Reinforcement learning for autonomous and novel behavior in OneSAF

Potential for new TTPs, technology evaluation, and Requirements definitions

Eric Kelly, Ph.D., Senior Deep Learning Architect
Kim Pevey, Ph.D., Research Software Engineer
Dillon Roach, Ph.D., Data Science Engineer

Reinforcement Learning (RL) can enhance OneSAF by adding autonomy to simulation entities

Autonomous entities create potential for:

- Generating new battlefield strategies with minimal human bias
- Helping warfighters learn in a much more creative, stimulating, and instructive environment
- Expanding how the military engages in future conflicts
- Allowing experimentation and evaluation of new technologies
- Informing Requirements definitions for the future force

One Semi-automated Forces (OneSAF)

A simulation tool for modeling real-world representations of combat and non-combat operations

- Used for design, experimentation, analysis, and training, including human-in-the-loop scenarios
- Physics-based (realistic) environment
- Uses behavior models to control entities
- Behaviors are rule-based and customizable

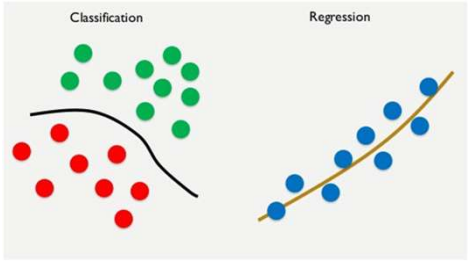
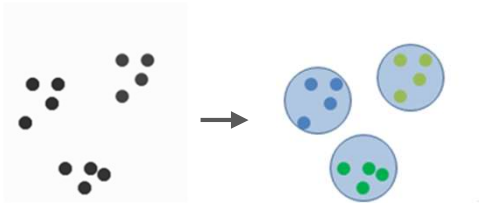
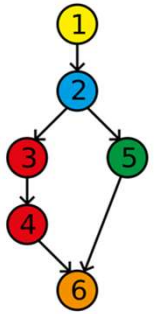
Goal: add RL capability to enhance entity behaviors

https://ict.usc.edu/prototypes/onesaf/#gallery-1

Advantages of RL over rule-based models

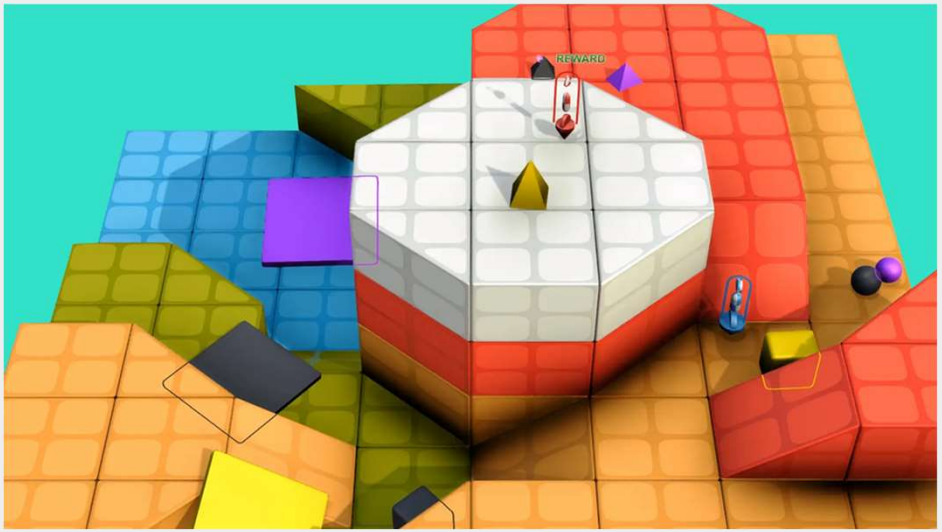
- **Complexity**: Complex situations require more rules than humans can derive
- **Adaptation**: Decisions are made even if situation is new
- **Novelty**: Thorough exploration allows new solutions to be found

