Single-Agent Policies for the Multi-Agent Persistent Surveillance Problem via Artificial Heterogeneity

Tom Kent[1], Arthur Richards[1] & Angus Johnson[2]

EUMAS 2020

14-09-20

[1]University of Bristol, Bristol, UK - thomas.kent@bristol.ac.uk,
[2]University of Bristol, Bristol, UK - arthur.richards@bristol.ac.uk
[3]Thales UK, Reading, UK - angus.johnson@uk.thalesgroup.com

T-B PHASE

T-B Partnership in Hybrid Autonomous Systems Engineering

- **Five-year project** (2017-22) fundamental autonomous system design problems
- **Hybrid Autonomous Systems Engineering** 'R3 Challenge':
  - **Robustness, Resilience, and Regulation**.
- Innovate **new design principles and processes**
- Build **new tools** for analysis and design
- Engaging with **real Thales use cases**:
  - Hybrid Low-Level Flight
  - Hybrid Rail Systems
  - Hybrid Search & Rescue.
- **Engaging stakeholders** within Thales
- Finding a balance between academic and industrial outputs

University of BRISTOL

Academic PIs
Seth Bullock
Eddie Wilson
Jonathan Lawry
Arthur Richards

Post-Docs
Tom Kent
Michael Crosscombe
Debora Zanatto

PhDs
Elliot Hogg
Will Bonnell
Chris Bennett
Charles Clarke

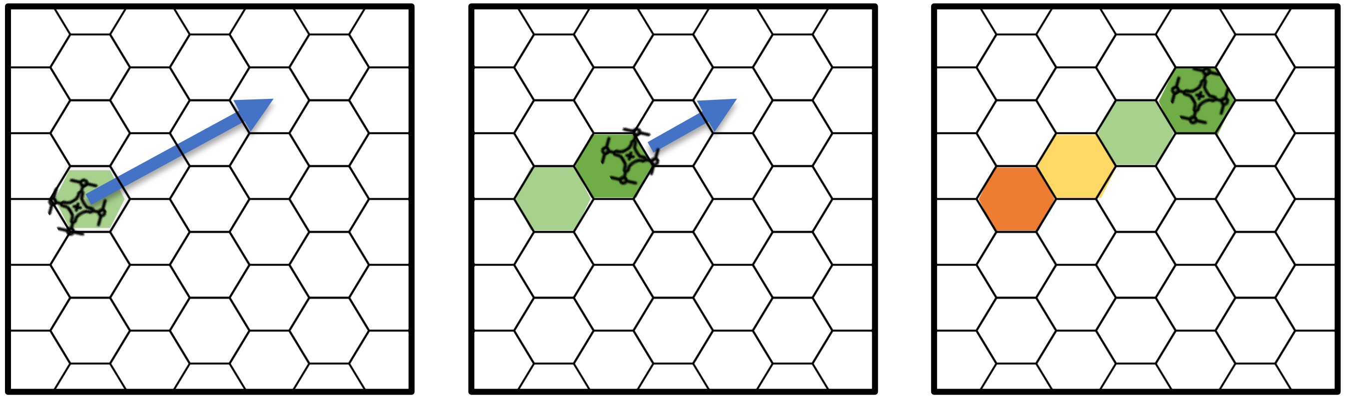
Persistent Surveillance

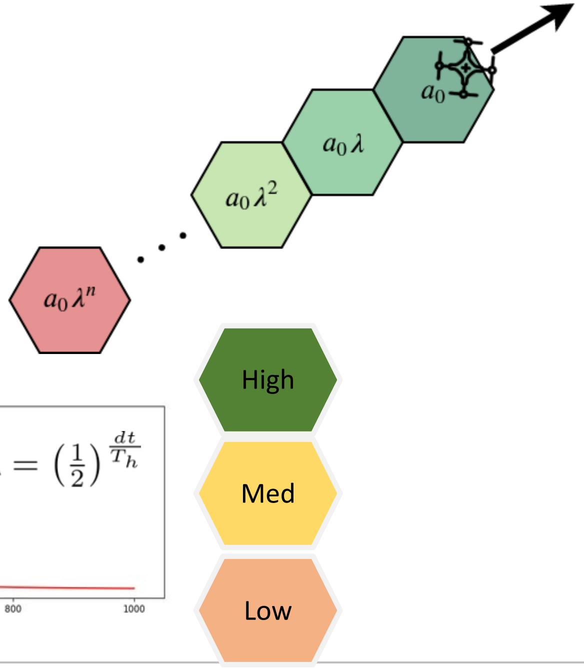
Objective

Maximise Surveillance *Score*
(Sum of all the cells/hexes)

Method

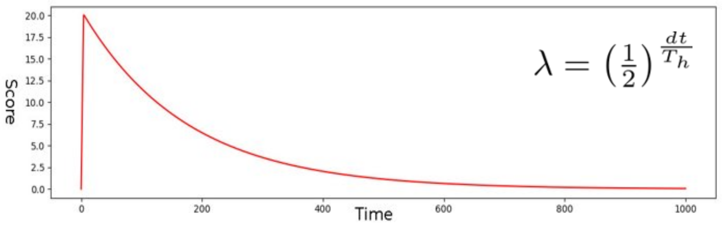
Visit cells to increase scores and revisit to maintain higher scores





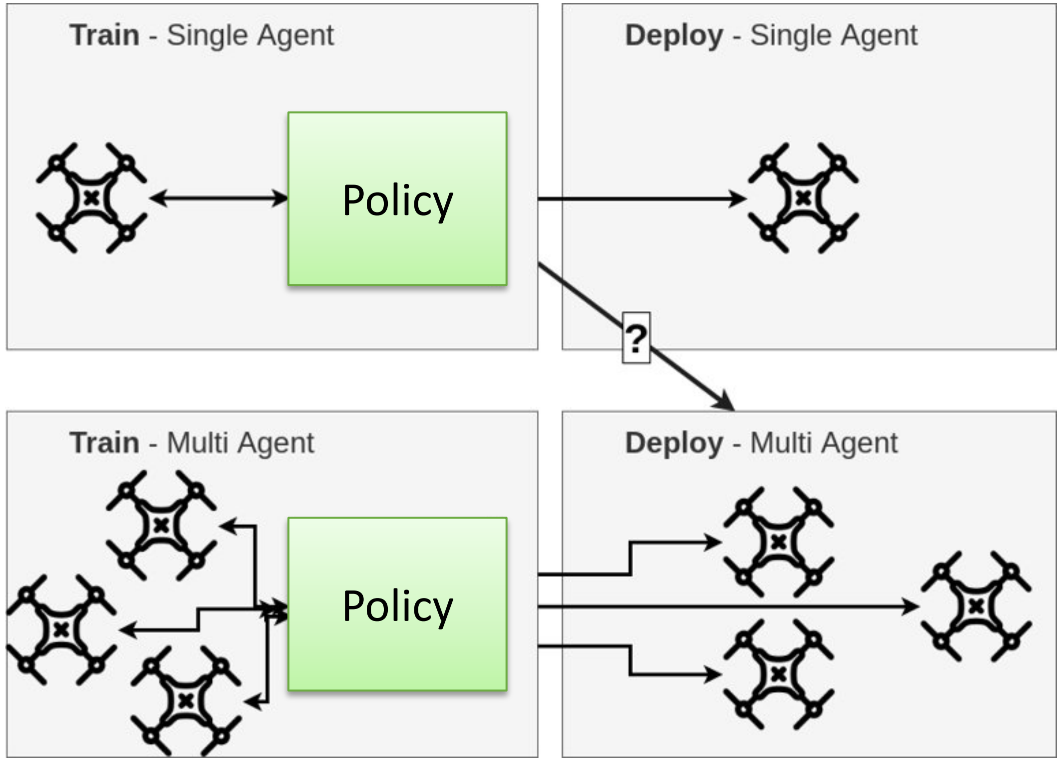
Score Function

Occupied -> Rapid increases
Not Occupied -> Exponentially Decays



Motivating Question

Can we train single-agent policies in isolation that can be successfully deployed in multi-agent scenarios?



