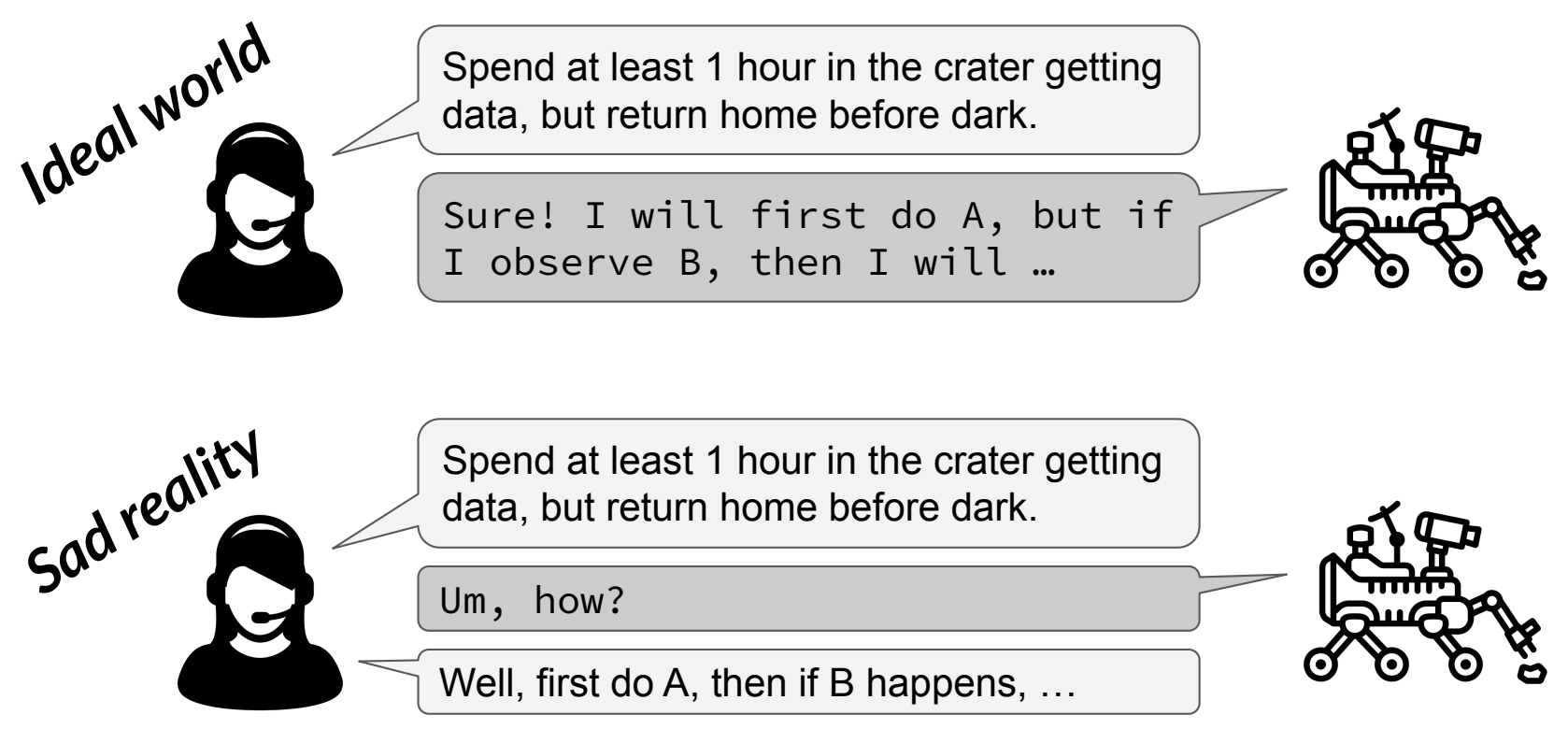
Robust Counterexample-guided Optimization for Planning from Differentiable Temporal Logic

PRESENTER:
Charles Dawson

BACKGROUND: There is a gap between how users would ***like*** to program our robots and what robots will accept in reality.

