Recurrent Reinforcement Learning for Unguided Counter-UAS Engagement

Martin Krieg · Helmut Hlavacs

Theresan Military Academy, Wiener Neustadt, Austria
University of Vienna, Austria

MESAS 2026 · Modelling and Simulation for Autonomous Systems

THE PROBLEM

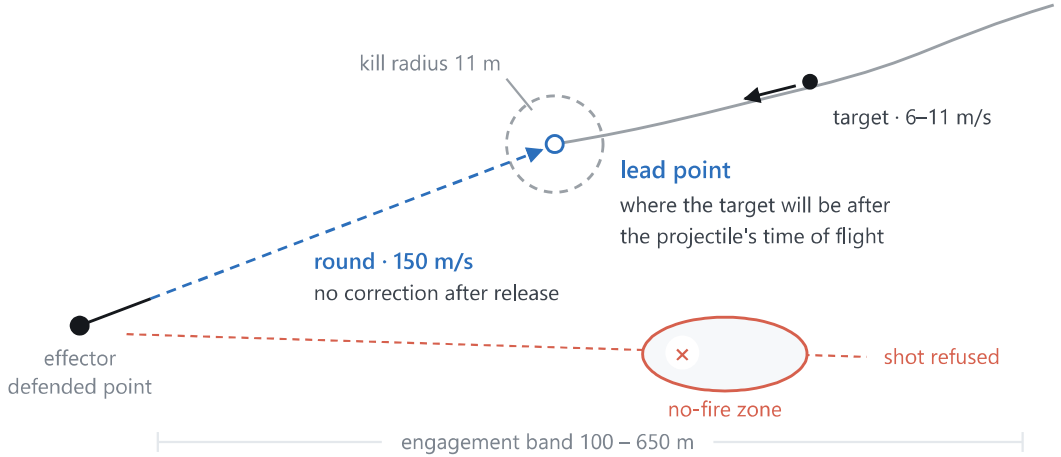
An unguided round must be aimed where the target *will* be

Small drones are a cheap, widespread threat. Guided interceptors cost far more than the target they destroy.

A gun is the cheap answer, but its round cannot steer after the shot.

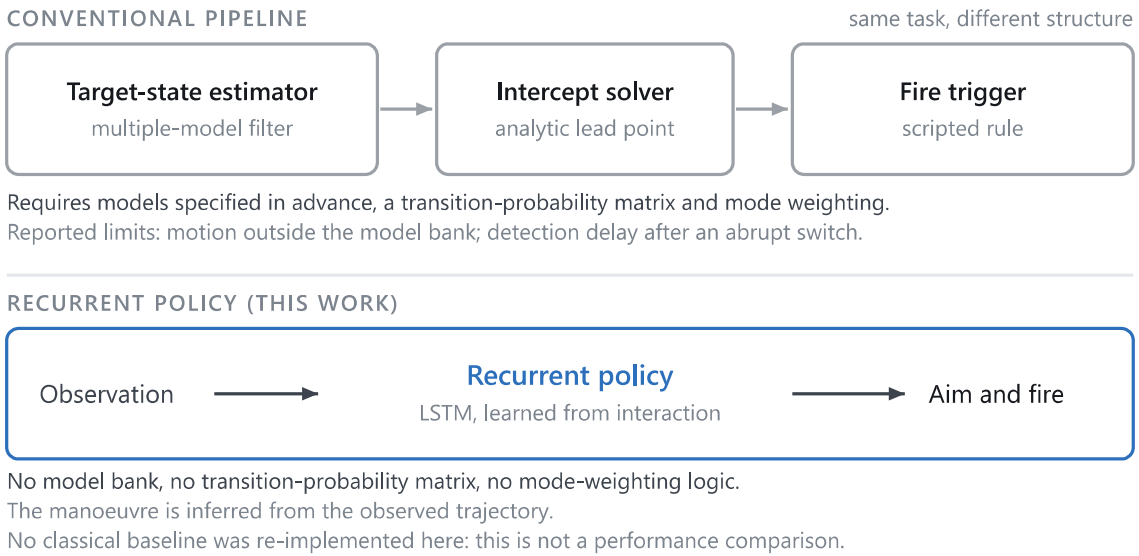
So the gun must aim at the **lead point**: where the drone will be once the round arrives.