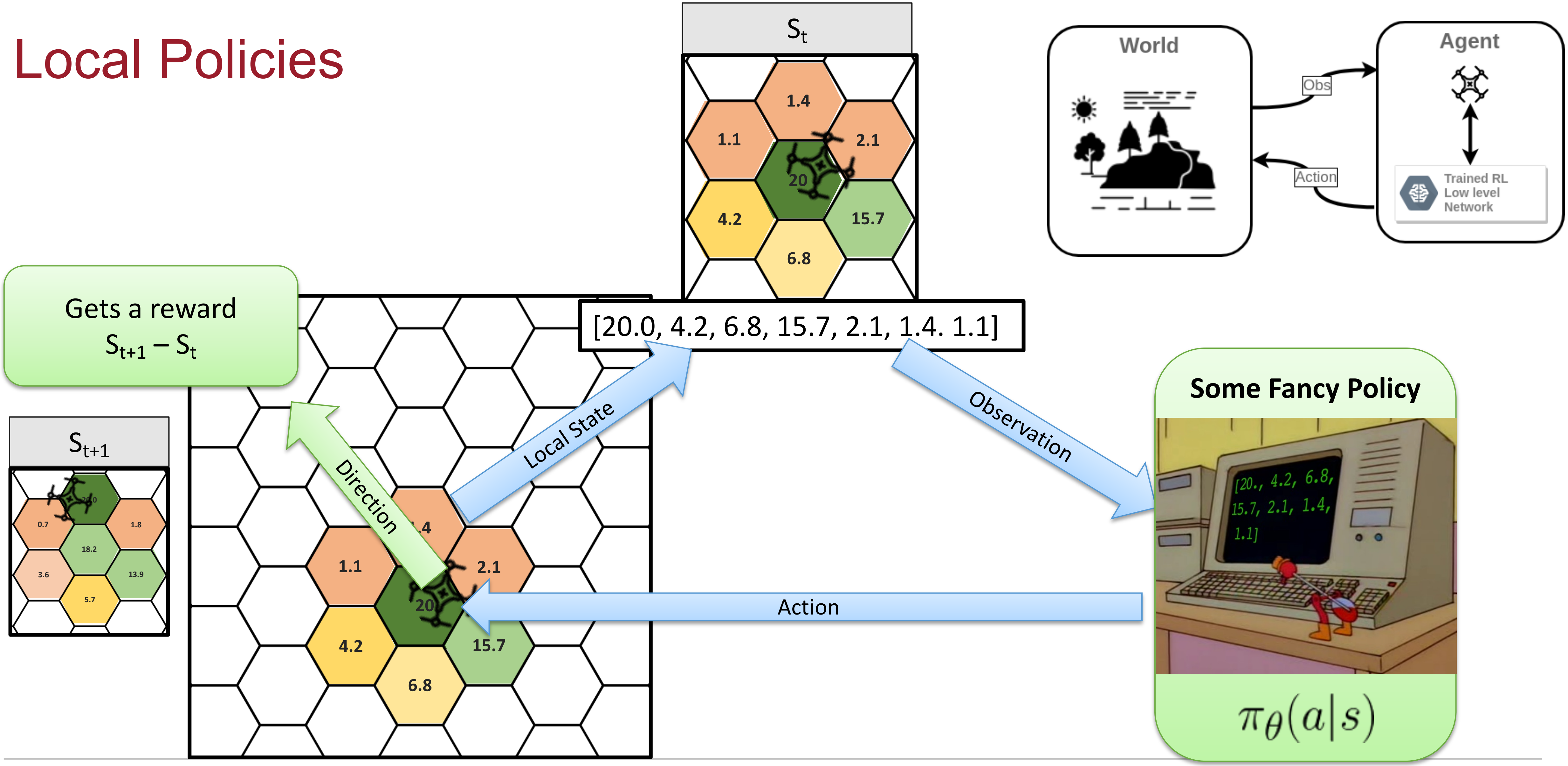
Assumptions

- No Coordination
- No Communication
- Train on a single agent with a single agent environment
- Perfect knowledge of the state

Questions

- Do we need to coordinate?
- Do we need to communication?
- Do these need to be trained for?
- Is perfect knowledge of state of the world beneficial?

Local Policies



Local Policies

Performance

Best

Good

Poor

Heuristics

Random — Move random direction

Gradient Descent — Move towards lowest value

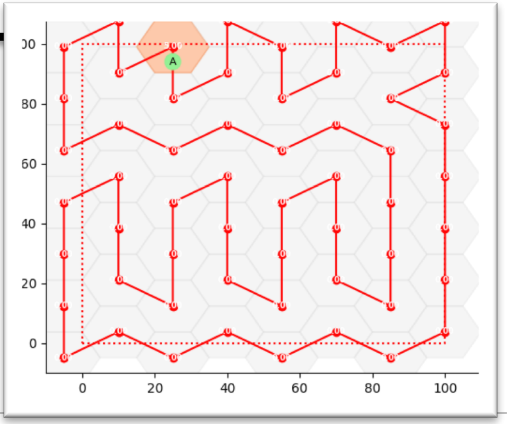
'AI'

DDPG — Deep Deterministic Policy Gradient – Trained neural net – Deterministic policy

NEAT — Neuro-Evolution of Augmenting Topologies – Evolved NN (approximates gradient descent)

Benchmark

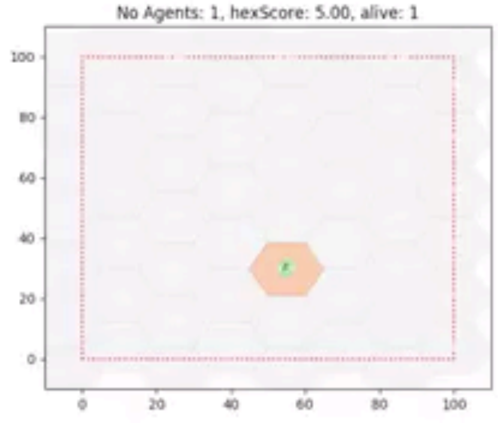
Trail — Pre-defined trail to follow – visiting each hex in turn and continuing in a loop
Requires global knowledge / localisation

