From Nectar to Network: Training an Autonomous Hummingbird

A Reinforcement Learning Approach to 3D Hummingbird Foraging

AE4350: Bio-inspired Intelligence and learning for Aerospace Applications

Mauritius van Maurik

Delft University of Technology

From Nectar to Network: Training an Autonomous Hummingbird

A Reinforcement Learning Approach to 3D Hummingbird Foraging

Mauritius van Maurik
(6328369)

Responsible Instructor:	Dr. G.C.H.E. de Croon
Instructor:	Dr.ir. E. van Kampen
Project Duration:	June, 2025 - August, 2025
Faculty:	Faculty of Aerospace Engineering, Delft

Cover: Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Contents

1 Introduction

1.1. The Hummingbird as a Model

The hummingbird's unique ability to hover is the result of natural engineering, made possible by its flight mechanics. Birds commonly generate lift on the downstroke of their wings; the hummingbird can generate lift on both the upstroke and the downstroke. This is achieved through the rotation of their wings in a figure-eight pattern, therefore allowing them to remain stationary in the air with incredible precision [5, 1]. This hovering capability allows the hummingbird to access nectar from flowers that other birds cannot reach.

This aerial ability comes at the significant cost of a very high metabolic rate, which is one of the highest among all vertebrates [8, 1, 4]. Hummingbirds sustain extremely high rates of aerobic metabolism, fueled by their fast wingbeats during flight [1]. While specific numbers vary by species and activity, hummingbirds wings can beat up to 200 times per second, and their heart rates up to 1200 beats per minute, reflecting the intense physiological demands of flight [1]. The hummingbird's heart is proportionally large relative to its body size, which it requires for its high metabolic demands [1, 4]. This physiological demand requires a constant and substantial energy supply in order for the bird to sustain its being [8, 7].

Therefore, the hummingbird is required to continuously forage. The bird must visit hundreds to thousands of flowers every day, such as estimates of 1,000 to 2,000 flowers daily, to obtain nectar for its high energy requirements [3]. A Rufous Hummingbird, for instance, can consume about 1.4 times its own body weight in nectar daily [2]. Other estimates suggest consumption between 1.5 to 3 times its body weight in nectar each day [3]. A highly efficient foraging strategy is needed since the bird has limited energy storage capacity and high fixed metabolic costs, making it sensitive to daily fluctuations in energy availability and only a few hours away from energy depletion if food is unavailable [7, 5].

The hummingbird's way of life showcases a challenge similar to those in the field of autonomous systems. The hummingbird's behavior can be simulated as an optimization problem: an agent having to navigate a 3D environment to maximize its energy intake whilst minimizing metabolic cost. This involves elaborate decision-making, having to balance immediate reward of a nearby flower against a potential for a richer source farther away. The agent must manage its internal resource (energy), learn an efficient path finding strategy, and operate under the risk of system failure (starvation).

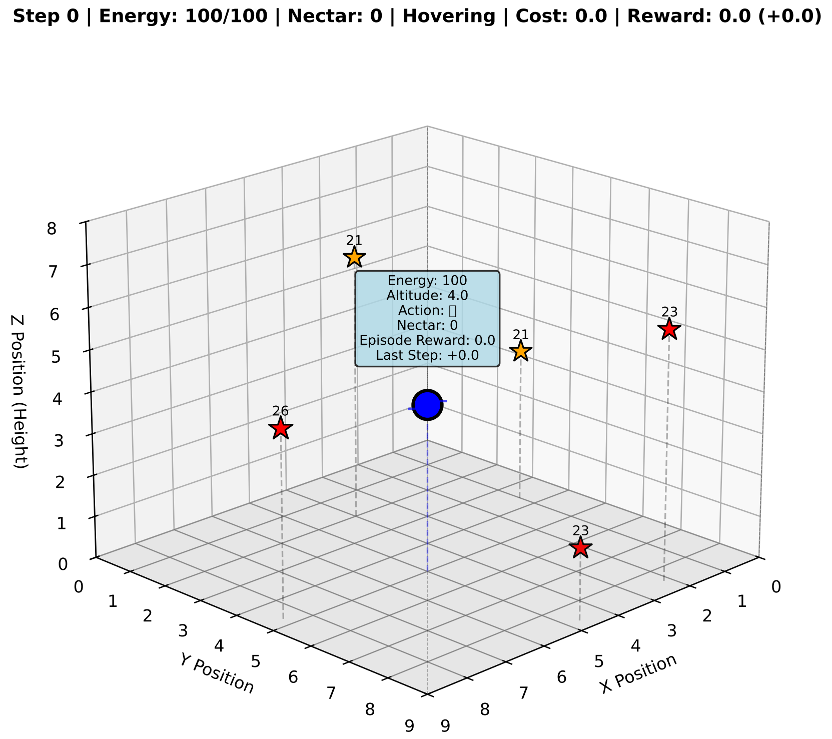
This project aims to use the hummingbird's foraging challenge as a basis for developing and testing a Reinforcement Learning agent. The objective is to train an agent using Proximal Policy Optimization (PPO) with a 3D simulation to see whether it can learn the intelligent and sustainable foraging strategies found in nature. The complete implementation is available for review by visit the GitHub repository here.

