimization
- increased understanding of complex systems
- advanced testing and safety validation




A Cyber-Physical System (CPS) is a system that integrates physical components and processes with computer systems and software to enable real-time control and coordination.

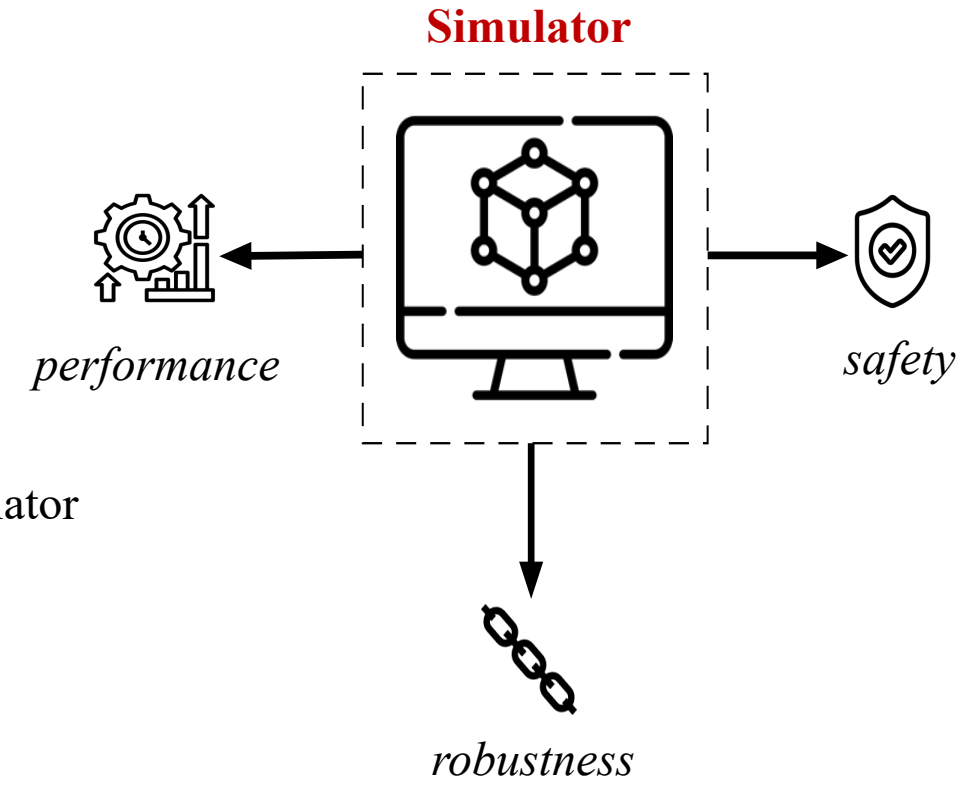
# Simulators and safety validation of learning-based systems

Simulators can be used to evaluate the *performance, safety, and robustness* of a learning-based control system through testing under various scenarios and conditions.

**Key challenges:**

**compute cost:**
high-fidelity simulators can be computationally expensive

**accuracy:**
low-fidelity simulators may not be as accurate as high-fidelity simulator



# Fidelity in simulators

- fidelity in simulation refers to the degree of realism and accuracy with which a simulation represents a real-world system or process.

- fidelity measures how closely the simulation models the behavior and interactions of the physical components, processes, and environment of the real-world system.

# Fidelity in simulators

- fidelity in simulation refers to the degree of realism and accuracy with which a simulation represents a real-world system or process.

- fidelity measures how closely the simulation models the behavior and interactions of the physical components, processes, and environment of the real-world system.

**Key research question:**

Is there an *optimal* trade-off between high-fidelity and low-fidelity simulators in the safety validation of learning-based control systems?