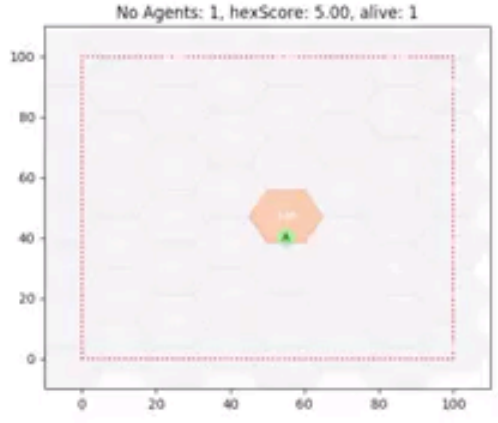


T-B PHASE
T-B Partnership in Hybrid Autonomous Systems Engineering

Comparison of Local Policies

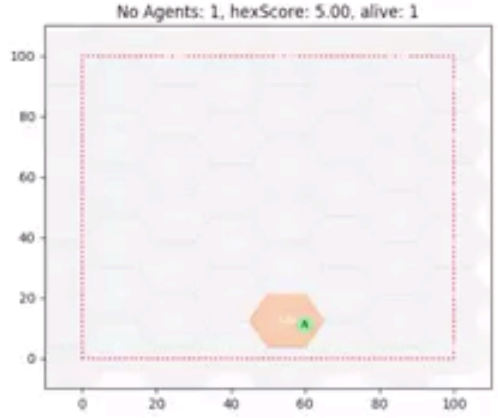
Random

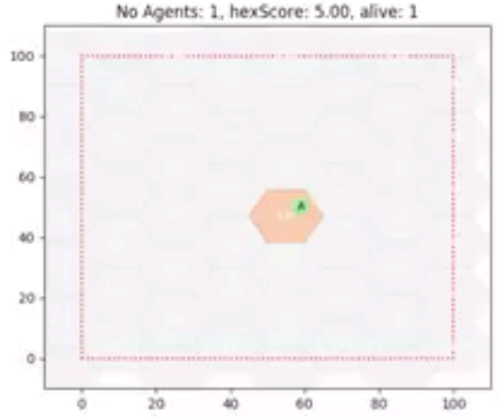




Gradient Descent

DDPG





Trail

Performance

Best

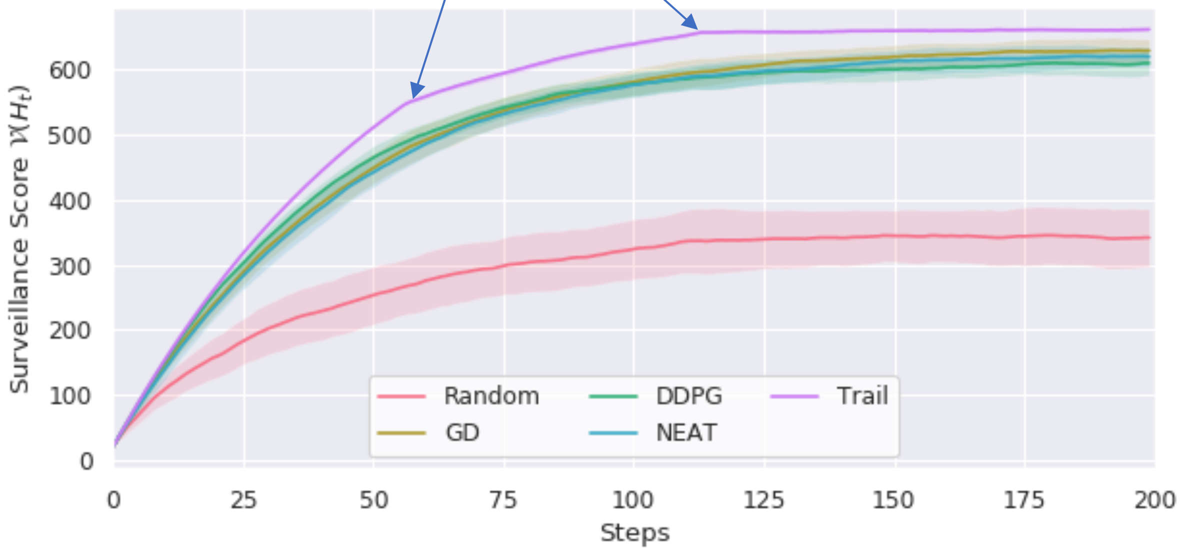
Good

Poor

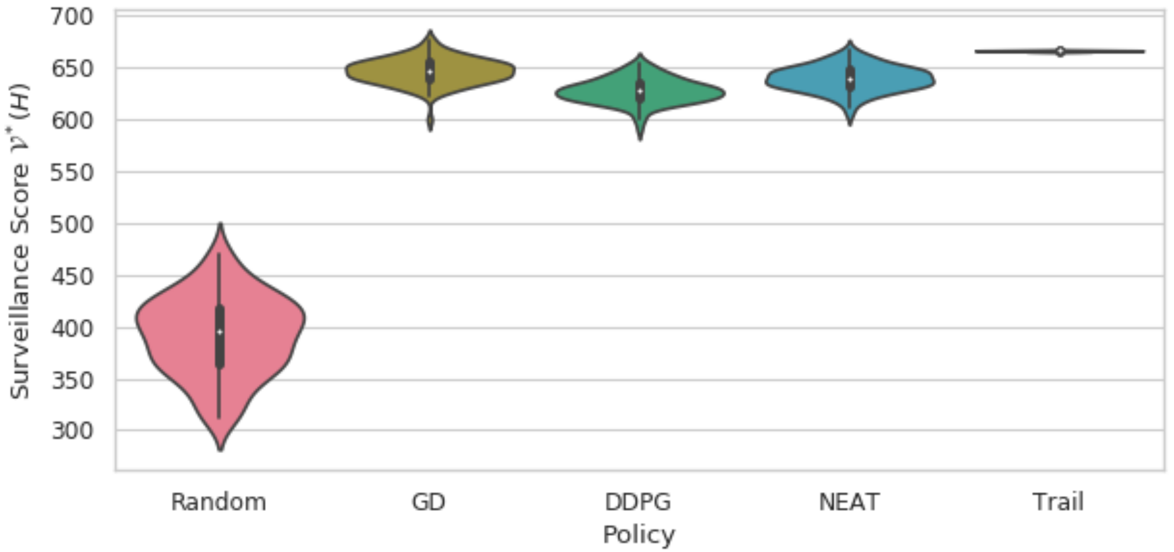
Policy Performance – 1 Agent

Laps around trail

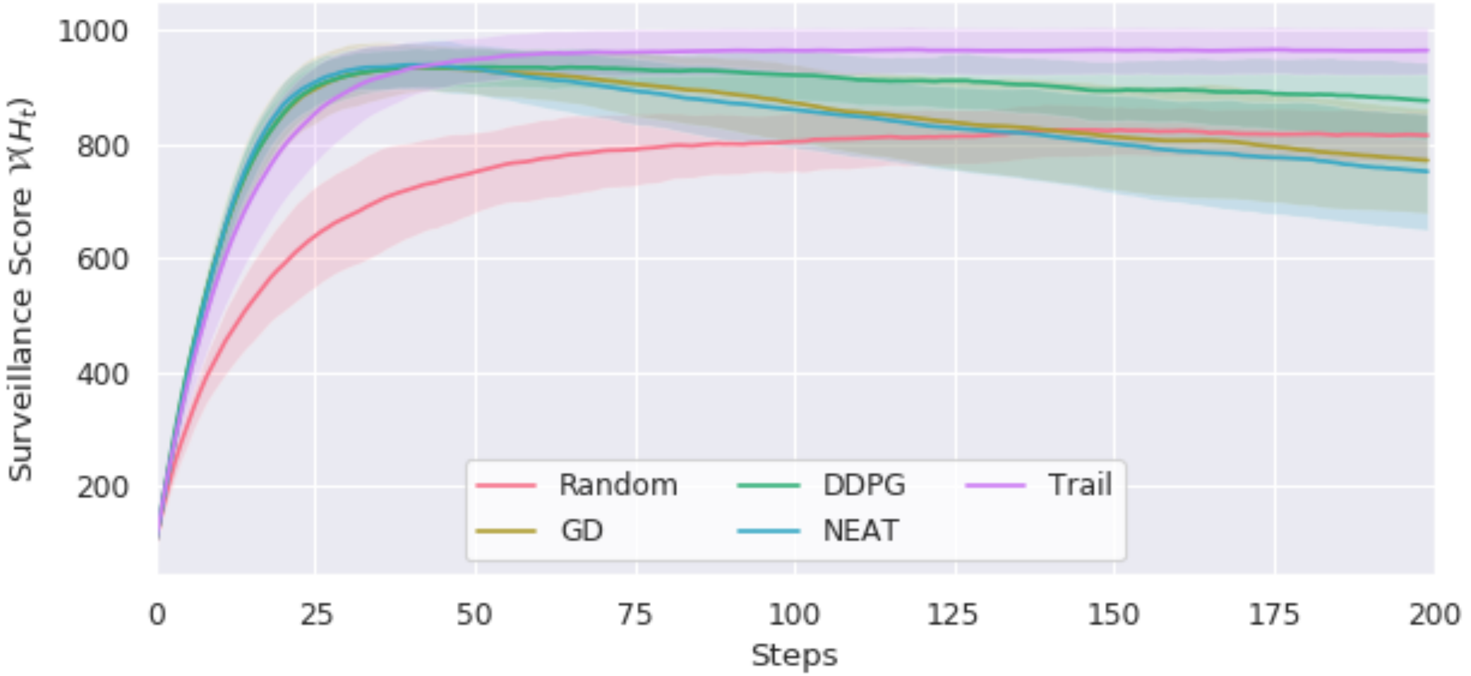
