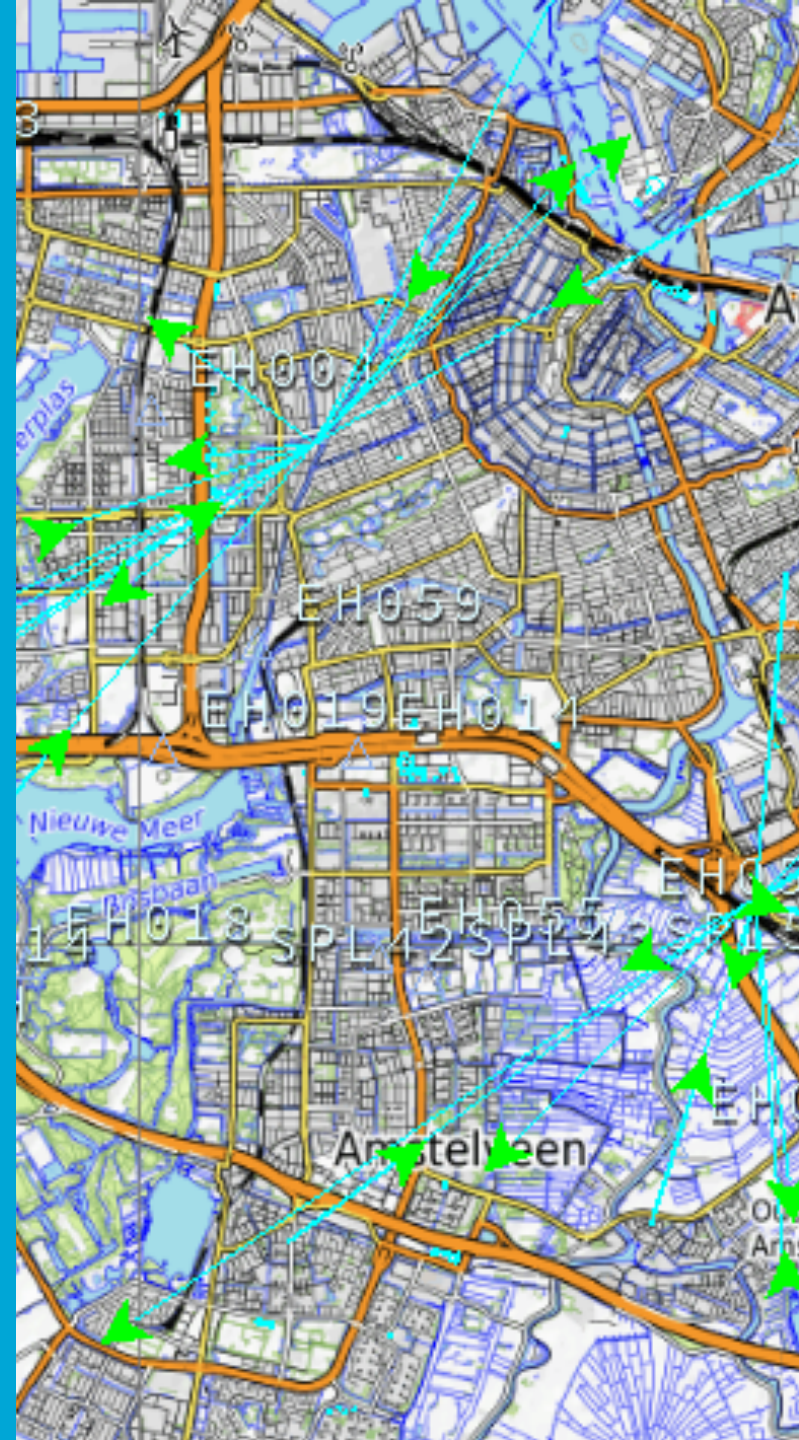


Deep Reinforcement Learning for Drone Conflict Resolution Under Uncertainty

SESAR Innovation Days 2025, Bled

Sasha Vlaskin, Muhammad Fazlur Rahman, Emmanuel Sunil,
Joost Ellerbroek, Jacco Hoekstra, Dennis Nieuwenhuisen



Who am I?



Aleksandr 'Sasha' Vlaskin

PhD Candidate

NLR *and* TU Delft



Dr. Emmanuel Sunil
(NLR)
emmanuel.sunil@nlr.nl



Dr. Dennis Nieuwenhuisen
(NLR)
Dennis.nieuwenhuisen@nlr.nl



Prof. Dr. Jacco Hoekstra
(TUD)
j.m.hoekstra@tudelft.nl



Dr. Joost Ellerbroek
(TUD)
j.ellerbroek@tudelft.nl



Aleksandr 'Sasha' Vlaskin
PhD Candidate
NLR *and* TU Delft



Research Questions

1. Can a learning-based model outperform a classical iterative method (MVP) in terms of safety on a conflict resolution task with GNSS sensor error?
2. Does training a reinforcement learning model for CR on episodes with GNSS position noise improve its resultant performance?



Research Questions

1. Can a RL agent outperform the current baseline for CR with position error?
2. Should this agent be trained on said position noise?

Why do we need this topic?

- Studies: 400k commercial drones in EU by 2050
- Heavy investment in different applications



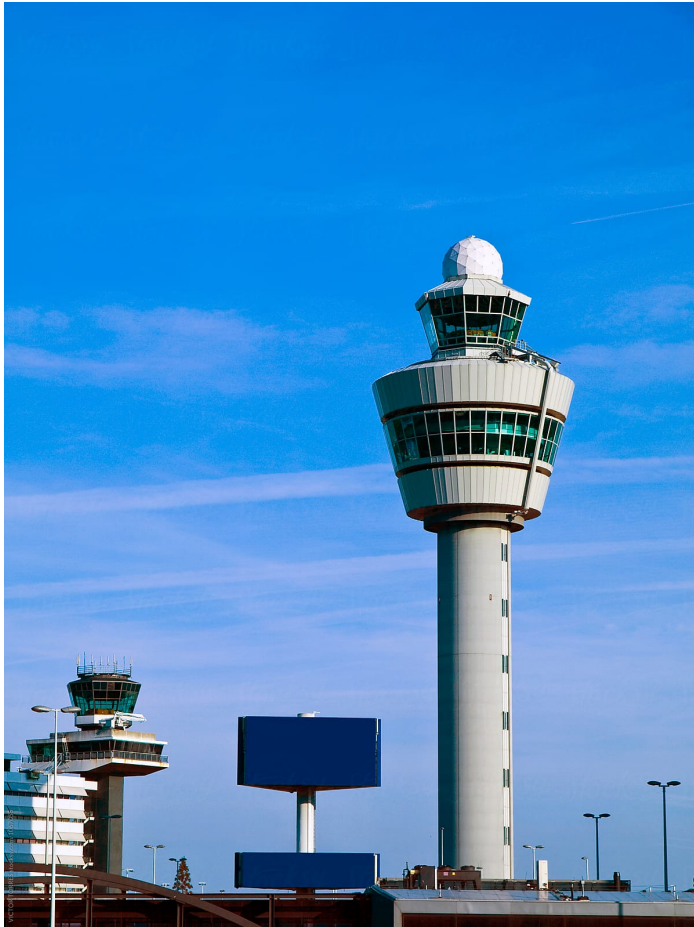
Source: Sabca UAS



Source: <https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries>

What is the issue?

- Conventional Air Traffic Control deals with large-scale problems



What is the issue?

Conventional:



0.01 to 0.08 aircraft per km²

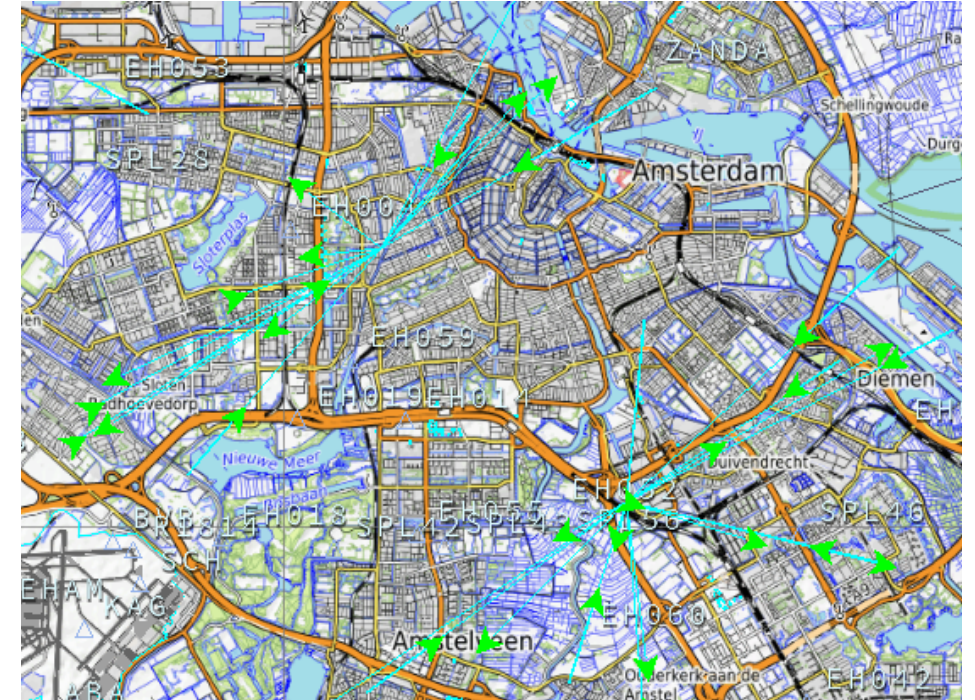
What is the issue?

Conventional:



0.01 to 0.08 aircraft per km²

Drone:



Up to 10 aircraft per km²

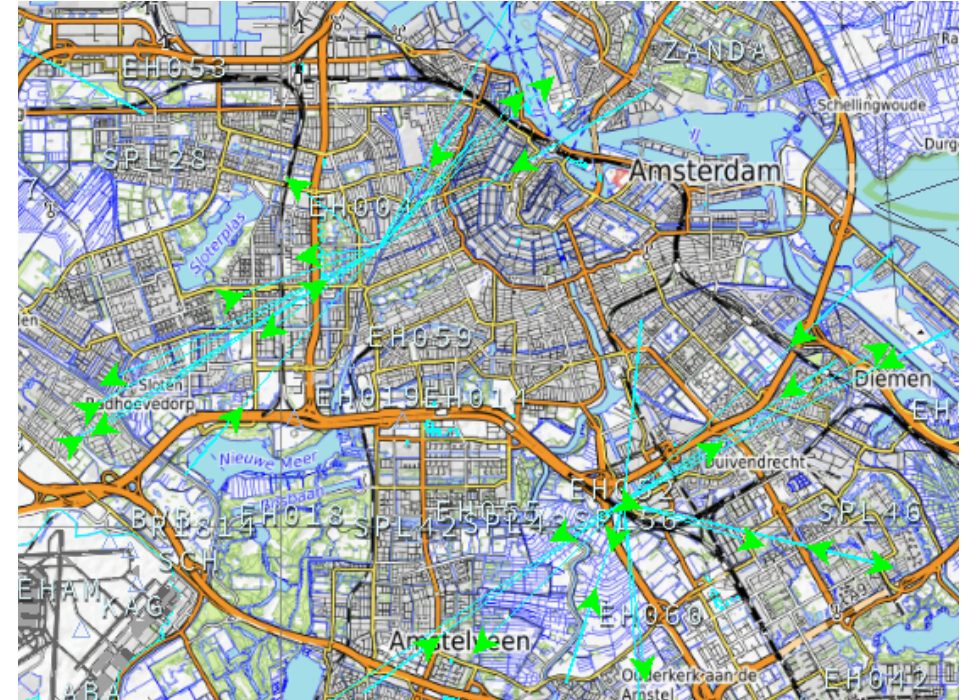
What is the issue?

Conventional:



0.01 to 0.08 aircraft per km²

Drone:



Up to 10 aircraft per km²

Impossible for Human ATCO!

My PhD Research

We propose an **Autonomous Separation Management System**

- Separation management: “keeping vehicles safely apart”
- System: layers should compliment each other



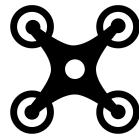
If we really distill it

Separation management = respecting personal space



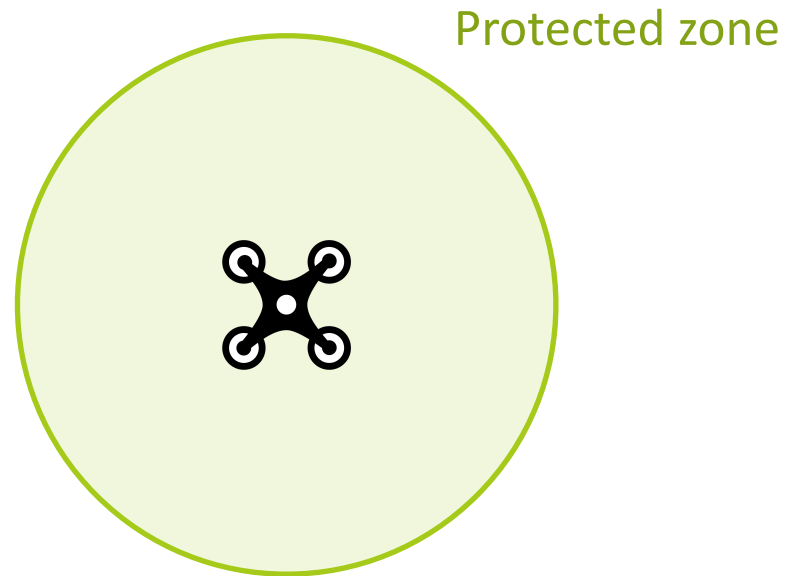
If we really distill it

Separation management = respecting personal space



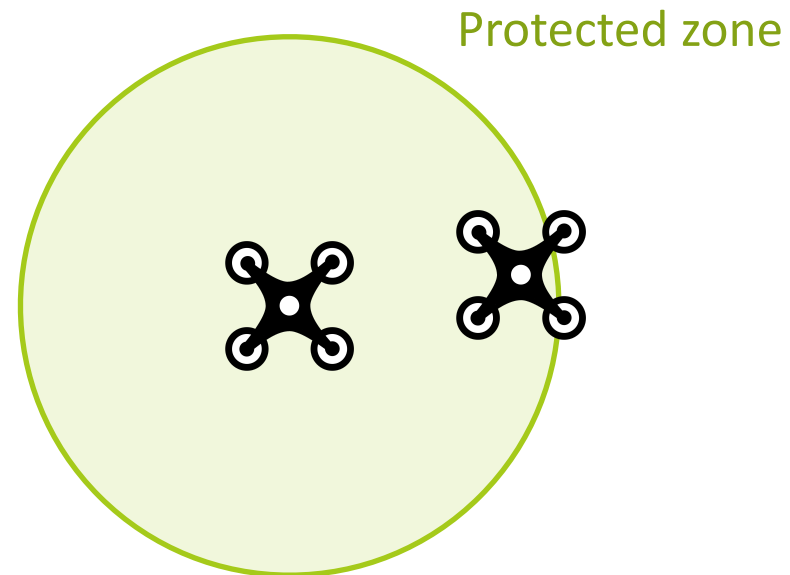
If we really distill it

Personal space = **protected zone**



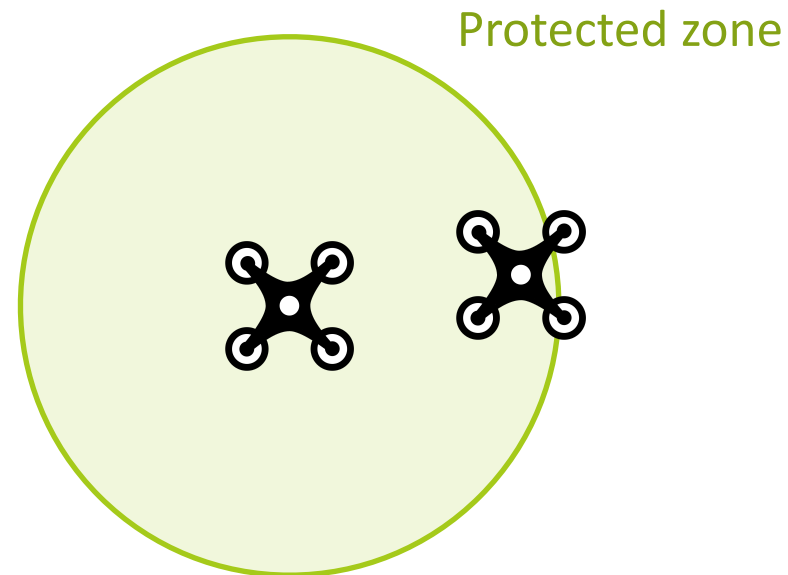
If we really distill it

Loss of Separation or **Intrusion**



If we really distill it

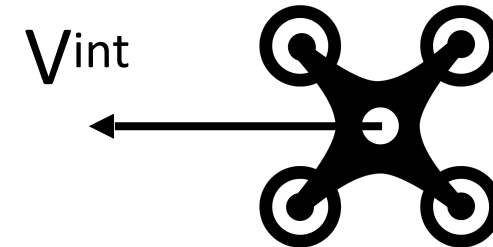
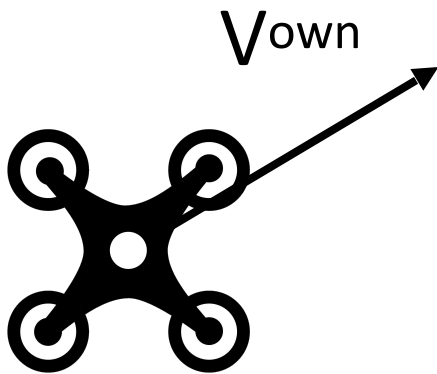
Loss of Separation or **Intrusion**



How to prevent this?