2 Methodology

The development of the foraging agent is approached by firstly creating a realistic bio-inspired simulation. Thereafter a suitable Reinforcement Learning (RL) algorithm is deployed to train the agent within the environment. This chapter aims to discuss the three components of the methodology: the environment, the dynamic resource model and the learning agent.

2.1. The Hummingbird Simulation Environment

To simulate the foraging behavior of a hummingbird, a custom 3D reinforcement learning environment, consisting out of a 10x10x8 grid shown in Figure 2.1, developed using the Gymnasium framework [10]. This environment provides the agent with a set of sensory information and a defined set of the actions for the interactions with its world. The agent's observation space is defined as a vector containing its own position and energy level, as well as the 3D coordinates and current nectar status of all flowers in the environment, shown in Table 2.1. Although a real hummingbird does not possess this perfect global knowledge, this design choice is made to simplify the learning problem. Our goal was to investigate the agent's ability to learn an efficient foraging strategy rather than its ability to perceive its environment from raw sensory data. This approach is analogous to simulating a hummingbird's well-developed spatial memory, which allows it to form a cognitive map of its territory and plan efficient flight paths.



Step 0 | Energy: 100/100 | Nectar: 0 | Hovering | Cost: 0.0 | Reward: 0.0 (+0.0)

Energy: 100
Altitude: 4.0
Action: □
Nectar: 0
Episode Reward: 0.0
Last Step: +0.0

Figure 2.1: A screenshot of the 'ComplexHummingbird3DMatplotlibEnv'. The blue sphere represents the hummingbird agent, and the starts represent flowers with available nectar. The axes indicate the 3D grid space.

Table 2.1: Hierarchical Observation Space Structure

Component	Feature	Shape	Description
Agent	Position	‘(3,)‘	3D coordinates (x, y, z) of the agent.
	Energy	‘(1,)‘	Current energy level representing survival capacity.
Flowers	Coordinates	‘(N, 3)‘	Position of each of the N flowers.
(per flower (N))	Nectar	‘(N, 1)‘	Current nectar availability.
	Cooldown	‘(N, 1)‘	Remaining regeneration time.
	Available	‘(N, 1)‘	Binary flag indicating if the flower is ready.
Environment	Costs	‘(3,)‘	Metabolic, horizontal, and vertical movement costs.
Parameters	Physics	‘(3,)‘	Gravity, interaction radius, and max energy.

The agent interacts with its environment through a discrete set of seven actions. Each action corresponds to a unit movement in the 3D grid and has an associated cost with it. The action space is shown in Table 2.2. Note that there is an additional metabolic cost of 0.2 applied every timestep in order to simulate the metabolic cost of the hummingbirds high metabolism.

Table 2.2: Discrete Action Space with Energy Costs

Action ID	Maneuver	Energy Cost
0	Forward	0.8
1	Backward	0.8
2	Left	0.8
3	Right	0.8
4	Ascend	1.2
5	Descend	0.5
6	Hover	2.0

A main feature of the environment is its physics-informed energy model, designed to reflect the real-world metabolic challenges of a hummingbird's flight. The costs are structured to enforce a learning strategy that mirrors biological efficiency. Hovering is the most metabolically expensive action, reflecting the high energy demand of stationary flight. Ascending costs more than descending, as the former works against gravity while the latter is assisted by it. Horizontal movement represents a baseline expenditure for directed flight.