Running Score



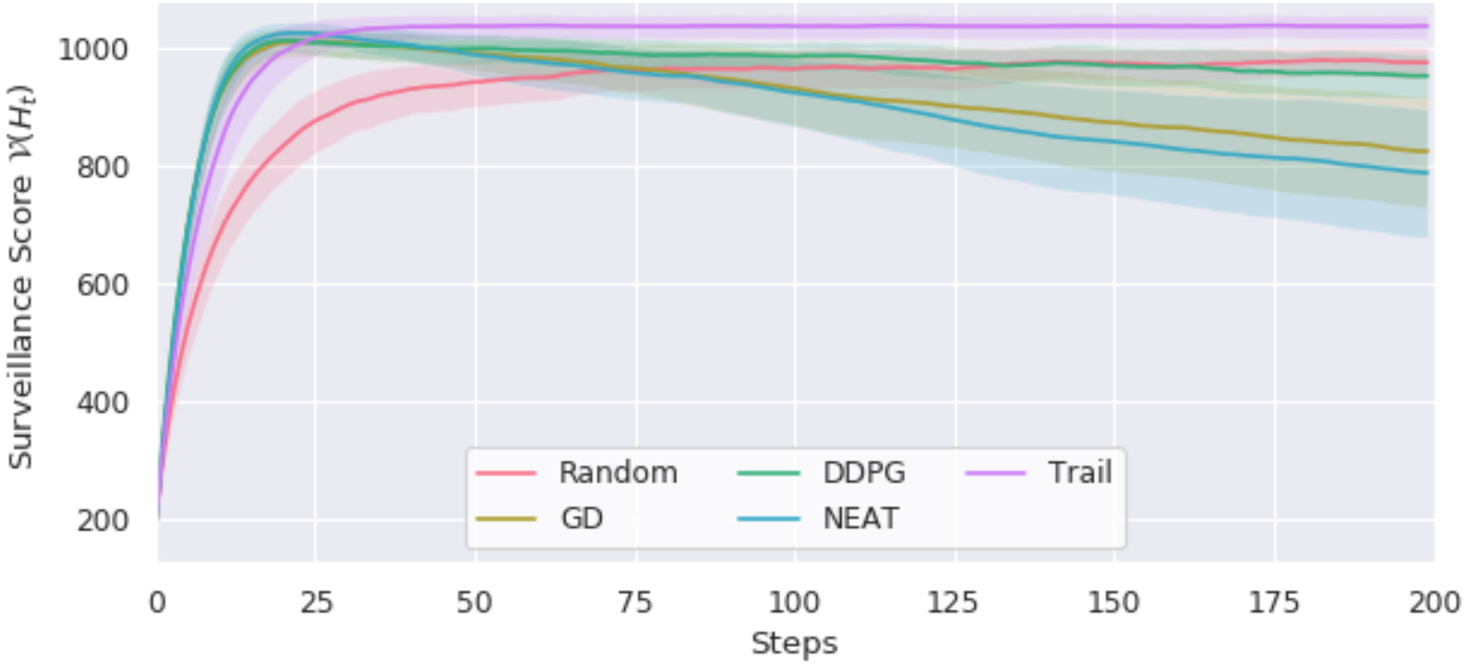
Best Score



5 Agents



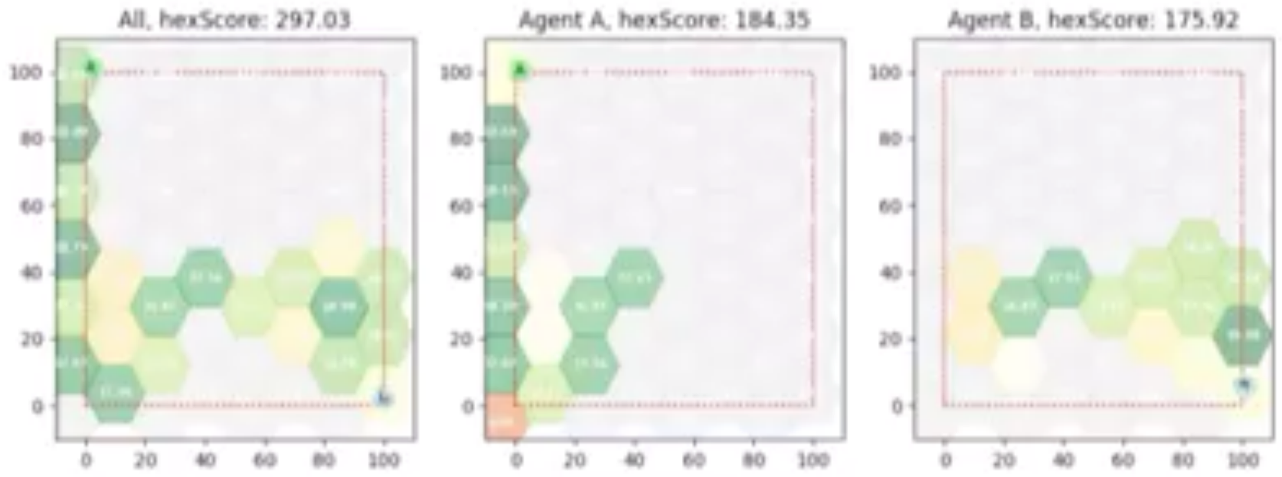
10 Agents



Homogeneous-policy convergence problem

1) Agents move to the same hex
2) Agents get an identical local state observation
3) Identical, deterministic policies π, return identical action choices
4) Agents in the same hex, perform identical actions, and move to the same hex, as the other agents - thus returning to step 1)