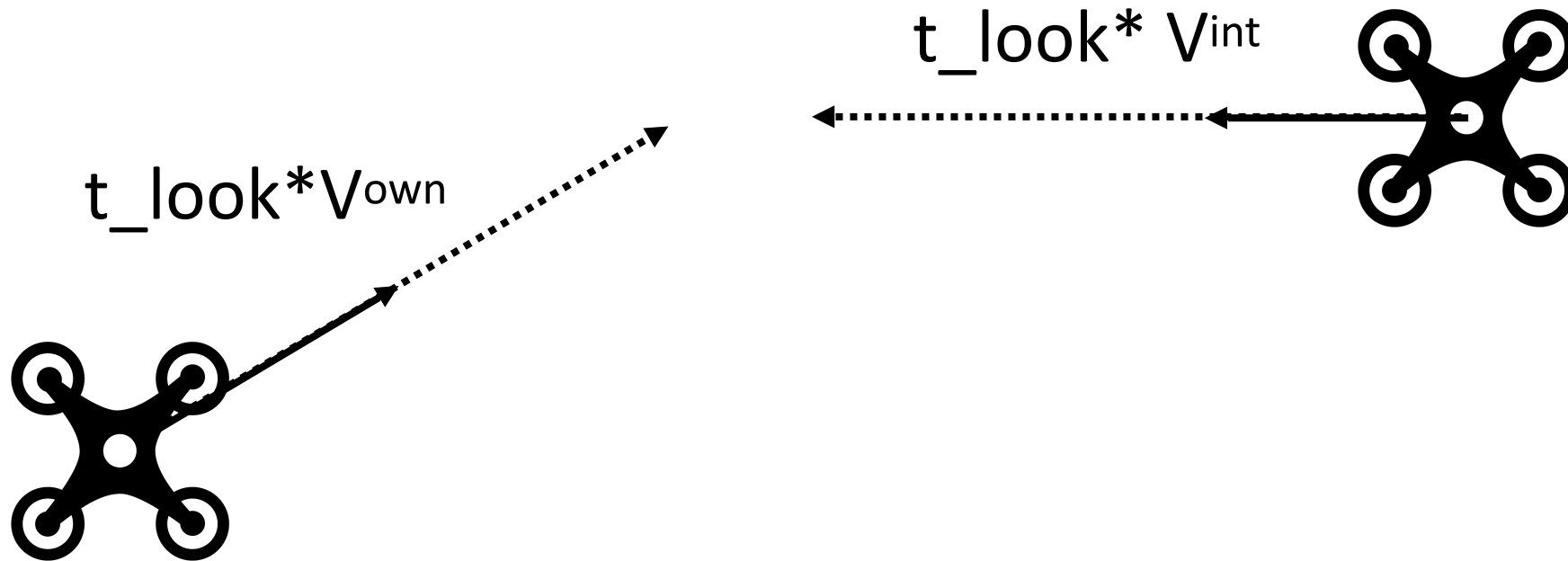
State-Based Conflict Detection

Speed and headings captured in **velocity vectors**



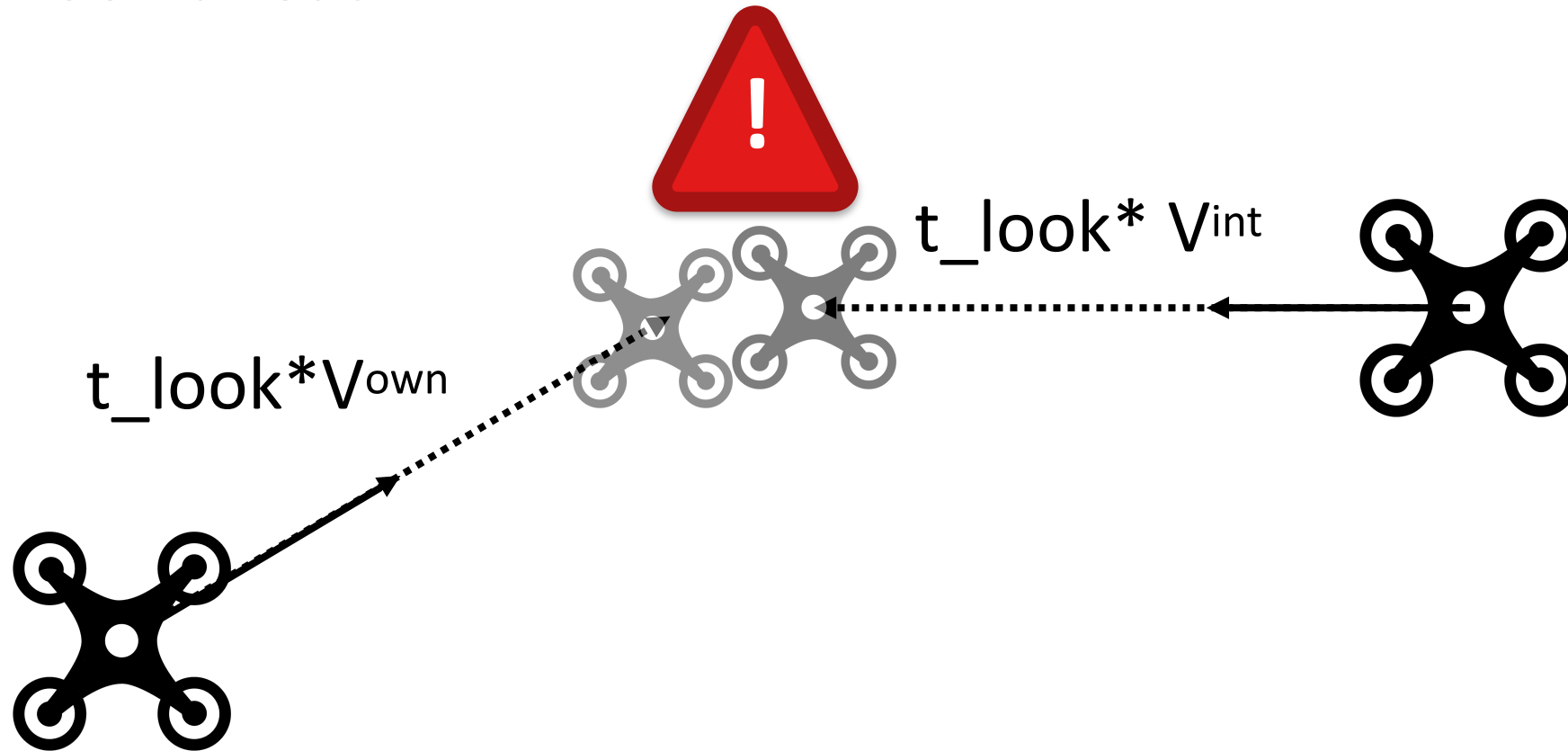
State-Based Conflict Detection

Let's look ahead in time



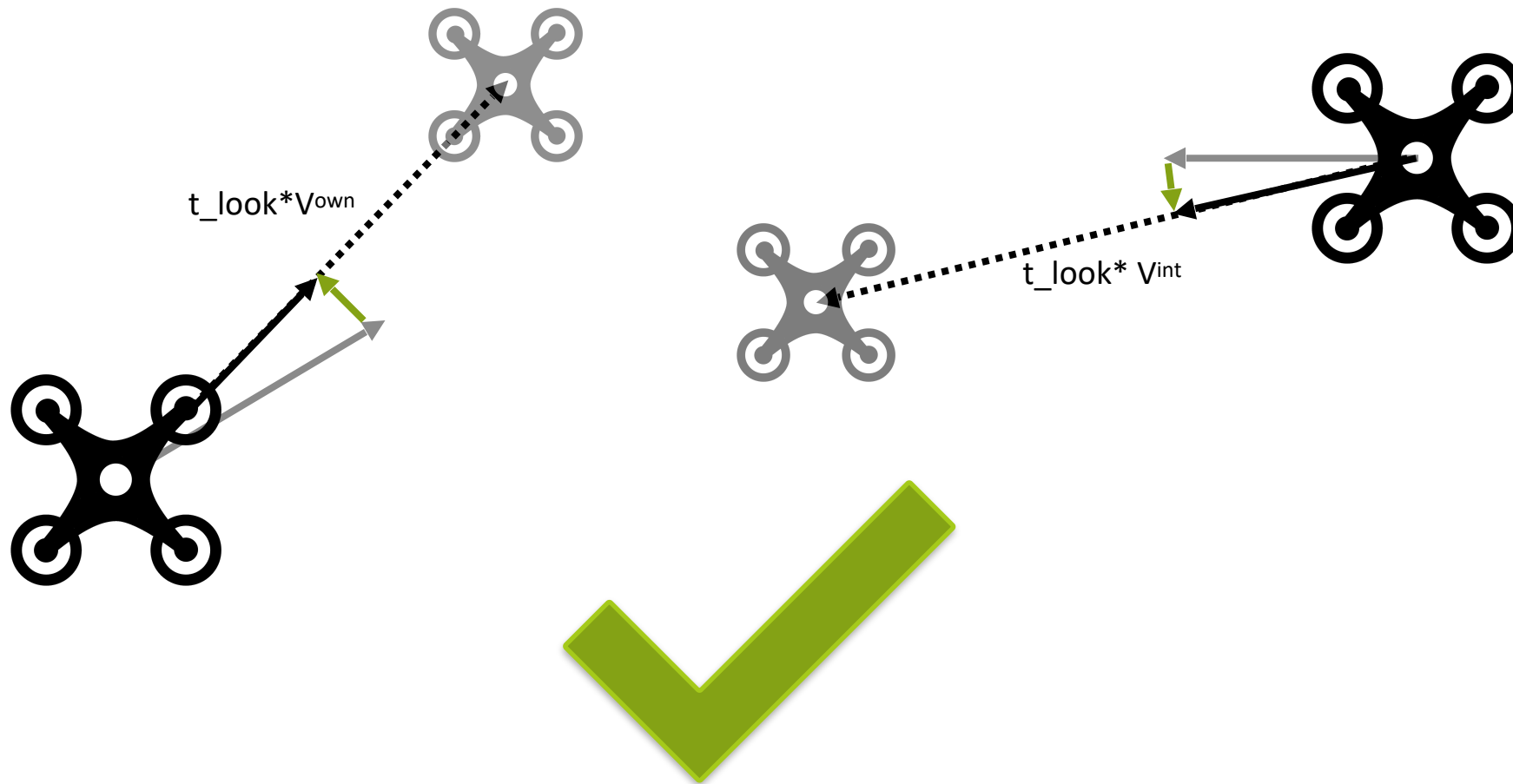
State-Based Conflict Detection

Let's look ahead in time



State-Based Conflict Detection and Resolution

Let's resolve our conflict by **turning**

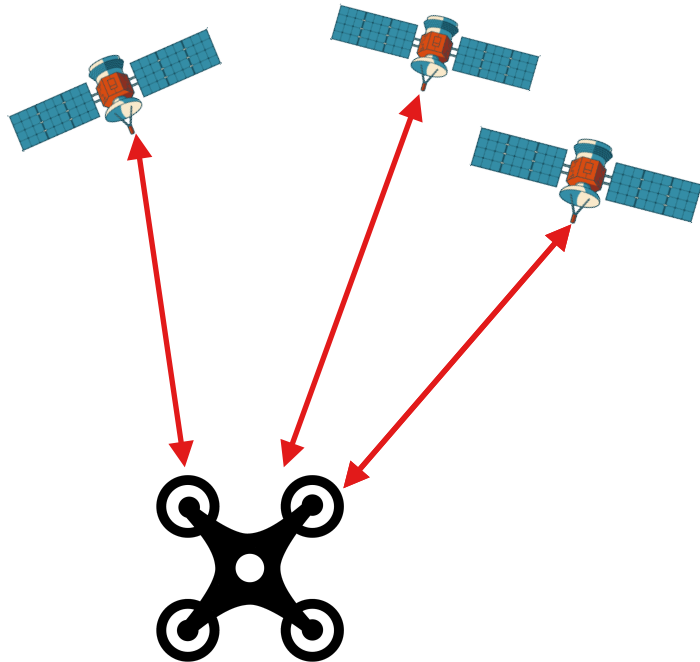


What is the problem?

- **State-based** requires that you **know your state**

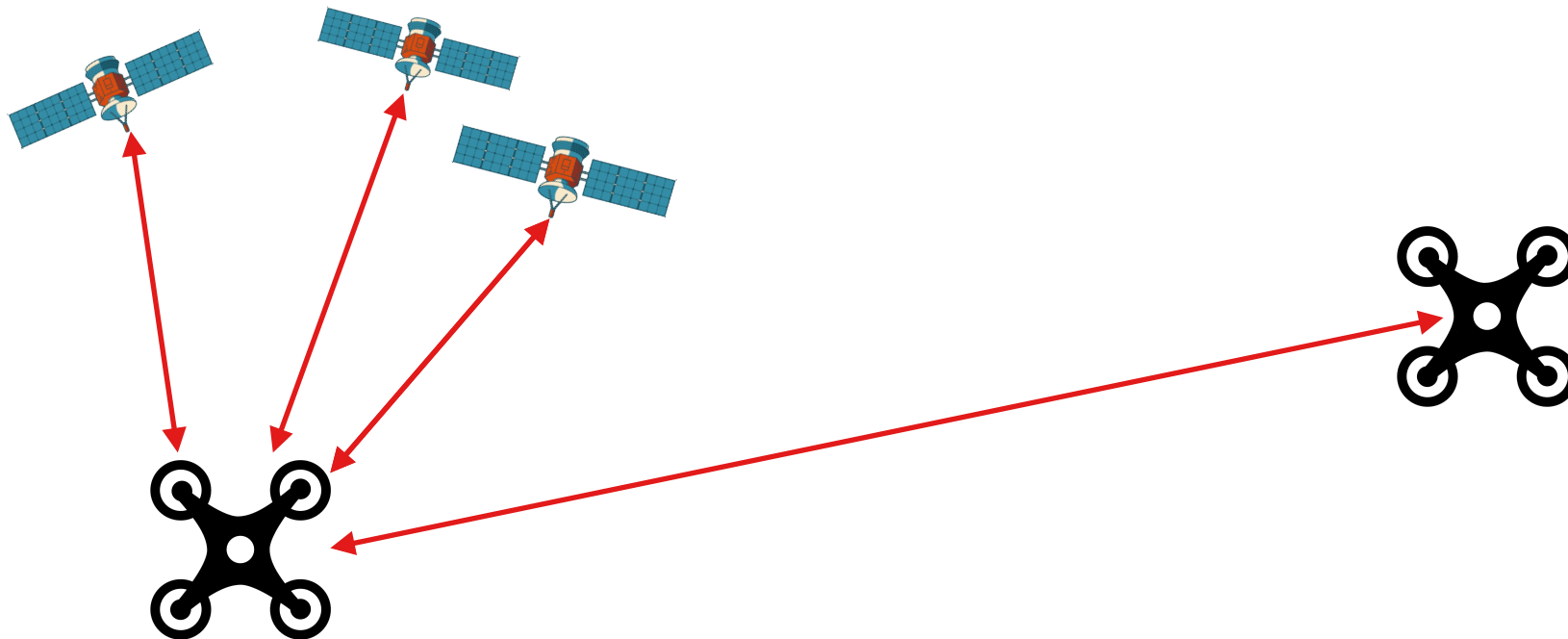
What is the problem?

- **State-based** requires that you **know your state**
- This is derived from GNSS!

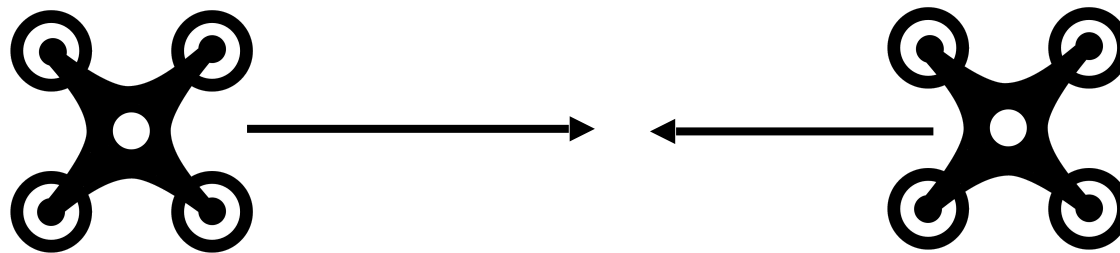


What is the problem?

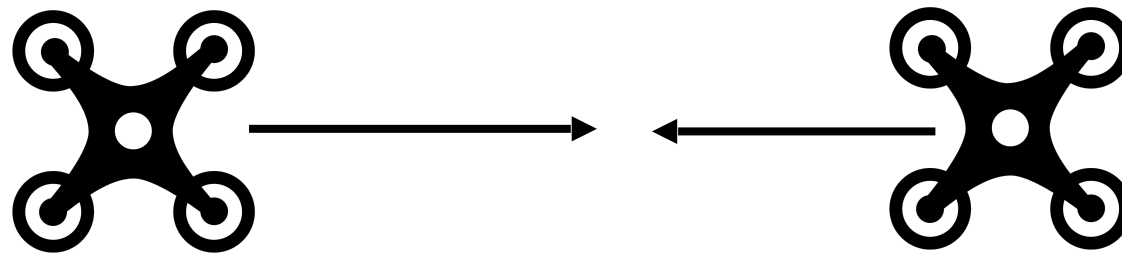
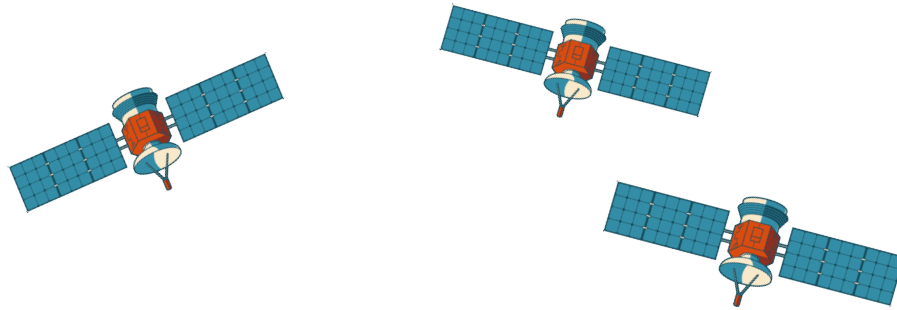
- **State-based** requires that you **know your state**
- This is derived from GNSS!
- And broadcast via ADS-L



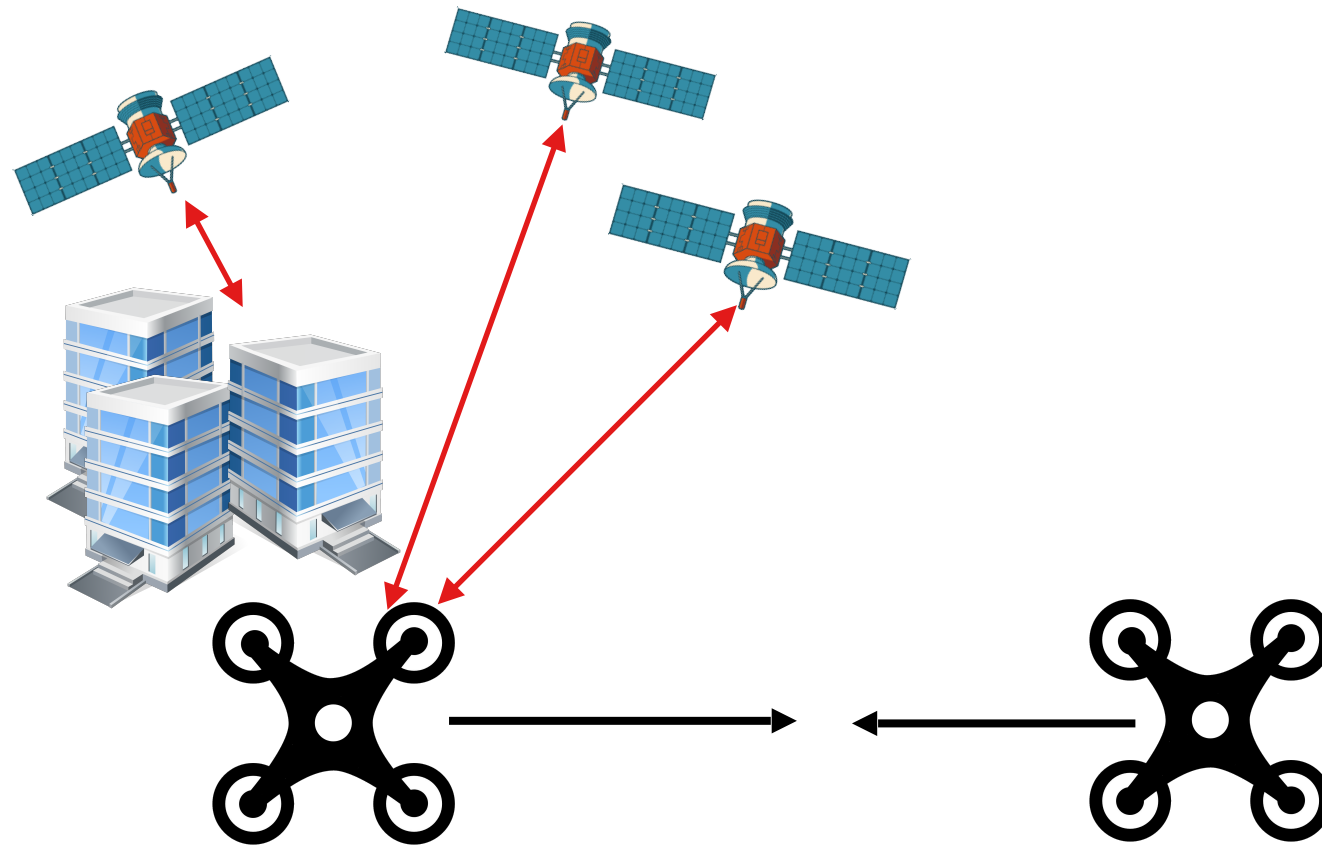
The problem:



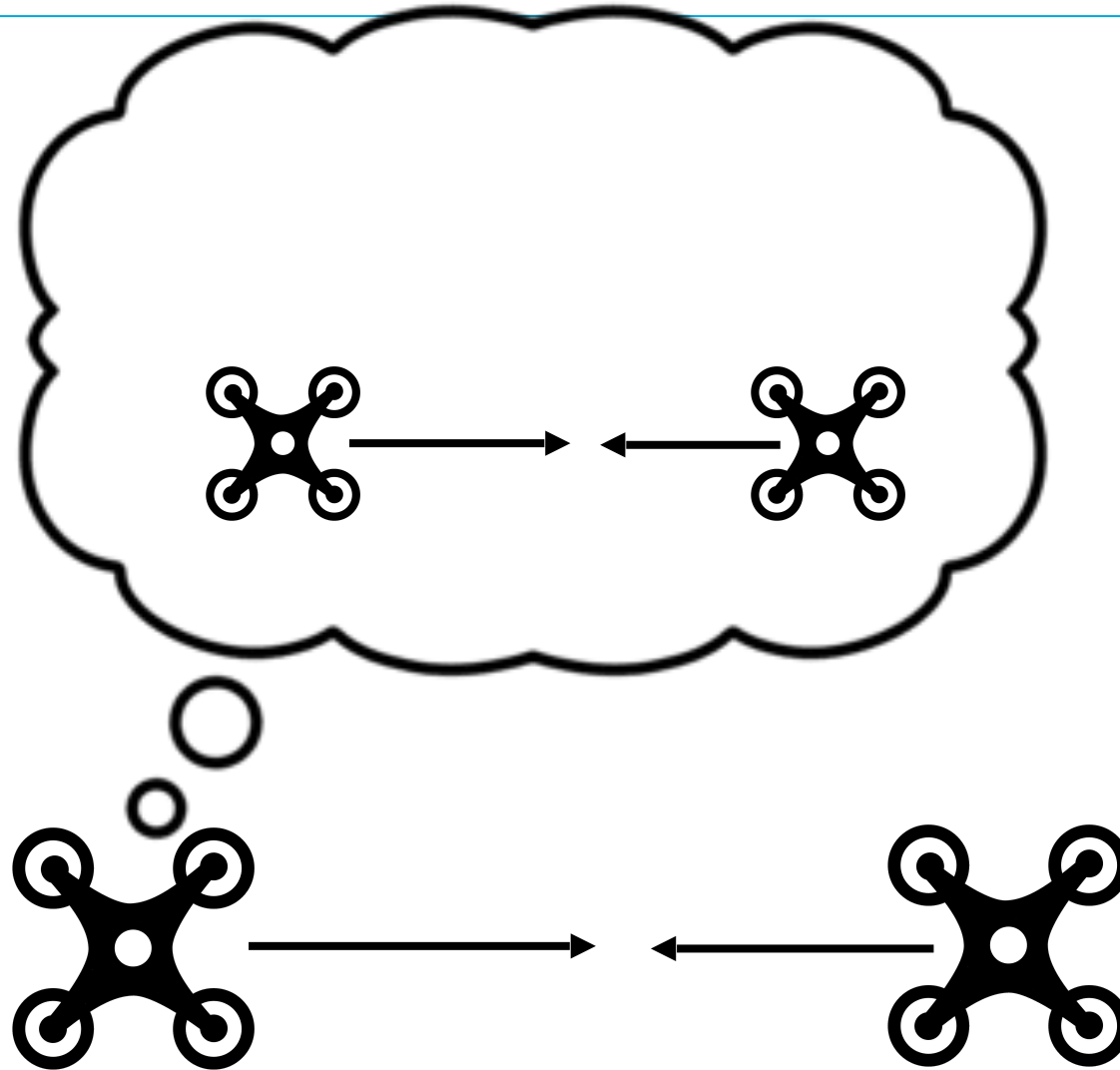
The problem:



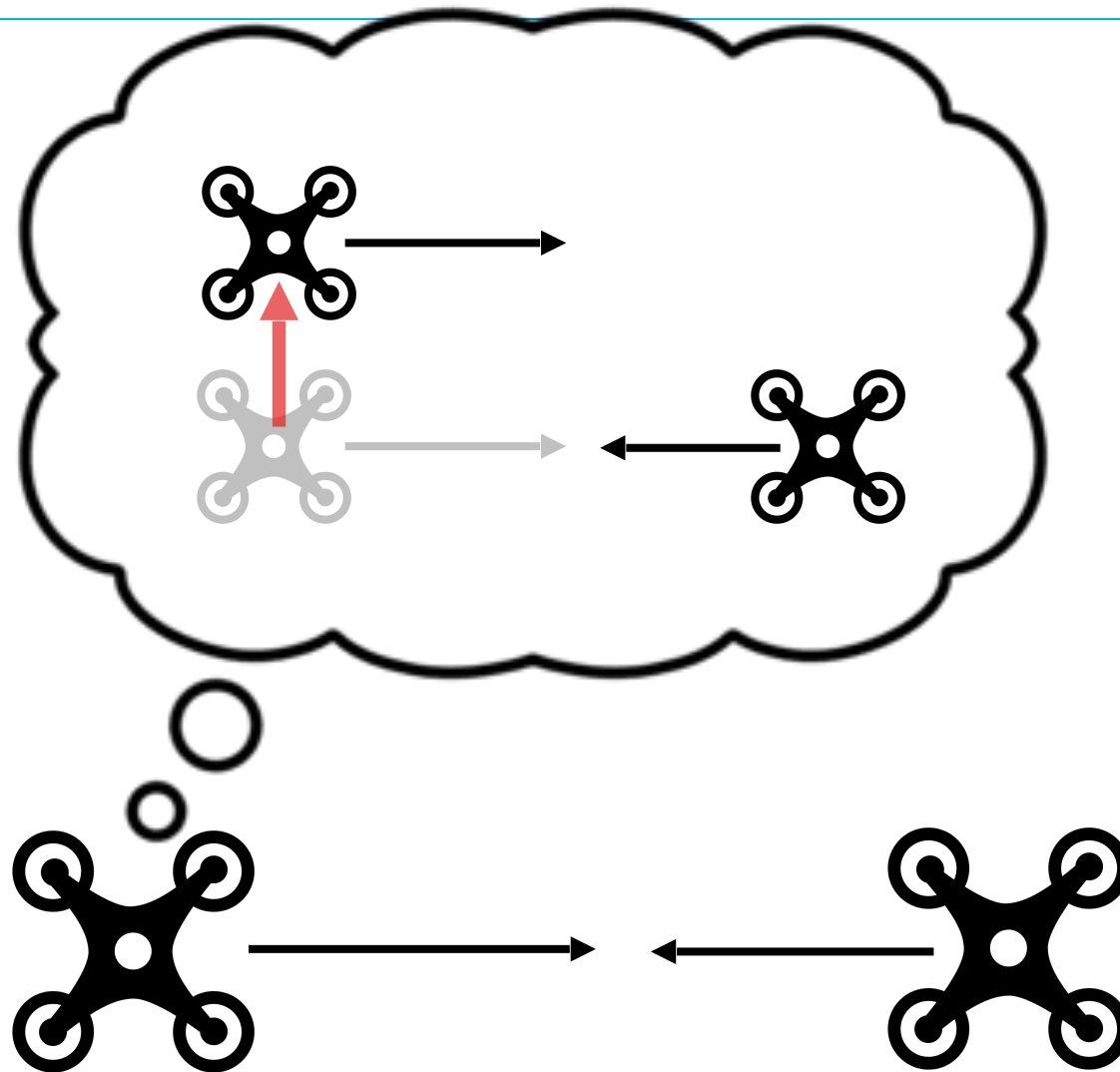
The problem:



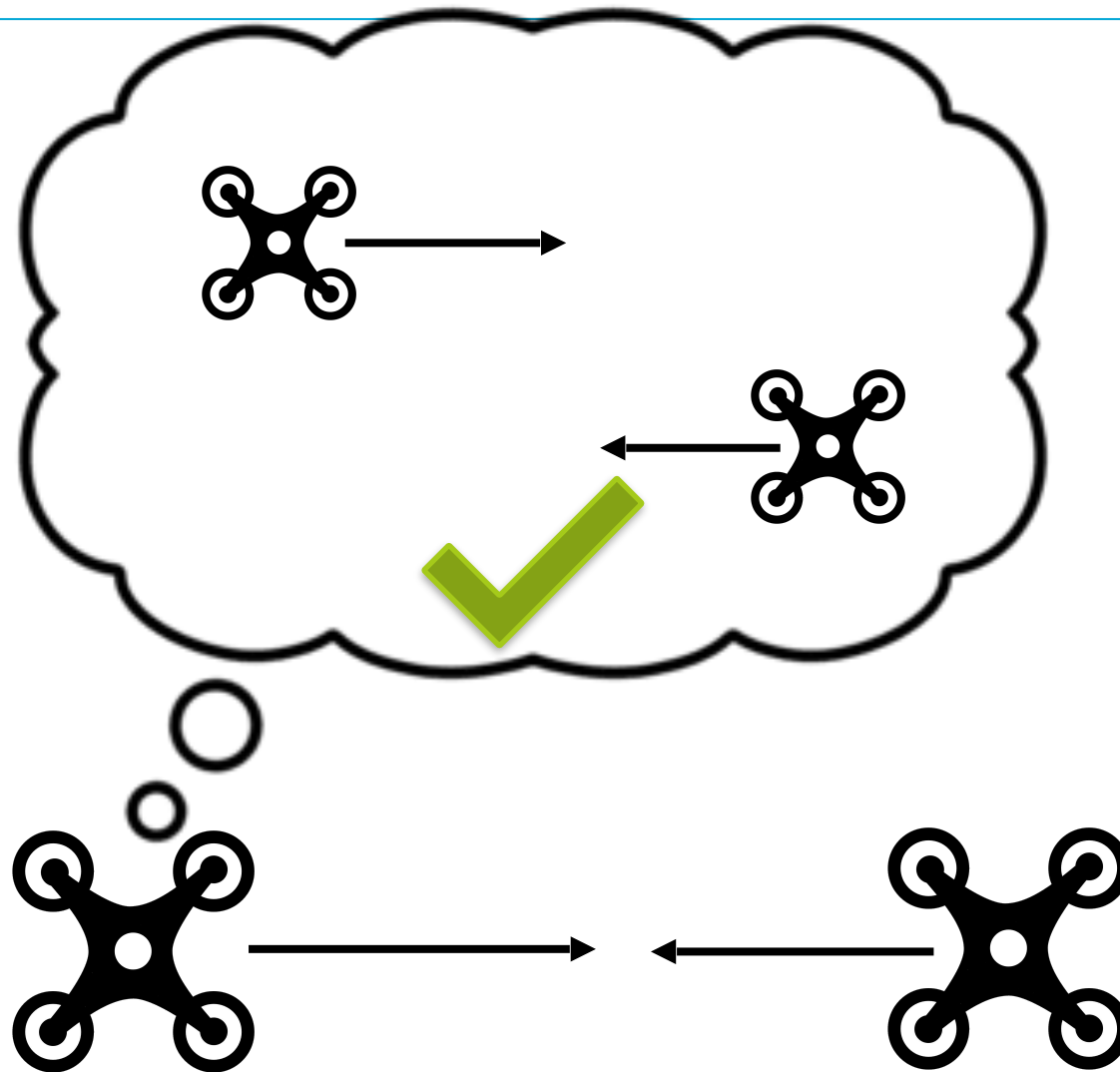
The problem:



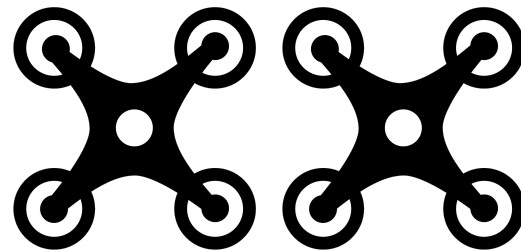
The problem:



The problem:



The problem:



Is this position noise a problem?

- Yes! ^[1]

[1] Rahman, M. F., Vlaskin, A., Ellerbroek, J., & Hoekstra, J. (2024). *Autonomous Separation in U-Space: Assessing the Impact of Position Uncertainty*.

Is this position noise a problem?

- Yes! ^[1]
- Up to 4% more intrusions on average under realistic noise

[1] Rahman, M. F., Vlaskin, A., Ellerbroek, J., & Hoekstra, J. (2024). *Autonomous Separation in U-Space: Assessing the Impact of Position Uncertainty*.

Is this position noise a problem?

- Yes! ^[1]
- Up to 4% more intrusions on average under realistic noise
- Can we make it better?

[1] Rahman, M. F., Vlaskin, A., Ellerbroek, J., & Hoekstra, J. (2024). *Autonomous Separation in U-Space: Assessing the Impact of Position Uncertainty*.

Is this position noise a problem?

- Yes! ^[1]
- Up to 4% more intrusions on average under realistic noise
- Can we make it better?

Reinforcement Learning

[1] Rahman, M. F., Vlaskin, A., Ellerbroek, J., & Hoekstra, J. (2024). *Autonomous Separation in U-Space: Assessing the Impact of Position Uncertainty*.

Other works using RL in the ATM context

“Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning” — *Ribeiro, M., Ellerbroek, J & Hoekstra J.M., 2022:*

RL agents control heading, speed and altitude to deconflict aircraft; can match or surpass geometric CR in moderate densities by adapting manoeuvres to conflict geometry.

“A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance” — *Brittain, Yang & Wei, 2020:*

Attention-augmented RL agents (centralized training, decentralized execution) that handle variable numbers of intruder aircraft; validated in simulation, showing efficient separation assurance in dynamic, high-density airspace.

“Using Relative State Transformer Models for Multi-Agent Reinforcement Learning in Air Traffic Control” — *Groot, D.J. & Ellerbroek, J. & Hoekstra, J.M. 2023*

Introduces a transformer-based approach (relative-state encoding) to handle variable number of agents (aircraft) in multi-agent RL for ATC, improving scalability and flexibility when the traffic load fluctuates.

Other works using RL in the ATM context

RL agents control heading, speed and altitude

Attention-augmented RL agents

Transformer-based approach



What is Reinforcement Learning?

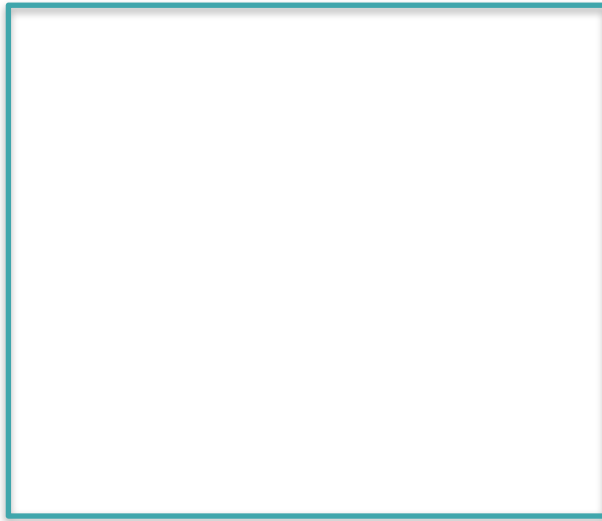
- “Learning from experience”

Reinforcement Learning

- Learning from experience
- Possible to **train on noise**?

How to make this a RL Problem?

Key Components:



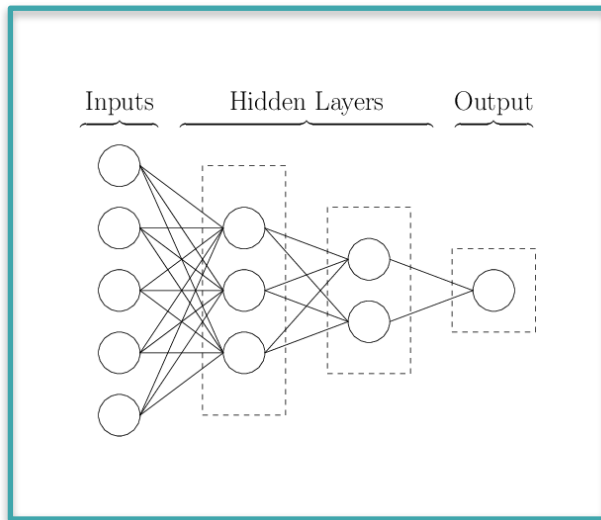
Model



Environment

How to make this a RL Problem?

Key Components:



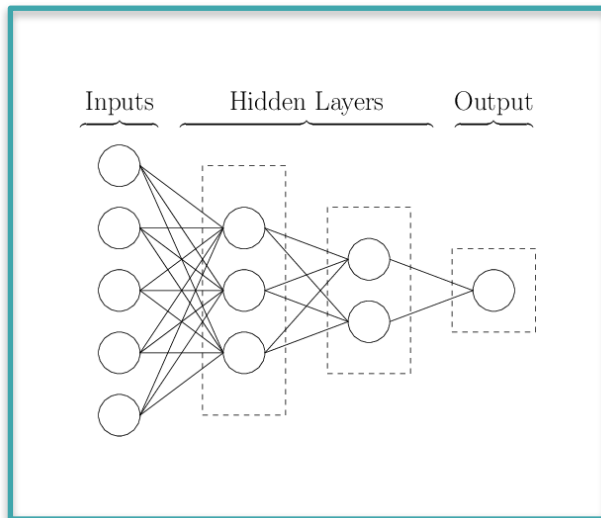
Model



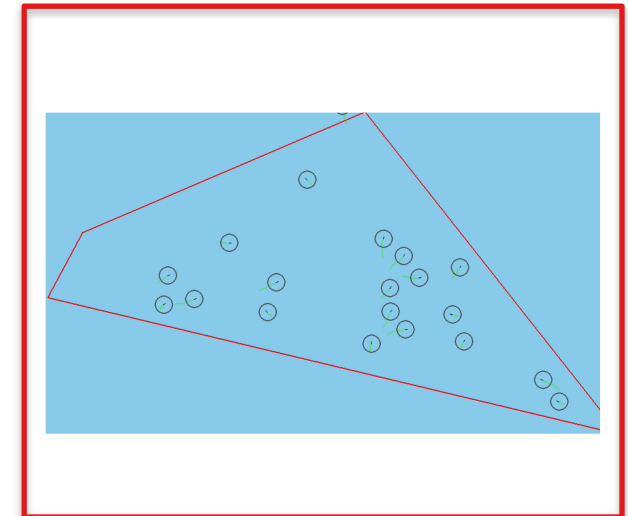
Environment

How to make this a RL Problem?