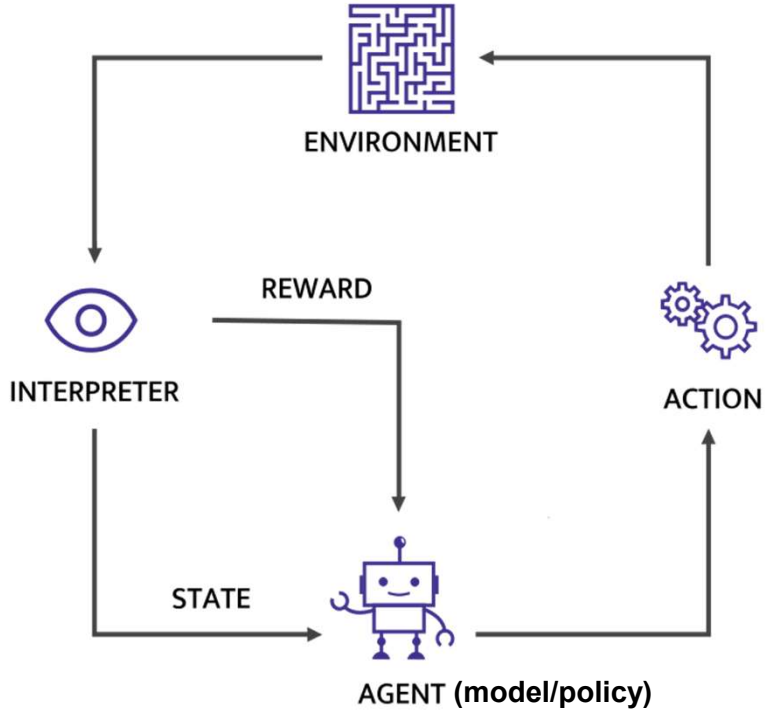
RL is a type of Machine Learning (ML)

	Supervised Learning (SL)	Unsupervised Learning (UL)	Reinforcement Learning (RL)
Data Constraints	Existing Dataset		Create Data via Exploration
Training Data	Labeled (tagged)	Unlabeled	Simulation/Environment
Common Objectives	Classification, Regression	Labeling, Segmentation, Clustering	Decision/Action Sequences
			

RL Is an Exploratory Learning Approach

Learns from potentially infinite examples in environment (complex problem)





Result is a **policy for decision making** that optimizes expected reward

https://deepmind.com/blog/article/generally-capable-agents-emerge-from-open-ended-play

