Comparing Genetic Algorithms, Q-Learning, and Hybrid Evolutionary-RL Strategies for Light-Seeking Braitenberg Robots in Simulation and Reality

Ruairi Bradley

University Of Sussex

1 Introduction

In the field of robotics and AI, adaptive intelligence is a focus for study. With Adaptive Intelligence referring to machines that can learn and interact with their environment. Even if the environment changes around them. Creating intelligent systems. Two well researched areas to build this kind of agent are evolutionary algorithms and reinforcement learning. Evolutionary algorithms are designed to mimic natural selection, where a population mutates, competes, and gradually improves over generations. On the other hand, Reinforcement learning, teaches a single agent through trial and error, rewarding good moves and punishing bad ones. Both methods produce useful behaviour and solutions over time, but how they solve tasks come with their own strengths and limitations. This study compares three approaches against each other. Firstly, a Genetic Algorithm (GA). Secondly, a standard Q learner (Reinforcement learning). And lastly, a hybrid that evolves Q learning agents with a GA to observe if blending the two approaches improves performance or reliability. All three systems control a Braitenberg vehicle - a small robot with two light sensors and two wheels. Their shared goal is simple; start from random positions and orientations, then drive towards a light source. Training happens entirely in simulation, with light locations and starting conditions shuffled every run so the robots pick up genuinely flexible strategies (generalisation) instead of memorising specific patterns which was noted without the randomised start position. To analyse how well these strategies generalise, each trained agent faced five brand new light placements, first in simulation and then on a real robot to show the effects of bridging the reality gap. Doing so, is famously tricky, thanks to real world noise, imperfect sensors, motors, or environment and hardware quirks that simulations don't have to deal with. Earlier work shows that evolutionary and reinforcement methods can both produce adaptive robot controllers, while hybrids often strike a neat balance and improve robustness seen in the work by (Patle et al., 2022; Carvalho Santos et al., 2014; C. Chen et al., 2008). Q learning has been widely studied on its own for navigation because it's conceptually simple and copes well with uncertainty in such works as (Huang et al., 2016; Wicaksono, 2011). The aim here is to find out which approach: pure GA, pure Q learning, or the GA evolved Q learner handles the light seeking task best in simulation, and then especially when it comes to adapting to fresh scenarios and holding up in the real world.

2 Hypothesis and Methods

This study hypothesises that:

> *"Reinforcement learning agents trained with Q-Learning will have superior performance in the task of generalising to approach unseen light positions in simulation and reality compared to Genetic Algorithms (GA), while a hybrid system combining the two could improve it further. "*

This hypothesis is motivated by the difference in the way the models learn. Reinforcement learning (RL) enables an agent to adapt to their environment through trial and error feedback during their lifetime. However, genetic algorithms (GAs) rely on evolution at the population level over generations - highlighting the difference in the paths the systems take. Because of these differences, RL agents are predicted to have better generalisation to novel and new environments based on their interaction-driven, real-time learning, where genetic algorithms might struggle in comparison as they don't have this ability.

To try and optimise a solution and further investigate the relationship, a hybrid approach was introduced that evolves Q-learners using a GA. The combination of adaptive learning from RL and structural optimisation of evolution, especially early in the lifetime, led to the prediction that the hybrid model will balance policy flexibility with the robustness acquired from evolution, potentially leading to a better sim to real transfer than either pure model.

To test these predictions, all three agents were evaluated: -GA, Q-Learning, and Hybrid GA+QL-on their ability to:

1. Generalise to previously unseen light source positions in simulation after training
2. Transfer learned behaviour to a physical robot in real-world trials, to test the robustness of learned policies against the noise and variability of the real world

In this setting, generalisation is referring to an agents ability to move towards the light successfully in the unseen test positions, after training on randomly positioned lights. This was done with the aim of displaying genuinely learned and adapted behaviour rather than specific light location memorisation.

3 Methods

3.1 Overview

Braitenberg vehicles were trained using three learning methods:

- Genetic Algorithm (GA) - evolution-based optimisation of weights that control motors
- Q-Learning (QL) - a discrete, model-free reinforcement learning method
- Hybrid GA + Q-Learning - Q-learning agents evolved via a GA to optimise the Q-tables

Each method was trained under the same three stimuli. Being exposed to 2, 5 or 10 randomly positioned light sources, in a 2D continuous simulation environment. The best performers were then tested on the five unseen lights, subsequently transferred and tested on the real robot, for analysis on how the systems hold up and to analyse the reality gaps impact on performance.