The drone's future path is unknown, and it can change at any moment.



STATE OF THE ART

The conventional answer, and what it presumes



The classical stack needs its motion models **written down in advance**, plus a transition matrix and mode-weighting logic.

The literature reports where that hurts: a manoeuvre outside the model bank, and a lag after an abrupt change while the filter catches up.

RESEARCH QUESTION

Can a single learning system do the same job without being told the motion models in advance?

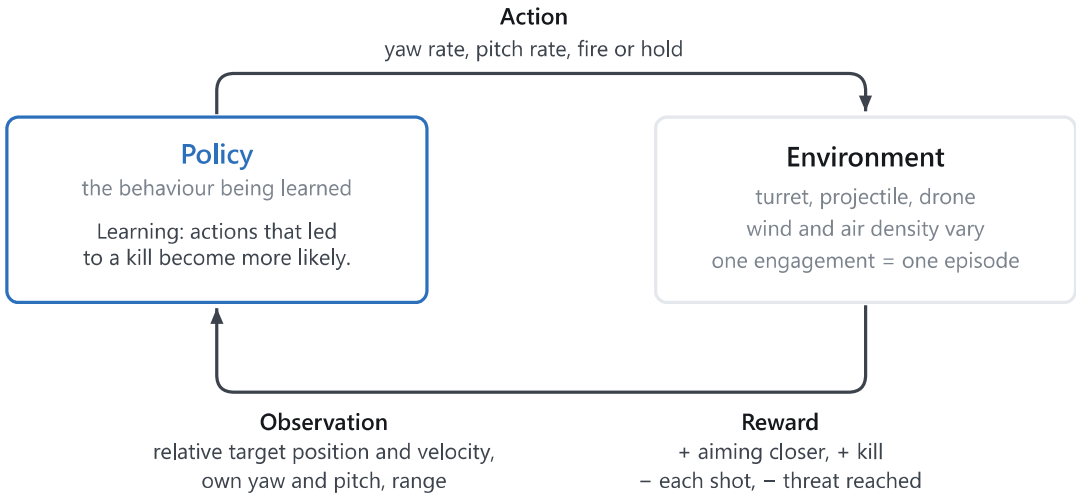
Can it learn to follow a drone, lead it, and decide when to fire?

Does it still work when the drone flies a manoeuvre we kept out of training?

Does it need to measure the wind? And can it learn a safety rule from reward alone?

BACKGROUND 1 / 3

A short primer: reinforcement learning

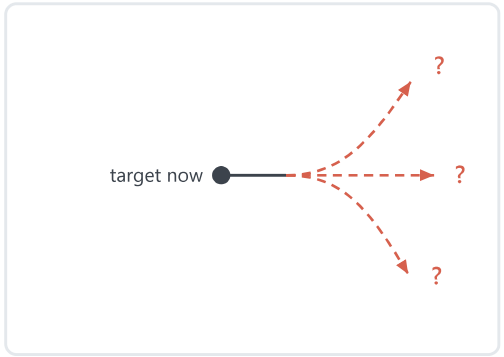


HOW IT WORKS

The policy acts on the environment, the environment changes in response, and the policy receives a new observation together with a reward. Everything it learns comes from repeating that loop.

Why one look at the drone is not enough

ONE OBSERVATION



Position, speed and heading known.
The manoeuvre, and so the lead point, is not.