# Simulation-based testing using optimization

**Goal:** finding counterexamples (i.e., failure scenarios) of a learning-based controller under uncertainty.

# Simulation-based testing using optimization

**Goal:** finding counterexamples (i.e., failure scenarios) of a learning-based controller under uncertainty

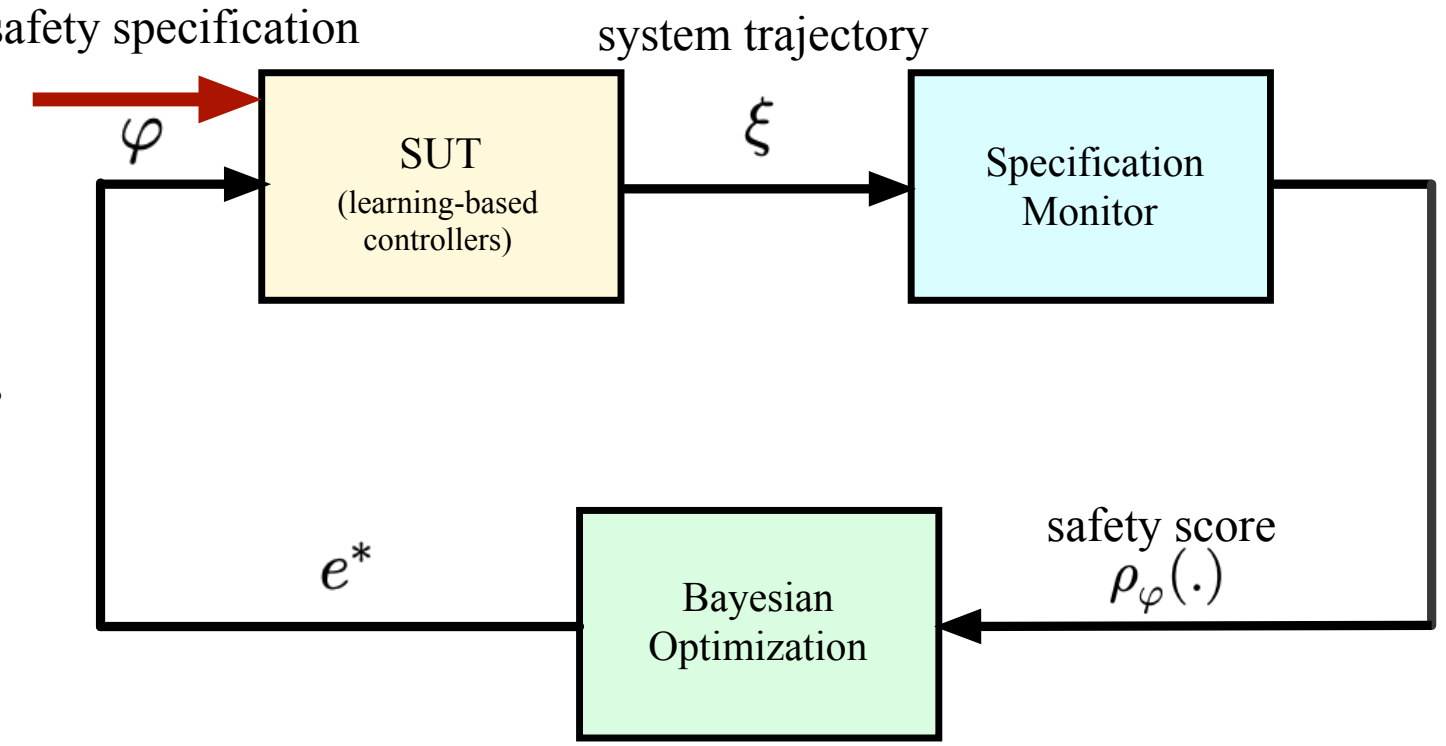
**Assumptions:**

$$\varphi := \mu | \neg\mu | \varphi \wedge \psi \mid \varphi \vee \psi$$

- $\varphi$ is evaluated on the system trajectories

- we use the specification robustness:

$$\rho_\varphi : \mathcal{E} \rightarrow \mathbb{R}$$



# Simulation-based testing for a learning-based system using Bayesian optimization

**Goal:** we would like to determine whether there exists a counterexample where the specification is violated, i.e., $\rho_{\varphi}(\mathbf{e}) < 0$

Optimization formulation:

$$\underset{\mathbf{e}}{\operatorname{argmin}}\, \rho_{\varphi}(\mathbf{e})$$

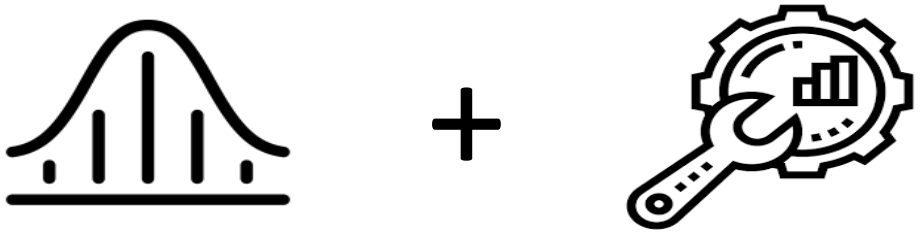
# Simulation-based testing for a learning-based system using Bayesian optimization

**Goal:** we would like to determine whether there exists a counterexample where the specification is violated, i.e., $\rho_\varphi(\mathbf{e}) < 0$

Optimization formulation:

$$\underset{\mathbf{e}}{\operatorname{argmin}}\, \rho_\varphi(\mathbf{e})$$

We can use *Bayesian optimization (i.e.,* a *probabilistic tuning* technique) to solve this problem.



**We have a *chain of simulators* that range from low-fidelity to high-fidelity. How can we use the information from these simulators to accelerate the safety validation process?**

- **Challenge 1:** how do we model information across different simulators?
- **Challenge 2:** how can we determine which simulator to perform a function evaluation on?

# Challenge 1: multi-fidelity modeling

- relation between robustness values on different simulators

$$\rho_{\varphi}^{i}(\mathbf{e}) = \eta_i \rho_{\varphi}^{i-1}(\mathbf{e}) + \rho_{gap}^{i}(\mathbf{e})$$

- $\eta_i$ constant regression parameter indicates the magnitude of the correlation between the fidelities $\rho_{gap}^{i}(\mathbf{e})$ bias term / independent GP with $m_{gap}^{i}$ mean function and $k_{gap}^{i}(\mathbf{e}, \mathbf{e}')$ kernel function

- prior distribution

$$\begin{bmatrix} \rho_{\varphi}^{i-1} \\ \rho_{\varphi}^{i} \end{bmatrix} \sim GP\left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k_{i-1} & \eta_i k_{i-1} \\ \eta_i k_{i-1} & \eta_i^2 k_{i-1} + k_{gap}^{i} \end{bmatrix} \right)$$

# Challenge 2: multi-fidelity optimization

- Assume there is a chain of simulators with different levels of fidelity $\mathcal{S}_1, \ldots, \mathcal{S}_q$

**- Goal:** our goal is to efficiently determine the minimum specification robustness value of the highest-fidelity simulator through querying all simulators, utilizing *fewer* experiments.