3.2 Simulated Environment

- Implemented in Python using NumPy and matplotlib

- 2D continuous plane bounded in $x, y \in [-3, 3]$
- Sensor values follow inverse-square law to simulate light intensity
- Gaussian noise added to sensor/motor output ($\sigma = 0.05$)
- Start positions sampled within the range (2.5, 3.5) around (3, 3)
- Starting angles were sampled from: [0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°]

3.3 Physical Environment

- Constructed using the EduBot platform + Raspberry Pi Pico microcontroller
- Analog light sensors (Left = GP28, Right = GP26)
- Motor speeds based on agent policy: evolved genotype (GA), learned Q-table (QL), or hybrid design
- Scale matched to simulation: 1 sim unit was equated to 30 cm in real world
- Each trial: 20 seconds cap, manual observation and timing if reached light before time limit
- Start positions and bearings randomised to align with simulation
- Light source locations match unseen simulation test lights

3.4 Agent Architecture

All agents shared a Braitenberg-style controller with sensor-to-motor mappings:

Motor equations:

$$\text{Left Motor:} \quad L = w_{ll} \cdot I_l + w_{rl} \cdot I_r + b_l$$
$$\text{Right Motor:} \quad R = w_{lr} \cdot I_l + w_{rr} \cdot I_r + b_r$$

Where:

- I_l, I_r = left and right sensor values
- w_x = weights connecting sensors to motors
- b_l, b_r = motor biases

Each policy is defined by a vector:

$$[w_{ll}, w_{lr}, w_{rl}, w_{rr}, b_l, b_r]$$

3.5 Adaptation Strategies

3.5.1 Genetic Algorithm (GA) The GAs hyper-parameters were tuned for best performance. Population size of 50 genotypes, generations of 200 for the 2 light set up, 300 for the 5 light and 500 for the 10 light to allow enough time for convergence and behaviours to emerge. Elitism was used for selection, keeping the top 5 individuals from each generation with a Mutation & Gaussian noise ($\mu = 0$, $\sigma = 0.4$). Crossover wasn't implemented and the fitness function was defined as distance reduction and a + 1.0 bonus if agent reached within 0.5 of the light source.

3.5.2 Q-Learning (QL) For the Q-learner, the hyper-parameters were defined as follows. The action space was 5 predefined behaviours such as soft left, hard left, forward and so on. The state space was a 10x10 bins set up. Next, the 2 light trained version ran for 10,000 episodes, while the 5 and 10 light ran for 20,000 to allow for more training in a more complex environment. Learning Rate (α) of 0.3 $\rightarrow$ 0.1 (decay), with an exploration rate of (ϵ) 1.0 $\rightarrow$ 0.05 (decay). Finally, a reward function of $\Delta d \cdot 2 + 5$ bonus if within 0.5 units was applied.

3.5.3 Hybrid GA + Q-Learning Lastly, the Hybrid was set up with a population size of 20 Q-learners, which ran for 200 generations (2 and 5 lights) and 150 generations (10 light). The episodes per evaluation was 30 per light. Mutation & Gaussian noise ($\sigma = 0.5$) applied to Q-tables. And the fitness metric was average reward across all episodes.

3.6 Evaluation Procedure

Each trained agent (GA, QL, Hybrid × 2, 5, 10 lights) were evaluated the same for consistency across analysis:

Simulation Testing 5 fixed "unseen" test lights at: $(-1.5, 1.5)$, $(2.5, -1.5)$, $(-2.5, 1)$, $(1, 2.5)$, $(3.5, 0)$, with 10 trials per light giving 50 total per agent. For analysis, we tracked the success rate (if agent reached within 0.5 distance) and the performance curve (GA: fitness and QL/Hybrid: reward).

Real-World Testing The same test positions were used as simulation, but scaled to the real world, see Appendix 1 for the real robot code. 20 trials per agent were ran, with 2 start positions X2 start angles X5 test lights. Metrics tracked were: success (if agent reached within 0.5units of light), final distance (cm), time taken to reach light (capped at 20sec) and behavioural observations (circling, minimal drift etc).

3.7 Statistical Analysis

Paired t-tests were conducted for GA vs QL, GA vs Hybrid and QL vs Hybrid. And statistical visualisations produced were temporal line plots (fitness / reward over time), bar charts (unseen success rates) and comparative plots across agent types also.

Diagrams

Figure 1 shown below details the schematic of the Braitenberg vehicle setup, showing the left/right sensors, light source, and motor wiring.