Reinforcement Learning: Applications

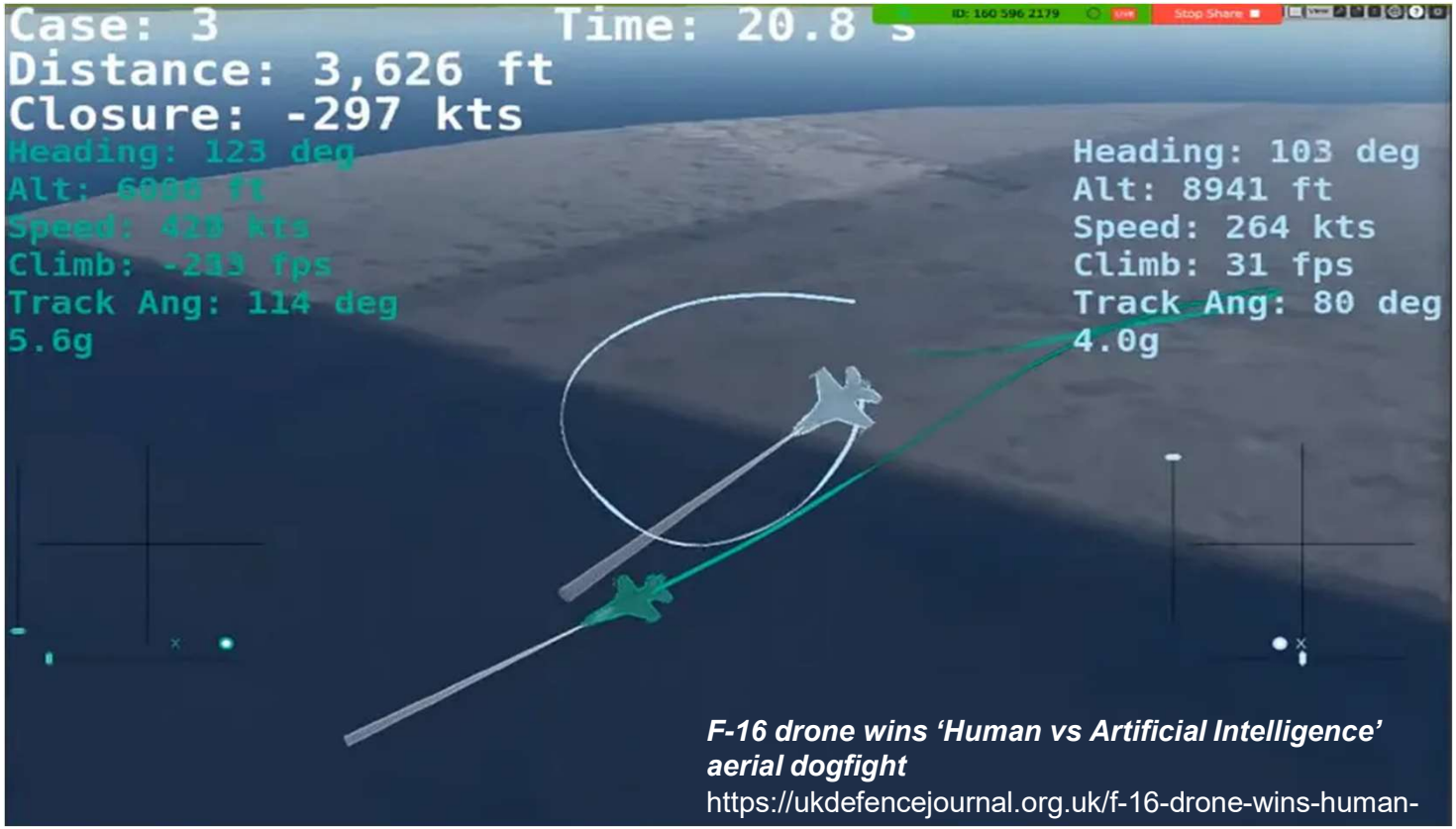
- Manufacturing: Robotics, computer chip design
- Warehouse process efficiency
- **Data center cooling and control (Google)**
- Autonomous navigation of vehicles
- Healthcare Treatment Policy
- Trading and Bidding strategies

https://blog.google/outreach-initiatives/environment/data-centers-get-fit-on-efficiency/

- High level competitive games (DoTA:2, Starcraft2)
  - **OpenAI's 'Five' beats human world-champion team in 5v5 match**
  - AlphaGo/lee/Zero defeat world's top players