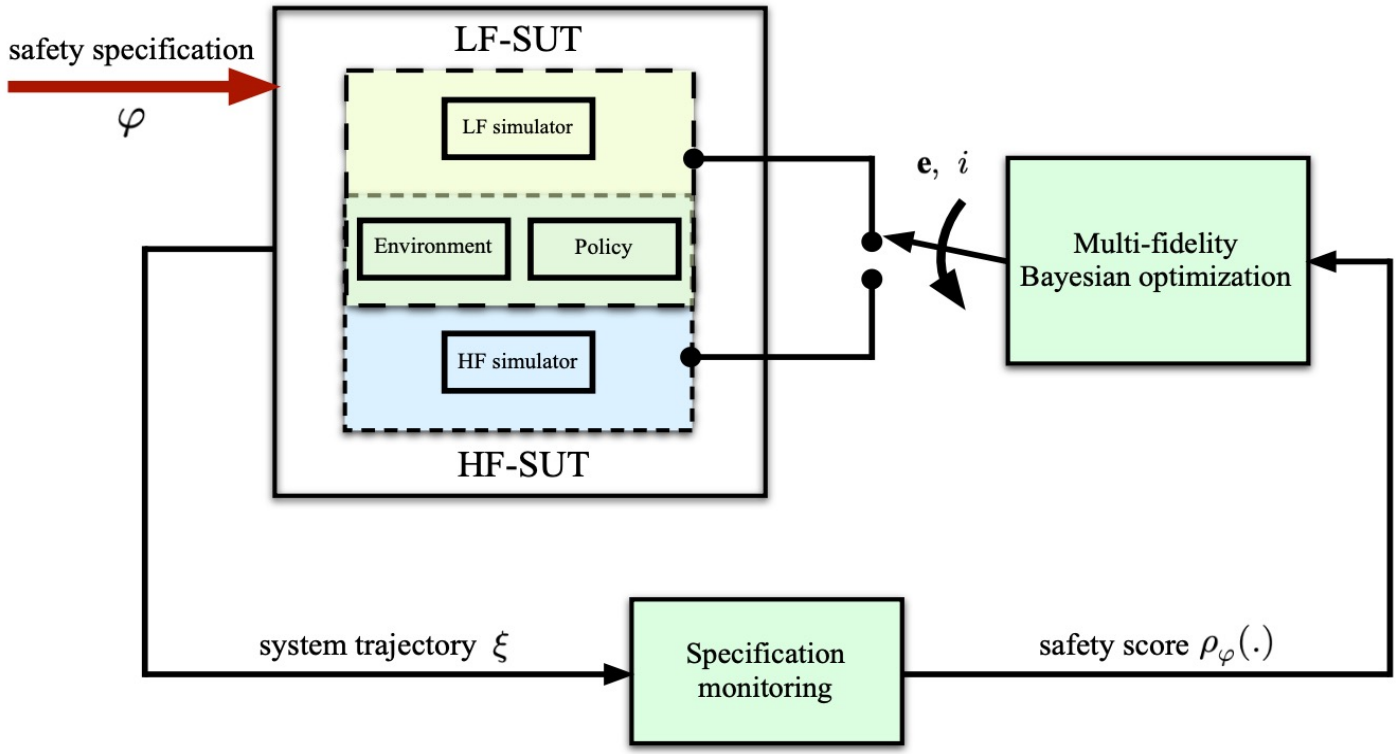
- **Proposed solution:** Bayesian optimization with modified entropy search

$$\mathbf{e}_n, i = \underset{\mathbf{e}\in\mathcal{E}, i\in\{1,\ldots,q\}}{\operatorname{argmax}} \alpha_i^{\mathrm{ES}}(\mathbf{e})/\lambda_i$$

# Simulation-based testing by multi-fidelity Bayesian optimization

- select the environment configuration ***along*** with the *fidelity level* which we should perform the next experiment

- monitor trajectories on a low-fidelity simulator or a high-fidelity simulator to find counterexamples or otherwise verify the system



# How do we choose fidelity settings?

**Scenario 1.** *sensor fidelity:* the difference between the low-fidelity simulator and the high-fidelity simulator is due to measurement errors in sensor data.

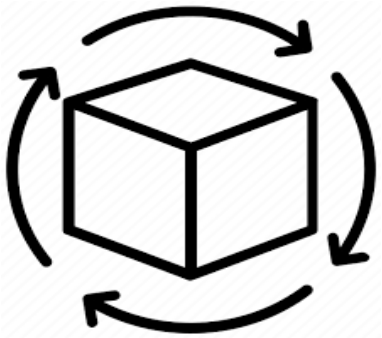
**Scenario 2.** *precision fidelity*: the fidelity difference could be the precision of the simulator's states.

# How do we choose fidelity settings?

**Scenario 1.** *sensor fidelity:* the difference between the low-fidelity simulator and the high-fidelity simulator is due to measurement errors in sensor data.