2.2. Dynamics Resource Model

The environment's primary resources is nectar, which is found in flowers. The properties of the nectar are governed by four key parameters, which are defined in Table 2.3. The placement and the state of flowers are decided by a ”Fair Flower Distribution” algorithm, which is executed at the start of each episode. The aim of this algorithm is to create balanced and solvable environments by enforcing the following principles:

- **Regional Distribution:** The 3D space is divided into eight distinct sectors. The algorithm distributes the flowers among these regions to prevent clustering and encourage the agent to explore the entire volume of the environment.
- **Minimum Spacing:** A minimum distance is enforced between any two flowers, ensuring that the agent must engage in meaningful travel and path-planning to forage effectively.
- **Energy Accessibility:** Every flower's position is validated by an energy accessibility check ($is - energy - accessible$). This check estimates the energy cost to travel from the agent's starting point to the potential flower location. Only flowers deemed reachable within a reasonable energy budget are spawned, guaranteeing that every episode is solvable and fair.

The distribution systems forces the agent to learn a generalized exploration strategy rather then memorizing static spawn patterns. It increases the complexity of the task by making spatial awareness and efficient 3D path finding essential for survival.

Table 2.3: Dynamic Resource Model Parameters

Parameter Name	Value	Function
`NECTAR_GAIN`	30	Determines the amount of energy the agent receives from a successful flower visit.
`MAX_NECTAR`	15	Defines the maximum nectar capacity of a single flower.
`NECTAR_REGEN_RATE`	0.3	Controls the rate at which a flower replenishes its nectar after being visited.
`FLOWER_COOLDOWN_TIME`	15	Sets the inactive period (in timesteps) for a flower after its nectar has been completely depleted.

2.3. The Learning Agent and Training Protocol

The agent is trained using Proximal Policy Optimization (PPO), this algorithm has been chosen due to its stability and performance [6]. A MultiInputPolicy method is employed due to the dictionary-based observation space. This method allows the agent to process the separate agent and flower data streams before combining them to a final decision. The agent's learning is shaped by a reward function with several components aimed at encouraging effective foraging behavior and survival, shown in Table 2.5. The main hyperparameter of the PPO algorithm are obtained through experimental tuning. The final most effective configuration used for training the agent is shown in Table 2.4.

Table 2.5: Reward Function Components and Rationale

Component	Reward	Description
Nectar Collection	$[+1, +15]$	Rewards the agent for successful foraging.
Discovery Bonus	$+5$	Encourages the exploration of new flowers.
Inefficiency Penalty	-2	Discourages revisiting empty or inactive flowers.
Step Penalty	-0.05	Encourages the agent to find an efficient foraging strategy.
Survival Bonus	$+100$	Strong positive reinforcement for surviving the episode.
Death Penalty	-100	Penalty for energy depletion, discouraging inefficient behavior.

2.4. Evaluation Protocol

An evaluation protocol is developed to provide an unbiased and statistically significant assessment of the agent's performance. During the evaluation, the agent's learning is frozen to measure its ability to generalize its learned policy to unseen situations. Each model is evaluated on a total of 1000 independent episodes, split into 10 separate runs of 100 episodes each, to ensure the reliability of the final performance metrics. The metrics used during evaluation are the Survival Rate, Total Nectar Collected, Total Rewards, and Episode Length.