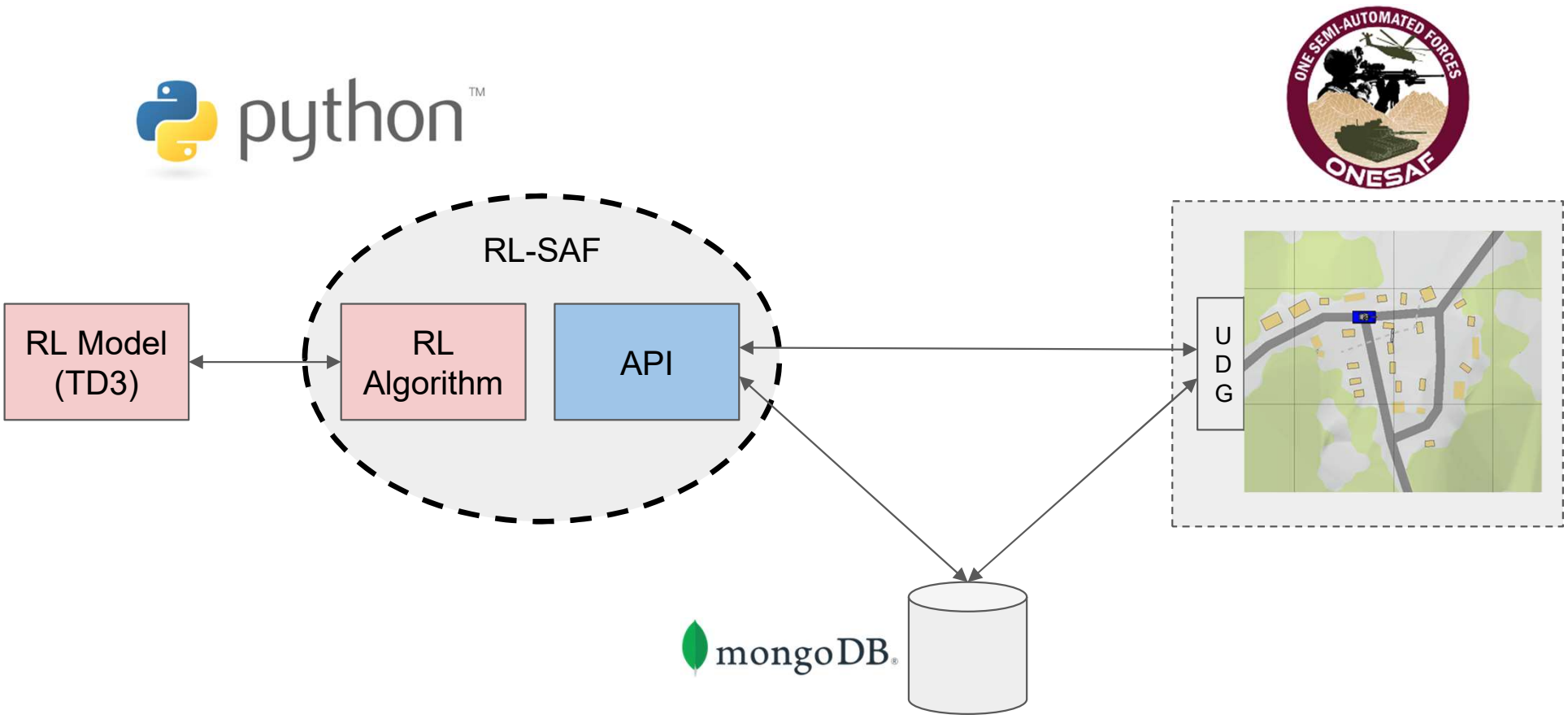
RL in Aircraft Autonomy: Air combat

Complex flight behavior (AI is pilot)