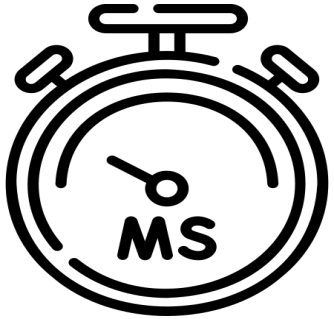
**Scenario 2.** *precision fidelity*: the fidelity difference could be the precision of the simulator's states.

**Other types of fidelity settings** (e.g., for an autonomous driving case study):



environment model fidelity

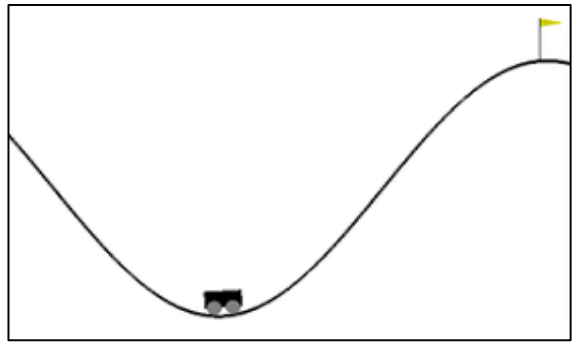
traffic model fidelity




latency fidelity

# Results

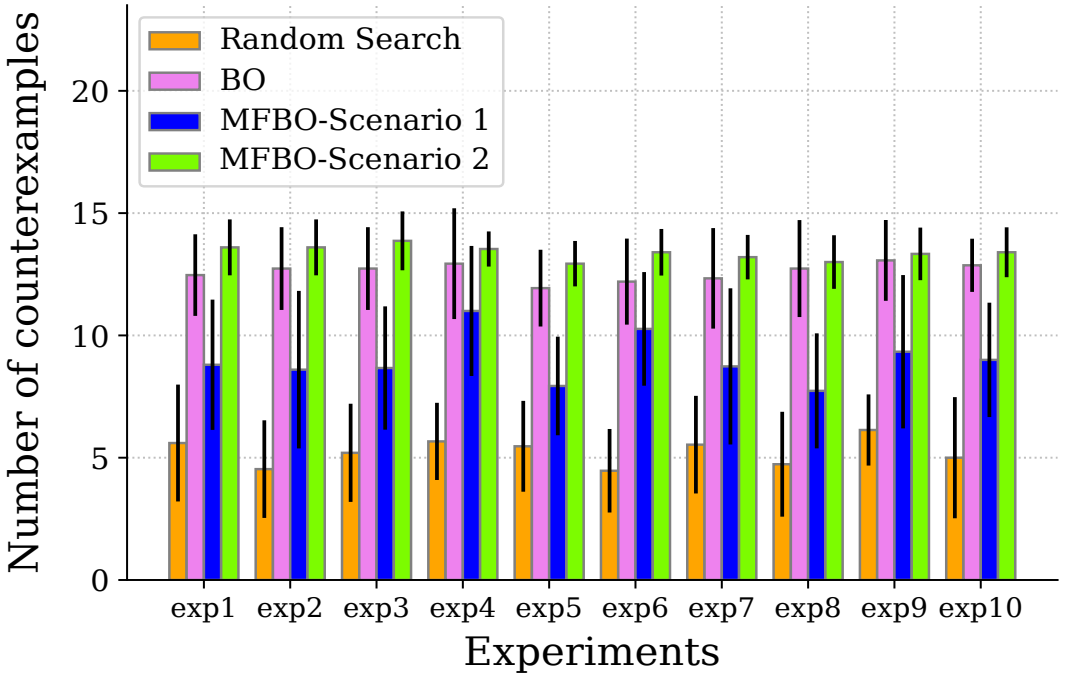
## *Case Study 1.* Mountain Car environment