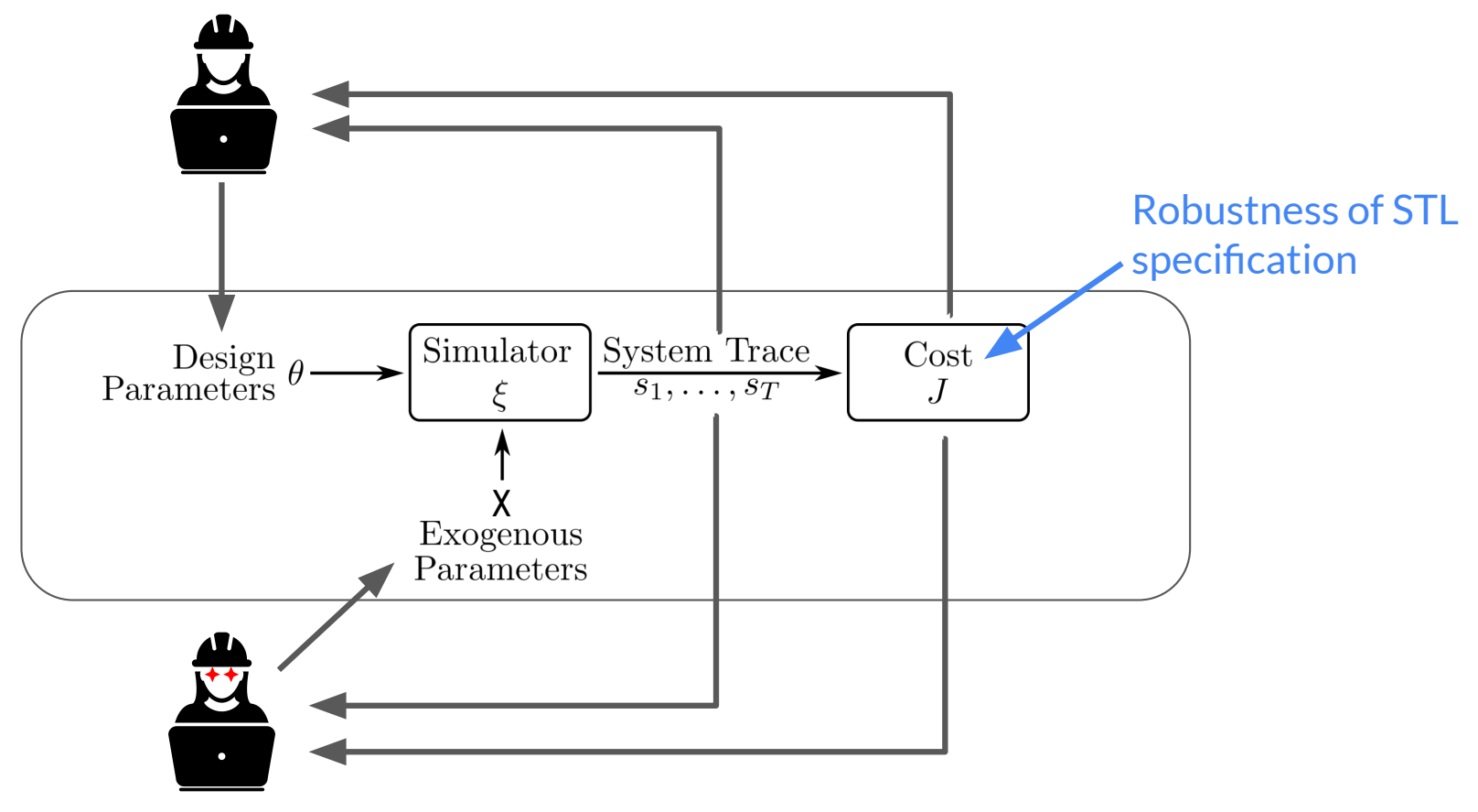

Ideal world
Spend at least 1 hour in the crater getting data, but return home before dark.
Sure! I will first do A, but if I observe B, then I will …
Sad reality
Spend at least 1 hour in the crater getting data, but return home before dark.
Um, how?
Well, first do A, then if B happens, …