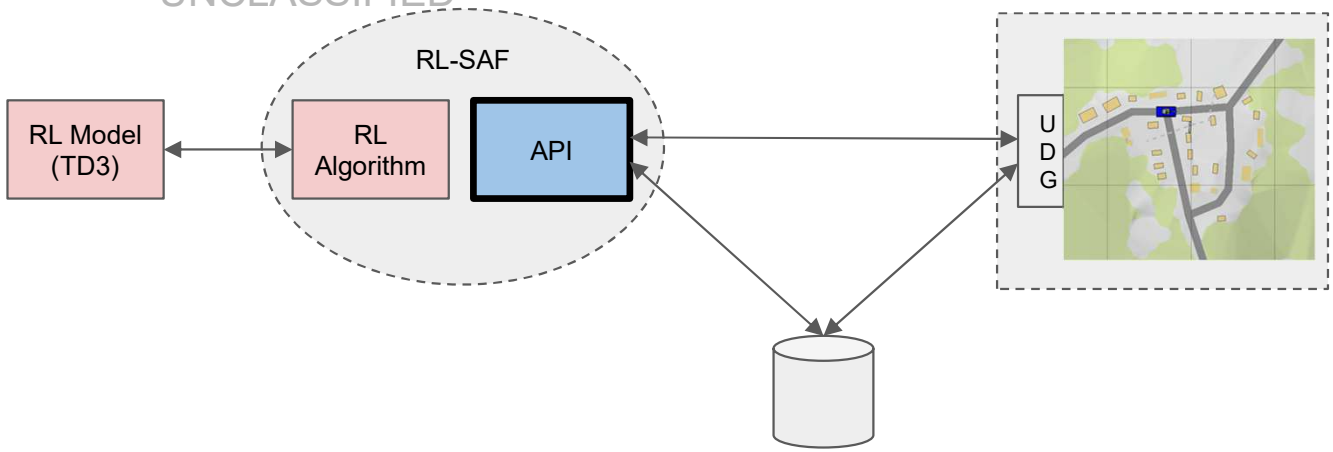


F-16 drone wins 'Human vs Artificial Intelligence' aerial dogfight
https://ukdefencejournal.org.uk/f-16-drone-wins-human-

RL-SAF: RL for OneSAF

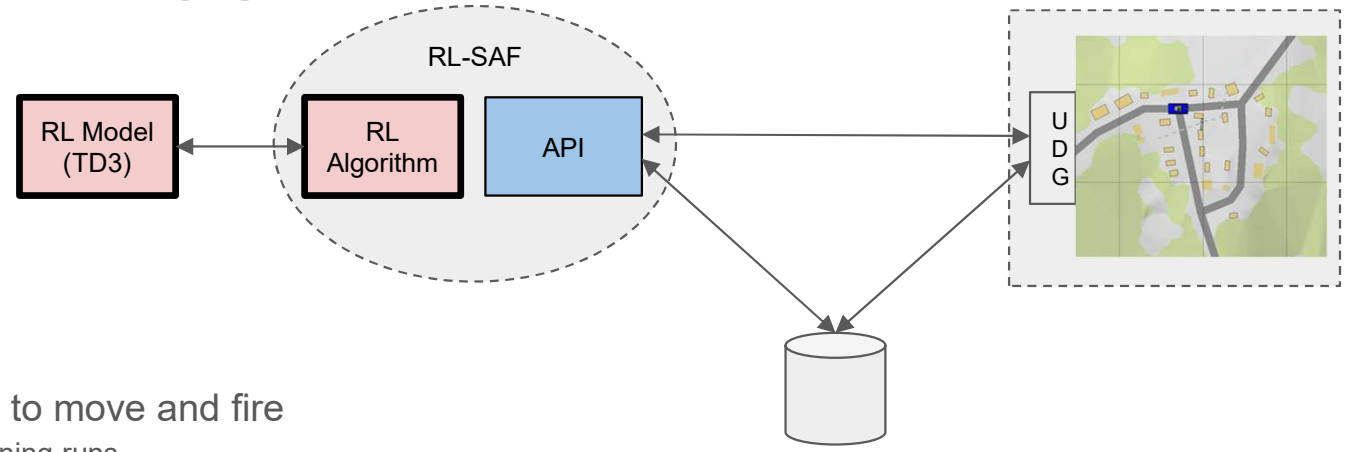


RL-SAF API (beyond RL)



- Controls OneSAF via Python
- Communicates with OneSAF via REST endpoints
- Can run sequences of tasks with or without user interaction
  - Examples
    - Load a scenario via scene file/template
    - Move and orient entities via magic commands
    - Initialize, run, and reset a scenario
    - Get/Set entity and simulation properties
- Independent of RL algorithm

RL Algorithm and TD3



- Makes use of OneSAF commands to move and fire
  - More efficient learning for shorter training runs
- Uses OpenAI Gym framework and TD3 algorithm
  - TD3 works well for continuous action spaces (infinite solutions)
  - Overview: https://spinningup.openai.com/en/latest/algorithms/td3.html
- Observation space (available to RL actor/model)
  - RL actor: speed, heading, distance to destination, heading to destination, health, distance to nearby buildings, whether fire command is queued, simulation time
  - Enemy: speed, heading, distance, bearing, health
- Action space (available to RL actor/model)
  - RL actor: forward distance, forward direction, fire
- Reward (sum of components)
  - Time/speed, distance to destination, health, enemy health, out of bounds penalty