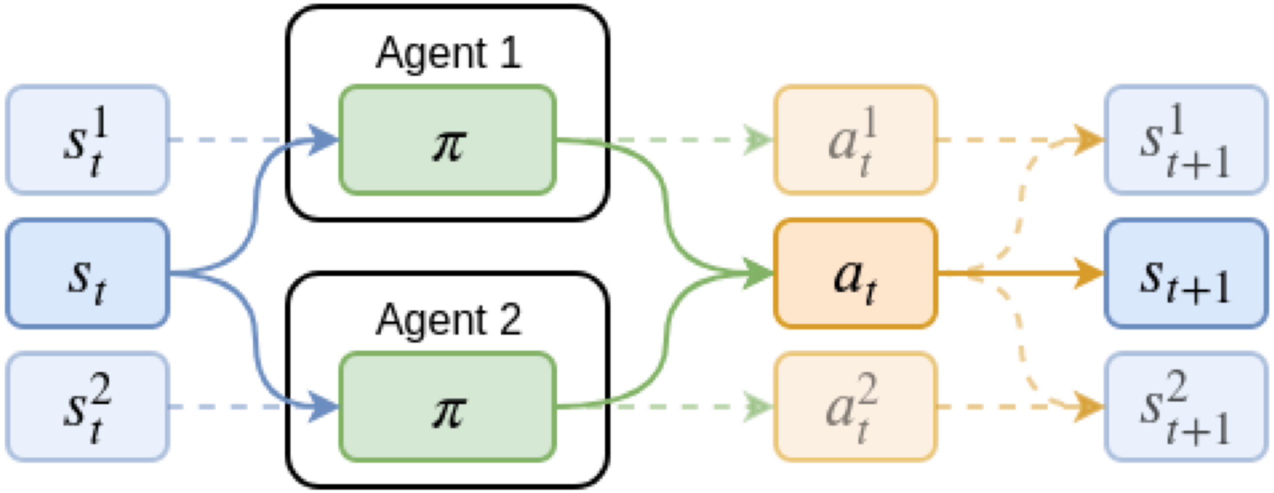
Communication isn’t always beneficial



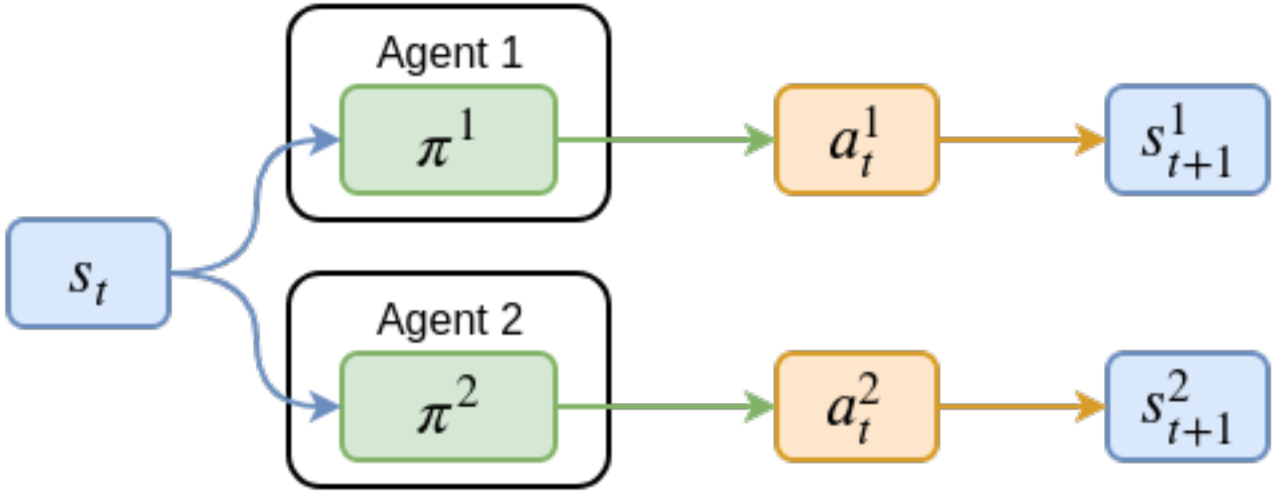
Centralised State

Local States

Homogeneous-policy convergence problem



How to break the cycle:

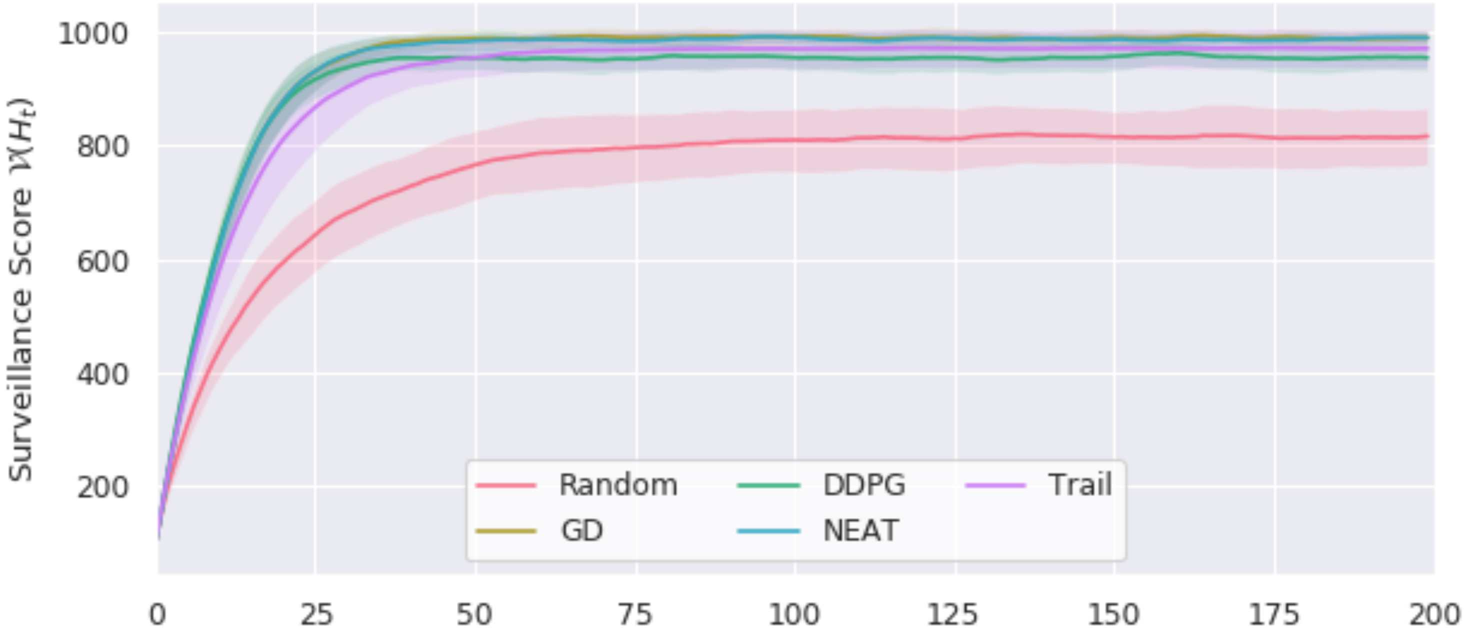


Policy Noise

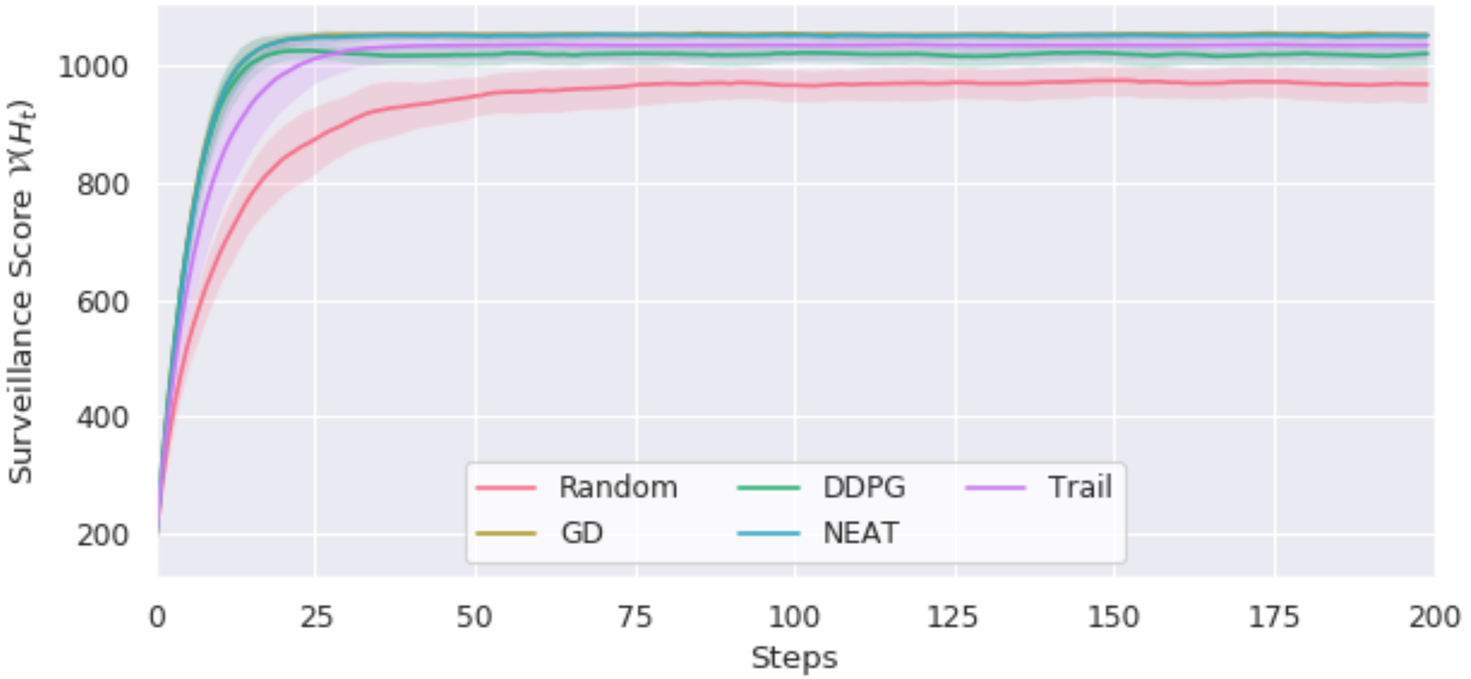
- Distinct policies
- Stochastic policies