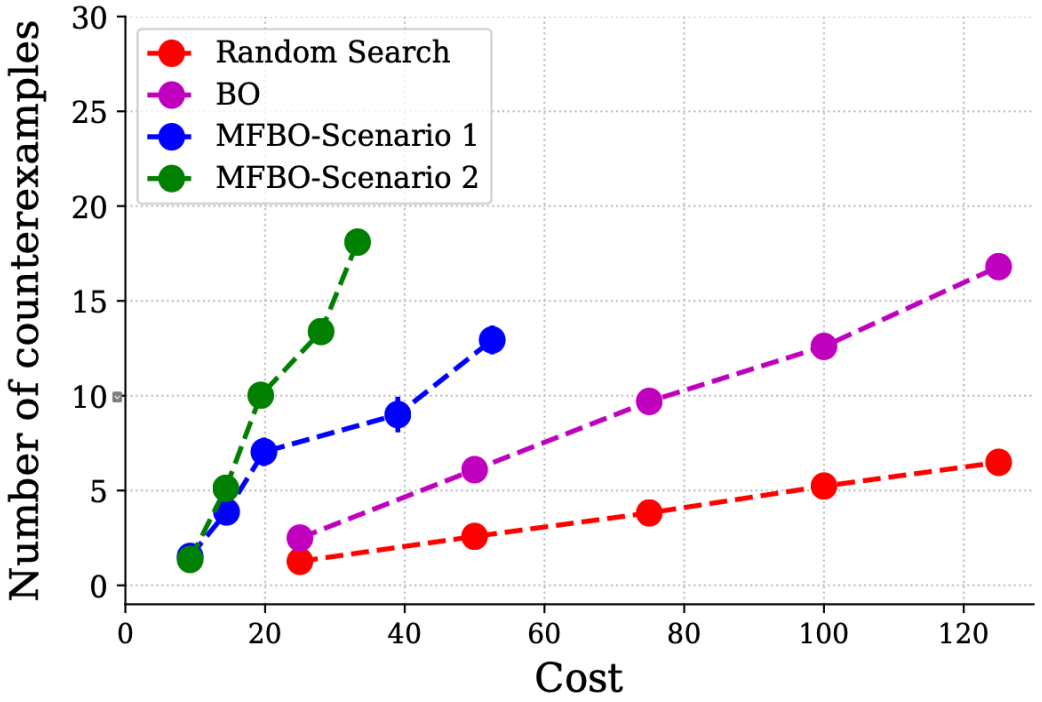


- **uncertainty space:**
  - initial position and velocity are $[-0.6, -0.4]$ and $[-0.003, 0.003]$.
  - goal position $[0.4, 0.6]$
  - maximum speed $[0.055, 0.075]$
  - maximum power magnitude $[0.0005, 0.0025]$
- given an instance of $\mathbf{e} \in \mathcal{E}$, the system's trajectory $\xi = (x(t), v(t), \theta(t), \dot{\theta}(t))$ is uniquely defined.
- **policy:** PPO
- **specification:**
  - the car will be on a safe trajectory if it either reaches the goal soon or does not deviate too much from its initial location.
  - we require that the car always maintain its velocity in $[-0.04, +0.04]$.
  - 25 BO iterations with 15 random seeds

# Results

We compare between MFBO, standard BO on the high-fidelity simulator, and random search.



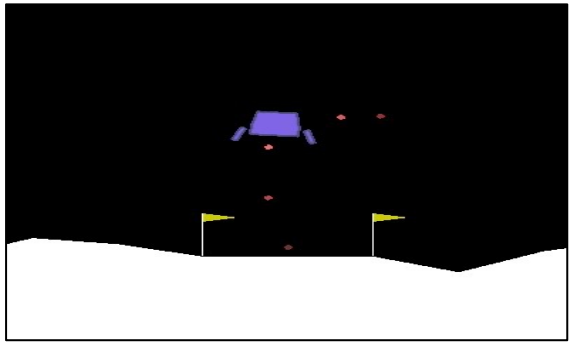






*Submitted to ECC 2023*

# Results

## *Case Study 2.* Lunar Lander environment



- **uncertainty space:**
  - coordinate perturbations $\delta_x \in [-0.5, 0.5]$ and $\delta_y \in [0, 3]$
  - velocities $\delta_{vx} \in [-2, 2]$ and $\delta_{vy} \in [0, 2]$

- given an instance of $\mathbf{e} \in \mathcal{E}$, the system's trajectory $\xi = \left(x(t), y(t), v_x(t), v_y(t), \theta(t), \dot{\theta}(t)\right)$ is uniquely defined.