Contributions:

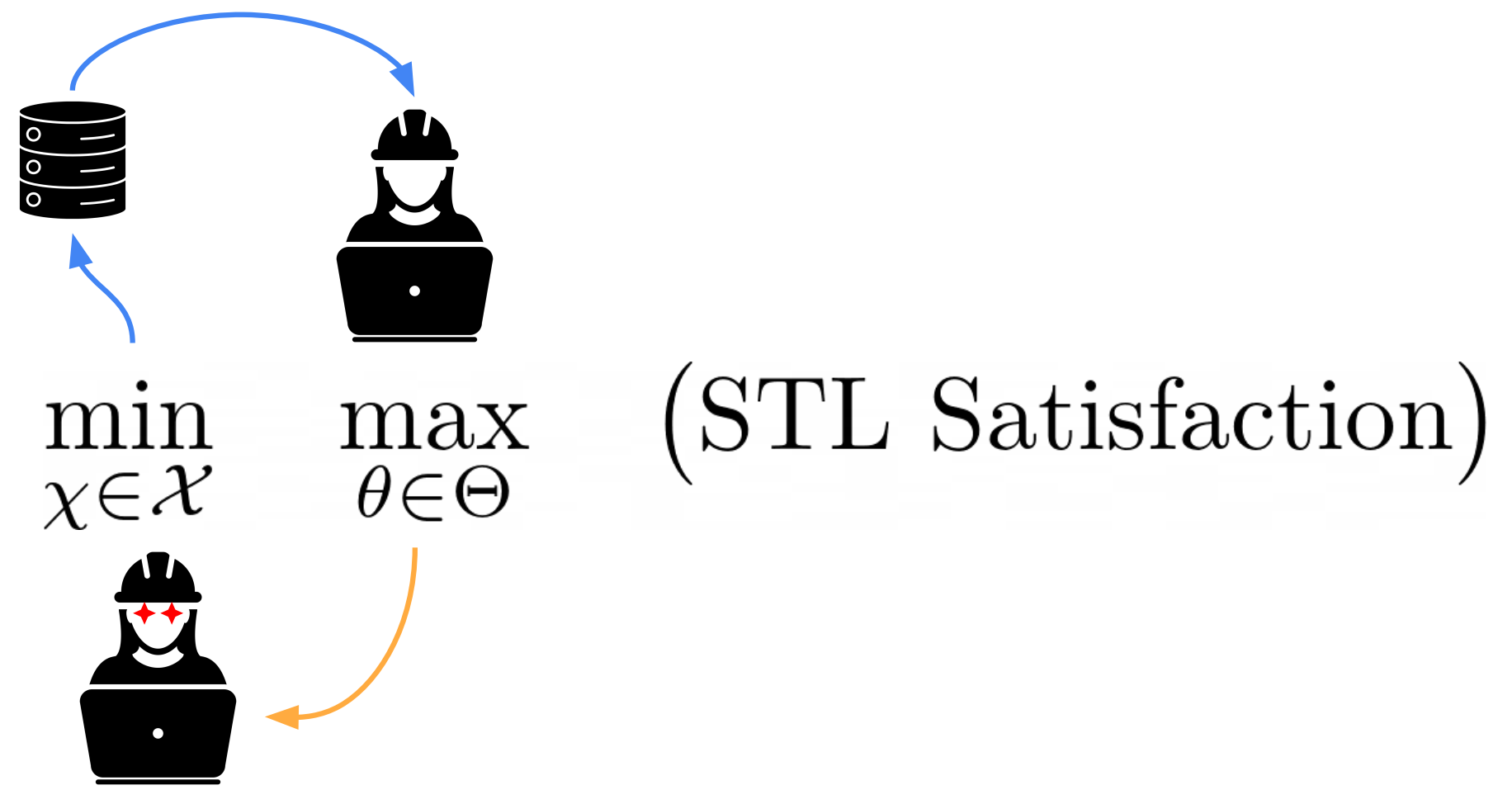
- Efficient, robust planner for STL tasks.
- Robustness from:
  - 2-player game formulation.
  - Counterexample-guided optimization.
- Efficiency from:
  - Differentiable simulation.
  - Differentiable temporal logic.
- Open-source implementation.