OBSERVATIONS OVER TIME

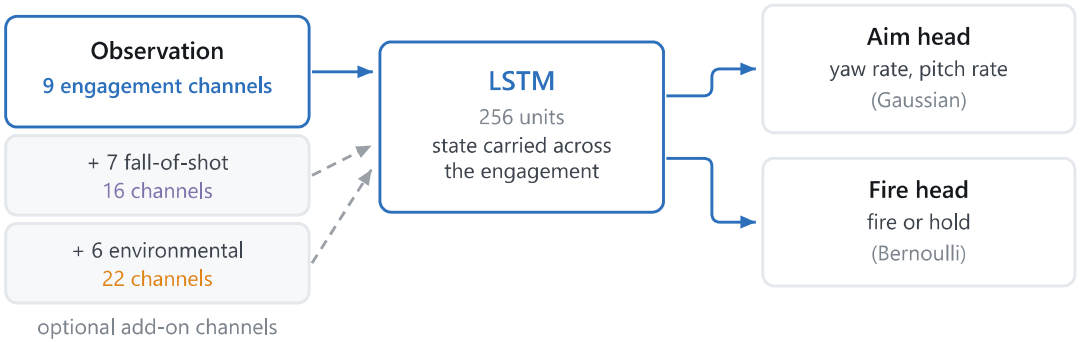


The trajectory implies the manoeuvre.
A lead point can be chosen.

WHY THIS MATTERS

One snapshot gives position, speed and heading, but it does not give the *manoeuvre*. Only a sequence of observations shows which way the drone is turning, and that is what fixes the aim point.

One network, two decisions



One policy, trained end to end with PPO.
The headline result uses the 9-channel path; the add-ons are optional variants.

Since one observation cannot reveal the manoeuvre, the policy has to work on the **trajectory** rather than on single snapshots.

That is what the **LSTM** layer provides: it carries a running summary of the engagement so far from one step to the next.

Aiming and firing are learned **together**, in one network, not stacked as separate modules.

METHOD

The simulated engagement

Built in the Godot engine. One engagement is one episode.

Wind, air density and muzzle velocity vary from engagement to engagement and push the round off course.

The policy is trained in stages: aim first, then fire, then a moving target, then harder conditions.



A pass-by: the drone crosses the scene while the turret tracks it.