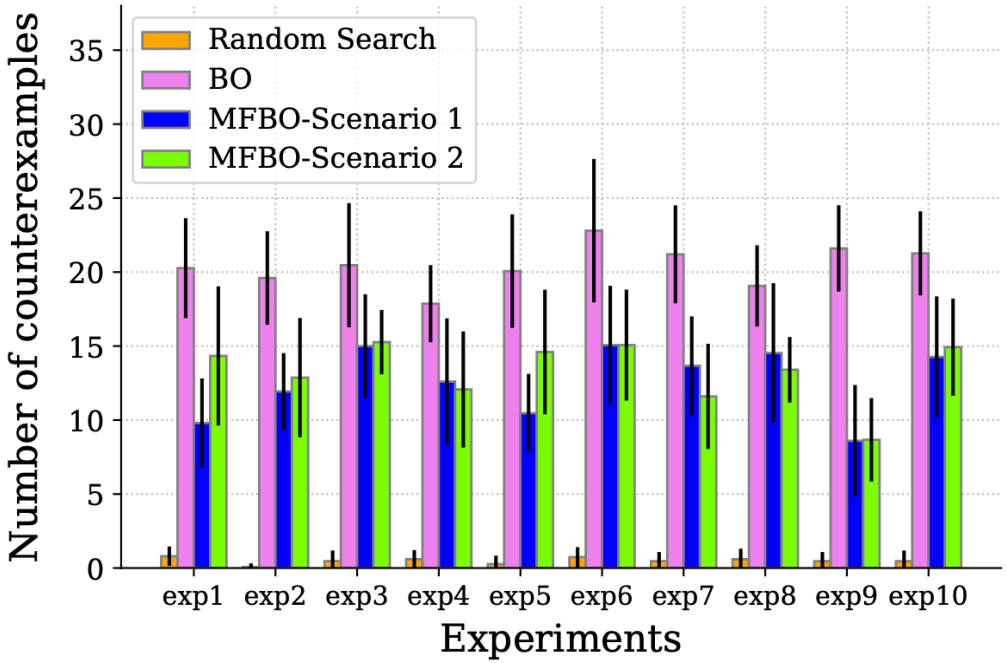
- **policy:** DDPG

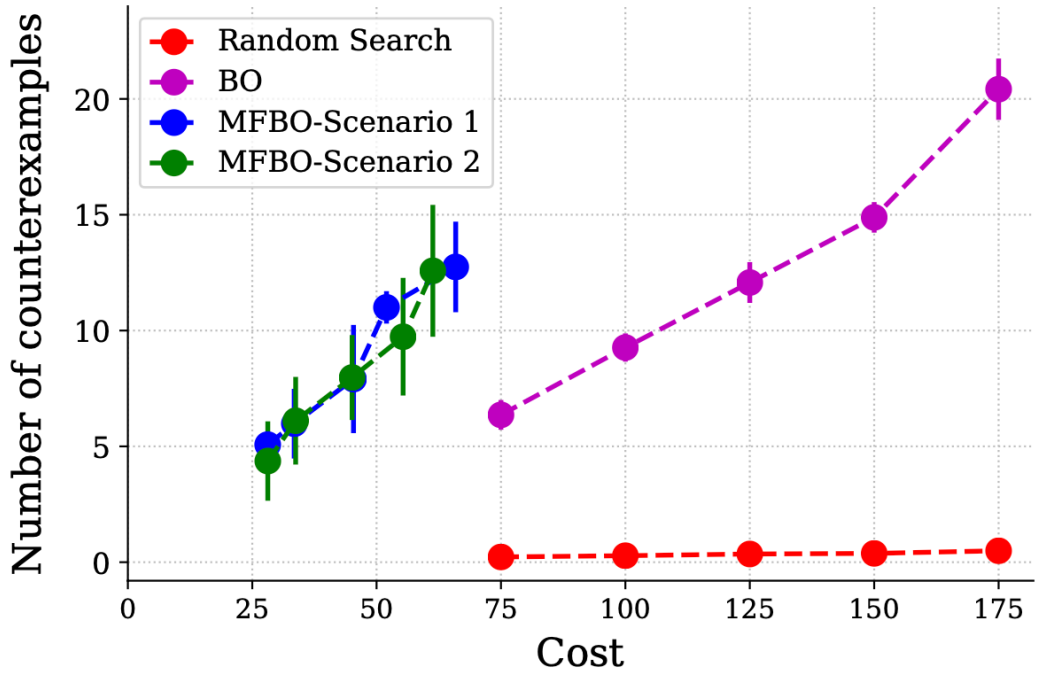
- **specification:**
  - the lander must always maintain its horizontal coordinate close to the origin, with a tilt angle not exceeding $\pi/4$ and a rotation rate no greater than 0.2 radians per second.

- 35 BO iterations with 15 random seeds

# Results

We compare between MFBO, standard BO on the high-fidelity simulator, and random search.



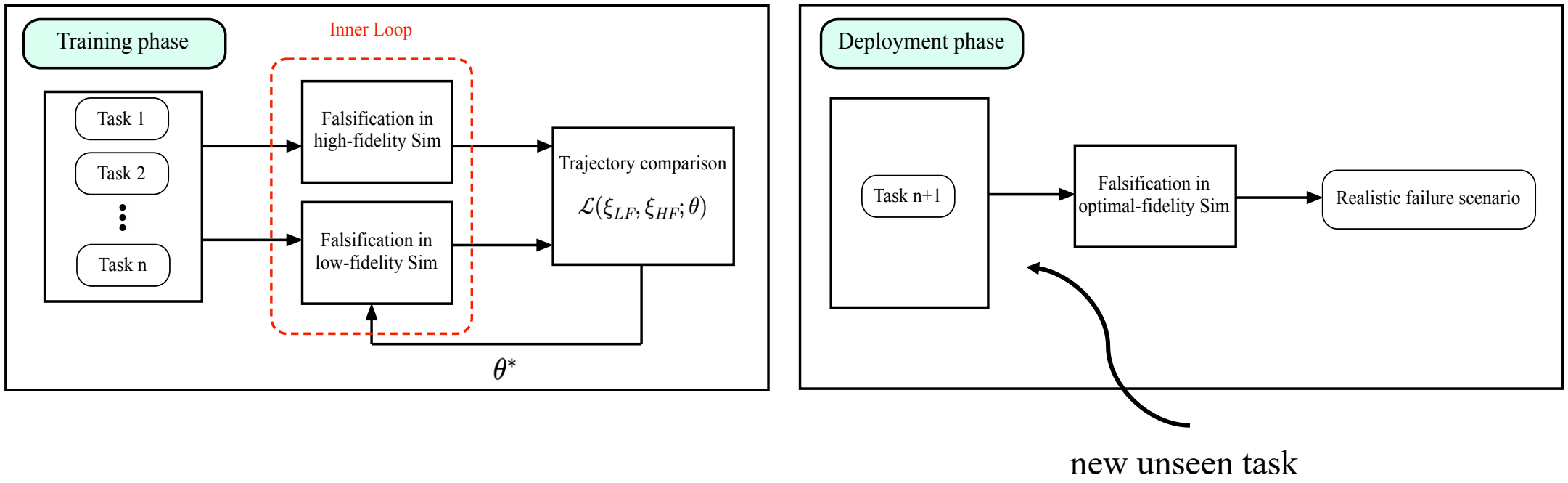



*Submitted to ECC 2023*

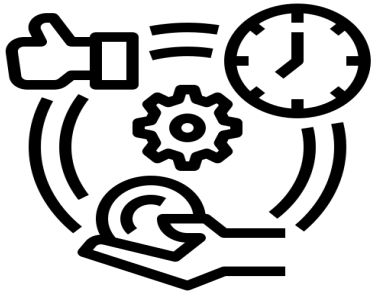
# Ongoing work:
# Can we find the optimal fidelity settings?