Simulation scenario:

- RL Actor (blue) starts in random locations
- Goal is to kill enemy (red) and occupy town (pink cross)
- Enemy follows patrol loop through town
- Both actors may fire at any time

(First animation will show a different scenario)

Results: What to look for in next few slides

- Movement and navigation
- Firing ability
- Novel solutions: improvement over rule-based behavior models

(Alternate scenario)

- **Movement/Navigation**: smooth and efficient
- **Firing ability**: Blue does not engage Red
- **Novel behavior**: Blue avoids road to hide from Red

- **Movement/Navigation**: Blue navigates in urban environment
- **Firing ability**: Blue does not fire even when conditions are right
- **Novel behavior**: Blue avoids damage while attempting to occupy town

- **Movement/Navigation:** Finds center of town and high point for firing
- **Firing ability:** Blue acquires Red then fires from safe distance (compensates for poor firing performance)
- **Novel behavior**: Fires from distance, potentially taking advantage of Red's firing accuracy at long range

- **Movement/Navigation:** Blue moves along tree line behind buildings
- **Firing ability:** Blue should fire earlier
- **Novel behavior:** Blue hides and waits for Red to stop patrolling, then fires

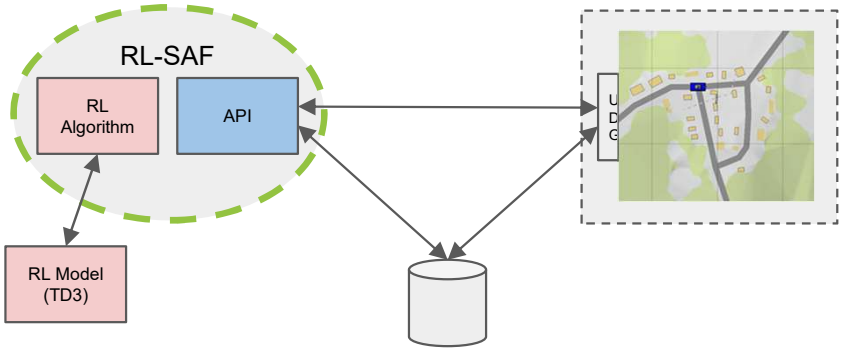
Current RL model state

Topic	Result		Explanation / Details
Movement and navigation	☆☆☆☆	• Fast and efficient • Easily avoids obstacles/threats • Goes to destination	**Easy learning** • Abundant feedback (every simulation step)
Firing ability	☆	• Effective when conditions are optimal • Rarely fires immediately	**Difficult learning** • Delayed firing chain feedback (need more observations) • Infrequent enemy engagement (need more directed training)
Novel solutions	☆☆☆☆	• Learned to achieve goal in realistic situation (loss of firing capability) • Hides to avoid enemy • Waits for enemy to stop, then fires • Acquires enemy, then moves far away to fire	**Few constraints and strong adaptation** • Essentially no rules (find a way to win) • Lack of firing capability is partial driver
Improvement over rule-based behavior models	☆☆☆	• Uses best capabilities • Adapts to situation • Learns enemy behavior (without human input)	**Few constraints** • Essentially no rules (find a way to win) • Rule-based firing is currently better, but with improvements, RL firing should excel

RL-SAF enhances OneSAF by adding autonomy to simulation entities

Autonomous entities create potential for:

- New battlefield strategies
- Helping warfighters learn
- Expanding how the military engages in future conflicts
- Allowing testing and evaluation of new technologies
- Informing Requirements definitions for the future force



High-level explanation of reinforcement learning:
https://quansight.com/post/exploring-reinforcement-learning

Practical RL Short Course (deeper dive and model building):
https://github.com/Quansight/Practical-RL