METHOD

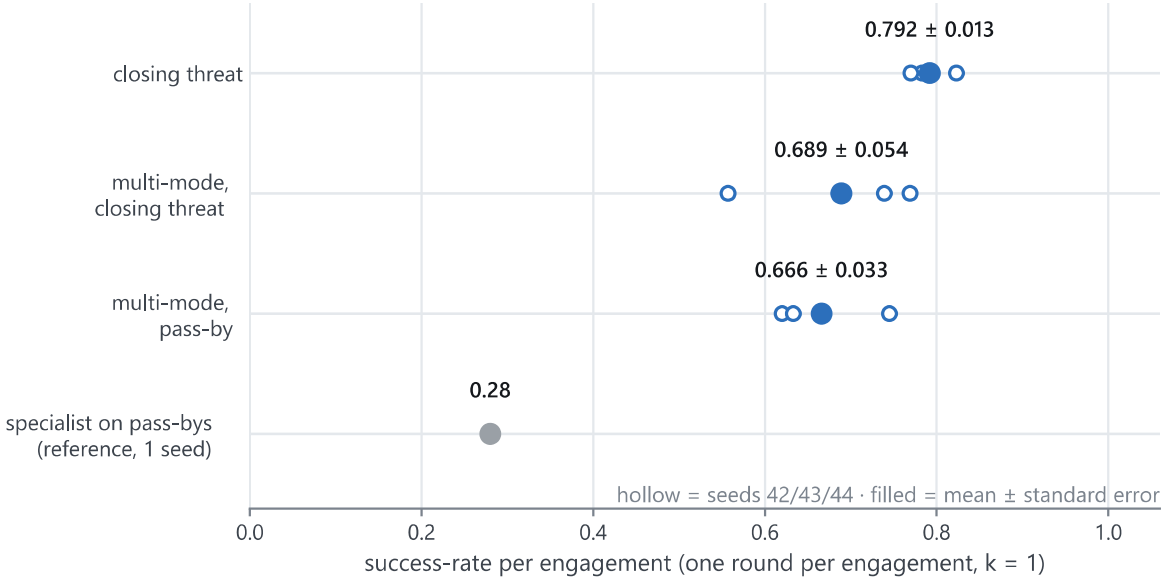
What the target does

SEEN FROM ABOVE



RESULT 1 / 5

It learns to follow, lead and fire



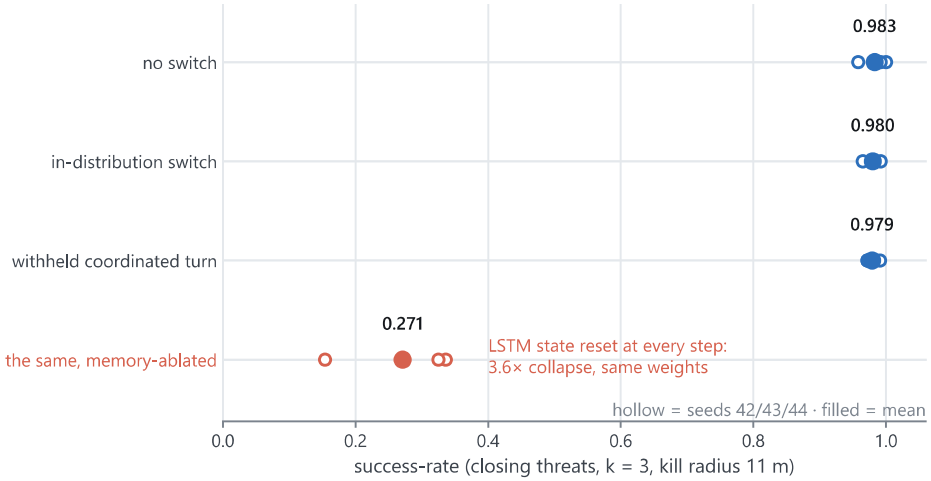
Against a drone flying straight at the gun, with **one single round** per engagement, it succeeds in **79 %** of engagements.

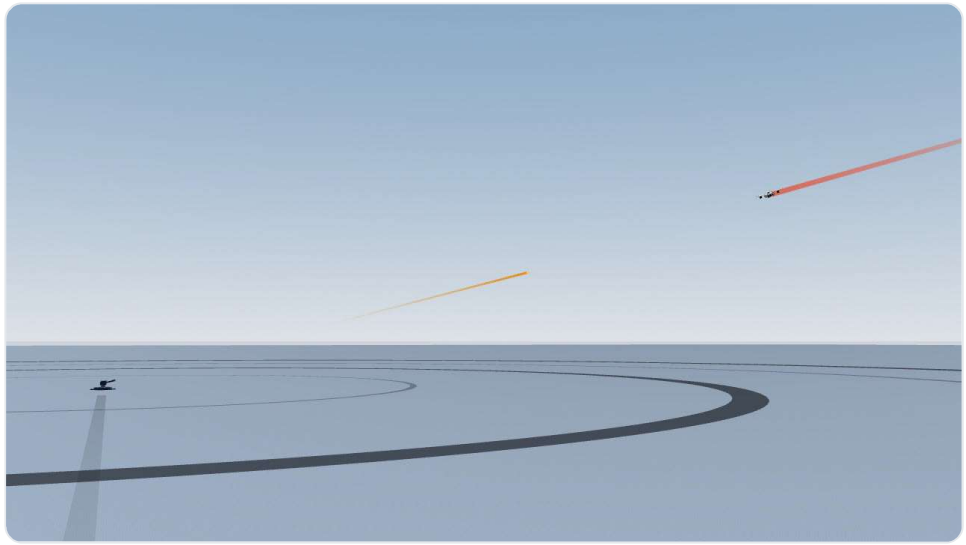
One policy handles **both** flight patterns at once (69 % and 67 %).

A policy trained on only one pattern collapses to **28 %** on the other.