Key Components:



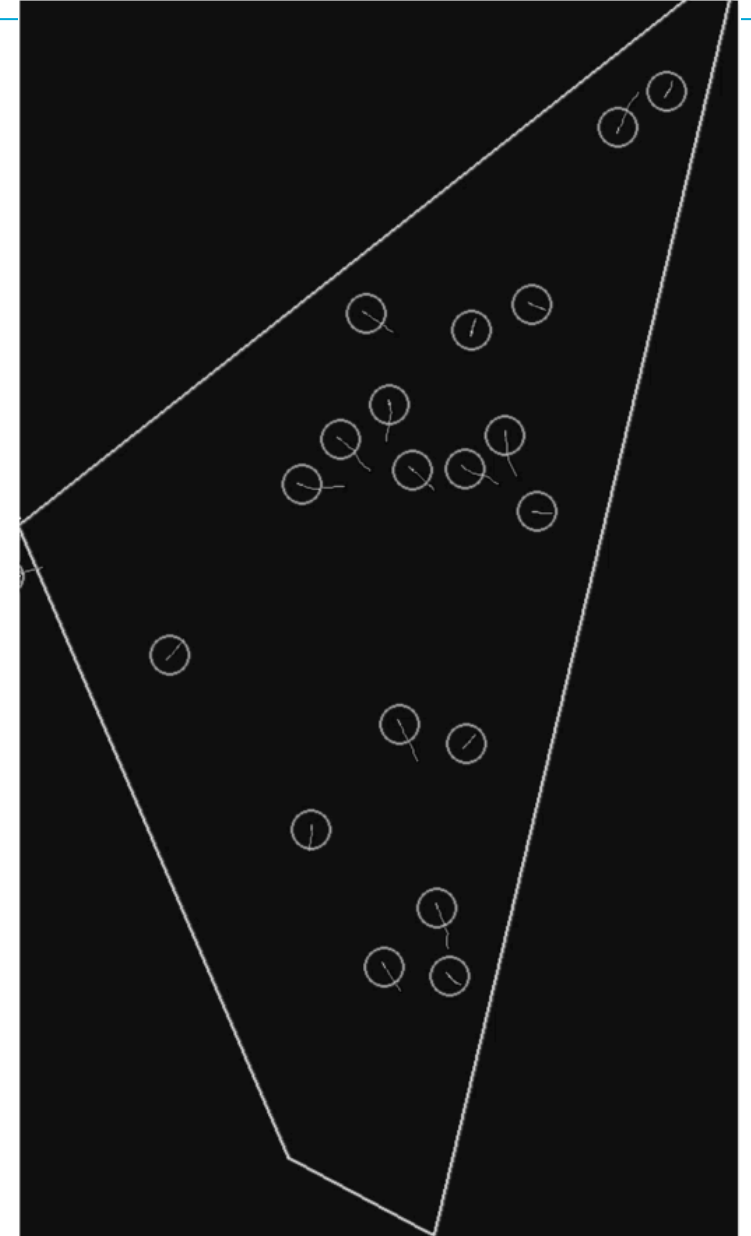
Model



Environment

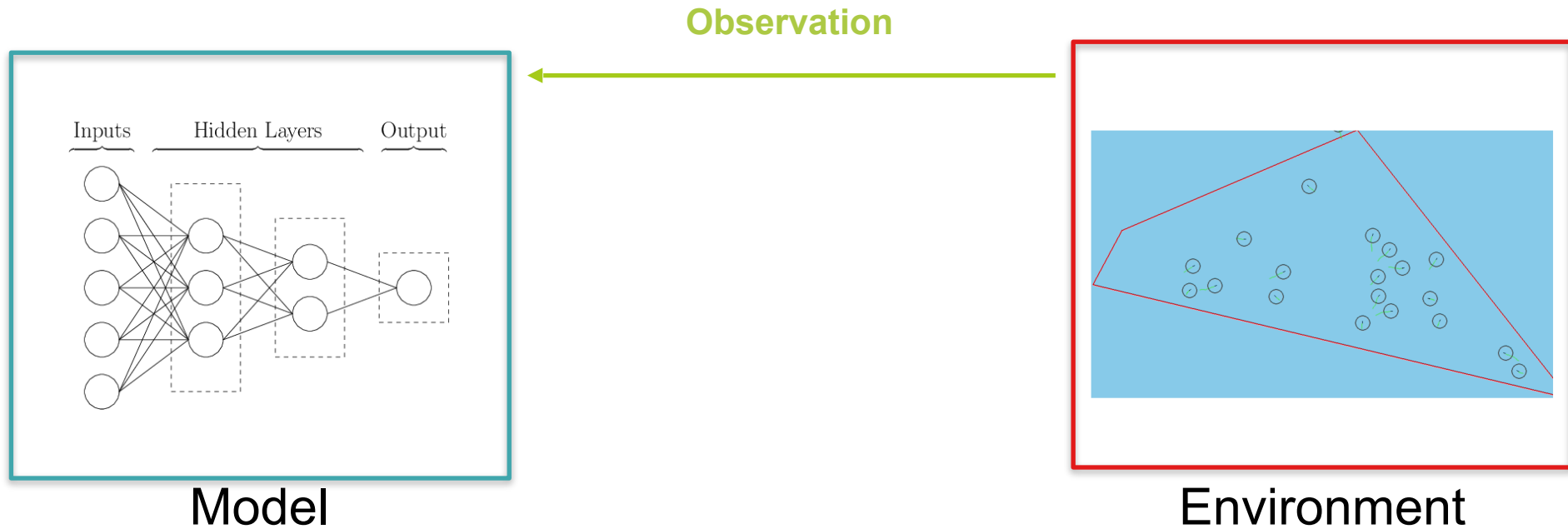
RL Environment: CR in a Sector

- Multi-vehicle **sector** scenario
- Vehicles must head to prescribed waypoints while avoiding others

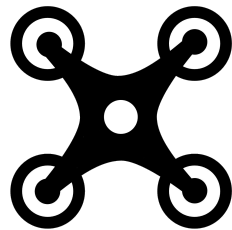


How to make this a RL Problem?

Key Components:

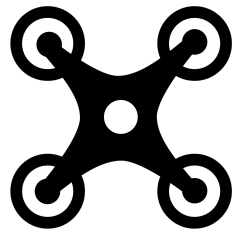


Observation Vector

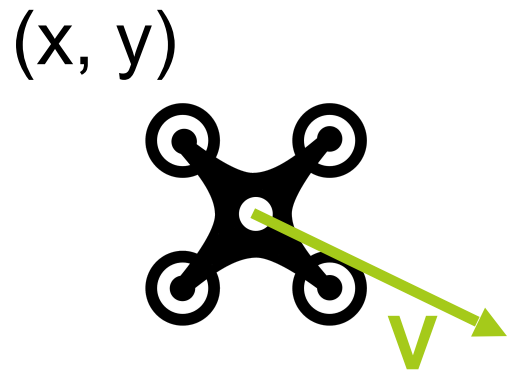


Observation Vector

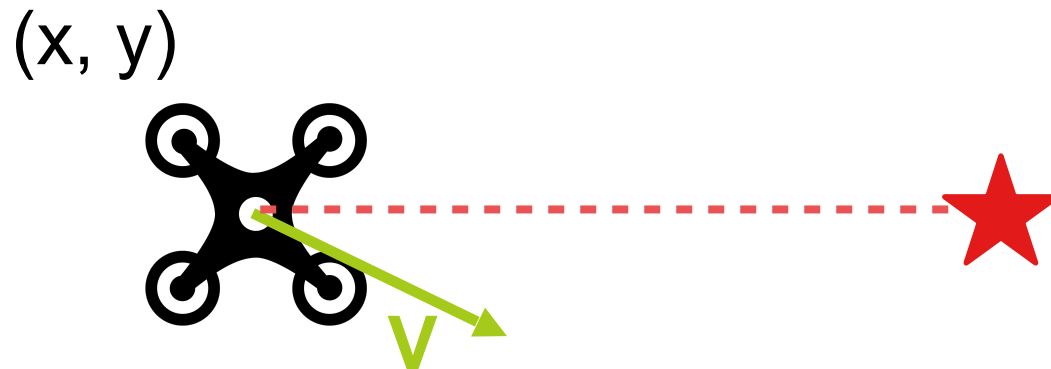
(x, y)



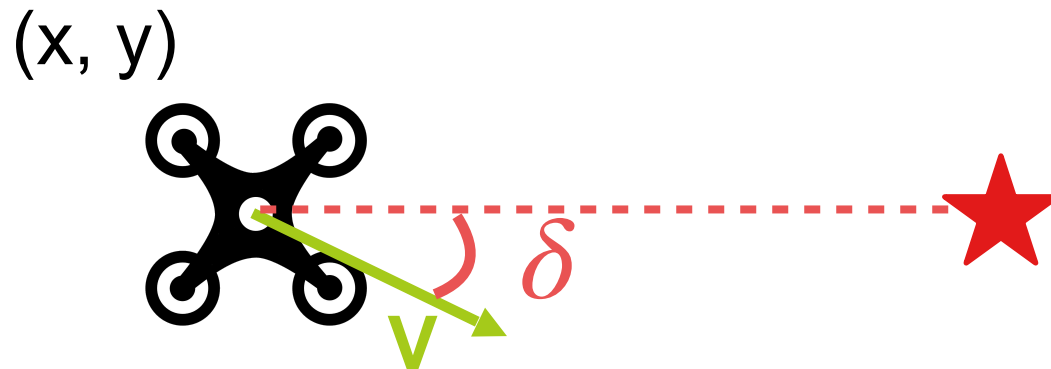
Observation Vector



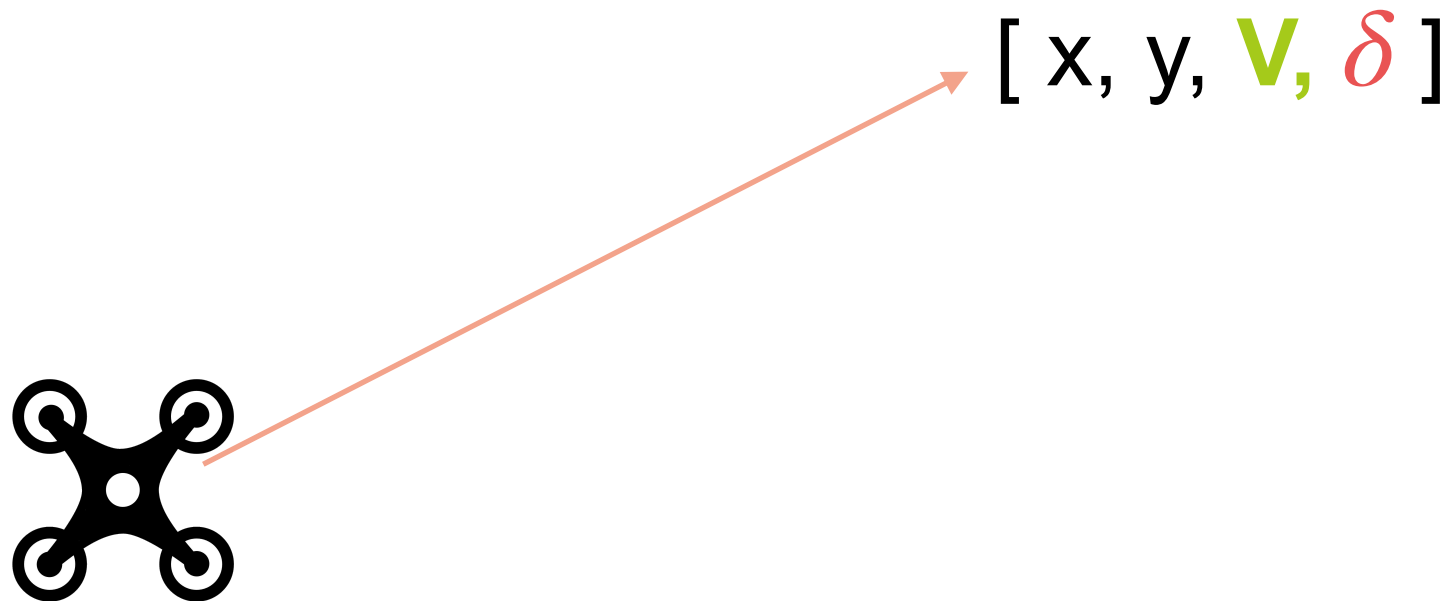
Observation Vector



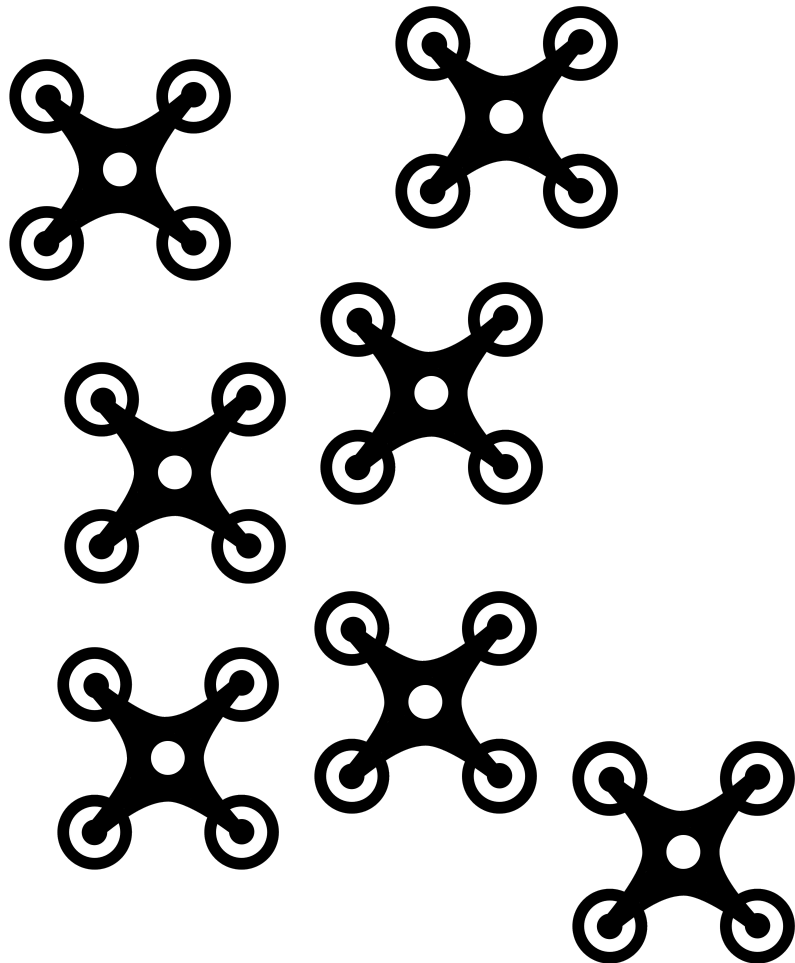
Observation Vector



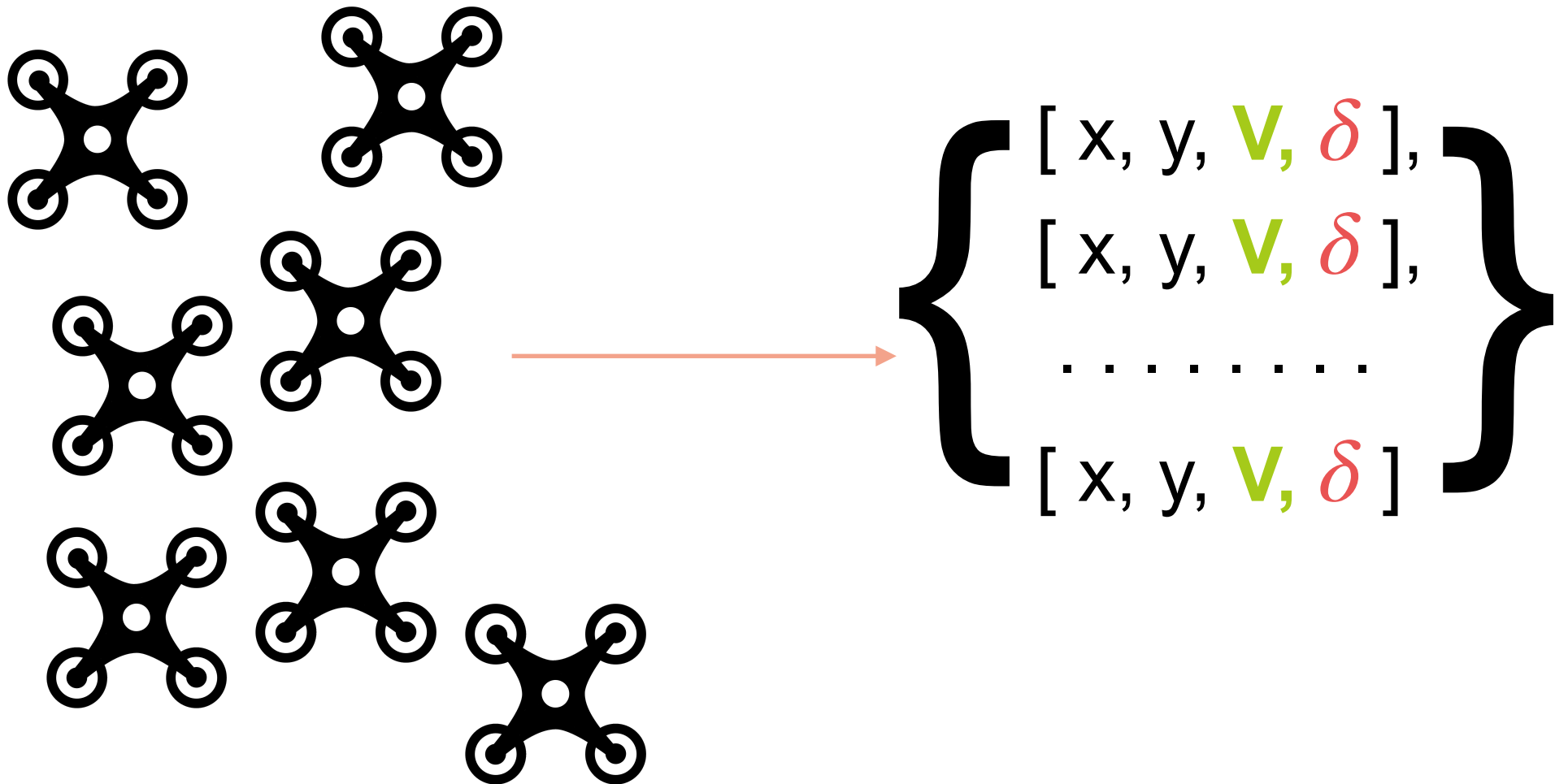
Observation Vector



Observation Vector



Observation Vector

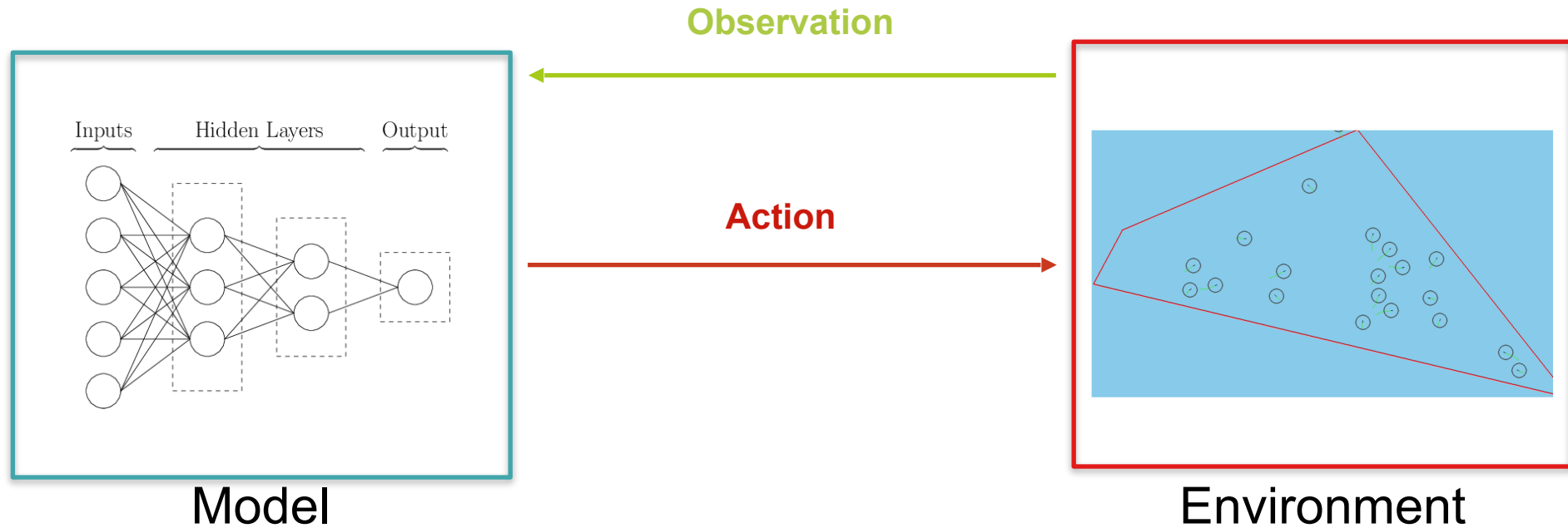


Observation Vector

Category	Variable	Description	Size
Ownship	$\cos(\delta)$	Cosine of the drift angle	1
	$\sin(\delta)$	Sine of the drift angle	
	v	Ground Speed (m/s)	
	x	x -position (m)	
	y	y -position (m)	
	v_x	x -component of velocity (m/s)	
	v_y	y -component of velocity (m/s)	

How to make this a RL Problem?

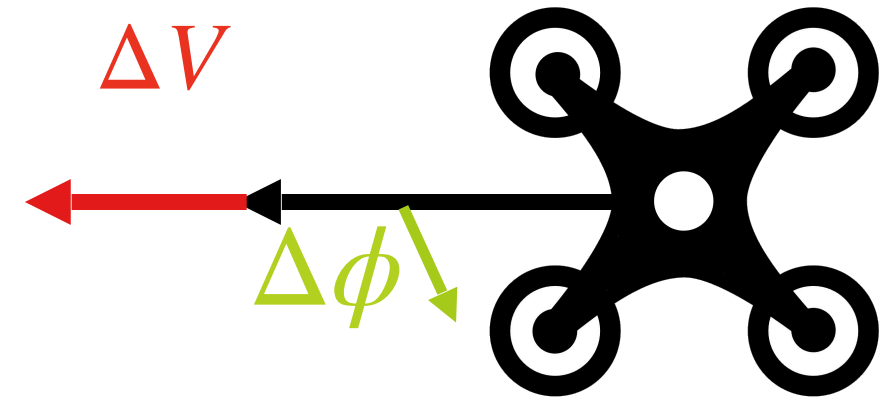
Key Components:



Action Space

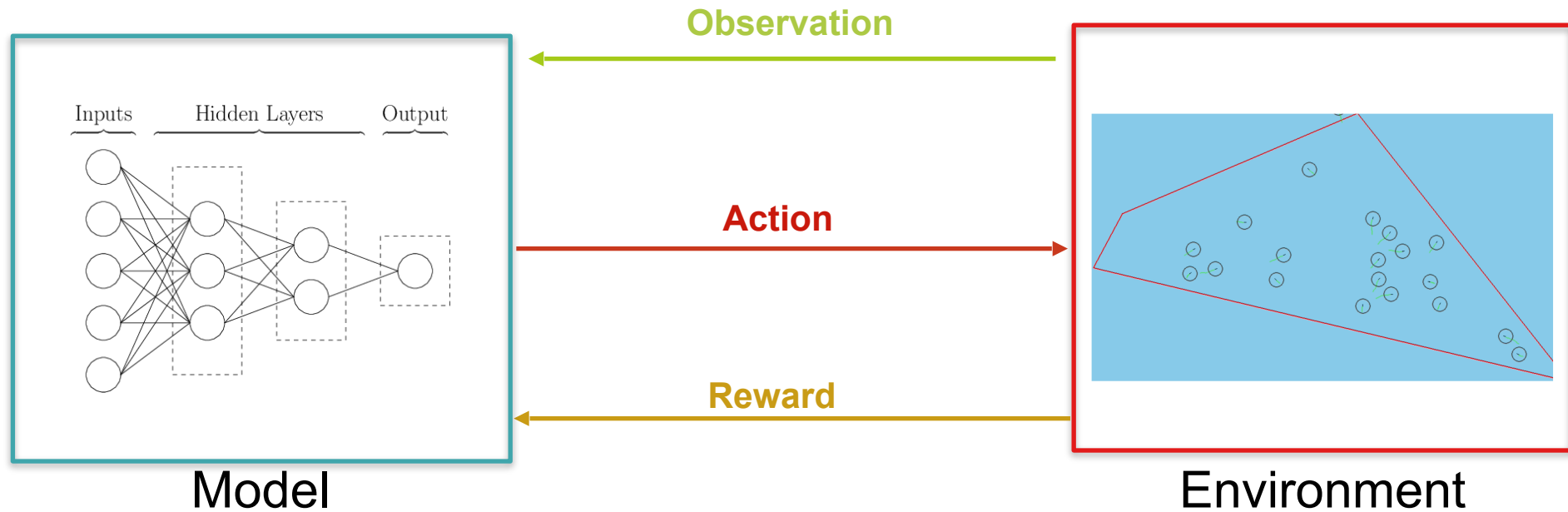
The network takes an action based on what it observes:

Action	Bounds
Heading Change	$[-1,1] \rightarrow [-22.5^\circ, +22.5^\circ]$
Speed Change	$[-1,1] \rightarrow [-10 \text{ kts}, +10 \text{ kts}]$



How to make this a RL Problem?

Key Components:



Reward Function

$$r = \text{Efficiency} + \text{Safety}$$

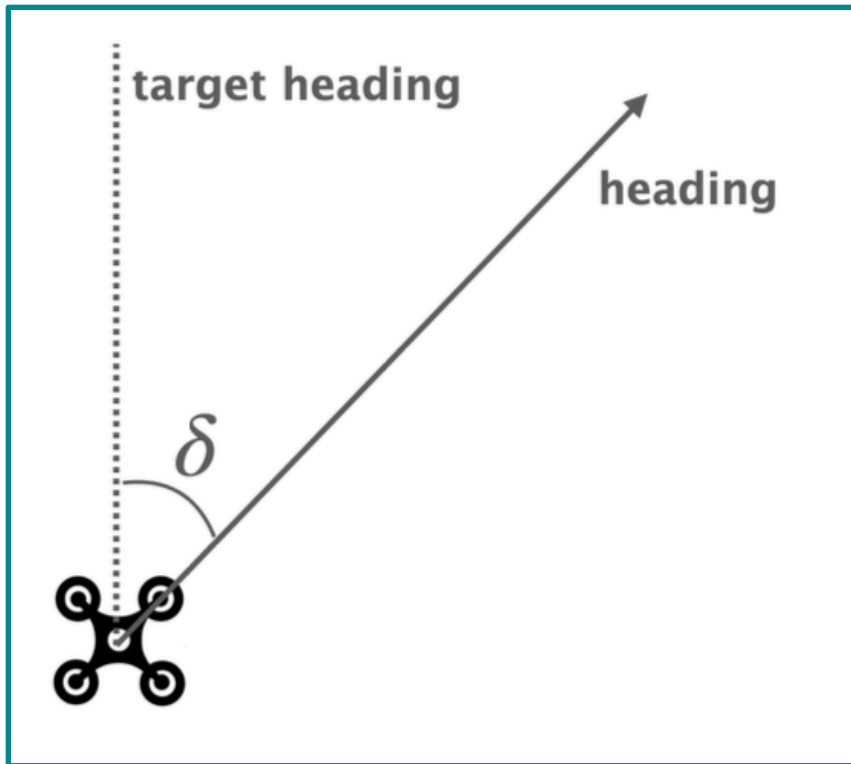


Reward Function

$$r = R_{drift} \cdot \phi_{dr} + \begin{cases} 0 & \text{if not in intrusion} \\ -1 & \text{if in intrusion} \end{cases}$$

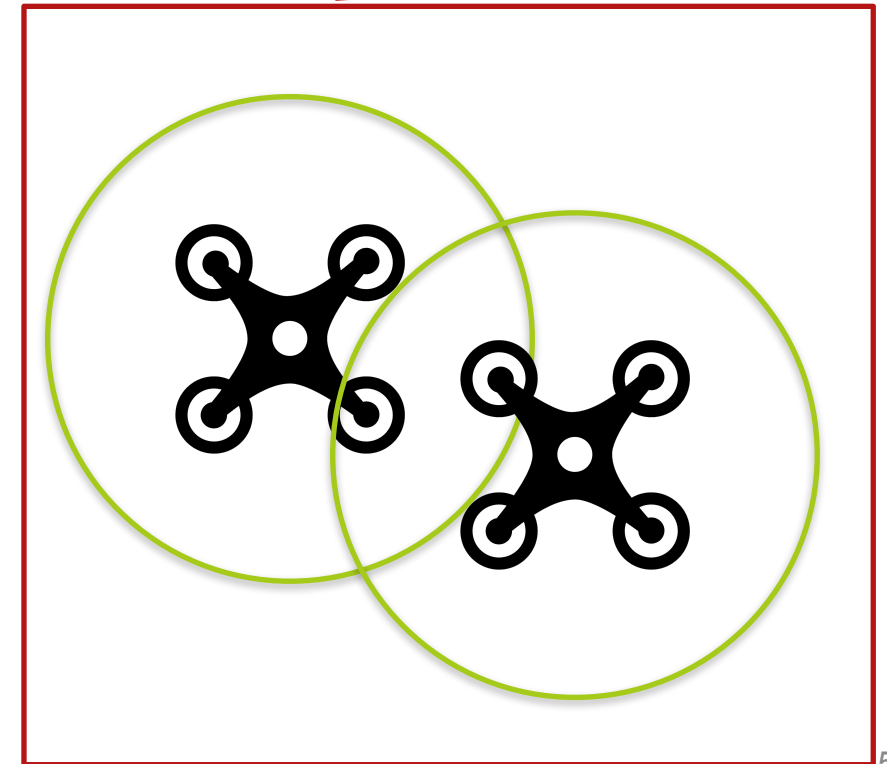
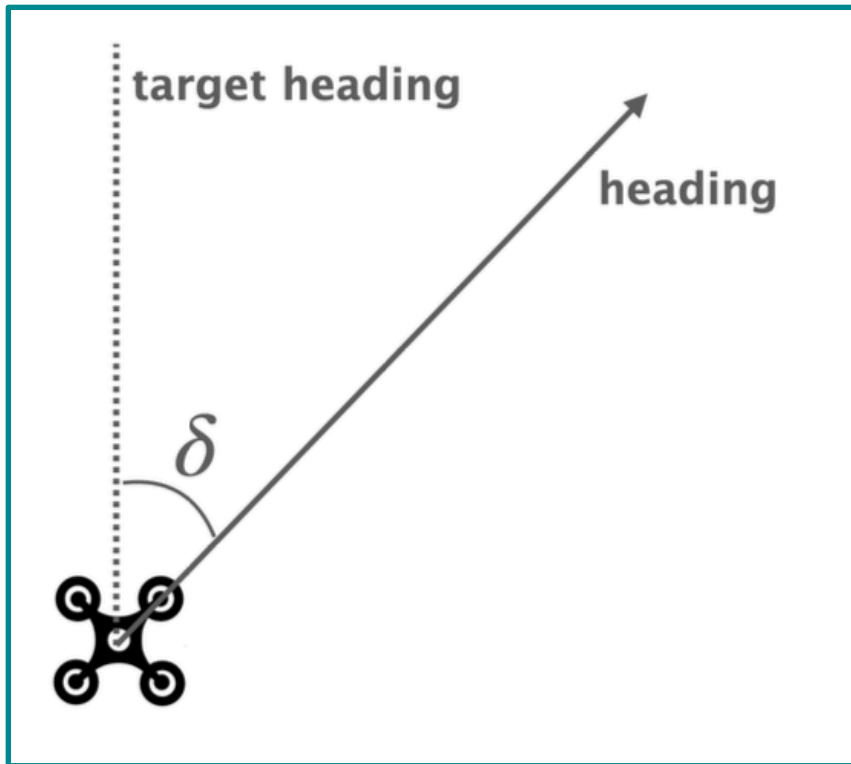
Reward Function

$$r = \boxed{R_{drift} \cdot \phi_{dr}} + \begin{cases} 0 & \text{if not in intrusion} \\ -1 & \text{if in intrusion} \end{cases}$$

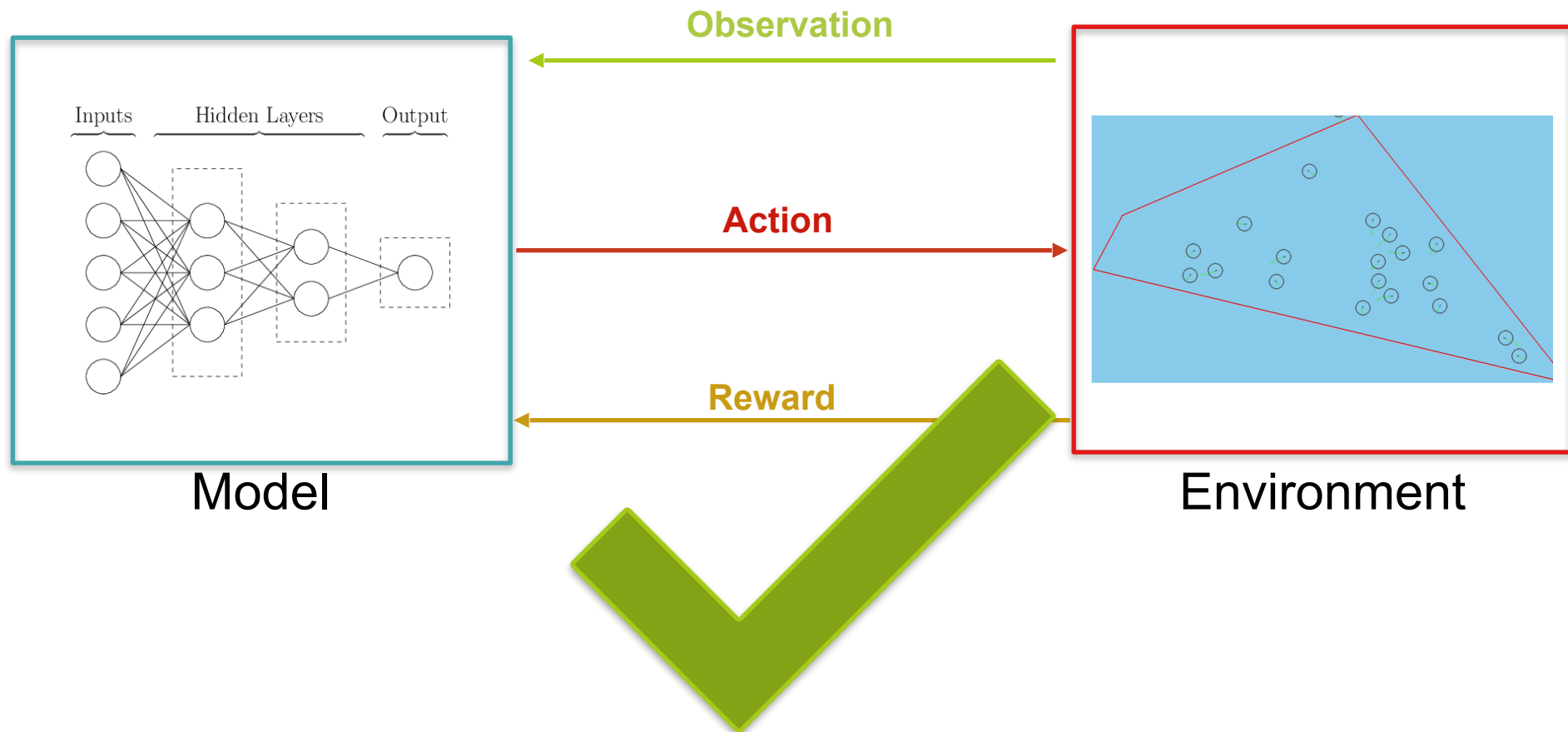


Reward Function

$$r = \boxed{R_{drift} \cdot \phi_{dr}} + \begin{cases} 0 & \text{if not in intrusion} \\ -1 & \text{if in intrusion} \end{cases}$$



How to make this a RL Problem?



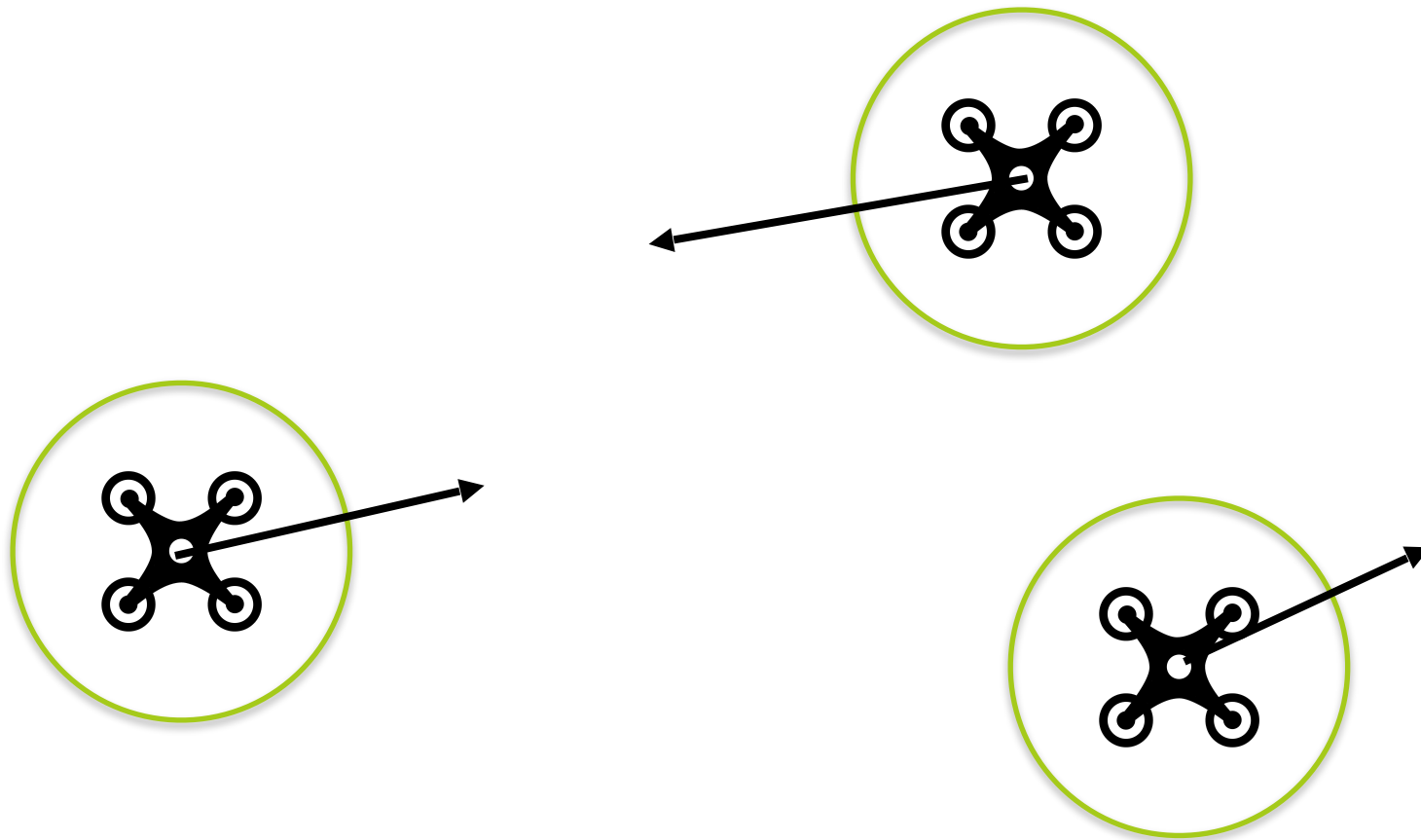
Reinforcement Learning Algorithm

- Soft Actor-Critic
- Off-Policy method
- Trains several critic and actor networks



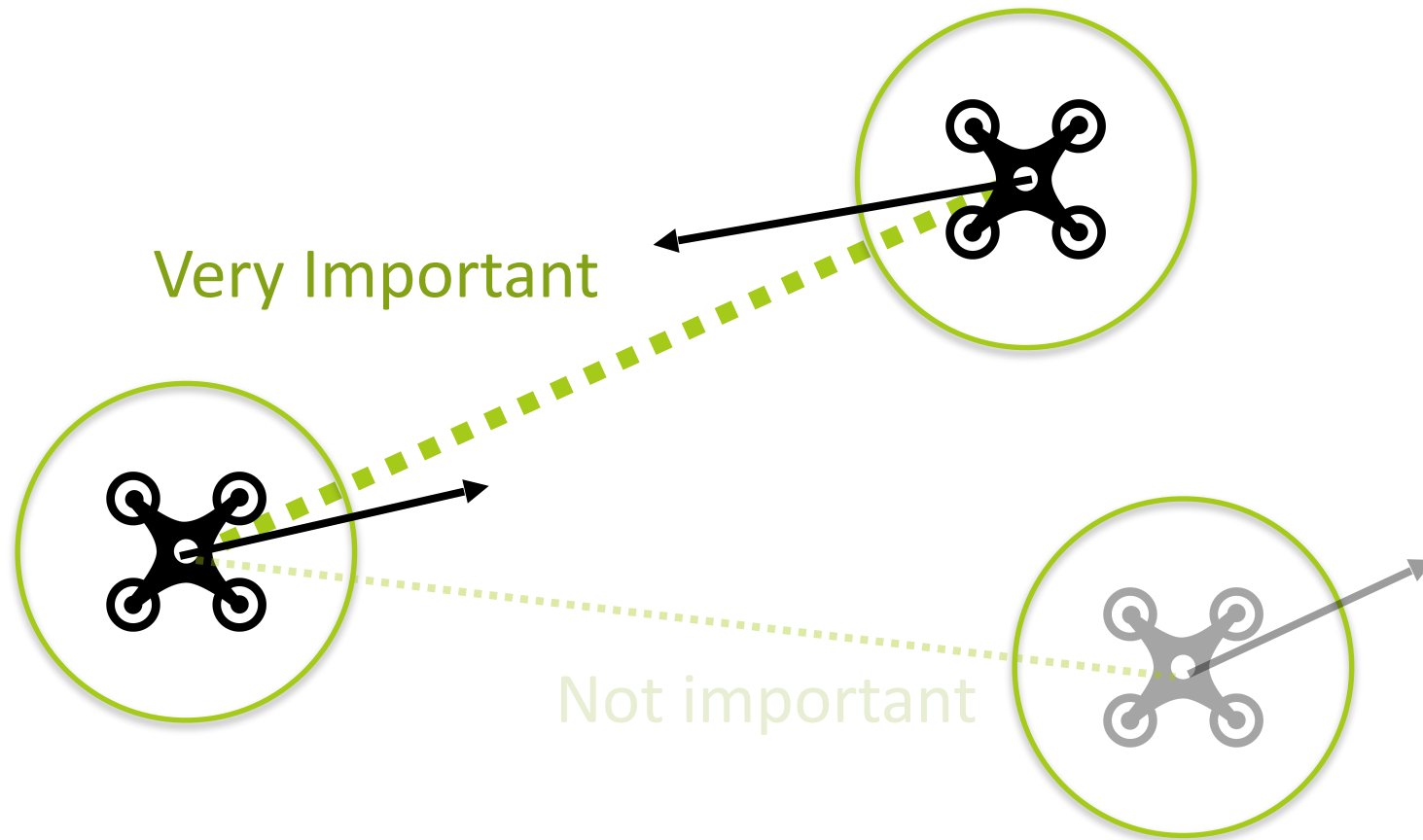
Model Architecture: Attention Mechanism

- Allows focus on the most important parts of the input



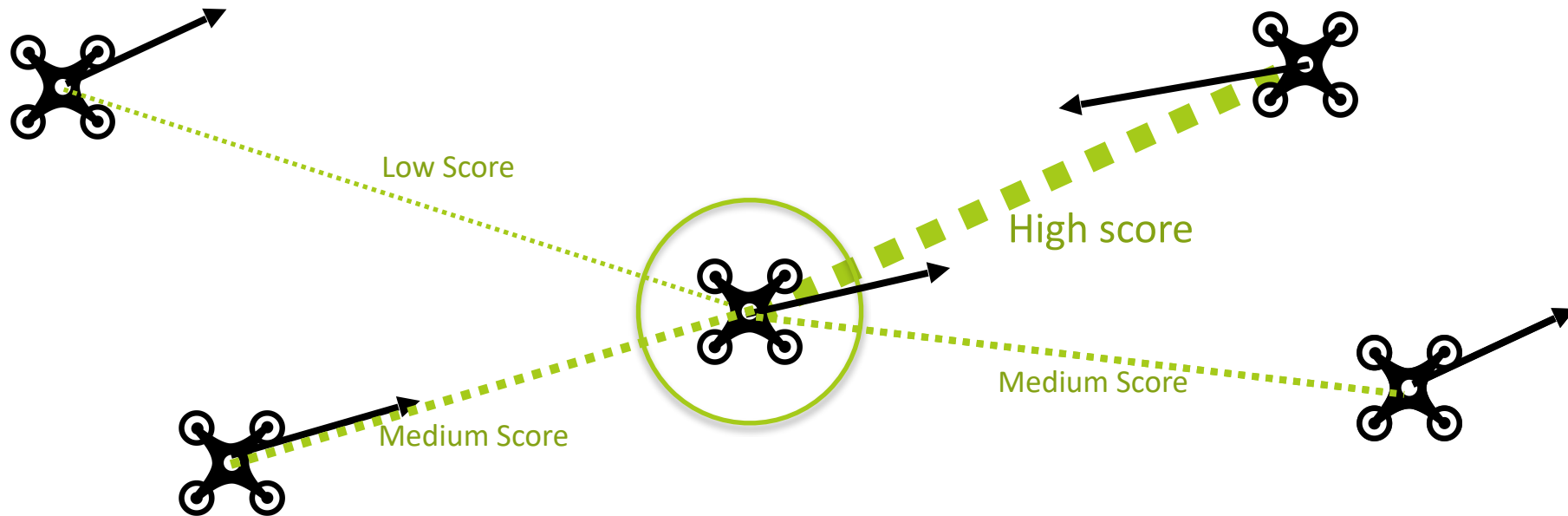
Model Architecture: Attention Mechanism

- Allows focus on the most important parts of the input



Model Architecture: Attention Mechanism

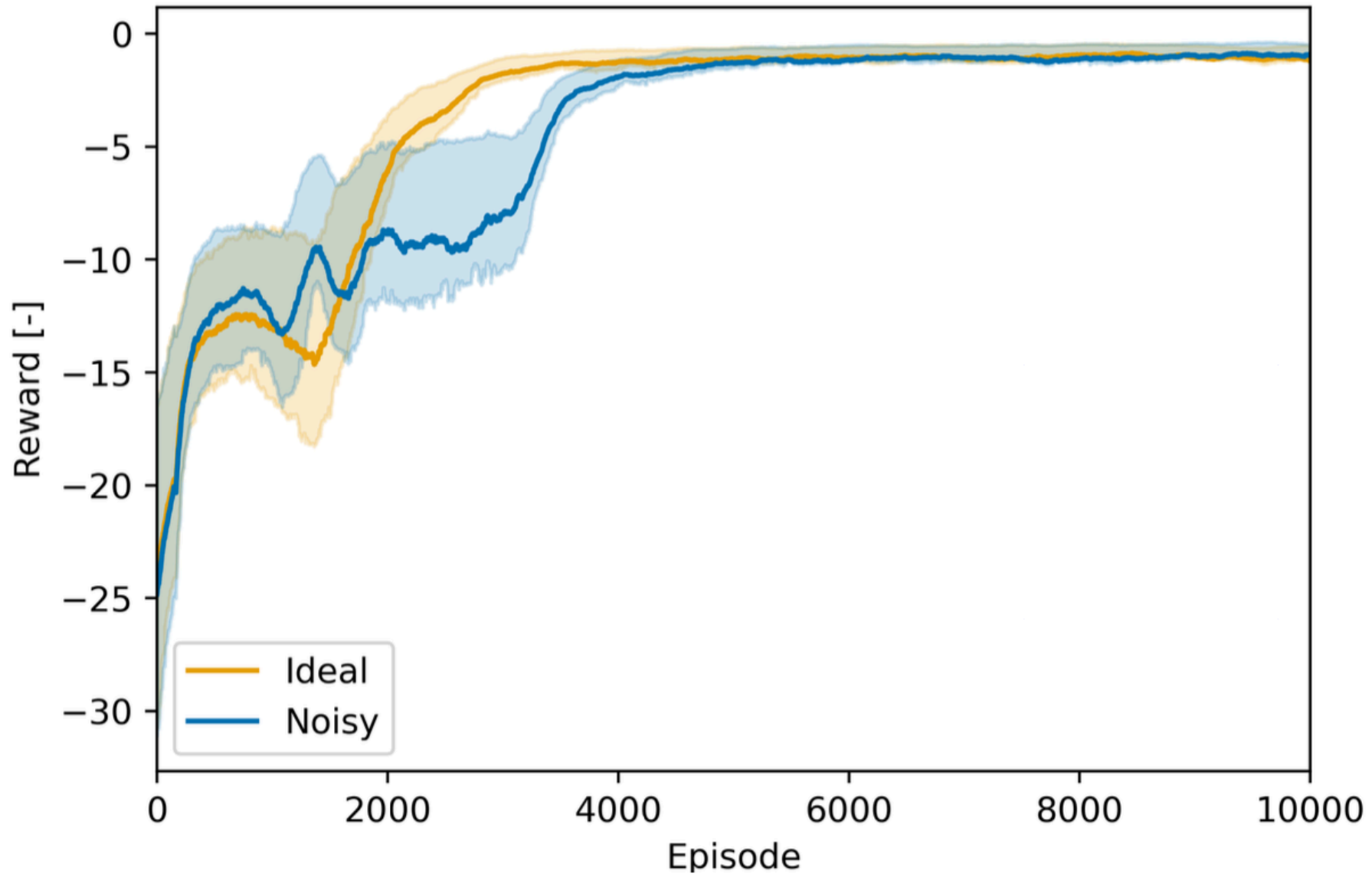
- Allows focus on the most important parts of the input
- **Flexible Input Size**



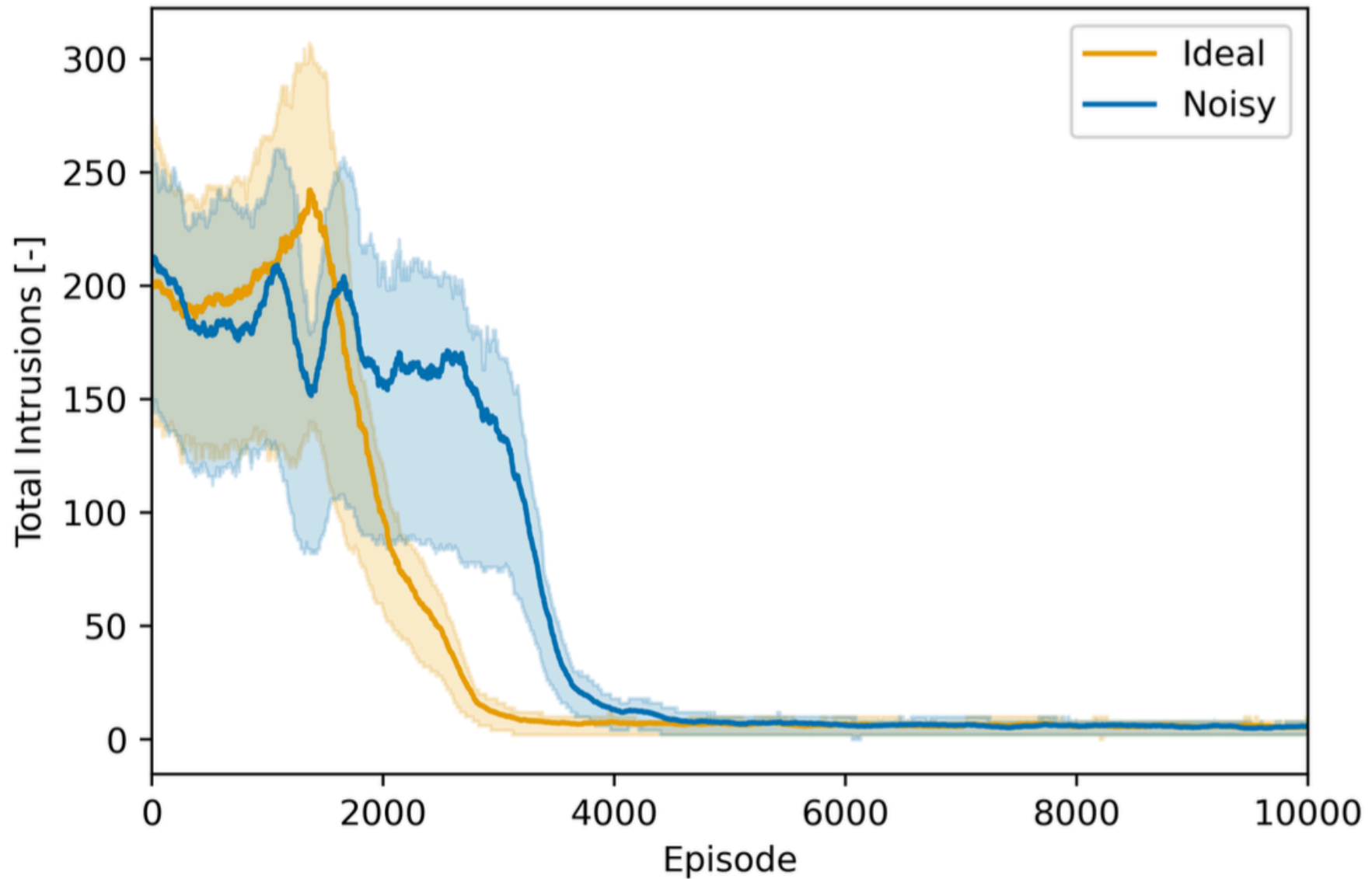
Training Settings

- 10,000 randomised episodes
- With and without position noise
- Noise: Gaussian, zero mean, 3.5 STD

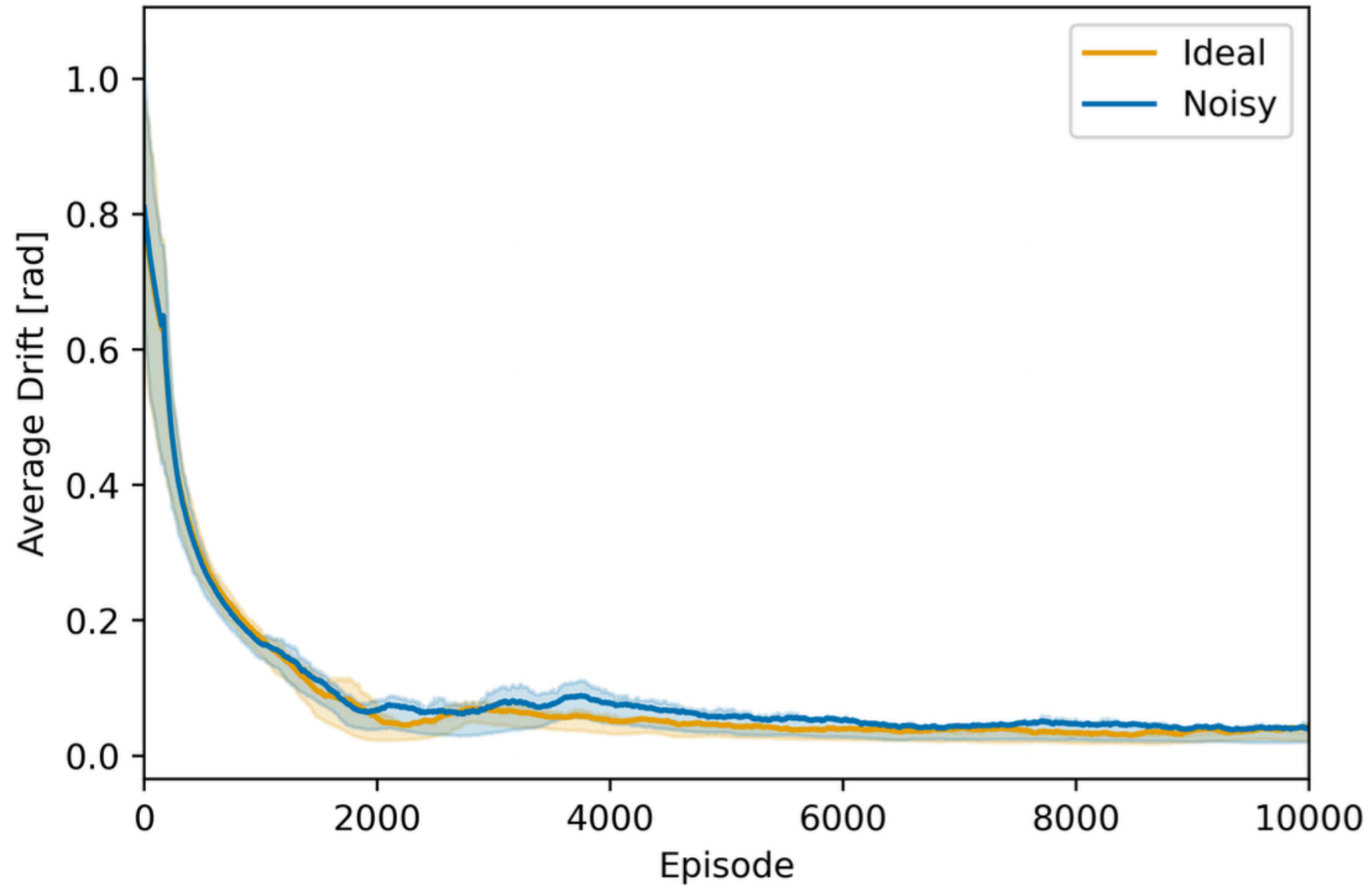
Reward Evolution



Intrusion Evolution

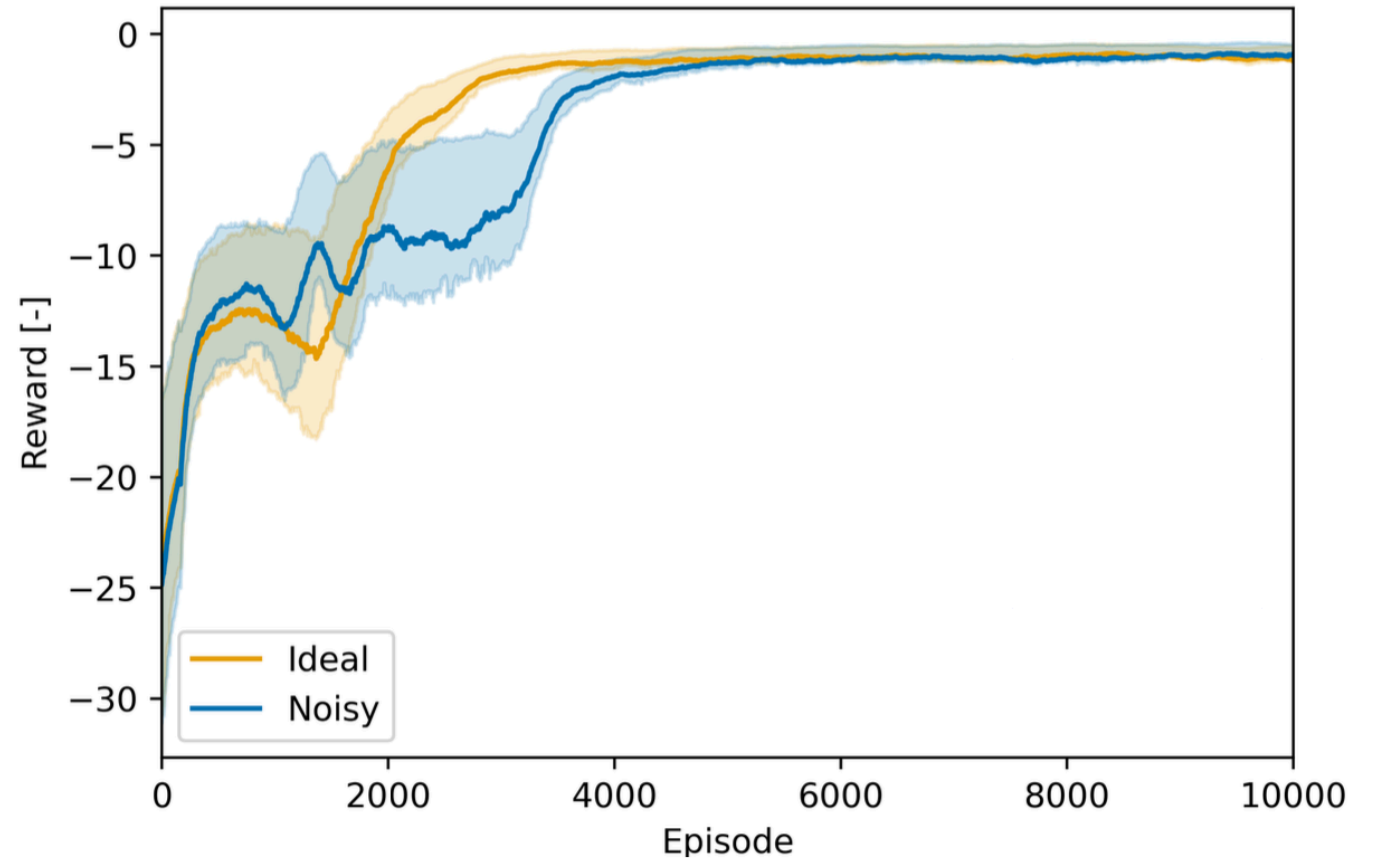


Drift Evolution



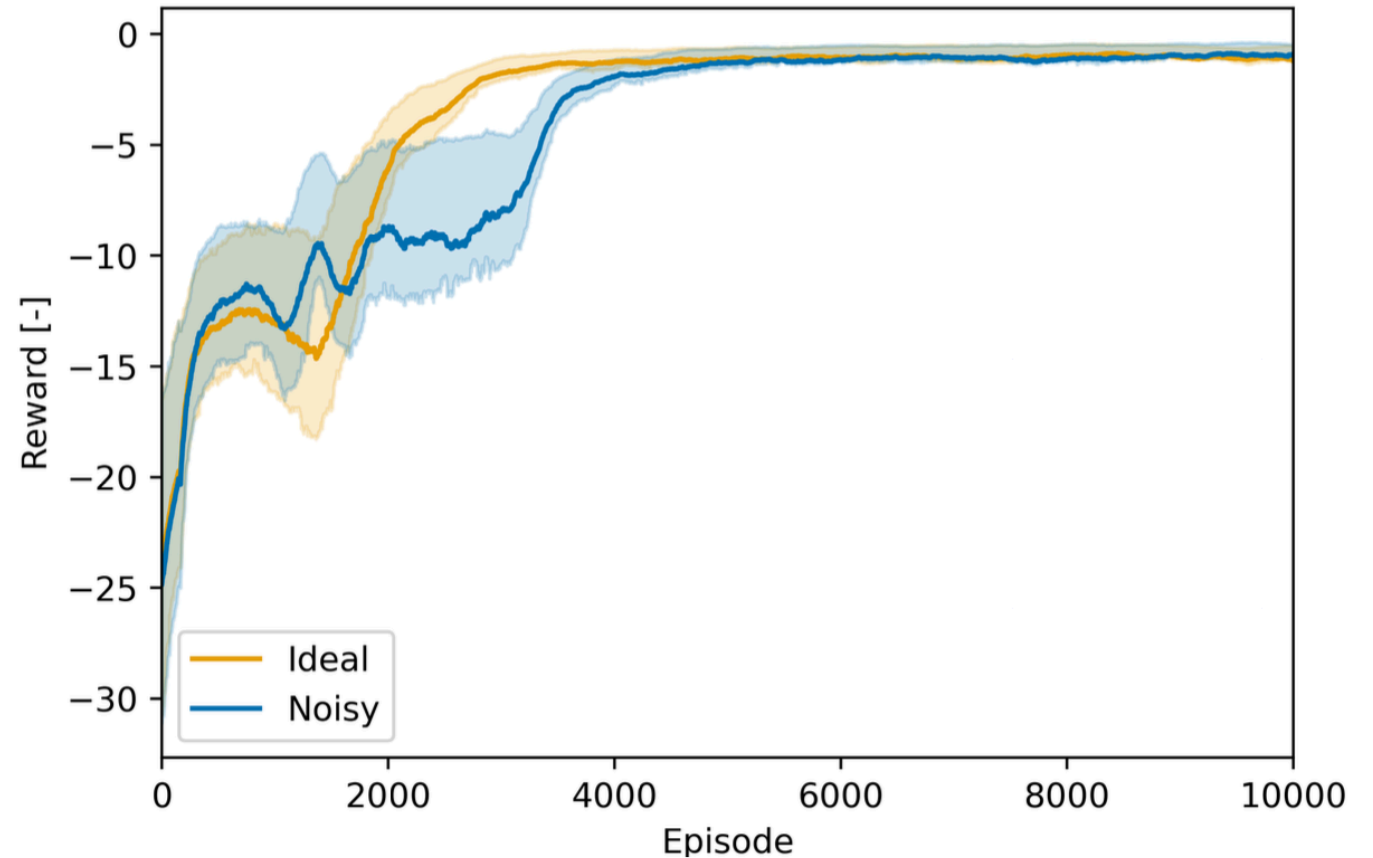
Overall Training Outcomes

- Asymptotic performance is reached!
- Similar performance for both noise and clean trained models
- Intrusions successfully decreased



Overall Training Outcomes

- Asymptotic performance is reached!
- Similar performance for both noise and clean trained models
- Intrusions successfully decreased

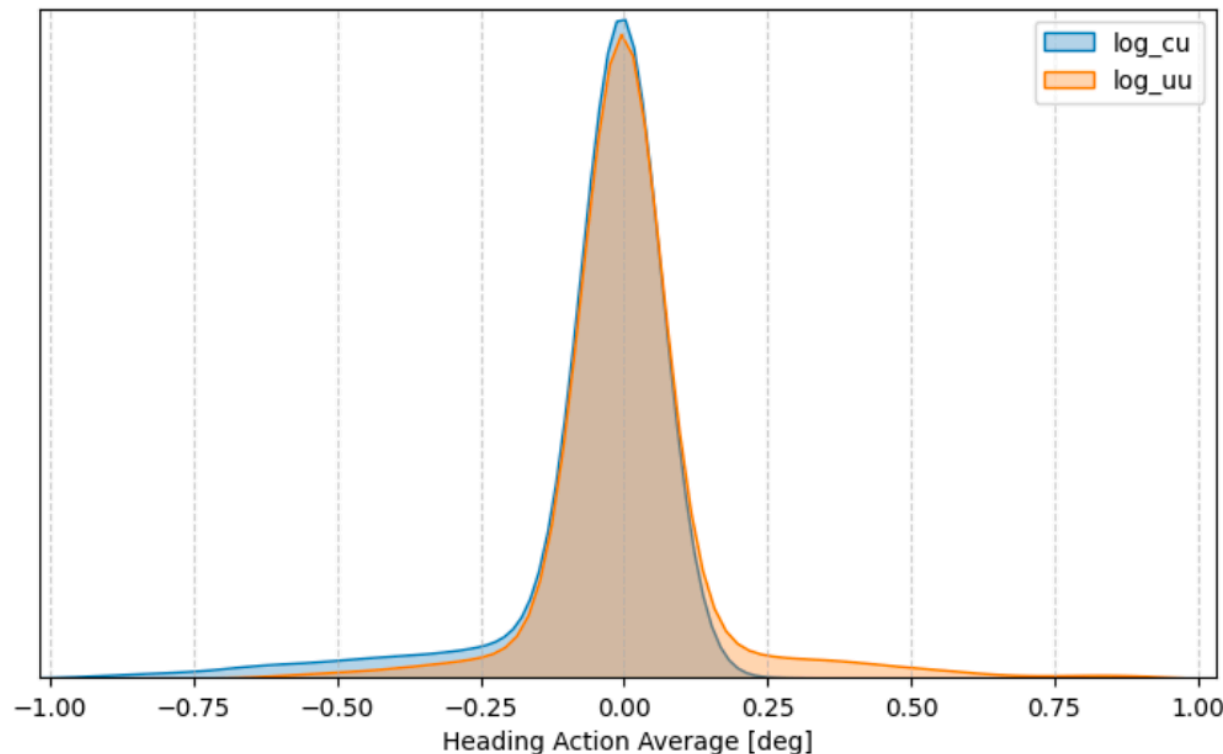


Policy Comparison

- Do some testing runs of trained models (**noisy** and **ideal**)
- **Log all of the actions taken**
- Evaluate based on those through a KDE

Policy: Noise VS Ideal-Trained

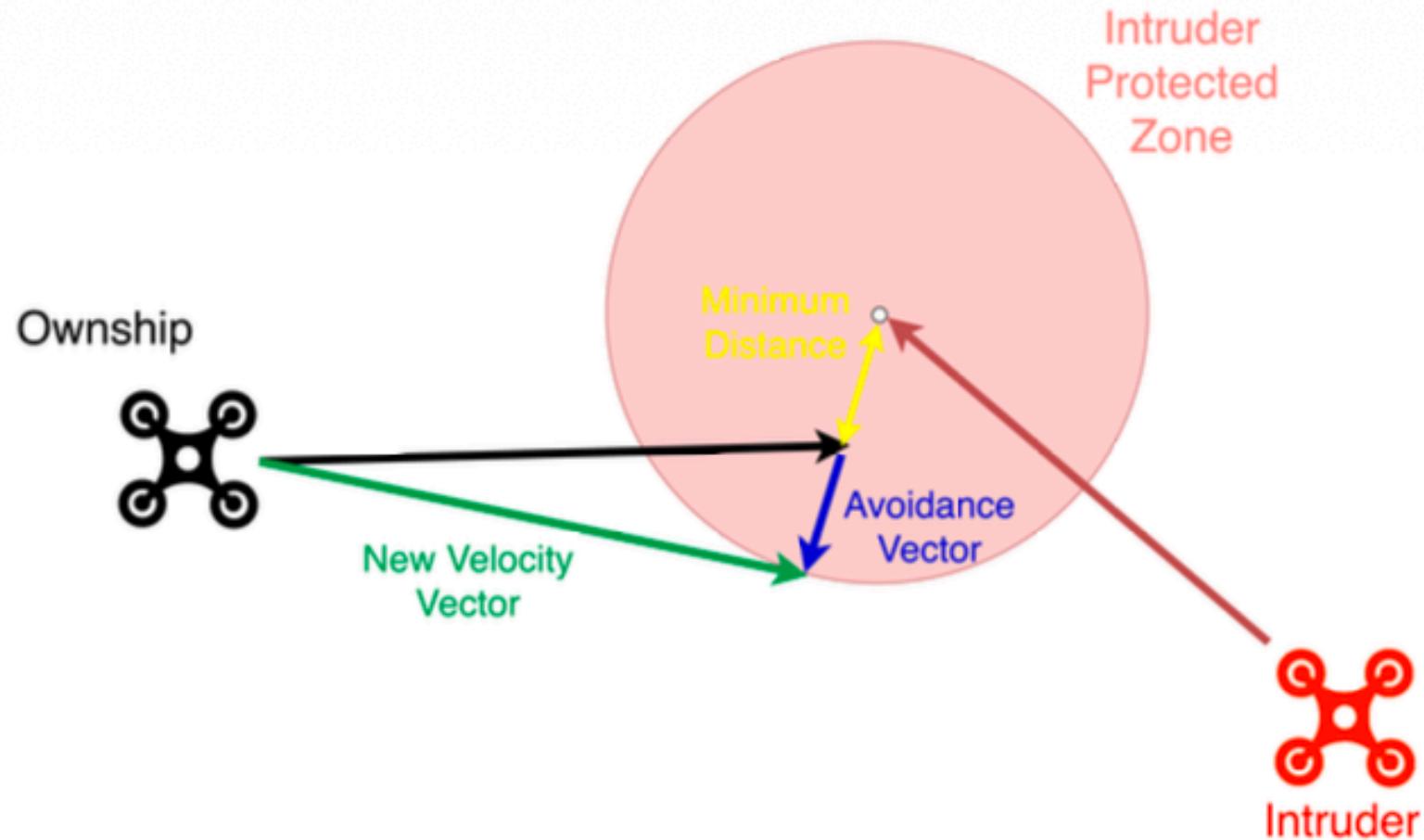
- Both models **reduce speed**
- More spread in speed inputs for **noise-trained**
- **Ideal-trained** has learned right-priority rules



Benchmarking

- Need to compare to existing geometric methods
- Modified Voltage Potential (MVP)

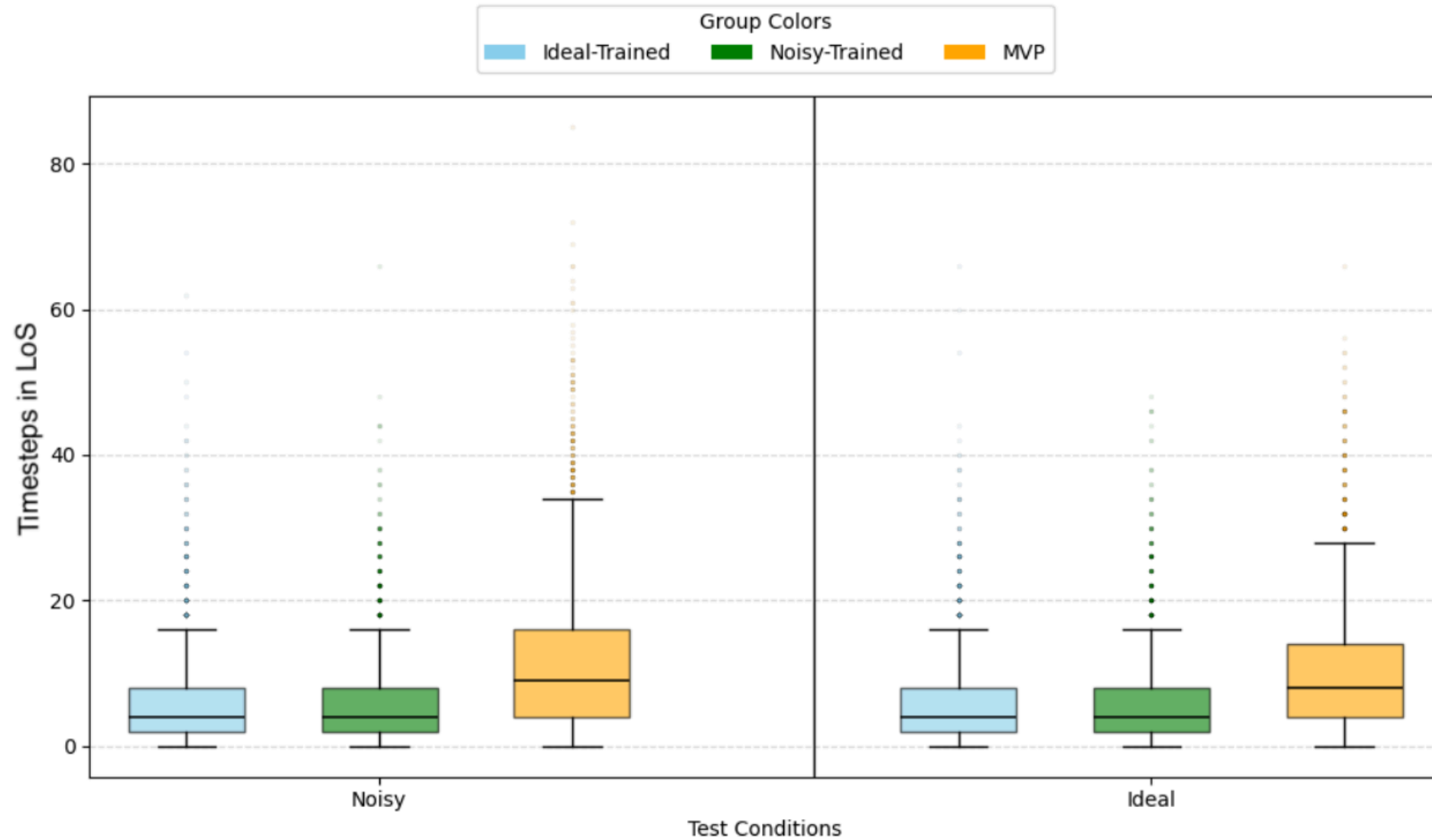
Modified Voltage Potential



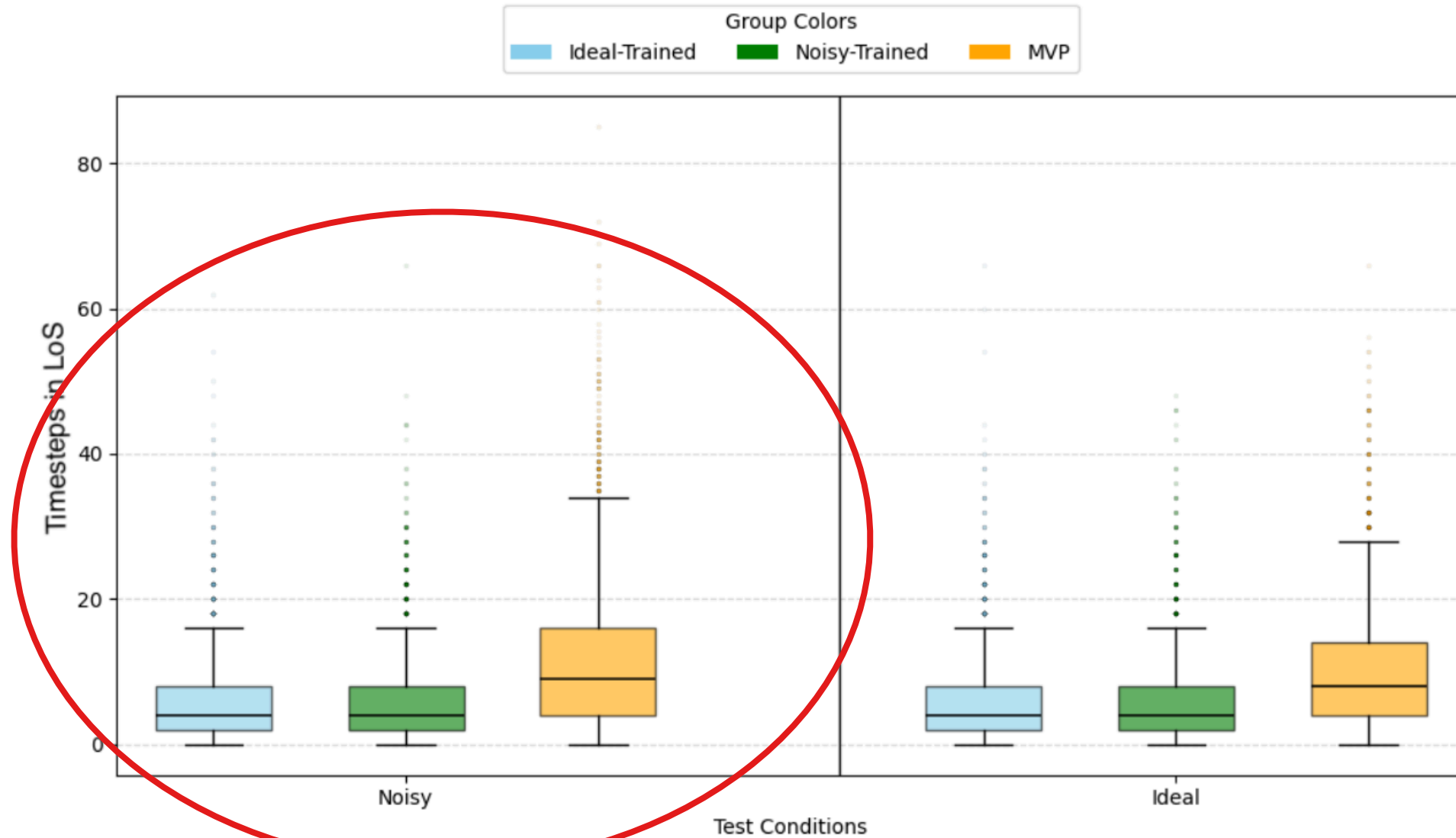
Benchmark: settings

- MVP compared to CR Model (with and without noise)
- 10,000 test episodes
- Logged metrics:
 - Reward
 - Intrusion Timesteps
 - Drift

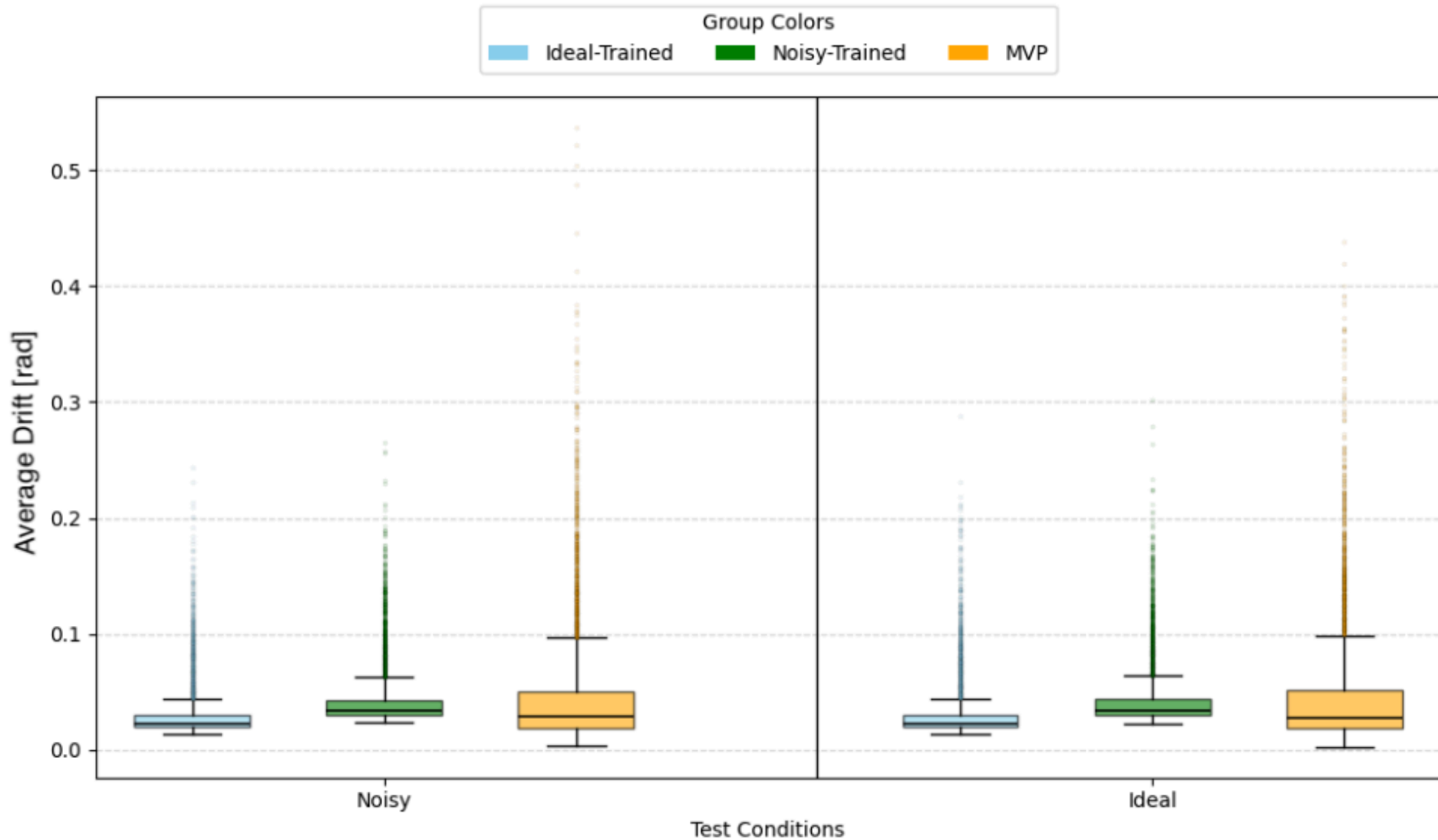
Low-Noise Benchmarks [$\sigma_n = 3.5m$]



Low-Noise Benchmarks [$\sigma_n = 3.5m$]



Low-Noise Benchmarks [$\sigma_n = 3.5m$]



Low Noise Benchmarks: Outcomes

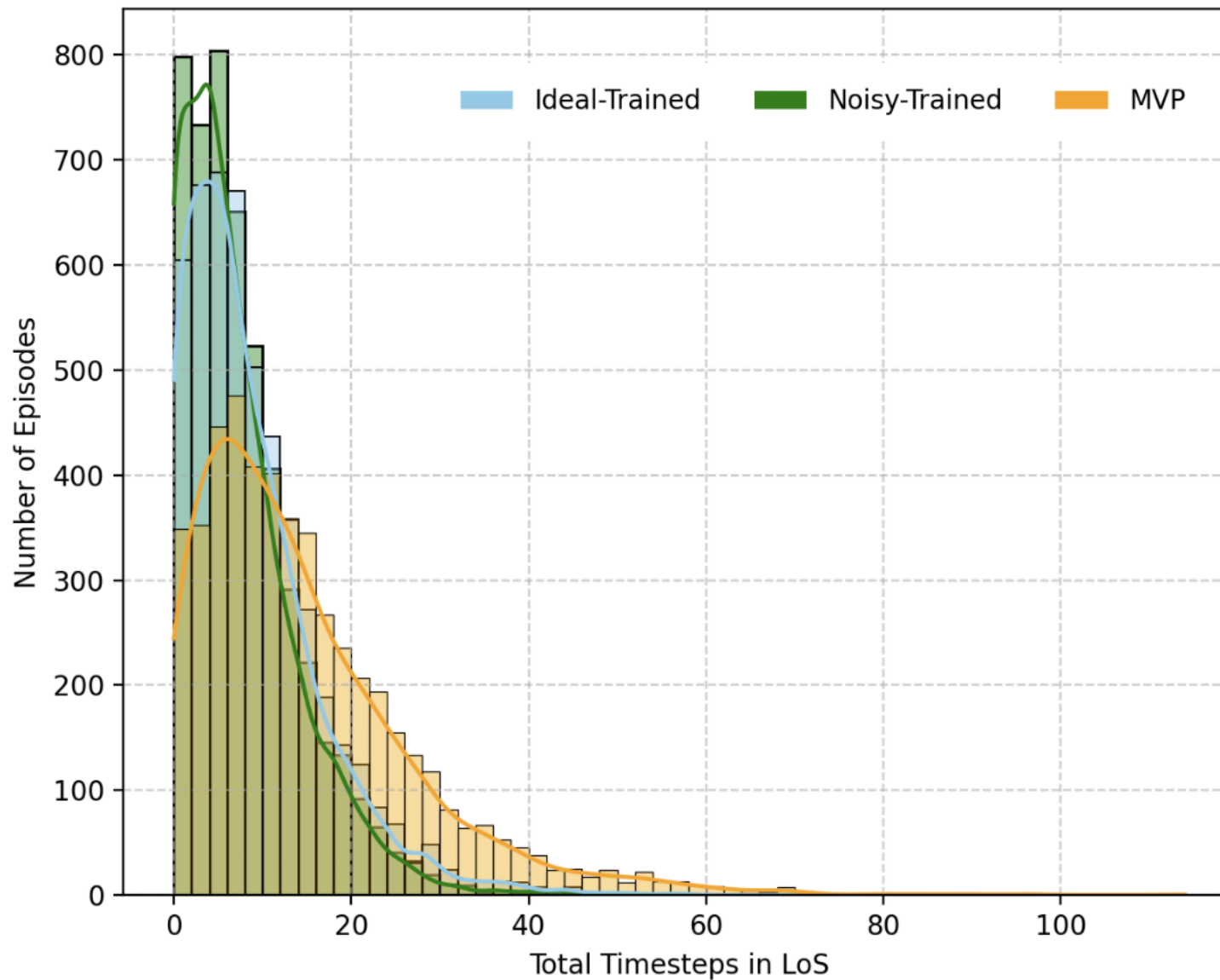
- RL-derived methods are **safer** by a small margin
- Noisy-trained model shows more drift
- Little difference between Ideal and Noisy model in safety

Low Noise Benchmarks: Outcomes

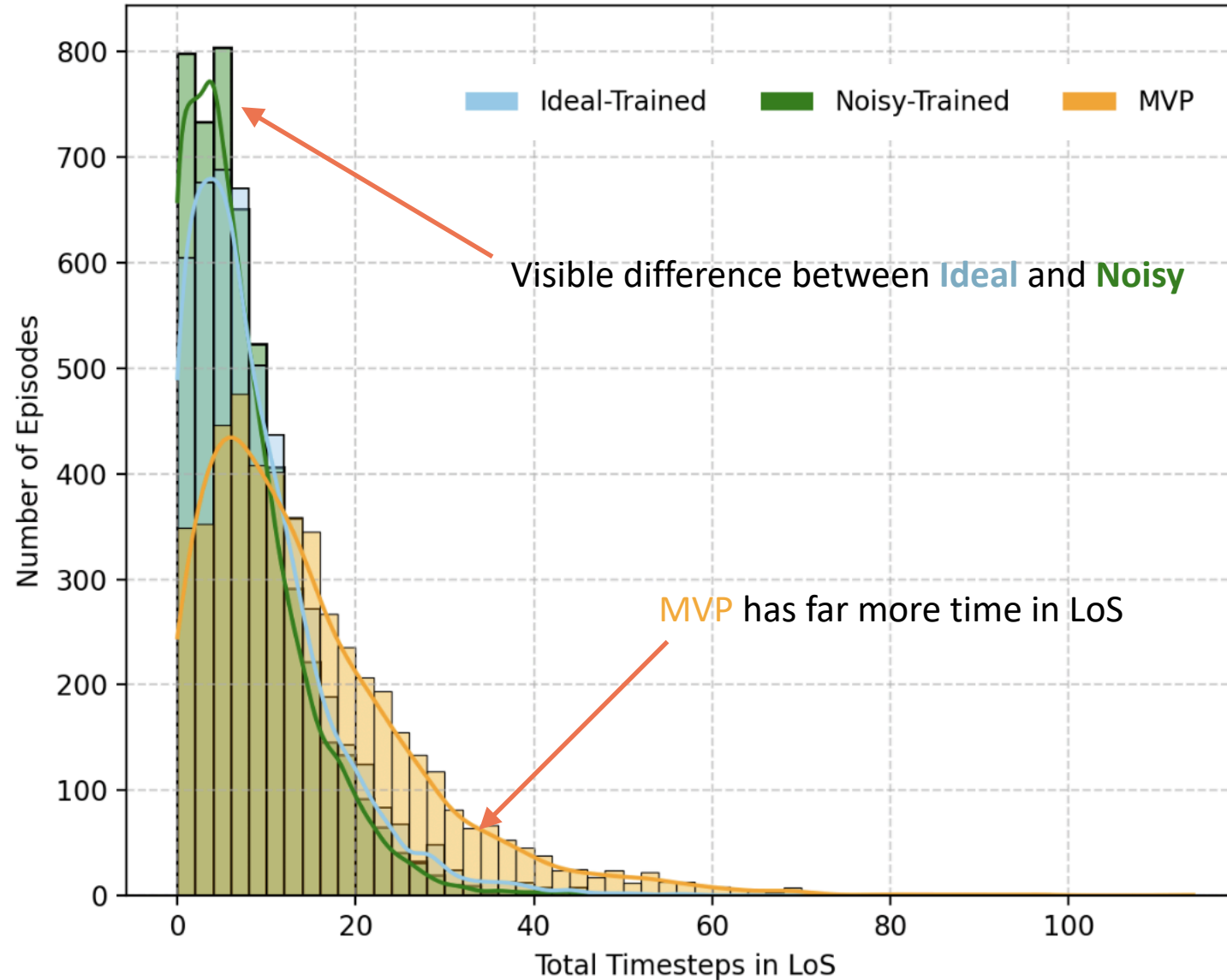
- RL-derived methods are **safer** by a small margin
- Noisy-trained model shows more drift
- Little difference between Ideal and Noisy model in safety

Shall we increase the noise standard deviation?

High-Noise Benchmarks [35m]



High-Noise Benchmarks [35m]



Time in Intrusion

<u>CR Method</u>	<u>Mean + STD</u>
RL Model	6.87 ± 6.41
MVP	14.41 ± 12.30

Time in Intrusion

<u>CR Method</u>	<u>Mean + STD</u>
RL Model	6.87 $\pm$ 6.41
MVP	14.41 $\pm$ 12.30

MVP 2X higher!

Intrusions: Some Numbers

- We have only looked at **time in intrusion** so far

Intrusions: Some Numbers

- We have only looked at **time in intrusion** so far
- How many unique intrusions happened?

Intrusions: Some Numbers

- We have only looked at **time in intrusion** so far
- How many unique intrusions happened?
- How bad were they?

Intrusions: Some Numbers

- We have only looked at **time in intrusion** so far
- How many unique intrusions happened?
- How bad were they?

$$I_{\text{sev}} = \frac{I}{R_{\text{PZ}}} = \frac{R_{\text{PZ}} - d_{\text{CPA}}}{R_{\text{PZ}}}$$

Intrusions: Some Numbers

- We have only looked at **time in intrusion** so far
- How many unique intrusions happened?
- How bad were they?

$$I_{\text{sev}} = \frac{I}{R_{\text{PZ}}} = \frac{R_{\text{PZ}} - d_{\text{CPA}}}{R_{\text{PZ}}}$$

In other words: how close did we get?

How bad were the unique intrusions?

Train	Test	Intrusions	Severity
Uncertainty	Uncertainty	2.43 ± 1.50	0.256 ± 0.168
Clean	Uncertainty	2.75 ± 1.67	0.232 ± 0.161
MVP	Uncertainty	3.54 ± 2.17	0.288 ± 0.162

How bad were the unique intrusions?

Train	Test	Intrusions	Severity
Uncertainty	Uncertainty	2.43 ± 1.50	0.256 ± 0.168
Clean	Uncertainty	2.75 ± 1.67	0.232 ± 0.161
MVP	Uncertainty	3.54 ± 2.17	0.288 ± 0.162

- Clean-trained far better in terms of unique intrusions
- Second-best in severity!

Research Questions

- Can a RL agent outperform the current baseline for conflict resolution with position error?
- Should this RL agent be trained on said position noise?

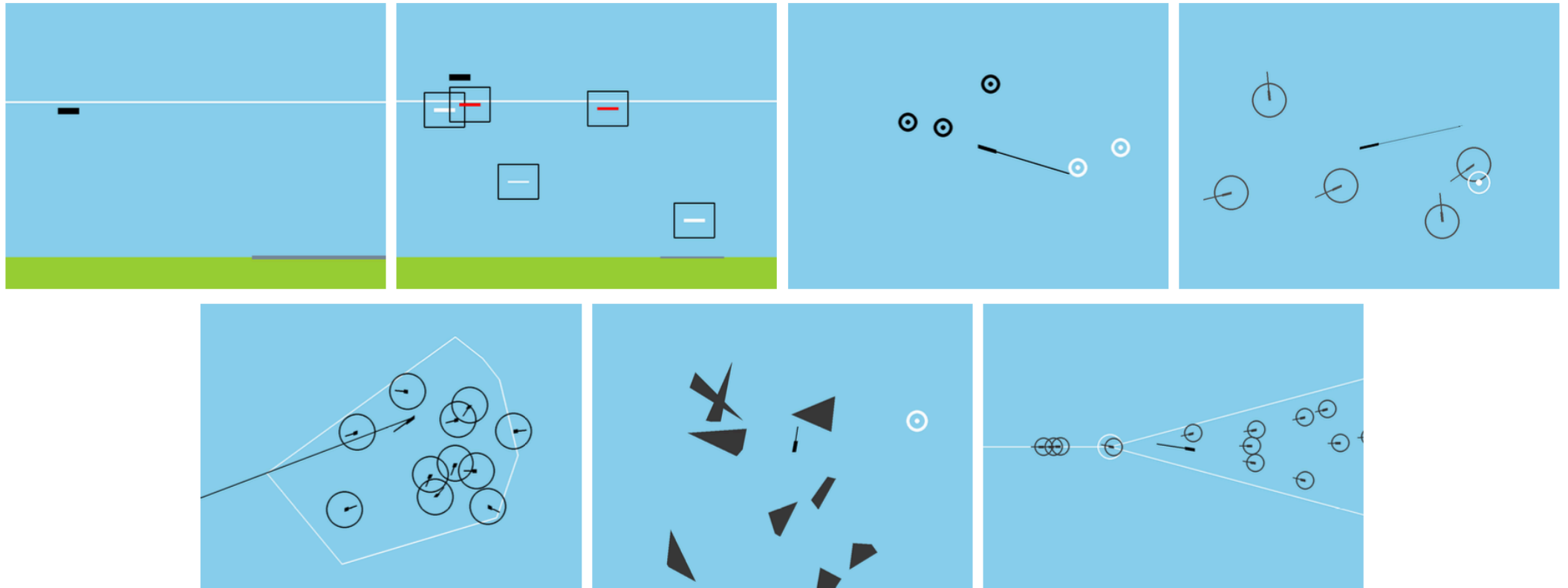
Conclusions

- Yes!
- The margin is small, but the noise-trained model is slightly better

Recommendations

- Ablation Study
- Other scenarios and geometry
- MVP tuning
- Expanding uncertainty model
- Other RL algorithms
- Don't take my word for it

BlueSky-Gym

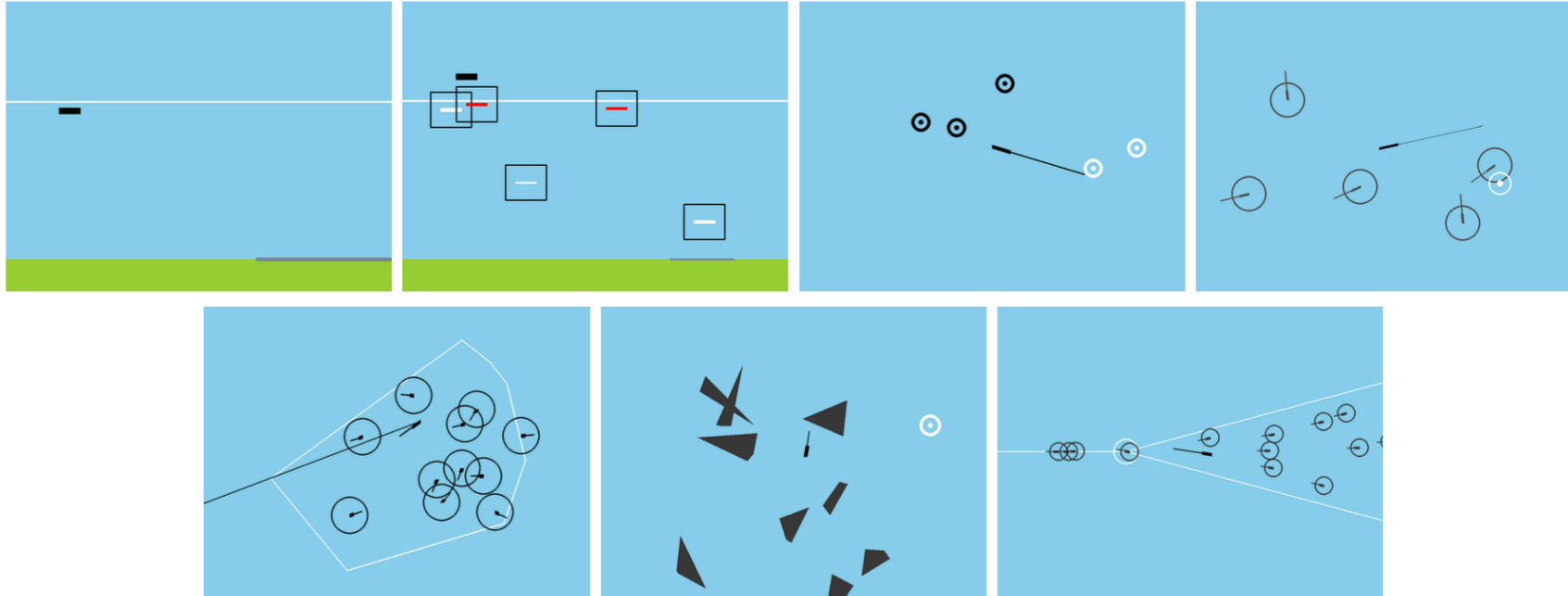


✉ sasha.vlaskin@nlr.nl

🔗 <https://github.com/TUdelft-CNS-ATM/bluesky-gym>

BlueSky-Gym: Reinforcement Learning Environments for Air Traffic Applications

D.J. Groot¹ & G. Leto¹ & A. Vlaskin^{1,2} & A. Moec¹. and J. Ellerbroek¹



```
~ % pip install bluesky-gym
```

✉ sasha.vlaskin@nlr.nl

🔗 <https://github.com/TUdelft-CNS-ATM/bluesky-gym>

Thank you for your attention!!

Any questions?