APPROACH:

The planner plays a zero-sum game with an adversary to maximize the robustness of its plan.


Robustness of STL specification
Design Parameters θ
Simulator ξ
System Trace $s_1, \ldots, s_T$
Cost J
X
Exogenous Parameters

The planner caches counterexamples found by the adversary to guide optimization. Automatic differentiation of the simulator and the temporal logic specification enables efficient planning.



$$\min_{\chi \in \mathcal{X}} \max_{\theta \in \Theta} \text{(STL Satisfaction)}$$

Architect

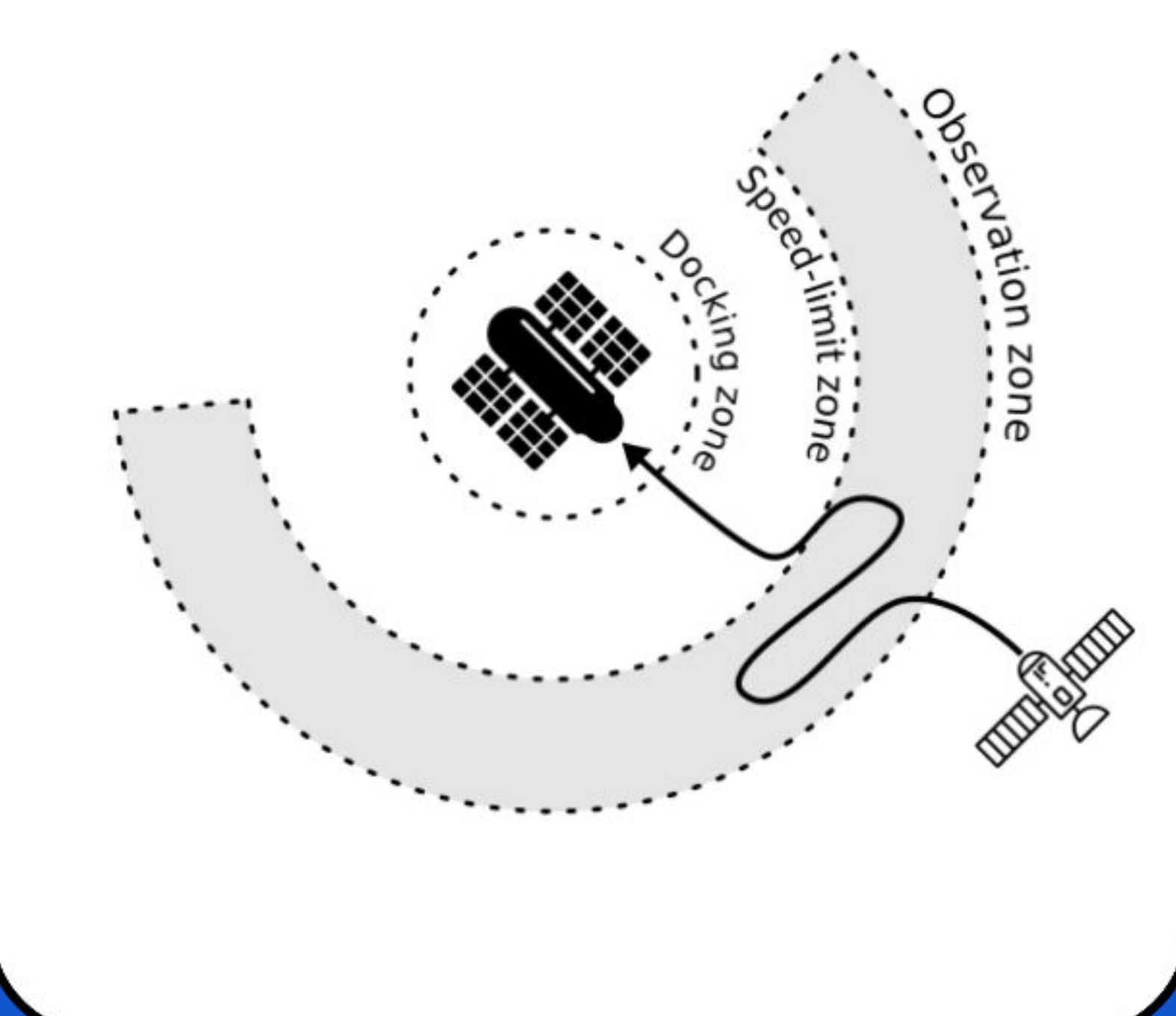
Automated, Robust Co-Design

Automatic differentiation

+ Counterexample-guided optimization

= Fast, safe planning for temporal logic.