Table 2.4: Key Hyperparameter for the PPO Agent Training

Parameter	**Value**	**Description**
Core PPO Parameters		
Learning Rate	$3 \times 10^{-4} \rightarrow 1 \times 10^{-6}$	A linear learning rate decay schedule to encourage a more stable convergence.
γ (gamma)	0.999	The discount factor, which determines the importance of future rewards. A higher value prioritizes long-term rewards.
ent_{coef}	0.02	The entropy coefficient, which encourages exploration by penalizing a deterministic policy.
λ (gae_lambda)	0.95	The Generalized Advantage Estimation (GAE) lambda parameter for balancing bias and variance in advantage estimation.
clip_range	$0.2 \rightarrow 0.1$	The PPO clipping parameter, which linearly decays to control the magnitude of policy updates.
vf_{coef}	0.5	The coefficient for the value function loss, balancing its importance against the policy loss.
max_grad_norm	0.5	The maximum L2 norm of the gradients, used to prevent exploding gradients during training.
Training and Batching		
n_envs	25	The number of parallel environments used for efficient data collection.
n_steps	81	The number of environment steps collected from each environment before a policy update.
Total Steps per Update	2025	The total number of steps collected across all parallel environments (81×25).
n_epochs	10	The number of times the agent iterates over the collected data for policy and value function updates.
Batch Size	225	The mini-batch size used for each gradient update from the collected data.
Observation and Policy		
Observation Normalization	Disabled	The agent's observations are not normalized.
Policy Architecture	`MultiInputPolicy`	A feedforward neural network with three hidden layers of size [512, 256, 128] for both the policy and value functions.
Activation Function	`nn.Tanh`	The hyperbolic tangent activation function used in the policy network.
use_sde	False	State-dependent exploration is disabled to use a more standard PPO approach.

3 Results and Analysis

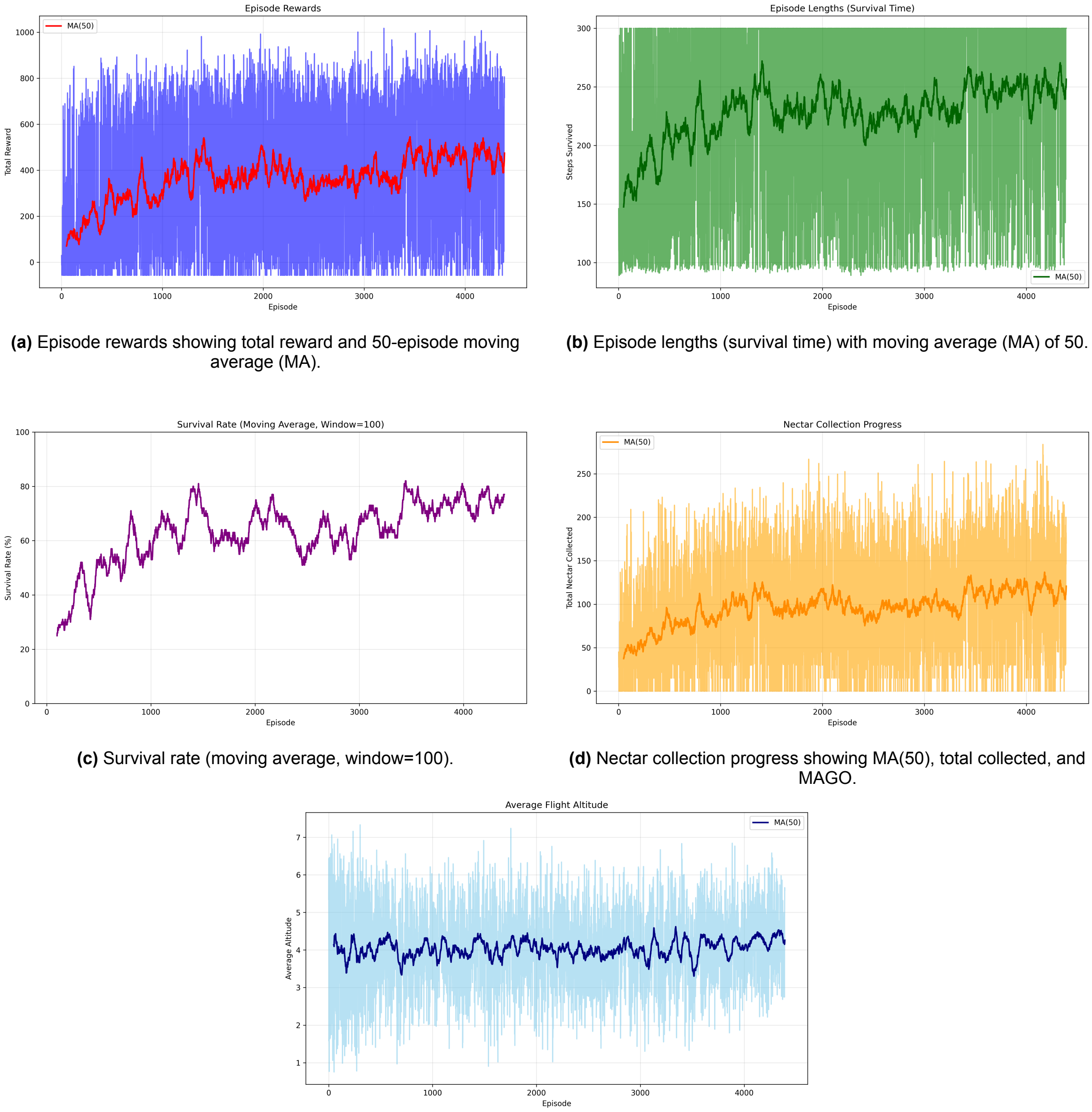
This chapter presents the performance of the trained PPO agent. Firstly a baseline model performance from the stable training protocol is discussed. After which a sensitivity analysis is conducted in order to investigate how the agent's performance is affected by changes its environments.