RESULT 2 / 5 · THE CENTRAL TEST

A manoeuvre it never saw in training





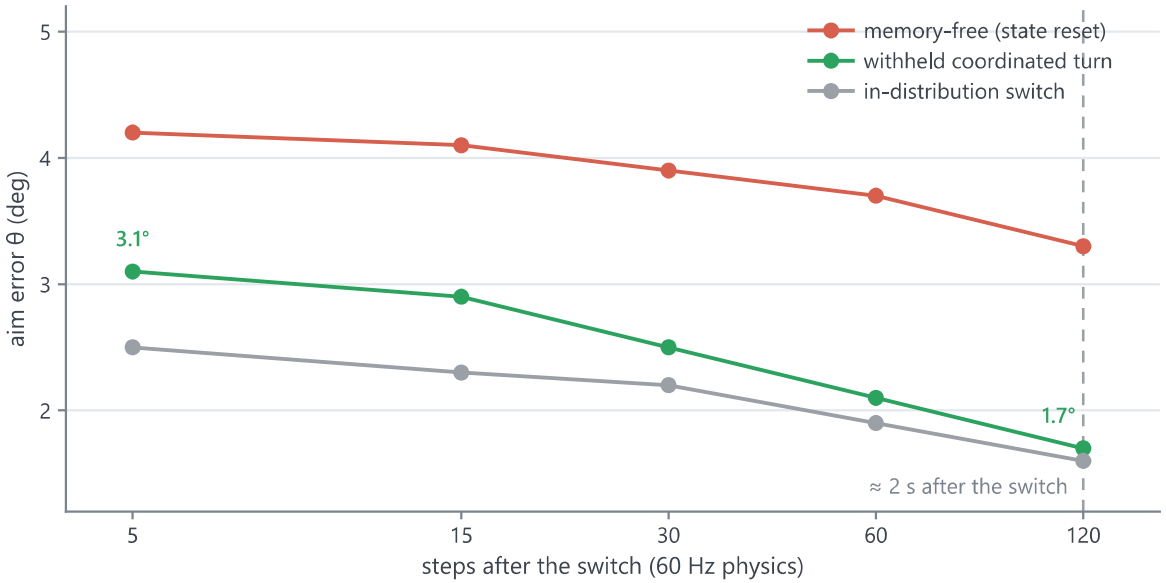
Mid-flight the drone turns into a manoeuvre withheld from training.

THE RESULT

The unseen manoeuvre costs almost nothing, 97.9 % against 98.3 % with no change at all. The policy achieves that without a model bank, without a transition matrix and without any mode-weighting logic.

RESULT 3 / 5

The memory is what makes it work



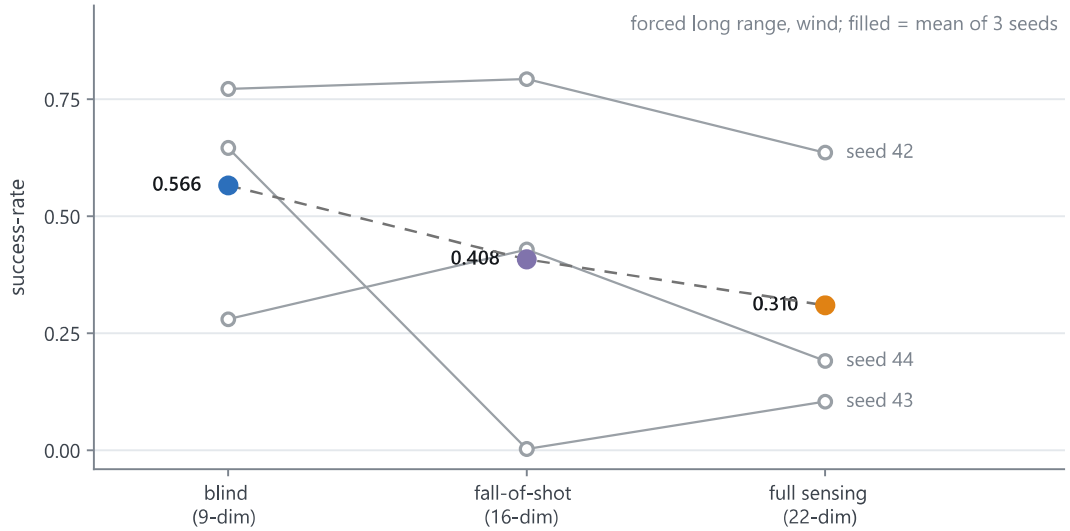
Take the **same trained network** and wipe its memory at every step: success falls from **98 %** to **27 %**.