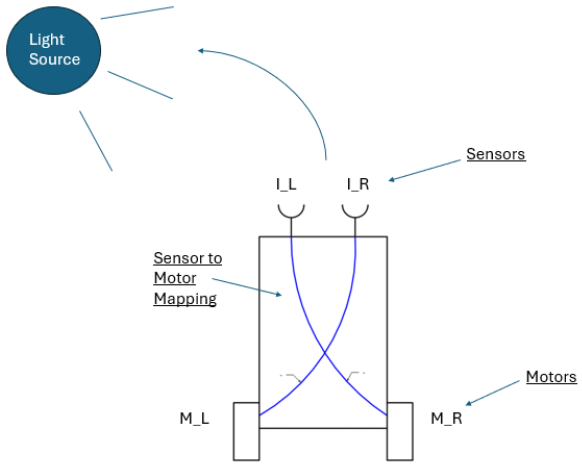

Figure 1. Schematic of Braitenberg vehicle with sensor motor connections.

4 Results

4.1 Learning Performance Over Time

Figures 2 to 5 show the smoothed reward (for trend clarity) and fitness values over time for the Hybrid GA+QL, Genetic Algorithm (GA), and Q-Learning (QL) models under 2, 5, and 10 light training conditions.

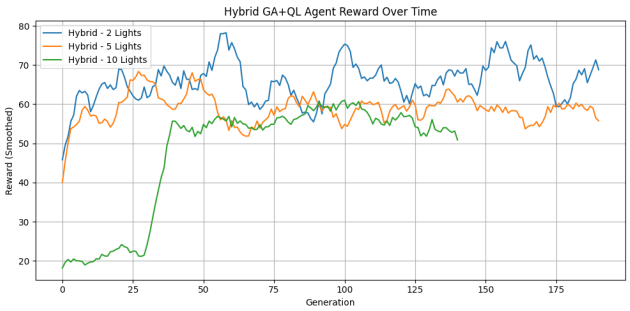

Figure 2. Hybrid GA+QL reward over generations. The 2-light condition peaks highest, due to simple environment, with 5-light showing the most consistent performance.

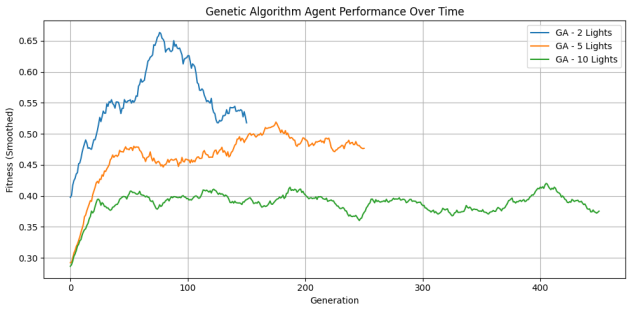

Figure 3. GA reward over generations. Early improvements are followed by plateauing, especially for 10-light training in the most complex setting.

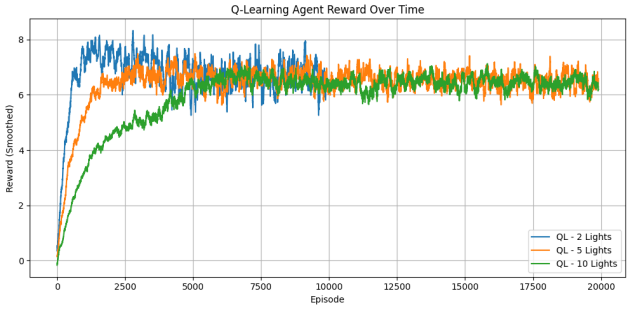

Figure 4. Q-Learning reward progression. All configurations improve early; 10-light shows slower but steady gains.

4.2 Generalisation to Unseen Light Positions

Success rates were recorded for each agent type across five unseen light positions. Table 1 shows these results. The Hybrid model consistently achieved the highest success rates across all training conditions. Q-Learning outperformed GA in all cases.

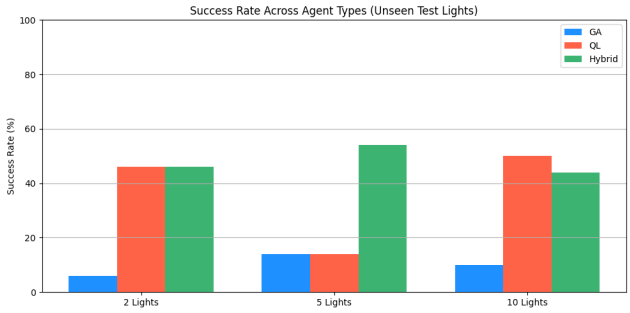

Figure 5. Success rate comparison across unseen lights. Hybrid agents outperform QL and GA at most training levels.

Table 1
Average Success Rate on Unseen Test Lights (% over 5 positions)

Training Lights	GA	Q-Learning	Hybrid
2	7	46	46
5	15	17	54
10	11	50	44

4.3 Statistical Comparisons

Table 2 presents p-values and statistical significance results.