3.1. Training Performance and Learning Curves

Key performance metrics are tracked throughout the training process in order to visualize the agent's learning. Figure 3.1 show the progression of reward, survival time, survival rate, nectar collection, and flight altitude throughout the training episodes. The learning curves demonstrate several key aspects of the agent's performance:

- The reward progression (Fig.3.1a) shows a notable learning trend. The total reward, shown by the red line (which indicates the moving average) increases significantly during the initial 1500 episodes, rising from 100 to over 400. After this initial stage the learning reward stabilizes but fluctuates from 300 to 500 for the remainder of the training. This stabilization of the reward shows the converging of a successful strategy.
- Episode lengths (Fig.3.1b) showcase an improvement in the agent's ability to survive. The moving average of steps survives (indicated by the green line) shows a rapid increase.
- The survival rate (Fig.3.1c) reveals a similar learning curve to the episode length. The moving average shows that the agent's survival rate increased from under 30% to a stable range of 60% to 80% after about 1500 episodes. This metric directly correlates with the agent's improved ability to survive within the environment for longer durations.
- Nectar collection metrics (Fig.3.1d) demonstrate a positive learning progression in the agent's ability to find flowers. The total nectar collected per episode (orange moving average) rises from a low of around 50 to a peak near 125, before stabilizing and fluctuating in the 100-125 range. This indicates the agent learned to locate and collect nectar efficiently, contributing directly to the observed increase in reward.
- Altitude control (Fig.3.1e) suggests that the agent maintained a consistent flight altitude throughout the training. The average flight altitude (dark blue moving average) remains relatively stable, hovering around 4.0 units with minor fluctuations. This demonstrates that the agent learned to control its vertical position and maintained a consistent flight path.

The figures indicate that the agent's training progresses toward convergence on a stable policy.



(a) Episode rewards showing total reward and 50-episode moving average (MA).

(b) Episode lengths (survival time) with moving average (MA) of 50.

(c) Survival rate (moving average, window=100).

(d) Nectar collection progress showing MA(50), total collected, and MAGO.

(e) Average flight altitude with MA(50).

Figure 3.1: Training performance metrics showing the agent's learning progression across 4,000 episodes. (a) Reward accumulation, (b) Survival duration, (c) Survival rate, (d) Nectar collection efficiency, and (e) Altitude control.

3.2. Baseline Performance of the Final Model

The baseline model for this study is an agent trained within an environment containing five flowers. This specific amount of flowers is chosen as it represents a meaningful balance between simplicity and complexity. Environments with fewer flowers might allow the agent to learn a basic, non-generalizable policy, while a larger number of flowers (10+) could overwhelm the agent, causing it to explore and forage less effectively. The PPO agent is trained for 25 million time steps, after which its performance is evaluated using the previously described methodology.