Adding State Noise

5 Agents

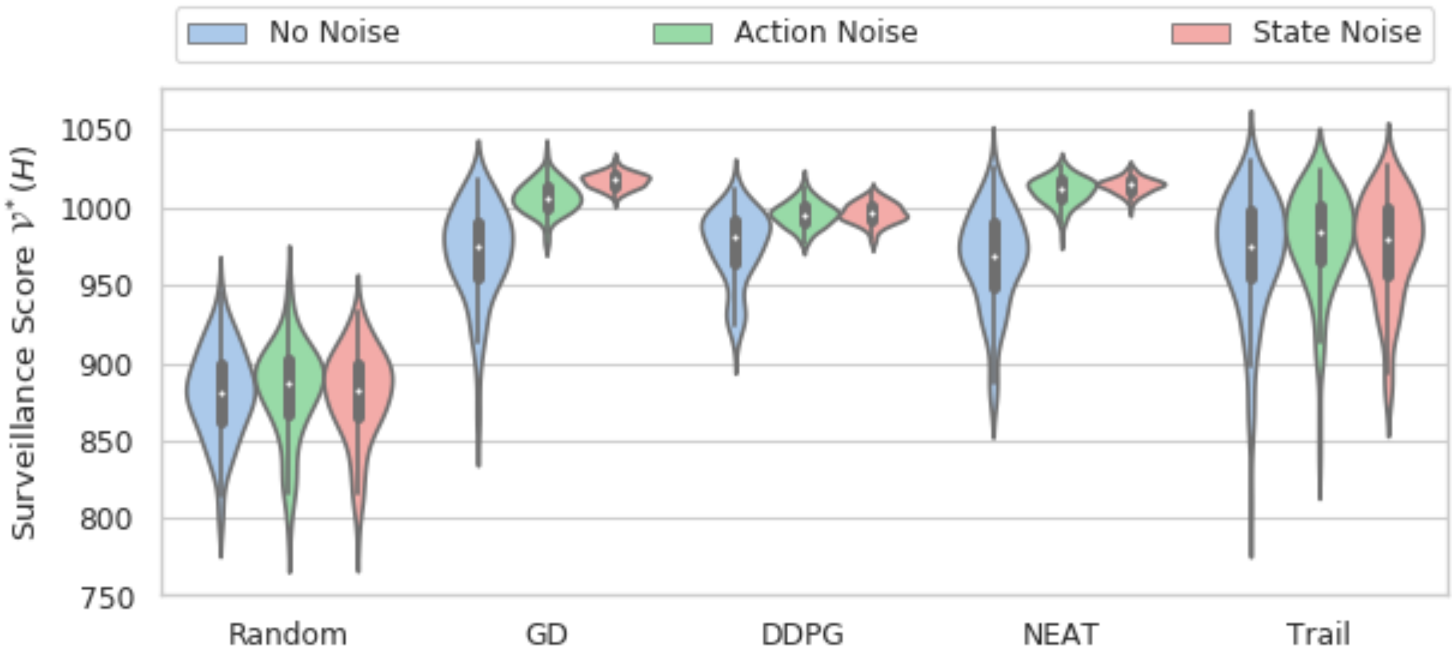


10 Agents

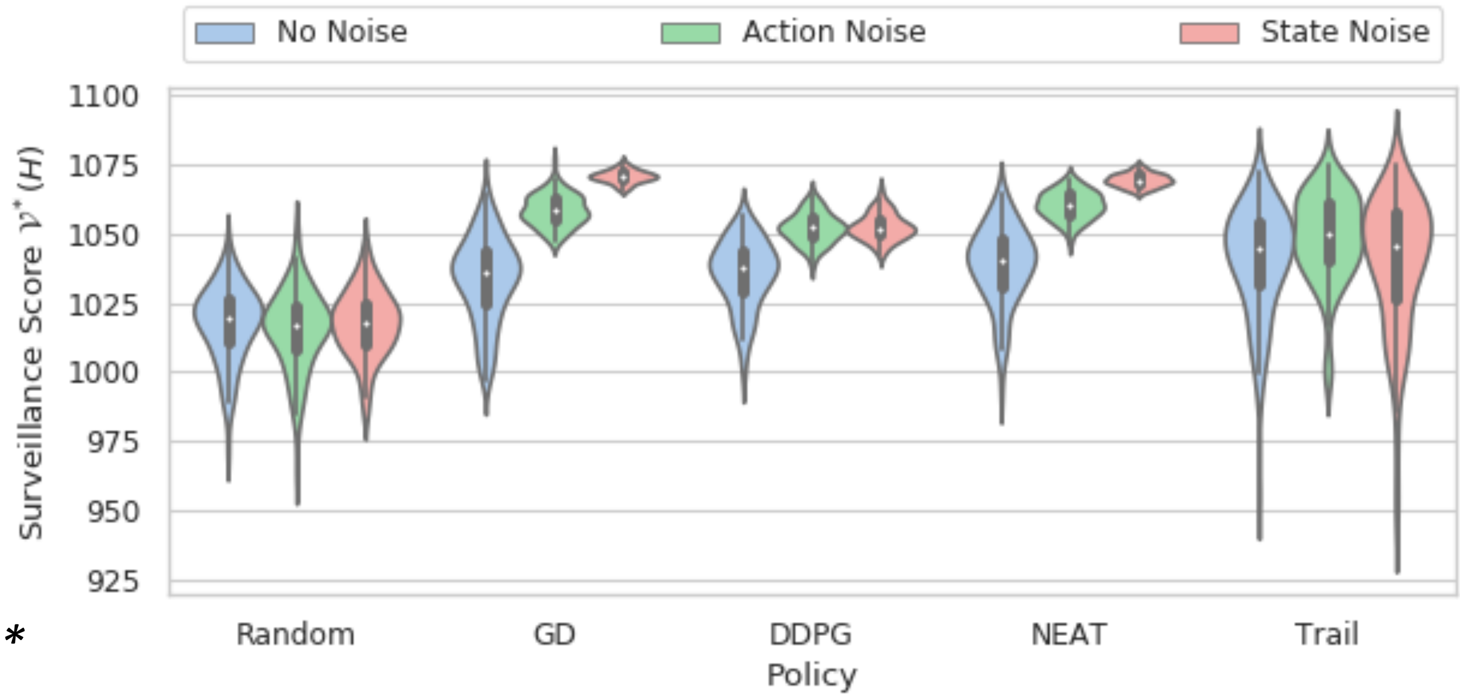


Adding State Noise

5 Agents



10 Agents



T-B PHASE
T-B Partnership in Hybrid Autonomous Systems Engineering

** Non-zero y-axis **

Conclusion

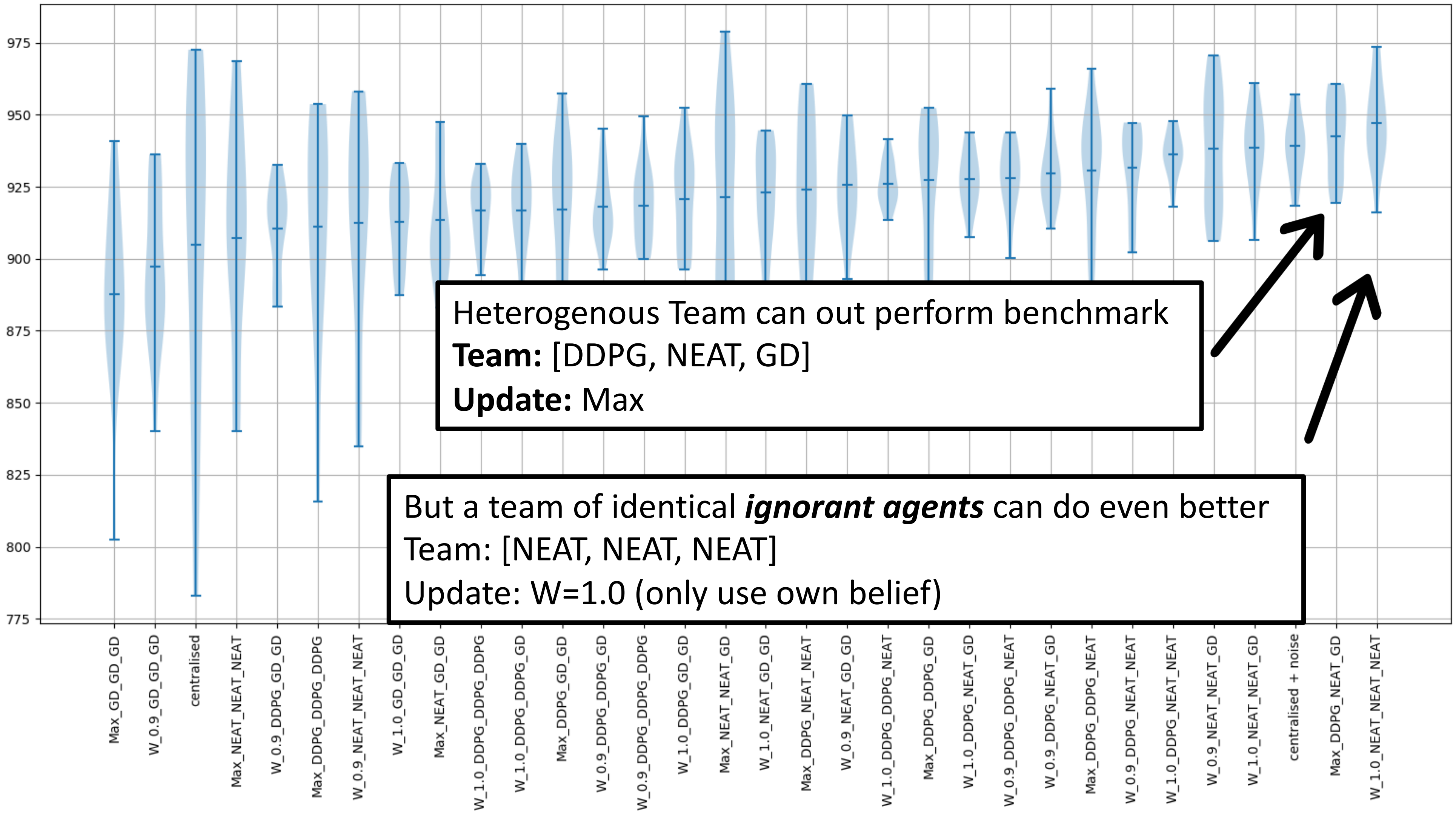
- Short term planning can be effective in solving the MAPSP
- **Agents trained in isolation can still perform in a multi-agent scenario**
  - Global 'trail' policies perform better -> require coordination
  - Simplistic gradient descent approaches perform sufficiently
- **Emergent behaviour**
  - A property almost entirely the result of homogeneity and determinism.
  - This or a similar class of emergent properties could easily occur in other scenarios
- **Homogeneous-policy convergence cycle is a problem** and can be avoided by essentially becoming more heterogeneous
  - **Action stochasticity** – adding noise
  - **State/observation stochasticity** – agent specific state beliefs
  - **Heterogenous policies** – teams of different agents

Questions

Email: Thomas.kent@bristol.ac.uk
tomekent.com

Appendix

Decentralised State Heterogeneous Policies



Heterogenous Team can out perform benchmark
Team: [DDPG, NEAT, GD]
Update: Max

But a team of identical ***ignorant agents*** can do even better
Team: [NEAT, NEAT, NEAT]
Update: W=1.0 (only use own belief)

Team Size
3

Policies
Gradient Descent
DDPG
NEAT

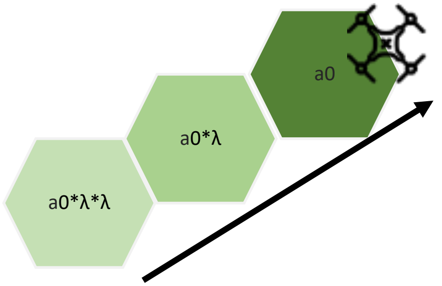
Belief Update
Max
W = 1.0
W = 0.9

Benchmark
Centralised + action noise
Centralised

Theoretical Max

- Number of hexes n = 56
- Hex height (width) = 15m
- Agent speed 5m/s => **3dt to cross**
- Linear Increase per timestep:
  ld = 5 -> adds 15 to the hex so **a0 = 15**
- Th = 120, dt = 3
- If we make a trail around all n=56 hexes we can hit **542**.
- If we continue and re-join 'tail' we can max out each hex so a0 = 20 and we can then hit **723**

$$\lambda = \left(\frac{1}{2}\right)^{\frac{dt}{T_h}}$$



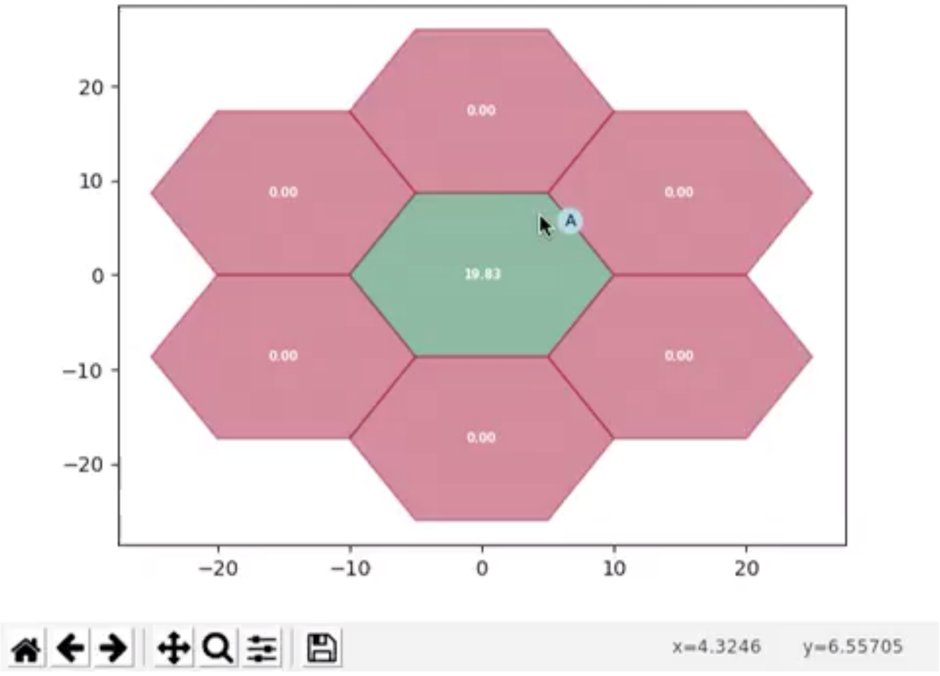
Geometric Series

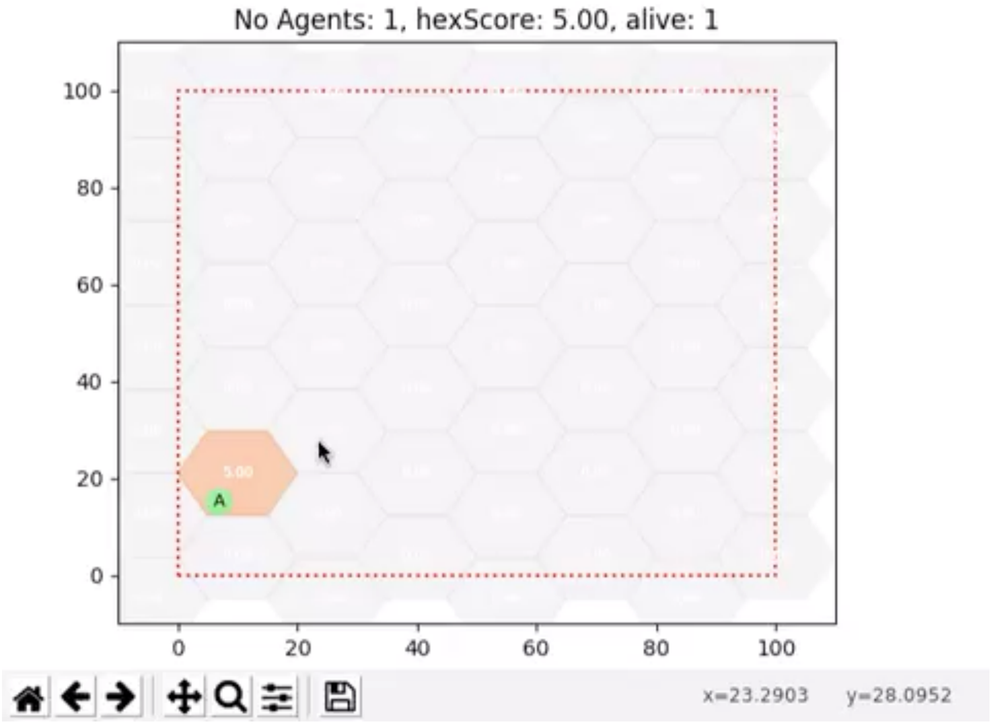
$$a_0^0 + a_0\lambda^1 a_0\lambda^2 + ...a_0\lambda^n = \sum_{k=0}^{n-1} a_0\lambda^k = a_0\left(\frac{1-\lambda^n}{1-\lambda}\right)$$

Multi-Agent: Geometric Series

$$a_0\left(\frac{1-\lambda^{n_1}}{1-\lambda}\right) + a_0\left(\frac{1-\lambda^{n_2}}{1-\lambda}\right) + ... + a_0\left(\frac{1-\lambda^{n_{N_a}}}{1-\lambda}\right)$$

Human input (aka graduate descent)





Local view
- Agent moves in direction of cursor
- Attempt to build global picture & localise
- Users tend to do gradient descent

Global view
- Agent moves in direction of cursor
- Can more easily plan ahead
- Users tend to attempt a trail

Human performance Local/Global State