Table 2
Paired t-Test Results

Comparison	Training	p-value	Significant?
GA vs QL	10 Lights	0.0474	Yes
GA vs Hybrid	2 Lights	0.0309	Yes
GA vs Hybrid	5 Lights	0.0189	Yes
GA vs Hybrid	10 Lights	0.0961	No
QL vs Hybrid	2 Lights	1.0000	No
QL vs Hybrid	5 Lights	0.0111	Yes
QL vs Hybrid	10 Lights	0.5529	No

4.4 Real-World Performance

The best trained agent from each setup was transferred and trialled on the physical robot. Table 3 summarises the success rates and behaviour descriptions of said trials. Overall, the Hybrid and QL agents performed better than GA, corresponding with sim results.

4.5 Simulation vs Real-World Comparison

Table 4 compares simulated success rates with real-world outcomes. While GA showed slight improvement in real tests, Hybrid suffered the largest sim-to-real performance drop.

5 Discussion

By comparing the three learning stratagies against one another, with the end goal of learning light seeking behaviour, it provided insight into how each model achieves the task and their potential advantages over other models.

Based on the original hypothesis, QL agents were expected to outperform GA agents on the surface due to their capacity to adapt during their lifetime, aligning with the goals of the task. Also, the Hybrid model was predicted to beat or be competitive with the QL, as it combines the structural optimisation from evolution with the

Table 3
Real-World Performance Summary average % across 5 light pos trials

Model	Success (%)	Distance (cm)	Time (s)	Notes
GA	13.3	71.2	18.8	circling
QL	33.3	33.6	14.6	mid-path corrections
Hybrid	36.7	27.0	11.2	minimal drift

Table 4
Simulation vs Real Success Rates (Averaged)

Model	Sim (%)	Real (%)	Δ Success (%)
GA	11.0	13.3	+2.3
QL	37.7	33.3	−4.4
Hybrid	48.0	36.7	−11.3

adaptive learning of QL.

5.1 Simulation Performance and Generalisation

After concluding the simulations, performance trends in the results were clear. With consistent differences across models. As seen in Table 1 and visually plotted in Figure 5, the hybrid model was able to outperform the other models, achieving the highest success rate. Particularly, in the 5 light condition, reaching 54% (approximately three times more than the QL at 17% and the GA at 15%) in the moderately complex environment (5 light), this shows the benefit of the hybrid approach. Potentially showing how the balance of structural diversity (evolution) and adaptive refining (learning) can lead to the best performance. Furthermore, this correlates with early studies that showed hybrid GA and reinforcement learning can improve learning efficiency and generalisation in robot navigation tasks (Iglesias et al., 2006). However, the 5 light environment might be the sweet spot for sufficient training diversity that enables generalisable behaviour while task complexity doesn't interfere with performance, as the 10 light environment saw no noticeable improvement for the hybrid model. This is interesting, as you might expect more training lights to equal deeper behaviours being learned, but it was not the case.

Training curve analysis further supports these findings; Figure 2 shows hybrid agents achieved the most consistent upward reward trends. While the 2 light condition reached the highest peak reward, the 5 light had the most stable and continued performance over time, supporting its generalisation advantage. The 10 light condition showed slower learning and a lower final reward, again pointing towards difficulty optimising the Q-tables for the more complicated environment.

Similarly, QL agents were also able to display good generalising performance, shining brightest in the 10 light condition, achieving a 50% success rate and beating the hybrid model. Although, this is what makes theoretical sense due to the way QL completes the task, with their ability to adapt through trial and error in their lifetime being what makes them well suited to learn flexible policies - and therefore perform well in diverse tasks. The QL performance was stable in the 2 light and 10 light conditions (46% / 50%) further supporting their ability to reliably perform the task. This is in line with findings from broader benchmarking efforts, such as those by Cobbe et al. (2019), which show that agents exposed to structured environmental diversity during training are more likely to develop robust and generalisable policies (Cobbe et al., 2019). Despite this, the 5 light condition saw a big drop in performance to 17%, which could be attributed to the increase in environment complexity from 2 lights to 5, forcing it to learn real light following behaviour.

Across all conditions the QL agents showed fast early improvements in Figure 4, with rewards stabilising after a few thousand episodes. The 10 light had a slower but overall steadier trajectory, eventually reaching similar levels than other conditions. The consistent convergence implies that QL agents were able to form flexible and reactive policies, even in the most complex environments. However, it was not without its issues, as early reward noise and oscillation were present, which could be traced to the coarse state discretisation, which can cause sharp policy changes in neighboring bins.

In large contrast but not entirely unexpectedly, GA agents suffered poor performance across all conditions, with success rates ranging from 7% to 15%. This low threshold clearly shows the limitations of pure evolutionary performance in this specific set up of the task. This comes as a consequence of GAs evolving fixed weight policies offline, where they are unable to adapt their behaviour after deployment. So in the task of generalising to unseen light positions it is evidently clear why this set up will fail. In original testing the training light positions were not randomised and the GAs had great performance, but testing highlighted that this was memorisation rather than the development of good generalising behaviour.