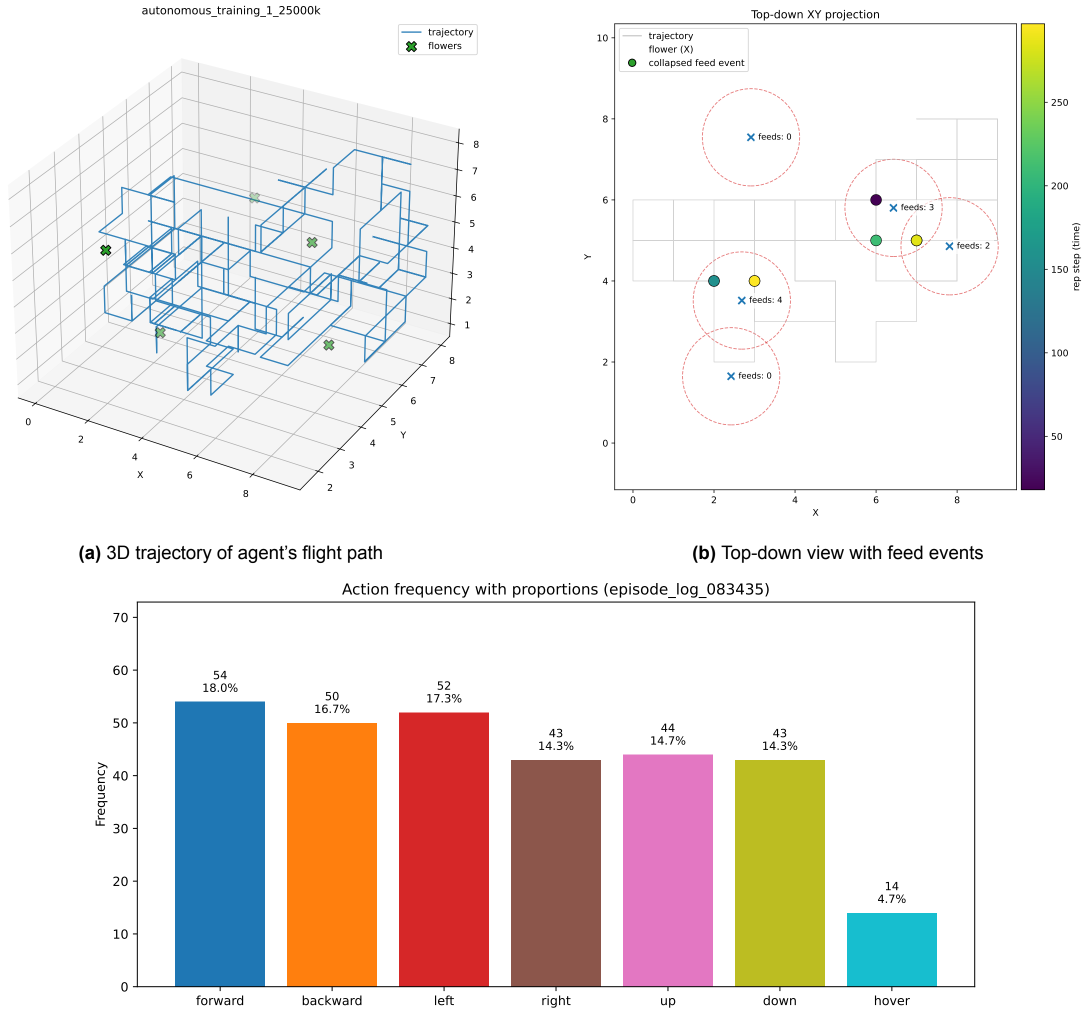
The evaluation results confirm that the agent learned a successful foraging policy. The model achieved a mean survival rate of $47.80\,\%$ ($\sigma = 4.47\,\%$) and demonstrated resource utilization by collecting a mean of 53.08 nectar per episode. A key finding is the strong positive correlation of 0.82 between the episode length and total nectar collected, which indicates that the agent's ability to survive is directly tied to its ability to collect resources.

The agent's trajectory, shown in Figure 3.2, reveals a purposeful and cyclical flight path instead of a random search. The agent focused its foraging on just three of the five flowers, a strategy of partial exploitation that proved sufficient for survival. This behavior is based on a learned "memory," as the agent repeatedly returns to the same flowers.

This traplining behavior is a well-known foraging strategy in hummingbirds, who use spatial memory to efficiently revisit food sources [9]. This learned memory is the main reason for the strong link between the agent's survival time and the amount of nectar it collects.

The agent's action history, also shown in Figure 3.2, provides an interesting insight: its primary movements are horizontal and vertical, and hovering is its least frequent action at just $4.7\,\%$. This confirms that the agent has learned to avoid the most energy-expensive action in its environment (hovering cost: 2), showcasing a rational and energy-efficient policy.