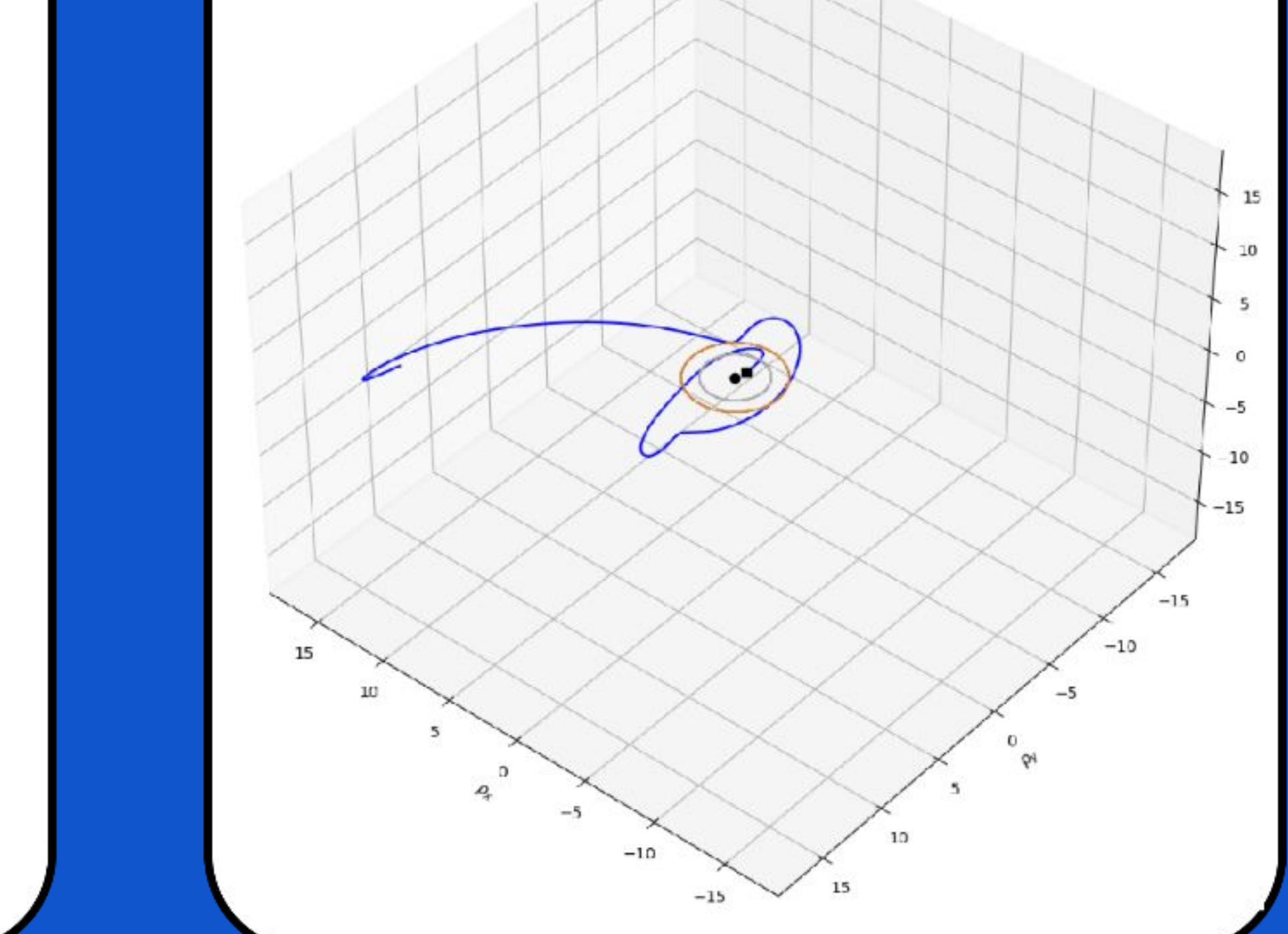
Robot task specification



STL formula

$$\psi_{\text{speed limit}} = (r \geq 2.0)\, \mathcal{U}\, \square (v \leq 0.1)$$
$$\psi_{\text{loiter}} = \lozenge \square_{[0, T_{obs}]} (2.0 \leq r \wedge r \leq 3.0)$$
$$\psi_{\text{reach target}} = \lozenge (r \leq 0.1)$$

Robust plan



Take a picture to **download** the **full paper and code**

IROS KYOTO 2022

Signal Temporal Logic (STL)

Formal task specification:

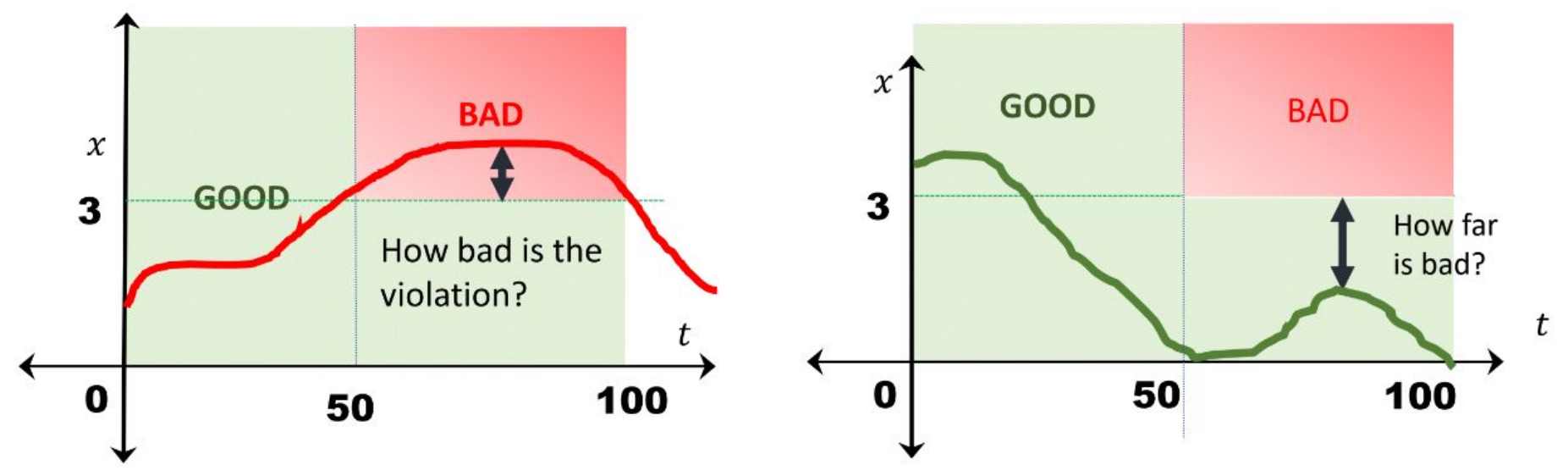
When close to target, **always** limit speed

Eventually be in grey region **for 2 minutes**

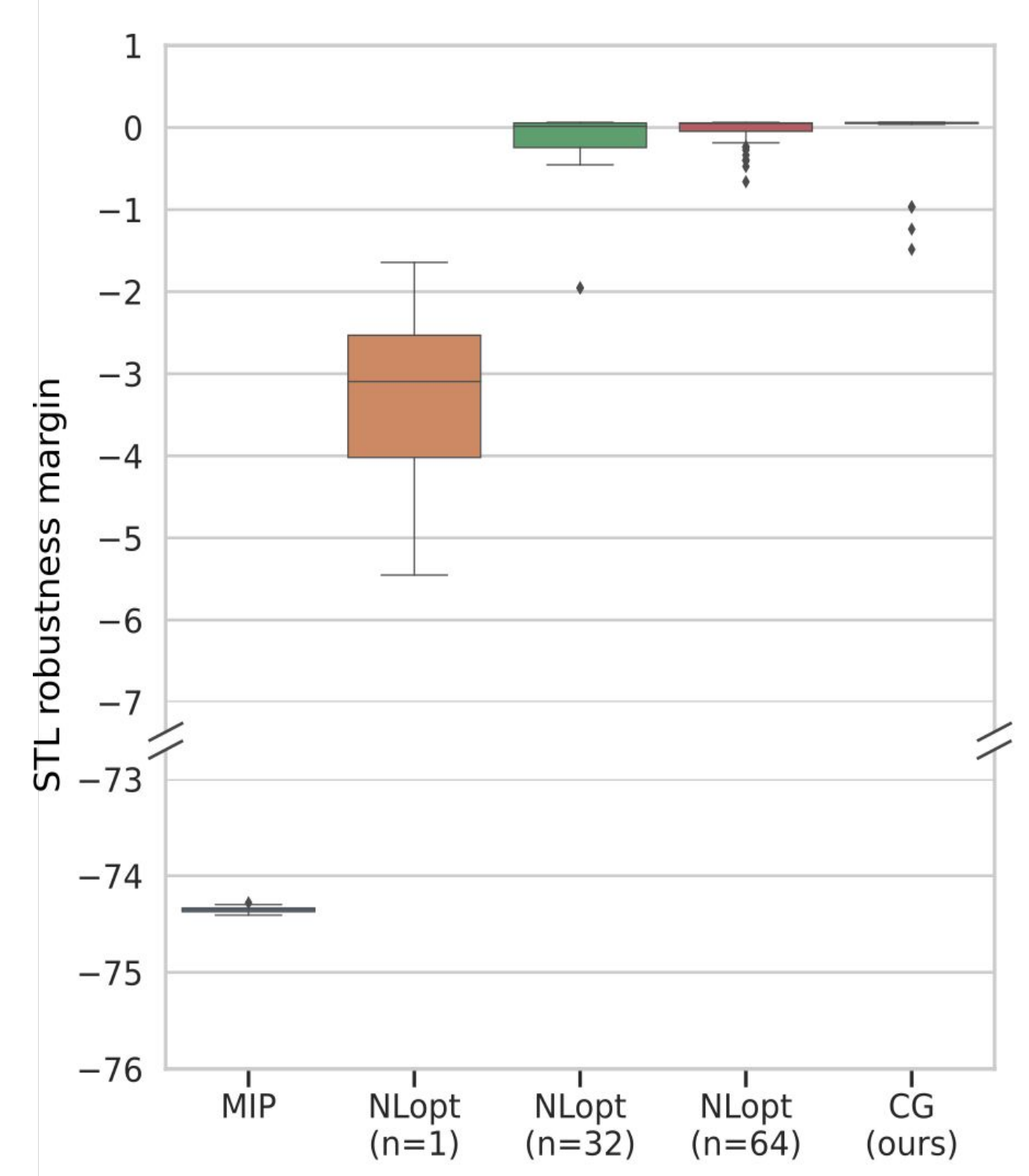
Eventually reach the target

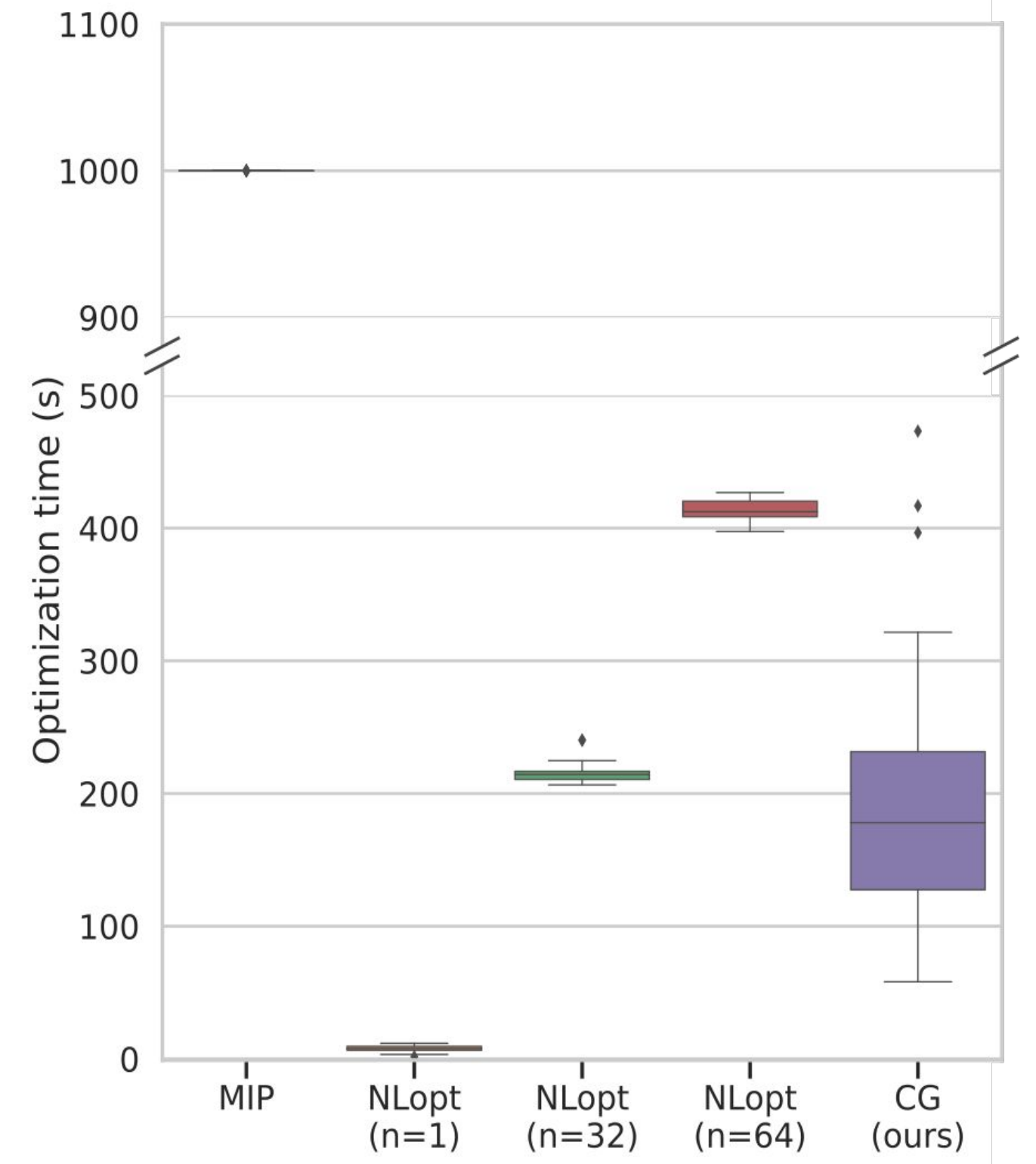
$$\psi_{\text{speed limit}} = (r \geq 2.0)\, \mathcal{U}\, \square (v \leq 0.1)$$
$$\psi_{\text{loiter}} = \lozenge \square_{[0, T_{obs}]} (2.0 \leq r \wedge r \leq 3.0)$$
$$\psi_{\text{reach target}} = \lozenge (r \leq 0.1)$$

(Differentiable) STL Robustness Metric



Experimental Results





Authors
Charles Dawson, Chuchu Fan

MIT AEROASTRO
REALM
NSF
MIT-IBM Watson AI Lab
DSTA Defence Science & Technology Agency