- Can we find the optimal fidelity settings for a low-fidelity simulator that closely *replicates* the behavior of a high-fidelity simulator, while also effectively identifying potential failure scenarios in a safety-critical system?

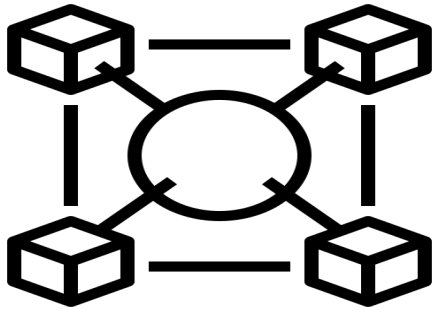
- Can we discover *realistic* failure scenarios for a new, unseen task using the ***optimal fidelity*** simulator?



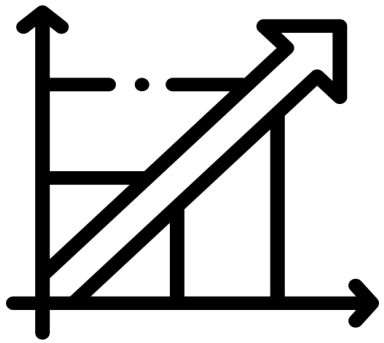
# Why should we learn optimal fidelity settings?



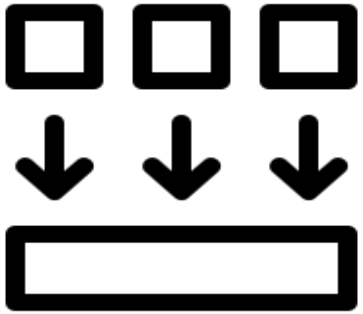
Improved efficiency



Better interpretability



Improved scalability

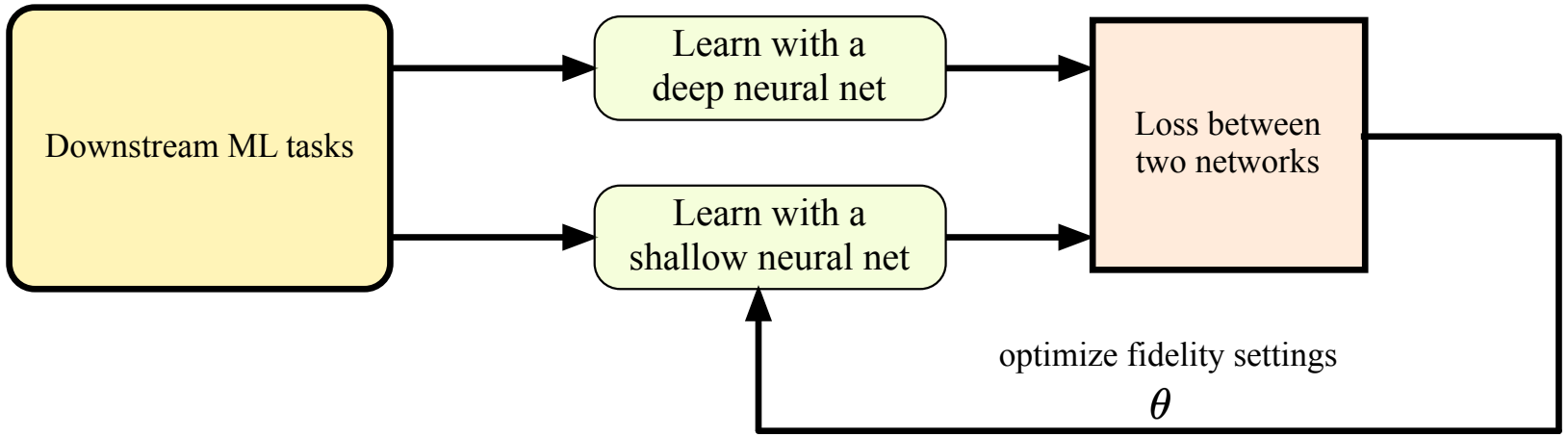


Better generalization

# Ongoing work:
# Can we treat neural networks as simulators?

- Let us assume that a shallow neural network represents a low-fidelity simulator, while a deep neural network represents a high-fidelity simulator.

- Can we find the *optimal configuration* of the shallow neural network such that its performance replicates that of the deep neural network?



# Can we treat neural networks as simulators?

- Let us assume that a shallow neural network represents a low-fidelity simulator, while a deep neural network represents a high-fidelity simulator.

- Can we find the *optimal configuration* of the shallow neural network such that its performance replicates that of the deep neural network?