The analysis confirms that the PPO agent has learned a non-trivial and effective foraging strategy that is a direct consequence of the environment's reward structure. The agent's ability to exhibit memory-based behavior and its energy-conscious movement policy highlight a significant understanding of the task, even though it ultimately settled on a locally optimal solution.



(a) 3D trajectory of agent's flight path

(b) Top-down view with feed events

(c) Action frequency distribution with proportions

Figure 3.2: Analysis of agent behavior showing (a) 3D flight trajectory, (b) top-down projection with feed events demonstrating memory-based re-visitation, and (c) action frequency distribution highlighting energy-efficient movement strategies.

3.3. Sensitivity Analysis

A sensitivity analysis is performed to evaluate the robustness of the trained policy by systematically varying key environmental parameters: the number of flowers and the agent's initial energy level. The results of each analysis are presented in the following sub-sections.

3.3.1. Sensitivity to Number of Flowers

The amount of flowers in the environment is varied from 2 to 10 to evaluate the robustness of the trained policy. The results, summarized in Table 3.1, reveal a non-linear relationship between resource availability and agent performance.

The most notable finding is the survival rate of $100.00\,\%$ observed in the 2-flower environment. Further investigation of this agent's flightpath has shown that agent is continuously looping between the two static flowers in order to maintain maintain its energy levels.

As the number of flowers increases from 4 to 10, the problem complexity increases. The agent's survival rate generally decreases, ranging from $39.70\,\%$ to $53.00\,\%$. This decline is expected, as the agent must now learn a more complex foraging strategy to navigate a larger and more complex space. However, the mean nectar collected per episode generally increases with the number of flowers, peaking at 72.74 in the 8-flower environment. This suggests that while the agent may not always survive for the full episode, it is effective at collecting nectar when it successfully finds a resource.

Furthermore, a strong positive correlation between episode length and nectar collection is observed in all environments with 4 or more flowers. This indicates that for these more challenging scenarios, survival is directly tied to the agent's foraging efficiency; the longer the agent stays alive, the more nectar it collects, creating a virtuous cycle.

Table 3.1: Summary of the sensitivity analysis results, showing the performance of the trained PPO agent in environments with a varying number of flowers. The evaluation was performed over 200 episodes for each model.

Number of Flowers	Mean Survival Rate (%)	Mean Nectar Collected	Corr. (Length vs. Nectar)
2	100.00	45.89	N/A
4	41.70	47.66	0.81
5	47.80	53.08	0.82
6	43.40	57.19	0.85
8	53.00	72.74	0.76
10	39.70	66.14	0.76

3.3.2. Sensitivity to Initial Energy Level

A second sensitivity analysis is conducted to evaluate the robustness of the agent's policy to changes in its initial energy level. The agent, trained on a baseline of 100 energy, is evaluated in environments where the initial energy is varied from 25 to 150. The results, summarized in Table 3.2 showcase a positive relationship between the agent's initial energy and its performance.

As the maximum energy increases from 25 to 150, the mean survival rate consistently climbs from $10.0\,\%$ to a high of $76.0\,\%$. This finding is a direct consequence of the agent having a larger energy buffer, which allows it to execute its foraging strategy without succumbing to energy depletion. Concurrently, the mean nectar collected per episode also increases with initial energy, suggesting that the agent's ability to forage is directly tied to its survival time.

Table 3.2: Sensitivity Analysis: Agent Performance vs. Maximum Energy

Max Energy	Survival Rate (%)	Mean Nectar Collected
25	10.0	15.78
50	12.0	27.43
75	32.0	46.05
100 (Baseline)	52.0	64.69
125	60.0	64.84
150	76.0	64.77

4 Conclusion

The primary objective of this project is to apply a reinforcement learning approach to a hummingbird model aiming to solve the foraging problem. The chosen methodology, which combined a physics-informed simulation with the Proximal Policy Optimization (PPO) algorithm, proved to be a viable solution.

The training performance shows that the agent successfully learned an effective navigation and foraging strategy. This is proven by the consistent increase in both nectar collection and survival rate over the training period. The agent's final policy revealed two insights into its learned behavior.

Firstly, the agent developed an energy-efficient strategy that minimizes costly actions, such as hovering and long-distance travel. This behavior reflects the metabolic constraints of a real hummingbird, showing the agent's ability to modify its actions to optimize for energy conservation.

Secondly, the agent learned a ”memory-based” behavior by repeatedly visiting a familiar set of flowers. This strategic foraging, known as traplining in the biological literature, led to a locally optimal solution. The behavior demonstrates the agent's ability to create a consistent, repeatable foraging circuit, which is an essential trait of efficient foraging animals.

The conducted sensitivity analysis showcases the robustness of the agent's policy. The agent's performance remained consistent across variations in the number of flowers and its initial energy level. This showcases that the agent is not overly dependent on its training environment. An especially interesting finding is the agent's 100% survival rate in a simplified two-flower environment. This indicates that it learned a truly optimal and deterministic strategy for that specific environment.

It is essential to acknowledge the limitations of the current approach. The agent's tendency to settle for a local optimum suggests that the PPO algorithm, as implemented, may lack the exploratory capabilities needed to find a globally optimal solution in more complex environments.

Further improvements can be made in future work to enhance this project. Firstly, alternative RL algorithms could be investigated that are better suited for complex exploration. For instance, Soft Actor-Critic (SAC), deep Q-learning, or evolutionary strategies could be evaluated. These algorithms could be used to find a better solution which is not a local optimum.

Secondly, the simulation environment can be improved to better reflect real-world foraging. The environmental complexity can be increased by including flowers with varying nectar levels and different placements, such as flower clusters. Moreover, hummingbirds are known to compete with each other and are highly territorial; this behavior could be introduced to mimic competition for resources.

Lastly, the learned policy could be deployed on a physical robot, such as a bio-inspired flapping-wing drone. This would not only test the scalability of the solution but also provide insights into the sim-to-real gap.

References

[1] Angelica R Chavez, Adolfo Rico-Guevara, and Douglas L Altshuler. “Aerobic power and flight capacity in birds: a hummingbird perspective”. In: *The Journal of Experimental Biology* 221.1 (2018), jeb162693. DOI: 10.1242/jeb.162693.

[2] Geoffrey L Holroyd and J Cam Finlay. “Daily consumption of nectar by Rufous Hummingbirds at a feeder in Victoria, British Columbia”. In: *British Columbia Birds* 26 (2016), pp. 32–34.

[3] Christina N Lotz and Alissa Schondelmayer. “How much can hummingbirds increase nectar consumption during extreme energetic demand?” In: *Journal of Ornithology* 157.3 (2016), pp. 749–756. DOI: 10.1007/s10336-016-1339-3.

[4] Marshall J Ray et al. “Genomic insights into metabolic flux in hummingbirds”. In: *Genome Research* 33.7 (2023), pp. 1107–1123. DOI: 10.1101/gr.276779.122.

[5] Alyssa Sargent, Derrick Groom, and Alejandro Rico-Guevara. “Locomotion and Energetics of Divergent Foraging Strategies in Hummingbirds: A Review”. In: *Integrative and Comparative Biology* 61 (June 2021). DOI: 10.1093/icb/icab124.

[6] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: (July 2017). DOI: 10.48550/arXiv.1707.06347.

[7] Anusha Shankar, Kenneth J Welch, and Douglas L Altshuler. “Hummingbirds budget energy flexibly in response to changing resources”. In: *Functional Ecology* 34.1 (2019), pp. 152–164. DOI: 10.1111/1365-2435.13454.

[8] Raul K Suarez, Charles-André Darveau, and Peter W Hochachka. “Fuel selection in rufous hummingbirds: ecological implications of metabolic biochemistry”. In: *Proceedings of the National Academy of Sciences* 87.23 (1990), pp. 9207–9210. DOI: 10.1073/pnas.87.23.9207.

[9] Maria Cristina Tello-Ramos, T. Andrew Hurly, and Susan D. Healy. “Traplining in hummingbirds: flying short-distance sequences among several locations”. In: *Behavioral Ecology* 26.3 (Mar. 2015), pp. 812–819. ISSN: 1045-2249. DOI: 10.1093/beheco/arv014. eprint: https://academic.oup.com/beheco/article-pdf/26/3/812/13897864/arv014.pdf. URL: https://doi.org/10.1093/beheco/arv014.

[10] Emmanuel Todorov et al. “Gymnasium: A Standard Interface for Reinforcement Learning Environments”. In: *arXiv preprint arXiv:2407.17032* (2024).