It still *fires* just as often. It simply can no longer **aim**.

After the manoeuvre change, its aim recovers within about **two seconds**.

RESULT 4 / 5 · THE SURPRISE

Measuring the wind did not help



We gave the policy wind, temperature, humidity and muzzle speed as sensor inputs. It learned to **ignore them**.

Instead it sidesteps the problem: it **waits until the drone is close**, so the round is barely in the air, and it fires a burst.

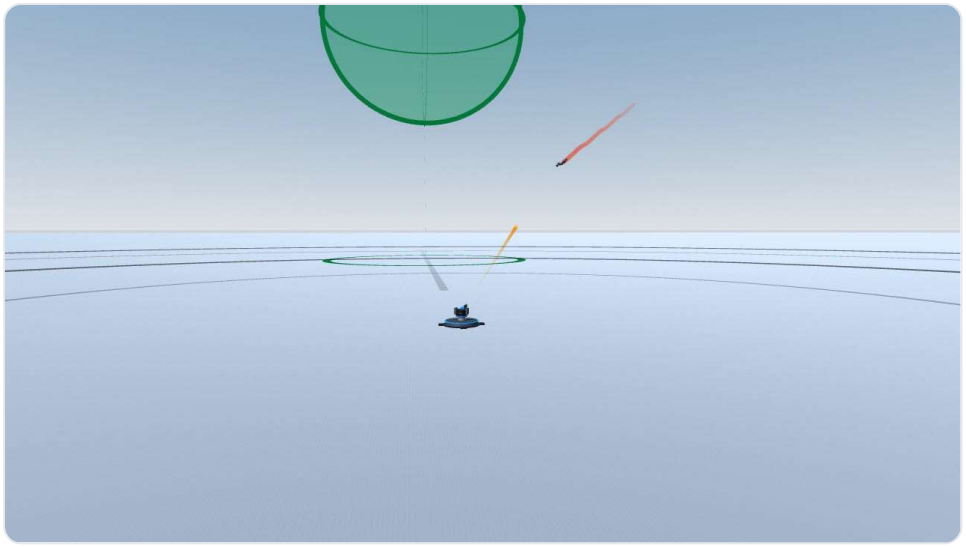
Operationally that is the wrong instinct. You want to engage **early and far**, not let the threat come to you.

THE LESSON

The policy handles the disturbance by changing *when* it acts, not by measuring the physics. Take that option away and what pays is still not the weather sensors, but watching where its own rounds went.

RESULT 5 / 5

Safety is where explicit structure pays



The protected volume is green. The burst runs clear of it.

Teaching the rule through **penalties fails**: hide the zone and the policy stops firing; show it and it only partly trades performance for safety.

A small **analytic rule** takes over the trigger: it scores the shot's chance to hit and its distance from the zone, and fires at the best moment.

Result: **zero** violations in 900 engagements, and the success rate *rose* from 62 % to 85 %.

DISCUSSION

What is learned, and what is built

Learned

Following, leading and the firing decision, all in one network.

Coping with an unseen manoeuvre, with no model of it anywhere.

Handling wind by firing close and firing a burst.

Built by hand

The safety rule, because it can be stated exactly and must hold **every time**.

It governs only the trigger. The learned aiming is untouched.

This is not a claim that learning beats classical fire control. It is a claim about which of its prerequisites turn out to be dispensable.

CONCLUSION

Three things to take away

A recurrent policy learns to lead and fire an unguided gun, and **holds up against a manoeuvre withheld from training**, without any motion model being specified.

Its **memory** is what makes that work: remove it and the same network can no longer aim.

Robustness to wind comes from **when it fires**, not from measuring the wind. But a hard safety constraint still needs an explicit rule.

Next steps: more withheld manoeuvres and harsher conditions, a clearer boundary for when extra sensing begins to pay, and calibration against real data.

Thank you for your attention

Questions are welcome.

Martin Krieg · m.krieg@hbv.gv.at
Theresan Military Academy · University of Vienna