- Is the idea the same as hyperparameter optimization (HPO) or neural architecture search (NAS)? **NO!**

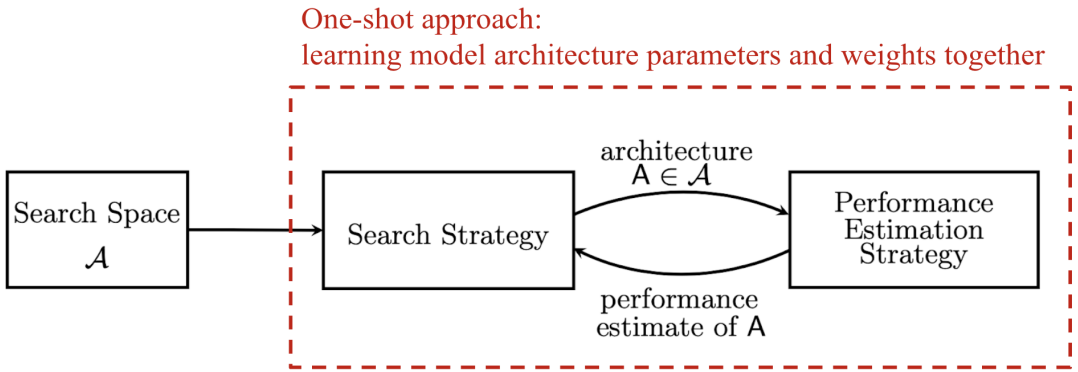
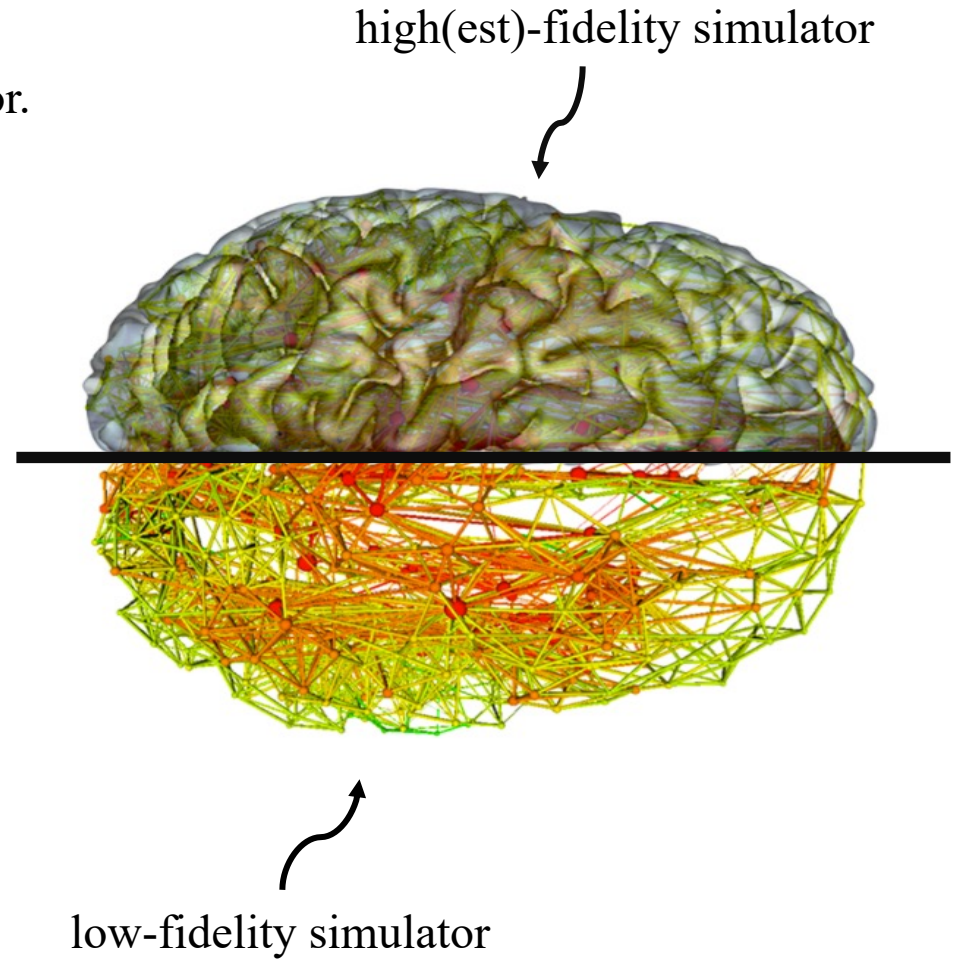



Image source: Elsken, et al. 2019



# Takeaways

- **Simulators** play a crucial role in learning-based systems by providing a virtual environment for testing and training control algorithms before implementation in real-world systems.

- A **multi-fidelity simulator** allows for the use of both high- and low-fidelity simulations, providing a trade-off between accuracy and computational efficiency while still enabling robust and effective learning.

- Finding **optimal fidelity setting** involves the process of determining the level of complexity in a simulation that balances the accuracy and computational efficiency of the simulation.

- The **treatment of neural networks as simulators** offers several potential benefits, including improved explainability, better generalization, and increased efficiency.

# Thank you!

**Safe AI Lab:** alibaheri@github.io

akbeme@rit.edu

We are always looking to motivated students. Feel free to reach out!

Thanks to my collaborator, student, and sponsors: