TEXAS STATE UNIVERSITY

BLOC

Mina Guirguis

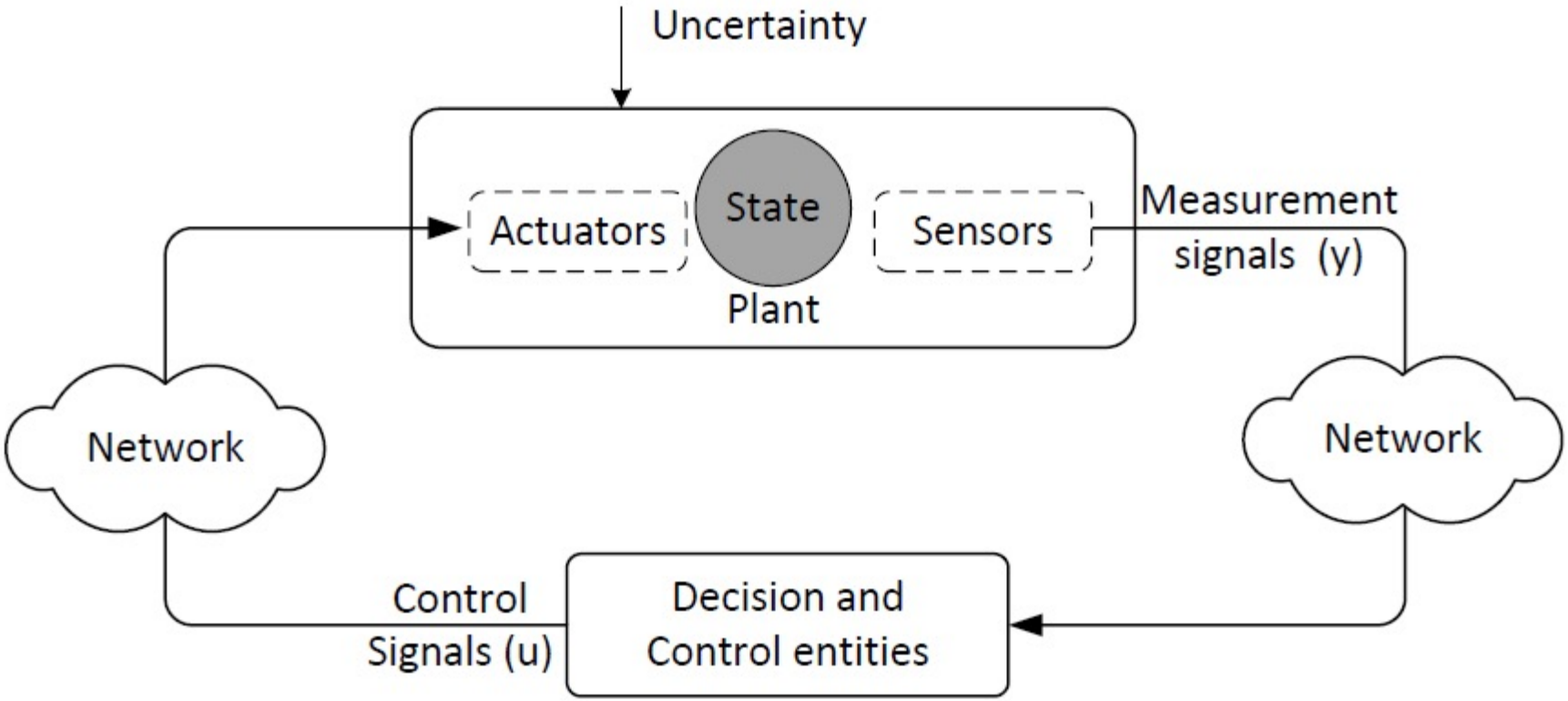
Joint work with: Noah Dunstatter, Alireza Tahsini, Clara Novoa, Khan Siddique, Justin Moore, Chuadhry Ahmed (Singapore University), and Christine Julien (UT)

Industrial Control System Security Workshop
December 6, 2022

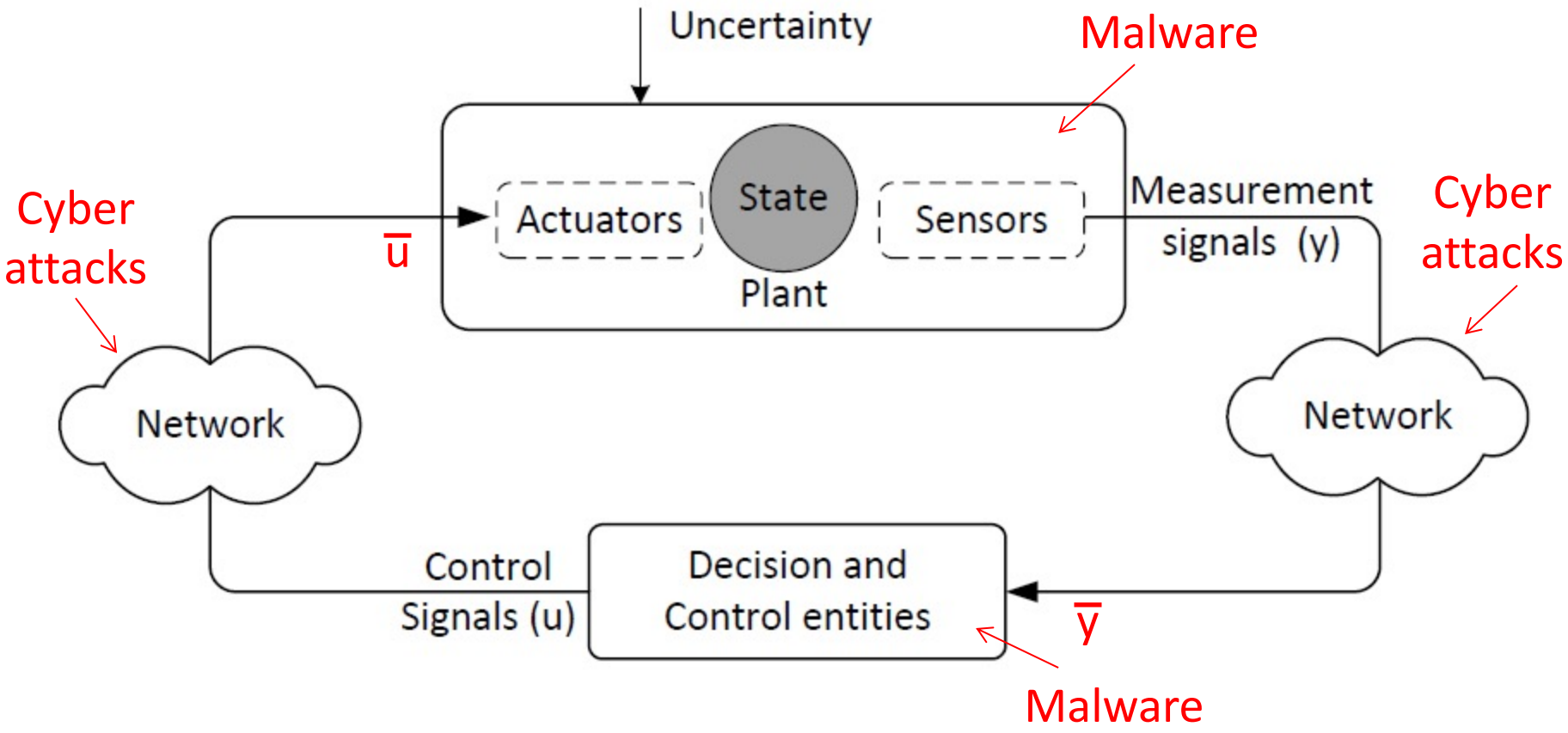
Cyber-Physical Systems

[utilities] [defense] [manufacturing] [transportation] [healthcare]

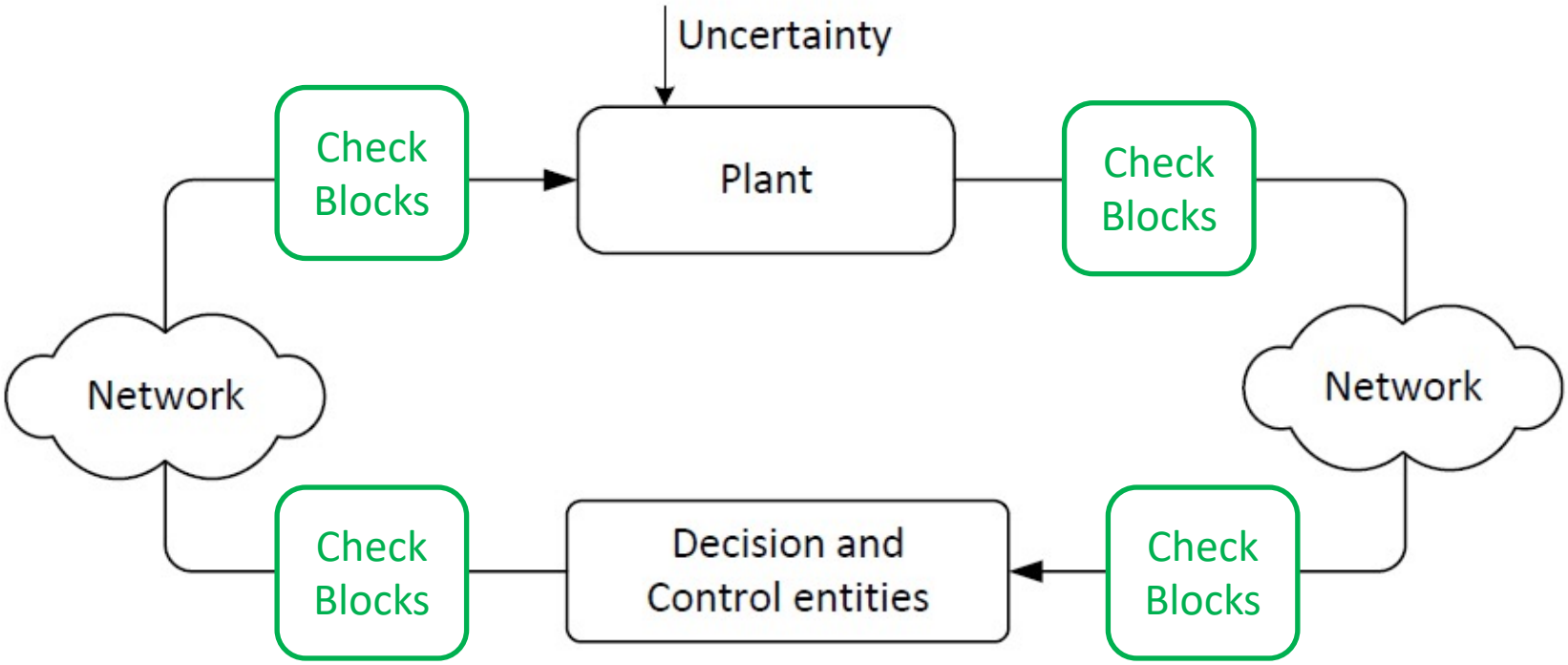
Feedback control loop



Many attack vectors

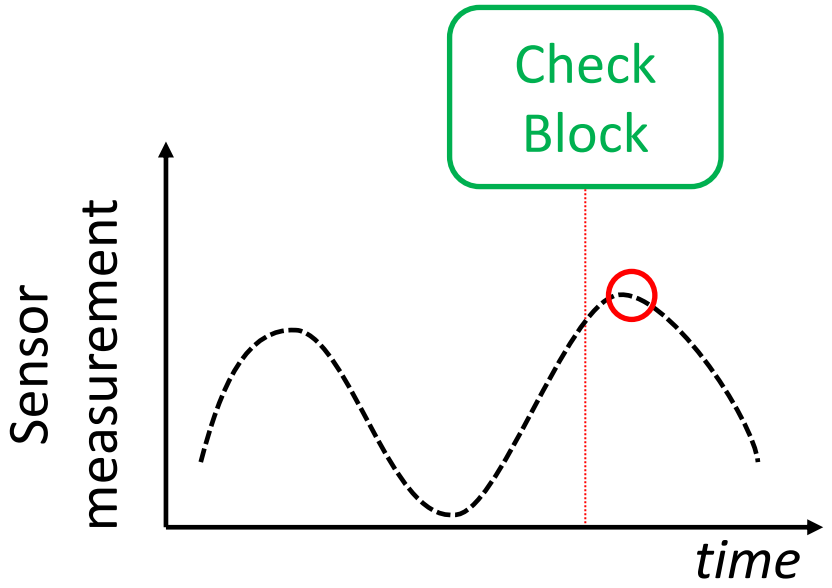


Many defense algorithms



A check block

- Threat/anomaly detection algorithm
  - Inspects a signal and flags suspicious behavior
  - Stateless/stateful
  - Incurs a cost to inspect



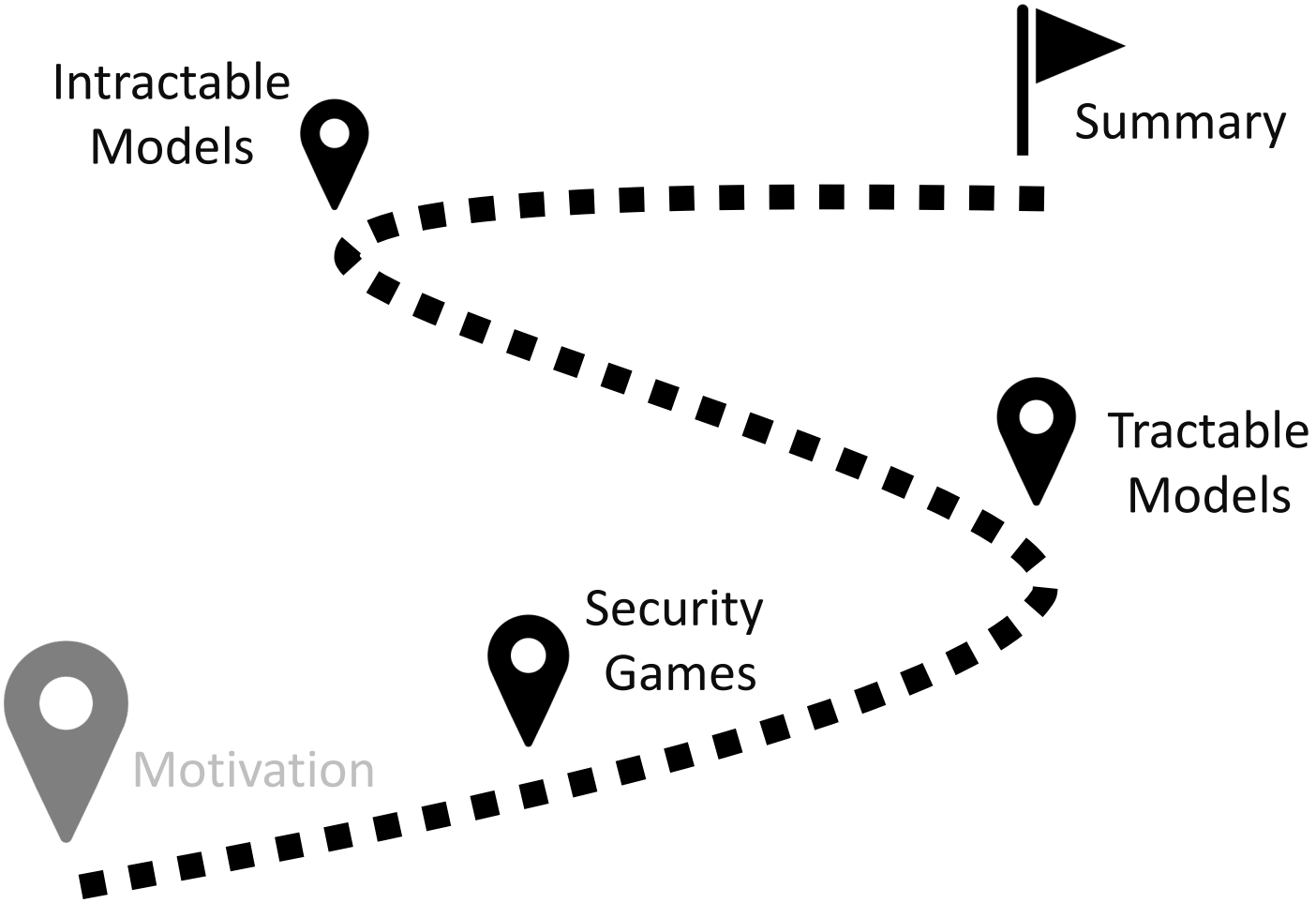
What is missing?

- A static assignment of check blocks can be exploited
  - Attackers can probe to learn and adjust their attack methods
- Cannot assign all check blocks
  - May violate the control loop delay constraints
- Check blocks are introduced in isolation of each other
  - Each block checks for a particular attack

Contributions

- BLOC: A Game-Theoretic framework to orchestrate the control loop with check blocks
  - Randomizes the assignment of check blocks
  - Abides to the delay constraints at run-time
  - Captures the effectiveness of various check blocks
  - Scales defense resources in tandem with importance

Talk roadmap



Security games

- Two players
  - A defender (D) has two targets to protect
  - An adversary (A) can choose which target to attack
- Rewards for attacking/defending
  - Target 1 is more valuable

	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

Security games

- Two players
  - A defender (D) has two targets to protect
  - An adversary (A) can choose which target to attack

- Rewards for attacking/defending
  - Target 1 is more valuable

	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

Security games

- Two players
  - A defender (D) has two targets to protect
  - An adversary (A) can choose which target to attack

- Rewards for attacking/defending
  - Target 1 is more valuable

	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

Security games

- Two players
  - A defender (D) has two targets to protect
  - An adversary (A) can choose which target to attack

- Rewards for attacking/defending
  - Target 1 is more valuable

	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

Security games

- Pure strategies can be defeated
- Players play mixed strategies
  - A probability distribution assigned to the actions
- Defender defends T1 with π_1 and T2 with π_2

	Attack Target 1	Attack Target 2
π_1 Defend Target 1	(4,-4)	(-1,1)
π_2 Defend Target 2	(-5,5)	(2,-2)

Solving security games

- How to pick the probabilities?

π_1

π_2

	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

  - Make your opponent indifferent

Attack T1 Attack T2

$$4*\pi_1 - 5*\pi_2 = -1*\pi_1 + 2*\pi_2$$

$$\pi_1 + \pi_2 = 1$$

$$\pi_1 , \pi_2 >= 0$$

$\pi_1 = 7/12$

$\pi_2 = 5/12$

  - Maximin policy -- obtained by solving a linear program

Value of the game

- The mixed strategy for the Attacker

1/4 3/4

7/12

5/12

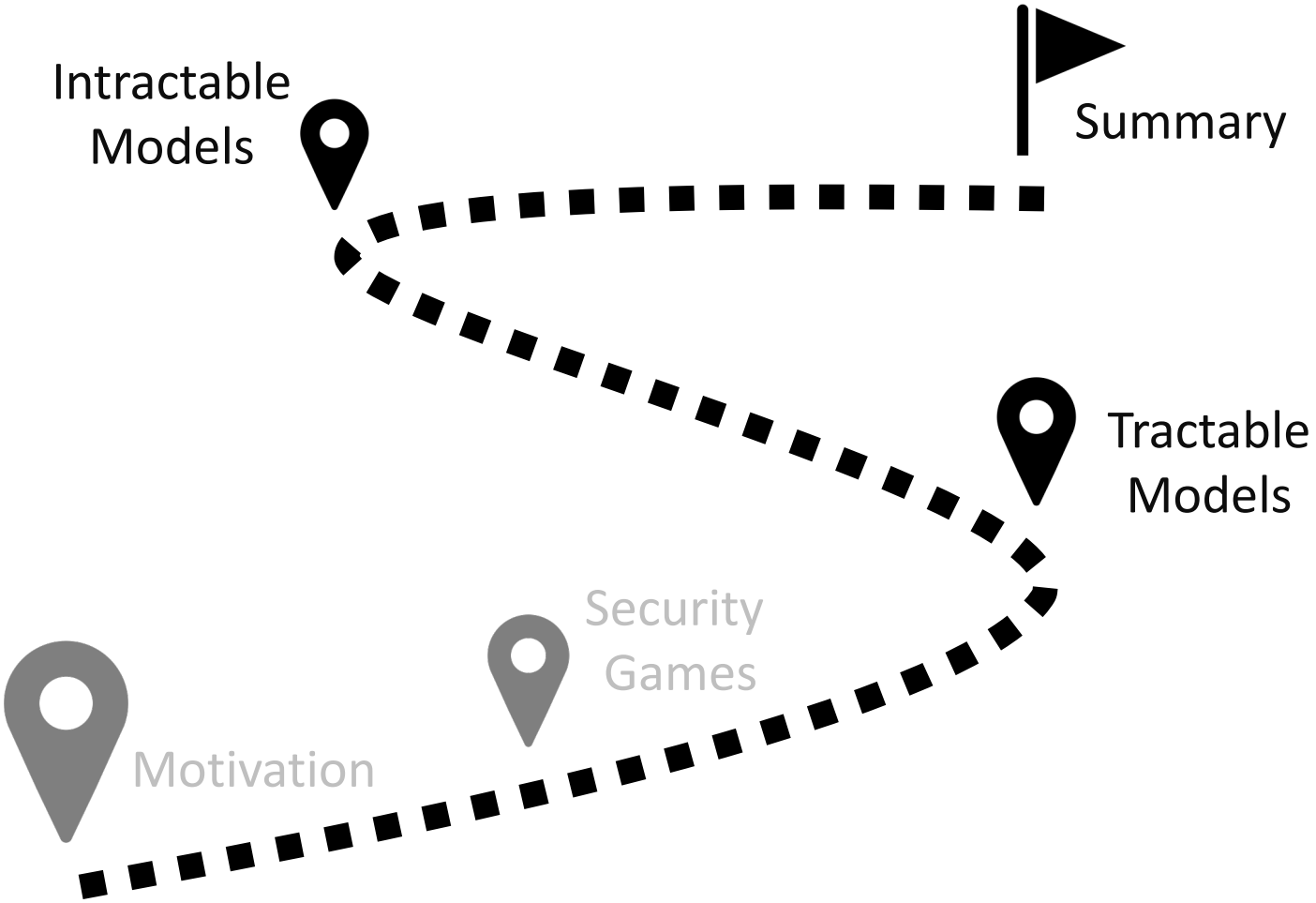
	Attack Target 1	Attack Target 2
Defend Target 1	(4,-4)	(-1,1)
Defend Target 2	(-5,5)	(2,-2)

- This becomes Nash Equilibrium (NE)

- Value of the game is the expected reward
  - For the Defender:

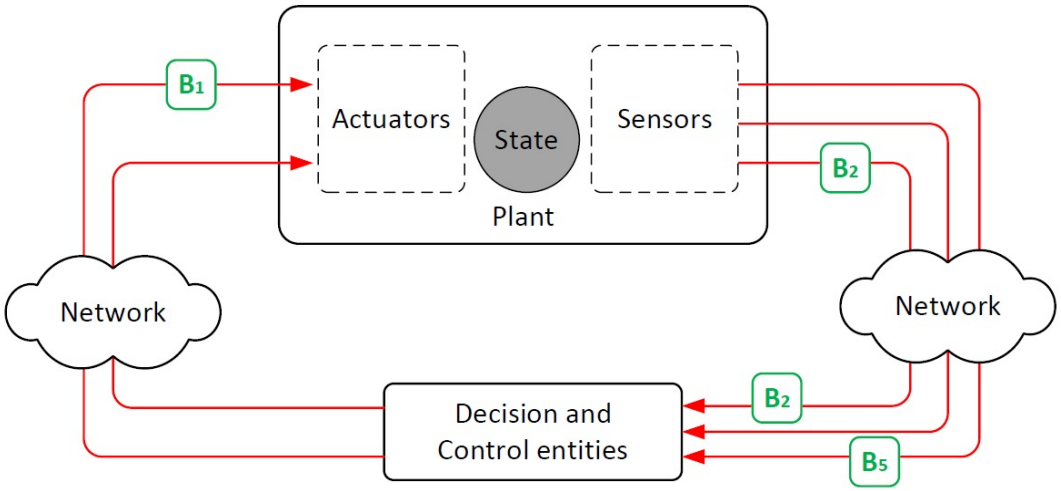
$$\begin{bmatrix} 7/12 & 5/12 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ -5 & 2 \end{bmatrix} \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}$$

Talk roadmap



Stateless model

- Two-player zero-sum game
  - Adversary: chooses a target signal and attack method
  - Defender: chooses check blocks to assign to targets



Stateless model

- A set of Target signals (T)
  - A target refers to an asset -- a measurement or control signal
  - Defender gains utility U_t^p when target *t* is protected and U_t^u when it is unprotected

- A set of Attack methods (A)
  - Adversary chooses an attack method *a* to attack target *t*
  - Category *c: <a, t>*

Stateless model

- A set of check Blocks (B)
  - E_b^a is the effectiveness of block *b* against attack *a*
  - A block *b* takes T_b to check an incoming signal

$$\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0.5 & 0 & 0.1 \\ 0.2 & 0.3 & 0.4 \\ 0 & 0.9 & 0 \end{bmatrix} \end{array}$$

- A set of adversaries (Θ)
  - Value targets differently
  - Encountered with different probabilities z_θ

- Assumptions:
  - N_t is the number of signals that arrive under each target t
  - C_t is the maximum delay tolerated by target *t*

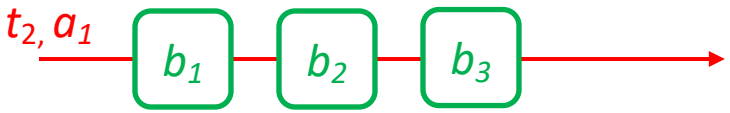
Stateless model

- An assignment $n_{b,t}$ of check blocks to targets
  - Consider the following assignment

$$\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0.5 & 0 & 0.1 \\ 0.2 & 0.3 & 0.4 \\ 0 & 0.9 & 0 \end{bmatrix} \end{array}$$

Effectiveness



$$(1 - 0.5) \times (1 - 0.2) \times (1 - 0)$$

  - An attack would go undetected, if all the blocks fail
  - To detect an attack = 1 – [probability that they all fail]

Stateless model

- Seek an assignment $n_{b,t}$ of check blocks to targets
  - Maximize the defender's worst-case utility, s_θ , given the probability z_θ of encountering adversary θ

$$\max_{\mathbf{n}} \quad \sum_{\theta \in \Theta} z_\theta s_\theta \tag{1}$$

$$s_\theta \leq x_c U_t^p + (1 - x_c) U_t^u \qquad \forall \theta, c \tag{2}$$

$$x_c = 1 - \Pi_{b \in B} (1 - E_b^a \times n_{b,t}) \qquad \forall c \tag{3}$$

$$\sum_{b \in B} n_{b,t} \times N_t \times T_b \leq C_t \qquad \forall t \tag{4}$$

$$n_{b,t} \in \{0, 1\} \qquad \forall b, t \tag{5}$$

Marginal assignment

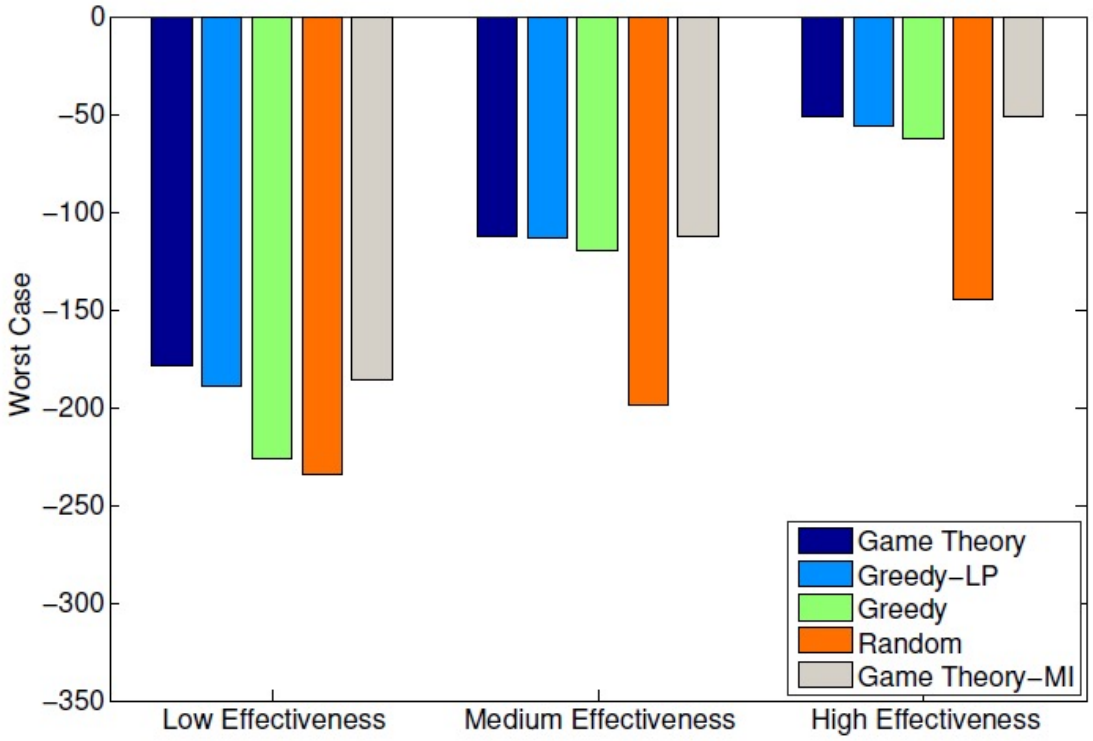
- Realizing a marginal assignment:
  - Sampling signals
  - Adjusting the resolution of check blocks
  - A convex combination of pure strategies [Budish'13]

$$X = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.5 & 0.5 & 0 \\ 0.8 & 0 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}$$

$$X = 0.5 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + 0.3 \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + 0.2 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

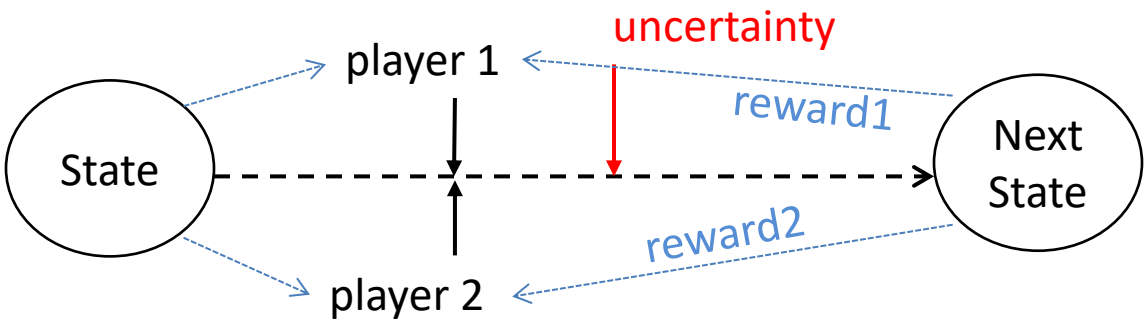
Results: Stateless model

- Average worst-case utility



Stochastic games

- Players play a series of games with a state maintained in between



- A state captures:
  - Current allocation of resources to the players
  - Actions available to the players

Stochastic games

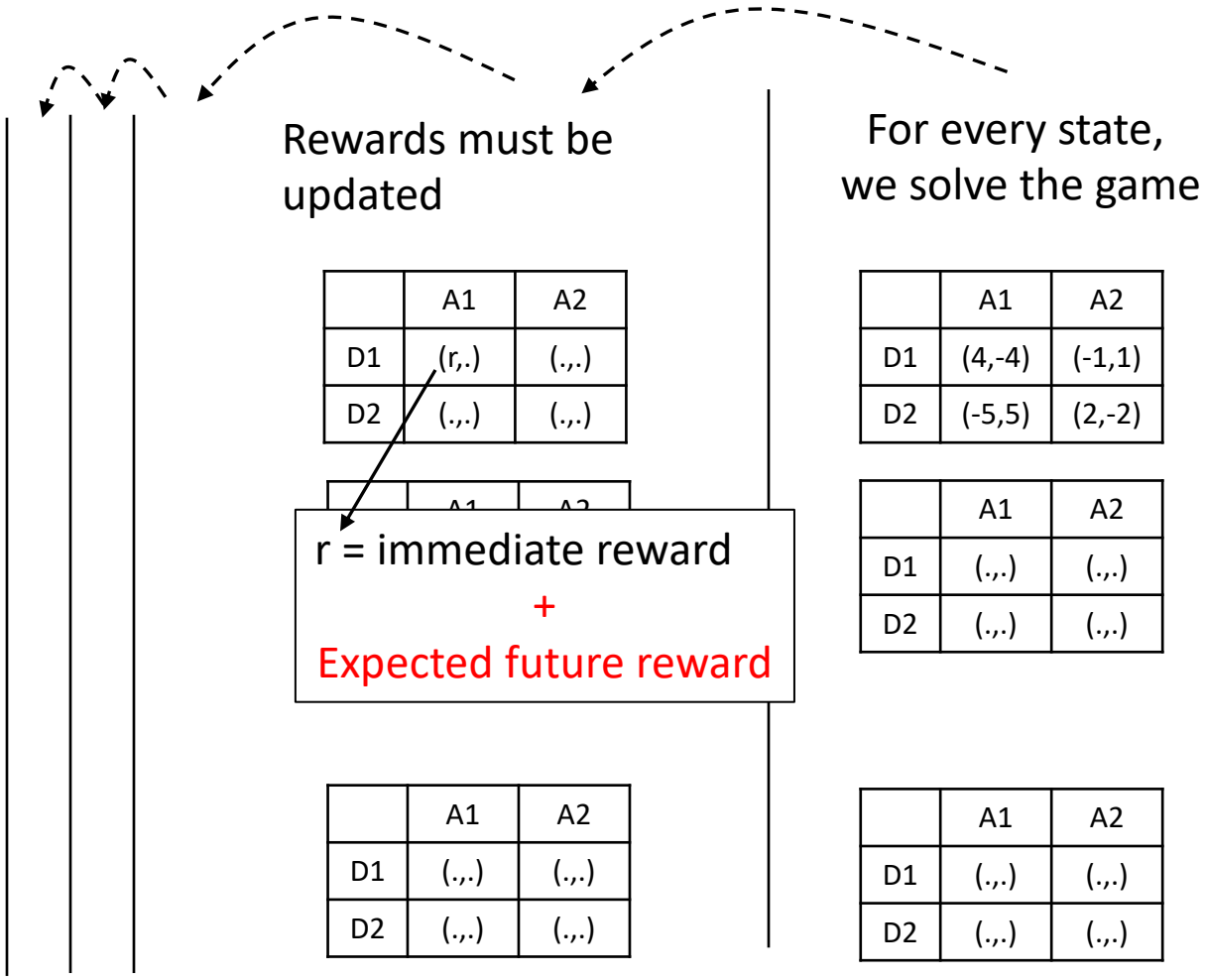
- A stochastic game:
  - A set of states S
  - A set of players M
  - A set of actions U_m available to player m
  - A transition probability T(i,u,j)
  - An immediate reward vector R(i,u,j)
  - Starting state and a horizon (discount factor if infinite)

- Each player observes the current state and picks u to maximize their expected reward in the worst case
  - Rewards depends on the actions taken by other players

Solving stochastic games

- Illustration

Convergence
[Littman'94]



Stateful model

- A set of states (S)
  - A state *s* describes the current assignment of blocks to targets

$$\begin{array}{c} \\ b_1 \\ b_2 \\ b_3 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{bmatrix} 0 & -1 & 2 \\ -1 & 3 & 0 \\ -1 & -1 & 0 \end{bmatrix} \end{array}$$

-1: no assigned
0: assigned and effective
i: steps to be effective

  - W_b denotes the maximum warm-up period for block *b*

- Two players – an adversary (A) and a defender (D)
  - Adversary's actions ($\mathcal{A}_a$): pick an attack method and target
  - Defender's actions ($\mathcal{A}_d$): remove, add, no change to blocks

Stateful model

- A transition function
  - $\mathcal{T}(s, a, d, s')$ gives the probability of going into state s' from s under actions *a* and *d*

- A reward function
  - $\mathcal{R}(s, a, d, s')$ gives the reward obtained from going to state s' from s under actions *a* and *d*

$$\mathcal{R}(s, a, d, s') = \sum_{t \in T} x_{a,t,d} U_t^p + (1 - x_{a,t,d}) U_t^u$$

- Discount factor for an infinite-horizon setting
  - $\beta : \quad 0 < \beta < 1$

Stateful model

- We use value iteration to obtain optimal strategies

$$Q(s,a,d) = \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,d,s') R(s,a,d,s') + \beta \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,d,s') V(s')$$

Quality of action pair in State s

Expected immediate reward

Expected future reward (discounted)

$$V(s) = \max_{\pi \in \Pi(\mathcal{A}_d)} \min_{a \in \mathcal{A}_a} \sum_{d \in \mathcal{A}_d} \pi(s,d)\ Q(s,a,d)$$

Defender mixed strategy given the adversary would minimize

Stateful model

- In every sub-game, we solve the following game based on the actions available to the players to obtain the defender's policy $\pi(s, d)$

$$\max_{\pi(s,d)\in\Pi(\mathcal{A}_d)} z$$

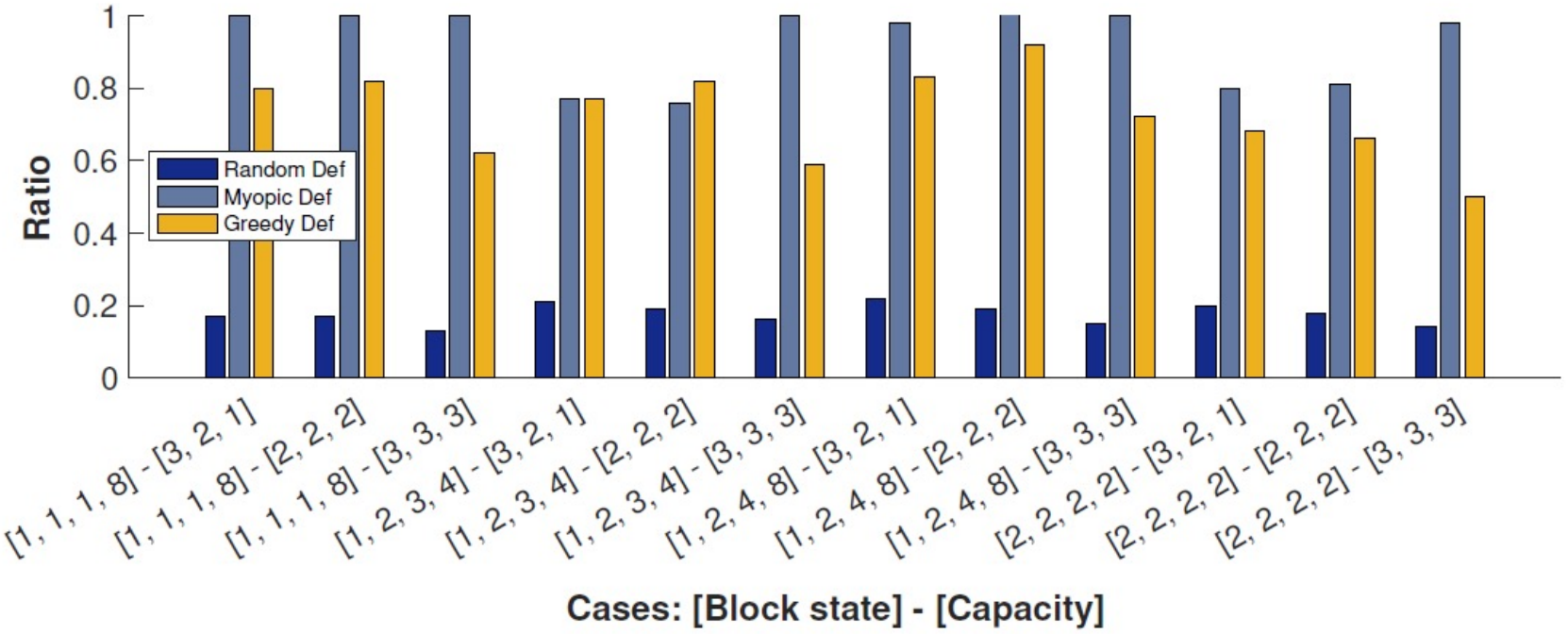
$$z \leq \sum_{d\in\mathcal{A}_d} \pi(s,d)Q(s,a,d), \quad \forall a \in \mathcal{A}_a$$

$$\sum_{d\in\mathcal{A}_d(s)} \pi(s,d) = 1$$

$$0 \leq \pi(s,d) \leq 1 \qquad \forall d \in \mathcal{A}_d(s)$$

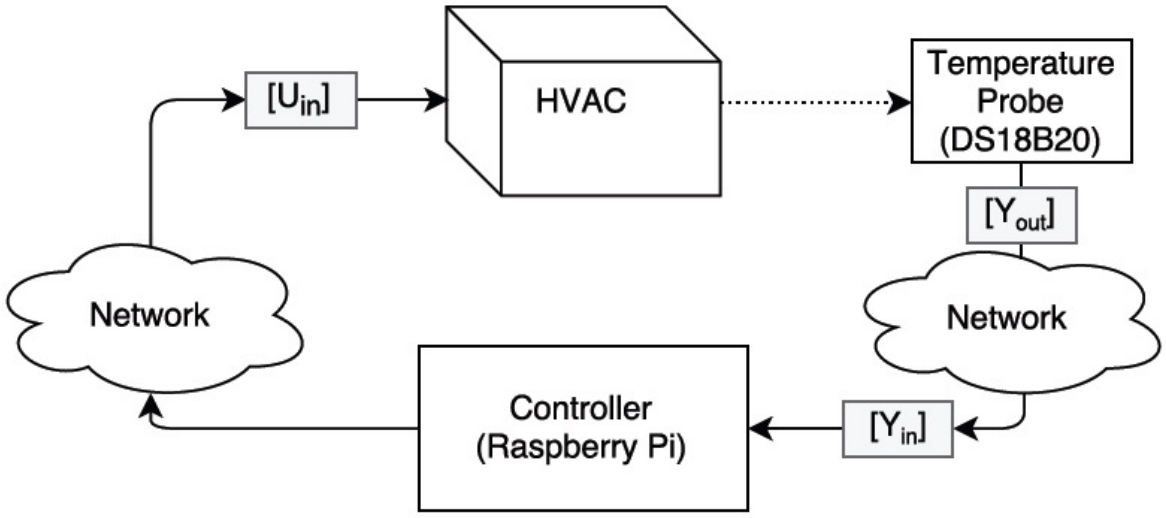
Results: Stateful model

- Relative performance of Random, Myopic and Greedy to Rational



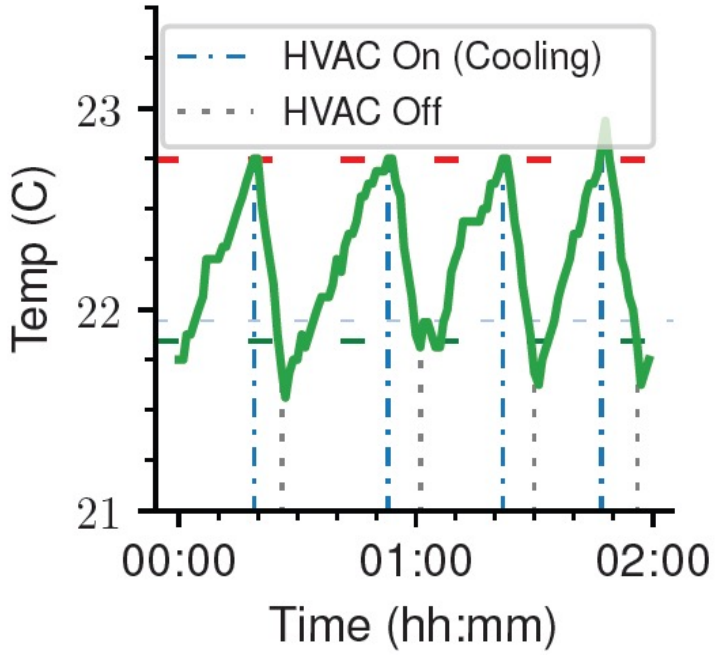
Implementation setup

- We orchestrated a Heating, Ventilation, and Air Conditioning (HVAC) system



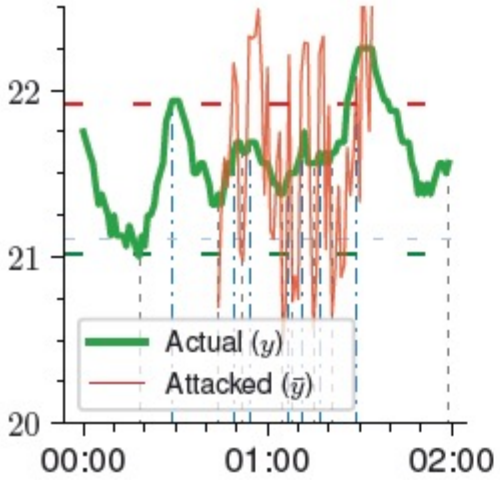
Implementation setup

- Normal operation
  - Target temperature set to ~22 C (71.5 F)

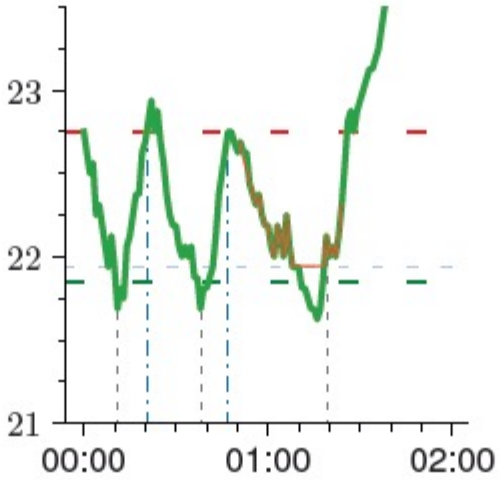


Implementation results: HVAC

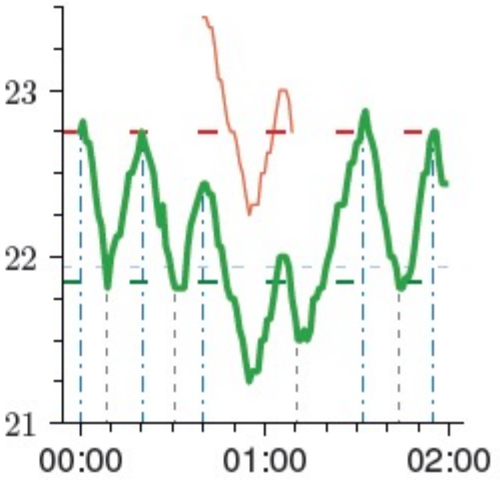
- Operation under attack



Fuzzy



Clamp



Offset

	Normal	Fuzzy	Clamp	Offset
Control Events	100%	375%	50%	0%
Time Running	26.9%	55%	100%	100%

Implementation results: HVAC

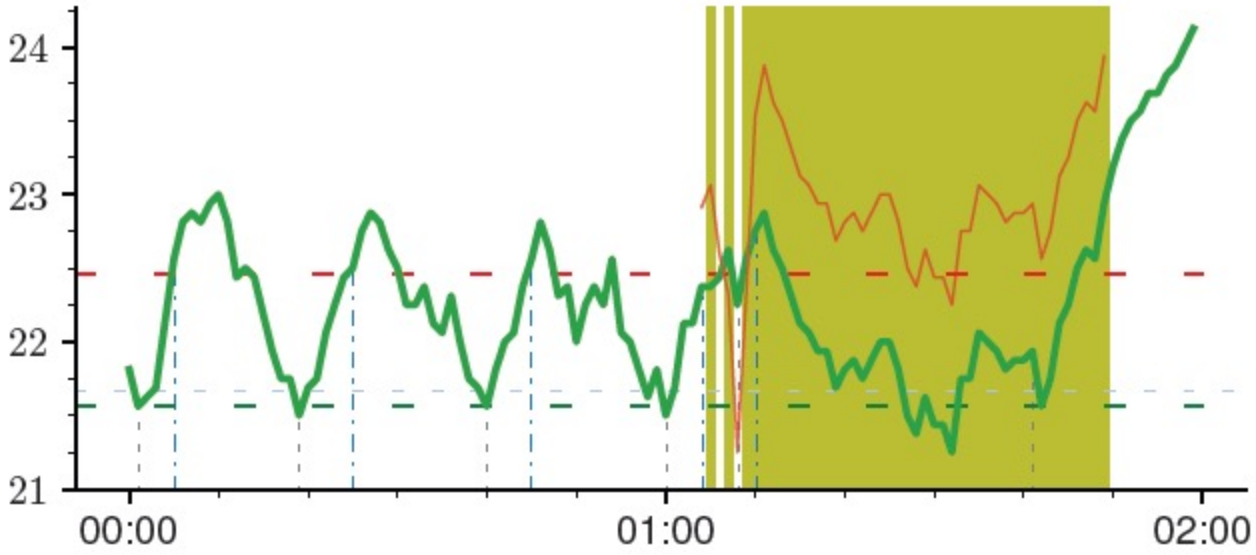
- Used 3 types of check blocks [1 stateless]
  - Control frequency block
  - Delta temperature block
  - Watermark block

- Effectiveness of blocks against attacks:

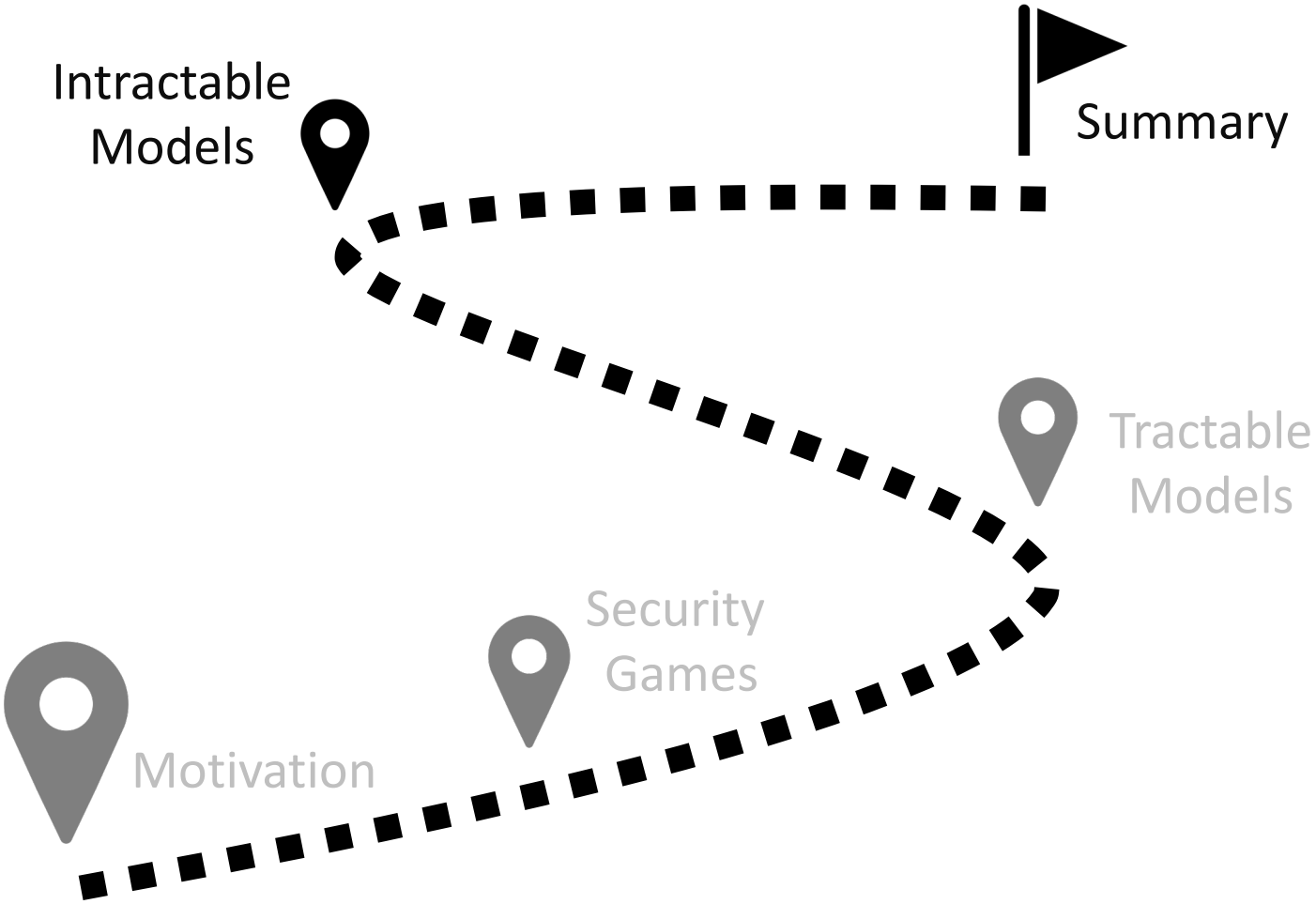
	Fuzzy	Clamp	Offset
Control Frequency (state=6)	0.99	0	0
Delta Temp. 1 (state=2)	0.7	0	0.8
Delta Temp. 2 (state=5)	0	1	0
Watermark (stateless)	0.9	0.9	0.9

Implementation results: HVAC

- Attack launched after ~1 hour and ran for 45 minutes
  - A sub-game is played every one minute
  - Worst-case was offset attack
  - Detectable 90% of the time



Talk roadmap



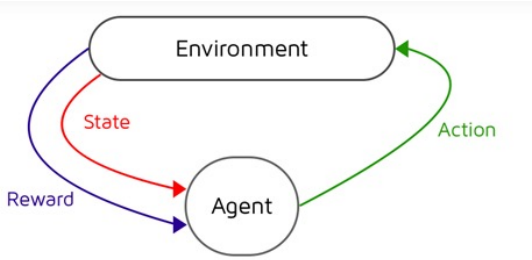
The curse of dimensionality

- An assignment of 5 check blocks on 3 target signals with each block's warm-up period of 4 steps
  - State space exceeds 1 billion states
  - Action spaces grow exponentially large

- Deep Reinforcement Learning

Reinforcement Learning (RL)

- An agent learns from interacting with the environment



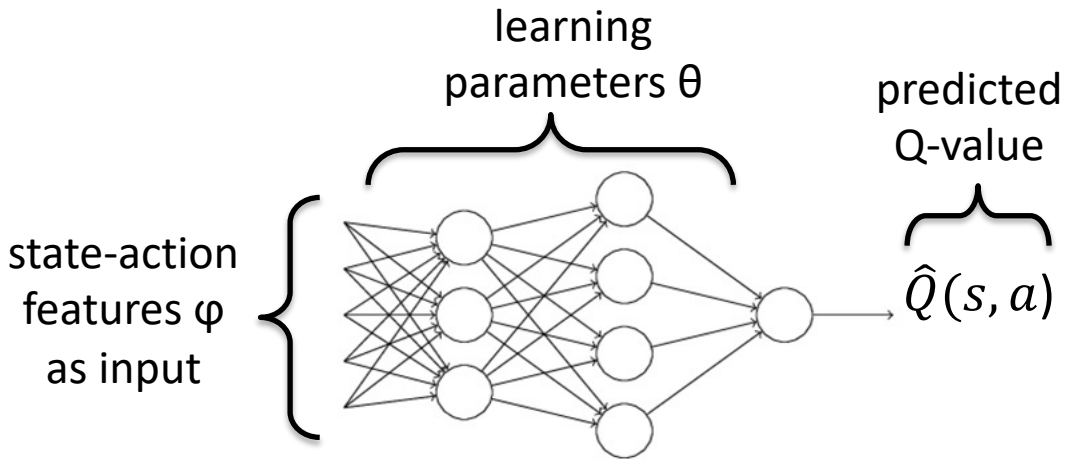
- Goal: find a policy that maximizes the long-term reward

- Q-learning

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha)\underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q(s_{t+1}, a_{t+1})}_{\text{estimate of optimal future value}}\right)}^{\text{learned value}}$$

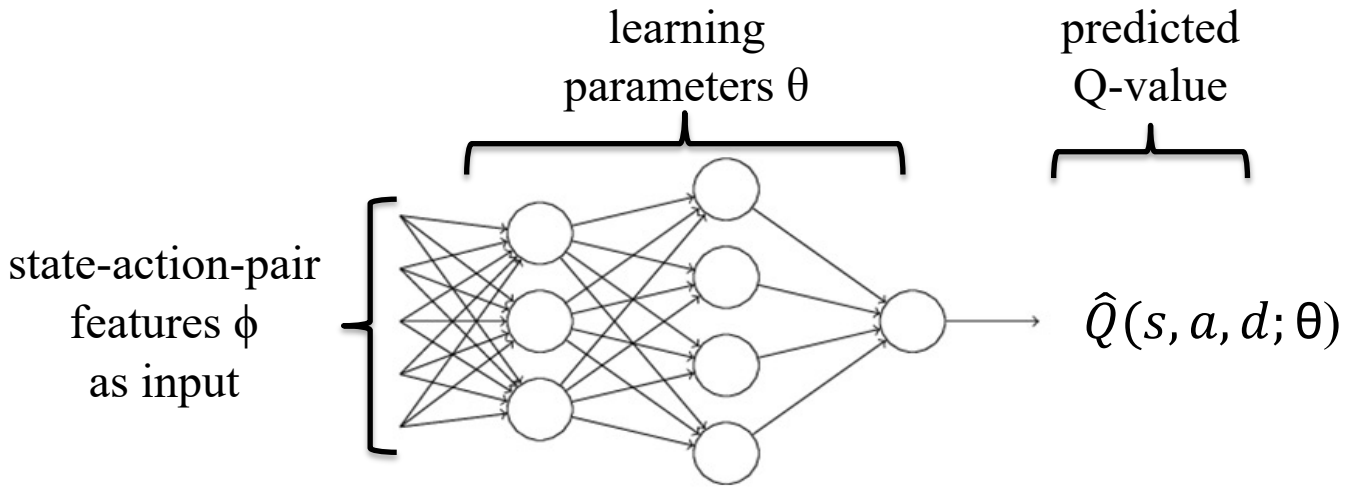
Deep Reinforcement Learning

- Infeasible to keep a table for each state-action pair
  - Rely on a set of features and weights to approximate the Q(s,a)



DeepBLOC

- Extend Deep Q-learning into stochastic games
  - Asymmetric players with different action spaces
  - Replaces the max operator with maximin
  - Model free
  - Off policy



More approximations

- Limit the action spaces to solve the sub-games
  - Introduced the attack budget

- Fictitious play
  - Iterative game solver where players track the empirical frequency of an opponent's actions and best respond
  - Enables us to tradeoff accuracy and time -- 7-20x speed up with 94% accuracy

Evaluation experiments

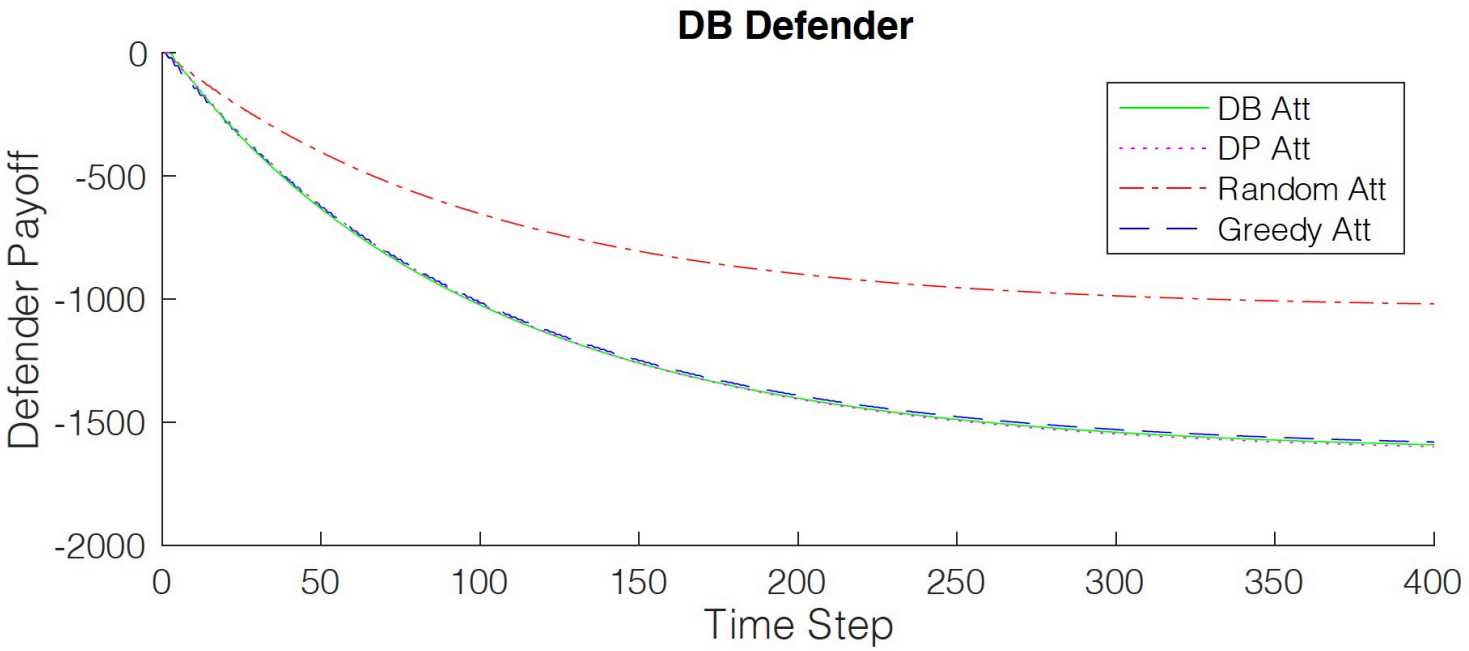
- Small environment
  - Targets: 3 Blocks: 3 Attack Methods: 3
  - State space $3x10^5$
- Medium environment
  - Targets: 5 Blocks: 6 Attack Methods: 6
  - State space $7x10^{18}$
- Large environment
  - Targets: 8 Blocks: 10 Attack Methods: 10
  - State space $3x10^{89}$

Evaluation experiments

- Defender's assignment policies:
  - Random
  - G1: assigns the block with the highest effectiveness relative to its warm-up period to the highest-utility least-protected target
  - G2: as G1, but adds a limit and moves to the next highest utility target
  - DB: action based on DeepBLOC
- Attacker's policies:
  - Random
  - Greedy: action with highest immediate reward
  - DB: action based on DeepBLOC

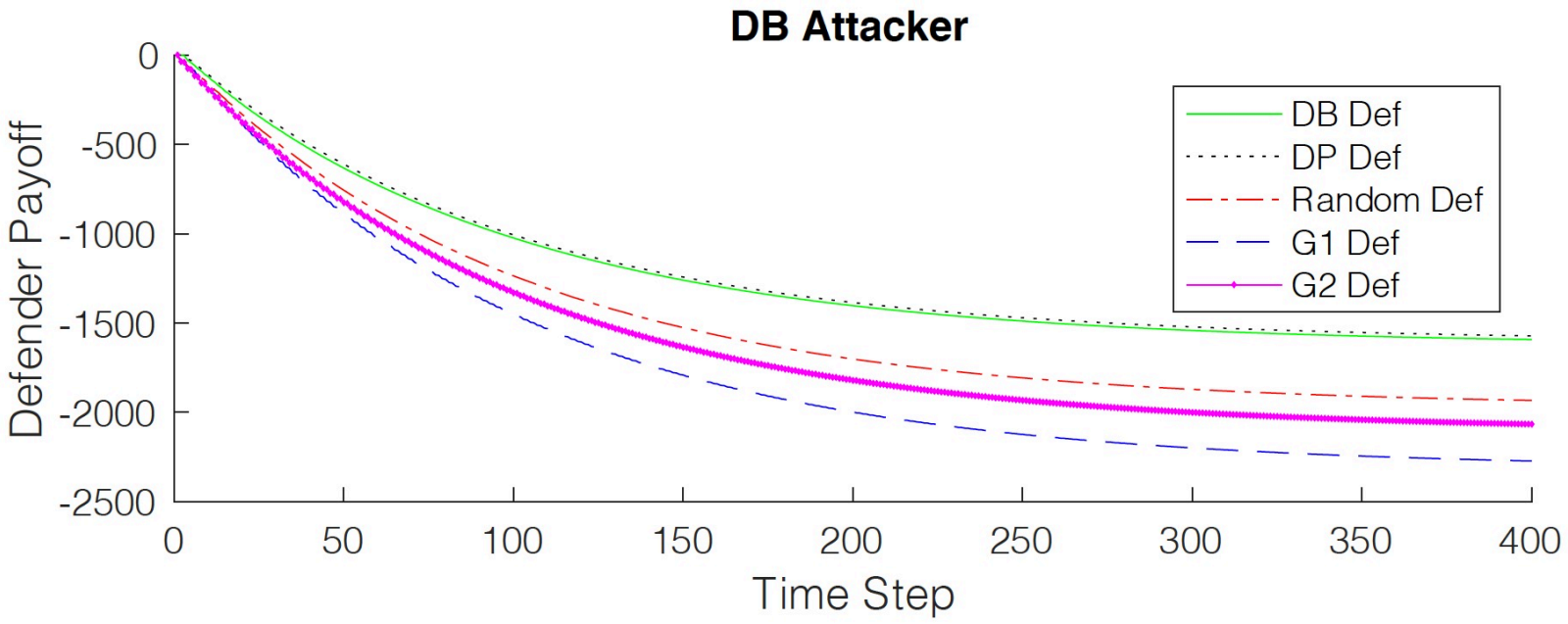
Results: DeepBLOC

- Small environment
  - DB defender payoff against various attacks



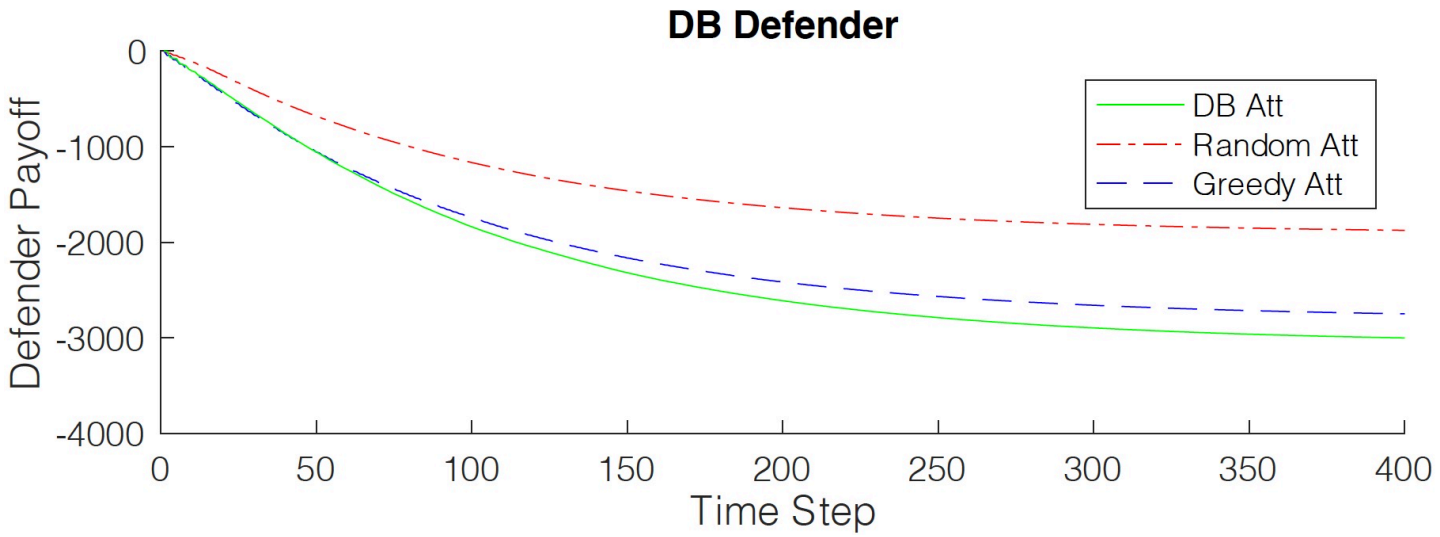
Results: DeepBLOC

- Small environment
  - Defender payoff against DB attacker



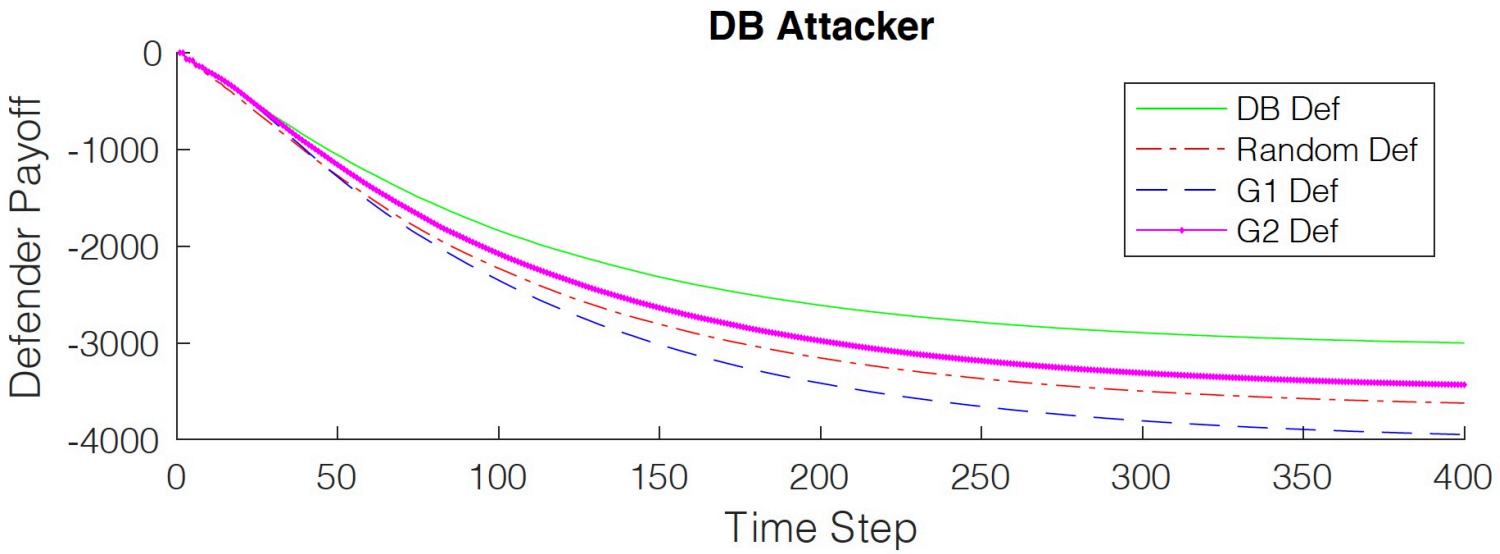
Results: DeepBLOC

- Large environment
  - DB defender payoff against various attacks



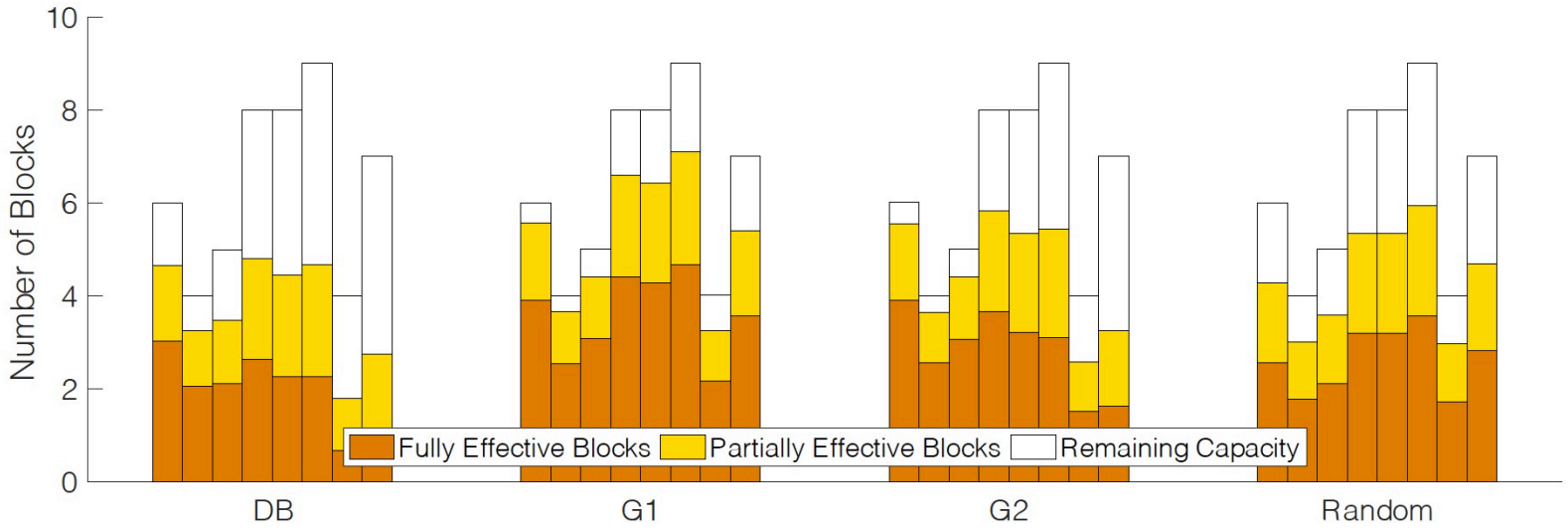
Results: DeepBLOC

- Large environment
  - Defender payoff against DB attacker



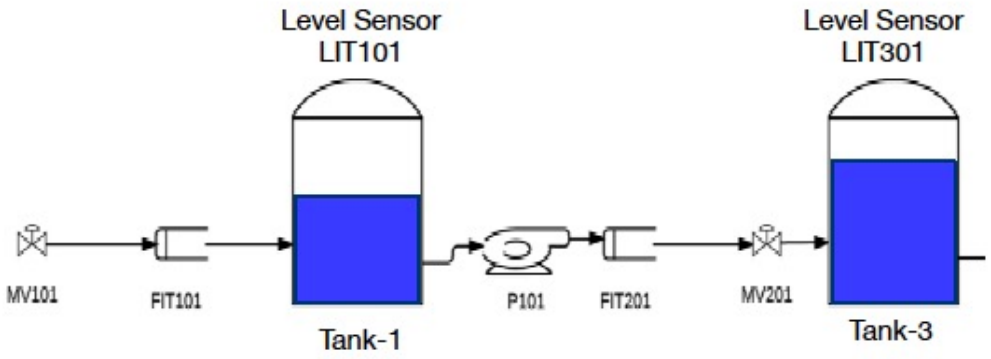
Results: DeepBLOC

- Large environment
  - Number of blocks assigned to targets



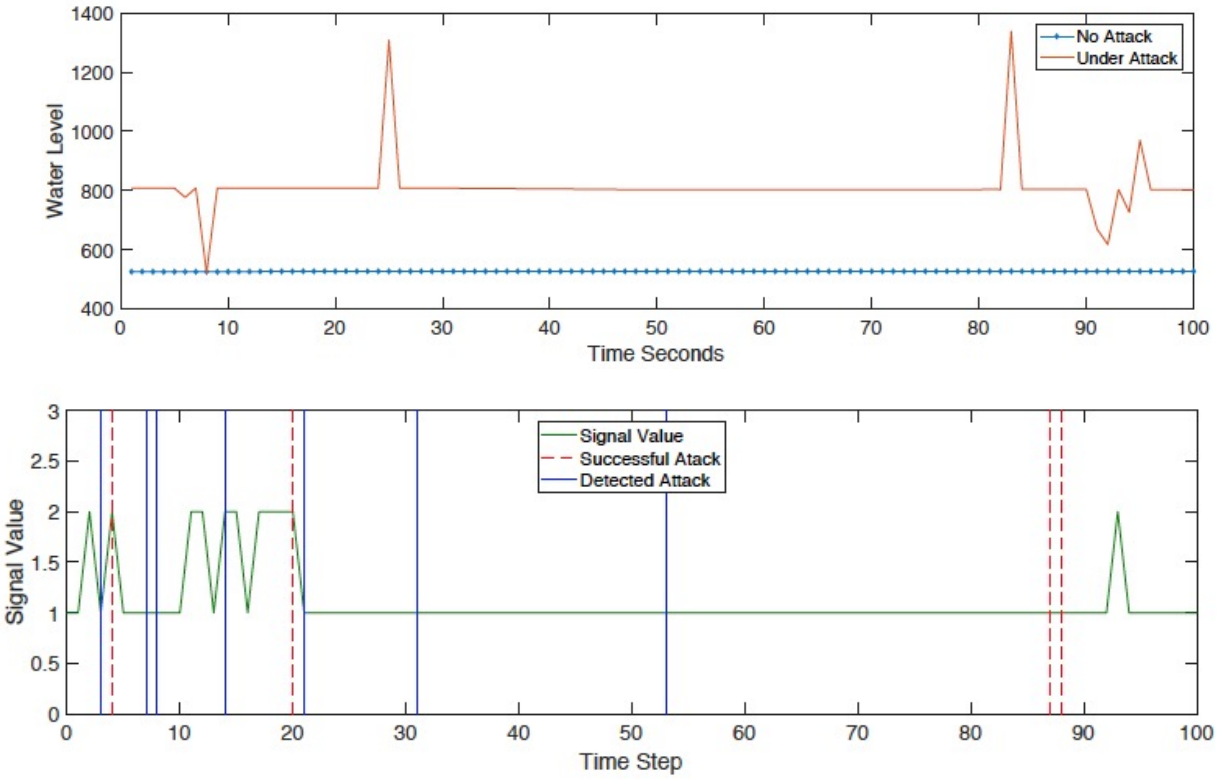
Implementation setup

- Six-stage water purification system



- Consider 6 attack methods on the Level Sensor 101
- Consider 6 check blocks (5 stateful)

Implementation results



- DB detected 55% of the attacks in comparison to greedy that detected 39%

Summary of contributions

- Game theory allows us to operate under the worst-case scenarios
- BLOC strategically assigns check blocks to the CPS
  - Capitalize on randomization through mixed strategies
  - Capture the relative importance of various targets
  - Adhere to the control loop constraints
- Identify policies that exhibit intelligence and outperform other strategies

BLOC

Mina Guirguis