Figure 3 showcases the limitations of the GA model. Although the 2-light configuration initially improved, its performance peaked early and subsequently declined, indicating overfitting to simpler patterns that failed to generalise. The 5-light and 10-light GA agents plateaued quickly at much lower performance levels, suggesting poor scalability and an inability to evolve meaningful behaviours under increased environmental variation and complexity.

Overall, the found results all support the hypothesis that Q-learners, particularly the hybrid approach can better generalise to new unseen lights than those relying solely on evolution. The QL showed the most scalability, while the hybrid performed best in moderately complex environment and the GA were consistently outperformed, highlighting their limitations of their static policy design for the task.

5.2 Agent Behaviours

Alongside the quantitative results, agent behaviours were tracked for comparison to reveal insights into how each model translated into real robot control in the physical setting. Real trial results and routing of each models 10 light condition can be observed in Appendix B.

Firstly, GA agents as simulation results would imply, consistently exhibited inefficient behaviours, most commonly circling, overshooting or veering completely away from the light source. It was found that even small sensor changes or floor inconsistencies disrupted navigation, as seen in Figure 6. This was compounded, by the fact that GA policies are represented as fixed weight mappings from sensor to motor outputs, with no real time adjustment. As a result, agents were completely unable to recover or compensate once the trajectory was incorrect, leading to the majority of trials resulting in poor light seeking behaviour. This again highlited the core limitation of purely evolutionary strategy for this

task.

QL agents, in contrast demonstrated better performance and responsive patterns. There were instances of mid path corrections likely due to changing sensor readings, which unlike the GA, allowed them to recover from poor initial headings. This adaptability is why reinforcement learning is often used in navigation tasks for robotics. But, it was by no means perfect. In particular, the 5 light condition the behaviour was highly unstable, showing rapid turning and indecisiveness near the light, caused most likely by suboptimal state discretisation. Additionally, coarse binning can cause similar inputs to map to different states, which can cause rough policy decisions. Example behaviour of the QL can be seen in Figure 7.

Lastly, the hybrid agents showed the most consistent and also efficient behaviours. The trajectory was often smooth, unlike other models and showed minimal drift or hesitation. The 10-light Hybrids performance is visualised in Figure 8. This robustness is likely due to the combination of evolutionary structure shaping (via GA) and lifetime learning (via QL), which biases the agent toward more learnable Q-tables while allowing for adaptive fine-tuning during interaction, leading to the output of smooth trajectories and goal seeking movement.

However, the 5-light Hybrid agent, despite being the best performer in simulation, experienced a notable drop in physical success rate—down to 20%. This suggests possible over-fitting to the simulated environment or insufficient robustness in the evolved Q-table structure. In contrast, the 10-light Hybrid, while showing slightly lower simulation success, achieved more reliable and efficient real-world behaviour. This trade-off highlights the complex interplay between training diversity and policy generalisation: broader training can promote robustness but may slow or complicate convergence; narrower training can accelerate learning but at the risk of brittleness when the environment is novel or shifts.

To summarise, the behavioural observations aligned and reinforced the quantitative findings, GAs were fragile, QL agents were flexible but sensitive and the hybrid appeared to offer the most balance of stability and reliable control.

5.3 Why Hybridisation Improves Generalisation

The hybrid models success over other models in generalising, can be attributed to its dual adaptation mechanism and its alignment to the Baldwin effect. Within evolutionary computation, the Baldwin effect refers to when individuals that can learn more effectively during their lifetime are favoured over generations, even though the learned behaviours themselves are not directly inherited. Instead, evolution gradually selects for genotypes that produce more learnable phenotypes.

In this study, the Genetic Algorithm did not evolve complete solutions to the light-seeking task but instead evolved Q-table policy scaffolds - that the Q-learning process could optimise through environment interaction. This dual mechanism enabled evolution to bias the prior of each agent toward regions of the policy space that were not just high-performing, but also highly learnable. As a result, Hybrid agents benefited from global and population-level exploration (via mutation) and individual, episode level exploitation (via reward-based learning).

This two-tiered learning system allowed the Hybrid model to balance exploration and exploitation. The GA introduced structural diversity by mutating the Q-table across generations, maintaining variability in the agent population and enabling broad policy search. Q-Learning, in turn, locally refined each individual's policy using feedback from trial-and-error episodes, allowing for precise adaptation to sensor states. Yusof et al. also highlighted the benefits of hybrid reinforcement learning combined with unsupervised adaptation, especially for physical robot systems requiring high reactivity (Yusof et al., 2017). This leads to increased performance and the likelihood of robust behaviours developing, while mitigating the risk of early convergence.

However, the model's relative underperformance in the 10-light condition (compared to QL) suggests that this evolutionary shaping process becomes less efficient as task complexity increases. The larger state space introduced by high task diversity potentially weakens evolutionary pressure, making it harder to converge toward learnable structures within a limited number of generations. This shows the trade-off of the importance of tuning training diversity and population size to maximise the benefits of hybridisation.

5.4 Sim-to-Real Transfer Challenges

Transferring from simulation to reality and crossing the reality gap brought noticeable challenges and decreases in performance across the board. Recent work by Ouyang and Cui (2024) highlights how combining imitation learning with reinforcement learning can soften this gap, enabling robots to transfer skills more reliably from simulation to physical environments (Ouyang et al., n.d.).

The reality gap can clearly be observed in Table 4, where the hybrid model saw an -11.3% drop from simulation to real world testing. Additionally, the Q-learner saw a -4.4% drop showing the two models sensitivity to noise and inconsistency in the real world. The real world sensors bring latency, non-Gaussian noise and imperfections, along with motor inconsistencies, surface friction and degradation of hardware over time, real world trials had much more interference. This struggle can be seen in similar work by Chen et al. where a deep reinforcement learning approach was developed that similarly struggled with noisy transfer from simulation to physical sensors when using discretised control policies (G. Chen et al., 2020). Surprisingly, the GA saw an increase of 2.3%, despite lowest performance over all simulations. Although there was improvement it cannot be accredited to good performance, but rather the low baseline and coincidence.

Beyond algorithmic factors, hardware limitations also appeared, extended real-world testing led to physical wear, including motor degradation, which affected speed, symmetry, and reliability. Wait times for the robot to cool down to avoid burning out the motors led to drawn out trials and the wear likely introduced additional noise into the outcome metrics. Such inconsistencies show the need for robust and reliable hardware when evaluating hypothesis to avoid noisy or inaccurate data.

5.5 Statistical Comparisons

Paired t-tests (Table 2) reinforce the conclusion that Hybrid and QL models significantly outperform GA under various training conditions. Hybrid was significantly better than GA at 2 and 5 lights ($p = 0.0309$ and 0.0189, respectively), while QL was significantly better than GA at 10 lights ($p = 0.0474$). QL outperformed Hybrid at

5 lights (p = 0.0111), though performance between QL and Hybrid was statistically equivalent in other cases.

These results suggest that QL and Hybrid models are consistently more robust than GA, with Hybrid being the most versatile.

5.6 Computational Cost

A big trade-off with the hybrid model is the computational cost, which is important to consider. All models were trained to adequate depth to allow enough time for behaviours to emerge. However, each generation of the hybrid requires evaluating a population of Q-learners over multiple episodes, leading to long runtimes. Whereas, the QL trains one agent and is computationally cheaper. In this study the setup for GA was much more expensive than usual. The setup for GA is simple in comparison, but to allow for convergence on complex tasks, long training durations 200-500 generations were applied as slow evolution was observed in early testing.

Despite being more expensive, the hybrid training appears to be well justified in this scenario and potentially others that prioritise sim to real performance over computation speed.

5.7 Limitations and Future Work

The results from this study were promising and showed clear trends and insights into why certain models succeed or fail at the task, but there are several limitations that could be addressed in future work.

First of all, the Q-learning relied on a fixed, discrete action space and coarse state binning. This limits behavioural flexibility and the resolution of any learned policies, especially in the real world. Initial work for this study considered using a continuous action state space or to adopt a function approximation technique such as Deep Q-networks (DQNs) that could offer finer control and improved robustness and subseuently transfer to real. Deep Q-Networks (DQNs), offer scalable solutions by approximating Q-values using neural networks, enabling flexible policies in continuous state spaces (Mnih et al., 2015).

Next, the Genetic Algorithm only utilised mutation based variation. In future crossover or more advanced evolutionary strategies could be applied to improve the GAs performance and level the analysis playing field. This design choice likely limited the GAs exploration and contributed to the low success rates seen in Table 1. Considering the poor performance, early plateaus in training and poor fitness, the addition of crossover recombining high performing traits from individuals may improve the search process, convergence and behavioural diversity For example, multi-objective evolutionary strategies like NSGA-II have demonstrated the ability to maintain Pareto-optimal diversity while progressing toward optimal solutions, making them well-suited for navigating trade-offs in multi-dimensional controller design (Deb et al., 2002). But, it is not certain given the task demands and GAs design, if it would lead to higher success rates.

Third, in experiment set up the training lights were defined as 2, 5 and capped at 10, with the aim of sufficiently showing trends in the different conditions. However, 10 may not have provided enough environmental variation to develop deep policies, expanding the training set would be a clear way to further test the results gathered here.

Additionally, the real world testing also faced constraints, motor burnout was a worry and led to a reduction in the planned amount of real world trials. While it didn't effect the analysis for full depth of sim to real transfer increasing the physical trials with more reliable hardware would concrete the analysis in future studies.

Lastly, similar to the training light, for clear comparison only five "unseen" fixed light positions were used to test the learned generalising behaviour. Even though the positions were spread in all quadrants, future evaluations should test a wider range for deeper study.

To bring it together, the current results are encouraging, but further refinements in algorithm design, environment setup and hardware integration are needed to explore this task fully.

6 Conclusion

To conclude, this study compared three learning strategies of Genetic Algorithms, Q-Learning and a hybrid QL+GA approach for training a Braitenberg vehicle to perform light seeking navigation. The goal was to assess each model in simulation and observe how they transferred their learned behaviours to real world conditions - crossing the reality gap.

The found results strongly supported the original hypothesis, with the hybrid model emerging as the most promising approach. Showing it's ability to balance stability and efficient behaviour, in simulations and the real world alike. The analysis revealed the important consideration of computational cost, the QL was much less expensive and offered similar results in the 10 light condition. So, in this setting, when computational speed isn't a concern, the hybrid model is the best choice. GA was determined to be ill-performing and unsuited for the task in simulation and the real world, repeating the study the GA setup would definitely be reviewed for a fairer comparison.

In summary, combining evolutionary search and local learning leads to the most optimal adaptation. The hybrid model shows potential for sim to real robotics when generalising is important and training time isn't a pressure. Further improvements to state representation are necessary, but this study showed valuable insights into adaptive, learning robot systems.

A Code Availability

All code used to generate the results and figures in this report is provided in the accompanying `.py` file submitted alongside the report PDF. These include implementations of the Genetic Algorithm, Q-Learning agent, Hybrid GA+QL agent, and the associated analysis and plotting scripts.

Each file is clearly named to reflect its contents and corresponds to the experimental sections discussed in the main report. Instructions for reproducing the results are provided as comments within the scripts.

B Real Robot 10 Light Test Images

This appendix presents photographic evidence of real-world robot trials conducted with the GA, Q-Learning, and Hybrid models. All robots started from the varied range described around the point (3,3), and were tested against an unseen light position marked by a headlamp. The photos illustrate the final positions reached by each agent type.

Figure 6. Final position of GA agent after navigating toward the light source. The agent failed to approach the light directly, following a looping behaviour and finishing 73cm from Light 2 on this test run.

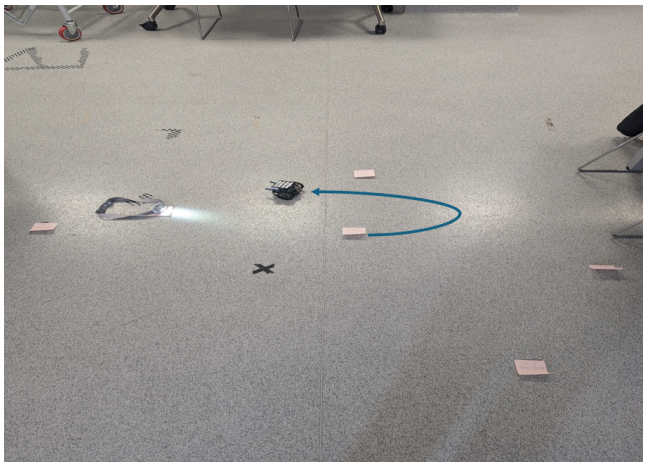

Figure 7. Final position of Q-Learning agent. The robot approached the light more closely than GA but still showed some deviation from an optimal path, finishing 34cm away from Light 1 on the pictured trial.

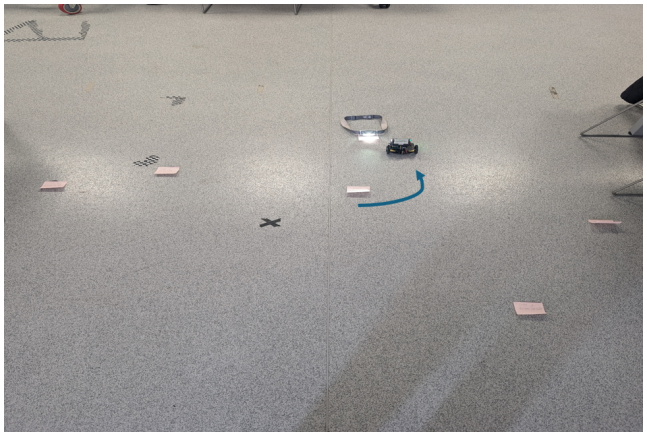

Figure 8. Final position of Hybrid GA+QL agent. This model successfully reached closest to the light source, following a smoother and more direct trajectory more consistently than other models.

C Appendix: Real-World Robot Control Code

This section includes the Python code used to control the physical Braitenberg robot during real-world trials. The script executes a fixed policy based on an evolved or learned genotype and uses analog light sensor readings to drive the motors.

```
"""
Braitenberg robot for crossing the reality gap.
Real-world trial script for evolved/learned genotypes.

Author: Dexter R Shepherd + integrated by you!
"""

from EduBot_CP import wheelBot
import time
import board
from analogio import AnalogIn

# Pin setup
s1 = AnalogIn(board.GP28)  # Left sensor
s2 = AnalogIn(board.GP26)  # Right sensor

# Load genotype:
genotype = [3.031324322596496, 4.135784637017988,
     0.7798305835689324, 3.0161060732233507,
     4.479874266943583, 4.26612122613619]

sensor_gain = 0.5
motor_scale = 0.5

# Robot object
bot = wheelBot(board_type="pico_1")

# Sensor reading function
def get_intensity(pin):
    return (pin.value * 3.3) / 65536

# Calculate motor speeds using genotype
def calc(geno):
    w_ll, w_lr, w_rl, w_rr, bl, br = geno

    il = get_intensity(s1) * 10
    ir = get_intensity(s2) * 10

    # Weighted motor activation
    lm = il * w_ll + ir * w_rl + bl
    rm = il * w_lr + ir * w_rr + br

    return lm, rm

# Running The Trial
trial_duration = 20  # seconds
start_time = time.monotonic()

while time.monotonic() - start_time < trial_duration:
    left_speed, right_speed = calc(genotype)

    # Clamp motor output for safety
    left_speed = min(max(left_speed, -100), 100)
    right_speed = min(max(right_speed, -100), 100)

    # Scale and apply
    bot.motor1_move(left_speed * motor_scale)
    bot.motor2_move(right_speed * motor_scale)

    time.sleep(0.05)  # Small update delay

# Stop after trial
bot.stop()
```

Listing 1: Python control script for real-world Braitenberg robot trials.

References

Carvalho Santos, Valéria de et al. (2014). "A hybrid GA-ANN approach for autonomous robots topological navigation". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. SAC '14'. Gyeongju, Republic of Korea: Association for Computing Machinery, pp. 148–153. ISBN: 9781450324694. DOI: 10.1145/2554850.2554990. URL: https://doi.org/10.1145/2554850.2554990.

Chen, Chunlin, Han-Xiong Li, and Daoyi Dong (2008). "Hybrid Control for Robot Navigation - A Hierarchical Q-Learning Algorithm". In: *IEEE Robotics Automation Magazine* 15.2, pp. 37–47. DOI: 10.1109/MRA.2008.921541.

Chen, Guangda et al. (2020). "Robot navigation with map-based deep reinforcement learning". In: *2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, pp. 1–6.

Cobbe, Karl et al. (2019). "Quantifying generalization in reinforcement learning". In: *International conference on machine learning*. PMLR, pp. 1282–1289.

Deb, K. et al. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. DOI: 10.1109/4235.996017.

Huang, L. et al. (2016). "Path Navigation For Indoor Robot With Q-Learning". In: *Intelligent Automation & Soft Computing* 22.2, pp. 317–323. DOI: 10.1080/10798587.2015.1095485. URL: https://doi.org/10.1080/10798587.2015.1095485.

Iglesias, Roberto et al. (2006). "Combining reinforcement learning and genetic algorithms to learn behaviours in mobile robotics." In: *ICINCO-RA*, pp. 188–195.

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *nature* 518.7540, pp. 529–533.

Ouyang, Yutao and Jingzhi Cui (n.d.). "Bridging the Sim-to-Real Gap for Efficient and Robust Robotic Skill Acquisition". In: *Tsinghua University Course: Advanced Machine Learning*.

Patle, B. K. et al. (2022). "Hybrid FA-GA Controller for Path Planning of Mobile Robot". In: *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, pp. 1–6. DOI: 10.1109/ICICCSP53532.2022.9862422.

Wicaksono, Handy (2011). "Q Learning Behavior on Autonomous Navigation of Physical Robot". In: *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 50–54. DOI: 10.1109/URAI.2011.6145931. URL: https://doi.org/10.1109/URAI.2011.6145931.

Yusof, Yusman, HM Asri H Mansor, and HM Dani Baba (2017). "Applying Hybrid Reinforcement and Unsupervised Wieghtless Neural Network Learning Algorithm on Autonomous Mobile Robot Navigation." In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 9.1-3, pp. 133–138.